

Couche Application

Prof. Rami Langar

LIGM/UPEM

Rami.Langar@u-pem.fr

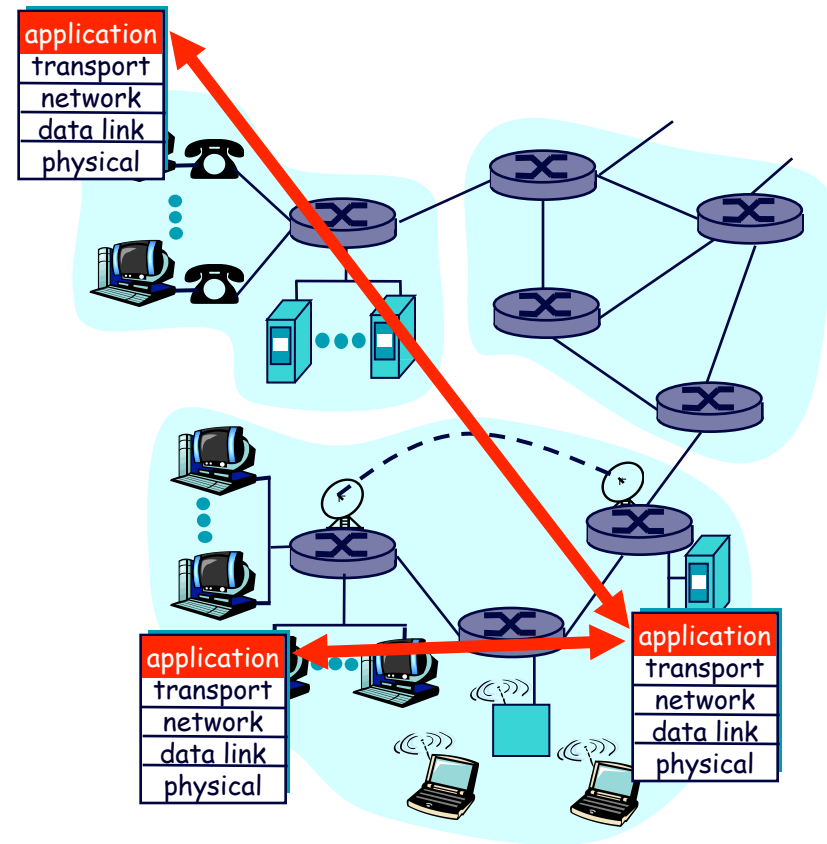
<http://perso.u-pem.fr/~langar>

Plan

- Introduction
- Applications et protocoles applicatifs
- Le Web: protocole HTTP
- Domain Name System (DNS)

Applications réseaux : le jargon

- Un **processus** est un programme qui s'exécute sur un hôte
- Deux processus s'exécutant sur deux hôtes différents communiquent en s'échangeant des **messages** avec un **protocole de couche application**
- Le processus suppose qu'il existe une infrastructure de **transport** en-dessous
- Un agent utilisateur est une interface entre l'utilisateur et l'application réseau
 - Web: browser
 - E-mail: eudora, outlook
 - streaming audio/video: real player, media player



Applications et protocole applicatif

Applications

- Logiciels distribués entre plusieurs systèmes
- S'exécutent dans les hôtes dans l'espace utilisateur
- Échangent des messages
- Ex : email, transfert de fichier, Web

Protocole Applicatif

- Définit :
 - Le type des messages échangés : requête, réponse...
 - La syntaxe des différents types de messages : champs du message
 - La sémantique des champs, çàd la signification des informations qui y sont contenues
 - Les règles pour déterminer quand et comment un processus envoie des messages et y répond
- Certains protocoles applicatifs sont spécifiés dans des RFCs : domaine public (ex HTTP)
Beaucoup sont propriétaires (ex téléphonie IP)

