

Introduction aux réseaux

Olivier Dalle

Université de Nice - Sophia Antipolis

<http://deptinfo.unice.fr/>

D'après le cours original de

Sacha Krakowiak

Université Joseph Fourier

Projet Sardes (INRIA et IMAG-LSR)

<http://sardes.inrialpes.fr/~krakowia>

Plan pour les réseaux

Cours 6 : Introduction aux réseaux

Notions de base

Introduction aux protocoles

Fonctionnement de l'Internet, DNS

Cours 7 : Programmation pour les réseaux

Schéma client-serveur

Sockets



TD/TP :

Client-serveur avec *sockets*

Cours 8 : Le World Wide Web

Protocole HTTP

Organisation d'un serveur Web



mini-projet :

Construction d'un serveur Web

Introduction aux réseaux informatiques

- Un **réseau informatique** (*computer network*) est un système de communication (ensemble matériel + logiciel) qui permet à un ensemble d'ordinateurs (au sens large) d'échanger de l'information
 - ◆ sens large : points d'accès, terminaux de paiement, téléphones, capteurs divers, etc.
 - L'échange d'information n'est pas une fin en soi. Les réseaux servent avant tout à réaliser des **services**
 - ◆ accessibles à partir de tout organe connecté au réseau
 - ◆ mis en œuvre par un ensemble d'ordinateurs sur le réseau
 - ◆ exemples de services

le courrier électronique (<i>mail</i>)	l'accès au World Wide Web
le transfert de fichiers (<i>ftp</i>)	les services utilisant le Web :
l'accès à distance (<i>telnet</i>)	documentation, commerce électronique, etc.
- Lien entre réseaux et systèmes d'exploitation
 - ◆ Le réseau (support de communication) comme organe d'entrée-sortie
 - ◆ Le réseau (ensemble de serveurs) comme super-machine

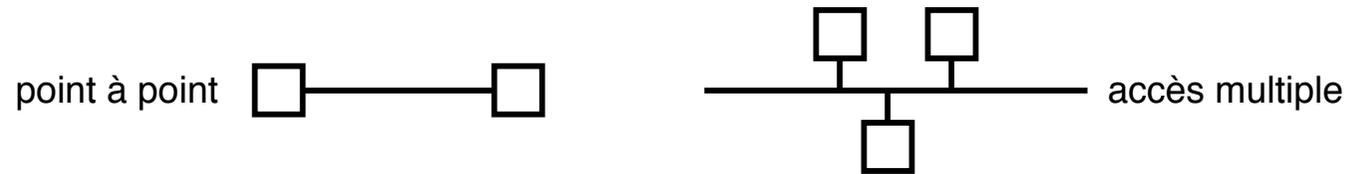
Types de réseaux (1)

■ Les réseaux peuvent être classés selon différents critères

◆ Nature de la liaison entre les organes connectés (nœuds)

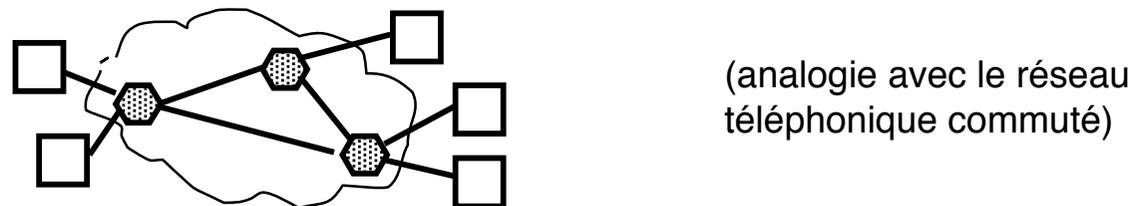
❖ Liaison directe

- ▲ il y a un lien direct entre deux nœuds du réseau



❖ Liaison commutée

- ▲ la liaison passe par des organes intermédiaires



❖ Les supports physiques de la communication peuvent être divers

- ▲ paires de fils, câble coaxial, fibre optique, radio, infra-rouge, etc.

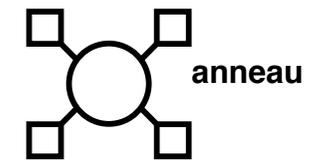
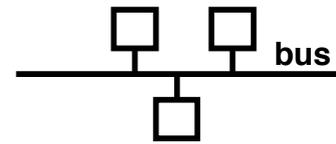
◆ Couverture géographique (réseau local, à grande distance, etc.)

Types de réseaux (2)

La classification par étendue de la couverture géographique est souvent utilisée, bien que non stricte

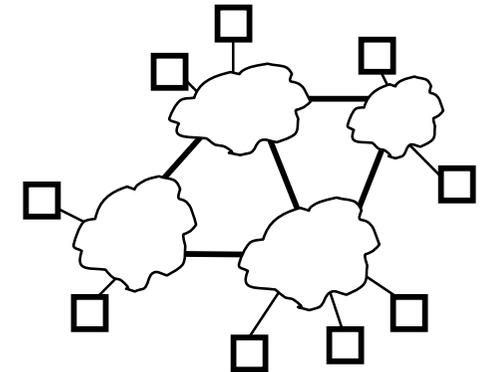
■ Réseaux locaux (*Local Area Networks, LAN*)

- ◆ Communication au sein d'une organisation (département d'entreprise, etc.)
- ◆ Administration unique
- ◆ Couverture géographique limitée (~1 km)
- ◆ Débit élevé, taux d'erreur faible
- ◆ Topologies diverses : bus, anneau



■ Réseaux à grande distance (*Wide Area Networks, WAN*)

- ◆ Communication entre des organisations diverses
- ◆ Administrations multiples
- ◆ Couverture géographique étendue : un pays, toute la planète
- ◆ Débit variable, taux d'erreur parfois non négligeable
- ◆ Topologie maillée ; interconnexion de réseaux (exemple : l'Internet)



■ Réseaux métropolitains (*Metropolitan Area Networks, MAN*)

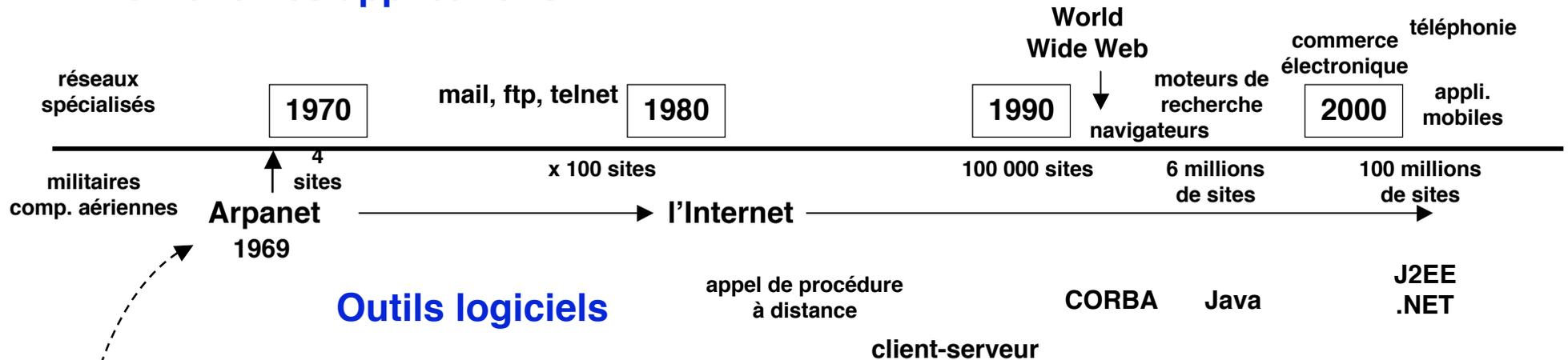
- ◆ Intermédiaires entre LAN et WAN - qq dizaines de km, ville ou région

