UNIVERSITÉ DE LYON

Université Claude Bernard Lyon 1

Licence Math-Informatique 1ère année

Partie 8

Olivier Glück

Université LYON 1 / Département Informatique

Olivier.Gluck@univ-lyon1.fr

http://perso.univ-lyon1.fr/olivier.gluck

Copyright

Plan du cours

- CM1 : Internet, les réseaux et le web
- CM2 : Pages HTML et feuilles de styles CSS
- CM3 : Web interactif, formulaires, pages dynamiques et PHP
- CM4 : Protocole HTTP, méthodes GET et POST
- CM5 : Les applications d'Internet
- CM6 : La couche transport : les protocoles TCP et UDP
- CM7 : Le protocole IP
- CM8 : Les protocoles Ethernet, ARP et ICMP. Synthèse des échanges entre un client et serveur Web

Université Claude Bernard Lyon 1

Faculté des Sciences et Technologies
Dpt Informatique

CM8 – Les protocoles Ethernet, ARP et ICMP

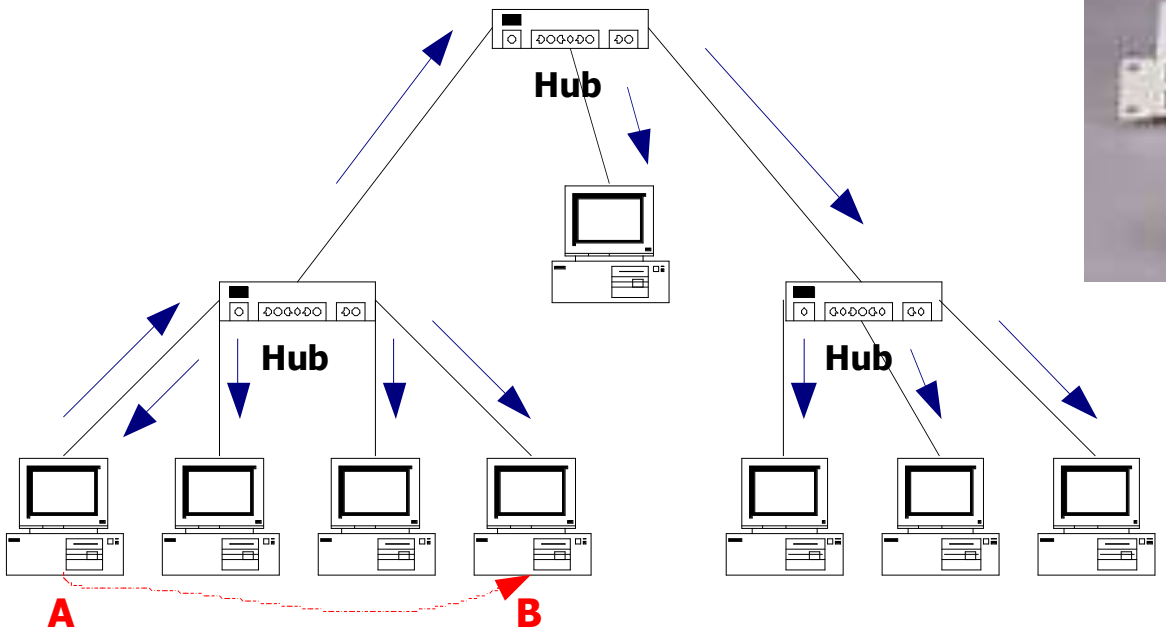
Format de la trame Ethernet
Le protocole ARP
L'architecture TCP/IP
Synthèse des échanges entre un navigateur Web et un serveur Web

Plan du CM8

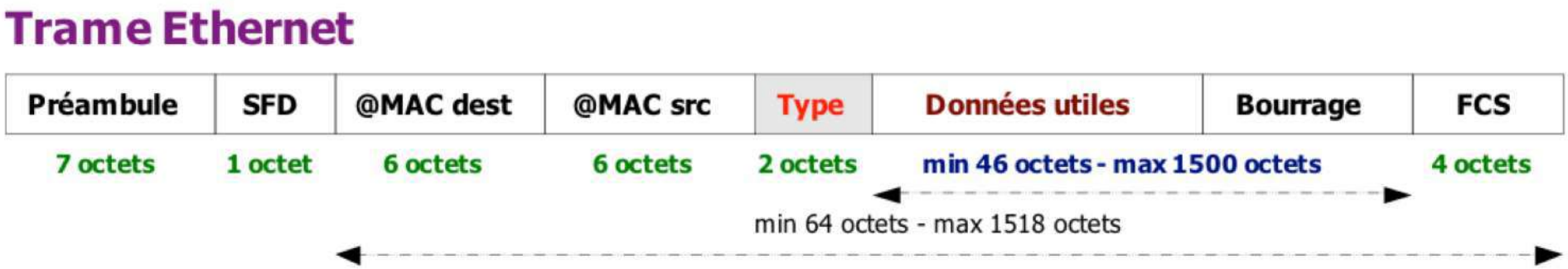
- Format de la trame Ethernet
- Le protocole ARP
- L'architecture TCP/IP
- Synthèse des échanges entre un navigateur Web et un serveur Web

Ethernet, le protocole des cartes réseaux

- Une carte réseau Ethernet fabrique un signal qui est envoyée à une autre carte réseau Ethernet
- Les répéteurs (Hub) ou commutateurs (Switch) interconnectent les ordinateurs entre eux
- Exemple avec des Hub



