R102: Introductions aux Réseaux Informatiques

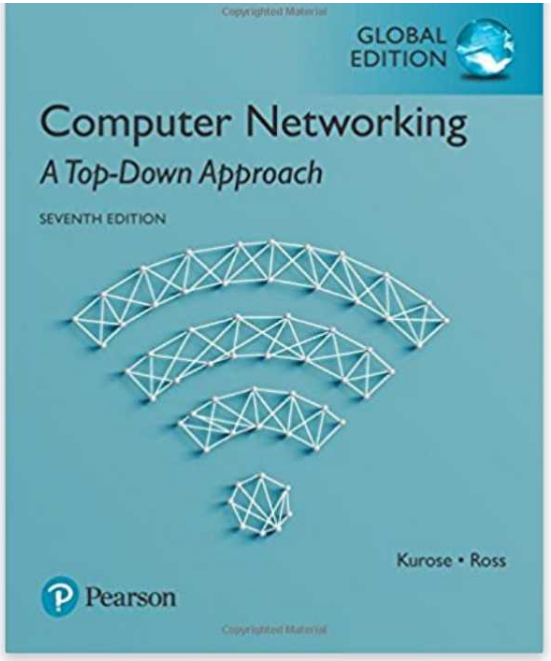
G. Urvoy-Keller

Sources de ce document

Computer Networking: A Top-Down Approach (Anglais)

Broché – 21 novembre 2016

de James Kurose (Auteur), Keith Ross (Auteur)



Agenda

- M1101 : Tendances de trafic dans l'Internet
- M1101 : Les acteurs : Cloud, data centers, ISP, CDN, mobile
- M1101 : Identification des besoins lors du transfert réseau
- Formalisation
  - Fonctions d'un réseau
  - Protocole et modèle en couche
  - Préparation et transfert des données
  - Les besoins des applications
  - Unicast/broadcast
  - Coeur du réseau
  - Réseaux d'accès

Agenda M1104

- Couche Application : HTTP, DNS
- Couche Transport : TCP, UDP
- Couche Réseau : IP
- Couche Liaison : principes généraux et ARP

Formalisation

Fonctions d'un réseau

Protocole et modèle en couche

Préparation et transfert des données

Les besoins des applications et les services fournis par le réseau

Unicast/broadcast

Coeur du réseau

Réseaux d'accès

Réseaux et Télécommunications

	Réseaux informatiques	*Télécommunications*
Historiquement	Données à l'usage de machine. Plus généralement : transport de données numériques	Transport de la voix, de l'image ... Information analogique.
Actuellement	Les mêmes infrastructures techniques sont utilisées pour transporter : Des données informatiques Du son De l'image De la téléphonie	

- Dans le département R & T :
  - Télécommunications : Comment transporter l'information sur un support physique
    - représentation des données
    - études des supports de transmission (câble, fibres optiques, ondes électromagnétiques)
  - Réseaux : Etude des équipements réseaux et des protocoles permettant à n (n grand) machines de communiquer.

PAN : Personal Area Network

- PAN (Personal Area Network)
- tailles : 10 mètres

- Interconnexion des équipements personnels :
  - périphériques (clavier, souris …)
  - agenda électronique
  - téléphone portable
  - Capteur biométriques (personnes malades)

- Technologie : liaison sans fil (radio, infrarouge) faible débit
  - Ex : Bluetooth

LAN : Local Area Network

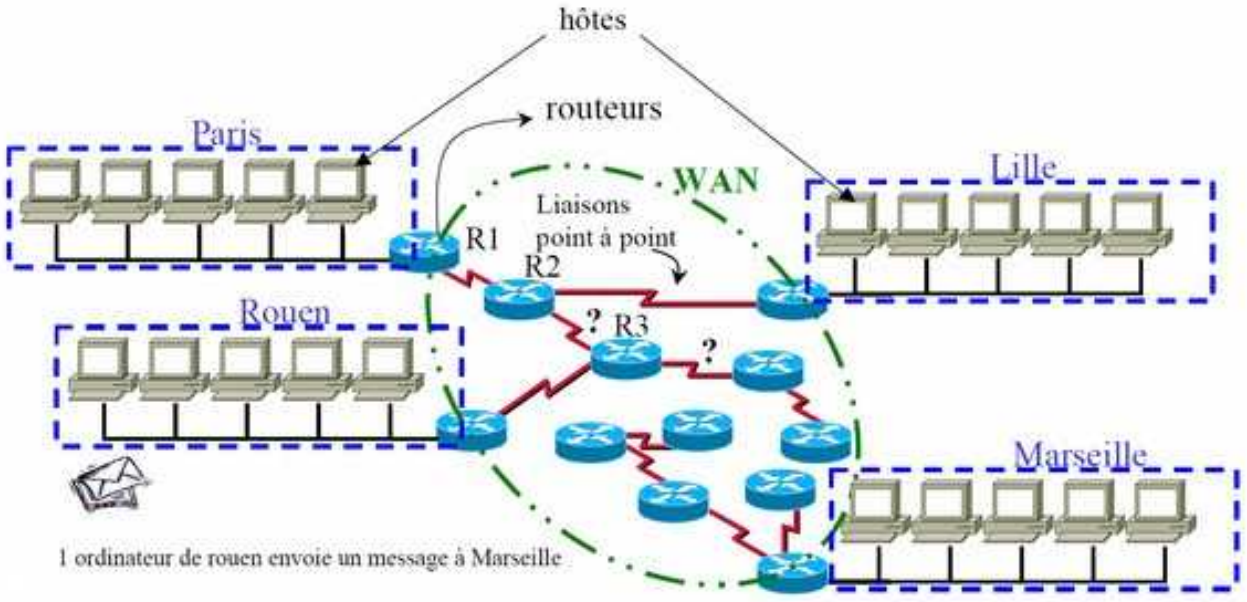
- Taille : Quelques centaines/milliers de mètres
- Caractéristiques :
  - réseau haut débit (actuellement 100 Mbits/s à 1Gbit/s, voire 10 Gbit/s)
  - faible taux d'erreur
- Technologies utilisées :
  - Ethernet (IEEE 802.3) et Wifi (IEEE 802.11)

MAN : Metropolitan Area Network

- Taille : quelques dizaines de km
- Technologie :
  - Technologies spécifiques.
    - SDH (Synchronous Digital Hierarchy)
    - Des variantes d' Ethernet soit directement sur fibre optique (natif) soit sur une architecture SDH.
    - Réseau câblé.
  - M3102 et M3103

WAN : Wide Area Network

- Taille : 1 pays, 1 continent, la terre entière
- Caractéristique :
  - Technologie : Les commutateurs ou routeurs sont interconnectés par des liaisons point à point.



Objectifs d’un réseau informatique

- Objectifs :
  - Communiquer
  - Partager des ressources
  - Fiabilisé/sécurisé l'information
  - Maintenance simplifiée du parc de machines

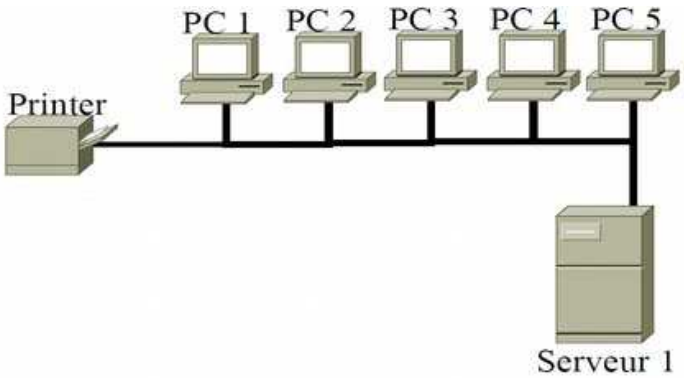
- Réseau informatique :
  - Entreprise : intranet
  - Internet : réseaux d'opérateurs
  - Chez les particuliers (à la maison)

Communiquer

- Mettre à disposition des informations
  - Site Web
  - Bases de données
- Echanger des informations:
  - Mail
  - Forum, blog
  - Chat
  - Visioconférence

Partager les ressources

- Partager : rendre accessible à chaque machine du réseau
- Ressources :
  - des programmes
    - Ex : un serveur hébergeant un logiciel de PAO
  - des données indépendamment de leur localisation physique
    - Ex : un serveur de fichier accessible sur le réseau de l'entreprise
  - des équipements
    - Ex : imprimante



Fiabilité, Sécurité, Maintenance

- Fiabilité
  - Sauvegarde : les données d'une entreprise sont centralisées sur une seul serveur dont le contenu est sauvegardé toutes les nuits.

- Sécurité
  - Contrôle de l'accès aux données : seuls les utilisateurs authentifiés ont accès
    - aux données
    - aux matériels : un utilisateur en entreprise ne peut pas installer de logiciels sur son PC

- Maintenance :
  - Surveillance des composants (serveurs/machines/équipement)
  - Mise à jour des logiciels et systèmes d'exploitation

Qu’est-ce-qu’un protocole?

Modèle en couche

Qu'est-ce qu'un protocole?

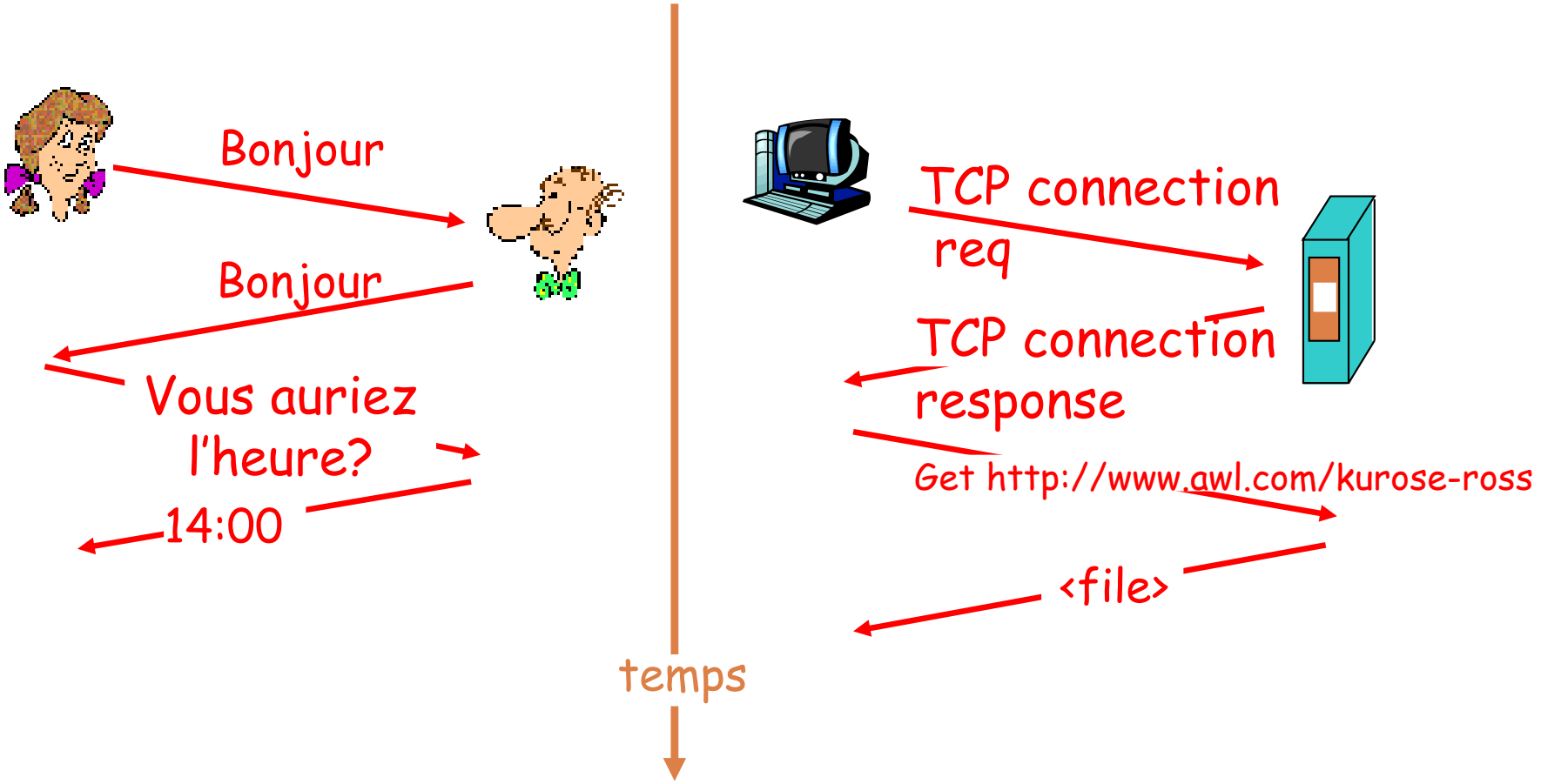
- Protocoles humain:
  - Pour demander l'heure "Quelle heure est-il, síl vous plait?"
  - Lors d'une rencontre « Bonjour/Bonjour »
- … des messages spécifiques envoyés
- … des actions spécifiques effectuées après la réception de ces messages ou d'autres évènements
  - Ex: si pas de réponse au « bonjour », on essaie encore une fois après quelques secondes
- Protocoles Réseau:
  - Entre machines plutôt qu'être humains
- Toutes les communications entre machines sont dictées par des protocoles

Qu’est-ce qu’un protocole?

- Protocoles définissent :
  - le format
    - Ex: d’abord adresse émetteur sur 16 bits, puis adresse récepteur sur 16 bits puis données sur 0-1400 octets
  - l’ordre d’envoi et de réception entre entités réseaux (machine, serveur, imprimante, routeurs)
  - et les actions à prendre pour l’envoie et la réception des données

Qu’est-ce qu’un protocole?

Un protocole humain et un protocole entre machines:



Couches protocolaires

Les réseaux sont complexes

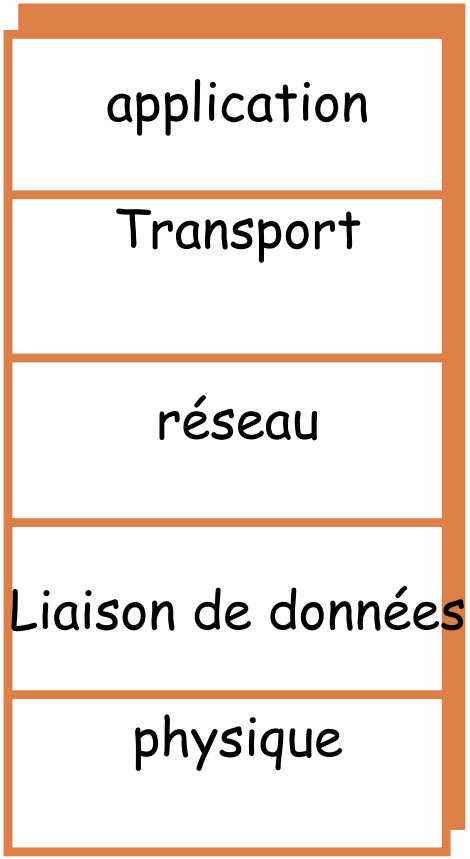
- De nombreuses « pièces » :
  - machines
  - routeurs
  - Liens divers
  - applications
  - protocoles

Question:

Peut-on trouver une représentation adéquate?

Couches Protocolaires de l'Internet

- application: applications réseaux
  - Ex: HTTP
- transport: transferts de données d'hôte à hôte
  - TCP, UDP
- réseau: routage des paquets de source à destination
  - IP, protocoles de routage
- Liaison de données: transferts de données entre éléments
  - Ex: Ethernet, Wifi
- physique: bits "sur le fil"



Couches Protocolaires de l'Internet

- application: les applications que vous utilisez directement (Firefox, thunderbird pour le mail, WhatsApp, etc.) et des applications utilisées par ces applications (serveur DNS, serveur de temps)
- transport: permet de localiser l'application dans une machine ayant une adresse IP
- réseau: routage des paquets de source à destination
- liaison de données: Contrôle la couche physique, par exemple couche liaison Wifi décide quand
- physique: envoie le signal sur ordre couche liaison

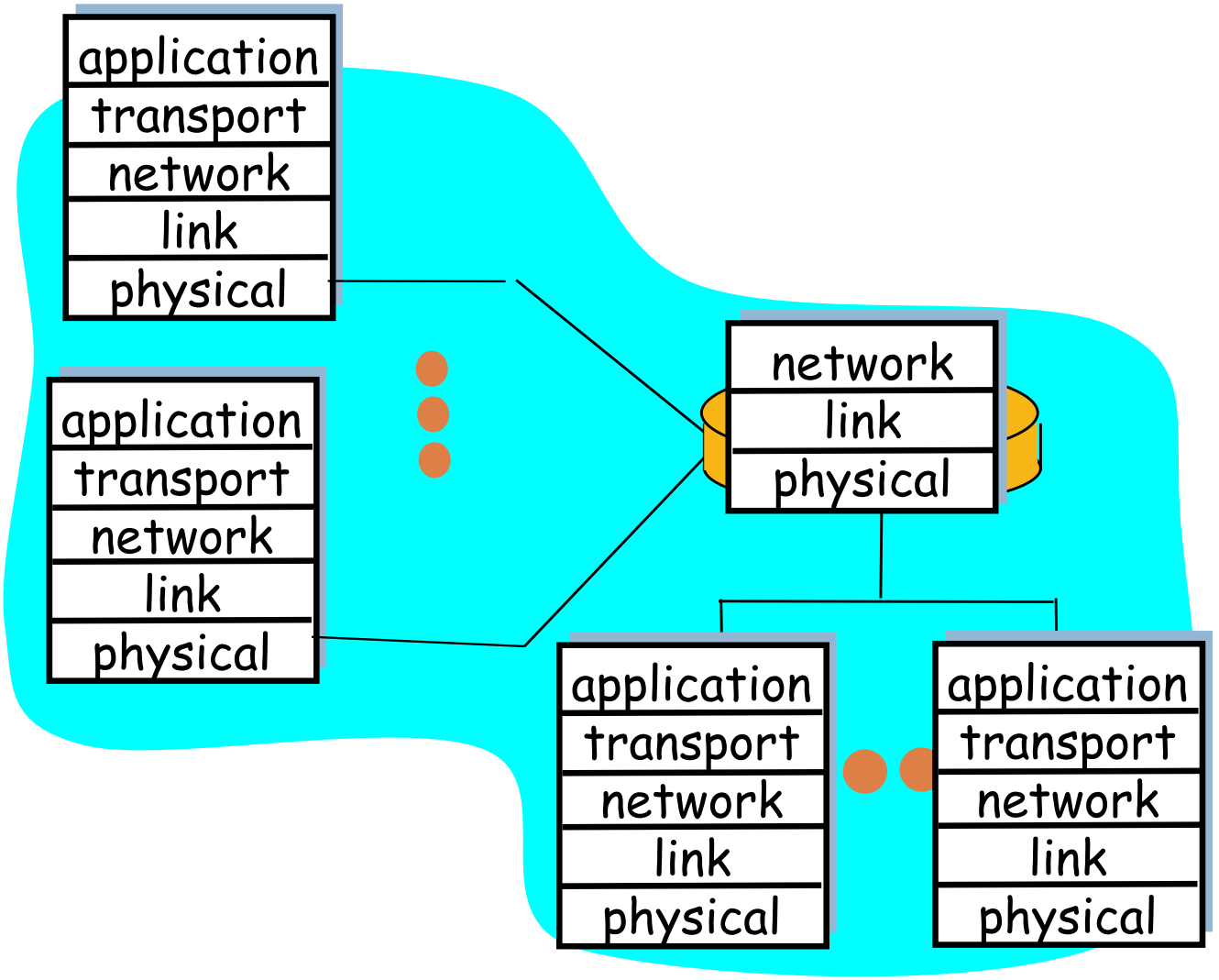
Pourquoi faire des couches?

- Pour gérer des systèmes complexes
  - permet l'identification explicite des fonctions et des relations entre fonctions
  - Modèle pour permettre des discussions (organismes de normalisation)
- Modularité simplifie la mise à jour
  - Des changements internes dans une couche n'affecte pas les autres couches

Couches : communications logiques

Chaque couche:

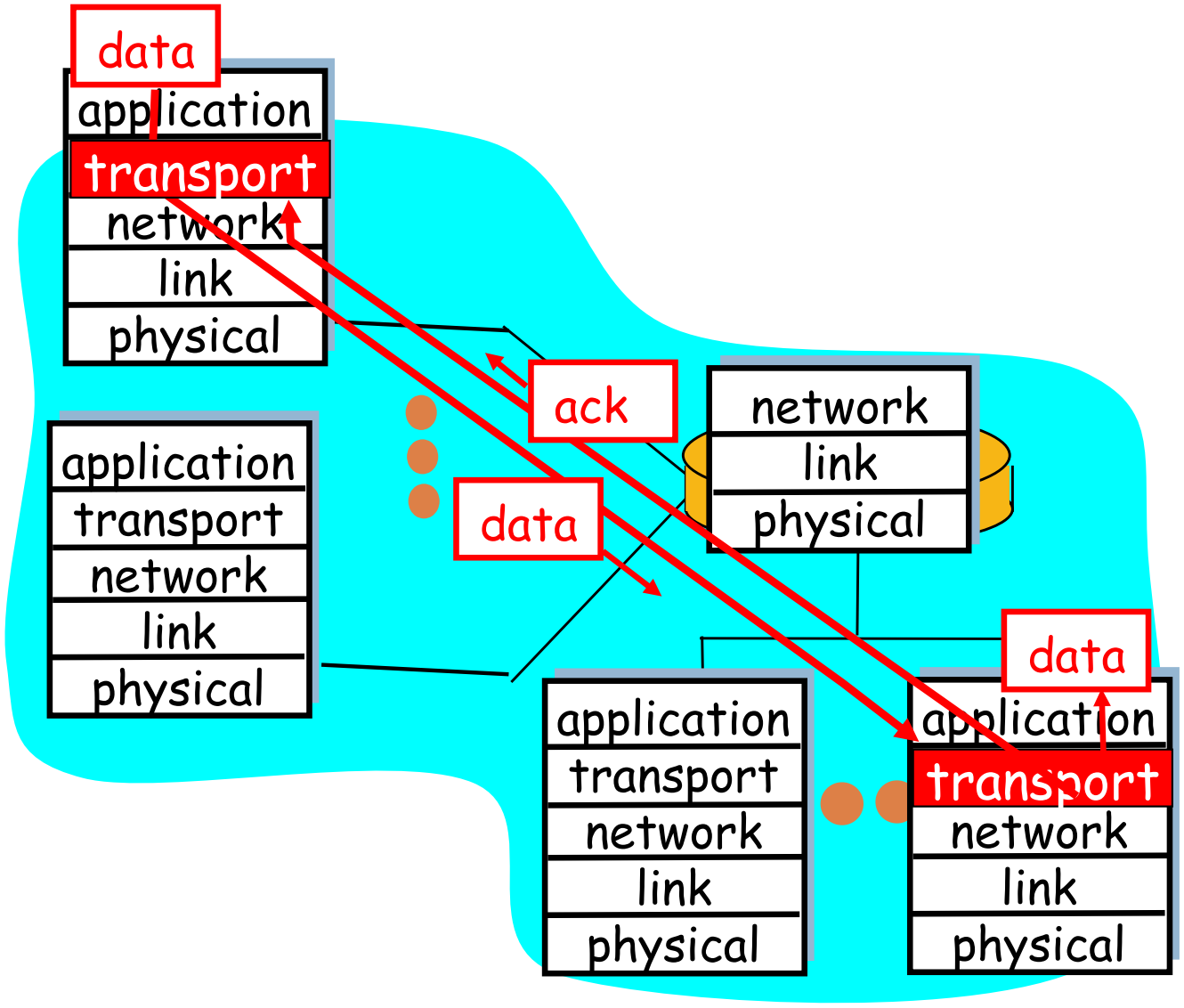
- distribuée
- “entités” implémentent couches à chaque noeud
- Entités exécutent des action, échangent des messages avec entité distante



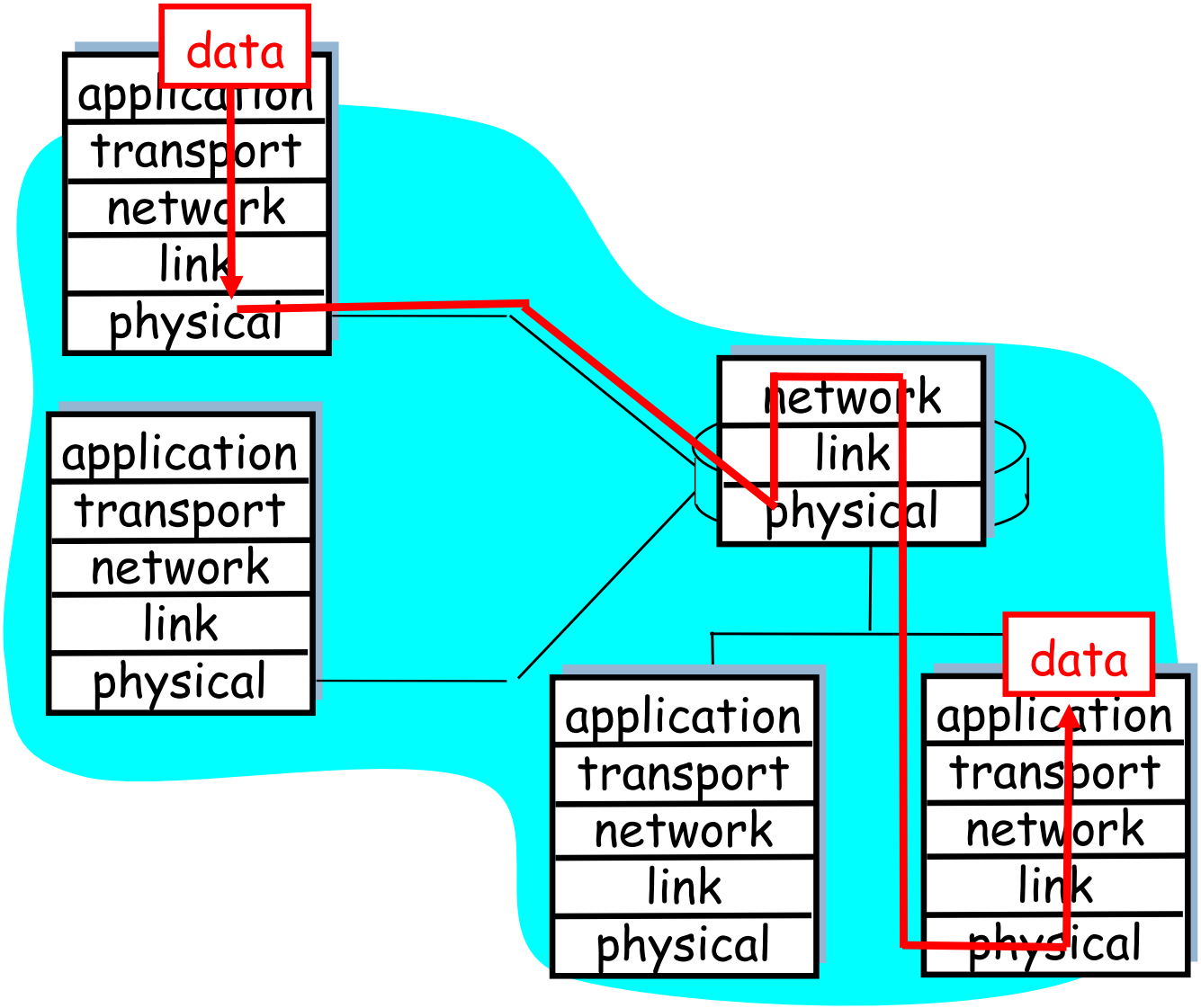
Couches: communications logiques

Ex.: couche transport

- Prend les données de l'appli.
- Ajoute adressage, ajoute informations pour transfert fiable et forme un paquet appelé datagramme
- Envoie le datagramme à la machine distante
- Attend l'accusé de réception



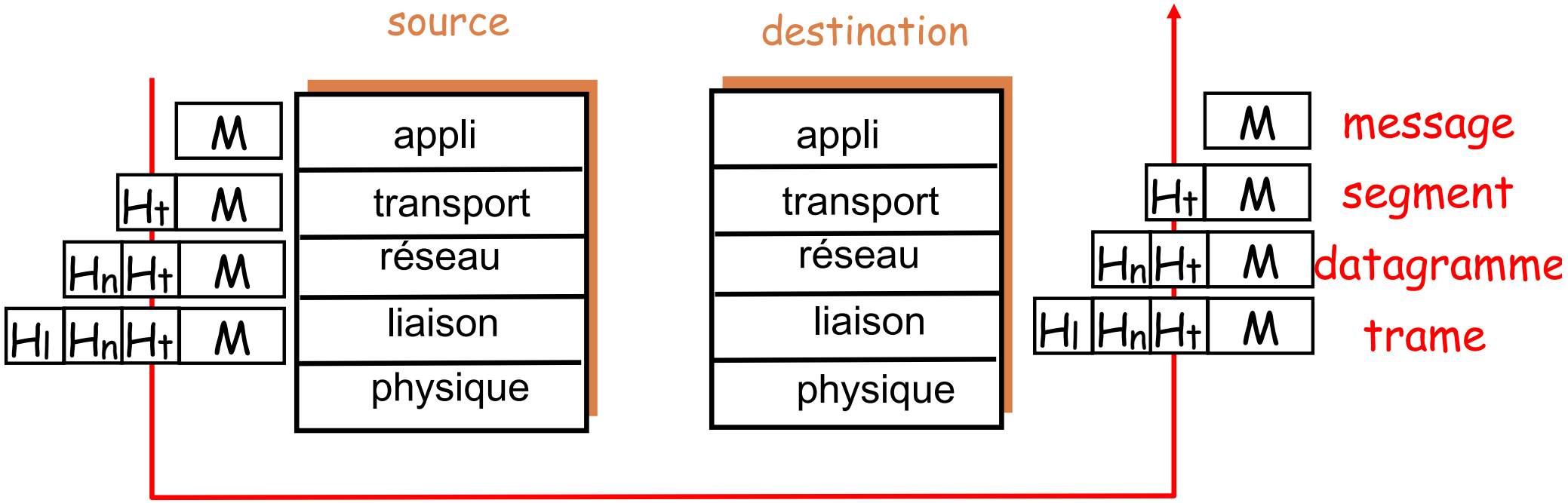
Couches : communications physiques



Couches : protocoles et données

Chaque couche prend les données de la couche supérieure

- Ajoute un « en-tête » utilisé par l'entité paire (H comme header en anglais ci-dessous)
- Donne les données à la couche inférieure



Couches : protocoles et données

- Local : assure le transfert entre 2 machines
  - Ex : wifi gère entre le point d'accès Wifi et votre PC
- De bout en bout : on donne les données sans avoir à connaître le routage
  - Ex : client Web et serveurWeb

- Couches locales : physique/liaison de données et réseau
  - Les seules couches des routeurs de l'Internet car il n'a besoin que de décoder les adresses réseaux
- Couches de bout en bout : transport/application
  - Présentent sur les machines/serveurs seulement

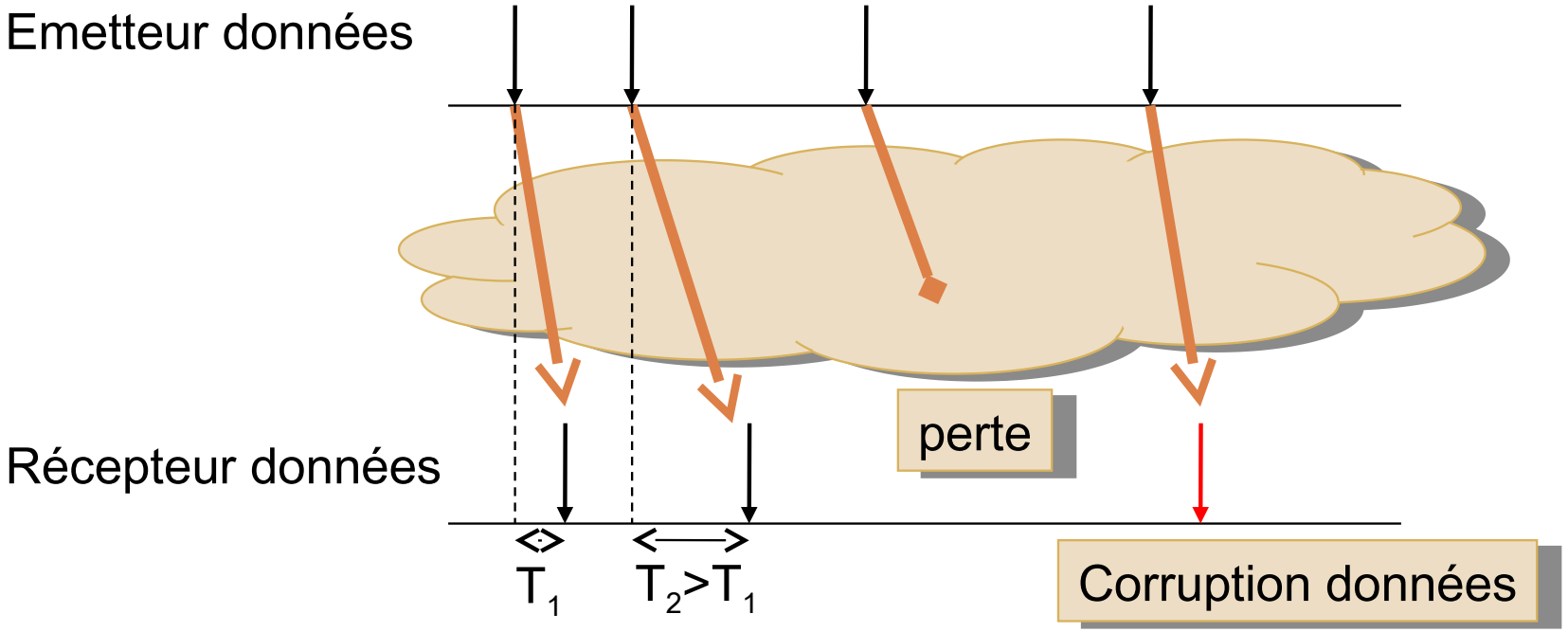
Caractéristiques réseaux et besoins des applications

Avant le transport sur un réseau

- Avant le transport :
  - Les informations sont numérisées = transformées en une suite de 1 et 0
  - Vrai pour Internet mais vrai aussi pour le réseau téléphonique filaire et le mobile !
- Pendant le transport. Le réseau offre :
  - Débit de transfert
  - Temps de transfert
  - Taux de pertes

Débit/Délai/Pertes ou Erreurs

- Le réseau transmet les informations (paquets pour l'internet, échantillon de voix pour réseau téléphonique) :
  - A une certaine vitesse (= débit)
    - Ex : ADSL à 20 Mbit/s ou Wifi à 54 Mbits/s ou mobile à 1,5 Mbits/s
  - en un temps variable : temps de transferts T1, T2
  - avec des erreurs (corruptions symboles) ou des pertes



Caractéristiques réseau versus besoin des applications

- Téléphonie
  - Temps de transfert minimal
    - Ex: une conversation téléphonique au travers d'un satellite géostationnaire (500 ms d'Aller/Retour terre satellite) est très désagréable
  - Débit relativement faible, 30 kbit/s
  - Le temps de transferts doivent être tous les mêmes
    - Récurrence temporelle stricte
  - Si le réseau perd des échantillons de voix de manière aléatoire, pas de problème : le cerveau humain compense
- Téléphonie : débit faible, délai fixe perte, supporte (niveau faible de) pertes

Caractéristiques réseau versus besoin des applications

- Vidéo :
  - Enregistrée vs. Live
    - Live: Visioconférence, Télévision sur IP
    - Enregistrée: YouTube/Dailymotion et Vidéo à la demande sur Internet
  - Débit importante
    - Youtube/Dailymotion. Vidéo codées à plusieurs centaines de kbit/s
    - Télévision sur IP (en ADSL): plusieurs Mbit/s
  - Récurrence temporelle stricte
    - Utilisation de mémoire de stockage possible dans le cas de vidéo enregistrée seulement
  - Pertes : si une pixel manque, on affiche la moyenne de ses voisines
- Vidéo : débit moyen à élevé, délai fixe (surtout en live), pertes admissibles

Buffer
YouTube
Broadcast Yourself
Home
Videos
Channels
Community
Search
Upload
yael naim new soul clip
Views: 6,369,340
Share
Favorite
Playlists
Flag
Email
MySpace
Facebook
Commentary
Statistics & Info
Point de lecture
From: totoutardnet
Joined: 7 months ago
Videos: 1
Subscribe
Added: August 31, 2007
yael naim shows the first video clip of her new...
More From: totoutardnet
Related Videos
Yael Naim - Toxic
Yael Naim - Paris -
Yael Naim | New Soul
Yael Naim - New Soul
New soul Live Yael Naim & Lucie Star Academy
Promoted Videos

Caractéristiques réseau versus besoin des applications

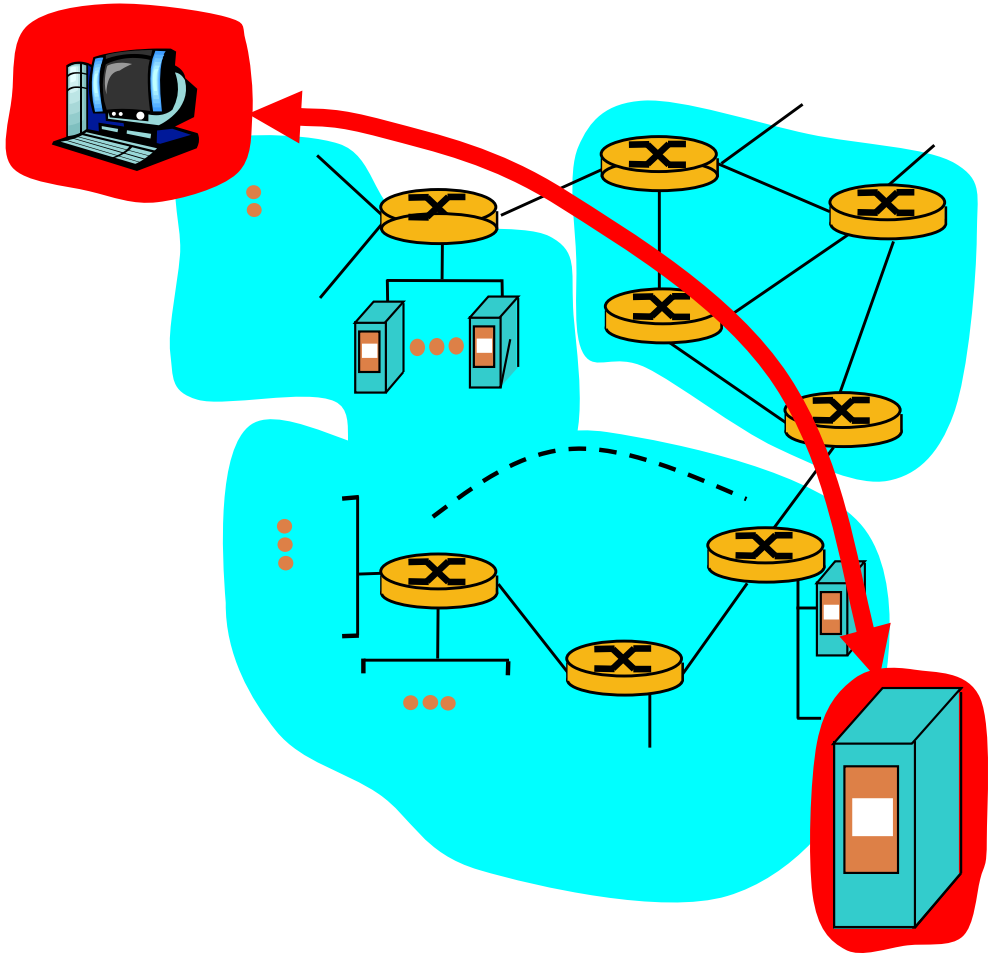
- Transfert de fichier (ex : mail ou back-up)
  - Aucune perte
  - Indifférent au délai et à ses variations
  - Indifférent au débit (tant qu'il n'est pas trop faible)

Services offerts par le réseaux

- Le réseau offre un service aux applications qui sont hébergées de manière distribuée sur les bords du réseaux
  - Ex : un serveur Web et un client Web
    - Forment une appli. distribuée
    - se parlent au travers du réseau : le réseau leur offre un service
- On distingue 2 types de services dans l'Internet offerts aux applications par la couche transport :
  - Orienté connexion
  - Sans connexion

Services

Client Web :
- Chrome
- Firefox
- Safari, etc



Serveur Web :
- www.google.fr
- www.unice.fr

Service orienté connexion

- Objectif : transfert de données entre 2 machines
- Etablissement de connexion avant l´envoie de données
  - Équivalent du “bonjour, bonjour” entre humains

Applications utilisant TCP:

- HTTP (Web), SMTP (messagerie électronique)

- Exemple : TCP dans l’ Internet (Transport Control Protocol)
  - Transferts d’ octets entre machines fiable :
    - Les pertes sont combattues à l´aide de retransmissions
  - Contrôle de congestion :
    - Emetteur ralentit si le réseau (qui fonctionne en commutation de paquets) est congestionné

Service sans connexion

- Objectif: transfert de données entre machines

- Exemple dans l' Internet : UDP - User Datagram Protocol :
  - On ne renvoie pas si perte
  - C'est à l'application de le faire
  - Rapide (pas de bonjour/bonjour)

Applications utilisant UDP:

- Téléphonie sur Internet, DNS (conversion nom/ addresse)

Algorithme de communication d'une machine connectée à Internet (PC, mobile, tablette, XBox)

- Cas 1 : La source sait que la destination est locale (même réseau IP)
  - Elle contacte tous ses voisins
    - Envoi d'un message en mode diffusion
      - **Broadcast** en anglais
    - Toutes les machines locales le reçoive
    - La destination répond
  - Un fois la destination localisée, envoie à cette seule machine
    - Envoi de messages en mode **unicast**
  - Ex : les machines d'une salle de l'IUT sont dans le même réseau local

Algorithme de communication d'une machine connectée à Internet (PC, mobile, tablette, XBox)

- Cas 2 : La source sait que la destination est distante = un autre réseau IP
  - Elle envoie le message à la **passerelle**
  - Le message contient l'adresse de la destination

  - Ex : les machines d'une salle de l'IUT, par exemple la salle 105, qui veulent communiquer avec un serveur Web sur Internet envoient à la passerelle 10.4.105.254

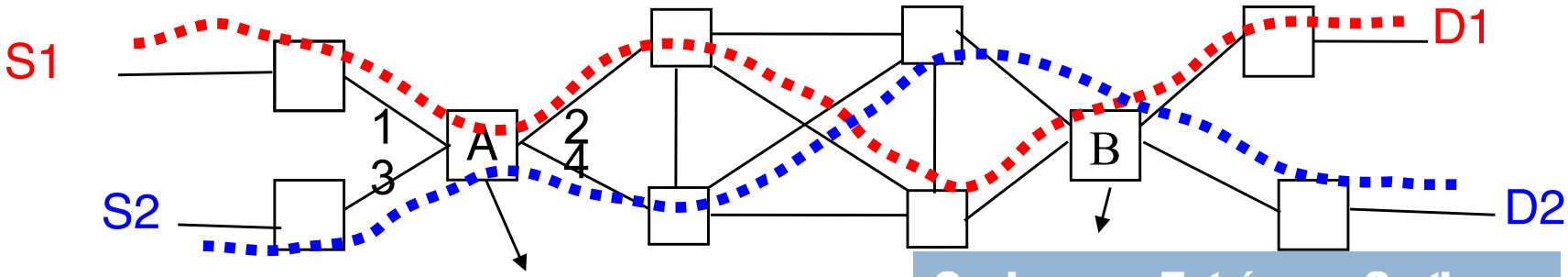
Notions importantes

- Local / distant : la machine qui veut envoyer sait ce qui est local (en première approximation « même début d'adresse IP »)
- Passerelle : le routeur à qui une machine envoie ses paquets si la destination est distante
- Unicast/Broadcast : une ou toutes les machines du réseau local vont recevoir le message

Coeur du Réseau

Commutation de circuits

- Principe
  - Avant de transmettre les données, un circuit est établi de la source à la destination
  - Chaque équipement intermédiaire doit connaître où renvoyer des informations reçues pour chaque circuit
- Exemple : Le système de téléphonie fixe (RTC: Réseau Téléphonique commuté)

