William Stallings
Comunicaciones y Redes de Computadores

Capítulo 17

Protocolos de transporte

Mecanismos del protocolo de la capa de transporte orientado a conexión

- Conexión lógica.
- Establecimiento.
- Mantenimiento y cierre.
- Seguridad.
- Ejemplo: TCP

Servicio de red de secuenciamiento seguro

- Acepta mensajes con un tamaño arbitrario.
- Envía los mensajes en secuencia al destino con una seguridad virtual del 100%:
  - Ejemplo: una red de conmutación de paquetes altamente segura con una interfaz X.25.
  - Ejemplo: una red de retransmisión de tramas (*frame relay*) utilizando el protocolo de control LAPF.
  - Ejemplo: una LAN IEEE 802.3 utilizando el servicio LLC orientado a conexión.
- El protocolo de transporte se utiliza como un protocolo extremo-a-extremo entre dos sistemas finales conectados en la misma red.

Elementos de un protocolo de transporte sencillo

- Direccionamiento.
- Multiplexación.
- Control de flujo.
- Establecimiento/cierre de la conexión.

Direccionamiento

- El usuario debe ser especificado mediante:
  - Identificación de usuario:
    - Normalmente, la dirección de usuario se especifica como *estación* o *puerto.*
      - En TCP, la combinación del puerto y la estación se refiere como un conector.
    - La variable puerto representa un usuario TS particular en la estación especificada.
  - Identificación de la entidad de transporte:
    - En general, habrá una sola entidad de transporte en cada estación.
    - Si están presentes más de una entidad de transporte, normalmente, hay una de cada tipo:
      - Indicación del tipo de protocolo de transporte (por ejemplo, TCP, UDP).
  - Dirección de la estación:
    - En el caso de una red única, estación identifica un dispositivo de red conectado a la misma.
    - En un conjunto de redes, estación es una dirección internet global.
  - Número de red.

Cómo encontrar direcciones

- Existen cuatro métodos:
  - El usuario debe conocer la dirección si desea un tiempo de respuesta más rápido.
    - Ejemplo: un proceso que recoge estadísticas sobre prestaciones.
  - Se asignan direcciones bien conocidas.
  - Proporciona un servidor de nombres.
  - Se envía una petición a un proceso a una dirección bien conocida.

Multiplexación

- Usuarios múltiples emplean el mismo protocolo de transporte.
- Los usuarios se distinguen unos de otros por números de puerto o puntos de acceso al servicio.
- También podría implementar una función de multiplexación con respecto a los servicios de red usados:
  - Ejemplo: multiplexación de un circuito virtual X.25 a un número de usuarios del servicio de transporte:
    - X.25 consume el tiempo de conexión en cada circuito virtual.

Control de flujo

- El retardo de transmisión entre entidades de transporte es grande comparado con el tiempo de transmisión real:
  - Retardo en la comunicación de la información de control de flujo.
- Variabilidad en la cantidad de retardo en la transmisión:
  - Dificultad para utilizar el mecanismo de temporizadores.
- El flujo puede ser controlado por dos razones:
  - El usuario de la entidad receptora no puede mantener el flujo de datos.
  - La entidad de transporte receptora no puede mantener el flujo de segmentos.
- Esto dará como resultado la saturación de las memorias temporales.

Formas de hacer frente al requisito de control de flujo

- No hacer nada:
  - Los segmentos que llegan se descartan.
  - La entidad de transporte emisora no recibirá confirmación y retransmitirá los segmentos:
    - El emisor incrementa sus envíos para incluir los segmentos nuevos además de los originalmente retransmitidos.
- Rechazar la aceptación de más segmentos del servicio de red:
  - Mecanismo poco riguroso.
  - En conexiones multiplexadas, el control se ejerce en el agregado de todas las conexiones de transporte.

Formas de hacer frente al requisito de control de flujo

- Usar un protocolo fijo de ventana deslizante:
  - Véase el Capítulo 7 para conocer más detalles sobre su funcionamiento.
  - Funciona bien con un servicio de red seguro:
    - La falta de confirmación se interpreta como una táctica de control de flujo.
  - No funciona bien en una red no segura:
    - No distingue entre el control de flujo y la pérdida de un segmento.
- Usar un esquema de créditos.

