Réseaux

Les couches liaison et réseau

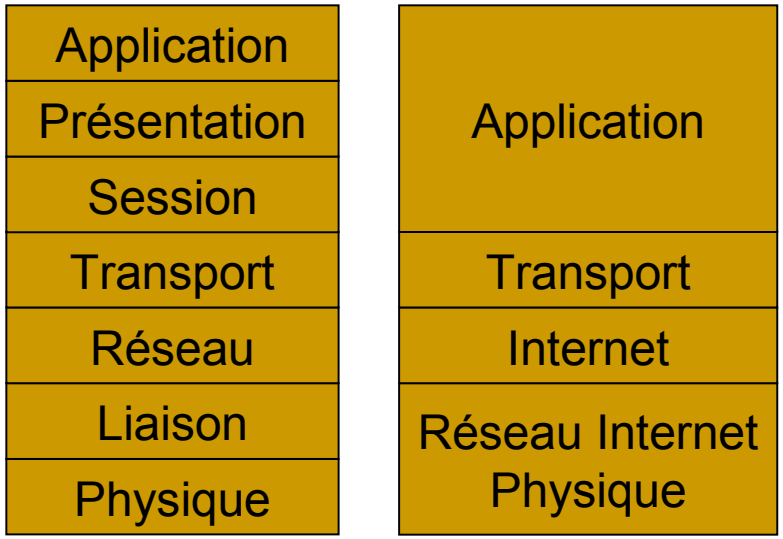
La couche transport

Protocoles UDP-TCP et Applications

Les bases

Le modèle OSI

(Open Systems Interconnect Reference Model et le modèle IP)



TCP/IP ne suit pas scrupuleusement les préconisations de l'ISO.

Niveau 1 : Couche Physique
Signaux électriques, lumineux, le format des connecteurs

Niveau 2 : Couche Liaison
Trames de bits entre deux émetteurs en liaison directe, ethernet, fast ethernet.

Niveau 3 : Couche Réseau
Routage dans les machines du réseau et démultiplexage dans les extrémités : IP

Niveau 4 : Couche Transport
Contrôle de flux, reprise sur erreur, remise dans l'ordre des paquets.

Niveau 7 : Couche application
Applications réseau, messageries, transfert de fichier, etc.

- Les équipements de routage n'implémentent que les trois premières couches
- Seuls les ordinateurs source et destination implémentent les 7 couches
- L'utilisateur ne se sert que de la couche 7

La couche liaison de données : Ethernet

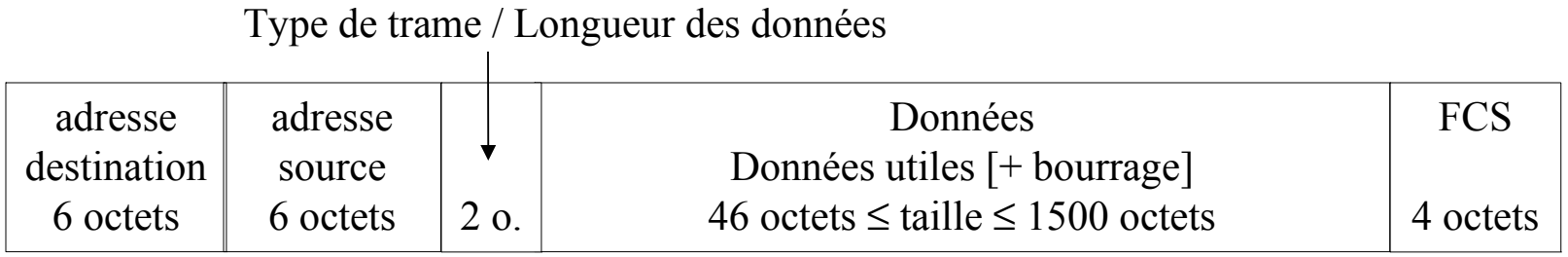
Principes d'Ethernet

- Support de transmission
  - brin = segment = bus = câble coaxial
  - pas de boucle
  - pas de sens de circulation
- Chaque carte Ethernet possède une adresse unique au niveau mondial (adresse MAC)
- Pas de multiplexage en fréquence $\Rightarrow$ une seule trame à un instant donné
- Réception par tous les transceivers du réseau d'une trame émise par une station

Principe du CSMA/CD

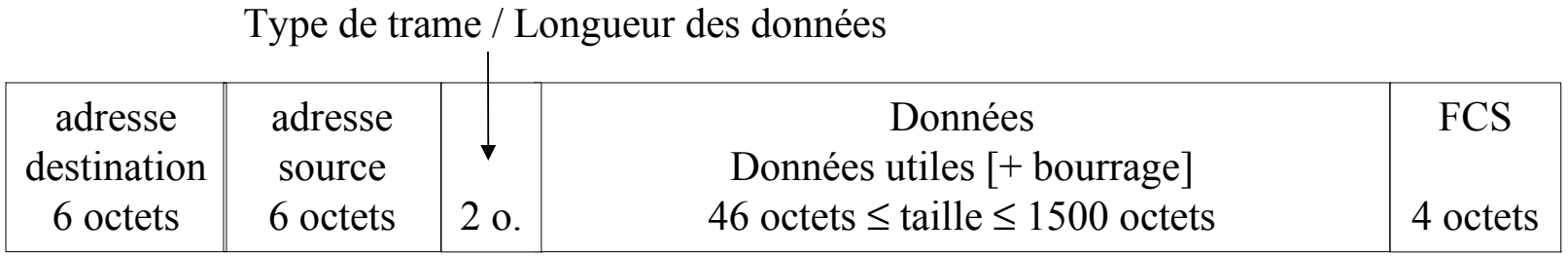
- Carrier Sense Multiple Access with Collision Detection
- Si rien à transmettre, alors station silencieuse
- Si besoin d'émettre
  - écoute pendant 9,6 µs minimum (IFG)
  - si quelqu'un émet on recommence à écouter
  - sinon envoie de la trame mais écoute pendant 51,2 µs (slot time)
    - si trafic reçu pendant slot time alors collision !!!
    - si collision alors émission d'un jam (enforcement de collision) pour que tout le monde détecte la collision pendant au moins 32 bit times
    - attente d'un délai aléatoire (algorithme de backoff) avant réémission

Format des trames Ethernet (1/3)



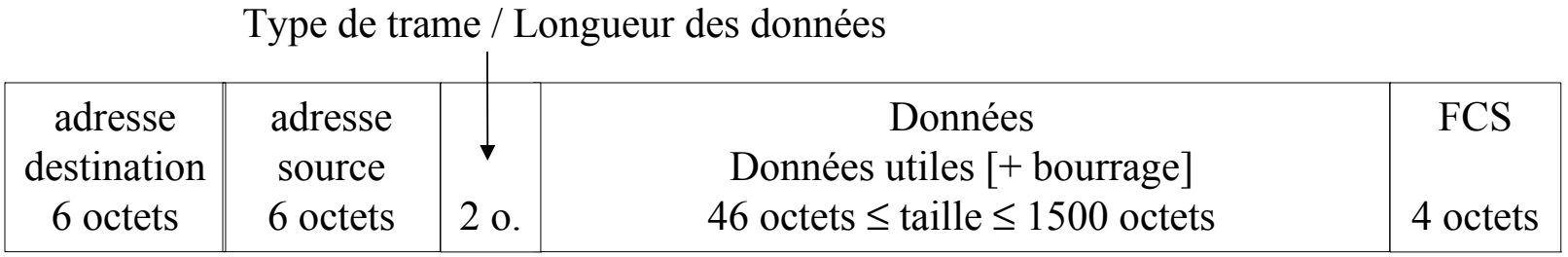
- Préambule de 56 bits pour la synchronisation des horloges + SFD
- Adresses attribuées par l'IEEE (notation hexadécimale)
  - `08:00:20:xx:xx:xx` pour Sun
  - `00:00:0C:xx:xx:xx` pour Cisco
  - `00:A0:24:xx:xx:xx` pour 3Com
  - diffusion (broadcast) : `FF:FF:FF:FF:FF:FF`
  - diffusion de groupe Internet (multicast) : `01:00:5E:xx:xx:xx`

Format des trames Ethernet (1/3)



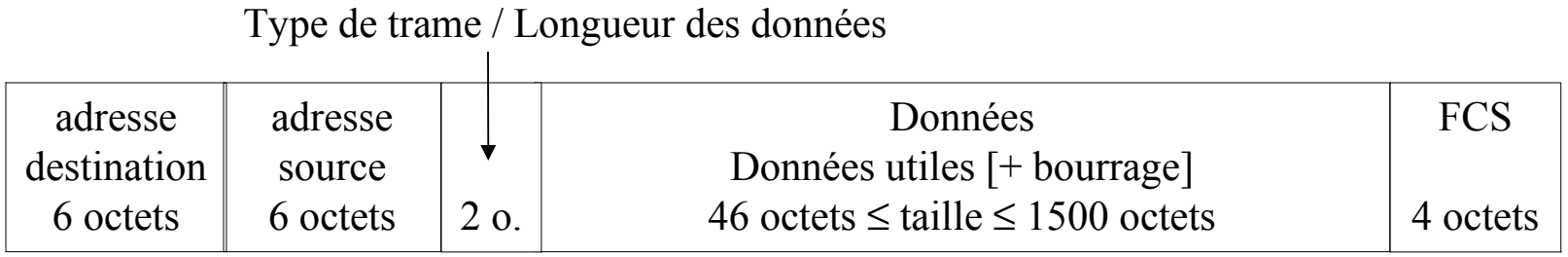
- Préambule de 56 bits pour la synchronisation des horloges + SFD
- Adresses attribuées par l'IEEE (notation hexadécimale)
  - `08:00:20:xx:xx:xx` pour Sun
  - `00:00:0C:xx:xx:xx` pour Cisco
  - `00:A0:24:xx:xx:xx` pour 3Com
  - diffusion (broadcast) : `FF:FF:FF:FF:FF:FF`
  - diffusion de groupe Internet (multicast) : `01:00:5E:xx:xx:xx`

Format des trames Ethernet (2/3)



- Champ type identifie le protocole utilisé dans la trame
  - administré globalement par Xerox (valeur supérieure à 1500)
  - liste dans le fichier `/usr/include/netinet/if_ether.h`
    - `0x0800` : IP
    - `0x0806` : ARP
- Longueur des données si pas de type
  - taille inutile car déduite de SFD à fin de porteuse
  - taille fixe des champs autres que données

Format des trames Ethernet (3/3)



- Données utiles
  - de 1 à 1500 octets
  - MTU maximum de 1500 octets
  - si moins de 46 octets alors bourrage (padding) pour faire au moins 46 octets
- FCS (Frame Control Sequence)
  - Code détecteur d'erreur
  - CRC calculé sur la totalité de la trame

La couche réseau : IP

Internet Protocol

Le protocole IP

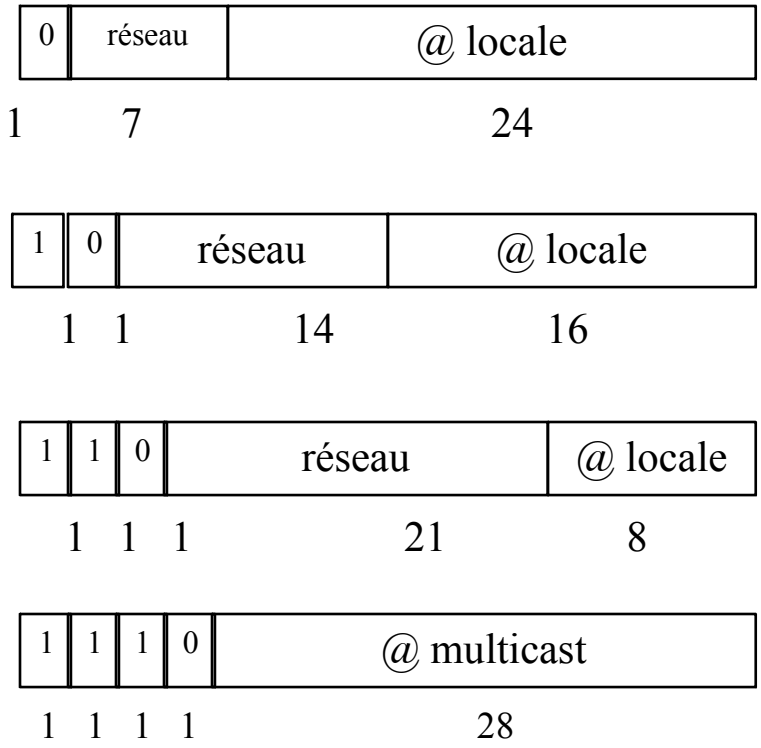
- Service fourni par IP :
  - transmission de paquets machine à machine
  - service sans connexion et non fiable
    - des paquets peuvent être perdus
    - des paquets peuvent être erronés
    - des paquets peuvent être dupliqués
    - Pas de garantie de remise (« *Best effort* »).
  - les paquets peuvent être reçus dans le désordre
  - fragmentation des datagrammes en fonction des réseaux traversés (Maximum Transport Unit).

Adressage IP

- Adresse IP unique au monde (ne pas confondre avec Ethernet)
- Configurable par logiciel
- Attribuées par le NIC (Network Information Center)
- Adresse sur 32 bits en notation décimale pointée
  - exemple : 194.199.20.90
- Découpage en 2 :
  - adresse de réseau
  - adresse de machine

Adressage IP - Classes de réseaux

- Classe A : N.H.H.H
  - de 0.0.0.0 à 127.255.255.255
- Classe B : N.N.H.H
  - de 128.0.0.0 à 191.255.255.255
- Classe C : N.N.N.H
  - de 192.0.0.0 à 223.255.255.255
- Classe D
  - de 224.0.0.0 à 239.255.255.255
- Classe E
  - de 240.0.0.0 à 247.255.255.255



Adressage IP

- Adresse de réseau :
  - identificateur de réseau suivi de bits à 0
  - Exemples :
    - 125.0.0.0 = réseau 125 de classe A
    - 129.15.0.0 = réseau 129.15 de classe B
    - 192.168.30.0 = réseau 192.168.30 de classe C
- Adresse de diffusion ou broadcast :
  - identificateur de réseau suivi de bits à 1
  - Exemples :
    - 125.255.255.255 = diffusion sur le réseau 125 de classe A
    - 129.15.255.255 = diffusion sur le réseau 129.15 de classe B
    - 192.168.30.255 = diffusion sur le réseau 192.168.30 de classe C

Adressage IP

- Adresse de machine
  - Exemples :
    - 125.5.6.7 = machine 5.6.7 du réseau 125 de classe A
    - 129.15.106.213 = machine 106.213 du réseau 129.15 de classe B
    - 192.168.30.11 = machine 11 du réseau 192.168.30 de classe C
- 127.x.x.x
  - adresse de bouclage (loopback localhost)
  - désigne la machine locale
- 0.0.0.0
  - utilisé quand une machine ne connaît pas son adresse

Adressage de sous-réseau

- Utilisation des bits d'identificateur de machines pour identifier des sous-réseaux
- Exemple : Réseau de classe B 140.30

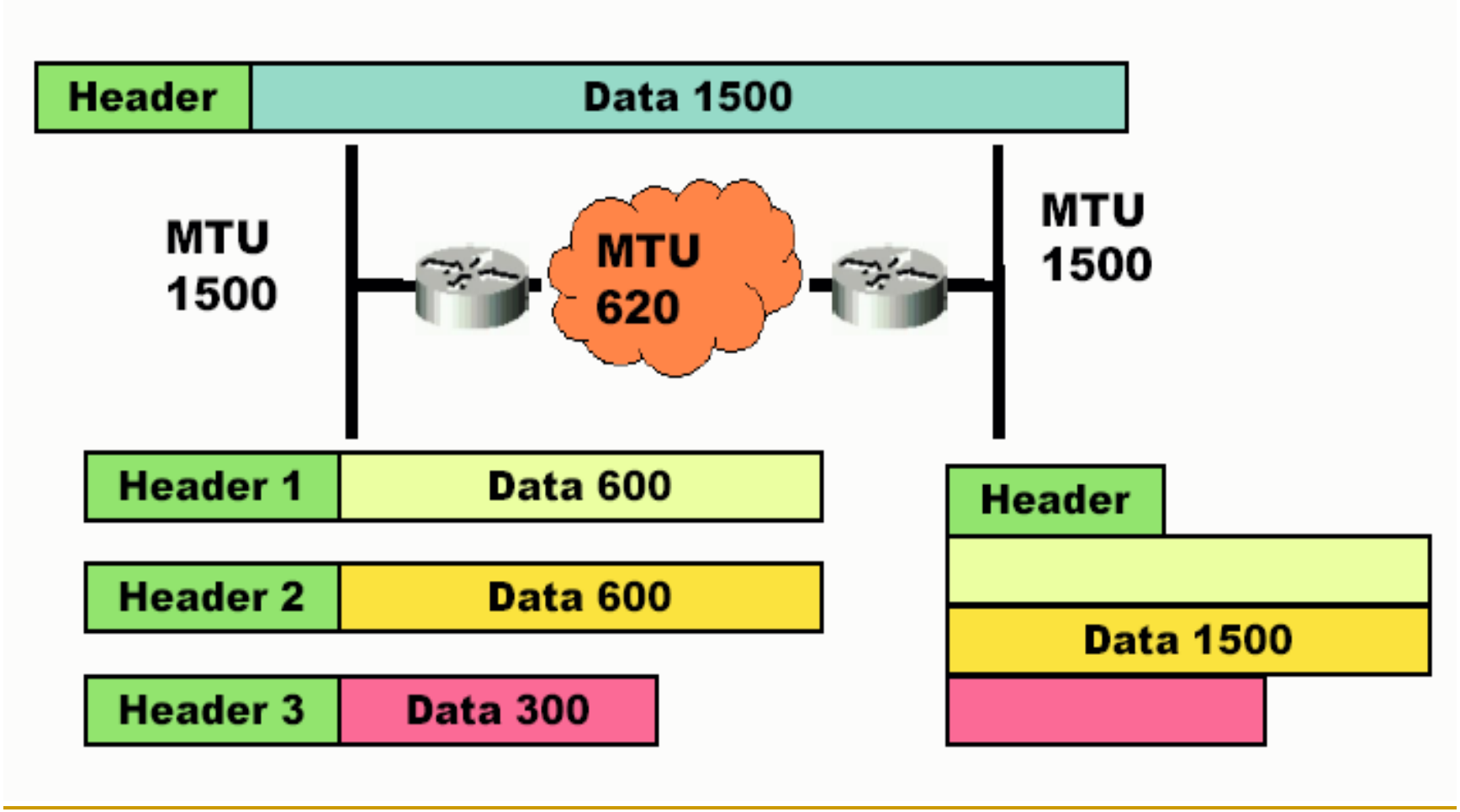
Id réseau 16 bits 140.30	Id sous-réseau 8 bits	Id machine 8 bits

  - autorise 254 réseaux de 254 machines
  - masque de réseau classe B : 255.255.0.0
  - masque de sous-réseau : 255.255.255.0
  - si (@IP_dest & masque == mon@IP & masque)
    - alors envoi_direct (datagramme, @IP_dest)
    - sinon envoi_indirect(datagramme, @IP_dest, routeur(@IP_dest & masque))

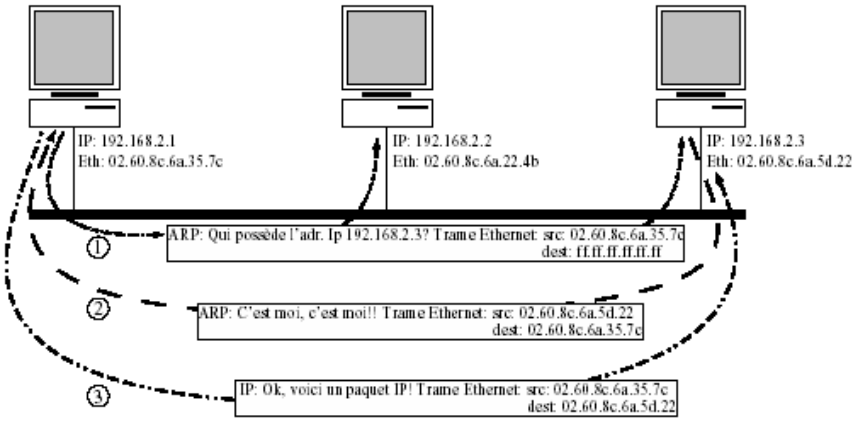
L’adressage IP

- RFC 761
- Codage sur 32 bits
- Unique au monde (organismes de gestion des adresses)
- Notation décimale pointée :
  - 194.199.229.197 (0xC2C7 E5C5)
- Adresses spéciales :
  - Adresse de réseau (ID machine ne contient que des 0)
  - Adresse de diffusion (ID machine ne contient que des 1)
  - Adresse de bouclage (127.0.0.1)
  - Les adresses de 10.0.0.0 à 10.255.255.255 ne sont pas routables

Fragmentation IP



ARP



- RFC 826.
- Champ type d'une trame Ethernet : 0x0806
- Détermination de l'adresse MAC (physique) d'une machine du réseau LAN à partir de son adresse IP.
- Diffusion d'un message ***ARP request*** à toutes les machines du LAN
- Seule la machine qui reconnaît son adresse IP répond par ARP

RARP

- RFC 903.
- Champ type d'une trame Ethernet : 0x8035
- L'adresse IP d'une machine est configurable (elle dépend du réseau sur lequel elle se trouve).
- Elle est souvent enregistrée dans un fichier par le système d'exploitation.
- Ce fonctionnement usuel n'est plus possible dès lors que la machine est une station sans disque.
- RARP est un mécanisme permettant à la station d'obtenir son adresse IP depuis le réseau.