Paradigme client-serveur

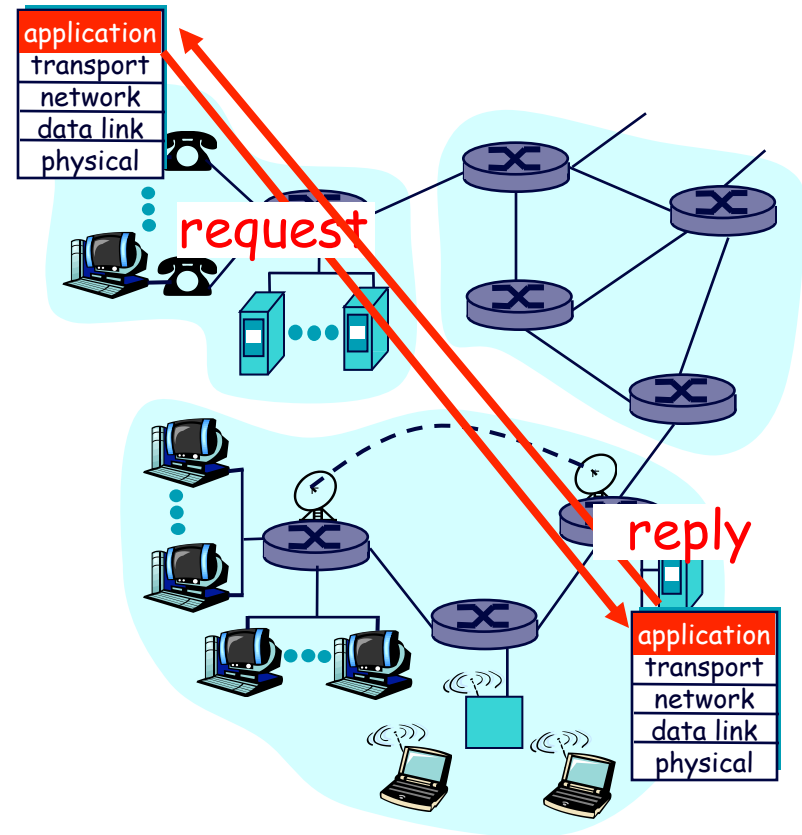
Les réseaux typiques ont deux parties :
le *client* et le *serveur*

Client:

- Initie le contact avec le serveur (“parle en premier”)
- Typiquement demande un service du serveur
- Pour le Web, le client est implanté dans le browser; pour le e-mail dans le lecteur de mail

Serveur:

- Propose les services demandés par le client
- e.g., Le Web serveur envoie les pages Web demandés



Protocole applicatif

API : Application Programming Interface

- Définit l'interface entre l'application et la couche transport
- Socket : API Internet (Deux processus communiquent en émettant et recevant des données via les sockets).

Q : Comment un processus « identifie » un processus distant pour communiquer?

- Nom ou adresse de l'hôte distant :
 - *Adresse IP* de l'hôte distant : 32 bits qui identifient de manière unique l'interface qui connecte l'hôte à l'Internet
- Identifiant du processus récepteur chez l'hôte distant:
 - “*Numéro de port*” - permet de différencier les différents processus locaux auxquels le message doit être transmis
 - Les protocoles applicatifs usuels ont des numéros de port réservés :
 - 80 pour le processus serveur Web
 - 25 pour le protocole de serveur de mail (utilisant SMTP)

Quel est le service de transport nécessaire à une application?

□ Perte de données

- Certaines applis (e.g., audio) peuvent tolérer des pertes
- D'autres applis (e.g., ftp, telnet) nécessitent une fiabilité à 100%

□ Délai

- Certaines applis (e.g., voix sur IP, jeux interactifs) nécessitent un délai de bout-en-bout faible (moins de quelques centaines de ms)

□ Bande passante

- Certaines applis (e.g., multimedia) requièrent une bande passante minimale
 - Téléphonie sur Internet : si la voix est codée à 32 Kbps, les données doivent être transmises à ce débit
- D'autres applis ("applis élastique") utilisent la bande passante disponible
 - Courrier électronique, transfert de fichiers, accès distant, Web

Services proposés dans Internet

Service TCP:

- *Orienté connexion*: connexion nécessaire entre le client et le serveur
- *Transport fiable* entre le processus émetteur et récepteur
- *Contrôle de flot*: l'émetteur ne submerge pas le récepteur
- *Contrôle de Congestion* : réduit le débit de l'émetteur quand le réseau est congestionné
- *Ne propose pas*:
 - de garanties de délai,
 - de bande passante minimale

Service UDP:

- Transfert de données non fiable
- Ne propose pas
 - de connexion,
 - de fiabilité,
 - de contrôle de flot,
 - de contrôle de congestion,
 - de garantie temporelle,
 - de bande passante

Applis Internet: protocoles applicatifs et protocoles de transport

Application	Protocole applicatif	Protocole de transport
e-mail	SMTP [RFC 821]	TCP
Accès distant	telnet [RFC 854]	TCP
Web	HTTP [RFC 2068]	TCP
Transfert de fichiers	FTP [RFC 959]	TCP
streaming multimedia	propriétaire (ex : RealNetworks)	TCP ou UDP
Fichier distant	NFS	TCP ou UDP
Voix sur IP	propriétaire (ex : Vocaltec)	En général UDP

Le Web : jargon

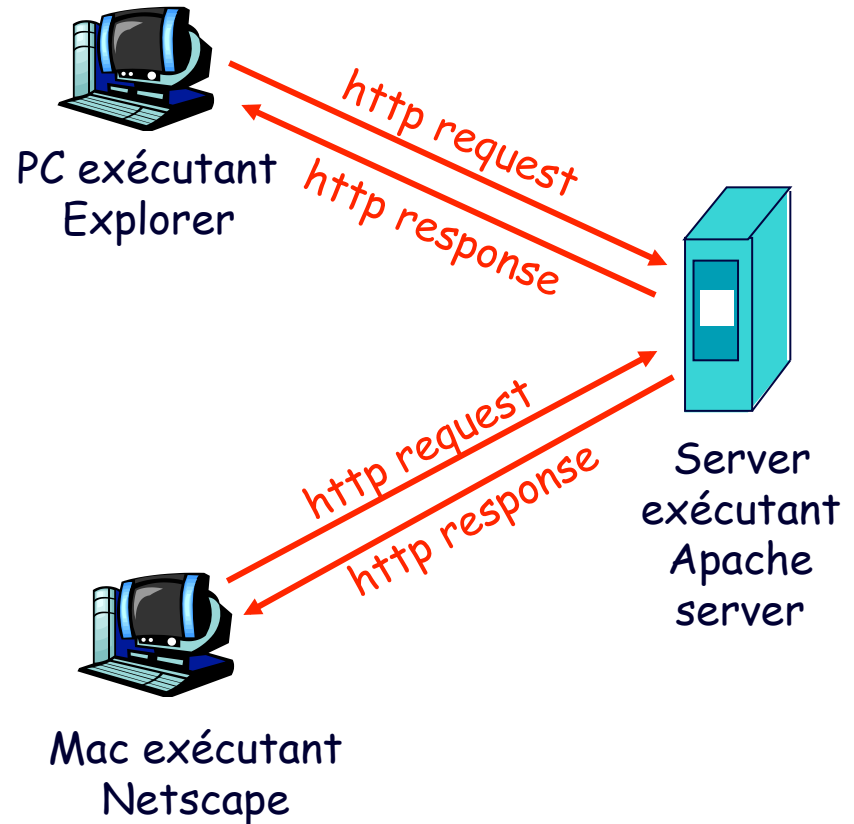
- Page Web :
 - Contient des “objets”
 - Adressée par une URL
- La plupart des pages Web contiennent :
 - Page HTML de base
 - Objets référencés
- L'URL a deux composantes :
 - nom du serveur
 - chemin d'accès
- L'Agent Utilisateur pour le Web est le browser :
 - MS Internet Explorer
 - Netscape Communicator
- Le serveur Web :
 - Apache (domaine public)
 - MS Internet Information Server

`www.someSchool.edu/someDept/pic.gif`

Le Web : le protocole HTTP

HTTP : HyperText Transfer Protocol

- Couche applicative Web
- Modèle client/serveur
 - *Client* : le browser, qui demande, reçoit, affiche les objets Web
 - *Serveur* : le serveur Web, qui envoie les réponses aux requêtes
- http1.0 : RFC 1945
- http1.1 : RFC 2068



Le protocole HTTP

HTTP :

service de transport TCP

- Le client initie une connexion TCP (crée une socket) avec le serveur, port 80
- Le serveur accepte la connexion TCP du client
- Les messages HTTP (protocole applicatif) sont échangés entre le browser (client HTTP) et le serveur Web
- La connexion TCP est close

