Cours 4

Couche Transport (TCP)

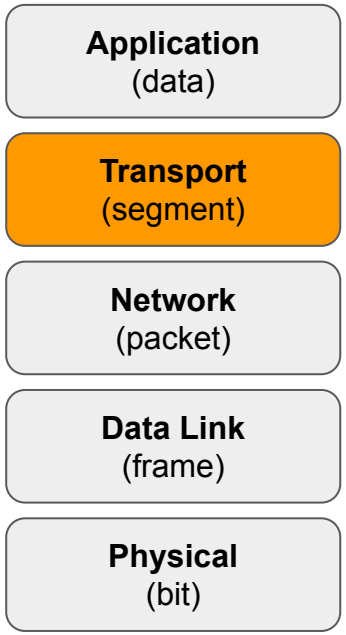
Introduction

La couche réseau (IP)

- Communication de bout-en-bout entre deux machines sur Internet
- Transfert de paquet en “best-effort” (non fiable)

La couche transport

- Communication de bout-en-bout entre deux applications (processus).



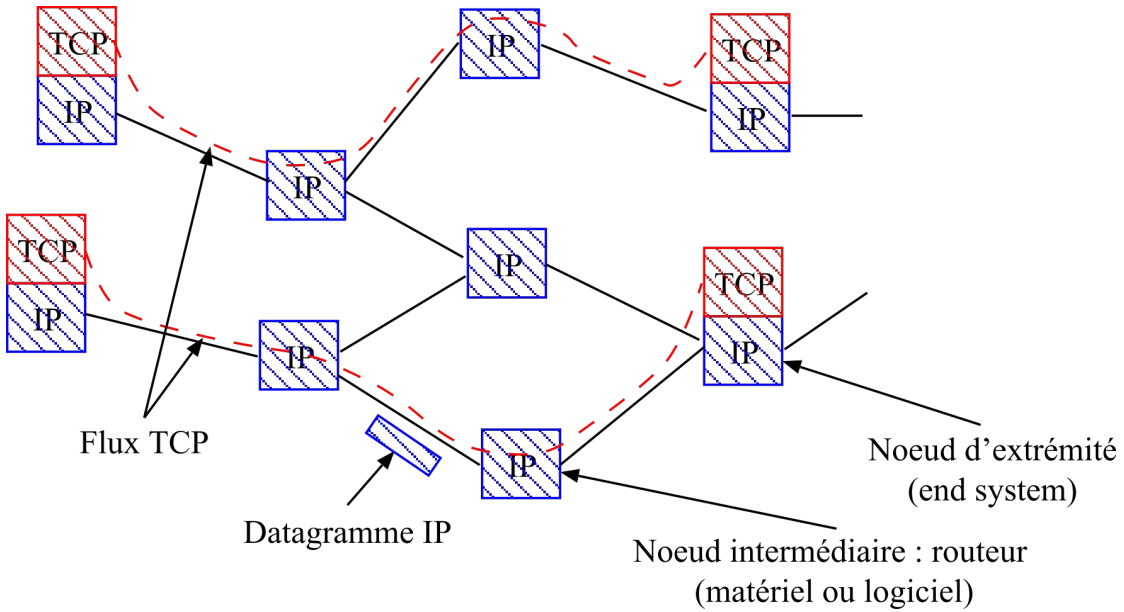
Les deux principaux protocoles de transport

- **TCP** (Transmission Control Protocol) : orienté connexion, fiable.
- **UDP** (User Datagram Protocol) : sans connexion, non fiable, rapide.

