

IPv4

Adresse IP écrite sur 4 octets.

Deux parties : - adresse réseau (net id)
- adresse machine (host id)

Masque de sous-réseau : placer à 1 les bits de la partie réseau et à 0 ceux de la partie machine.

Exemple : adresse IP : 207.142.131.245

masque : 255.255.255.0 → 11111111.11111111.11111111.00000000

de 3 octets adresse réseau et 1 octet adresse machine.

de cette adresse correspond à la machine 245 sur le réseau 207.142.131.0

Classes	premier octet	masque
A :	0-127	255.0.0.0
B :	128-191	255.255.0.0
C :	192-223	255.255.255.0
D :	224-239	← utilisé pour le multicast
E :	240-255	← réservée.

adresses de réseau et de broadcast :

IP	masque	adresse réseau	adresse broadcast
10.10.10.10	255.0.0.0	10.0.0.0	10.255.255.255
192.168.150.35	255.255.255.0	192.168.150.0	192.168.150.255

↑ tous les bits partie machine à 0

↑ tous les bits partie machine à 1

on ne peut pas attribuer ces deux adresses à une machine

adresses réservées : les tout à 0 et les tout à 1 (partie machine)

à 10.255.255.255	
classe A : 10.0.0.0	(tout 0.x.x.x)
A : 127.0.0.0	(tout 127.x.x.x)
B : 172.16.0.0	à 172.31.255.255
C : 192.168.0.0	à 192.168.255.255

réservées à des réseaux privés.

24 bits 0 en binaire
masque 255.0.0.0

↓

IRga
Réseau de classe A compte 2^{24} machines (4671) machines
B compte 2^{16} machines (65536) machines
C compte 2^8 machines (258) machines

Notation CIDR: réseau 150.89.0.0
masque 255.255.0.0 → 16 bits à 1
notation CIDR du réseau 150.89.0.0/16

réseau : 192.168.144.0
masque : 255.255.252.0 → 11111111 11111111 11110000 00000000
notation CIDR : 192.168.144.0/20

⇒ Un réseau peut contenir $2^m - 2$ machines avec m nombre de bits utilisés à l'adressage de l'hôte.

Découpage en sous réseaux: soit 134.214.0.0/16
on veut le découper en 8 sous-réseaux

→ $8 = 2^3$. Le masque de chaque sous-réseau est obtenu en ajoutant 3 bits 1 au masque initial.

nouveau masque 255.255.224.0 (16 + 3 bits à 1 = 19 bits à 1)

→ net id de chaque sous-réseau : 16 premiers bits sont ceux de l'écriture binaire du préfixe d'adresse 134.214

les 3 suivants sont le numéro du sous-réseau : 000 (0), 001 (1), 010 (2)
... 111 (7).

le sous-réseau 0 : 134.214.(000 00000).0 = 134.214.0.0

sous-réseau 1 : 134.214.(001 00000).0 = 134.214.32.0

sous-réseau 2 : 134.214.(010 00000).0 = 134.214.64.0

...
sous-réseau 7 : 134.214.(111 00000).0 = 134.214.224.0

→ adresses de broadcast:

sous réseau 0: 134.214. (000 11111).255 = 134.214.31.255

sous réseau 7: 134.214. (111 11111).255 = 134.214.255.255

Plages d'adresses:

19 bits à 1 de petite machine fait 13 bits et
petite réseau fait 19 bits

Coupe 192.168.0.1/255.224.0.0

↳ adresse du réseau: toute la petite machine à 0:

sur 192.168.0.1 → 11000000 10101000.00000000.00000001
↓ (petite machine à 0)

192.160.0.0 ← 11000000 10100000 00000000 00000000

* adresse de broadcast: toute la petite machine à 1:

sur 192.168.0.1 → 11000000 10101000 00000000 00000001
↓

sur 192.191.255.255 ← 11000000 10111111 11111111 11111111

plage d'adresses définies par le coupe: 192.160.0.0 → 192.191.255.255

Méthode magique: 256 - octet significatif du masque

$$\text{ici } 256 - 224 = 32$$

multiples de 32: 0, 32, 64, 96, 128, 160, 192, 224, 256

→ 1^{ère} adresse = multiple du reste magique ≤ à l'octet correspondant

→ dernière = multiple suivant - 1.

ii le 2^{ème} octet est concerné de on prend 168 → multiple proche = 160

de 1^{ère} adresse 192.160.0.0

dernière adresse 192.191.255.255.

