

# Sommaire

---

## Sommaire

- Définitions
- Adressage physique
- Adressage logique
- Transmission d'un message
- Trame Ethernet
- ARP
- IP et ICMP
- UDP / TCP
- DNS
- TCP
- Routage

## TP

- Adresses MAC
- Adresses IP
- ARP
- Ping
- DNS
- NTP
- SMTP

# Définitions

---

**Réseau** : ensemble d'équipements informatiques reliés entre eux et qui échangent des données.

**Topologie** : étude des propriétés des figures géométriques. Par extension : **étude de l'architecture des réseaux.**

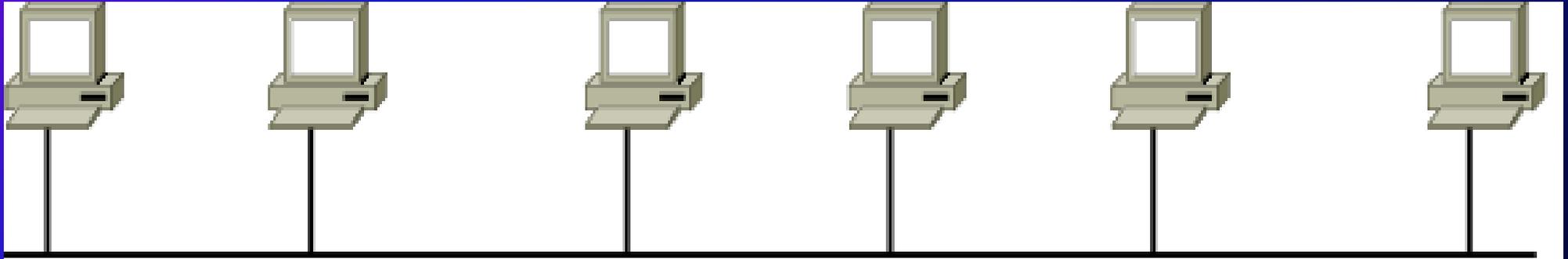
# Réseau P2P

- Pair à pair (p2p). C'est le plus petit réseau possible



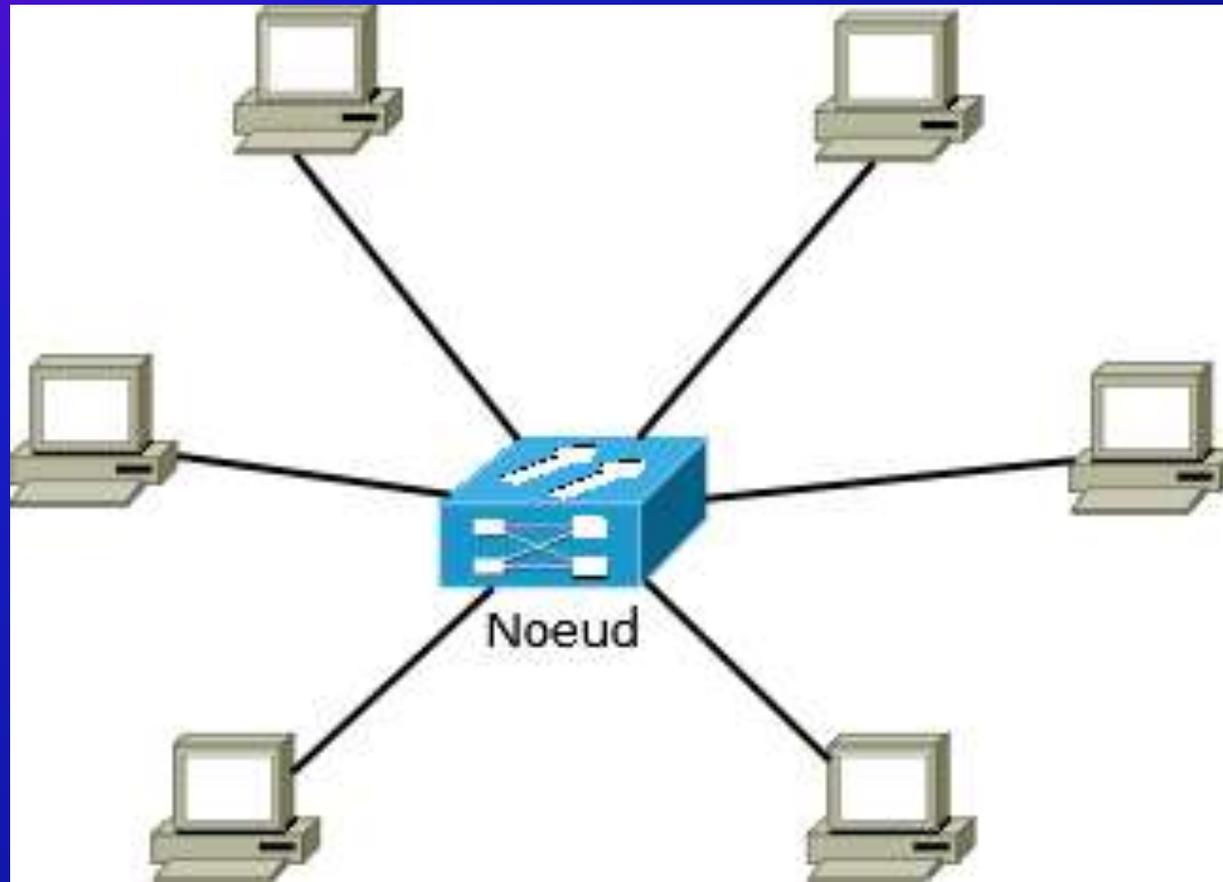
# Réseau bus

- En bus. L'architecture la plus ancienne.



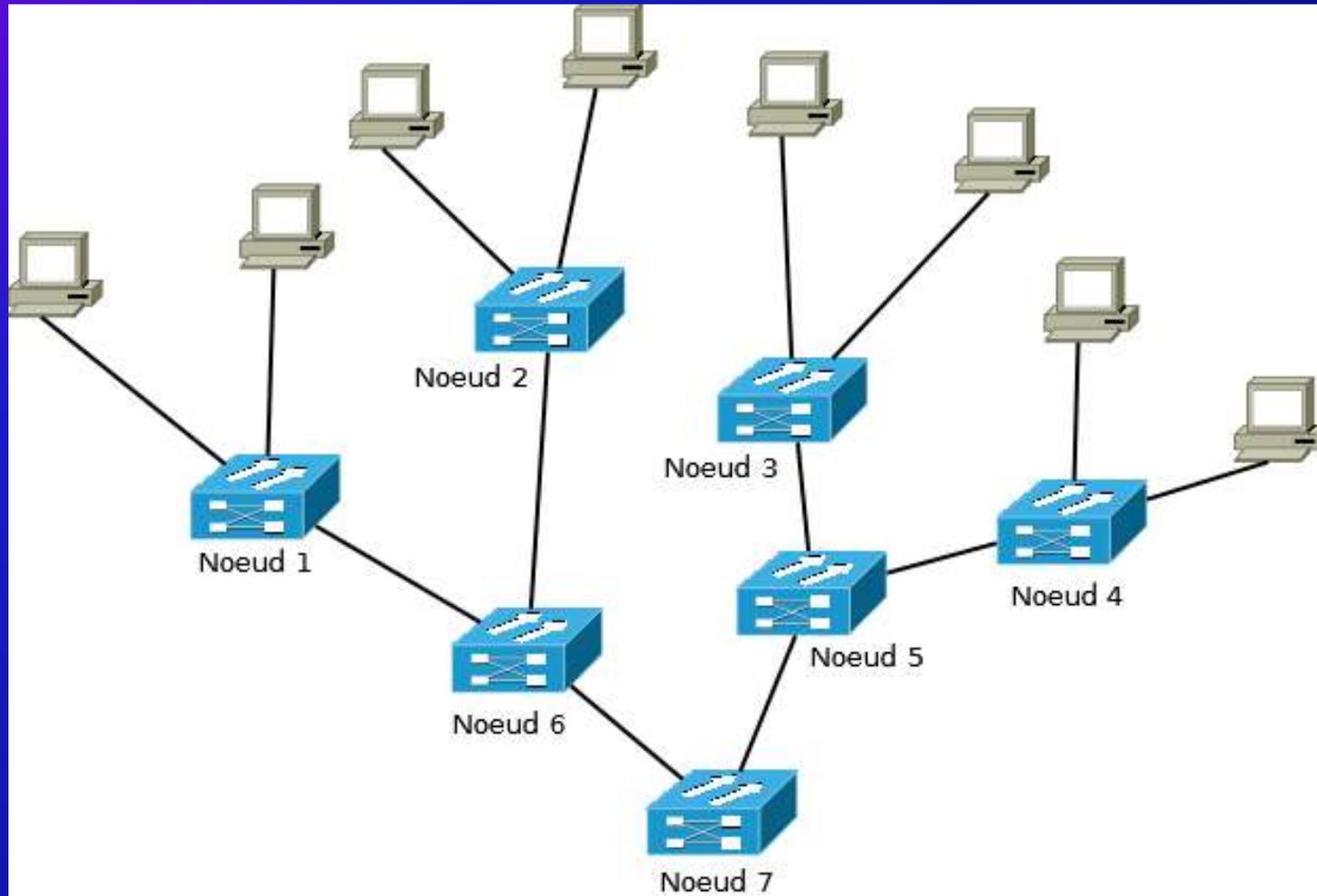
# Réseau en étoile

- En étoile



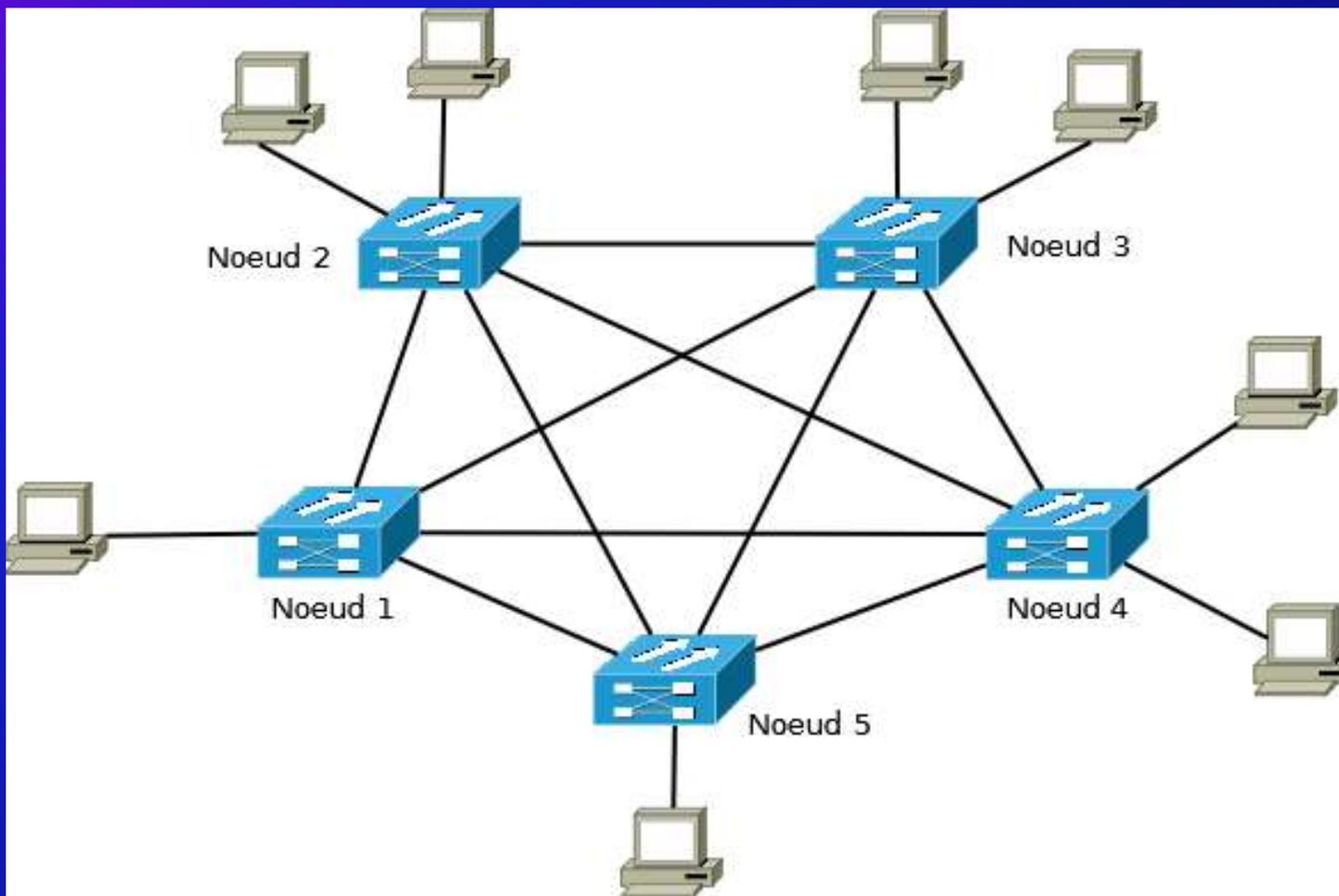
# Réseau arborescent

- En arbre. Architecture classique des réseaux locaux et métropolitains



# Réseau maillé

- Maillé. La perte d'un lien n'empêche pas le réseaux de fonctionner

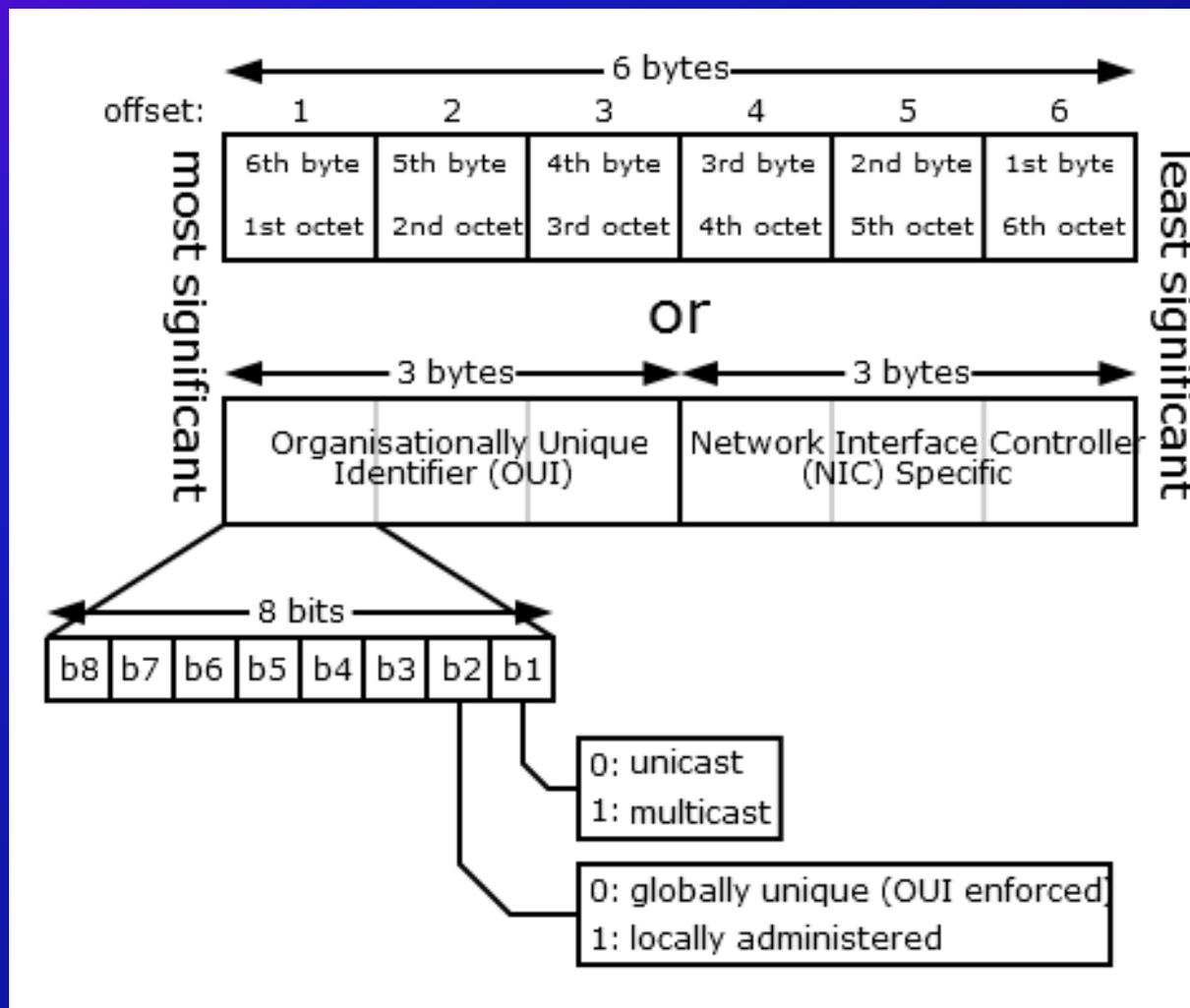


# Adresse physique

- Pour pouvoir adresser un message d'un point à un autre il faut savoir qui est l'émetteur et qui est le récepteur.
- Il n'y a toujours qu'un seul émetteur et celui-ci est identifié par une adresse forcément unique codée sur six octets : l'adresse MAC (Media Access Control).
  - Les trois premiers octets sont attribués par IANA (*Internet Assigned Numbers Authority*) au constructeur de l'équipement de réseau.
  - Les trois derniers octets sont un numéro de série attribué par le constructeur lui même à son équipement
- Pour le destinataire trois cas se présentent :
  - Mode **unicast** : un seul récepteur. L'adresse de destination est l'adresse MAC du récepteur.
  - Mode **multicast** : plusieurs récepteurs. Les données ne sont émises qu'une seule fois et seront acheminées vers toutes les machines du groupe de diffusion sans que le contenu ne soit dupliqué; c'est donc le réseau qui se charge de reproduire les données. Les adresse commençant par 01:00:5e sont réservées à cette utilisation.
  - Mode **broadcast** : tous les équipements connectés reçoivent le message. Dans ce cas l'adresse de destination est ff:ff:ff:ff:ff:ff

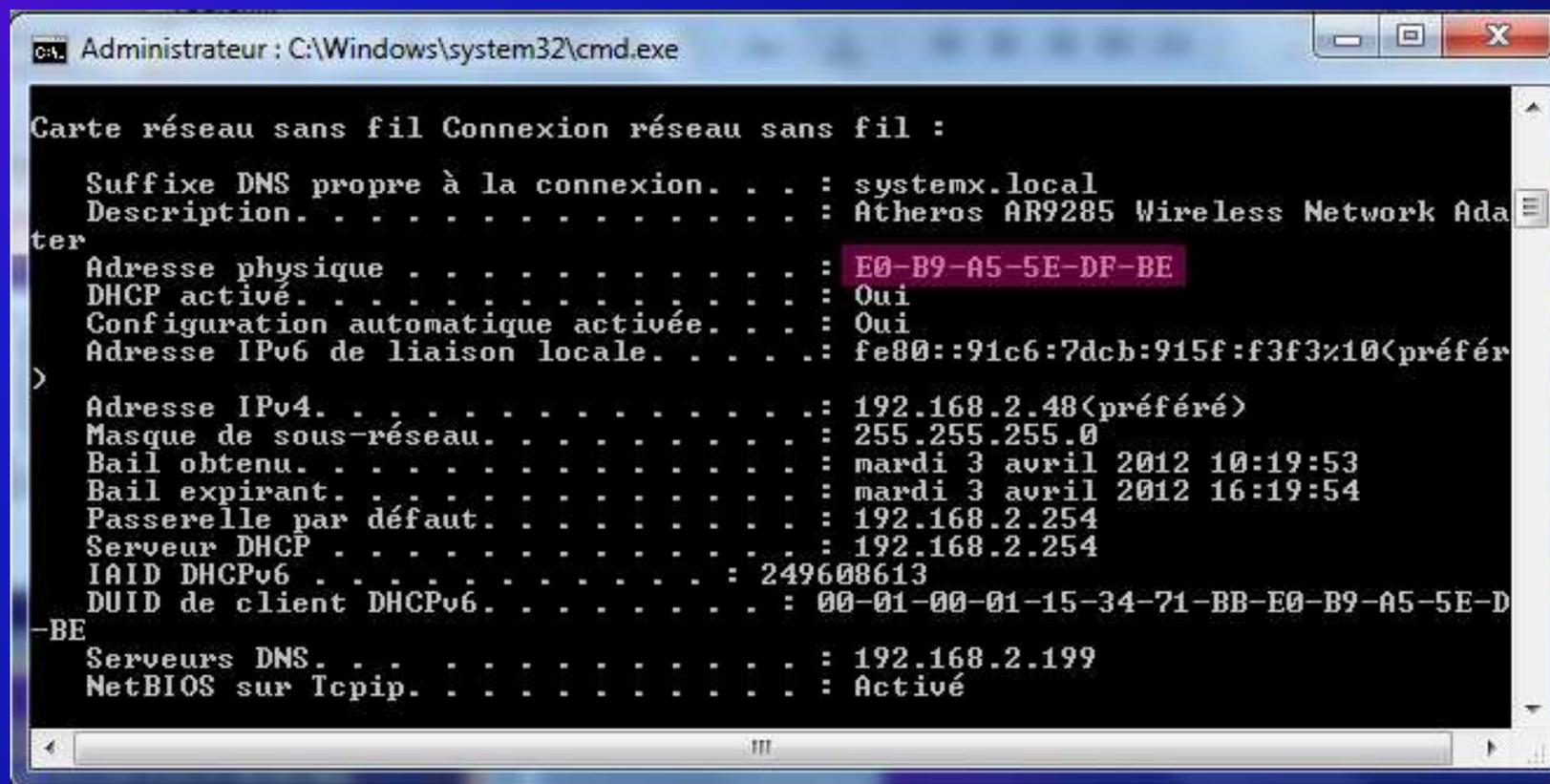
# Adresse MAC

- Normalisation



# Adresse MAC

- Identifier le constructeur et le numéro de série de votre carte réseau :
  - Sous Windows, en mode commande : `ipconfig /all`
  - Sous Linux / FreeBSD : `ifconfig` (éventuellement `-a`)
- Dans ce cas `e0 b9 a5` est Azurewave. Le numéro de série est `5e df be`



```
Administrateur : C:\Windows\system32\cmd.exe

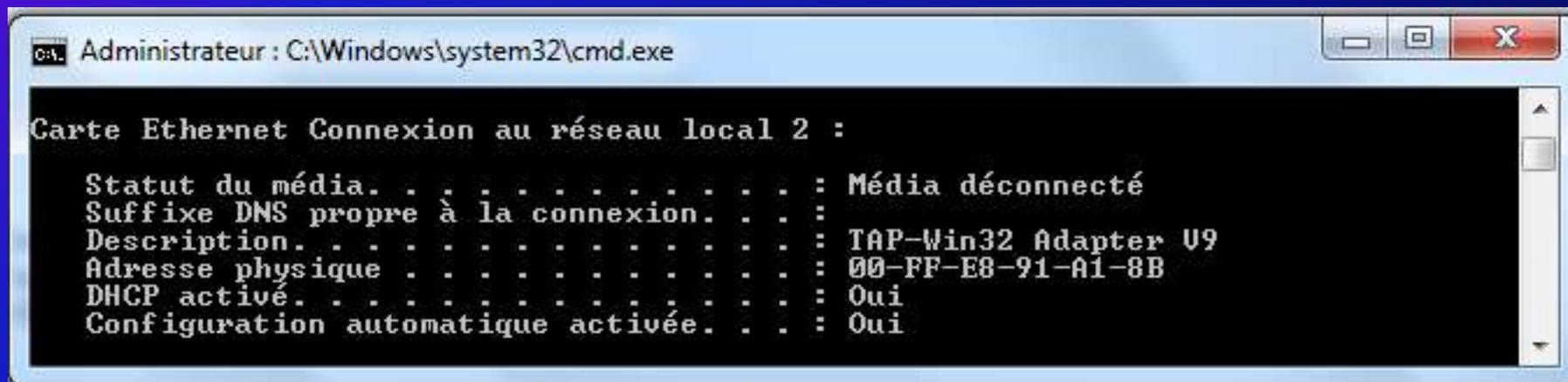
Carte réseau sans fil Connexion réseau sans fil :

    Suffixe DNS propre à la connexion. . . . : systemx.local
    Description. . . . . : Atheros AR9285 Wireless Network Adapter
    Adresse physique . . . . . : E0-B9-A5-5E-DF-BE
    DHCP activé. . . . . : Oui
    Configuration automatique activée. . . . : Oui
    Adresse IPv6 de liaison locale. . . . . : fe80::91c6:7dcb:915f:f3f3%10(préféré)

    Adresse IPv4. . . . . : 192.168.2.48(préféré)
    Masque de sous-réseau. . . . . : 255.255.255.0
    Bail obtenu. . . . . : mardi 3 avril 2012 10:19:53
    Bail expirant. . . . . : mardi 3 avril 2012 16:19:54
    Passerelle par défaut. . . . . : 192.168.2.254
    Serveur DHCP . . . . . : 192.168.2.254
    IAID DHCPv6 . . . . . : 249608613
    DUID de client DHCPv6. . . . . : 00-01-00-01-15-34-71-BB-E0-B9-A5-5E-D
    -BE
    Serveurs DNS. . . . . : 192.168.2.199
    NetBIOS sur Tcpi. . . . . : Activé
```

