

Comunicación de datos

Capa 4: TCP



Profesores: Lic. Alejandro Mansilla
Ing. Rodrigo A. Elgueta
2019

Capa de transporte

- El protocolo de transporte proporciona un servicio de transferencia de datos extremo a extremo que aísla las capas superiores de los detalles de la red o redes intermedios.
- Orientado a conexión como TCP o no orientado a conexión: UDP.
- Si el servicio de la red o interconexión subyacente no es fiable, como ocurre en el caso de IP, un protocolo fiable de transporte orientado a conexión resulta ser muy complejo.

Servicio de entrega ordenado y fiable

- El protocolo de transporte se utiliza como un protocolo extremo a extremo entre dos sistemas finales conectados a la misma red.
- Cuatro aspectos importantes a tener en cuenta:
 - Direccionamiento
 - Multiplexación
 - Control de flujo
 - Establecimiento y cierre de la conexión

Direccionamiento

- La dirección del usuario se identifica normalmente como:
<estación : puerto>
- Si dos servicios están presentes en el mismo puerto deberá especificarse el tipo de protocolo *UDP* o *TCP* (solo 1 por cada tipo)
- El encaminamiento o ruteo, no es tarea de esta capa, por lo tanto el protocolo de transporte solo informa a la capa de red la dirección destino. El puerto se agrega a las cabeceras para que lo use la capa de transporte del destino.
- Para conocer la dirección y puerto destino:
 - Servidores de nombre y puertos bien conocidos
 - Conocer de antemano esos valores

Multiplexación

- Los protocolos realizan multiplexado/demultiplexado en relación a capas superiores
- Múltiples usuarios utilizando la misma capa de transporte se distinguen mediante el puerto y el punto de acceso.
- Cada conexión se identifica por:
- **Dirección origen, puerto origen, Dirección de destino, puerto destino, Protocolo.**