■ Autres réseaux

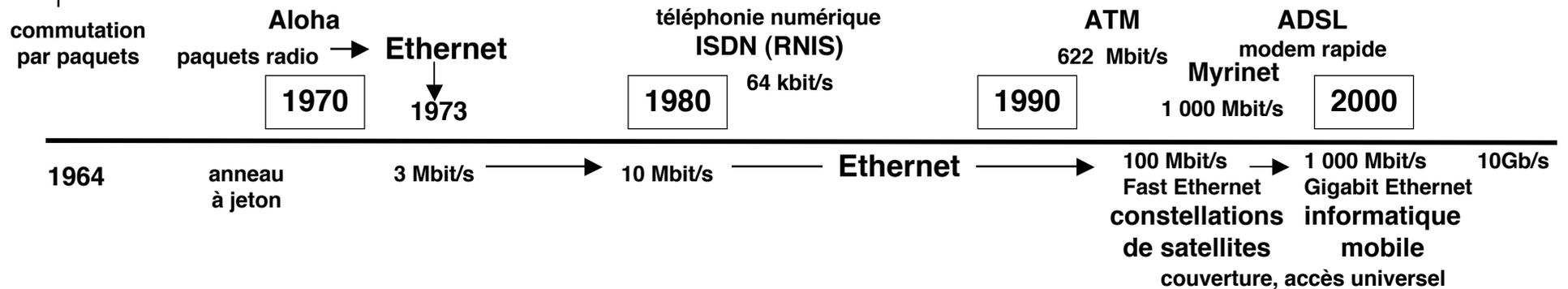
- ◆ Interconnexion de grappes de machines (*clusters*) - *Desk Area Networks* (DAN) - *Storage Area Networks* (SAN) : réseaux pour le stockage de données
- ◆ Réseaux de mobiles, réseaux de capteurs, ...

Bref historique des réseaux informatiques

L'Internet et les applications



Quelques avancées technologiques



Performances des réseaux (1)

■ Deux mesures principales de performance

- ◆ **Débit** (*throughput*) : quantité d'information par unité de temps. Unité : bit/s (Kbit/s, Mbit/s, ...)
 - ❖ une mesure corrélée au débit est la bande passante (Hz, KHz, ...)
- ◆ **Latence** (*latency*) : temps écoulé entre l'émission d'un bit et sa réception. Unité : s (ms, μ s, ...)
 - ❖ on s'intéresse aussi parfois au temps d'aller-retour (*round trip time* ou RTT)

■ Facteurs de performance

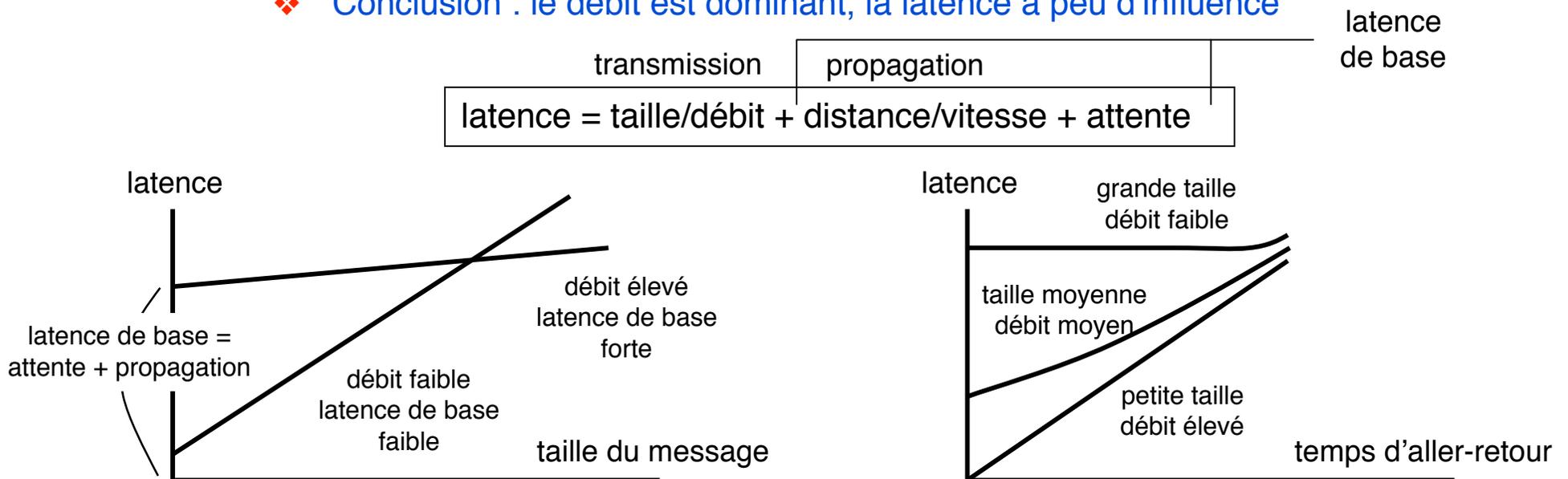
Latence = durée de transmission + temps de propagation + temps d'attente

- ◆ Durée de transmission : taille du message / débit
 - ❖ temps nécessaire pour transmettre les données (les envoyer sur le réseau)
- ◆ Temps de propagation : distance / vitesse de propagation
 - ❖ temps nécessaire pour que les données aillent de l'émetteur au récepteur
- ◆ Temps d'attente
 - ❖ temps "perdu" par le système de communication (notamment à cause de l'occupation des ressources)
- ◆ La somme (temps de propagation + temps d'attente), ou **latence de base**, est un délai incompressible (temps écoulé avant de recevoir le 1-er bit d'un message)

Performances des réseaux (2)

■ Débit et latence ont un impact variable selon les applications

- ◆ Exemple 1 : message bref (interaction question-réponse)
 - ❖ Envoi 1 Kbit, retour 1 Kbit, distance 10 000 km. Temps incompressible d'aller-retour = 100 ms (si vitesse = $2c/3$). Temps de transmission = 1 ms à 1 Mbit/s, 0,01 ms à 100 Mbit/s.
 - ❖ Conclusion : la latence est dominante, le débit a peu d'influence
- ◆ Exemple 2 : message très gros (fichier multimédia)
 - ❖ Envoi 100 Mbit, distance 10 000 km. Temps incompressible de propagation = 50 ms. Temps de transmission = 100 s à 1 Mbit/s, 1 s à 100 Mbit/s.
 - ❖ Conclusion : le débit est dominant, la latence a peu d'influence

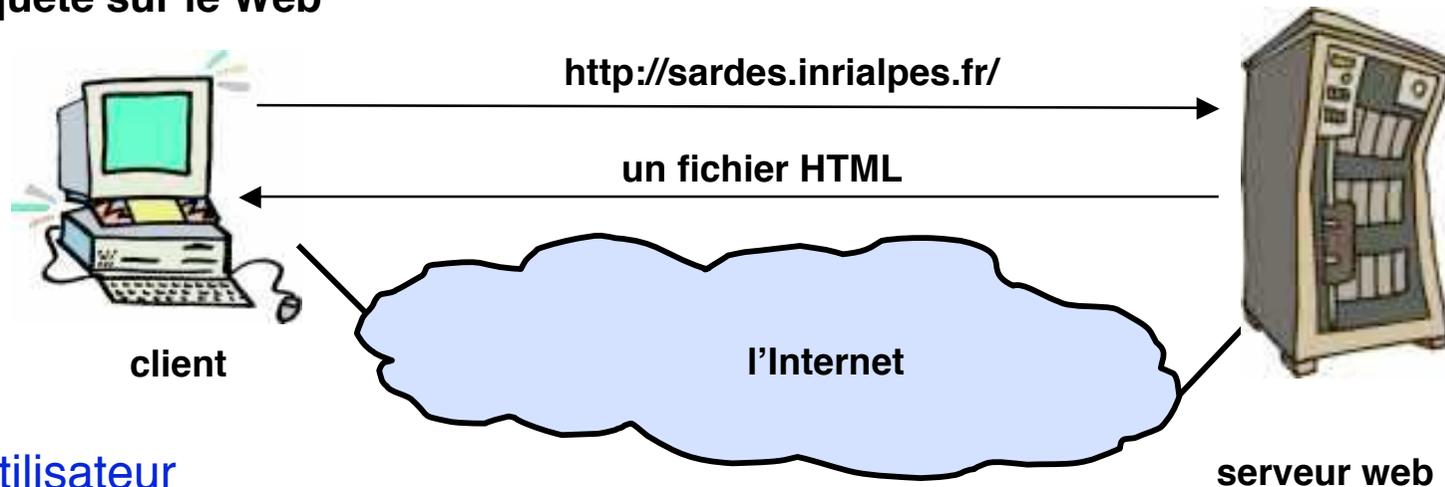


Qualité de service

- La qualité de service (*Quality of Service*, QoS) désigne un ensemble de facteurs de qualité nécessaires aux besoins d'une application particulière
 - ◆ Cette définition est générique et doit être précisée dans chaque cas
 - ◆ Les besoins en QoS dépendent de la nature des applications
- Exemples
 - ◆ Faible taux d'erreur (probabilité pour qu'un bit ou un message soit perdu ou modifié)
 - ❖ nécessaire si les données sont peu redondantes
 - ◆ Stabilité de la latence (absence de gigue)
 - ❖ nécessaire pour les applications multimédia (son et vidéo)
 - ◆ Garantie d'une limite supérieure sur la latence
 - ❖ nécessaire pour les applications critiques liées au temps réel
- Garanties de qualité de service
 - ◆ Problème difficile !
 - ◆ Voies d'approche : réservation de ressources