# Adresse MAC

- Vérifier, pour votre interface réseau, que les deux bits de poids faible du premier octet sont conformes à la norme.
- Montrer que Microsoft ne respecte pas la norme (c'est une mauvaise habitude pas étonnante chez eux) pour l'interface ci-dessous.



```
Administrateur : C:\Windows\system32\cmd.exe

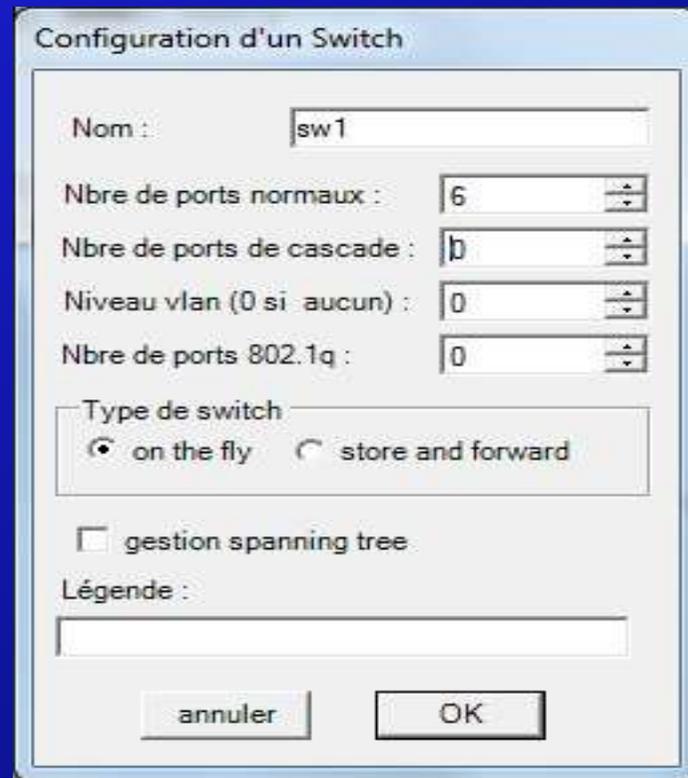
Carte Ethernet Connexion au réseau local 2 :

Statut du média. . . . . : Média déconnecté
Suffixe DNS propre à la connexion. . . . . :
Description. . . . . : TAP-Win32 Adapter V9
Adresse physique . . . . . : 00-FF-E8-91-A1-8B
DHCP activé. . . . . : Oui
Configuration automatique activée. . . . . : Oui
```

- Quels devraient être les trois premiers octets de cette interface ?

# Adresse MAC TP SR3

- Créer un réseau composé de 4 stations (st1 à st4) connectées à un switch. On gardera les adresses MAC par défaut (mac01 à macc04) des stations, ce sera plus facile à manipuler même si ce n'est pas conforme à la norme !
- Configuration du switch : supprimer tout ce qui concerne les VLANs, les ports de cascade, le spanning tree et passer le switch en mode « on the fly »

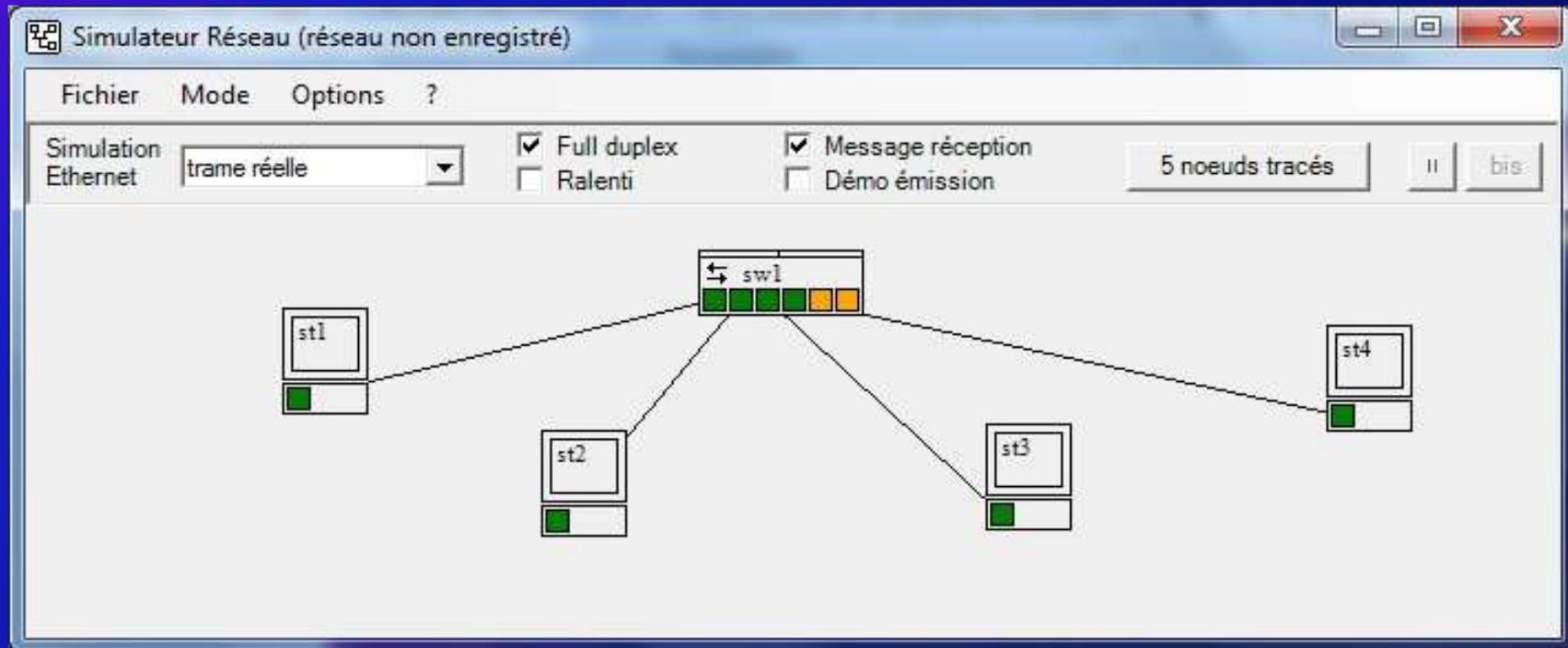


The image shows a dialog box titled "Configuration d'un Switch" with the following fields and options:

- Nom : sw1
- Nbre de ports normaux : 6
- Nbre de ports de cascade : 0
- Niveau vlan (0 si aucun) : 0
- Nbre de ports 802.1q : 0
- Type de switch:
  - on the fly
  - store and forward
- gestion spanning tree
- Légende : (empty text box)
- Buttons: annuler, OK

# Adressage MAC TP SR3

- Passer en mode Ethernet
  - Trame réelle
  - Full duplex
  - Message réception
- Tracer les 5 nœuds



# Adressage MAC TP SR3

1. Vider la table MAC/Port du switch
2. Envoyer une trame de st1 à st3 (unicast)
  - a. Comment st1 trouve st3 ?
  - b. Quel est le contenu de la table MAC/Port du switch ?
3. Faire découvrir le réseau au switch
  - a. Quel est le maintenant contenu de la table MAC/Port du switch ?
  - b. Envoyer une trame de st1 à st3
    - Quel est le nouveau comportement ?
4. Vider à nouveau la table MAC/Port du switch
  - a. Éditer manuellement la table afin que l'envoi d'une trame de st3 à st2 se fasse sans broadcast.
  - b. Sans modification manuelle de la table comment envoyer une trame de st2 à st4 sans broadcast

# Adressage logique

Les adresses MAC étant difficiles à manipuler il a été décidé de leur associer des adresses logiques qui facilitent la représentation de la topologie du réseau. Ces adresses sont associées au protocole IP (Internet Protocol) que l'on verra plus loin.

Les adresses logiques sont codées sur 4 octets (ce qui est curieux car il y a moins d'adresses IP que d'adresses MAC) : octet1.octet2.octet3.octet4.

Il y a donc  $256^4$  adresses soit un peu moins de 4.3 milliards.

Par exemple : 82.234.79.107 (en décimal)

Le protocole associant les adresses MAC et les adresses IP est **ARP**

(**A**ddress **R**esolution **P**rotocol)

Enfin on associe à une adresse IP un nom choisi arbitrairement selon de nombreuses règles contraignantes. Ces associations sont gérées par le protocole

**DNS** (**D**omain **N**ame **S**ystem)

Quelle est l'adresse IP de votre machine (commande `ipconfig` sous Windows et `ifconfig` sous Linux) et quel est son nom (`hostname` sous Linux) ?

# Adresse logique

- Votre PC mémorise les associations adresse MAC / adresse logique.  
Commande : `arp -a` (la même chose sous Windows et Linux ??? ...)



```
Administrateur : C:\Windows\system32\cmd.exe
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\Bernard>arp -a

Interface : 192.168.2.48 --- 0xa
  Adresse Internet      Adresse physique      Type
  192.168.2.49          00-16-17-ac-a6-be    dynamique
  192.168.2.199         00-1e-2a-ba-4a-86    dynamique
  192.168.2.254         54-52-00-26-19-31    dynamique
  192.168.2.255         ff-ff-ff-ff-ff-ff    statique
  224.0.0.22            01-00-5e-00-00-16    statique
  224.0.0.252           01-00-5e-00-00-fc    statique
  239.255.255.250      01-00-5e-7f-ff-fa    statique
  255.255.255.255      ff-ff-ff-ff-ff-ff    statique

Interface : 192.168.56.1 --- 0x14
  Adresse Internet      Adresse physique      Type
  192.168.56.255       ff-ff-ff-ff-ff-ff    statique
  224.0.0.22            01-00-5e-00-00-16    statique
  224.0.0.252           01-00-5e-00-00-fc    statique
  239.255.255.250      01-00-5e-7f-ff-fa    statique

C:\Users\Bernard>
```

# Adresse IP

- En septembre 1993 est publiée la RFC 1519 « *Classless Inter-Domain Routing* » (CIDR) qui introduit la notion de masque de réseau. La RFC 4632 d'août 2006 rend obsolète la précédente RFC. Dorénavant, à chaque adresse, est associé un masque de réseau codé sur 4 octets représentant une suite continue de 1 de longueur variable partant de l'octet de gauche. Cette technique permet d'avoir, avec un seul identifiant, une adresse réseau et une adresse d'équipement.
- Exemple :
  - 11111111.11111111.00000000.00000000 = 255.255.0.0      **noté /16**
  - 11111111.11111111.11111100.00000000 = 255.255.252.0      **noté /22**
  - 11111111.11111111.11111111.00000000 = 255.255.255.0      **noté /24**
  - 11111111.11111111.11111111.11111000 = 255.255.255.248      **noté /29**
- Ainsi les adresses sont maintenant codées comme suit :
  - 192.168.1.101/22
  - 192.168.1.101/24

Ces adresses, bien qu'identiques, auront un « comportement » différent que nous analyserons ultérieurement.

# Adresse IP

- Le masque de réseau (netmask) permet, en le combinant à l'adresse IP, d'extraire deux éléments de celle-ci :
  - l'adresse de réseau
  - l'adresse de l'équipement.

L'adresse de réseau peut être considéré (en première approche) comme la zone de laquelle ne sortiront pas les broadcasts.

L'adresse de l'équipement est le numéro de l'équipement sur le réseau

L'addition de l'adresse de réseau et de l'adresse de l'équipement donne l'adresse complète.

```
Exemple :      Réseau      192.168.  1.  0
                +   Équipement  0.  0.  0.101
                -----
                192.168.  1.101
```

# Adresse IP

- Par exemple on a l'adresse 192.168.1.101/24 soit :  
IP            11000000.10101000.00000001.01100101  
Masque       11111111.11111111.11111111.00000000
- **Adresse du réseau** en appliquant un ET logique entre l'adresse IP et le masque  
IP            11000000.10101000.00000001.01100101    **ET**  
Masque       11111111.11111111.11111111.00000000  
Network      11000000.10101000.00000001.00000000 = 192.168.1.0
- **Adresse de l'équipement** en appliquant un ET logique entre l'adresse IP et le complément à 1 du masque  
IP            11000000.10101000.00000001.01100101    **ET**  
**C<sub>1</sub> Masque** 00000000.00000000.00000000.11111111  
Host           00000000.00000000.00000000.01100101 = 0.0.0.101  
**Adresse complète** = Network + host  
Network + host = 192.168.1.0 + 0.0.0.101 = 192.168.1.101

# Adresse IP

- On remarque que lorsque tous les bits d'un octet du masque sont à 1 soit :

$$1111\ 1111 = 255_{10}$$

$$C_1\ 0000\ 0000 = 0$$

Le calcul du réseau et de l'équipement est très simple puisque :

$$X\ \text{ET}\ 255 = X$$

$$X\ \text{ET}\ 0 = 0$$

Donc pour 192.168.1.101/24

$$192.168.1.101\ \text{ET}\ 255.255.255.0 = 192.168.1.0$$

$$192.168.1.101\ \text{ET}\ 0.0.0.255 = 0.0.0.101$$

# Adresse IP

- Il y a deux adresses réservées qui ne peuvent pas être attribuées à un équipement
  - **Network** soit le host 0, dans notre cas **192.168.1.0**
  - **Broadcast** : tous les bits de host sont à 1. Cela revient à prendre le  $C_1$  du masque de réseau soit  $0.0.0.11111111 = 0.0.0.255$ .  
L'adresse de broadcast est :  
**Broadcast** = Network +  $C_1$  masque =  $192.168.1.0 + 0.0.0.255 = 192.168.1.255$
- Le premier équipement est **network + 1** = 192.168.1.1
- Le dernier équipement est **broadcast - 1** = 192.168.1.254
- Si on note n le nombre de bits à 1 du masque de réseau le nombre maximum d'équipements possibles est de  $2^{32-n}-2$  soit dans notre cas  $2^{32-24}-2=254$

# Adresse IP

---

- Exercice : donner le netmask, les adresses de network, de broadcast, les adresses mini et maxi des équipements ainsi que le nombre maximum possible d'équipements pour les adresses CIDR suivantes
  - 10.0.0.0/8
  - 10.24.0.0/16
  - 192.168.8.0/24

# Adresse IP

- 10.0.0.0/8    00001010.00000000.00000000.00000000
- Netmask      11111111.00000000.00000000.00000000 = **255.0.0.0**
- C<sub>1</sub> netmask 00000000.11111111.11111111.11111111 = 0.255.255.255
- Network      **10.0.0.0**
- Broadcast
  - Équipement = 0.255.255.255
  - Broadcast = 10.0.0.0 + 0.255.255.255 = **10.255.255.255**
- Host mini = 10.0.0.0 + 0.0.0.1 = **10.0.0.1**
- Host maxi = 10.255.255.255 – 0.0.0.1 = **10.255.255.254**
- Nombre de hosts =  $2^{32-8}-2$  = **16 777 214**

# Adresse IP

- 10.24.0.0/16 00001010. 00011000.00000000.00000000
- Netmask 11111111.11111111.00000000.00000000 = **255.255.0.0**
- C<sub>1</sub> netmask 00000000.00000000.11111111.11111111 = 0.0.255.255
- Network **10.24.0.0**
- Broadcast
  - Équipement = 0.0.255.255
  - Broadcast = 10.24.0.0 + 0.0.255.255 = **10.24.255.255**
- Host mini = 10.24.0.0 + 0.0.0.1 = **10.24.0.1**
- Host maxi = 10.24.255.255 – 0.0.0.1 = **10.24.255.254**
- Nombre de hosts =  $2^{32-16}-2 =$  **65 534**

# Adresse IP

- 192.168.8.0/24      11000000.10101000.00001000.00000000
- Netmask            11111111.11111111.00000000.00000000 = **255.255.255.0**
- C<sub>1</sub> netmask        00000000.00000000.00000000.11111111 = 0.0.0.255
- Network      **192.168.8.0**
- Broadcast
  - Équipement = 0.0.0.255
  - Broadcast = 192.168.8.0 + 0.0.0.255 = **192.168.8.255**
- Host mini = 192.168.8.0 + 0.0.0.1 = **192.168.8.1**
- Host maxi = 192.168.8.255 – 0.0.0.1 = **192.168.8.254**
- Nombre de hosts =  $2^{32-24}-2 = \mathbf{254}$

# Adresse IP

Cas des masques qui ne sont pas multiples de 8. par exemple l'adresse CIDR suivante : 192.168.1.15/22

IP                    11000000.10101000.00000001.00001111 = 192.168.1.15

Masque            11111111.11111111.11111100.00000000 = 255.255.252.0

$C_1$  masque    00000000.00000000.00000011.11111111 = 0.0.3.255

**Network = IP ET Masque = 192.168.0.0**

**Broadcast = IP +  $C_1$  Masque = 192.168.0.0 + 0.0.3.255 = 192.168.3.255**

**Host = IP ET  $C_1$  Masque = 0.0.1.15**

**Host mini = Network + 1 = 192.168.1.1**

**Host maxi = Broadcast – 1 = 192.168.3.254**

On peut avoir  $2^{32-22}-2=1022$  équipements.

# Adresse IP

- En conclusion les caractéristiques de l'adresse 192.168.1.15/22 sont :
  - Network 192.168.0.0
  - Netmask 255.255.252.0
  - Broadcast 192.168.3.255
  - Host min 192.168.0.1
  - Host max 192.168.3.254

Remarque 1 : les adresses 192.168.0.15, 192.168.1.15, 192.168.2.15 et 192.168.3.15 correspondent à des équipements d'adresses 0.0.0.15, 0.0.1.15, 0.0.2.15 et 0.0.3.15 qui appartiennent toutes au réseau 192.168.0.0.

Remarque 2 : les adresses de network doivent être des puissances de 2. Les réseaux suivants seraient 192.168.4.0, 192.168.8.0 etc.

