

# Technologies de l'Internet

## Module TR2

Laure Petrucci

IUT R&T Villetaneuse

28 janvier 2008

# Plan du cours

- 1 Modèles en couches
- 2 TCP/IP — Internet
- 3 Filtrage — iptables
- 4 Noms de domaines — DNS
- 5 Transfert de fichiers — FTP
- 6 Gestion du courrier — SMTP
- 7 Annuaires — LDAP
- 8 RPC — Appels de procédure à distance

- 1 Modèles en couches
  - Architecture OSI
  - SDU, PDU et encapsulation

# Le modèle OSI (Open System Interconnection)

## Un modèle normalisé

- **norme** de l'ISO (International Standards Organisation)
- assurer une **compatibilité** entre entités hétérogènes

## Une structuration en couches

Chaque couche :

- assure un **rôle spécifique**
- dialogue avec les **couches adjacentes** de la **même entité**
- dialogue avec la **couche de même niveau** de l'**autre entité**

# Le modèle OSI (Open System Interconnection)

## Un modèle normalisé

- norme de l'ISO (International Standards Organisation)
- assurer une **compatibilité** entre entités hétérogènes

## Une structuration en couches

Chaque couche :

- assure un **rôle spécifique**
- dialogue avec les **couches adjacentes** de la **même entité**
- dialogue avec la **couche de même niveau** de **l'autre entité**

# Les couches du modèle OSI

Entité A

7.	Application
6.	Présentation
5.	Session
4.	Transport
3.	Réseau
2.	Liaison de données
1.	Physique

# Les couches du modèle OSI

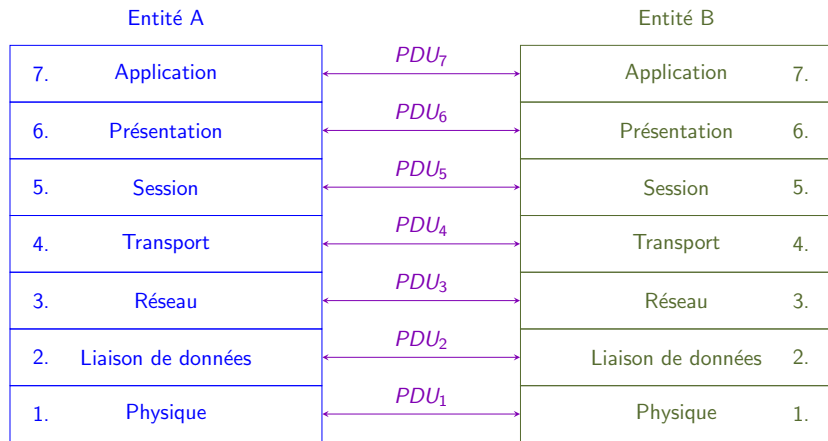
Entité A

7.	Application
6.	Présentation
5.	Session
4.	Transport
3.	Réseau
2.	Liaison de données
1.	Physique

Entité B

Application	7.
Présentation	6.
Session	5.
Transport	4.
Réseau	3.
Liaison de données	2.
Physique	1.

# Les couches du modèle OSI



**PDU — Protocol Data Unit**

$PDU_n$  : protocole d'échange entre deux couches de niveau  $n$

# Les couches du modèle OSI

Entité A

7.	Application
6.	Présentation
5.	Session
4.	Transport
3.	Réseau
2.	Liaison de données
1.	Physique

*SDU<sub>6</sub>*

Entité B

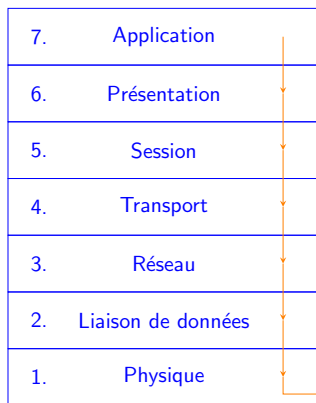
Application	7.
Présentation	6.
Session	5.
Transport	4.
Réseau	3.
Liaison de données	2.
Physique	1.

**SDU — Service Data Unit**

*SDU<sub>n</sub>* : service fourni par la couche *n* à la couche *n + 1*

# Les couches du modèle OSI

Entité A



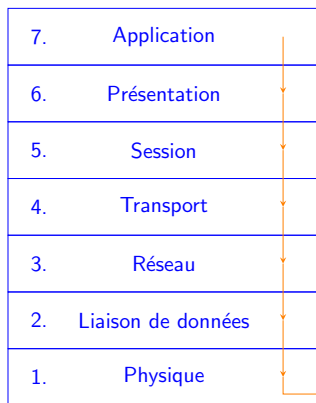
Entité B



Request

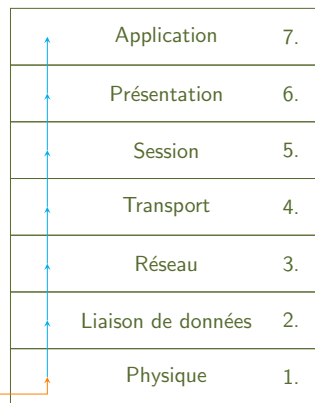
# Les couches du modèle OSI

Entité A



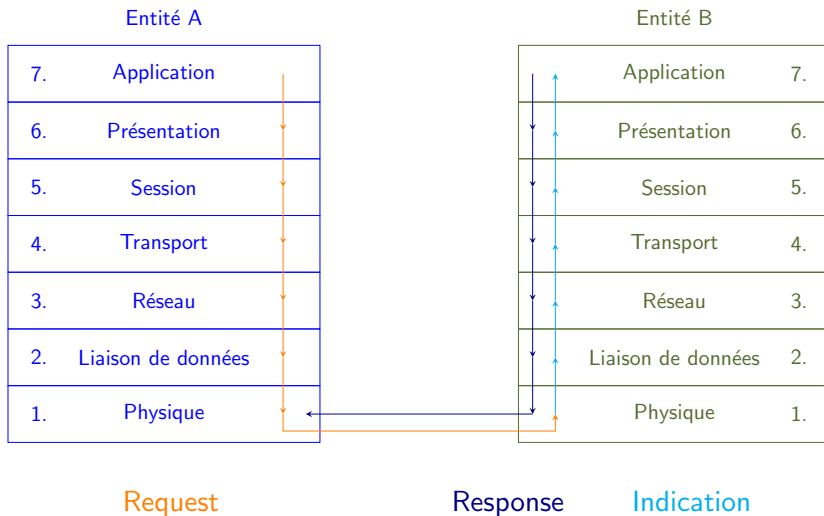
Request

Entité B

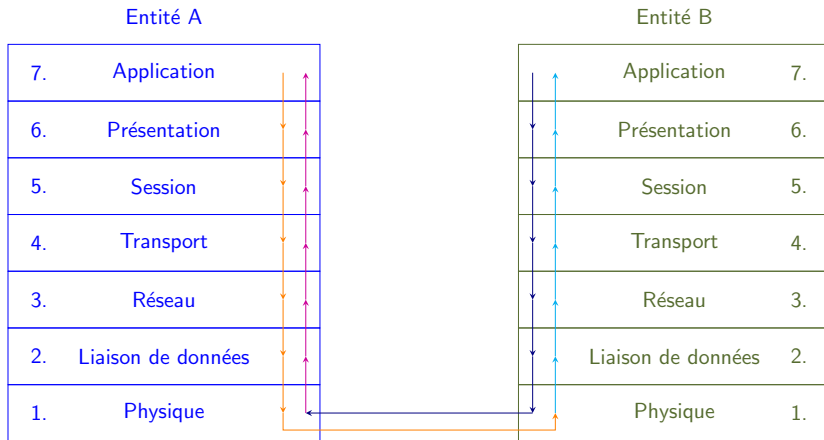


Indication

# Les couches du modèle OSI



# Les couches du modèle OSI



Request

Confirm

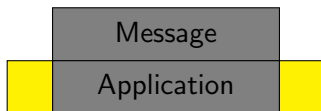
Response

Indication

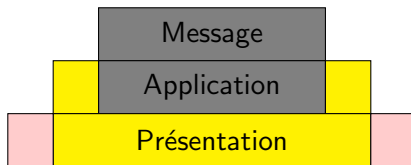
# Encapsulation

Message

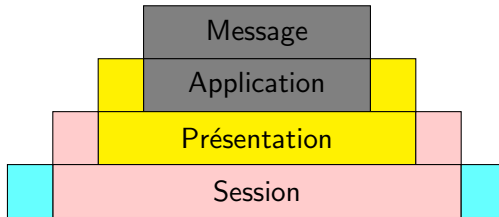
# Encapsulation



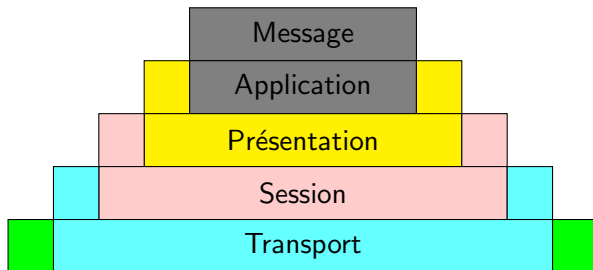
# Encapsulation



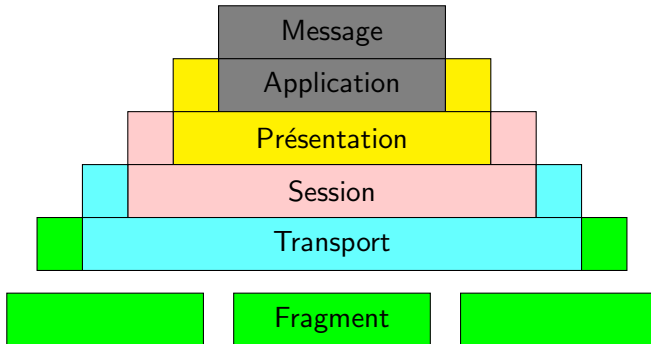
# Encapsulation



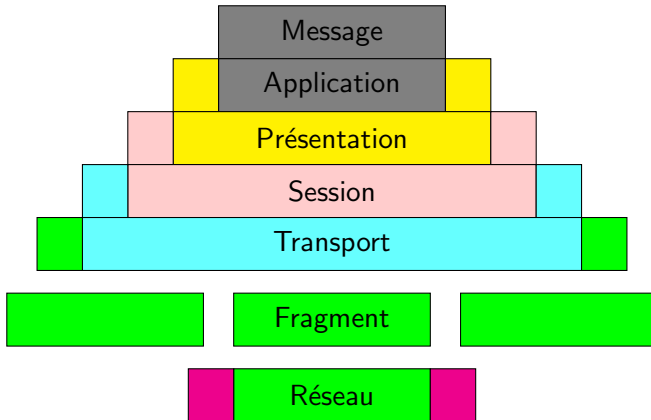
# Encapsulation



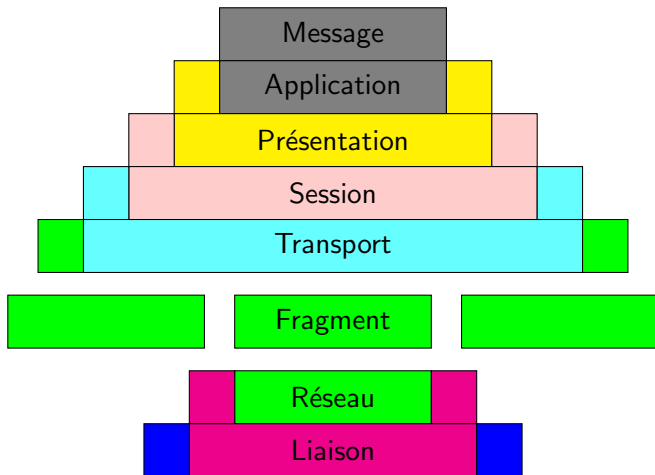
# Encapsulation



# Encapsulation



# Encapsulation



## 2 TCP/IP — Internet

- Petit historique
- Architecture du DoD (Internet)
- Structure d'un paquet IP
- Adresses IP
- Routage internet
  - Table de routage
  - Sous-adressage
- Protocoles
  - ARP — Address Resolution Protocol
  - RARP — Reverse Address Resolution Protocol
  - ICMP — Internet Control Message Protocol
  - IGMP — Internet Group Management Protocol
  - TCP — Transmission Control Protocol
  - UDP — User Datagram Protocol

# Petit historique

- Développé par le DARPA (Defence Advanced Research Project Agency), à la demande du DoD (Department of Defense) américain.
- 1970 : interconnexion des réseaux américains
- développement/coordination assurés par :
  - 1979 – 1983 : ICCB (Internet Control and Configuration Board)
  - 1983 – 1989 : IAB (Internet Activities Board)
  - 1989 – ... : IRTF (Internet Research Task Force) et IETF (Internet Engineering Task Force)
- pas de norme, mais des RFC (Request For Comments)

# Architecture Internet

## Architecture OSI

7.	Application
6.	Présentation
5.	Session
4.	Transport
3.	Réseau
2.	Liaison de données
1.	Physique

## Internet



# Protocoles

## Processus

Telnet, FTP, SMTP, NFS, SNMP, Ping, ...

