

Introducción a las Redes de Computadoras

Práctico 3

P1. Suponga que el cliente A inicia una sesión Telnet con el servidor S. Aproximadamente en el mismo instante, el cliente B también inicia una sesión Telnet con el servidor S. Proporcione los posibles números de puerto de origen y de destino para:

- Los segmentos enviados de A a S.
- Los segmentos enviados de B a S.
- Los segmentos enviados de S a A.
- Los segmento enviados de S a B.
- Si A y B son hosts diferentes, ¿es posible que el número de puerto de origen en los segmentos que van de A a S sea el mismo que en los segmentos que van de B a S?
- ¿Qué ocurre si A y B son el mismo host?

P2. Considere la Figura 3.5. ¿Cuáles son los valores de los puertos de origen y de destino en los segmentos que fluyen desde el servidor de vuelta a los procesos cliente? ¿Cuáles son las direcciones IP de los datagramas de la capa de red que transportan los segmentos de la capa de transporte?

P3. UDP y TCP utilizan el complemento a 1 para calcular sus sumas de comprobación. Suponga que tiene los tres bytes de 8 bits siguientes: 01010011, 01010100, 01110100. ¿Cuál es el complemento a 1 de la suma de estos bytes? (Observe que aunque UDP y TCP utilizan palabras de 16 bits para calcular la suma de comprobación, en este problema le pedimos que considere sumas de 8 bits). Explique cómo funciona. ¿Por qué UDP utiliza el complemento a 1 de la suma; es decir, por qué no simplemente emplea la suma? Con el esquema del complemento a 1, ¿cómo detecta el receptor los errores? ¿Es posible que un error de un solo bit no sea detectado? ¿Qué ocurre si hay 2 bits erróneos?

P4. a. Suponga que tiene los 2 bytes siguientes: 01011100 y 01010110. ¿Cuál es el complemento a 1 de la suma de estos 2 bytes?

b. Suponga que tiene los 2 bytes siguientes: 11011010 y 00110110. ¿Cuál es el complemento a 1 de la suma de estos 2 bytes?

c. Para los bytes del apartado (a), proporcione un ejemplo en el que un bit cambie de valor en cada uno de los 2 bytes y aún así el complemento a 1 no varíe.

P5. Suponga que el receptor UDP calcula la suma de comprobación de Internet para el segmento UDP recibido y comprueba que se corresponde con el valor almacenado en el campo de suma de comprobación. ¿Puede el receptor estar completamente seguro de que no hay ningún bit erróneo? Explique su respuesta.

P6. Recuerde el motivo de corregir el protocolo `rdt2.1`. Demuestre que el receptor mostrado en la Figura 3.57 y el emisor mostrado en la Figura 3.11 pueden llegar a entrar en un estado de bloqueo tal que cada uno de ellos esté esperando a que se produzca un suceso que no ocurrirá nunca.

P7. En el protocolo `rdt3.0`, los paquetes ACK que fluyen del receptor al emisor no tienen números de secuencia (aunque tienen un campo ACK que contiene el número de secuencia del paquete que están reconociendo). ¿Por qué estos paquetes ACK no requieren números de secuencia?

P8. Dibuje la máquina de estados finitos correspondiente al lado receptor del protocolo `rdt3.0`.