# Adresse IP

---

- Exercice : donner le netmask, les adresses de network, d'équipement, de broadcast, les adresses mini et maxi des équipements ainsi que le nombre maximum possible d'équipements pour les adresses CIDR suivantes
  - 192.168.5.2/23
  - 172.16.12.13/12
  - 82.234.79.106/30

# Adresse IP

- 192.168.5.2/23            11000000. 10101000.00000101.00000010
- Netmask                    11111111.11111111.11111110.00000000 = **255.255.254.0**
- C<sub>1</sub> netmask                00000000.00000000.00000001.11111111 = 0.0.1.255
- Network            **192.168.4.0**
- Équipement **0.0.1.2**
- Broadcast = 192.168.4.0 + 0.0.1.255 = **192.168.5.255**
- Host mini = 192.168.4.0 + 0.0.0.1 = **192.168.4.1**
- Host maxi = 192.168.5.255 – 0.0.0.1 = **192.168.5.254**
- Nombre de hosts =  $2^{32-23}-2 = \mathbf{510}$

# Adresse IP

- 172.16.12.13/12      10101100.00010000.00001100.00001101
- Netmask              11111111.11110000.00000000.00000000 = **255.240.0.0**
- C<sub>1</sub> netmask        00000000.00001111.11111111.11111111 = 0.15.255.255
- Network      **172.16.0.0**
- Équipement **0.0.12.13**
- Broadcast = 172.16.0.0 + 0.15.255.255 = **172.31.255.255**
- Host mini = 172.16.0.0 + 0.0.0.1 = **172.16.0.1**
- Host maxi = 172.31.255.255 – 0.0.0.1 = **172.31.255.254**
- Nombre de hosts =  $2^{32-12}-2 =$  **1 048 574**

# Adresse IP

- 82.234.79.106/30      01010010.11101010.01001111.01101010
- Netmask              11111111.11111111.11111111.11111100 = **255.255.255.252**
- C<sub>1</sub> netmask          00000000.00000000.00000000.00000011 = 0.0.0.3
- Network      **82.234.79.104**
- Équipement **0.0.0.2**
- Broadcast = 82.234.79.104 + 0.0.0.3 = **82.234.79.107**
- Host mini = 82.234.79.104 + 0.0.0.1 = **82.234.79.105**
- Host maxi = 82.234.79.107 – 0.0.0.1 = **82.79.234.106**
- Nombre de hosts =  $2^{32-30}-2 = 2$

# Adresse IP privées

- La RFC 1918 précise plusieurs types d'adresses IP :
  - **publiques.** Elles sont uniques dans le monde. Par exemple 8.8.8.8 est un serveur de Google (pour l'instant ... ). Ces adresses sont attribuées par IANA (Internet Assigned Numbers Authority [www.iana.org](http://www.iana.org)) département de ICANN (Internet Corporation for Assigned Names and Numbers [www.icann.org](http://www.icann.org)) société privée de droit californien à but non lucratif.
  - **privées.** Elles sont uniques sur un même réseau privé mais deux réseaux privés peuvent utiliser la même adresse sur l'un et sur l'autre des réseaux. Elles sont choisies par l'administrateur du réseau local. Ces adresses sont :
    - 10.0.0.0 / 8 soit de 10.0.0.0 à 10.255.255.255
    - 172.16.0.0 / 12 soit de 172.16.0.0 à 172.31.255.255
    - 192.168.0.0 / 16 soit de 192.168.0.0 à 192.168.255.255

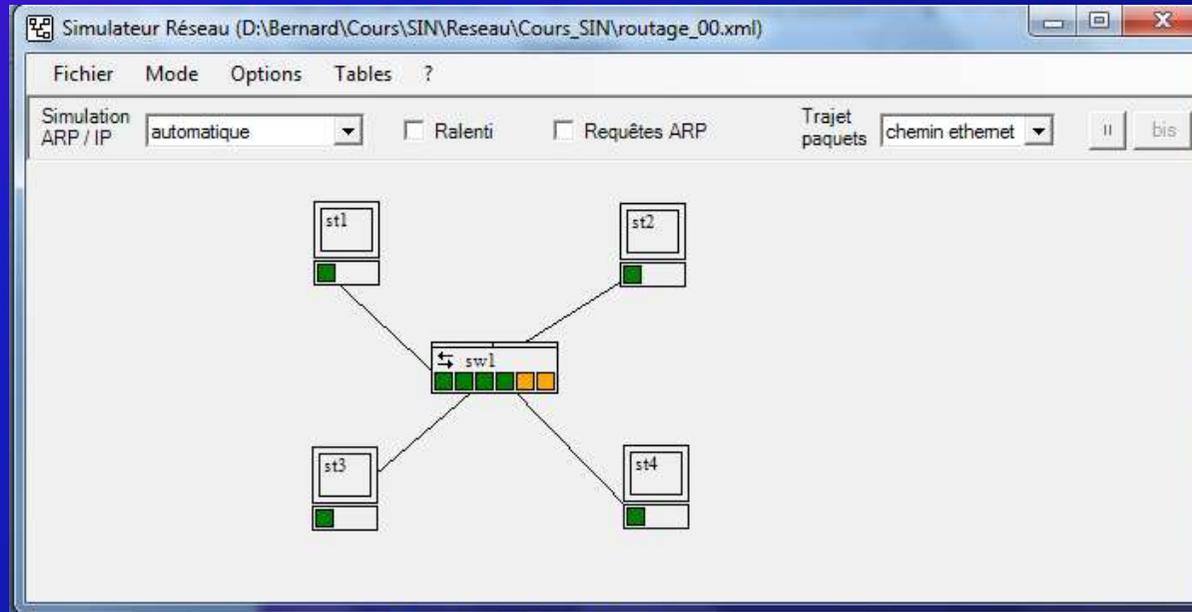
# Adresse IP réservées

- **Réseau par défaut**
  - 0.0.0.0 / 8
- **Bouclage** (loopback)
  - 127.0.0.0 / 8
- **Adresses locales auto configurées (Automatic Private Internet Protocol Addressing = APIPA** Utilisé essentiellement par Window\$ car ce système ne sait pas démarrer s'il n'y a pas d'interface réseau avec une adresse ...)
  - 169.254.0.0 / 16
- **Multicast**
  - 224.0.0.0 / 4
- **Broadcast**
  - 255.255.255.255 / 32

# TP Adresses IP / SR3

Créer le réseau suivant :

- Les deux PC du haut 192.168.1.1/24 et 192.168.1.2/24
- Les deux PC du bas 192.168.3.1/24 et 192.168.3.2/24



Faire un ping entre les PC du haut puis entre un PC du haut et un PC du bas  
Analyser le comportement

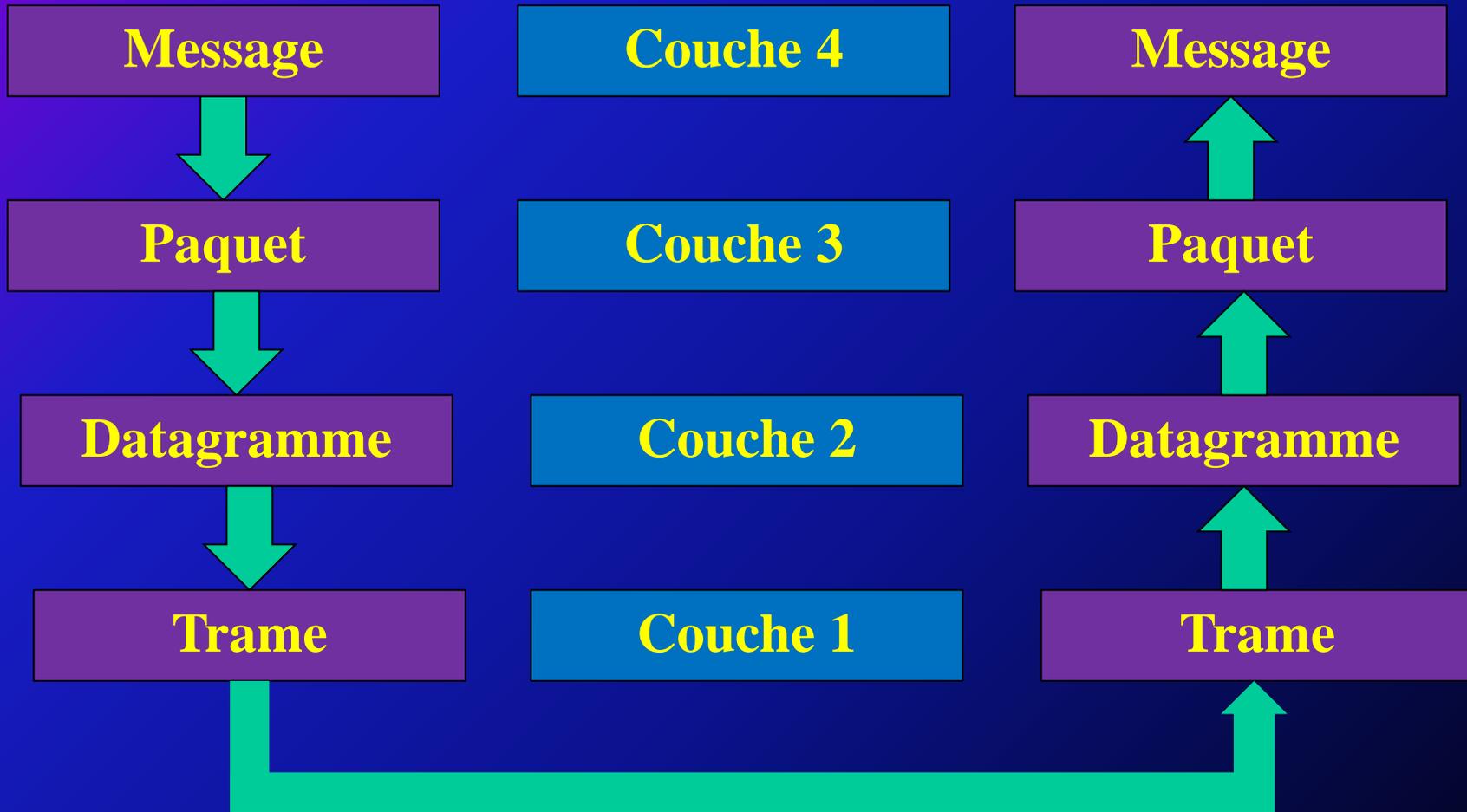
Quelle différence de comportement observez-vous si le netmask des adresses est maintenant 22.?

# Transmission d'un message

- Le message à transmettre est créé par une application de l'utilisateur (mail, web, etc.). Ce message étant de forme quelconque il ne peut pas être envoyé sur le réseau tel quel. Il est transmis à une application normalisée qui va le découper en paquets standardisés.
- Le **paquet** est mis en forme suivant les exigences d'un protocole (principalement TCP). Une fois mis en forme (ajout des informations TCP) le paquet de **données** est à nouveau transmis à une autre application normalisée qui va le mettre sous forme de datagramme.
- La **datagramme** est à son tour mise en forme suivant les exigences d'un autre protocole (principalement IP) avec l'ajout, entre autre, des **adresses IP** de l'émetteur et du récepteur. Une fois mis en forme (ajout des informations IP) il est transmis à une application qui va le mettre sous forme de trame
- La **trame** est mise au format normalisé Ethernet IEEE 802.3 avec, entre autre, l'ajout, des **adresses MAC** émetteur récepteur et enfin envoyée sur le lien physique
- Le processus inverse se déroule lors de la réception

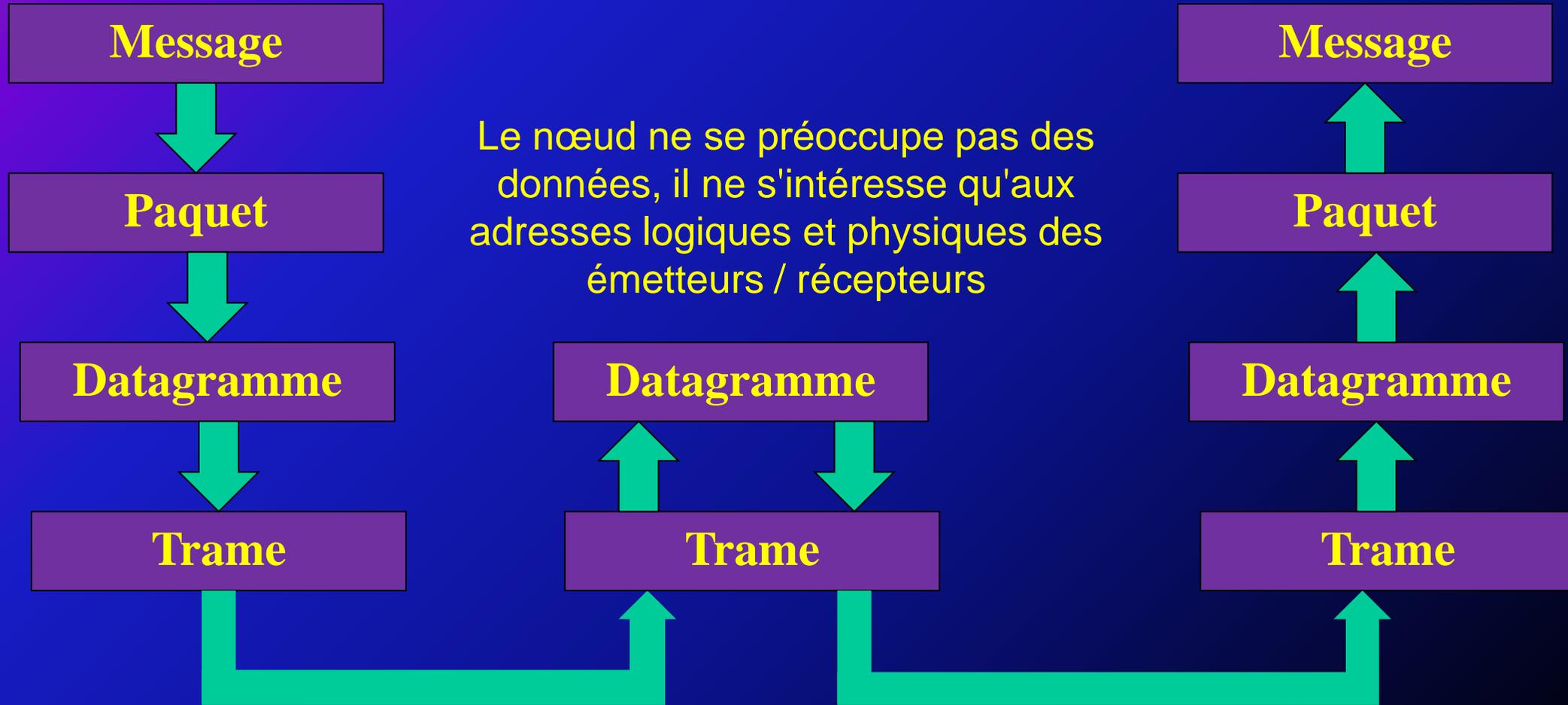
# Modèle en couches

- Modèle ISO/TCP/IP en 4 couches



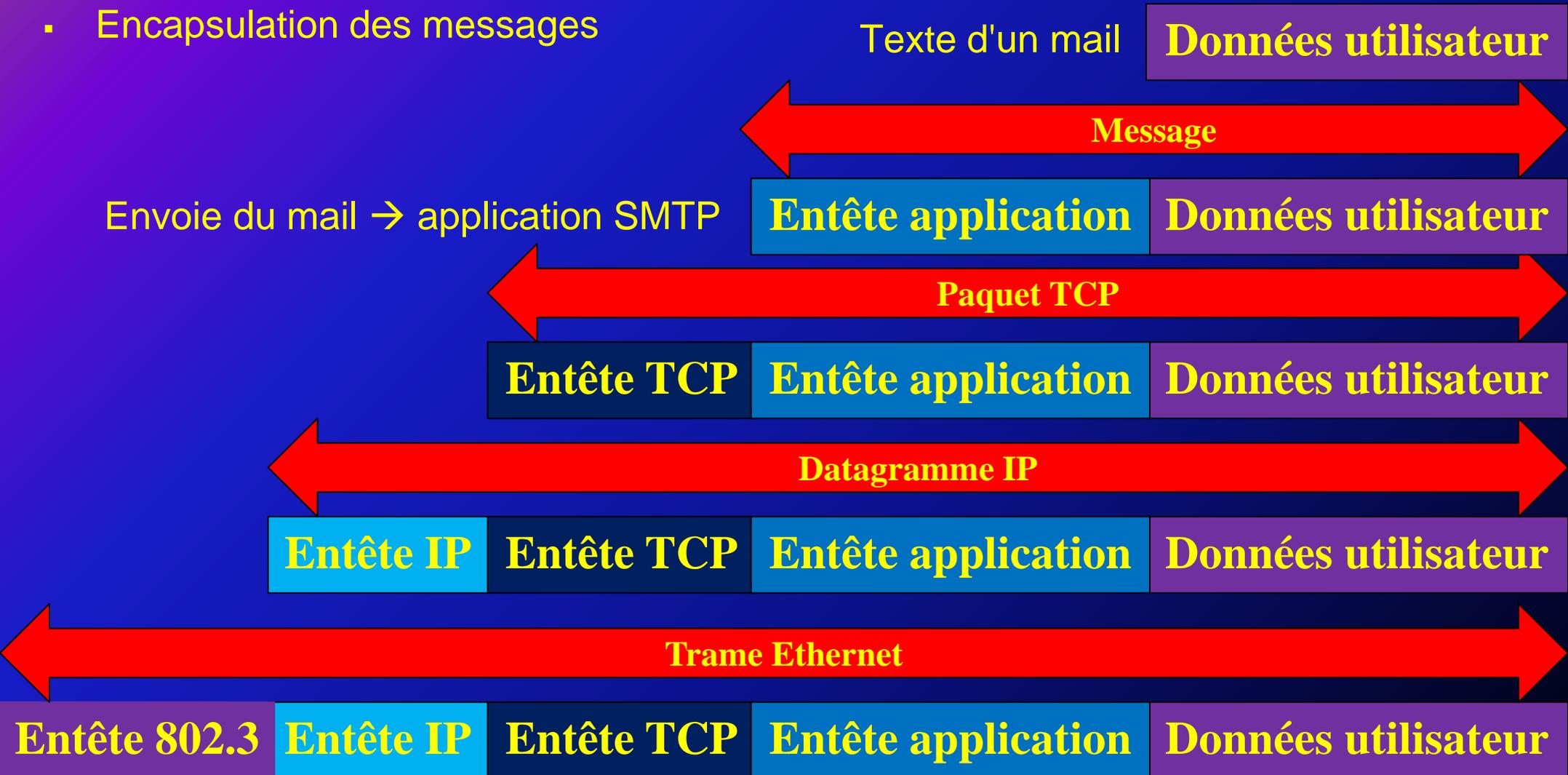
# Modèle en couches

- Acheminement d'un message en passant par un nœud



# Modèle en couches

- Encapsulation des messages



# Modèle en couches

La diapositive précédente est un cas assez courant mais il existe d'autres protocoles dans les couches du modèle TCP / IP. Par exemple :

- Couche 3
  - TCP (Transmission Control Protocol)
  - UDP (User Datagram Protocol)
  - ICMP (Internet Control Message Protocol)
- Couche 2
  - IP (Internet Protocol)
  - ARP (Address Resolution Protocol)
- Couche 1
  - Ethernet
  - Token-Ring

# Trame Ethernet

Il faut maintenant coder l'information et l'envoyer sur un support. Nous allons voir comment la mettre en forme à la mode Ethernet en envoyant une suite normalisée de bits représentant les 4 couches :

- Message
- Paquet
- Datagramme
- Trame

Nous commencerons par le bas de la couche, c'est à dire la trame qui sera transmise sur la liaison. C'est une suite de bits en format normalisé de longueur minimale 72 octets et de longueur maximale 1524 octets.

Il existe d'autres formats de trames mais le format Ethernet est le plus utilisé.

# Trame Ethernet

- Préambule 7 octets : alternance de 1 et de 0 (suite de aa<sub>16</sub>). Il permet à l'horloge du récepteur de se synchroniser sur celle de l'émetteur
- SOF (Start Of Frame) 1 octets : 10101011=ab<sub>16</sub>
- Adresse MAC destination 6 octets
- Adresse MAC source 6 octets
- Longueur ou Type 2 octets. Signification différente selon l'équipement
  - $\leq 1500_{10}$  Équipement de réseau (switch, routeur) : Longueur des données
  - $> 1500_{10}$  Utilisateur (client, serveur) : protocole du niveau supérieur. Exemples :
    - 800<sub>16</sub>=2048<sub>10</sub> IP
    - 806<sub>16</sub>=2054<sub>10</sub> ARP
- Données de 46 à 1500 octets. Si il y a moins de 46 octets de données ajout d'un champ de bourrage (padding) pour compléter (par des 0).
- FCS (Frame Check Sequence) 4 octets. Contient le CRC (Cyclic Redundancy Code) de la trame. La station réceptrice détecte si la trame est correcte et si c'est le cas la transmet à la couche supérieure. Dans le cas contraire la trame est perdue.

# Trame Ethernet

En résumé :

Sans compter le préambule et le SOF (8 octets) la longueur minimale de la trame est de 64 octets et maximale de 1518 octets

- 6 octets MAC destination
- 6 octets MAC source
- 2 octets longueur / type
- 46 octets mini, 1500 octets maxi données
- 4 octets FCS

Quelques problèmes :

- Si la trame est perdue il n'y a pas d'information transmise, c'est un protocole de niveau supérieur sur l'émetteur qui est chargé de la détection de non livraison
- S'il y a plus de 1500 octets de données celles-ci sont acheminées en plusieurs trames sans séquençement entre elles. Là aussi c'est la communication entre protocoles de niveau supérieur qui règle ce problème.

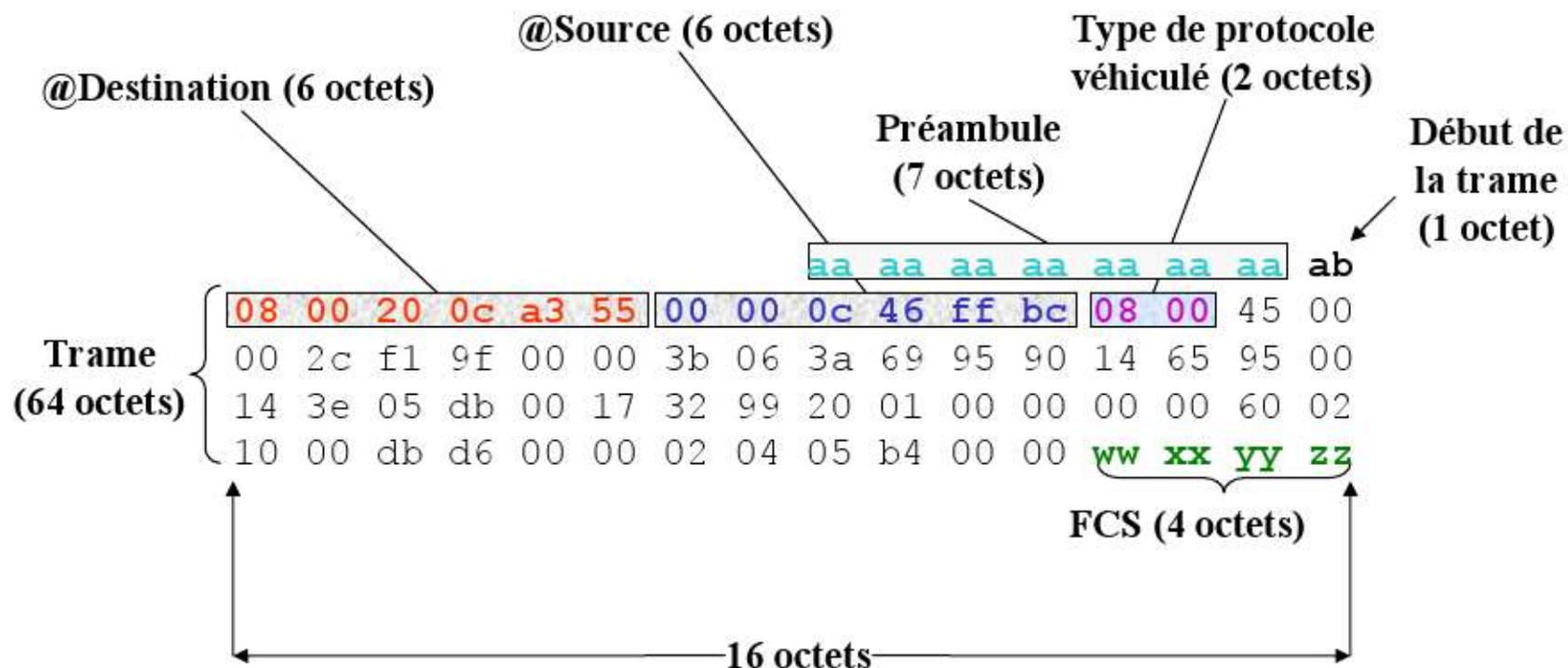
# Trame Ethernet

## Temps d'émission des trames

- 10 Base T
  - Fréquence = 10 Mb/s =  $10^7$  b/s
  - Période =  $10^{-7}$  s = 0.1  $\mu$ s
  - Soit pour envoyer 64 octets = 512 bits : 51.2  $\mu$ s (Slot Time normalisé)
  - Temps entre deux trames (Inter Frame Gap) : 9.6  $\mu$ s soit 96 bits (normalisé)
- 100 Base T
  - Fréquence = 100 Mb/s
  - Temps d'émission pour 64 octets : 5.12  $\mu$ s (normalisé)
  - Temps inter trame : 0.96  $\mu$ s (normalisé)

# Trame Ethernet

## Exemple de trame *Ethernet II*



# Protocole ARP

C'est le protocole qui va associer l'adresse MAC et l'adresse IP d'une station Ethernet. Il va donc être utilisé en permanence.

