



VNIVERSIDAD
D SALAMANCA

<http://informatica.usal.es/gii>

Tema 7: El nivel de transporte

Introducción

Servicios proporcionados al nivel de sesión

Protocolo de transporte

Elementos de los protocolos de transporte

El Nivel de transporte en Internet

TCP y UDP



VNIVERSIDAD
D SALAMANCA



Introducción

- **Función:** proporcionar fiabilidad y un transporte de datos eficiente, efectivo y seguro, independientemente de la red o redes en uso (optimizar los servicios del nivel de red)
- Se encarga de:
 - Selección del servicio de red (Negociación)
 - Determinar la necesidad de multiplexado
 - Optimizar el tamaño de la unidad de datos
 - Correspondencia de las direcciones de transporte en direcciones de red
 - Regular el flujo entre puntos finales
 - Segmentación/Concatenación
 - Detección y recuperación de errores
- Frontera entre los niveles:
 - Niveles 1 a 4 proporcionan servicio al transporte
 - Niveles 5 a 7 proporcionan servicio al usuario. Son usuarios del nivel de transporte
- El nivel de transporte es el corazón de la arquitectura OSI





Servicios proporcionados al nivel de sesión I

- El Nivel de transporte busca proporcionar un servicio eficiente y fiable a sus usuarios, normalmente **procesos de la capa de aplicación**
- Existen dos tipos de servicios:
 - Orientados a conexión
 - No orientados a conexión
- Son muy similares a los del nivel de red. Si son tan parecidos ¿por qué dos niveles distintos?:
 - El nivel de red es una parte de la subred de comunicaciones, sobre la que el usuario no tiene control. Si la subred le ofrece un servicio poco fiable (pierde paquetes) el usuario no tiene forma de resolverlo
 - La solución es poner un nivel por encima que mejore la calidad del servicio
 - Todas las redes no son iguales (unas fallan menos que otras), es necesario, por tanto, aislar a los niveles altos de la tecnología, de los diseños y de las imperfecciones de la subred
 - Si el Servicio de red es impecable ==> nivel de transporte tiene un trabajo fácil
 - Si el Servicio de red es pobre ==> el nivel de transporte hace ver al usuario que la red es buena





Protocolo de Transporte I

- Dependiendo de la calidad de la subred se agrupan en tres categorías:
 - **TIPO A** - Red perfecta
 - La fracción de paquetes perdidos, duplicados o alterados es casi nula
 - El nivel de red no genera "reset"
 - Sólo hace fragmentación y reensamblaje
 - No ofrece servicio de transporte urgente
 - Por ejemplo las LAN
 - **TIPO B**
 - La más común
 - Los paquetes individuales raramente se pierden, los niveles 2 y 3 se encargan de recuperarlos de forma transparente al nivel de transporte.
 - El nivel de red puede emitir algún "reset", entonces el nivel de transporte se encarga de recuperar esos errores, establecer una nueva conexión, resincronizar la comunicación y continuar, ocultándolo al usuario.
 - Ej: X.25
 - **TIPO C**
 - Nada se puede asegurar, los paquetes se pierden, se duplican y hay muchos "reset"
 - El protocolo de transporte debe resolver todo esto, luego será complejo
 - Ej: WAN con servicio sin conexión, IP, redes vía radio, etc.





Protocolo de Transporte IV

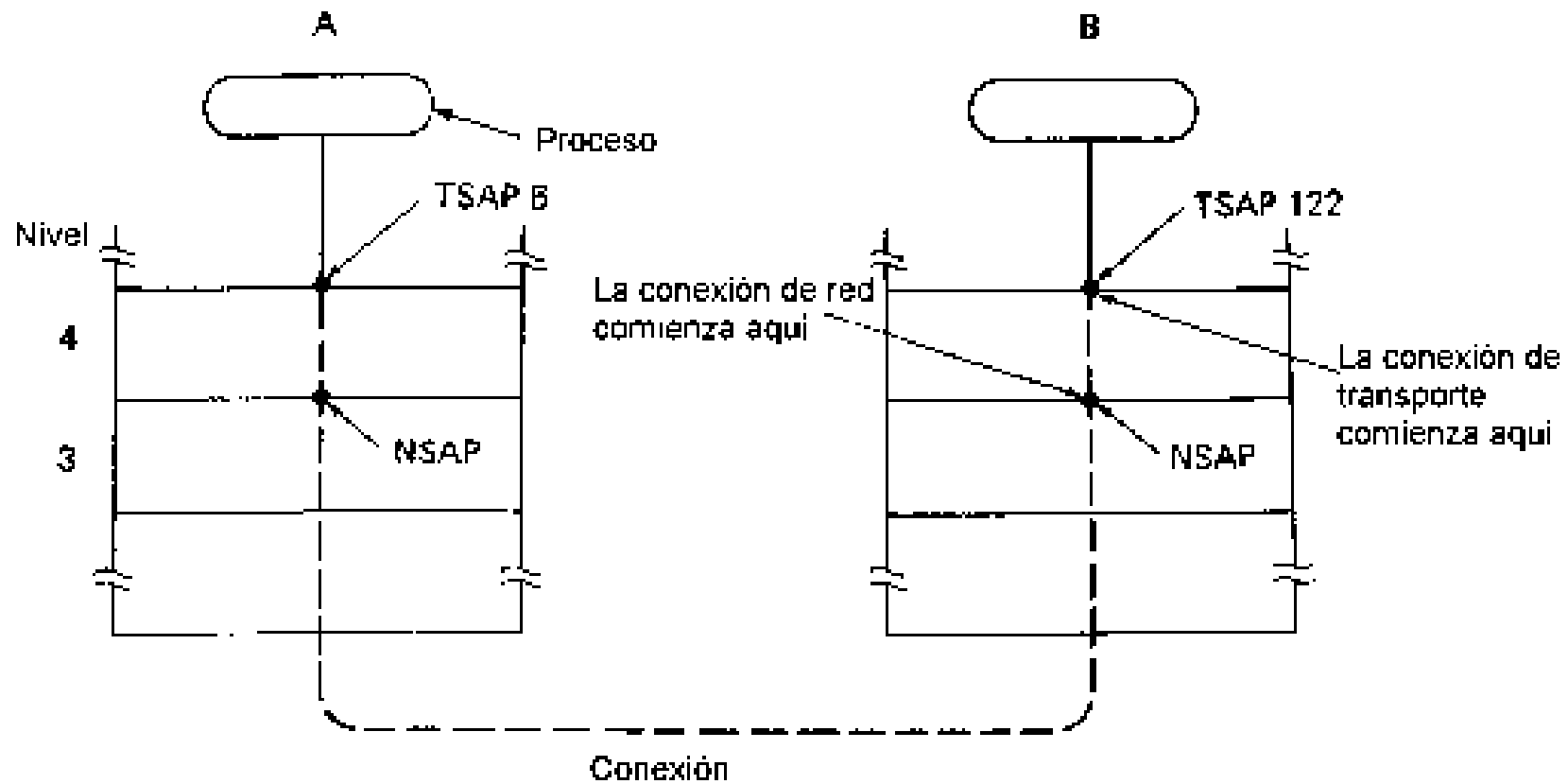
- Semejanzas con el nivel de enlace:
 - Los dos se ocupan del control de errores, secuenciamiento y control de flujo
- Diferencias:
 - Direccionamiento de los extremos
 - Establecimiento de la conexión es más complicado
 - Mensajes emergentes como consecuencia de la capacidad de almacenamiento de la subred
 - Control de flujo más complicado





