Technologies de l'Internet

– TR2 –

Camille Coti, Laure Petrucci
camille.coti@lipn.univ-paris13.fr

Département R&T, IUT de Villetaneuse, Université de Paris XIII

Plan du cours

Le modèle OSI (Open System Interconnection)

Un modèle normalisé

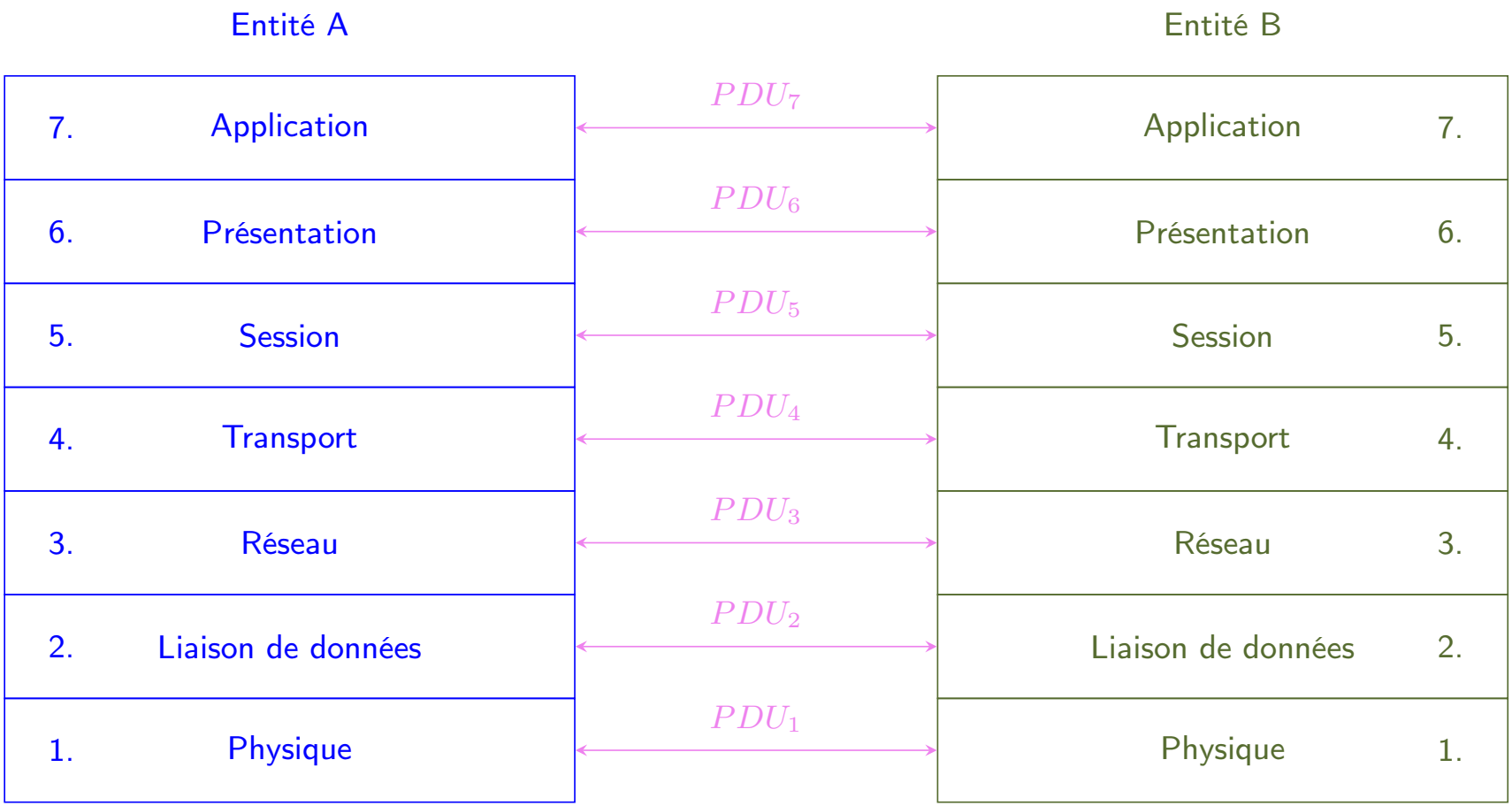
- norme de l'ISO (International Standards Organisation)
- assurer une compatibilité entre entités hétérogènes

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

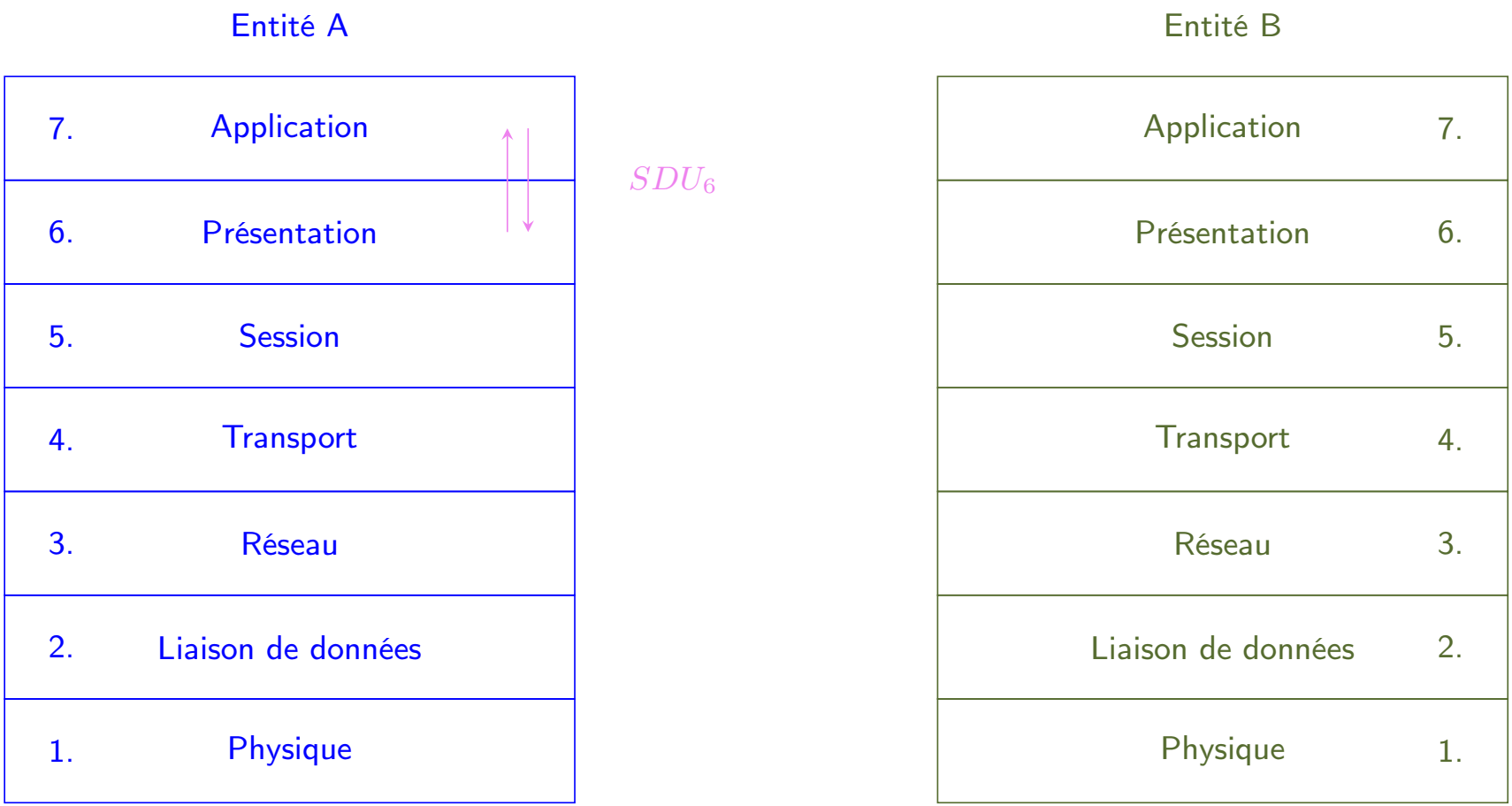
Les couches du modèle OSI



PDU — Protocol Data Unit

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

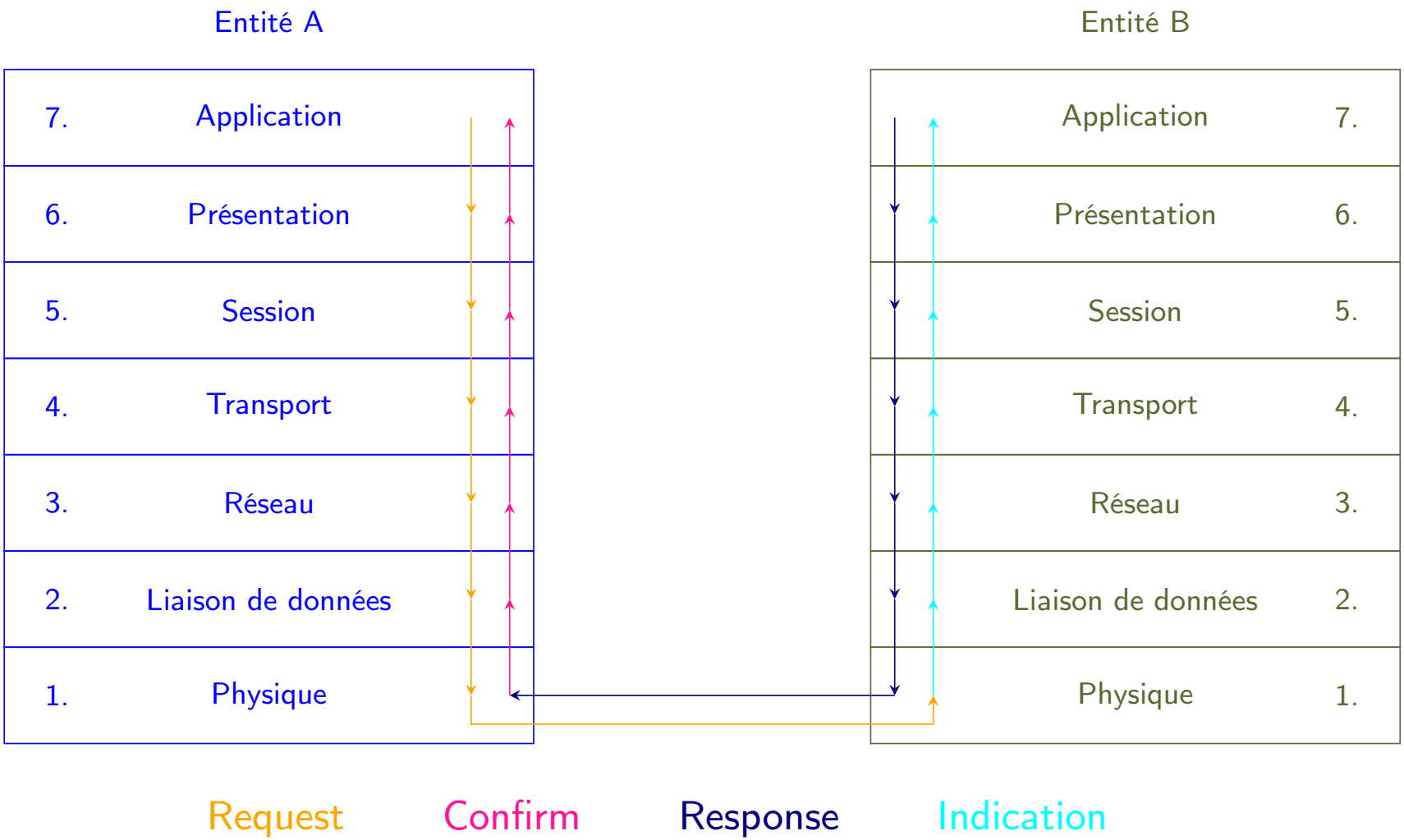
Les couches du modèle OSI



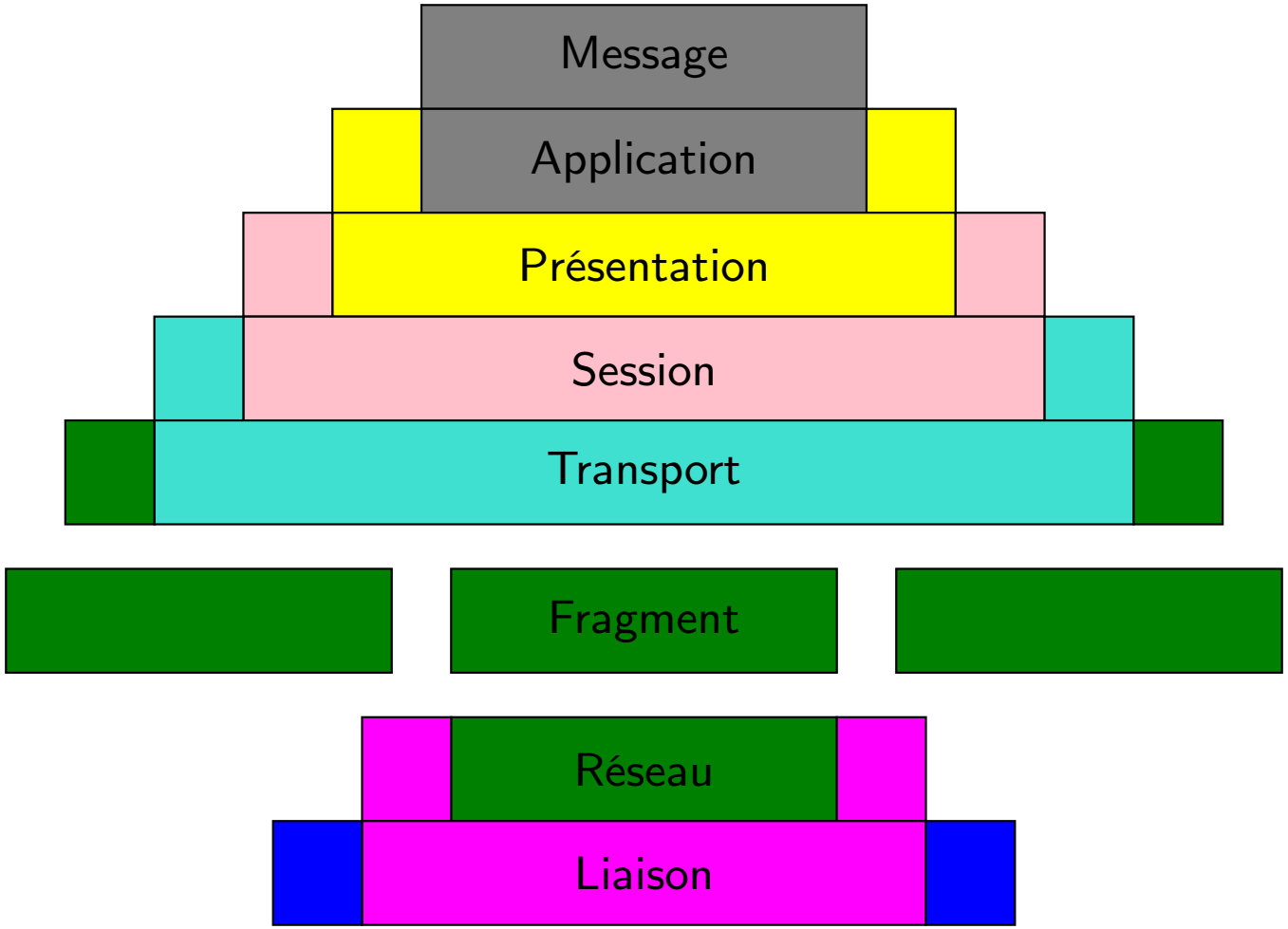
SDU — Service Data Unit

SDU_n : service fourni par la couche n à la couche $n+1$

Les couches du modèle OSI



Encapsulation



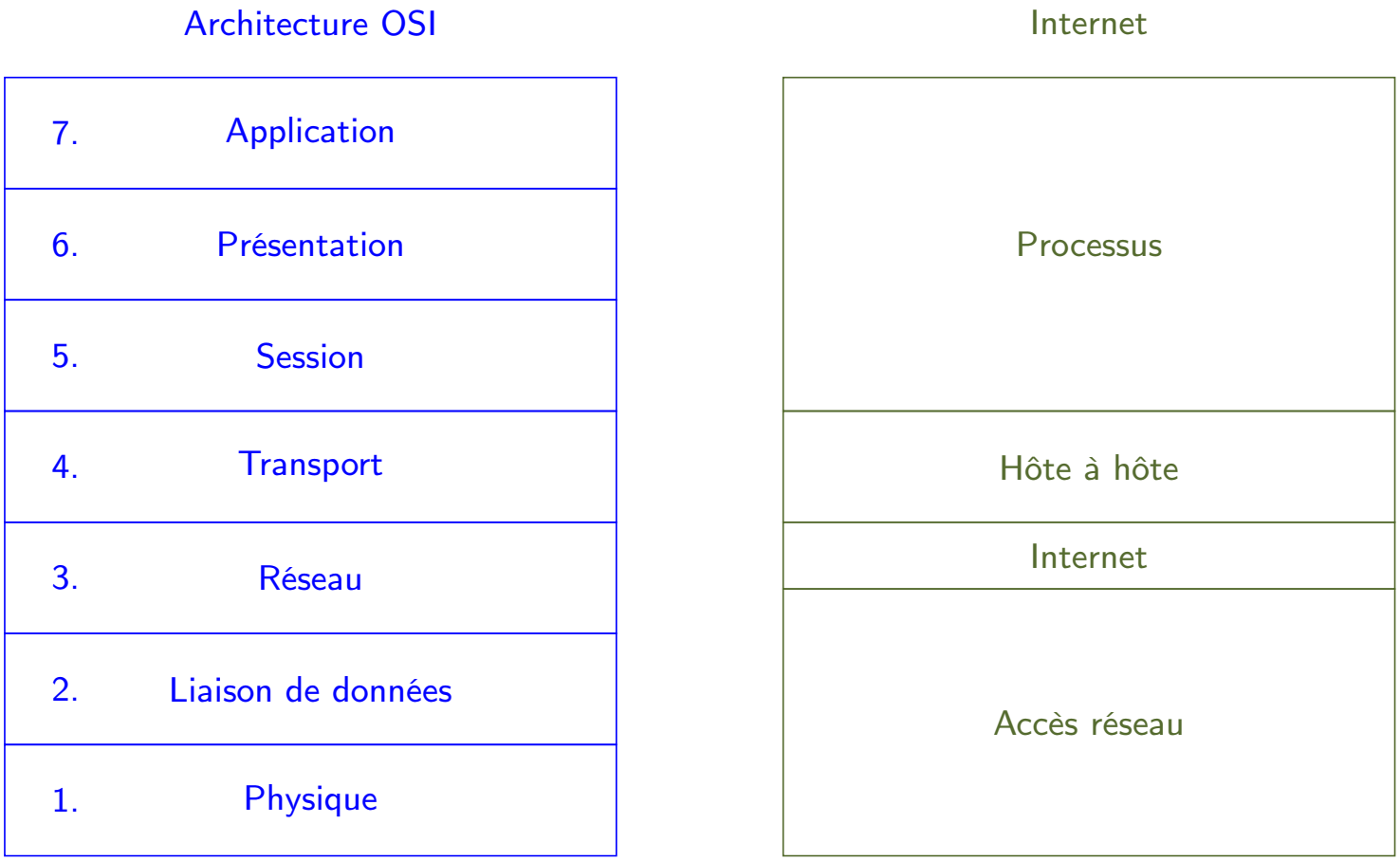
2 TCP/IP — Internet

- Petit historique
- Architecture du DoD (Internet)
- Structure d'un paquet IP
- Adresses IP
- Routage internet
- Protocoles
- Mécanisme d'une connexion TCP

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



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 (Differenciated 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

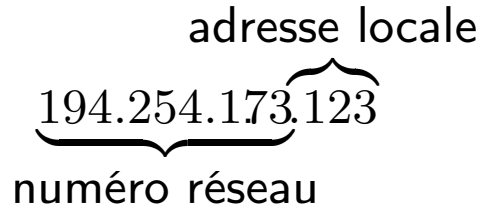
Numéro

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

Caratéristiques de l'adressage IP

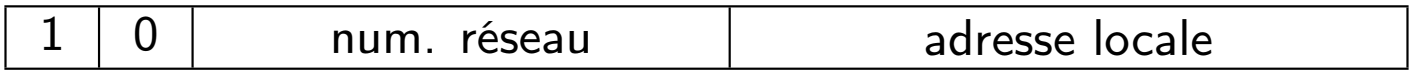
- addresses uniques (dans le monde)
- 4 octets :
  - représentés en décimal
  - séparés par des points
  - premiers octets : numéro de réseau
  - derniers octets : adresse locale de l'entité sur le réseau



Classes d'adresses

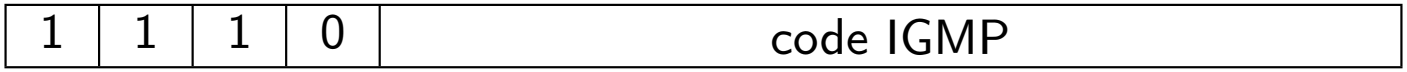
Classe A (grands réseaux) :

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 $\leftrightarrow$ 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

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 $\texttt{ARP.request}(\texttt{MAC}_A, \texttt{IP}_A, 0, \texttt{IP}_B)$
- B répond par un paquet $\texttt{ARP.reply}(\texttt{MAC}_B, \texttt{IP}_B, \texttt{MAC}_A, \texttt{IP}_A)$

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

$$\text{Unicast} \neq \text{Multicast} \neq \text{Broadcast}$$

TCP — Transmission Control Protocol

Objectif

Fournir un service de transport fiable

Comment ?

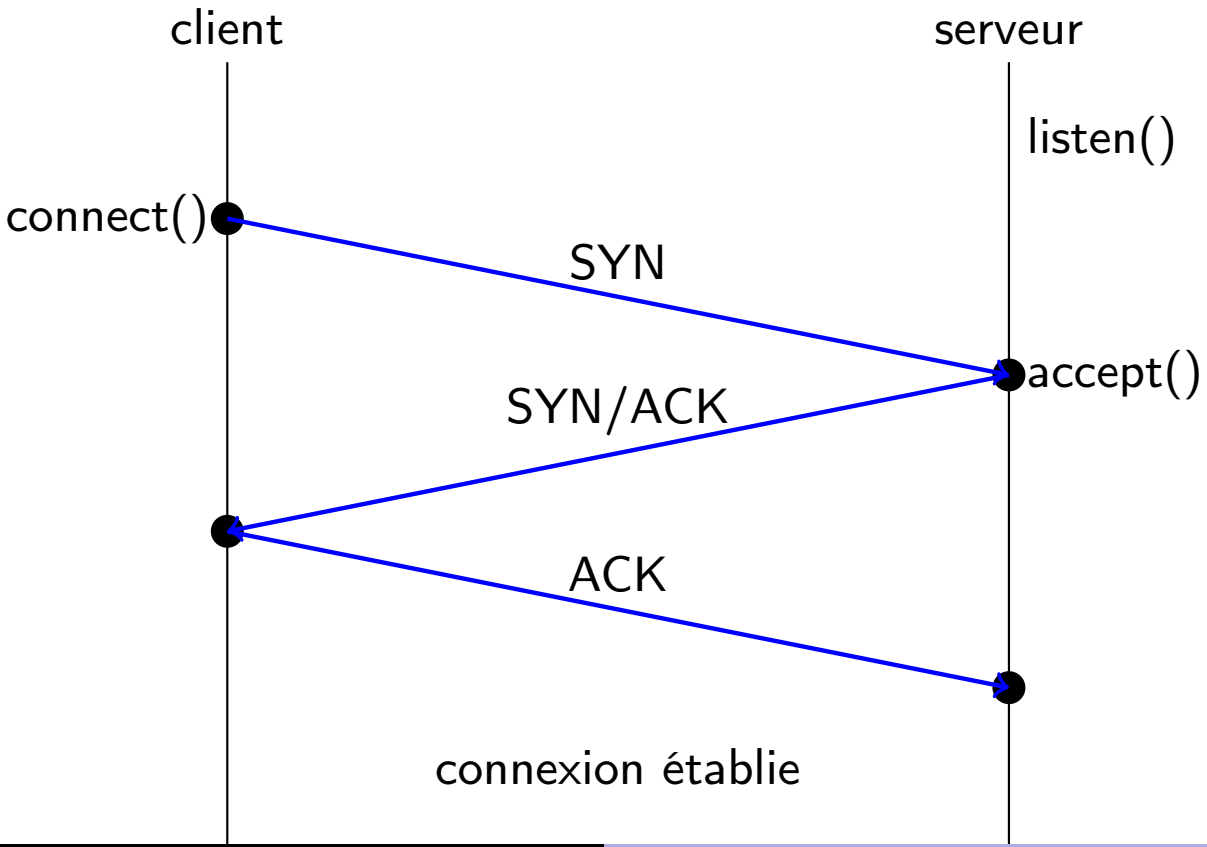
- indépendant des protocoles inférieurs
- connexions bi-directionnelles
- processus serveurs et processus clients
- communication identifiable de manière unique par :
  - adresse source
  - port source
  - adresse destination
  - port destination

Protocole TCP

Établissement de la connexion

Triple poignée de main :

- Le client envoie une requête de connexion : paquet SYN
- Le serveur accepte : paquet SYN/ACK
- Le client acquitte la réception de l'acceptation : paquet ACK



Protocole TCP

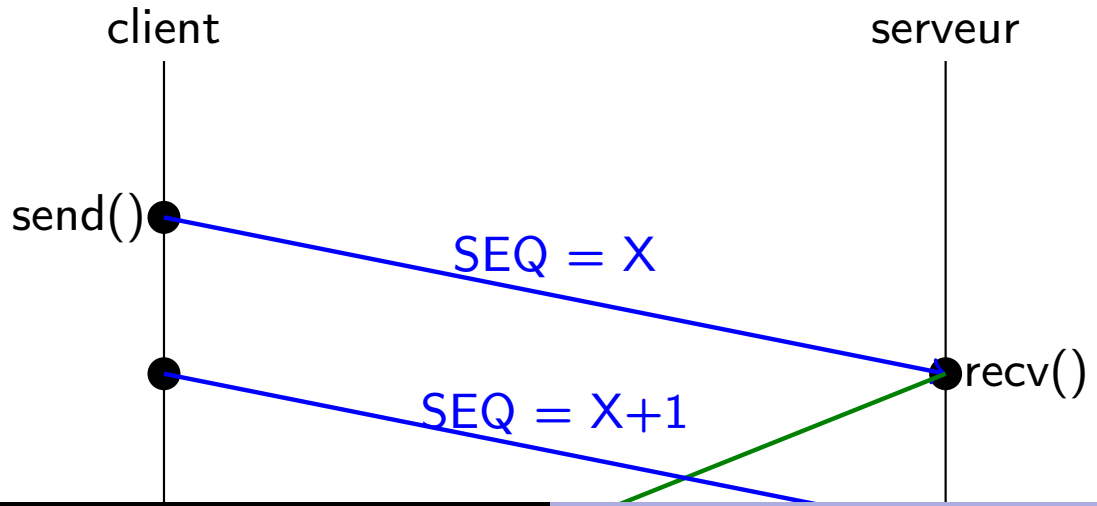
Communications

Les envois sont acquités : messages ACK

- Problèmes : messages supplémentaires, risque d'engorgement du réseau, bloquage en attente de l'acquittement

Technique de la fenêtre glissante

- Anticipation : l'émetteur continue d'envoyer des trames sans bloquer sur l'attente de l'acquittement
- Il en envoie un certain nombre avant de s'assurer qu'une trame est acquitée
- On n'acquitte pas *toutes* les trames mais seulement toutes les X trames
- Utilisation du numéro de séquence pour l'acquittement

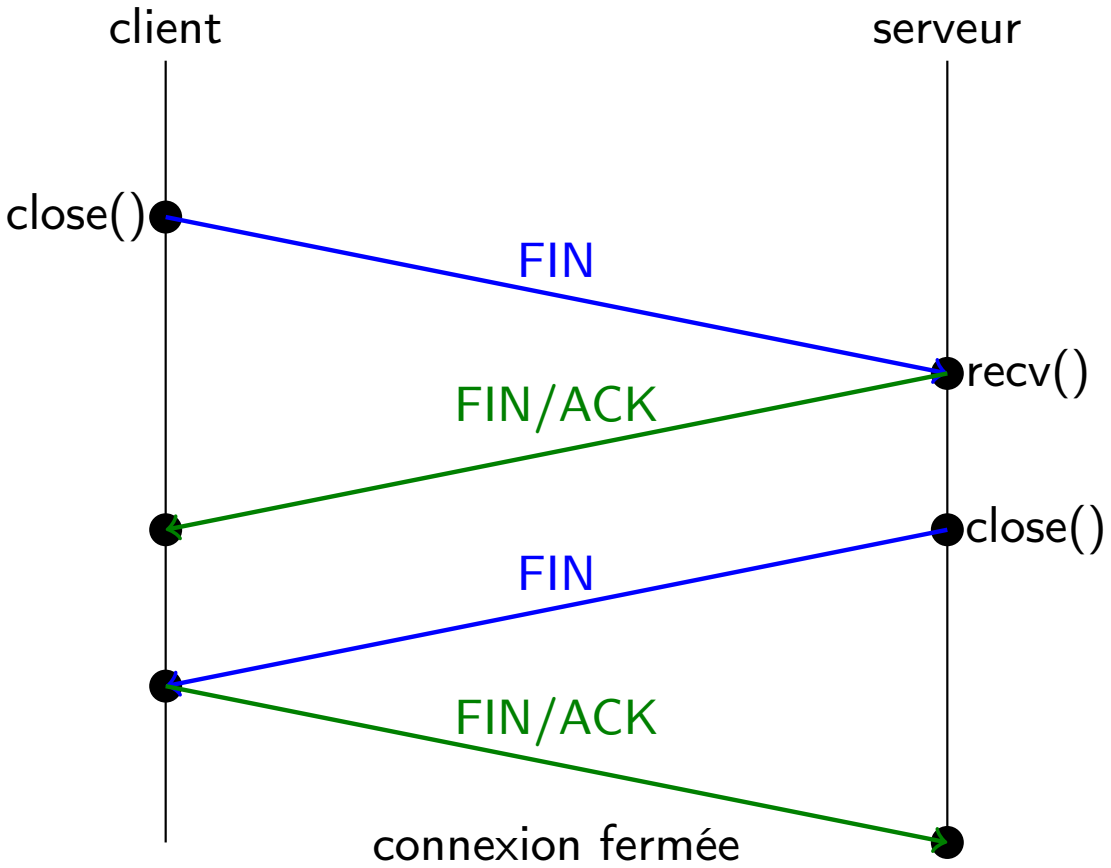


Protocole TCP

Fermeture de la connexion

Les deux machines doivent fermer la connexion de leur côté

- *two-way handwhake* (poignée de main bidirectionnelle)



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 fenêtre (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

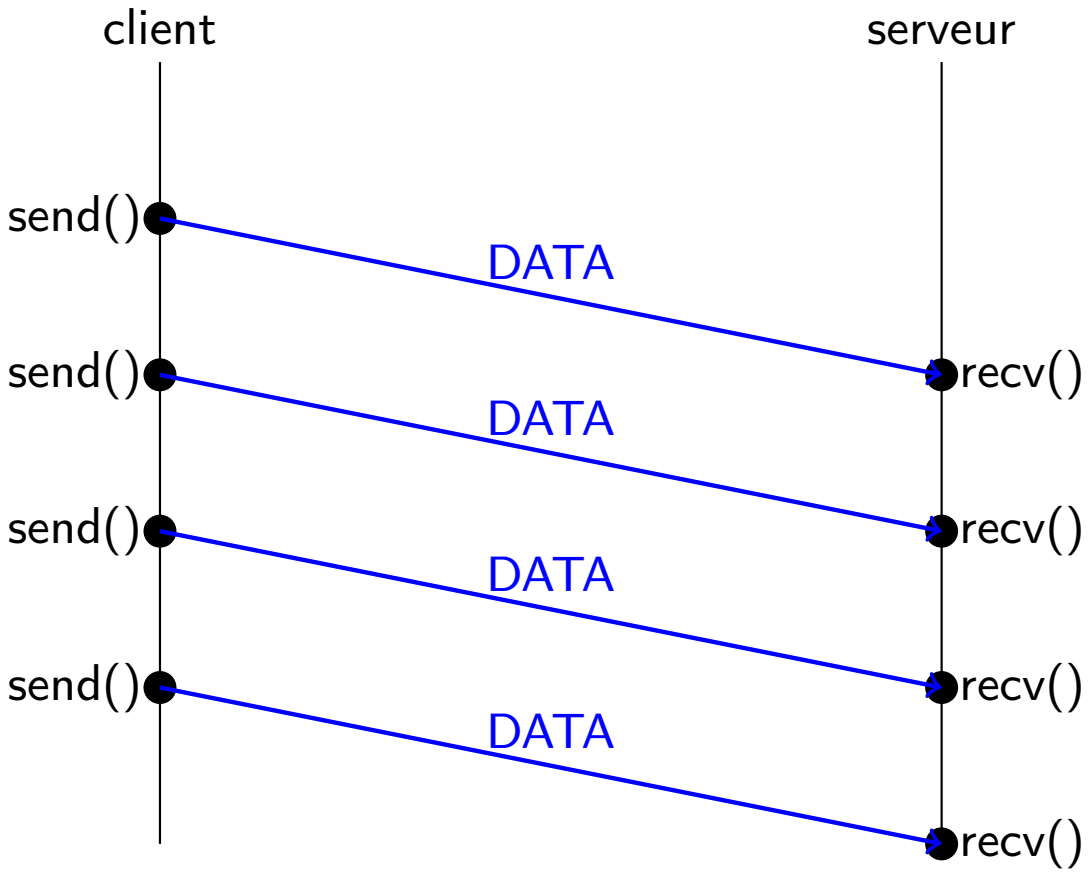
- Protocole très simple
- sans connexion
- sans acquittement
- utilisation des numéros de ports

Protocole UDP

Protocole de communications

Très simple, non fiable

- Non-connecté : pas d'établissement de connexion
- Non-fiable : pas d'acquittements



Programmation socket entre processus

Définitions

Une socket est un point terminal de communication entre deux processus.
À chaque socket sont associés :

- Un descripteur (de fichier sous Unix)
- Un nom par rapport à un domaine (une famille d'adresses)
  - Adresse IP + port (internet)
  - Nom de fichier (socket Unix)
- Un type
  - Datagramme, flot d'octets, raw
- Un protocole
  - TCP, UDP...