- Protocole de niveau 2 (même couche que IP). Se situe au niveau datagramme.
- Address Resolution Protocol, défini dans la RFC 826
- La résolution d'adresse est le mécanisme permettant a une station d'obtenir l'adresse MAC d'une station possédant une certaine adresse IP **dans le même réseau**. IP sur Ethernet utilise donc systématiquement ARP
- Fonctionnement (on note Src la source et Dst la destination):
  - Src envoie en broadcast une requête ARP signifiant qu'il souhaite obtenir l'adresse MAC associée a l'adresse IP de Dst. Message : « who as IP de Dst »
  - La requête est reçue et traitée par **toutes** les stations du réseau
  - Seule la station d'adresse IP Dst répond en envoyant en unicast à Src une réponse ARP contenant l'adresse MAC demandée

# Datagramme ARP

- Exemple de visualisation de ARP avec WireSark (fichier ping\_vm.cap)

The screenshot shows the Wireshark interface with a packet capture filter set to 'arp.proto.type'. The packet list pane displays four packets, with the first one selected. The packet details pane shows the structure of the ARP request, including Ethernet II, Address Resolution Protocol (request), and the specific MAC and IP addresses involved.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CadmusCo_00:c4:9b	Broadcast	ARP	42	who has 192.168.56.101? Tell 192.168.56.1
2	0.000385	CadmusCo_f2:84:bc	CadmusCo_00:c4:9b	ARP	42	192.168.56.101 is at 08:00:27:f2:84:bc
7	4.997287	CadmusCo_f2:84:bc	CadmusCo_00:c4:9b	ARP	42	who has 192.168.56.1? Tell 192.168.56.101
8	4.997308	CadmusCo_00:c4:9b	CadmusCo_f2:84:bc	ARP	42	192.168.56.1 is at 08:00:27:00:c4:9b

**Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)**  
Ethernet II, Src: CadmusCo\_00:c4:9b (08:00:27:00:c4:9b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
Address Resolution Protocol (request)  
Hardware type: Ethernet (1)  
Protocol type: IP (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: request (1)  
[Is gratuitous: False]  
Sender MAC address: cadmusCo\_00:c4:9b (08:00:27:00:c4:9b)  
Sender IP address: 192.168.56.1 (192.168.56.1)  
Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00)  
Target IP address: 192.168.56.101 (192.168.56.101)

```
0000  ff ff ff ff ff ff 08 00 27 00 c4 9b 08 06 00 01  .....8.
0010  08 00 06 04 00 01 08 00 27 00 c4 9b c0 a8 38 01  .....8.
0020  00 00 00 00 00 00 c0 a8 38 65  .....8e
```

# Trame Ethernet : entête 802.3

---

- Remarque : Le préambule, le SOF et le FCS (7 + 1 + 4 octets) ne sont pas capturés par les logiciels d'analyse de réseau.
- La trame comprend l'entête Ethernet soit 14 octets + le datagramme ARP soit 28 octets. La longueur totale de la trame faisant 42 octets

# Trame Ethernet : entête 802.3 / ARP demande

## Entête 802.3 (14 octets)

- MAC destination ff ff ff ff ff ff (broadcast)
- MAC source 08 00 27 00 c4 9b (mon PC)
- Protocole du niveau supérieur 08 06 (ARP)

# Datagramme ARP demande

- Taille : 28 octets pour IP
- Type de réseau physique 00 01 (Ethernet)
- Résolution pour quel protocole 08 00 (IP)
- Longueur de l'adresse physique 06 (6 octets)
- Longueur de l'adresse logique 04 (4 octets)
- Opération 00 01 (demande)
- MAC source 08 00 27 00 c4 9b
- IP Source c0 a8 38 01 (192.168.56.1)
- MAC destination 00 00 00 00 00 00 (inconnue)
- IP destination c0 a8 38 65 (192.168.56.101)

La trame sera de la forme :

Mac dst ff:ff:ff:ff:ff:ff

Mac src 08:00:27:00:c4:9b

Datagramme ARP demande

# Trame Ethernet : entête 802.3 / ARP réponse

## Entête 802.3 (14 octets)

- MAC destination 08 00 27 00 c4 9b (mon PC)
- MAC source 08 00 27 f2 84 bc (la cible)
- Protocole du niveau supérieur 08 06 (ARP)

# Datagramme ARP réponse

- Type de réseau physique 00 01 (Ethernet)
- Adresse pour quel protocole 08 00 (IP)
- Longueur de l'adresse physique 06
- Longueur de l'adresse logique 04
- Opération 00 02 (réponse)
- MAC source 08 00 27 f2 84 bc
- IP Source c0 a8 38 65 (192.168.56.101)
- MAC destination 08 00 27 00 c4 9b
- IP destination c0 a8 38 01 (192.168.56.1)

## Remarque

Bien que ce soit inutile au bon déroulement des opérations le destinataire va demander l'adresse MAC de l'émetteur afin de la mémoriser.

La trame sera de la forme :

Mac dst 08:00:27:00:c4:9b

Mac src 08:00:27:f2:84:bc

Datagramme ARP réponse

# La commande « arp »

- Afficher la table des combinaisons MAC / IP connues : `arp -a`
- Associer une adresse MAC à une adresse IP : `arp -s IP MAC`



```
Administrateur : C:\Windows\system32\cmd.exe

C:\Users\Bernard>arp -a

Interface : 192.168.2.48 --- 0xa
  Adresse Internet      Adresse physique      Type
  192.168.2.49          00-16-17-ac-a6-be    dynamique
  192.168.2.199         00-1e-2a-ba-4a-86    dynamique
  192.168.2.254         54-52-00-26-19-31    dynamique
  192.168.2.255         ff-ff-ff-ff-ff-ff    statique
  224.0.0.22            01-00-5e-00-00-16    statique
  224.0.0.252           01-00-5e-00-00-fc    statique
  239.255.255.250       01-00-5e-7f-ff-fa    statique
  255.255.255.255       ff-ff-ff-ff-ff-ff    statique

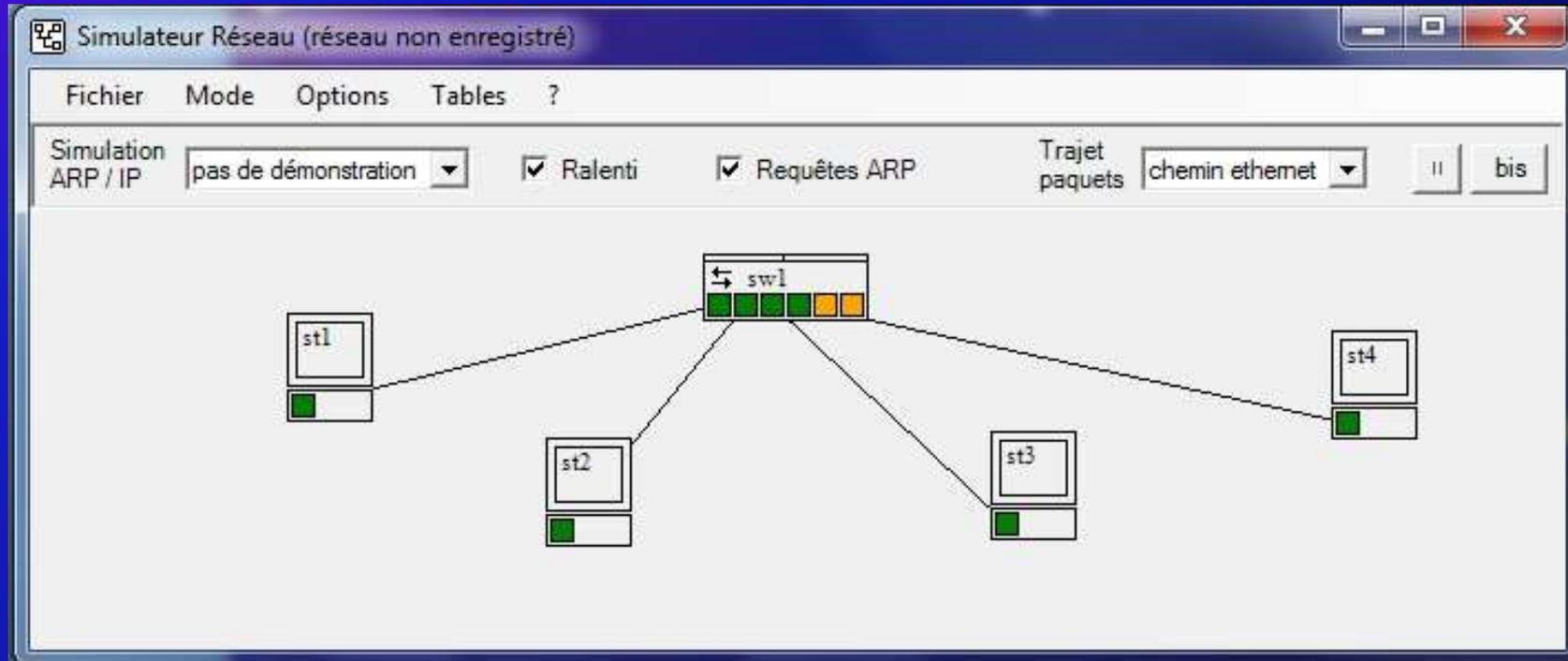
Interface : 192.168.56.1 --- 0x14
  Adresse Internet      Adresse physique      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff    statique
  224.0.0.22            01-00-5e-00-00-16    statique
  224.0.0.252           01-00-5e-00-00-fc    statique
  239.255.255.250       01-00-5e-7f-ff-fa    statique

C:\Users\Bernard>
```

# TP ARP / PING SR3

Dans ce TP on analysera le fonctionnement de ARP. Le protocole n'étant pas directement accessible par une commande du système d'exploitation on utilisera la commande `ping` qui sert à savoir si une station est joignable par son adresse IP. Puisqu'une station est adressée par son adresse IP il faudra connaître son adresse MAC donc utiliser le protocole ARP.

- Créer un réseau avec des adresses IP 10.0.0.1, 2, 3 et 4 et un masque de 8 bits.



# TP ARP / PING SR3

1. Vérifier que le cache ARP de toutes les stations est vide
2. Depuis st1 faire un ping vers st2
  - a. Analyser comment st1 contacte st2
  - b. Contenu des tables ARP sur toutes les stations ? Explications ?
  - c. Quelle sont, en format simplifié, sur st1, les entêtes :
    - trame Ethernet ?
    - datagramme IP ?
3. Depuis st2 faire un ping vers st1
  - a. Différence de comportement par rapport aux précédents ping ?
  - b. Analyse
4. Depuis st3 faire un ping vers st1
  - a. Différence de comportement par rapport aux précédents ping ?
  - b. Analyse
5. Modifier les tables ARP nécessaires afin que un ping de st4 vers st2 ne provoque pas de broadcast

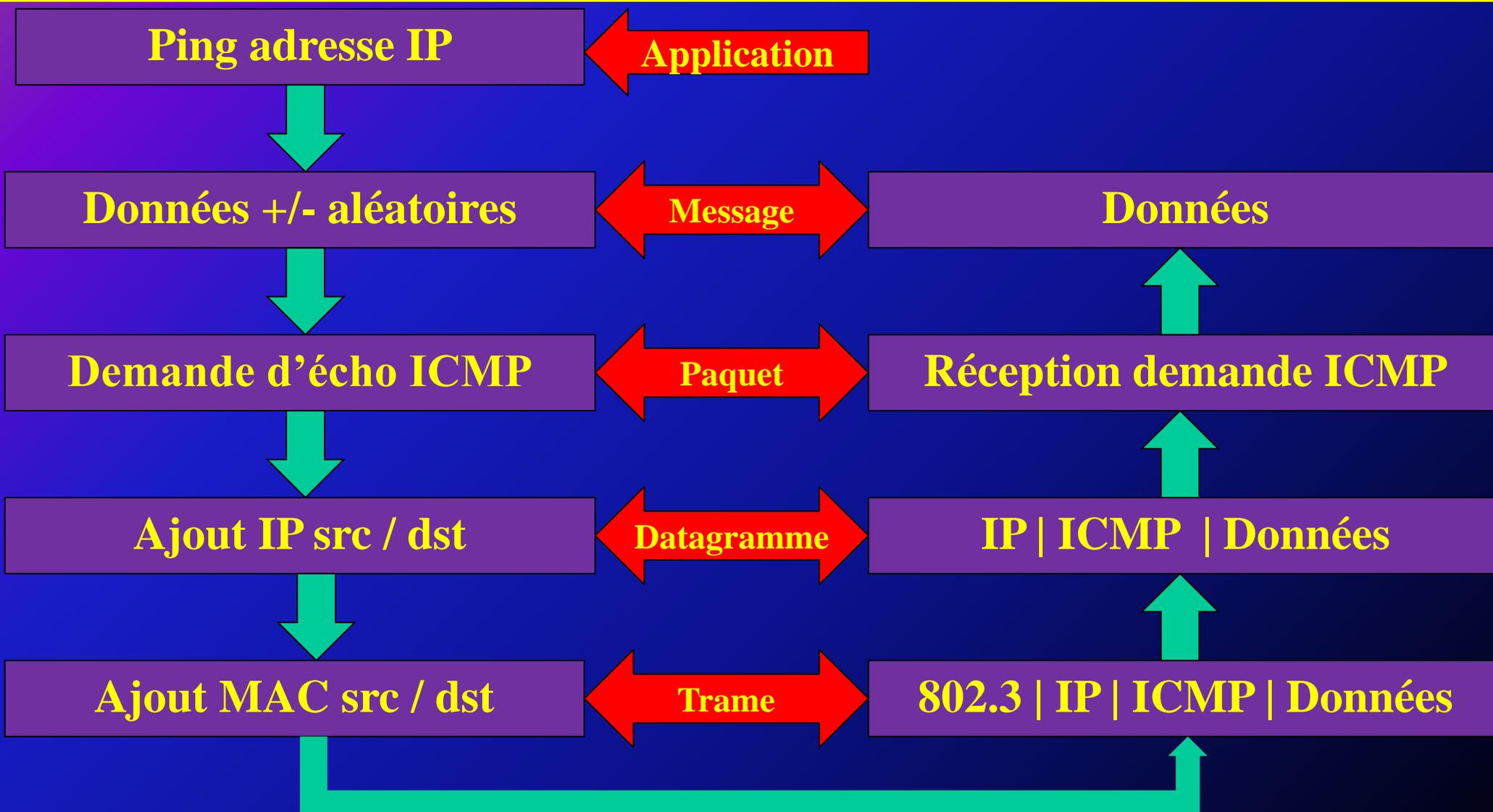
# TP ARP / PING « en vrai »

- Reprendre le TP précédent sur votre PC qui sera st1
- Vous utiliserez WireShark pour voir ce qui se passe sur l'interface réseau
- Dans le champ « filter » de WireShark vous pourrez indiquer « arp or icmp » afin de ne visualiser que ce qui nous intéresse.
- Vous mettre d'accord avec un(e) camarade pour faire des ping vers sa machine qui sera st2 et réciproquement
- Reprendre les questions 1 à 3
- Expliquer les différences de comportement de ARP par rapport au simulateur.

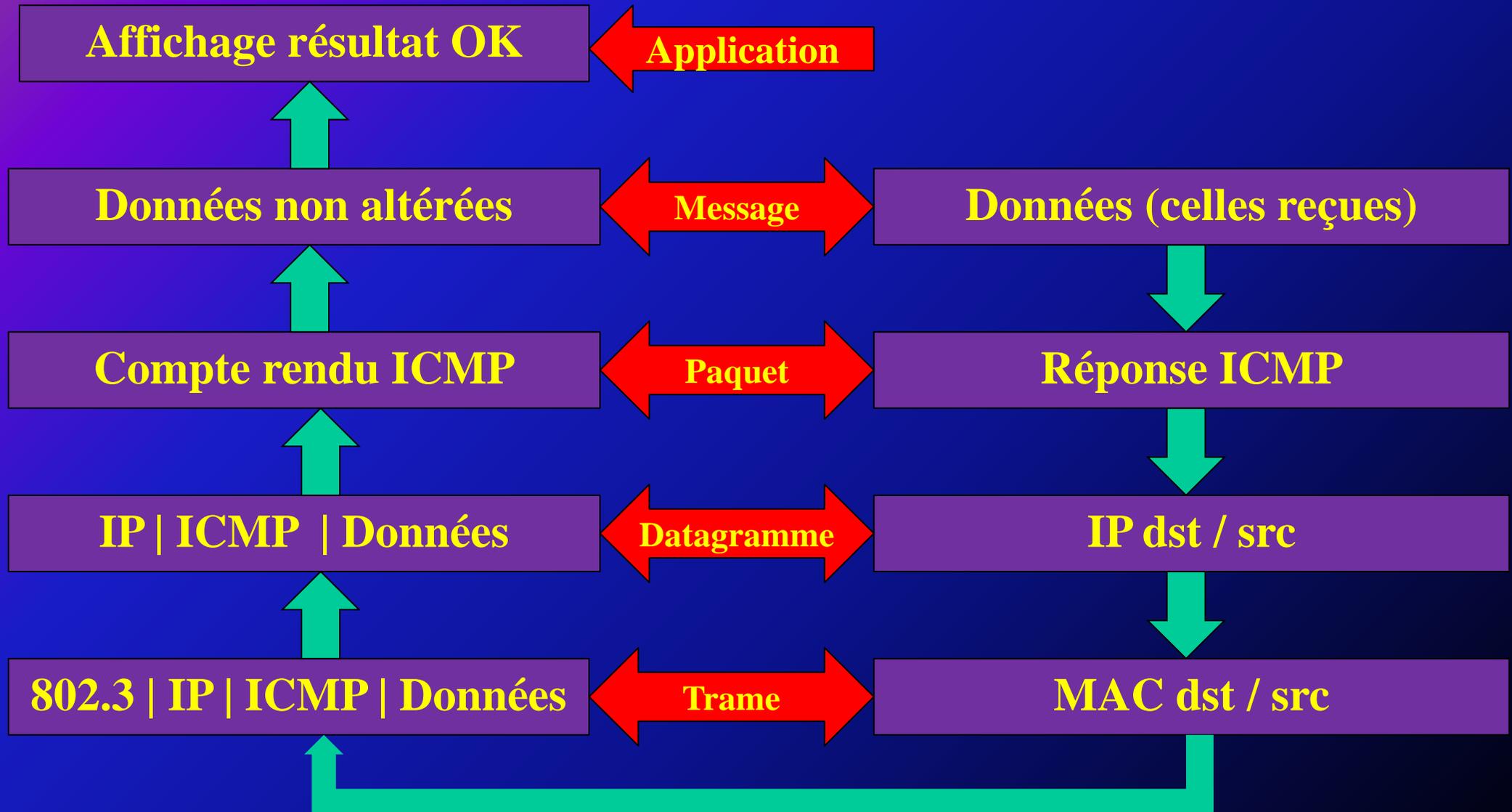
# La commande « ping » et le protocole ICMP

- La commande ping teste l'accessibilité de stations à travers un réseau. Cette commande met en œuvre le protocole ICMP : **I**nternet **C**ontrol **M**essage **P**rotocol.
- ICMP est un module obligatoire de IP qui assure deux fonctions principales :
  - Tester l'accessibilité d'une machine
  - Rendre compte d'un éventuel problème
- les messages ICMP sont de deux natures :
  - les messages d'interrogation / information définis par le type dans le premier octet de l'entête
    - Demande d'écho (8)
    - Réponse à une demande d'écho (0)
  - les messages d'erreur définis par le code dans le second octet de l'entête
    - Réseau inaccessible (0)
    - Station inaccessible (1)
    - Réseau de destination inconnu (6)
    - Station de destination inconnue (7)

# Demande d'écho



# Réponse à une demande d'écho



# Trame Ethernet ping

Exemple de de ICMP (ping\_vm2.pcap). Commande ping -n 1 192.168.1.26

The screenshot shows the Wireshark interface with a filter set to 'arp or icmp'. The packet list pane displays five packets:

No.	Time	Source	Destination	Protocol	Length	Info
2	3.927253	Azurewav_5e:df:be	Broadcast	ARP	42	who has 192.168.1.26? Tell 192.168.1.25
3	3.927728	CadmusCo_ae:38:3e	Azurewav_5e:df:be	ARP	42	192.168.1.26 is at 08:00:27:ae:38:3e
4	3.927741	192.168.1.25	192.168.1.26	ICMP	74	Echo (ping) request id=0x0001, seq=49/12544, ttl=128
5	3.927858	192.168.1.26	192.168.1.25	ICMP	74	Echo (ping) reply id=0x0001, seq=49/12544, ttl=64

