# Administration Réseau

2016 – 2017

Ludovic Néve

l.neve24@gmail.com

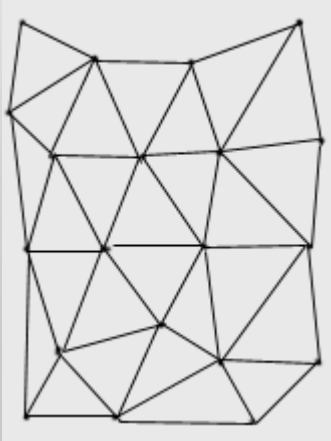
# Introduction

# Plan

- Quelques chiffres et informations
- Définitions
- Le modèle OSI

# Quelques chiffres et informations

- Quand on parle d'Internet on parle des réseaux TCP/IP.
- Internet est un réseau maillé.
- Dans les années 90, Internet va accroitre grâce à la création de HTTP et HTML qui permettent la création des pages Web.



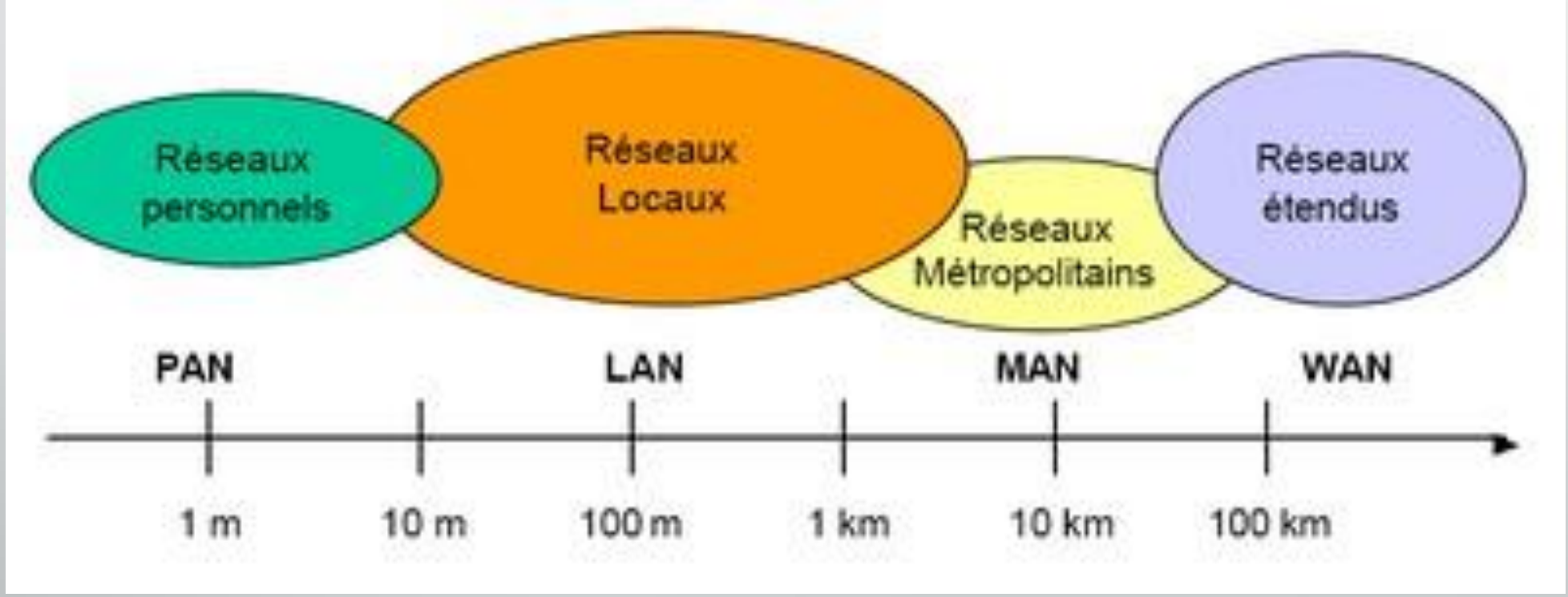
# Quelques chiffres et informations

- En 2010, Internet c'est:
  - 2,53 milliards d'internautes
  - 200 millions de serveurs
  - 42% des internautes viennent d'Asie
  - La France représente 6% des internautes du monde
  - 1 personne sur 3 a accès à Internet
  - La progression a été de 4,5% entre 2000 et 2010 dans le monde.

# Définitions

- **Réseau** : Ensemble d'éléments matériels et logiciels, qui met en relation physique et logique, des ordinateurs et leurs périphériques, à l'intérieur d'un site géographique.
- **PAN** : Personnal Area Network : désigne un type de réseau informatique restreint en terme d'équipements, généralement mis en œuvre dans un espace d'une dizaine de mètres.
- **LAN** : Local Area Network : C'est un réseau à une échelle géographique relativement restreinte, par exemple une salle informatique, une habitation particulière, un bâtiment ou un site d'entreprise.
  Dans le cas d'un réseau d'entreprise, on utilise aussi le terme RLE pour réseau local d'entreprise.
  - **WLAN** : Wireless Local Area Network : c'est un réseau local sans fil
- **MAN** : Metropolitan Area Network : désigne un réseau composé d'ordinateurs habituellement utilisé dans les campus ou dans les villes.
- **WAN** : Wide Area Network : est un réseau informatique couvrant une grande zone géographique, typiquement à l'échelle d'un pays, d'un continent, voire de la planète entière.

# Définitions



# Définitions

- **Serveurs** : Ce sont des machines rapides, puissantes et fiables en charge de la mise à disposition et de la gestion de ressources communes de travail. Certains serveurs sont affectés à un rôle précis => serveurs dédiés.
- **Clients:** Ce sont souvent des ordinateurs personnels ou des appareils individuels qui émettent des requêtes vers les serveurs.
  - Note: Un machine peut être client et serveur.
- **Services** : à différencier des services applicatifs (ou applications). Il s'agit de services essentiels pour le réseau dont le but est de faciliter l'installation et la configuration du réseau. Parmi ces services il y a le service de nommage, le service de configuration des  hôtes TCP/IP, la messagerie électronique, les serveurs de fichiers, les serveurs d'impressions …

# Définitions

- **Topologie** : c’est une définition de l'architecture d'un réseau, définissant les connexions entre les hôtes et une hiérarchie éventuelle entre eux.
- **Protocole** : c’est un ensemble de règles et de procédures permettant de définir un type de communication particulier.

# Le modèle OSI

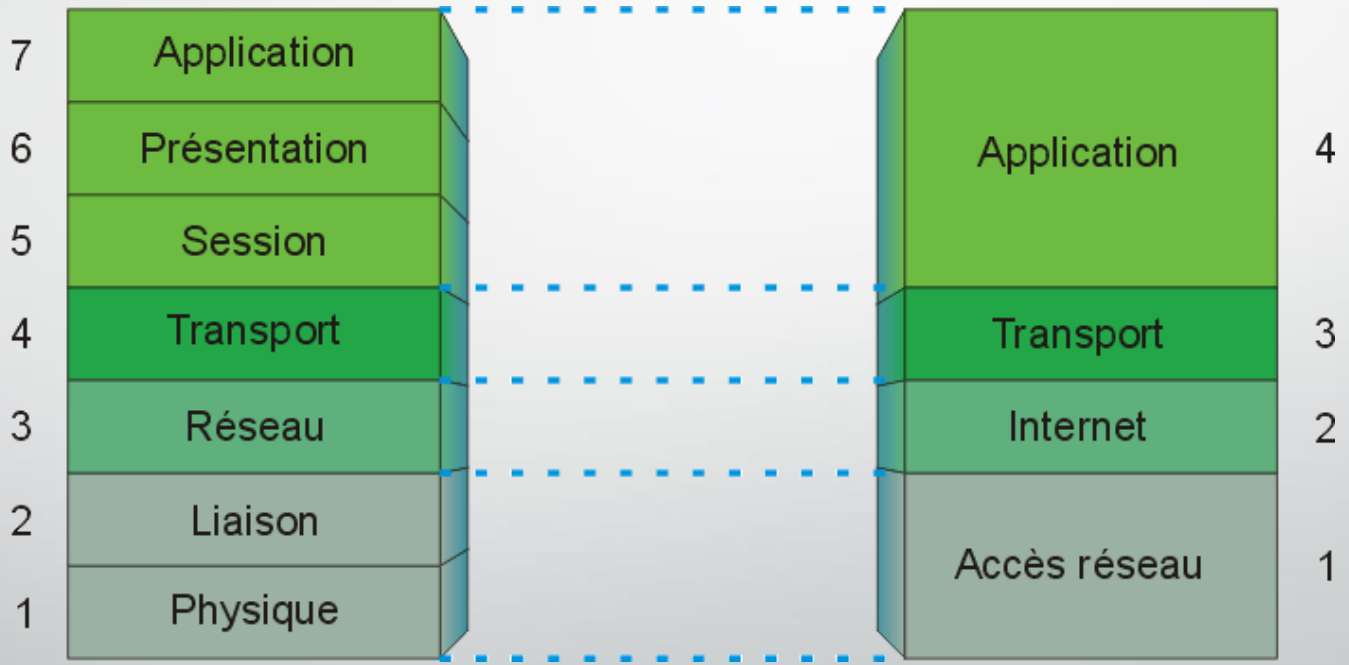
- Nécessité de mettre de l'ordre dans les communications.
- Retour à la base des communications (parole, téléphone, courrier…) pour définir un modèle

➔ Création du modèle OSI en 1984

# Le modèle OSI

- Le modèle OSI est une norme précisant comment les ordinateurs doivent communiquer entre eux.
  Il indique comment travailler si on veut mettre en place un réseau et donne des indications pour les constructeurs de matériels réseau.
- C'est un modèle en couche. Chacune des couches a un rôle défini et ne fait rien d'autre.
- Il existe 7 couches qui couvrent tous les besoins d'une communication.
  Les couches sont indépendantes et ne peuvent communiquer qu'avec les couches adjacentes.
- Le modèle OSI est un modèle théorique, dans la pratique c'est TCP/IP.
- Dans les réseaux TCP/IP, les couches 5 et 6 ne sont pas utilisées.

# Le modèle OSI



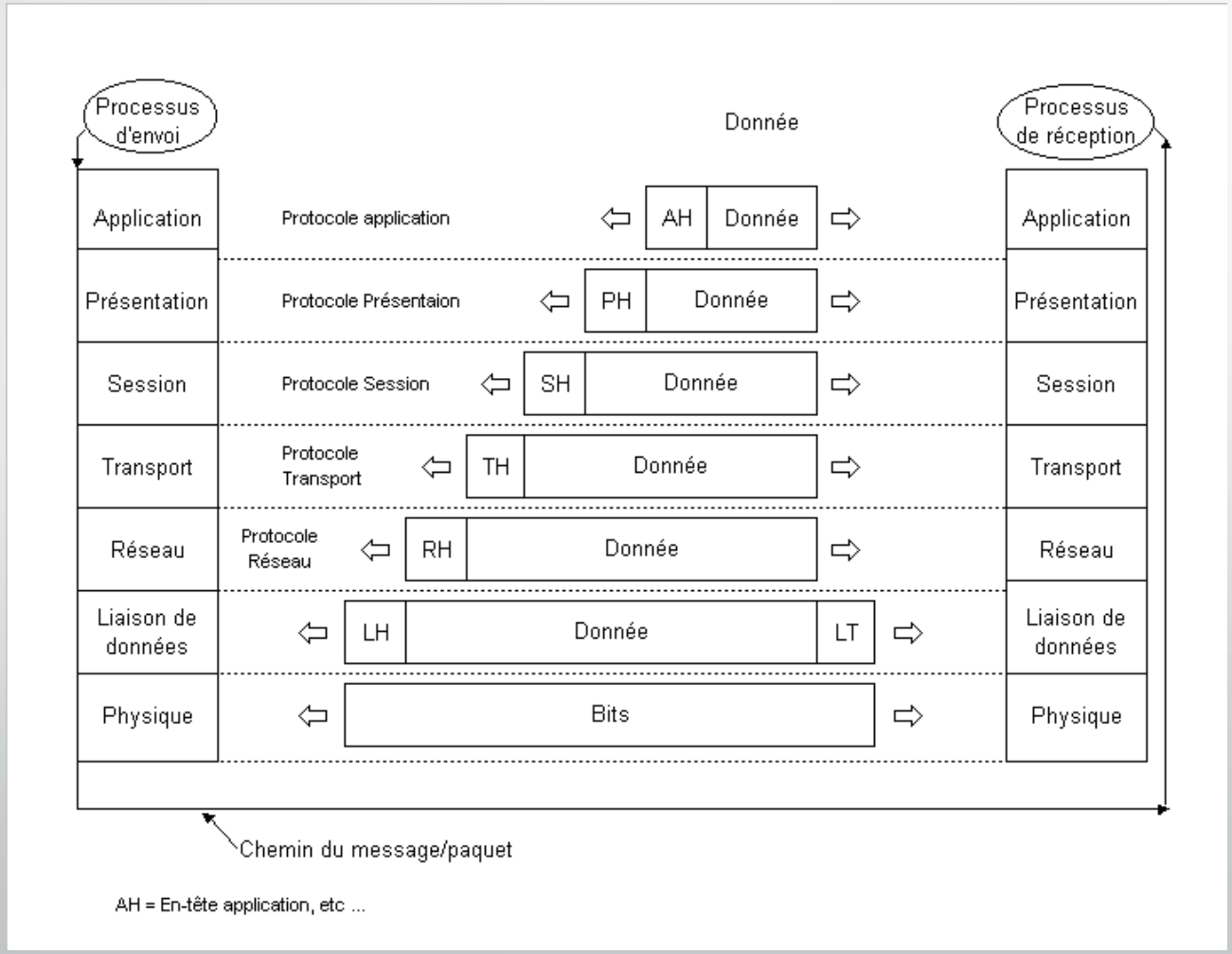
# Le modèle OSI

- *Couche : Nom : Rôle*
- Couche 1 : Physique : Support de transmission
- Couche 2 : Liaison : Connecter les machines entre elles sur un réseau local + détection des erreurs
- Couche 3 : Réseau : Interconnecter les réseaux + fragmenter les paquets
- Couche 4 : Transport : Gérer les connexions applicatives + garantir la connexion
- Couche 5 et 6 : Pas utilisées dans TCP/IP
- Couche 7 : Application : RAS

# Le modèle OSI

- Lors d'une communication sur le réseau, les données vont traverser successivement les couches du modèle OSI jusqu'à la couche 1 ou les données seront envoyées dans le réseau.
- Chaque couche va ajouter des informations utiles à l'attention de la machine distante : ces informations sont les en-tête et le principe s'appelle **l'encapsulation**

# L’encapsulation



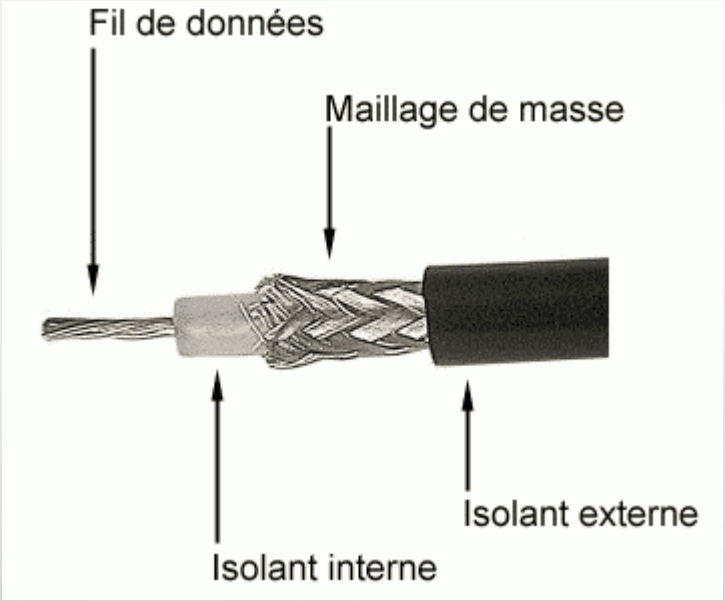
# Couche 1 : La couche Physique

# Couche 1 : Physique

- La couche physique est la première couche du modèle OSI.
- Elle fournit le support de transmission : les câbles (ou l'air dans le cas du wifi)
- Elle est chargée de la transmission effective des signaux électriques ou optiques entre les interlocuteurs.

# Les câbles : câbles coaxiaux

- Câbles coaxiaux :



# Les câbles : câbles coaxiaux

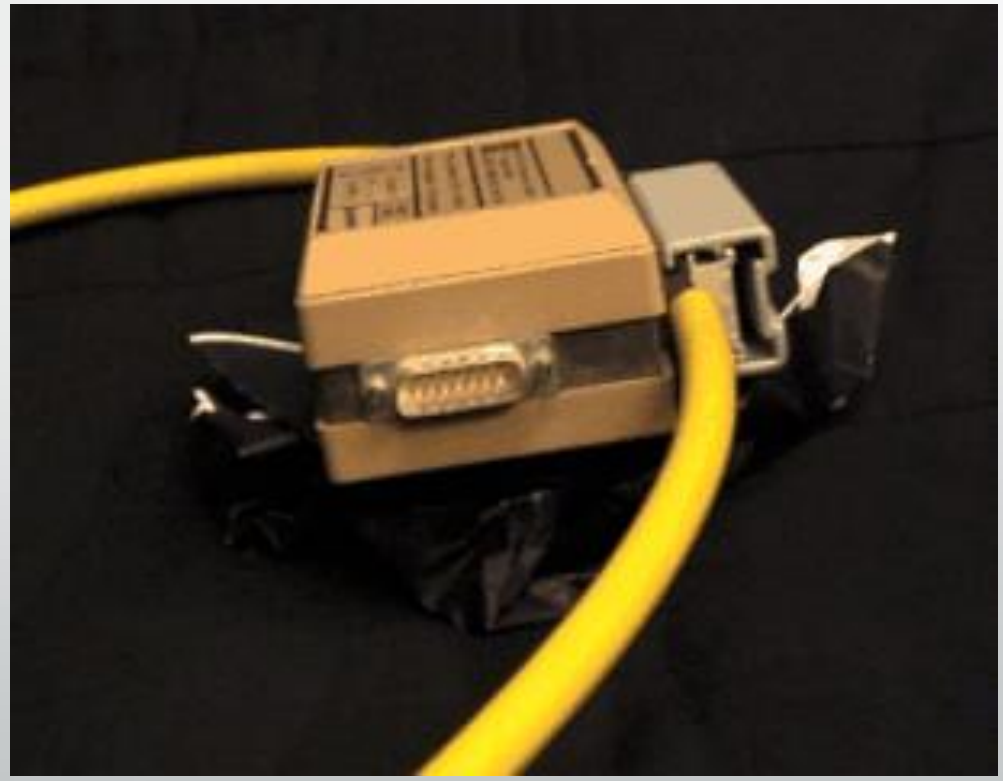
- On fait circuler le signal électrique dans le fil de données central. On obtient le signal électrique en faisant la différence de potentiel entre le fil de données et la masse.
- On se sert du maillage de masse, ou grille, pour avoir le signal de référence 0V
- Les noms scientifiques du câble coaxial sont 10B2 et 10B5 (dix base deux et dix base cinq) : 10 pour le débit, B pour Bande de Base et 2 ou 5 pour la taille maximale en centaines de mètres du réseau.

