R3.06 Architectures des réseaux

Jean-François Remm

IUT Nantes département Informatique

2024–2025

Objectif

Compétences ciblées :

- Installer, configurer, mettre à disposition et maintenir en conditions opérationnelles des infrastructures, des services et des réseaux et optimiser le système informatique d'une organisation

Savoir de références étudiés :

- Technologie des réseaux (piles, couches transport, TCP/IP/UDP, DHCP, DNS, ...)
- Interconnexions de réseaux (par ex : routage, NAT, filtrage, proxy)

Organisation du module

Volume horaire : 24H

- par semaine : 1H20 CM + 2H40 TD

Modalité d'évaluation :

- 1 test coeff. 1
- 1 note de TD coeff. 1

Équipe pédagogique

- CM : Jean-François Remm
- TD : Erwann Helleu, Saïd El Mamouni et Jean-François Remm

Remerciements – Contributeurs – Sources

- cours ASR (Administration Système–Réseau) de P. Levasseur
- travail collaboratif réalisé avec J.-F. Berdjugin et P.-A. Jacquot à de l'IUT1 Grenoble
- cours d'administration réseau de H. Pinvidic (M2 Alma)

1 Modèles

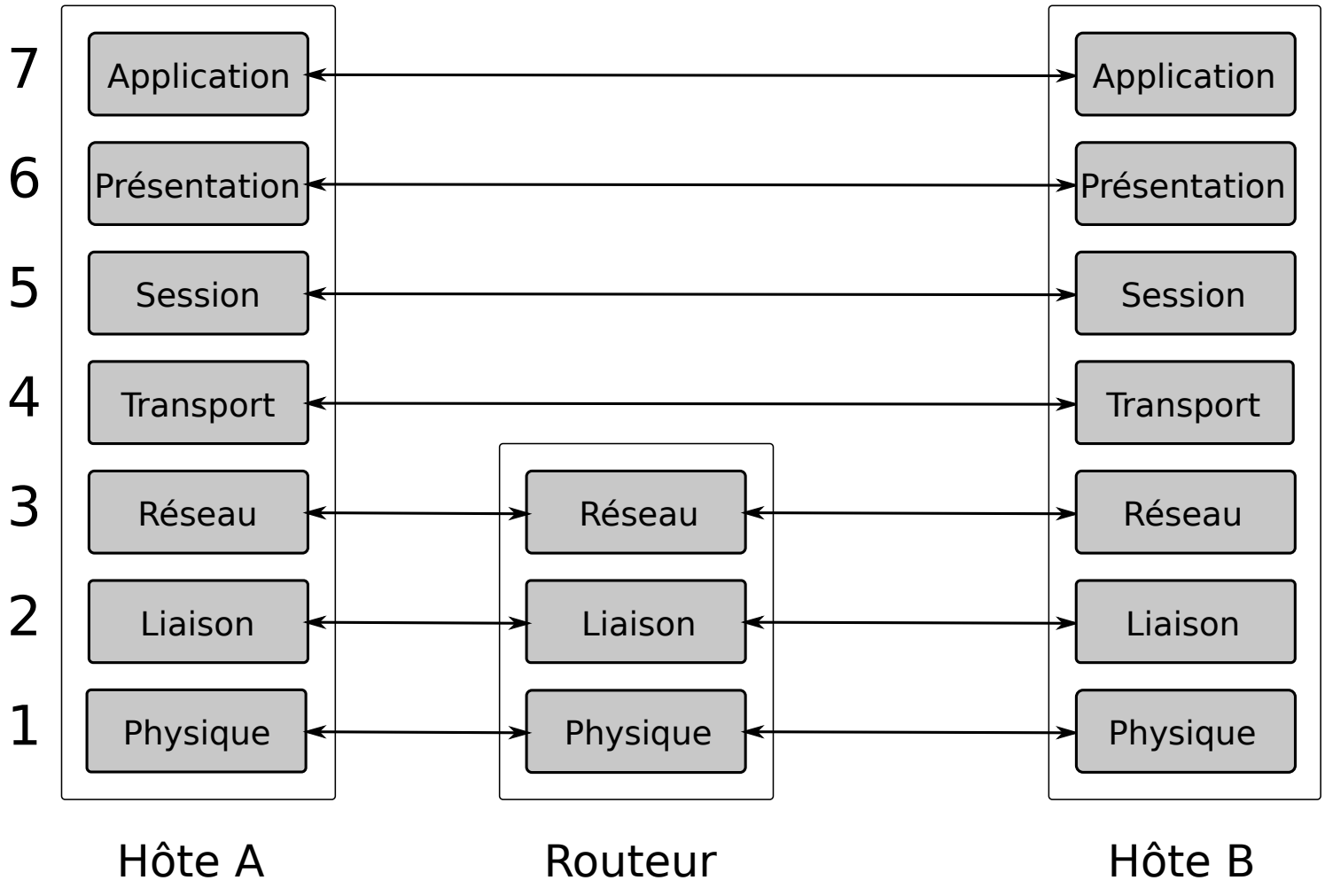
2 Couches basses

3 Couche réseau
- Configuration IP
- Routage

4 Couche transport

5 Services réseaux
- Web
- Firewall
- NAT
- DNS

Modèle OSI


7
6
5
4
3
2
1
Application
Présentation
Session
Transport
Réseau
Liaison
Physique
Application
Présentation
Session
Transport
Réseau
Liaison
Physique
Réseau
Liaison
Physique
Hôte A
Routeur
Hôte B

Couche 1 : physique

- fournit les moyens nécessaires aux transferts des élément binaires
  - câblage (coaxial, paire torsadées, fibres optique)
  - interfaçage de connexion (prise RJ45, connecteur BNC, ...)
  - codage des bits (niveau électrique)
  - équipement de transmission (modems, commutateurs, ...)
  - topologie du réseau
- spécifie **canal** et **signal** sans aucune sémantique de l'information

$\Longrightarrow$ niveau bit

Canal

- le canal est constitué de tout médium capable d'assurer le transfert d'une information binaire
- caractérisé par une **bande passante** (bande des fréquences utilisés) :

$$W = F_{max} - F_{Min}$$

- Exemples :

type	bande passante
paire torsadée	$>$ 100 kHz
cable coaxial	$>$ 100 MHz
fibre optique	$>$ 1 GHz
espace entre 2 antennes	variable

- les données binaires sont transportés par un **signal**

Débit

- les éléments transmis sur un réseaux : 0 ou 1
- le **débit** mesure la rapidité d'une communication numérique
- débit = nombre d'éléments binaires transmis par seconde
- unité : bits par seconde (bps)
- Exemple : D = 10 Mbps

Signal

- **Signal** : variation d'une grandeur physique, porteuse d'informations
- Exemple :
  - signal électrique : tension
  - signal optique : onde lumineuse
  - signal hertzien : onde électromagnétique

Transmission

- Deux types de transmissions :
  1. transmission en bande de base
  2. transmission d'un signal modulé

Transmission en bande de base

- suite de bits représentant les données numériques,
- changement d'états discret du signal physique,
- pas de transposition en fréquence,
- durée de chaque bit est constante

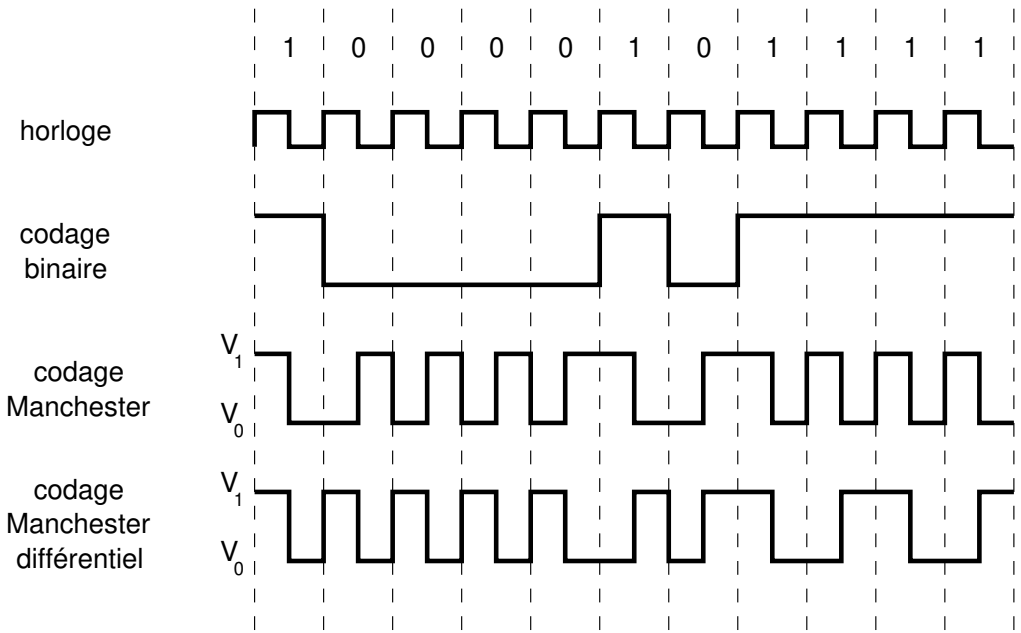


Figure – Transmission en bande de base

Transmission d'un signal modulé

- utilisation d'une onde porteuse
- modification pour augmenter le débit, diminuer le taux d'erreur

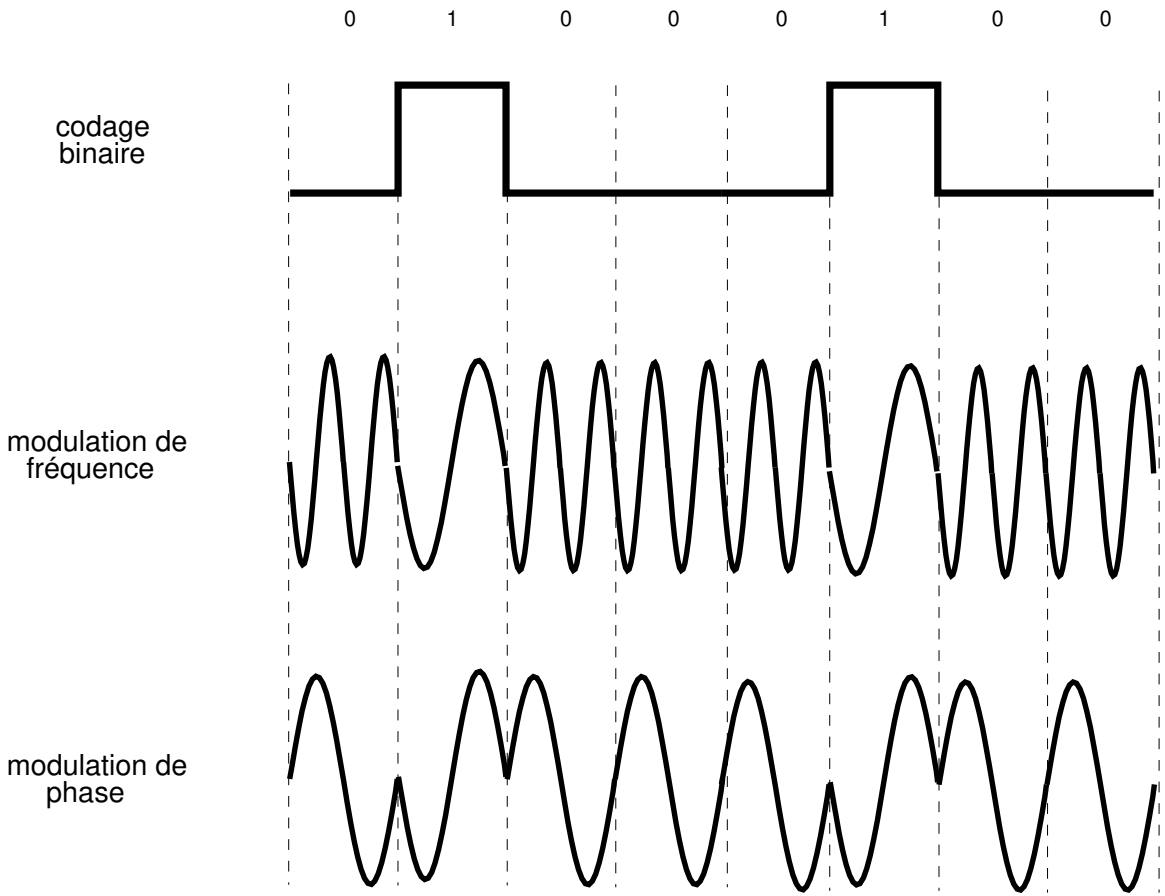


Figure – Transmission d'un signal modulé

Couche 2 : liaison de données

- transforme un flot binaire brut en **trames**
- gère l'établissement, le maintien et la libération de la liaison entre **terminaux**
  - transmission
  - contrôle de flux
  - contrôle d'erreur
  - adressage des terminaux
  - accusé de réception

$\Longrightarrow$ niveau trame

Couche 1bis : MAC *Medium Acces Control*

- sous-couche de contrôle d'accès au canal pour un réseau à diffusion
- mécanisme d'adressage des hôtes : adresses MAC
- protocole de gestion d'accès :
  1. CSMA/CD
  2. CSMA/CA

Couche 3 : réseau

- permet la connexion de réseaux entre systèmes ouverts :
  - fonction d'**adressage** des réseaux
  - fonction de **routage/relayage** pour l'acheminement d'un datagramme
- doit permettre l'interconnexion de réseaux hétérogènes

$\Longrightarrow$ niveau paquet

Couche 4 : transport

- reçoit des messages des couches supérieures et découpe ces messages en **segments** ou **datagramme** avant transmission aux couches inférieures
- optimise les ressources réseau :
  - contrôle de flux : ordonnancement, gestion des pertes
  - correction des erreurs

$\Longrightarrow$ niveau segment ou datagramme

Couche 5 : session

- fournit à la couche 6 des moyens de synchroniser le dialogue
- définit les séquences de l'échange :
  - échange bi- ou unidirectionnel
  - point de reprise, retour arrière

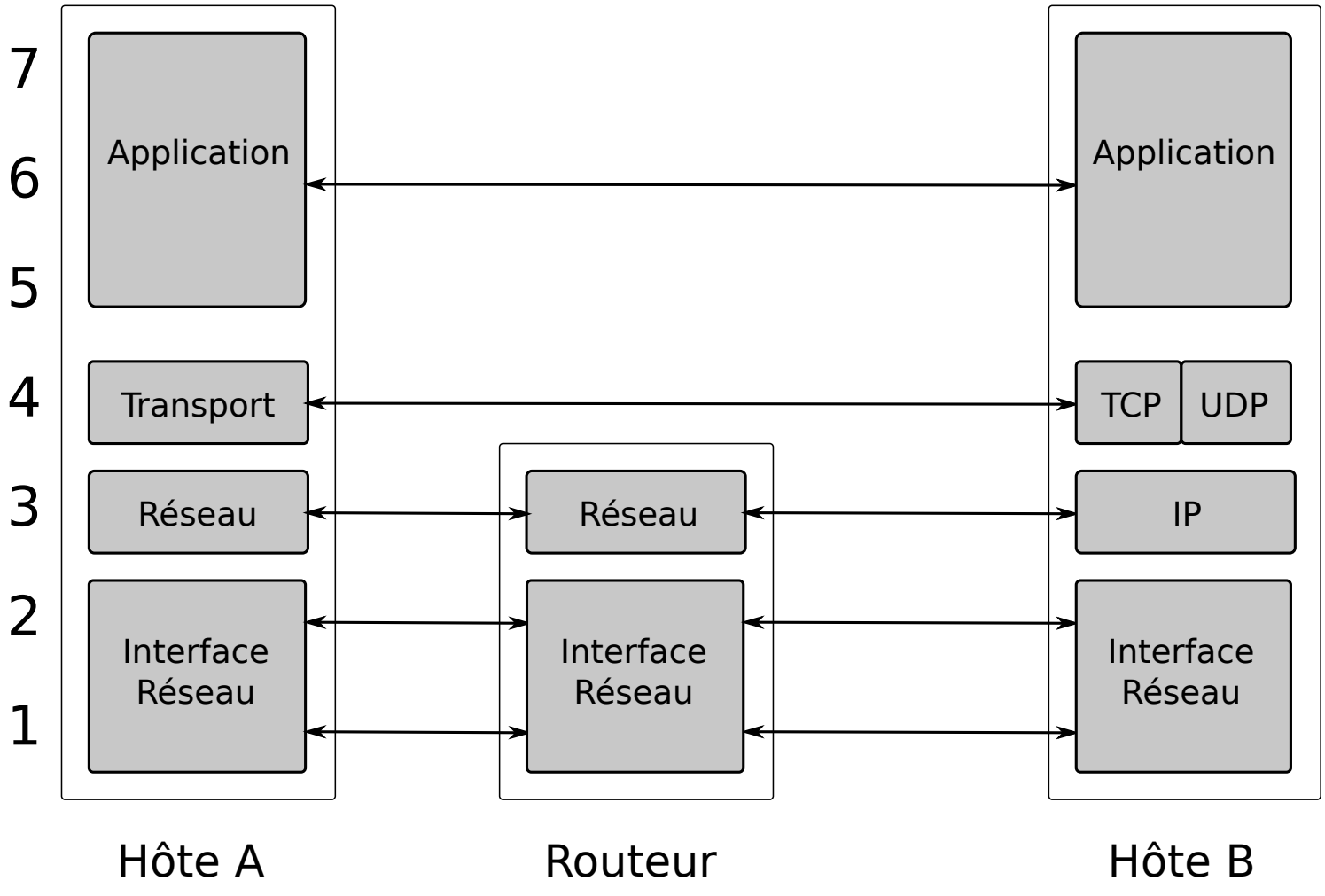
Couche 6 : présentation

- Définit syntaxe et sémantique des informations :
  - traduction (ex : ASN.1)
  - compression et décompression de données
  - chiffrement, déchiffrement