Elementos de los protocolos de transporte I

- DIRECCIONAMIENTO
 - En Internet nº de puerto





Elementos de los protocolos de transporte II

- Establecimiento de la conexión

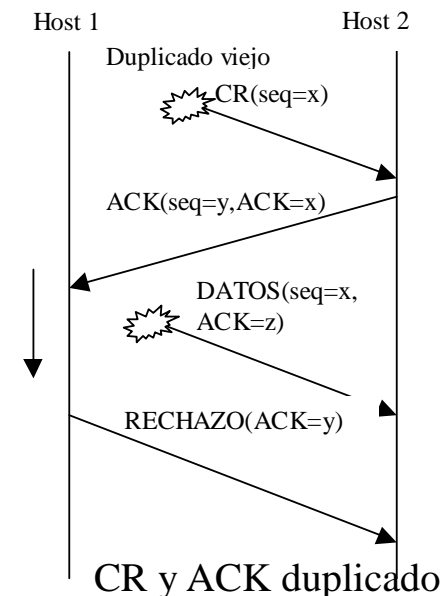
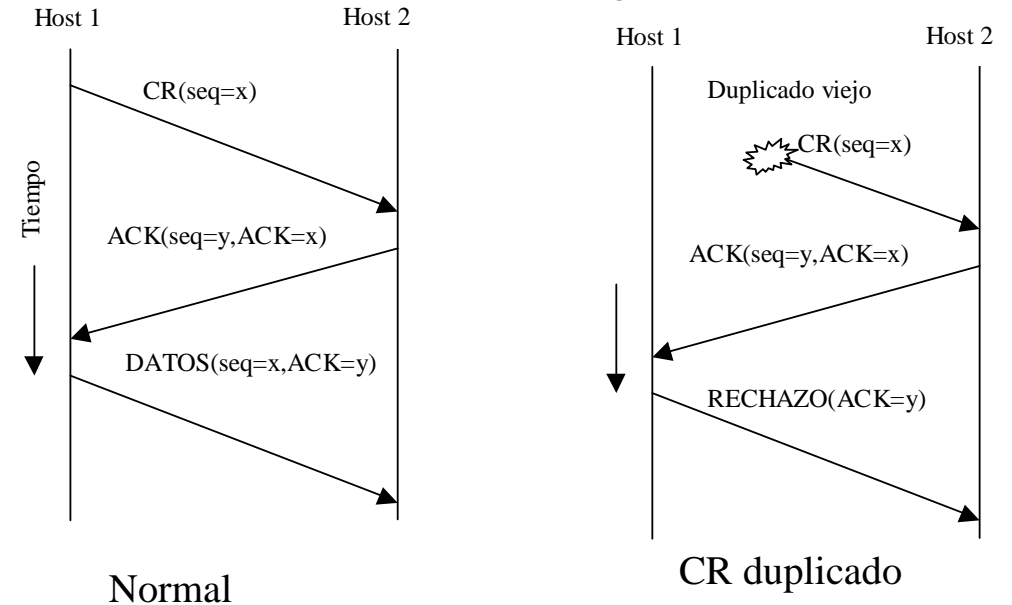
- El problema aparece cuando la red puede perder, almacenar (llegaran más tarde) y duplicar paquetes
- El punto crítico del problema está en los duplicados retardados
- Puede abordarse de varias maneras:
 - Utilizar TSAP desechables
 - Hace imposible el esquema de servidor
 - Con un identificador para cada conexión
 - Problema
 - » Si la máquina falla no podríamos saber que números fueron utilizados previamente
 - Este problema se ve atenuado si utilizamos algún mecanismo para evitar que los paquetes viajen indefinidamente por la subred (limitamos el tiempo de vida). Estos pueden ser:
 - » Colocación de un contador de saltos en cada paquete
 - » Sellado de cada paquete con un *timestamp* (hora en la que fue creado). Requiere que los relojes estén sincronizados (lo cual no es un problema trivial)
- En la práctica no sólo se necesita garantizar que el paquete esté efectivamente muerto, sino que también lo están sus asentimientos para evitar que surjan súbitamente de la nada para complicar la situación
- La idea es asegurar que **no aparecerán dos TPDU's idénticamente numeradas**