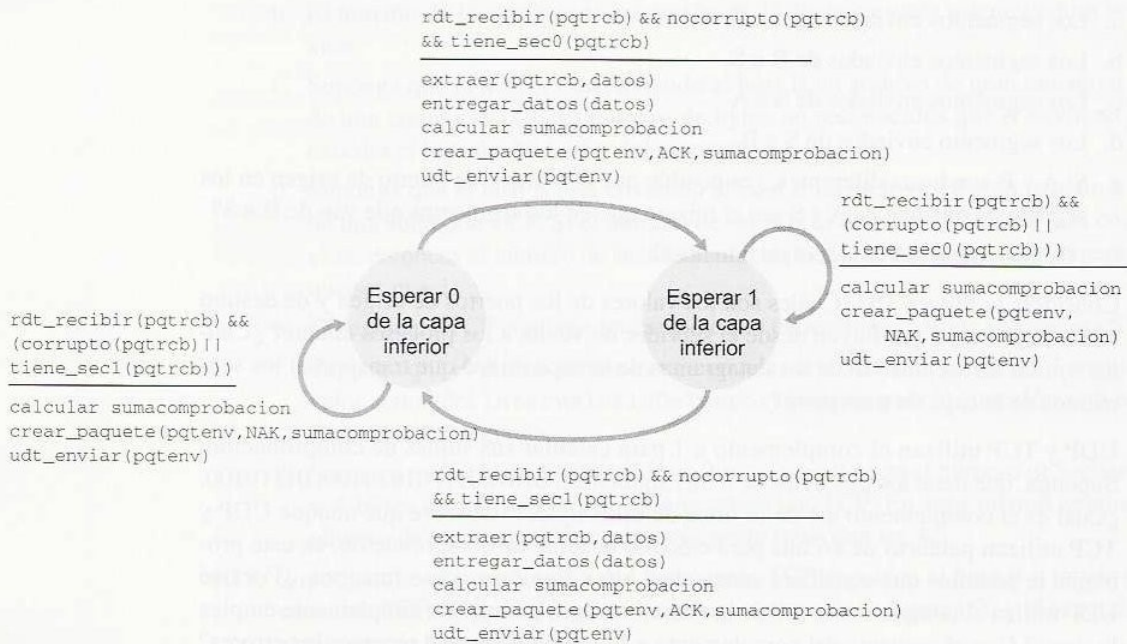


Figura 3.57 • Un receptor incorrecto para el protocolo rdt 2.1.

- P9. Dibuje un esquema que muestre la operación del protocolo rdt3.0 cuando los paquetes de datos y los paquetes de reconocimiento están corrompidos. Utilice un esquema similar al mostrado en la Figura 3.16.
- P10. Sea un canal que puede perder paquetes pero del que se conoce su retardo máximo. Modifique el protocolo rdt2.1 para incluir los fines de temporización y las retransmisiones del emisor. Argumente de manera informal por qué su protocolo puede comunicarse correctamente a través de este canal.
- P11. El lado del emisor de rdt3.0 simplemente ignora (es decir, no realiza ninguna acción) todos los paquetes recibidos que contienen un error o que presentan un valor erróneo en el campo número de reconocimiento (acknum) de un paquete de reconocimiento. Suponga que, en tales circunstancias, rdt3.0 simplemente retransmite el paquete de datos actual. ¿Funcionaría en estas condiciones el protocolo? (*Sugerencia:* piense en lo que ocurriría si sólo hubiera errores de bit; no se producen pérdidas de paquetes pero sí pueden ocurrir sucesos de fin prematuro de la temporización. Considere cuántas veces se envía el paquete n , cuando n tiende a infinito.)
- P12. Considere el protocolo rdt 3.0. Dibuje un diagrama que muestre que si la conexión de red entre el emisor y el receptor puede reordenar los mensajes (es decir, que dos mensajes que se propagan por el medio físico existente entre el emisor y el receptor pueden ser reordenados), entonces el protocolo de bit alternante no funcionará correctamente (asegúrese de identificar claramente el sentido en el que no funcionará correctamente). En el diagrama debe colocar el emisor a la izquierda y el receptor a la derecha, con el eje de tiempos en la parte inferior de la página

y deberá mostrar el intercambio de los mensajes de datos (D) y de reconocimiento (A). No olvide indicar el número de secuencia asociado con cualquier segmento de datos o de reconocimiento.

P13. Considere un protocolo de transferencia de datos fiable que sólo utiliza paquetes de reconocimiento negativo. Imagine que el emisor envía datos con muy poca frecuencia. ¿Sería preferible un protocolo que sólo emplea paquetes NAK a uno que utilice paquetes ACK? ¿Por qué? Suponga ahora que el emisor tiene muchos datos que transmitir y que la conexión terminal a terminal experimenta muy pocas pérdidas. En este segundo caso, ¿sería preferible un protocolo que sólo emplee paquetes NAK a otro que utilice paquetes ACK? ¿Por qué?

P14. Considere el ejemplo mostrado en la Figura 3.17. ¿Cuál tiene que ser el tamaño de la ventana para que la tasa de utilización del canal sea mayor del 95 por ciento? Suponga que el tamaño de un paquete es de 1.500 bytes, incluyendo tanto los campos de cabecera como los datos.

P15. Suponga que una aplicación utiliza el protocolo rdt 3.0 como su protocolo de la capa de transporte. Como el protocolo de parada y espera tiene una tasa de utilización del canal muy baja (como se ha demostrado en el ejemplo de conexión que atraviesa el país de costa a costa), los diseñadores de esta aplicación permiten al receptor devolver una serie (más de dos) de ACK 0 y ACK 1 alternantes incluso si los correspondientes datos no han llegado al receptor. ¿Debería este diseño aumentar la tasa de utilización del canal? ¿Por qué? ¿Existe algún problema potencial con esta técnica? Explique su respuesta.