Définitions des structures dans le système

Définitions des structures de données des familles d'adresses :

- Famille d'adresses Unix : `/usr/include/sys.un.h`
- Famille d'adresses Internet : `/usr/include/netinet/in.h`

Définition des types :

- `/usr/include/sys/socket.h`

Mécanisme d'une communication TCP

Création de la socket : `socket()`

- Prend en paramètres le domaine, le type et le protocole
- Retourne le descripteur de fichier

À ce moment, la socket est anonyme. Il faut lui associer un nom : `bind()`

- Prend en paramètres le descripteur de fichier et le nom
- Sert à associer la socket à une adresse (port, adresse)

On doit ensuite établir la connexion.

Côté serveur :

- On écoute sur la socket

```
listen( fd, backlog );
```

- On accepte les connexions entrantes sur la socket d'écoute

```
fd2 = accept( fd, &addr,
&addrlen );
```

Côté client :

- On se connecte au serveur

```
connect( fd, addr, addrlen );
```

Attention ! `accept()` retourne une socket : c'est sur cette socket que la connexion est établie.

Communications sur socket TCP

Envoi de données :

- Fonction `send()`
- Prototype :
  ```
  ssizt_t send(int sockfd, const void *buf, size_t len, int flags);
  ```
- On passe
  - la socket sur laquelle on envoie
  - un pointeur vers le début de la zone à envoyer
  - la taille de la zone à envoyer (en octets)
  - un drapeau (options)
- Retourne le nombre d'octets effectivement envoyés

Réception de données :

- Fonction `recv()`
- Prototype :
  ```
  ssize_t recv(int sockfd, void *buf, size_t len, int flags);
  ```

Une fois qu'on a fini de communiquer, il faut fermer la socket

- Fermeture des deux côtés
- `close( fd );`

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

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 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 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 chaîne 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)?

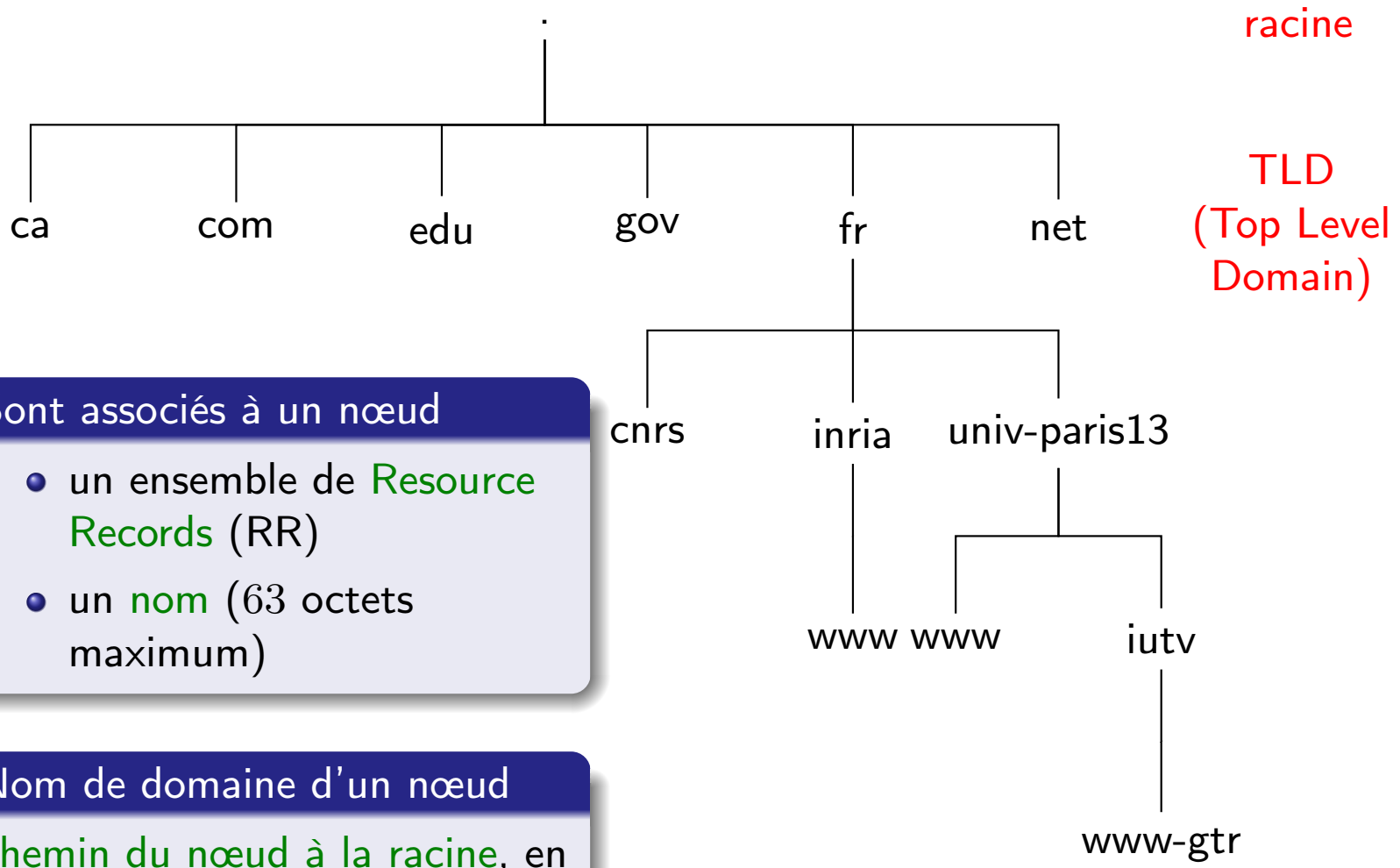
- adresses IP difficiles à retenir, peu explicites
- outil pour effectuer de la résolution d'adresses :
  - symbolique → IP
  - IP → symbolique
- facilité de gestion et de structuration des adresses symboliques.

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 .

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 »)

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

- récursif : 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

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
  - 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
- ...

Configuration de zone — fichier

```
$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
```

Configuration de zone — explications

- $ORIGIN : domaine par défaut
- $TTL : TTL de 3 jours (3 days) par défaut
- @ : 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

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, 0002 = 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

Dynamic Host Configuration Protocol

Protocole permettant la configuration réseau automatique d'une machine

- Au minimum : adresse IP et masque de sous-réseau (IPv4 et IPv6)
- Potentiellement : passerelle par défaut, nom DNS, serveurs DNS
- L'adresse est attribuée pendant une durée fixe : bail

Avantages :

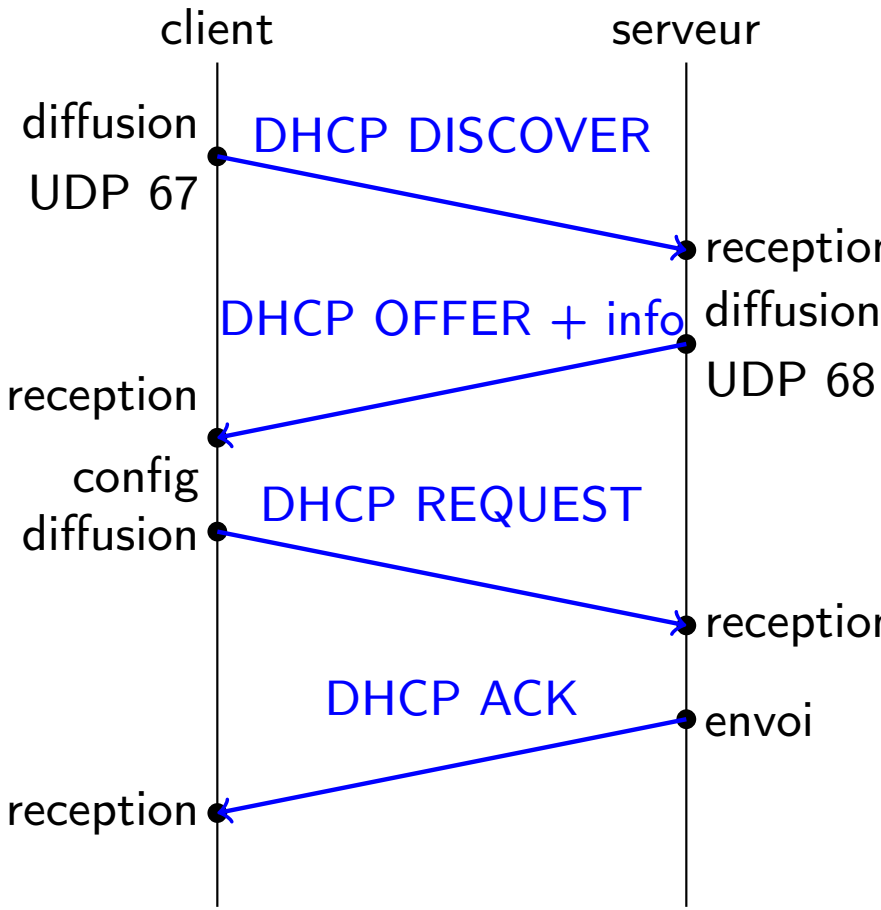
- Pas besoin d'intervenir individuellement sur les machines
- Gestion d'un pool d'adresses réduit
- Évite les conflits d'IP (1 machine = 1 adresse IP)

Mais nécessite un serveur DHCP pour orchestrer et attribuer les adresses.

Protocole

- Protocole réseau : UDP
- Ports utilisés : 67 pour le serveur, 68 pour les clients

- Le client diffuse un datagramme DHCP DISCOVER pour le port 67
- Les serveurs reçoivent le datagramme et diffusent un datagramme DHCP OFFER pour le port 68 : proposition de configuration
- Le client choisit un serveur parmi les offres et diffuse un datagramme DHCP REQUEST contenant l'IP du serveur retenu et celle du client. Les autres serveurs le reçoivent et apprennent qu'ils n'ont pas été retenus.
- Le serveur acquitte avec un datagramme DHCP ACK envoyé au client.



6 Gestion du courrier — SMTP

- Fonctionnalités à assurer
- SMTP — Simple Mail Transfer Protocol
- POP — Post Office Protocol
- IMAP — Internet Message Access Protocol

Historique

La préhistoire :

- Envoi de messages entre différents utilisateurs des mainframes à la fin des années 60 (MAILBOX)
- Premier envoi d'un message entre deux machines du réseau ARPANET courant 1972 (1ère utilisation du caractère @)
- En 1973, les mails représentent les trois quarts du trafic ARPANET
- Nombreux problèmes d'interropérabilité (RFC 680, 724, 733, etc.)
- Eric Allman développe delivermail (utilisation de FTP over NCP pour transmettre les messages)

Contribution de Nicolas Grenèche

Historique

Les temps anciens (... ou l'avènement de TCP/IP) :

- Avant la naissance du protocole SMTP, la transmission des mails se fait au moyen de FTP, UUCP, etc.
- Eric Allman développe sendmail
- La RFC 821 donne les premières spécifications du protocole SMTP (Simple Mail Transfer Protocol) (en 1982)
- La RFC 822 spécifie le format des messages

Contribution de Nicolas Grenèche

Historique

Les temps modernes...

- L'évolution vers la messagerie actuelle :
  - Encapsulation de contenus multimédias (MIME)
  - Interconnexion avec les annuaires LDAP, Active Directory
- Les besoins émergents en terme de sécurité :
  - Authentification, confidentialité, etc.
  - Filtrage applicatif : spam, antivirus, etc.

Contribution de Nicolas Grenèche

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

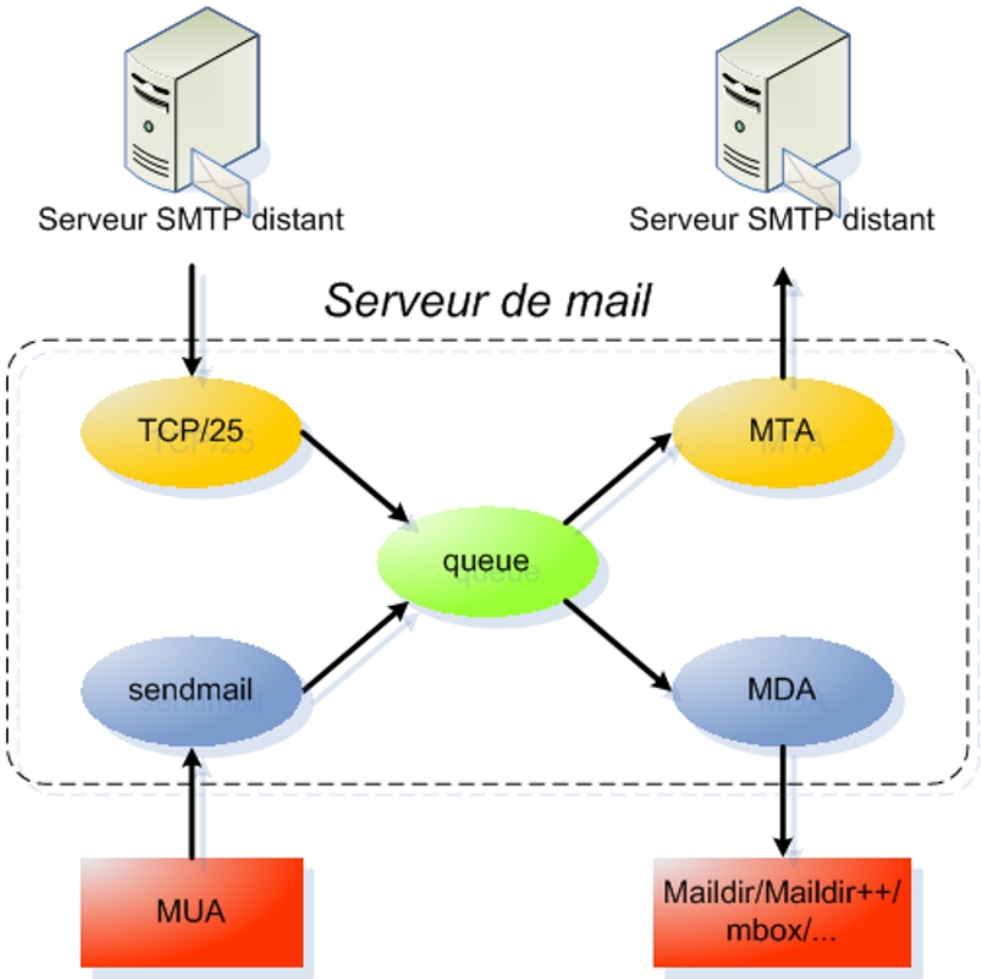
Caractéristiques de SMTP

- 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

- MTA (Mail Transfer Agent)
  - Programme responsable de l'acheminement des messages
- MUA (Mail User Agent)
  - Programme permettant à l'utilisateur de lire et d'envoyer des messages
  - Dialogue avec le MTA via le protocole SMTP (client uniquement) et avec le serveur de boîtes aux lettres via POP ou IMAP
- MDA (Mail Delivery Agent)
  - Interface entre le MTA et la boîte aux lettres de l'utilisateur

Architecture d'un serveur de mail



Contribution de Nicolas Grenèche

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

ESMTP (Extended SMTP)

Extensions définies dans la RFC 1425, maintiennent une compatibilité ascendante avec SMTP

- Utilisation de EHLO à la place de HELO, puis code réponse 250

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

Le format des messages 1/3

Un message est constitué de trois parties :

- L'enveloppe est utilisée par le MTA pour l'acheminement du courrier, il s'agit essentiellement de l'expéditeur et du destinataire
- Les en-têtes (ou headers) sont utilisés par les MUA
- Le corps est le contenu du message envoyé au destinataire

Contribution de Nicolas Grenèche

Le format des messages 2/3

- Le format et l'interprétation des en-têtes sont spécifiés dans la RFC 822 (1982)
  "822 allows much more flexibility than a typical mail-reading program can actually handle; meanwhile it imposes restrictions that millions of users violate every day"
- Les en-têtes sont de la forme "Nom: valeur"

Exemple :

Date: Wed, 30 Aug 2006 14:03:22 +0200 (CEST)
X-Spam-Flag: YES
From: Bob Marley <bob@marley.com>
To: Mathieu Blanc <moutane@rstack.org>
Subject: Fw: failure notice

Humm... ?!

Contribution de Nicolas Grenèche

Le format des messages 3/3

- Suivi de l'acheminement sur chaque serveur de transit
  - Provenance du message
  - Identification du serveur
  - Destinataire
  - Heure de réception
- MIME : encapsulation des pièces jointes
  - Encodage en base64
  - Séparateurs
  - Indication de type de contenu

Contribution de Nicolas Grenèche

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) : autres destinataires, en copie, mais dont l'adresse est masquée

Le routage des mails et le DNS 1/2

Problème : à quel serveur doit-on s'adresser pour envoyer un mail destiné à "local@domaine.com" ?

- Le MTA effectue une requête DNS afin de connaître l'enregistrement MX de "domaine.com"

Commande :

```
$ host -t mx gmail.com
gmail.com mail is handled by 5 gmail-smtp-in.l.google.com.
gmail.com mail is handled by 10 alt1.gmail-smtp-in.l.google.com.
gmail.com mail is handled by 10 alt2.gmail-smtp-in.l.google.com.
```

- En cas d'échec de la requête DNS, cela ne veut pas dire qu'il n'y a pas d'enregistrement MX pour le domaine concerné : le client doit réessayer un peu plus tard...
- Si la requête DNS ne renvoie aucun enregistrement MX, alors le MTA doit prendre comme MX l'hôte lui-même

Contribution de Nicolas Grenèche

Le routage des mails et le DNS 2/2

- Le serveur prend ensuite la liste des serveurs destinataires par ordre croissant de préférence et se connecte ensuite sur le port TCP/25 pour envoyer le message par SMTP
  - En cas d'échec permanent, le client retourne un message d'erreur à l'expéditeur (bounce message)
  - En cas d'échec temporaire, le client réessaye d'envoyer le message en utilisant les serveurs de mails ayant une priorité inférieure. S'il ne restent plus de serveurs dans la liste, le message est alors mis en attente
- L'utilisation des MX est spécifiée dans la RFC 974

Contribution de Nicolas Grenèche

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

Accès aux boîtes aux lettres : POP3 1/5

- POP3 = Post Office Protocol version 3
  - RFC 1939
  - Protocole permettant à un programme local de récupérer les mails stockés dans une BAL distante (ie. présente sur un serveur distant)
- Utilise le port TCP/110
- Encore majoritairement utilisé par les ISP :
  - Le protocole est très simple, donc peu d'incompatibilités avec les programmes clients
  - Le fonctionnement par défaut favorise le contrôle de la taille des boites aux lettres
  - Convient parfaitement à un accès à Internet non-permanent

Contribution de Nicolas Grenèche

Accès aux boîtes aux lettres : POP3 2/5

- Les mécanismes d'authentification POP3
  - Comme beaucoup de protocoles un peu anciens, le mot de passe est transmis en clair par défaut
  - La commande (optionnelle) APOP
  - Permet une authentification par challenge/response
  - Le serveur envoie une chaîne contenant (entre-autre) un marqueur de temps dans sa bannière de connexion
    De la forme : <marqueur-unique@nom-serveur>
  - Le client concatène ce marqueur et son mot de passe et renvoie le résultat au serveur qui peut ainsi vérifier son identité

Contribution de Nicolas Grenèche

Accès aux boîtes aux lettres : POP3 3/5

Exemples :

```
$ nc pop.free.fr 110
+OK <14876.1102792612@pop2-q.free.fr>
+OK <14876.1102792612@pop6-g25.free.fr>


$ nc pop.wanadoo.fr 110
+OK connected to pop3 on 0401
+OK connected to pop3 on 0404
```

Contribution de Nicolas Grenèche

Accès aux boîtes aux lettres : POP3 4/5

Exemples (suite) :

```
$ nc pop.laposte.net 110
+OK POP3 server ready (7.0.028) <7FA75DAB...26@mx.laposte.net>
+OK POP3 server ready (7.2.060.1) <73D3317A...@mx.laposte.net>


$ nc pop.tiscali.fr 110
+OK POP3 server ready (7.1.026) <04B411DC18@mail.libertysurf.net>
+OK POP3 server ready (7.1.026) <8500B05B45@mail.libertysurf.net>


$ nc pop.club-internet.fr 110
+OK POP3 FLASH Server
+OK POP3 FLASH Server
```

Contribution de Nicolas Grenèche

Accès aux boîtes aux lettres : POP3 5/5

- Quelques extensions au protocole POP
  - POP over SSL (commande STARTTLS)
  - Autres mécanismes d'authentification (SASL)
  - Login delay

Contribution de Nicolas Grenèche

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

Accès aux boîtes aux lettres : IMAP 1/2

- IMAP = Internet Message Access Protocol
  - La version courante est la version 4 révision 1 généralement nommée IMAP4rev1 (RFC 3501)
- Caractéristiques
  - TCP/143 (ou TCP/993 pour imap over SSL)
  - Les messages restent sur le serveur, l'utilisation se fait généralement en mode connecté
  - Boîtes aux lettres partagées, accès concurrentiels, etc.
  - Création de dossiers IMAP résidant sur le serveur

Contribution de Nicolas Grenèche

Accès aux boîtes aux lettres : IMAP 2/2

- Contraintes d'exploitation
  - Gourmand en ressources
    - Au niveau système, il s'agit d'un protocole complexe et certaines fonctionnalités (s'exécutant sur le serveur) peuvent être coûteuses (tri, threading, etc.)
    - Maîtrise de la taille des BALs
    - Au niveau réseau, de multiples connexions IMAP peuvent être ouvertes simultanément lors de l'accès à une BAL
  - La compatibilité avec le protocole peut se révéler très différente entre les divers clients de messagerie
    - Protocole complexe

Le problème du Spam 1/6

- Pourriel, pollupostage, etc
  - Courrier électronique non sollicité, envoyé le plus souvent massivement, sans l'accord des destinataires
  - Représente 60% du trafic mail sur internet
- Le Spam est avant tout un problème économique
  - Le coût d'envoi d'un message est très faible
  - Le stockage des messages est à la charge des destinataire
- A l'heure actuelle, il n'existe aucune solution définitive au problème du Spam
  ... mais différentes techniques peuvent être mises en place au niveau du serveur de mail

Contribution de Nicolas Grenèche

Le problème du Spam 2/6

- Différents types de nuisances peuvent être considérées comme du Spam :
  - Informations à caractère commercial
  - Entreprises peu scrupuleuses vis-à-vis de la Netiquette
  - Escroqueries diverses
  - Pornographie, médicaments, crédit financier, etc.
  - Phishing (password harvesting fishing)
  - Obtention d'informations confidentielles (ex. : numéro de carte bancaire) en se faisant passer auprès des victimes pour quelqu'un digne de confiance
  - Hoax (ou canular informatique)

Contribution de Nicolas Grenèche

Le problème du Spam 3/6

- Méthode : vérifications DNS
  - Requête de type MX sur le domaine de l'adresse mail de l'expéditeur
  - Recherche du domaine à partir de l'adresse IP et comparaison avec le champ From: du message
    - Dans les 2 cas, les risques de faux-positifs sont très importants
    - Interopérabilité

Contribution de Nicolas Grenèche

Le problème du Spam 4/6

- Méthode : Black list IP
  - RBLs (Realtime Blackhole Lists)
  - Principe : pour chaque mail entrant, rechercher l'adresse IP de l'expéditeur dans des listes publiques d'adresses IP de spammers connus
  - S'appuient généralement sur le DNS (DNSRBLs)
  - Le déploiement est simple, relativement peu coûteux en terme de ressources CPU, et l'impact au niveau réseau reste faible
  - Inconvénients :
    - Risques importants de faux-positifs
    - La mise à jour des listes publiques est une intervention humaine

Contribution de Nicolas Grenèche

Le problème du Spam 5/6

- Méthode : Greylisting
  - Technique relativement récente et assez efficace
  - Codes d'erreurs SMTP (RFC 821) :
    - 1yz, 2yz, 3yz : Positive Reply
    - 4yz : Transient Negative Completion Reply
    - 5yz : Permanent Negative Completion Reply
  - Les sources de Spam ne prennent pas en compte les erreurs SMTP temporaires
  - Fontionne aussi pour les virus de messagerie

Contribution de Nicolas Grenèche

Le problème du Spam 6/6

- Méthode : Filtrage applicatif
  - Recherche de mot-clés spécifiques
  - Efficace mais facilement contournable, par exemple remplacer le terme Viagra par V1agra
  - Rule-based scoring systems
  - Systèmes capables d'éliminer 90% du Spam
  - Exemple : SpamAssassin
    - Filtrage basé sur des méthodes probabilistes
    - Filtrage bayésien

Contribution de Nicolas Grenèche

Historique

Origine étymologique : un sketch des Monty Python

- Dans un café donc tous les éléments du menu sont à base de viande en conserve Spam

1er spam : télégraphique (1864)

- Envoi d'un télégraphe à plusieurs destinataires autorisé par Western Union

Premier spam de l'histoire : 1978

- Publicité pour DEC envoyée sur ARPANET à 393 destinataires *en même temps*

Plusieurs sortes de spam :

- Publicité pour des sites web
- Vente de produits plus ou moins légaux
- Escroquerie, extorsion d'information (phishing)

Produits vantés par le spam :

- Médicaments : 81,00%
- Reproductions : 5,40%
- Améliorateurs de performances : 2,30%
- Phishing : 2,30%
- Diplômes : 1,30%
- Casino : 1,00%
- Perte de poids : 0,40%
- Autres : 6,30%

Étendue du problème à l'échelle d'Internet

TODO

Exemple : le cas de McColo

D'après Symantec : 85% du trafic d'email mondial est du spam
McColo : service d'hébergement Web, basé en Californie

- Fermé en novembre 2008 (déconnecté d'Internet)
- Ses serveurs généraient une *grande* quantité de spam

Reponsable à l'époque de 70% du spam mondial !!!

Depuis : spam retourné au même niveau.

Quantité de spam mondial fin 2008 :

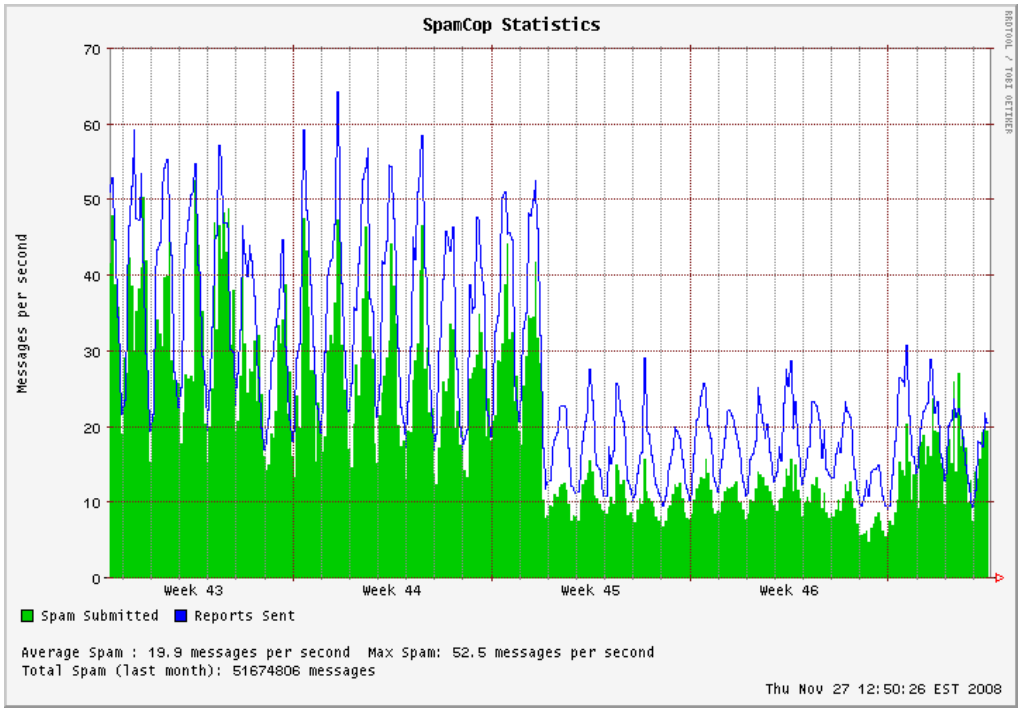


Image : Wikipedia

Le cas du port 25

Le port dédié à SMTP est le port TCP 25

- Mais pas d'authentification / contrôle d'identité sur SMTP

Les MTA spammers peuvent donc utiliser le port 25 pour envoyer du spam via un serveur SMTP donné

- Idée : bloquer le port TCP 25
  - Au niveau des firewalls : en entrée comme en sortie
  - Au niveau des routeurs : ne pas router les paquets
- En cas de NAT : on regarde ce qui est encapsulé

DNS blacklisting

Idée de base : recenser les IP des MTA des spammers et les blacklister

- Problème : comment y accéder rapidement pour déterminer si on laisse passer ou non ?

DNSBL : DNS Blackhole List

- Quand le MTA va envoyer un mail à une adresse `local@domaine.com`, il interroge le serveur DNS de `domaine.com`
- Les IP des MTA connus pour envoyer ou relayer du spam sont recensées dans une zone DNS sur un nom de domaine donné (par exemple, `bl.spamcop.net`)
- Faible latence, (quasiment) toujours disponible, supporte beaucoup de requêtes
- Aujourd'hui : plus de 70 telles bases de données

Efficacité du blacklisting : bloque environ 80% du spam

Honeypot

Idée : attirer les spammers

On crée une adresse mail "un peu trop attirante" : `communication@domaine.com`, `direction@domaine.com`...

- Les robots spammers vont essayer ce genre d'adresses pour spammer les utilisateurs du domaine `domaine.com`
- Dans la boite aux lettres on récupère les emails reçus et on extrait les adresses IP des MTA expéditeurs
- Constitution d'une base d'adresses IP à blacklister

Image : Wikipedia

Greylisting

Les serveurs de spams ou machines ayant un virus ne gèrent pas les files d'attentes de messages en ne réessayant pas un renvoi lors d'un échec temporaire (erreur SMTP 4XX)

- Principe du greylisting : rejeter temporairement tout message arrivant d'un MTA inconnu

Utilisation d'un triplet (IP du MTA, adresse email du destinataire, adresse email de l'expéditeur)

- Si le triplet est inconnu, greylisting du triplet et message rejeté temporairement
- Si il est connu (liste blanche), on laisse passer
- Si le triplet est dans la greylist : on le met dans la liste blanche et on laisse passer

Conséquence : la première fois qu'un MTA envoie un message, celui-ci va arriver en retard (mais l'email est par nature asynchrone)

Filtrage bayesien

Utilisation de probabilités (formule de Bayes) :

- Sachant qu'un message contient un mot donné, quelle est la probabilité qu'il soit un spam ?

Phase d'apprentissage :

- L'anti-spam règle les coefficients de sa matrice de classification
- Possibilité d'ajuster ensuite avec les erreurs (faux positifs / non-détectés)

Utilisé par Spamassassin, très efficace

Présentation de Spamassassin

Script tentant de séparer le bon et le mauvais mail

- Mécanisme de notation pour chaque email
- Utilisation d'un ensemble de règles
- Chaque règle a un score
- Si la somme des scores dépasse un certain seuil : l'email est considéré comme spam

Différents types de règles :

- Blacklists
- Pattern matching
- Filtrage bayésien
- Modules externes (Razor, Pyzor...)