Elementos de los protocolos de transporte III

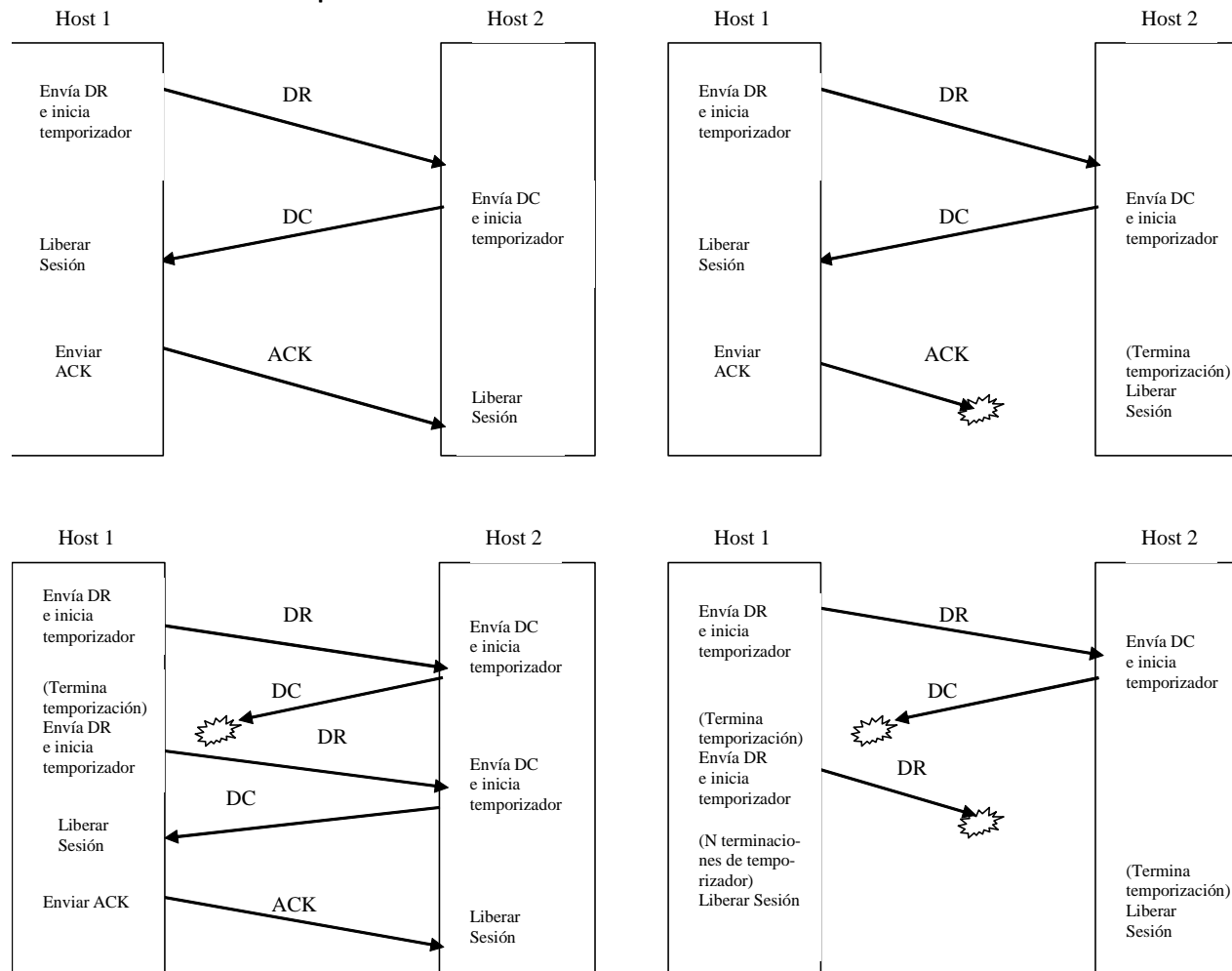
- Tomlinson planteó otra alternativa denominada "ida-vuelta-ida"
 - La petición de conexión también lleva número de secuencia
 - Los interlocutores eligen cada uno un nº de secuencia de forma que el número de secuencia no se repita para conexiones diferentes
 - No requiere que dos interlocutores envíen información con el mismo número de secuencia por lo que no es necesaria la presencia de un reloj común



Elementos de los protocolos de transporte V

- Liberación de la conexión

- Aunque parece un problema sencillo vamos a ver que no es tan trivial. Todas las formas de liberar una conexión tienen una característica en común: son abruptas y pueden llevar a pérdida de datos
- Podemos pensar en una forma de acabar la conexión con acuerdo de ambos extremos. Algo así como yo termine, ¿tu también terminaste?, si el otro responde afirmativamente romper la conexión





Elementos de los protocolos de transporte VI

- **Control de flujo**

- Es parecido en ciertos aspectos al control de flujo en la capa de enlace
- Diferencia fundamental está en que la capa de transporte puede tener muchas conexiones mientras que la capa de enlace tiene pocas, lo que hace impracticable la técnica de almacenamiento temporal de tamaño fijo
- Como las conexiones se abren y cierran continuamente es necesario que emisor y receptor varíen el tamaño de sus buffers dinámicamente
- **Ventana de tamaño variable**
 - Inicialmente el emisor solicita un cierto número de buffers, el receptor otorga tantos como pueda ofrecer
 - Cada vez que el emisor envía una TPDU, deberá decrementar su asignación, deteniéndose por completo cuando alcance el valor 0
 - El receptor superpone de forma separada tanto los asentimientos como las asignaciones de buffers en el tráfico que va en sentido contrario





El nivel de Transporte en Internet I

- Internet ofrece los dos tipos de servicios en el nivel de transporte
 - Con conexión (TCP)
 - Sin conexión (UDP)
- **Direccionamiento** - Números de puerto
 - Los puntos finales de una comunicación son dos procesos de usuario, uno en cada sistema
 - Los dos sistemas pueden estar en redes diferentes, conectadas por uno o más routers
 - Esto requiere tres niveles de direccionabilidad:
 - red
 - host
 - Identificador único para cada comunicación en un host. A este identificador se le denomina **Número de Puerto**





El nivel de Transporte en Internet II

- Números de puerto
 - Como en un host existirán más de una comunicación entre proceso de usuario que use TCP o UDP, necesitamos, algún método que identifique los datos asociados con cada comunicación.
 - Tanto TCP como UDP usan un entero de 16 bits denominado número de puerto para esta identificación
- Los numero de puerto en el rango 1 a 255 están reservados para los **puertos bien conocidos** (RFC 2500)). Cada servicio de Internet tiene reservado un número de puerto
 - Son asignados por la organización llamada *Internet Assigned Number Authority*
 - Algunos sistema operativos reservan puertos adicionales para programas privilegiados (4.3 BSD reserva los puertos 1-1023 para procesos de superusuario)
 - El resto son para puertos del usuario que se denominan **puertos efímeros**
 - Por ejemplo: FTP es un servicio bien conocido en el puerto 21