Format de la trame Ethernet

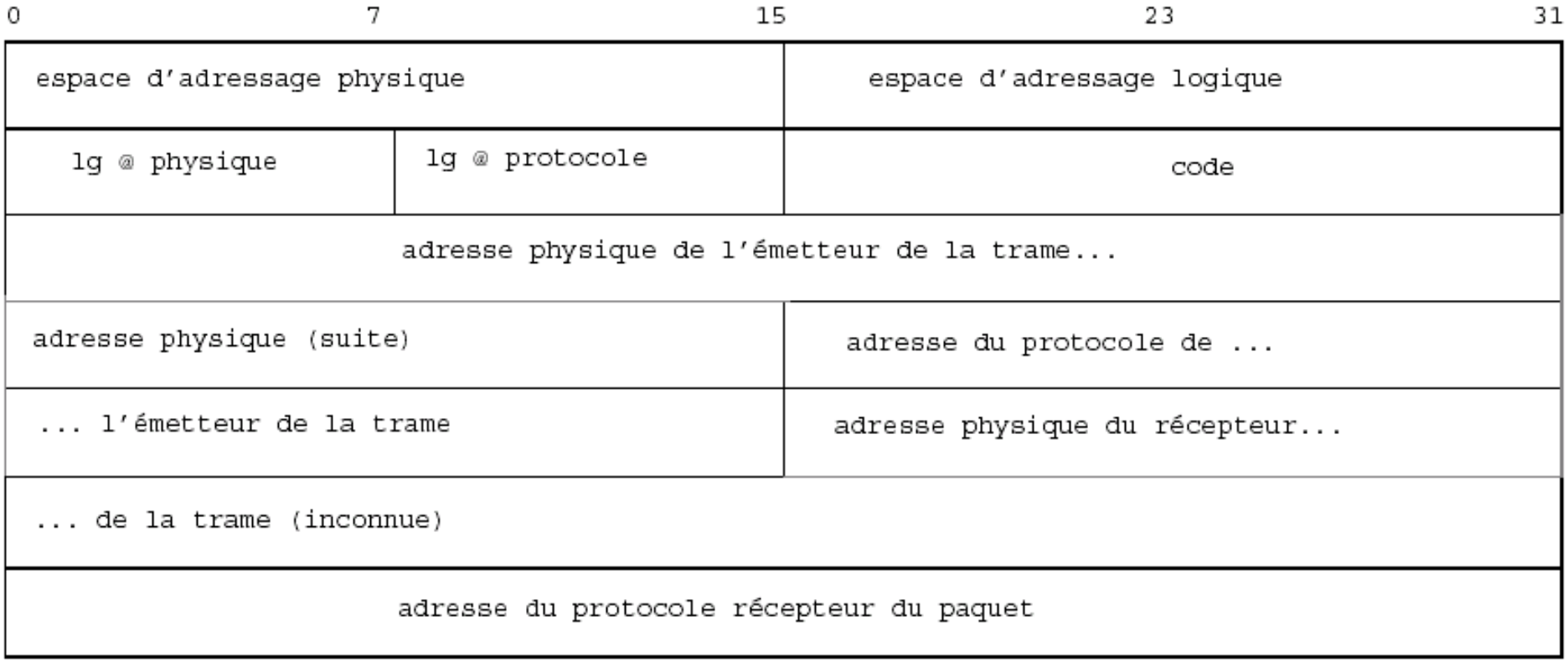


- Préambule : 7 fois 10101010 pour la synchronisation bit
- SFD (*Start Frame Delimitor*) : 10101011 pour la synchronisation octet
- Bourrage si Lg < 46 octets pour détection collision
- FCS sur 4 octets pour la détection d'erreur
- Différence IEEE 802.3/Ethernet : champ Lg/Type

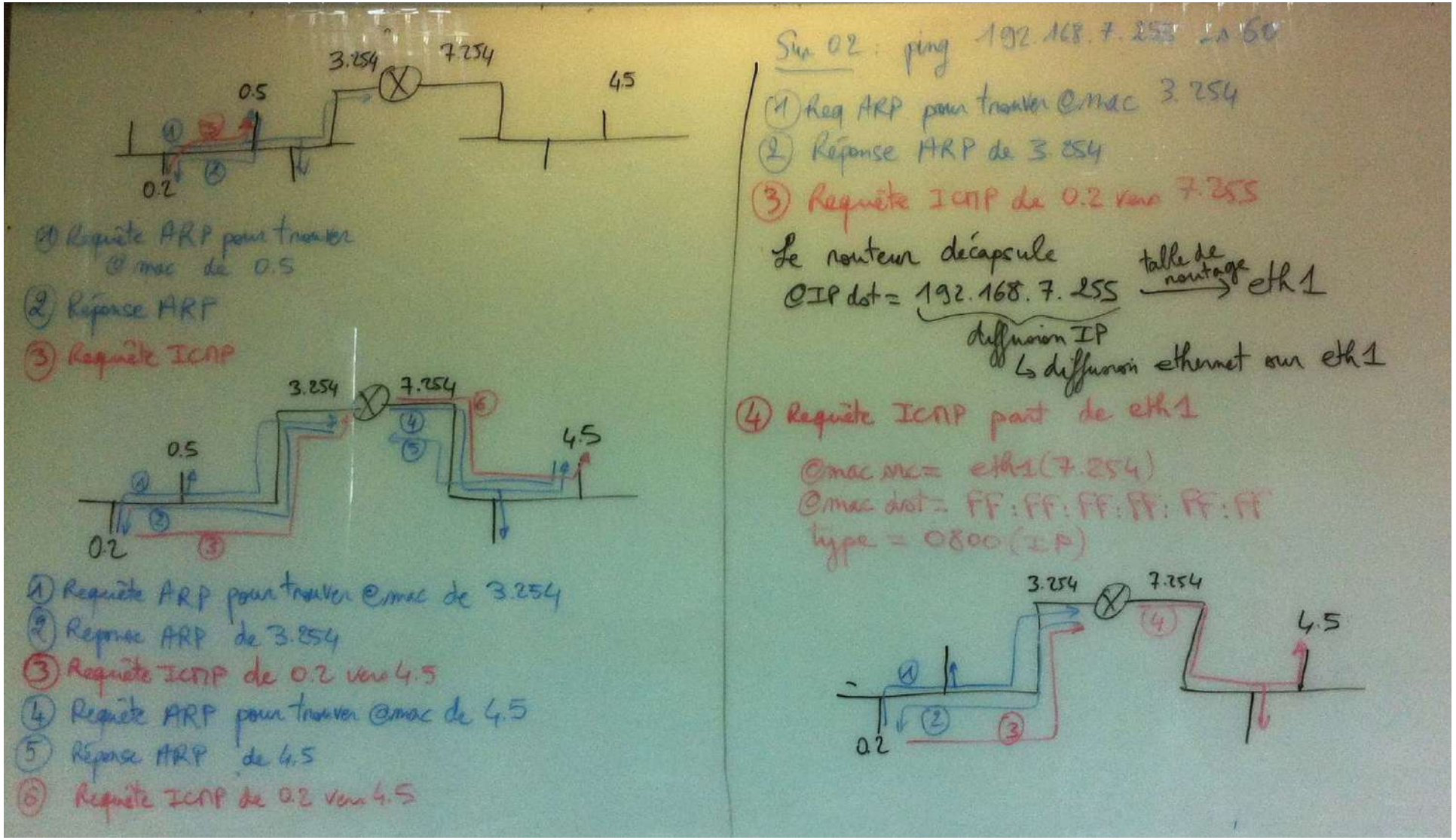
ARP – Rôle et fonctionnement

- ARP (Address Resolution Protocol) sert à trouver l'adresse MAC d'une carte réseau alors que l'on connaît son adresse IP
- Si la machine source et destinataire sont sur le même réseau
  - 1 - requête ARP (broadcast MAC)
  - 2 - réponse ARP (le destinataire a reçu le broadcast et s'est reconnu, il envoie son @MAC)
  - 3 - la source peut envoyer ses données vers le destinataire (adresse MAC destination connue)
- Si elles ne sont pas sur le même réseau
  - la diffusion ne passe pas le routeur
  - résolution de proche en proche : le client envoie ses données au prochain saut dont l'adresse MAC est trouvée par ARP, puis le prochain envoie au suivant en utilisant sa table de routage et ARP…

ARP - Format du paquet



ARP - ICMP


3.254
7.254
0.5
4.5
0.2
① Requête ARP pour trouver @mac de 0.5
② Réponse ARP
③ Requête ICMP
3.254
7.254
0.5
4.5
0.2
① Requête ARP pour trouver @mac de 3.254
② Réponse ARP de 3.254
③ Requête ICMP de 0.2 vers 4.5
④ Requête ARP pour trouver @mac de 4.5
⑤ Réponse ARP de 4.5
⑥ Requête ICMP de 0.2 vers 4.5
Sur 0.2 : ping 192.168.7.255 -n 60
① Req ARP pour trouver @mac 3.254
② Réponse ARP de 3.254
③ Requête ICMP de 0.2 vers 7.255
Le routeur décapsule
@IP dst = 192.168.7.255
diffusion IP
table de routage
eth1
↳ diffusion ethernet sur eth1
④ Requête ICMP part de eth1
@mac src = eth1(7.254)
@mac dst = FF:FF:FF:FF:FF:FF
type = 0800 (IP)
3.254
7.254
4.5
0.2