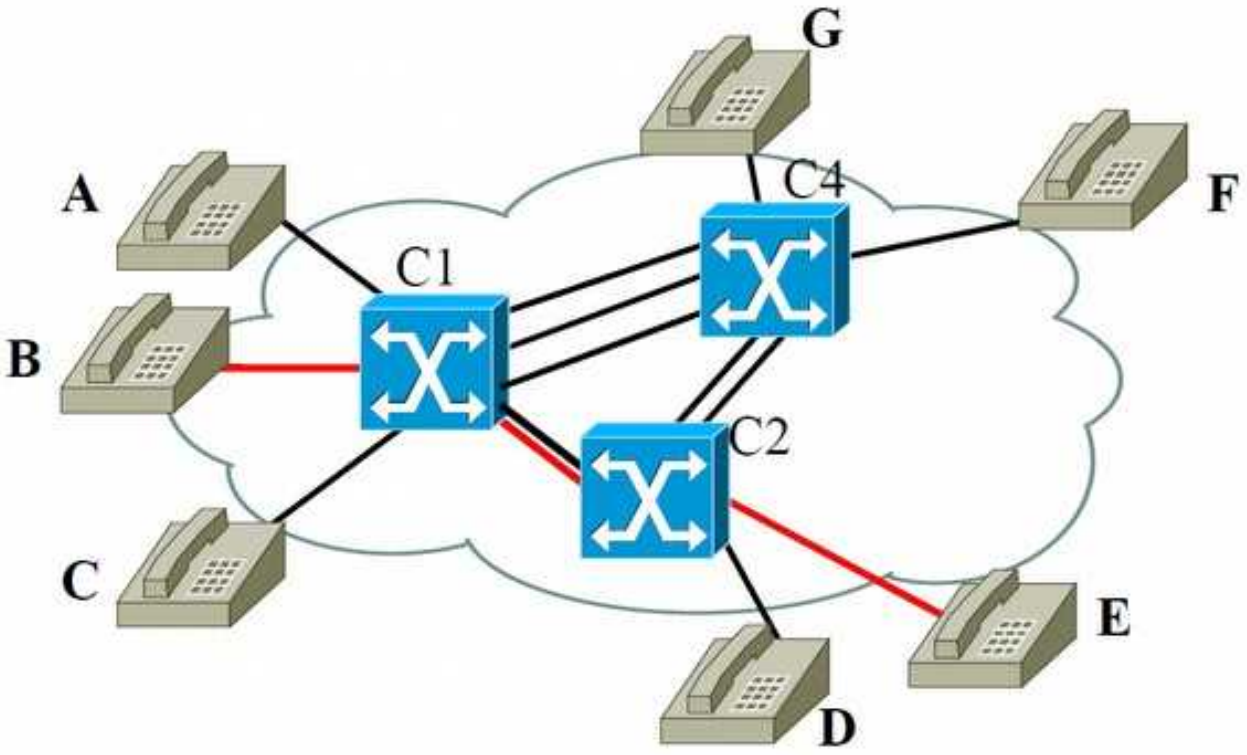


Couleur	Entrée	Sortie
Rouge	1	2
Bleu	3	4

Couleur	Entrée	Sortie
Rouge	3	2
Bleu	1	4

Le RTC (principe)

- Si A communique avec D et B avec D
  - Chaque ciruit réserve 64 kbits/s
  - Il faut 128 kbits/s au minimum sur C1-C2



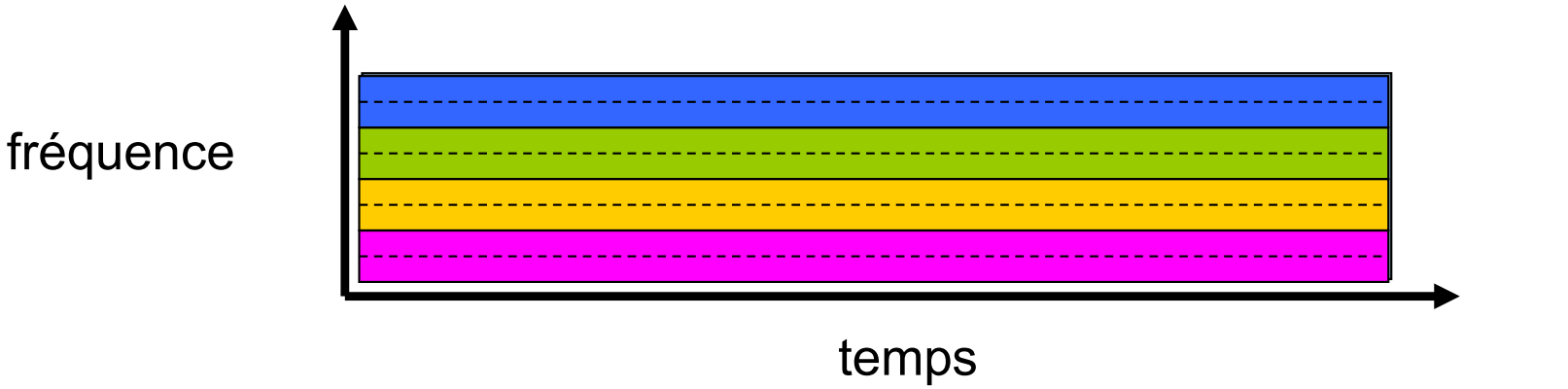
Commutation de circuits: partage des resources

- Resources réseaux (=tuyau) découpées en “parties”
- Parties allouées aux appels
- Une fois appel établi les ressources sont réservées pour toute la durée de l'appel!
  - … même si aucune données ne passent
- Comment diviser le tuyau?
  - Division en fréquences
  - Division temporelle

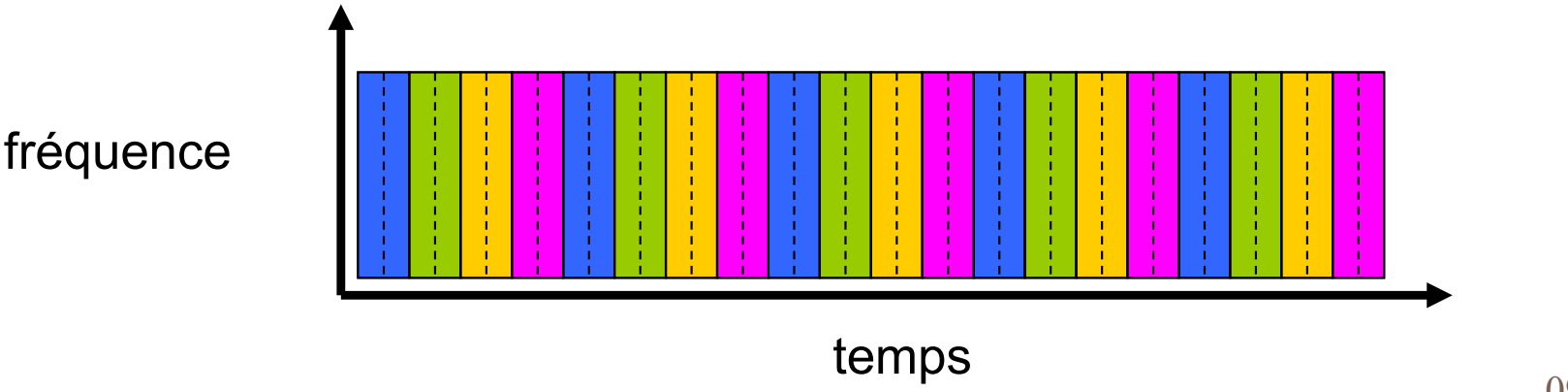
Division en temps ou en fréquences

Accès multiple par division en fréquences

Example:
4 utilisateurs

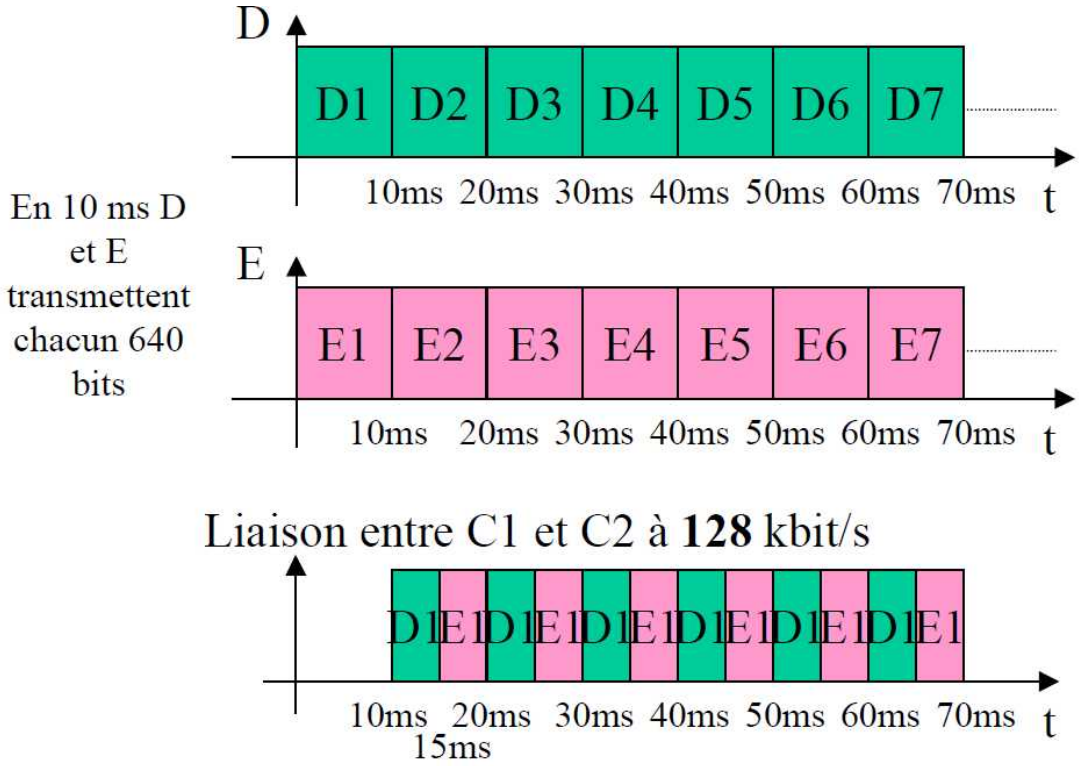


Accès multiple par division temporelle



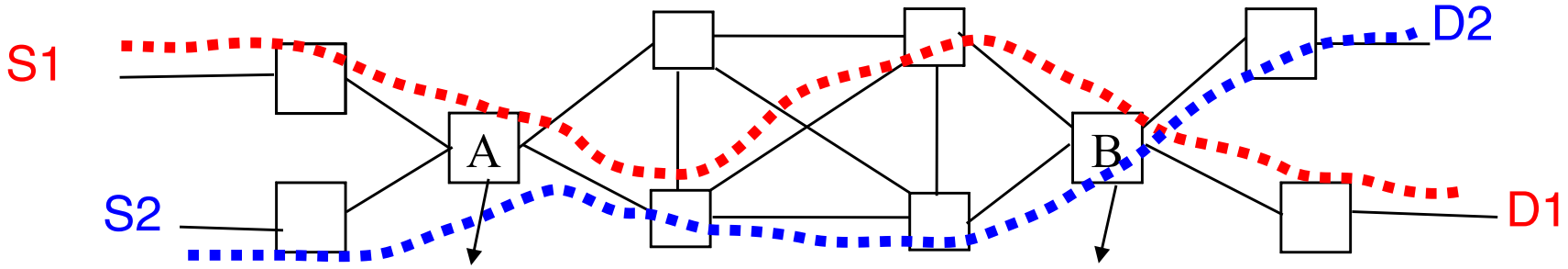
Le RTC : partage temporel

- 64 kbits/s : 640 bits en 10ms
- Si le lien C1-C2 est à 128 kb/s, on peut multiplexer (=partager) temporellement les 2 communications



Commutation de paquets

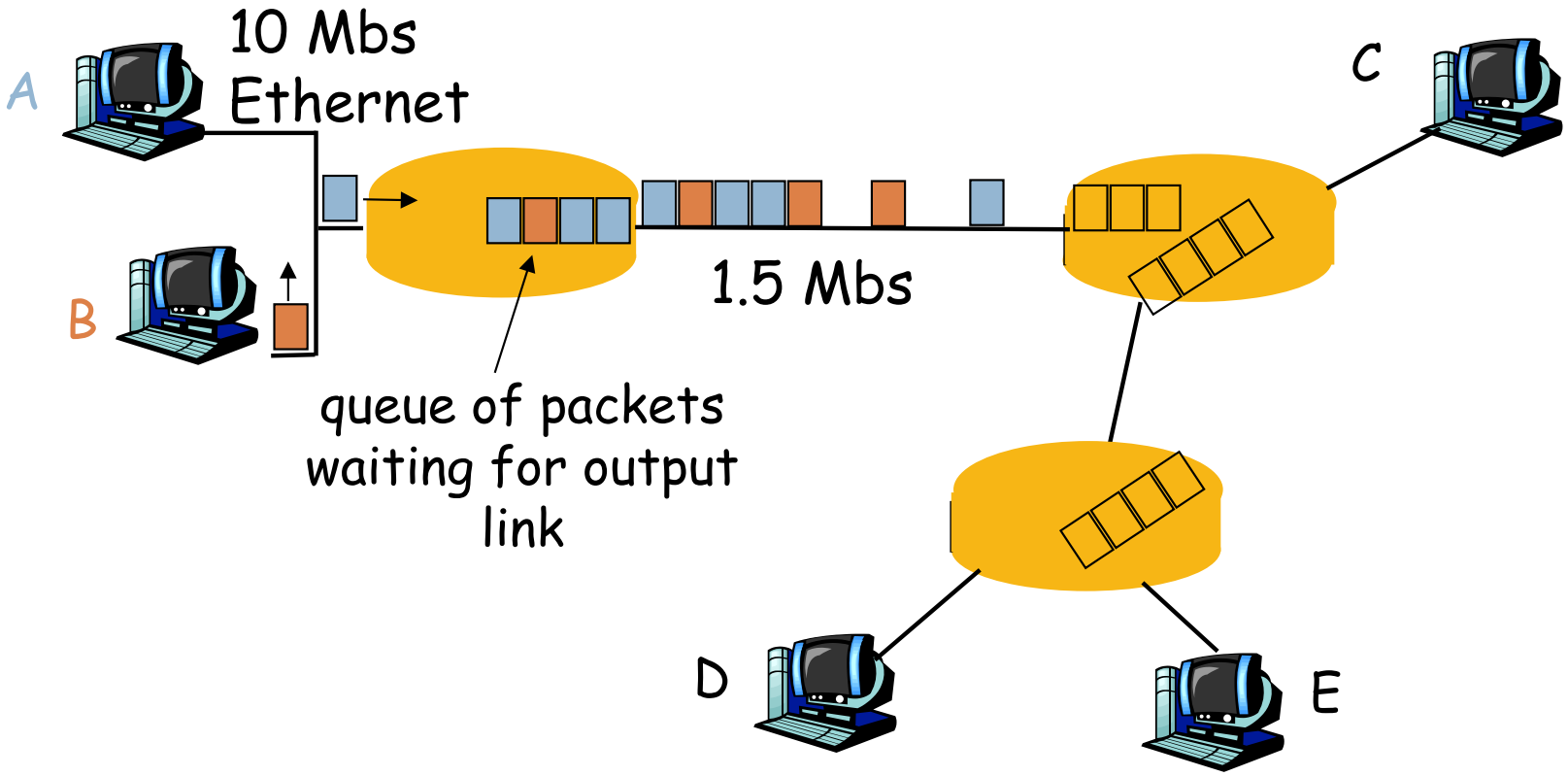
- Principe
  - Une adresse est associée à chaque destination
  - Les données sont divisées en petits paquets
    - Chaque paquet contient (en plus des données)
      - L´adresse de la source
      - L´adresse de la destination
  - Chaque nœud intermédiaire doit savoir le chemin vers chaque destination
  - Pas de réservation de ressources, mais si trop de paquets sur un lien → ralentissement ou pertes…
  - Exemple : le système postal, Internet



Destination	Lien
D1	4
D2	4

Destination	Lien
D1	4
D2	2

Commutation de paquets : Multiplexage des liens



Séquence de paquets de A et B entrelacés sur le médium meilleure utilisation que la commutation de circuits MAIS pas de ressources réservées → perte de paquets possibles!!

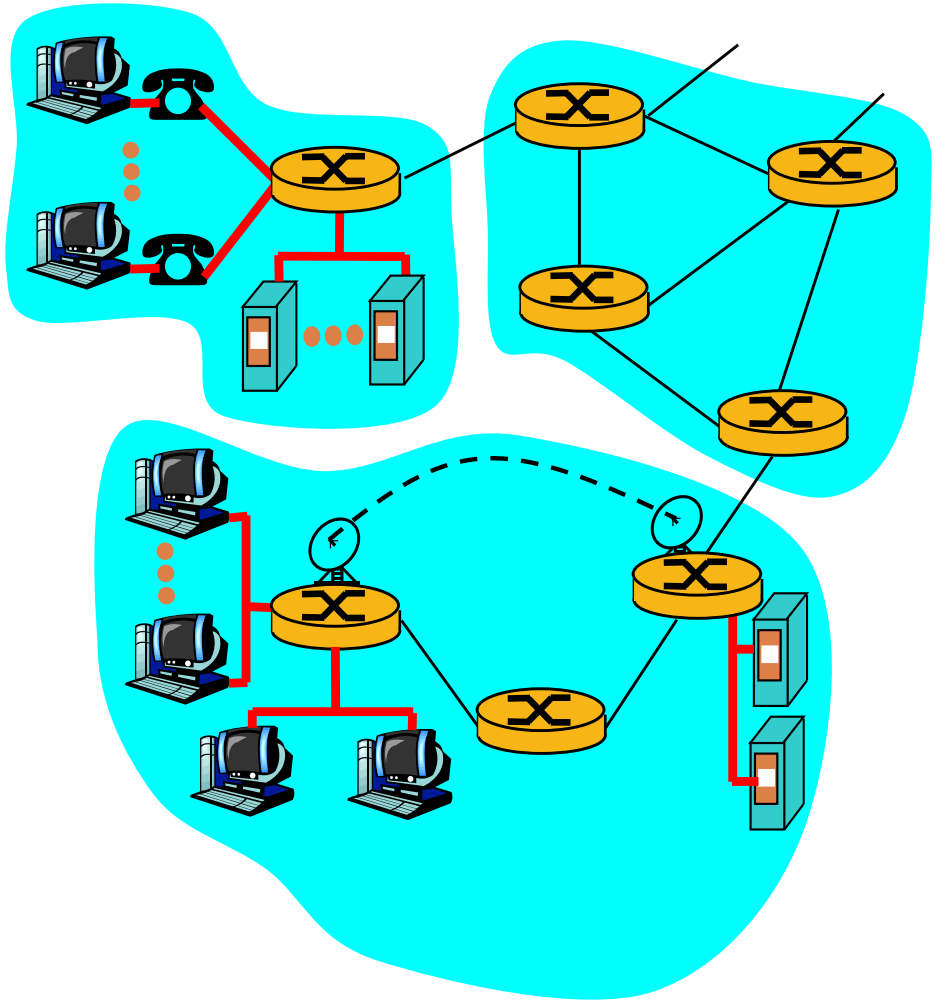
Service orienté connexion et réseau à commutation de circuits

- A bien comprendre :
  - Internet n'est pas orienté connexion alors que le réseau téléphonique commuté (RTC) l'est
  - Au dessus de l'Internet, TCP fournit un service orienté connexion
- Traduction :
  - Le réseau téléphonique gère les connexions
  - TCP permet aux applications (clients et serveurs) de se connecter, mais le réseau ne gère pas ses connexions

Réseaux d’accès

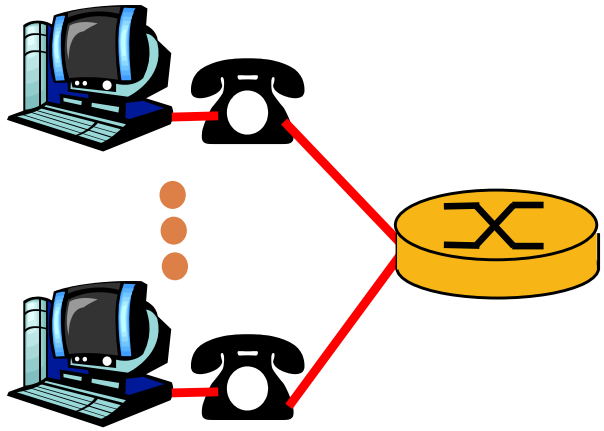
Réseaux d'accès

- Question : Comment raccorder les machines au réseau?
- Type :
  - Accès résidentiel pour les particuliers
  - Réseaux d'accès pour les institutions (école, entreprise, administrations, etc.)
  - Réseau d'accès mobile
- Des paramètres importants :
  - Débits (bits/s) du réseau d'accès?
  - Réseau partagé ou dédié?



Accès résidentiel point-à-point (dédié)

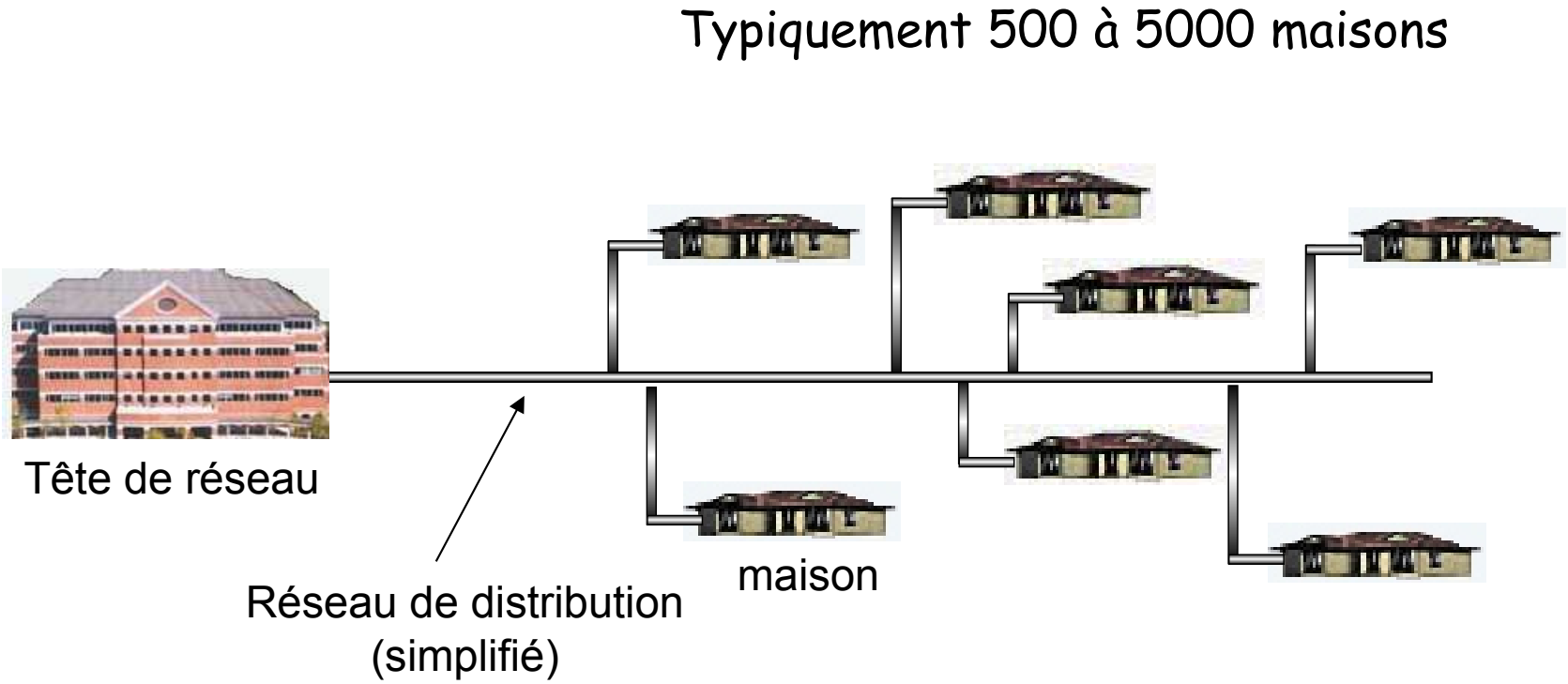
- Accès modem
  - → 56Kbps direct (souvent moins)
  - Populaire pour accéder Internet → 2000
  - On ne pouvait pas « surfer » et téléphone
- ADSL: asymmetric digital subscriber line
  - Jusqu`à 1 Mbits/s dans le sens montant (vers Internet),
  - Jusqu`à 20 Mbits/s dans le sens descendant
    - Valeur exacte dépend de la longueur de la ligne entre la maison et le local technique de l'opérateur
  - On peut surfer et téléphoner grâce au partage en bande passante:
    - 50 kHz - 1 MHz pour sens descendant
    - 4 kHz - 50 kHz pour sens montant
    - 0 kHz - 4 kHz pour téléphone classique (c'est la bande utilisée par les modems)



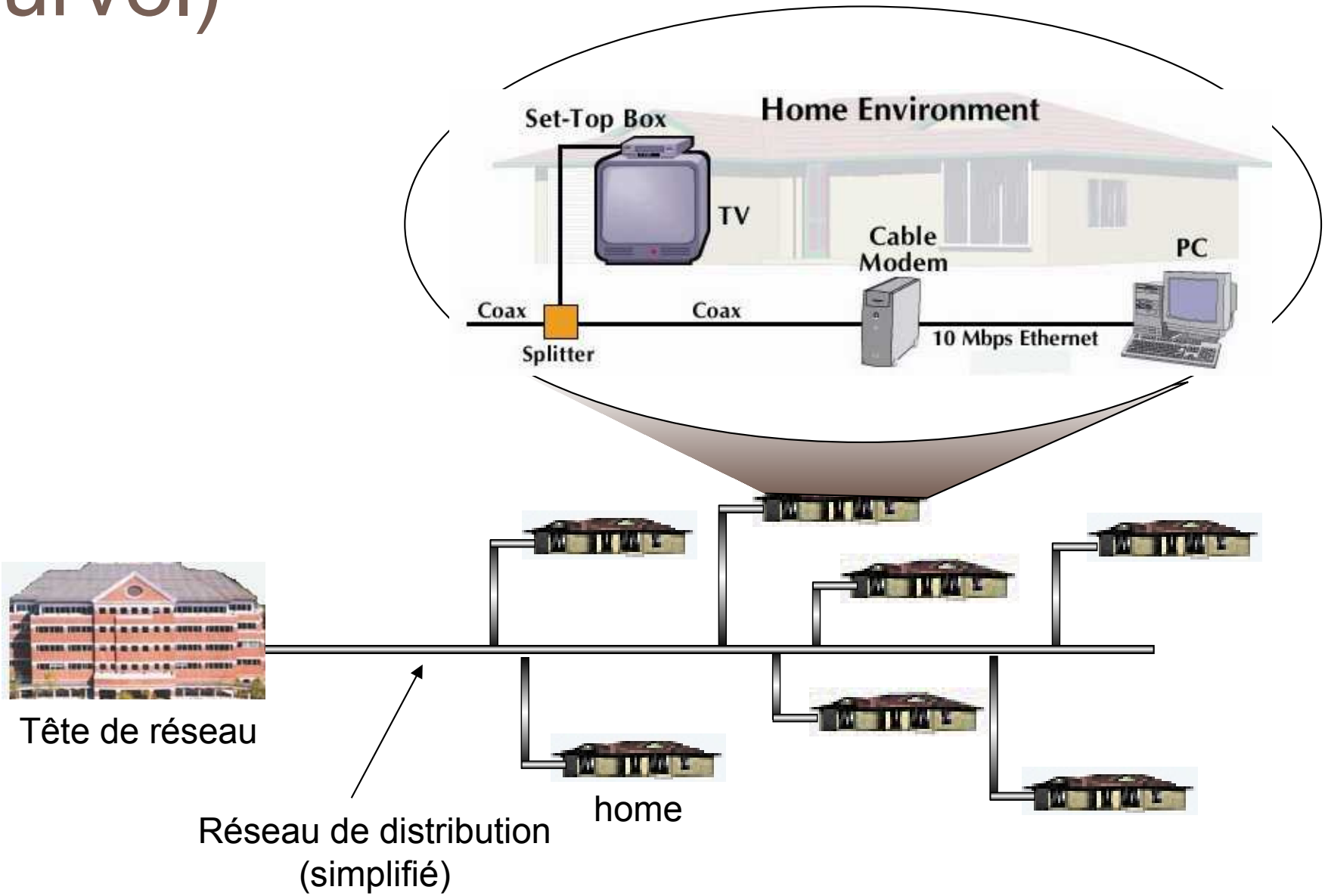
Accès résidentiel par réseau câble

- Technologie HFC: Hybrid Hiber Coax
  - Asymétrique: 10Mbps descendant, 1 Mbps montant
- Réseau de câble et de fibre optique attache les maisons au réseau de l'opérateur
  - Accès partagé entre toutes les maisons
  - Problème de congestion (=perte de paquets) et dimensionnement (quel débit mettre pour 1000 utilisateurs sachant que tous ne sont pas actifs simultanément…)
- Déploiement : entreprises spécialisées, par exemple, Numericable

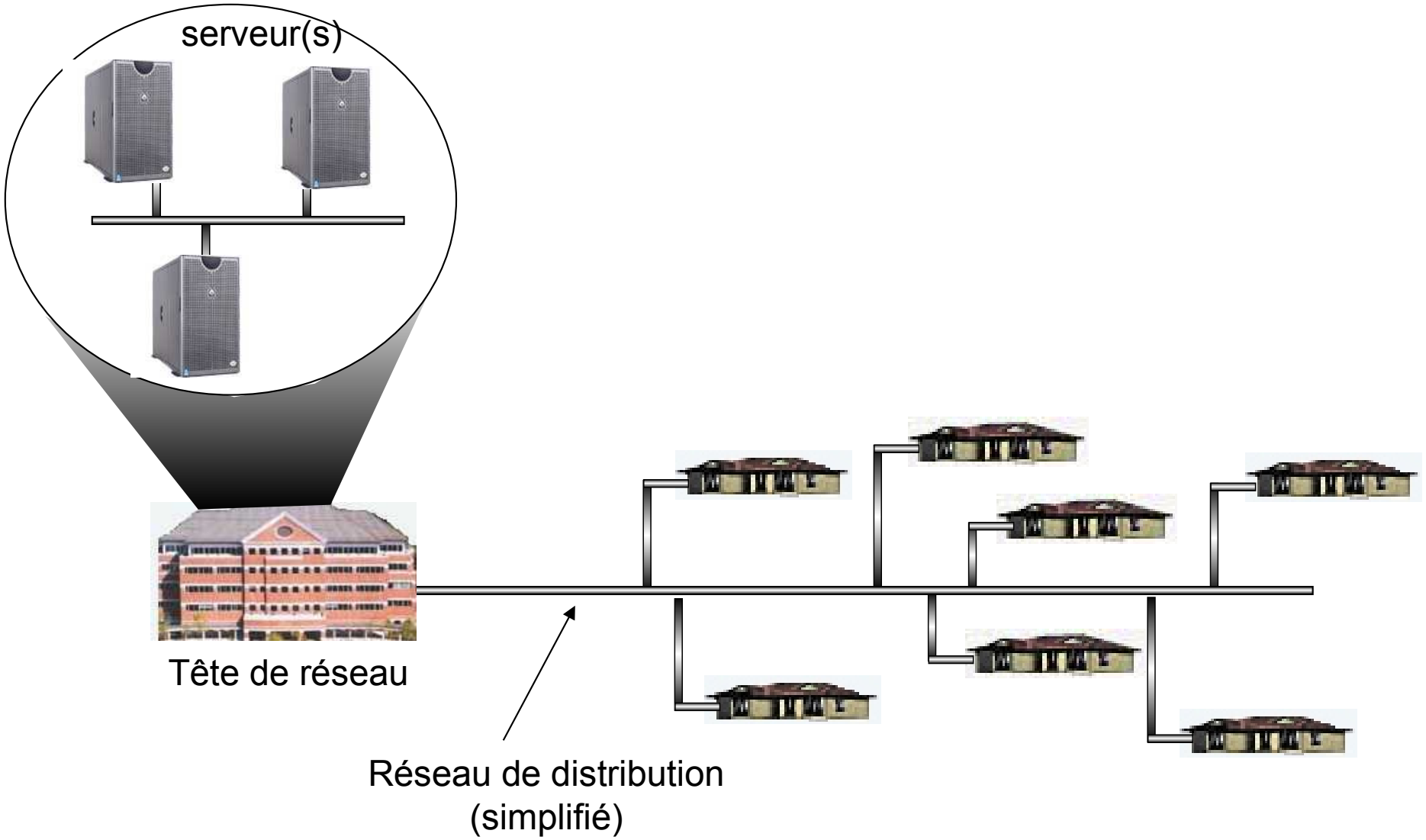
Architecture d'un réseau câble (survol)



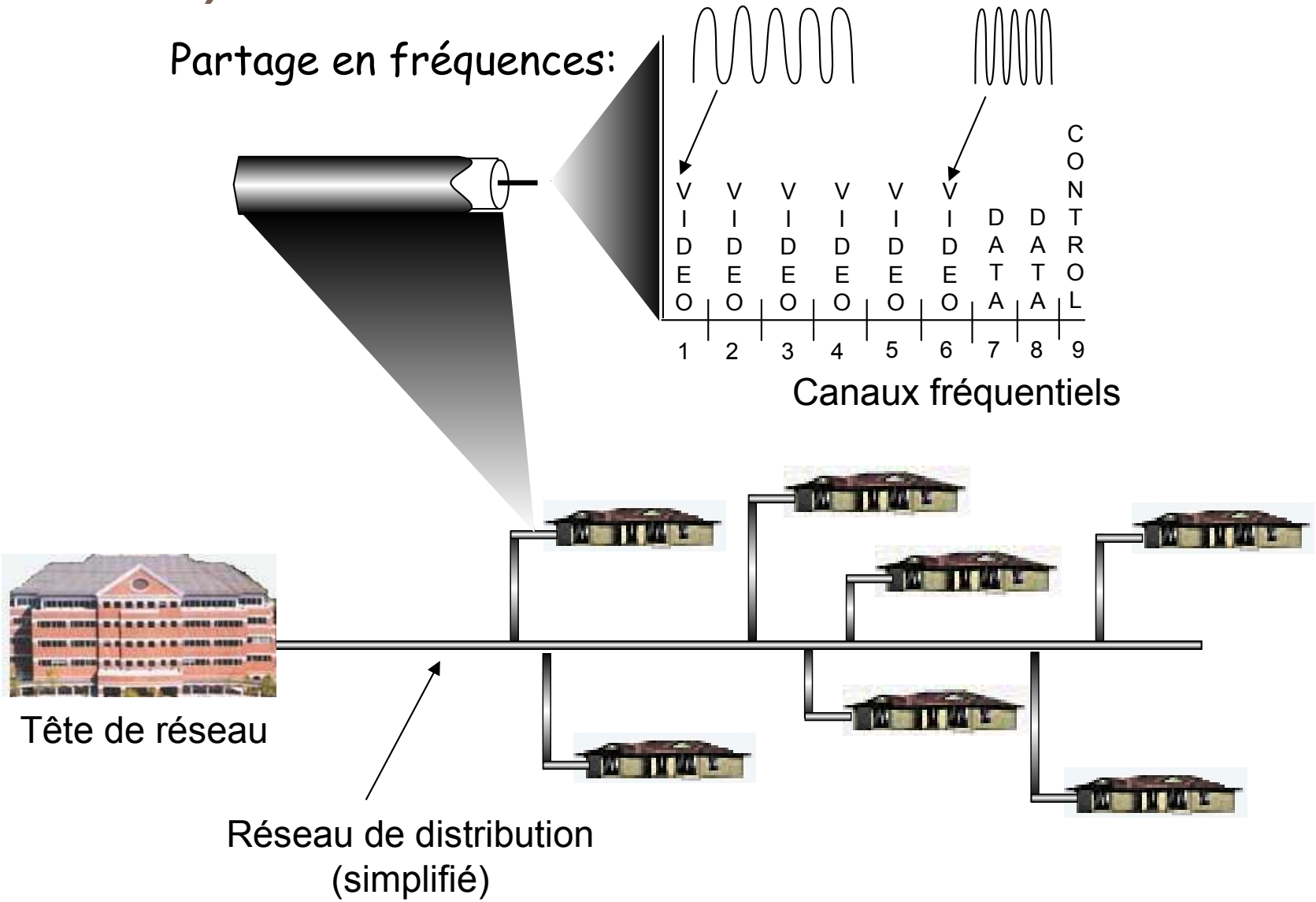
Architecture d'un réseau câble (survol)



Architecture d'un réseau câble (survol)

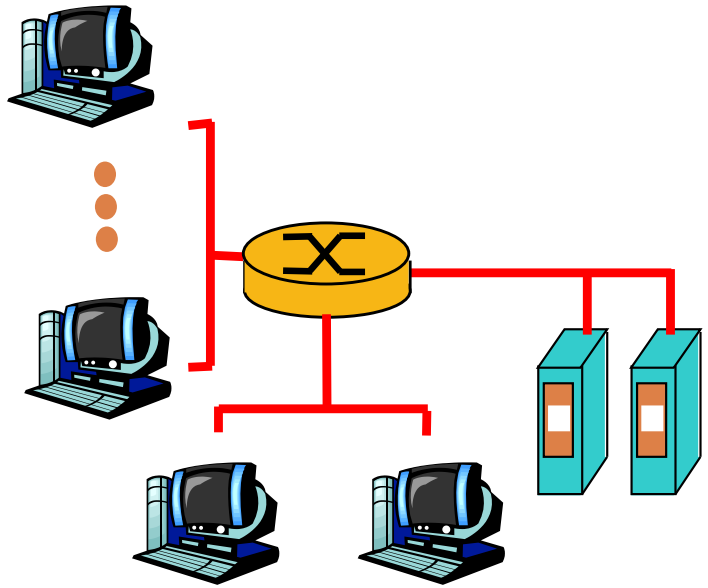


Architecture d'un réseau câble (survol)



Accès entreprise : réseau d'accès local (LAN)

- Technologie Ethernet:
  - Dédié ou partagé par machine
  - 10 Mbps, 100Mbps, 1 Gbps



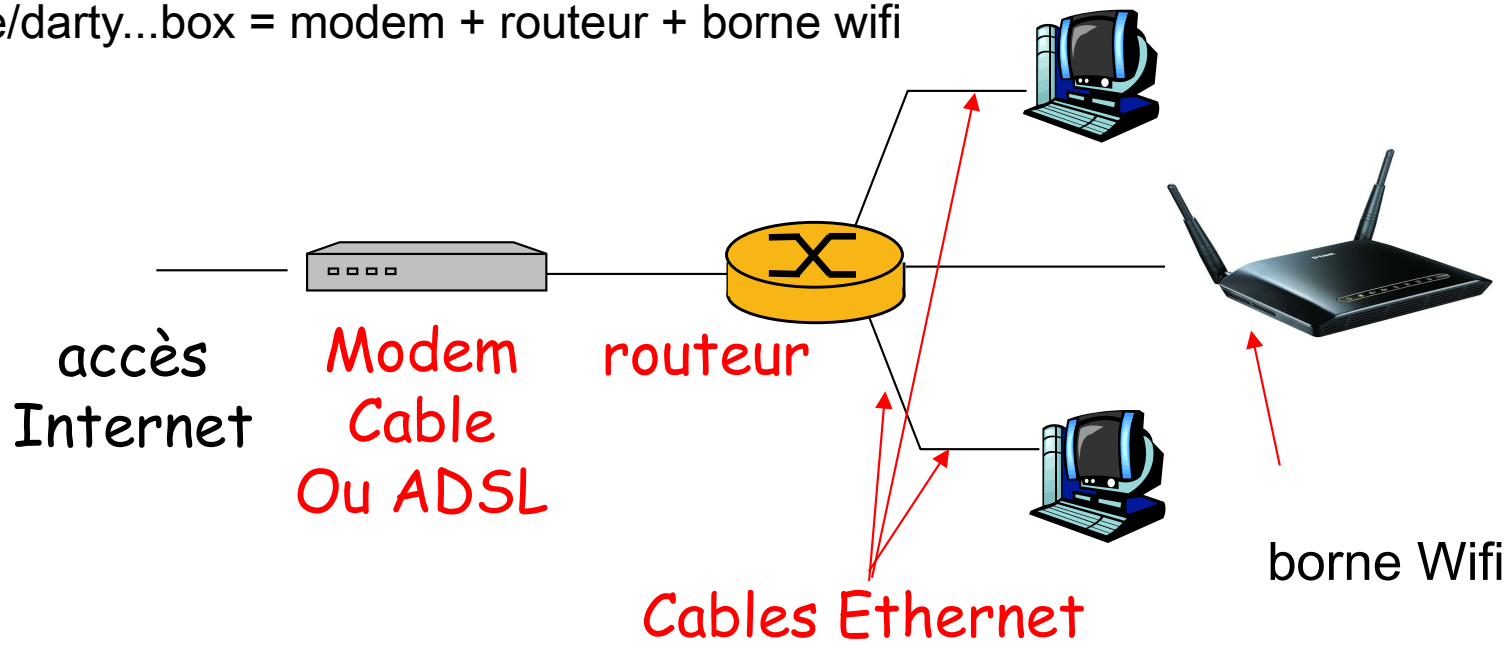
Accès sans fil

- Machines se connectent au point d'accès (Wifi) ou stations de base (2G/3G)
- Wifi :
  - Médium partagé
  - 802.11b/g (WiFi): 11 ◊ 54 Mbps
  - Courtes distances
- Réseaux nationaux
  - Fournis par opérateurs de télécommunication
  - Médium « dédié »

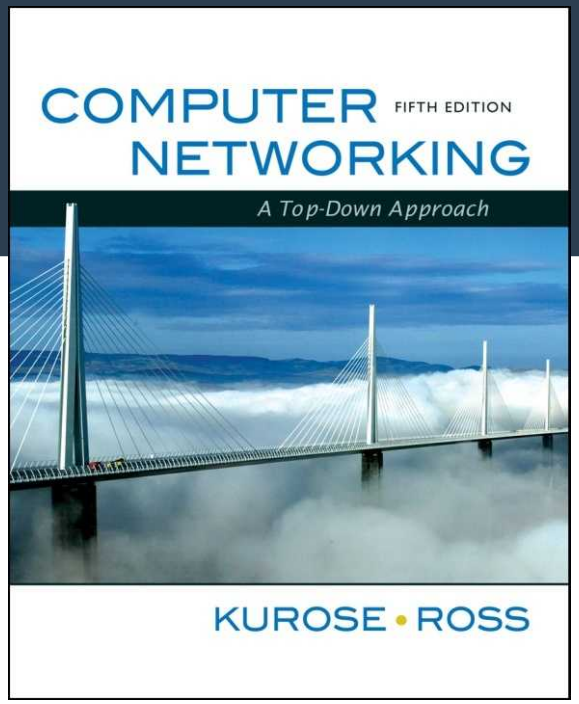
	Peak network capacity	A
GPRS (2G)	115 kbps	
EDGE (2G+)	473 kbps	
UMTS (3G)	2 Mbps	
HSDPA (3G+)	14 Mbps	

Réseau à la maison

- Composants typiques:
- Modem câble ou ADSL
- routeur/pare-feu (sécurité) /NAT
  - NAT : Network Address Translation – permet de partager une seule adresse IP entre plusieurs machines
- Ethernet
- Point d'accès wifi
- Live/free/darty...box = modem + routeur + borne wifi



Couche applicative



Computer Networking: A Top Down Approach 5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.

A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in powerpoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2009
J.F Kurose and K.W. Ross, All Rights Reserved

Agenda

- **Principes de la couche applicative**
  - Clients et serveurs
  - Besoins des applications
- **Web et HTTP**
- **DNS**

PRINCIPES D´UNE APPLICATION RÉSEAU

Applications réseaux: jargon

Processus: programme s'exécutant sur une machine.

- Des processus s'exécutant sur des machines différentes communiquent à l'aide d'un protocole de niveau applicatif

Agent utilisateur (User agent): interface entre l'utilisateur et le réseau.

