

# La couche application

## TCP/IP

# Couche application

## Généralités

Orientés utilisateur final

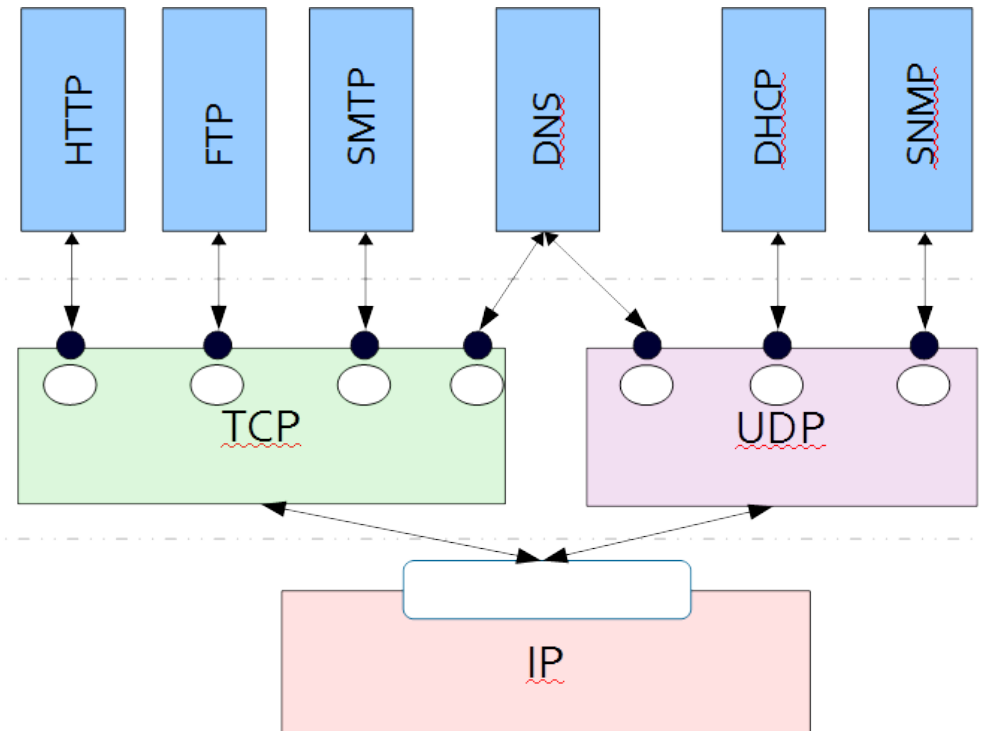
Application

Protocoles applicatifs avec  
un impact sur les couches  
basses

- DNS
- DHCP

Transport

Réseau



Pile TCP/IP : implémentée par les équipements en réseau

# *Domain Name System*

## Introduction

DNS: *Domain Name System* [RFC 1034, 1035]

- Le rôle d'un serveur DNS est de traduire un alias IP en adresse IP
- Ex: *www.univ-jfc.fr* -> *194.57.185.239*

L'alias IP permet à un être humain de mémoriser l'adresse des serveurs

En effet retenir une adresse numérique est particulièrement pénible (surtout en IPv6)

Le serveur DNS permet donc de résoudre les alias (ou nom) en adresse IP

Il peut également faire l'opération inverse: on parle alors de résolution inverse

DNS est décentralisé et hiérarchisé

- Avec DNS l'espace de nommage est organisé sous la forme d'une hiérarchie
- Cette hiérarchie se présente sous la forme d'un arbre avec en haut une racine et en dessous les TLD (*Top-Level Domain*)
- Les TLD sont gérés par l'ICANN (*Internet Corporation for Assigned Names and Numbers*)
- Ex : *.com* pour le commerce, *.edu* pour les universités, les codes pays sur 2 lettres (*fr*)

# *Domain Name System*

## Introduction

Comme pour les adresses IP, la gestion de sous-domaine n'est pas directement gérée par l'ICANN mais déléguée à différents organismes

- Verisign gère les .com et les .net notamment
- L'INTERNIC gère les .edu et les .org
- Pour la France (.fr et .re) la gestion est assurée par l'AFNIC (Association Française pour le Nommage Internet en Coopération)

Ces gestionnaires vendent les sous-domaines qui sont ensuite libres d'utilisation par leurs propriétaires

Ex: univ-jfc.fr a été fournie par l'AFNIC

L'Université peut maintenant créer n'importe quel sous-domaine pour peu qu'elle dispose de suffisamment d'adresses IP

- *La seule limite est de ne pas dépasser 63 caractères au total*

Le sous domaine castres.univ-jfc.fr est géré localement

# *Domain Name System*

## Vocabulaire

Le terme de domaine désigne indifféremment un domaine ou un sous domaine

- Le DNS est insensible à la casse
- Les points séparent des *labels* ou composants: **castres.univ-jfc.fr** est donc composé de trois *labels*: castres, univ-jfc, fr

FQDN: *Fully Qualified Domain Name* (Nom de domaine absolu)

- Présente le nom complet (totalement qualifié) faisant le lien entre le sous-domaine et la racine principale
- Il doit normalement se terminer par un « . »
- Ainsi: **castres.univ-jfc.fr** n'est pas un FQDN mais **castres.univ-jfc.fr.** l'est

Un FQDN est limité à 253 caractères

- Le solveur est le client DNS installé sur un hôte pour la résolution de nom
- DNS peut utiliser indifféremment TCP ou UDP (port 53) En pratique la plupart des serveurs DNS utilisent UDP car plus rapide

# *Domain Name System*

## Obtenir un domaine

L'achat d'un sous-domaine se fait auprès d'un *registrar* (ou registraire)