Comment fonctionne un réseau ? (1)

une requête sur le Web



Vu de l'utilisateur

on clique sur un lien
une "page web" s'affiche sur l'écran

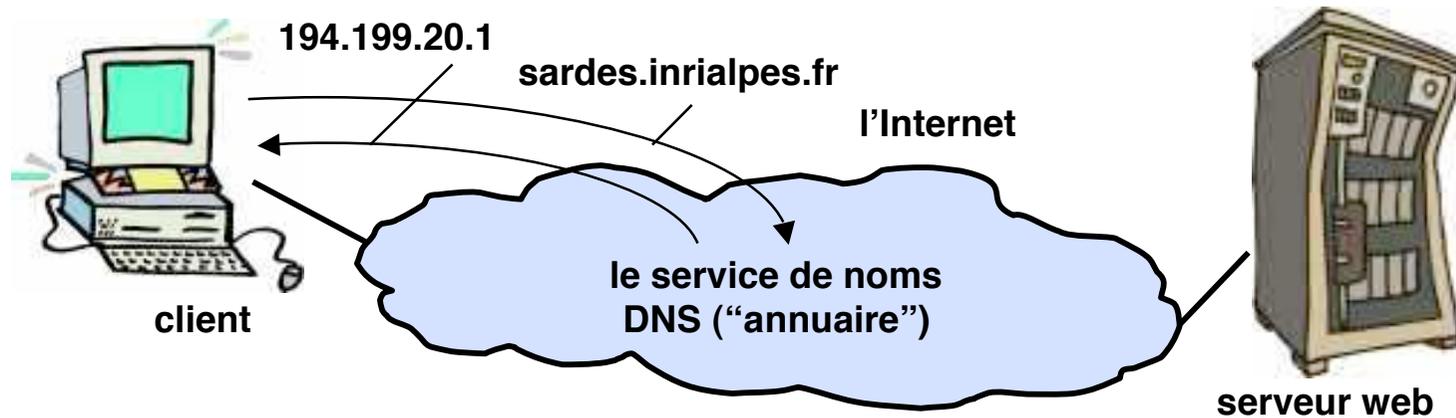
Sur la station client

le programme navigateur envoie une requête au serveur correspondant à l'URI
(*Uniform Resource Identifier*) associé au lien
(si tout se passe bien) le programme navigateur reçoit un fichier HTML qu'il sait afficher
sur l'écran

Sur le réseau, il faut :

trouver le bon serveur (celui qui correspond à l'URI)
transporter la requête depuis la station client vers le serveur
transporter le fichier depuis le serveur à la station client

Comment fonctionne un réseau ? (2)



Première étape : localiser le serveur

Via le service de noms de l'Internet (DNS) qui associe un nom à une "adresse IP"
(fonctionne comme un annuaire - détails plus loin)

Toute machine connectée à l'Internet a une **adresse IP**

Question : comment trouver l'annuaire ?

Réponse : on connaît l'adresse IP d'un point d'entrée

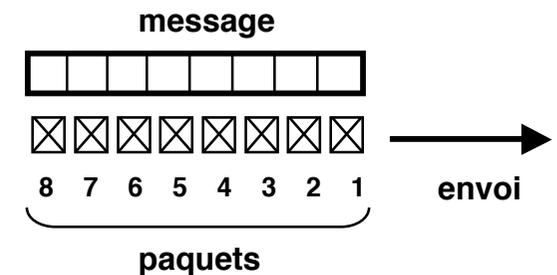
Deuxième étape : envoyer la requête au serveur

Comment est transmise la requête ?

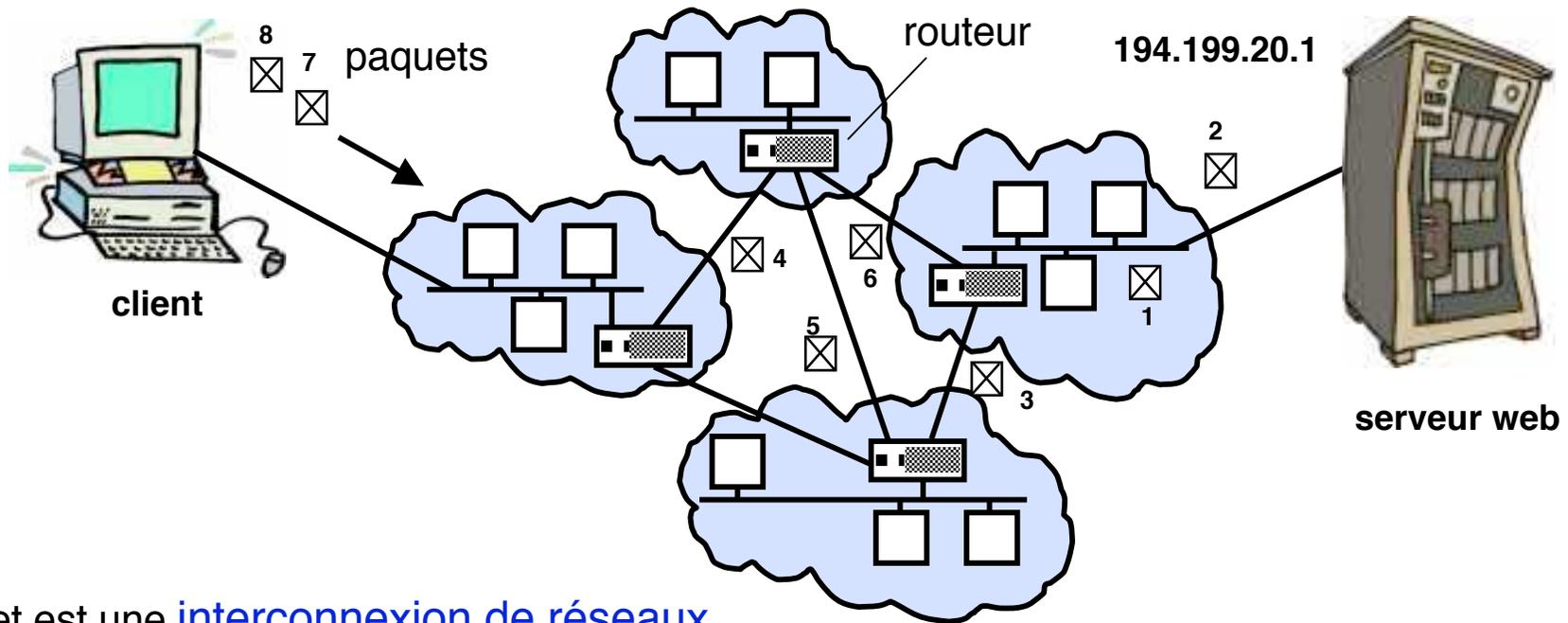
La requête est un message (une suite de bits).

