

Redes de Computadoras

Práctico 4

Curso 2024

Objetivos

- Comprender los conceptos relacionados con la transferencia confiable de datos.
- Comprender el diseño y funcionamiento del protocolo TCP.
- Comprender los conceptos del control de congestión y su implementación en TCP.

Duración

- 3 clases.

Ejercicio 1 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, 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.

Ejercicio 2 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 solo 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 solo emplee paquetes NAK a otro que utilice paquetes ACK? ¿Por qué?

Ejercicio 3 Represente las máquinas de estados del protocolo `rdt 2.0`, que implementa un servicio de transporte confiable asumiendo que no se pierden ni se alteran los mensajes de reconocimiento ACK y NAK.

Ejercicio 4 Se considera un escenario donde los hosts A y B quieren enviar mensajes al host C . A y C están conectados por un canal que puede perder y corromper mensajes pero no reordenarlos, mientras que B y C están conectados por otro canal independiente con las mismas características. La capa de transporte del host C debe entregar mensajes a la capa superior provenientes de los hosts A y B , en forma alternada; es decir, primero debe entregar un mensaje de A , luego uno de B , luego uno de A , y así por siempre.

Se pide: Diseñe un protocolo confiable del tipo stop-and-wait para el transporte de mensajes desde A y B hacia C , que asegure la entrega alternada que se describió anteriormente. Proponga descripciones de las máquinas de estado de A/B y C , y del formato de los paquetes utilizados.

Ejercicio 5 Sea un protocolo GBN con una ventana de emisor de 3 y un rango de números de secuencia de 1024. 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 las siguientes preguntas:

- ¿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.

Ejercicio 6 Responda verdadero o falso a las siguientes preguntas y justifique brevemente sus respuestas:

- (a) Con el protocolo SR, el emisor puede recibir un ACK para un paquete que se encuentra fuera de su ventana actual.
- (b) Con GBN, el emisor puede recibir un ACK para un paquete que se encuentra fuera de su ventana actual.
- (c) 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.
- (d) 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.

Ejercicio 7 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 segmento 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*.

- (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?
- (b) 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?
- (c) Si el segundo segmento llega antes que el primero, ¿cuál es el número de reconocimiento en el ACK correspondiente al primer segmento?
- (d) 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, suponiendo 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.

Ejercicio 8 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.

Ejercicio 9 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 *B* y el host emisor *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*.

- (a) ¿Cuántos segmentos ha enviado en total el host *A*? ¿Cuántos ACK ha enviado en total el host *B*? ¿Cuáles son sus números de secuencia? Responda esta pregunta para los tres protocolos.
- (b) 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?

Ejercicio 10 Considere la Figura 1. Suponiendo que TCP Reno es el protocolo que presenta el comportamiento mostrado en la figura, responda a las siguientes preguntas. Justifique siempre 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**.
- Después del ciclo de transmisión 16, ¿se detecta la pérdida de segmento mediante tres ACK duplicados o mediante un fin de temporización?
- Después del ciclo de transmisión 22, ¿se detecta la pérdida de segmento mediante tres ACK duplicados o mediante un fin de temporización?
- ¿Cuál es el valor inicial de **umbral** en el primer ciclo de transmisión?
- ¿Cuál es el valor de **umbral** transcurridos 18 ciclos de transmisión?
- ¿Cuál es el valor de **umbral** transcurridos 24 ciclos de transmisión?
- ¿Durante cuál ciclo de transmisión se envía el segmento 70?
- 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 **umbral**?
- 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 umbral en el ciclo de transmisión 19?
- 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?

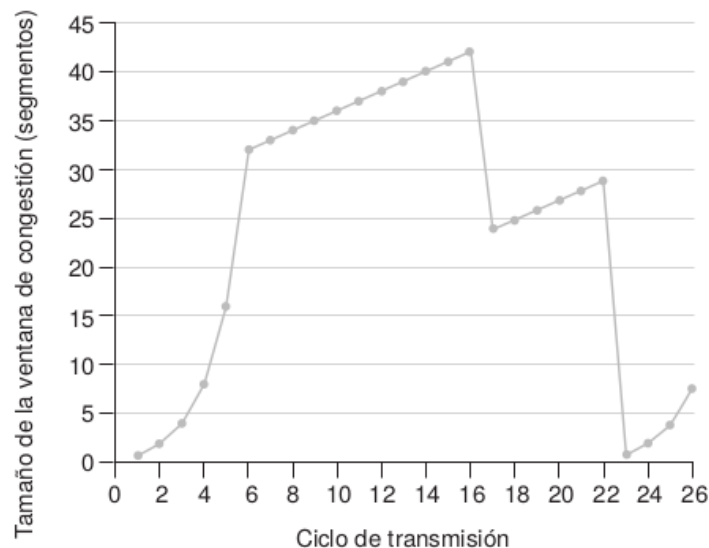


Figura 1: Tamaño de ventana de TCP en función del tiempo.

Ejercicio 11 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 $S = 10 \times R$. 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 u otro factor? Justifique su respuesta.

Ejercicio 12 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 también dentro de un paquete UDP. Si un cliente con la dirección IP X suplanta su dirección con la dirección Y , ¿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 SYN-ACK, 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)?

Ejercicio 13 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 $15S$, 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:

- $4S/R > S/R + RTT > 2S/R$.
- $S/R + RTT > 4S/R$.
- $S/R > RTT$.

Ejercicio 14 En el protocolo "Selective Repeat" (SR) genérico que hemos estudiado en el curso, 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 el curso (`udt_enviar()`, `iniciar_temporizador()`, `rdt_recibir()`, etc.), defina claramente las acciones que realizan.
- Proporcione una gráfica temporal del emisor y del receptor que muestre cómo este protocolo se recupera de la pérdida de un paquete.