ICMP
(Internet Control and Error message Protocol)

- Les réseaux IP envoient des datagrammes
- Le chemin des paquets n'est pas connu
- Les sources sont informées des problèmes (erreurs, congestion, …) à l'aide de messages ICMP
- **Une erreur engendrée par un message ICMP ne peut donner naissance à un autre message ICMP (évite l'effet cumulatif).**

0 Echo Reply [RFC792]
1 Unassigned [JBP]
2 Unassigned [JBP]
3 Destination Unreachable [RFC792]
4 Source Quench [RFC792]
5 Redirect [RFC792]
6 Alternate Host Address [JBP]
7 Unassigned [JBP]
8 Echo [RFC792]
9 Router Advertisement [RFC1256]
10 Router Selection [RFC1256]
11 Time Exceeded [RFC792]

12 Parameter Problem [RFC792]
13 Timestamp [RFC792]
14 Timestamp Reply [RFC792]
15 Information Request [RFC792]
16 Information Reply [RFC792]
17 Address Mask Request [RFC950]
18 Address Mask Reply [RFC950]
19 Reserved (for Security) [Solo]
20-29 Reserved (for Robustness Experiment) [ZSu]
30 Traceroute [RFC1393]
31 Datagram Conversion Error [RFC1475]
32 Mobile Host Redirect [David Johnson]

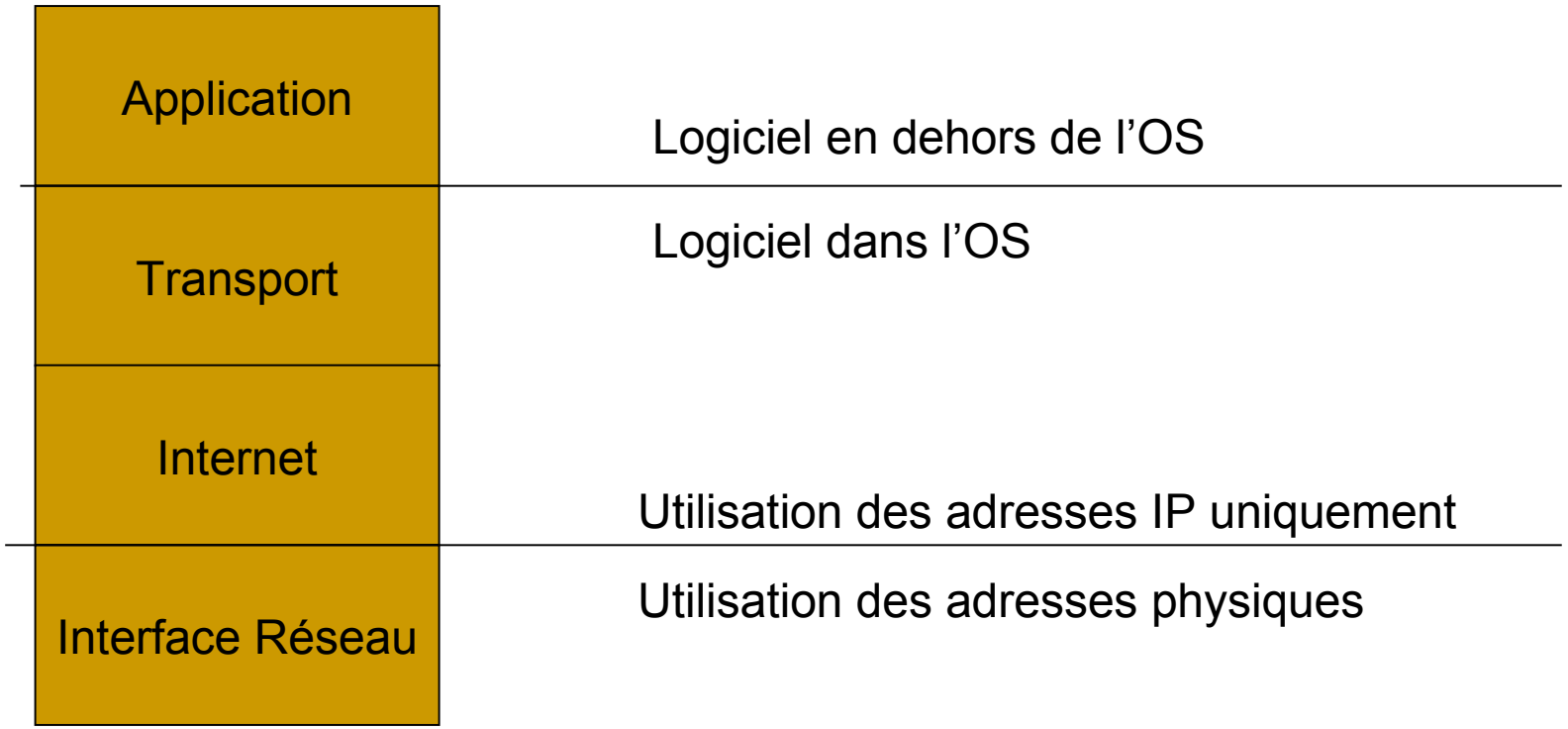
33 IPv6 Where-Are-You [Bill Simpson]
34 IPv6 I-Am-Here [Bill Simpson]
35 Mobile Registration Request [Bill Simpson]
36 Mobile Registration Reply [Bill Simpson]
37-255 Reserved [JBP]

La couche transport

Problématique

Deux protocoles de transport pour Internet : UDP et TCP

La couche de transport dans Internet



La couche transport problèmes

- Type de services:
  - De même que pour les services au niveau de la couche réseau il y a les services de transport :
    - sans connexion
    - orientés connexion (établissement d'une connexion,…)
- Pourquoi ?
  - Que se passe-t-il si le fournisseur de service offre un service non fiable?
  - Que se passe-t-il si un des routeurs tombe en panne ?
  - Que se passe-t-il si des paquets sont perdus ?
  - Que se passe-t-il si une connexion se termine de manière non prévisible ?

Les utilisateurs n'ont aucun contrôle sur le réseau

La couche transport objectifs

- **Rôle de la couche transport** :
  - Résoudre les problèmes qui peuvent apparaître dans les couches sous jacentes.
  - Améliorer la qualité du service
  - Les primitives de la couche transport doivent être indépendantes de la couche réseau. Ainsi les applications peuvent être écrites indépendamment du réseau et de la topologie.

- distinction entre les couches :
  - **fournisseurs de service de transport** : 4 couches inférieures
  - **utilisateurs des services de transport :** couche(s) supérieure(s)

La couche transport
Le problème de la qualité de service

- Qualité du service (QoS) :
  - Il devrait être spécifié des qualités de type ”bonne”, “acceptable”, “insuffisante” pour les utilisateurs
  - faire la traduction entre les spécifications des utilisateurs et des paramètres de QoS plus concrets tels que le délai, la priorité, le débit, la taille des paquets.
- Quelles sont les valeurs minimums et désirées pour les paramètres ?
- Négociation entre entités paires.

La couche transport *vs.* couche liaison de données

- Éléments des protocoles de transport
  - Les fonctionnalités de la couche transport sont similaires à celles de la couche liaison de données : les deux traitent le contrôle d'erreur, le séquensement, le contrôle de flux…
- **Différence majeure** : Les environnements sont différents
  - La couche liaison n'est pas obligée de spécifier le routeur destination
  - L'établissement de la connexion est plus complexe à la couche transport. **Plusieurs connexions peuvent être établies en même temps**, la taille des tampons est plus difficile à prévoir.
- Les paquets peuvent rester dans le réseau un temps inconnu.

Fonctions de la couche transport

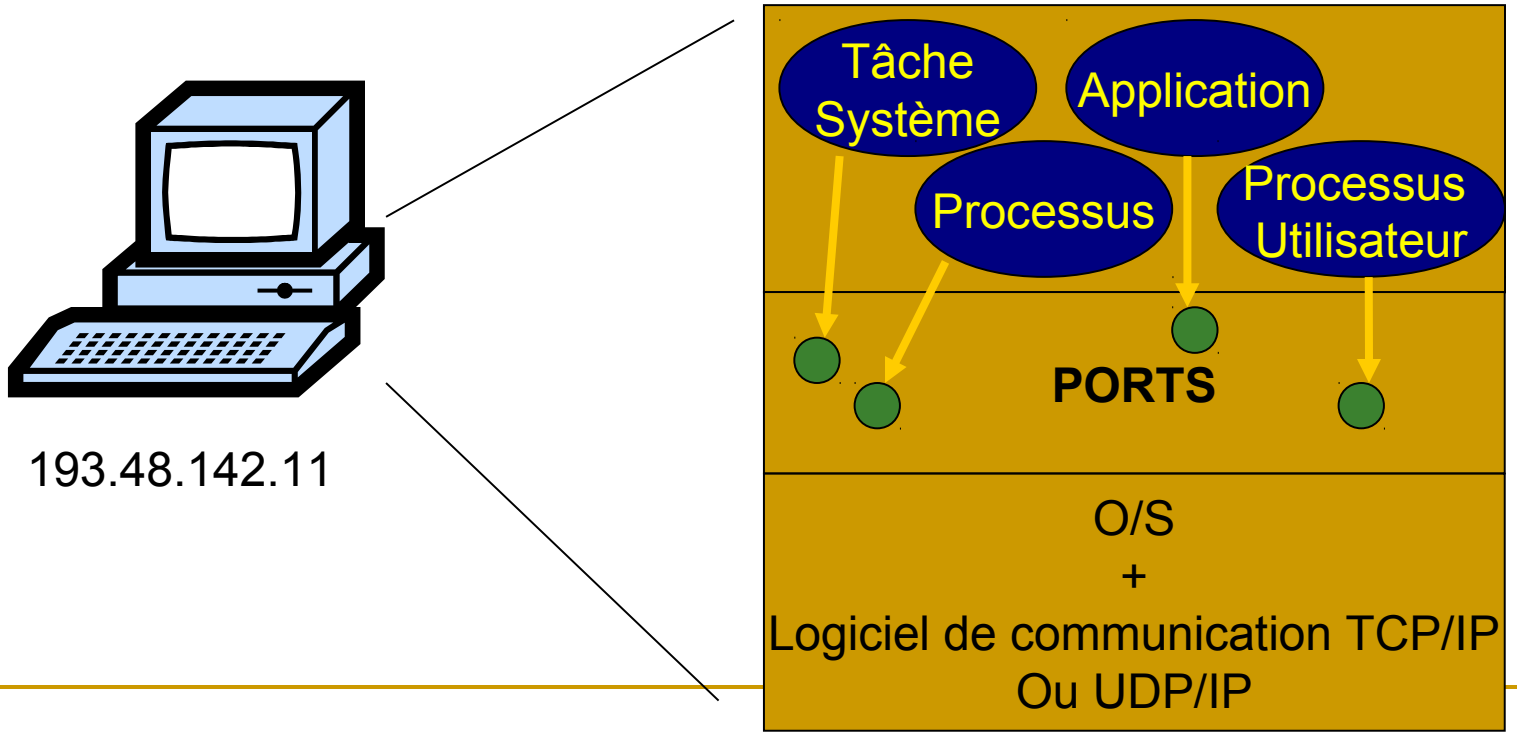
- Adressage : point d'accès à l'application
- Établissement/fermeture d'une connexion
- Segmentation : découper un message en paquets
- Contrôle de flux et tampons : pour éviter les débordements
- Multiplexage et démultiplexage
- Recouvrement d'erreurs (IP n'est pas fiable)

Transport et adressage (1)

- Une application désireuse d'ouvrir une connexion avec une autre application distante, doit définir « *l'adresse* » de la couche transport où écouter les communication
- Différence avec IP :
  - applications *vs.* machines

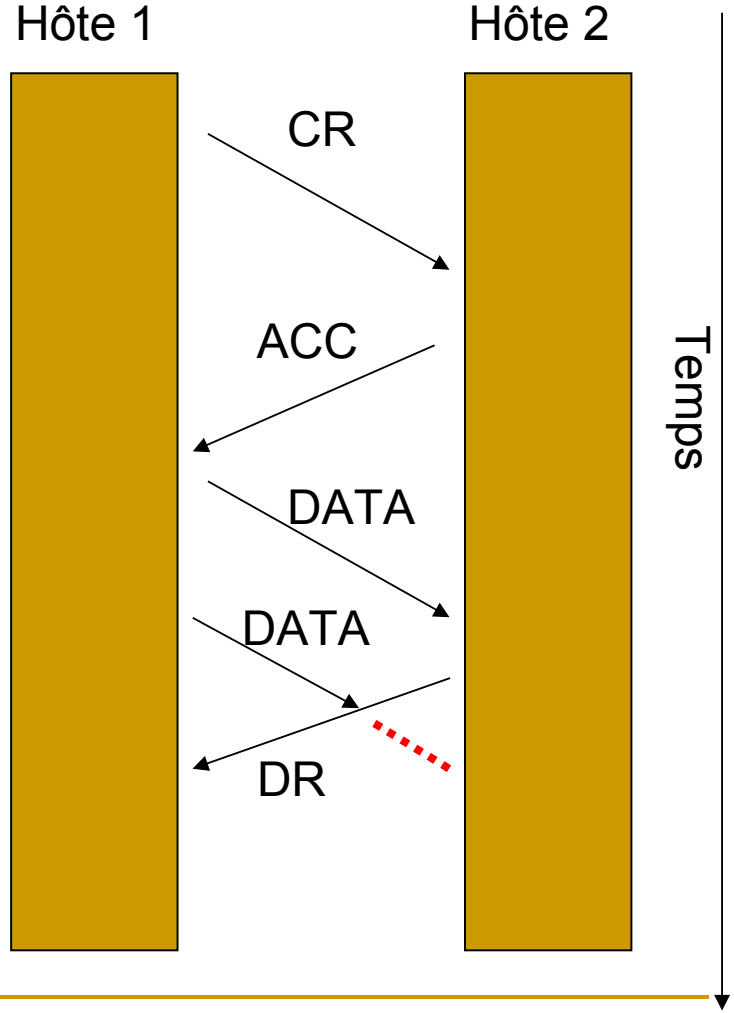
Transport et adressage (2)

- Quelle est l'origine du message ?
- Quelle est la destination finale ?



