01 - Réseau et configuration

Sécurité des réseaux, M2 E-Secure

Gaétan Richard

16 septembre 2019

Université de Caen Normandie

Contenu

Aspects administratifs

Déroulement :

- 4 Cours magistraux (2h) ;
- 7 TPs (3h30) en salle S3-406 ;
- 2 Interventions (2h) ;
- Des présentations (3h30).

Évaluation :

- (2/3) Une épreuve machine (2h) ;
- (1/3) Une présentation orale.

Apsects pédagogiques

Contenu :

- Configuration réseau ;
- Configuration services bases (VPN, Mail, ...) ;
- Configuration avancées (Kerberos, Volp, ...).

Matériel :

- Alix (lien) : machines physiques ;
- GNS3 : machine virtuelles ;
- Réseau dédié en Salle S3-406.

Distributions : Debian 10, openBSD6.5.

Introduction

Introduction

Système

Système et configuration

Trois niveaux de configuration :

- gui ;
- ligne de commande ;
- fichier.

Services : les services du systèmes peuvent être manipulés par la commande

sudo service <*nom du service*> { start | stop | restart | reload }

Logs : les logs sont disponibles dans le dossier `/var/log` et en partie via la commande **dmesg**.

FHS

L'organisation du système de fichier se base sur la Filesystem Hierarchy Standard. Elle peut varier selon les systèmes.

- la configuration des services et du système est définie dans `/etc/` ;
- Les répertoires utilisateurs sont généralement définis dans `/home/` ;
- les périphériques sont définis dans `/dev/` ;
- le noyau de démarrage est installé dans `/boot/` ;
- les programmes, de manière générale, sont installés dans `/usr/` avec les sous-dossiers `bin/`, `src/`, `share/doc` ;
- les programmes hors système (installation manuelles) sont mis dans `/opt` ou `/usr/local`

RAID

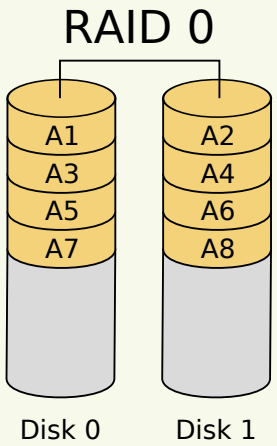
RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

- RAID 0 : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- RAID 1 : deux disques contenant les mêmes données ;
- RAID 5 : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.

RAID

RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

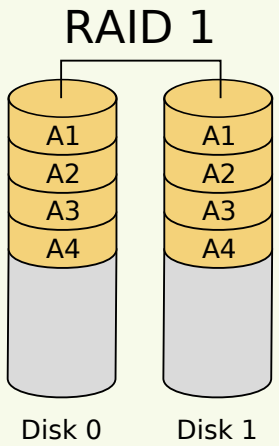
- RAID 0 : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- RAID 1 : deux disques contenant les mêmes données ;
- RAID 5 : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.



RAID

RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

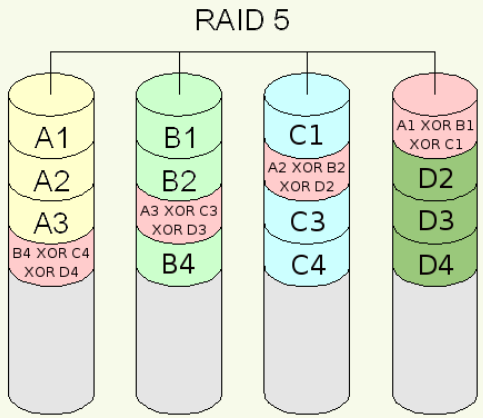
- RAID 0 : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- RAID 1 : deux disques contenant les mêmes données ;
- RAID 5 : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.



RAID

RAID (Redundant Array of Independent Disks) est une méthode pour dispatcher un bloc au travers de plusieurs disques et assurer une certaine redondance :

- RAID 0 : deux disques mis en parallèle pour des raisons de performances (pas de redondance) ;
- RAID 1 : deux disques contenant les mêmes données ;
- RAID 5 : $n + 1$ disques permettant de retrouver l'intégralité des données si 1 est hors-service.



Paquet

Dans un paquet, on trouve généralement :

- le programme (sous forme d'exécutable, de source ou de lien) ;
- des patchs ;
- des scripts d'installation ;
- son numéro de version ;
- les dépendances ;
- une description ; ...

Debian (Ubuntu) : Format deb (archive `ar`)

La gestion de paquet se fait à l'aide de la commande dpkg (bas-niveau), apt-get (ligne de commande), apt (ligne de commande interactif), synaptics (gui).

Introduction

Système (installation d'un logiciel)

Installation manuelle

Utilisation : pour disposer d'une version plus à jour ou personnalisée (utilisation avancée).

Étapes :

- Récupérer le code source ;
- configurer l'installation ;
- compiler ;
- installer ;
- configurer l'utilisation ;
- utiliser ;
- **maintenir à jour**.

Récupérer le code source

Release :

- Correspond à une version stable ;
- le plus souvent fournit sous la forme d'une archive `.tar.gz` ou `.tar.bz2` ;
- À décompresser avec la commande **tar**.

Développement :

- Version en mouvement ;
- Accès direct au dépôt de travail collaboratif (format *cvs*, *svn*, *mercurial*, *git*, ...) ;
- Utiliser le logiciel ad-hoc ;
- Existe parfois également en « snapshot ».

Fichiers utiles

Dans la plupart des projets, les fichiers et dossiers suivants sont présents :

- `README` : un peu de lecture ;
- `INSTALL` : documentation d'installation ;
- `Changelog` : les changements par version.
- `Doc/` : la documentation ;
- `src/` : les sources.

Versions

La numérotation des versions est propre à chaque application mais on peut distinguer le schéma général suivant :

major.minor-variante

- major est le numéro de version majeure, il est incrémenté lors de gros changement introduisant des incompatibilités (0 est souvent utilisé pour le début de développement) ;
- minor est le numéro de version mineur (dans certain cas, un numéro impair indique une version de développement, 99 désigne souvent la version préparatoire à la version majeure suivante) ;
- variante sert à indiquer le stade de développement
  - a1, a2, ... : versions *alpha* (ajout de fonctionnalités) ;
  - b1, b2, ... : versions *beta* (correction de bugs, test) ;
  - RC1, RC2, ... : versions *Release Candidate* (test avant release) ;
  - 1, 2, ... : *patchlevel* (patchs)

Configuration de l'installation

Le script **configure** permet de vérifier la présence (et la version) des dépendances, de choisir différentes options de compilation. Il génère alors les fichiers (`Makefile`) nécessaire pour la compilation.