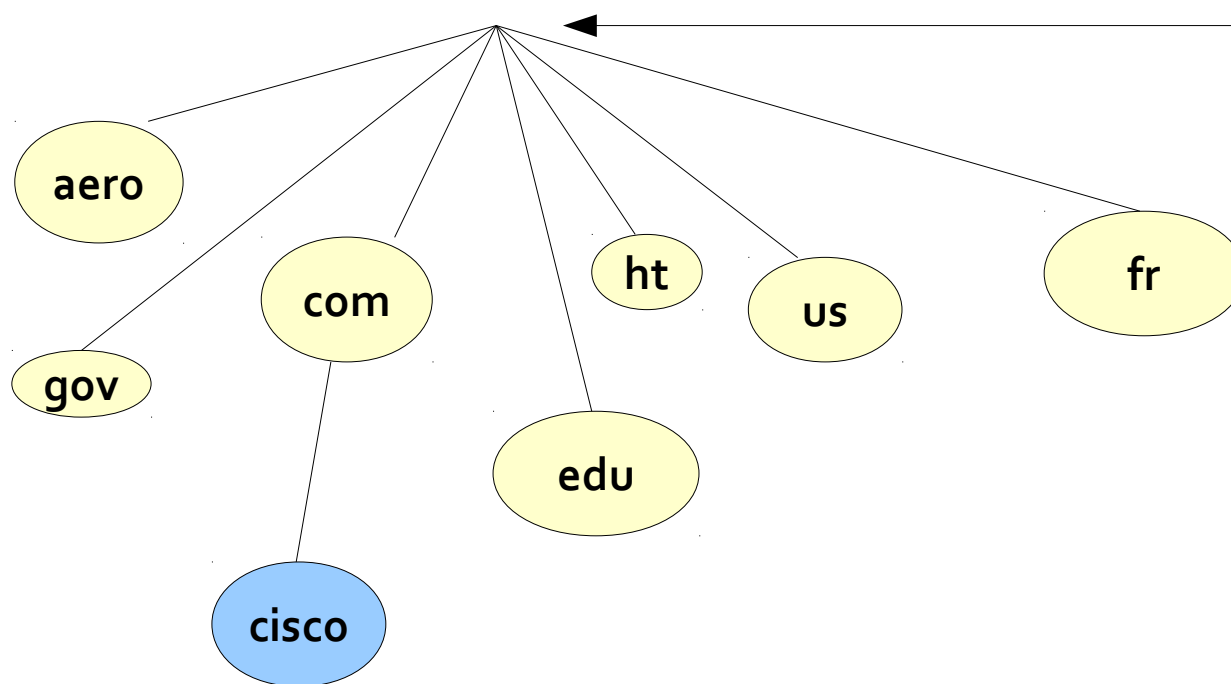
- La possession d'un sous domaine est limitée dans le temps en général 10ans

Pour acheter un sous domaine il faut associer l'identité et l'adresse postale d'une personne

- Il lui faut fournir trois informations
  - *Administrative contact*
  - *Billing contact*
  - *Technical contact*
- Cette identité est publique et peut être retrouvée grâce à la commande whois sous Linux par exemple (ou <http://www.whois.net>)

# Domain Name System

## Exemple de hiérarchie DNS



- Serveurs racine
- De *a-root-server* jusqu'à *m-root-server*
- Réplication forte => fiabilité et performances
- Présence en de multiples points du globe et routage anycast

Où se trouve `impot.gouv.fr` ?  
`Amazon.com` ?

# Domain Name System

## Résolution de nom

Réponse à une requête DNS : enregistrement(s) de ressources ou *resource record*

- DNS : fonction de *mapping* entre enregistrements de ressources et nom de domaine
- Contenu
  - Nom de domaine
  - Durée de vie : stabilité/volatilité de l'enregistrement → *caching*
  - Classe : IN pour Internet
  - Type
  - Valeur
  - Ex :
    - monpc 86400 IN A 130.37.16.5 : l'adresse de monpc est...
    - monpc 86400 IN MX monpc : monpc est capable de recevoir le courrier entrant

Le solveur de noms a deux possibilités pour résoudre un nom de domaine

- La résolution récursive
- La résolution itérative



# *Domain Name System*

## Résolution de nom

### La résolution itérative

- Le solveur interroge le serveur de nom de son domaine
- Si le serveur ne connaît pas la réponse il renvoie un message indiquant quel serveur est susceptible de connaître la réponse (s'il le connaît)
- Le solveur contacte ensuite seul le serveur indiqué

### La résolution récursive

- Le solveur interroge le serveur DNS de son sous-domaine
- Si le serveur ne connaît pas la réponse il fait suivre la requête à un autre serveur
- Le processus se répète récursivement jusqu'à obtenir une réponse
- C'est le mode par défaut utilisé par les solveurs
- Il est limité aux hôtes appartenant au même réseau local

# Domain Name System

## Résolution de nom

### Le *caching* DNS

- Lorsqu'un serveur met en œuvre une résolution récursive, il stocke les résultats dans ses propres tables
- Il fournit ensuite aux hôtes qui le sollicitent pour la même adresse une réponse en indiquant qu'il n'est pas l'autorité à solliciter normalement (*Non-Authoritative Answer*)

