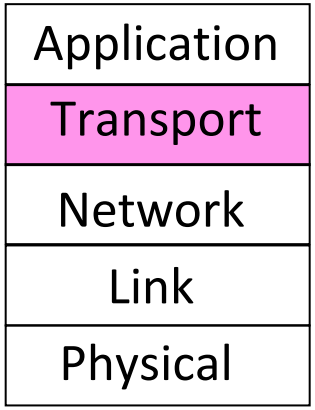
Capítulo 3

Capa de Transporte

Establecimiento de conexiones



Agenda

- **Aprenderemos los siguientes asuntos:**
  1. **Establecimiento de Conexión**
  2. Establecimiento de Conexión en TCP
  3. Liberación de Conexiones
  4. Liberación de conexiones en TCP

Comparación de segmentos

- **Asumimos que**:
  - *T sec es el **tiempo de vida de paquete***
    - Se eliminan paquetes viejos que andan dando vueltas por ahí.
  - *El origen etiqueta los segmentos con n° de secuencia que no van a reutilizarse dentro de T sec.*
  - *El T debería ser lo suficientemente grande como para incluir retransmisión confirmada (i.e. retransmisión y confirmación de la retransmisión) de un paquete.*

Establecimiento de Conexión

- Como al establecer una conexión se usan segmentos, una conexión debería tener un N° inicial de secuencia con el que comienza a operar.
- **Idea**: vincular N° inicial de secuencia de algún modo al tiempo y para medir el tiempo usar un reloj.

Establecimiento de Conexión

- **Implementación de la idea (de Tomlinson)**:
  - Cada host tiene un **reloj de hora del día**.
    - Los relojes de los hosts no necesitan ser sincronizados;
    - se supone que cada reloj es un contador binario que se incrementa a si mismo en intervalos uniformes.
    - **El reloj continua operando aun ante la caída del host**
  - Cuando se establece una conexión los *k* bits de orden mayor del reloj = **número inicial de secuencia**.

Establecimiento de Conexión

- **Para lograr que al regresar al principio de los n° de secuencia, los segmentos viejos con el mismo n° de secuencia hayan desaparecido hace mucho tiempo**
  - el espacio de secuencia debe ser lo suficientemente grande.

Caída de Hosts

- **Problema**: Cuando un host se cae, al reactivarse sus ET no saben dónde estaban en el espacio de secuencia.
- Este es un problema porque para el siguiente segmento a enviar no se sabe qué números de secuencia generar;
  - si se genera mal, entonces el nuevo segmento podría tener el mismo número de secuencia que otro segmento distinto circulando por la red.
- **Solución**: requerir que las ET estén inactivas durante *T* segundos tras una recuperación para permitir que todos los segmentos viejos expiren (entonces no vamos a tener dos segmentos diferentes con el mismo número de secuencia).

Establecimiento de Conexión

- **Problema: ¿Cómo hacer para establecer una conexión entre dos hosts?**
- **Idea**: Para **establecer conexión** el host de origen envía un segmento CONNECTION REQUEST al destino y espera una respuesta CONNECTION ACCEPTED.
- Supongamos que se establecen conexiones haciendo que un host 1 envía segmento *S* = CR *N*, *P* a host 2, donde *N* es n° de secuencia y *P* es n° de puerto.
  - Host 2 confirma ese pedido con segmento CA *N* (connection accept).