Parmi les options courante de configure, on retrouve en particulier :

- `--help` : affichez la liste des options disponibles ;
- `--prefix=`*`/chemin/`* : installe le logiciel dans le répertoire */chemin/* ;
- `--with-`*X* / `--without-`*X* : active / désactive la fonctionnalité *X* ;
- `--enable-`*X* / `--disable-`*X* : comme précédemment.

Note : dans certains cas, ils est nécessaire de générer l'exécutable **configure** à partir du fichier `configure.in` en utilisant le programme **autoconf**.

Compilation

La compilation s'effectue en utilisant la commande **make** qui utilise les fichiers `Makefile`. Il est possible de préciser à make un *cible* pour désigner la tâche à effectuer.

Usuellement les tâches reconnues sont les suivantes :

- make : compile le programme ;
- make install : effectue l'installation (nécessite parfois d'être super-utilisateur) ;
- make clean : nettoie les fichiers générés ;
- make doc : génère la documentation.

Note : il existe également des alternatives à make utilisés par certains projets : *scons*, *ant*, ...

Configuration du programme

Finalisation : Il reste alors à adapter les fichiers de configuration généraux et par utilisateur. Cette étape est dépendante de l'application mais il est courant de fournir des fichiers commentés avec les options les plus courantes.

Mise à jour : Attention si vous avez inclus des bibliothèque de façon statique dans votre programme.

Introduction

Réseaux

Le but

Objectif : Faire transiter des informations au travers d'un réseau physique de machines.

Méthodes :

- Séparation en couches suivant les tâches ;
- Utilisation de protocoles.

Le modèle OSI ...

Le modèle OSI définit de façon théorique des couches et leurs rôles respectifs.

7-Application	Communique avec l'application
6-Présentation	Mets en forme les informations échangées
5-Session	Assure l'ouverture et la fermeture d'une session
4-Transport	Assure le transfert des données et la fragmentation
3-Réseau	Assure l'acheminement ou le routage
2-Liaison	Assure le transfert des trames sur une ligne
1-Physique	Transmet les bits sur le support physique

...et la réalité

Application	
Transport	ex : *TCP / UDP*
Réseau	ex : *IPv4 / IPv6*
Liaison - Physique	ex : *Wifi, ethernet, ppp, ...*

Couche physique

Adresse MAC

Identifiant : Chaque interface d'un machine possède un identifiant unique : **l'adresse MAC**. Cette adresse se compose de 5 blocs de 8 bits (octets) que l'on écrit usuellement en hexadécimal séparé par des `:`.

Ex : `00:26:4a:19:a1:70`

Décomposition : cette adresse se décompose en une portion constructeur et un compteur.

Usurpation : Cette adresse peut être modifiée.

ARP

Principe : Permet de faire le liens entre l'adresse IP et l'adresse MAC dans le cas d'une machine locale afin de permettre la transmission.

Encapsulation : Les paquets ARP sont encapsulés dans les trames Ethernet.

Couche physique

Adresses

Adresses IPv4 / IPv6

IPv4	IPv6
32 bits	128 bits
4 blocs d'un octet	8 blocs de deux octets
Entiers décimaux séparés par des .	Entiers hexadécimaux séparés par des :
Ex : `192.160.0.1`	Ex : `fe80::226:bbff:fe04:4e3`

Sous-réseaux

Pour grouper les adresses ensembles, on utilise la notion de sous-réseau (*netmask*).

10	74	254	1
00001010	01001010	11111110	00000001

Sous-réseaux

Pour grouper les adresses ensembles, on utilise la notion de sous-réseau (*netmask*).

10	74	254	1
00001010	01001010	11111110	00000001

- On sépare une partie identifiant réseau de la partie identifiant machine ;

Sous-réseaux

Pour grouper les adresses ensembles, on utilise la notion de sous-réseau (*netmask*).

10	74	254	1
00001010	01001010	11111110	00000001
11111111	11111111	11100000	00000000
255	255	224	0

- On sépare une partie identifiant réseau de la partie identifiant machine ;
- On obtient alors un masque réseau ;

Sous-réseaux

Pour grouper les adresses ensembles, on utilise la notion de sous-réseau (*netmask*).

10	74	254	1
00001010	01001010	11111110	00000001
11111111	11111111	11100000	00000000
255	255	224	0

- On sépare une partie identifiant réseau de la partie identifiant machine ;
- On obtient alors un masque réseau ;
- On peut également noter ce réseaux `10.74.254/19`

Choix des adresses

Choix des adresses : Il y a de nombreux choix possible pour les adresses, cependant, certaines conventions sont souvent prises. Par exemple, le routeur prend l'identifiant de machine 1 ou 254 en IPv4, on réserve les adresses basses pour les serveurs, ...

Adresses privées : Dans le cas d'IPv4, il est courant de ne pas avoir suffisamment d'adresses. Pour résoudre ce problème, on utilise des plages d'adresses privés :

- `192.168.0.0/16`;
- `172.16.0.0/12`;
- `10.0.0.0/8`.

/etc/network/interfaces

Configuration : Sous Linux, la configuration se fait par l'intermédiaire du fichier `/etc/network/interfaces`. Cette configuration peut être fixe ou dynamique (DHCP).

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
   post-up /sbin/iptables-restore < /etc/iptables.save

# The secondary interface
auto eth1
iface eth1 inet static
   address 192.168.1.2
   netmask 255.255.255.0
   gateway 192.168.1.1
```

/etc/hostname.XXX

Configuration : Sous BSD, la configuration se fait par l'intemédiaire d'un fichier `/etc/hostname.if` pour chaque interface *if*.

Il est possible de charger la nouvelle configuration à l'aide de la commande `sh /etc/netstart pnc0`.

