

L'architecture de TCP/ IP (1)

Une version simplifiée du modèle OSI

Application FTP, WWW, telnet, SMTP, ...

Transport TCP, UDP (entre 2 processus aux extrémités)

- ★ TCP : transfert fiable de données en mode connecté

- ★ UDP : transfert non garanti de données en mode non connecté

Réseau IP (routage)

Physique transmission entre 2 sites

TCP → Transport Control Protocol

UDP → User Datagram Protocol

IP → Internet Protocol

L'architecture de TCP/ IP (2)

OSI

7
6
5

HTTP FTP TELNET SMTP DNS NFS . . .

Sockets

Applications (processus utilisateur)

4
3

TCP UDP

Protocoles de transport

Système d'exploitation

Protocoles de contrôle de l'Internet

IP ICMP ARP RARP BOOTP DHCP

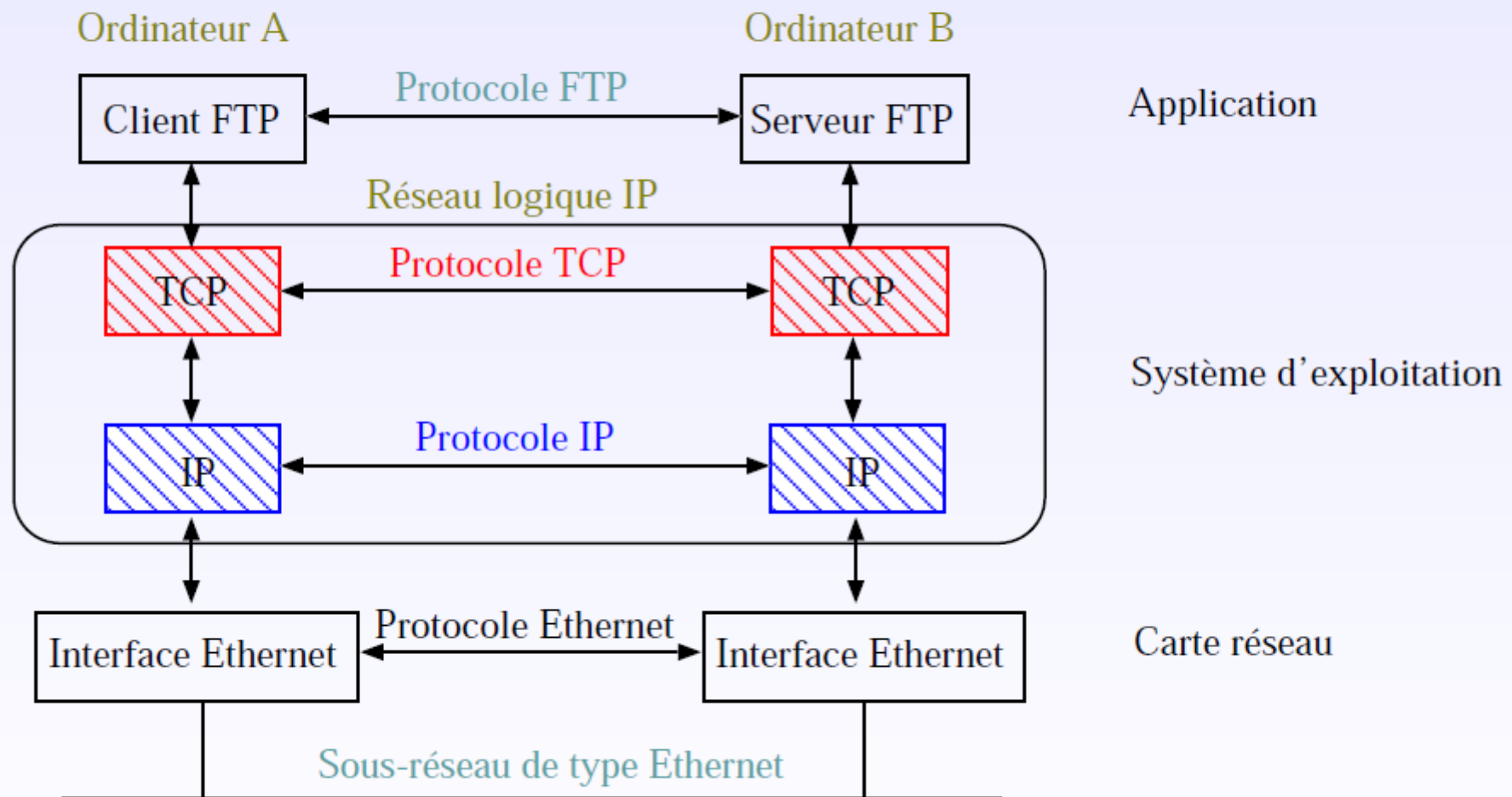
2
1

SLIP PPP ATM FRelay Ethernet, Token Ring, ...

