

Universidad de Los Andes
Facultad de Ingeniería
Escuela de Sistemas

Capa de Transporte

Mérida - Venezuela
Prof. Gilberto Díaz

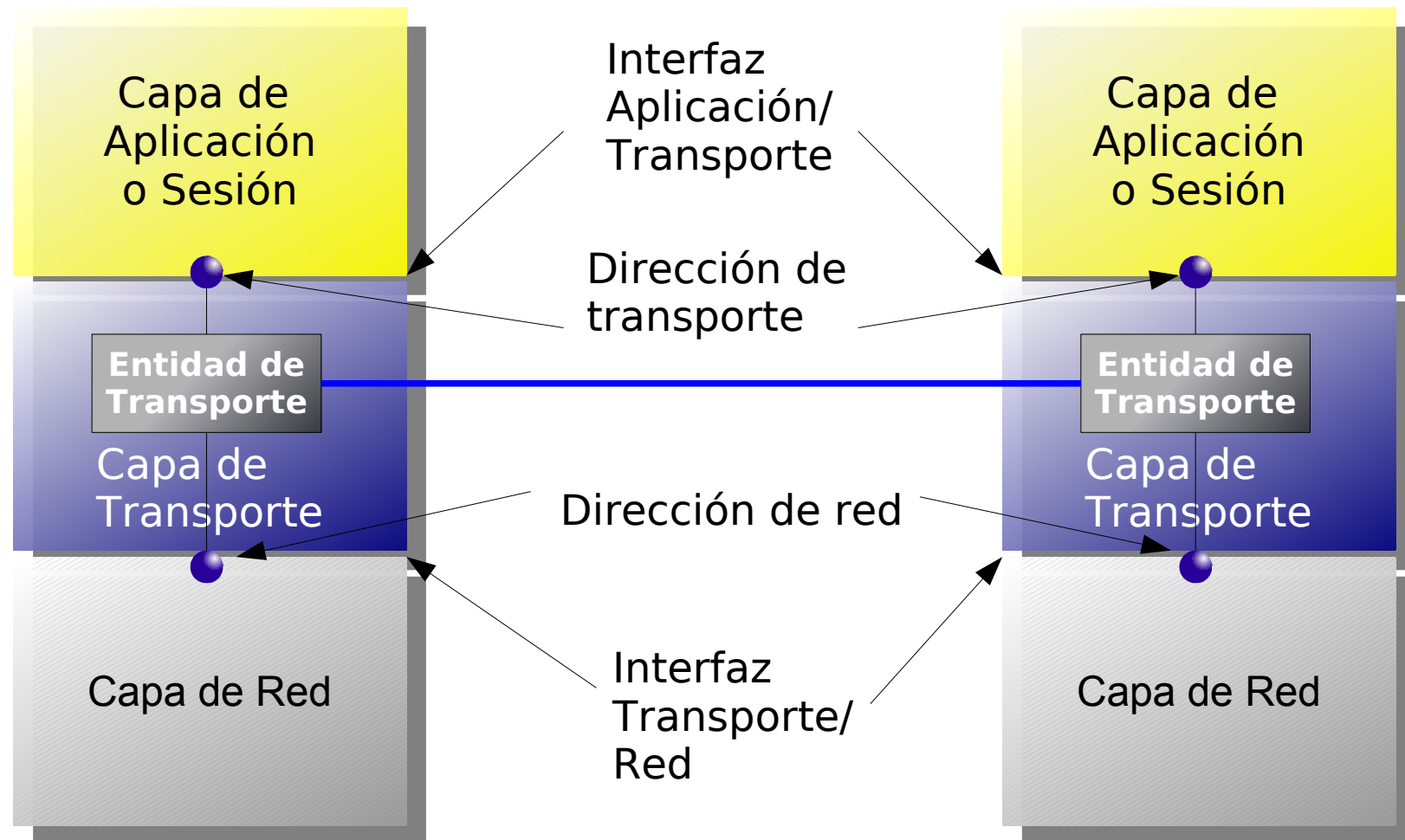
Capa de Transporte

Transmission Control Protocol (TCP)

- Fue diseñado para proporcionar una transferencia confiable sobre una capa de red no confiable
- Hay muchos factores que difieren en un contexto de múltiples redes
 - Topología
 - Demoras
 - Ancho de banda

Capa de Transporte

Transmission Control Protocol (TCP)



Capa de Transporte

Transmission Control Protocol (TCP)

- Cada nodo que soporta TCP tiene una entidad de transporte de TCP, y uno de los siguientes componentes:
 - biblioteca de funciones,
 - proceso de usuario o
 - una porción del kernel.

Capa de Transporte

Transmission Control Protocol (TCP)

- En cualquier caso se debe gestionar
 - Flujos (*streams*)
 - La interfaz de la capa de red

Capa de Transporte

Transmission Control Protocol (TCP)

- En general la entidad TCP recibe un flujo de los procesos locales, los fracciona en trozos que no exceden los 64KB
- En la práctica se utilizan tamaños de 1460 bytes para que quepan en un frame de ethernet y envía cada pieza como un datagrama
- Cuando un datagrama que contiene información TCP arriba al nodo destino, le son pasados a la entidad TCP quien reconstruye el flujo

Capa de Transporte

Modelo de servicio TCP (sockets)

- El servicio TCP se presta en los dos extremos de la comunicación (emisor y receptor) a través de *sockets*
- Estos sockets están compuesto de la dirección IP y un número de 16 bits llamado puerto
- Antes de transmitir un flujo se debe establecer explícitamente una conexión

Capa de Transporte

Modelo de servicio TCP (sockets)

- Un socket puede ser utilizado como punto final de múltiples conexiones al mismo tiempo.
- Las conexiones son identificadas utilizando los sockets de ambos extremos (*socket1*, *socket2*).
- Los puertos menores a 1024 se denominan puertos bien conocidos y son reservados para servicios estándares

Capa de Transporte

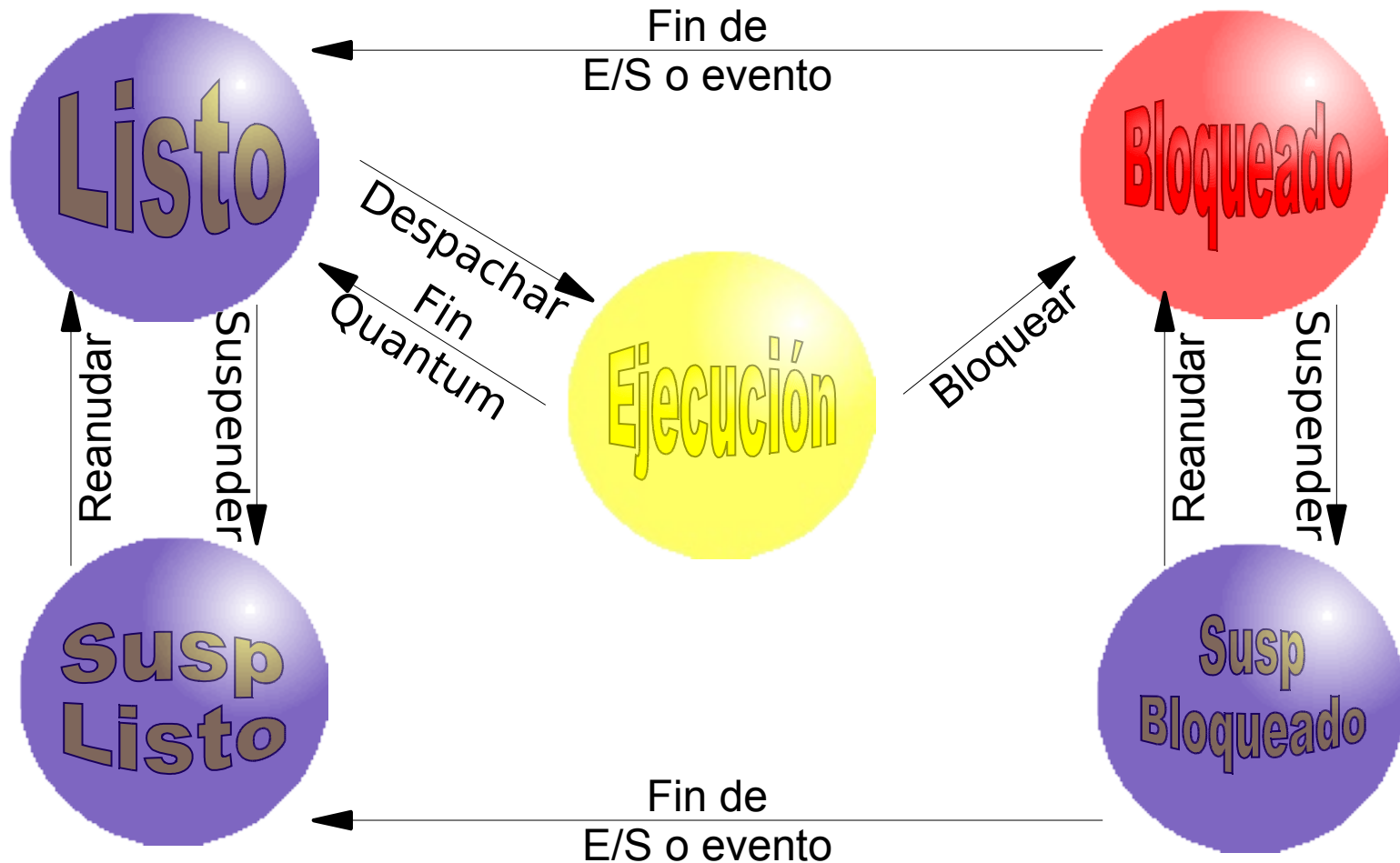
Modelo de servicio TCP (sockets)

- Ejemplo de puertos bien conocidos:

21	FTP
22	SSH
23	Telnet
25	SMTP
80	HTTP
110	POP

Capa de Transporte

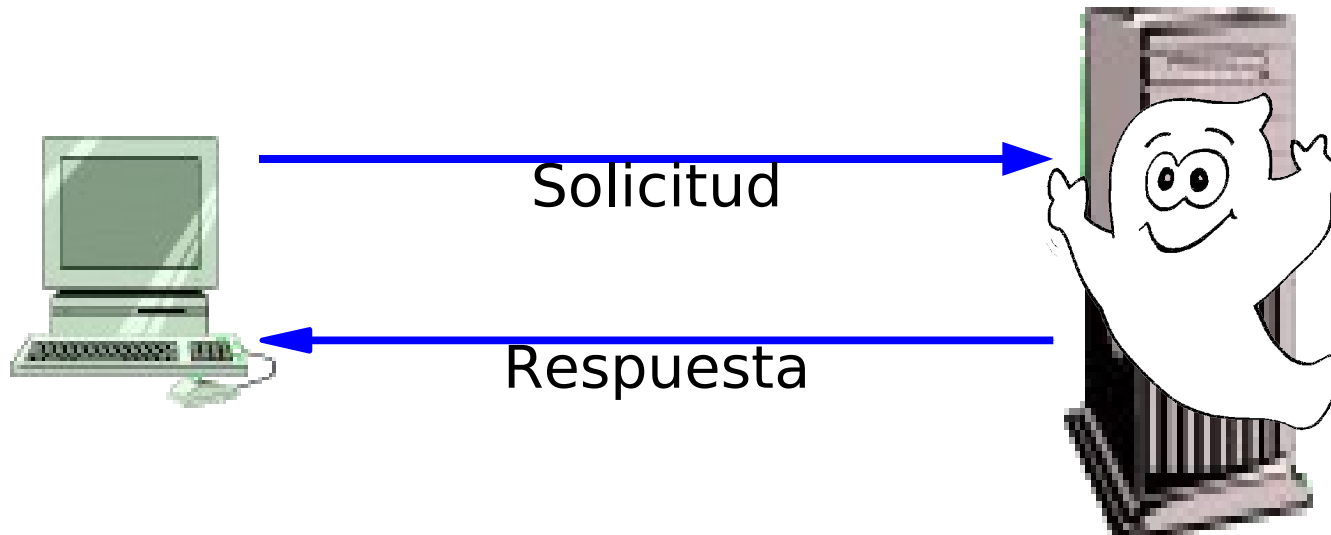
Procesos



Capa de Transporte

Daemons

Procesos que esperan solicitudes, despiertan, atienden la solicitud, envían la respuesta y vuelven a dormir



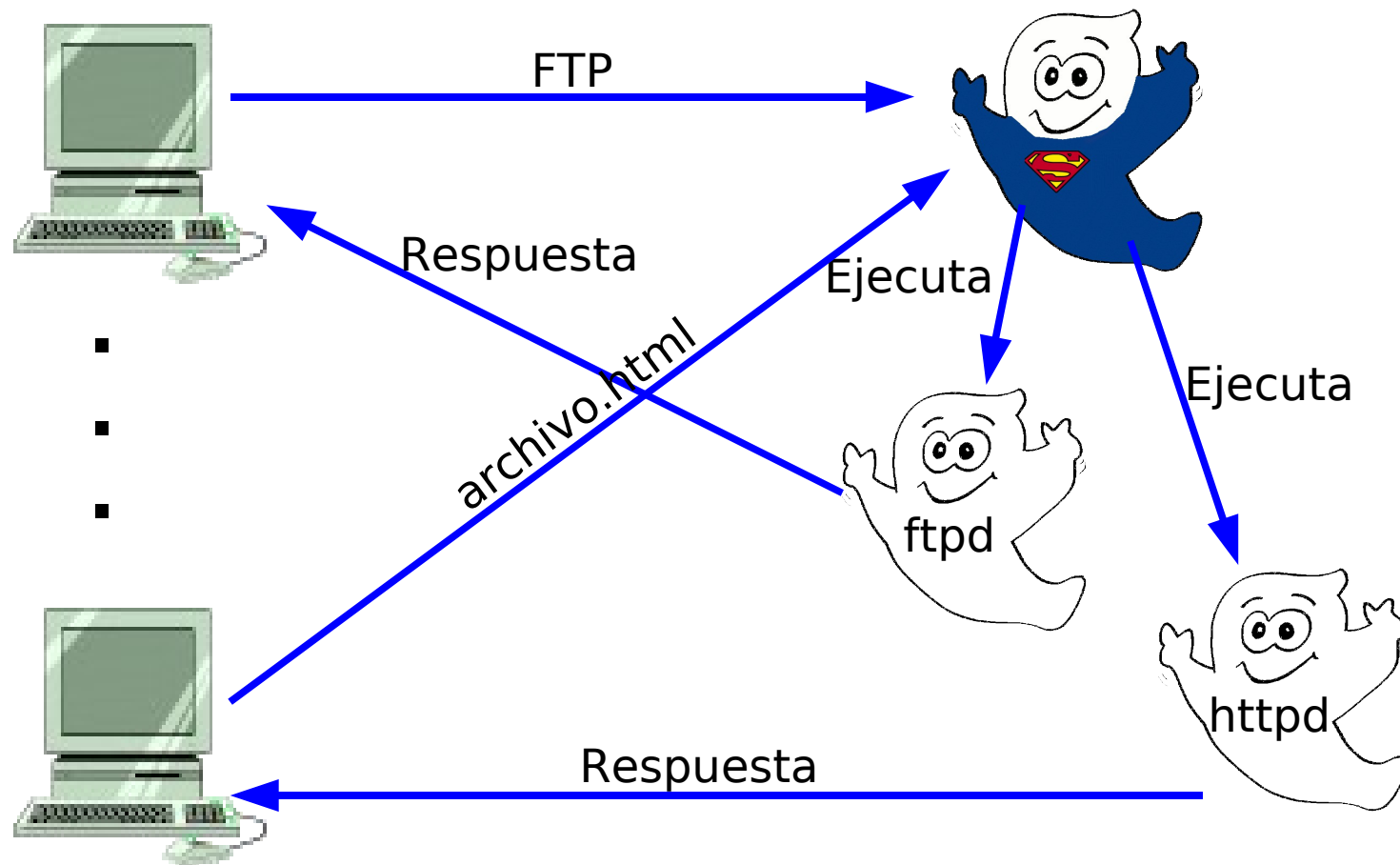
Capa de Transporte

Modelo de servicio TCP (sockets)

- En una máquina que preste muchos servicios podríamos tener muchos demonios atendiendo varios puertos, pero esto nos llenaría la memoria de programas ociosos la mayor parte del tiempo
- Por eso se creó un super demonio llamado *internet daemon* (inetd) quien recibe toda tipo de solicitud, ejecuta el demonio correspondiente para atenderla y luego finaliza. El inetd sigue escuchando otras solicitudes

Capa de Transporte

Modelo de servicio TCP (sockets)



Capa de Transporte

Modelo de servicio TCP (sockets)

- El archivo de configuración del demonio inetd:

`/etc/inetd.conf`

- El formato de las líneas es:

Nombre	tipo	protocolo	wait	status	uid	demonio	args
	dgram		wait				
	stream		nowait				

Capa de Transporte

Modelo de servicio TCP (sockets)

- Todas las conexiones TCP son full-duplex, es decir, la transmisión de datos se puede hacer en ambos sentidos al mismo tiempo
- Todas las conexiones TCP tienen sólo dos puntos: origen y destino. No soportan comunicación multicast o broadcast.

Capa de Transporte

Modelo de servicio TCP (sockets)

- Cuando un proceso le pasa a TCP datos para transmitir este lo puede hacer de inmediato o almacenarlos en un buffer en espera de más datos y tener suficiente para enviar.
- Sin embargo, a veces un proceso puede desear que los datos sean transmitidos inmediatamente. Para esto las aplicaciones cuentan con una bandera (PUSH)

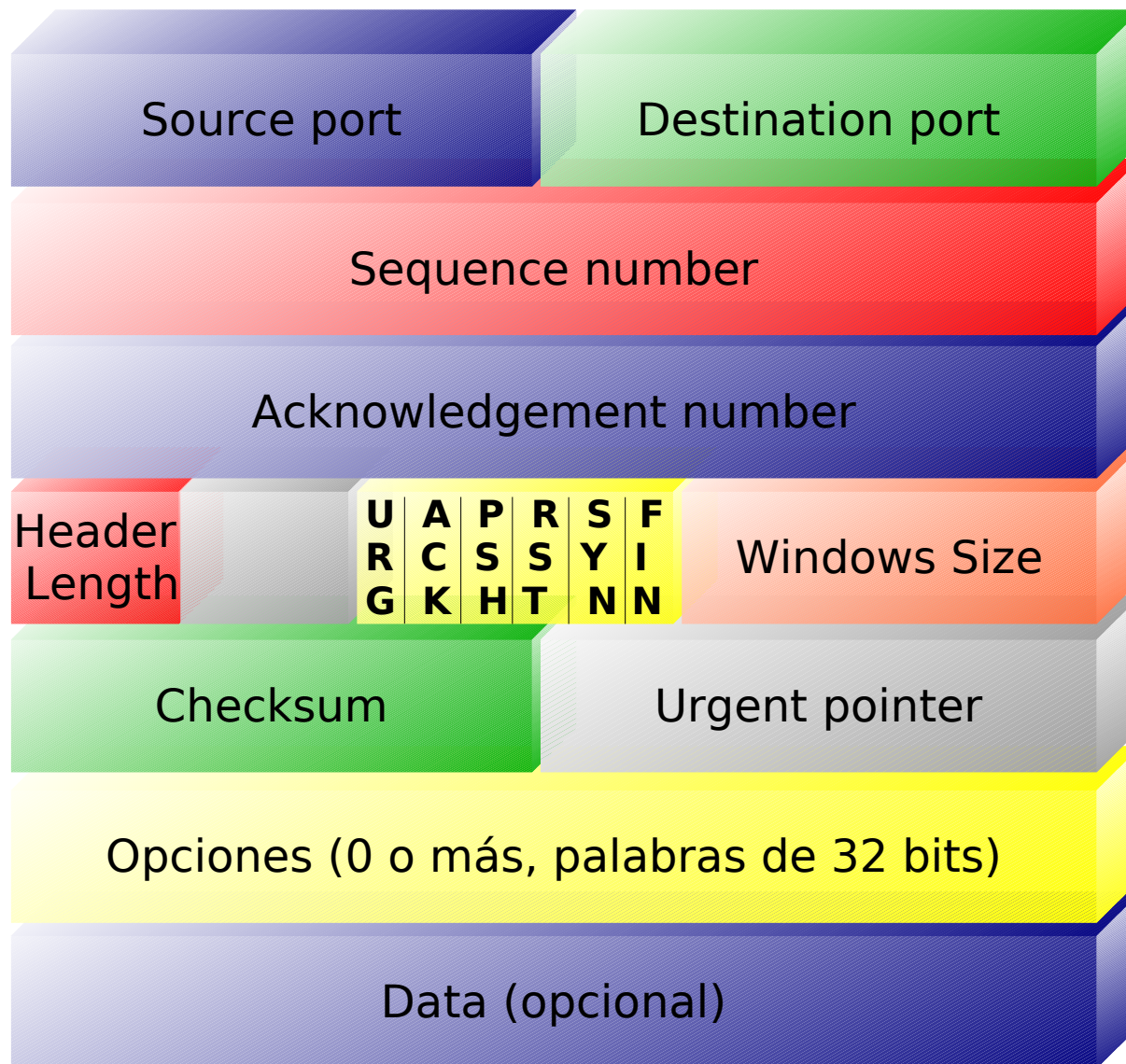
Capa de Transporte

El protocolo TCP

- Los PDU de TCP son denominados segmentos
- Cada segmento consta de un encabezado de 20 bytes
- TCP decide cuan grande es el contenido de los segmentos
- Sólo existen dos restricciones
 - Cada segmento de caber en el *payload* de un datagrama IP
 - Cada red tiene un MTU generalmente 1500 Bytes

