département
Informatique

Université Claude Bernard Lyon 1

Licence Math-Informatique 1ère année

Partie 6

Olivier Glück

Université LYON 1 / Département Informatique

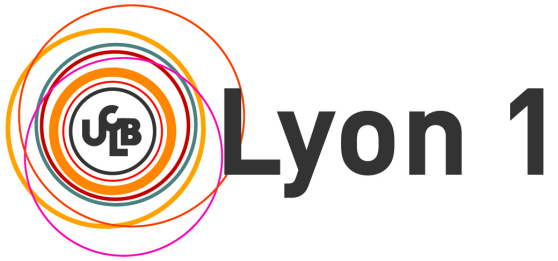
Olivier.Gluck@univ-lyon1.fr

http://perso.univ-lyon1.fr/olivier.gluck

Copyright

Plan du cours

- CM1 : Internet, les réseaux et le web
- CM2 : Pages HTML et feuilles de styles CSS
- CM3 : Web interactif, formulaires, pages dynamiques et PHP
- CM4 : Protocole HTTP, méthodes GET et POST
- CM5 : Les applications d'Internet
- CM6 : La couche transport : les protocoles TCP et UDP
- CM7 : Le protocole IP
- CM8 : Les protocoles Ethernet, ARP et ICMP. Synthèse des échanges entre un client et serveur Web



département
Informatique

Université Claude Bernard Lyon 1

CM6 - La couche transport : protocoles TCP/UDP

Le modèle Client/Serveur

Le protocole UDP et la rapidité

Le protocole TCP et la fiabilité

Plan du CM6

- Le modèle Client/Serveur

  Qu'est-ce que le modèle client/serveur ? Les sockets et les numéros de port, L'architecture TCP/IP

- Le protocole UDP et la rapidité

  L'en-tête UDP, Le mode non connecté, Quelles applications utilisent UDP ?

- Le protocole TCP et la fiabilité

  Qu'est-ce que la fiabilité ? Les mécanismes de la fiabilité, Le mode connecté, L'en-tête TCP, Quelles applications utilisent TCP

département
Informatique

Université Claude Bernard Lyon 1

Le modèle Client/Serveur

Qu'est-ce que le modèle client/serveur ?
Les sockets et les numéros de port
L'architecture TCP/IP

Les applications vues dans ce cours

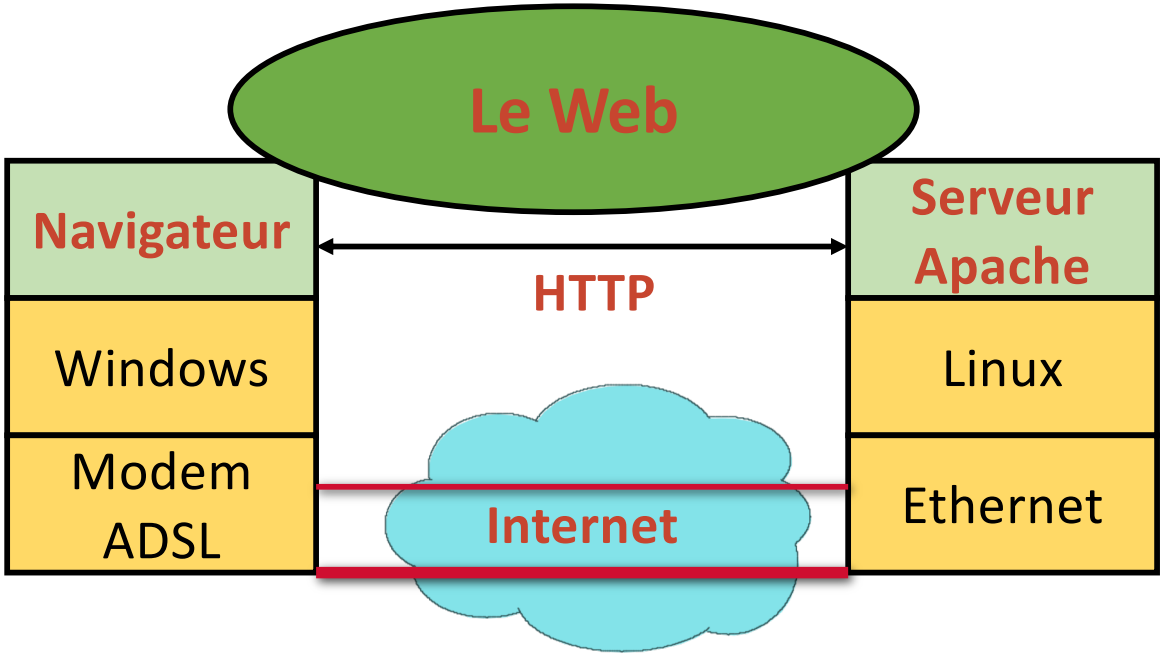
- Le web (**HTTP**)
- La connexion à distance (**telnet**, **ssh** et **X**)
- Le courrier électronique (**SMTP**, **POP**, **IMAP**, **Webmail**)
- La résolution des noms (**DNS**)
- Le transfert de fichiers (**FTP**)
- L'accès aux fichiers distants (**NFS**, **SMB**)
- L'annuaire fédérateur (**LDAP**)

Toutes ces applications fonctionnent selon le modèle Client/Serveur !

Par exemple le Web…

- Une application d'Internet qui permet le partage de documents liés entre eux et appelés "pages web"
- Une page web peut contenir du texte, des images, des programmes, des liens vers d'autres pages web…
- Fonctionne en mode Client/Serveur au dessus de l'architecture TCP/IP

L'application est répartie sur le client et le serveur qui dialoguent selon un protocole applicatif spécifique