# Les câbles : câbles coaxiaux

- 10B5 :
  - On se connecte au réseau à l'aide de prise Vampire.
  - Ces prises comportent une pointe en métal qui rentre en contact avec le câble de données.
- 10B2 :
  - On utilise de Té BNC et des bouchons.
  - Les bouchons permettent de fermer le réseau ( sur l'un des côté du Té BNC)
  - On connecte la carte réseau sur l'un des côtés du Té BNC et le câble sur les autres côtés.

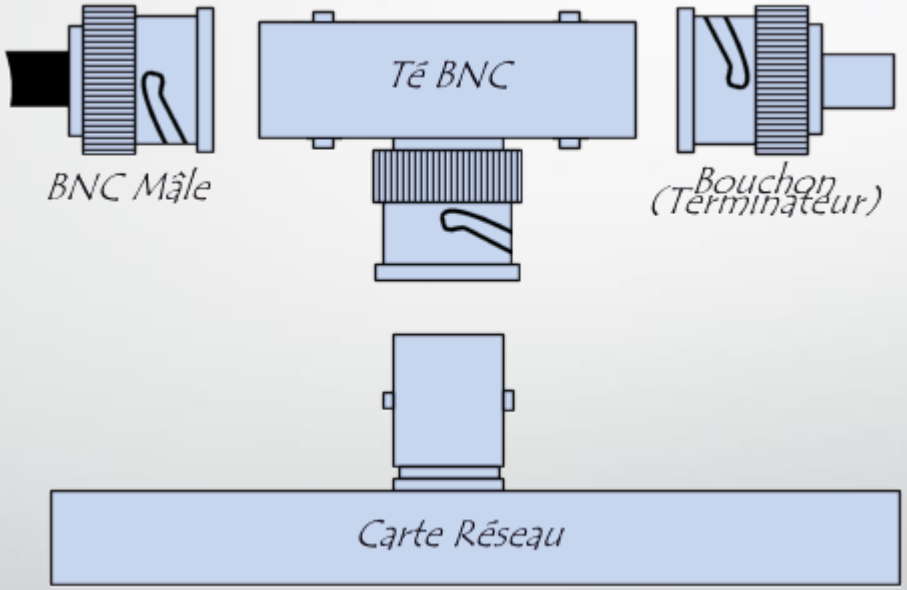
# Les câbles : câbles coaxiaux

- Prise Vampire:



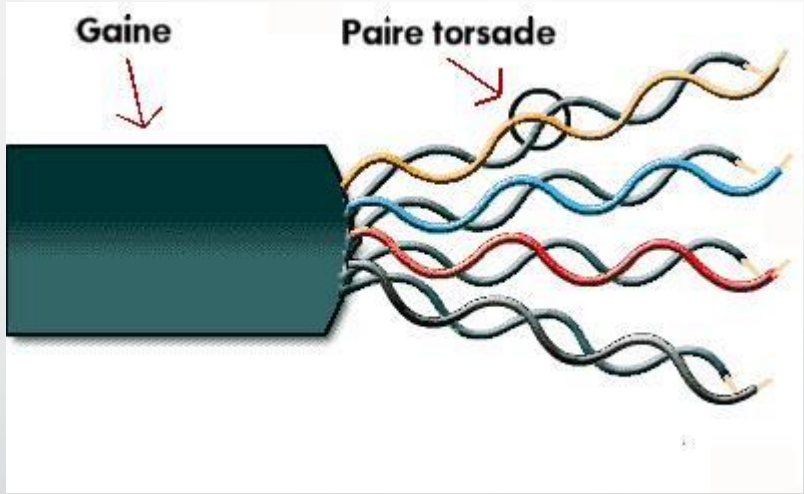
# Les câbles : câbles coaxiaux

- Prise BNC:



# Les câbles : câble à paires torsadées

- Paires torsadées:



# Les câbles : câble à paires torsadées

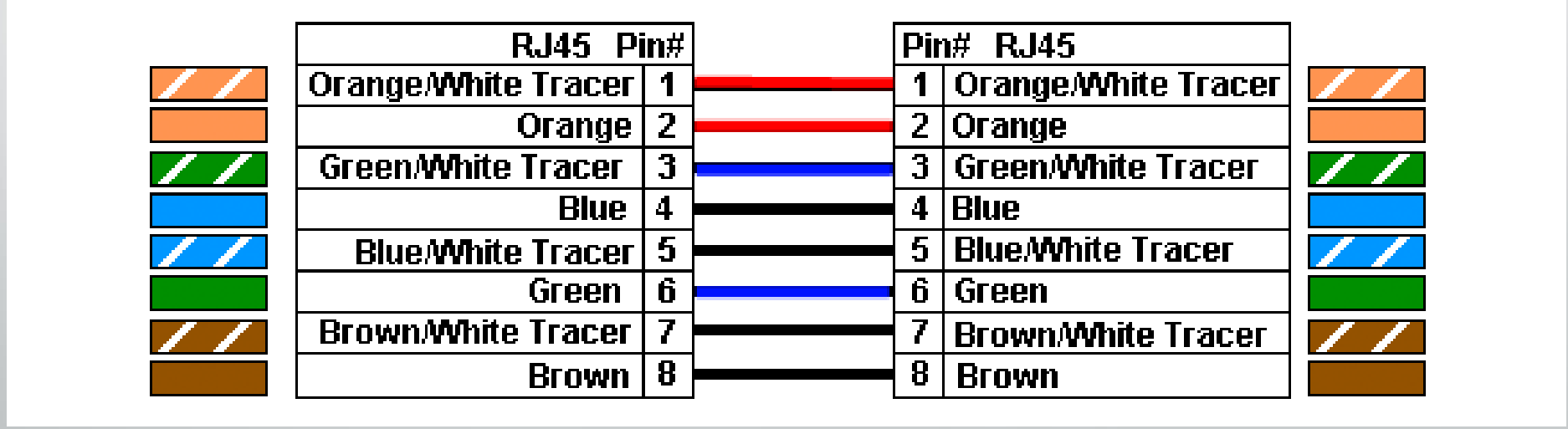
- Le câble à paires torsadées est composé de huit fils torsadés deux à deux.
- On utilise en général 4 fils sur les 8 : Une paire pour envoyer et une paire pour recevoir.
- Le nom scientifique du câble à paires torsadés est 10/100/1000BT (dix base T) : 10/100/1000 en fonction du débit, B pour Bande de Base et T pour paire torsadée ou Twisted.
- C'est la connexion la plus répandue.

# Les câbles : câble à paires torsadées

- On branche le câble à paires torsadées à l'aide d'une prise RJ45:

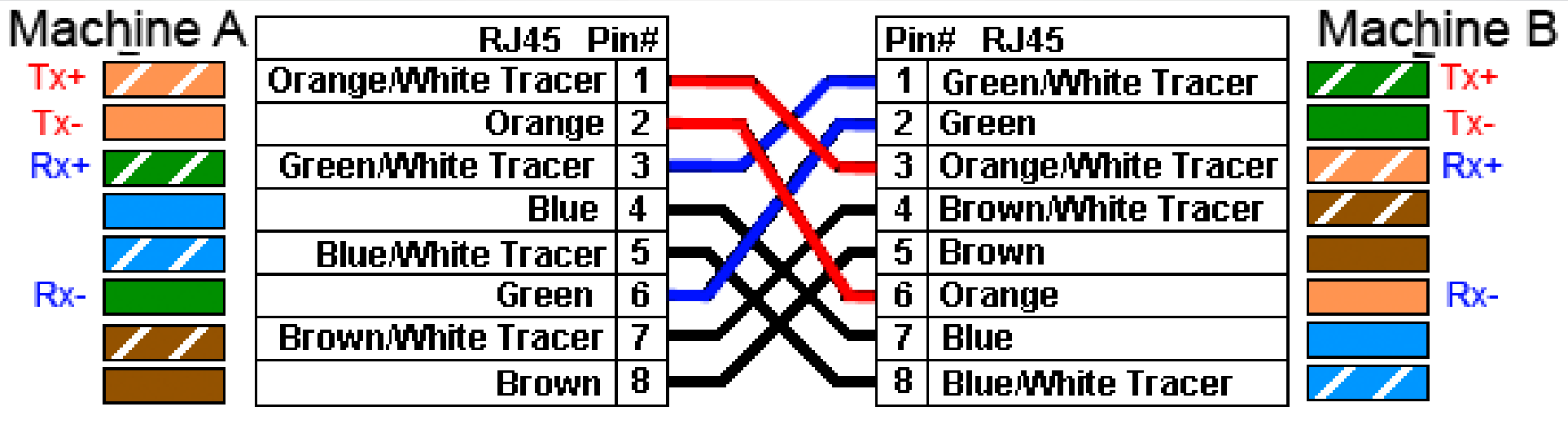
# Les câbles : câble à paires torsadées

- Le fait de ne pas utiliser toutes les paires ne nous permet pas de faire ce qu'on veut: Il faut utiliser les fils 1, 2, 3 et 6.



# Les câbles : câble à paires torsadées

- Pour connecter deux machines de même type, on utilise un câble croisé.
- C'est un câble qui inverse le sens des connexions afin de pouvoir aligner la transmission d'une machine avec la réception d'une autre machine et inversement.
- La plupart des cartes réseau d'aujourd'hui ont la capacité d'inverser les connexions, nous permettant d'utiliser un câble droit.

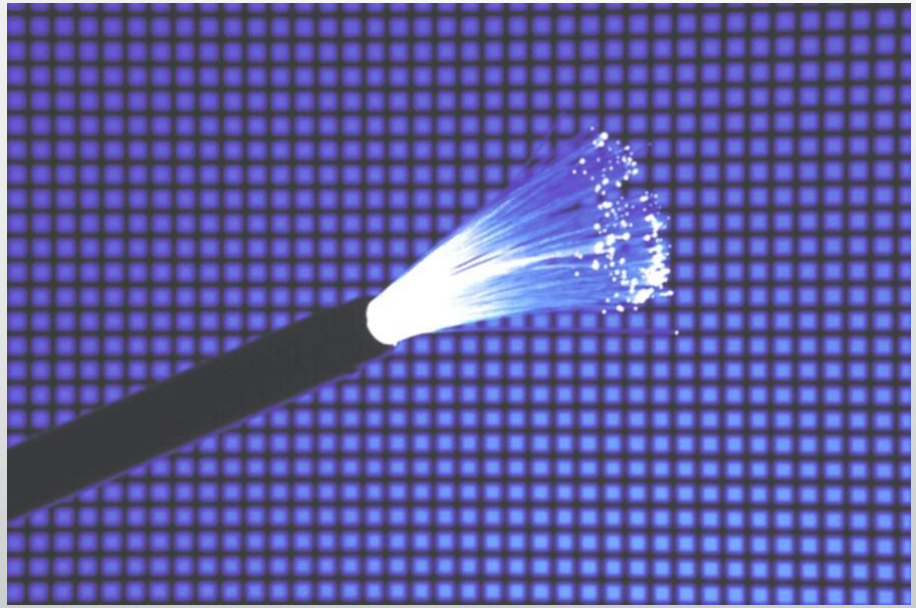


# Les câbles : câble à paires torsadées

- Pour connecter plusieurs machines entre elles sur la couche 1, on utilise un HUB.
- C'est une machine qui possède plusieurs prises RJ45 femelles permettant de relier les machines entre elles.

# Les câbles : fibre optique

- Fibre optique:



# Les câbles : fibre optique

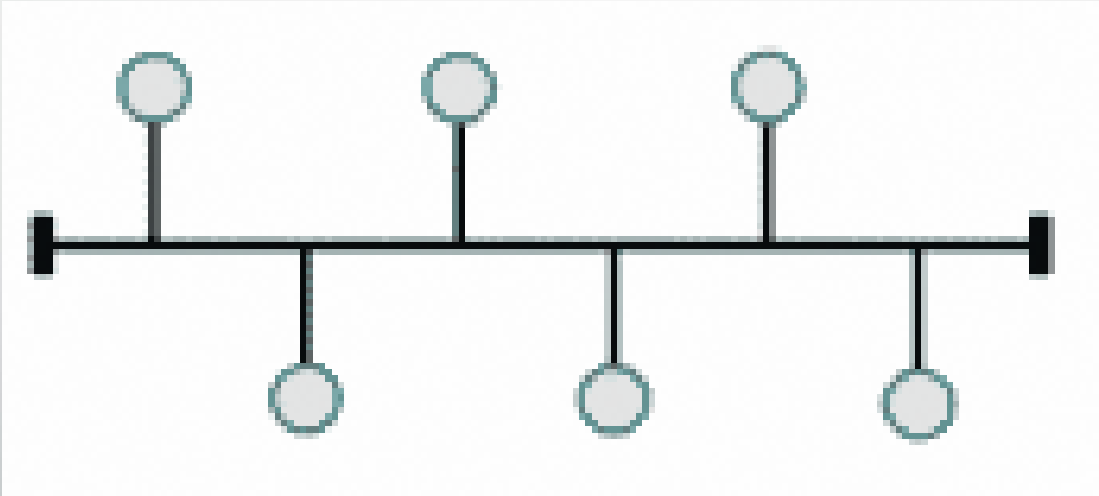
- Avec la fibre, on transmet les 0 et les 1 à l'aide de la lumière.
- Il existe deux types de fibre optique: la fibre monomode et la fibre multimode.
- La fibre monomode est plus performante.
- Le nom scientifique de la fibre optique est 1000BF : 1000 pour le débit (Gigabit), B pour Bande de Base et F pour Fibre.
- Elle est principalement utilisée par les opérateurs et les grandes entreprises.

# Les Topologies

- C’est la manière utilisée pour brancher les machines entre elles.
- Il y a trois principales topologies:
  - Le Bus
  - L’anneau
  - L’étoile
- La topologie en étoile est celle qui est le plus utilisée de nos jours.

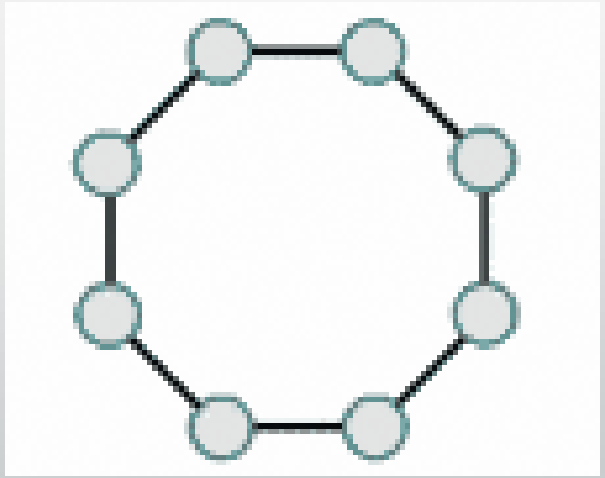
# Les Topologies

- La topologie en Bus:
  - Toutes les machines sont branchées sur le même câble.
  - Une seule machine parle à la fois.
  - La taille du réseau est contrainte par la taille du câble.



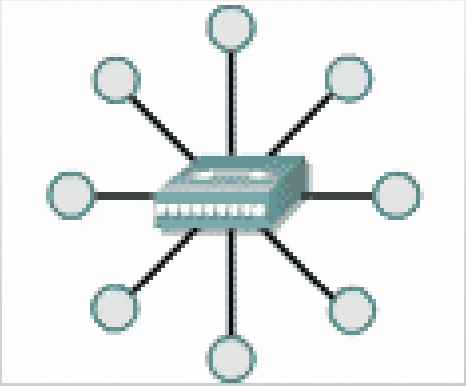
# Les Topologies

- La topologie en Anneau:
  - Toutes les machines sont branchées sur le même câble qui boucle sur lui-même.
  - Une seule machine parle à la fois quand elle possède le Jeton.
  - On ne lit le message que s'il nous est destiné quand on reçoit le Jeton.
  - Comme pour le Bus, la taille du réseau est contrainte par la taille du câble.



# Les Topologies

- La topologie en Etoile:
  - Les machines sont branchées sur un équipement central capable de relayer l'information.
  - Toutes les communications passent par le point central qui aiguille l'information.
  - Le nombre de machine dépend de la capacité de traitement du point central.
  - On peut relier plusieurs points centraux ensemble pour augmenter la taille du réseau.



# CSMA/CD

- Carrier Sense Multiple Access/ Collision Detection
- C'est une méthode qui permet de limiter les cas de collisions (pour les topologies en Bus notamment).
- Le principe:
  - On écoute pour savoir si le réseau est libre.
  - Si le réseau est libre on commence à émettre.
  - En cas de collision, on attend pour ré- émettre.
  - On réémet après avoir attendu une durée aléatoire.

# Couche 2 : La couche Liaison

# Couche 2 : Liaison

- La couche liaison est la seconde couche du modèle OSI.
- Elle permet aux machines de communiquer sur un réseau local.
- Son rôle secondaire est de détecter les erreurs.
- Afin de communiquer, nous devons distinguer les machines ; et pour ce faire nous avons une adresse pour la couche 2 : C'est l'adresse MAC !

# L’adresse MAC

- C’est l’adresse de la carte réseau.
- Elle est codée en hexadécimal sur 6 octets (soit 48 bits)
- Chaque adresse MAC est unique au monde ; Il en existe $2^{48}$ soit environ 280 mille milliards.
- Un constructeur peut acheter les 3 premiers octets pour identifier ses cartes réseau et dispose des 3 autres pour en faire ce qu’il veut.

# L’adresse MAC

- Il existe une adresse MAC particulière : **ff:ff:ff:ff:ff:ff**
- Il s’agit de l’adresse de broadcast.
- C’est une adresse universelle qui identifie n’importe quelle carte réseau.
- On peut l’utiliser pour envoyer un message à toutes les cartes réseau présentes sur un réseau en une seule fois.

# Le protocole de la couche 2 : Ethernet

- Pour communiquer sur la couche 2, il nous faut un « langage ».
- Ce langage s'appelle protocole et pour la couche 2 il s'agit d'Ethernet.
  - Ethernet n'est pas le seule protocole de couche 2 mais il est de loin le plus utilisé.
- L'usage d'un protocole est nécessaire pour définir des normes permettant d'établir des communications entre des machines de constructeurs différents ayant des OS différents sur un même réseau.
- Le protocole va définir le format des messages envoyés sur le réseau.
- En couche 2, ce message s'appelle une **trame**.

# Format d'une trame Ethernet

- La trame va contenir les informations suivantes:
  - Adresse MAC de l'émetteur
  - Adresse MAC du destinataire
  - Le message
  - Le protocole de couche 3
  - Le CRC

# Format d'une trame Ethernet

- L'adresse MAC du destinataire sera la première information de la trame Ethernet, permettant de savoir, lors de la réception, si un message nous est destiné ou non (si non le message sera simplement ignoré)
- Le CRC est une valeur mathématique représentative du message envoyé. C'est une valeur unique pour chaque message.
- Lors de l'envoi d'un message, la machine émettrice calcule le CRC et l'ajoute à la fin de la trame. La machine qui reçoit le message calcule le CRC à son tour et compare avec le CRC présent dans la trame reçue, si les deux CRC sont identiques, alors le message est validé.