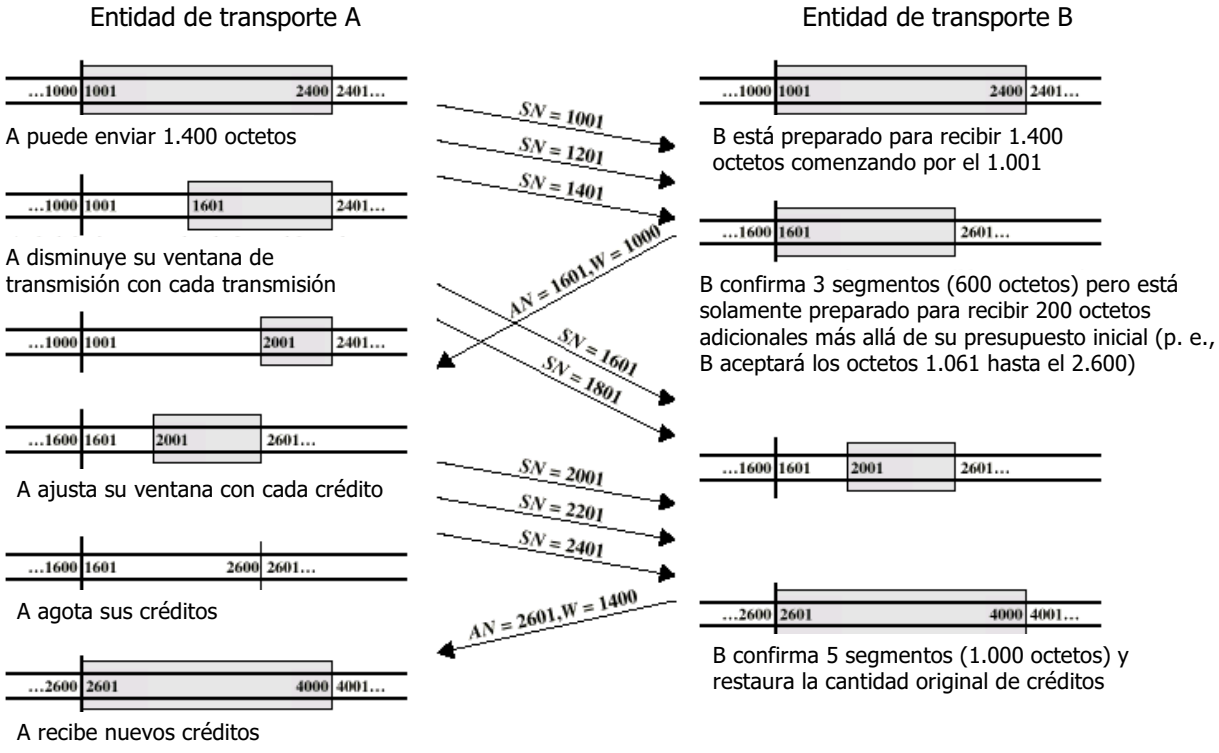
Esquema de créditos

- Mayor grado de control con un servicio de red seguro.
- Más efectivo con un servicio de red no seguro.
- Desliga las confirmaciones del control de flujo:
  - Se puede confirmar un segmento sin obtener un nuevo crédito, y viceversa.
- Cada octeto tiene un número de secuencia.
- Cada segmento transmitido incluye en su cabecera un número de secuencia, un número de confirmación y la ventana.

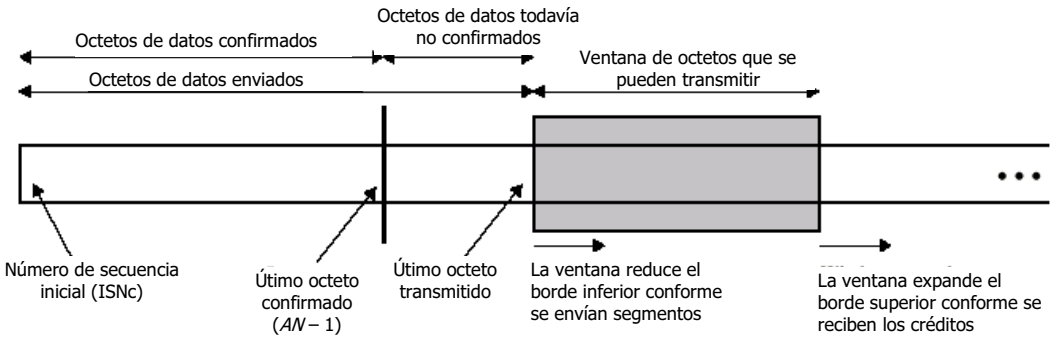
Uso de segmentos de cabecera

- Cuando una entidad de transporte envía un segmento, incluye el número de secuencia del primer octeto en el segmento.
- La confirmación incluye AN=*i*, W=*j*.
- Se confirman todos los octetos hasta SN= *i* -1:
  - El siguiente octeto esperado es *i.*
- Se concede permiso para enviar una ventana adicional de W= *j* octetos de datos:
  - Es decir: los *j* octetos correspondientes a los números de secuencia *i* hasta *i*+*j* – 1.

