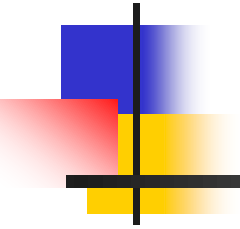


# ADMN Réseaux

## Séance 3

---





# Contenu du cours

---

1. Introduction (1h30)
2. Introduction (1h30)
3. Couches Physique et Liaison (1h30)
4. Ethernet (1h30)
5. Couche Réseau (3h)
6. Couche Transport (3h)
- 7. Couche Application (6h)**



# Contenu de la séance

---

## Couche Application (3/4)

- Protocoles d'administration réseaux (suite)
  - SNMP
- Protocoles applicatifs :
  - DNS
- Protocoles d'auto-configuration réseaux
  - BOOTP
  - DHCP



# Couche Application (3/4)

---

Protocoles applicatifs



# 1. DNS

---

- **DNS : Domain Name System**
- le système de noms de domaines fournit la correspondance entre un **nom de machine** et son **numéro IP**.
- Le mécanisme qui permet la résolution d'un nom en une adresse IP est géré par des serveurs de noms qui représentent **une base de données distribuée des noms de domaine**
- En France : <http://www.afnic.fr/>
- **RFC 1034 et 1035**



# DNS : le contexte Internet

---

- Adresse d'un serveur
  - Adresse IP de la machine sur laquelle il tourne
  - numéro de port (TCP ou UDP)
  - en général numéro bien connu pour chaque type de serveur
- Inconvénient
  - Adresse IP difficile à mémoriser par un humain
- Idée
  - remplacer la dresse IP par un nom de machine
  - plus facile pour les humains
  - mais adresse IP nécessaire pour utiliser TCP ou UDP...
- Comment traduire un nom en une adresse IP ?



# DNS : le contexte Internet

---

- Fichier `"/etc/hosts"`
  - contient la table adresse IP - nom de machine
  - doit être distribué régulièrement
- inutilisable dans un grand réseau

```
#  
# Internet host table  
#  
127.0.0.1 localhost  
138.48.32.99 babbage  
  
138.48.32.100 leibniz  
138.48.32.1 routeur  
138.48.26.1 bab-fast  
138.48.26.2 leib-fast  
138.48.32.92 corneille  
138.48.32.107 backus  
138.48.20.152 arzach  
138.48.32.137 almin01  
138.48.32.170 duke
```



# DNS : le contexte Internet

---

- **Problème à résoudre**
  - Éviter que deux machines différentes aient le même nom
- **Comment attribuer les noms de machines ?**
  - Choisi par l'utilisateur
    - risque d'avoir plusieurs fois le même nom
    - exemple : pc1, mac2, sun3, ...
  - Au hasard
    - exemple : GHTYUIOKSY
    - pas très user-friendly
  - De façon hiérarchique
    - nom d'une machine composé de deux parties
      - 1. nom d'une machine composé de deux parties
      - 2. nom du domaine
      - 3. exemple : www.belnet.be



# DNS : le contexte Internet

---

- Comment garantir l'unicité des noms hiérarchiques ?
  - Attribuer les noms de domaine de façon hiérarchique
    - domaines de premier niveau
      - exemple : com, edu, net, gov, mil, int,..., be, fr, de, uk,...
    - chaque domaine de premier niveau peut déléguer des sous-domaines
      - exemple : lip6.fr, paris13.fr, paris11.fr...
    - chaque sous-domaine peut déléguer des sous-sous domaines ou des noms de machines
      - exemple : www.lip6.fr,...
- Garantir l'unicité à chaque niveau pour unicité globale



# DNS : Domain Name System

---

- Objectif : convertir des noms littéraux des hôtes de l'Internet en adresses numériques (Adresse IP)
- annuaire standard de l'Internet (RFC 1034 et RFC 1035)



# DNS : Domain Name System

---

- Historique

- Jusqu'en 1988, le service de noms de l'Internet était centralisé
  - Acceptable pour quelques milliers de machines, mais ne passe pas à l'échelle (cf plus loin)
- 1987-88 : Conception et mise en place de DNS
  - Documents : RFC 1034 et 1035 + compléments ultérieurs
    - Voir <http://www.dns.net/dnsrd/>
- Depuis, a démontré sa capacité de croissance : de x.103 à x.108...



# DNS : Domain Name System

---

- serveurs de noms (serveurs DNS)
  - base de la hiérarchie des serveurs liés à la base de données distribuée
  - gèrent les requêtes DNS
  - transport sur UDP ou TCP, port 53
  - les applications y accèdent à travers le resolver (UNIX)
    - gethostbyname
    - gethostbyaddr



# DNS : Domain Name System

---

- Les «resolvers» sont les processus clients qui contactent les serveurs de nom
- Fonctionnement :
  - contacte un name serveur (dont l' (les) adresse(s) est (sont) configurées sur la machine exécutant ce resolver)
  - interprète les réponses
  - retourne l'information au logiciel appelant
  - gestion de cache (dépend de la mise en œuvre)
  - Le serveur de nom interroge également d'autres serveurs de nom, lorsqu'il n'a pas autorité sur la zone requise (fonctionnement itératif ou récursif)
  - Si le serveur de nom est en dehors du domaine requis, il peut être amené à contacter un serveur racine ( ne pas confondre avec un domaine racine)



# DNS: Domain Name System

---

## Domain Name System:

- *Base de données distribuées* implémentée dans une hiérarchie de serveurs de noms
- *Protocole applicatif*
  - hôtes, routeurs, serveurs de noms qui communiquent pour effectuer la traduction
  - DNS utilisé par d'autres protocoles applicatifs
  - La complexité est repoussée à la bordure du réseau



# DNS : espace de "nommage"

---

- **Systeme de nommage hiérarchique**
  - structure arborescente (~système de fichier Unix)
  - label d'un noeud : 63 caractères (majuscules ou minuscules)
  - domain name = liste des labels en parcourant l'arbre vers la racine (". " séparateur)
    - relatif : pc24.CS.keio
    - absolu (FQDN) : pc24.CS.keio.ac.jp.



# DNS : espace de "nommage"

- La désignation sur l'Internet utilise un schéma hiérarchique (noms de domaines)
  - Exemples : thales.e.ujf-grenoble.fr, cmu1.acs.cmu.edu, [www.w3.org](http://www.w3.org), firewall.ma02.bull.com
- Un domaine est une unité pour la construction et la gestion des noms (analogue à un répertoire dans un système de fichiers)

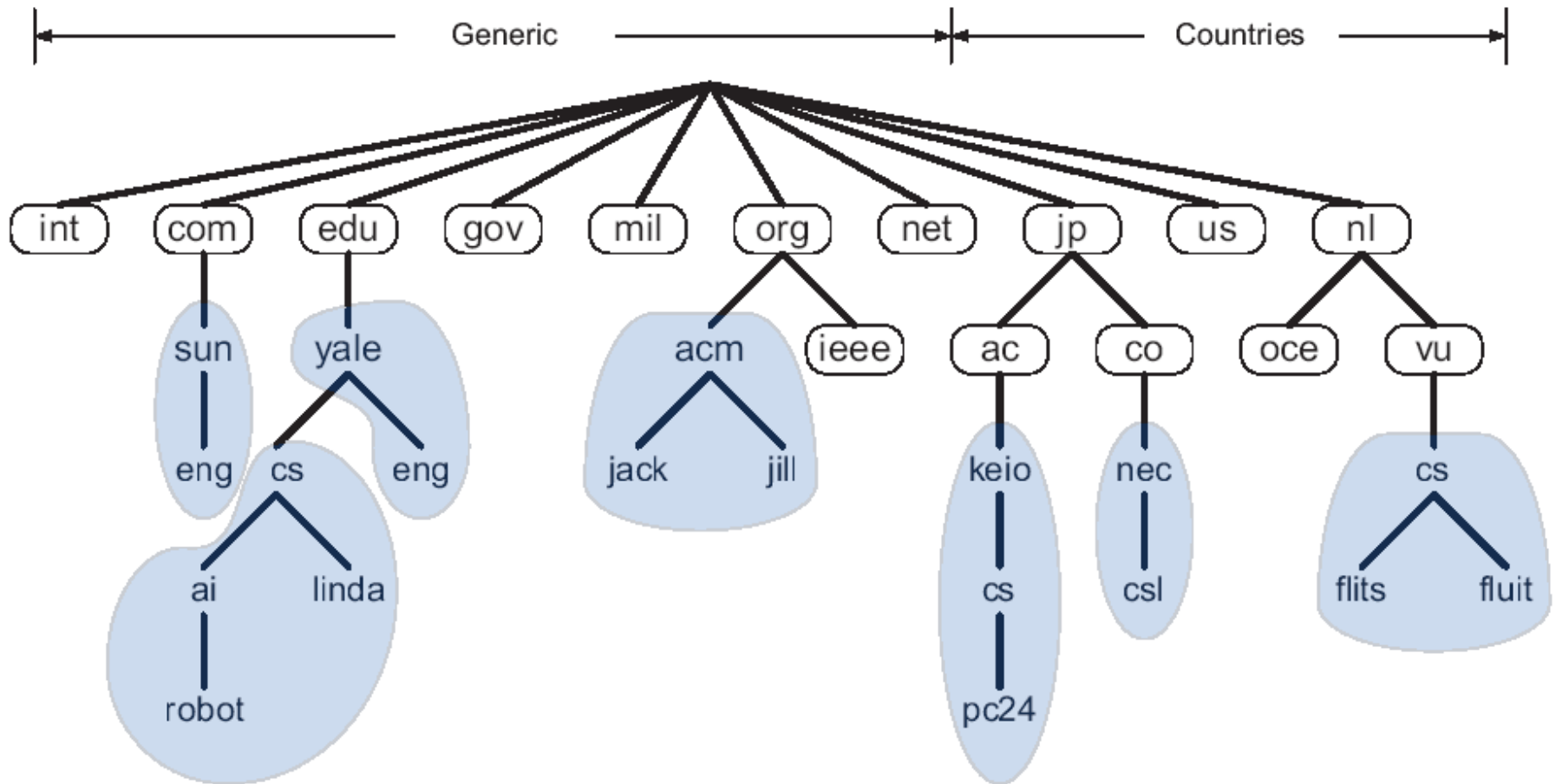


# DNS : espace de "nommage"

---

- La hiérarchie des noms a une racine multiple : les domaines du premier niveau
  - domaines "géographiques" : fr (France), uk (Royaume Uni), de (Allemagne), jp (Japon), ...
    - USA est souvent compris "par défaut", bien qu'il existe un domaine us
  - domaines "d'activité", génériques : com (commercial), org (organisations), edu (universités, aux USA), net (l'Internet), ...

