La capa de Transport

LA PILA OSI

Nivel de Aplicación

Servicios de red a aplicaciones

Nivel de Presentación

Representación de los datos

Nivel de Sesión

Comunicación en la dispusitivos de la rec

N. rel de Transporte

conexión extremo-a-extremo y bilidad de los dates

Ni el de Red

Determinación de ruta e IP (Direccionamiento lógico)

Nivel de Enlace de Datos

Direccionamiento físico (MAC y LLC)

Nivel Físico

Señal y transmisión binaria

La capa de transport

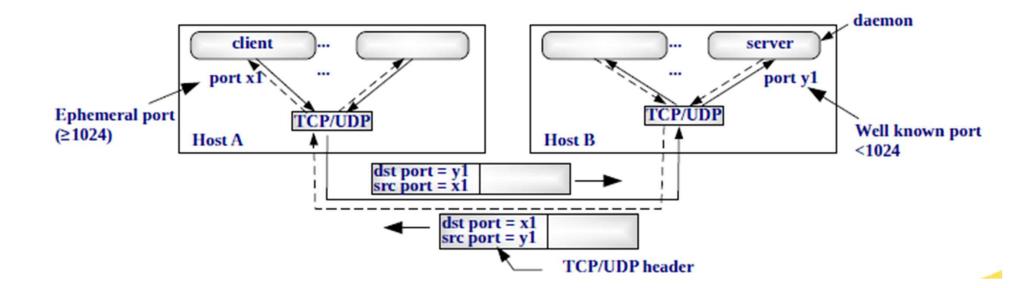
- La capa de transport la tenim per sobre de la capa de xarxa
- La capa de transport s'encarrega de proveir comunicacions lògiques entre processos i s'executa en equips finals.
- Assegura el control de flux i d'errors entre equips finals

LA PILA OSI Nivel de Aplicación Servicios de red a aplicaciones Nivel de Presentación Representación de los datos Nivel de Sesión Comunicación en dispusitivos Novel de Transporte conexión extremo-a-extremo bilidad de los date Ni el de Red Determinación de ruta e IP (Direccionamiento lógico) Nivel de Enlace de Datos Direccionamiento físico (MAC y LLC) Nivel Físico Señal v transmisión binaria

- Dos són els protocols que es fan servir bàsicament a la capa de transport:
 - User Datagram Protocol. Servei datagrama. No fiable
 - Transport Control Protocol. Servei fiable. QoS
- La capa de transport ofereix comunicació entre aplicacions extrem a extrem.
- Els punts d'accés de la capa de Transport estan identificats per un número de port de 16 bits.

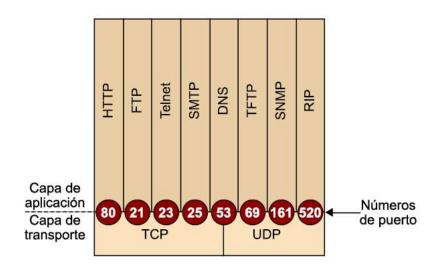
- En el cas de l'estructura TCP/IP, es defineixen dos adreces que relacionen el nivel de transport amb els nivells superiors i inferior:
 - L'adreça IP (32 bits), adreça lògica de la màquina
 - El port (16 bits), idenfitica l'apicació que requereix la comunicació
- Per identificar les diferents aplicacions, els protocols TCP i UDP associen cada paquet amb un identificador de 16 bits anomenat port

- Ports coneguts: Regulats per la IANA. Rang inferior a 1024 i son utilitzats per accedir a serveis oferts per servidors
- Ports efímers: Assignats de forma dinàmica pels clients dins d'un rang específic per sobre de 1023. Identifica el procés del client mentre duri la connexió.



Stack TCP/IP

TELNET, F	TP, SMTP	DNS, RIP, SNMP							
To	CP	UDP							
ARP	ı	Р	IGMP ICMP						
ETHERNET	TOKEN RING	FRAME RELAY	ATM						



TCP: Definicions I conceptes

- ▼ El protocol de transport (TCP) es defineix en la RFC 793
- És un servei fiable d'entrega de paquets orientats a la connexió. Garanteix l'entrega de paquets IP
 - Error Recovery
 - **ACKs**
 - Orientat a conexió
 - Control de Flux
- TCP segmenta I torna a unir els grans blocs de dades fent servir números de seqüència. Mida màxima de segment variable (MSS)
- Opera en full dúplex

