

Niveau 4

TCP – UDP

UDP ou TCP ?

- IP est « Best effort »
- Adresse: numéro de port (16 bits)
- TCP
 - Connexion oriented
 - Fiable
 - Complexe
 - Similaire au téléphone
- UDP
 - Packet switching oriented (connexionless)
 - Non Fiable
 - Simple
 - Similaire à la poste

Ports

- Réservé: 0 – 1023 (IANA)
- Disponibles: les autres (1024 – 65535)
- Ceci, c'est la théorie, la pratique est légèrement différente !

TCP – Caractéristiques

- Transmission Control Protocol (pas ‘Transport’)
- RFC 793
- Basé sur la notion de flux (les données sont vues comme un flux, pas comme des paquets)
- Fiable
- Permet un contrôle de flux
- Full duplex
- Basé sur des numéros de séquence

Paquet TCP

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Source port																Destination port															
Sequence Number																															
Acknowledgement Number																															
Data Offset		Reserved				U R G	A C K	P S H	R S T	S Y N	F I N	Window Size																			
Checksum																Urgent Pointer															
Options																								Padding							
DATA . . .																															

TCP – Session

- Etablissement d'une session de bout en bout !
- Basé sur les 'sliding windows'
- Offre un transfert d'information fiable en:
 - renvoyant, si besoin est, des paquets perdus ou arrivant hors délais
 - reséquençant les paquets, s'ils arrivent dans un ordre différent de celui dans lequel ils ont été envoyés
- Ces opérations rendent le protocole complexe !