Sur le routeur : décapsulation IP. Comment aller vers 4.5 ?

entête IP → @IP dst = 192.168.4.5 → table de routage du routeur

eth0 192.168.3.254/22 — 192.168.0.0/22 → eth0

eth1 192.168.7.254/22 — 192.168.4.0/22 → eth1

Pour aller vers 4.5, on passe par eth1

14 | 20 | 8 | 60 | 4

→ @mac src = eth1 (7.254)

type = 0800 (IP)

@mac dst = ? celle de 4.5 que le routeur ne connait pas

④ Requête ARP

Trame 4 — ARP [@mac src = eth1 (7.254) ; @IP src = 192.168.7.254 ; @mac dst = 0 (?) ; @IP dst = 192.168.4.5]

Eth [@mac dst = FF:FF:FF:FF:FF:FF ; @mac src = eth1 (7.254) ; type = 0806 (ARP)]

⑤ Réponse ARP

Trame 5 — ARP [@mac src = eth1 (7.254) ; @IP src = 192.168.7.254 ; @mac dst = eth0 (4.5) ; @IP dst = 192.168.4.5]

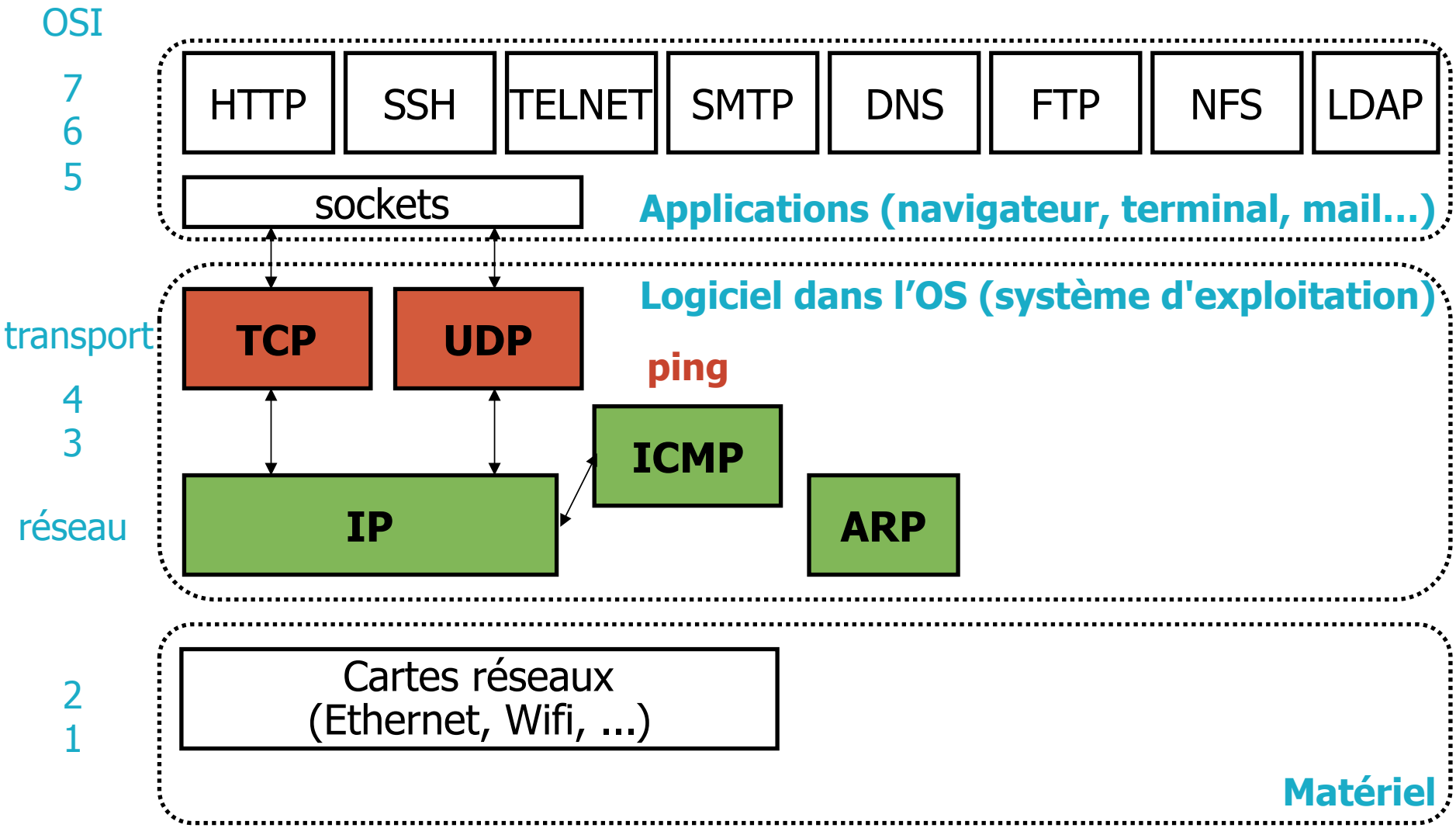
Eth [@mac dst = eth1 (7.254) ; @mac src = eth0 (4.5) ; type = 0806 (ARP)]

Trame ⑥ Requête ICMP

@mac dst = eth0 (4.5)

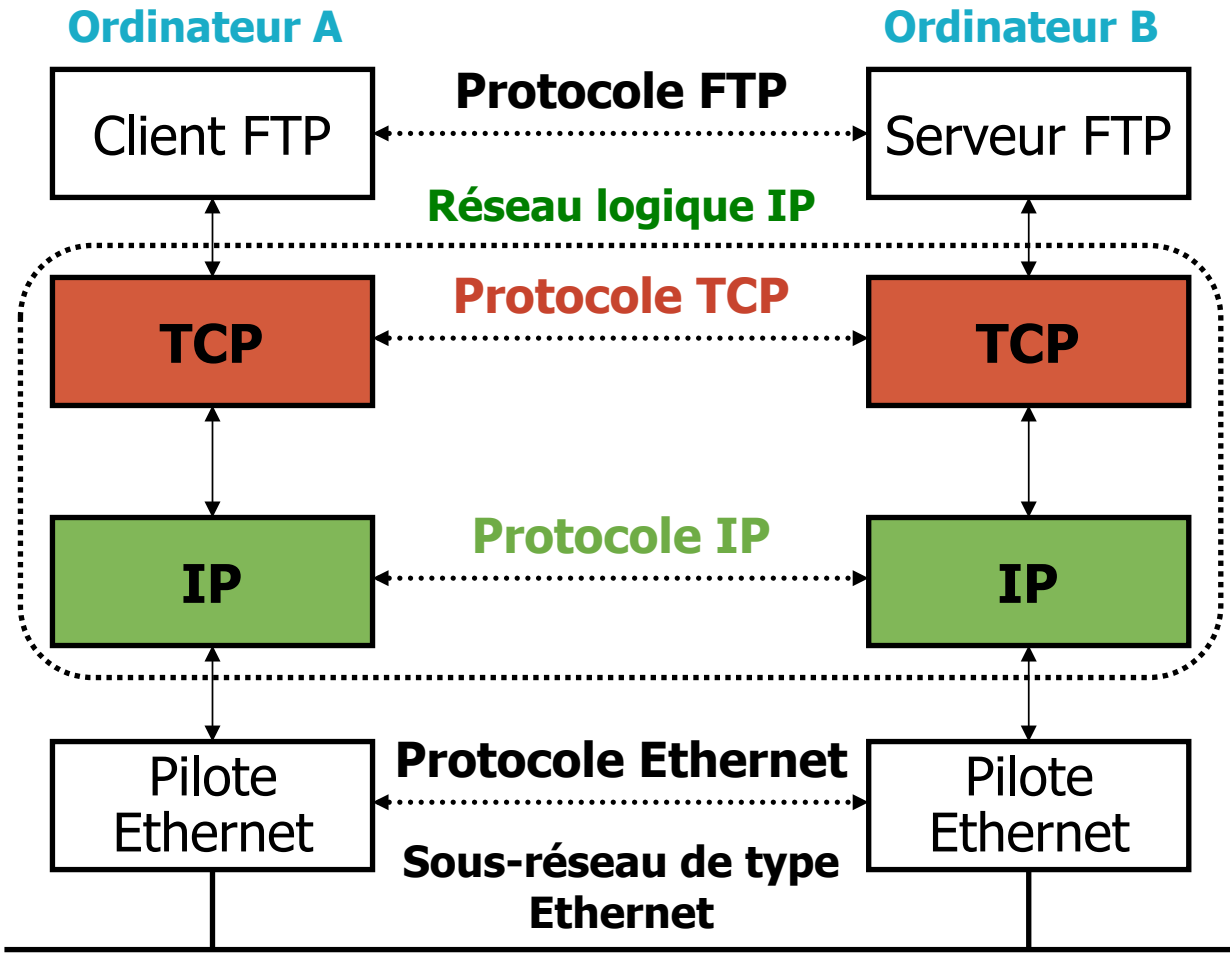
20 entête IP [@IP dst = 192.168.4.5 ; @IP src = 192.168.0.2 ; proto = 1 (ICMP)]