Spamassassin : blacklists

Blacklister manuellement une adresse email en particulier :

- `blacklist_from spammer@spamdomain.com`
- Groupes d'adresses : `*@spamdomain.com`, `*@*.spamdomain.com`...
- Possibilité de le mettre en place au niveau d'une machine ou d'un utilisateur : dans `/etc/mail/spamassassin/local.cf` ou `$HOME/.spamassassin/user_prefs`

Le fait d'être blacklisté donne automatiquement un score supérieur à 100 : toujours rejeté.

Utilisation d'une blacklist publique (DNSBL)

- Déclaration des BL dans le fichier `20_dnsbl_tests.cf`
- Requête DNSBL pour chaque message transitant par SA
  - Installer un DNS cache pour accélérer

Exemple de déclaration de blacklist

```
header __RCVD_IN_ZEN eval:check rbl('zen', 'zen.spamhaus.org.')
describe RCVD IN ZEN Received via a relay in Spamhaus Zen
tflags __RCVD_IN_ZEN net
reuse __RCVD_IN_ZEN
```

RBL = Real-time Blackhole List

On regarde si l'email a transité par un relai qui est recensé dans cette base DNSBL

- Si c'est le cas, le message aura l'étiquette `__RCVD_IN_ZEN`

Spamassassin : pattern matching

Travaille à la fois sur l'en-tête et le corps du message

- Un modèle (pattern) est défini avec un langage d'expressions régulières (Perl Regex)
- SA regarde si on le retrouve dans le message
- Patterns fournis, possibilité d'en ajouter

Définition d'une règle :

- Domaine d'application de la règle (body, header, uri, rawbody)
- Identifiant
- Expression régulière à rechercher

Et on donne le score à appliquer à cette règle

Exemples :

- `header __LOCAL_FROM_NEWS From ~= /news@example\.com/i`
- `body __LOCAL_DEMONSTRATION_RULE /\btest\b/i`
- `score __LOCAL_DEMONSTRATION_RULE 0.1`

Spamassassin : Bayes

Repose sur le filtrage bayésien et l'apprentissage

- Décomposition des messages pour analyser la fréquence des mots
- Analyse statistique sur les mots pour déterminer la probabilité qu'il s'agisse d'un spam

Filtrage antispam distribué

Distributed Checksum Clearinghouse (DCC)

Architecture composés de deux parties :

- Un client, sur chaque client mail
- Un serveur, quelque part sur le réseau

Algorithme de *fuzzy checksum* : on calcule un checksum sur une partie du message

- Pas son intégralité pour ne pas prendre en compte des parties qui pourrait être sujettes à des altérations mineures ("cher blablabla")

Chaque client calcule un fuzzy checksum et prend une emprunte des parties classiques de l'en-tête des emails reçus et interroge le serveur

- Si un client a déjà signalé ce message comme était un spam, on le considère comme spam

Quand un client reçoit un spam, il le déclare comme tel

- Remontée au serveur que ce message est un spam

Efficacité : détecte 37% du spam passant par Spamassassin

8 Annuaires — LDAP

- Notion d'annuaire
- Annuaires X500
- LDAP
- Network Information System

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

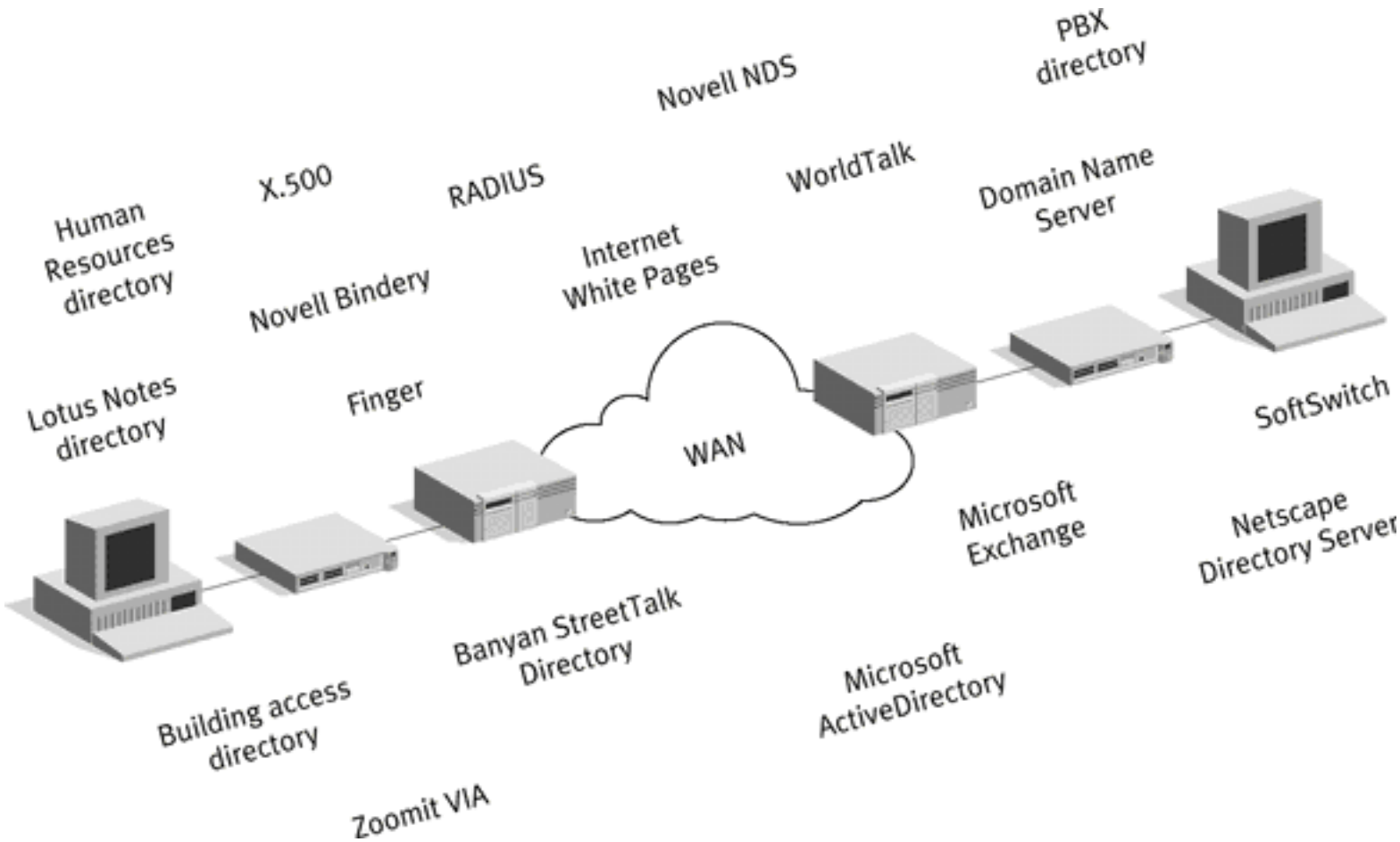
Informations d'un annuaire

- 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 $\leftrightarrow$ 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)

Annuaires X500

X500 : norme ISO (1988)

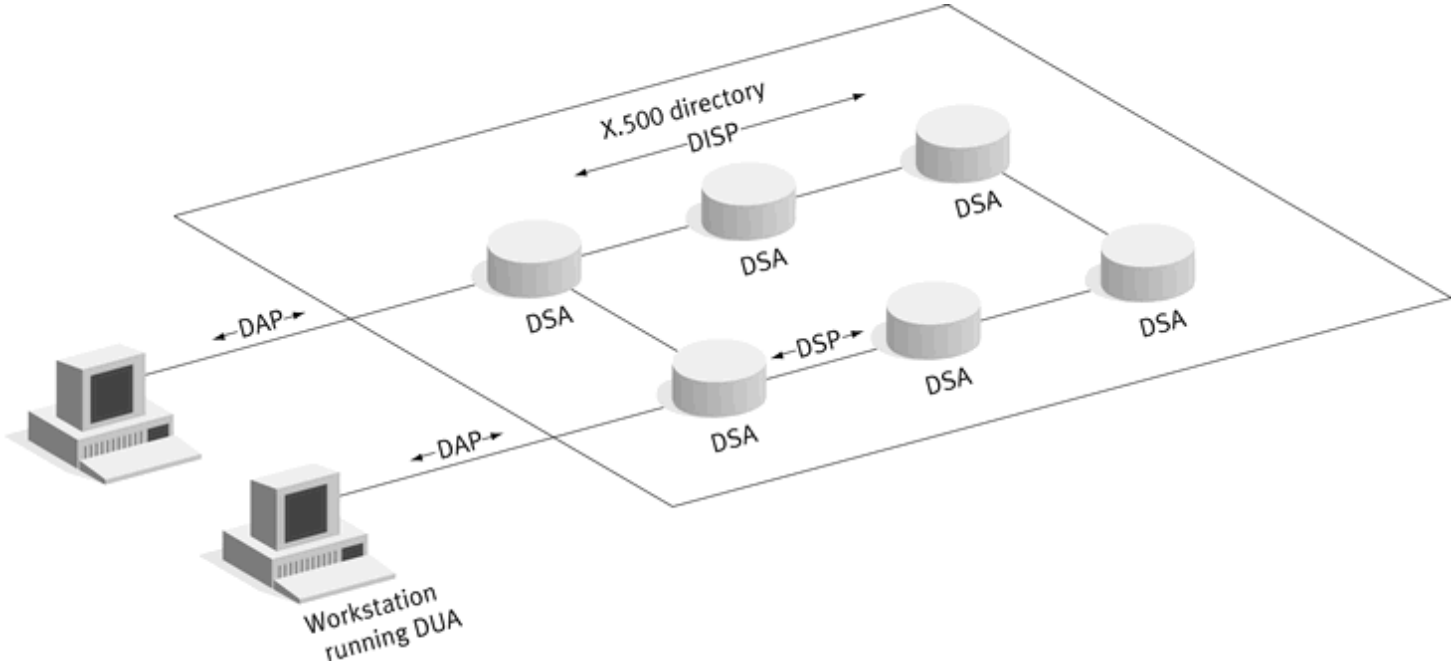
Spécifie comment l'information doit être stockée et consultée dans un service global d'annuaire.
Ne définit pas le fonctionnement interne du serveur.

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

Annuaires 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

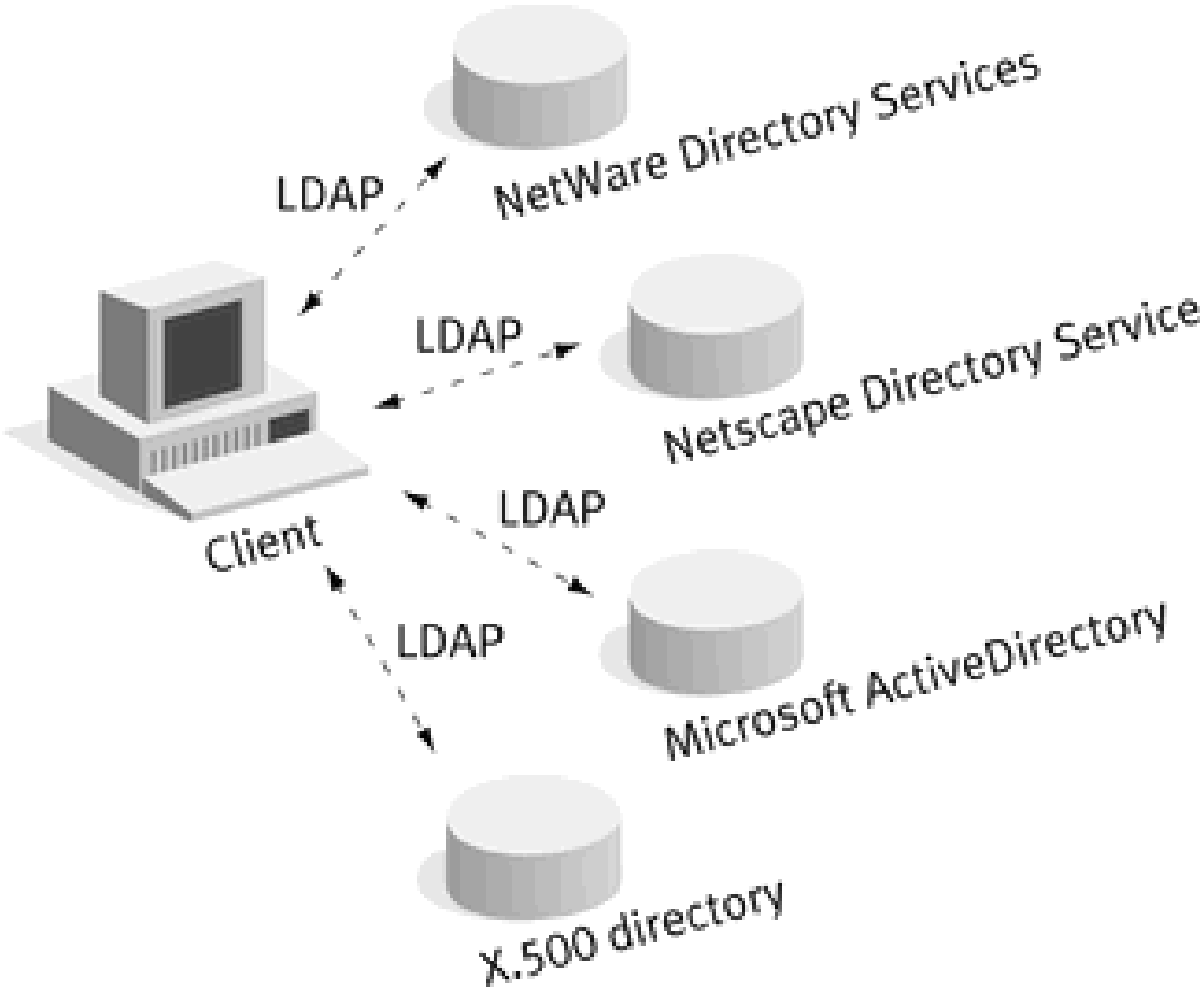
Inconvénients de X500

- 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

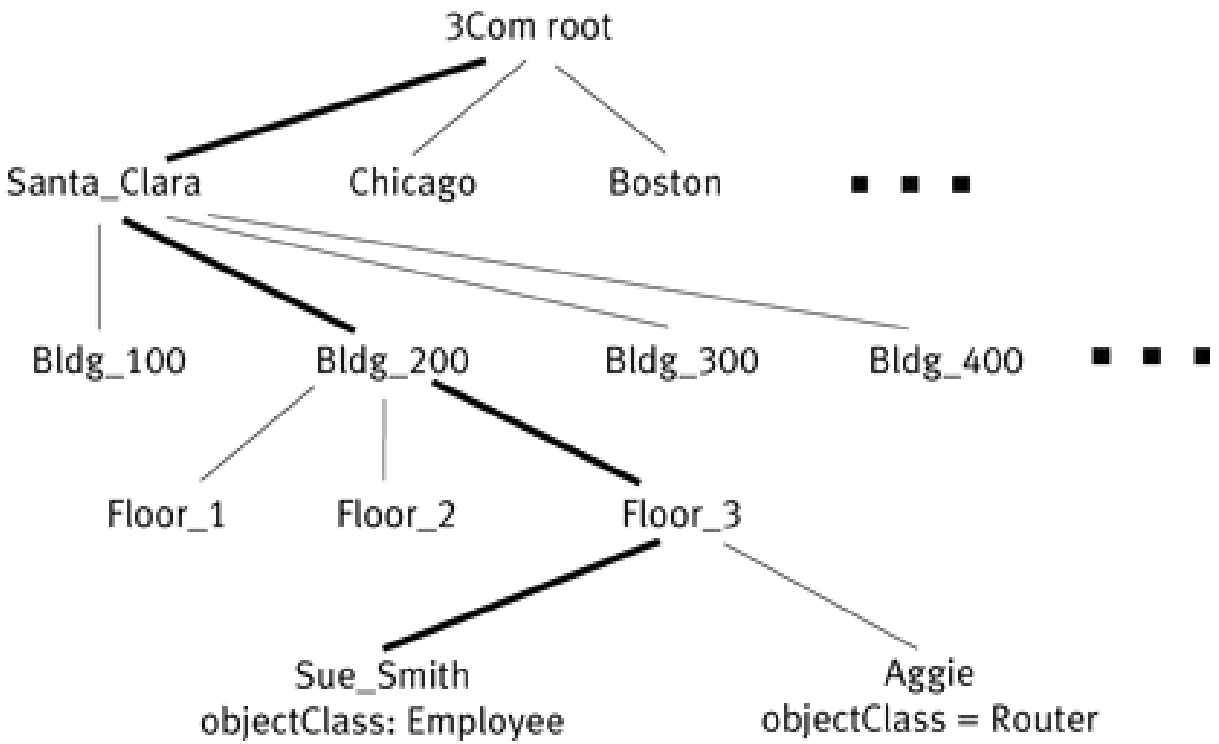


Stockage dans un annuaire LDAP ?

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 a son DN

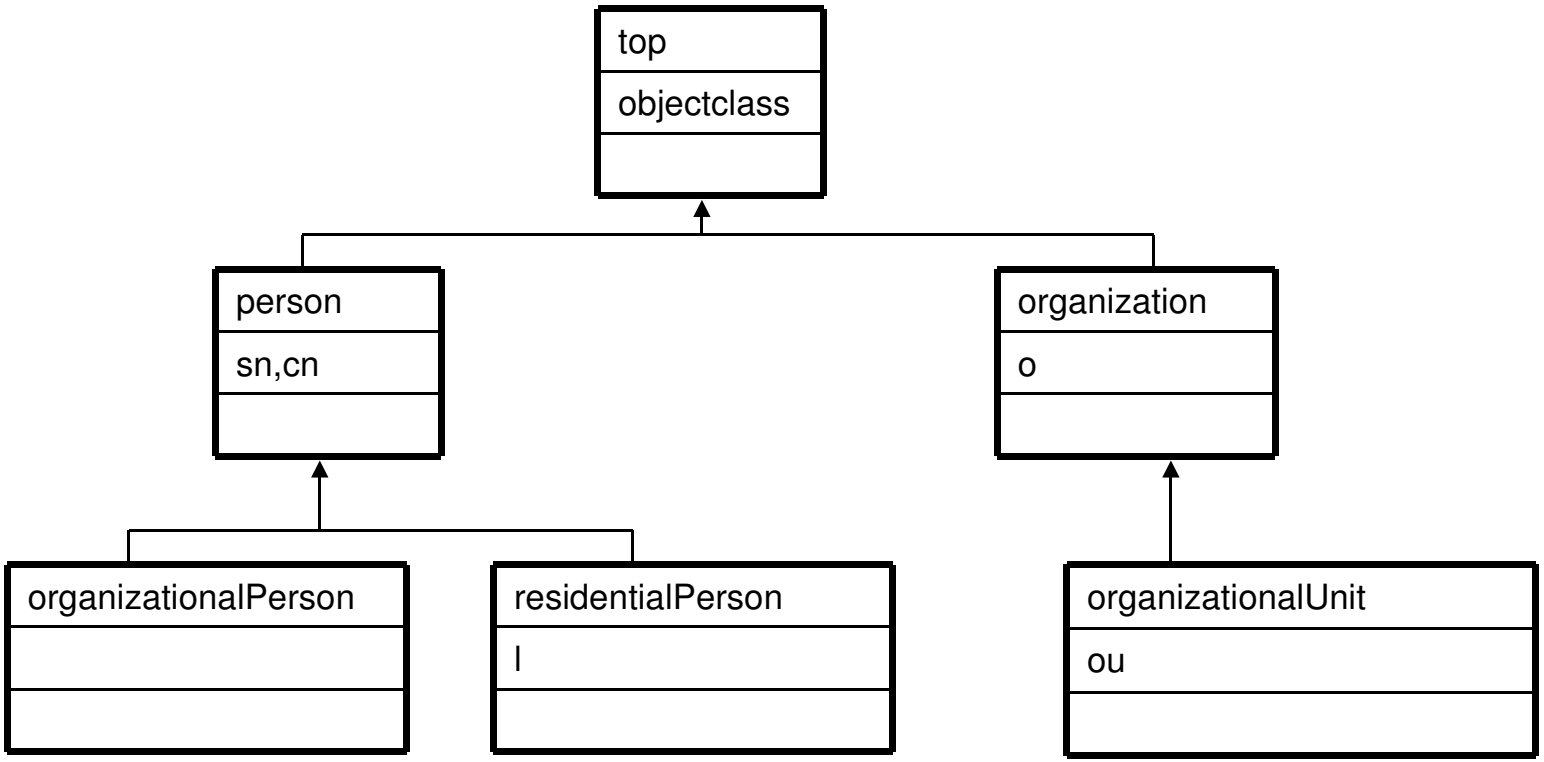
- construit en prenant le nom de l'élément RDN (Relative Distinguished Name) et en lui ajoutant les noms des entrées parentes dans la hiérarchie.
- on utilise donc une suite de paires attribut/valeur permettant de repérer l'entrée de manière unique.
- Exemple de DN : "uid=209,ou=rt,o=iut"

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 :	débute 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

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

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
- ...

Principe du NIS

Objectif : simplifier l'administration d'un ensemble de machines sur un réseau local

- Ne pas avoir à configurer machine par machine
- Gestion des comptes utilisateurs
- Souvent couplé à NFS

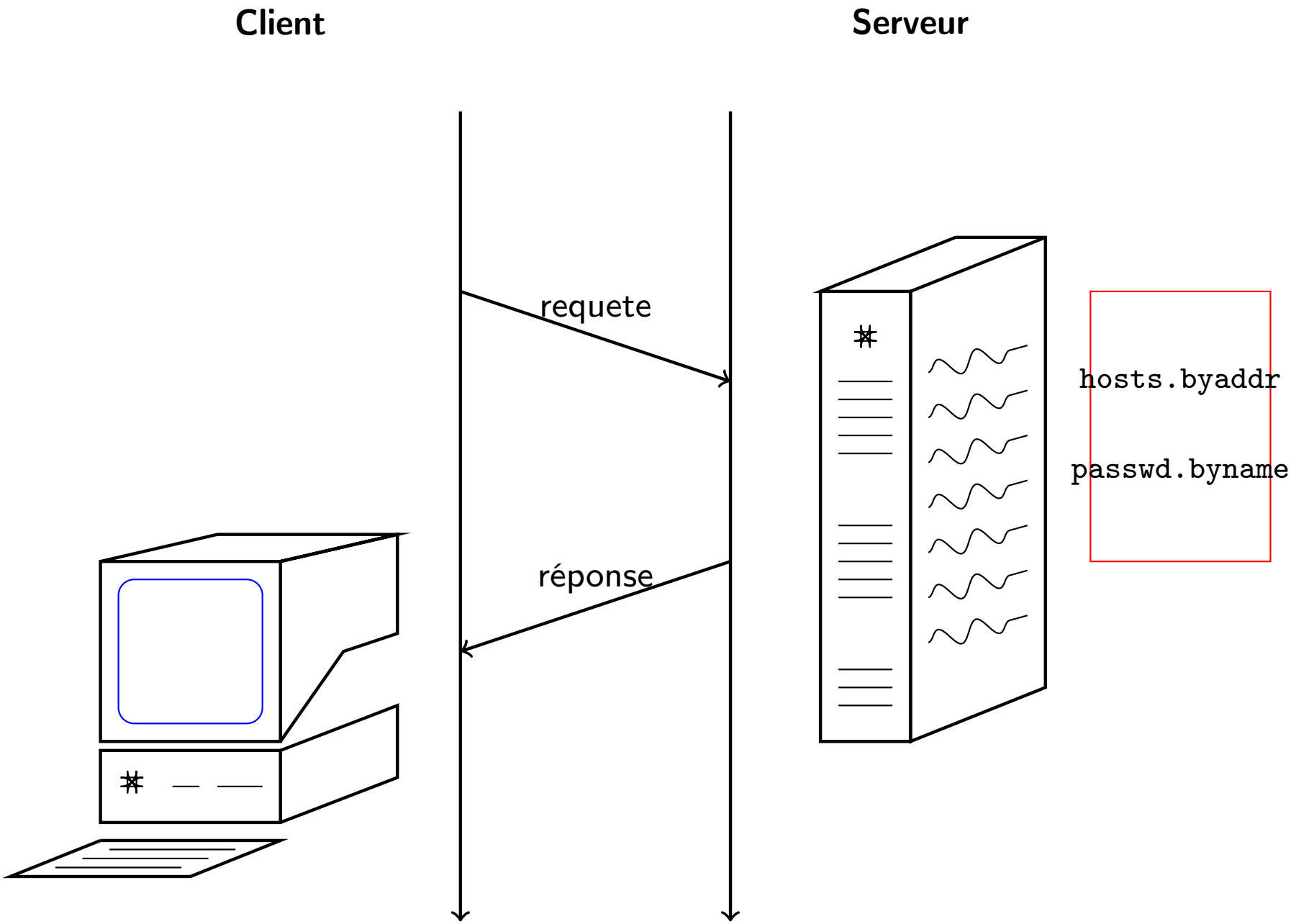
Un serveur par réseau local

- Notion de domaine NIS
- Centralisation des informations
- Plusieurs serveurs : maître-esclave ou coopération entre serveurs

À l'origine : pages jaunes (yp = yellow pages)

- Introduit par Sun en 1985
- Pas un standard, mais très largement utilisé

Exemple de fonctionnement



Architecture

Au moins un serveur NIS par réseau

- Le serveur tient à jour les informations
- Les clients viennent l'interroger

Si plusieurs serveurs :

- Le maître maintient les informations
- Le maître réplique les infos vers les serveurs secondaires
- Seul le maître peut modifier les infos contenues dans les bases
- Les esclaves diffusent les infos mais ne peuvent pas les modifier

Le maître maintient des maps

- Les maps stockent les infos sous forme de couples clé/valeur
- Les maps sont générées à partir des fichiers système : /etc/hosts, /etc/passwd...

Configuration de NIS

Au niveau du serveur

Les maps sont stockées dans le répertoire `/var/yp/domaine` (avec `domaine` le nom du domaine)

- Avec `make` on génère les fichiers contenant les maps

Les maps sont générées à partir des fichiers `/etc/hosts` et `/etc/passwd`

Au niveau du client

Fichier `/etc/nsswitch.conf`

- Ordre utilisé pour l'authentification :

```
hosts:          files nis dns
passwd:         files nis
group:          files nis
shadow:         files nis
```

Fichier `/etc/passwd` :

- Ajouter `+::::::` à la fin de la liste des utilisateurs locaux

Remarque : ancienne méthode, plus nécessaire avec les versions actuelles de NIS.

Outils NIS

Sur le serveur :

- `ypserv` : répond aux requêtes des clients
- `rpc.yppasswd` : répond aux requêtes de changement de mot de passe (`passwd`)

Si le serveur est un serveur maître :

- `ypxfrd` : répond aux requêtes de mise à jour de la map des esclaves

Si le serveur est lui-même un client NIS :

- `ypbind`

Client NIS

Le client doit être "lié" (*bindé*) à un serveur

- Fichier `/etc/yp.conf` pour le serveur, `/etc/defaultdomain` pour le positionnement du domaine

```
$ cat /etc/yp.conf
ypserver 10.10.0.10
$ cat /etc/defaultdomain
lipn.univ-paris13.fr
```

- Nom du serveur auquel on est bindé : commande `ypwhich`

```
$ ypwhich
nslipn.lipn.univ-paris13.fr
```

- Possibilité de le fixer avec la commande `ypset`

Maps NIS

Voir le contenu d'une map : ypcat

```
$ ypcat -x
Use "ethers" for map "ethers.byname"
Use "aliases" for map "mail.aliases"
Use "services" for map "services.byname"
Use "protocols" for map "protocols.bynumber"
Use "hosts" for map "hosts.byname"
Use "networks" for map "networks.byaddr"
Use "group" for map "group.byname"
Use "passwd" for map "passwd.byname"
```

```
$ ypcat passwd.byname | grep coti
coti:8fVPKGXHmOy9Y:3456:100:Camille Coti:/users/coti:/bin/bash
```

9 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

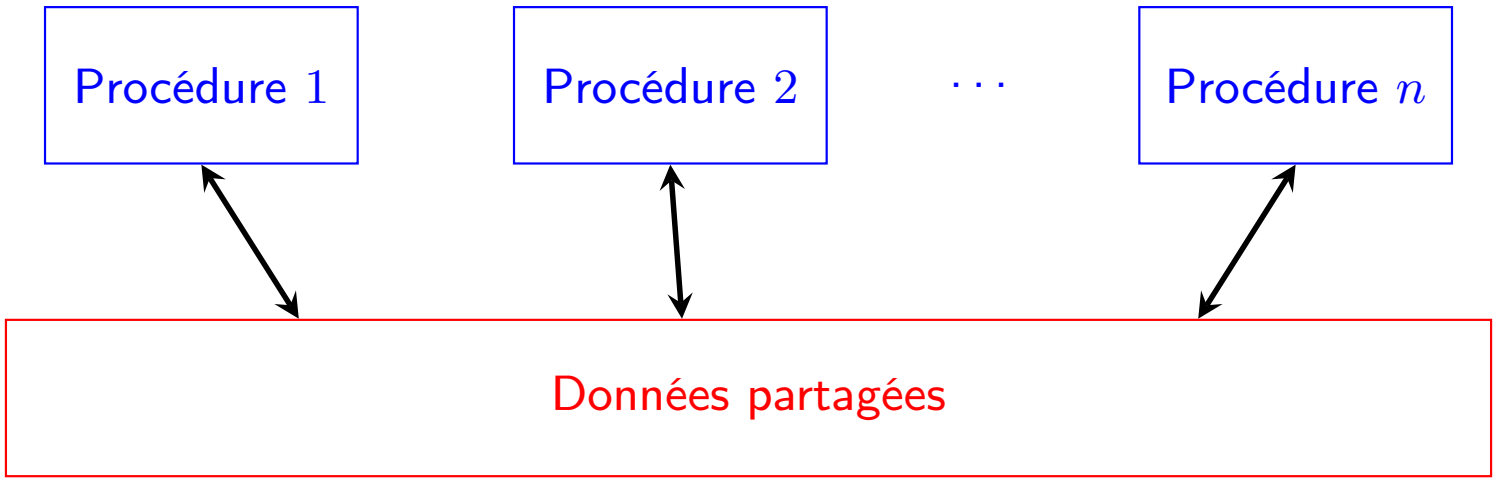
- programmation d'applications client/serveur
- étendre la notion d'appel de fonction de locale à distante
- éviter d'inclure la gestion du réseau

Services associés

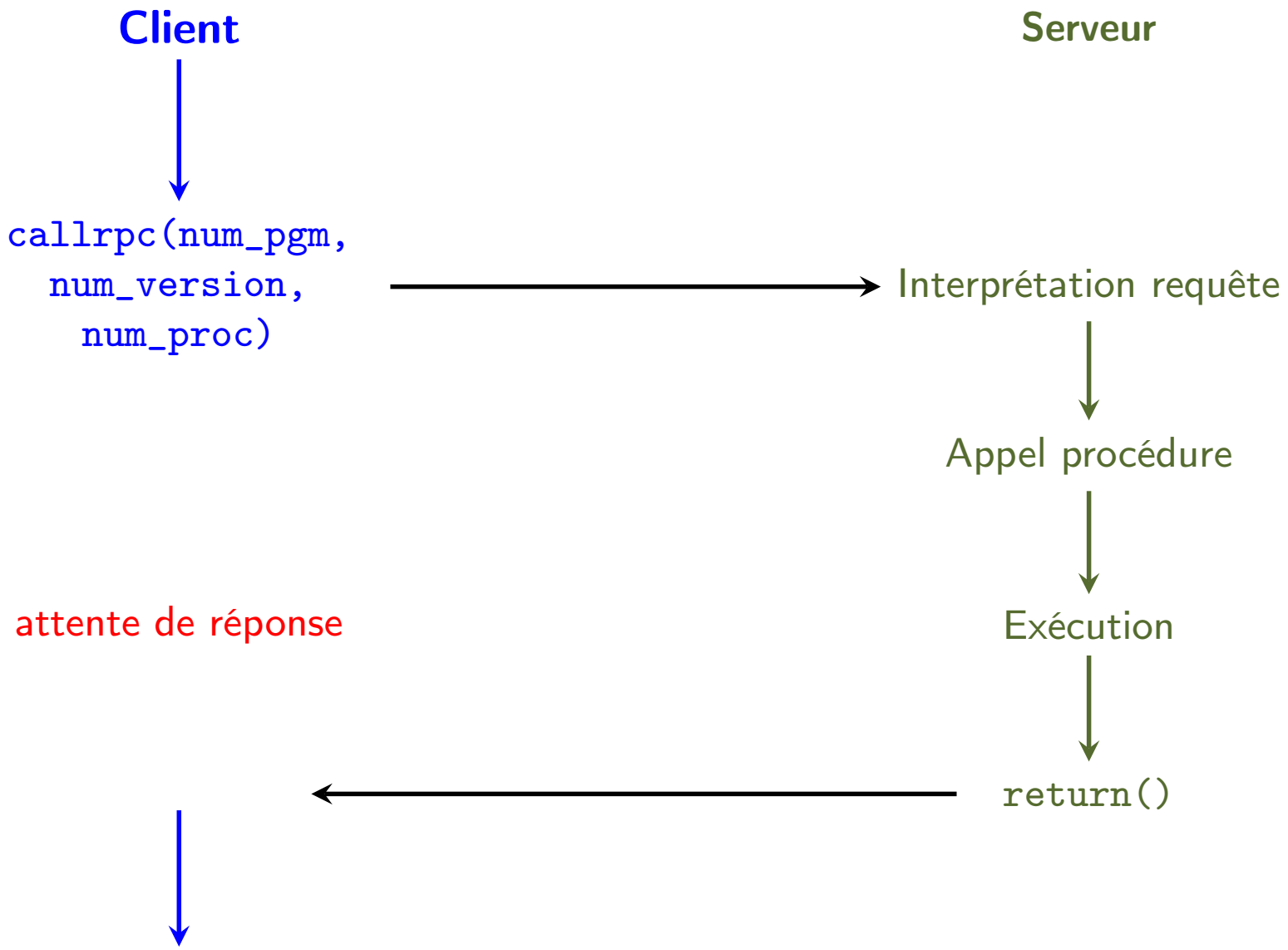
- Présentation : XDR (eXternal Data Representation)
- Transport : TCP ou UDP

