



TEMA 4: CAPA DE TRANSPORT

UF1: INTRODUCCIÓ A LES XARXES
M07: PLANIFICACIÓ I ADMINISTRACIÓ DE XARXES

Administració de Sistemes Informàtics en Xarxa

LA CAPA DE TRANSPORT

- ▶ Rep les dades de la capa d'aplicació
 - ◉ Les segmenta (creant segments, la PDU de la capa)
 - ◉ Les encapsula (afegint-hi una capçalera de la capa)
 - ◉ Les envia a la capa d'Internet
- ▶ Rep les dades de la capa de transport
 - ◉ Les desencapsula (llegeix la capçalera de la capa)
 - ◉ Recobla els diferents segments (segons la informació de la capa)
 - ◉ Les entrega a l'aplicació corresponent (segons el port)

FUNCIONS DE LA CAPA

- ▶ Permetre múltiples connexions
 - ◉ De diferents aplicacions
 - ◉ En un mateix dispositiu
 - ◉ Gràcies a la segmentació
- ▶ Assegura que les dades arribin correctament
 - ◉ De forma ordenada
 - ◉ De forma confiable
- ▶ Gestiona els possibles errors

► Control bàsic

- ◉ Múltiples converses al mateix temps
 - ✦ Segmentació
 - ✦ Multiplexació
- ◉ Identificació de les aplicacions amb els ports

► Control addicional

- ◉ Converses orientades a la connexió i entrega fiable
 - ✦ Utilitza missatges de control per assegurar que els missatges han arribat
- ◉ Entrega en el mateix ordre
- ◉ Control del flux
 - ✦ Demanar disminució de la velocitat perquè no hi hagi tanta pèrdua

► Com aconseguir-la?

- ◉ Seguiment de les dades (en ordre)
 - ♦ Amb un número de seqüència
- ◉ Avís de rebut
- ◉ Retransmissió de segments perduts

► Genera un augment de trànsit considerable

- ◉ Hi ha més dades a la capçalera del segment
- ◉ Depenent el protocol escollit, decidim si l'incloem o no

PROTOCOLS MÉS HABITUALS

► UDP

- ◉ User Datagram Protocol
- ◉ Senzill i no orientat a connexió
- ◉ No és fiable
- ◉ Exemples: DNS, streaming de vídeo, VoIP, etc.

► TCP

- ◉ Transport Control Protocol
- ◉ Orientat a connexió
- ◉ Garanteix que les dades arriben en ordre, fiables i amb control de flux (la capçalera és més gran)
- ◉ Exemples: web, mail, transferència de fitxers, etc.

PORTS (I)

- ▶ La capa de transport identifica les aplicacions
 - ◉ Amb un número únic a cada dispositiu
- ▶ Si el procés és un servei...
 - ◉ Tenen ports estàtics assignats
 - ◉ Generalment, associats al protocol
 - ✦ HTTP = 80
 - ✦ FTP = 21
- ▶ Si el procés és un client...
 - ◉ S'assigna de forma aleatòria
 - ◉ El sol·licita a la capa de transport, que en té un control

PORTS (II)

