

# Couche Interface Réseau

---

*Martin Delacourt, Université d'Orléans*

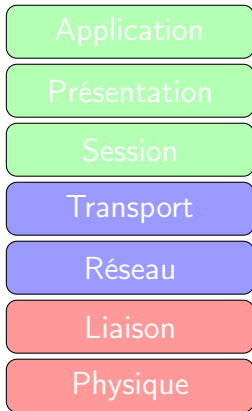
L3 Réseaux — 2023/2024



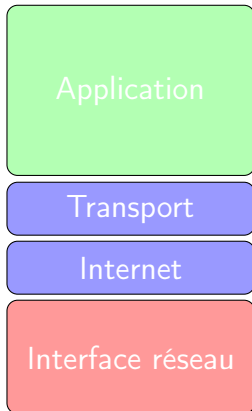
# Les modèles

---

OSI

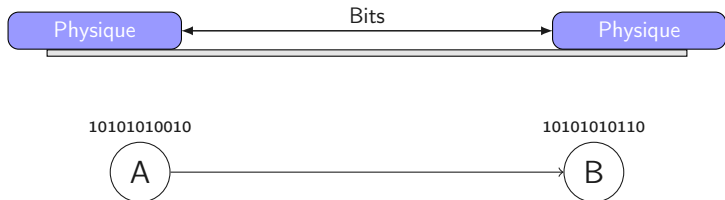


TCP/IP



# Couche physique (Physical)

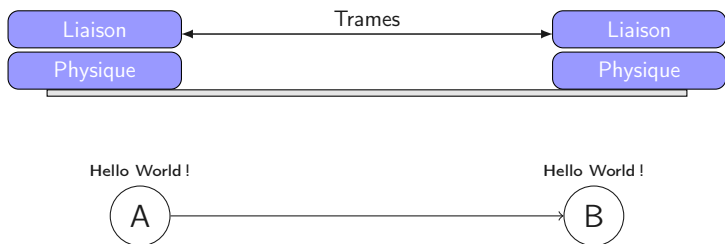
---



- Lien **point à point** entre deux interfaces, pas d'intermédiaires ni de redirection attendue. Lien direct : câble (paire torsadée, fibre), air (wifi),...
- Transmission de bits uniquement.
- Potentiellement des erreurs.

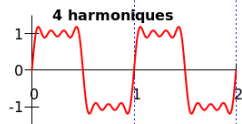
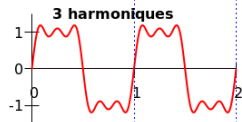
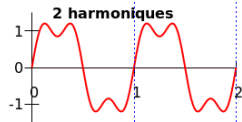
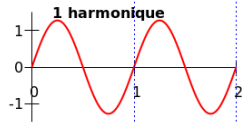
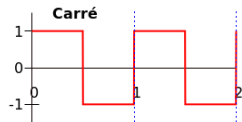
# Couche Liaison (Data link)

---



- Toujours lien **point à point** entre deux interfaces reliées. Gestion de l'accès au médium pour éviter les interférences.
- Transmission de trames.
- Ni erreur, ni perte, ni duplication.

# Couche physique



# Le médium

---

La couche physique a pour but de transporter des séquences de bits sur différents médias :

- Paire de cuivre (ADSL)
- Fibre optique
- Air (Wifi)

Un émetteur et un récepteur directement liés par le médium,  
sans intermédiaire.

001011110101111001

# Transformation du signal

---

Pour envoyer une séquence de bits dans un signal électrique :

- une horloge
- deux (plus ?) valeurs de tension représentant 0 et 1
- signal en créneau.

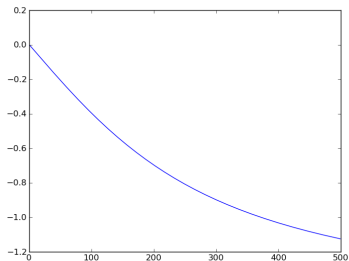
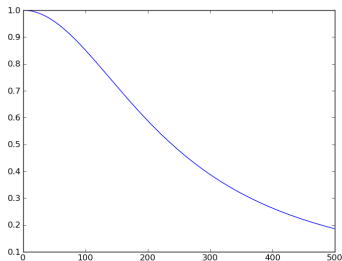
Problème : le signal est atténué, transformé et pas nécessairement récupérable.

# Le pouvoir de la sinusoïde

---

Un **signal sinusoïdal** est transformé en un autre signal sinusoïdal de même fréquence (perturbation de la phase et de l'amplitude).

Diagrammes de réponses en amplitude ( $A(f)$ ) et en phase ( $\Phi(f)$ ) en fonction de la fréquence.

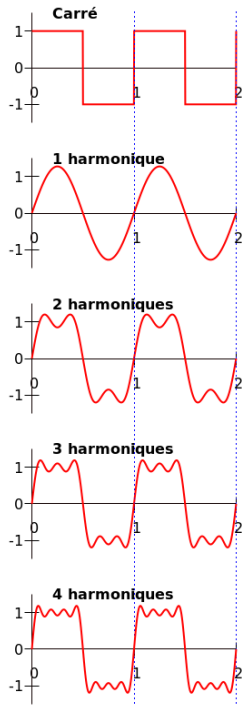
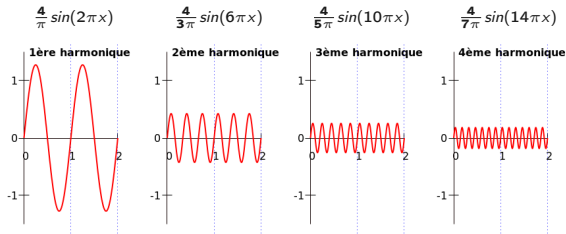




# Fourier

Tout signal se décompose en somme de sinusoides.

Chaque harmonique augmente la précision



# Limite en débit

---

Le débit est limité par le médium, sa taille, son environnement :

- Au delà d'une certaine fréquence appelée **fréquence de coupure**, la réponse en amplitude est nulle.
- On appelle **bande passante** la largeur de l'espace des fréquences utilisées (pas trop perturbées).  
**Théorème de Nyquist** : le débit maximal (en bits/s) vaut 2 fois la bande passante
- Plus le bruit est important, plus le débit permis est faible : **théorème de Shannon**

...et la capacité de décodage du récepteur.

# Utilisation

---

On s'autorise à utiliser toutes les fréquences incluses dans la bande passante :

- Plus le nombre d'harmoniques dans la bande passante est grand, plus le signal transmis ressemble à un signal numérique.
- Le récepteur reconnaît le signal d'origine s'il n'est pas trop perturbé (limite décrite par les théorèmes de Nyquist et Shannon).