# Format d'une trame Ethernet

- Voici à quoi ressemble la trame Ethernet qui circule sur le réseau:

| @ MAC DST | @ MAC SRC | PROTOCOLE COUCHE 3 | MESSAGE | CRC |
|---|---|---|---|---|

- Certains éléments de la trame Ethernet ne varient jamais, c'est l'en-tête Ethernet.
- La taille de l'en-tête Ethernet est de 18 octets.
- La taille minimale de la trame Ethernet est de 64 octets et la taille maximale est de 1518 octets.

# Le matériel de la couche 2 : Le Switch

- Le switch ou commutateur est un matériel qui permet de relier plusieurs machines entre-elles.
- Il porte plusieurs prises RJ45 sur lesquelles on connecte des machines à l'aide de câbles à paires torsadées.
- Le switch aiguille les trames à l'aide des adresses MAC. Pour cela il utilise une table qui associe l'adresse MAC et le port : c'est la **table CAM**.
- Le switch est un élément passif, il ne peut pas découvrir le réseau.

# Le matériel de la couche 2 : Le Switch

# La table CAM

- Elle se met à jour dynamiquement au fur et à mesure que le switch voit passer des trames.
- Lorsqu'une machine envoie une trame, il est facile d'associer son port et son adresse MAC (l'adresse source).
- Si l'adresse MAC de la destination est dans la table CAM alors le switch transmet la trame sur le bon port, sinon il transmet la trame à tous les ports à l'exception du port de la machine source.
  - La machine qui répondra, suite à la réception de cette trame, sera donc la destination et le switch mettra à jour sa table CAM avec cette nouvelle information.
- Les enregistrements de la tables CAM ont une durée de vie limitée: il s'agit d'un TTL.

# La table CAM

telnet | bash | bash | bash | bash

```
6509E#show mac-address-table
Legend: * - primary entry
        age - seconds since last seen
        n/a - not available

  vlan   mac address     type    learn     age              ports
------+----------------+--------+-----+----------+--------------------------
*  255   00d0.03e2.a000   dynamic  Yes          5   Gi5/2
*  ---   0000.0000.0000    static  No           -   Router
*  255   3333.0000.000d    static  Yes          -   Gi1/1,Gi1/2,Gi1/13,Gi1/14
                                                    Gi5/2,Router,Switch
*  255   3333.0000.0001    static  Yes          -   Switch
*    7   001c.251a.5886   dynamic  Yes          0   Gi1/13
*    7   0023.6924.c38c   dynamic  Yes         60   Gi1/14
*    7   0023.6948.b89c   dynamic  Yes         20   Gi1/2
*  ---   0000.0000.aaaa    static  No           -   Switch
*  255   3333.0000.0016    static  Yes          -   Switch
*    7   000f.f86d.d800    static  No           -   Router
*    7   0023.dfa0.cf50   dynamic  Yes          0   Gi1/1
*   50   3333.0000.000d    static  Yes          -   Gi1/1,Gi1/2,Gi1/13,Gi1/14
                                                    Gi5/2,Router,Switch
*    1   3333.0000.000d    static  Yes          -   Gi1/1,Gi1/2,Gi1/13,Gi1/14
                                                    Gi5/2,Router,Switch
*    7   3333.0000.000d    static  Yes          -   Gi1/1,Gi1/2,Gi1/13,Gi1/14
                                                    Gi5/2,Router,Switch
*   10   3333.0000.000d    static  Yes          -   Gi1/1,Gi1/2,Gi1/13,Gi1/14
                                                    Gi5/2,Router,Switch
*   20   3333.0000.000d    static  Yes          -   Gi1/1,Gi1/2,Gi1/13,Gi1/14
                                                    Gi5/2,Router,Switch
*   20   3333.0000.0001    static  Yes          -   Switch
*   10   3333.0000.0001    static  Yes          -   Switch
*    1   3333.0000.0001    static  Yes          -   Switch
*    7   3333.0000.0001    static  Yes          -   Switch
*   50   3333.0000.0001    static  Yes          -   Switch
*  255   000f.f86d.d800    static  No           -   Router
*   20   3333.0000.0016    static  Yes          -   Switch
*   10   3333.0000.0016    static  Yes          -   Switch
*    7   3333.0000.0016    static  Yes          -   Switch
*    1   3333.0000.0016    static  Yes          -   Switch
*   50   3333.0000.0016    static  Yes          -   Switch
```

# Les avantages du switch

- L'utilisation des paires torsadées qui permet de s'affranchir du CSMA/CD
  - Les cartes réseau fonctionnent en full-duplex.
- Le switch possède une mémoire capable de stocker les trames à destination d'une machine qui communique déjà sur le réseau.
- Les conversations sont isolées ce qui améliore la sécurité.

# Les VLAN

- Un VLAN est un LAN Virtuel.
- Il s'agit de la capacité à séparer les ports d'un switch dans des réseaux différents. De ce fait certains ports ne pourront plus communiquer ensemble.
- Les VLAN permettent de couper les switchs en plusieurs morceaux ce qui est pratique pour les gros réseaux et pour des raisons de sécurité.
- En théorie, il n'est pas possible de passer d'un VLAN à un autre.
  Mais il existe une technique qui permet de le faire : c'est le VLAN Hopping.
  - Les failles qu'exploitent cette technique ont été corrigées depuis, mais il est fort possible que l'on trouve de nouvelles failles le permettant.

# Couche 3 : La couche Réseau

# Couche 3 : Réseau

- La couche réseau est la troisième couche du modèle OSI.
- Elle permet aux machines de communiquer avec d'autres réseaux.
- Son rôle secondaire est la fragmentation des paquets.
- Pour distinguer les machines sur la couche 3, nous avons également une adresse : C'est **l'adresse IP** !

# Les adresses IP

- Il s'agit de l'adresse du réseau **et** de la machine.
- Une partie de l'adresse représente le réseau et l'autre partie identifie la machine sur ce réseau.
- Elle est codée sur 4 octets (32 bits).
- On utilise la notation décimale pointée XXX.XXX.XXX.XXX
  - Ex: 192.168.1.2
- L'adresse IP est codée en binaire au niveau de la machine.

# Les adresses IP

- A l'origine, les adresses IP étaient divisées en 5 classes:
  - **Classe A** : Une adresse IP de classe A dispose d'un seul octet pour identifier le réseau et peut comporter jusqu'à $2^{24}$ machines (soit 16 777 216). Le premier octet d'une adresse IP de classe A commence toujours par le bit *0*, il est donc compris entre 0 et 127.
  - **Classe B** : Une adresse IP de classe B dispose de deux octets pour identifier le réseau et peut comporter jusqu'à $2^{16}$ machines, (soit 65 536). Le premier octet d'une adresse IP de classe B commence toujours par la séquence de bits 10, il est donc compris entre 128 et 191.
  - **Classe C** : Une adresse IP de classe C dispose de trois octets pour identifier le réseau et peut comporter jusqu'à $2^{8}$ machines (soit 256). Le premier octet d'une adresse IP de classe C commence toujours par la séquence de bits 110, il est donc compris entre 192 et 223.
  - **Classe D** : Les adresses de classe D sont utilisées pour les communications multicast. Le premier octet d'une adresse IP de classe D commence toujours par la séquence de bits 1110, il est donc compris entre 224 et 239.
  - **Classe E** : Les adresses de classe E sont réservées à un usage non déterminé. Elles commencent toujours par la séquence de bits 1111, ils débutent donc en 240.0.0.0 et se terminent en 255.255.255.255.

# Les adresses IP

- Tableau des classes:

| Classes | Masque de sous-réseau par défaut | Plages d'adresses |
|---|---|---|
| Classe A | 255.0.0.0 (/8) | 0.0.0.0 -> 127.255.255.255 |
| Classe B | 255.255.0.0 (/16) | 128.0.0.0 -> 191.255.255.255 |
| Classe C | 255.255.255.0 (/24) | 192.0.0.0 -> 223.255.255.255 |
| Classe D | N/A | 224.0.0.0 -> 239.255.255.255 |
| Classe E | N/A | 240.0.0.0 -> 255.255.255.255 |

# Les adresses IP

- Avec les classes A, B et C, il est possible d'utiliser un certain nombre de bits de la partie machine pour définir des sous-réseau.

- Pour pouvoir déterminer la partie réseau de la partie machine de l'adresse IP, nous utilisons le **masque de sous-réseau**.

# Le masque de sous-réseau

- Il indique quelle est la partie réseau de l'adresse IP.
- Les bits à 1 dans le masque représentent la partie réseau.
- Dans un masque en binaire, il doit obligatoirement y avoir des 1 à gauche et des 0 à droite.
- L'adresse IP et le masque de sous-réseau sont inséparables.

# Notation CIDR

- Classless Inter-Domain Routing
- Elle définit un routage Internet sans classe pour répondre aux problèmes rencontrés avec les classes initiales.
- Avec la notation CIDR, on définit une plage d'adresse correspondant à un réseau en donnant la première adresse de la plage suivi par une barre oblique (ou slash, « / ») et d'un nombre correspondant au nombre de bits à 1 dans la notation binaire du masque de sous-réseau.
  - Ex: Le sous réseau 192.168.1.0 ayant pour masque 255.255.255.0 deviendra 192.168.1.0/24.
- Avec la notation CIDR vient la notion de VLSM (Variable Length Subnet Mask) qui permet de diviser un réseau en sous-réseau de tailles différentes. VLSM et CIDR sont très liés.

# Exemple

Prenons l'adresse IP 192.168.1.2 et le masque 255.255.255.0

La notation de l'adresse sera 192.168.1.2/255.255.255.0 ou 192.168.1.2/24

L'adresse en binaire est : 11000000.10101000.00000001.00000010

Le masque en binaire est : 11111111.11111111.11111111.00000000

➔ L'adresse IP 192.168.1.2/24 est la seconde adresse du réseau 192.168.1.0/24

# Notions sur les adresses IP

- Il y a $2^X$ adresses possibles dans un sous-réseau. X étant le nombre de 0 dans le masque.
- Parmi les adresses d'une plage il y en a deux spécifiques:
  - La **première** adresse de la plage qui est **l'adresse du réseau**
  - La **dernière** adresse de la plage qui est **l'adresse de broadcast**
  - Nous ne pourrons pas utiliser ces adresses pour adresser des machines ⇔Le nombre de machines adressables dans un réseau est de $2^x$-2, X étant le nombre de 0 dans le masque de sous réseau.
- L'adresse de broadcast ou adresse de diffusion permet d'adresser toutes les machines d'un réseau en même temps.

# La RFC 1918

- Elle définit des plages d'adresses, non routées sur Internet, réservées pour une utilisation privée.
- Ces plages d'adresses sont :
  - 10/8 ou 10.0.0.0/8 ou 10.0.0.0/255.0.0.0
  - 172.16/12 ou 172.16.0.0/12 ou 172.16.0.0/255.240.0.0
  - 192.168/16 ou 192.168.0.0/16 ou 192.168.0.0/255.255.0.0
- RFC (Request For Comment) : C'est un document qui propose et présente une technologie que l'on souhaite voir utilisée sur Internet.

# Exercice

- Soit l'adresse 192.168.0.15/255.255.255.240, déterminez s'il s'agit d'une adresse de réseau, de machine ou de broadcast.

# Exercice

- Soit l'adresse 192.168.0.15/255.255.255.240, déterminez s'il s'agit d'une adresse de réseau, de machine ou de broadcast.

  - 192.168.0.15 : 1100000 . 10101000 . 00000000 . 00001111
  - 255.255.255.240 : 11111111 . 11111111 . 11111111 . 11110000

- Dans ce réseau, les adresses varient entre :

  1100000 . 10101000 . 00000000 . 00000000 (192.168.0.0) et,

  1100000 . 10101000 . 00000000 . 00001111 (192.168.0.15)

- **Il s'agit de l'adresse de broadcast !**

# Découpage d'une plage d'adresses

- Le découpage d'une plage d'adresses nous permet d'organiser notre réseau.
- Prenons l'exemple suivant:

*Nous administrons le réseau d'une entreprise de 1220 employés composée de 1000 techniciens, 200 commerciaux et 20 directeurs.*
*Pour cela nous disposons de la plage d'adresse 10.0.0.0/16.*

*Définissez les sous-réseaux pour chacun des groupes en économisant au maximum les adresses IP.*

# Découpage d'une plage d'adresses

1) On vérifie que nous avons suffisamment d'adresses dans la plage donnée:

   10.0.0.0/16

   - Nombre d'adresses dans une plage : $2^X$, avec X représentant le nombre de 0 dans le masque de sous-réseau:

     $2^{(32-16)} = 2^{16} = 65536$ adresses possibles

   Nous disposons de 65536 adresses pour pourvoir 1220 machines.

# Découpage d'une plage d'adresses

2) On calcule les masques de sous-réseau pour les différents groupes:

- Techniciens:

On sait que le nombre d'adresses dans une plage est déterminé par $2^X$, avec X représentant le nombre de 0 dans le masque de sous-réseau. Nous devons donc déterminer X tel que $2^X$ > Nombre de techniciens.

$2^X > 1000$

$\Leftrightarrow \text{Log}_2(2^X) > \text{Log}_2(1000)$

$\Leftrightarrow X > \text{Log}_2(1000)$

$\Leftrightarrow X > \frac{\text{Log}(1000)}{\text{Log}(2)} \sim 10$

Nous aurons donc 10 zéros dans le masque et le masque de sous-réseau pour les techniciens sera /22 (pour 32 – 10)

En binaire:

11111111 . 11111111 . 11111100 . 00000000 ➔ 255.255.252.0

# Découpage d'une plage d'adresses

2) On calcule les masques de sous-réseau pour les différents groupes:

- Commerciaux:

Même principe, nous devons donc déterminer X tel que $2^X$ > Nombre de commerciaux.

$2^X > 200$ , ici X = 8

Le masque de sous-réseau pour les commerciaux sera /24 (pour 32 – 8)

En binaire:

11111111 . 11111111 . 11111111 . 00000000 ➔ 255.255.255.0

# Découpage d'une plage d'adresses

2) On calcule les masques de sous-réseau pour les différents groupes:

- Directeurs:

déterminons X tel que $2^X$ > Nombre de directeurs.

$2^X > 20$, ici X = 5

Le masque de sous-réseau pour les directeurs sera /27 (pour 32 – 5)

En binaire:

11111111 . 11111111 . 11111111 . 11100000 ➔ 255.255.255.224

# Découpage d'une plage d'adresses

3) On choisit les plages d'adresses :

- Nous commencerons par l'adresse la plus basse de la plage donnée (soit 10.0.0.0) et par le plus grand des trois sous-réseaux à adresser.
  Nous calculerons la première et la dernière adresse de chaque plage pour déterminer par quelle adresse commencera le sous-réseau suivant.
- Le plus gros des 3 sous-réseaux est celui des techniciens.
- Techniciens : 10.0.0.0/22

10.0.0.0 : 000001010 . 00000000 . 000000**00** . 00000000
255.255.252.0 : 11111111 . 11111111 . 11111**00** . 00000000
Dernière @ : 000001010 . 00000000 . 000000**11.11111111** ➔ 10.0.3.255

Le réseau des techniciens s'étend de 10.0.0.0 à 10.0.3.255.

Le réseau suivant commencera par l'adresse 10.0.4.0.

# Découpage d'une plage d'adresses

3) On choisit les plages d'adresses :

- Le sous-réseau des commerciaux est le plus gros des deux sous-réseaux restants:
- Commerciaux : 10.0.4.0/24

10.0.4.0 : 00001010 . 00000000 . 00000100 . 00000000
255.255.255.0 : 11111111 . 11111111 . 11111111 . 00000000
Dernière @ : 00001010 . 00000000 . 00000100.11111111 ➔ 10.0.4.255

Le sous-réseau des commerciaux s'étend de 10.0.4.0 à 10.0.4.255.

Le sous-réseau suivant commencera par l'adresse 10.0.5.0.

# Découpage d'une plage d'adresses

3) On choisit les plages d'adresses :

- Enfin le sous-réseau des directeurs:
- <u>Directeurs</u> : 10.0.5.0/27

10.0.5.0 : 00001010 . 00000000 . 00000101 . 00000000
255.255.255.224 : 11111111 . 11111111 . 11111111 . 11100000
Dernière @ : 00001010 . 00000000 . 00000101.00011111 ➔ 10.0.5.31

Le sous-réseau des directeurs s'étend de 10.0.5.0 à 10.0.5.31.

# La méthode magique

- Il s'agit d'une méthode qui permet de simplifier le calcul de la première et de la dernière adresse d'une plage.
- Pour cela on utilise le nombre magique que l'on calcule grâce à l'octet significatif.
  - Nombre magique = 256 – octet significatif.
  - Ex : 255.224.0.0.
    Octet significatif : 224
    Nombre magique = 256-224 = 32
- L'adresse du réseau sera un multiple du nombre magique et la dernière sera le multiple suivant -1.

