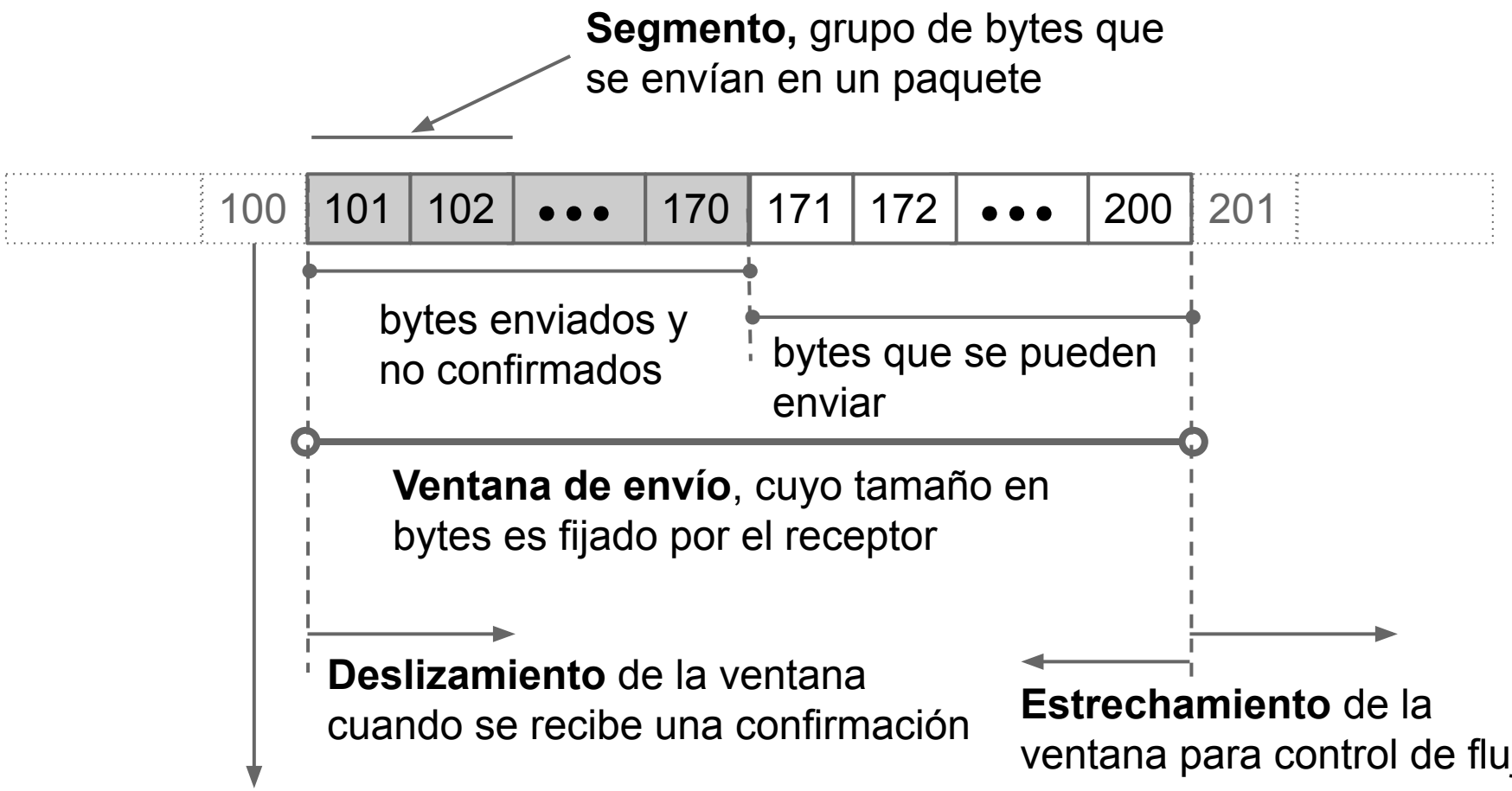
TEMA 1.2. Conceptos Avanzados del Protocolo TCP

PROFESORES:
Rubén Santiago Montero
Eduardo Huedo Cuesta
Rafael Rodríguez Sánchez

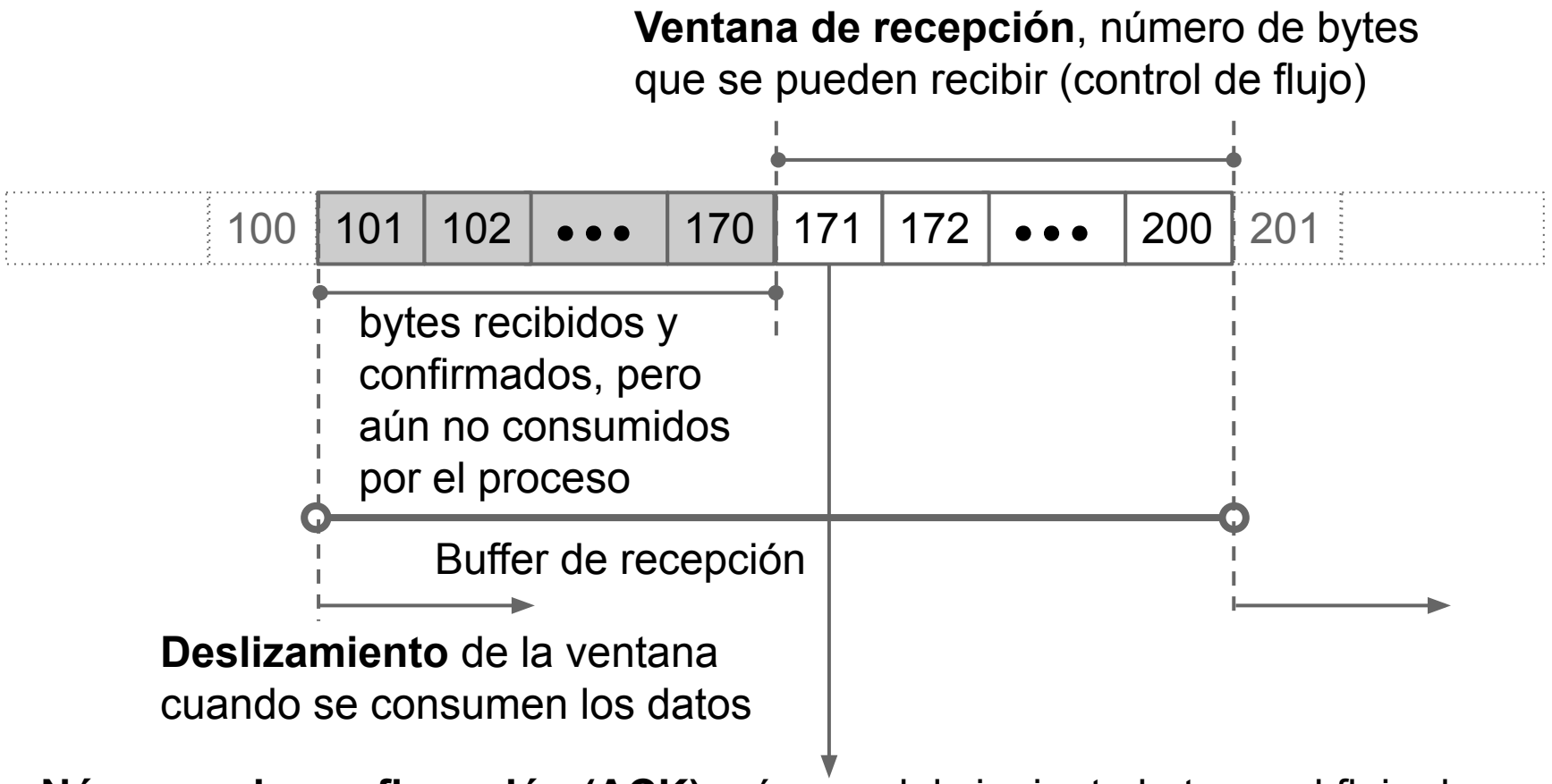
El protocolo TCP: Características

- Servicios ofrecidos por TCP:
  - Comunicación lógica proceso-proceso, usando números de puerto
  - Transferencia como flujo de bytes (*byte stream*)
  - Transmisión orientada a conexión y fiable
  - Full-duplex y multiplexación
- Orientado a conexión, define las siguientes fases para la transmisión:
  - Establecimiento de conexión
  - Transferencia de datos
  - Cierre de conexión
- Unidad de transferencia: Segmento TCP
- Fiable, incluye mecanismos de control de errores de tipo ventana deslizante con:
  - Códigos de comprobación (*checksum*)
  - Numeración de segmentos
  - Confirmaciones selectivas y acumuladas, superpuestas del receptor
  - Retransmisión de segmentos perdidos o erróneos
  - Temporizadores

Ventana Deslizante: La Ventana de Envío



Ventana Deslizante: La Ventana de Recepción



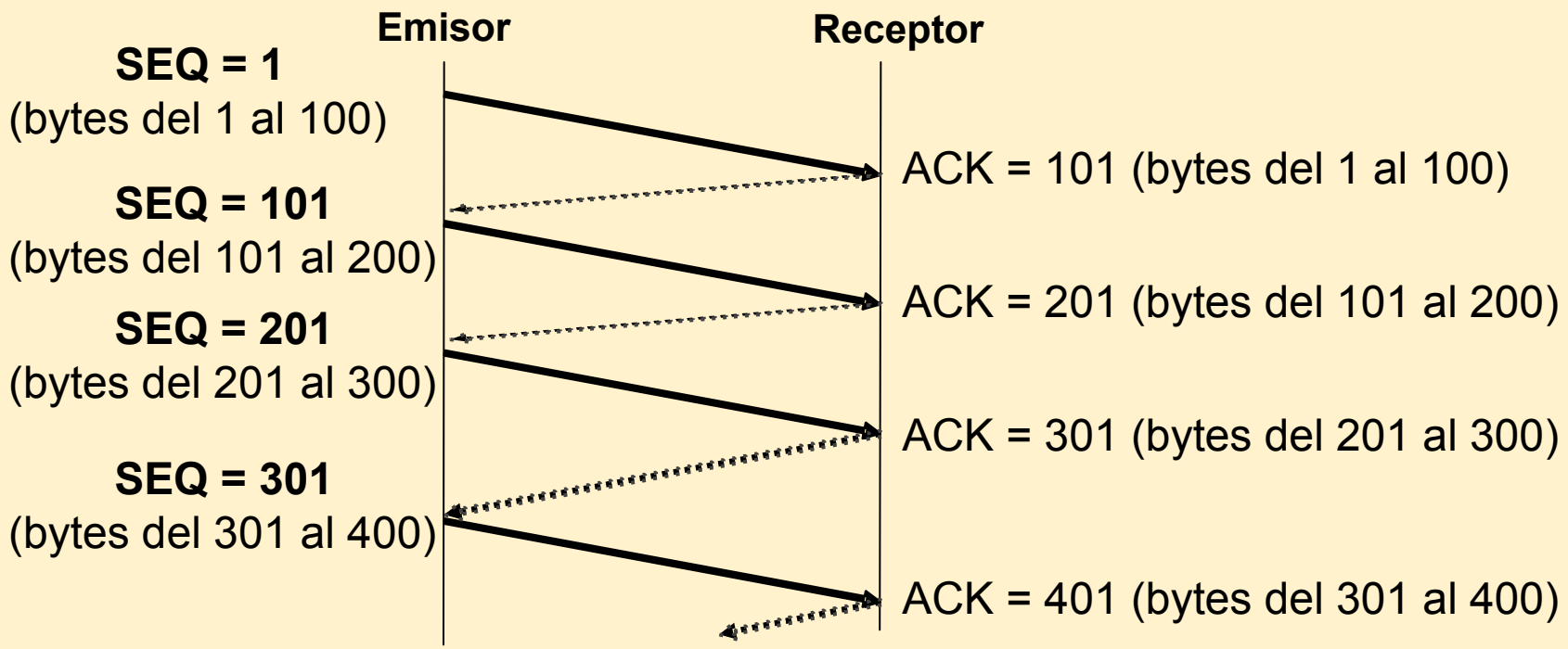
Números de confirmación (ACK), número del siguiente byte en el flujo de datos que se espera recibir.

- Confirman todos los bytes anteriores al de ACK (acumulativos)
- Se solapan con el envío de datos (*piggybacking*)

Ventana Deslizante: Funcionamiento

Ejemplo: Transmisión sin errores.

Tamaño de la ventana = 100 bytes, Tamaño del segmento = 100 bytes.