Chaque entrée dans le cache a un TTL (obtenu dans la réponse du serveur autorité)

```
D:\Users\rdalce.CASTRES>nslookup
Serveur par défaut :  bbox.lan
Address:  192.168.1.254

> 9gag.com
Serveur :  bbox.lan
Address:  192.168.1.254

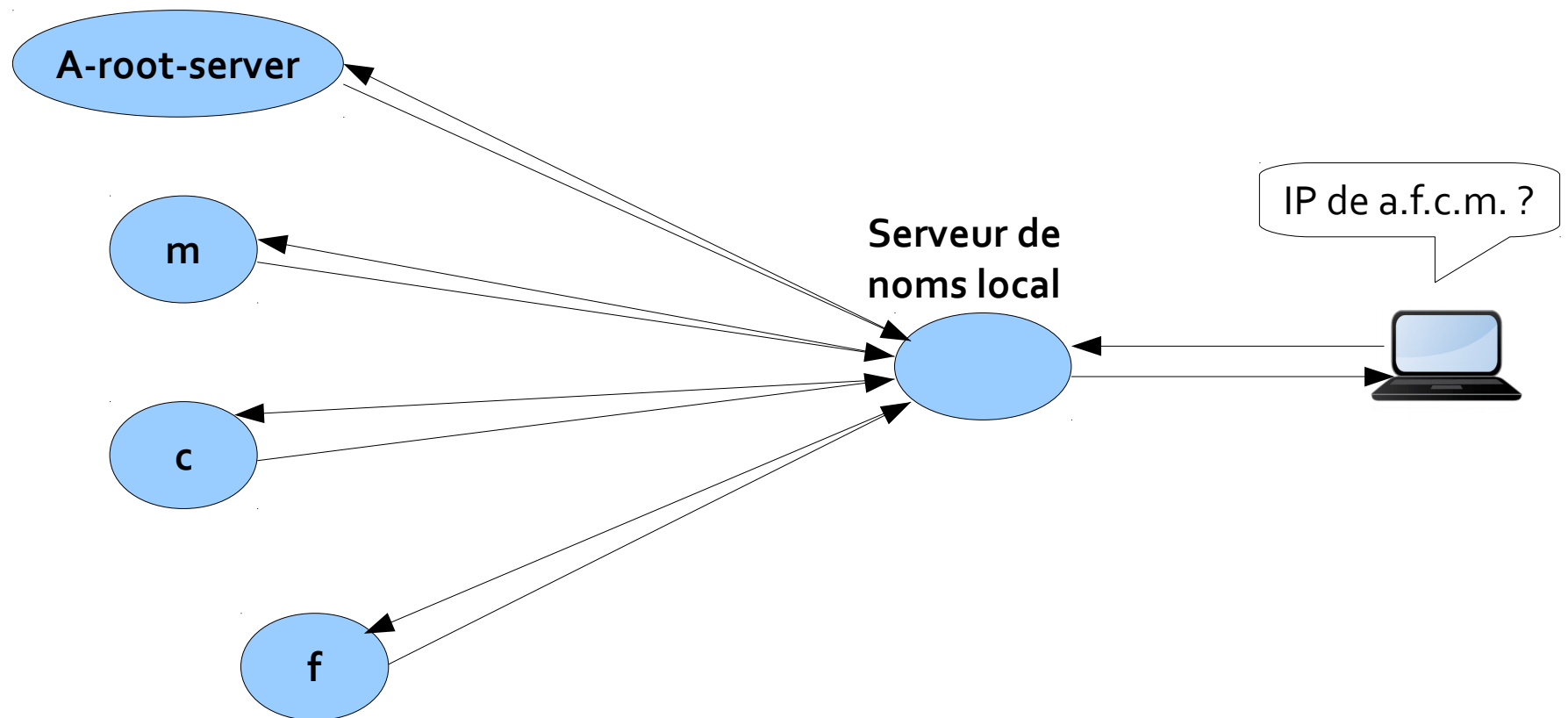
Réponse ne faisant pas autorité :
Nom :      9gag.com
Addresses: 151.101.2.133
           151.101.66.133
           151.101.194.133
           151.101.130.133

> a.edu-servers.net
Serveur :  bbox.lan
Address:  192.168.1.254

Réponse ne faisant pas autorité :
Nom :      a.edu-servers.net
Address:  192.5.6.30
```