The packet details pane for Frame 2 shows the following structure:

- Ethernet II, Src: Azurewav\_5e:df:be (e0:b9:a5:5e:df:be), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  - Destination: Broadcast (ff:ff:ff:ff:ff:ff)
    - Address: Broadcast (ff:ff:ff:ff:ff:ff)
      - .... ..1 .... = IG bit: Group address (multicast/broadcast)
      - .... ..1. .... = LG bit: Locally administered address (this is NOT the factory default)
    - Source: Azurewav\_5e:df:be (e0:b9:a5:5e:df:be)
      - Address: Azurewav\_5e:df:be (e0:b9:a5:5e:df:be)
        - .... ..0 .... = IG bit: Individual address (unicast)
        - .... ..0. .... = LG bit: Globally unique address (factory default)
      - Type: ARP (0x0806)
- Address Resolution Protocol (request)

The packet bytes pane shows the following hex data:

```
0000 ff ff ff ff ff ff e0 b9 a5 5e df be 08 06 00 01 ..... ^.....
0010 08 00 06 04 00 01 e0 b9 a5 5e df be c0 a8 01 19 ..... ^.....
0020 00 00 00 00 00 00 c0 a8 01 1a ..... ..
```

# Trame Ethernet ping

On observe que avant de pouvoir faire un ping vers 192.168.1.26 il faut trouver l'adresse MAC associée. Le processus se décompose donc en deux étapes :

- ARP
  - Broadcast « who as 192.168.1.26 » source > destination
  - Réponse destination > source
- ICMP
  - Demande d'écho à 192.168.1.26
  - Réponse de 192.168.1.26

# Trame Ethernet : entête 802.3 / IP

## Entête 802.3 (14 octets)

- MAC destination 08 00 27 ae 38 3e
- MAC source e0 b9 a5 5e df be
- Protocole du niveau supérieur 08 00 (IP)

# Datagramme IP: entête

- L'entête IP est définie comme suit :

0	4	8	16	19	24	31
Version	Lg entête	Service	Lg totale			
Numéro de paquet			drapeaux	Numéro de fragment		
Time To Live		proto.	CRC			
adresse Internet émetteur						
adresse Internet destinataire						
Options				bourrage		
Zone de données						

# Datagramme IP: entête

## Entête IP (20 octets)

- Version (sur 4 bits) 4 (IPv4)
- Longueur entête (sur 4 bits) 5 (soit  $5 \times 4 = 20$  octets)
- Type de service 00
- Longueur totale du datagramme 00 3c (60 octets = 20 + 8 + 32)
- Identifiant 05 b9
- Drapeau 00 (not set)
- Fragment 00
- Time to live 80 ( $128_{10} = \text{nb de nœuds traversés}$ )
- Protocole du niveau supérieur 01 (ICMP)
- Header checksum 01 84
- IP source c0 a8 01 19 (192.168.2.25)
- IP destination c0 a8 01 1a (192.168.2.26)
- Pas d'options

# Paquet ICMP: entête

## Entête ICMP (8 octets)

- Type 08 (echo request)
- Code 00
- Checksum 4d 2a
- Identifiant 00 01 (permet d'associer la réponse a la demande)
- N° de séquence 00 31 (incrémenté a chaque demande)

# Paquet ICMP : données

Données (32 octets) séquence en boucle de minuscules de a à w

```
61 62 63 64 65 66 67 68 69 6a 6b 6c 6d 6e 6f 70
 a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69
 q  r  s  t  u  v  w  a  b  c  d  e  f  g  h  i
```

## Remarque

32 octets de données est la valeur par défaut de Windows. Elle peut être changée avec le commutateur `-l`.

Exemple : `ping -n 2 -l 16 192.168.1.26` envoie deux (-n 2) requêtes d'écho de longueur 16 octets (-l 16) à l'hôte 192.168.1.26

Sous Linux on aurait : `ping -c 2 -s 8 192.168.1.26`

La taille du paquet envoyé sera bien de 16 octets car on ajoute 8 octets d'entête ICMP à la valeur du paramètre `-s`. Linux sera toujours plus rigoureux ... Il affiche 16 octets comme Windows alors que ce dernier en envoie 24 !

# Composition des données sur le lien

L'assemblage des trois couches va constituer les données envoyées sur le lien

Paquet ICMP

ICMP

Datagramme IP

IP dst 192.168.2.26

IP src 192.168.2.25

Trame Ethernet

Mac dst 08:00:27:ae:38:3e

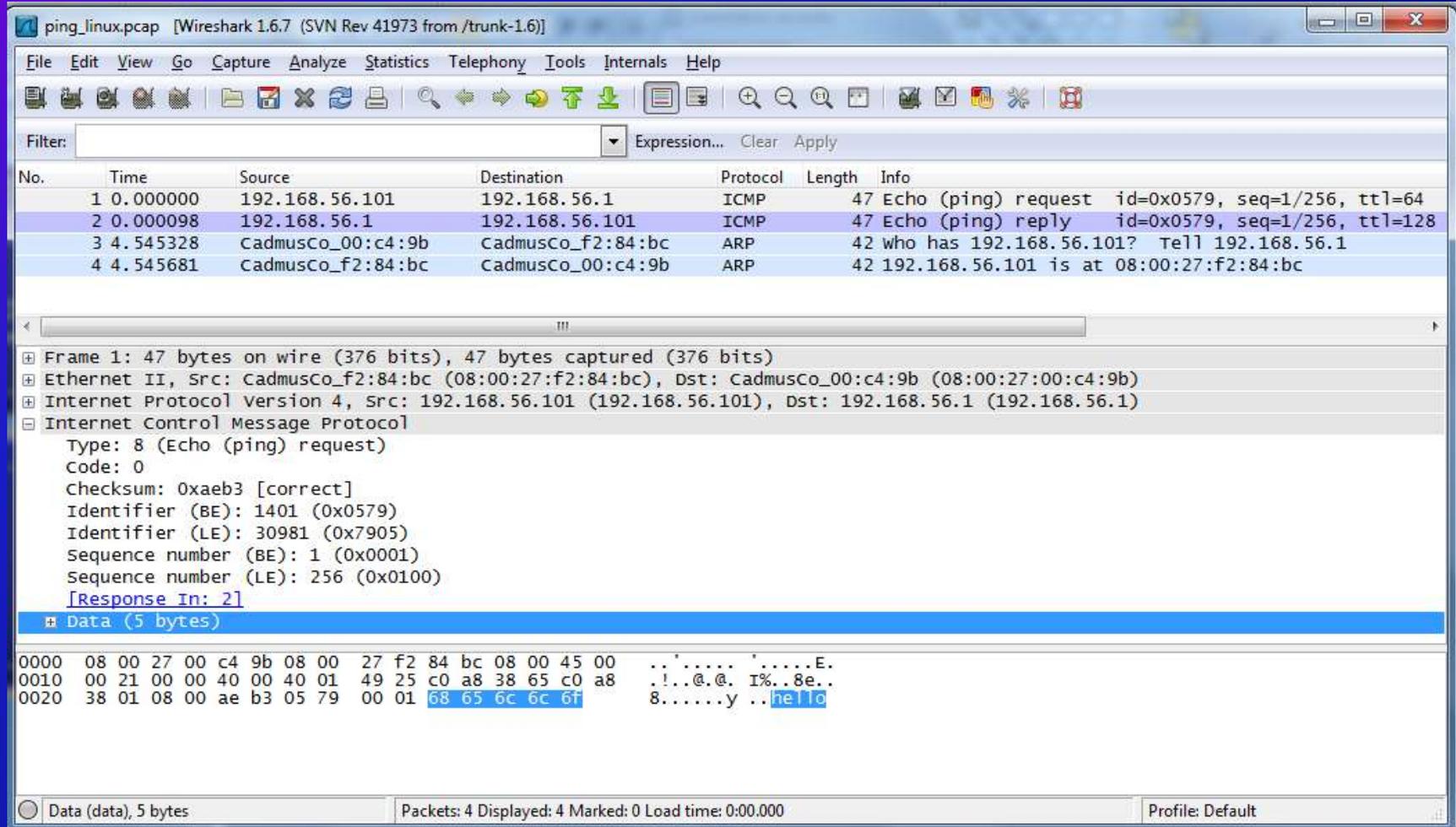
Mac src e0:b9:a5:5<sup>e</sup>:df:be

# Un « vrai » ping

- Depuis une machine Linux faire : `ping -c 1 -s 5 -p 68656c6c6f 192.168.56.1`
  - -c 1 envoie un seul paquet
  - -s 5 envoie 5 octets. La longueur de la trame sera de 47 octets hors préambule, SOF et FCS.
    - Entête Ethernet 14 Octets
    - Entête IP 20 octets
    - Entête ICMP 8 octets
    - Données 5 octets
  - -p 68656c6c6f est un motif de 5 octets à envoyer. Dans ce cas c'est « hello »
  - 192.168.56.1 est l'interface virtuelle fournie par Virtualbox à la machine Windows. Le client linux à l'adresse IP 192.168.56.101
- Remarque : bien que la trame fasse 51 octets (47 + 4 pour FCS) < 64, les octets de padding n'apparaissent pas.  
La capture est dans le fichier ping\_linux.cap

# Un « vrai » ping

## Fichier ping\_linux.pcap



ping\_linux.pcap [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.56.101	192.168.56.1	ICMP	47	Echo (ping) request id=0x0579, seq=1/256, ttl=64
2	0.000098	192.168.56.1	192.168.56.101	ICMP	47	Echo (ping) reply id=0x0579, seq=1/256, ttl=128
3	4.545328	CadmusCo_00:c4:9b	CadmusCo_f2:84:bc	ARP	42	who has 192.168.56.101? Tell 192.168.56.1
4	4.545681	CadmusCo_f2:84:bc	CadmusCo_00:c4:9b	ARP	42	192.168.56.101 is at 08:00:27:f2:84:bc

Frame 1: 47 bytes on wire (376 bits), 47 bytes captured (376 bits)

- Ethernet II, Src: CadmusCo\_f2:84:bc (08:00:27:f2:84:bc), Dst: CadmusCo\_00:c4:9b (08:00:27:00:c4:9b)
- Internet Protocol Version 4, Src: 192.168.56.101 (192.168.56.101), Dst: 192.168.56.1 (192.168.56.1)
- Internet Control Message Protocol
  - Type: 8 (Echo (ping) request)
  - Code: 0
  - Checksum: 0xaeb3 [correct]
  - Identifier (BE): 1401 (0x0579)
  - Identifier (LE): 30981 (0x7905)
  - Sequence number (BE): 1 (0x0001)
  - Sequence number (LE): 256 (0x0100)
  - [\[Response In: 2\]](#)
- Data (5 bytes)

```
0000  08 00 27 00 c4 9b 08 00 27 f2 84 bc 08 00 45 00  ..'.....'.....E.
0010  00 21 00 00 40 00 40 01 49 25 c0 a8 38 65 c0 a8  .!..@.@. I%..8e..
0020  38 01 08 00 ae b3 05 79 00 01 68 65 6c 6c 6f    8.....y ..hello
```

Data (data), 5 bytes | Packets: 4 Displayed: 4 Marked: 0 Load time: 0:00.000 | Profile: Default

# TP « ping »

---

- Vider la table arp, vérifier.
- Capturer le trafic sur l'interface réseau en limitant ce qui est capturé au minimum
- Faire un ping de deux paquets vers la passerelle puis arrêter la capture
  - Quelle est l'adresse MAC de la passerelle ?
  - Quelle est la taille des données de ping ?
  - Quelles sont ces données ?
  - Quel est l'identifiant de la seconde séquence ?

# Erreurs ICMP

- Envoyer un ping vers une adresse non attribuée sur votre réseau. Ex: 192.168.1.199
  - Réponse de ping
  - Indicateur dans le datagramme IP
- Envoyer un ping vers une adresse d'un autre réseau (qu'il y ait une station ou pas au bout n'a pas d'importance). Ex : 192.168.2.2
  - Réponse de ping
  - Indicateur dans le datagramme IP
- Comparaison des deux séquences

# Bilan de IP : Avantages / inconvénients

- **Avantages**
  - Remise de datagrammes entre hôtes identifiés par adresse IP
  - Détection des erreurs au niveau trame par comparaison du FCS reçu avec celui calculé
  - Détections des erreurs au niveau datagramme par ICMP
- **Inconvénients**
  - Livraison des datagrammes non garantie
  - En cas d'erreur le datagramme est perdu
  - Si les données sont véhiculées par plusieurs datagrammes, ces derniers peuvent ne pas être en séquence.
  - Erreurs non ICMP pas signalées
  - Pas de contrôle de flux

Les remèdes à la faiblesse de IP

# Remèdes à la faiblesse de IP

- Assurer la correction des erreurs et demander la réémission des trames erronées
  - Signalées par ICMP
  - Non signalées (erreurs de CRC)
- Assurer le séquençement des paquets contenant les données issues d'un même message
- Protocoles UDP (**U**ser **D**atagram **P**rotocol) et TCP (**T**ransport **C**ontrol **P**rotocol)
  - UDP : transport rapide, non connecte, permettant la multidiffusion
  - TCP : transport fiable en mode connecte point-à-point

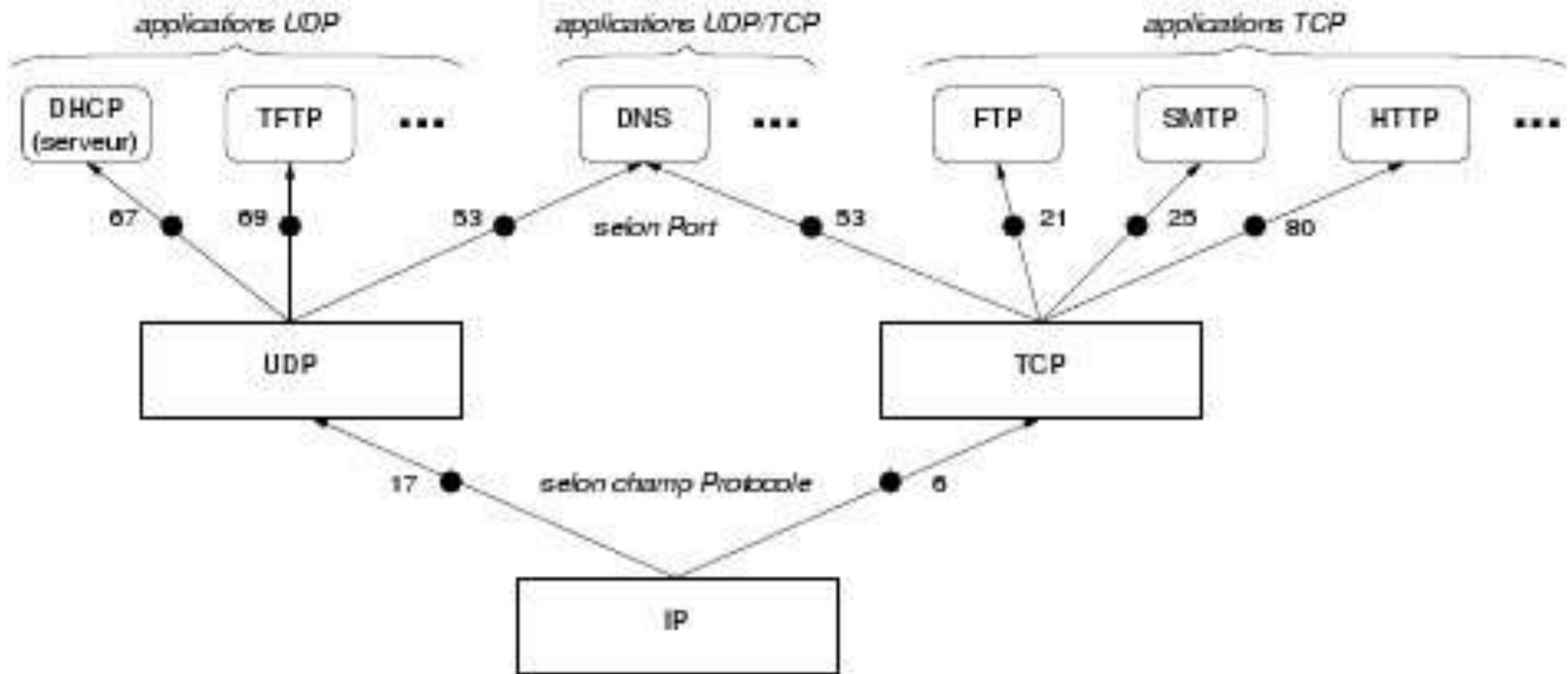
Ces deux protocoles distinguent les applications au sein d'un même hôte (par exemple DHCP, DNS, FTP, HTTP, SSH, etc. et garantissent l'indépendance des communications => Plusieurs applications réseau peuvent s'exécuter sur le même hôte.

- La solution retenue sur Internet est l'utilisation de destinations abstraites : les ports (ne pas confondre avec les ports physiques des hubs/switches)

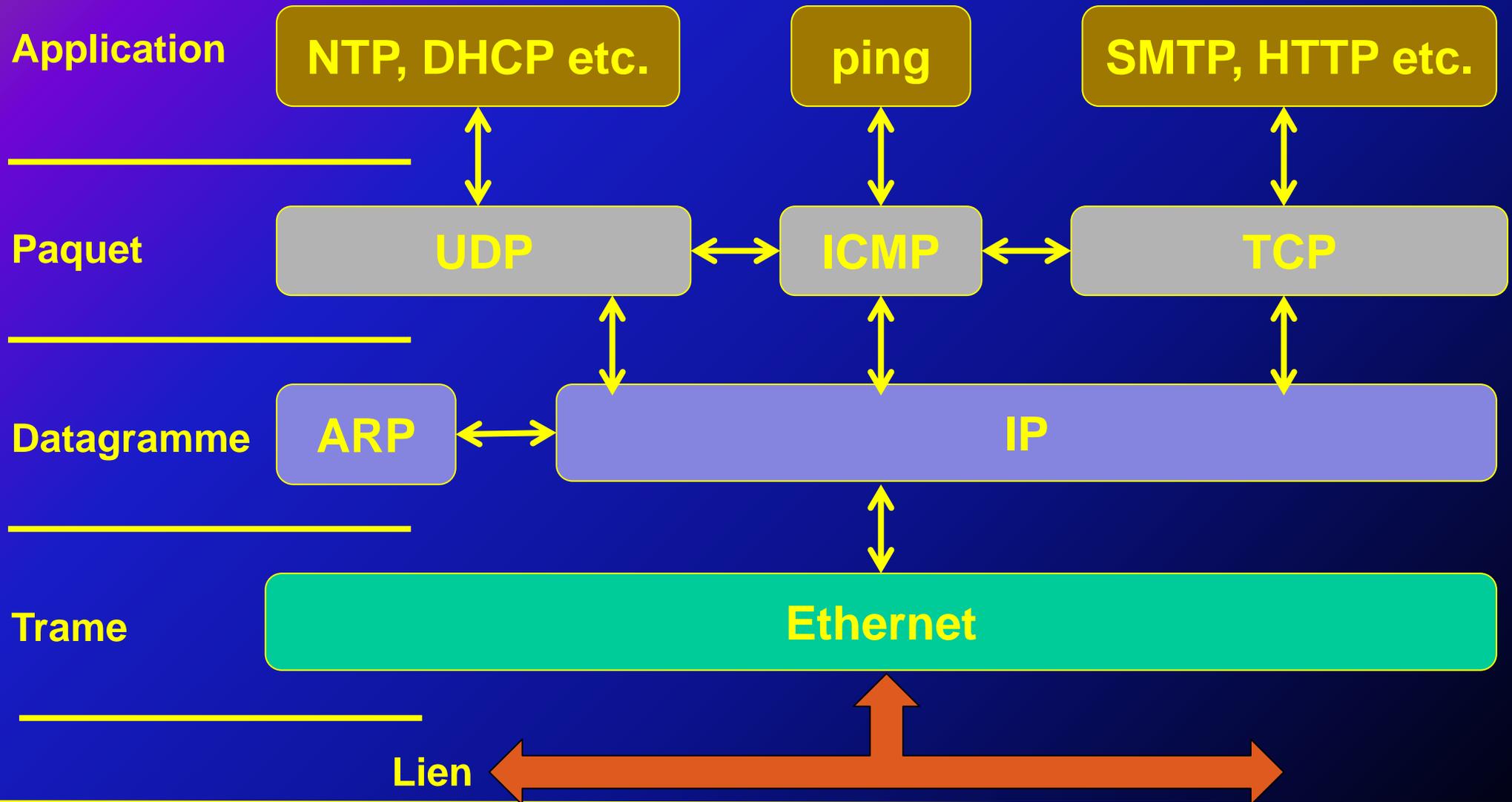
# Les ports UDP / TCP

- le système permet aux applications de se voir affecter un port UDP et/ou TCP (choisi ou de manière arbitraire). Exemple de ports normalisés :
  - smtp 25 (Single Mail Transfert Protocol TCP)
  - dns 53 (Domain Name Service UDP et TCP)
  - http 80 (Hyper Text Transfert Protocol TCP)
  - ntp 123 (Network Time Protocol UDP)
- UDP et TCP fournissent chacun un ensemble de ports indépendants : le port n de UDP est indépendant du même port n de TCP
- L'adresse d'une application Internet est le triplet : (adresse IP, protocole de transport, numéro de port). Par exemple <https://192.168.1.26:7071> précise que l'on va contacter la machine 192.168.1.26 sur le port 7071 en utilisant le protocole https.
- Les port de 1 à 1023 sont réservés et correspondent a des services particuliers (voir ci dessus). Les ports de 1024 à 49151 sont enregistrés mais peuvent être utilisés. Les ports de 49152 à 65535 sont dynamiques.
- Le n° de port est défini par un entier codé sur 16 bits