# En pratique

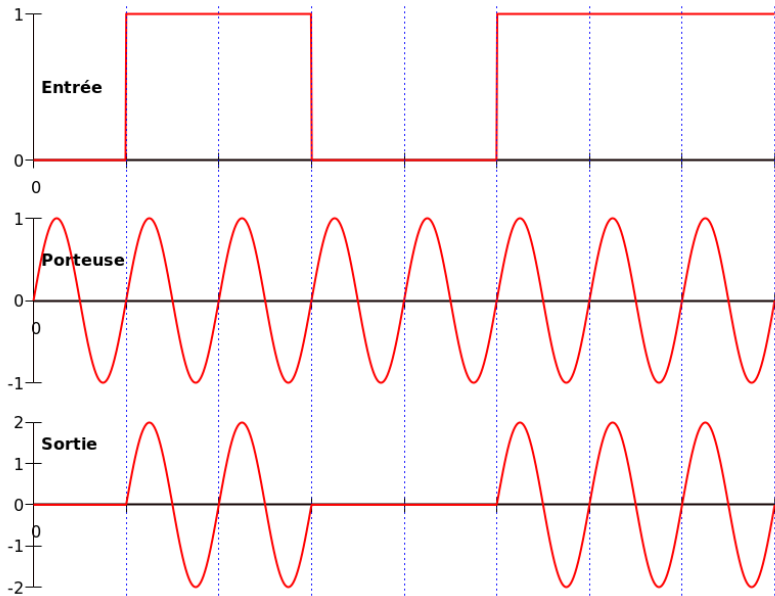
---

S'il y a trop de bruit (e.g. boucle locale : bande passante = 3kHz), on change de stratégie. On utilise une fréquence adaptée au médium, le signal sinusoïdal correspondant sera appelé **porteuse**. Des transformations de la porteuse permettent d'encoder des valeurs différentes :

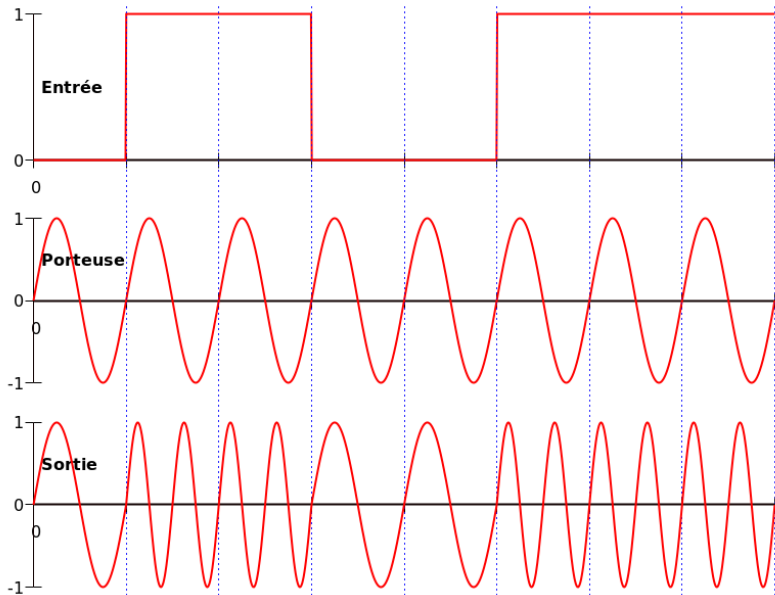
- Modulation d'amplitude
- Modulation de fréquence
- Modulation de phase

Dans chaque cas, le récepteur est capable de détecter les changements de valeur.

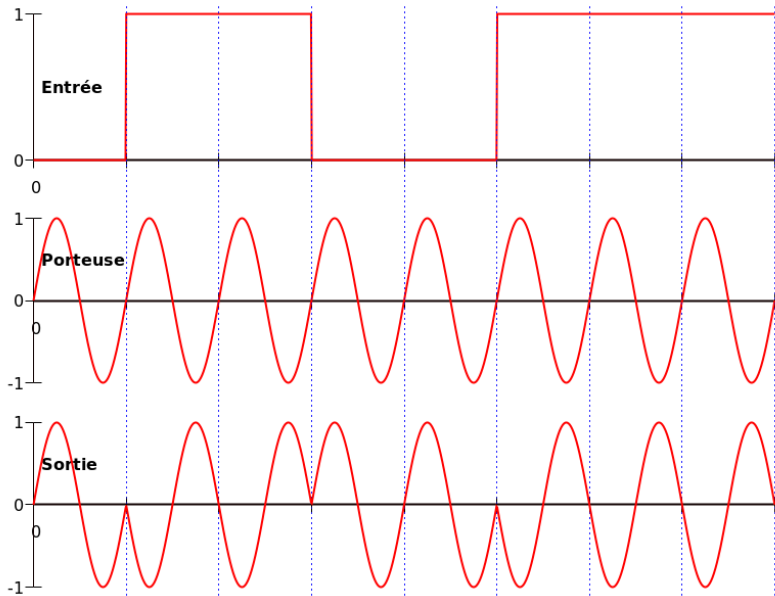
# Modulation d'amplitude



# Modulation de fréquence



# Modulation de phase



# Différents médias

---

Chaque médium a un débit et une latence :

- Séquence de bits écrite sur papier et envoyée par la poste
- Clé usb sur pigeon voyageur  
<https://www.ietf.org/rfc/rfc1149.txt>

Débit très élevé, pas de bruit, mais grande latence et coût élevé.



# Paire torsadée, câble ethernet

---

- Deux fils de cuivre enroulés l'un sur l'autre (paire torsadée)
- Porteuses tous les 4,3KHz de 30KHz à 110KHz, modulation en amplitude et en phase
- Maximum 100m sans répéteur
- Débit jusqu'à 100 Mb/s avec astuces de codage (4B/5B, MLT-3)

# Fibre optique

---

- Très haut débit théorique, record actuel à 10 Pb/s ( $10^7$  Gb/s)
- Fréquences de 200THz à 350THz (fréquences visibles de 430THz à 750THz), modulation d'intensité
- Goulot d'étranglement : Conversion électrique-optique
- Très longue distance sans répéteur
- Plus cher

# Ondes électromagnétiques, wifi 802.11n

---

- Jusqu'à 1 Gb/s avec 4 antennes ([MIMO](#))
- 2 gammes de fréquences, 2.4 et 5 GHz réparties en bandes passantes de 20 à 25MHz
- Jusqu'à 250m de portée

# Couche liaison

```
0000 FF FF FF FF FF FF 18 A6 F7 49 47 41 08 06 00 01 .....IGA....
0010 08 00 06 04 00 01 18 A6 F7 49 47 41 C0 A8 01 20 .....IGA...
0020 00 00 00 00 00 00 C0 A8 01 0C
```

```
0000 A2 05 1E 29 BB 04 18 A6 F7 49 47 41 08 00 45 00 ...). ....IGA..E.
0010 00 3A 00 01 00 00 40 06 5A C3 88 10 16 0C D0 61 ..:....@.Z.....a
0020 B1 7C 00 14 00 50 00 00 00 00 00 00 00 00 50 02 .|...P.....P.
0030 20 00 90 D2 00 00 47 45 54 20 2F 20 48 54 54 50 .....GET / HTTP
0040 2F 31 2E 30 0D 0A 0D 0A /1.0....
```

# LAN vs WAN

---

Divers protocoles selon les types de réseaux :

- Dans les WAN, protocole PPP ([Point to Point Protocol](#)), connecté, avec authentification et compression.
- Dans les LAN, protocole [ethernet](#) pour paires torsadées ou fibres optiques.
- Dans les réseaux wifi (LAN), ensemble de normes [802.11](#) (groupe de travail 11 du comité de standardisation 802 de l'IEEE).

Le focus de ce cours sera porté sur les protocoles ethernet et wifi.

# Rôle

---

- Échanger des trames.
- Éviter les interférences.
- Vérifier que les trames reçues sont celles envoyées.