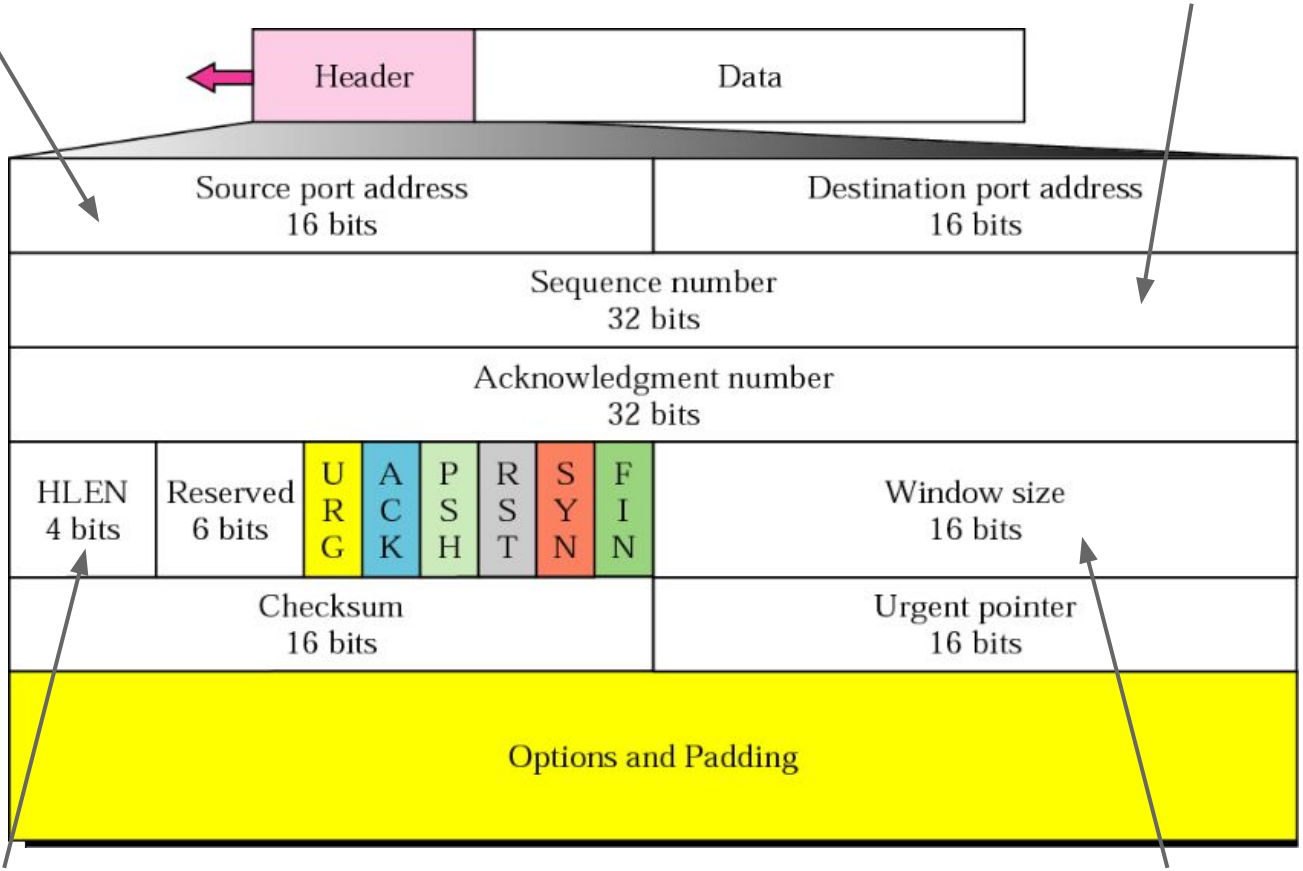


Ejemplo: ¿Cuáles serían los números de secuencia y confirmación para un tamaño de segmento de 50 bytes? (Nota, las confirmaciones en TCP no tienen que realizarse de forma inmediata)

Formato del Segmento TCP

Puertos, identifican los extremos de la conexión

Números de secuencia y confirmación, expresados en bytes en el flujo de datos



Longitud de la cabecera en palabras de 32 bits (20-60 bytes)

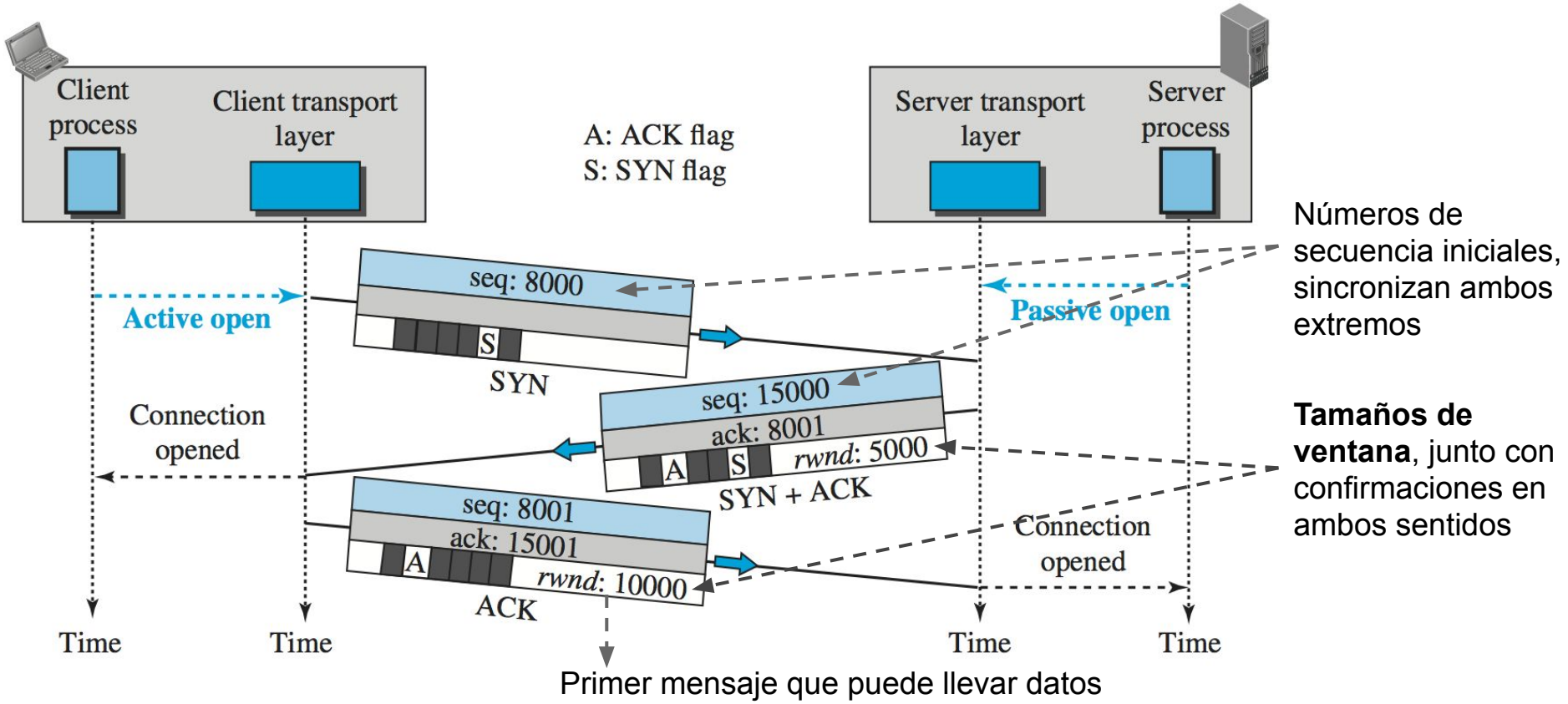
Tamaño de la ventana en bytes (control de flujo)

Formato del Segmento TCP

Flags del campo de control (6 bits)

- **SYN**: Utilizado en el establecimiento de la conexión y sincronizar los números de secuencia iniciales
- **FIN**: Utilizado en la finalización de la conexión
- **ACK**: El segmento contiene un número de confirmación válido (ACK=1). Todos los segmentos de una conexión TCP, excepto el primero, llevan ACK=1
- **RST**: Utilizado para denegar o abortar una conexión
- **PSH**: Los datos deben ser entregados inmediatamente a la aplicación (PSH=1), o pueden almacenarse en el buffer (PSH=0)
- **URG**: El segmento transporta datos urgentes (URG=1) desde el primer byte hasta el nº de byte especificado en el campo **Urgent pointer**
    - TCP notifica a la aplicación de los datos urgentes (SIGURG)
    - El tratamiento de *urgencia* corresponde a la aplicación, no a TCP

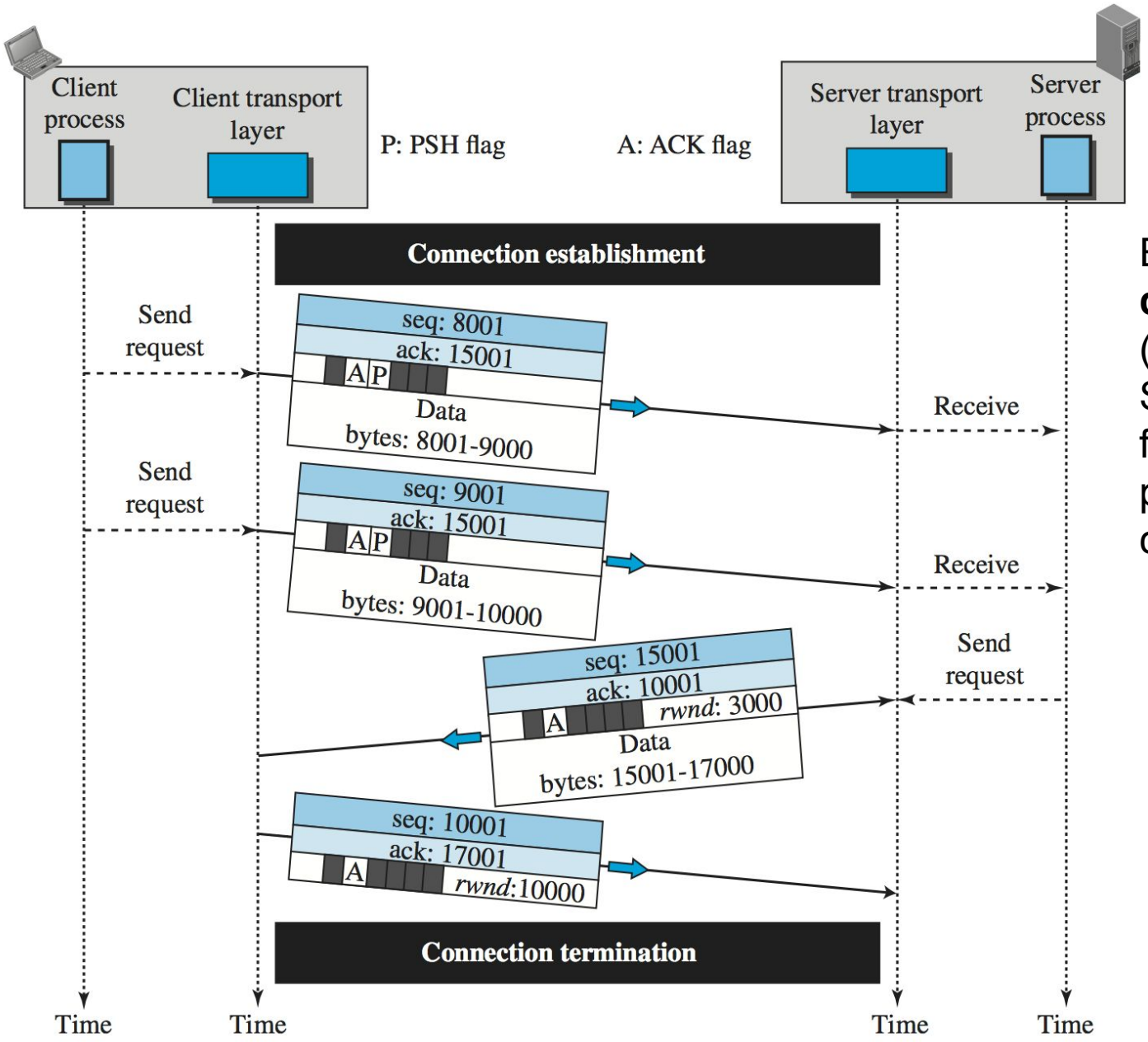
Fases de la Conexión: Establecimiento (3-way)