P16. En el protocolo SR genérico que hemos estudiado en la Sección 3.4.4, el emisor transmite un mensaje tan pronto como está disponible (si se encuentra dentro de la ventana) sin esperar a recibir un paquete de reconocimiento. Suponga ahora que deseamos disponer de un protocolo SR que envíe mensajes de dos en dos. Es decir, el emisor enviará una pareja de mensajes y enviará la siguiente pareja de mensajes solo cuando sepa que los dos mensajes de la primera pareja se han recibido correctamente.

Suponga que el canal puede perder mensajes pero no corromperlos ni tampoco reordenarlos. Diseñe un protocolo de control de errores para un servicio de transferencia de mensajes fiable y unidireccional. Proporcione una descripción de las máquinas de estados finitos del emisor y del receptor. Describa el formato de los paquetes intercambiados por el emisor y el receptor. Si utiliza alguna llamada a procedimiento distinta de las empleadas en la Sección 3.4 (por ejemplo, `udt_enviar()`, `iniciar_temporizador()`, `rdt_recibir()`, etc.), defina claramente las acciones que realizan. Proporcione un ejemplo (una gráfica temporal del emisor y del receptor) que muestre cómo este protocolo se recupera de la pérdida de paquetes.

P17. Considere un escenario en el que el host A desea enviar simultáneamente paquetes a los hosts B y C. El host A está conectado a B y C a través de un canal de multidifusión (*broadcast*) (un paquete enviado por A es transportado por el canal tanto a B como a C). Suponga que el canal de multidifusión que conecta A, B y C puede perder y corromper de manera independiente los paquetes (es decir, puede ocurrir, por ejemplo, que un paquete enviado desde A llegue correctamente a B, pero no a C). Diseñe un protocolo de control de errores similar a un protocolo de parada y espera que permita

transferir paquetes de forma fiable de A a B y C, de manera que A no obtendrá nuevos datos de la capa superior hasta que separe que tanto B como C han recibido correctamente el paquete actual. Proporcione las descripciones de las máquinas de estados finitos de A y C. (*Sugerencia:* la FSM de B será prácticamente la misma que la de C.) Proporcione también una descripción del formato o formatos de paquete utilizados.

- P18. Considere un escenario en el que el host A y el host B desean enviar mensajes al host C. Los hosts A y C están conectados mediante un canal que puede perder y corromper (pero no reordenar) los mensajes. Los hosts B y C están conectados a través de otro canal (independiente del canal que conecta a A y C) que tiene las mismas propiedades. La capa de transporte del host C tiene que alternar la entrega de los mensajes que A y B tienen que pasar a la capa superior (es decir, primero entrega los datos de un paquete de A y luego los datos de un paquete de B, y así sucesivamente). Diseñe un protocolo de control de errores de tipo parada y espera para transferir de forma fiable los paquetes de A y B a C, con una entrega alternante en el host C, como hemos descrito anteriormente. Proporcione las descripciones de las FSM de A y C. (*Sugerencia:* la FSM de B será prácticamente la misma que la de A.) Proporcione también una descripción del formato o formatos de paquete utilizados.
- P19. Sea un protocolo GBN con una ventana de emisor de 3 y un rango de números de secuencia de 1.024. Suponga que en el instante t el siguiente paquete en orden que el receptor está esperando tiene el número de secuencia k . Suponga que el medio de transmisión no reordena los mensajes. Responda a las siguientes cuestiones:
- ¿Cuáles son los posibles conjuntos de números de secuencia que pueden estar dentro de la ventana del emisor en el instante t ? Justifique su respuesta.
 - ¿Cuáles son todos los valores posibles del campo ACK en todos los posibles mensajes que están actualmente propagándose de vuelta al emisor en el instante t ? Justifique su respuesta.
- P20. Suponga que tenemos dos entidades de red, A y B. B tiene que enviar a A un conjunto de mensajes de datos, cumpliendo los siguientes convenios. Cuando A recibe una solicitud de la capa superior para obtener el siguiente mensaje de datos (D) de B, A tiene que enviar un mensaje de solicitud (R) a B a través del canal que va de A a B. Sólo cuando B recibe un mensaje R puede devolver un mensaje de datos (D) a A a través del canal de B a A. A tiene que entregar exactamente una copia de cada mensaje D a la capa superior. Los mensajes R se pueden perder (pero no corromper) en el canal de A a B; los mensajes D, una vez enviados, siempre son correctamente entregados. El retardo a lo largo de ambos canales es desconocido y variable.
- Diseñe (proporcione una descripción de la FSM de) un protocolo que incorpore los mecanismos apropiados para compensar las pérdidas del canal de A a B e implemente el paso de los mensajes a la capa superior de la entidad A, como se ha explicado anteriormente. Utilice sólo aquellos mecanismos que sean absolutamente necesarios.
- P21. Considere los protocolos GBN y SR. Suponga que el tamaño del espacio de números de secuencia es k . ¿Cuál es la máxima ventana de emisor permitida que evitará la ocurrencia de problemas como los indicados en la Figura 3.27 para cada uno de estos protocolos?