HTTP est « sans état »

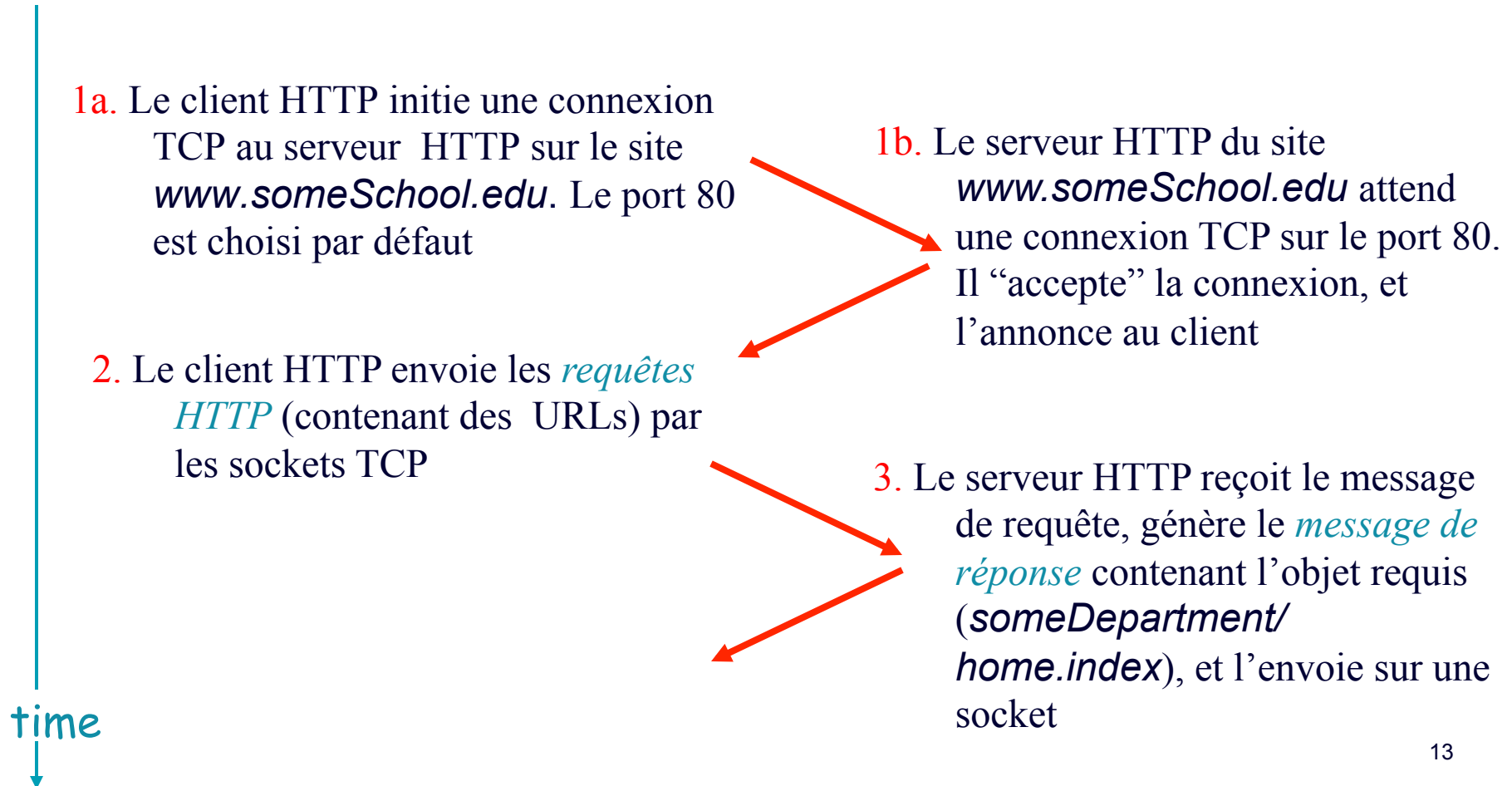
- Le serveur ne maintient aucune information au sujet des requêtes précédentes des clients

Les protocoles gardant un “état” sont complexes !