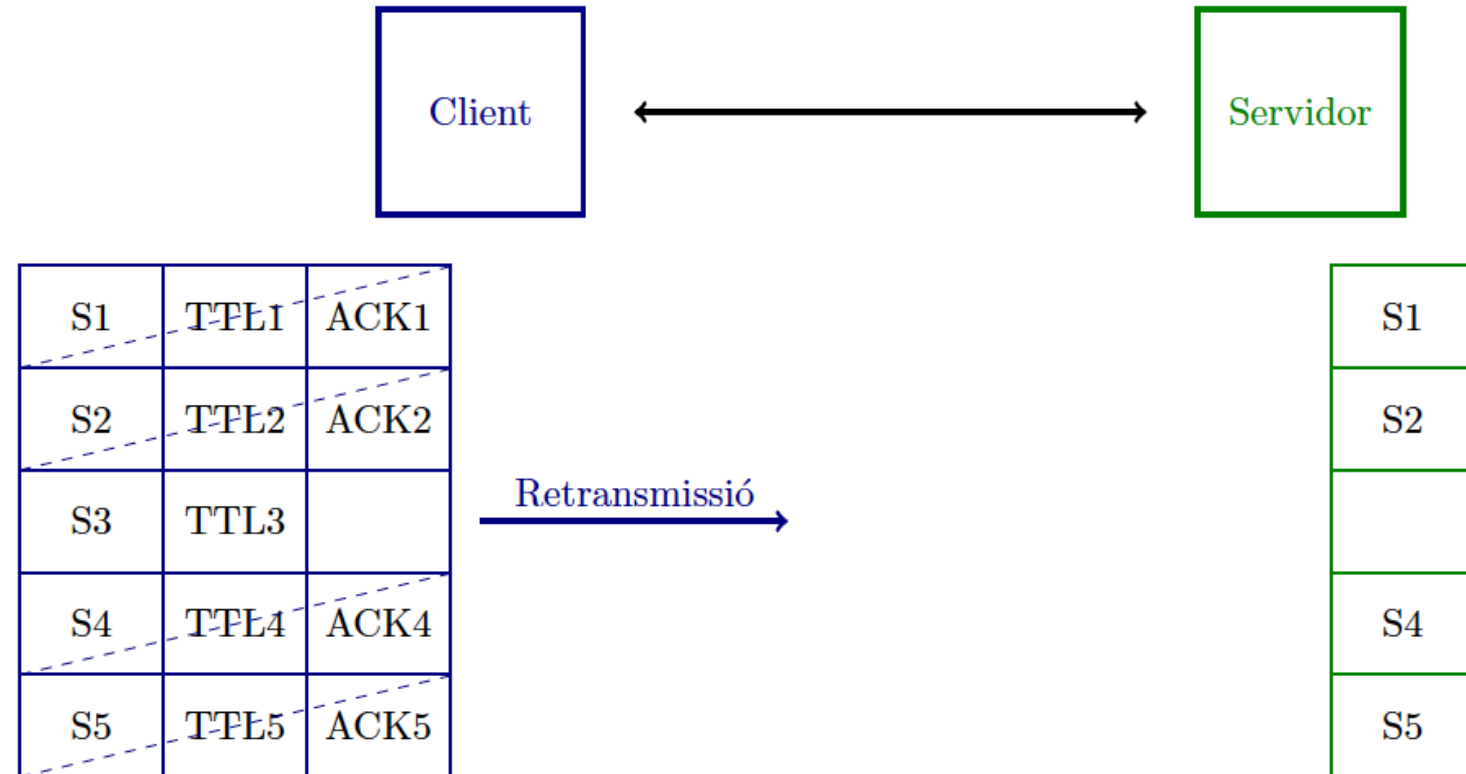
- ▶ Un *socket* o sòcol és la combinació d'IP i port
 - ◉ Identifica de manera unívoca una aplicació en un dispositiu
 - ◉ Socket = IP : Port
- ▶ IANA és l'organisme que regula l'assignació de ports:
 - ◉ Ports Coneguts (0-1023)
 - ✦ De sistema i estàndard (serveis o aplicacions de sistema)
 - ◉ Ports Registrats (1024-49152)
 - ✦ Reservats a aplicacions de client
 - ◉ Ports Dinàmics o Privats (49152-65535)
 - ✦ No reservats
- ▶ La capa de transport s'encarrega de no assignar un port a dues aplicacions diferents

- ▶ Si no segmentéssim...
 - ◉ Ocuparíem molta més estona el medi
 - ◉ Qualsevol error implicaria tornar a enviar des del principi
 - ◉ Els dispositius no tindrien *buffers* tan grans per emmagatzemar les dades mentre s'envien
- ▶ Els dos protocols segmenten de forma diferent
 - ◉ TCP etiqueta cada segment de forma que es puguin reacoblar segons el número de seqüència
 - ◉ UDP no indica número de segment: es reacoblen segons l'ordre d'arribada dels datagrames
 - ✦ A UDP la PDU s'anomena "datagrama"