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



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

Services RPC

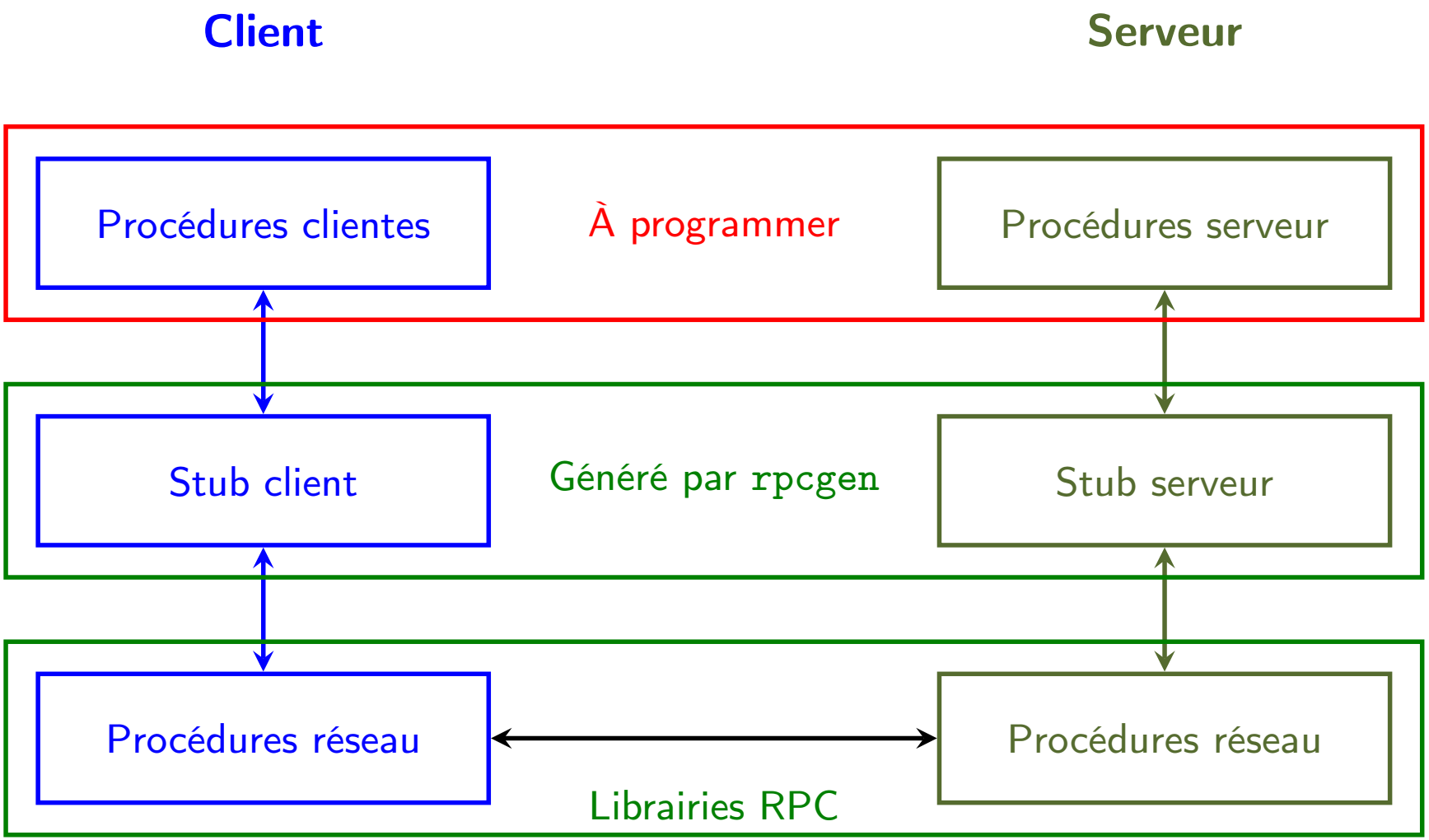
Pour obtenir la liste des services RPC sur une machine : commande `rpcinfo`

```
coti@maximum:~$ rpcinfo -p abidjan
   program no_version protocole  no_port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp   4000  status
    100024    1   tcp   4000  status
    100007    2   udp    755  ypbind
    100007    1   udp    755  ypbind
    100007    2   tcp    756  ypbind
    100007    1   tcp    756  ypbind
    391002    2   tcp    678  sgi_fam
```

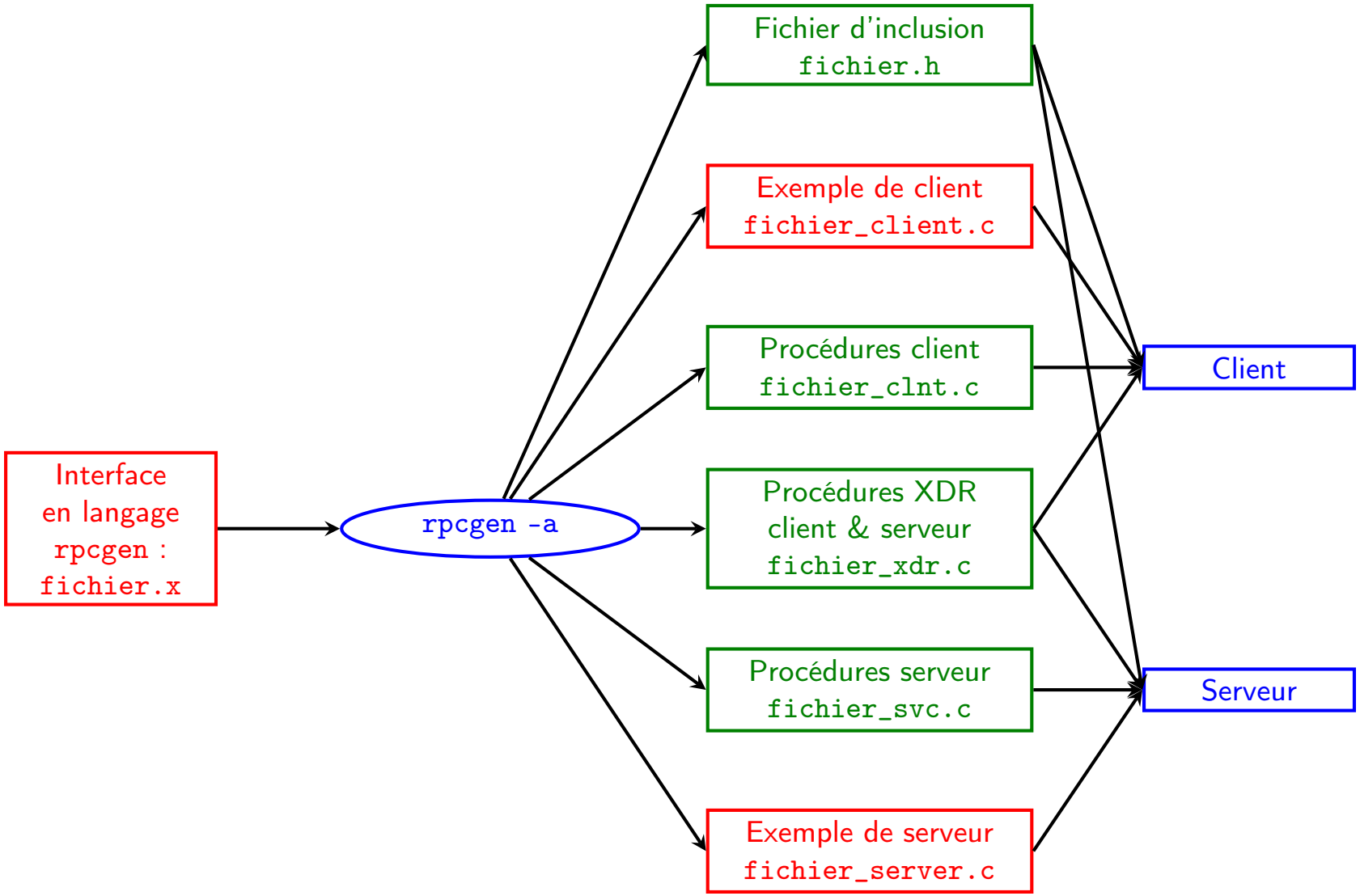
Infos sur un service en particulier : on utilise le numéro du programme

```
coti@maximum:~$ rpcinfo -u abidjan 100007
Le programme 100007 de version 1 est prêt et en attente.
Le programme 100007 de version 2 est prêt et en attente.
```

Fonctionnement



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 $\rightarrow$ codage $\rightarrow$ réseau
- flux de réception : réseau $\rightarrow$ décodage $\rightarrow$ données

Autres RPCs et applications

Autres RPCs

- XML-RPC
- CORBA
- SOAP

Applications

- NIS
- NFS

Monitoring

But : être alerté lorsqu'un problème survient

- Disque plein
- Charge d'un serveur trop élevée
- Température CPU qui augmente trop
- Afflux de paquets réseau erronés

Pour l'administrateur : permet d'être pro-actif

- Régler une cause de panne potentielle
- Migrer un service avant que la machine ne tombe en panne
- Doubler un serveur sous-dimensionné...

Surveillance de l'état d'une machine

Sondes matérielles et logicielles

- Fournies par le système d'exploitation : donnent une idée sur ce qui est en train de tourner
- Surveillance de l'état des ressources
- Pseudo-système de fichiers `/proc`
- Utilisation de capteurs de températures

Charge d'une machine

Charge CPU : commande `top`

- Donne la liste des processus exécutés sur la machine
- Pour chaque processus, quelques infos : pourcentage CPU utilisé, mémoire utilisée, temps écoulé depuis le début de l'exécution...
- Possibilité de trier selon un critère en particulier

```
top - 14:47:39 up 14 days,  2:35, 10 users,  load average: 0,96, 0,99, 0,92
Tasks: 244 total,   2 running, 242 sleeping,   0 stopped,   0 zombie
%Cpu(s):  2,3 us,  0,5 sy,  0,0 ni, 96,8 id,  0,5 wa,  0,0 hi,  0,0 si,  0,0 st
KiB Mem:   8153328 total,  7021852 used,  1131476 free,   196668 buffers
KiB Swap:  7811068 total,     2644 used,  7808424 free,  2781416 cached

  PID USER      PR  NI  VIRT  RES  SHR S  %CPU %MEM    TIME+  COMMAND
28710 coti      20   0  986m 148m  23m S   7,0  1,9   0:36.82 chrome
 3529 coti      20   0 1189m  68m  11m S   3,7  0,9  35:04.63 chrome
 2989 coti       9 -11  434m 8824 5584 S   2,3  0,1  22:57.37 pulseaudio
 3167 coti      20   0 1308m 352m  48m S   2,3  4,4 134:44.69 chrome
 1734 root      20   0  260m 153m  27m S   1,7  1,9 185:30.21 Xorg
27381 coti      20   0  993m 127m  28m S   1,3  1,6   1:26.38 chrome
29027 coti      20   0  966m  95m  22m S   1,0  1,2   0:04.14 chrome
22163 coti      20   0  231m  46m  14m S   0,7  0,6   0:14.43 emacs
27744 coti      20   0  945m 124m  20m S   0,7  1,6   0:01.98 chrome
 1788 avahi     20   0 35928 3700 1488 S   0,3  0,0  23:25.46 avahi-daemon
 4301 coti      20   0  386m  24m  12m S   0,3  0,3   0:55.60 gnome-terminal
```

État de la mémoire vive

Commande `free` : mémoire utilisée

- En ko par défaut, sinon octets, Mo, Go, compatible avec les humains : options `-b`, `-m`, `-g`, `-h`

```
coti@maximum:~$ free -h
             total       used       free     shared    buffers     cached
Mem:          7,8G       7,0G       827M          0B       192M       2,7G
-/+ buffers/cache:       4,1G       3,7G
Swap:         7,4G       2,6M       7,4G
```

Utilisation de la mémoire : commande `vmstat`

- Possibilité de mesurer au cours du temps

```
coti@maximum:~$ vmstat 2 5
procs -----------memory---------- ---swap-- -----io---- -system-- ----cpu----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa
 0  1   2644 802256 196804 2802480    0    0     0     1    4    2  1  0 99  0
 0  0   2644 789608 196804 2803268    0    0     0     0 6362 9583  5  2 91  2
 0  0   2644 790072 196804 2803332    0    0     0     0 3893 6205  3  1 94  2
 0  1   2644 774516 196804 2803684    0    0     0     0 8501 12415  6  2 87  5
 1  0   2644 770200 196804 2803800    0    0     0     0 3082 5239  3  1 85 11
```

Communications

Communications inter-processus : IPC V5, listées avec la commande `ipcs`

```
coti@maximum:~$ ipcs
------ Segment de mémoire partagée --------
clé        shmid      propriétaire perms      octets     nattch     états
0x00000000 0          coti       600        393216     2          dest
0x00000000 32769      coti       600        393216     2          dest
0x00000000 342130690  coti       777        2050328    2          dest
0x00000000 273055747  coti       777        561600     2          dest
[...]
------ Tableaux de sémaphores --------
clé        semid      propriétaire perms      nsems
------ Queues de messages --------
clé        msqid      propriétaire perms      octets utilisés messages
```

Fichiers ouverts

Fichiers ouverts : `lsof`

- Sous Unix tout est fichier : sockets IP avec l'option `-i`

```
coti@maximum:~$ lsof -i
COMMAND     PID USER   FD   TYPE  DEVICE SIZE/OFF NODE NAME
icedove-b  3160 coti   56u  IPv4 1816971      0t0  TCP maximum:45516->wg-in-f109.1e100.net:imaps (ESTABLIS
icedove-b  3160 coti   58u  IPv4   18438      0t0  TCP maximum:55337->mail.iutv.univ-paris13.fr:imaps (EST
icedove-b  3160 coti   59u  IPv4 2022424      0t0  TCP maximum:56597->mail:imaps (ESTABLISHED)
icedove-b  3160 coti   69u  IPv4 2022443      0t0  TCP maximum:32872->mail.iutv.univ-paris13.fr:imaps (EST
icedove-b  3160 coti   76u  IPv4 2496423      0t0  TCP maximum:32953->par08s09-in-f19.1e100.net:http (ESTA
icedove-b  3160 coti   79u  IPv4 1162259      0t0  TCP maximum:59881->mail:imaps (ESTABLISHED)
```

- Passage du chemin vers un répertoire : fichiers se trouvant dans ce répertoire et ses sous-répertoires

```
coti@maximum:~$ lsof /

COMMAND     PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
bash       1746 coti  rtd    DIR    8,1     4096       2 /
bash       1746 coti  txt    REG    8,1   975488 5373975 /bin/bash
bash       1746 coti  mem    REG    8,1    47616 4769217 /lib/x86_64-linux-gnu/libnss_files-2.13.so
bash       1746 coti  mem    REG    8,1    43552 4769213 /lib/x86_64-linux-gnu/libnss_nis-2.13.so
bash       1746 coti  mem    REG    8,1    89056 4769211 /lib/x86_64-linux-gnu/libnsl-2.13.so
```

Communications réseau

Sockets ouvertes : commande netstat

- Possibilité de préciser le protocole : `-u` pour UDP, `-t` pour TCP
- Donne le PID et le nom du programme utilisant la socket
- Port local, adresse et port distants
- État de la socket

```
coti@maximum:~$ netstat -lapute
Connexions Internet actives (serveurs et établies)
Proto Recv-Q Send-Q Adresse locale          Adresse distante        Etat        User       Inode      PI
tcp        0      0 *:sunrpc                *:*                     LISTEN      root       3906       -
tcp        0      0 *:ssh                   *:*                     LISTEN      root       7730       -
tcp        0      0 localhost:ipp           *:*                     LISTEN      root       1861089    -
tcp        0      0 localhost:smtp          *:*                     LISTEN      root       6142       -
tcp        0      0 *:841                   *:*                     LISTEN      root       10374      -
tcp        0      0 *:33388                 *:*                     LISTEN      root       10711      -
tcp        0      0 *:42989                 *:*                     LISTEN      statd      3925       -
tcp        0      0 maximum:41813           wg-in-f125.1e100.n:5223 ESTABLISHED coti       1874088    27
tcp        0      0 maximum:54638           magi.univ-paris13.:2822 ESTABLISHED coti       2025047    31
tcp        0      0 maximum:36752           we-in-f109.1e100.:imaps ESTABLISHED coti       1872420    31
```

État des cartes réseau

État des cartes réseau : commande `ifconfig`

- Utilisé par le super-utilisateur pour configurer la carte réseau
- En lecture seule pour les autres utilisateurs
- Donne notamment le nombre de paquets erronés reçus

```
coti@maximum:~$ /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr d4:be:d9:9c:7a:9c
          inet adr:10.10.0.217  Bcast:10.10.255.255  Masque:255.255.0.0
          adr inet6: fe80::d6be:d9ff:fe9c:7a9c/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:42594748 errors:0 dropped:0 overruns:0 frame:0
          TX packets:28853636 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:18842740245 (17.5 GiB)  TX bytes:26078786297 (24.2 GiB)
          Interruption:20 Mémoire:e4c00000-e4c20000
```

Utilisation des disques

Espace restant sur les disques : `df` (disk free)

```
coti@maximum:~$ df
Sys. fich.                    1K-blocks      Util. Disponible Uti% Monté sur
/dev/sda1                      96120588   12105232   79132620  14% /
tmpfs                           1630668      92104    1538564   6% /tmp
/dev/sdb5                     384499764     203560  364764684   1% /home
lipn-sfa:/export4/vol04/coti 1922471424 1629569536  195246080  90% /users/coti
```

Espace utilisé par un fichier : `du` (disk used)

```
coti@maximum:~$ sudo du -sh /
```

Attention : méfiance si `du` et `df` ne donnent pas un résultat cohérent...

Statistiques d'entrée-sorties sur les disques : `iostat`

```
coti@maximum:~$ iostat
Linux 3.2.0-3-amd64 (maximum)  27/09/2012  _x86_64_ (8 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           0,71    0,00    0,14    0,22    0,00   98,93

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               0,33         1,19         5,45    1463193    6703576
sdb               0,00         0,01         0,00       6456        120
```

Sondes de température

Les constructeurs placent souvent des sondes de températures à des endroits critiques : CPU, disques, GPU...

- Sous Linux : utilitaire `lm_sensors`

```
coti@maximum:~$ sensors
coretemp-isa-0000
Adapter: ISA adapter
Physical id 0:  +34.0 C  (high = +80.0 C, crit = +98.0 C)
Core 0:         +30.0 C  (high = +80.0 C, crit = +98.0 C)
Core 1:         +31.0 C  (high = +80.0 C, crit = +98.0 C)
Core 2:         +32.0 C  (high = +80.0 C, crit = +98.0 C)
Core 3:         +32.0 C  (high = +80.0 C, crit = +98.0 C)
```

Pseudo-système de fichiers /proc

`/proc` est un pseudo-système de fichiers : on y accède (en lecture seule) comme à un système de fichiers monté comme un système de fichiers, mais ça n'en est pas un.

- Utilisé par le noyau pour représenter des informations sur les processus qui tournent
- `/proc/<pid>` : répertoire relatif au processus de pid `<pid>`

Contenu :

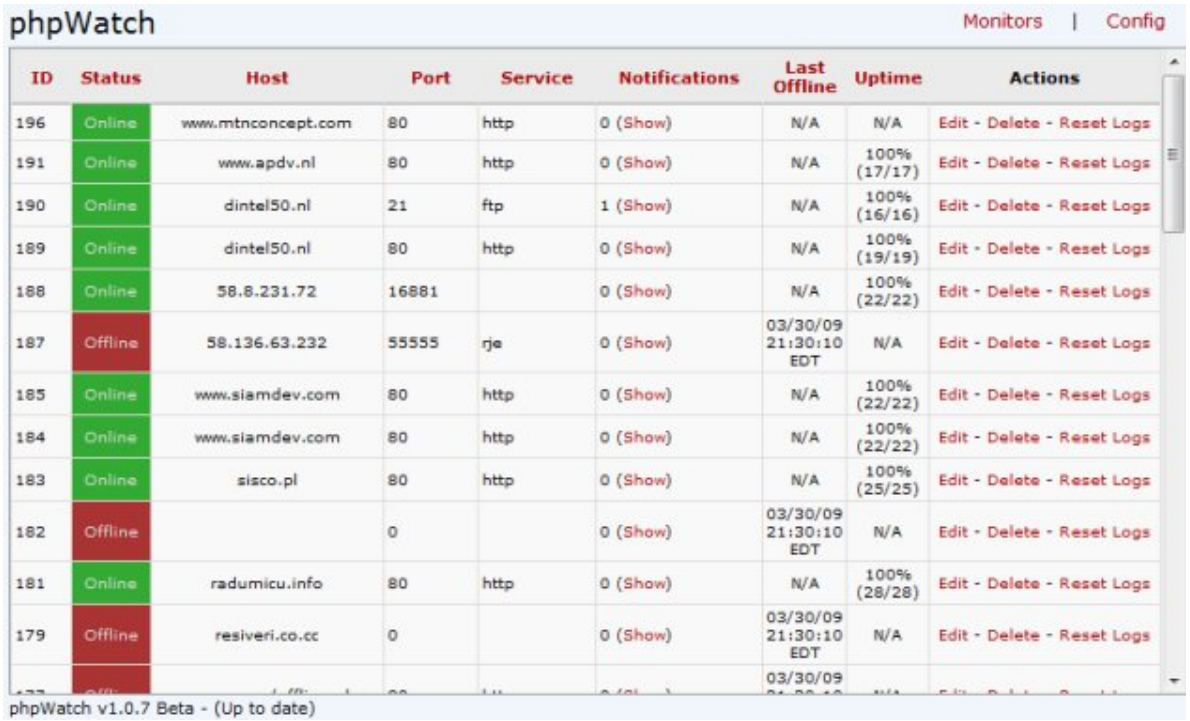
```
coti@maximum:~$ sudo ls /proc/772
[sudo] password for coti:
attr      coredump_filter  io mountstats     pagemap     stat
autogroup   cpuset       limits net         personality  statm
auxv      cwd       loginuid ns         root      status
cgroup      environ       maps numa_maps       sched     syscall
clear_refs  exe       mem oom_adj         sessionid    task
cmdline     fd       mountinfo oom_score       smaps     wchan
comm      fdinfo       mounts oom_score_adj  stack
```

- `cmdline` : ligne de commande ayant lancé le processus
- `statm` : statistiques sur la mémoire (taille, pages partagées...)
- `fd` : sous-répertoire contenant des liens vers tous les fichiers ouverts par le programme

Surveillance d'un ensemble de serveurs

Principe : avoir une vue d'ensemble sur les serveurs et centraliser les alertes

- L'administrateur ne va pas regarder sur chaque serveur un par un
- Plusieurs aspects :
  - Centralisation des messages journalisés (`tail /var/log/messages`)
  - État de la machine : température, disques...
  - Fonctionnement des applications : simulation de clients mail, web...
- Afficher ce qui va bien, alerter en cas de problème
  - Affichage en rouge
  - Envoi de mail, SMS...

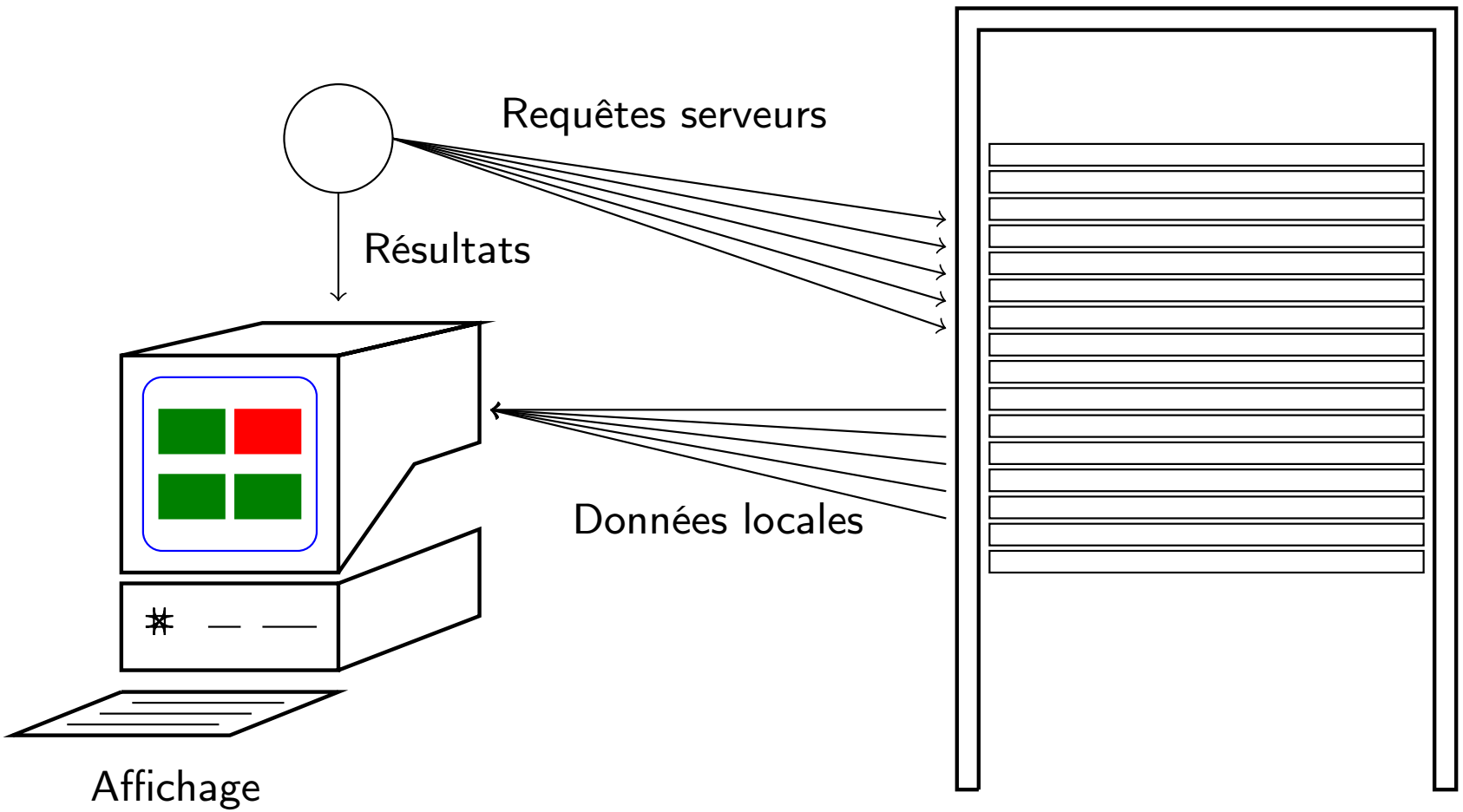


ID	Status	Host	Port	Service	Notifications	Last Offline	Uptime	Actions
196	Online	www.mtnconcept.com	80	http	0 (Show)	N/A	N/A	Edit - Delete - Reset Logs
191	Online	www.apdv.nl	80	http	0 (Show)	N/A	100% (17/17)	Edit - Delete - Reset Logs
190	Online	dintel50.nl	21	ftp	1 (Show)	N/A	100% (16/16)	Edit - Delete - Reset Logs
189	Online	dintel50.nl	80	http	0 (Show)	N/A	100% (19/19)	Edit - Delete - Reset Logs
188	Online	58.8.231.72	16881		0 (Show)	N/A	100% (22/22)	Edit - Delete - Reset Logs
187	Offline	58.136.63.232	55555	rje	0 (Show)	03/30/09 21:30:10 EDT	N/A	Edit - Delete - Reset Logs
185	Online	www.siamdev.com	80	http	0 (Show)	N/A	100% (22/22)	Edit - Delete - Reset Logs
184	Online	www.siamdev.com	80	http	0 (Show)	N/A	100% (22/22)	Edit - Delete - Reset Logs
183	Online	sisco.pl	80	http	0 (Show)	N/A	100% (25/25)	Edit - Delete - Reset Logs
182	Offline		0		0 (Show)	03/30/09 21:30:10 EDT	N/A	Edit - Delete - Reset Logs
181	Online	radumicu.info	80	http	0 (Show)	N/A	100% (28/28)	Edit - Delete - Reset Logs
179	Offline	resiveri.co.cc	0		0 (Show)	03/30/09 21:30:10 EDT	N/A	Edit - Delete - Reset Logs
						03/30/09		

Architecture d'un système de surveillance

Au moins deux composants :

- Un composant collecte les informations locales sur chaque serveur
  - Un démon sur chaque serveur : collecte les informations locales
  - Ou un pseudo-client qui interroge les serveurs et teste l'état des services
- Un composant rassemble et traite ces résultats
  - Affichage des données, statistiques
  - Le cas échéant, envoi des alertes



Ganglia

L'ancêtre : Ganglia

Un démon sur chaque serveur : `gmond`

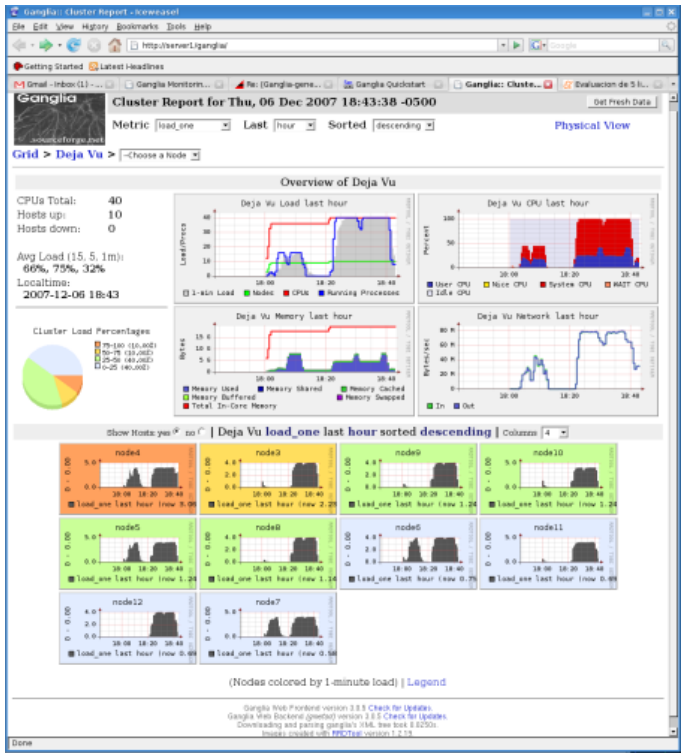
- Collecte des informations sur l'état du serveur : charge CPU, RAM occupée / libre, trafic réseau...
- Possibilité de définir de nouvelles métriques
- Les transfère au serveur Ganglia sur demande

Récupération des données : `gmetad`

- Interroge à intervalles régulier les `gmond` qui tournent sur les machines serveur
- Agrège et stocke les résultats des mesures

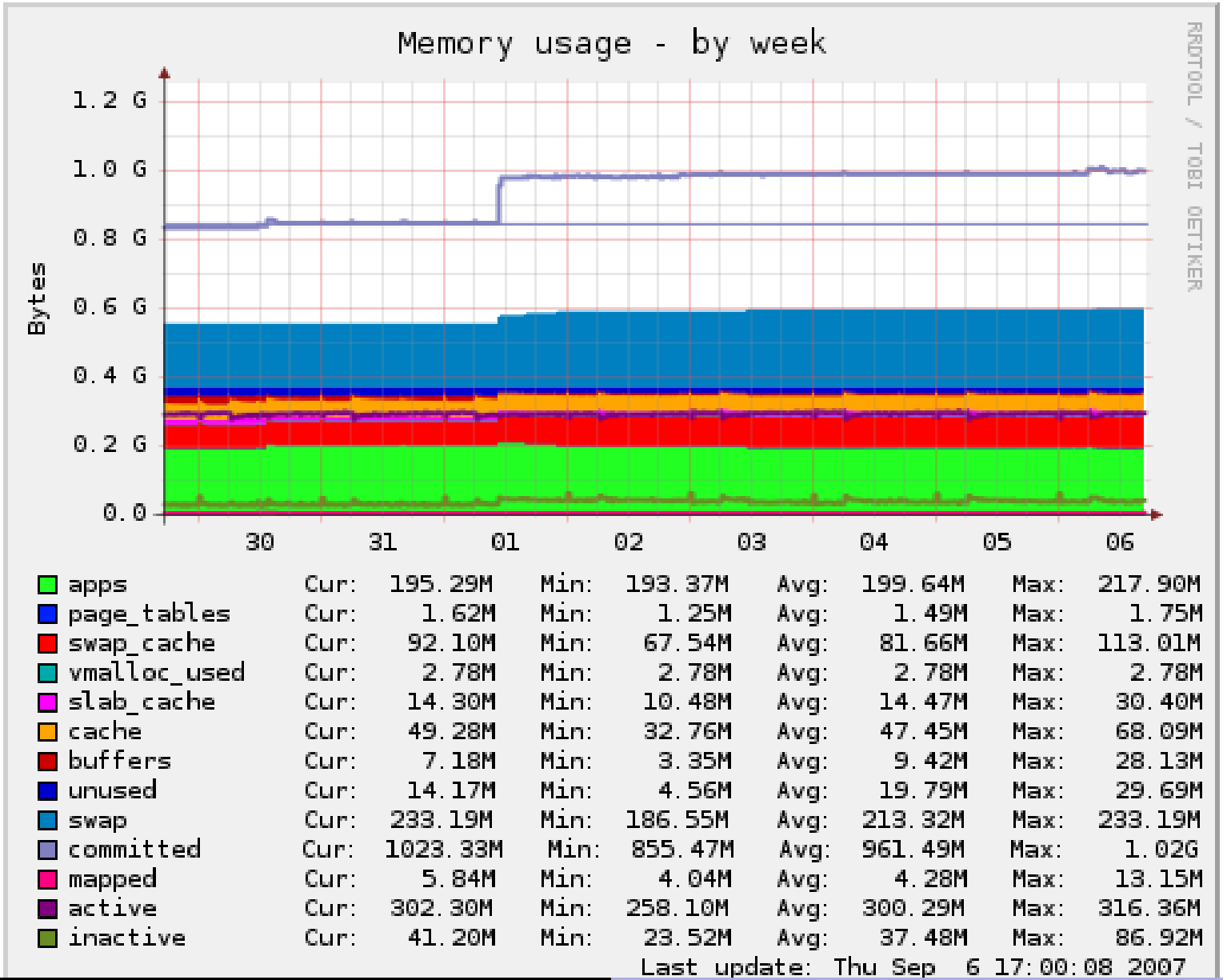
Affichage des résultats : `ganglia-web`

- Génère et affiche les graphiques
  - Hôte par hôte
  - Somme pour tous les hôtes



Munin

Outil de surveillance de serveurs mettant l'accent sur la simplicité



Munin

Architecture de Munin :

- Un démon sur chaque hôte prend des mesures
- Un programme sur le noeud maître rassemble et affiche les données
- À intervalles réguliers le maître interroge les hôtes et récupère leurs données, générer les graphes et envoyer d'éventuelles alertes

La surveillance hôte par hôte se fait *localement* : c'est le démon local qui prend les mesures sur sa machine

Extension des métriques mesurées avec des plugins

- Possibilité de mesurer ce que l'on veut
- Et de l'afficher comme on veut
- Nombreux plugins disponibles

Nagios

Outil de surveillance *modulaire* composé de :

- Un ordonnanceur, qui coordonne les tâches de supervision
- Une interface Web, qui affiche l'état du système supervisé
- Des plugins, qui effectuent chaque tâche de supervision, spécifiques à une application ou un contrôle

Chaque plugin retourne une valeur :

- 0 : OK
- 1 : warning
- 2 : critique
- 3 : erreur non définie

Par exemple : si la température monte au-dessus de 80C c'est un warning, au-dessus de 95C c'est critique.