Elle est découpée en "paquets" de taille fixe

Chaque paquet est envoyé sur le réseau



Comment fonctionne un réseau ? (3)



L'Internet est une **interconnexion de réseaux** (*internetwork*)

Les réseaux sont reliés entre eux par des **routeurs**

Chaque paquet contient son numéro et son adresse de destination. Quand il arrive sur un réseau, et que le site destinataire n'en fait pas partie, le paquet est transmis à un routeur.

Les routeurs contiennent les informations qui permettent d'acheminer le paquet vers son site destinataire

Comment fonctionne un réseau ? (4)

Fonctionnement du serveur

Le serveur reconstruit le message initial en mettant les paquets dans l'ordre de leurs numéros. Que fait le serveur à la réception de la requête ?

Le site serveur interprète la requête comme une demande de fourniture de fichier (HTML)
Il envoie le fichier au client (sous forme d'une suite de paquets, comme précédemment)

Que se passe-t-il si un paquet s'est perdu, ou a mal été transmis ?

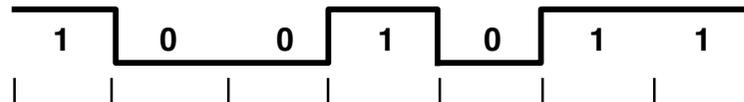
Le destinataire est capable de le détecter et demande qu'on lui renvoie le paquet manquant ou erroné.

Transmission de l'information

Comment sont transmis concrètement les paquets ?

Chaque paquet est une suite de bits. La transmission utilise un support physique : fils, fibre optique, ondes radio, etc., sur lequel sont envoyés des signaux (électriques, lumineux, etc.). Chaque bit (0 ou 1) est représenté par une configuration particulière du signal

Exemple
(non réaliste)



Connaître le chemin suivi (1)

La commande `traceroute` permet de montrer la suite de routeurs utilisés

Exemples :

```
{6}kernighan traceroute boole
traceroute to ufrima.imag.fr (195.221.225.1), 30 hops max, 40 byte packets
 1 ufrima (195.221.225.1) 0.518 ms 0.321 ms 0.309 ms
```

même réseau
pas de routeur

```
{5}kernighan traceroute thales.e.ujf-grenoble.fr
traceroute to thales.e.ujf-grenoble.fr (152.77.4.2), 30 hops max, 40 byte packets
 1 imagate.imag.fr (195.221.225.254) 0.579 ms 0.366 ms 0.365 ms
 2 backujf-2-iron-ima.ujf-grenoble.fr (152.77.32.50) 85.543 ms 0.540 ms 0.511 ms
 3 admin-giga-2-backujf.ujf-grenoble.fr (152.77.32.114) 0.975 ms 0.773 ms 0.736 ms
 4 c3750-dsu-2-admin-giga.ujf-grenoble.fr (152.77.32.194) 0.708 ms 0.603 ms 0.576 ms
 5 thales.e.ujf-grenoble.fr (152.77.4.2) 0.535 ms 0.476 ms 0.463 ms
```

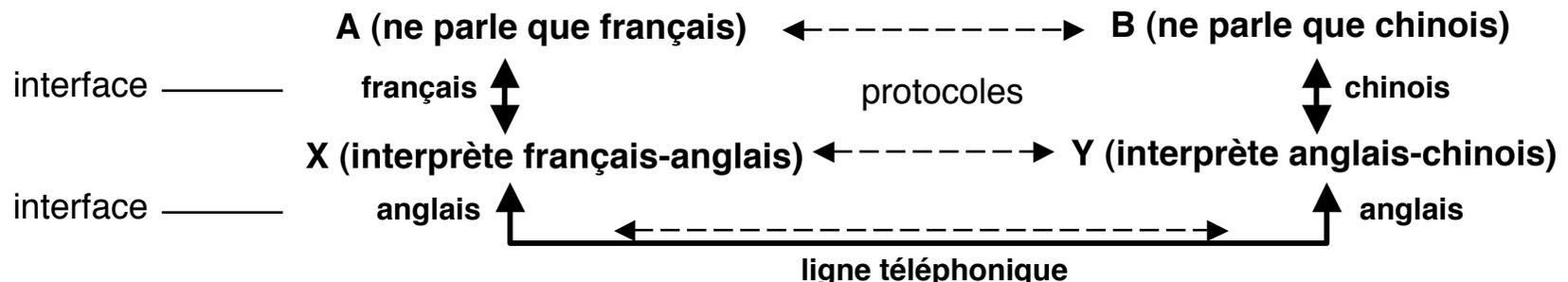
Connaître le chemin suivi (2)

```
{4}kernighan traceroute www.mit.edu
traceroute to www.mit.edu (18.7.22.83), 30 hops max, 40 byte packets
 1  imagate.imag.fr (195.221.225.254)  0.614 ms  0.362 ms  0.365 ms
 2  backujf-2-iron-ima.ujf-grenoble.fr (152.77.32.50)  0.649 ms  0.528 ms  0.692 ms
 3  tlc-gate-2-backujf.ujf-grenoble.fr (152.77.32.2)  0.862 ms  0.730 ms  0.719 ms
 4  smh-gate-2-tlc-gate.ujf-grenoble.fr (152.77.32.46)  2.896 ms  0.725 ms  0.692 ms
 5  r-campus.grenet.fr (193.54.185.120)  0.882 ms  0.811 ms  0.741 ms
 6  tigre1.grenet.fr (193.54.184.33)  0.854 ms  0.799 ms  0.814 ms
 7  grenoble-g3-0.cssi.renater.fr (193.51.181.94)  0.721 ms  0.665 ms  0.645 ms
 8  lyon-pos13-0.cssi.renater.fr (193.51.179.237)  13.178 ms  13.048 ms  13.038 ms
 9  nri-b-pos5-0.cssi.renater.fr (193.51.179.129)  12.866 ms  12.735 ms  12.724 ms
10  renater-10G.fr1.fr.geant.net (62.40.103.161)  13.152 ms  13.157 ms  13.177 ms
11  fr.uk1.uk.geant.net (62.40.96.90)  20.460 ms  20.385 ms  20.392 ms
12  uk.ny1.ny.geant.net (62.40.96.169)  89.096 ms  89.155 ms  89.182 ms
13  198.32.11.61 (198.32.11.61)  98.717 ms  89.140 ms  89.555 ms
14  noxgs1-PO-6-0-NoX-NOX.nox.org (192.5.89.9)  94.468 ms  94.342 ms  94.206 ms
15  noxgs1-PEER-NoX-MIT-192-5-89-90.nox.org (192.5.89.90)  106.598 ms  94.445 ms
    94.500 ms
16  W92-RTR-1-BACKBONE.MIT.EDU (18.168.0.25)  95.237 ms  94.869 ms  94.714 ms
17  WWW.MIT.EDU (18.7.22.83)  95.243 ms  94.605 ms  95.076 ms
{5}kernighan
```

Notions de protocole et d'interface (1)

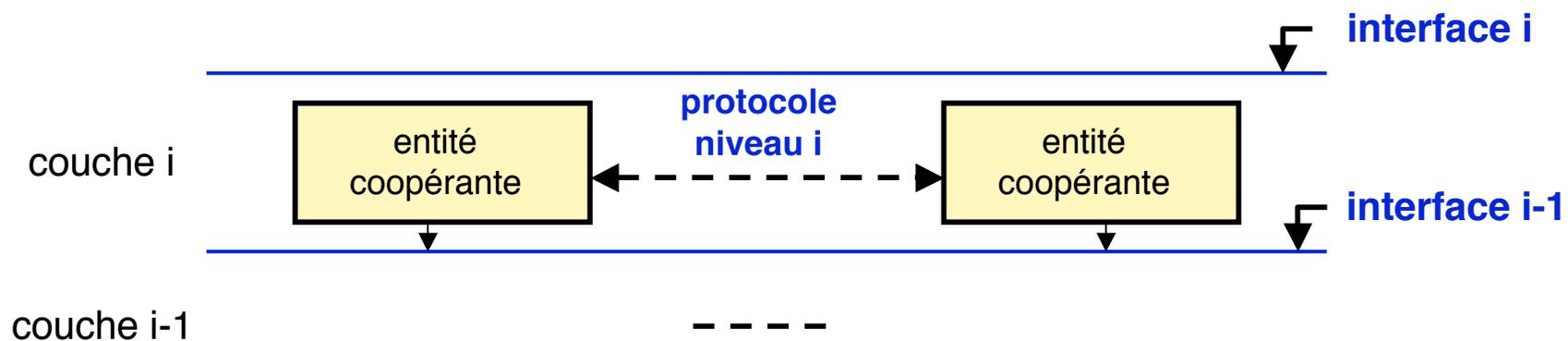
- L'exemple de la requête sur le web permet d'identifier divers "niveaux" d'échange entre le client et le serveur
 - ◆ le niveau de l'application : le client clique sur un lien, le serveur renvoie une page web
 - ◆ le niveau des messages : le client envoie un message contenant une URI, le serveur renvoie un message contenant un fichier HTML
 - ◆ le niveau des paquets : le message du client est découpé en paquets, les différents routeurs du réseau les acheminent vers le serveur (idem pour le retour)
 - ◆ le niveau de la transmission des bits : pour envoyer les paquets, chaque bit (0 ou 1) est transmis comme un signal électrique sur une ligne.
 - ◆ chaque niveau utilise les fonctions du niveau inférieur
- Les notions de protocole et d'interface visent à représenter ce mode de fonctionnement