Ataque TCP SYN Flooding

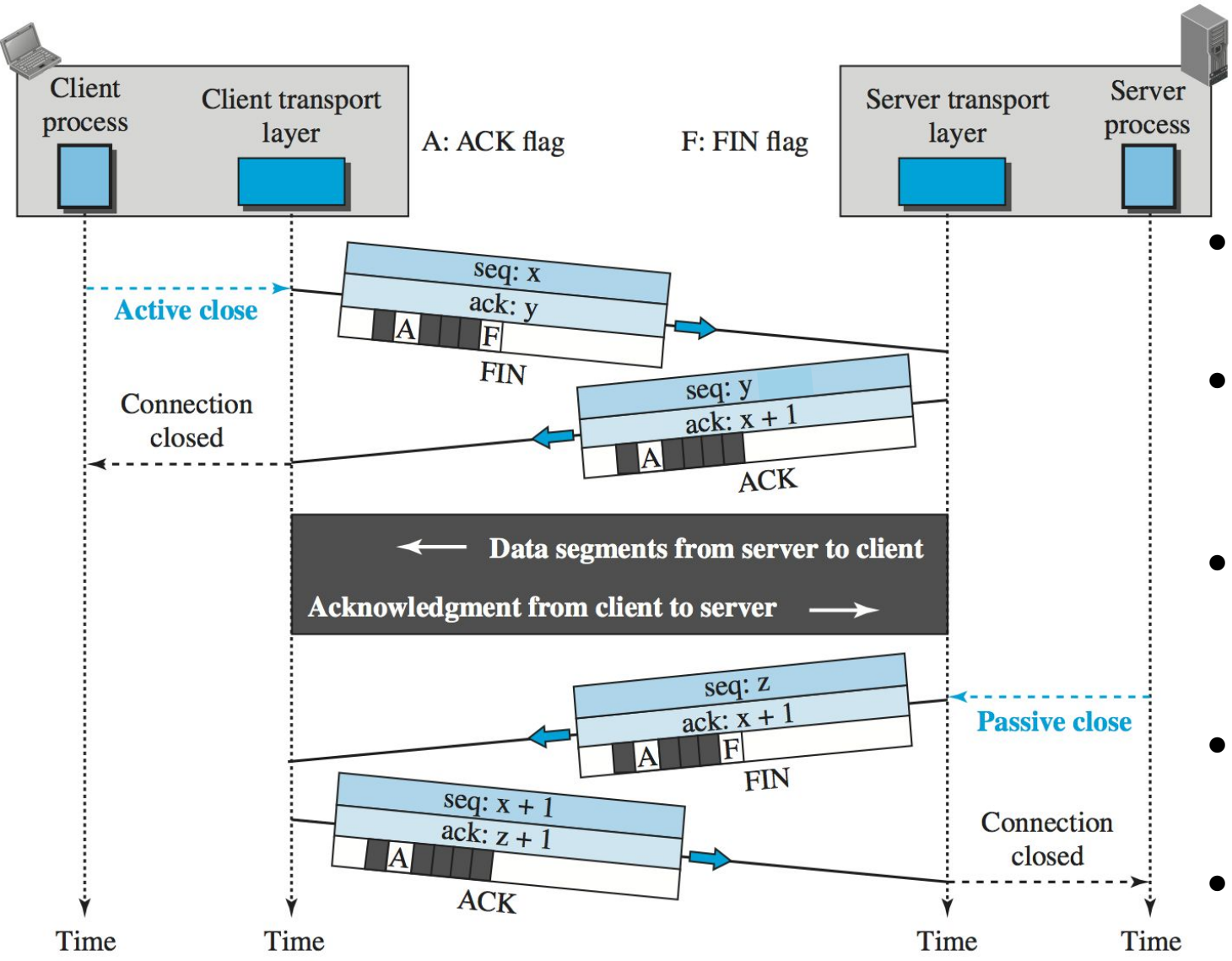
- Se envía una gran cantidad de segmentos TCP con el flag SYN activado, que consumen recursos del servidor y puede llegar a no responder (DoS)
- Contramedidas: filtrar conexiones (limitar tasa o detectar IPs suplantadas), aumentar recursos o retrasar la asignación de recursos (usando SYN *cookies*)

Fases de la Conexión: Transferencia



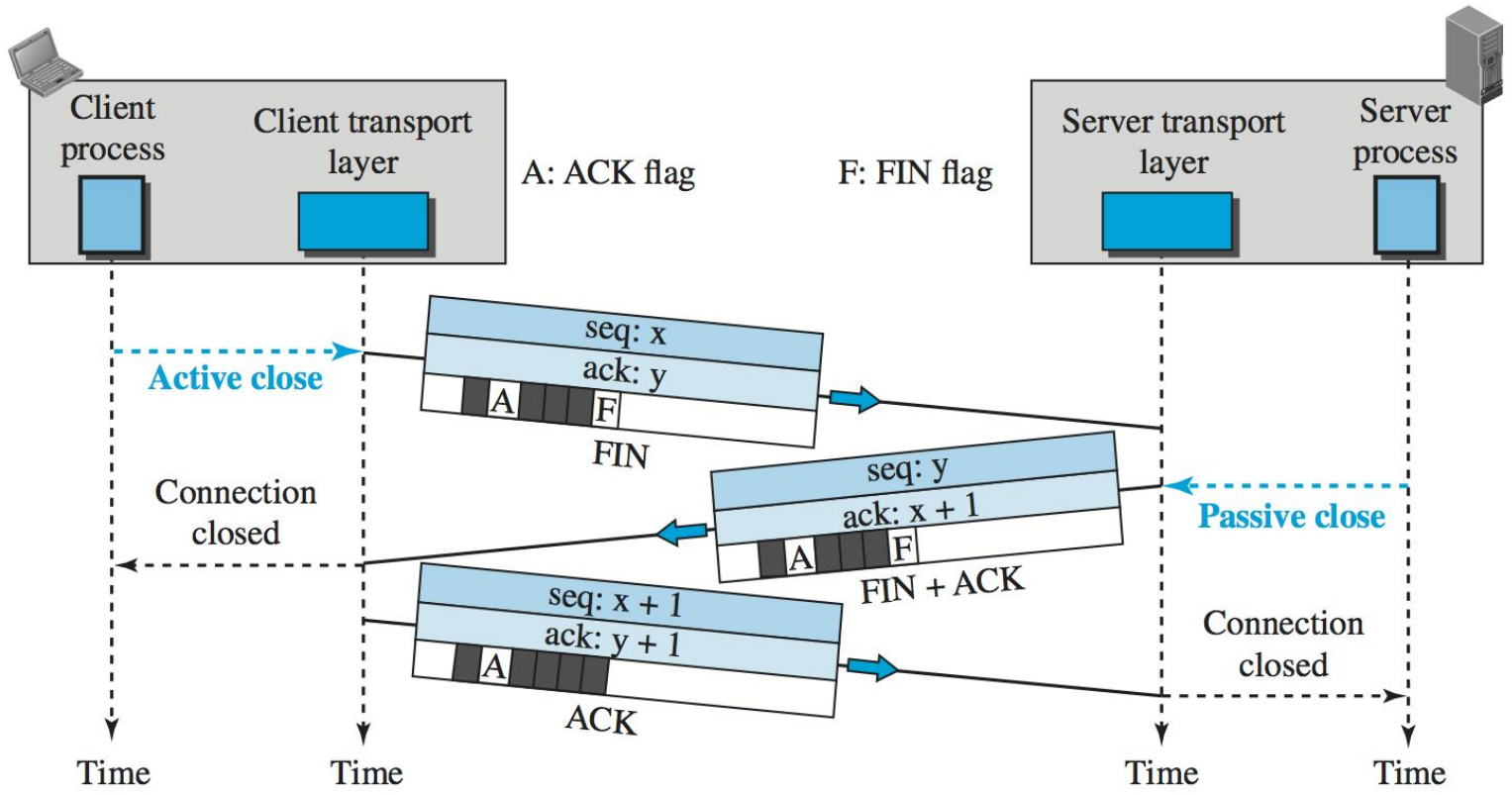
El **tamaño máximo del segmento** (Maximum Segment Size, MSS) es fijado de forma independiente por cada extremo en el campo Options

Fases de la Conexión: Finalización (4-way)



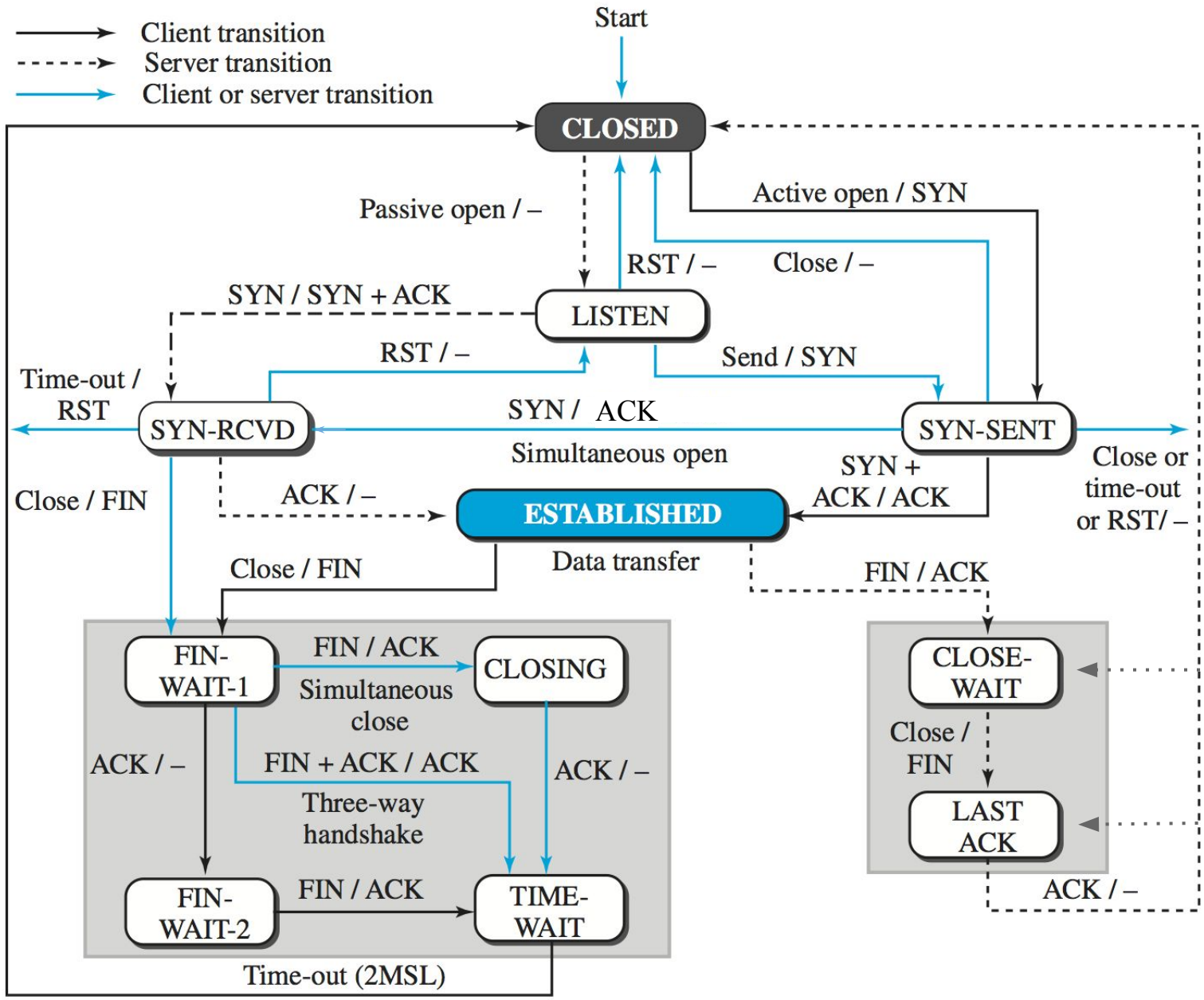
- El cliente inicia el cierre con un mensaje FIN
- El servidor confirma el mensaje pero no inicia la finalización de su extremo
- El cliente deja de enviar datos, pero el servidor continúa enviando datos
- El servidor termina la conexión con un mensaje FIN
- El cliente confirma el mensaje

Fases de la Conexión: Finalización (3-way)



- Ambos extremos dejan de enviar información
- Los mensajes de FIN pueden contener datos. Siempre consume un número de secuencia como mínimo ya que deben ser confirmados
- El último ACK no lleva datos

