Licence Professionnelle ASSR MRIT
Université Sorbonne Paris Nord
(USPN)

Module M51
Réseaux TCP/IP

Jean-Vincent Loddo
Département R&T – IUT de Villetaneuse

Supports

- **Ce document**

qui complète vos notes de cours **manuscrites**
en reproduisant la plupart des contenus dessinés au tableau par votre enseignant

- **Logiciel (pour les travaux pratiques) :**

Marionnet
J.V. Loddo et L. Saiu

Téléchargeable à l'adresse :
`https://www.marionnet.org/downloads/Ubuntu_16.04_LTS_20180905.ova`

- **Livre conseillé :**

Cours d'introduction à TCP/IP
François Laissus
(Version du 25 février 2009)

En consultation libre et téléchargeable (entre autres) à l'adresse :
`https://doc.lagout.org/network/Cours%20d.introduction%20tcp-ip.pdf`

Plan des séances du module
M51 Réseaux TCP/IP

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)
2) En-tête IP (chapitre 4 du livre de Laissus)

Cours n°4
1) routage (statique) IP
2) ICMP

Cours n°5
1) UDP (chapitre 5 du livre de Laissus)
2) TCP (chapitre 6)

Plan du cours n°6
2) SNAT (Source NAT)
3) DNAT (Destination NAT)

Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

Plan des séances du module

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) En-tête IP (chapitre 4 du livre de Laissus)
2) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)

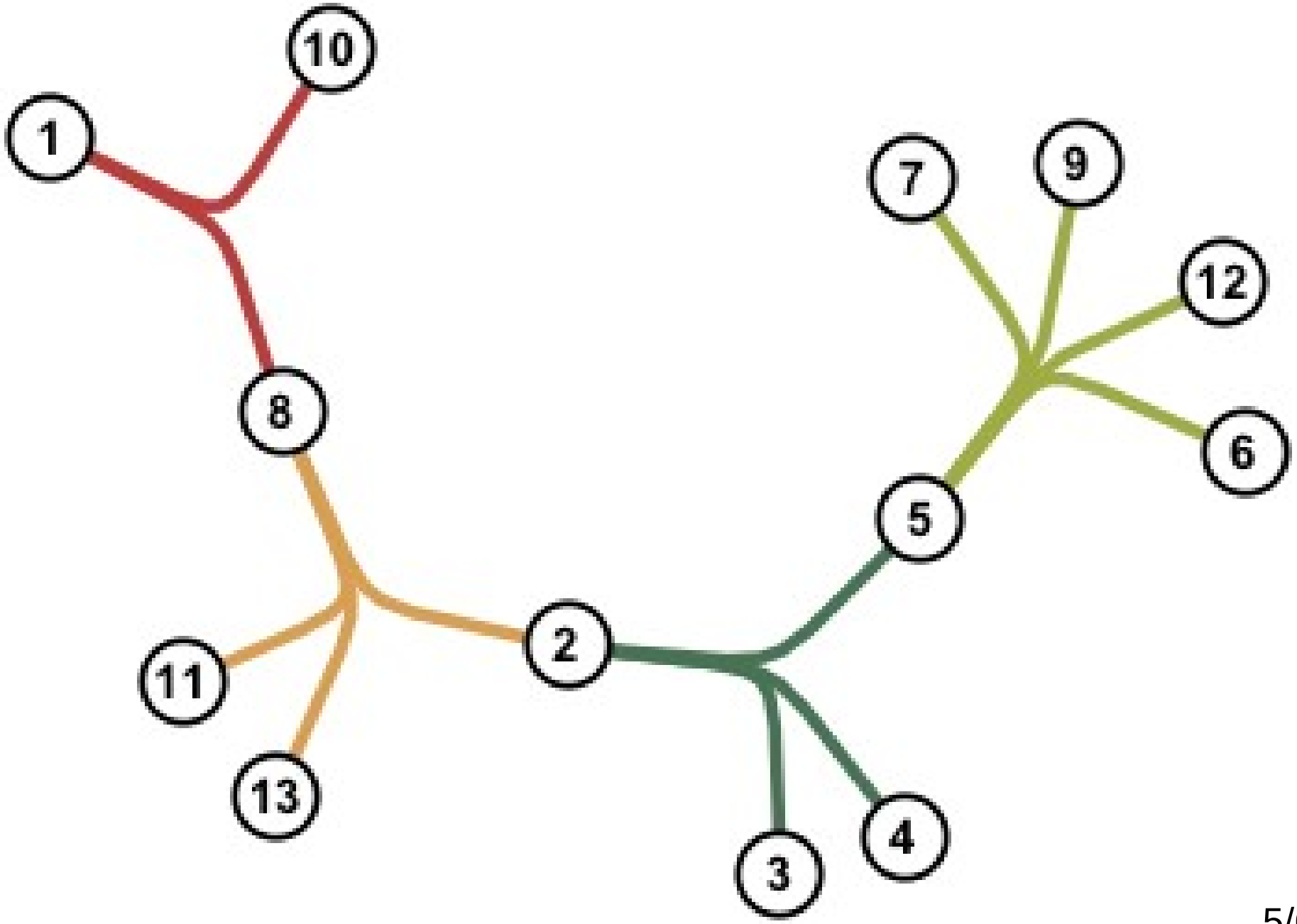
Cours n°4
1) routage (statique) IP

Cours n°5
1) ICMP
2) UDP (chapitre 5 du livre de Laissus)
3) TCP (chapitre 6)

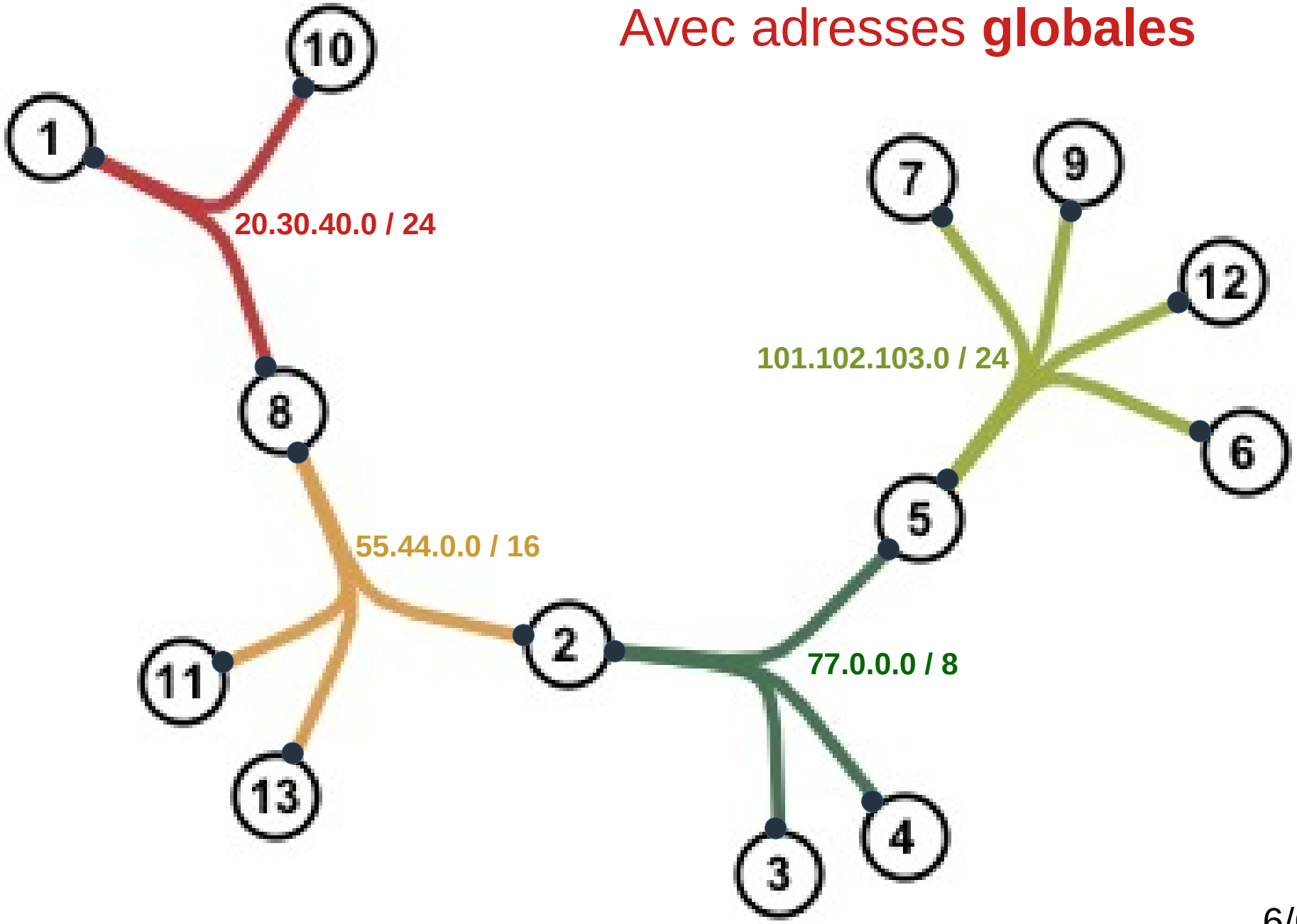
Plan du cours n°6
1) En-tête TCP
2) SNAT (Source NAT)
3) DNAT (Destination NAT)

Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

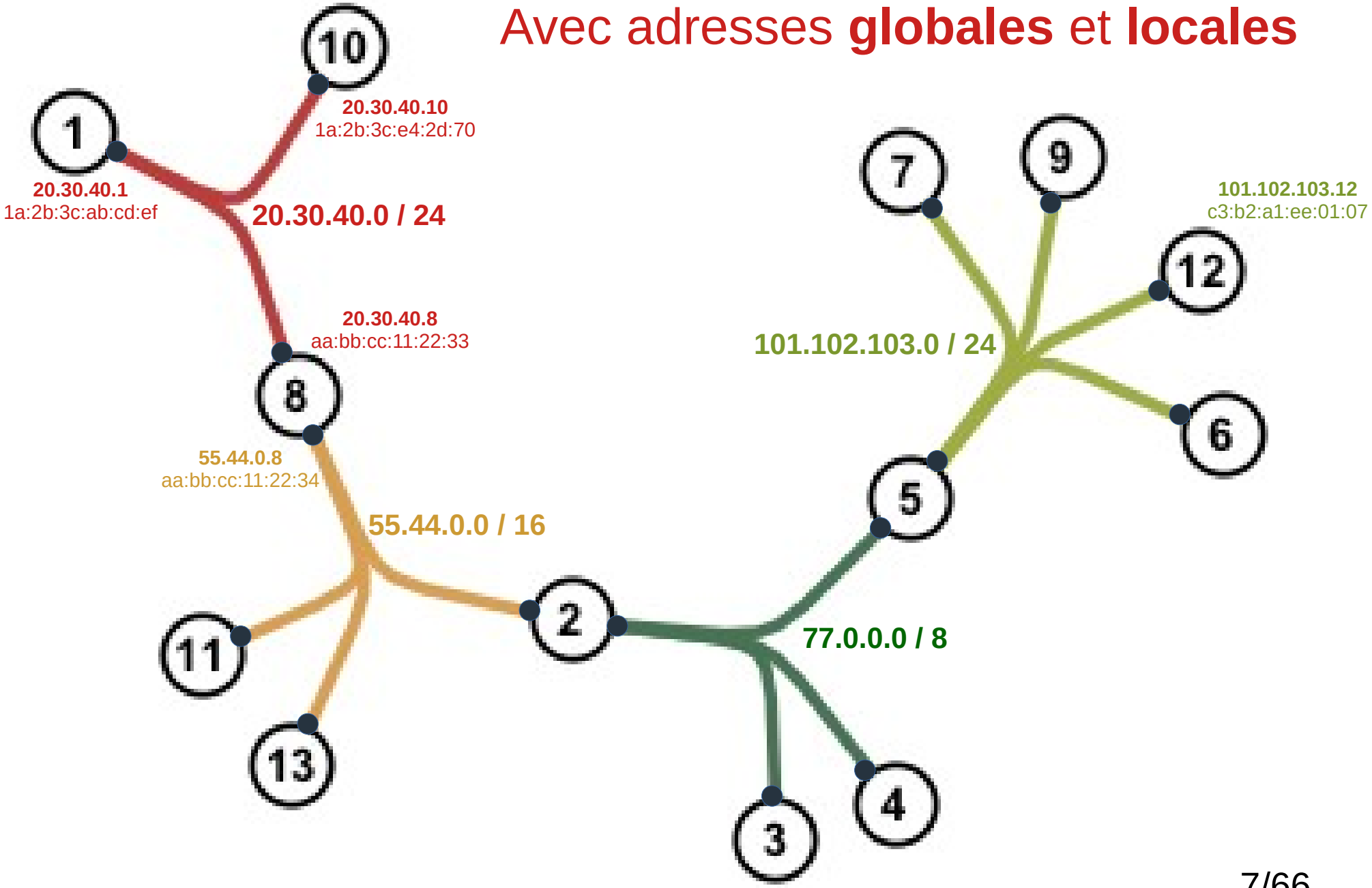
Un hypergraphe maquette d’Internet



Un hypergraphe maquette d'Internet
Avec adresses **globales**

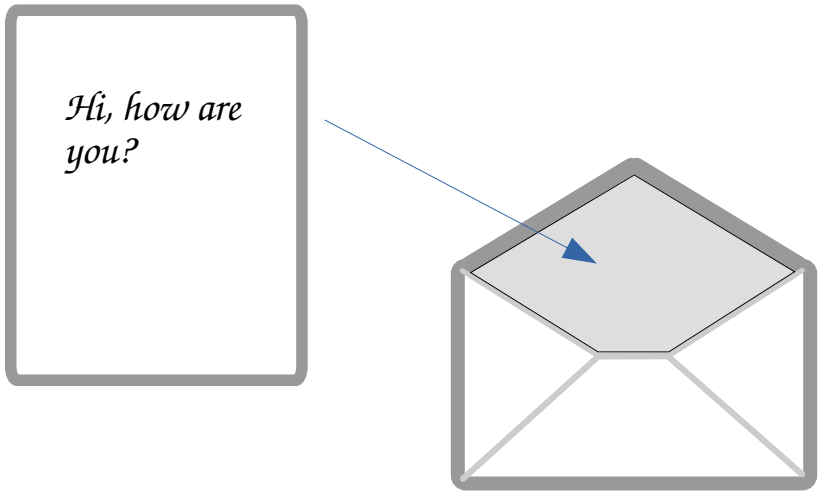


Un hypergraphe maquette d'Internet
Avec adresses **globales** et **locales**



Le principe de l'enveloppe

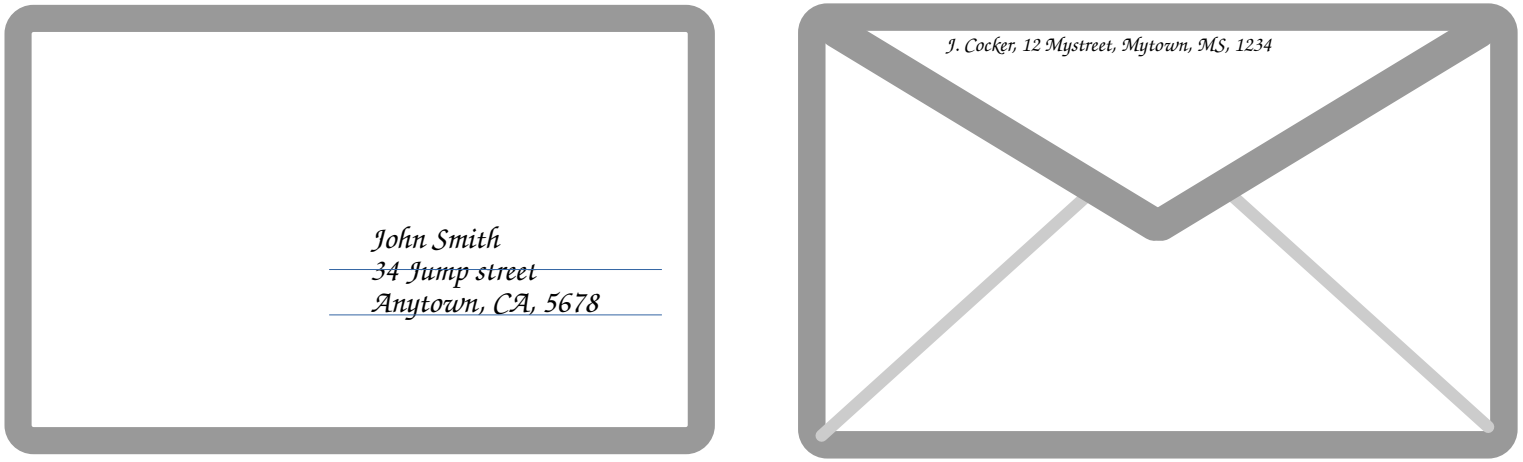
un **message**



une **enveloppe**

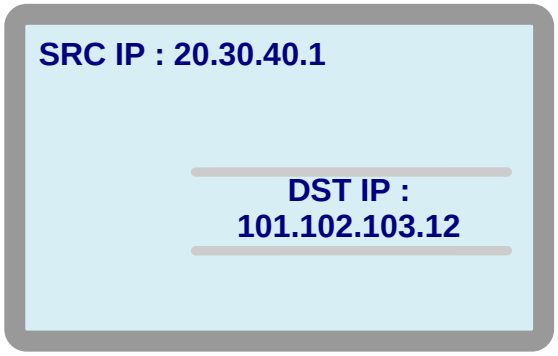
un **destinataire**

un **expéditeur**

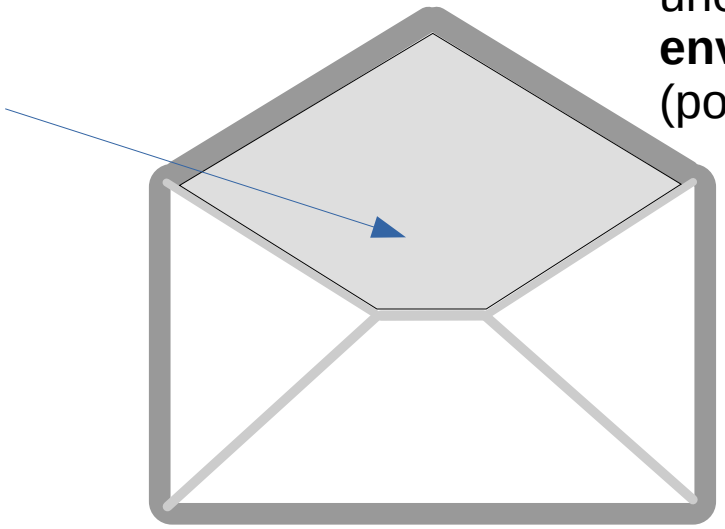


Le principe des enveloppes imbriquées (« encapsulation »)

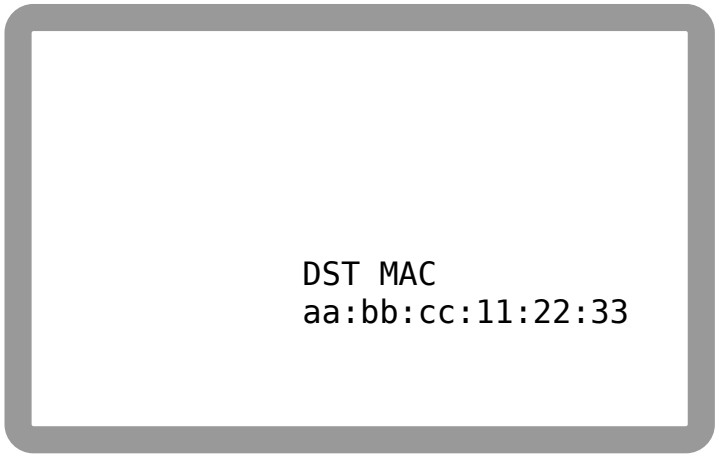
une **1ère** enveloppe, avec le **message**, bien **adressée** (avec adresses **globales**, pour un **voisin ou pas**)



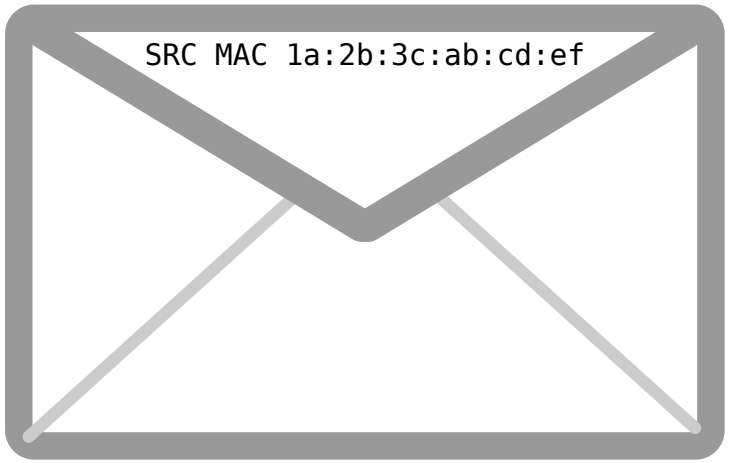
une **2ème enveloppe** (pour un **voisin**)



le **destinataire** (adresse **locale**, voisin de la même hyper-arête)

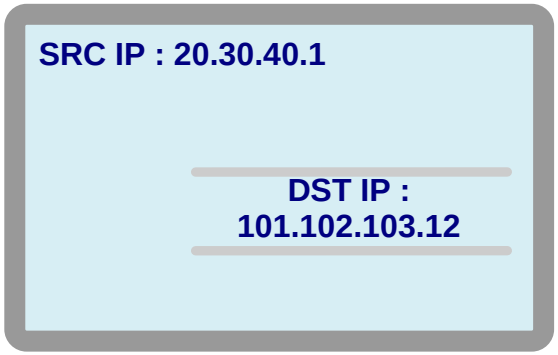


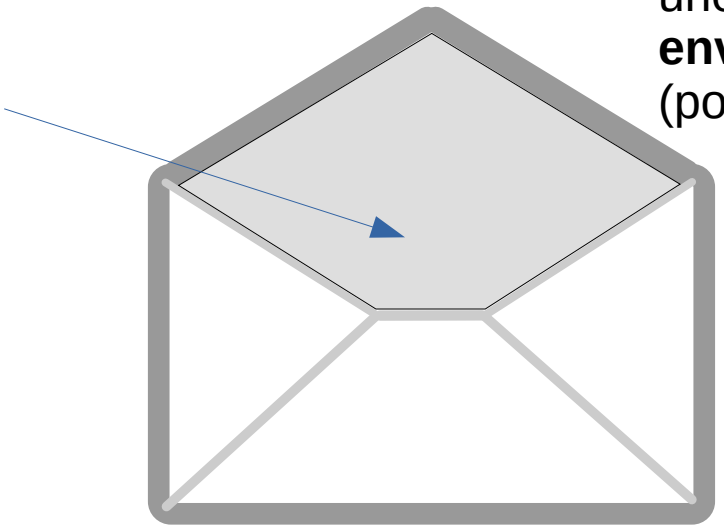
l'**expéditeur**



Le principe des enveloppes imbriquées (« encapsulation »)

une **1ère** enveloppe, avec le **message**, bien **adressée** (avec adresses **globales**, pour un **voisin ou pas**)

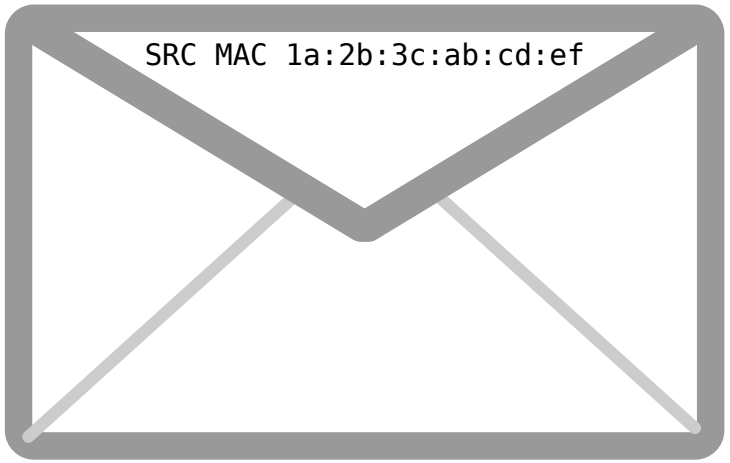


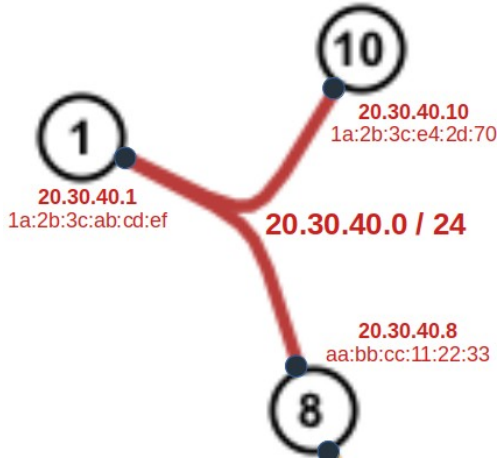


une **2ème enveloppe** (pour un **voisin**)