TCP

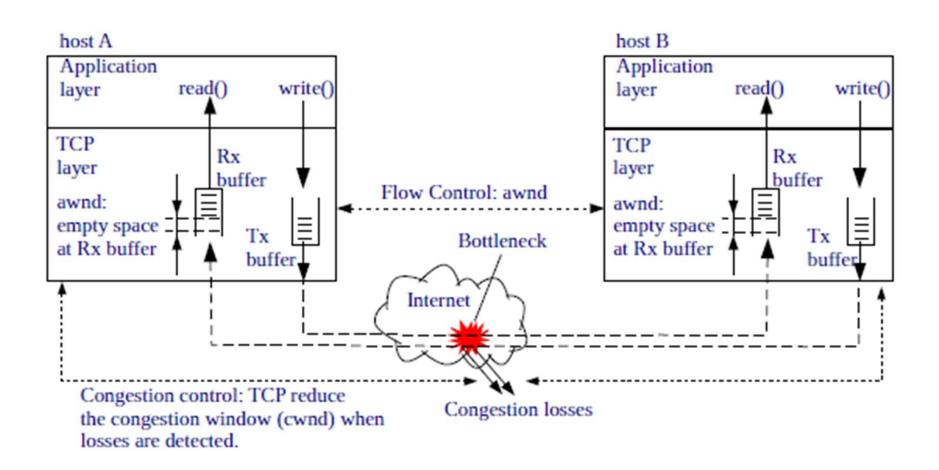
- TCP dona fiabilitat a l'aplicació ja que garanteix l'entrega de tota la información en el mateix ordre que ha estat transmesa. TCP proporciona un servei orientat a connexió amb control de Flux i errors
- TCP és un protocol ARQ extrem a extrem, orientat a connexió i bidireccional (dúplex)
- La unitat de dades en TCP és el SEGMENT. Intenta generar sempre segments de mida òptima denominats MSS (Maximum Segment Size) per minimitzar el sobrecost de les capceleres i que no generi fragmentació en IP

TCP – Operacions bàsiques

- ARQ implementat amb mida de finestra variable
- Cada segment de TCP rebut implica enviar un ACK sense esperar la lectura per part de la capa superior.
- Implementa dues finestres:
 - Advertised Window (awnd) utilitzada per control de flux
 - Congestion Window (cwnd) utilitzada per control de congestió
 - → La mida de la finestra per ARQ es calcula fent:

wnd = min(awnd, cwnd)

TCP – Operacions bàsiques



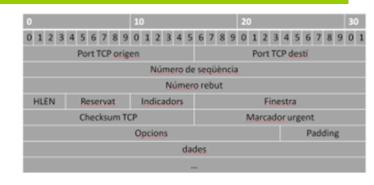
Format de TCP

- Transporta I regula el flux d'informació per garantir la connectivitat d'extrem a extrem entre aplicacions de host, que s'estan comunicant en la xarxa de forma confiable, eficient i precisa.
- QoS, fiable
- Protocol orientat a la connexió
- És un protocol de flux no estructurat (bytes de control I dades)
- Protocol amb transferència de memòria intermèdia
- **7** Fa servir connexions full-dúplex

Segment TCP

0	0 10										20									30						
0 1 2 3	4 5	6	7	8 9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Port TCP origen								Port TCP destí																		
Número de seqüència																										
Número rebut																										
HLEN Reservat Indicadors										Finestra																
Checksum TCP Marcador urgent																										
Opcions								Padding																		
dades																										
•••																										

- Port origen: 16 bits. Número del port origen
- Port destí: 16 bits. Número del port destí



Identifiquen l'app receptora i emisora, que juntament amb les addr IPs identifica de forma unívoca cada connexió.

Num. Seqüència: 32 bits. El número de seqüència del primer octet de dades d'aquest segment, excepte quan l'indicador de SYN el tenim a 1.

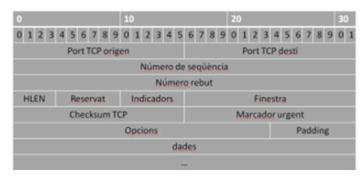
Si s'està establint la connexió => SYN = 1. El camp del num. de seqüència conté el ISN (Initial Sequence Number), valor escollit pel host per aquesta connexió i correspon al número de seqüència original i llavors, el primer byte de dades tindrà un num. de seq. igual a ISN+1

https://es.wikipedia.org/wiki/Anexo:Números_de_puerto

- Número rebut: 32 bits. Si el bit de control de ACK és 1, aquest camp conté el valor del següent número de seqüència que l'emissor del segment espera rebre. Un cop una connexió esta establerta, aquest número s'envia sempre.
- HLEN: 4 bits. Posició de les dades. Aquest número indica on comencen les dades. La capçalera TCP sempre té un número sencer de paraules de 32 bits.
- Reservat: 6 bits. Val 0