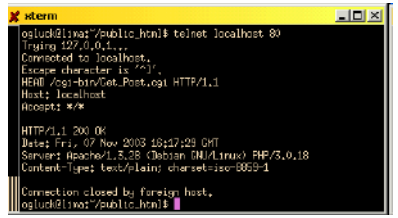
L'architecture de TCP/IP



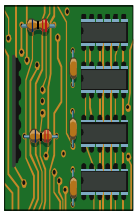
Communications sans routeur

- Deux machines sur un même sous réseau





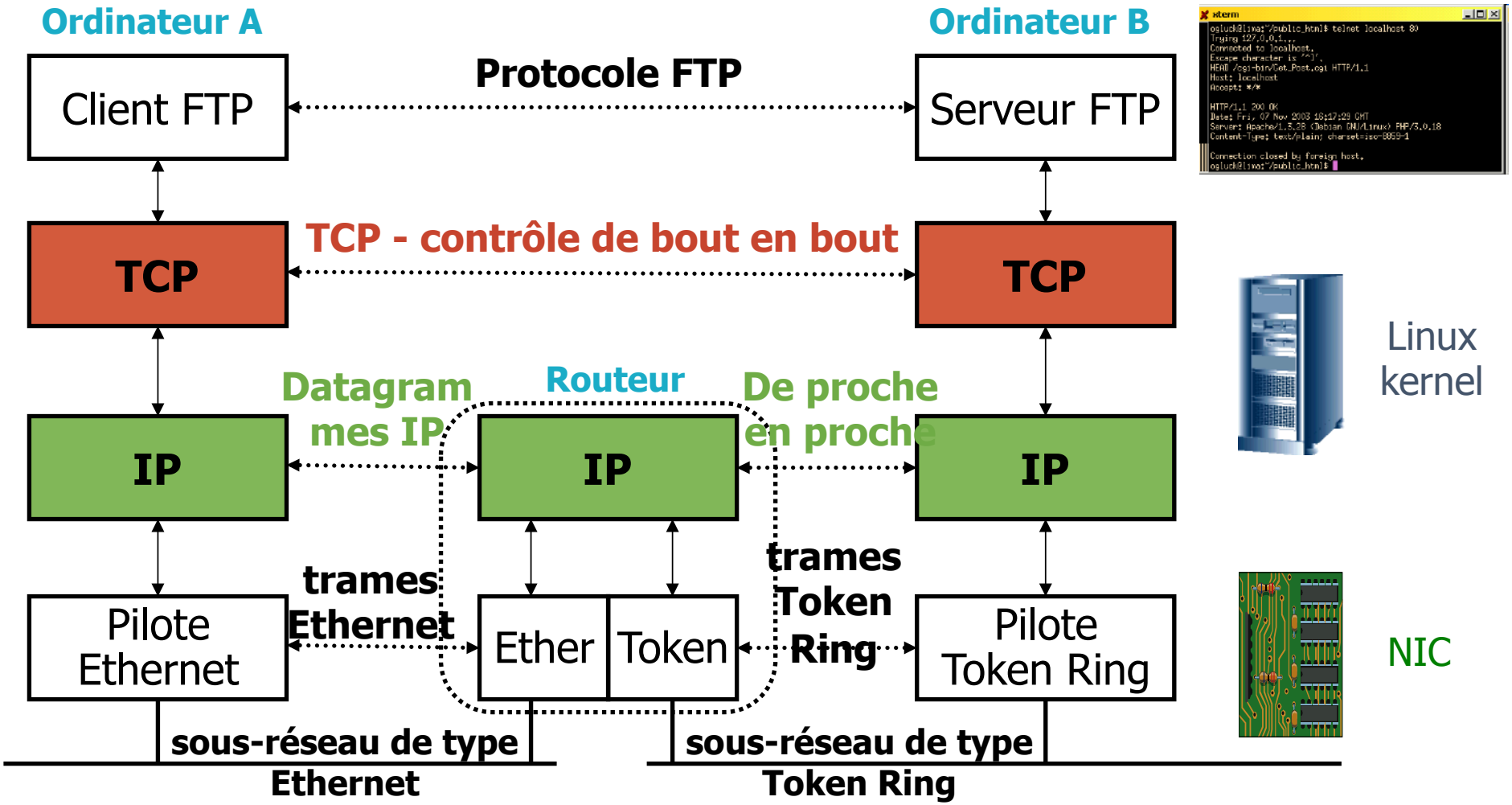
Linux kernel



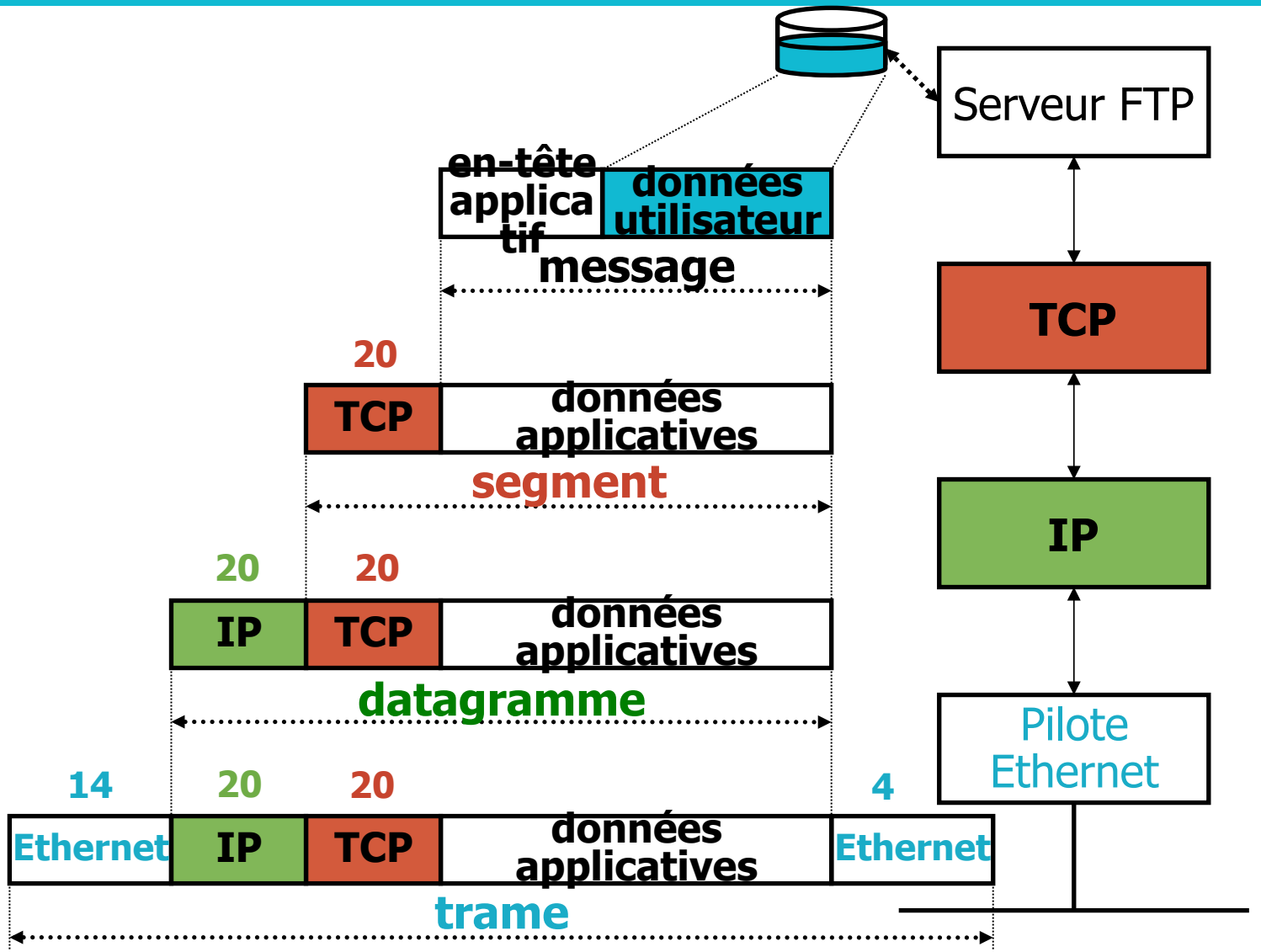
NIC

Communications avec routeur(s)