- Implémente simultanément l'interface utilisateur et le protocole de niveau applicatif
  - Web: browser (butineur)
  - E-mail (Mél)
  - Diffusion audio/vidéo: Real audio player, Windows media player

Applications et protocoles de niveau applicatif

Application: des processus communiquant distribués

ex:.e-mail, Web, partage de fichier pair-à-pair, messagerie instantanée

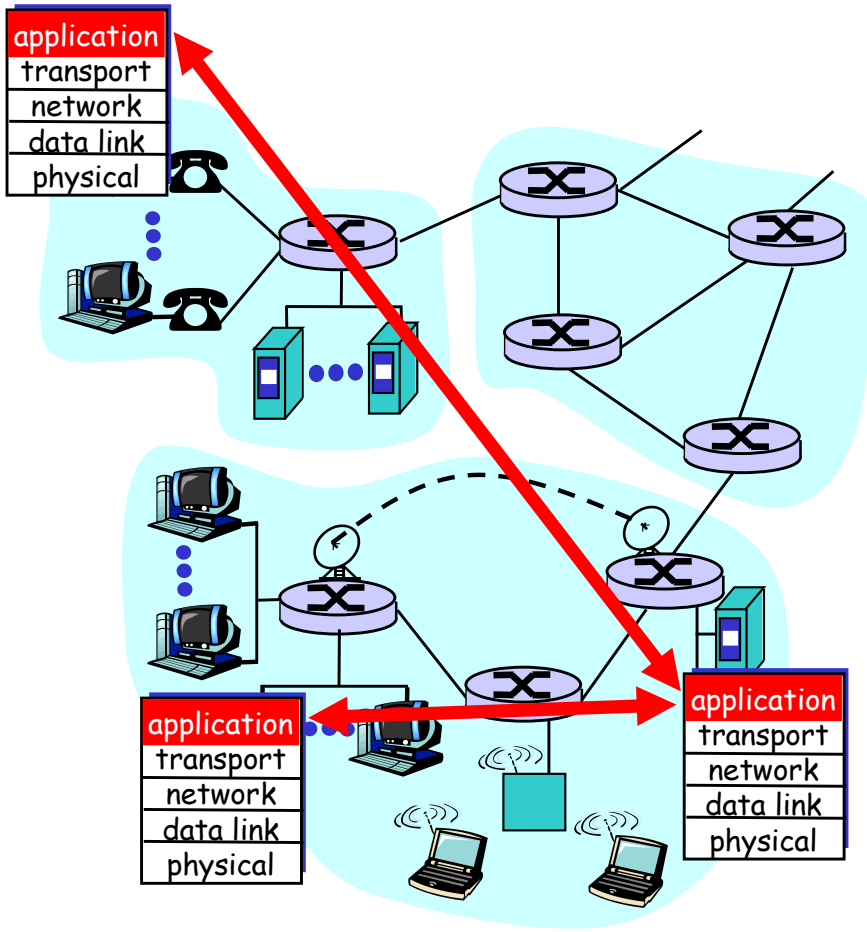
Protocoles de niveau applicatif

Une partie de l'application

Autres parties : encodage/décodage de flux, interface utilisateur

Définit les messages échangés par les applications et les actions à effectuer

Utilisent les services de communications offerts par la couche inférieure: TCP ou UDP



Les protocoles de niveau applicatif

- Types de messages échangés, par ex: requête et réponse
- Syntaxe des différents types de message : quels champs dans les messages et comment ils sont séparés
- Sémantique des champs, c'est-à-dire le sens des informations dans ces champs
- Règles pour quand et comment les processus doivent envoyer et répondre aux messages

Protocoles du domaine public :

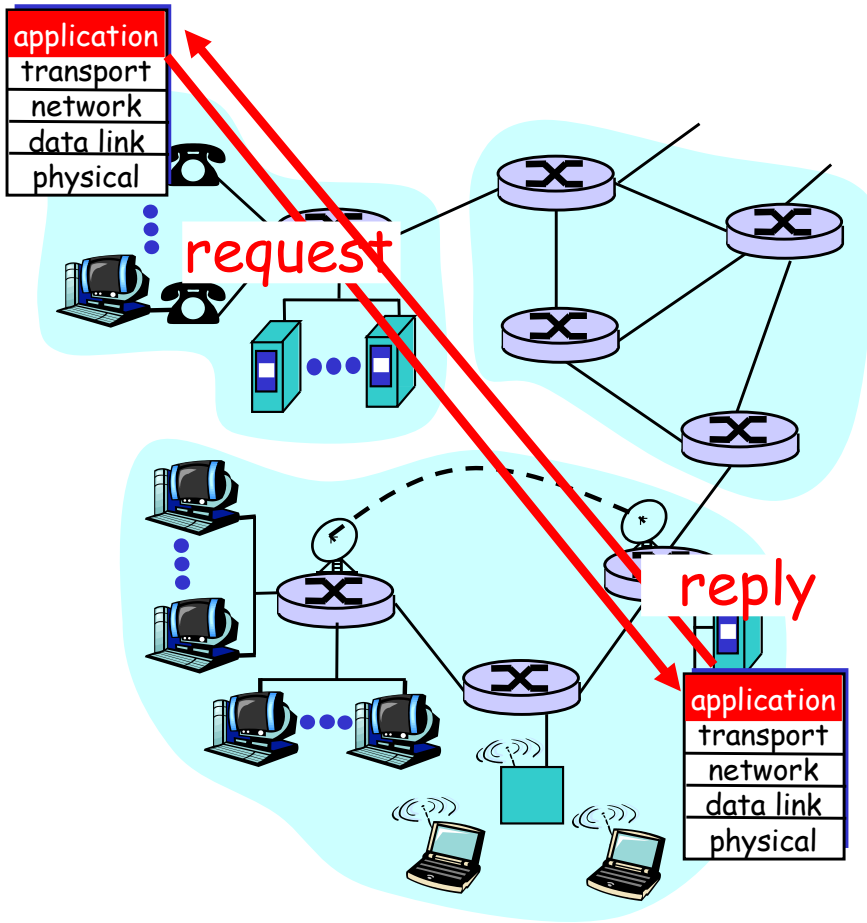
- Définis dans les RFCs (Requests For Comments) = normes de l'Internet
- Permettent l'interopérabilité des différents agents utilisateurs et des processus côté serveur
- Ex : HTTP (Web), SMTP (mail)

Protocoles propriétaires:

- Ex : Skype

Paradigme client/serveur

Application réseau typique à 2 composants : le client et le serveur



Client:

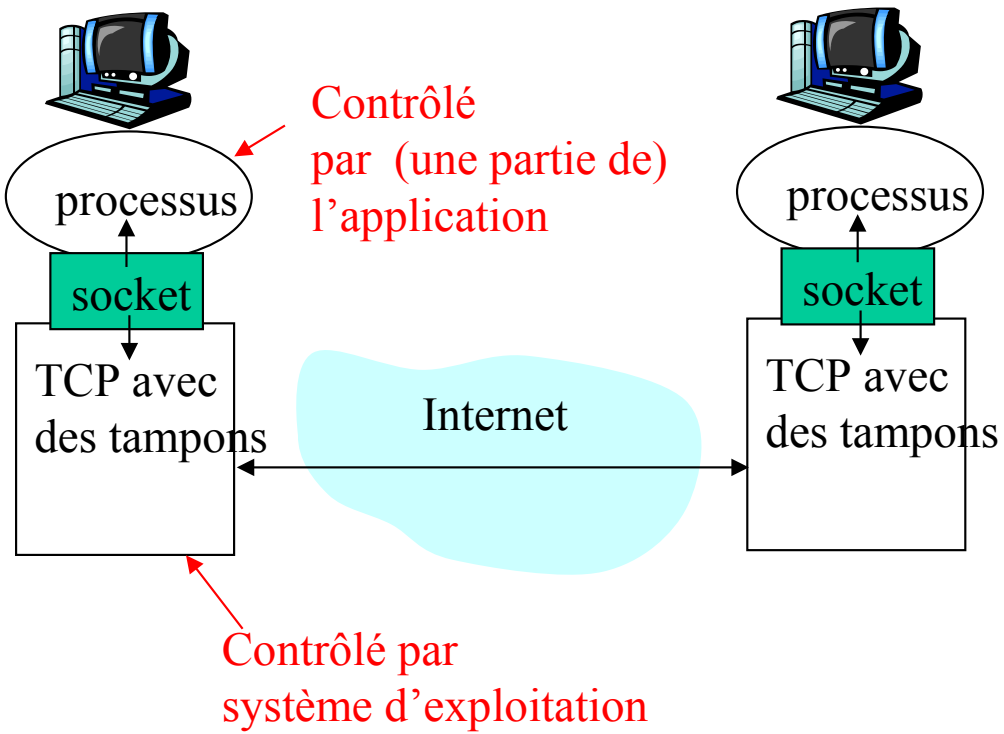
- Initie le contact avec le serveur (parle en premier)
- Typiquement, requêtes servies par serveur
- Web: client implanté dans le browser; e-mail: dans le lecteur de mail

Serveur:

- Fournit le service requis au client
- Ex: serveurs Web envoie les pages Web au client, le serveur de mail délivre les mails

Les processus communiquent au travers du réseau

- **Les processus s'envoient/reçoivent des messages au travers de sockets**
- **Sockets analogues à des portes dans les systèmes d'exploitation**
  - Émetteur pousse le message au travers de la porte
  - Émetteur fait l'hypothèse que la couche transport achemine le message à une autre porte sur une autre machine, où le récepteur attend



Adressage des processus

- **Pour qu'un processus reçoivent les messages, il doit avoir un identifiant**
- **Chaque machine a un identifiant unique : une adresse IP sur 32 bits**
- **Q: est-ce que cette @ IP suffit pour identifier le processus?**
- **Réponse: Non, car plusieurs processus différents tournent sur la machine (Web, mail,etc.)**

- **Identifiant processus : couple (adresse IP, numéro de port)**
- **Exemples:**
  - Serveur HTTP: 80
  - Serveur Mail : 25
- **« Une maison, plusieurs portes »**

De quel service offert par la couche transport une application a besoin?

Perte de données

- **Certaines applications (ex: audio) peuvent tolérer des pertes**
- **D'autres (ex : transfert de fichier, mise à jour Windows) demandent une fiabilité de 100%**

Délai

- **Certaines applications (ex : voix sur IP, jeux) requièrent des délais très faibles**

Débit

- Certaines applications (ex: multimedia) demandent un débit minimum pour fonctionner
- D'autres applications dites "élastiques" se contentent du débit qu'on leur offre

Services de transport requis par les principaux types d'applications

Application	Perte	Débit	Sensibilité au délai
Transfert de fichier	Pas de perte	élastique	non
mail	Pas de perte	élastique	non
Documents Web	Pas de perte	élastique	non
audio/video temps réel (TR)	tolérant	audio: 5kbps-1Mbps video:10kbps-5Mbps	oui, qques centaines de msec
audio/video stocké	tolérant	idem	oui, qques secs
Jeux interactifs	tolérant	Quelques kbit/s	Idem vidéo TR
Messagerie instantanée	Pas de perte	élastique	Oui et non

Protocoles de transport du monde IP

Service TCP :

- ***Orienté-connexion :*** **établissement de connexion obligatoire entre processus client et serveur**
- ***Transport fiable :*** **entre émetteur et récepteur**
- ***Contrôle de flux :*** **émetteur ne doit pas saturer récepteur**
- ***Contrôle de congestion :*** **émetteur module son débit en fonction de la congestion du réseau**
- ***Ne fournit pas:*** **de garanties en délai ou débit**

Service UDP:

- **Transport non fiable de données entre 2 processus**
- **Ne fournit pas : établissement de connexion, de fiabilité (résistance aux pertes), de contrôle de congestion ou de flux, de délai ou de débit**

  → RIEN, sauf la possibilité de communiquer entre deux application !!

Quelle couche de transport pour quelle application?

Application	Protocole de couche Applicative	Protocole de transport
e-mail	SMTP [RFC 2821]	TCP
Accès terminal distant	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Transfert de fichiers	FTP [RFC 959]	TCP
streaming multimedia	propriétaire (e.g. YouTube)	TCP or UDP
Téléphonie Internet	proprietary (e.g., Skype)	typiquement UDP

WEB ET HTTP

Web et HTTP

Un peu de jargon

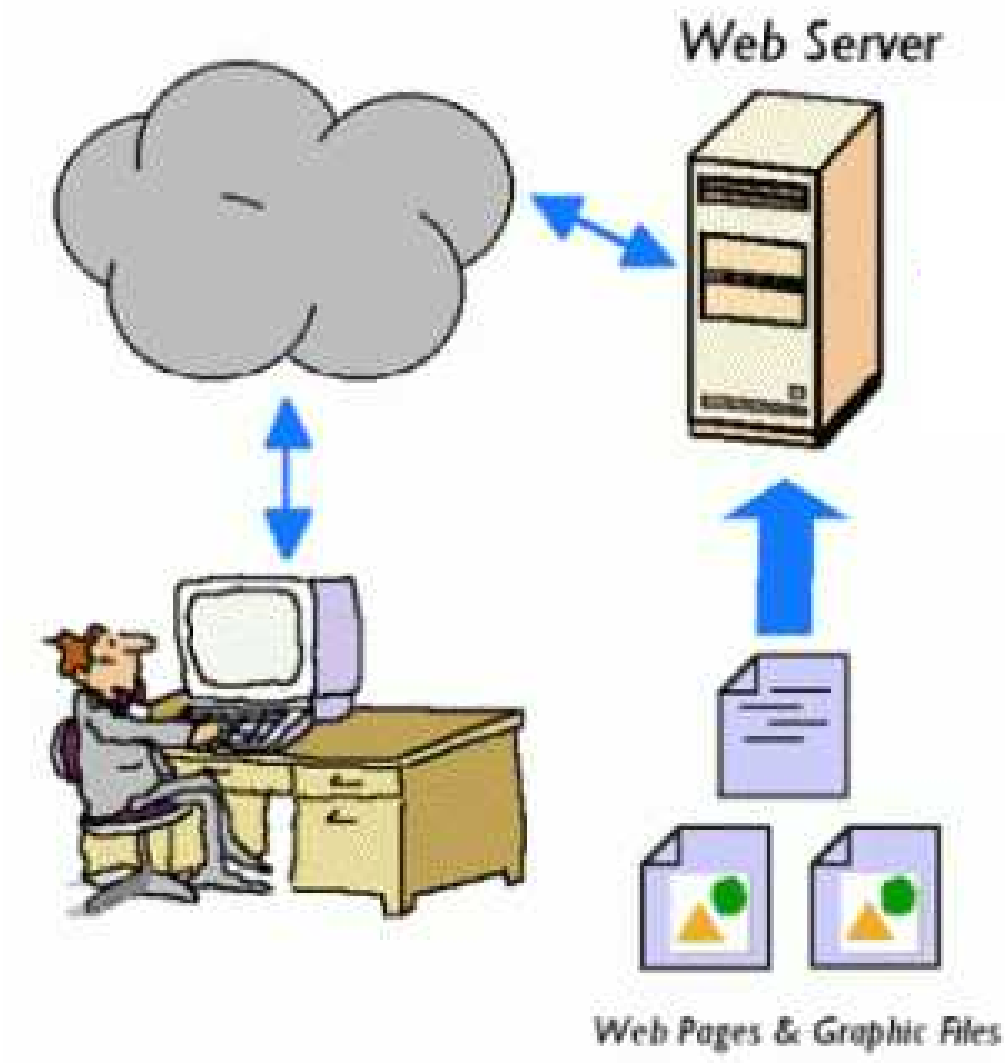
- **Une page Web contient des objets**
- **Un objet peut-être un fichier HTML, une image JPEG, une applet Java, un fichier audio,…**
- **Une page Web consiste en un fichier de base HTML qui contient les références à plusieurs objets**
- **Chaque objet est adressable à partir d'une URL**
- **Example URL:**

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

Nom machine

Chemin sur la machine

Un serveur Web sert des objets (source : theydiffer.com)



HTML (wikipedia)

- L'*Hypertext Markup Language*, généralement abrégé HTML, est le format de données conçu pour représenter les pages web.
- Langage de balisage qui permet d'écrire de l'hypertexte.
- HTML permet de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des *applets*.

HTML (w3school)

```
<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>

<p>My first paragraph.</p>

</body>
</html>
```

```
<!DOCTYPE html>
<html>
<body>

<a href="https://www.w3schools.com">This is a link</a>

</body>
</html>
```

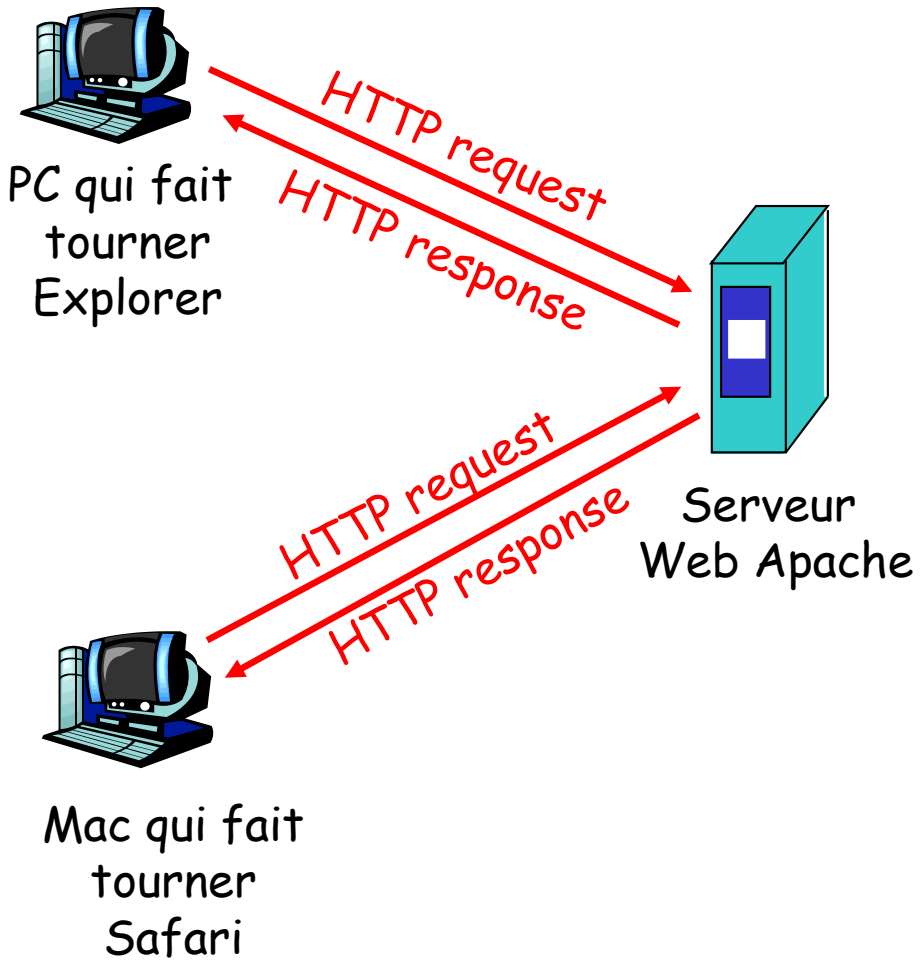
JPEG, Applet Java

- ***JPEG***
  - ***Joint Photographic Experts Group*** (**JPEG**)
  - Norme de compression des images numériques
  - D'autre norme existe: GIF, etc.
  -
- **Applet Java: programme écrit dans un langage de programmation (Java) qui est :**
  - transféré du serveur au client
  - exécuté au niveau du client

HTTP : un survol rapide

HTTP: hypertext transfer protocol

- **Protocole applicatif du Web**
- **Basé sur modèle client/serveur**
  - *client:* browser qui émet les requête et affiche les objets Web
  - *serveur:* envoie les objets en réponse des requêtes
- **HTTP 1.0: RFC 1945 (1996)**
- **HTTP 1.1: RFC 2068 (1997)**
- **HTTP 2.0 : RFC 7540 (2015)**



HTTP : un survol rapide (suite)

Utilise TCP:

- client initie une connexion TCP (crée socket) avec le serveur, sur le port 80
- Le serveur accepte la connexion TCP
- Des messages HTTP (messages de la couche applicative) sont échangés entre le browser (client HTTP) et le serveur Web (serveur HTTP)
- La connexion TCP est fermée

HTTP ne garde pas d'états ("stateless" en anglais)

- Le serveur ne maintient pas d'historique des requêtes passées du client
- D'où les cookies !!!

En marge

Les protocoles qui maintiennent des états sont complexes

- Il faut stocker les données historiques
- Si le serveur/client tombe en panne, leur vues de l'état peuvent être différentes → réconcilation nécessaire

Connexions HTTP

HTTP non persistant

- Au plus un objet est envoyé par connexion TCP
- HTTP version 1.0 (HTTP/1.0) est non persistant

HTTP Persistant

- Plusieurs objets peuvent être envoyés sur une même connexion TCP entre client et serveur.
- HTTP/1.1 utilise le mode persistant par défaut

HTTP non-persistant

Ex: utilisateur entre URL www.someSchool.edu/someDepartment/home.index

(contient **texte, references à 10 Images jpeg)**

1a. Client HTTP initie une connexion TCP au serveur HTTP (processus) à l'adresse www.someSchool.edu sur le port 80

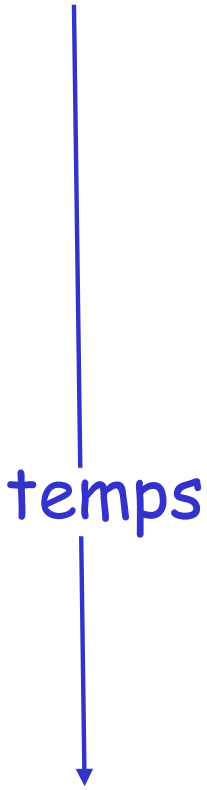
1b. Serveur HTTP sur la machine www.someSchool.edu attend les connexions TCPsur le port 80. Il "accepte" la connexion et notifie le client

2. Client HTTP emet une requête HTTP (contenant une URL) au travers de la socketTCP éatblie entre le client et le serveur. Le message indique que le client veut l'objet someDepartment/home.index

3. Serveur HTTP reçoit le message de requête, prépare le message de réponse contenant l'objet en envoie ce message au travers de la socket

temps

HTTP non persistant (suite)



4. Serveur HTTP ferme la connexion TCP.

5. Client HTTP reçoit le message de réponse contenant le fichier html et l'affiche. Il analyse le fichier html et trouve 10 objets jpeg référencés

6. Les étapes 1 à 5 sont répétées pour chacun des 10 objets jpeg

Messages HTTP

- **Deux types de messages HTTP : *requête, réponse***
- **Requête:**

ASCII (format lisible par un humain)

Ligne de requête (commandes : GET, POST, HEAD)

ligne en-tête

```
GET /répertoire/page.html HTTP/1.1
Host: www.someschool.edu
User-agent: Mozilla/4.0
Connection: close
Accept-language:fr
```

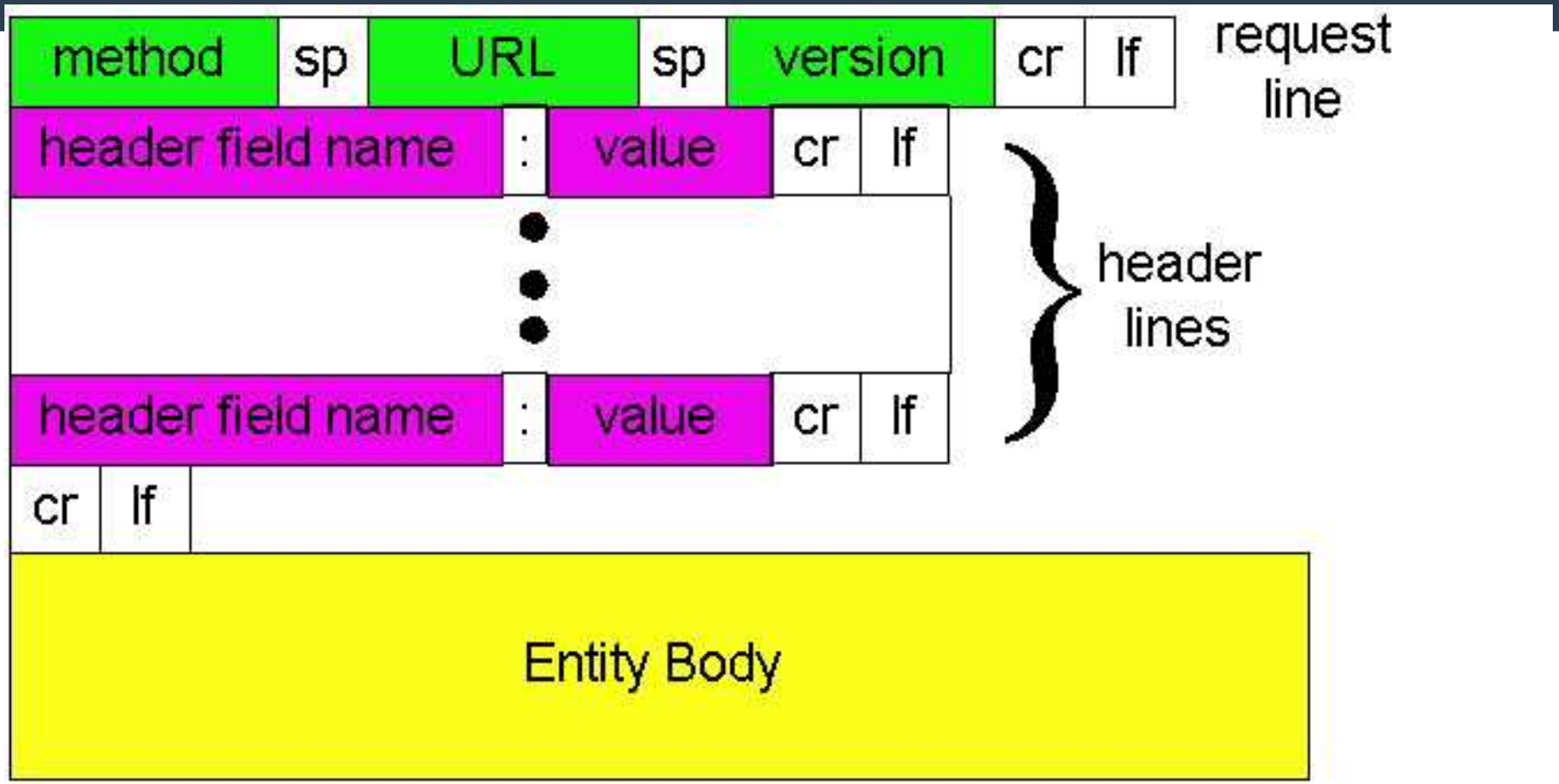
Retour chariot
Passage à la ligne
Indique fin message

http://www.useragentstring.com

http://unice.fr/robots.txt

https://www.amazon.fr/robots.txt

Messages de requête : format général



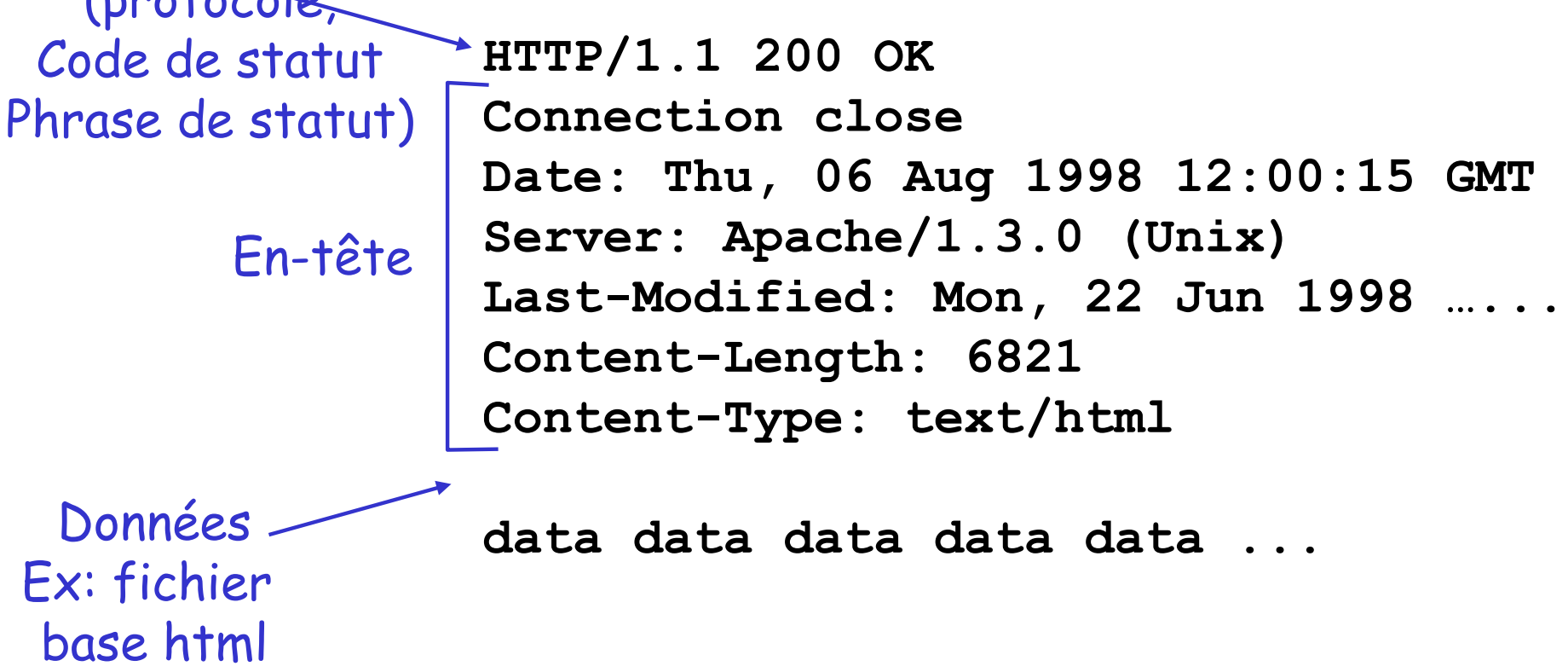
cr= carriage return (retour chariot)
lf : line feed (passage à la ligne)
sp = space = espace
Body = corps du message
Header = en-tête

Réponses HTTP

Ligne de statut (protocole, Code de statut Phrase de statut)

En-tête

Données Ex: fichier base html



Code statut réponse (ex 100, 200, 404..)

- **Catégories :**
  - 1xx Informations
  - 2xx succès
  - 3xx redirection,
  - 4xx erreur client
  - 5xx erreur serveur

https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml

Codes statut des réponses HTTP

Les plus courants :

200 OK

- Succès de la requête client : les objets demandés sont dans la suite de ce message

301 Moved Permanently

- L'objet demandé a changé d'emplacement. Nouvel emplacement indiqué ci-après (mot clef Location:)

400 Bad Request

- Message de requête non compris par le serveur

404 Not Found

- Document n'a pas été trouvé par le serveur

505 HTTP Version Not Supported

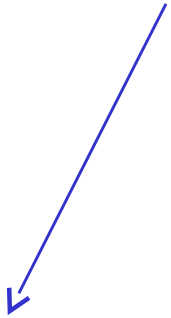
Envoyer des données à un serveur Web

Message POST:

- **Les pages incluent souvent des informations à remplir par l'utilisateur**
- **Ex: identifiant nom/mot de passe**
- **Les données sont envoyées dans le corps**

Méthode par ajout dans URL:

- **Utilise le message GET**
- **Données sont ajoutées dans l'URL:**



http://www.le-dictionnaire.com/definition.php?mot=statut

Méthodes (Requêtes) – liste non exhaustive

HTTP/1.0

- **GET**
- **POST**
- **HEAD**
  - Demande au serveur de ne pas renvoyer l'objet demandé
  - Utilisé pour des tests

HTTP/1.1

- **GET, POST, HEAD**
- **PUT**
  - Télécharge un fichier, contenu dans le corps à l'emplacement spécifié par le champ URL
- **DELETE**
  - Détruit le fichier correspondant à l 'URL
  - Opération soumise à des autorisation sur le serveur

HTTP 2.0 (HTTP/2)

- **Etude W3Techs, Mai 2017 : 13.7 % des 10 millions de serveurs les plus populaires supportent HTTP/2**

- **Supporté par de nombreux navigateurs : Chrome, Opera, Firefox,Internet Explorer 11 , Edge, Safari, Amazon Silk(Kindle Fire)**

HTTP/2

- **RFC en Mai 2015**
- **A l'origine, SPDY de Google qui vise à diminuer le temps de chargement d'une page**
- **Techniques utilisées :**
  - Compression des en-têtes
  - Plusieurs requêtes en même temps sur une même connexion (HTTP 1.1 : une requête puis une réponse puis une requête….)
  - Push du serveur pour anticiper demande client
- **Méthodes, statuts, URI de HTTP/1.1 inchangées**
- **Fonctionne uniquement au dessus de TLS (SSL) actuellement – pas dans la norme, mais en pratique, c'est le cas**

HTTP/2 et « sharding »

- **Si HTTP/1.1 est lent et ne permet pas le pipelining, alors on peut distribuer le contenu d'un site sur plein de sous-sites qu'on interroge en parallèle !**
- **Ex : quand on charge la page web du journal Le Monde, on reçoit du contenu de**
  - S1.lmde.fr
  - Asset.lmde.fr
  - Live.lmde.fr
- **C'est la technique utilisée à l'heure actuelle**
- **Dit autrement, acceptater HTTP/2 pour gagner en performance demande de ré-architecturer les sites Web**
- **L'avenir nous dira si cela fonctionne**

DNS

DNS: Domain Name System

Des identifiants multiples pour les personnes :

- No sécu, passeport

Routeur et machines Internet:

- Adresses (32 bit) – utilisé pour adresser des datagrammes
- “nom”, e.g., www.unice.fr- utilisé par les humains

Q: Comment se fait la correspondance adresses IP/noms??

DNS: Domain Name System

- **Base de données distribuée implantée par une hiérarchie de serveurs de noms**

- **Protocole de niveau applicatif**
  - Protocole client/serveur
  - Suite de requêtes/réponses

- **Utilise UDP comme couche transport**
  - Port : 53
  - UDP est suffisant car requêtes/réponses tiennent dans un datagramme
  - Avec TCP, il faudrait établir la cx □ perte de temps!
  - Si requête (ou réponse perdue) le client réémet au bout d'un certain temps

- **Sur windows/unix, commande nslookup**

Adresses IP?

- **Structure rigide : 4 octets**

- **Représentation usuelle :**
  - Chaque octet traduit en décimale (chiffre en 0 et 255) et séparé par un point
  - Ex: 121.7.106.32

Usage typique DNS

- A chaque fois que vous envoyez un mail à copain@truc.com le client mail fait une requête DNS pour connaître l'adresse IP du serveur de mail de truc.com

- A chaque fois que vous surfez sur un site Internet. Pour chaque URL, votre butineur fait une requête pour résoudre la partie machine, par exemple www.truc.com

Services DNS

- **Traduction nom adresse IP**
  - www.truc.com → 121.7.106.32
- **Alias de machines**
  - Noms canoniques, alias
  - Ex: une entreprise achète un gros serveur qu'elle nomme grosserveur.entreprise.com
    - Grosserveur.entreprise.com est le nom canonique
    - Ce serveur est le serveur de mail de l'entreprise alias mail.entreprise.com pour grosserveur.entreprise.com
    - Ce serveur est le serveur web aussi alias www.entreprise.com pour grosserveur.entreprise.com

Services DNS

- **Alias des serveurs de mail**
  - Vous utilisez copain@truc.com quand vous écrivez un mail
  - Truc.com est un nom de domaine
  - Le DNS permet de trouvez le serveur de mail pour un domaine
  - Par exemple truc.com aura pour serveur de mail mail.truc.com
  - On résoud ensuite l'adresse IP de mail.truc.com

Services DNS

- **Distribution de charge**
  - Ex: serveurs Web repliqués : ensemble d'adresse IP correspondant à un même nom canonique
  - www.google.com correspond en fait à plusieurs serveurs physiques, donc plusieurs adresses IP

```
> google.com
Server:  resolver1.elisa.fi
Address:  193.229.0.40

Non-authoritative answer:
Name:    google.com
Addresses:  74.125.39.106, 74.125.39.147, 74.125.39.99, 74.125.39.103
          74.125.39.104, 74.125.39.105
```

Services DNS

- **Distribution de charges (suite)**
  - Si vous faites plusieurs requêtes à la suite sur google.com, l'ordre des adresses change → permet de répartir la charge car votre client prend la première adresse

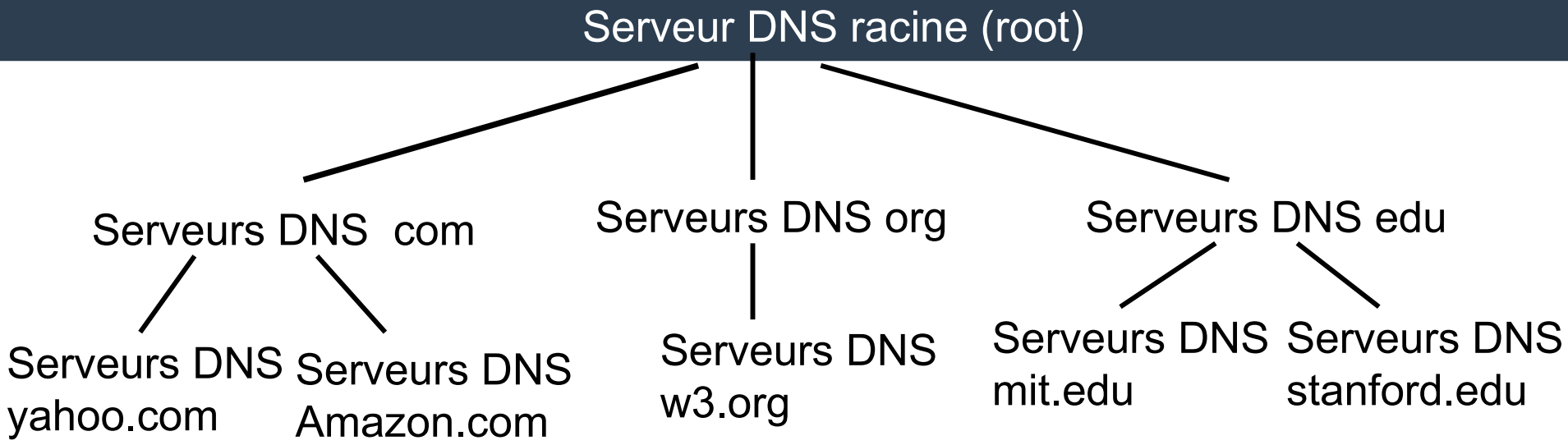
```
> google.com
Server:  resolver1.elisa.fi
Address:  193.229.0.40

Non-authoritative answer:
Name:    google.com
Addresses:  74.125.39.106, 74.125.39.147, 74.125.39.99, 74.125.39.103
          74.125.39.104, 74.125.39.105

> google.com
Server:  resolver1.elisa.fi
Address:  193.229.0.40

Non-authoritative answer:
Name:    google.com
Addresses:  74.125.39.99, 74.125.39.103, 74.125.39.104, 74.125.39.105
          74.125.39.106, 74.125.39.147
```

Pourquoi le DNS n'est pas centralisé?

- Si le serveur tombe en panne, catastrophe
- Le volume de trafic à traiter serait trop important
- Maintenance
- Performance : si un seul serveur, par ex. au US, le temps de réponse pour les clients en Europe serait long car il faut un Aller/Retour pour la requête/réponse

Ne passe pas à l'échelle*!*

Base de données hiérarchique, distribuée



Client DNS de votre machine veut obtenir l'IP de amazon.com (1ère approximation!)

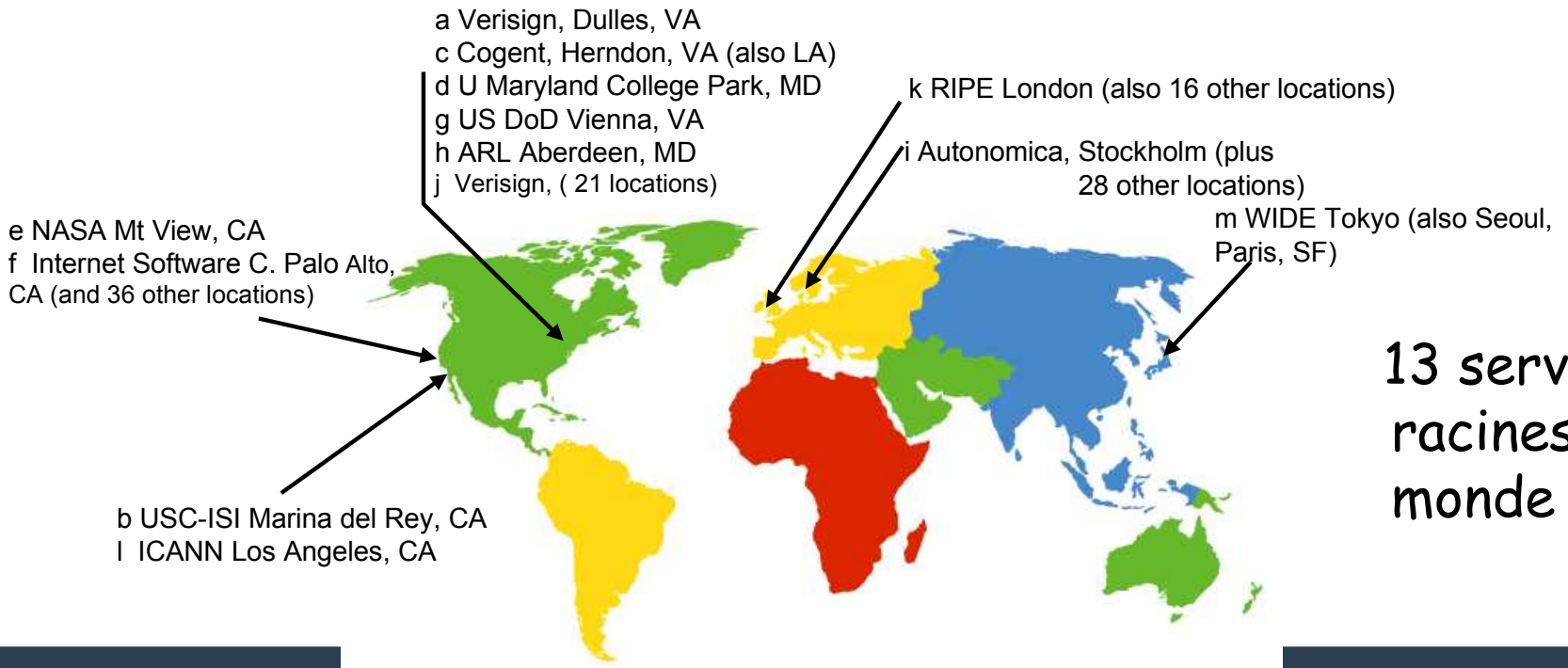
- **client interroge le serveur racine pour trouver un serveur pour com**
- **client interroge un serveur DNS pour com pour trouver un serveur pour amazon.com**
- **client interroge un serveur DNS amazon.com pour avoir l'IP de www.amazon.com**

3+1 classes de serveurs DNS

- **Serveurs Racines: 13 dans le monde**
  - Le saint des saints!
  - « Root servers » en anglais
- **Serveurs de domaine de premier niveau**
  - « TLD (top level domain) servers » en anglais
- **Serveurs ayant autorité sur un domaine**
  - « Authorized servers » en anglais
  -
- **Serveur local (appelé « resolver »)**
  - Géré par l'entité qui vous fournit l'accès Internet
  - Ne font pas stricto sensu partis de la hiérarchie mais servent de point d'entrée pour les clients DNS des machines
    - Adresse serveur local fournit à la configuration de la machine

DNS: Les serveurs DNS racines

- **Contactés par les serveurs DNS locaux lorsqu'ils ne savent pas traduire un nom**
- **Serveur DNS racine:**
  - Le serveur racine connait la traduction ou connait un serveur ayant autorité!