## Hôte à hôte

TCP, UDP, ICMP

## Internet

IP, ARP, RARP

## Accès réseau

FDDI, X25, Xerox Ethernet, IEEE 802 (couches LLC et MAC), Arpanet,  
...

# Protocoles

## Processus

Telnet, FTP, SMTP, NFS, SNMP, Ping, ...

## Hôte à hôte

TCP, UDP, ICMP

## Internet

IP, ARP, RARP

## Accès réseau

FDDI, X25, Xerox Ethernet, IEEE 802 (couches LLC et MAC), Arpanet,

...

# Protocoles

## Processus

Telnet, FTP, SMTP, NFS, SNMP, Ping, ...

## Hôte à hôte

TCP, UDP, ICMP

## Internet

IP, ARP, RARP

## Accès réseau

FDDI, X25, Xerox Ethernet, IEEE 802 (couches LLC et MAC), Arpanet,  
...

# Protocoles

## Processus

Telnet, FTP, SMTP, NFS, SNMP, Ping, ...

## Hôte à hôte

TCP, UDP, ICMP

## Internet

IP, ARP, RARP

## Accès réseau

FDDI, X25, Xerox Ethernet, IEEE 802 (couches LLC et MAC), Arpanet,  
...

# Structure d'un paquet IP

v. IP (4)	lg. entête (4)	DSCP (6)	ECN (2)	lg. tot. en octets (16)			
Identification (16)				0	DF	MF	offset (13)
Durée de vie — TTL (8)		protocole (8)		ctrl. erreur entête (16)			
Adresse IP émetteur (32)							
Adresse IP destinataire (32)							
Options éventuelles							
bourrage éventuel pour les options							
Données							

Les tailles des champs sont en nombre de bits.

- **v. IP** : version du protocole (IPv4, IPv6)
- **lg entête** : longueur de l'entête en paquets de 4 octets.
- **DSCP** (Differentiated Service Code Point) : permet aux routeurs de traiter au mieux le paquet.
- **ECN** (Explicit Congestion Notification) : gestion de congestions
- **DF** (Don't Fragment) : pas de fragmentation
- **MF** (More Fragments) : il y a d'autres fragments
- **Offset** : position du fragment dans le message, en nombre de blocs de 8 octets

# Options

Copie (1)	Classe (2)	Numéro (5)	paramètres éventuels
-----------	------------	------------	----------------------

## Copie

1 impose à une passerelle de **recopier** le champ **options** dans tous les fragments

## Classe

- 0 : datagramme ou supervision réseau
- 2 : mesures et mise au point
- 1, 3 : réservés

## Numéro

Signification de l'action. Par exemple, pour la classe 0 :

- 0 : fin de la liste des options
- 3 : routage approximatif
- 7 : enregistrement de la route
- 9 : routage exact

# Options

Copie (1)	Classe (2)	Numéro (5)	paramètres éventuels
-----------	------------	------------	----------------------

## Copie

1 impose à une passerelle de **recopier** le champ **options** dans tous les fragments

## Classe

- 0 : **datagramme** ou **supervision** réseau
- 2 : **mesures** et mise au point
- 1, 3 : réservés

## Numéro

Signification de l'action. Par exemple, pour la classe 0 :

- 0 : fin de la liste des options
- 3 : routage approximatif
- 7 : enregistrement de la route
- 9 : routage exact

# Caractéristiques de l'adressage IP

- adresses **uniques** (dans le monde)
- **4 octets** :
  - représentés en **décimal**
  - séparés par des points
  - premiers octets : **numéro de réseau**
  - derniers octets : **adresse locale** de l'entité sur le réseau

adresse locale  
194.254.173 .123  
numéro réseau

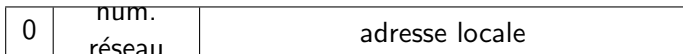
# Caractéristiques de l'adressage IP

- adresses **uniques** (dans le monde)
- **4 octets** :
  - représentés en **décimal**
  - séparés par des points
  - premiers octets : **numéro de réseau**
  - derniers octets : **adresse locale** de l'entité sur le réseau

adresse locale  
194.254.173 .123  
numéro réseau

# Classes d'adresses

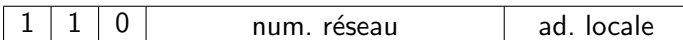
**Classe A** (grands réseaux) :



**Classe B** (réseaux moyens) :



**Classe C** (petits réseaux) :



**Classe D** (utilisé par IGMP) :



# Adresses particulières

- octets de l'adresse locale à 0 : nom de réseau
- octets de l'adresse locale à 255 : adresse de diffusion sur le réseau
- 127.0.0.1 : bouclage local
- numéro de réseau 169.254 : adresses « link-local » pour l'autoconfiguration
- adresses privées

# Obtention de numéro de réseau

Les demandes sont gérées par une autorité centrale, l'**IANA** (Internet Address Network Authority) qui les distribue au **NIC** (Network Information Center) et au RIPE (Réseaux IP Européens).

Peu de numéros sont encore disponibles. On contourne le problème en utilisant le mécanisme de translation d'adresse (**NAT**).

# Gestion des adresses sous UNIX

## Adresses numériques et symboliques

- **adresses symboliques** faciles à retenir. Ex :  
`machine.iutv.univ-paris13.fr`
- **correspondance** **adresse numérique** ↔ **adresse symbolique** :
  - dans le fichier `/etc/hosts`
  - dans la **base de données** d'un serveur **NIS**
  - gérées par un **serveur de noms** (**DNS**)

# Routage dans internet

## Hôte

- table de routage simplifiée
- pas de relai des paquets

## Routeur (passerelle, gateway)

- 2 interfaces réseau
- relai des paquets :
  - table de routage
  - peut décider qu'un paquet ne peut pas être délivré

# Routage dans internet

## Hôte

- table de routage simplifiée
- pas de relai des paquets

## Routeur (passerelle, gateway)

- 2 interfaces réseau
- relai des paquets :
  - table de routage
  - peut décider qu'un paquet ne peut pas être délivré

# Structure d'une table de routage

Destination	Gateway	Netmask	Flags	Interface
...	...	...	...	...

- **Destination** : adresse IP de destination
- **Gateway** : adresse du prochain routeur à emprunter pour atteindre l'adresse, si nécessaire
- **Netmask** : masque (pour le sous-adressage)
- **Flags** :
  - U (in Use) : ligne opérationnelle
  - G (Gateway) : chemin vers un routeur
  - H (Host) : chemin vers un hôte
- **Interface** : interface réseau sur laquelle envoyer le paquet

# Éléments d'une table de routage

Destination	Gateway	Netmask	Flags	Interface
127.0.0.0	—	255.0.0.0	U	lo0
194.254.173.17	—	255.255.255.255	UH	eth0
194.254.173.0	—	255.255.255.0	U	eth0
default	gw.iutv.univ-paris13.fr	0.0.0.0	UG	eth0

- **boucle locale** : interface **lo** (localhost)
- **adresse hôte** (ex : 194.254.173.17)
- **réseau local** sur lequel le paquet est envoyé si le « et » binaire de l'adresse de l'hôte destinataire et du masque est égal à la destination
- accès à un **routeur par défaut**

# Sous-adressage

## Pourquoi ?

- numéro de réseau attribué à une organisation
- subdivision en sous-réseaux (par ex. départements)

## Sous-adressage

Définition d'un masque binaire sur la partie **adresse locale**

Par exemple, l'hôte 182.33.200.35 fait partie du sous-réseau de masque 255.255.240.0 du réseau 182.33.0.0, c'est-à-dire 182.33.192.0, car :

	182.33.200.35	1011 0110.0010 0001.1100 1000.0010 0011
et	255.255.240.0	1111 1111.1111 1111.1111 0000.0000 0000
=	182.33.192.0	1011 0110.0010 0001.1100 0000.0000 0000

# NAT — Network Address Translation

- utilisation d'**adresses privées** sur le réseau local
- seul le routeur est connu de l'extérieur
- le routeur **traduit** les adresses privées

# ARP — Address Resolution Protocol

## Objectif

Trouver une **adresse MAC** à partir d'une **adresse IP**

## Pourquoi ?

Besoin d'adresses MAC Ethernet

## Comment ?

Une machine A veut obtenir l'adresse MAC d'une machine B :

- A envoie un paquet **ARP.request** ( $MAC_A, IP_A, 0, IP_B$ )
- B répond par un paquet **ARP.reply** ( $MAC_B, IP_B, MAC_A, IP_A$ )

# RARP — Reverse Address Resolution Protocol

## Objectif

Trouver une **adresse IP** à partir d'une **adresse MAC**

## Pourquoi ?

- redémarrage d'une station sans disque
- stations rarement utilisées

## Comment ?

- **diffusion** de l'adresse **MAC**
- un **serveur RARP** renvoie l'adresse **IP** correspondante

# ICMP — Internet Control Message Protocol

## Objectif

Contrôler les erreurs sur le réseau

## Pourquoi ?

- réponse à un routage raté
- test de l'état d'une station distante
- test du délai de réponse
- acheminement de messages d'erreur

## Commandes UNIX

ping, traceroute, tracepath

## Commandes WINDOWS

ping, tracert

# IGMP — Internet Group Management Protocol

## Objectif

Diffusion de messages à un groupe **multicast**

## Pourquoi ?

- gérer des **groupes de stations**
- **synchronisation d'horloges**

## Comment ?

- une station s'associe à un groupe en envoyant une requête (groupe, interface)
- le **routeur multicast** diffuse sur les interfaces

**Unicast  $\neq$  Multicast  $\neq$  Broadcast**

# TCP — Transmission Control Protocol

## Objectif

Fournir un **service de transport fiable**

## Comment ?

- indépendant des protocoles inférieurs
- connexions bi-directionnelles
- processus **serveurs** et processus **clients**
- **communication** identifiable de manière **unique** par :
  - adresse source
  - **port** source
  - adresse destination
  - **port** destination

# Structure d'un *segment* TCP

port source (16)		port destination (16)	
numéro de séquence (32)			
numéro d'acquittement (32)			
Lg. entête (4)	réservé (4)	drapeaux (8)	taille fenêtr (16)
checksum (16)		pointeur urgent (16)	
Options éventuelles			
Données			

## Drapeaux

- **CWR**, **ECE** (Congestion Window Reduced) : gestion de la congestion
- **URG** (urgent) : utilisation du pointeur urgent
- **ACK** (acknowledgement) : utilisation du numéro d'acquittement, ou réponse à l'ouverture de connexion
- **PSH** (push) : demande de transmission des données à la couche supérieure
- **RST** (reset) : réinitialisation de la connexion
- **SYN** (synchronise) : synchronisation des numéros de séquence et établissement de la connexion
- **FIN** (finished) : demande de fermeture de la connexion

# Structure d'un *segment* TCP

port source (16)		port destination (16)	
numéro de séquence (32)			
numéro d'acquittement (32)			
Lg. entête (4)	réservé (4)	drapeaux (8)	taille fenêtré (16)
checksum (16)		pointeur urgent (16)	
Options éventuelles			
Données			

## Drapeaux

- **CWR**, **ECE** (Congestion Window Reduced) : gestion de la **congestion**
- **URG** (urgent) : utilisation du **pointeur urgent**
- **ACK** (acknowledgement) : utilisation du **numéro d'acquittement**, ou réponse à l'**ouverture de connexion**
- **PSH** (push) : demande de **transmission** des données à la couche supérieure
- **RST** (reset) : **réinitialisation** de la connexion
- **SYN** (synchronise) : synchronisation des **numéros de séquence** et **établissement** de la connexion
- **FIN** (finished) : demande de **fermeture** de la connexion

# UDP — User Datagram Protocol

- Protocole très simple
- sans connexion
- sans acquittement
- utilisation des numéros de ports