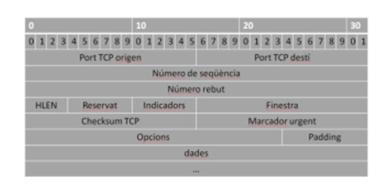


- Indicadors. 6 bits de control:
 - URG: Fa referència al camp que indica "marcador urgent"
 - ACK: Fa referència al camp : Num rebut
 - PSH: Funció de "Entregar dades immediatament" (push)
 - RST: Reset de la connexió
 - SYN: Sincronitzar els números de seqüència.
 - FIN: Darreres dades de l'emissor



TCP – Flags TCP

- Com hem comentat TCP implementa els següents flags:
 - URG (Urgent) -> Indica Punter a dades urgents. Apunta al primer (inicialment) byte urgent. No es fa servir gaire. Exemple CTRL + C en una sessió TELNET
 - ACK. Utilitzat sempre excepte en el primer segment enviat pel client.
 - PSH (Push). Qui envia indica "to push" les dades que estan al buffer a l'aplicació del receptor. El flag PSH s'activa quan el buffer està buit.
 - RST. Reset. Tallem la conexió
 - SYN (Utilitzat en el mecanisme de connexió)
 - **7** FIN. Termina la connexió



Finestra

Padding

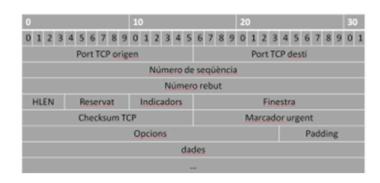
Checksum TCP

Opcions

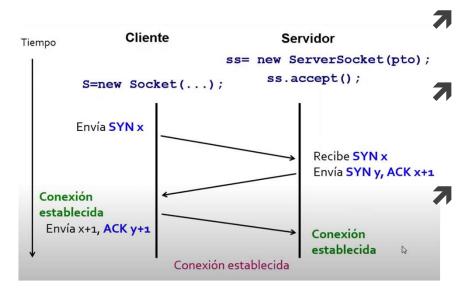
dades

- Finestra: 16 bits. Nombre de dades que l'emissor del segment està disposat a acceptar.
- CRC: 16 bits. Complement a 1 de 16 bits de la suma dels complements a 1 de totes les paraules de 16 bits de la capçalera i text. L'últim byte s'omple amb 0 a la dreta per formar una paraula de 16 bits complerta i poder així fe la suma de control. En aquest càlcul el propi camp de CRC es considera omplert per 0s.

- Marcador urgent: 16 bits. Consisteix en un desplaçament positiu des del número de seqüència fins a on tindrem les dades urgents. Només s'interpreta si el bit de URG està marcat a 1.
- Opcions: Variable, però sempre múltiple de 8 bits.



Establiment de la connexió a nivell TCP (Protocol de tres Fases)



El receptor espera la connexió executant les primitives LISTEN i ACCEPT

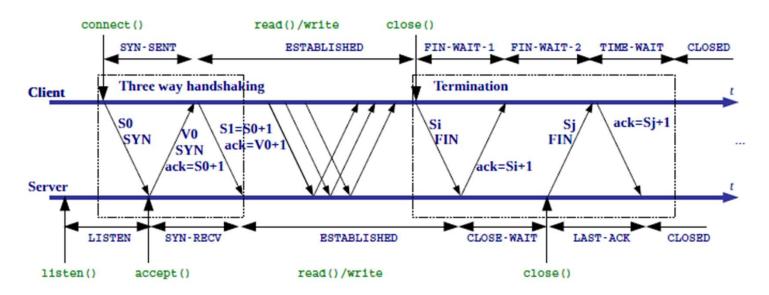
El host que desitja iniciar la connexió executa la primitiva CONNECT, especificant IP i port on es vol connectar

CONNECT fa una apertura activa, enviant l'altre host un paquet que té el bit de SYN activat i on s'indica el nombre de seqüència inicial 'x' que farà servir per enviar els missatges.