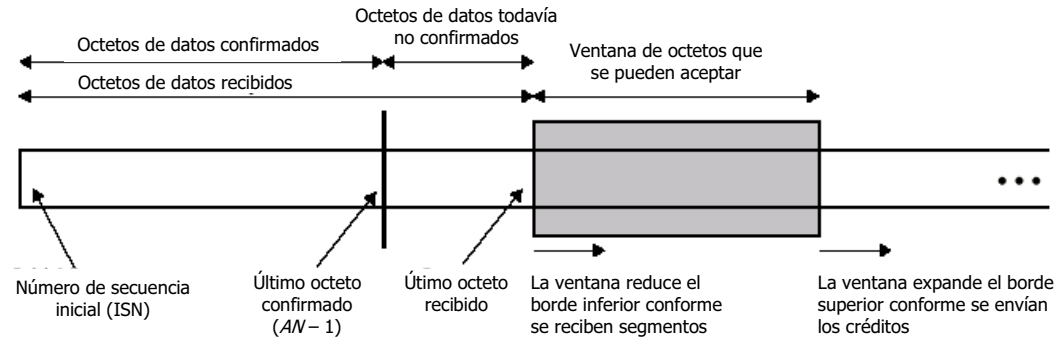
Asignación de créditos



Perspectivas del control de flujo emisor y receptor



(a) Espacio de la secuencia de emisión

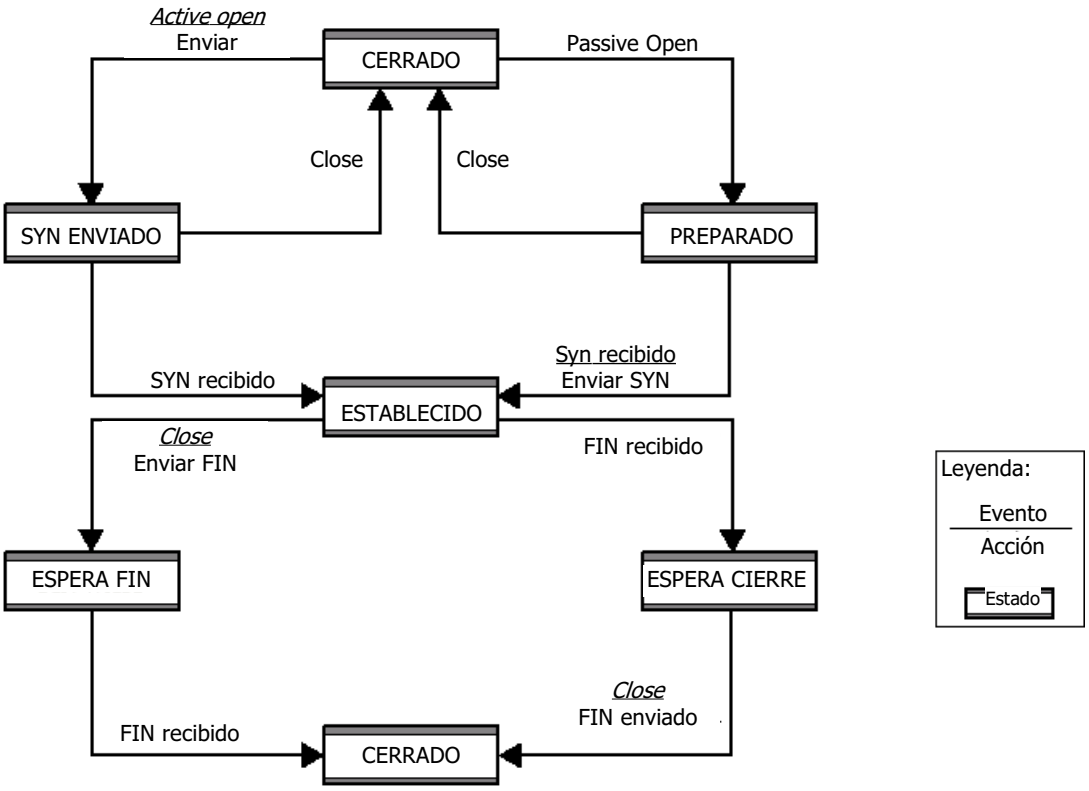


(b) Espacio de la secuencia de recepción

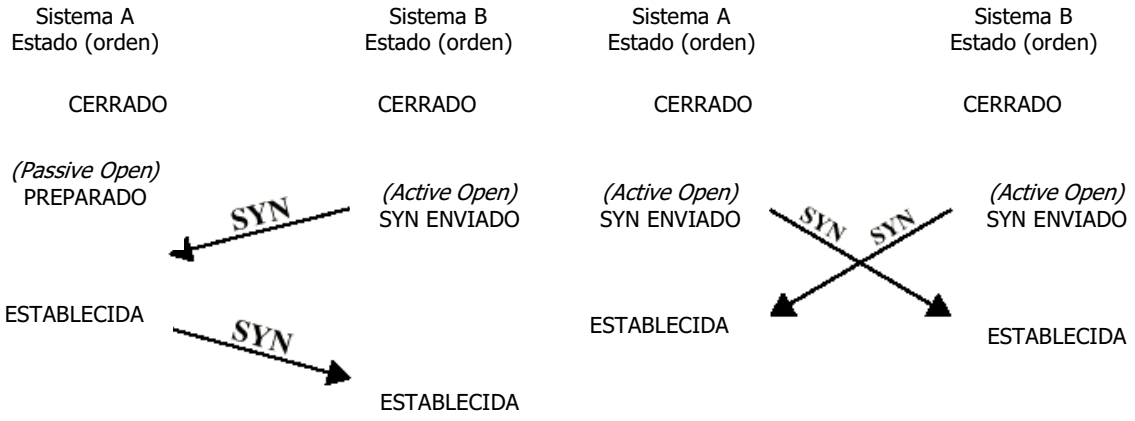
Establecimiento y cierre de la conexión

- Permite a cada extremo asegurarse de que el otro existe.
- Negociación de parámetros opcionales.
- Pone en marcha la reserva de recursos de la entidad de transporte.
- Establecimiento de la conexión por mutuo acuerdo.

Diagrama de estados de una conexión sencilla



Establecimiento de conexión



(a) *Active/Passive Open*

(b) *Active/Active Open*

No atender a llamadas

- Se rechaza la petición enviando un segmento RST (reiniciar).
- La petición se puede poner en cola hasta que el usuario TS emita una orden *Open*.
- La entidad de transporte informa al usuario de la existencia de una petición pendiente:
  - Una orden *Passive Open* se puede reemplazar por una orden *Accept*.

Cierre de la conexión

- Cualquier lado, o ambos, pueden iniciar el cierre.
- Por mutuo acuerdo.
- Permite un cierre abrupto.
- O un cierre ordenado:
  - Una conexión que está en el estado ESPERA CIERRE debe continuar aceptando segmentos de datos hasta que se reciba un segmento FIN.

Inicio del proceso de cierre

- En respuesta a una primitiva *Close* del usuario TS.
- Se envía un segmento FIN al otro extremo de la conexión, solicitando el cierre.
- La conexión se sitúa en el estado ESPERA FIN:
  - Continúa aceptando datos y los pasa a su usuario.
  - No envía más datos.
- Cuando se recibe como respuesta un FIN, la entidad de transporte informa a su usuario y cierra la conexión.

No inicio del cierre

- Se recibe un segmento FIN.
- La entidad de transporte informa a su usuario de la petición de cierre y sitúa la conexión en el estado ESPERA CIERRA:
  - Continúa aceptando datos de su usuario y los transmite en segmentos de datos al otro extremo.
- El usuario emite una primitiva *Close*.
- La entidad de transporte envía un segmento FIN.
- Cierre de la conexión.

- Ambos extremos reciben todos los datos pendientes.
- Ambos están de acuerdo en terminar la conexión antes del cierre real.

Servicio de red no seguro

- Ejemplos:
  - Una interconexión de redes utilizando IP.
  - Una red *frame relay* utilizando solamente el núcleo del protocolo LAPF.
  - Una LAN IEEE 802.3 usando un servicio no orientado a conexión LLC sin confirmaciones.
- Los segmentos pueden perderse ocasionalmente.
- Los segmentos pueden llegar fuera de secuencia debido al retardo de tránsito variable.

Problemas

- Transporte en orden.
- Estrategia de retransmisión.
- Detección de duplicados.
- Control de flujo.
- Establecimiento de la conexión.
- Cierre de la conexión.
- Recuperación de las caídas.

