Développement et administration des services réseaux : Base des Services Réseaux

– M2106 –

Camille Coti
camille.coti@lipn.univ-paris13.fr

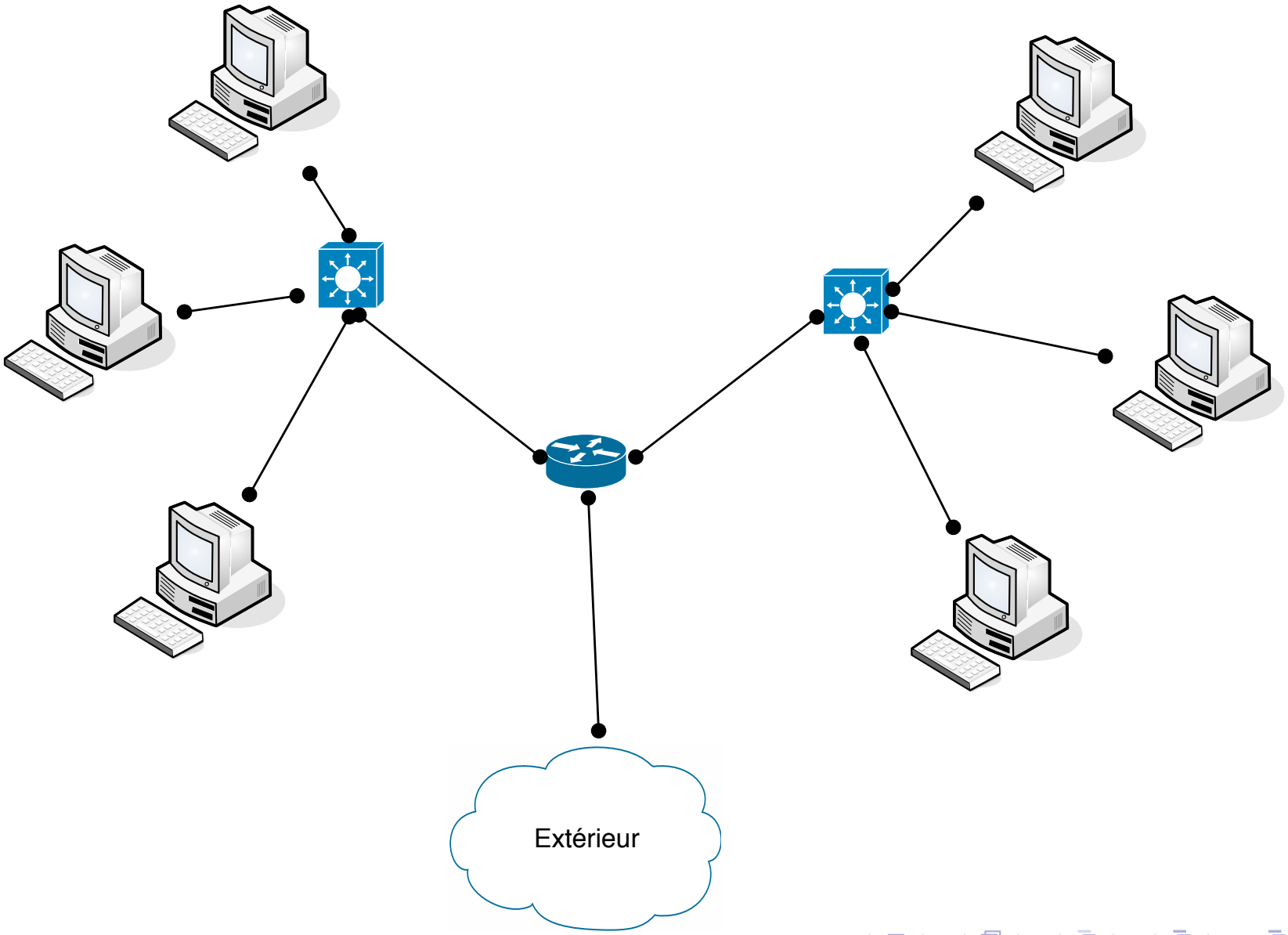
Département R&T, IUT de Villetaneuse, Université de Paris XIII

Plan du cours

Plan du cours

Éléments de structuration d'un réseau

Réseau local :



Éléments de structuration d'un réseau

Réseau local :

- *"réseau informatique tel que les terminaux qui y participent s'envoient des trames au niveau de la couche de liaison sans utiliser d'accès à internet"* (Wikipedia)
- En clair :
  - Ensemble de terminaux (serveurs, stations de travail, imprimantes, etc)...
  - ... interconnectés par une couche physique de réseau à courte portée (Ethernet, WiFi, X25, etc)...
  - ... communiquant les uns avec les autres...
  - ... et, éventuellement, avec un réseau extérieur (MAN, WAN).

Éléments de structuration d'un réseau

Réseau local :

- *"réseau informatique tel que les terminaux qui y participent s'envoient des trames au niveau de la couche de liaison sans utiliser d'accès à internet"* (Wikipedia)
- En clair :
  - Ensemble de terminaux (serveurs, stations de travail, imprimantes, etc)...
  - ... interconnectés par une couche physique de réseau à courte portée (Ethernet, WiFi, X25, etc)...
  - ... communiquant les uns avec les autres...
  - ... et, éventuellement, avec un réseau extérieur (MAN, WAN).

Question : comment permettre ces communications ?

Éléments de structuration d'un réseau

Réseau local :

- *"réseau informatique tel que les terminaux qui y participent s'envoient des trames au niveau de la couche de liaison sans utiliser d'accès à internet"* (Wikipedia)
- En clair :
  - Ensemble de terminaux (serveurs, stations de travail, imprimantes, etc)...
  - ... interconnectés par une couche physique de réseau à courte portée (Ethernet, WiFi, X25, etc)...
  - ... communiquant les uns avec les autres...
  - ... et, éventuellement, avec un réseau extérieur (MAN, WAN).

Question : comment permettre ces communications ?

- Au niveau physique : assurer la connectivité des couches physiques.
- Au niveau réseau : un certain nombre de **services réseaux** sont nécessaires.

Communications sur le réseau

Premère chose à faire : **permettre aux machines de communiquer entre elles**

- Techniquement : leur attribuer à chacune une adresse IP, un masque et une passerelle par défaut
- Comment ?

Communications sur le réseau

Premère chose à faire : **permettre aux machines de communiquer entre elles**

- Techniquement : leur attribuer à chacune une adresse IP, un masque et une passerelle par défaut
- Comment ?
  - Manuellement : configuration de chaque machine une par une
    - Fastidieux, source d'erreur (côté administrateur comme côté utilisateur), gestion du pool d'adresses compliquée

Communications sur le réseau

Premère chose à faire : **permettre aux machines de communiquer entre elles**

- Techniquement : leur attribuer à chacune une adresse IP, un masque et une passerelle par défaut
- Comment ?
  - Manuellement : configuration de chaque machine une par une
    - Fastidieux, source d'erreur (côté administrateur comme côté utilisateur), gestion du pool d'adresses compliquée
  - Automatiquement : utilisation d'un serveur **DHCP (Dynamic Host Configuration Protocol)**
    - Simple pour l'administrateur, intuitif pour l'utilisateur

Communications sur le réseau

Premère chose à faire : **permettre aux machines de communiquer entre elles**

- Techniquement : leur attribuer à chacune une adresse IP, un masque et une passerelle par défaut
- Comment ?
  - Manuellement : configuration de chaque machine une par une
    - Fastidieux, source d'erreur (côté administrateur comme côté utilisateur), gestion du pool d'adresses compliquée
  - Automatiquement : utilisation d'un serveur **DHCP (Dynamic Host Configuration Protocol)**
    - Simple pour l'administrateur, intuitif pour l'utilisateur