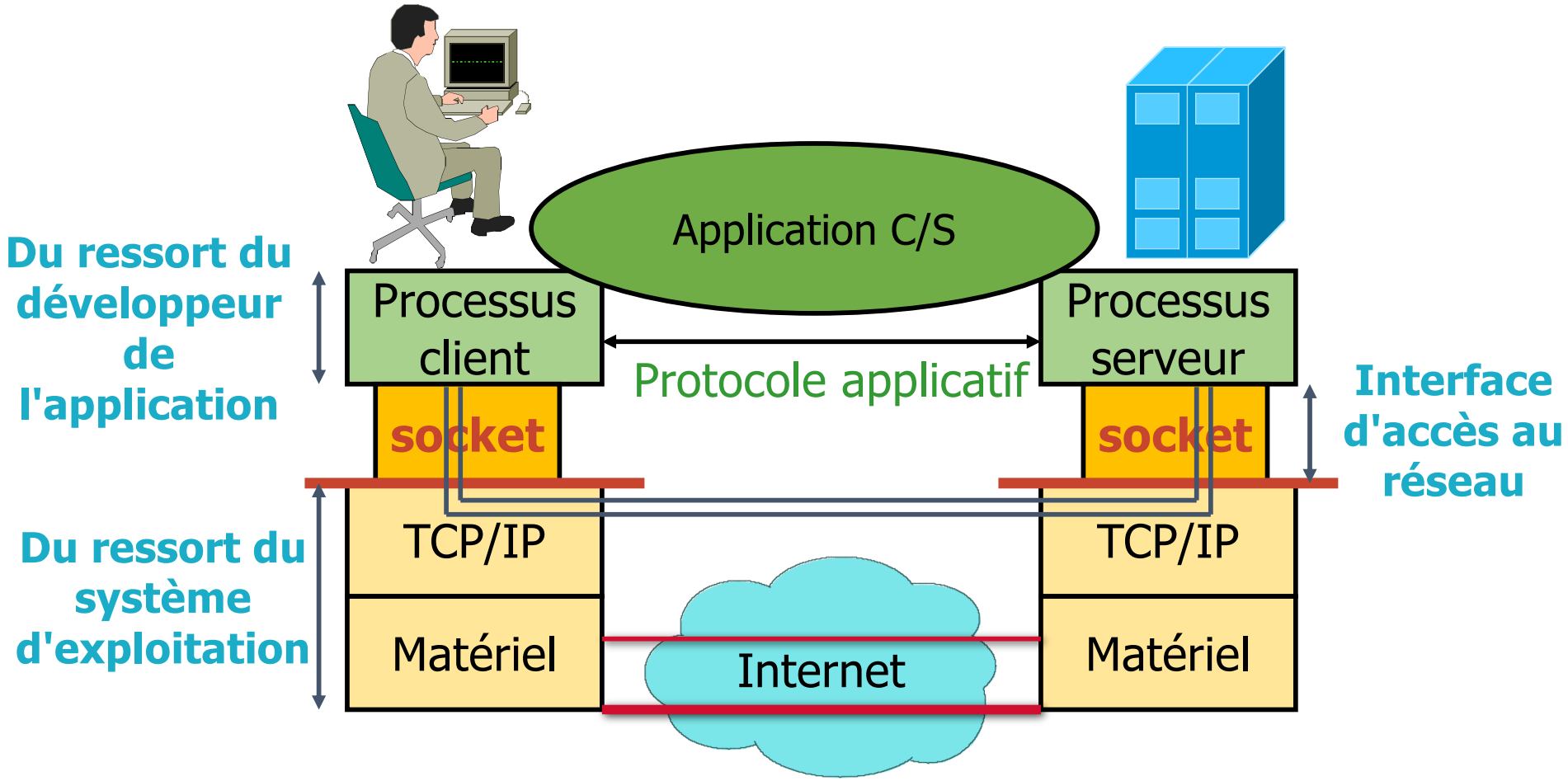
Le modèle Client / Serveur

- Idée : l'application est répartie sur différents sites pour optimiser le traitement, le stockage...
- Le client
  - effectue une demande de service auprès du serveur (**requête**)
  - initie le contact (parle en premier), ouvre la session
- Le serveur
  - est la partie de l'application qui offre un service
  - est à l'écoute des requêtes clientes
  - répond au service demandé par le client (**réponse**)
- Le client et le serveur ne sont pas identiques, ils forment un système coopératif
- Un serveur peut répondre à plusieurs clients simultanément

Le modèle Client / Serveur

- Une application Client/Serveur, c'est
  - **une partie cliente** qui exécute des requêtes vers un serveur
  - **une partie serveur** qui traite les requêtes clientes et y répond
  - **un protocole applicatif** qui définit les échanges entre un client et un serveur
  - **un accès via une API** (interface de programmation, généralement les sockets) à la couche de transport des messages
- Bien souvent les parties cliente et serveur ne sont pas écrites par les mêmes programmeurs --> rôle important des RFCs qui spécifient le protocole !

Les sockets, interface d'accès au réseau

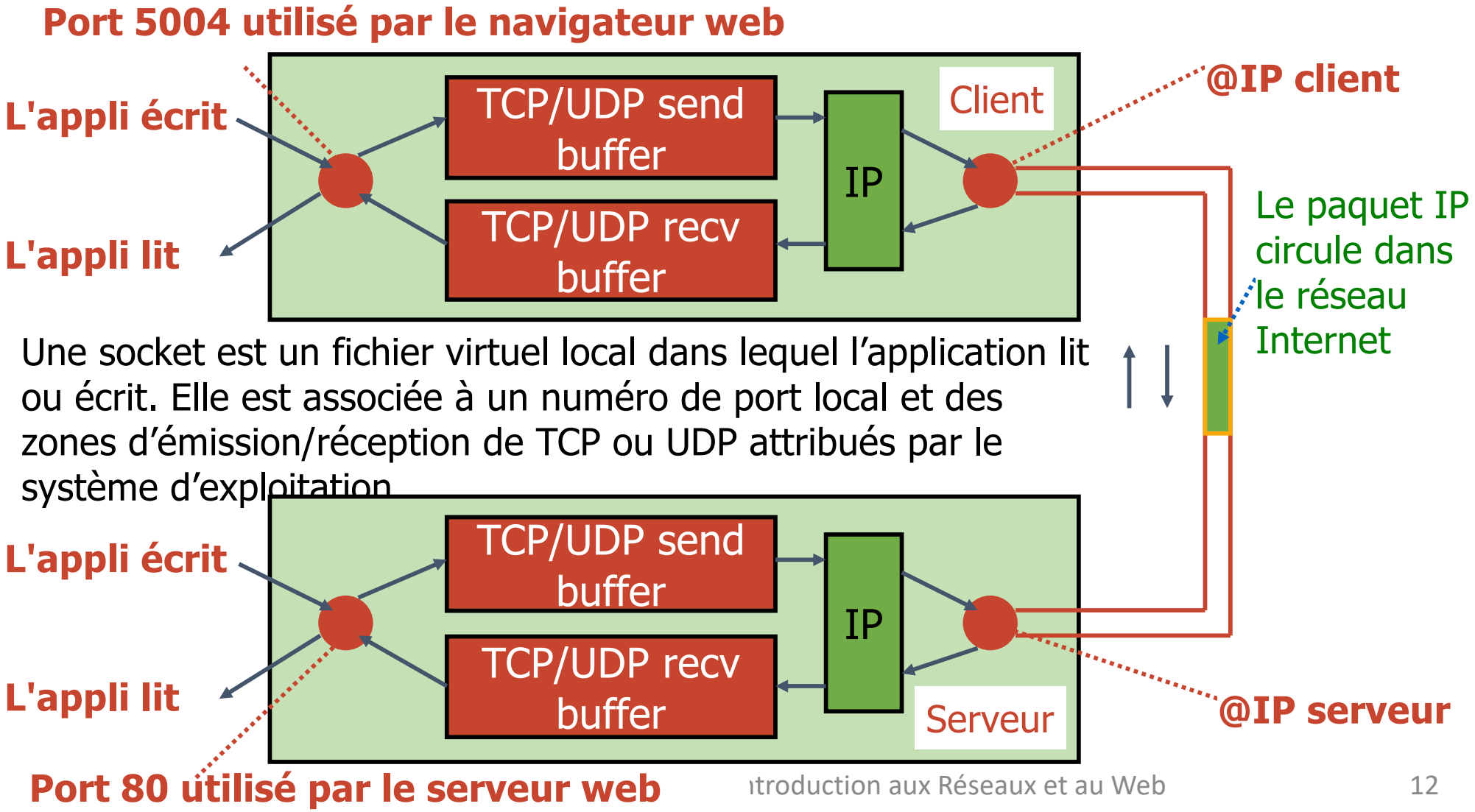


Une socket : interface locale à l'hôte, créée par l'application, contrôlée par l'OS