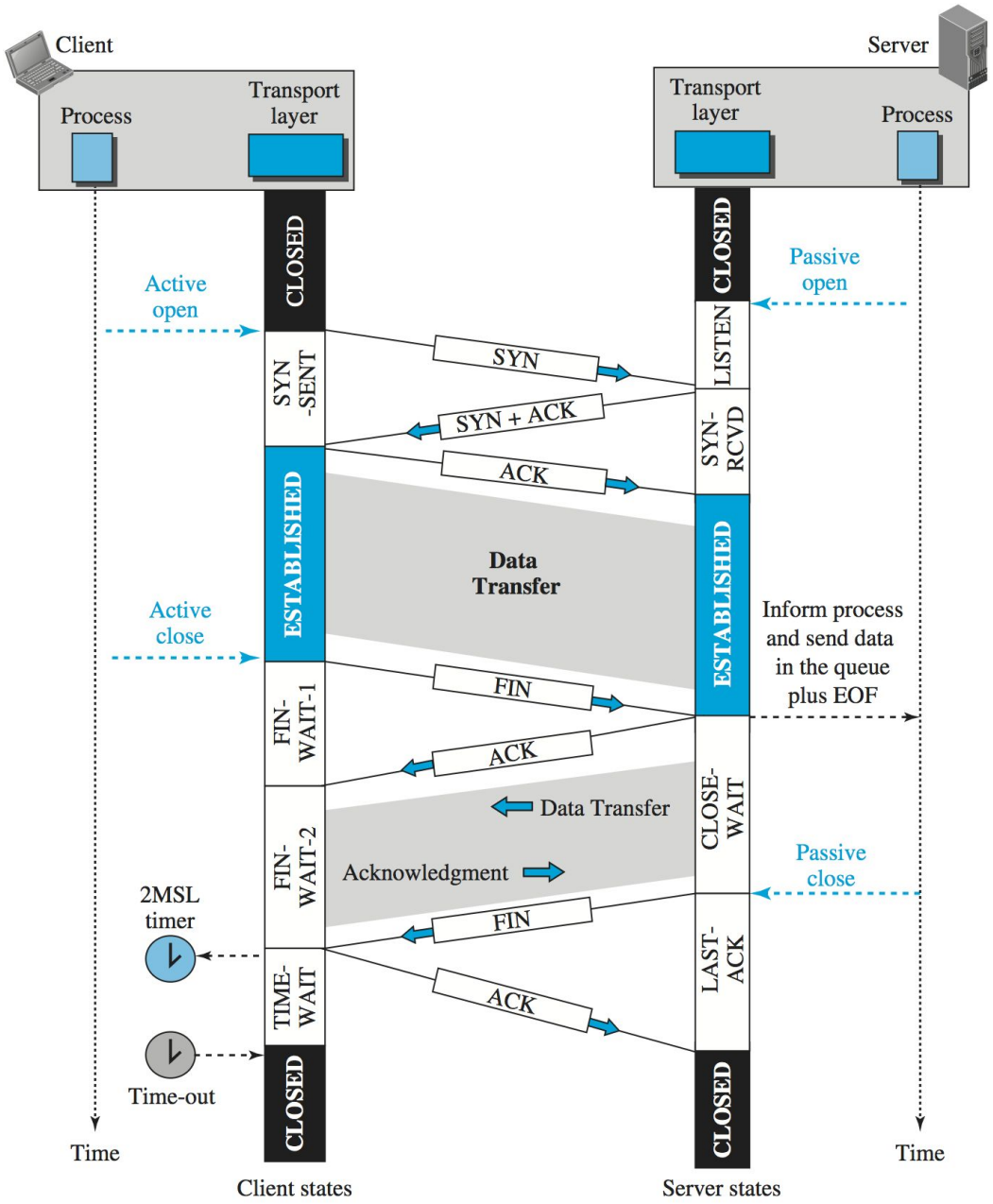
Fases de la Conexión: Máquina de Estados

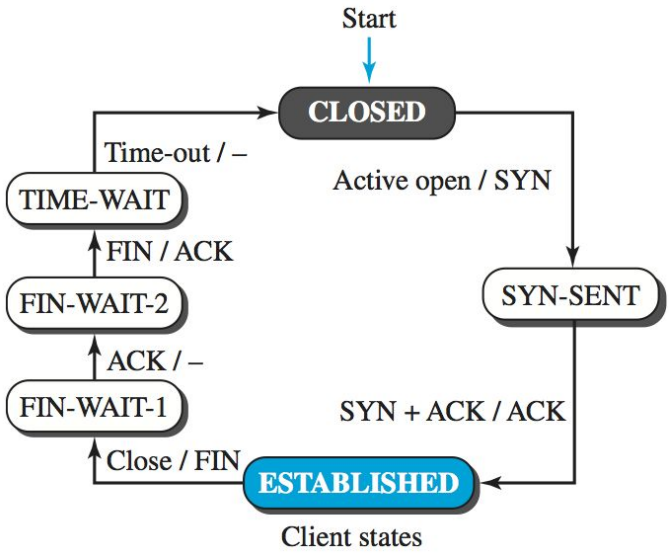


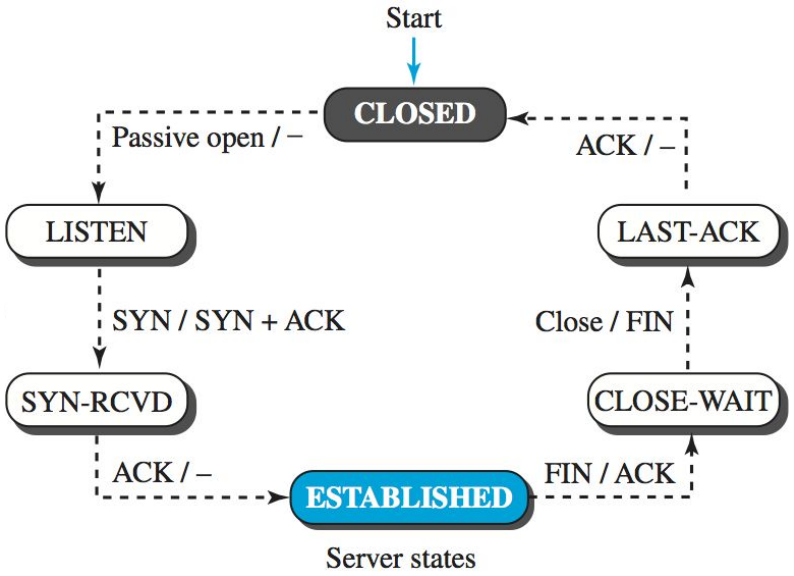
Esperando el cierre de la aplicación

2º FIN enviado

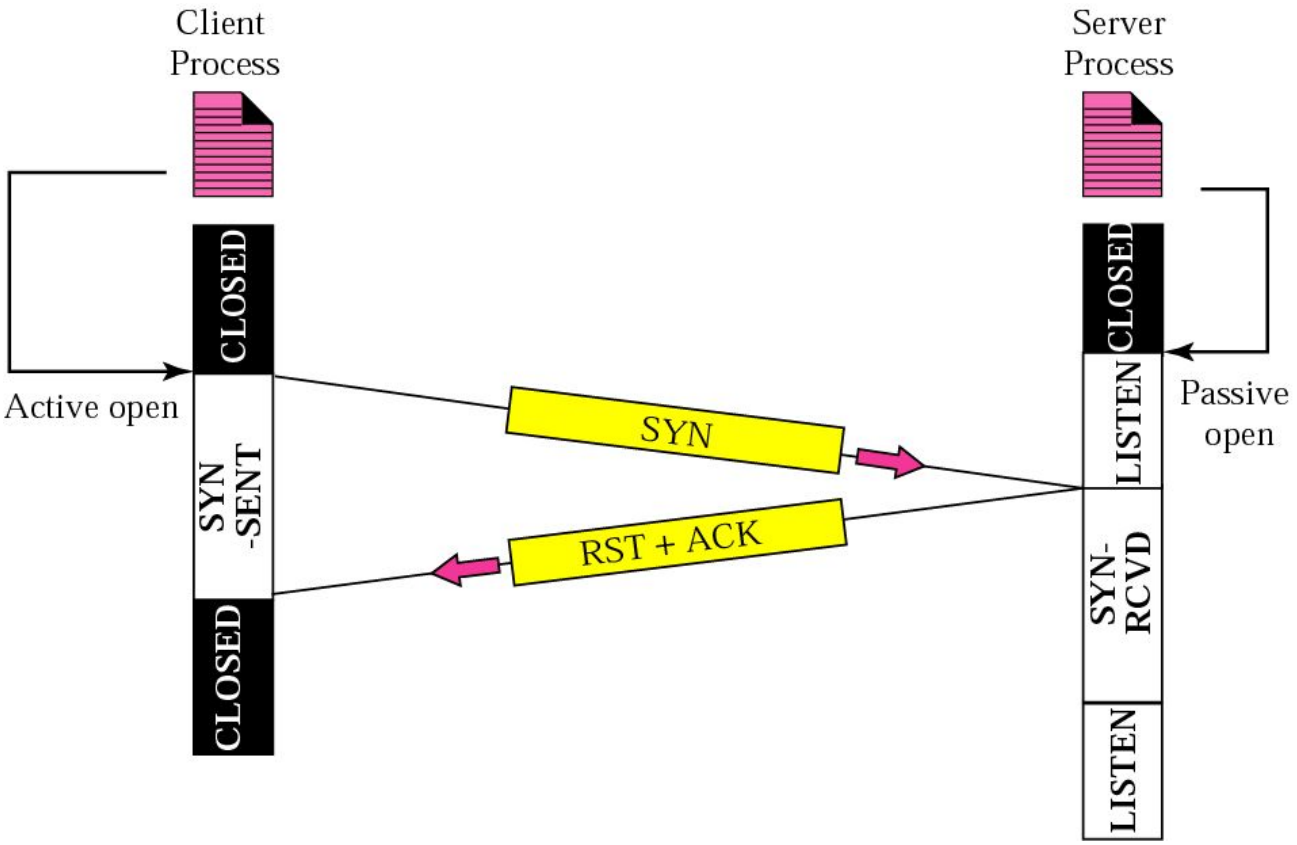
Fases de la Conexión: Máquina de Estados







Fases de la Conexión: Máquina de Estados



Ejemplos: Describir la secuencia de estados para el cliente y servidor durante:

- El cierre de tres vías
- El cierre simultáneo

Control de Errores: Confirmaciones

- El control de errores se realiza usando el mecanismo de ventana deslizante que permite gestionar:
  - La retransmisión de segmentos erróneos o perdidos
  - La recepción de segmentos duplicados o fuera de orden

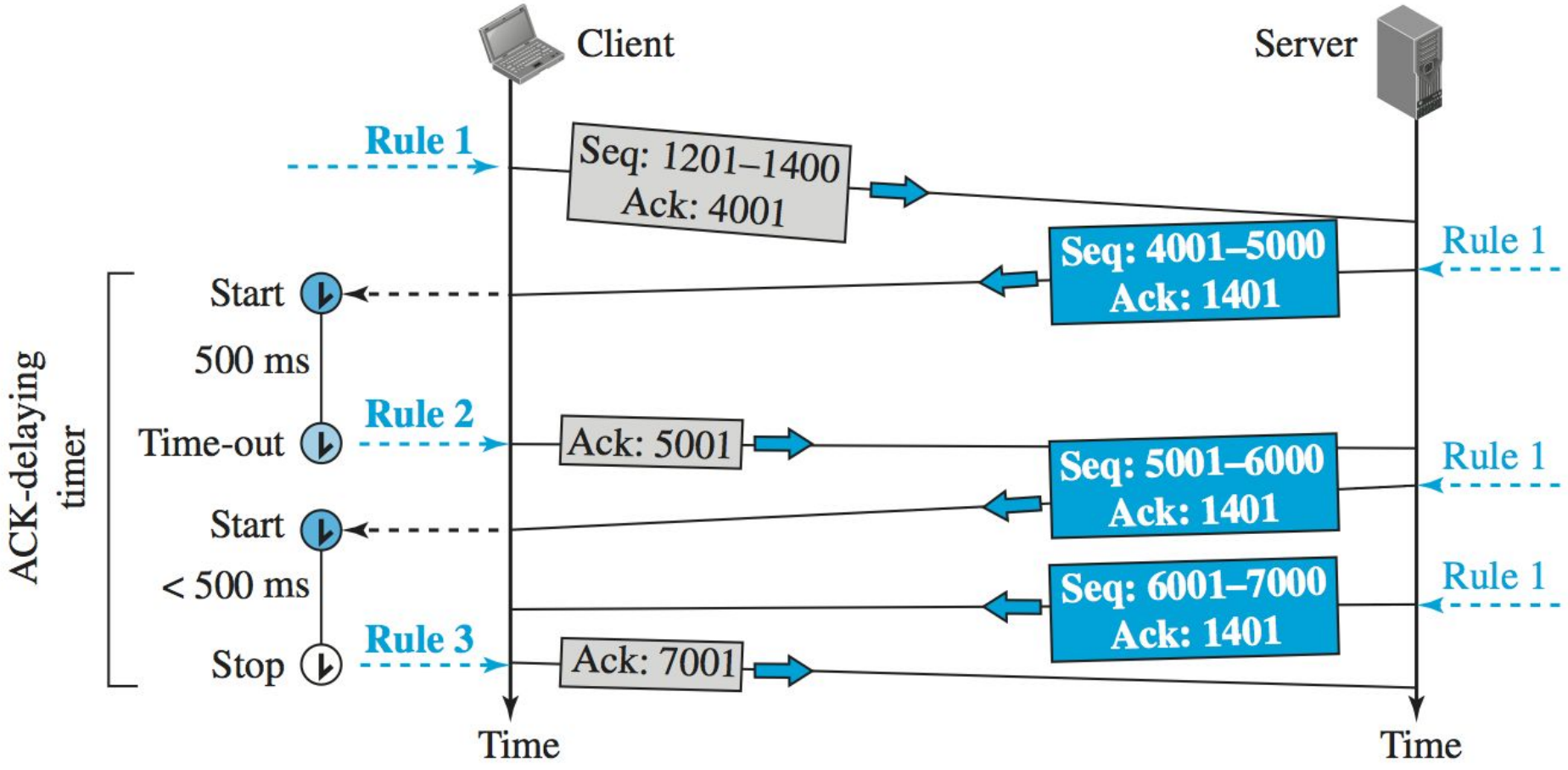
Reglas para las confirmaciones

1. Los segmentos de datos deben incluir una confirmación (*piggyback*) indicando el siguiente número de secuencia esperado, que confirma todos los anteriores
2. Si no hay datos que enviar, las confirmaciones de segmentos recibidos en orden pueden retrasarse 500 ms para incluirlas en un segmento de datos
3. Solo puede retrasarse la confirmación de un segmento
4. Los segmentos recibidos fuera de orden se confirman inmediatamente (esto provocará una retransmisión rápida, como veremos)
5. Los segmentos que completan huecos se confirman inmediatamente
6. Los segmentos duplicados se confirman para solucionar la pérdida de un ACK