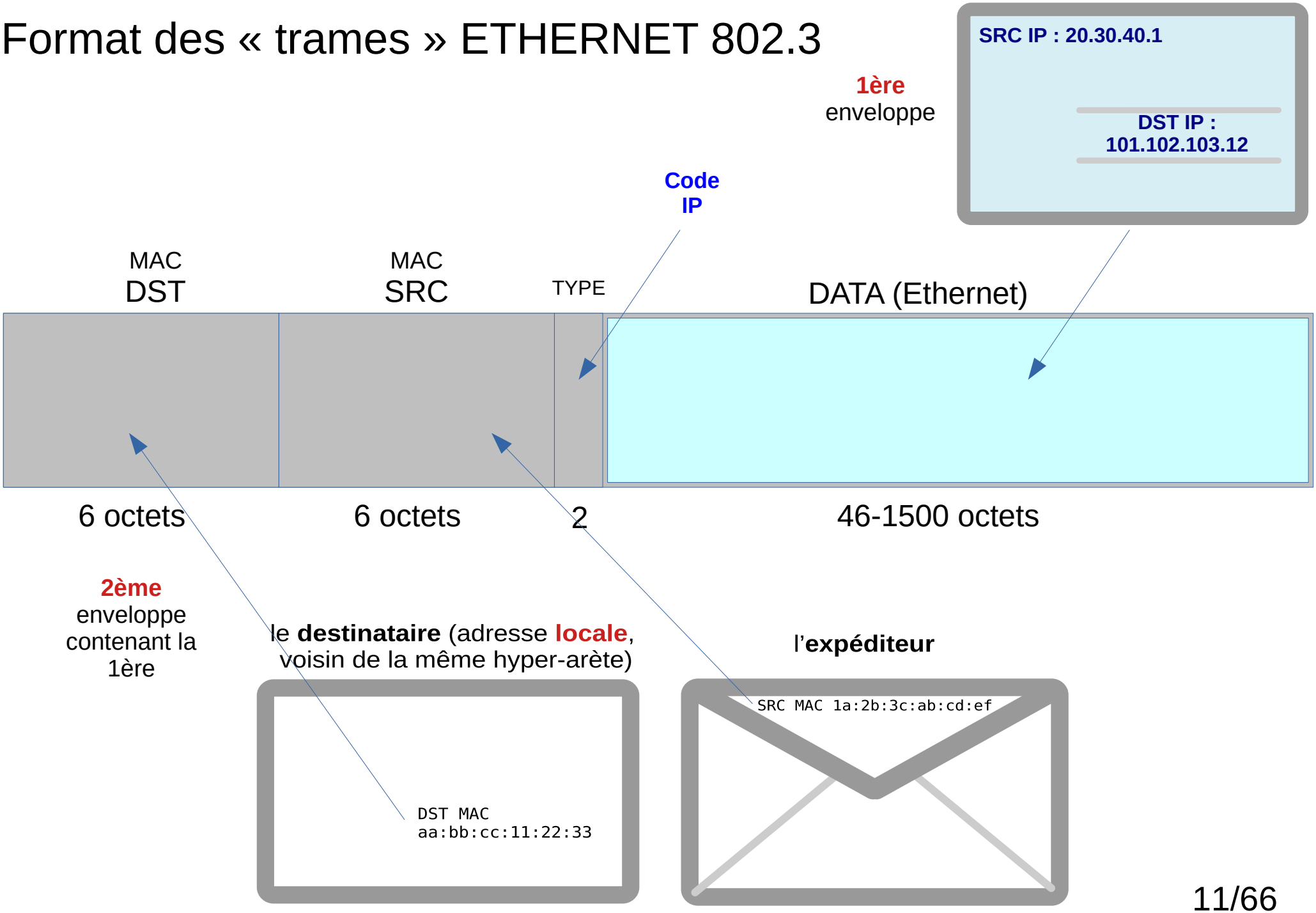
le **destinataire** (adresse **locale**, voisin de la même hyper-arête)

l'**expéditeur**



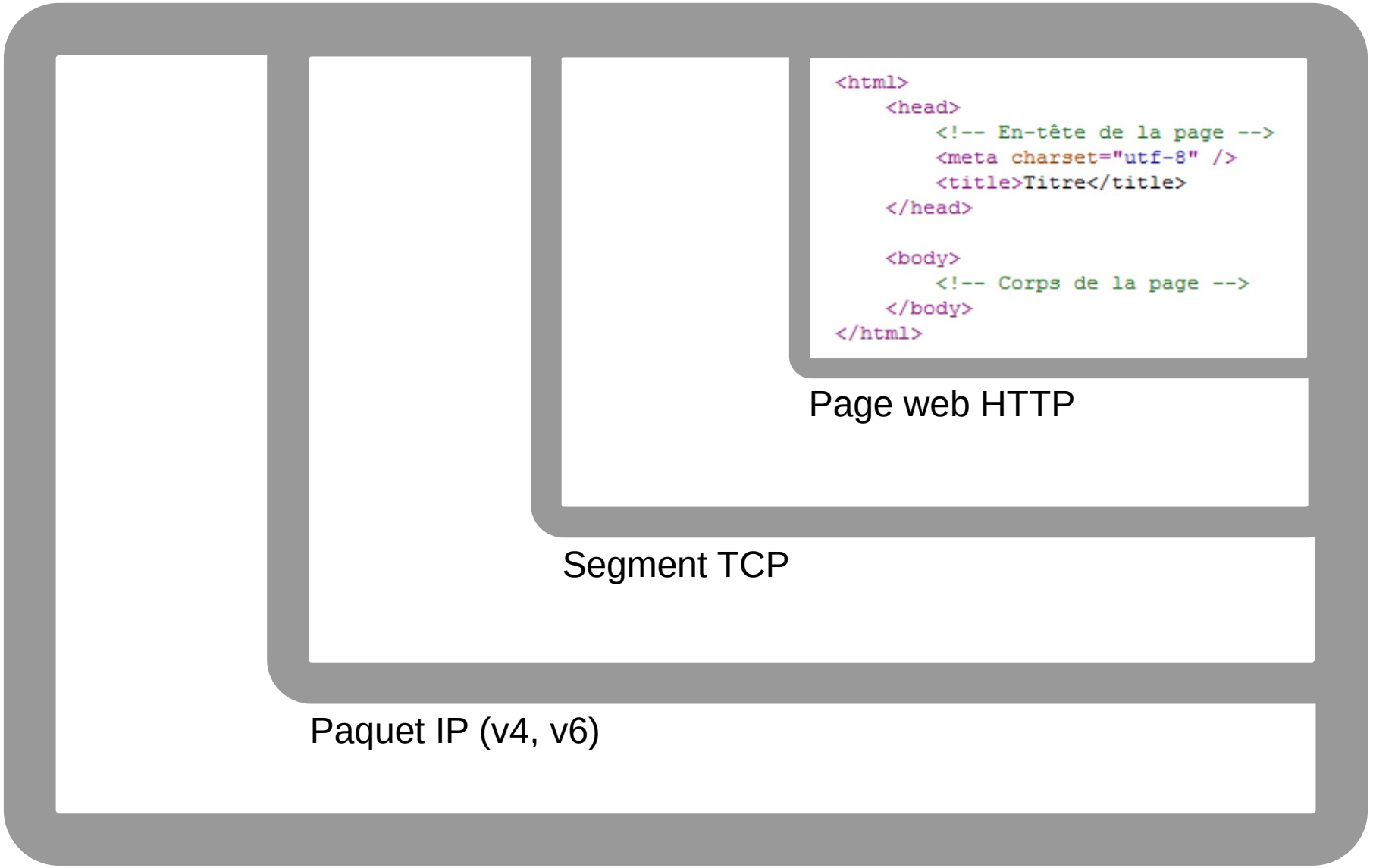


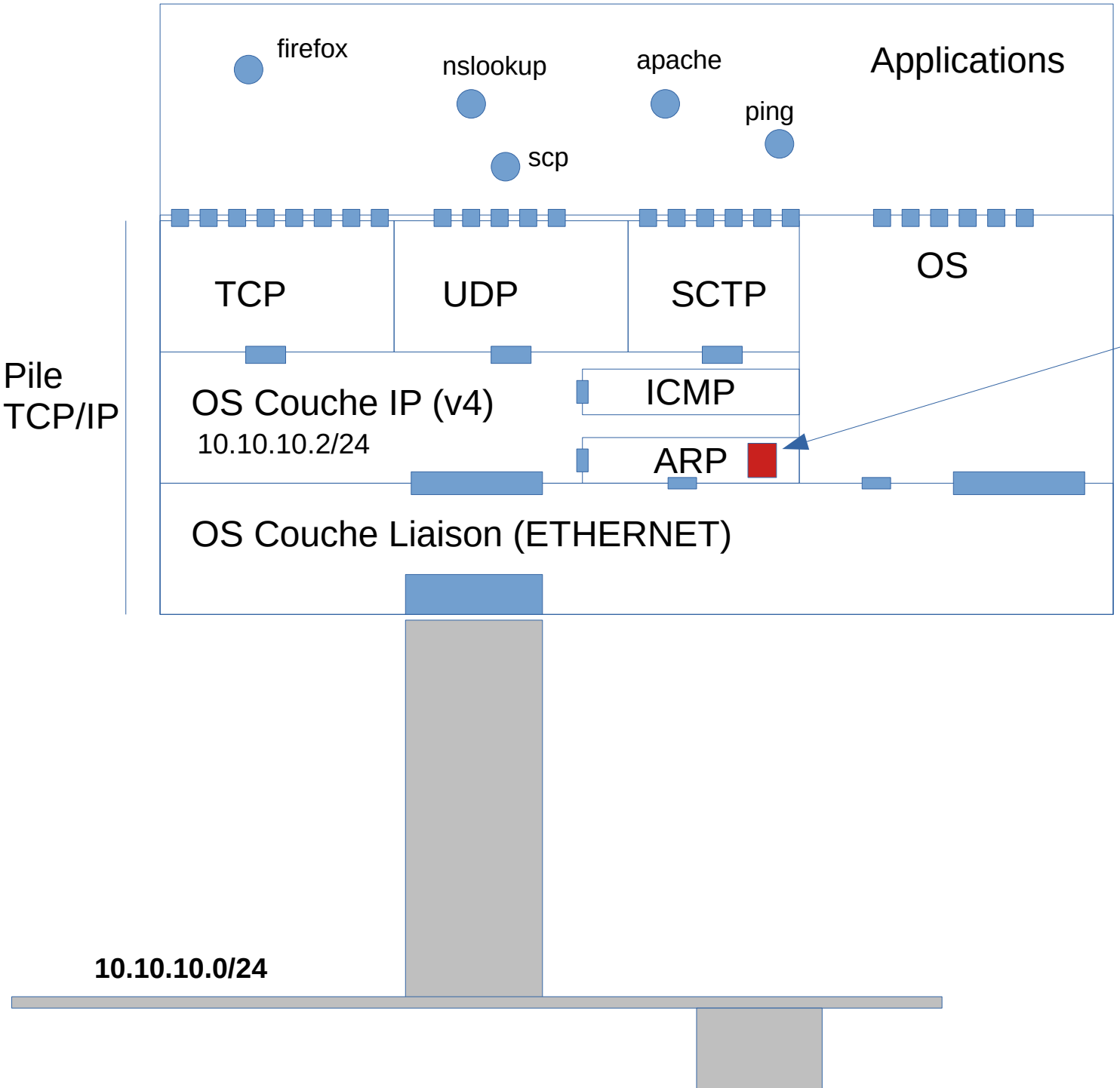
Format des « trames » ETHERNET 802.3



Encapsulation à plusieurs niveaux

Exemple : ETH / IP / TCP / HTTP





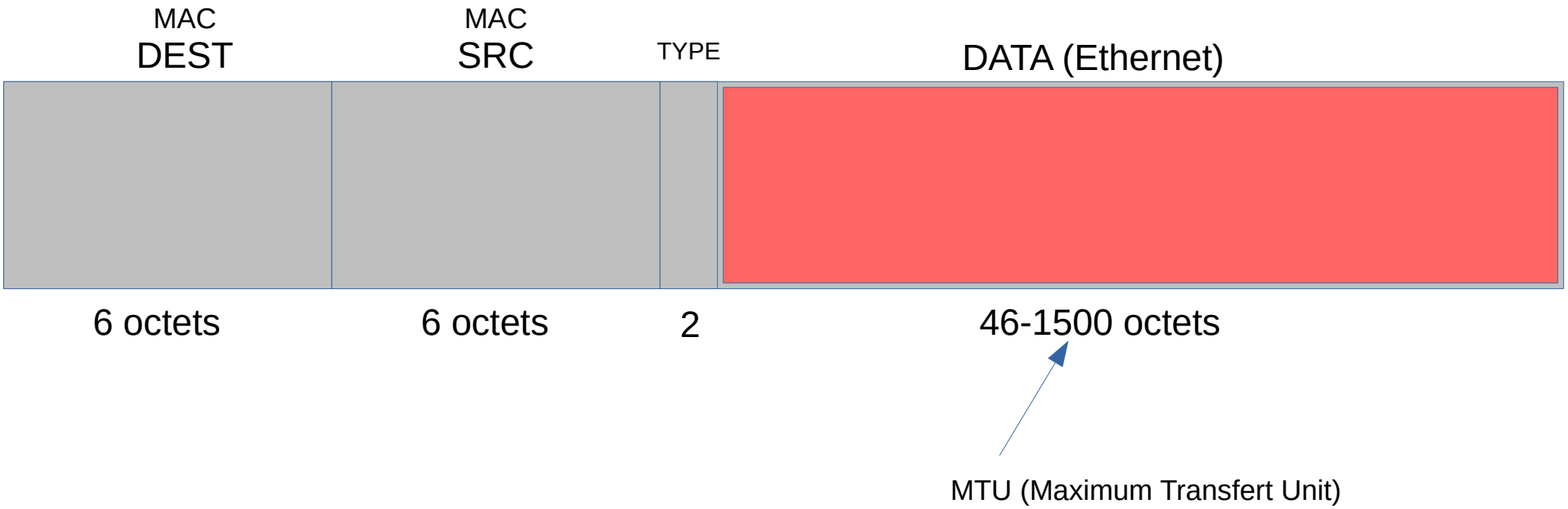
Address Resolution Protocol:

Traduction

Adresse IP →	Adresse Physique (MAC)
4 octets	6 octets
192.168.42.4	02:04:06:e5:5f:61

Table ARP
des associations
Adresse IP → Adresse MAC

« trame » ETHERNET 802.3



Plan des séances du module

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)
2) En-tête IP (chapitre 4 du livre de Laissus)

Cours n°4
1) routage (statique) IP
2) ICMP

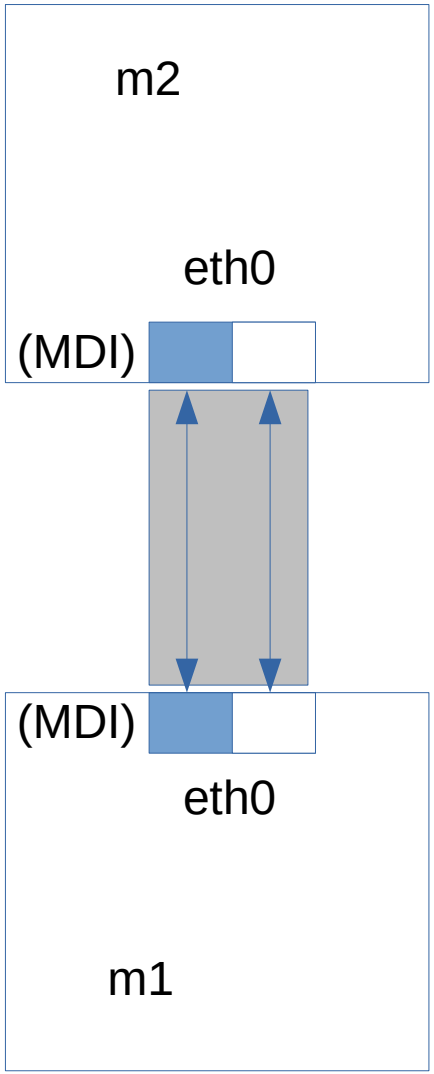
Cours n°5
1) UDP (chapitre 5 du livre de Laissus)
2) TCP (chapitre 6)

Plan du cours n°6
2) SNAT (Source NAT)
3) DNAT (Destination NAT)

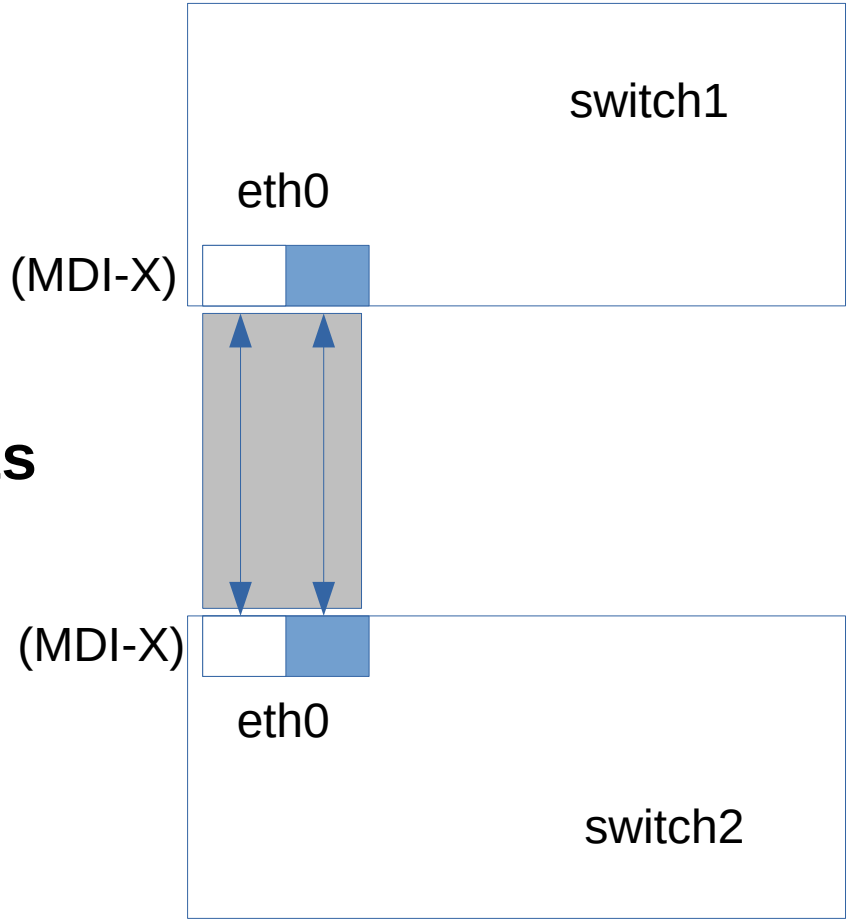
Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

Câblage Ethernet (802.3)

ERREURS (de câblage)

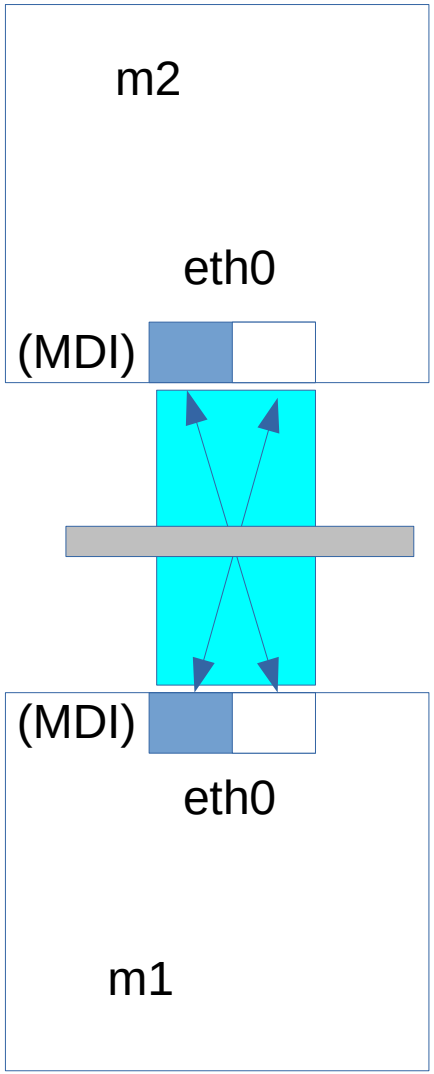


Câbles droits

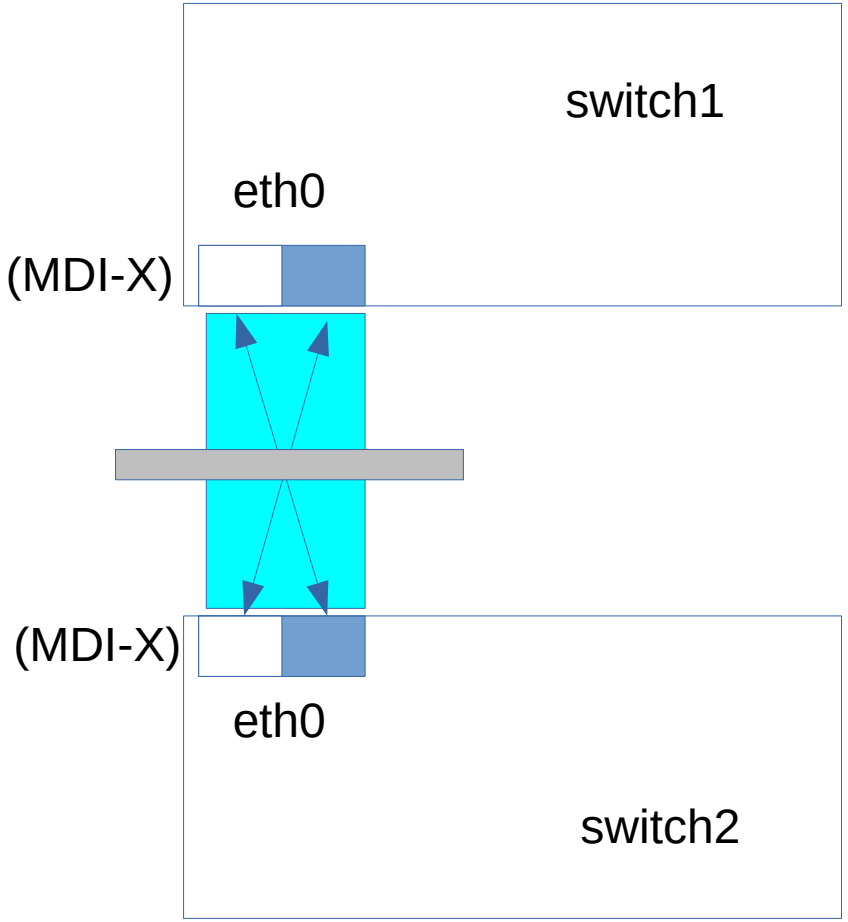


Câblage Ethernet (802.3)