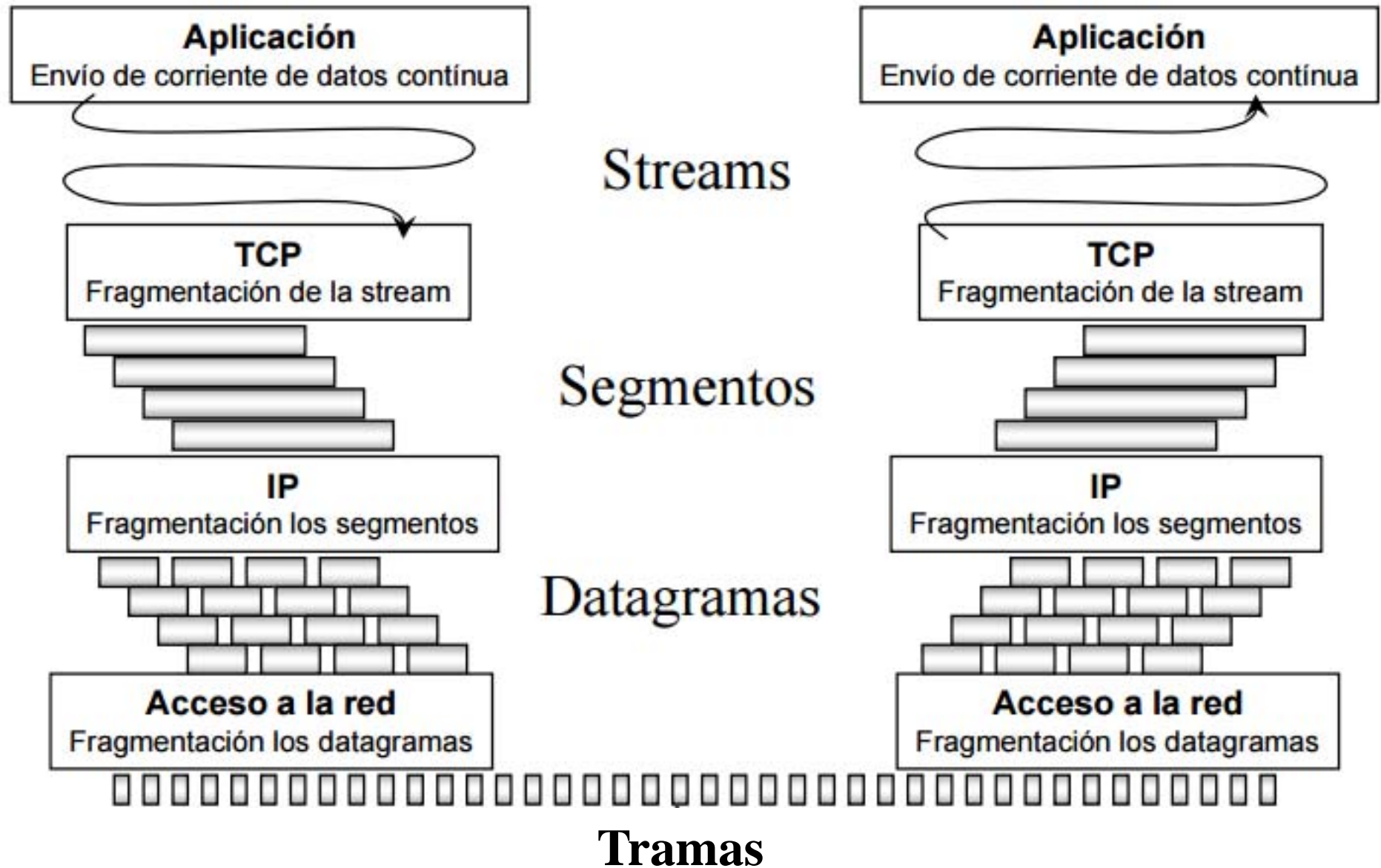


El nivel de Transporte: TCP

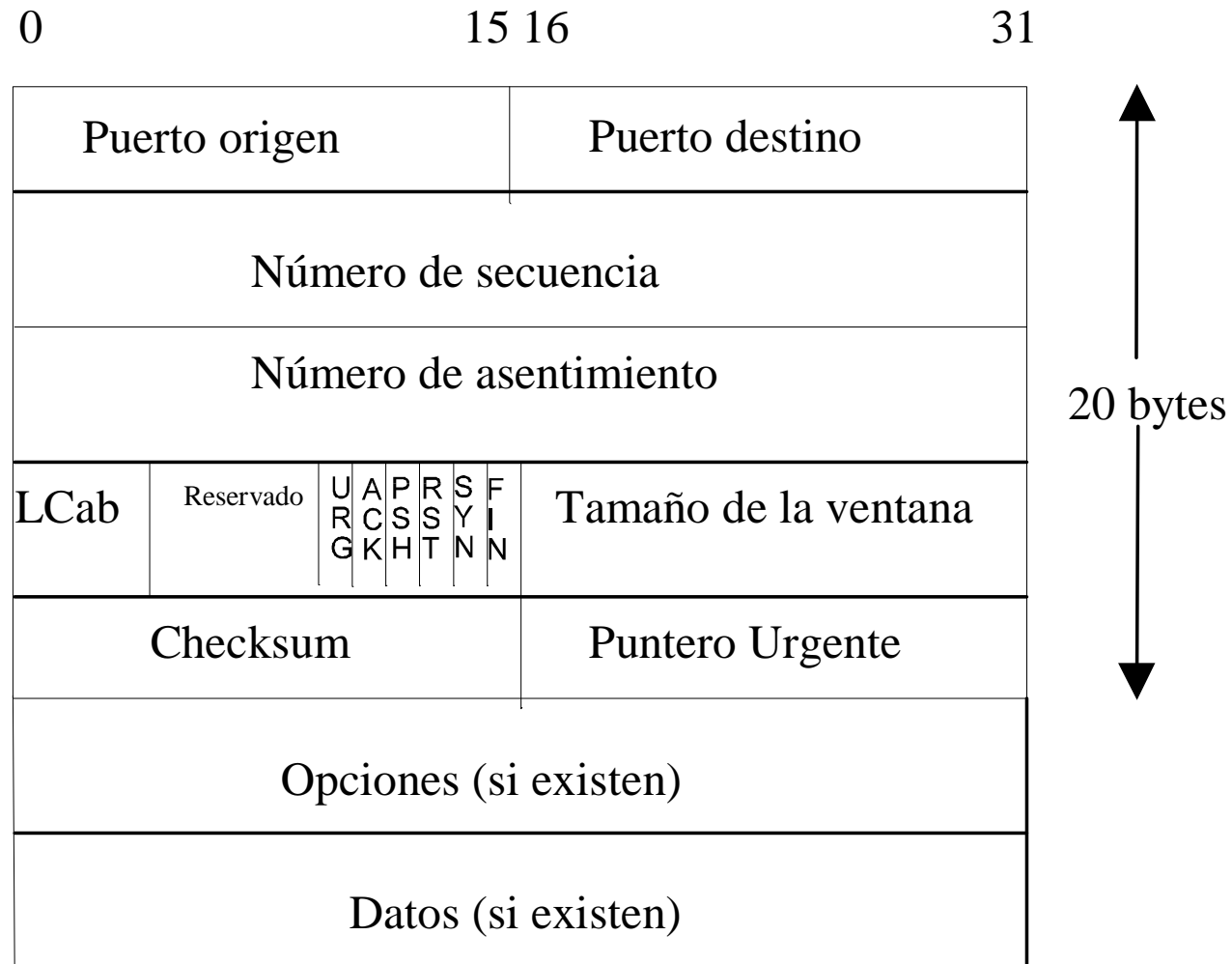
- TCP - Transmisión control protocol
 - Es un protocolo orientado a conexión y fiable
 - TSAP = Conexión (Dirección IP origen, nº puerto), (Dirección IP destino, nº puerto)
 - Un puerto TCP se pueden compartir entre varias conexiones
 - Es responsable de
 - Establecer, mantener y terminar una conexión entre procesos
 - Permitir múltiples conexiones entre procesos distintos en un mismo host
 - Fraccionar el mensaje en datagramas, reensamblarlo en destino (ordenando los datagramas si no han llegado en orden)
 - Garantizar que la transmisión sea segura extremo a extremo (checksum, asentimientos, retransmisiones, timeouts,...)
 - Proporcionar un control de flujo extremo a extremo utilizando la técnica de ventana deslizante
 - Como vemos TCP se diseñó para tolerar una red insegura (de tipo C, en términos del modelo OSI)
 - En la RFC 793 se describe el estándar TCP