# DNS : Zones





# DNS : espace de "nommage"

---

- Dans chaque domaine, les noms sont attribués par une autorité responsable du domaine
  - Pour établir la liste des domaines du premier niveau : l'Internet Society (ISOC) via un groupe technique ad hoc
  - Pour les domaines "publics" du premier niveau (com, org, net, ...) : une autorité centrale, l'ICANN ([www.icann.org](http://www.icann.org)), avec des autorités déléguées
  - Pour les domaines géographiques : une autorité nationale par pays - en France, l'AFNIC ([www.nic.fr](http://www.nic.fr)) - Association Française pour le Nommage sur l'Internet en Coopération)
  - Pour les domaines inclus : autorités locales (entreprise, administration, etc.)



# DNS : TLD

- **Top Level Domain**

- ICANN (Internet Corporation for Assigned Names and Numbers)
- partage du premier niveau et délégation à des registrars

gTLD	intro.	description	operator
.aero	2001	Air-transport ind.*	SITA
.biz	2001	Businesses	NeuLevel
.com	-	Unrestricted	VeriSign
.coop	2001	Cooperative *	DotCooperative
.edu	-	US educ. inst. *	EDUCAUSE
.gov	-	US government	US Admin.
.info	2001	Unrestricted	Afilias
.int	1998	Int organisations	ICANN
.mil	-	US military *	US DoD NIC
.museum	2001	Museums *	MuseDoma
.name	2001	Individuals	GNR
.net	-	Unrestricted	VeriSign
.org	-	Unrestricted	PIR
.pro	2001	Professionals	RegistryPro
.arpa	-	<b>special LTD</b>	ICANN

ccTLD ISO 3166	240 Countries and external territories
.ac	Ascension Island
.ad	Andorra
.ae	United Arab Emirates
.af	Afghanistan
...	...
.fr	France
...	...
.uk	United Kingdom (gb)
.us	United States
...	...
.yu	Yugoslavia
.za	South Africa
.zm	Zambia
.zw	Zimbabwe



# DNS : Zones

---

- Sous-arbre de l'arbre DNS administré séparément
  - (~partitions physiques d'un système de fichier Unix)
  - délégation des noms de sous-domaines correspondants
    - exemple : keio.jp.ac.
  - des serveurs de noms y sont associés



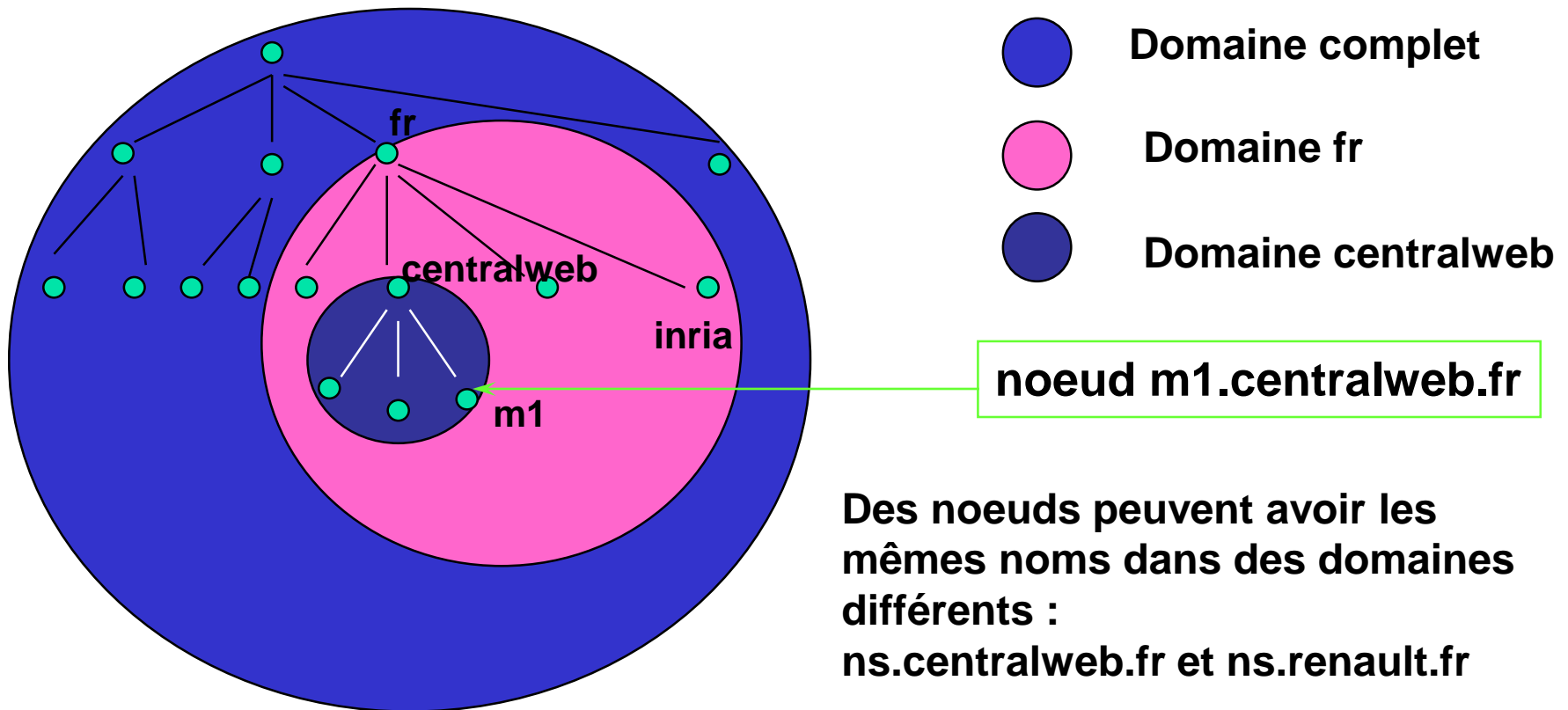
# DNS : serveurs de noms

---

- Domaines et zones
  - Domaine = unité de désignation (espace de noms)
  - Zone = unité de gestion administrative (serveur de noms propre à la zone)
- Le plus souvent, un domaine est aussi une zone, mais une zone peut grouper plusieurs domaines, administrés en commun

# Le domaine

Un domaine est un sous-arbre de l'espace nom de domaine



Des noeuds peuvent avoir les mêmes noms dans des domaines différents :  
ns.centralweb.fr et ns.renault.fr



# Lecture des noms de domaine

- A l'inverse de l'adressage IP la partie la plus significative se situe à gauche de la syntaxe :

sun2.ethernet1.centralweb.fr

193.148.37.201

←  
**vers le plus significatif**

→  
**vers le plus significatif**

**sun2. ethernet1. centralweb.fr**

→ **domaine français (.fr)**

→ **domaine de l'organisation CentralWeb**

→ **sous-domaine CentralWeb**

→ **machine sun2 du domaine ethernet1. centralweb.fr**



# DNS : serveurs de noms

---

- **Différents types de serveurs de noms dans une zone**
- un primaire (primary name server)
  - information autoritaires sur la zone (authoritative record)
  - initialisation local (disque)
- un ou plusieurs secondaire (secondary name server)
  - redondance : indépendant du primaire (voire même distant)
  - initialisation : m-à-j. à partir du primaire (transfert de zone)
- requêtes récursives ou itératives
- utilisation de cache



# DNS : serveurs de noms

---

- Aucun serveur n'a toutes les relations nom-vers-@IP
- **Serveurs de noms locaux:**
  - Chaque ISP ou entreprise a son propre (*default*) *name server*
  - Les requêtes DNS vont en premier au serveur de nom local
- **Serveurs de noms racines (Root Name Servers):**
  - Il existe une douzaine de root name servers dans l'Internet
- **Serveurs de noms "authoritative":**
  - Chaque hôte est enregistré auprès d'un serveur "authoritative", qui stocke son adresse IP et son nom
  - Peut effectuer la traduction nom/adresse pour cet hôte



# DNS : Root name servers

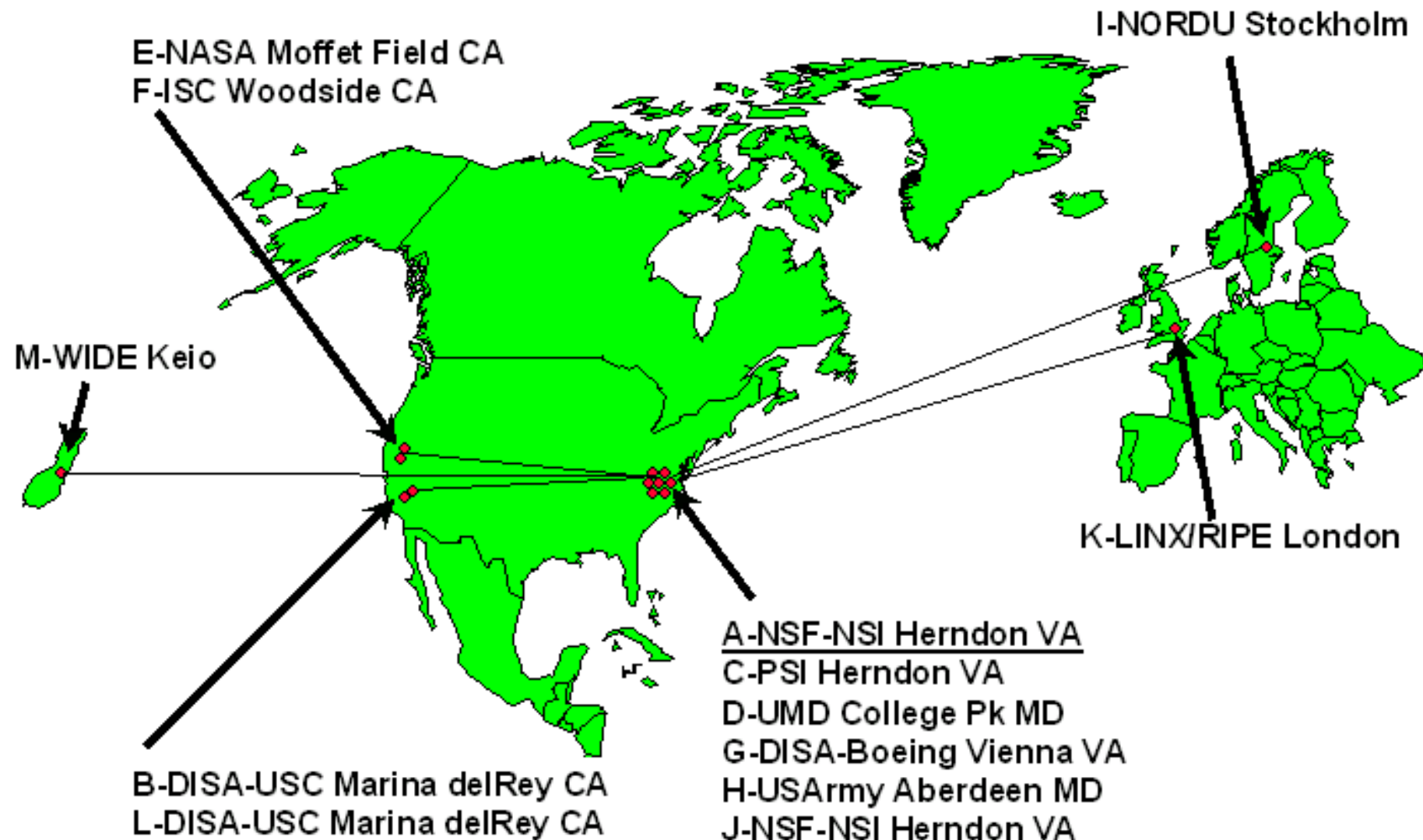
---

- Contactés par les serveurs de noms locaux qui n'arrivent pas à résoudre ce nom
- Serveur de nom racine :
  - Contacte le serveur de nom "authoritative" si la correspondance nom/adresse IP n'est pas connue
  - Obtient la correspondance
  - Renvoie la correspondance au serveur de noms local
- ~ une douzaine de serveurs de noms racines dans le monde