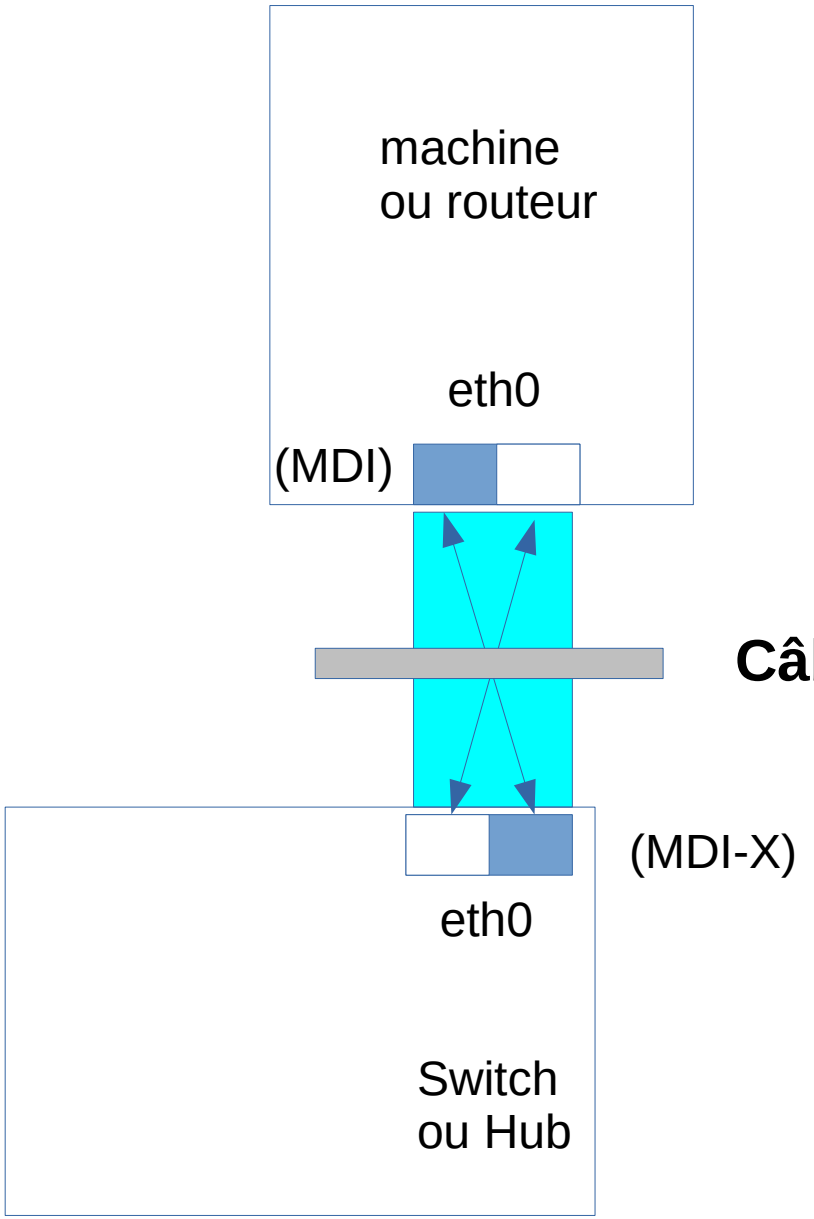
Câblage correct



Câbles croisés



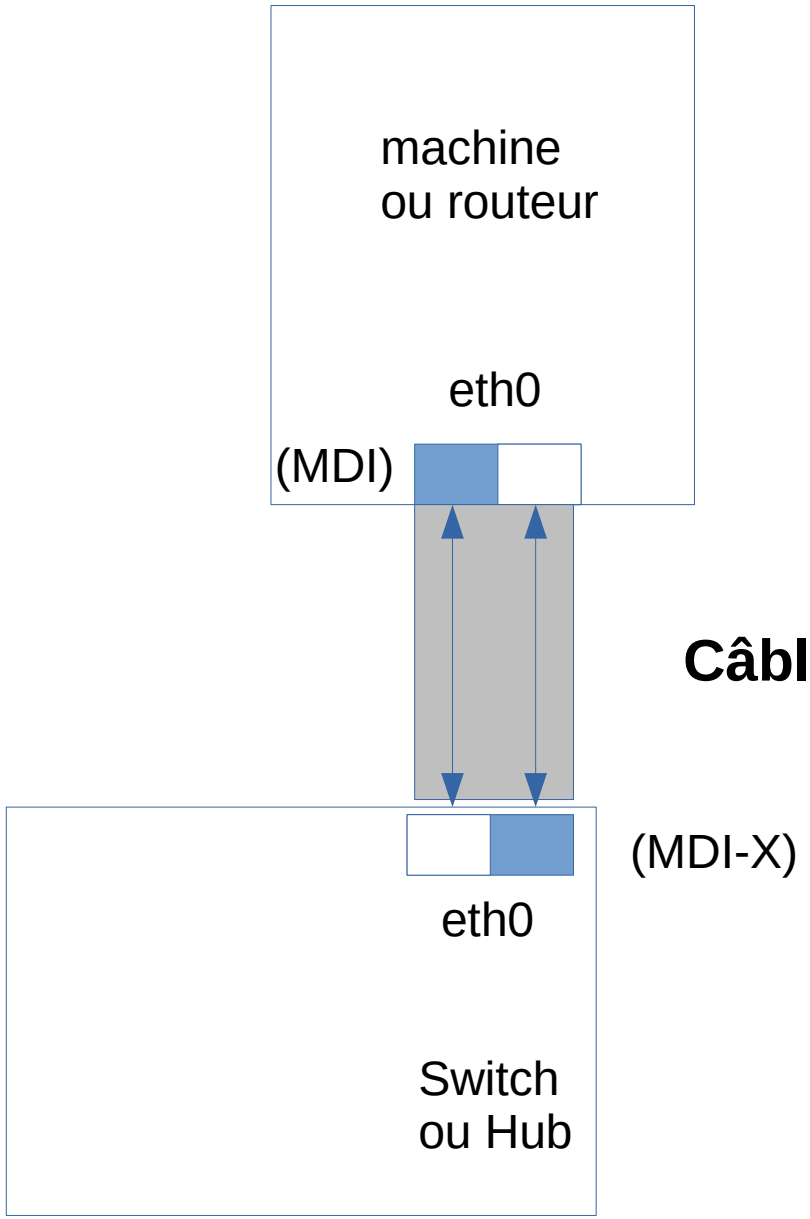
Câblage Ethernet (802.3)



Câbles croisé

ERREUR
(de câblage)

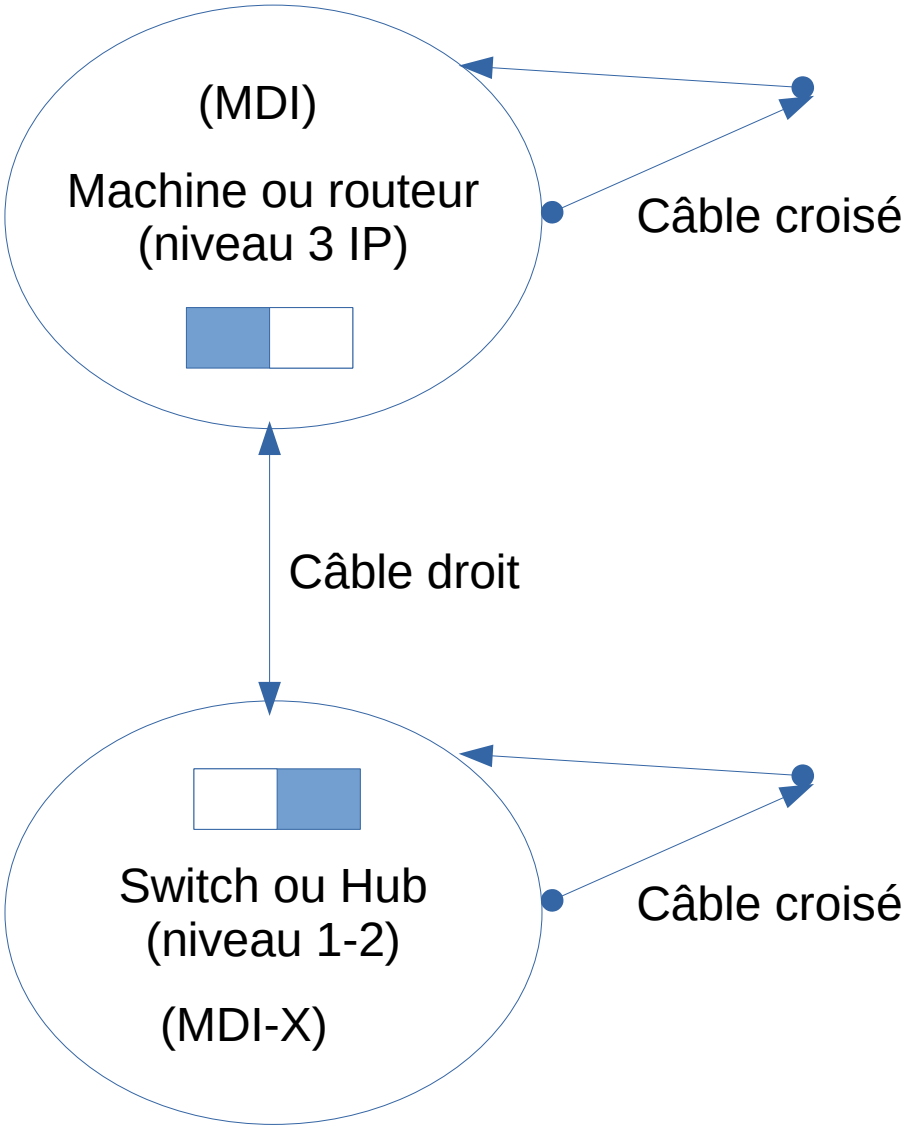
Câblage Ethernet (802.3)



Câble droit

Câblage CORRECT

Câblage Ethernet (802.3)



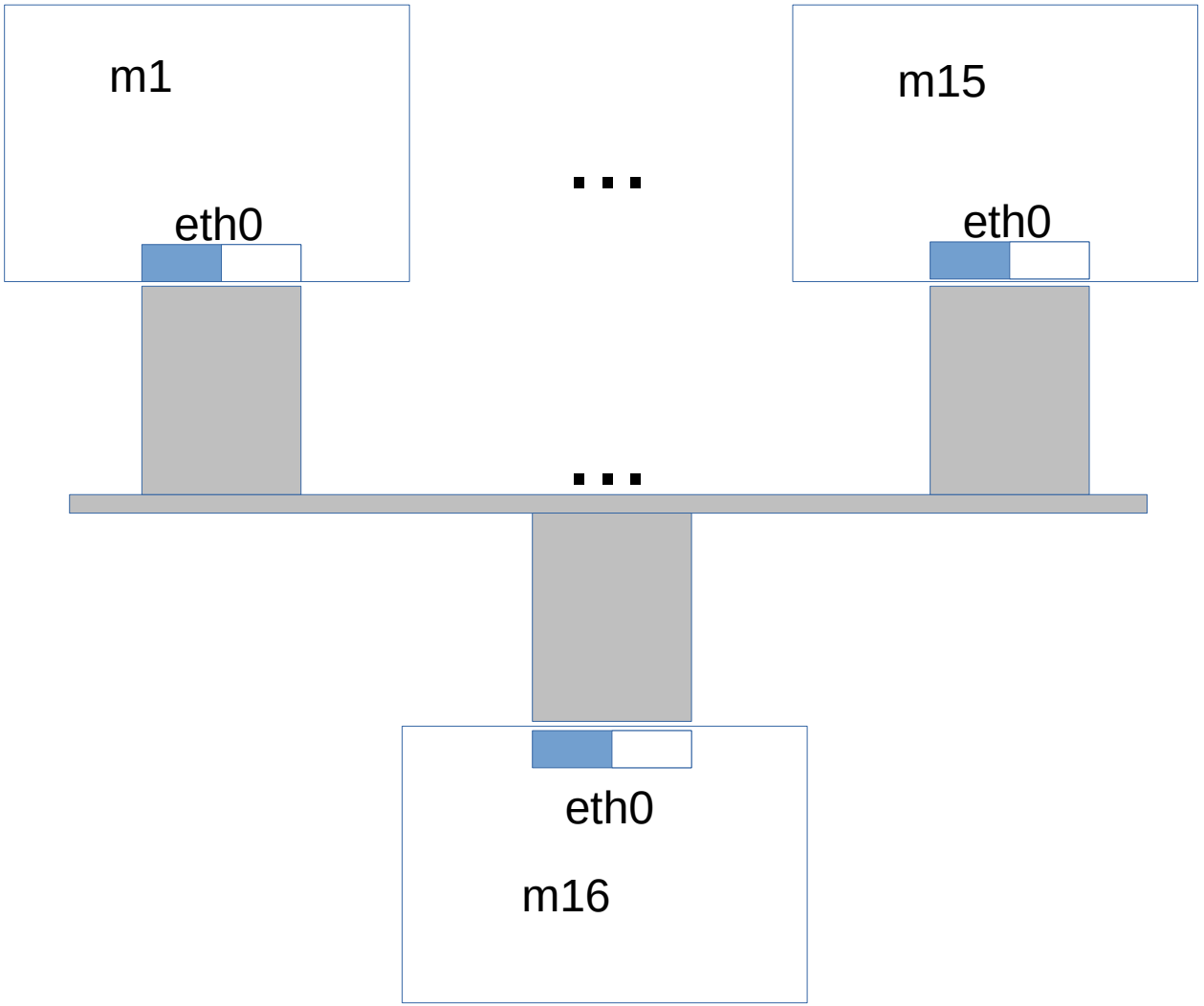
Câblage Ethernet (802.3)
DIFFERENCE switch VS hub
(« commutateur » VS « concentrateur »)

On appelle souvent « **couche liaison** » le regroupement des couches 1 (« physique ») et 2 (« liaison », au sens strict) de la pile TCP/IP constituée de 5 étages (1=physique, 2=liaison, 3=IP, 4=TCP-UDP-SCTP, 5=Applications). On évitera ce petit abus de langage en appelant l'ensemble 1 et 2 « couche hyper-arête » au sens où il s'agit de la couche de l'OS permettant la connexion du nœud IP à l'hyper-arête (ici de type Ethernet)

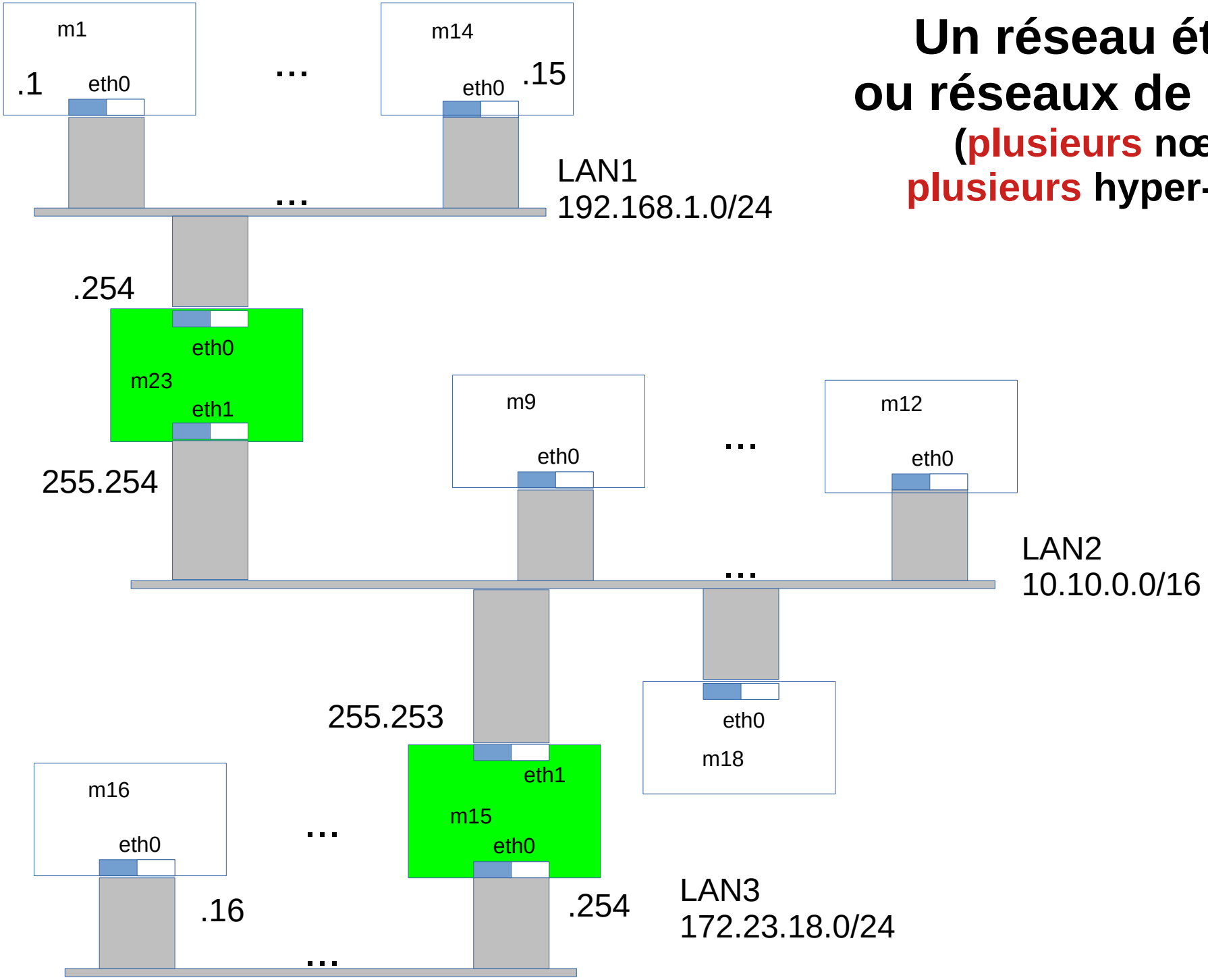
Le **HUB** ou « concentrateur » ou « répéteur » est un appareil de la couche physique 1, qui **répète** chaque trame lue sur un port, sur tous les autres ports

Le **SWITCH** ou « commutateur » est un appareil de niveau 2, plus intelligent, qui contient une table d'association (adresse-MAC → port) grâce à laquelle il pourra adresser chaque trame reçue sur le bon port, sans déranger les interlocuteurs (nœuds) branchés sur les autres ports

Un réseau local
(plusieurs nœuds, une seule hyper-arête)



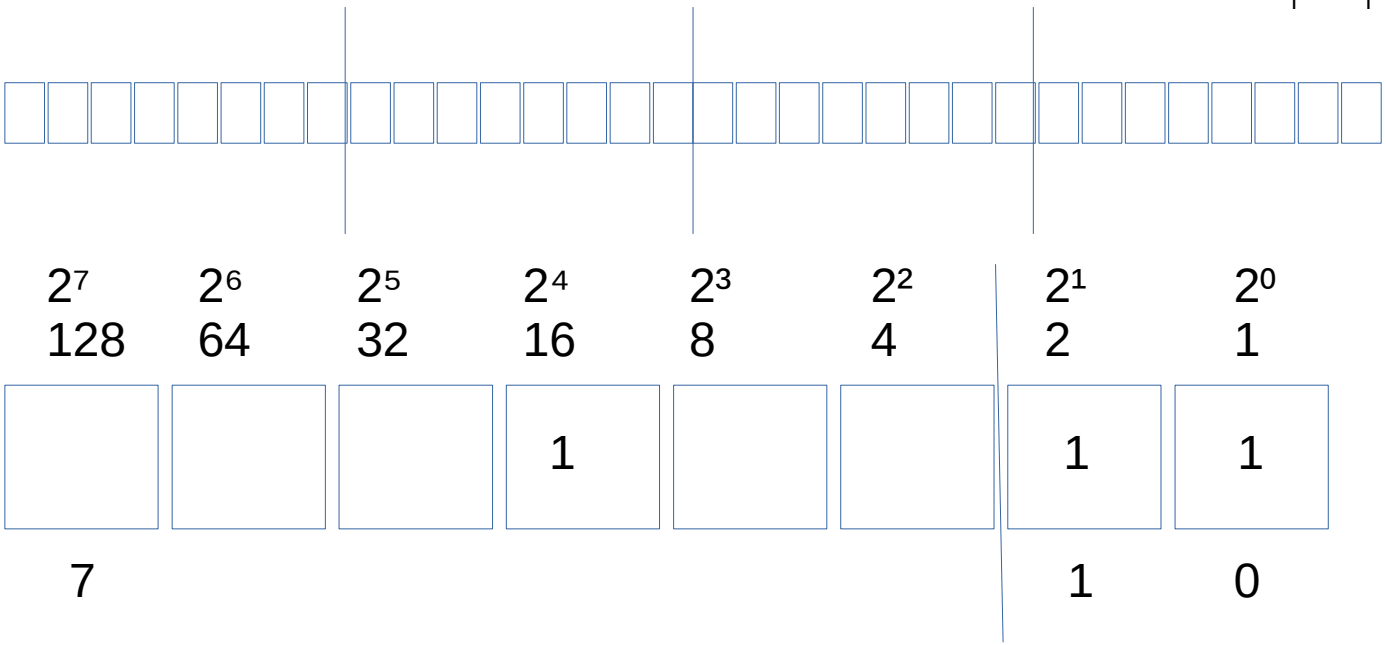
Un réseau étendu ou réseaux de réseaux (plusieurs nœuds, plusieurs hyper-arêtes)



Adresses IP v4 (4 octets = 32 bits)

Déf. Une **adresse IP (v4)** est un nombre naturel qu'on peut représenter avec 32 bits, c'est-à-dire 0..(2^32 – 1). Par **convention**, on représente cette valeur par 4 nombres décimaux (0..255) correspondants aux 4 octets constituant l'adresse, séparés par des points. Ex : **77.66.55.44**

$2^{32} = 2^{(10+10+10+2)} = 2^{10} * 2^{10} * 2^{10} * 2^{2} = 1024 * 1024 * 1024 * 4 = K_i * K_i * K_i * 4 = 4\ G_i$



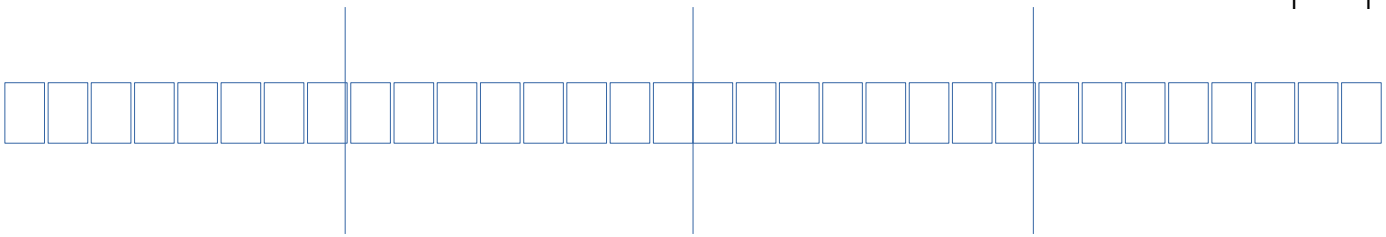
Déf. Une **adresse réseau** est la spécification d'un préfixe commun partagé par toutes les adresses IP d'un réseau local, c'est-à-dire une adresse IP et une longueur de préfixe (en nombre de bits) où les bits qui n'appartiennent pas au préfixe sont fixés à 0.
Par exemple :

172.23.18.0/24 => implicitement le segment 172.23.18.0 .. 172.23.18.255
172.23.18.98/24 incorrect
172.23.19.0/24 correct => implicitement le segment 172.23.19.0 .. 172.23.19.255
172.23.19.0/22 incorrect
172.23.16.0/22 correct => implicitement le segment 172.23.16.0 .. 172.23.19.255
0.0.0.0/0 correct => implicitement toutes les adresses IPv4 : 0.0.0.0 .. 255.255.255.255

Adresses IP v4 (4 octets = 32 bits)

Déf. Une **adresse IP (v4)** est un nombre naturel qu'on peut représenter avec 32 bits, c'est-à-dire 0..(2^32 – 1). Par **convention**, on représente cette valeur par 4 nombres décimaux (0..255) correspondants aux 4 octets constituant l'adresse, séparés par des points. Ex : **77.66.55.44**

$2^{32} = 2^{(10+10+10+2)} = 2^{10} * 2^{10} * 2^{10} * 2^{2} = 1024 * 1024 * 1024 * 4 = K_i * K_i * K_i * 4 = 4\ G_i$