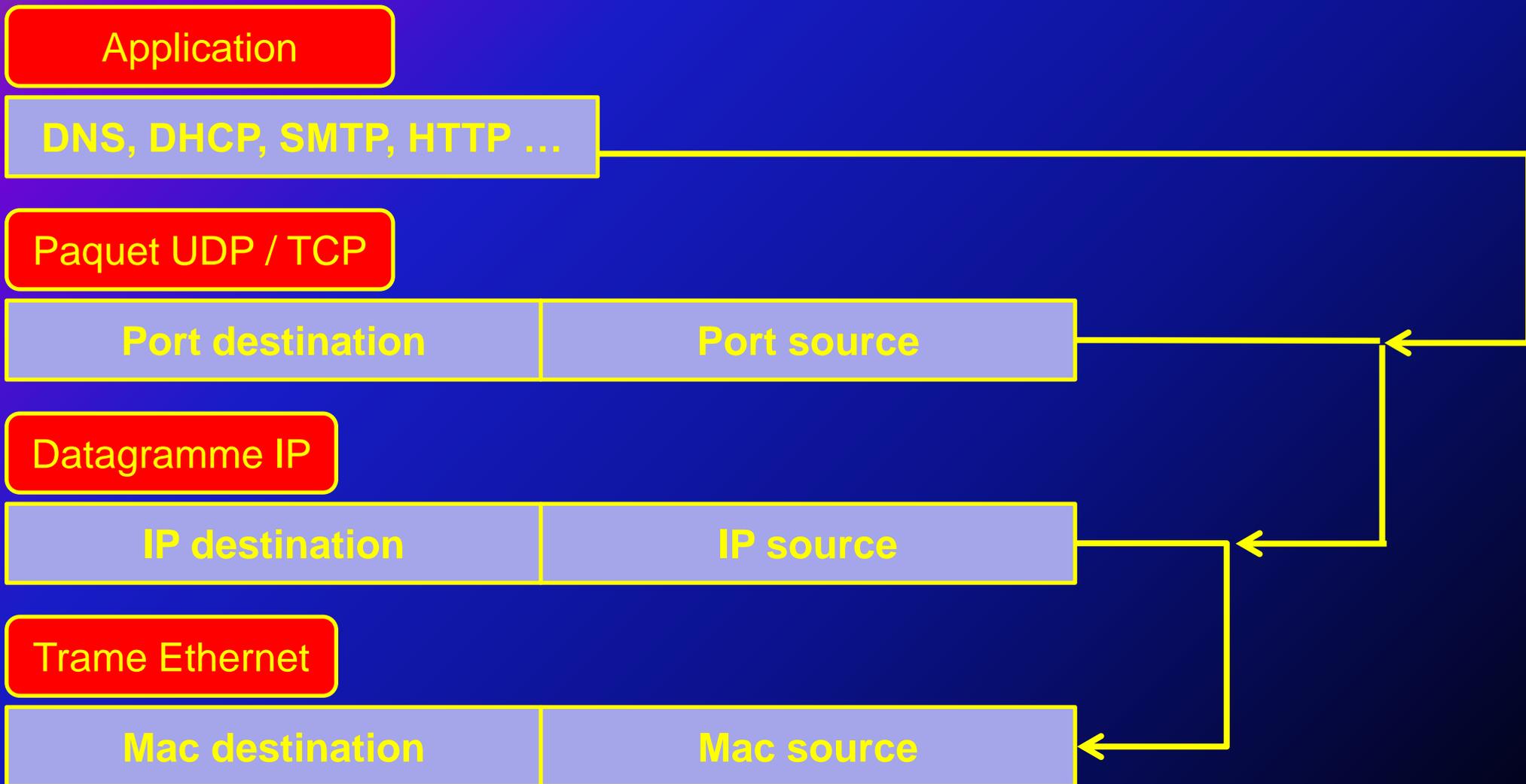
# Les ports UDP / TCP



# Modèle en couches IP



# Modèle en couches IP



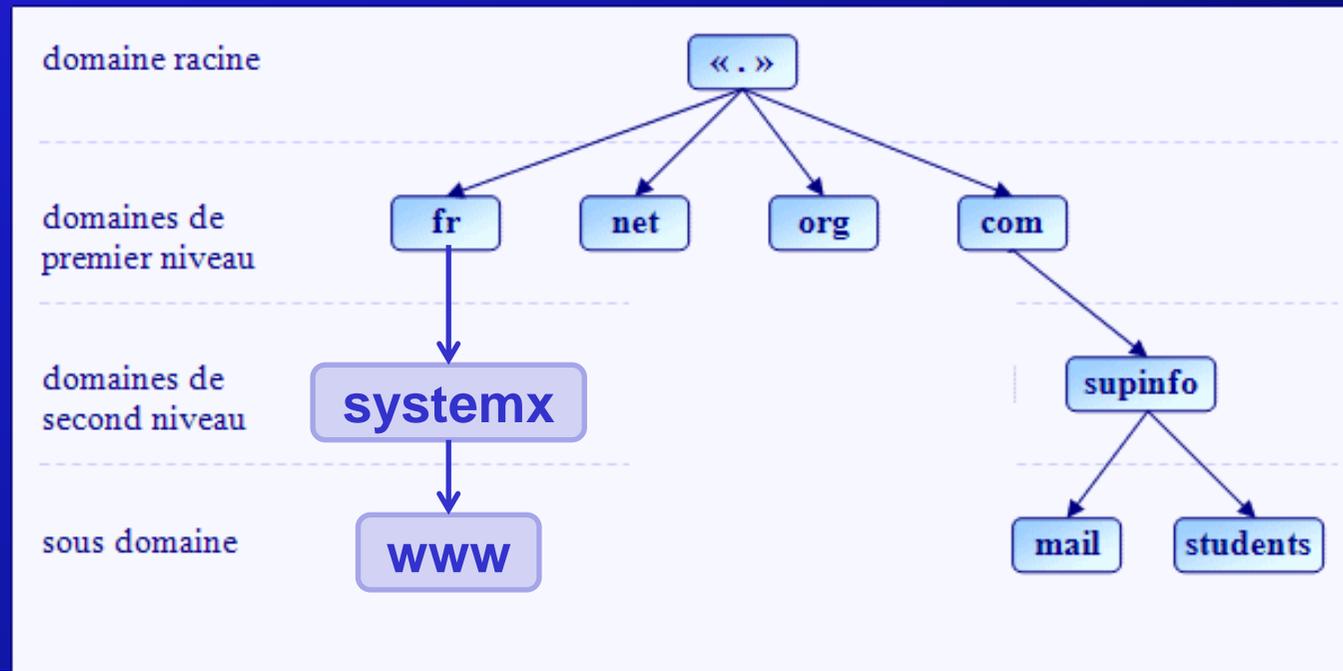
# UDP

---

DNS, NTP, TFTP, etc.

# Le DNS

- DNS = Domain Name Service
- Association d'un nom à une adresse IP ou résolution de nom
  - Ex : www.systemx.fr = 82.234.79.107
- Résolution inverse : quel est le nom de cette adresse IP
  - Ex : 82.234.79.107 = www.systemx.fr
- L'espace des noms est hiérarchisé



# Le DNS

- Avec le DNS, l'espace de noms est organisé en une hiérarchie au sommet de laquelle figure la racine (".") et immédiatement en dessous les TLD (Top-Level Domain) ou domaines de niveau supérieur
- L'ICANN (Internet Corporation for Assigned Names and Numbers) a en charge la création des TLD et a créé notamment les TLD suivants :
  - com : entreprises commerciales
  - edu : établissements d'enseignement
  - org : organisations diverses
  - mil : comme son nom l'indique ...
- Puis un TLD par pays / région / activité sur 2 à 4 lettres (norme ISO 3166) :
  - fr : France
  - uk : Royaume-Uni
  - tv : ile Tuvalu (qui en profite bien. . . )
  - cat : Catalogne (région espagnole)
  - aero : comme son nom l'indique
  - xxx : ce n'est ni un pays ni une région ...

# Le DNS

- L'ICANN a ensuite délégué la gestion des sous-domaines des TLD a des entreprises ou organisations gouvernementales :
  - com et net a la société VeriSign (société privée)
  - edu, org et autres a l'INTERNIC (association américaine)
  - fr et re (île de la Réunion) a l'AFNIC (Association Française pour le Nommage Internet en Coopération)
- Le propriétaire d'un sous-domaine peut ensuite le décliner a sa guise afin de nommer des ordinateurs et/ou de créer des sous-sous-domaines :
  - www.free.fr (ordinateur)
  - hd.free.fr (sous-sous-domaine)
  - releves.hd.free.fr (ordinateur)
- Ces informations sont publiques et consultables par tout le monde :
  - sous Linux, en utilisant la commande whois, sur le Web, grâce a des serveurs WHOIS.
  - Ex : <http://www.whois.net>  
<http://www.afnic.fr/fr/produits-et-services/services/whois>  
<http://www.gandi.net> (excellent prestataire français)

# Le DNS

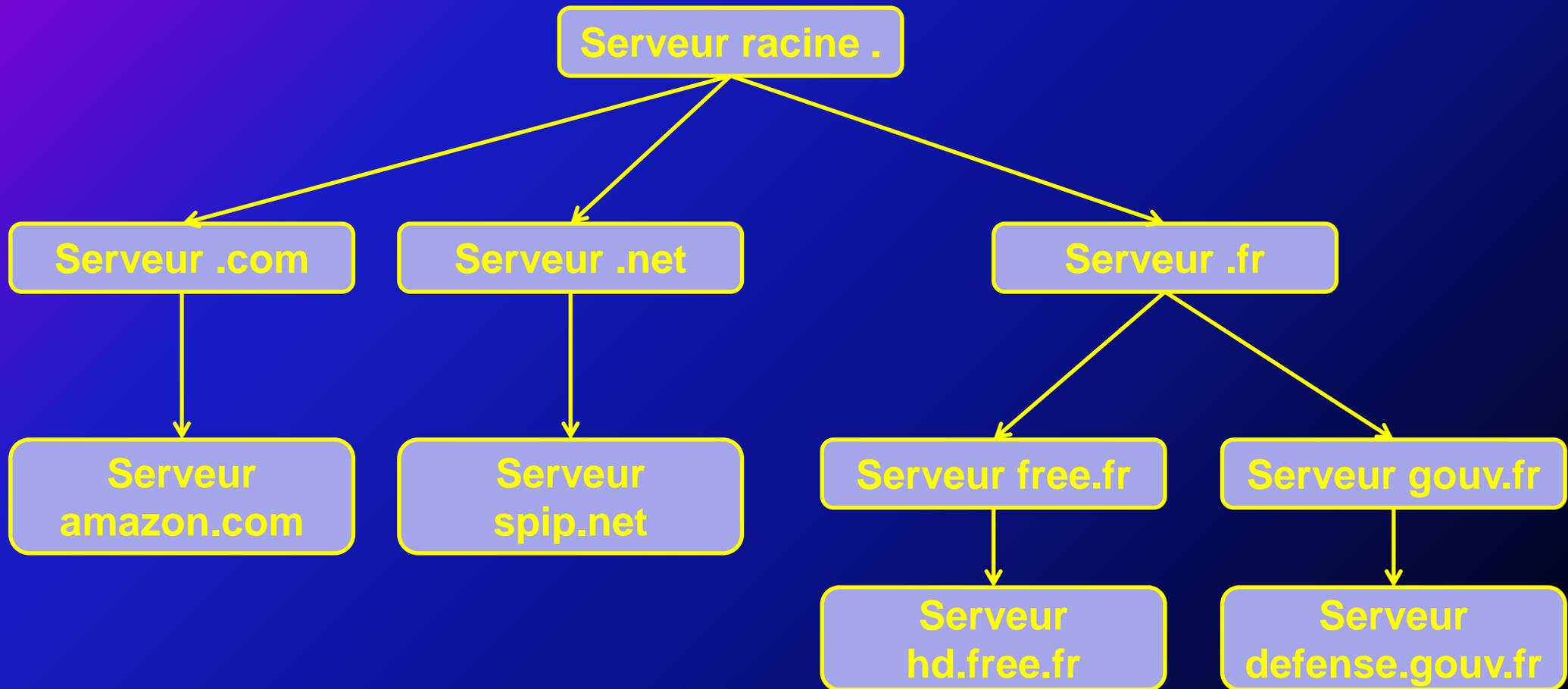
- Le terme domaine désigne à la fois un domaine, un sous-domaine.
  - Ex : fr, gouv.fr et education.gouv.fr sont des domaines  
fr est un sous domaine de la racine « . », gouv est un sous domaine de fr et education est un sous domaine de gouv.fr
- On ne peut pas distinguer un domaine d'un nom d'ordinateur
  - Ex : hd.free.fr est un domaine et [www.free.fr](http://www.free.fr) est un ordinateur
- Un même nom peut avoir plusieurs adresses IP
  - Ex: www.google.com a actuellement 4 adresse IP : 74.125.132.103 à 106
- Une même IP peut avoir plusieurs noms
  - Ex : hetb-concept.com et hit-nxdomain.opendns.com ont la même adresse IP : 67.215.65.132

# Le DNS

- Le nom d'un domaine ne doit pas dépasser 63 caractères ordinaires (sans accents, cédille ...) et doit commencer par une lettre ou un chiffre. Le caractère « - » (moins ou tiret) est autorisé à l'exclusion de tout autre
- Le DNS est insensible à la casse
  - Ex : education.gouv.fr = Education.Gouv.fr
- Les points sont les séparateurs des labels
  - Ex : iut.univ-aix.fr est composé des 3 labels iut, univ-aix et fr
- Un nom complètement qualifié ou FQDN (Fully Qualified Domain Name) est un domaine contenant sa position dans la hiérarchie et « devrait » se terminer par un point
  - Ex : univ-aix.fr (référence absolue). Le point est omis.

# Le DNS

- Les serveurs de noms d'un domaine doivent connaître les serveurs de noms racines, du domaine parent et des domaines fils (délégation de zone)



# Le DNS

---

- La racine a 13 serveurs de a à m.gtld-servers.net
- La zone fr a 4 serveurs de d à g.ext.nic.fr
- La zone free.fr a 2 serveurs freens1 et 2-g20.free.fr
- La zone hd.free.fr a 2 serveurs ns2 et 3-rev.proxad.net

# Le DNS

- Un dépositaire de nom de domaine doit mettre a disposition au moins 2 serveurs de noms (sur des réseaux en principe différents) chargés de répondre aux requêtes concernant les noms locaux.
- Toute modification sur un serveur parent est répliqué sur les serveurs enfant par transfert de la zone modifiée
- Tout hôte (PC) devrait connaitre au moins un serveur de noms (en principe de son domaine) mais ce n'est pas une obligation
- DNS est un protocole de la couche application. Il utilise
  - UDP pour les requêtes
  - TCP pour les transferts de zone
- Que ce soit en UDP ou en TCP le port normalisé est **53**

# Le DNS

- Sur un hôte, le client DNS effectuant la résolution de noms est appelé solveur de noms
- Pour résoudre un nom, le solveur s'adresse à son serveur de noms. Celui-ci a deux manières de réagir :
  - résolution **récurive** : s'il ne connaît pas la réponse, le serveur est chargé de la trouver. Il contactera un autre serveur, etc., et la réponse reviendra apparaissant au solveur comme venant de son serveur.
  - résolution **itérative** : s'il ne connaît pas la réponse, le serveur peut indiquer quel serveur est susceptible de la connaître. Le solveur doit ensuite effectuer la recherche seul en contactant ce serveur, etc., jusqu'à contacter un serveur en mesure de lui répondre
- Dans le cas normal, les solveurs demandent toujours une résolution récursive. Pour des raisons de sécurité, elle n'est autorisée qu'aux serveurs qui deviennent hôtes du serveur parent et elle est interdite aux hôtes qui ne sont pas eux même serveur DNS.

# Exemple de résolution

On souhaite résoudre `releves.hd.free.fr`

- Interrogation de la racine pour savoir qui est le serveur pour la zone fr
  - À la racine `a.gtld-servers.net` répond que la zone fr est gérée par `d.ext.nic.fr`
- Interrogation de la zone fr pour savoir qui est le serveur pour le (sous) domaine free
  - `d.ext.nic.fr` répond que la zone free est gérée par `freens1-gv.free.fr`
- Interrogation de la zone free pour savoir qui est le serveur pour le (sous) domaine hd
  - `freens1-gv.free.fr` répond que la zone hd est gérée par `ns2-rev.proxad.net`
- Interrogation de la zone hd pour connaître l'adresse de releves
  - `ns2-rev.proxad.net` répond que releves a l'adresse `82.66.178.10`

## Remarques

- La première résolution faisant appel au serveur racine doit résoudre un .net qui est sous la racine. Le serpent se mord la queue mais le protocole DNS a prévu le cas
- On suppose que l'on sait déjà que releves est un host

# Protocole DNS / UDP

- Nous allons utiliser une application qui va mettre en œuvre le protocole DNS s'appuyant sur UDP pour associer un nom de host à une adresse IP.
- Sous Windows en mode commande (les paramètres de nom et d'adresse peuvent varier)
  - `nslookup`
  - `server 192.168.2.45` (indique quel serveur DNS sera utilisé)
  - `esx` (demande quelle est l'adresse IP de la machine nommée esx)
  - `Ctrl+C` pour quitter une fois la réponse obtenue
- Sous Linux
  - `host esx 192.168.2.45`

La réponse est 192.168.2.199

# Protocole DNS / UDP

- Sur une machine Windows vous devriez avoir ceci :



```
Administrateur : C:\Windows\system32\cmd.exe - nslookup
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\Bernard>nslookup
Serveur par défaut :   esx.systemx.local
Address: 192.168.2.199

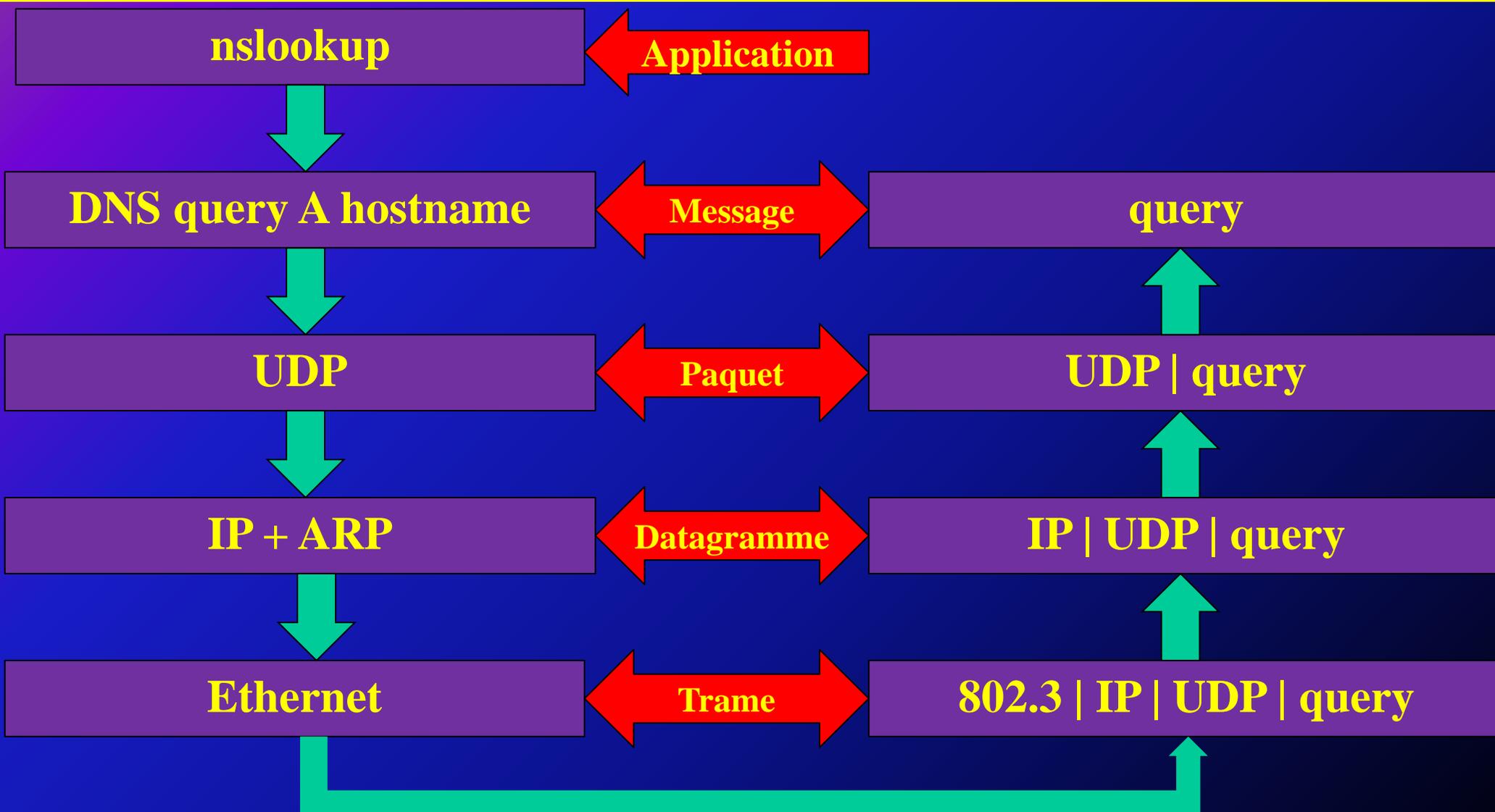
> server 192.168.2.45
Serveur par défaut :   [192.168.2.45]
Address: 192.168.2.45

> esx
Serveur :   [192.168.2.45]
Address: 192.168.2.45

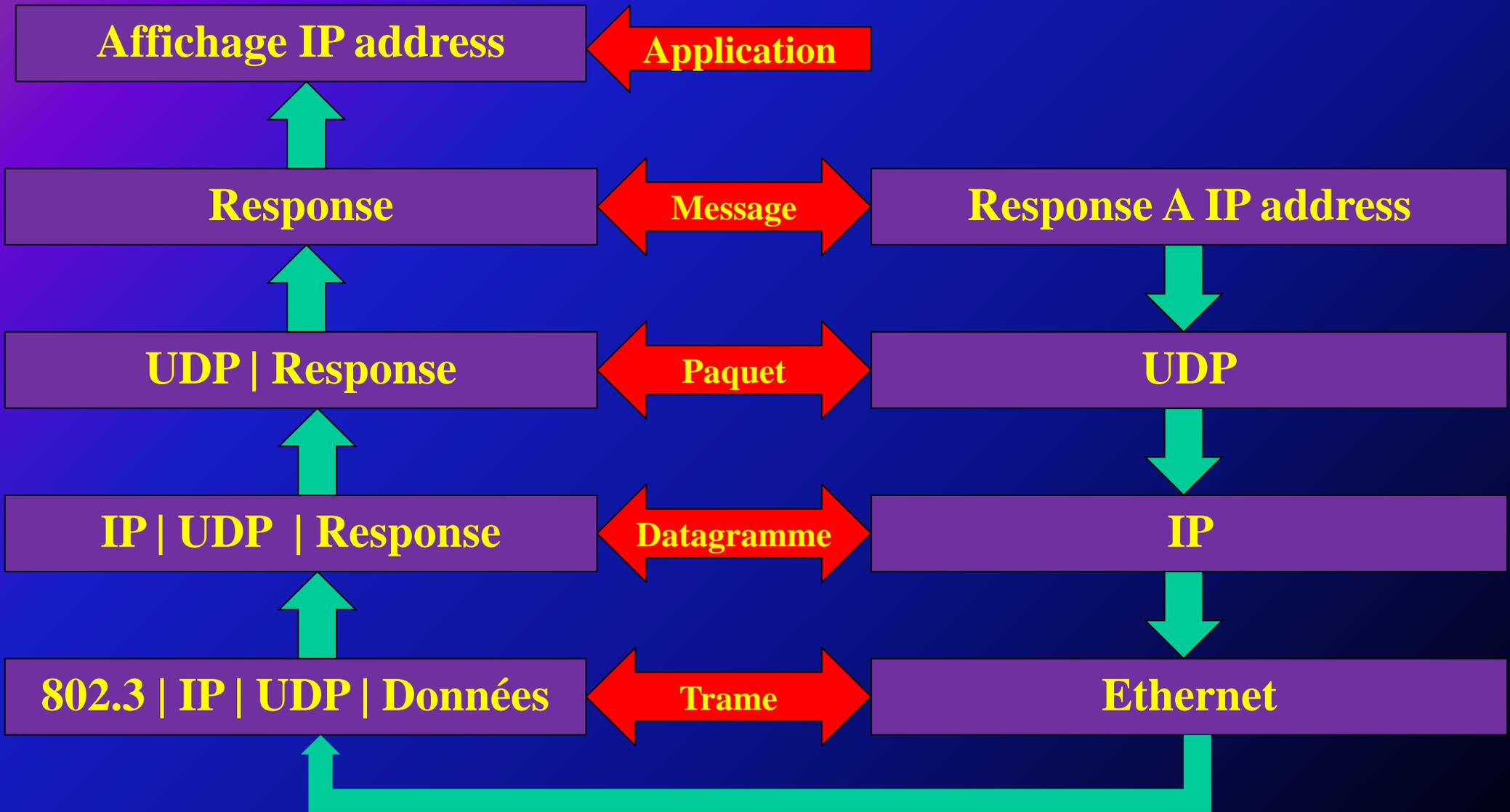
Nom :      esx
Address: 192.168.2.199

> -
```