# La méthode magique

- Ex: Soit l'adresse 192.168.0.1/255.224.0.0

Nombre magique = 256 – 224 = 32

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

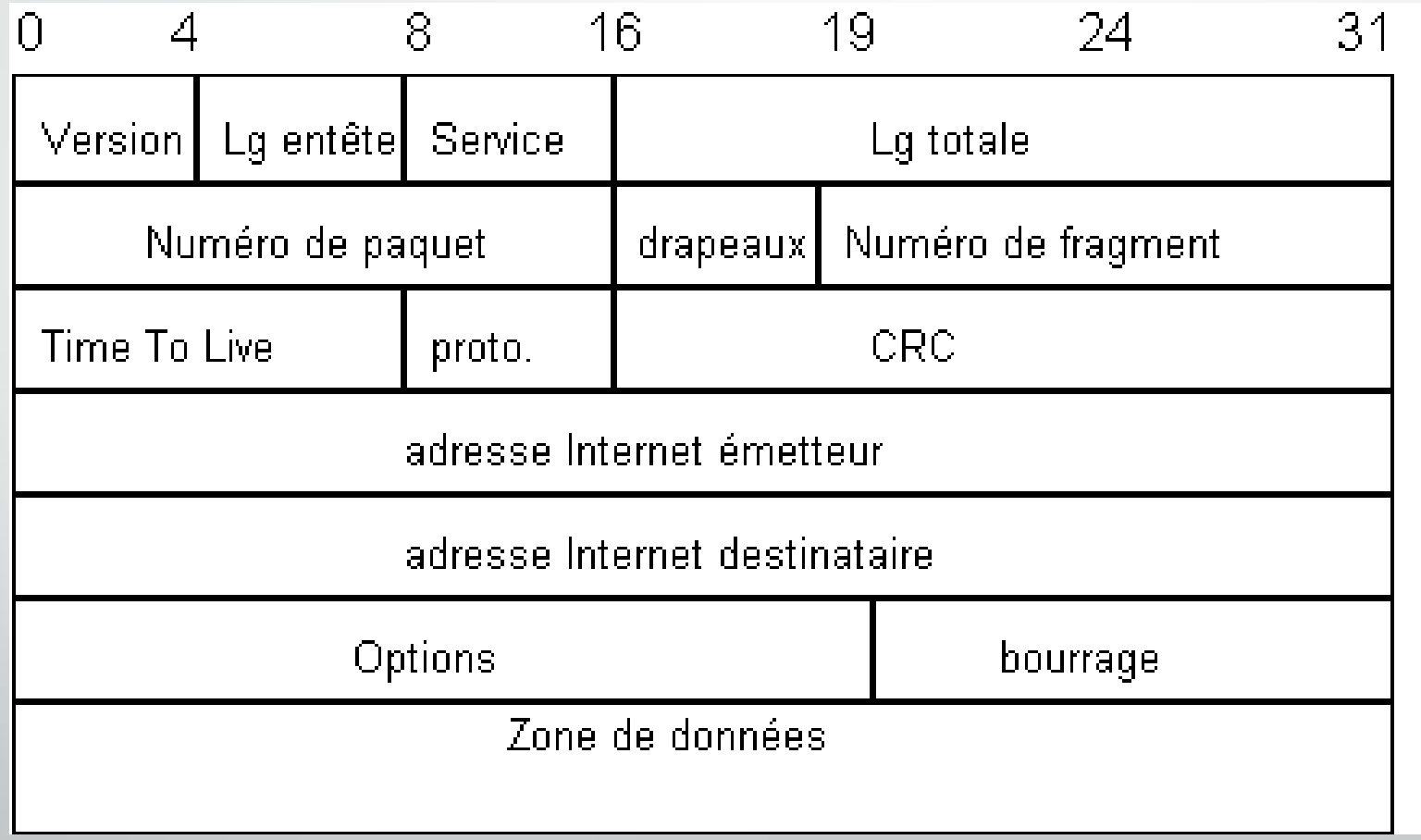
Les multiples de 32 les plus proches de l'octet significatif de l'adresse (ici 168) sont : 160 et 192.

L'adresse du réseau sera donc **192.160.0.0** et la dernière adresse de la plage sera **192.191.255.255**

# Le protocole de la couche 3 : IP

- Le protocole de la couche 3 est IP (Internet Protocole)
- Comme Ethernet pour la couche 2, IP va définir les données de couche 3 qui transiteront sur le réseau et l'ordre dans lequel elles apparaitront dans le message.
- Le message de la couche 3 s'appelle **datagramme** ou **paquet**.

# Format du datagramme IP



# Le matériel de la couche 3 : Le routeur

- Le routeur est une machine qui dispose de plusieurs interfaces.
- Chacune des interfaces est reliée à un réseau permettant ainsi de relier plusieurs réseaux entre eux.
- Il aiguille les paquets entre les différents réseaux.
- Pour router les paquets, un routeur va consulter sa table de routage.

# Le matériel de la couche 3 : Le routeur

# La table de routage

- Elle contient la liste des routeurs auxquels envoyer les datagrammes en fonction de la destination.
- Chaque entrée de la table de routage fait la correspondance entre un réseau et la passerelle à emprunter pour joindre ce réseau.
- Il existe une entrée spécifique dans la table de routage qui permet d'envoyer les paquets dont le réseau n'est pas connu dans la table de routage vers une passerelle : C'est la route par défaut.
- Toutes les machines d'un réseau possèdent une table de routage.
- La table de routage peut être remplie de manière statique (à la main) ou dynamique (à l'aide de protocole de routage dynamique).

# Le routage statique

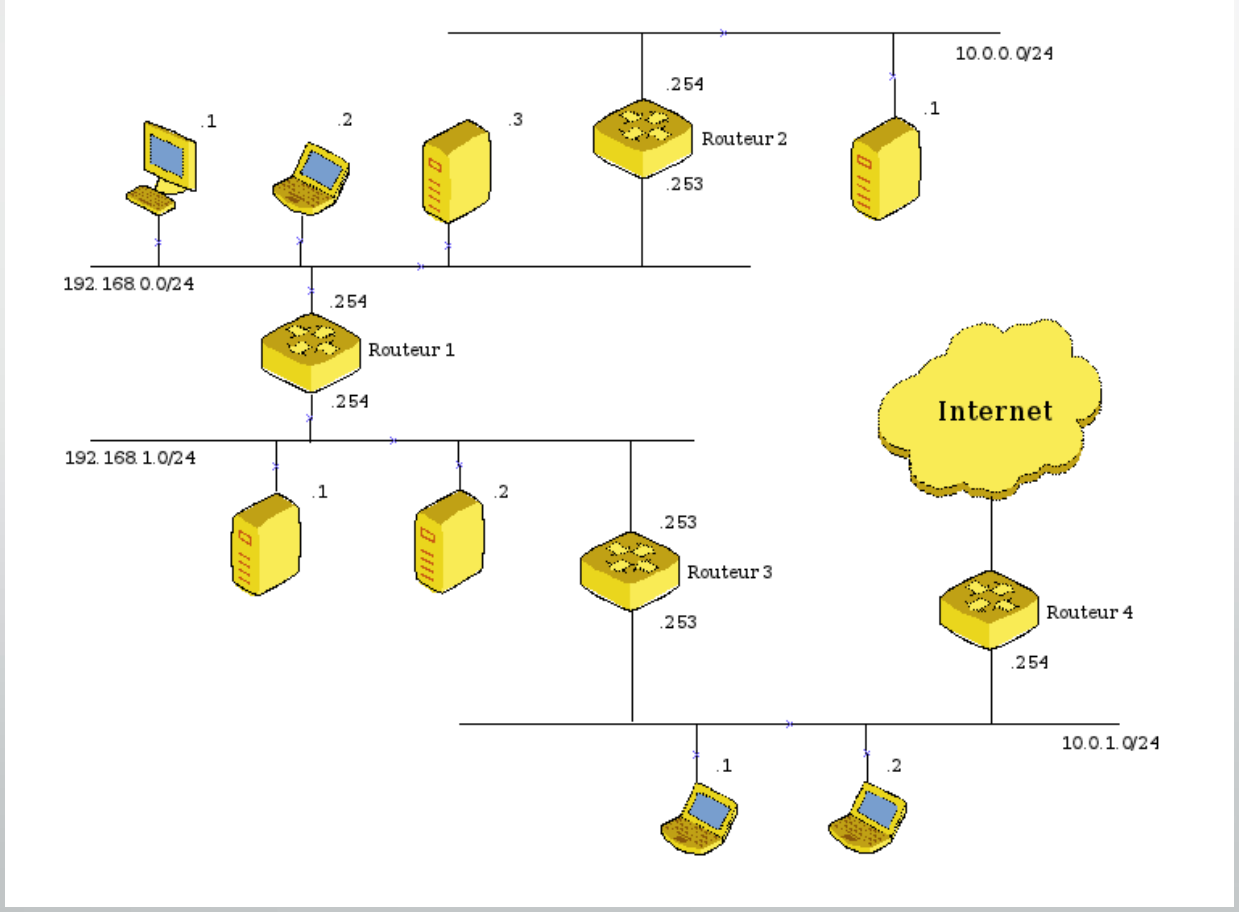
- Il s'agit de remplir manuellement la table de routage.
- Pour chaque réseau que l'on veut joindre on ajoute une entrée dans la table avec ce réseau et la passerelle à prendre pour le joindre.

  /!\ la passerelle doit appartenir à un des réseaux de la machine /!\

- Pour remplir la table de routage on suit les étapes suivantes:
  1. Indiquer les réseaux de ma machine
  2. Indiquer la route par défaut
  3. Indiquer tous les réseaux que je ne peux pas joindre à la suite des deux étapes précédentes.

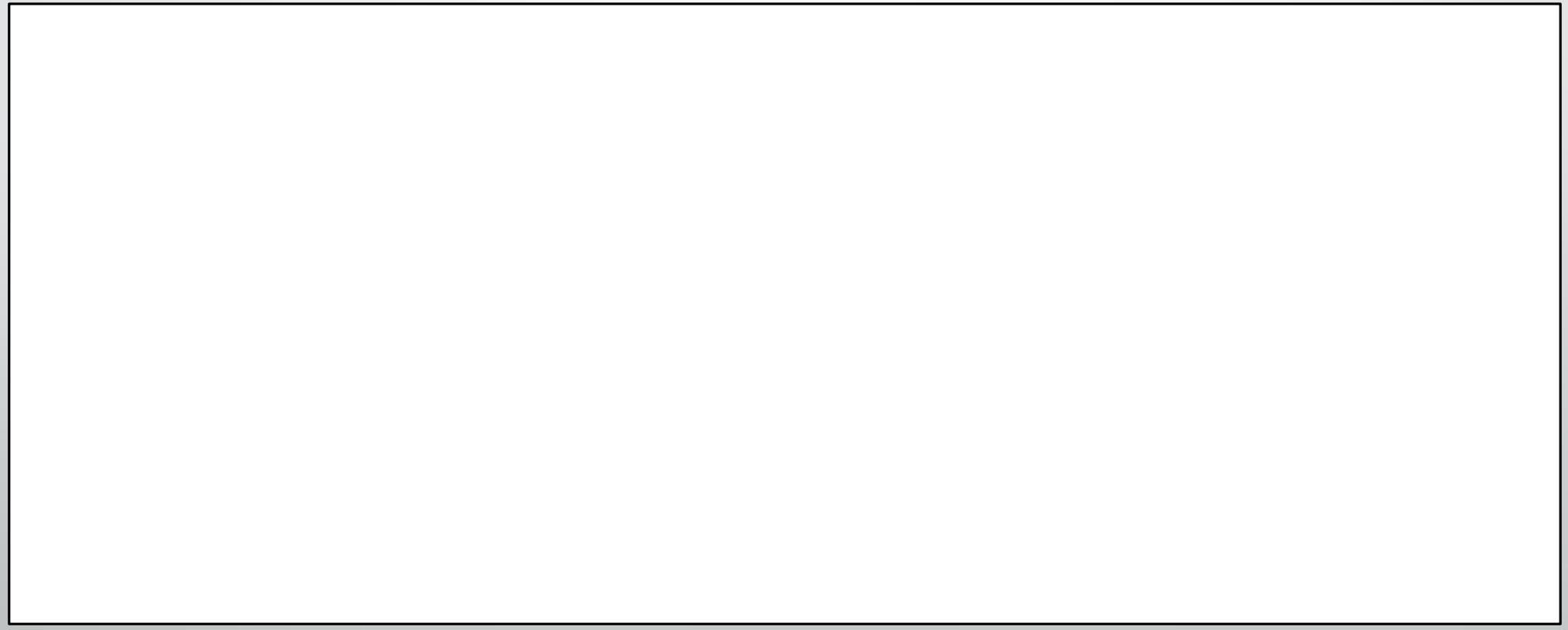
# Le routage statique

- Ex : Soit le réseau suivant:



# Le routage statique

- Donnez la table de routage du routeur 1.



# Le routage statique

- Donnez la table de routage du routeur 1.

| Table de routage de routeur 1 | |
|---|---|
| 192.168.0.0/24 | 192.168.0.254 |
| 192.168.1.0/24 | 192.168.1.254 |
| 0.0.0.0/0 | 192.168.1.253 |
| 10.0.0.0/24 | 192.168.0.253 |

# Le routage dynamique

- Il permet de remplir automatiquement la table de routage à l'aide d'algorithmes et de protocoles de routage dynamique.
- Parmi les protocoles de routage dynamique, on rencontre principalement:
  - RIP / RIPv2
  - OSPF
  - BGP

# RIP / RIPv2

- Routing Information Protocol
- Il s'appuie sur l'algorithme de détermination des routes Bellman-Ford.
- C'est un protocole de routage à vecteurs de distance:
  - Chaque routeur communique aux routeurs voisins la métrique, c'est-à-dire la distance (le nombre de sauts) qui les sépare d'un réseau IP déterminé.
  - Les routeurs ne possèdent pas la vision globale du réseau, la diffusion des routes se faisant de proche en proche.
- RIPv2 reprend la base de RIP en ajoutant la prise en compte des contraintes des réseaux actuels (découpages des plages en sous-réseaux, authentification par mot de passe …).

# OSPF

- Open Shortest Path First
- Il s'appuie sur l'algorithme de Dijkstra.
- C'est un protocole de routage de type état de liens :
  - Les routeurs transmettent la totalité des informations de routage à tous les routeurs participants (on dit qu'ils inondent les informations) et établissent des tables de voisins directs.
  - Les routeurs vont pouvoir établir une cartographie complète du réseau et calculer les chemins les plus courts (les moins couteux) pour joindre une destination.
- Deux autres versions d'OSPF ont été écrites pour répondre aux spécificités d'IP (OSPFv2) et d'IPv6 (OSPFv3).

# BGP

- Border Gateway Protocol
- C'est un protocole de routage à état de chemin (une variante des protocoles à vecteurs de distance):
  - Les routeurs s'échangent des informations d'accessibilités entre AS (Système autonome => un réseau de réseaux).
  - Les routeurs choisissent les routes en fonction de plusieurs critères tels que le poids du chemin, la préférence locale, l'origine, le chemin le plus court, les métriques etc etc…
- C'est le protocole de routage dynamique utilisé sur Internet.
- A la différence de RIP ou OSPF, c'est un protocole de routage externe, c'est-à-dire qu'il échange des informations avec d'autres systèmes autonomes (alors que pour RIP ou OSPF, l'échange d'informations se fait à l'intérieur d'un système autonome).
- Nous sommes à la version 4 de BGP et la prise en compte d'IPv6 se fait via l'utilisation d'extensions de BGP.

# Le protocole ARP

- C'est un protocole qui permet de connaitre l'adresse MAC en fonction de l'adresse IP et inversement.
- Pour obtenir l'adresse MAC, ARP envoie une requête en broadcast.
  La machine dont c'est l'adresse IP répondra avec son adresse MAC.

  ➔ C'est la requête ARP (on parle également de gratuitous ARP).
- Afin de ne pas surcharger le réseau avec des requêtes ARP, les machines du réseau enregistrent les informations dans la table ARP.
- Les informations de la table ARP ont une durée de vie limitée (TTL).
- Comme il manipule des informations de la couche 2 et de la couche 3 on dit qu'ARP est un protocole à cheval.

# Le protocole ARP

- Déroulement d'une requête ARP:
  - La machine A consulte sa table ARP.
  - Si elle possède l'information, elle transmet le message. FIN.
  - Sinon la machine A envoie un broadcast ARP sur le réseau.
  - La machine B reconnait son adresse IP et répond avec son adresse MAC.
  - La machine A ajoute l'information dans sa table ARP.
  - Elle transmet le message. FIN.

# Le protocole ICMP

- Le protocole ICMP est un protocole de couche 3.
  Il n'est pas concurrent à IP.
- Son rôle est de contrôler les erreurs de transmission et d'aider au débogage réseau.
- Il y a deux informations intéressantes dans l'en-tête ICMP:
  - Le type et,
  - Le code
- Le type précise à quoi sert le message d'erreur.
- Le code précise le rôle du message.

# Le protocole ICMP

- ICMP indique automatiquement les erreurs quand elles surviennent sur le réseau en retournant un code d'erreur spécifique.
- Pour voir les messages d'erreurs circuler sur le réseau, il faut écouter le trafic à l'aide d'un sniffer.
- Quelques messages d'erreurs pratiques
  - Type 3 : Host unreachable
  - Type 5 : ICMP redirect
  - Type 11 : TTL exceeded

# Le protocole ICMP

- ICMP fournit des outils de débogage pour le réseau.
- Parmi ces outils il y a le ping et le traceroute (ou tracert sous windows).
- Le ping est la combinaison des messages ICMP echo request (Type 8) et echo reply (Type 0).
- Pour le traceroute, on joue avec le message d'erreur « TTL exceeded ».
  On envoie un paquet sur le réseau avec un TTL de 1. Une fois que le message arrive sur le premier routeur, il décrémente le TTL et comme ce TTL arrive à 0, le routeur envoie un message d'erreur ICMP « TTL exceeded » et on connait ainsi l'adresse IP du premier routeur.
  On recommence cette opération jusqu'à la destination en incrémentant le TTL initial de 1 à chaque fois.

# IPv6

- L'explosion d'Internet a permis de mettre en évidence les lacunes de la technologie d'Internet et plus particulièrement d'IPv4.
- Parmi ces lacunes il y a notamment le nombre d'adresses limité pour IPv4, l'explosion des tables de routage et le manque de sécurité perturbant la qualité de service.
- IPv6 regroupe un ensemble de protocole ayant des rôles respectifs permettant de pallier aux lacunes d'IPv4.

# IPv6

- Les apports d'IPv6:
  - Les adresses sont codées en hexadécimale sur 128 bits regroupés en 8 groupes de 16 bits séparés par « **:** ».

    Ex: 2001:0db8:0000:85a3:0000:0000:ac1f:8001
  - L'allègement du traitement des paquets au niveau des routeurs.
  - L'amélioration de la qualité de service.
  - L'autoconfiguration des équipements IPv6 à l'aide de nombreux protocoles (DHCPv6, amélioration d'ICMP, découverte des voisins …).
  - La transparence de la mobilité (quel que soit le type de réseau, l'adresse IPv6 ne change pas).
  - Le renforcement de la sécurité à l'aide d'IPsec (gestion de l'authentification et chiffrement des données).

# Configuration de la couche 3

- Configuration de l'adresse IP sous linux:
  - Configuration à l'aide de la commande ifconfig

  Exemple:

  ifconfig etho 10.0.0.11 netmask 255.255.255.0

  ou

  ifconfig etho 10.0.0.11/24
  - Configuration permanente dans le fichier /etc/network/interfaces (sous Debian) ou /etc/sysconfig/network-scripts/ifcfg-ifXXX.cfg (sous CentOS)

# Configuration de la couche 3

- Exemples configuration permanente:

pi@jumper: ~

```
pi@jumper ~ $ cat /etc/network/interfaces
# LOOPBACK
auto lo
iface lo inet loopback

#ETH0
auto eth0
iface eth0 inet static
        address 10.0.0.11
        netmask 255.255.255.0
        network 10.0.0.0
        broadcast 10.0.0.255
        gateway 10.0.0.1

#WLAN0
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

#DEFAULT
iface default inet dhcp
pi@jumper ~ $
```

root@destiny:~

```
[root@destiny ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=eth0
BOOTPROTO=static
IPADDR=10.0.0.12
NETMASK=255.255.255.0
NETWORK=10.0.0.0
BROADCAST=10.0.0.255
HWADDR=EC:A8:6B:F6:37:20
TYPE=Ethernet
ONBOOT=yes
[root@destiny ~]#
```

# Configuration de la couche 3

- Configuration du routage sous linux:
  - Configuration à l'aide de la commande route

  Exemple:

  route add -net 172.16.0.0 netmask 255.255.255.0 gw 10.0.0.2 dev etho

  - Configuration permanente dans le fichier /etc/network/interfaces (sous Debian) ou /etc/sysconfig/network-scripts/route-ethX (sous CentOS)

# Configuration de la couche 3

- Exemples configuration permanente:

pi@jumper: ~

```
root@jumper:~# cat /etc/network/interfaces
# LOOPBACK
auto lo
iface lo inet loopback

#ETH0
auto eth0
iface eth0 inet static
        address 10.0.0.11
        netmask 255.255.255.0
        network 10.0.0.0
        broadcast 10.0.0.255
        gateway 10.0.0.1
up route add -net 172.16.0.0 netmask 255.255.255.0 gw 10.0.0.2
down route del -net 172.16.0.0 netmask 255.255.255.0 gw 10.0.0.2

#WLAN0
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
```

root@destiny:~

```
[root@destiny ~]# cat /etc/sysconfig/network-scripts/route-eth0
172.16.0.0/24 via 10.0.0.2
[root@destiny ~]#
```

# Couche 4 : La couche Transport

# Couche 4 : Transport

- La couche transport est la quatrième couche du modèle OSI.
- Elle fait le lien entre la couche applicative et les couches réseau. Son rôle est de permettre la communication entre applications.
- Pour identifier les applications la couche 4 va utiliser une adresse (un identifiant) : **Le port**.

