

1. Considerad una conexión TCP entre un cliente (C) y un servidor(S). Suponed que S envía un segmento con el bit ACK activado y SN=2000, el cual, se pierde. ¿El protocolo TCP incluye algún mecanismo para detectar la pérdida de un ACK? Razonad vuestra respuesta.

Supongamos que hay activa una transferencia de datos de C->S y que, regularmente S responde a C enviando los ACKs correspondientes, uno de los cuales, se pierde. Si los ACKs que siguen al ACK perdido siguen avanzando (snd.una creciente), el receptor no apreciará ningún problema, ya que los ACKs son acumulados; por tanto, no será necesaria retransmisión alguna.

Supongamos, que el canal C->S detiene la transferencia de datos durante un periodo de tiempo relativamente largo; asumamos que el ACK perdido es el último ACK enviado antes de detener las transferencias, correspondiente al último segmento enviado por C->S. Puesto que por el momento, C no va a recibir ACKs acumulados que cubran el ACK perdido, el timer RTO disparará en C provocando la retransmisión desde snd.una hasta el final de la ventana actual.

2. En redes IP ¿Por qué se pierden los paquetes? ¿Cómo protege el TCP contra las pérdidas de paquetes?

Los paquetes se pueden perder como consecuencia de un número excesivo de errores en algún enlace o de fallos en la electrónica o en los medios físicos empleados en la transmisión.

Los paquetes también se pueden perder cuando la carga ofrecida sobre la red (Switches o Routers) es excesiva, lo cual, provoca el desbordamiento de alguna de las colas de transmisión de switches y routers. Un transmisor TCP detecta la pérdida de algún paquete cuando recibe varios ACKs por el mismo ACK SN (Es decir, que snd.una no avanza). TCP protege de las pérdidas de paquetes reenviando los bytes incluidos en los segmentos perdidos, de nuevo. El reenvío se lleva a cabo como consecuencia de la recepción de 3 ACKs duplicados (3 acks por el mismo extremo snd.una) o, si el segmento perdido es uno de los últimos enviados y no quedan suficientes para generar 3DUP, es el RTO timer el que hará que el transmisor reenvie los bytes/segmentos mencionados.

Para proteger los recursos de la red, cuando tiene lugar alguna pérdida de paquetes, el TCP reacciona disminuyendo su carga ofrecida hacia la red en la conexión TCP en la que tuvo lugar la pérdida del paquete. De este modo, se reducirá la carga agregada sobre el router o switch que constituye el cuello de botella, de forma que, en algún momento se recuperará y dejará de perder paquetes si los otros flujos TCP que cruzan el cuello de botella, reducen su carga ofrecida también. Esta es la forma en que el TCP detecta congestión y la forma en que trata de mitigarla.

El TCP no es capaz de distinguir la causa de las pérdidas de paquetes (Las ocasionadas por un enlace defectuoso o aquellas derivadas del exceso de demanda sobre la red) y reacciona, en ambos casos aplicando los algoritmos de control de Gestión, lo cual, no es acertado en el primer caso, porque, el hecho de reducir la demanda sobre la red, no puede reparar el enlace defectuoso.

3. Asumid una conexión TCP que se encuentra en el estado SS (Slow Start) y que no hay ninguna pérdida ni duplicación de paquetes ni éstos llegan desordenados. En el Rtt actual, el transmisor envía 4 segmentos de tamaño *full* (MSS = 1448 Bytes), el SN incluido en el primero de ellos es 10000. A continuación, el receptor envía un segmento con el bit ACK activado y con ACK sequence number = 14344.

- a. Razonad cuántos segmentos enviará el transmisor después de recibir este segmento con el bit ACK activado

Segmento 1: [10000, 11447]

Segmento 2: [11448, 12895]

Segmento 3: [12896, 14343] -> ACK 14344

Segmento 4: [14344, 15791] -> ACK 15792

ACK 14344 es un *stretch* ACK que cubre tres segmentos seguidos (Avanza tres MSS), el transmisor podrá enviar  $3 \times 2 = 6$  segmentos, ya que se encuentra en SS

- b. Explicad los campos más relevantes del ACK devuelto por el receptor después de recibir el último de los cuatro segmentos ACK activado con ACK SN = 15792. El resto de campos, no resultan relevantes a esta pregunta.