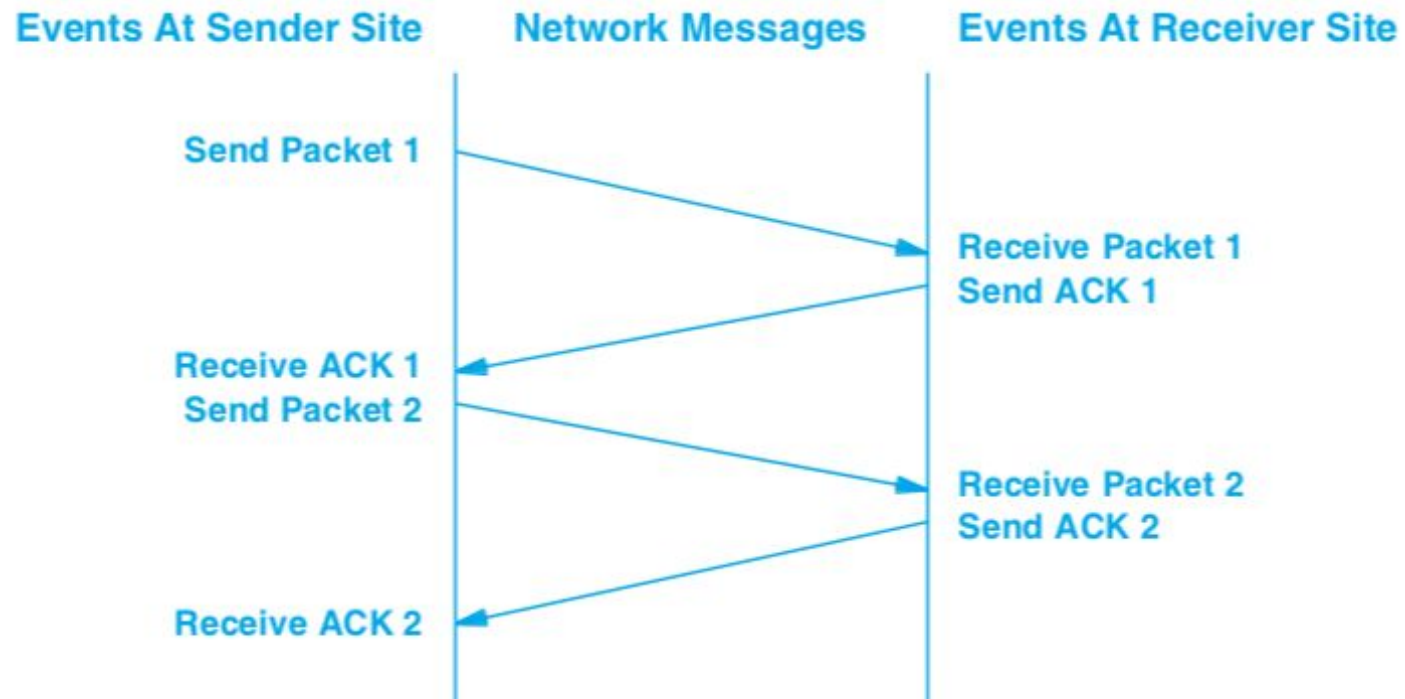
Control de flujo

- Mecanismo bastante más complejo que en Capa 2
- Dos motivos:
 - El retardo de transmisión entre entidades de transporte es generalmente grande comparado con el tiempo de transmisión real.
 - Ya que la capa de transporte opera sobre una red o una interconexión de redes (*normalmente IP*), la cantidad de retardo en la transmisión puede ser muy variable. Esto hace difícil utilizar de forma efectiva un mecanismo de tiempos de expiración para la retransmisión de datos perdidos.
- Ante un requisito de control de flujo las estaciones pueden:
 - No hacer nada
 - Rechazar aceptación de segmentos
 - Usar protocolo de ventana deslizante
 - Usar un esquema de crédito