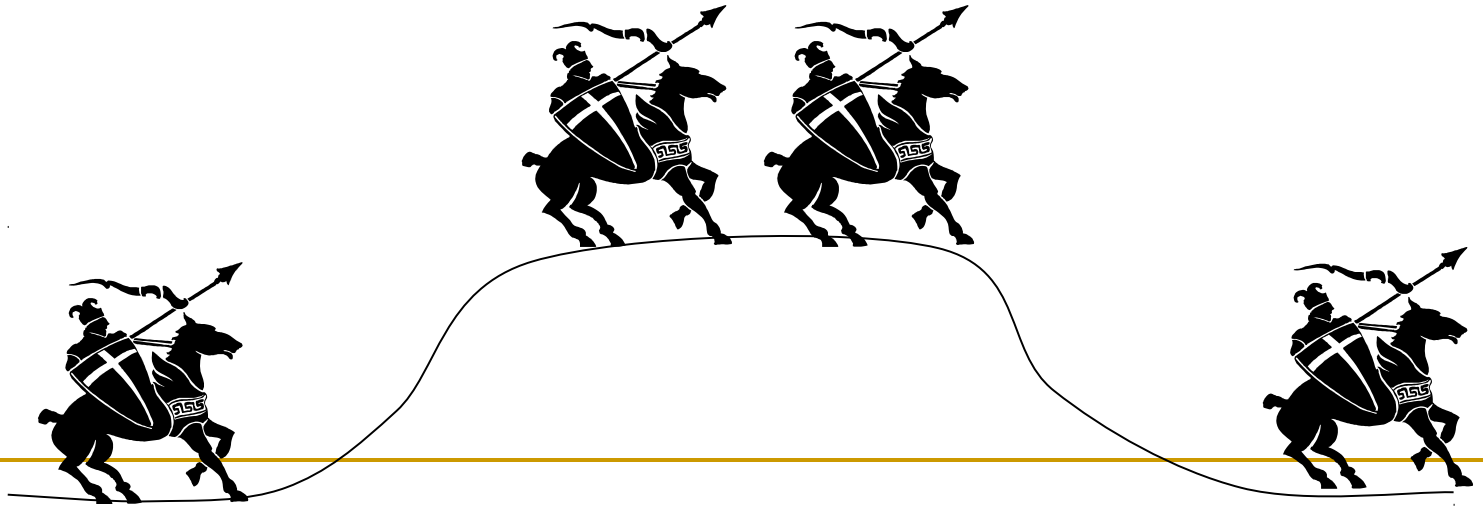
Établissement et fin d'une connexion

- Libération d'une connexion
  - une déconnexion abrupte peut conduire à la perte de données. Il y a deux manières de terminer une connexion :
    - **symétrique** : chaque direction est terminée indépendamment l'une de l'autre.
      - Cette approche est utilisée quand on connaît le volume de données à transmettre.
    - **asymétrique**:
      - **hôte 1:** "J'ai terminé, avez-vous terminé ?"
      - **hôte 2**: "j'ai également terminé, au revoir"



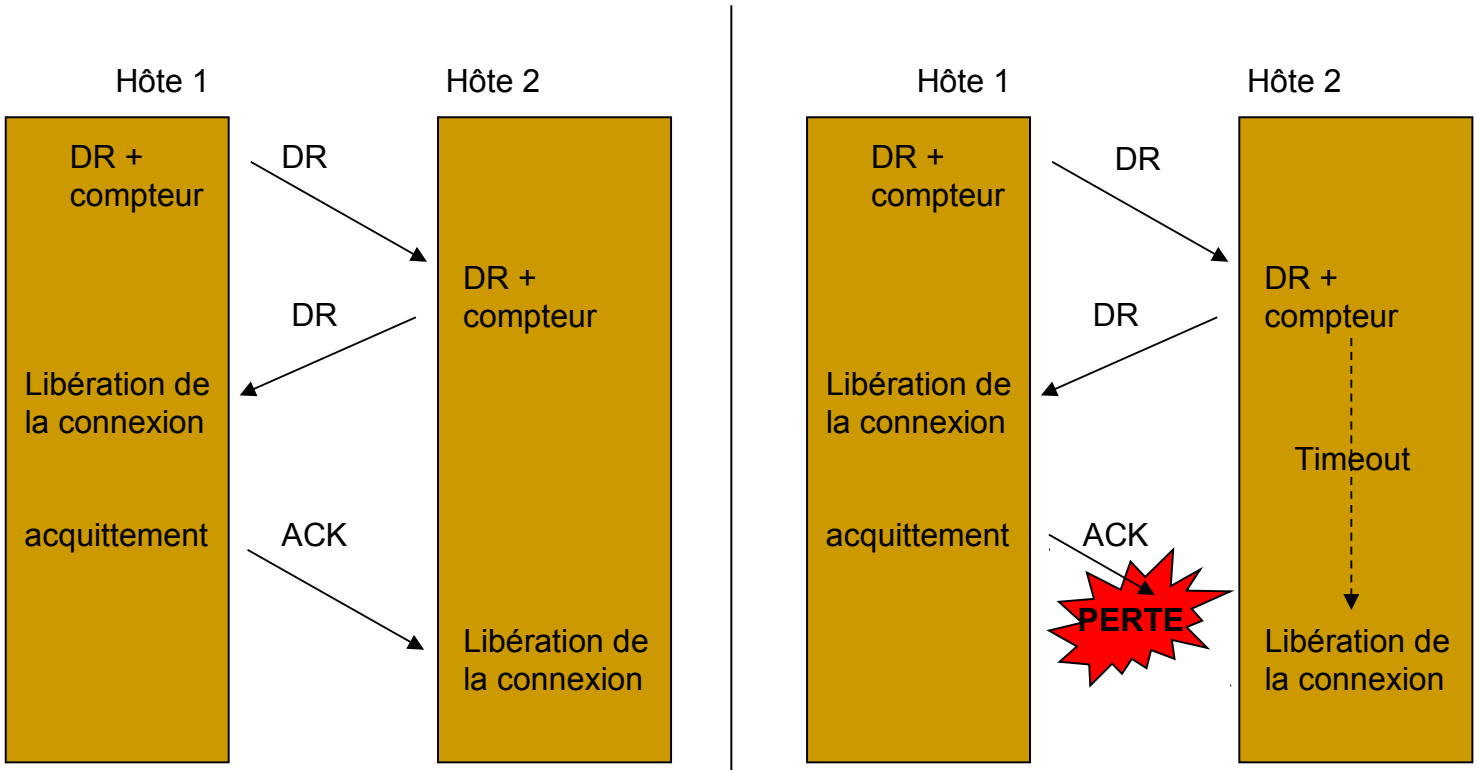
Établissement et fin d'une connexion (2)

- Obtenir une fiabilité pour les connexions asymétriques est théoriquement impossible.
- Exemple: le problème des deux armées les bleus gagnent s'ils peuvent trouver un protocole par envoi de messager qui coordonne leur attaque.
  - **Problème**: comment l'armée #1 sait, que l'armée #2 sait, que l'armée #1 sait, … ?



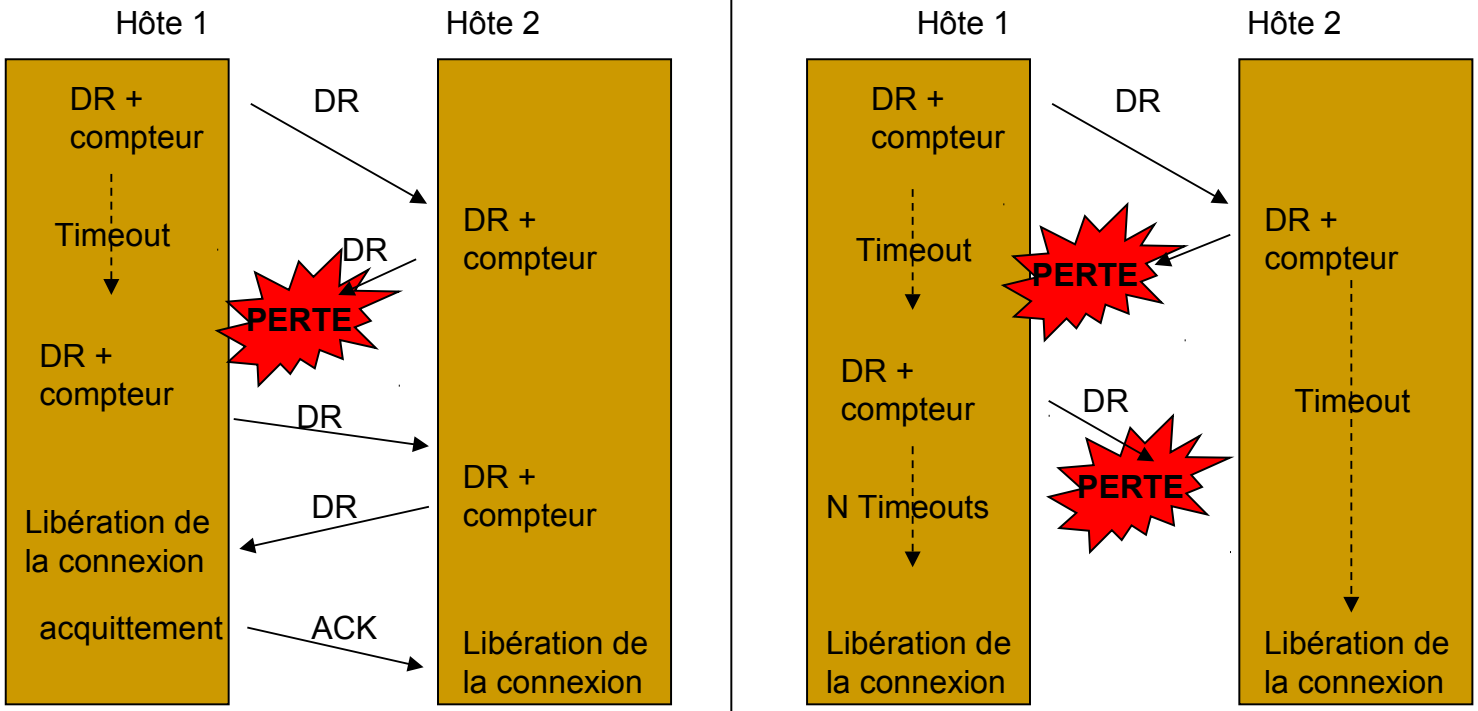
Établissement et fin d'une connexion (3)

- Solution: "Je vais terminer puis vous attendre"



Établissement et fin d'une connexion (4)

- La majorité des problèmes sont résolus avec ce protocole, mais il y a toujours un risque d'avoir une connexion à moitié ouverte. Comment ?



Les protocoles de la couche transport

- UDP : User Datagram Protocol
  - Protocole en mode **sans connexion**
- TCP : Transmission Control Protocol
  - Protocole en mode **orienté connexion**

Partie commune à TCP et UDP

- Orientation application *vs* orientation machine (cf. IP)
- Les sockets, identification d'une application
  - Adresse IP d'une machine
  - Numéro de port
- Un couple de sockets définit une connexion TCP ou un échange UDP
  - Exemple 10.1.73.26:23 et 10.1.73.58:1094
    - Connexion entre un processus client à partir du port 1094 de la machine 10.1.73.58 et un daemon telnet sur la machine 10.1.73.26

Les ports réservés à TCP/UDP

- Comment un hôte peut-il savoir si un numéro de port est utilisé par une machine distante ?
- Deux approches:
  - Fixer des ports standards
  - Attribuer dynamiquement les ports
- Les protocoles Internet adoptent une approche mixte

Les ports réservés à TCP/UDP (2)

- Exemples de ports réservés
  - 7 ECHO *Ping d'une station*
  - 21 FTP *File Transfer Protocol*
  - 53 DOMAIN *Domain name server*

UDP

User Datagram Protocol

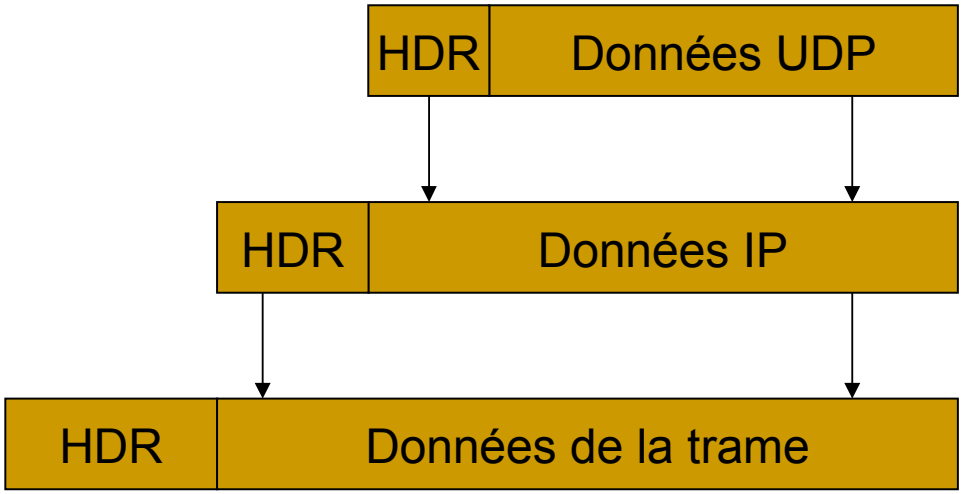
- Service simple **sans connexion**
- Simple en-tête ajouté aux paquets IP

UDP : Objectifs

- UDP fournit les mécanismes primaires dont les applications ont besoin pour envoyer des datagrammes à d'autres applications.
- UDP fournit un service non fiable sans connexion :
  - Les messages peuvent être perdus, dupliqués ou arriver dans le désordre.
- Il permet de distinguer différentes destination pour un hôte donné.
- Les n° de port destination et source sont fournis.

- Messages sans liens:
  - L'application doit réordonner et contrôler le flux

Encapsulation UDP

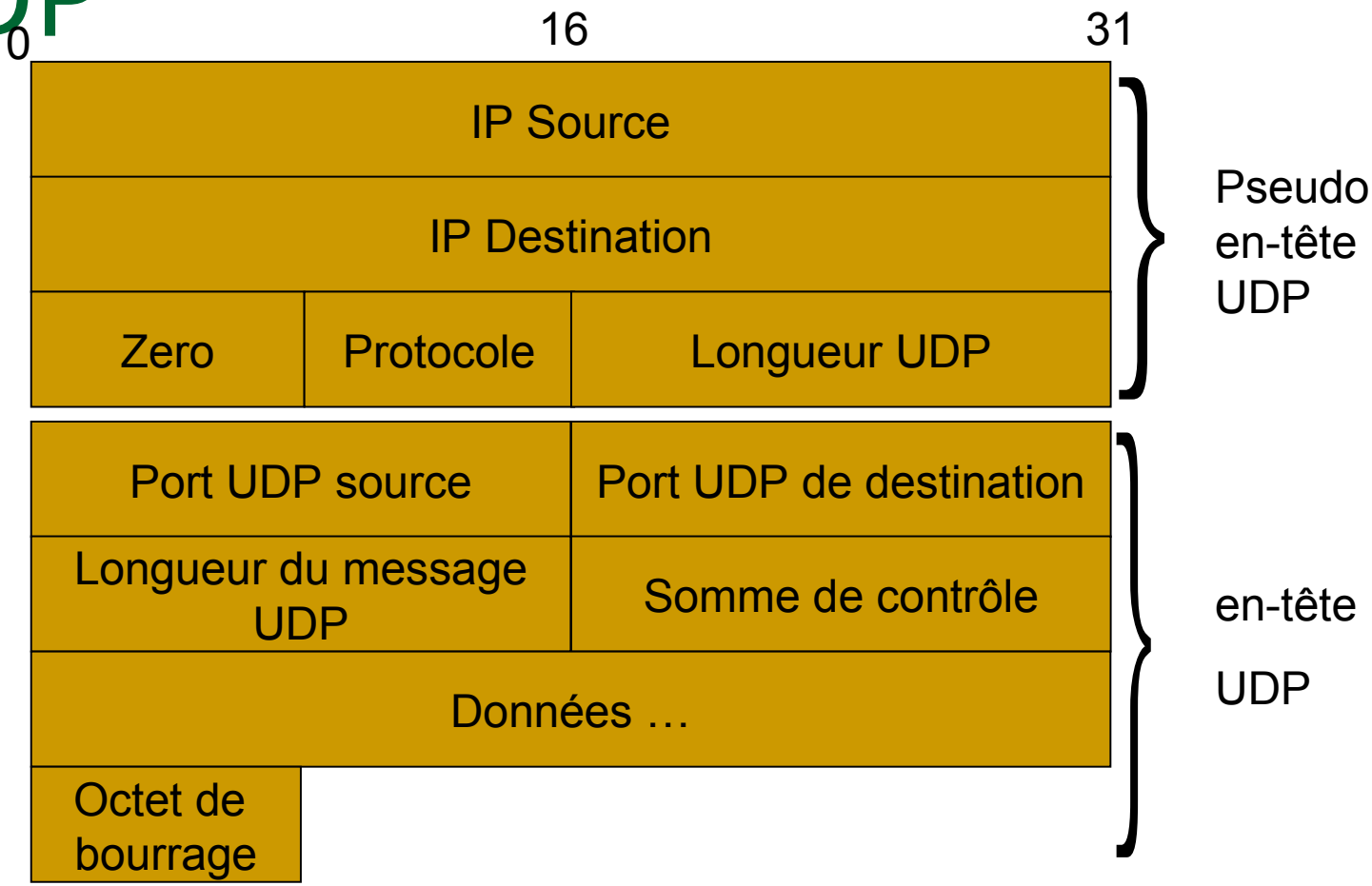


La couche UDP différencie plusieurs sources et destinations sur un hôte

IP gère le transfert entre les hôtes

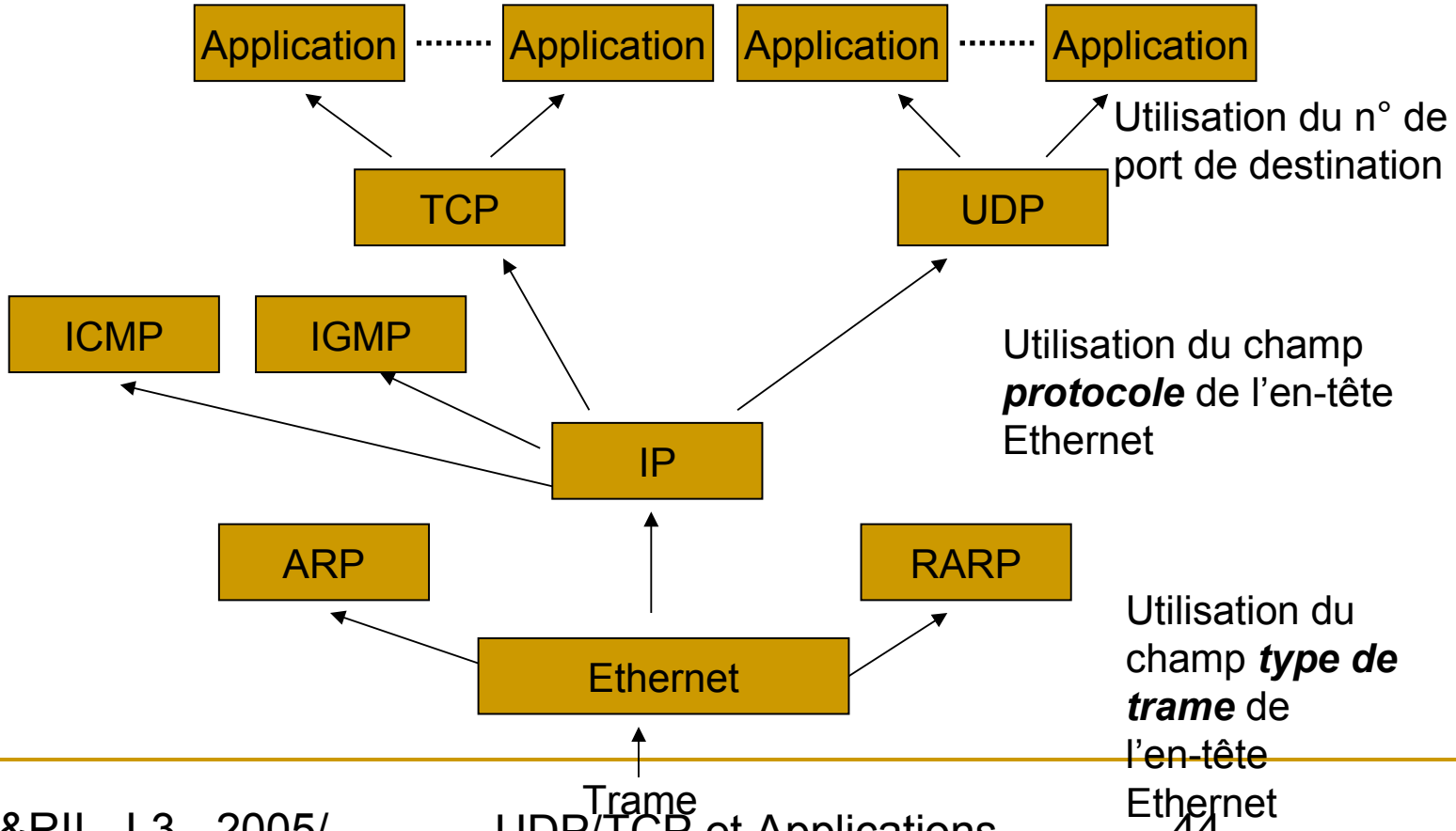
La couche physique transfère une trame sur le réseau physique

Pseudo en-tête et en tête UDP



Le multiplexage et le démultiplexage

- Les application négocient avec l'O/S pour obtenir un numéro de port



Ports UDP

- No. Port Mot-clé Description
- 7 - 0x07 ECHO Echo
- 11 - 0x0B USERS Active Users
- 13 - 0x0D DAYTIME Daytime
- 37 - 0x25 TIME Time
- 42 - 0x2A NAMESERVER Host Name Server
- 53 - 0x35 DOMAIN Domain Name Server
- 67 - 0x43 BOOTPS Boot protocol server
- 68 - 0x44 BOOTPC Boot protocol client
- 69 - 0x45 TFTP Trivial File transfert protocol
- 123 - 0x7B NTP Network Time Protocol
- 161 - 0xA1 SNMP Simple Network Management protocol

File d'attente et UDP

- L'O/S crée une file d'attente par port
- La taille de cette file est spécifiée et changée par l'application
- Quand UDP reçoit un datagramme, il contrôle le numéro de port de la destination avec la liste des ports actifs en cours d'utilisation
- En cas d'erreur un message ICMP « port unreachable error » est envoyé

UDP et la fragmentation IP

- IP négocie avec l'interface locale la MTU (Maximun trasfert unit)
- Si le datagramme à une taille supérieure, IP fragmente, et seule la destination réassemble.
- En changeant de réseau un fragment de datagramme IP (paquet) peut être à nouveau fragmenté

- Conséquence : **Si un paquet IP est perdu tout le datagramme UDP doit être retransmis**.