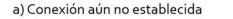
Primitives TCP

Primitives	Estats en TCP
LISTEN: Escolta el canal per detectar una connexió	CLOSED: no hi ha connexió activa ni pendent
ACCEPT: Accepta una connexió	LISTEN: El servidor espera trucada
CONNECT: Connexió amb client remot	SYN RCVD: Arriba sol·licitud de connexió. Espera ACK
READ: Lectura de dades en una connexió	SYN SENT: L'app comença a obrir una connexió
WRITE: Manda dades a través de una connexió	ESTABLISHED: Estat normal de transferència de dades
CLOSE: Termina la connexió	FIN WAIT 1: L'app diu que ja a terminat la connexió
BIND: Vincula una IP i un port de protocol a un socket	FIN WAIT 2: L'altre costat allibera la connexió
SHUTDOWN: Termina una connexió TCP	TIMED WAIT: Espera que tots els paquets terminin
	CLOSING: Tancant la connexió
	CLOSE WAIT: L'altre costat inicia l'alliberament de la connexió

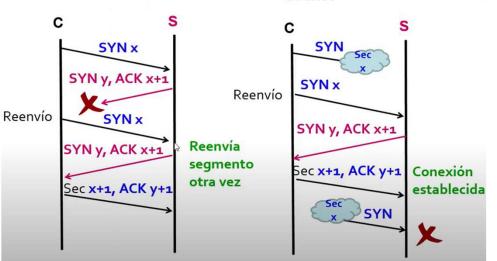
- Establiment de connexió i terminació
 - **7** El client sempre envia el primer segment.
 - El triple handshaking no té payload
 - SYN i FIN tenen número de seqüència
 - **7** El nombre de seq. Inicial és aleatori.



Duplicats de SYN



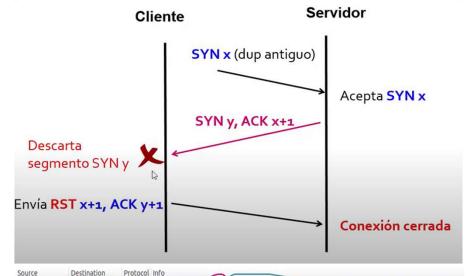
b) Conexión ya establecida y en curso



- RESET: aborta la conexión TCP
- Causas:
 - El puerto destino no está en uso (no ServerSocket)
 - Número de secuencia imposible



c) Conexión ya cerrada



192.168.0... 158.42.4.... TCP 64839 - 80 SYN Seq=3872861799 Win=8192 Len=0 MSS=1460 WS=4 SACK_F 158.42.4.... 192.168.0... TCP 80 - 64839 [SYN, ACK] Seq=3858251934 Ack=3872861800 Win=29200 Len=0 192.168.0... 158.42.4.... TCP 64839 - 80 [ACK] Seq=3872861800 Ack=3858251935 Win=66364 Len=0

Bits activados

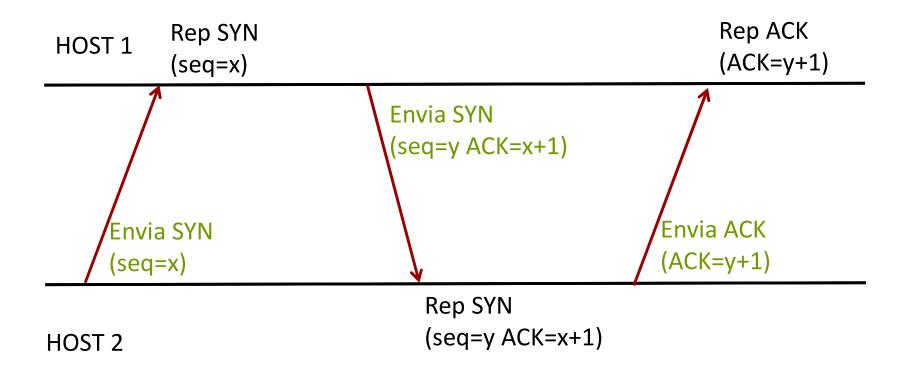
Números de secuencia iniciales (NSI)

- Cliente
 - Dirección IP: 192.168.o...
 - NSI: 3872861799
 - Puerto: 64839
- Servidor
 - Dirección IP 158.42.4...
 - NSI: 3858251934
 - Puerto: 8o

El host receptor rep el segment i revisa si hi ha algun procés actiu que hagi executat un LISTEN en el port sol·licitat. Si existeix, el procés que rep el missatge rep el segment TCP entrant, registra el número de seqüència 'x' i, si desitja obrir la connexió, respon amb un ACK 'x+1' I el bit de SYN activat. Inclou el seu número de seqüència inicial 'y' deixant oberta la connexió al seu extrem. El número d'espera 'x+1' significa que ha rebut fins al número 'x' l espera 'x+1'. Si no desitja establir la connexió envia una resposta amb el bit RST activat.