Décapage d'une plage: rien 192.168.160.0 / 255.255.255.0

- 550 tech
- 130 cam
- 10 dir

on veut découper en 3 sous-réseaux chaque population.

La plage d'origine est: nombre magique 256 - 224 = 32
octet significatif de l'adresse 160

c'est un multiple de 32. et $160 + 32 - 1 = 191$

de première adresse: 192.168.160.0 dernière 192.168.191.255

• Nombre d'adresses = $2^{\text{nombre de 0 des le masque}}$

par 550 tech on a $2^9 = 512$ et $2^{10} = 1024$

donc il y a 10 zéros dans le masque est.

1111111.1111111.11111100.0000000

est 255.255.252.0

• par 130 cam

$130 < 2^8$ de masque: 255.255.255.0

• par 10 dir

$10 < 2^4$ de masque 255.255.255.240

→ La plage dérivée en 192.168.160.0 ← début plage tech

le premier masque est 255.255.252.0

nombre magique 256 - 252 = 4

$160 + 4 - 1 = 163$

de dernière adresse fin = 192.168.163.255 ← fin plage tech

par les cam on continue en 192.168.164.0

on finit donc en 192.168.164.255

par les dir début = 192.168.165.0

fin: 192.168.165.15

← nombre magique = 16 sur le 4 octet
 $0 + 16 - 1 = 15$

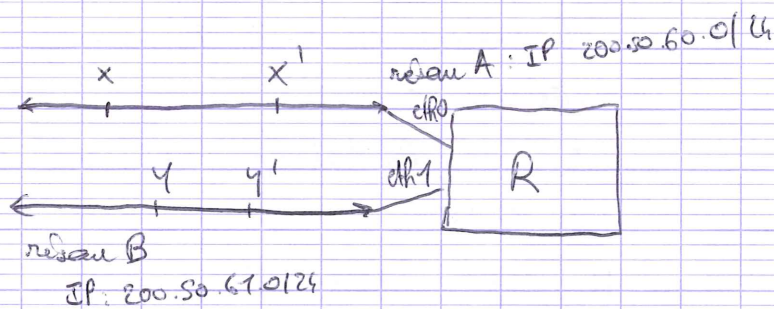
Routage

Un routeur est relié à au moins 2 réseaux

Son travail est de déterminer le prochain nœud du réseau auquel un paquet de données doit être envoyé. Il utilise pour ça une table de routage.

Le but du routage est que chaque machine et chaque routeur puisse envoyer un datagramme IP à n'importe quelle autre machine.

Exemple:



Les interfaces ont toutes une adresse IP :

- X interface eth0 IP 200.50.60.1
- X' interface eth0 IP 200.50.60.2
- Y interface eth1 IP 200.50.61.1
- Y' interface eth1 IP 200.50.61.2

R a deux interfaces et des 2 adresses IP : eth0 d'IP 200.50.60.3
 eth1 d'IP 200.50.61.3

→ table de routage de X :
 pour envoyer un datagramme à une machine du réseau 200.50.61.0/24, X peut directement le remettre grâce à l'interface 200.50.60.1

@ réseau	Masque	Passerelle	Interface
200.50.60.0	255.255.255.0	200.50.60.1	200.50.60.1
200.50.61.0	255.255.255.0	200.50.60.3	200.50.60.1

→ par exemple si 200.50.61.924, X doit envoyer au routeur 200.50.60.3 à l'aide de l'interface 200.50.60.1

→ table de X' :