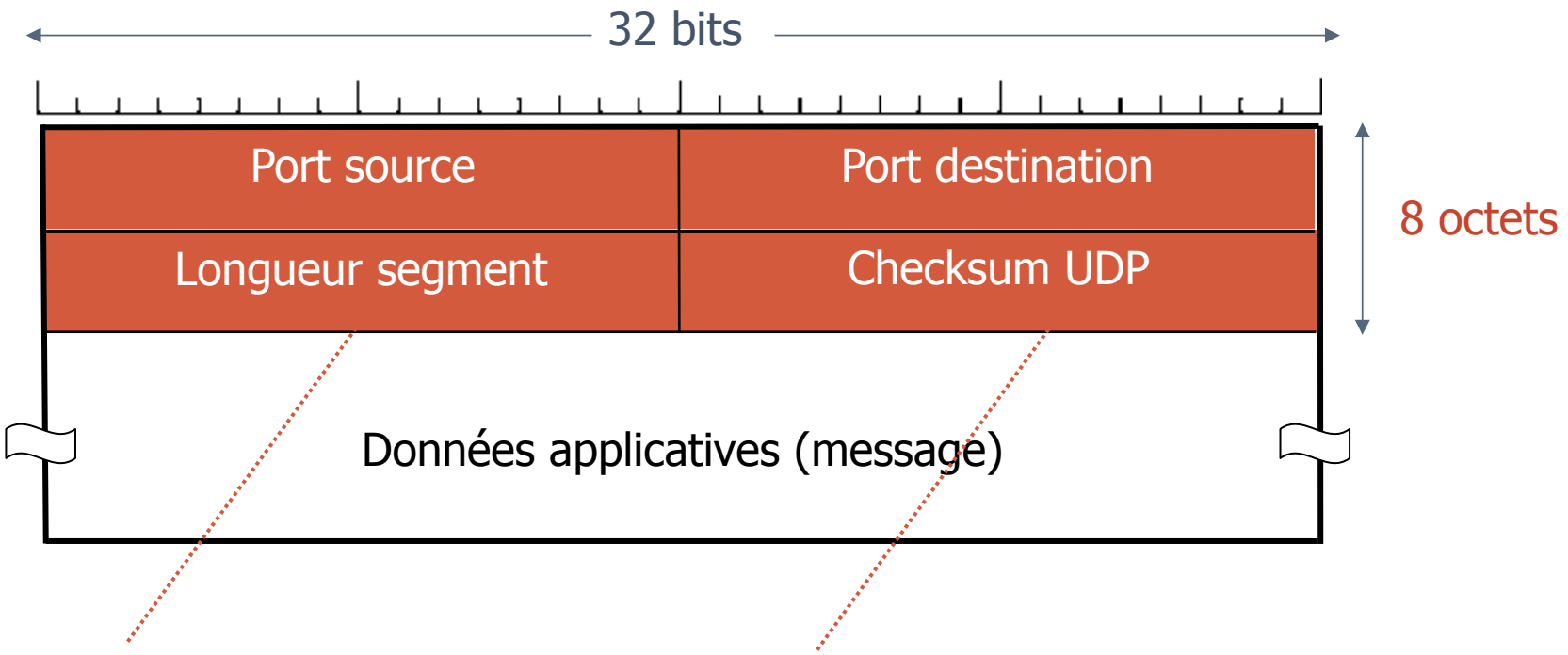
- Prise en compte de l'hétérogénéité



L'encapsulation : ajout des en-têtes



L'en-tête UDP : 8 octets



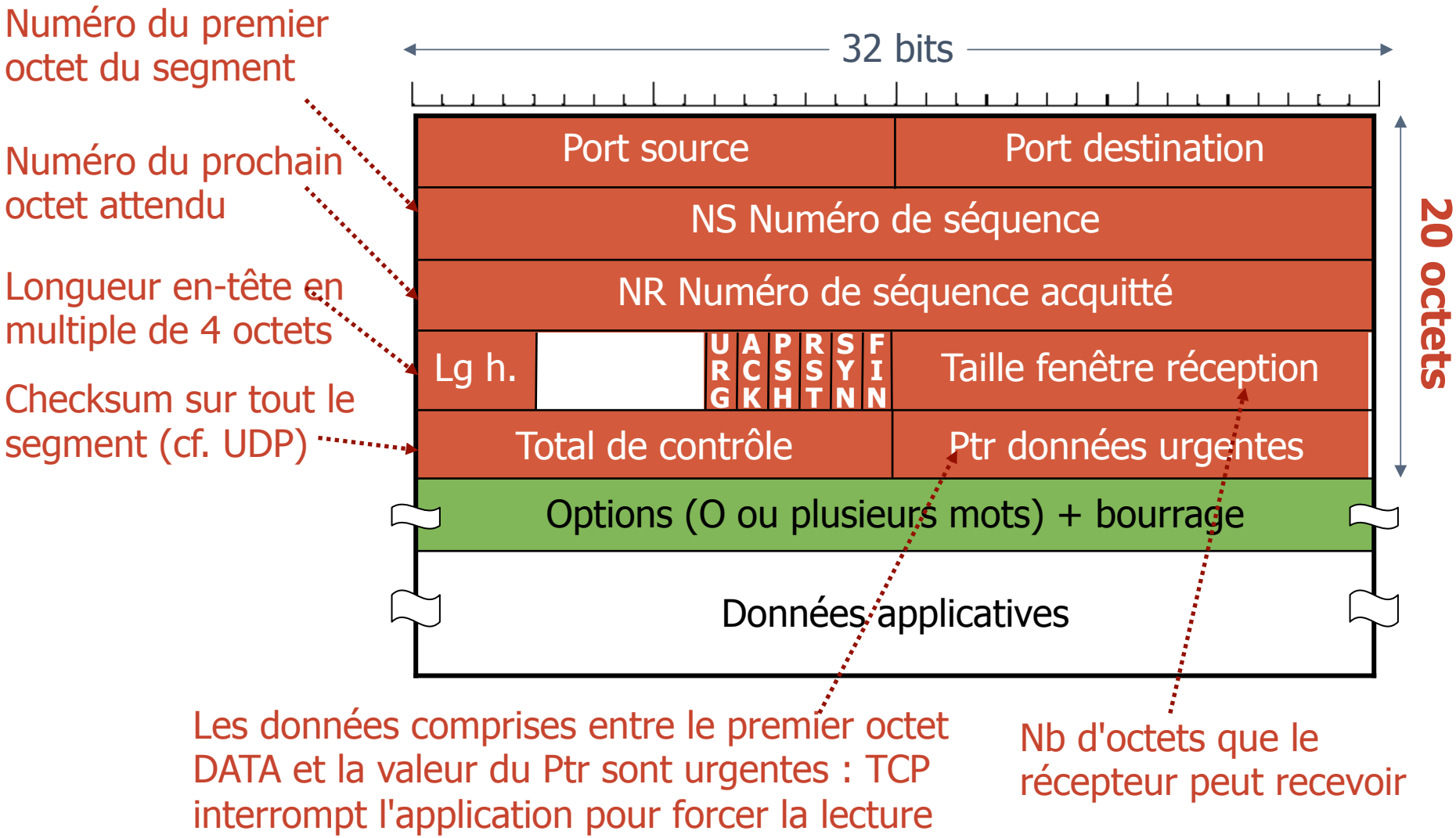
Taille totale du datagramme (en-tête+données)

Somme de contrôle du datagramme (en-tête+données)

optionnel : peut être à 0

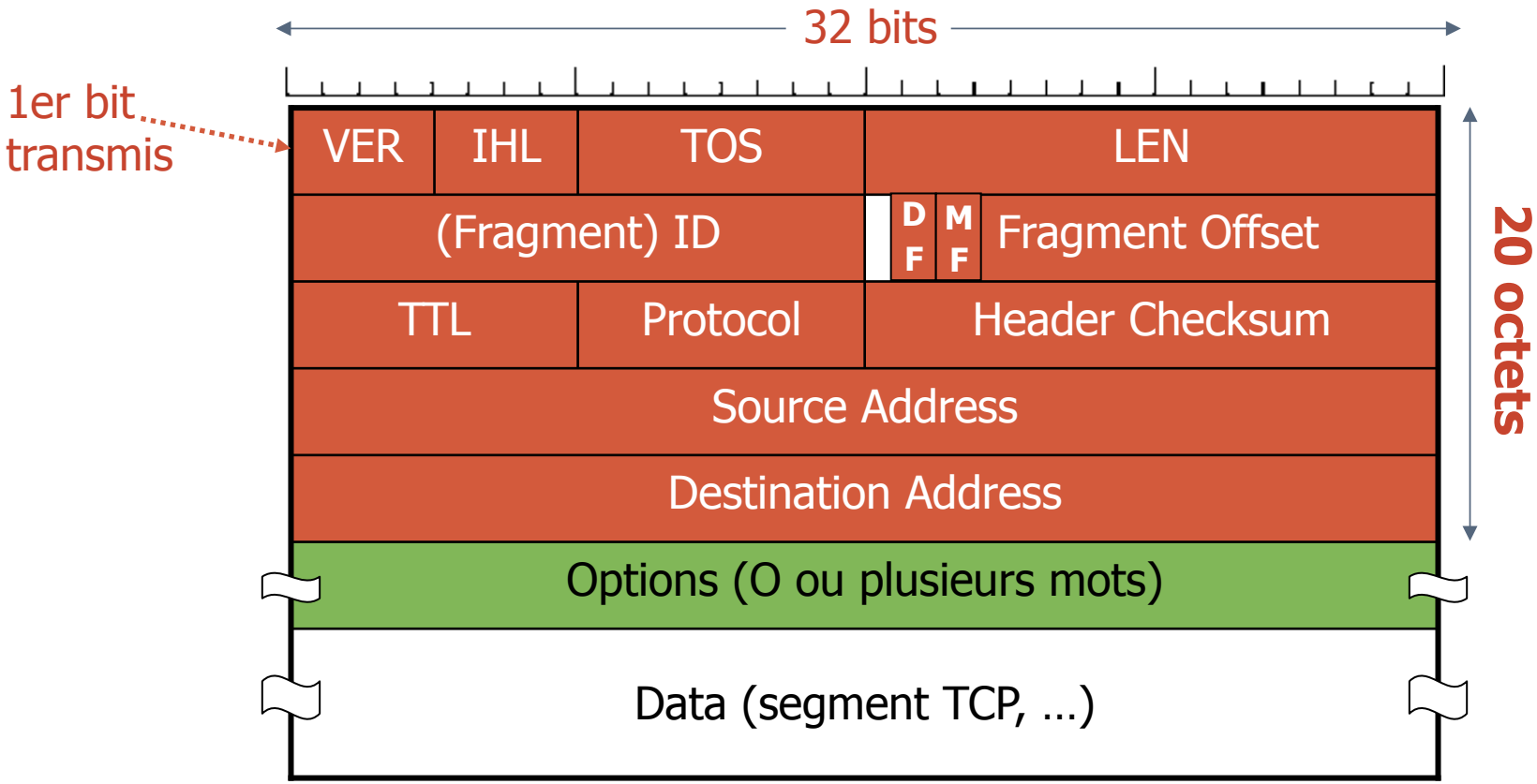
UDP = IP + numéros de port !!