13 serveurs racines dans le monde

TLD and Authoritative Servers

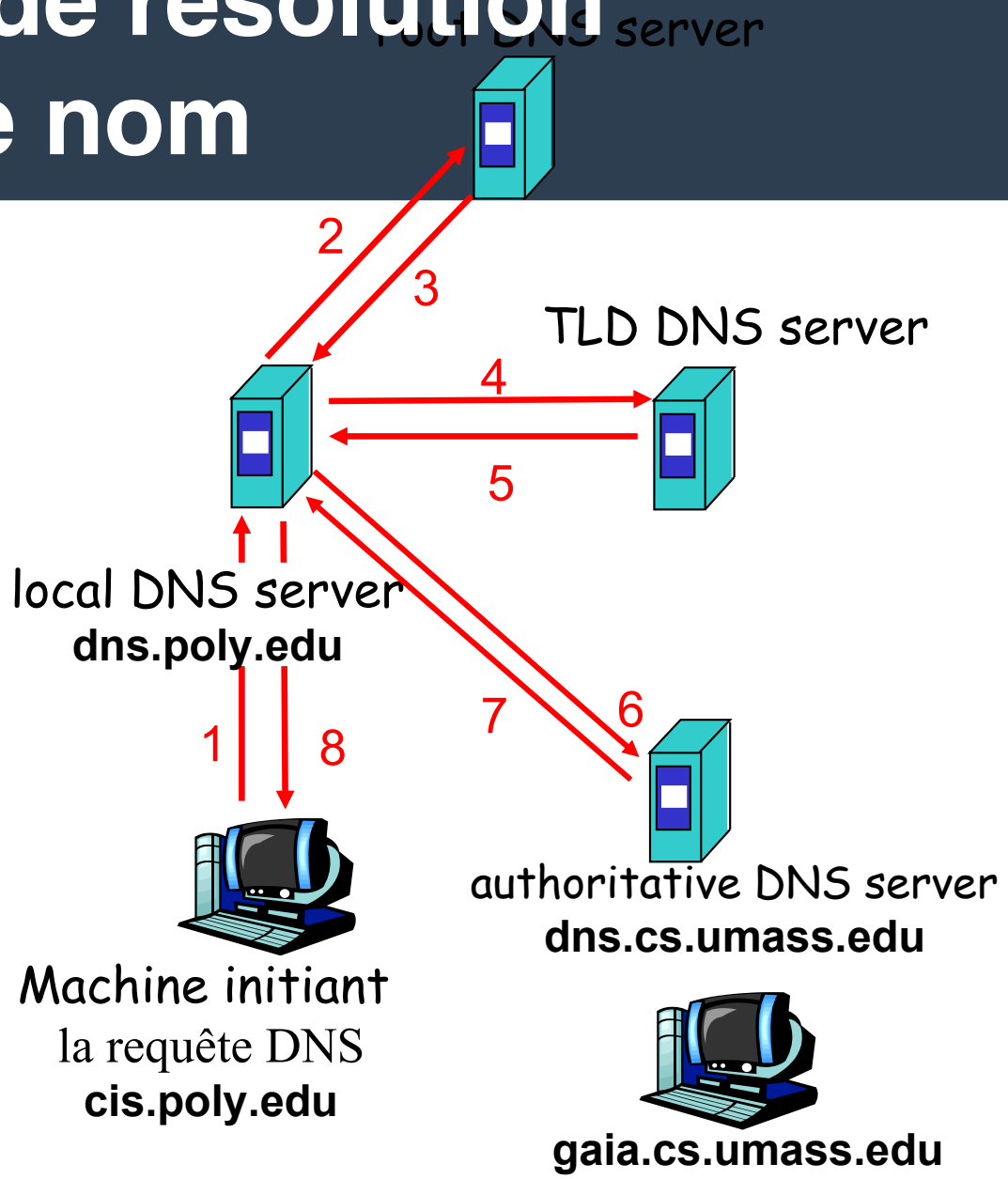
- **Top-level domain (TLD) servers:**
  - responsables de com, org, net, edu, int, etc, et de toutes les extensions de nom de pays
  - Ex:
    - l'entreprise Network Solutions maintient des serveurs pour le TLD com
    - Educause pour le TLD edu
    -
- **Authoritative DNS servers:**
  - Serveurs DNS d'organisations qui fournissent la traduction nom/adresse IP pour les serveurs de ces organisations (ex: Web, mail).
  - Les serveurs sont opérés par l'organisation elle-même ou un fournisseur de service (entreprise privée)

Exemple de résolution de nom

- **La machine cis.poly.edu veut l'adresse IP de gaia.cs.umass.edu**

Requête itérative

- Le serveur contacté répond avec un serveur plus bas dans la hiérarchie ("plus près") à contacter
- Cela veut dire qu'il fournit l'adresse IP du nouveau serveur à contacter



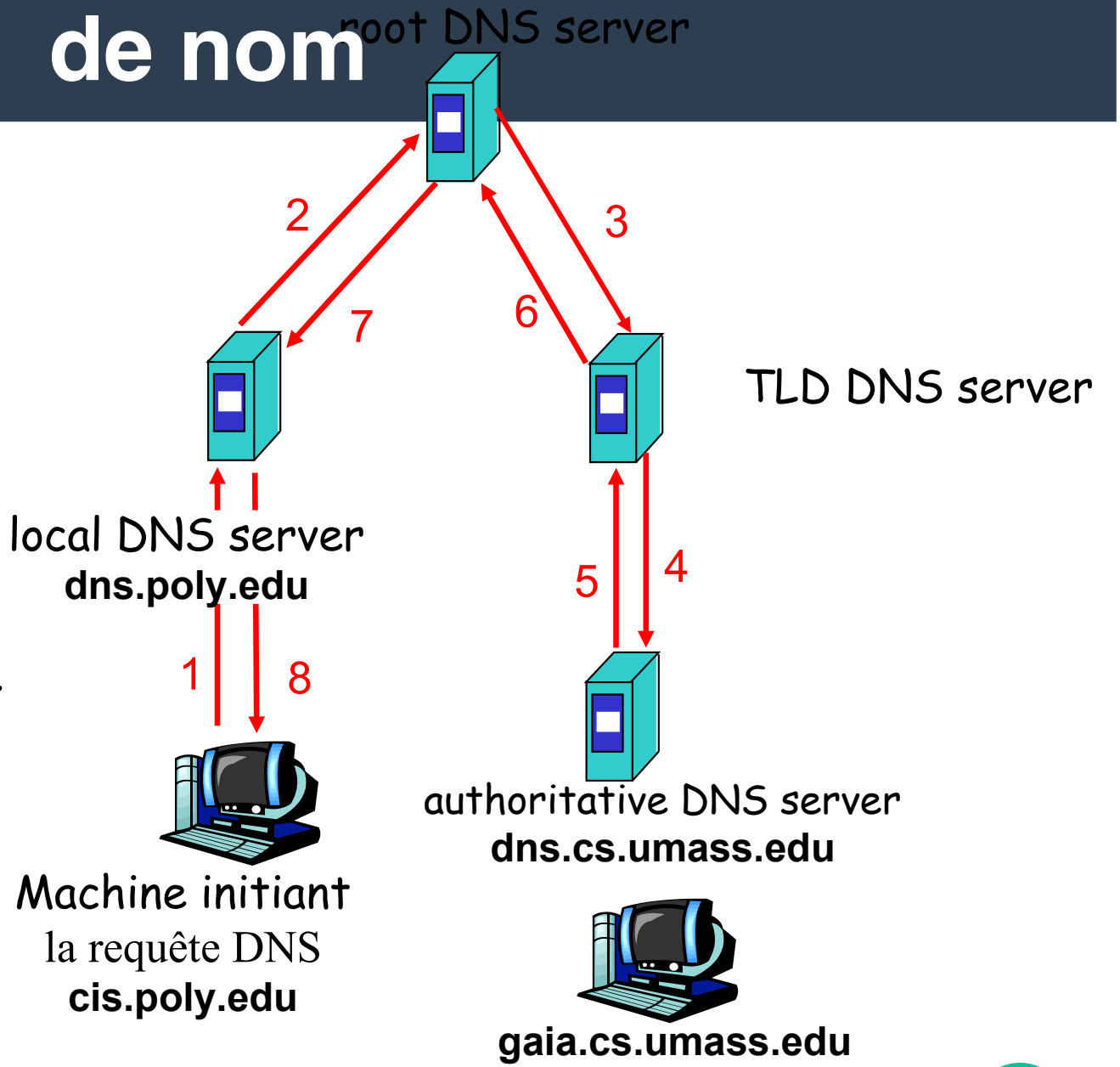
Exemple de résolution DNS

- **Etape 1 : client → resolver**
  - gaia.cs.umass.edu ?
- **Etape 2 : resolver → server root**
  - gaia.cs.umass.edu ?
- **Etape 3 : server root → resolver**
  - Connais pas mais va voir un serveur .edu (serveur TLD car edu est un TLD) dont voici l'adresse
- **Etape 4 : resolver → serveur TLD**
  - gaia.cs.umass.edu ?
- **Etape 5 : serveur TLD → resolver**
  - Connais pas mais va voir le serveur dns du domaine cs.umass.edu
- **Etape 6 : resolver → serveur DNS domaine cs.umass.edu**
  - gaia.cs.umass.edu ?
- **Etape 5 : serveur DNS domaine cs.umass.edu → resolver**
  - 128.119.245.12

Exemple de résolution de nom

Requête récursive :

- On demande au serveur suivant de chercher lui-même
- Ce faisant, on augmente sa charge de travail!
- En général, la requête est itérative
  - Sauf entre la machine qui initie et son serveur local (voir transparent précédent)



DNS: mise en cache des réponses

- **A chaque fois qu'un serveur apprend une traduction, il la stocke**
  - On dit qu'il la cache
    - (souvenez-vous des mémoires cache des ordinateurs)
  - Les entrées cachées ont une période d'expiration → après cette période, le serveur les détruit
  - Les serveurs de noms locaux cachent les adresses IP des serveurs TLD
    - Ainsi, les serveurs root sont moins souvent visités

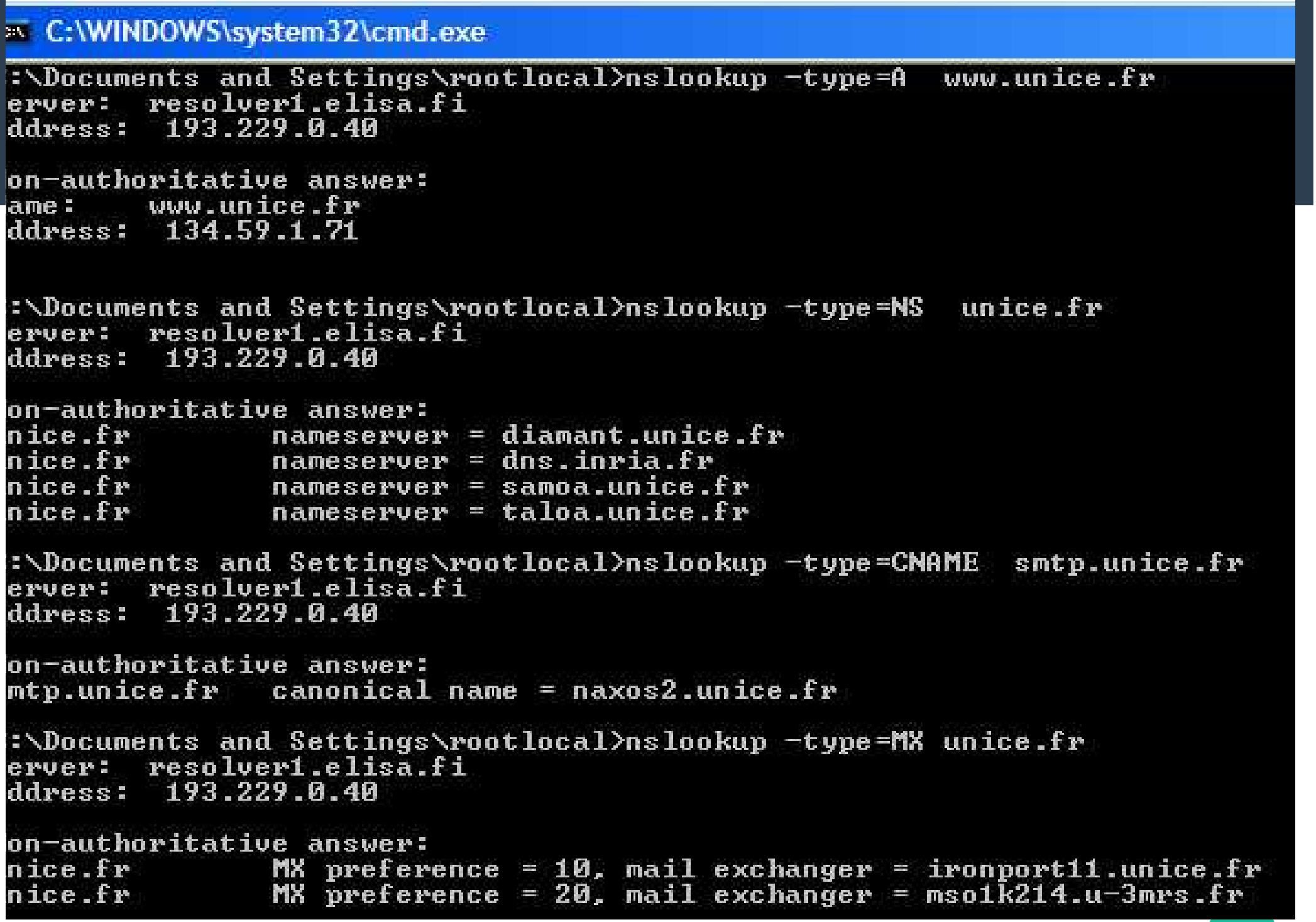
Enregistrements DNS

- **DNS: base de données distribuée**
- **Base de données : collection de données ayant une structure similaire**
  - Par exemple : (nom, numéro de téléphone)
- **Ces données sont appellés enregistrements (records en anglais)**
  - Pour le cas du DNS, on parle de “resource record” ou RR
- **Distribuée : les RR sont distribués sur un ensemble de machines physiques (les serveurs DNS)**

Enregistrements DNS

RR format: **(nom, valeur, type, ttl)**

- Type=A
  - **nom** = machine
  - **valeur** = addresse IP
- Type=NS
  - **nom** = domain (e.g. unice.fr)
  - **valeur** = nom du serveur DNS ayant autorité sur le domaine
- Type=CNAME
  - **nom** =alias
  - **valeur** = nom canonique (réel)
    - `www.lemonde.fr` est un alias de `www.lemonde.edgesuite.net`
- Type=MX
  - **valeur** est le nom du serveurde mail associé au nom

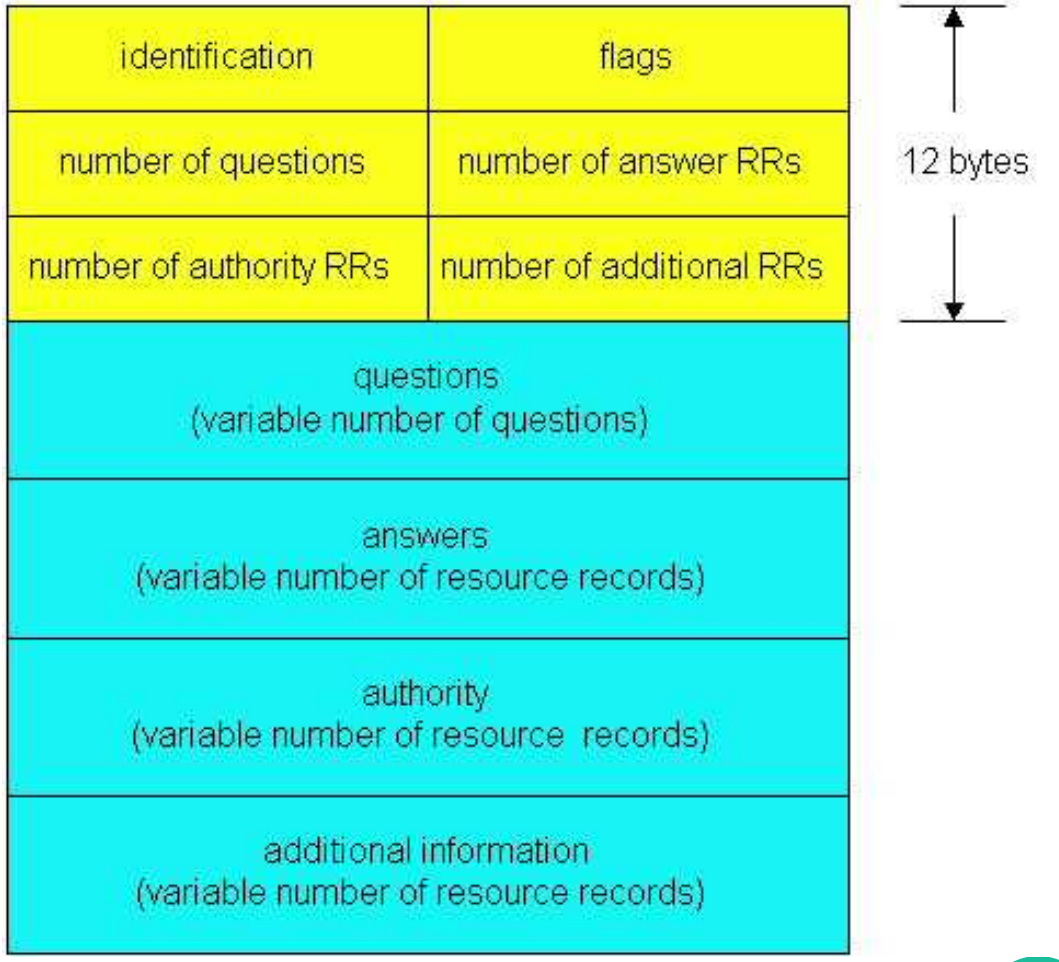

C:\WINDOWS\system32\cmd.exe
:\Documents and Settings\rootlocal>nslookup -type=A www.unice.fr
erver: resolver1.elisa.fi
ddress: 193.229.0.40
on-authoritative answer:
ame: www.unice.fr
ddress: 134.59.1.71
:\Documents and Settings\rootlocal>nslookup -type=NS unice.fr
erver: resolver1.elisa.fi
ddress: 193.229.0.40
on-authoritative answer:
nice.fr   nameserver = diamant.unice.fr
nice.fr   nameserver = dns.inria.fr
nice.fr   nameserver = samoa.unice.fr
nice.fr   nameserver = taloa.unice.fr
:\Documents and Settings\rootlocal>nslookup -type=CNAME smtp.unice.fr
erver: resolver1.elisa.fi
ddress: 193.229.0.40
on-authoritative answer:
mtp.unice.fr   canonical name = naxos2.unice.fr
:\Documents and Settings\rootlocal>nslookup -type=MX unice.fr
erver: resolver1.elisa.fi
ddress: 193.229.0.40
on-authoritative answer:
nice.fr   MX preference = 10, mail exchanger = ironport11.unice.fr
nice.fr   MX preference = 20, mail exchanger = mso1k214.u-3mrs.fr

Protocole DNS et messages

Protocole DNS : ***requêtes*** **et** ***reponses*** **suivant le même format de message**

En-tête

- identification: identifiant de 16 bits pour la requête. Réponse utilise le même identifiant
- flags: (drapeaux – série de bits à 1 (drapeau levé) ou 0 (drapeau baissé)) indiquent :
  - Requête ou réponse
  - Récursion désidée
  - Récursion disponible
  - La réponse émane d'un serveur ayant autorité



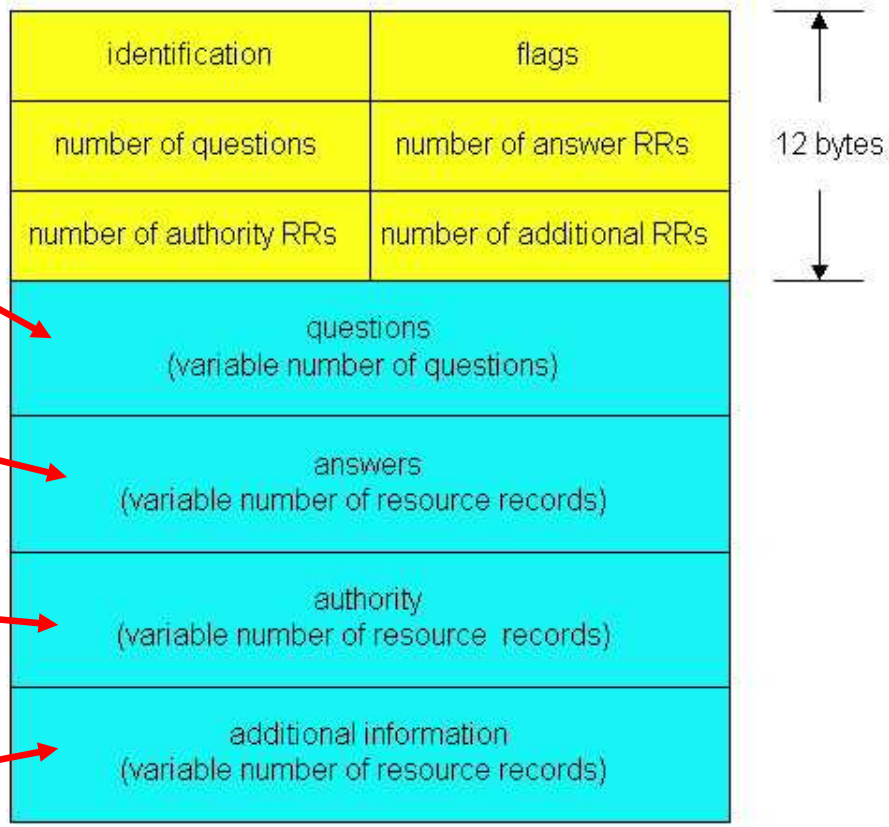
Messages DNS

Nom, type pour une requête

Enregistrements de la réponse (RR)

Enregistrement de la réponse si le serveur a autorité

Informations Supplémentaires
Ex: si réponse = nom de serveur de nom, info sup. = adresse IP de ce serveur



Comment insérer des enregistrements dans le DNS?

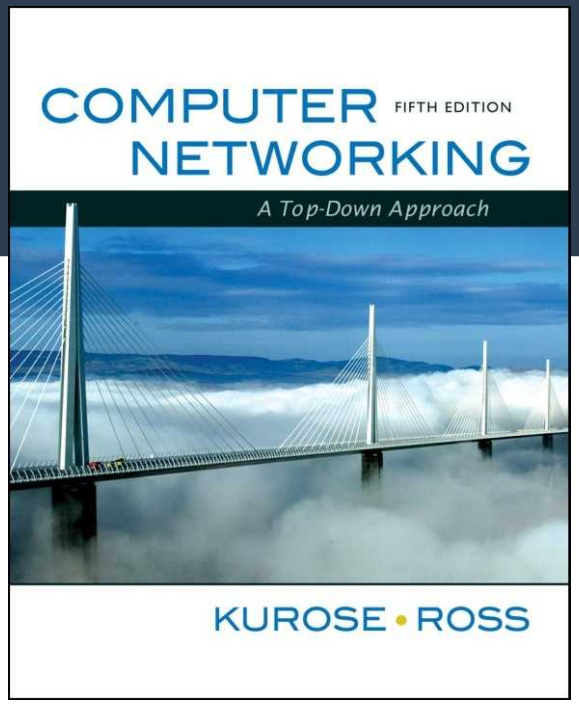
- **Exemple: une jeune pousse (startup) “Network Utopia”**

- **Etape 1: fait enregistrer le nom networkuptopia.com auprès d’une entité accréditée appelée *DNS registrar* (e.g., Network Solutions)**
  - NetworkUtopia doit fournie les noms et les adresses IP de serveurs de nom (un primaire, un secondaire) ayant autorité pour son domaine
  - registrar introduit les RRs correspondant dans le serveur TLD com :

  (networkutopia.com, dns1.networkutopia.com, NS)
  (dns1.networkutopia.com, 212.212.212.1, A)

- **Etape 2: insertion par Network Utopia d’un RR de Type A pour www.networkuptopia.com et de Type MX pour networkutopia.com dnas SON serveur ayant autorité**

Couche Transport



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in powerpoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2002
J.F Kurose and K.W. Ross, All Rights Reserved

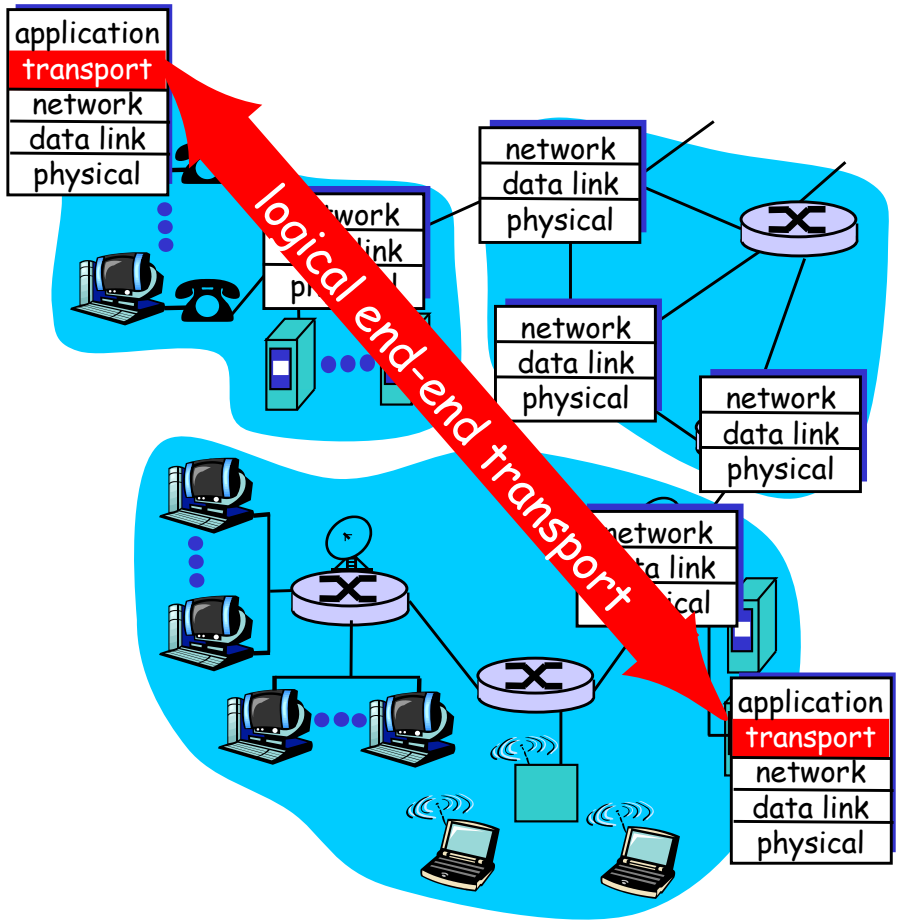
Computer Networking: A Top Down Approach 5th edition. Jim Kurose, Keith Ross Addison-Wesley, April 2009.

Plan

- Services offert par la couche Transport
- Multiplexage et démultiplexage
- Service sans connexion : UDP
- Service orienté connexion : TCP

Services de la couche transport et protocoles

- **Couche transport fournit *une communication logique* entre les processus applicatifs tournant sur des machines différentes**
  - Logique s'oppose ici à physique
- **Les protocoles de transport fonctionnent sur les machines du bord du réseau (client/serveur)**
  - Emetteur: coupe les messages applicatifs en segments et les passe à la couche réseau
  - Récepteur: re-assemble les segments en message qui sont donnés à la couche applicative
- **Deux protocoles de transport dans le monde IP: TCP et UDP**



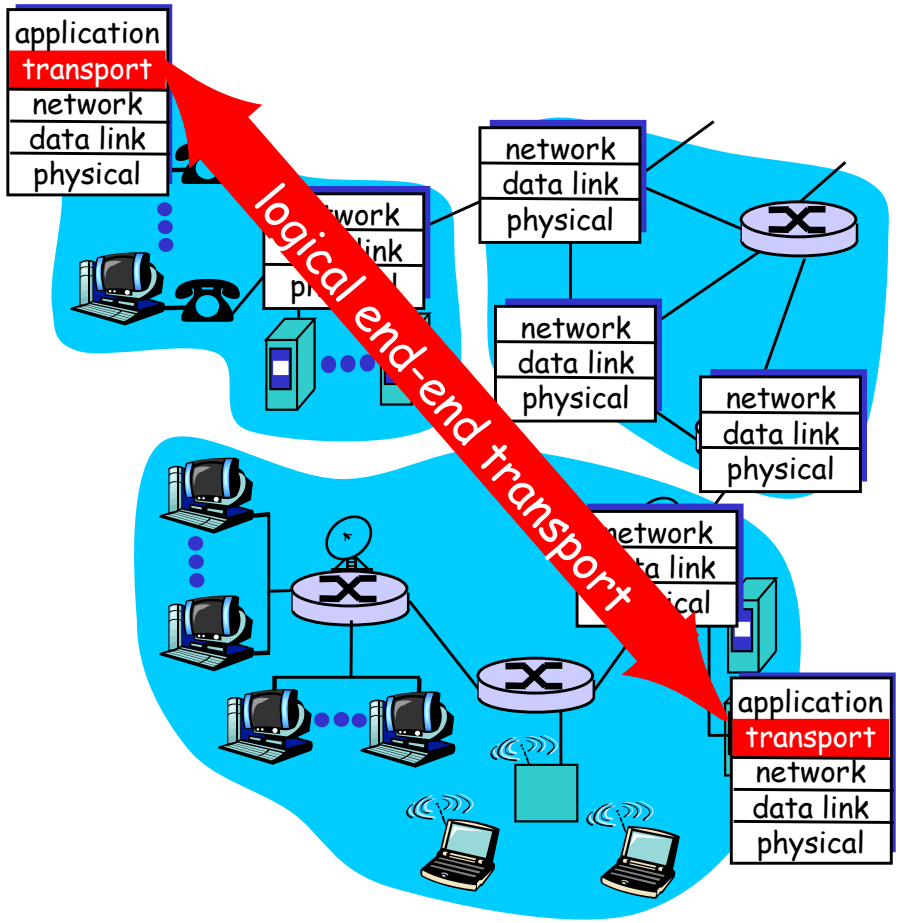
Couche transport versus couche réseau

- **Couche réseau: communication logique entre machines**
- **Couche transport: communication logique entre des processus**
  - Se base sur la couche réseau pour l'envoie effectif des données

Couches transport de l'Internet

- **2 services :**
  - Fiable (TCP)
  - Non Fiable (UDP)

- **Fiable (sans perte), livraison dans l'ordre des données : TCP**
  - Contrôle de congestion
  - Contrôle de flux
  - Etablissement de cx
- **Non fiable, sans garantie d'ordre dans la livraison: UDP**
- **Services non disponibles :**
  - Garanties de délai
  - Garanties de bande passante



Plan

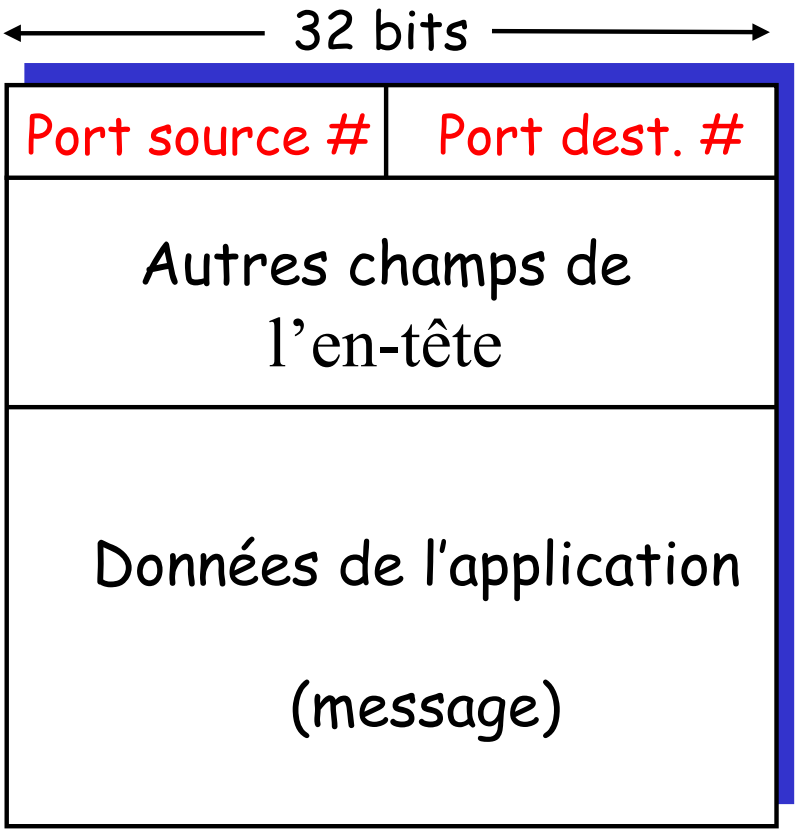
- Services offert par la couche Transport
- **Multiplexage et démultiplexage : service commun de TCP et UDP**
- Service sans connexion : UDP
- Service orienté connexion : TCP

Adresse d'un processus (= un programme)

- **Pour identifier un programme sur une machine, il faut :**
  - L'adresse de la machine → adresse IP
  - L'adresse du programme → numéro de port
    - Ex: adresse IP google + port 80 (port HTTP)
- **Pour une communication, on a 4 identifiants :**
  - Adresse machine source
  - Adresse machine destination
  - Adresse processus source : port source
  - Adresse processus destination : port destination
- **UDP et TCP utilisent ces 4 identifiants de manière différente**

Adressage d'un processus

- **Numéros de ports dans l'en-tête de couche transport (TCP ou UDP)**



Format segment TCP/UDP

Interface entre processus (application) et couche transport

- Pour ne pas avoir de blocage (ex: la couche transport attend que l'application soit prête à recevoir ses données), on met un tampon (réservoir d'octets) entre la couche transport et l'application!

- Ce buffer est appelé « socket »

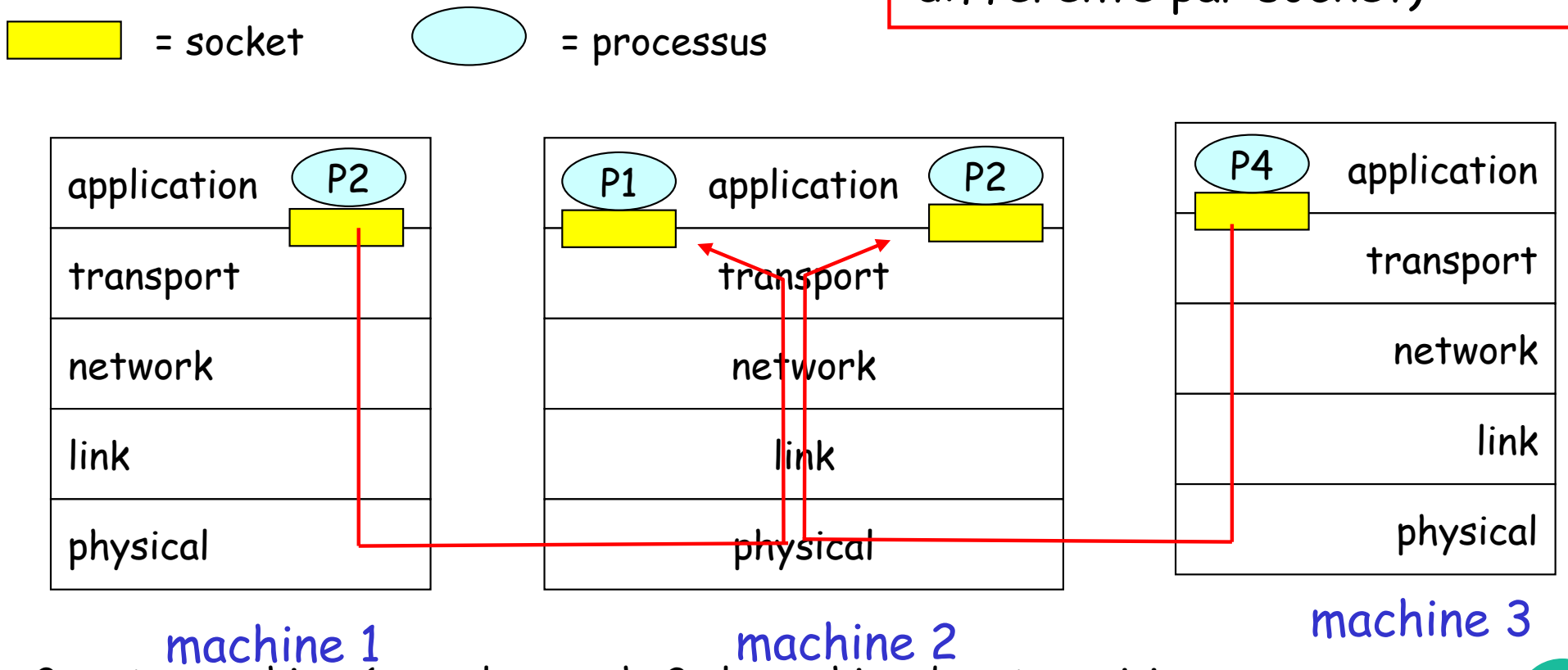
Multiplexage/démultiplexage

Démulti. côté récepteur:

Livraison des segments reçus aux bonnes sockets

Multi. Côté émetteur:

Collecte des données de toutes les sockets et ajoue d'en-têtes spécifiques (une différente par socket)



Ex: 2= votre machine, 1=google search, 3= la machine de votre voisin

Démultiplexage pour service sans connexion

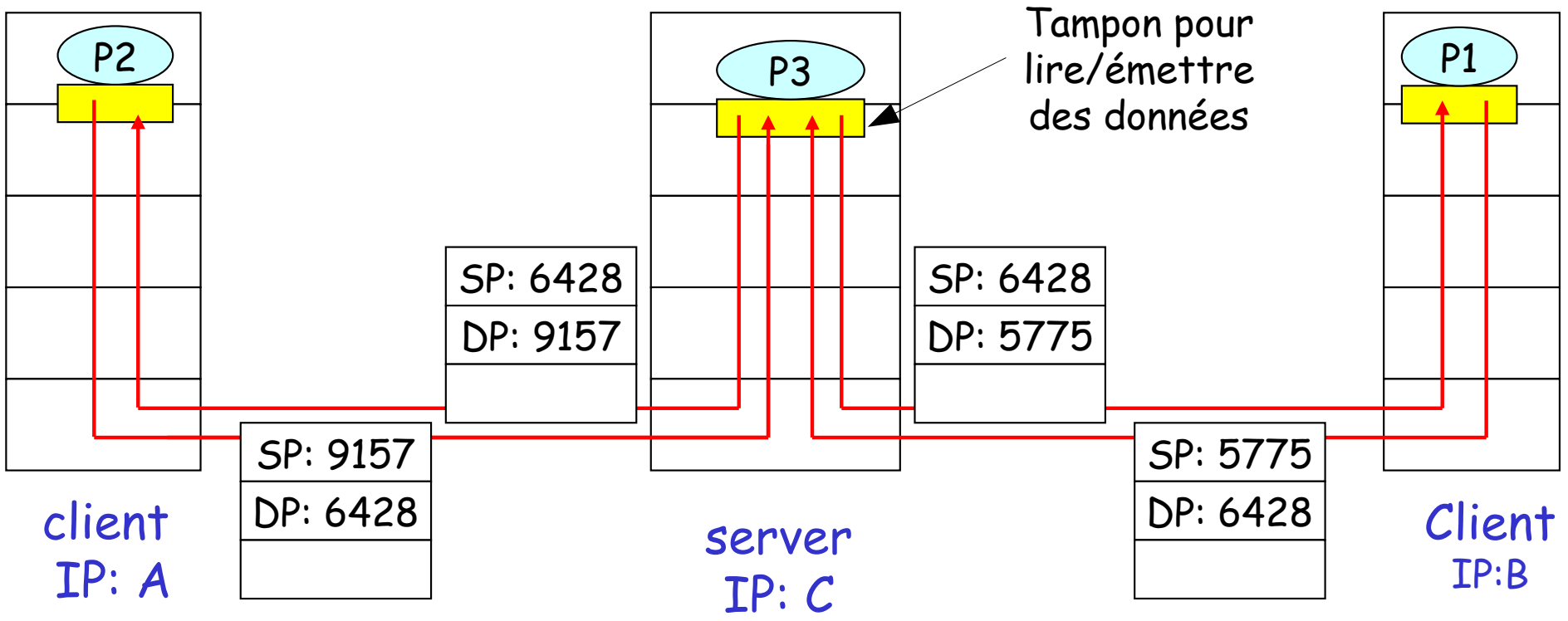
- **Socket UDP (tampon de données) identifiée par couple :**

(addresse IP dest., numéro port dest)

- **Quand une machine reçoit un segment UDP :**
  - Lit port destination dans segment
  - Dirige le segment UDP dans la socket correspondante
- **Datagrammes IP avec adresse IP source/port source différents mais même port destination dirigé sur la même socket → un processus pour toutes les machines distantes qui accèdent à ce port (service)**

Démultiplexage pour service sans connexion
Identifiant socket = (IP_C, 6428)

```
DatagramSocket serverSocket = new DatagramSocket(6428);
```



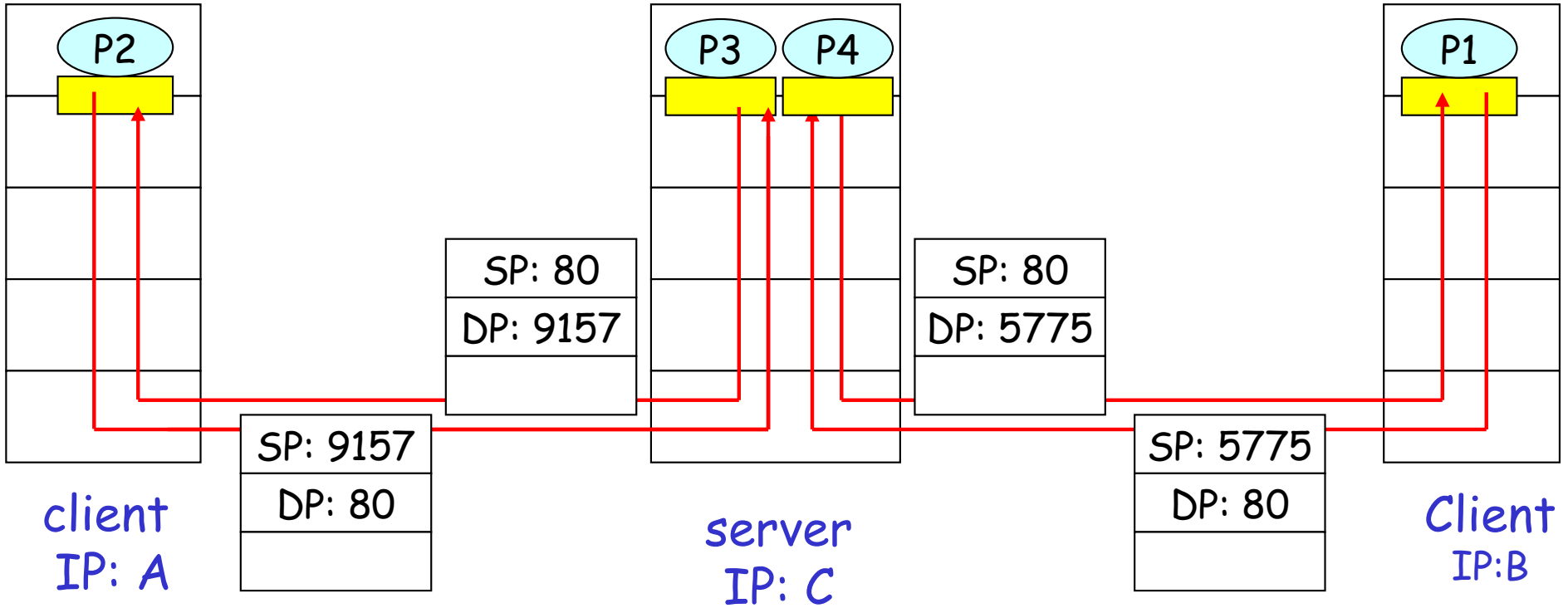
SP= port source → fournit l'adresse de retour
DP= port destination (adresse locale du processus applicatif)

Démultiplexage pour le service orienté connexion

- **Les sockets TCP sont identifiées par un quadruplet :**
  - Adresse IP source
  - Numéro port source
  - Adresse IP destination
  - Numéro port dest.
- **Récepteur utilise les 4 infos pour aiguiller le segment vers la socket appropriée**

- **Ex: on aura un processus Web pour chaque client**
- **Plus simple qu'un même processus qui gère tous les clients**

Démultiplexage pour le service orienté connexion
Identifiants sockets :
(IP_A, 9157, IP_C,80) et (IP_B, 5775, IP_C,80)



Visualisation des sockets dans votre machine (Linux)

- **Sockets serveur**
  - Socket sur lesquelles écoutent des services
  - Commande netstat avec les options l = listen, t=tcp, u=udp, p= programme qui écoute
  - Attention, si pas ‘u’ et ‘t’ vous avez aussi les sockets unix entre processus dans la machine…. Et il y en a beaucoup !
- **Sockets clients**
  - Pas de ‘l’

```
root@mininet-vm:/home/mininet# netstat -ltunp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1120/sshd
udp        0      0 0.0.0.0:21280           0.0.0.0:*                           1001/dhclient
udp        0      0 0.0.0.0:31211           0.0.0.0:*                           3254/ntpd
udp        0      0 0.0.0.0:68              0.0.0.0:*                           3264/dhclient
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1001/dhclient
udp        0      0 172.19.130.56:123       0.0.0.0:*                           3254/ntpd
udp        0      0 127.0.0.1:123           0.0.0.0:*                           3254/ntpd
udp        0      0 0.0.0.0:123             0.0.0.0:*                           3254/ntpd
root@mininet-vm:/home/mininet# netstat -tunp
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 172.19.130.56:22        172.19.130.7:56499      ESTABLISHED 3270/sshd: mininet
root@mininet-vm:/home/mininet#
```

Plan

- Services offert par la couche Transport
- Multiplexage et démultiplexage
- Service sans connexion : UDP
- Service orienté connexion : TCP

UDP: User Datagram Protocol [RFC 768]

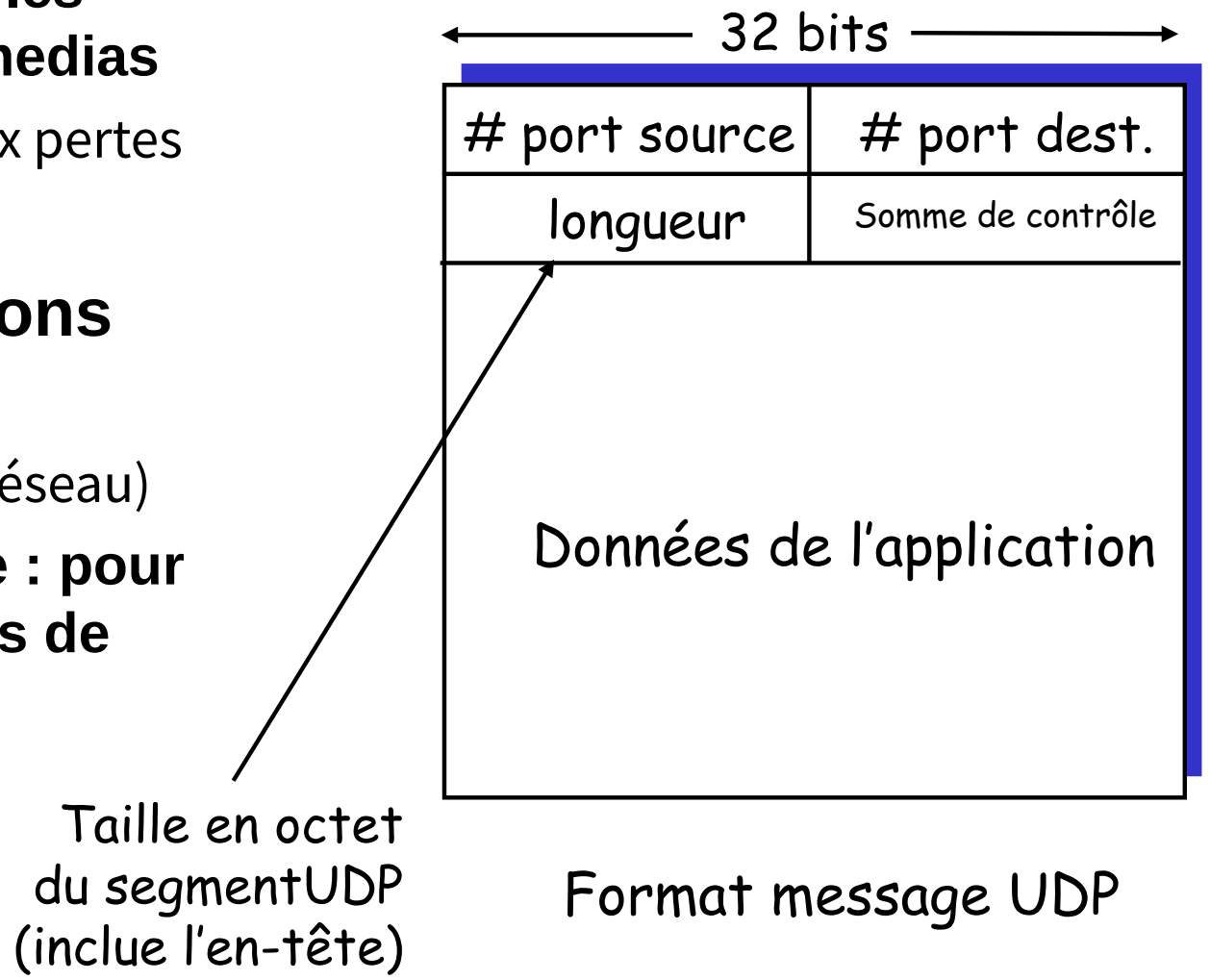
- **Service de transport Internet minimum**
- **Fonctionne en mode “best effort”, i.e., fait de son mieux. Les segments peuvent être:**
  - Perdus
  - livrés dans le désordre
- ***Sans connexion***
  - Pas de poignée de main (voir TCP) entre émetteur et récepteur

Pourquoi UDP existe-il?