```
media 100baseTX description Uplink
inet 10.0.1.12 255.255.255.0 10.0.1.255
inet alias 10.0.1.13 255.255.255.255 10.0.1.13
inet alias 10.0.1.14 255.255.255.255 NONE
inet alias 10.0.1.15 255.255.255.255
inet alias 10.0.1.16 0xffffffff
# This is an example comment line.
inet6 alias fec0 ::1 64
inet6 alias fec0 ::2 64 anycast
!route add 65.65.65.65 10.0.1.13
up
```

Routage statique

Routage

Utilisation : Le routage consiste à diriger les paquets IP suivant leurs adresses pour qu'ils arrivent à destination.

Ce routage peut se faire de façon statique ou dynamique.

Principe : En fonction de la destination du paquet (donné par un sous-réseau), on décide quelle interface doit être utilisée et s'il est nécessaire de contacter un autre relais.

Configuration : Sur un client, on a en général une seule route passant par la gateway (passerelle).

Attention : Penser à activer le routage ipv4 et ipv6 dans le noyau via la commande **sysctl** ou le fichier `/etc/sysctl.conf` (BSD et Linux).

Route

Sous Linux, la commande **route** permet d'afficher et de modifier les tables de routage.

```
$ /sbin/route -n
Table de routage IP du noyau
Destination     Passerelle      Genmask         Indic Metric Ref    Use Iface
10.130.4.0      0.0.0.0         255.255.255.0   U     0      0        0 eth0
192.168.129.0   10.130.4.254    255.255.255.0   UG    0      0        0 eth0
192.168.128.0   10.130.4.254    255.255.255.0   UG    0      0        0 eth0
192.168.0.0     10.130.4.254    255.255.128.0   UG    0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
0.0.0.0         10.130.4.1      0.0.0.0         UG    100    0        0 eth0
```

Adressage automatique

Adressage automatique

DHCP

Fonctionnement

- Un serveur DHCP fonctionne avec un état (**statefull**) ;
- Pour connaître l'état des ressources gérées, il maintient une liste d'associations entre chaque client et les paramètres attribués ;
- Chaque client est identifié par un DUID (DHCP unique identifier) qu'il génère en fonction de son adresse de lien local. Le client ne doit plus changer de DUID, même s'il change d'adresse de lien local.
- Découverte d'un serveur DHCP par l'envoi d'un message multicast à l'adresse `FF02::1:2` ;
- Requêtes DHCP envoyées sur le port UDP 547, réponses reçues sur le port UDP 546 ;
- Le DHCP fourni usuellement un serveur DNS.

Messages DHCPv6

Le protocole DHCPv6 met en jeu 12 messages différents

- Sollicitation (DHCP Solicit) : interrogation de la présence de serveurs DHCP
- Annonce (DHCP Advertise) : émis par les serveurs en réponse à une sollicitation
- Requête (DHCP Request) : demande de paramètres de configuration de la part d'un client
- Réponse (DHCP Reply) : contient les valeurs des paramètres de configuration demandés
- Confirmation (DHCP Confirm) : demande de confirmation de la validité des paramètres alloués au client
- Renouvellement (DHCP Renew) : demande de prolongation de l'adresse IP affectée
- Réaffectation (DHCP Rebind) : idem, mais un autre serveur DHCP peut répondre
- Libération (DHCP Release) : indication du client que l'adresse allouée est libérée
- Refus (DHCP Decline) : indication du client qu'une ou plusieurs adresses affectées sont déjà utilisées sur son lien

Exemple de configuration

Debian : paquet isc-dhcp-server

Exemple de fichier `/etc/dhcp/dhcpd.conf`

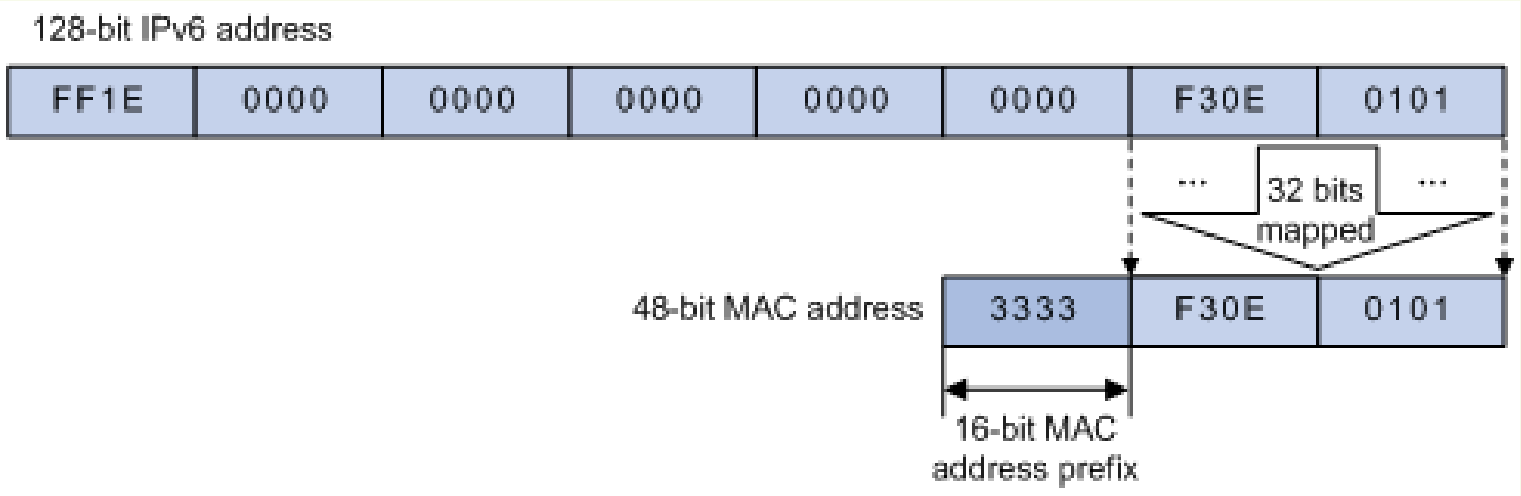
```
option domain-name "mydebian";
# Utilisation du serveur DNS public de Google (ou bien utilisez l'adresse du serveur DNS f
option domain-name-servers 8.8.8.8, 8.8.4.4;
# Configuration de votre sous-réseau (subnet) souhaité :
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.101 192.168.1.254;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.100;
    option domain-name-servers home;
}
default-lease-time 600;
max-lease-time 7200;
# Indique que nous voulons être le seul serveur DHCP de ce réseau :
authoritative;
```

Adressage automatique

EUI

Multicast IPv6 et MAC

On associe une adresse MAC à chaque adresse IPv6.



La machine utilise alors l'adresse MAC produite pour "écouter" les paquets multicast.

Adresse link-local

Adresse unique :

- Adresse MAC : `xx:xx:xx:xx:xx:xx` (6 octets)
- Adresse EUI-64 : `xx:xx:xx:`**`ff:fe`**`:xx:xx:xx` (8 octets)
- Adresse link-local : `fe80::xxxx:xx`**`ff:fe`**`xx:xxxx`.

$x = x \oplus 2$

Validité : Cette adresse n'est valide que sur le réseau local.

Adressage automatique

SLAAC

Stateless autoconfiguration : Étape 1

Vérifier que l'adresse n'est pas utilisée :

La machine envoie un paquet Neighbor Solicitation à l'adresse multicast sollicitée forgée à partir de l'adresse link-local désirée. À cette étape, on utilise encore l'adresse `::` (adressse non spécifiée).

Réponse :

Toute machine qui reconnaît son adresse répond à l'adresse `ff02::1`. Si une telle réponse est reçu, on doit prendre une autre adresse (aléatoire ou manuelle).

Stateless autoconfiguration : Étape 2

Trouver un routeur : L'interface ayant une adresse lien-local, elle l'utilise (comme adresse source) pour envoyer un message de "découverte de routeurs". Ce message est envoyé à `ff02::2`. Tous les routeurs sont à l'écoute sur l'adresse MAC `33:33:00:00:00:02`.

Réponse : Les routeurs répondent à la sollicitation (si programmés pour cela – radvd) en envoyant un préfixe utilisable (un /64 en général). La réponse est envoyée à tous les nœuds du lien local avec l'adresse `ff02::1`, afin d'avertir de la présence du routeur toutes les machines du réseau

Stateless autoconfiguration : Étape 3

Création d'adresse : La ou les machines qui reçoivent la réponse du routeur vont utiliser le préfixe reçu pour se forger une adresse globale : préfix : EUI-64.

Ultime vérification : L'adresse ainsi forgée doit toutefois être unique, il faut donc vérifier qu'elle n'est pas déjà prise sur le réseau local (même procédure que pour la première étape, non obligatoire toutefois car partie machine identique à celle de l'ad lien-local)

Route : En outre, l'adresse du routeur (souvent la lien-local) sera utilisée comme route par défaut.

Précisions :

Multiples routeurs : En cas de plusieurs routeurs sur le réseau, chaque machine pourra se forger plusieurs adresses globales. Chaque routeur peut annoncer des routes précises, ainsi qu'une priorité (basse, moyenne, haute) pour l'utiliser comme route par défaut.

Données annexes : Chaque machine peut ainsi s'autoconfigurer avec une adresse globale (au moins), un masque de sous-réseau, une route par défaut Il manque toutefois l'adresse d'un DNS, le nom du domaine,...

Ceci peut être obtenu avec un DHCP, en lui demandant seulement ces paramètres (pas l'adresse, puisqu'on en a déjà une)

Confidentialité

Problème : On peut suivre une machine avec l'unicité de l'UEI-64.

Solution : Privacy Extensions for Stateless Address Autoconfiguration in IPv6 (RFC 4941). Ajout d'une nouvelle adresse temporaire semblant aléatoire.

Routage dynamique

Neighbor Discovery Protocol

Router Advertisement Daemon (**radvd**)

Fonctionnement : Utilise des messages ICMP :

- Router Solicitation (Type 133) ;
- Router Advertisement (Type 134) ;
- Neighbor Solicitation (Type 135) ;
- Neighbor Advertisement (Type 136) ;
- Redirect (Type 137).

DNS : il est possible d'envoyer dans une option une liste de serveurs DNS mais cette RFC est récente (2010).

RIP

Routing Information Protocol : RFC 2453

- Basé sur le principe du vecteur de distance : les réseaux directement accessibles sont à une distance 0, n s'il faut traverser n routeurs.
- Les routeurs munis d'un logiciel compatible RIP envoient régulièrement leurs informations aux routeurs voisins.
- Ces derniers ajustent éventuellement leurs tables de routage en fonction des informations apprises : nouvelles routes, changement de chemin dû à une amélioration ou une détérioration de la distance.

Principes de configuration

- Le protocole RIP étant supposé activé (mode actif : envoi et écoute des infos ; mode passif : seulement écoute des infos reçues), chaque routeur décide :
  - de quelles infos il envoie précisément
  - du ou des réseaux sur le(s)quel(s) il les envoie.
- Ce peut être tout, partout ou plus sélectif. En général, il y a des réseaux sur lesquels il est inutile d'envoyer les infos.

Informations apprises :

- Si un routeur A reçoit une info concernant le réseau X venant d'un routeur voisin B, la distance de X à B étant d :
  - Si cette info n'améliore pas la distance de A à X, on ne change rien.
  - Si cette info améliore la distance de A à X (ou si X n'était pas encore connu), alors on modifie ou on ajoute la ligne de la table de routage concernant X : X est à une distance $d+1$, via le routeur B (et via l'interface par laquelle on a reçu l'information)

RIP6

Configuration linux :

- Via la suite de logiciels de routage quagga ;
- Fichier `ripngd.conf` (IPv6).