- L’histoire passée doit être gardée
- Si le serveur ou le client se crashe les états peuvent être incohérents

Exemple HTTP

Si un utilisateur entre l'URL : *www.someSchool.edu/someDepartment/home.index*



Exemple HTTP (suite)

4. Le serveur HTTP ferme la connexion TCP

5. Le client HTTP reçoit la réponse contenant le fichier HTML file et l'affiche.

En décodant le fichier, le browser trouve les URLs référencées

time



6. Les étapes 1-5 sont répétées pour chaque URL référencée

Connexions Persistantes et Non-persistantes

Non-persistante

- ❑ HTTP/1.0
- ❑ Le serveur interprète les requêtes, répond et ferme la connexion TCP
- ❑ 2 RTTs sont nécessaires pour lire chaque objet
- ❑ Chaque transfert doit supporter le slow-start
- ❑ Exemple : page contenant :
 - 1 fichier HTML
 - 10 images JPEG

Persistante

- ❑ Par défaut dans HTTP/1.1
- ❑ Une seule connexion TCP est ouverte vers le serveur
- ❑ Le client envoie la requête de tous les objets requis dès qu'ils sont référencés dans le HTML
- ❑ Moins de RTTs et moins de slow start.
- ❑ Deux versions : avec/sans pipeline

Format de message http : requête

- Deux types de messages http : *requête, réponse*
- **message de requête http :**
 - ASCII

Ligne de requête
(commandes
GET, POST, HEAD)

Lignes
d'entête

```
GET /somedir/page.html HTTP/1.0
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/4.0
Accept-language:fr
```

Le retour chariot
indique la fin
du message

Format de message http : réponse

Ligne d'état
(protocole,
code d'état,
message d'état)

Lignes
d'entête

HTTP/1.0 200 OK

Connection: close

Date: Thu, 12 Oct 2009 12:00:15 GMT

Server: Apache/1.3.0 (Unix)

Last-Modified: Mon, 22 Jun 2009

Content-Length: 6821

Content-Type: text/html

données, e.g.,
Le fichier
html

data data data data data ...

Code de réponse HTTP

Dans la première ligne de la réponse serveur->client.

200 OK

- La requête a réussi et l'objet demandé est à la suite

301 Moved Permanently

- L'objet demandé a changé définitivement de place, son nouvel emplacement est donné dans la suite du message

400 Bad Request

- La requête est erronée

404 Not Found

- Le document demandé n'est pas disponible sur le serveur

505 HTTP Version Not Supported

Essayer le serveur http

1. Telnet à votre serveur web favori

telnet www.eurecom.fr 80

Ouvre une connexion TCP vers le port 80 de www.eurecom.fr.

