

# Introduction aux réseaux informatique

**Thomas Vantroys**

`thomas.vantroys@univ-lille.fr`

Polytech'Lille  
Université de Lille

**DIU**

2018 - 2019



Université  
de Lille



**POLYTECH'**  
LILLE

# Objectifs

## Objectifs :

- Comprendre le fonctionnement d'un réseau local
- Comprendre le routage
- Connaître les principaux protocoles d'Internet

# Rappel du programme officiel

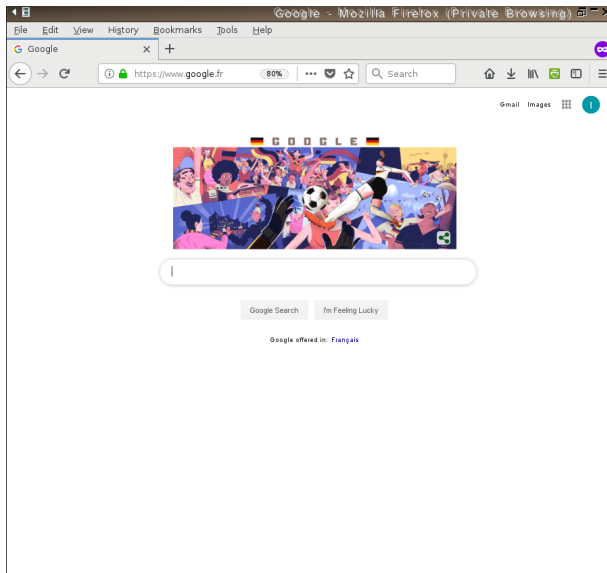
## Programme de NSI première générale :

- Contenus
  - Transmission de données dans un réseau
  - Protocoles de communication
  - Architecture d'un réseau
- Capacités attendues
  - Mettre en évidence l'intérêt du découpage des données en paquet et de leur encapsulation
  - Dérouler le fonctionnement d'un protocole simple de récupération de perte de paquets (bit alterné)
  - Simuler ou mettre en oeuvre un réseau
- Commentaires
  - Le protocole peut être expliqué et simulé en mode débranché
  - Le lien est fait avec ce qui a été vu en classe de seconde sur le protocole TCP/IP
  - Le rôle des différents constituants du réseau local de l'établissement est présenté

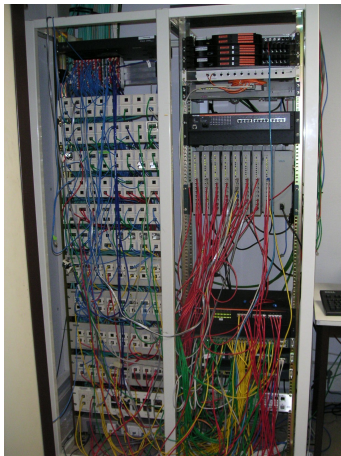
# Plan

- 1 Modèle OSI et IETF
- 2 TCP/IP
  - IP
  - ICMP
- 3 Passerelles Applicatives
- 4 La couche transport
- 5 DNS
- 6 Résumé

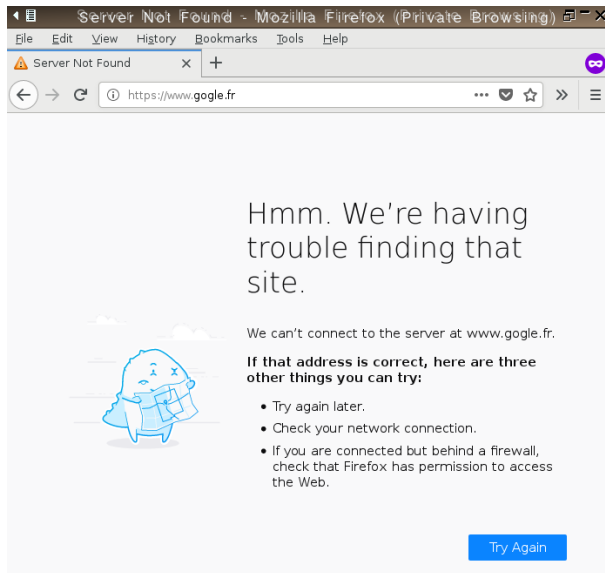
# Le réseau : vue utilisateur



# Le réseau : vue administrateur



# Que faire ?



# Définitions

- **Réseau** : structure définie par des relations entre entités - Ensemble d'ordinateur ou de terminaux interconnectés par des télécommunications générales permanentes ou non.
- Un réseau peut se résumer à trois catégories de constituants :
  - ❶ le **matériel**, i.e., transmission "physique" de l'information .  
Par exemple : commutateur, routeur, ... ;
  - ❷ les **protocoles**, i.e., l'ensemble de règles définissant le mode de communication entre deux entités. Par exemple : IPv4, Ethernet, TCP, HTTP, ... ;
  - ❸ les **applications**. Par exemple : navigateur web, lecteur de mail, ...

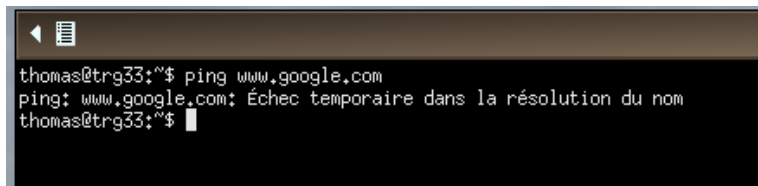


# Résolution de problème

- déterminer l'origine du problème (matériel, protocole application)
- première approche : la commande ping
  - envoie d'une "demande d'écho" à une machine distante
  - exemple :

```
thomas@trg33:~$ ping www.google.com
PING www.google.com (216.58.209.228) 56(84) bytes of data:
64 bytes from par10s29-in-f4.1e100.net (216.58.209.228): icmp_seq=1 ttl=53 time=5.19 ms
64 bytes from par10s29-in-f4.1e100.net (216.58.209.228): icmp_seq=2 ttl=53 time=10.1 ms
64 bytes from par10s29-in-f4.1e100.net (216.58.209.228): icmp_seq=3 ttl=53 time=5.74 ms
64 bytes from par10s29-in-f4.1e100.net (216.58.209.228): icmp_seq=4 ttl=53 time=5.23 ms
^C
--- www.google.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 5.198/6.583/10.157/2.076 ms
thomas@trg33:~$
```

# Résolution de problème

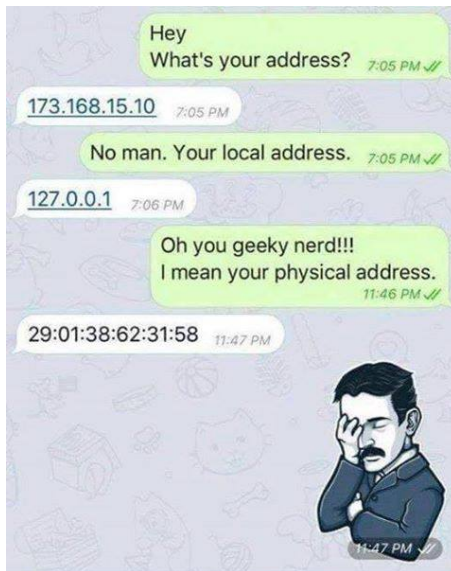
A terminal window with a dark background and a light-colored title bar. The title bar contains a back arrow icon and a list icon. The terminal text shows a user named 'thomas' at a prompt 'trg33' running the command 'ping www.google.com'. The output of the command is 'ping: www.google.com: Échec temporaire dans la résolution du nom', followed by a new prompt 'thomas@trg33:~\$' and a cursor.

```
thomas@trg33:~$ ping www.google.com
ping: www.google.com: Échec temporaire dans la résolution du nom
thomas@trg33:~$
```

- à quel niveau (matériel, logiciel) se situe le problème ?
- dans l'exemple, le problème est lié au nom de la machine

**Comment est adressée une machine ?**

# Adresse(s)



# Adresse(s)

Exemple d'adresses associées à une seule machine :

- nom d'une machine : `trg38.univ-lille1.fr`
- adresse IPv4 : `134.206.90.38`
- adresse MAC (matérielle) : `00:20:AF:04:E8:FF`

## **Pourquoi autant d'adresses pour une seule machine ?**

- la communication réseau s'appuie sur une pile protocolaire
- chaque protocole possède son adressage

Pour comprendre les différentes couches protocolaires, un modèle "théorique" : le modèle OSI

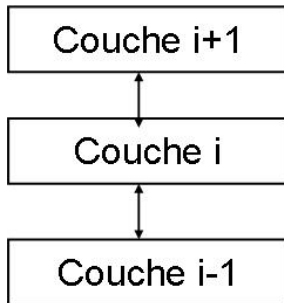
# Le modèle OSI

# Modèle OSI

- **OSI** = **O**pen **S**ystems **I**nterconnection de l'ISO
- Un cadre utilisé pour aider au développement des standards de télécommunication
- Structuration en 7 couches pour mieux déterminer les fonctionnalités prises en charge par chacun des composants du logiciel et du matériel
- C'est un modèle de référence, de très haut niveau
- Ne définit pas les protocoles

# Principes de bases

Une couche fournit un service à une couche supérieure en s'appuyant sur le service fourni par la couche inférieure



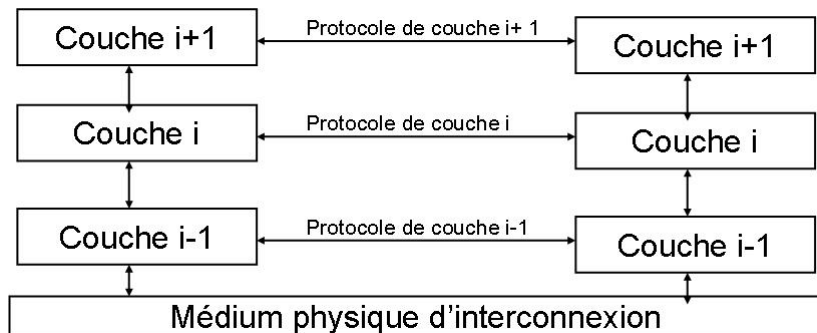
# Principes de bases

- Une couche = une boîte noire
- La couche de niveau supérieur utilise le service mais ignore le détail de réalisation (la mise en œuvre)
- Une couche inférieure présente le service avec un haut niveau d'abstraction



# Principes de bases

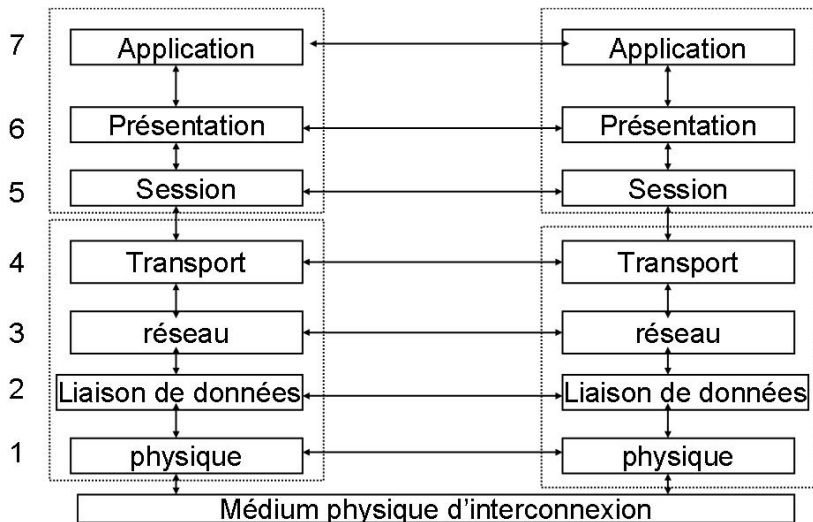
Deux couches de même niveau appartenant à deux systèmes interconnectés dialogue en utilisant le même protocole



# Protocole

- C'est un ensemble de règles et de conventions pour la conversation
- Évidemment deux systèmes interconnectés doivent utiliser le même protocole pour dialoguer entre eux
- Le protocole de niveau inférieur "encapsule" toutes les informations gérées par le protocole du niveau supérieur (sans avoir à en connaître la structuration) : il y a donc indépendance entre les niveaux

# Les 7 couches



# Le modèle OSI par D. Comer

<http://www.cs.purdue.edu/homes/dec/essay.network.layers.html>

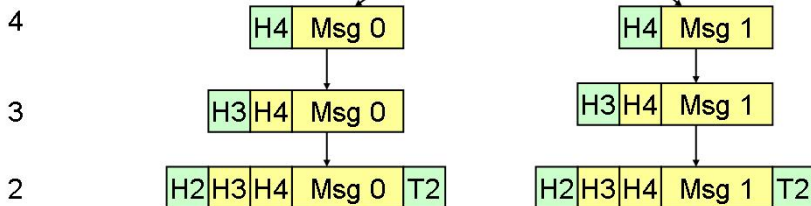
Couche	Nain	Nom
1	Dormeur (Sleepy)	Physique
2	Atchoum (Sneezy)	Liaison
3	Joyeux (Happy)	Réseau
4	Prof (Doc)	Transport
5	Simplet (Dopey)	Session
6	Timide (Bashful)	Présentation
7	Grincheux (Grumpy)	Application

# Exemple : Fonctions de Transport couches 1 à 4

Soit une machine ayant à envoyer un message msg : ici msg est trop long, il est découpé en deux messages (msg0 et msg1).

Chaque couche ajoute des éléments de son protocole dans des entêtes Hx ou/et des terminaisons Tx

## Émission

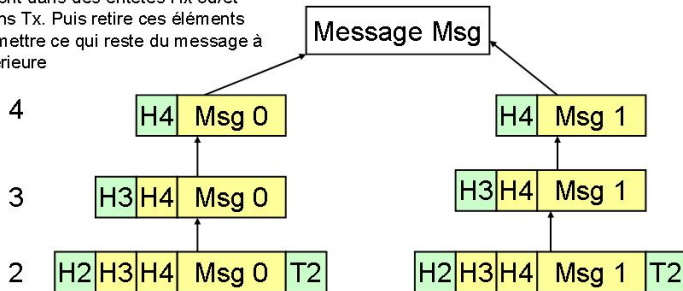


# Exemple : Fonctions de Transport couches 1 à 4

Soit une machine recevant deux messages  
msg0 et msg1 :

Chaque couche analyse les éléments de son  
protocole qui sont dans des entêtes Hx ou/et  
des terminaisons Tx. Puis retire ces éléments  
avant de transmettre ce qui reste du message à  
la couche supérieure

Réception



# La couche physique

- moyens mécaniques, électriques, radio, . . . , fonctionnels pour maintenir, activer, désactiver les connexions physiques pour la transmission d'éléments binaires entre liaisons
- unité = bit
- ex : codeur, modulateur, multiplexeur
- la partie "électronique" des réseaux
- Préoccupations :
  - Nb de volts (pallier), fréquences, synchronisation
  - Durée d'un signal/bit
  - Support physique

# Équipements réseau

- Câblage : cuivre (paire torsadée), fibre optique
- Transceiver : Transformation de signal (ex : électrique vers optique)
- Repeater : Répète et amplifie les signaux
- Concentrateur (*hub*)



# Un hub 8 ports



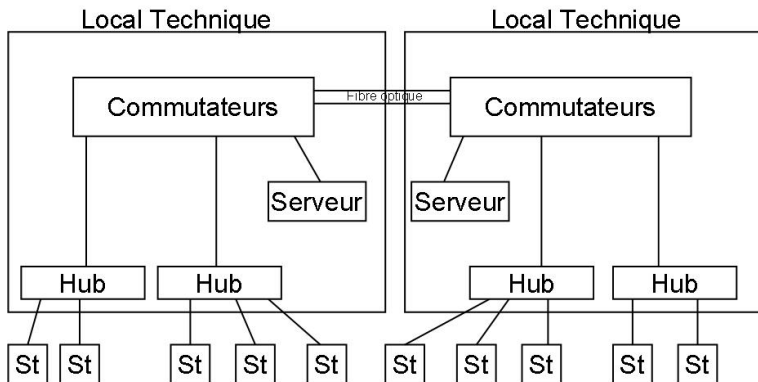
# La couche liaison de données

- Structure les données en trames
- Transmet les trames en séquences
- gestion des erreurs de transmission
- doit optimiser l'utilisation de la connexion physique
- unité = trame
- ex : HDLC
- Préoccupations :
  - Éviter le submergement
  - Contrôler l'accès partagé à un canal
- Matériel : commutateur (*switch*)

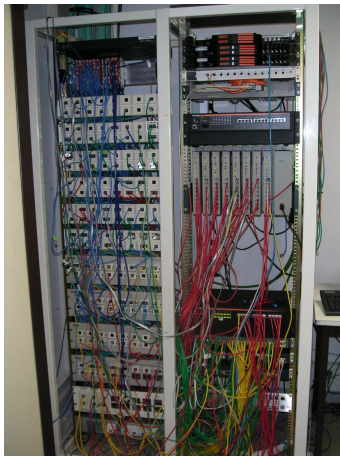
# Commutateur Ethernet



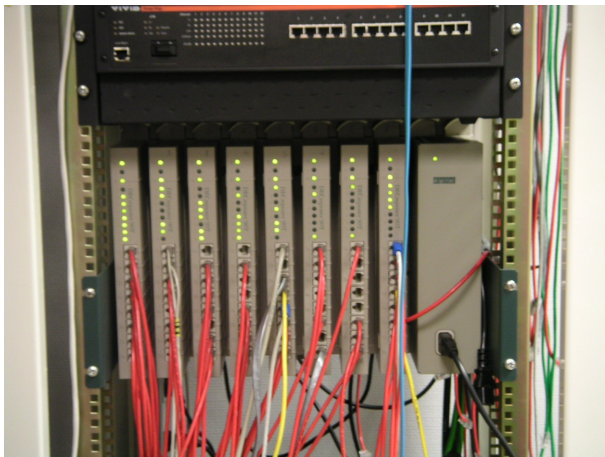
# Organisation physique des réseaux



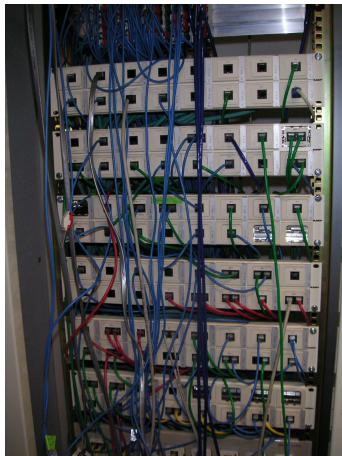
# Organisation physique des réseaux



# Organisation physique des réseaux



# Organisation physique des réseaux



# La couche réseau

- Éventuellement établit et gère une connexion de réseau
- Acheminement de paquets de la source vers la destination
- Pour cela, elle définit :
  - adressage
  - routage
  - contrôle de flux
- ex : IP
- Matériel : Routeur



# La couche transport

- Découpe les données en "paquets" de taille acceptable par le réseau utilisé
- En réception "ordonne" les paquets avant de reconstituer le message pour le groupe "application"
- Optimise les ressources réseau
- Assure un service de "bout en bout"
- Permet le multiplexage de plusieurs communications d'applications
- ex : TCP, UDP

# La couche session

- Responsable de la synchronisation
- Cadence le dialogue des applications (peut émettre, attendre, ...)
- Permet d'effectuer des reprises dans le dialogue
- Gère les transactions
- ex : les cookies HTTP

# La couche présentation

- S'intéresse à la syntaxe et sémantique des données transmises
- Service de chiffrement ou de compression
- Ex : HTML, XML

# La couche application

- Offre aux applications les moyens d'accès à la "pile OSI"
- fournit des classes de services
- ex : terminal virtuel, transfert de fichiers, courrier électronique, web

# Deux grands modes de fonctionnement

- Une couche peut fonctionner en mode connecté ou non (ex : IP non connecté, TCP connecté)
- La connexion permet la négociation de paramètres propres à chaque utilisation du service
- La connexion "allourdit" la communication car il y a un préambule
- Exemple : UDP est un protocole de transport (niveau 4) **non connecté** tandis que TCP est aussi un protocole de transport mais lui est **connecté**

# Modèle IETF

# Modèle ISO et IETF

- IETF = Internet Engineering Task Force
- Définit les protocoles de l'Internet et les règles d'utilisation du réseau
- Spécifie les 4 couches liées aux fonctions de transport
- IETF : Plus que des protocoles des API simples et largement diffusées (socket)
- Moins strict que l'ISO avec l'indépendance entre les couches (TCP est un peu lié à IP)

# Modèle de l'IETF

- Modèle en 4 couches :
  - physique
  - liaison de données
  - réseau (IP et ICMP)
  - transport (TCP ou UDP)
- Et un ensemble de protocoles pour les applications (ex : HTTP) et les services (ex : DNS)



# Comparaion OSI vs TCP/IP

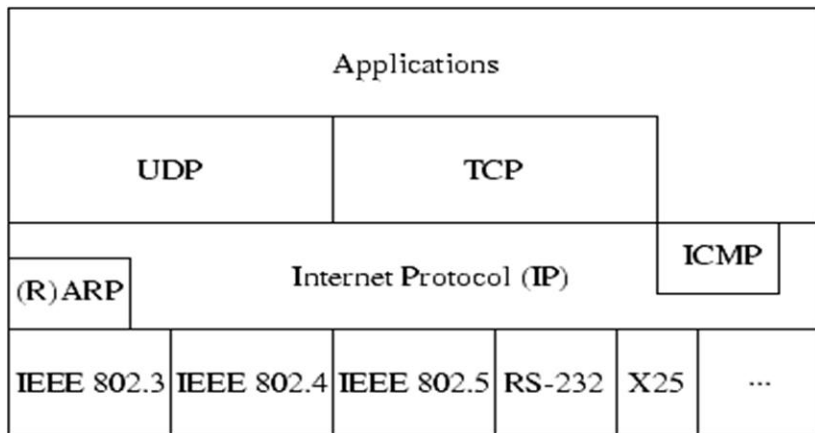
## Couches OSI

Application
Présentation
Session
Transport
Réseau
Liaison
Physique

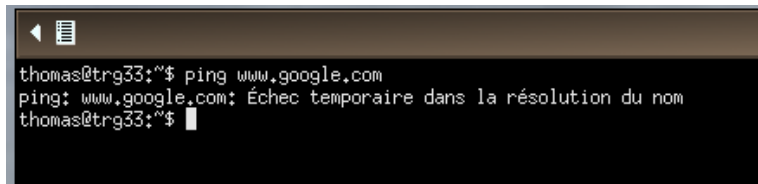
## Couches TCP/IP

Application
Transport (UDP,TCP)
Internet (IP)
Interface
Matériel (IEEE 802)

# Les protocoles d'internet



# Retour au problème

A terminal window with a dark background and a light-colored title bar. The title bar contains a back arrow icon and a list icon. The terminal text shows a user named 'thomas' at a prompt 'trg33:~\$' running the command 'ping www.google.com'. The output of the command is 'ping: www.google.com: Échec temporaire dans la résolution du nom', followed by the prompt 'thomas@trg33:~\$' and a cursor.

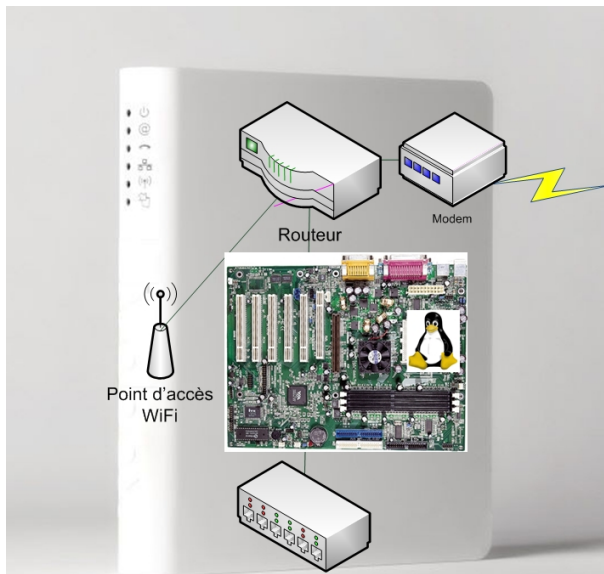
```
thomas@trg33:~$ ping www.google.com
ping: www.google.com: Échec temporaire dans la résolution du nom
thomas@trg33:~$
```

**Remonter le modèle OSI pour trouver l'origine du problème**

# Le matériel



# Le matériel



# La configuration logicielle

- la commande `ip` permet de connaître la configuration complète de la machine

```
thomas@trg33:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
   link/ether 00:26:2d:f7:4b:d8 brd ff:ff:ff:ff:ff:ff
   inet 134.206.25.45/22 brd 134.206.27.255 scope global eth0
       valid_lft forever preferred_lft forever
   inet6 fe80::226:2dff:fef7:4bd8/64 scope link
       valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
   link/ether cc:3d:82:bc:d4:dc brd ff:ff:ff:ff:ff:ff
```

# Paramètres réseau d'une machine

En réseau IPv4, pour communiquer une machine doit posséder :

- Une adresse IP de 4 octets où chaque octet est codé en décimal et un point sépare chaque octet
- Dans cette adresse IP, il y a deux informations :
  - Bits de poids fort : numéro de réseau
  - Bits de poids faible : numéro de machine dans ce réseau
- Pour extraire ces deux informations de l'adresse IP, la machine doit connaître le nombre de bits du numéro de réseau
- Élément important car méthodes d'envoi différentes si les machines sont dans le même réseau ou si elles sont dans des réseaux différents

# Numéro de réseau

- La représentation précise d'une adresse IP peut être donnée sous la forme : @IP/nb de bits du numéro de réseau
  - Ex : 132.231.3.2/16 signifie que le numéro de réseau est de longueur 16 bits donc 132.231 et par convention on le note 132.231.0.0
  - Ex : 193.201.125.4/24, 24 bits pour le rx, donc 193.201.125 et par convention 193.201.125.0
- Le numéro de réseau peut prendre un nombre de bits quelconque et donc ne pas être calé sur une frontière d'octet
  - Ex : 172.168.129.4/20, rx sur 20 bits donc 172.168.128 et par convention 172.168.128.0



# Masque de sous-réseau

En général dans les logiciels de paramétrage IP, on indique le numéro du réseau par l'intermédiaire du masque de sous-réseau qui se note comme l'adresse IP mais où il y a des valeurs 1 pour chaque bit représentant le numéro de réseau

- 132.231.3.2/16 se note en fait :
  - @IP : 132.231.3.2
  - masque de sous-réseau : 255.255.0.0
- 193.201.125.4/24
  - @IP : 193.201.125.4
  - masque : 255.255.255.0
- 172.168.129.4/20
  - @IP : 172.168.129.4
  - masque : 255.255.240.0

# Les paramètres IP

En résumé, la liste minimale des paramètres IP à indiquer à une machine est :

- adresse IP de la machine
- Masque de sous-réseau
- adresse IP de la passerelle du réseau (le routeur)
- adresse IP du serveur de nom (DNS)

# Paramétrage IP

- Le paramétrage IP peut être statique :  
L'attribution des adresses IP se fait par l'administrateur qui tient à jour une table. Quand on met une machine en service on lui fixe sa configuration.
- Le paramétrage IP peut être dynamique. Lorsque la machine se met sous tension, elle demande sa configuration à un serveur (protocole DHCP) à qui l'administrateur a indiqué toutes les informations. @ attribuées à la volée et une machine n'aura pas toujours la même @

# Paramétrage IP : exemple

- Adresse Physique : 00 :20 :af :04 :e8 :ff
- Adresse IP : 134.206.90.38
- Masque de Réseau : 255.255.0.0
- Adresse Réseau : 134.206.0.0
- Adresse de diffusion : 134.206.255.255

# Interconnexion des réseaux

# Le mode non-connecté

- Définit le **datagramme** comme unité d'échange
  - des éléments d'informations de taille limitée pouvant circuler entre les machines et entre les réseaux
  - nécessite en en-tête avec des **adresses** sources et destinations
  - le datagramme doit pouvoir être *fragmenté* par des équipements réseaux intermédiaires

# La communication élémentaire

- Chaque machine d'un réseau doit offrir un mécanisme de remise de datagrammes :
  - à une machine particulière de son réseau local
  - à toutes les machines de son réseau local
- Un protocole de liaison de données (couche 2) est obligatoire (ex : PPP ou Ethernet IEEE 802.3) pour transporter le datagramme

# L'adressage

- Chaque machine dispose d'une adresse physique fournie par son raccordement au réseau physique
  - ex : Ethernet : adresse MAC
  - ex : RNIS : numéro de téléphone
- Nécessité d'un adressage logique global
  - indépendance de l'opérateur ou de la nature du réseau
  - numéro de réseau suivi du numéro de machine pour ce réseau
- Les éléments d'informations circulant entre les réseaux ne portent que l'adresse logique globale
- Il peut y avoir deux types d'adresses
  - des adresses privées (ne circulant pas entre des réseaux publics)
  - des adresses publiques



# Mécanisme de base

Lorsqu'une machine doit remettre un datagramme :

- elle teste l'adresse globale pour savoir si le destinataire appartient au même réseau
- Si oui : elle doit remettre le datagramme à cette machine destinataire elle-même (remise directe)
- Si non : elle doit remettre le datagramme à un équipement d'interconnexion de réseau (ex : routeur)

# Remise d'un datagramme à une machine d'un même réseau

Pour remettre un datagramme à une machine d'un même réseau :

- l'expéditeur (la machine) doit posséder son adresse physique
- s'il ne l'a pas, il doit l'obtenir
- pour l'obtenir il doit la demander :
  - à un serveur de correspondance d'adresses physiques à adresses globales
  - à l'ensemble des machines du réseau (diffusion) et seule la machine possédant cette adresse répond à l'émetteur (ex : ARP)

# Communication avec les équipements d'interconnexion de réseaux

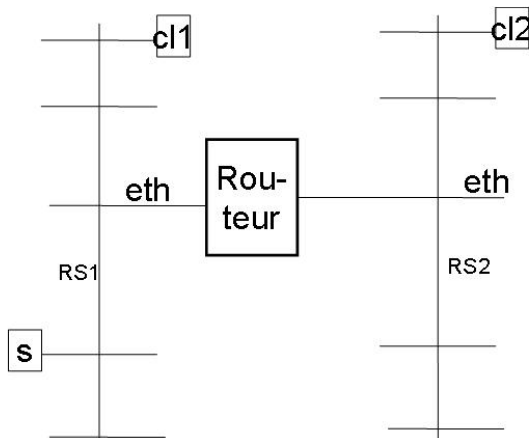
- L'expéditeur doit connaître l'adresse globale de l'équipement d'interconnexion de réseau
- Il remet alors le datagramme qui porte les adresses globales de la station source et du destinataire final à cet équipement dont le rôle est de prendre en charge l'acheminement du datagramme

# Exemple d'échange sur le réseau avec hub

S doit remettre un datagramme à cl

- s rédige le datagramme avec :
  - ad source : @g(s)
  - ad destination : @g(cl)
- s teste @g(cl) avec @g(s)
- ici même réseau donc **remise directe** du datagramme
- s demande et obtient @p(cl,rs1)
- s envoie une "trame" ethernet à destination de @p(cl,rs1) portant le datagramme

# Interconnexion de deux réseaux locaux par un routeur



@g(m) : adresse globale machine m

@p(m,rs) : adresse physique machine m sur réseau rs

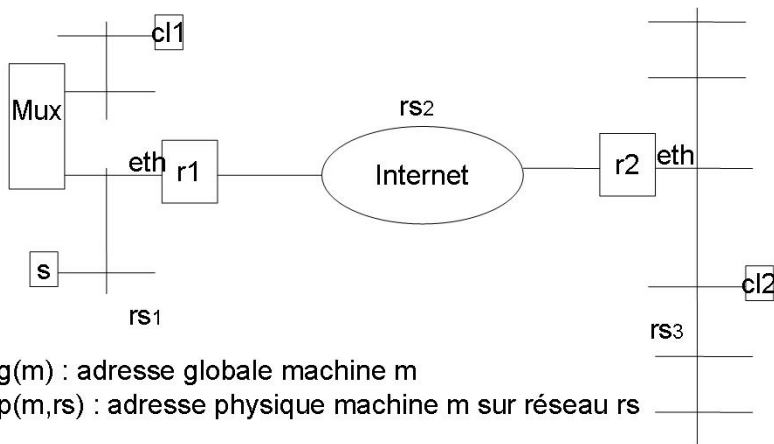
# Un routeur



# Travail des équipements d'interconnexion

- Chaque accès de l'équipement peut avoir une adresse globale différente dont les numéros de réseau sont différents
- Pour les réseaux qui n'ont pas de machines connectées (uniquement des équipements réseaux) il n'y a pas besoin d'adresse globale
- Lire les en-têtes de datagrammes
- Réaliser les opérations de routage
- Optimiser la traversée des datagrammes. Pour cela les équipements vont dialoguer entre eux
- Détecter les vieux datagrammes qui ne trouvent pas la sortie et les éliminer

# Interconnexion de deux réseaux locaux au travers d'internet



@g(m) : adresse globale machine m

@p(m,rs) : adresse physique machine m sur réseau rs



# Application aux réseaux IP

- En version 4 :
  - @ globale 4 octets
  - numéro réseau suivi du numéro de machine sur le réseau
  - le numéro de réseau possède une taille variable
- IP accompagné d'un protocole de "contrôle" : ICMP (Internet Control Message Protocol)
- ICMP constitue le cœur de deux commandes de test d'un réseau IP
  - ping (vérification de l'envoi/réception d'information d'une machine vers une autre)
  - traceroute (obtention de la liste des équipements de niveau 3 qui sont traversés (routeurs))
- Sur réseau Ethernet : le protocole ARP établit la correspondance entre @ logique globale (@g) et l'@ physique MAC (@P)

# Commandes utilisateurs

- La commande `ip r` permet de connaître la table de routage d'une machine
- La commande `traceroute` permet de connaître les routeurs vers une machine destination



```
thomas@trg33:~$ ip r
default via 134.206.24.1 dev eth0
134.206.24.0/22 dev eth0 proto kernel scope link src 134.206.25.45
169.254.0.0/16 dev eth0 scope link metric 1000
172.16.1.0/24 dev vmnet1 proto kernel scope link src 172.16.1.1
172.16.238.0/24 dev vmnet8 proto kernel scope link src 172.16.238.1
```



# Internet Protocol

# Internet Protocol

- Un espace d'adressage unique (au dessus de l'adressage physique des réseaux)
- Un service pour acheminer « au mieux » les paquets
- Un service « sans connexion »
- Un mécanisme simple de conversion de nom à adresse (DNS)
- Une opération de routage à chaque sortie de réseau
- Un mécanisme de résolution d'adresse IP en adresse physique (MAC)

# En-tête IPv4

0	4	8	16	19	24	31
version	Lg-en	Type de service	Lg totale			
identification			Flags	Dep Fragment		
ttl		protocole	Somme de contrôle			
@ ip source						
@ ip destination						
options					remplissage	

# Internet Protocol

- La montée en puissance de la micro-informatique : les stations supporteront des logiciels de communication complexes (TCP)
- Donc le réseau peut fournir un service plus simple et moins contraignant
  - Plus besoin d'établir de circuit
  - Des paquets pourront manquer ou arriver dans le désordre

# La fragmentation d'IP ou la souplesse d'un protocole

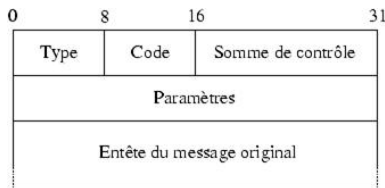
- Pour s'adapter à tout type de réseaux :
  - Le datagramme peut être « fragmenté » pour ne pas excéder la taille limite des éléments échangés dans le réseau (e.g. X25, PPP)
- À son arrivée dans la station destinataire les fragments sont rassemblés pour reconstituer le datagramme
  - Un équipement de réseau ne perd pas de temps à réassembler un datagramme
  - Les fragments ont pu prendre des chemins différents



# Internet Control Message Protocol

# ICMP : Protocole de contrôle

- ICMP : Internet Control Message Protocol
- Encapsulé dans IP
- Véhicule les messages d'erreurs
- Format :



# ICMP : Types de message

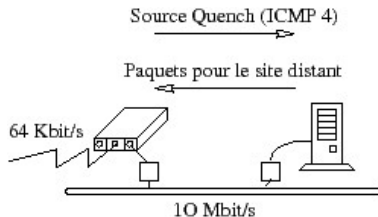
Type	Description
0	Réponse d'écho
3	Destination inaccessible
4	Limitation du débit de la source
5	Modification de route
8	Demande d'écho
11	Datagramme trop vieux
12	Problème de paramètre
13	Demande d'estampille de temps
14	Réponse d'estampille de temps
17	Demande de masque
18	Réponse de masque

# Code possible pour le type 3

Code	Description
0	Réseau inaccessible
1	Machine inaccessible
2	Protocole inaccessible
3	Port inaccessible
4	Fragmentation impossible
5	Route impossible à suivre

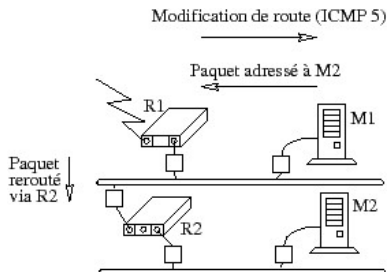
# Résolution de congestion

- Un routeur peut être submergé
- Émission du paquet Source Quench
- La source réduit son débit



# Redirection de paquets

- Origine : une erreur de routage
- Le fauteur est averti de son erreur
- Le paquet est réorienté

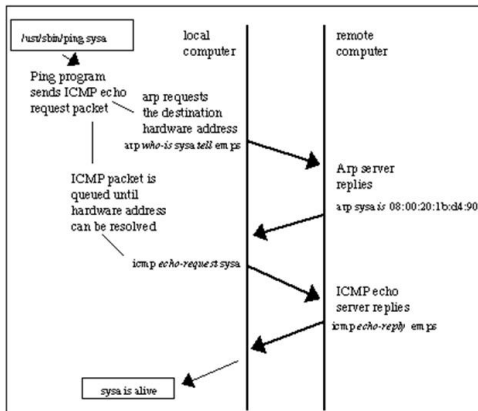


# Commande ping

Utilitaire ping : Tester la connexion de bout en bout entre deux machines

- Envoi d'un paquet ICMP de type **8** (Echo request)
- Réception d'un paquet ICMP de type **0** (Echo reply)
- + calcul d'un délai de propagation
- + émission de plusieurs Echo Request

# Fonctionnement





# Commande traceroute

Utilitaire traceroute : Connaître tous les routeurs entre deux machines

# Passerelles Applicatives

# Différentes adresses

Il existe des adresses IP :

- publiques
- privées (non acheminées sur les réseaux publics)

Classe d'@	adresses réseaux
A	10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
B	172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
C	192.168.0.0 - 192.168.255.255 (192.168.0.0/16)

# Traitement des adresses privées

2 possibilités :

- Le service d'acheminement effectue une substitution de l'adresse privée et lui affecte temporairement une adresse publique (service NAT)
- Le service d'acheminement détruit les datagrammes comportant une adresse privée

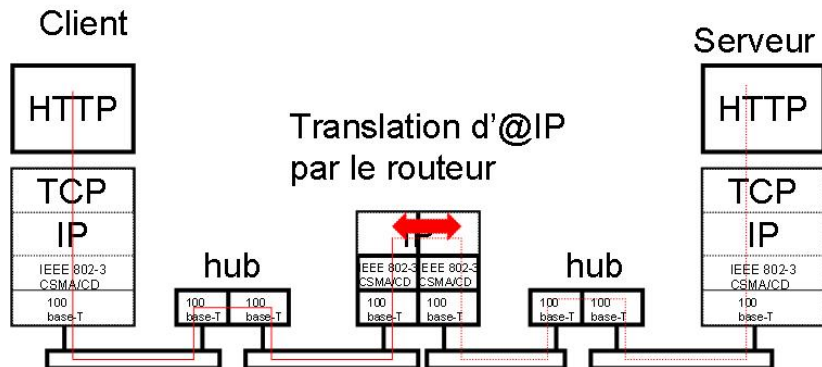
# Intérêt des adresses privées

- Les adresses privées peuvent donc être réemployées dans plusieurs réseaux privés
- Les adresses privées permettent de mettre en place un premier niveau de sécurité
  - filtrage du trafic de et/ou vers ces stations

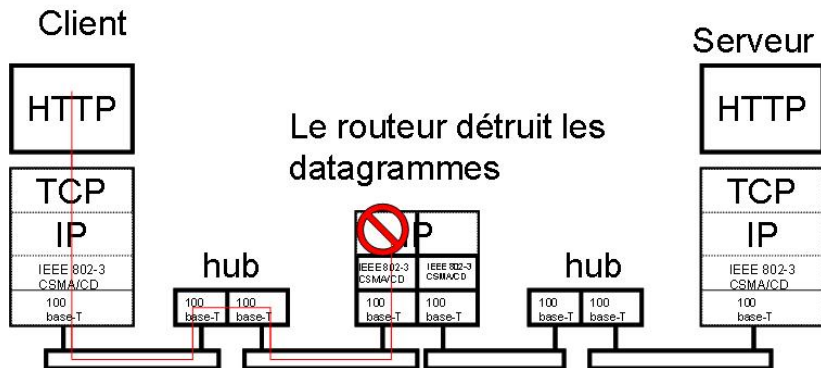
# Des passerelles d'applications

- Pour accéder à un service en dehors du réseau privé, ces stations devront contacter des passerelles applicatives
- Ces machines qui ont un accès complet au service d'acheminement effectueront les requêtes pour le compte des stations
- ex : proxy/cache web

# Contrôle du trafic avec des adresses privées

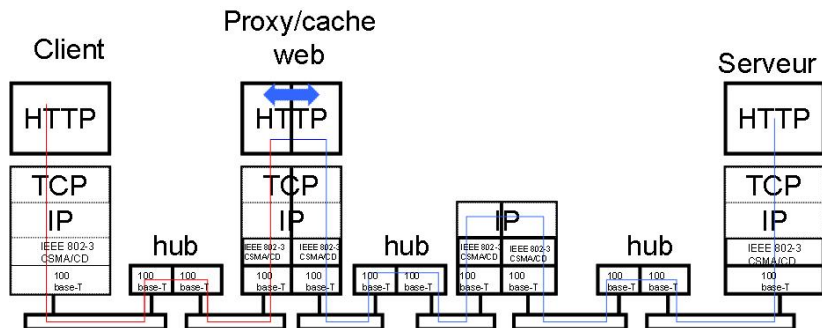


# Blocage du trafic IP par des adresses privées





# Acheminement du trafic web avec des adresses privées

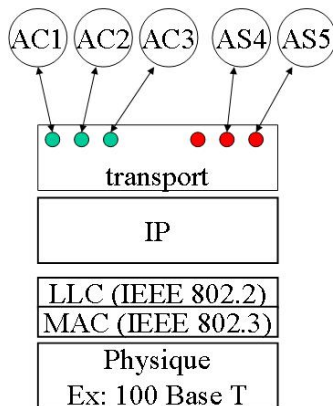


# La couche transport

# La couche transport en bref

- Service utilisé par la couche transport
  - Une couche réseau capable **d'acheminer** un datagramme d'une machine à une autre au travers d'un réseau complexe (ex : internet)
- Service minimal offert par la couche transport (offert par UDP)
  - aiguiller les datagrammes vers plusieurs applications sources et destinations résidentes sur la même machine -> notion de **multiplexage** (concept de numéro de port)
- Service plus complexe (offert par TCP)
  - Assurer un transfert fiable des informations entre applications
    - Quelque soit le nombre d'erreurs introduites par le réseau
    - Quelque soit la puissance des machines hébergeant les applications (régulation de flux)
  - Éviter de monopoliser l'usage du réseau (utilisation équitable de ce dernier)

# Multiplexage de la couche transport : concept de port



- Ports identifiés (connus à l'avance)
- Ports sélectionnés par l'OS (parmi les ports libres de la machine)

**Asx** : Application Serveur (ce type d'application se met en écoute sur un port connu comme par exemple le port 80 quand il s'agit d'une application web)

**Acx** : Application Cliente qui se voit attribuer un port au hasard parmi les ports libres gérés par la couche transport)

# En résumé

- La couche transport offre deux types de protocoles :
  - Service minimal (UDP)
  - Service à haute fiabilité (TCP)
- Pour aller d'une application à une autre, un datagramme a besoin de l'adresse de l'application source et de l'adresse de l'application destination

# Adressage des applications

- La source et la destination sont identifiées par :
  - adresse IP de la machine hébergeant l'application
  - type de protocole transport, UDP ou TCP
  - numéro de port

# Transport

- La couche transport ajoute un entête à chaque envoi de datagramme
- Elle y précise (notamment) :
  - Le type de protocole
  - Le numéro de port
- Cet entête est beaucoup plus complexe en mode TCP qu'en mode UDP

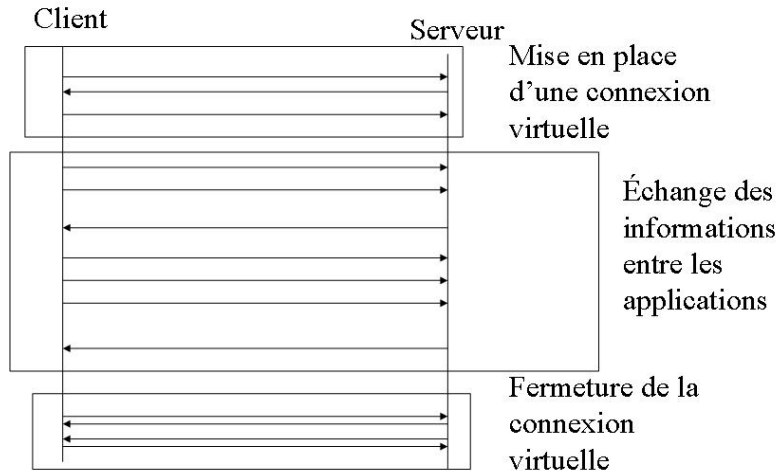
# Transport TCP

Pour assurer la fiabilité de la communication, TCP :

- Met en place une connexion virtuelle entre les applications avant d'échanger les données des applications
- Indique explicitement à l'émetteur que les données sont reçues
- Indique une capacité de réception pour ralentir l'émetteur (régulation de flux)
- La connexion virtuelle est fermée à la fin de l'échange entre les applications



# Les trois phases de la communication avec TCP



# UDP versus TCP

- UDP simple et plus rapide mais n'apporte pas la fiabilité à l'application
- TCP complexe, charge le réseau mais assure un échange fiable
- UDP sera utilisé pour les échanges peu volumineux ou pour des données de typ audio ou vidéo en temps réel
- TCP sera utilisé lors des échanges volumineux et où la fiabilité est primordiale (ex : transfert de fichier)

# Analogie avec la distribution du courrier

- La couche **transport** est un **secrétaire** particulier attaché à une application
- Ce secrétaire obéit à un **mode de fonctionnement** (UDP ou TCP) et porte un **numéro unique** au sein d'un immeuble (la machine)
- Cet immeuble possède une **adresse** : équivalent à l'**adresse IP**

# Table des analogies

Lettre	Informations
Immeuble	Machine
Adresse Immeuble	Adresse IP
Secrétaire	Couche transport
Mode de fonctionnement	UDP ou TCP
Numéro	Numéro de port
Enveloppe transport	Entête UDP ou TCP
Service postal	Couche réseau
Enveloppe réseau	Entête IP

# Analogie avec la distribution du courrier

La couche réseau c'est le **service postal** utilisé par le secrétaire qui prend l'enveloppe transport et la glisse dans une enveloppe réseau normalisé pour le service postal comme une **lettre** enfermée dans une **enveloppe** portant l'adresse de l'immeuble, le numéro et le type de secrétaire du destinataire. Au dos de l'enveloppe, il indique sa propre adresse d'immeuble, son numéro et son type.

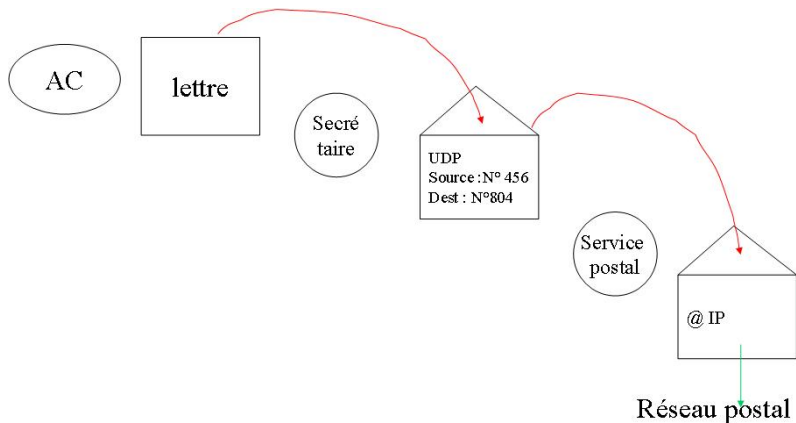
# Limitation du service postal

- Le service postal limite la taille de la lettre et oblige le secrétaire "lorsqu'il émet" à découper l'information de l'application en un grand nombre de lettres (taille environ 1500 octets)
- Le service postal va au plus vite pour remettre la lettre au destinataire : il fait au mieux. En cas de trafic important, les lettres arrivent plus lentement
- Parfois une lettre peut se perdre ou ne jamais arriver. Dans certains cas, elle peut aussi arriver en double !

# Secrétaire "simple" (mode de fonctionnement UDP)

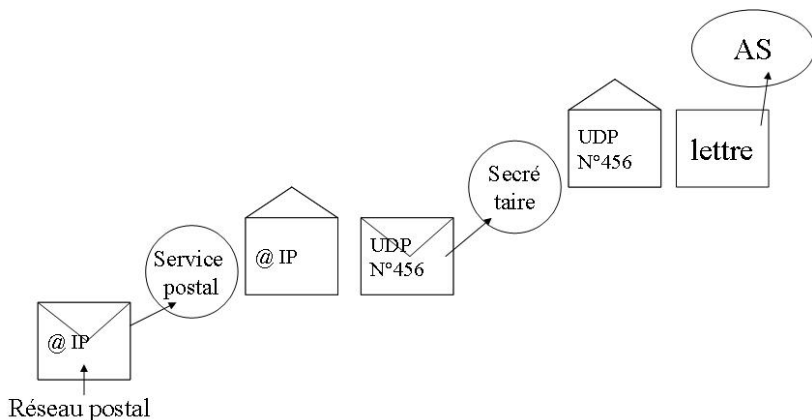
- Ce type de secrétaire remet les lettres dès qu'elles arrivent et ne vérifie pas si une lettre s'est égarée
- Il envoie aussi les lettres dès que l'application lui remet l'information
- Conclusion : UDP propose peu de valeur ajoutée par rapport à IP mais ne ralentit pas les applications en exploitant au mieux les capacités du réseau

# Principe de l'émission UDP





# Principe de la réception UDP



# Et avec TCP

- Pour TCP, le principe global des échanges d'enveloppe reste le même
- Ce qui change :
  - Les informations à remplir sur l'enveloppe de transport sont plus nombreuses
  - Avant de commencer un envoi de lettre, elles mettent en place une connexion "virtuelle" en s'envoyant des enveloppes transport vides
  - Les deux secrétaires peuvent s'échanger régulièrement des enveloppes de transport vides (sans lettre) pour indiquer où elles en sont dans leur travail de réception
  - Quand l'application leur indique que l'échange est terminée, elles ferment la connexion virtuelle en s'échangeant des enveloppes vides

# Types de message transport

- Demandes (sans données)
  - D'ouverture de connexion à partir d'un index de départ  $dc$  avec une capacité de réception  $w$
  - De fermeture de connexion
- Acceptation (sans données)
  - D'ouverture de connexion à partir de l'index de départ  $ds$  avec une capacité de réception  $w$
  - De fermeture de connexion
- Envoi de données avec indications :
  - Acquittement de données jusqu'à l'index  $r$
  - Capacité de réception  $w$
- Indications uniquement (sans données) :
  - Acquittement de données jusqu'à l'index  $r$
  - Capacité de réception  $w$

# Conclusion TCP

- Le contenu de la lettre n'est jamais examiné par la couche transport
- Ce qui implique que les informations sont une suite d'octets non structurés (un flot), fiabilisé, sans perte, sans duplications ou erreurs
- Cependant nous verrons que ce flot est lui aussi structuré par les applications
- Cette structure échappe à TCP

# Le système de noms de domaines (DNS)

# Présentation

- Une machine est identifiée grâce à son adresse IP.
- Problème : aucun humain ne peut réellement se souvenir de beaucoup de suite de 4 nombres (c'est pire en IPv6 !)
- Solution : donner un nom aux machines
- DNS : Domain Name System
- Objectif : trouver l'adresse IP d'une machine à partir de son nom
- Au départ et jusqu'en 1984, une méthode de résolution de nom : le fichier `/etc/hosts`
- Problème de maintenance, de recherche, de mise à jour
- À partir de 1984, mise en place d'un nouveau service : le DNS
- Il est décentralisé :
  - On administre les ressources locales uniquement
  - Mais les informations sont accessibles à tout l'Internet

# Présentation

Le service d'annuaire permet :

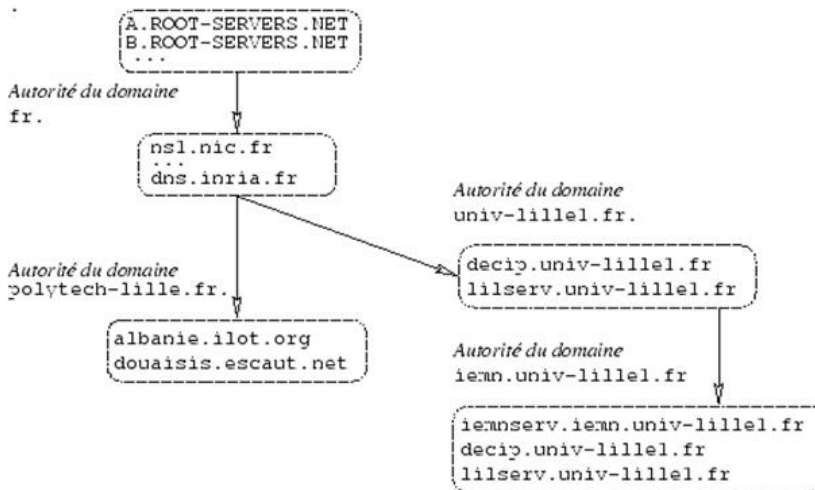
- de connaître le nom de la machine serveur du domaine et celle du serveur du domaine qui lui a délégué l'autorité
- la translation des noms de machines en @IP et l'inverse
- d'extraire des renseignements officiels sur un domaine administratif
  - noms et coordonnées des gestionnaires (personnes)
  - machines de courrier

# Organisation

- niveaux de domaines
  - root "."
  - niveau haut (net., edu., fr., com., ...)
  - niveau secondaire (univ-lille1., cnrs., ...)
  - ...
- À chaque nœud
  - nom (univ-lille1.)
  - et des ressources (machine, services)
- notion de sous-domaines (etudiant.univ-lille1.fr) avec autorité déléguée
- Un organisme assure l'unicité des noms de domaine
  - UREC pour le CNRS
  - AFNIC pour fr (délégation de RIPE-NCC au niveau européen)
- l'autorité suprême : Network Information Center



Autorité du domaine



# Noms de domaines

.com	Organisations commerciales
.edu	Le monde de l'éducation
.gov	Organisation gouvernementales
.mil	Organisation militaire
.org	Les autres organisations
.int	Organisations internationales

# Résumé

# Outils

- ping : demande d'écho à une machine distante
- traceroute : voir la route (les routeurs) entre deux machines
- wireshark : récupère et analyse les trames réseau

# Commandes

- `ip` : configuration réseau de la machine, adressage et routage
- `ssh` : connexion à distance à une autre machine