Nagios

Deux catégories de contrôles possibles :

- Contrôle du fonctionnement des services : envoi d'une requête ad hoc
  - Exemple : le plugin envoie une requête HTTP sur le port d'un serveur Web
  - S'effectuent à distance : pas besoin d'intervenir sur le serveur
- Contrôle de données locales d'une machine
  - Nécessitent l'installation d'un démon sur l'hôte
  - Le plugin interroge ce démon

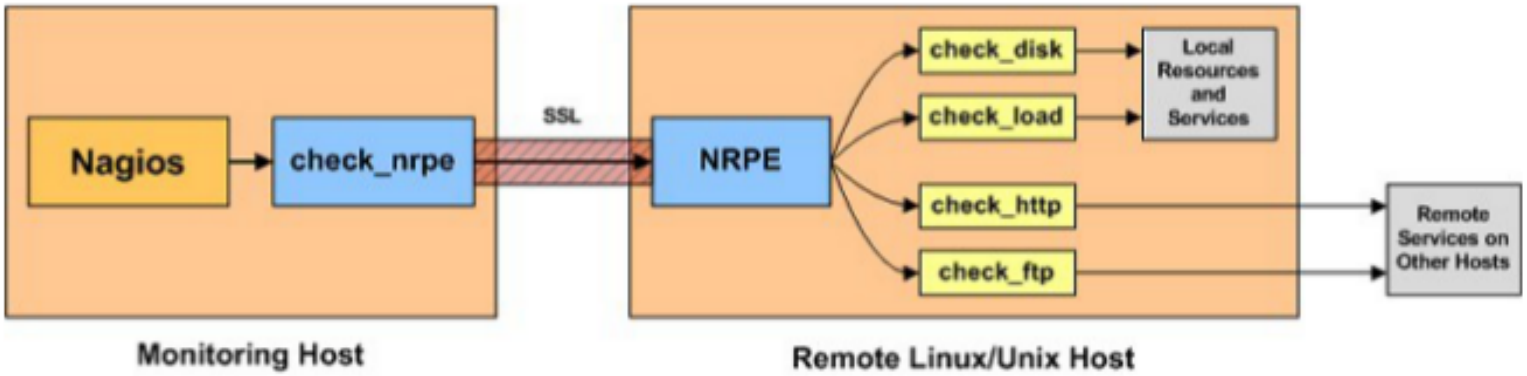
L'agent de contrôle peut-être délocalisé et envoyer les résultats à Nagios

- Exemple : si les serveurs sont dans une DMZ
- L'agent est mis dans la DMZ
- On n'a que l'agent à laisser sortir de le DMZ vers l'intérieur du réseau

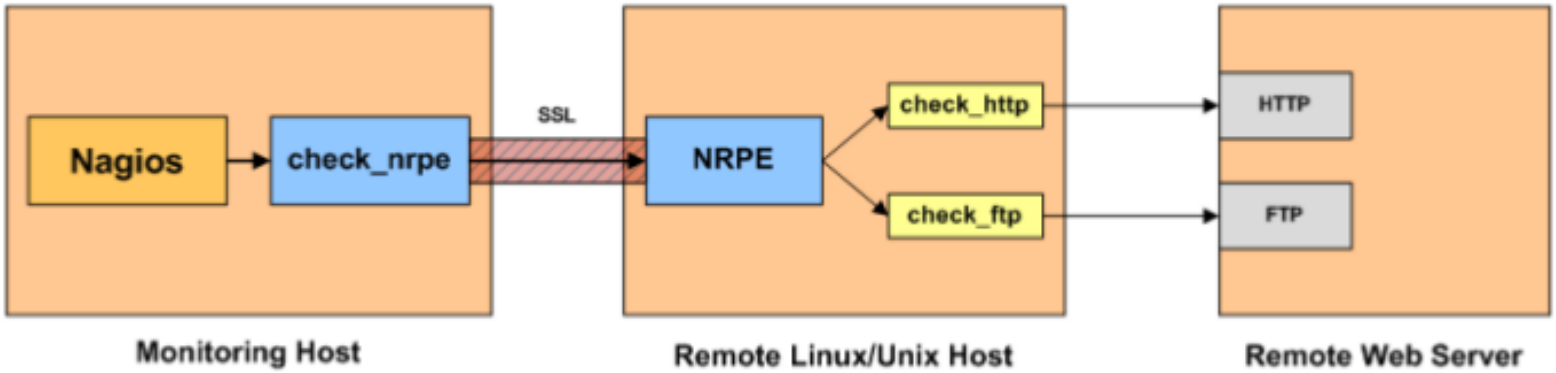
Nagios

Exemples d'architecture :

- Avec contrôles directement sur la machine



- Contrôles distants avec agent intermédiaire



Nagios

Hôtes	Services	Statut	Durée	Dernier contrôle	Essais	Information sur le statut
METIER_apogee	memory	UNKNOWN	8M 2w 6d 6h 19s	28/09/2012 - 15:49:42	3	ERROR: Description Table hrStorageType : No response from remote host '192.168.20.2'.
	load	UNKNOWN	8M 1w 1d 4h 34m 50s	28/09/2012 - 15:49:15	3	UNKNOWN: No response from remote host '192.168.20.2'.
METIER_chimed	load	UNKNOWN	8M 1w 1d 4h 34m 50s	28/09/2012 - 15:46:16	3	UNKNOWN: No response from remote host [redacted].
METIER_apogee	/	UNKNOWN	8M 1w 1d 4h 34m 12s	28/09/2012 - 15:49:42	3	ERROR: hrStorageDescr Table : No response from remote host '192.168.20.2'.
METIER_chimed	/	UNKNOWN	8M 1w 23h 19m 23s	28/09/2012 - 15:45:45	3	ERROR: hrStorageDescr Table : No response from remote host [redacted]
	memory	UNKNOWN	8M 1w 23h 41s	28/09/2012 - 15:45:45	1	ERROR: Description Table hrStorageType : No response from remote host [redacted]
WEB_videos	load	UNKNOWN	4M 5d 8h 44m 16s	28/09/2012 - 15:49:41	1	UNKNOWN: No response from remote host [redacted]
	memory	UNKNOWN	4M 5d 8h 44m 9s	28/09/2012 - 15:49:40	1	ERROR: Description Table hrStorageType : No response from remote hos[redacted].
HEBERGES_acft	/	UNKNOWN	4M 5d 8h 43m 31s	28/09/2012 - 15:49:41	1	ERROR: hrStorageDescr Table : No response from remote host [redacted].
WEB_videos	/	UNKNOWN	4M 5d 8h 43m 31s	28/09/2012 - 15:49:42	1	ERROR: hrStorageDescr Table : No response from remote host [redacted].
HEBERGES_acft	memory	UNKNOWN	4M 5d 8h 42m 54s	28/09/2012 - 15:45:39	1	ERROR: Description Table hrStorageType : No response from remote host [redacted].
	load	UNKNOWN	4M 5d 8h 40m 21s	28/09/2012 - 15:49:15	1	UNKNOWN: No response from remote host [redacted]
SYSTEME_cam	load	UNKNOWN	3M 1d 58m 46s	28/09/2012 - 15:49:40	3	UNKNOWN: No response from remote host '192.168.30.33'.
	/	UNKNOWN	3M 1d 57m 9s	28/09/2012 - 15:49:15	3	ERROR: hrStorageDescr Table : No response from remote host '192.168.30.33'.
	memory	UNKNOWN	3M 1d 56m 49s	28/09/2012 - 15:49:38	3	ERROR: Description Table hrStorageType : No response from remote host '192.168.30.33'.
SYSTEME_data-recovery	/	UNKNOWN	2M 4w 12h 25m 14s	28/09/2012 - 15:49:15	3	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.160'.
HEBERGES_moodle	Serveur HTTP_etudiant	CRITICAL	1M 2w 6d 15h 50m 22s	28/09/2012 - 15:47:41	1	No route to host
MESSAGERIE_w-services	load	UNKNOWN	1M 2w 6d 15h 50m 12s	28/09/2012 - 15:45:50	1	UNKNOWN: No response from remote host '192.168.0.181'.
SYSTEME_SmartDisk	/srv	UNKNOWN	1M 2w 6d 15h 50m 3s	28/09/2012 - 15:46:18	3	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.220'.
WEB_webdav	memory	UNKNOWN	1M 2w 6d 15h 49m 54s	28/09/2012 - 15:46:11	3	ERROR: Description Table hrStorageType : No response from remote host '192.168.0.107'.
WEB_w-services	/	UNKNOWN	1M 2w 6d 15h 49m 24s	28/09/2012 - 15:49:41	1	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.181'.
WEB_webdav	/	UNKNOWN	1M 2w 6d 15h 49m 22s	28/09/2012 - 15:49:40	3	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.107'.
SYSTEME_SmartDisk	ping	CRITICAL	1M 2w 6d 15h 49m 2s	28/09/2012 - 15:48:44	3	GPING CRITICAL - 3 packets transmitted, 0 received, +3 errors, 100percent packet loss, time 2031ms
MESSAGERIE_w-services	memory	UNKNOWN	1M 2w 6d 15h 48m 31s	28/09/2012 - 15:49:38	1	ERROR: Description Table hrStorageType : No response from remote host '192.168.0.181'.
SYSTEME_SmartDisk	load	UNKNOWN	1M 2w 6d 15h 48m 11s	28/09/2012 - 15:45:41	3	UNKNOWN: No response from remote host '192.168.0.220'.
WEB_w-services	load	UNKNOWN	1M 2w 6d 15h 47m 43s	28/09/2012 - 15:48:55	1	UNKNOWN: No response from remote host '192.168.0.181'.
WEB_webdav	load	UNKNOWN	1M 2w 6d 15h 47m 42s	28/09/2012 - 15:49:15	3	UNKNOWN: No response from remote host '192.168.0.107'.
MESSAGERIE_w-services	/	UNKNOWN	1M 2w 6d 15h 46m 46s	28/09/2012 - 15:49:42	1	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.181'.
HEBERGES_moodle	Serveur HTTP_personnel	CRITICAL	1M 2w 6d 15h 46m 41s	28/09/2012 - 15:47:41	1	No route to host
SYSTEME_SmartDisk	memory	UNKNOWN	1M 2w 6d 15h 46m 41s	28/09/2012 - 15:47:47	1	ERROR: Description Table hrStorageType : No response from remote host '192.168.0.220'.
WEB_w-services	memory	UNKNOWN	1M 2w 6d 15h 46m 2s	28/09/2012 - 15:49:14	1	ERROR: Description Table hrStorageType : No response from remote host '192.168.0.181'.
SYSTEME_SmartDisk	/	UNKNOWN	1M 2w 6d 15h 44m 35s	28/09/2012 - 15:47:46	3	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.220'.
WEB_php-oc2	/	UNKNOWN	1M 2w 6d 10h 44m 11s	28/09/2012 - 15:49:40	1	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.131'.
	memory	UNKNOWN	1M 2w 6d 10h 43m 53s	28/09/2012 - 15:49:42	1	ERROR: Description Table hrStorageType : No response from remote host '192.168.0.131'.
	load	UNKNOWN	1M 2w 6d 10h 42m 34s	28/09/2012 - 15:49:37	1	UNKNOWN: No response from remote host '192.168.0.131'.
WEB_tomcat	load	UNKNOWN	1M 1d 17h 3m 7s	28/09/2012 - 15:49:15	1	UNKNOWN: No response from remote host '192.168.0.8'.
	/	UNKNOWN	1M 1d 17h 46s	28/09/2012 - 15:48:55	1	ERROR: hrStorageDescr Table : No response from remote host '192.168.0.8'.
	memory	UNKNOWN	1M 1d 17h 36s	28/09/2012 - 15:49:38	1	ERROR: Description Table hrStorageType : No response from remote host '192.168.0.8'.

Diagnostic de l'état d'un réseau

Comment diagnostiquer l'état d'un réseau ?

- `ping`, `traceroute` : vérifier la réactivité d'un équipement et d'un lien
- Sur un gros réseau, composé d'un fédération de réseaux
- Fastidieux et laborieux !

Motivation de SNMP : rassembler des informations sur les noeuds qui composent le réseau

- Simple Network Monitoring Protocol
- Hôtes hétérogènes
- RFC 1157 (mai 1990) : SNMP v1 pour le protocole, RFC 1155 pour l'information échangée, RFC 1212 pour la SMI (Structure of Management Information)
- SNMPv2 : RFC 1441 à 1452
- SNMPv3 (décembre 1998) : RFC 2570, 2572 à 2575 et 2578 à 2580 (SMI v2)

Le protocole SNMP

Modèle d'un réseau vu par SNMP :

- Noeuds administrés
- Stations d'administration
- Information d'administration
- Protocole d'administration

Principe d'architecture :

- Chaque noeud monitoré fait tourner un agent local
- L'agent local maintient une base d'informations : la MIB : Management Information Base)
- Les noeuds d'administration communiquent avec les agents locaux via le protocole SNMP pour récupérer ces infos

Format de la MIB

La MIB2 (définie dans le cadre de SNMPv2) regroupe les objets traités en 10 catégories :

Groupe	Nb	Description
System	7	Nom, emplacement et description
Interfaces	23	Interfaces réseau et trafic mesuré
AT	3	Translation d'adresses (désapprouvée)
IP	42	Stats sur les paquets IP
ICMP	26	Stats sur les paquets ICMP reçus
TCP	19	Algorithmes, paramètres et stats TCP
UDP	6	Stats de trafic UDP
EGP	20	Stats de trafic EGP
Transmission	0	Réservé
SNMP	29	Stats de trafic SNMP

Les évènements significatifs

Un équipement réseau peut subir des évènements particuliers :

- Reboot, plantage, rupture de liaison, congestion...

On appelle ces évènements significatifs des traps.

Un équipement qui subit un tel évènement l'annonce à toutes les stations

- Ensuite les stations d'administration viennent demander plus d'informations

Communications non-fiables (sur UDP) : les stations d'administration demandent généralement des infos sur des traps éventuels étant survenus : trap (direct) polling.

Langage de description de données

Description des données normalisé : ASN.1 (Abstract Syntax Notation 1), codage norme IS 8825

- But : optimiser le nombre de bits à transmettre
- Langage description de données : combinaison d'objets de base pour donner des objets plus complexes

Types de base :

Type primitif	Signification	Code
Integer	Entier de longueur arbitraire	02
Bit string	Chaîne de bits (éventuellement nulle)	03
Byte string	Chaîne d'octets non signés	04
NULL	Aucun type	05
Object identifier	Type de données officiellement défini	06

NB : ASN.1 défini des types Real et Boolean, qui ne sont pas autorisés dans SNMP ; SNMP ajoute quelques types spécifiques (compteurs...)

Syntaxe de transfert

BER (comme LDAP)

Pour chaque valeur transmise, on envoie au moins 4 champs :

1. L'identificateur (type ou étiquette)
2. La longueur du champ de données, en octets
3. Le champ de données
4. Le drapeau de fin de données si la longueur est inconnue (défini dans ASN.1 mais interdit dans SNMP)

Identificateur (1er octet) :

Étiquette (2)	Type (1)	Valeur de l'étiquette (5)

Les longueurs des champs sont données en bits

Étiquette :

- 00 Universel
- 01 Application
- 10 Spécification du contexte
- 11 Privé

Type :

- 0 Type primitif
- 1 Type construit

Exemples

Exemples de codages ASN.1 :

Valeur	*Identificateur*			*Longueur*	*Valeur*	
49 :	00	0	00010	0000 0001	0011 0001	
110 :	00	0	00011	0000 0010	0000 0101	0111 1001
NULL :	00	0	00101	0000 0000		

Protocole SNMP

Utilisation normale :

- La station d'administration demande à un agent de lui fournir une info ou de se mettre à jour
- L'agent lui répond

7 messages possibles :

- Get-request : demande la valeur d'une ou plusieurs variables
- Get-next-request : demande la variable suivante
- Get-bulk-request : Chargement d'une grande table
- Set-request : Mise à jour d'une ou plusieurs variables
- Inform-request : Message de description d'une MIB locale
- Snmpv2-trap : Indication de déroutement provenant d'un agent
- Response ou Get-response : réponse de l'agent aux messages Set et Get

SNMP et la sécurité

1er mécanisme de sécurité : la communauté : chaîne de caractères censée identifier la provenance du message

- Les objets du groupe SNMP décrivent le nb de requêtes reçues, mal formulées etc
- Début de contrôle sur la provenance des paquets SNMP

Faible !!

SNMPv2 : attention portée à la sécurité

- Authentification, encryption : alourdissent le protocole (on y renonce souvent)
- Notion de groupe (SnmpParty) et de contexte (local, distant)

Format d'une trame SNMPv1

Format d'une trame SNMPv1

- Pour une requête Get-request, Get-next-request ou Set-request :

Version	Communauté	Requête	0	0	Variables

- Pour une requête Get-response :

Version	Communauté	Requête	Code d'erreur	Index d'erreur	Variables

- Pour une requête Trap :

Version	Comm.	Requête	Type d'objet générant l'alarme	Adresse IP de l'agent	Code trap de l'entreprise	Time stamp	variables

- Chaque message est envoyé dans un seul datagramme UDP, port 161 (messages) et 162 (trap).
- Code d'erreur : noError (0), tooBig(1), noSuchName(4), genErr(5).
- Le champ Version est sur 4 octets, les autres champs sont de taille variable.

Format d'une trame SNMPv2

Format d'une trame SNMPv2

- Pour une requête Get-request, Inform-request, ... :

Version	Comm.	Requête	Numéro de requête	État d'erreur	Index d'erreur	Variables

- Pour une requête Get-bulk-request :

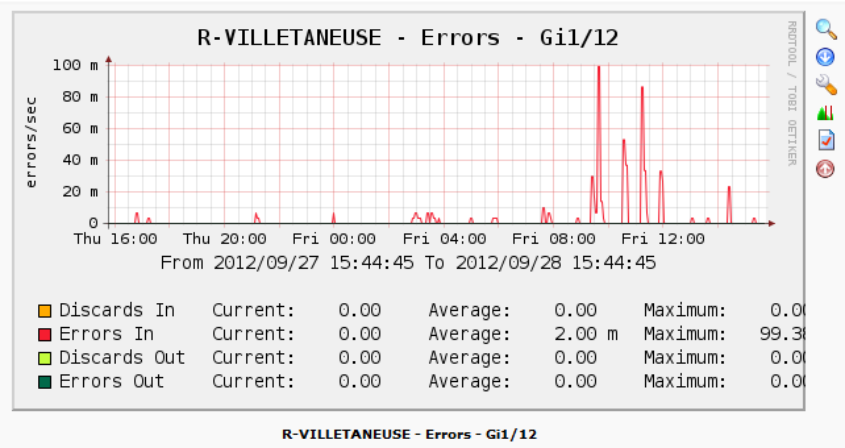
Version	Comm.	Requête	Numéro de requête	Nb de non-répétitions	Nb max de répétitions	Variables

Cacti

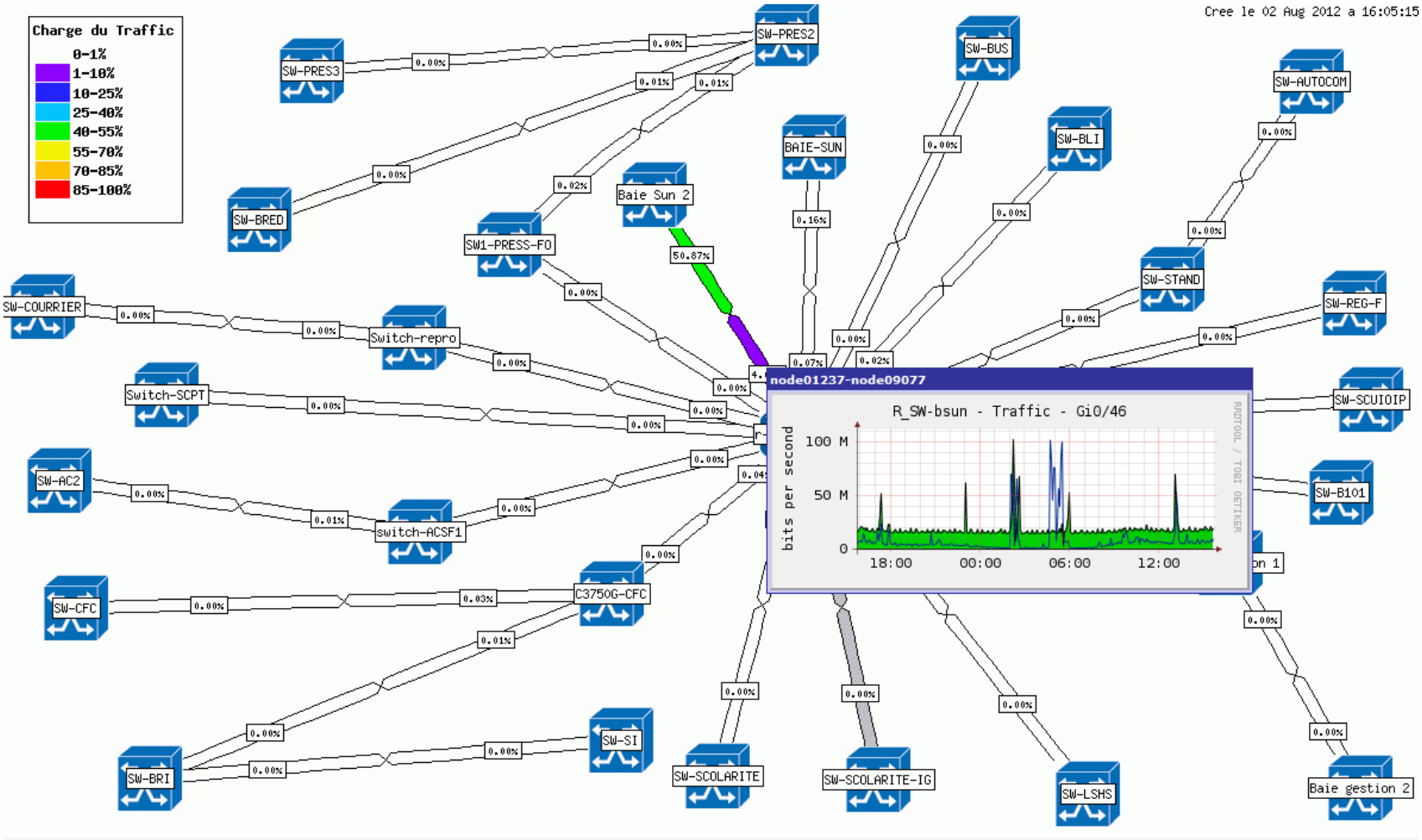
Surveillance réseau :

- Trafic réseau
- Erreurs sur les ports des commutateurs
- Paquets malformés
- Erreurs d'adresse MAC
- ...

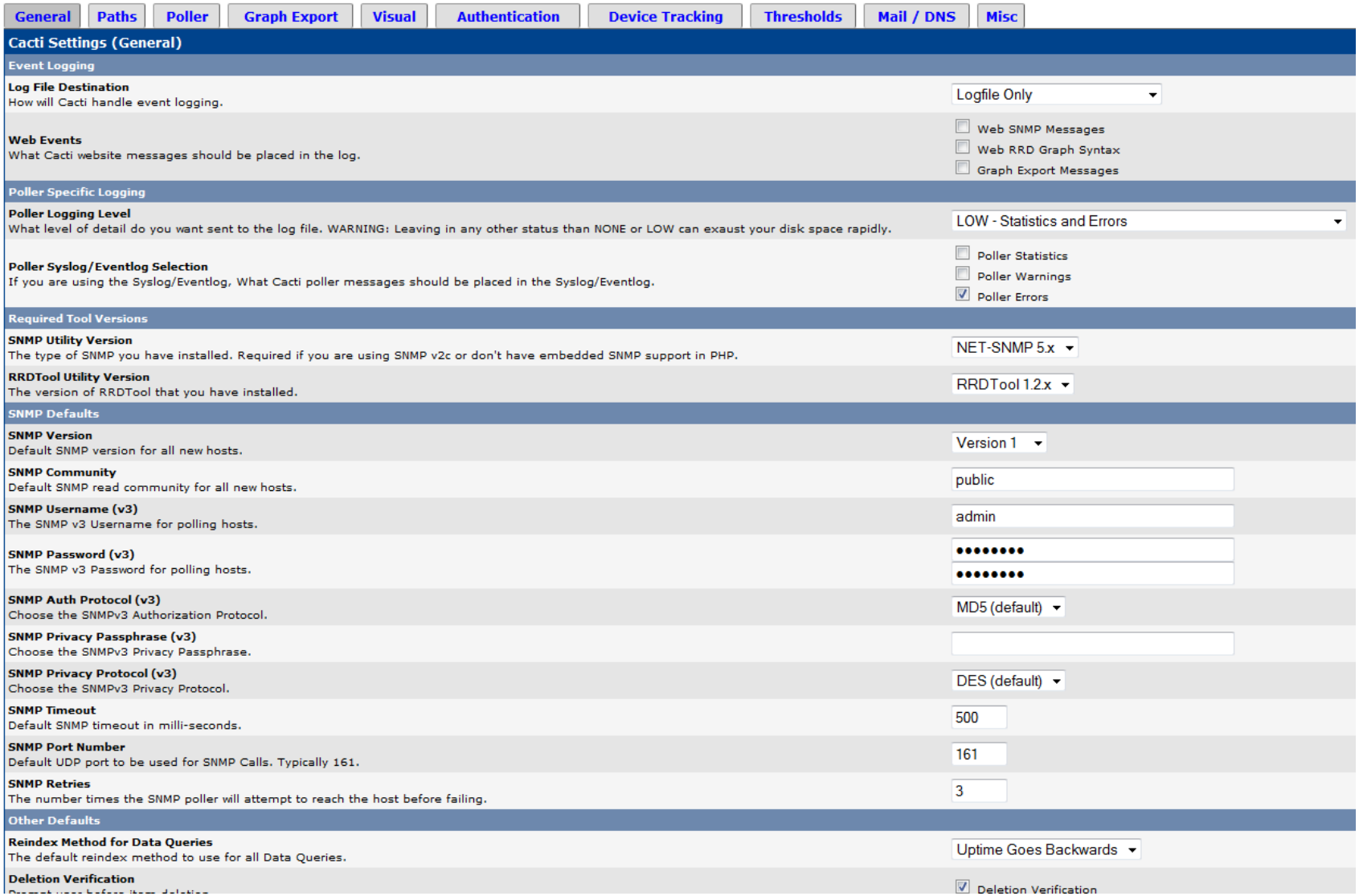
Utilisation de SNMP



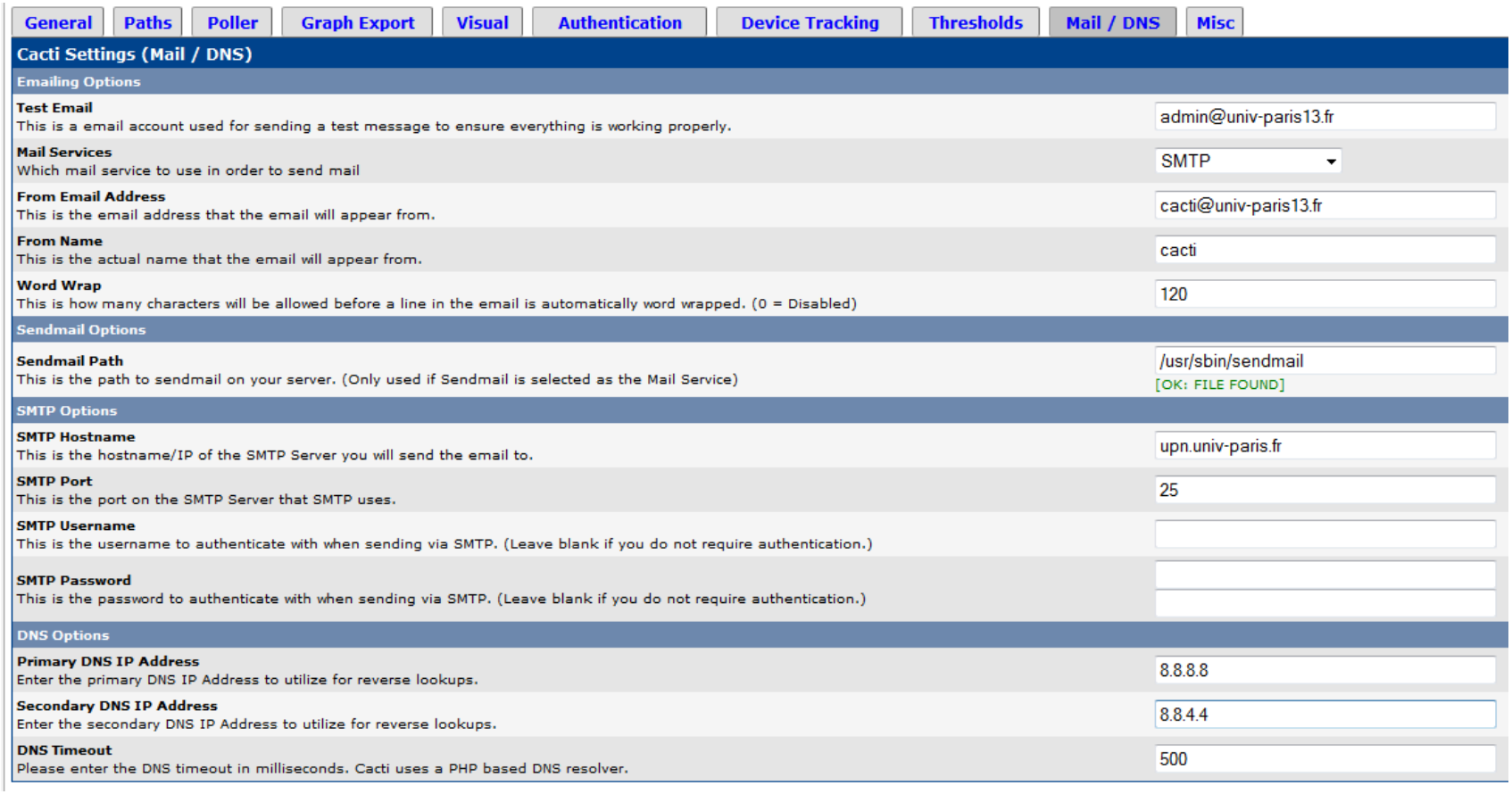
Cartographie du réseau avec Cacti : débit



Configuration de Cacti : SNMP



Configuration de Cacti : administrateur



Routage statique vs dynamique

Routage statique

Tables de routage entrées "manuellement" dans les machines

- Configuration simple, rapide, une fois pour toutes
- Routage simple à calculer
- Tout est local : pas besoin de connaître le reste de l'état du réseau

Mais toute modification du réseau nécessite une mise à jour de toutes les tables de routage.