2. Taper une requête HTTP

GET /~ross/index.html HTTP/1.0

GET conditionnel

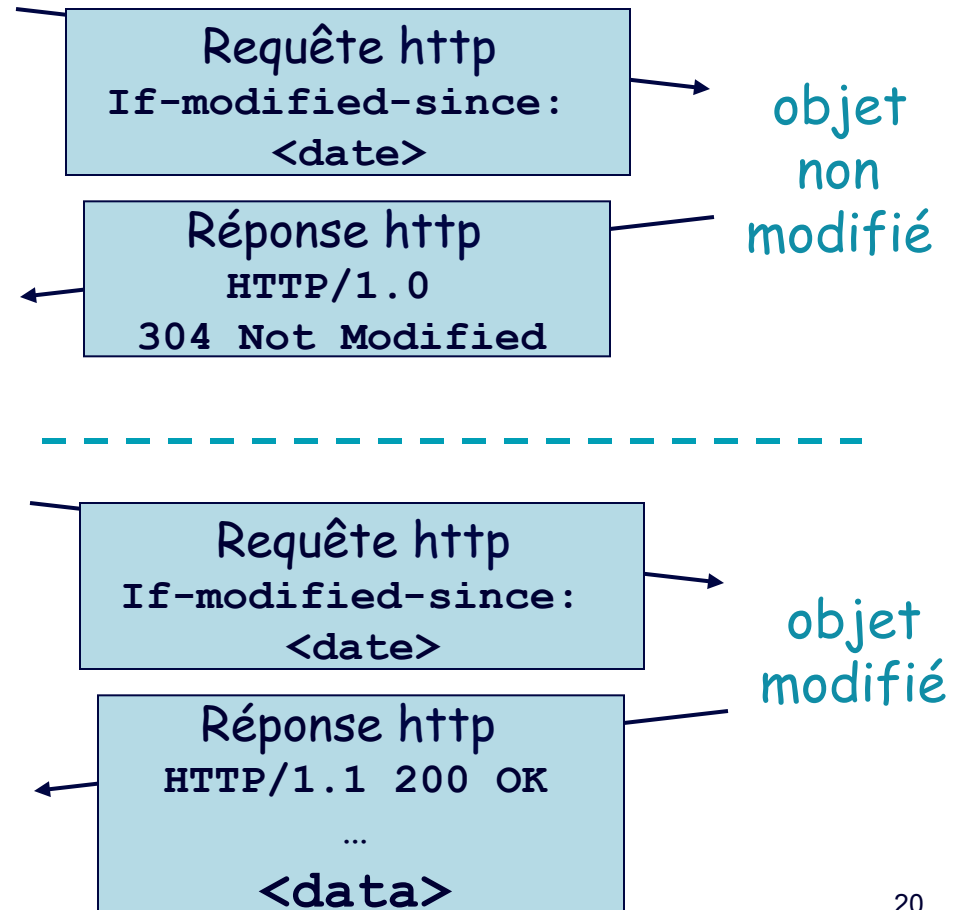
- ❑ **Objectif** : ne pas envoyer un objet que le client a déjà dans son cache
- ❑ Problème : les objets contenus dans le cache peuvent être obsolètes
- ❑ client: spécifie la date de la copie cachée dans la requête http

If-modified-since:
<date>

- ❑ serveur: la réponse est vide si la copie cachée est à jour
HTTP/1.0 304 Not Modified

client

serveur



Identification d'une machine TCP/IP

- ❑ Chaque machine TCP/IP est identifiée par un nom et une adresse IP
- ❑ Exemple
 - tf1.fr – 193.169.66.26
 - lip6.fr – 132.227.73.20
- ❑ Essayer sur votre machine
 - ping tf1.fr
 - ping lip6.fr

Nom vs. adresse

- ❑ Un nom, tel que tf1.fr ou lip6.fr, est facile à retenir pour les utilisateurs
- ❑ Une adresse IP, telle que 193.169.66.26 ou 132.227.73.20, est utilisée par les protocoles réseaux pour acheminer les paquets à la machine destination
- ❑ DNS (Domain Name System) est le système qui fait la traduction d'un nom en adresse IP et vice versa