L'en-tête TCP : 20 octets

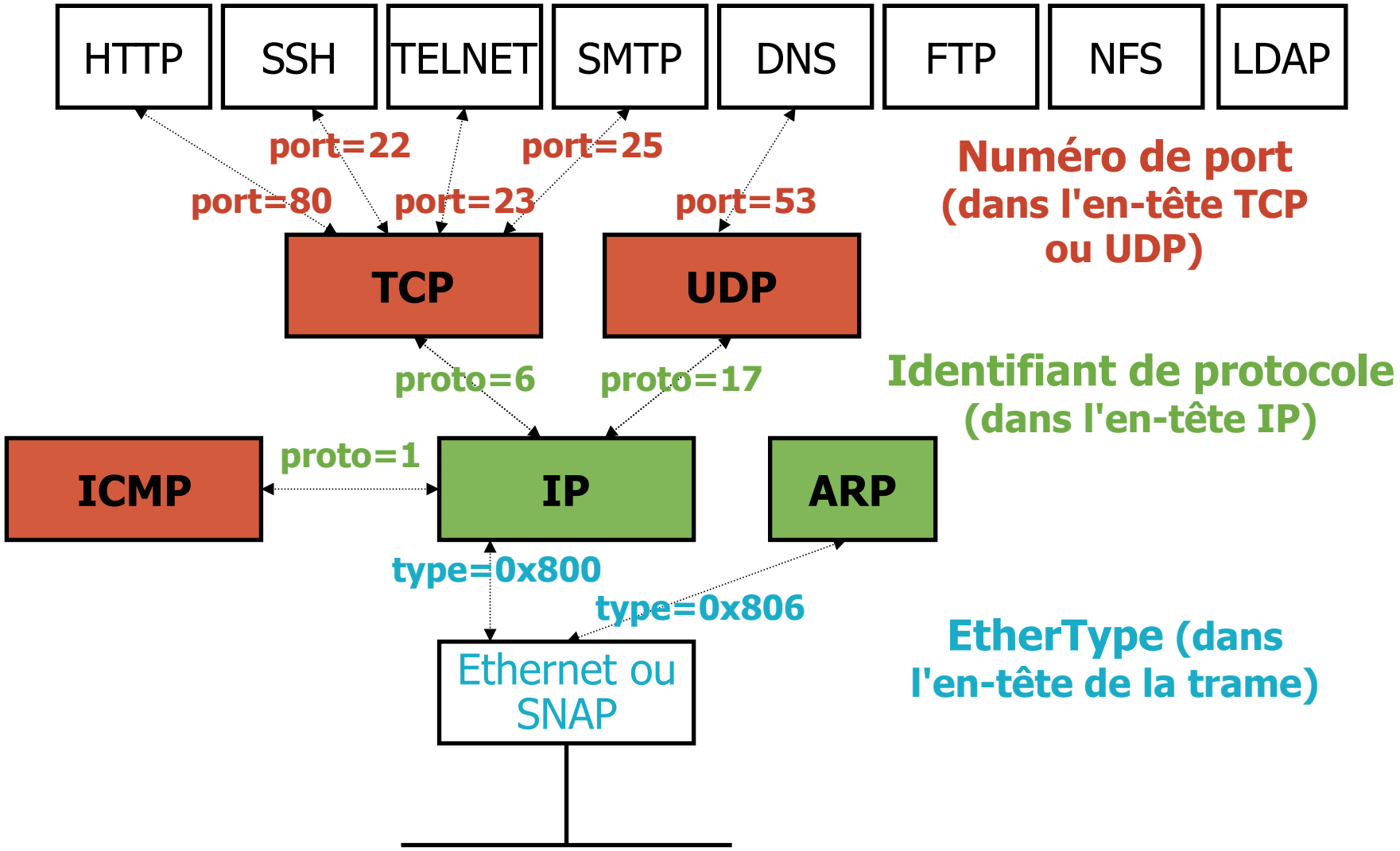


Format de l'en-tête IPv4

- Un en-tête de 20 octets + une partie facultative de longueur variable (options)