TCP: Unidad de datos



La cabecera TCP I





La cabecera TCP II

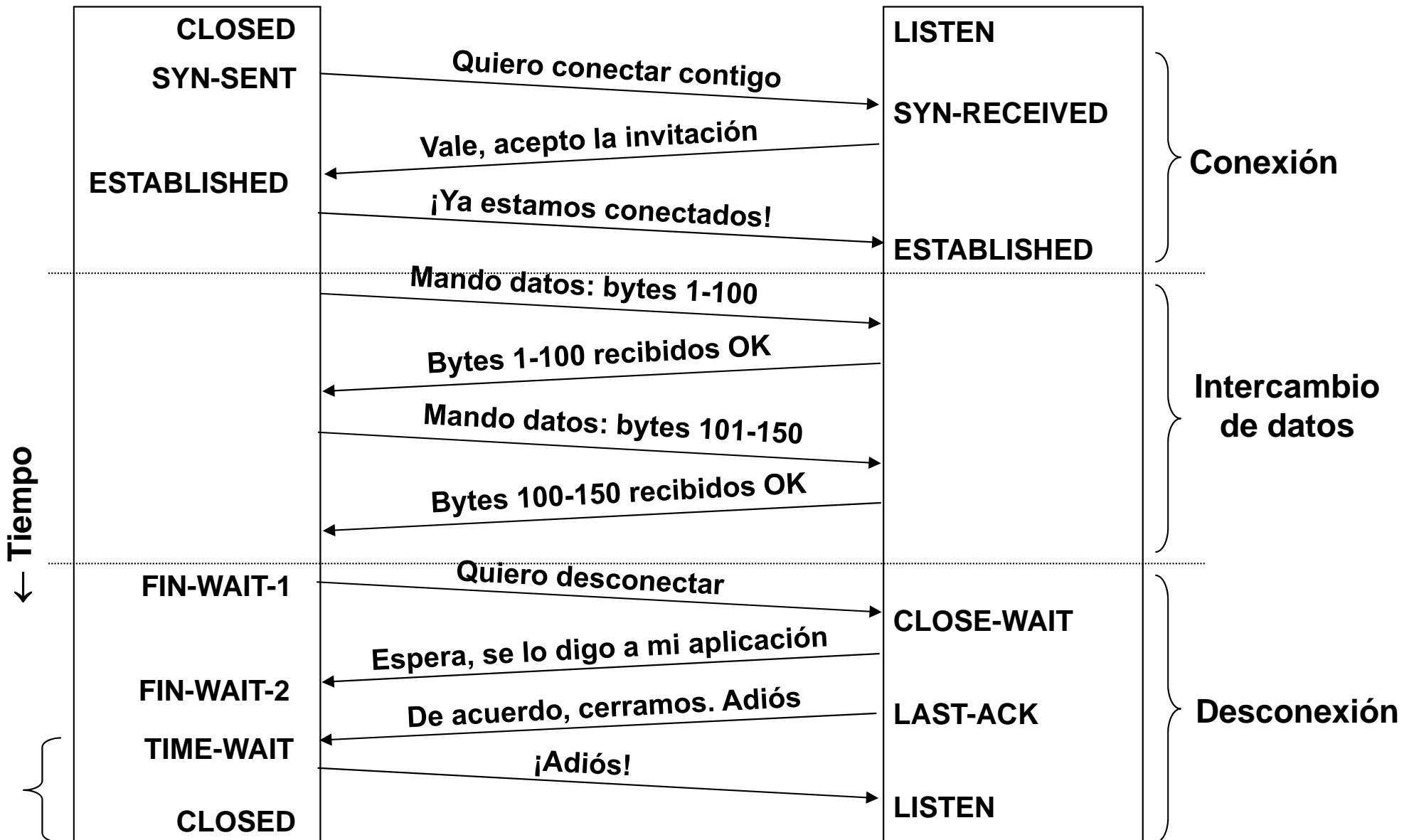
- Puerto origen / Puerto destino
- Número de secuencia: Identificación del fragmento del mensaje
- Número de reconocimiento: asentimiento a un fragmento recibido
- Longitud de la cabecera
- Bits de control:
 - URG: Indica urgencia
 - ACK: Asentimiento
 - RST: Rechazo de conexión o segmento no válido
 - PSH: Fin de mensaje
 - SYN: Establecimiento de conexiones
 - FIN: Fin de la conexión
- Ventana: Tamaño de la ventana usada para el reconocimiento
- Checksum: Detección de errores
- Puntero de urgente: Sólo se interpreta si el bit URG está activo
- Opciones: Campo de longitud variable.
 - Ej. Tamaño máximo del segmento (Mínimo 536 bytes => Paquete estándar de IP de 576 bytes), uso de la RFC 1106 para rechazo selectivo (en lugar de vuelta atrás N) utilizando NAK, final de la lista de opciones, no operación.