### 3 Filtrage — iptables

- Notion de parefeu
- Différents types de filtrage
- Utilisation d'iptables

# Parefeu (firewall)

## Objectif

parer à des attaques, **protéger** des intrusions

## Pourquoi ?

Machine connectée en permanence ⇒

- pas de surveillance
- large bande passante
- pas de changement d'adresse IP

## Comment ?

- **filtrage** des données échangées avec le réseau
- disposer d'au moins **2 interfaces**
  - réseau interne
  - réseau externe
- en général une **machine dédiée** ou un **matériel spécifique**

# Parefeu (firewall)

## Objectif

parer à des attaques, **protéger** des intrusions

## Pourquoi ?

Machine **connectée en permanence** ⇒

- **pas de surveillance**
- **large** bande passante
- pas de changement d'**adresse IP**

## Comment ?

- **filtrage** des données échangées avec le réseau
- disposer d'au moins **2 interfaces**
  - réseau **interne**
  - réseau **externe**
- en général une **machine dédiée** ou un **matériel spécifique**

# Parefeu (firewall)

## Objectif

parer à des attaques, **protéger** des intrusions

## Pourquoi ?

Machine **connectée en permanence** ⇒

- **pas de surveillance**
- **large** bande passante
- pas de changement d'**adresse IP**

## Comment ?

- **filtrage** des données échangées avec le réseau
- disposer d'au moins **2 interfaces**
  - réseau **interne**
  - réseau **externe**
- en général une **machine dédiée** ou un **matériel spécifique**

# Fonctionnement

## Règles de filtrage

Le filtrage obéit à un ensemble de **règles** permettant de :

- **autoriser** la connexion (**allow**)
- **bloquer** la connexion (**deny**)
- **rejeter** la demande de connexion **sans prévenir l'émetteur** (**drop**)

## Politique de fonctionnement

Plusieurs politiques peuvent être appliquées :

- **autorisations explicites** uniquement : tout ce qui n'est pas autorisé est par défaut interdit ⇒ **contraignant**
- **interdictions explicites** uniquement : tout ce qui n'est pas interdit est autorisé ⇒ **peu sûr**

# Fonctionnement

## Règles de filtrage

Le filtrage obéit à un ensemble de **règles** permettant de :

- **autoriser** la connexion (**allow**)
- **bloquer** la connexion (**deny**)
- **rejeter** la demande de connexion **sans prévenir l'émetteur** (**drop**)

## Politique de fonctionnement

Plusieurs politiques peuvent être appliquées :

- **autorisations explicites** uniquement : tout ce qui n'est pas autorisé est par défaut interdit  $\Rightarrow$  **contraignant**
- **interdictions explicites** uniquement : tout ce qui n'est pas interdit est autorisé  $\Rightarrow$  **peu sûr**

# Filtrage simple (stateless packet filtering)

- **Analyse de l'entête** de chaque datagramme circulant entre les différents réseaux  
⇒ IP source, IP destination, type (TCP, UDP, ...), port
- **Blocage/autorisation** selon le port utilisé
  - ports standard : de 0 à 1023 (25=smtp, 80=http, ...)
  - bloquer les ports non indispensables (23=telnet, connexion non sécurisée)
  - autoriser les ports correspondant à des services indispensables (22=ssh, connexion sécurisée)

# Filtrage dynamique (statefull packet filtering)

## Pourquoi ?

- **filtrage simple** : examen **individuel** des paquets
- beaucoup de communications basées sur **TCP** ⇒
  - **mode connecté**
  - obtention **dynamique** d'un numéro de **port** ⇒ impossible de connaître le numéro a priori

## Comment ?

- suivi des **échanges de messages**
- **règles** de filtrage appliquées en fonction des **paquets précédents**
- à partir du moment où une connexion est acceptée, les autres paquets du même dialogue sont automatiquement acceptés

# Filtrage dynamique (statefull packet filtering)

## Pourquoi ?

- **filtrage simple** : examen **individuel** des paquets
- beaucoup de communications basées sur **TCP** ⇒
  - **mode connecté**
  - obtention **dynamique** d'un numéro de **port** ⇒ impossible de connaître le numéro a priori

## Comment ?

- suivi des **échanges de messages**
- **règles** de filtrage appliquées en fonction des **paquets précédents**
- à partir du moment où une connexion est acceptée, les autres paquets du même dialogue sont automatiquement acceptés

# Filtrage applicatif (proxy)

- présence de **failles** dans les applications utilisées
- adapté à l'utilisation des protocoles et des ports spécifique à chaque application

# Choix d'iptables

## Avantages :

- utilisation plus **simple** que sur des matériels dédiés (comme l'IOS des routeurs CISCO)
- **filtrage dynamique**
- gratuit, disponible sur Linux

# Tables d'iptables

## filter (table par défaut)

Précise le **filtrage** des paquets. Chaînes : **OUTPUT**, **INPUT** et **FORWARD**

## nat

Effectue la **translation d'adresses** ; utilisée lors de la création d'une nouvelle connexion. Chaînes : **PREROUTING**, **OUTPUT** et **POSTROUTING**

## mangle

Modification spécialisée des paquets. Chaînes : **PREROUTING**, **OUTPUT**, **INPUT**, **FORWARD** et **POSTROUTING**

## raw

Évite de tracer la connexion. Chaînes : **PREROUTING** et **OUTPUT**

# Chaînes

## Notion de chaîne

- liste **ordonnée** de règles
- **politique** par défaut

## Types de chaînes

- **INPUT** : appliquée aux paquets destinés à des sockets locales
- **FORWARD** : appliquée aux paquets routés par le parefeu
- **OUTPUT** : appliquée aux paquets générés localement, avant routage
- **PREROUTING** : modification des paquets entrants
- **POSTROUTING** : modification des paquets sortants

# Chaînes

## Notion de chaîne

- liste **ordonnée** de règles
- **politique** par défaut

## Types de chaînes

- **INPUT** : appliquée aux paquets destinés à des sockets locales
- **FORWARD** : appliquée aux paquets routés par le parefeu
- **OUTPUT** : appliquée aux paquets générés localement, avant routage
- **PREROUTING** : modification des paquets entrants
- **POSTROUTING** : modification des paquets sortants

# Cibles (target) d'iptables

Actions à effectuer :

- **DROP** : détruire le paquet
- **REJECT** : détruire le paquet et renvoyer un paquet ICMP
- **ACCEPT** : accepter le paquet
- **LOG** : tracer le paquet dans les logs
- cibles définies par l'utilisateur

# Exemple d'ajout de règle de filtrage

```
iptables -A FORWARD -s 1.1.1.1 -p tcp --dport 80 --syn -j DROP
```

(1)                      (2)                      (3)                      (4)                      (5)                      (6)

- 1 ajouter une règle à la cible **FORWARD**, qui, pour tous les paquets traversant le routeur
- 2 dont l'adresse **IP source** est **1.1.1.1**
- 3 qui encapsulent des segments du **protocole TCP**
- 4 dont le **port destination** est **80**
- 5 qui ont le **drapeau SYN** (ouvertures de connexion),
- 6 a pour effet de **détruire** le paquet

# Exemple de politique par défaut

```
iptables -P FORWARD DROP
```

La **politique** pour la chaîne **FORWARD**, utilisée par défaut si aucune règle ne s'applique, est de **détruire** le paquet.

# Exemple de création d'une cible utilisateur

- 1 Création d'une **nouvelle cible** appelée **LOGDROP** :  
`iptables -N LOGDROP`
- 2 **Ajout** à la **cible LOGDROP** d'une règle **traçant** le paquet :  
`iptables -A LOGDROP -j LOG`
- 3 **Ajout** à la **cible LOGDROP** d'une règle **détruisant** le paquet :  
`iptables -A LOGDROP -j DROP`

# Suivi de connexion

Utiliser `-m state` pour utiliser le module `state` pour pouvoir tester l'état du paquet par rapport à la connexion.

## États des paquets

- **NEW** : paquet correspondant à un nouvel échange
- **INVALID** : le paquet ne peut pas être identifié
- **ESTABLISHED** : paquet faisant partie d'une communication établie
- **RELATED** : paquet pour une nouvelle communication reliée à une autre déjà en cours

## 4 Noms de domaines — DNS

- Principes généraux
- L'espace de noms
- Serveurs de noms de domaine
- Résolution d'adresse
- Fichiers de configuration
- Résolution inverse
- Format des messages

# Pourquoi le DNS (Domain Name System) ?

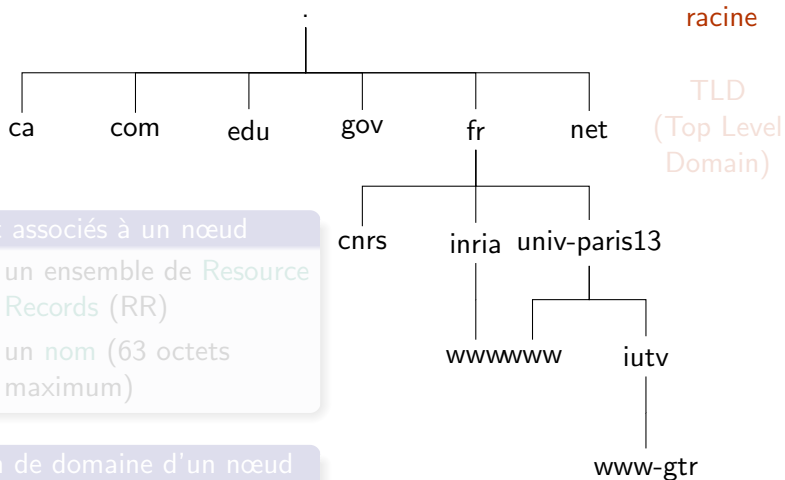
- adresses IP difficiles à retenir, peu explicites
- outil pour effectuer de la résolution d'adresses :
  - symbolique → IP
  - IP → symbolique
- facilité de gestion et de structuration des adresses symboliques.

# Composants du DNS

Le système de noms de domaines comporte :

- un **espace de noms** de domaines :
  - **structure hiérarchique**
  - garantit l'**unicité** des noms
- un ensemble de **serveurs** de noms distribué
- des **clients** permettant de « **résoudre** » des noms de domaines en interrogeant les serveurs

# Une structure arborescente



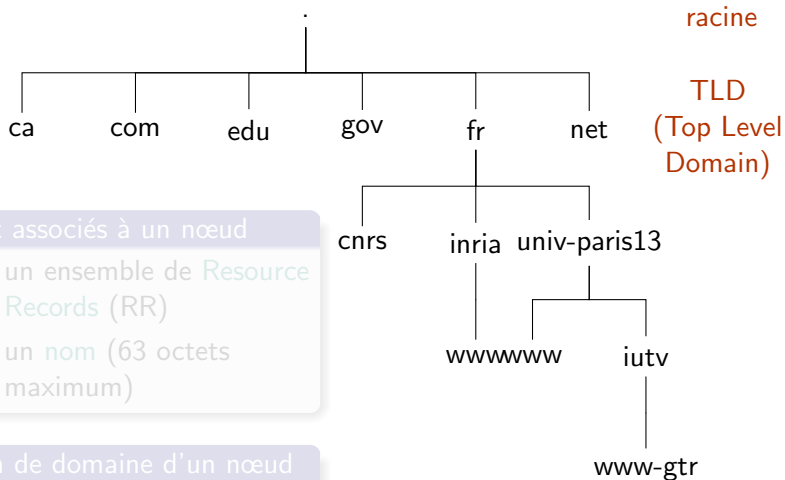
Sont associés à un nœud

- un ensemble de Resource Records (RR)
- un nom (63 octets maximum)

Nom de domaine d'un nœud

chemin du nœud à la racine, en utilisant le séparateur .

# Une structure arborescente



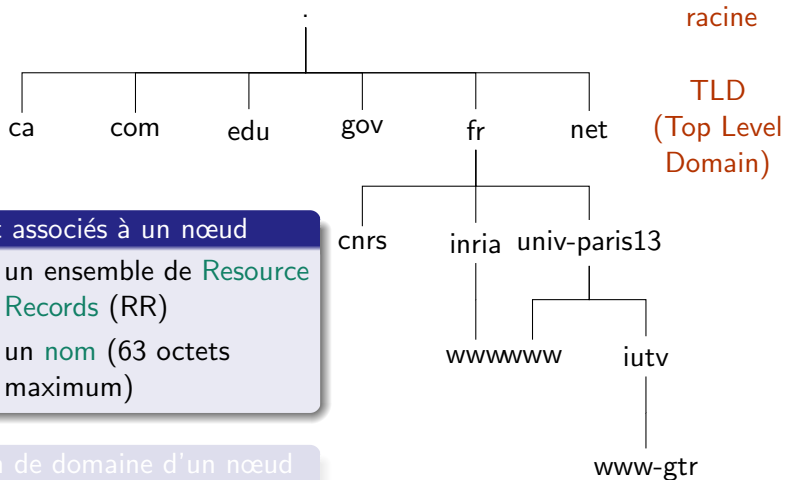
Sont associés à un nœud

- un ensemble de Resource Records (RR)
- un nom (63 octets maximum)

Nom de domaine d'un nœud

chemin du nœud à la racine, en utilisant le séparateur .