- **Pas de temps d’établissement de connexion → gain de temps sur le premier segment**
  - Intéressant pour le DNS
- **Attention, cela ne veut pas dire qu’UDP est plus rapide que TCP.**
  - Seulement pour le premier segment !
- **UDP envoie au débit demandé par l’application. Si l’application envoie trop vite, données perdues.**
  - Intéressant pour les applications de voix type VoIP ou Skype qui envoie à un débit faible et constant
  - Dangereux pour les autres applications car pas de contrôle de congestion !

UDP: quelques détails

- **Souvent utilisé par les applications multimedias**
  - Appli tolérantes aux pertes
  - Sensibles au délai
- **Autres applications**
  - DNS
  - SNMP (gestion de réseau)
- **Somme de contrôle : pour détecter des erreurs de transmission**



Format message UDP

UDP en pratique

- **Ex d'applications multimedia**
  - YouTube, Dailymotion (vidéo) - streaming
  - Deezer, Shoutcast (audios) - streaming
  - Skype, Voix sur IP en entreprise ou sur Internet (voix)

- **Streaming :**
  - indique que les données sont consommées à la volée par le client
  - Les données sont stockés sur le serveur (fichier audio ou vidéo)
  - Les données vont dans un seul sens (serveur → client)

- **En pratique**
  - Dans la liste précédente, seuls Skype et les clients Voix sur IP utilisent UDP
  - Les autres utilisent TCP!

Plan

- **Services offert par la couche Transport**
- **Multiplexage et démultiplexage**
- **Service sans connexion : UDP**
- **Service orienté connexion : TCP**
  - Principes et Structure segment
  - Ouverture de connexion
  - Les accusés de réception
  - Contrôle de congestion
  - Contrôle de flux

TCP

- Si on analyse un lien Internet quelconque, 90% des octets sont transportés par TCP
- Le reste est en général UDP
- Et souvent, UDP correspond au DNS

- Un exemple de capture journalière de trafic est accessible à l'adresse :

  http://mawi.wide.ad.jp/mawi/

TCP: principes

1) Etablissement de connexion préalable au transfert des données → ouverture de socket, réservation de buffer

2) Récupération des pertes (perte ≠ erreur) avec accusés de réception (ACK) et lutte contre les déséquencements de segments.

3) Pertes interprétées comme signal de congestion → diminution débit

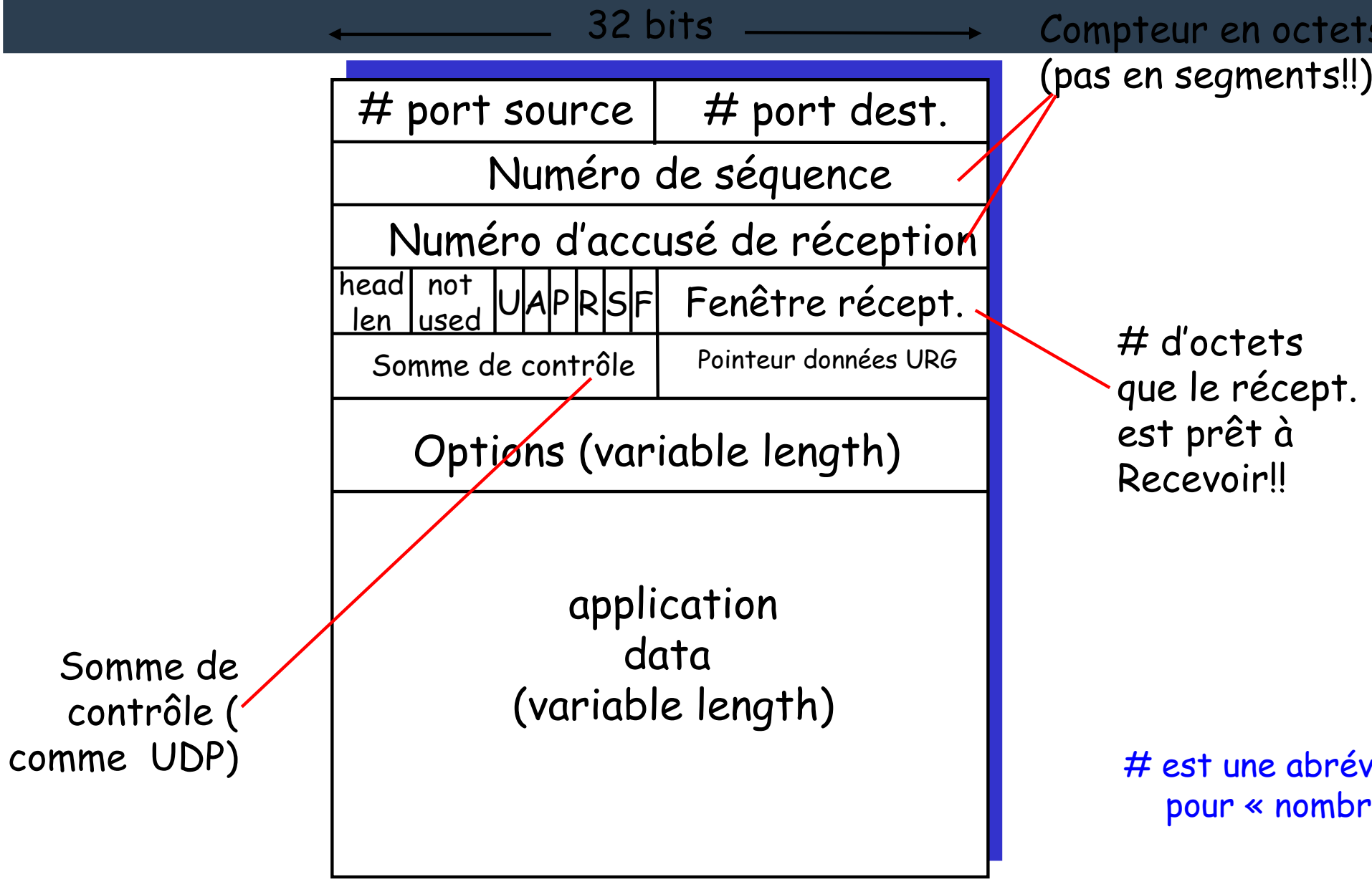
4) Contrôle de flux: pour que l'émetteur n'émette pas trop vite par rapport au récepteur

Principe 1 : établissement de connexion – la triple poignée de main

- Séquence de segments TCP (vides – pas de données) mais avec des drapeaux spéciaux

- Réservation de buffer (la socket)

Structure du segment TCP



est une abréviation pour « nombre »

Principe 1 : la triple poignée de main – utilisation des drapeaux

- **L'en-tête TCP contient des drapeaux (bits que l'on met à 0 ou 1) : U,A,P,R,S,F = drapeaux (1 bit – oui ou non)**

- **ACK: pour les seg. qui sont seulement des acquittements**
  - = pas de données (applicatives) transportées
  - Acknowledgment en anglais – on emploiera l'abbréviation ack ci-après

- **SYN, FIN:**
  - Pour l'établissement et la fermeture des connexions

- **RST: RESET - URG: données urgentes - PSH: push data now**
  - Non traités en M1104

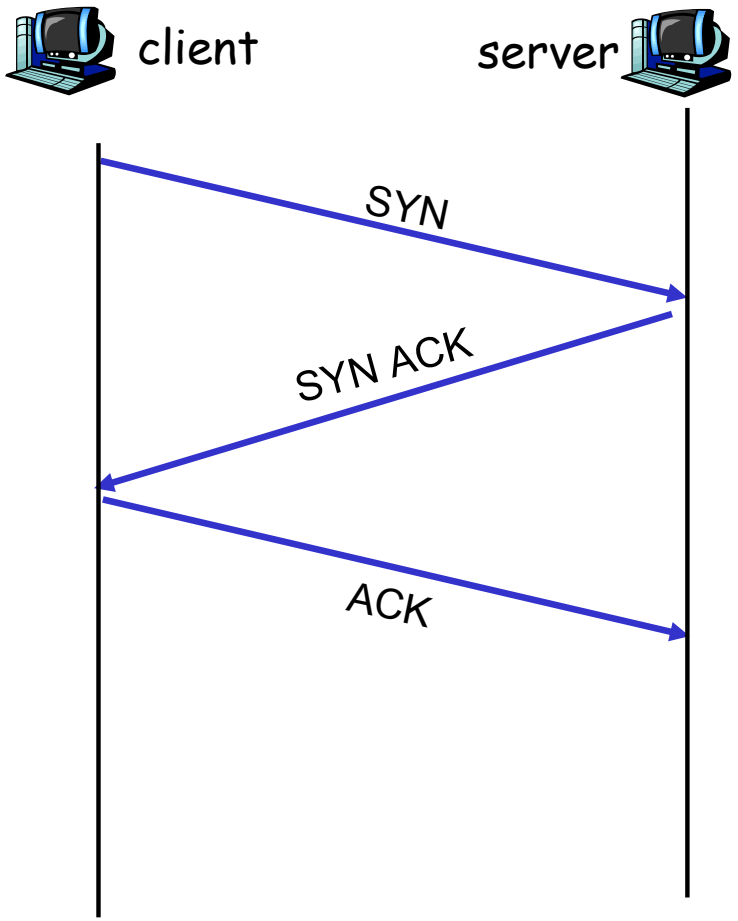
Ouverture de connexion = triple poignée de main

Étape 1 : client envoie un segment avec drapeau SYN au serveur

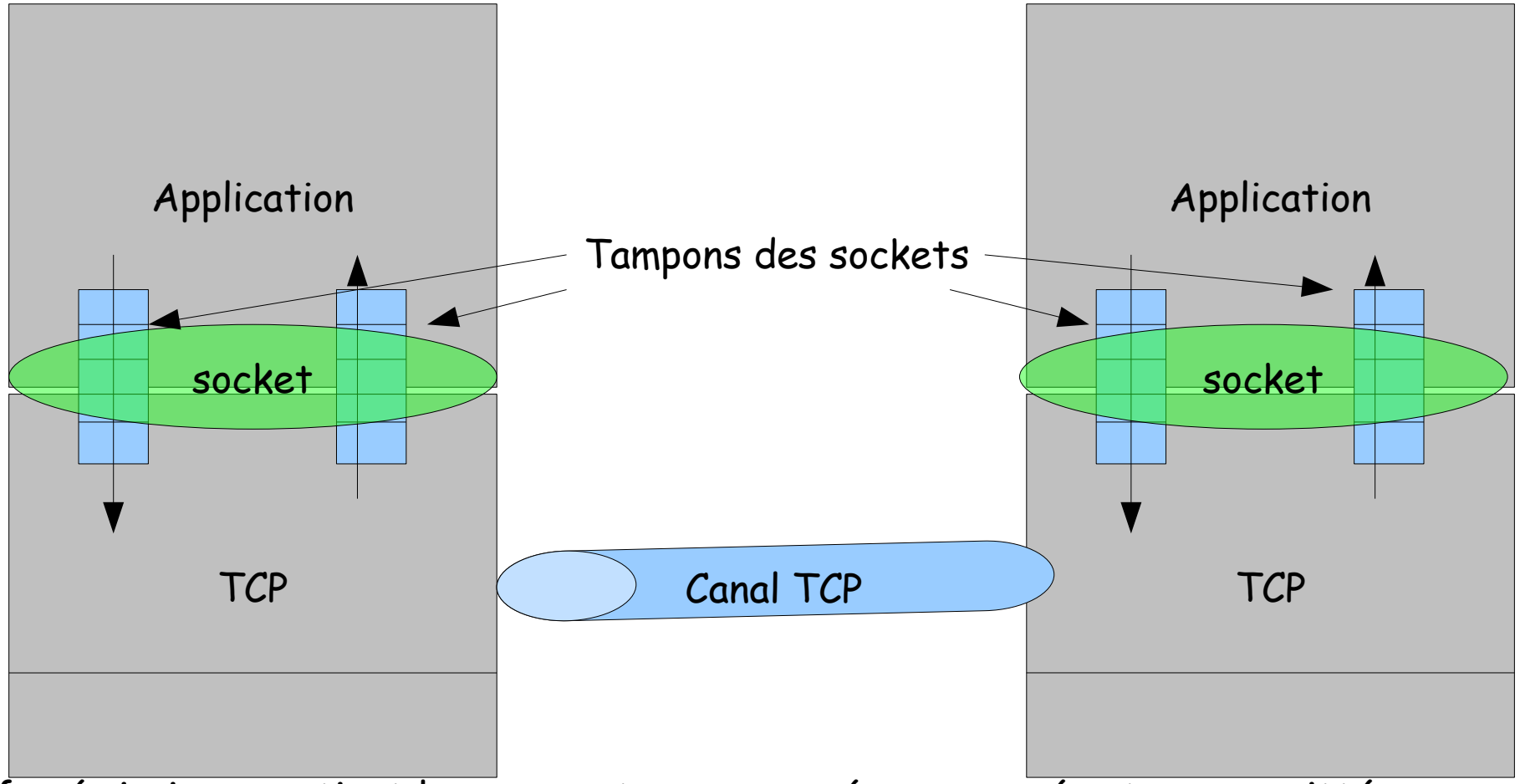
Étape 2: serveur reçoit SYN et répond avec SYN et ACK

Étape 3: client répond avec un ACK

Les segments des étapes 1 et 2 sont vides (pas de données). A l'étape 3, on peut avoir des données.



Les différents tampons



Buffer émission : contient les segments non envoyés ou envoyés et non acquittés
Buffer réception : contient les segments reçus.
L'application n'a le droit de les lire que si ils sont dans l'ordre.
Ex : si '1' et '3 ' reçus mais pas '2', l'application a le droit de lire '1' seulement

Les différents tampons

- **Une socket contient un buffer pour émission des données Application → TCP et un pour réception TCP → Application**
  - Buffer emission : segments envoyés sont gardés en mémoire tant que l'émetteur n'est pas sûr qu'ils ont été reçus
  - Buffer réception: si récepteur reçoit les segments 1,3 puis 2 : il donne 1 à l'application puis garde 2 en mémoire tant qu'il n'a pas reçu 2 car les données doivent être données dans l'ordre!

Principe 2: utilisation d'accusés de réception

- **Les données peuvent circuler dans les 2 sens**
- **TCP transporte des octets et donc tous les numéros de séquence/ack sont en octets!**
- **Pour un sens donné:**
  - Le numéro de sequénce correspond au premier octet porté par le segment
  - Le numéro d'ack au numéro du prochain octet attendu dans l'autre sens.

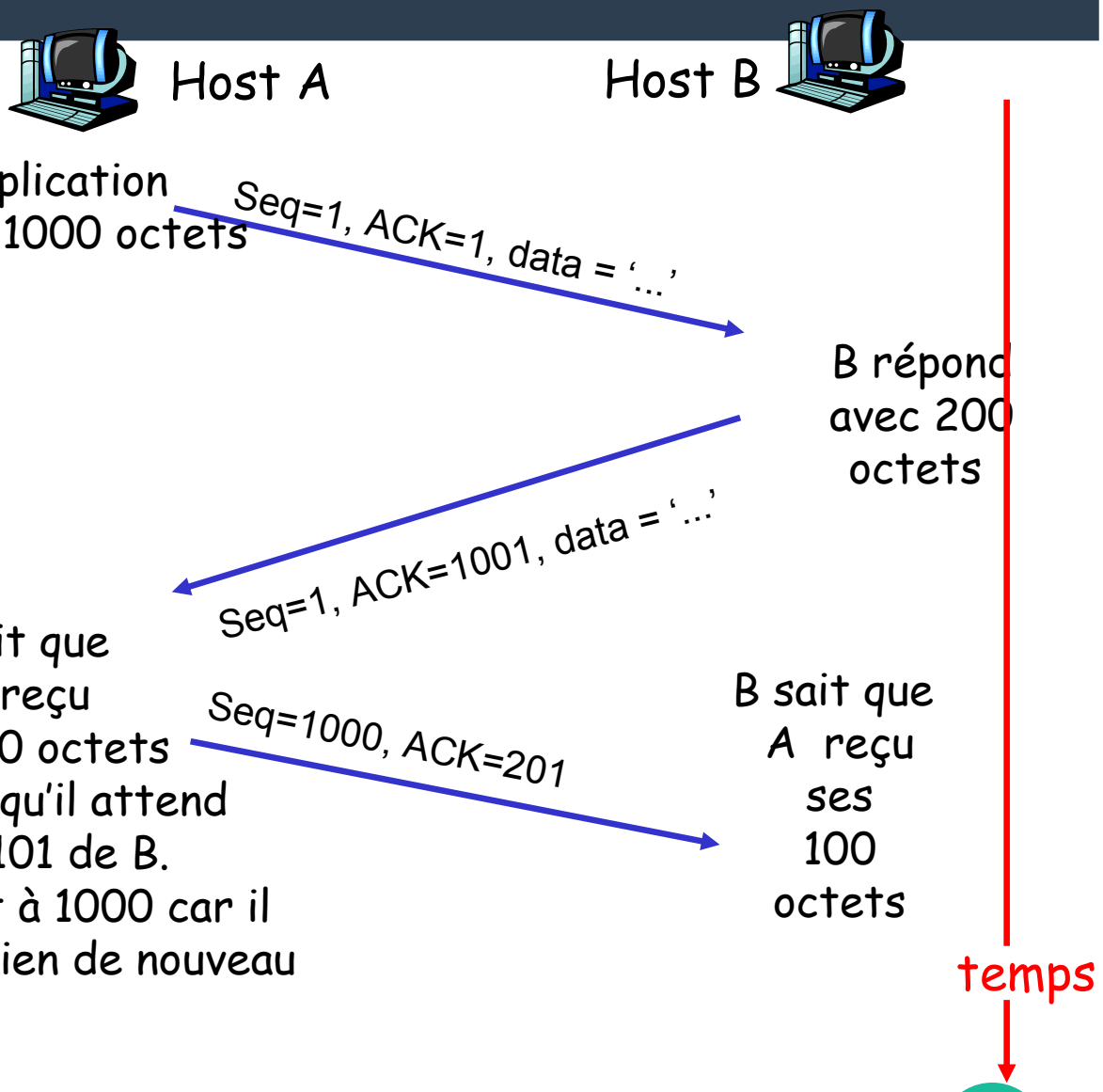
Numéro de seq. et d'ack.

Seq. numbers:

- Numéro, dans le flux d'octets, du premier octet transporté de A vers B ou de B vers A

ACKs:

- Numéro de seq. du prochain octet attendu par émetteur
- ACK cumulatifs
  - Voir ci-après



Transfert fiable de données avec TCP

- **Comment l'émetteur sait qu'il y a une perte?**
- **TCP observe les numéros de séquence !!!!**
- **Principe : si il observe des trous, il renvoie**
  - Plus de détails en 2ème année

Principe 3: contrôle de congestion

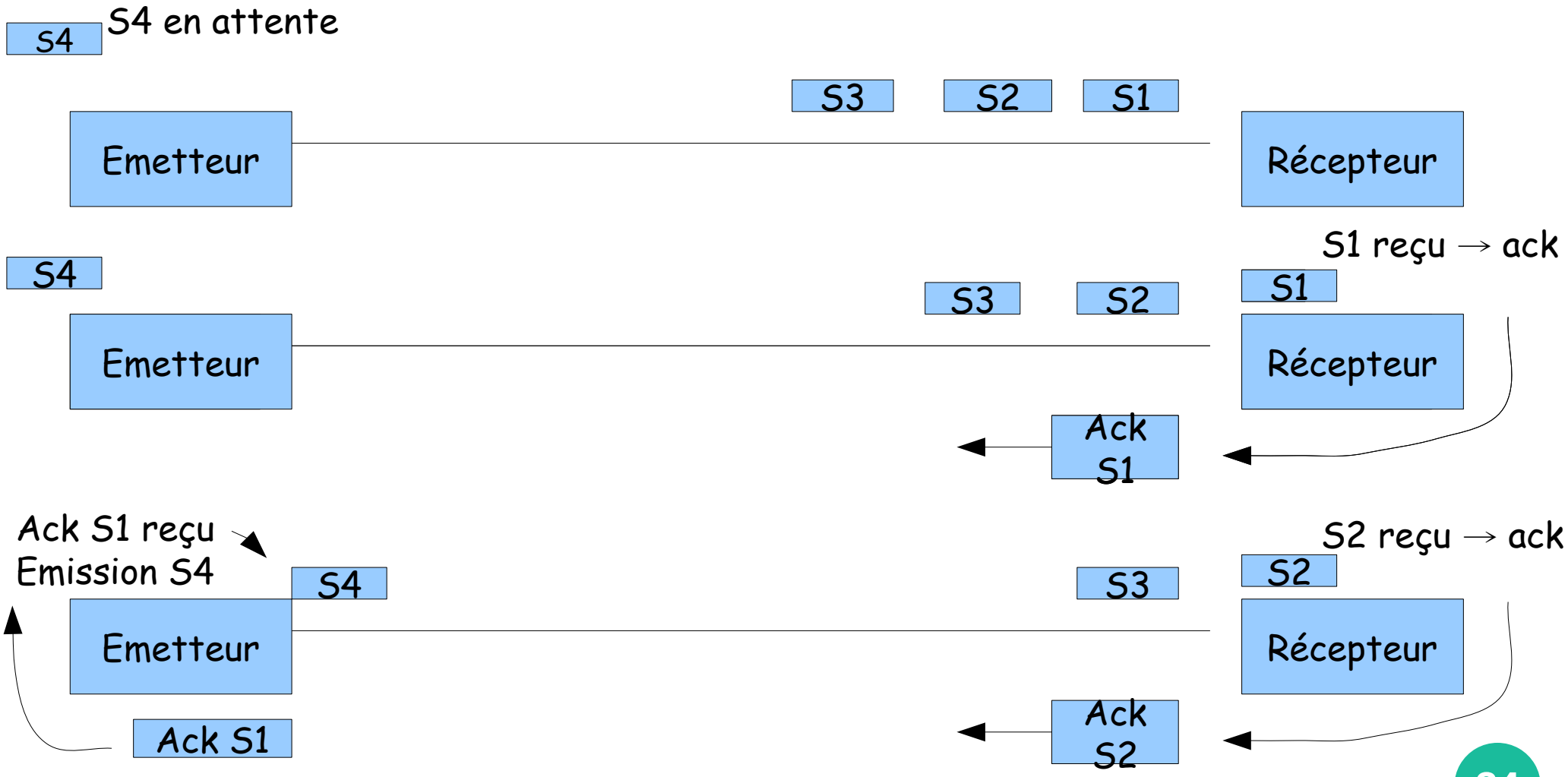
- **Renvoyer un paquet perdu n'est pas suffisant**
- **Il faut aussi diminuer le débit d'envoie des données**
  - Pourquoi? Car une perte est le signe d'une congestion dans le réseau

Contrôle de congestion

- **Contrôle de congestion par fenêtrage:**
  - TCP envoie plusieurs segments correspondant à W octets maximum sur la ligne à la suite
  - A chaque fois qu'il reçoit l'accusé de réception d'un segment, il a le droit d'envoyer un autre segment
  -
  - W (=nombre d'octets « en-vol ») est modulé en fonction de la charge du réseau suivant le principe
    - Si perte de segment détectée, TCP réduit W
    - Sinon, on cherche à augmenter W

Exemple simplifié (fenêtre en segment et non en octets – comme TCP)

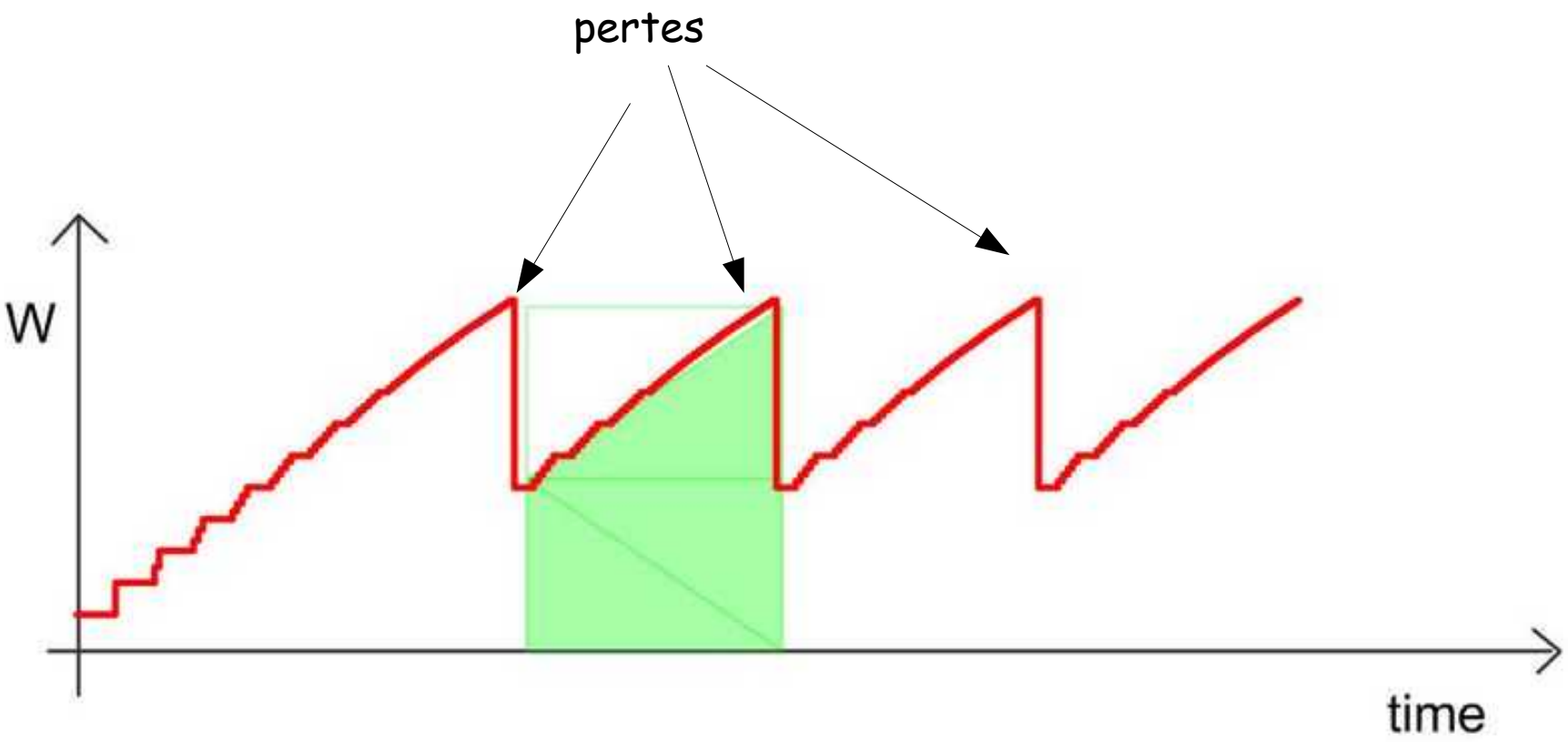
- Un exemple simplifié où la fenêtre fait 3 segments
- W=3 => Emetteur bloqué au 4ème segment, en attente de ACK de 1



Exemple simplifié

- **Suite de l'exemple. Emetteur reçoit l'ACK des 3 premiers paquets => pas de perte et fenêtre augmente (W=W+1 – détail en 2ème année)**

Variation de la fenêtre de congestion en pratique : un mouvement... perpétuel



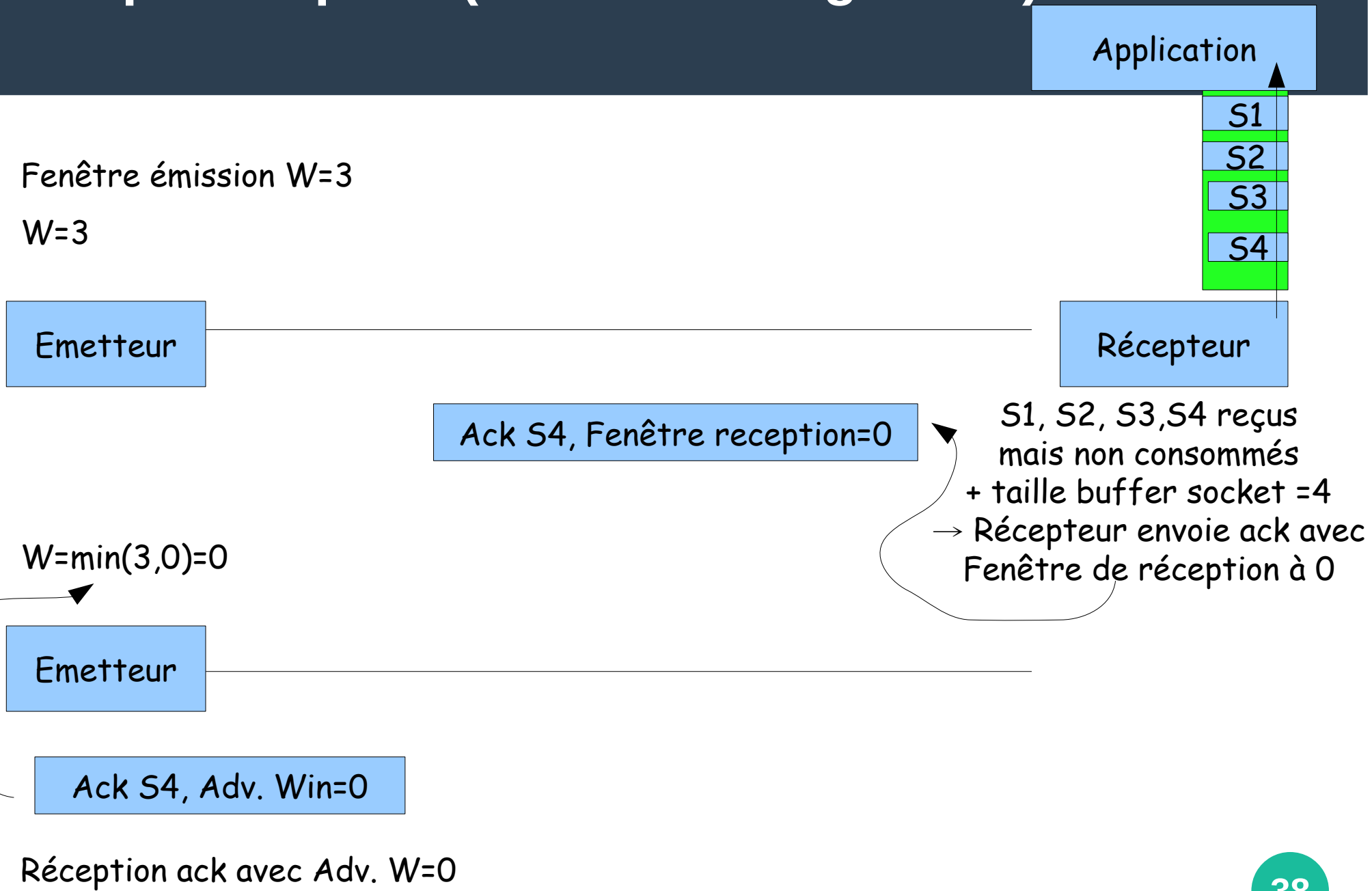
Principe 4 : Contrôle de flux

- Contrôle de congestion traite le problème suivant :
  - Pas de perte dans le réseau MAIS
  - L'émetteur envoie trop vite par rapport à l'application réceptrice
  - Pertes au niveau du récepteur

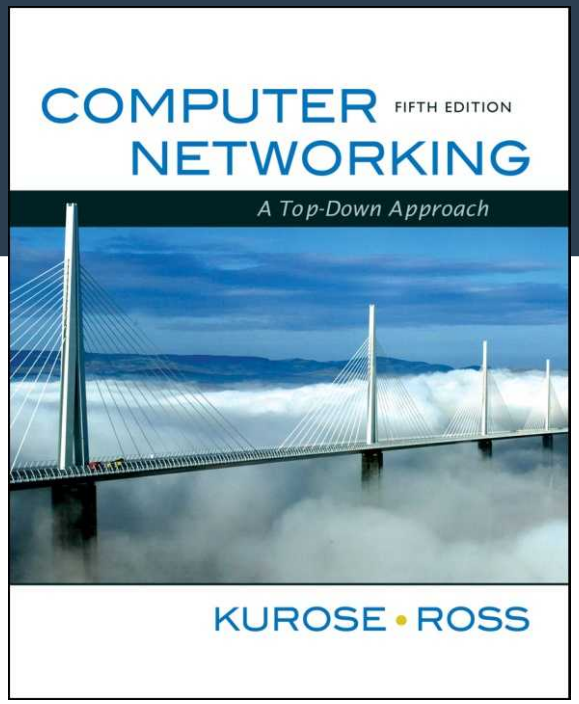
- Pour éviter ce problème :
  - TCP, on surveille le buffer TCP/Appli côté récepteur (niveau socket)

- TCP envoie le nombre d'octets restant dans le buffer dans le champ fenêtre de réception (Adv_Window) de l'en-tête

Exemple simplifié (fenêtre en segments)



Couche Réseau



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in powerpoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2009
J.F Kurose and K.W. Ross, All Rights Reserved

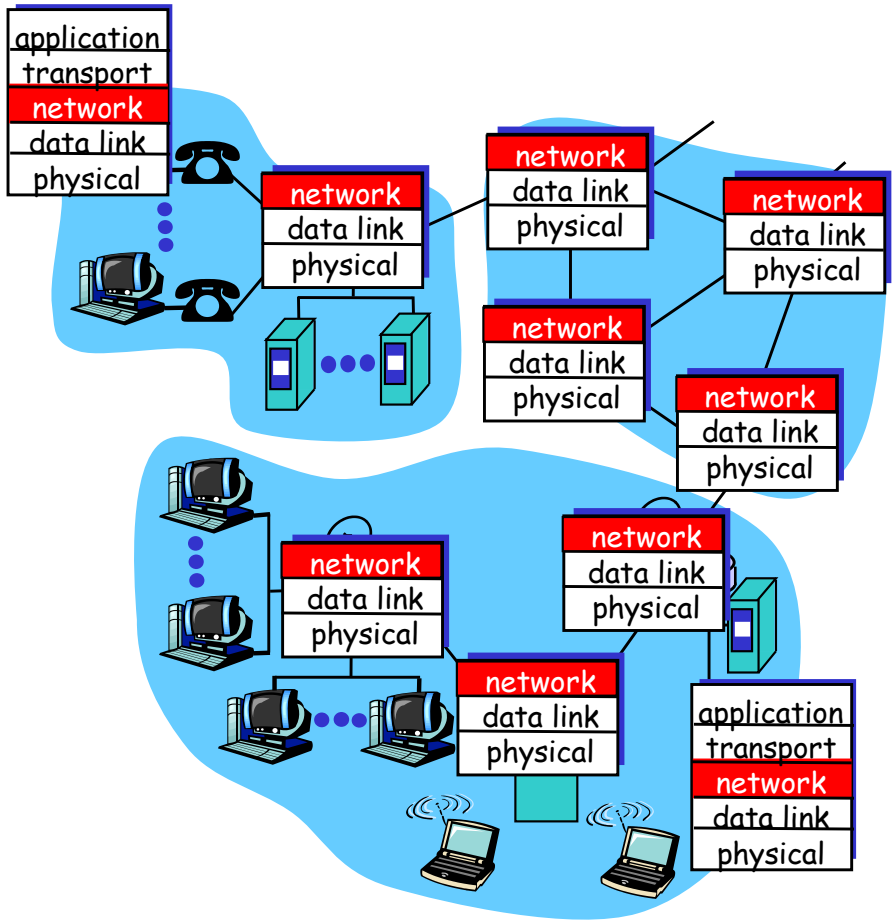
Computer Networking: A Top Down Approach 5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.

Agenda

- Introduction et Service de la couche Réseau
- Principes du routage
- Le routage hiérarchique de l’Internet
- Le protocole Internet (IP)
- Le protocole ICMP

Functions de la couche réseau

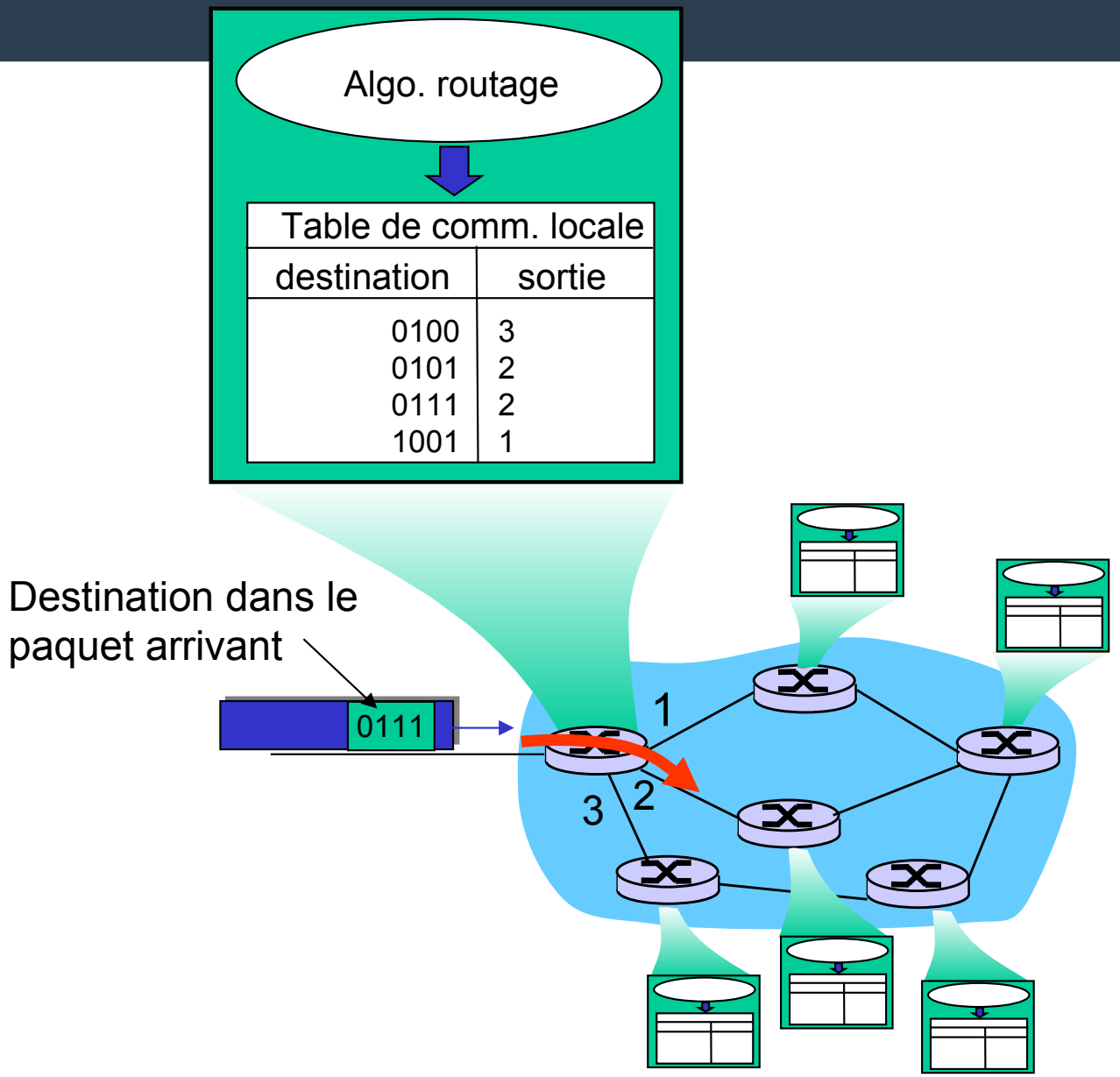
- **Acheminement des paquets entre un émetteur et un récepteur**
- **La couche réseau est implantée dans chaque machine et dans chaque routeur de l'Internet**



Deux functions importantes:

- ***Détermination du chemin:*** **route prise par les paquets de la source à la destination → algorithme de routage**
- ***Commutation de paquets:*** **choix de l'interface de sortie d'un routeur en fonction de la destination du paquet**

Routage et commutation

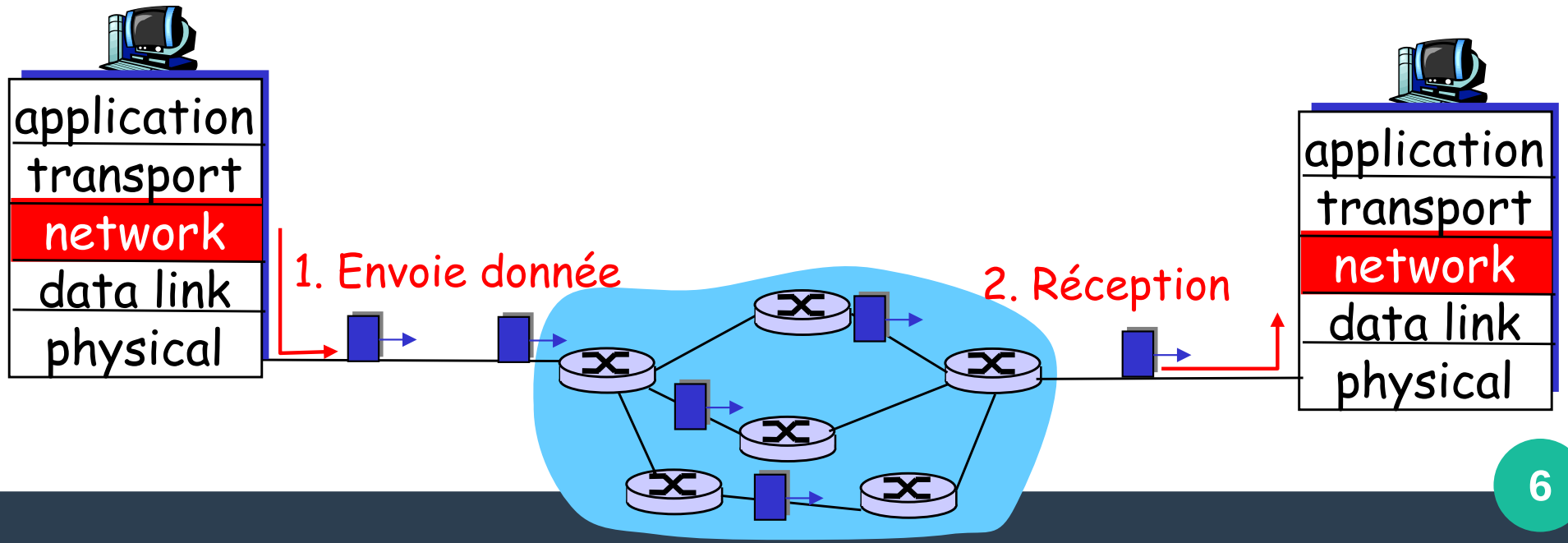


Routage vs. commutation

- **La commutation utilise les informations de routage pour choisir les interfaces de sorties en fonction des interfaces d'entrées**
  -
- **La commutation est une opération sensible car elle a lieu en temps réel**
  - 1 paquet arrive sur une interface d'entrée à 10 Gbps et il faut prendre de suite la décision vers quelle sortie le commuter
  - Algorithmes temps-réels complexes
  - Non traitée dans le cours

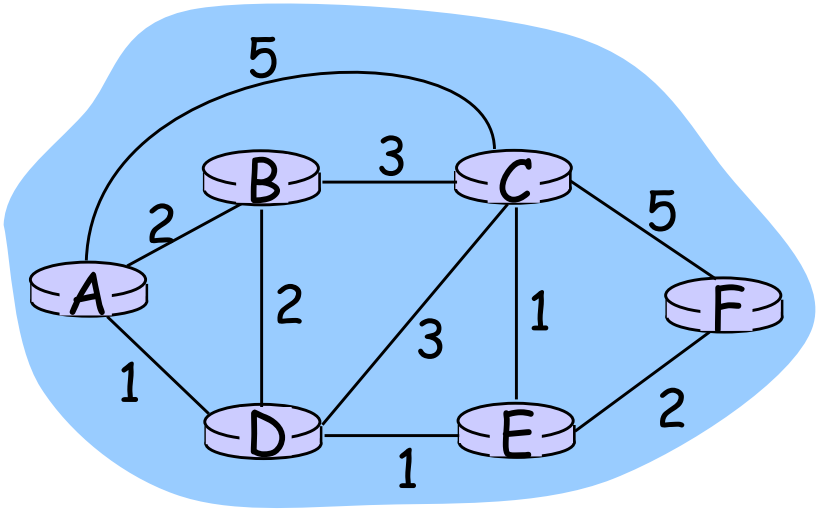
Le modèle de l'Internet : le mode datagramme

- **Pas d'établissement de connexion au niveau de la couche réseau**
  - S'oppose au modèle du réseau téléphonique commuté ou du minitel par exemple
- **Routeurs: ne gardent pas d'état sur les connexions de bout en bout**
- **Les paquets (datagrammes) sont commutés uniquement en fonction de leur adresse (IP) de destination**
  - Les paquets d'une même paire source-destination peuvent suivre des chemins différents dans le réseau
  - Rare en pratique, mais néanmoins observable



Protocole de routage

But: determiner le bon chemin (séquence de routeurs) au travers du réseau pour toutes les destinations



Le problème de ces algorithmes est un problème de graphe :

- **Les nœuds du graphe sont les routeurs**
- **Les liens du graphe les liens physiques**
  - Liens ont des coûts: délai, prix, niveau de congestion

- Qu'est-ce qu'un "bon chemin"?
  - Typiquement, un chemin de coût minimum

Classification des algorithmes de routage

Premier niveau important de classification : chaque nœud possède une vision totale ou partielle du réseau?