@ réseau	Masque	Passerelle	Interface
200.50.60.0	255.255.255.0	200.50.60.2	200.50.60.2
200.50.61.0	255.255.255.0	200.50.60.3	200.50.60.2

→ table de Y :

@ réseau	Masque	Passerelle	Interface
200.50.61.0	255.255.255.0	200.50.61.1	200.50.61.1
200.50.60.0	—	200.50.61.3	200.50.61.1

↑

Y:

@	IP	Pass	Inter
~	~	200.50.61.2	200.50.61.2
~	~	200.50.61.3	200.50.61.2

R:

@	IP	Pass	Inter
200.50.60.2	/	200.50.60.3	200.50.60.3
200.50.61.0	/	200.50.61.3	200.50.61.3

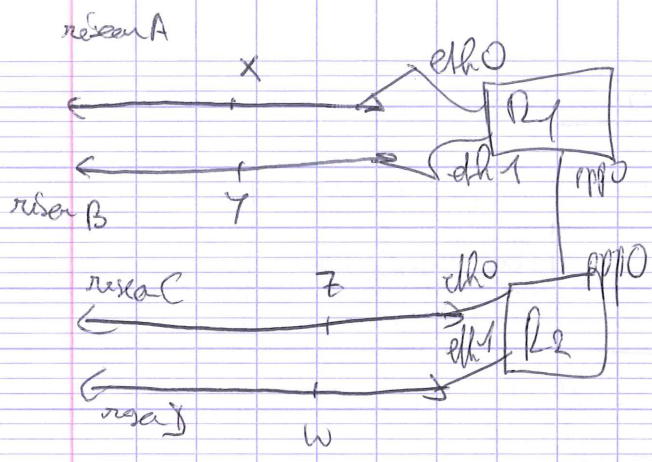
X envoie un datagramme à Y:

- X regarde sa table de routage. Y appartient au réseau 200.50.61.0/24
- X envoie le datagramme à 200.50.60.3 grâce à l'interface 200.50.60.1
- R reçoit datagramme
- R regarde destination de table de routage. Y appartient à 200.50.61.0/24
- R envoie le datagramme sur son interface 200.50.61.3
- Y reçoit le datagramme et sait qu'il est pour lui.

Route par défaut:

@	IP	Pass	Inter
200.50.60.0	/	200.50.60.1	200.50.60.1
0.0.0.0	0.0.0.0	200.50.60.3	200.50.60.1

pour tout le reste on envoie vers 200.50.60.3 qui s'appelle alors la passerelle par défaut de X.



A: 200.50.60.0
 B: 200.50.61.0
 C: 200.50.62.0
 D: 200.50.63.0

X: 200.50.60.1
 Y: 200.50.60.2
 Z: 200.50.62.1
 W: 200.50.63.1

R1 interfaces
 eth0 200.50.60.2
 eth1 200.50.61.2
 ppp0 200.50.64.1

R2:
 eth0: 200.50.62.2
 eth1: 200.50.63.2
 ppp0: 200.50.64.2

table de R1:

@	π	Pass	Inter
200.50.60.0	/	200.50.60.2	200.50.60.2
200.50.61.0	/	200.50.61.2	200.50.61.2
200.50.64.2	255.255.255.0	200.50.64.2	200.50.64.1
0.0.0.0	/	200.50.64.2	200.50.64.1