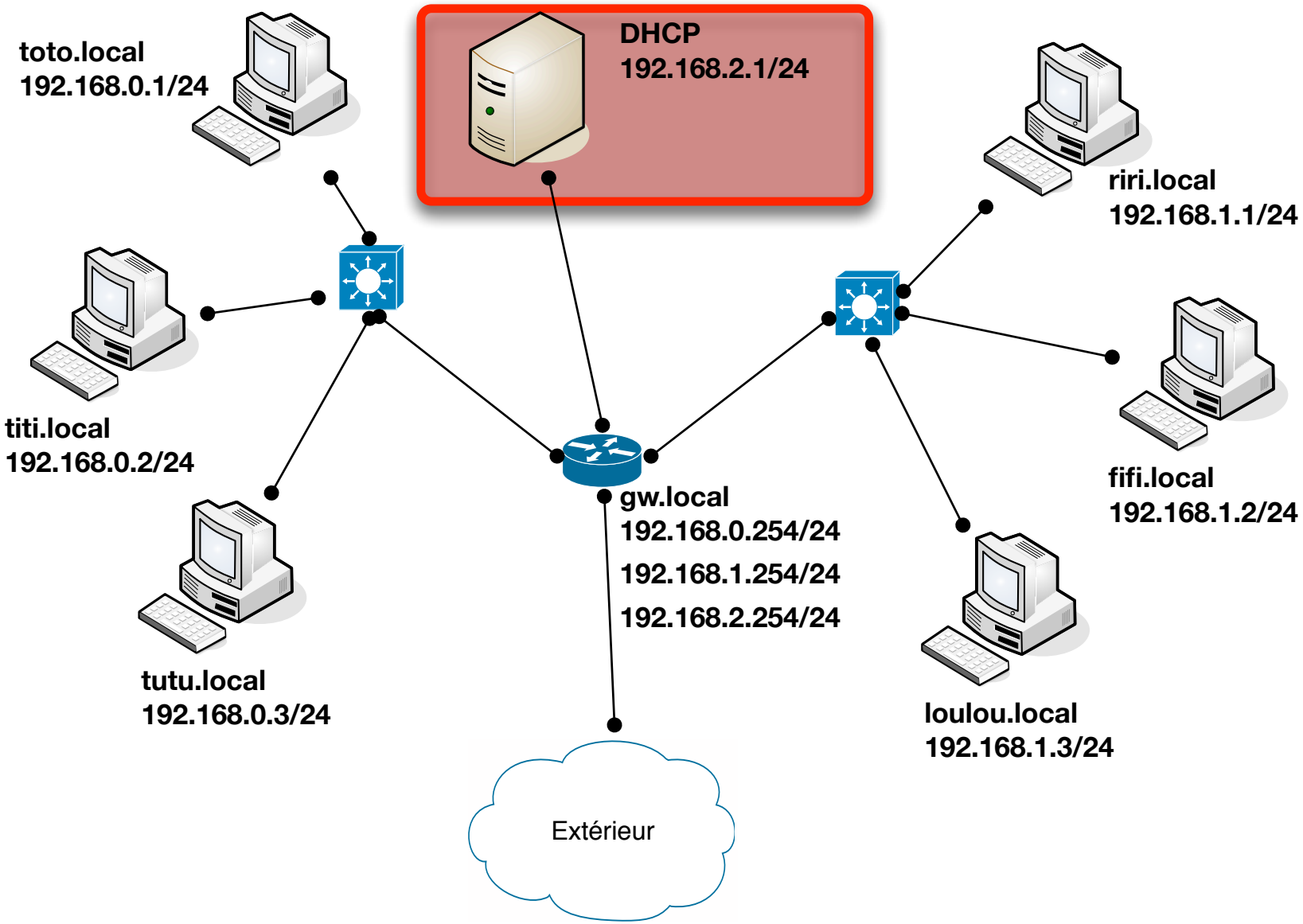
Une fois que chaque terminal du réseau dispose de

- une adresse IP
- un masque de sous-réseau
- une passerelle par défaut

Les terminaux peuvent communiquer entre eux et, si possible, sortir vers l'extérieur du réseau.

Éléments de structuration d'un réseau

Réseau local :



Résolution des noms

Problème : permettre l' **association entre une adresse IP et un nom symbolique** (alphanumérique)

- Adresse IP : compréhensible par les machines
- Adresse symbolique : compréhensible par les humains

Résolution des noms

Problème : permettre l' **association entre une adresse IP et un nom symbolique** (alphanumérique)

- Adresse IP : compréhensible par les machines
- Adresse symbolique : compréhensible par les humains

Deux possibilités :

- Manuelle : liste exhaustive des associations dans `/etc/hosts`
- Automatique : interrogation d'un **serveur qui effectue cette conversion**

Résolution des noms

Problème : permettre l' **association entre une adresse IP et un nom symbolique** (alphanumérique)

- Adresse IP : compréhensible par les machines
- Adresse symbolique : compréhensible par les humains

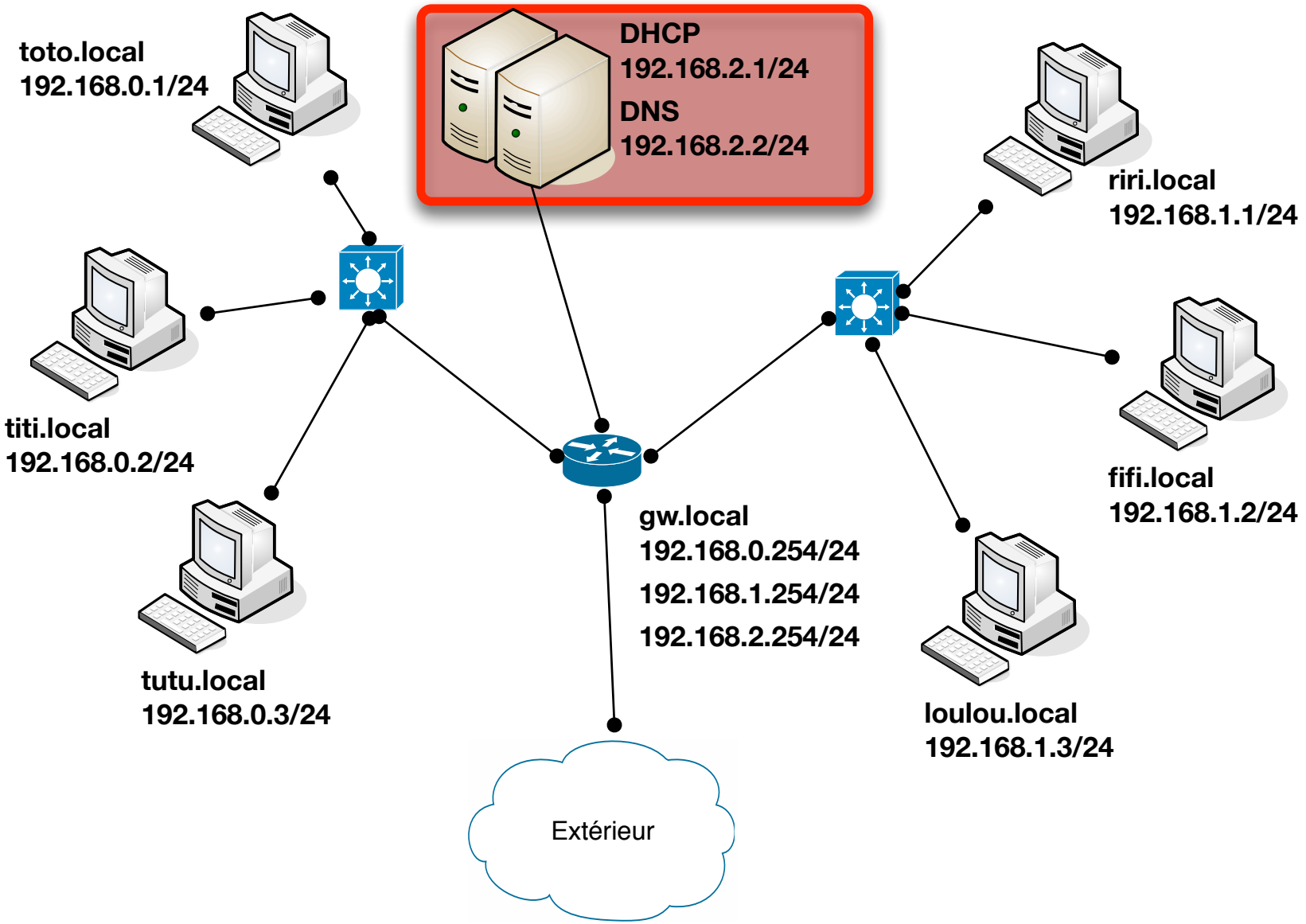
Deux possibilités :

- Manuelle : liste exhaustive des associations dans `/etc/hosts`
- Automatique : interrogation d'un **serveur qui effectue cette conversion**

Service concerné : **DNS (Domain Name System)**

Éléments de structuration d'un réseau

Réseau local :



Gestion des utilisateurs

Problème : permettre aux utilisateurs de **s'authentifier sur les machines**

- Un seul login/mot de passe sur tous les postes
- Informations sur le compte : nom réel, bureau, dernière connexion, adresse email...
- Centralisé : prise en compte des modifications (création de compte, changement de mot de passe, etc) partout

Comment ?

- Manuel : propagation d'un `/etc/password`
  - Pas instantané, problèmes de cohérence, compliqué à grande échelle
- Automatique : interrogation d'un **serveur d'annuaire**
  - Centralisé sur le serveur, simple et sécurisé (dans une certaine mesure)

Gestion des utilisateurs

Problème : permettre aux utilisateurs de **s'authentifier sur les machines**

- Un seul login/mot de passe sur tous les postes
- Informations sur le compte : nom réel, bureau, dernière connexion, adresse email...
- Centralisé : prise en compte des modifications (création de compte, changement de mot de passe, etc) partout

Comment ?