UDP - Conclusion

- UDP est un protocole simple
- Il fournit (au dessus de IP) :
  - Numéro de port
  - Somme de contrôle optionnelle
- IP vérifie uniquement les en-têtes, UDP vérifie aussi les données
- Compatible avec la fragmentation IP

TCP

Le protocole de transport de référence pour Internet

Les caractéristiques de TCP

- TCP fournit aux applications
  - Une communication point à point
  - Un mode connecté
  - Une orientation « flux de données »
  - L'ouverture et la fermeture des connexions propre et fiable
  - Une communication en « full duplex »
  - Une interface applicative
  - Ne supporte pas le multicast

TCP un service de transport fiable

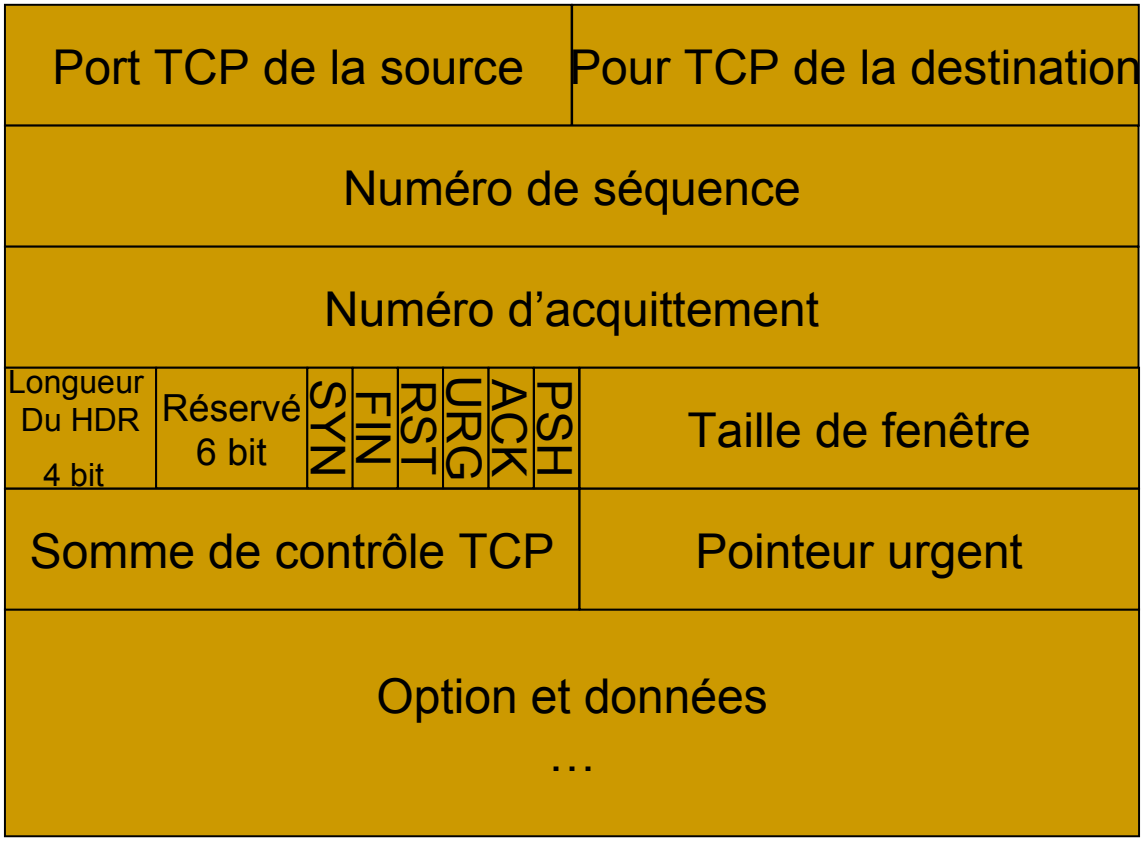
- La fiabilité est parfois fondamentale même en dépit de la performance et de la simplicité
  - Diffusion multimédia « temps réel »
  - Écriture sur un système de fichier (NFS)

- Réception des données
  - Dans le bon ordre
  - Sans perte, ni duplication

Les ports TCP

No. port	*Mot-clé*	*Description*
20 - 0x14	FTP-DATA	File Transfer [Default Data]
21 - 0x15	FTP File	Transfer [Control]
23 - 0x17	TELNET	Telnet
25 - 0x19	SMTP	Simple Mail Transfer Protocol
37 - 0x25	TIME	Time
42 - 0x2A	NAMESERVER	Host Name Server
43 - 0x2B	NICNAME	Who Is
53 - 0x35	DOMAIN	Domain Name Server
79 - 0x4F	FINGER	Finger
80 - 0x50	HTTP	Hyper Text Transfert Protocol (WWW)
110 - 0x6E	POP3	Post Office Protocol - Version 3
111 - 0x6F	SUNRPC	SUN Remote Procedure Call

L'en-tête TCP

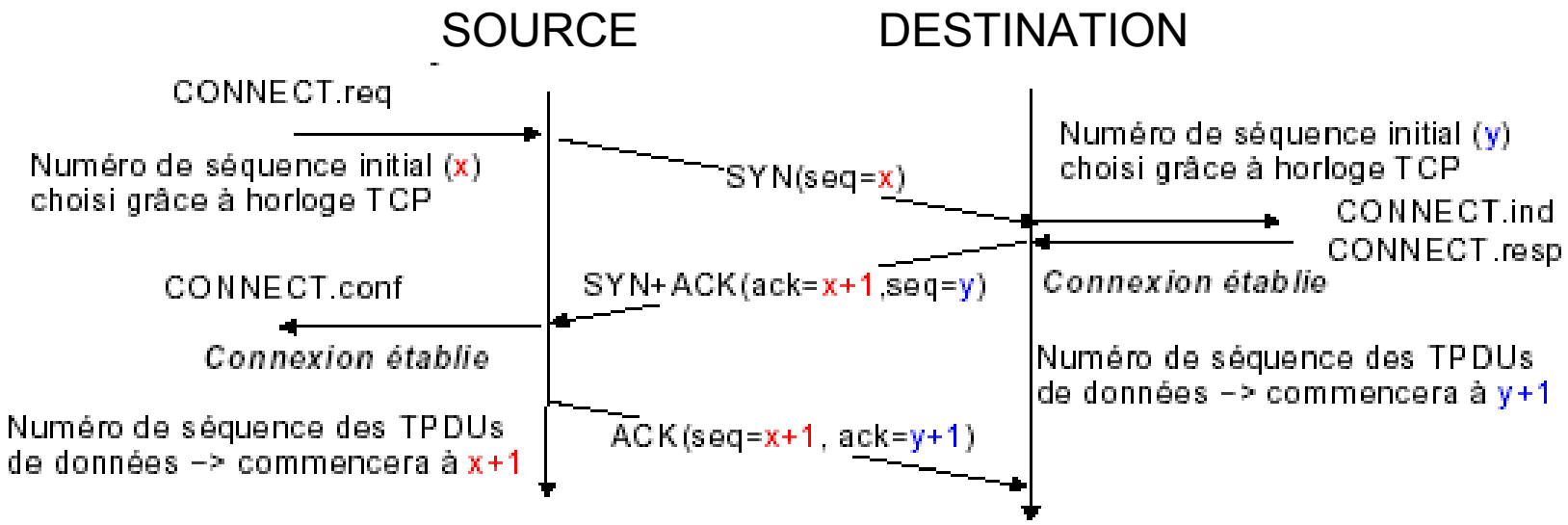


- Offset : longueur de l'entête en mots de 32 bits (e.g. 0x5).
- **URG** (Urgent : exemple Ctrl+C).
- **ACK** (Acknowledge : accusé de réception).
- **PSH** (Push : délivrer immédiatement les donées).
- **RST** (Reset : reprise d'une connexion).
- **SYN** (Synchronisation : demmande d'établissement d'une connexion).
- **FIN** (Finalize : Termine la connexion).
- Fenêtre : Nombre d'octets que l'on peut envoyer sans recevoir d'acquittement.

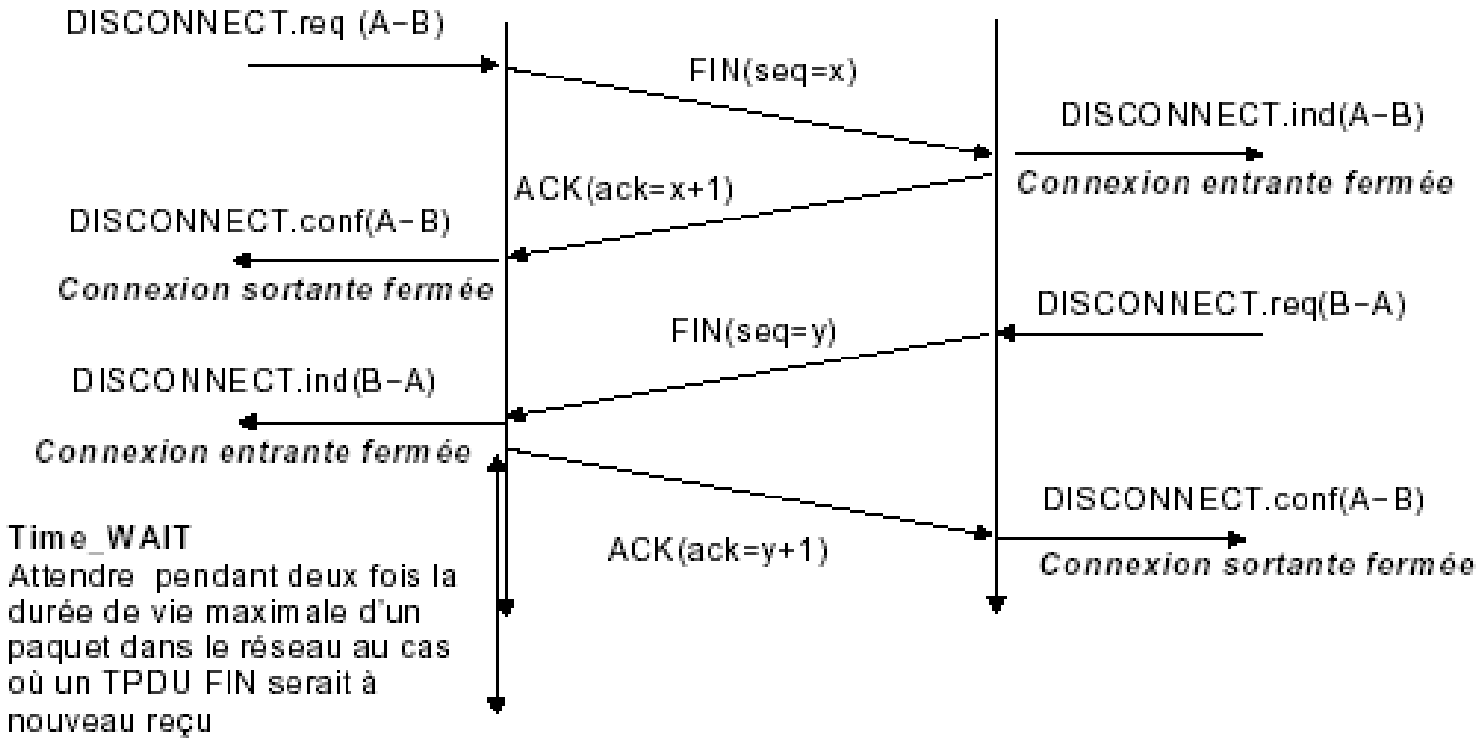
Un TPDU (Transport Protocol Data Unit) TCP est appelé un ***segment***

Un paquet a une durée de vie de 120 secondes

Ouverture de connexion TCP



Fermeture de connexion TCP



Transfert fiable

- Chaque segment TCP contient
  - 16 bits **checksum** : utilisé pour détecter les erreurs de transmission sur l'en-tête **et** le contenu
- **sequence number** (un octet consomme un n° séquence)
  - utilisé par l'émetteur pour délimiter les segments transmis
  - utilisé par le receveur pour réordonner les segments reçus
- **acknowledgement number**
  - utilisé par le receveur (si ACK est vrai) pour annoncer à l'émetteur le numéro de séquence du prochain **octet** attendu

- Comment faire face aux pertes de segments ?
  - protéger chaque segment par un temporisateur
  - si le temporisateur expire avant la réception de l'acquittement correspondant, retransmettre
- TCP utilise GO-BACK-N dans ces conditions

Les acquittements

- Ils Peuvent être transportés avec les données
- Ils Acquittent un nombre d'octets de données reçus
  - Pas forcément un segment entier
  - L'acquittement peut être retardé (avec un timeout)
- Problème des paquets arrivés en « désordre »

- Retransmission
  - A l'envoi d'un paquet un « timer » est déclenché
  - Adaptation automatique du délai d'acquittement
    - « Segment Round Trip Time »

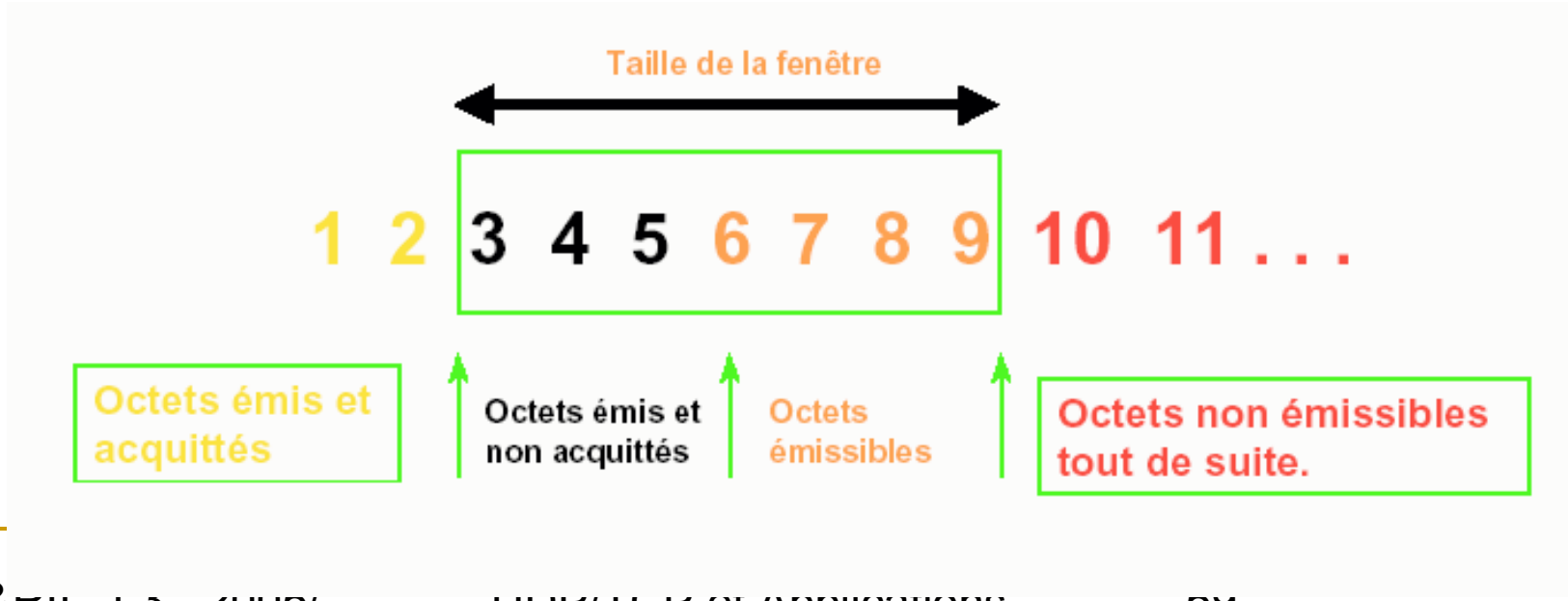
- **TCP s'adapte sans paramétrage, à tous les débit et à tous les temps de réponse, et donc à tous les réseaux.**

Retransmission adaptative

- Les paquets IP peuvent être perdus
- TCP attend des acquittements
- A l'envoi, un compteur est déclanché, en cas d'expiration avant l'acquittement les données sont retransmises
- Quelle valeur pour le timer ? **TRES DIFFICILE**
- RTT : temps d'aller retour, α coefficient de lissage
  - $RTT_i = \alpha * RTT_{i-1} - (1- \alpha) RTT_{mesurée}$
  - Timeout = β*RTT, β>1 (β=2)

Le fenêtrage

- Mécanisme de fenêtre glissante
- L'émetteur peut envoyer plusieurs paquets avant de recevoir un acquittement
- La réception d'un acquittement décale la fenêtre

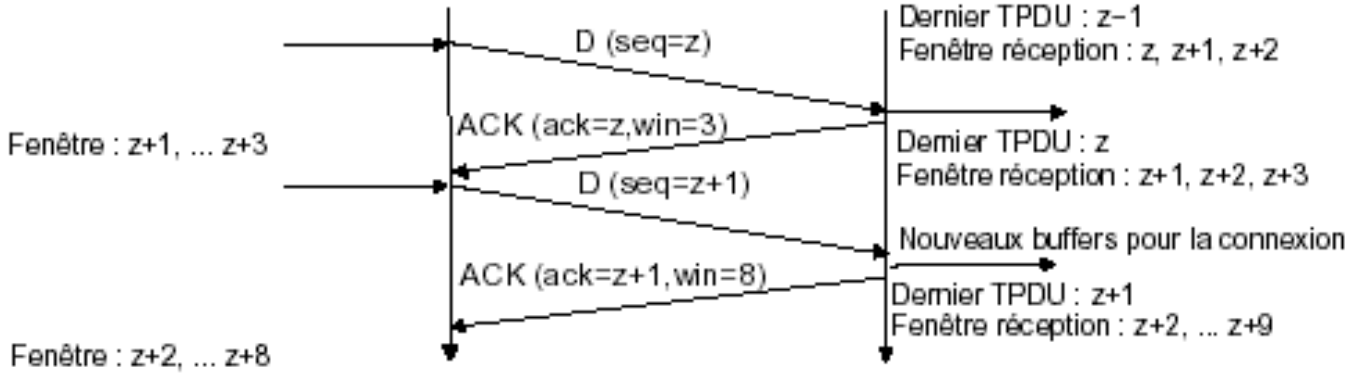


Contrôle de flux dynamique (1)

- Le destinataire dispose d'une place limitée (buffer)
- La quantité de buffers allouées à une connexion transport peut varier dynamiquement
- Le contrôle du flux et de la congestion est indispensable à Internet
  - Hétérogénéité des machines
  - Les réseaux et les routeurs ont des capacités différentes
- La perte de segments est interprétées comme un signe de congestion
  - Utilisation du « slow start » : on réduit la fenêtre de congestion

Contrôle de flux dynamique (2)

- Le destinataire dispose d'un place limitée (buffer)
- La quantité de buffers allouées à une connexion transport peut varier dynamiquement
- TCP propose un mécanisme de fenêtre glissante
- Le receveur doit annoncer la taille de buffer disponible à l'émetteur, il indique la fenêtre de réception (rwin) dans chaque TPDU d'acquittement



Contrôle de flux dynamique (3)

- Fenêtre TCP encodée dans un champ de 16 bits dans l'entête du segment TCP
- taille maximale de la fenêtre TCP "normal" 65535 bytes

- Après avoir transmis toute une fenêtre de segments, une entité TCP doit s'arrêter et attendre le retour des acquittements

Fonctionnement de TCP

- Établir des connexions
- Transférer des données
- Envoyer des acquittements
- Modifier la taille des fenêtres
- Fermer les connexions

Le contrôle de congestion (1)

- La retransmission avec mécanismes de *timer* dans TCP permet d'obtenir du contrôle de flux et ainsi d'améliorer la fiabilité des systèmes.
- Cependant TCP doit réagir aux problèmes de congestion
- Les congestions sont la résultante de délais importants causés par une surcharge de datagrammes dans un ou plusieurs nœuds du réseau (les routeurs).
- Puisque les routeurs ont une capacité de stockage finie et que les datagrammes se disputent le stockage, un routeur peut épuiser sa capacité et commencer à perdre des paquets.