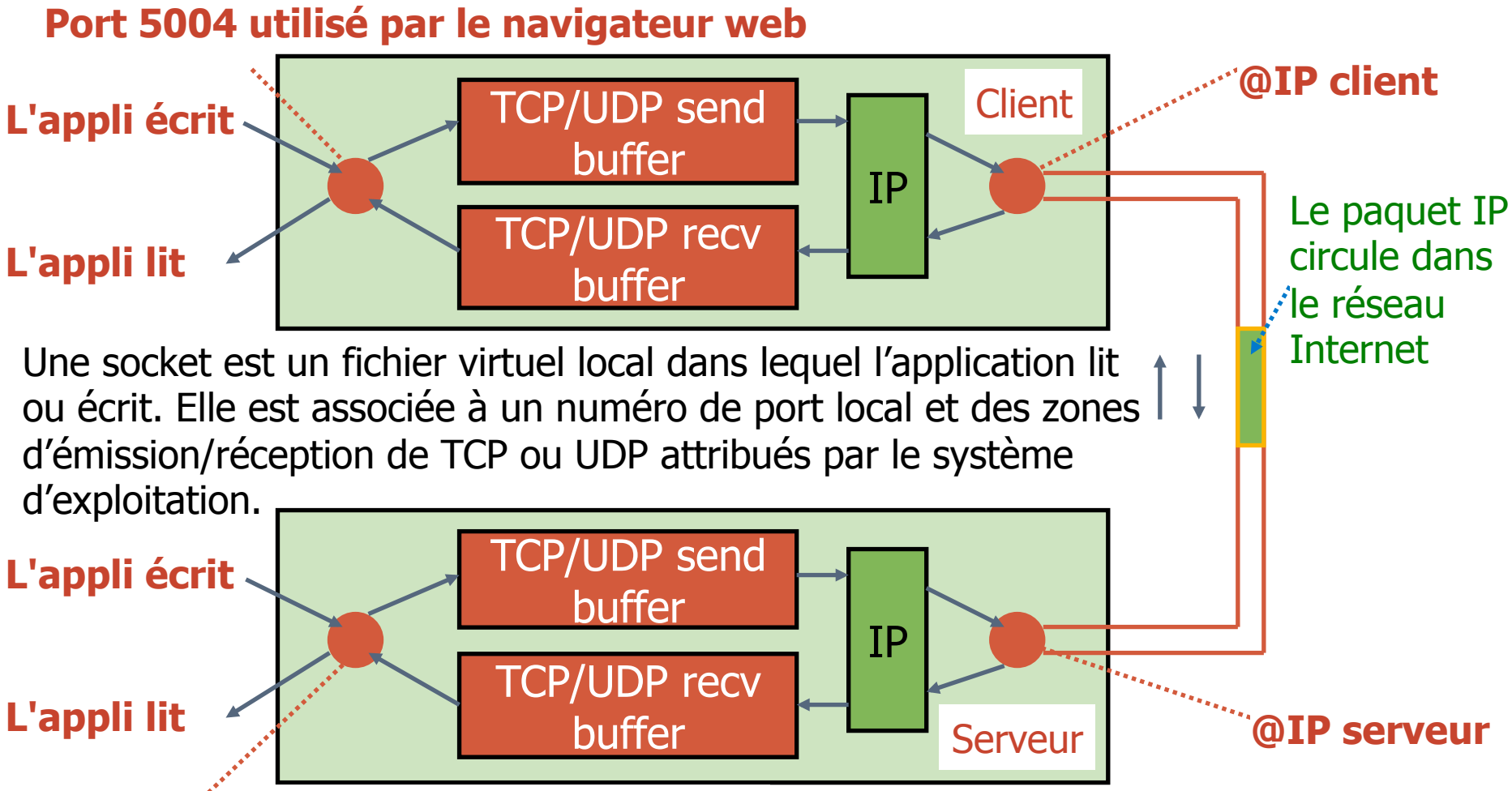


La désencapsulation : identifier la couche >

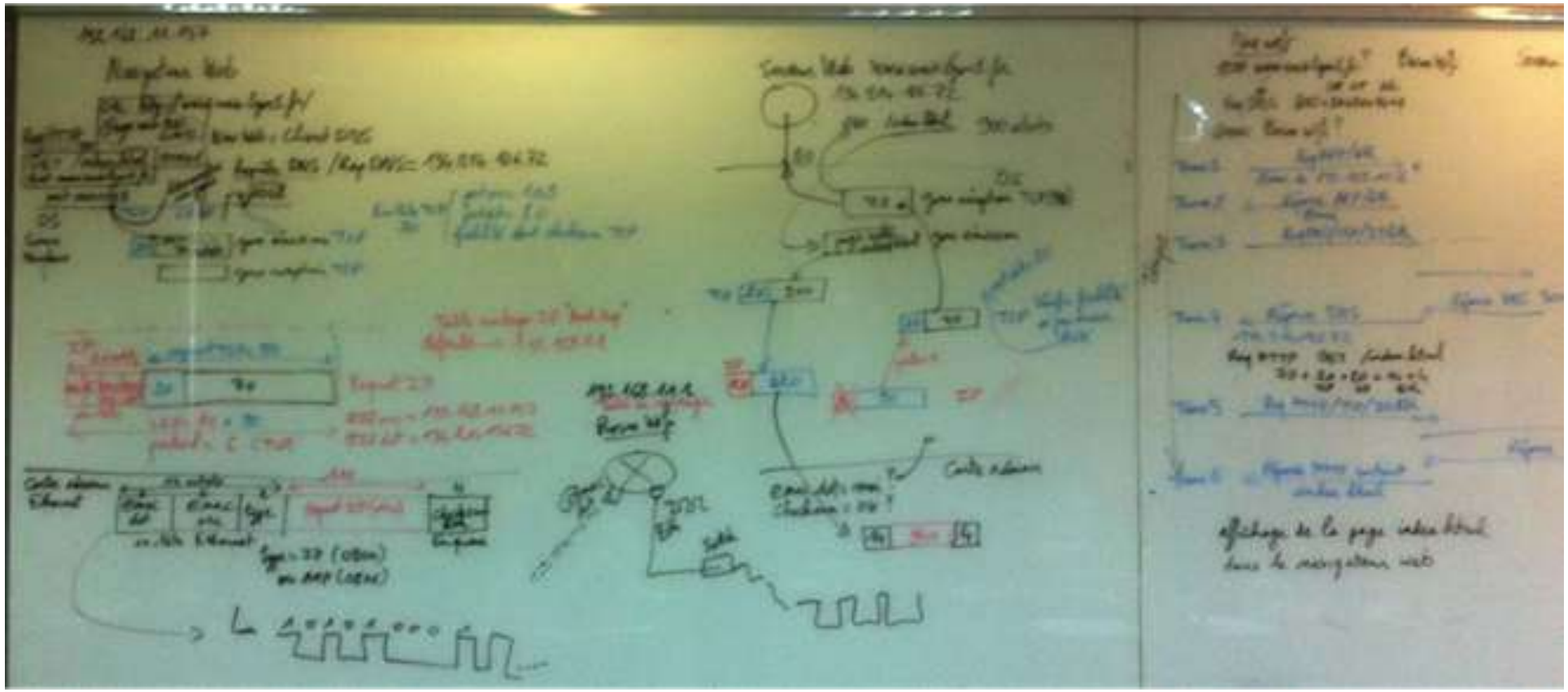


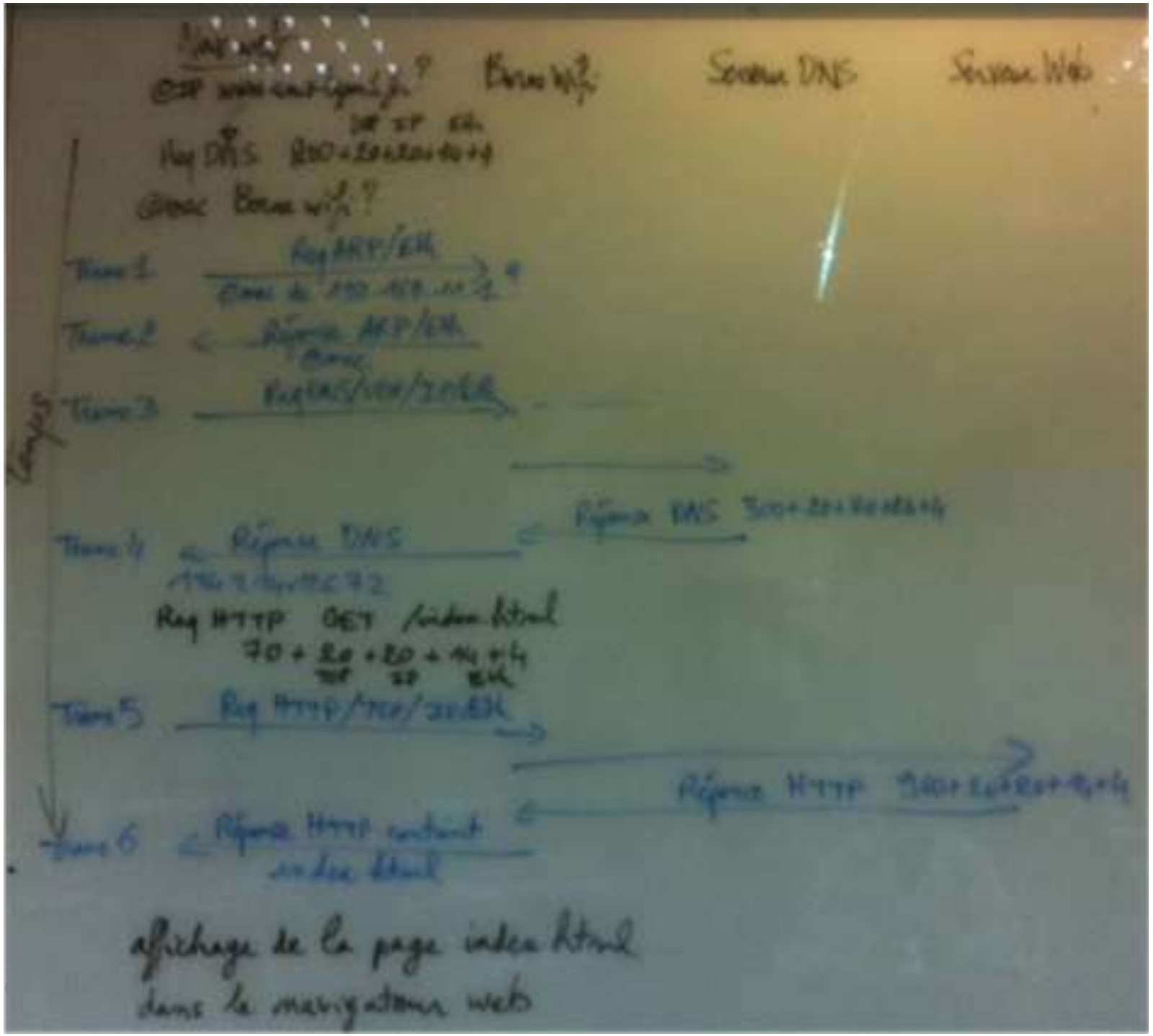
Les sockets et les numéros de port

- Un échange Client/Serveur = **(@IP_src,port_src,@IP_dest,port_dest)**



A la fin du module !




Serveur DNS
Serveur Web
Temps
Temps 1
Temps 2
Temps 3
Temps 4
Temps 5
Temps 6
Req HTTP GET /index.html
70 + 20 + 20 + 14
TCP IP Eth
affichage de la page index.html
dans le navigateur web