Capa de Transporte

El protocolo TCP



Capa de Transporte

El protocolo TCP

- Si ACK es 1 significa que el *Acknowledgement number* es válido, de lo contrario éste último no se toma en cuenta.
- *SYN* se utiliza para establecer la conexión, tanto para solicitarla (*CONNECTION REQUEST*) como para aceptarla (*CONNECTION ACCEPTED*)
 - Solicitar conexión $SYN = 1, ACK = 0$
 - Aceptar conexión $SYN = 1, ACK = 1$

Capa de Transporte

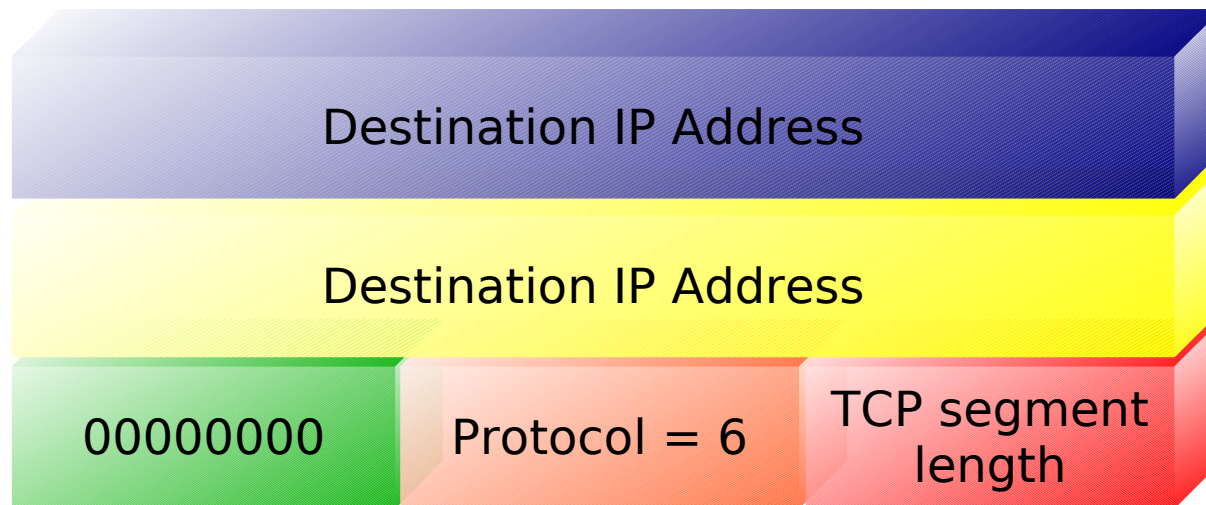
El protocolo TCP

- *Window size* indica cuantos bytes pueden ser enviados luego del segmento reconocido
- *Options* se utiliza para hacer indicaciones adicionales que el protocolo en si no maneja. Por ejemplo: el tamaño de los segmentos que el receptor puede manejar y la retransmisión por segmentos y no por flujos.

Capa de Transporte

El protocolo TCP

- El pseudo encabezado se incluye en el *checksum*



Capa de Transporte

El protocolo TCP

- El protocolo básico que utiliza TCP es el de ventana deslizante.
- Cuando el emisor envía un segmento también activa un contador
- La entidad TCP del receptor envía otro segmento de vuelta (*acknowledgement*) con el próximo número de secuencia que el espera recibir.

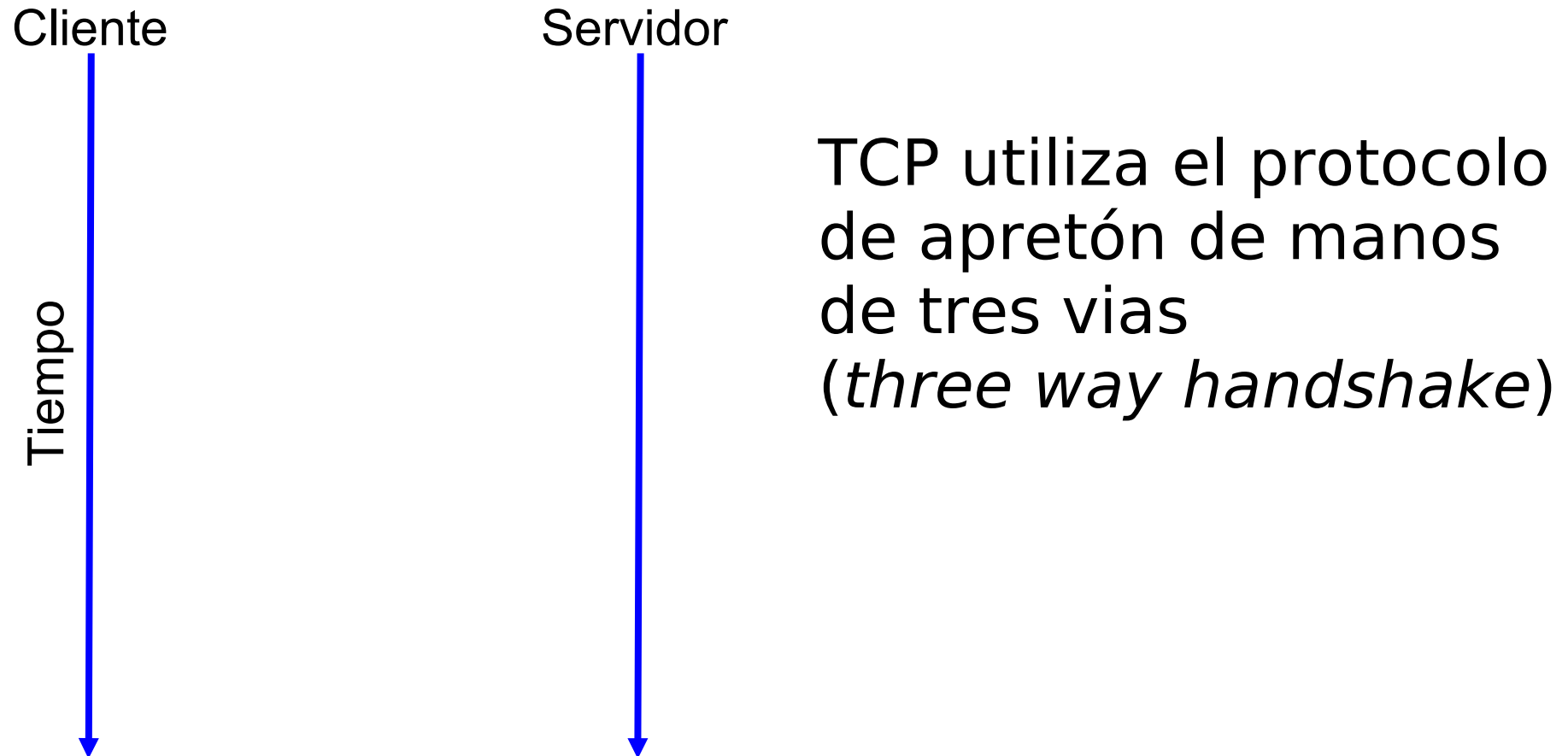
Capa de Transporte

El protocolo TCP

- Si el contador del emisor llega al final (*timeout*) antes del que llegue el reconocimiento, reenvía el segmento.
- Aunque este protocolo puede parecer simple, hay una serie de situaciones que deben ser manejadas
 - Los segmentos pueden llegar desordenados
 - Los segmentos pueden ser demorados