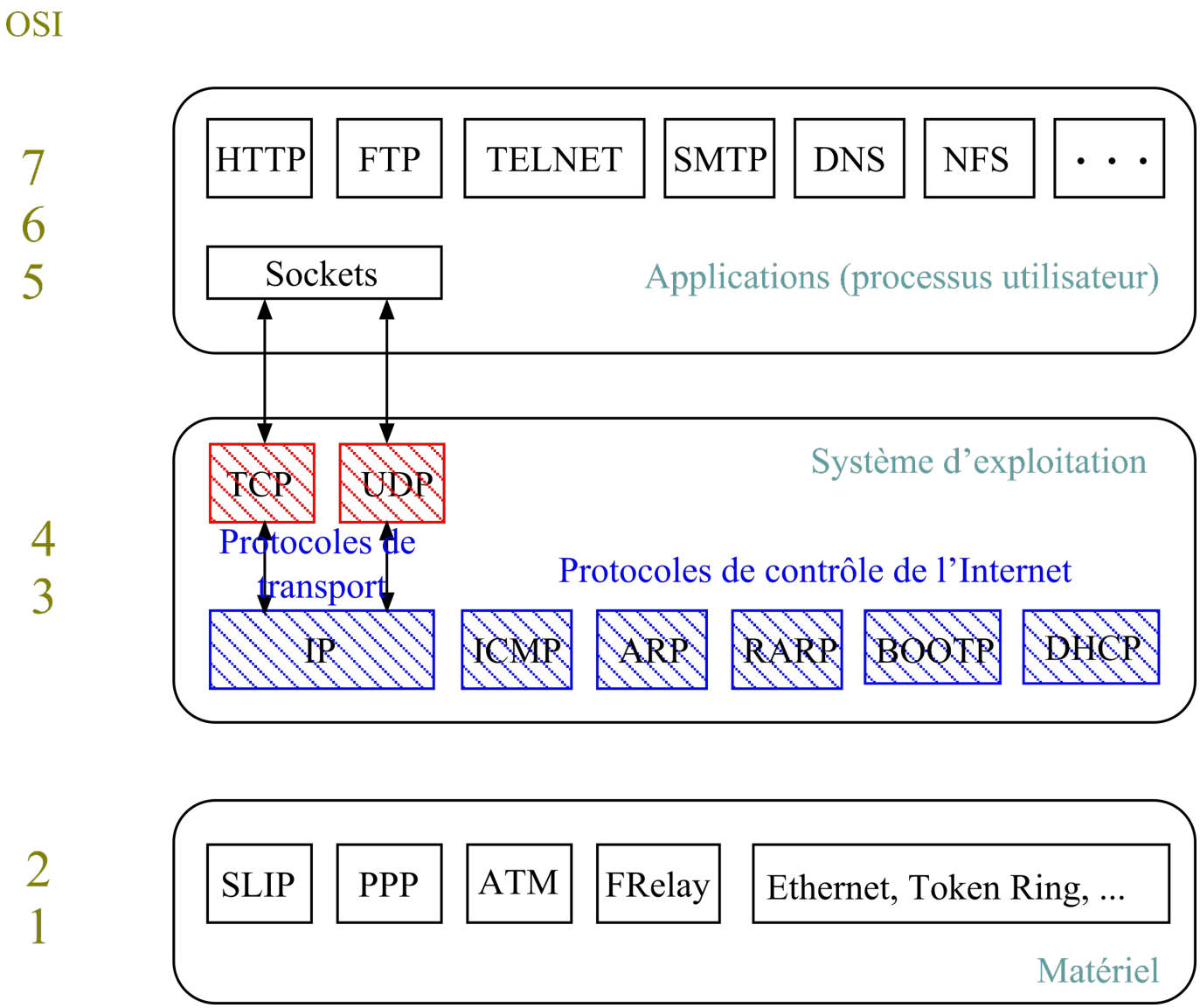
Le Protocole TCP

Caractéristiques

- uniquement présent aux extrémités
- conversation bidirectionnelle en mode connecté
- transport fiable de segments (séquencement)
- protocole complexe : retransmission, gestion des erreurs, congestion, …