- Il primer host rep el segment i envia la confirmació. A partir d'aquest moment pot enviar dades des del seu extrem.
- El segon host rep la confirmació i entén que l'altre host ja ha obert la seva connexió

CONNEXIÓ ESTABLERTA!!



ESTABLIMENT CONNEXIÓ TCP (EXTREM A EXTREM)

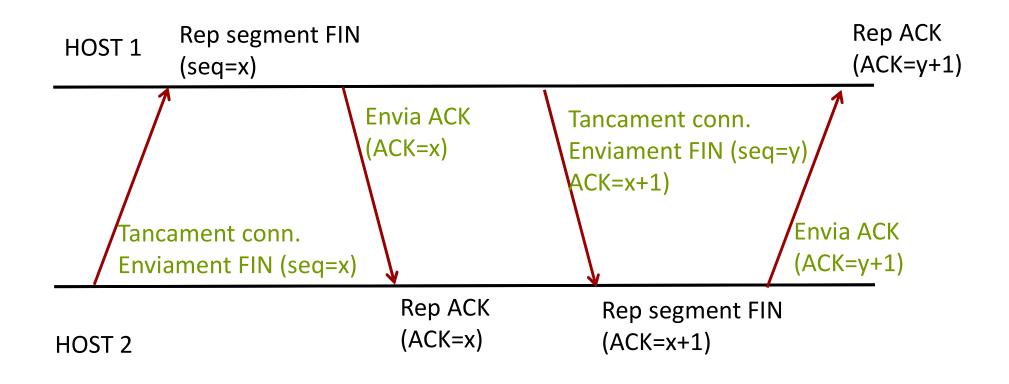
Tancament connexió TCP

- Tancament de la connexió
 - La màquina emissora, quan no té més dades per transmetre, envia un segment TCP amb el bit FIN activat I tanca el sentit d'enviament en la connexió, mantenint obert el sentit de recepció
 - El host receptor rep el segment amb el bit de FIN activat, retorna la confirmació I informa a l'aplicació receptora de la petició de tancament. És aquesta la que decideix el tancament de l'altre sentit de la connexió. Mentre, envia un segment ACK

Tancament de la connexió TCP

- Quan la màquina receptora decideix tancar la comunicació envia un segment TCP amb el bit de FIN activat
- La primera màquina rep aquest segment I envia el corresponent ACK(tot I que ha tancat el seu sentit de connexió segueix enviant confirmacions)
- Quan la màquina receptora rep el segment ACK tanca definitivament la connexió

Tancament de la connexió TCP



TANCAMENT CONNEXIÓ TCP (EXTREM A EXTREM)

Control d'errors en TCP

- Detecció d'errors:
 - → Bit de paritat: 1->num. '1s' senar, 0-> num '1s' parell
 - Checksum: Complement a 1 de la suma en mòdul 2¹⁶+1 del segment. Inclou també la capçalera IP
 - CRC: comprovació de redundància cíclica
- Confirmació positiva I expiració per interval de temps. L'emissor espera un temps I si no rep un ACK torna a enviar la trama
- Confirmació negativa I transmissió: Només per demanar reenviament

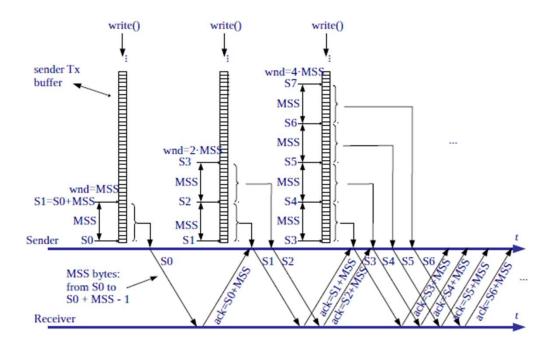
Control de Flux en TCP

- ARQ: Mètode de sol·licitud de repetició automàtica
- PAR: Recepció positiva amb retransmissió (ACK)
 - RQ-Inactiva o RQ-Continua
- Control de transmissió -> Cada datagrama es numera per al seu control
 - Datagrama _0 de 900 bytes
 - → Datagrama _900 de 400 bytes
 - → Datagrama _1300 de 500 bytes