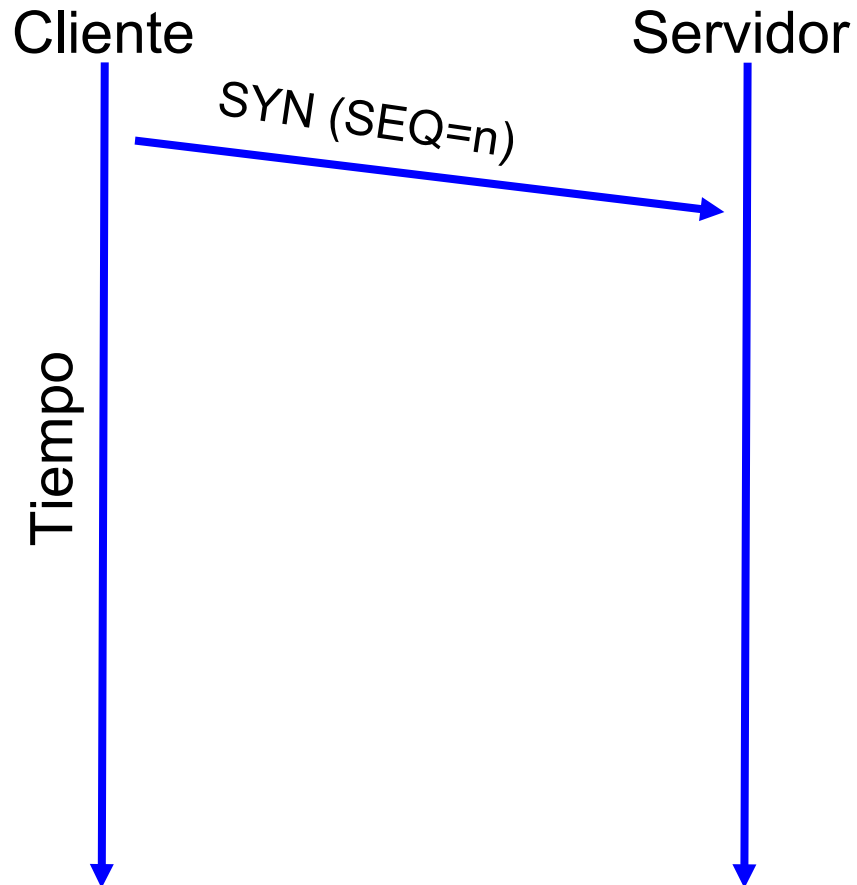
Capa de Transporte

El protocolo TCP: estableciendo la conexión



Capa de Transporte

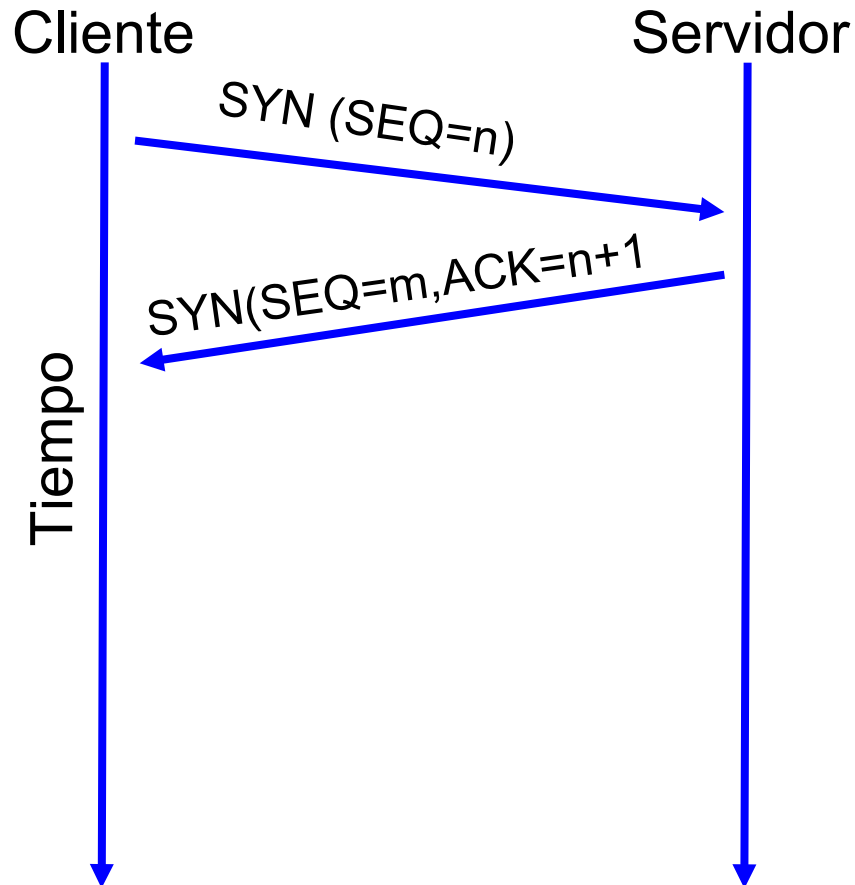
El protocolo TCP: estableciendo la conexión



El cliente envía una solicitud de conexión

Capa de Transporte

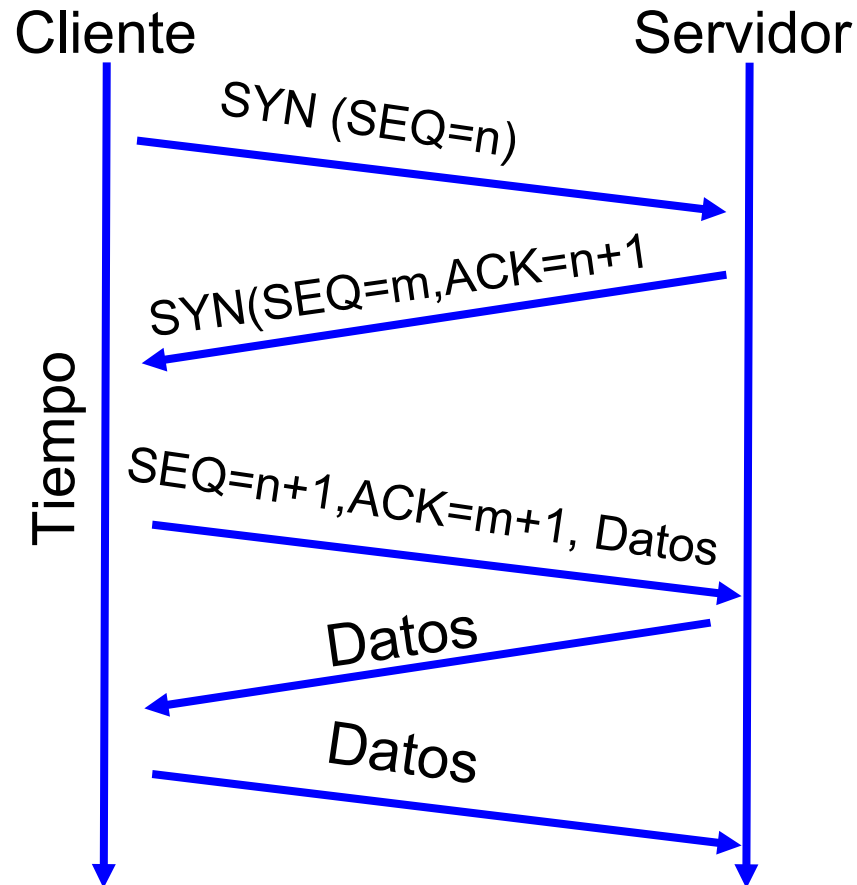
El protocolo TCP: estableciendo la conexión



La entidad TCP verifica que haya un proceso que atienda el puerto solicitado y responde aceptando la conexión. Si no envía $RST = 1$

Capa de Transporte

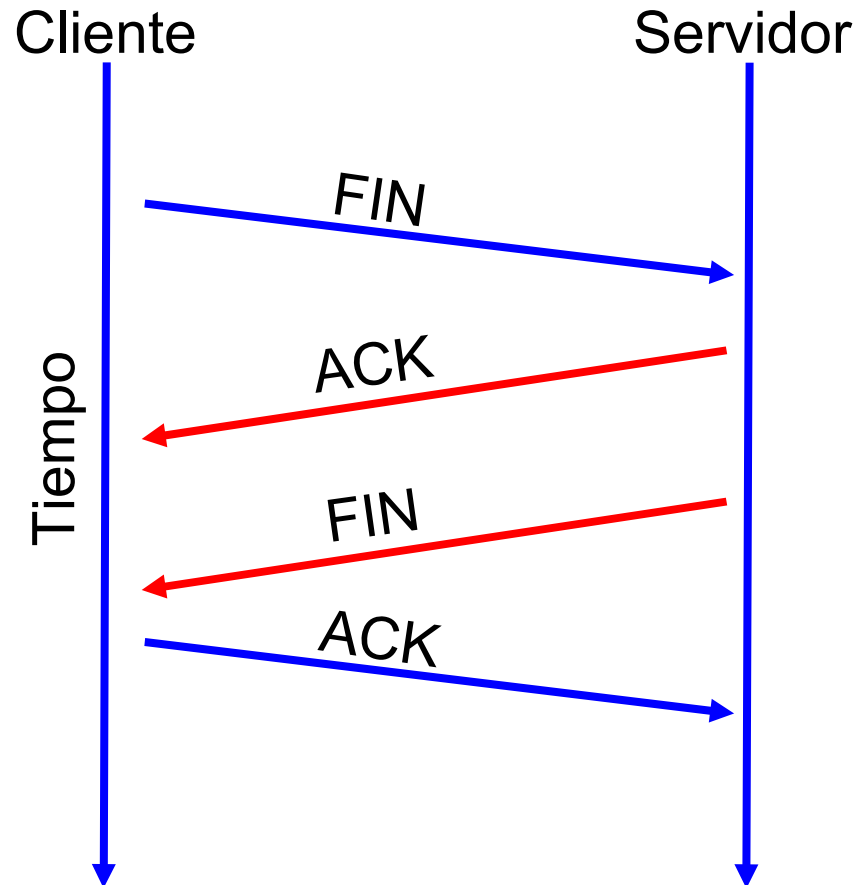
El protocolo TCP: estableciendo la conexión



Luego el cliente envía el reconocimiento de la aceptación y datos

Capa de Transporte

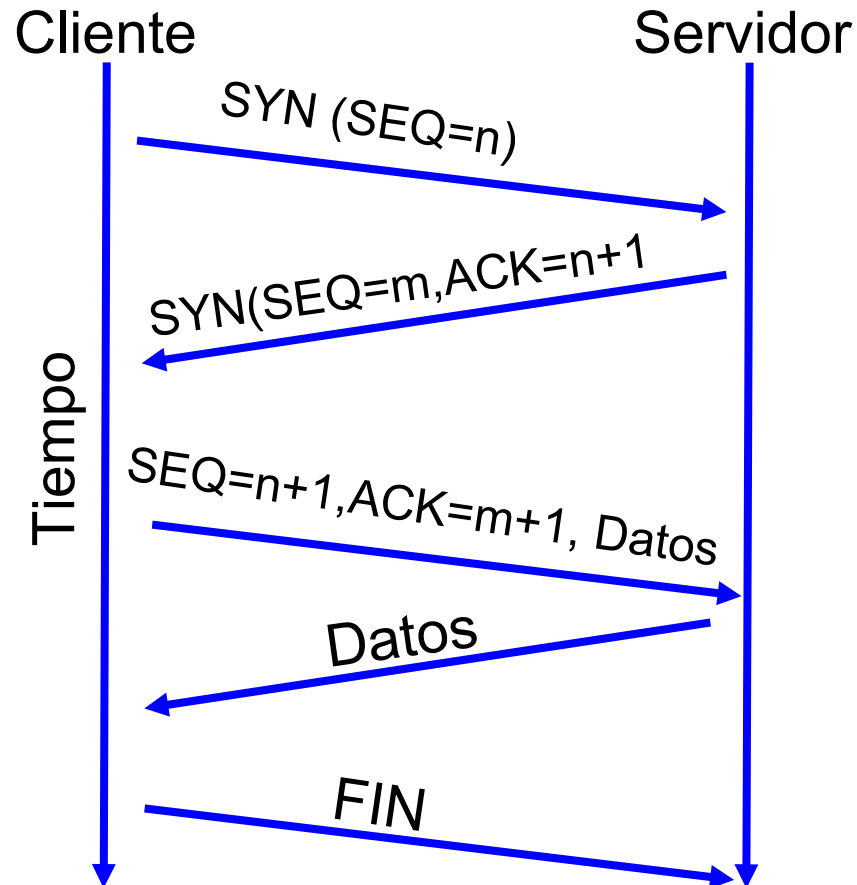
El protocolo TCP: cerrando la conexión



Se necesitan 4 segmentos para cerrar la conexión pues es full-duplex. Un FIN y un ACK por cada dirección