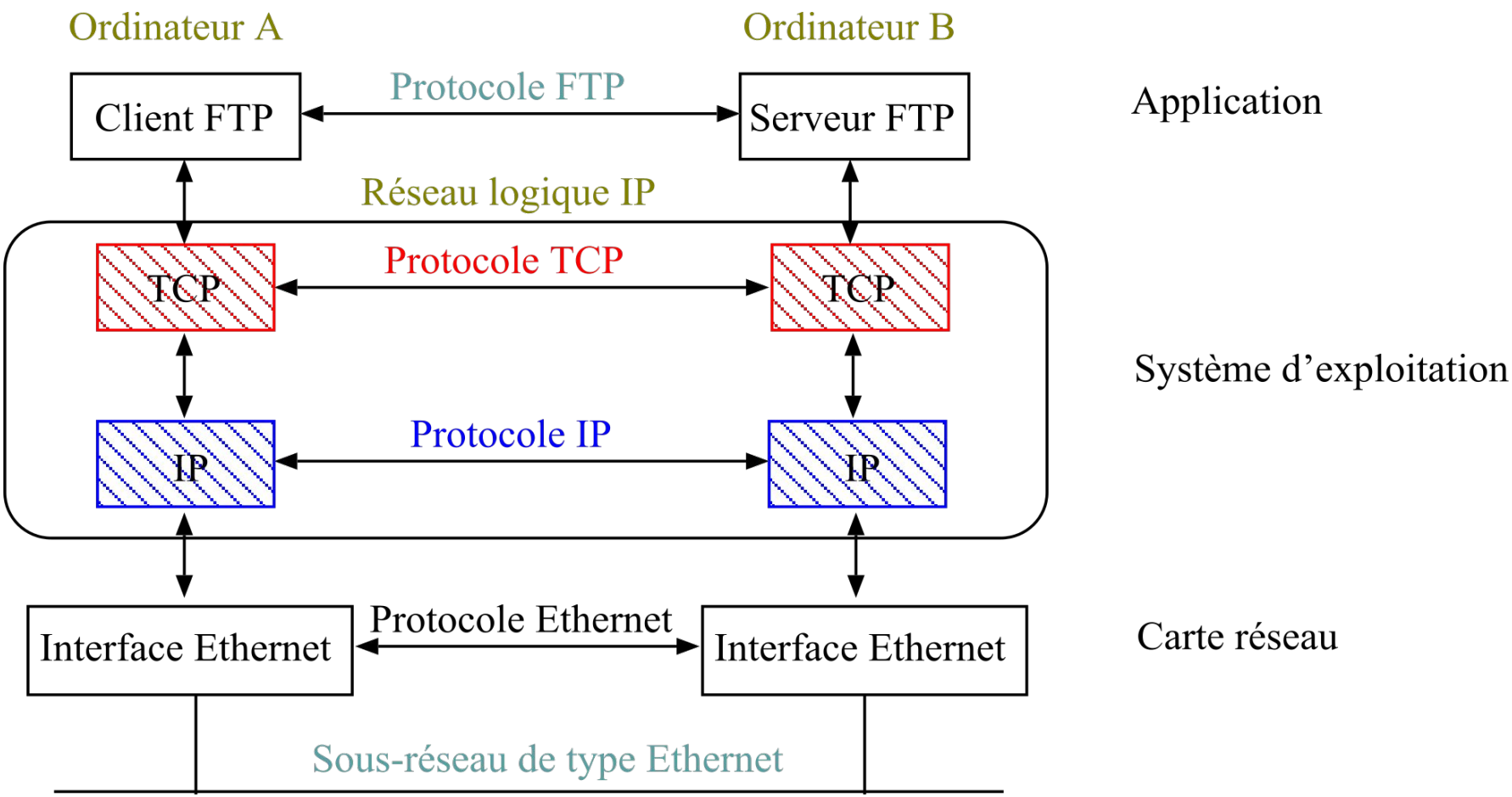


Pile de Protocoles



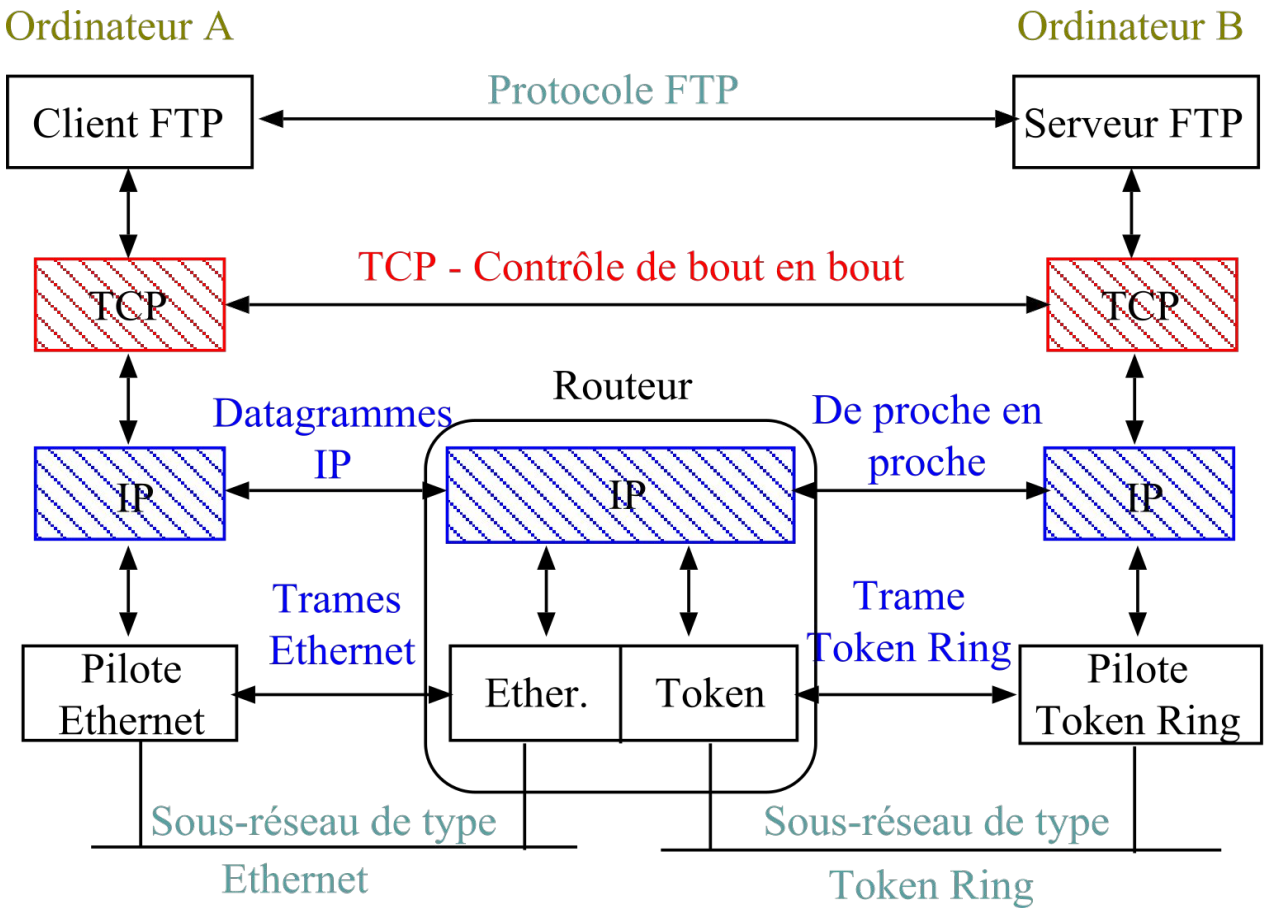
Pile de Protocoles

Deux machines dans un même réseau local et homogène...



Pile de Protocoles

Deux machines dans des réseaux distants et hétérogènes...