Analogie : deux personnes peuvent dialoguer même si elles ne parlent pas la même langue

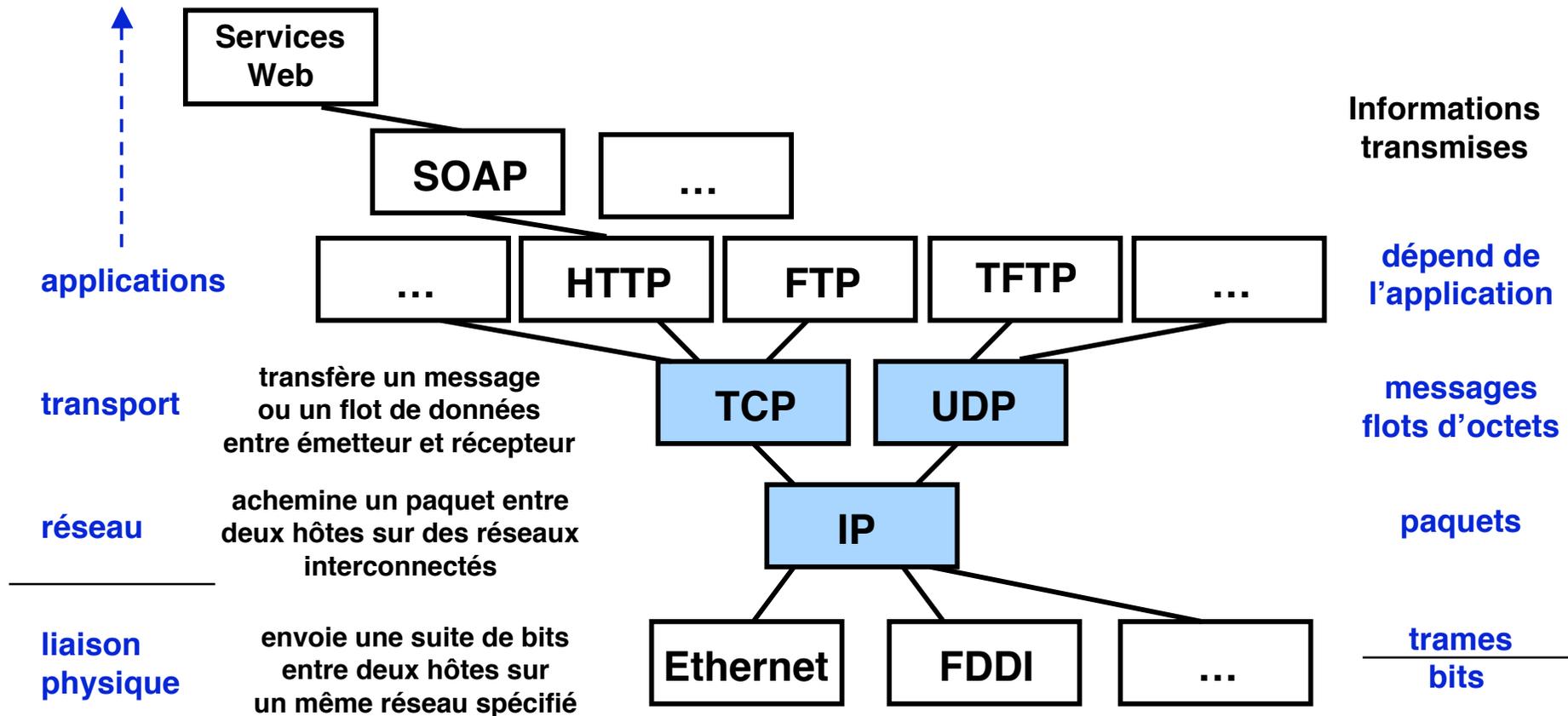


Notions de protocole et d'interface (2)

- Interface (d'un service) : ensemble de fonctions (logicielles ou matérielles) et de règles d'accès utilisables pour accéder au service
- Protocole : ensemble de conventions définissant les échanges entre des entités qui coopèrent pour réaliser un service
- Relations entre protocoles et interfaces
 - ◆ une interface définit l'accès à un service, un protocole définit la réalisation d'un service
 - ◆ la construction d'un protocole utilise souvent des protocoles de niveau inférieur (plus élémentaires), en accédant à leurs interfaces
- Protocoles en couches



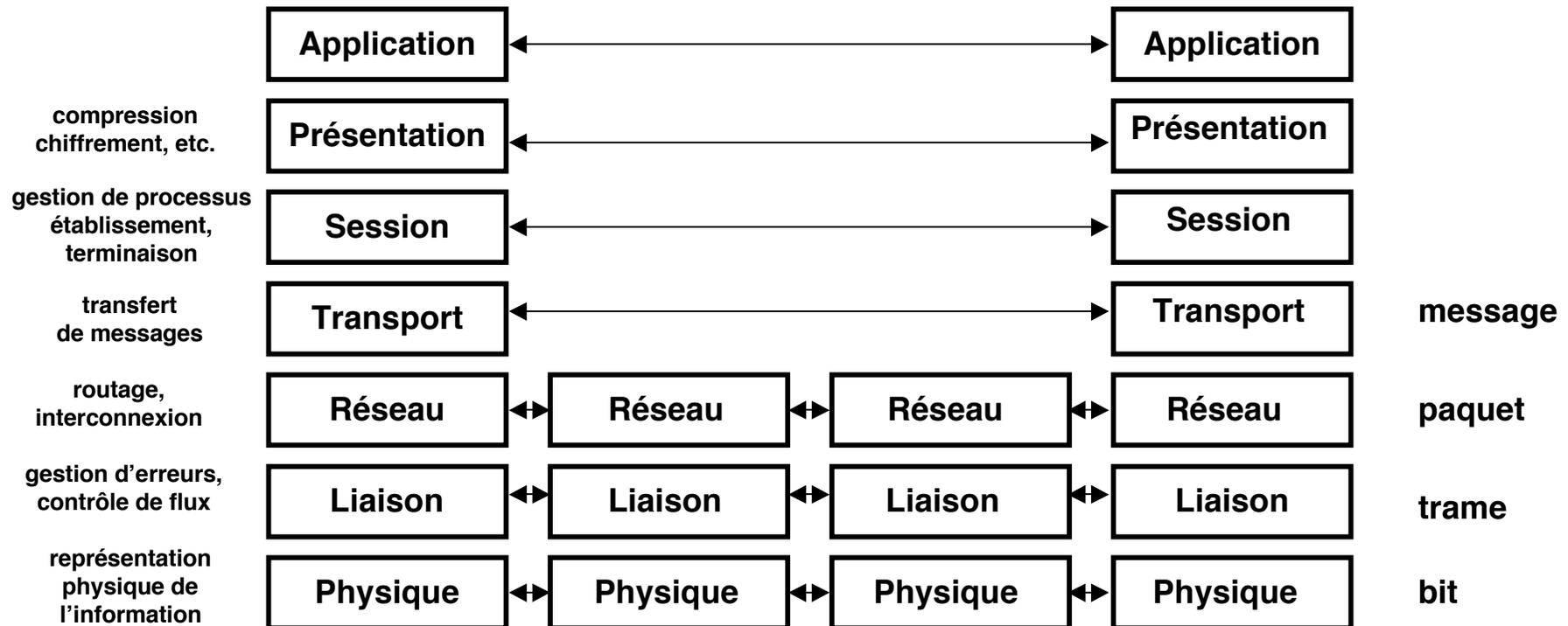
Les protocoles de l'Internet



HTTP : *HyperText Transfer Protocol* : protocole du Web
TFTP, FTP : (*Trivial*) *File Transfer Protocol*) : transfert de fichiers
TCP : *Transmission Control Protocol* : transport en mode connecté
UDP : *User Datagram Protocol* : transport en mode non connecté
IP : *Internet Protocol* : Interconnexion de réseaux, routage