Capa de Transporte

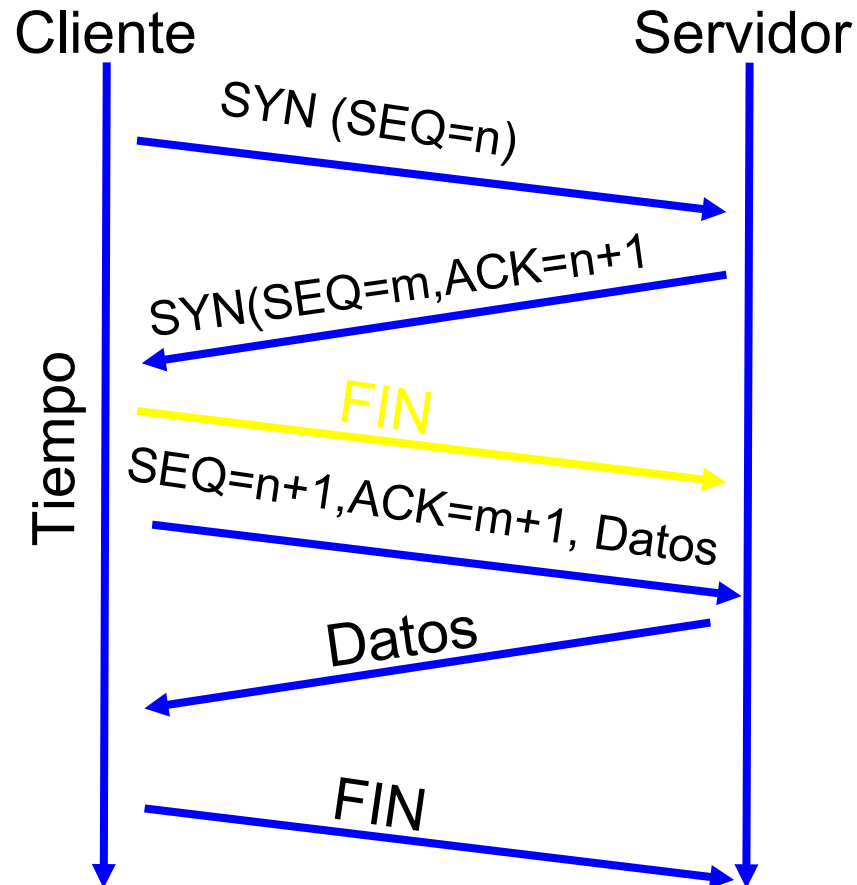
El protocolo TCP: Ataques



El flujo de segmentos normal

Capa de Transporte

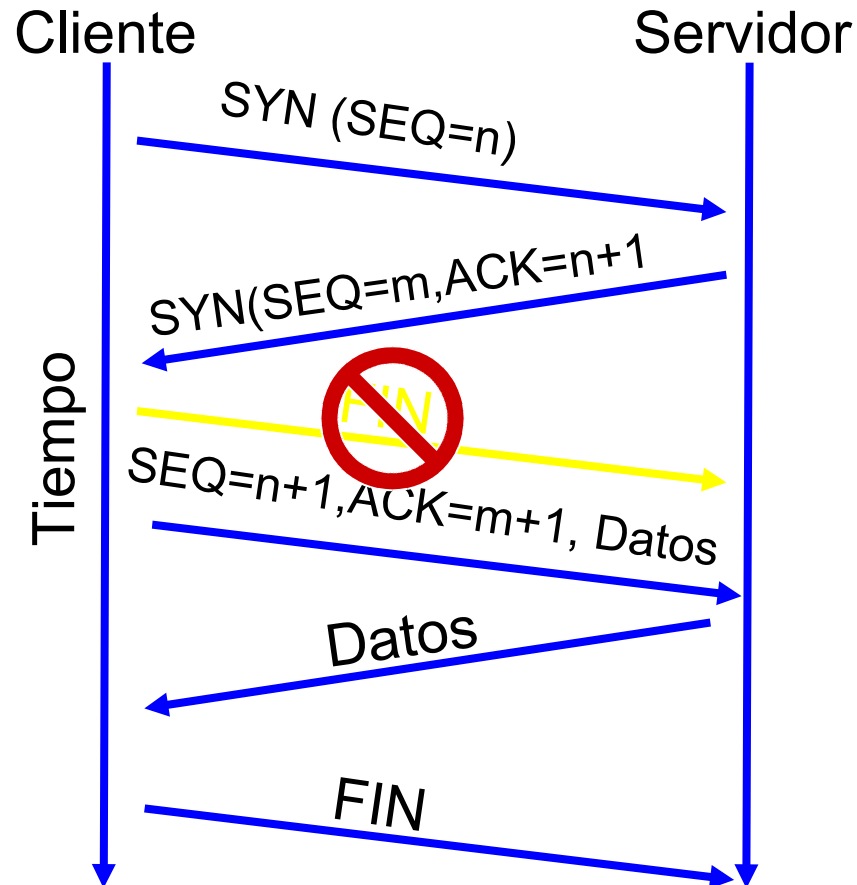
El protocolo TCP: Ataques



- **Ataque**
Paquetes legales
Protocolo ilegal

Capa de Transporte

El protocolo TCP: Ataques



- Remedio
recordar estado del protocolo

Capa de Transporte

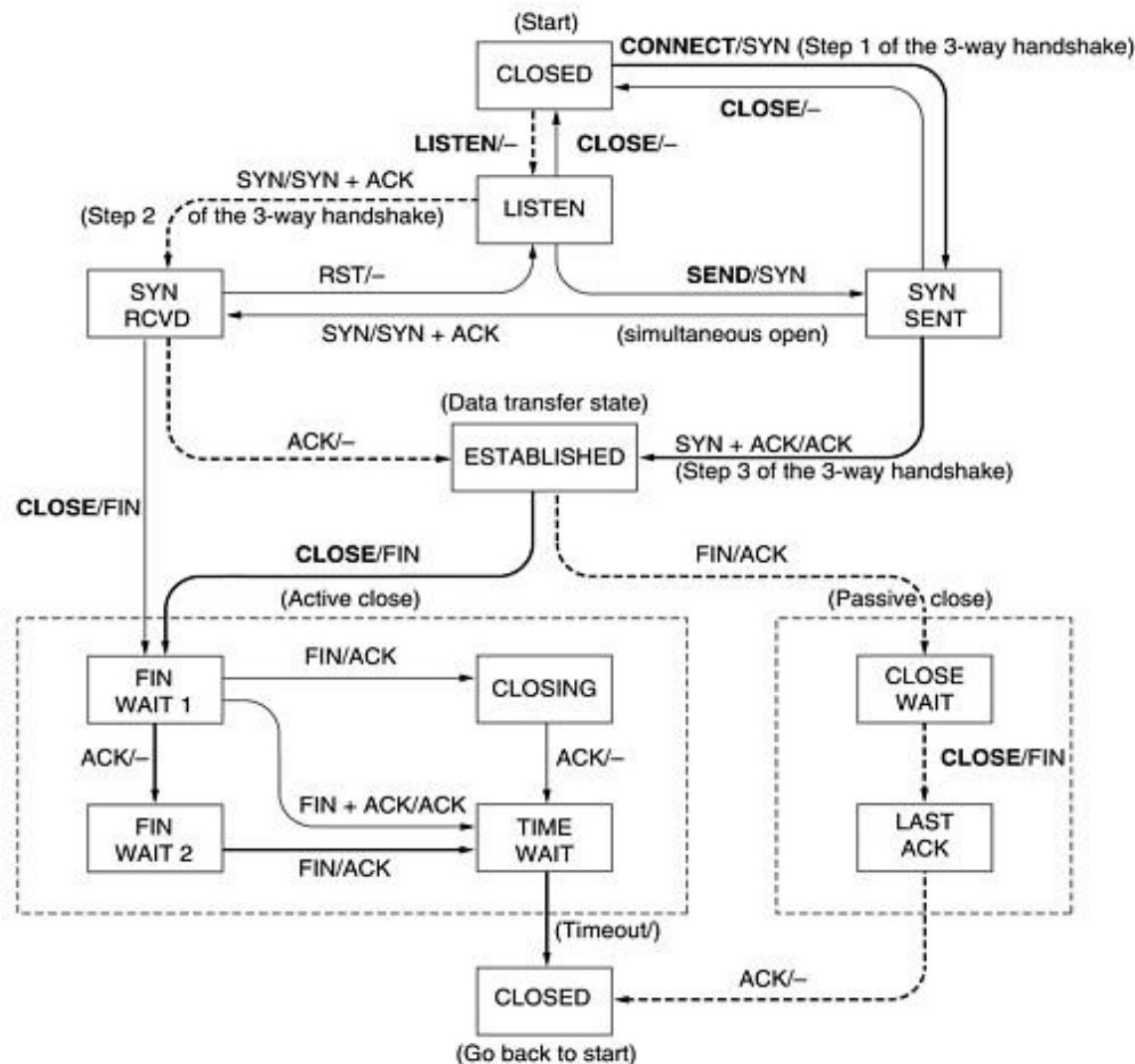
Modelo de gestión de conexiones en TCP

TCP puede ser representado por una máquina de estado finito con los siguientes estados:

CLOSED	No hay conexión activa o pendiente
LISTEN	El servidor espera por una solicitud
SYNC RCVD	Arriba una solicitud de conexión. Se espera un ACK
SYNC SENT	La aplicación comenzó a establecer una conexión
ESTABLISHED	Estado normal de transmisión de datos
FIN WAIT 1	La aplicación indicó que ya terminó
FIN WAIT 2	El otro extremo indicó que ya terminó
TIMED WAIT	Espera por todos los paquetes para morir
CLOSING	Ambos extremos han indicado cerrar la conexión simultaneamente
CLOSE WAIT	El otro extremo inició un cierre de conexión
LAST ACK	Espera por todos los paquetes para morir

Capa de Transporte

Modelo de gestión de conexiones en TCP



Capa de Transporte

Transmisión de datos en TCP

La transmisión de datos en TCP se hace utilizando un mecanismo denominado ventana deslizable.