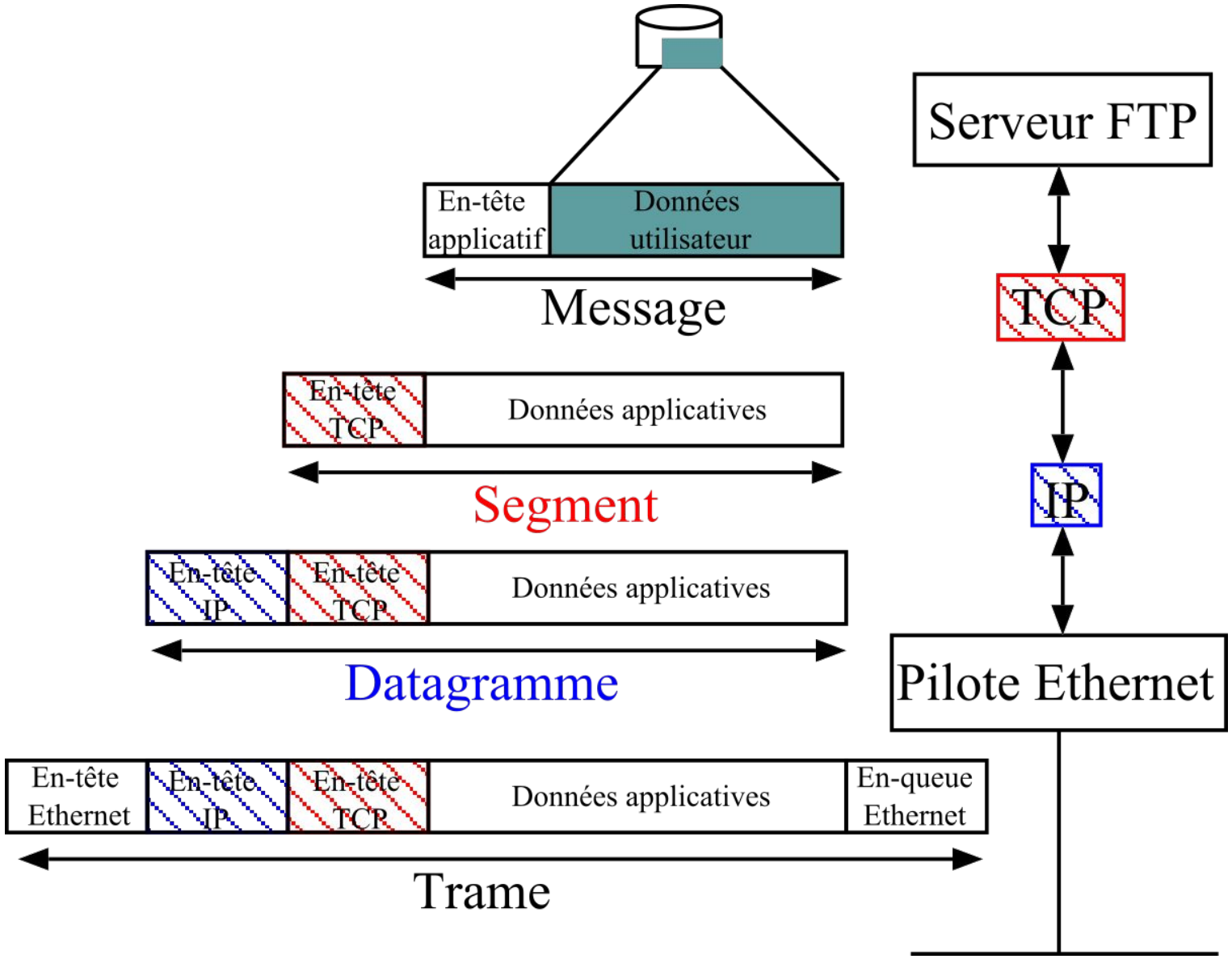


Application

Système d'exploitation

Carte réseau

Encapsulation



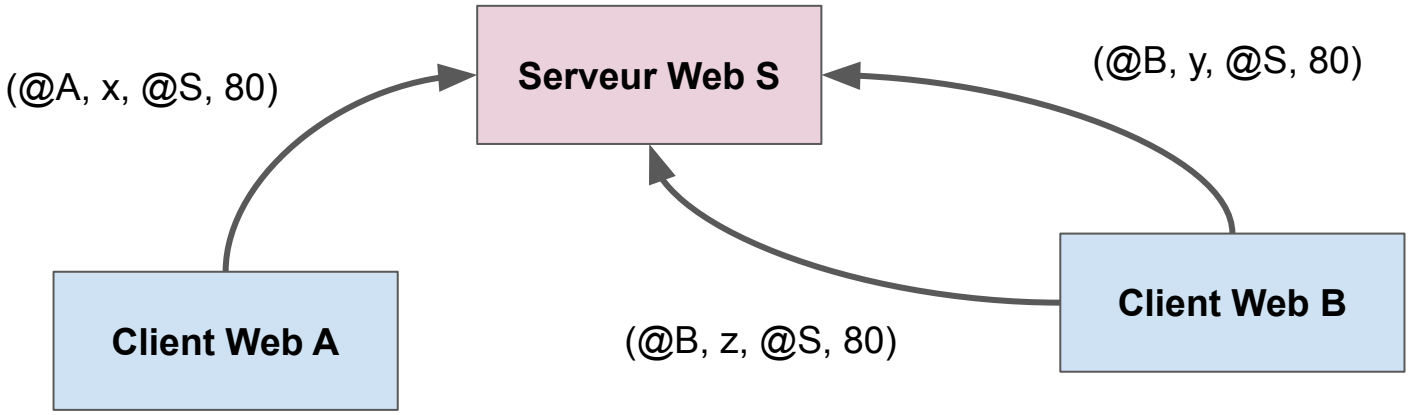
Numéro de Ports et Connexion

Adresse de Transport : une adresse IP (32 bits) + un numéro de port (16 bits)

Une connexion point-à-point : un quadruplet ($@IP_{src}$, $\#Port_{src}$, $@IP_{dest}$, $\#Port_{dest}$)