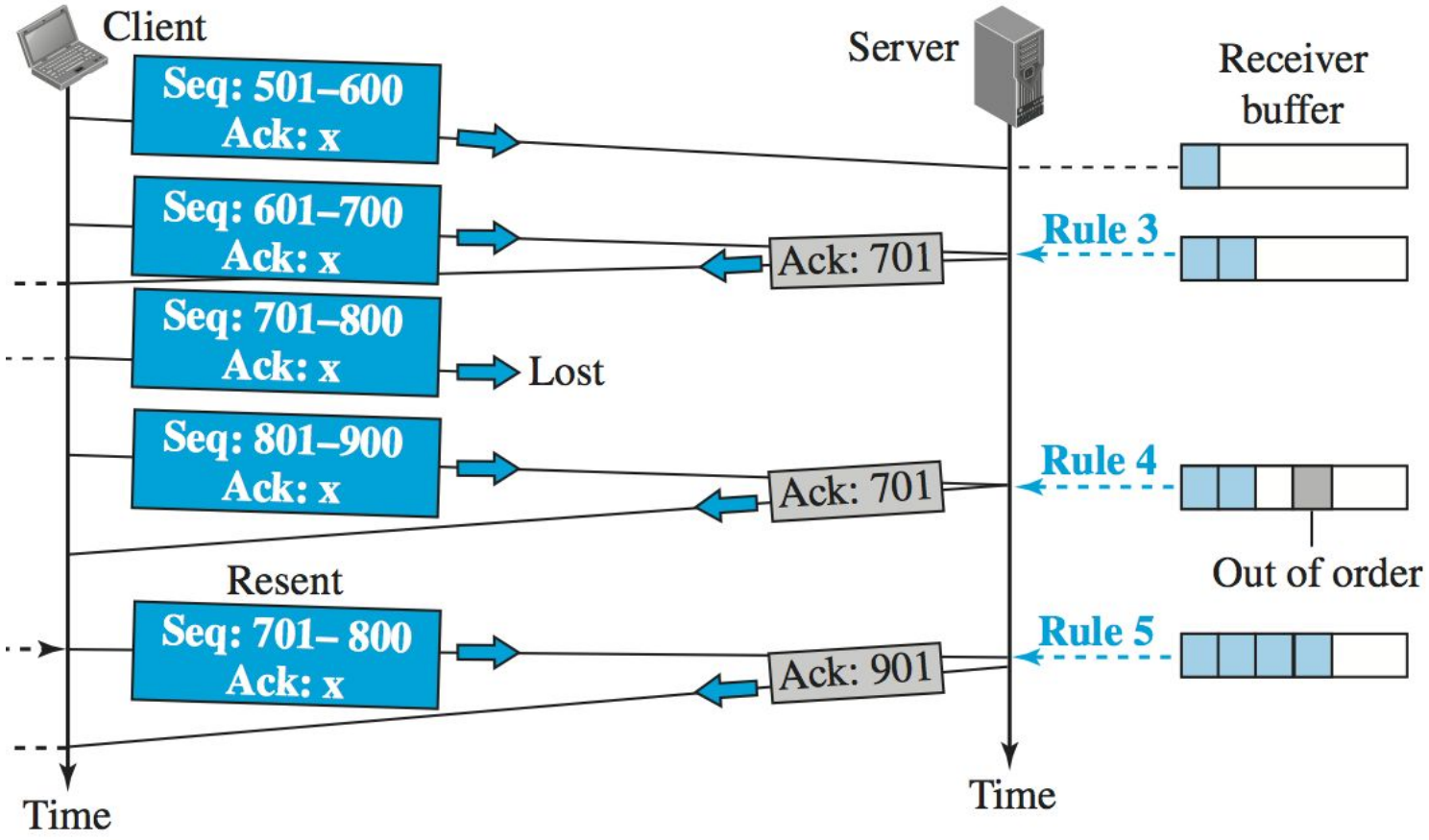
Opcional

- Confirmaciones selectivas (SACK) de segmentos fuera de orden
  - No reemplazan los ACK, informativos para el emisor
  - Implementados como opción TCP

Control de Errores: Transmisión sin Error



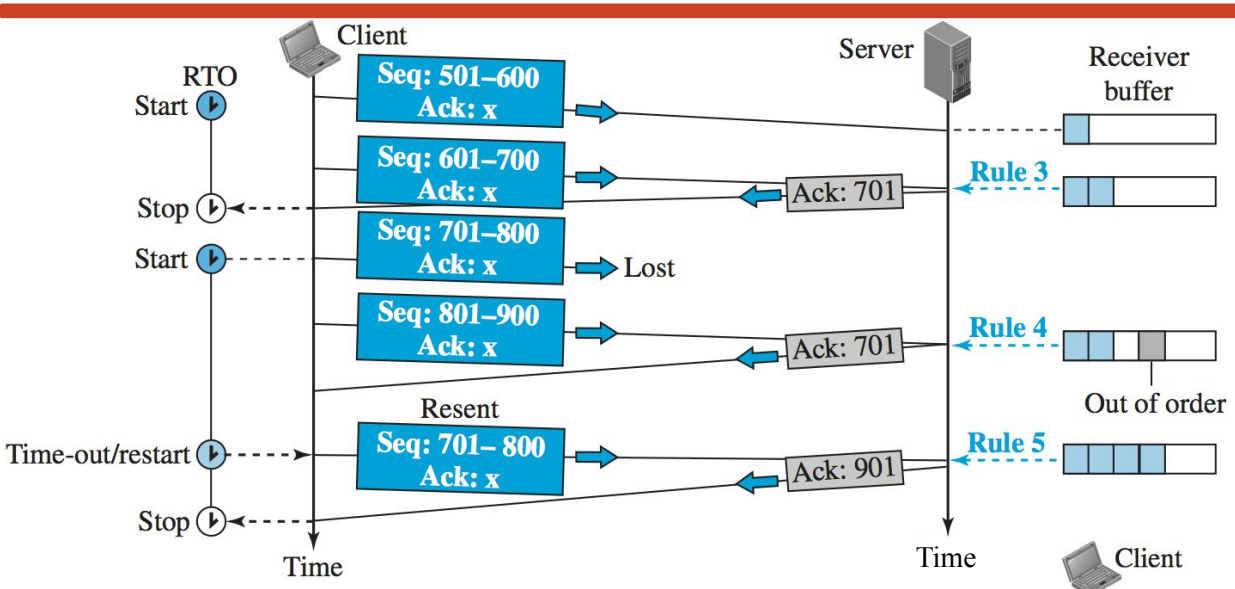
Control de Errores: Recepción fuera de orden



Control de Errores: Retransmisión

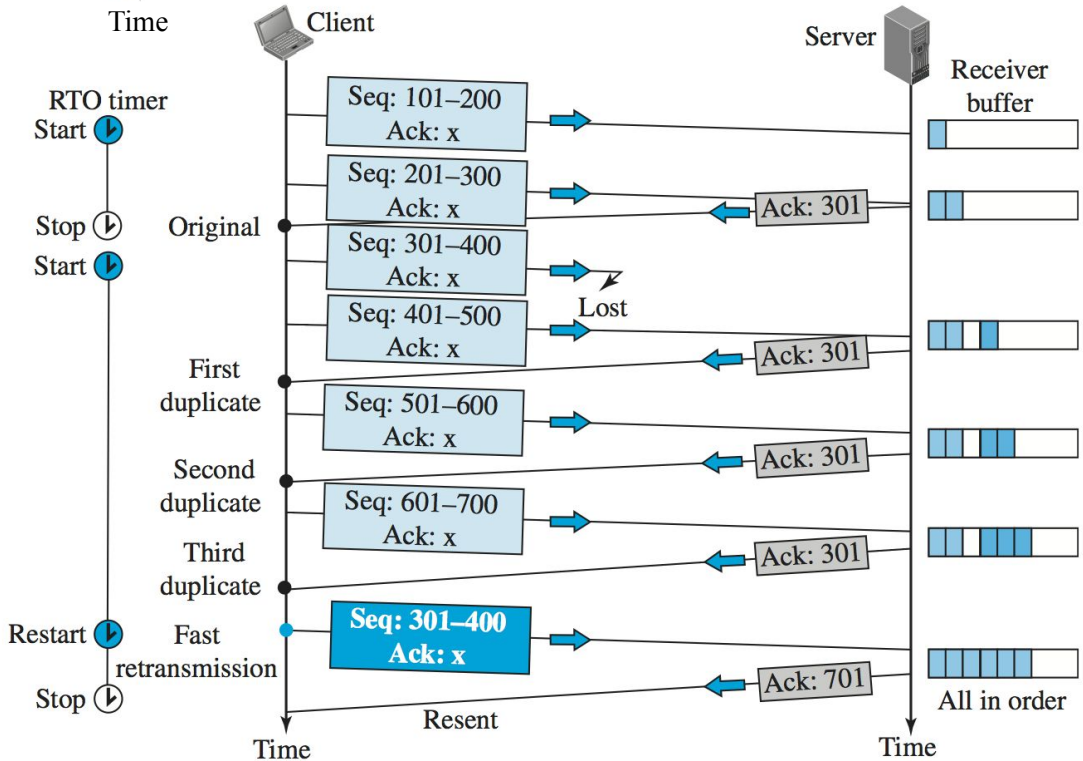
- La capacidad para volver a transmitir un segmento TCP cuando no se recibe o se recibe erróneamente es el núcleo del control de errores
- TCP dispone de dos mecanismos de retransmisión:
  - **Temporizador de retransmisión**
    - Se inicia cuando se envía un segmento y se para al recibir la confirmación
    - Si expira, se retransmite el primer segmento sin confirmar de la ventana
    - Existen diversos algoritmos para fijar el RTO (Retransmission Time-Out), que debe ser mayor que el RTT (Round-Trip Time)
    - Cada conexión debe usar un único temporizador (ver RFC 6298)
  - **Retransmisión rápida**
    - Se retransmite cuando se reciben 3 ACKs duplicados
    - No requiere que expire el temporizador de retransmisión