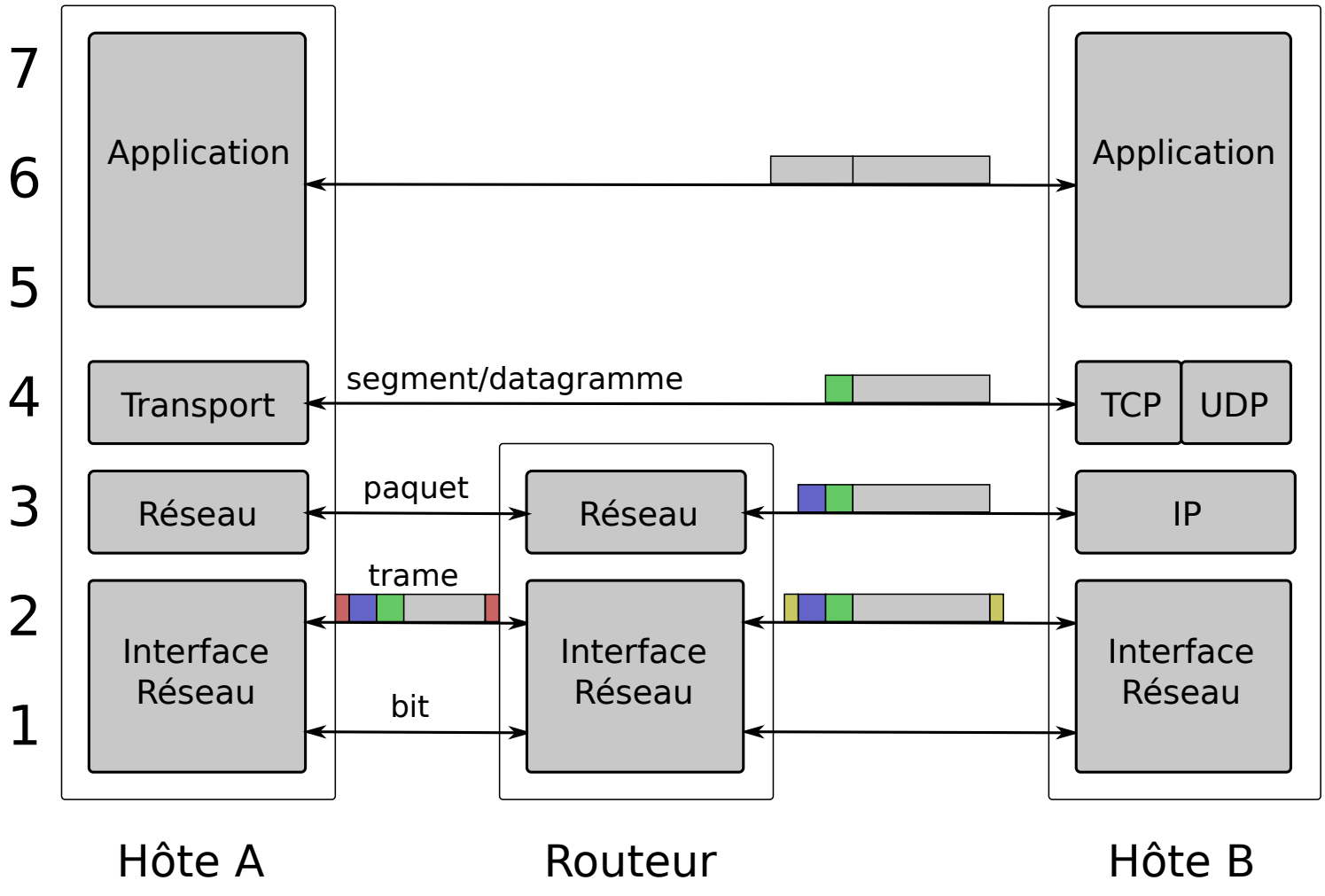
Couche 7 : application

- chargé de l'exécution de l'application
- applications "classiques"
  - mail
  - web
  - transfert de fichiers
  - groupes de discutions

Modèle TCP/IP



Encapsulation



1 Modèles

2 Couches basses

3 Couche réseau
- Configuration IP
- Routage

4 Couche transport

5 Services réseaux
- Web
- Firewall
- NAT
- DNS

Rappel

Éléments d'interconnexion :

- couche 1 : câbles, cartes réseaux, répéteur (*repeater*), concentrateur (*hub*)
- couche 2 : pont (*bridge*), commutateur (*switch*)
- couche 3 : routeur (*router*)
- couches supérieures : passerelles applicatives (*gateway*).

ARP

Association entre adresse réseau (IP) et adresse MAC (Éthernet)

- arp

```
#arp -s 192.168.1.1 ca:fe:00:ca:fe:00
#arp
Address         HWtype  HWaddress           Flags Mask  Iface
172.21.60.1     ether   00:e0:b1:a9:75:c0   C           eth0
192.168.1.1     ether   00:ca:fe:00:ca:fe   CM          eth0
```

- iproute2

```
#ip neighbor add 192.168.1.1 lladdr 00:ca:fe:00:ca:fe dev eth0

#ip neighbor show
172.21.60.1 dev eth0 lladdr 00:e0:b1:a9:75:c0 STALE
192.168.1.1 dev eth0 lladdr 00:ca:fe:00:ca:fe PERMANENT
```

Configuration MAC

consulter ou changer son adresse MAC.

- utiliser `ifconfig`

```
#ifconfig eth0 hw ether 00:ca:fe:00:ca:fe

#ifconfig
eth0      Link encap:Ethernet  HWaddr 00:ca:fe:00:ca:fe
          ...
```

- utiliser les fonctionnalités d'iproute2

```
#ip link set eth0 addr 00:ca:fe:00:ca:fe

#ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state
    UNKNOWN mode DEFAULT group default
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state
    UP mode DEFAULT group default qlen 1000
    link/ether 00:ca:fe:00:ca:fe brd ff:ff:ff:ff:ff:ff
```

1 Modèles

2 Couches basses

3 Couche réseau
- Configuration IP
- Routage

4 Couche transport

5 Services réseaux
- Web
- Firewall
- NAT
- DNS

1. Modèles
2. Couches basses
3. Couche réseau
   - Configuration IP
   - Routage
4. Couche transport
5. Services réseaux
   - Web
   - Firewall
   - NAT
   - DNS

Adresse IP

- Chaque interface réseau (i. e. carte réseau) d'un hôte du réseau possède (au moins) une adresse IP unique dans le réseau.
- Une interface ethernet est désignée par ethX ou X est le numéro de l'interface.
- Une adresse IP est un nombre de 32 bits souvent noté en décimal pointé ; quatre entiers (compris entre 0 et 255) séparés par des points. Exemple : 192.168.2.200.
- L'adresse IP est structurée en deux parties :
  1. partie réseau : permet de désigner le réseau (*netID*)
  2. partie hôte : permet de désigner l'hôte dans le réseau (*hostID*)
- Un masque de sous-réseau permet de séparer partie réseau et partie hôte.

CIDR

- CIDR *Classless Inter-Domain Routing* : suppression de la notion de classe réseau
- notation /X où X désigne le nombre de bits à 1 dans le masque de sous-réseau
- permet d'agréger plusieurs réseaux en un seul
- permet de découper un réseau en plusieurs
- permet de simplifier les tables de routages
- exemple :

```
125.0.0.0/8
135.18.0.0/16
195.220.84.0/24
193.48.96.0/20
192.168.128.0/27
```

Utilisation du masque

- Ce masque est une succession de 1 suivi d'une succession de 0 qui donne l'étendue de la partie réseau.
- mettre tous les bits de la partie réseaux à 0 nous donne la partie hôte.
- mettre tous les bits de la partie hôte à 0 nous donne la partie réseau.
- exemple : 192.168.168.38 avec masque de 255.255.255.224

11000000	10101000	10101000	00100110	192.168.168.38
11111111	11111111	11111111	11100000	255.255.255.224
11000000	10101000	10101000	00100000	$\Longrightarrow$ 192.168.168.32

- mettre tous les bits de la partie hôte à 1 nous donne l'adresse de diffusion dans le réseaux..

Configuration IP

Trois solutions :

- configurer le fichier `/etc/network/interface` en remplaçant

  ```
  iface eth0 inet dhcp
  ```

  par :

  ```
  iface eth0 inet static
  address 192.168.1.1
  netmask 255.255.255.0
  ```

- utiliser `ifconfig`

  ```
  #ifconfig eth0 192.168.1.1/24
  ```

- utiliser les fonctionnalités d'iproute2 (consultation `ip addr show`)

  ```
  #ip addr add 192.168.1.1/24 dev eth0
  ```

Configuration IP

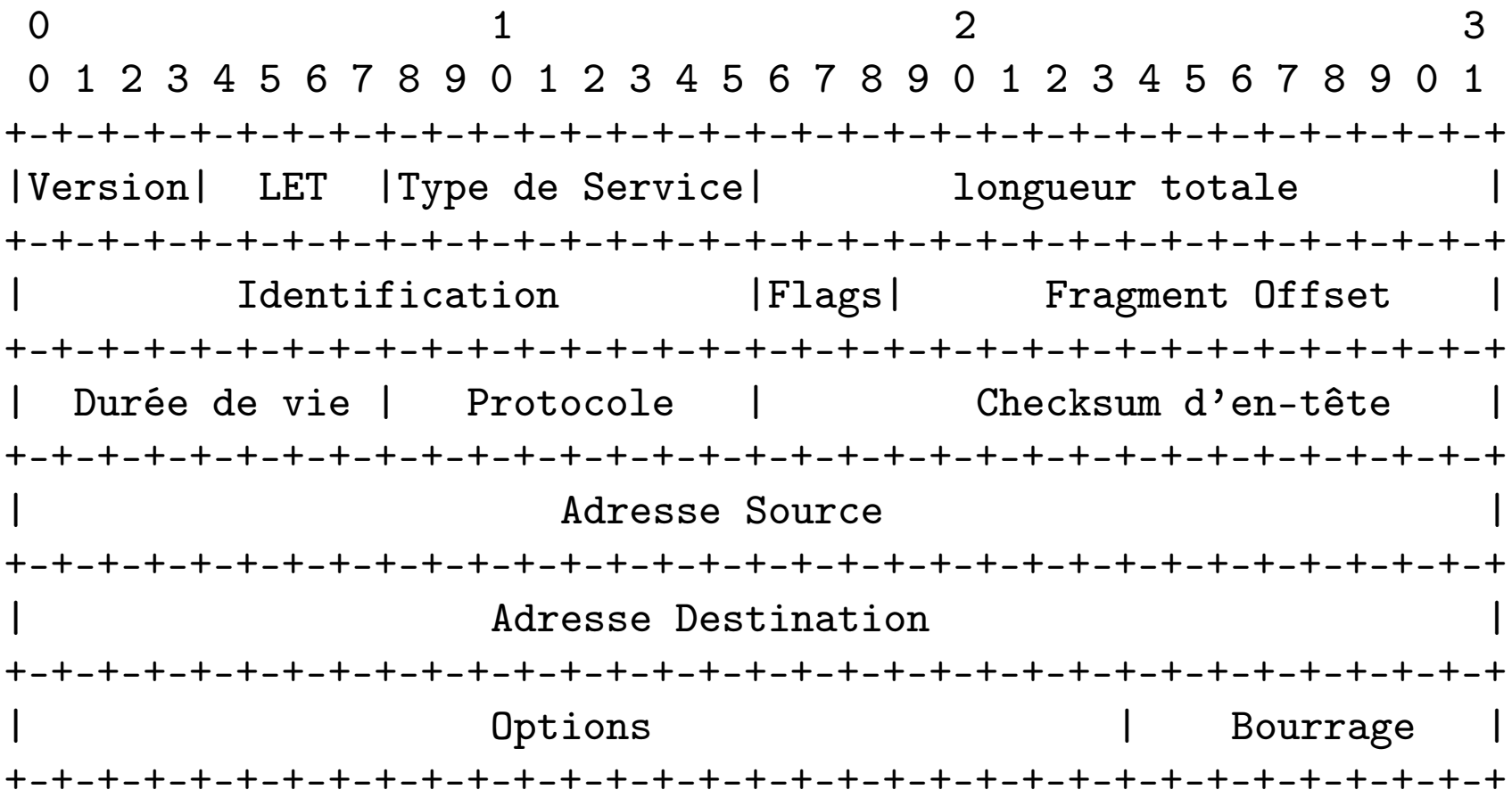
```
#ifconfig
eth0    Lien encap:Ethernet  HWaddr 00:00:C0:9A:01:F2
        inet adr:192.168.0.7 Bcast:192.168.0.255 Masque:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:0 errors:197 dropped:0 overruns:0 carrier:197
        collisions:0 lg file transmission:100
        RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
        Interruption:10 Adresse de base:0xc400

lo      Lien encap:Boucle locale
        inet adr:127.0.0.1  Masque:255.0.0.0
        UP LOOPBACK RUNNING  MTU:16436  Metric:1
        RX packets:188 errors:0 dropped:0 overruns:0 frame:0
        TX packets:188 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 lg file transmission:0
        RX bytes:14264 (13.9 Kb)  TX bytes:14264 (13.9 Kb)
```

Configuration IP

```
#ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast st
    link/ether 00:00:c0:9a:01:f2 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.1/24 brd 192.168.1.255 scope global eth0
    inet6 fe80::200:c0ff:fe9a:01f2/64 scope link
       valid_lft forever preferred_lft forever
```

Paquet IP


0 1 2 3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
Version
LET
Type de Service
longueur totale
Identification
Flags
Fragment Offset
Durée de vie
Protocole
Checksum d'en-tête
Adresse Source
Adresse Destination
Options
Bourrage

1. Modèles
2. Couches basses
3. Couche réseau
   - Configuration IP
   - Routage
4. Couche transport
5. Services réseaux
   - Web
   - Firewall
   - NAT
   - DNS

Routage

- le routage permet l'interconnexion de réseaux
- il est inutile sur un LAN isolé.
- l'information transite d'un réseau à l'autre par des hôtes spécialisés appelés routeur
- un routeur possède une connexion sur chacun des réseaux qu'il interconnecte
- le routeur maintient une table de routage

Routage : principe

- Un hôte voulant faire une transmission constitue un paquet IP
- Ce paquet contient l'adresse du destinataire et l'adresse de l'expéditeur.
- Au niveau de la couche réseau, le routage utilise une **table de routage** qui contient une ou plusieurs lignes contenant chacune essentiellement trois informations :
  1. une adresse de réseau : la destination
  2. un masque de réseau
  3. comment atteindre le réseau :
     - soit **directement** par une interface connectée sur ce réseau (on parle de routage direct),
     - soit **en passant par un routeur** (on parle de routage indirect) qui est identifié par son IP et l'interface à utiliser pour l'atteindre.
- Un routeur peut être un équipement spécialisé ou simplement un hôte ordinaire relié à plusieurs réseaux.

Routage : mécanisme

- la décision de routage se fait par la recherche d'une correspondance dans la table de routage
- on applique pour chaque ligne, le masque de réseau à l'adresse de destination.
- Quatre cas peuvent alors se présenter :
  1. le réseau de la destination est directement connecté. Il y une remise directe en utilisant le réseau local sous-jacent.
  2. le réseau de la destination est accessible via un routeur. Le paquet est transmis au routeur sans changer les adresses IP de l'émetteur et du destinataire.
  3. le réseau de la destination est absent de la table de routage, mais il existe une route par défaut. Le paquet est transmis au routeur désigné.
  4. le réseau de la destination est absent de la table de routage, et il n'existe pas de route par défaut. Envoi d'un message ICMP à l'émetteur : `Network is unreachable`
- Chaque routeur recevant un paquet IP applique le même algorithme.

Configuration du routage

- routage non adaptatif ou routage statique : la table de routage est gérée manuellement
- routage adaptatif ou routage dynamique : la table de routage est gérée automatiquement. C'est à dire qu'un routeur du réseau applique un algorithme qui, en fonction d'information sur l'état du réseau, lui permet de calculer sa table de routage.

Routage statique

Trois solutions :

- configurer le fichier `/etc/network/interface` en remplaçant

  ```
  iface eth0 inet dhcp
  ```

  par :

  ```
  iface eth0 inet static
  address 192.168.1.1
  netmask 255.255.255.0
  gateway 192.168.1.254
  # route statique supplementaire
  up route add -net 172.20.11.0/16 gw 192.168.1.253 dev eth0
  ```

- utiliser la commande `route`

  ```
  #route add -net 172.20.11.0/16 gw 192.168.1.253
  ```

- utiliser la commande `ip`

  ```
  #ip route add 172.20.11.0/16 via 192.168.1.253
  ```

Routes

- `route`

```
#route
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref
172.21.60.0     *               255.255.252.0   U     0      0
10.0.0.0        172.21.63.5     255.0.0.0       UG    0      0
default         172.21.60.1     0.0.0.0         UG    100    0
```

- `ip route show`

```
#ip route
172.21.60.0/22 dev eth0  proto kernel  scope link  src 172.21.62.8
10.0.0.0/8 via 172.21.63.5 dev eth0
default via 172.21.60.1 dev eth0  metric 100
```

Routage dynamique

Principe :

- les routeurs échangent des informations avec leurs voisins
- ils se servent de ces informations pour établir leur table de routage
- les modifications de le topologie des réseaux (coupures, nouvelles liaisons) sont prises en comptes automatiquement.