Matériel

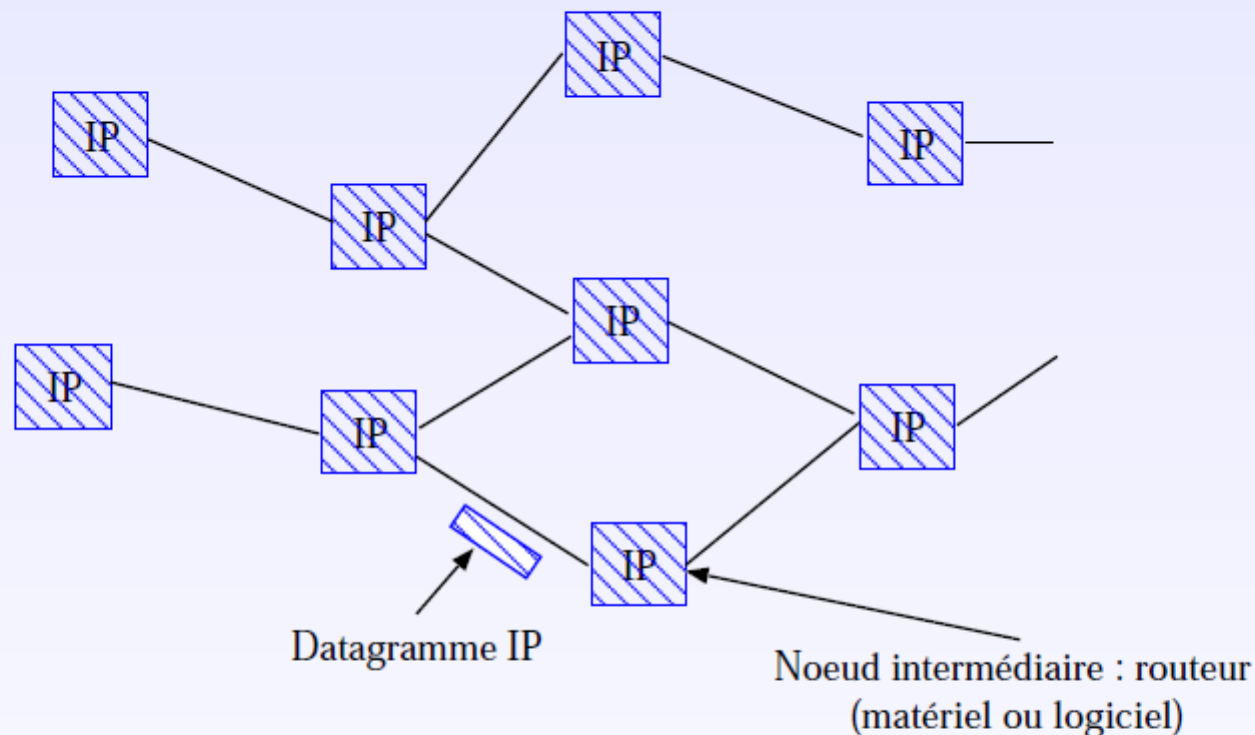
L'architecture de TCP/ IP (3)

Deux machines sur un même sous-réseau IP



L'architecture de TCP/ IP (5)