Un diálogo TCP típico

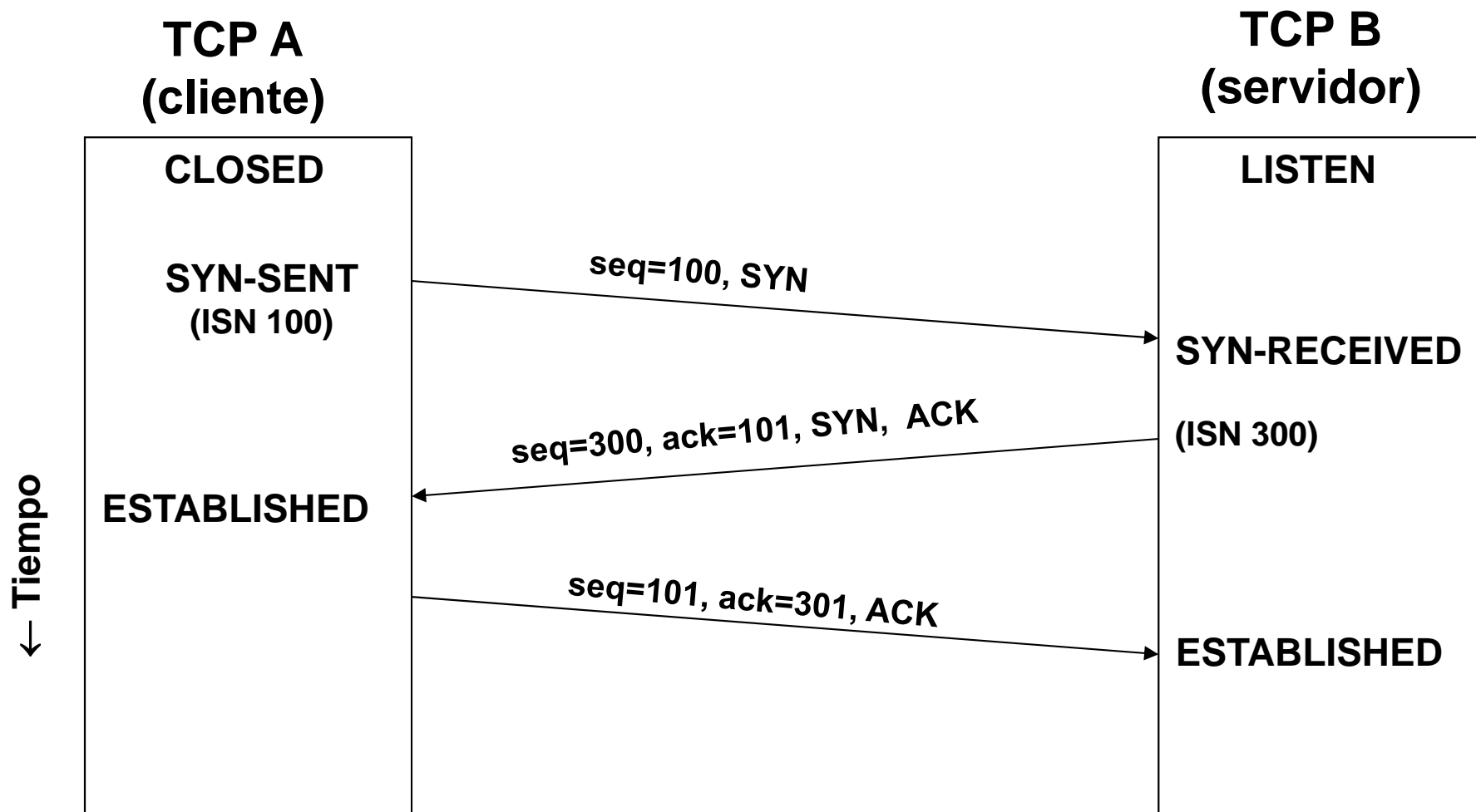
TCP Cliente

TCP Servidor



Establecimiento de una conexión TCP

- Tomlinson. “Ida-vuelta-ida”
 - Evitar el problema de los duplicados retardados



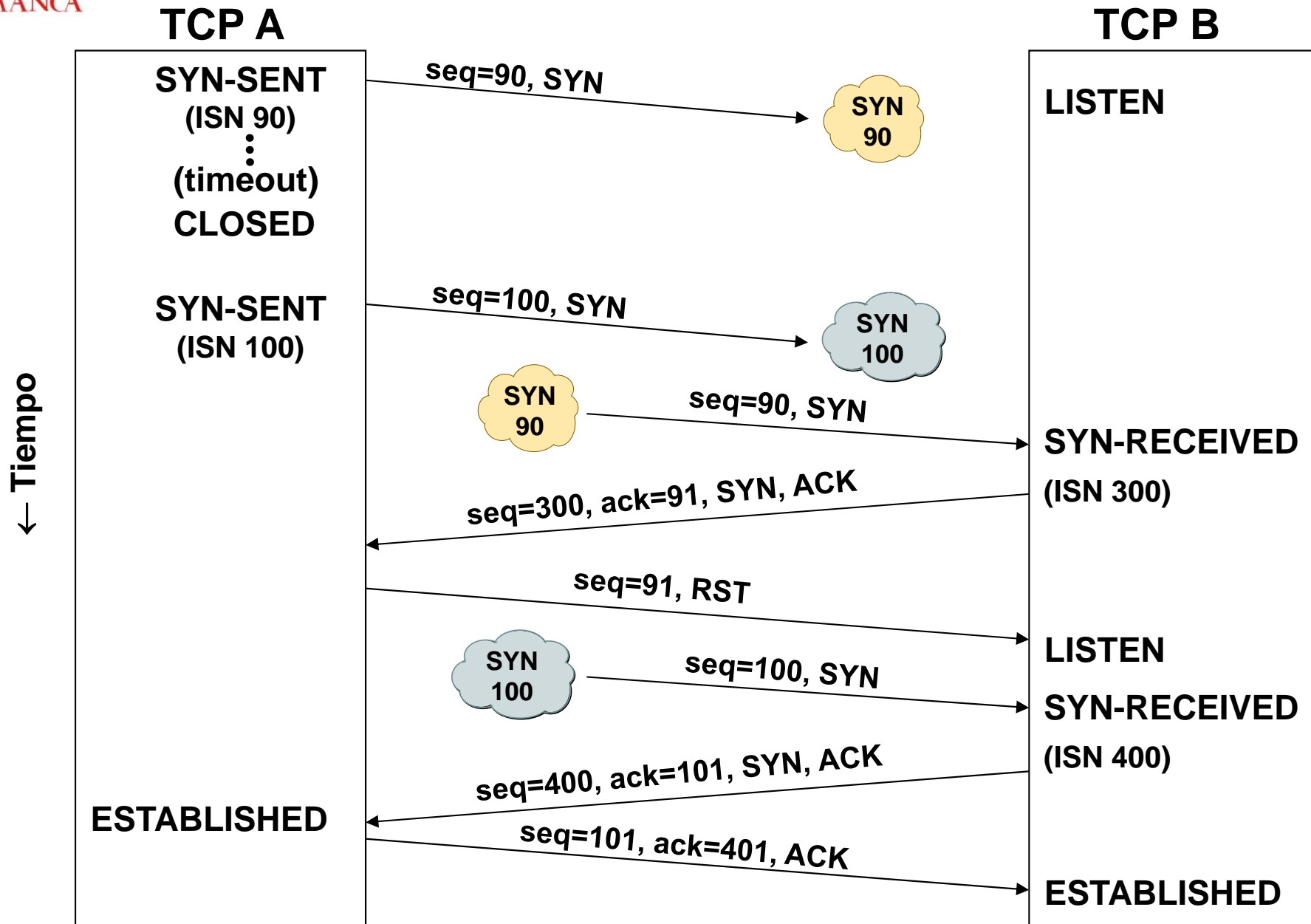
ISN, *Initial Sequence Number*





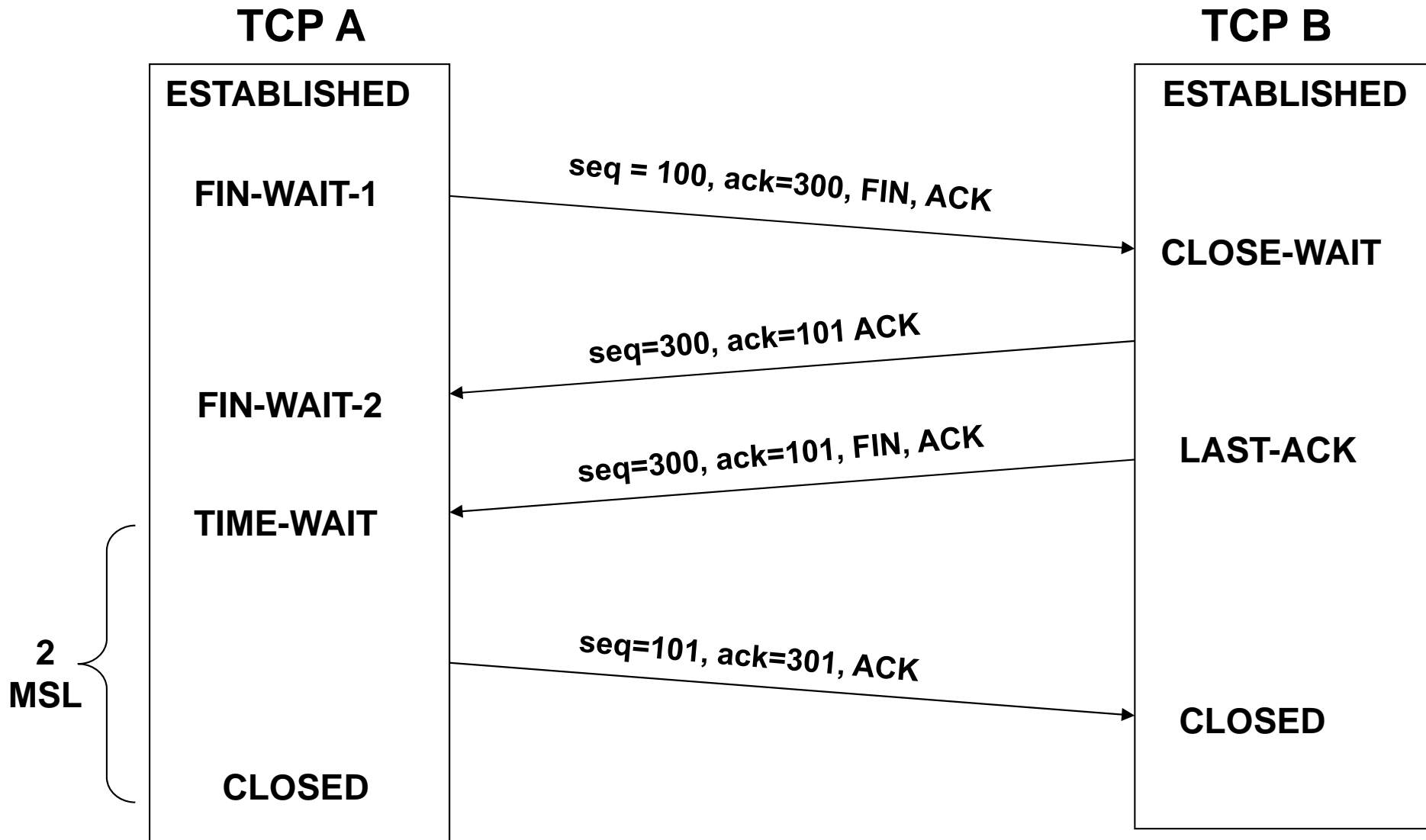
Conexión con SYN duplicado

ISN, *Initial Sequence Number*