Transporte en orden

- Los segmentos pueden llegar de forma desordenada.
- La solución es numerar los segmentos secuencialmente.
- TCP numera cada octeto de datos secuencialmente.
- Cada octeto de datos que se transmite está implícitamente numerado.

Estrategia de retransmisión

- El segmento puede dañarse en el camino.
- El segmento no llega a su destino.
- El transmisor no sabe que la transmisión del segmento se realizó sin éxito.
- El receptor debe confirmar cada recepción de un segmento con éxito.
- Se utiliza una confirmación acumulativa.
- El tiempo que se tarda en confirmar los segmentos favorece la retransmisión.

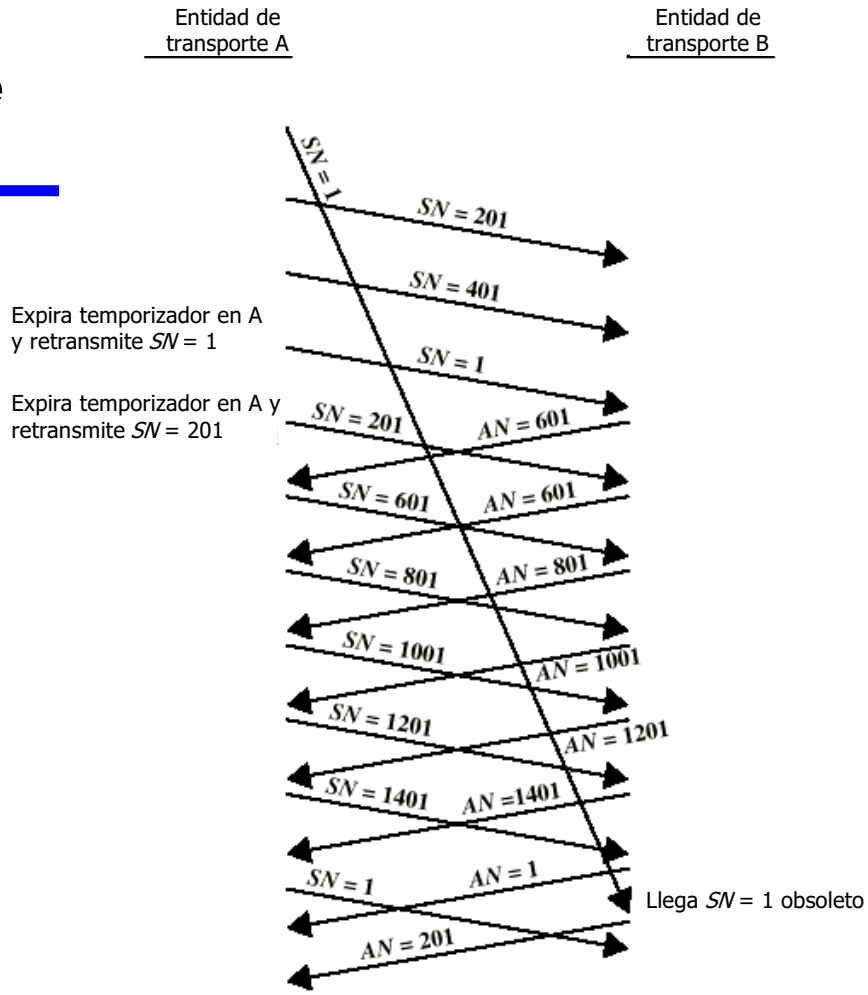
Valor del temporizador

- Temporizador con valor fijo:
  - Basado en el conocimiento del comportamiento típico de la red.
  - No sabe adaptarse a las condiciones cambiantes de la red.
  - Si el valor es muy pequeño, habrá muchas retransmisiones innecesarias.
  - Si el valor es muy grande, el protocolo será muy lento en dar respuesta a la pérdida de un segmento.
  - El temporizador se debe fijar a un valor un poco mayor que el retardo de ida y vuelta.
- Esquema adaptativo:
  - Puede que no confirme inmediatamente un segmento.
  - No puede distinguir entre la confirmación recibida del segmento inicial y la retransmisión del mismo.
  - Las condiciones de la red pueden cambiar rápidamente.

Detección de duplicados

- Si se pierde un ACK, se retransmitirá el segmento.
- El receptor debe reconocer los duplicados.
- El duplicado se recibe antes del cierre de la conexión:
  - El receptor asume que la confirmación se perdió y debe confirmar el duplicado.
  - El emisor no debe confundirse si recibe múltiples ACK.
  - El espacio de números de secuencia debe ser lo suficientemente grande como para no agotarse en menos tiempo que la vida máxima posible de un segmento.
- El duplicado se recibe después del cierre de conexión.