# Le port

- C'est l'identifiant des applications sur la couche 4 (c'est l'adresse de l'application).
- Derrière chaque port ouvert sur une machine, il y a une application qui tourne.
- Les ports sont codés en décimal sur deux octets (65535 ports possibles).
- Historiquement tous les ports inférieurs à 1024 sont réservés, mais beaucoup d'applications ont un port appartenant au range supérieur à 1024 (ex: MySQL tourne sur le port 3306).
- Les applications clientes utilisent également un numéro de port, attribué aléatoirement dans les ports supérieurs à 1024.

# Liste de ports connus

- **20/21,** pour l'échange de fichiers via FTP
- **22,** pour l'accès à un shell sécurisé Secure SHell, également utilisé pour l'échange de fichiers sécurisés SFTP
- **23,** pour le port telnet
- **25,** pour l'envoi d'un courrier électronique via un serveur dédié SMTP
- **53,** pour la résolution de noms de domaine en adresses IP : DNS
- **67/68,** pour DHCP et bootpc
- **80,** pour la consultation d'un serveur HTTP par le biais d'un navigateur web
- **110,** pour la récupération de son courrier électronique via POP
- **123** pour la synchronisation de l'horloge : Network Time Protocol (NTP)
- **143,** pour la récupération de son courrier électronique via IMAP
- **389,** pour la connexion à un LDAP
- **443,** pour les connexions HTTP utilisant une surcouche de sécurité de type SSL : HTTPS
- **465,** pour l'envoi d'un courrier électronique via un serveur dédié utilisant une surcouche de sécurité de type SSL : SMTPS
- **500,** port utilisé pour le canal d'échange de clés IPsec
- **636,** pour l'utilisation d'une connexion à un LDAP sécurisé par une couche SSL/TLS
- **1521,** serveur de base de données Oracle Database
- **1723,** pour l'utilisation du protocole de VPN PPTP
- **3306,** serveur de base de données MySQL
- **3389,** pour la prise de contrôle à distance RDP
- **5432,** serveur de base de données PostgreSQL
- **6667,** pour la connexion aux serveurs IRC

# Les protocoles de la couche 4

- Deux protocoles ont été créés pour répondre aux principaux besoins de communication:
  - **Un protocole fiable, sans nécessité de rapidité** => TCP qui est un protocole connecté avec lequel chaque paquet sera acquitté.
  - **Un protocole rapide, sans nécessité de fiabilité** => UDP qui est un protocole non connecté avec lequel chaque paquet sera envoyé dès que possible sans contrôle sur la réception des données.

# Les protocoles de la couche 4 : UDP

- UDP est un protocole simple dont le but est d'assurer la rapidité.
- Le format des datagrammes UDP sera simple afin d'assurer cette rapidité:

| Port source | Port destination | Longueur | Checksum | Données |
|---|---|---|---|---|

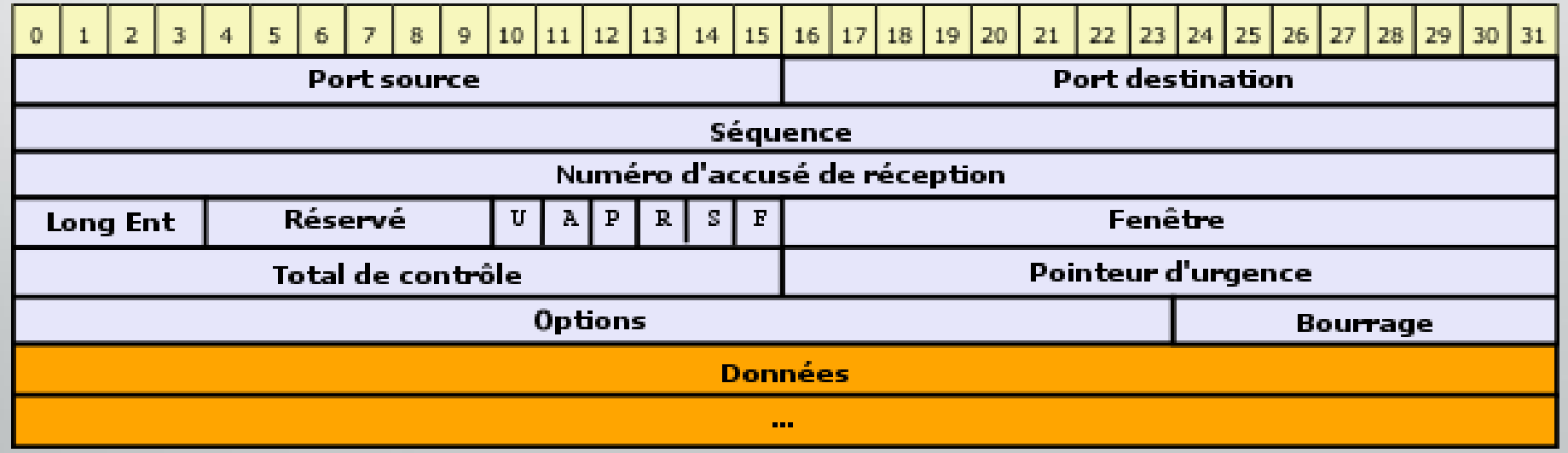
- Nous aurons 4 informations dans l'en-tête UDP, chacune faisant 2 octets (soit 8 au total).
- La taille maximale du datagramme UDP est de 65536 octets mais dans la pratique, les datagrammes UDP dépassent rarement 512 octets.
- Le checksum permet d'assurer l'intégrité des données.

# Les protocoles de la couche 4 : UDP

- Les principales applications qui utilisent UDP sont:
  - Le streaming
  - VOIP / TOIP
  - DNS
  - SNMP

# Les protocoles de la couche 4 : TCP

- Le but de TCP est d'assurer l'échange des informations. Chaque octet transmis devra être acquitté.
- Le message pour le protocole TCP s'appelle Segment.
- L'en-tête TCP fait 20 octets.



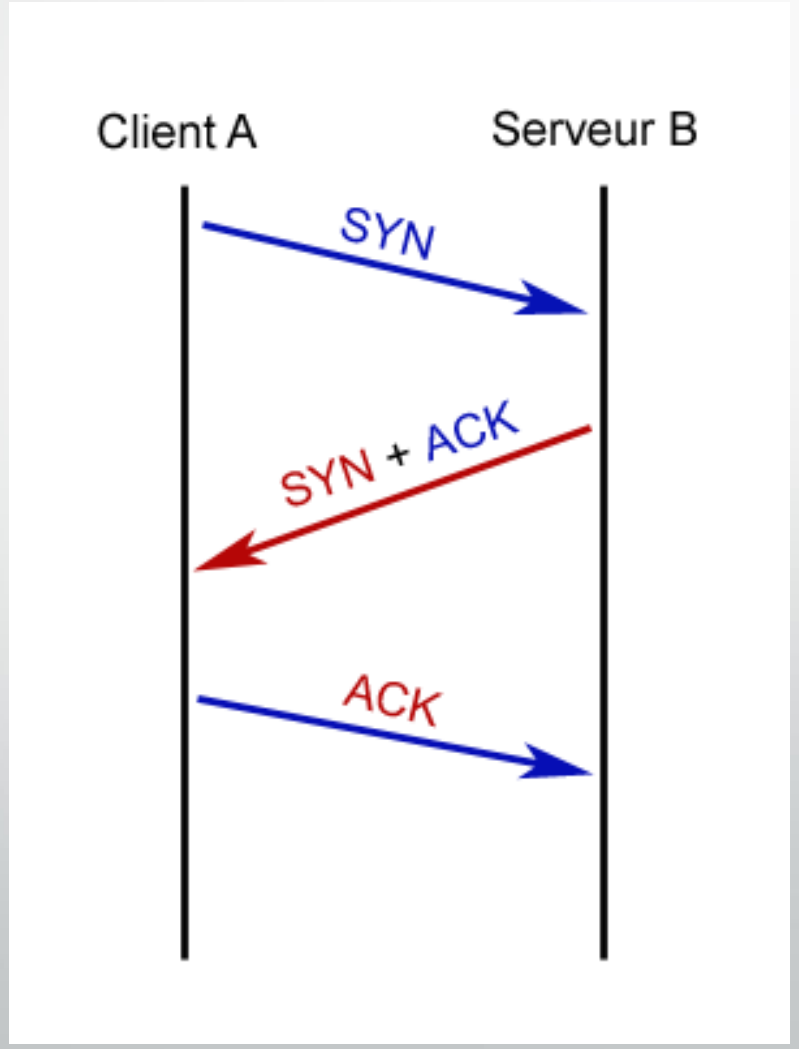
# Les protocoles de la couche 4 : TCP

- Il y aura trois phases dans un échange TCP:
  - L'initiation de la connexion
  - L'échange des informations
  - La fin de connexion
- Afin d'identifier les différents paquets TCP durant chacune des phases, nous aurons recourt aux drapeaux TCP(flags) qui sont des bits dans l'en-tête que l'on positionnera à 1 ou 0 en fonction du type de message TCP.
- Les différents flags TCP sont : SYN, ACK, RST, PSH, URG, FIN.
- RST permet d'arrêter une connexion en cas d'anomalie.
- PSH et URG signalent les paquets à traiter en priorité.

# Les protocoles de la couche 4 : TCP

- Initiation de la connexion:
  - Les trois premiers paquets d'un échange TCP seront dédiés à l'initiation de la connexion.
  - Le premier paquet du client sera une demande de synchronisation, dans lequel le flag SYN sera positionné à 1.
  - La réponse du serveur sera un paquet avec les flags SYN et ACK (on parle de SYN/ACK), car avec TCP on initie une connexion dans les deux sens.
  - Le troisième sera l'acquittement par le client de la synchronisation envoyée par le serveur.
- L'initiation de connexion TCP s'appelle le **Three Way Handshake**.

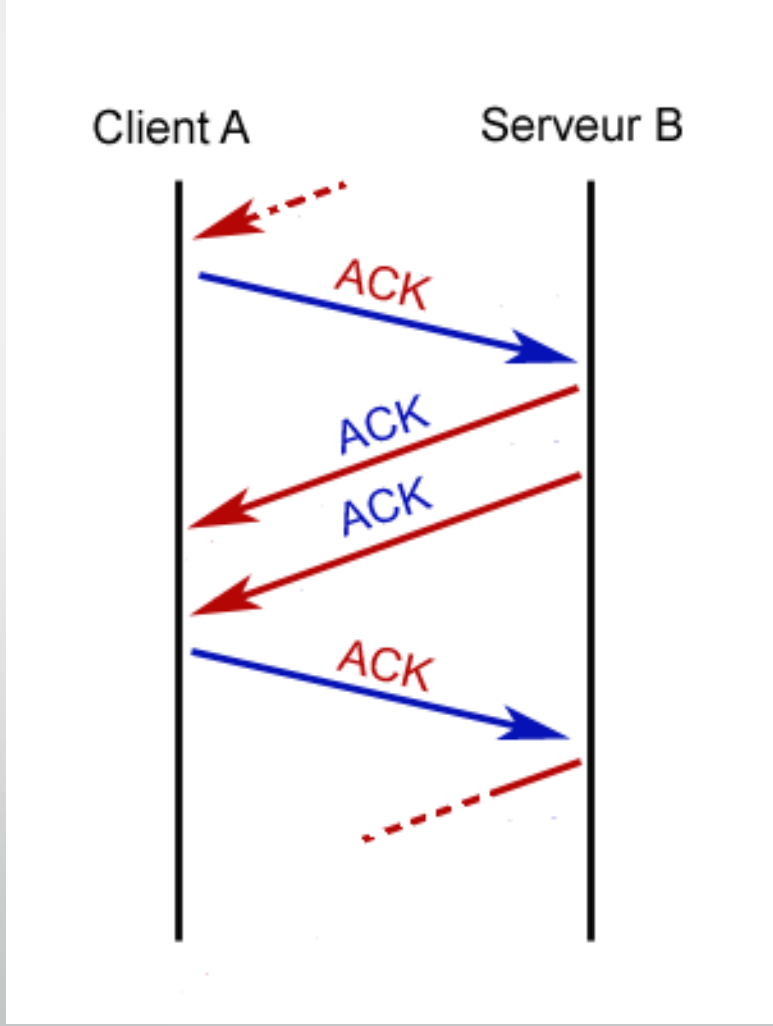
# Les protocoles de la couche 4 : TCP



# Les protocoles de la couche 4 : TCP

- Echange des informations:
  - Lorsque que la communication a été établie, les machines vont échanger autant qu'elles le souhaitent.
  - Tous les segments devront être acquittés (à l'aide du flag ACK).

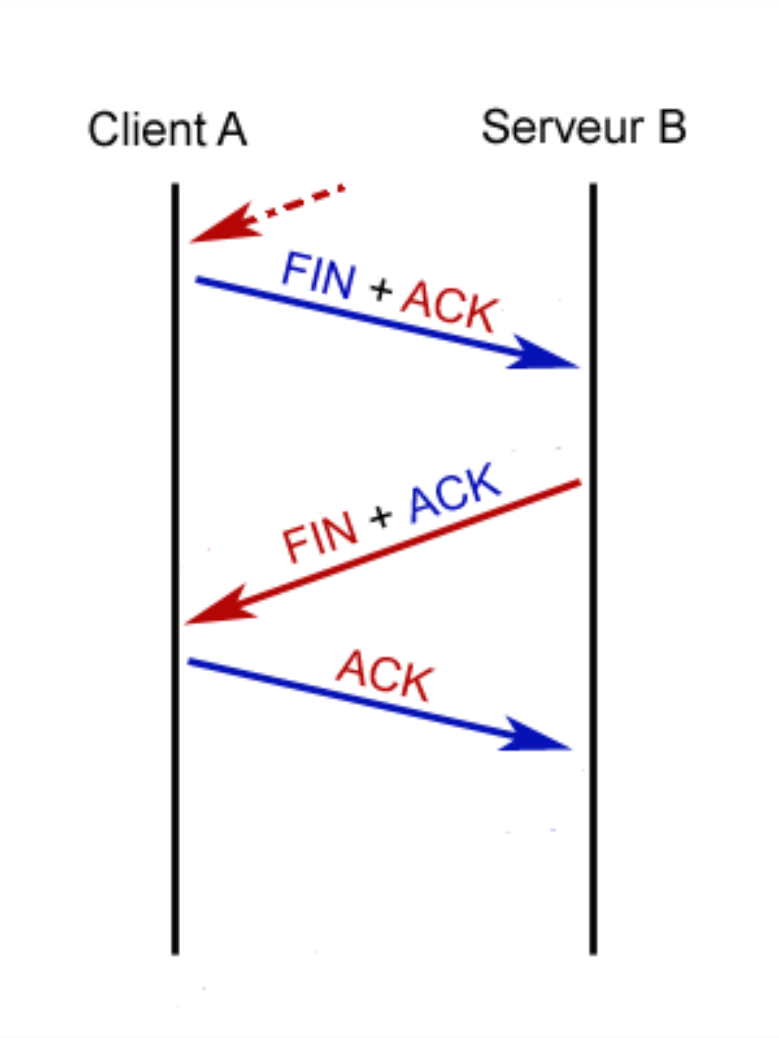
# Les protocoles de la couche 4 : TCP



# Les protocoles de la couche 4 : TCP

- Fin de connexion:
  - Lorsque les échanges sont terminés, les deux connexions établies devront être terminées proprement.
  - Le client enverra un paquet TCP FIN que le serveur acquittera puis ce sera au tour du serveur d'envoyer un paquet FIN que le client acquittera.

# Les protocoles de la couche 4 : TCP



# Les protocoles de la couche 4 : TCP

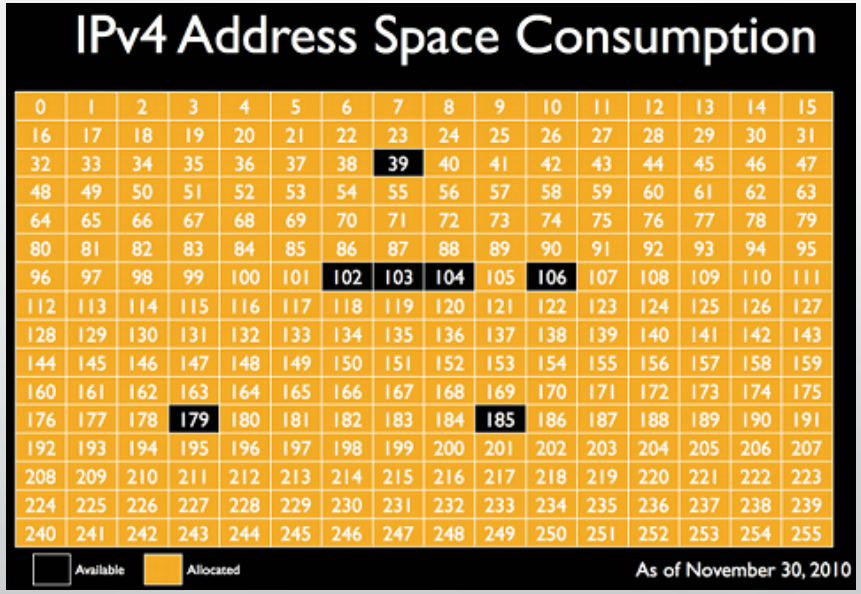
root@destiny:~

```
[root@destiny ~]# tcpdump -i eth0 host 10.0.0.11 and host 10.0.0.12
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
20:11:15.048850 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [S], seq 1903431199, win 14600, options [mss 1460,sackOK,TS val 60035831 ecr 0,nop,wscale 5], leng
th 0
20:11:15.048895 IP 10.0.0.12.http > 10.0.0.11.38162: Flags [S.], seq 2429524697, ack 1903431200, win 14480, options [mss 1460,sackOK,TS val 601527167 ecr 600
35831,nop,wscale 7], length 0
20:11:15.049793 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [.], ack 1, win 457, options [nop,nop,TS val 60035831 ecr 601527167], length 0
get
20:11:18.316260 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [P.], seq 1:6, ack 1, win 457, options [nop,nop,TS val 60036158 ecr 601527167], length 5
20:11:18.316290 IP 10.0.0.12.http > 10.0.0.11.38162: Flags [.], ack 6, win 114, options [nop,nop,TS val 601530435 ecr 60036158], length 0

20:11:20.686166 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [P.], seq 6:8, ack 1, win 457, options [nop,nop,TS val 60036395 ecr 601530435], length 2
20:11:20.686191 IP 10.0.0.12.http > 10.0.0.11.38162: Flags [.], ack 8, win 114, options [nop,nop,TS val 601532805 ecr 60036395], length 0

20:11:21.596136 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [P.], seq 8:10, ack 1, win 457, options [nop,nop,TS val 60036486 ecr 601532805], length 2
20:11:21.596161 IP 10.0.0.12.http > 10.0.0.11.38162: Flags [.], ack 10, win 114, options [nop,nop,TS val 601533715 ecr 60036486], length 0

20:11:22.056196 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [P.], seq 10:12, ack 1, win 457, options [nop,nop,TS val 60036532 ecr 601533715], length 2
20:11:22.056220 IP 10.0.0.12.http > 10.0.0.11.38162: Flags [.], ack 12, win 114, options [nop,nop,TS val 601534175 ecr 60036532], length 0

20:11:22.636791 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [P.], seq 12:14, ack 1, win 457, options [nop,nop,TS val 60036590 ecr 601534175], length 2
20:11:22.636816 IP 10.0.0.12.http > 10.0.0.11.38162: Flags [.], ack 14, win 114, options [nop,nop,TS val 601534755 ecr 60036590], length 0
20:11:27.937473 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [F.], seq 14, ack 1, win 457, options [nop,nop,TS val 60037120 ecr 601534755], length 0
20:11:27.937553 IP 10.0.0.12.http > 10.0.0.11.38162: Flags [F.], seq 1, ack 15, win 114, options [nop,nop,TS val 601540056 ecr 60037120], length 0
20:11:27.938450 IP 10.0.0.11.38162 > 10.0.0.12.http: Flags [.], ack 2, win 457, options [nop,nop,TS val 60037120 ecr 601540056], length 0
^C
16 packets captured
16 packets received by filter
0 packets dropped by kernel
[1]+  Done                    nc -l 80
[root@destiny ~]#
```

# La NAT et le port forwarding

# La NAT

- NAT pour Network Address Translation
- La NAT répond à deux problèmes majeurs:
  - La pénurie des adresses IP




  - L'impossibilité aux IP privées d'aller sur Internet.