Desconexión a tres vías, caso normal

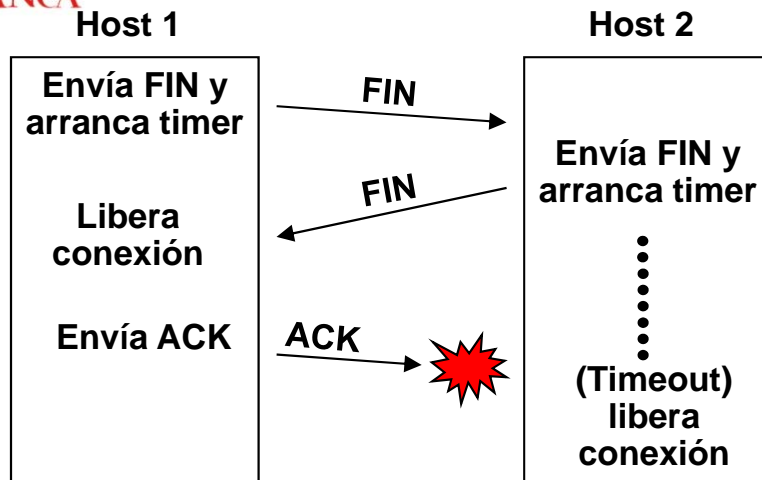


MSL: *Maximum Segment Lifetime* (normalmente 2 minutos)

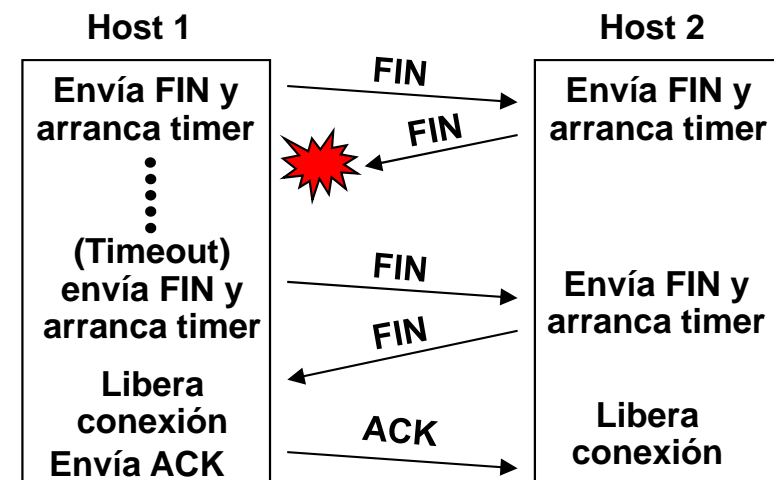
En olivo su valor se puede consultar con la orden: `ndd -get /dev/tcp tcp_time_wait_interval`
60000 (60 segundos)



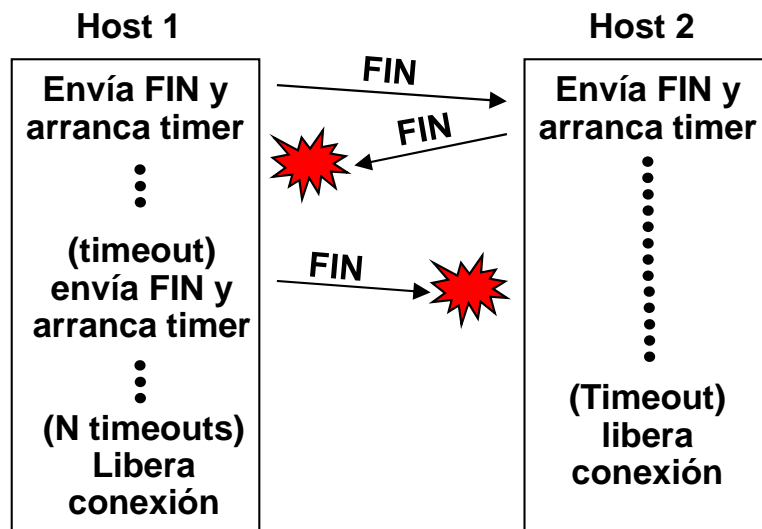
Desconexión a tres vías, casos anormales