4. Explicad el algoritmo de control de congestión empleado en la versión TCP Reno.

Key words: RTO fires, 3-DUP, sstrsh, cwnd, acks that advance snd.una, SS(exp), CA(lin), Fast Retransmit, Fast Recovery

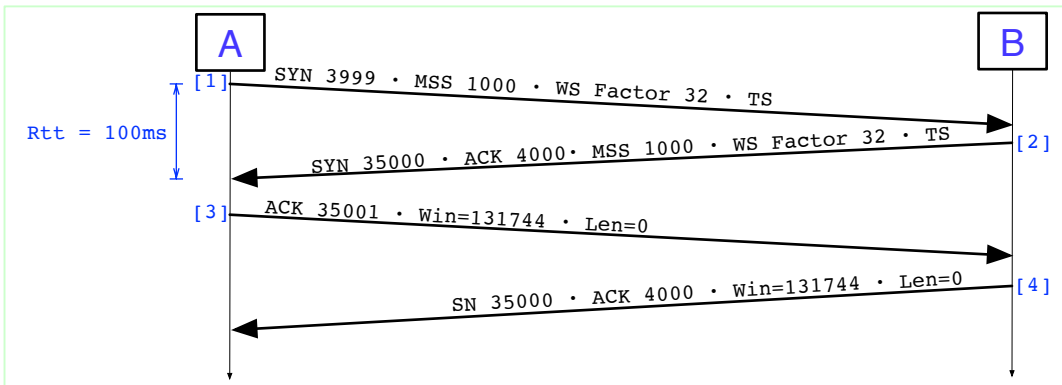
5. [ M ] El cliente (C) de una conexión TCP recibe un segmento procedente del servidor (S), el cual, contiene los siguientes campos: ACK activado, ACK SN = 2000, AWS = 0. Marcad las afirmaciones verdaderas:

- a. El espacio libre del buffer de recepción de S no puede ser 0
- b. El tamaño de la ventana de control de flujo anunciada por C es 0
- c. El espacio de buffer de recepción de S puede ser 0

- d. El tamaño efectivo de la ventana de control de flujo usable por C es 0
  - e. El tamaño de la ventana de control de congestión anunciada por S es 0
6. [ M ] Considerad el mecanismo de control de flujo TCP y el canal formado por el transmisor (C) y el receptor (S) de una conexión TCP que involucra a los hosts C y S. Asumid que el último segmento ACK enviado por el receptor (S) tiene los siguientes campos: ACK 1001, WS 1000 y no usa ninguna opción TCP ¿Cuáles de las siguientes opciones representan mensajes válidos de acuerdo con el protocolo TCP, los cuales, por tanto, sí podrían ser enviados por el transmisor (C) en el escenario explicado?
- a. Segmento de 498 Bytes con SN = 1500, ACK =1851, WS = 699
  - b. Segmento de 0 Bytes con SN = 1000, ACK =3200, WS = 700
  - c. Segmento de tamaño 1001 Bytes con SN = 1001, ACK =1249, WS = 0
  - d. Segmento de 20 Bytes con SN = 1982, ACK =1850, WS = 700
  - e. Segmento de tamaño 1001 Bytes con SN = 1000, ACK =1850, WS = 0
  - f. Segmento de 0 Bytes con SN = 2001, ACK =3200, WS = 700

Fig. 2

7. Asumid el contexto inicial de la fig. 2 para esta pregunta. Suponed que, con posterioridad al handshake allí representado, tienen lugar las transmisiones de la fig. 3. Observad que los valores de los campos TSVa(Timestamp value) y TSecr(Timestamp Echo Reply) son los números en coma fija X0 a X7 (Los valores concretos no son relevantes, por ello, los representamos simbólicamente).



- a. ¿Cuánto vale X5 en función de los valores anteriores de TSVa y TSecr?  
 The earliest TSVa of [1] and [2], therefore:  
 $X5 < X0$ ;
- b. ¿Cómo calcula el host A que la muestra RttSample actual es 100ms en el instante [4] (Fig. 3)?  
 $100ms = TimeOfDay - X5 = TimeOfDay - X0$
- c. Justo antes de [4] el valor del EstimatedRtt es de 91,2 ms ¿Qué tiempo programará A para el RTO (Retransmission Timeout)? Asumid que el TCP ejecuta el algoritmo de Karn-Partridge con  $\alpha = 0,9$ .  
 $91,2ms * 0,9 + 100ms * 0,1 = 92,08ms$   
 $RTO = 2 * 92,08ms = 184,16ms$
- d. ¿En qué instante el host A arrancará el RT (Retransmission Timer), en [4] o en [5]?  
 RTO reset at [4] with an RTO = 184,16ms; in [5] it will not be necessary to start the timer since it is already running (It was just reset at [4]).
- e. Asumid que el segmento [5] se pierde y que A no transmite más segmentos con datos a partir de [5]. Razonad si la retransmisión del segmento [5] tiene lugar como resultado de un 3-DUP (3 Duplicates) o un RTO (Retransmission Timeout).  
 RTO, not enough ensuing ACKs for 3-DUP.