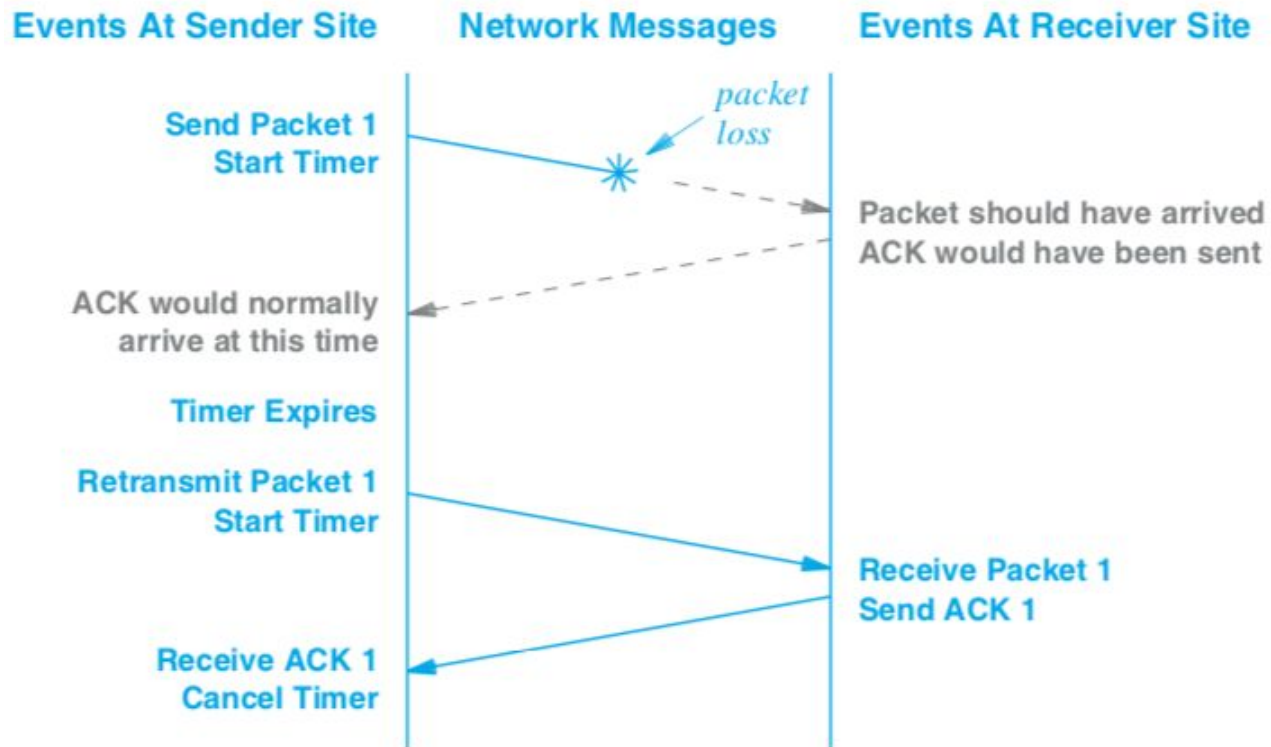
Entrega confiable

- TCP es un protocolo de servicio de entrega confiable
- Características principales:
 - Orientado a flujos
 - Conexión de circuitos virtuales
 - Transferencia de memoria inmediata
 - Flujo no estructurado (*tcp no identifica el contenido de lo que transporta, ni impone límites*)
 - Full Duplex

Funcionamiento



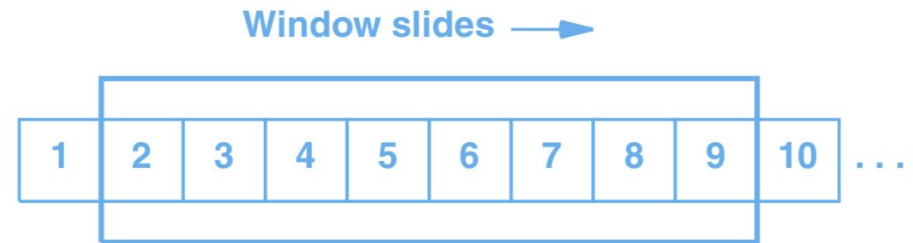
Funcionamiento



Ventana deslizable



(a)



(b)

- Un protocolo simple de acuses de recibo positivos ocupa una cantidad sustancial de ancho de banda de red debido a que debe retrasar el envío de un nuevo paquete hasta que reciba un acuse de recibo del paquete anterior.
- **Un protocolo de ventana deslizable bien establecido mantiene la red completamente saturada de paquetes, con él se obtiene una generación de salida mas alta que con un protocolo simple de acuse de recibo positivo.**