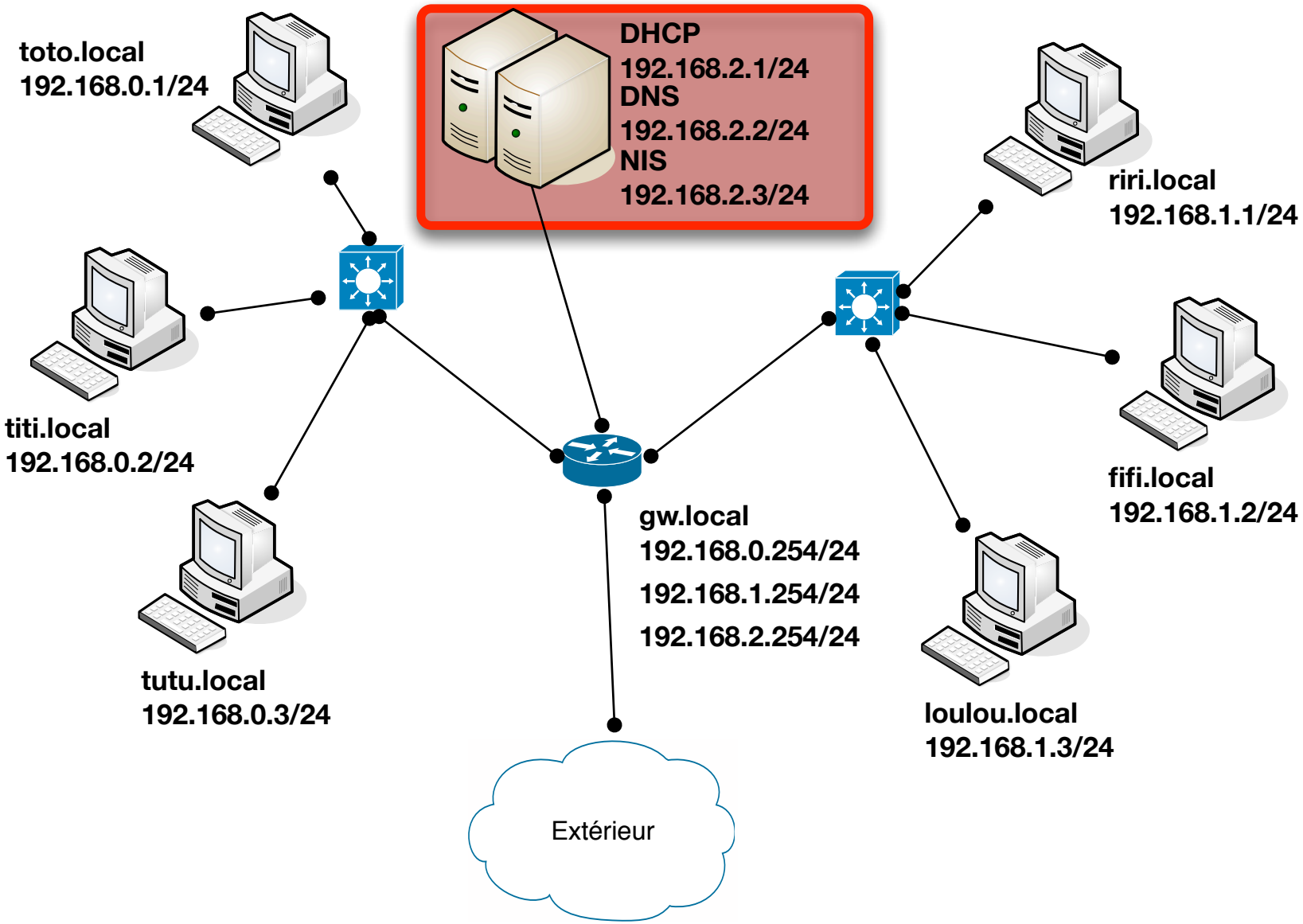
- Manuel : propagation d'un `/etc/password`
  - Pas instantané, problèmes de cohérence, compliqué à grande échelle
- Automatique : interrogation d'un **serveur d'annuaire**
  - Centralisé sur le serveur, simple et sécurisé (dans une certaine mesure)

Service concerné : **NIS (Network Information Service)** (sous Unix)

- Service d'annuaire (à l'origine : *Yellow Pages*)

Éléments de structuration d'un réseau

Réseau local :



Système de fichiers en réseau

Accès aux fichiers :

- **Locaux** : sur un stockage local (disque dur...), accessibles depuis une machine seulement
- **Sur le réseau** : accessibles depuis n'importe quelle machine du réseau

Système de fichiers en réseau

Accès aux fichiers :

- **Locaux** : sur un stockage local (disque dur...), accessibles depuis une machine seulement
- **Sur le réseau** : accessibles depuis n'importe quelle machine du réseau

Problème : permettre aux utilisateurs d' **accéder à leurs fichiers depuis n'importe quelle machine**

- Manuellement : copie des fichiers sur toutes les machines (!!!), synchronisation des répertoires utilisateurs `$HOME` (`rsync` par exemple)
- Utilisation d'un service de fichiers en réseau : les fichiers sont sur le serveur, tous les terminaux peuvent y accéder

Système de fichiers en réseau

Accès aux fichiers :

- **Locaux** : sur un stockage local (disque dur...), accessibles depuis une machine seulement
- **Sur le réseau** : accessibles depuis n'importe quelle machine du réseau

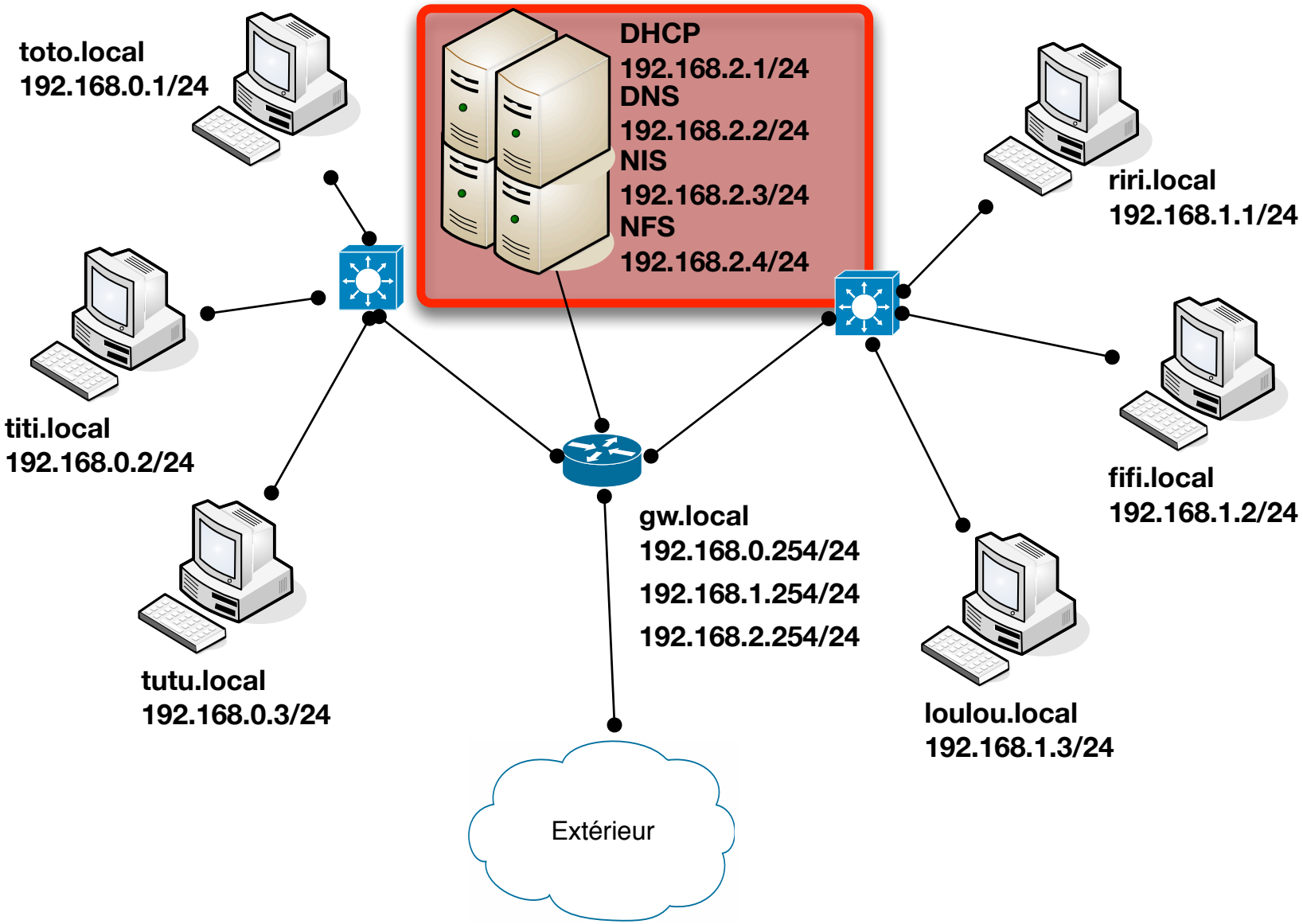
Problème : permettre aux utilisateurs d' **accéder à leurs fichiers depuis n'importe quelle machine**

- Manuellement : copie des fichiers sur toutes les machines (!!!), synchronisation des répertoires utilisateurs `$HOME` (`rsync` par exemple)
- Utilisation d'un service de fichiers en réseau : les fichiers sont sur le serveur, tous les terminaux peuvent y accéder

Service concerné : **NFS (Network File System)**
Sous Windows uniquement : Samba.

