

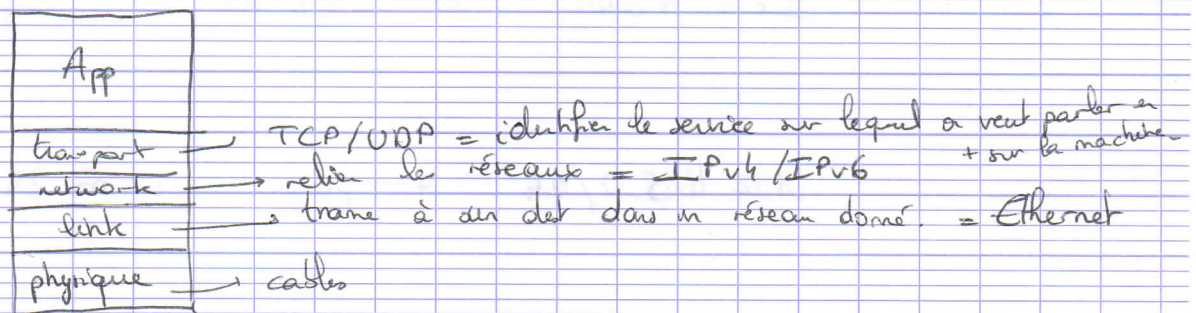
On va parler

- Procs / Thread / task
- scheduling
- Mécanisme de Synchro
- Réseau → OSI / TCP/IP / Switching / Routing / Cache physique
- File system

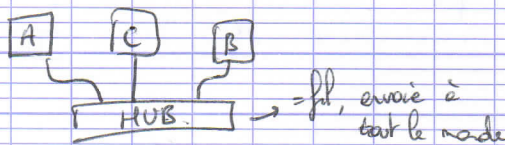


$N \times M \rightarrow N+M$

→ Ajouter un langage = faire que le front end.



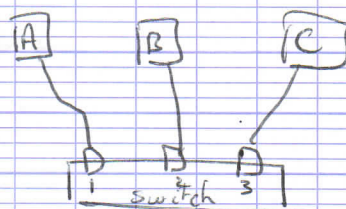
Ethernet:



/usr/include/net
arpa

trame:

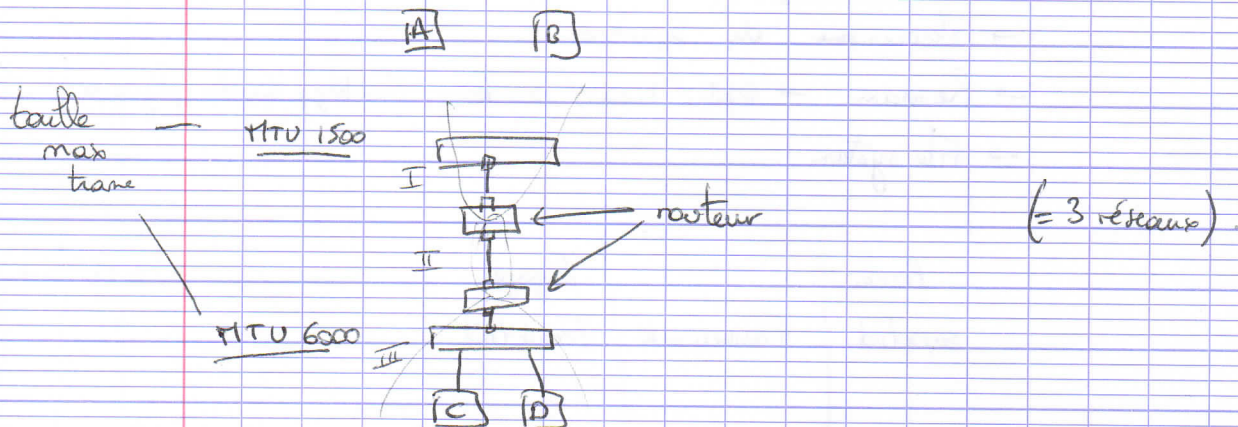
SRC MAC	Dest MAC	Type	size?	DATA	CRC
---------	----------	------	-------	------	-----



lorsque on envoie

mac	Port
A	1

Connait pas à qui envoyé → envoie à tout le monde
 Connait le port → envoie uniquement à lui.



IP = 4 octets : XX.XX.XX.XX (= 4 milliards d'adresses)

192.168.0/24 ← préfixe : m 24 premiers bits = m réseau

Table de routage :

pour aller au réseau III, passer au routeur I/II.
 " " " " II, " " " "

(par sa adresse IP).

IP

SRC Mac	Dest Mac
Type / Size	
IP SRC	IP Dest
Protocol	TTL
CRC	

de la suite

1^{er} dest. Mac pour A → C
 = routeur

→ decapsule, change adresse MAC
 → envoie au 2^e routeur ...

IP: fragmente les paquets (pour gérer les différentes taille max: MTU)

TTL: time to live du paquet (si un fragment n'arrive jamais, n'attend pas indéfiniment).
(en seconde)
(descend à chaque routeur)

0.0.0.0/0
= internet.

	Mac SRC Mac Dest
	type size
IP	IP SRC IP Dest
	Protocol TTL
UDP	Port SRC Port Dest

DNS: Domain Name System.

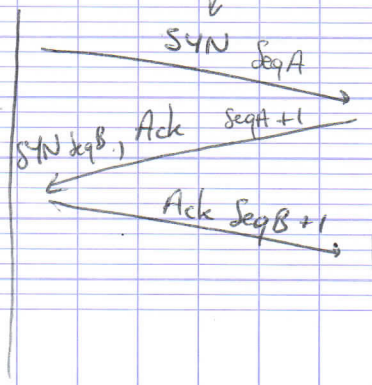
ex fr. epita. fr. ^{route}
France
Epita

AXFR. → récupère tous les noms
(UDP → Amplification Attack)

TCP:
(Ack)

Client

Server



1 bit: requête de synchronisation
(avec mon numéro de séquence)

2 boucles + 2 acknowledge
↳ d'accord sur n° de séquence.

App.

Socket = API ne change pas en fct de l'OS.

$fd = \text{socket}(\underbrace{\text{INET}}_{\substack{\uparrow \\ \text{info couche network}}}, \underbrace{\text{Datagram}}_{\text{UDP}} \text{ ou } \underbrace{\text{STREAM}}_{\text{TCP}})$

$\text{connect}(fd, \text{struct sockaddr}) \quad (= \text{SYN})$

$\text{bind}(fd, \text{sockaddr}) \rightarrow \text{port}$

$\text{listen}(fd, \text{---}) \rightarrow \text{se peut recup des paquets}$

$fd = \text{accept}(fd, \&\text{sockaddr}) \quad (= \text{SYN 2})$

$\text{send}()$

$\text{sendto}()$

$\text{recv}()$

$\text{recvfrom}()$