Porte de communication entre le processus client et le processus serveur

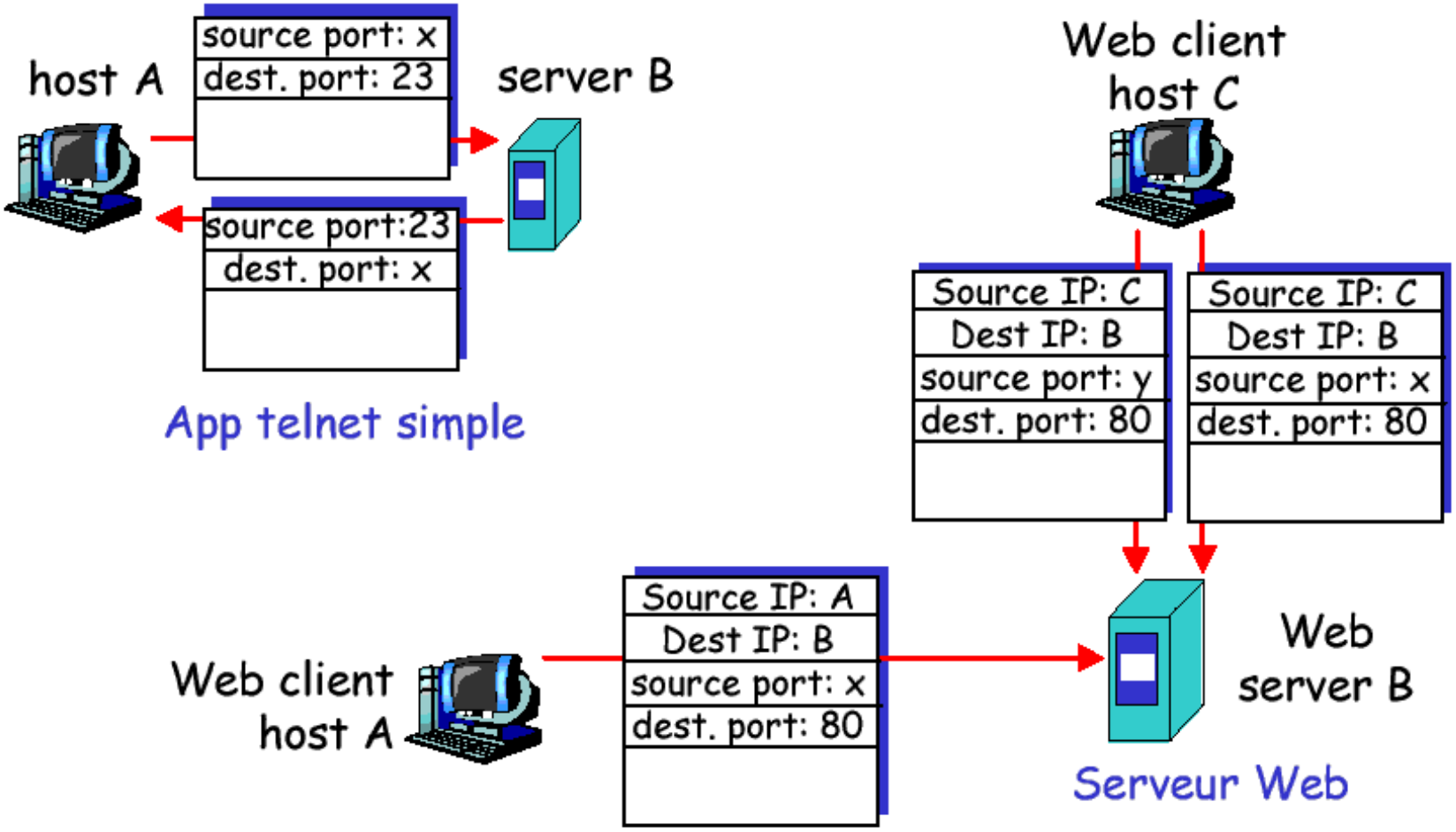
Les sockets et les numéros de port

- Un échange Client/Serveur = **(@IP_src,port_src,@IP_dest,port_dest)**

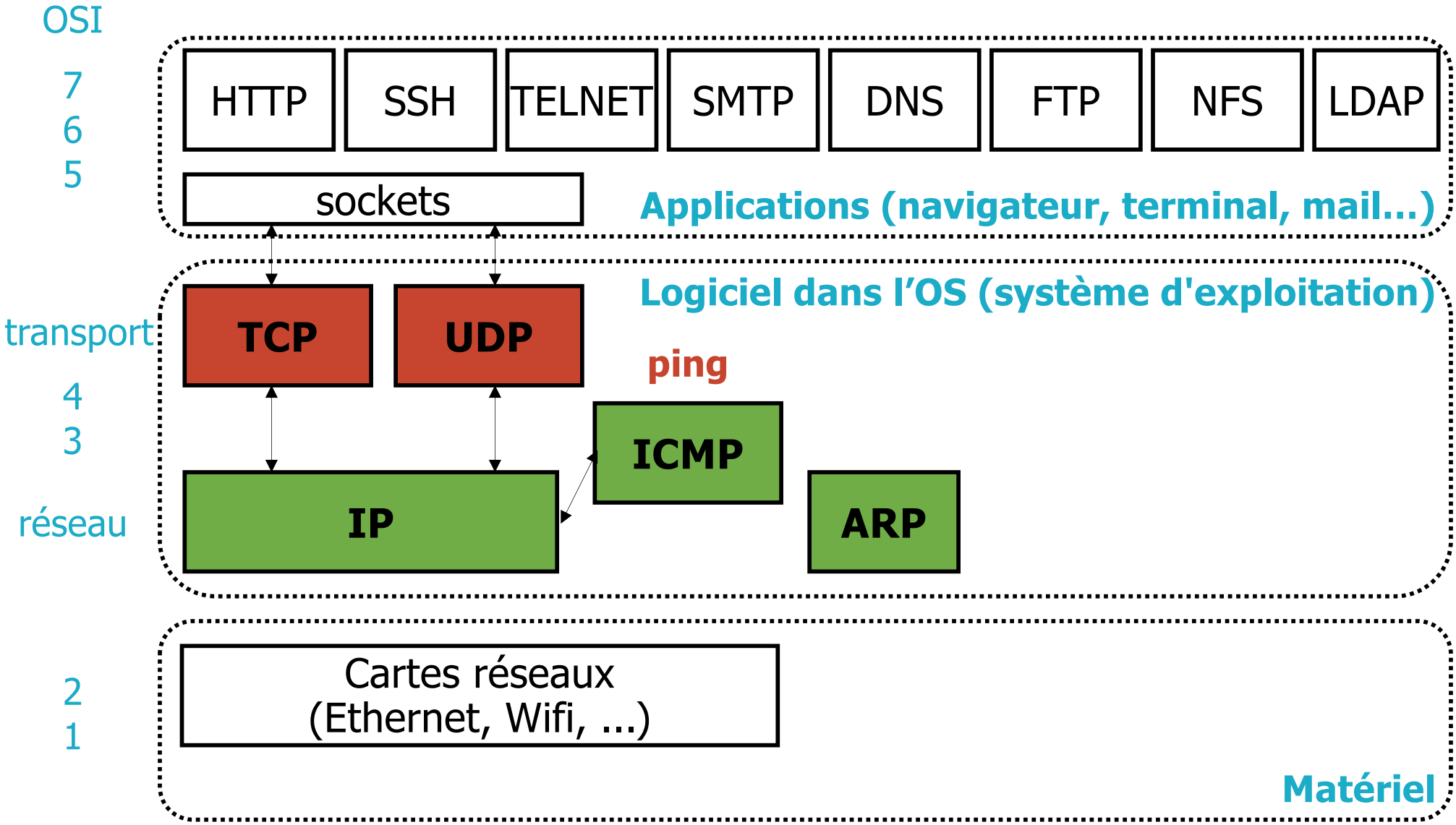


Les numéros de ports et les adresses IP

Les ports inférieurs à 1024 sont réservés pour les serveurs car le port du serveur doit être fixé et connu à l'avance pour que le client puisse faire sa demande.