Éléments de structuration d'un réseau

Réseau local :



Plan du cours

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.

Utilisation

Machine non configurée

```
# /sbin/ifconfig  eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:e5:d6:d2
          adr inet6: fe80::a00:27ff:fee5:d6d2/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:0 (0.0 KiB)  TX bytes:500 (0.5 KiB)
```

Requête DHCP

Machine configurée

Utilisation

Machine non configurée

Requête DHCP

```
coti@gauss:~$ sudo dhclient eth0
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 10.0.2.2
DHCPREQUEST on eth0 to 255.255.255.255 port 67
DHCPACK from 10.0.2.2
bound to 10.0.2.15 -- renewal in 39525 seconds.
```

Machine configurée

Utilisation

Machine non configurée

Requête DHCP

Machine configurée

```
coti@gauss:~$ /sbin/ifconfig  eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:e5:d6:d2
          inet adr:10.0.2.15  Bcast:10.0.2.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fee5:d6d2/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:9 errors:0 dropped:0 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:2080 (2.0 KiB)  TX bytes:13543 (13.2 KiB)
```

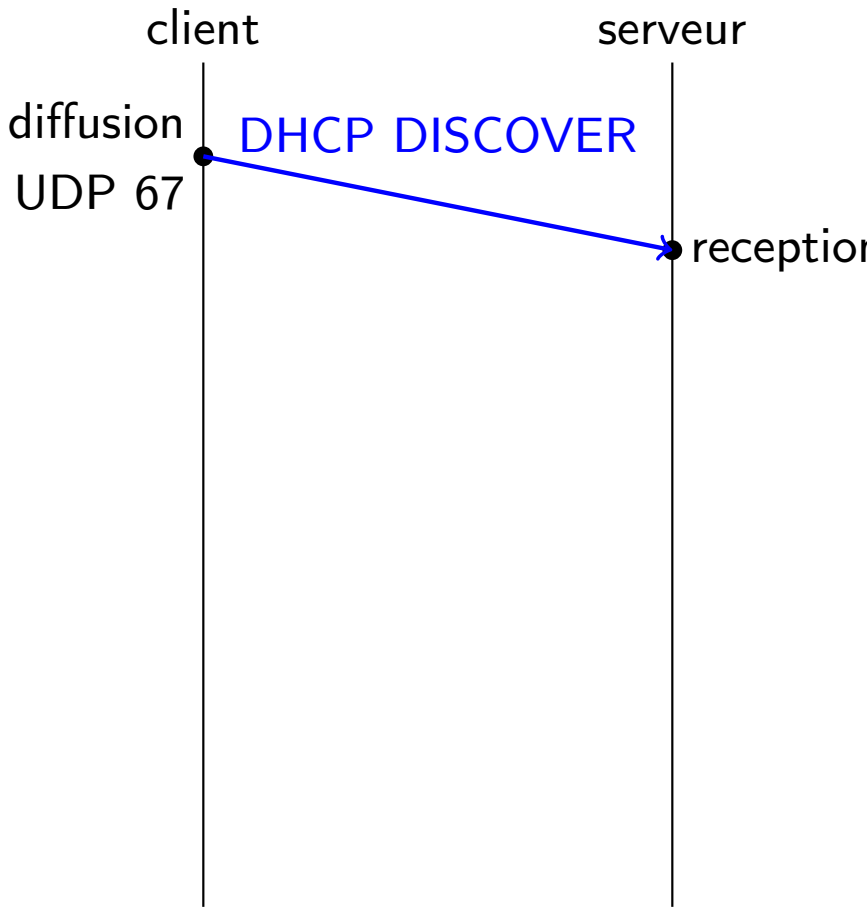
Protocole

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

Protocole

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

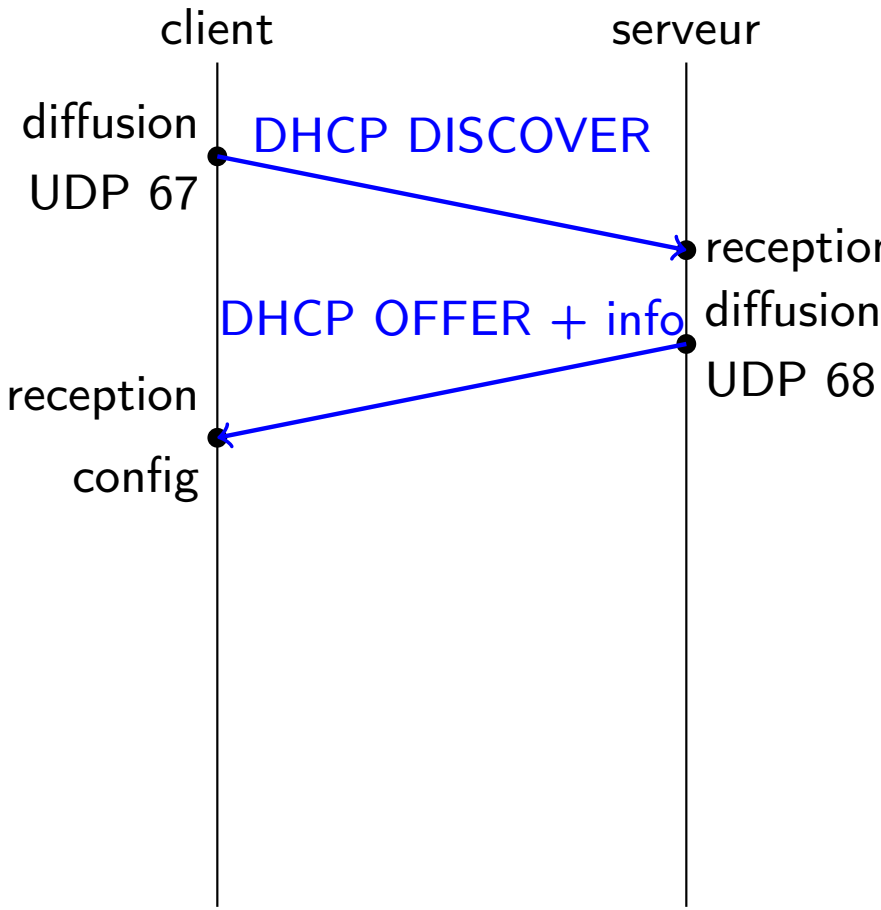
- Le client diffuse un datagramme DHCP DISCOVER pour le port 67