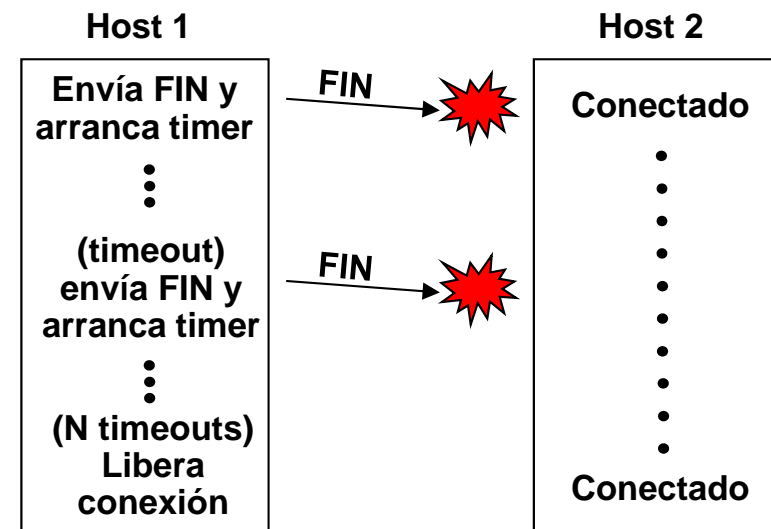
Pérdida de ACK final



Pérdida de respuesta FIN

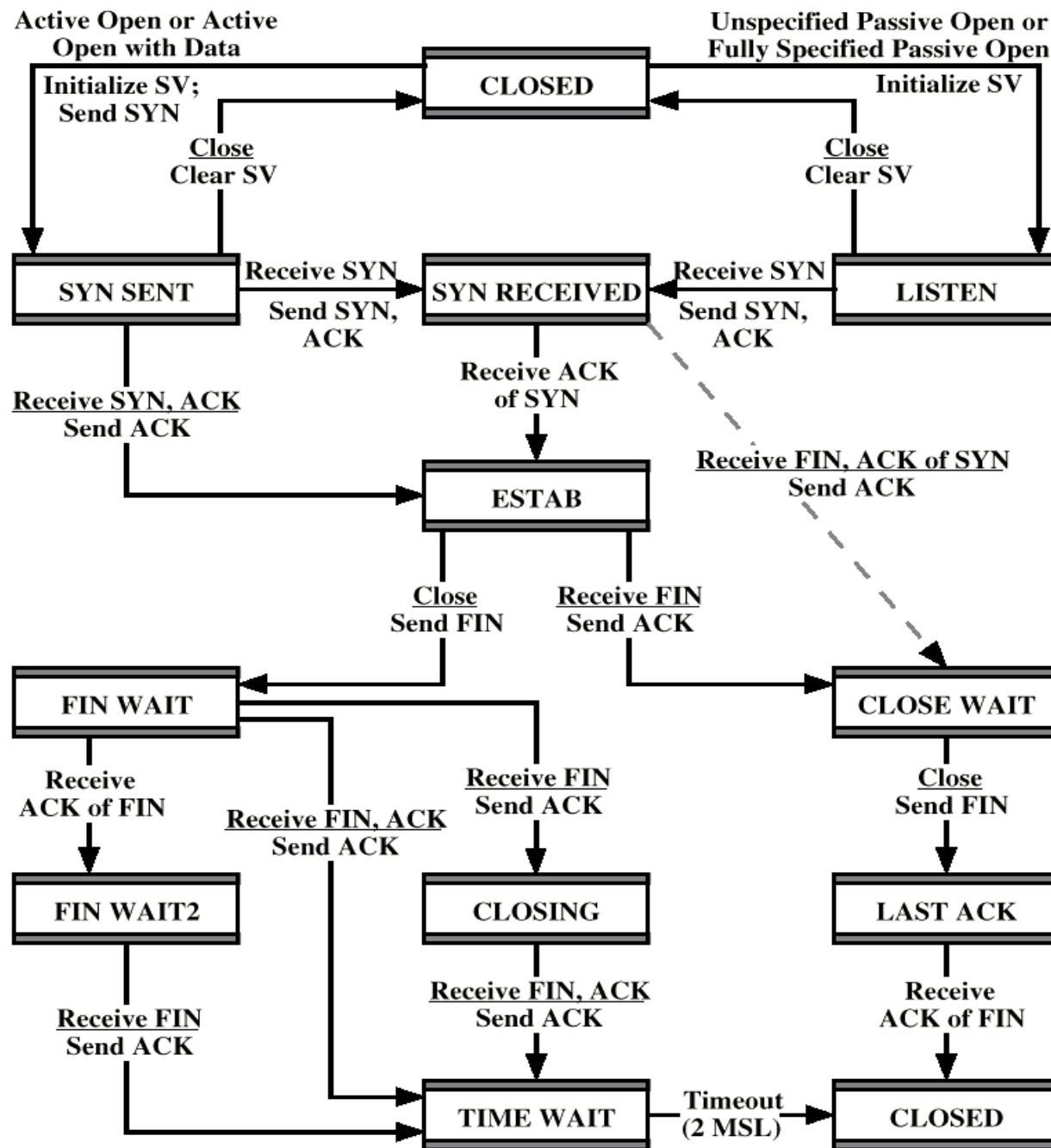


Pérdida de todo menos primer FIN



Pérdida de todos los FIN de host 1





SV = state vector
MSL = maximum segment lifetime





El nivel de Transporte: UDP

- UDP: "User Datagram Protocol"
 - Proporciona un servicio de transporte no fiable, sin conexión
 - TSAP = (Dirección IP, nº Puerto)
 - No proporciona asentimiento de los datagramas recibidos al receptor. Al igual que IP los paquetes pueden llegar desordenados, perderse o duplicarse
 - La única facilidad que incorpora UDP es la asignación y gestión de los números de puerto para identificar aplicaciones individuales que se ejecutan en un host
 - UDP es más rápido que TCP debido a la sobrecarga en las funciones que realiza TCP
 - Se describe en la RFC 768

Puerto origen	Puerto destino
Longitud	Checksum
Datos	