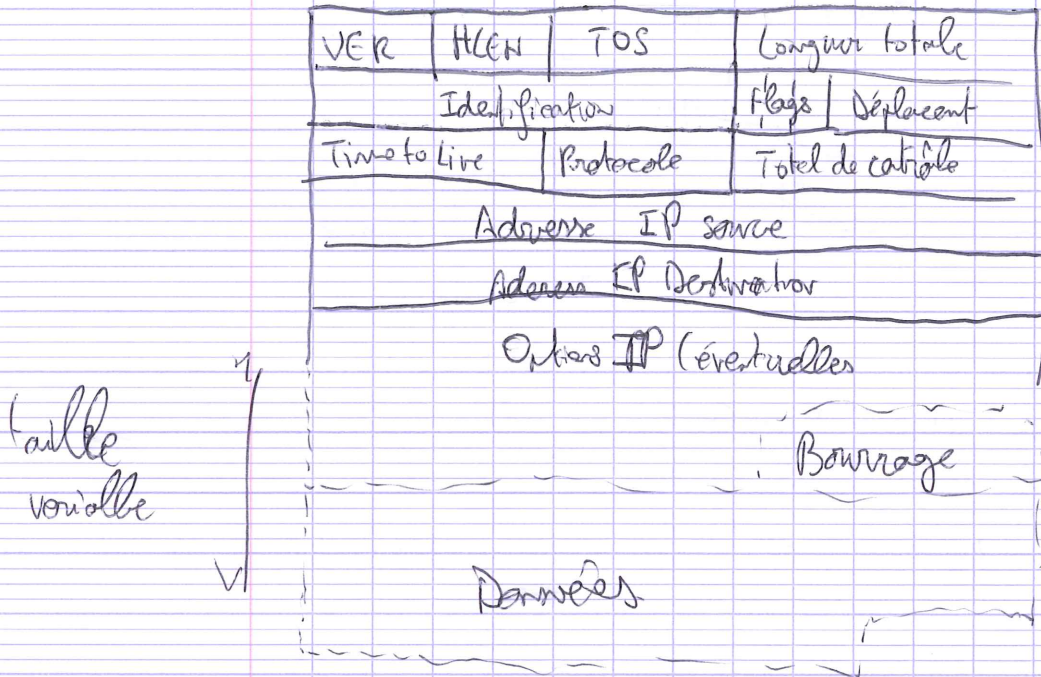
route vers w
 table de R2:

@	π	Pass	Inter
200.50.62.0	/	200.50.62.2	200.50.62.2
200.50.63.0	/	200.50.63.2	200.50.63.2
200.50.64.1	255.255.255.0	200.50.64.2	200.50.64.2
0.0.0.0	/	200.50.64.1	200.50.64.2

X envoie à W: X → R1 → R2 → W



Format d'un datagramme IP:



VER (4 bits): Version du protocole IP qui doit interpréter le datagramme
La plus déployée est 4 (0-100)

HLLEN (4 bits): à multiplier par 4 permet de connaître le nombre d'octets constituant l'entête (du début jusqu'aux données) valeur de 5 (20) à 15 (60) selon si y'a des options ou pas.

TOS (8 bits) qualité du service demandé pour ce datagramme. découpé comme ça:

Priorité	D	T	R	Inutilisé			
0	1	2	3	4	5	6	7

Prio (3 bits): 000 routine, 001 priority, 010 immédiate, 011 flash, 100 flash override, 101 critical, 110 internetwork control, 111 network control

D: à 1 l'acheminement du datagramme doit privilégier le délai (arriver le + vite)

T: à 1 indique que le datagramme peut être d'une communication qui nécessite un grand débit

R: à 1, indique qu'il faut privilégier la fiabilité (acheminer avec le + faible taux d'erreur)

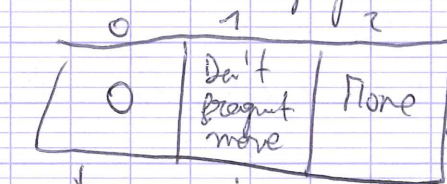
inutilisés: doivent être à 0.

Si le datagramme est + grand que ce que le réseau peut transporter, on le fragmente

Longueur totale (16 bits): taille totale en octets du datagramme
 $< \text{à } 65535 (2^{16} - 1)$

Identification (16 bits): numéro identifiant le datagramme de façon non ambiguë par rapport à sa source (@ source)
cela permet de rassembler les fragments d'un datagramme.

Flags (3 bits): Indicateurs de fragmentation



↓
utilisé à 0

↓
si à 1
indique qu'il
ne faut pas
fragmenter le
datagramme

↓
si le routeur n'a pas le choix de fragmenter il le détruit et renvoie une erreur.