Globale:

- **Tous les routeurs connaissent la topologie (le graphe) complet avec le coût des liens**
  - Échange des information de topologie entre voisins
  - Calcul des chemins effectué indépendamment par chaque noeud
- **Classe d'algorithme à "état de liens"**

Décentralisée:

- **Un routeur connaît ses voisins qui lui sont physiquement connectés avec le coût de ces liens**
- **Processus iteratif de calcul des chemins par échange d'information entre voisins**
- **Classe d'algorithmes à "vecteur de distance"**

Agenda

Introduction et Service de la couche Réseau

Principes du routage

Le routage hiérarchique de l’Internet

Le protocole Internet (IP)

Le routage hiérarchique

Notre introduction au routage était idéalisée. Nous supposions :

- Que tous les routeurs étaient identiques
- Que le réseau était homogène

… pas le cas en pratique

Taille: 200 millions de destinations possibles:

- **On ne peut pas stocker ces infos dans une table de routage!**
- **L'échange des tables de routage ne serait pas possible non plus!**

Autonomie administrative

- **internet = réseau de réseaux**
- **Chaque administrateur de réseau veut contrôler le routage dans son réseau**

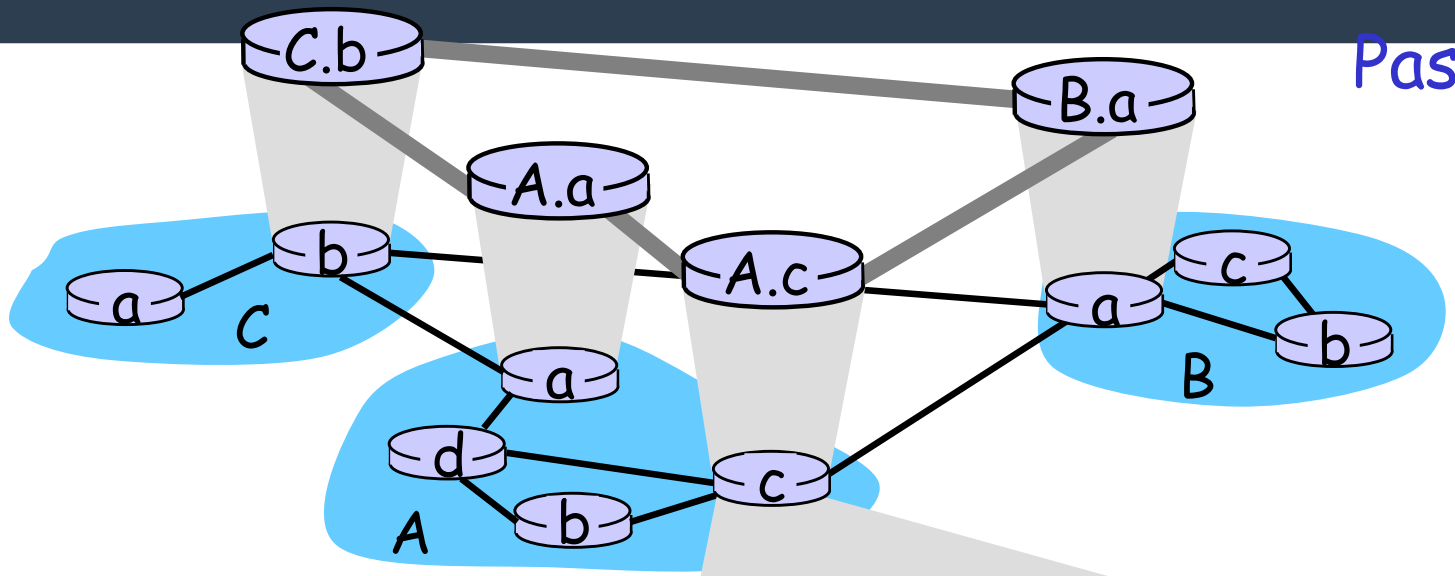
Routage hiérarchique

- **Les routeurs sont agrégés en , "systèmes autonomes" (AS pour Autonomous System en anglais)**
  - Un AS correspond à une entité, par exemple Orange, Free, Sprint
- **Routeurs d'un même AS utilisent le même protocole de routage en général**
  - → protocole de routage "intra-AS"
  - routeurs dans AS différents peuvent utiliser des protocoles intra-AS différents

Routeurs passerelles

- **Routeurs spéciaux en bordure de l'AS**
- **Parlent le protocole de routage interne à leur AS**
- **Parlent aussi un protocole de routage dit inter-AS → routage au niveau des AS (graphe des AS et non plus des routeurs)**
- **Ce protocole inter-AS s'appelle BGP (Border Gateway Protocol)**
- **Le même pour tous les AS!!!**

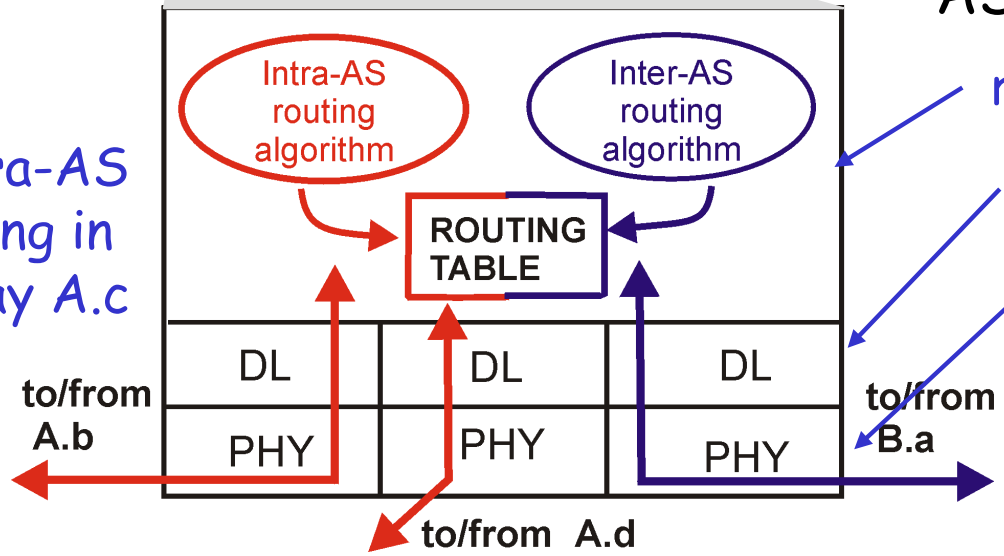
Routage intra-AS et inter-AS



Passerelles:

- font tourner le protocole inter-AS entre elles
- Font tourner le protocole intra-AS avec les routeurs de leur AS

inter-AS, intra-AS routing in gateway A.c

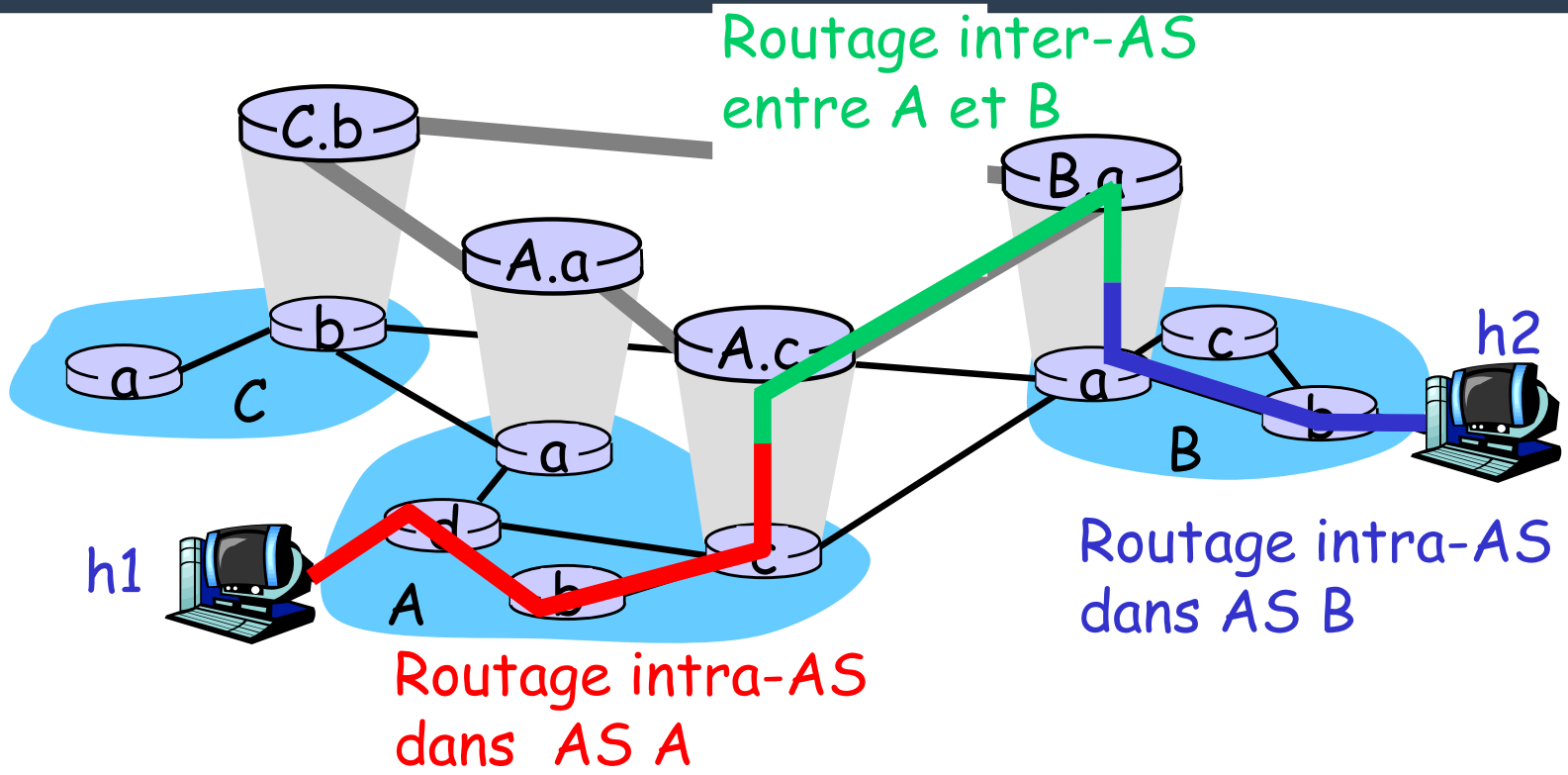


réseau

Liaison de données

Couche physique

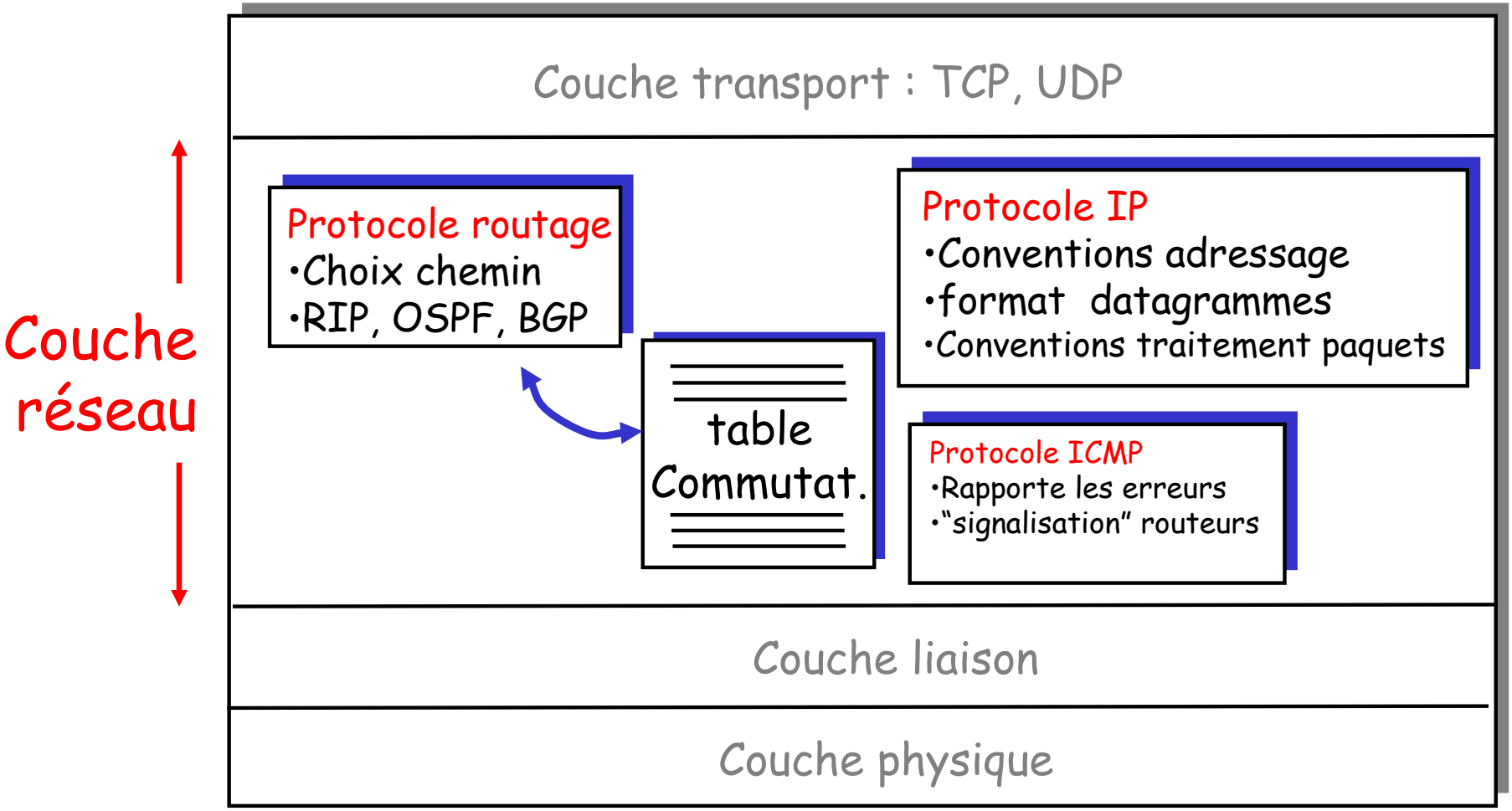
Routage intra-AS et inter-AS



- Le routage intra-AS cherche à maximiser les performances
- Le routage inter-AS sert d'abord à refléter les relations commerciales entre AS. Performance dans un second temps.

La couche réseau de l'Internet

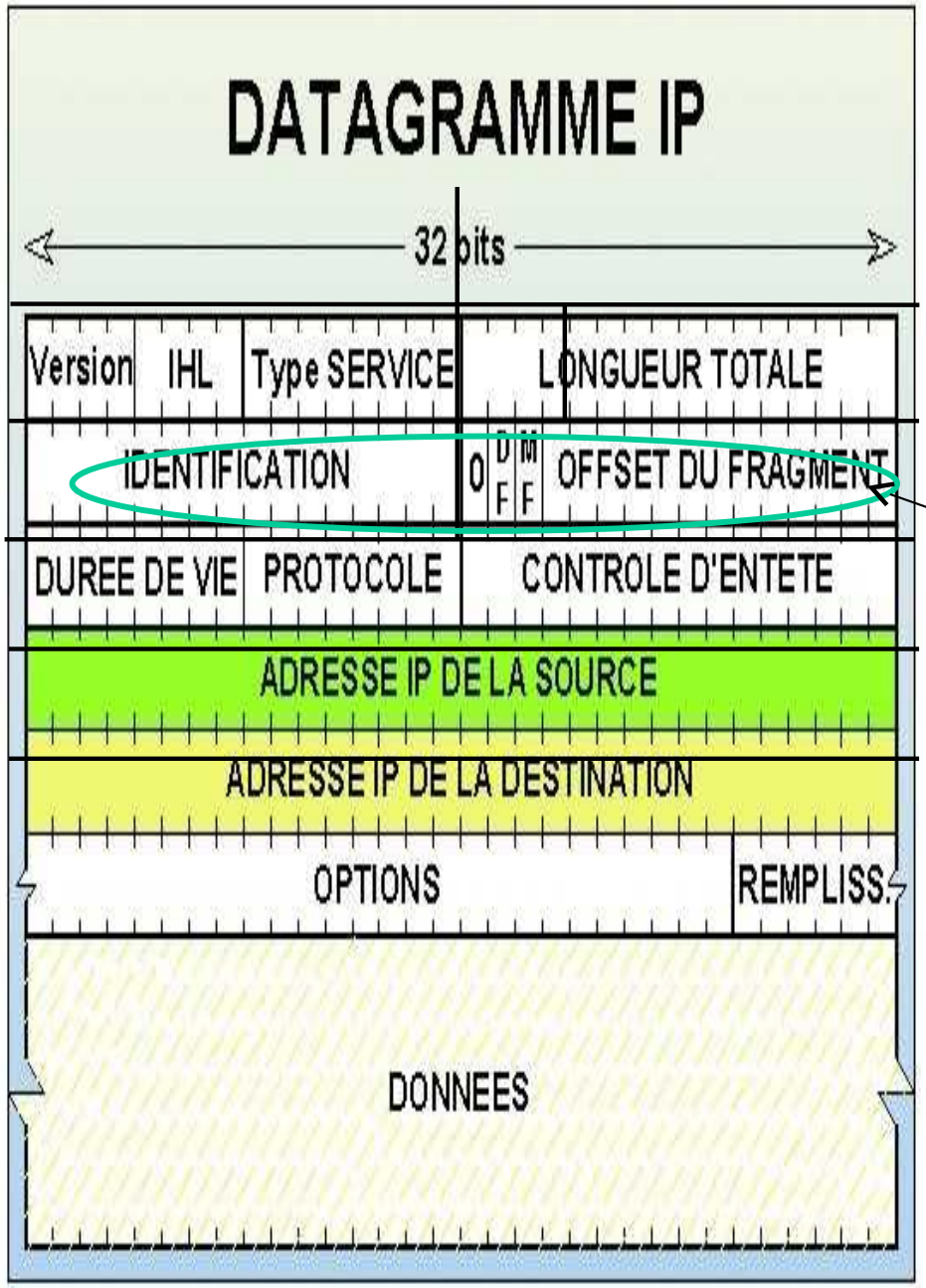
Les machines et les routeurs implantent cette couche!



Format datagramme IP

Combien d'octets de contrôle?

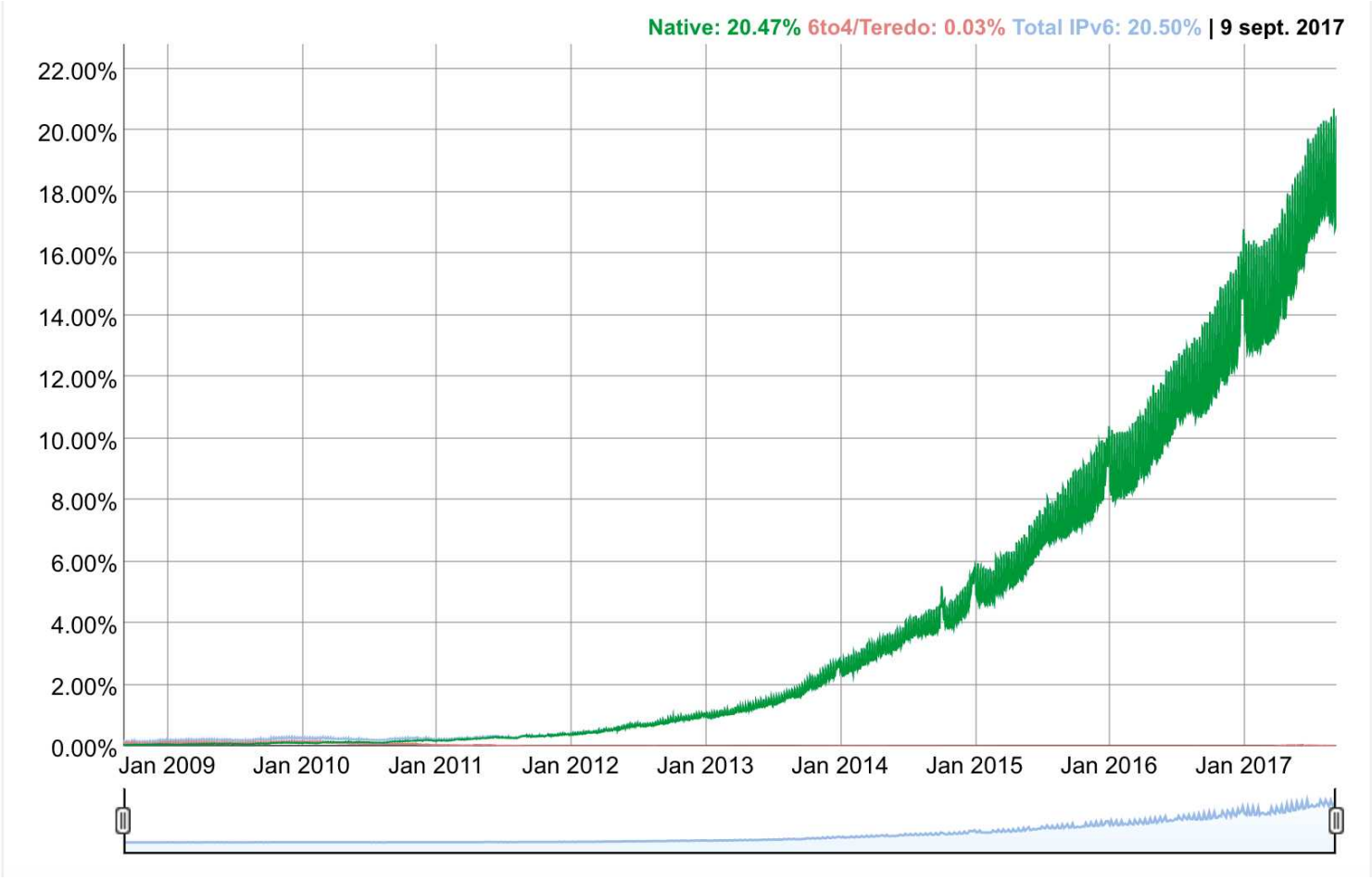
- 20 octets pour TCP ou 8 pour UDP
- 20 bytes of IP
- = 28 pour UDP ou 40 pour TCP + en-tête applicative (ex: HTTP)
- Il faut ajouter couche liaison (Ethernet, Wifi, etc



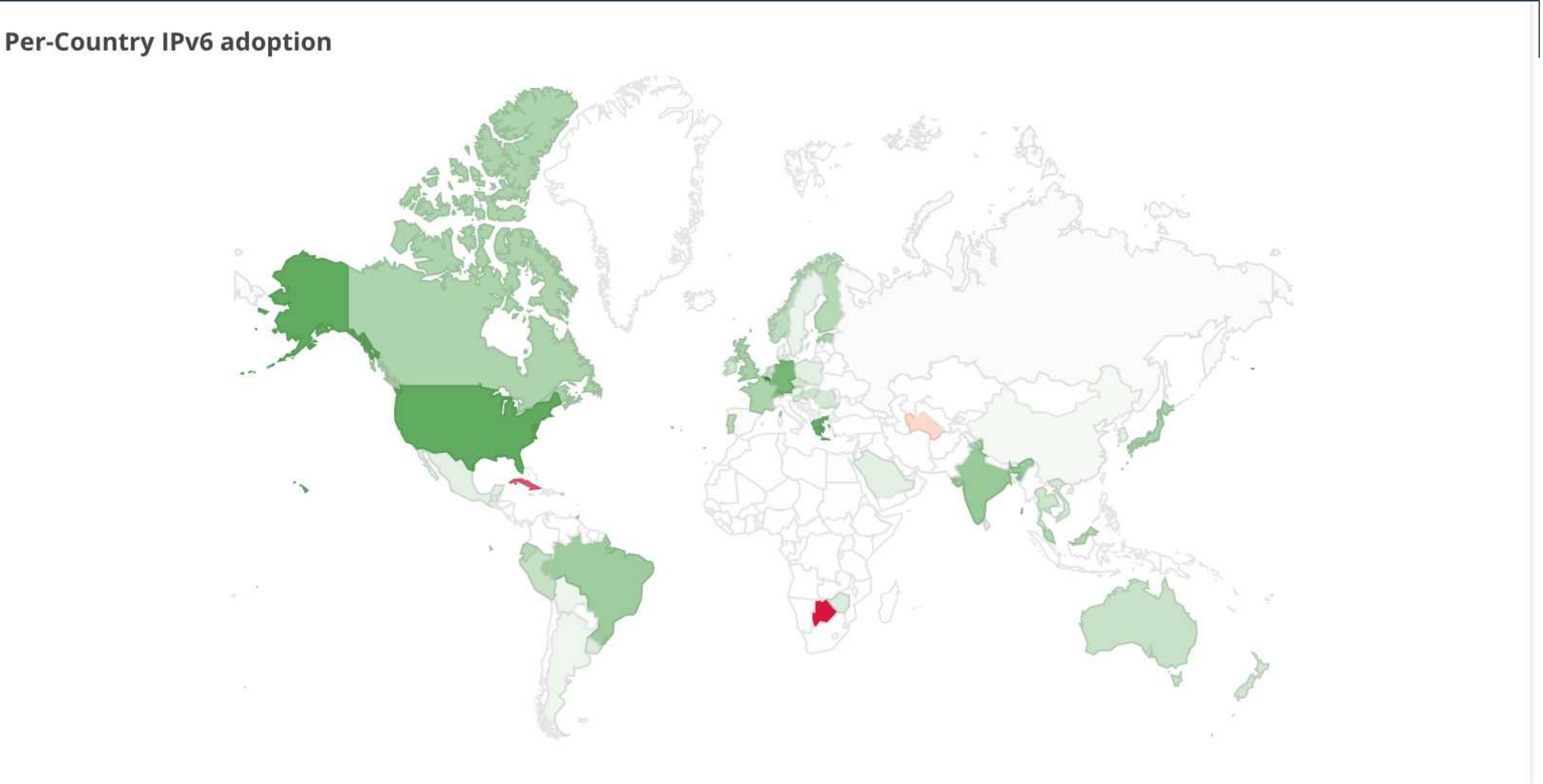
3 champs pour la fragmentation (vu dans M1101)

Format datagramme IP

- **Version : 4 ou 6**
  - majoritairement IPv4 dans le monde
  - Google offre des statistiques en continu : https://www.google.com/intl/en/ipv6/statistics.html



IPv6 : adoption par pays



Format datagramme IP

- IHL (Internet Header Lenght) : taille de l'en-tête IP comptée en mots de 32 bits → multiplé par 4 pour avoir en octets

- Type of service (ToS) : rarement utilisé. Peut permettre d'indiquer des données prioritaires

- Longueur totale : en octets!

Format datagramme IP

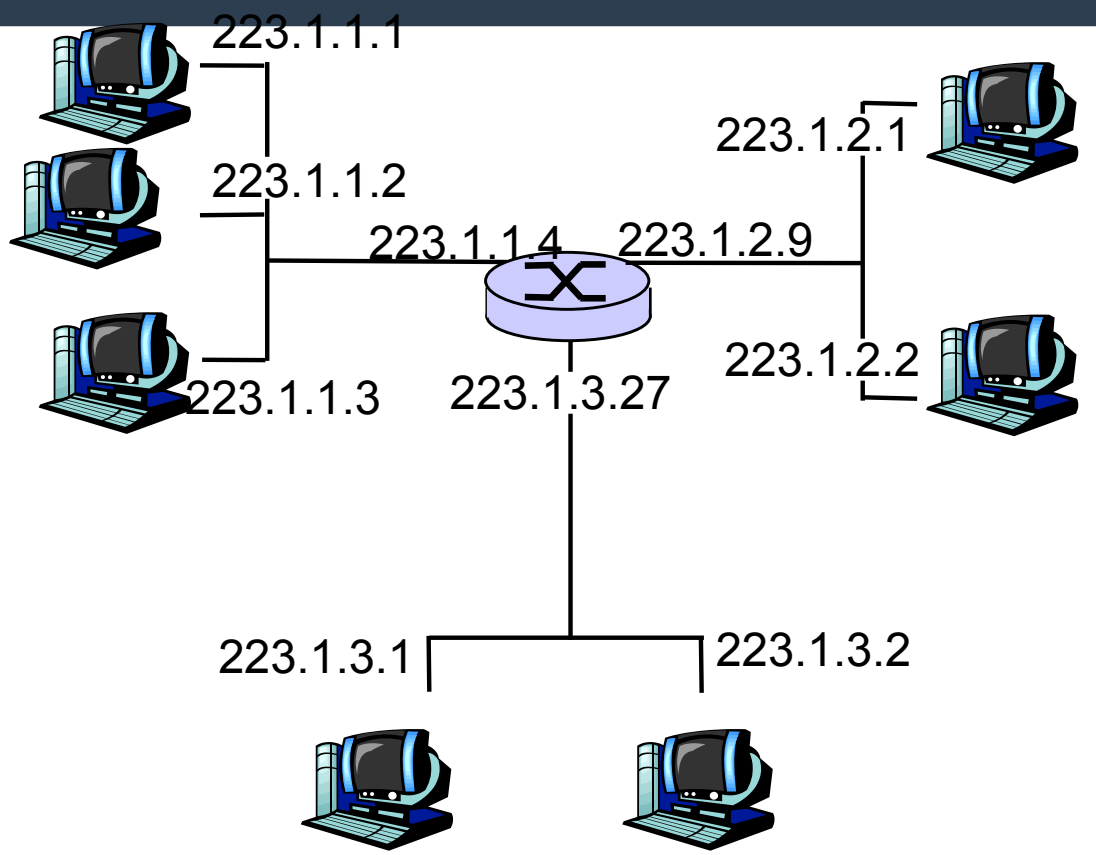
- **Durée de vie (TTL : Time To Live): pour qu'un paquet ne tourne pas infiniment dans le réseau, il a une durée de vie en nombre de routeurs qu'il traverse**
  - Généralement, TTL= 128 ou 256 à la source
  - Décrémenté à chaque routeur
  - Si TTL tombe à 0, le routeur le détruit et.. est censé émettre un message à la source (c'est ce qu'utilise traceroute!!!)

- **Protocole : permet de savoir à quelle couche transport délivrer le paquet à la destination : TCP, UDP, protocoles de routage (IGRP, EIGRP, OSPF), données chiffrées (ESP,AH), etc.**
  - TCP : 6
  - UDP : 11
  - EIGRP : 88

Adressage IP : introduction

- **Adresses IP: 32 bits, identifie une machine ou l'interface d'un routeur**

- ***interface:* connexion entre un routeur et un lien physique**
  - Les routeurs ont de multiples interfaces
  - Les machine aussi (par ex : filaire et sans fil) mais elles ne les utilisent pas simultanément
  - Une adresse IP est associée à chaque interface



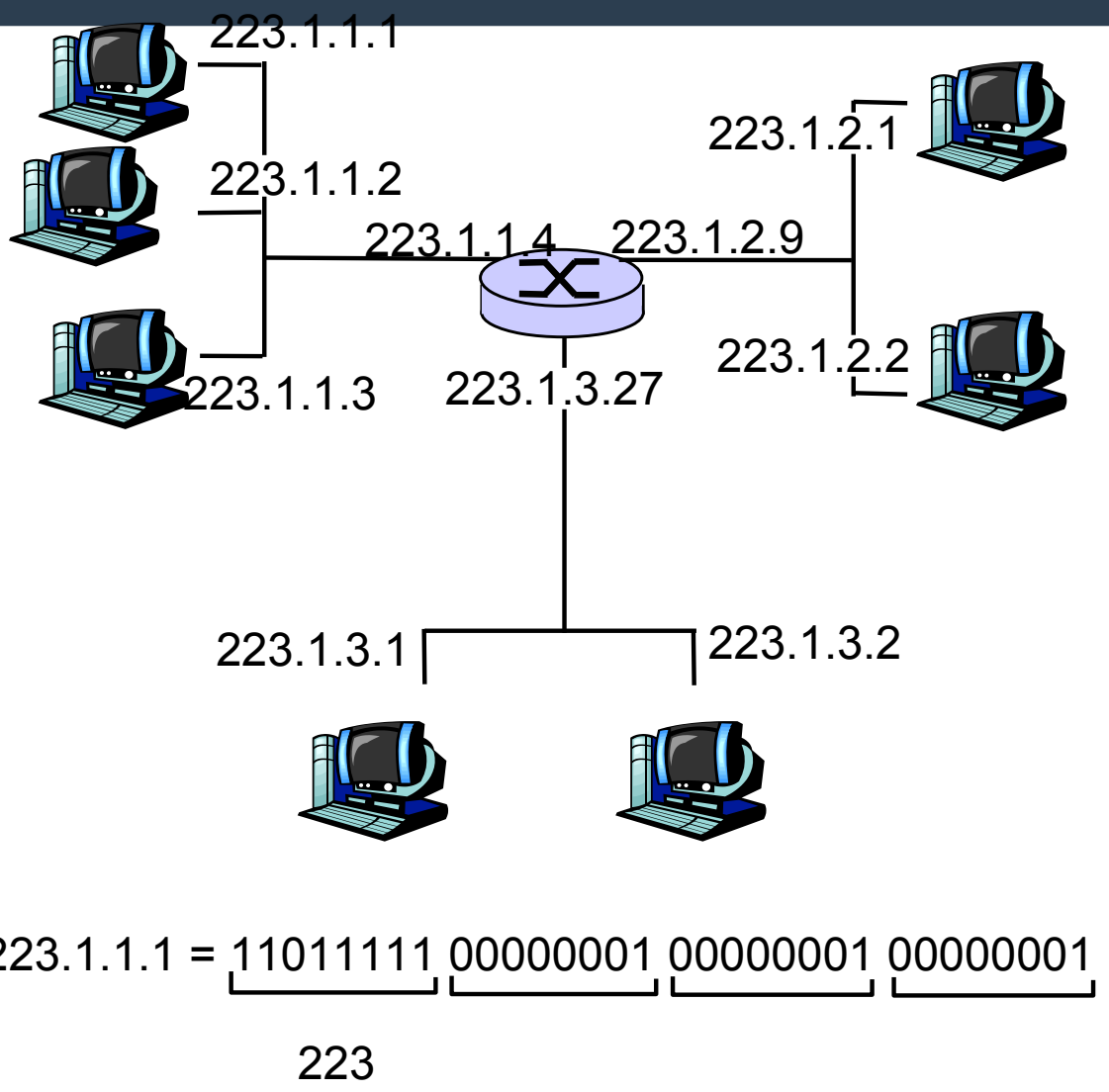
223.1.1.1 = 11011111 00000001 00000001 00000001

223

Adressage IP : introduction

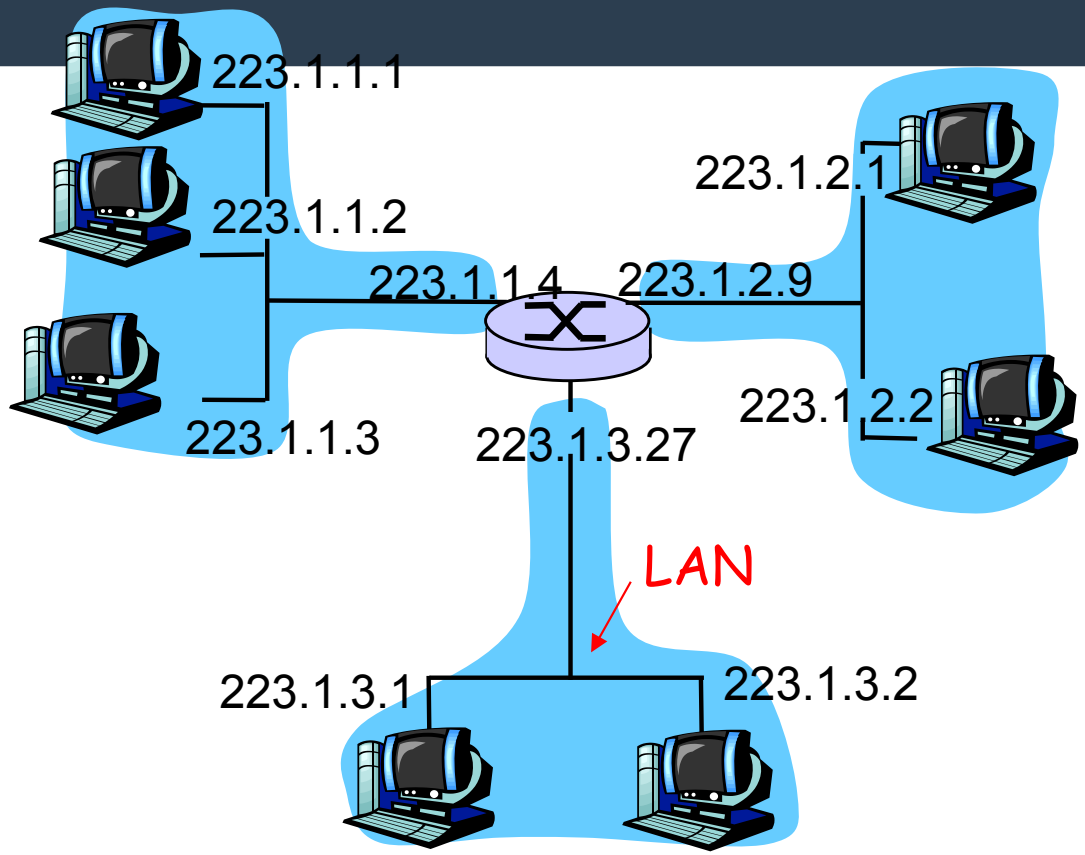
- **Adresses IP: 32 bits, identifie une machine ou l'interface d'un routeur**

- ***interface:* connexion entre un routeur et un lien physique**
  - Les routeurs ont de multiples interfaces
  - Les machine aussi (par ex : filaire et sans fil) mais elles ne les utilisent pas simultanément
  - Une adresse IP est associée à chaque interface



Adressage IP

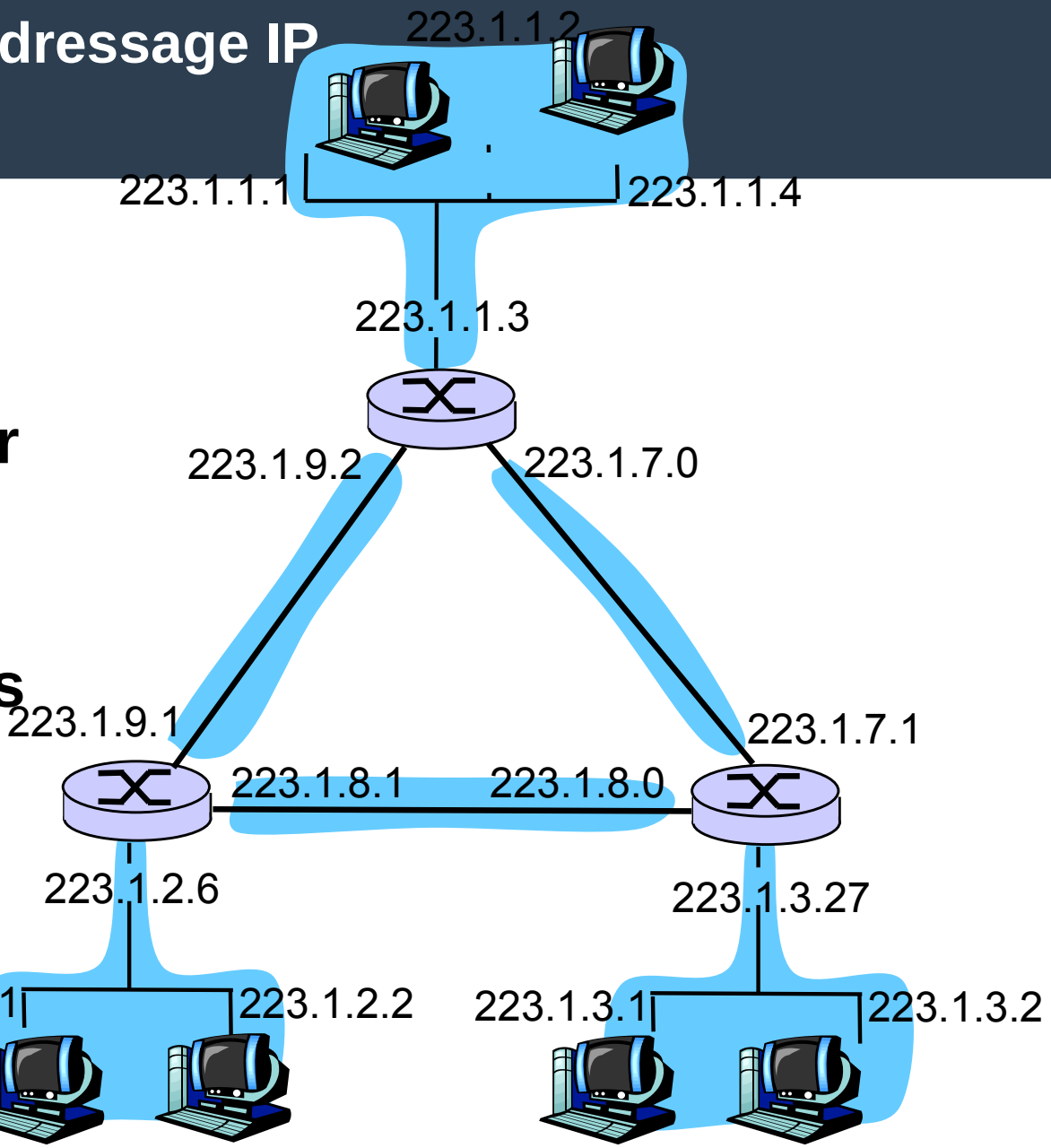
- **Une adresse IP se découpe en :**
  - Une partie réseau (bits de poids fort → à gauche!)
  - Une partie machine (bits de poids faibles)
- ***Qu'est ce qu'un réseau ?*** **(de point de vue de l'adressage IP!)**
  - interfaces des équipements avec même partie réseau dans leur adresse
  - Ces machines peuvent se joindre physiquement SANS un routeur
  - On parle de réseau local ou LAN en anglais



Réseau global constitué de 3 réseaux IP (pour les adresses IP commençant par 223 24 bits pour le réseau et 8 pour la machine)
→ 223.1.1 223.1.2 et 223.1.3

Comment compter les réseaux IP?