Control de Errores: Pérdida de un Segmento

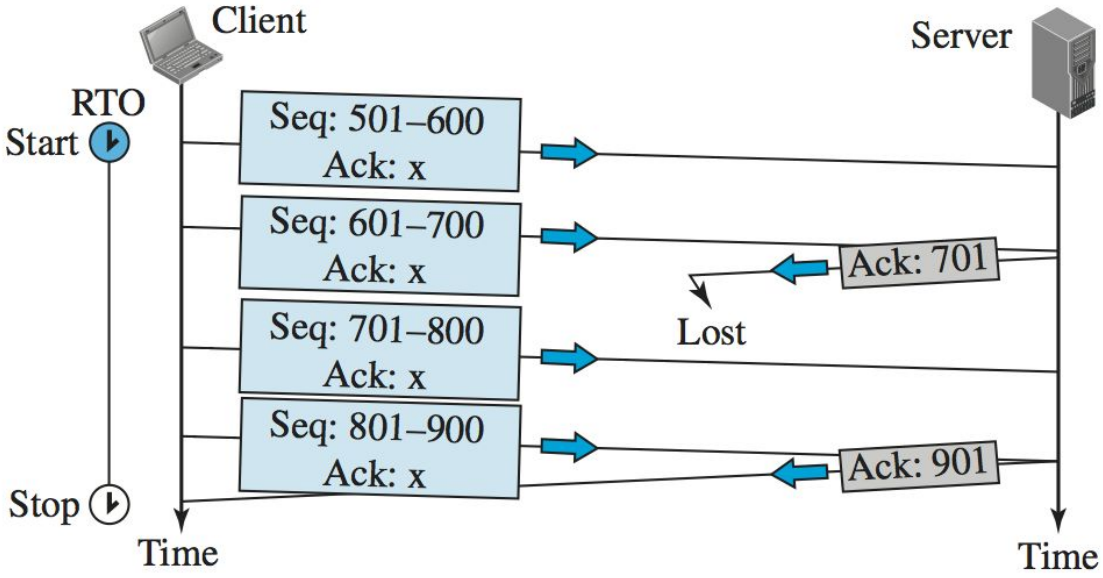


Temporizador de retransmisión expirado

Retransmisión rápida

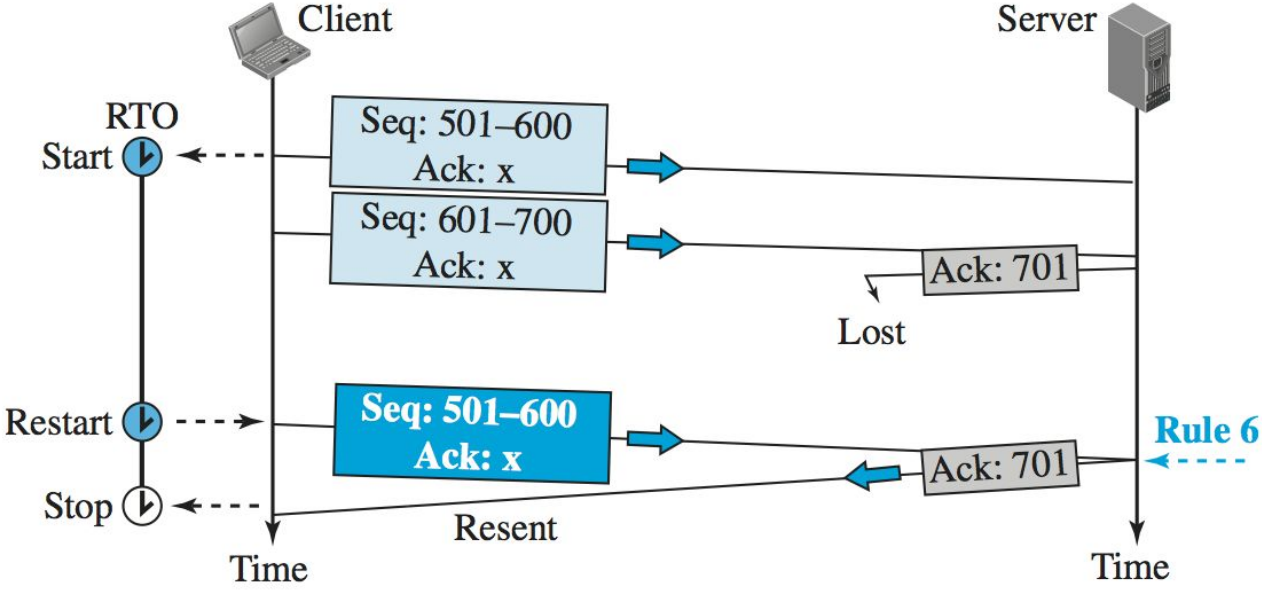


Control de Errores: Pérdida de un ACK



Sin expirar el temporizador de retransmisión

Temporizador de retransmisión expirado



Temporizadores TCP

- **Retransmisión**
- **Keepalive**, que evita mantener conexiones indefinidamente
  - Una conexión puede estar en silencio `tcp_keepalive_time` segundos
  - Después, se envía un máximo de `tcp_keepalive_probes` sondas cada `tcp_keepalive_intvl` segundos
  - Si no se recibe ningún ACK para las sondas, se cierra la conexión
  - Ej. 2 horas, 10 sondas cada 75 segundos
- **TIME-WAIT**, útil en dos situaciones:
  - Reenviar el último ACK durante el cierre activo si se retransmite FIN
  - Permitir que expiren los posibles segmentos duplicados
  - Impide una nueva conexión con los mismos parámetros (direcciones y puertos origen y destino)
  - 2*MSL (Maximum Segment Lifetime). Ej. 60 segundos
- **Temporizador de persistencia**, asociado al anuncio de un tamaño de ventana 0
  - Recupera la pérdida de un ACK posterior con el nuevo tamaño
  - Se envía una sonda que fuerza el envío de un ACK
  - Ej. 60 segundos

Temporizador de Retransmisión

- La elección del tiempo de vencimiento del temporizador de retransmisión (RTO) está basada en los retardos observados en la red (RTT)
- Los retardos en la red pueden variar dinámicamente, por tanto el temporizador debe adaptarse a esta situación
- Las principales técnicas utilizadas para fijar el temporizador de retransmisión son las siguientes:
  - Método de la media ponderada (algoritmo de Jacobson)
  - Método de la varianza (algoritmo de Jacobson/Karels)
  - Algoritmo de Karn

Temporizador de Retransmisión

Tiempo de ida y vuelta medido (RTT_M)

- Cuando se envía segmento, se mide el tiempo transcurrido desde que se envía el segmento hasta que se recibe el ACK, denominado RTT_M (Measured Round-Trip Time)
- Sólo hay un temporizador RTT_M
- El valor del RTT_M puede experimentar grandes fluctuaciones

Tiempo de ida y vuelta suavizado (RTT_S)

- Evitar las fluctuaciones del RTT
- RTT_S (Smoothed Round-Trip Time) es la media ponderada entre el RTT_M y el último RTT_S calculado:

  1ª medida: $RTT_S = RTT_M$

  Siguientes: $RTT_S = (1 - \alpha) \times RTT_S + \alpha \times RTT_M$, con $\alpha < 1$ (ej. $\alpha = ⅛$)

Desviación del RTT (RTT_D)

- Para considerar la variación del tiempo de ida y vuelta

  1ª medida: $RTT_D = RTT_M / 2$

  Siguientes: $RTT_D = (1 - \beta) \times RTT_D + \beta \times |RTT_S - RTT_M|$, con $\beta < 1$ (ej. $\beta = 1/4$)

Temporizador de Retransmisión

Algoritmo de Jacobson

- Considera únicamente el RTT_S
- El RTO se calcula después de cada medida del RTT como

  $RTO = \gamma \times RTT_S$ (ej. $\gamma=2$, el doble del RTT estimado)

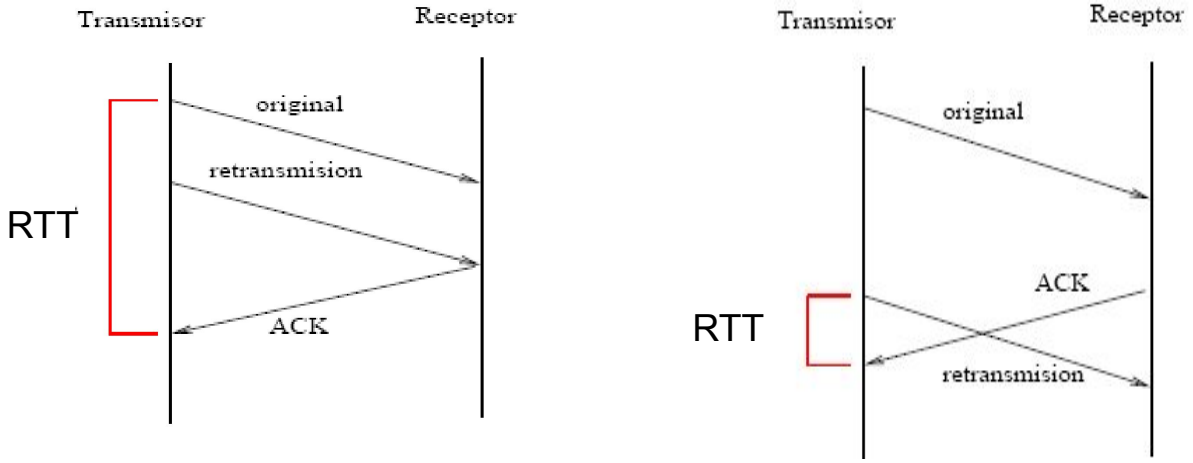
Algoritmo de Jacobson/Karels

- Combina RTT_S y RTT_D

  $RTO = RTT_S + 4 \times RTT_D$

Algoritmo de Karn

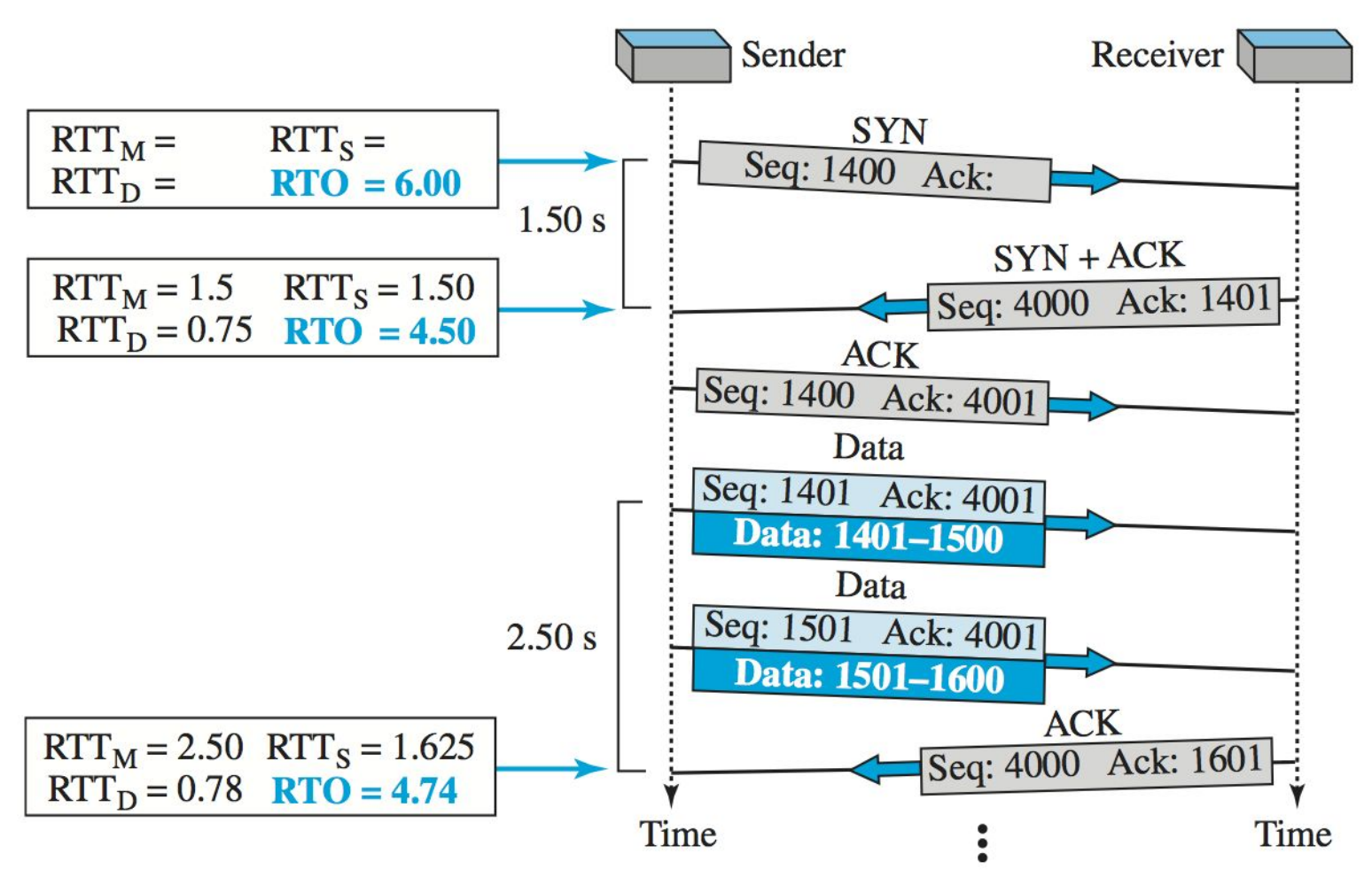
- Previene la ambigüedad en el cálculo del RTT_M cuando hay retransmisiones
- En esos casos, no actualiza RTT_S y RTT_D y duplica el RTO (***exponential backoff***)



Temporizador de Retransmisión

Ejemplo: Calcular los valores de los temporizadores:

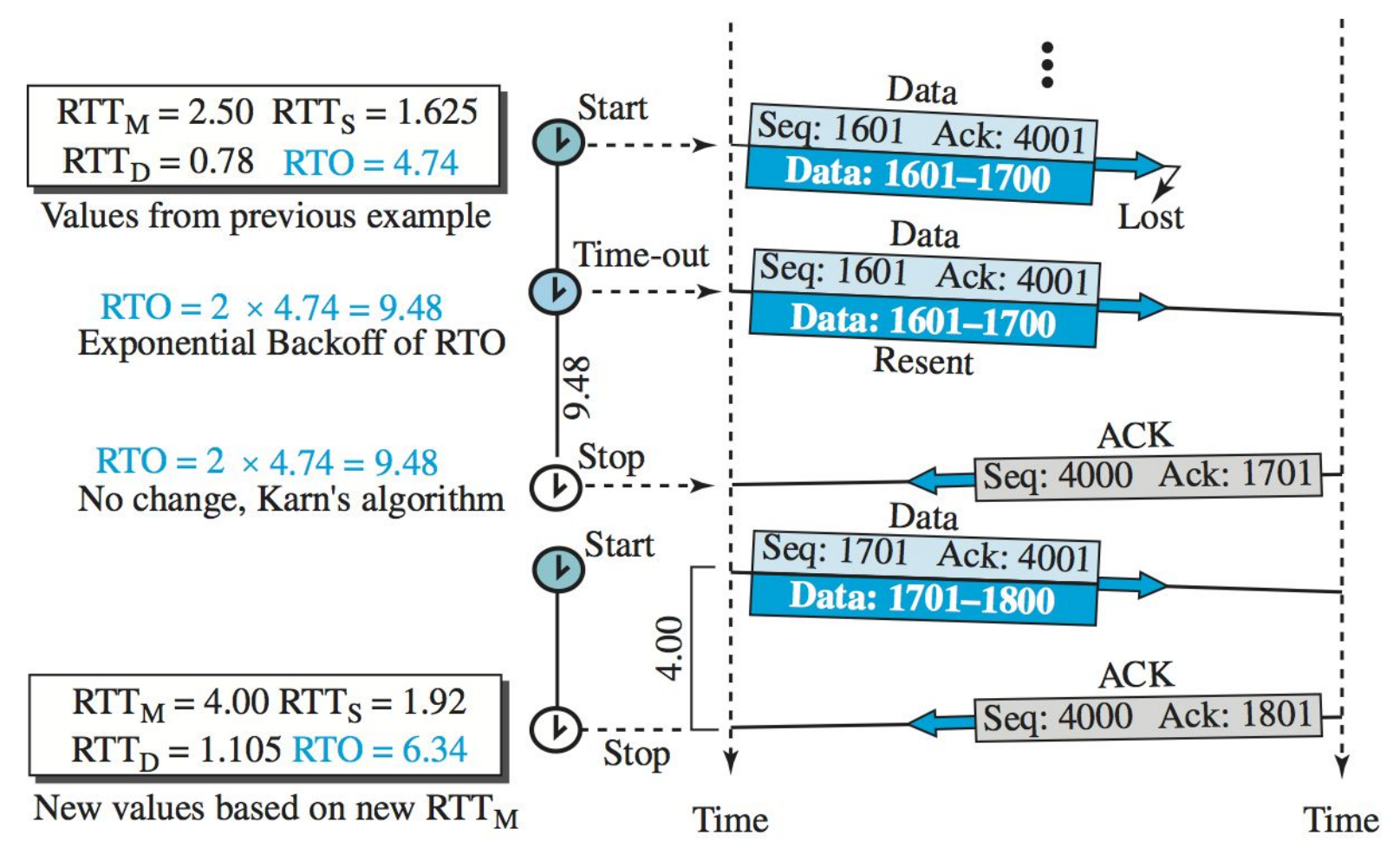
- α=1/8, β=1/4



Temporizador de Retransmisión

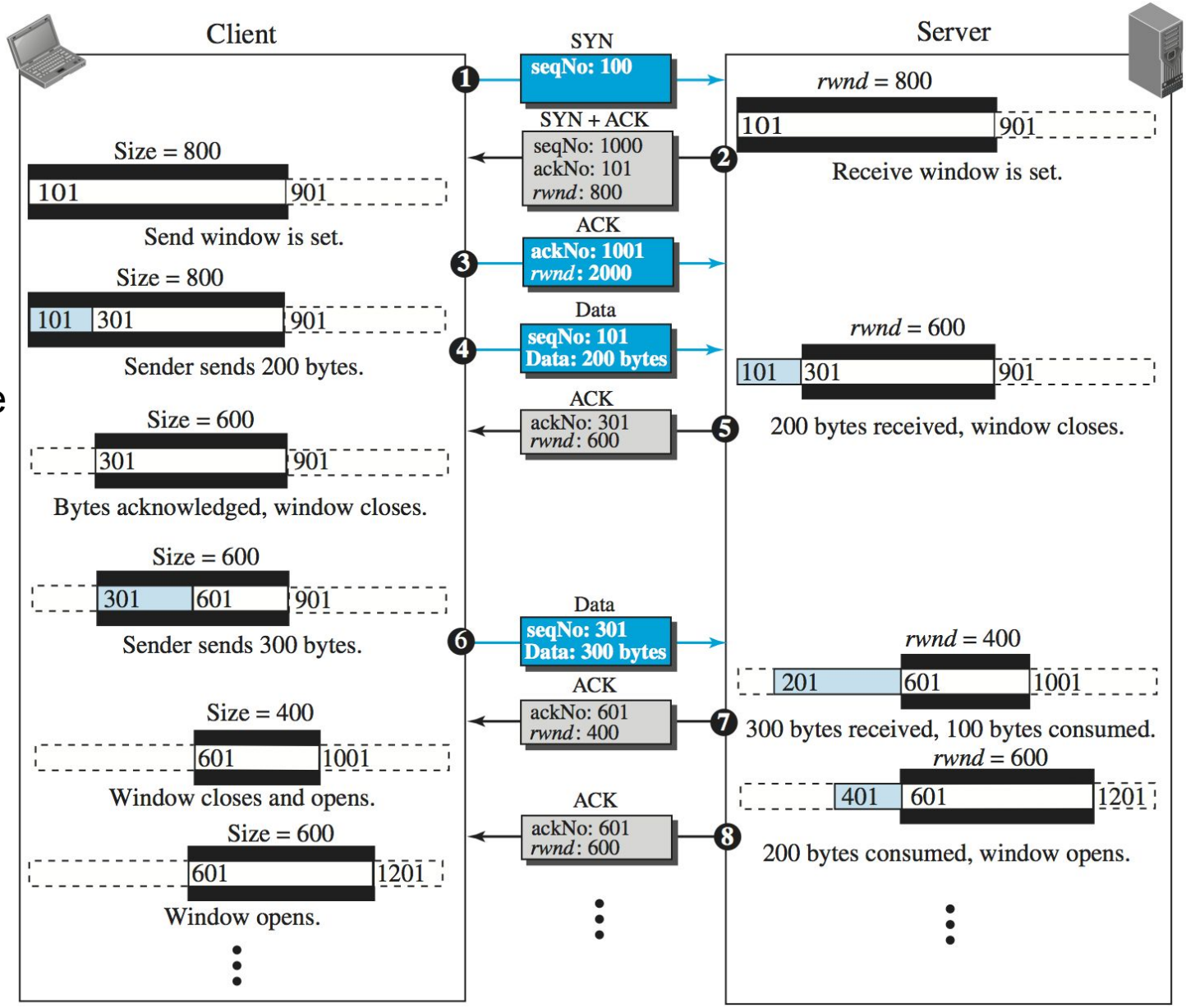
Ejemplo: Calcular los valores de los temporizadores:

- α=1/8, β=1/4



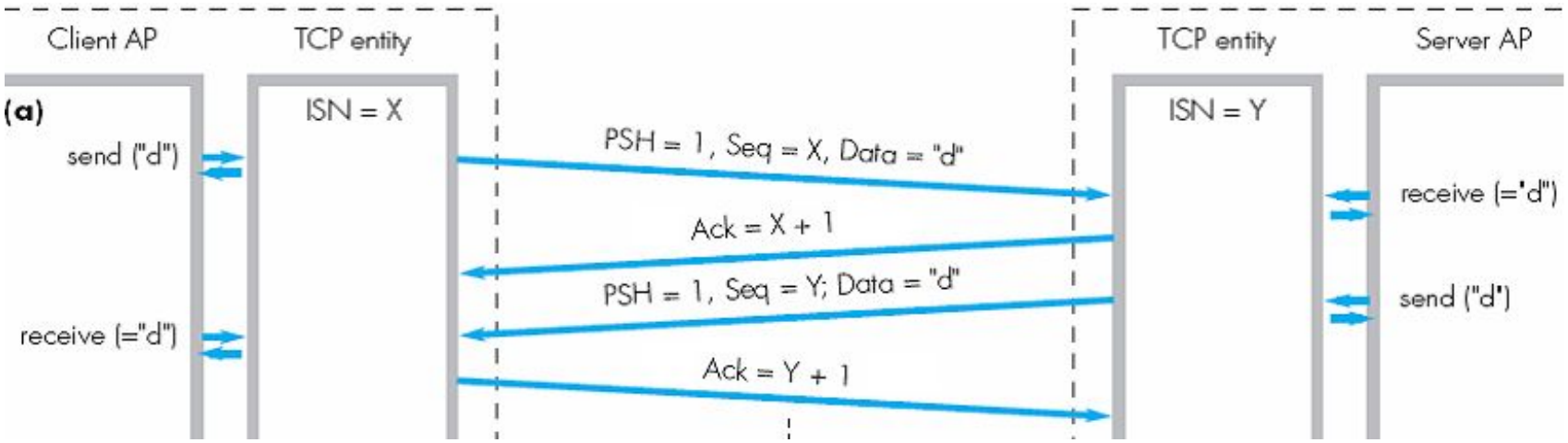
Control de Flujo

- Controla la tasa de envío de datos para evitar la sobrecarga del receptor
- El control de flujo se realiza mediante la ventana de recepción, anunciada en cada ACK



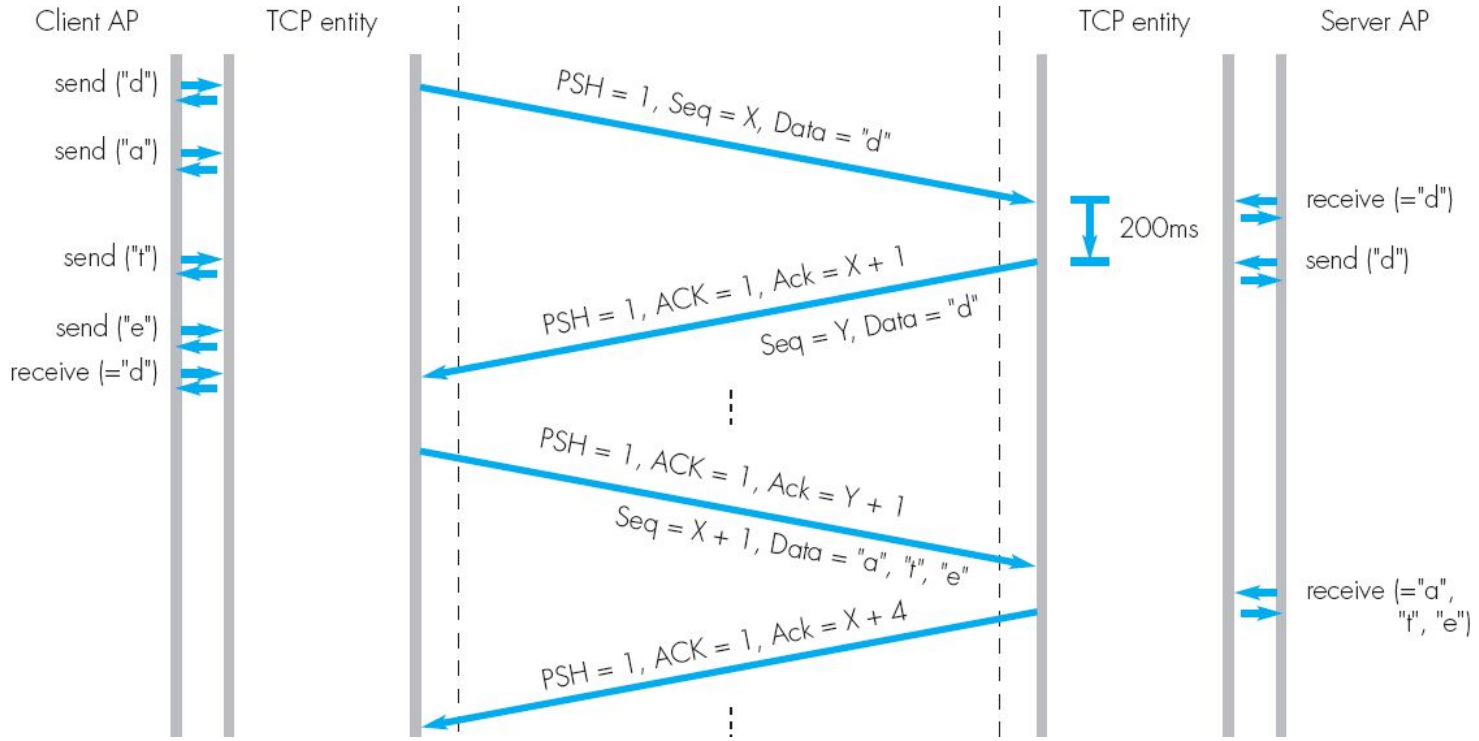
Control de Flujo: Síndrome de la ventana trivial

- El síndrome de la ventana trivial (*silly window syndrome*) se produce cuando:
  - La aplicación emisora genera datos a un ritmo muy lento (ej. byte a byte)
  - La aplicación receptora consume datos a un ritmo muy lento
- **Ventana trivial en el emisor** (ej. aplicaciones interactivas)
  - Cada carácter necesita 4 mensajes TCP/IP (40 bytes de cabeceras)
  - Un carácter (1 bytes) usa más de 160 bytes



Control de Flujo: Síndrome de la ventana trivial

- **Algoritmo de Nagle (RFC 1122, Sec. 4.2.3.4)**
  - El emisor envía el primer mensaje (aunque sea un solo byte)
  - Los siguientes mensajes se retrasan hasta que:
    - se recibe un ACK del receptor
    - se acumula un segmento completo (MSS bytes) de la aplicación
    - expira el RTO