# DNS Root Servers

1 Feb 98

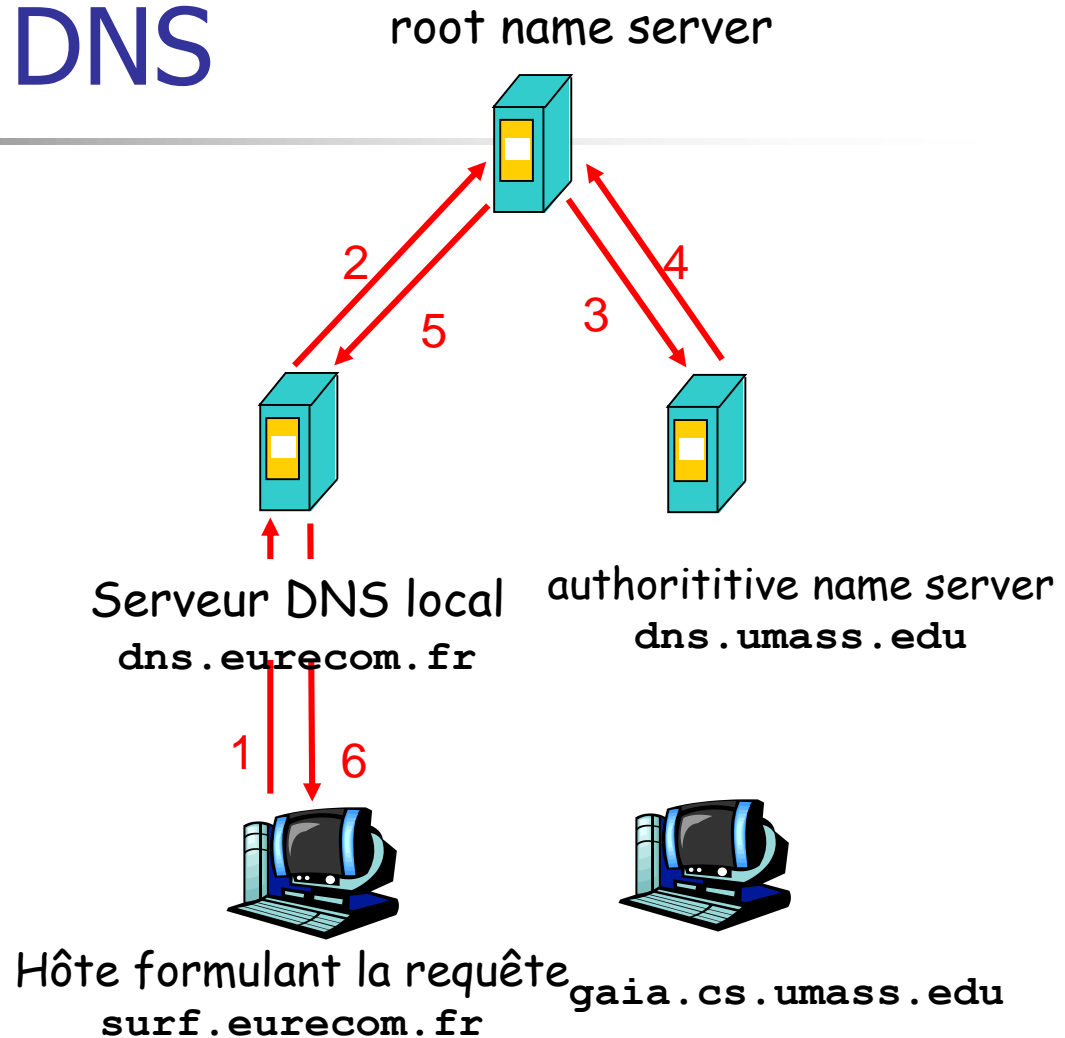
## Designation, Responsibility, and Locations



# Exemple de DNS

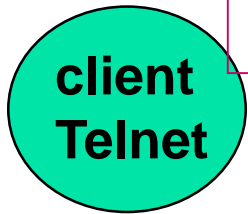
L'hôte **surf.eurecom.fr** veut connaître l'adresse IP de **gaia.cs.umass.edu**

1. Contacte son serveur DNS local, **dns.eurecom.fr**
2. **dns.eurecom.fr** contacte le serveur de noms racine, si nécessaire
3. le serveur de noms racine contacte le serveur de nom "authoritative", si nécessaire



# Principe (illustration)

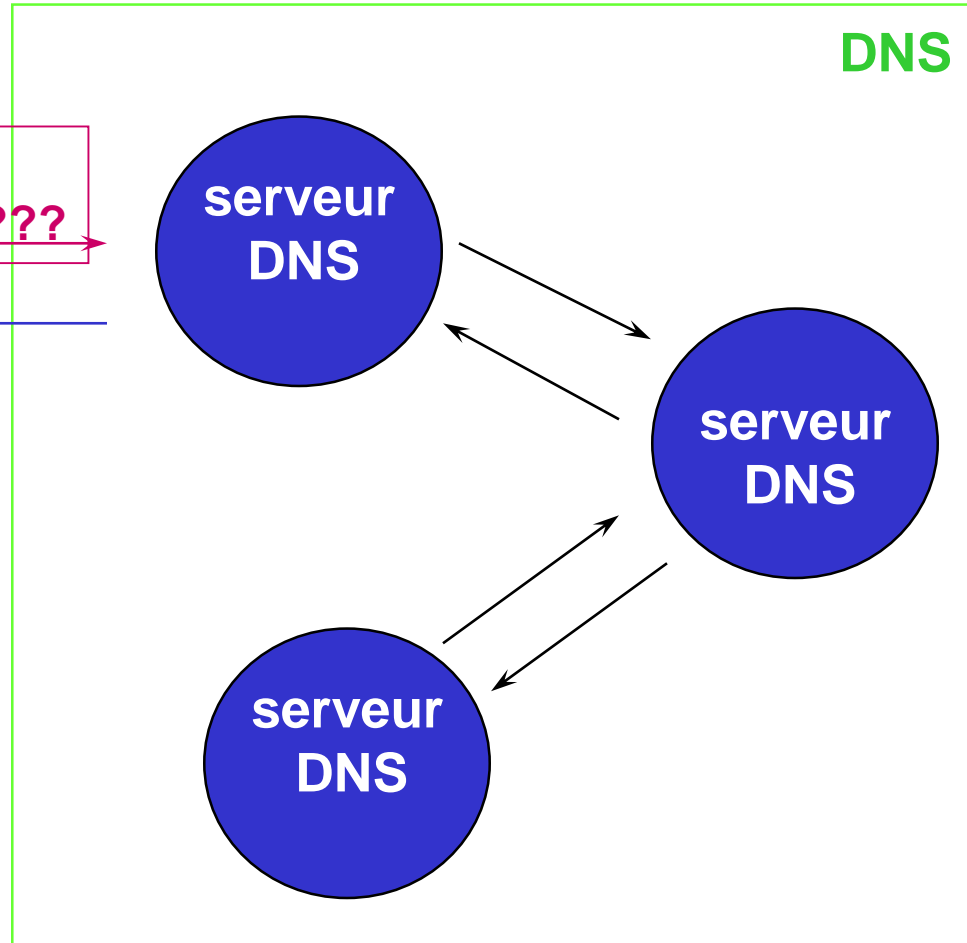
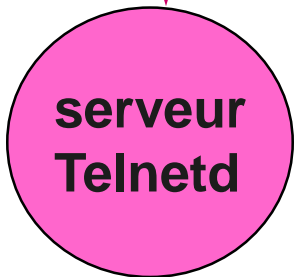
\$ telnet m1.centralweb.fr



Demande de résolution  
m1.centralwebfr ????

Réponse  
193.148.37.201

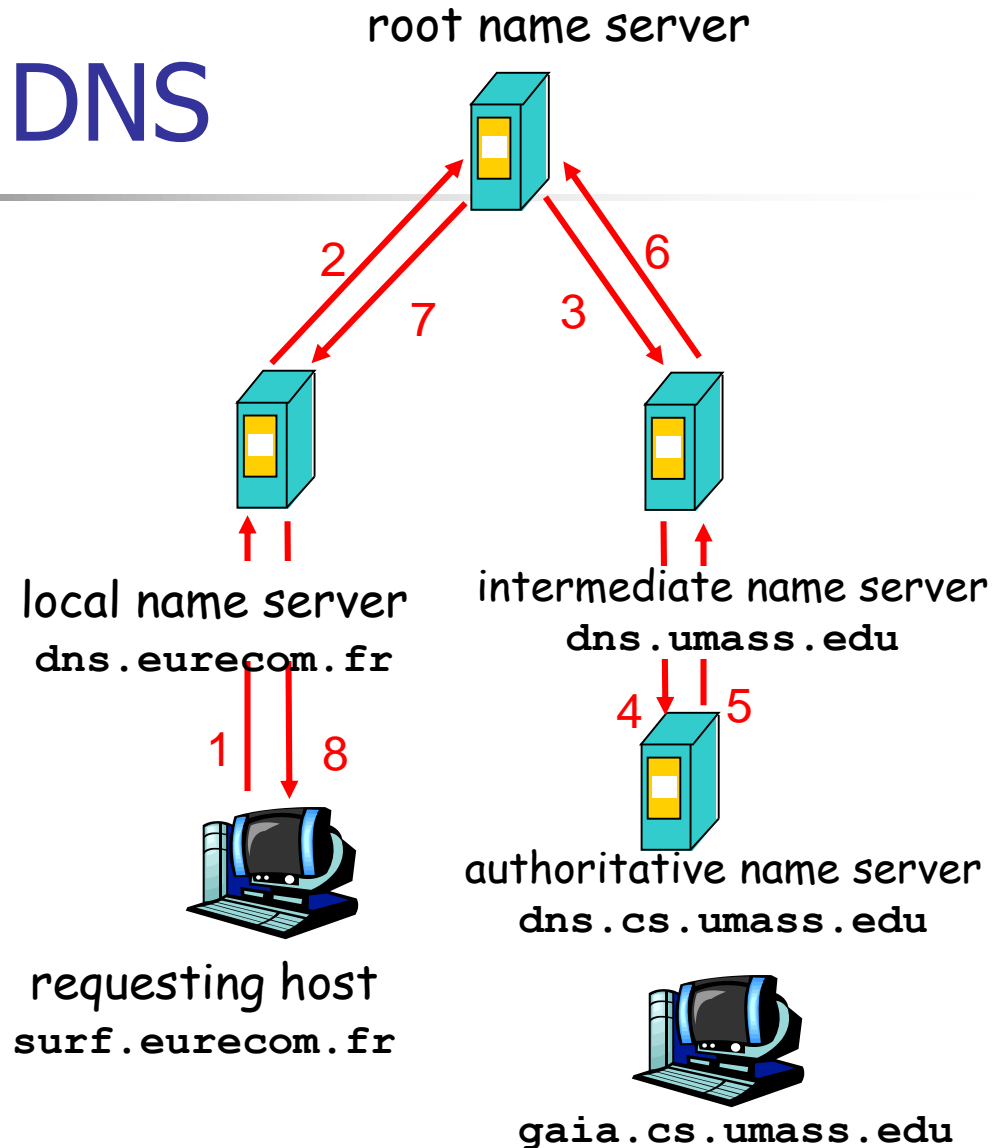
193.148.37.201



# Exemple de DNS

Le serveur de noms racine :