Detección incorrecta de duplicados



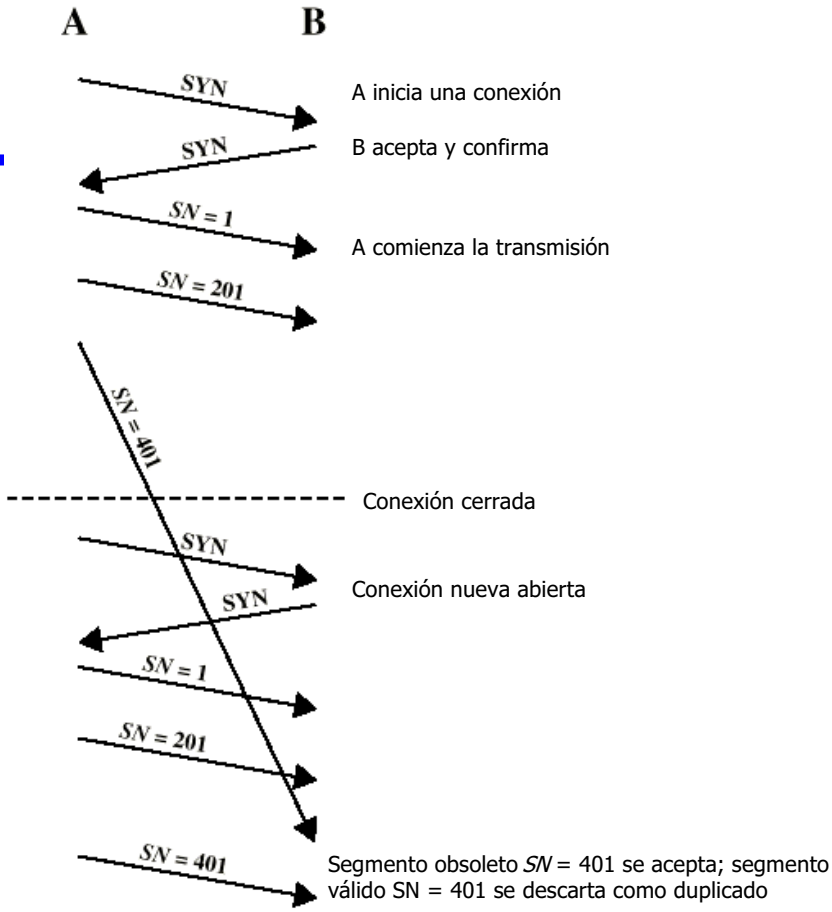
Control de flujo

- Asignación de créditos.
- Considere la situación *AN*=*i*, *W*=0 cerrando temporalmente la ventana.
- Se envía *AN*=*i*, *W*=j para reabrir la ventana, pero este segmento se pierde.
- El emisor piensa que la ventana está cerrada, mientras que el receptor piensa que está abierta.
- Para resolver este problema se puede utilizar un Temporizador de ventana.
- Si el temporizador expira, se le requiere a la entidad de transporte que envíe un segmento:
  - Puede retransmitir un segmento anterior.

Establecimiento de la conexión

- Diálogo en dos sentidos:
  - A emite un SYN, B responde con SYN.
  - Un SYN perdido puede tratarse mediante retransmisión:
    - Esto puede ocasionar la aparición de SYN duplicados.
  - Ignorar los SYN duplicados una vez que la conexión se haya establecido.
- Los segmentos de datos perdidos o retrasados pueden producir problemas en el establecimiento de conexión:
  - Segmentos de conexiones antiguas.
  - Empezar cada nueva conexión con un número de secuencia diferente, elegido lejos del último número de secuencia de la conexión más frecuente:
    - Petición de la conexión es de la forma SYN *i.*
    - Necesita confirmación para incluir *i.*
    - Diálogo en tres direcciones.

Diálogo en dos sentidos: segmento de datos obsoleto



El diálogo en dos sentidos: segmentos SYN obsoletos

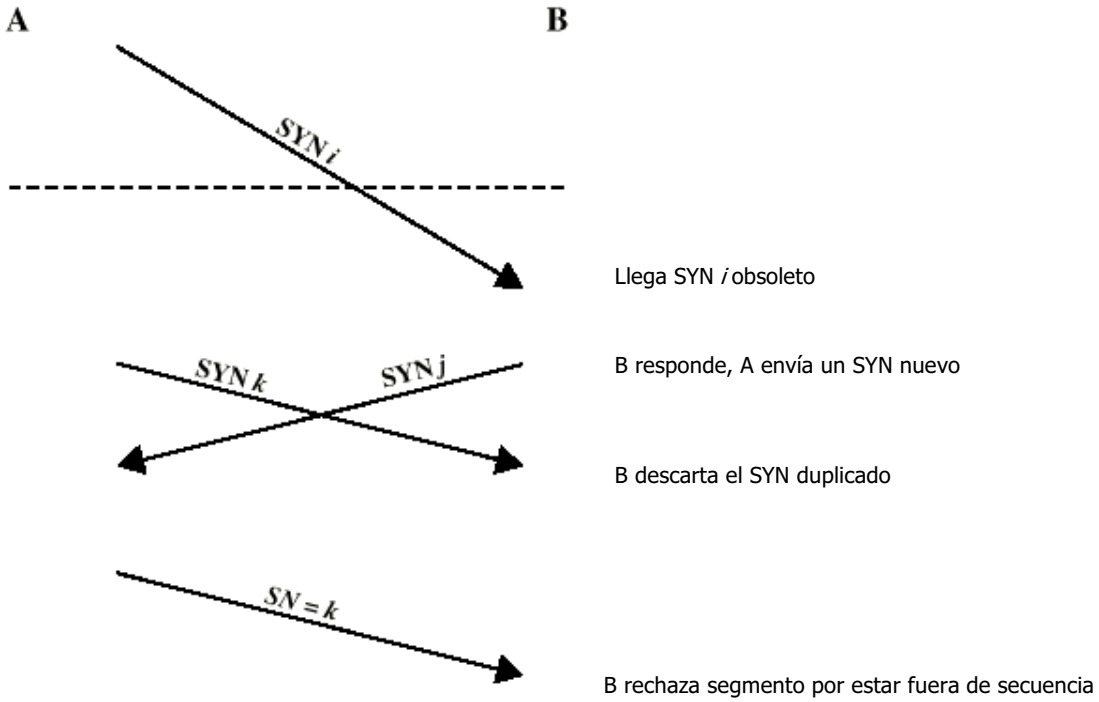
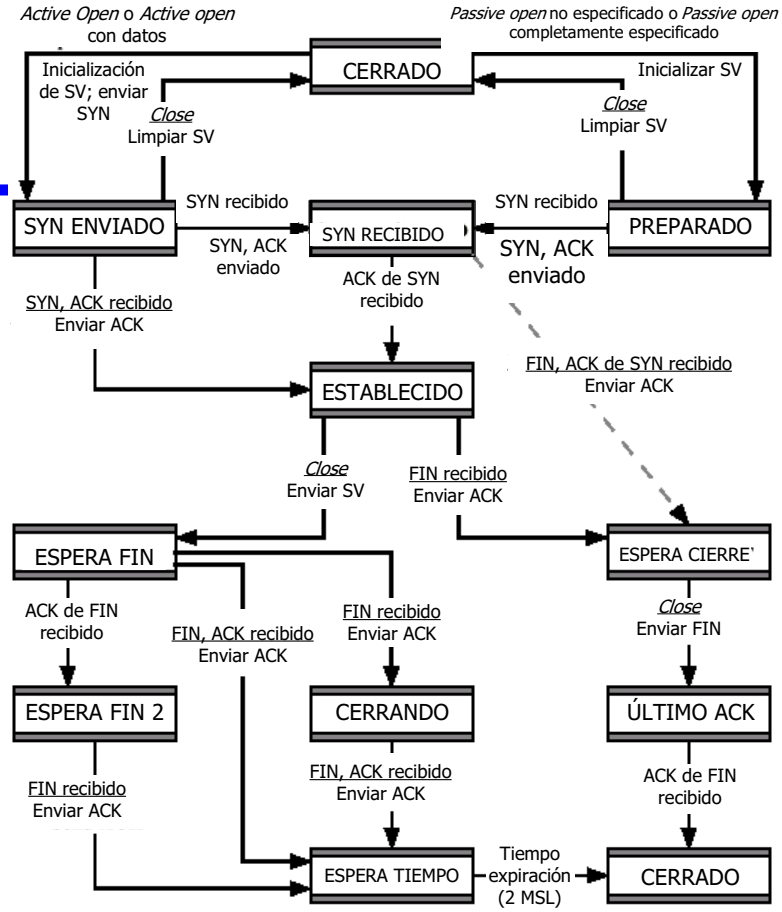
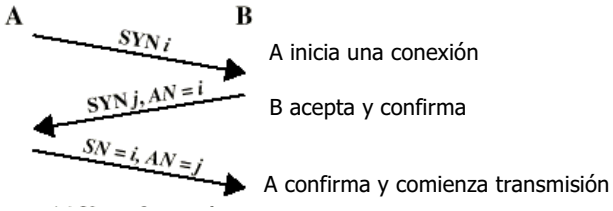