Les protocoles normalisés de l'ISO (*International Standards Organisation*)

Open Systems Interconnection (OSI)



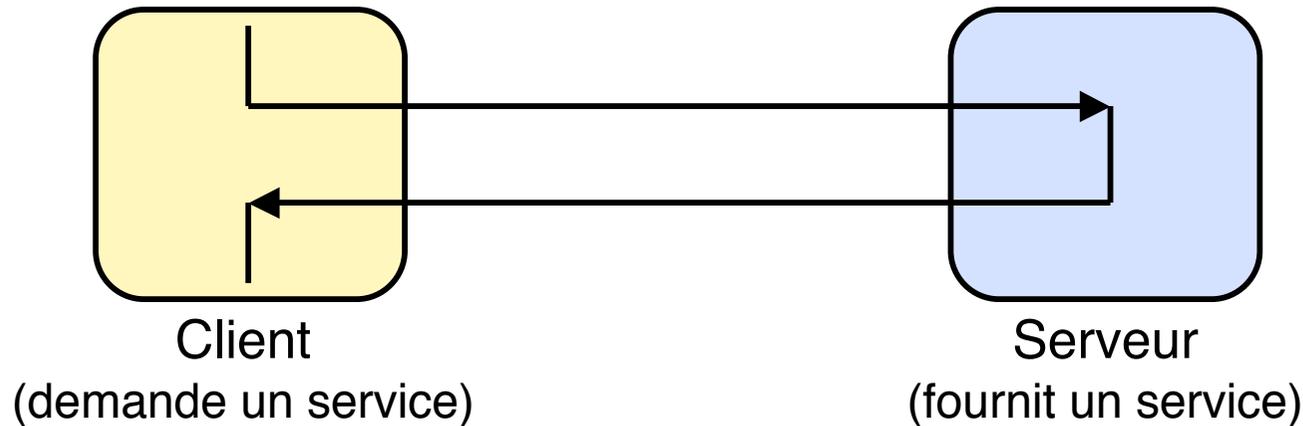
Les protocoles OSI servent plutôt de cadre de référence pour la définition des fonctions que de normes de réalisation. La normalisation de fait est autour de TCP/IP

Les services de l'Internet

- Services principaux (applications) disponibles sur l'Internet
 - ◆ Courrier électronique (`mail`) - protocole SMTP (*Simple Mail Transfer Protocol*)
 - ❖ inclut maintenant tous types de documents, grâce au format MIME (*Multipurpose Internet Mail Extensions*)
 - ◆ Forums de discussion (`news`) - protocole NNTP (*Network News Transfer Protocol*)
 - ◆ Transfert de fichiers (`ftp`) - protocole FTP (*File Transfer Protocol*)
 - ◆ Accès à une machine distante (`telnet`) - protocole *telnet* (terminal virtuel)
 - ◆ X-Window, service de fenêtres pour client distant
 - ◆ Accès au World Wide Web - protocole HTTP, formats HTML, XML, ...
 - ◆ Services divers utilisant le Web (bibliothèques virtuelles, commerce électronique, installation de logiciel, ...)
 - ◆ Extensions diverses pour la sécurité (SSL, SET, PGP, ...)
- Ces services utilisent les protocoles de transport (TCP, UDP), ainsi que le service de noms DNS (*Domain Name Service*)



Le réseau vu de l'utilisateur (1)

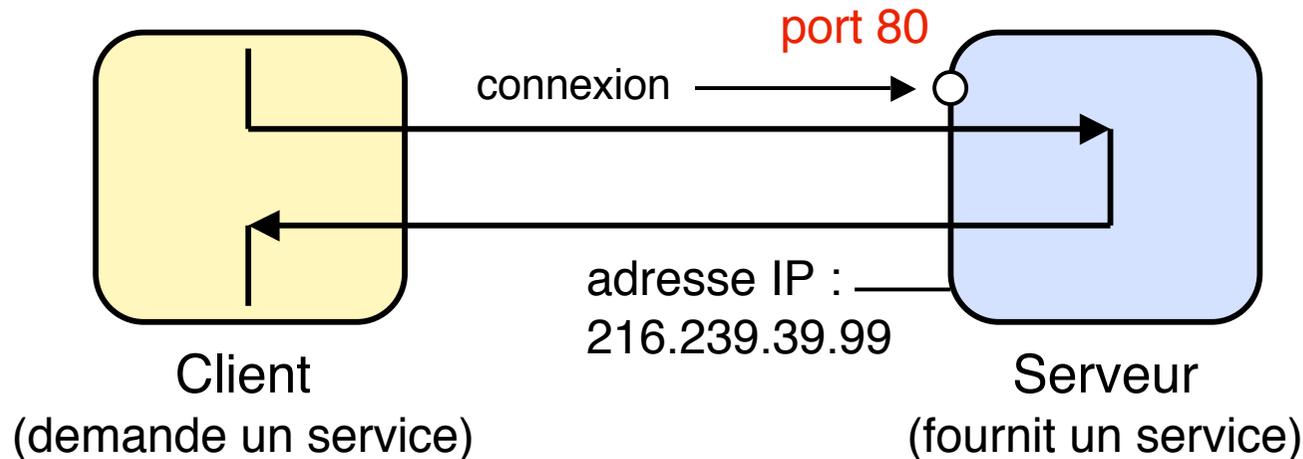


Le schéma **client-serveur** a été vu en TD pour des processus sur une même machine. Ce schéma se transpose à un réseau, où les processus client et serveur sont sur des machines différentes.

Pour le client, un service est souvent désigné par un nom symbolique (par exemple mail, <http://...>, telnet, etc.). Ce nom doit être converti en une adresse interprétable par les protocoles du réseau.

La conversion d'un nom symbolique (par ex. <http://www.google.fr>) en une adresse IP (216.239.39.99) est à la charge du service DNS (vu plus loin)

Le réseau vu de l'utilisateur (2)



En fait, l'adresse IP du serveur ne suffit pas, car le serveur (machine physique) peut comporter différents services; il faut préciser le service demandé au moyen d'un **numéro de port**, qui permet d'atteindre un processus particulier sur la machine serveur.