# La NAT

- Il existe 2 types de NAT:
  - La NAT statique : 1 IP privée = 1 IP publique.
  - La NAT dynamique : 1 IP publique = n IP privées.
- Nous étudierons la NAT dynamique qui répond au problème de pénurie d'adresses
  - Pour identifier les différentes requêtes vers une même destination avec la NAT dynamique, on utilise le port source.
  - Pour retenir les informations de connexion, on utilise la table de NAT.

| **Table NAT** | |
|---|---|
| @ IP source (privée), @ IP destination, Port source, Port destination | @ IP source (publique), @ IP destination, Port source, Port destination |

  - Afin d'éviter les doublons, le routeur (ou la box) va attribuer les ports sources sur le réseau publique.

# Le port forwarding

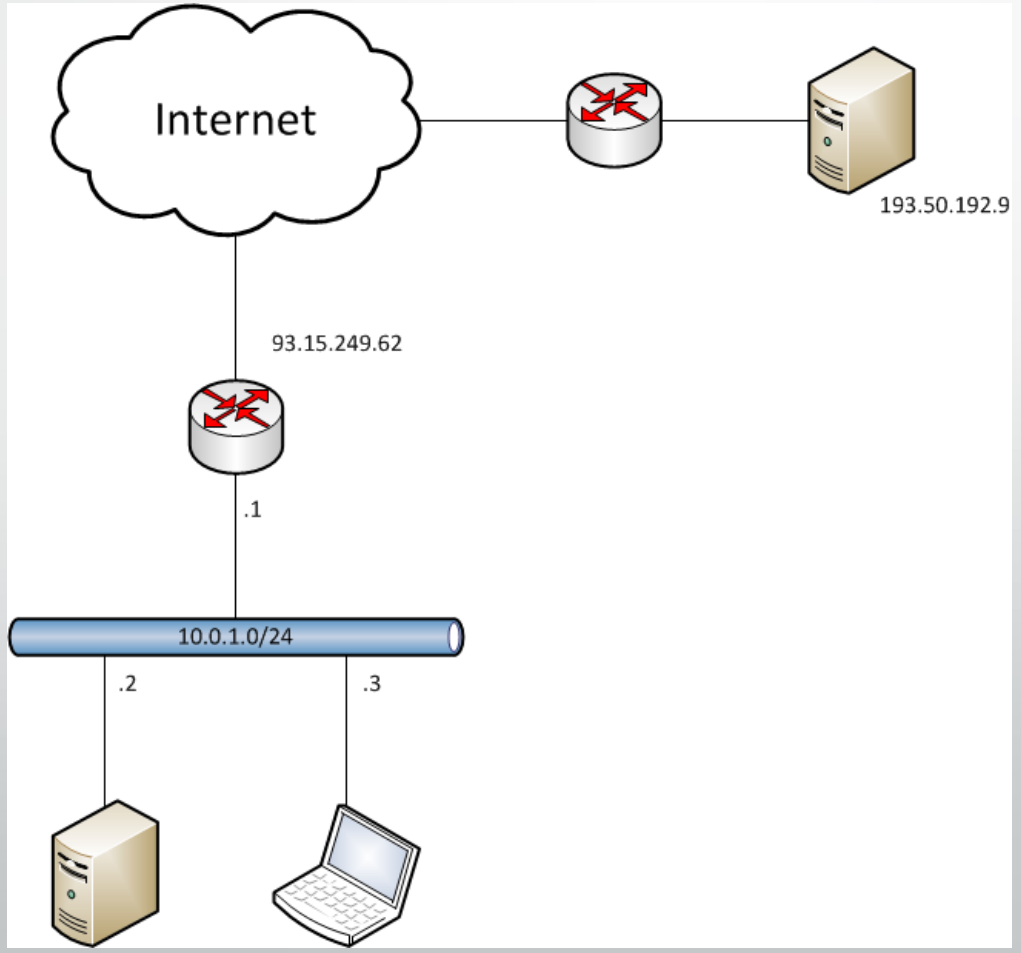
- Le port forwarding permet d'être joignable depuis Internet en cas de NAT dynamique.
- Il consiste à donner au routeur (ou à la box) les instructions de redirection en fonction d'un port (c'est-à-dire qu'on redirige un port du routeur vers un port donné sur une machine locale).

# La NAT et le port forwarding

- Avantages:
  - Economie d'adresses publiques.
  - Possibilité de joindre Internet depuis une machine locale (IP privée).
  - Gain en sécurité (n'est accessible que ce qui est nécessaire).
- Inconvénient:
  - Utilisation de port non standard pour certaines applications.

# La NAT et le port forwarding

- Exemple 1 : Requête vers www.univ-valenciennes.fr



# La NAT et le port forwarding

- Trame sur le réseau local:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| @MAC routeur | @MAC 10.0.0.3 | Protocole couche 3 | … | 10.0.0.3 | 193.50.192.9 | Port source 10725 | Port destination 80 | CRC |

- Trame sur Internet:

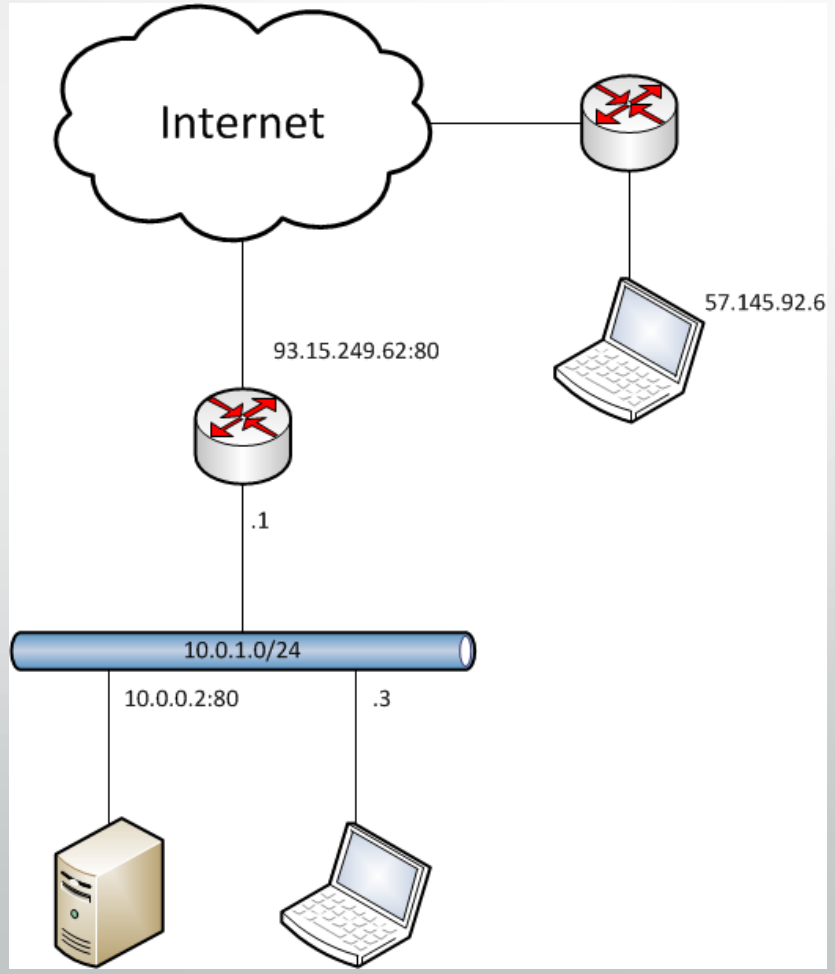
| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| @MAC Routeur opérateur | @MAC routeur | Protocole couche 3 | … | 93.15.249.62 | 193.50.192.9 | Port source 9345 | Port destination 80 | CRC |

- Table NAT:

| **Table NAT** | |
|---|---|
| 10.0.0.3, 193.50.192.9, 10725, 80 | 93.15.249.62, 193.50.192.9, 9325, 80 |

# La NAT et le port forwarding

- Exemple 2 : Requête vers mon site local sur 10.0.0.2



# La NAT et le port forwarding

- Trame sur Internet:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| @MAC routeur | @MAC Routeur opérateur | Protocole couche 3 | … | 57.145.92.6 | 93.15.249.62 | Port source 10725 | Port destination 80 | CRC |

- Table de port forwarding:

| **Table de port forwarding** | | | |
|---|---|---|---|
| @ IP externe | Port externe | @ IP interne | Port interne |
| 93.15.249.62 | 80 | 10.0.0.2 | 80 |

- Trame sur le réseau local:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| @MAC 10.0.0.2 | @MAC routeur | Protocole couche 3 | … | 57.145.92.6 | 10.0.0.2 | Port source 10725 | Port destination 80 | CRC |

# Les ACL

# Les ACL

- Access Control List
- Les ACL permettent de filtrer les accès entre les différents réseaux ou de filtrer les accès au routeur lui même.
- Les paramètres contrôlés sont:
  - Adresse source
  - Adresse destination
  - Protocole utilisé
  - Numéro de port
- Il existe deux types d'ACL: les ACL standards et les ACL étendues.

# Les ACL

- L'ACL standard filtre uniquement sur les adresses IP sources.
  - Ex: access-list numéro-de-la-liste {permit|deny} {host|source source-wildcard|any}
- L'ACL étendue filtre sur les adresses source et destination, sur le protocole et le numéro de port.
  - Ex: access-list numéro de la liste {deny|permit} protocole source masque-source [operateur [port]] destination masque-destination [operateur [port]][established][log]

# Les ACL

- Les ACL peuvent être appliquées sur le trafic entrant ou sortant.
- Il y a deux actions possibles: soit le trafic est interdit, soit le trafic est autorisé.
- Par défaut, tout le trafic est interdit.
- Les ACL sont prises en compte de façon séquentielle. Il faut donc placer les instructions les plus précises en premier et l'instruction la plus générique en dernier.
- On placera les ACL étendues au plus près de la source du paquet que possible pour le détruire le plus vite possible et les ACL standard au plus près de la destination sinon, on risque de détruire un paquet trop tôt.

# Les ACL

- Exemple: Création d'une entrée d'une access-list

On autorise la machine 192.168.2.12 à se connecter via ssh à toutes les machines du réseau 192.168.3.0/24, on autorise les réponses DNS en provenance de la machine 192.168.2.30, on autorise les paquets entrants pour les connexions tcp établies, enfin on supprime le reste du trafic qui va apparaitre dans les logs.

```
R2(config)#ip access-list extended reseau-licence-pro-RT
R2(config-ext-nacl)#permit tcp host 192.168.2.12 gt 1023 192.168.3.0 0.0.0.255 eq 22
R2(config-ext-nacl)#permit udp host 192.168.2.30 eq 53 192.168.3.0 0.0.0.255 gt 1023
R2(config-ext-nacl)#permit tcp any any established
R2(config-ext-nacl)#deny ip any any log
```

Enfin on appliquera cette ACL à une interface sur le routeur pour qu'elle puisse prendre effet.

# Les masques inverses

- Les masques inverses ou wildcard mask sont utilisés pour certains protocoles ou certaines fonctionnalités (les ACLs, OSPF, …).
- Ils servent à identifier des sous-réseaux ou des plages d'adresses comme peuvent le faire les masques de sous-réseaux.
- Leur particularité réside dans la façon dont ils sont appliqués:
  - Un 0 (zéro) dans le masque signifie que l'on va vérifier la correspondance du bit dans l'adresse.
  - Un 1 (un) dans le masque signifie que l'on va ignorer la valeur.
- Les masques inverses sont généralement utilisés pour désigner des sous-réseaux.

# Les masques inverses

- Ex: Soit l'adresse et le masque inverse suivants:

  172.16.15.64 : 10101100 . 00010000 . 00001111 . 01000000

  0.0.0.63 : 00000000 . 00000000 . 00000000 . 00111111

Ici, chaque bit de l'adresse qui correspond à un bit à 0 dans le masque devra correspondre au réseau cible.

Seuls les 6 derniers bits, qui correspondent à des 1 dans le masque, pourront varier.

# Les masques inverses

- Pour calculer rapidement un masque inverse correspondant à un masque de sous-réseau, on utilise la méthode suivante:

255.255.255.255

\- MASQUE DE SOUS-RESEAU

---

= MASQUE INVERSE

# Les masques inverses

- Ex: Soit le masque /19:

$$
\begin{array}{rl}
 & 255.255.255.255 \\
- & 255.255.224.0 \\
\hline
= & \quad 0.0.31.255
\end{array}
$$

# Les services réseau

# DHCP

- Dynamic Host Configuration Protocol
- Il permet d'obtenir automatiquement les informations pour que la machine puisse communiquer sur le réseau.
- Il faut configurer la machine client pour qu'elle puisse obtenir les informations du DHCP.
- Pour découvrir le serveur DHCP sur le réseau, nous utiliserons les trames Ethernet et les adresses MAC.

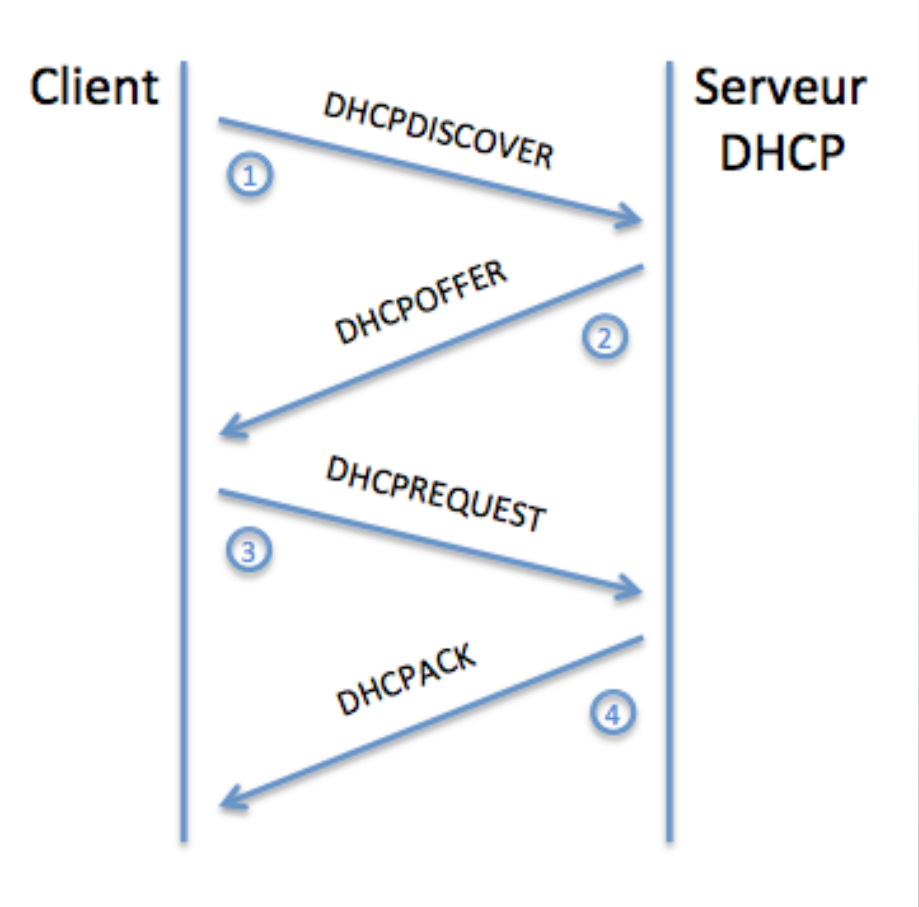
# DHCP

- Déroulement d'un échange DHCP:
  1) Envoi d'une trame DHCPDISCOVER en broadcast (sur l'adresse MAC ff:ff:ff:ff:ff:ff)

  Note: les machines doivent être sur le même réseau car les routeurs séparent les domaines de broadcast.