Les protocoles de la couche liaison (en particulier la stratégie de contrôle d'accès au médium) dépendent du médium utilisé.

# Trames

---

- **Trame** : message unitaire de la couche liaison
- Des **données** et des **commandes**
- Codage pour délimiter les trames.

# Couche liaison :

## Découper en trames

```
0000  FF FF FF FF FF FF 18 A6  F7 49 47 41 08 06 00 01  .....IGA....
0010  08 00 06 04 00 01 18 A6  F7 49 47 41 C0 A8 01 20  .....IGA...
0020  00 00 00 00 00 00 C0 A8  01 0C
```

```
0000  A2 05 1E 29 BB 04 18 A6  F7 49 47 41 08 00 45 00  ...)...IGA..E.
0010  00 3A 00 01 00 00 40 06  5A C3 88 10 16 0C D0 61  .:...@.Z.....a
0020  B1 7C 00 14 00 50 00 00  00 00 00 00 00 00 50 02  .|...P.....P.
0030  20 00 90 D2 00 00 47 45  54 20 2F 20 48 54 54 50  .....GET / HTTP
0040  2F 31 2E 30 0D 0A 0D 0A
```



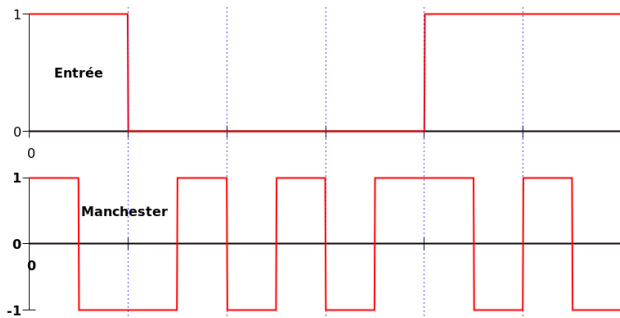
# Codage de Manchester

---

Sur un signal créneau :

- Difficulté de distinguer le bit 0 et la tension de repos
- Difficulté de distinguer 111 et 1111 (synchronisation)

Le **codage de Manchester** procure une horloge commune :



# Codage de début et de fin

---

La solution est de mettre un code de début et de fin :

- Un **drapeau** qui sert de délimiteur :  
01111110 dans une suite binaire.
- Lorsqu'on rencontre une suite de cinq 1 quelque part dans le texte, on la remplace par 111110 pour éviter la confusion.
- Cela permet de synchroniser et de découper aussi en octets.

# Couche liaison :

## Détecter les erreurs

```
0000 FF FF FF FF FF FF 18 A6 F7 49 47 41 08 06 00 01 .....IGA....
0010 08 00 06 04 00 01 18 A6 F7 49 47 41 C0 A8 01 20 .....IGA...
0020 00 00 00 00 00 00 C0 A8 01 0C
```

```
0000 A2 05 1E 29 BB 04 18 A6 F7 49 47 41 08 00 45 00 ...). ....IGA..E.
0010 00 3A 00 01 00 00 40 06 5A C3 88 10 16 0C D0 61 ..:....@.Z.....a
0020 B1 7C 00 14 00 50 00 00 00 00 00 00 00 50 02 .|...P.....P.
0030 20 00 90 D2 00 00 47 45 54 20 2F 20 48 54 54 50 .....GET / HTTP
0040 2F 31 2E 30 0D 0A 0D 0A /1.0....
```

# Coder l'information

---

- Les sources d'erreurs de la transmission dans le médium ne sont pas maîtrisables.
- Pour détecter des erreurs potentielles, on introduit de la redondance dans le message envoyé.
- Un code de paramètre  $(n, m)$  est une manière déterministe et réversible de transformer  $n$  bits en  $m > n$ .
- Par exemple en ajoutant  $m - n$  bits aux  $n$  premiers.
- Pour un mot  $u$  donné, on note  $c(u)$  le mot produit par le code.

# Détecter vs corriger

---

- Suite à l'envoi de  $c(u)$ , le récepteur reçoit  $w$  qu'il essaie de décoder.
- On dit que  $c(u)$  est transmis avec  $k$  erreurs lorsque  $k$  bits diffèrent entre  $c(u)$  et  $w$ .
- Un code **détecte**  $k$  erreurs lorsque, si  $c(u)$  est transmis avec moins de  $k$  erreurs, il permet de savoir que la transmission est erronée.
- Un code **corrige**  $k$  erreurs lorsque, si  $c(u)$  est transmis avec moins de  $k$  erreurs, il permet de retrouver  $c(u)$ .

# Bit de parité

---

Premier exemple :  $m = n + 1$ .

- Le dernier bit est la somme de tous les autres, i.e. la parité du nombre de bits à 1.
- 1101001000110 devient 11010010001100.
- Détection d'une erreur.
- Pas de correction.

# Bit de parité en 2D

---

On utilise la même technique sur une grille de taille connue :

$$\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 \end{array}$$

$$\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 1 & 1 & 1 & 1 \end{array}$$

Correction d'une erreur, détection de 2.

# Checksum

---

Autre méthode (rapide mais sans correction) :

- On voit un mot à envoyer comme une suite de mots de  $k$  bits pour  $k$  donné.
- On ajoute à la fin de ce mot le XOR bit à bit de tous les sous-mots de taille  $k$ .
- Ex : 110001101
- Résultat : 110001101010



# Code minimal

---

- On veut un code corrigeant 1 erreur. Il y a  $2^n$  mots d'entrée et  $m + 1$  dispositions pour l'erreur éventuelle.
- Si 2 de ces configurations coïncident, on ne pourra pas choisir, donc :

$$2^n(m + 1) \leq 2^m$$

Si  $r = m - n$ , alors  $2^r - r \geq n + 1$ .

- Le [code de Hamming](#) atteint ce minimum.

# Code de Hamming

---

On place en position  $2^i$  un bit de parité pour les positions ayant  $2^i$  dans leur décomposition binaire.

Ex :

1	1	0	1
---	---	---	---

$2^0$	$2^1$		$2^2$			
x	y	1	z	1	0	1

$$x = 1 + 1 + 1$$

$$y = 1 + 0 + 1$$

$$z = 1 + 0 + 1$$

# Cyclic Redundancy Check

---

- On représente le mot  $a$  par le polynôme  $\sum_i a_i X^i$  .
- On calcule le reste d'une division euclidienne par un polynôme  $P$  (modulo 2).
- Très facile à calculer par circuit électronique donc adapté à la couche liaison !
- Efficace en détection d'erreurs.