CONVERSES FIABLES A TCP

- El dispositiu origen té una llista amb tots els segments enviats

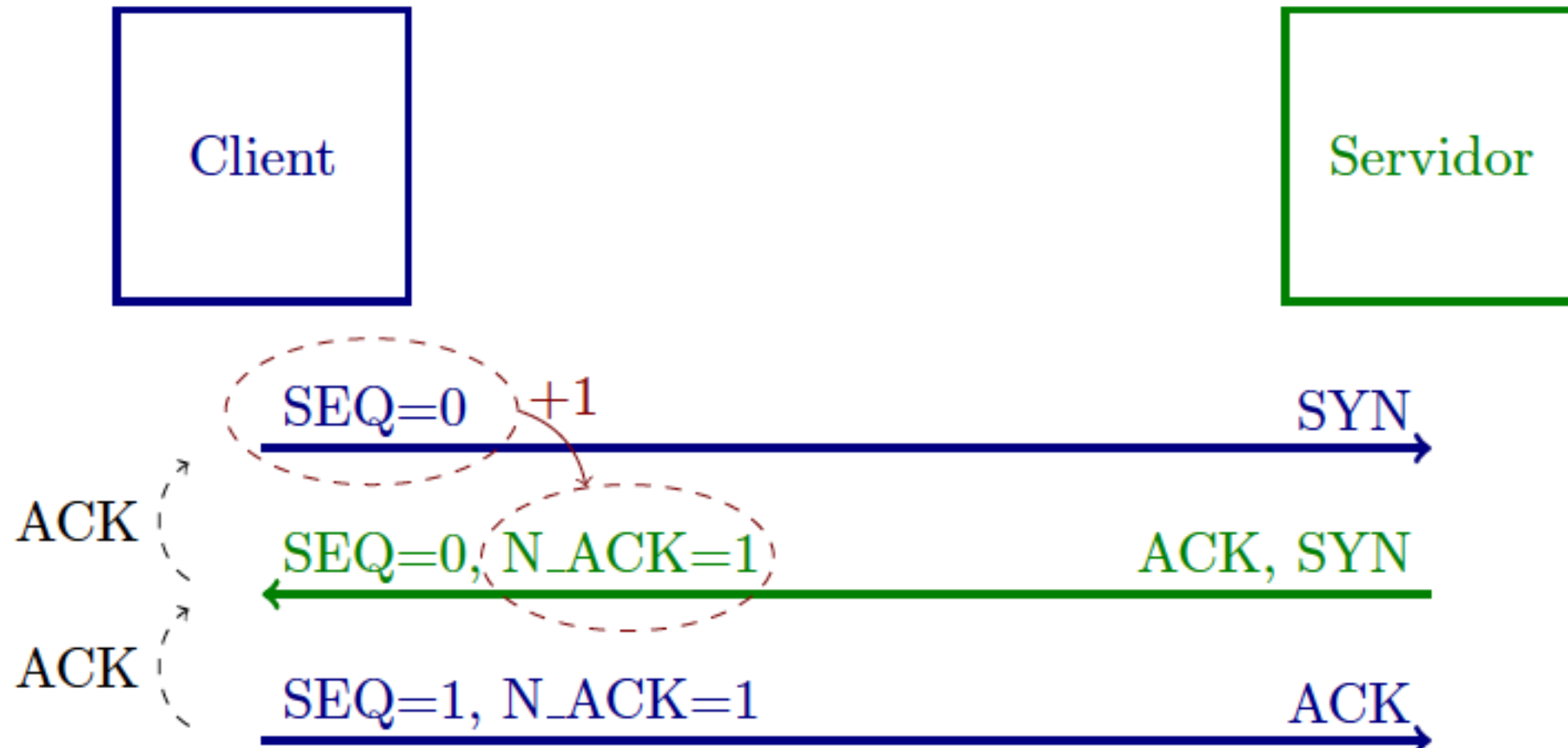
- ◉ S'anota quan en rep confirmació
- ◉ Passat un temps, si no ha rebut confirmació, el torna a enviar