Diagrama de estado de una entidad TCP

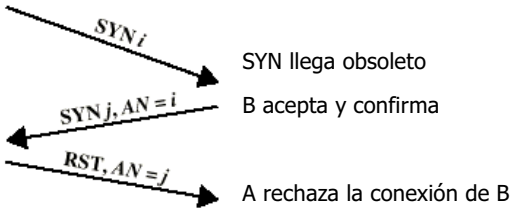


SV = Vector de estado
MSL = Tiempo de vida máximo de segmento

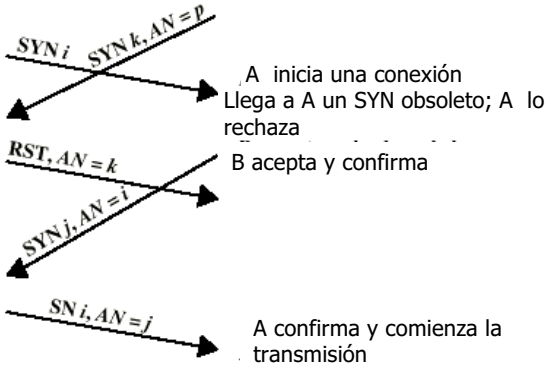
Ejemplo de diálogo en tres sentidos



(a) Operación normal



(b) SYN retrasado



(c) SYN, ACK retrasado

Cierre de la conexión

- Una entidad de transporte en el estado ESPERA CIERRE envía su último segmento de datos, seguido por un segmento FIN.
- El segmento FIN llega al otro extremo antes que su último segmento de datos.
- Cuando el receptor acepte el segmento FIN:
  - Se cerrará la conexión.
  - Se perderá al último segmento de datos.
- Asociar un número de secuencia con FIN.
- El receptor espera a todos los segmentos antes de cerrar la conexión.
- Pérdida de segmentos y presencia de segmentos obsoletos:
  - Debe confirmar el segmento FIN.

Cierre ordenado

- Enviar un FIN *i* y recibir un *AN* = *i*.
- Recibir un FIN *j* y enviar un *AN* = *j*.
- Esperar un intervalo igual a dos veces el máximo del tiempo de vida esperado de un segmento.

Recuperación de caídas

- Después de rearrancar un sistema, toda la información de estado se pierde.
- Las conexiones están medio abiertas:
  - El lado que no se vio afectado por la caída todavía cree que sigue conectado.
- Cerrar la conexión mediante un temporizador de renuncia:
  - Esperar una confirmación. Segmento retransmitido el máximo número de veces.
  - Cuando expira, cierra la conexión e informa al usuario.
- Envío de un RST *i* en respuesta a cada segmento *i* que recibe.
- El usuario debe decidir cuándo reabrir la conexión:
  - Problemas con los datos perdidos o duplicados.