Ventana deslizable

Events At Sender Site

Network Messages

Events At Receiver Site

Send Packet 1

Send Packet 2

Send Packet 3

Receive ACK 1

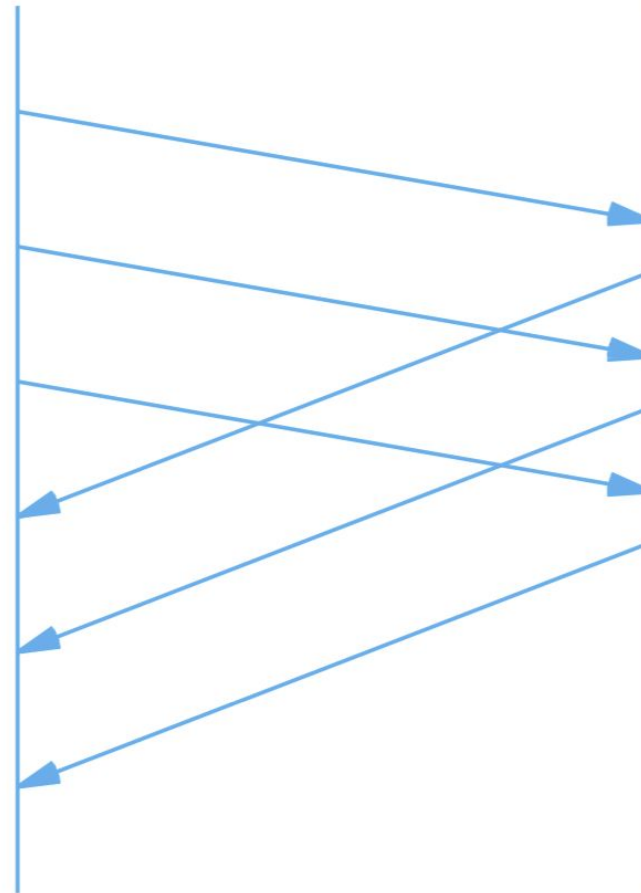
Receive ACK 2

Receive ACK 3

Receive Packet 1
Send ACK 1

Receive Packet 2
Send ACK 2

Receive Packet 3
Send ACK 3



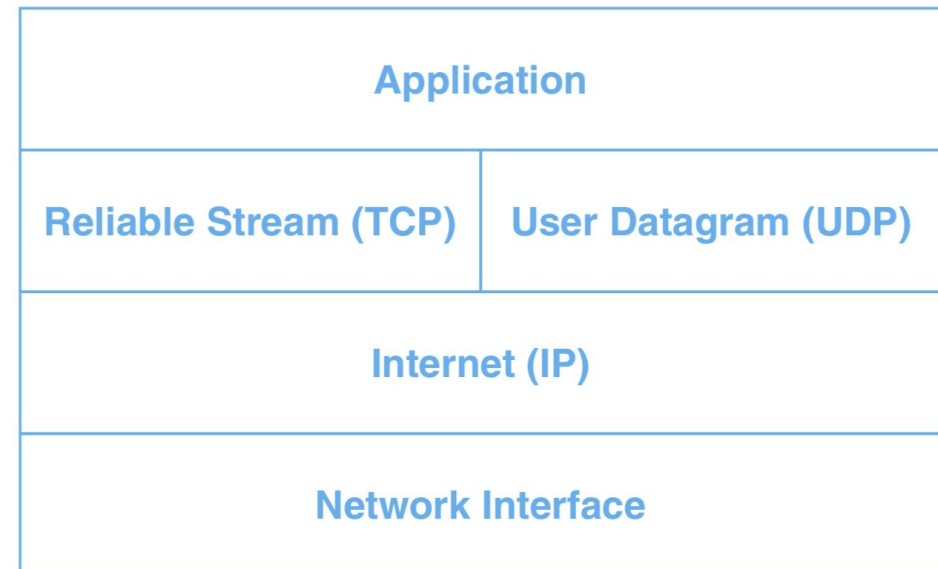
TCP

- Transmission Control Protocol
- Es **EL** protocolo de servicio de flujo confiable proporcionado por el grupo de protocolos TCP/IP de Internet.
- Es tan importante, que define el nombre del conjunto
- TCP define:
 - Formato de mensajes y acuses de recibo
 - Procedimientos para asegurar que los datos lleguen
 - Cómo distinguir el destino correcto
 - Cómo resolver pérdidas o duplicidades de paquetes
 - Cómo iniciar y finalizar conexiones entre origen y destino

TCP: puertos y conexiones

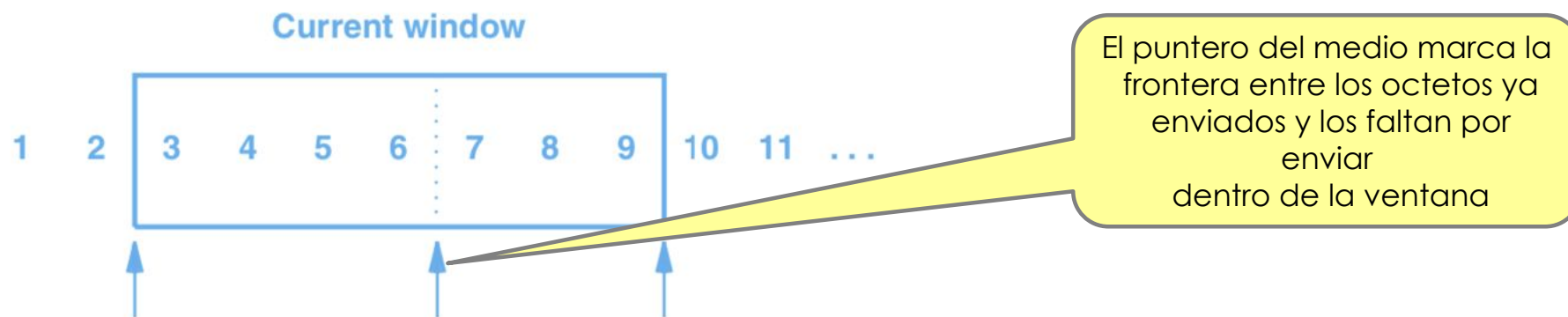
- TCP vive sobre IP
- La abstracción fundamental de TCP es la **conexión** la cual se identifica por un par de puntos extremos.
- Los puntos extremos son identificados por (*IP Host, puerto*) expresados como números enteros ej:
(192.168.3.1, 25)
- Ejemplo de conexión: (8.8.8.8, 53) y (192.168.3.104, 1184)