- P22. Responda verdadero o falso a las siguientes preguntas y justifique brevemente sus respuestas:
- Con el protocolo SR, el emisor puede recibir un ACK para un paquete que se encuentra fuera de su ventana actual.
 - Con GBN, el emisor puede recibir un ACK para un paquete que se encuentra fuera de su ventana actual.
 - El protocolo de bit alternante es igual que el protocolo SR pero con un tamaño de ventana en el emisor y en el receptor igual a 1.
 - El protocolo de bit alternante es igual que el protocolo GBN pero con un tamaño de ventana en el emisor y en el receptor igual a 1.
- P23. Hemos dicho que una aplicación puede elegir UDP como protocolo de transporte porque UDP ofrece a la aplicación un mayor grado de control (que TCP) en lo relativo a qué datos se envían en un segmento y cuándo.
- ¿Por qué una aplicación tiene más control sobre qué datos se envían en un segmento?
 - ¿Por qué una aplicación tiene más control sobre cuándo se envía el segmento?
- P24. Se desea transferir un archivo de gran tamaño de L bytes del host A al host B. Suponga un MSS de 536 bytes.
- ¿Cuál es el valor máximo de L tal que los números de secuencia de TCP no se agoten? Recuerde que el campo número de secuencia de TCP tiene 4 bytes.
 - Para el valor de L que haya obtenido en el apartado (a), calcule el tiempo que tarda en transmitirse el archivo. Suponga que a cada segmento se añade un total de 66 bytes para la cabecera de la capa de transporte, de red y de enlace de datos antes de enviar el paquete resultante a través de un enlace a 155 Mbps. Ignore el control de flujo y el control de congestión de modo que A pueda bombear los segmentos seguidos y de forma continuada.
- P25. Los hosts A y B están comunicándose a través de una conexión TCP y el host B ya ha recibido de A todos los bytes hasta el byte 126. Suponga que a continuación el host A envía dos segmentos seguidos al host B. El primer y el segundo segmentos contienen, respectivamente, 70 y 50 bytes de datos. En el primer segmento, el número de secuencia es 127, el número del puerto de origen es 302 y el número de puerto de destino es 80. El host B envía un paquete de reconocimiento cuando recibe un segmento del host A.
- En el segundo segmento enviado del host A al B, ¿Cuáles son el número de secuencia, el número del puerto de origen y el número del puerto de destino?
 - Si el primer segmento llega antes que el segundo segmento, ¿cuál es el número de reconocimiento, el número del puerto de origen y el número del puerto de destino en el ACK correspondiente al primer segmento?
 - Si el segundo segmento llega antes que el primero, ¿cuál es el número de reconocimiento en el ACK correspondiente al primer segmento?
 - Suponga que los dos segmentos enviados por A llegan en orden a B. El primer paquete de reconocimiento se pierde y el segundo llega después de transcurrido el primer intervalo de fin de temporización. Dibuje un diagrama de temporización que