# Domain Name System

## Exemple de résolution de nom

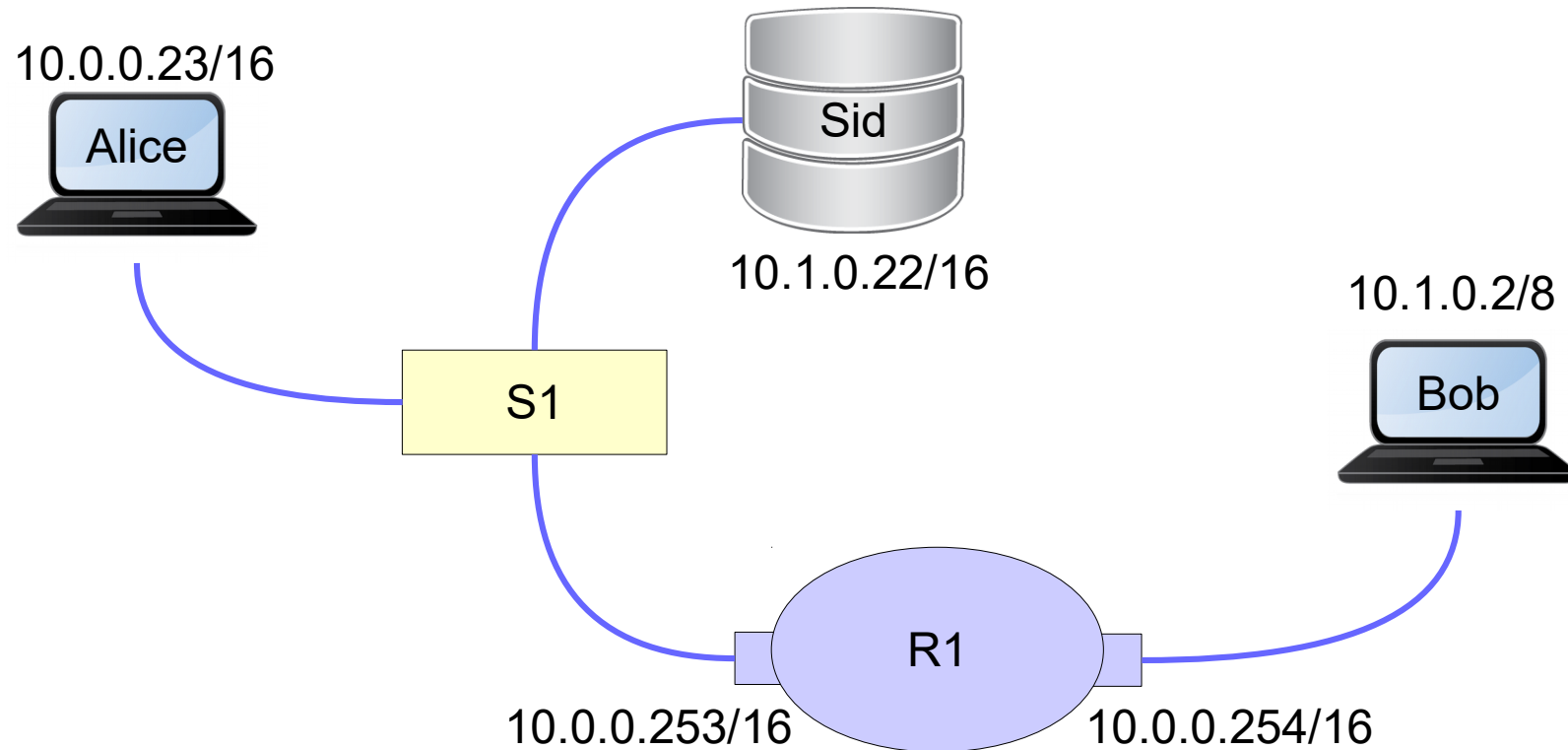


Renvoi au serveur de noms le  
plus pertinent  
Processus \_\_\_\_\_

Résolution faite pour le client  
Processus \_\_\_\_\_

# *Dynamic Host Configuration Protocol*

## Introduction

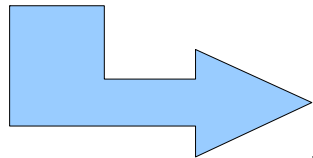


Remarques ?

# *Dynamic Host Configuration Protocol*

## Introduction

- Evolution des réseaux
  - Multiplication et accroissement des LAN
  - Population dynamique des LAN
    - Mobilité des utilisateurs
    - Utilisation de machines diverses
  - Administration réseau
    - Configuration IP et capacité à communiquer
    - *Default gateway*
    - IP collision et erreur de config
- Automatisation de la configuration
- Fichier de configuration global pour le réseau
  - « Centralisé » pour un LAN
- Protocole de distribution de l'information
- Protocole de configuration dynamique des hôtes



# *Dynamic Host Configuration Protocol*

## Introduction

- Protocole client-serveur
  - UDP (port #\_\_)
- Échange initié par le client
  - Récupération d'une adresse IP, d'un masque de réseau...
    - Cette IP peut être statique !!
  - Affectation des paramètres de passerelle

### Serveur

- Rôles
  - Stockage persistant des informations de configuration réseau
  - Allocation dynamique d'adresses
    - Pool d'adresses IP
      - « Location » d'une adresse IP par le client
    - Bail : limité dans le temps
      - Réutilisation des adresses IP
      - Pour vérifier le bon fonctionnement, ping par le serveur et ARP par le client (recommandés)
- Non-unicité du serveur DHCP
- Non-connu à priori du client

# Dynamic Host Configuration Protocol

## Format du message

OP code	HwType	HwLen	Hops
Transaction ID			
Seconds		Flags	
Ciaddr			
Yiaddr			
Siaddr			
Giaddr			
Chaddr			
Sname			
File			
Options			

### OP code

- Code de l'opération
- Client : requête, serveur : réponse

### HwType

- Type d'adresse de niveau 2
- Ex : '1' = 10Mb/s Ethernet

### HwLen

- Longueur en octets de l'adresse de niveau 2
- Ex : '\_\_\_' pour 10Mb/s Ethernet

### Hops

- '0' pour les clients
- Utilisés optionnellement par les relais DHCP

### Transaction ID

- Nombre aléatoire choisi par le client
- Utilisé par le client et le serveur pour apparier les messages dans la transaction

### Seconds

- Durée depuis le début de la procédure de renouvellement
- Renseigné par le client

# Dynamic Host Configuration Protocol

## Format du message

OP code	HwType	HwLen	Hops
Transaction ID			
Seconds		Flags	
Ciaddr			
Yiaddr			
Siaddr			
Giaddr			
Chaddr			
Sname			
File			
Options			

### Flags

- 15 bits réservés, à « 0 »
- LSB à 1 pour indiquer le *broadcast*

### Ciaddr

- Adresse IP courante du client
- Renseigné durant le renouvellement

### Yiaddr

- Adresse IP allouée au client

### Siaddr

- Adresse IP du serveur
- Renseigné dans le DHCPOFFER et le DHCPREQUEST

### Giaddr

- Adresse du relais DHCP si utilisé

### Chaddr

- Adresse MAC du client

### Sname

- Nom du serveur (facultatif)

### File

- Chemin du fichier de démarrage



# *Dynamic Host Configuration Protocol*

## Format du message

OP code	HwType	HwLen	Hops
Transaction ID			
Seconds		Flags	
Ciaddr			
Yiaddr			
Siaddr			
Giaddr			
Chaddr			
Sname			
File			
Options			