Couche réseau : communications entre machines

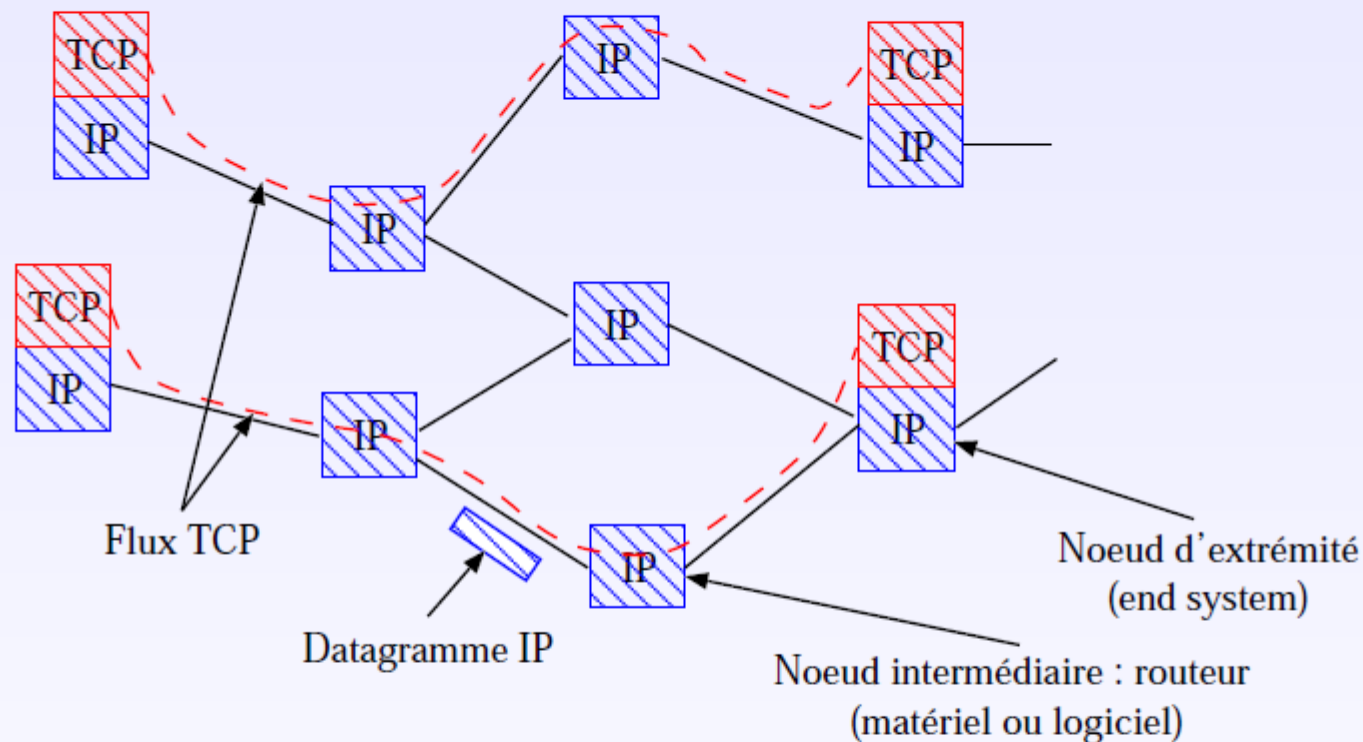


IP - protocole d'interconnexion, best-effort

- ★ acheminement de **datagrammes** (mode **non connecté**)
- ★ peu de fonctionnalités,
- ★ pas de garanties simple mais robuste (défaillance d'un noeud intermédiaire)

L'architecture de TCP/ IP (6)