- Ne connaît pas forcément le serveur de noms authoritative
- Peut connaître un **serveur de noms intermédiaire**, à contacter pour trouver le serveur de noms authoritative



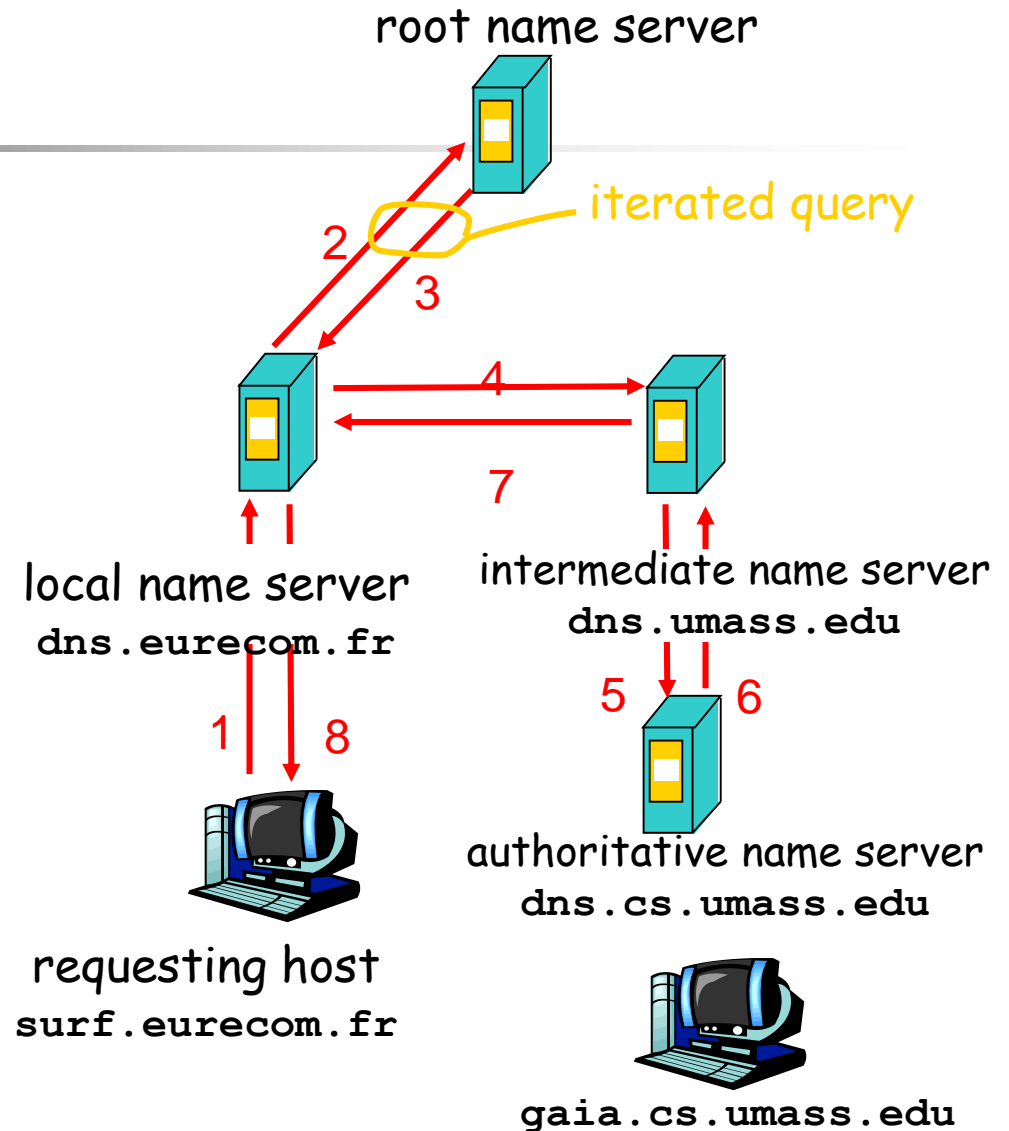
# DNS: Requêtes itératives

## Requête récursive :

- Confie la tâche de la résolution de nom au serveur de noms contacté

## Requête itérative :

- Le serveur de noms contacté fournit en réponse le nom du serveur à contacter
- "Je ne connais pas ce nom, mais demande à ce serveur"





# Cache DNS

---

- Une fois qu'un serveur de noms (quelconque) apprend une nouvelle correspondance nom/adresse IP, il stocke cette correspondance dans son **cache**
  - Les données du cache expirent (disparaissent) après un certain temps
- Mécanismes de mise à jour et de notification à l'étude à l'IETF
  - RFC 2136
  - <http://www.ietf.org/html.charters/dnsind-charter.html>



# Enregistrements DNS

DNS : BD distribuée stockant des enregistrements de Ressources (Resource Records - **RR**)

RR format: (nom, valeur, type, TTL)

- Type=A
  - Nom = hostname
  - Valeur = adresse IP
- Type=NS
  - Nom = domaine (ex. foo.com)
  - Valeur = adresse IP du serveur de nom d'origine pour ce domaine
- Type=CNAME
  - Nom = alias à la place d'un nom "canonique" (vrai nom)
  - Valeur = nom canonique
- Type=MX
  - Valeur = hostname du serveur de mail associé au nom

# DNS : protocole, messages

Protocole DNS : messages de **requête** et de **réponse**, avec le **même format de message**

## En-tête des messages

- **Identification** : numéro de 16 bits pour la requête, la réponse à cette requête utilise le même numéro
- **Fanions** :
  - Requête ou réponse
  - Récursion souhaitée
  - Récursion disponible
  - Réponse autoritative

identification	flags
number of questions	number of answer RRs
number of authority RRs	number of additional RRs
questions (variable number of questions)	
answers (variable number of resource records)	
authority (variable number of resource records)	
additional information (variable number of resource records)	



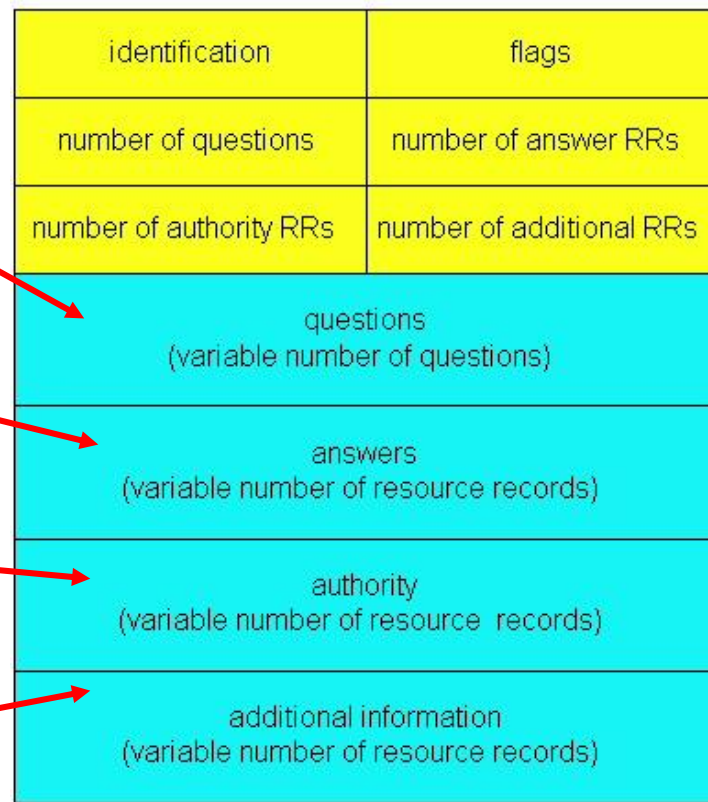
# DNS : protocole, messages

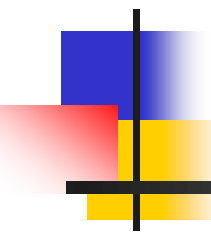
Nom, champs de type  
pour la requête

RRs dans la réponse  
à la requête

enregistrements pour  
les serveurs authoritative

Information "utile"  
additionnelle





# Protocoles d'auto- configuration réseau des hôtes

---



# Configuration des paramètres réseau

---

- Paramètres réseau nécessaires à la configuration d'une machine :
  - Adresse IP
  - Masque du sous réseau
  - Adresse IP du routeur
  - Adresse IP du serveur DNS
  - Etc..
- Possibilités de configuration
  - Statique (à la main) : fichier de config sur le disque local
  - Dynamique (depuis un serveur)
- Comment demander à un serveur l'envoi des informations de configuration si la machine n'a (ne connaît) pas son adresse IP ?



# Protocoles pour l'autoconfiguration

---

- Basique (jusqu'à 1985) : Reverse ARP
  - Retourne l'adresse IP correspondant à une adresse MAC
  - Ne peut fournir que l'adresse IP !
- Mieux (1985-1993) : BOOTP
  - Fourni plus de paramètres de configuration réseau
- Le meilleur & plus utilisé (depuis 1993) : DHCP
  - Infos de config à durée de vie limitée



# Le protocole BOOTstraP

---



# BOOTP

---

- BOOTP (Bootstrap Protocol ) est un protocole de démarrage de terminaux X ou stations sans disque qui utilise UDP comme couche de transport et est généralement associé à TFTP ou NFS
  - BOOTP peut servir à démarrer un serveur, un terminal X en renvoyant le nom du fichier de démarrage qui sera récupéré par TFTP.
  - BOOTP n'accepte et ne traite que la première réponse.