# Une structure arborescente

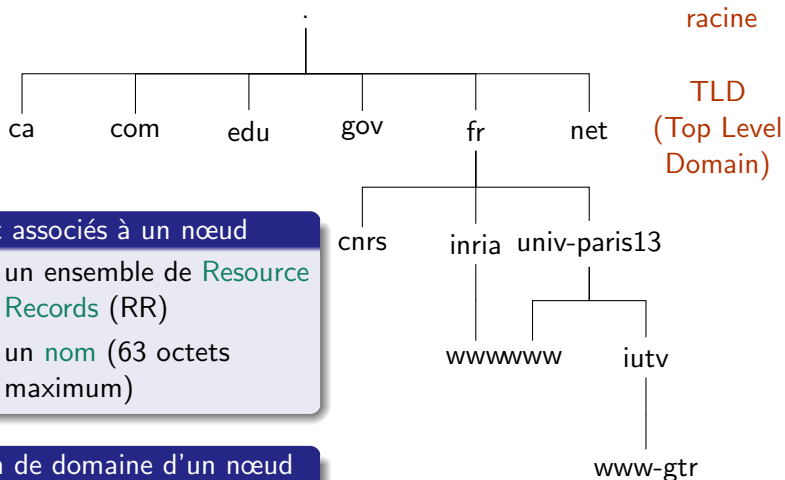


Sont associés à un nœud

- un ensemble de **Resource Records (RR)**
- un **nom** (63 octets maximum)

Nom de domaine d'un nœud  
chemin du nœud à la racine,  
en utilisant le séparateur .

# Une structure arborescente



Sont associés à un nœud

- un ensemble de **Resource Records (RR)**
- un **nom** (63 octets maximum)

Nom de domaine d'un nœud

**chemin du nœud à la racine**, en utilisant le séparateur .

# Adressage hiérarchique

- **Fully Qualified Domain Name (FQDN)** : adresse symbolique
- se termine par un .
- contient uniquement des lettres, chiffres, -
- insensible à la casse : pas de différenciation des majuscules et minuscules
- **codage** :
  - longueur de l'étiquette (1 octet)
  - code ASCII de l'étiquette
  - 00 : point final

# Serveurs de noms

## Rôle

Établir la **correspondance** entre **nom** de domaine et **adresse IP**

## Domaine/sous-domaine

- **domaine** : **sous-arborescence** ayant pour origine le nom du domaine
- **sous-domaine** : domaine **inclus** dans un autre

## Responsabilités

- **unicité** des noms à un même niveau
- **zone** de responsabilité : partie descriptive d'un nœud, incluse dans un domaine
- **autres zones** du domaine déléguées à d'autres serveurs
- un serveur peut **gérer plusieurs zones** (« faire autorité »)

# Serveurs de noms

## Rôle

Établir la **correspondance** entre **nom** de domaine et **adresse IP**

## Domaine/sous-domaine

- **domaine** : **sous-arborescence** ayant pour origine le nom du domaine
- **sous-domaine** : domaine **inclus** dans un autre

## Responsabilités

- **unicité** des noms à un même niveau
- **zone** de responsabilité : partie descriptive d'un nœud, incluse dans un domaine
- **autres zones** du domaine déléguées à d'autres serveurs
- un serveur peut **gérer plusieurs zones** (« faire autorité »)

# Serveurs de noms

## Rôle

Établir la **correspondance** entre **nom** de domaine et **adresse IP**

## Domaine/sous-domaine

- **domaine** : **sous-arborescence** ayant pour origine le nom du domaine
- **sous-domaine** : domaine **inclus** dans un autre

## Responsabilités

- **unicité** des noms à un même niveau
- **zone** de responsabilité : partie descriptive d'un nœud, incluse dans un domaine
- **autres zones** du domaine déléguées à d'autres serveurs
- un serveur peut **gérer plusieurs zones** (« faire autorité »)

# Autres serveurs

## Serveur secondaire (slave)

- prend le **relai** en cas de **panne**
- assurer la **répartition de charge**

## Serveur relai (forwarder)

- **relaie** les demandes vers un autre serveur
- **garde** l'information en **cache**
- permet de **limiter la charge** du serveur primaire

# Autres serveurs

## Serveur secondaire (slave)

- prend le **relai** en cas de **panne**
- assurer la **répartition de charge**

## Serveur relai (forwarder)

- **relaie** les demandes **vers un autre serveur**
- **garde** l'information **en cache**
- permet de **limiter la charge** du serveur primaire

# Autorité

Un serveur :

- fait autorité pour une zone dans un domaine.
- connaît et fait autorité pour toutes les correspondances de sa zone.
- connaît et déclare les responsables des sous-domaines de son domaine.
- ne peut faire autorité que s'il se déclare comme tel et que le serveur de niveau supérieur le déclare aussi comme tel.

# Résolution statique

Utilisation du fichier `/etc/hosts`.

```
#Adresse_IP  adresse_symbolique  alias
127.0.0.1    localhost
192.16.1.2   machine.domaine.fr  machine
192.16.1.1   serveur.domaine.fr  serveur
```

# Algorithme de résolution

## Modes de recherche

- **récuratif** : la résolution se propage entièrement de serveur en serveur
- **itératif** : un serveur renvoie l'adresse du prochain serveur à contacter

## Étapes de résolution

- le client contacte son serveur DNS local  $NS_1$  :
  - sur le port 53, dans un datagramme UDP
  - contenant la demande, par exemple `aaa.bbb.ccc.fr`
  - la demande est en général récursive
- si  $NS_1$  fait autorité ou l'adresse est en cache : réponse immédiate
- sinon : recherche du plus grand suffixe connu (`bbb.ccc.fr`, `ccc.fr`, `fr` ou `.`) avec un RR de type NS (par exemple  $NS_2$ )
  - mode récursif :  $NS_1$  demande à  $NS_2$  d'effectuer toute la résolution
  - mode itératif :  $NS_1$  répond l'adresse de  $NS_2$  à contacter

# Algorithme de résolution

## Modes de recherche

- **récuratif** : la résolution se propage entièrement de serveur en serveur
- **itératif** : un serveur renvoie l'adresse du prochain serveur à contacter

## Étapes de résolution

- le client contacte **son serveur DNS local**  $NS_1$  :
  - sur le port 53, dans un datagramme UDP
  - contenant la demande, par exemple `aaa.bbb.ccc.fr`
  - la demande est en général récurative
- si  $NS_1$  **fait autorité** ou l'adresse est en **cache** : réponse immédiate
- **sinon** : recherche du plus grand suffixe connu (`bbb.ccc.fr`, `ccc.fr`, `fr` ou `.`) avec un RR de type NS (par exemple  $NS_2$ )
  - **mode récuratif** :  $NS_1$  demande à  $NS_2$  d'effectuer toute la résolution
  - **mode itératif** :  $NS_1$  répond l'adresse de  $NS_2$  à contacter

# Service et résolution de noms

`/etc/nsswitch.conf`

Configuration des **bases de données système** et du **service de noms**

**hosts files dns**

indique qu'il faut chercher les adresses des hôtes d'abord dans les **fichiers locaux** puis sur le **DNS**.

`/etc/resolv.conf`

Configuration de la **résolution de noms** :

- **nameserver** *IP\_serveur\_de\_noms* : serveur de noms à interroger
- **search** *liste\_de\_domaines* : liste de domaines dans lesquels chercher des noms d'hôtes

# Service et résolution de noms

## /etc/nsswitch.conf

Configuration des **bases de données système** et du **service de noms**

**hosts files dns**

indique qu'il faut chercher les adresses des hôtes d'abord dans les **fichiers locaux** puis sur le **DNS**.

## /etc/resolv.conf

Configuration de la **résolution de noms** :

- **nameserver** *IP\_serveur\_de\_noms* : serveur de noms à interroger
- **search** *liste\_de\_domaines* : liste de domaines dans lesquels chercher des noms d'hôtes

# /etc/named.conf

## Options

```
options {  
    directory "/var/named"; }  
● directory chemin : répertoire dans lequel se trouvent les fichiers de configuration de zone
```

- **recursion** *yes/no* : mode récursif ou itératif (défaut : yes)

## Définition des zones

```
zone "p13.fr" {  
    type master;  
    file "p13.fr"; }  
● type master/forward/... : type de zone
```

- **type** *master/forward/...* : type de zone
  - **master** : le serveur a autorité
  - **forward** : transférer toutes les requêtes à d'autres serveurs
- **file** *nom\_de\_fichier* : fichier de configuration de la zone

# /etc/named.conf

## Options

```
options {  
    directory "/var/named"; }  
}
```

- **directory** *chemin* : répertoire dans lequel se trouvent les fichiers de configuration de zone
- **recursion** *yes/no* : mode récursif ou itératif (défaut : yes)

## Définition des zones

```
zone "p13.fr" {  
    type master;  
    file "p13.fr"; }  
}
```

- **type** *master/forward/...* : type de zone
  - **master** : le serveur a autorité
  - **forward** : transférer toutes les requêtes à d'autres serveurs
- **file** *nom\_de\_fichier* : fichier de configuration de la zone

# Resource Records

**nom\_de\_domaine [TTL] [classe] type\_RR données**

## Types de RR

- **A** : associe un nom à une adresse IP
- **CNAME** : définit un alias sur un nom
- **TXT** : texte sans signification particulière (description du domaine, commentaire)
- **NS** : nom du serveur responsable
- **SOA** (Start Of Authority) : début d'une zone d'autorité
- **MX** : déclaration d'un serveur ou relai de mail
- **PTR** : résolution inverse
- ...

# Resource Records

**nom\_de\_domaine [TTL] [classe] type\_RR données**

## Types de RR

- **A** : associe un nom à une adresse IP
- **CNAME** : définit un alias sur un nom
- **TXT** : texte sans signification particulière (description du domaine, commentaire)
- **NS** : nom du serveur responsable
- **SOA** (Start Of Authority) : début d'une zone d'autorité
- **MX** : déclaration d'un serveur ou relai de mail
- **PTR** : résolution inverse
- ...

# Configuration de zone

```
$ORIGIN p13.fr.
```

Domaine par défaut

# Configuration de zone

```
$ORIGIN p13.fr.
```

```
$TTL 3D
```

TTL de 3 jours (3 days) par défaut

# Configuration de zone

```
$ORIGIN p13.fr.
```

```
$TTL 3D
```

```
@ IN SOA dns.p13.fr. root.dns.p13.fr. 2007092101 24H 6H 3W 1D
```

- @ : remplace le nom de domaine par \$ORIGIN
- IN : classe internet
- SOA : Start of Authority
- adresse mail de l'administrateur avec @ remplacé par un .
- numéro de version : pour mettre à jour les caches...
- Refresh : délai d'interrogation du numéro de version
- Retry : délai d'attente avant de refaire un refresh raté
- Expire : durée de vie des données si aucun refresh n'a pu avoir lieu
- negTTL : durée de vie des réponses négatives

# Configuration de zone

```
$ORIGIN p13.fr.
```

```
$TTL 3D
```

```
@ IN SOA dns.p13.fr. root.dns.p13.fr. 2007092101 24H 6H 3W 1D  
    NS dns.p13.fr.
```

Le serveur responsable

# Configuration de zone

```
$ORIGIN p13.fr.
```

```
$TTL 3D
```

```
@ IN SOA dns.p13.fr. root.dns.p13.fr. 2007092101 24H 6H 3W 1D  
    NS dns.p13.fr.
```

```
    TXT "le domaine de p13"
```

[Commentaire](#)

# Configuration de zone

```
$ORIGIN p13.fr.
```

```
$TTL 3D
```

```
@ IN SOA dns.p13.fr. root.dns.p13.fr. 2007092101 24H 6H 3W 1D
```

```
NS dns.p13.fr.
```

```
TXT "le domaine de p13"
```

```
MX 10 mail
```

```
MX 20 dns
```

déclaration de serveur/relai de mail

# Configuration de zone

```
$ORIGIN p13.fr.
```

```
$TTL 3D
```

```
@ IN SOA dns.p13.fr. root.dns.p13.fr. 2007092101 24H 6H 3W 1D
```

```
NS dns.p13.fr.
```

```
TXT "le domaine de p13"
```

```
MX 10 mail
```

```
MX 20 dns
```

```
dns A 150.10.0.1
```

```
mail A 150.10.0.2
```

Association adresse symbolique ↔ IP

# Configuration de zone

```
$ORIGIN p13.fr.  
$TTL 3D  
@ IN SOA dns.p13.fr. root.dns.p13.fr. 2007092101 24H 6H 3W 1D  
    NS dns.p13.fr.  
    TXT "le domaine de p13"  
    MX 10 mail  
    MX 20 dns  
dns A 150.10.0.1  
mail A 150.10.0.2  
iutv NS dns.iutv  
dns.iutv A 150.10.0.3
```

Déclaration d'un sous domaine et de son serveur de noms

# Résolution inverse

Pseudo-domaine `in-addr.arpa.` avec des adresses inversées.