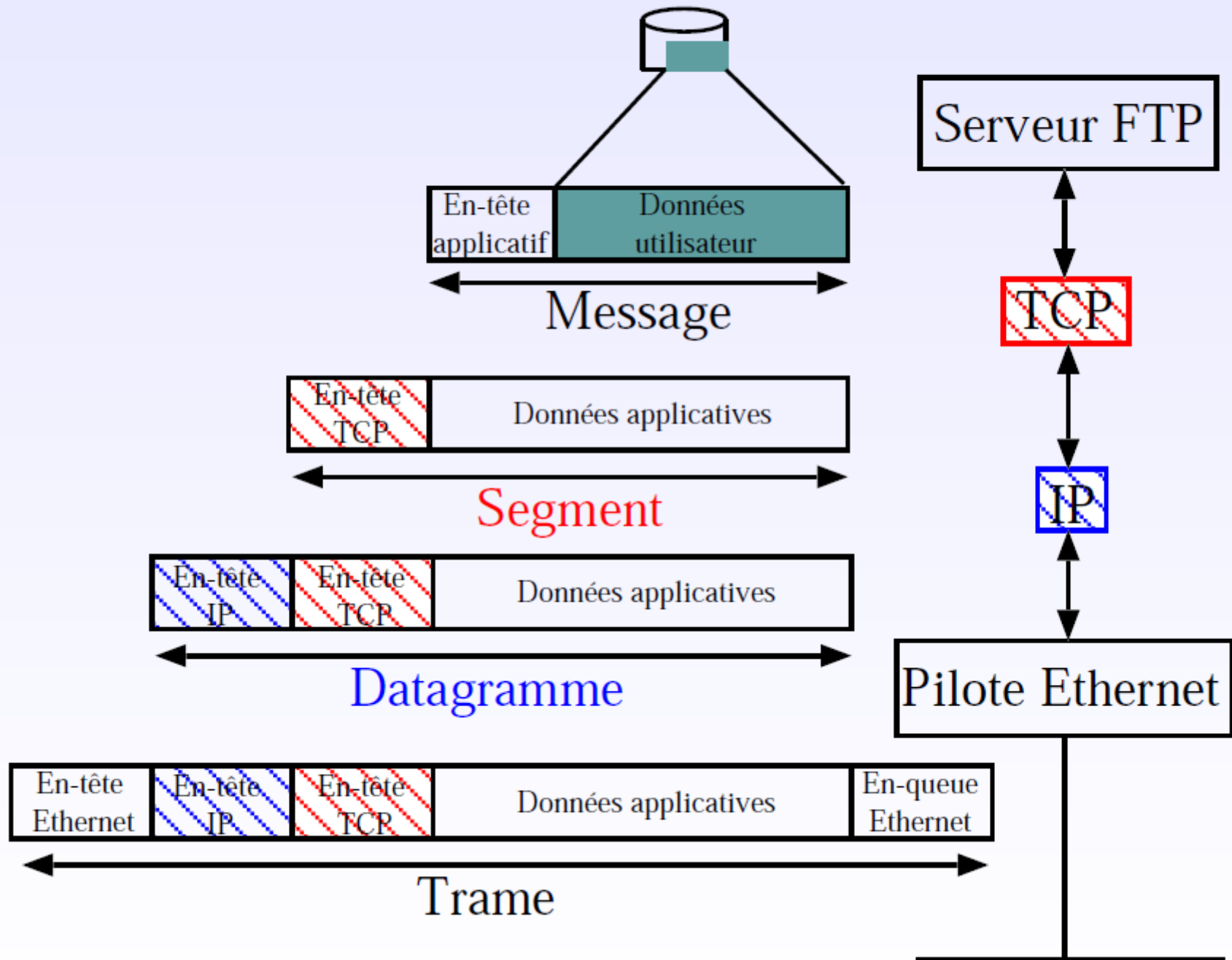
Couche transport : communications entre applications