Conceptual Layering



Segmentos, flujos y números de secuencia

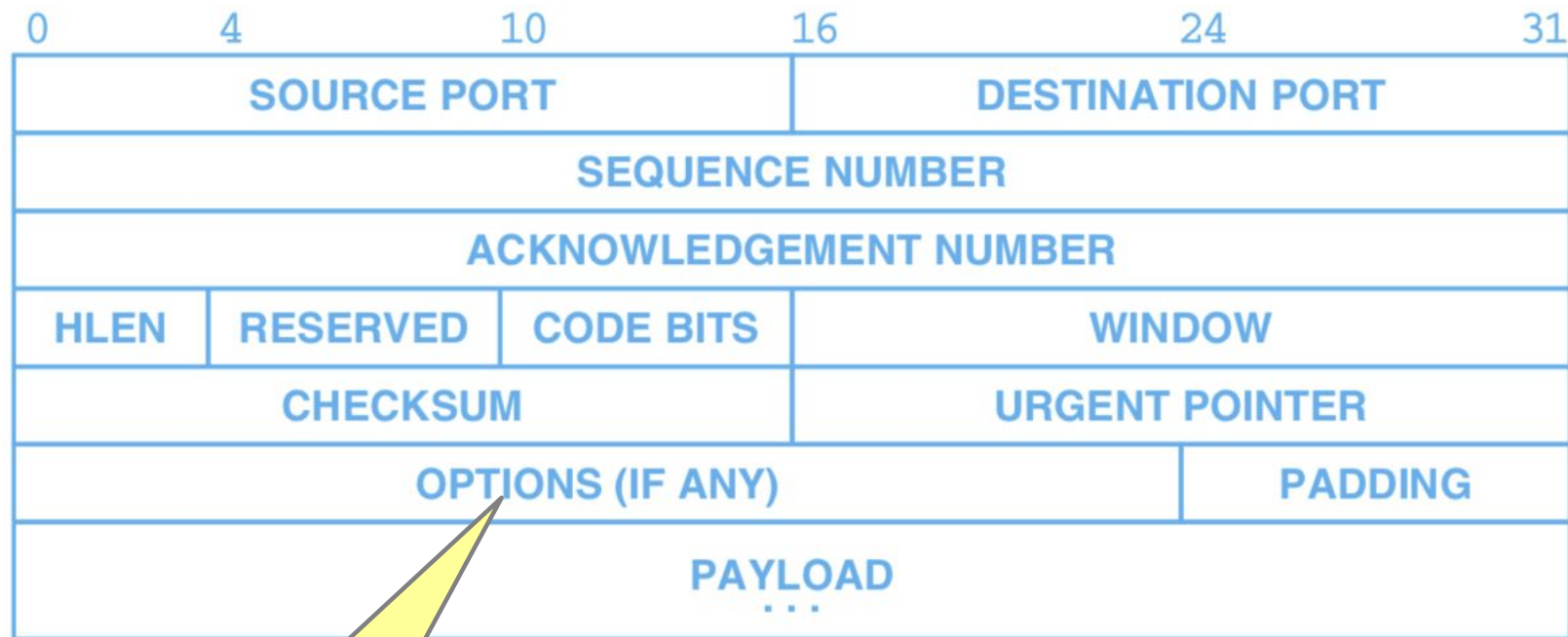
- TCP visualiza el flujo como una secuencia de bytes que divide en **Segmentos**
- El mecanismo de ventana deslizante de TCP opera a nivel de **octetos** y no de segmentos o de paquete.
- Los octetos del flujo se numeran de manera secuencial
- La ventana deslizante se define a través de 3 apuntadores