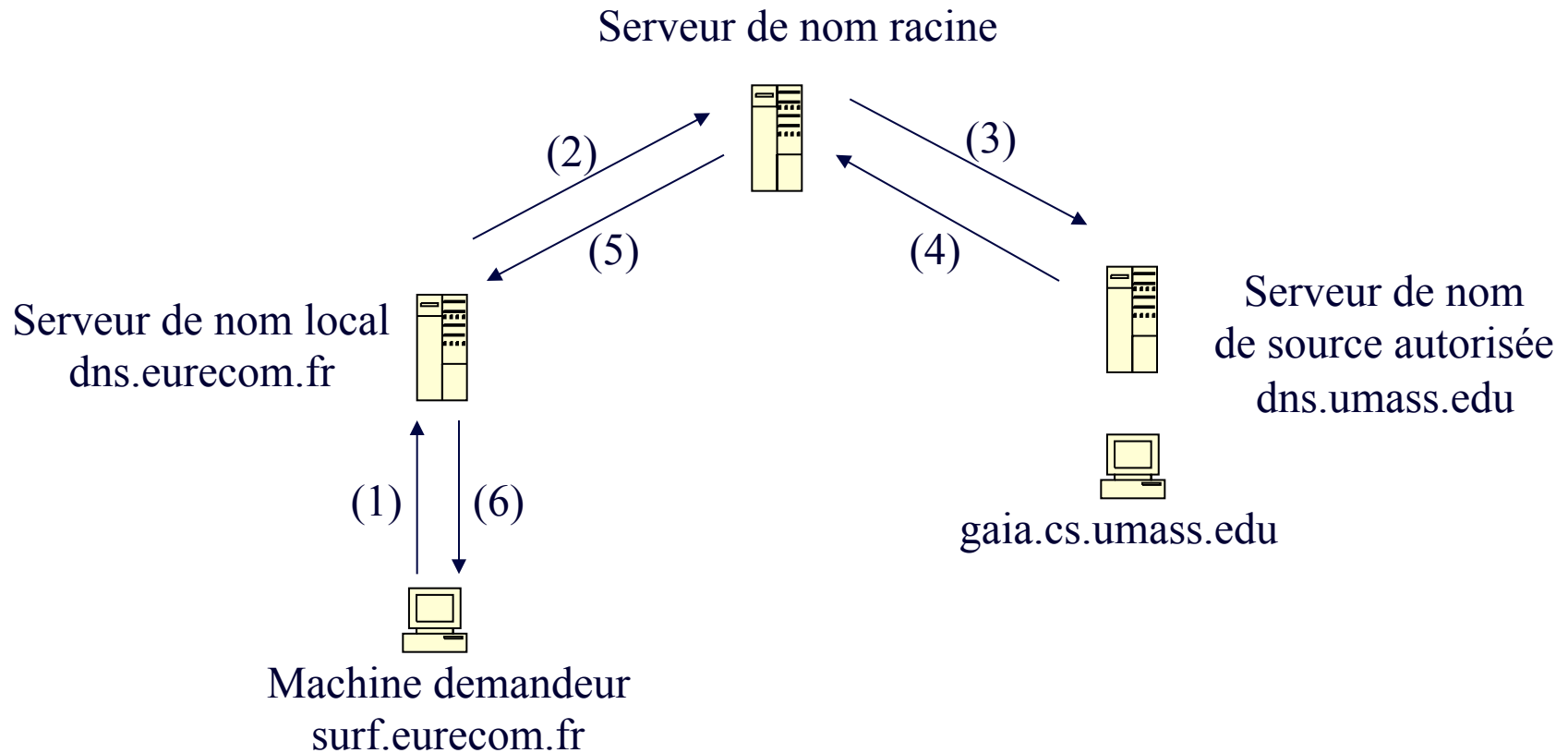
DNS

- Une hiérarchie de serveurs de noms
- Le protocole DNS pour la communication entre les machines avec les serveurs DNS ainsi qu'entre les serveurs DNS
 - Basé sur UDP, port 53
- Le DNS est souvent sollicité par des applications, telles que Web, courrier électronique ou transfert de fichier, pour obtenir les adresses IP correspondant aux noms de serveur demandés par les utilisateurs