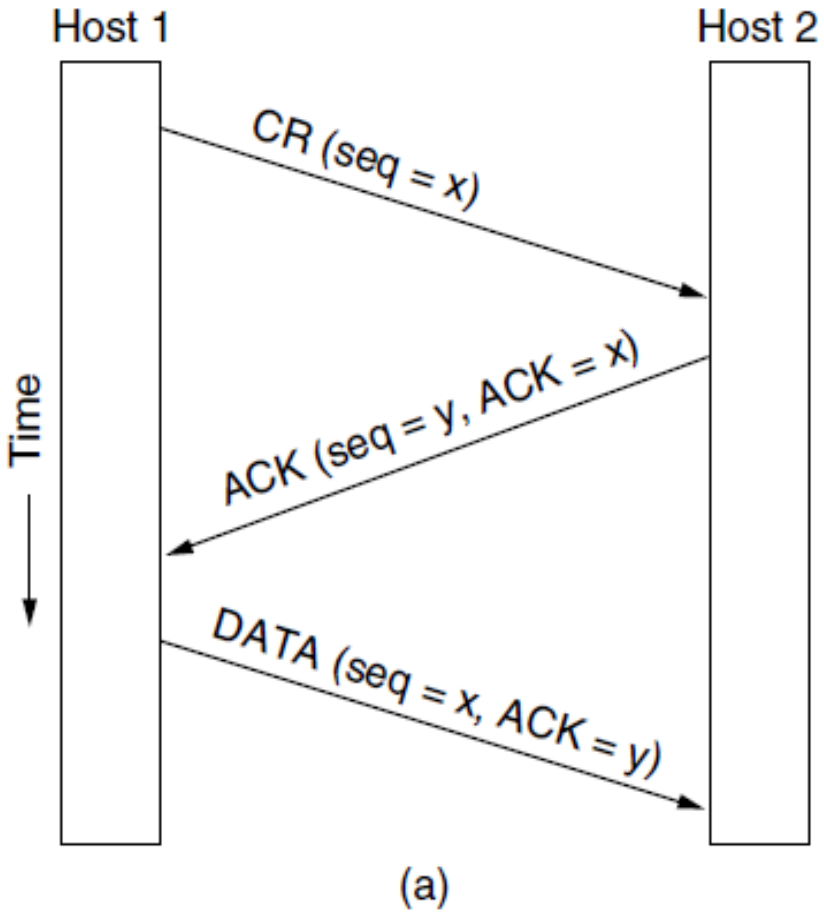
Establecimiento de conexión

- **Caso**: *S* se demora demasiado en llegar a host 2, vence timer en host 1 y host 1 manda un duplicado *S´*= CR, *N*, *P* al host 2.
  - Luego puede pasar que host 2 reciba *S*' y un buen tiempo después *S*.
- **Situación: No se recuerda en el destino n° de secuencias para conexiones.**
- **Problema:** No tenemos forma de saber si un segmento CR conteniendo un n° de secuencia inicial es un duplicado de una conexión reciente o una conexión nueva.
  - No sabe si mandar un segmento CA o no.

Establecimiento de Conexión

Solución: **Acuerdo de tres vías** de Tomlinson de 1975.

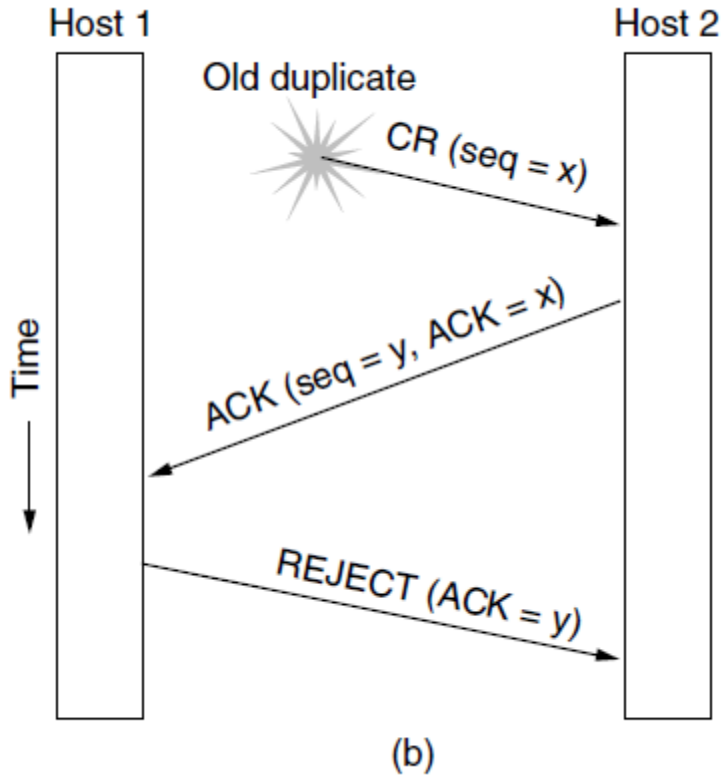
- Caso de operación Normal
- **Fijarse** en el número de secuencia del segmento de datos enviado.
- **¿Cómo sería el caso que llega un segmento CR duplicado al host 2?**



Establecimiento de Conexión

Solución: **Acuerdo de tres vías** de Tomlinson de 1975. Caso de **segmento CR duplicado con retraso**:

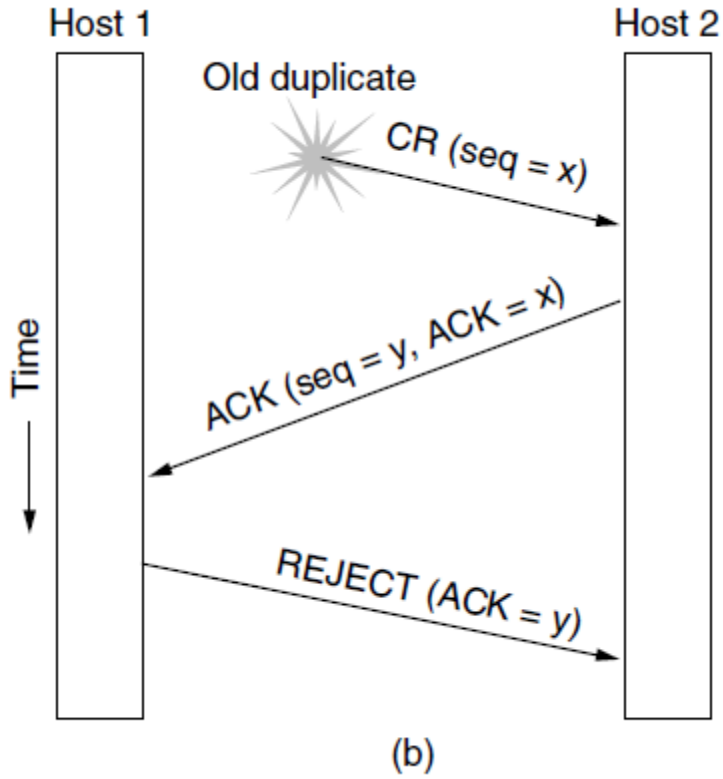
- **¿Qué pasa luego del rechazo del host 1?**



Establecimiento de Conexión

Solución: **Acuerdo de tres vías** de Tomlinson de 1975. Caso de **segmento CR duplicado con retraso**:

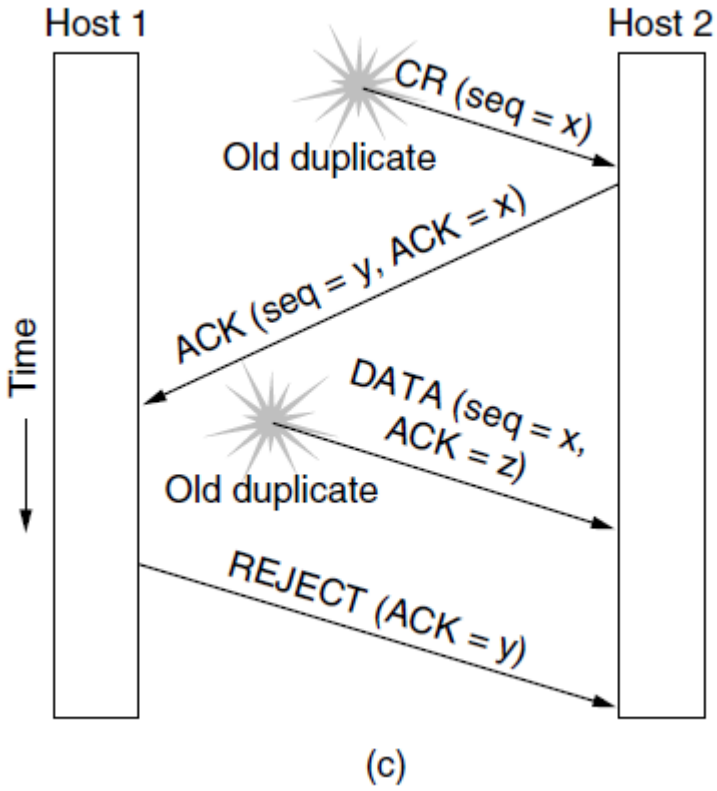
- Al rechazar el host 1 el intento establecimiento de conexión del host 2,
- el host 2 se da cuenta de que fue engañado por un duplicado con retardo y **abandona la conexión**;
- así, un duplicado con retardo no causa daño.



Establecimiento de Conexión

Solución: **Acuerdo de tres vías** de Tomlinson de 1975. Caso de tanto **segmento CR como de datos con retraso**.