Le contrôle de congestion (2)

Que fait TCP ?

- Les points terminaux des communications ne savent pas où sont les points de congestion ni la raison, car congestion = augmentation du délai.
- La plupart des protocoles de transport utilisent des mécanismes de timeout et de retransmission, ils réagissent ainsi aux extensions de délais.
- De telles retransmissions aggravent la situation et s'ils continuent à envoyer des paquets sans s'assurer de la cause, peuvent entraîner un écroulement du réseau.
- Pour éviter les écroulements du aux congestions, TCP a été conçu pour limiter automatiquement le nombre de segments qu'il transmet sur l'Internet. Il utilise deux algorithmes :
  - « Slow start » et «décroissance »

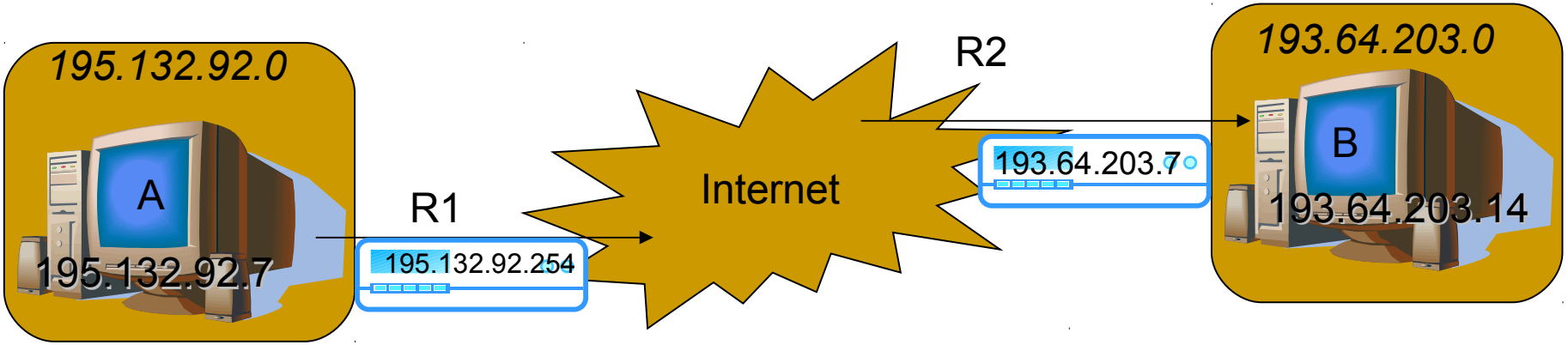
Le contrôle de congestion (3)

- TCP manipule les fenêtres pour gérer le contrôle de flux.
- La taille de ces fenêtres permet de limiter la quantité de paquets émis sur le réseau.
- Sur les réseaux filaires, les pertes de paquets dues aux erreurs de transmissions sont rares, aussi TCP assimile les pertes de paquets à des problèmes de congestion et non à des problèmes de liens à faibles performance
- Quand TCP détecte une congestion au niveau du récepteur il ajuste la taille de la ***fenêtre récepteur*** (liée à la taille du tampon de réception) pour éviter les débordements.
- Pour gérer les congestion dans le réseau, TCP utilise une deuxième fenêtre appelée la ***fenêtre de congestion***.
- **Le nombre d'octets qui peut être transmis est le minimum entre les deux fenêtres.**

La couche application

Un exemple d’utilisation de TCP/IP

Utilisation entre deux stations A1 et A2 sur deux réseaux R1 et R2



- Sur la machine A : `telnet machineB`
  - Que se passe-t-il ?
- Traduction nom->IP
  - Table hosts, cache, ou DNS, en cas d'erreur : *host unkown*
- Comment atteindre 193.64.203.14 ?
  - Pas le même réseau (!= 195.132.92.0)
  - Passage par un routeur (table de routage)
    - En cas d'erreur : *network unreachable*

Utilisation entre deux stations A1 et A2 via deux routeurs R1 et R2

- R1 reçoit la trame ethernet
  - Extrait le datagramme IP, trouve l'adresse IP du destinataire et cherche où l'envoyer
  - Via son interface sur internet et avec les protocoles de routage, le datagramme IP arrive sur R2
- R2 recherche l'adresse MAC de 193.64.203.14 (table ou broadcast ARP)
  - Envoi du datagramme à B
- B reçoit le datagramme IP
  - Extrait le segment TCP
  - Ouvre une session TCP
  - Avec l'indication du port 23 appelle le démon telnetd (via inetd sous unix)

Applications TCP/UDP

Applications UDP

Les ports UDP

- No. Port Mot-clé Description
- 7 - 0x07 ECHO Echo
- 11 - 0x0B USERS Active Users
- 13 - 0x0D DAYTIME Daytime
- 37 - 0x25 TIME Time
- 42 - 0x2A NAMESERVER Host Name Server
- 53 - 0x35 DOMAIN Domain Name Server
- 67 - 0x43 BOOTPS Boot protocol server
- 68 - 0x44 BOOTPC Boot protocol client
- 69 - 0x45 TFTP Trivial File transfert protocol
- 123 - 0x7B NTP Network Time Protocol
- 161 - 0xA1 SNMP Simple Network Management protocol

Amorçage et autoconfiguration

- Démarrage de stations « diskless »
- Configuration de machines mobiles
  - Découverte de l'adresse IP
  - Téléchargement du système
- ARP: trop proche du matériel ?
- FTP trop complexe ?

- Deux protocoles de configuration :
  - Bootp, dhcp
- Un protocole de transfert
  - tftp

Similitudes entre BOOTP et DHCP (1)

- BOOTP et DHCP : des caractéristiques en commun
- **Même structure de format pour l'échange des messages entre le serveur et les clients.**
  - Messages de demande et de réponses quasiment identiques.
  - Datagramme UDP de 576 octets pour encadrer chaque message de protocole.
  - Les en-têtes de message sont identiques pour BOOTP et DHCP à une exception près : le champ d'en-tête du message final utilisé pour transmettre les données facultatives
    - Pour BOOTP, ce champ facultatif se nomme *zone spécifique au fournisseur* et se limite à 64 octets.
    - Pour DHCP, cette zone se nomme *options* et peut transmettre jusqu'à 312 octets d'informations d'options DHCP.
- **Utilisation des ports UDP connus pour la communication client/serveur.** BOOTP comme DHCP utilisent les mêmes ports de protocole réservés pour l'envoi et la réception des messages entre les serveurs et les clients. Les serveurs BOOTP et DHCP utilisent tous deux le port **UDP 67** pour écouter et recevoir les messages de demande des clients. Les clients BOOTP et DHCP réservent généralement le port **UDP 68** à l'acceptation des réponses aux messages provenant d'un serveur BOOTP ou d'un serveur DHCP.

Similitudes entre BOOTP et DHCP (2)

- Les messages DHCP et BOOTP utilisant presque les mêmes types de formats et de structures de paquets, et généralement les mêmes ports connus de service, les programmes agents relais BOOTP et DHCP traitent généralement les messages BOOTP et DHCP comme des messages essentiellement de même type, sans faire de différence entre eux.
- **La distribution des adresses IP fait partie intégrante du service de configuration**
- BOOTP et DHCP allouent tous deux les adresses IP au démarrage, mais utilisent des méthodes d'allocation différentes :
  - BOOTP allocation fixe d'une unique adresse IP à chaque client, adresse permanente dans la base de données du serveur BOOTP.
  - DHCP allocation dynamique des adresses IP disponibles, réservant chaque adresse de client DHCP de façon temporaire

BOOTP

- Conçu avant DHCP.
- Prévu pour configurer des stations de travail sans disque avec des capacités d'amorçage limitées.
- Prend en charge un nombre limité de paramètres de configuration client appelés *extensions de fournisseur*.
- Décrit le processus de configuration de démarrage en deux phases suivant :
  - Les clients contactent les serveurs BOOTP pour effectuer une détermination d'adresse et une sélection de nom de fichier d'amorçage.
  - Les clients contactent les serveurs TFTP (Trivial File Transfer Protocol) pour effectuer le transfert de leur image de démarrage.
- Les clients BOOTP ne relient pas ou ne renouvellent pas la configuration avec le serveur BOOTP sauf au redémarrage du système.

DHCP

- Conçu après BOOTP.
- Conçu pour configurer des ordinateurs en réseau fréquemment relocalisés (tels que des portables) ayant des lecteurs de disque locaux et l'intégralité des capacités d'amorçage.
- Prend un charge un nombre plus important et extensible de paramètres de configuration client appelés *options*.
- Décrit un processus de configuration de démarrage en une phase au moyen duquel un client DHCP négocie avec un serveur DHCP pour déterminer son adresse IP et obtenir tous les autres détails de configuration initiale nécessaires au fonctionnement du réseau.
- Les clients DHCP n'ont pas besoin d'un redémarrage du système pour relier ou renouveler la configuration avec le serveur DHCP. Ils peuvent entrer automatiquement en condition de reliaison à intervalles de temps fixés pour renouveler leur allocation de bail d'adresse avec le serveur DHCP. Ce processus s'effectue en arrière-plan en totale transparence pour l'utilisateur.

DHCP - RFC 1533 et 1534

- extension de BOOTP (RFC 1532)
- gère l'attribution des informations de configuration IP en affectant automatiquement les adresses IP
- Fonctionnement:
  - Demande de bail IP (DHCPDISCOVER) avec adresse IP source 0.0.0.0 et adresse IP destination 255.255.255.255 et adresse MAC
  - Proposition de bail IP (DHCPOFFER) les serveurs DHCP disposant d'adresses valides envoient une proposition au client avec une durée de bail et l'adresse IP du serveur DHCP
  - Sélection de bail IP (DHCPREQUEST) : le client sélectionne les informations de la première proposition reçue et diffuse une demande de location de l'adresse
  - Accusé de réception (DHCPACK) : le serveur répond au message, les autres serveurs retirent leur proposition.

TFTP (Trivial File Transfer Protocol)

- Utilise UDP
- Pas de contrôle d'accès
  - Problème de sécurité
  - Accès à un nombre de fichiers restreints
- Utilisé pour charger en mémoire le système dans des matériels sans mémoire de masse

Gestion de périphériques réseaux : SNMP

- Surveillance et administration de routeurs et d'ordinateurs
- Simple Network Management Protocol v1
- Publication en 1988
- Gestion des bases de données MIB
- Trois types d'opération
  - lecture (GetRequest, GetNextRequest)
  - écriture (SetRequest)
  - rapport (Trap)
- SNMP v2 (1993)
  - Deux nouvelles opérations :
    - GetBulk (gros blocs de données)
    - Inform (envoi d'un Trap vers un autre NMS)
    - Un agent SNMPv2 peut agir comme un proxy pour un agent SNMPv1
- SNMP v3
  - Les améliorations :Authentification par clefs, Cryptage des données, Contrôle d'accès aux informations

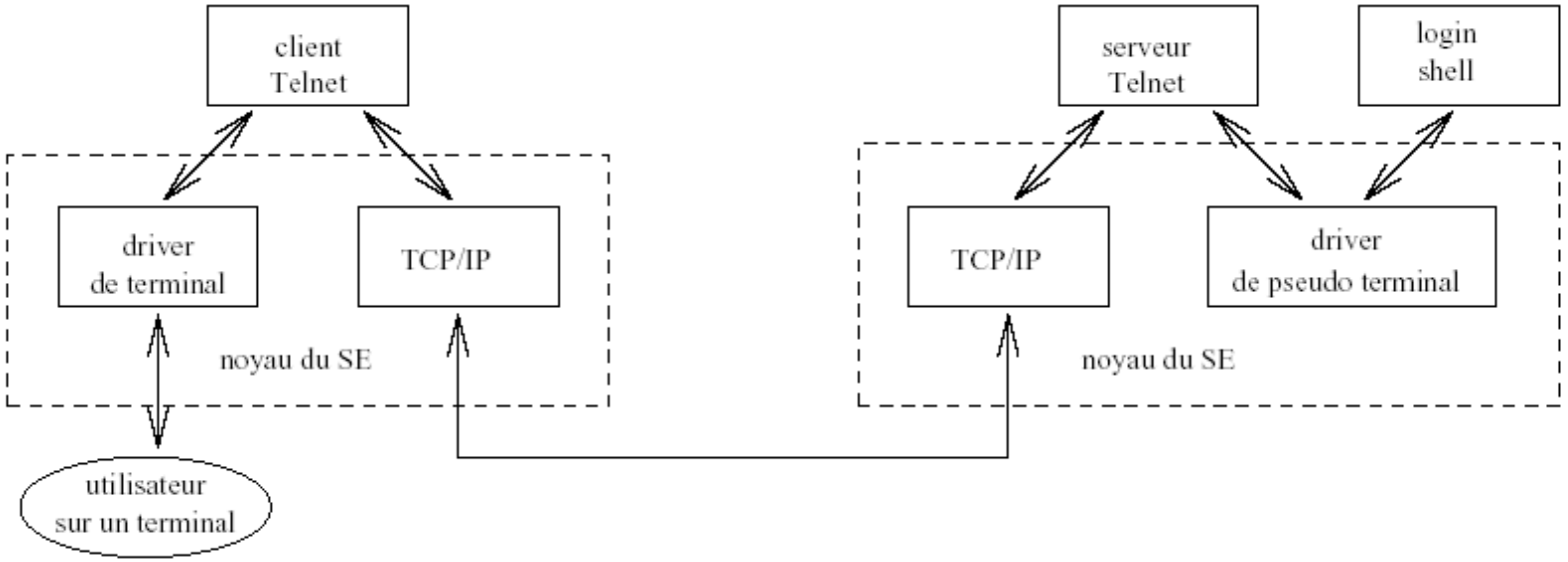
Applications TCP

Connexion à distance

Telnet (RFC 854)

- Telnet (Telecommunications Network)
- Port TCP 23.
- . Accès distant à l'invite de commande d'une machine en mode texte. (e.g. bash sous UNIX).
- Une machine disposant d'un serveur telnet permettra donc à n'importe quelle machine de part le réseau de s'y connecter, au moyen d'un client telnet. Les clients telnet existent sur la quasi-totalité des plates-formes (Windows, Unix, MacOS, BeOS…).

Schéma de fonctionnement de Telnet



Telnet

- Vérification de l'identité du client
  - Login + mot de passe, droit d'accès
- La connexion est ouverte
  - Échanges composés d'une suite d'octets
  - Caractères sur 8bits
  - Émulation de terminal
- Les commandes
  - Marquée par le caractère 255 (IAC. Interpret As Command)

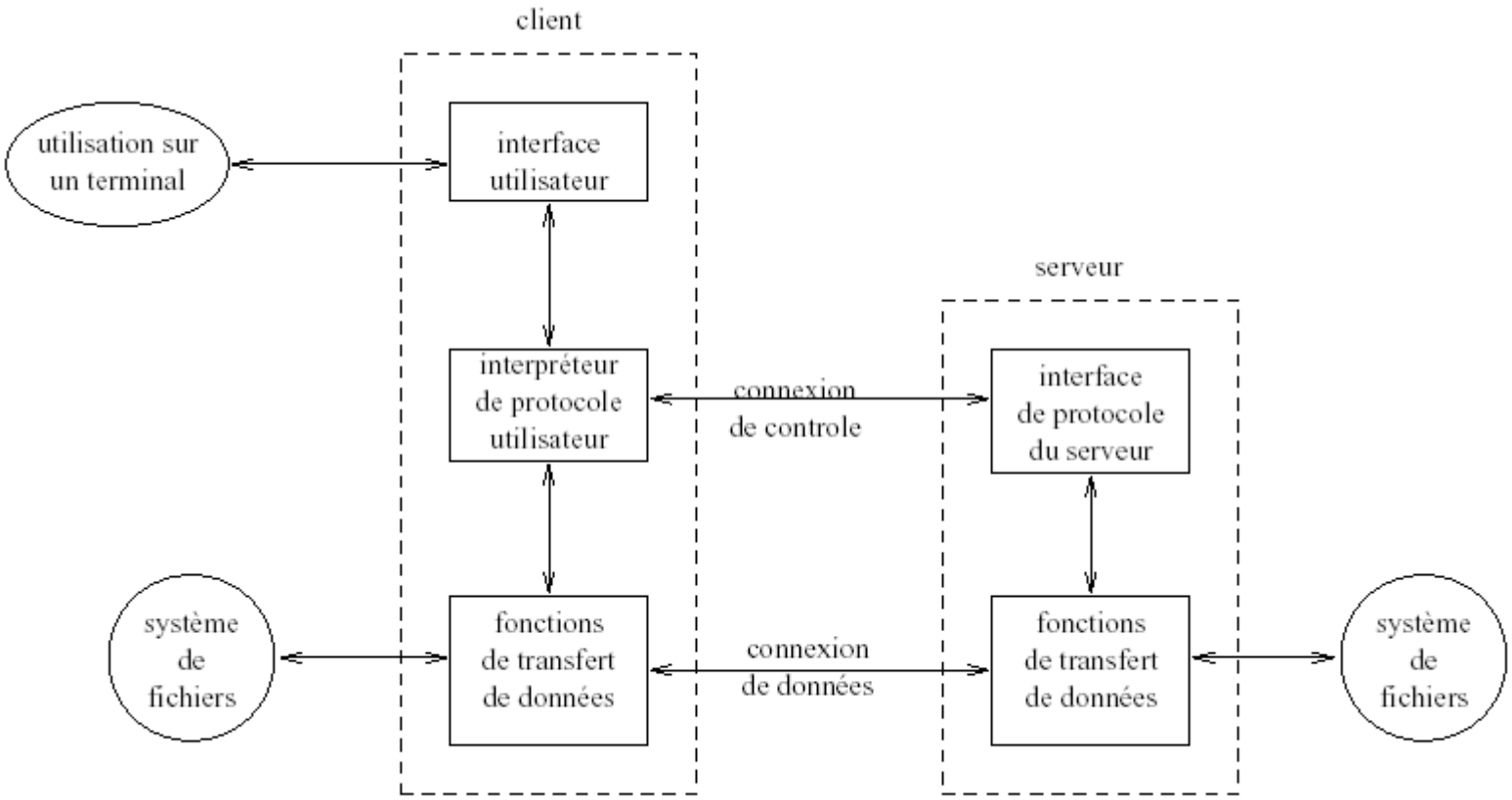
Accès et transfert de fichiers

- Des contextes d'utilisation très variés :
  - Serveurs de fichiers centralisés
  - Archivage à distance
  - Partage de fichiers entre systèmes
- Partages en ligne
  - Accès en ligne
  - Duplication de fichiers
- Partage par transfert de fichiers