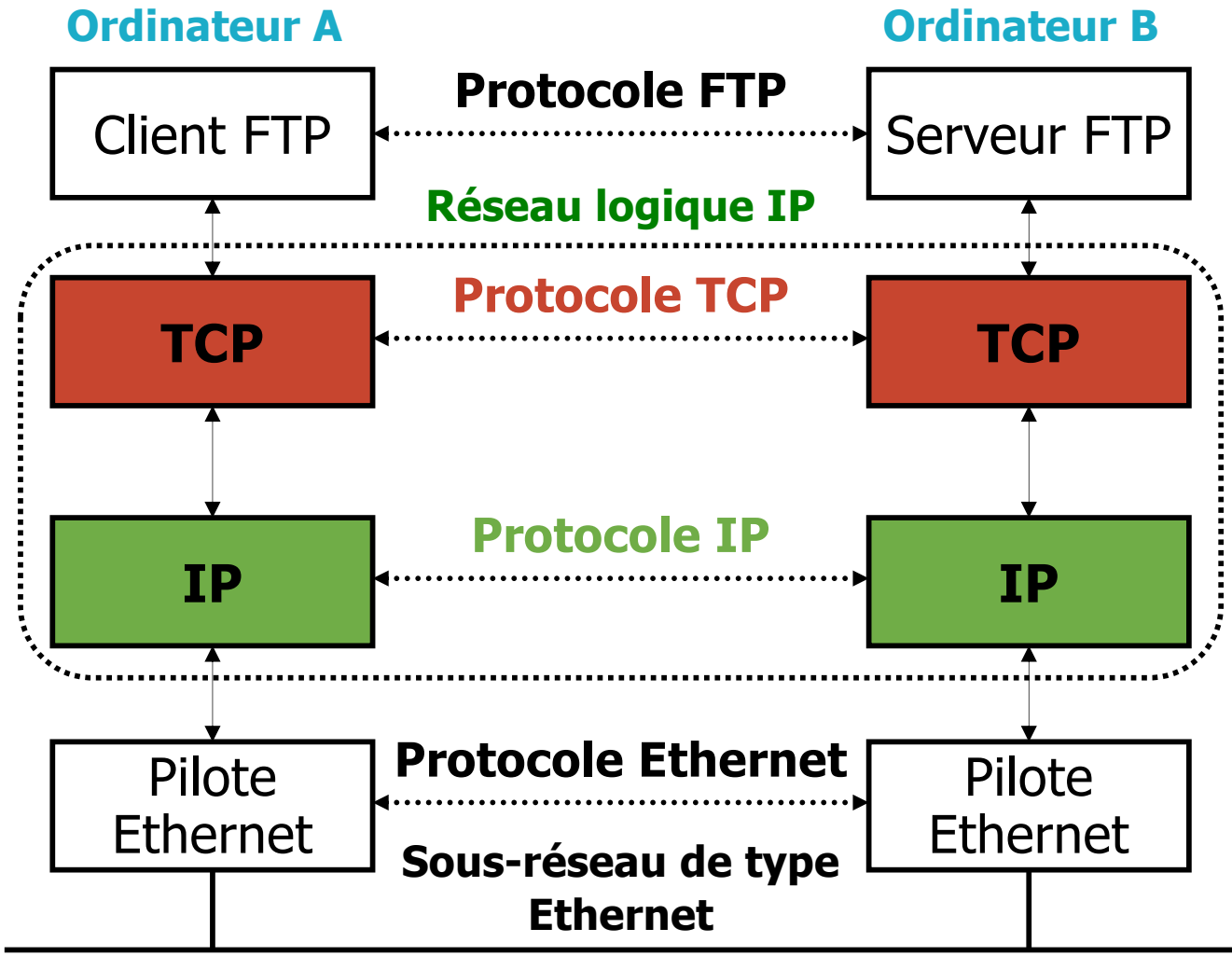


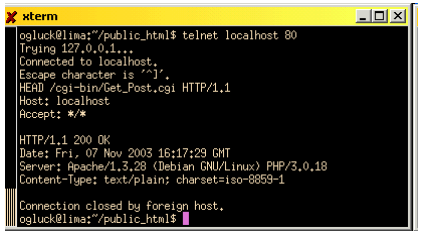
L'architecture de TCP/IP



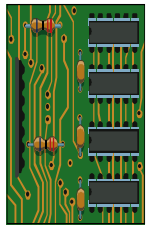
Communications sans routeur

- Deux machines sur un même sous réseau





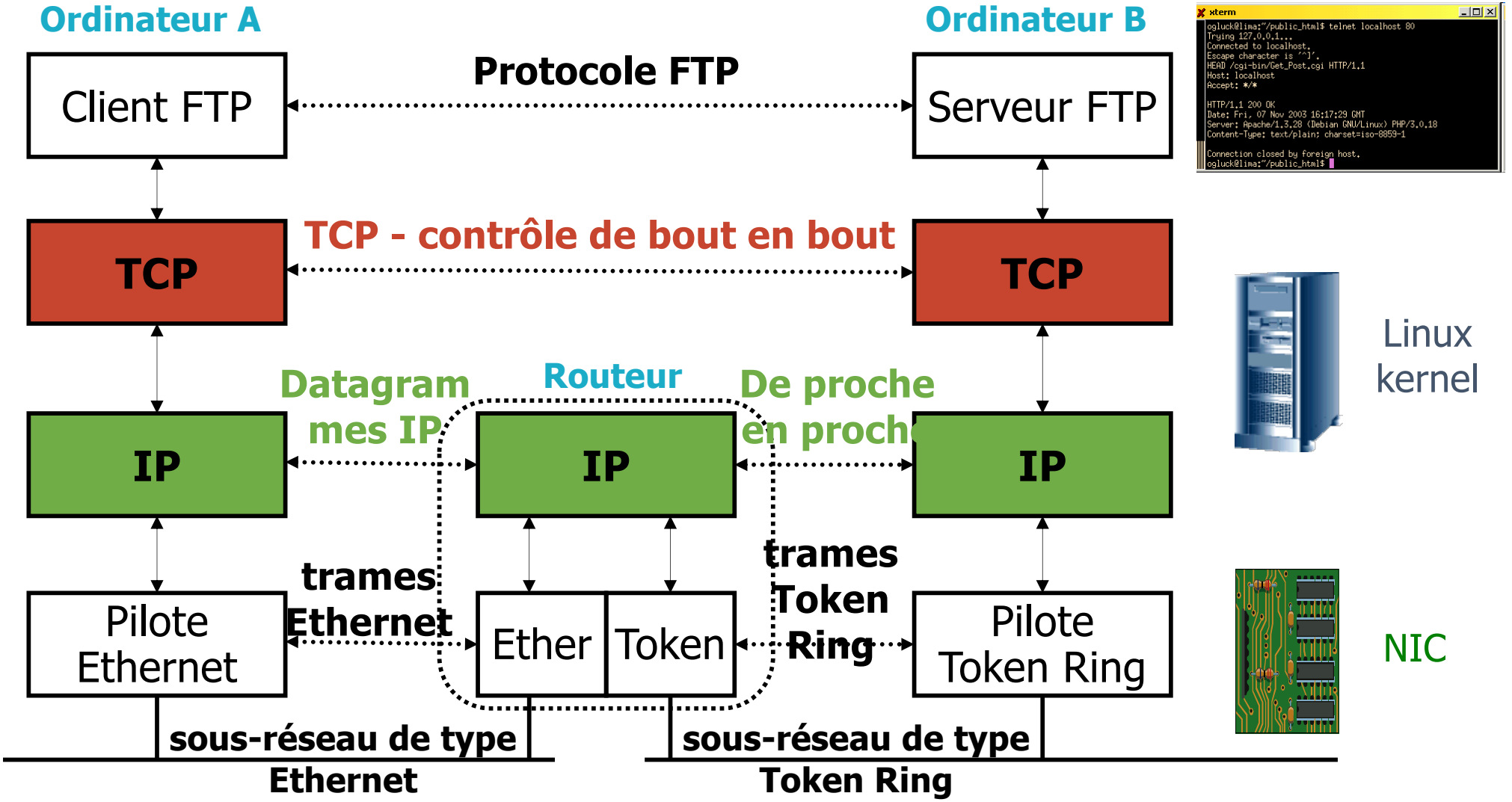
Linux kernel



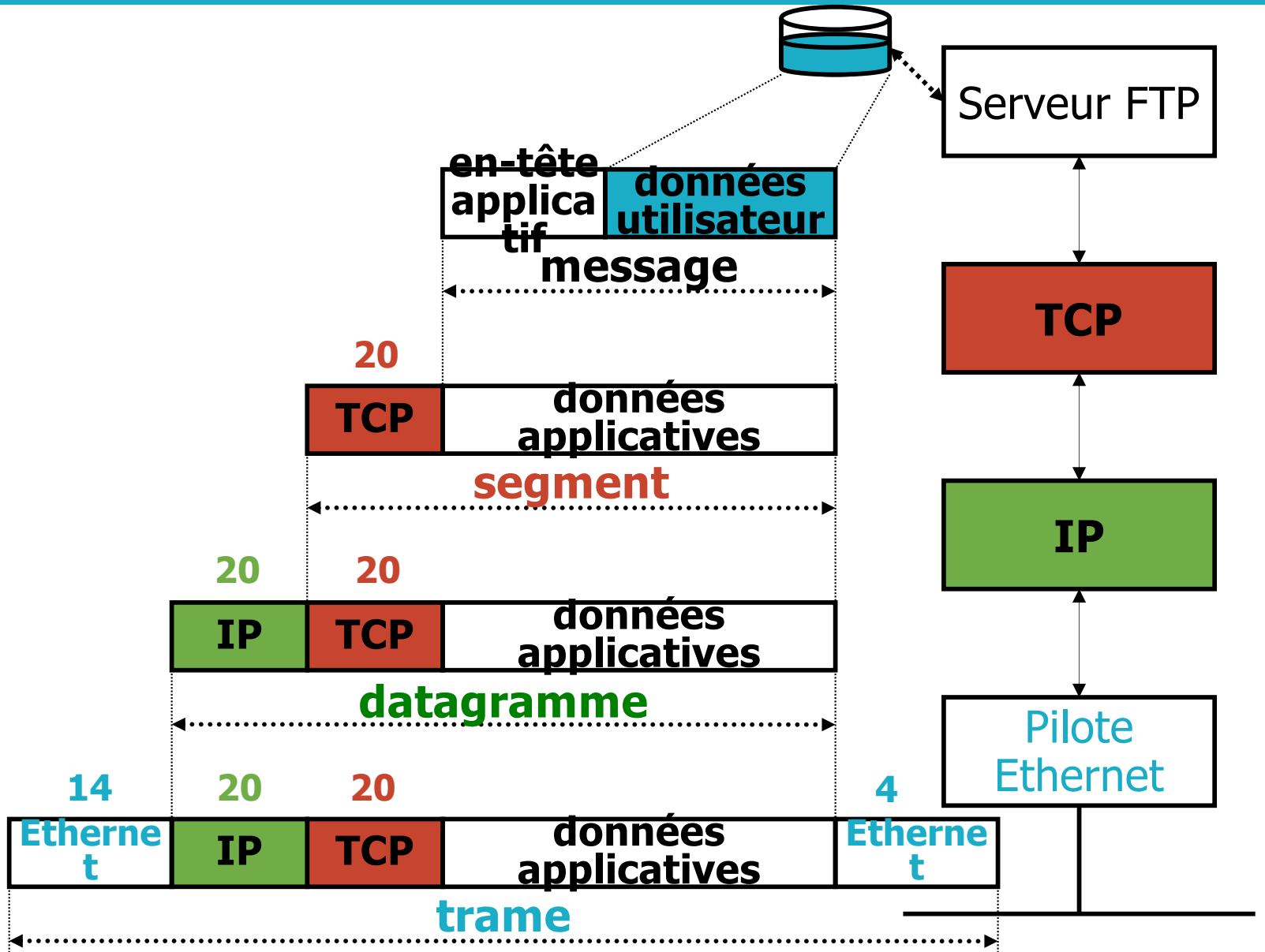
NIC

Communications avec routeur(s)