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{cccccccccc|c} x^8 & +x^7 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & x^5 + x^2 + 1 \\ \hline & & & & & & & & & \end{array}$$

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{cccccccccc|l} x^8 & +x^7 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & x^5 + x^2 + 1 \\ & & & & & & & & & \hline & & & & & & & & & x^3 \end{array}$$

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{rcccccccc|l} & x^8 & +x^7 & +0 & +0 & +0 & +0 & +0 & +0 & x^5 + x^2 + 1 \\ - & x^8 & & & x^5 & & x^3 & & & \hline & & & & & & & & & x^3 \end{array}$$

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

	$x^8$	$+x^7$	$+0$	$+0$	$+0$	$+0$	$+0$	$+0$	$+0$	$ $	$x^5 + x^2 + 1$
$-$	$x^8$			$x^5$		$x^3$				$ $	$x^3$
		$x^7$	$+0$	$+x^5$	$+0$	$+x^3$	$+0$	$+0$	$+0$	$ $	

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{cccccccccc|c} x^8 & +x^7 & +0 & +0 & +0 & +0 & +0 & +0 & +0 & x^5 + x^2 + 1 \\ - x^8 & & & x^5 & & x^3 & & & & \hline & x^7 & +0 & +x^5 & +0 & +x^3 & +0 & +0 & +0 & x^3 + x^2 \end{array}$$



# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

	$x^8$	$+x^7$	$+0$	$+0$	$+0$	$+0$	$+0$	$+0$	$+0$		$x^5 + x^2 + 1$
-	$x^8$			$x^5$		$x^3$					$x^3 + x^2$
		$x^7$	$+0$	$+x^5$	$+0$	$+x^3$	$+0$	$+0$	$+0$		
-		$x^7$			$x^4$		$x^2$				

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

	$x^8$	$+x^7$	$+0$	$+0$	$+0$	$+0$	$+0$	$+0$	$+0$	$x^5 + x^2 + 1$
-	$x^8$			$x^5$		$x^3$				$x^3 + x^2$
		$x^7$	$+0$	$+x^5$	$+0$	$+x^3$	$+0$	$+0$	$+0$	
-		$x^7$			$x^4$		$x^2$			
				$x^5$	$+x^4$	$+x^3$	$+x^2$	$+0$	$+0$	

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{rcccccccc|l} & x^8 & +x^7 & +0 & +0 & +0 & +0 & +0 & +0 & x^5 + x^2 + 1 \\ - & x^8 & & & x^5 & & x^3 & & & \hline & & x^7 & +0 & +x^5 & +0 & +x^3 & +0 & +0 & x^3 + x^2 + 1 \\ - & & x^7 & & & x^4 & & x^2 & & \\ & & & & x^5 & +x^4 & +x^3 & +x^2 & +0 & +0 \\ - & & & & x^5 & & & x^2 & & 1 \end{array}$$



# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{rcccccccc|c} & x^8 & +x^7 & +0 & +0 & +0 & +0 & +0 & +0 & x^5 + x^2 + 1 \\ - & x^8 & & & x^5 & & x^3 & & & \hline & & x^7 & +0 & +x^5 & +0 & +x^3 & +0 & +0 & x^3 + x^2 + 1 \\ - & & x^7 & & & x^4 & & x^2 & & \\ & & & & x^5 & +x^4 & +x^3 & +x^2 & +0 & +0 \\ - & & & & x^5 & & & x^2 & & 1 \\ & & & & & x^4 & +x^3 & +0 & +0 & +1 \end{array}$$

Résultat :  $x^4 + x^3 + 1$ , i.e. 11001.

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2 + 1$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{rcccccccc|c} - & x^8 & +x^7 & +0 & +0 & +0 & +0 & +0 & +0 & x^5 + x^2 + 1 \\ & x^8 & & & x^5 & & x^3 & & & \hline - & & x^7 & +0 & +x^5 & +0 & +x^3 & +0 & +0 & \\ & & x^7 & & & x^4 & & x^2 & & \\ - & & & & x^5 & +x^4 & +x^3 & +x^2 & +0 & +0 \\ & & & & x^5 & & & x^2 & & 1 \\ - & & & & & x^4 & +x^3 & +0 & +0 & +1 \end{array}$$

Résultat :  $x^4 + x^3 + 1$ , i.e. 11001. On envoie donc **110011001**.

# CRC : exemple

---

Calcul du CRC de 1100 pour  $x^5 + x^2 + 1$  (CRC-5-USB). On multiplie  $x^3 + x^2 + 1$  par  $x^5$  (plus grand monôme de  $x^5 + x^2 + 1$ ).

$$\begin{array}{r} x^8 + x^7 + 0 + 0 + 0 + 0 + 0 + 0 + 0 \\ - x^8 \\ \hline x^7 + 0 + x^5 + 0 + x^3 + 0 + 0 + 0 \\ - x^7 \\ \hline x^5 + x^4 + x^3 + x^2 + 0 + 0 \\ - x^5 \\ \hline x^4 + x^3 + 0 + 0 + 1 \end{array} \left| \begin{array}{l} x^5 + x^2 + 1 \\ \hline x^3 + x^2 + 1 \end{array} \right.$$

Résultat :  $x^4 + x^3 + 1$ , i.e. 11001. On envoie donc **110011001**.

En réception, le calcul en partant de 110011001 doit terminer sur 00000. Dans le cas contraire, on a détecté une erreur de transmission.

# Couche liaison :

## Accès au médium

```
0000 FF FF FF FF FF FF 18 A6 F7 49 47 41 08 06 00 01 .....IGA....
0010 08 00 06 04 00 01 18 A6 F7 49 47 41 C0 A8 01 20 .....IGA...
0020 00 00 00 00 00 00 C0 A8 01 0C
```

```
0000 A2 05 1E 29 BB 04 18 A6 F7 49 47 41 08 00 45 00 ...). ....IGA..E.
0010 00 3A 00 01 00 00 40 06 5A C3 88 10 16 0C D0 61 ..:....@.Z.....a
0020 B1 7C 00 14 00 50 00 00 00 00 00 00 00 00 50 02 .|...P.....P.
0030 20 00 90 D2 00 00 47 45 54 20 2F 20 48 54 54 50 .....GET / HTTP
0040 2F 31 2E 30 0D 0A 0D 0A /1.0....
```



# Accès au médium

---

La couche liaison contient 2 sous-couches ayant des rôles complémentaires :

- Medium Access Control : dépendante de la couche physique. Selon la topologie et le médium, gestion des temps de parole.
- Logical Link Control : ajout de contrôles, notamment contrôle des erreurs ou contrôle de flux. Moins présente pour ethernet.

# Adressage

---

Chaque interface réseau possède une adresse **MAC** ou **physique** ou **LAN** codée en dur par le fabricant et attribuée par l'Institute of Electrical and Electronics Engineers (IEEE) :

- Adresse unique (à peu près...)
- Sur 48 bits (ou 6 octets, ou 6 couples de caractères hexadécimaux), ex : *B8 : 31 : 56 : BA : C2 : 01*.
- Deux bits pour distinguer adresses locales/globales(gérées par l'IEEE) et pour le multicast, d'où  $2^{46}$  adresses.
- Adresse de broadcast : *ff : ff : ff : ff : ff : ff*.

# Adressage

---

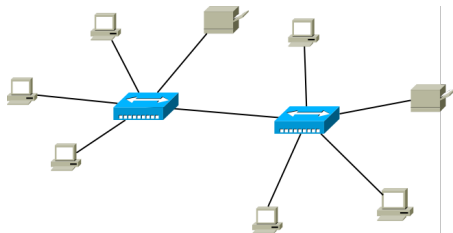
```
# ip link
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN \\  
    mode DEFAULT group default qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast \\  
    state UP mode DEFAULT group default qlen 1000  
    link/ether 4c:cc:6a:b0:c3:4b brd ff:ff:ff:ff:ff:ff
```

# Topologie locale : LAN

---

Le plus souvent pour des réseaux filaires : des machines reliées par des **hub**.