Protocole

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

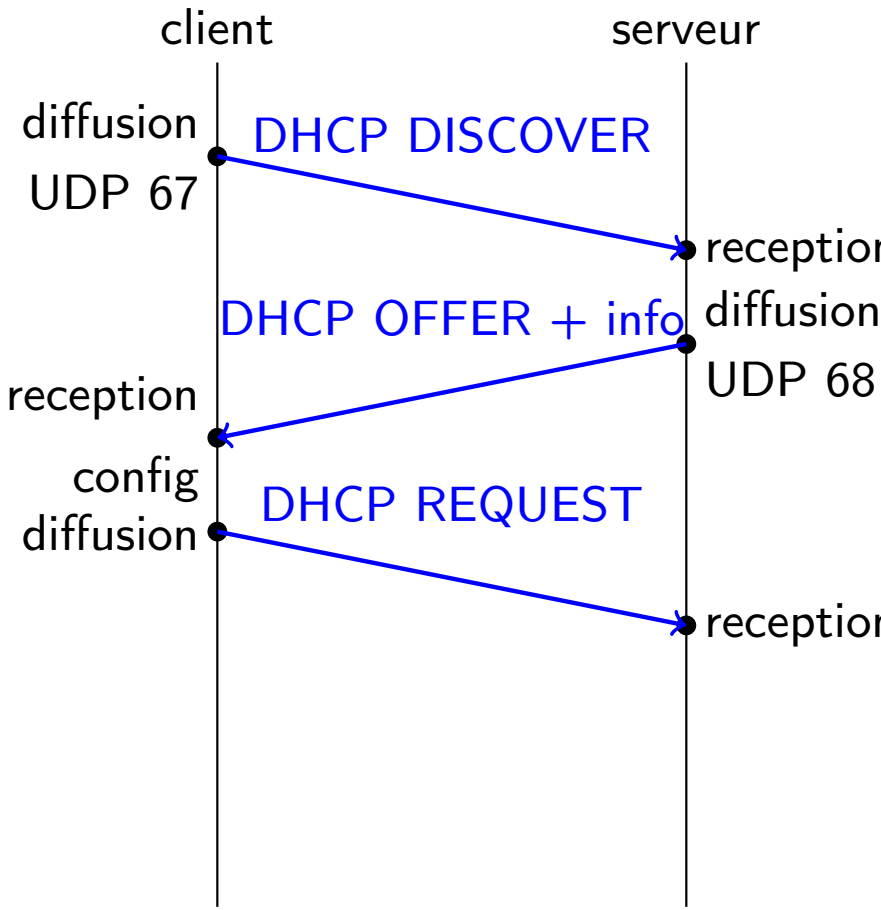
- Le client diffuse un datagramme DHCP DISCOVER pour le port 67
- Les serveurs reçoivent le datagramme et diffusent un datagramme DHCP OFFER pour le port 68 : proposition de configuration



Protocole

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

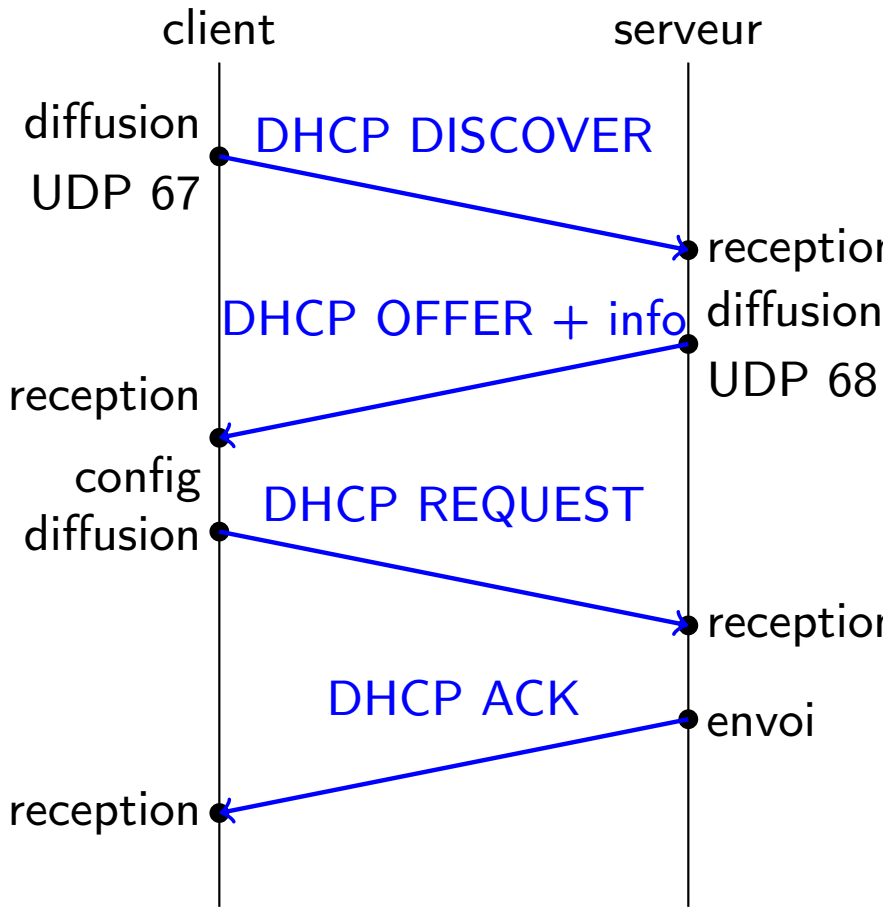
- Le client diffuse un datagramme DHCP DISCOVER pour le port 67
- Les serveurs reçoivent le datagramme et diffusent un datagramme DHCP OFFER pour le port 68 : proposition de configuration
- Le client choisit un serveur parmi les offres et diffuse un datagramme DHCP REQUEST contenant l'IP du serveur retenu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



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.



Plan du cours

Résolution des noms de domaines

Adresse d'une machine :

- Compréhensible par un humain : **alpha-numérique** (symbolique) : `www.google.com`
- Compréhensible par une machine : **adresse IP** : `173.194.45.82` ou `2a00:1450:4007:807::1010`

But de la **résolution DNS** : passer de l'un à l'autre

- L'humain utilise l'adresse symbolique
- Interrogation du serveur DNS : à quelle adresse IP correspond cette adresse symbolique
- L'application se connecte en utilisant l'adresse IP

Comment interroger la résolution ?

Résolution des noms de domaine : configuré dans le fichier `/etc/nsswitch.conf`

```
hosts:          files dns
```

Ici :

- On regarde d'abord dans les **fichiers locaux**
- Si on n'a pas trouvé, on interroge le **serveur DNS**

Résolution statique locale

Rappel : fichier `/etc/hosts`

```
127.0.0.1          localhost
255.255.255.255    broadcasthost
::1                localhost
fe80::1%lo0        localhost

192.168.1.1        riri.local        riri
192.168.1.2        fifi.local        fifi
192.168.1.3        loulou.local      loulou
```

On y trouve :

- Boucle locale (obligatoire)
- Définitions statiques

Résolution statique locale

Rappel : fichier `/etc/hosts`

```
127.0.0.1          localhost
255.255.255.255    broadcasthost
::1                localhost
fe80::1%lo0        localhost

192.168.1.1        riri.local        riri
192.168.1.2        fifi.local        fifi
192.168.1.3        loulou.local      loulou
```

On y trouve :

- Boucle locale (obligatoire)
- Définitions statiques

```
coti@gauss:~$ ping riri
PING riri (192.168.1.1) 56(84) bytes of data.
```

Serveur DNS

Interrogation d'un **serveur distant**

- Le serveur connait les correspondances
- Le client demande : "à quelle adresse IP correspond cette adresse symbolique ?"
- Le serveur répond

Deux algorithmes de résolution :

- **Itératif** : si le serveur ne connait pas la réponse, il renvoie l'adresse d'un autre serveur
- **Récursif** : si le serveur ne connait pas la réponse, il demande lui-même à un autre serveur

Interrogation d'un serveur DNS

Commandes Unix :

- `nslookup` : obsolète

```
coti@thorim:~$ nslookup www.google.com
Server:         212.27.40.240
Address:        212.27.40.240#53

Non-authoritative answer:
Name:   www.google.com
Address: 173.194.40.115
Name:   www.google.com
Address: 173.194.40.112
Name:   www.google.com
Address: 173.194.40.113
Name:   www.google.com
Address: 173.194.40.116
Name:   www.google.com
Address: 173.194.40.114
```

Interrogation d'un serveur DNS

Commandes Unix :

- nslookup : obsolète
- dig

```
coti@thorim:~$ dig www.google.com

; <<>> DiG 9.6-ESV-R4-P3 <<>> www.google.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30365
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.com.                        IN      A

;; ANSWER SECTION:
www.google.com.         326     IN      A       173.194.45.83
www.google.com.         326     IN      A       173.194.45.81
www.google.com.         326     IN      A       173.194.45.82
www.google.com.         326     IN      A       173.194.45.84
www.google.com.         326     IN      A       173.194.45.80

;; Query time: 22 msec
;; SERVER: 212.27.40.240#53(212.27.40.240)
;; WHEN: Sun Dec 29 22:46:33 2013
;; MSG SIZE  rcvd: 112
```

Interrogation d'un serveur DNS

Commandes Unix :

- `nslookup` : obsolète
- `dig`
- `host`

```
coti@thorim:~$ host www.google.com
www.google.com has address 173.194.40.180
www.google.com has address 173.194.40.177
www.google.com has address 173.194.40.176
www.google.com has address 173.194.40.178
www.google.com has address 173.194.40.179
www.google.com has IPv6 address 2a00:1450:4007:806::1010
```

Interrogation d'un serveur DNS

Commandes Unix :

- `nslookup` : obsolète
- `dig`
- `host`

En C :

- `gethostbyname()`, `getaddrinfo()`

Interrogation d'un serveur DNS

Commandes Unix :

- `nslookup` : obsolète
- `dig`
- `host`

En C :

- `gethostbyname()`, `getaddrinfo()`

Définition des serveurs interrogés par le client :

- Fichier `/etc/resolv.conf`

```
coti@thorim:~$ cat /etc/resolv.conf
nameserver 212.27.40.240
nameserver 212.27.40.241
```

Mise en place d'un serveur DNS

Quelques implémentations disponibles :

- **BIND (named)** : sous Unix, licence BSD, très utilisé (le plus utilisé)
- **djbdns** : autre implémentation Libre, accent mis sur la sécurité
- **Microsoft DNS** : fait partie de Windows Server
- **Cisco Network Registrar** : propriétaire
- ...