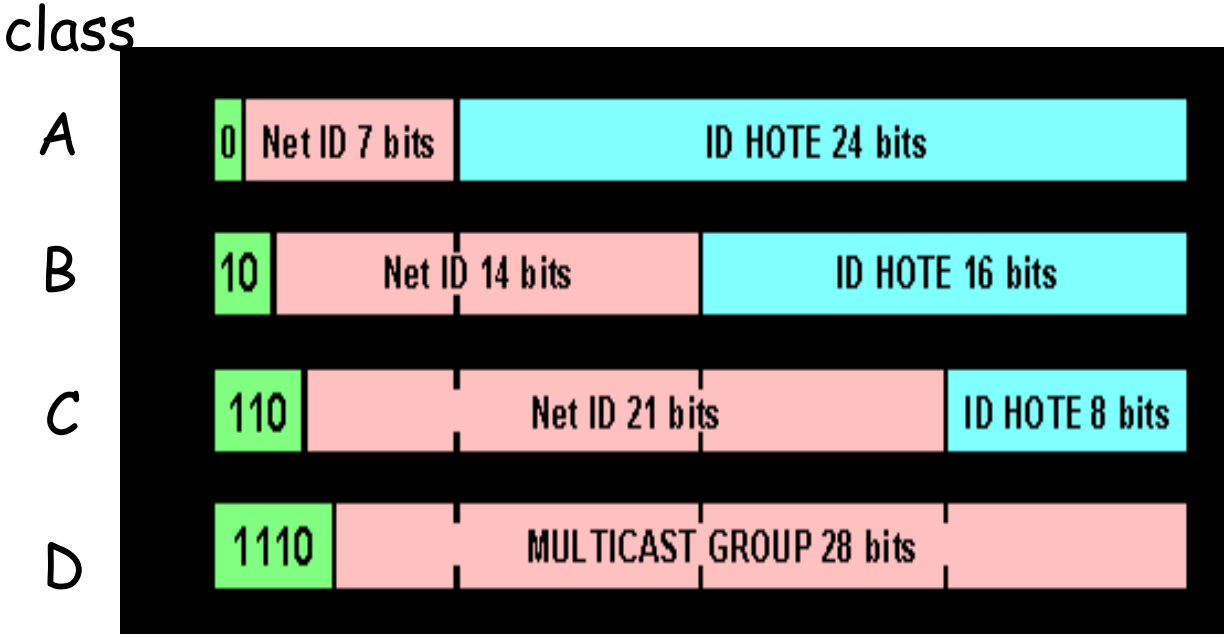
- **Detacher chaque interface du routeur ou de la machine**
- **Cela créer des ilots de réseaux IP isolés**



Combien de réseaux IP ici?

Combien de bits pour le réseau et pour la machine?

Historique et plus utilisé : Classes d'adressses du système dit "class-full":



1.0.0.0 à
127.255.255.255

128.0.0.0 à
191.255.255.255

192.0.0.0 à
223.255.255.255

224.0.0.0 à
239.255.255.255

Une adresse spécifique : 255.255.255.255: adresse de diffusion (=broadcast)
Pour joindre toutes les machines (du monde? Non locales – les routeurs bloquent)

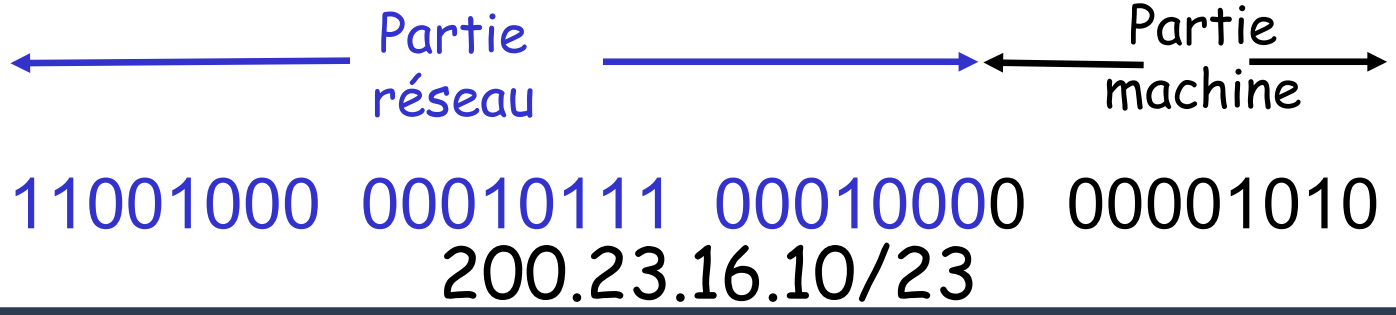
Adressage IP: CIDR

- **Adressage par classes:**
  - Utilisation inefficace de l'espace des adresses → on serait arriver rapidement au manque d'@
  - Pourquoi inefficace? Si vous avez 2000 machines qui doivent accéder à Internet, il faudrait vous donnez une adresse de class B qui correspond à 2^{16}= 65536 machines…

- **CIDR: Classless InterDomain Routing**

Class-less

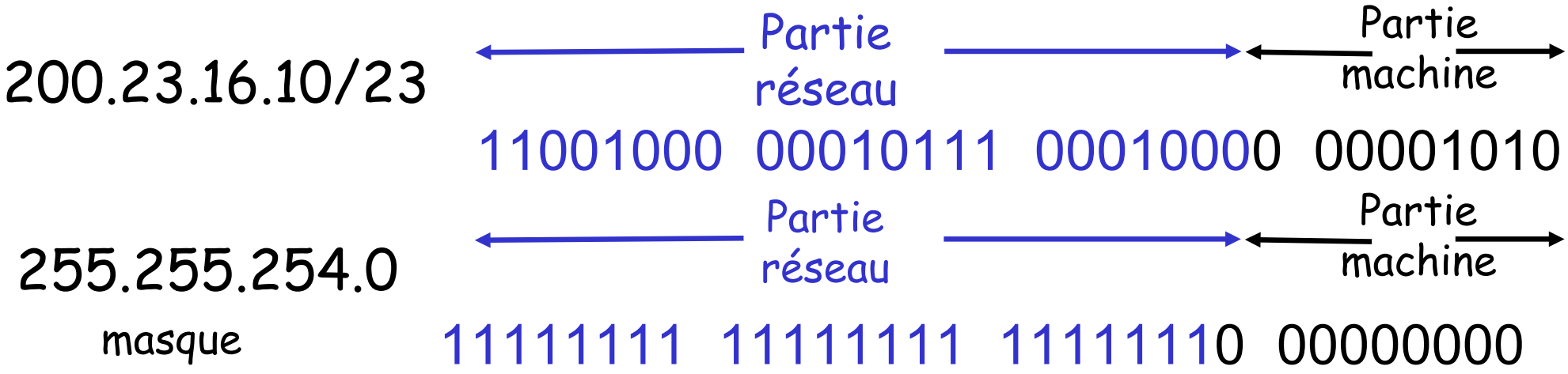
  - La portion réseau de l'adresse est maintenant de taille variable
  - Format d'adresse: a.b.c.d/x, où x est le nombre de bits de la portion réseau de l'adresse



Adresse IP: adresse d'un réseau

- **Adresse réseau obtenue en mettant à 0 tous les bits de la partie machine**
- **Ne peut être assignée à une machine**
  - Ex : 200.23.16.10/23 transparent précédent

- **Masque de sous réseau : des '1' dans la partie réseau et des '0' dans la partie machine :**



Exemple

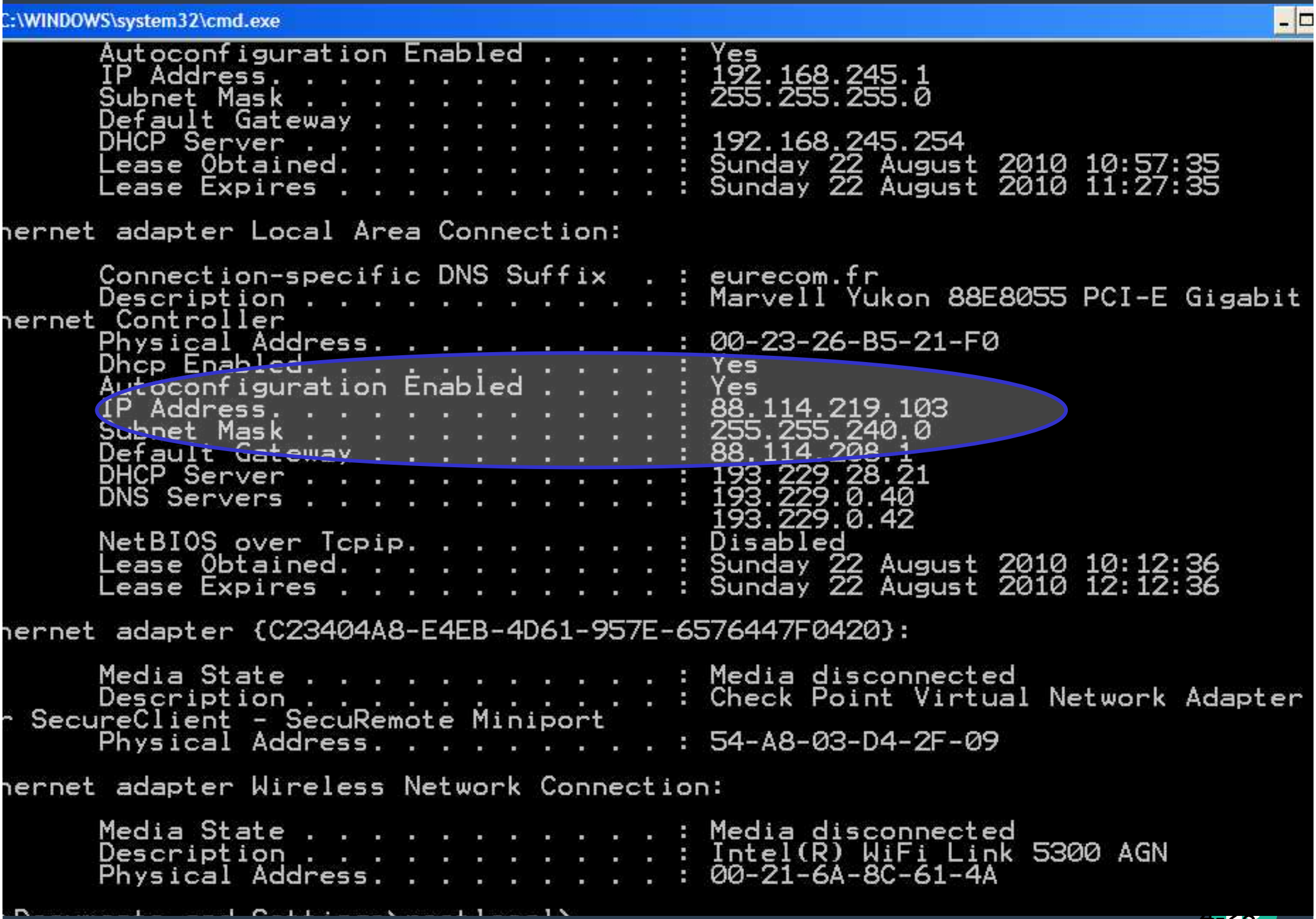
Assigner 200.23.16.10/23 est équivalent à assigner 200.23.16.10 et masque 255.255.254.0

Adresse réseau sur lequel est la machine : 200.23.16.0/23

Adresse de broadcast :200.23.17.255

Ces 2 adresses ne peuvent être données à des machines

Les adresses assignables vont de 200.23.15.1 à 200.23.16.254


C:\WINDOWS\system32\cmd.exe
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . : 192.168.245.1
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . :
DHCP Server . . . . . : 192.168.245.254
Lease Obtained. . . . . : Sunday 22 August 2010 10:57:35
Lease Expires . . . . . : Sunday 22 August 2010 11:27:35
ernet adapter Local Area Connection:
Connection-specific DNS Suffix . : eurecom.fr
Description . . . . . : Marvell Yukon 88E8055 PCI-E Gigabit
ernet Controller
Physical Address. . . . . : 00-23-26-B5-21-F0
Dhcp Enabled. . . . . : Yes
Autoconfiguration Enabled . . . . : Yes
IP Address. . . . . : 88.114.219.103
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . : 88.114.208.1
DHCP Server . . . . . : 193.229.28.21
DNS Servers . . . . . : 193.229.0.40
193.229.0.42
NetBIOS over Tcpip. . . . . : Disabled
Lease Obtained. . . . . : Sunday 22 August 2010 10:12:36
Lease Expires . . . . . : Sunday 22 August 2010 12:12:36
ernet adapter {C23404A8-E4EB-4D61-957E-6576447F0420}:
Media State . . . . . : Media disconnected
Description . . . . . : Check Point Virtual Network Adapter
r SecureClient - SecuRemote Miniport
Physical Address. . . . . : 54-A8-03-D4-2F-09
ernet adapter Wireless Network Connection:
Media State . . . . . : Media disconnected
Description . . . . . : Intel(R) WiFi Link 5300 AGN
Physical Address. . . . . : 00-21-6A-8C-61-4A

Adresses IP privées

- **Des adresses IPs qu'on peut utiliser pour gérer son réseau privé pour les particuliers/entreprises.**
- **3 classes d'adresse :**
  - 10.0.0.0/8
  - 172.15.0.0/12
  - 192.168.0.0/16
- **Les routeurs de l'Internet détruisent les paquets avec des adresses IP privées.**
  - Contrairement aux routeurs d'entreprise qui si ils sont configurés avec des adresses privées acceptent

Adresse IP privées

- **Exemple simple d'architecturation.**
- **Vous avez quatre groupes de travail : développeur, finance, direction**
- **Il faut ajouter les serveurs (tous dans le même réseau IP)**
- **Vous pouvez architecturer ainsi :**
  - 172.15.0.0/16 : développeur
  - 172.16.0.0/16 : finance
  - 172.17.0.0/16 : direction
  - 172.18.0.0/16 : serveur

  Le tout avec un routeur central qui permet de passer entre les réseaux
- **Pour sortir de l'entreprise (ou passez au travers votre box à la maison) il faut faire du NAT : Traduction d'adresse réseau que vous verrez dans d'autre cours**
  - Idée de base avec une box : la box a une adresse sur son interface de sortie ADSL/Fibre que les différentes machines se partagent

Une adresse IP : comment l'obtenir?

- **Entrée manuellement par l'administrateur système**
  - On le fait pour les serveurs (Web, mail, etc) dont les adresses sont fixes

- **DHCP: Dynamic Host Configuration Protocol: la machine demande dynamiquement (à chaque fois qu'elle démarre) une adresse à un serveur spécifique, le serveur DHCP**
  - Pour les machines clients

DHCP: Dynamic Host Configuration Protocol

Objectif: obtenir dynamiquement une adresse quand une machine joint un nouveau réseau

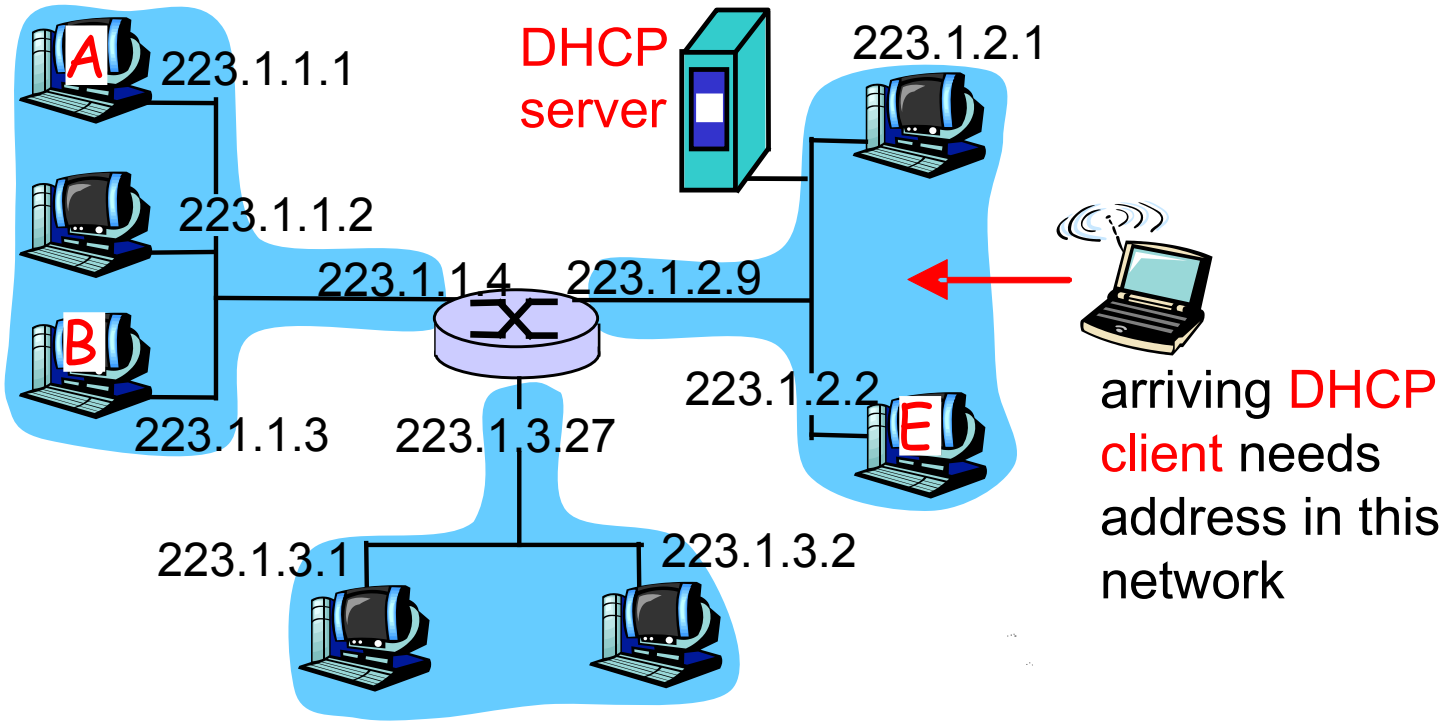
Adresses données pour un certain temps seulement

Intérêt : si une machine disparait, l'adresse peut-être ré-assignée rapidemment

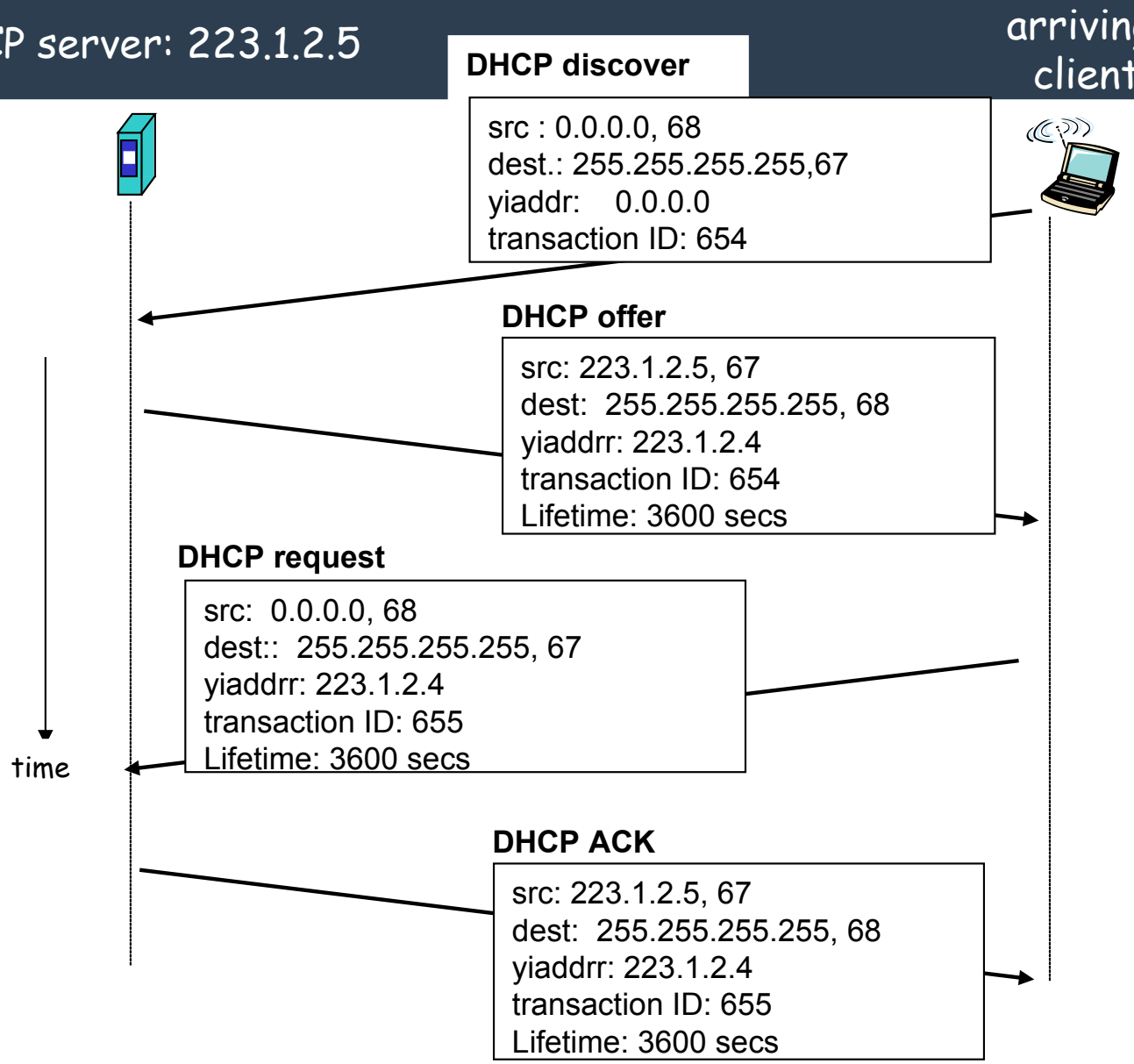
Protocole DHCP :

- Machine émet un message (msg) broadcast "DHCP discover"
- Serveur DHCP répond avec un msg "DHCP offer"
- Hôte demande une adresse IP via msg "DHCP request"
- Serveur DHCP envoie adresse: "DHCP ack"

DHCP scenario client-server



DHCP scenario client-server

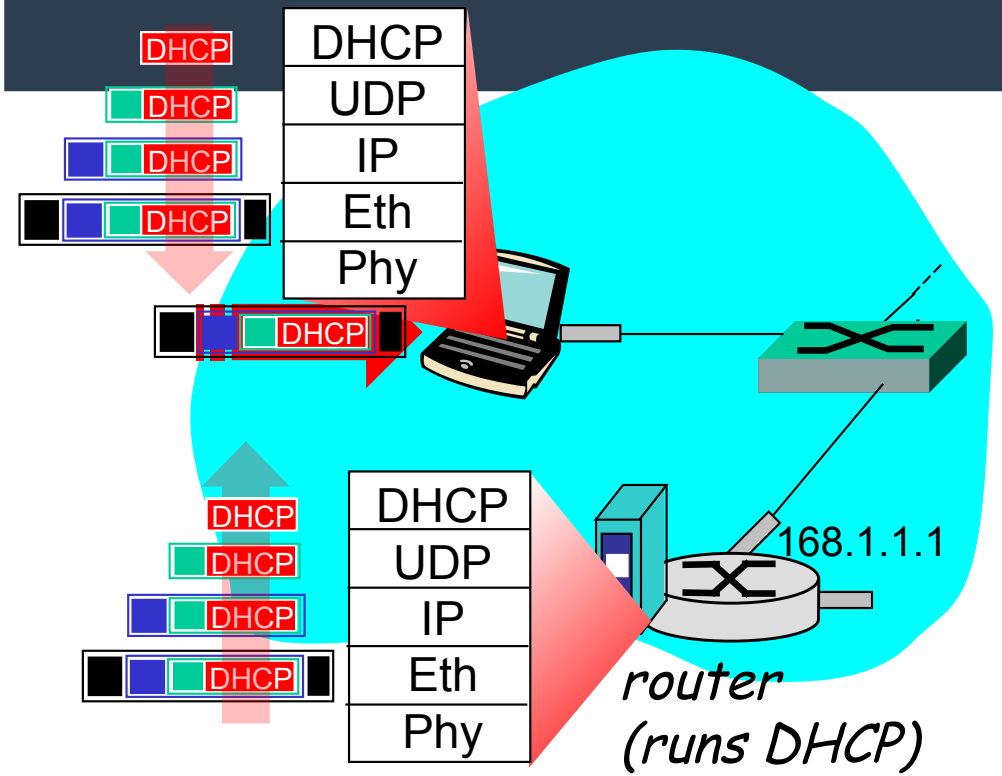


DHCP: more than IP address

DHCP retourne en plus de l'adresse IP :

- La passerelle (premier routeur)
- Nom et IP serveur DNS
- Masque réseau
- (et même d'autres choses comme dans le cas d'un boot réseau...)

DHCP: exemple



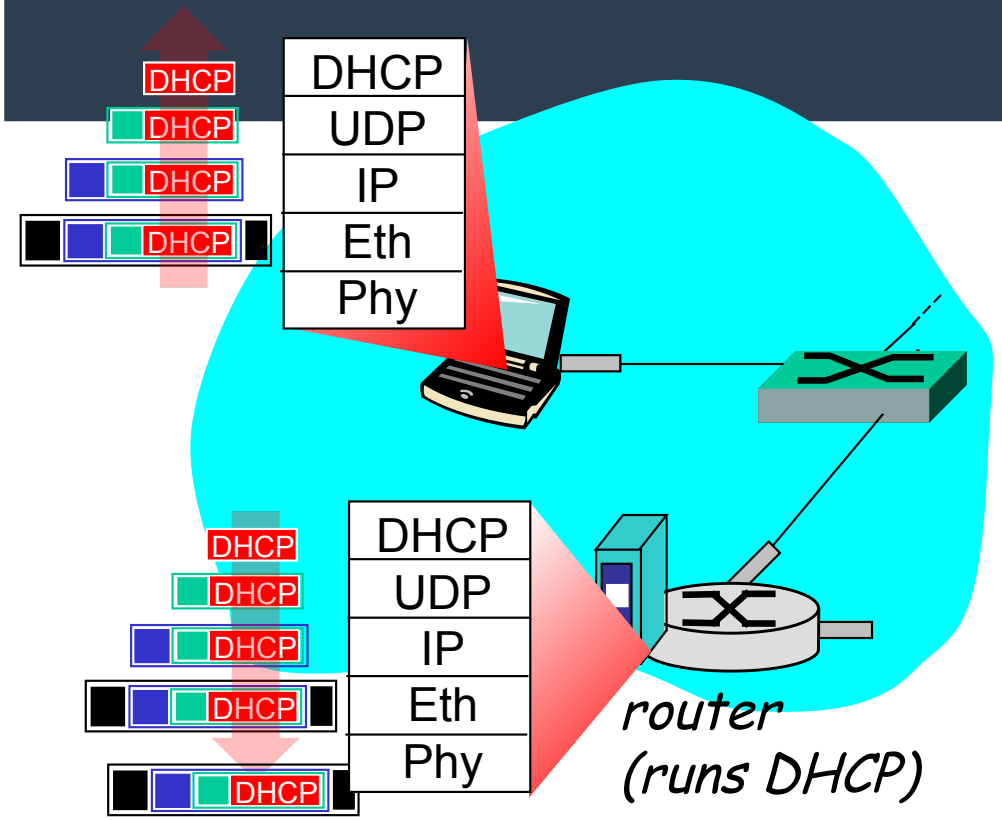
Machine a besoin d'une adresse IP + masque+passerelle+ serveur DNS → DHCP

RequêteDHCP encapsulée en UDP, encapsulée in IP, encapsulée in 802.1 Ethernet

Trame Ethernet broadcast (dest: FFFFFFFFFFFF) sur le LAN, reçu pa routeur qui fait serveur DHCP (peut être un serveur indépendant) router running DHCP server

Ethernet démultiplexée vers IP puis UDP puis DHCP

DHCP: exemple



- Serveur DCP construit DHCP ACK avec IP client,passerelle et serveurs DNS

- encapsulation DHCP server, trame envoyée au client et démultiplexée au client
- client connait maintenant toutes les infos

DHCP: wireshark

request

Message type: **Boot Request (1)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) **DHCP Message Type = DHCP Request**
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
    ……

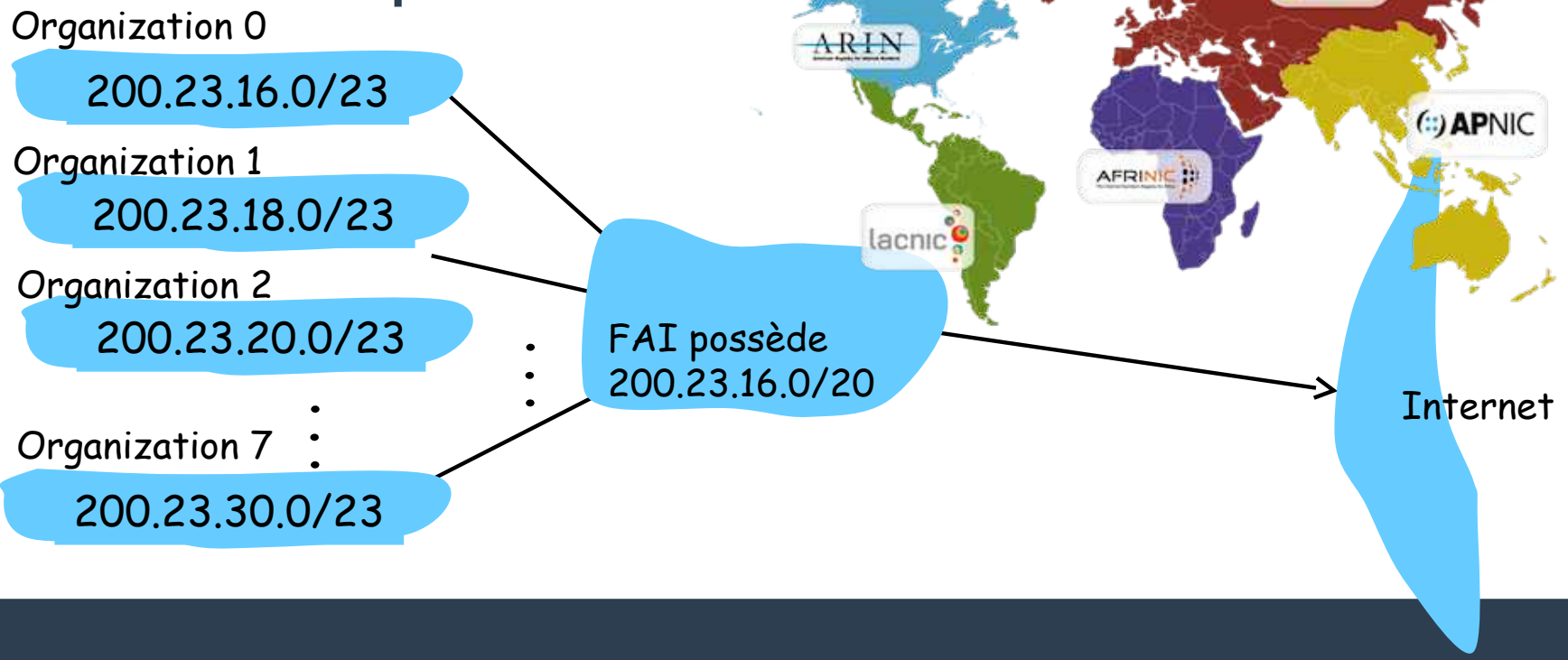
reply

Message type: **Boot Reply (2)**
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."

Un réseau IP, comment l'obtenir?

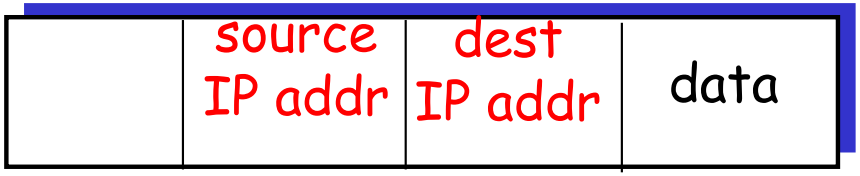
- **Une organisation/entreprise l'obtient de son fournisseur d'accès Internet**
  - Subdivision avec CIDR (on ne respecte pas les classes)

- **Celui-ci a un ensemble d'adresses obtenu de l'IANA (Internet Assigned Numbers Authority) ou plus vraisemblablement d'un représentant local, par exemple RIPE en Europe**



Transfert d'un datagramme de sa source à sa destination

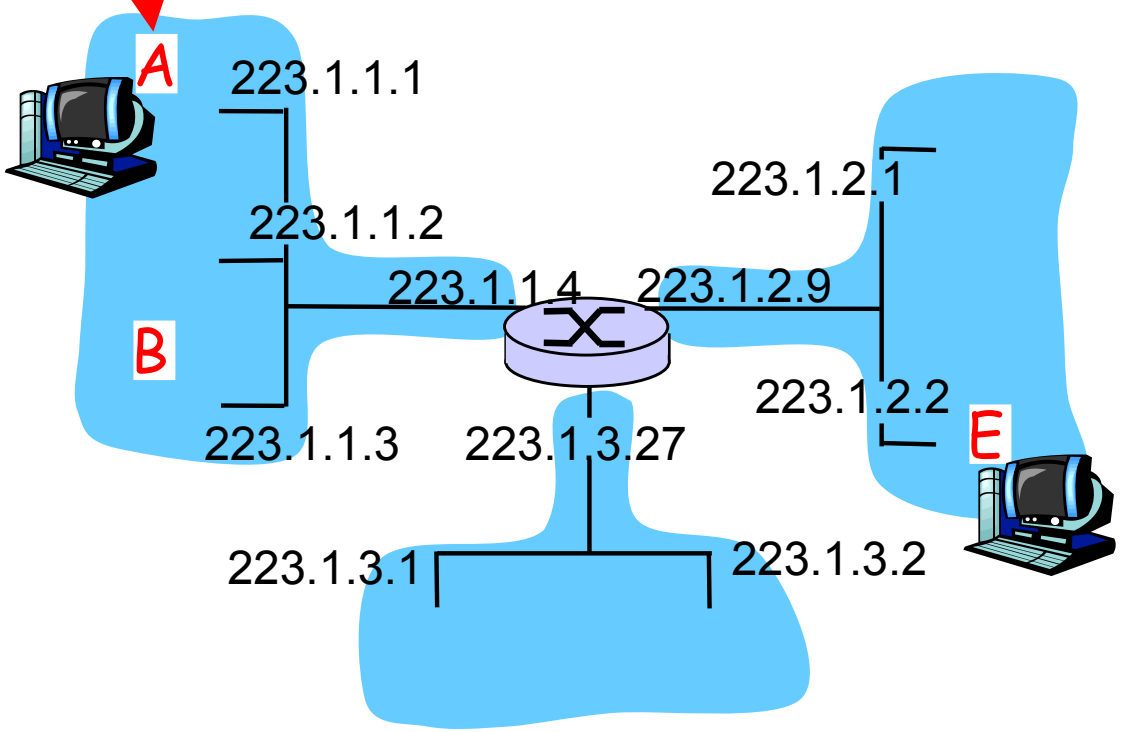
Datagramme IP:



- Aucune modification entre la source et la destination (sauf TTL décrémenté)
- Ce sont les @ qui nous intéressent ici!

table commutation A

Réseau Dest.	Prochain routeur	Nb sauts
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



Scénario 1: datagramme entre A et B

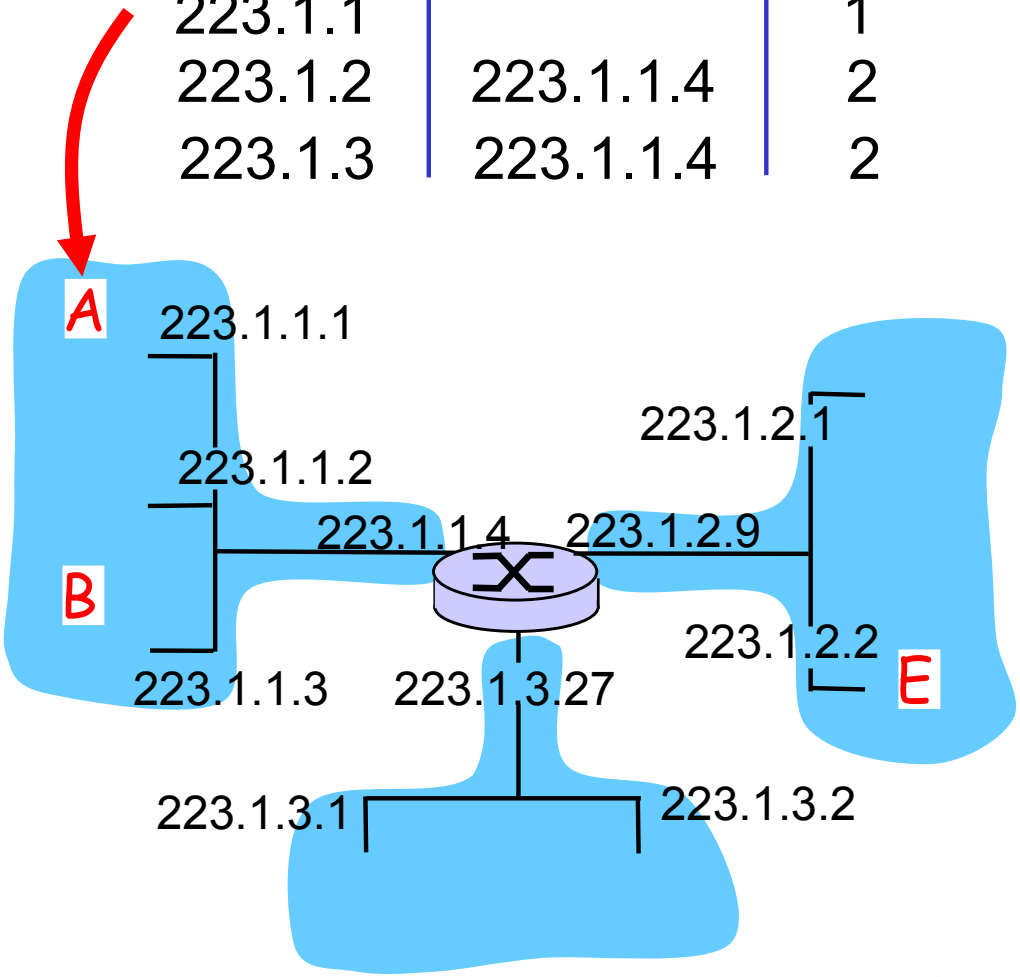
	223.1.1.1	223.1.1.3	data

table commutation A

Réseau Dest.	Prochain routeur	Nb sauts
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2

Opération de A

- Cherche adresse réseau de B dans table commutation
- Si B et A sont dans même sous-réseau, c'est la couche sous IP, la couche liaison qui prend le relai!

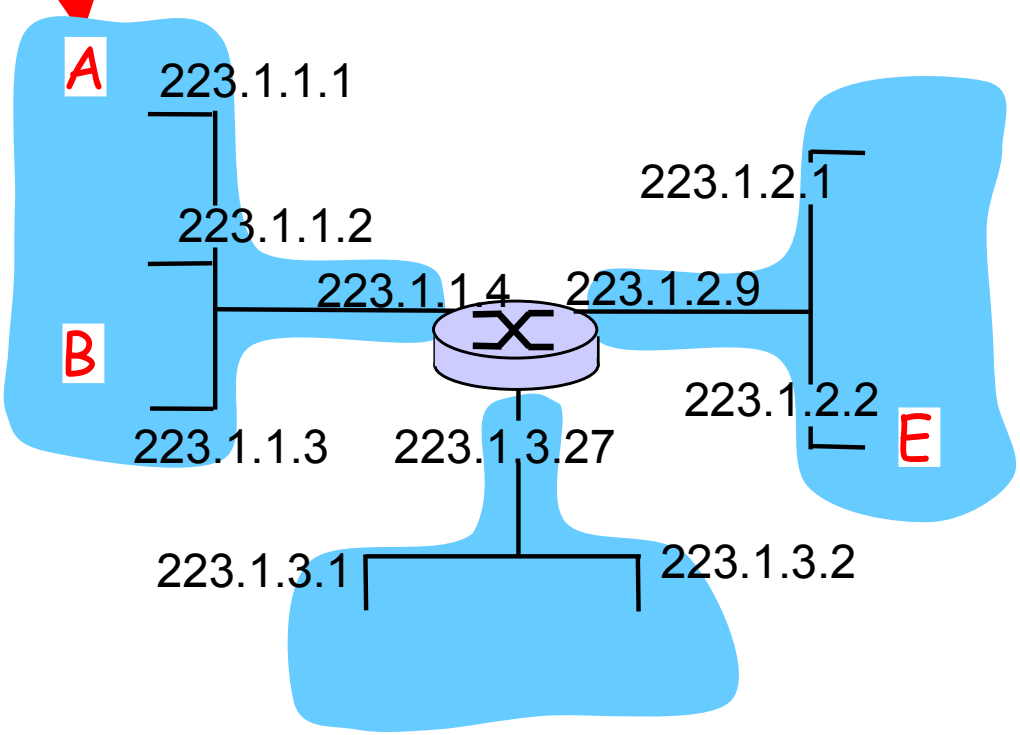


Scénario 2: datagramme entre A et E

misc fields	223.1.1.1	223.1.2.3	data

table commutation A

Réseau Dest.	Prochain routeur	Nb sauts
223.1.1		1
223.1.2	223.1.1.4	2
223.1.3	223.1.1.4	2



Opérations:

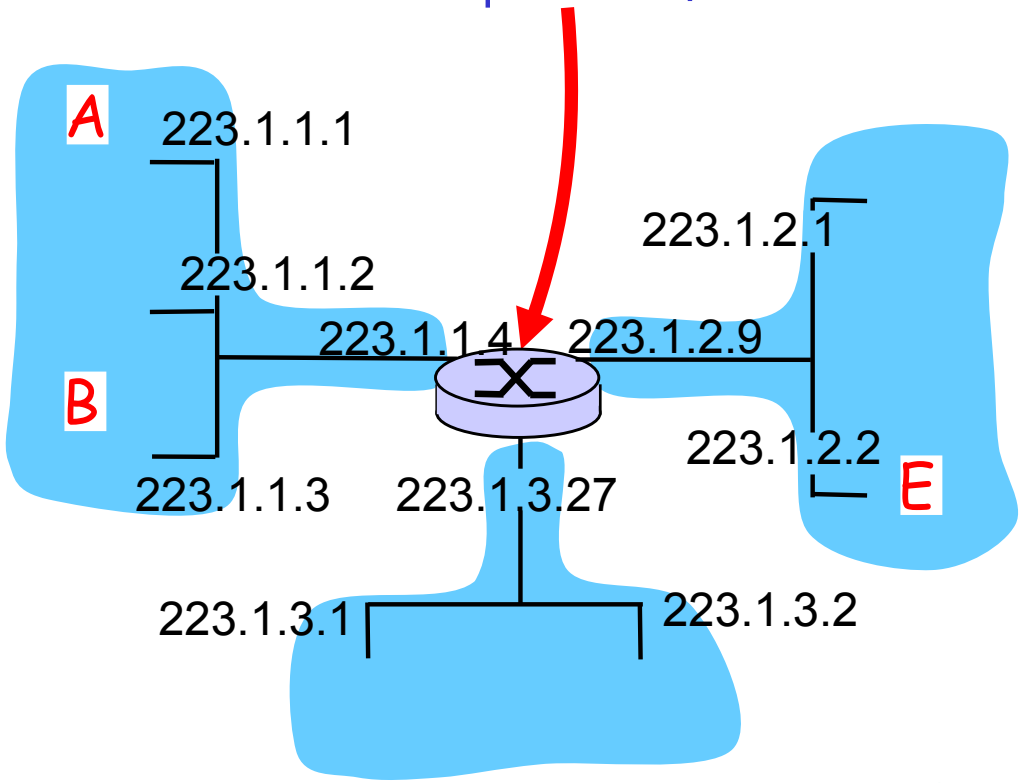
- A observe que E est dans un réseau IP différent!
- Table de routage indique que le prochain routeur est 223.1.1.4
- Couche liaison de donnée envoie datagramme de A à 223.1.1.4 (puisqu'il sont dans le même sous-réseau)
- datagramme arrive à 223.1.1.4
- On continue.....