- **¿qué significa para el host 2 la llegada del segundo segmento retrasado del host 1?**

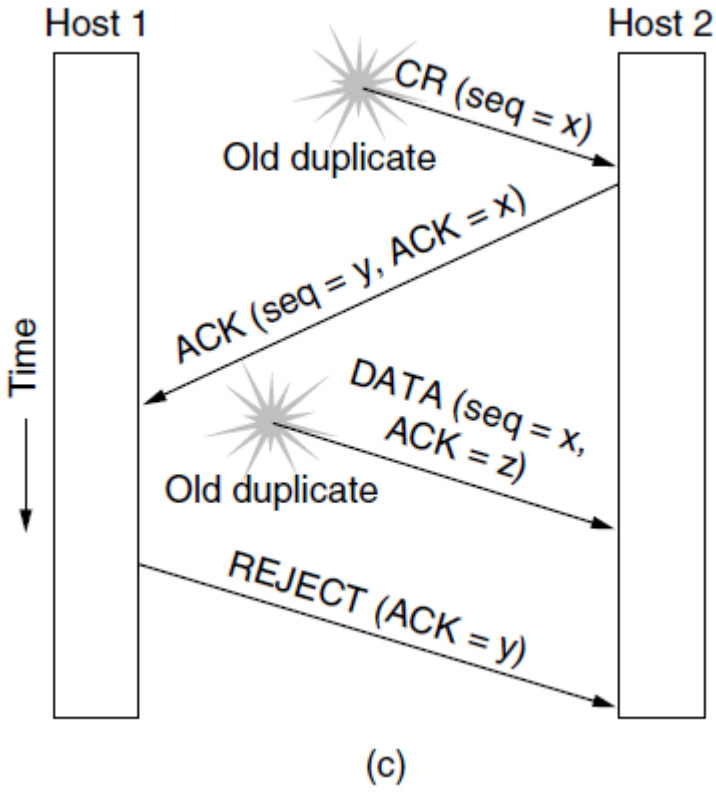


(c)

Establecimiento de Conexión

Solución: **Acuerdo de tres vías** de Tomlinson de 1975. Caso de tanto **segmento CR como de datos con retraso**.

- cuando llega el segundo segmento retrasado al host 2,
- el hecho de que se confirmó la recepción de *z* en lugar de *y* indica al host 2 que este también es un duplicado viejo.



(c)

Agenda

- **Aprenderemos los siguientes asuntos:**
  1. Establecimiento de Conexión
  2. **Establecimiento de Conexión en TCP**
  3. Liberación de Conexiones
  4. Liberación de conexiones en TCP

Establecimiento de una conexión TCP

- El n° de secuencia inicial de una conexión **no es 0**.
  - Se usa un **esquema basado en reloj** con un pulso de reloj cada **4 μsec**.
  - Al caerse un host, no podrá reiniciarse durante el **tiempo máximo de paquete** (120 seg),
  - para asegurar que no haya paquetes de conexiones previas vagando por Internet.

Establecimiento de una conexión TCP

- **Campos del encabezado TCP para el establecimiento de conexiones**
- **SYN** se usa para establecer conexiones.
  - Solicitud de conexión: SYN = 1 y ACK = 0.
  - La respuesta de conexión sí lleva una confirmación de recepción, por lo que tiene SYN = 1 y ACK = 1.
    - Recordar que además hay campo con N° de secuencia confirmado.

