

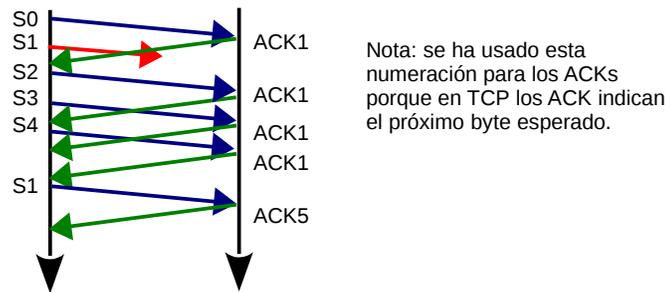
**Segundo Certamen**

Pregunta 1: Para el protocolo TCP- 50 pts.

10 puntos cada una, el resultado se multiplica por 5/6.

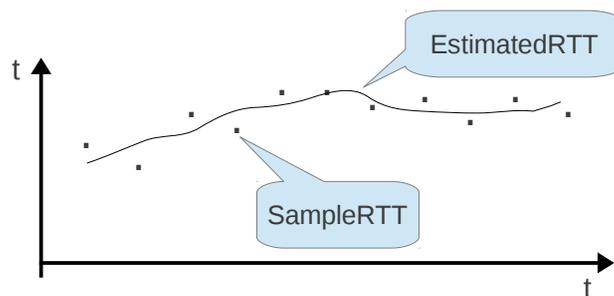
a. Explique y dé un ejemplo en un diagrama temporal del algoritmo de retransmisión rápida.

El algoritmo de retransmisión rápida cuenta el número de acuses de recibos duplicados. Ante la llegada del tercer acuse de recibo duplicado (es decir el cuarto acuse de recibo igual), el transmisor supone el paquete siguiente se perdió y lo reenvía inmediatamente sin esperar por ocurrencia de timeout.



b. Describa la diferencia entre SampleRTT y EstimatedRTT. Dé un ejemplo concreto.

SampleRTT es al valor de RTT medido ante la llegada de un ACK mientras que EstimatedRTT corresponde a una estimación para el valor del próximo SampleRTT.



c. Si envía un archivo muy grande de L bytes desde terminal A a B. Hay una conexión TCP entre ambos. El terminal A puede enviar datos de capa aplicación a su socket TCP a una tasa de 150 Mbps. El terminal B puede leer de su receive buffer de TCP a una tasa de 50 Mbps. Describa en detalle el efecto del control de flujo en TCP.

Para apreciar el efecto del control de flujo supondré que el enlace puede enviar a tasas superiores a 150 Mbps. La aplicación comenzará enviando sus datos a tasa 150 Mbps y el receptor los leerá a tasa 50 Mbps. Luego llegarán más datos al buffer receptor de los leídos por B. La tasa de crecimiento de los datos del buffer receptor será de aproximadamente 100 Mbps (150-50 Mbps). Cuando el buffer se llene, TCP receptor enviará un paquete con ventana de recepción cero, con lo cual TCP transmisor dejará de transmitir. Cuando B lea y algo desocupe el buffer, TCP informará una nueva RcvWindows y se reactivará el envío. Así TCP transmisor no enviará más datos que el espacio libre del buffer receptor. Esto se mantiene hasta el envío del archivo completo.

d. ¿Cuál es el rol de SendBase y NextSeqNum en TCP? ¿Cuál es la función del RcvBuffer, RcvWindow, LastByteRcvd y LastByteRead?

SendBase es una variable que ocupa TCP en su transmisión para indicar el número de secuencia del byte más antiguo sin acuse de recibo.

NextSeqNum es una variable que ocupa TCP en su transmisión para indicar el número de secuencia del próximo byte a ser enviado.

La función de RcvBuffer es registrar el tamaño total del buffer usando en TCP receptor.