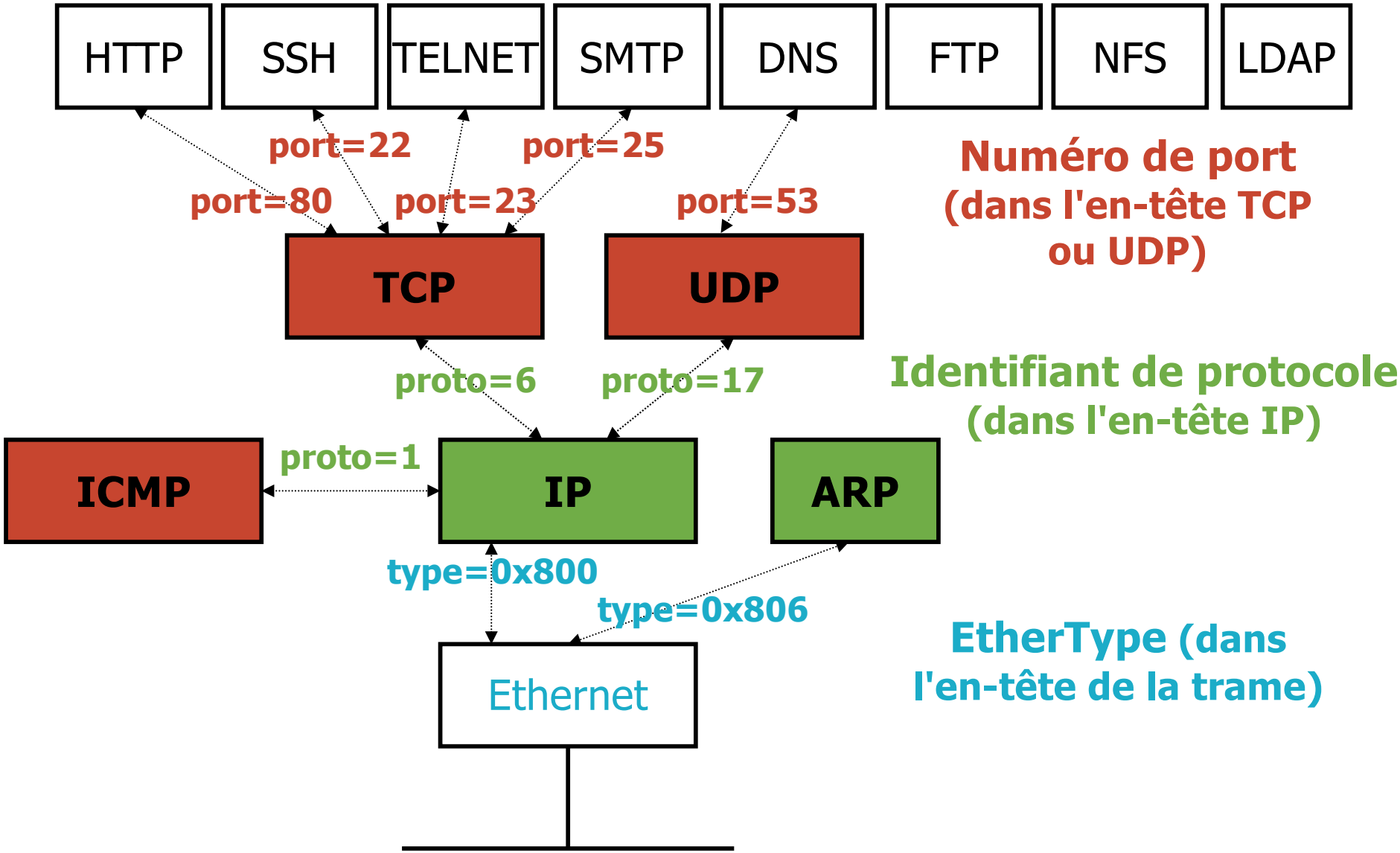
- Prise en compte de l'hétérogénéité

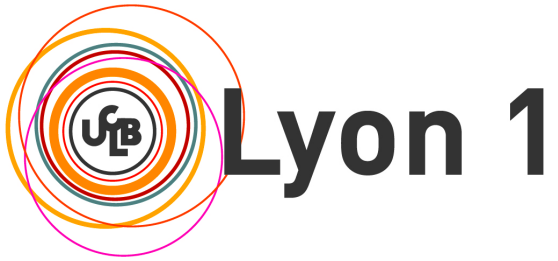


L’encapsulation : ajout des en-têtes



La désencapsulation : identifier la couche >





département
Informatique

Université Claude Bernard Lyon 1

Le protocole UDP et la rapidité

L'en-tête UDP

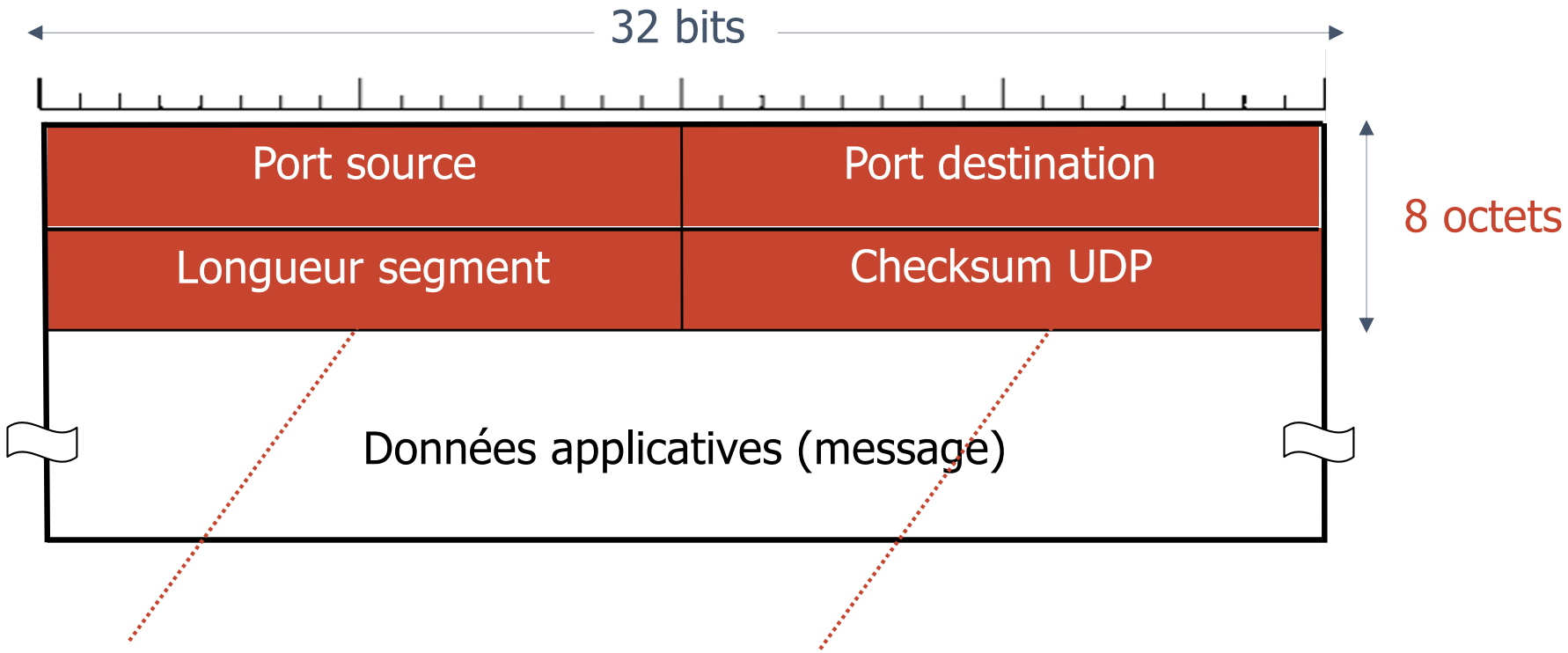
Le mode non connecté

Quelles applications utilisent UDP ?