- Un hub est un composant de la couche physique.
- Lorsqu'il reçoit une trame sur un port, il la transmet sur tous les autres ports.
- S'il reçoit 2 trames en même temps : collision.



# Accès au médium : le problème

---

- Plusieurs machines veulent parler en même temps sur un même médium.
- Lorsqu'une trame est émise sur le LAN, toutes les machines connectées la reçoivent.
- Si deux trames circulent en même temps, il y a **collision**, elles sont perdues toutes les 2.
- Chaque machine peut savoir si une trame est en train de circuler.
- Chaque machine peut détecter une collision éventuelle.

# Des solutions

---

Une solution pour une topologie en anneau (Token Ring) :

- Un unique jeton circule autorisant son détenteur à parler.
- Il envoie le message à son voisin avec le jeton.
- Lorsqu'une machine reçoit le message qu'elle a envoyé, elle le supprime.

# Des solutions

---

Autre solution : un système centralisé.

- Un arbitre répartit les temps de parole
- En écoutant la porteuse, l'arbitre sait quand il peut autoriser une nouvelle machine à parler.

# Des solutions en couche physique

---

Diviser le médium en plusieurs canaux, par exemple un par machine :

- des fréquences différentes (Wifi).
- des temps de parole individuels réguliers (synchronisation des machines sur une même horloge).
- Code de Walsh (e.g. : CDMA pour le wifi).



# ALOHA

---

- Chaque machine émet quand elle veut.
- En cas de collision, la trame est réémise après un temps aléatoire.
- Si une seule machine parle, efficacité 100% en utilisation du canal, si plusieurs machines : 18%.
- Version discrète : les émissions se font au début d'un slot (les machines sont synchronisées).
- Dans ce cas : si une seule machine parle, efficacité 100%, si plusieurs machines : 37%.

# Écoute de la porteuse

---

Dans certaines solutions, ou pour compléter ALOHA, on écoute la porteuse avant d'émettre.

Problème : la diffusion des trames n'est pas instantannée.

A

B

C

D

# Écoute de la porteuse

---

Dans certaines solutions, ou pour compléter ALOHA, on écoute la porteuse avant d'émettre.

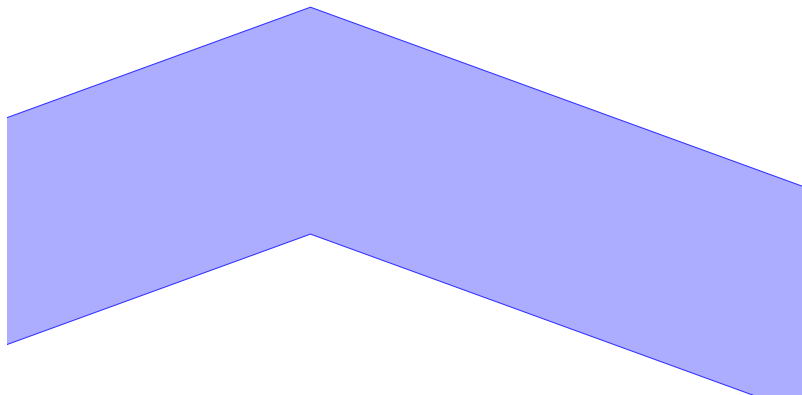
Problème : la diffusion des trames n'est pas instantannée.

A

B

C

D



# Écoute de la porteuse

---

Dans certaines solutions, ou pour compléter ALOHA, on écoute la porteuse avant d'émettre.

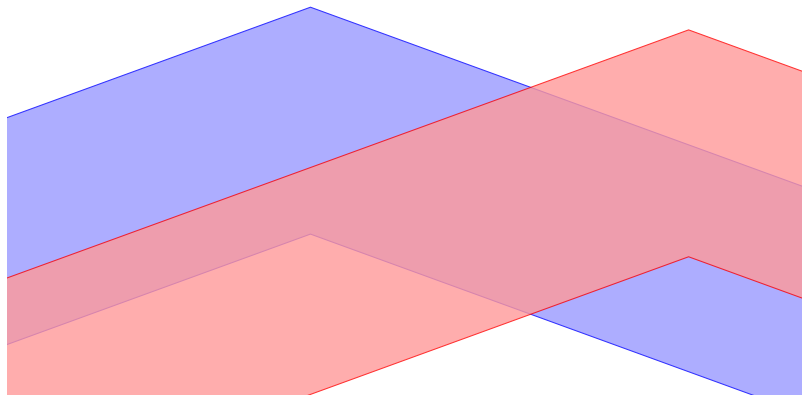
Problème : la diffusion des trames n'est pas instantannée.

A

B

C

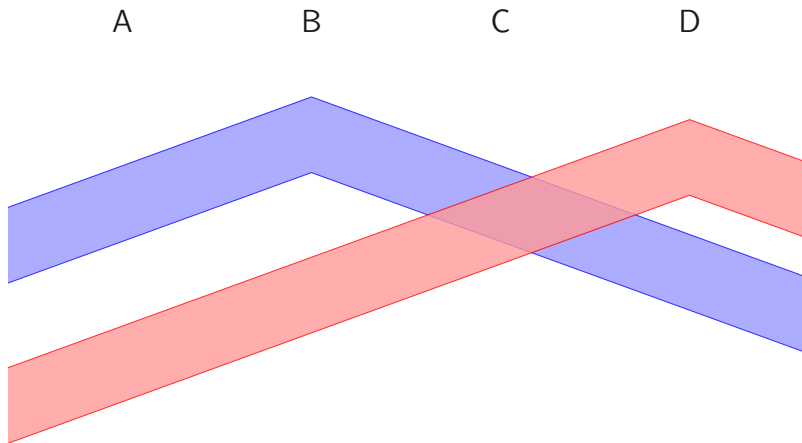
D



# Écoute de la porteuse

---

Short-frame collision problem



# Pour ethernet : CSMA/CD

---

Carrier-Sense Multiple Access / Collision Detection

ALOHA augmenté d'une détection des collisions.

- Écoute de la porteuse.
- Si le canal est libre, on émet.
- En cas de collision, on s'arrête et on envoie un signal de collision pour avertir toutes les machines.
- Attente d'un temps aléatoire avant de réémettre.

Attente exponentielle : Lors du  $n$ -ème réenvoi de la trame, on attend un temps compris entre 0 et  $2^n$  (limité à 1024).

# WIFI : Éviter les collisions

```
0000 FF FF FF FF FF FF 18 A6 F7 49 47 41 08 06 00 01 .....IGA....
0010 08 00 06 04 00 01 18 A6 F7 49 47 41 C0 A8 01 20 .....IGA...
0020 00 00 00 00 00 00 C0 A8 01 0C
```

```
0000 A2 05 1E 29 BB 04 18 A6 F7 49 47 41 08 00 45 00 ...). ....IGA..E.
0010 00 3A 00 01 00 00 40 06 5A C3 88 10 16 0C D0 61 ..:....@.Z.....a
0020 B1 7C 00 14 00 50 00 00 00 00 00 00 00 50 02 .|...P.....P.
0030 20 00 90 D2 00 00 47 45 54 20 2F 20 48 54 54 50 .....GET / HTTP
0040 2F 31 2E 30 0D 0A 0D 0A /1.0....
```

# Spécificités

---

- Difficulté d'écouter la porteuse et émettre en même temps : pas de détection des collisions.
- Un signal émis par une machine  $A$  peut être reçu par  $B$  alors que  $C$  est hors de portée :
  - ▶ Zones de portée de l'émetteur wifi.
  - ▶ Obstacle entre  $A$  et  $C$ .