# BOOTP

---

- Comme RARP, BOOTP sert principalement à fournir son adresse IP à une machine que l'on démarre sur un réseau.
  - Mais RARP passe par des protocoles de niveau 1, non routables. De plus seule l'adresse IP est récupérée.
  - BOOTP marche au niveau IP/UDP, il est donc moins lié au type de matériel du réseau et transmet plus d'information que RARP.
  - BOOTP utilise deux ports UDP : le port serveur 67 et le port client 68.



# BOOTP (RFCs 951, 1542, 2132)

---

- Remplaçant de RARP
  - Renvoi de l'adresse IP
  - + Amorçage (Bootstrapping)
    - Amorçage du système d'exploitation (image) et des paramètres de configuration depuis un serveur
    - Utile pour les machines ne possédant pas de capacité de stockage (i.e. disques durs)
  - Repose sur le principe Client/Serveur
    - Client : Utilise (envoi et réception) le port 68
    - Serveur : Utilise (envoi et réception) le port 67 et possède 1 adresse IP statique
  - Communication en mode datagramme : UDP



# BOOTP

0	8	16	24
Code Opération (1 requête , 2 réponse )	Type de matériel 1=ETHERNET	Longueur adresse matérielle 6 si ETHERNET	compteur de saut (0 en général sauf routeur)
Identificateur de transaction (tiré au hasard , envoyé et renvoyé tel que)			
Nombre de secondes		Non utilisé	
adresse IP du client (souvent 0.0.0.0)			
votre adresse IP (renvoyée par serveur)			
Adresse IP du serveur (rare)			
adresse IP du routeur (si un routeur route la demande)			
adresse matérielle du client (16 octets) (émise et retournée)			
nom de machine du serveur (64 octets) si boot			
nom du fichier de démarrage (128 octets) si boot			
<b>information spécifique (64 octets) (retour des infos)</b>			



# BOOTP

---

- Pour démarrer, le client fait un broadcast ETHERNET avec dans cette trame, comme adresses IP 0.0.0.0 , destination 255.255.255.255 et remplit l'adresse matérielle, port 67.
- Le serveur renvoie sans broadcast la réponse sur la machine. Elle évite de faire un ARP pour renvoyer la réponse car le client ne connaît pas encore son adresse. Le serveur ajoute « à la main » l'entrée dans la cache.



# BOOTP : Principes

---

- Le client-BOOTP envoie une requête au serveur-BOOTP :
  - Pb : Le client ne connaît pas l'adresse IP du serveur ni la sienne :
    - utilise 255.255.255.255 comme adresse de destination
      - Diffusion limitée seulement sur le LAN !
    - et 0.0.0.0 comme adresse source
- Le serveur consulte sa base de données :
  - Retourne l'@ IP correspondant à l'@ MAC source



# BOOTP : Principes

---

- Diffusion des requêtes du client-BOOTP limitée seulement sur le LAN
  - Peut-on contacter des serveurs-BOOTP localisés dans d'autres réseaux locaux ?
    - Oui : Les routeurs ou les équipements réseau doivent être configurés pour être des agents de relais BOOTP
    - e.g. configurer le routeur pour forwarder les paquets UDP spécifiques : @s 255.255.255.255, ports 68 & 67



# Informations d'amorçage

---

1. Le client-BOOTP envoie une requête (diffusion limitée) au serveur-BOOTP :
  2. Le serveur renvoie les informations d'amorçage requis, typiquement :
    1. L'@ IP d'une machine qui contient les fichiers d'amorçage (bootfile) requis (image + configuration)
    2. Le nom des fichiers d'amorçage
  3. Le client récupère les fichiers d'amorçage grâce à TFTP
- 
- La séparation entre les tâches BOOTP et TFTP signifie :
    - Serveur BOOTP n'a à maintenir qu'une petite base de données
    - Les fichiers image et de configuration peuvent être stockés sur n'importe quelle autre machine

# Format des messages BOOTP

Code opération (1 octet)	Type de matériel (1 octet)	Longueur adresse matériel (1 octet)	Compteur de sauts (1 octet)
ID de transaction (4 octets)			
Nombre de secondes (2 octets)		Octets réservés (2 octets)	
Adresse IP du client (4 octets)			
Votre adresse IP (4 octets)			
Adresse IP du serveur (4 octets)			
Adresse IP du routeur (4 octets)			
Adresse matérielle du client (16 octets)			
Nom de machine du serveur (64 octets)			
Nom du fichier de boot (128 octets)			
Options (Longueur variable)			



# Format des messages BOOTP

---

- Code opération :
  - 1 pour BOOTREQUEST, 2 pour BOOTREPLY etc..
- Type de matériel :
  - Types de réseaux
  - Mêmes chiffres que dans ARP (e.g. 1 pour ethernet)
- Longueur adresse matériel :
  - Longueur des adresses physiques (e.g. 6 octets pour ethernet)
- Compteur de sauts :
  - Champ optionnel (pour amorçage multi-serveurs)
  - Initialisé à 0 par le client
  - Incrémenté de 1 par chaque Serveur-BOOTP relayant la requête vers d'autres serveurs.
- ID de transaction :
  - Identification pour l'association Requête-Réponse (identifie le client)
  - Chiffre initialisé aléatoirement



# Format des messages BOOTP

---

- Nombre de secondes :
  - Le temps écoulé (en secondes) depuis que le client a émis sa requête d'amorçage
- Octets réservés :
  - e.g. utilisés par DHCP
- Adresse IP du client :
  - Adresse IP du client, lorsqu'il en a déjà une
- Votre adresse IP :
  - La future adresse IP du client si la requête était 0.0.0.0
- Adresse IP du serveur :
  - Adresse IP du serveur qui contient le fichier image
  - Adresse IP du (prochain) serveur à utiliser



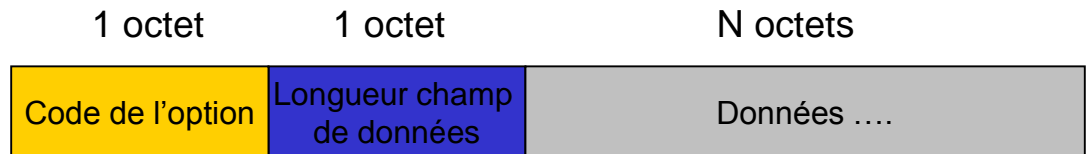
# Format des messages BOOTP

---

- Adresse IP du routeur :
  - Adresse IP de l'agent relais BOOTP (passerelle par exemple) lorsque la connexion directe client/serveur n'est pas possible
- Adresse matérielle du client :
  - Adresse MAC du client
- Nom de machine du serveur :
  - Champ optionnel. Nom du serveur (chaîne de caractères).
- Nom du fichier de boot :
  - Chemin (*path*) et nom du fichier à utiliser pour l'amorçage
- Options :
  - Champ réservé pour les options (voir RFC2132).
  - Première version de BOOTP : champ limité à 64 octets
  - Version actuelle : pas de limites.

# Les options (RFC 2132)

- Le champ d'options commence toujours par le « magic cookie » :
  - 4 octets : 99,130,83 et 99
- Suit par un ensemble de sous-champs d'option
- Chaque sous-champ d'option a le format :

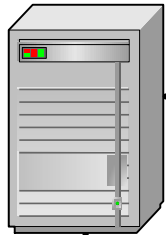


- Plusieurs codes d'options (max : 256):
  - e.g. 15 : nom de domaine du réseau
- Plusieurs options peuvent être envoyées dans le même message
- Toujours finir la zone d'option par l'option 255 (End)
- L'option 57, permet d'imposer au serveur une taille max pour les champs d'options
  - Le serveur peut utiliser les champs *server name* et *file name* pour envoyer le reste des options. Le client en est averti par l'option 52 dans la zone d'option
- Certaines options (e.g. 255) ne possèdent pas de données complémentaires :  
Pas de champs longueur ni données

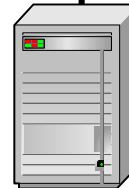
# Exemple

MAC D: ff:ff:ff:ff:ff:ff MAC S: 00:a0:24:71:e4:44
IP D: 255.255.255.255 IP S: 0.0.0.0
UDP Port D: 67 UDP Port S: 68
Code: <b>BOOTREQUEST</b> ID: 77 Client IP: 0.0.0.0 MAC: 00:a0:24:71:e4:44 Your IP: ? Server IP: ? Imagefile: ?

Serveur BOOTP  
*Zeus*  
@MAC: 0b.e2:26:a1:33:72  
@IP: 192.64.30.100



Client



Serveur TFTP

@MAC: 00:b4:28:c3:fe:30  
@IP: 192.64.30.20

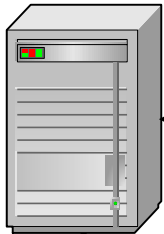
@MAC: 00:a0:24:71:e4:44  
@IP: ?  
Image server: ?  
Image file: ?

# Exemple

MAC D: **ff:ff:ff:ff:ff:ff**  
MAC S: **0b.e2:26:a1:33:72**  
IP D: **255.255.255.255**  
IP S: **192.64.30.100**  
UDP Port D: **67**  
UDP Port S: **68**  
BOOTP: **BOOTREPLY**  
ID: **77**  
Client IP: **0.0.0.0**  
MAC: **00:a0:24:71:e4:44**  
Your IP: **192.64.30.12**  
Server IP: **192.64.30.20**  
Imagefile: **/tftpboot/dl.img**

Serveur BOOTP  
*Zeus*

@MAC: **0b.e2:26:a1:33:72**  
@IP: **192.64.30.100**



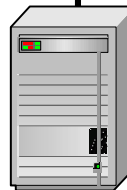
Client

@MAC: **00:a0:24:71:e4:44**

@IP: ?

Image server: ?

Image file: ?