## Exemple

L'adresse `192.16.1.1`, de classe C est  
dans le domaine `1.16.192.in-addr.arpa.`  
qui est lui-même dans `16.192.in-addr.arpa.`

# Configuration de zone inverse

```
$ORIGIN 10.150.in-addr.arpa.  
$TTL 3D  
@ IN SOA dns.p13.fr. root.dns.p13.fr. 2007092101 24H 6H 3W 1D  
    NS dns.p13.fr.  
    TXT "le domaine de p13"  
1.0 PTR dns  
2.0 PTR mail  
3.0 PTR dns.iutv
```

adresses locales inverses des machines

# Format des messages

Datagrammes **UDP** d'au plus 512 octets.

TransID	Flags	requests	responses	authoritative	additional
2	2	2	2	2	2

S1	Qname	Qtype	Qclass			
S2	Qname	Qtype	Qclass	TTL	lg. données	IP

- **S1** : RRs de requête
- **S2** : RRs de réponse
- **S3** : RRs pointant vers un autre NS
- **S4** : RRs supplémentaires
- **Qname** : nom de domaine = suite de (longueur d'étiquette, étiquette), et 00 pour terminer
- **Qtype** : 0001 = A, 002 = NS, 0005 = CNAME, 000C = PTR, 000F=MX
- **Qclass** : 0001 = IN

# Drapeaux

QR	Opcode	AA	TC	RD	RA	000	RCODE
1 b	4 b	1 b	1 b	1 b	1 b	3 b	4 b

- **QR** : Query = 0, Response = 1
- **OpCode** : Standard Query = 0, Inversed Query = 1, Status Request = 2
- **AA** : Authoritative Answer (réponse d'un NS)
- **TC** : Truncated, si le message est trop long
- **RD** : Recursion Demanded
- **RA** : Recursion Available
- **RCODE** : Response code = 0 s'il n'y a pas d'erreur

## 5 Transfert de fichiers — FTP

- Généralités
- FTAM — File Transfer, Access and Management
- FTP — File Transfer Protocol
  - Principes généraux de FTP
  - Les différents processus
  - Commandes de FTP

# Généralités

## Objectifs

- **échange** de fichiers
- **modification** et **lecture** de fichiers **à distance**
- **création** et **maintenance** des **paramètres** de fichiers distants

## Mise en œuvre

- **FTAM** (File Transfer, Access and Management) : norme ISO 8571-1
  - structure de **fichiers virtuels** ⇒ gestion transparente de l'arborescence
  - accès **sécurisé**
- **FTP** (File Transfer Protocol) : simple, utilisé sur internet

# Généralités

## Objectifs

- **échange** de fichiers
- **modification** et **lecture** de fichiers **à distance**
- **création** et **maintenance** des **paramètres** de fichiers distants

## Mise en œuvre

- **FTAM** (File Transfer, Access and Management) : norme ISO 8571-1
  - structure de **fichiers virtuels** ⇒ gestion transparente de l'arborescence
  - accès **sécurisé**
- **FTP** (File Transfer Protocol) : simple, utilisé sur internet

# Fonctionnalités de FTAM

## Opérations sur les fichiers

- création, suppression de fichier
- sélection, désélection de fichier
- accès sécurisé avec mot de passe
- lecture, modification des propriétés de fichiers
- ouverture en lecture ou en écriture
- fermeture d'un fichier

## Opérations sur le contenu des fichiers

- lecture des données
- recherche de données
- insertion, remplacement, suppression, ajout en fin de fichier

# Fonctionnalités de FTAM

## Opérations sur les fichiers

- création, suppression de fichier
- sélection, désélection de fichier
- accès sécurisé avec mot de passe
- lecture, modification des propriétés de fichiers
- ouverture en lecture ou en écriture
- fermeture d'un fichier

## Opérations sur le contenu des fichiers

- lecture des données
- recherche de données
- insertion, remplacement, suppression, ajout en fin de fichier

# Principes généraux

## Structure d'un échange

- **ouverture** de connexion avec **login** et **mot de passe**
- **transfert** de fichiers
- **fermeture** de connexion

## Modèle client/serveur

Le **client** envoie des requêtes qui sont des **ordres**.

Le **serveur** attend les requêtes et effectue les actions. Il renvoie des **réponses**.

Ce mécanisme utilise **deux ports** :

- **port 21** : canal de **contrôle**
- **port 20** : canal de **données**

# Principes généraux

## Structure d'un échange

- **ouverture** de connexion avec **login** et **mot de passe**
- **transfert** de fichiers
- **fermeture** de connexion

## Modèle client/serveur

Le **client** envoie des requêtes qui sont des **ordres**.

Le **serveur** attend les requêtes et effectue les actions. Il renvoie des **réponses**.

Ce mécanisme utilise **deux ports** :

- **port 21** : canal de **contrôle**
- **port 20** : canal de **données**

# Processus exécutés

## DTP (Data Process Transfer)

- **établit** la connexion
- **gère le canal de données**

## PI (Protocol Interpreter)

**Commande le DTP** grâce aux commandes reçues sur le canal de contrôle

## Fonctionnement

- **ouverture** de la connexion avec identification de l'utilisateur
- **transfert** de fichiers et **paramétrage** du transfert
- **fermeture** de la connexion

# Processus exécutés

## DTP (Data Process Transfer)

- **établit** la connexion
- **gère le canal de données**

## PI (Protocol Interpreter)

**Commande le DTP** grâce aux commandes reçues sur le canal de contrôle

## Fonctionnement

- **ouverture** de la connexion avec identification de l'utilisateur
- **transfert** de fichiers et **paramétrage** du transfert
- **fermeture** de la connexion

# Commandes de FTP

## Contrôle d'accès

- **USER** : identification de l'utilisateur
- **PASS** (password) : mot de passe de l'utilisateur (suit immédiatement une commande USER)
- **CWD** (change working directory) : changement de répertoire dans l'arborescence distante
- **CDUP** (change directory up) : remonter dans l'arborescence distante
- **QUIT** : fin de la session

# Commandes de FTP

## Paramétrage du transfert

- **TYPE** : type de format d'échange des données (A=ASCII)
- **STRU** (structure) : structures échangées (F=FILE)
- **MODE** : mode de transfert des données (S=STREAM)
- **PORT** : numéro de port auquel le DTP-serveur doit se connecter

## Commandes de service

- **RETR** (retrieve) : copie sur la machine locale
- **STOR** (store) : copie sur la machine distante
- **LIST** : liste des fichiers distants
- **HELP** : liste des commandes

# Commandes de FTP

## Paramétrage du transfert

- **TYPE** : type de format d'échange des données (A=ASCII)
- **STRU** (structure) : structures échangées (F=FILE)
- **MODE** : mode de transfert des données (S=STREAM)
- **PORT** : numéro de port auquel le DTP-serveur doit se connecter

## Commandes de service

- **RETR** (retrieve) : copie sur la machine locale
- **STOR** (store) : copie sur la machine distante
- **LIST** : liste des fichiers distants
- **HELP** : liste des commandes

## 6 Gestion du courrier — SMTP

- Fonctionnalités à assurer
- Le protocole X400
  - Composants
  - Adressage
- SMTP — Simple Mail Transfer Protocol
  - Généralités
  - Adresses
  - Fonctionnement
- POP — Post Office Protocol
- IMAP — Internet Message Access Protocol

# Fonctionnalités à assurer

- **communication** de messages contenant :
  - du **texte** brut
  - des **fichiers** son, images, ...
- gestion de **boîte aux lettres**
- diffusion de messages à des **destinataires multiples**
- **suivi** des messages et réponses

# Caractéristiques du courrier électronique

- service **asynchrone** : le **dépôt** et la **réception** de messages n'ont **pas** lieu **simultanément**
- l'**enveloppe** a un format précis
- le **corps** du message est **libre**

# Composants de ISO X400

## User Agent (UA)

**Interface utilisateur** permettant d'accéder au service de messagerie

## Message Store (MS)

**Boîte aux lettres** qui permet de stocker, filtrer, trier, ... des messages

## Message Transfer Agent (MTA)

Agents qui coopèrent pour **transmettre** les messages.

Ces agents forment un **Message Transfer System (MTS)**.

## Access Unit

Interface permettant de gérer un **envoi physique**, par exemple par fax.

# Composants de ISO X400

## User Agent (UA)

**Interface utilisateur** permettant d'accéder au service de messagerie

## Message Store (MS)

**Boîte aux lettres** qui permet de stocker, filtrer, trier, ... des messages

## Message Transfer Agent (MTA)

Agents qui coopèrent pour **transmettre** les messages.  
Ces agents forment un **Message Transfer System (MTS)**.

## Access Unit

Interface permettant de gérer un **envoi physique**, par exemple par fax.

# Composants de ISO X400

## User Agent (UA)

**Interface utilisateur** permettant d'accéder au service de messagerie

## Message Store (MS)

**Boîte aux lettres** qui permet de stocker, filtrer, trier, ... des messages

## Message Transfer Agent (MTA)

Agents qui coopèrent pour **transmettre** les messages.  
Ces agents forment un **Message Transfer System (MTS)**.

## Access Unit

Interface permettant de gérer un **envoi physique**, par exemple par fax.

# Composants de ISO X400

## User Agent (UA)

**Interface utilisateur** permettant d'accéder au service de messagerie

## Message Store (MS)

**Boîte aux lettres** qui permet de stocker, filtrer, trier, ... des messages

## Message Transfer Agent (MTA)

Agents qui coopèrent pour **transmettre** les messages.  
Ces agents forment un **Message Transfer System (MTS)**.

## Access Unit

Interface permettant de gérer un **envoi physique**, par exemple par fax.

# Adressage

## Différents types d'adressage

- vers un **terminal** : télex, fax
- **numérique** : téléphones
- **postal** : envois papier
- **mnémonique** : utilisateurs de messagerie électronique

## Adressage mnémonique

- **A** : nom de domaine global
- **P** : nom de domaine privé
- **C** : nom du pays
- **O** (Organisation) : organisation
- **OU** (Organisation Unit) : nom du département dans l'organisation
- **G** (Given name) : prénom
- **S** (Surname) : nom de famille
- ...

# Adressage

## Différents types d'adressage

- vers un **terminal** : télex, fax
- **numérique** : téléphones
- **postal** : envois papier
- **mnémonique** : utilisateurs de messagerie électronique

## Adressage mnémonique

- **A** : nom de domaine global
- **P** : nom de domaine privé
- **C** : nom du pays
- **O** (Organisation) : organisation
- **OU** (Organisation Unit) : nom du département dans l'organisation
- **G** (Given name) : prénom
- **S** (Surname) : nom de famille
- ...

# Caractéristiques de SMTP

- sur **internet**
- encapsulé dans des paquets **TCP**
- indépendant du système d'exploitation utilisé
- pas d'authentification de l'émetteur
- pas de confidentialité
- pas d'accusé de réception
- transmission de caractères ASCII uniquement (sinon, codage avant l'émission et décodage lors de la réception)

# Composants

- **User Agent** : lecteur de mail (/bin/mail, mutt, mh, ...)
- **Message Transfer Agent** : sendmail, postfix
- **Message Store** : répertoire /var/mail

# Adresses

## Syntaxe

```
<partie_locale@domaine>
```

## domaine

adresse symbolique (MX du DNS) ou IP

## partie\_locale

- gérée par la machine destinatrice
- doit inclure la possibilité d'utiliser d'autres protocoles comme X400

# Fonctionnement de SMTP

- création d'une **communication bidirectionnelle** avec le destinataire via des **sockets**
- séquence de **commandes/réponses**
- **transfert** lorsque le destinataire est prêt
- **terminaison** demandée par l'émetteur

# Commandes de SMTP

## Commandes de contrôle

- **HELO** : présentation de la machine cliente
- **RSET** : interruption de la session
- **QUIT** : fermeture de la session
- **TURN** : inversion des rôles émetteur/destinataire

## Commandes d'envoi de messages

- **MAIL From: adresse** : adresse de l'émetteur
- **RCPT To: adresse** : adresse du destinataire
- **DATA texte** : corps du message découpé en lignes, terminé par un `.` en début de ligne

# Commandes de SMTP

## Commandes de contrôle

- **HELO** : présentation de la machine cliente
- **RSET** : interruption de la session
- **QUIT** : fermeture de la session
- **TURN** : inversion des rôles émetteur/destinataire

## Commandes d'envoi de messages

- **MAIL From: adresse** : adresse de l'émetteur
- **RCPT To: adresse** : adresse du destinataire
- **DATA texte** : corps du message découpé en lignes, terminé par un . en début de ligne

# Entête d'un message

## Champs précisés par l'utilisateur

- adresse du destinataire
- sujet
- copie
- ...