Routage dynamique ou adaptatif

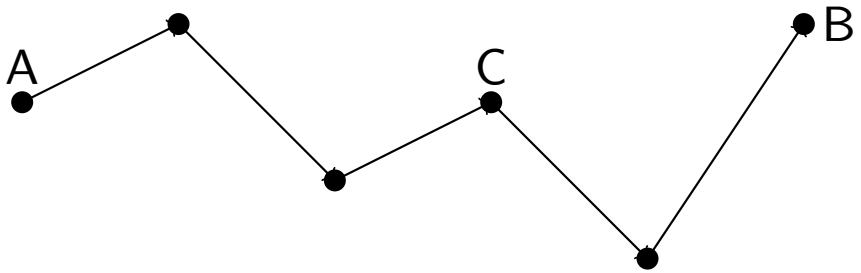
Pas de connaissance a priori du réseau

- Décisions de routage fondées sur la topologie, le trafic...
- Possibilité de modifier les décisions de routage en cas de modifications de la topologie (pannes, modification du réseau)

Principes d'optimalité

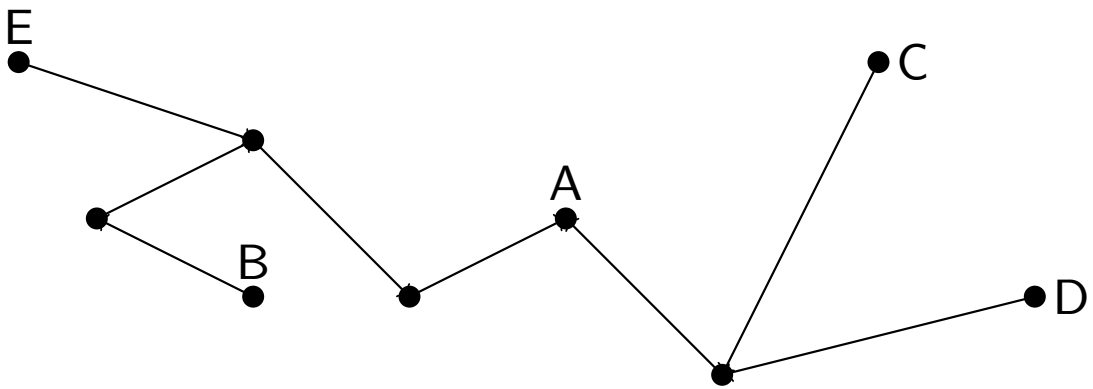
Principe d'optimalité

Si un chemin de A vers B est optimal, alors si C est sur le chemin entre A et B la portion $[AC]$ et la potion $[CB]$ de $[AB]$ sont optimales.



Arbre collecteur

Un arbre collecteur rassemble tous les chemins optimaux de toutes les sources vers une destination donnée.



Métriques pour mesurer la performance d'un algorithme de routage

Objectif

Router les messages d'une source vers une destination à moindre coût.

Comment évaluer la performance d'un algorithme de routage ? Détermination d'une métrique (ou notion de coût)

- Nombre de sauts : par combien d'intermédiaires le message doit transiter
- Temps : somme des délais de transmission
- Robustesse, résilience : le message peut-il arriver en cas de pannes
- Nombre de messages générés, congestion provoquée par la propagation du message
- ...

Un algorithme ne peut pas être optimal partout !

- Suivant le contexte, on choisira un algorithme plutôt qu'un autre.

Algorithme glouton : par inondation

Principe : on envoie le paquet à tous ses voisins, sauf le lien d'origine

- De proche en proche, le paquet finira bien par arriver à son destinataire...

Avantages

- Très *robuste* : si il y a des pannes sur le réseau, le paquet passe quand même tant que le réseau reste connexe
- Le plus court chemin est toujours trouvé

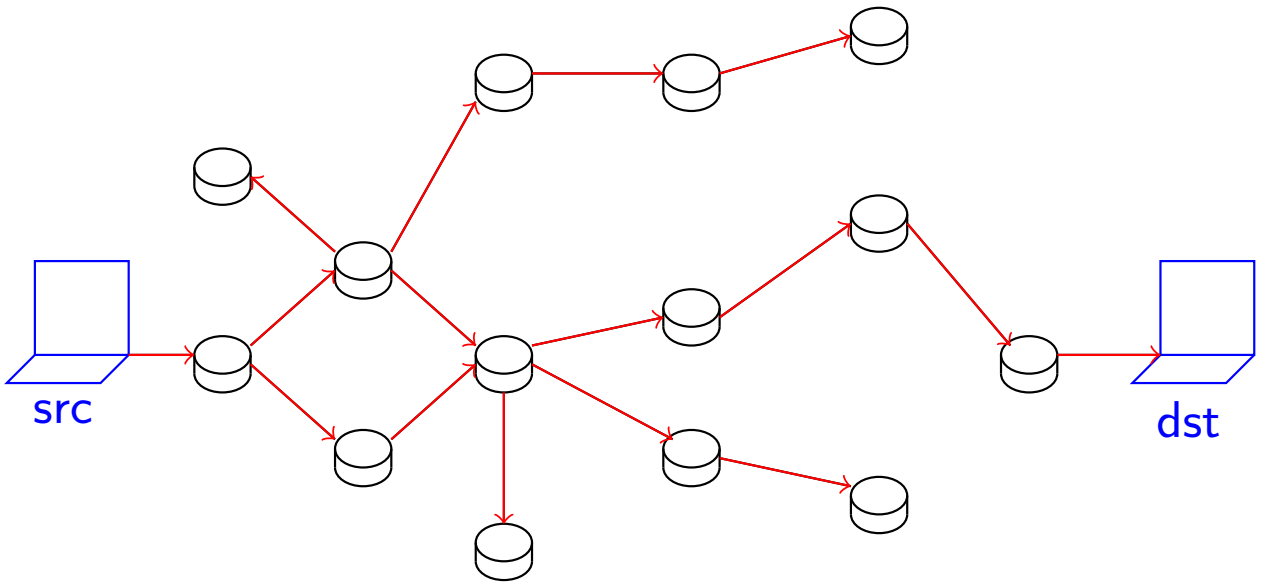
Inconvénients

- Génère un *grand* nombre de messages
  - Infini puisque chaque noeud qui reçoit le message le transmet à tous ses voisins
  - Pour éviter ce problème : utilisation du TTL
- Envoi sur plusieurs routes en même temps : messages non-indispensables (beaucoup !)

Utilisation : systèmes critiques

Exemple

Envoi d'un message de la machine *src* vers la machine *dst*



Algorithme de la patate chaude

Principe : on se débarrasse le plus vite possible du paquet

- Envoi vers le lien le moins chargé
  - à l'exclusion du lien d'origine
- Algorithme *adaptatif* : tient compte de l'état du réseau

Avantages

- Adaptatif
- Vise à diminuer la charge (on garde le paquet le moins longtemps possible)

Inconvénients

- Non optimal en nombre d'étapes pour atteindre un destinataire
- L'arrivée n'est même pas garantie !
- Possibilité de création de boucles

Évolution : routage réparti

Chaque noeud diffuse à ses voisins l'état de ses liens

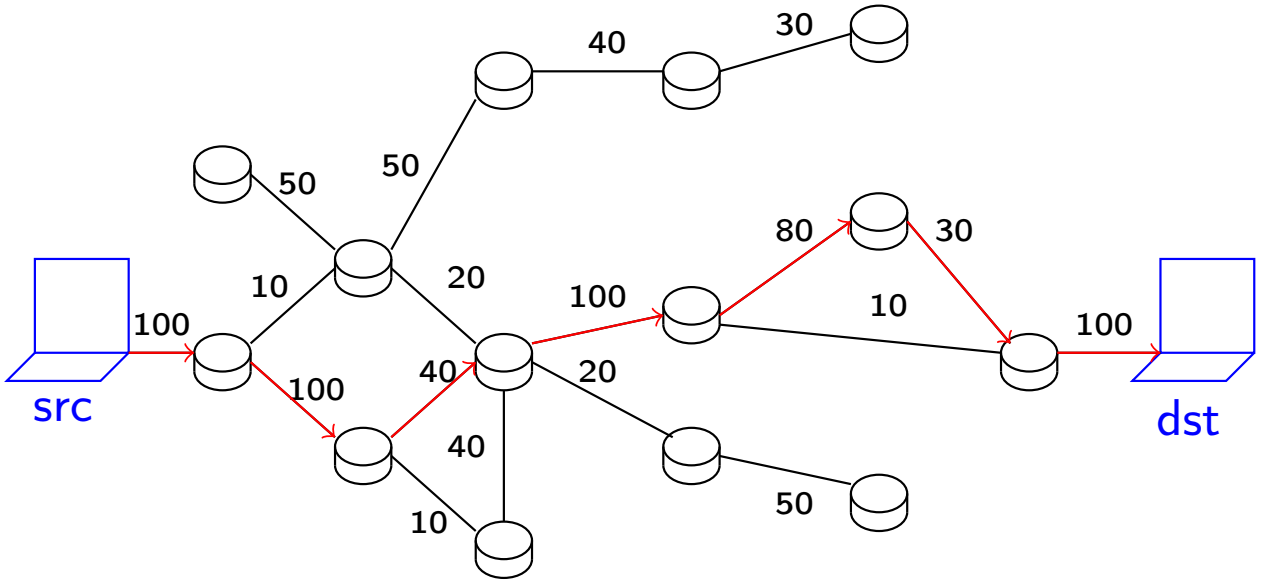
- Permet de calculer la route avec plus de visibilité

Utilisation : souvent combiné à un algorithme statique qui mémorise les routes

- Plus qu'un algorithme de routage, c'est un algorithme de sélection de route

Exemple

Considérons les débits associés aux routes du réseau suivantes. La machine *src* envoie un message vers la machine *dst*.



NB : il s'agit dans cet exemple de débits associés aux liens. Il peut aussi s'agit de chargements des liens (on va alors vers le lien le moins chargé), de latence, etc.

Algorithme à vecteurs de distance : Bellman-Ford

Principe

Chaque routeur échange avec ses voisins des informations sur les tables de routage dont il dispose.

Informations : couple (destination, coût)

- À partir de ces informations, construction d'une table de routage selon le principe d'optimalité
- Coût = généralement nombre de sauts

Quand on ne connaît pas de route vers un destinataire :

- Coût initialisé à l'infini

Idem en cas de panne :

- Tous les destinataires ayant une route passant par un bouteur tombé en panne ont un coût mis à l'infini
- Adaptation des routes selon l'algorithme

Informations diffusées

- De proche en proche, de voisin en voisin
- À l'initialisation d'un routeur qui rejoint le réseau
- À intervalles réguliers (adaptativité)

Algorithme à états de liens : Dijkstra

Principe

Chaque routeur échange avec ses voisins des informations sur l'état de ses liens.

Mesure des coûts de transmission d'un message avec ses voisins

- Généralement : latence

Diffusion des coûts à ses voisins

- Construction petit à petit d'une matrice des coûts sur le réseau
- Calcul du plus court chemin sur le réseau

Informations découvertes et diffusées

- De proche en proche, de voisin en voisin
- À l'initialisation d'un routeur qui rejoint le réseau et à intervalles réguliers (adaptativité)
- Ponctuelles : ne tiennent pas compte du trafic

Agrégation de réseaux

Internet est fait d'une agrégation de réseaux de différents opérateurs :

- Chaque opérateur route en interne ce qui reste dans son réseau
- Protocole de routage entre les réseaux de différents opérateurs

AS : Autonomous System

- Ensemble de réseaux contrôlés par une seule autorité

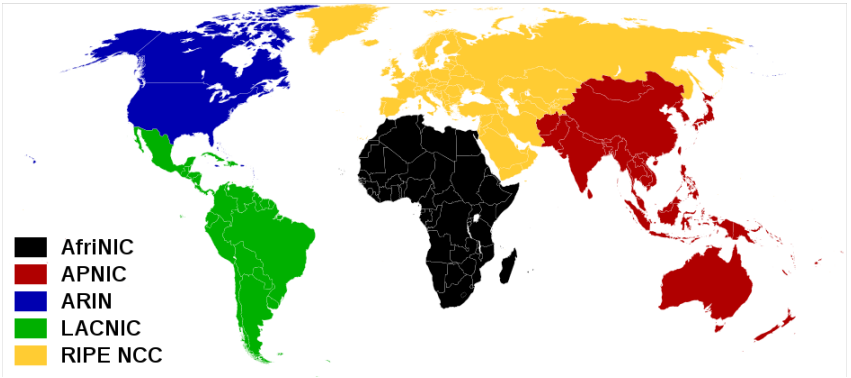
Cette autorité est régionale : il y en a 5, appelées Regional Internet Registry (RIR)

- Europe élargie : RIPE NCC (Réseaux IP Européens Network Coordination Centre)
- Afrique : AfriNIC
- Amérique du nord : ARIN
- Amérique du sud, caraïbes : LACNIC
- Asie-Pacifique : APNIC

Rôle d'un Regional Internet Registry

Allocation des ressources dans sa région

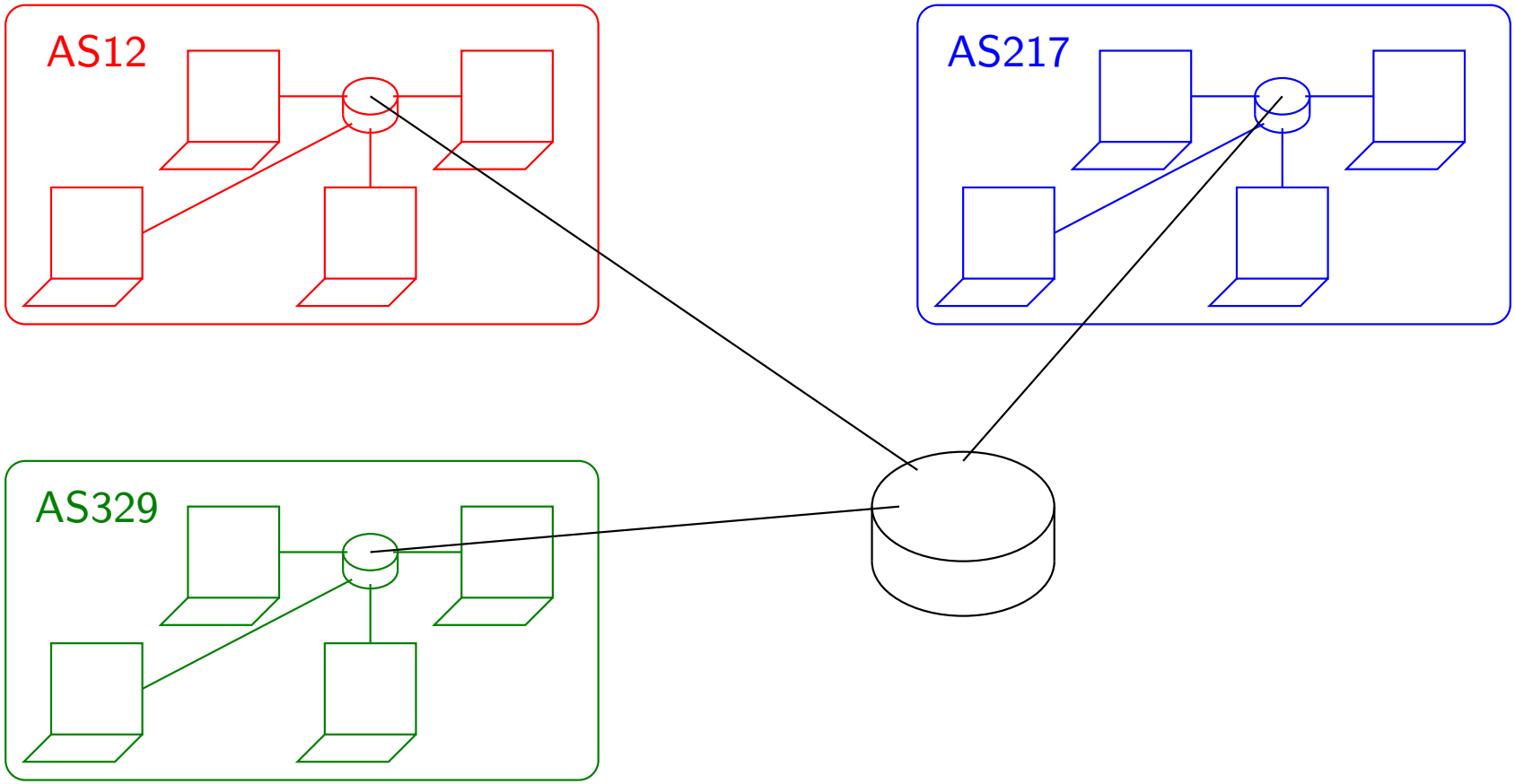
- Adresses IPv4 et IPv6
- Numéros d'AS



Autres activités :

- Maintenance de la base de données d'infos sur les réseaux de sa zone
- Maintenance des tables de routage à l'intérieur de sa zone
- Serveurs de noms racine (en Europe : K-root)
- Statistiques sur le réseau, ses performances et son développement

Agrégation de réseaux



Architecture d'Internet

Internet est découpé en systèmes autonomes

- Routage entre les systèmes autonomes : Exterior Gateway Protocol (EGP)
- Puis routage au sein d'un système autonome : Interior Gateway Protocol (IGP)

Les systèmes autonomes (AS) sont identifiés par un *numéro de système autonome*

- 2 octets puis 4 octets
- Défini par la RFC 1771 (mars 1995), mis à jour par la RFC 4893 (mai 2007)
- Alloué par le RIR

Routage sur Internet : combinaison de protocoles

- Protocoles externes (EGP) entre les AS (BGP)
  - Granularité : l'AS
- Protocoles internes (IGP) au sein de chaque AS (RIP, OSPF)
  - Granularité : le routeur

Protocole externe : BGP

Chaque AS a au moins une passerelle

- Cette passerelle lui permet de communiquer avec les autres AS
- Les messages envoyés entre différents AS transitent par cette passerelle

Chaque AS est connecté à un AS via une ou plusieurs passerelles. Ils sont *tous* interconnectés.
But d'un protocole externe : router les messages entre les passerelles des AS

- Border Gateway Protocol (BGP)
- Défini par la RFC 1771, étendu à IPv6 par la RFC 2545

BGP sélectionne les routes :

- Basé sur les IP accessibles : *best prefix match*
- En minimisant le nombre de sauts
- Variante : vecteur de distance

Notion de *peering* : router le trafic d'un AS vers un autre

- Exemple : abonné Free qui va sur Google
- Google n'est pas dans le même AS que Free : Free doit router le trafic vers l'AS de Google
- Coût ! Partagé / facturé

BGP : suite

Chaque routeur BGP est connecté avec ses voisins

- TCP, port 179 : protocole *fiable*
- Échange d'informations sur l'accessibilité IP dans les AS

Initialisation :

- Envoi de *toutes* les informations de routage

Mises à jour :

- Envoi des modifications uniquement
- Utilisation d'un marqueur (timestamp) donnant le numéro de mise à jour

Connexions TCP : mode connecté

- Envoi de messages *keepalive* à intervalles réguliers

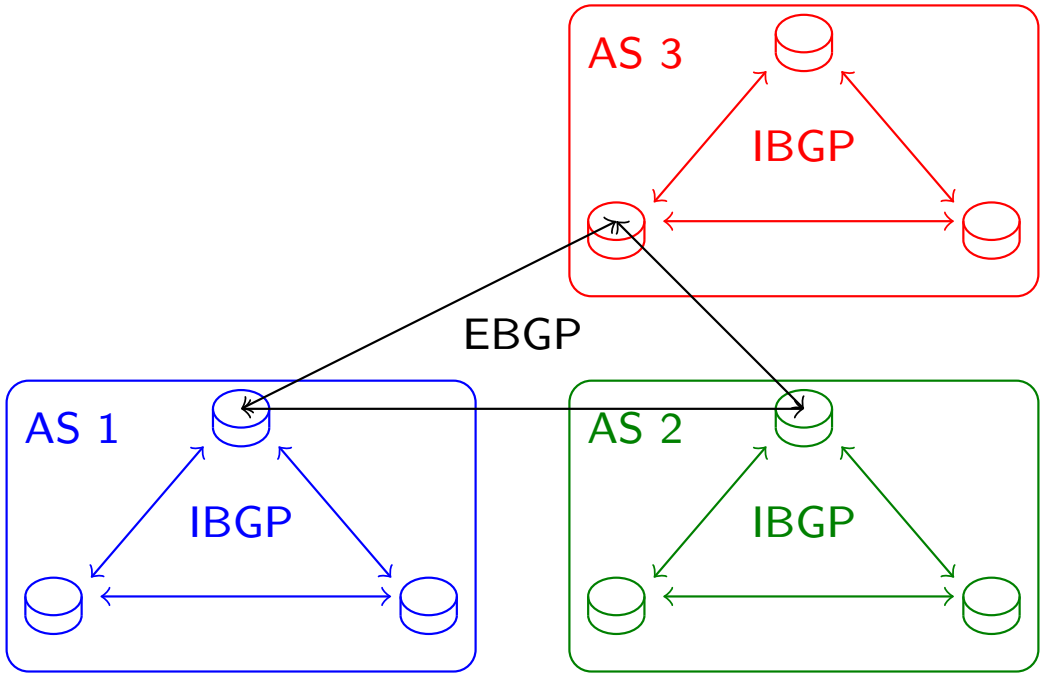
Points BGP

BGP distingue deux sortes de points (= routeurs) du point de vue d'un routeur :

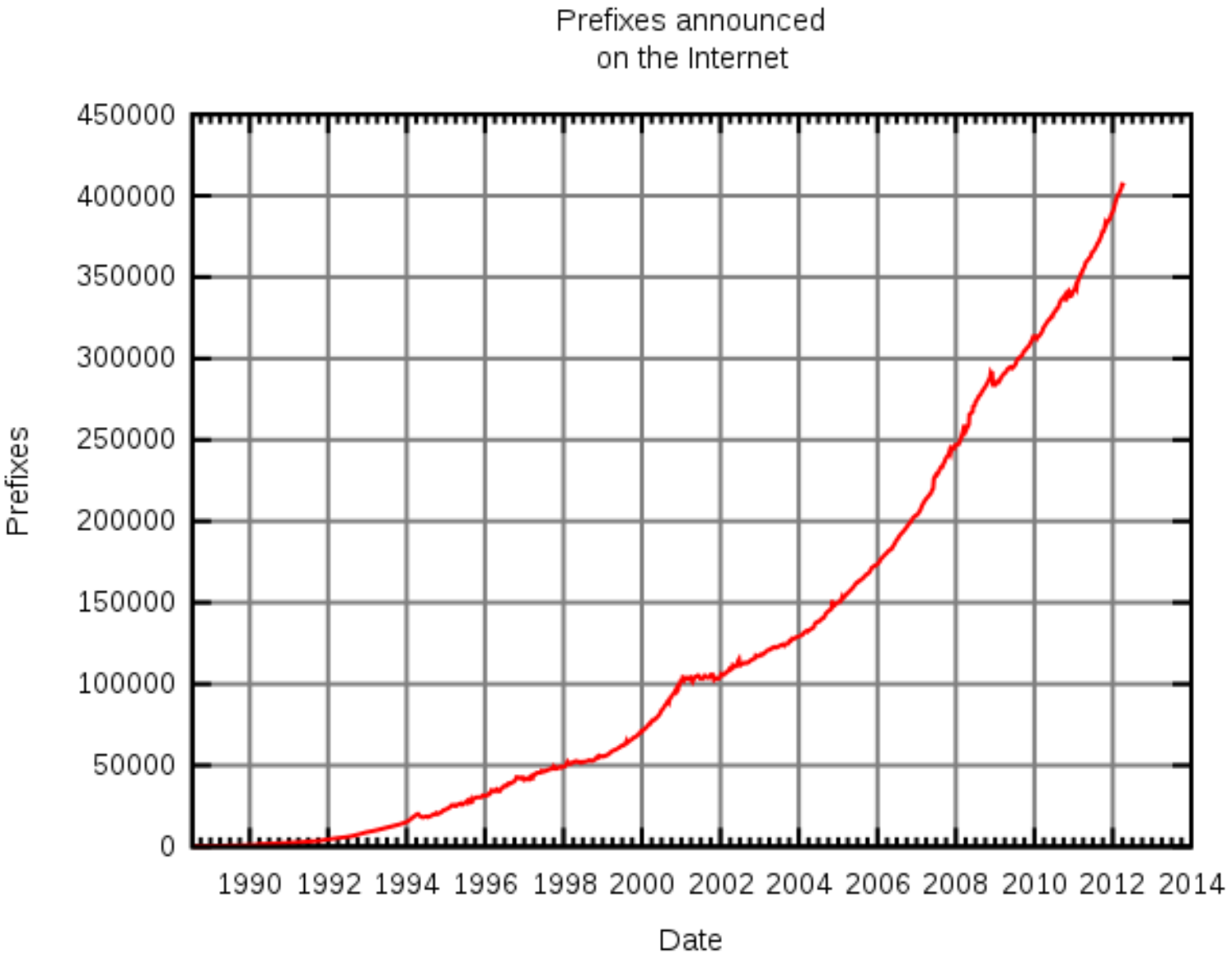
- Les points internes : points qui ont le même numéro d'AS que soi-même.
- Les points externes : points qui ont un numéro d'AS différent du sien.

On distingue alors deux parties du protocole :

- External BGP (EBGP) : messages échangés entre les points de différents AS
- Les informations obtenues sont transférées dans l'AS entre les routeurs BGP via Internal BGP (IBGP)



Taille des tables de routage



Source : Wikipedia

Détection de boucles

Mécanisme de détection de boucles

- Les mises à jour contiennent le chemin par lequel le paquet de mise à jour a transité (numéros d'AS)
- On ne transfère pas à un AS dont le numéro est déjà présent dans le chemin

Attention : fonctionne uniquement entre AS, pas à l'intérieur d'un AS

- Pour éviter les problèmes en interne : les routeurs BGP au sein d'un AS sont tous connectés les uns aux autres (= topologie de graphe complet)

Hiérarchie du protocole

Le protocole est alors hiérarchique :

- Les passerelles des AS parlent entre eux
- Les informations sont transmises par les passerelles à l'intérieur de leur AS

Avantages :

- Réduction de la taille des tables de routage (factorisation)
- Réduction du nombre de connexions entre routeurs !

Inconvénients :

- Temps de transmission des mises à jour des routes
- Updates toutes les 90 secondes : si une route a été modifiée, on peut ne pas le savoir
- Certains routeurs peuvent être accessibles ou non par intermittence. Les temps de propagation des mises à jour continuelles rend le système incohérent : on parle de *route flapping*

En-tête BGP

Taille des champs en bits

Marqueur (128)		
Longueur (16)	Type (8)	Bourrage (8)

- Marqueur : généralement tous les bits mis à 1 (compatibilité)
- Longueur : longueur totale du message (incluant l'en-tête) en octets
- Type : type de message BGP
  - 1 : open
  - 2 : update
  - 3 : notification
  - 4 : keepalive
  - 5 : route-refresh

Messages open

Type de message 1.

Utilisés à l'ouverture d'une session BGP (*i.e.*, lorsqu'un routeur est connecté) pour s'annoncer Format d'un message d'erreur :
Taille des champs en bits

Version (8)	Num. d'AS (16)	HDT (16)
ID BGP	Lg options (8)	
Paramètres optionnels (lg variable)		

- HDT : Hold Down Timer
- Lg options : longueur des paramètres optionnels
- ID BGP : identifiant BGP du message, donnant l'identifiant (sur 32 bits) du routeur (= adresse IP)

Messages update

Type de message 2.

Contiennent les routes elles-mêmes :

- Au démarrage : toutes les routes jusqu'à ce que toute la table de routage soit passée
- En fonctionnement : modifications du réseau (nouvelles routes et routes inaccessibles)

Format : champs de longueurs variables

- Routes inaccessibles
  - Longueur (16 bits)
  - Routes inaccessibles
- Attributs de chemin
  - Longueur (16 bits)
  - Chemin
- Informations d'accessibilité réseau
  - Longueur (8 bits)
  - Préfixe, en notation 204.129.10/24
  - ...

Messages notification

Type de message 3.

Servent à prévenir que quelque chose s'est mal passé

- Une option non supportée a été envoyée dans un message open
- Un routeur n'a pas envoyé de keepalive ni d'update

Format d'un message d'erreur :
Taille des champs en bits

Code d'erreur (8)	Sous-code (8)	*Données (16)*
Données...		

Codes d'erreur :

- 1 : Erreur dans un en-tête
- 2 : Erreur dans un message OPEN
- 3 : Erreur dans un message d'update
- 4 : Timer Hold Down expiré
- 5 : Erreur d'un état
- 6 : fin

Messages keepalive

Type de message 4.

- Envoyés à intervalles réguliers
- Maintiennent la session ouverte en l'absence de modifications
- Le timer FIT (Keepalive Interval Timer) spécifie cet intervale dépend de la configuration de chaque routeur (par défaut 60 secondes)

Si un routeur BGP rate trois messages keepalive (par défaut, 180 secondes de silence)

- Il est suspecté d'être HS ou inatteignable
- On attend l'écoulement du timer HDT (Hold Down Timer) pour supprimer toutes les routes en provenance de ce routeur

Protocoles internes

Routage au sein d'un AS, entre les routeurs.

- Un protocole donné au sein d'un AS
- Différents AS peuvent ne pas utiliser le même protocole

Routing Information Protocol (RIP) :

- Routage s'appuyant sur l'algorithme de *Bellman-Ford*
- Objectif : minimiser le nombre de sauts effectués entre la source et le destinataire
- RFC 1058 (RIP 1), RFC 2080 (RIPng, support de IPv6), RFC 2453 (RIP 2)

Plutôt pour des petits réseaux : maximum 15 routeurs.

Open Shortest Path First (OSPF) :

- Routage s'appuyant sur l'algorithme de *Dijkstra*, s'affranchissant de la limite de RIP à 15 routeurs
- Objectif : minimiser le coût de communication entre 2 routeurs
- Routage *hiérarchique* : segmentation des ensemble de routeurs, notion d'aire
- ORFC 1131 (OSPF v1), RFC 2328 (OSPF v2), RFC 5340 (OSPF v3, support de IPv6)

Routing Information Protocol

Chaque routeur annonce périodiquement (toutes les 30 secondes environ) tous ses réseaux et le nombre de sauts pour les atteindre

- Chaque machine et chaque routeur écoute les annonces de ses passerelles et actualise sa table de routage.
- Timeout : si au bout d'un certain temps un réseau n'est plus annoncé, il est supprimé de la table de routage
- Les modifications sont annoncées immédiatement.

Timers utilisés :

- Entre deux annonces : 30 secondes, plus un petit nombre de secondes tiré aléatoirement pour éviter que tous les routeurs s'annoncent en même temps.
- Pour considérer une route comme invalide : 90 secondes d'inactivité (aucune mise à jour ni annonce). Le routeur annonce à ses voisins que la route est invalide.
- Pour retirer complètement de la liste des routes : 240 secondes.