Plutôt que de se contenter de détecter les collisions (ethernet), on utilise un protocole pour éviter les collisions.

Les erreurs de transmission étant beaucoup plus fréquentes que sur un LAN ethernet, on ajoute aussi un mécanisme d'accusés de réception.



# Basic Service Set

---

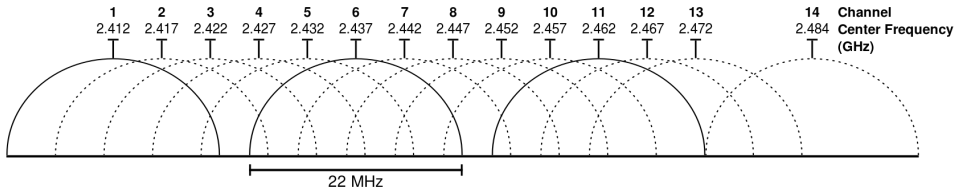
Les réseaux locaux wifi sont structurés autour de **points d'accès** qui sont les stations wifi, connectées à internet.

- Des machines groupées autour de leur point d'accès constituent un **BSS**.
- Les points d'accès sont reliés entre eux (ethernet) pour former un **Extended Service Set**.

# Division en canaux

Canaux 2.4 GHz (802.11b/g/n)

2 bandes de fréquence (autour de 2.4 et 5 GHz), découpées en canaux de bande passante 22 MHz.



- Pas d'interférences si les canaux ne se recoupent pas.
- Au maximum 3 canaux orthogonaux (1, 6 et 11), 4 avec 14 (Japon).

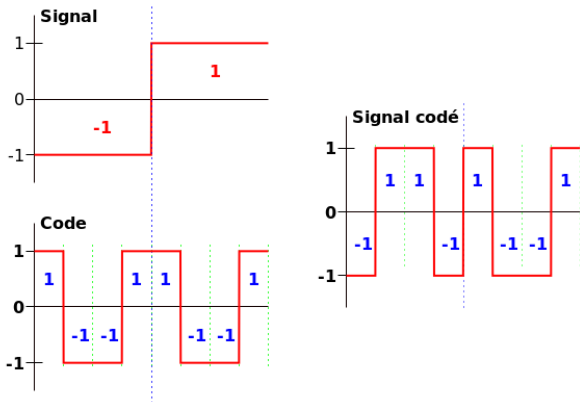
# CDMA

---

- Multiplexage utilisé par exemple en téléphonie.
- Les utilisateurs partagent une même fréquence.
- Le bit 1 est représenté par 1, et 0 par  $-1$ .
- Chaque utilisateur possède un code (séquence de 1 et de  $-1$  de longueur fixée).
- Pour coder des données, on multiplie chaque bit par le code.
- Pour décoder, on multiplie le signal codé par le code.
- Tous les codes doivent être orthogonaux !

# CDMA

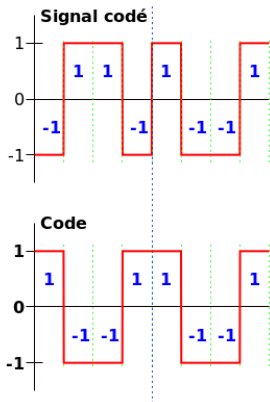
Codage de  $b_0b_1 = 01$



- Un bit à 1 est remplacé par le code.
- Un bit à 0 est remplacé par l'opposé du code.

# CDMA

## Décodage

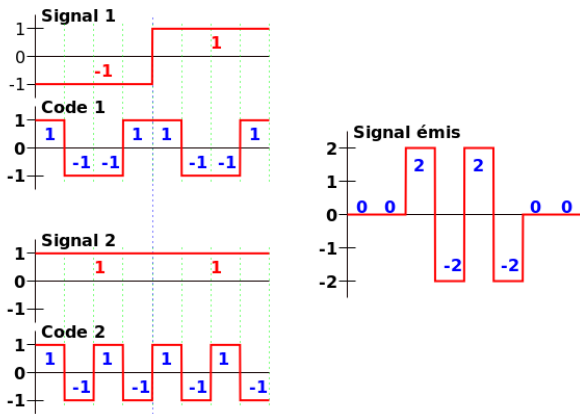


$$b_0 = \frac{1 * (-1) + (-1) * 1 + (-1) * 1 + 1 * (-1)}{4}$$

$$b_1 = \frac{1 * 1 + (-1) * (-1) + (-1) * (-1) + 1 * 1}{4}$$

# CDMA

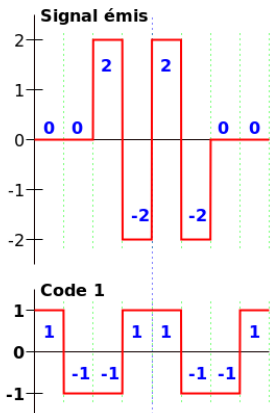
Deux émetteurs



- Les deux signaux émis s'ajoutent.
- On prend deux codes orthogonaux : le produit somme à 0.  
 $c1 \otimes c2 = 1 * 1 + (-1) * (-1) + (-1) * 1 + 1 * (-1) = 0$

# CDMA

Deux émetteurs



$$b_0 = \frac{0 * 1 + 0 * (-1) + 2 * (-1) + (-2) * 1}{4}$$

$$b_1 = \frac{2 * 1 + (-2) * (-1) + 0 * (-1) + 0 * 1}{4}$$

# CSMA/CA

---

## Carrier-Sense Multiple Access/Collision Avoidance

- Évitement de collisions, utilisé pour le wifi.
- Écoute de la porteuse.
- **Distributed coordination function** : Horloge partagée.
- Trames d'acquittement.
- Trames de réservation et libération du support **RTS** et **CTS**.



# CSMA/CA

---

CSMA/CA : écoute de la porteuse

- Détection de l'utilisation du médium : s'il est utilisé on attend.
- Utilisation d'un mécanisme d'écoute virtuelle (NAV : Network Allocation Vector)
  - ▶ Temporisateur qui prend en compte les réservations du support par les autres machines.
  - ▶ Mise à jour en fonction des trames reçues.

# CSMA/CA

---

## CSMA/CA : DCF

- Pour émettre, le support doit être libre et  $NAV = 0$ .
- On attend un **DIFS** (Distributed Inter-Frame Space) puis un nombre aléatoire de *timeslots*.
- Si le support est toujours libre, on émet.

# CSMA/CA

---

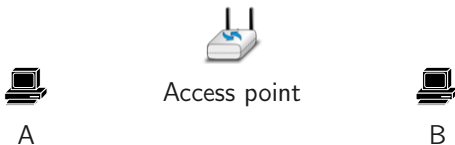
## CSMA/CA : Acquittements

- Si une station reçoit une trame sans erreurs, elle envoie une trame d'acquittement.
- Pour faciliter l'accès au support dans ce cas, le temps d'attente de la DCF est **SIFS** (Short Inter-Frame Space) où  $SIFS = DIFS - 2 \text{ timeslots}$ .