Routage dynamique

Les deux familles de protocoles les plus répandues sont :

1. protocoles à vecteurs de distance
2. protocoles à états de lien

Dans les deux cas,

Protocoles à vecteur de distance

- échange local d'informations globales
- table d'un routeur transmise à ses voisins
- table de routage $\simeq$ unions tables de routage
- la distance est le nombre de sauts à faire (*hops*)
- algorithme de Bellman-Ford.
- exemple : RIP
- inconvénients : convergence lente, 15 sauts maximum, problème du "comptage à l'infini", ...

Protocoles à état de lien

- échange global d'informations locales
- état des liens transmis à tous les routeurs
- principe :
  - découverte des voisins
  - mesure de la distance à chaque voisin
  - construction d'un paquet contenant ces informations
  - transmissions aux autres routeurs
  - chaque routeur calcule le chemins le plus courts vers les autres (algorithme *Shortest Path First* de Dijkstra)
- différentes métriques peuvent être utilisées : qualité du lien, encombrement, le coût financier...
- exemple : OSPF
- inconvénients : charge de calcul

1 Modèles

2 Couches basses

3 Couche réseau
- Configuration IP
- Routage

4 Couche transport

5 Services réseaux
- Web
- Firewall
- NAT
- DNS

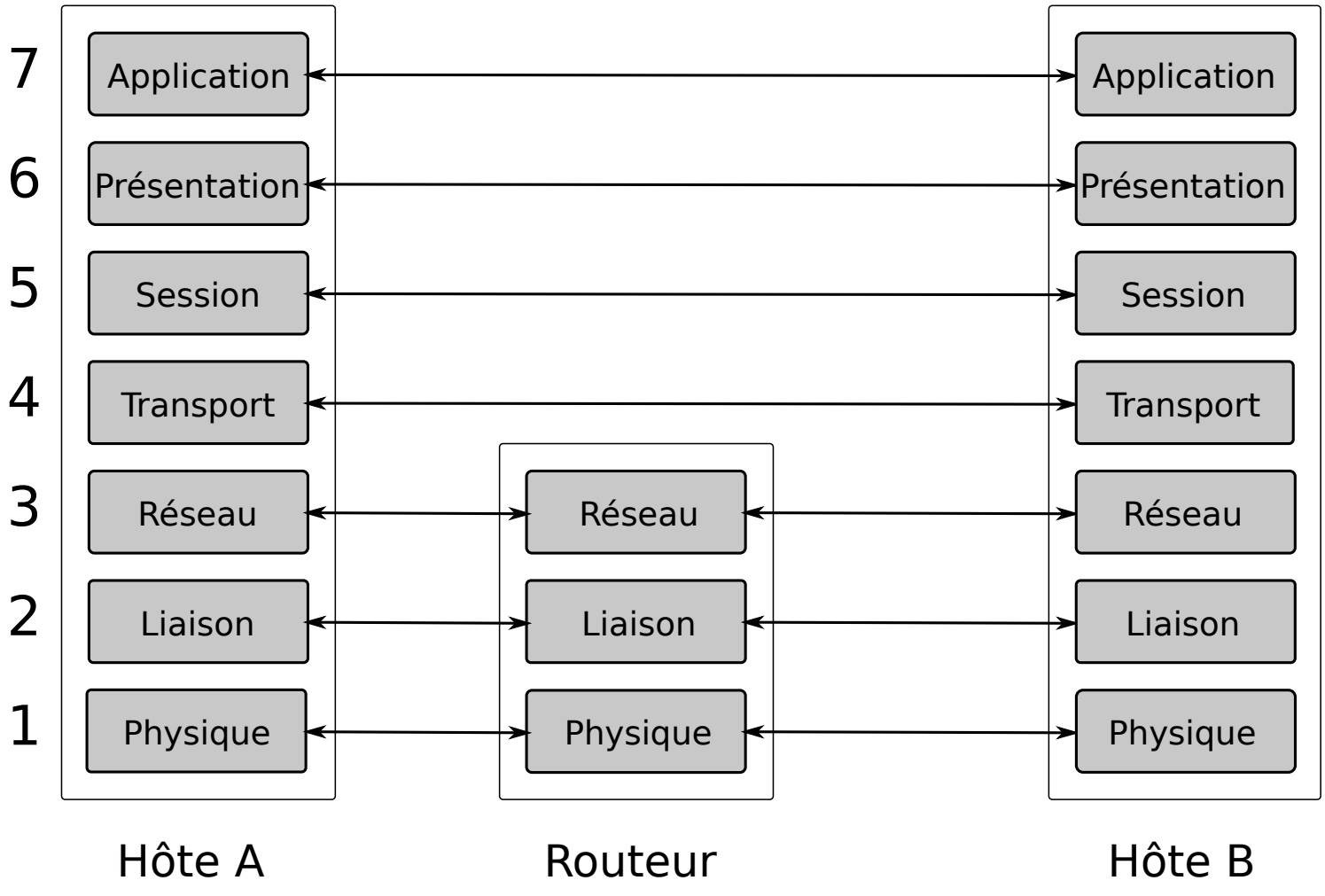
But

- La couche réseau offre une communication de machine à machine.
- La couche transport permet :
  - une communication d'application à application,
  - un transfert fiable :
    - sans corruption : checksum
    - sans pertes : FOO BAR $\nrightarrow$ BAR
    - dans l'ordre : FOO BAR $\nrightarrow$ BAR FOO
    - sans duplication : FOO BAR $\nrightarrow$ FOO FOO BAR
  - des services avec ou sans connexion,
  - d'offrir différentes qualités de service (QoS) pour le mode connecté,
  - une communication de bout en bout.

Connecté/Non Connecté

- Mode connecté : établissement d'une connexion avant transmission puis transmission et libération de la connexion (téléphone)
- Mode non connecté : transmission directe (courrier postal)

Transport

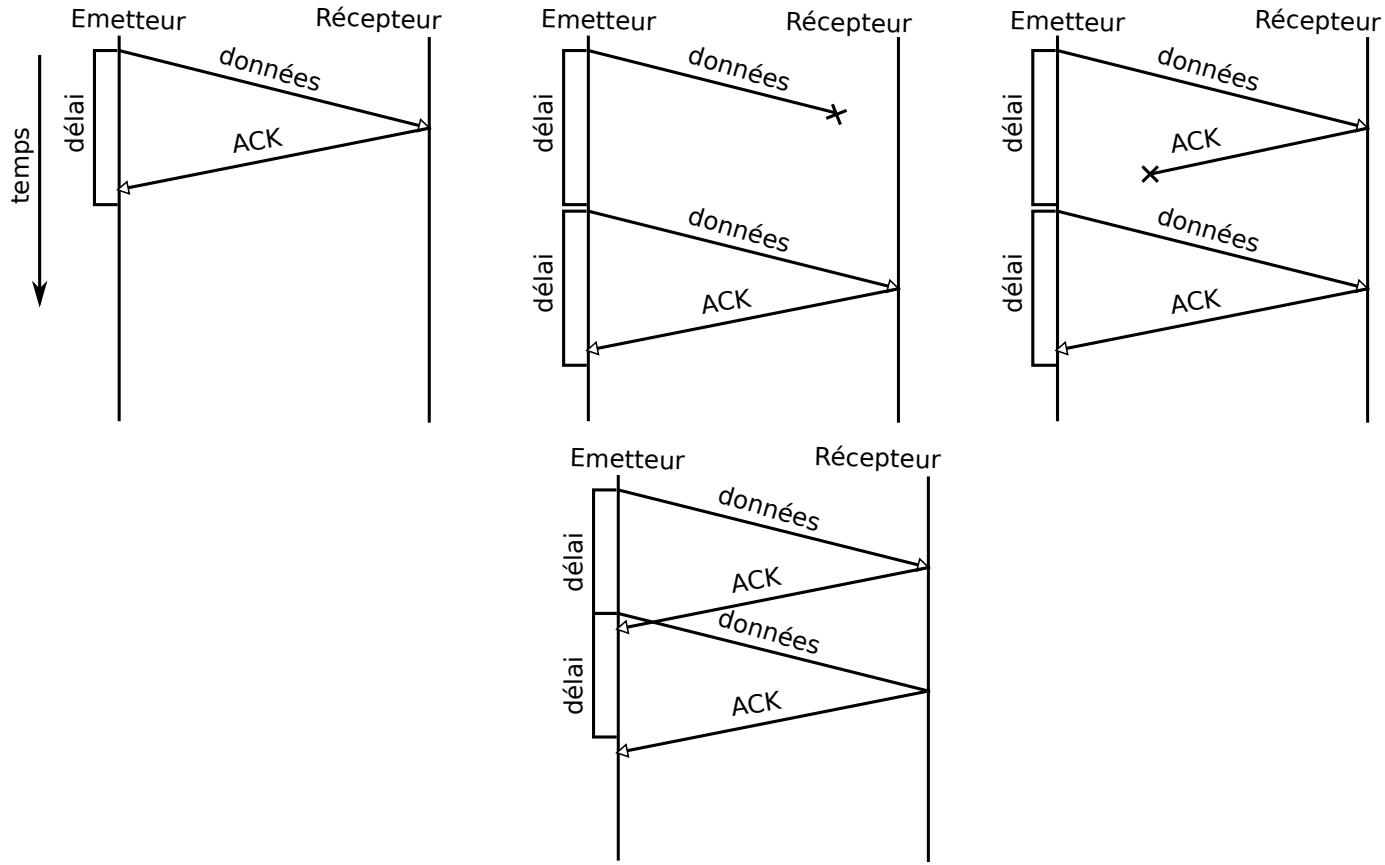

7
6
5
4
3
2
1
Application
Présentation
Session
Transport
Réseau
Liaison
Physique
Application
Présentation
Session
Transport
Réseau
Liaison
Physique
Réseau
Liaison
Physique
Hôte A
Routeur
Hôte B

Problèmes à résoudre

- Transport fiable
- Adressage
- Établissement/Libération d'une connexion
- Contrôle de flux ($\rightarrow$ contrôle de congestion)
- Multiplexage / segmentation

Transport fiable

- Principe général : *Automatic Repeat reQuest* (ARQ) : acquittements + timeouts
- Basic : émettre et attendre (*Stop-and-wait ARQ*)



- nécessité du *numéro de séquence* → identification des données

Transport fiable

- Problème : délai entre deux paquets égal au double du temps de transmission
- Solution : envoyer plusieurs paquets successivement sans attendre d'acquittement
- Plusieurs déclinaisons :
  - *Go-Back-N ARQ* : retransmission complète à partir de la détection de perte
  - *Selective Repeat ARQ* : possibilité d'acquitter/de redemander des données sélectivement

Flux vs Congestion

- contrôle de congestion :
  - problème global du réseau pour éviter trop de trafic dans un sous-réseau
  - cas typique : réseau lent, nombreux terminaux transmettant de gros fichiers
- contrôle de flux :
  - éviter qu'un émetteur rapide ne sature un récepteur lent
  - cas typique : réseau rapide, un terminal rapide transmet à un terminal lent

1 Modèles

2 Couches basses

3 Couche réseau
- Configuration IP
- Routage

4 Couche transport

5 Services réseaux
- Web
- Firewall
- NAT
- DNS

Services réseaux

- Web
- Firewall
- NAT
- DNS

1. Modèles
2. Couches basses
3. Couche réseau
   - Configuration IP
   - Routage
4. Couche transport
5. Services réseaux
   - Web
   - Firewall
   - NAT
   - DNS

Web

- *World Wide Web* (toile d'araignée mondiale) ou web :
  - un des service d'internet
  - crée par Tim Berners-Lee au CERN en 1989
  - concept d'information distribuée et d'hypermédia
  - document $\longrightarrow$ référence aux autres documents
  - 1993 première interface utilisateur conviviale : Mosaic

Modèle client-serveur

Deux acteurs :

- le client (ex : firefox) : effectue des requêtes vers le serveur
- le serveur (ex : Apache) : exécute les requêtes et renvoie le résultat au client .

Rôle d'un programme client :

- traduire les ordres de l'utilisateur en messages conformes à un protocole d'échange avec un serveur
- contacter le serveur adéquat et lui passer la requête
- attendre la réponse du serveur
- mettre en forme la réponse et la présenter à l'utilisateur

Trois mécanismes

- schéma d'adressage uniforme $\rightarrow$ localiser : URLs
- protocoles $\rightarrow$ communication : HTTP
- documents hypermédias $\rightarrow$ navigation : HTML

URL : *Uniform Ressource Locator*

Les documents hypermédias sont répartis à travers le monde $\Longrightarrow$ identification et localisation de manière unique d'un document.

Format des URLs

1. nom du protocole
2. nom de la machine (i. e. serveur)
3. nom ressource

```
protocole://serveur[:port][/chemin][/fichier][#position]
```

protocole	le protocole d'échange entre le client et le serveur. Le plus souvent on utilise `http` ou `ftp`
serveur	l'adresse internet du serveur qui diffuse les documents.
port	Numéro du port
chemin	le chemin (suite de répertoires séparés par des /)
fichier	le nom du document qui nous intéresse
position	une position précise à l'intérieur du document

Protocole HTTP

- défini les échanges entre un serveur web et un client
- client $\rightarrow$ serveur :
  - `GET` demande une ressource
  - `POST` transmission de données à une ressource
  - `DELETE` suppression d'une ressource
  - `HEAD` demande des informations sur la ressource, sans la ressource elle-même.
  - `PUT` ajout ou remplacement d'une ressource
  - ...
- serveur $\rightarrow$ client :
  - codes 500 : erreur serveur
  - codes 400 : erreur client
  - codes 300 : redirection
  - codes 200 : succès
  - codes 100 : information

Protocole HTTP

- HTTP/0.9 : ouverture, GET, réponse, fermeture
- HTTP/1.0 : utilisation d'en-têtes (`Host`, `Referer`, `Content-Type`, `Content-Length`, ...), autres méthodes, ...
- HTTP/1.1 : `Host` obligatoire, connexions persistantes
- HTTP/2.0 :
  1. multiplexage
  2. compression des entêtes,
  3. push,
  4. chiffrement (non obligatoire),
- HTTP/3.0 : transport QUIC basé sur UDP

Dialogue HTTP

```
GET /demo/ HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Demo HTTP
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
Accept-Encoding: gzip, deflate, br
Accept-Language: fr-FR,fr;q=0.9

HTTP/1.1 200 OK
Date: Fri, 15 Sep 2023 14:09:34 GMT
Server: Apache/2.4.57 (Unix) PHP/8.2.10
X-Powered-By: PHP/8.2.10
Content-Length: 14
Content-Type: text/html; charset=UTF-8

<html>...
```

Dialogue HTTP : Rechargement de page

```
GET /demo/index.html HTTP/1.1
...
If-Modified-Since: uneDate
If-None-Match: "e-60525898e5800"
Cache-Control: max-age=0

HTTP/1.1 304 Not Modified
Date: Date: Fri, 15 Sep 2023 14:25:50 GMT
Server: Apache/2.4.57 (Unix) PHP/8.2.10
Connection: Keep-Alive
Keep-Alive: timeout=15, max=100
ETag: "e-60525898e5800"
```

Dialogue HTTP : Redirection

```
GET /demo HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Demo HTTP
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
Accept-Encoding: gzip, deflate, br
Accept-Language: fr-FR,fr;q=0.9

HTTP/1.1 301 Moved Permanently
Date: Fri, 15 Sep 2023 14:30:03 GMT
Server: Apache/2.4.57 (Unix) PHP/8.2.10
Location: http://127.0.0.1:8080/demo/
Content-Length: 235
Content-Type: text/html; charset=iso-8859-1
```

Dialogue HTTP : Erreurs Client

```
GET /demo/erreur403.html HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Demo HTTP
Accept: */*

HTTP/1.1 403 Forbidden
Date: Fri, 15 Sep 2023 14:36:46 GMT
Server: Apache/2.4.57 (Unix) PHP/8.2.10
Content-Length: 199
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
page erreur
...
```

Dialogue HTTP : Erreurs Client

```
GET /demo/erreur404.html HTTP/1.1
Host: 127.0.0.1:8080
User-Agent: Demo HTTP
Accept: */*

HTTP/1.1 404 Not Found
Date: Fri, 15 Sep 2023 14:37:48 GMT
Server: Apache/2.4.57 (Unix) PHP/8.2.10
Content-Length: 196
Content-Type: text/html; charset=iso-8859-1
...
```

Dialogue HTTP : Erreurs Client

```
GET / HTTP/1.1

HTTP/1.1 400 Bad Request
Date: Wed, 08 Dec 2023 16:33:53 GMT
Server: Apache/2.0.55 (Ubuntu)
Content-Length: 301
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>400 Bad Request</title>
</head><body>
<h1>Bad Request</h1>
<p>Your browser sent a request that this server could not understand.<br />
</p>
<hr>
<address>Apache/2.2.10 (Ubuntu) Server at 127.0.0.1 Port 80</address>
</body></html>
```

Proxy HTTP

proxy ou serveur mandataire :

- relaie des requêtes entre un client et un serveur
- fait office de cache
- aide à la sécurisation du réseau local
- journalisation des requêtes (log)

Serveurs HTTP

nombreuses implémentations du protocole HTTP

- Apache
- Nginx
- IIS (Microsoft)
- Lighttpd,
- ...