Paquets RIP 1

Protocole utilisant UDP sur le port 520

Taille des champs en bits

Commande (8)	Version (8)	Zéros (16)
ID de famille d'adresse (16)		Zéros (16)
Adresse IP (32)		
Zéros (32)		
Zéros (32)		
Métrique (32)		
Payload		

- Commande :
  1. Request : demande tout ou partie d'une mise à jour de table d'un autre routeur RIP
  2. Response : réponse à une request. Toutes les mises à jour de routes utilisent cette commande
  3. Traceon : obsolète et ignoré
  4. Traceoff : obsolète et ignoré
  5. Reserved : Utilisé par les routeurs Sun
- Version : version du protocole RIP (1 ou 2)
- ID de famille d'adresse : identifie le protocole d'adressage utilisé (CLNS, IPX, IP...)
- Métrique : indique combien de routeurs ont été traversés. 1 à 15 : route valide, 16 = route impossible à atteindre

Paquets RIP 2

Taille des champs en bits

Commande (8)	Version (8)	Zéros (16)
ID de famille d'adresse (16)		Route tag (16)
Adresse IP (32)		
Masque de sous-réseau (32)		
Saut suivant (32)		
Métrique (32)		
Payload		

- Route tag : permet de distinguer les routes entre elles, notamment interne et externes
- Saut suivant : adresse IP du prochain routeur auquel le paquet doit être envoyé
- Adresse IP, masque : ceux de l'entrée dont il est question
- Payload : Entrées de la table de routage, au maximum 25

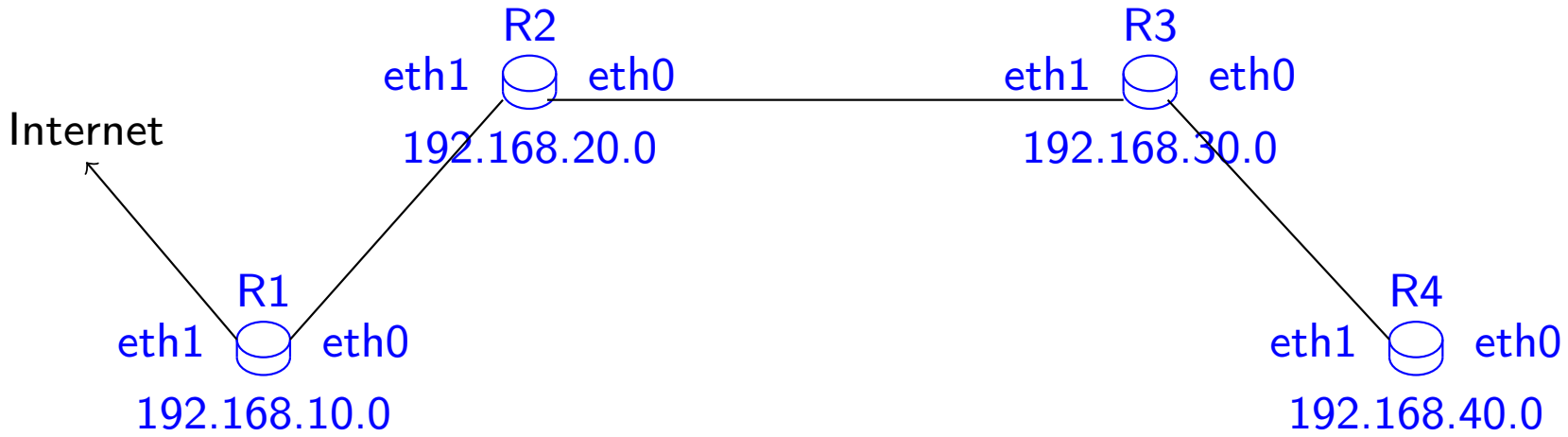
Fonctionnement de RIP

Au démarrage :

- Un routeur RIP diffuse une requête pour demander à ses voisins leurs tables de routage
- Les voisins envoient une réponse contenant leurs tables de routage
  - Si on reçoit une nouvelle route : on l'insère dans la table de routage
  - Si on reçoit une meilleure route : on remplace la route existante par la nouvelle
- Le routeur qui vient de démarrer envoie alors une mise à jour contenant sa table de routage
  - Ses voisins peuvent mettre à jour leurs tables de routage en cas de nouvelle ou de meilleure route

Exemple

Considérons le réseau suivant :



Extraits des tables de routage (simplifiées) :

R1 :

Adresse	Interface
192.168.20.0	eth0
192.168.30.0	eth0
192.168.40.0	eth0
*	eth1

R2 :

Adresse	Interface
192.168.30.0	eth0
192.168.40.0	eth0
*	eth1

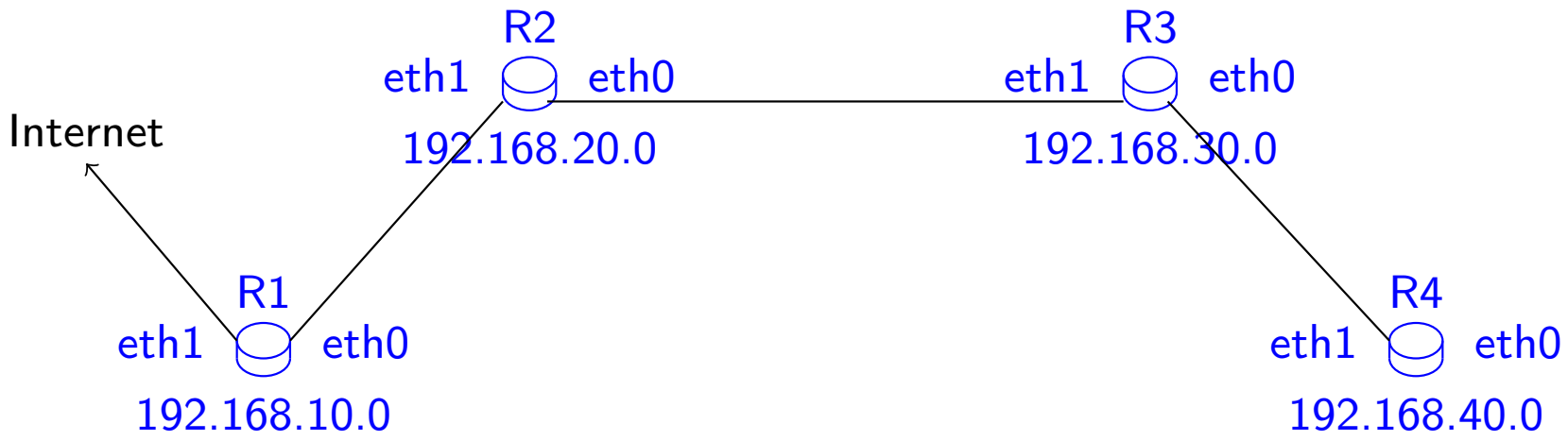
R3 :

Adresse	Interface
192.168.40.0	eth0
*	eth1

R4 :

Adresse	Interface
*	eth1

Exemple

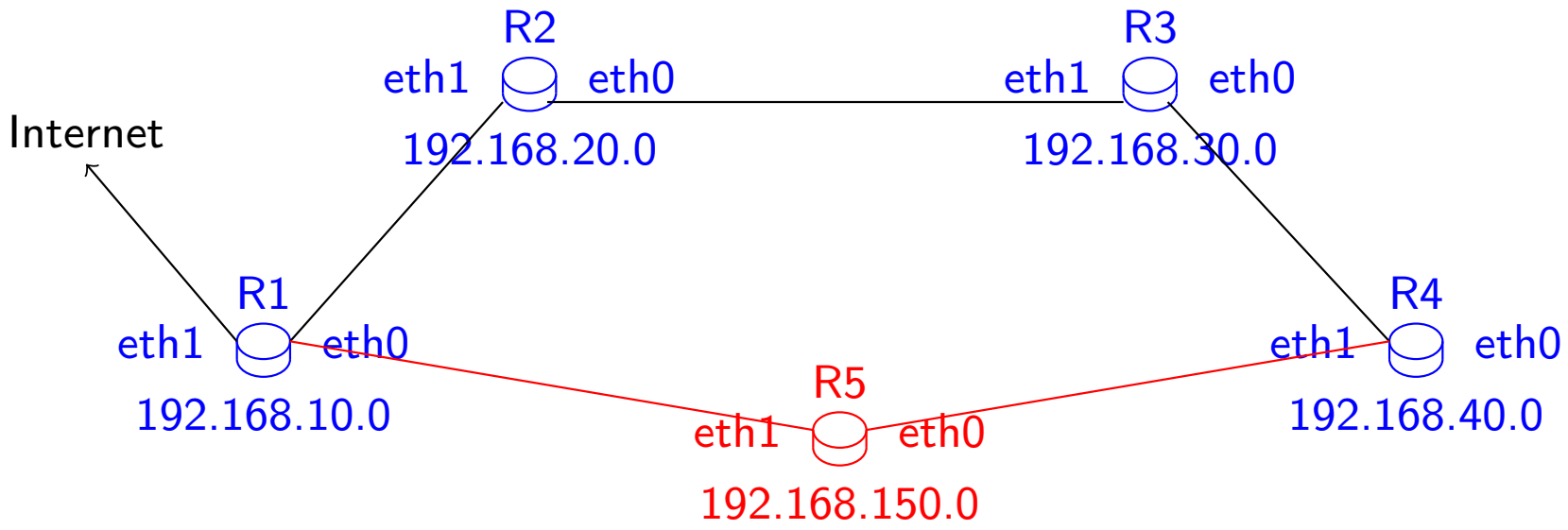


Nombre de sauts nécessaires entre deux routeurs :

	R1	R2	R3	R4
R1	0	1	2	3
R2	1	0	1	2
R3	2	1	0	1
R4	3	2	1	0

Exemple

On ajoute un routeur dans le réseau :



- R5 envoie une requête à ses voisins : R1 et R4
- R1 et R4 lui envoient une réponse contenant leurs tables de routage avec le nombre de sauts nécessaires incrémenté de 1

Réseau	Nombre de sauts	Passerelle	Réseau	Nombre de sauts	Passerelle
192.168.10.0	1	R1	192.168.40.0	1	R4
192.168.20.0	2	R1	192.168.30.0	2	R4
192.168.30.0	3	R1	192.168.20.0	3	R4
192.168.40.0	4	R1	192.168.10.0	4	R4
*	2	R1	*	5	R4

Exemple

R5 choisit les routes les plus courtes pour atteindre chaque réseau :

Réseau	Nombre de sauts	Passerelle
192.168.10.0	1	R1
192.168.20.0	2	R1
192.168.30.0	2	R4
192.168.40.0	1	R4
*	2	R1

Et R5 envoie sa table de routage avec le nombre de sauts à R1 et R4 pour qu'ils actualisent les leurs :

- En passant par R5, le routeur R1 peut atteindre R4 en deux sauts (et inversement)

Comme R1 et R4 ont modifié leurs tables de routage, ils envoient une mise à jour à leurs voisins, respectivement R2 et R3.

OSPF

OSPF est un protocole à états de liens

- La métrique est le coût associé à une route ou un lien.

Le coût d'une route est la somme des coûts associés aux liens empruntés par la route.

Le réseau est vu de façon hiérarchique et divisé en zones

- plusieurs zones
- un backbone

OSPF nécessite plus de mémoire et de puissance de calcul que d'autres protocoles comme RIP.

Routeurs OSPF

OSPF définit plusieurs types de routeurs :

- Routeur interne (Internal Router, IR) : connecté à une et une seul zone OSPF
- Routeur de backbone (Backbone routeur, BR) : connecté au backbone par au moins une de ses interfaces
- Routeur de frontière de zone (Area Border Router, ABR) : connecté à plus d'une zone, généralement pour connecter le backbone à une autre zone
- Routeur de frontière de système autonome (Autonomous System Boundary Router, ASBR) : connecte un AS à un autre AS
- Routeur élu (Designated Router, DR) : un routeur élu parmi les routeurs d'une zone
- Routeur élu suppléant (Backup Designated Router, BDR) : prend le relai du DR en cas de panne

Zones OSPF

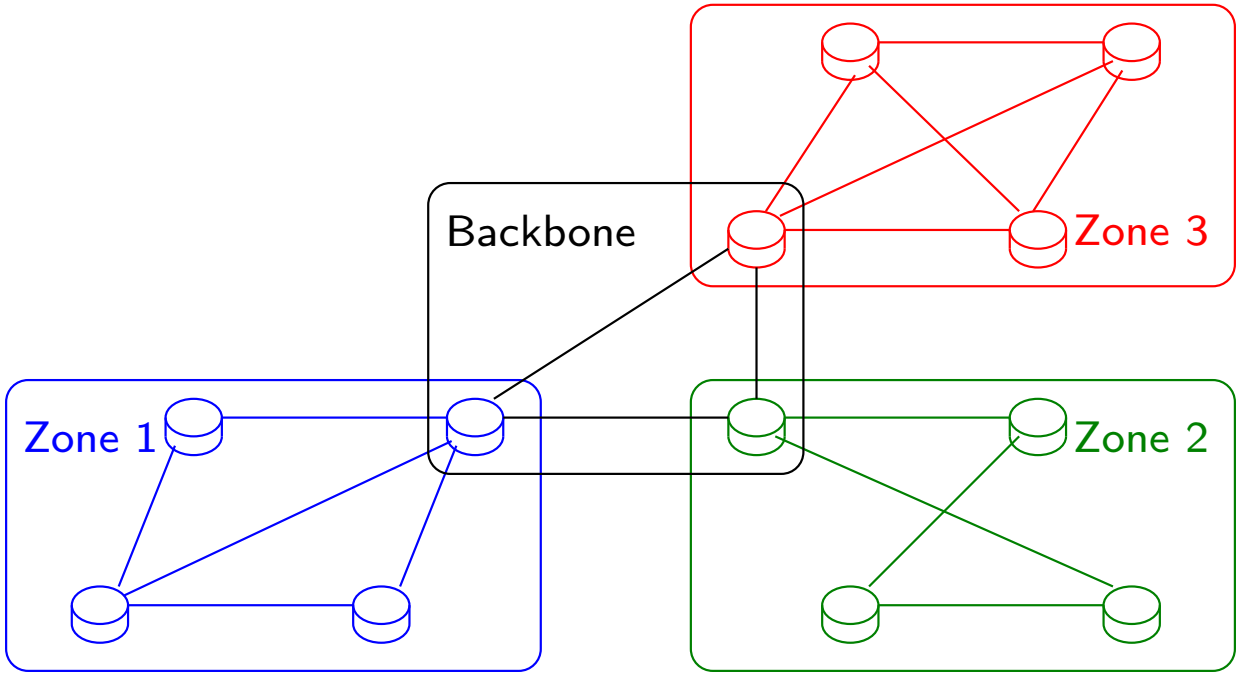
- Un réseau utilisant OSPF a toujours au moins une zone
- Si il compte plus d'une zone, l'une d'entre elles est le backbone du réseau

Seulement 2 niveaux de hiérarchie : le backbone et les autres zones.

Différents types de zones :

- Backbone (zone 0) : zone à laquelle toutes les zones sont reliées. La première zone à structurer. Attention : même en cas de panne, le backbone ne doit pas être scindé en deux.
- Zone stub (Stub Area, SA) : zone connectée uniquement au backbone, ne recevant pas de route en provenance d'en-dehors de l'AS mais uniquement de l'intérieur de l'AS (éventuellement d'autres zones également).
- Zone totalement stub (Totally Stub Area, TSA) : zone connectée uniquement au backbone. Une TSA ne propage pas les routes qu'elle connaît (pas de LSA envoyés) : une route définie par défaut (vers le backbone) lui permet de communiquer avec le reste du réseau.
- Zone pas vraiment stub (Not-So-Stubby Area, NSSA) : zone spécifique à un sous-réseau reliant le reste du réseau à Internet. On utilise un routeur de frontière de système autonome à l'intérieur et un routeur de frontière de zone pour relier au backbone, et des LSA d'un type particulier (type 7).
- Liens virtuels : utilisés pour relier deux réseaux utilisant OSPF.

Zones OSPF



OSPF – Timers

Trois timers sont utilisés par OSPF :

- Timer HELLO : par défaut, 10 secondes
- Timer DEAD : généralement 4 fois le timer HELLO, soit 40 secondes par défaut
- Fréquence de mise à jour de la base des états des liens : à chaque fois que la table des routes est modifiée (découverte de nouvelles routes), ou sinon 30 minutes après la dernière mise à jour.

En-tête OSPF

Utilisé directement au-dessus d'IP.

Taille des champs en bits

Version (8)	Type (8)	Longueur (16)
ID de routeur (32)		
ID de zone (32)		
Checksum (16)		Type d'auth. (16)
Authentification (64)		
Payload (LSA)		

Type : type de LSA

- 1 : HELLO
- 2 : description de la base de données
- 3 : requête d'état de lien
- 4 : mise à jour de l'état d'un lien
- 5 : acquittement de l'état d'un lien

Type d'authentification :

- 0 : aucune
- 1 : mot de passe simple
- 2 : cryptographique
- 3 à 65535 : réservé

En-tête OSPFv3

Évolutions depuis les versions précédentes : possibilité d'utiliser plusieurs instances d'OSPF sur un même lien, prise en charge de IPv6

Taille des champs en bits

<table>
<tr><td>Version (8)</td><td>Type (8)</td><td colspan="2">Longueur (16)</td></tr>
<tr><td colspan="4">ID de routeur (32)</td></tr>
<tr><td colspan="4">ID de zone (32)</td></tr>
<tr><td colspan="2">Checksum (16)</td><td>ID d'instance (8)</td><td>Réservé (8)</td></tr>
<tr><td colspan="4">Authentification (64)</td></tr>
<tr><td colspan="4">Payload (LSA)</td></tr>
</table>

ID d'instance : permet d'utiliser plusieurs instances d'OSPF sur un même lien. Signification locale (sur un lien donné) uniquement.

Le routeur élu

But : limiter le trafic des messages HELLO

- Si tout le monde diffuse ses messages HELLO toutes les 10 secondes : beaucoup de trafic
- Donc charge du réseau

Effet : réduction du nombre de diffusions et donc du nombre de messages protocolaires en circulation.

Utilisation des routeurs élus :

- Chaque routeur envoie son message HELLO au routeur élu
- Le routeur élu agrège et diffuse les messages HELLO aux autres routeurs

Découverte de voisins

Quand un nouveau routeur OSPF est connecté au réseau :

- Il envoie sur toutes ses interfaces un LSA de type HELLO à l'adresse 225.0.0.5 (adresse IP de multicast OSPF)

Ses voisins reçoivent le paquet et en déduisent :

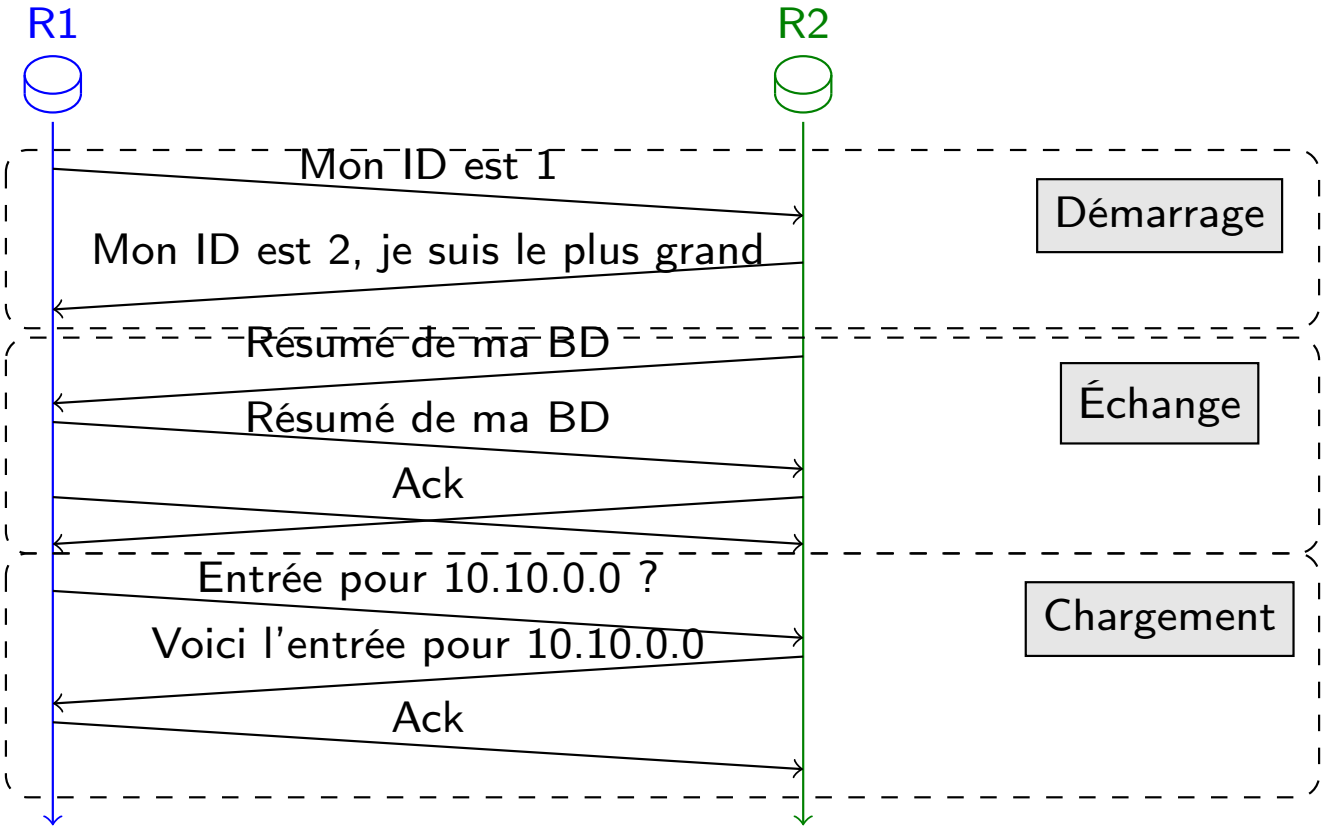
- Le routeur ID du noveau routeur
- La zone à laquelle il appartient (ID de zone)

Initiation d'une conversation entre 2 routeurs : comparaison des routeur IDs, celui qui a le plus grand commence l'échange de données

D'où :

- Découverte de nouvelles routes
- Calcul des chemins optimaux d'après les mises à jour
- Mise à jour des tables de routage
- Diffusion des mises à jour (messages de type 2 : description)

Paquets échangés



Autres protocoles de routage dynamique

Intermediate System to Intermediate System (IS-IS)

- À états de liens
- Définit des systèmes terminaux (les machines des utilisateurs), des systèmes intermédiaires (les routeurs), des zones (groupes locaux) et des domaines (regroupement de plusieurs zones)
- Hiérarchique
- Utilise son propre système d'adressage

Interior Gateway Routing Protocol (IGRP

- Protocole propriétaire CISCO
- Supporte plusieurs métriques pour chaque route : bande passante, latence, charge... Combinaison de plusieurs critères par une somme pondérée
- Créé à l'origine pour outrepasser les limites de RIP (15 sauts)

Enhanced Interior Gateway Routing Protocol (EIGRP)

- Protocole propriétaire CISCO
- Utilise l'algorithme DUAL (Diffusing Update ALgorithm)
- Évolution de IGRP, compatible
- Algorithme hybride vecteur de distance / état de liens
- Consomme peu de bande passante : pas de mises à jour régulières, pas de diffusions, seules les modifications sont communiquées

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

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 Transfer Process)

- 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

Réponses FTP

Le client envoie une commande sur le canal de contrôle, le serveur répond un code de réponse :

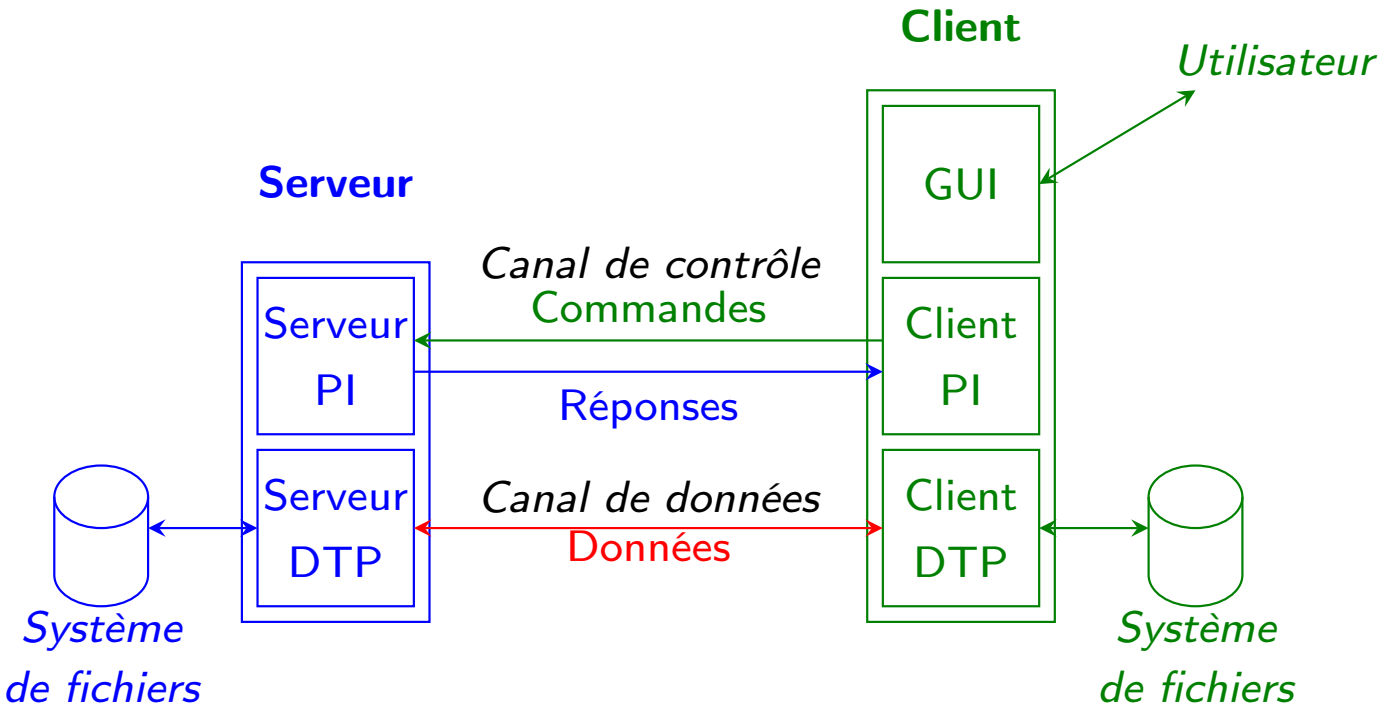
- En trois chiffres
- 1er chiffre : status de la requête
- 2eme chiffre : référence de la réponse
- 3eme chiffre : spécifie la réponse

Exemples :

- 1xx : réponse préliminaire positive
  - 125 : canal de données déjà ouvert ; début du transfert
- 2xx : réponse positive de réalisation
  - 200 : ok
- 3xx : Réponse intermédiaire positive (action supplémentaire nécessaire)
  - 331 : nom d'utilisateur ok ; entrer le mot de passe.

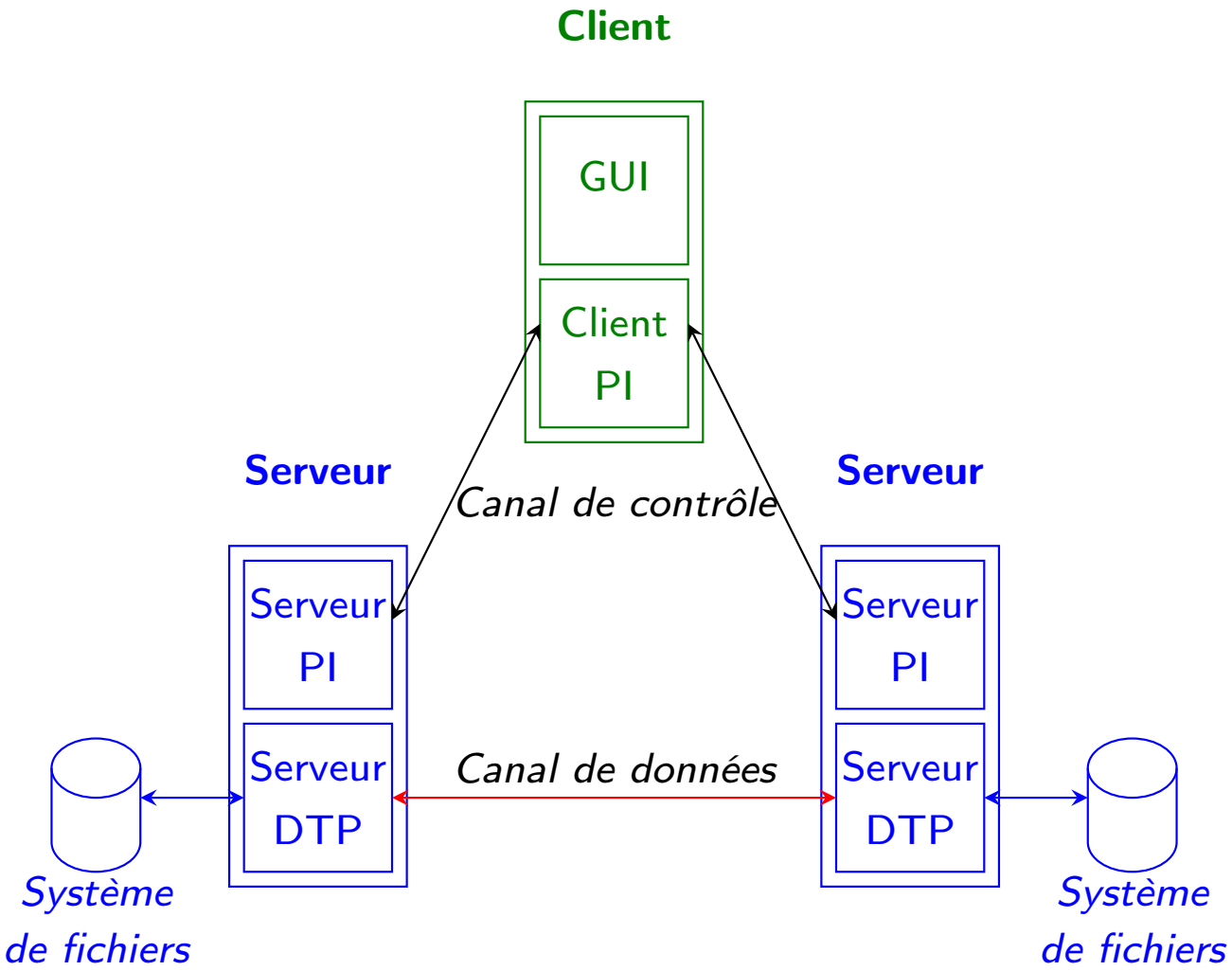
Fonctionnement de FTP : client-serveur

Transfert de fichiers entre un client et un serveur (dans un sens ou dans l'autre):



Fonctionnement de FTP : serveur-serveur

Transfert de fichiers entre deux serveurs:



Serveurs FTP

- vsftpd : très répandu, Unix/Linux, axé sur la sécurité
- zftp : transfert de fichiers compressés Zebra
- Enhanced Customer Data Repository (ECuRep FTP, IBM)
- Ipswitch WS_FTP (Windows)
- IIS (Microsoft)

Sécurité avec FTP

FTP n'a pas été conçu pour être sécurisé !

- On parle de "secure FTP"
- Ensemble de méthodes pour sécuriser un transfert FTP

FTPS

- Cryptage de la session FTP utilisant TLS
- Explicite : commande AUTH TLS envoyée par le client
- Implicite : port TCP 991 (contrôle) et 989 (données)
- Le serveur peut ou non accepter les connexions non cryptées

SFTP

- Extension de SSH pour utiliser FTP sur SSH

Accès FTP

Le serveur définit ce que chaque utilisateur peut faire

- Il faut donc commencer par authentifier les utilisateurs

Par défaut : authentification en clair

- Le mot de passe circule en clair sur le réseau !!

Utilisation d'une méthode d'authentification sécurisée (FTPS, SafeTP, TLS FTP, proxy...)

Accès anonyme = attention

- Un utilisateur non-identifié ne doit pas pouvoir modifier de fichiers sur le serveur (sauf cas exceptionnes – imprimantes...)

Ne pas donner au serveur FTP accès à tout le disque

- Limiter les droits de l'utilisateur qui fait tourner le serveur FTP
- Utiliser `chroot`...

Exemple avec vsftpd

Utilisateurs = utilisateurs de la machine locale

- Utilisateurs interdits = `/etc/ftpusers` (root, utilisateurs système...)