Scénario 2: datagramme entre A et E

table commutation A

misc fields	223.1.1.1	223.1.2.3	data

Dest.	routeur	Sauts	interface
223.1.1	-	1	223.1.1.4
223.1.2	-	1	223.1.2.9
223.1.3	-	1	223.1.3.27



Arrivé à 223.1.4 et destiné à 223.1.2.2

- Le routeur observe que la destination est dans un sous-réseau correspondant à une de ces interfaces → il passe à la couche liaison sur cette interface
- Le datagramme va ainsi arriver à 223.1.2.2!!!

ICMP: Internet Control Message Protocol

- **utilisé par machines et routeurs pour communiquer des information de niveau réseau**
  - Report d'erreurs : destination (routeur, port, réseau…) injoignable
  - echo request/reply (utilisé par ping)
- **Msgs ICMP msgs transporté directemet au dessus d'IP (pas de couche TCP/UDP)**
- **Message ICMP : type, code plus les 8 premiers octets du datagramme ayant causé l'erreur**

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

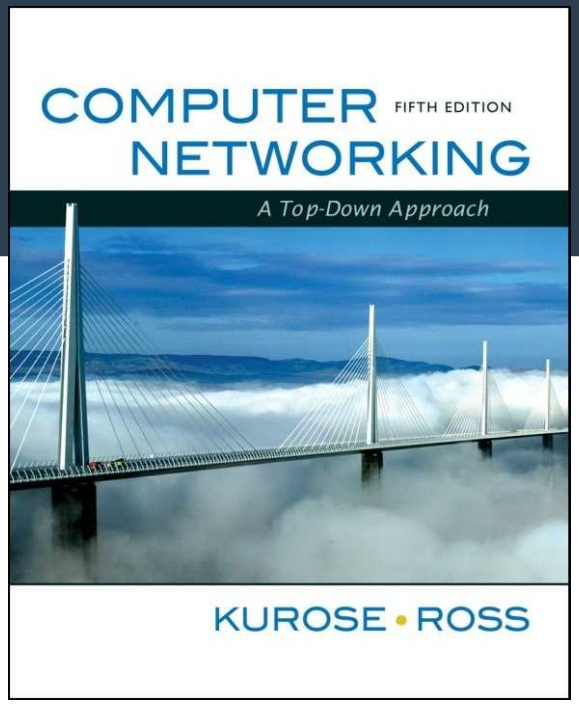
Traceroute et ICMP

- **Source envoie une série de segment UDP à la destination**
  - Premier TTL =1
  - Second TTL=2, etc.
  - Numéro de ports UDP réservé à traceroute
- **Quand le n-ième datagramme arrive au n-ième routeur :**
  - Router détruit datagramme car TTL =0 mais envoie messsage ICMP avec les premiers octets du message comme données
  - Message ICMP type 11, code 0
  - Message inclue IP routeur
- **Quand l'ICMP message arrive, source calcule le RTT**
- **Traceroute, par défaut envoie 3 paquet par destination → 3 ICMP**

Critère d'arrêt

- **Un segment UDP arrive à la destintion**
- **Un message ICMP de type "host unreachable" (type 3, code 3) est reçu par source**
- **Au bout de 30 sauts (TTL=30) si aucune réponse**

Couche liaison de données et réseaux locaux (LAN)



A note on the use of these ppt slides:

We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) in substantially unaltered form, that you mention their source (after all, we'd like people to use our book!)
- If you post any slides in substantially unaltered form on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2009
J.F Kurose and K.W. Ross, All Rights Reserved

Computer Networking: A Top Down Approach 5th edition.
Jim Kurose, Keith Ross
Addison-Wesley, April 2009.

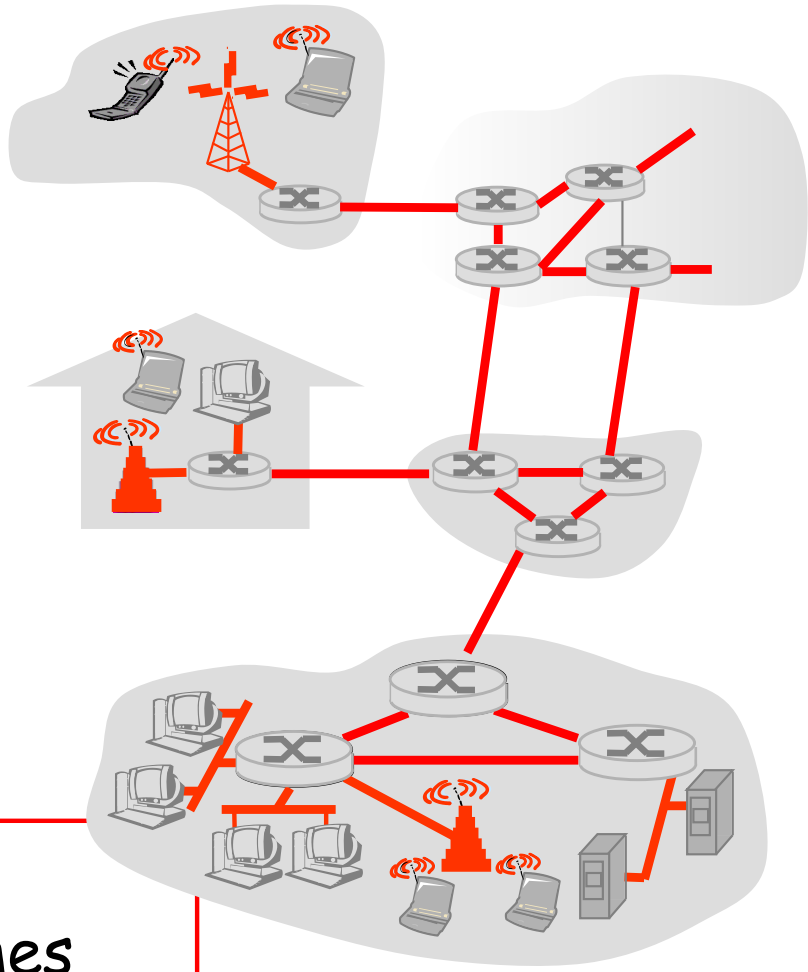
Agenda

- **Introduction et services**
- **Protocoles pour accès multiples**
- **Adressage couche liaison et ARP**
- **Commutateurs**

Couche liaison de données : Introduction

Terminologie :

- **Les machines et les routeurs sont des noeuds**
- **Les canaux de communication qui connectent les nœud adjacents le long d'un chemin sont appelés liens**
  - Liens filaires
  - Liens sans fils
- **Une paquet de niveau 2 est appelé une trame (et encapsule un datagramme)**



La couche liaison de données
est chargée de transférer des trames d'un noeud à un autre noeud adjacent sur un lien

Couche liaison de données : contexte

- **Les datagrammes sont transférés par différents protocoles de niveau liaison de données sur différents liens :**
  - Ex: Ethernet sur premier lien, frame relay sur les liens intermédiaires et 802.11 (wifi) sur le dernier
- **Chaque lien offre un service différent**
  - Ex: peut ou non fournir un transfert de données fiable

Analogie dans le domaine du transport

- **touriste = datagramme**
- **Étape du voyage= lien de communication**
- **Mode de transport= protocole de liaison de données**
- **Agent de voyage = algorithme de routage**

Services de la couche liaison de données

- ***Tramage (mise en trame), accès au lien :***
  - Les datagrammes sont encapsulés dans des trames avec ajout d'un en-tête et d'une queue (des données de contrôle à la suite du datagramme)
  - Accès au médium si le médium est partagé (par exemple en 802.11)
  - Les adresses "MAC" (adresses de la couche liaison de données) des trames identifient la source et la destination pour chaque lien traversé
    - differentes des addresses IP (qui identifient les extrémités)!
- ***Half-duplex ou full-duplex***
  - Half-duplex: une des 2 extremités seulement parlent à un instant donné (ex : wifi)
  - Full-duplex : les 2 peuvent parler simultanément (ex : adsl)

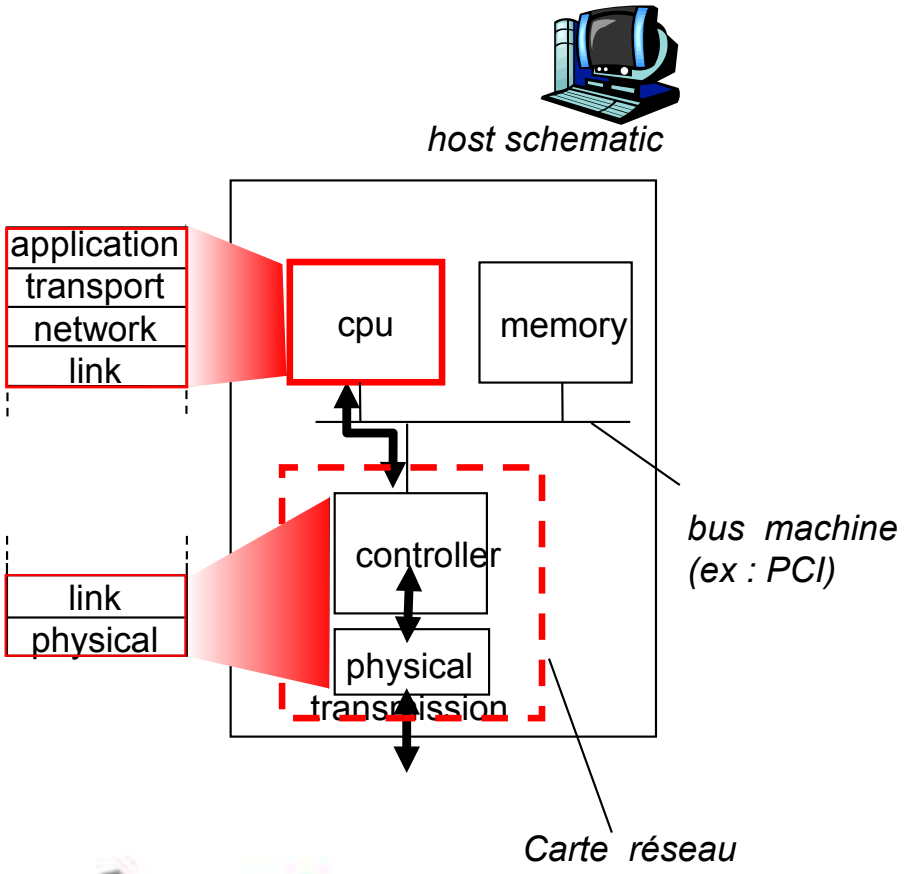
Services de la couche liaison de données

La présence des services ci-après dépend de la couche exacte :

- ***Transfert de données fiable***
  - On l'a vu avec TCP (TCP fonctionne de bout en bout!)
  - Au niveau 2, rarement utile sur les liens filiaires, mais peut l'être sur le sans fil où les pertes/corruptions peuvent être nombreuses
- ***Contrôle de flux:***
  - Contrôle du débit d'envoie sur un lien
- ***Détection d'erreur*:**
  - Erreurs causées par atténuation du signal ou bruit.
  - Le récepteur détecte la présence d'erreur et le signal à l'émetteur (sur le lien!!) pour une réemission
- **Correction d'erreur:**
  - Le récepteur identifie et peut corriger les bits en erreurs sans retransmission (dépend de la gravité de l'erreur. Certaines sont récupérables, d'autres non et il faut alors réemettre)

Où la couche liaison de données est-elle implantée?

- **Dans chaque appareil relié à un réseau**
- **La couche liaison de données est implantée dans la carte d'interface réseau (*network interface card* NIC)**
  - Carte Ethernet, carte 802.11
  - Implémente la couche liaison donnée et la couche physique
- **La carte est attachée au bus système de la machine**

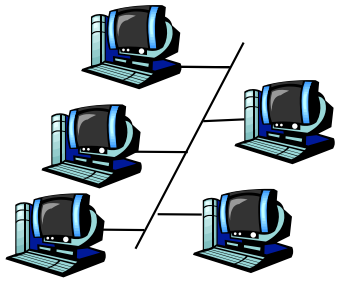


Agenda

- Introduction et services
- Protocoles pour accès multiples
- Adressage couche liaison et ARP
- Commutateurs

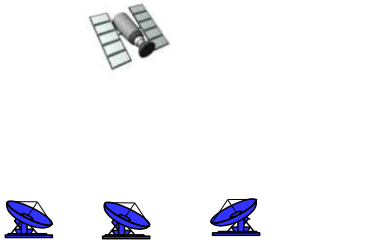
Liens à accès multiples et protocoles

- **point-à-point**
  - pour les accès modem type ADSL
  - Pour l'Ethernet dit « switché », lorsque chaque machine est sur un port de switch (=commutateur) distinct. Cas le plus courant!
- **diffusion (medium, filaire ou non, partagé)**
  - Ethernet traditionnel. Plus jamais le cas actuellement.
  - Ligne montante sur les réseaux câbles (ex: numéricable, etc)
  - LAN sans fil 802.11



Cable partagé (ex:., cable Ethernet)

Fréquence radio partagée (ex: 802.11 WiFi)



Fréquence radio partagée (satellite)

Protocoles d’accès au canal

Trois grandes familles :

- **Partitionnement du canal**
  - Le canal est découpé en sous-canaux (temporel, fréquentiel, avec des codes pour les réseaux mobiles 3G)
  - Les sous-canaux sont assignés à un nœud unique (tout le temps ou pour certaines périodes de temps)
  - Ex: 3G, modem ADSL
- **Accès aléatoire**
  - Canal non découpé, des collisions sont possibles
  - Algorithmes tentent de limiter les collisions
  - Ex : Ethernet, Wifi
- **“Chacun son tour”**
  - Chaque nœud a le droit d’émettre quand c’est son tour. Il utilise alors tout le canal
  - Ex : Token Ring (historique – n'est plus utilisé), bluetooth

Protocole d’Accès Aléatoire

- **Quand un nœud veut envoyer**
  - Transmet à la capacité maximale R
  - Pas de coordination *a priori*
- **Deux nœuds ou plus émettent en même temps ➜ “collision”,**
- **Protocoles MAC d’accès alétatoire specife:**
  - Comment détecter la collision
  - Comment renvoyer les données
- **Exemples de protocoles MAC d’accès aléatoire :**
  - slotted ALOHA
  - CSMA/CD (base d’ Ethernet), CSMA/CA (base de Wifi)

ALOHA slotté

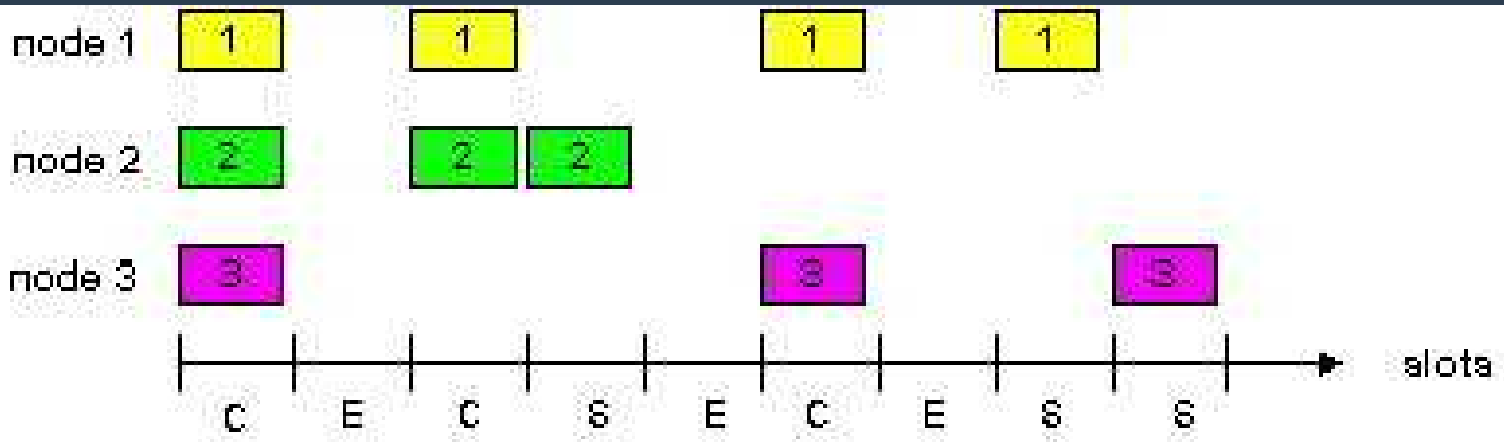
Hypothèses :

- **Toutes les trames ont la même taille**
- **Temps divisé en slots égaux (temps transmission d'une trame)**
- **Noeud peut transmettre au début d'un slot seulement**
- **Noeuds sont synchronisés**
- **Si 2 nœuds ou plus transmettent dans une slot, collision**

Opération:

- **Quand un nœud veut transmettre, il attend le prochain slot**
  - *Si pas collusion : peut continuer*
  - *Si collision:* nœud retransmet dans prochain slot avec une probabilité p

ALOHA Slotté



Avantages

- **Nœud actif utilise bien tout le canal**
- **Très décentralisé : seuls les slots doivent restés synchronisés**
- **simple**

Inconvénients

- **collisions**
- **Synchronisation horloge nœuds.**

CSMA (Carrier Sense Multiple Access)

CSMA: Ecouter avant de transmettre (on ne le faisait pas en Aloha!)

- Si canal inutilisé, on transmet
- Si canal utilisé, on attend prochain slot

- Dans Ethernet, on a du CSMA et on peut facilement détecté quand il y a collision car le signal revient brouillé.

- Dans Wifi, on a aussi du CSMA mais on arrive pas à détecter les collisions → c'est le récepteur qui doit renvoyer un acquittement

[Plus de détails sur ces protocoles dans d'autres cours à l'IUT]

Agenda

- Introduction et services
- Protocoles pour accès multiples
- Adressage couche liaison et ARP
- Commutateurs

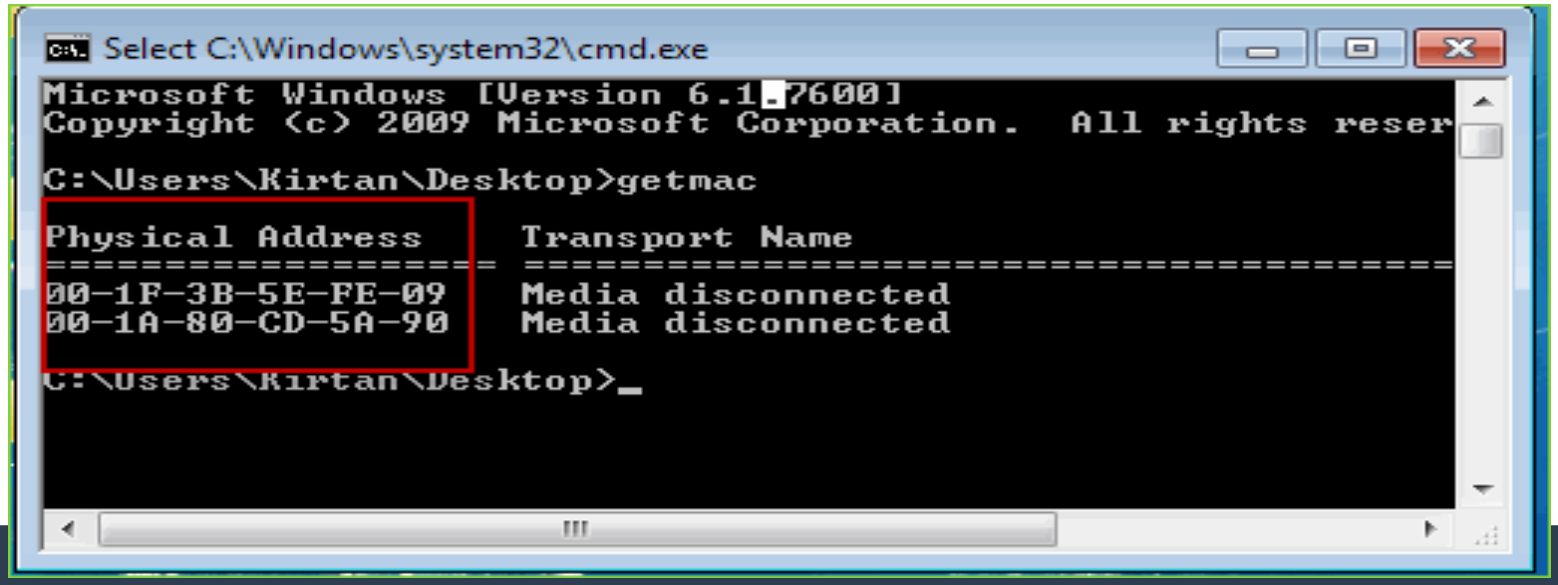
Adresses MAC

- **Les adresses IP font 32 bits**
  - *Contiennent une partie réseau*
  - Utilisé pour que le datagramme trouve le réseau IP destination
- **Adresses MAC (ou LAN ou physique ou Ethernet) :**
  - C'est le nom de l'adresse de la couche liaison de donnée !
  - Fonction : permet de transférer une trame d'une interface réseau à une autre interface sur le même réseau IP
  - 48 bits (pour la pluspart des LANs)
    - codée dans la mémoire morte de la carte réseau → choisie par le constructeur (garantie unicité de l'adresse)

Adresse MAC

- On représente souvent les adresses IP avec 4 octets sous forme décimale
- Pour les adresses MAC, on a une représentation en 6 octets… mais en hexadécimal

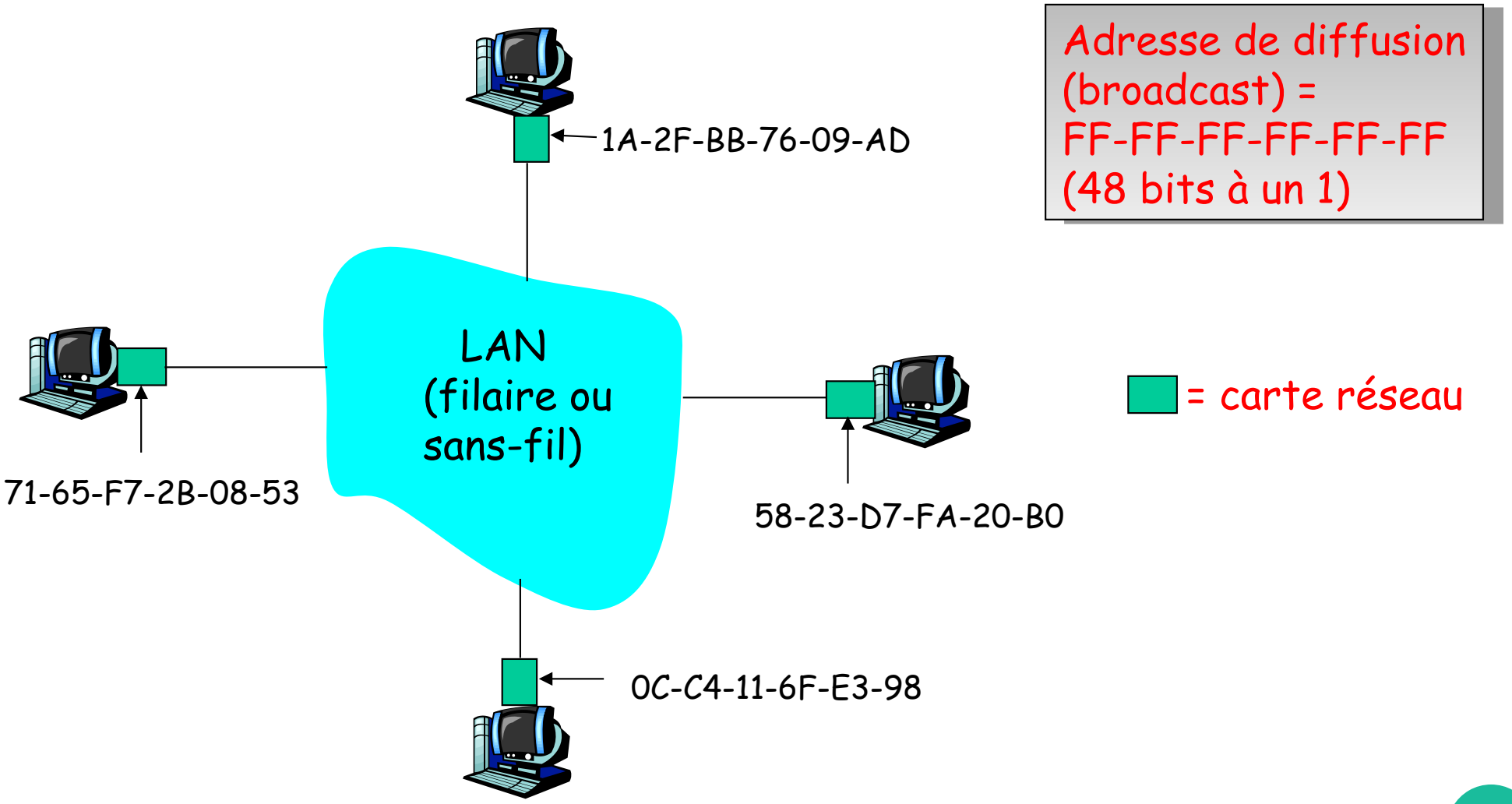
- Les machines s'en moquent, elles travaillent sur 32 ou 48 bits de toute façon
  - Ce ne sont que des conventions



2 cartes Réseaux → 2 @ MAC sur cette machine

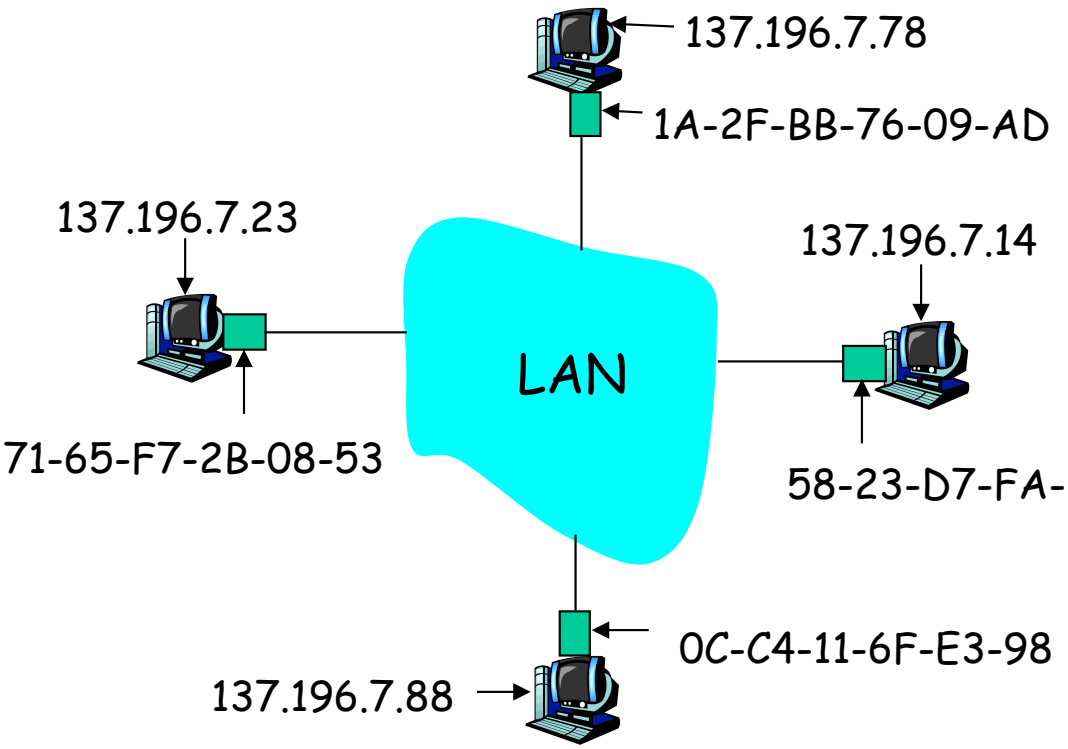
Adresses MAC dans un LAN

Chaque carte sur le LAN à une adresse unique



ARP: Address Resolution Protocol

Question: Comment déterminer l'adresse MAC de B à partir de son @ IP ?



- **Chaque nœud IP (machine, routeur) sur un LAN à une table ARP**

- **Table ARP : correspondance IP/MAC. Table organisée en enregistrements:**

< IP address; MAC address; TTL>

  - TTL (Time To Live): temps après lequel la correspondance est oubliée (typiquement 20 min)

Protocole ARP en action

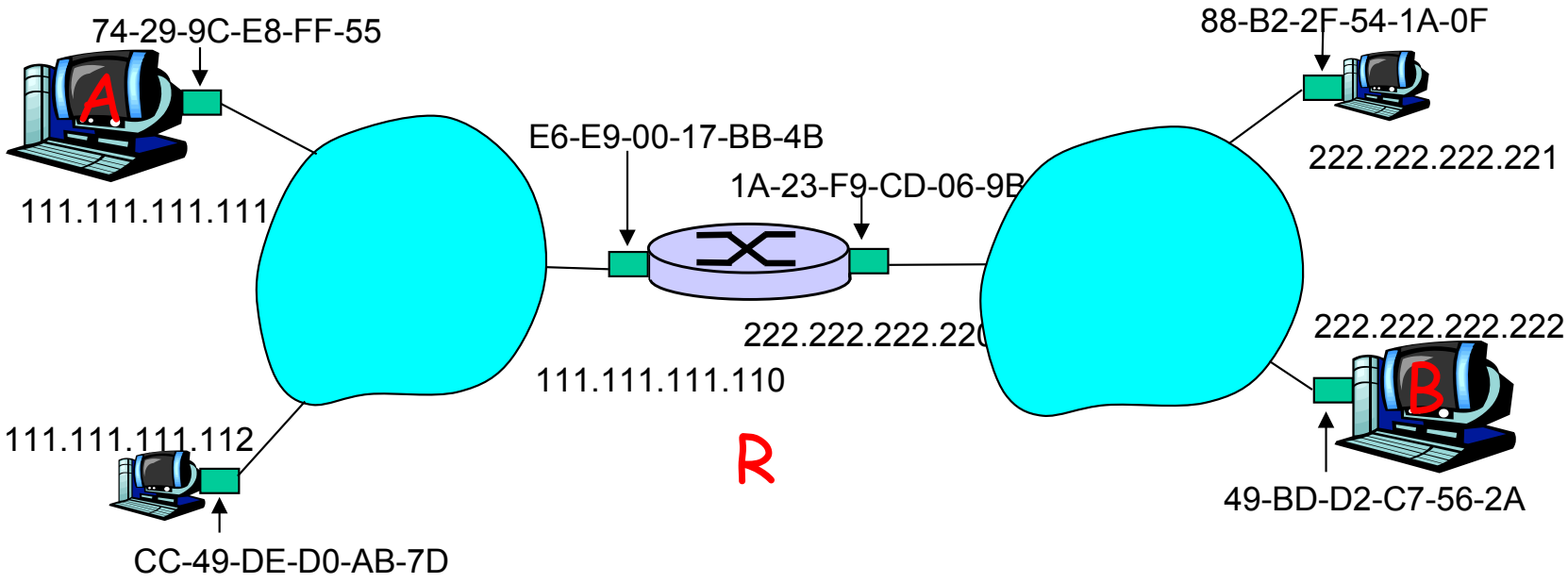
- **A veut envoyer un datagramme à B, et l'adresse MAC de B n'est pas dans la table ARP de A**
- **A diffuse une requête ARP qui contient l'adresse IP de B :**
  - Adresse MAC dest. = FF-FF-FF-FF-FF-FF
  - Toutes les machines sur le LAN reçoivent la requête ARP
- **B reçoit donc la requête ARP et envoie une réponse ARP.**
  - La réponse sert à savoir que la requête à réussie
  - L'adresse MAC de B est dans la trame de réponse!!!

- **ARP est "plug-and-play":**
  - Les nœuds créent leur table ARP sans l'intervention d'un administrateur
- **ARP est un protcole au même niveau qu'IP**

Routage d'un datagramme dans un autre sous réseau

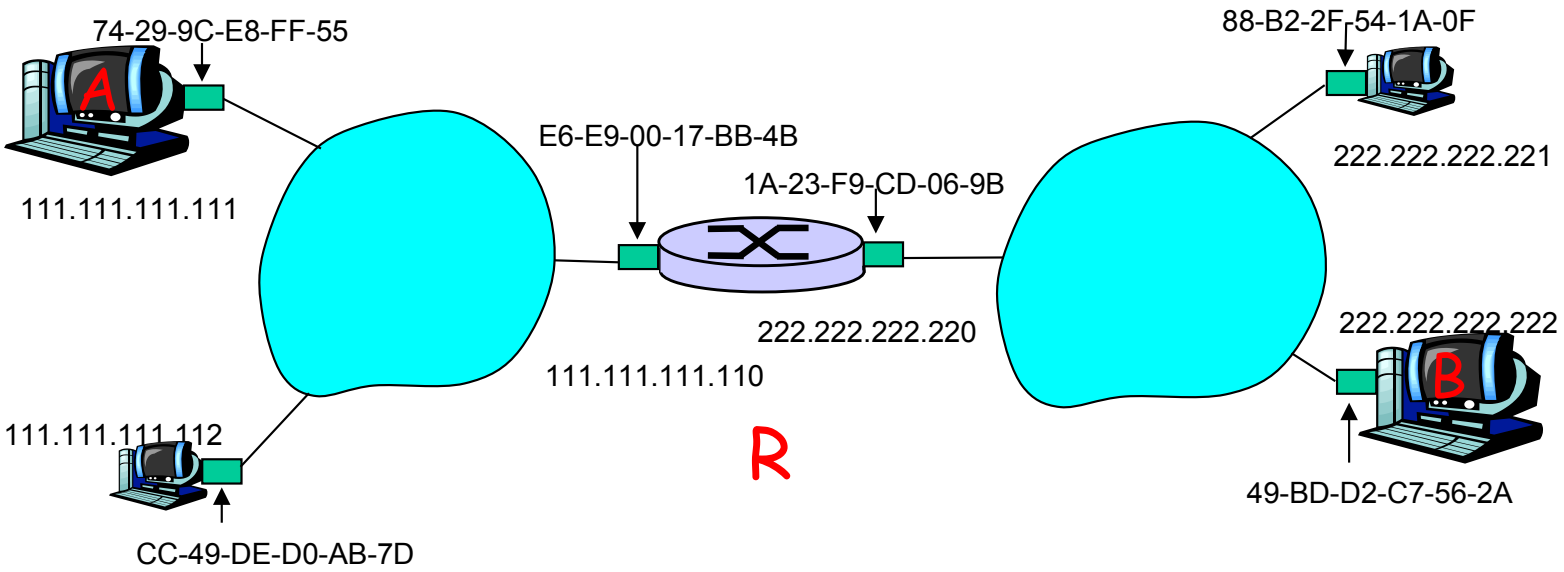
A envoie un datagramme à B via le routeur R

supposons que A connait l'adresse IP de B



- **Deux tables ARP dans le routeur R : une pour chaque LAN**
- **Dans le schéma, les @ MAC sont en hexadécimal et les @ IP en décimale**

- A crée un datagramme IP avec pour les adresses IP de A comme source et de B comme destination
- A utilise ARP pour trouver l'adresse MAC du routeur R à partir de son adresse IP qu'il connait (DHCP!!) 111.111.111.110
- A crée (et envoie) une trame avec l'adresse MAC de R comme destination et y insert le datagramme
- R retire le datagramme IP de la trame et voit qu'il est pour B
- R utilise ARP pour obtenir l'adresse MAC de B
- R crée (et envoie) une trame avec l'adresse MAC de B comme destination et y insert le datagramme



Agenda

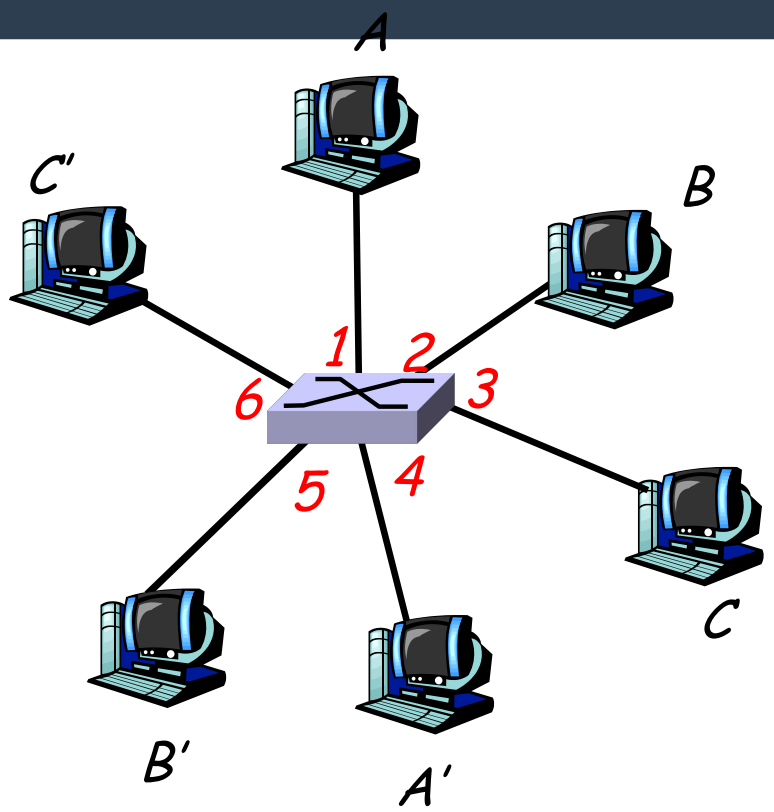
- Introduction et services
- Protocoles pour accès multiples
- Adressage couche liaison et ARP
- Commutateurs

Commutateurs - Switch

- **Equipement travaille niveau MAC**
  - Stocke re envoie des trames Ethernet
  - Examine adresse MAC soutrce, renvoie vers un ou plusieurs de ses liens ensuite. Utilise CSMA/CD quand il envoie
- ***Transparent***
  - Machines ne sont pas conscients de sa présence
- ***plug-and-play, auto-apprentissage***
  - Pas besoin d'être configurés

Switch: permettent plusieurs transmission simultanées

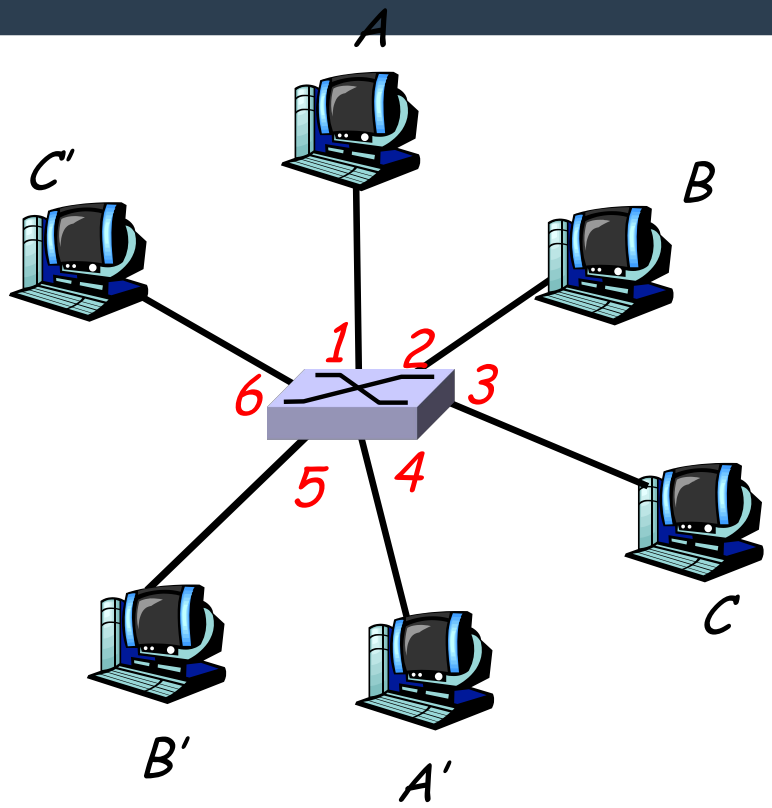
- A-vers-A’ et B-vers-B’ peuvent avoir lieu simultanément sans collision



switch with six interfaces (1,2,3,4,5,6)

Table du Switch

- **Q: comment le switch sait où que A' est joignable via l'interface 4 et B' via 5 ?**
- **R: chaque switch a une table de commutation qui contient des lignes du type:**
  - (Adresse MAC machine, interface pour joindre cette machine, estampille temporelle)
- **Q: comment ces entrées sont apprises ?**



switch with six interfaces (1,2,3,4,5,6)

Switch: auto apprentissage

- **switch apprend la correspondance machine/interface**
  - Quand une trame est reçue, le switch apprend où est l'émetteur

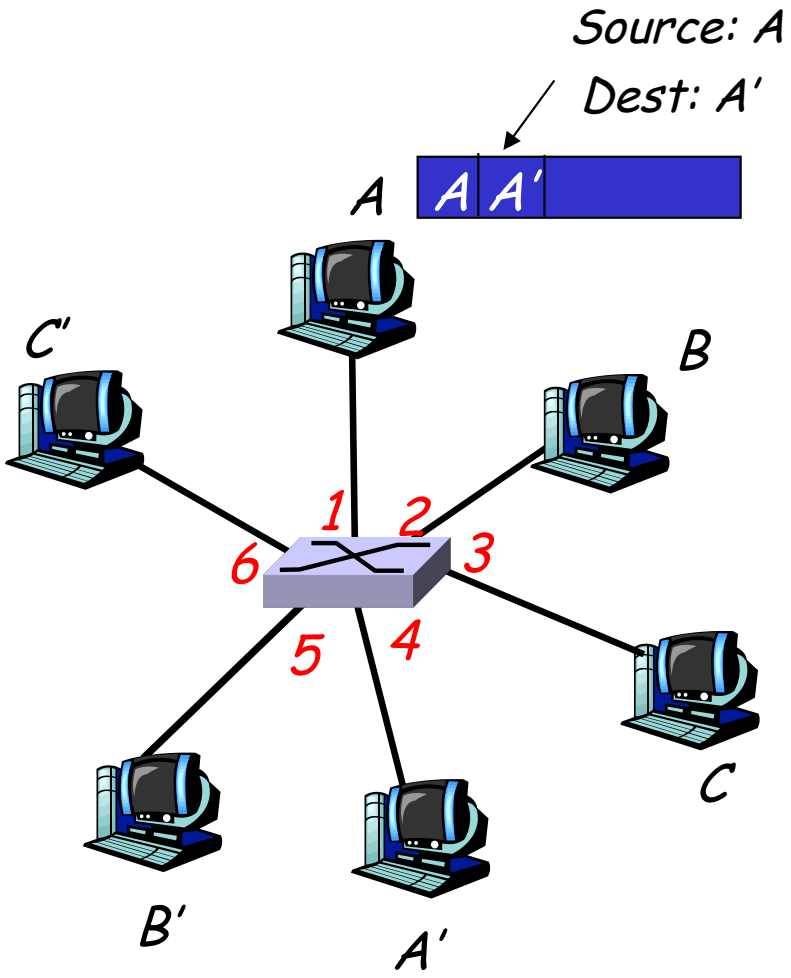


Table ARP switch (initialement vide)

MAC addr	*interface*	*TTL*
A	*1*	*60*

Switch: Algorithme

Quand trame reçue:

1. **Enregistre la correspondance source et lien d'entrée**
2. **Cherche l'interface de la destination**
3. **Si destination existe dans table du switch**

Alors {

Si c'est le même lien → destruction trame # la source et la destination # sont sur le même lien

Sinon on transmet

}

sinon on envoie sur toutes les interfaces (sauf celle d'arrivée)