- muestre estos segmentos y todos los restantes segmentos y paquetes de reconocimiento enviados. (Suponga que no se producen pérdidas de paquetes adicionales.) para cada uno de los segmentos que incluya en su diagrama, especifique el número de secuencia y el número de bytes de datos; para cada uno de los paquetes de reconocimiento que añada, proporcione el número de reconocimiento.
- P26. Los hosts A y B están directamente conectados mediante un enlace a 100 Mbps. Existe una conexión TCP entre los dos hosts y el host A está transfiriendo al host B un archivo de gran tamaño a través de esta conexión. El host A puede enviar sus datos de la capa de aplicación a su socket TCP a una velocidad tan alta como 120 Mbps pero el host B sólo puede leer los datos almacenados en su buffer de recepción TCP a una velocidad máxima de 60 Mbps. Describa el efecto del control de flujo de TCP.
- P27. En la Sección 3.5.6 se han estudiado las cookies SYN.
- ¿Por qué es necesario que el servidor utilice un número de secuencia inicial especial en SYNACK?
 - Suponga que un atacante sabe que un host objetivo utiliza cookies SYN. ¿Puede el atacante crear conexiones semi-abiertas o completamente abiertas enviando simplemente un paquete ACK al host objetivo? ¿Por qué?
 - Suponga que un atacante recopila una gran cantidad de números de secuencia iniciales enviados por el servidor. ¿Puede el atacante hacer que el servidor cree muchas conexiones completamente abiertas enviando paquetes ACK con esos números de secuencia iniciales? ¿Por qué?
- P28. Considere la red mostrada en el escenario 2 de la Sección 3.6.1. Suponga que ambos hosts emisores A y B tienen definidos valores de fin de temporización fijos.
- Demuestre que aumentar el tamaño del buffer finito del router puede llegar a hacer que se reduzca la tasa de transferencia (λ_{out}).
 - Suponga ahora que ambos hosts ajustan dinámicamente sus valores de fin de temporización (como lo hace TCP) basándose en el retardo del buffer del router. ¿Incrementar el tamaño del buffer ayudaría a incrementar la tasa de transferencia? ¿Por qué?
- P29. Considere el procedimiento de TCP para estimar RTT. Suponga que $\alpha = 0,1$. Sea $RTT_{Muestra_1}$ la muestra de RTT más reciente, $RTT_{Muestra_2}$ la siguiente muestra de RTT más reciente, y así sucesivamente.
- Para una conexión TCP determinada, suponga que han sido devueltos cuatro paquetes de reconocimiento con las correspondientes muestras de RTT, $RTT_{Muestra_4}$, $RTT_{Muestra_3}$, $RTT_{Muestra_2}$ y $RTT_{Muestra_1}$. Exprese $RTT_{Estimado}$ en función de las cuatro muestras de RTT.
 - Generalice la fórmula para n muestras de RTT.
 - En la fórmula del apartado (b), considere que n tiende a infinito. Explique por qué este procedimiento de cálculo del promedio se conoce como media móvil exponencial.
- P30. En la Sección 3.5.3, se ha estudiado la estimación de RTT en TCP. ¿Por qué cree que TCP evita medir $RTT_{Muestra}$ para los segmentos retransmitidos?

- P31. ¿Cuál es la relación entre la variable `EnviarBase` de la Sección 3.5.4 y la variable `UltimoByteRecibido` de la Sección 3.5.5?
- P32. ¿Cuál es la relación entre la variable `UltimoByteRecibido` de la Sección 3.5.5 y la variable `y` de la Sección 3.5.4?
- P33. En la Sección 3.5.4 hemos visto que TCP espera hasta que ha recibido tres ACK duplicados antes de realizar una retransmisión rápida. ¿Por qué cree que los diseñadores de TCP han decidido no realizar una retransmisión rápida después de recibir el primer ACK duplicado correspondiente a un segmento?
- P34. Compare GBN, SR y TCP (sin paquetes ACK retardados). Suponga que los valores de fin de temporización de los tres protocolos son los suficientemente grandes como para que 5 segmentos de datos consecutivos y sus correspondientes ACK puedan ser recibidos (si no se producen pérdidas en el canal) por el host receptor (host B) y el host emisor host (host A), respectivamente. Suponga que el host A envía 5 segmentos de datos al host B y que el segundo segmento (enviado desde A) se pierde. Al final, los 5 segmentos de datos han sido recibidos correctamente por el host B.
- ¿Cuántos segmentos ha enviado en total el host A y cuantos ACK ha enviado en total el host B? ¿Cuáles son sus números de secuencia? Responda a esta pregunta para los tres protocolos.
 - Si los valores de fin de temporización para los tres protocolos son mucho mayores que 5 RTT, ¿qué protocolo entregará correctamente los cinco segmentos de datos en el menor intervalo de tiempo?
- P35. En la descripción de TCP de la Figura 3.53, el valor del umbral se define como $\text{umbral} = \text{VentanaCongestion} / 2$ en varios sitios y el valor de `umbral` se hace igual a la mitad del tamaño de la ventana cuando se produce un suceso de pérdida. ¿Tiene que ser la velocidad a la que el emisor está transmitiendo cuando se produce un suceso de pérdida aproximadamente igual a `VentanaCongestion` segmentos por RTT? Explique su respuesta. Si su respuesta es no, ¿puede sugerir una forma diferente en la que se podría fijar el valor de `umbral`?
- P36. Considere la Figura 3.46(b). Si λ'_{in} aumenta por encima de $R/2$, ¿puede λ_{out} incrementarse por encima de $R/3$? Explique su respuesta. Considere ahora la Figura 3.46(c). Si λ'_{in} aumenta por encima de $R/2$, ¿puede λ_{out} aumentar por encima de $R/4$ suponiendo que un paquete será reenviado dos veces como media desde el router al receptor? Explique su respuesta.
- P37. Considere la Figura 3.58.
- Suponiendo que TCP Reno es el protocolo que presenta el comportamiento mostrado en la figura, responda a las siguientes preguntas. En todos los casos, deberá proporcionar una breve explicación que justifique su respuesta.
- Identifique los intervalos de tiempo cuando TCP está operando en el modo de arranque lento.
 - Identifique los intervalos de tiempo cuando TCP está operando en el modo de evitación de la congestión.