Apache

- dérivé de NCSA httpd
- modulaire $\rightarrow$ chargement de fonctionnalités supplémentaires :
  - CGI, SSI,
  - réécriture d'URL,
  - négociation de contenu,
  - protocoles de communication additionnels,
  - pages personnelles,
  - ...

Configuration serveur

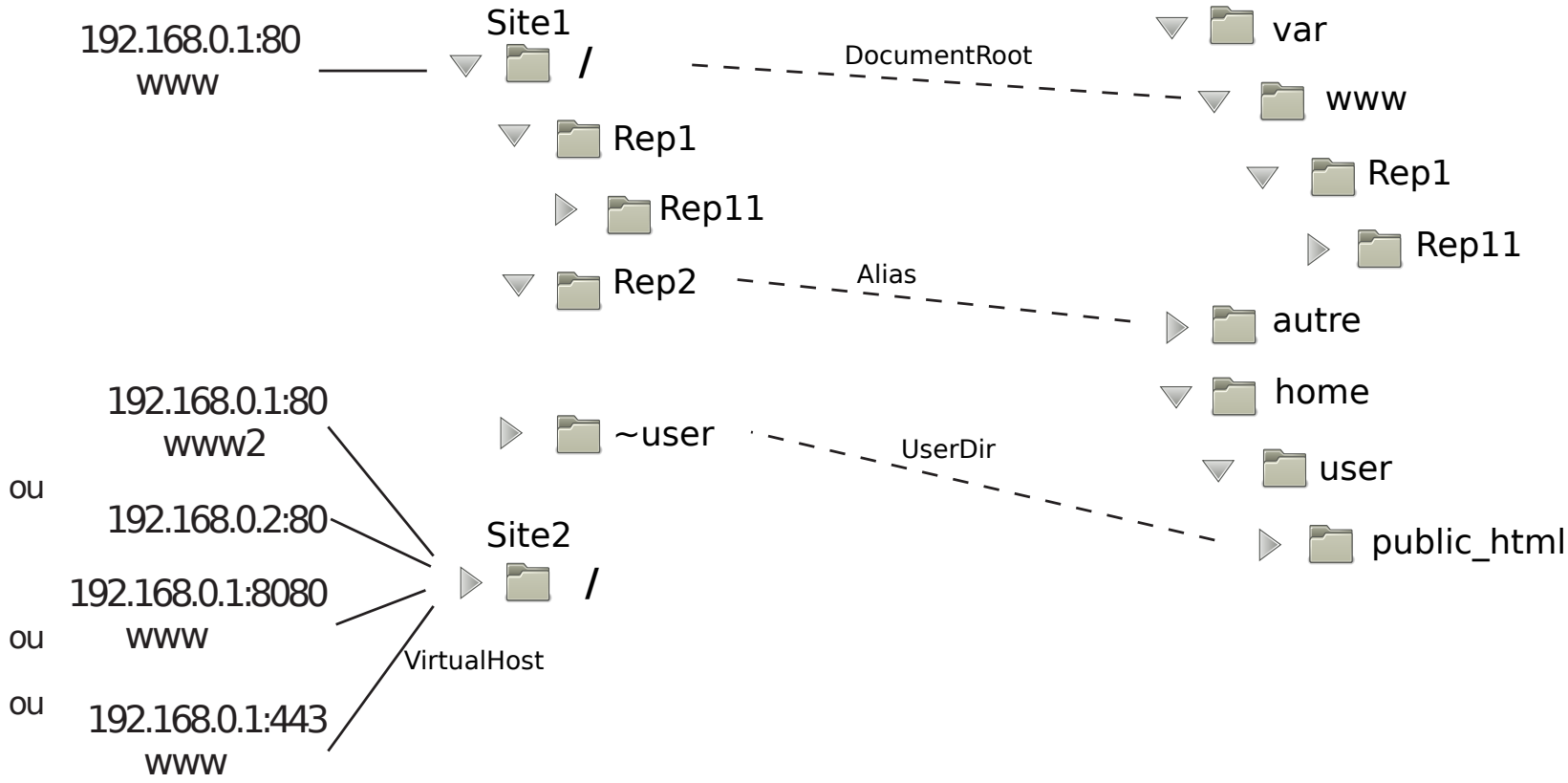


Figure – Serveur HTTP

Configuration

- fichiers de configuration :

```
/etc/apache2/
├── apache2.conf
│       └──  ports.conf
├── mods-enabled
│       ├── *.load
│       └── *.conf
├── conf-enabled
│       └── *.conf
└── sites-enabled
        └── *.conf
```

- configuration par des directives :

```
Directive valeur1 valeur2
```

ou

```
<portée>
    Directive valeur1 valeur2
</portée>
```

- portée de la configuration :
  - globale au serveur,
  - pour un serveur virtuel (`<VirtualHost>`),
  - un répertoire ou un fichier (`<Directory>`, `<Location>` et `<Files>`)
  - ou définie par un utilisateur pour un répertoire → `.htaccess`

Configuration serveur (base)

```
# sockets d'écoutes
Listen 192.168.0.1:80
# nom du serveur par défaut
ServerName www.fai.com
#
DocumentRoot "/Site1/"
# utilisateur non-privilégié
User apache2
Group apache2

<Directory "/Site1/">
# options parmi :
#   Indexes Includes FollowSymLinks SymLinksifOwnerMatch ExecCGI
    Options Indexes FollowSymLinks
# voir plus loin -> .htaccess
    AllowOverride None
# droits d'accès
    Require all granted
</Directory>
```

Configuration serveur (hôtes virtuels)

```
# différentiation sur nom/ip/port -> sites différents
<VirtualHost 192.168.0.1:80>
    DocumentRoot /Site1/
    ServerName www.fai.com
</VirtualHost>

<VirtualHost 192.168.0.1:80>
    DocumentRoot /Site2/
    ServerName www2.fai.com
</VirtualHost>

<VirtualHost 192.168.0.2:*>
    DocumentRoot /Site2/
</VirtualHost>

<VirtualHost 192.168.0.1:8080>
    DocumentRoot /Site2/
    ServerName www.fai.com
</VirtualHost>
```

Configuration serveur (listing)

```
# fichiers du répertoire à charger par défaut (dans l'ordre)
DirectoryIndex index.html index.html.var default.html

# voir plus loin -> .htaccess
AccessFileName .htaccess

<Files ~ "^\.ht">
    Require all denied
</Files>

IndexOptions FancyIndexing VersionSort
ReadmeName README.html
HeaderName HEADER.html
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

Configuration serveur (logs)

```
ErrorLog logs/error_log

# debug, info, notice, warn, error, crit, alert, emerg.
LogLevel warn

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

CustomLog logs/access_log common
```

Configuration serveur (alias)

```
Alias /Rep2 "/autre/"

<Directory "/autre/">
    Options Indexes MultiViews
    AllowOverride None
    Require all granted
</Directory>

ScriptAlias /cgi-bin/ "/opt/apache2/cgi-bin/"
<Directory "/opt/apache2/cgi-bin">
    AllowOverride None
    Options None
    Require all granted
</Directory>

Redirect permanent /essai http://www.fai.com
```

Configuration serveur (modules)

- chargement d'un module

```
LoadModule negotiation_module /usr/lib/apache2/modules/mod_negotiation.so
```

- évaluation conditionnelle des directive

```
<IfModule mod_negotiation.c>
    LanguagePriority en ca cs da de el eo es et fr
</IfModule>
```

Configuration serveur (public_html)

```
<IfModule mod_userdir.c>
UserDir public_html

  <Directory /home/*/public_html>
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    <Limit GET POST OPTIONS PROPFIND>
      Require all granted
    </Limit>
    <LimitExcept GET POST OPTIONS>
      Require all denied
    </LimitExcept>
  </Directory>
</IfModule>
```

Configuration serveur (erreurs)

```
Alias /error/ "/opt/apache2/error/"
<Directory "/opt/apache2/error">
    AllowOverride None
    Options IncludesNoExec
    AddOutputFilter Includes html
    AddHandler type-map var
    Order allow,deny
    Allow from all
    LanguagePriority en fr de es it nl sv
    ForceLanguagePriority Prefer Fallback
</Directory>
ErrorDocument 400 /error/HTTP_BAD_REQUEST.html.var
ErrorDocument 401 /error/HTTP_UNAUTHORIZED.html.var
ErrorDocument 403 /error/HTTP_FORBIDDEN.html.var
ErrorDocument 404 /error/HTTP_NOT_FOUND.html.var
ErrorDocument 405 /error/HTTP_METHOD_NOT_ALLOWED.html.var
ErrorDocument 408 /error/HTTP_REQUEST_TIME_OUT.html.var
ErrorDocument 410 /error/HTTP_GONE.html.var
ErrorDocument 411 /error/HTTP_LENGTH_REQUIRED.html.var
ErrorDocument 412 /error/HTTP_PRECONDITION_FAILED.html.var
ErrorDocument 413 /error/HTTP_REQUEST_ENTITY_TOO_LARGE.html.var
ErrorDocument 414 /error/HTTP_REQUEST_URI_TOO_LARGE.html.var
ErrorDocument 415 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 500 /error/HTTP_INTERNAL_SERVER_ERROR.html.var
ErrorDocument 501 /error/HTTP_NOT_IMPLEMENTED.html.var
ErrorDocument 502 /error/HTTP_BAD_GATEWAY.html.var
ErrorDocument 503 /error/HTTP_SERVICE_UNAVAILABLE.html.var
ErrorDocument 506 /error/HTTP_VARIANT_ALSO_VARIES.html.var
```

.htaccess

- permet à un utilisateur de (re)définir des directives pour son (sous)site
- se comporte comme une directive `<Directory>`
- soumis à autorisation : `AllowOverride option`
- parmi : `AuthConfig`, `FileInfo Indexes`, `Limit`, ...
- le nom du fichier peut être redéfini : `AccessFileName`
- couteux en temps de calcul

1. Modèles
2. Couches basses
3. Couche réseau
   - Configuration IP
   - Routage
4. Couche transport
5. Services réseaux
   - Web
   - Firewall
   - NAT
   - DNS

Firewall

- les protocoles d'Internet ont été conçus pour réaliser un transport robuste, non pour leur sécurité
- mise en oeuvre de solutions de sécurité et parmi elles les firewalls ou pare-feu
- permet le passage sélectif des flux d'information entre deux réseaux
  - protection d'un réseau interne d'intrusions entrantes
  - limitation des accès sortant depuis le réseau interne
- les firewalls sont logiciels ou matériels
- services :
  - filtrage ip (+ transport)
  - NAT/PAT
  - proxy

Principe

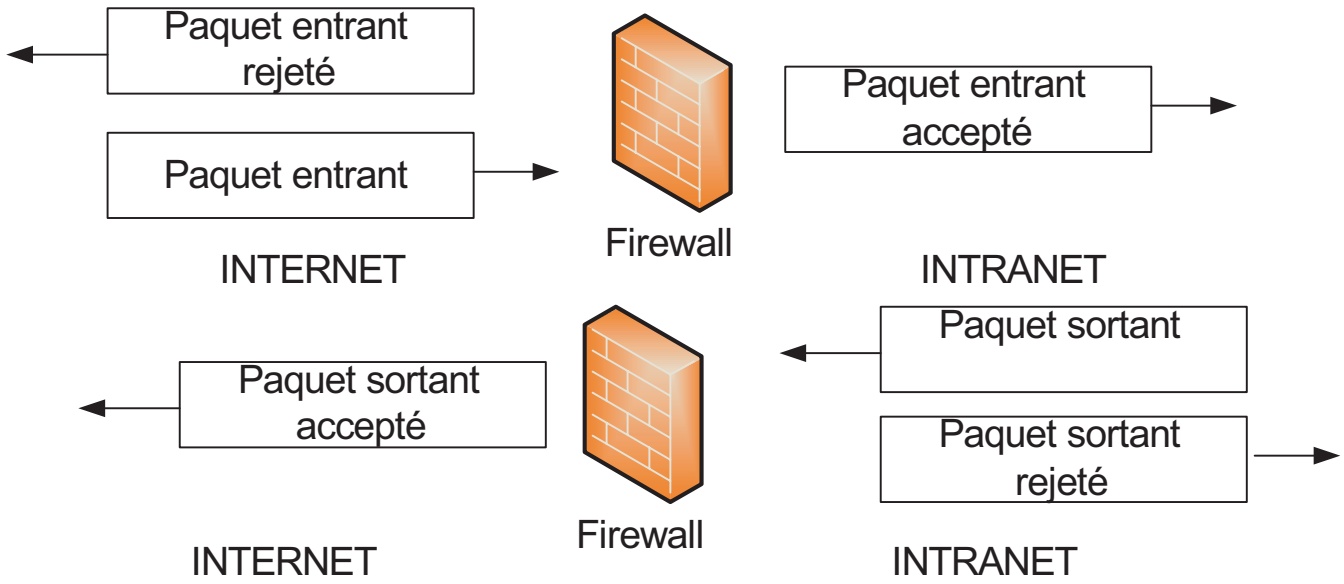
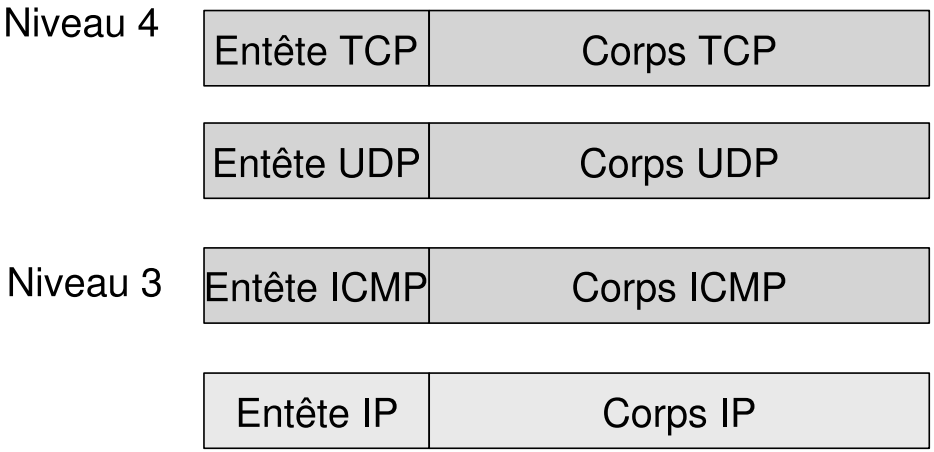


Figure – Principe

Inspection des paquets

- filtrage sur :
  - origine ou la destination des paquets (interfaces, IP, ports, ...)
  - options des données (flags, fragmentation, validité, ...)
  - données (taille, correspondance à un motif, ...)



- $\Rightarrow$ des règles
- problème en cas de fragmentation

Catégories

- Pare-feu sans état (*stateless firewall*) : le plus basique ; fonction de filtrage statique (ex : ipchains, iptables). Analyse de chaque paquet indépendamment des autres
- Pare-feu à états (*stateful firewall*) : vérification de la conformité d'un paquet à une communication normale (ex : conntrack). Analyse des paquets comme faisant partie d'une connexion.
- Pare-feu applicatif : utilisation de proxys, spécialisés dans un protocole particulier, pour faire du filtrage (ex : l7 Filter). $\rightarrow$ gourmand en temps cpu, problème de HTTPS

DMZ

- zone démilitarisée
  - en anglais *demilitarized zone*
  - sous-réseau séparé et isolé du réseau local et d'Internet par un pare-feu
  - contient les machines susceptibles d'être accédées depuis Internet.
  - compromission service dans la DMZ $\not\Leftrightarrow$ compromission dans le réseau local.

Architectures possibles

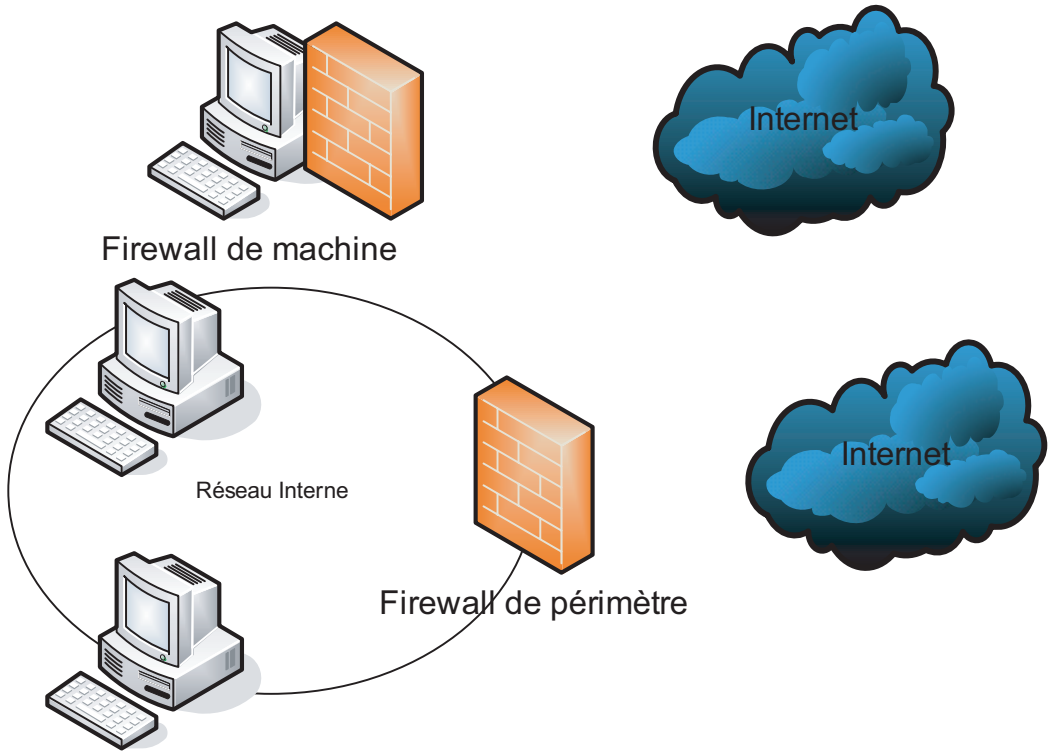


Figure – Architectures possibles (1)

Architecture possibles

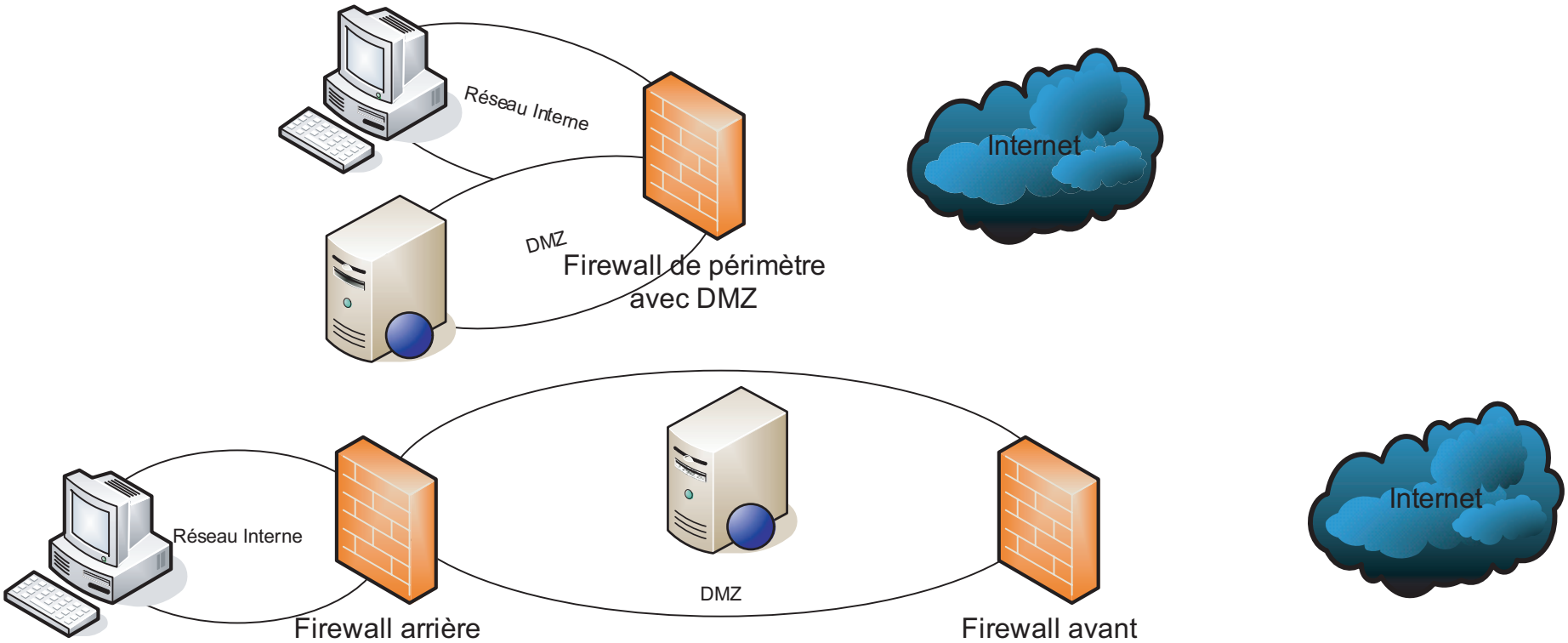


Figure – Architectures possibles (2)

Attaque

- Attaque par exploitation de faille : sondage, identification, intrusion, exploit, rootkit
- Attaque par déni de service : SYN flood, UDP Flood, ...
- detournements de flux : ARP poisoning, désynchronisation TCP, Man In the Middle
- attaque de protocoles particuliers

$\Rightarrow$ détails au module réseaux suivant

netfilter

- fonctionnalités :
  - filtrage de paquets réseaux
  - NAT
  - marquage de paquet
- LE firewall linux
- dans le kernel
- pas de fichier de configuration (`/etc`)
- se pilote (uniquement) par la commande `iptables`

- iptables c'est :
  - des **tables** (`Filter`, `Nat`, `Mangle`) qui contiennent des
  - **chaînes** (`PREROUTING`, `INPUT`, `FORWARD`, `OUTPUT` et `POSTROUTING`) qui contiennent des
  - **règles**

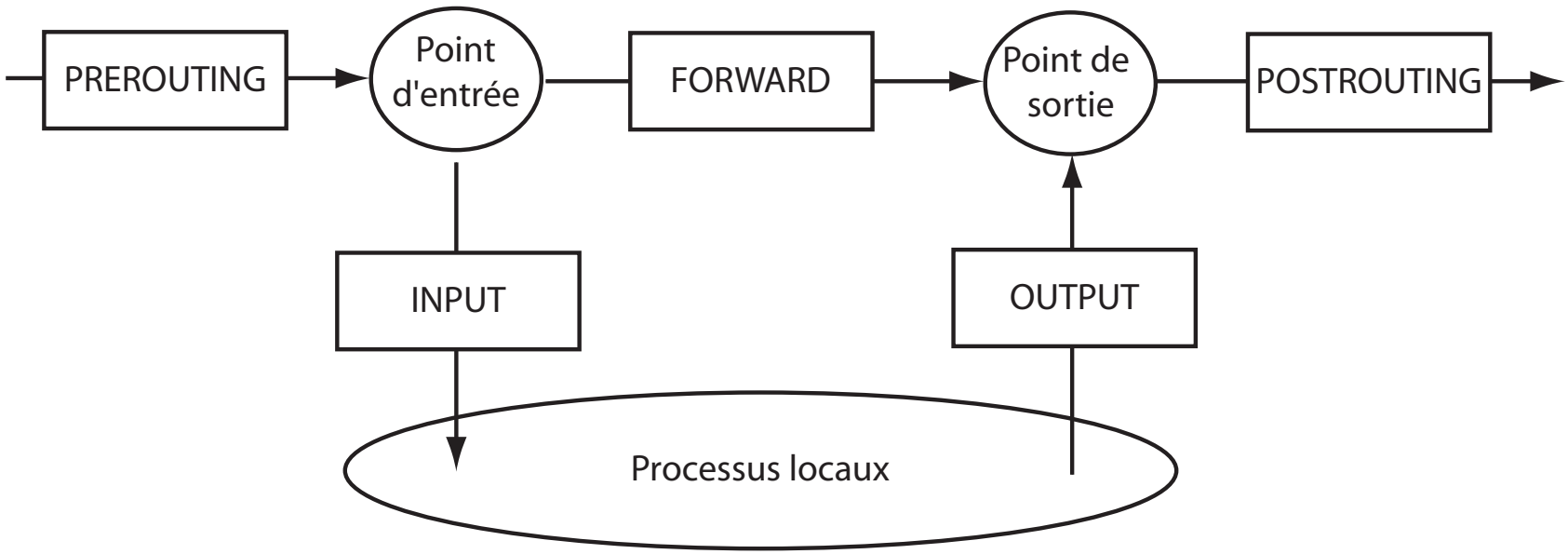


Figure – Principe iptables

Tables et chaînes

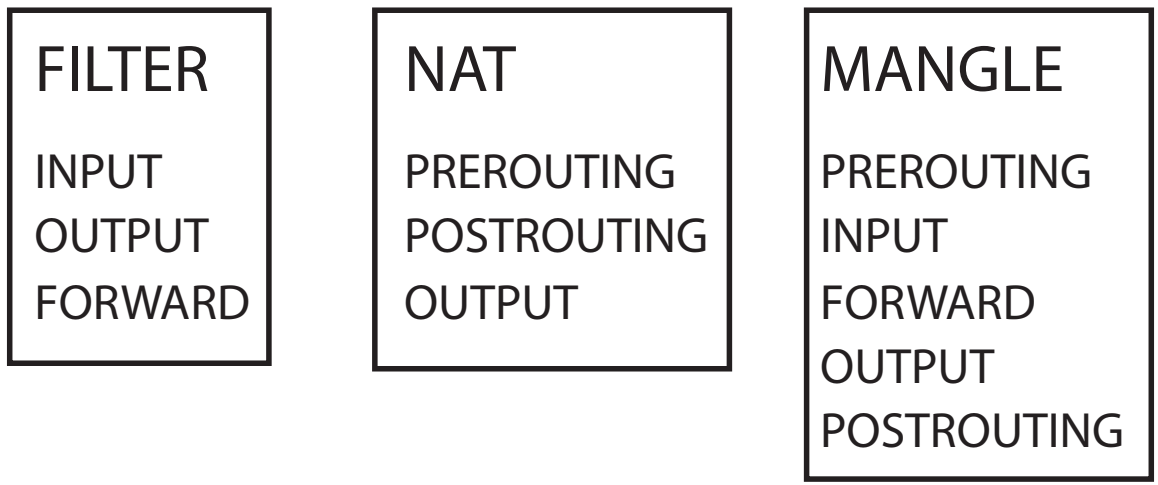


Figure – Tables et chaînes

Règles

`iptables [-t table] commande [correspondance] [cible]`

- commande :
  - `-A -apend`,
  - `-L -list`
  - ...
- correspondance :
  - `-s -source`
  - `-p -protocol`
  - `-i -in-interface`
  - ...
- cible ("action")
  - Filter : `DROP`, `ACCEPT`, `REJECT`, ...
  - Nat : `DNAT`, `SNAT`, `MASQUERADE`, ...
  - Mangle : `MARK`, ...
  - + des options pour certaines cibles

Exemples