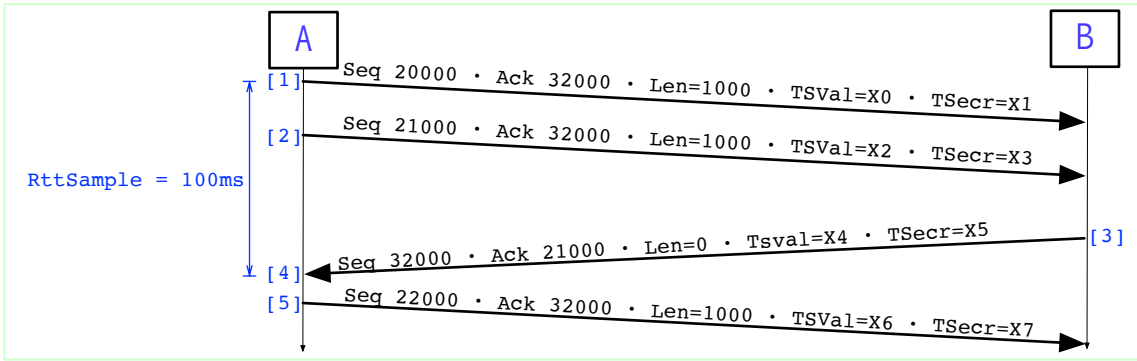


Fig. 3

8. Asumiendo el contexto inicial de la Fig. 2, un TCP Reno y que no se pierde ningún segmento, respondida a las siguientes preguntas relativas al cronograma adjunto (Fig. 4).
  - a. ¿Está al algoritmo de Nagle activado en A?  
 No, en un Rtt ha enviado varios partial segments sin necesidad de esperar al último ACK antes de enviar un seg parcial.
  - b. ¿Y en B?  
 No contamos con suficiente información, ya que no vemos transmisiones de B
  - c. ¿Son correctos los números de secuencia?  
 Si partimos de 4000, la secuencia de SNs es coherente con la longitud de cada segmento
  - d. ¿En cuántos de los segmentos de la fig. 4 no está activado el flag ACK?  
 En ninguno; todos deben tener el ACK activado
  - e. Indicad los números de ACK correspondientes a la recepción correcta de los dos primeros segmentos  
 ACK = 4560  
 ACK = 5120
  - f. ¿En cuántos de los segmentos de la fig.4 está activado el flag SYN?  
 No puede estar activado en ninguno ya que el 3-way handshake ya tuvo lugar (Fig.2)

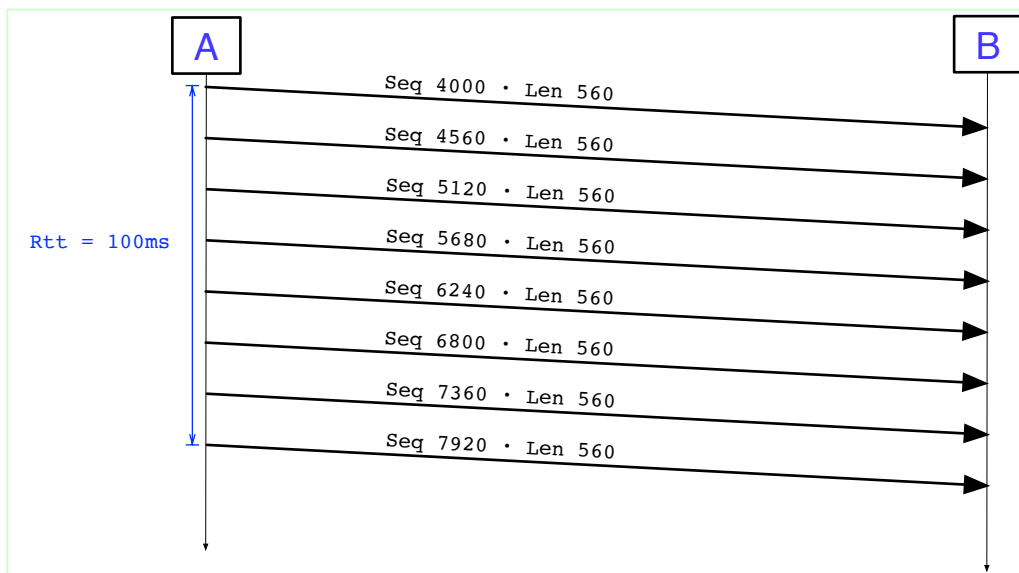


Fig. 4

9. Comentario sobre la última tutoría: *Las opciones del TCP que sirven para muestrear el Rtt a la llegada de cada ACK, se denominan TSVal y TSecr. Cuando el TCP transmite un nuevo segmento, copia su hora interna (Llamémosla TimeOfDay) en el campo TSVal. Cuando el TCP receptor recibe este segmento, procede a copiar el campo TSVal en el campo TSecr del segmento ACK que va a enviar. Cuando el ACK llega al transmisor original, éste calcula el Rtt haciendo la diferencia TimeOfDay – TSecr.*