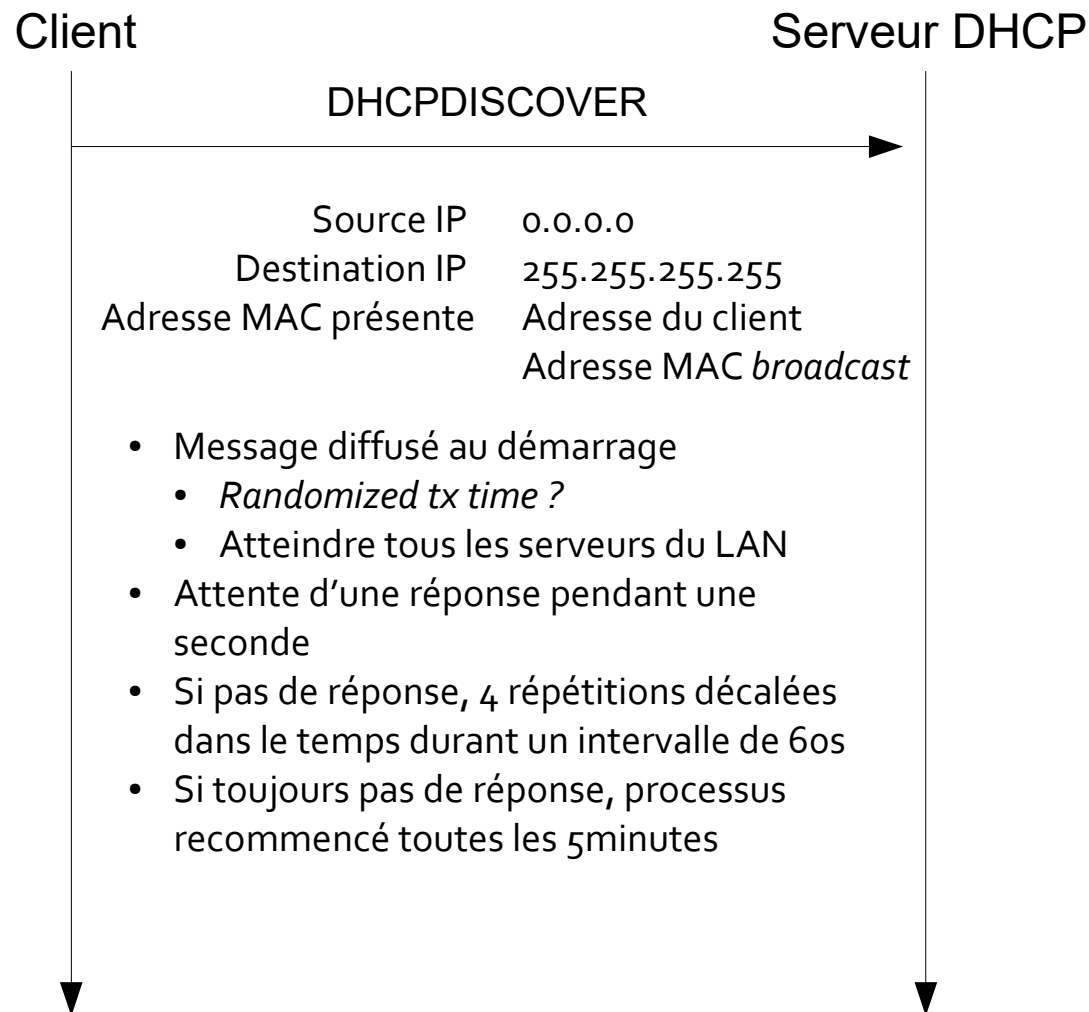
Quelques options :

- Masque de sous-réseau
- Adresse IP du serveur DNS
- Nom de la station
- Paramètres IP, TCP, ARP (MTU, TTL...)
- Adresse du routeur par défaut
- Routes statiques
- Adresse des serveurs de messagerie (SMTP et POP)

# Dynamic Host Configuration Protocol

## Obtention d'un bail

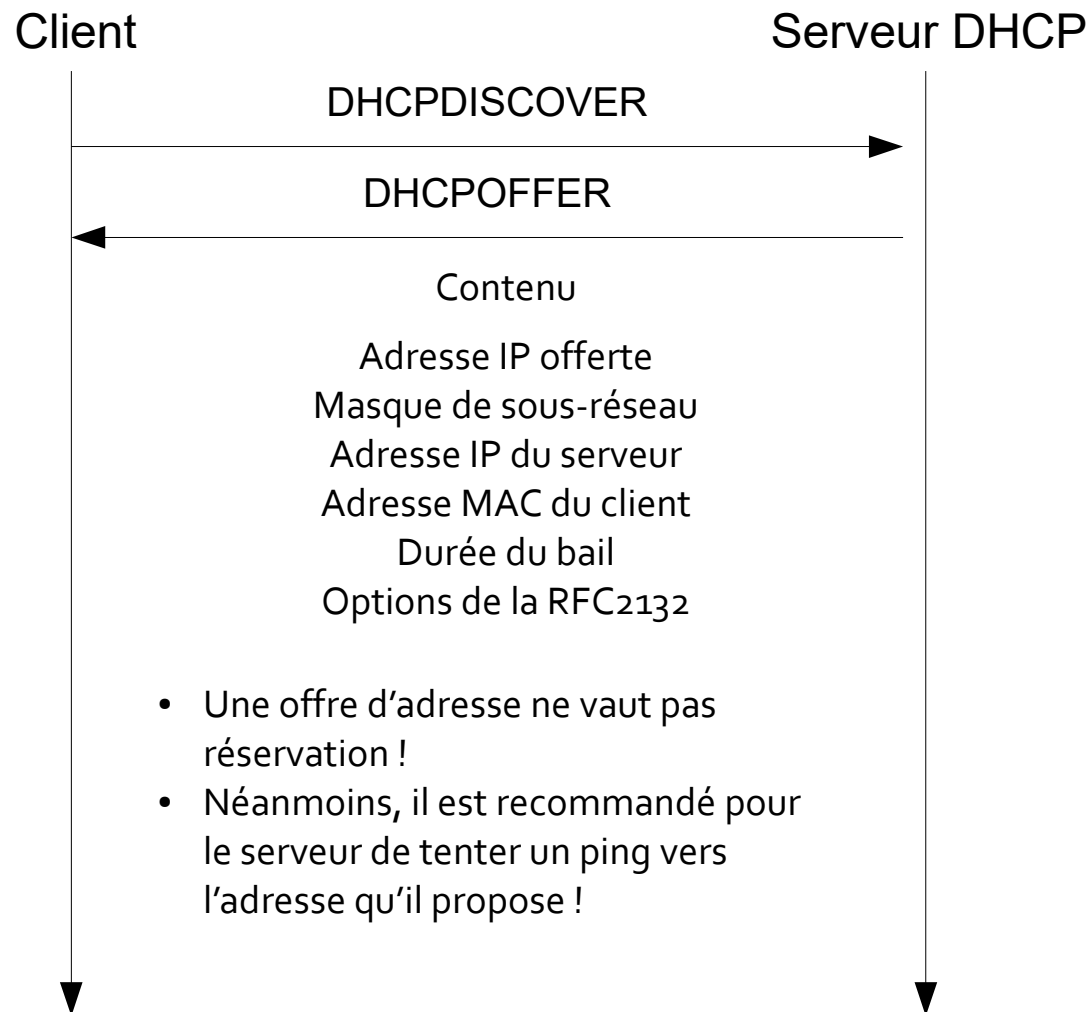
- Processus en 4 étapes
- Protocole de couche transport : UDP
- Diffusion des requêtes et des réponses