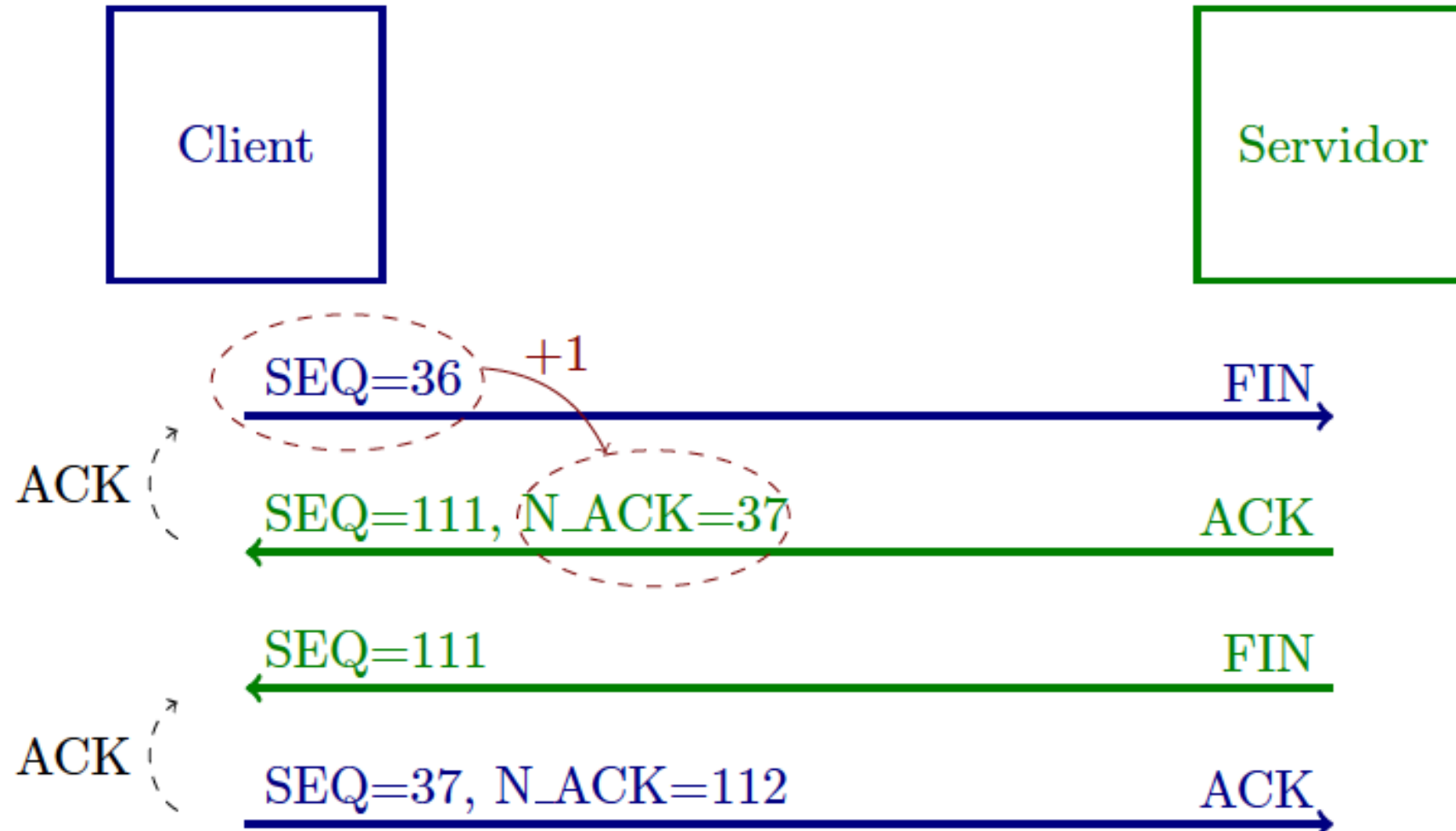
FLAGS DEL TCP

- ▶ URG: indica que el Segment es urgent (*urgent*)
- ▶ ACK: indica que el Segment es un avís de rebut (*acknowledgment*)
- ▶ PSH: demana l'enviament del Segment (*push*)
- ▶ RST: indica que el Segment demana reiniciar la sessió (*reset*)
- ▶ SYN: indica que el Segment demana fer una sincronització del numero de seqüència entre els dispositius origen i destí (*synchronization*)
- ▶ FIN: indica que el Segment demana finalitzar la sessió