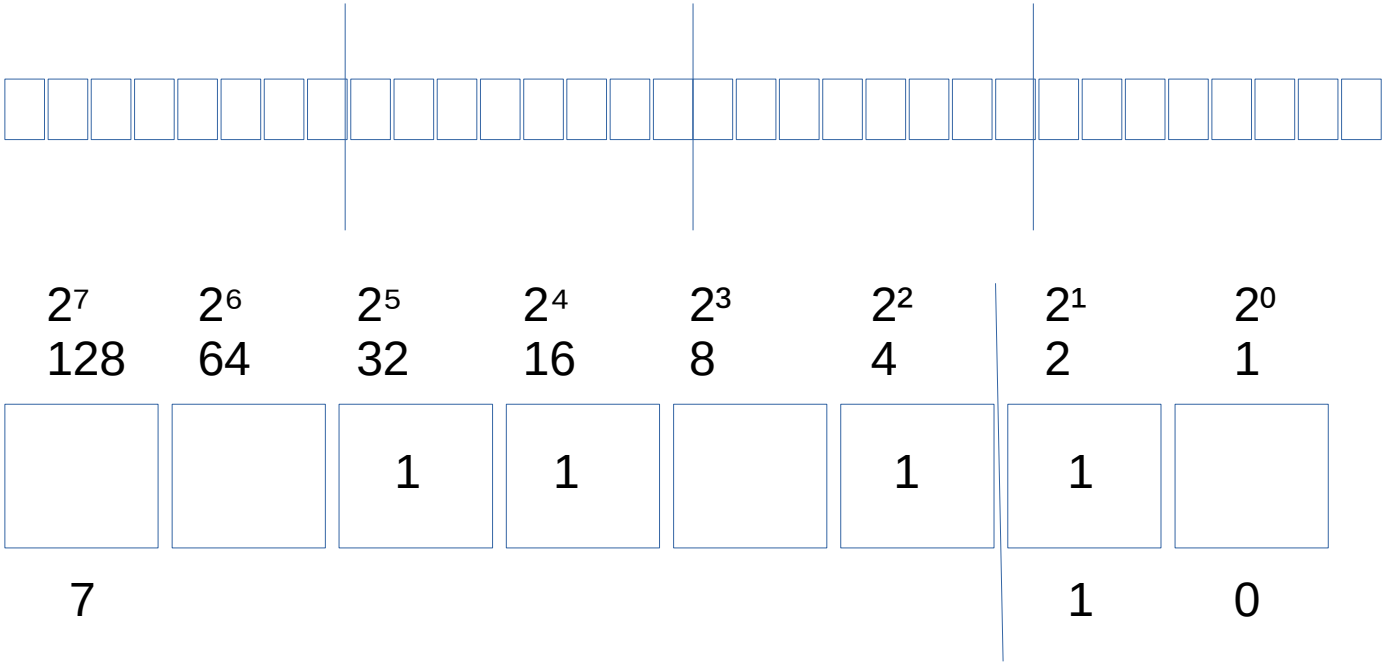


Déf. Une **adresse de diffusion** est la spécification d'un préfixe commun partagé par toutes les adresses IP d'un réseau local, c'est-à-dire une adresse IP et une longueur de préfixe (en nombre de bits) où les bits qui n'appartiennent pas au préfixe sont fixés à 1.
Par exemple :
172.23.18.255/24
172.23.18.98/24 incorrect
172.23.19.255/24 correct
172.23.19.255/22 correct

Déf. Un **masque réseau (netmask)** est une configuration de 32 bits où la partie préfixe est à 1 et la partie non-préfixe (hôte) est à 0, qui permet de spécifier (autrement que /..) la longueur du préfixe en nombre de bits.
Par exemple :
255.255.255.0 (équivalent à dire /24)
255.255.252.0 (équivalent à dire /22)

Def. Une **adresse IP (v4)** est un nombre naturel qu'on peut représenter avec 32 bits, c'est-à-dire 0..(2^32 – 1)

2^32 = 2^(10+10+10+2) = 2^10 * 2^10 * 2^10 * 2^2 = 1024 * 1024 * 1024 * 4 = Ki * Ki * Ki * 4 = 4 Gi



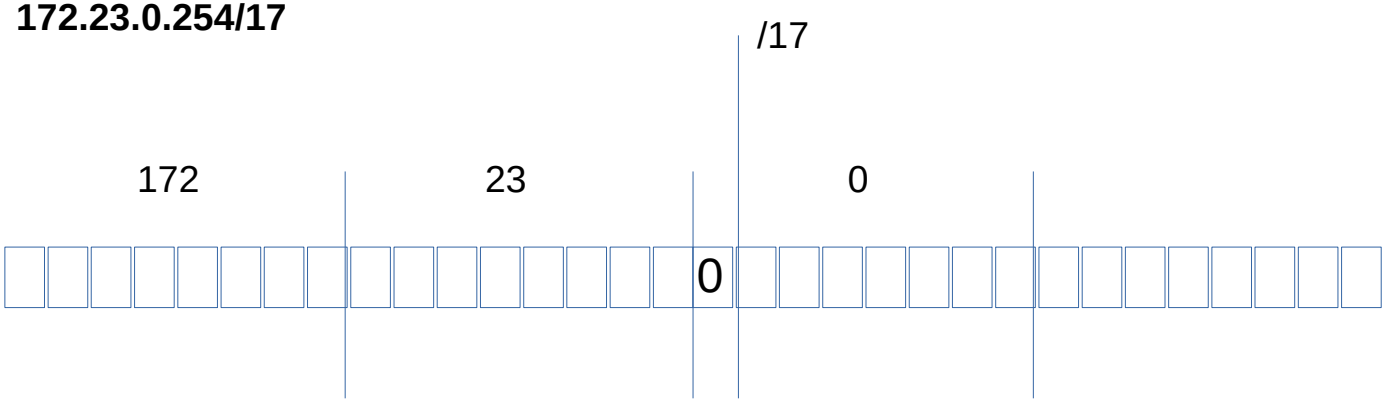
Déf. (**Capacité maximale d'un réseau**, c'est-à-dire nombre d'hôtes maxi)

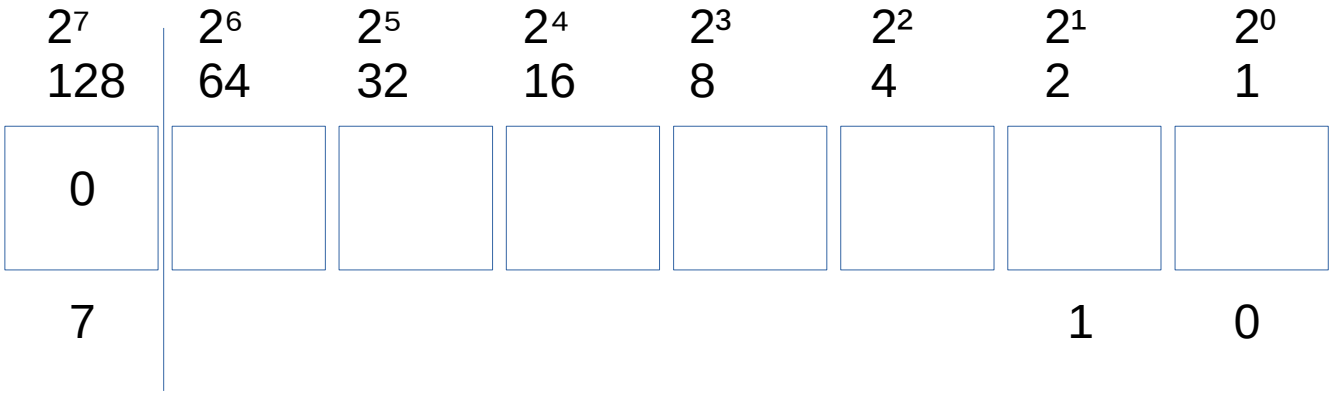
Étant donné une longueur de préfixe **k** (par exemple /24 ou 255.255.255.0, donc k=24), la **capacité maximale du réseau**, c'est-à-dire le nombre d'adresses IP disponibles est : **$2^{(32-k)}$ - 2**

Par exemple :
/19 (k=19) => $2^{(32-19)}$ - 2 = 2^{13} -2 ~ $2^{(10+3)}$ ~ 2^{10} * 2^3 = 8Ki

Correction du TD

Par exemple, découper :
172.23.0.254/17

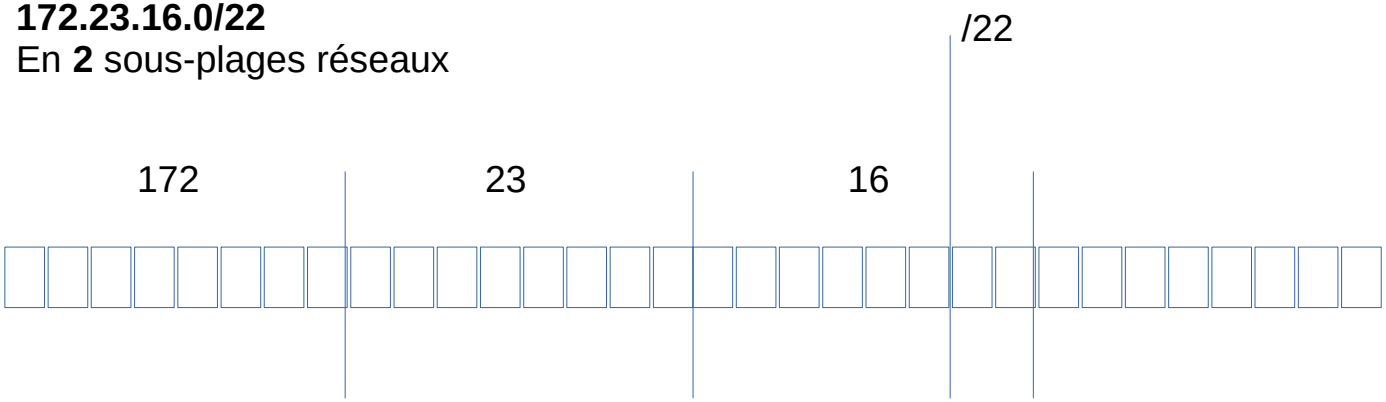


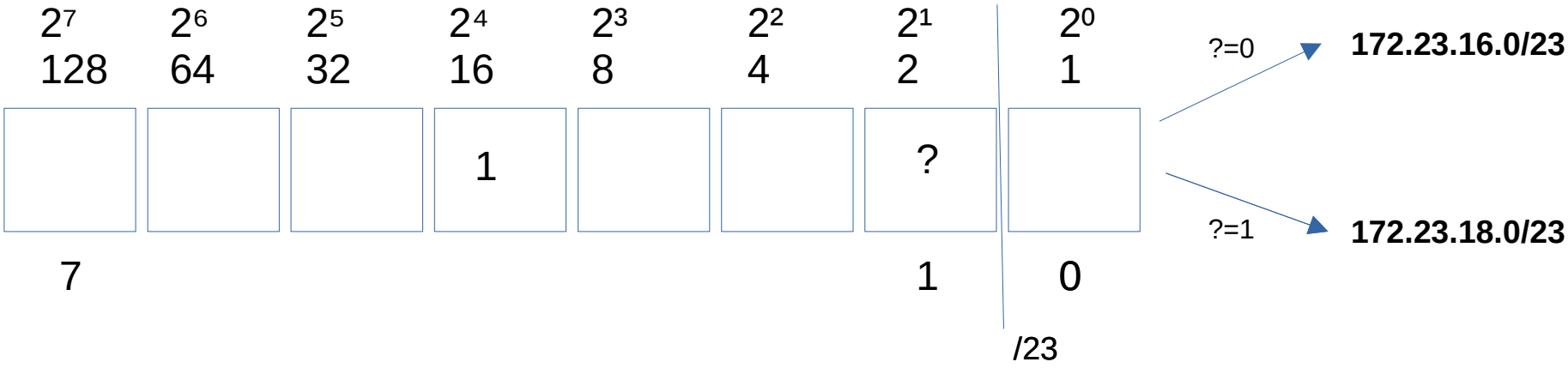


Adresse réseau : **172.23.0.0/17**
première adresse : **172.23.0.1**
dernière adresse : **172.23.127.254**
Adresse diffusion : **172.23.127.255/17**
Nb Adresses : **2^(32-17) = 2^(15) = 2^(10+5) = 2^10*2^5 = 32Ki**

Découper une plage réseau (Ex. 1)

Par exemple, découper :
172.23.16.0/22
En **2** sous-plages réseaux



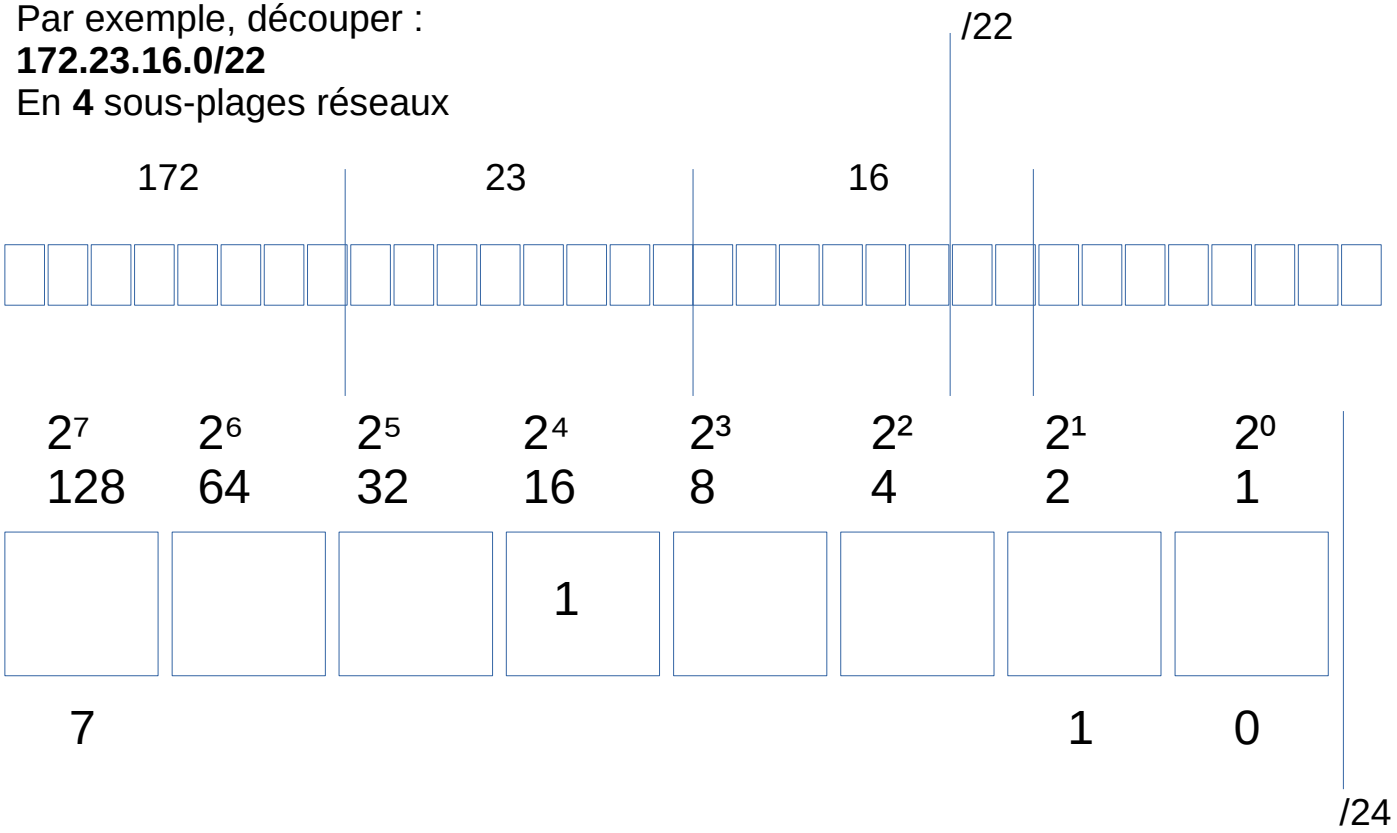


Si on fixe ? à **0** la case de poids 2 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.16.0/23**

Si on fixe ? à **1** la case de poids 2 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.18.0/23**

Découper une plage réseau (Ex. 2)

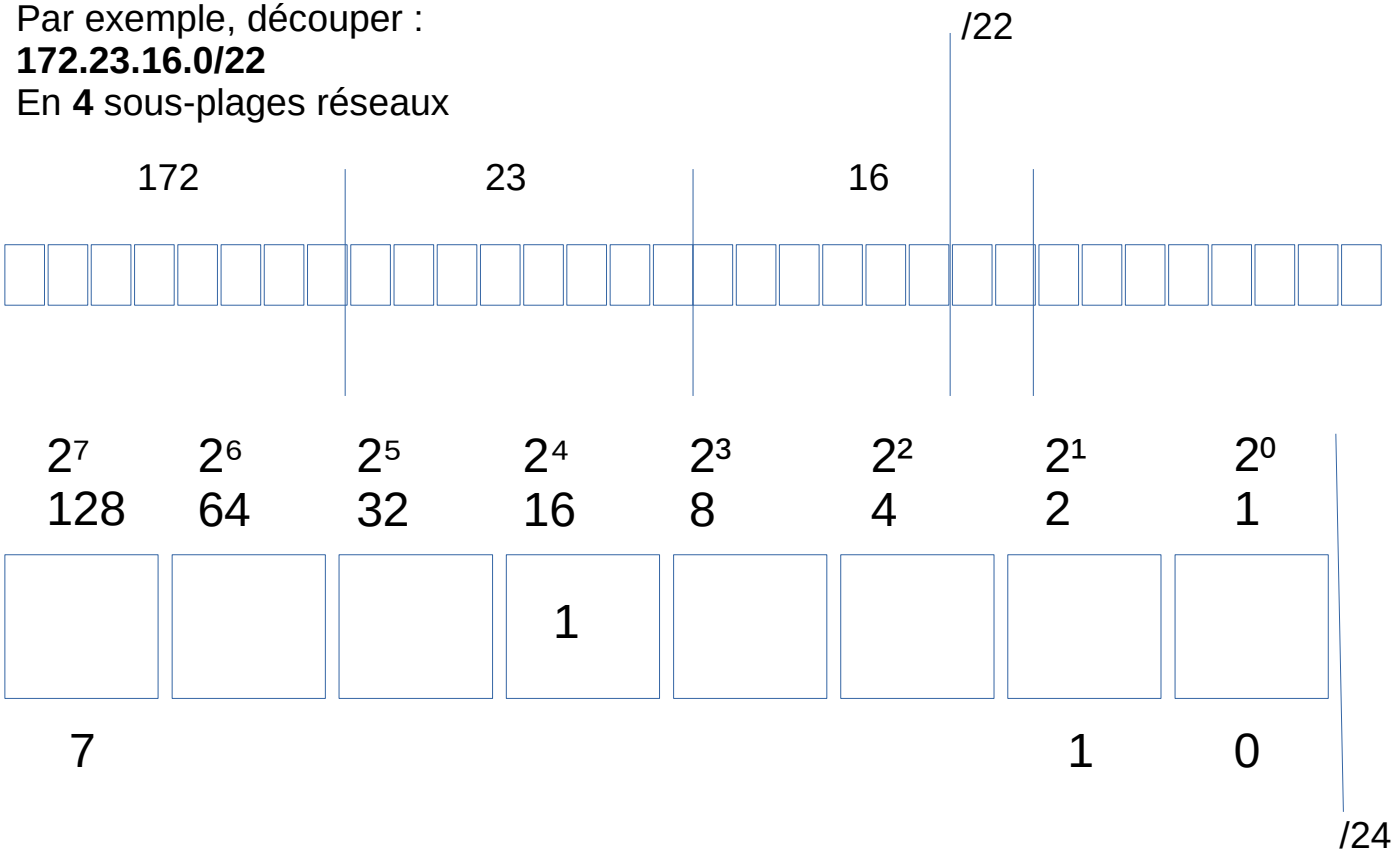
Par exemple, découper :
172.23.16.0/22
En **4** sous-plages réseaux



- Si on fixe à **00** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.16.0/24**
- Si on fixe à **01** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.17.0/24**
- Si on fixe à **10** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.18.0/24**
- Si on fixe à **11** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.19.0/24**

Découper une plage réseau (Ex. 2)

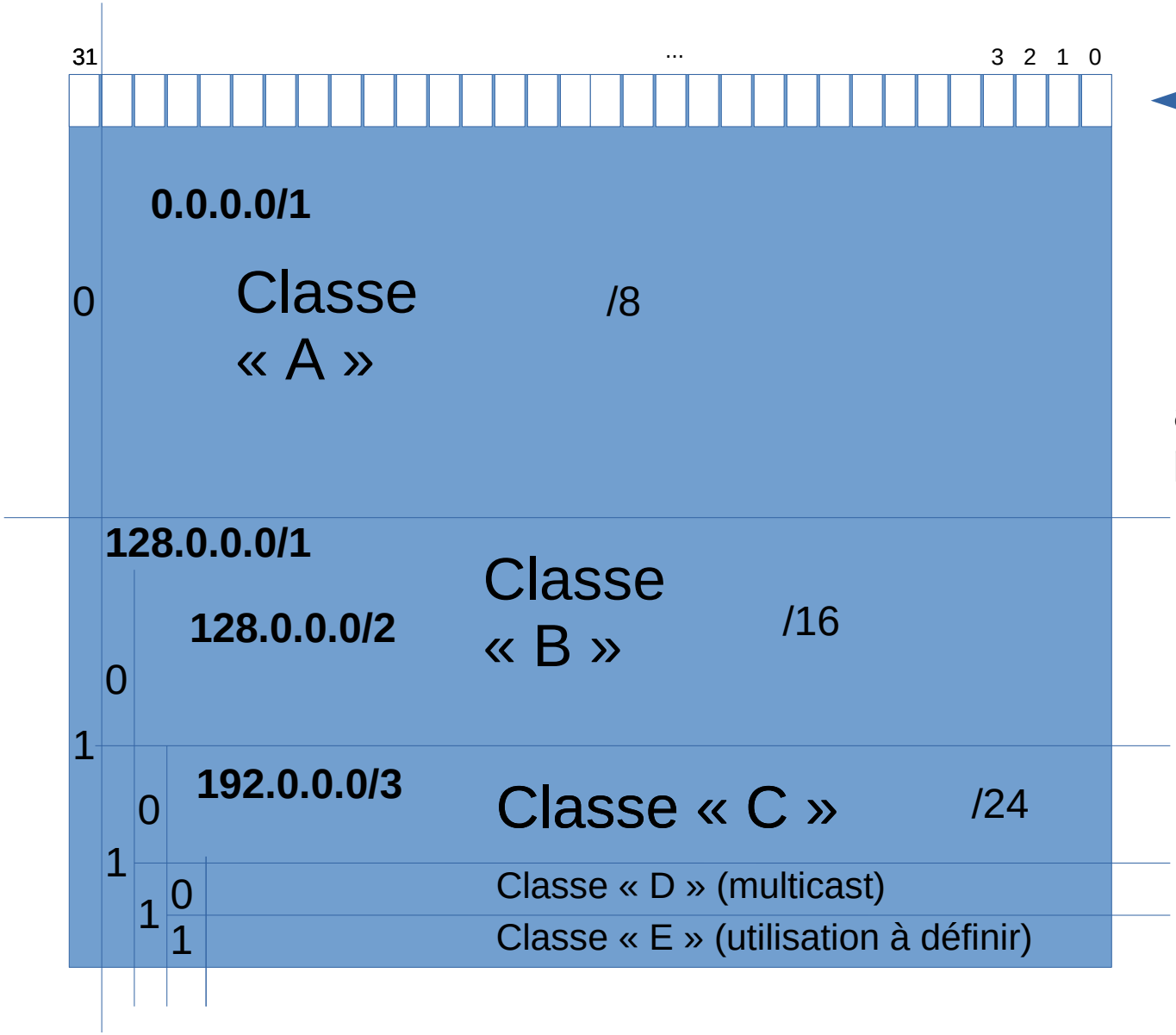
Par exemple, découper :
172.23.16.0/22
En **4** sous-plages réseaux



- Si on fixe à **00** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.16.0/24**
- Si on fixe à **01** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.17.0/24**
- Si on fixe à **10** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.18.0/24**
- Si on fixe à **11** les cases de poids 2 et 1 du 3ème octet, on obtient la nouvelle plage et adresse réseau **172.23.19.0/24**

Remarque :

en ajoutant au préfixe **une case supplémentaire**
on découpe toujours exactement **à moitié** l'ensemble des adresses disponibles



32 cases binaires (bits)

Ensemble de **toutes** les adresses IPv4, représentable par l'adresse réseau :