Como ejemplo podemos mencionar:

Si el receptor tiene un buffer de 4096 Bytes y el receptor envía un segmento de 2048 Bytes éste es recibido sin ningún inconveniente y el receptor envía un ACK, sin embargo, mientras el receptor saca los datos del buffer, le indicará al emisor que ahora sólo dispone de 2048 Bytes en el buffer.

Capa de Transporte

Transmisión de datos en TCP

Emisor



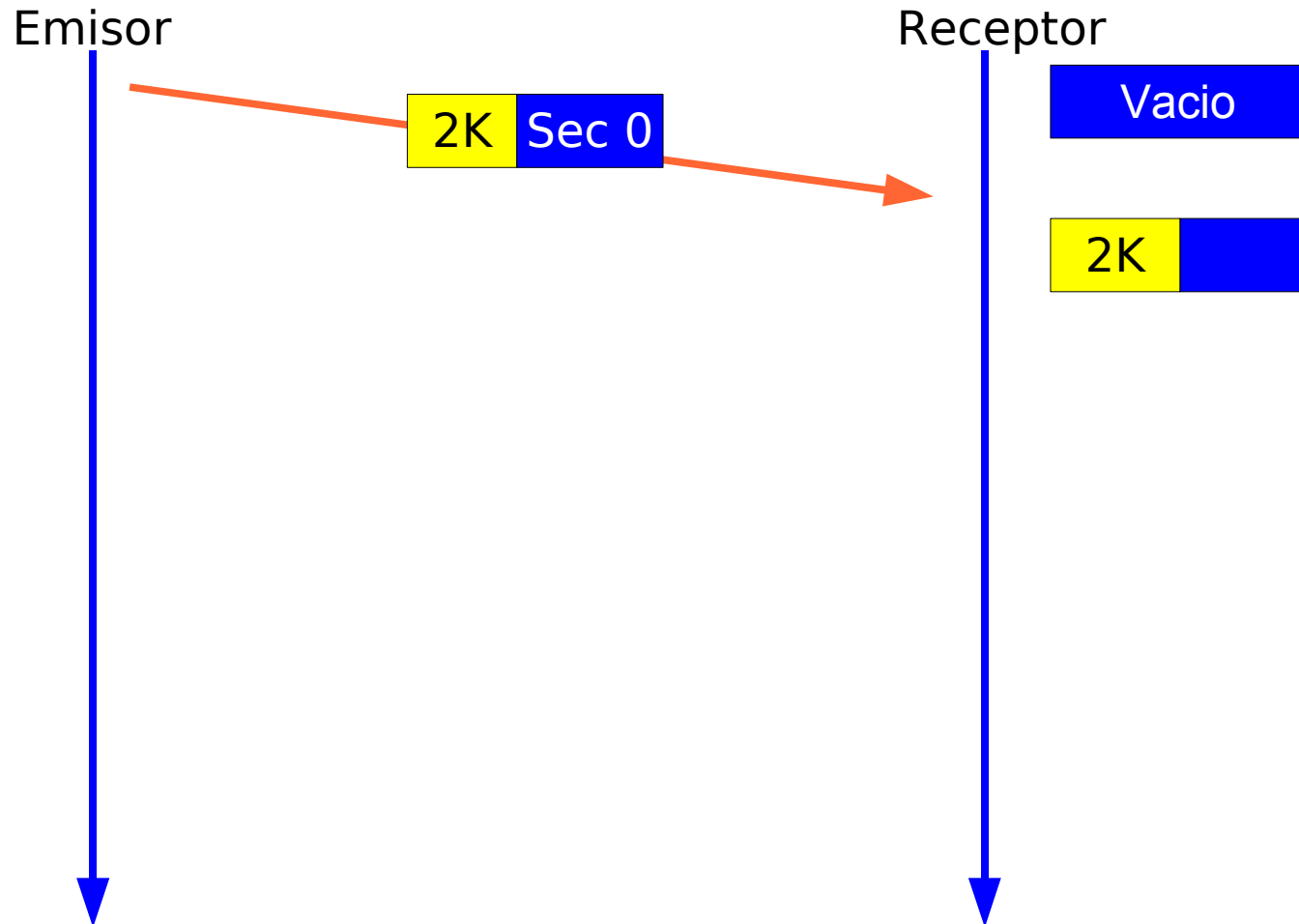
Receptor

Vacio