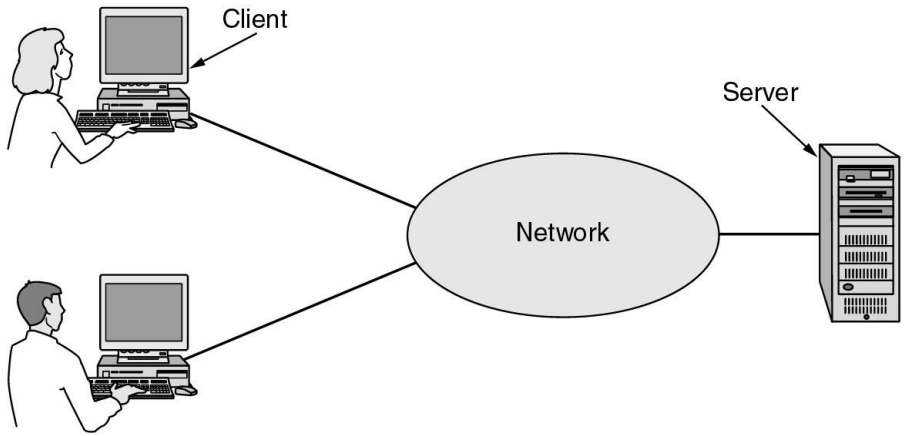
Numéro de Port (< 65535)

- Les ports permettent un multiplexage de connexions au niveau transport.
- Les services standards utilisent des numéros de ports réservés, inférieurs à 1024. Par exemple : web → 80.
- Le numéro de port désigne un processus et un seul dans le système.
- Le client utilise le plus souvent un numéro de port aléatoire.



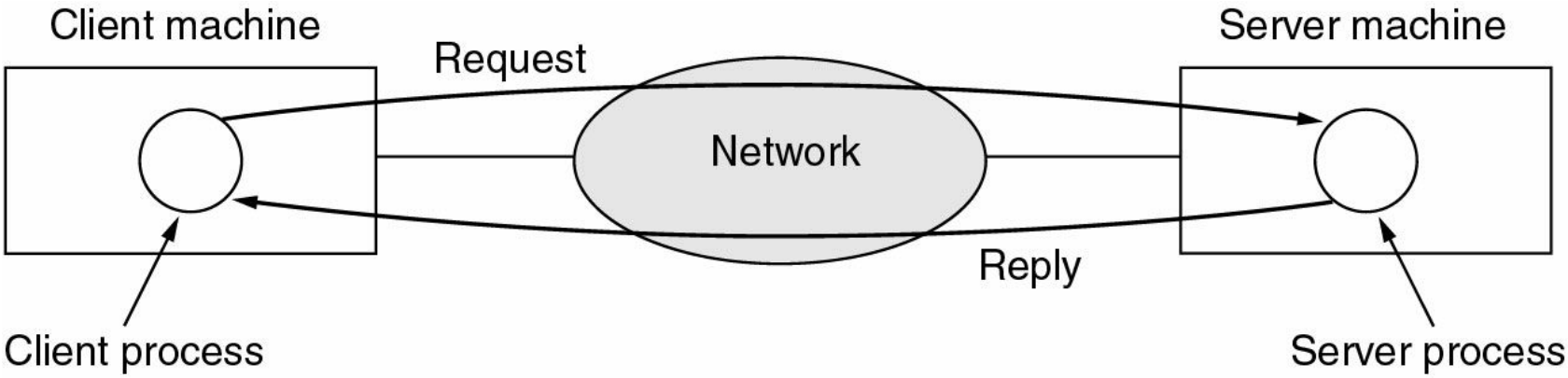
Nota Bene : *y* et *z* doivent être différents pour arriver à distinguer les connexions entre B et S !

Modèle Client-Serveur (TCP/IP)



- Un serveur S est une application, qui offre un service réseau à de multiples clients.
- Le serveur S est à l'écoute (*listen*) sur un port P des demandes de connexion des clients.
- Pour utiliser le service de S, un client C doit initier une demande de connexion auprès de S sur le port P.
- Plusieurs clients peuvent être connectés simultanément à un même serveur.

Modèle Client-Serveur (TCP/IP)