Le protocole UDP

- UDP (RFC 768) - User Datagram Protocol
  - protocole de transport le plus simple
  - service de type best-effort (comme IP)
    - les datagrammes UDP peuvent être perdus
    - les datagrammes UDP peuvent arriver dans le désordre
  - mode non connecté : chaque datagramme UDP est traité indépendamment des autres
- Pourquoi un service **non fiable sans connexion** ?
  - simple donc **rapide** (pas de délai de connexion, pas d'état entre émetteur/récepteur)
  - petit en-tête donc économie de bande passante
  - UDP peut émettre aussi rapidement qu'il le souhaite : pas de limite à l'envoi contrairement à TCP (contrôle de congestion)

L'en-tête UDP : 8 octets



Taille totale du datagramme (en-tête+données)

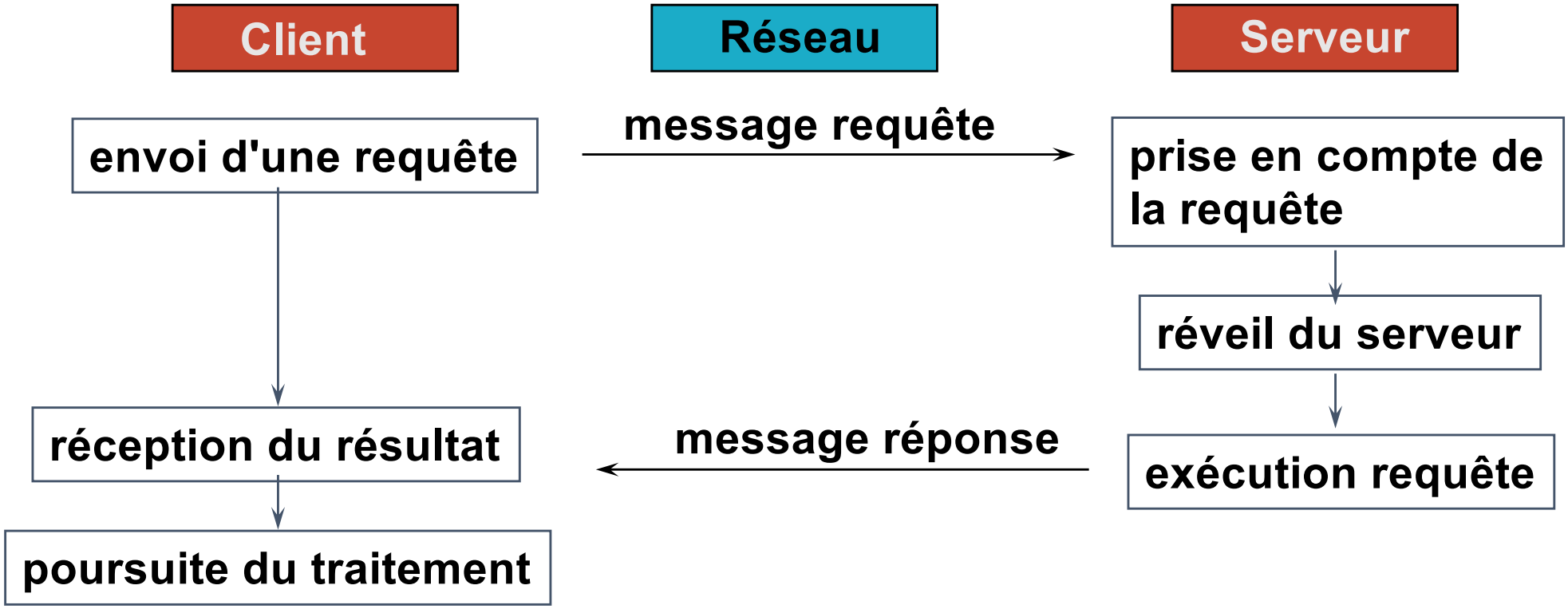
Somme de contrôle du datagramme (en-tête+données)

optionnel : peut être à 0

UDP = IP + numéros de port !!

UDP est en mode non connecté

Permet d'envoyer rapidement des messages mais sans l'autorisation du destinataire donc aucune garantie sur la réception du message



Exemple de mode non connecté : courrier postal, mail

Tous les protocoles d'Internet sont en mode non connecté SAUF TCP

Quelles applications utilisent UDP ?

- Pour les applications qui ont besoin d'envoyer rapidement
  - UDP permet des envois rapides sans limitation mais sans garantie donc sans fiabilité
- Souvent utilisé pour les **applications multimédias**
  - Vidéos, son, musique, streaming, visioconférence, voix sur IP
  - Ces applications sont tolérantes aux pertes/erreurs et sensibles au débit (les données doivent arriver à la bonne vitesse)
- Autres utilisations d'UDP
  - Applications qui envoient peu de données et qui ont besoin de rapidité
  - exemple : **DNS**
- Transfert fiable sur UDP
  - Comme UDP n'apporte aucune garantie, l'application peut ajouter des mécanismes pour réparer les pertes ou erreurs (acquittements…)

Lyon 1

département Informatique

Université Claude Bernard Lyon 1

Le protocole TCP et la fiabilité

Qu'est-ce que la fiabilité ?

Les mécanismes de la fiabilité

Le mode connecté

L'en-tête TCP

Quelles applications utilisent TCP ?

Le protocole TCP et la fiabilité

- Transmission Control Protocol (RFC 793, 1122, 1323, 2018, 2581)
- Transport **fiable** en **mode connecté**
  - entre un client et un serveur : (@IP src, port src) --> (@IP dest, port dest)
  - transporte un flot d'octets (ou flux)

    L'application lit/écrit des octets dans un tampon, TCP assure la fiabilité
- Fiabilité : faire en sorte que tout ce qui arrive est exactement ce qui a été envoyé
  - Les données ne doivent pas être perdues : **sans perte**
  - Les données ne doivent pas subir d'erreurs : **sans erreur**

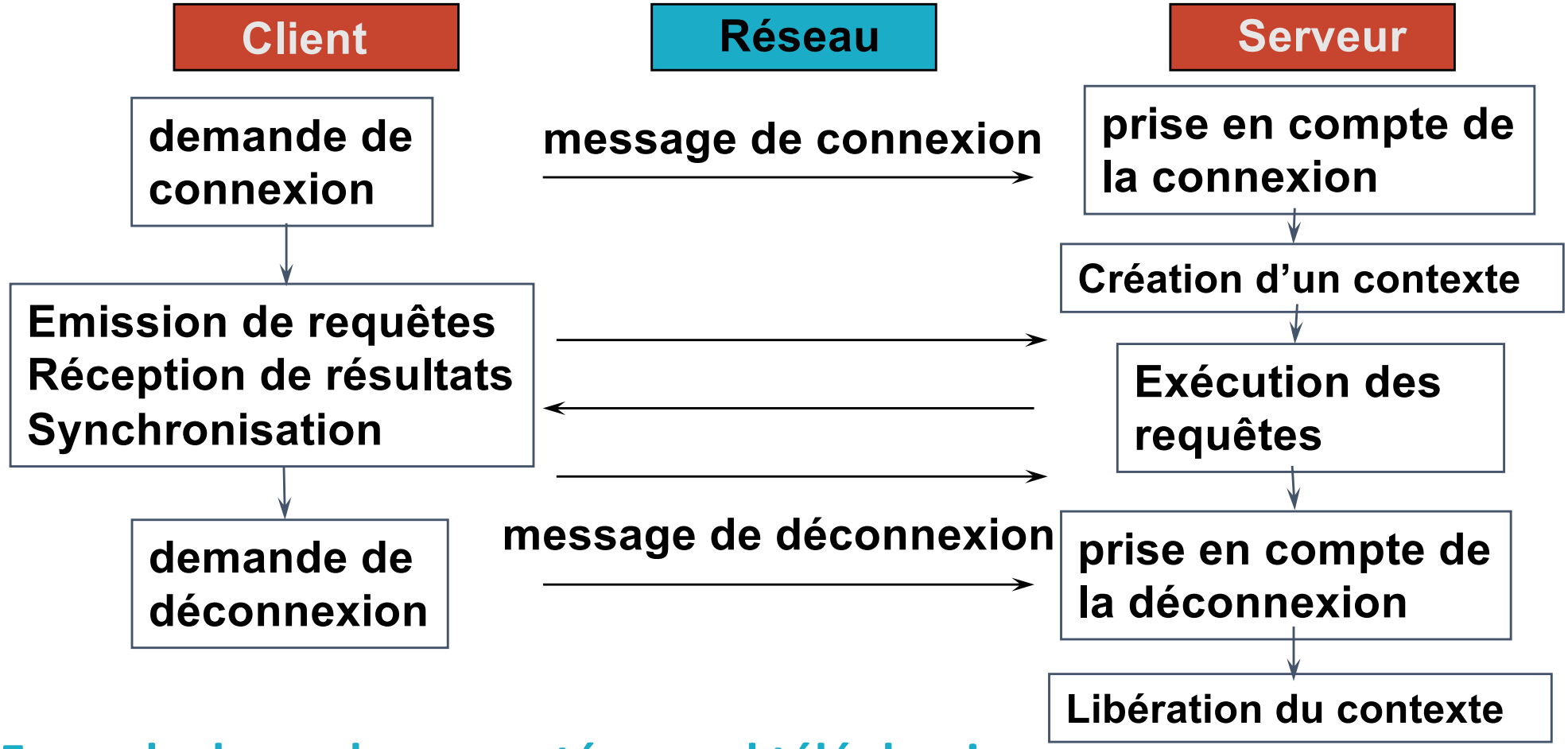
    Une erreur = un bit qui change de valeur pendant le transfert
  - Les données doivent arriver **dans l'ordre**
  - Les données ne doivent pas arriver en double : **sans duplication**