```
! -*- rip -*-
!
hostname ripngd
password zebra
! debug ripng events
! debug ripng packet
!
router ripng
  network eth0
  route 2001 :660 :7101 :XX ::0/64
  distribute-list local-only out eth0
!
```

Configuration BSD :

- Via le logiciel `route6d` (inclus dans l'installation de base).
- Au travers des variables d'environnements de `/etc/rc.local`.

```
route6d_flags=""        # for normal use : ""
                        # be sure to set net.inet6.ip6.forwarding=1
```

DNS

Contenu du DNS

Contenu : Le DNS contient :

- Des informations sur le gestionnaire du domaine (champ *SOA*) ;
- Des serveurs de nom de domaine (champ *NS*) ;
- Des adresses IPv4 correspondant au nom des machines (champ *A*) ;
- Des adresses IPv6 (champ *AAAA*) ;
- Des alias (champ *CNAME*) ;
- Des noms correspondant à des adresses (champ *PTR*) ;
- Des serveurs de mail (champ *MX*) ;
- ...

Principe du DNS

dns

dns racine

dns `.fr`

dns `unicaen.fr`

Principe du DNS

`www.unicaen.fr ?`

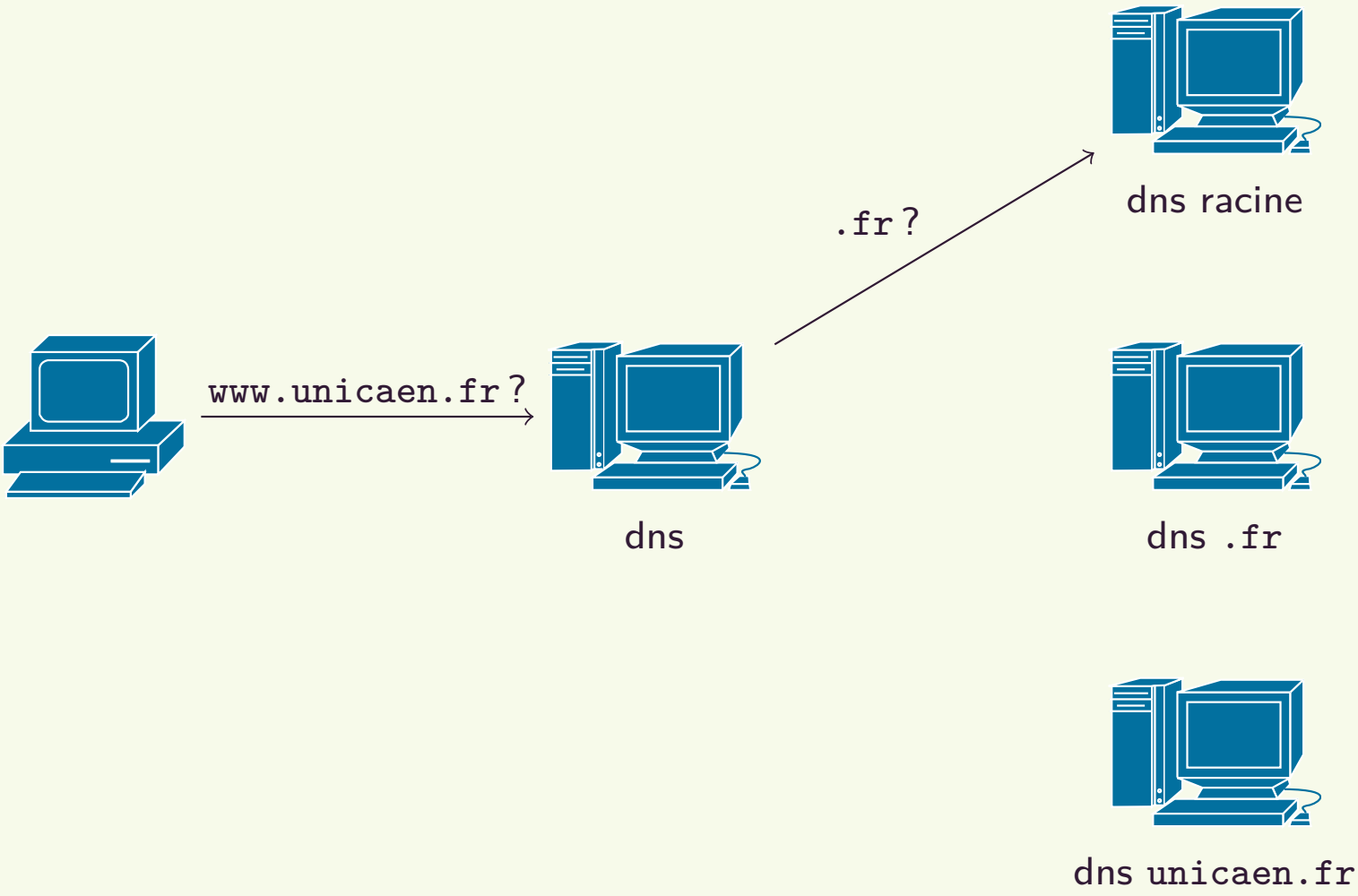
dns

dns racine

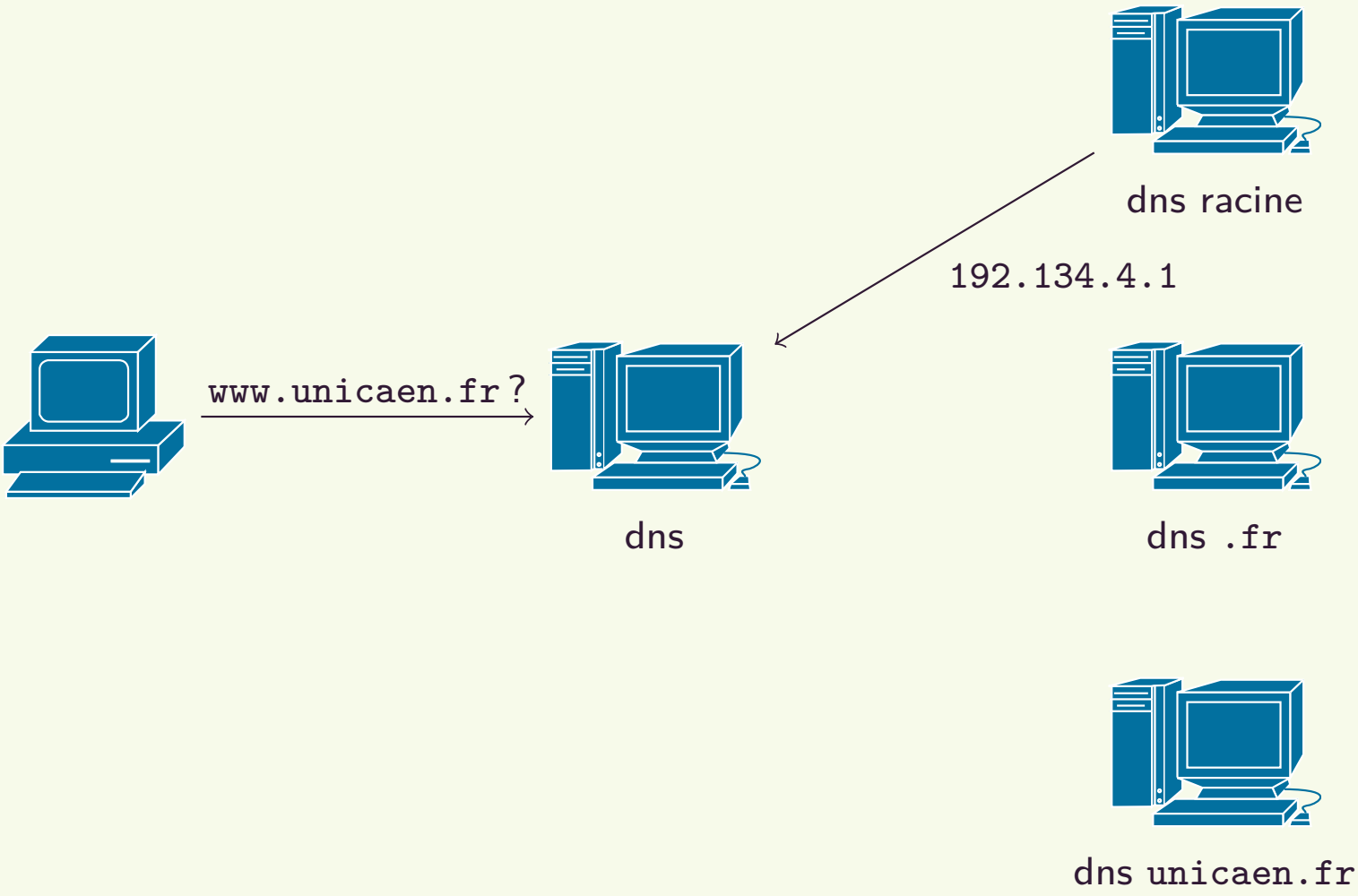
dns `.fr`

dns `unicaen.fr`

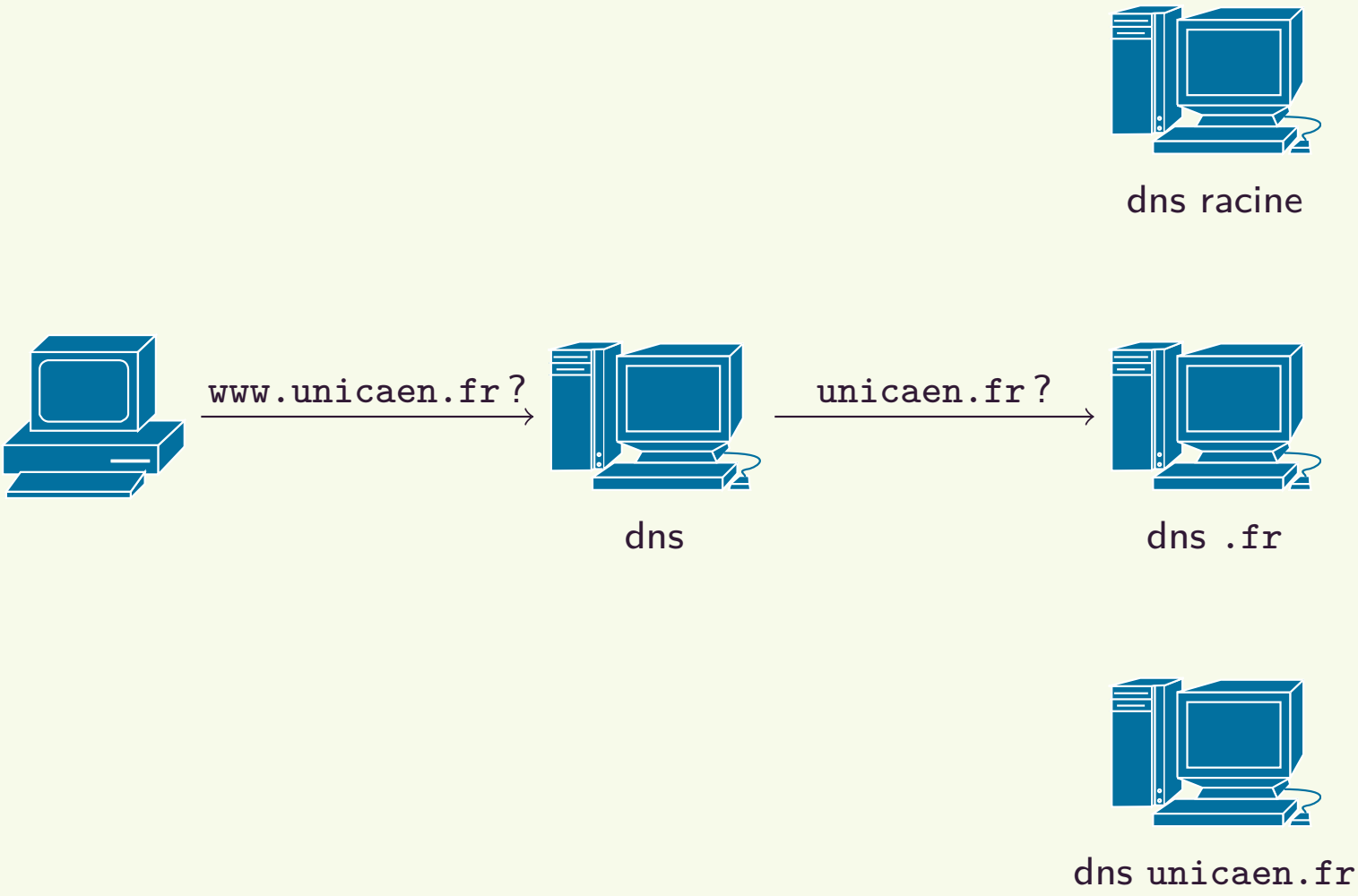
Principe du DNS



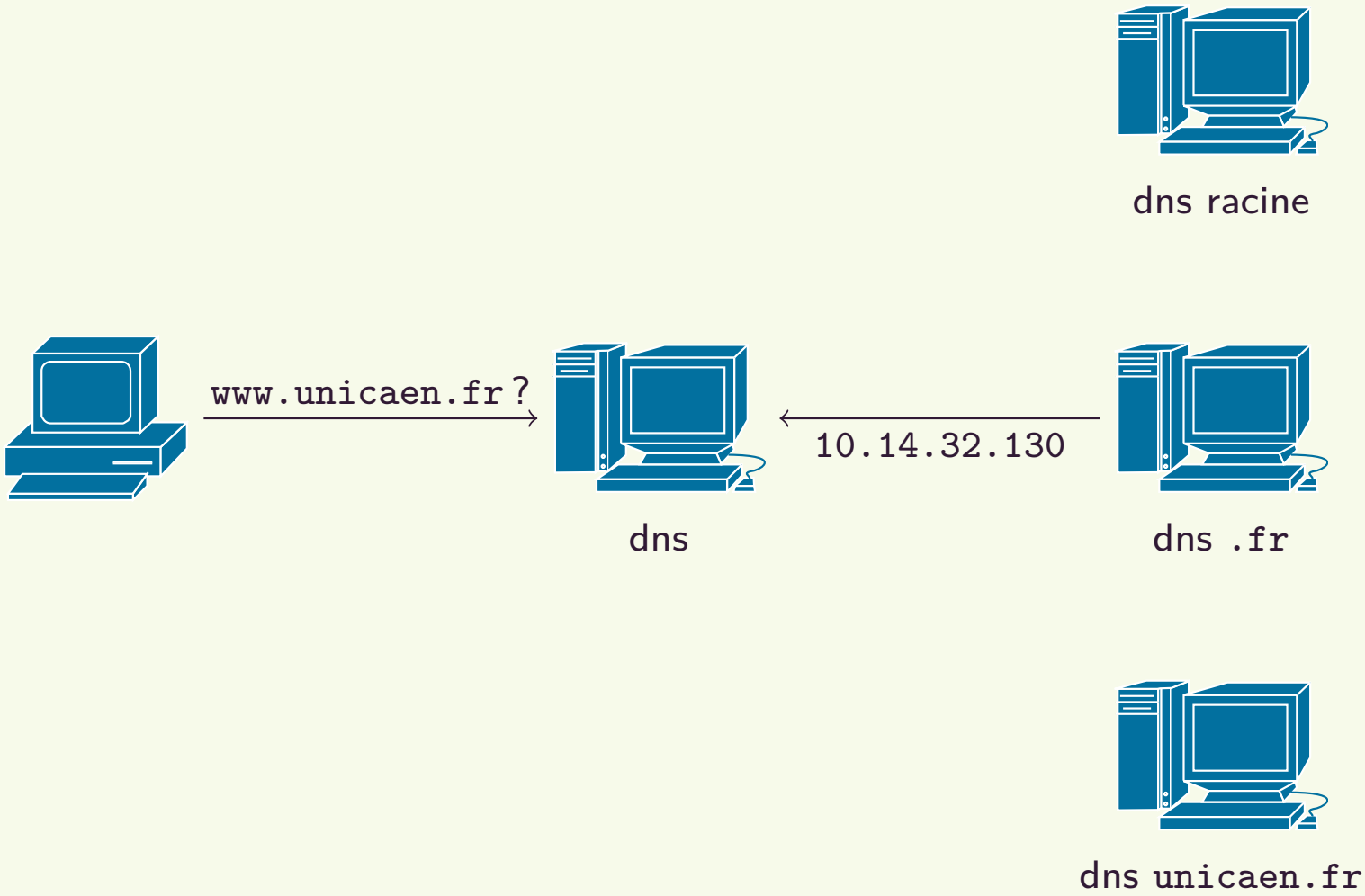
Principe du DNS



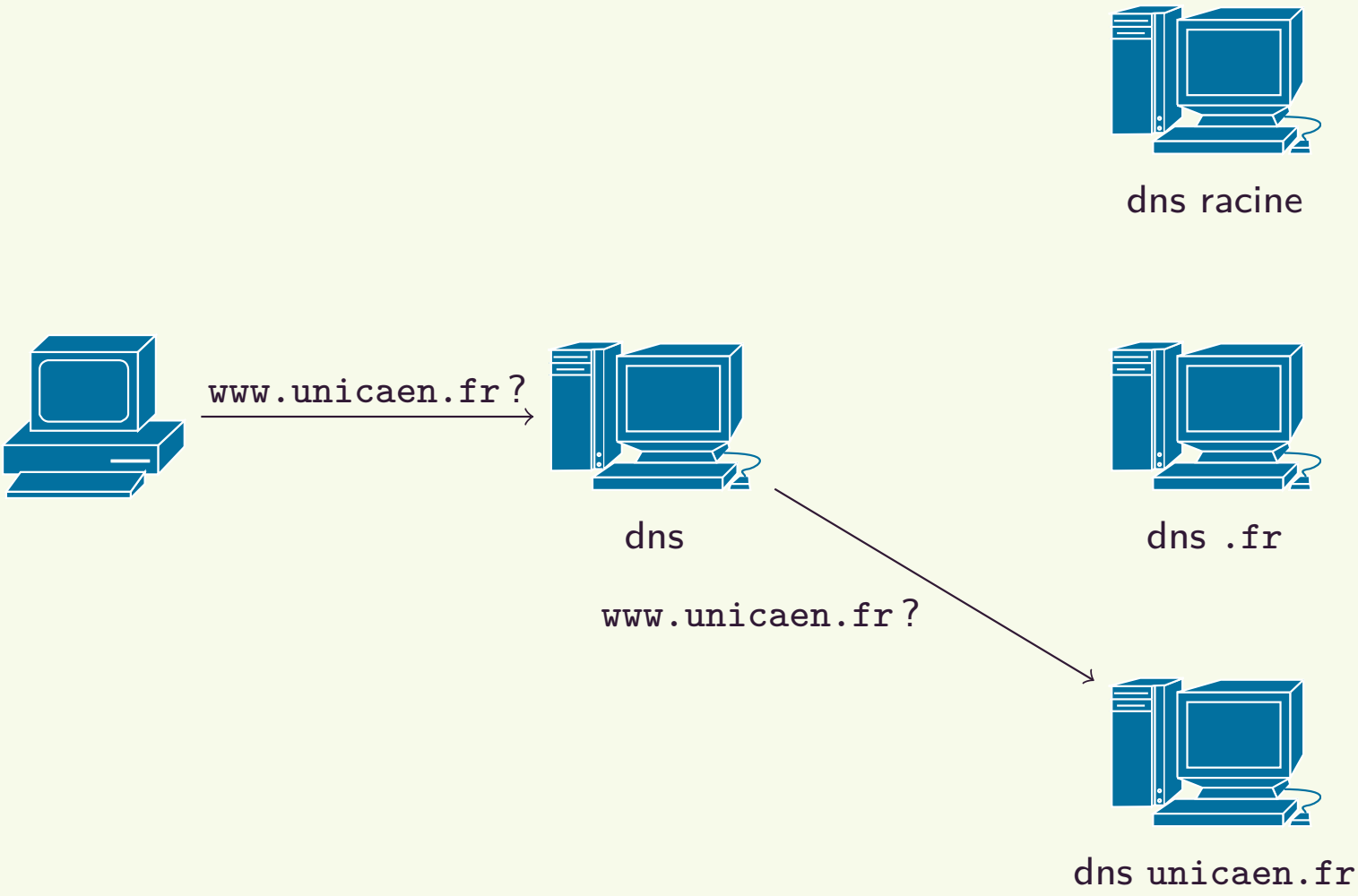
Principe du DNS