Tamaño de ventana y control de flujo

- Cada acuse de recibo que informa cuantos octetos se recibieron incluye un aviso de ventana. Que indica cuanto mas puede recibir el receptor.
- De esta manera se realiza el control de flujo, agrandando o achicando el tamaño de la ventana
- Este tipo de mecanismos son esenciales en entornos como Internet o una red de redes

Formato del Segmento TCP

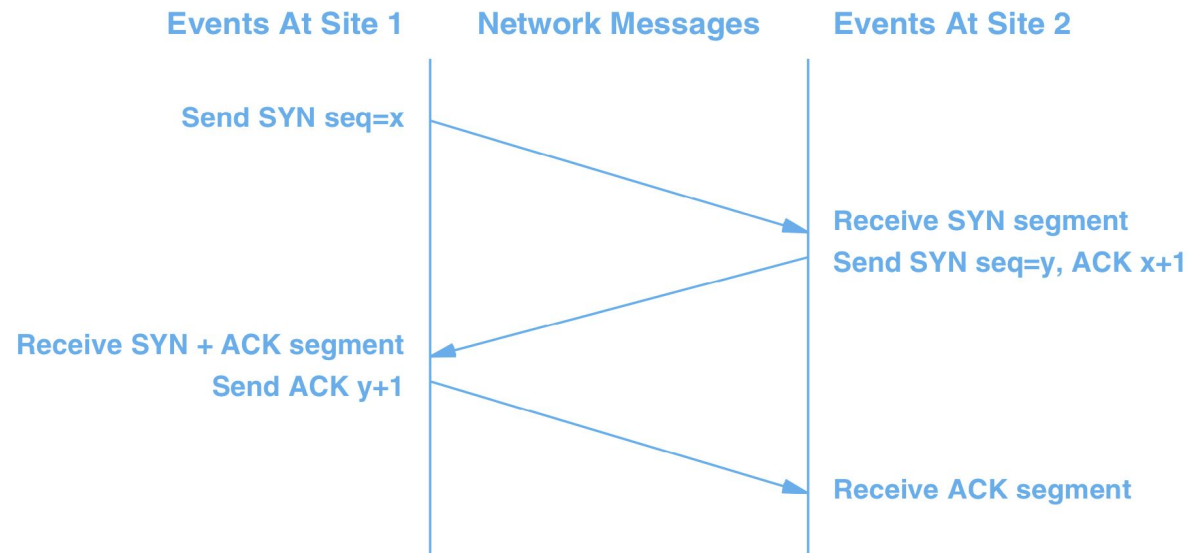


Un valor que suele ir acá es el tamaño máximo de segmento o MSS.

Bit (left to right)	Meaning if bit set to 1
URG	Urgent pointer field is valid
ACK	Acknowledgement field is valid
PSH	This segment requests a push
RST	Reset the connection
SYN	Synchronize sequence numbers
FIN	Sender has reached end of its byte stream