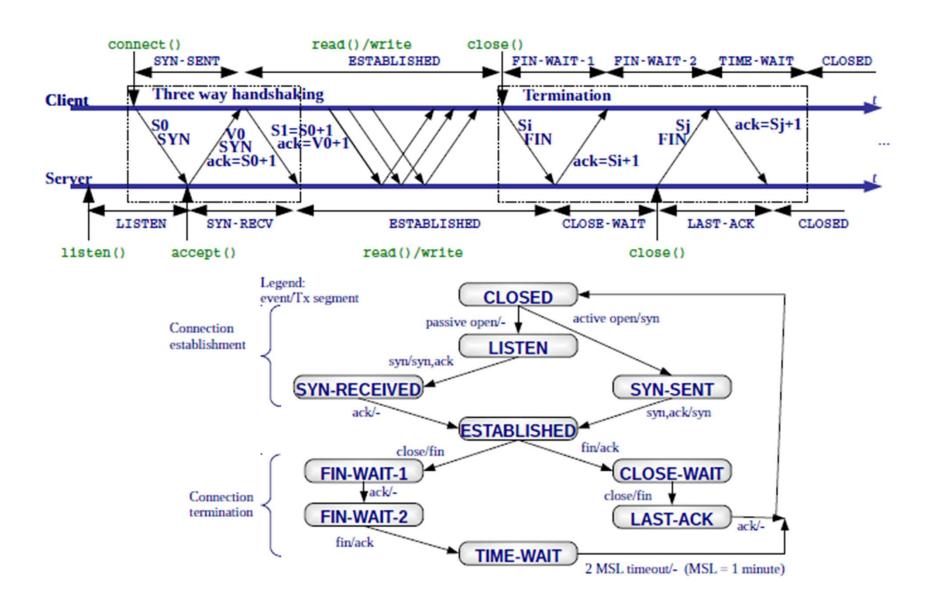
Control de Flux en TCP

- In TCP els nombres de seqüència i les Finestres es mesuren en Bytes, indentificant el primer byte de dades del segment.
- Si un segment té un nombre de seqüència S_i i porta MSS bytes, el segon nombre de seqüència será S_i +MSS
- La confirmació del segon segment amb nombre de seqüència S_i i MSS bytes de dades tindrà el valor ACK amb seqüència S_i + MSS

- El número de seqüència identifica el primer byte del payload
- El número del ACK identifica el següent byte que espera el receptor.



TCP – Diagrama de estats



Control de Flux TCP

- The TCP header fields that are relevant to managing the reliability of the transmission are:
 - Sequence Number: Tracks the byte stream's sequence number to ensure data is received in the correct order.
 - Acknowledgment Number: Indicates the next sequence number that the sender of the segment is expecting to receive.
 - Window Size: Specifies the size of the receive window, which helps manage flow control and congestion avoidance. Window Size changes **DINAMICALLY**
- These fields contribute to managing reliability and flow control in TCP transmissions.

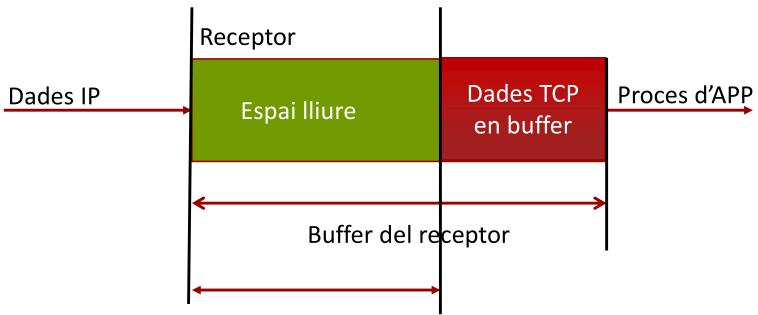
Control de Flux

- The maximum window size in TCP is 65.535 bytes. This value is determined by the 16-bit field used to represent the window size in the TCP header
- In a TCP flow control scenario, if a sender has a large amount of data to transmit but the receiver's buffer is nearing capacity, the receiver can advertise a smaller window size in the TCP header. This informs the sender to slow down and send data at a rate that the receiver can handle without overflowing its buffer.

For example, if the receiver advertises a window size of 1000 bytes, the sender will only send up to 1000 bytes of data before waiting for acknowledgment and further window updates from the receiver. This dynamic adjustment of the window size helps regulate the flow of data and prevents overwhelming the receiver.