```
iptables -t filter -F  # suppression des tables prédéfinies

# Définition de la politique (ensemble de règles) par défaut
#   On interdit tout
iptables -t filter -P INPUT DROP
iptables -t filter -P OUTPUT DROP
iptables -t filter -P FORWARD DROP

# trafic autorisé
iptables -t filter -A OUTPUT -o lo -s 127.0.0.0/8 -d 127.0.0.0/8 -j ACCEPT
iptables  -t filter -A INPUT -i lo -s 127.0.0.0/8 -d 127.0.0.0/8 -j ACCEPT
iptables  -t filter -A INPUT -i eth1 -s 192.168.114.0/24 -d 192.168.114.0/24 -j ACCEPT
iptables  -t filter -A OUTPUT -o eth1 -s 192.168.114.0/24 -d 192.168.114.0/24 -j ACCEPT

# Autorise HTTP (80/TCP)
iptables -t filter -A OUTPUT -o eth0 -s 195.168.114.0/24 -d 0.0.0.0/0  -p tcp --dport 80  -
iptables -t filter -A INPUT  -i eth0 -s 0.0.0.0/0  -d 195.168.114.0/24 -p tcp --sport 80  -

# Accès SSH depuis le Net
iptables -A INPUT -p tcp --dport ssh -i eth0 -j ACCEPT
iptables -A OUTPUT -p tcp --sport ssh -o eth0 -j ACCEPT

#Autoriser NAT
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth1 -j MASQUERADE
```

Suivi de connection (conntrack)

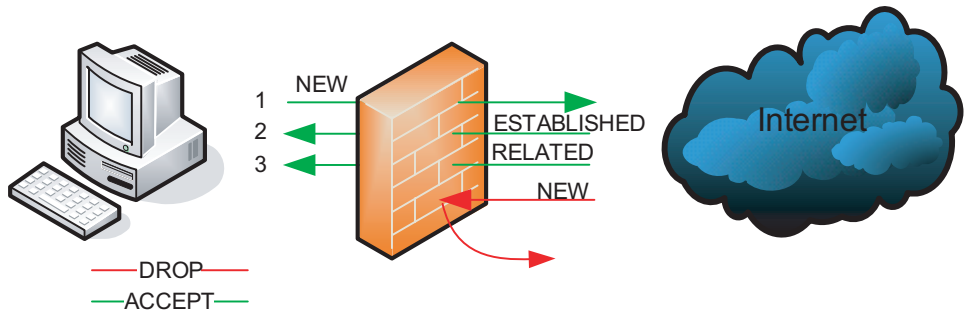


Figure – Tables et chaînes

```
# connexions LAN vers le Net acceptées
iptables -A FORWARD -i eth0 -o eth1 -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

# connexions Net vers le LAN acceptées
# soit déjà établies
# soit en relation avec  des connexions établies
iptables -A FORWARD -i eth1 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

1. Modèles
2. Couches basses
3. Couche réseau
   - Configuration IP
   - Routage
4. Couche transport
5. Services réseaux
   - Web
   - Firewall
   - NAT
   - DNS

NAT

- *Network Address Translation* (RFC 1631)
- nécessité par l'utilisation, dans des réseaux internes, d'adresses IP invalides ou confidentielles pour un usage externe
- typique : ipv4 privée `192.168.X.0/24` non valide hors du réseaux domestique
- modification des adresses IP dans l'en-tête d'un paquet IP effectuée par un routeur
- permet de traduire (réécrire à la volée) les adresses IP

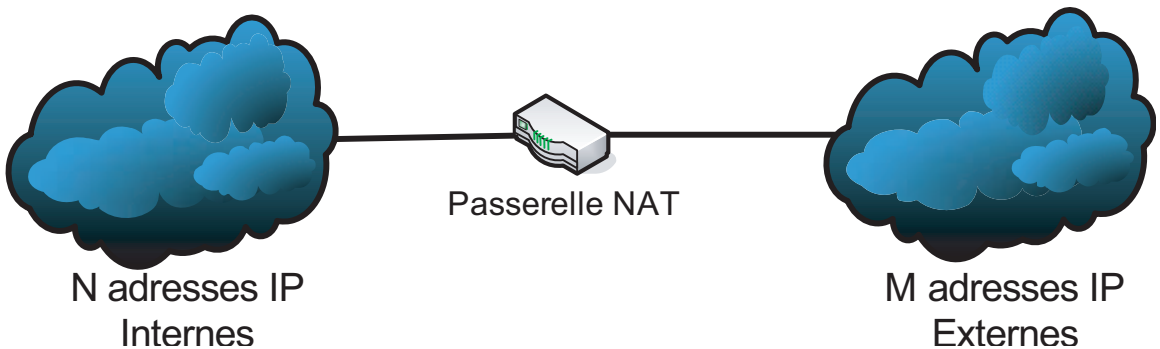


Figure – Nat

Concepts

- NAT se décline en différents concepts :
  - NAT Statique
  - NAT Dynamique
  - Source NAT (SNAT)
  - Destination NAT (DNAT)
- en différentes utilisations :
  - IP Masquerading
  - Port forwarding et mapping
  - Serveurs Virtuels (équilibrage de charge)
  - Routes Virtuelles
- en trois actions :
  - Source NAT (SNAT)
  - Destination NAT (DNAT)
  - Port NAT (PAT)

Nat statique/dynamique

- fait correspondre n adresses internes à n adresses externes
- Nat statique : toujours la même correspondance
- Nat dynamique : utilisation d'un pool d'adresse externe
- dans le sens interne vers externe l'adresse IP source et réécrite en l'adresse IP correspondante coté externe (SNAT)
- dans l'autre sens l'adresse IP de destination est réécrite en l'adresse IP interne (DNAT)
- avantage : une adresse privée peut accéder à Internet, les adresses privées sont masquées.
- inconvénient : il faut plusieurs adresses externes, pénurie d'adresses non résolue