Un numéro de port comprend 16 bits (0 à 65 535). Les numéros de 0 à 1023 sont réservés, par convention, à des services spécifiques. Exemples :

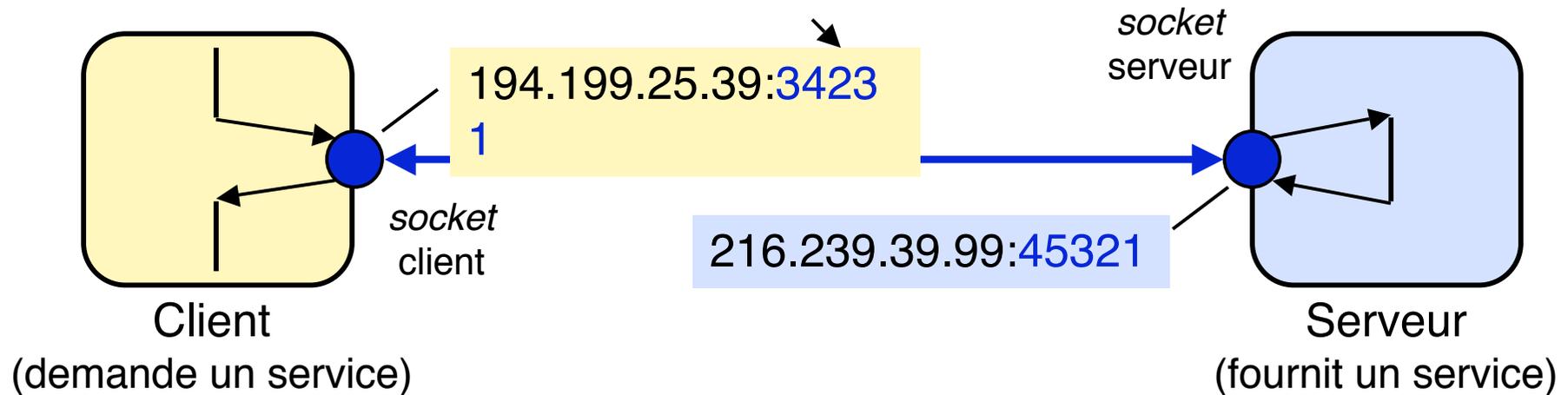
7 : echo

80 : serveur web

23 : telnet (connexion à distance)

25 : mail

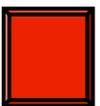
Le réseau vu de l'utilisateur (3)



Pour programmer une application client-serveur, il est commode d'utiliser les *sockets* (disponibles en particulier sous Unix). Les *sockets* fournissent une interface qui permet d'utiliser facilement les protocoles de transport TCP et UDP

Un *socket* est simplement un moyen de désigner l'extrémité d'une connexion, côté émetteur ou récepteur, en l'associant à un port. Une fois la connexion (bidirectionnelle) établie via des *sockets* entre un processus client et un processus serveur, ceux-ci peuvent communiquer en utilisant les mêmes primitives (*read*, *write*) que pour l'accès aux fichiers.

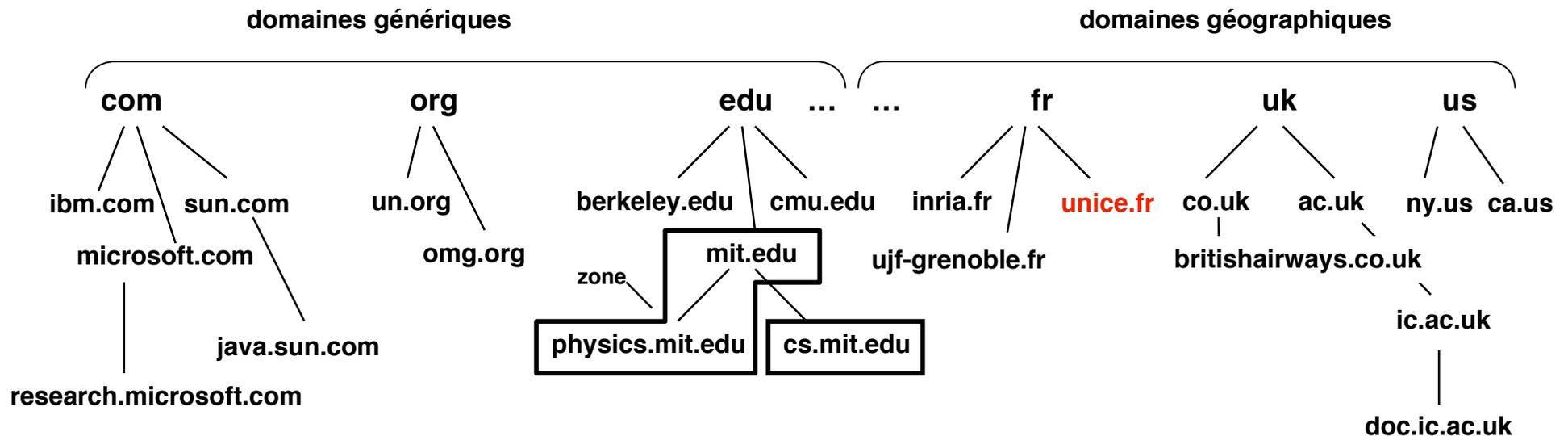
Les *sockets* sont étudiés en détail dans le prochain cours



DNS : noms de domaines (1)

- 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, 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)
- 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), ...
- 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), avec des autorités déléguées
 - ◆ Pour les domaines géographiques : une autorité nationale par pays - en France, l'AFNIC (www.nic.fr) - Association Française pour le Nommage sur l'Internet en Coopération)
 - ◆ Pour les domaines inclus : autorités locales (entreprise, administration, etc.)

DNS : noms de domaines (2)



■ 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

Service de noms sur l'Internet : problème et principes

■ Le problème

- ◆ Trouver l'adresse IP d'un site sur l'Internet, connaissant son nom symbolique
 - ❖ le DNS actuel résout aussi le problème inverse

■ Bref historique

- ◆ Premières années (jusqu'à quelques milliers de sites) : un serveur de noms unique pour tout l'Internet, gérant une table de correspondance nom symbolique - adresse IP
 - ❖ Inconvénients
 - ▲ mauvaise résistance aux pannes
 - ▲ goulot d'étranglement, pas de capacité de croissance
- ◆ À partir de 1985 : introduction d'un service réparti, DNS, qui a bien résisté depuis à une croissance exponentielle

■ Principes

- ◆ Algorithme de recherche décentralisé (pas de point de décision unique)
- ◆ Hiérarchie de serveurs calquée sur la hiérarchie des zones (voisines des domaines)
- ◆ Usage intensif de caches (informations dupliquées) et d'indicateurs (informations probablement valides permettant un accès rapide la plupart du temps)
 - ❖ toute information peut être obtenue par plusieurs voies
 - ❖ la validité de toute information peut être confirmée si nécessaire
- ◆ Ces règles favorisent aussi la tolérance aux fautes (service assuré en cas de pannes)

Fonctionnement de DNS (1)

- Toute zone (unité de gestion) comporte au moins deux serveurs de noms
 - ◆ Motivation de la duplication : pour performances, et surtout tolérance aux fautes
 - ◆ Chaque serveur maintient une table de correspondance Nom de domaine-Adresse IP
 - ◆ Plus précisément, une table est une collection d'enregistrements de la forme :

Nom, Valeur, Type, Classe, Durée de vie

- Principe de la recherche

- ◆ Ou bien le serveur contient l'adresse recherchée, ou bien il contient l'adresse d'un autre serveur qui a davantage de chances que lui de la connaître