← si à 1 indique que le datagramme est en fragment et que ce n'est pas le dernier.
Si à 0 indique que c'est le dernier fragment (ou qu'il n'est pas fragmenté)

Déplacement (13 bits)

En faisant $\times 8$ on obtient la position dans le datagramme d'origine du premier octet de données de ce datagramme.

Si à 0 le datagramme n'est pas fragmenté ou c'est le 1^{er} fragment
Si $\neq 0$ datagramme fragmenté.

TTL (8 bits): durée de vie en secondes du datagramme.

On élimine le datagramme s'il n'arrive pas dans le délai imparti.

Protocole (8 bits): Indique à quel protocole il faut remettre les données du datagramme.

0: IP // 1: ICMP // 6: TCP // 47: UDP

Checksum (16 bits): contrôle l'intégrité de l'entête.
Si celui calculé par le destinataire est \neq de celui du donneur: on le détruit.

@ source: @ IP de l'émetteur
@ Dest: @ IP du destinataire.

Options: facultatives

Remarque: Pour limiter la taille des options jusqu'à un multiple de 4 octets

Données: seront communiquées au protocole du champ protocole si le checksum est le bon.

Taille max 65535 - taille entête.

Ethernet: elle est de forme : 00:DJ:FF:17:1E:03
 → 48 bits (6 octets)
 3 premiers octets indiquent le constructeur.

Adresse de diffusion dans les réseaux ethernet est ff:ff:ff:ff:ff:ff (que des 1)
hexadécimel.

Trame de données:

si < 46 octets
↑

Norme 802.3: Préambule | SFD | @ Dest | @ Source | Long | Données | Remplissage | CRC

70 10 60 60 20 46 à 1500 40

norme ethernet: Préambule | SFD | @ Dest | @ Source | Type | Données | Remplissage | CRC

70 10 60 60 20 46 à 1500 40

- Débit d'émission / réception : 10 Mb/s (10B/ps)
- longueur des trames : 26 octets pour le protocole
 de 72 octets mini (46 utiles)
 et 1526 octets maxi (1500 utiles)

ICMP/IGMP:

Données
 Dans le champ ~~type~~ on peut avoir une trame de différent type selon la valeur de ~~chaque~~ ^{chaque} type.

champ type	type de protocole
0x 0800	IP (Internet) ← cf routage page 3
0x 0806	ARP
0x 80D5	ICMP SNA
0x 8035	RARP
0x 86DD	IPv6

0-7

8-15

16-31

frame ARP:

Hardware type		Protocol type
Hardware address length	Protocol Address length	Operation
Sender Hardware Address		
Sender Protocol Address		
Target Hardware Address		
Target Protocol Address		

HT: 01 - Ethernet 02 - Experimental EthernetPT: 0x0800 - IPHAL: 01 - Token ring 06 - EthernetPAL: 04 - IPv4 16 - IPv6OP: 01 - Request 02 - ReplySHA: Adresse MAC source des le cadre EthernetSPA: Adresse IP de source des le cadre de TCP/IPTMA: Adresse MAC destination des le cadre Ethernet (si c'est une demande ARP → 0)IPA: Adresse IP de destination des le cadre de TCP/IP.segment TCP:

Port source		Port destination	
Numéro de séquence			
Acquittement			
lg. tête	6 bits réservés	6 options	fenêtre
Checksum		Pointeur de données urgent	
Options			
Données			

PS: n° du port source

PD: n° du port destination

Num seq: numéro de séquence du premier octet de ce segment

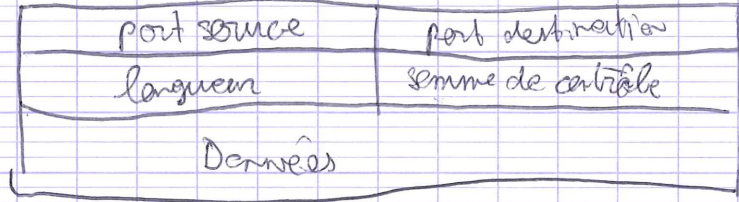
num acquit: numéro de séquence du prochain octet attendu.