Nat statique

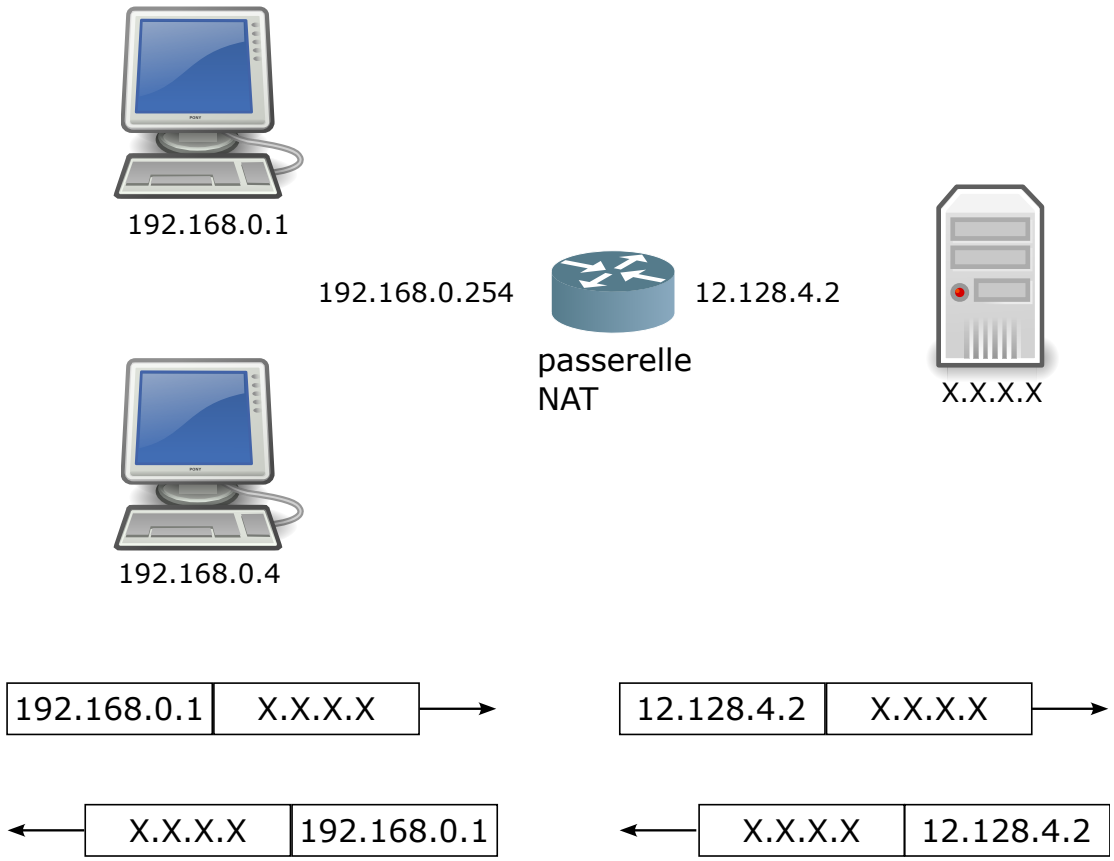
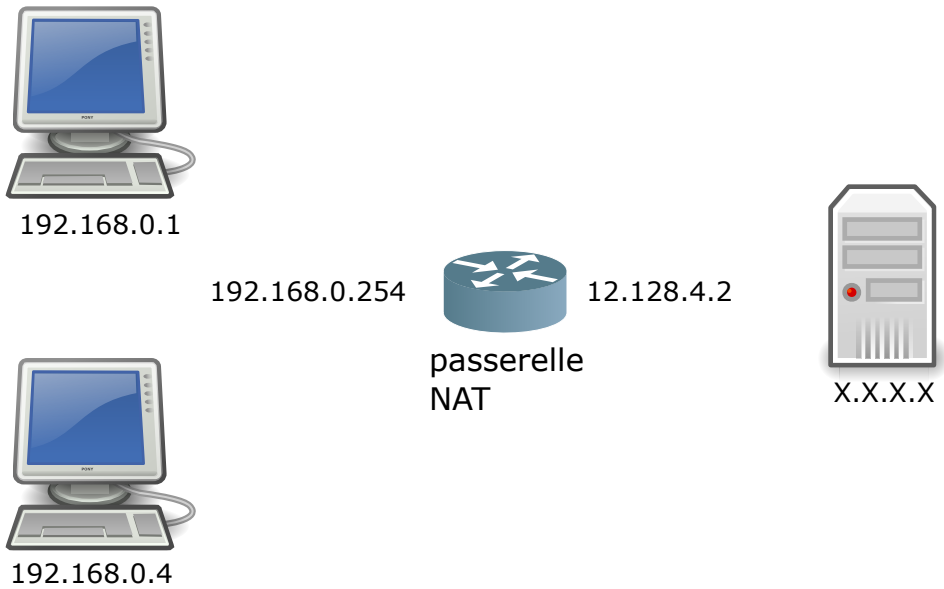


Figure – Nat statique

PAT (*Port Address Translation*)

- fait correspondre n adresses internes à m<n adresses externes (souvent 1)
- plus de correspondance directe ; on joue sur les ports
- avantages : plus de pénurie, sécurité
- inconvénient :
  - machines internes non joignables depuis Internet,
  - fragmentation : comment gérer un paquet ne contenant pas d'en-tête TCP ou UDP,
  - pour certains protocoles, problème de modification des adresses et des ports : FTP, ICMP (pas de port et un paquet dans la charge utile ICMP), DHCP, ...

PAT (*Port Address Translation*)



192.168.0.1	X.X.X.X
2509	80

X.X.X.X	192.168.0.1
80	2509

12.128.4.2	X.X.X.X
4242	80

X.X.X.X	12.128.4.2
80	4242

192.168.0.4	X.X.X.X
1818	80

X.X.X.X	192.168.0.4
80	1818

12.128.4.2	X.X.X.X
4244	80

X.X.X.X	12.128.4.2
80	4244

Port forwarding

- le routeur NAT redirige le trafic extérieur reçu sur un port spécifique vers une adresse IP interne
- un client externe accède à un serveur virtuellement placé sur la passerelle NAT et physiquement placé sur une machine interne
- problème : on ne peut avoir qu'un seul serveur d'un seul type accessible depuis l'extérieur

Port forwarding

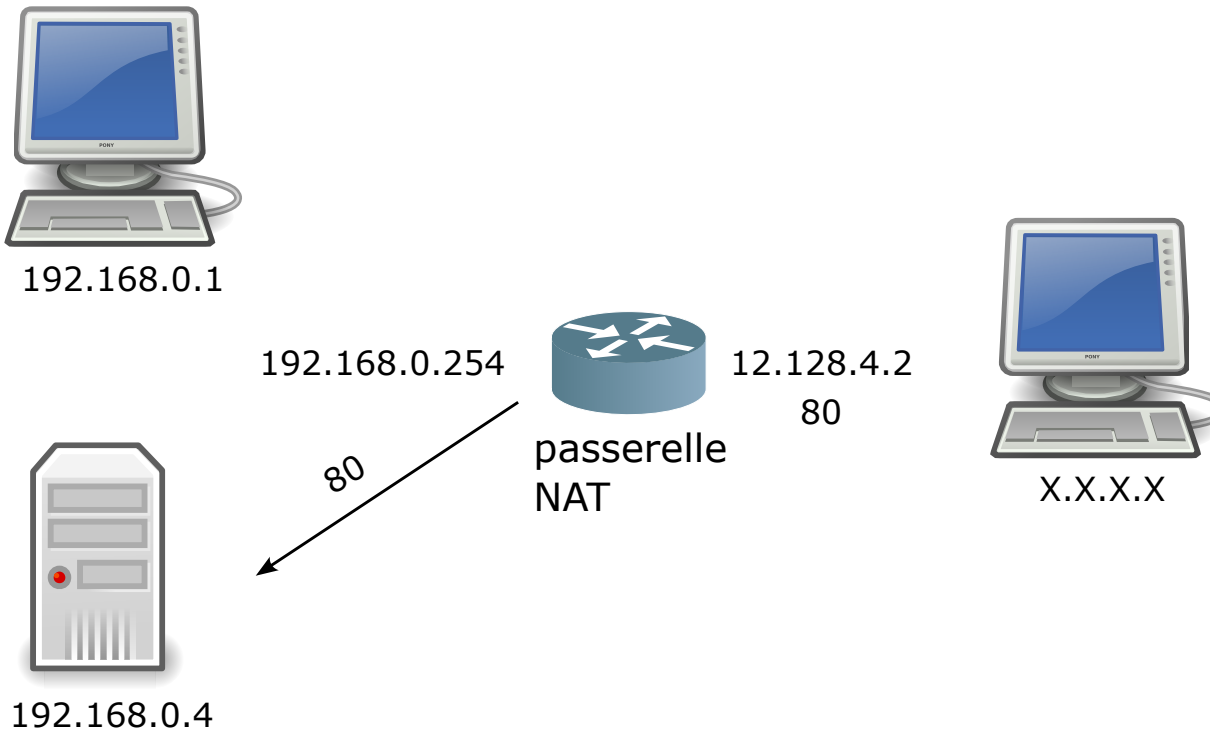


Figure – Port forwarding

Port mapping

- le routeur NAT redirige le trafic extérieur reçu sur un port spécifique vers une adresse IP interne sur un port différent
- un client externe accède à un serveur virtuellement placé sur la passerelle NAT et physiquement placé sur une machine interne

Port mapping

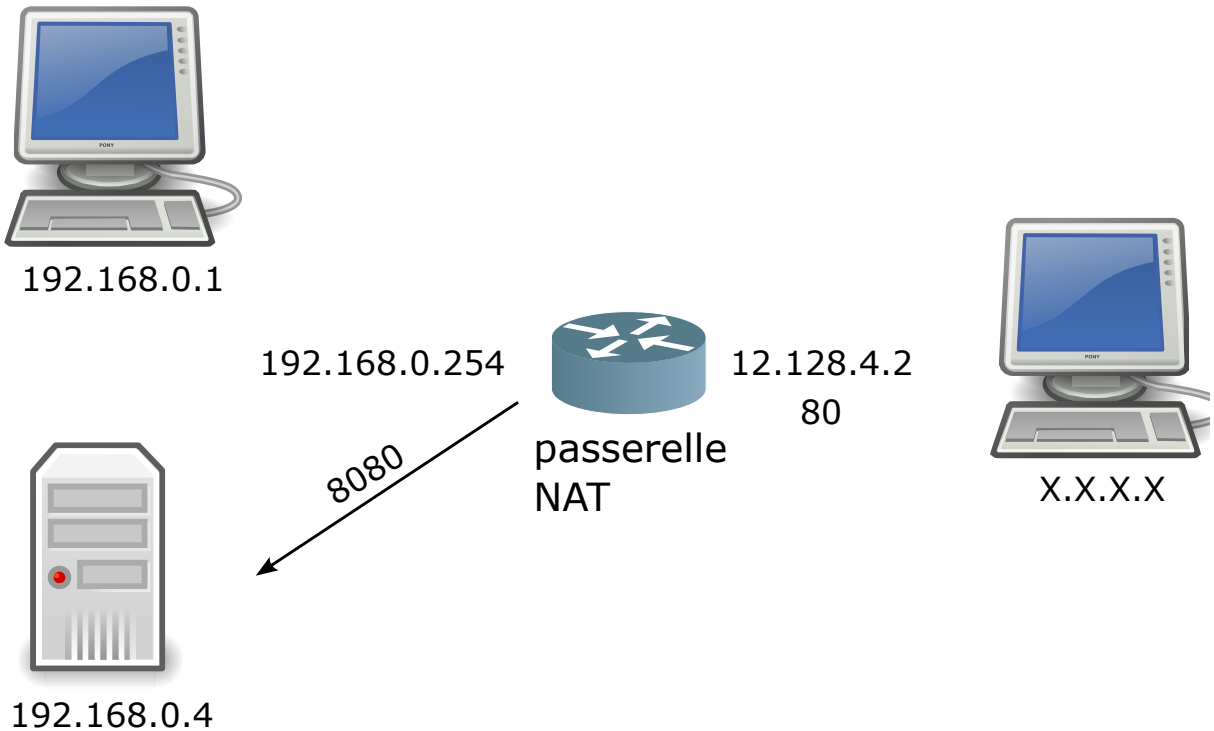


Figure – Port mapping

1. Modèles
2. Couches basses
3. Couche réseau
   - Configuration IP
   - Routage
4. Couche transport
5. Services réseaux
   - Web
   - Firewall
   - NAT
   - DNS

DNS : *Domain Name Service*

- les machinent communiquent via des adresses IP (193.52.104.60)
- les humains préfèrent les noms (ex : `www.univ-nantes.fr`)

$\Rightarrow$ besoin de faire correspondre **nom** $\leftrightarrow$ **IP**
Solution : DNS : *Domain Name Service*

- service associant un nom et une adresse IP
- équivalent pour Internet de l'annuaire (et de l'annuaire inversé du téléphone)

Généralités

- historique : fichier `hosts.txt` centralisé
- croissance d'Internet $\Rightarrow$ problème de cohérence et de taille du fichier

$\Rightarrow$ DNS RFC 1032, 1033, 1034 et 1035

- Espace de nommage
  - mondial,
  - hiérarchique,
  - indépendant des protocoles et des systèmes de communications sous-jacents
- fournit les protocoles de communication entre serveur de noms et applications clientes.

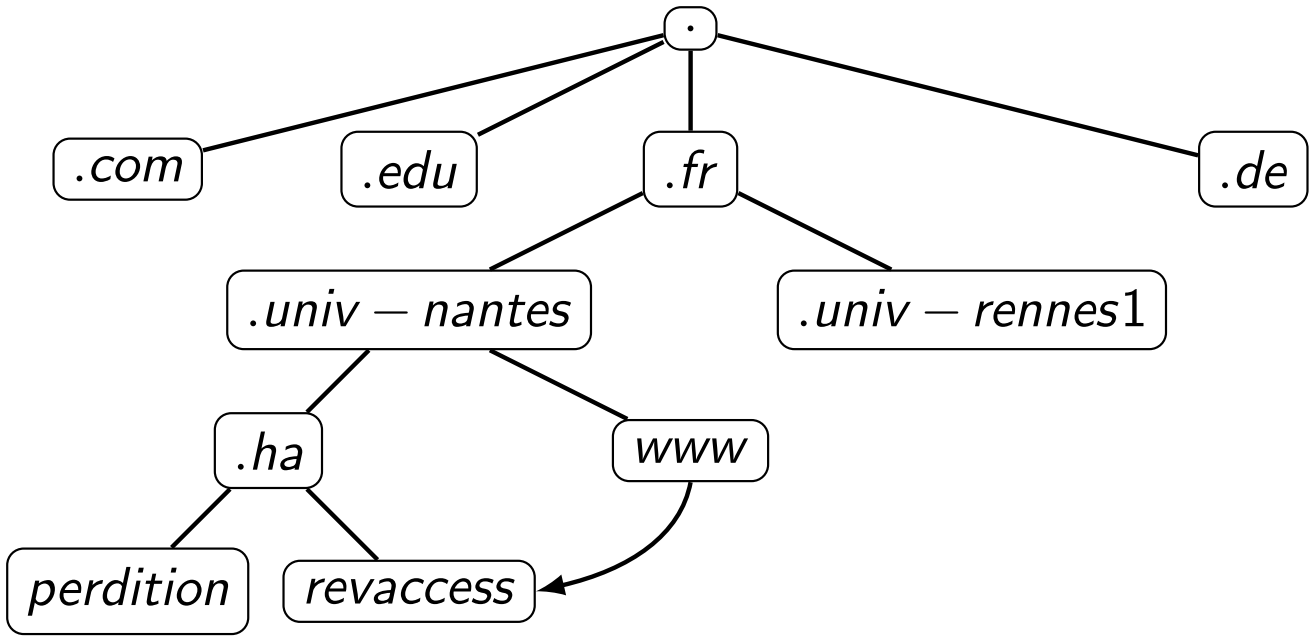
$\Rightarrow$ notion de domaine

$\Rightarrow$ base de données **répartie** mondialement

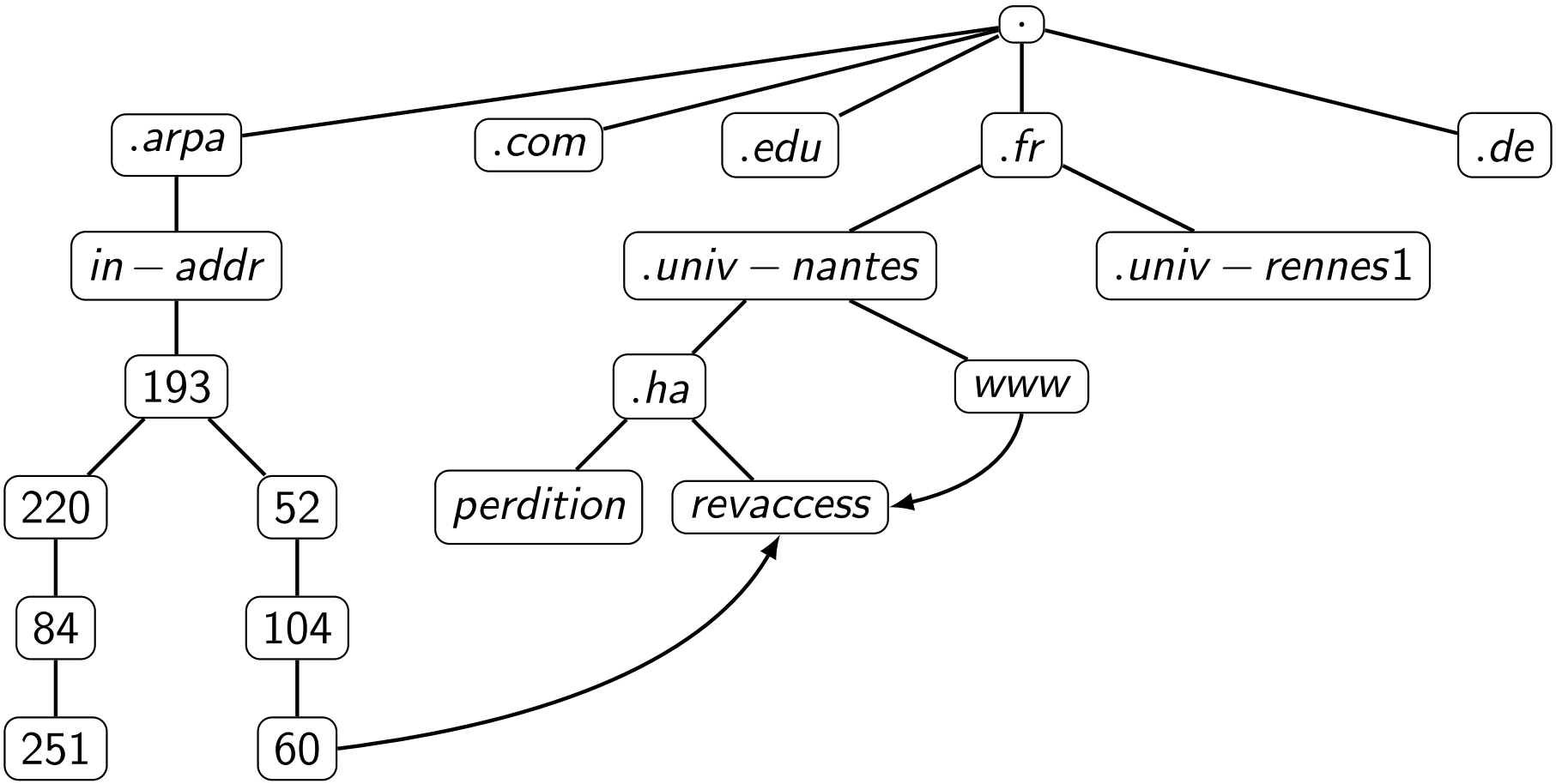
Généralités (suite)