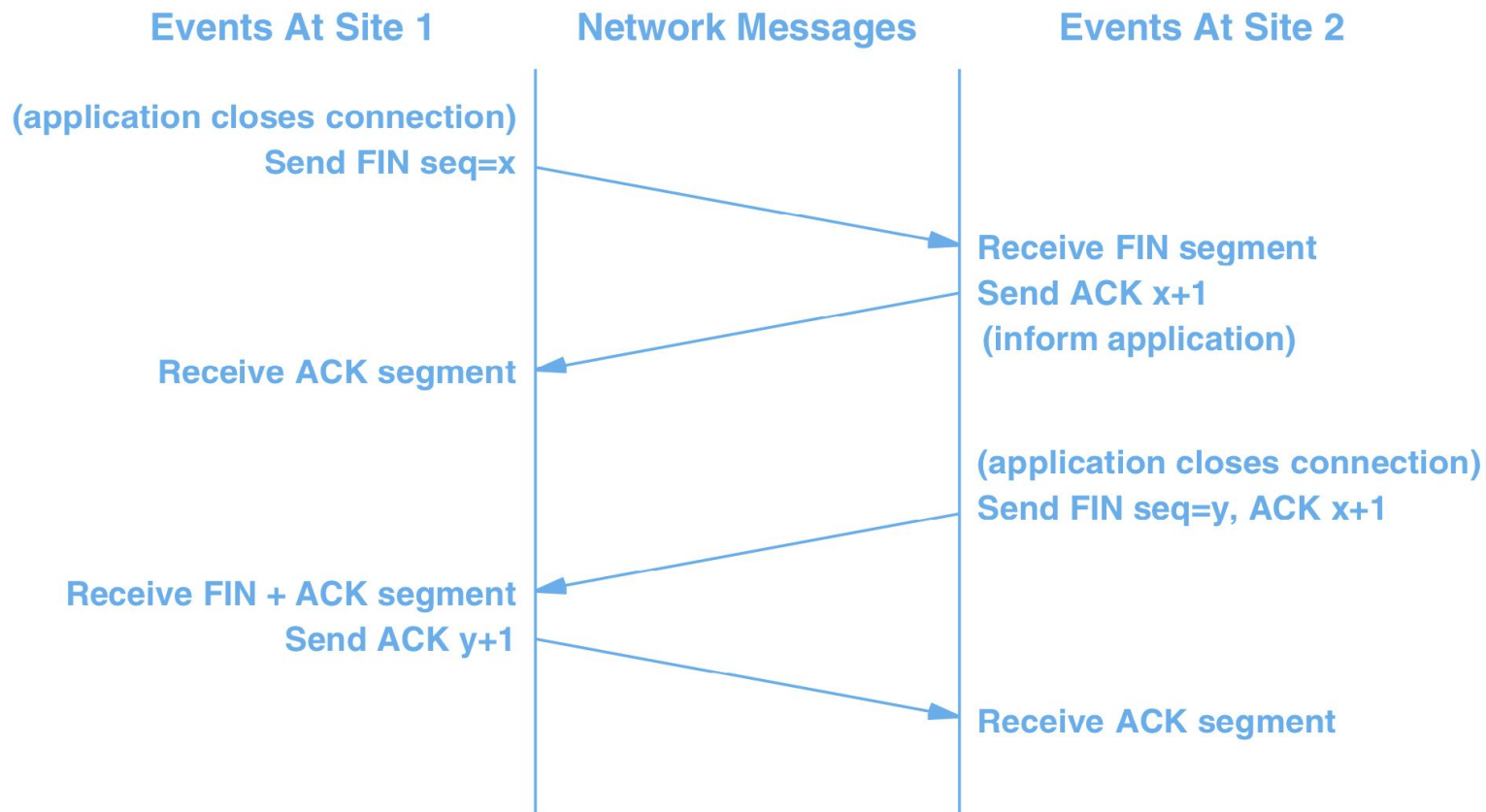
Establecimiento de una conexión TCP

- TCP utiliza un saludo o **Handshake** de 3 etapas
 - 1^{er} segmento de saludo, identificable porque tiene el bit **SYN** activo en el campo *CODE*
 - 2^{do} segmento tiene tanto el bit **SYN** como el **ACK** activo indicando acuse de recibo del primer segmento **SYN** y que continúa con el intercambio.
 - 3^{er} y último segmento con el bit **ACK** confirmando que la conexión está establecida.