## Champs ajoutés au fur et à mesure par les relais

- **Return-path**: : pour renvoyer la réponse
- **Received**: : chemin parcouru par le message

# Entête d'un message

## Champs précisés par l'utilisateur

- adresse du destinataire
- sujet
- copie
- ...

## Champs ajoutés au fur et à mesure par les relais

- **Return-path:** : pour renvoyer la réponse
- **Received:** : chemin parcouru par le message

# Entête d'un message

## Champs d'adressage

- **From:** : identité de l'expéditeur
- **Sender:** : utilisé pour une réexpédition automatique
- **Reply-to:** : adresse à laquelle envoyer la réponse
- **To:** : adresse du destinataire
- **Cc:** (Carbon Copy) : autres destinataires en copie du message
- **Bcc:** (Blind Carbon Copy) : autre destinataires, en copie, mais dont l'adresse est masquée

# POP — Post Office Protocol

- utilisé pour aller chercher les messages dans le MS
- protocole interactif
- authentification par login et mot de passe en clair
- messages téléchargés vers le client
- libère le MS

# IMAP — Internet Message Access Protocol

- similaire à POP
- gestion de répertoires pour ranger les messages
- boîte de courrier entrant (INBOX) sur le serveur
- les autres boîtes ou répertoires peuvent être gérés localement sur le client

## 7 Annuaire — LDAP

- Notion d'annuaire
- Annuaire X500
- LDAP
  - Introduction à LDAP
  - Protocole
  - Sécurisation
  - Format d'échange
  - URL LDAP
  - Clients LDAP
  - Serveurs LDAP

# Notion d'annuaire

**Annuaire** : ensemble de **matériels**, **logiciels** et de **traitements** permettant de fournir des informations.

## Base de données spécialisée

- Accès optimisés en lecture
- Information changeant peu fréquemment (noms, adresses)
- Mises à jour, écriture assez lente
- Recherche multi-critères d'attributs
- Utilisés pour la gestion des données associées aux utilisateurs (noms, droits d'accès, ...) et aux machines.

# Notion d'annuaire

**Annuaire** : ensemble de **matériels**, **logiciels** et de **traitements** permettant de fournir des informations.

## Base de données spécialisée

- Accès optimisés en lecture
- Information changeant peu fréquemment (noms, adresses)
- Mises à jour, écriture assez lente
- Recherche multi-critères d'attributs
- Utilisés pour la **gestion des données** associées aux **utilisateurs** (noms, droits d'accès, ...) et aux **machines**.

# Utilisation d'un annuaire

- localisation de ressources
- recherche/navigation
- gestion des droits (habilitations)
- assurer l'interopérabilité
- garantir la sécurité

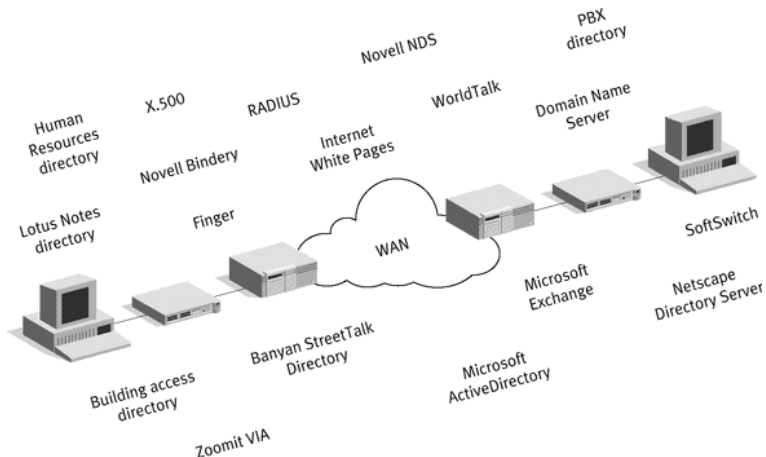
# Informations d'un annuaire

- **identités** (noms, prénoms, état civil)
- informations relatives à l'**organisation de l'entreprise** (organigramme)
- informations de **contact** (adresses, téléphones, mails, ...)
- **accès** au système (login, mot de passe, date d'expiration du compte, ...)
- **inventaire** des postes de travail et matériels divers
- informations sur la **configuration du réseau**
- **applications réseau** (informations de configuration, préférences des utilisateurs)

# Divers annuaires connus

- **DNS** (Domain Name System) : association adresses symboliques ↔ adresses IP
- **WHOIS** : informations sur un domaine ou une adresse IP
- **NIS** (Network Information Service) : spécifique à UNIX
- **X500** : système d'annuaire complexe

# Nombreux annuaires



# Trop d'annuaires !

Si chaque application a son annuaire :

- la **complexité d'administration** augmente très rapidement car chaque annuaire a ses propres fonctions, protocoles d'accès, formats de données, conventions de nommage. . .
- **risques d'incohérences**
  - Exemple : le nom d'un utilisateur pourrait être "John Dœuf" dans un annuaire, "J. Dœuf" dans un autre, et "John C. Dœuf" dans un troisième.
- À chaque **mise à jour**, il faut agir sur tous les annuaires concernés.

# Éléments définis par un annuaire

- ① **règles de nommage** des entités et des objets
- ② **protocole d'accès** (client/serveur, serveur/serveur) et **format** de transfert des données
- ③ **modèle de sécurité** : protection des données, cryptage, règles d'accès (ACL)
- ④ **API** pour développer facilement des applications clientes
- ⑤ **format d'échange** (import/export de données, LDIF pour LDAP)

# Annuaire X500

## X500 : norme ISO (1988)

Spécifie **comment** l'information doit être **stockée** et **consultée** dans un service global d'annuaire.

Ne définit pas le fonctionnement interne du serveur.

## Standards définis par X500

X500 définit un ensemble de standards complexes et lourds :

- espaces de noms
- modèle d'information (schémas)
- modèle fonctionnel (DAP — Directory Access Protocol)
- modèle d'authentification
- modèle distribué d'exécution

# Annuaire X500

## X500 : norme ISO (1988)

Spécifie **comment** l'information doit être **stockée** et **consultée** dans un service global d'annuaire.

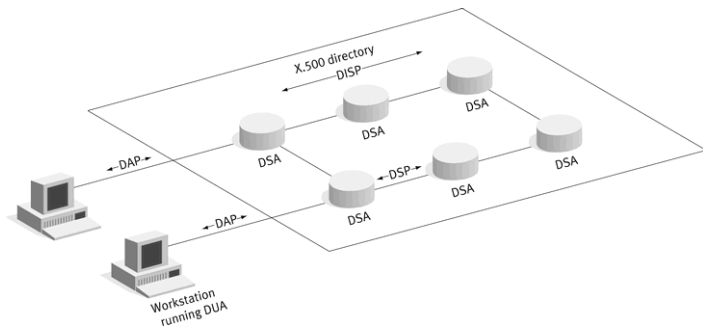
Ne définit pas le fonctionnement interne du serveur.

## Standards définis par X500

X500 définit un ensemble de standards complexes et lourds :

- espaces de noms
- modèle d'information (schémas)
- modèle fonctionnel (DAP — Directory Access Protocol)
- modèle d'authentification
- modèle distribué d'exécution

# Annuaire X500 — modèle fonctionnel



## Composants

- **DSA** (Directory System Agents) : maintiennent l'annuaire
- **DSP** (Directory System Protocol) : protocole entre serveurs
- **DUA** (Directory User Agent) : client accédant à un annuaire
- **DAP** (Directory Agent Protocol) : protocole entre client et serveur

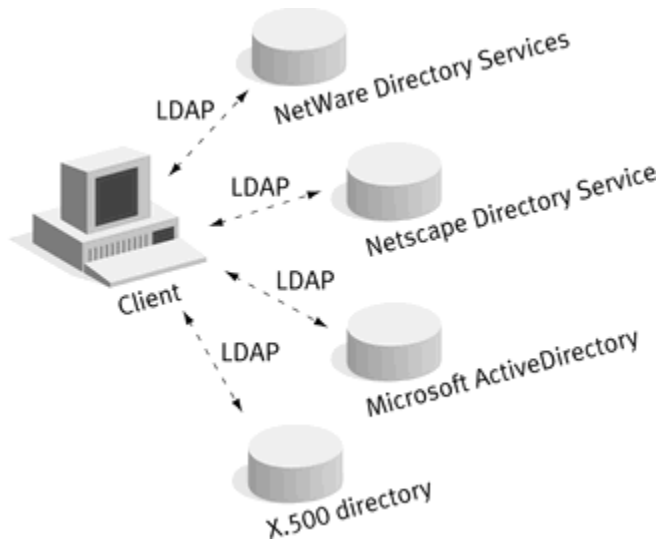
# Inconvénients de X500

- complexe à configurer et requiert **trop de ressources matérielles**
- **trop complexe** pour les besoins de la plupart des organisations
- très **peu de logiciels** implémentent X500
- basé sur le modèle ISO, **mal adapté à TCP/IP**

# LDAP — Lightweight Directory Access Protocol

- développé au début des années 90 à l'Université du Michigan, comme une **alternative légère à X500**
  - RFC 1487 (1993) : LDAP v1
  - RFC 1777 (1995) : LDAP v2
  - RFC 2251 (1997) : LDAP v3
- **adapté à TCP/IP**
- protocole client/serveur et serveur/serveur (réplicats)
- LDAP ne spécifie que le **protocole d'accès (DAP)**, le serveur est **libre de choisir son stockage** des données

# Nombreuses implémentations de LDAP

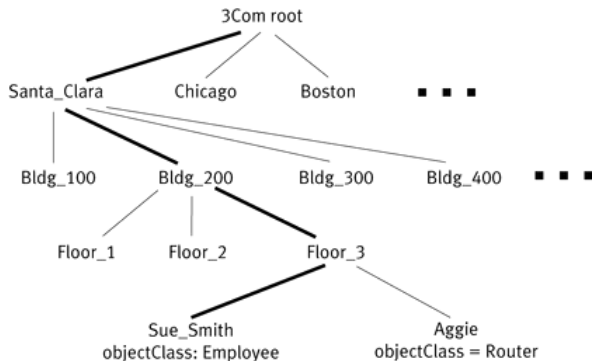


# Stockage dans un annuaire LDAP ?

## Annuaire organisé en « entrées »

- une entrée est un **ensemble d'attributs**
- une entrée est identifiée par un nom, le **DN** (Distinguished Name)
- chaque **attribut** a un **type** et **une ou plusieurs valeurs**
- les **types** sont des chaînes de caractères mnémoniques, comme :
  - cn (common name) : nom
  - givenname : prénom
  - mail : adresse e-mail
- les **valeurs** possibles dépendent du type de l'attribut

# DIT — Directory Information Tree



# Repérage des entrées

## Chaque entrée à son DN

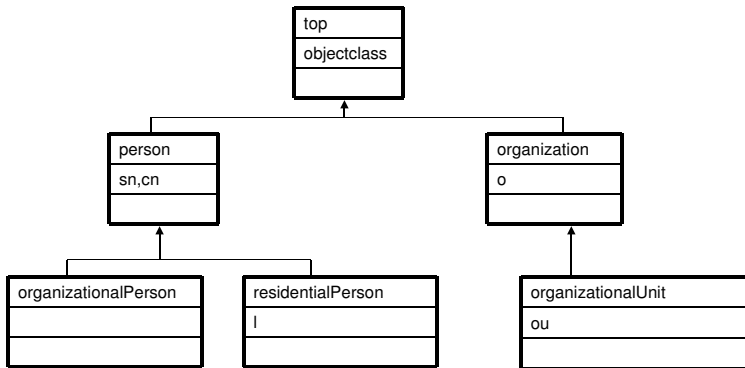
- construit en prenant le nom de l'élément **RDN** (Relative Distinguished Name ) et en lui **ajoutant les noms des entrées parentes** dans la hiérarchie.
- on utilise donc une suite de **paires attribut/valeur** permettant de repérer l'entrée de manière unique.
- Exemple de DN : "uid=209,ou=rt,o=iut"

# Notion de schéma

- Le **schéma** regroupe un **ensemble de définitions** d'objets et d'attributs :
  - **attributs** obligatoires ou facultatifs
  - valeurs possibles pour chaque attribut (**type**)
- Nombreux schémas prédéfinis et standardisés.
- Chaque entrée de l'annuaire doit faire référence à un ou plusieurs schémas, et ne contenir que des attributs rattachés aux types correspondants.

