


Re seau

Semestre 1

Topologie réseau

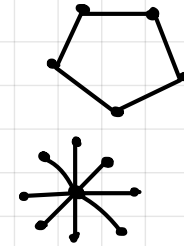
• Point à point 

• Bus 

• Full mesh 

• Ring

• Star



• Etoile étendue

[Exigence 1 : Connectivité (p.19)

Point à Point : $\# \text{ nœuds} = \frac{m(m-1)}{2}$ (gros problème de coût)

Bus : $\# \text{ nœuds} = 1 + m$

Lexique des composants réseau

1. Application : utilise le réseau (skype ...)
2. Hôte : support apps (laptop)
3. Routeur : relaie les messages entre des liens (access point)
4. Lien, canal : Connecte les nœuds

Valeur d'un réseau

La valeur d'un réseau est proportionnel à son nombre de nœud au carré

Exigence 2 : Partage des ressources efficace

Multiplexage (temporel, synchrone) : chaque interval de temps \rightarrow nouveau slot alloué même si il n'est pas utilisé.

Capacité d'accès limité au nombre de slot.

Multiplexage statistique : Les données sont stockées dans l'équipement et en fonction de la capacité de calcul de l'équipement, envoie les données.
Capacité d'accès limité par la puissance de l'équipement.

Exigence 3 : Abstraction commune aux applications

Chaque app doit avoir une couche d'abstraction commune pour communiquer.
(les sockets)

Modèles de communication courants

1. Client - Server (topologie en étoile)

- Quasi-toutes les apps sont comme ça.

Si le serv tombe \Rightarrow l'app tombe.

Confidentialité pas top

+ Facile à gérer

2. Pair à Pair

Chaque machine est client et serveur.

- Faut découvrir les autres du réseau

+ Plus résilient

Exigence 4 : (inter) opérabilité

Impacte majeur sur notre réseau, aujourd'hui

Comment fait-on y face ?

1. On ne change rien

2. on structure

3. on automatise

4. on standardise

Architecture du réseau

Plus on est dans une haute couche, plus il y a d'innovation car moins de contraintes.