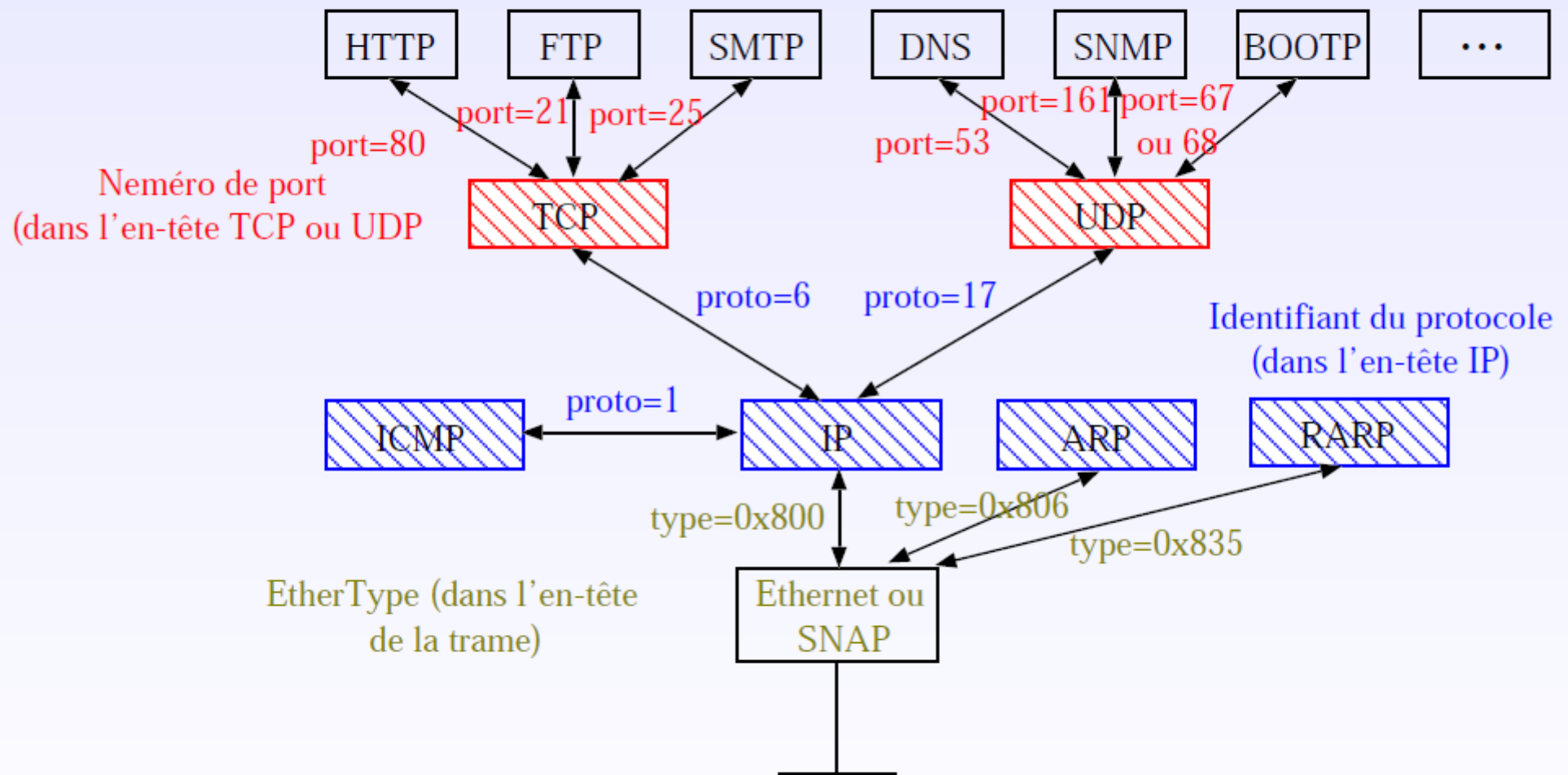
TCP - protocole de transport **de bout en bout**