# *Dynamic Host Configuration Protocol*

## Obtention d'un bail

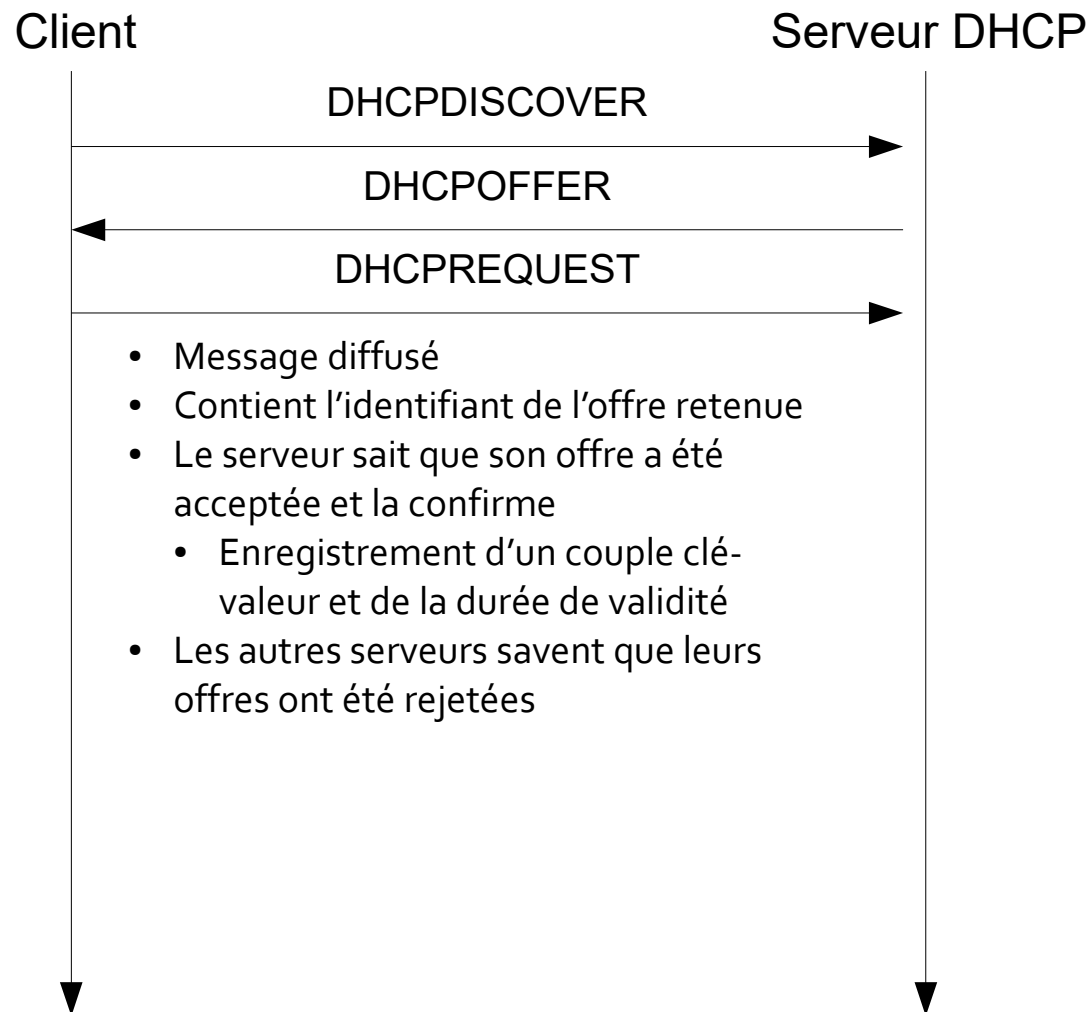
- Processus en 4 étapes
- Protocole de couche transport usuel : UDP
- Diffusion des requêtes et des réponses