- Contenu des tables du serveur de noms

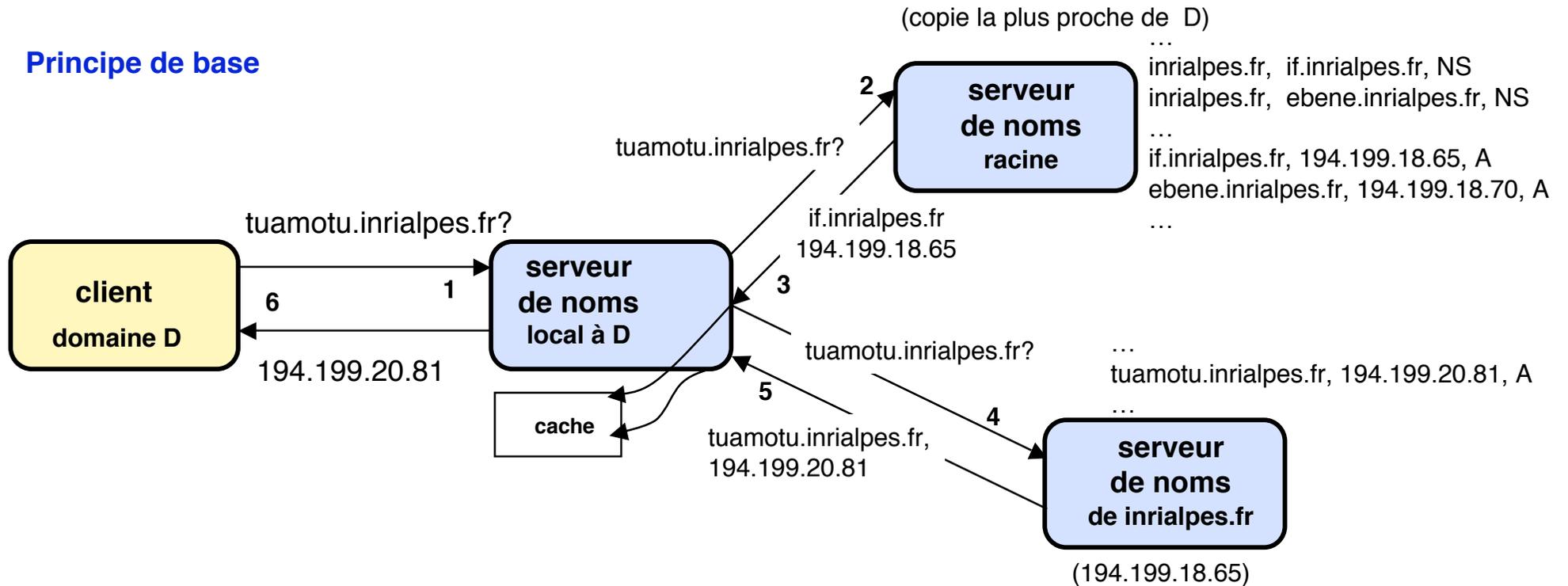
- ◆ *Type* : indique comment le champ *Valeur* doit être interprété
 - ❖ *Type = A* : la *Valeur* est une adresse IP
 - ❖ *Type = NS* : la *Valeur* est le nom de domaine d'un serveur de noms
 - ❖ *Type = CNAME* : la *Valeur* est le nom "canonique" d'un hôte (le nom de référence si alias)
 - ❖ autres : non traités ici (aide à la gestion du *mail*, etc.)
- ◆ *Classe* : IN (l'Internet), presque toujours (DNS utilisé comme annuaire pour d'autres services de noms)
- ◆ *Durée de vie* (TTL ou *Time To Live*) : durée de validité de l'enregistrement
 - ❖ Application d'un principe général : toute information doit être périodiquement validée

Fonctionnement de DNS (2)

- Le serveur racine contient les noms des serveurs de noms des domaines du second niveau et les adresses IP de ces serveurs
 - <inrialpes.fr, if.inrialpes.fr, NS, IN, ...> -- if est un serveur de noms du domaine inrialpes.fr
 - <if.inrialpes.fr, 194.199.18.65, A, IN, ...> -- l'adresse IP de if est 194.199.18.65
 - ◆ Le serveur racine est très largement dupliqué
 - ❖ Plus un serveur est "haut", plus souvent on en a besoin, plus il a de copies, et moins souvent il change (long TTL)
- De même, chaque serveur de zone contient
 - ◆ Des couples (noms, adresses) d'hôtes appartenant aux domaines contenus dans la zone
 - ◆ Éventuellement des couples (noms, adresses) de serveurs de noms de sous-domaines inclus
 - ◆ Par exemple, if.inrialpes.fr contient
 - <tuamotu.inrialpes.fr, 194.199.20.81, A, IN, ...>
- Pour amorcer la résolution
 - ◆ Tout hôte doit connaître l'adresse IP d'un serveur de noms local (dans sa zone) - il est recommandé de connaître deux ou trois adresses, pour la tolérance aux fautes
 - ◆ Ces adresses sont fournies aux utilisateurs de l'hôte par l'administrateur de sa zone (ou son fournisseur d'accès à l'Internet)
 - ◆ Elles sont inscrites "à la main" dans les tables de configuration pour l'accès à l'Internet

Fonctionnement de DNS (3)

Principe de base



Mécanismes d'accélération

Tout serveur conserve dans un cache les couples (nom, adresse IP) récemment résolus
Dans la pratique, il est rare de consulter plus de deux serveurs

Autorités

Tous les serveurs ne sont pas mis à jour en permanence de toutes les modifications. Certains seulement le sont, et font autorité
d'où les réponses "*authoritative answer*" ou "*non authoritative answer*"

Utiliser DNS (1)

DNS est automatiquement utilisé par toute application sur l'Internet utilisant des noms de domaines. Mais on peut aussi y accéder directement

Au niveau du langage de commande (*shell*)

Commande `dig` (*Domain Information Groper*) - voir man

```
dig www.google.com
```

```
dig -x 194.199.25.39 # -x : adresse vers nom
```

Au niveau des appels systèmes Unix

```
#include <netdb.h>
```

```
struct hostent *gethostbyname(const char *name);
```

```
struct hostent *gethostbyaddr(const char *addr,  
    int len, 0);
```

Utiliser DNS (2)

```
{27}kernighan dig www.google.com
```

```
;; <<>> DiG 8.3 <<>> www.google.com
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 0, ADDITIONAL: 0
;; QUERY SECTION:
;;      www.google.com, type = A, class = IN
```

```
;; ANSWER SECTION:
```

```
www.google.com.      23S IN CNAME   www.google.akadns.net.
www.google.akadns.net. 2m22s IN A      216.239.59.147
www.google.akadns.net. 2m22s IN A      216.239.59.99
www.google.akadns.net. 2m22s IN A      216.239.59.104
```

← nom canonique

← entrées multiples

```
;; Total query time: 4 msec
;; FROM: kernighan to SERVER: default -- 195.221.225.20
;; WHEN: Sun Feb 27 11:22:45 2005
;; MSG SIZE sent: 32 rcvd: 115
```

```
{28}kernighan dig www.google.com
```

Utiliser DNS (3)

```
{11}kernighan dig -x 194.199.25.39
; <<>> DiG 8.3 <<>> -x
;; res options: init recurs defnam dnsrch
;; got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 4
;; QUERY SECTION:
;;     39.25.199.194.in-addr.arpa, type = ANY, class = IN
;; ANSWER SECTION:
39.25.199.194.in-addr.arpa. 1h29m48s IN PTR yttrium.inrialpes.fr.

;; AUTHORITY SECTION:
25.199.194.in-addr.arpa. 1h29m48s IN NS if.inrialpes.fr.
25.199.194.in-addr.arpa. 1h29m48s IN NS imag.imag.fr.
25.199.194.in-addr.arpa. 1h29m48s IN NS ebene.inrialpes.fr.

;; ADDITIONAL SECTION:
if.inrialpes.fr.          1h20m23s IN A    194.199.18.65
imag.imag.fr.            2H IN A        129.88.30.1
imag.imag.fr.            2H IN AAAA     2001:660:5301:1e::101
ebene.inrialpes.fr.     29m35s IN A        194.199.18.70

;; Total query time: 4 msec
;; FROM: kernighan to SERVER: default -- 195.221.225.20
;; WHEN: Sun Feb 27 11:27:24 2005
;; MSG SIZE sent: 44 rcvd: 215
{12}kernighan
```

} serveurs
de noms



Résumé de la séance 7

- Introduction aux réseaux informatiques
 - ◆ Types et caractéristiques (LAN, MAN, WAN, mobiles, etc.)
 - ◆ Performances : latence, débit. Qualité de service
- Protocoles et interfaces
 - ◆ Fonctionnement d'ensemble d'un réseau
 - ◆ Notions de protocole et d'interface
 - ◆ Protocoles en couches, fonctions des principales couches
- Service de noms de l'Internet : protocole DNS
 - ◆ Désignation symbolique sur l'Internet : domaines
 - ◆ Exemple de gestion décentralisée, capacité de croissance
- Plan de la suite (les 2 prochaines séances)
 - ◆ Utilisation de l'interface de transport : les *sockets* Unix (TCP)
 - ◆ Un service de l'Internet : le World Wide Web - organisation, protocoles

Bibliographie sommaire sur les réseaux

A. Tanenbaum, *Les réseaux*, 3-ème édition, Dunod, 1999

L. Peterson, B. S. Davie, *Computer Networks, a systems approach*, 3rd edition, Morgan Kaufmann, 2003 (il existe une traduction française, mais seulement pour la 1-ère édition actuellement)

J. F. Kurose, K. W. Ross, *Computer Networking, a top-down approach featuring the Internet*, Addison-Wesley, 3rd ed. 2004 (il existe une traduction française)