taille de l'entête: longueur de l'entête en mots de 32 bits

fenêtre: taille de fenêtre demandée

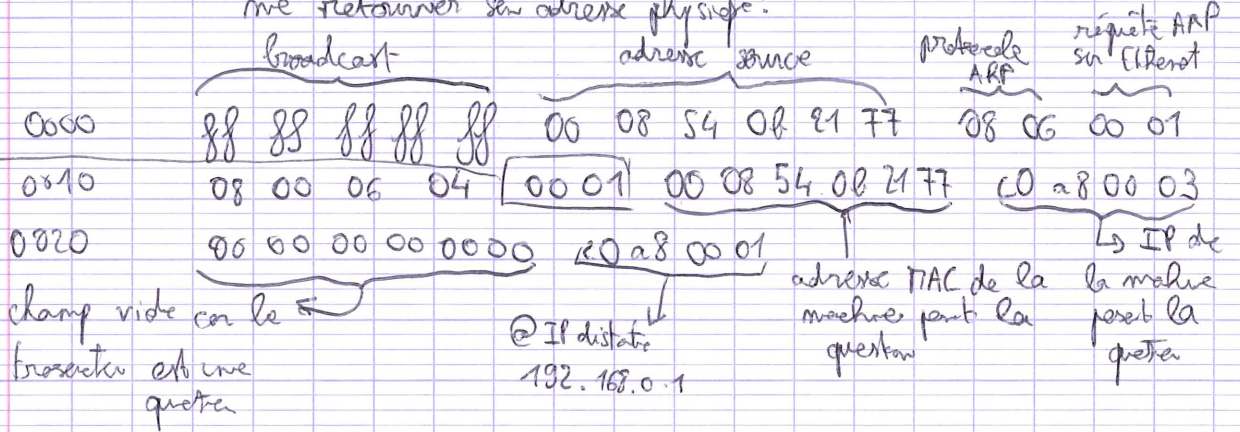
pointeur: position des données données urgentes

segment UDP:



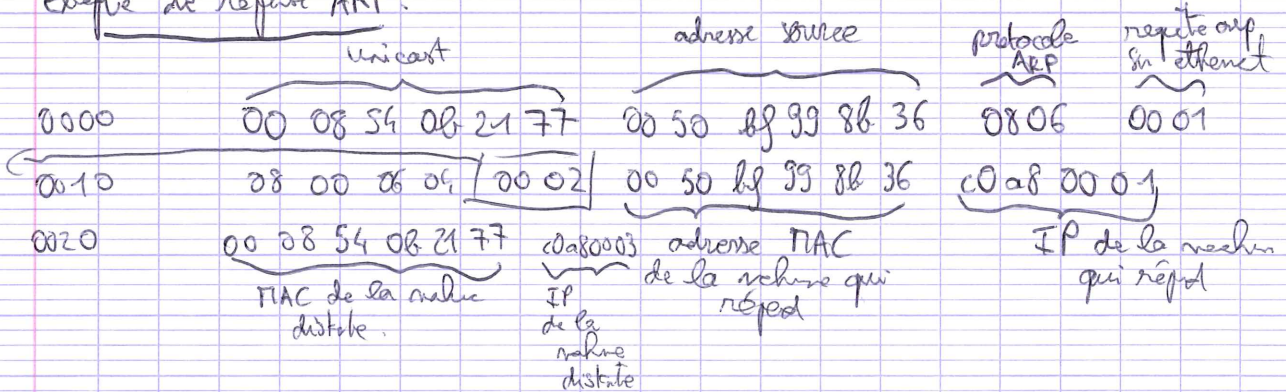
Exemple de requête ARP: "Je suis l'hôte "00 08 54 08 21 77", est-ce que l'hôte possédant l'adresse IP 192.168.0.1 peut me retourner son adresse physique?"