# *Dynamic Host Configuration Protocol*

## Obtention d'un bail

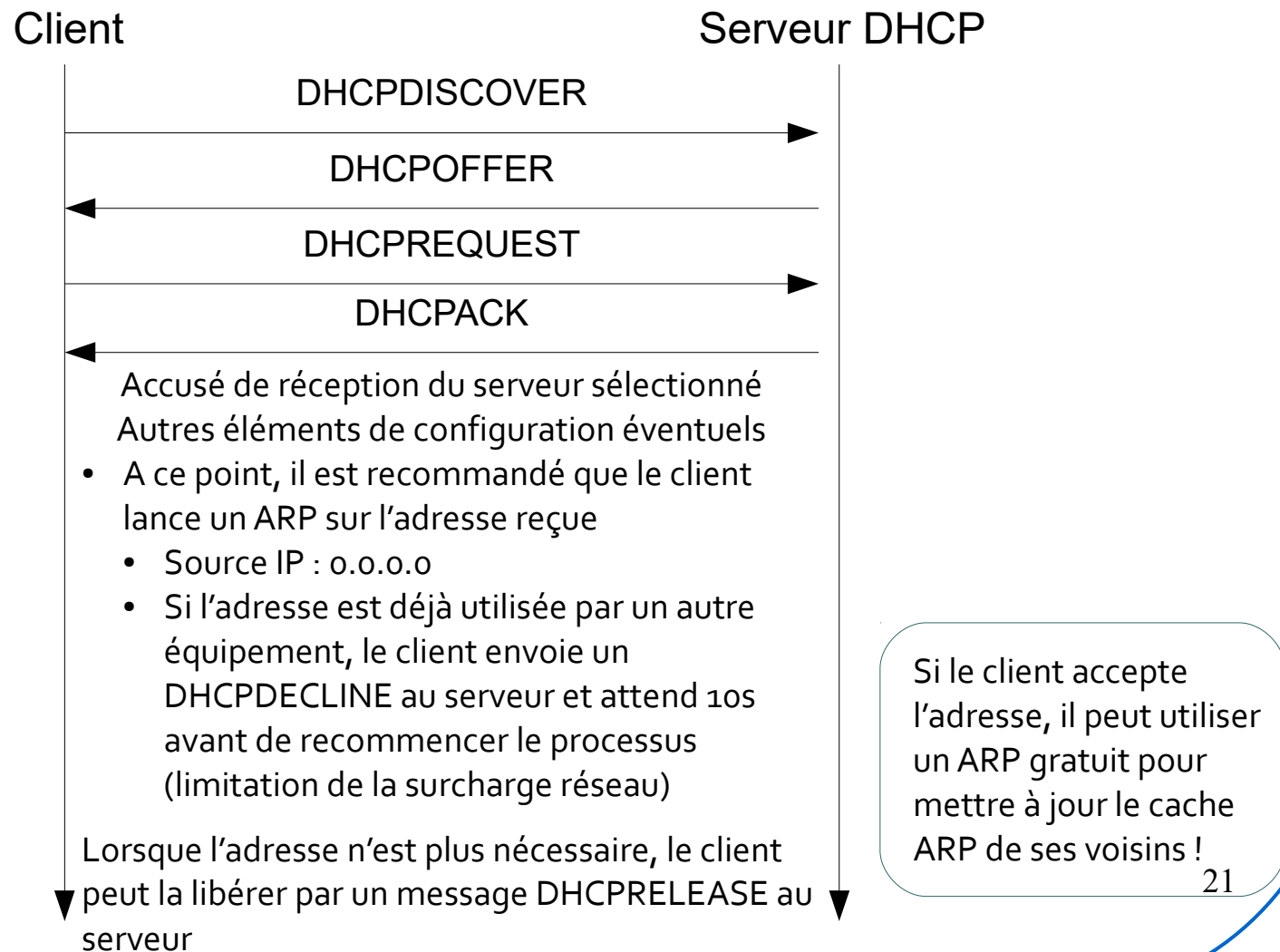
- Processus en 4 étapes
- Protocole de couche transport usuel : UDP
- Diffusion des requêtes et des réponses



# Dynamic Host Configuration Protocol

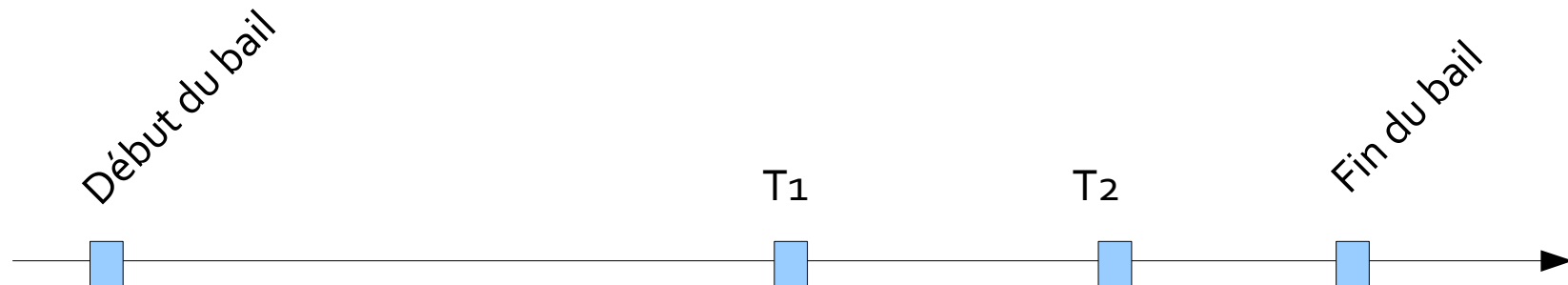
## Obtention d'un bail

- Processus en 4 étapes
- Protocole de couche transport usuel : UDP
- Diffusion des requêtes et des réponses