Fichier de configuration : `/etc/vsftpd.conf`

Accès anonyme autorisé :

- `anonymous_enable=Yes`

Autoriser les utilisateurs authentifiés à déposer des fichiers :

- `write_enable=YES`

Limiter l'accès aux fichiers locaux :

- `chroot_local_user=YES`
- `chroot_list_enable=YES`
- `chroot_list_file=/etc/vsftpd.chroot_list`

Utilisation de FTPS :

- `ssl_enable=Yes`
- `rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem`
- `rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key`

13 HTTP et Web

- Présentation
- Protocole
- Architecture
- Serveur Web Apache HTTP

Présentation

HTTP, HTML et Web

Le protocole HTTP et le langage d'affichage HTML sont liés depuis leur conception

- Serveur HTTP
- Navigateur
  - communique avec le serveur HTTP
  - affiche du HTML

L'apparition de HTTP et HTML est considérée comme la naissance du *World Wide Web*

Le World Wide Web

Ensemble de documents

- contenant des références les uns vers les autres (*hyperliens*)
- accessibles via Internet

Histoire

Débuts au CERN

But : améliorer les diffusions des données en interne au CERN

- Début du travil : 1984
- Création du WWW et proposition pour le projet HyperText : 1990
  - Premier navigateur
  - Trois concepts de base : URL, HTML HTTP

Normalisation

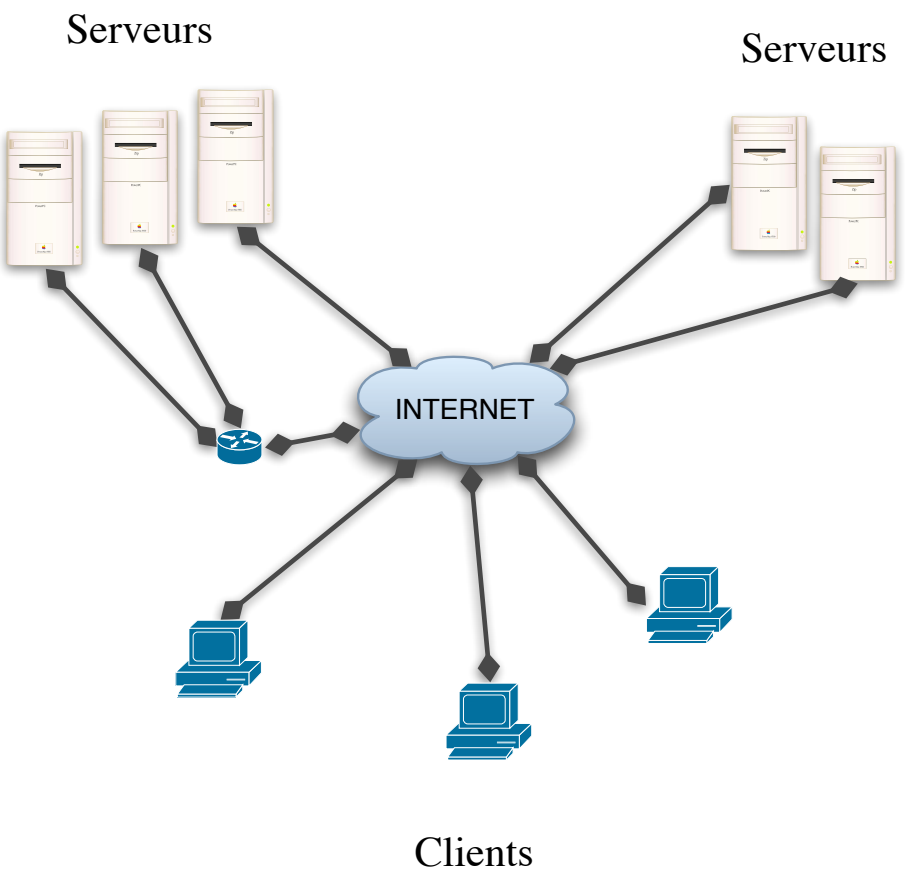
Publication de RFC :

- 1994 : RFC 1738 (notion d'URL), 2010 : RFC 5785 (notion d'URI)
- 1995 : RFC 1866 (HTML 2.0)
- 1996 : RFC 1945 (HTTP/1.0)
- 1997 puis 1999 puis 2000 : RFC 2068 puis 2616 puis 2817 (HTTP/1.1)

Consortiums :

- HTTP Working Group
- World Wide Web Consortium (W3C)

Le World Wide Web



Modèle client-serveur

- Internet relie les clients et les serveurs
  - Les documents sont sur les serveurs
  - Les clients accèdent aux serveurs et récupèrent les documents

Communication client-serveur

Protocole HTTP

- Requête-réponse

Affichage par le client

Langage d'affichage HTML

- Formattage de l'affichage

Protocole de transfert de données

Protocole de transfert de données

- FTP ne suffit pas au transfert de documents
- Affichage dans un navigateur
- Liens entre les documents

Protocole réseau

TCP/IP généralement utilisé

- Mais pas d'hypothèse faite a priori dans le protocole
- Mode connecté
- Les données transmises sont considérées comme un flux

Transmission d'une page

Modèle client-serveur

Le serveur tourne en permanence

- Un client se connecte et envoie une requête
- Le serveur lui répond
- Fin de la conversation

Communication mode ASCII

Requête : mode texte

- Syntaxe : `COMMANDE RESSOURCE`

Réponse : mode texte

- Encodage en texte, utilisation du type MIME pour les contenus non-texte

Commandes disponibles

Principales commandes

Toutes les commandes ne sont pas forcément disponibles ! (sécurité)

- `GET` : demande d'une ressource (commande la plus utilisée)
- `POST` : ajout d'une nouvelle ressource, contient des données
- `HEAD` : obtenir des informations sur la ressource
- `OPTIONS` : obtenir des informations sur les possibilités offertes par le serveur
- `CONNECT` : utilisation d'un proxy
- `TRACE` : retourner ce que le serveur a reçu (utilisé comme diagnostic)

Commandes spécifiques

Nécessitent un accès privilégié (authentifié)

- `PUT` : remplacer une ressource sur le serveur
- `DELETE` : supprimer une ressource du serveur

Exemple : récupération d'une page

Connexion au serveur

```
$ telnet www.iutv.univ-paris13.fr 80
Trying 194.254.173.2...
Connected to www.iutv.univ-paris13.fr.
Escape character is '^]'.
```

Demande d'une page

```
GET /iutv/index.php
```

Réponse du serveur

```
<HTML> [...]
</HTML>

Connection closed by foreign host.
```

- C'est ce qui est interprêté par le navigateur

Demandes partielles

GET conditionnel

Page récupérée seulement si elle a été modifiée depuis une date donnée

- If-Modified-Since: Sat, 29 Oct 1994 19:43:31 GMT

Si la page n'a pas été modifiée :

- Le serveur n'envoie qu'une ligne (code 304)
- Le navigateur affiche une version en cache

Sinon :

- Le serveur envoie la dernière version

HEAD

Ne demande que l'en-tête de la page

- Permet de déterminer la date de dernière modification
- Obtient les informations d'indexation de la page
- Vérifie la validité d'une URL

Réponse du serveur

La réponse du serveur vient avec un code (3 chiffres) indicateur du résultat de la requête :

- 1xx : information
- 2xx : succès
- 3xx : redirection
- 4xx : erreur du client
- 5xx : erreur du serveur

200 : OK

200 : requête traitée avec succès

100 : OK

100 : attente de la suite de la requête

401 : Unauthorized

401 : authentification nécessaire pour accéder à la ressource demandée

403 : Forbidden

403 : authentification refusée

404 : Not found

404 : la ressource demandée n'a pas été trouvée

408 : Request timeout

408 : temps d'attente d'une réponse du serveur écoulé

409 : Conflict

409 : la requête ne peut être traitée pour le moment (conflit de versions)

413 : Request entity too large

413 : requête trop grosse

414 : Request IRU too long

414 : l'URI demandée est trop longue

418 : I am a teapot

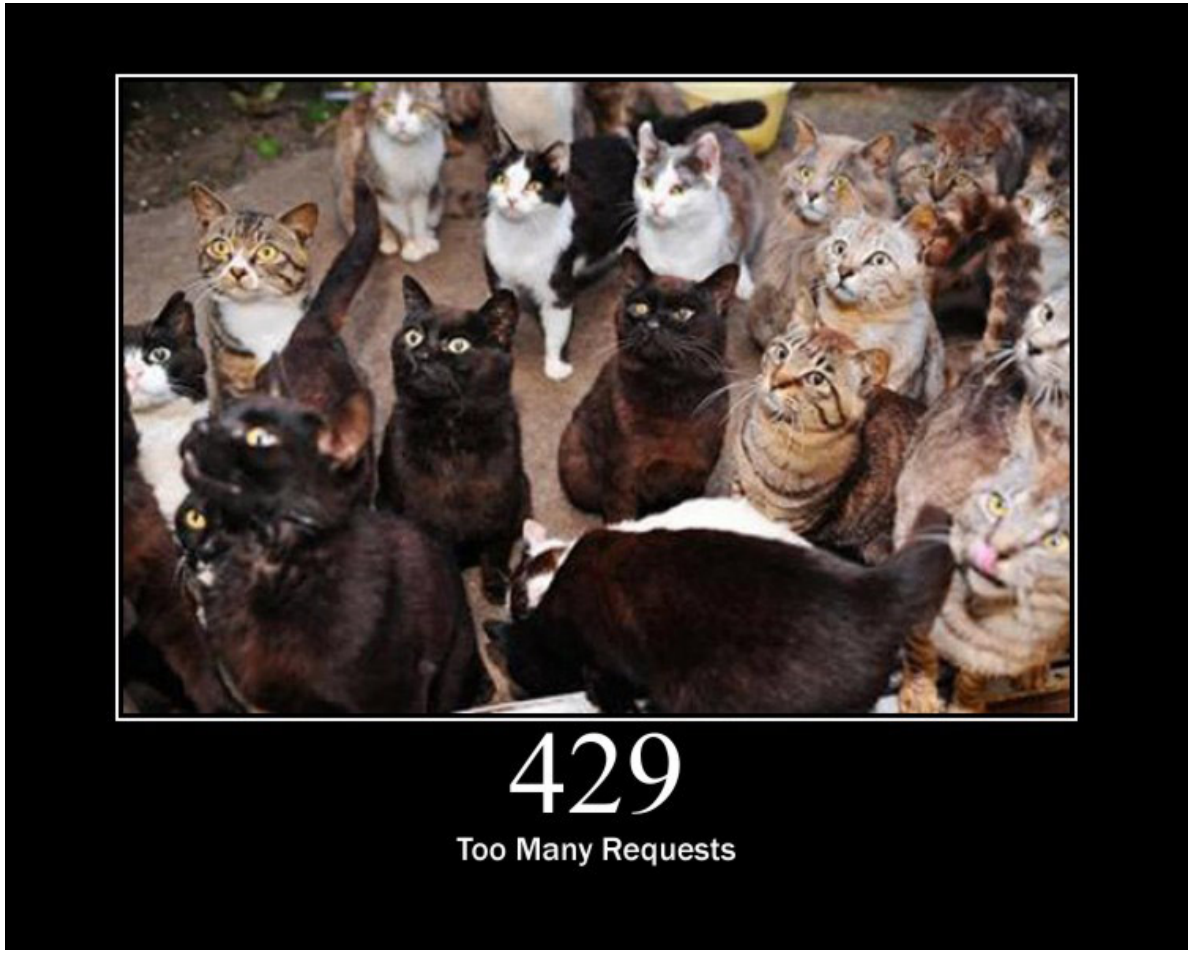
418 : je suis une théière (RFC 2324 définissant le Hyper Text Coffee Pot Control Protocol)

426 : Upgrade required

426 : le client utilise une version trop ancienne du protocole

429 : Too many requests

429 : le client a envoyé trop de requêtes en un temps donné

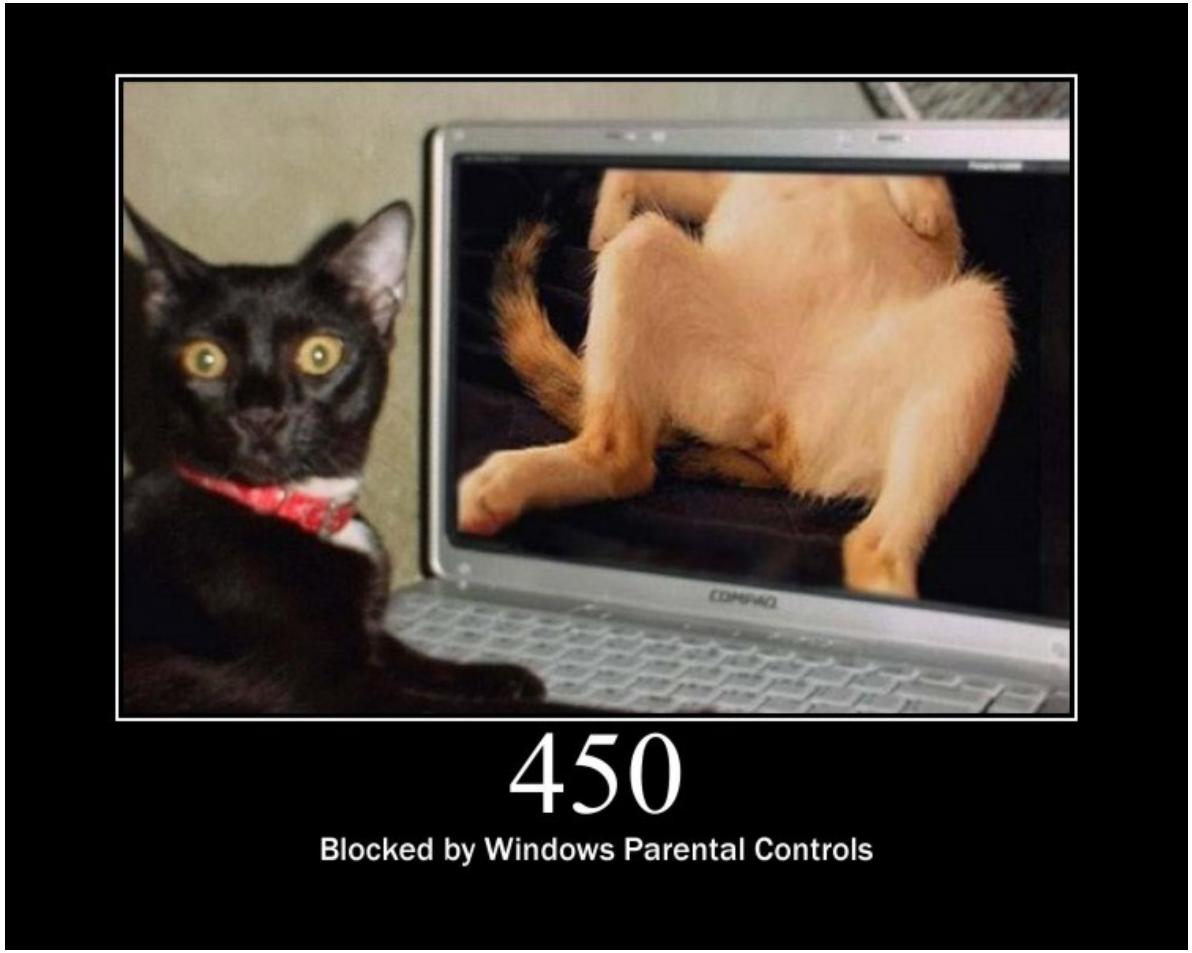


431 : Request Header Fields Too Large

431 : l'en-tête de la requête ou un de ses champs est trop grand

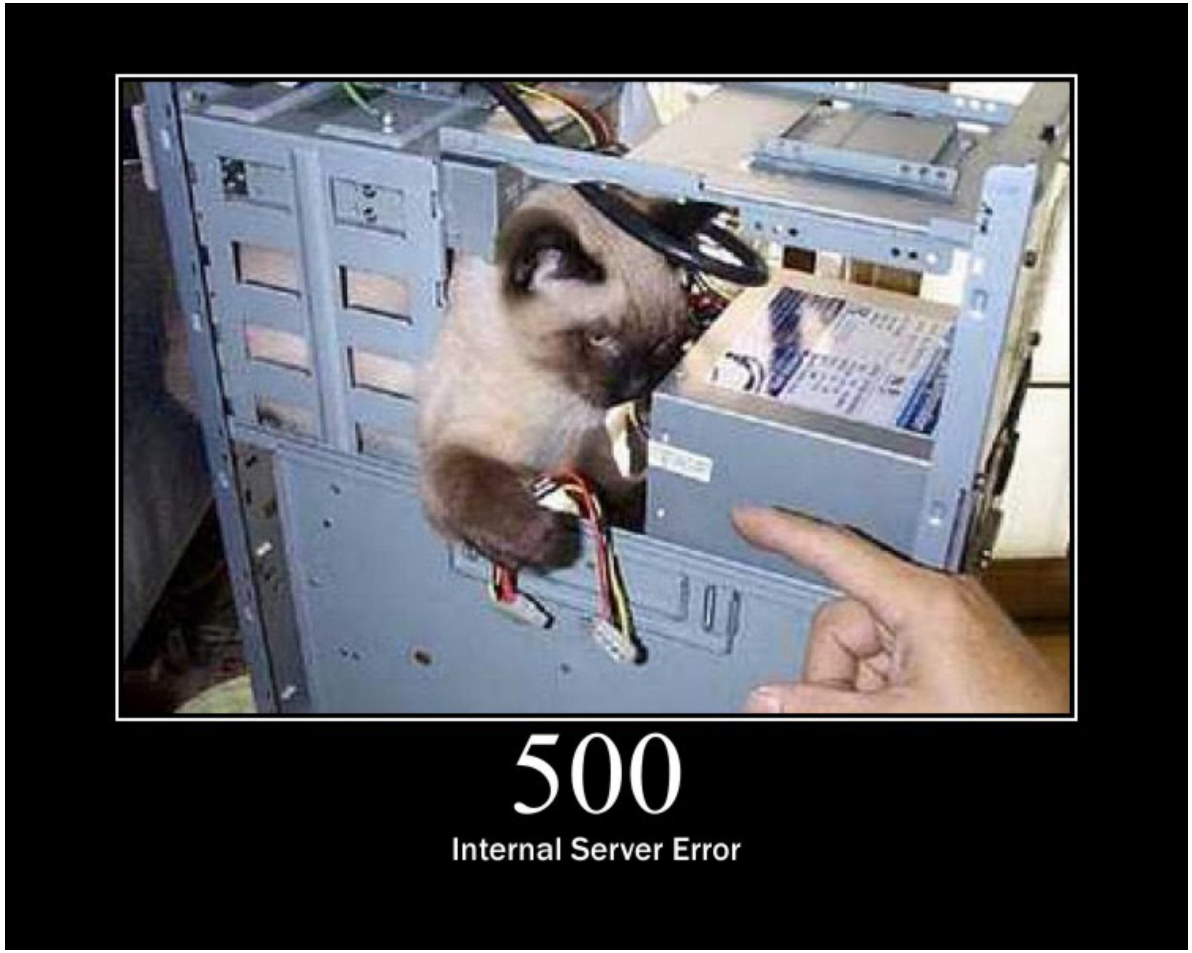
450 : Blocked by Windows Parental Controls

450 : extension du contrôle parental de Microsoft



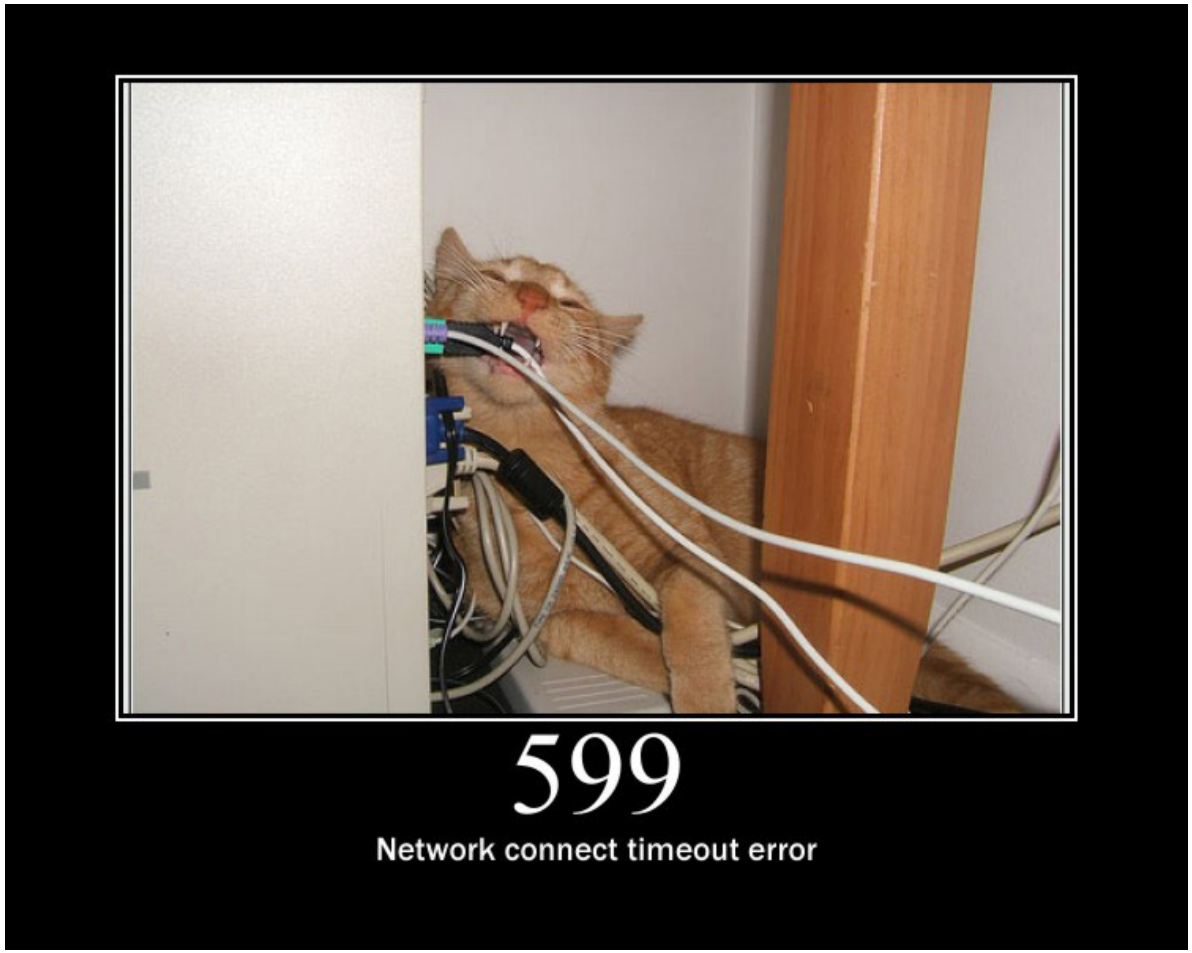
500 : Internal server error

500 : erreur côté serveur



599 : Network connect tomeout error

599 : timeout côté serveur (proxy)



Le type MIME

Multipurpose Internet Mail Extensions

Standard Internet

- RFC 2045, RFC 2046, RFC 2047, RFC 2048 et RFC 2077
- Transmettre n'importe quel contenu dans de l'ASCII

Origine

L'encodage des mails est en ASCII 7 bits

- Impossibilité de transmettre des langages basés sur l'alphabet Latin nécessitant des caractères spéciaux
- Impossibilité de transmettre des langages basés sur un autre alphabet
- Impossibilité de transmettre autre chose que du texte

Utiliser un type MIME permet d'encode autre chose dans le corps ou dans l'en-tête d'un mail

- Par extension, dans toute communication basée sur ASCII

MIME : exemple dans un mail

En-tête

On spécifie la version

- `MIME-Version: 1.0`

On annonce ce qui vient dans le corps

- `Content-Type: Multipart/related;`
  `charset="ISO-8859-1";`
  `type="multipart/alternative";`
  `boundary="------Boundary-00=_JXS9QL80000000000000"`

Séparation entre les éléments

Utilisation d'un séparateur quand le type change

- `-------Boundary-00=_JXS9LVC0000000000000-`

Chaque élément est transmis comme une partie du mail

MIME : exemple dans un mail

Transmission d'une image

- ```
  -------Boundary-00=_JXS9QL80000000000000
  Content-Type: image/gif;
  name="ATT1.gif"
  Content-Transfer-Encoding: base64
  Content-ID: <D0DEE211-0A46-4E4F-AEC1-23417EEED16A>
  ```

Texte

L'image est terminée, on repasse à du texte

- ```
  -------Boundary-00=_JXS9LVC0000000000000
  Content-Type: Text/Plain;
  charset="ISO-8859-1"
  Content-Transfer-Encoding: quoted-printable
  ```

HTML (pouvant être interprêté par le client mail)

- ```
  -------Boundary-00=_JXS9LVC0000000000000
  Content-Type: Text/HTML;
  charset="ISO-8859-1"
  Content-Transfer-Encoding: quoted-printable
  <HTML><HEAD>
  ```
  ```

Serveur Web

Fonctionnement

Le serveur Web écoute sur un port (généralement le port 80)

- Les clients se connectent sur ce port
  - un client demande une page
  - le serveur lui envoie cette page
  - le client se déconnecte

Exemples

- Open source : Apache HTTP Server (Apache), Tomcat, Jigsaw...
- Propriétaire : Microsoft IIS, Zeus...

Adressage des documents

Notion d'URL

- Universal Ressource Locator

L'URL contient :

- L'adresse IP ou symbolique (DNS) du domaine de la machine
- Le chemin vers le document sur ce serveur

Adressage des documents

Notion d'URL (Universal Ressource Locator)

Permet d'identifier

- Le serveur sur lequel se situe le document
- Le document lui-même

Composition d'une URL

L'URL contient :

- L'adresse IP ou symbolique (DNS) du domaine dont fait partie la machine
  - ex : univ-paris13.fr
- Le nom de la machine sur laquelle tourne le serveur
  - ex : www.iutv
- Le chemin relatif vers le document (par rapport au répertoires du serveur Web)
  - ex : /iutv/reseaux_telecommunications.php

L'URL de ce document est alors :

www.iutv.univ-paris13.fr/iutv/reseaux_telecommunications.php

Client Web

Fonctionnement

Envoi des requêtes aux serveurs

- Interprétation des URL
- Résolution DNS, localisation du serveur
- Mise en forme des requêtes HTTP

Interprétation des réponses

Affichage des documents

- Interprétation de l'HTML
- Exécution des scripts côté client
- Conformation aux standards vérifiée par les tests Acid

Si le document contient d'autres parties (images, etc)

- Émission de requêtes pour les obtenir

Exemples

FireFox, Fennec, Opera, Chrome, Safari, Konqueror, Internet Explorer...

Applications Web

Types d'applications

Applications côté serveur

- Le serveur fait le calcul
- Sa réponse dépend de ce qu'il a calculé
- Envoi de code HTML
- Exemple : PHP

Applications côté client

- Le client fait le calcul
- Exemple : Javascript, applet Java...

Comparatif

Confidentialité

- Applications côté serveur : le code exécuté est invisible du client

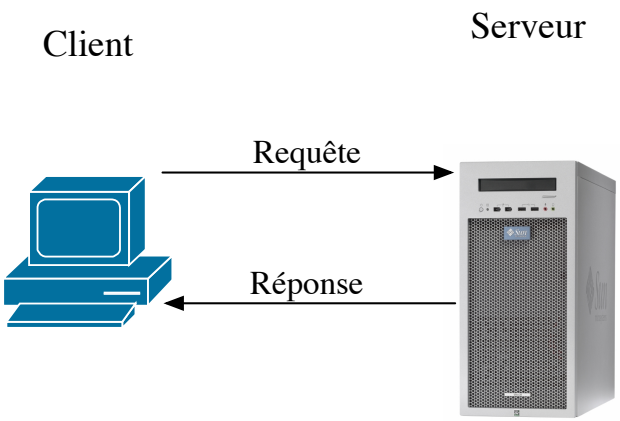
Charge

- Applications côté serveur : le client fait le travail, pas le serveuur

Illustation

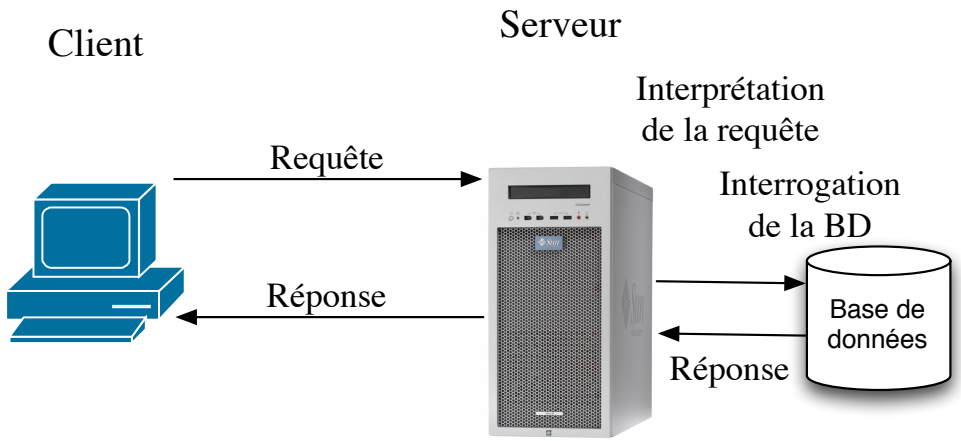
Requête simple

Exemple : affichage d'une page



Application serveur

Exemple : interrogation d'une base de données



De l'affichage de documents aux applications distribuées

Limites du modèle

Le client est toujours à l'initiative de l'échange

- Pas de push du serveur vers le client
- Pas de notifications d'évènements
  - Bricolage avec AJAX
  - Timer : toutes les X secondes, le client demande au serveur s'il y a quelque chose de nouveau (exemple : notifications Facebook)
- La connexion est fermée à la fin de l'échange
  - Le flux est coupé
  - HTML5 : WebSockets
  - Applets Java : connexion vers un servlet situé sur la machine serveur

Applications déployées

Soit centralisées

- tout se fait côté serveur : BD, blog, diffusion de contenu...

Soit entièrement côté client

- tout se fait côté client : animations / jeux Flash, applets Java...

Serveur Apache HTTP

Présentation

Serveur le plus répandu

- Environ 72,5% de parts de marché (décembre 2008)
- Deuxième : IIS, environ 18% (décembre 2008)

Bien intégré dans une combinaison open source

- LAMP : Linux Apache MySQL PHP

Mais *très* large compatibilité avec d'autres technologies

Modularité

Ajout de fonctionalités par modules

- Permet de proposer une très large gamme de fonctionalités
- Sans forcément les charger toutes (plus léger, failles de sécurité...)

Configuration d'Apache

Fichiers de configuration

Généralement dans `/etc/apache2/` (variable selon les versions et/ou la distribution Linux)

- httpd.conf et apache.conf

Modules

Configuration des modules

- mods-enabled/*.load
- mods-enabled/*.conf

Sites hébergés

Possibilité d'héberger plusieurs sites sur un serveur

- sites-enabled/
- sites-available/

Détermination : selon le port ou le nom d'hôte

`<VirtualHost *:80>`

Sécurité

Au niveau des communications

Possibilité de crypter les communications avec SSL ou TLS

- Permet d'éviter de faire passer des informations en clair (mots de passe, numéros de carte banquaire, de sécurité sociale...)
- Utilisation d'un certificat signé par une autorité de certification
- Port 443 par défaut
- Protocole HTTPs

Restrictions d'accès

Utilisation d'un fichier `.htaccess`

- Définition de règles par répertoire et pour les sous-répertoires
- Global (tout le serveur ou tout le site) ou local (un répertoire donné et ses sous-répertoires)
- Définit les restrictions : par adresse IP du client, par type de ficheir, authentification du client...

Traces et logs

Traces d'accès au serveur

Disponibles dans `/var/log/apache2/`

- `access.log` : toutes les requêtes arrivées au serveur

```
127.0.0.1 - - [22/Jan/2011:17:51:17 +0100] "GET
/server-status?auto HTTP/1.1" 200 633 "-" "libwww-perl/5.836"
```

- `errors.log` : toutes les erreurs

```
[Sun Jan 16 07:43:37 2011] [notice] Apache/2.2.16 (Debian)
PHP/5.3.2-2 with Suhosin-Patch configured - resuming normal
operations
```

Autres traces

- Possibilité de mettre en place une trace par Virtual Host
- PID du serveur Apache : `/var/run/apache2.pid`

Fichiers sur le serveur

Site principal

Localisation définie dans la configuration du serveur ou du virtual host

```
/etc/apache2/sites-available/default:  DocumentRoot /var/www
```

Comptes utilisateurs

Localisation dans un répertoire particulier du répertoire utilisateur, généralement `~/public_html` ou `~/WWW` (défini dans la configuration du serveur ou du virtual host)

```
/etc/apache2/mods-available/userdir.conf:  UserDir public_html
```

Droits d'accès

Le serveur tourne généralement sous un utilisateur particulier : www-data, apache, root (peu sécurisé donc peu fréquent)...

- Les fichiers doivent être lisibles par cet utilisateur
- Les répertoires doivent être traversables par cet utilisateur
- Les scripts doivent être exécutables par cet utilisateur
- etc