- ★ uniquement présent aux **extrémités**
- ★ transport **fiable** de **segments** (mode **connecté**)
- ★ protocole complexe (retransmission, gestion des erreurs, séquençement, ...)

L'architecture de TCP/ IP (7)



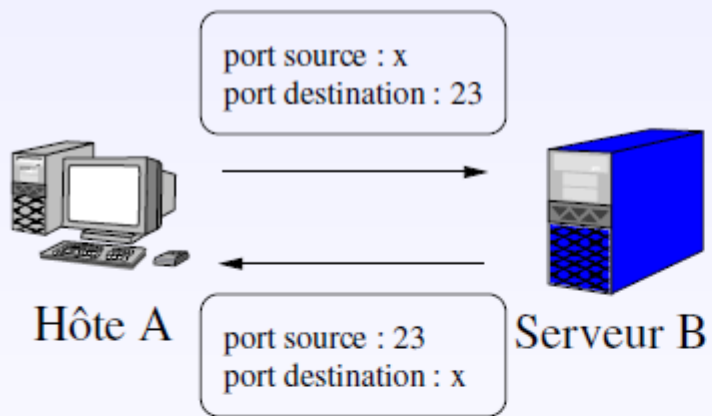
Identification des protocoles (1)



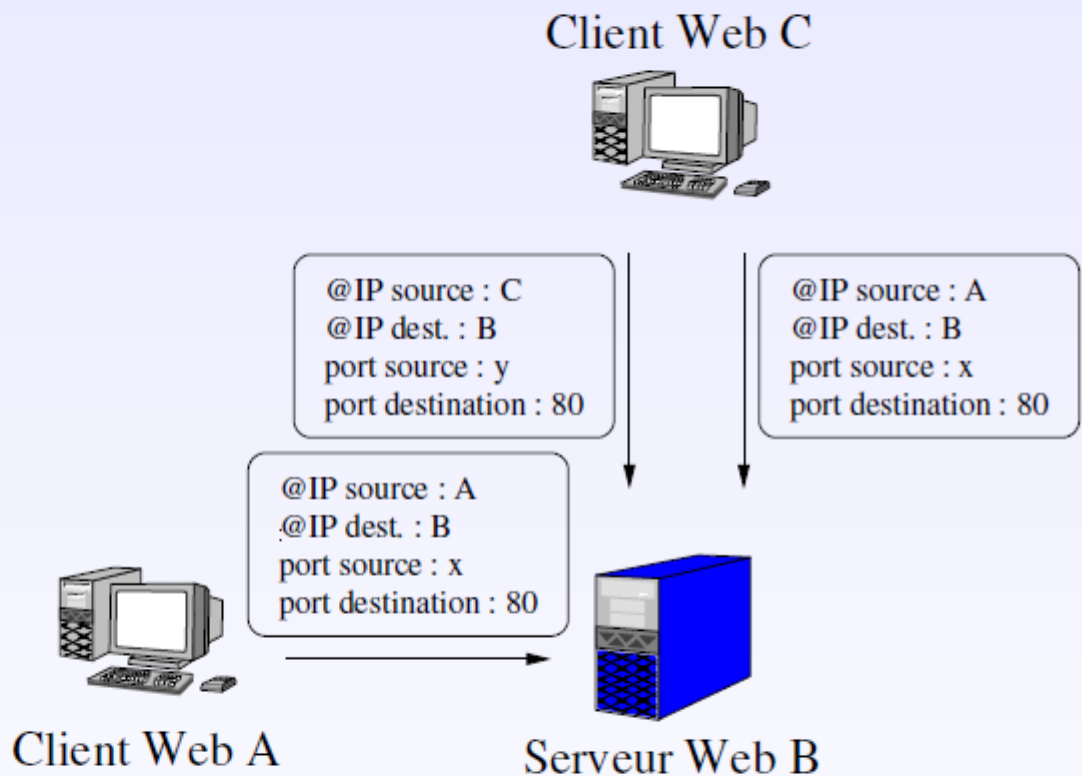
Identification des protocoles (2)

- ★ Une adresse de transport = une adresse IP + un numéro de port (16 bits) → adresse de socket
- ★ Une connexion s'établit entre une socket source et une socket destinataire → une connexion = un quintuplé (proto, src, port src, dest, port dest)
- ★ Deux connexions peuvent aboutir à la même socket
- ★ Les ports permettent un multiplexage ou démultiplexage de connexions au niveau transport
- ★ Les ports inférieurs à 1024 sont appelés **ports réservés**

Identification des protocoles (3)



Telnet Simple



Serveur Web

Le protocole TCP

Transport Control Protocol (RFC 793, 1122, 1323, 2018, 2581)

Attention: les RFCs ne spécifient pas tout - beaucoup de choses dépendent de l'implémentation

Transport fiable en mode connecté

- ★ point à point, bidirectionnel : entre deux adresses de transport (@IP src, port src) → (@IP dest, port dest)
- ★ transporte un flot d'octets (ou flux)
 - ▶ l'application lit/écrit des octets dans un tampon
- ★ assure la délivrance des données en séquence
- ★ contrôle la validité des données reçues
- ★ organise les reprises sur erreur ou sur temporisation
- ★ réalise le contrôle de flux et le contrôle de congestion (à l'aide d'une fenêtre d'émission)

Les utilisations d'UDP

- ★ Performance sans garantie de délivrance
- ★ Souvent utilisé pour les applications multimédias
 - ▶ tolérantes aux pertes
 - ▶ sensibles au débit
- ★ Autres utilisations d'UDP
 - ▶ applications qui envoient peu de données et qui ne nécessitent pas un service fiable
 - ▶ exemples : DNS, SNMP, BOOTP/DHCP

Transfert fiable sur UDP

- ▶ ajouter des mécanismes de compensation de pertes (reprise sur erreur) au niveau applicatif
- ▶ mécanismes adaptés à l'application

Adressage IP

- Adressage codé sur 4 octets noté en décimale pointé A.B.C.D
 - Les adresses réseau (préfixes) sont assignées globalement (autorités de l'Internet) et identifient le réseau
 - classe A et B par l'ICANN (Internet Corporation for Assigned Names and Numbers) Internet
 - classe C en France : NIC France (AFNIC)
 - Les adresses machine (suffixes) sont assignées localement et identifient la machine sur le réseau
- L'adressage :
 - Configuration manuelle par le logiciel
 - Autoconfiguration : DHCP

Adressage IP

- Quelle taille pour le préfixe et le suffixe ?
- Réponse : la taille des réseaux
- Les adresses IP sont divisées en 3 classes primaire
 - A, B et C
- Et 2 classes secondaires
 - D et E
- Les 4 premiers bits des 4 octets spécifient la classe à laquelle appartient l'adresse
- Les classes sont décomposables en sous-réseaux
- Epuisement des adresses IPv4
 - projection pessimiste : juillet 2012, puis décembre 2012, puis ? ...
 - reste une petite vingtaine de préfixes libres à l'IANA ...