FTP (File transfert Protocol)

- RFC 959.
- Transfert de fichiers d'une machine à une autre.
- Ports TCP 20 (données) et 21 (contrôle).
- Deux modes
  - Client : par exemple un processus d'un utilisateur
  - Serveur: ex. démon ftpd lancé sous unix (via inetd)
- Fonctionnement
  - Le client ouvre la connexion
  - Le serveur vérifie l'identité du client
    - Mot de passe, droit d'accès, accès anonyme
  - Commandes
    - Suite de caractères simples terminées par CRLF (comme telnet mais vers le port 21)

Transfert par FTP



FTP (2)

- Le serveur répond
  - Nombre de trois caractères
    - 1er chiffre : à quoi se rapporte la réponse
      - 1?? La commande commence a être exécutée, il va y avoir une autre réponse
      - 2?? La commande a été exécutée avec succès, un autre envoi est possible
      - 5?? Commande non acceptée.
- Commandes
  - Help, status, open, user, passwd, ls, cd, get put, type, delete, quit

Les commandes FTP

```
Commands may be abbreviated. Commands are:
! debug mdir sendport site
$ dir mget put size
account disconnect mkdir pwd status
append exit mls quit struct
ascii form mode quote system
bell get modtime recv sunique
binary glob mput reget tenex
bye hash newer rstatus tick
case help nmap rhelp trace
cd idle nlist rename type
cdup image ntrans reset user
chmod lcd open restart umask
close ls prompt rmdir verbose
cr macdef passive runique ?
delete mdelete proxy send
```

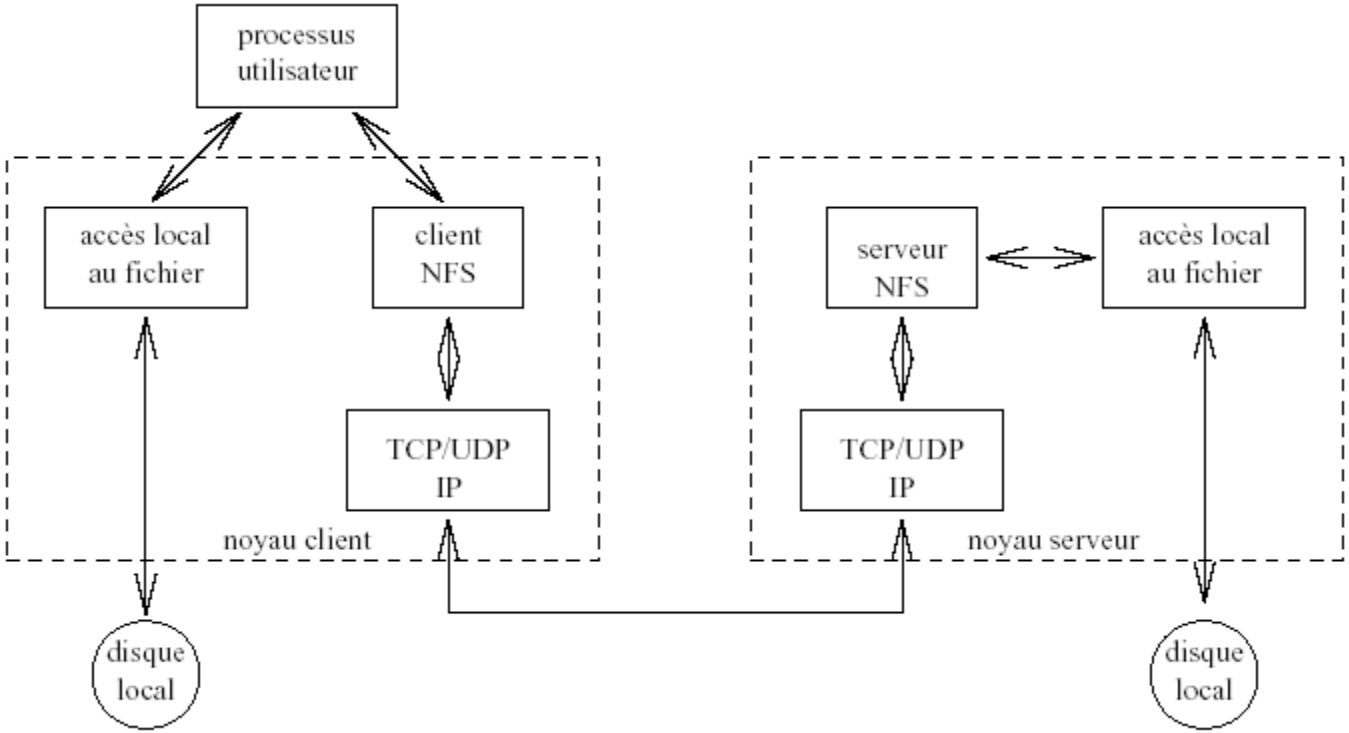
FTP (3)

- FTP est utile dès qu'il s'agit de transférer des données entre deux machines A et B.
- Comme en telnet, la machine A doit être équipée d'un client ftp, alors que la machine B est elle équipée d'un serveur FTP.
- Connexions anonymes ou non
- Le protocole TCP utilise par convention le port TCP/21 pour les commandes, et le port TCP/2 pour les données.
  - Le port TCP/21 est appelé l'interpréteur de protocole (Protocol Interpreter ou PI)
  - le port TCP/20 est appelé processus de transfert de données (data transfert process ou DTP).

NFS (network file system)

- Système de fichiers en réseau développé par Sun
- Accès partagé et transparent
- Construit avec trois briques :
  - Le protocole NFS
  - Un mécanisme d'appel de procédures distantes (RPC)
  - Représentation de données XDR (external data representation)

Schéma de fonctionnement de NFS



NFS (network file system) (2)

- L'utilisation de NFS est transparente:
  - Une fois installé, les programmes accèdent aux fichiers distants en utilisant les même opérations que pour les fichiers locaux
- RPC et XDR sont utilisables par les programmeurs pour développer des applications client/serveur

Conclusion sur le transfert de fichiers

- L'accès aux fichiers distants prend deux formes :
  - Recopie intégrale
  - Partage en ligne
- FTP est LE protocole de transfert de fichier le plus important de la famille TCP/IP
- TFTP est un protocole plus simple qui s'appuie sur UDP

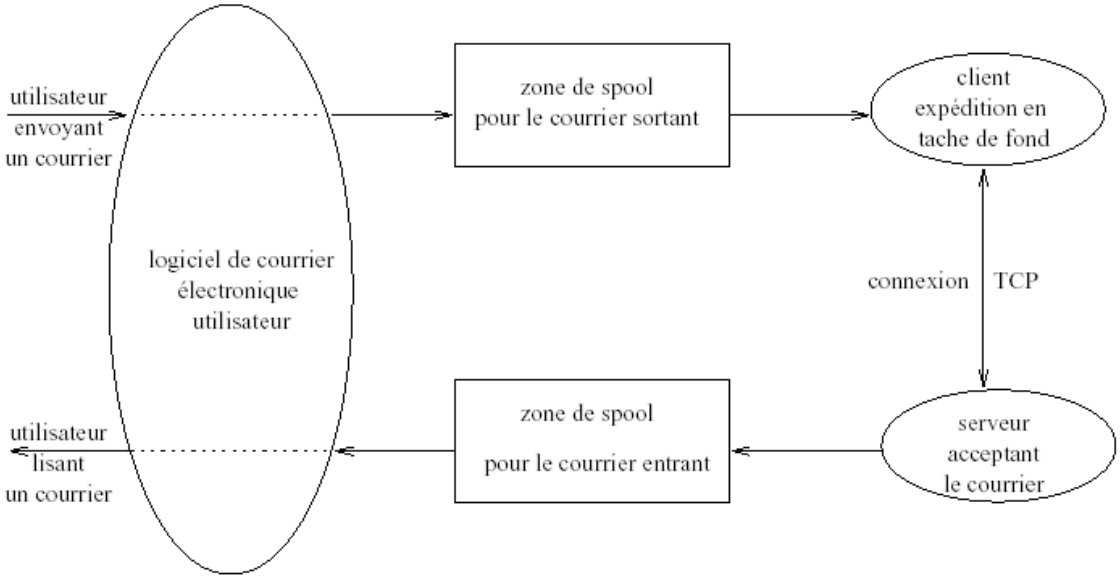
Le courrier électronique

- Application la plus utilisée sur Internet
  - Trois protocoles
    - Envoi du courrier
    - Réception des messages
    - Gestion du dossier distant

SMTP (1)

- Simple Mail Transfert Protocol (RFC 821)
- Service d'envoi de courriers électroniques.
- Port TCP 25.
- Similaire au protocole FTP,(langage de commande)
- Sur système Unix :
  - sendmail : client et serveur.

Messagerie SMTP



- SMTP utilise des files d'attente pour gérer les transferts de courrier
- Lorsqu'un message est envoyé au serveur SMTP,
  - celui-ci le place dans une file d'attente,
  - puis tente de le livrer à la machine de destination.
  - Si cette machine n'est pas accessible, transmission ultérieure.
  - Tous les messages sont transférés dans un format ascii (codage sur 7 bits)
- La fin d'un message est indiquée par un '.' sur une ligne vierge.

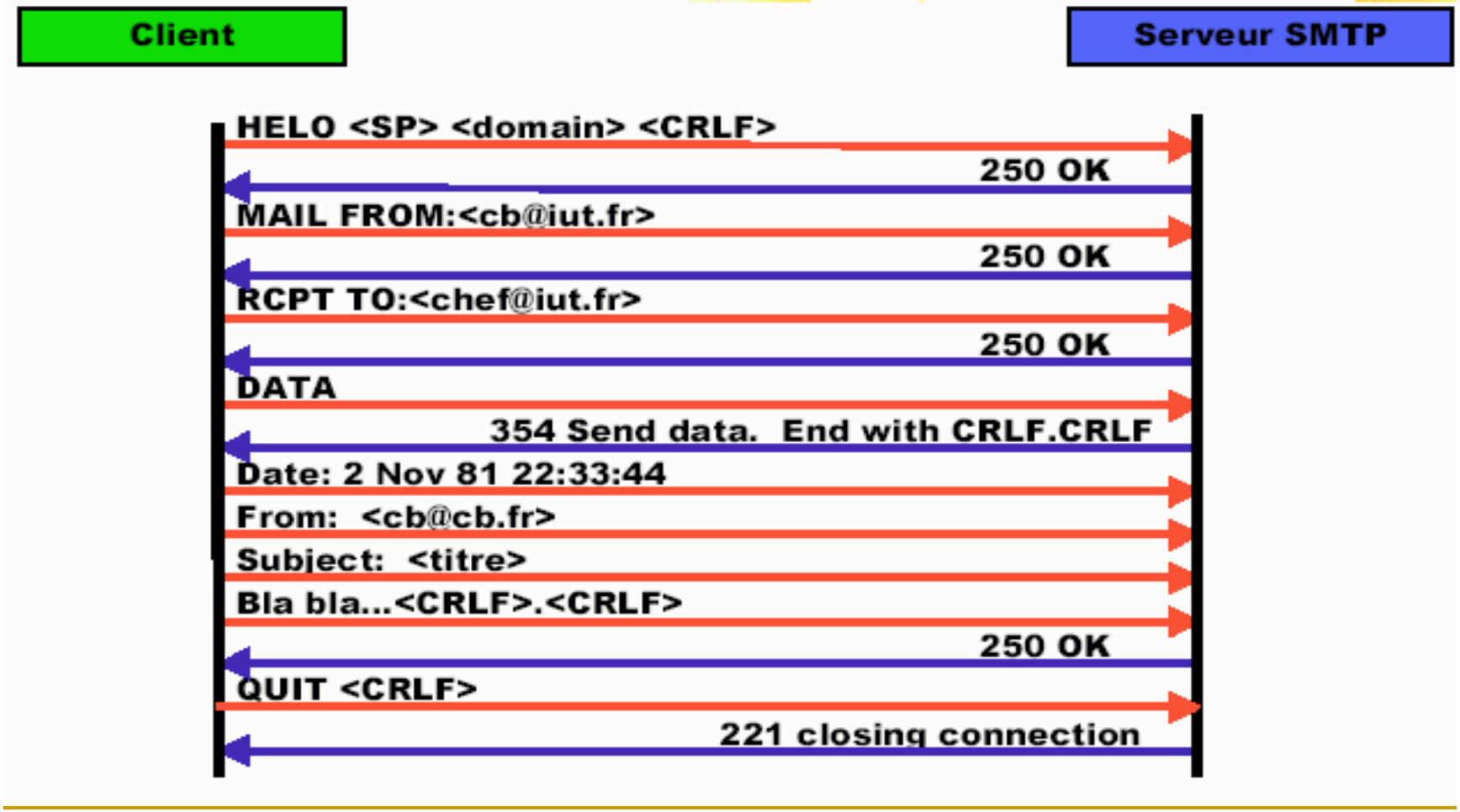
SMTP (2)

- Première phase est l'**authentification de la machine émettrice**.
- Émetteur : envoi la commande *HELO* suivi de son nom de domaine.
- Récepteur : message de bienvenue, liste les commande disponibles.
- Émetteur : donne le nom de l'expéditeur *MAIL FROM: login login*.
- Émetteur : destinataire : *RCPT TO: login login*.

- Les machines sont prêtes à échanger les messages.
- Émetteur : *DATA data…*, puis termine cette phase de transfert du message en envoyant un point sur une ligne vierge.
- La connexion reste alors établie les deux machines peuvent continuer à transférer des courriers, ou retourner leur mode de connexion (celle qui émettait devient réceptrice, et celle qui recevait devient émettrice).

- Si plusieurs destinataires sont spécifiées dans le champ RCPT RCPT, le message est alors envoyé à tous les destinataires, mais il n'est transféré qu'une fois entre les deux serveurs.

Un échange SMTP



MIME

- Multipurpose Internet Mail Extension
- Extensions pour permettre, principalement aux e-mails, de transporter autre chose que du texte
  - son, des images, de la vidéo
  - la messagerie n'est *a priori* pas faite pour cela
- Ces extensions servent également sur le Web, lorsque l'on utilise HTTP pour transporter autre chose que du texte (ce qui est souvent le cas)
- MIME rassemble deux choses distinctes :
  - Une description normalisée d'un type de document (non texte).
  - Le mode de codage employé pour le transporter.

MIME et SMTP

- « juste un texte légèrement accentué... *Suivi d'une image gif*. »

Return-Path: <test@free.fr>
...
From: « Test » < test@free.fr >
To: < test@free.fr >
Subject: demo MIME Date: Sat, 9 Nov 2002 11:29:09 +0100
MIME-Version: 1.0
Content-Type: multipart/mixed; *On est averti qu'il y aura plusieurs morceaux de type différents...*
boundary="----=_NextPart_000_0044_01C287E3.38B13A20" *Avec un séparateur bien défini.*
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 6.00.2800.1106
X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2800.1106

This is a multi-part message in MIME format.

MIME et SMTP (3)

------=_NextPart_000_0044_01C287E3.38B13A20
Content-Type: text/plain;
charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
juste un texte l=E9g=E8rement accentu=E9... Suivi d'une image gif.

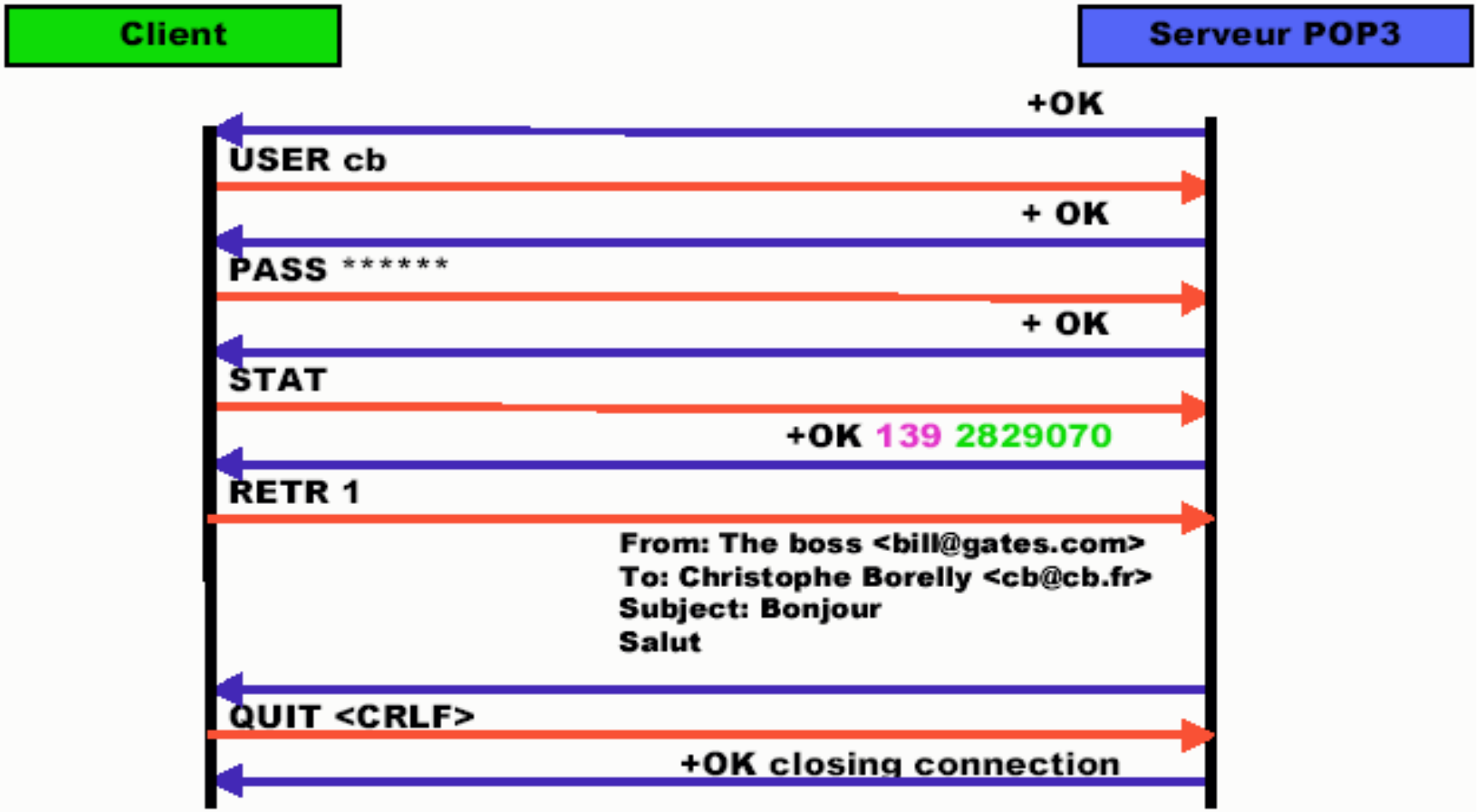
MIME et SMTP (2)

------=_NextPart_000_0044_01C287E3.38B13A20
Content-Type: application/octet-stream; name="moineau1.gif"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename=« test.png »
R0lGODlhcgH8APf/AP//////zP//mf//Zv//M///AP/M///MzP/Mmf/MZv/M
M//MAP+Z//+ZzP+Z ...
GZACDvqwAvWAOgEBADs=
-----=_NextPart_000_0044_01C287E3.38B13A20--

POP3

- Post Office Protocol.
- RFC 1939.
- Port TCP 110.
- Service de lecture à distance des messages électroniques.