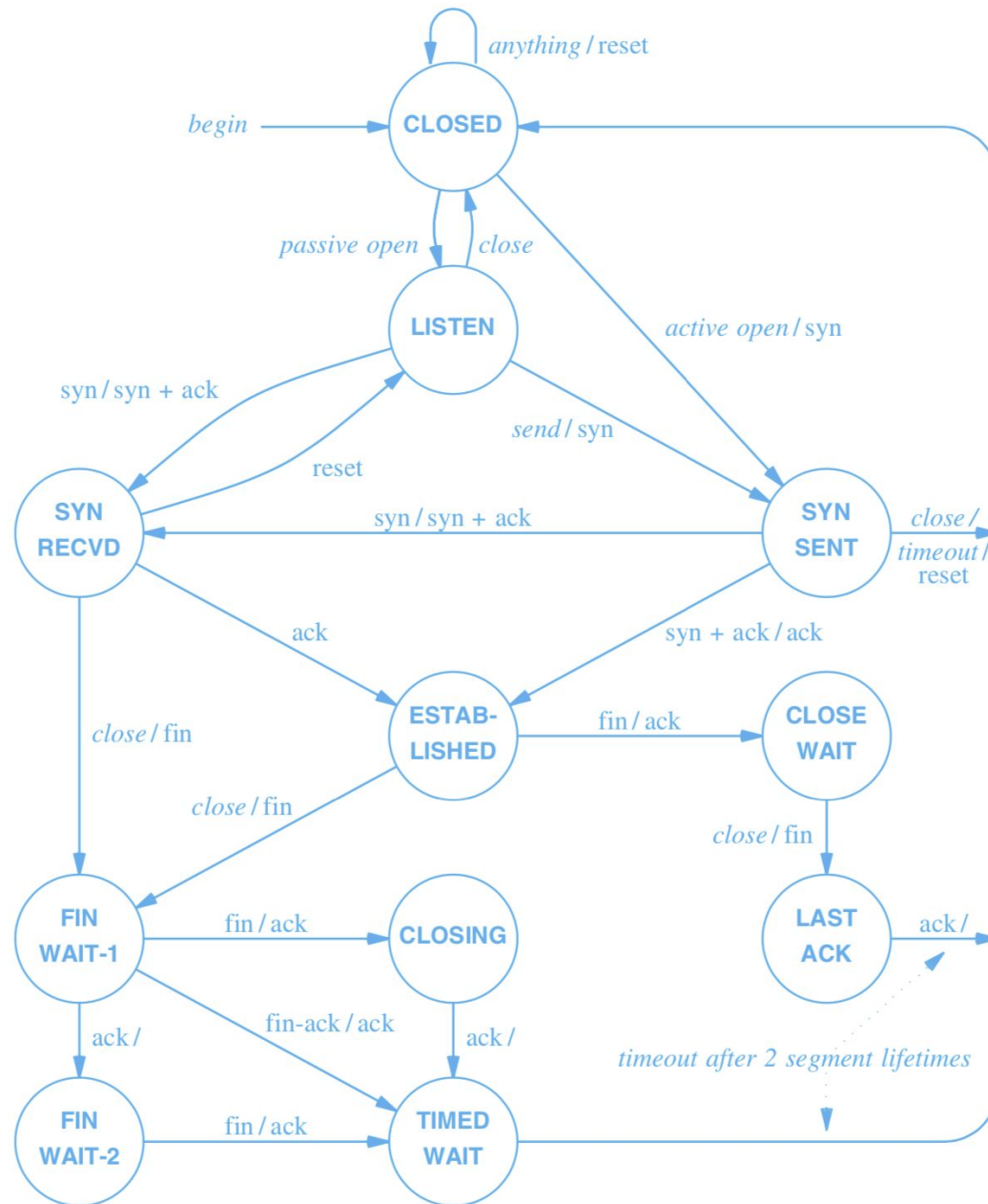


Terminación de una Conexión

- Saludo de 3 etapas modificado para cerrar la conexión
- Se usa el bit **FIN** y la modificación es que no vuelve a mandar **FIN,ACK** como pasa en el establecimiento, sino que el receptor solo envía un **ACK**



Máquina de estado finito TCP



FIN



SYN

ACK