requête ARP



Exemple de réponse ARP:

réponse ARP



IPv4: 192.168.0.1 / 255.255.255.0

Masque de subnet → 111...100...0
 1 ⇒ Partie réseau
 0 ⇒ Partie machine

Adresse de broadcast:
 Dernière addr. du subnet (Partie machine = 1)
 ⇒ 192.168.0.255 ici

Adresse de réseau:
 Prem. addr. du subnet (Partie machine = 0)
 ⇒ 192.168.0.1 ici

Ex:
 IP: 10.10.10.10 Masque: 255.0.0.0
 Addr. réseau: 10.0.0.0 Addr. broadcast: 10.255.255.255

Adresses privées:

- Classe A: 10.0.0.0
- Classe A: 127.0.0.0 (loopback/localhost)
- Classe B: 172.16.0.0 à 172.31.0.0
- Classe C: 192.168.0.0 à 192.168.255.0

+ classes D/E

- Classes Premier octet
- A: 255.0.0.0 (Pre. 0-127)
 - B: 255.255.0.0 (128-191)
 - C: 255.255.255.0 (192-223)
 - D: multicast (224-239)
 - E: réservée (240-255)

CIDR

CIDR	Masque	etc...
/1	128.0.0.0	128 etc...
/2	192.0.0.0	128 255.255.128.0
/3	224.0.0.0	128 255.255.192.0
/4	240.0.0.0	128 255.255.224.0
/5	248.0.0.0	etc...
/6	252.0.0.0	128 255.255.255.128 2 subnets
/7	254.0.0.0	128 255.255.255.192 4 subnets
/8	255.0.0.0	128 255.255.255.224 8 subnets
/9	255.128.0.0	etc...
/10	255.192.0.0	131 255.255.255.254 none
		132 255.255.255.255 1/256 C

Protocoles:
 ± P. Paquets
 ARP/RARP: Résolution d'adresses (RARP = machine demand son IP)
 MTU: Injection
 ICMP: Ping & messages de ctrl

MAC = 48 bits

OSI

- Couche 1: Bit
- Couche 2: Trame
- Couche 3: Paquet
- Couche 4: Segment (TCP) / Datagramme (UDP)

Table range

passerelles, VLAN
 ↳ quelle est l'adresse de base

↳ Mapping addresses

↳ Créer 1 réseau

eth/24/16

QCM

- 1) 10.1.11.
 172.16.4.4
 192.168.5.5
 224.66.6

2)

3) 255...240 = 16 subnets
 $255 \div 16 \Rightarrow$ 14 machines

4) Addr IP + table routage

5) /30 \Rightarrow 255.255.255.252

192.168.17.67

64 subnets

6) /29 \Rightarrow 255.255.255.248

32 subnets

192.168.17.134

256 - 252 = 4
 4
 8
 16
 32
 40
 60 15
 64 16
 68
 67 dans les multiples
 $\hookrightarrow 16 \times 4 = 64$
 $68 - 1 = 67$

256 - 248 = 8

8
 16
 24
 32
 40
 48
 56

80
 88
 120 15
 128 16
 136

192.168.17.128

.135

134 dans les multiples

de 8 : $16 \times 8 = 128$ prem. adr. réseau

$136 - 1 = 135$ dern. adr. réseau

\Rightarrow 192.168.17.64
 .67

D
D de L.
Comm.

	PERS	ENR	ACT
Juge	CONSULTER	VOIR	VOIR
Proc	~		
Pol:	~	VOIR	
OBJ	?	CREATE	
Pol: to OBJ			

Note: A circled 'VPOPT' is written between 'ENR' and 'ACT' with arrows pointing to 'VOIR' in both columns.

Polices

```
CREATE VIEW en-juge AS  
SELECT n.debut, statut,  
FROM ENQUETES e, Personnes p  
WHERE e.juge = p.num;
```