Les mécanismes de la fiabilité

- Fiabilité : **sans perte, sans erreur, dans l'ordre, sans duplication**
- En cas de perte ou erreur, il faut retransmettre si on n'a pas reçu d'acquittement (ACK) au bout d'un certain temps
- Pour détecter une perte ou une duplication, il faut des ACK et numéroter les messages et les ACK
- Pour détecter les erreurs, on utilise les checksum
- Pour retransmettre, il faut conserver les messages envoyés qui n'ont pas encore été acquittés
- Mécanismes : **retransmissions, timeout, ACK, stockage des messages non acquittés, checksum, numérotation des messages et des acquittements**

TCP est en mode connecté

Permet d'envoyer des messages avec fiabilité mais limitation du débit à l'envoi : contrôle de flux et contrôle de congestion

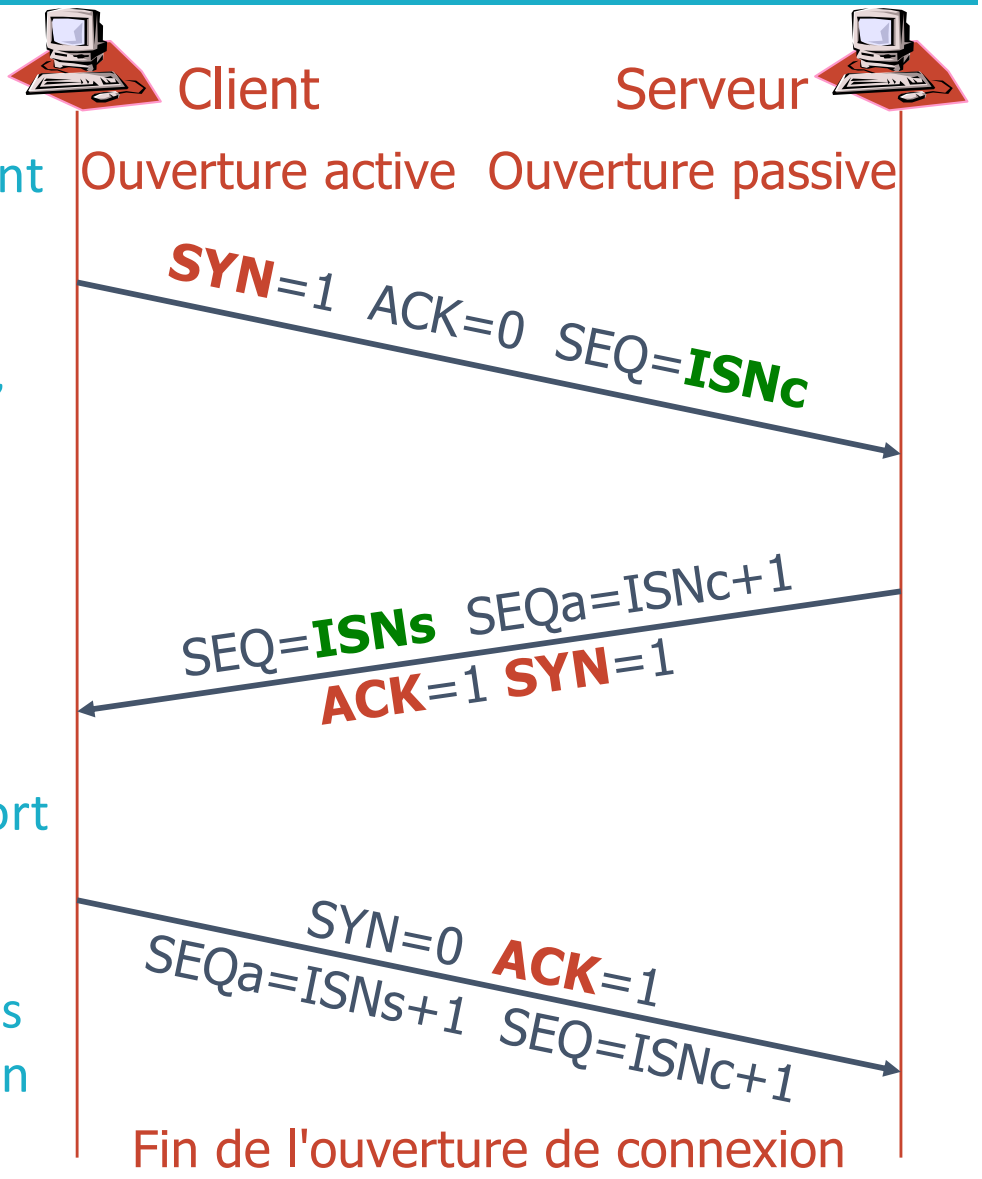


Exemple de mode connecté : appel téléphonique

TCP est le seul protocole d'Internet en mode connecté

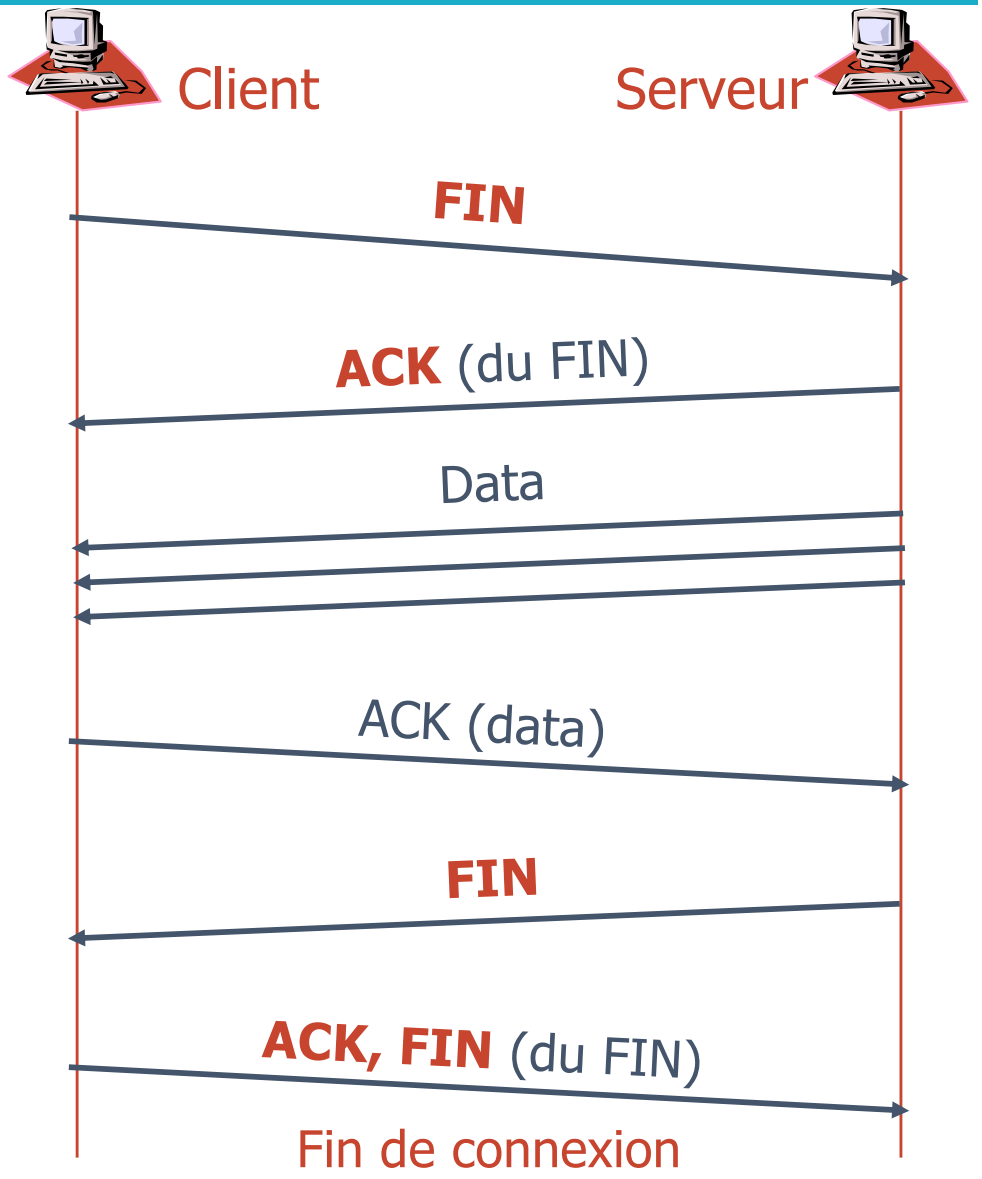
Etablissement d'une connexion TCP

- Connexion en trois phases
  - 1 - demande d'ouverture par le client (SYN), choix ISNc
  - 2 - acceptation par le serveur (SYN+ACK), allocation des tampons, choix ISNs
  - 3 - le client acquitte l'acceptation (ACK)
- Mode Client/Serveur
  - Le TCP du client fait une demande d'ouverture de connexion vers le port du serveur qui doit être connu à l'avance
  - Le TCP du serveur est en attente des demandes d'ouverture de connexion en provenance des clients