# Classes d'objets

Les classes d'objets (schémas) forment une hiérarchie.



# OID — Object Identifier

- **Identifiant unique** associé à chaque classe d'objet et à chaque type d'attribut
- Instance de normalisation : IANA (*Internet Assigned Numbers Authority*)

# Protocole LDAP

- architecture **client/serveur** (dialogue à l'initiative du client)
- **ports** TCP/389 pour LDAP, TCP/636 pour LDAPS
- **protocole texte**, utilisant le **codage BER** (Basic Encoding Rule)

## Opérations LDAP

Abandon :	abandonne l'opération en cours
Add :	ajoute une entrée au répertoire
Bind :	début une nouvelle session sur le serveur LDAP
Compare :	compare les entrées d'un répertoire selon des critères
Delete :	supprime une entrée d'un répertoire
Extended :	effectue des opérations étendues
Rename :	modifie le nom d'une entrée
Search :	recherche des entrées d'un répertoire
Unbind :	termine une session sur le serveur LDAP

# Protocole LDAP

- architecture **client/serveur** (dialogue à l'initiative du client)
- **ports** TCP/389 pour LDAP, TCP/636 pour LDAPS
- **protocole texte**, utilisant le **codage BER** (Basic Encoding Rule)

## Opérations LDAP

<b>Abandon</b> :	abandonne l'opération en cours
<b>Add</b> :	ajoute une entrée au répertoire
<b>Bind</b> :	début une nouvelle session sur le serveur LDAP
<b>Compare</b> :	compare les entrées d'un répertoire selon des critères
<b>Delete</b> :	supprime une entrée d'un répertoire
<b>Extended</b> :	effectue des opérations étendues
<b>Rename</b> :	modifie le nom d'une entrée
<b>Search</b> :	recherche des entrées d'un répertoire
<b>Unbind</b> :	termine une session sur le serveur LDAP

# Gestion de la connexion

- LDAP travaille sur TCP, donc en **mode connecté**
- mot de passe ou connexion anonyme
- tous les messages contiennent un **identifiant de session** et un code **identifiant la commande**
- toutes les **commandes** sont **acquittées**, avec un code indiquant le **déroulement de l'opération**
- LDAPv3 comprend d'autres commandes, en particulier pour l'authentification



# Encodage BER

**BER** (Basic Encoding Rules) est un système de codage (en texte), utilisé pour coder les échanges LDAP et SNMP.

## Exemple

### Personnel

Record	Length	Contents				
60	8185					
		Name	Length	Contents		
		61	10			
				VisibleString	Length	Contents
				1A	04	"John"
				VisibleString	Length	Contents
				1A	01	"P"
				VisibleString	Length	Contents
				1A	05	"Smith"
				DateOfBirth	Length	Contents
				A0	0A	
				Date	Length	Contents
				43	08	"19590717"

Données transférées : 60 81 85 61 10 1A 04 ... 0A 43 08 19 59 07 17

# LDAP et la sécurité

## Menaces

- accès non autorisé à des informations confidentielles
- modification non autorisée (atteinte à l'intégrité des données)
- dénis de service (DoS)

↪ Définir une politique de sécurité

## Moyens offerts par LDAP

- 1 authentification des clients
- 2 contrôle de l'accès aux données
- 3 chiffrement des échanges

# LDAP et la sécurité

## Menaces

- accès non autorisé à des informations confidentielles
- modification non autorisée (atteinte à l'intégrité des données)
- dénis de service (DoS)

⇒ Définir une politique de sécurité

## Moyens offerts par LDAP

- 1 authentification des clients
- 2 contrôle de l'accès aux données
- 3 chiffrement des échanges

# LDAP et la sécurité

## Menaces

- accès non autorisé à des informations confidentielles
- modification non autorisée (atteinte à l'intégrité des données)
- dénis de service (DoS)

⇒ Définir une politique de sécurité

## Moyens offerts par LDAP

- 1 authentification des clients
- 2 contrôle de l'accès aux données
- 3 chiffrement des échanges

# Authentification des clients & contrôle d'accès

## Authentification lors de l'ouverture de session

- 1 **simple** : mot de passe (**circule en clair sur le réseau !**)
- 2 **simple + SSL ou TLS ou tunnel SSH** : les échanges sont chiffrés
- 3 **SASL** (Simple Authentication and Security Layer) : basée sur "tickets" Kerberos

## ACL (Access Control List)

- **définir les droits d'accès** des utilisateurs sur l'annuaire

`<target> <permission> <bind rule>`

- `<target>` : point d'entrée de l'annuaire auquel s'applique la règle
- `<permission>` : autorise ou refuse un type d'accès (lecture, écriture...)
- `<bind rule>` : identifie l'utilisateur
- syntaxe non standardisée

# Authentification des clients & contrôle d'accès

## Authentification lors de l'ouverture de session

- 1 **simple** : mot de passe (**circule en clair sur le réseau !**)
- 2 **simple + SSL ou TLS ou tunnel SSH** : les échanges sont chiffrés
- 3 **SASL** (Simple Authentication and Security Layer) : basée sur "tickets" Kerberos

## ACL (Access Control List)

- **définir les droits d'accès** des utilisateurs sur l'annuaire

`<target> <permission> <bind rule>`

- `<target>` : point d'entrée de l'annuaire auquel s'applique la règle
  - `<permission>` : autorise ou refuse un type d'accès (lecture, écriture...)
  - `<bind rule>` : identifie l'utilisateur
- **syntaxe non standardisée**

# Format d'échange — LDIF

## LDAP Data Interchange Format

- Format de **fichier texte** pour import/export de données LDAP
- défini par la RFC 2849
- **utilisations** typiques :
  - 1 sauvegarde régulière de l'annuaire
  - 2 échange avec d'autres logiciels (LDAP ou non)

# Format LDIF — exemple

```
dn: cn=Barbara Jensen, ou=Product Development, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Barbara Jensen
cn: Barbara J Jensen
cn: Babs Jensen
sn: Jensen
uid: bjensen
telephonenumber: +1 408 555 1212
description: A big sailing fan.
```

```
dn: cn=Bjorn Jensen, ou=Accounting, dc=airius, dc=com
objectclass: top
objectclass: person
objectclass: organizationalPerson
cn: Bjorn Jensen
sn: Jensen
telephonenumber: +1 408 555 1212
```

# URL LDAP

## Syntaxe pour interroger un annuaire LDAP (RFC 1959)

```
<ldapurl> ::= "ldap://" [ <hostport> ] "/" <dn> [ "?" <attributes>
    [ "?" <scope> "?" <filter> ] ]
```

```
<hostport> ::= <hostname> [ ":" <portnumber> ]
```

```
<dn> ::= a string as defined in RFC 1485
```

```
<attributes> ::= NULL | <attributelist>
```

```
<attributelist> ::= <attributetype>
    | <attributetype> [ "," <attributelist> ]
```

```
<attributetype> ::= a string as defined in RFC 1777
```

```
<scope> ::= "base" | "one" | "sub"
```

```
<filter> ::= a string as defined in RFC 1558
```

# URL LDAP, suite

## Exemples

- toute l'entrée :

```
ldap://ldap.itd.umich.edu///o=University%20of%20Michigan,c=US
```

- juste l'adresse :

```
ldap://ldap.itd.umich.edu/o=University%20of%20Michigan,c=US?postalAddress
```

- toutes les entrées avec cn = "Babs Jensen"

```
ldap:///o=University%20of%20Michigan,c=US??sub?(cn=Babs%20Jensen)
```

# Clients LDAP

- Clients de messagerie (MUA), comme Outlook, Evolution ou Thunderbird (gestion du carnet d'adresse)
- Authentification linux : `pam_ldap`
- Apache : `mod_auth_ldap` (PKI)
- RADIUS, Kerberos
- Navigateurs spécialisés LDAP :
  - `GQ` : client libre GNOME/Linux <http://biot.com/gq>
  - Softerra LDAP Administrator (payant) : <http://ldapadministrator.com/>
  - Plusieurs clients en JAVA :
    - `JXplorer` (libre) <http://pegacat.com/jxplorer/>

Voir aussi sur <http://www.cru.fr/ldap/>

# Serveurs LDAP

## Logiciel libre

- OpenLDAP : Unix, Windows NT/2000.

## Logiciels commerciaux

- Sun Java System Directory Server
- Novell eDirectory
- IBM Tivoli Directory Server
- ...

## 8 RPC — Appels de procédure à distance

- Généralités
- Principes de fonctionnement
- Allocation dynamique de ports
- Création de programmes
- Langage XDR
- Autres RPCs et applications

# RPC — Remote Procedure Call

- standard développé par SUN : Sun RPC
- généralisé en Open Network Computing : ONC RPC

## Objectifs

- programmation d'applications client/serveur
- étendre la notion d'appel de fonction de locale à distante
- éviter d'inclure la gestion du réseau

## Services associés

- Présentation : XDR (eXternal Data Representation)
- Transport : TCP ou UDP

# RPC — Remote Procedure Call

- standard **développé par SUN** : Sun RPC
- généralisé en **Open Network Computing** : ONC RPC

## Objectifs

- programmation d'**applications client/serveur**
- étendre la notion d'**appel de fonction** de **locale** à **distante**
- **éviter** d'inclure la **gestion du réseau**

## Services associés

- **Présentation** : XDR (eXternal Data Representation)
- **Transport** : TCP ou UDP

# RPC — Remote Procedure Call

- standard **développé par SUN** : Sun RPC
- généralisé en **Open Network Computing** : ONC RPC

## Objectifs

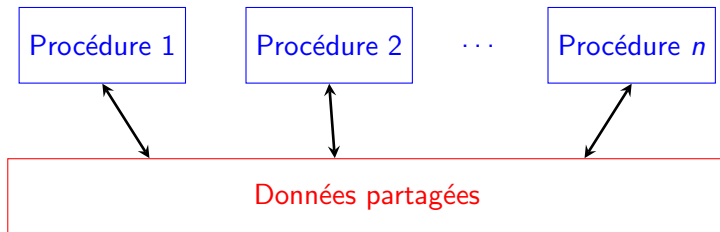
- programmation d'**applications client/serveur**
- étendre la notion d'**appel de fonction** de **locale** à **distante**
- **éviter** d'inclure la **gestion du réseau**

## Services associés

- **Présentation** : XDR (eXternal Data Representation)
- **Transport** : TCP ou UDP

# Programme distant

- identifié par **numéro de programme** et **numéro de version** (entiers sur 32 bits)
- **ensemble de procédures** avec leur **propre identificateur** (32 bits)
- les procédures **partagent de la mémoire**
- par contre, les **programmes clients** ont leurs **données propres**



# Appel de procédures distantes

**Client**

**Serveur**

# Appel de procédures distantes

**Client**



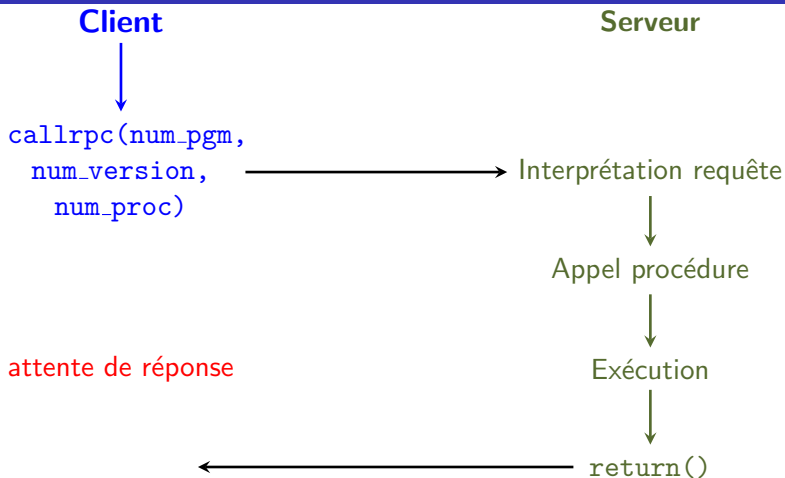
```
callrpc(num_pgm,  
        num_version,  
        num_proc)
```