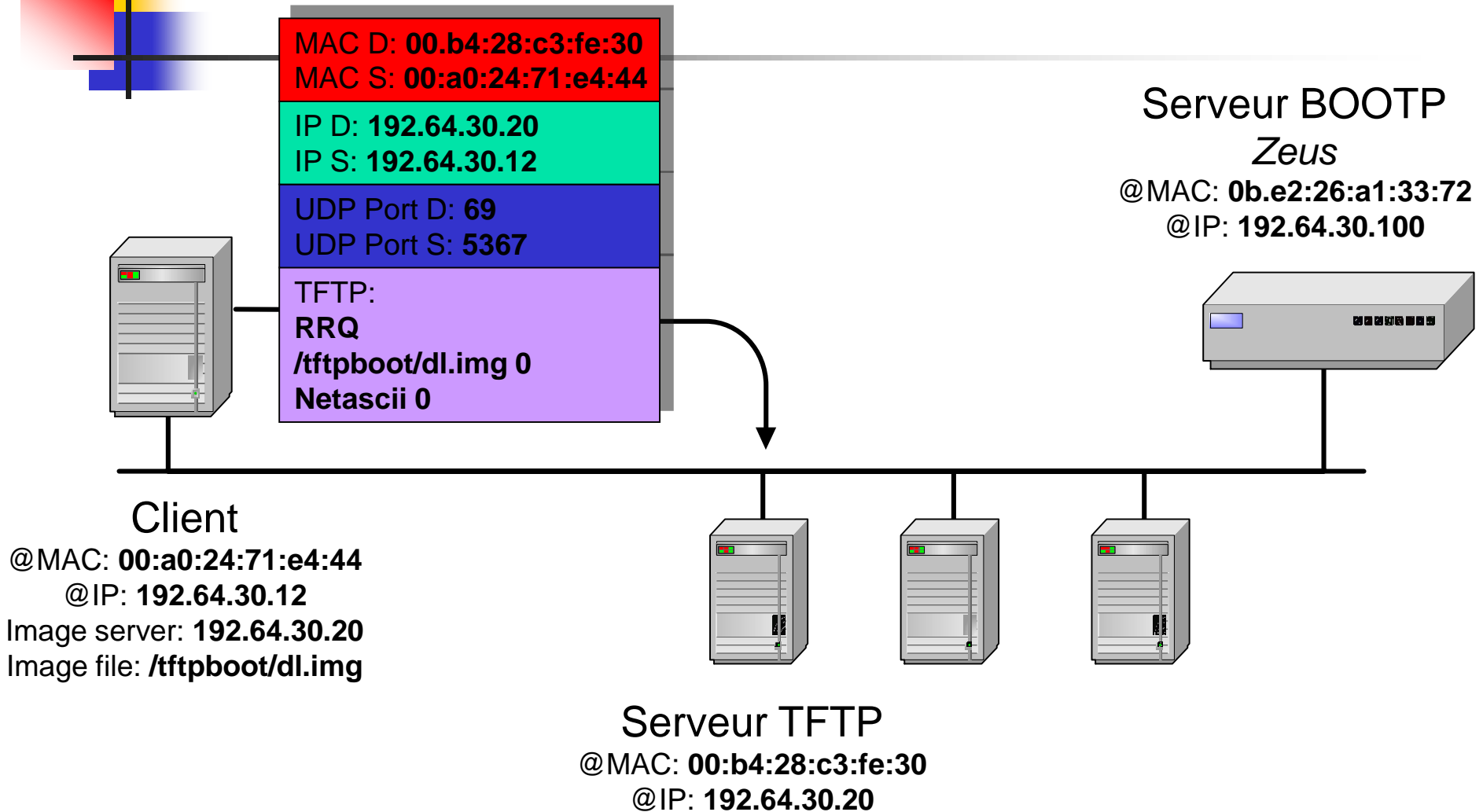


Serveur TFTP

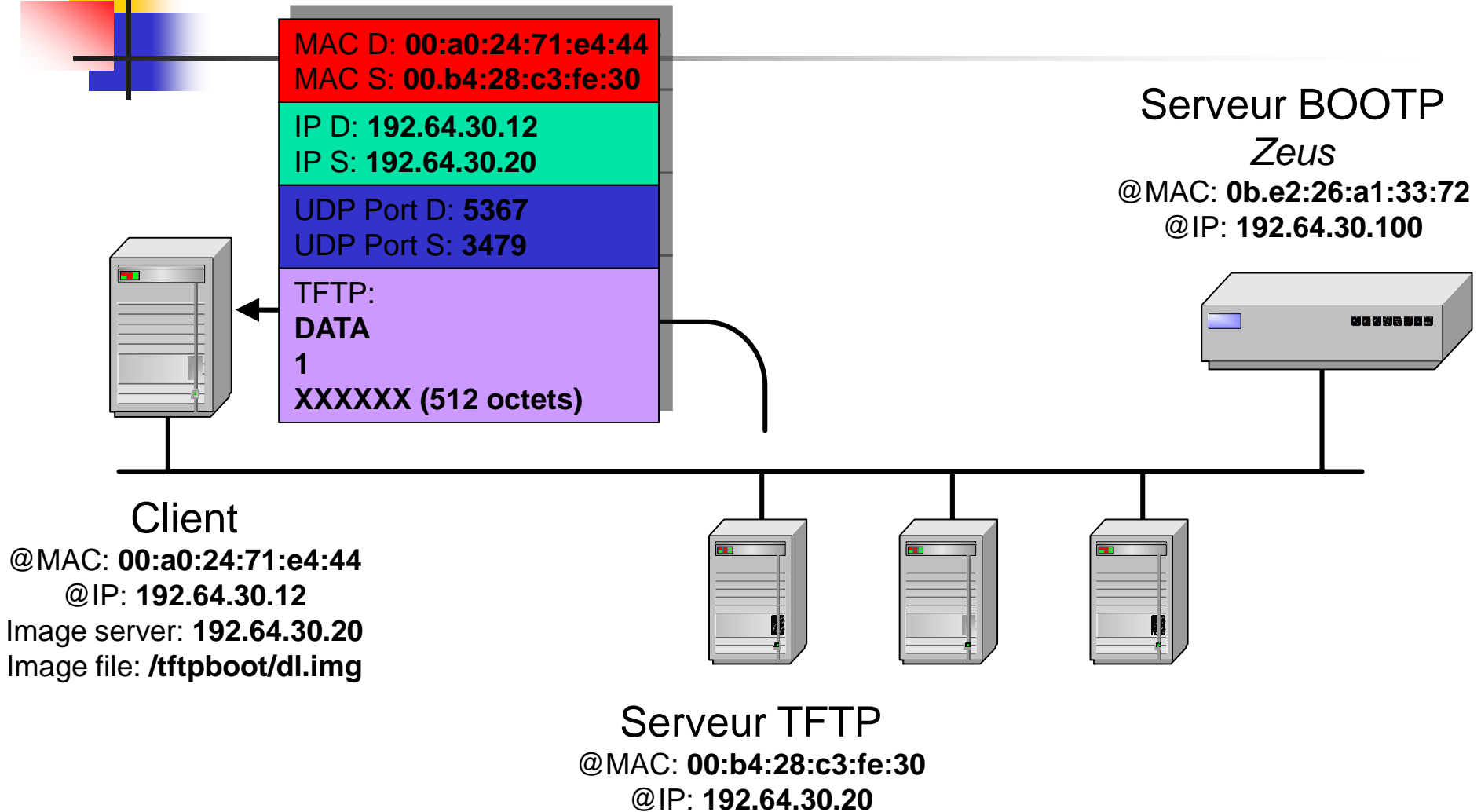
@MAC: **00:b4:28:c3:fe:30**

@IP: **192.64.30.20**

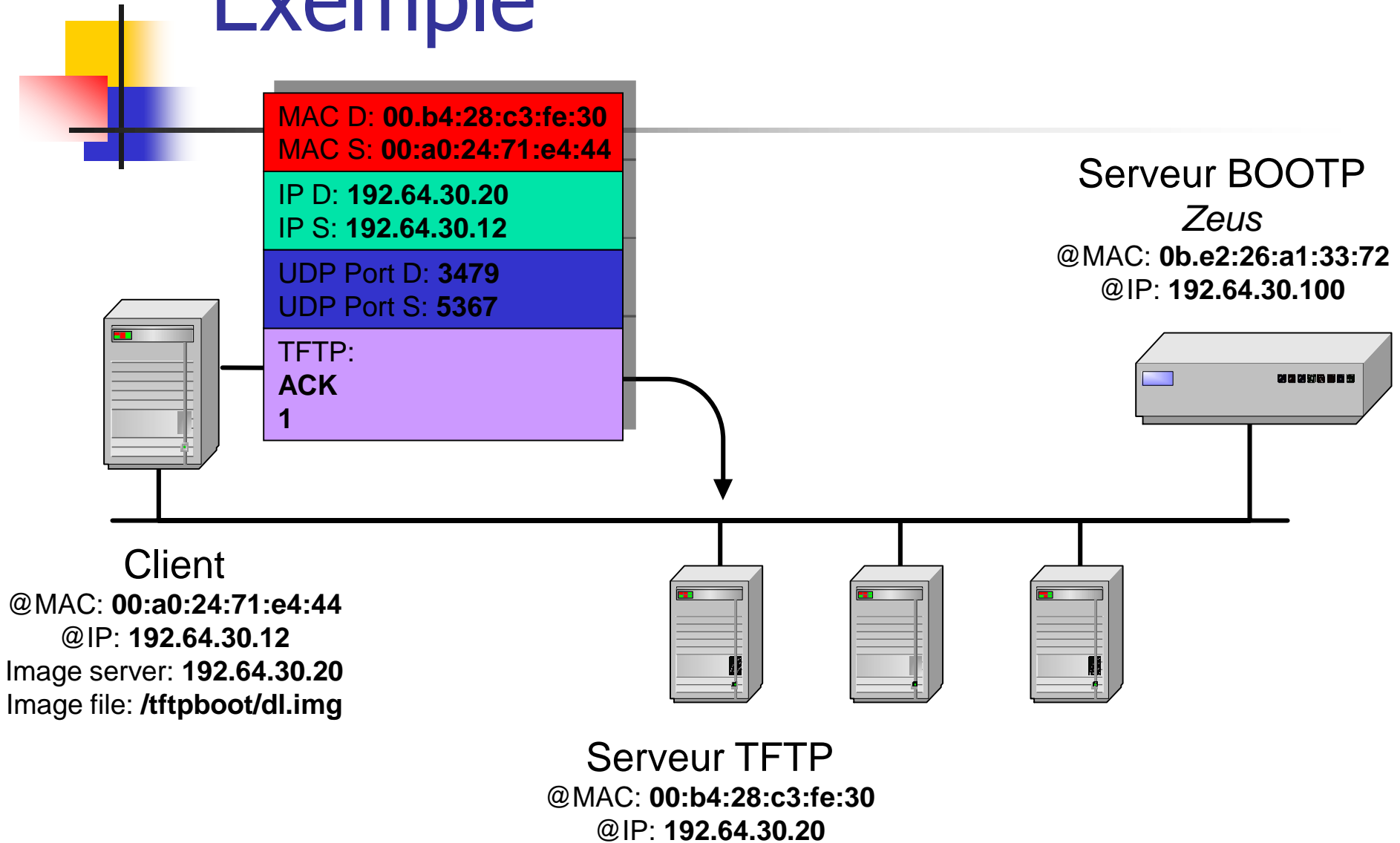
# Exemple



# Exemple



# Exemple





# BOOTP et DHCP

---

- Un autre protocole, DHCP (Dynamic Host Configuration Protocol ) permet, lui, d'attribuer cette adresse IP dynamiquement, c'est-à-dire que l'adresse IP affectée à la machine qui démarre peut changer d'un démarrage à l'autre.
- BOOTP fait cela de manière statique en utilisant un serveur (ou plusieurs) qui contient dans un fichier l'adresse IP à distribuer à chaque machine. Le fichier est maintenu à jour par l'administrateur du réseau



# BOOTP et DHCP

---

- Hormis les machines fixes (serveurs) du réseau, toute machine cliente devrait utiliser DHCP pour son adresse. Ca évite bien des problèmes de copies de configurations avec la même adresse.
- DHCP est standard sous Windows95 et WindowsNT46
- DHCP utilise le format de BOOTP et s'appuie sur les passerelles pour faire parvenir les requêtes au serveur.
- Le champ non utilisé contient des options DHCP et la trame dépasse 300 octets.