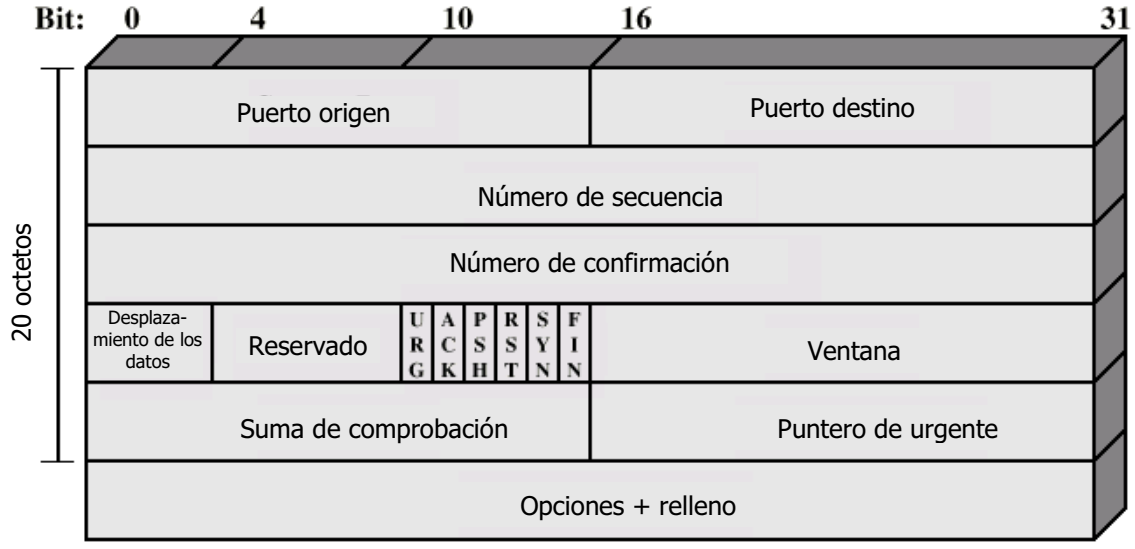
TCP y UDP

- Protocolo de control de la transmisión (TCP):
  - Orientado a conexión.
  - Especificado en el RFC 793.
- Protocolo datagrama de usuario (UDP):
  - No orientado a conexión.
  - Especificado en el RFC 768.

Servicios TCP

- Comunicación segura entre pares de procesos.
- A través de una gran variedad de redes seguras e inseguras, así como sobre un conjunto de redes interconectadas.
- Dos facilidades para etiquetar datos:
  - Cargar flujo de datos:
    - El usuario TCP puede requerir la transmisión de todos los datos etiquetados con un indicador de carga.
    - El receptor entregará los datos en la misma forma.
    - Evita la espera de la memoria temporal.
  - Indicación de datos urgentes:
    - Informa de que en el flujo de datos entrantes existen datos urgentes o significativos.
    - El usuario decide cómo realizar la acción apropiada.

Cabecera TCP



Parámetros pasados a IP

- TCP pasa algunos parámetros a IP:
  - Prioridad.
  - Retardo-normal/bajo retardo.
  - Rendimiento-normal/rendimiento-alto.
  - Seguridad -normal/seguridad-alta.
  - Protección.

Mecanismos TCP

- Establecimiento de la conexión:
  - Diálogo en tres sentidos.
  - Entre un único par de puertos.
  - Un puerto puede conectarse a múltiples destinos.

Mecanismos TCP

- Transferencia de datos:
  - Flujo lógico de octetos.
  - Cada octeto es numerado, módulo 2^{23}.
  - El control de flujo se realiza mediante asignación de número de octetos.
  - Los datos se almacenan en memoria temporal tanto en la transmisión como en la recepción.

Mecanismos TCP

- Cierre de la conexión:
  - Cierre ordenado.
  - Los usuarios TCP emiten una primitiva CLOSE.
  - La entidad de transporte establece el bit FIN en el último segmento enviado.
  - Cierre abrupto mediante una primitiva ABORT:
    - La entidad abandona todos los intentos de enviar o recibir datos.
    - Se envía un segmento RST.

Opciones en los criterios de implementación

- Criterio de envío.
- Criterio de entrega.
- Criterio de aceptación.
- Criterio de retransmisión.
- Criterio de confirmación.

Criterio de envío

- Si no existe indicador de carga (PUSH) o una ventana de transmisión cerrada, la entidad TCP transmite a su propia conveniencia.
- Los datos se almacenan en las memorias temporales de transmisión.
- TCP puede construir un segmento por cada lote de datos.
- Puede esperar hasta que se acumula una cierta cantidad de datos.

Criterio de entrega

- En ausencia del indicador PUSH, TCP entrega los datos de acuerdo a su propia conveniencia.
- Puede entregar los datos conforme se reciben los segmentos.
- Puede almacenar los datos de varios segmentos en las memorias temporales.

Criterio de aceptación

- Los segmentos pueden llegar fuera de secuencia.
- Aceptación en-orden:
  - Sólo acepta segmentos que llegan en orden.
  - Descarta los segmentos que llegan fuera de secuencia.
- Aceptación en-ventana:
  - Acepta todos los segmentos que están dentro de la ventana de recepción.

Criterio de retransmisión

- TCP mantiene una lista de segmentos transmitidos, pero que no han sido confirmados.
- TCP retransmitirá un segmento si no recibe una confirmación dentro de un tiempo determinado:
  - Primero-solamente.
  - Por lotes.
  - Individual.

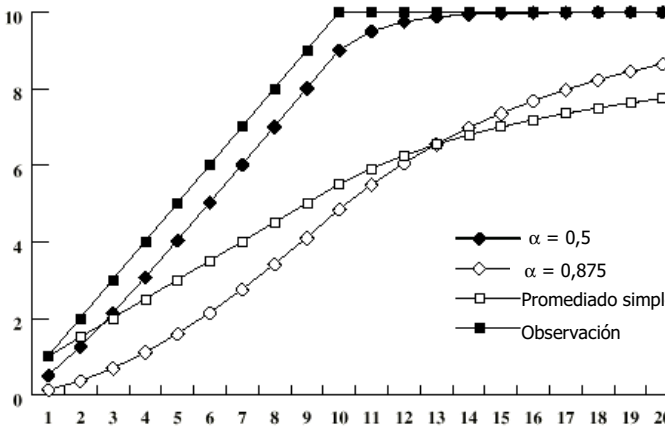
Criterio de confirmación

- Inmediato.
- Acumulativo.