Echange POP3



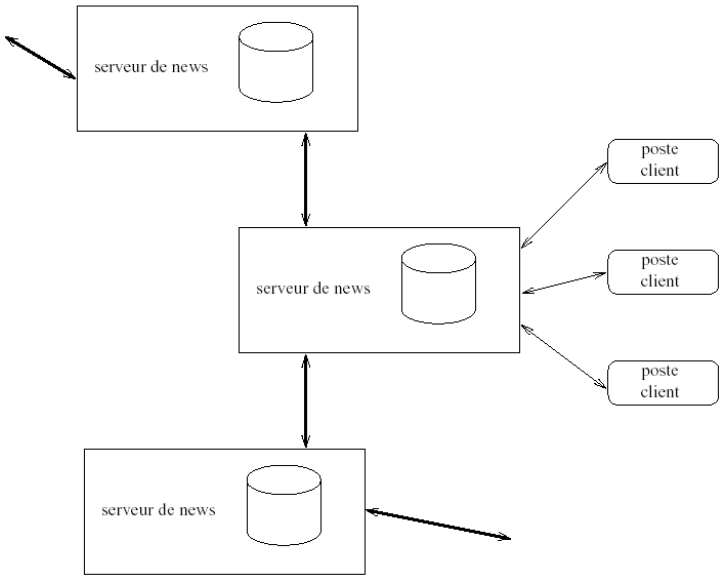
IMAP (Internet Message Access Protocol)

- Compatible avec les standards de messagerie (MIME)
- Accès aux messages depuis plusieurs machines
- Supporte les modes
  - En-ligne, hors-ligne et déconnecté
- Gestion de la concurrence
- Indépendance client/format de stockage

- Cf. http://www.imap.org/papers/biblio.html

Les news : NNTP (Network News Transfert Protocol)

- Échange de news ou forum de discussion
- Formation d'un réseau logique d'échange:
  - Usenet
- Échange serveurs-serveurs et serveurs-clients
- Système de relais :
  - **Un** client poste **un** message sur **un** serveur
  - Le serveur le transmet à d'autres serveurs
- TCP port 119



Et la sécurité ?

SSH

- Connexion sécurisée
- Les transferts sont cryptés
- "Empreinte digitale" des machines
  - Une machine ne peut pas se faire passer pour une autre
- Authentification à l'aide de paires de clefs RSA
  - les clefs sont fabriquées par 2 : publique, privée
- Principe : ce qui est codé par la clef publique ne peut être décodé que par la clef privée, et vice-versa
  - la clef privée est secrète : en principe elle ne doit jamais circuler sur le réseau
  - La clef publique est (doit être) librement accessible : elle sert soit à coder, soit à décoder les données sensibles

Principe de SSH

- Le serveur envoie sa clé publique au client.
- Le client génère une clé secrète et l'envoie au serveur, en cryptant l'échange avec la clé publique du serveur (cryptographique asymétrique). Le serveur décrypte la clé secrète en utilisant sa clé privée, ce qui prouve qu'il est bien le vrai serveur.
- Pour le prouver au client, il crypte un message standard avec la clé secrète et l'envoie au client. Si le client retrouve le message standard en utilisant la clé secrète, il a la preuve que le serveur est bien le vrai serveur.
- Une fois la clé secrète échangée, le client et le serveur peuvent alors établir un canal sécurisé grâce à la clé secrète commune (cryptographie symétrique).
- Une fois que le canal sécurisé est en place, le client va pouvoir envoyer au serveur le login et le mot de passe de l'utilisateur pour vérification. La canal sécurisé reste en place jusqu'à ce que l'utilisateur se déloggue.

Utilisation de ssh

- Les commandes ssh
- ssh host [–l login]
  - Configure automatiquement le display ! (création d'un tunnel avec –x ou paramétrage par défaut)
- scp host:fichier host:fichier
  - scp machine1:tmp/mon_fichier ./toto/titi
  - scp mon_fichier machine2:toto/titi
- ssh-agent : mémorise les clefs privées
  - pas obligatoire, mais évite de ressaisir tout le temps la passphrase
- ssh-add : enregistre une nouvelle clef privée auprès de l'agent
- sftp : équivalent ftp
- ssh-keygen : fabrique des paires de clefs

Création des clefs SSH

- ssh-keygen –t type
  - type = 'rsa1' (SSH V1), 'rsa' (SSH v2) ou 'dsa‘ (SSH v2)
- Chaque type produit 2 fichiers, en ~/.ssh/
  - rsa1 : identity (privée) + identity.pub (publique)
  - rsa2 : id_rsa + id_rsa.pub
  - dsa : id_dsa + id_dsa.pub
- demande une "passphrase"
  - Les clefs privées sont stockées sous forme codée
  - il est conseillé de choisir quelque chose de long (> 10 caractère) et compliqué …

Installation des clefs SSH

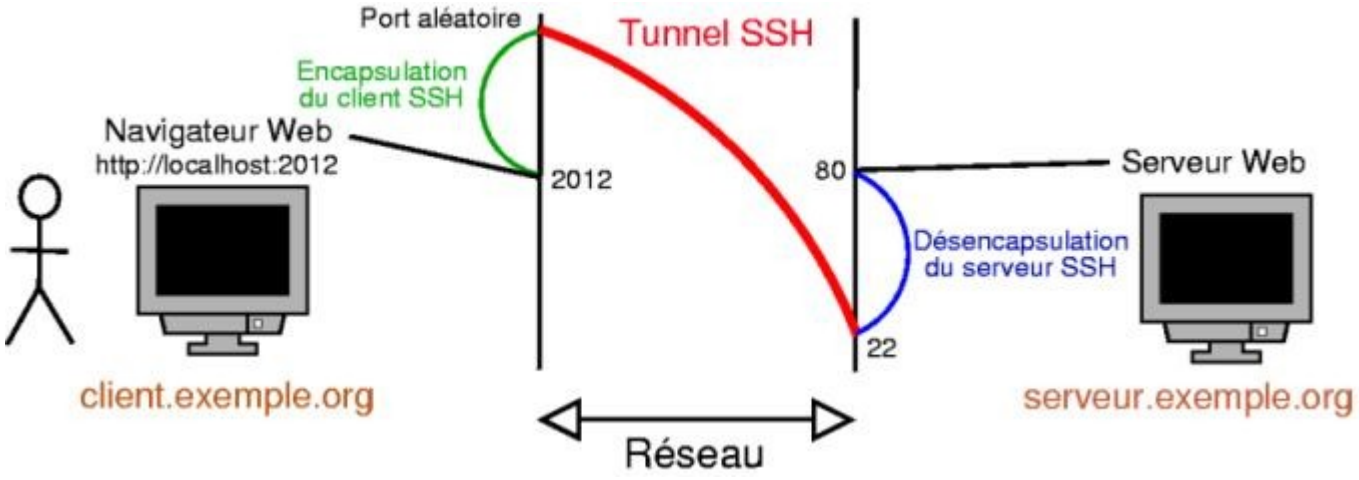
- Sur les machines où l'on souhaite se connecter :
- La (les) clefs publiques doivent être stockées dans le fichier ~/.ssh/authorized_keys
- Exemple : pour autoriser l'accès à machineA depuis machineB (type rsa2)
  - Sur machineB : ssh_keygen –t rsa (si besoin)
  - copier le contenu du fichier machineB:~/.ssh/id_rsa.pub **à la fin** du fichier machineA:~/.ssh/authorized_keys
- **Attention : à ne pas écraser machineA:~/.ssh/id_rsa.pub !**

Utilisation de l'agent SSH

- Lors du login (ex : .zlogin)
- Lancer ssh-agent
- SSH-AGENT affiche un script :
  - Ce script définit des variables d'environnement
    - ~> ssh_agent
    - SSH_AUTH_SOCK=… ; export SSH_AUTH_SOCK
    - SSH_AGENT_PID=xxxx ; export SSH_AGENT_PID
  - Les shells qui exécutent ce script (.zshrc) savent ensuite comment contacter l'agent pour utiliser ses services
- Lancer ssh-add pour enregistrer les clefs secrètes
- Dans chaque nouveau shell (.zshrc)
- Exécuter le script (il faut l'avoir sauvé qq-part !)

Les tunnels SSH 1/2

- Faire un tunnel SSH est un moyen simple de crypter n'importe quelle communication TCP entre votre machine et une machine sur laquelle vous avez un accès SSH.



Les tunnels SSH 2/2

- Par exemple, pour établir un tunnel SSH pour une connexion HTTP vers la machine *serveur.exemple.org* :
- % **ssh -L 2012:serveur.exemple.org:80 toto@serveur.exemple.org**
  - où *2012* est le port sur la machine cliente à partir duquel la connexion entre dans le tunnel SSH (le port doit être supérieur à 1024 si on ne veut pas avoir à lançer le tunnel en tant que *root*).
- Ensuite, il suffit de lançer un navigateur Web en lui demandant de se connecter en local sur ce port :
  - % **lynx http://localhost:2012**

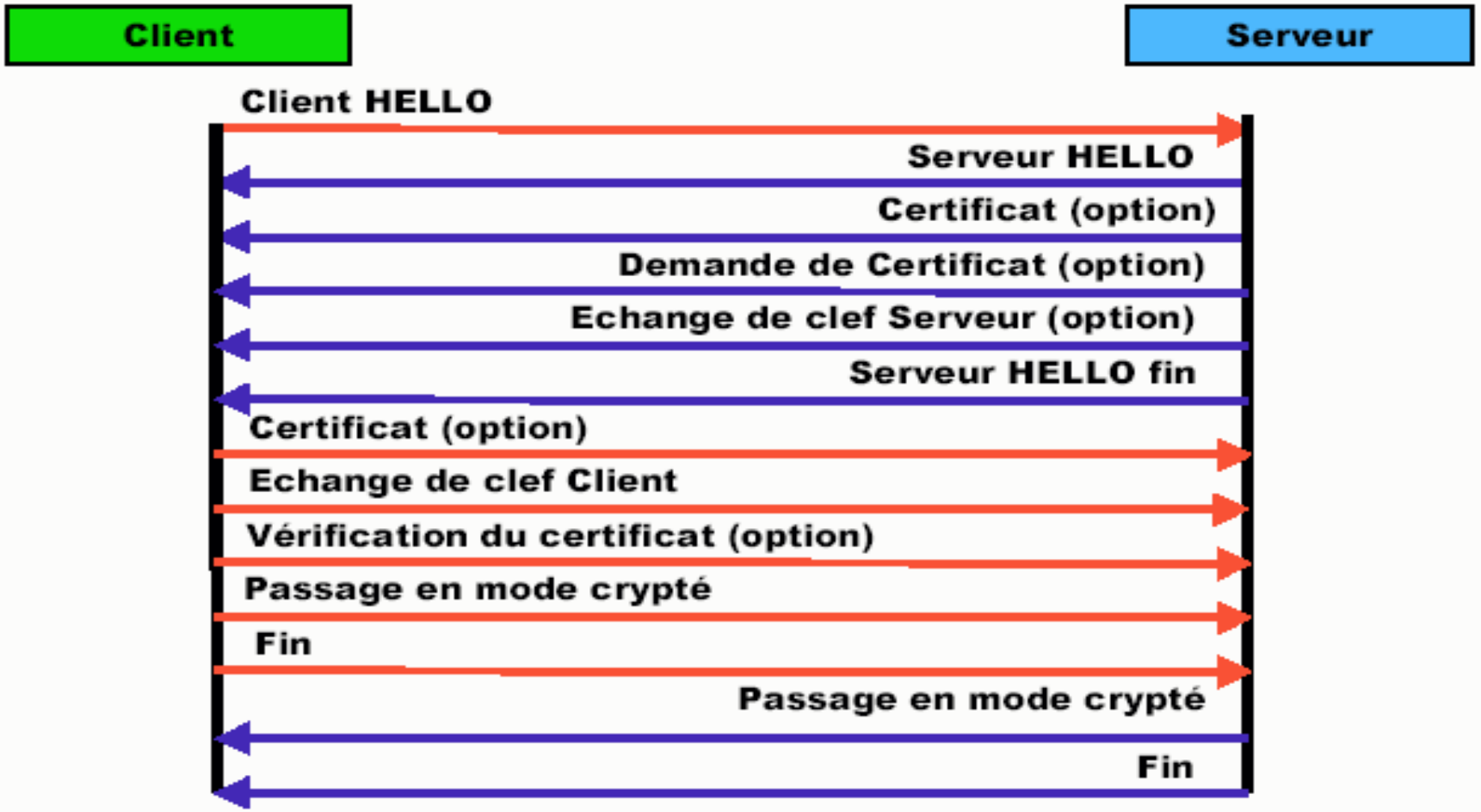
SSL/TLS

- Secure Socket Layer (Netscape en 1994).
- Couche de chiffrement pour la confidentialité des échanges de données.
- Protocole HTTPS (TCP/UDP 443).
- http://sitesearch.netscape.com/eng/ssl3/
- TLS : Transport Layer Security depuis janvier 1999

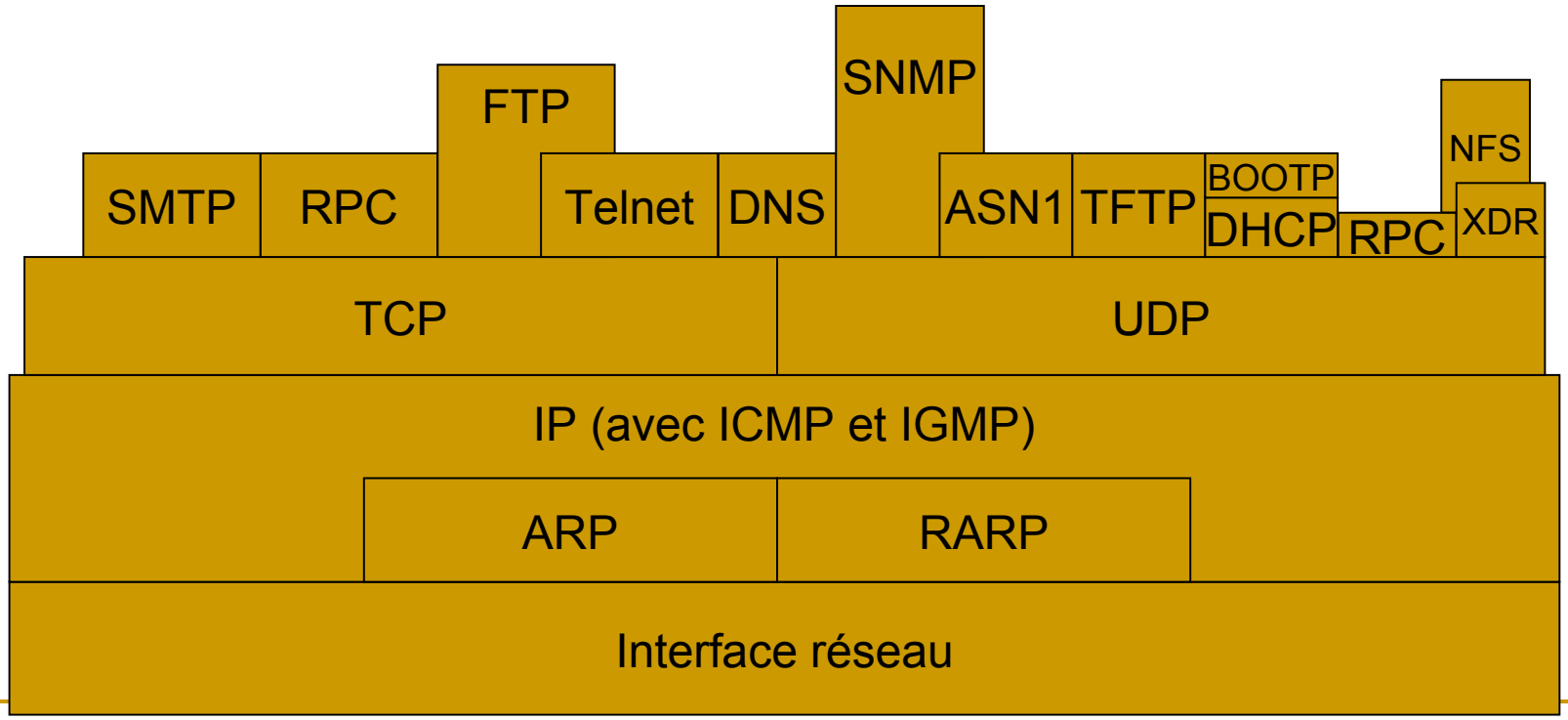
Fonctionnement de SSL

- SSL utilise un système à clef publique
- pour l'authentification (certificats).
- La signature des données et leur intégrité est réalisée avec un système à clef secrète.