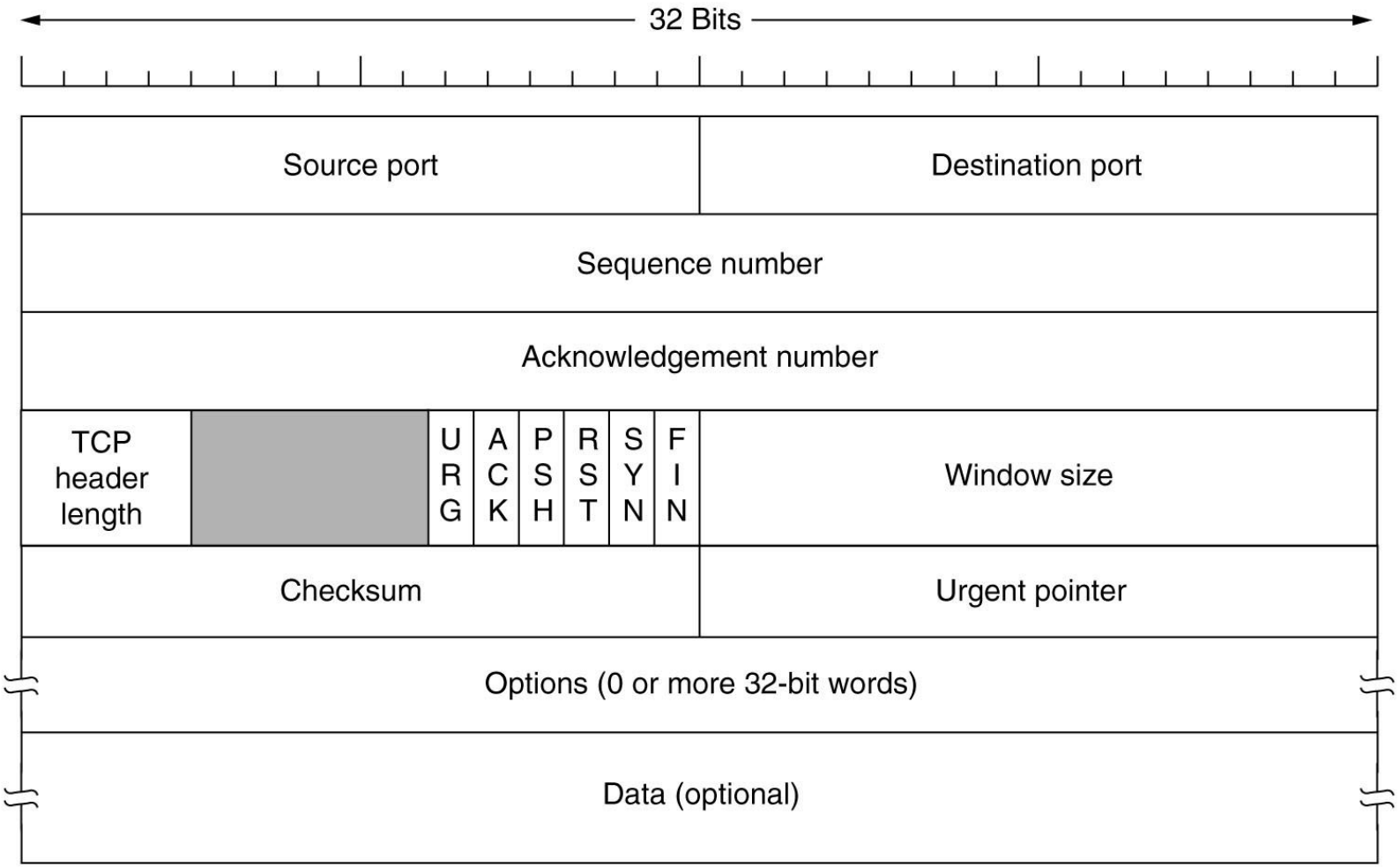
- Une fois la connexion établie (*established*) démarre une session d'échange de messages entre C & S.
- La communication C/S est bidirectionnelle, mais utilise le plus souvent le modèle requête-réponse.
  - Web : Le client effectue une requête HTTTP GET d'une certaine page HTML...
- Plusieurs requêtes & réponses peuvent s'effectuer durant la session… avant la déconnexion.
- La session se termine à la demande de déconnexion du client ou du serveur.

Services Standards

Quelques services standards de TCP

- 21 : FTP (File Transfer Protocol)
- 22 : SSH
- 23 : Telnet
- 25 : SMTP
- 69 : TFTP
- 80 : HTTP
- 110 : POP3 (Post Office Protocol)
- 123 : NTP (Network Time Protocol)
- 143 : IMAP (Internet Message Access Protocol)
- 194 : IRC
- 443 : HTTPS (HTTP Secure)

En-Tête TCP


32 Bits
Source port
Destination port
Sequence number
Acknowledgement number
TCP header length
URG
ACK
PSH
RST
SYN
FIN
Window size
Checksum
Urgent pointer
Options (0 or more 32-bit words)
Data (optional)

En-Tête TCP

- **Source Port** : numéro de port source [16 bits]
- **Destination Port** : numéro de port destination [16 bits]
- **Sequence Number**: numéro de séquence du premier octet de ce segment [32 bits]
- **Acknowledgement Number** : numéro de séquence du prochain octet attendu [32 bits]
- **Header Length** : longueur de l'en-tête en mots de 32 bits [4 bits]
- **Flags** (binaires) [6 bits]
  - ACK : le paquet est un accusé de réception
  - SYN : demande d'établissement d'une connexion
  - FIN : interruption de la connexion
  - RST : réinitialisation ou rejet de la connexion (*reset*)
  - PSH : données à recevoir tout de suite
  - URG : paquet à traiter de manière urgente
- **Window Size** : nombre d'octets souhaités pour la réception (0 pour stopper temporairement la transmission) [16 bits]
- **Checksum** : somme de contrôle calculée sur l'en-tête et les données [16 bits]
- **Urgent Pointer** [16 bits]
- **Options** : facultatives...