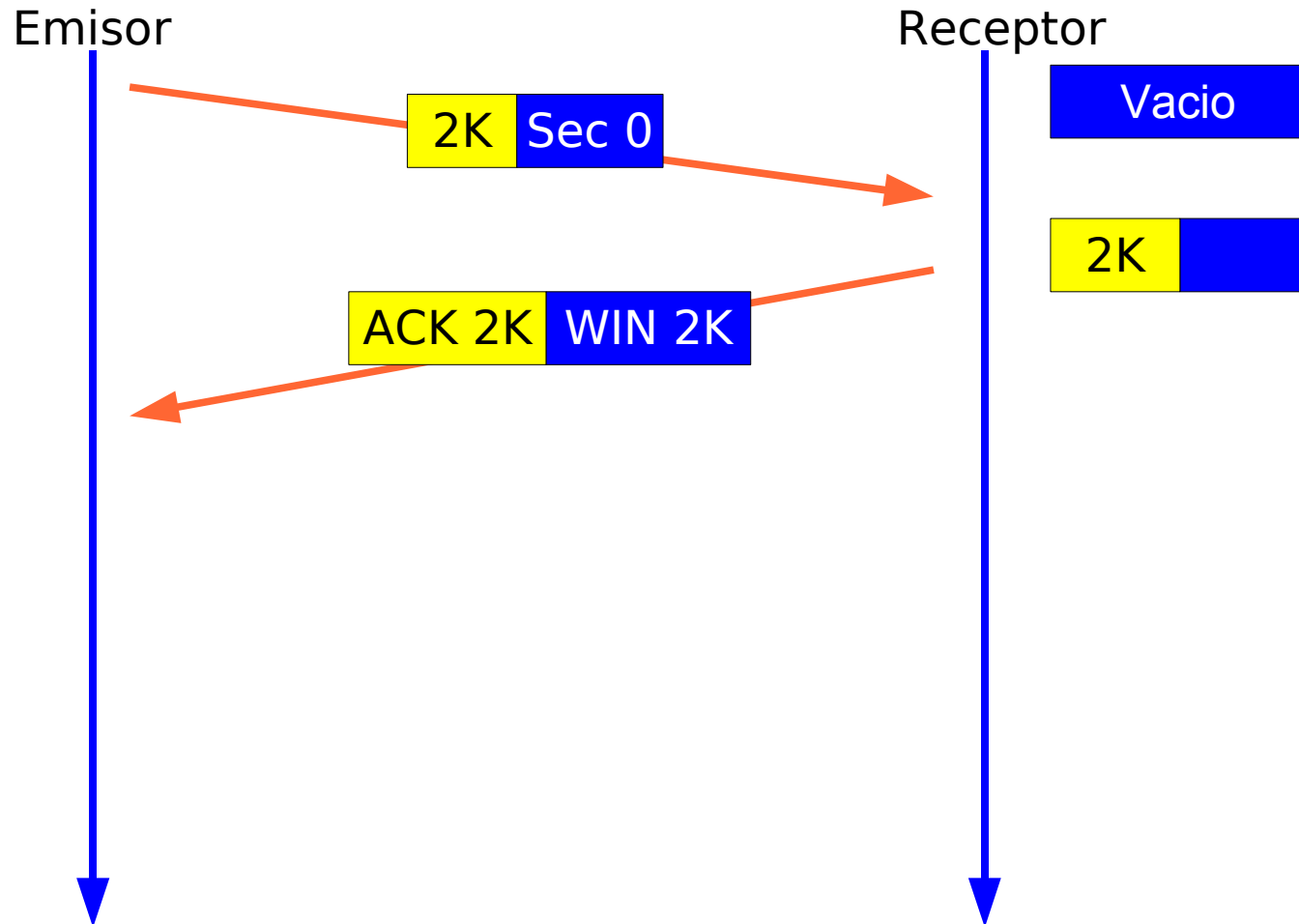
Capa de Transporte

Transmisión de datos en TCP



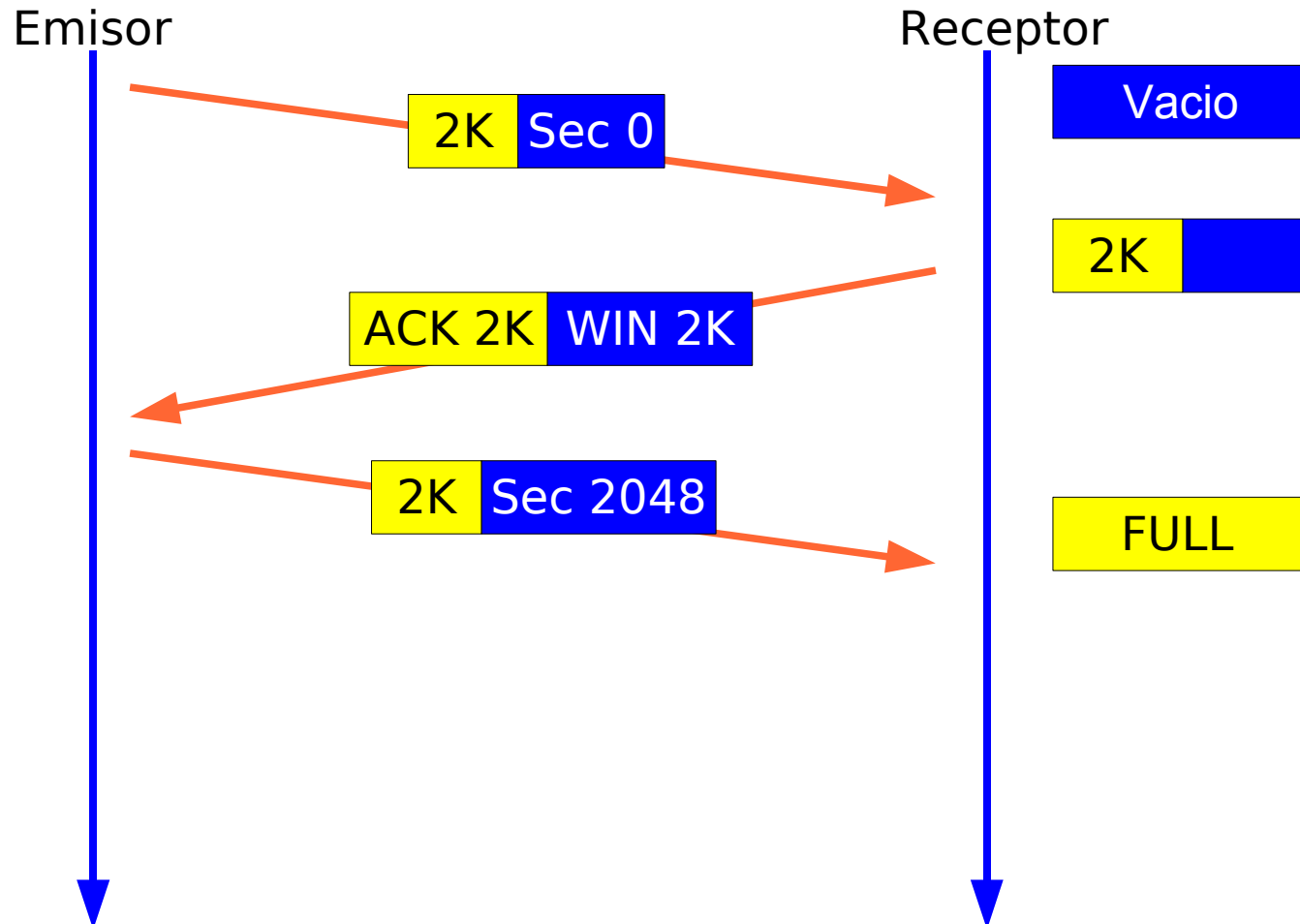
Capa de Transporte

Transmisión de datos en TCP



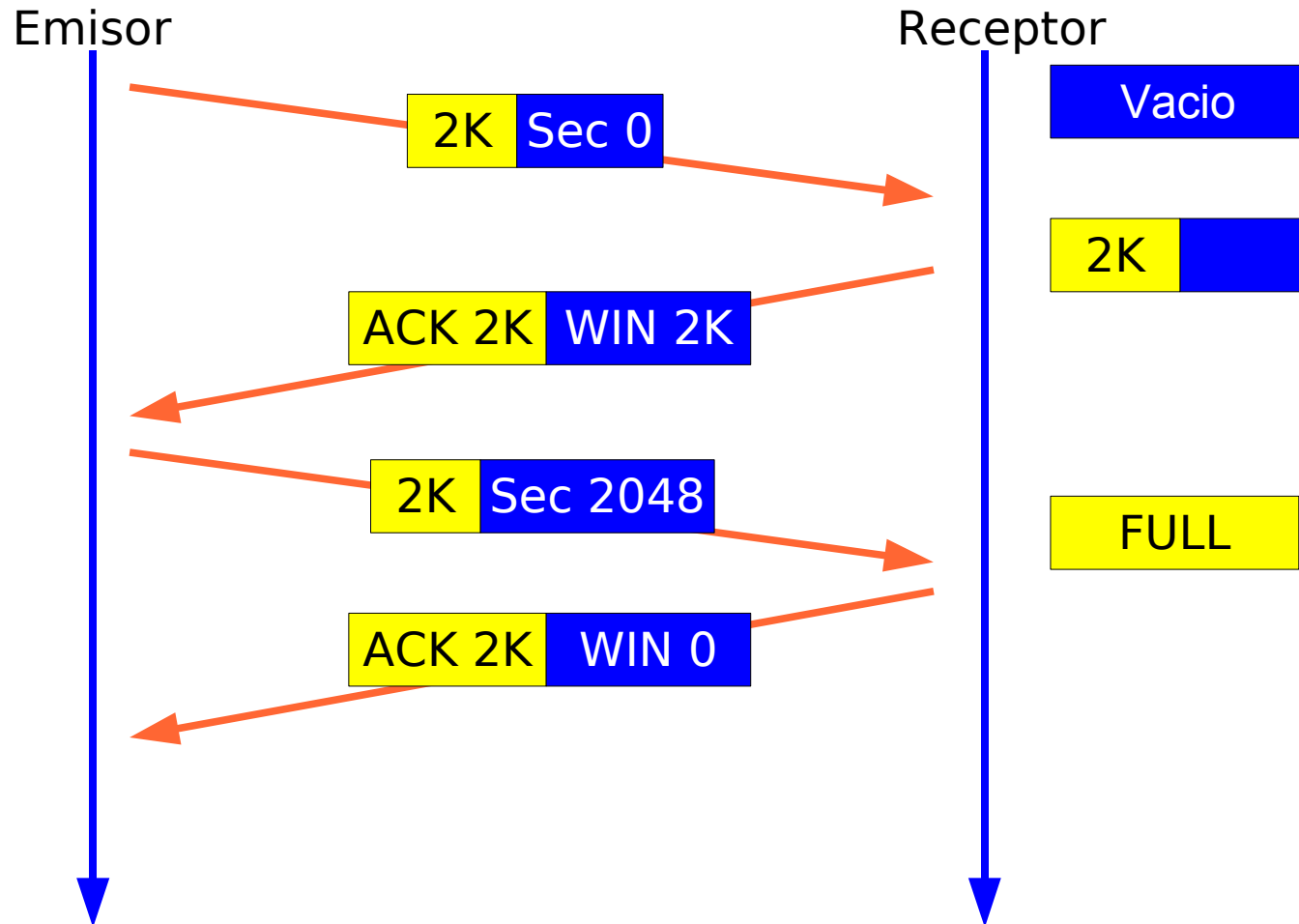
Capa de Transporte

Transmisión de datos en TCP



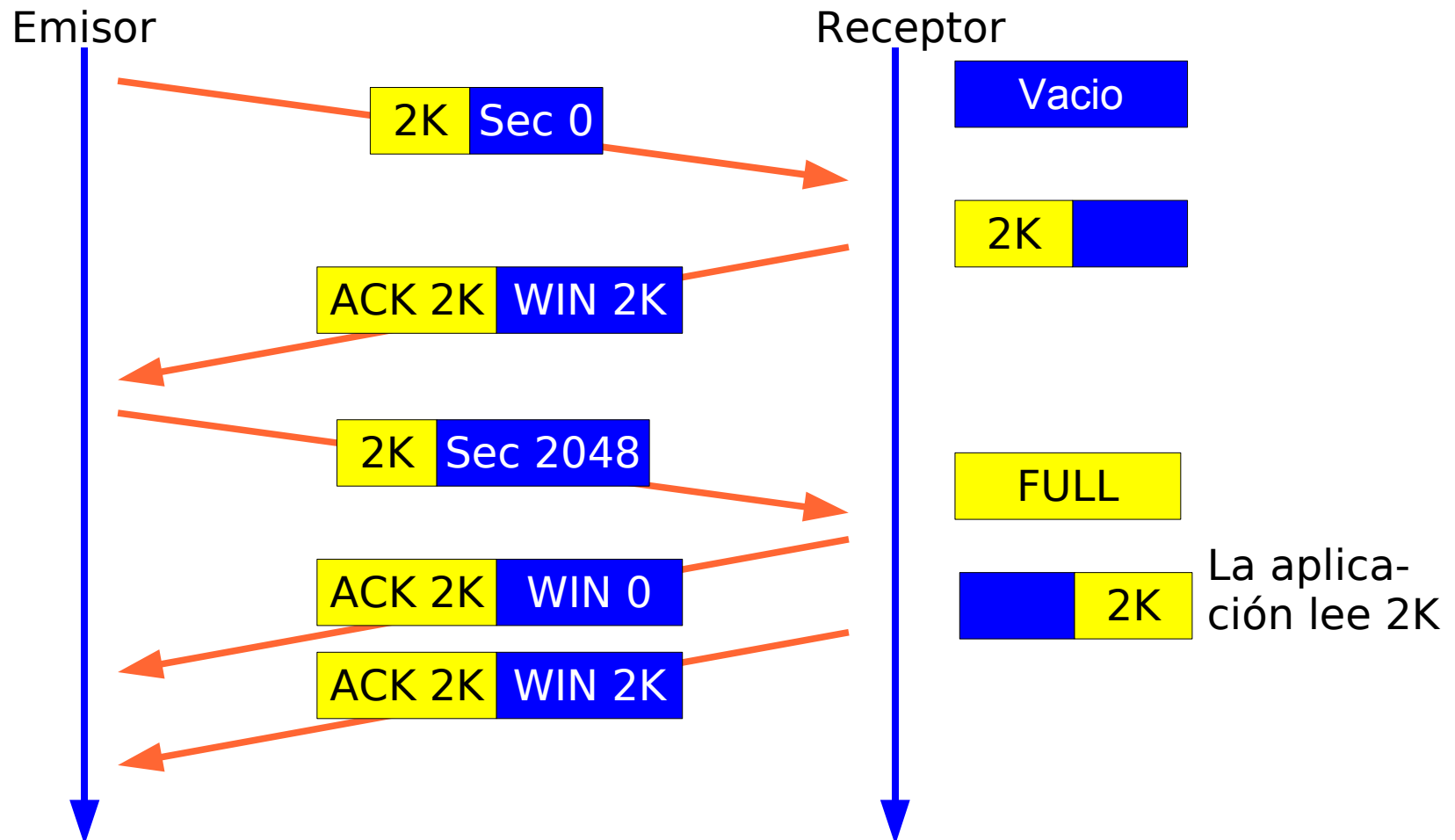
Capa de Transporte

Transmisión de datos en TCP



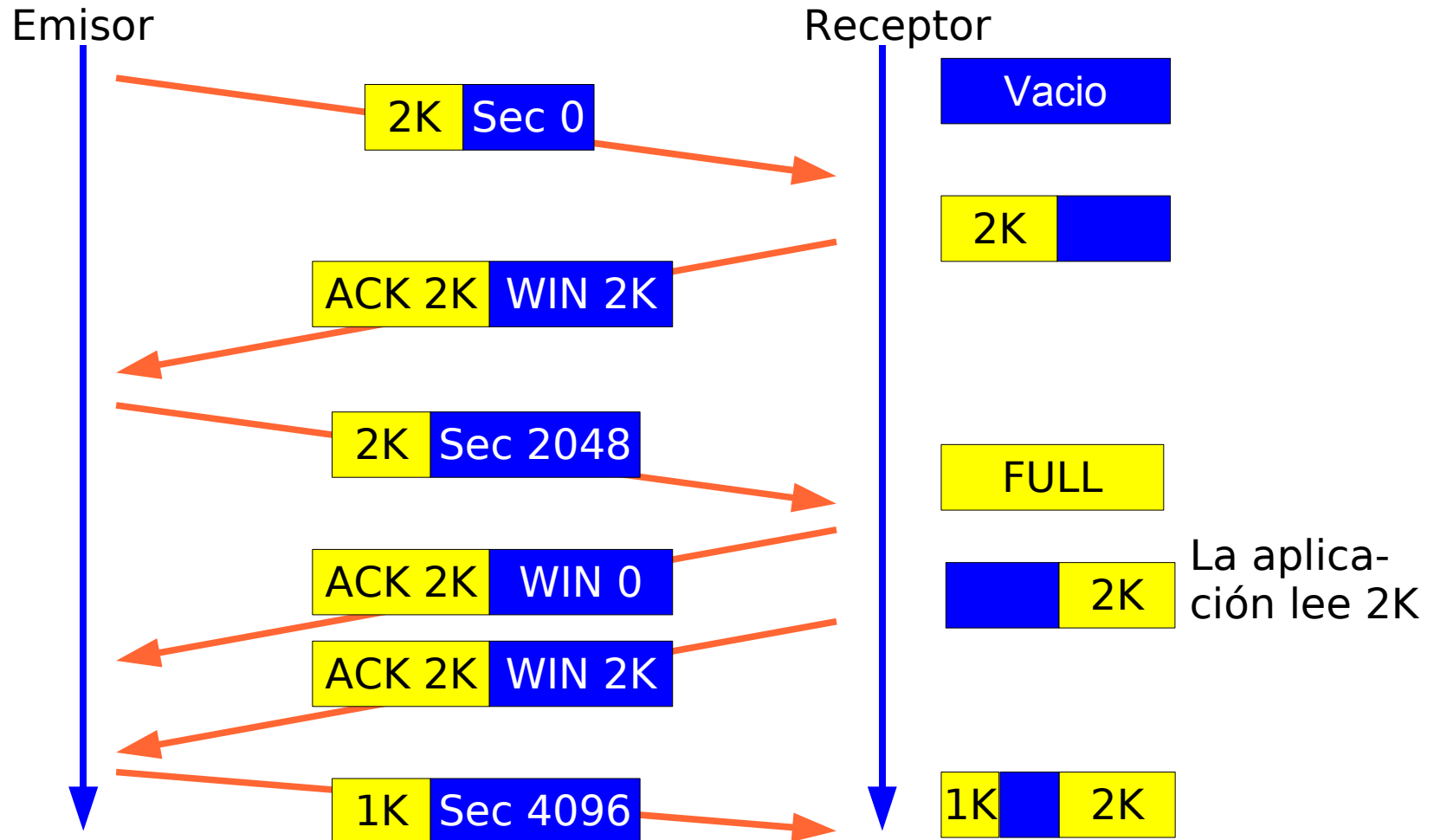
Capa de Transporte

Transmisión de datos en TCP



Capa de Transporte