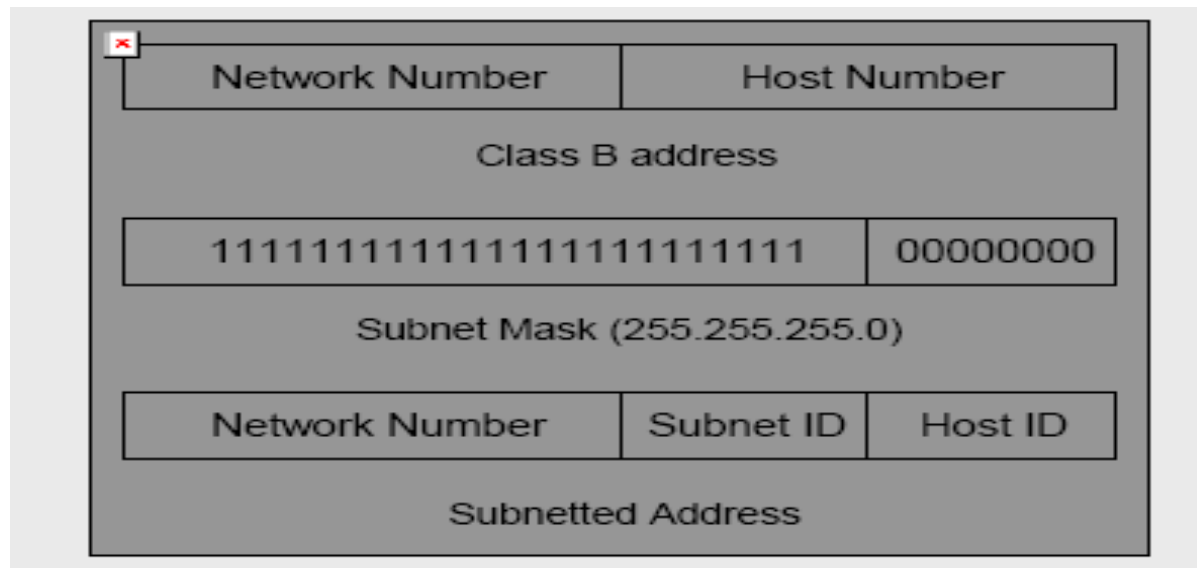
Sous Adressage IP

- Ajout de niveaux hiérarchiques
 - Découpage d'un réseau en entités plus petites
 - Sous-réseaux ou *subnets* permettant une structuration d'un site
- Le concept utilisé pour la mise en oeuvre du sous-adressage IP est celui des masques (*subnet mask*)
- Les sous-réseaux ne sont visibles qu'à l'intérieur d'un réseau ou de la classe correspondante
 - Stratégie décidée localement par l'administrateur
 - Le découpage est totalement invisible depuis l'extérieur
 - Interconnexion des sous-réseaux impérative par des routeurs



Sous Adressage IP

- Principe de l'utilisation des masques :
 - permet la définition de partitions variables des adresses de réseaux de classe A, B ou C.
 - longueur en bits décidée par l'administrateur
 - adresse de sous-réseau prélevée sur la partie réservée à l'adresse machine
 - tous les équipements du réseau doivent utiliser la même notion de sous-réseau



Sous Adressage IP

- Exemple 1 en classe B :
 - 6 sous réseaux demandés
 - $2^3 = 6$ sous réseaux \Rightarrow 3 bits nécessaires pour les coder
 - Donc masque à 3 bits
 - Le masque sera égal à 1110 0000, c'est à dire 224 en décimal
- Classe B :
 - Masque par défaut : 255.255.0.0
 - Le nouveau masque sera donc 255.255.224.0

Sous Adressage IP

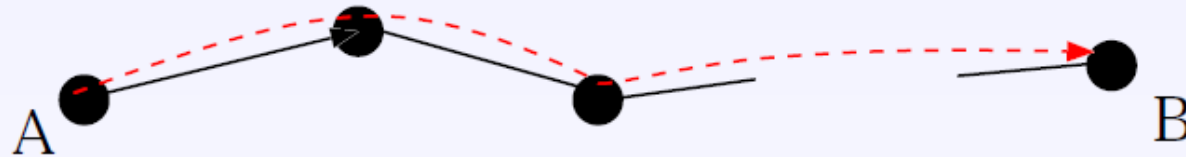
Exemple 2 sur une classe B

- 141.14.0.0 : adresse d'un réseau de classe B
- masque par défaut : 255.255.0.0
 - pas de sous-réseau
- 255 sous réseaux demandés
 - 8 bits nécessaires pour coder 255
 - masque sur 8 bits soit 11111111 donc 255 converti en décimal
- masque 255.255.255.0
 - présence d'au plus 255 sous-réseaux

Problématique du routage

Objectif : Acheminer des datagrammes IP d'une machine source A vers une machine destination B.