Checksum sur 16 bits

Méthode utilisée par IP / TCP / UDP

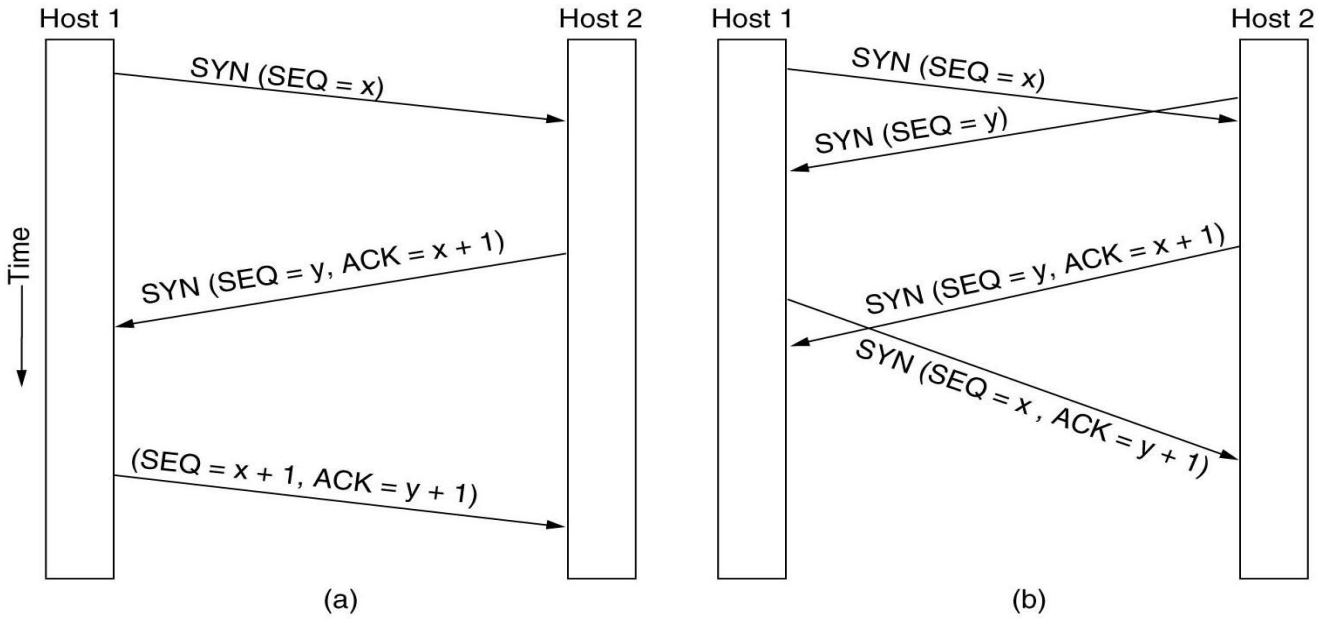
- Considérons les données suivantes : `D = 4500 0073 0000 4000 4011 c0a8 0001 c0a8 00c7`
- On ajoute les données par mots de 16 bits : `4500 + 0073 + 0000 + 4000 + 4011 + c0a8 + 0001 + c0a8 + 00c7 = 2 479c`
- On ajoute la retenue : `479c + 2 = 479e`
- La checksum `C(D)` est alors le complément à 1 : `K = ~479e = b861`
- Pour contrôler la checksum, on vérifie `C(D|K) = 0000`

***Checksum TCP** : The 16-bit checksum field is used for error-checking of the TCP header, the payload and an IP pseudo-header. The pseudo-header consists of the source IP address, the destination IP address, the protocol number for the TCP protocol (6) and the length of the TCP headers and payload (in bytes).*

Établissement de Connexion

La *poignée de main* TCP en 3 étapes

- Synchronisation des numéros de séquence



Connexion standard

Connexion simultanée

Lister les Connexions

netstat : lister les services à l'écoute et les connexions en cours sur ma machine...

```
$ netstat -tanp
Proto      R-Q     S-Q          Local Address          Foreign Address    State       PID/Program name
tcp        0       0            127.0.0.1:2208         *:*                LISTEN      3266/hpiod
tcp        0       0            127.0.0.1:34818        *:*                LISTEN      3275/python
tcp        0       0            127.0.0.1:3306         *:*                LISTEN      3642/mysqld
tcp        0       0            0.0.0.0:25             *:*                LISTEN      3525/exim4
tcp        0       0            82.225.96.37:35551     147.210.8.143:993  ESTABLISHED 10503/mozilla
tcp        0       0            82.225.96.37:39243     147.210.13.65:22   ESTABLISHED 13758/ssh
tcp        0       0            82.225.96.37:35750     147.210.9.15:22    ESTABLISHED 13763/ssh
tcp6       0       0            *:80                   *:*                LISTEN      3979/apache2
tcp6       0       0            *:22                   *:*                LISTEN      3746/sshd
tcp6       0       0            *:25                   *:*                LISTEN      3525/exim4
```

Les principaux états d'une connexion TCP/IP

- LISTEN : un service à l'écoute
- ESTABLISHED : une connexion établie
- CLOSED : connexion fermée

State	Description
CLOSED	No connection is active or pending
LISTEN	The server is waiting for an incoming call
SYN RCVD	A connection request has arrived; wait for ACK
SYN SENT	The application has started to open a connection
ESTABLISHED	The normal data transfer state
FIN WAIT 1	The application has said it is finished
FIN WAIT 2	The other side has agreed to release
TIMED WAIT	Wait for all packets to die off
CLOSING	Both sides have tried to close simultaneously
CLOSE WAIT	The other side has initiated a release
LAST ACK	Wait for all packets to die off

Scan d’un Réseau

nmap : outil permettant de découvrir les machines “en vie” dans un réseau, et les services disponibles sur une machine !

Un port peut être ouvert, fermé, ou filtré (cas d’un firewall).