Protocole

à compléter

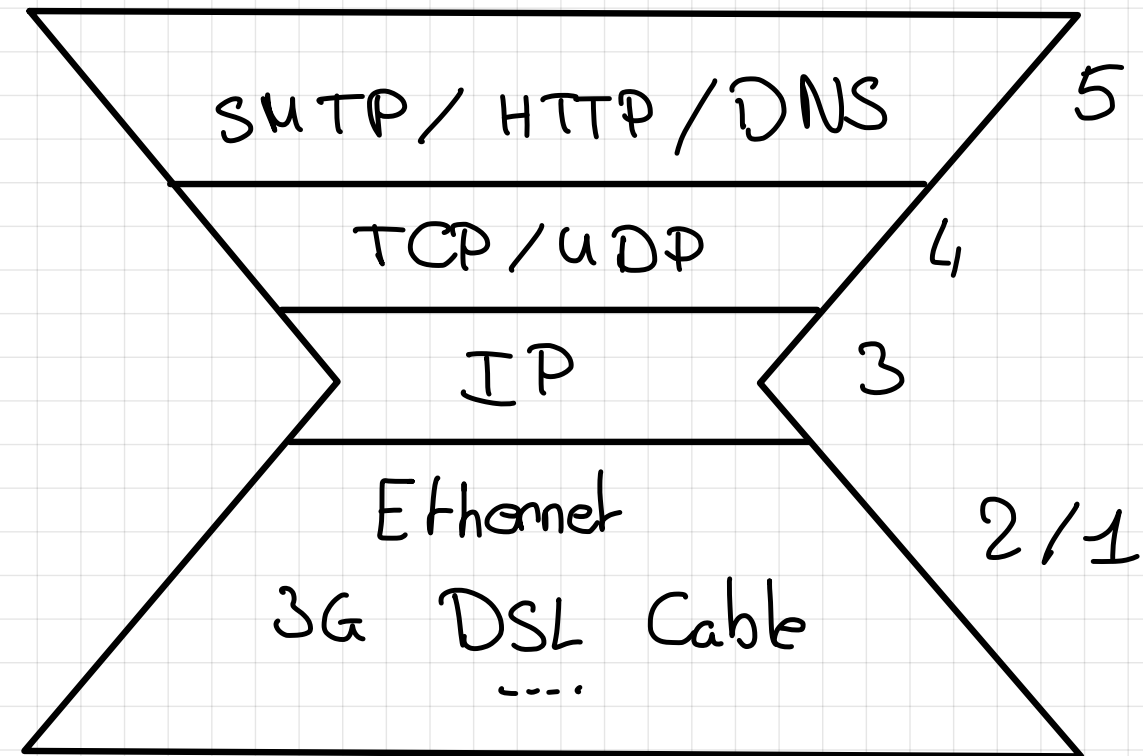
Couche de réseau

Modèle OSI (théorique)

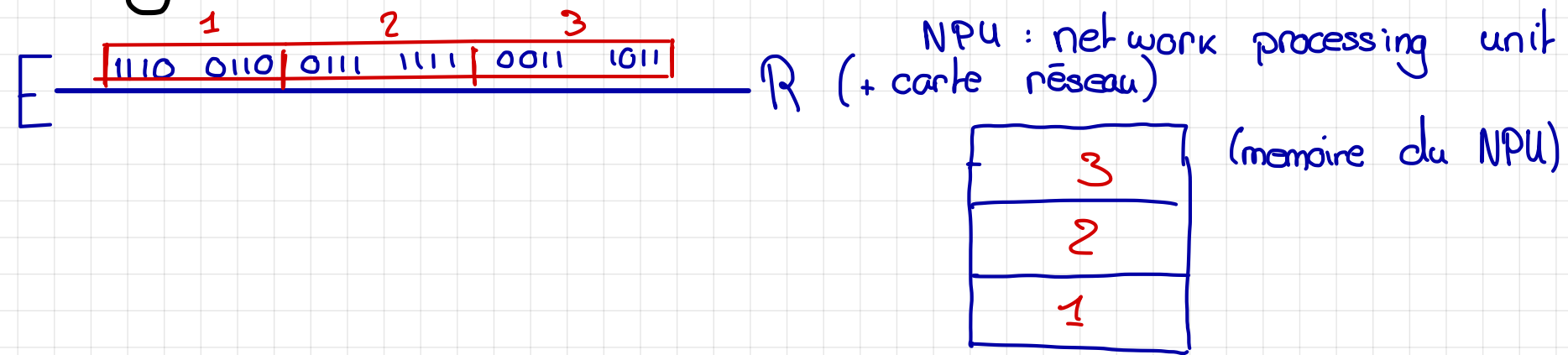
1. Couche physique (analogique to binaire)
2. Couche de liaison (collecte un flux, framing)
3. Couche réseau (gère le routage)
4. Couche transport (envoie les infos de bout en bout et gère les erreurs)
5. Couche session (fournir un espace de noms pour relier les flux)
6. Couche de présentation (format donnée)
7. Couche d'application (normaliser le type commun d'échange (HTTPS...))