- Espace de nommage hiérarchique :
  - une racine *root* (gérée par l'InterNIC) (13 serveurs DNS !)
  - des noeuds dont le premier niveau est appelé TLD (*Top-Level Domain*) :
    1. "génériques" – gTLD : com, edu, org, mil, net, ...
    2. liés à un pays (env. 200) – ccTLD (*country code* TLD ISO 3166) : fr (géré par le NIC France), de, ...
    3. "nouveaux génériques" – gTLD : biz, info, tel, asia, bzh, sncf, ...
  - des feuilles correspondant aux noms des machines
  - À ces noms sont associés des données : enregistrements
- un nom est de moins de 63 caractères
- gestion décentralisée par délégation.
  exemple : `univ-nantes.fr` géré par l'université de nantes
- informations stockées sur des hôtes appelés serveur de nom
- nom pleinement qualifié *Fully Qualified Domain Name* (FQDN) : nom d'hôte *hostname* + *domain name*

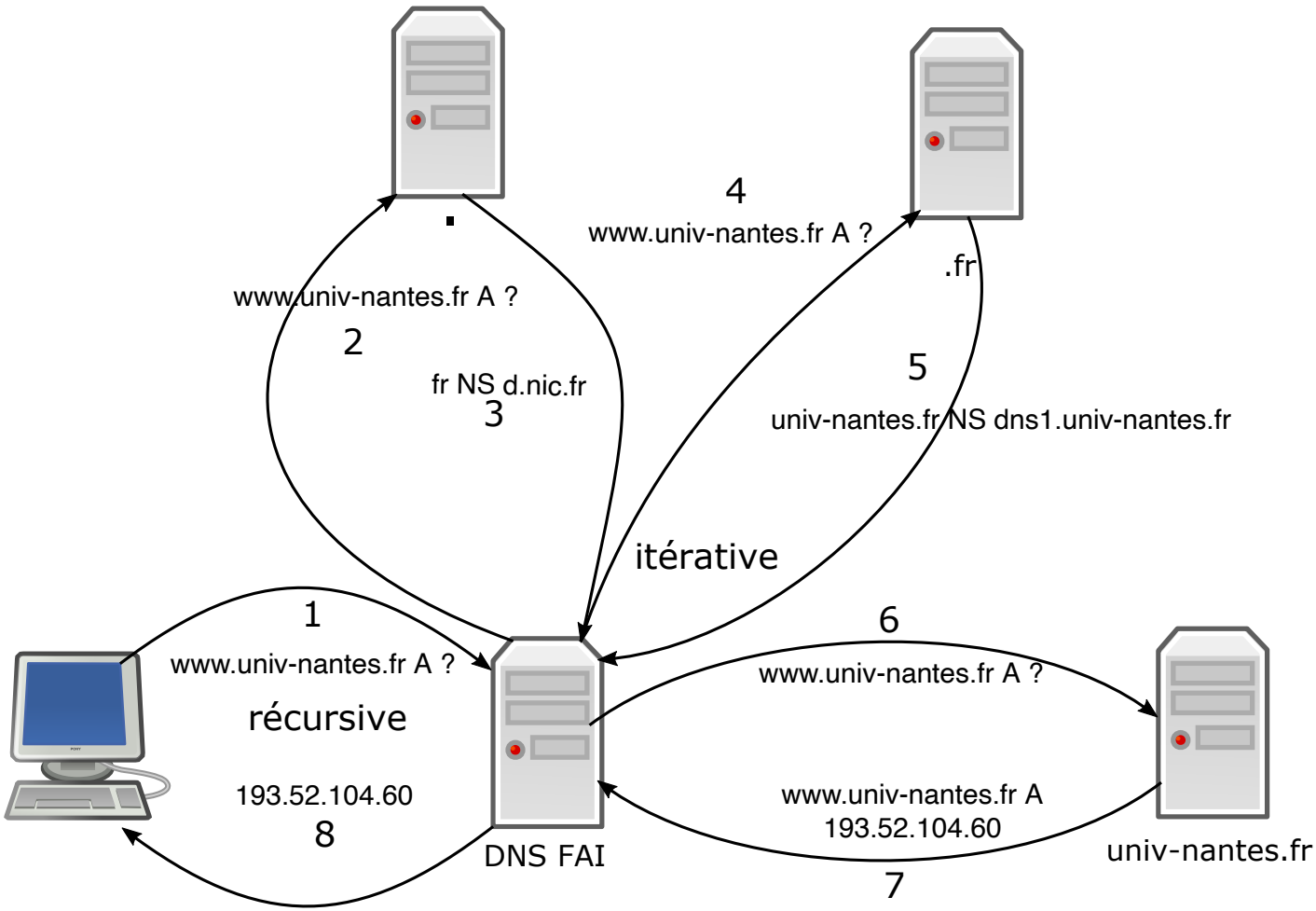
Généralités (suite)



Généralités (suite)



Principe de résolution



Principe de résolution

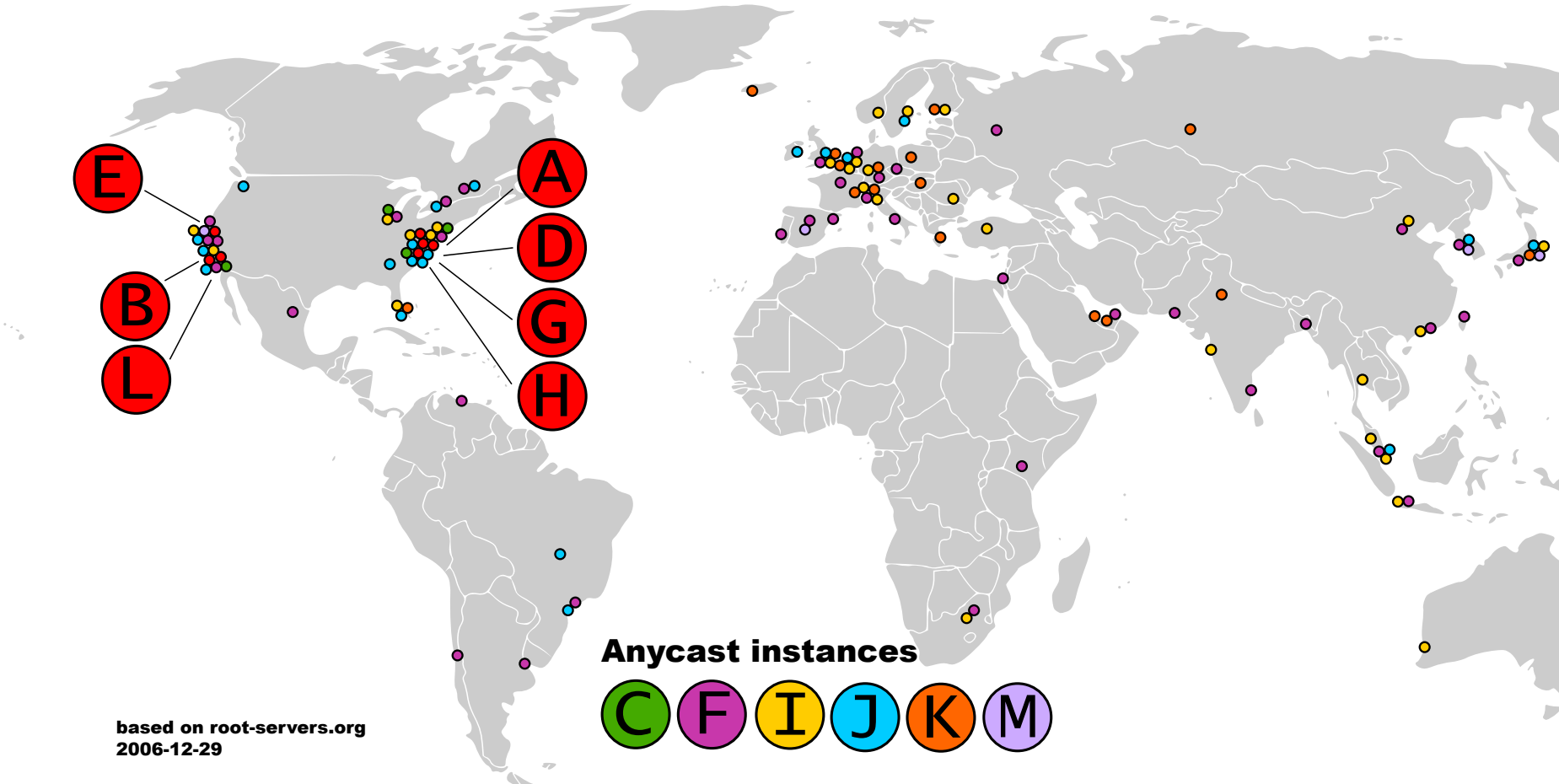
```
dig @192.36.148.17 www.univ-nantes.fr +trace
; <<>> DiG 9.8.3-P1 <<>> @192.36.148.17 www.univ-nantes.fr +trace
.   518400  IN  NS  m.root-servers.net.
... 13 serveurs racines
.   518400  IN  NS  b.root-servers.net.
;; Received 508 bytes from 192.36.148.17#53 in 2480 ms

fr.   172800  IN  NS  d.ext.nic.fr.
fr.   172800  IN  NS  d.nic.fr.
fr.   172800  IN  NS  e.ext.nic.fr.
fr.   172800  IN  NS  f.ext.nic.fr.
fr.   172800  IN  NS  g.ext.nic.fr.
;; Received 344 bytes from 198.97.190.53#53 in 583 ms

univ-nantes.fr. 172800  IN  NS  dns2.univ-nantes.fr.
univ-nantes.fr. 172800  IN  NS  dns1.univ-nantes.fr.
univ-nantes.fr. 172800  IN  NS  resone.univ-rennes1.fr.
univ-nantes.fr. 172800  IN  NS  ufc.univ-fcomte.fr.
;; Received 202 bytes from 193.176.144.22#53 in 376 ms

www.univ-nantes.fr. 345600  IN  CNAME revaccess.ha.univ-nantes.fr.
revaccess.ha.univ-nantes.fr. 345600 IN  A 193.52.104.60
univ-nantes.fr. 345600  IN  NS  ufc.univ-fcomte.fr.
univ-nantes.fr. 345600  IN  NS  resone.univ-rennes1.fr.
univ-nantes.fr. 345600  IN  NS  dns1.univ-nantes.fr.
univ-nantes.fr. 345600  IN  NS  dns2.univ-nantes.fr.
;; Received 241 bytes from 193.52.101.20#53 in 74 ms
```

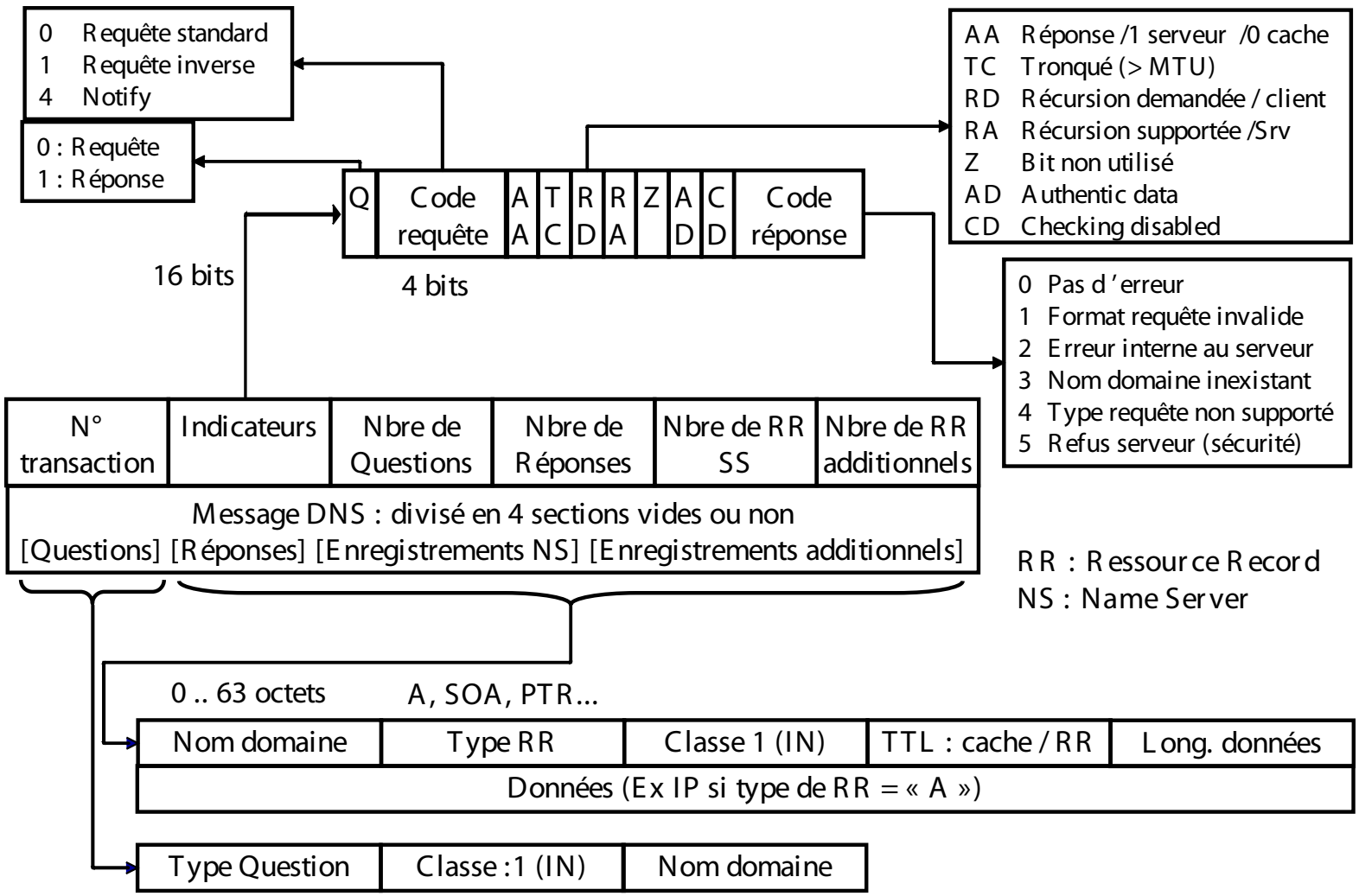
Serveurs racines



Principes

- noeud de l'arbre : domaine
- chemin jusqu'au noeud : nom de domaine
- zone : point de délégation de l'arbre
- un domaine est partitionné en zones
- les données d'une zone sont données sous forme de RR (*Ressource Record*) :
  1. SOA : *Start Of Authority* informations générales de la zone
  2. NS : *Name Server* serveurs DNS du domaine
  3. A : *Address* (ou AAAA en IPv6) nom $\rightarrow$ IP
  4. PTR : *PoinTeR* IP $\rightarrow$ nom
  5. MX : *Mail eXchanger*
  6. SRV : *SeRVer* généralisation de MX
  7. ...