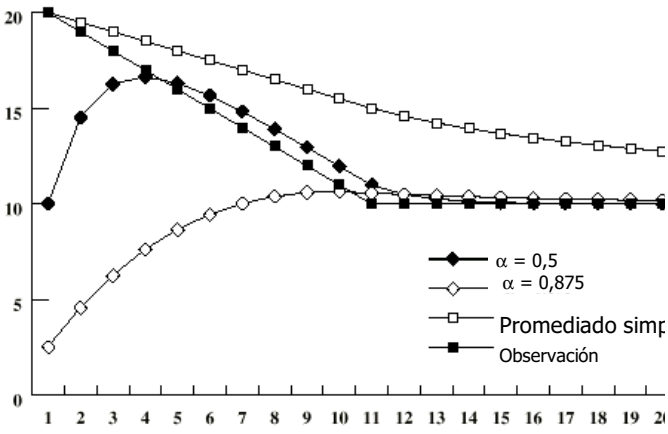
Control de la congestión

- RFC 1122 (Obligaciones de los Computadores en Internet).
- Gestión de los temporizadores de retransmisión:
  - Estima el retardo de ida y vuelta mediante la observación del patrón de retardo.
  - Establece el temporizador a un valor un poco mayor que el estimado.
  - Promediado simple.
  - Promedio exponencial.
  - Estimación de la varianza RTT (Algoritmo de Jacobson).

Utilización del promediado exponencial

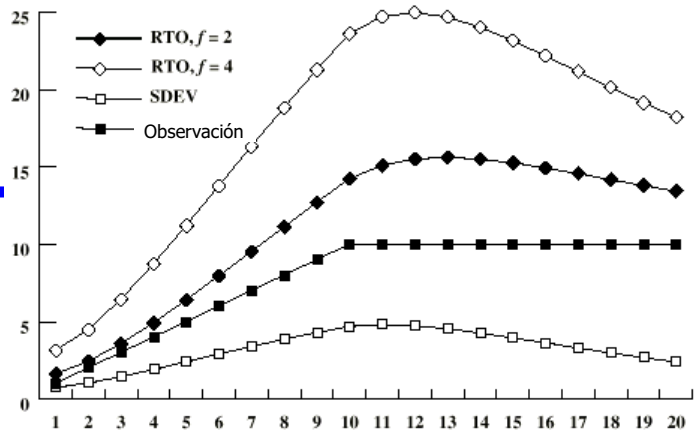


(a) Función creciente

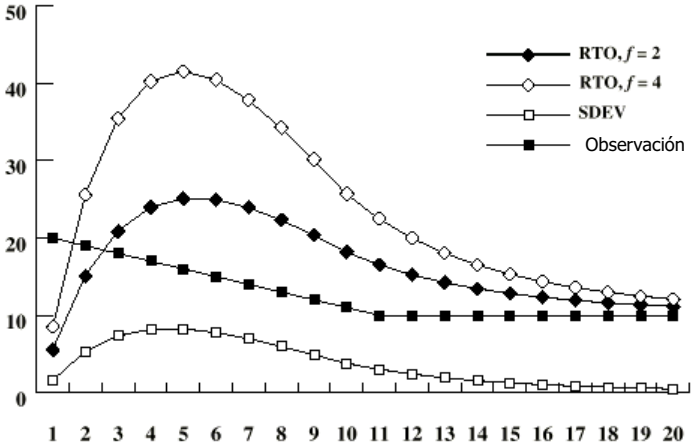


(b) Función decreciente

Cálculo del RTO de Jacobson



(a) Función creciente



(b) Función decreciente

Decaimiento exponencial de RTO

- Aunque el temporizador expire a causa de la congestión (descarte del paquete o retardo grande en el tiempo de ida y vuelta), no es aconsejable mantener el mismo valor de RTO.
- RTO incrementa cada vez que se retransmite un segmento.
- RTO = q *x* RTO
- El valor de q más utilizado es 2:
  - Decaimiento exponencial binario.

Algoritmo de Karn

- Si se retransmite un segmento, el ACK que se recibe puede ser:
  - De la primera transmisión del segmento:
    - RTT es simplemente mayor que el esperado.
  - De la segunda transmisión.
- TCP no puede distinguir entre estos dos casos.
- No mide el RTT para segmentos retransmitidos.
- Calcular el decaimiento cuando ocurra una retransmisión.
- Utilizar el valor del RTO de decaimiento hasta que llegue el ACK para un segmento que no se ha retransmitido.

Gestión de la ventana

- Comienzo lento:
  - *awnd* = MIN[*crédito*, *cwnd*]
  - Se abre una conexión con *cwnd* = 1.
  - Se incrementa *cwnd* en cada ACK, hasta algún valor máximo.
- Ajuste dinámico de la ventana en caso de congestión:
  - Cuando expira un temporizador.
  - Establecer un umbral de comienzo lento igual a la mitad de la ventana de congestión actual:
    - *ssthresh* = *cwnd*/2
  - Hacer *cwnd* = 1 y comienzo lento hasta que *cwnd* = *ssthresh*:
    - Se incrementa *cwnd* a 1 por cada ACK.
  - Para *cwnd* > = *ssthresh*, incrementar *cwnd* a 1 para cada tiempo de ida-y-vuelta.

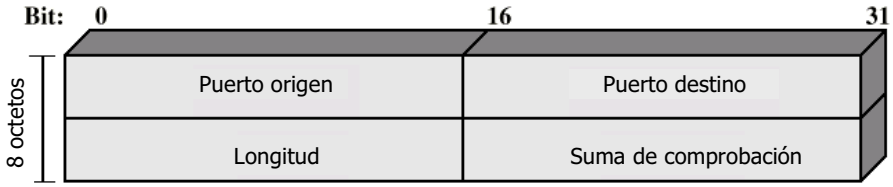
UDP

- Protocolo de datagrama de usuario.
- Especificado en el RFC 768.
- Servicio no conectado a conexión para procedimientos de la capa de aplicación:
  - Servicio no seguro.
  - La entrega y la protección contra duplicados no están garantizadas.
- Se reduce la información suplementaria.
- Ejemplo: gestión de red (véase el Capítulo 19).

Usos del UDP

- Recolección de datos de entrada.
- Diseminación de datos de salida.
- Petición-respuesta.
- Aplicaciones en tiempo real.

Cabecera UDP



Lecturas recomendadas

- Stallings, W. *Comunicaciones y Redes de Computadores*, sexta edición. Madrid: Prentice Hall, 2000: Capítulo 17.
- Stevens, W. *TCP/IP Illustrated, Volume 1: The protocols*. Reading, MA: Addison-Wesley, 1994.