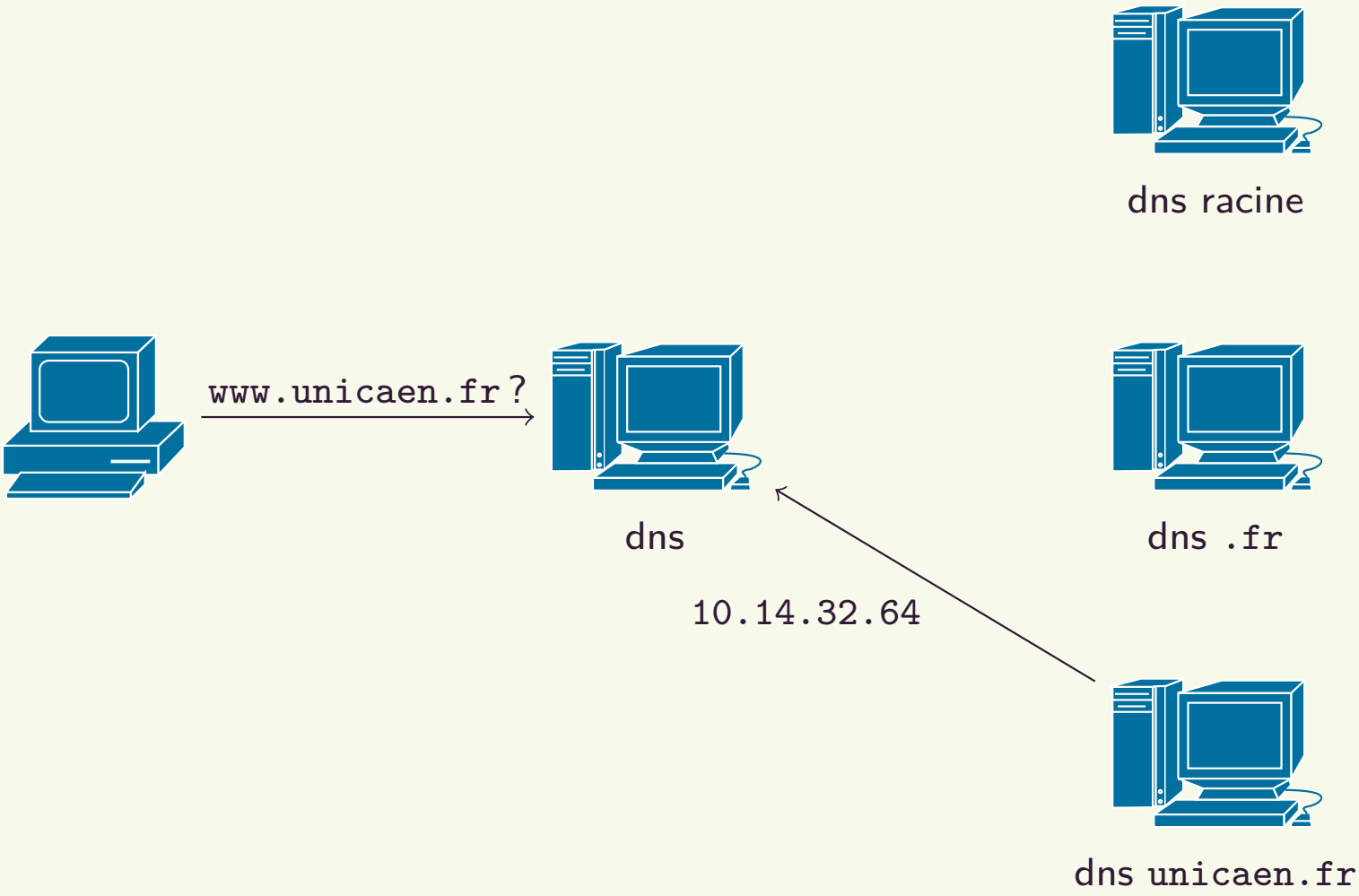
Principe du DNS



Principe du DNS



Principe du DNS



Principe du DNS

www.unicaen.fr ?

10.14.32.64

dns

dns racine

dns .fr

dns unicaen.fr

Configuration d'un DNS

Principe : Le logiciel *bind* se configure à l'aide d'un fichier global de configuration `/etc/bind/named.conf` et à l'aide de fichiers contenant les informations (de la forme `/etc/bind/db.x`).

Exemple :