Transmisión de datos en TCP



Capa de Transporte

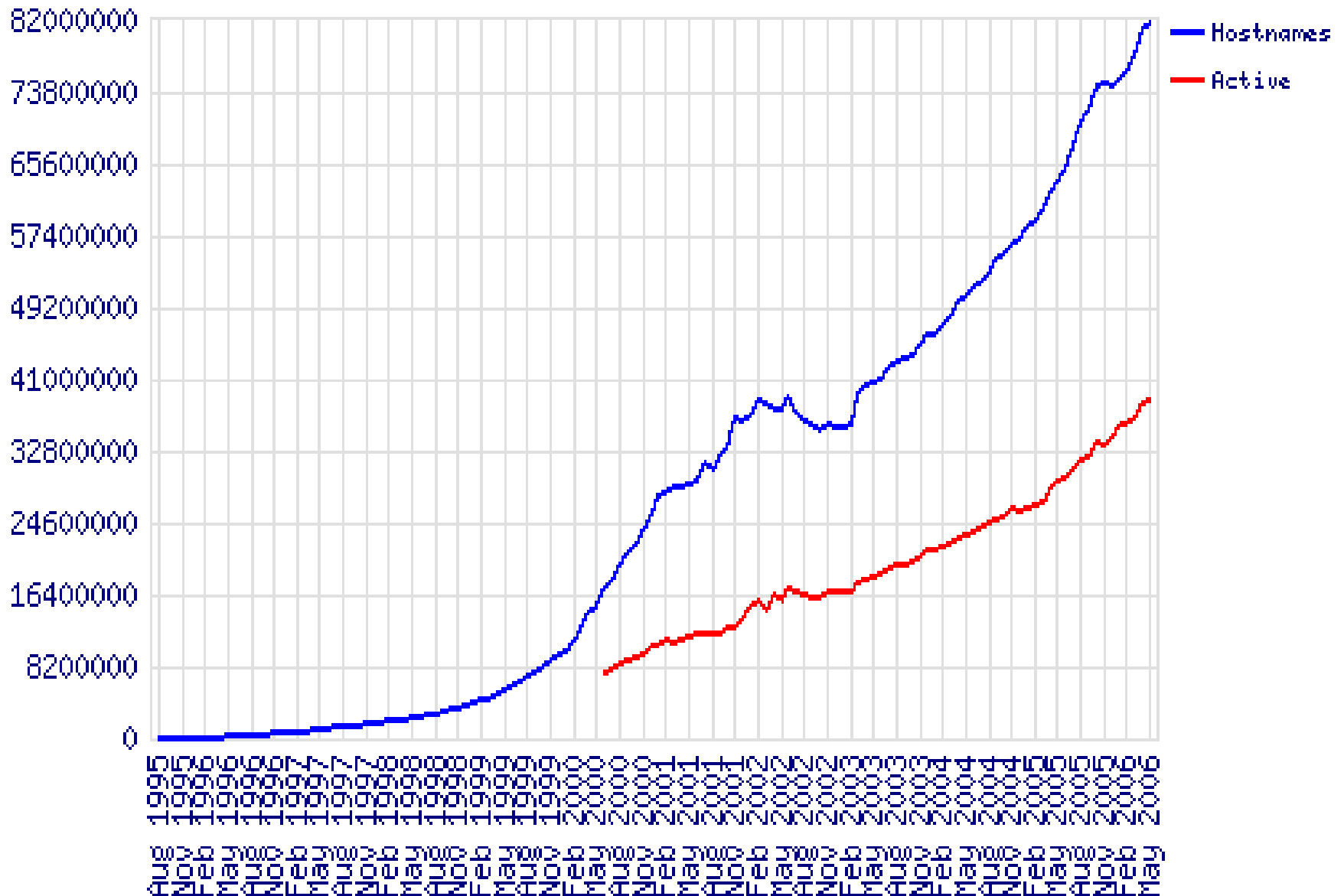
Protocolo de Ejemplo (HTTP)

Uno de los más utilizados protocolos en Internet se fundamenta en TCP.

El servidor por excelencia utilizado para este servicio es apache

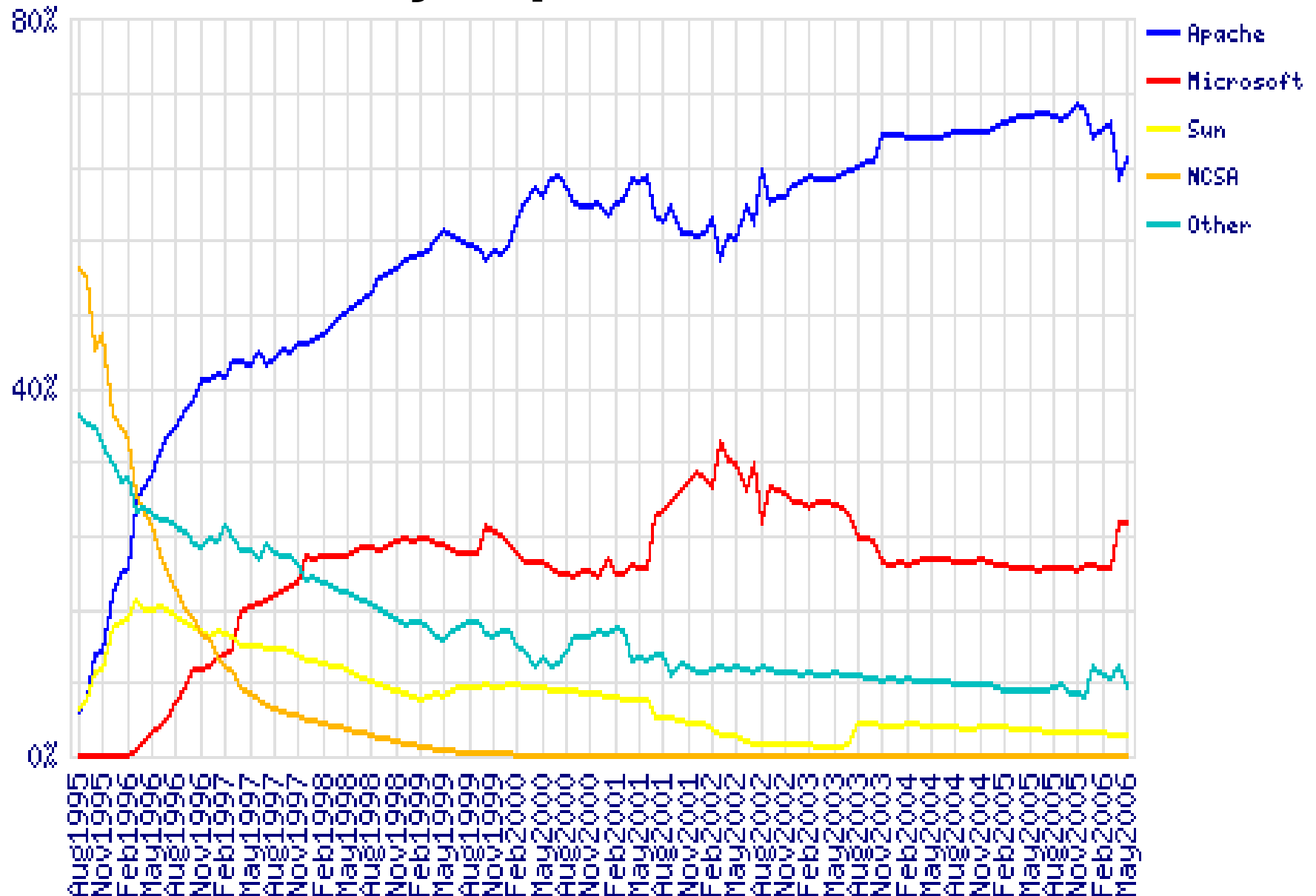
Capa de Transporte

Protocolo de Ejemplo (HTTP)



Capa de Transporte

Protocolo de Ejemplo (HTTP)



Capa de Transporte

Control de Congestión en TCP