# CSMA/CA

---

## CSMA/CA : Réserveation du support

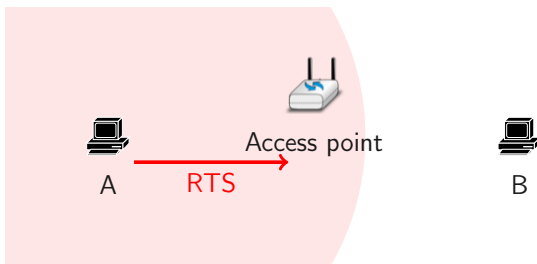


- **RTS** : Request To Send et **CTS** : Clear To Send, pour réserver le support et autoriser la communication.
- Pour répondre au problème du noeud caché et éviter des interférences au niveau du destinataire.

# CSMA/CA

---

## CSMA/CA : Réserveation du support

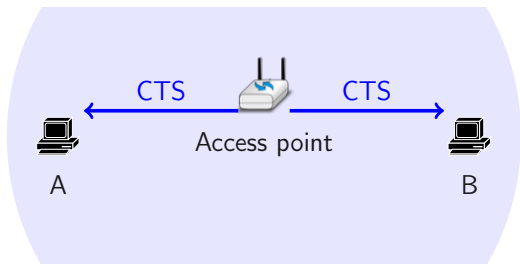


- **RTS** : Request To Send et **CTS** : Clear To Send, pour réserver le support et autoriser la communication.
- Pour répondre au problème du noeud caché et éviter des interférences au niveau du destinataire.

# CSMA/CA

---

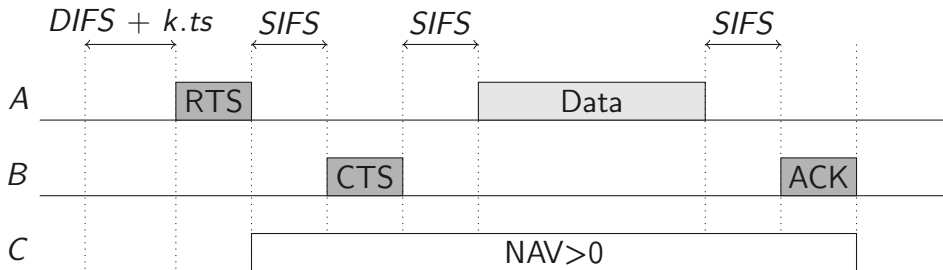
## CSMA/CA : Réserveation du support



- **RTS** : Request To Send et **CTS** : Clear To Send, pour réserver le support et autoriser la communication.
- Pour répondre au problème du noeud caché et éviter des interférences au niveau du destinataire.

# CSMA/CA

CSMA/CA : Échange typique



- Le RTS de *A* réserve le médium jusqu'à la fin de l'envoi des Data.
- Le CTS de *B* réserve le canal jusqu'à la fin de l'envoi du ACK.
- Le NAV de *C* est mis à jour à chaque fois.

# Couche liaison :

## Trames

```
0000 FF FF FF FF FF FF 18 A6 F7 49 47 41 08 06 00 01 .....IGA....
0010 08 00 06 04 00 01 18 A6 F7 49 47 41 C0 A8 01 20 .....IGA...
0020 00 00 00 00 00 00 C0 A8 01 0C
```

```
0000 A2 05 1E 29 BB 04 18 A6 F7 49 47 41 08 00 45 00 ...). ....IGA..E.
0010 00 3A 00 01 00 00 40 06 5A C3 88 10 16 0C D0 61 ..:....@.Z.....a
0020 B1 7C 00 14 00 50 00 00 00 00 00 00 00 50 02 .|...P.....P.
0030 20 00 90 D2 00 00 47 45 54 20 2F 20 48 54 54 50 .....GET / HTTP
0040 2F 31 2E 30 0D 0A 0D 0A /1.0....
```



# Ethernet IEEE 802.3

---

- Paire torsadée ou fibre optique notamment.
- Ensemble de standards définis par l'IEEE.
- Protocole d'accès au médium : CSMA/CD.
- Détection des trames et éventuelles collisions en mesurant la tension.
- **MTU**(Maximum Transmission Unit) de 1500 octets (données uniquement).

# Trames Ethernet

---

Flag	Destination	Source	Type	Données	CRC
8	6	6	2	46-1500	4

- Minimum 46 octets (caractère ASCII) de données.
- Taille max 1518 en oubliant les flags (synchronisation).
- CRC 0x04C11DB7

# Hub vs switch

---

- Un **hub** transmet chaque message reçu sur l'ensemble de ses ports.
- Si le hub reçoit deux messages en même temps : collision et envoi d'un signal de collision.
  
- Un **switch** ne transmet un message reçu qu'au destinataire.
- Il maintient une table de correspondance ports/adresses MAC.
- Il utilise CSMA/CD sur chaque port.
- Il ne possède pas d'adresse MAC !

# Table d'un switch

---

- Table vide au début.
- Mise à jour lors de la réception depuis une adresse inconnue.
- Envoi à tout le monde si le destinataire est inconnu.
- Filtrage des paquets ayant même émetteur et destinataire.

```
brctl showmacs br
```

```
port no mac addr                ageing timer
1      00:c0:ff:ee:00:00            16.64
1      0a:ba:d0:ba:be:00            0.51
2      ca:fe:d0:0d:00:00            0.42
3      00:00:de:ad:be:ef            13.37
```

# WIFI IEEE 802.11

---

- Normes 802.11a,b,g,n.
- Protocole d'accès au médium : CSMA/CA.
- **MTU**(Maximum Transmission Unit) de 2312 octets (données uniquement).

# Trames Wifi

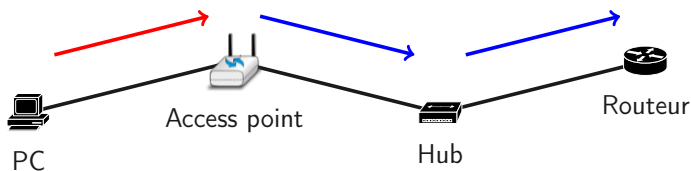
---

FC	Durée	@1	@2	@3	SC	@4	Données	CRC
2	2	6	6	6	2	6	0-2312	2

- **Durée** sert à transmettre le temps de réservation du support.
- **FC**(Frame Control) contient des informations sur le type de trame.
- **SC**(Sequence Control) pour ordonner les fragments si nécessaire.

# Trames Wifi

Pourquoi plus de 2 adresses MAC ?



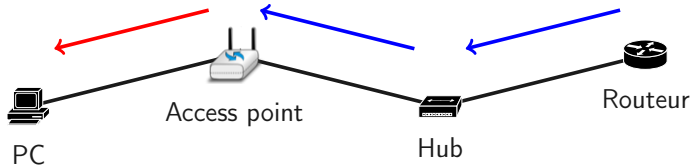
**Wifi**  
@1 : dst = Access Point  
@2 : src = PC  
@3 : Routeur

**Ethernet**  
@1 : dst = Routeur  
@2 : src = PC

Note : La 4ème adresse sert en mode ad-hoc (pas d'AP).

# Trames Wifi

Pourquoi plus de 2 adresses MAC ?



Ethernet	Wifi
@1 : dst = PC	@1 : dst = PC
@2 : src = Routeur	@2 : src = Access Point
	@3 : Routeur

Note : Si c'est un switch, plusieurs @ MAC sur le même port.