Fermeture d'une connexion TCP

- Fermeture négociée
  - 1 - demande de fin de connexion (FIN) par une des extrémités
  - 2 - acquittement du FIN (ACK) mais mise en attente de la demande (Le serveur a encore des données non transmises)
  - 3 - envoi des données en attente
  - 4 - acquittement des données (ACK)
  - 5 - acceptation de la fin de connexion par le serveur (FIN)
  - 6 - acquittement de la fin de connexion (ACK)



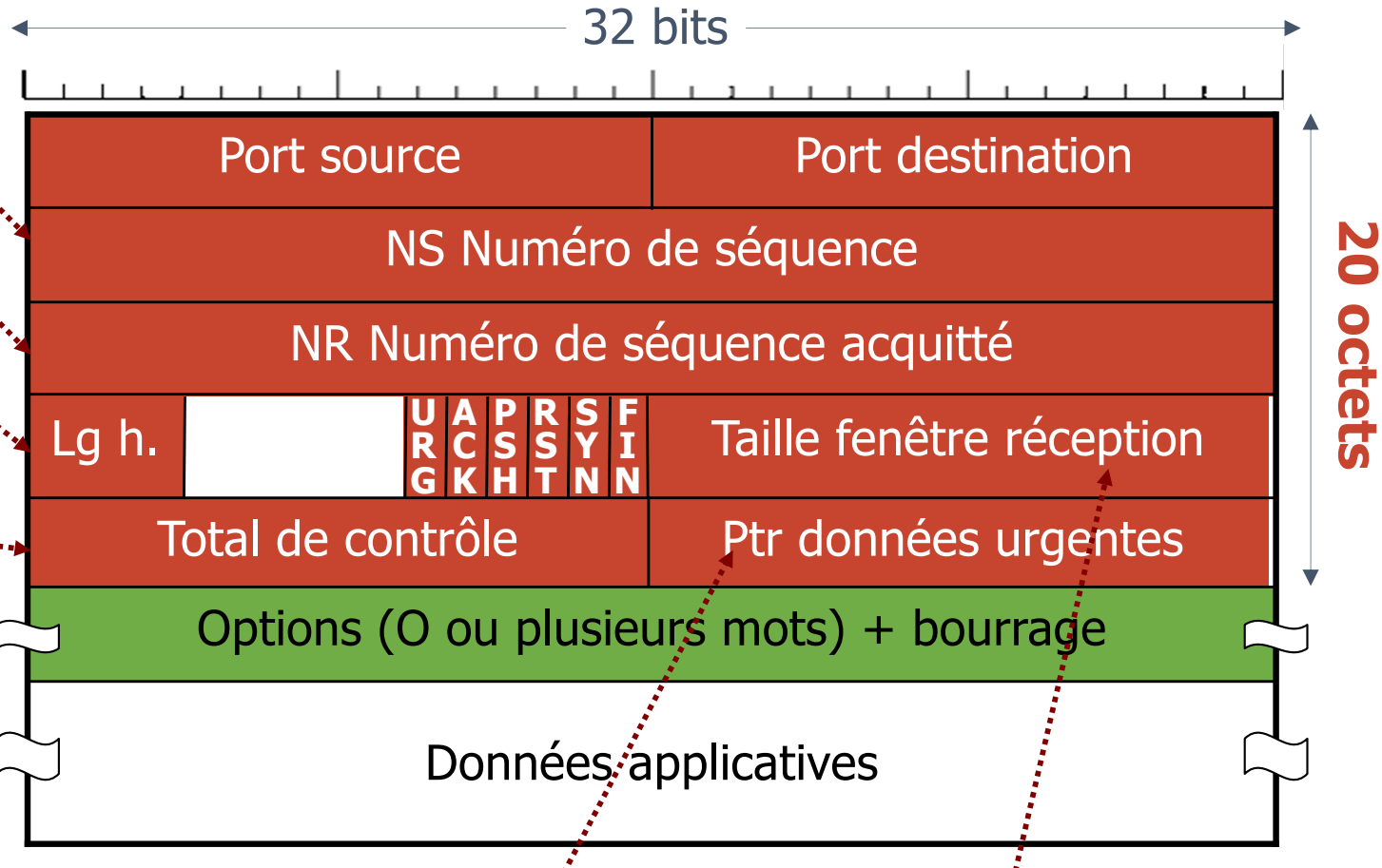
L'en-tête TCP : 20 octets

Numéro du premier octet du segment

Numéro du prochain octet attendu

Longueur en-tête en multiple de 4 octets

Checksum sur tout le segment (cf. UDP)

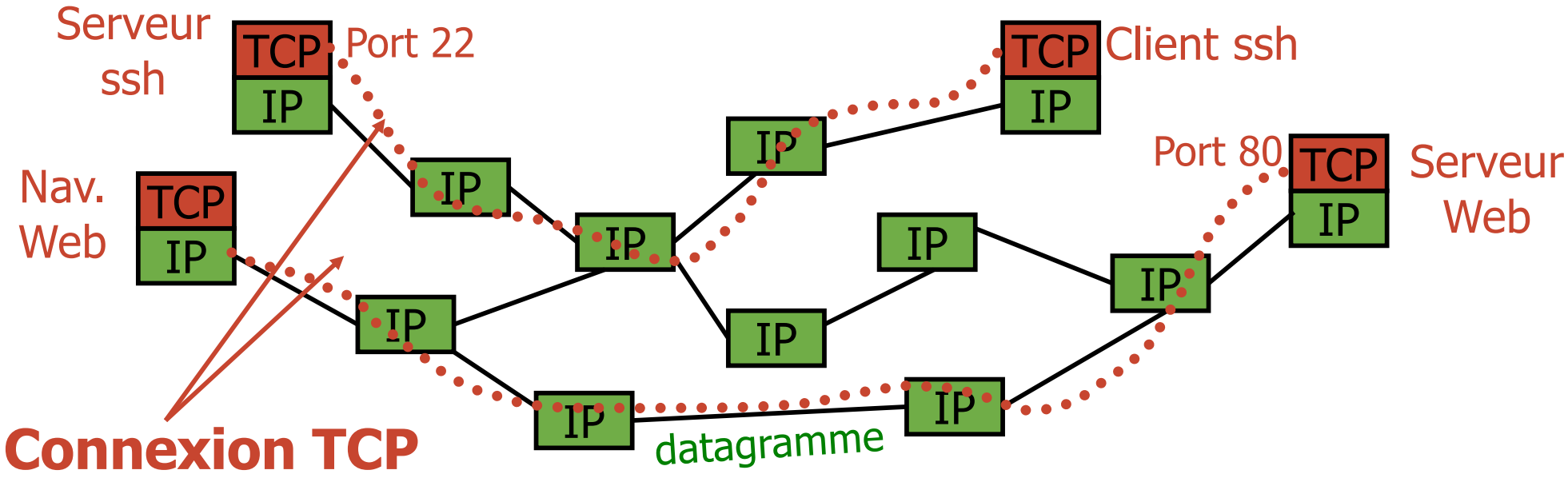


Les données comprises entre le premier octet DATA et la valeur du Ptr sont urgentes : TCP interrompt l'application pour forcer la lecture

Nb d'octets que le récepteur peut recevoir

Caractéristiques du protocole TCP

Couche transport : communications entre applis



- TCP - protocole de transport **de bout en bout**
  - uniquement présent **aux extrémités**
  - transport **fiable** de **segments** (mode **connecté**)
  - protocole complexe (retransmission, gestion des erreurs, séquencement, …)

Quelles applications utilisent TCP ?

Toutes celles qui ne peuvent pas se passer de la fiabilité c'est à dire presque toutes !

- Le web (**HTTP**)
- La connexion à distance (**telnet**, **ssh** et **X**)
- Le courrier électronique (**SMTP**, **POP**, **IMAP**, **Webmail**)
- Le transfert de fichiers (**FTP**)
- L'annuaire fédérateur (**LDAP**)

Les applications multimédias, NFS et le DNS utilisent généralement UDP.