Control de Flux – Buffer de Recepció

Cada extrem receptor TCP disposa de un buffer de recepció



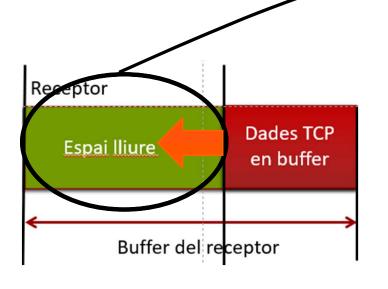
Mida de la finestra, que dependrà de l'espai lliure al buffer

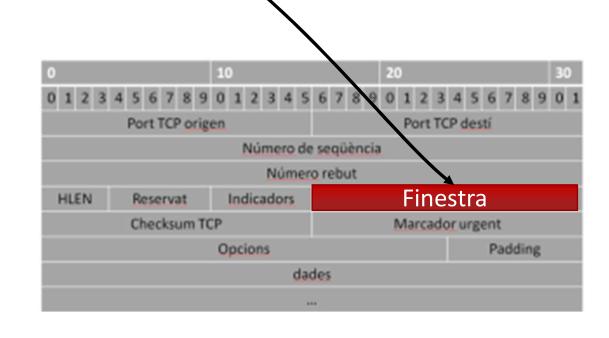
Sense un sistema de control de flux es pot producir un desbordament de buffer

Control de Flux – Finestra de recepció

La finestra de recepció anirà variant en funció de la mida de l'espai lliure que hi ha al buffer.

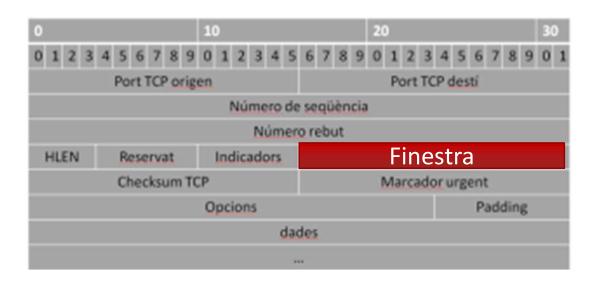
És dinàmica i pot prendre el valor 0





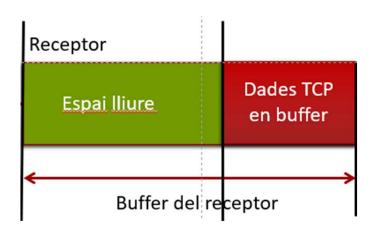
Control de Flux – Finestra de recepció

- Si la finestra de recepción es queda sense espai, posa el valor de la finestra a 0.
- L'emisor enviarà segments amb 1 Byte de dades, esperant que la resposta del receptor



Control de Flux – La finestra de Tx

Fixe'm-nos que la finestra de l'emisor ha de guardar aquelles dades que encara no han estat reconegudes



Espai enviat
NO reconegut

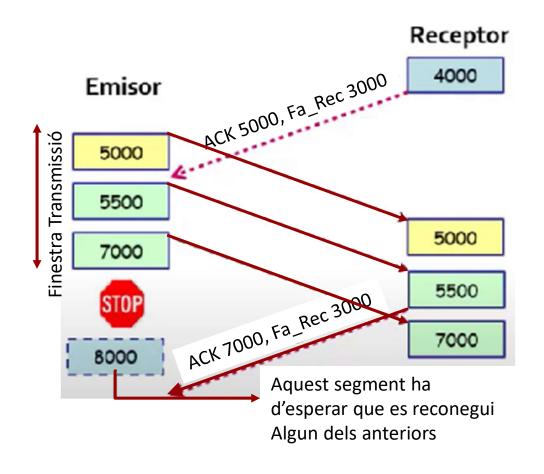
Buffer de l'emisor

Finestra de transmissió

Per altra banda, la mida de la finestra no pot ser superior a la finestra del receptor

Control de Flux – Finestra de Tx

- Donat que la finestra de Tx no POT SER > que la de Rx
 - No s'envia cap dada que no "càpiga" al receptor
 - No es poden acceptar noves dades del nivell superior si s'arriba al limit de la finestra



UDP: Definicions I conceptes

- El protocol de datagrama d'usuari (UDP) està definit en la RFC 768.
- No garanteix l'entrega de paquets IP. Són les pròpies aplicacions les que han de proporcionar mecanismes de fiabilitat
- Es fa servir per transportar dades de forma ràpida, compacta l no fiable.
 - Intercanvi de dades i missatges curts: DHCP, DNS, RIP
- La velocitat d'entrega esta per sobra de la seguretat (multimèdia en temps real)
 - ▼ Voice over IP, Videoconferències, streamming de video/audio

UDP

- Protocol no confiable
- No orientat a connexió
- Premia la velocitat en l'entrega
- No existeix control de errors amb reenviament
- S'especifica amb el número 17 en el camp de capçalera IP
- Una única cua de missatges
- Important per apps que necessiten multimèdia en temps real

UDP

- UDP és útil en aplicacions on la perdua de paquets no és un factor crític (telefonia, videoconferència, ...)
- No tolera retards. Si un paquet arriba amb més tard de l'instant en que hauria d'arribar es descarta.