0.0.0.0/0

Plan des séances du module

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)
2) En-tête IP (chapitre 4 du livre de Laissus)

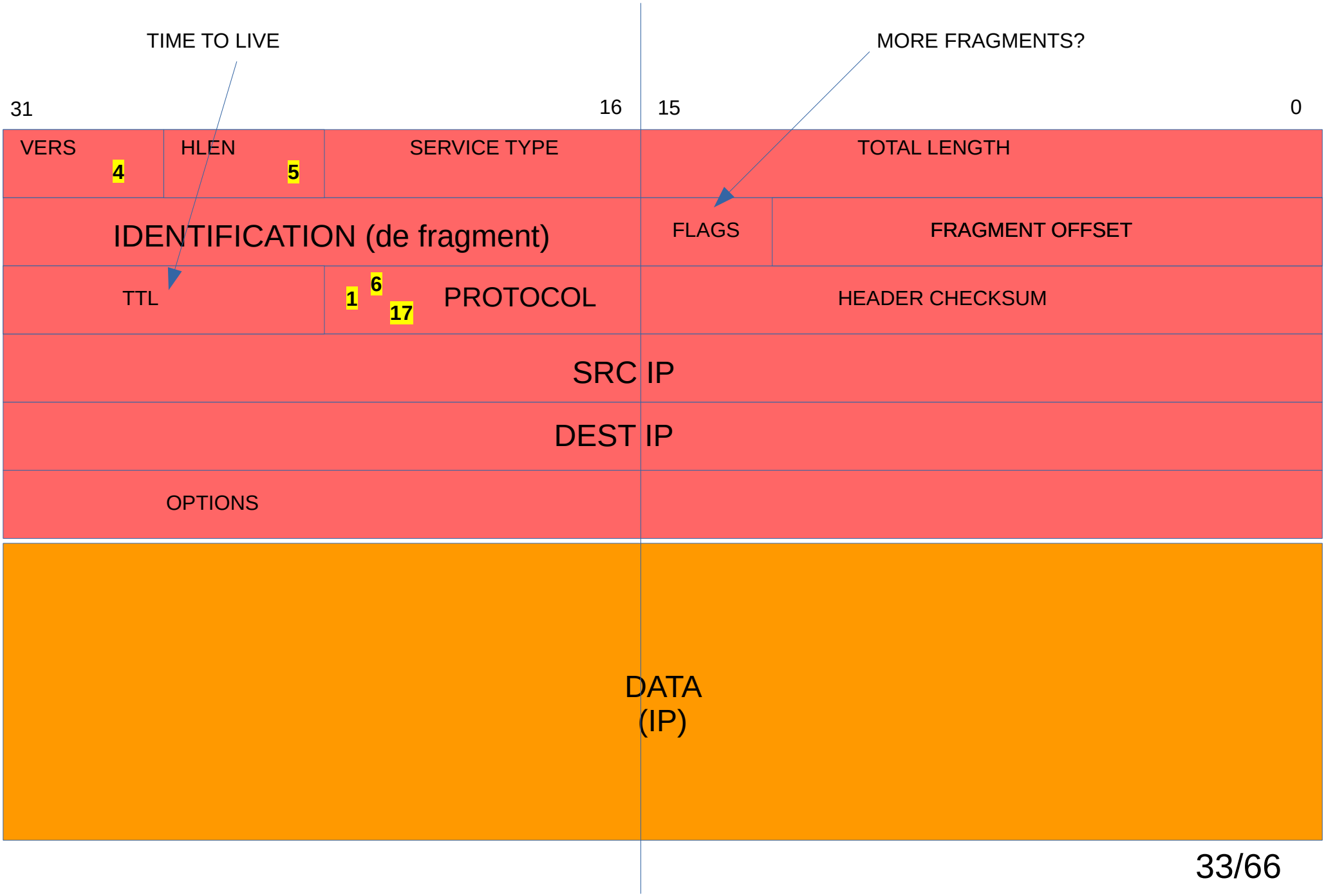
Cours n°4
1) routage (statique) IP
2) ICMP

Cours n°5
1) UDP (chapitre 5 du livre de Laissus)
2) TCP (chapitre 6)

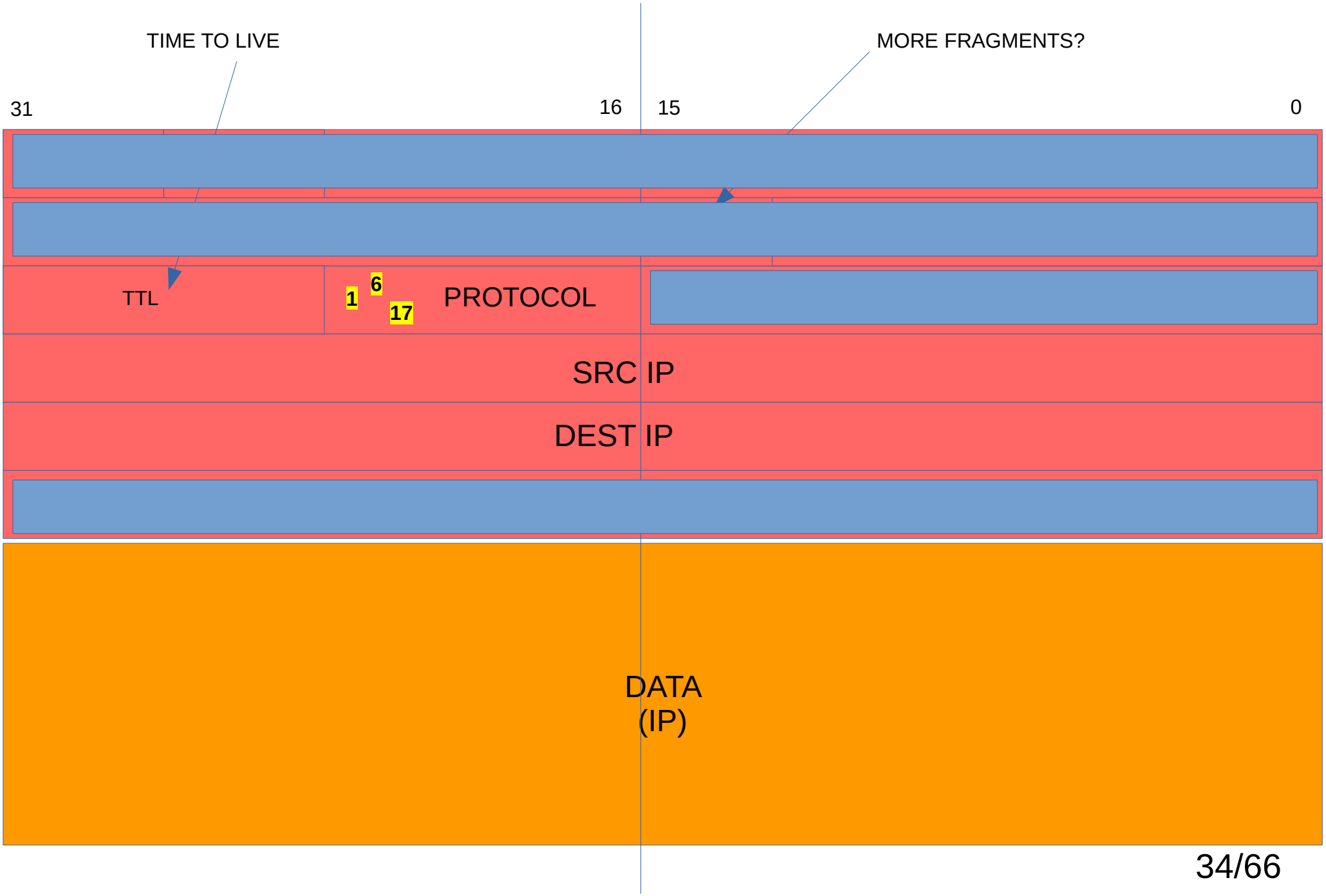
Plan du cours n°6
1) SNAT (Source NAT)
2) DNAT (Destination NAT)

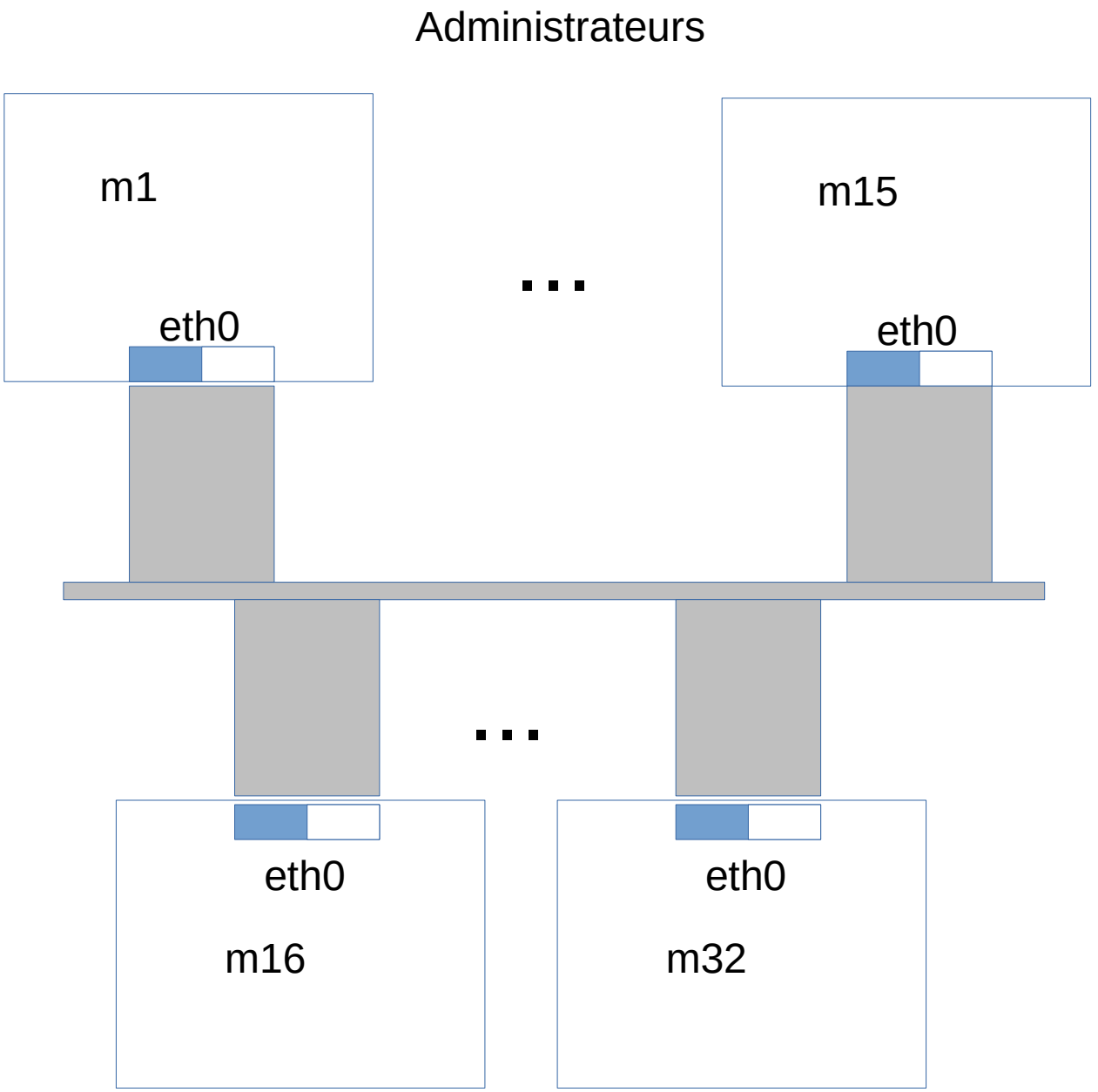
Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

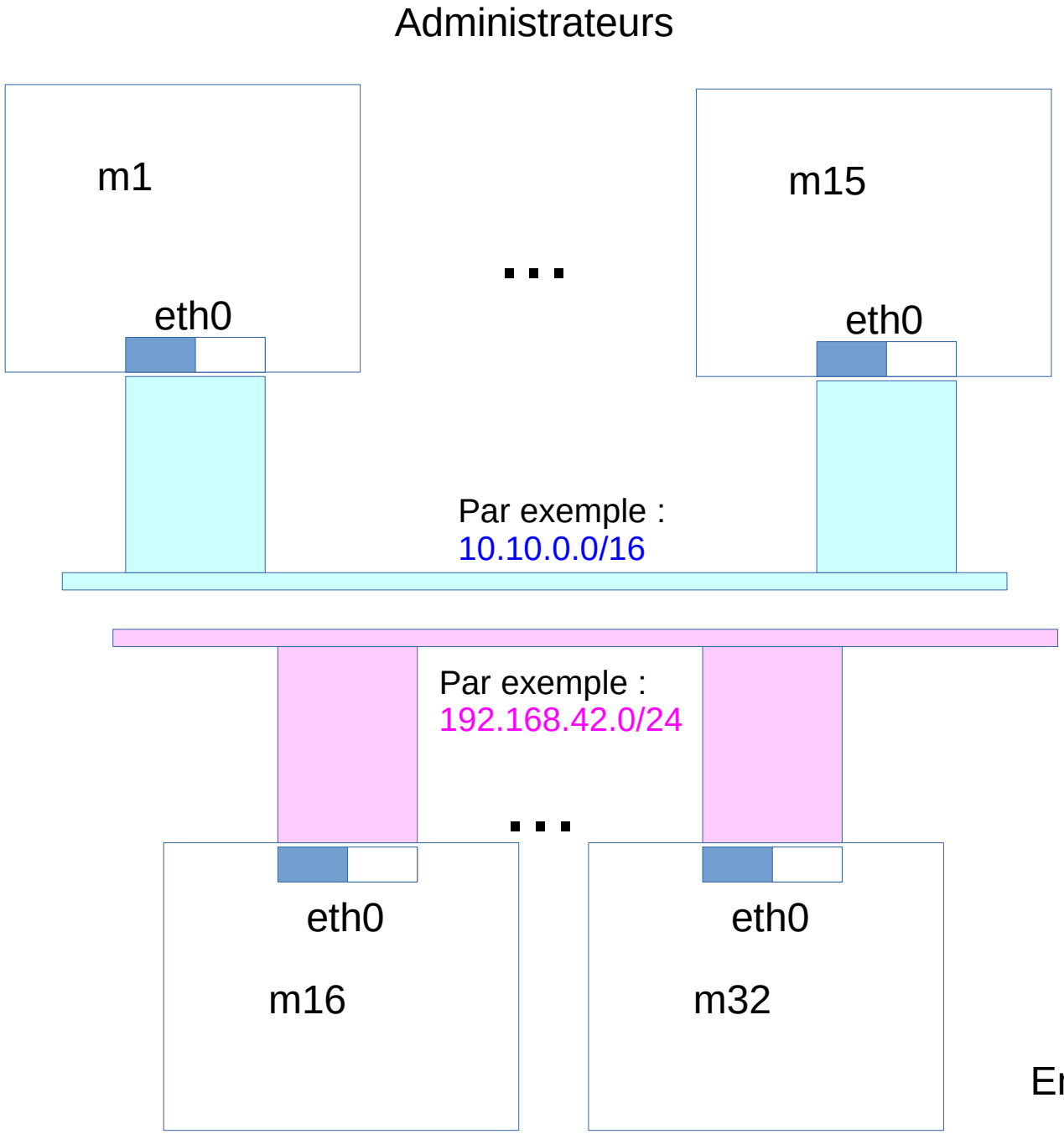
Format (en-tête) des « paquets » IP v4



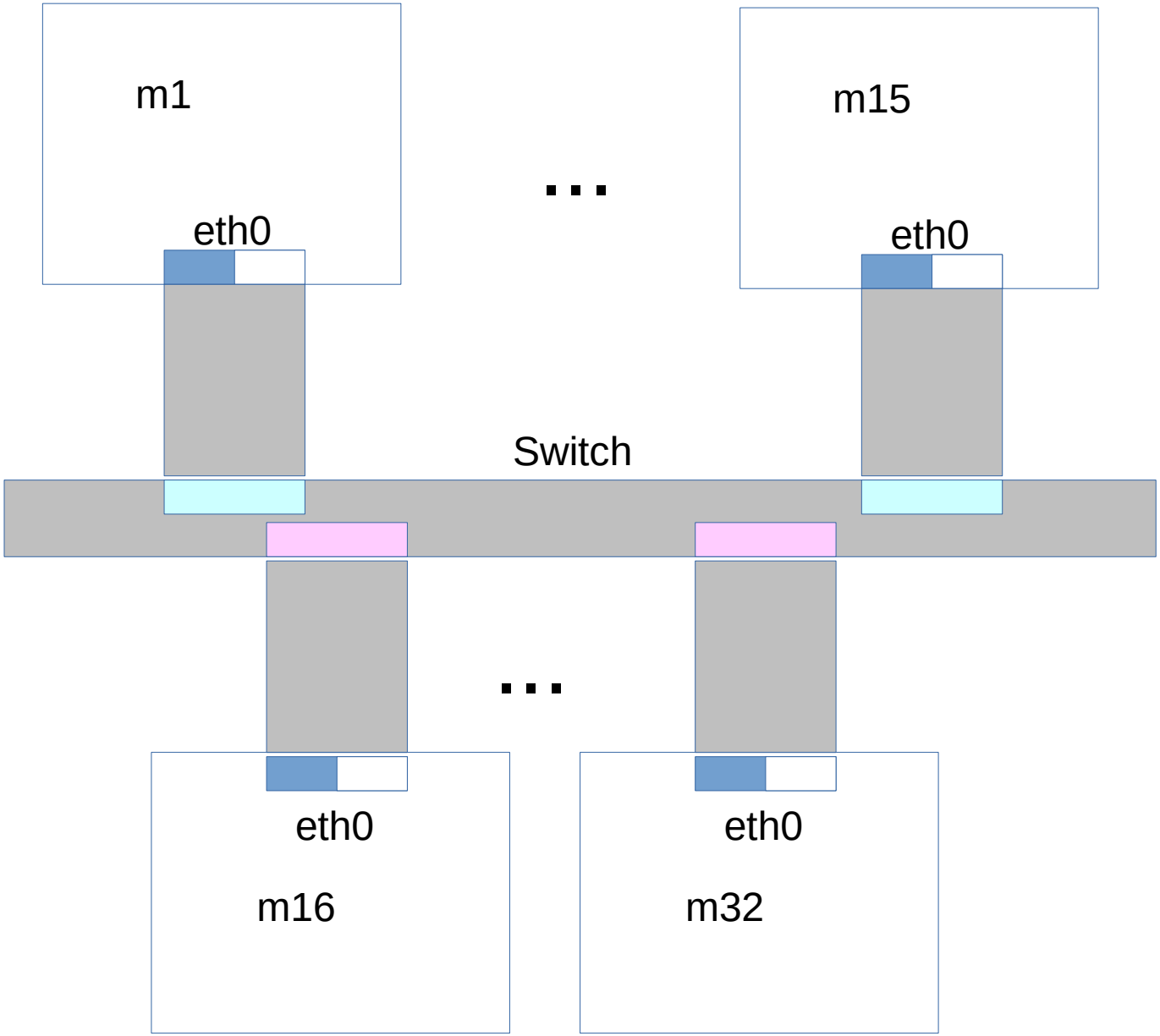
Format (en-tête) des « paquets » IP v4




Administrateurs
m1
m15
eth0
eth0
eth0
eth0
m16
m32
Employés


Administrateurs
m1
m15
eth0
eth0
Par exemple :
10.10.0.0/16
Par exemple :
192.168.42.0/24
eth0
eth0
m16
m32
Employés

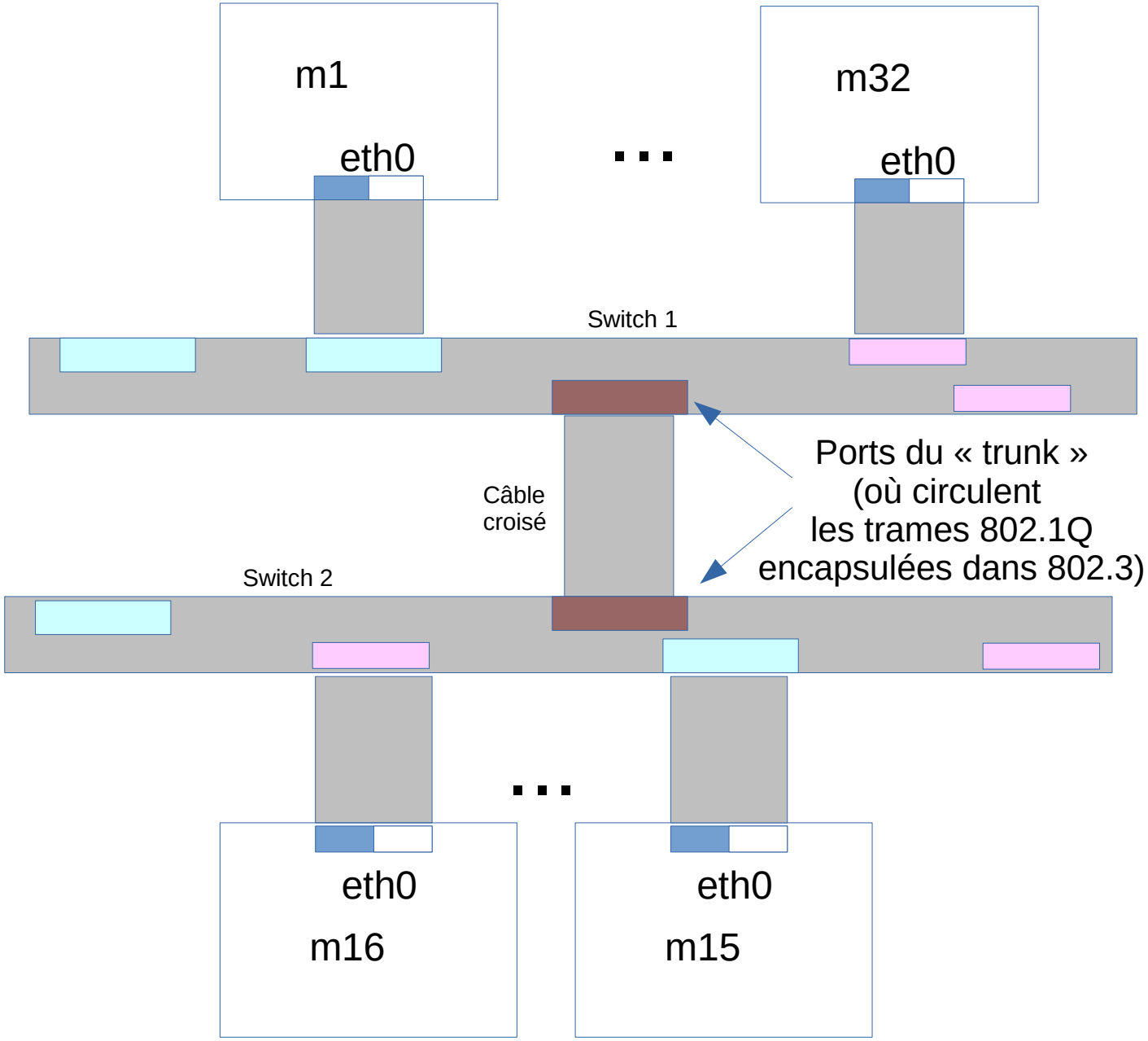
VLAN de type 1 (basées sur les ports)
(1) CAS : 1 seul commutateur (switch) => FACILE