# Demande de résolution DNS



# Réponse à une demande de résolution



# TP NSLOOKUP

- Au prompt de nslookup :
  - server IP : utilise le serveur IP pour répondre aux requêtes
  - set q= : spécifie quelle question va être posée
    - ns : serveur de nom. Taper un nom domaine pour obtenir l'adresse de ses serveurs de noms
    - mx : serveur de mail . Taper un nom domaine pour obtenir l'adresse de ses serveurs de mail
    - a : adresse. Taper un nom d'hôte FQDN pour obtenir son adresse IP ou son adresse IP pour obtenir son nom.
    - ptr : pointeur. Taper une adresse IP pour obtenir son nom d'hôte FQDN

# TP NSLOOKUP

- Quels sont les serveurs de nom de domaine de google.com ?
- Quels sont les serveurs de nom de domaine des zones
  - fr
  - free.fr
  - hd.free.fr
- Quelle est l'adresse IP de releves.hd.free.fr ?
- Faire la résolution inverse de 194.2.0.20 ?
- Passer le type de requête à PTR et faire la même résolution inverse. Qu'est-ce qui différencie les deux réponses ?
- Quel est le nom du serveur DNS attaché à votre connexion réseau (ipconfig /all)
- Quel est le nom de votre domaine ?
- Est-ce que ce serveur fait autorité pour cette zone ?
- Quel est le mail exchanger du domaine systemx.fr ? Qu'observer-vous ?
- Changer de serveur DNS pour le premier de hd.free.fr
- Résoudre releves.hd.free.fr. Qu'observez-vous ?

# Résolution DNS

On visualise le processus (dns\_vm.pcap)

dns\_vm.pcap [Wireshark 1.6.7 (SVN Rev 41973 from /trunk-1.6)]

Filter: dns

No.	Time	Source	Destination	Protocol	Length	Info
8	1.857021	192.168.2.48	192.168.2.45	DNS	63	Standard query A esx
9	1.857534	192.168.2.45	192.168.2.48	DNS	79	Standard query response A 192.168.2.199
10	1.857744	192.168.2.48	192.168.2.45	DNS	63	Standard query AAAA esx
11	1.858003	192.168.2.45	192.168.2.48	DNS	63	Standard query response

Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))  
Total Length: 49  
Identification: 0x6dcb (28107)  
Flags: 0x00  
Fragment offset: 0  
Time to live: 128  
Protocol: UDP (17)  
Header checksum: 0x4743 [correct]  
Source: 192.168.2.48 (192.168.2.48)

```
0000 08 00 27 ae 38 3e e0 b9 a5 5e df be 08 00 45 00  ..'.8>.. .^....E.
0010 00 31 6d cb 00 00 80 11 47 43 c0 a8 02 30 c0 a8  .1m.... GC...0..
0020 02 2d e5 0a 00 35 00 1d 1a df 00 09 01 00 00 01  .-...5.. ....
0030 00 00 00 00 00 00 03 65 73 78 00 00 01 00 01  .....e sx.....
```

Protocol (ip.proto), 1 byte      Packets: 101 Displayed: 4 Marked: 0 Load time: 0:00.006      Profile: Default

# Trame question DNS / UDP

Application

DNS query A hostname

Paquet UDP

Port destination 53

Port source 58634

Datagramme IP

IP dst 192.168.2.45

IP src 192.168.2.48

Trame Ethernet

Mac dst 08:00:27:ae:38<sup>e</sup>3e

Mac src e0:b9:a5:5<sup>e</sup>:df:be

# Résolution DNS

On observe l'enchaînement de tâches suivantes :

1. De 192.168.2.48 à 192.168.2.45 question : A esx (quelle est l'adresse IP de esx)
2. De 192.168.2.45 à 192.168.2.48 réponse : 192.168.2.199
3. Les deux requêtes suivantes sont faites pour IPv6 (pas étudié pour le moment)

On constate :

- Le champ « protocol » du datagramme IP indiquant que le protocole de niveau supérieur utilisé est  $17_{10}=11_{16}$  soit UDP
- La longueur de l'entête UDP est fixe, de 8 octets
  - Source port e5 0a (58634 fixé aléatoirement par la source)
  - Destination port 00 35 (53 normalisé par IANA)
  - Longueur du paquet 00 1d (29 octets = 8 entête UDP + 21 message DNS)
  - Checksum 1a df

Total de l'entête UDP : 8 octets

- Données DNS sur 21 octets

Total du paquet UDP : 29 octets variable car les données ne sont pas de longueur fixe.

# Résolution DNS question

- N° de transaction      00 09 (n° unique tout au long de la transaction)
- Drapeaux                01 00 (la question est récursive)
- Question                00 01 (c'est une question)
- Réponse                00 00 (pas de réponse)
- Authority RRs        00 00 (il n'y a pas de DNS plus « proche » que celui interrogé, utile en réponse seulement)
- Additional RRs       00 00 (il n'y a pas d'informations complémentaires données par le serveur DNS, utile en réponse seulement)
- Nom demandé         03 65 73 78 00 (end\_of\_text **esx**null)
- Type                    00 01 (adresse)
- Classe                  00 01 (internet)

Total 21 octets

# Trame réponse DNS / UDP

Application

DNS response A IP address

Paquet UDP

Port destination 58634

Port source 53

Datagramme IP

IP dst 192.168.2.48

IP src 192.168.2.45

Trame Ethernet

Mac dst e0:b9:a5:55:df:be

Mac src 08:00:27:ae:38:3e

# Résolution DNS retour UDP

- Pour le retour il suffit d'inverser le port source et le port destination. L'entête du paquet UDP devient :
  - Source port 00 35 (53)
  - Destination port e5 0a (58634)
  - Longueur du paquet 00 2d (45 octets = 8 entête UDP + 37 message DNS)
  - Checksum 13 ba
  - Données DNS sur 37 octets
- On constate que la réponse (37 octets) n'a pas la même longueur que la question (21 octets)

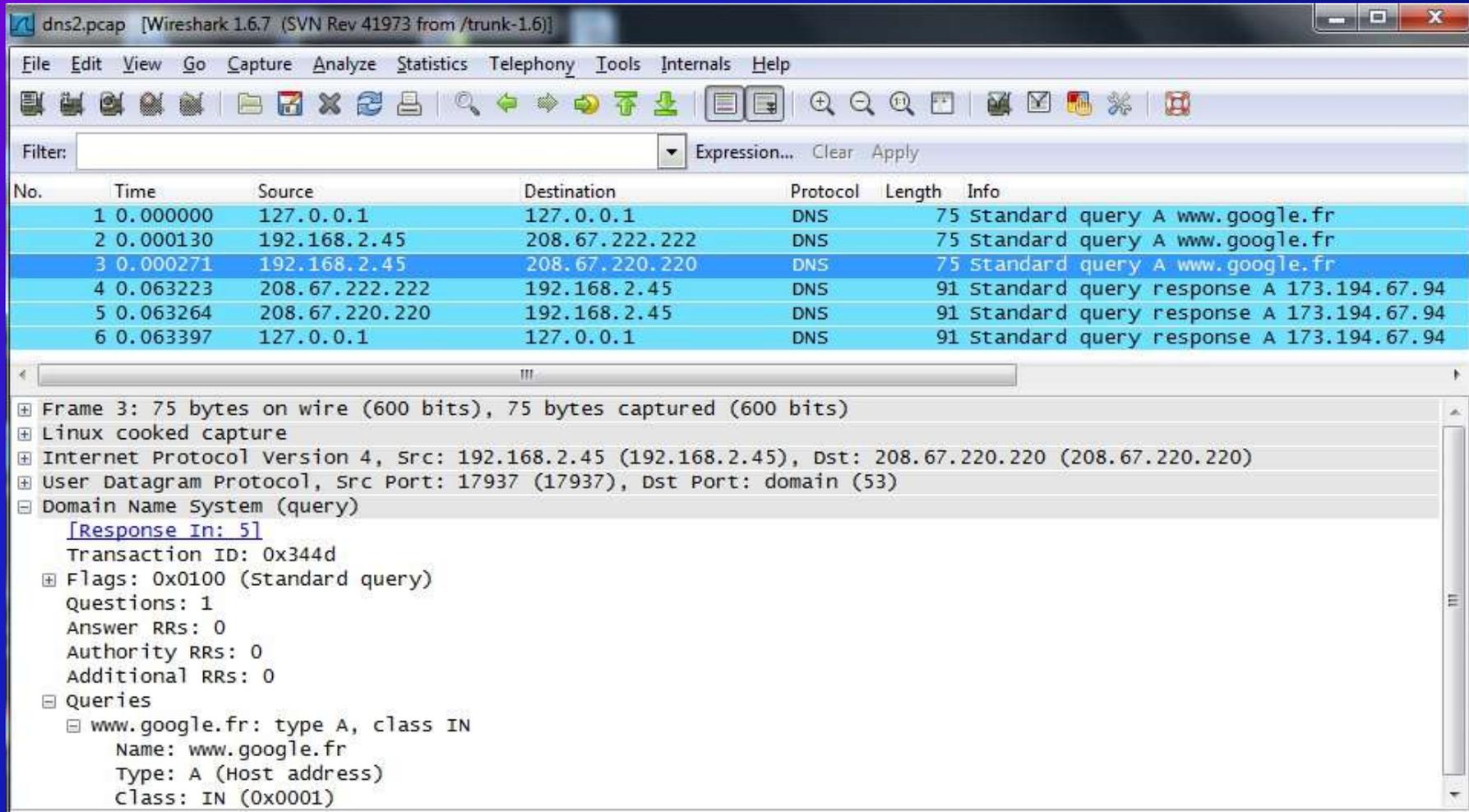
# Résolution DNS réponse

- N° de transaction 00 09 (c'est bien le même que pour la question)
- Drapeaux 85 80 c'est une réponse
  - serveur faisant autorité pour ce domaine
  - réursion demandée, réursion disponible
- Question 00 01 (il y a eu une question)
- Réponse 00 01 (il y a une réponse)
- Authority RRs 00 00 (pas de serveur DNS plus « proche »)
- Additional RRs 00 00 (pas d'informations complémentaires)
- Question (identique à la demande, 9 octets)
- Réponse (les trois premiers paramètres identiques à la question, 16 octets)
  - Durée de vie 00 00 (la réponse n'est pas en cache)
  - Longueur des données 00 04 (4 octets)
  - Adresse c0 a8 02 c7 (IP 192.168.2.199)

Total 37 octets

# Résolution « en vrai »

Analyser la résolution suivante (fichier dns2.pcap)



The screenshot shows the Wireshark interface with a packet capture of a DNS query and response. The packet list pane shows six packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	DNS	75	Standard query A www.google.fr
2	0.000130	192.168.2.45	208.67.222.222	DNS	75	Standard query A www.google.fr
3	0.000271	192.168.2.45	208.67.220.220	DNS	75	Standard query A www.google.fr
4	0.063223	208.67.222.222	192.168.2.45	DNS	91	Standard query response A 173.194.67.94
5	0.063264	208.67.220.220	192.168.2.45	DNS	91	Standard query response A 173.194.67.94
6	0.063397	127.0.0.1	127.0.0.1	DNS	91	Standard query response A 173.194.67.94

The packet details pane for packet 3 shows the following structure:

- Frame 3: 75 bytes on wire (600 bits), 75 bytes captured (600 bits)
- Linux cooked capture
- Internet Protocol Version 4, Src: 192.168.2.45 (192.168.2.45), Dst: 208.67.220.220 (208.67.220.220)
- User Datagram Protocol, Src Port: 17937 (17937), Dst Port: domain (53)
- Domain Name System (query)
  - [Response In: 5]
  - Transaction ID: 0x344d
  - Flags: 0x0100 (Standard query)
  - Questions: 1
  - Answer RRs: 0
  - Authority RRs: 0
  - Additional RRs: 0
  - Queries
    - www.google.fr: type A, class IN
      - Name: www.google.fr
      - Type: A (Host address)
      - Class: IN (0x0001)

# TP NTP

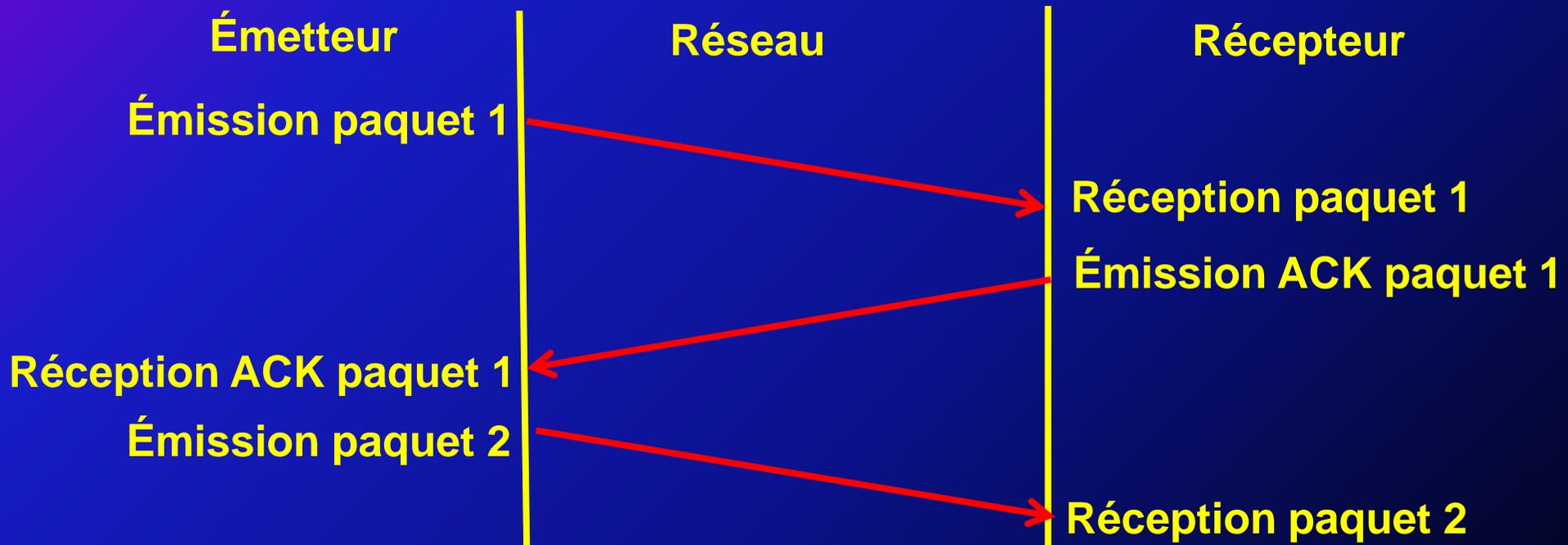
- Le but de ce TP est de découvrir le protocole NTP et d'analyser son fonctionnement.
- Installer le client NTP que vous trouverez ici :  
<http://www.timesynctool.com>  
Lire attentivement la documentation puis installer l'application comme un service Windows
- Répondre de manière détaillée et justifiée aux questions suivantes
  - Que fait le service que vous venez d'installer.? Tester en décalant l'heure de votre PC (en retard de préférence) ?
  - Quelle est la « strate » de 0.nettime.pool.ntp.org
  - Comment est organisé, du point de vue serveur, NTP ?
  - Quels sont en France les serveurs de strate 1 et 2 ?
  - En analysant ce qui se passe sur le réseau vous montrerez les différentes étapes de la synchronisation de votre PC. A chaque trame question / réponse vous montrerez les différentes couches de protocoles avec les entêtes liées à chaque niveau. Vous pouvez procéder par essais successifs puis une dernière fois en vidant la table ARP et le cache DNS. C'est à partir de ce dernier essai que vous rédigerez.

# Le protocole TCP

- Des applications importantes (HTTP, SMTP, FTP ...) ont besoin d'échanger de gros volumes de façon fiable.
- Une trame Ethernet ne peut transmettre, au maximum, que 1500 octets donc le message issu d'une application doit être découpé.
- Pour assurer une bonne transmission il faut ajouter à UDP :
  - Un accusé de réception
  - L'horodatage des paquets
  - La remise en séquence des paquets arrivées dans le désordre (le désordre n'est pas une erreur)
  - La possibilité de réémettre un paquet perdu ou erroné
- Ces ajouts plus d'autres qui sortent du cadre de ce cours conduisent au protocole TCP

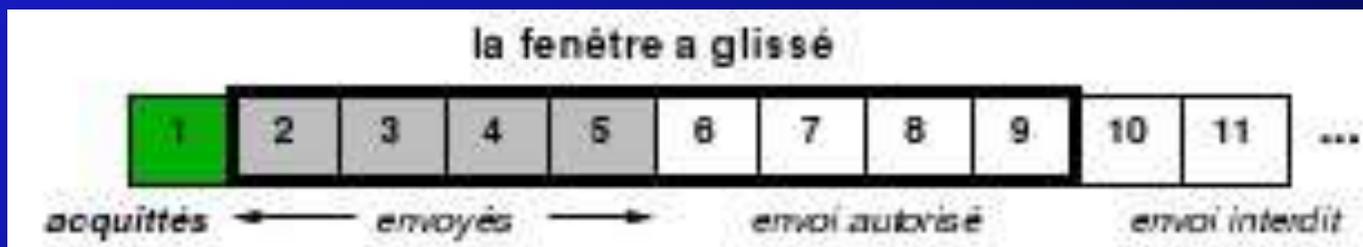
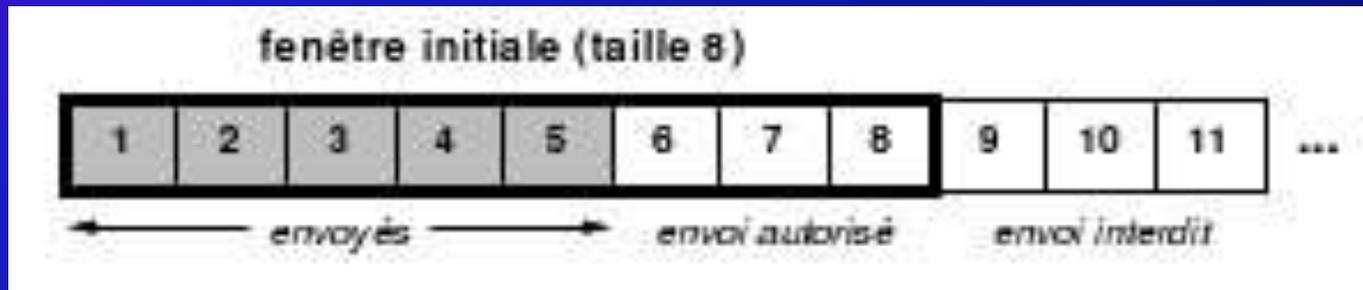
# Accusé de réception TCP

- L'émetteur d'un paquet attend une confirmation de réception de la part du récepteur avant d'envoyer un autre paquet
- Le récepteur accuse réception d'un paquet en envoyant un ACK
- C'est un protocole de type envoyer et attendre



# Fenêtre TCP

- Pour améliorer la vitesse de transmission on peut envoyer plusieurs paquets sans attendre l'ACK de chacun d'eux.
- Par exemple on peut envoyer 8 paquets à la suite. La taille de la fenêtre est donc de 8
- A réception du premier ACK en décale la fenêtre de 1 paquet