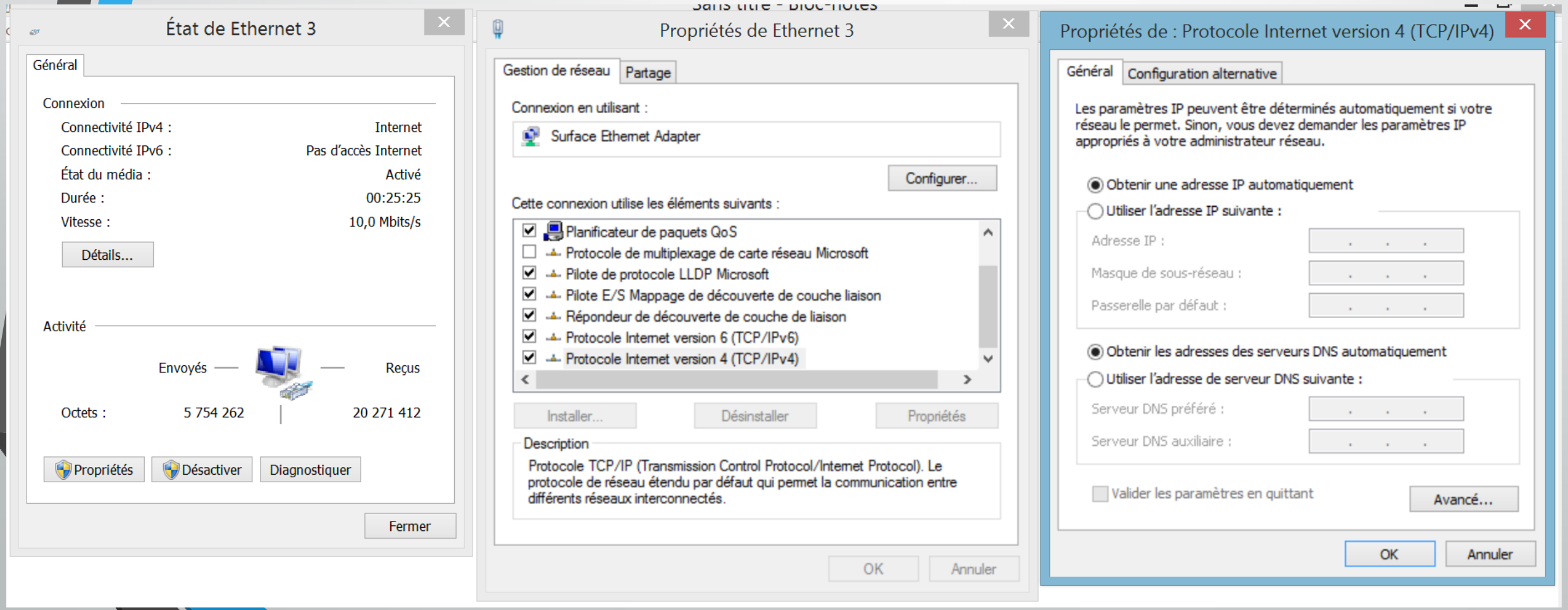
  2) Le serveur DHCP répond avec une proposition DHCPOFFER. Il propose une adresse IP et un masque et parfois une passerelle et un serveur DNS.
  3) Le client accepte la proposition via un DHCPREQUEST (toujours en broadcast).
  4) Le serveur valide la requête et envoie un DCHPACK qui valide l'allocation du bail.
- Pour renouveler le bail, le client envoie à nouveau un DHCPREQUEST.
- Les serveurs DHCP gardent en mémoire les adresses IP attribuées et les adresses MAC.

# DHCP


Client
Serveur DHCP
DHCPDISCOVER
1
DHCPOFFER
2
DHCPREQUEST
3
DHCPACK
4

# DHCP

- Configuration du poste client (Windows):



# DHCP

- Configuration du poste client (Linux):
  - CentOS:

```
[root@destiny ~]# cat /etc/sysconfig/network-scripts/ifcfg-eth0.dhcp
DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=EC:A8:6B:F6:37:20
TYPE=Ethernet
ONBOOT=yes
[root@destiny ~]#
```

# DHCP

- Configuration du poste client (Linux):
  - Debian:

```
root@jumper:/home/pi# cat /etc/network/interfaces.dhcp
# LOOPBACK
auto lo
iface lo inet loopback

#ETH0
auto eth0
iface eth0 inet dhcp

#WLAN0
allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf

#DEFAULT
iface default inet dhcp
root@jumper:/home/pi#
```

# DHCP

- Configuration du serveur DHCP dhcpd sous CentOS:
  - Installation:

    yum install dhcp
  - Configuration de l'interface dhcp dans le fichier /etc/sysconfig/dhcpd:

```
[root@destiny ~]# cat /etc/sysconfig/dhcpd
# Command line options here
DHCPDARGS=eth0
[root@destiny ~]#
```

  - Configuration du service dans le fichier /etc/dhcp/dhcpd.conf:

# DHCP

- Configuration du serveur DHCP dhcpd sous CentOS:

```
[root@destiny ~]# cat /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
#   see 'man 5 dhcpd.conf'
#

# Domain Name
option domain-name "licenceproRT.com";

# DNS server IP
option domain-name-servers 192.168.1.12, 8.8.8.8, 8.8.4.4;

# default and max lease time
default-lease-time 600;
max-lease-time 7200;

# Logs
log-facility local7;

# Subnet, IP range and Gateway
subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.129 192.168.1.254;
        option broadcast-address 192.168.1.255;
        option routers 192.168.1.1;
}

# Fixed ip addresses
host PC01 {
        hardware ethernet 60:45:BD:F9:BD:9C;
        fixed-address 192.168.1.11;
}
[root@destiny ~]#
```

# DNS

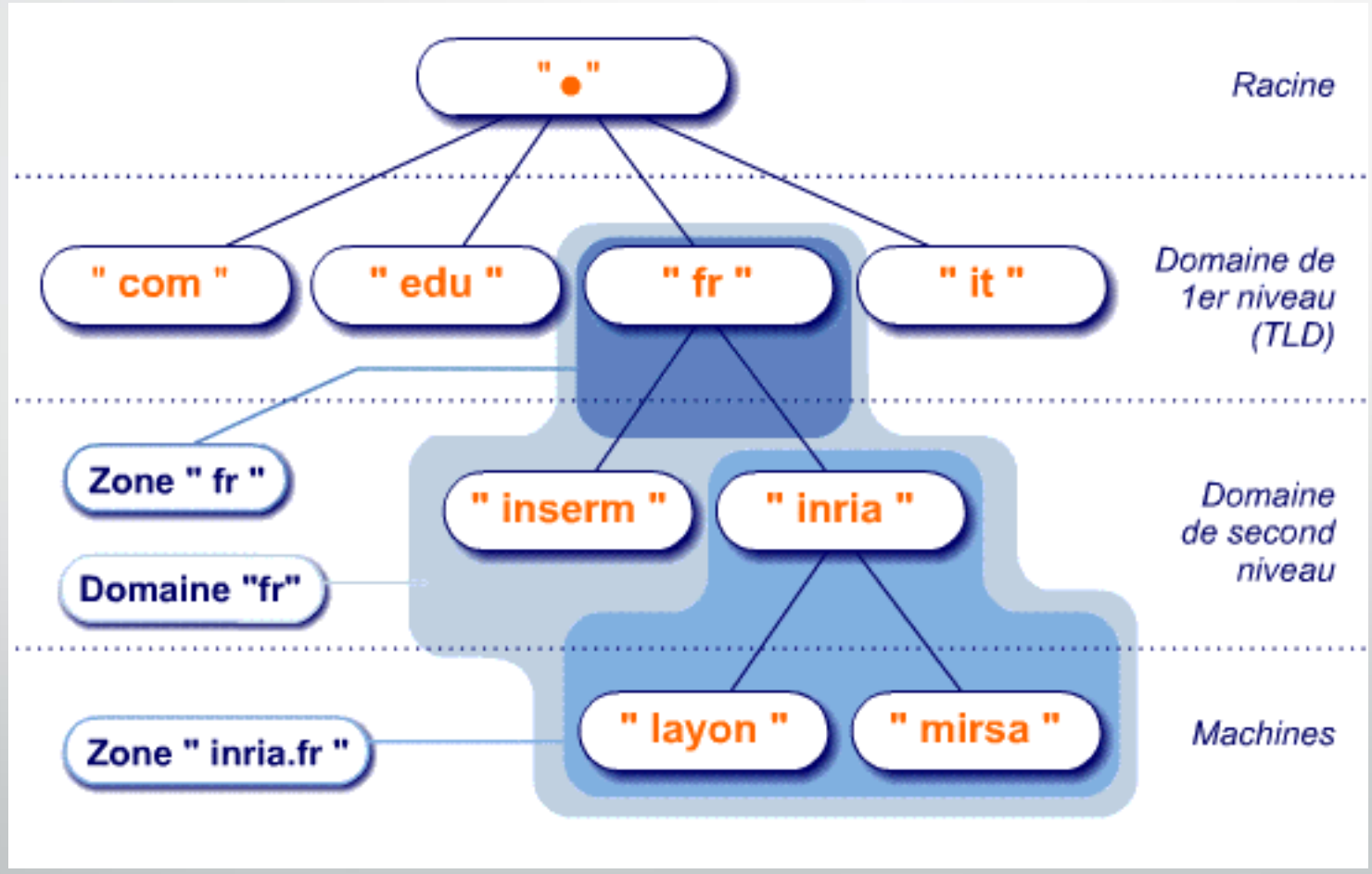
- Domain Name System
- Simplifie l'accès à Internet en proposant de joindre un site via un nom plutôt que par son adresse IP.
- DNS est indispensable au fonctionnement d'Internet!

# DNS

- DNS repose sur une arborescence.
- Chaque partie de l'arborescence est appelée label.
- Les premiers labels sous la racine sont les TLD (Top Level Domain)
  - Ex: .fr, .com, .net …
- L'ensemble des labels constituent un FQDN (Fully Qualified Domain Name)

  Note: Par convention, le FQDN doit se terminer par un point qui représente la racine.
- Chaque FQDN est unique.

# DNS


" . "
Racine
" com "
" edu "
" fr "
" it "
Domaine de 1er niveau (TLD)
Zone " fr "
" inserm "
" inria "
Domaine de second niveau
Domaine "fr"
" layon "
" mirsa "
Zone " inria.fr "
Machines

# DNS

- Chaque label est responsable des niveaux directement en dessous de lui dans l'arborescence.
- Pour obtenir la gestion d'un nom de domaine il est possible de l'acheter auprès des registrars ou que le responsable du label supérieur nous délègue la gestion du sous-domaine.
- Le système des nom de domaine est géré par l'ICANN qui est responsable des serveurs DNS racine.
- L'ICANN délègue la gestion des TLD à divers organismes (RIPE pour l'Europe, AFNIC pour le domaine .fr)

# DNS

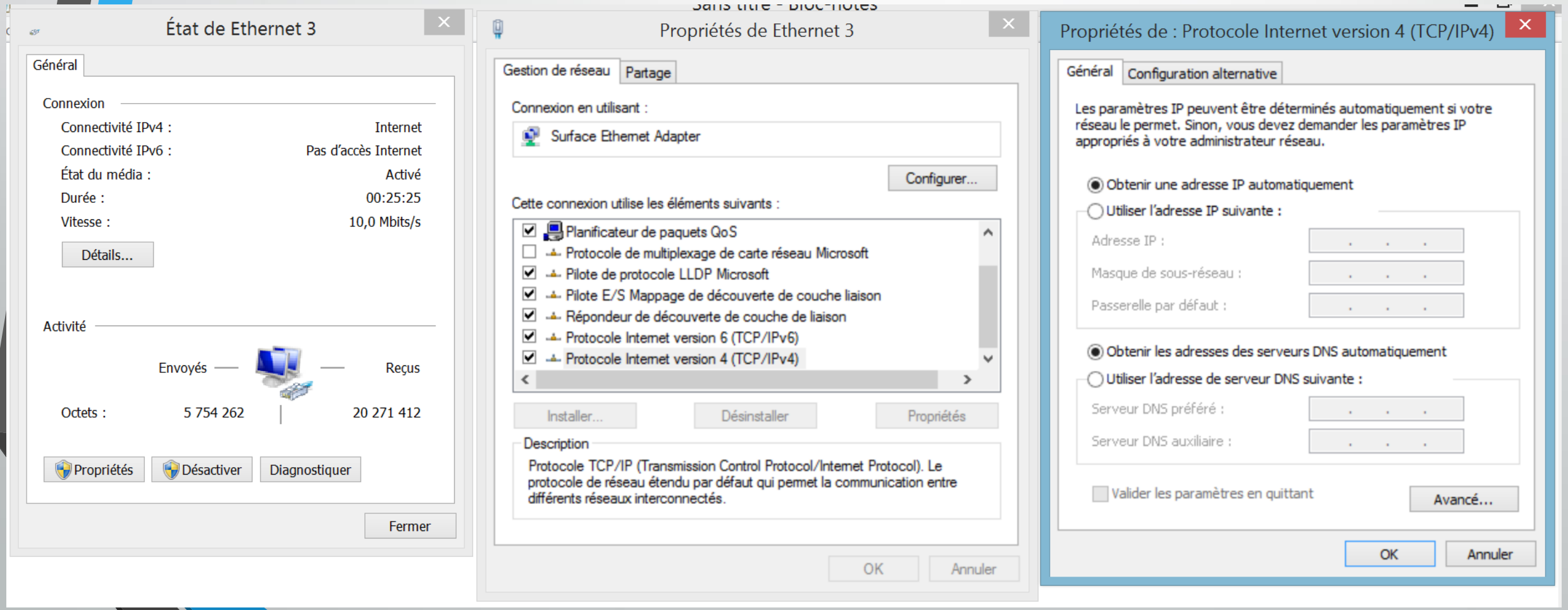
- Déroulement d'une requête DNS (on parle de résolution DNS):
  - Si le serveur connait la réponse, il la donne.
  - Sinon, il transmet la requête au serveur DNS racine qui redirigera vers le serveur DNS du TLD concerné et ainsi de suite jusqu'à obtenir la réponse pour le FQDN.
- Les serveur DNS sont capables de convertir une IP en nom de domaine: on parle alors de reverse DNS et de résolution inverse.

# DNS

- Les principaux types d’enregistrements DNS sont:
  - **A (ou AAAA sous IPv6)** : qui fait correspondre un nom d'hôte à une adresse IP.
  - **PTR** : qui associe une IP à un nom d’hôte. (Reverse DNS)
  - **CNAME** : qui permet de faire d'un domaine un alias vers un autre.
  - **MX** : qui définit les serveurs mail du domaine.
  - **NS** : qui définit les serveurs DNS du domaine.
  - **TXT** : permet d'insérer un texte quelconque dans un enregistrement DNS.

# DNS

- Configuration du poste client (Windows):



# DNS

- Configuration du poste client (Linux):
  - Dans le fichier /etc/resolv.conf

```
[root@destiny etc]# cat /etc/resolv.conf
search licenceproRT.com
nameserver 192.168.1.12
[root@destiny etc]#
```

# DNS

- Configuration du serveur DNS:
  - Installation:

    yum install bind bind-libs bind-utils
  - Ajout de l'IP du DNS dans le fichier /etc/resolv.conf:

```
[root@destiny etc]# cat /etc/resolv.conf
search licenceproRT.com
nameserver 192.168.1.12
[root@destiny etc]#
```

# DNS

- Configuration du serveur DNS:
  - Ajout des informations de zone et des paramètres du serveur DNS dans le fichier /etc/named.conf:

```
[root@destiny etc]# cat /etc/named.conf
//
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//

options {
        listen-on port 53 { 192.168.1.12; };
        directory       "/var/named";
        dump-file       "/var/named/data/cache_dump.db";
        statistics-file "/var/named/data/named_stats.txt";
        memstatistics-file "/var/named/data/named_mem_stats.txt";
        allow-query { any; };
        recursion yes;

        dnssec-enable yes;
        dnssec-validation yes;
        dnssec-lookaside auto;

        /* Path to ISC DLV key */
        bindkeys-file "/etc/named.iscdlv.key";

        managed-keys-directory "/var/named/dynamic";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
```

```
zone "." IN {
        type hint;
        file "named.ca";
};

zone "licenceproRT.com" {
        type master;
        file "db.licenceproRT.com";
};

zone "1.168.192.in-addr.arpa" {
        type master;
        file "db.192.168.1";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";

[root@destiny etc]#
```

# DNS

- Configuration du serveur DNS:
  - Configuration des fichiers de zones déclarés dans le /etc/named.conf:

Zone direct => /var/named/db.licenceproRT.com

```
[root@destiny ~]# cat /var/named/db.licenceproRT.com
$TTL 3600
$ORIGIN licenceproRT.com.
@       SOA     dns.licenceproRT.com.   root.licenceproRT.com. (
                        2014020201      ; Serial
                        28800           ; Refresh
                        14400           ; Retry
                        3600000         ; Expire
                        3600            ; Default TTL
)

@       IN      NS      dns
;@      IN      NS      dns.licenceproRT.com.
@       IN      MX      10      mx
;@      IN      MX      10      mx.licenceproRT.com.

dns     IN      A       192.168.1.12
mx      IN      A       192.168.1.12
www     IN      A       192.168.1.12

PC01    IN      A       192.168.1.11
[root@destiny ~]#
```

# DNS

- Configuration du serveur DNS:
  - Configuration des fichiers de zones déclarés dans le /etc/named.conf:

Zone reverse => /var/named/db.192.168.1

```
[root@destiny ~]# cat /var/named/db.192.168.1
$TTL 3600
$ORIGIN 1.168.192.in-addr.arpa.
@       SOA     dns.licenceproRT.com.   root.licenceproRT.com. (
                        2014020201      ; Serial
                        28800           ; Refresh
                        14400           ; Retry
                        3600000         ; Expire
                        3600            ; Default TTL
)

@       IN      NS      dns.licenceproRT.com.

1       IN      PTR     gateway.licenceproRT.com.

11      IN      PTR     PC01.licenceproRT.com.
12      IN      PTR     dns.licenceproRT.com.
[root@destiny ~]#
```

# DNS

- Cartouche de la zone:

  **$TTL 3600** => Durée de conservation des informations en cache.

  **$ORIGIN licenceproRT.com.** => Initiation de la variable ORIGIN (si absent, prendra la valeur définie dans le named.conf). Les @ que l'on retrouvera ensuite dans la zone y font référence.

  **@ SOA dns.licenceproRT.com. root.licenceproRT.com. (** => Enregistrement Start Of Authority suivi du nom du serveur DNS master de la zone et de l'adresse mail de l'administrateur de la zone.

  **2014020201 ; Serial** => Numéro de version de la zone.

  **28800 ; Refresh** => Durée de stockage des enregistrements sur le serveur Slave.

  **14400 ; Retry** => Temps d'attente avant que le serveur Slave tente de recontacter le serveur Master s'il est occupé.

  **3600 ; Expire** => Durée durant laquelle le serveur Slave tentera de contacter le serveur Master.

  **28800 ; Default TTL** => Durée minimale du cache.

  **)**

# NTP

- Network Time Protocole
- Il permet de synchroniser l'horloge d'un ordinateur avec celle d'un serveur de référence.
- Il existe différents niveaux (strates) qui correspondent à différentes précisions.
- Il y a trois méthodes pour diffuser l’heure:
  - Mode client/serveur.
  - Mode symétrique actif/passif : les nœuds échangent leur rôle tour à tour.
  - Mode broadcast : diffusion périodique est spontanée par le serveur dans un réseau local.
- Nous sommes actuellement à la version 4 du protocole.