Mise en place d'un serveur DNS

Quelques implémentations disponibles :

- **BIND (named)** : sous Unix, licence BSD, très utilisé (le plus utilisé)
- **djbdns** : autre implémentation Libre, accent mis sur la sécurité
- **Microsoft DNS** : fait partie de Windows Server
- **Cisco Network Registrar** : propriétaire
- ...

En TP : **BIND**

- Correspondances adresses IP $\leftrightarrow$ noms symboliques : dans des fichiers texte
- Définition d'un serveur de plus haut niveau

Types de serveurs DNS

Serveur DNS primaire (ou maître, ou autorité)

- Serveur faisant autorité sur la zone qu'il sert
- Répond aux requêtes si il connait la correspondance
- Effectue l'action demandée/disponible si il ne connait pas (récursif/itératif)

Serveur DNS secondaire (ou esclave)

- Si le serveur primaire ne répond pas, on demande au secondaire

Serveur DNS cache

- Serveur qui sert une zone mais n'y a pas autorité
- Exemple : serveur local sur le réseau

Plan du cours

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

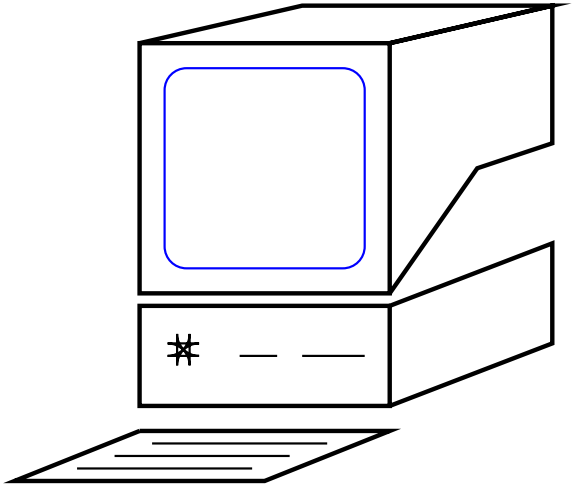
- Notion de domaine NIS
- Centralisation des informations
- Plusieurs serveurs : maître-esclave ou coopération entre serveurs

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

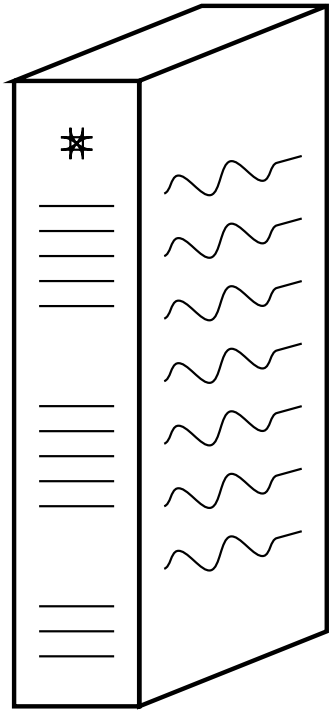
- Introduit par Sun en 1985
- Pas un standard, mais très largement utilisé