Aquest darrer punt genera disrupcions en el so o la

imatge

Aplicaciones	Puertos		
TFTP	69		
SNMP	161		
DHCP-BOOTP	67/68		
DNS	53		
RTP	Puertos pares dinámicos que empiezan en 1234		

UDP

BITS 0 - 15	BITS 16 - 31		
Port origen	Port destí		
Longitud del missatge	Suma de verificació		
d a d e s			

Ports

- Tant TCP com UDP fan servir ports per permetre la comunicació entre aplicacions. Aquest camp té una mida de 16 bits, per tant el rang de valors va des de 0 fins a 65535. El port 0 està reservat però es un valor permès com a port origen si el procés emissor no espera rebre missatges com a resposta.
- Els ports 1 a 1023 són els ports 'ben coneguts'. En sistemes operatius tipus Linux enllaçar amb un d'aquests ports requereix accés com a *superusuari*.

Ports

IIDD

- Els ports 1024 a 49151 son ports registrats però no regulats
- Els ports 49.152 a 65.535 són ports efímers I es fan servir com ports temporals.

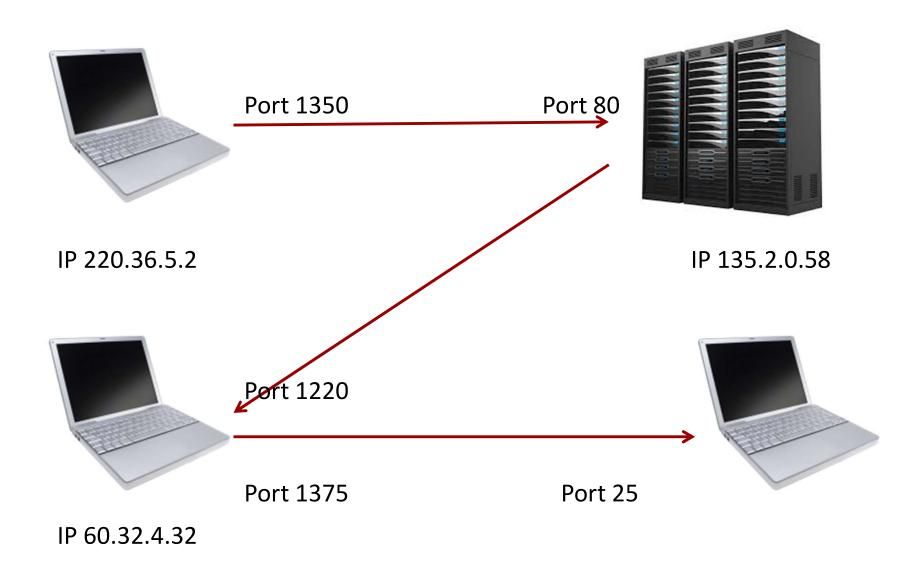
Exemples de ports associats:

	ICF			UDF	
Echo	→	7	DNS	→	53
FTP	→	20 (dades) 21(ccontrol)	BOOTP (server)	→	67
SSH, scp, SFTP	→	22	BOOTP (client)	→	68
Telnet	→	23	TFTP	→	69
SMTP	→	25	SMTP	→	25
Time Protocol	→	37	Time Protocol	→	37
HTTP	→	80	SNMP	→	161

SOCKETS

- Sockets: Punts terminals de connexió establerts entre dues màquines
 - **尽** Socket = adreça IP + port de protocol
- Primitives associades als sockets
 - SOCKET: crea un nou punt terminal de connexió
 - → BIND: connecta una adreça local a un socket
 - LISTEN: anuncia la disposició a acceptar connexions. Indica també la mida de la cua
 - ACCEPT: bloqueja al invocador fins l'arribada d'un intent de connexió
 - CONNECT: intenta establir activament la connexió
 - SEND: Envia dades a través de la connexió
 - RECEIVE: Rep dades de la connexió
 - CLOSE: Tanca la connexió

Connexions amb sockets



Sockets

- Els números inferiors a 255 es fan servir per aplicacions públiques
- Els números compresos entre 255 I 1023 s'assignen a empreses per a aplicacions comercials
- Els números superiors a 1023 no estan regulats I normalment s'utilitzen per a assignació dinàmica

Ports I sockets

La comanda netstat proporciona connexions actives de la màquina I els sockets en us d'aquesta

Linux# netstat -nt

Active Internet connections

- Is ports son punts d'entrada de paquets de dades a una màquina → Punts perillosos per la seva vulnerabilitat
- Només s'han de tenir oberts aquells ports que són necessaris