# Dynamic Host Configuration Protocol

## Renouvellement de bail



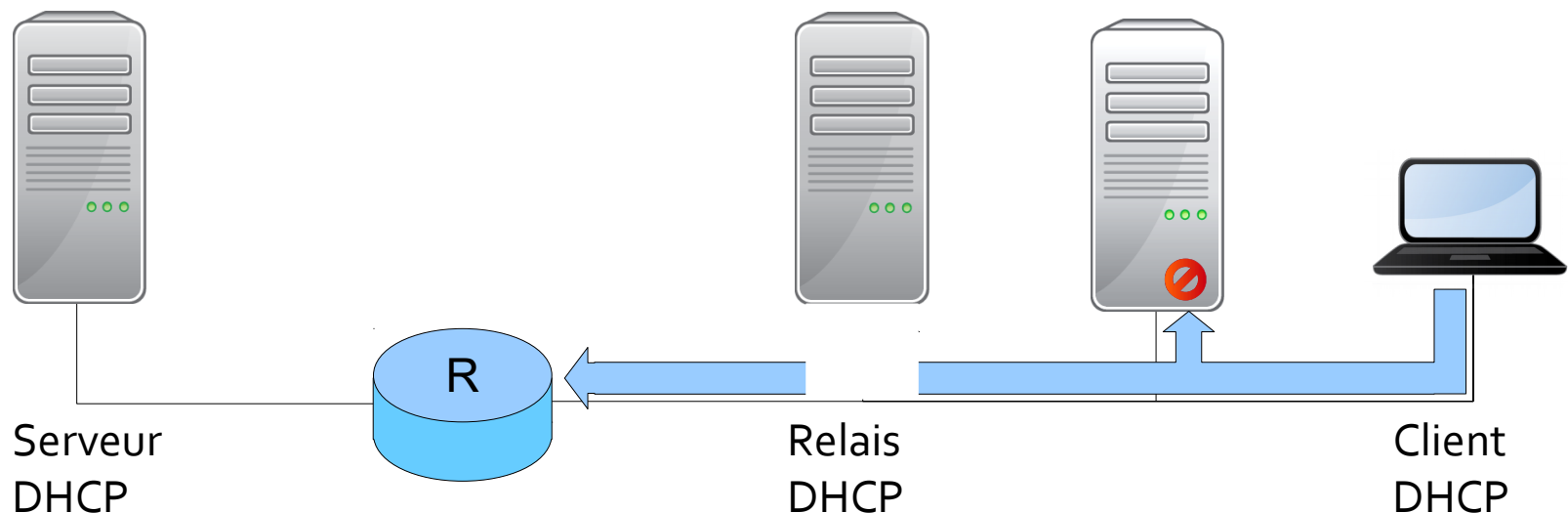
Bail à durée déterminée

- @T1 (50 % du bail par défaut)
  - Tentative de renouvellement du bail existant
  - Demande d'une extension au serveur spécifique
    - DHCPREQUEST en unicast
  - Si DHCPACK reçu avant T2, le client conserve son adresse jusqu'à une nouvelle date d'expiration
- @T2 (87,5 % du bail par défaut)
  - Tentative d'obtention d'un nouveau bail
  - *Broadcast* à tous les serveurs potentiels

# Dynamic Host Configuration Protocol

## Relais DHCP

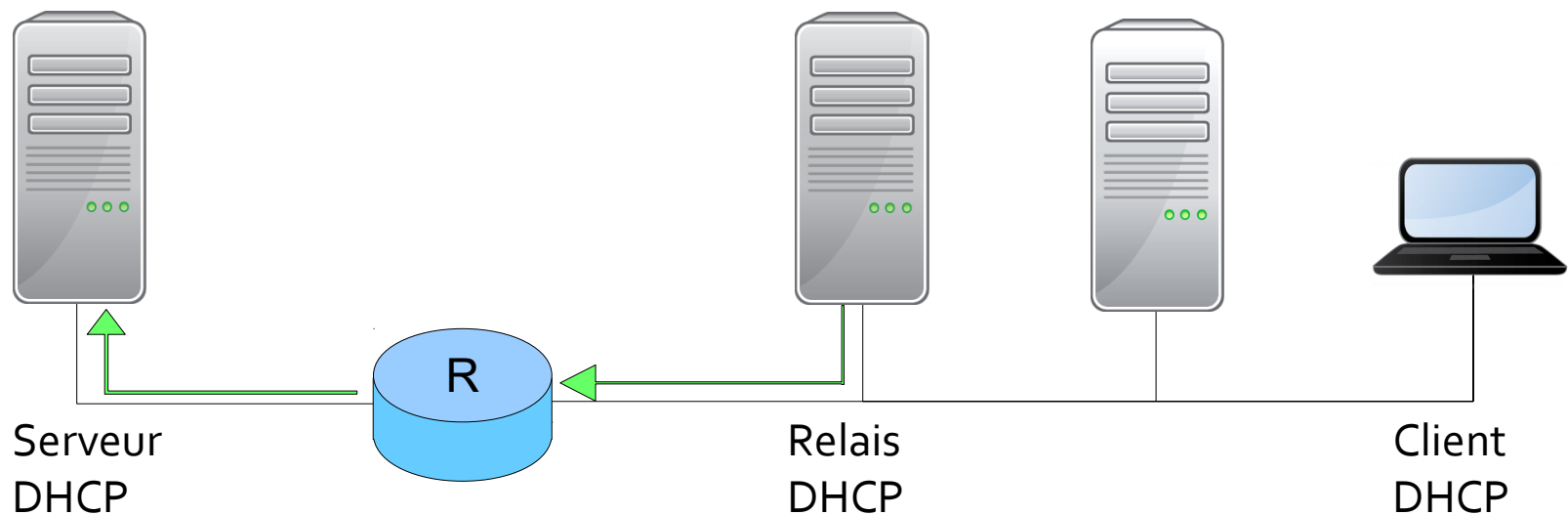
- Comment se comporte un routeur face à du trafic *broadcast* de niveau 2 ?
- Nécessité d'un équipement qui joue le rôle d'intermédiaire
  - Connaît le serveur DHCP
    - Communique en *unicast* avec le serveur
  - Dispose d'une adresse IP statique
  - Relais les messages du client au serveur
- Positionnement du relais DHCP et routeur



# Dynamic Host Configuration Protocol

## Relais DHCP

- Comment se comporte un routeur face à du trafic *broadcast* de niveau 2 ?
- Nécessité d'un équipement qui joue le rôle d'intermédiaire
  - Connaît le serveur DHCP
    - Communique en *unicast* avec le serveur
  - Dispose d'une adresse IP statique
  - Relais les messages du client au serveur
- Positionnement du relais DHCP et routeur





## Correction

Quelle est la différence entre le contrôle de flux et le contrôle de congestion ?  
(qui sont les acteurs ? Où se manifeste le problème?)

Comment le contrôle de flux est-il réalisé dans TCP ?

Que se passe-t-il si l'un des partenaires d'une connexion TCP, implémentée par exemple en Java, ferme la connexion sans notifier l'autre ?

A quoi bon la route par défaut dans la table de routage ?

Expliquez en vos propres termes le point 3.3 de la RFC 2131 : quel est le protocole concerné ? Quel est le problème adressé ?

Bonus : j'ai acheté une adresse de réseau de classe C pour définir 402 sous-réseau : ai-je pris une bonne décision ? Pourquoi ?