Exemple de fonctionnement

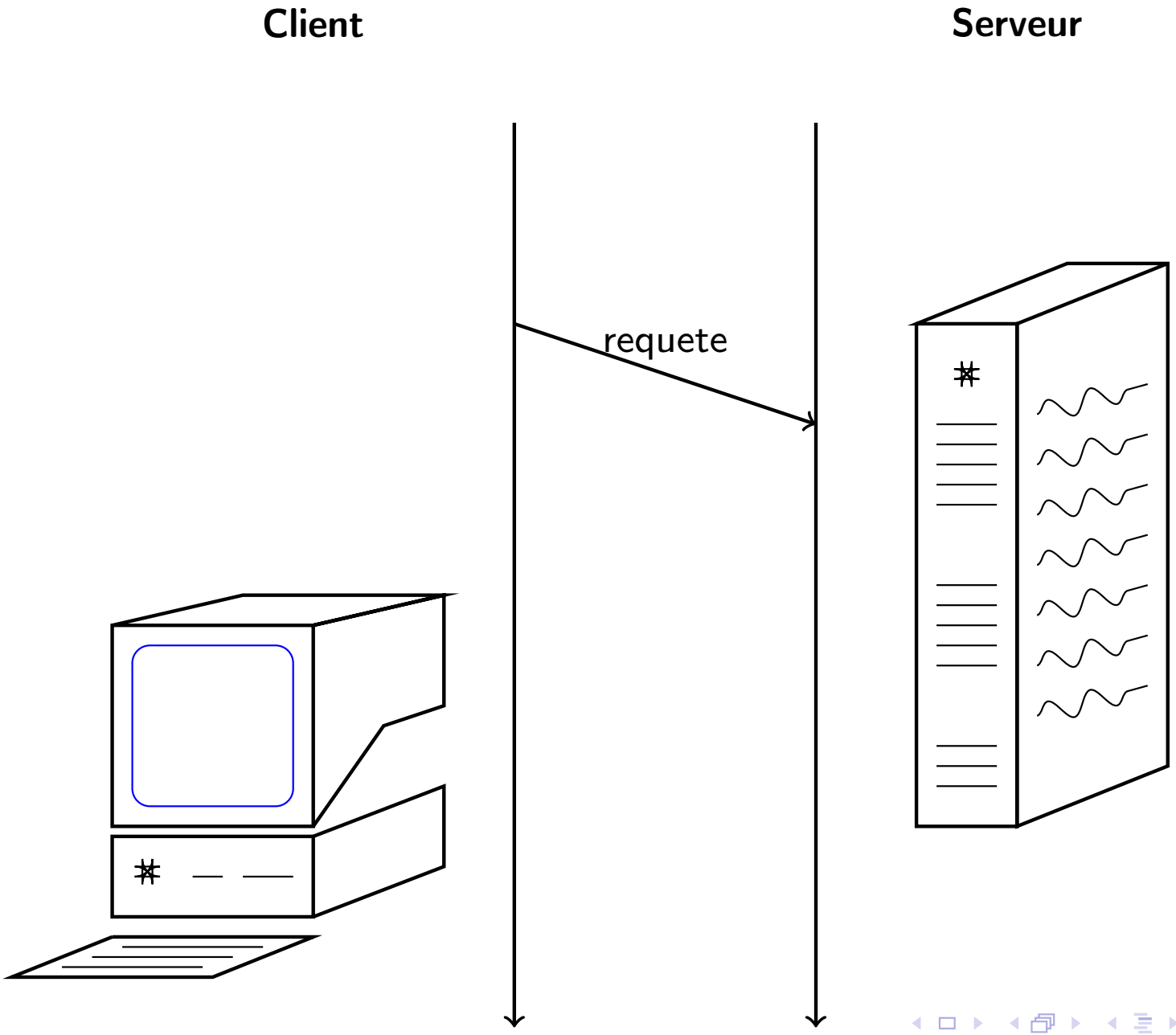
Client



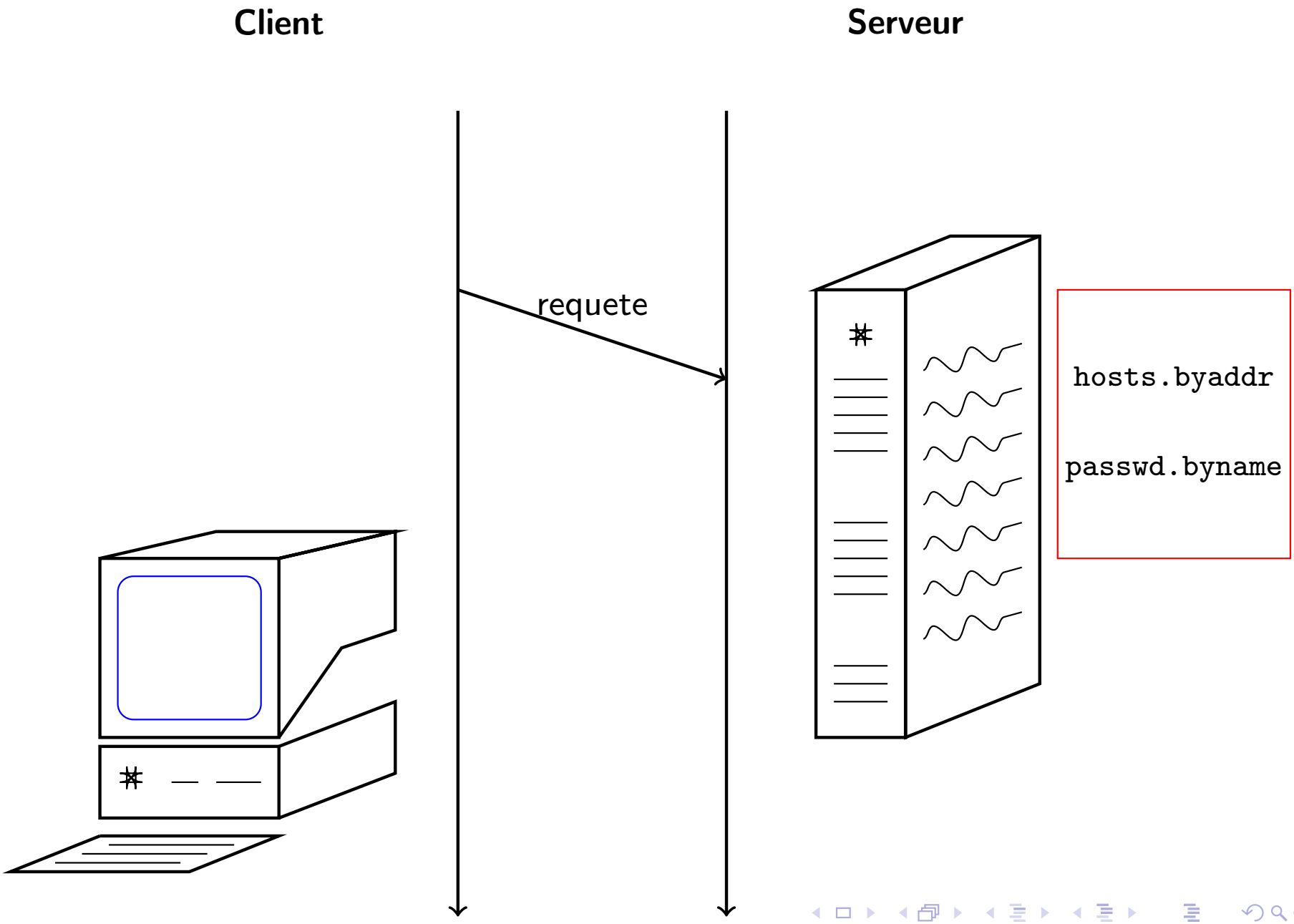
Serveur



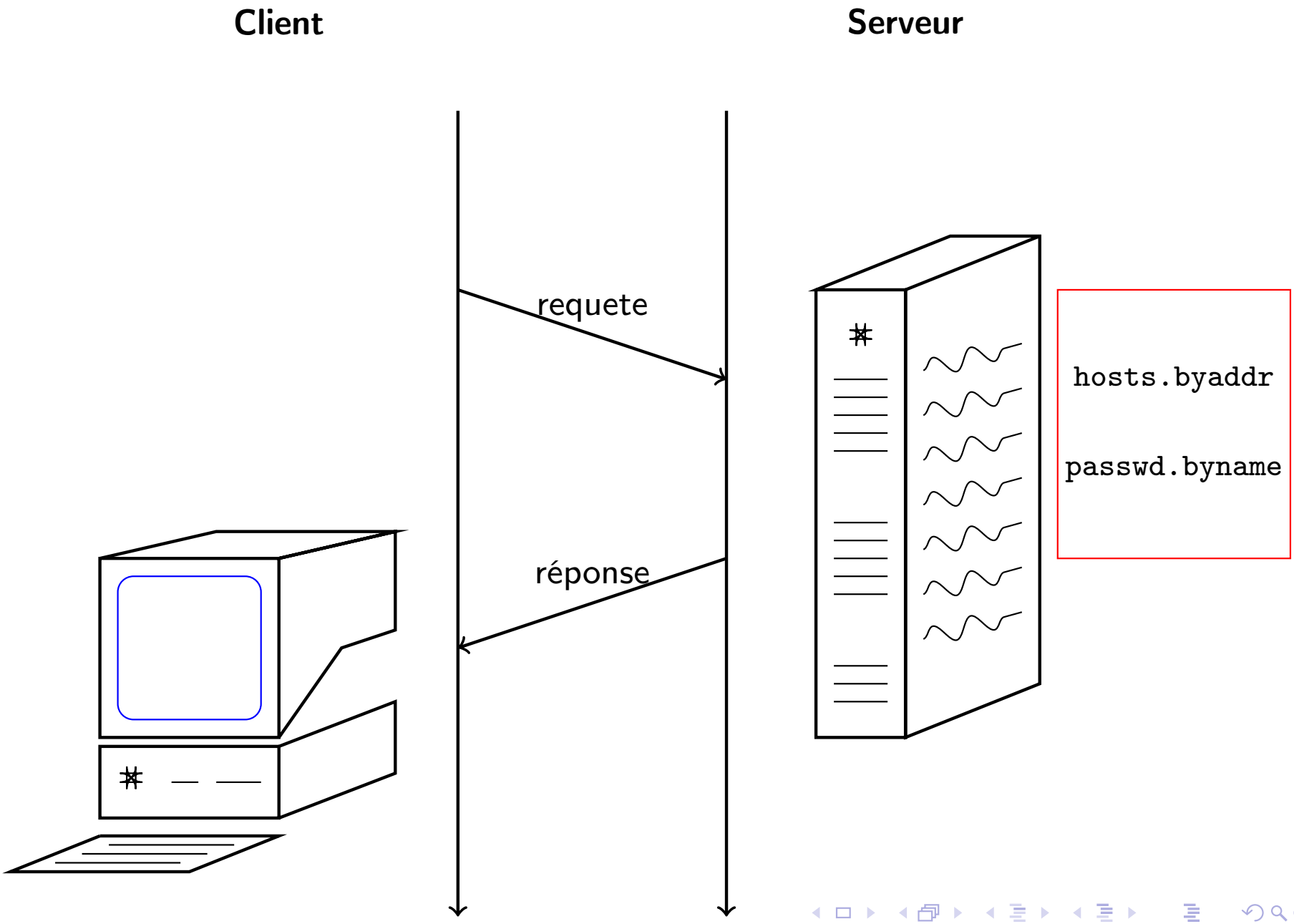
Exemple de fonctionnement



Exemple de fonctionnement



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
```

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 | grep coti
coti:8fVPKGXHmOy9Y:3456:100:Camille Coti:/users/coti:/bin/bash
```

Plan du cours

Système de fichiers en réseau : NFS

Principes du **système de fichiers en réseau** :

- Fichiers stockés **sur un serveur**
- Les **clients** accèdent à ces fichiers
- Le serveur **exporte un répertoire**

Système de fichiers en réseau : NFS

Principes du **système de fichiers en réseau** :

- Fichiers stockés **sur un serveur**
- Les **clients** accèdent à ces fichiers
- Le serveur **exporte un répertoire**

Sous forme d'un **montage** dans l'arborescence de fichiers locale

- **Transparent** pour l'utilisateur
- Apparait comme un répertoire local

Système de fichiers en réseau : NFS

Principes du **système de fichiers en réseau** :

- Fichiers stockés **sur un serveur**
- Les **clients** accèdent à ces fichiers
- Le serveur **exporte un répertoire**

Sous forme d'un **montage** dans l'arborescence de fichiers locale

- **Transparent** pour l'utilisateur
- Apparait comme un répertoire local

Deux principaux protocoles :

- **NFS** : Network File System (Unix)
- **SMB** : Server Message Block (Microsoft)

Plan du cours

Installation d'un logiciel

En quoi ça consiste ?

- **Copie des binaires exécutables, bibliothèques** , documentation, fichiers de configuration, scripts de démarrage... dans les **répertoires de destination** de l'installation

Puis : **configuration** , par l'administrateur.

Installation d'un logiciel

En quoi ça consiste ?

- **Copie des binaires exécutables, bibliothèques** , documentation, fichiers de configuration, scripts de démarrage... dans les **répertoires de destination** de l'installation

Puis : **configuration** , par l'administrateur.

Destination de l'installation : dans un répertoire donné, noté `$DEST`

- Binaires dans `$DEST/bin`
- Bibliothèques dans `$DEST/lib`
- Documentation dans `$DEST/share/man`
- Fichiers de config dans `$DEST/etc`

Installation d'un logiciel

En quoi ça consiste ?