Problématique : Comment atteindre la machine B en connaissant son adresse IP?



→ Nécessité d'identifier toutes les machines intermédiaires.

Routage IP : principe de base

Définition :

- ★ Processus de choix des chemins par lesquels les paquets sont transmis à la machine destinataire
- ★ Processus basé sur une table de routage *IP routing table* contenant les informations relatives aux différentes destinations possibles et à la façon de les atteindre
- ★ Exemple : netstat -r (sous UNIX)

Principe de base :

- ★ L'émetteur ne connaît pas la route complète mais l'adresse du prochain site IP qui le rapprochera de la destination (prochain saut)
- ★ Simplicité des tables de routage
- ★ Changements dynamiques possibles (en cas de pannes par exemple)

Routage IP : algorithme de base (1/2)

- ★ Extraire du datagramme l'adresse IP de destination (*IPDest*)
- ★ Calculer l'adresse du réseau de destination (*IPRes*)
- ★ Si cette adresse *IPRes* correspond à l'adresse réseau du réseau local :
 - ▶ *IPdest* est directement accessible sur le réseau élémentaire commun
 - ▶ La couche IP locale tente la translation adresse logique *IPdest* en une adresse physique à travers la table maintenue en cache
 - ▶ Si le réseau est de type Ethernet (Tokenring, ...), le protocole ARP est utilisé pour construire les éventuelles entrées manquantes dans la table et émettra le datagramme
 - ▶ Si le réseau est d'un autre type (Transpac, ...), les adresses physiques destinataires X21 auront dû être configurées à la main au préalable

Routage IP : algorithme de base (2/2)

- ★ Sinon (ce n'est pas une adresse accessible, il faut alors consulter la table de routage IP locale)
 - ▶ Si *IPres* est dans la table alors :
 - Router le datagramme selon les indications de la table (vers un autre nœud du réseau local, avec résolution adresse IP → adresse physique, ou vers un autre coupleur connecté à un réseau externe)
 - ▶ Sinon *IPres* n'est pas dans la table alors
 - Prendre la route par défaut indiquée dans la table
 - Router le datagramme selon les indications de l'entrée par défaut de la table (vers un autre nœud du réseau local, avec résolution adresse IP → adresse physique, ou vers un autre coupleur connecté à un réseau externe)

Tables de routage IP dans Linux

La consultation/modification de la table de routage peut être faite avec la commande `route`.

Exemple :

Destination	Passerelle	Genmask	Indic	Metric	Ref	Use	Iface
147.210.20.0	*	255.255.255.0	U	0	0	0	eth0
default	vlan2.labri.fr	0.0.0.0	UG	0	0	0	eth0

Cette table de routage montre que :

- ★ Notre hôte peut dialoguer directement avec les machines faisant partie du réseau 147.210.20.0/24
- ★ La route par défaut le fait passer par la passerelle `vlan2.labri.fr`

Tables de routage IP dans Linux

La consultation/modification de la table de routage peut être faite avec la commande **route**.

Exemple :

```
route add default gw @passerelle (ajouter une route par défaut)
```

```
route add -host @hôte gw @passerelle dev iface (ajouter une route utilisant l'interface réseau iface vers un hôte particulier)
```

```
route add -net @réseau netmask masque dev iface gw @passerelle (ajouter une route utilisant l'interface iface vers un réseau particulier)
```

Pour les suppressions de route, il suffit de remplacer l'opération **add** par **del**.

Linux et routage

- ★ Support pour le routage disponible dans le noyau Linux.
- ★ Nécessité d'activer la fonctionnalité de “relais” dans le noyau.
Deux méthodes peuvent être utilisée :

- ▶ Modifier “à chaud” le paramètre qui contrôle la fonctionnalité :
`echo 1 > /proc/sys/net/ipv4/ip_forward`
- ▶ Configuration automatique à chaque démarrage :
Ajout de `net.ipv4.ip_forward=1` au fichier
`/etc/sysctl.conf`

→ La machine en question fera alors office de relais entre ses différentes interfaces réseau.