Administrateurs

Employés

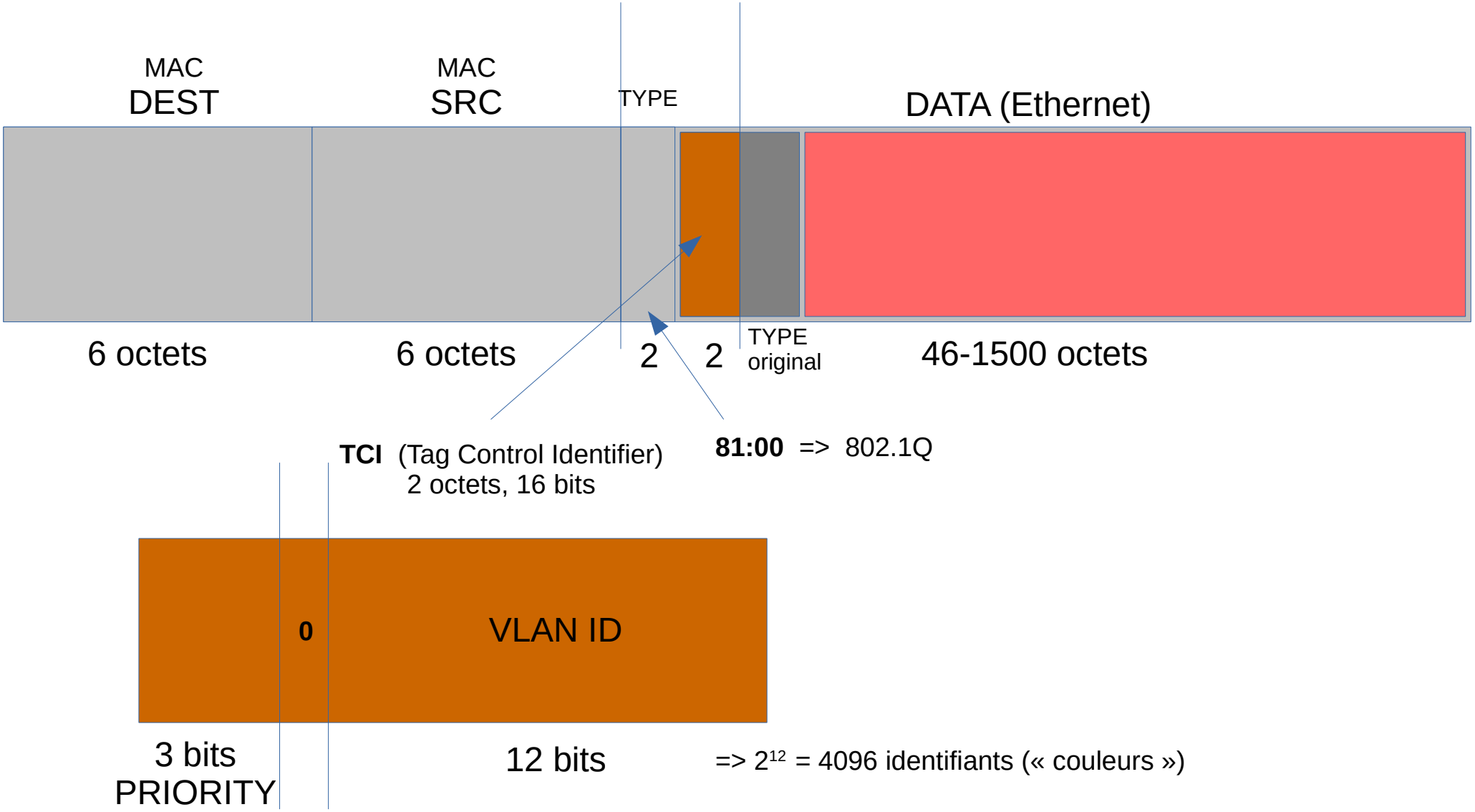
VLAN de type 1
(2) CAS : plusieurs commutateurs (switch) => 802.1Q



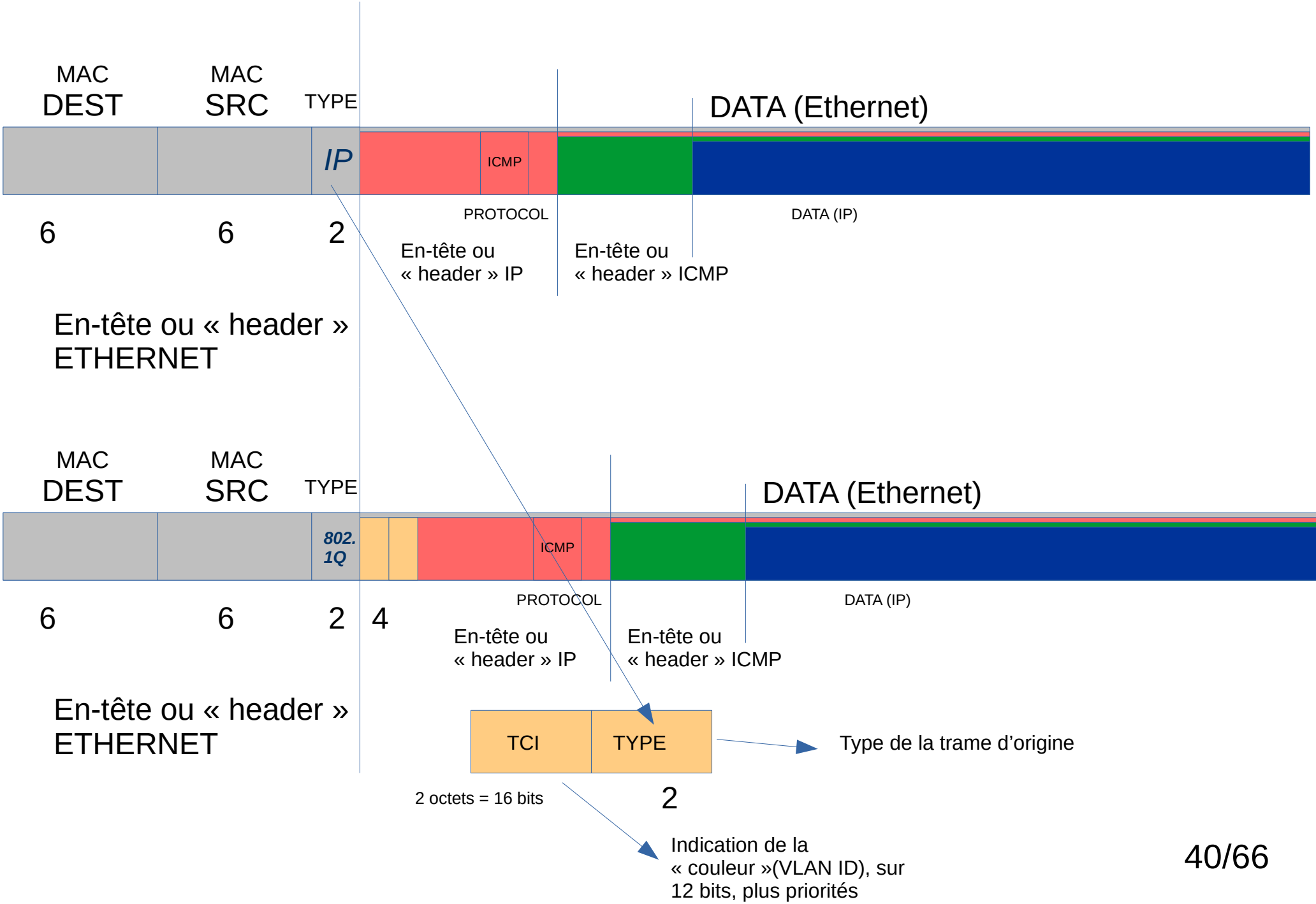
Administrateurs

Employés

« trame » ETHERNET 802.3 qui véhicule une trame 802.1Q

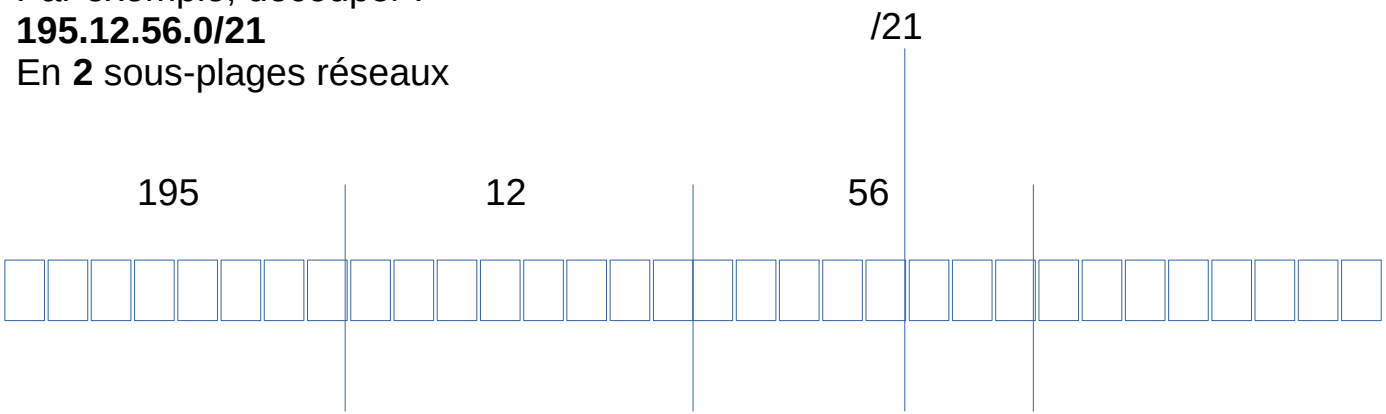


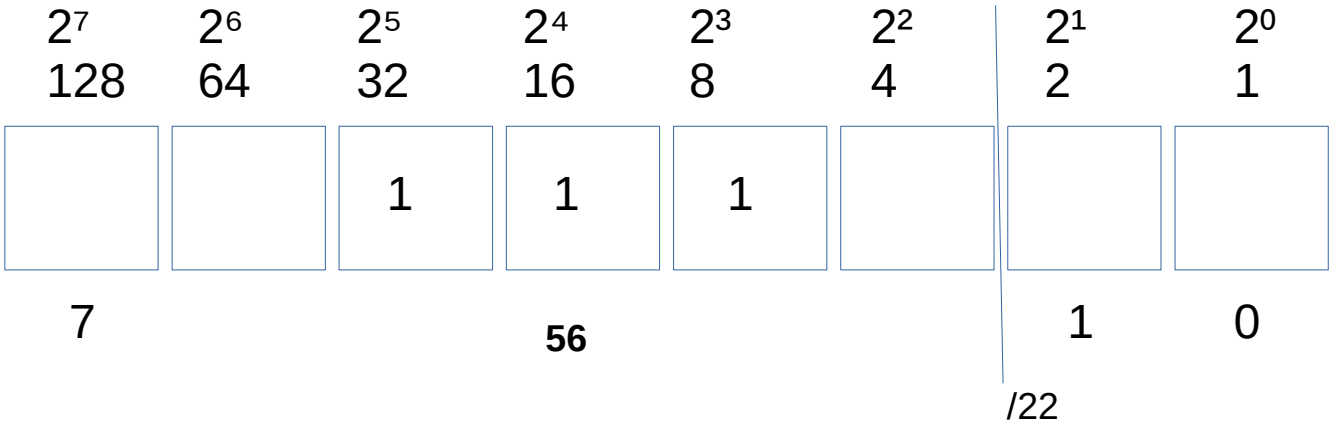
« trame » ETHERNET 802.3



Découper une plage réseau (Ex. du TP)

Par exemple, découper :
195.12.56.0/21
En **2** sous-plages réseaux



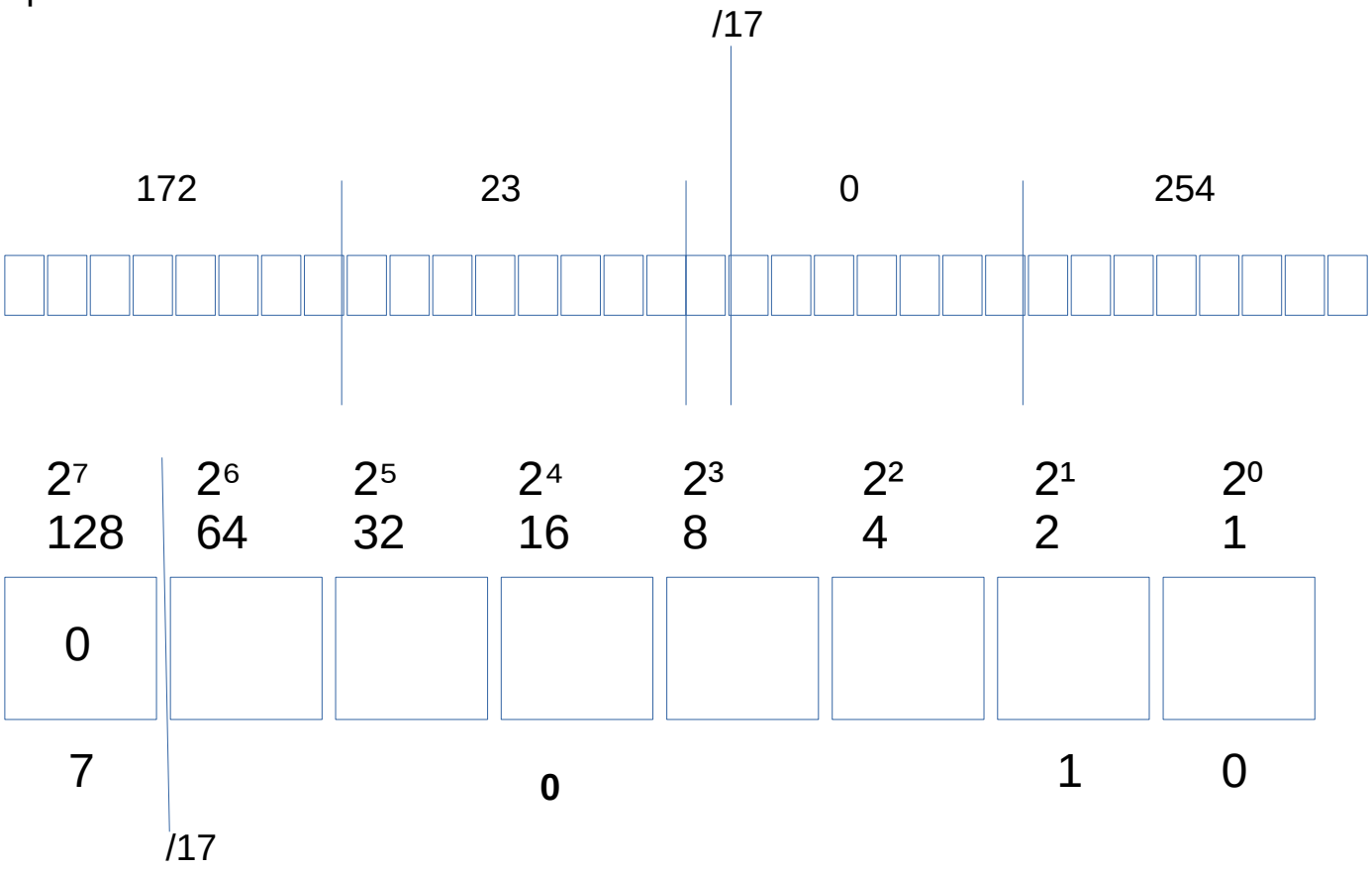


Si on fixe à 0 la case de poids 4 du 3ème octet, on obtient la nouvelle plage et adresse réseau 195.12.56.0/22

Si on fixe à 1 la case de poids 4 du 3ème octet, on obtient la nouvelle plage et adresse réseau 195.12.60.0/22

Correction TD

Ipcalc de **172.23.0.254/17**



Adresse réseau 172.23.0.0/17 (c'est-à-dire]172.23.0.0, 172.23.127.255[ )
Adresse de diffusion 172.23.127.255/17
Netmask 255.255.128.0 (c'est-à-dire /17)
Hostmin : 172.23.0.1
Hostmax : 172.23.127.254
Hosts : $2^{(32-17)} - 2 = 2^{15} - 2 = 2^{10+5} - 2 = 2^{10} * 2^5 - 2 = 32Ki - 2 \sim 32Ki$

Plan des séances du module

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) En-tête IP (chapitre 4 du livre de Laissus)
2) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)

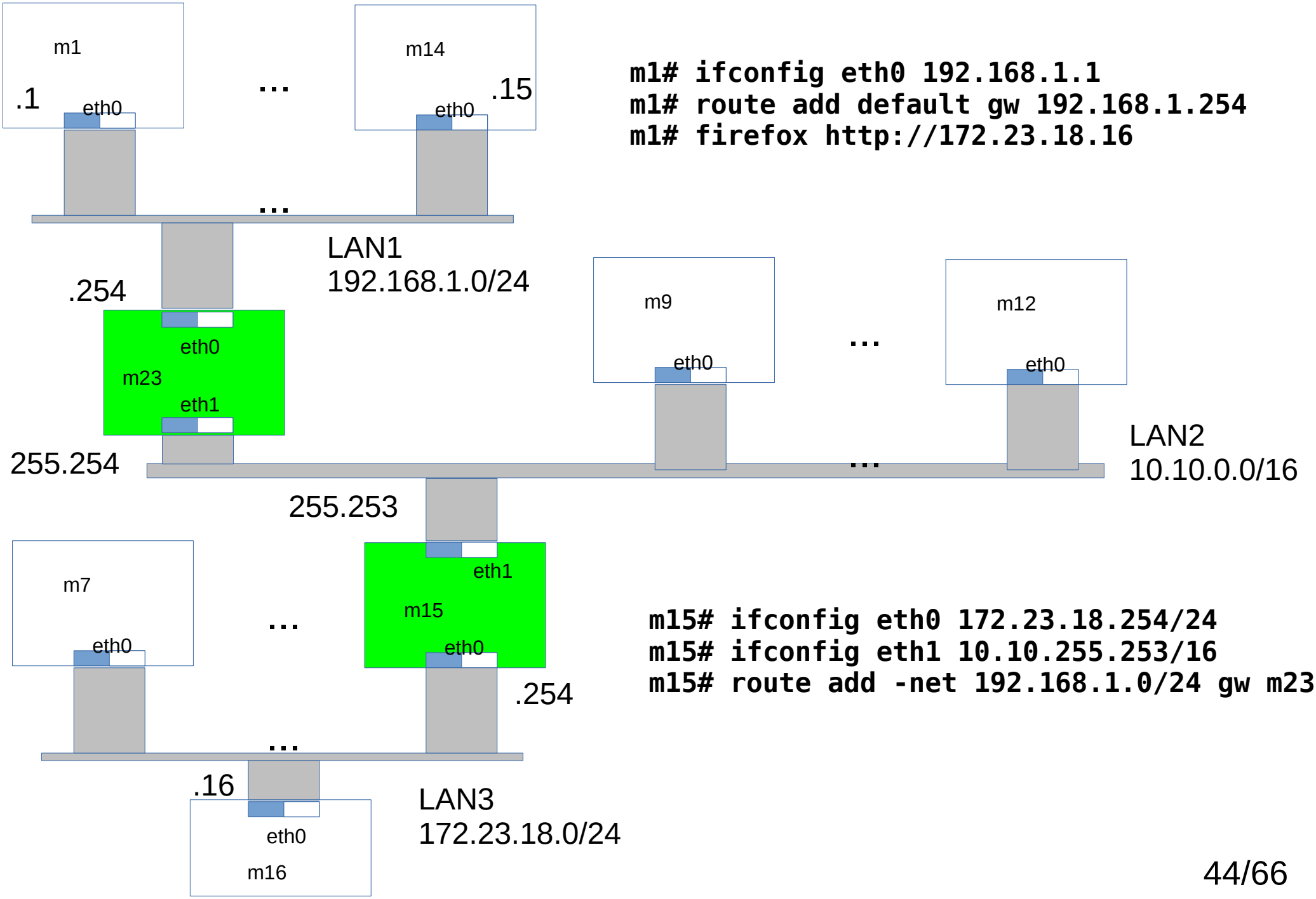
Cours n°4
1) routage (statique) IP
2) ICMP

Cours n°5
1) UDP (chapitre 5 du livre de Laissus)
2) TCP (chapitre 6)

Plan du cours n°6
1) SNAT (Source NAT)
2) DNAT (Destination NAT)

Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

Routage statique IP (v4)




```
m1# ifconfig eth0 192.168.1.1
m1# route add default gw 192.168.1.254
m1# firefox http://172.23.18.16
```

```
m15# ifconfig eth0 172.23.18.254/24
m15# ifconfig eth1 10.10.255.253/16
m15# route add -net 192.168.1.0/24 gw m23
```

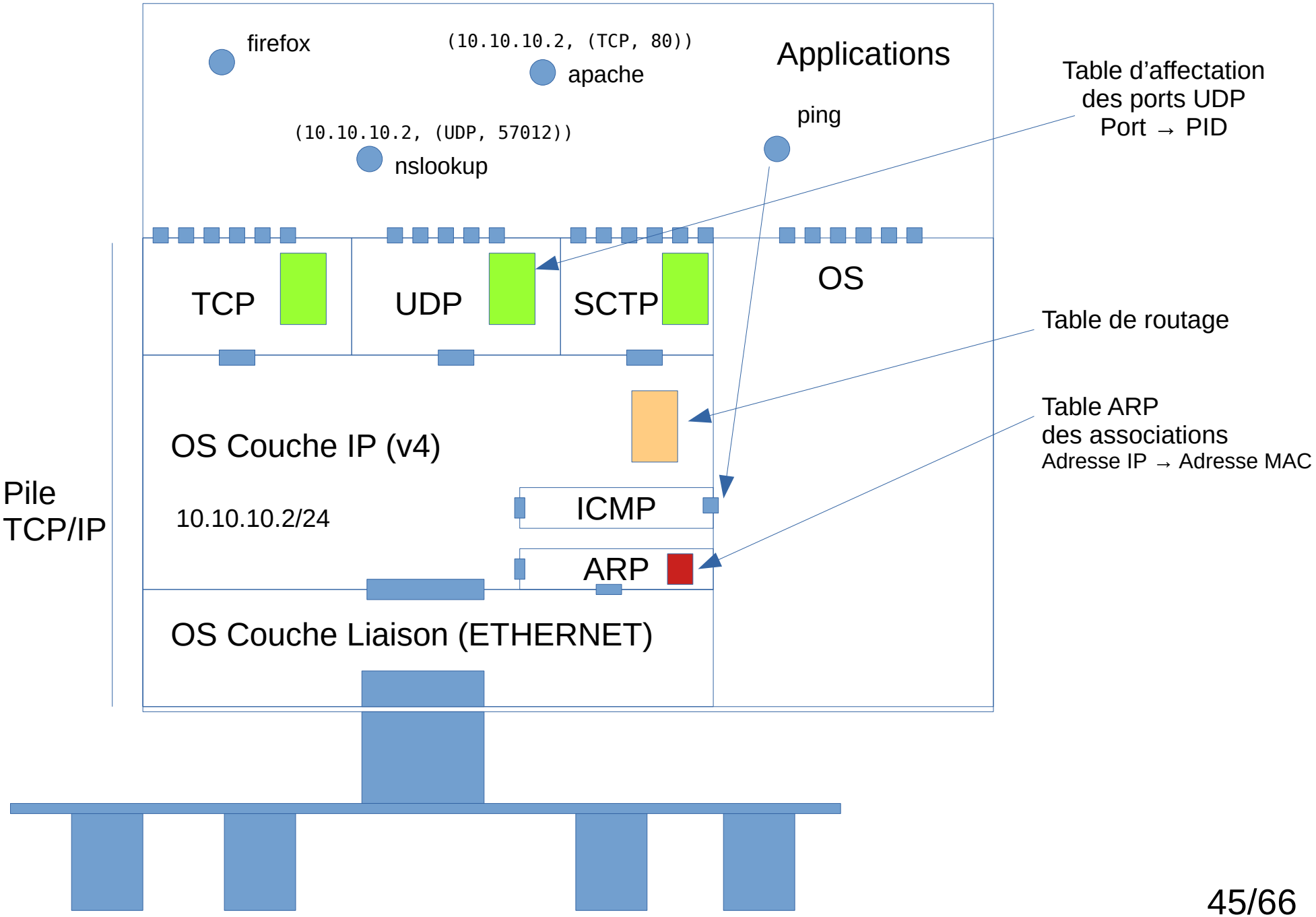
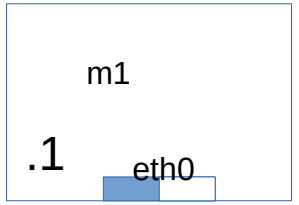

firefox
(10.10.10.2, (TCP, 80))
apache
Applications
ping
(10.10.10.2, (UDP, 57012))
nslookup
Table d'affectation
des ports UDP
Port → PID
TCP
UDP
SCTP
OS
Table de routage
Table ARP
des associations
Adresse IP → Adresse MAC
OS Couche IP (v4)
Pile
TCP/IP
10.10.10.2/24
ICMP
ARP
OS Couche Liaison (ETHERNET)

Table de routage

Adresse réseau de destination	Que faire avec ?	
	Passerelle ?	Interface
…	…	

Table de routage de m1



192.168.1.1/24

Que faire avec ?