Message SSL


Client
Serveur
Client HELLO
Serveur HELLO
Certificat (option)
Demande de Certificat (option)
Echange de clef Serveur (option)
Serveur HELLO fin
Certificat (option)
Echange de clef Client
Vérification du certificat (option)
Passage en mode crypté
Fin
Passage en mode crypté
Fin

Dépendances entre protocoles



Le Web

Généralités

- Origine
  - CERN, 1989, projet World Wide Web
  - Explosion en 1993 avec l'apparition du premier navigateur : Mosaïc
- Le web, c'est quoi donc ?
  - Système d'information universel
    - hypertexte : documents reliés entre eux par des *liens* accessibles à partir d'*ancres*
    - multimédia (hypermédia) : composé de différents média (textes, sons, images, vidéo, ...)
    - sur Internet
  - Accès gratuit (?) et facile

La « gestion » du Web

- Le World Wide Web Consortium (W3C)
  - Nombreux organismes
    - privés (Microsoft, Netscape, Sun, IBM, …)
    - publics (INRIA, MIT, …)
  - Objectifs :
    - développement et promotion du web
    - travaux de standardisation (HTML, HTTP...), développement, …
  - http://www.w3c.org

Les grands principes

- Modèle client / serveur
  - Le client demande un document
  - Le serveur fournit (ou non) le document
- Protocole utilisé entre clients et serveurs web :

  Hyper**T**ext **T**ransfert **P**rotocol (HTTP)
- Langage de définition de document :

  Hyper**T**ext **M**arkup **L**anguage (HTML)
- Schéma de nommage des ressources :
  - **U**niform **R**essource **L**ocator (URL)
  - **U**niform **R**essource **I**dentifier (URI)

URL

- Format d'une URL
- **<protocole>://[user[:password]@]<machine>[:port][/<path>[#label|?liste paramètres>]]**
  - <protocole> : méthode d'accès au document
  - <machine> : adresse de la machine
  - <port> : numéro de port sur la machine
  - <path> : chemin d'accès au document sur la machine
- Exemples d'URL
  - http://www.univ-tln.fr/
  - http://www.paris.org:80/Monuments/Eiffel/info.html

Format d'URL

- protocole**://**infosUtilisateur**@**nomDeLaMachine**:**port**/**chemin/Dacces/Document**#**reference
- protocole**://**infosUtilisateur**@**nomDeLaMachine**:**port**/**chemin/Dacces/Document**?**requete

- Exemples:
  - http://localhost:82/manual/index.html
  - http://pccb/cgi-bin/odbc.form.php3?sql=SELECT+*+FROM+Livres
  - ftp://cb@localhost:2100/img/

Protocole HTTP et Fonctionnement d’un serveur WEB

HTTP

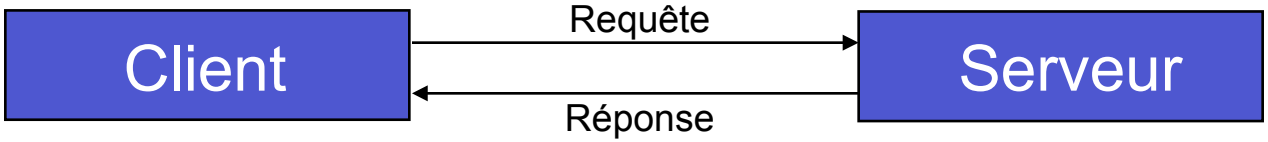
- **H**yper**T**ext **T**ransfert **P**rotocol
  - Version actuelle : HTTP 1.1
    Spécification décrite dans le RFC 2616.
  - Protocole client-serveur :
    - le navigateur demande une page HTML (requête)
    - le serveur répond à cette demande (réponse).
  - Ce protocole est sans état (en anglais, stateless) :

    le serveur traite les demandes indépendamment.

HTTP

- Dialogue entre un client et un serveur web
  1 - Le client demande un document via une URL
    - établissement de la connexion avec le serveur spécifié dans l'URL
    - demande du document sur le serveur

  2 - le serveur traite la demande
    - il envoie le document ou un message d'erreur au client
    - puis ferme la connexion



- Remarque
  - Chaque élément d'un document composé provoque ce type de dialogue (image dans une page, …).

HTTP... et ensuite ?

- Traitement du document
  - Réalisé par le client (navigateur)
  - HTML
    - mise en forme du texte
    - insertion d'images, de sons, d 'animations, etc.
    - spécification de liens hypertextes :
      - à travers les URL
      - un navigateur traite spécifiquement les liens : déclenchement de l'accès au document lorsque le lien est activé
    - documents non supportés par le client :
      - appel de visualisateurs externes

Requête HTTP

- **GET / HTTP/1.0**
- **Accept**: image/gif,image/xxbitmap,
- image/jpeg,image/pjpeg,application/vnd.msexcel,
- application/msword,application/vnd.mspowerpoint,*/*
- **Accept-Language**: fr
- **User-Agent**: Mozilla/4.0 (compatible; MSIE 5.0;
- Windows NT; DigExt)
- **Host**: localhost:8081
- **Connection**: Keep-Alive

Réponse du serveur HTTP

HTTP/1.1 200 OK
Date: Tue, 16 Jan 2001 14:54:44 GMT
Server: Apache/1.2.6
Last-Modified: Tue, 16 Jan 2001 14:34:25 GMT
ETag: "642fa-1ea3-3a645bf1"
Content-Length: 7843
Accept-Ranges: bytes
Connection: close
Content-Type: text/html
<HTML>
<BODY>

Serveur Web : Apache

- Apache est un serveur Web libre
- Standard comme serveur Web sous linux
- Le serveur Web le plus utilisé sur Internet (60% des sites d'Internet, contre environ 20% pour IIS)
- Stable et performant
- Disponible :
  - Source : Compilable pour l'adapter à ses besoins
  - Binaire : Version standard

- Configuration via **httpd.conf**, fichier de configuration d'apache.
- Anciennes versions trois fichiers : httpd.conf, srm.conf, access.conf
- Nous ne traitons ici que de la **configuration,** les distributions intègrent généralement une version que l'on peut étendre via les modules « DSO »

PROXY

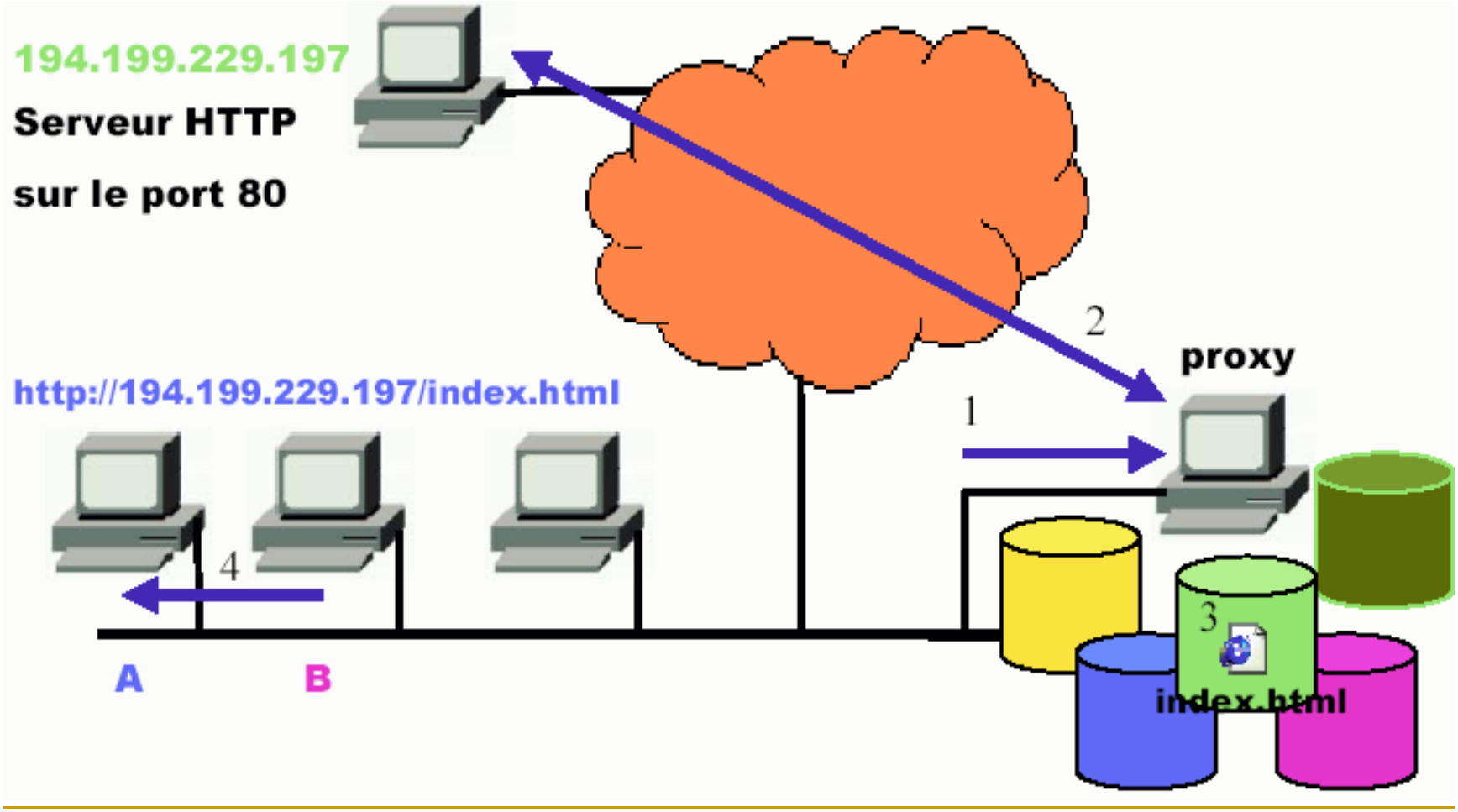
- Qu'est-ce donc ?
  - Intermédiaire entre clients et serveurs web

- À quoi ça sert ?
  - À la sécurité : nécessaire avec les gardes-barrière (fire-wall).
  - À la communication : un proxy permet d'accéder à des serveurs dont les clients web ne supportent pas le protocole (WAIS par exemple).
  - À réduire le trafic réseau : le proxy est généralement couplé à un cache qui stocke certaines données.
    Cela permet d'optimiser l'utilisation du réseau et de diminuer les temps de réponse des requêtes

Proxy HTTP

- Serveur de redirection.
- L'accès aux pages WEB de l'internet peut prendre du temps suivant l'heure de la journée (encombrements).
- Pour augmenter la rapidité d'accès des machines d'un LAN, on peut ajouter un serveur proxy qui va stocker les documents les plus demandés.
- Les requêtes mises en « cache » seront immédiatement renvoyées au client sans avoir à utiliser internet.

Exemple de proxy



CGI

- Common Gateway Interface.
- Génération dynamique de page HTML.
- 2 modes d'envoi des données (POST et GET).
- Tout type d'applications :
  - Scripts en shell UNIX.
  - Application en C, PERL, etc...
  - Scripts ASP (Active Server Pages).
  - Scripts PHP (Hypertext PreProcessor).
  - Servlet en JAVA.

Services d’annuaires

NIS (YP)

LDAP

Le Network Information System

- NIS ou YP (Yellow Pages) développé par Sun
- Gestion centralisée de fichiers communs à plusieurs machines (~système de base de données répartie) :
  - /etc/passwd, /etc/group, /etc/shadow, /etc/hosts, /etc/services, /etc/protocols, …
- Un serveur maître
  - programme *ypserv*
  - éventuellement des serveurs esclaves (en cas de panne)
- Des clients interrogeant les serveurs
  - programme *ypbind*
  - les serveurs sont aussi clients

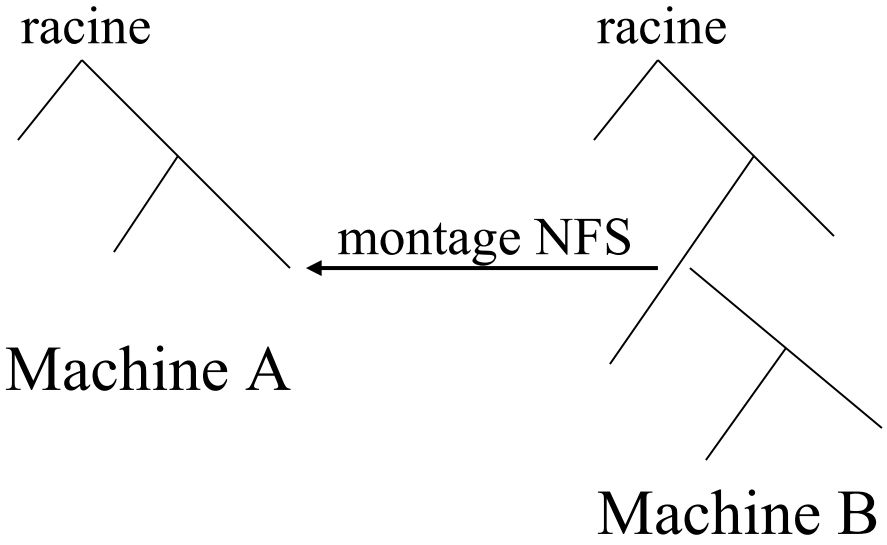
LDAP

- LDAP est un protocole d'annuaire standard et extensible. Il fournit :
  - le *protocole* permettant d'accéder à l'information contenue dans l'annuaire,
  - un *modèle d'information* définissant le type de données contenues dans l'annuaire,
  - un *modèle de nommage* définissant comment l'information est organisée et référencée,
  - un *modèle fonctionnel* qui définit comment on accède à l'information ,
  - un *modèle de sécurité* qui définit comment données et accès sont protégés,
  - un *modèle de duplication* qui définit comment la base est répartie entre serveurs,
  - des *APIs* pour développer des applications clientes,
  - *LDIF*, un format d'échange de données.

Partage de systèmes de fichiers

Le système de fichiers NFS

- Network File System développé par Sun
- But : partager l'espace utilisateur entre toutes les stations
- NFS permet le montage d'arborescences appartenant à d 'autres systèmes connectés au réseau



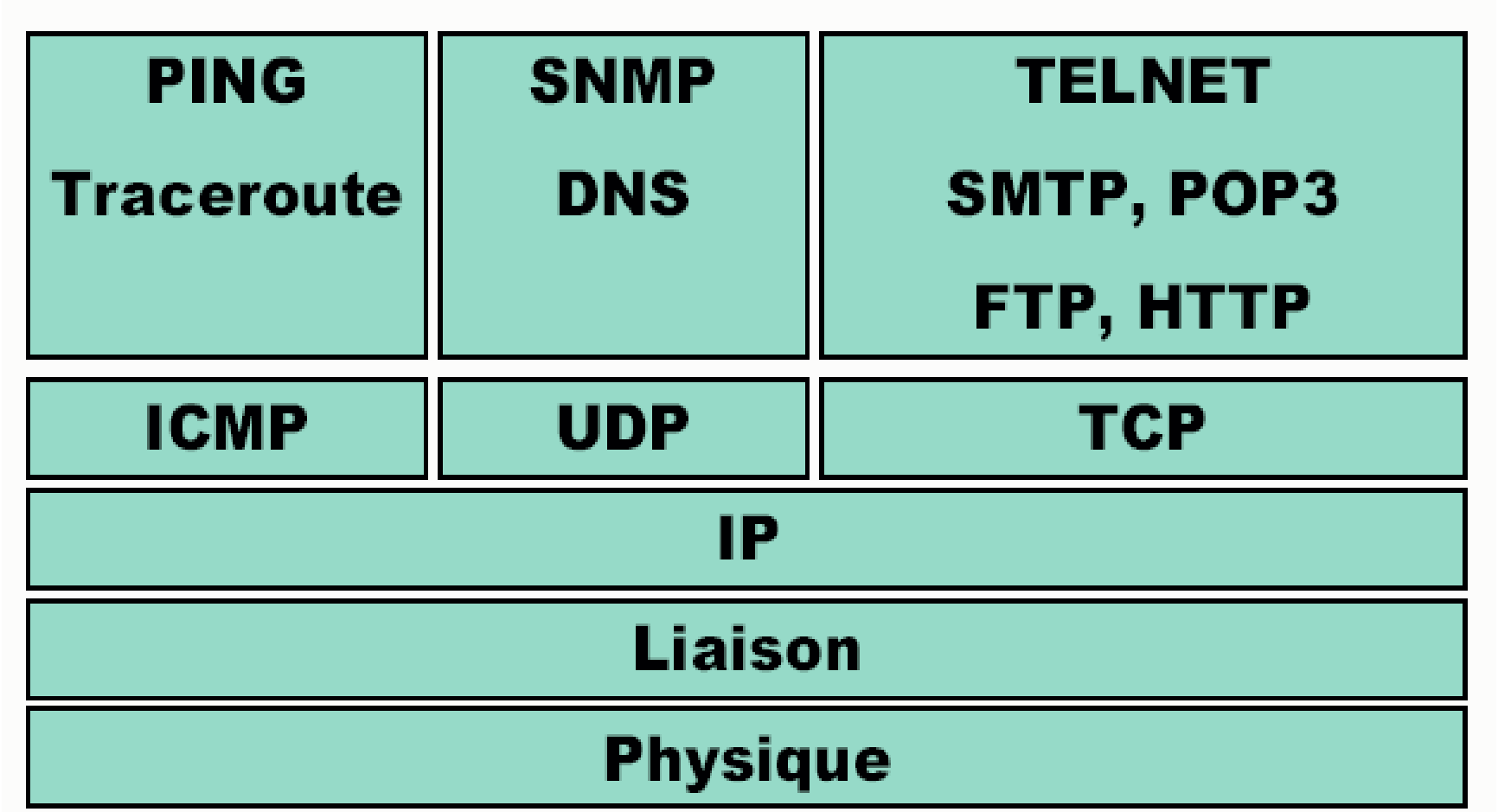
Le système de fichiers NFS

- Le client accède de façon transparente aux arborescences montées par NFS (cd, ls, cp, rm, …)
- Exemple de montage :
  - `mount -t nfs m2:/usr/local/users /home`
  - Monte le répertoire `/usr/local/users` de `m2` sur le répertoire `/home` de la machine locale
  - Le contenu de `/home` sur le disque local est masqué par celui de `/usr/local/users` sur `m2`
  - `ls /home` sur la machine locale revient à exécuter `ls /usr/local/users` sur `m2`
- Possibilité de définir les montages à faire au démarrage dans le fichier /etc/fstab

Serveurs SAMBA

- Partage de fichiers et d'imprimantes
- Implémente SMB (CIFS) au-dessus de NetBIOS sur TCP/IP
- Joue le rôle de serveur WINs et PDC (contrôleur de domaine)

En conclusion...



Pour tester vos connaissances

- **Exercice I – Réflexions sur le cours**
- 1 – Nous avons traité les formats de paquets suivants, chacun ayant une somme de contrôle dans son en-tête : IP, ICMP, IGMP, UDP et TCP. Pour chacun décrivez quelle partie du datagramme IP est couvert par la somme de contrôle et si celle-ci est obligatoire.
- 2 – Pourquoi les protocoles Internet traités (IP, ICMP, IGMP, UDP et TCP) rejètent silencieusement un paquet qui parvient avec une erreur de somme de contrôle ?
- 3 – TCP fournit un service de flux d'octet dans lequel les frontières d'enregistrements ne sont pas maintenues entre l'émetteur et le récepteur. Comment les applications peuvent-elles fournir leurs propres marques d'enregistrements ?
- 4 – Pourquoi les numéros de port source et destination sont-ils au début de l'en-tête TCP ?
- 5 – Pourquoi l'en-tête TCP a-t-il un champ longueur d'en-tête et non UDP ?
- **Exercice II – Questions sur TCP**
- 1 – Représenter le diagramme d'établissement de connexion de TCP.
- 2 – Pourquoi procéder à un échange en trois phases ?
- 3 – Pourquoi ne pas commencer la numérotation de séquence à 0 ?
- 4 – Pourquoi TCP structure les échanges de données en segment alors qu'il rend un service de flux d'octets ?

Correction

- **Exercice I – Réflexions sur le cours**
  - 1 – Tous sont obligatoires sauf la somme de contrôle UDP, la somme de contrôle IP couvre seulement l'en-tête IP, les autres commencent immédiatement après l'en tête.
  - 2 – L'adresse IP source, le numéro du port source ou le champ protocole pourraient avoir été corrompus
  - 3 – On peut utiliser un retour chariot et une fin de ligne. L'enregistrement peut être préfixé avec un compteur d'octets (cas des DNS)
  - 4 – Une erreur ICMP retourne les huit premiers octets du datagramme qui a provoqué l'erreur, pour pouvoir être traité par l'émetteur TCP il lui faut le port
  - 5 – Il y a des options à la fin de l'en-tête TCP et pas à la fin de l'en-tête UDP