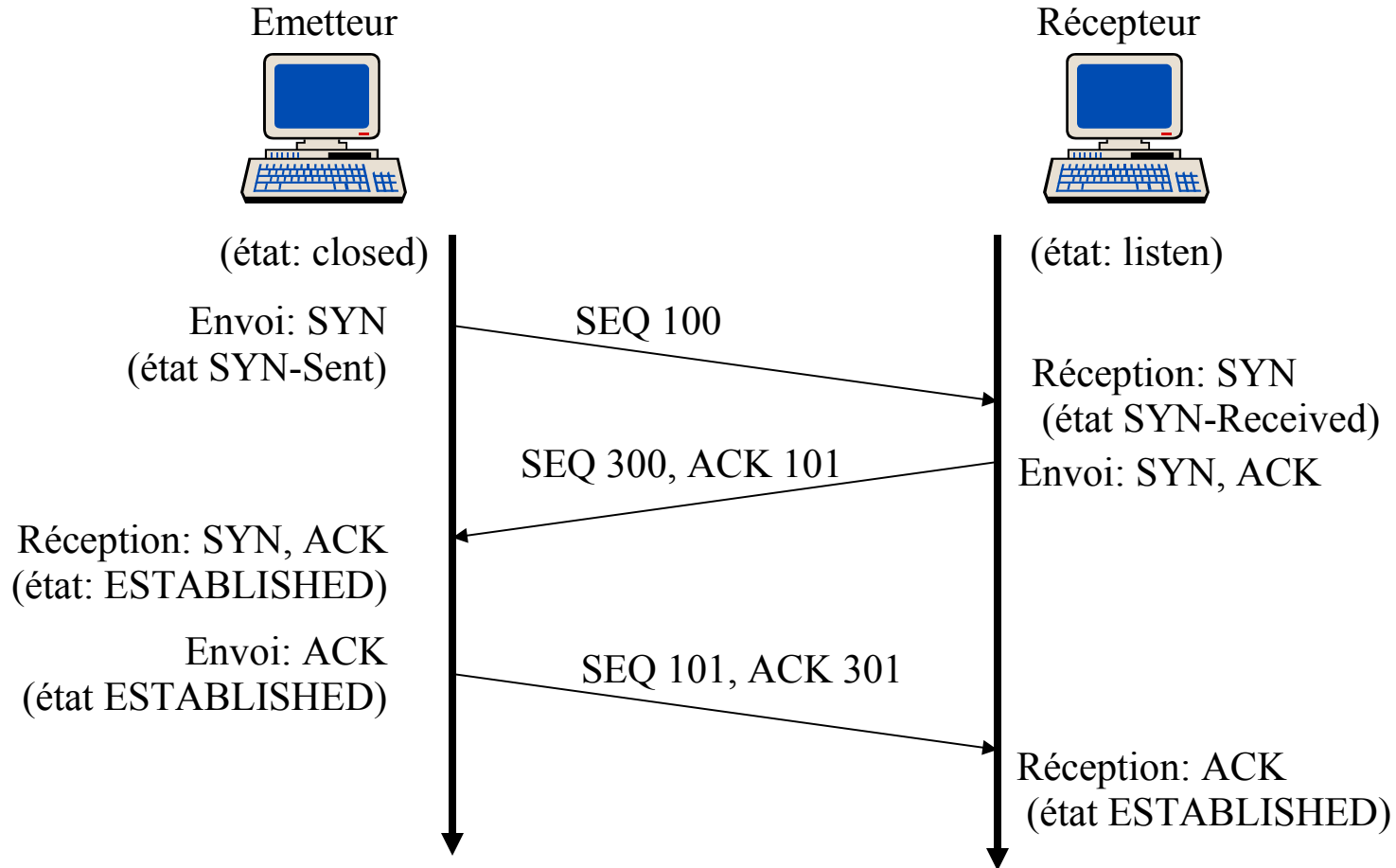
TCP – Seq & Ack numbers, Window Size

- Ce sont des nombres de 32 bits, indépendants l'un de l'autre, déterminant:
 - le numéro du premier byte du paquet envoyé (seq number)
 - Le numéro du dernier byte reçu (ack number)
- Le champ 'Window Size' indique le nombre de bytes disponibles pour recevoir des informations.
- Ces informations sont utilisées pour l'implémentation des « sliding windows » !

TCP – Exemple

- Transmission Control Protocol, Src Port: ftp (21), Dst Port: 1137 (1137), Seq: 607915605, Ack: 4403356
- Source port: **ftp** (21)
- Destination port: **1137** (1137)
- Sequence number: **607915605**
- Acknowledgement number: **4403356**
- Header length: **28** bytes
- Flags: 0x0012 (SYN, ACK)
 - 0... .. = Congestion Window Reduced (CWR): Not set
 - .0... .. = ECN-Echo: Not set
 - ..0. = Urgent: Not set
 - ...1 = Acknowledgment: Set
 - 0... = Push: Not set
 -0.. = Reset: Not set
 -1. = Syn: Set
 -0 = Fin: Not set
- Window size: **5840**
- Checksum: 0x7f70 (correct)
- **Options:** (8 bytes)
 - Maximum segment size: 1460 bytes
 - NOP
 - NOP
 - SACK permitted

TCP – 3 way handshake

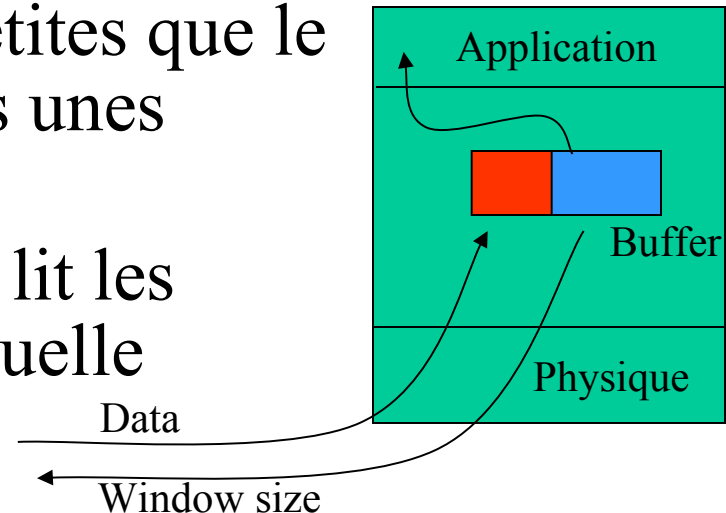


TCP – Contrôle de flux

- Idée: empêcher une machine rapide de ‘noyer’ une machine plus lente !
- Il faut donc un mécanisme permettant à la machine la plus lente de prévenir la plus rapide qu’elle ne sait plus suivre !

TCP – Sliding Window

- Idée: utiliser un buffer pour permettre le transfert de plusieurs trames l'une à la suite de l'autre.
- Si une trame est plus grosse que le buffer: problème !
- Si les trames sont plus petites que le buffer... on les stocke les unes derrière les autres !
- Idéalement, l'application lit les données à la vitesse à laquelle elles arrivent !



TCP – Principaux protocoles

- 20 – FTP (data)
- 21 – FTP (control)
- 22 – SSH
- 23 – Telnet
- 25 – SMTP
- 80 – HTTP
- 110 – POP3
- 143 – IMAP4
- 179 – BGP
- 646 – LDP

Paquet UDP

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32

Source port	Destination port
Length	Checksum
DATA . . .	

UDP

- User Datagram Protocol
- RFC 768
- On envoie et... on espère !
- N'offre que le checksum et... le multiplexage par port !
- Simple

UDP – Principaux protocoles

- 53 – DNS
- 67 – Serveur DHCP
- 68 – Client DHCP
- 69 – TFTP
- 123 – NTP
- 161 – SNMP
- 520 – RIP
- 521 – RIPng
- 5060 – SIP
- Port négocié en SDP (partie de SIP) – RTP