# NTP

- Sous linux, les configurations client et serveur se font dans le fichier /etc/ntp.conf (la différence étant le fait d'autoriser des IP à interroger le service NTP):

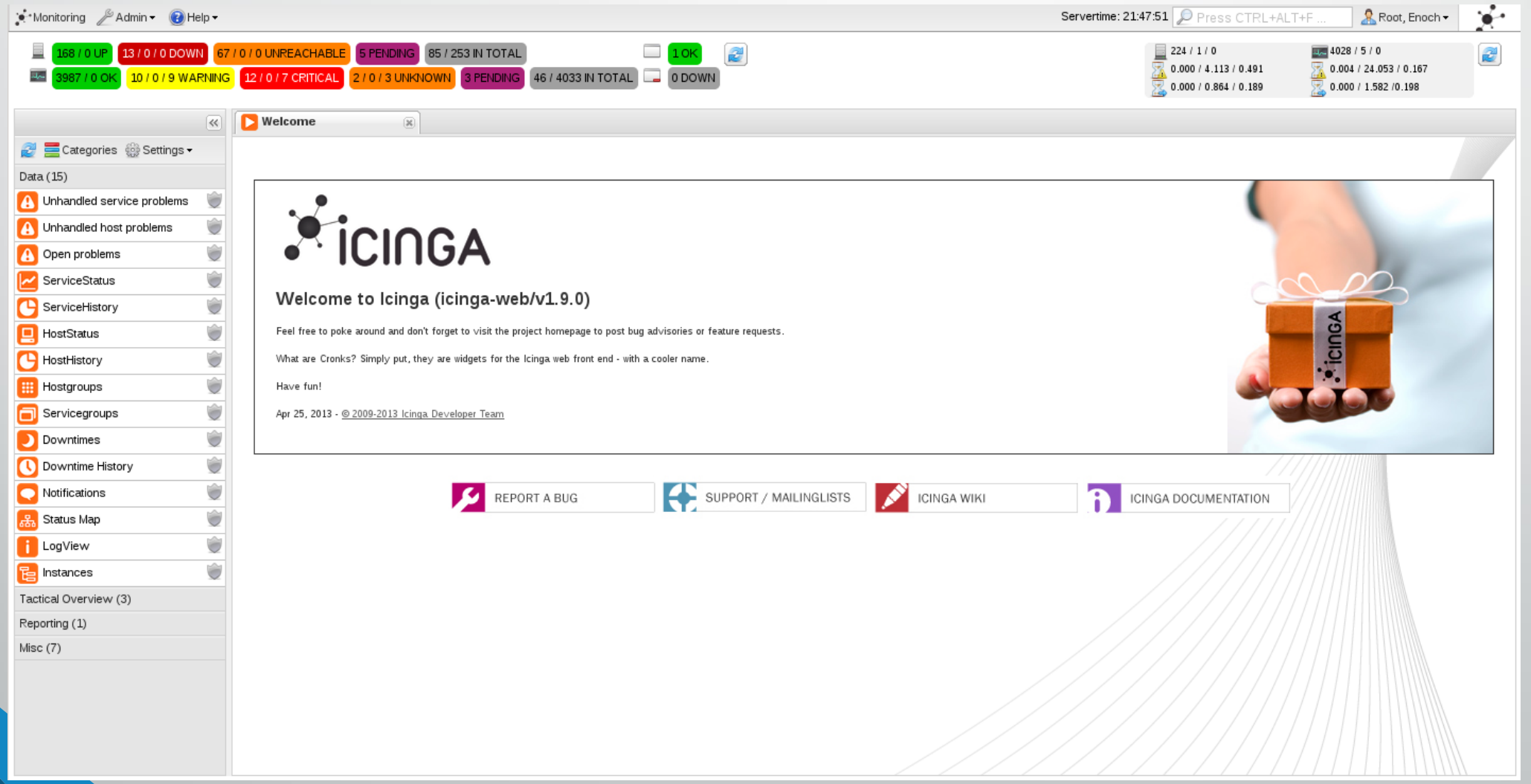
```
[root@destiny ~]# cat /etc/ntp.conf
# For more information about this file, see the man pages
# ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5), ntp_mon(
5).

driftfile /var/lib/ntp/drift

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
#restrict default kod nomodify notrap nopeer noquery
#restrict -6 default kod nomodify notrap nopeer noquery
restrict default ignore

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1

# Hosts on local network are less restricted.
restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.centos.pool.ntp.org iburst
server 1.centos.pool.ntp.org iburst
server 2.centos.pool.ntp.org iburst
server 3.centos.pool.ntp.org iburst

#broadcast 192.168.1.255 autokey        # broadcast server
#broadcastclient                        # broadcast client
#broadcast 224.0.1.1 autokey            # multicast server
#multicastclient 224.0.1.1              # multicast client
#manycastserver 239.255.254.254         # manycast server
#manycastclient 239.255.254.254 autokey # manycast client
```

```
# Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8

# Specify the key identifier to use with the ntpq utility.
#controlkey 8

# Enable writing of statistics records.
#statistics clockstats cryptostats loopstats peerstats
[root@destiny ~]#
```

# SNMP

- Simple Network Management Protocol
- C'est un protocole de communication qui permet la gestion et la supervision des équipements du réseau.
- Il permet la communication entre deux principaux éléments, le superviseur et les agents.
- Le superviseur est une console qui permet d'exécuter des requêtes.
- Les agents se trouvent au niveau des interfaces connectant l'équipement managé au réseau, et permettant de récupérer des informations.
- Les équipements contiennent des objets manageables (qui peuvent être des informations, des paramètres de configuration, des statistiques de performances…).
- Chaque objet est identifié à l'aide d'une suite d'entiers appelée OID (Object Identifier).
- Les OID sont classés dans une base appelée MIB (Management Information Base).

# SNMP


Monitoring
Admin
Help
Servertime: 21:47:51
Press CTRL+ALT+F ...
Root, Enoch
168 / 0 UP
13 / 0 / 0 DOWN
67 / 0 / 0 UNREACHABLE
5 PENDING
85 / 253 IN TOTAL
1 OK
3987 / 0 OK
10 / 0 / 9 WARNING
12 / 0 / 7 CRITICAL
2 / 0 / 3 UNKNOWN
3 PENDING
46 / 4033 IN TOTAL
0 DOWN
224 / 1 / 0
0.000 / 4.113 / 0.491
0.000 / 0.864 / 0.189
4028 / 5 / 0
0.004 / 24.053 / 0.167
0.000 / 1.582 /0.198
Welcome
Categories
Settings
Data (15)
Unhandled service problems
Unhandled host problems
Open problems
ServiceStatus
ServiceHistory
HostStatus
HostHistory
Hostgroups
Servicegroups
Downtimes
Downtime History
Notifications
Status Map
LogView
Instances
Tactical Overview (3)
Reporting (1)
Misc (7)
Welcome to Icinga (icinga-web/v1.9.0)
Feel free to poke around and don't forget to visit the project homepage to post bug advisories or feature requests.
What are Cronks? Simply put, they are widgets for the Icinga web front end - with a cooler name.
Have fun!
Apr 25, 2013 - © 2009-2013 Icinga Developer Team
REPORT A BUG
SUPPORT / MAILINGLISTS
ICINGA WIKI
ICINGA DOCUMENTATION

# Les outils réseau

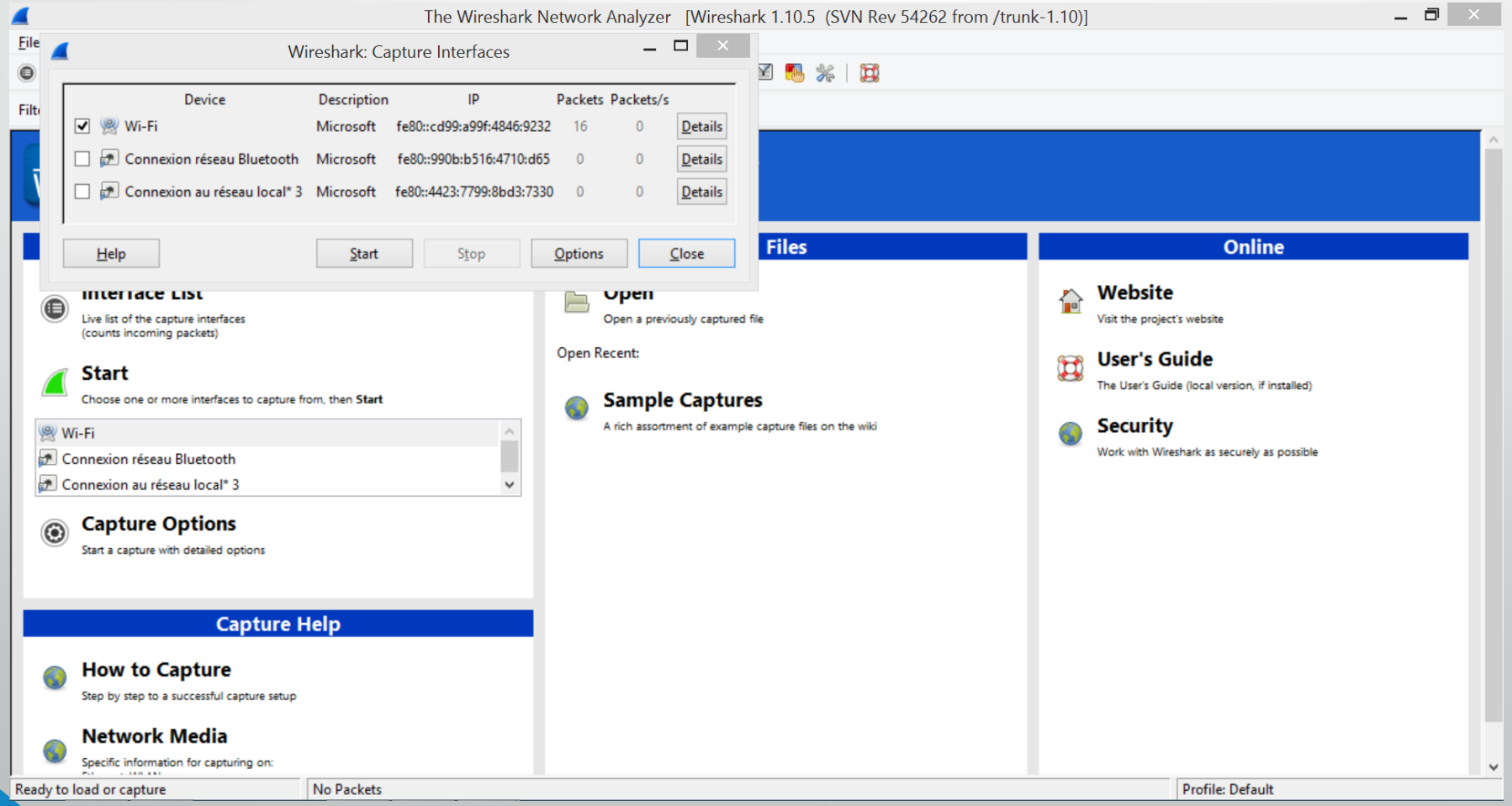
# Les outils réseau

- **ipconfig / ifconfig** : Voir les informations des cartes réseau.
  - Ifconfig permet également de configurer une interface réseau sous linux.
- **tracert / traceroute** : Voir le chemin parcouru par les paquets.
- **route PRINT / route -n ou netstat –nr** : Connaitre la table de routage.
  - Le commande route permet de modifier la table de routage d'une machine linux.
- **netstat –an** : Voir la liste des ports en écoute.
  - netstat –nl sous linux donne le même résultat, plus épuré.
- **ping** : Tester l'accès à une autre machine.
- **arp** : Voir la table arp d'une machine.
- **nslookup / host / dig** : Tester une résolution DNS.

**tcpdump** : Sniffer le trafic.

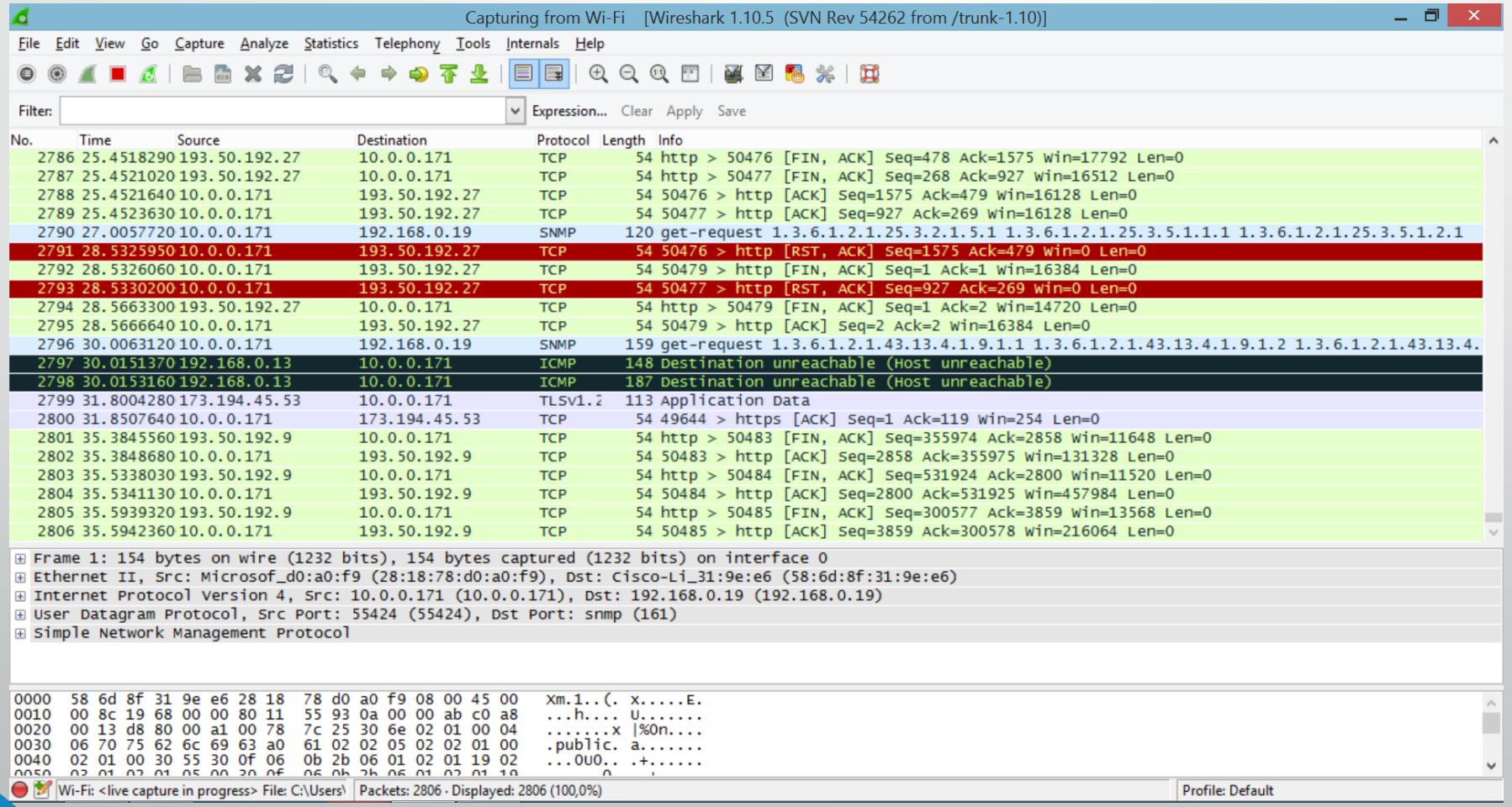
# Les outils réseau

- Wireshark: Sniffer de trafic.



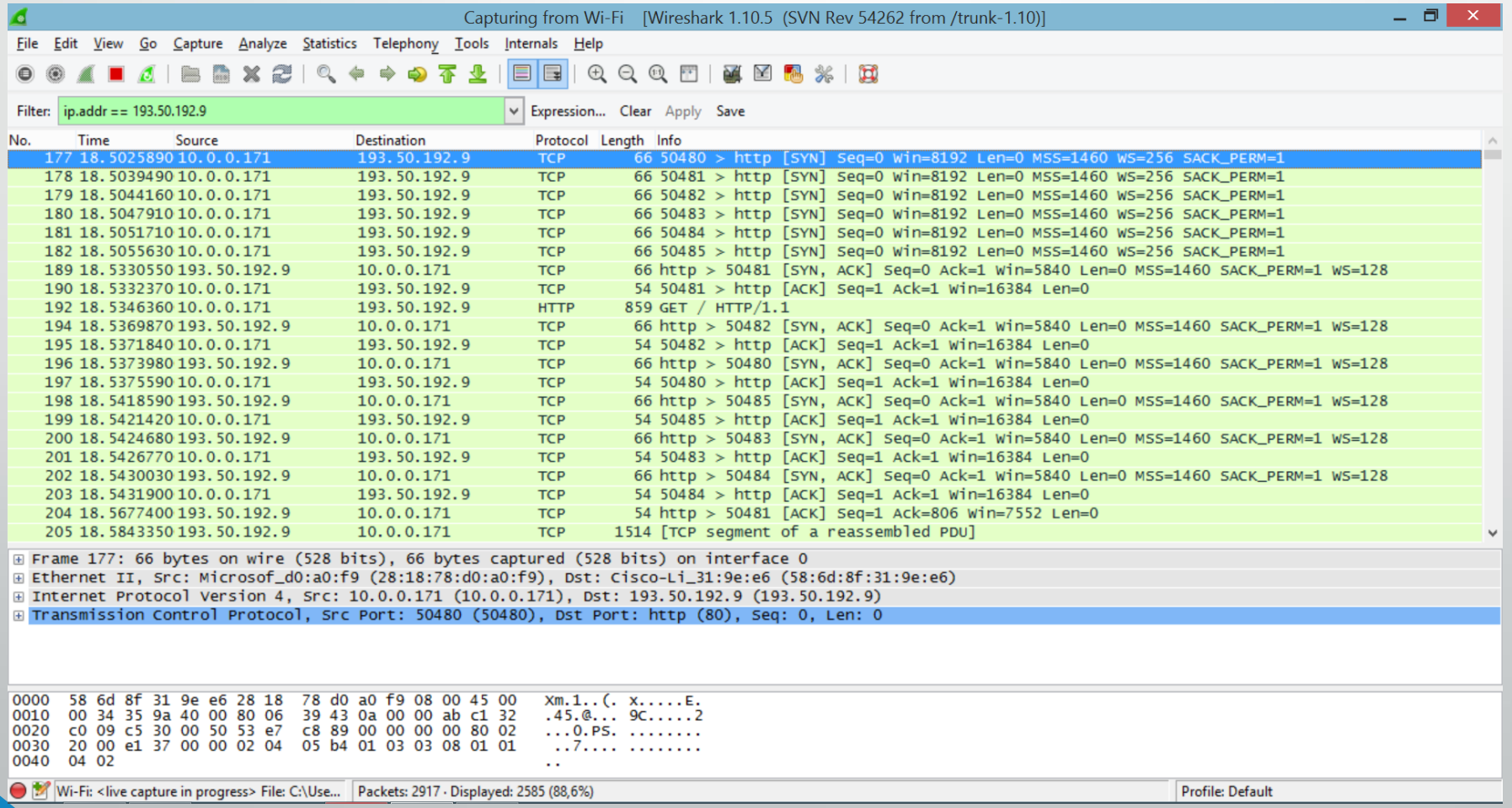
# Les outils réseau

- Wireshark:



# Les outils réseau

- Wireshark:



# Les outils réseau

- Wireshark: