Cours 3

Couche Réseau (IP)

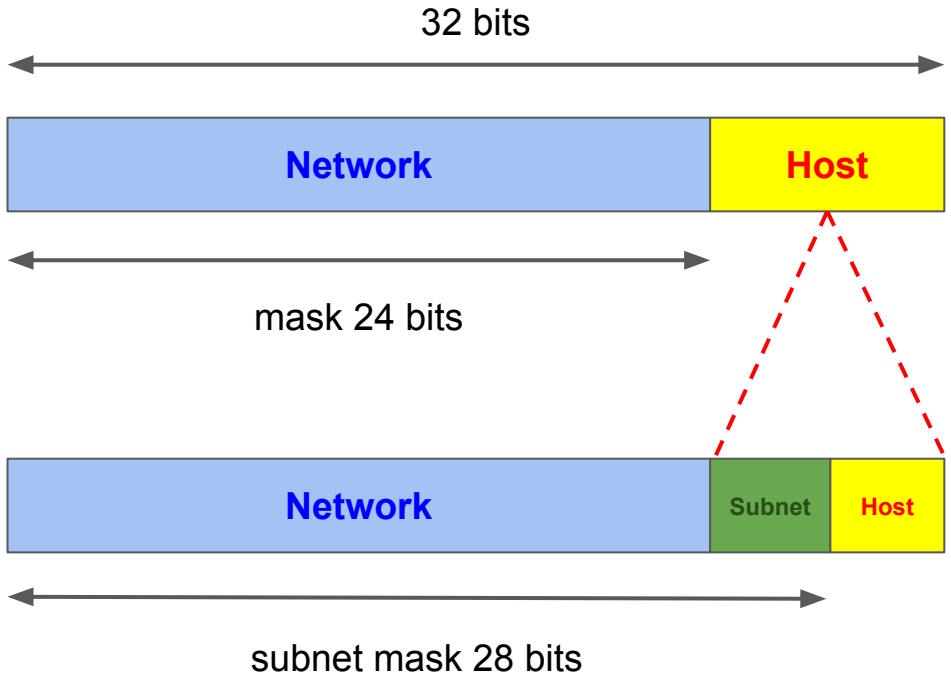
~

Sous-Réseaux, Routage

Sous-Réseaux

Découpage d'un réseau en plusieurs sous-réseaux

- Adresse IP découpée en trois parties : *network*, *subnet*, *host*
- Une partie des bits de *host* sert à identifier le sous-réseau (*subnet*)
- Masque de sous-réseau



RFC : https://tools.ietf.org/html/rfc1860 et https://tools.ietf.org/html/rfc1878

Sous-Réseaux

Exercice : Dans un réseau 193.51.199.0/24, on souhaite constituer 5 sous-réseaux (de même taille).

- Combien de bits sont nécessaires pour coder ces sous-réseaux ?
- Combien de machines trouve-t-on dans chaque sous réseau ?
- Quel est le masque de réseau et de sous-réseau ?
- A quel adresse de sous-réseau appartient la machine 193.51.199.67 ?
- Donner l'adresse de diffusion correspondant à ce sous-réseau ?
- Quel sont les adresses des autres sous-réseaux ?

```
orel@prout:~$ ipcalc 193.51.199.0/24
Address:   193.51.199.0          11000001.00110011.11000111. 00000000
Netmask:   255.255.255.0 = 24    11111111.11111111.11111111. 00000000
Wildcard:  0.0.0.255             00000000.00000000.00000000. 11111111
=>
Network:   193.51.199.0/24       11000001.00110011.11000111. 00000000
HostMin:   193.51.199.1          11000001.00110011.11000111. 00000001
HostMax:   193.51.199.254        11000001.00110011.11000111. 11111110
Broadcast: 193.51.199.255        11000001.00110011.11000111. 11111111
Hosts/Net: 254                    Class C
```

Sous-Réseaux

Correction

- Avec 3 bits, on peut coder un maximum de 2^3=8 sous-réseaux. En effet, 2 bits ne sont pas suffisants pour coder 5 sous-réseaux à choisir parmi 000, 001, 010, 011, 100, 101, 110 et 111.
- Il reste 5 bits pour la partie *host* ; le nombre de machines = 2^5-2=30. On retire les adresses min & max, qui sont réservés…
- Le masque du réseau /24 correspond à l'adresse 255.255.255.0.
- Le masque du sous-réseau est 255.255.255.224 car 224 = (1110 0000) en binaire
- Adresse du réseau : 193.51.199.67 & 255.255.255.0 = 193.51.199.0 (avec & l'opérateur "ET" binaire)
- Adresse du sous-réseau : 193.51.199.67 & 255.255.255.224 = 193.51.199.X avec X = 67 & 224 = 64 (en binaire : 010 00011 & 111 00000 = 010 0000) ; adresse sous-réseau = 193.51.199.64/27
- Adresse de diffusion du sous-réseau : 193.51.199.Y avec Y = 010 11111 = 95

Sous-Réseaux

Correction (suite)

- Adresses de sous-réseaux :

```
193.51.199.0/27
193.51.199.32/27
193.51.199.64/27
193.51.199.96/27
193.51.199.128/27
193.51.199.160/27 (unused)
193.51.199.192/27 (unused)
193.51.199.224/27 (unused)
```

- Pour calculer 5 sous-réseaux de taille 30 machines :

```
$ ipcalc 193.51.199.0/24 -s 30 30 30 30 30
```

```
1. Requested size: 30 hosts
Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
Network:   193.51.199.0/27      11000001.00110011.11000111.000 00000
HostMin:   193.51.199.1         11000001.00110011.11000111.000 00001
HostMax:   193.51.199.30        11000001.00110011.11000111.000 11110
Broadcast: 193.51.199.31        11000001.00110011.11000111.000 11111
Hosts/Net: 30                    Class C

2. Requested size: 30 hosts
Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
Network:   193.51.199.32/27     11000001.00110011.11000111.001 00000
HostMin:   193.51.199.33        11000001.00110011.11000111.001 00001
HostMax:   193.51.199.62        11000001.00110011.11000111.001 11110
Broadcast: 193.51.199.63        11000001.00110011.11000111.001 11111
Hosts/Net: 30                    Class C

3. Requested size: 30 hosts
Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
Network:   193.51.199.64/27     11000001.00110011.11000111.010 00000
HostMin:   193.51.199.65        11000001.00110011.11000111.010 00001
HostMax:   193.51.199.94        11000001.00110011.11000111.010 11110
Broadcast: 193.51.199.95        11000001.00110011.11000111.010 11111
Hosts/Net: 30                    Class C

4. Requested size: 30 hosts
Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
Network:   193.51.199.96/27     11000001.00110011.11000111.011 00000
HostMin:   193.51.199.97        11000001.00110011.11000111.011 00001
HostMax:   193.51.199.126       11000001.00110011.11000111.011 11110
Broadcast: 193.51.199.127       11000001.00110011.11000111.011 11111
Hosts/Net: 30                    Class C

5. Requested size: 30 hosts
Netmask:   255.255.255.224 = 27 11111111.11111111.11111111.111 00000
Network:   193.51.199.128/27    11000001.00110011.11000111.100 00000
HostMin:   193.51.199.129       11000001.00110011.11000111.100 00001
HostMax:   193.51.199.158       11000001.00110011.11000111.100 11110
Broadcast: 193.51.199.159       11000001.00110011.11000111.100 11111
Hosts/Net: 30                    Class C
```

Sous-Réseaux de Tailles Variables

VLSM (Variable-Length Subnet Masking)

- Possibilité de découper un réseau en sous-réseaux dont la taille n'est pas identique
- Meilleure utilisation des adresses disponibles !
- Nécessité de toujours associer une IP à son masque…

Exercice : Considérons le réseau 193.51.199.0/24.

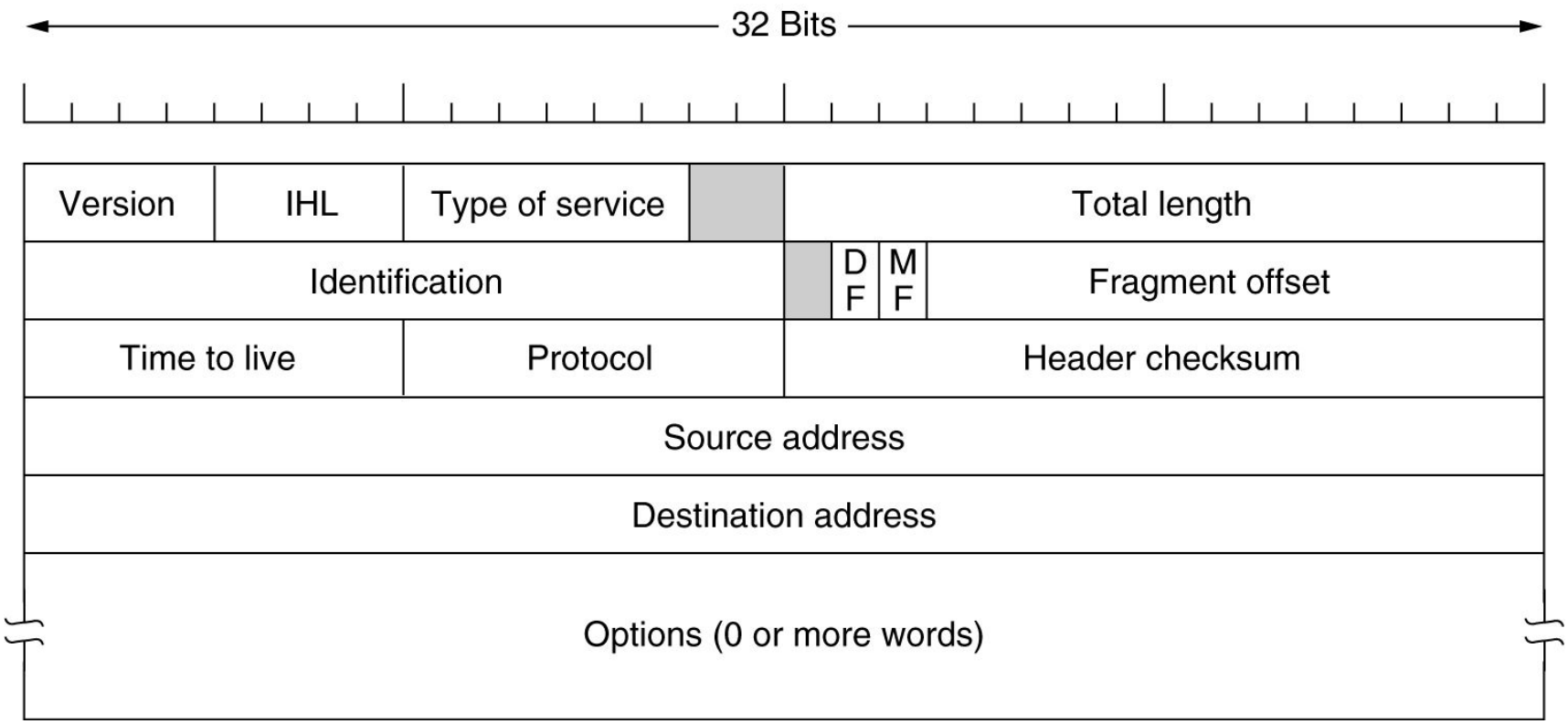
- Cas 1 : On souhaite constituer un sous-réseau avec 100 machines et un autre avec 60 machines.
- Cas 2 : On souhaite constituer un sous-réseaux avec 150 machines et un autre avec 50 machines.

Sous-Réseaux de Tailles Variables

Correction

- Cas 1 : Premier sous-réseau de taille 128 (193.51.199.0/25) et deuxième sous-réseau à la suite de taille 64 (193.51.199.128/26). Il reste un dernier sous-réseau de taille 64 (193.51.199.192/26) non utilisé
- Cas 2 : Ce n'est pas possible car le réseau est trop petit ! Il faut un sous-réseau de taille 256 (193.51.199.0/24) juste pour contenir les 150 machines du premier sous-réseau… Alors même que 150+50 < 256 !

En-Tête du Paquet IPv4



En-Tête du Paquet IPv4

- **Version** : 4 [4 bits]
- **Internet Header Length** : longueur de l'en-tête en mot de 32 bits [4 bits]
- **Type of Service** (ToS) : qualité de service (fiabilité, débit, …) [8 bits]
- **Identification** : identifiant d'un fragment pour leur rassemblage [16 bits]
- **Flags** : DF (Don't Fragment) / MF (More Fragment) [3 bits]
- **Fragment Offset** : position dans le paquet initial en mot de 8 octets [13 bits]
- **Time To Live** (TTL) : temps de vie maximal en *hop* [8 bits]
- **Protocol** : protocole encapsulé dans le paquet (ICMP, UDP, TCP, ...) [8 bits]
- **Header Checksum** : contrôle d'erreurs de l'en-tête [16 bits]
- **Source Address** [32 bits]
- **Destination Address** [32 bits]
- **Options** : usage peu fréquent...

Exercice : Quelle est la taille minimale & maximale de l'en-tête ?

*IHL sur 4 bits compris entre 5 (sans options) et 15 et donc l'en-tête a une taille entre 5*4=20 octets et 15*4=60 octets.*

Outils

ping : tester si machine est en vie (requête *echo* du protocole **IP/ICMP**)

```
$ ping 10.0.204.4
PING 10.0.204.4 (10.0.204.4) 56(84) bytes of data.
64 bytes from 10.0.204.4: icmp_seq=1 ttl=63 time=0.202 ms
64 bytes from 10.0.204.4: icmp_seq=2 ttl=63 time=0.206 ms
64 bytes from 10.0.204.4: icmp_seq=3 ttl=63 time=0.200 ms
64 bytes from 10.0.204.4: icmp_seq=4 ttl=63 time=0.213 ms
^C
--- 10.0.204.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 57ms
rtt min/avg/max/mdev = 0.200/0.205/0.213/0.011 ms
```

Le RTT (Round-Trip Time) mesure le temps d'aller-retour du paquet, qui est une approximation de la *latence x 2*.

ICMP

Internet Control Message Protocol (ICMP), RFC 792

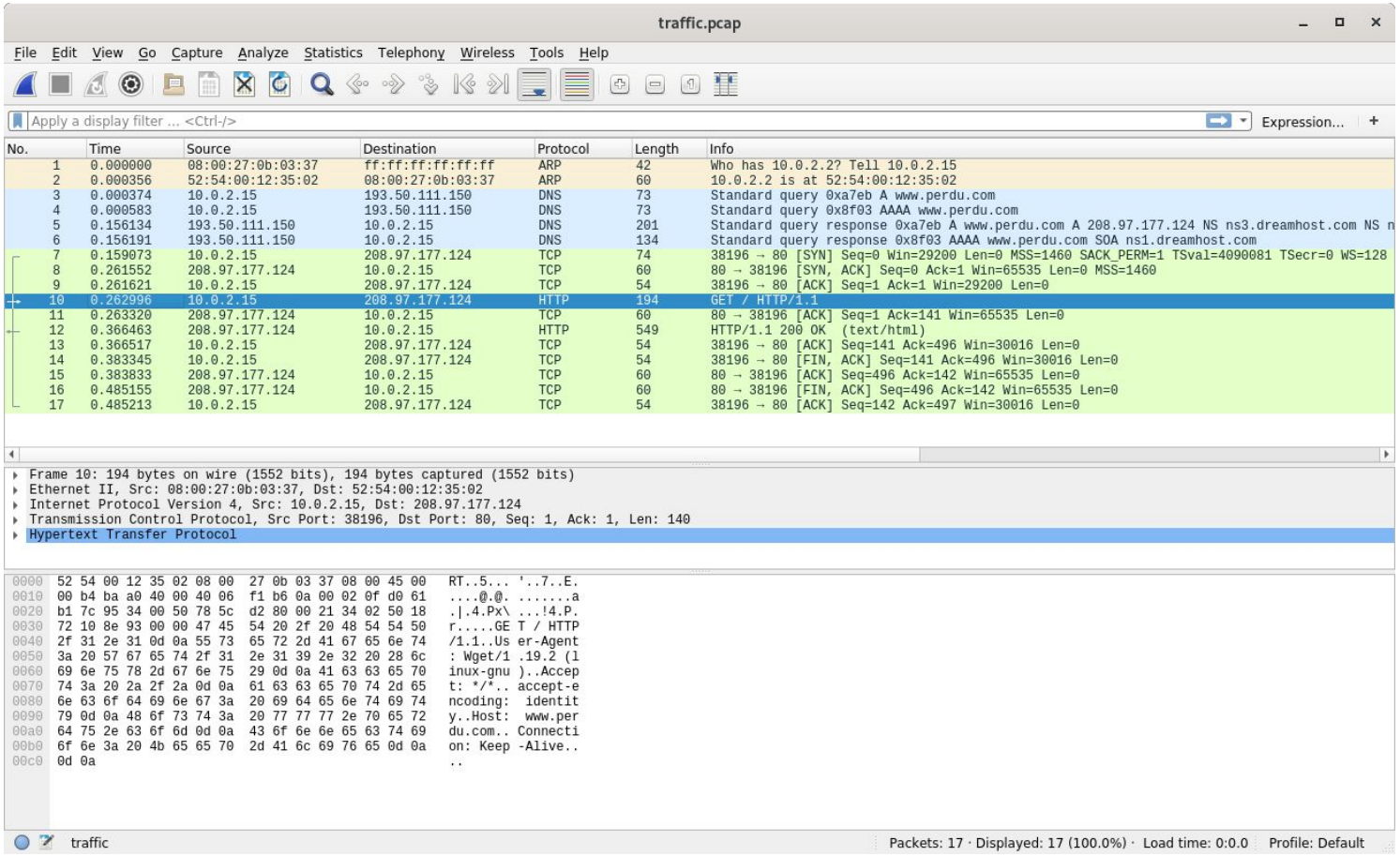
Accompagne IP pour gérer les erreurs et propager des informations de routage...

Message type	Description
Destination unreachable	Packet could not be delivered
Time exceeded	Time to live field hit 0
Parameter problem	Invalid header field
Source quench	Choke packet
Redirect	Teach a router about geography
Echo request	Ask a machine if it is alive
Echo reply	Yes, I am alive
Timestamp request	Same as Echo request, but with timestamp
Timestamp reply	Same as Echo reply, but with timestamp

Exemple du ping : envoi d'une requête ICMP *“echo request”* et attente de la réponse *“echo reply”*

Wireshark

Outil permettant l'analyse et la capture de trames réseau…

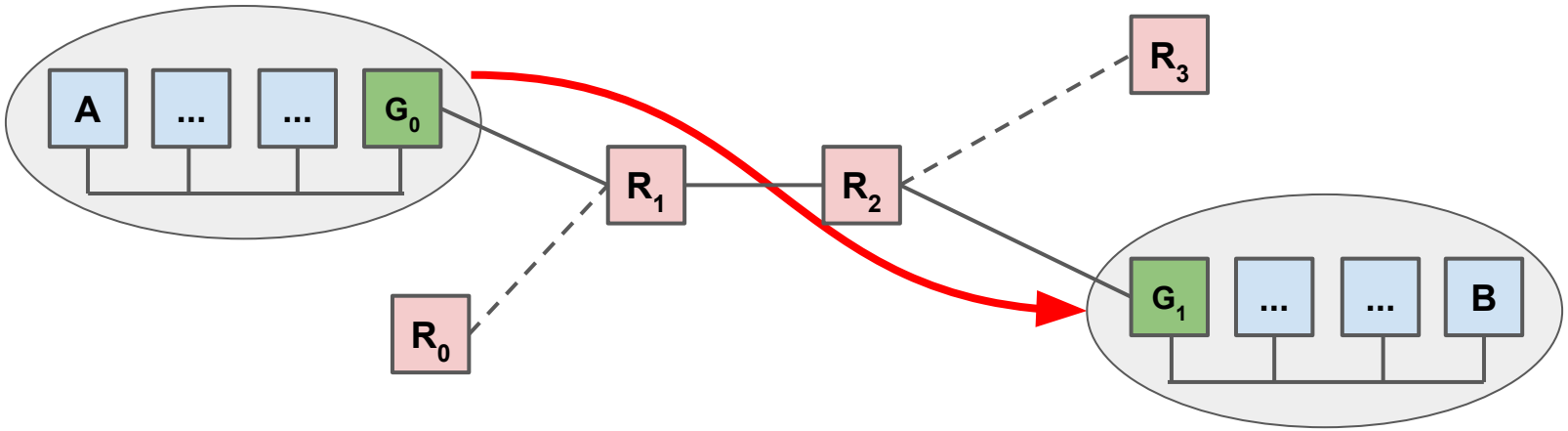


Démo : analyse d'un paquet IP/ICMP avec l'exemple ping.pcap

Routage

Principe : Mécanisme par lequel un paquet IP est acheminé d'un expéditeur (A) jusqu'à son destinataire (B), en s'appuyant sur les noeuds intermédiaires (Gi, Ri) du réseau Internet.

Les différents noeuds du réseau : les hôtes (A,B), les passerelles ou *gateway* (G_i) et les routeurs (R_i)



Routage statique & dynamique : manuel, DHCP, OSPF, BGP, ...

Table de Routage

Chaque nœud du réseau a besoin des informations sur le noeud suivant (next hop) vers lequel il doit envoyer un paquet pour atteindre la destination finale (dest addr)…

Exemple : table de routage d'un hôte

```
$ route -n
DestAddr        Gateway          GenMask          Flags    Interface
127.0.0.1       *                0.0.0.0          UH       lo
192.168.0.0     *                255.255.255.0    U        eth0
default         192.168.0.254    0.0.0.0          UG       eth0
```

Il faut distinguer :

- les **routes directes** vers un réseau (ou une machine) ;
- les **routes indirectes**, dont les **routes spécifiques** vers un réseau et la **route par défaut**.

Table de Routage

Un exemple plus compliqué :

```
$ route -n
DestAddr        Gateway         GenMask         Flags   Interface
127.0.0.1       *               0.0.0.0         UH      lo
147.210.10.0    *               255.255.255.0   U       eth1
147.210.0.0     *               255.255.255.0   U       eth0
10.0.0.0        147.210.0.100   255.255.0.0     UG      eth0
default         147.210.0.254   0.0.0.0         UG      eth0
```

Légende des Flags

- U : la route est active (Up)
- G : route indirecte qui passe par un routeur (Gateway) ; sinon route directe (pas G)
- H : l'adresse destination est une adresse de machine (Host) ; sinon l'adresse destination est celle d'un réseau (pas H)

Exercice

- Donner la signification de chaque ligne de la table de routage…
- En déduire la configuration réseau de cette machine…

Table de Routage

Correction

La machine a la configuration suivante :

- interface *eth0* avec une adresse dans le réseau 147.210.0.0/24
- interface *eth1* avec une adresse dans le réseau 147.210.10.0/24
- Il y a dans le réseau 147.210.0.0/24 un passerelle (.100) vers le réseau 10.0.0.0/16.
- La passerelle par défaut est 147.210.0.254… → porte de sortie vers Internet

```
$ route -n
DestAddr        Gateway          GenMask          Flags    Interface
127.0.0.1       *                0.0.0.0          UH       lo
147.210.10.0    *                255.255.255.0    U        eth1
147.210.0.0     *                255.255.255.0    U        eth0
10.0.0.0        147.210.0.100    255.255.0.0      UG       eth0
default         147.210.0.254    0.0.0.0          UG       eth0
```

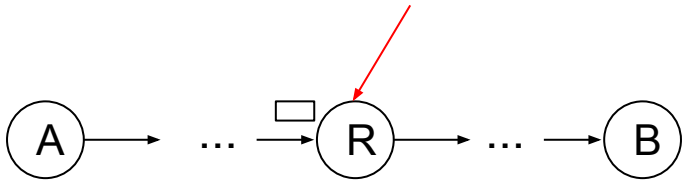
Algorithme de Routage

Algorithme exécuté sur chaque nœud intermédiaire (R)

Supposons que *DestFinal* est l'adresse de destination (B) du paquet à transmettre, *DestAddr* est une adresse dans la table de routage.

Pour chaque ligne dans la table de routage :

```
// host route
if (DestAddr = DestFinal)
     envoyer le paquet via la route directe ou indirecte vers le next hop
// net route
else if (DestAddr = DestFinal & GenMask)
     envoyer le paquet via la route directe ou indirecte vers le next hop
// default route
else
     envoyer au next hop de la route par défaut
```



Outils

traceroute : Outil qui permet de retrouver le chemin d'un paquet sur Internet.

Exemple : route vers le serveur Web du LaBRI

```
$ traceroute www.labri.fr

traceroute to www.labri.fr (147.210.8.59), 30 hops max, 60 byte packets

 1  _gateway (192.168.1.254) 0.931 ms
 2  p25.socabim.isdnet.net (194.149.169.41) 35.552 ms
 3  free-paris-por1.bb.ip-plus.net (193.5.122.58) 36.783 ms
 4  193.51.187.208 (193.51.187.208) 34.955 ms
 5  te2-2-lyon2-rtr-021.noc.renater.fr (193.51.177.43) 49.554
 6  te0-0-0-5-lyon1-rtr-001.noc.renater.fr (193.51.177.216) 50.311 ms
 7  te0-0-0-0-ren-nr-clermont-rtr-091.noc.renater.fr (193.51.177.227) 49.938 ms
 8  te0-0-0-0-ren-nr-bordeaux-rtr-091.noc.renater.fr (193.51.177.84) 51.531 ms
 9  reaumur-vl10-gi8-1-bordeaux-rtr-021.noc.renater.fr (193.51.183.37) 49.163 ms
10  147.210.246.205 (147.210.246.205) 48.959 ms
11  www3.labri.fr (147.210.8.59) 54.947 ms
```

La route est parfois longue ! On part de Talence (chez moi), on monte à Paris avec le *backbone* de Free, avant de revenir pas le réseau Renater par Lyon, Clermont, puis retour sur Talence !

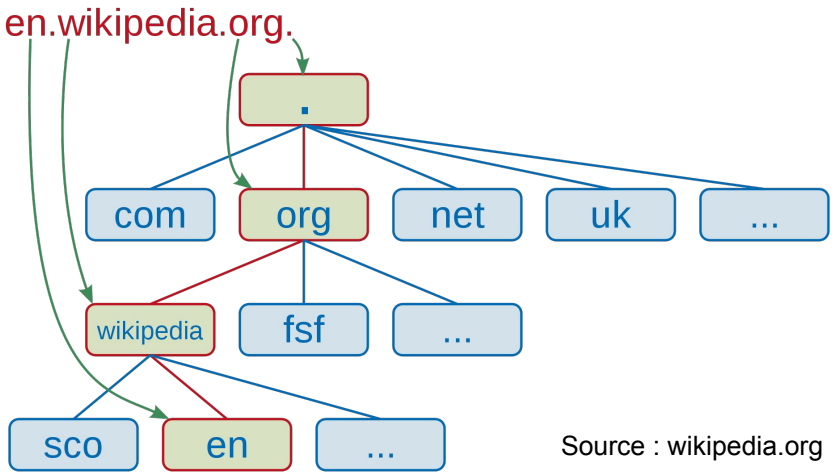
DNS

Problématique : Comment associer un nom lisible à une adresse IP numérique ?

Exemple : en.wikipedia.org ⇒ 91.198.174.192

Protocole DNS (Domain Name System)

- Protocole central dans Internet depuis 1985 (UDP, 53)
- Espace de noms hiérarchique gérés par une hiérarchie de serveurs
- Modèle client/serveur
  - Un client interroge un serveur de noms (serveur DNS) et attend la réponse (UDP, port 53)
  - Utilisé automatiquement par les applications, rarement directement par l'utilisateur



Source : wikipedia.org

DNS

nslookup & host : interroger le serveur **DNS** (Domain Name System) pour trouver l'adresse IP (IPv4 et/ou IPv6) associée à un nom de machine…

```
$ nslookup www.google.com

Server:         10.0.220.1                  <-- Server DNS local
Address:        10.0.220.13#53

Name:           www.google.com
Address:        172.217.18.228

Name:           www.google.com
Address:        2a00:1450:4006:809::2004


$ host www.google.com

www.google.com has address 172.217.19.228
www.google.com has IPv6 address 2a00:1450:4007:817::2004
```