# Connexion Wifi

---

La connexion à un réseau wifi se fait en 3 étapes :

- **Sondage** : envoi d'une trame pour détecter les points d'accès à proximité  
Réponse contenant les caractéristiques du réseau
- **Authentification** : contrôle d'identité
- **Association** : le point d'accès associe un port logique à la nouvelle station.

# Wifi et sécurité

---

Dans le cas du wifi, toute station à portée peut écouter les trames qui circulent :

- Nécessité d'une authentification sécurisée (secret partagé, échange de clés préalable, . . .)
- Nécessité de chiffrer les échanges (WEP, WPA).

# Wifi et configuration

---

```
lapin@terrier:~# iwconfig
```

```
wlp3s0    IEEE 802.11  ESSID:"boisjoli"  
Mode:Managed  Frequency:2.437 GHz  Access Point: 45:AB:7D:7D:4B:09  
Bit Rate=57.8 Mb/s   Tx-Power=15 dBm  
Retry short limit:7   RTS thr:off   Fragment thr:off  
Encryption key:off  
Power Management:off  
Link Quality=44/70   Signal level=-66 dBm  
Rx invalid nwid:0   Rx invalid crypt:0   Rx invalid frag:0  
Tx excessive retries:332   Invalid misc:648   Missed beacon:0
```

# Vers l'internet (et au-delà ?) :

## Des LAN à l'internet

```
0000  FF FF FF FF FF FF 18 A6  F7 49 47 41 08 06 00 01  .....IGA....
0010  08 00 06 04 00 01 18 A6  F7 49 47 41 C0 A8 01 20  .....IGA...
0020  00 00 00 00 00 00 C0 A8  01 0C
```

```
0000  A2 05 1E 29 BB 04 18 A6  F7 49 47 41 08 00 45 00  ...). ....IGA..E.
0010  00 3A 00 01 00 00 40 06  5A C3 88 10 16 0C D0 61  .:....@.Z.....a
0020  B1 7C 00 14 00 50 00 00  00 00 00 00 00 00 50 02  .|...P.....P.
0030  20 00 90 D2 00 00 47 45  54 20 2F 20 48 54 54 50  .....GET / HTTP
0040  2F 31 2E 30 0D 0A 0D 0A
```

# Comment connecter les réseaux ?

---



110



109



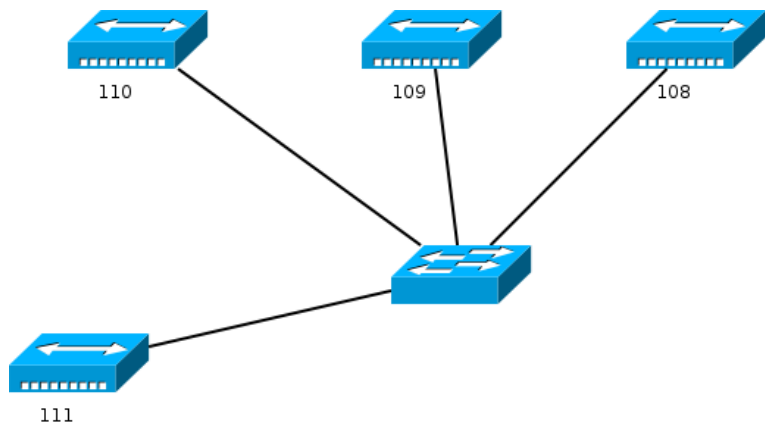
108



111

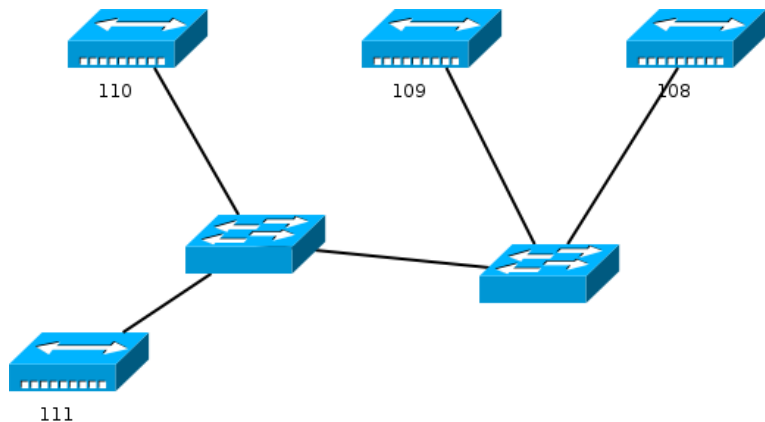
# Comment connecter les réseaux ?

---



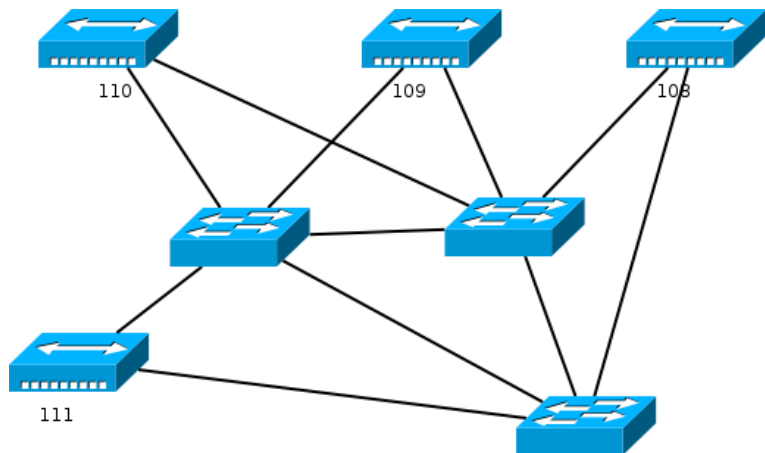
# Comment connecter les réseaux ?

---



# Comment connecter les réseaux ?

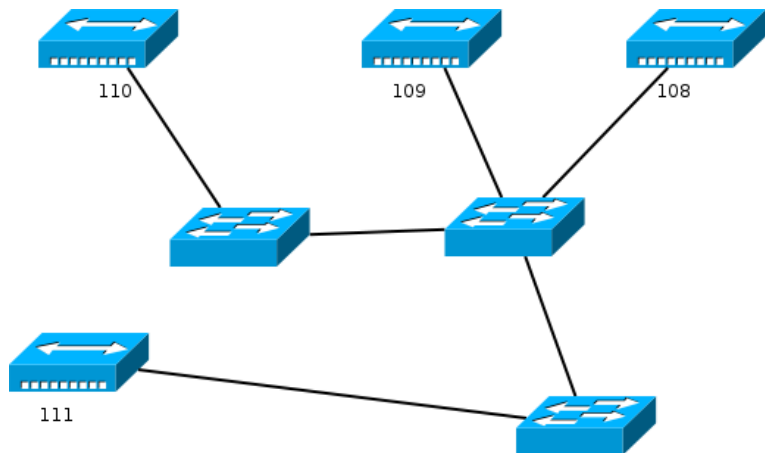
---





# Comment connecter les réseaux ?

---



# Comment connecter les réseaux ?

---

Différentes problématiques :

- Fiabilité : plusieurs chemins entre 2 machines.
- Efficacité : minimiser les temps de transfert.
- Réalisabilité : Ne pas installer des liens directs entre chaque couple de machines.