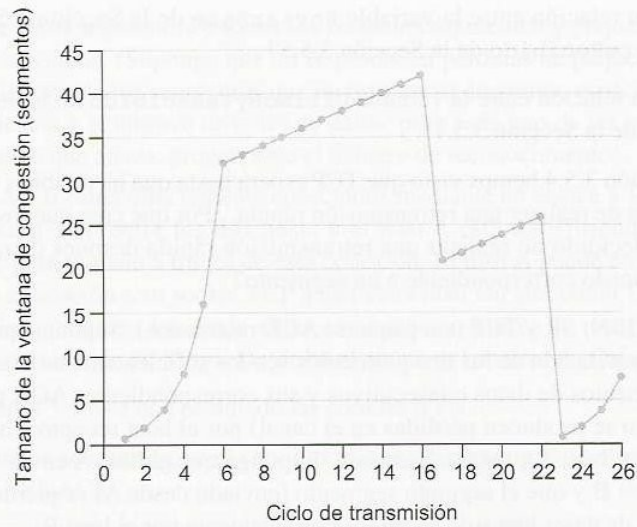


Figura 3.58 • Tamaño de ventana de TCP en función del tiempo.

- c. Después del ciclo de transmisión 16, ¿se detecta la pérdida de segmento mediante tres ACK duplicados o mediante un fin de temporización?
 - d. Después del ciclo de transmisión 22, ¿se detecta la pérdida de segmento mediante tres ACK duplicados o mediante un fin de temporización?
 - e. ¿Cuál es el valor inicial de `umbra1` en el primer ciclo de transmisión?
 - f. ¿Cuál es el valor de `umbra1` transcurridos 18 ciclos de transmisión?
 - g. ¿Cuál es el valor de `umbra1` transcurridos 24 ciclos de transmisión?
 - h. ¿Durante cuál ciclo de transmisión se envía el segmento 70?
 - i. Suponiendo que se detecta una pérdida de paquete después del ciclo de transmisión 26 a causa de la recepción de un triple ACK duplicado, ¿cuáles serán los valores del tamaño de la ventana de congestión y de `umbra1`?
 - j. Suponga que se utiliza TCP Tahoe (en lugar de TCP Reno) y que se han recibido triples ACK duplicados en el ciclo de transmisión 16. ¿Cuáles serán los valores del tamaño de la ventana de congestión y de `umbra1` en el ciclo de transmisión 19?
 - k. Suponga otra vez que se utiliza TCP Tahoe y que se produce un suceso de fin de temporización en el ciclo de transmisión 22. ¿Cuántos paquetes han sido enviados entre los ciclos de transmisión 17 a 22, ambos inclusive?
- P38. Utilice la Figura 3.56, que ilustra la convergencia del algoritmo AIMD de TCP. Suponga que en lugar de un decrecimiento multiplicativo, TCP disminuye el tamaño de la ventana en una cantidad constante. ¿Convergería el algoritmo AIAD resultante hacia un algoritmo de cuota equitativa? Justifique su respuesta utilizando un diagrama similar al de la Figura 3.56.
- P39. En la Sección 3.5.4, hemos explicado que el intervalo de fin de temporización se duplica después de un suceso de fin de temporización. Este mecanismo es una forma

de control de congestión. ¿Por qué TCP necesita un mecanismo de control de congestión basado en ventana (como hemos estudiado en la Sección 3.7) además de un mecanismo de duplicación del intervalo de fin de temporización?

P40. El host A está enviando un archivo de gran tamaño al host B a través de una conexión TCP. En esta conexión nunca se pierden paquetes y los temporizadores nunca caducan. La velocidad de transmisión del enlace que conecta el host A con Internet es R bps. Suponga que el proceso del host A es capaz de enviar datos a su socket TCP a una velocidad de S bps, donde $S = 10 \cdot R$. Suponga también que el buffer de recepción de TCP es lo suficientemente grande como para almacenar el archivo completo y que el buffer emisor sólo puede almacenar un porcentaje del archivo. ¿Qué impide al proceso del host A pasar datos de forma continua a su socket TCP a una velocidad de S bps? ¿El mecanismo de control de flujo de TCP, el mecanismo de control de congestión de TCP o alguna otra cosa? Razone su respuesta.

P41. Se envía un archivo de gran tamaño de un host a otro a través de una conexión TCP sin pérdidas.