Adresse réseau de destination	Passerelle ?	Interface
192.168.1.0/24	NON (direct)	eth0
0.0.0.0/0	192.168.1.254	eth0

Ordre de lecture

Sous GNU/Linux :

```
m1# ifconfig eth0 192.168.1.1/24
m1# route add –net 0.0.0.0/0 gw 192.168.1.254 dev eth0
```

Ou plus simplement :

```
m1# route add default gw 192.168.1.254
```

Table de routage de m23

Que faire avec ?

Adresse réseau de destination	Passerelle ?	Interface
192.168.1.0/24	NON (direct)	eth0
10.10.0.0/16	NON (direct)	eth1
0.0.0.0/0	10.10.255.253	eth1

Ordre de lecture

Sous GNU/Linux :

```
m23# ifconfig eth0 192.168.1.254/24
m23# ifconfig eth1 10.10.255.254/16
m23# route add –net 0.0.0.0/0 gw 10.10.255.253 dev eth1
```

Ou plus simplement :

```
m23# route add default gw 10.10.255.253
```

Rien à voir avec la table de routage, mais un routeur GNU/Linux doit avoir l'activité de routage activée par :

```
m23# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Messages ICMP principaux

TYPE	CODE	Signification (sens du message)
8	0	ECHO REQUEST (demande d'écho des données dans la partie DATA)
0	0	ECHO REPLY (réponse à la demande d'écho, avec données dans la partie DATA)
3	0	NETWORK UNREACHABLE
3	1	HOST UNREACHABLE
3	2	PROTOCOL UNREACHABLE
3	3	PORT UNREACHABLE
4	0	SOURCE QUENCH (trop de paquets à transmettre de la même source)
5	0-3	REDIRECT (il aurait fallu passer par une autre passerelle !)
11	0-1	TTL EXCEEDED (nombre de sauts dépassé)

Plan des séances du module

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)
2) En-tête IP (chapitre 4 du livre de Laissus)

Cours n°4
1) routage (statique) IP
2) ICMP

Cours n°5
1) UDP (chapitre 5 du livre de Laissus)
2) TCP (chapitre 6)

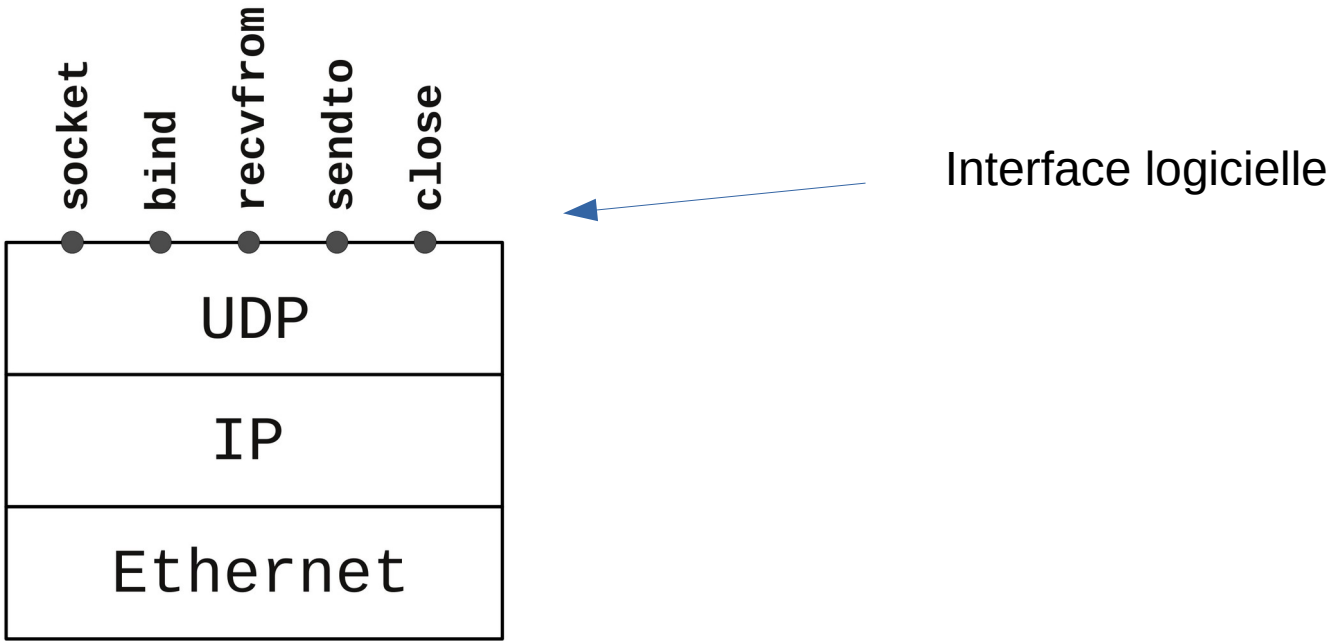
Plan du cours n°6
1) SNAT (Source NAT)
2) DNAT (Destination NAT)

Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

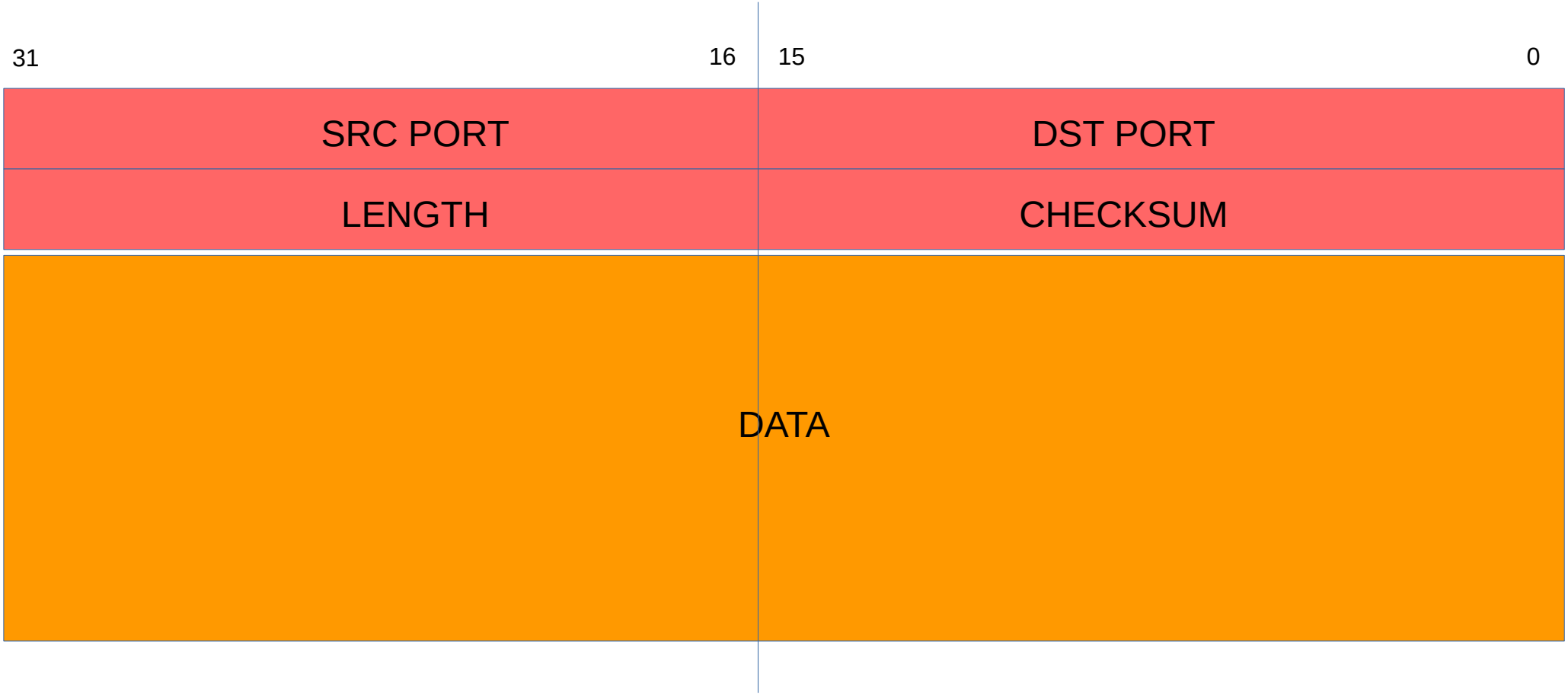
UDP = IP + ports + interface logicielle

Un **port** (UDP) est une sorte d'adresse numérique (sur 16 bits) qui peut être associée à un processus qui le demande (ou qui en demande un **quelconque**). Autrement dit, un port est une **sorte d'identité** (UDP) pour un processus sur le réseau.

Le **nombre de ports** (UDP) possibles et attribuables (à ses processus) est 2^16 = 2^10 * 2^6 = **64 Ki**



Format (en-tête) des « datagrammes » UDP



L’adresse du **processus expéditeur** est (IP-SRC, (UDP, PORT-SRC))

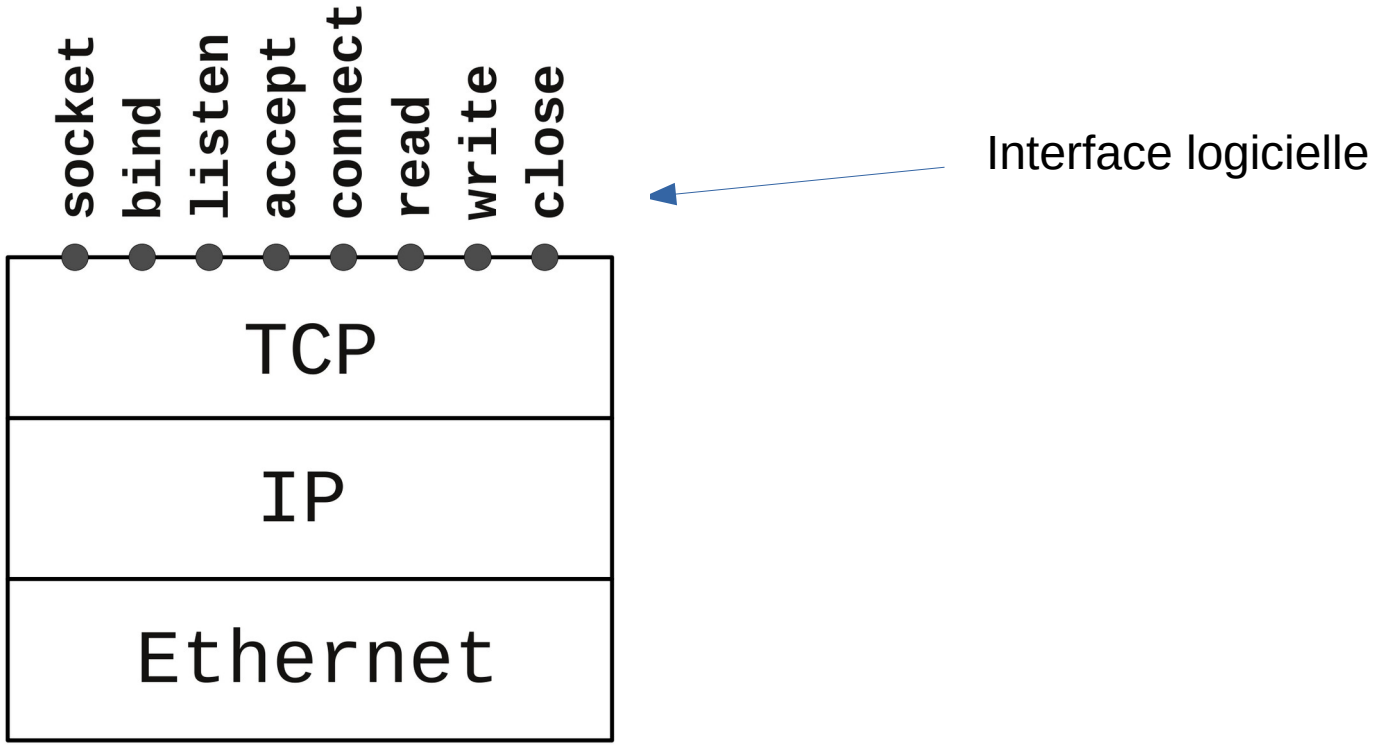
L’adresse du **processus destinataire** est (IP-DST, (UDP, PORT-DST))

Exemple d’adresse de **processus expéditeur** : **(101.102.103.104,(UDP, 61234))**

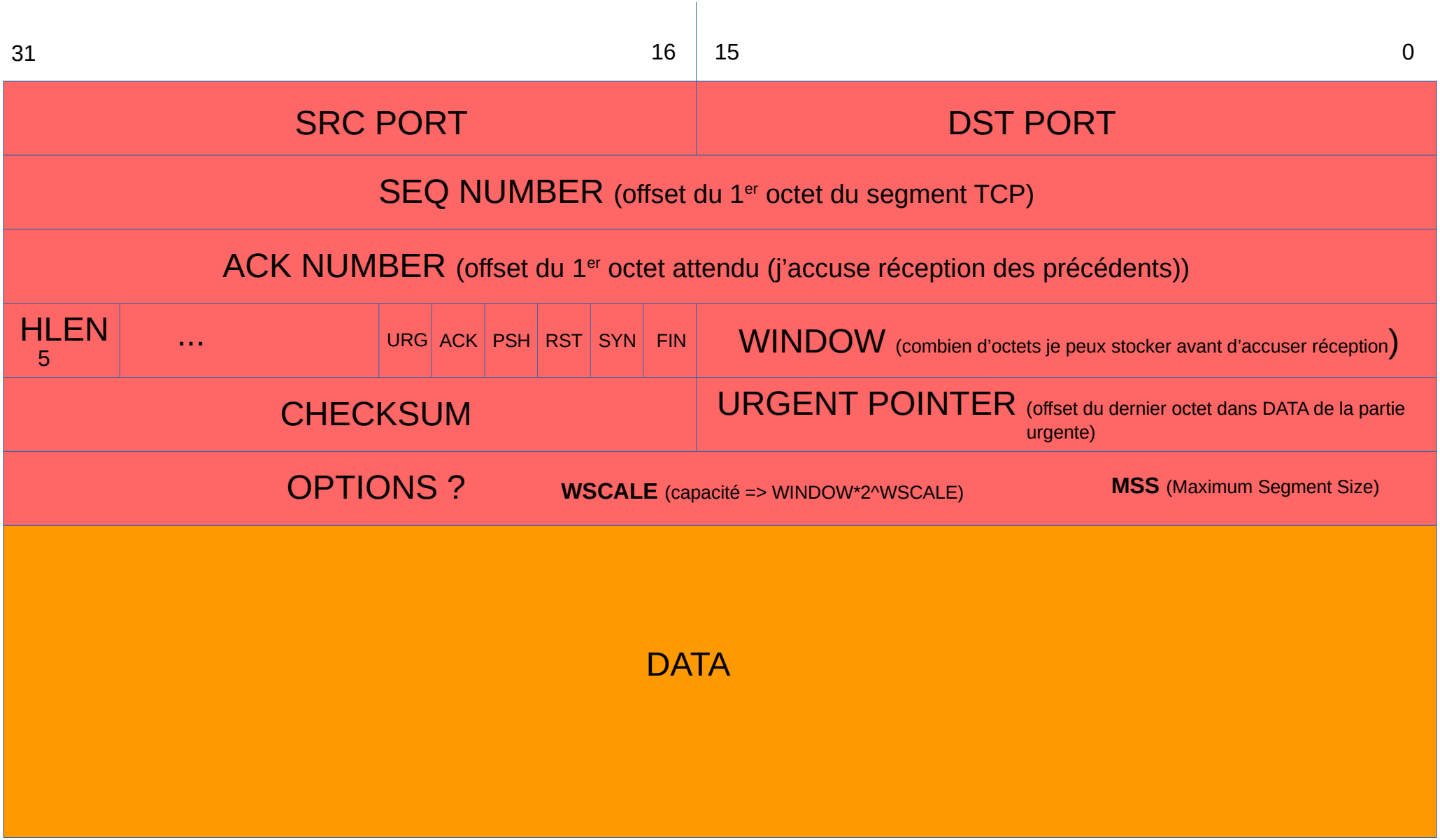
Exemple d’adresse de **processus destinataire** : **(77.66.55.44,(UDP, 53))**

TCP = IP + ports + interface logicielle
+ garantie du transport (acquittements)
+ longueur illimité (gestion de la fragmentation)
+ contrôle de flux (fenêtres)
(le tout en full-duplex)

Un **port** (TCP) est une sorte d'adresse numérique (sur 16 bits) tout comme pour les ports UDP. Le **nombre de ports** (TCP) possibles et attribuables (à ses processus) est 2^16 = 2^10 * 2^6 = **64 Ki**



Format (en-tête) des « segments » (« fragments ») TCP



L'adresse du **processus expéditeur** est (IP-SRC, (TCP, PORT-SRC))

L'adresse du **processus destinataire** est (IP-DST, (TCP, PORT-DST))

Plan des séances du module

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)
2) En-tête IP (chapitre 4 du livre de Laissus)

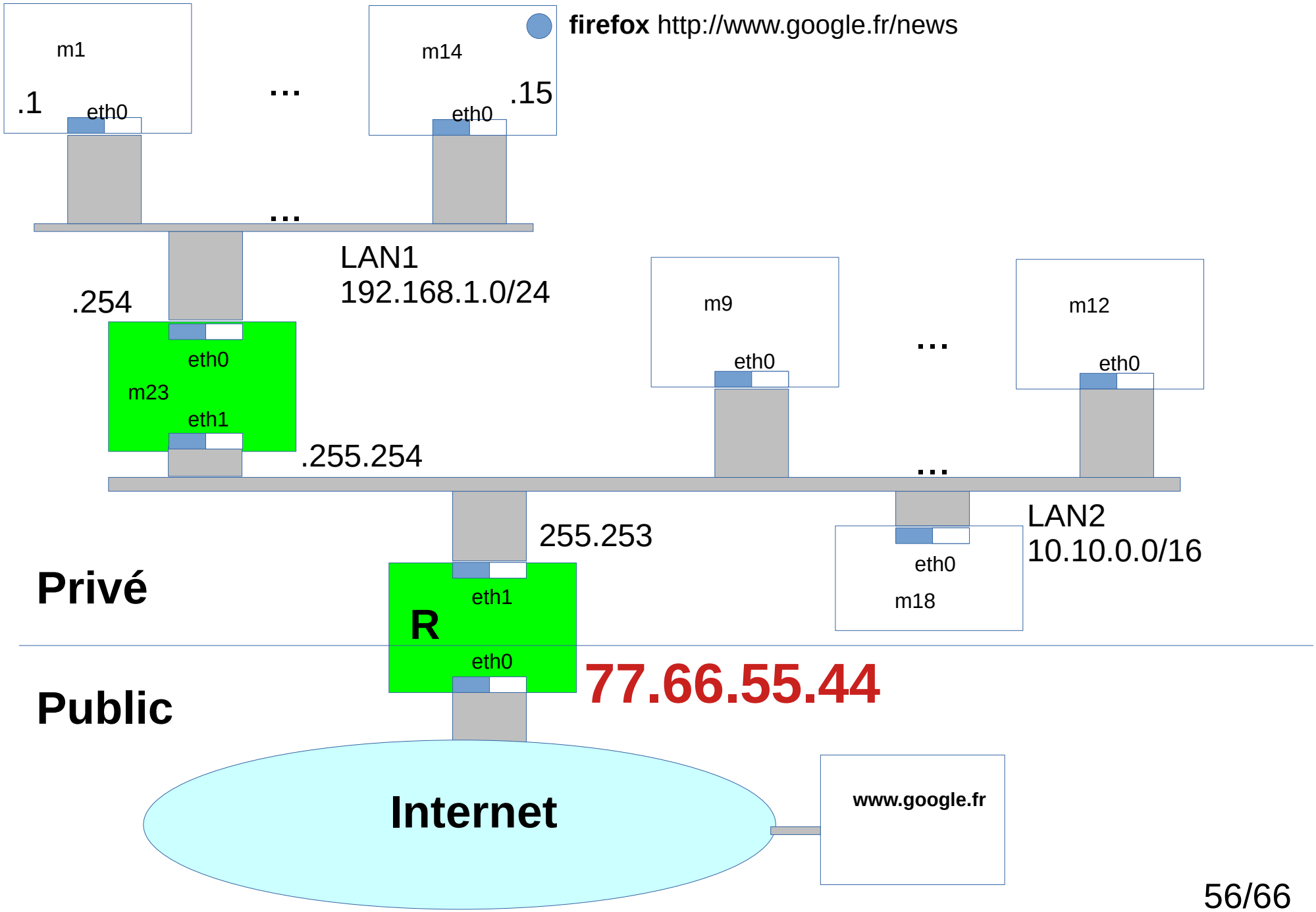
Cours n°4
1) routage (statique) IP
2) ICMP

Cours n°5
1) UDP (chapitre 5 du livre de Laissus)
2) TCP (chapitre 6)

Plan du cours n°6
2) SNAT (Source NAT)
3) DNAT (Destination NAT)

Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

NAT (Network Address Translation)

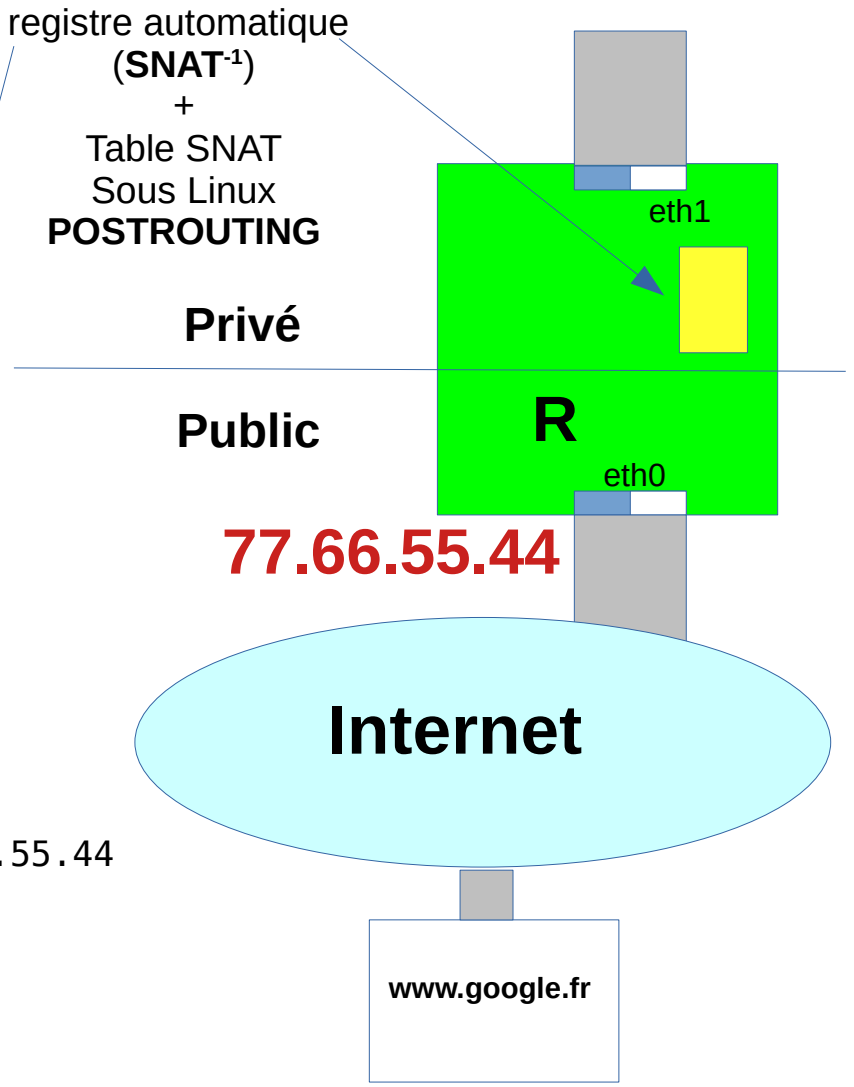


SNAT (Source NAT)

Le SNAT est l'activité (additionnelle) **d'un routeur** qui consiste à changer l'adresse SOURCE IP avec sa propre adresse (publique) pour que la réponse de DEST IP puisse revenir au bon endroit

registre automatique
(**SNAT^{-1}**)

Adresse SRC d'origine	PROT et Port SRC d'origine	(Trace du changement) Adresse SRC corrigée	PROT et Port SRC corrigé
192.168.1.15	TCP 59001	77.66.55.44	TCP 59001
192.168.1.12	TCP 60012	77.66.55.44	TCP 60012
10.10.0.7	UDP 65432	77.66.55.44	UDP 65432
…		…	
10.10.0.34	TCP 59001	77.66.55.44	TCP 59002

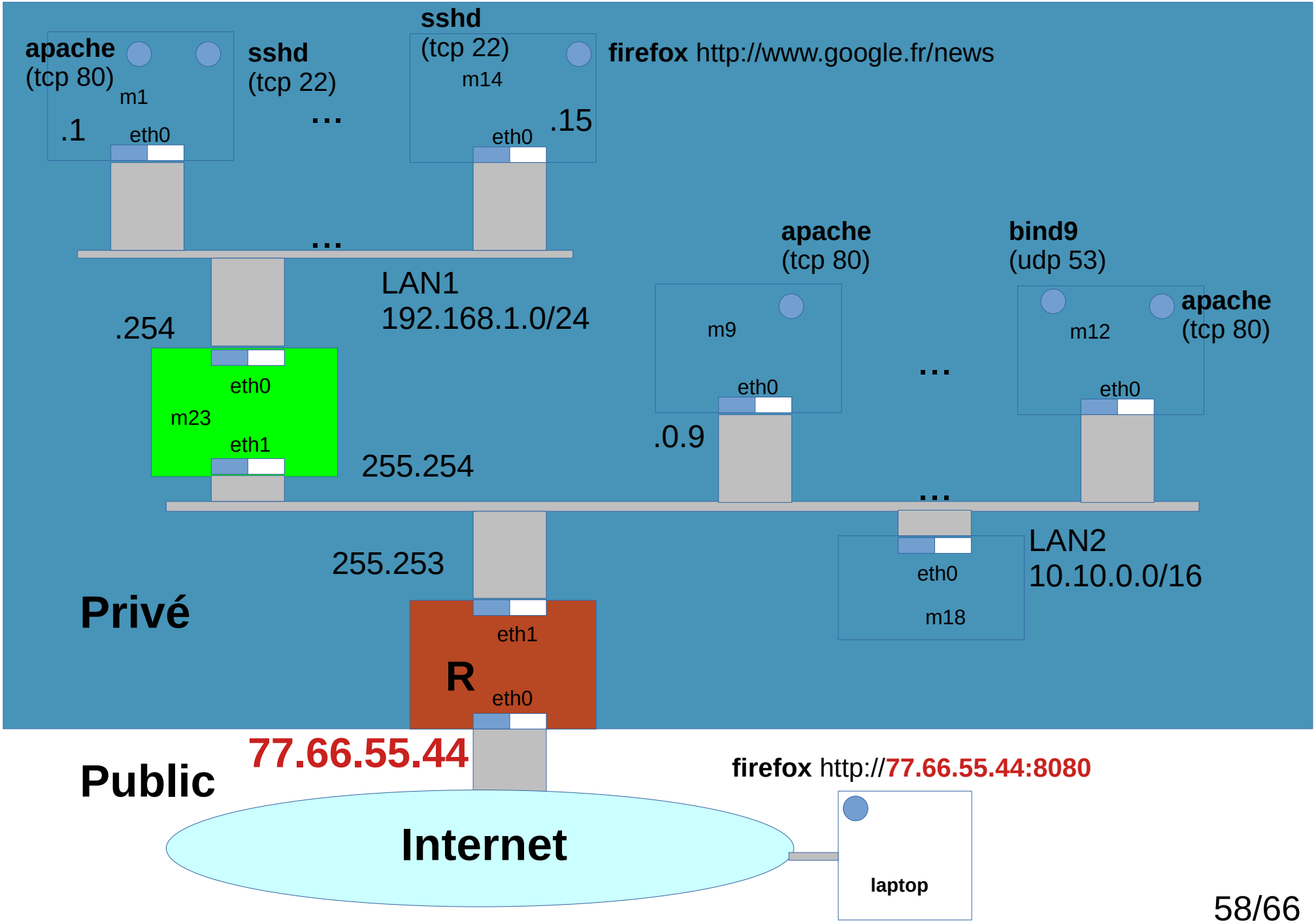