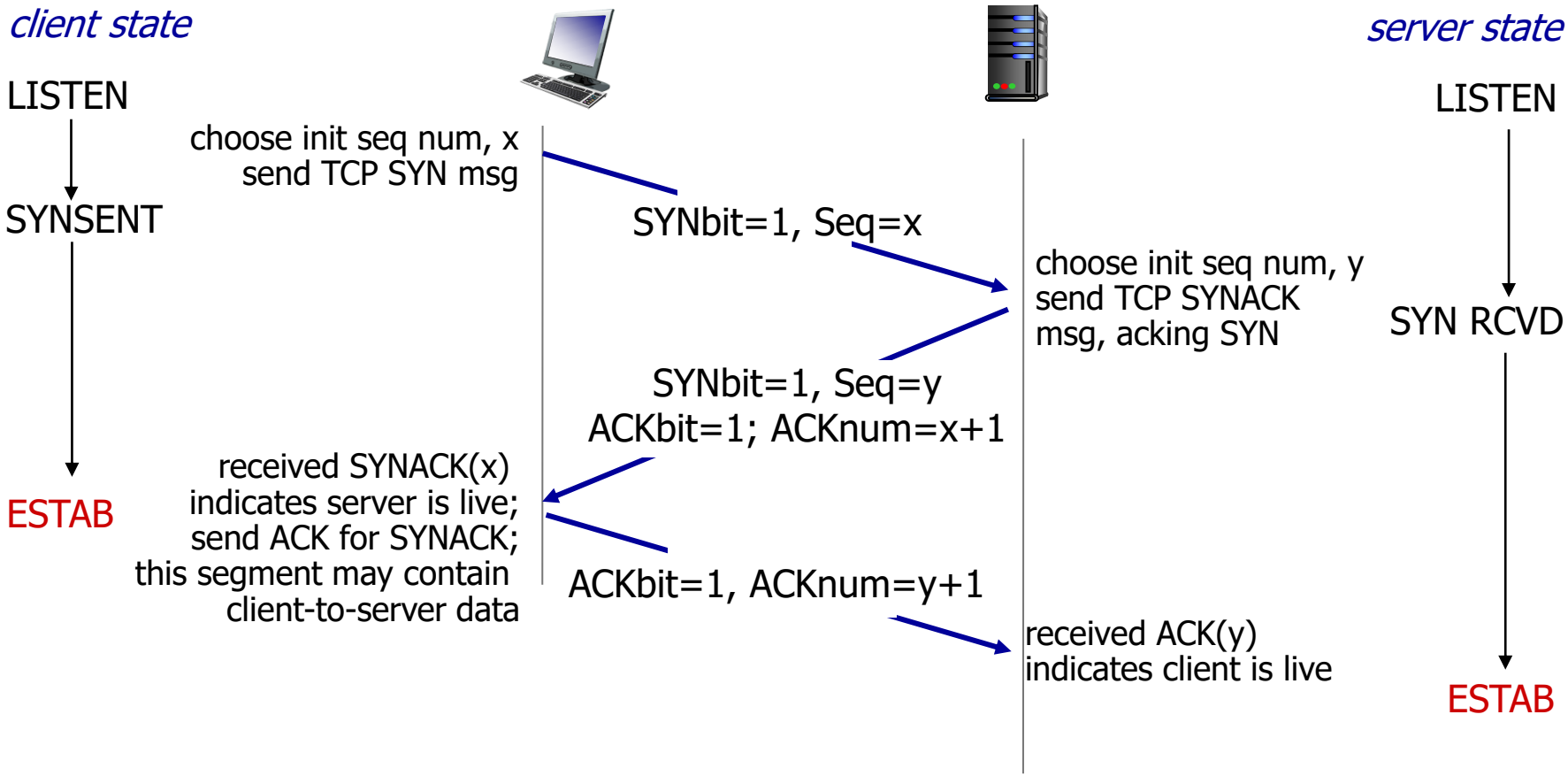
Establecimiento de una conexión TCP

- En TCP las conexiones usan el **acuerdo de 3 vías**
  1. Para establecer una conexión, el servidor, espera pasivamente una conexión entrante ejecutando LISTEN y ACCEPT y
     - especificando cierto origen o bien nadie en particular.
  2. En el lado del cliente ejecuta CONNECT
     - la cual envía un segmento TCP con el **bit SYN encendido y el bit ACK apagado**, y espera una respuesta.

Establecimiento de una conexión TCP

3. Al llegar el segmento al destino, la ETCP allí revisa si **hay un proceso** que haya ejecutado un LISTEN en el puerto indicado en el campo puerto de destino.
4. Si no lo hay envía una respuesta con el **bit RST encendido** para **rechazar la conexión**.
5. Si algún proceso está escuchando en el puerto ese proceso recibe el segmento TCP entrante y puede entonces aceptar o rechazar la conexión; si la acepta se envía un segmento de ack.
6. La secuencia de segmentos TCP enviados en el caso normal se muestra en la Figura siguiente.

Establecimiento de una conexión TCP



Establecimiento de una conexión TCP

- **Ejercicio**: El campo de números de secuencia en el encabezado TCP es de 32 bits de largo,
  - lo cual es suficientemente largo para cubrir 4 billones de bytes de datos.
  - Incluso si tantos bytes nunca fueran transferidos por una conexión única, ¿por qué puede el número de secuencia pasar de $2^{32} - 1$ a 0?
  - Tener en cuenta en la figura de la filmina anterior.