- Suponga que TCP utiliza el algoritmo AIMD para su control de congestión sin fase de arranque lento. Suponiendo que `ventanaCongestion` aumenta 1 MSS cada vez que se recibe un lote de paquetes ACK y suponiendo que los intervalos RTT son aproximadamente constantes, ¿Cuánto tiempo tarda `ventanaCongestion` en aumentar de 5 MSS a 11 MSS (si no se producen sucesos de pérdida de paquetes)?
- ¿Cuál es la tasa de transferencia media (en función de MSS y RTT) para esta conexión hasta llegar al periodo RTT número 6?

P42. Recuerde la descripción macroscópica de la tasa de transferencia de TCP. En el periodo de tiempo que va desde que la velocidad de la conexión varía entre $W/(2 \cdot RTT)$ y W/RTT , sólo se pierde un paquete (justo al final del periodo).

- Demuestre que la tasa de pérdidas (fracción de paquetes perdidos) es igual a:

$$L = \text{tasa de pérdidas} = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

- Utilice el resultado anterior para demostrar que si una conexión tiene una tasa de pérdidas igual a L , entonces su tasa promedio es aproximadamente igual a

$$\approx \frac{1,22 \cdot MSS}{RTT \sqrt{L}}$$

P43. Considere una única conexión TCP (Reno) que emplea un enlace a 10Mbps que no almacena en buffer ningún dato. Suponga que este enlace es el único enlace congestionado entre los hosts emisor y receptor. Suponga también que el emisor TCP tiene que enviar al receptor un archivo de gran tamaño y que el buffer de recepción del receptor es mucho más grande que la ventana de congestión. Haremos además las siguientes suposiciones: el tamaño de segmento TCP es de 1.500 bytes, el retardo de propagación de ida y vuelta de esta conexión es igual a 100 milisegundos y esta conexión TCP siempre se encuentra en la fase de evitación de la congestión, es decir, ignoramos la fase de arranque lento.

- a. ¿Cuál es el tamaño máximo de ventana (en segmentos) que esta conexión TCP puede alcanzar?
 - b. ¿Cuáles son el tamaño medio de ventana (en segmentos) y la tasa de transferencia media (en bps) de esta conexión TCP?
 - c. ¿Cuánto tiempo tarda esta conexión TCP en alcanzar de nuevo su tamaño de ventana máximo después de recuperarse de una pérdida de paquete?
- P44. Continuando con el escenario descrito en el problema anterior, suponga que el enlace a 10Mbps puede almacenar en buffer un número finito de segmentos. Razone por qué para que el enlace esté siempre ocupado enviando datos, deberíamos seleccionar un tamaño de buffer que sea al menos igual al producto de la velocidad del enlace C y el retardo de propagación de ida y vuelta entre el emisor y el receptor.
- P45. Repita el Problema 43, pero sustituyendo el enlace a 10 Mbps por un enlace a 10 Gbps. Observe que en la respuesta al apartado (c) habrá demostrado que se tarda mucho tiempo en que el tamaño de la ventana de congestión alcance su máximo después de recuperarse de una pérdida de paquete. Diseñe una solución que resuelva este problema.
- P46. Sea T (medido en RTT) el intervalo de tiempo que una conexión TCP tarda en aumentar el tamaño de su ventana de congestión de $W/2$ a W , donde W es el tamaño máximo de la ventana de congestión. Demuestre que T es una función de la tasa de transferencia media de TCP.
- P47. Considere un algoritmo AIMD de TCP simplificado en el que el tamaño de la ventana de congestión se mide en número de segmentos, no en bytes. En la fase de incremento aditivo, el tamaño de la ventana de congestión se incrementa en un segmento cada RTT. En la fase de decrecimiento multiplicativo, el tamaño de la ventana de congestión se reduce a la mitad (si el resultado no es un entero, redondee al entero más próximo). Suponga que dos conexiones TCP, C_1 y C_2 , comparten un enlace congestionado cuya velocidad es de 30 segmentos por segundo. Suponemos que tanto C_1 como C_2 están en la fase de evitación de la congestión. El intervalo RTT de la conexión C_1 es igual a 100 milisegundos y el de la conexión C_2 es igual a 200 milisegundos. Suponemos que cuando la velocidad de los datos en el enlace excede la velocidad del enlace, todas las conexiones TCP experimentan pérdidas de segmentos de datos.
- a. Si en el instante t_0 el tamaño de la ventana de congestión de ambas conexiones, C_1 y C_2 , es de 10 segmentos, ¿cuáles serán los tamaños de dichas ventanas de congestión después de transcurridos 2200 milisegundos?
 - b. ¿Obtendrán estas dos conexiones, a largo plazo, la misma cuota de ancho de banda del enlace congestionado? Explique su respuesta.
- P48. Continúe con la red descrita en el problema anterior, pero ahora suponga que las dos conexiones TCP, C_1 y C_2 , tienen el mismo intervalo RTT de 100 milisegundos. Suponga que en el instante t_0 , el tamaño de la ventana de congestión de C_1 es de 15 segmentos pero el tamaño de la ventana de congestión de C_2 es igual a 10 segmentos.
- a. ¿Cuáles serán los tamaños de las ventanas de congestión después de transcurridos 2200 milisegundos?