```
R# iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 77.66.55.44
```

OU

```
R# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

DNAT (Network Address Translation)



DNAT (Destination NAT)

Le DNAT est l'activité (additionnelle) **d'un routeur** qui consiste à changer l'adresse DEST IP avec une autre adresse (probablement privée) pour que l'initiative de SRC IP puisse arriver au bon endroit

		(Changement à faire)	
Adresse DST d'origine	PROT et Port DST d'origine	Adresse DST corrigée	PROT et Port DST corrigé
77.66.55.44	TCP 80	10.10.0.9	TCP 80
77.66.55.44	TCP 8080	10.10.0.12	TCP 80
77.66.55.45	TCP 81	192.168.1.1	TCP 80
…		…	
77.66.55.45	TCP 22	192.168.1.1	TCP 22

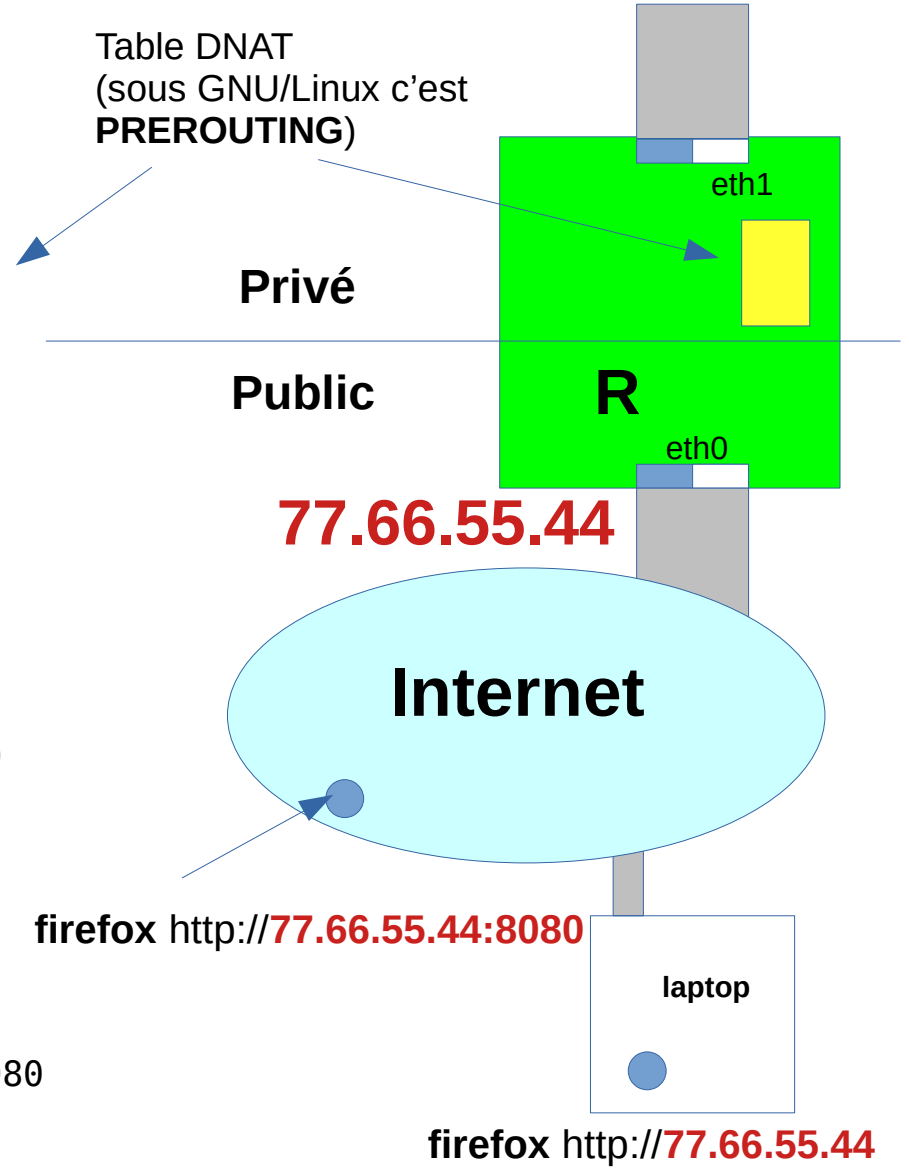
```
R# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80
      -j DNAT --to 10.10.0.9:80

R# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 22
      -j DNAT --to 192.168.1.1:22

R# iptables -t nat -A PREROUTING -i eth0 -p udp --dport 53
      -j DNAT --to 10.10.0.12:53

R# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 8080
      -j DNAT --to 192.168.1.1:80

R# iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 2222
      -j DNAT --to 192.168.1.15:22
```



Plan des séances du module

Cours n°1
1) Internet est un hypergraphe
2) Principe de l'encapsulation
3) Couche liaison Ethernet

Cours n°2
1) quelques détails sur notre exemple de couche liaison, c'est-à-dire ETHERNET (câblage)
2) Adresses IP v4 (sur 32 bits, chapitre 3 du livre de Laissus)

Cours n°3
1) VLAN de type 1 (ports)
sur 1 seul commutateur ou plusieurs
(baie de brassage)
2) En-tête IP (chapitre 4 du livre de Laissus)

Cours n°4
1) routage (statique) IP
2) ICMP

Cours n°5
1) UDP (chapitre 5 du livre de Laissus)
2) TCP (chapitre 6)

Plan du cours n°6
1) SNAT (Source NAT)
2) DNAT (Destination NAT)

Plan du cours n°7
1) Filtrage (sous Linux)
2) Filtrage avec SNAT et DNAT

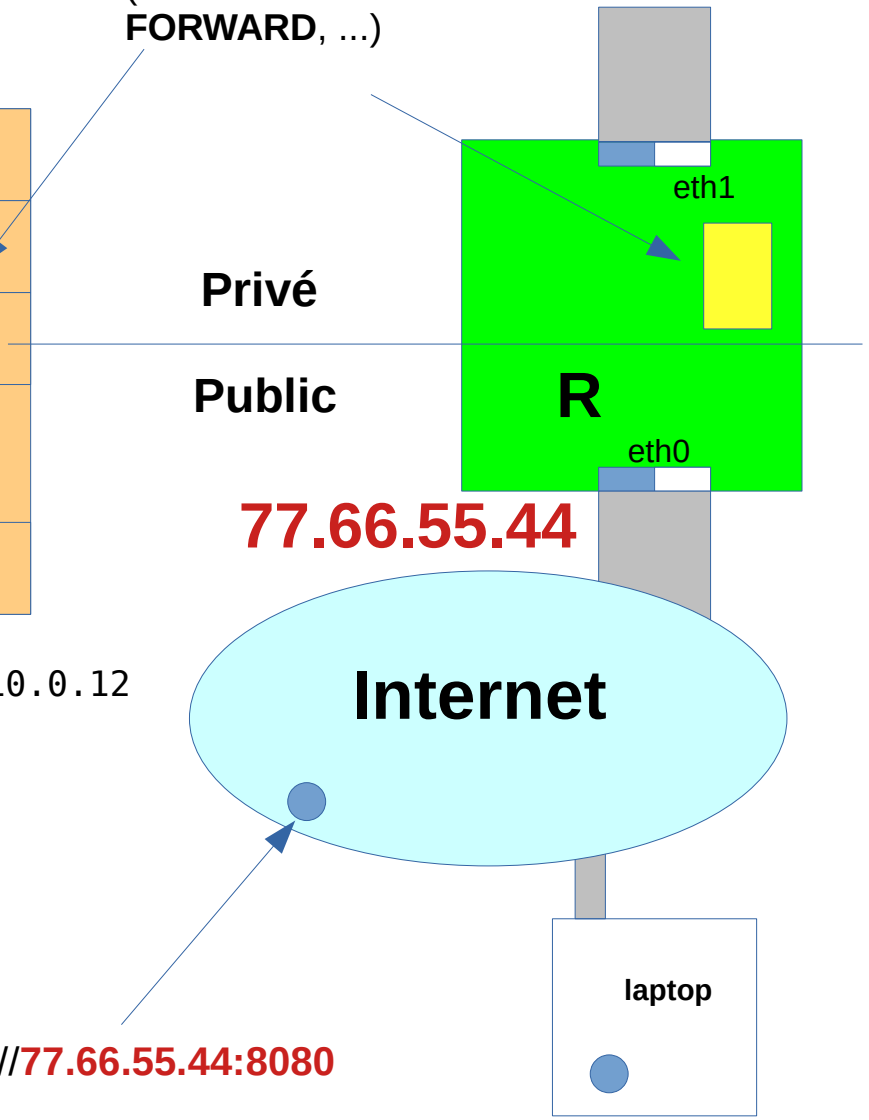
FILTRAGE (parefeu)

Le FILTRAGE est l'activité (additionnelle) **d'un routeur** qui consiste à **supprimer ou pas** certains paquets qui devraient être routés

Caractéristiques de la trame/paquet/datagramme	Que faire ? Laisser passer ou pas ?
SRC-IP = 88.77.66.0/24 port-dest 80	ACCEPT
IF-IN eth0 IF-OUT eth1 Dest-IP 10.10.0.12 TCP port-dest 80	ACCEPT
…	…
SINON (politique par défaut)	DROP

FORWARD

Tables de FILTRAGE (sous GNU/Linux c'est **FORWARD**, ...)




```
R# iptables -t filter -A FORWARD -i eth0 -o eth1 -d 10.10.0.12
      -p tcp --dport 80 -j ACCEPT

R# iptables -t filter -P FORWARD DROP
```

FILTRAGE (parefeu)

Le FILTRAGE est l'activité (additionnelle) **d'un routeur** qui consiste à **supprimer ou pas** certains paquets qui devraient être routés

```
R# iptables -t filter -A FORWARD -i eth1 -j ACCEPT
```

Laisser passer tout ce qui vient de chez nous (de nos réseaux locaux privés)

```
R# iptables -t filter -A FORWARD
      -i eth0 -o eth1
      -m state --state ESTABLISHED,RELATED
      -j ACCEPT
```

*Laisser passer les **réponses** depuis Internet (eth0)*

```
R# iptables -t filter -A FORWARD
      -i eth0 -o eth1
      -m state --state NEW
      -d 192.168.1.1 -p tcp --dport 22 -j ACCEPT
```

*Laisser passer les **initiatives** depuis Internet (eth0) à destination de 192.168.1.1 pour le service (TCP,22) (ssh)*

```
R# iptables -t filter -A FORWARD
      -i eth0 -o eth1
      -m state --state NEW
      -d 10.10.0.12 -p tcp --dport 80 -j ACCEPT
```

*Laisser passer les **initiatives** depuis Internet (eth0) à destination de 10.10.0.12 pour le service (TCP,80) (http)*

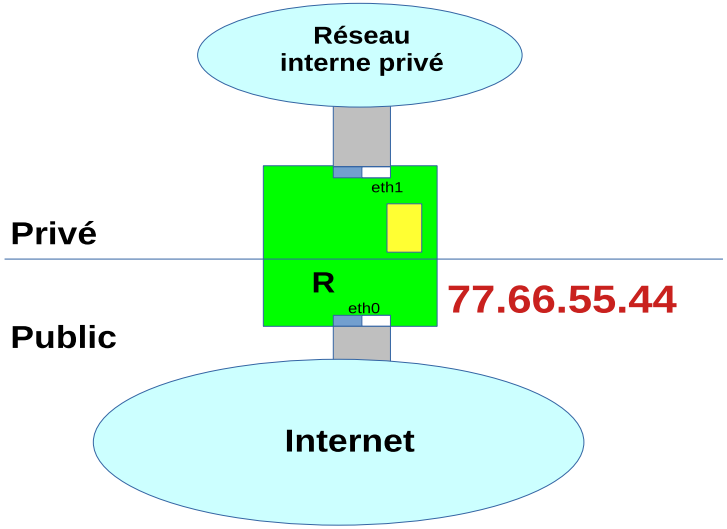
```
R# iptables -t filter -P FORWARD DROP
```

Par défaut, ne rien laisser passer (ni dans un sens, ni dans l'autre)

FILTRAGE (parefeu)

Que signifie « **ouvrir un port** » ?

Par exemple : ouvrir le port UDP 53 servi par 10.10.10.53



Réponse : **deux** choses à faire
(1) DNAT de **77.66.55.44 UDP 53** vers **10.10.10.53 UDP 53**
(2) ACCEPTER les requêtes (**initiatives**) vers **10.10.10.53 UDP 53**
C'est-à-dire :

```
R# iptables -t nat -A PREROUTING
      -i eth0 -p udp --dport 53
      -j DNAT --to 10.10.10.53:53
```

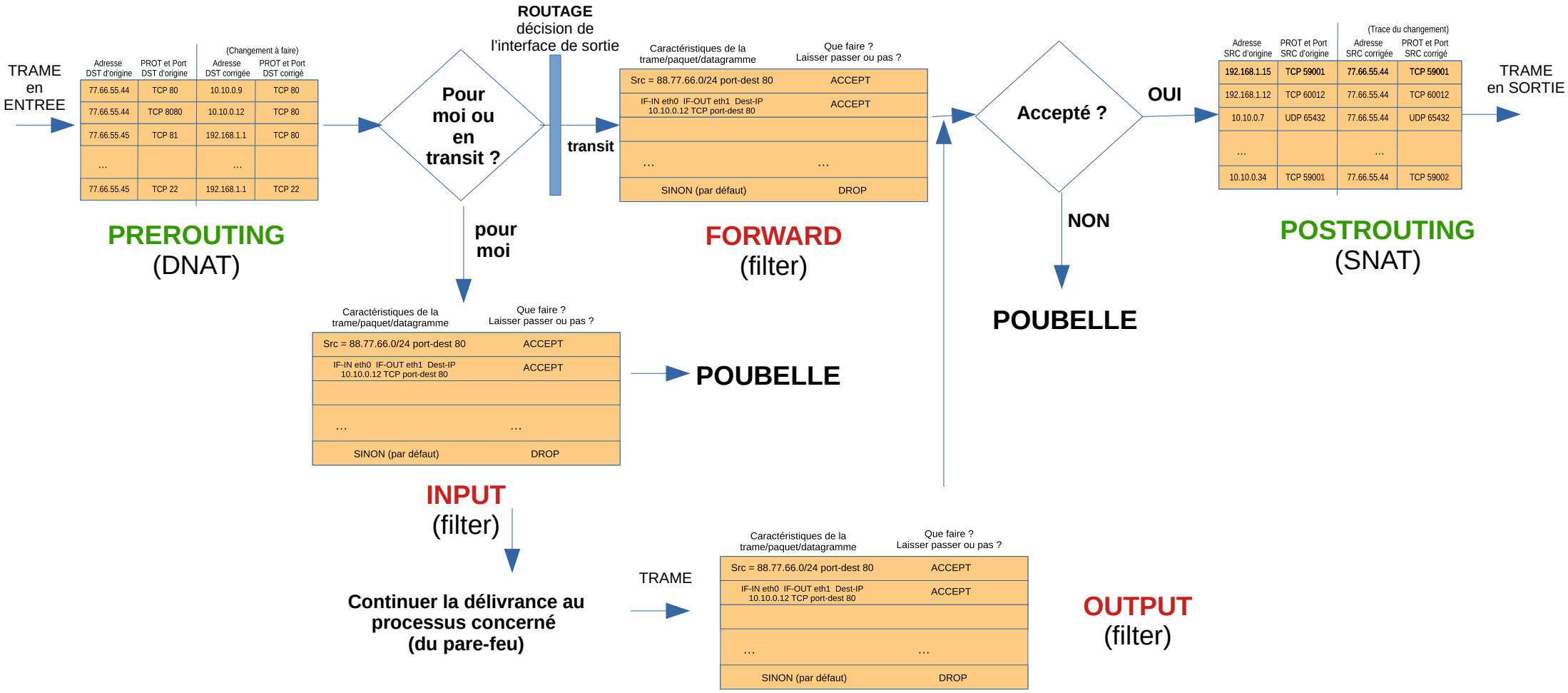
(1)

```
R# iptables -t filter -A FORWARD
      -i eth0 -o eth1
      -m state --state NEW
      -d 10.10.10.53 -p udp --dport 53
      -j ACCEPT
```

(2)

FILTRAGE (pare-feu) (sous Linux)

Le FILTRAGE est l'activité (additionnelle) **d'un routeur** qui consiste à **supprimer ou pas** certains paquets qui devraient être routés



Traitement des paquets en transit :

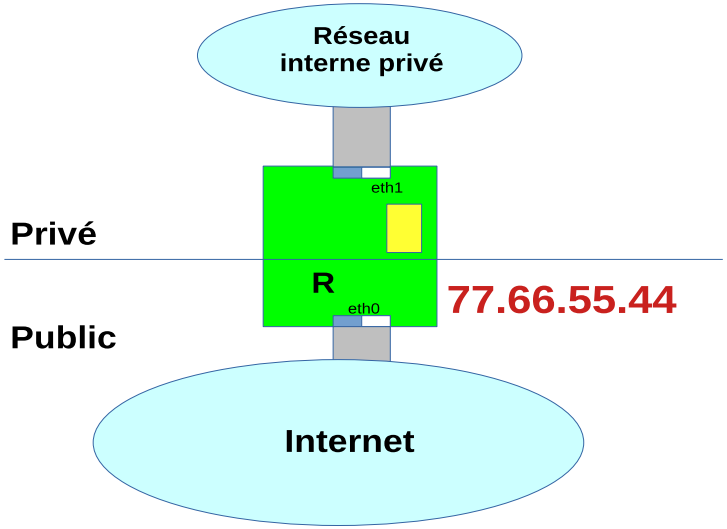
(1) PREROUTING (DNAT) → **(2) ROUTAGE (décision de l'interface de sortie)** → **(3) FORWARD (filtrage)** → **(4) POSTROUTING (SNAT)**

(3) FORWARD (filtrage) → **POUBELLE**

FILTRAGE (parefeu)

Que signifie « **ouvrir un port** » ?

Par exemple : ouvrir le port UDP 53 servi par 10.10.10.53



Réponse : **deux** choses à faire
(1) DNAT de **77.66.55.44 UDP 53** vers **10.10.10.53 UDP 53**
(2) ACCEPTER les requêtes (initiatives) vers **10.10.10.53 UDP 53**
C'est-à-dire :

```
R# iptables -t nat -A PREROUTING -i eth0 -p udp --dport 53
      -j DNAT --to 10.10.10.53:53

R# iptables -t filter -A FORWARD
      -i eth0 -o eth1
      -m state --state NEW
      -d 10.10.10.53 -p udp --dport 53 -j ACCEPT
```

Automatisation par une fonction Bash :

```
# Usage: fw_open_port PROTOCOL PORT SERVER SERVER_PORT
# ---
function fw_open_port {
  local PROTOCOL=$1
  local PORT=$2
  local SERVER=$3
  local SERVER_PORT=$4

  iptables -t nat -A PREROUTING -i eth0 -p $PROTOCOL --dport $PORT \
      -j DNAT --to $SERVER:$SERVER_PORT

  iptables -t filter -A FORWARD \
      -i eth0 -o eth1           \
      -m state --state NEW      \
      -d $SERVER -p $PROTOCOL --dport $SERVER_PORT -j ACCEPT
}
```

FILTRAGE (parefeu)

Que signifie « **ouvrir un port** » ?

Automatisation par une fonction Bash :

```
function fw_open_port {
  local PROTOCOL=$1
  local PORT=$2
  local SERVER=$3
  local SERVER_PORT=$4

  iptables -t nat -A PREROUTING -i eth0 -p $PROTOCOL --dport $PORT \
      -j DNAT --to $SERVER:$SERVER_PORT

  iptables -t filter -A FORWARD \
      -i eth0 -o eth1           \
      -m state --state NEW      \
      -d $SERVER -p $PROTOCOL --dport $SERVER_PORT -j ACCEPT
}
```

Exemples d'appels (dans un script de configuration du parefeu):

```
# Partie principale du script
# ---
# Usage: fw_open_port PROTOCOL PORT SERVER SERVER_PORT

# HTTP
fw_open_port tcp 80   10.10.0.9   80
fw_open_port tcp 8080 10.10.0.12  80
fw_open_port tcp 81   192.168.1.1 80

# SSH
fw_open_port tcp 22   192.168.1.1 22

# DNS
fw_open_port udp 53   10.10.10.53 53
```