Architecture Internet (modèle TCP/IP)

- 5. Application
- 4. TCP/UDP
- 3. IP
- 2/1. Subnetwork



Framing



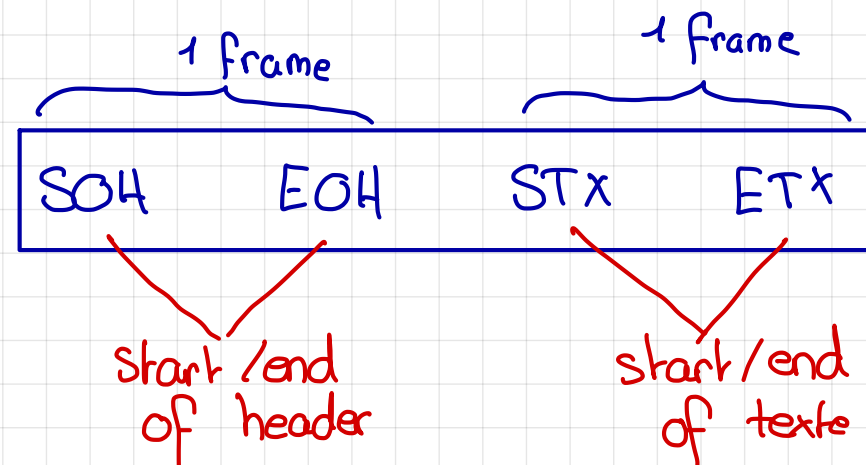
Deux familles de framing:

1. Orienté byte (historique)
2. Orienté bit

Approche Sentinelles

Caractère spécifique pour début, fin de la frame

exemple: BISYNC



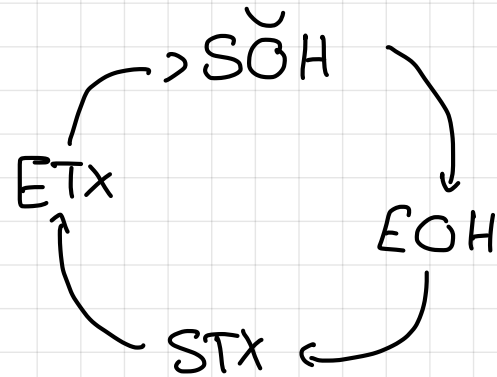
(à l'émission)
Byte stuffing: préfixer les caractères spéciaux par un caractère spécial (ESC)

(à la réception)
Byte unstuffing: faire l'opération inverse et enlever ESC lorsque c'est nécessaire

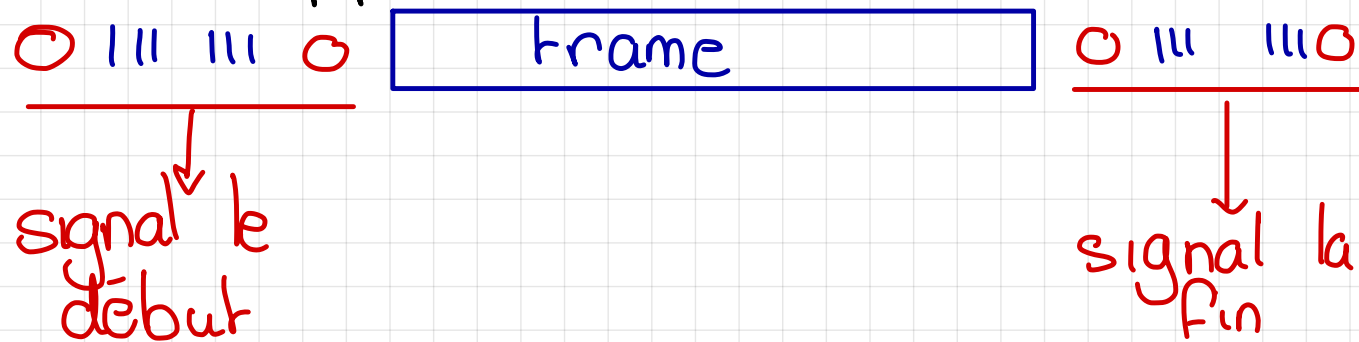
Exemple: - HelloWorld \ESC \ESC SOF

pas possible pq déjà dans une frame
manque un ETX donc BUG

State machine (hardware):



Bit stuffing: aussi approche sentinelle mais binaire. Un caractère sur 8 bit.



Si il voit une suite de 5×1 , il insère un 0

111101
111100