- b. ¿Obtendrán estas dos conexiones, a largo plazo, la misma cuota de ancho de banda del enlace congestionado?
- c. Decimos que dos conexiones están sincronizadas si ambas conexiones alcanzan su tamaño de ventana máximo al mismo tiempo y alcanzan su tamaño mínimo de ventana también al mismo tiempo. ¿Terminarán con el tiempo sincronizándose estas dos conexiones? En caso afirmativo, ¿cuáles son sus tamaños máximos de ventana?
- d. ¿Ayudará esta sincronización a mejorar la tasa de utilización del enlace compartido? ¿Por qué? Esboce alguna idea para evitar esta sincronización.
- P49. Veamos una modificación del algoritmo de control de congestión de TCP. En lugar de utilizar un incremento aditivo podemos emplear un incremento multiplicativo. Un emisor TCP incrementa su tamaño de ventana según una constante pequeña positiva a ($0 < a < 1$) cuando recibe un ACK válido. Halle la relación funcional existente entre la tasa de pérdidas L y el tamaño máximo de la ventana de congestión W . Demuestre que para esta conexión TCP modificada, independientemente de la tasa media de transferencia de TCP, una conexión TCP siempre invierte la misma cantidad de tiempo en incrementar el tamaño de su ventana de congestión de $W/2$ a W .
- P50. En nuestra exposición sobre el futuro de TCP de la Sección 3.7 hemos destacado que para alcanzar una tasa de transferencia de 10 Gbps, TCP sólo podría tolerar una probabilidad de pérdida de segmentos de $2 \cdot 10^{-10}$ (o lo que es equivalente, un suceso de pérdida por cada 5.000.000.000 segmentos). Indique de dónde se obtienen los valores $2 \cdot 10^{-10}$ y 1 por cada 5.000.000 para los valores de RTT y MSS dados en la Sección 3.7. Si TCP tuviera que dar soporte a una conexión a 100 Gbps, ¿qué tasa de pérdidas sería tolerable?
- P51. En nuestra exposición sobre el control de congestión de TCP de la Sección 3.7, implícitamente hemos supuesto que el emisor TCP siempre tiene datos que enviar. Consideremos ahora el caso en que el emisor TCP envía una gran cantidad de datos y luego en el instante t_1 se queda inactivo (puesto que no tiene más datos que enviar). TCP permanece inactivo durante un periodo de tiempo relativamente largo y en el instante t_2 quiere enviar más datos. ¿Cuáles son las ventajas y las desventajas de que TCP tengan que utilizar los valores de `ventanaCongestion` y `umbral` de t_1 cuando comienza a enviar datos en el instante t_2 ? ¿Qué alternativa recomendaría? ¿Por qué?
- P52. En este problema vamos a investigar si UDP o TCP proporcionan un cierto grado de autenticación del punto terminal.
- Considere un servidor que recibe una solicitud dentro de un paquete UDP y responde a la misma dentro de un paquete UDP (por ejemplo, como en el caso de un servidor DNS). Si un cliente con la dirección IP X suplanta su dirección con la dirección Y, ¿A dónde enviará el servidor su respuesta?
 - Suponga que un servidor recibe un SYN con la dirección IP de origen Y, y después de responder con un SYNACK, recibe un ACK con la dirección IP de origen Y y con el número de reconocimiento correcto. Suponiendo que el servidor elige un número de secuencia inicial aleatorio y que no existe ningún atacante interpuesto (*man-in-the-middle*), ¿puede el servidor estar seguro de que el cliente está en la dirección Y (y no en alguna otra dirección X que esté intentando suplantar a Y)?
- P53. En este problema, vamos a considerar el retardo introducido por la fase de arranque lento de TCP. Se tiene un cliente y un servidor web directamente conectados mediante un enlace a velocidad R . Suponga que el cliente desea extraer un objeto cuyo tamaño es exactamente igual a $15 S$, donde S es el tamaño máximo de segmento (MSS). Sea RTT el tiempo de transmisión de ida y vuelta entre el cliente y el servidor (suponemos que es constante). Ignorando las cabeceras del protocolo, determine el tiempo necesario para recuperar el objeto (incluyendo el tiempo de establecimiento de la conexión TCP) si
- $4 S/R > S/R + RTT > 2S/R$
 - $S/R + RTT > 4 S/R$
 - $S/R > RTT$.