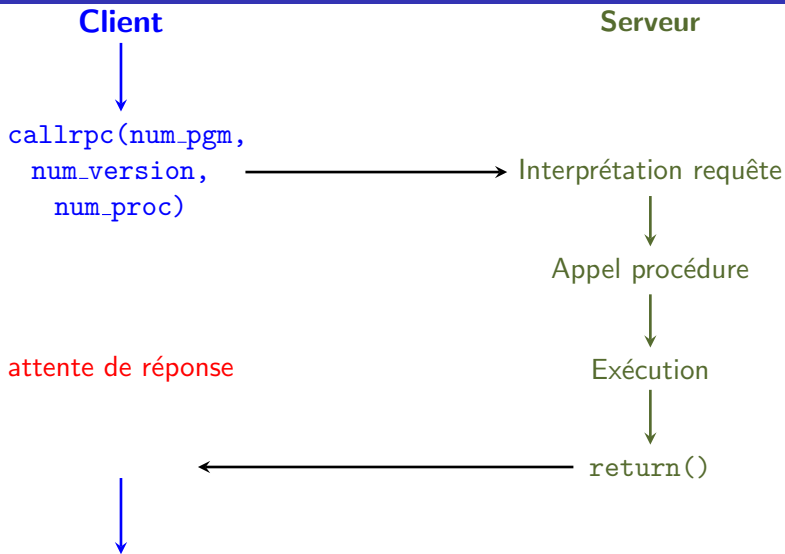
**Serveur**

attente de réponse

# Appel de procédures distantes



# Appel de procédures distantes



# RPC port mapper

## Serveur RPC

- **au démarrage** :
  - obtention d'un port de transport alloué par le système
  - enregistrement de ce port auprès du RPC port mapper : triplet (num\_prog, num\_version, num\_port)
- **en fonctionnement** :
  - réception des appels de procédure sur le port alloué par le système
  - envoi des résultats sur ce même port

## Client RPC

- **avant l'appel de procédure distante** :
  - contact du RPC port mapper sur le port 111
  - obtention du numéro de port du programme
- **appel de procédure distante** : contact du programme sur son numéro de port

# RPC port mapper

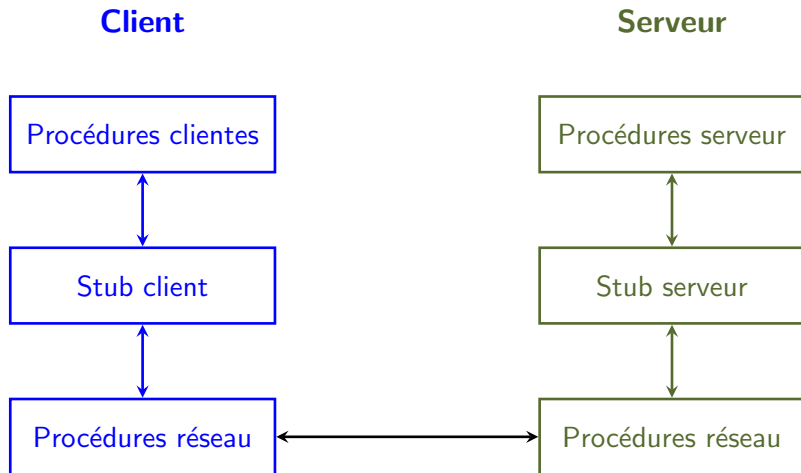
## Serveur RPC

- **au démarrage** :
  - obtention d'un port de transport alloué par le système
  - enregistrement de ce port auprès du RPC port mapper : triplet (num\_prog, num\_version, num\_port)
- **en fonctionnement** :
  - réception des appels de procédure sur le port alloué par le système
  - envoi des résultats sur ce même port

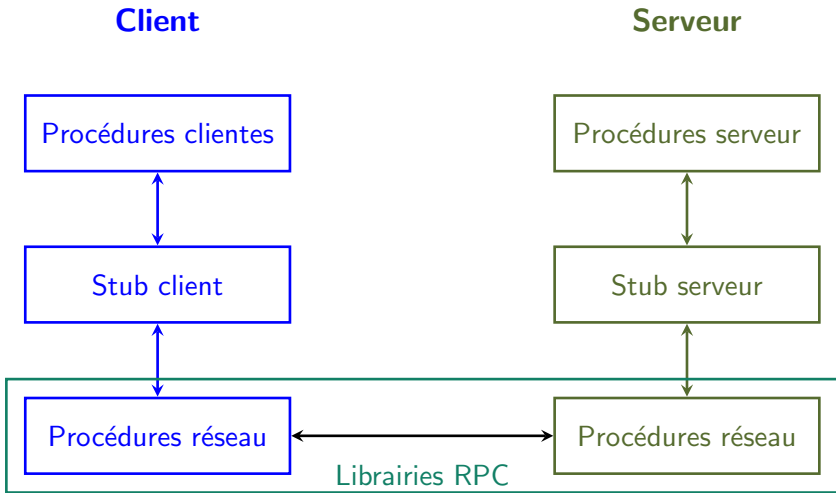
## Client RPC

- **avant l'appel de procédure distante** :
  - contact du RPC port mapper sur le port 111
  - obtention du numéro de port du programme
- **appel de procédure distante** : contact du programme sur son numéro de port

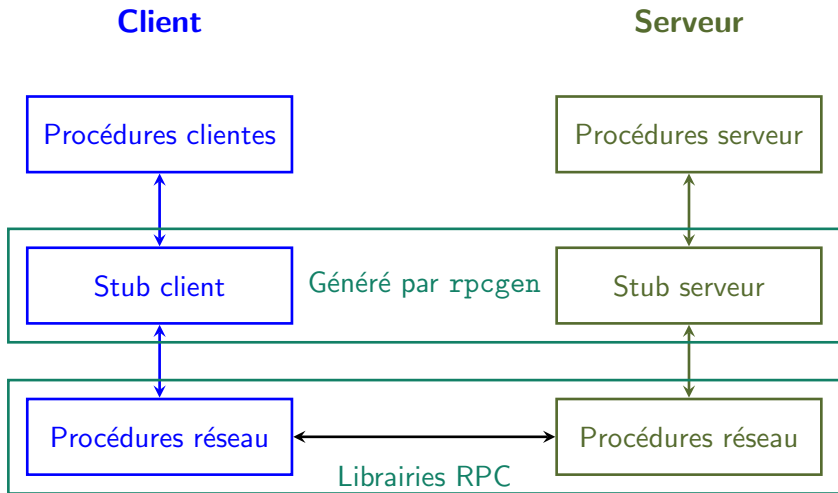
# Fonctionnement



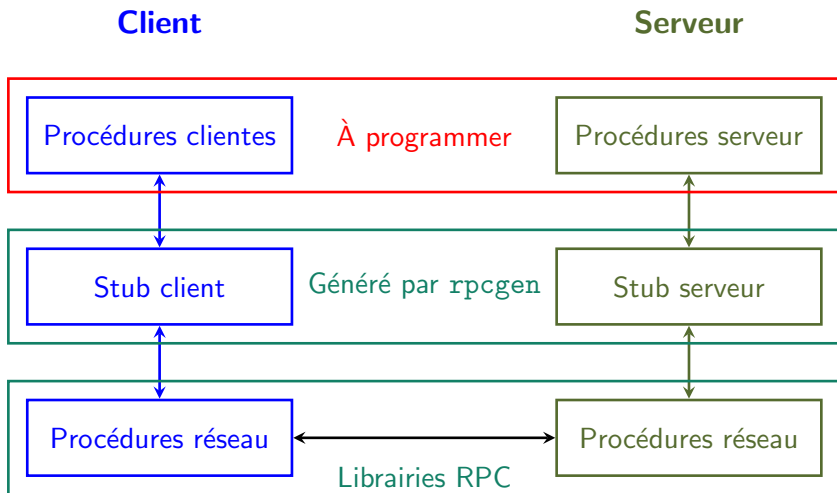
# Fonctionnement



# Fonctionnement



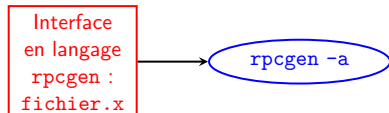
# Fonctionnement



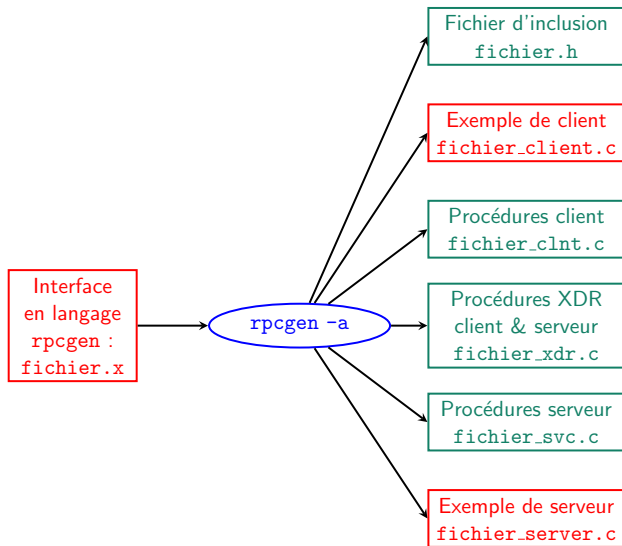
# Création de programmes avec rpcgen

Interface  
en langage  
rpcgen :  
fichier.x

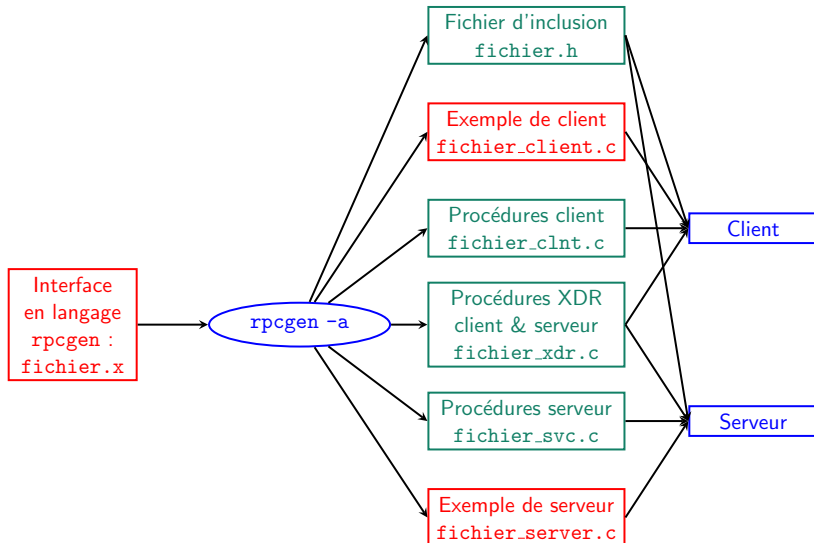
# Création de programmes avec rpcgen



# Création de programmes avec rpcgen



# Création de programmes avec rpcgen



# XDR — eXternal Data Representation

## Caractéristiques

- créé par SUN, proche du C
- **représentation des données** pour les échanges entre **machines hétérogènes**
- langage de **description des données**
- **indépendant** du matériel et des langages de programmation utilisés

## Données

- entiers sur 32 bits, codés en **big endian**, réels codés au format IEEE
- longueur des données toujours multiple de 4 octets, et utilisation de 0 de bourrage si nécessaire
- données non typées : accord nécessaire entre client et serveur
- flux d'envoi : données → codage → réseau
- flux de réception : réseau → décodage → données

# XDR — eXternal Data Representation

## Caractéristiques

- créé par SUN, proche du C
- **représentation des données** pour les échanges entre **machines hétérogènes**
- langage de **description des données**
- **indépendant** du matériel et des langages de programmation utilisés

## Données

- **entiers** sur **32 bits**, codés en **big endian**, **réels** codés au **format IEEE**
- **longueur** des données toujours **multiple de 4 octets**, et utilisation de 0 de bourrage si nécessaire
- **données non typées** : accord nécessaire entre client et serveur
- **flux d'envoi** : données → codage → réseau
- **flux de réception** : réseau → décodage → données

# Autres RPCs et applications

## Autres RPCs

- XML-RPC
- CORBA
- SOAP

## Applications

- NIS
- NFS