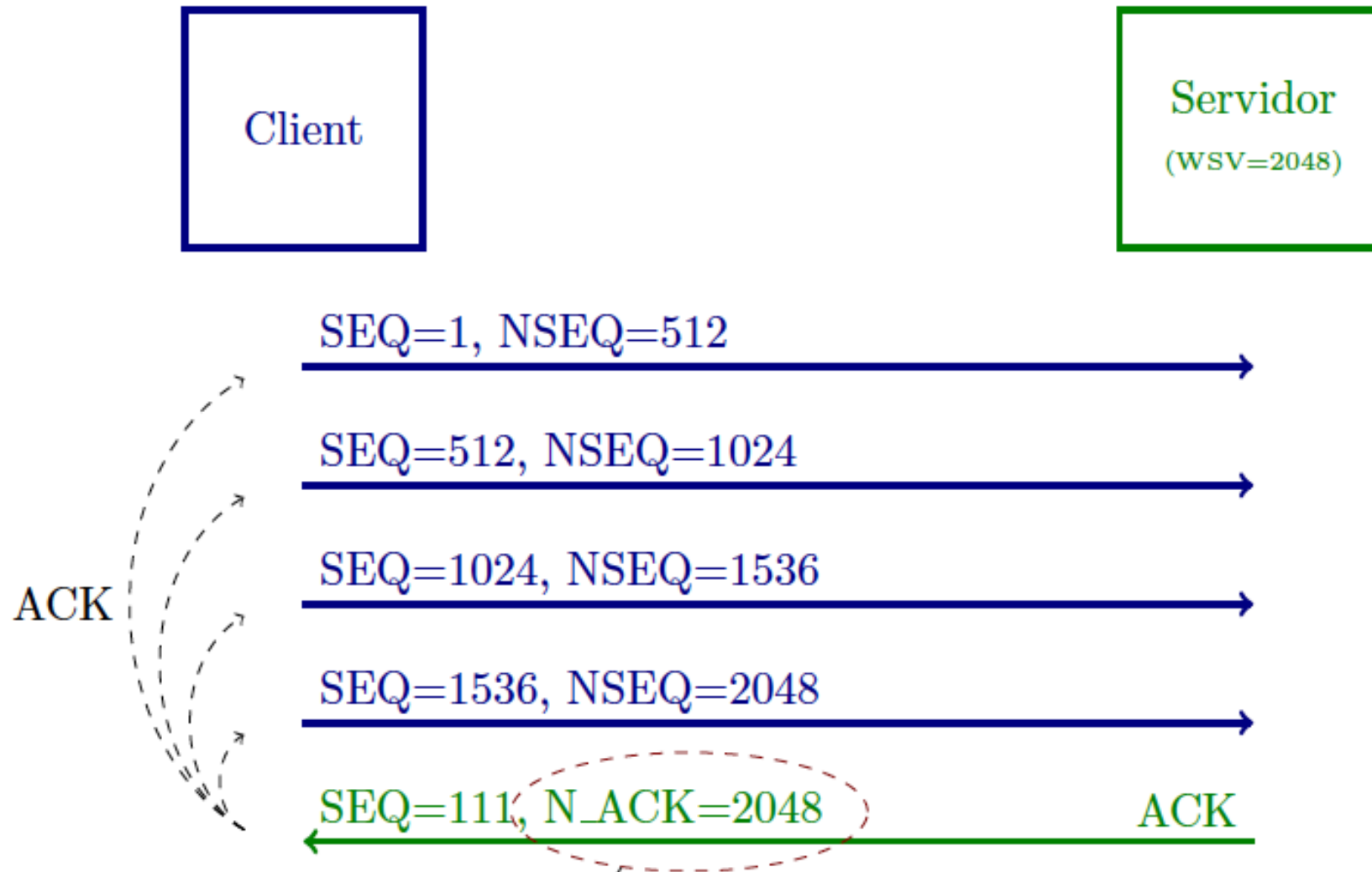
INICI DE SESSIÓ TCP



FINALITZACIÓ DE SESSIÓ TCP



AVÍS DE REBUT TCP



WINDOW SIZE VALUE

- ▶ El WSV ens permet saber quantes dades pot rebre un dispositiu sense col·lapsar-se
 - ◉ Forma part de la capçalera TCP
 - ◉ L'emissor pot enviar, sense haver-ne rebut confirmació, tantes dades com el receptor li hagi indicat
- ▶ El WSV permet el control de flux
 - ◉ Adaptem la velocitat d'enviament a la capacitat dels dispositius o del medi
 - ◉ Fins que no s'alliberi el buffer del receptor, l'emissor no envia més segments