DHCP

---

Dynamic Host Configuration  
Protocol



# Protocole DHCP

---

- Dynamic Host Configuration Protocol
- Permet à un ordinateur qui se connecte à un réseau d'obtenir dynamiquement sa configuration
- Distribution des adresses IP sur un réseau



# DHCP

---

- Dynamic Host Configuration Protocol
  - RFC 951, 1497, 1541, 1542, 2131, 2132
- Role : Obtenir dynamiquement sa configuration IP
- Avantages :
  - Mieux partager l'espace d'adresses IP d'un réseau
    - e.g. réseau local, ISP etc.
  - Simplifier l'administration réseau
    - Configurer automatiquement et dynamiquement les machines



# Les informations renvoyées par DHCP

---

- Subnet Mask, Name Server, Hostname, Domain Name, Forward On/Off, Default IP TTL, Broadcast Address, Static Route, Ethernet Encapsulation, X Window Manager, X Window Font, DHCP Msg Type, DHCP Renewal Time, DHCP Rebinding, Time SMTP-Server, SMTP-Server, Client FQDN, Printer Name, ...



# DHCP

- Extension de BOOTP
  - Même format de messages
  - Un message DHCP est une option de BOOTP
  - C'est l'option 53
  - Le champ de données est de longueur 1 octet
  - Il comporte le numéro identifiant la requête

<b>Valeur</b>	<b>Type de Message</b>
1	DHCPDISCOVER
2	DHCPOFFER
3	DHCPREQUEST
4	DHCPDECLINE
5	DHCPACK
6	DHCPNAK
7	DHCPRELEASE
8	DHCPINFORM



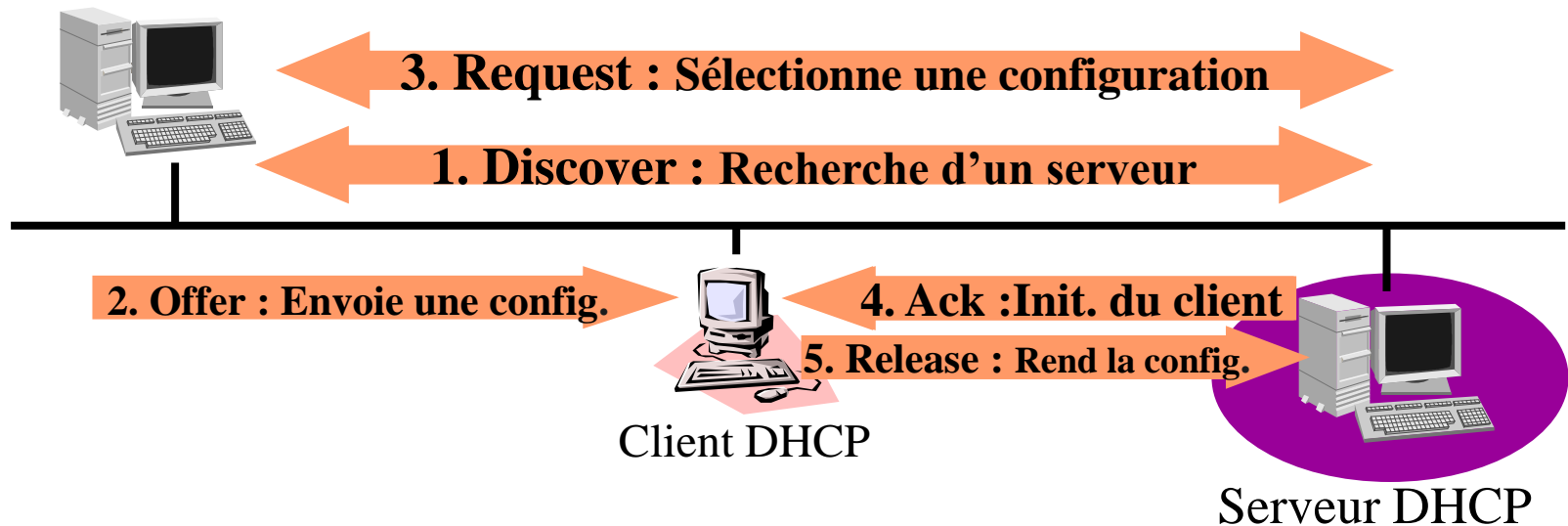
# Fonctionnement de DHCP

---

- Serveur DHCP :
  - Distribue les adresses IP
  - A une adresse IP fixe
- Déroulement :
  - Le client émet en broadcast un paquet de type **DHCPDISCOVER**, pour identifier les serveurs DHCP disponibles ;
  - Le serveur répond par un paquet **DHCPOFFER** (broadcast), qui contient les premiers paramètres ;
  - Le client établit sa configuration et envoie un **DHCPREQUEST** pour valider son adresse IP ;
  - Le serveur répond par un **DHCPACK** avec l'adresse IP pour confirmer l'attribution.