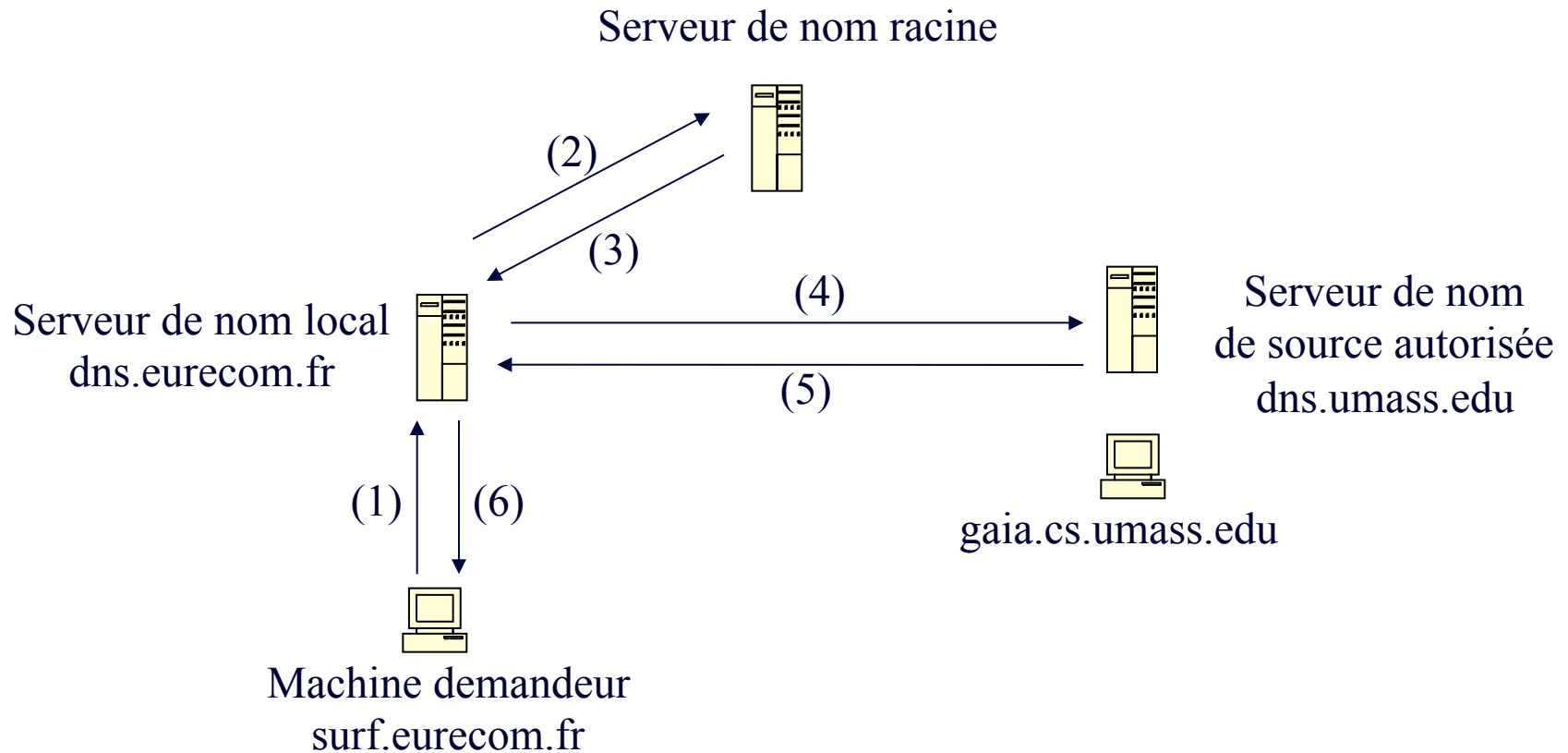
Hiérarchie de serveurs de noms

- Serveurs de noms locaux
 - Serveur de noms par défaut
 - Disposé par chaque fournisseur d'accès internet
 - Pour les conversions nom – adresse IP concernant des postes abonnés chez le même fournisseur d'accès
- Serveurs de noms « racine »
 - A solliciter par un serveur de noms local lorsque ce dernier n'est pas capable de faire la conversion
- Serveurs de noms de source autorisée
 - Serveur de noms auprès duquel une machine doit être enregistrée
 - Chaque machine doit être enregistré au minimum auprès de deux serveurs de noms de source autorisée
 - Essayer sur votre machine
 - `ipconfig /all`

Exemple de recherches récursives



Exemple de recherches itératives



Mise en mémoire cache DNS

- ❑ Lorsqu'un serveur de noms reçoit l'adresse IP correspondant à un nom de machine, il la place dans sa mémoire cache
- ❑ La mémoire cache est fréquemment actualisée (souvent après deux jours)
- ❑ Permettre de répondre immédiatement au deuxième client qui sollicite la même conversion

Enregistrements DNS

- ❑ Les correspondances nom – adresse IP sont conservées sur les serveurs de noms sous la forme des enregistrements de ressources (RR – Resource Record)
- ❑ Un enregistrement de ressources est constitué de quatre éléments
 - (Name, Value, Type, TTL)
- ❑ TTL (Time To Live)
 - La durée pendant laquelle l'enregistrement de ressources doit être conservé en mémoire cache

Les types d'enregistrement de ressources

- Type=A
 - Name: le nom d'une machine
 - Value: l'adresse IP correspondante
- Type=NS
 - Name: le nom d'un domaine
 - Value: le nom d'un serveur de noms de source autorisée qui sait faire la conversion recherchée
- Type=CNAME
 - Name: un alias
 - Value: le nom canonique de l'alias
- Type=MX
 - Name: un alias d'un serveur de messagerie
 - Value: le nom canonique d'un serveur de messagerie doté de cet alias