```
TTL    604800
@       IN      SOA     zoneZZ.tp.info.unicaen.fr. root.zoneZZ.tp.info.unicaen.fr. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
@       IN      NS      dns
@       IN      MX      m3

dns     IN    A     192.168.1.2
dns      IN    AAAA    2001 :660 :7101 :ZZ :1 ::2
m1     IN     A     192.168.1.1
m2     IN     A     192.168.1.2
m3     IN     A     192.168.1.3
m4      IN      CNAME   m3
```

Configuration d'un DNS

Principe : Le logiciel *bind* se configure à l'aide d'un fichier global de configuration `/etc/bind/named.conf` et à l'aide de fichiers contenant les informations (de la forme `/etc/bind/db.x`).

Exemple :

```
@       IN      SOA     zoneZZ.tp.info.unicaen.fr. root.zoneZZ.tp.info.unicaen.fr.
...

@    IN    NS    dns.zoneZZ.tp.info.unicaen.fr.
1.1    IN    PTR    m1.zoneZZ.tp.info.unicaen.fr.
2.1    IN    PTR    m2.zoneZZ.tp.info.unicaen.fr.
3.1     IN    PTR    m3.zoneZZ.tp.info.unicaen.fr.

...
```

En cas d'erreur

Attention : En cas d'erreur dans un des fichiers de données, le serveur `dns` "saute" ce fichier et ne le marque que dans le fichier de log `/var/log/daemon.log`.

Bonne pratique : Créer un script qui relance le serveur puis affiche la fin du fichier de log pour vérifier le bon déroulement.

Cache : Une donnée erronée peut persister plusieurs heures dans le réseau.

UDP *vs* TCP

Attention : Certaines requêtes (listing ou nom très long) nécessitent le protocole TCP au lieu de l'habituel UDP.

Ceci sert à éviter les attaques DOS par rebond.

Filtrage : le DNS du FAI est le point actuellement privilégié pour effectuer du filtrage administratif (compromis coût / efficacité / effets indésirables).

Redondance

Importance : le DNS est un service critique puisque quasi tous les autres services en dépendent.

Redondance : Il est possible de créer un serveur `dns` redondant à l'aide du mode *maître-esclave*.

Très souvent, ce serveur est mis sur un autre site géographiquement séparé.

Transition IPv4/IPv6