# DHCP : Allocation dynamique d'adresses IP

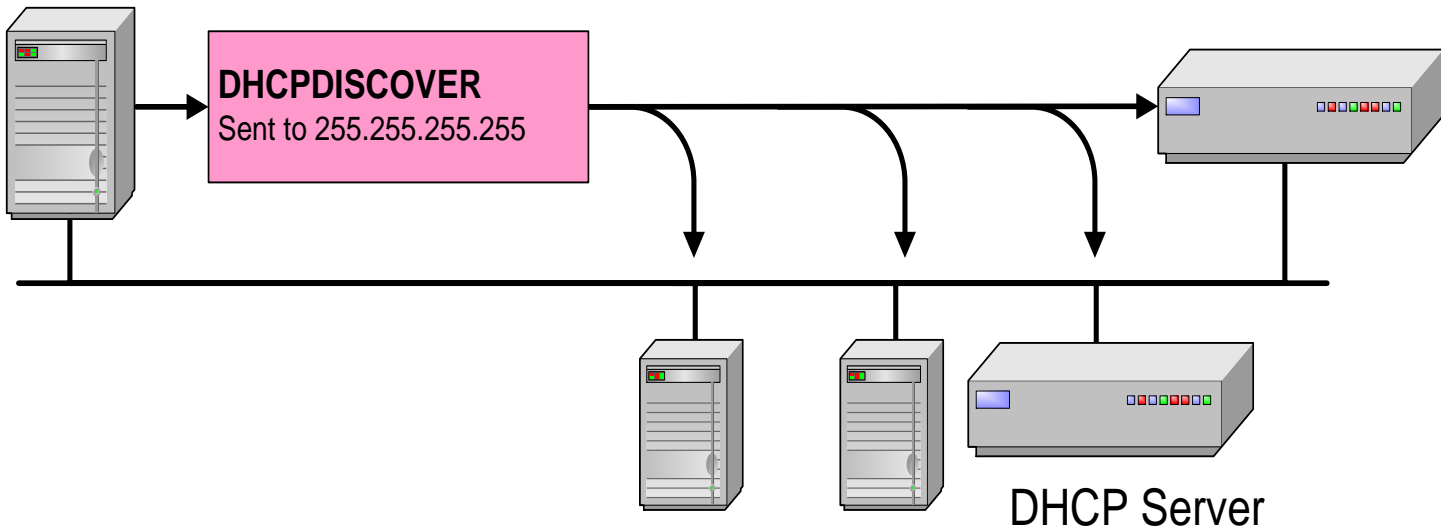
Serveur DHCP



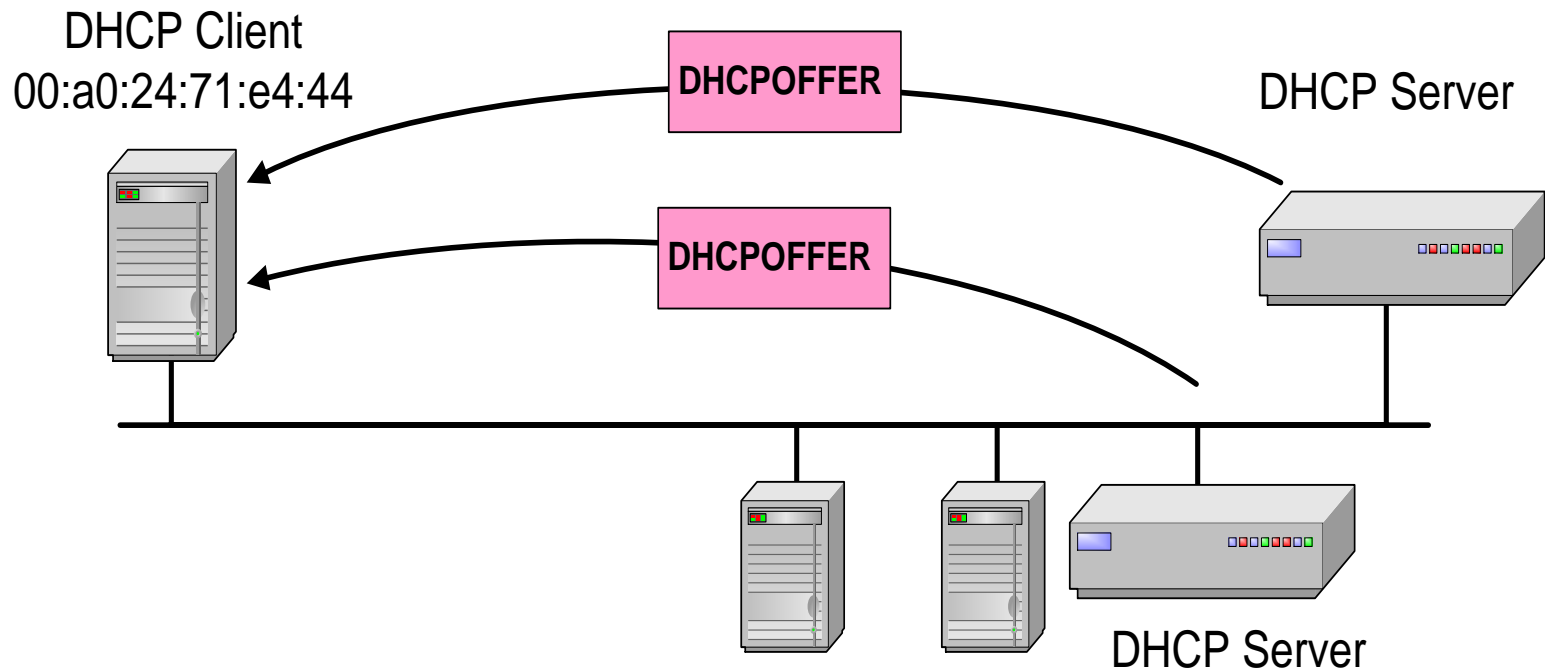
# DCHP DISCOVER

DHCP Client  
00:a0:24:71:e4:44

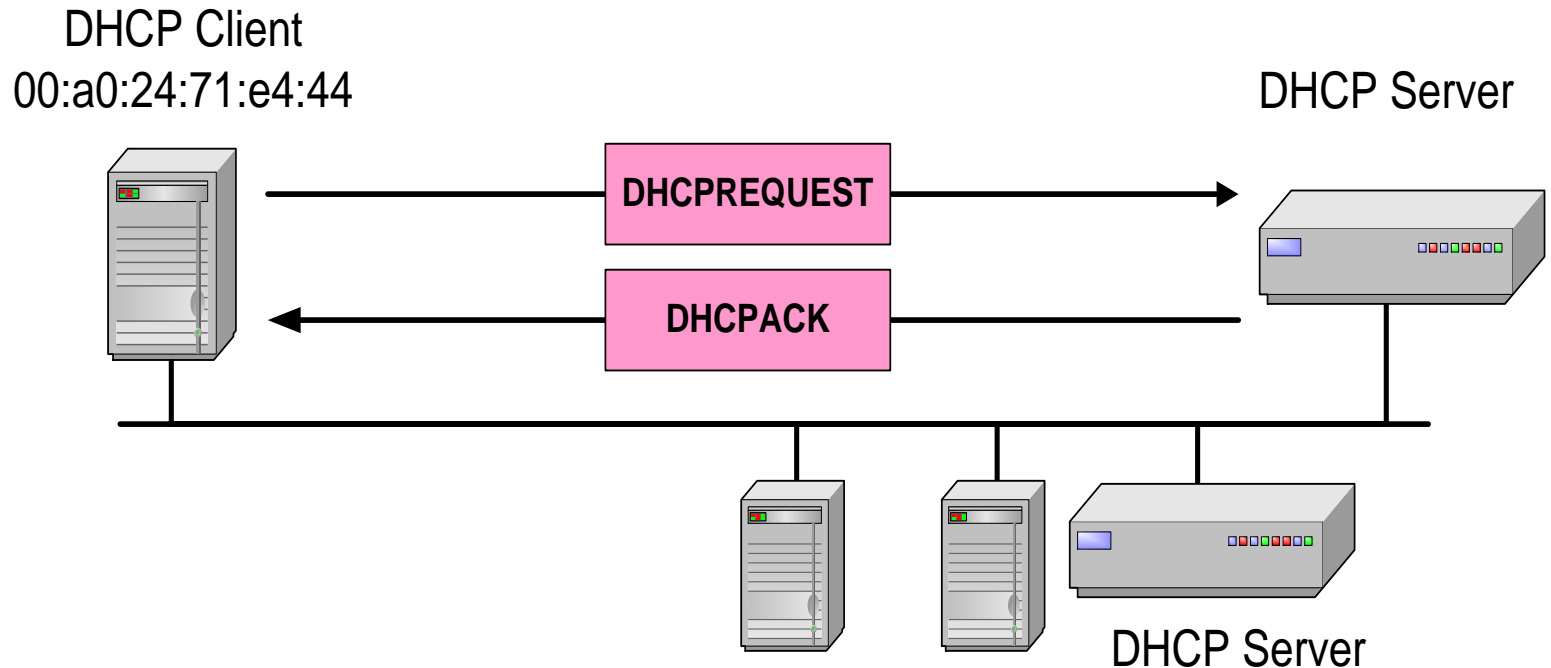
DHCP Server



# DCHP OFFER



# DHCP REQUEST



**Le client DHCP peut utiliser l'adresse IP**



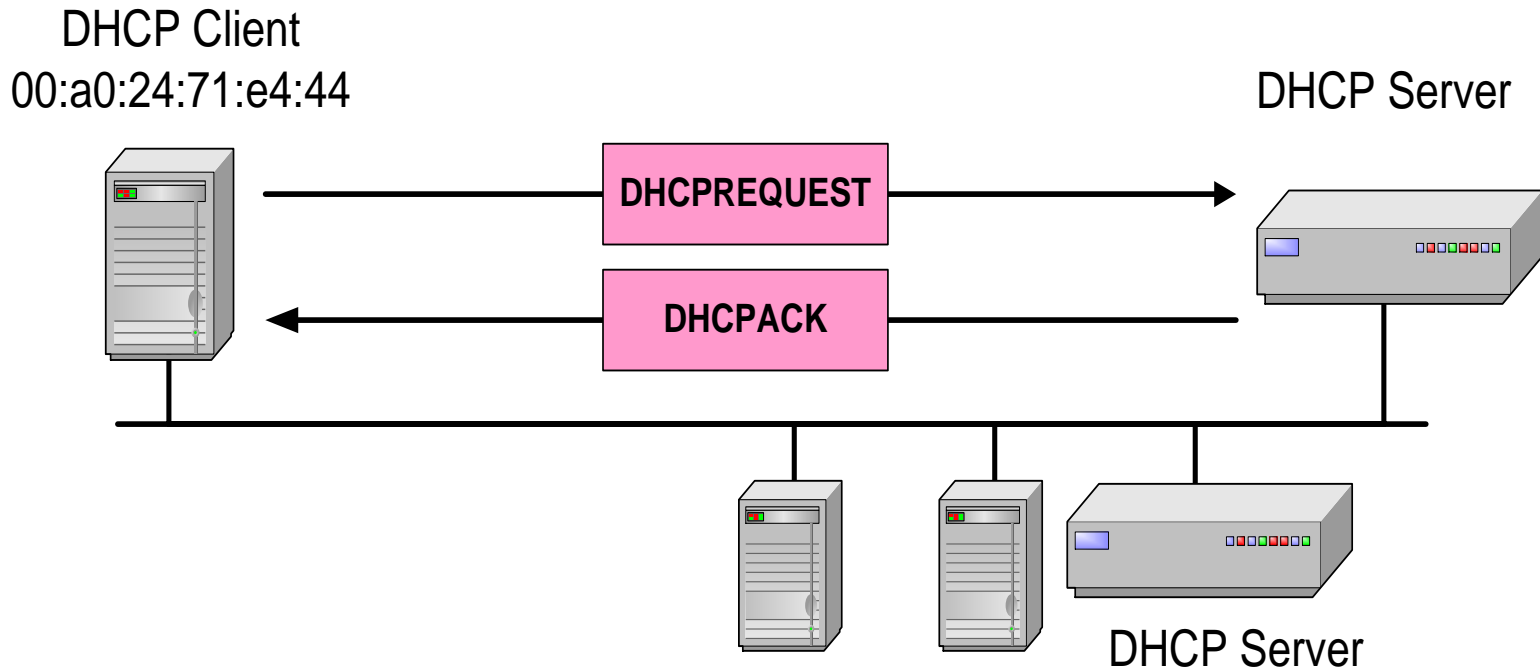
# Les baux DHCP

---

- Adresses IP délivrées avec une date de début et une date de fin de validité : bail.
- Demande (par le client) de prolongation du bail : **DHCPREQUEST** ;
- Proposition de prolongation du bail, lorsque celui-ci expire : **DHCPNAK** ;
- Optimisation de l'attribution des adresse IP en jouant sur la durée des baux
  - Courte durée pour les réseaux où les ordinateurs se branchent et se débranchent souvent,
  - Longue durée pour les réseaux constitués en majorité de machines fixes.
- Serveur DHCP le + répandu : développé par l'Internet Software Consortium.

# Renouveler le bail

(e.g. expiration de 50% du bail)

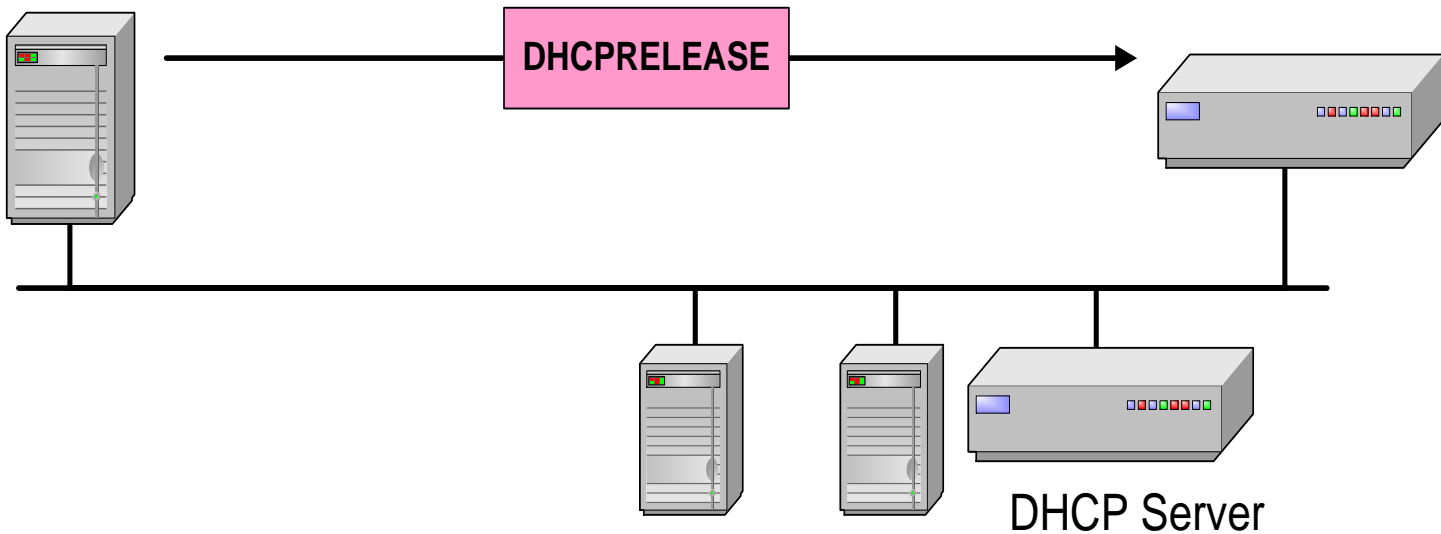


**L'adresse IP est libérée si le serveur DHCP envoie DHCPNACK**

# DHCP RELEASE

DHCP Client  
00:a0:24:71:e4:44

DHCP Server



**Le client DHCP a déjà libéré l'adresse IP**