Messages DNS



Requête DNS

```
Internet Protocol, Src: 192.168.2.100 (192.168.2.100), Dst: 192.168.2.254 (192.168.2.254)
User Datagram Protocol, Src Port: 1032 (1032), Dst Port: domain (53)
Domain Name System (query)
    Transaction ID: 0x10b0
    Flags: 0x0100 (Standard query)
        0... .... .... .... = Response: Message is a query
        .000 0... .... .... = Opcode: Standard query (0)
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... .0.. .... = Z: reserved (0)
        .... .... ...0 .... = Non-authenticated data OK: Non-authenticated data is unaccept
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.google.fr: type A, class IN
            Name: www.google.fr
            Type: A (Host address)
            Class: IN (0x0001)
```

Réponse DNS

```
Internet Protocol, Src: 192.168.2.254 (192.168.2.254), Dst: 192.168.2.100 (192.168.2.100)
User Datagram Protocol, Src Port: domain (53), Dst Port: 1032 (1032)
Domain Name System (response)
    Transaction ID: 0x10b0
    Flags: 0x8180 (Standard query response, No error)
        1... .... .... .... = Response: Message is a response
        .000 0... .... .... = Opcode: Standard query (0)
        .... .0.. .... .... = Authoritative: Server is not an authority for domain
        .... ..0. .... .... = Truncated: Message is not truncated
        .... ...1 .... .... = Recursion desired: Do query recursively
        .... .... 1... .... = Recursion available: Server can do recursive queries
        .... .... .0.. .... = Z: reserved (0)
        .... .... ..0. .... = Answer authenticated: Answer/authority portion was not authe
        .... .... .... 0000 = Reply code: No error (0)
    Questions: 1
    Answer RRs: 4
    Authority RRs: 0
    Additional RRs: 0
    Queries
        www.google.fr: type A, class IN
            Name: www.google.fr
            Type: A (Host address)
            Class: IN (0x0001)
```

Réponse DNS

```
Answers
    www.google.fr: type CNAME, class IN, cname www.google.com
        Name: www.google.fr
        Type: CNAME (Canonical name for an alias)
        Class: IN (0x0001)
        Time to live: 10 hours, 22 minutes, 31 seconds
        Data length: 16
        Primary name: www.google.com
    www.google.com: type CNAME, class IN, cname www.l.google.com
        Name: www.google.com
        Type: CNAME (Canonical name for an alias)
        Time to live: 6 days, 23 hours, 51 minutes, 9 seconds
        Data length: 8
        Primary name: www.l.google.com
    www.l.google.com: type A, class IN, addr 209.85.129.104
        Name: www.l.google.com
        Type: A (Host address)
        Time to live: 1 minute, 2 seconds
        Data length: 4
        Addr: 209.85.129.104
    www.l.google.com: type A, class IN, addr 209.85.129.147
        Name: www.l.google.com
        Type: A (Host address)
        Time to live: 1 minute, 2 seconds
        Data length: 4
        Addr: 209.85.129.147
```

Configuration client

1. *Name Service Switch* fichier `/etc/nsswich.conf`

```
passwd: files ldap
group: files ldap
hosts: files dns
```

2. `file` fichier `/etc/hosts`

```
127.0.0.1       localhost
192.168.1.10    toto.mondomaine.org    toto
192.168.1.13    titi.mondomaine.org    titi
216.234.231.5   master.debian.org      master
205.230.163.103 www.opensource.org
```

3. fichier `/etc/resolv.conf`

`nameserver` : ip du serveur de nom
`search` : liste de recherche pour les noms d'hôte.

```
search univ-nantes.fr
nameserver 192.52.109.10
nameserver 172.20.12.11
nameserver 172.20.12.22
nameserver 172.20.12.23
nameserver 8.8.8.8
```

Configuration serveur

- les configurations DNS peuvent prendre plusieurs formes :
  1. serveur cache : ne sert pas de zone DNS officielle, mais met en cache les requêtes DNS des clients
  2. serveur DNS primaire, servant un domaine DNS (zone)
  3. serveur DNS secondaire, fonctionnant en secours du DNS primaire pour la zone concernée
- seuls les serveurs primaires et secondaires font autorité sur leur zone.
- ils sont référencés de la même manière lors des délégations

Configuration serveur

- implantation de référence du protocole DNS : bind ()
- il en existe d'autres : djbdns, dbndns, Microsoft DNS Server, powerDNS, ...
- configuration :
  1. configuration globale dans `named.conf` ; propre à bind
  2. fichiers de zones : syntaxe définie dans la RFC 1035

named.conf

- options globales du serveur :

```
option {
  <option>;
  [<option>;]
}
```

- en particulier :
  - `directory` : répertoire des fichiers de zone (sinon path absolu)
  - `allow-query` : autorisation interrogation serveur
  - `blackhole` : interdiction accès serveur
- exemple :

```
options {
   directory "/var/named";
   pid-file "/var/run/named/named.pid";
};
```

named.conf

- déclaration de zone :

```
zone <nomZone> {
  <optionsZone>;
  [<optionsZone>;]
}
```

- options :
  - `type` : `hint` ou `master` ou `slave`
  - `file` : emplacement fichier

```
zone "." in {
   type hint;
   file "db.cache";
};

zone "mazone.fr" in {
   type master;
   file "mazone.zone";
};
```

```
zone "autre.fr" in {
   type slave;
   file "autre.zone";
   masters {195.15.12.10;}
};

// resolution inverse
zone "18.168.192.in-addr.arpa" in {
   type master;
   file "192.168.18.rev";
};
```

Fichiers de zone

- directives :
  - `$ORIGIN` nom de domaine pour les noms relatifs
  - `$TTL` valeur par défaut de la durée de vie des informations de la zone
- RR (*Ressource Record*) : les enregistrements (`SOA`, `NS`, `A`, `AAAA`, `CNAME`, `MX`, `SRV`, ...)
- commentaires ;

Description d'une ressource

- les enregistrements d'une zone sont donnés ainsi :
  `<name> [<ttl>] [<class>] <type> <data>`
  - `name` : le nom du champs sur lequel s'applique l'enregistrement. Si laissé blanc, nom du précédent enregistrement.
  - `ttl` : Time-to-live temps de conservation en cache.
  - `class` : spécifie le groupe de protocole. `IN` INternet ou `CH` CHaos.
  - `type` : type de donnée de l'enregistrement ; `SOA`, `A`, `MX`, ...
  - `data` : la donnée

Type des ressources

- `SOA` (*Start Of Authority*) : désigne l'autorité = serveur de nom maître
  - il n'y a qu'un seul enregistrement SOA par zone
  - `<name> [<ttl>] [<class>] SOA <origin> <person> ( <serial> <refresh> <retry> <expire> <minimum> )`
    - `name` : nom de la zone.
    - `origin` : nom de l'hôte maître de la zone
    - `person` : personne responsable (qui@ou remplacé par qui.ou)
    - `serial` : numéro de version (ex : 2006112002 version 2 du 20 novembre 2006)
    - `refresh` : Temps d'attente du serveur secondaire avant de contrôler si le serveur primaire a subi une modification au niveau de sa zone.
    - `retry` : Temps d'attente du serveur secondaire avant de faire à nouveau une demande si le serveur primaire n'a pas répondu à une requête
    - `expire` : Temps pendant lequel le serveur secondaire va conserver les données en cache. Si ce délai est dépassé, les données sont obsolètes et le secondaire cessera de servir tant qu'il n'aura pas réussi à contacter le primaire.
    - `minimum` : Valeur par défaut de ttl des enregistrements.

Type des ressources

- `NS` (*Name Server*) : défini les serveurs de noms pour une zone.
  - les enregistrements NS pour un domaine existent dans la zone opérant dans la délégation et dans le domaine lui-même
  - il y a plusieurs enregistrements NS par zone
  - `<domain> [<ttl>] [<class>] NS <server>`
  - exemple :

```
IN NS  ns1.fai.com.
IN NS  ns2
IN NS  ns.chezcopain.com.
```

  - exemple dans le cas d'une délégation (tiré de la zone racine)

```
FR. NS A.EXT.NIC.FR.
FR. NS C.EXT.NIC.FR.
FR. NS C.NIC.FR.
FR. NS A.NIC.FR.
FR. NS B.NIC.FR.
FR. NS D.EXT.NIC.FR.
FR. NS E.EXT.NIC.FR.
FR. NS E.NIC.FR.
FR. NS B.EXT.NIC.FR.
```

  - il faut une entrée `A` pour le serveur de nom
  - sauf en cas de délégation à un serveur de noms de ce domaine : *glue records*

Type des ressources

- `A` (*Address*) : défini l'adresse d'une machine.
  - il y a un seul enregistrement `A` par hôte, il y a autant de `A` que d'hôte dans la zone.
  - `<host> [<ttl>] [<class>] A <address>`
  - exemple :

```
A.ROOT-SERVERS.NET.  A  198.41.0.4
unemachine  A  195.18.12.20
```

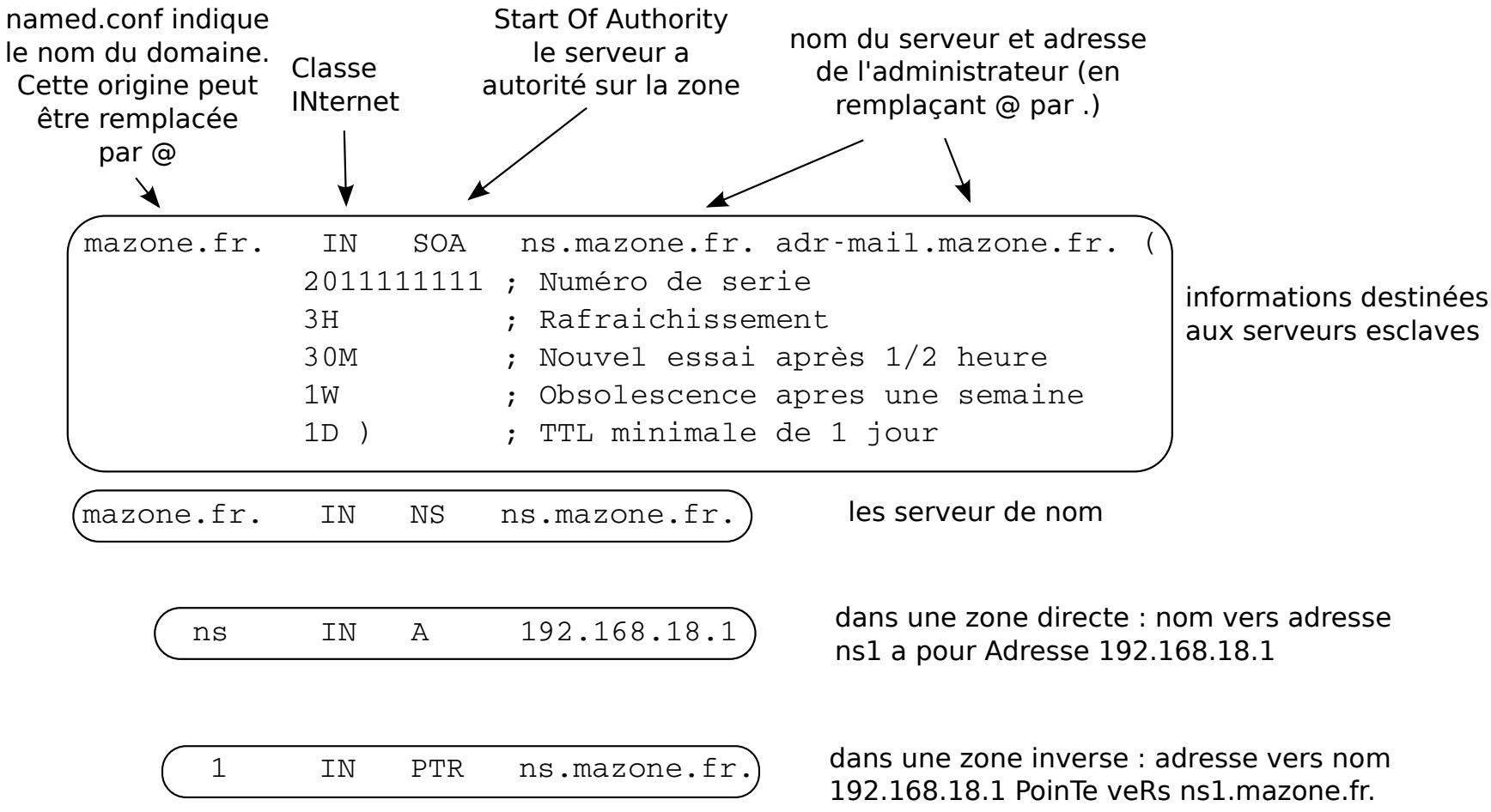
- `CNAME` (*Canonical Name*) : alias
  - `<nickname> [<ttl>] [<class>] CNAME <host>`
  - exemple :

```
www   CNAME unemachine
```

- `MX` (*Mail eXchanger*) : défini où les mails d'un domaine doivent être envoyés.
  - il peut y avoir plusieurs enregistrements `MX`
  - `<host> [<ttl>] [<class>] A <address>`
  - la préférence spécifie l'ordre dans lequel les serveurs de mails doivent être contactés (préférence croissante).
  - exemple :

```
domaine.fr.     MX    10    smtp.domaine.fr.
                MX    20    smtp2.domaine.fr.
```

Description d'une zône



Zone directe

```
$ttl 38400
@   IN   SOA   ns1.mazone.fr. adr-mail.mazone.fr. (
      1      ; Numero de serie
      3H     ; Rafraichissement
      30M    ; Nouvel essai apres 1/2 heure
      1W     ; Obsolescence après 1 semaine
      1D )   ; TTL minimale de 1 jour
; serveurs de noms pour la zone
   IN   NS   ns1.mazone.fr.
   IN   NS   ns2
   IN   NS   ns.chezcopain.com. ; pas de A

; delegation
sous.mazone.fr.   IN   NS   ns.sous.mazone.fr.
sous.mazone.fr.   IN   NS   ns.autre.fr. ; pas de A

; mail
mazone.fr.      IN   MX  10   smtp

; adresses
ns1             IN   A   192.168.18.1
ns2.mazone.fr.  IN   A   192.168.18.2
ns.sous         IN   A   192.18.12.12 ; "Glue record"
asterix         IN   A   192.168.18.100
;alias
smtp            IN   CNAME   asterix
```

Zone directe

```
$ttl 38400
mazone.fr. IN  SOA  ns1.mazone.fr. adr-mail.mazone.fr. (
      1     ; Numero de serie
      3H    ; Rafraichissement
      30M   ; Nouvel essai apres 1/2 heure
      1W    ; Obsolescence après 1 semaine
      1D )  ; TTL minimale de 1 jour
; serveurs de noms pour la zone
mazone.fr.   IN   NS   ns1.mazone.fr.
mazone.fr.   IN   NS   ns2
mazone.fr.   IN   NS   ns.chezcopain.com. ; pas de A

; delegation
sous.mazone.fr.   IN   NS   ns.sous.mazone.fr.
sous.mazone.fr.   IN   NS   ns.autre.fr. ; pas de A

; mail
mazone.fr.     IN   MX  10   smtp

; adresses
ns1            IN   A   192.168.18.1
ns2.mazone.fr. IN   A   192.168.18.2
ns.sous        IN   A   192.18.12.12 ; "Glue records"
asterix        IN   A   192.168.18.100
;alias
smtp           IN   CNAME   asterix
```

Zone reverse

```
$ttl 38400
$ORIGIN 18.168.192.in-addr.arpa.
@   IN   SOA   ns1.mazone.fr. postmaster.mazone.fr. (
      2006111201     ; Numero de serie
      3H    ; Rafraichissement
      30M   ; Nouvel essai apres 1/2 heure
      1W    ; Obsolescence après 1 semaine
      1D )  ; TTL minimale de 1 jour

      IN    NS    ns1.mazone.fr.
      IN    NS    ns2.mazone.fr.
      IN    NS    ns.chezcopain.com.

1     IN    PTR   ns1.mazone.fr.
2     IN    PTR   ns2.mazone.fr.
100   IN    PTR   asterix.mazone.fr.
```

Zone locale

```
$TTL 86400
$ORIGIN localhost.
@      1D     IN     SOA    @  root (
       42 ; serial (d. adams)
       3H ; refresh
       15M ; retry
       1W ; expiry
       1D ) ; minimum

   1D IN NS @
   1D IN A 127.0.0.1
```

Configuration avancée

- limiter le plus possible les accès au serveur
- vues : permet de délivrer des informations différentes en fonction des adresses IP des clients
- acl : pour regrouper une politique des droits

Configuration avancée : acl

- utilisation des acls

```
acl bogusnets { // RFC1918 + multicast + expérimental
  0.0.0.0/8;
  1.0.0.0/8;
  2.0.0.0/8;
  192.0.2.0/24;
  224.0.0.0/3;
  192.168.0.0/16; };

acl "xfer" { // ne mettre ici que les serveurs de noms
  193.52.108.41; // dns1.univ-nantes.fr
  193.52.101.20; // dns2.univ-nantes.fr
  129.20.254.1; // resone.univ-rennes1.fr
  172.20.12.11; // dns1.univ-nantes.prive
  172.20.12.22; // dns2.univ-nantes.prive
  172.20.12.23; // dns3.univ-nantes.prive
};

acl "trusted" { // le réseaux local
  172.16.0.0/16;
  localhost;
};
```

Configuration avancée : options

```
options {
    directory "/var/named";
    pid-file "/var/named/named.pid";

    notify no; // les secondaires viennent d'eux même
   // ou notify explicit; + also-notify pour les prévenir tout de suite

    transfer-format many-answers; // transfert de zone + efficace

    allow-transfer {xfer;}; // transfert de zone limitée aux secondaires

    allow-query {trusted;};// par défaut seul les locaux

    blackhole {bogunets;}; // aucun accès !!
};
```

Configuration avancée : vues

```
view "interne" in {
    match-clients { trusted; }; // les clients autorisés
    recursion yes;
      zone "." in {
          type hint;
          file "db.cache";
      };

      zone "0.0.127.in-addr.arpa" in {
          type master;
          file "master/127.0.0.rev";
          allow-query {any;};
          allow-transfer {none;};
      };

      zone "interne.mazone.com" in { // zone directe interne
          type master;
          file "master/interne.zone";
      };

      zone "9.16.172.in-addr.arpa" in {//zone reverse interne
          type master;
          file "master/172.16.9.rev";
      };
};
```

Configuration avancée : vues

```
view "externe" in {
    match-clients { any; };
    recursion no;

      zone "." in {
          type hint;
          file "db.cache";
      };

      zone "mazone.com" in {
          type master;
          file "master/mazone.zone";
          allow-query {any;};
      };

      zone "84.220.195.in-addr.arpa" in {
          type master;
          file "master/195.220.84.rev";
          allow-query {any;};
      };
};
```

Configuration d'un serveur cache

```
options {
      directory "/var/named";
};
zone "." IN {
      type hint;
      file "named.ca";
};
zone "localhost" IN {
      type master;
      file "localhost.zone";
      allow-update { none; };
};
zone "0.0.127.in-addr.arpa" IN {
      type master;
      file "named.local";
      allow-update { none; };
};
```

Outils

- `nslookup` : commande vénérable
- `dig`, `host` : dig avec une sortie plus brute favorisant le débogage ; host avec une sortie plus "user friendly"
- `zonecheck` sur le site de l'afnic (centre d'information et de gestion des noms de domaine internet .fr (France) et .re (Île de la Réunion)) : `http://www.afnic.fr/outils/zonecheck`