- **Copie des binaires exécutables, bibliothèques** , documentation, fichiers de configuration, scripts de démarrage... dans les **répertoires de destination** de l'installation

Puis : **configuration** , par l'administrateur.

Destination de l'installation : dans un répertoire donné, noté `$DEST`

- Binaires dans `$DEST/bin`
- Bibliothèques dans `$DEST/lib`
- Documentation dans `$DEST/share/man`
- Fichiers de config dans `$DEST/etc`

Choix du répertoire de destination :

- Dans le `$PATH` par défaut des utilisateurs normaux ? Uniquement du super-utilisateur ?
- Un répertoire spécial pour toutes les installations locales ?
- Dans le répertoire d'un utilisateur particulier ?

Installation à partir des sources

Compilation et installation à partir des sources du logiciel

- Fonctionne sur toutes les plate-formes supportées par le logiciel
- Uniformité : même procédure sur toutes les versions d'Unix

Installation à partir des sources

Compilation et installation à partir des sources du logiciel

- Fonctionne sur toutes les plate-formes supportées par le logiciel
- Uniformité : même procédure sur toutes les versions d'Unix

Trois étapes :

1. **Détection de l'environnement** (version du compilateur, des bibliothèques...), vérification que les dépendances sont présentes
   - Exécution d'un script fourni : `./configure`
   - Diverses options : localisation de bibliothèques, répertoire cible de l'installation (`--prefix=...`) : liste des options avec `--help`
   - Génère les fichiers Makefile si tout va bien
   - Si quelque chose ne va pas (mauvaise version du compilateur, dépendance manquante...) : affiche un message d'erreur (à lire !!!)

Installation à partir des sources

Compilation et installation à partir des sources du logiciel

- Fonctionne sur toutes les plate-formes supportées par le logiciel
- Uniformité : même procédure sur toutes les versions d'Unix

Trois étapes :

1. **Détection de l'environnement** (version du compilateur, des bibliothèques...), vérification que les dépendances sont présentes
   - Exécution d'un script fourni : `./configure`
   - Diverses options : localisation de bibliothèques, répertoire cible de l'installation (`--prefix=...`) : liste des options avec `--help`
   - Génère les fichiers Makefile si tout va bien
   - Si quelque chose ne va pas (mauvaise version du compilateur, dépendance manquante...) : affiche un message d'erreur (à lire !!!)
2. **Compilation** du code source
   - Commande `make`
   - Utilise les fichiers Makefile générés à l'étape précédente

Installation à partir des sources

Compilation et installation à partir des sources du logiciel

- Fonctionne sur toutes les plate-formes supportées par le logiciel
- Uniformité : même procédure sur toutes les versions d'Unix

Trois étapes :

1. **Détection de l'environnement** (version du compilateur, des bibliothèques...), vérification que les dépendances sont présentes
   - Exécution d'un script fourni : `./configure`
   - Diverses options : localisation de bibliothèques, répertoire cible de l'installation (`--prefix=...`) : liste des options avec `--help`
   - Génère les fichiers Makefile si tout va bien
   - Si quelque chose ne va pas (mauvaise version du compilateur, dépendance manquante...) : affiche un message d'erreur (à lire !!!)
2. **Compilation** du code source
   - Commande `make`
   - Utilise les fichiers Makefile générés à l'étape précédente
3. **Installation** du logiciel
   - Commande `make` avec l'option `install`
   - Installe les fichiers générés par la compilation
   - Suivant le répertoire cible de l'installation, peut-être besoin d'être root

Exemple

- **Téléchargement** des sources :

```
coti@gauss:/tmp$ wget ftp://alpha.gnu.org/pub/gnu/autoconf/autoconf-2.68b.tar.gz
```

- On **décompresse, désarchive** et on se place **dans le répertoire** obtenu :

```
coti@gauss:/tmp$ tar xzf autoconf-2.68b.tar.gz
coti@gauss:/tmp$ cd autoconf-2.68b/
```

- **Compilation et installation** à proprement parler :

```
coti@gauss:/tmp/autoconf-2.68b$ ./configure --prefix=/opt/logiciels/
coti@gauss:/tmp/autoconf-2.68b$ make
coti@gauss:/tmp/autoconf-2.68b$ sudo make install
```

- Les fichiers ont bien été copiés dans le répertoire de destination :

```
coti@gauss:/tmp/autoconf-2.68b$ ls /opt/logiciels/
bin  share
coti@gauss:/tmp/autoconf-2.68b$ ls /opt/logiciels/bin
autoconf  autoheader  autom4te  autoreconf  autoscan  autoupdate  ifnames
```

Utilisation d'un système de gestion de paquets

Système de gestion de paquets :

- Prévu par la distribution Linux
- **Spécifique** à la distribution
- Facilite l'installation en gérant automatiquement les dépendances
- **Une seule étape** pour tout

Utilisation d'un système de gestion de paquets

Système de gestion de paquets :

- Prévu par la distribution Linux
- **Spécifique** à la distribution
- Facilite l'installation en gérant automatiquement les dépendances
- **Une seule étape** pour tout

Qu'est-ce qu'un **paquet** ?

- Chaque logiciel est un **paquet** , avec un format spécifique
- Contient les sources ou les binaires pré-compilés du logiciel, éventuels scripts, documentation, des squelettes de fichiers de configuration...
- Contient les scripts d' **installation** de ce paquet en particulier : permet l'installation automatique

Utilisation d'un système de gestion de paquets

Système de gestion de paquets :

- Prévu par la distribution Linux
- **Spécifique** à la distribution
- Facilite l'installation en gérant automatiquement les dépendances
- **Une seule étape** pour tout

Qu'est-ce qu'un **paquet** ?

- Chaque logiciel est un **paquet** , avec un format spécifique
- Contient les sources ou les binaires pré-compilés du logiciel, éventuels scripts, documentation, des squelettes de fichiers de configuration...
- Contient les scripts d' **installation** de ce paquet en particulier : permet l'installation automatique

Système de gestion de paquets

- Connait les **dépendances** de chaque paquet
- Les installe automatiquement au préalable

Utilisation d'un système de gestion de paquets

Système de gestion de paquets :

- Prévu par la distribution Linux
- **Spécifique** à la distribution
- Facilite l'installation en gérant automatiquement les dépendances
- **Une seule étape** pour tout

Qu'est-ce qu'un **paquet** ?

- Chaque logiciel est un **paquet** , avec un format spécifique
- Contient les sources ou les binaires pré-compilés du logiciel, éventuels scripts, documentation, des squelettes de fichiers de configuration...
- Contient les scripts d' **installation** de ce paquet en particulier : permet l'installation automatique

Système de gestion de paquets

- Connait les **dépendances** de chaque paquet
- Les installe automatiquement au préalable

Exemples :

- **Debian** et dérivées (Mint, Ubuntu, Knoppix, etc) : paquets **.deb** , outils **dpkg** et **apt** (apt-get, apt-cache, aptitude...)
- **Red Hat** et dérivées (Fedora, Mandriva, Mageia, CentOS...) : paquets **.rpm** , outils **rpm** et dérivés (yum pour RedHat, Fedora et CentOS, urpmi, urpme, urpmq pour Mandriva et Mageia...)

Exemple (Debian)

- Dans **quel paquet** se trouve le logiciel à installer ?
  - `apt-cache search`

```
coti@gauss:~$ apt-cache search nfs-kernel-server
nfs-kernel-server - gestion du serveur NFS du noyau
```

- **Installation** du paquet :
  - `apt-get install`

```
coti@gauss:~$ sudo apt-get install nfs-kernel-server
```

- Quel est l'**état du paquet** ?
  - `dpkg -l`

```
coti@gauss:~$ dpkg -l nfs-kernel-server
||/ Nom             Version      Architecture Description
+++-===============-============-============-=================================
ii  nfs-kernel-ser 1:1.2.6-4    amd64        support for NFS kernel server
```

On voit `"ii"` : le paquet est installé en version 1.2.6-4

Fichier important : `/etc/apt/sources.list` : adresses des dépôts où `apt-get` va télécharger les paquets.

Exemple (Mandriva)

- Dans **quel paquet** se trouve le logiciel à installer ?
  - `urpmq` ou `urpmi -fuzzy -test`

```
[root@galois ~]# urpmq sudo
sudo
[root@galois ~]# urpmi --fuzzy --test sudo
Pas de paquetage nomme sudo
Les paquetages suivants contiennent sudo : fusiondirectory-plugin-sudo,
gnome-sudoku, ksudoku, ksudoku-handbook, ...
```

- **Installation** du paquet
  - `urpmi`

```
[root@galois ~]# urpmi sudo
```

- **Supprimer** un paquet
  - `urpme`

```
[root@galois ~]# urpme sudo
```