Exemple de scan dans mon réseau domestique

```
# ping sweep
$ nmap -sP -n 192.168.0.0/24

Nmap scan report for 192.168.0.1 => Host is up (0.0035s latency).
Nmap scan report for 192.168.0.100 => Host is up (0.10s latency).
Nmap scan report for 192.168.0.101 => Host is up (0.036s latency).
Nmap scan report for 192.168.0.106 => Host is up (0.00026s latency).
Nmap scan report for 192.168.0.10 => Host is up (0.0064s latency).
Nmap scan report for 192.168.0.11 => Host is up (0.0026s latency).
Nmap scan report for 192.168.0.50 => Host is up (0.013s latency).
Nmap scan report for 192.168.0.254 => Host is up (0.0030s latency).
Nmap done: 512 IP addresses (8 hosts up) scanned in 4.56 seconds
```

Scan d’un Réseau

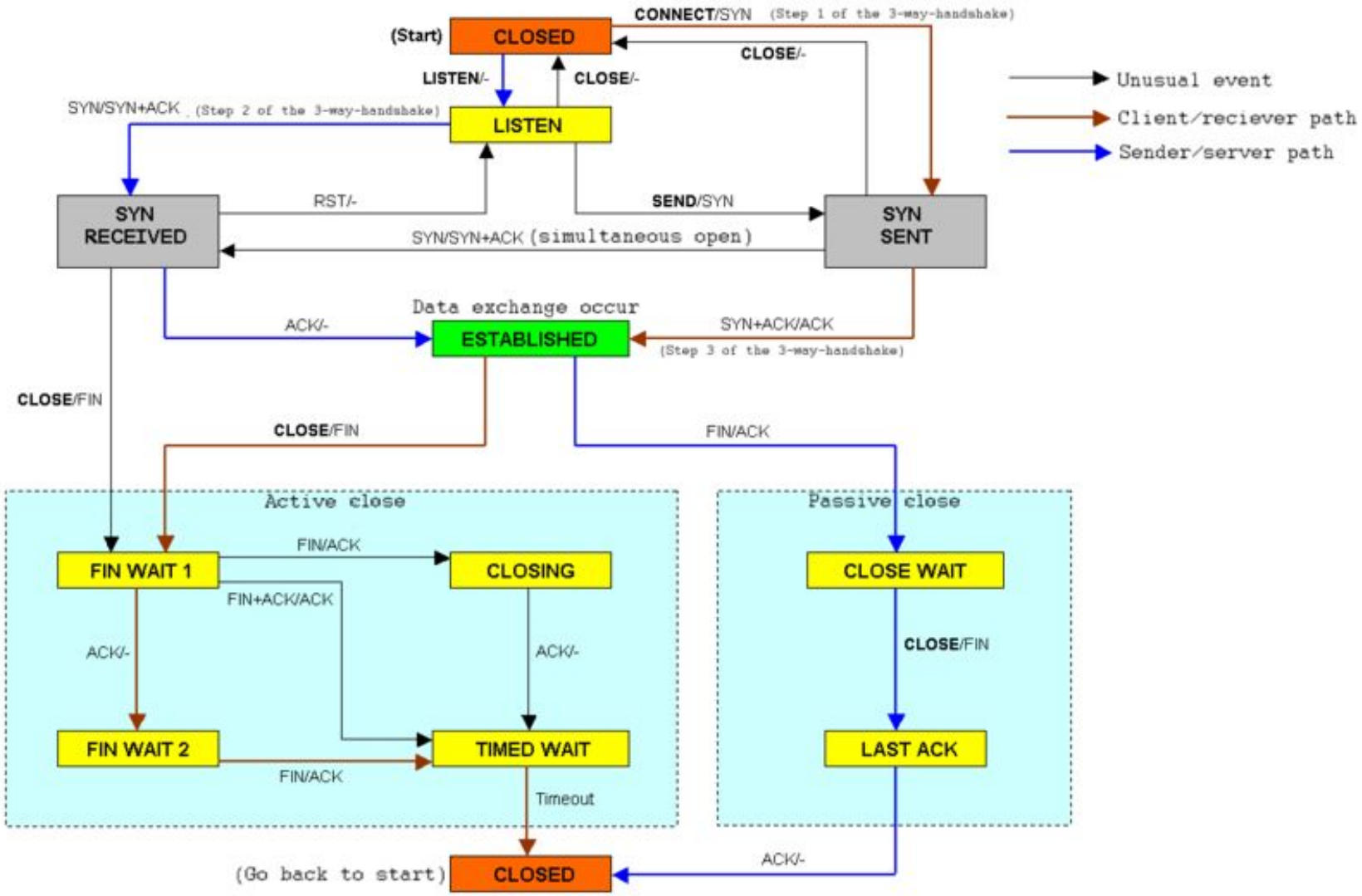
```
# basic scan
$ nmap 192.168.0.254

Nmap scan report for 192.168.0.254
Host is up (0.0031s latency).
Not shown: 985 filtered ports
PORT     STATE  SERVICE
21/tcp   closed ftp
53/tcp   open   domain
80/tcp   open   http
139/tcp  open   netbios-ssn
443/tcp  open   https
445/tcp  open   microsoft-ds
548/tcp  closed afp
554/tcp  open   rtsp
1723/tcp closed pptp
5000/tcp open   upnp
5001/tcp closed commplex-link
5678/tcp open   rrac
6000/tcp closed X11
8090/tcp open   opsmessaging
9091/tcp open   xmltec-xmlmail
```

```
# syn scan (root privilege required)
$ nmap -sS 192.168.0.100 -p 1-100

Nmap scan report for 192.168.1.11
Host is up (0.0045s latency).
Not shown: 96 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
80/tcp open  http
```

Un Protocole Complexe !



E/M : Lorsque l'evènement E se produit, envoyé le message M ou ne rien faire si M='-'.

UDP

User Datagram Protocol (UDP)

- sans connexion, numéro de port comme TCP
- pas de contrôle de flux, de contrôle d'erreurs, de retransmission
- transfert simple et rapide, mais non fiable
- Exemples : RTP (Real-time Transport Protocol), DNS, ...