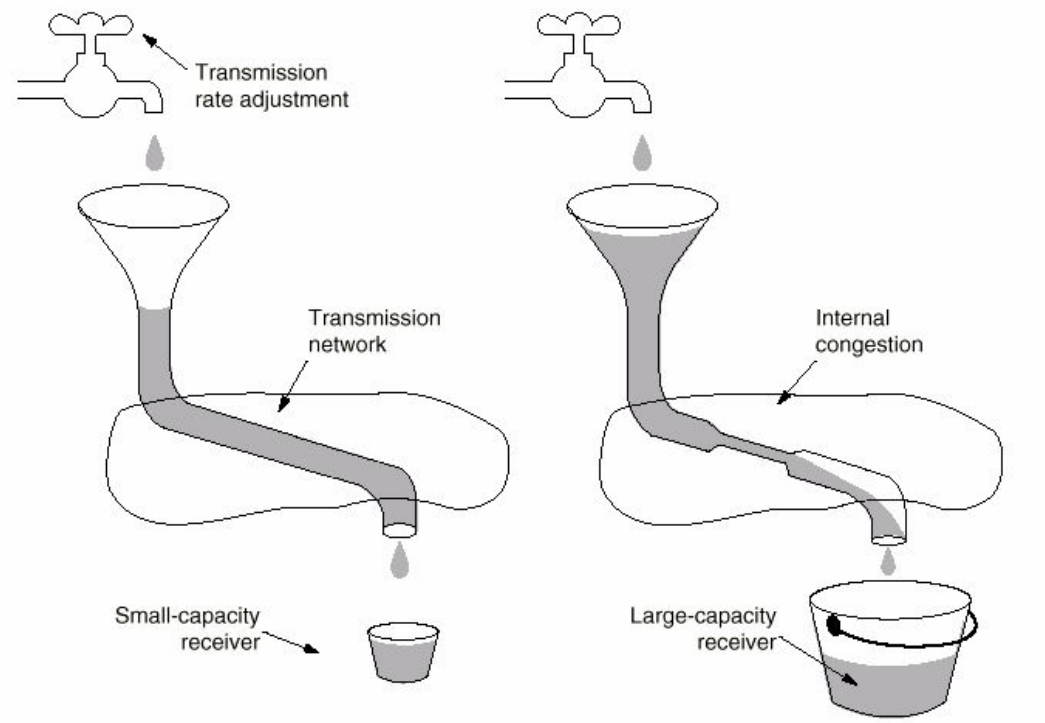


Control de Flujo: Síndrome de la ventana trivial

- **Ventana trivial en el receptor**
  - La aplicación consume los datos a un ritmo lento
  - Se anuncian ventanas de tamaño reducido, produciendo el efecto anterior
- **Algoritmo de Clark**
  - Anunciar un tamaño de ventana 0 hasta que:
    - Haya espacio para recibir un segmento completo (MSS)
    - Se haya liberado la mitad del buffer de recepción
- Retrasar los ACKs
    - Para evitar que el emisor desplace la ventana
    - Reduce el tráfico (número de ACKs), pero puede provocar retransmisiones innecesarias de segmentos no confirmados
    - TCP establece que no deben retrasarse más de 500 ms

Control de la Congestión

- Cuando se pierden paquetes en Internet, la mayoría de las veces se debe a un problema de congestión en algún punto de la red:
  - El router no puede procesar y reexpedir paquetes al ritmo al que los recibe
  - Cuando el router se satura, empieza a descartar paquetes (incluidas las confirmaciones)
- El control de la congestión y el flujo son dos mecanismos diferentes:



Control de la Congestión

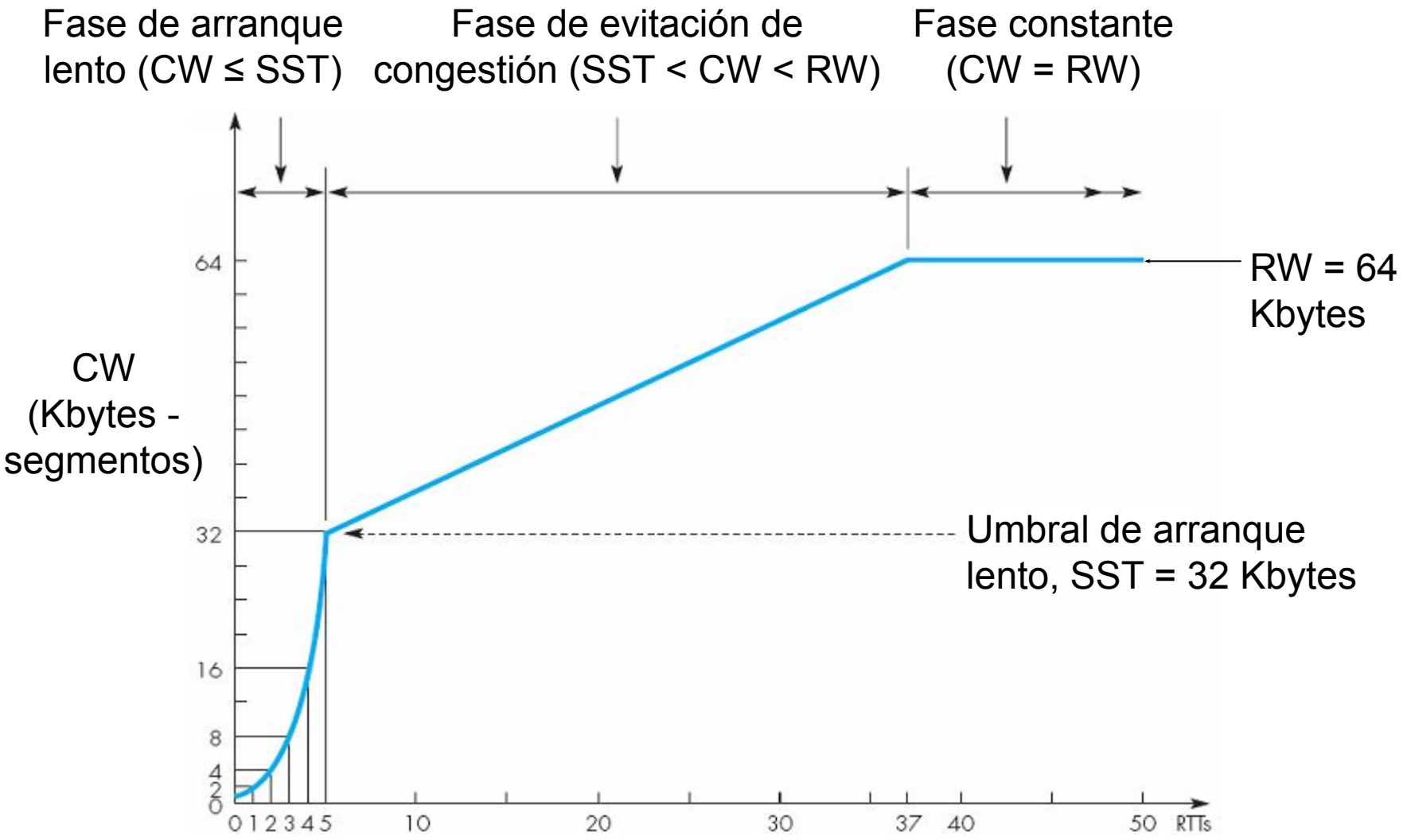
- El emisor utiliza el ritmo de llegada de confirmaciones para regular el ritmo de envío de segmentos de datos
- Esto se implementa mediante la **ventana de congestión (CW)**
  - La ventana de congestión es complementaria a la ventana de recepción (RW) usada para el control de flujo
  - En una situación de no congestión (sin pérdida o retraso de segmentos) la ventana de congestión alcanza el mismo tamaño que la ventana de recepción (CW=RW)
  - Cuando se produce una situación de congestión el tamaño de CW se va reduciendo progresivamente
  - Cuando la situación de congestión desaparece, el tamaño de CW se va aumentando progresivamente
  - El número máximo de bytes que puede enviar el emisor (AW, Allowed Window) es el mínimo de ambos tamaños de ventana:

$$AW = \min \{ RW, CW \}$$

Control de la Congestión

- La red está sin congestión cuando no se pierden o retrasan segmentos
- La transmisión comienza con un tamaño de ventana de congestión CW = 1
  - El emisor envía un único segmento de tamaño máximo igual a MSS
- A continuación, la CW va aumentando, pasando por tres fases distintas:
  - **Fase de arranque lento** (*slow start*)
    - La CW se incrementa en uno por cada segmento enviado y confirmado
    - Esto provoca un crecimiento exponencial (CW = 1, 2, 4, 8, 16, 32…)
    - Esta fase termina cuando el tamaño de CW alcanza un cierto umbral, denominado umbral de arranque lento (SST, *Slow Start Threshold*)
    - Inicialmente, el valor del SST suele ser de 64 Kbytes
  - **Fase de evitación de congestión** (*congestion avoidance*)
    - A partir del SST, la CW se incrementa en uno cada vez que se envía y se confirma una ventana completa (es decir, CW segmentos)
    - Esto provoca un crecimiento lineal
    - Esta fase termina cuando la CW alcanza el tamaño de la ventana de recepción (RW)
  - **Fase constante**
    - En esta fase, la CW se mantiene a un valor constante (CW = RW)

Control de la Congestión

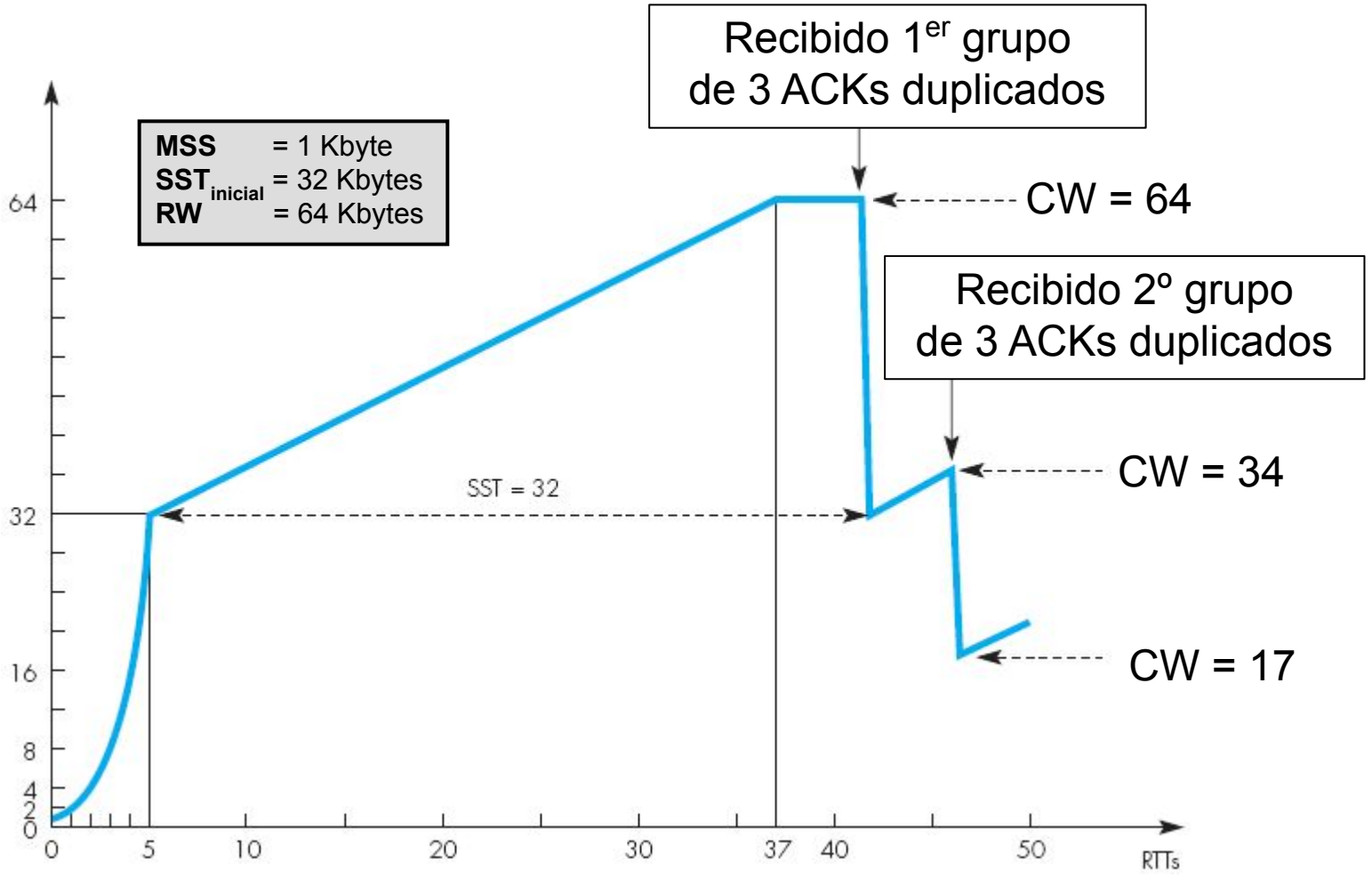


Control de la Congestión

- La situación de congestión en la red se detecta indirectamente
- **Recepción de 3 ACKs duplicados**
  - Nivel de congestión leve, sigue habiendo tráfico en la red (llegan las confirmaciones)
  - Se activa el método de recuperación rápida (*fast recovery*):
    - Se reduce el valor de CW y SST a la mitad del valor de CW
    - Se ejecuta el método de evitación de congestión
- **Expiración del temporizador de retransmisión** (RTO)
  - Nivel de congestión elevado, el tráfico en la red está interrumpido (no llegan confirmaciones)
  - En este caso se realizan las siguientes acciones:
    - Se reduce el valor de SST a la mitad del valor de CW
    - Se inicializa el tamaño de CW a 1
    - Se ejecuta el método de arranque lento

Control de la Congestión

- **Recepción de 3 ACKs duplicados**



Control de la Congestión

- **Expiración del temporizador de retransmisión**