# Fenêtre TCP

Émetteur

Réseau

Récepteur

Émission paquet 1

Émission paquet 2  
avant réception ACK1

Émission paquet 3  
avant réception ACK2

Réception ACK paquet 1

Réception ACK paquet 2

Réception ACK paquet 3

Réception paquet 1

Réception paquet 2

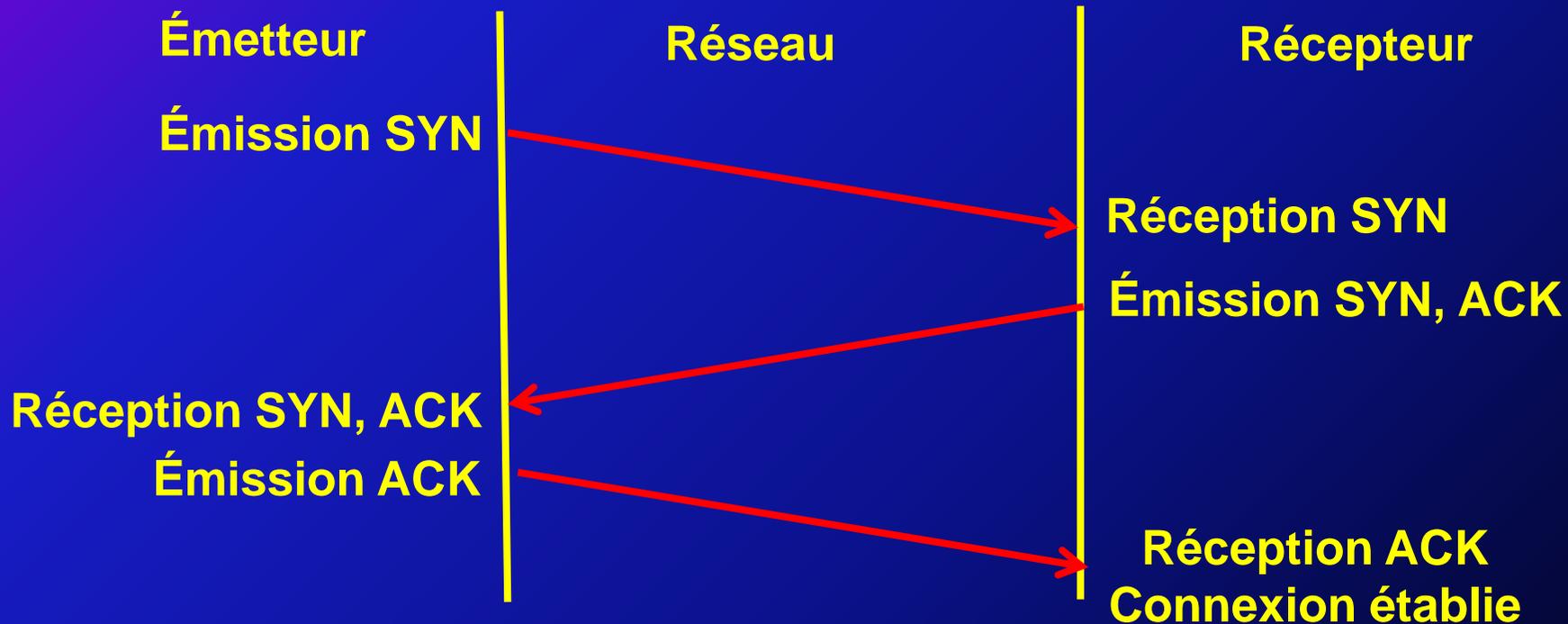
Réception paquet 3

# Connexion TCP

- A la différence de UDP on ne peut pas envoyer de message directement à une adresse. Il faut que préalablement le client établisse une connexion.
- Le serveur effectue une ouverture passive en écoutant sur un port dépendant du service (ex : HTTP=80) et en attendant qu'un client s'y connecte.
- Le client effectue une ouverture active en demandant l'établissement d'une connexion entre son adresse et celle du serveur. Le port du client est attribué de manière aléatoire.
- Une connexion est identifiée par le quadruplet formé par l'adresse de ses deux extrémités : adresse IP locale, port local, adresse IP distante, port distant.
- Chaque connexion est unique. Elles sont gérées indépendamment les unes des autres.
- Chaque connexion dispose de ses propres tampons en émission / réception et de chaque côté.
- Une fois la connexion établie le client et le serveur peuvent échanger des données à l'intérieur de ce circuit virtuel.

# Établissement d'une Connexion TCP

- Processus en 3 étapes



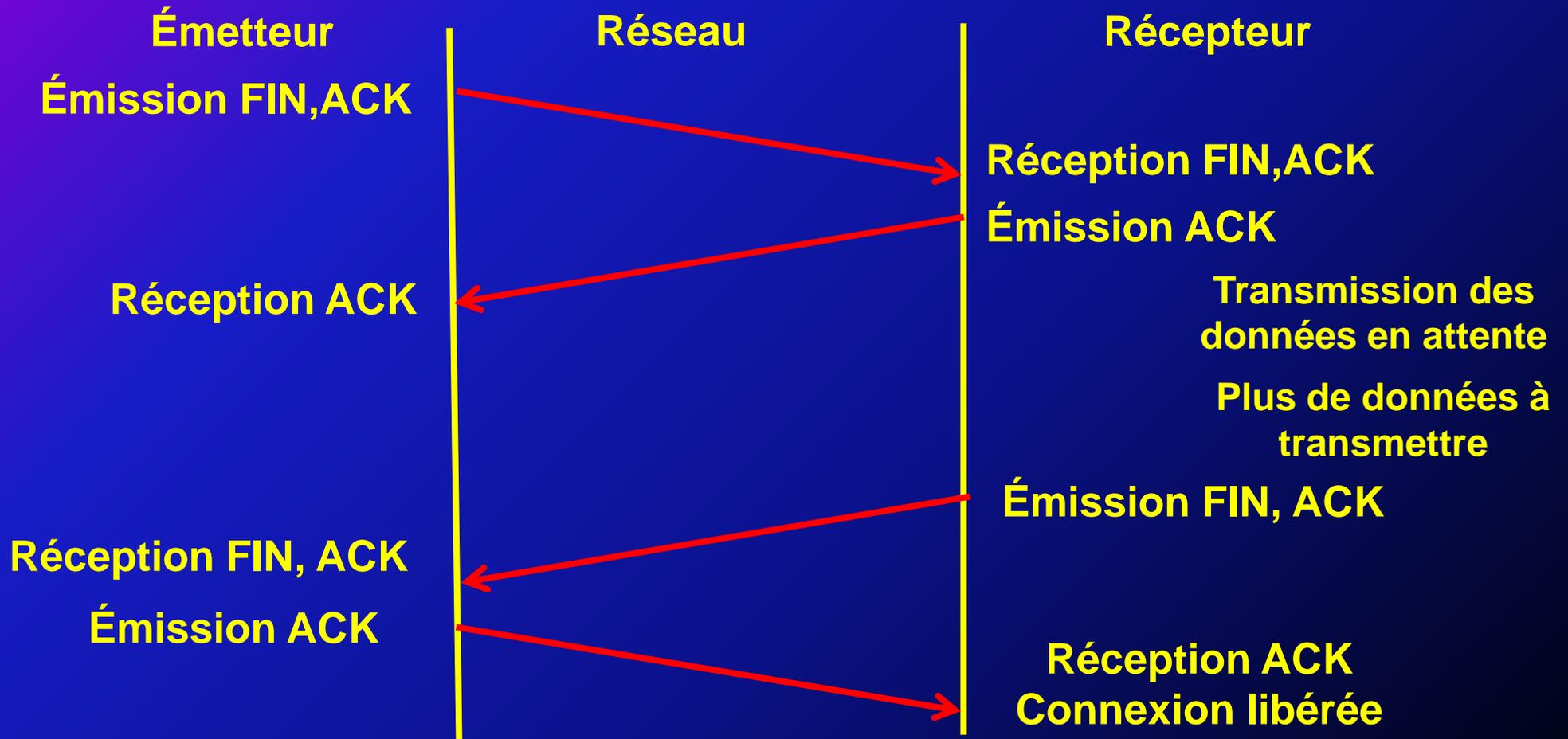
# Libération d'une connexion TCP

Il peut être mis fin à une connexion TCP soit du côté client soit du côté serveur. Une fermeture simultanée est possible mais rarissime

- L'émetteur envoie une demande de fin : flags FIN, ACK
- À part les données en attente d'envoi avant l'émission de la demande de fin plus aucun envoi ne sera accepté par TCP
- Le récepteur acquitte la demande de fin : flag ACK
- L'émetteur se met en attente de la réception d'éventuelles données non encore parvenues
- Lorsque il n'y a plus de données à transmettre à l'émetteur, le récepteur envoie sa propre demande de fin : flags FIN, ACK
- L'émetteur acquitte la demande de fin : flag ACK
- La connexion est close

# Libération d'une connexion TCP

La connexion libérée lorsque chaque côté a indiqué qu'il n'avait plus de données à transmettre.



# Protocole HTTP Connexion

## Établissement de la connexion (fichier html.pcap) trames 1 à 3

The image shows a Wireshark capture of an HTTP connection establishment sequence. The packet list pane shows the following packets:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.4.72	192.168.4.127	TCP	66	49221 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.000467	192.168.4.127	192.168.4.72	TCP	66	http > 49221 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=16
3	0.000512	192.168.4.72	192.168.4.127	TCP	54	49221 > http [ACK] Seq=1 Ack=1 win=17520 Len=0
4	0.000632	192.168.4.72	192.168.4.127	HTTP	407	GET / HTTP/1.1
5	0.000749	192.168.4.127	192.168.4.72	TCP	54	http > 49221 [ACK] Seq=1 Ack=354 win=6912 Len=0
6	0.001274	192.168.4.127	192.168.4.72	HTTP	538	HTTP/1.1 200 OK (text/html)
7	0.199672	192.168.4.72	192.168.4.127	TCP	54	49221 > http [ACK] Seq=354 Ack=485 win=17036 Len=0

Annotations with arrows pointing to the packets:

- Client → Serveur : demande de connexion = SYN (points to packet 1)
- Serveur → Client : acquittement de la demande = SYN-ACK (points to packet 2)
- Client → Serveur : Connexion établie = ACK (points to packet 3)

Packet details for Frame 1:

- Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
- Ethernet II, Src: Azurewav\_5e:df:be (e0:b9:a5:5e:df:be), Dst: cadmusCo\_ae:38:3e (08:00:27:ae:38:3e)
- Internet Protocol Version 4, Src: 192.168.4.72 (192.168.4.72), Dst: 192.168.4.127 (192.168.4.127)
- Transmission Control Protocol, Src Port: 49221 (49221), Dst Port: http (80), Seq: 0, Len: 0

Hex dump for Frame 1:

```
0000 08 00 27 ae 38 3e e0 b9 a5 5e df be 08 00 45 00  ..'.8>.. ^.....E.
0010 00 34 05 28 40 00 80 06 6b 84 c0 a8 04 48 c0 a8  .4.(@... k....H..
0020 04 7f c0 45 00 50 c6 ff 05 b7 00 00 00 00 80 02  ...E.P.....
0030 20 00 37 b2 00 00 02 04 05 b4 01 03 03 02 01 01  .7.....
0040 04 02 ..
```

# Protocole HTTP Requête

Envoie de la requête HTTP trames 4 et 5

Client → Serveur : Requête HTTP ( GET )

Serveur → Client : Acquittement de la requête = ACK

The screenshot shows a Wireshark capture of network traffic. The filter is set to 'ip.addr == 192.168.4.127'. The packet list shows the following:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.4.72	192.168.4.127	TCP	66	49221 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.000415	192.168.4.127	192.168.4.72	TCP	66	http > 49221 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=16
3	0.000512	192.168.4.72	192.168.4.127	TCP	54	49221 > http [ACK] Seq=1 Ack=1 win=17520 Len=0
4	0.000612	192.168.4.72	192.168.4.127	HTTP	407	GET / HTTP/1.1
5	0.000749	192.168.4.127	192.168.4.72	TCP	54	http > 49221 [ACK] Seq=1 Ack=354 win=6912 Len=0
6	0.001274	192.168.4.127	192.168.4.72	HTTP	538	HTTP/1.1 200 OK (text/html)
7	0.199672	192.168.4.72	192.168.4.127	TCP	54	49221 > http [ACK] Seq=354 Ack=485 win=17036 Len=0

The packet details pane shows the expanded content of the selected packet (No. 4):

```
Hypertext Transfer Protocol
GET / HTTP/1.1\r\n
Host: 192.168.4.127\r\n
User-Agent: Mozilla/5.0 (windows NT 6.1; WOW64; rv:15.0) Gecko/20100101 Firefox/15.0.1\r\n
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3\r\n
Accept-Encoding: gzip, deflate\r\n
Connection: keep-alive\r\n
Pragma: no-cache\r\n
Cache-Control: no-cache\r\n
\r\n
[Full request URI: http://192.168.4.127/]
```

# Protocole HTTP Réponse

Réception et affichage sur le navigateur trames 6 et 7

Serveur → Client : Envoie de la page HTML

Client → Serveur : Acquittement de la réception = ACK

The screenshot shows the Wireshark interface with a filter set to 'ip.addr == 192.168.4.127'. The packet list pane shows several packets, with packet 6 selected. Packet 6 is an HTTP 200 OK response from the server to the client. The packet details pane shows the content of the response, which is a text/html document.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.4.72	192.168.4.127	TCP	66	49221 > http [SYN] Seq=0 win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.000465	192.168.4.127	192.168.4.72	TCP	66	http > 49221 [SYN, ACK] Seq=0 Ack=1 win=5840 Len=0 MSS=1460 SACK_PERM=1 WS=16
3	0.000512	192.168.4.72	192.168.4.127	TCP	54	49221 > http [ACK] Seq=1 Ack=1 win=17520 Len=0
4	0.000632	192.168.4.72	192.168.4.127	HTTP	407	GET / HTTP/1.1
5	0.000749	192.168.4.127	192.168.4.72	TCP	54	http > 49221 [ACK] Seq=1 Ack=354 win=6912 Len=0
6	0.001127	192.168.4.127	192.168.4.72	HTTP	538	HTTP/1.1 200 OK (text/html)
7	0.019672	192.168.4.72	192.168.4.127	TCP	54	49221 > http [ACK] Seq=354 Ack=485 win=17036 Len=0

Content-encoded entity body (gzip): 146 bytes -> 177 bytes

Line-based text data: text/html

```
<html><body><h1>It works!</h1>\n
<p>This is the default web page for this server.</p>\n
<p>The web server software is running but no content has been added, yet.</p>\n
</body></html>\n
```

# Protocole HTTP Fin

Clôture de session par le serveur (suite à fermeture du navigateur client) frames 8 à 11

The image shows a Wireshark capture of an HTTP session termination sequence. The main pane displays a list of 11 packets. Packets 8, 9, 10, and 11 are highlighted in green. Packet 8 is a TCP FIN, ACK from the client to the server. Packet 9 is a TCP ACK from the server to the client. Packet 10 is a TCP FIN, ACK from the client to the server. Packet 11 is a TCP ACK from the server to the client. The packet details pane shows the selected packet (No. 8) with the following information:

- Internet Protocol Version 4, Src: 192.168.56.101 (192.168.56.101), Dst: 192.168.56.1 (192.168.56.1)
- Transmission Control Protocol, Src Port: http (80), Dst Port: groove-dpp (1211), Seq: 485, Len: 0
- Source port: http (80)
- Destination port: groove-dpp (1211)
- [Stream index: 0]
- Sequence number: 485 (relative sequence number)
- Acknowledgement number: 353 (relative ack number)
- Header length: 20 bytes
- Flags: 0x011 (FIN, ACK)

The packet bytes pane shows the raw data of the packet, including the FIN and ACK flags.

Annotations with arrows point to the following packets:

- Packet 8: Envoi FIN, ACK par le serveur
- Packet 9: Acquittement par le client
- Packet 10: Envoi FIN, ACK par le client
- Packet 11: Acquittement par le serveur

# Protocole HTTP données > 1500 octets

Le paquet TCP contient le n° de l'ACK (il n'y en a qu'un pour tout le transfert) et le n° de la séquence suivante.

The screenshot shows the Wireshark interface with a filter set to 'http'. The packet list pane displays several HTTP packets, all with a length of 1506 bytes and the info 'Continuation or non-HTTP traffic'. Packet 30 is selected. The packet details pane shows the Transmission Control Protocol (TCP) segment with the following fields:

- Source port: http (80)
- Destination port: matrix-vnet (4360)
- [Stream index: 1]
- Sequence number: 8763 (relative sequence number)
- [Next sequence number: 10215 (relative sequence number)]
- Acknowledgement number: 308 (relative ack number)
- Header length: 20 bytes

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII portion of the data is:

```
...^.... !..L..E.  
..CH@... ..*.(  
..P....j ..._PP.  
.7....5 ...:5..u  
3-.Tdb.. m.....  
..V...F. @.5.....  
h+6....q .L...tT.  
.KU.*V.. SU..)S..  
f.A....n ..f.....  
.....T. ...-V...  
.....
```

# TP SMTP

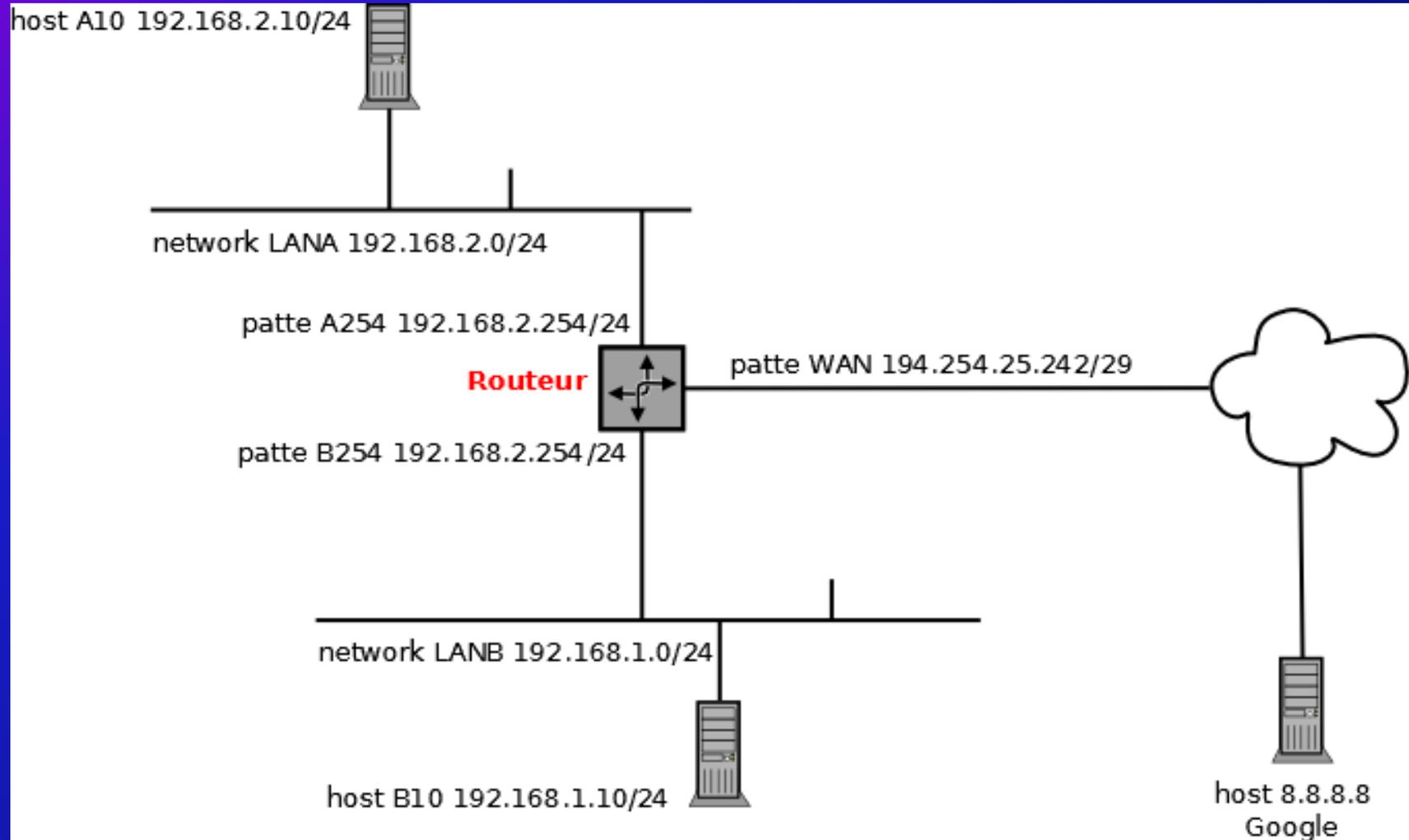
- Quel est le rôle du service SMTP ?
- Quel est le protocole de niveau 3 utilisé ?
- Quel est le port d'écoute ?
- Lancer WireShark à l'écoute sur votre interface (mode promiscuous disable)
- Établir une connexion (putty en mode raw) sur le serveur 192.168.0.254
- Taper les commandes suivantes :
  - ehlo sin.sti2d.org
  - quit
- Analyse du trafic

# Routage

- Lorsque un host client doit contacter un host serveur qui n'est pas sur le même réseau (la partie network des adresses IP est différente) il faut indiquer au host client comment atteindre le host serveur et à ce dernier comment renvoyer les données au host client. C'est là qu'intervient la notion de routage.
- Ex : client 192.168.1.10/24 → serveur 192.168.2.10/24. Ces deux hosts sont sur des réseaux différents, il faudra indiquer au host 10 du réseau 192.168.1.0 comment contacter le host 10 du réseau 192.168.2.0
- La décision de routage se déroule en 3 étapes :
  1. Vérifier si il existe une route spécifique. Si non étape suivante
  2. Vérifier si il existe une route dans une table de routage. Si non étape suivante
  3. Vérifier si il existe une route par défaut. Si non envoyer une erreur par ICMP

# Routage

## Exemple de réseau



# Routage

- Le host A10 du LAN A a comme route par défaut la patte A254 du routeur
- Le host B10 du LAN B a comme route par défaut la patte B254 du routeur
- Le routeur a la table de routage suivante
  - → LAN A passer par la patte A254
  - → LAN B passer par la patte B254
  - Sinon route par défaut = WAN
- Exemple 1 : depuis A10 on veut atteindre Google
  - Sur A10 : route par défaut jusqu'au routeur patte A254
  - Sur le routeur : route par défaut → sortie par la patte WAN jusqu'à Google
- Exemple 2 : depuis A10 on veut atteindre B10
  - Sur A10 : route par défaut jusqu'au routeur patte A254
  - Sur le routeur : table de routage → sortie par la patte B254
- **Remarques importantes** :
  - la route par défaut est toujours sur le même réseau que le host
  - L'adresse de destination par défaut est 0.0.0.0

# Routage

---

- Sous Windows\$ la table de routage s'obtient indifféremment par les commandes :
  - route print
  - netstat -r
- Sous Linux :
  - netstat -r
  - L'ajout du commutateur n permet d'éviter la résolution de nom
- Sur votre PC détailler la table de routage

# Routage TP SR3

---