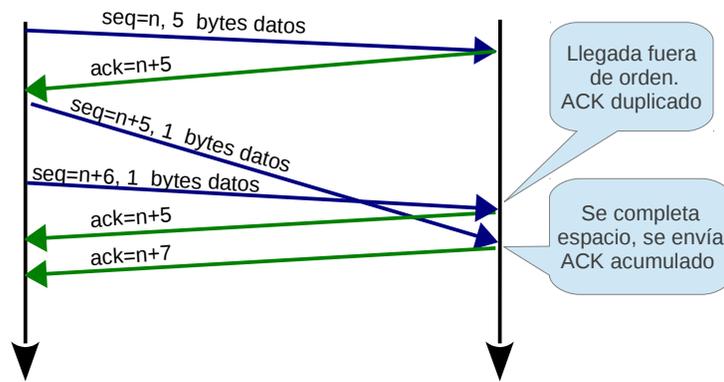
La función de RcvWindow es indicar el espacio disponible del buffer receptor para recibir más datos.

La función de LastByteRcvd es indicar el mayor número secuencia recibido en el buffer.

La función de LastByteRead es indicar el número de secuencia del último byte leído por la aplicación receptora.

e. Muestre un diagrama temporal de TCP que muestre el uso del ACK acumulativo. Incluya números de secuencia, número de ack, bytes transmitidos y cualquier otro elemento que sirva para entender el método.

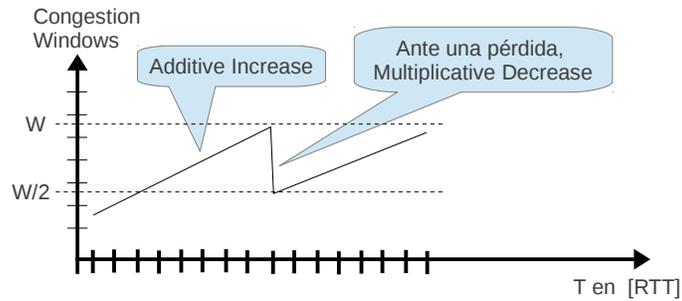
Varios escenarios pueden conducir al envío de un ACK acumulado, por ejemplo pedido en a). Aquí se muestra un caso simple donde un segmento de datos toma una ruta más larga y llega con posterioridad al segmento siguiente.



f. ¿Para qué sirve AIMD (Additive Increase Multiplicative Decrease) en TCP? Dé un ejemplo concreto.

AIMD sirve para aprovechar la capacidad de la conexión y controlar congestión.

Un ejemplo: Para aprovechar la capacidad de transmisión máxima de una conexión extremo a extremo, un transmisor TCP aumenta su ventana de congestión linealmente (Additive Increase). Se mantiene así hasta que recibe 3 acuses de recibo duplicados, TCP supone hubo una pérdida y asume que uno de los enlaces está congestionado. Para aliviar la congestión reduce su ventana de congestión a la mitad y con ello reduce también su tasa de transmisión. Gráficamente sería:



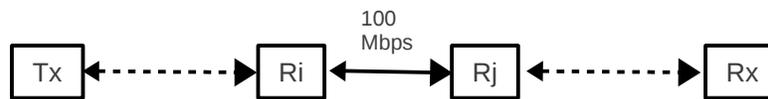
2.- Considerando TCP (Reno) con una conexión de 100 Mbps que no tiene buffer de data. Suponiendo que este link es el único link congestionado entre los terminales transmisores y receptores. Asuma que el transmisor TCP tiene un archivo muy grande que enviar y que el segmento TCP es de 1000 bytes, el retardo de propagación (ida y vuelta) es de 150 [ms] y que la conexión está en congestion avoidance (ignore slow start). (30 puntos)

- ¿Cuál es el máximo tamaño de la ventana (en segmentos) que esta conexión puede lograr?
- ¿Cuál es el tamaño promedio (en segundos) y throughput promedio (en bps) de esta conexión TCP?
- ¿Cuánto demoraría para que esta conexión TCP llegara al tamaño de ventana máximo después de recuperarse de una pérdida de paquetes?

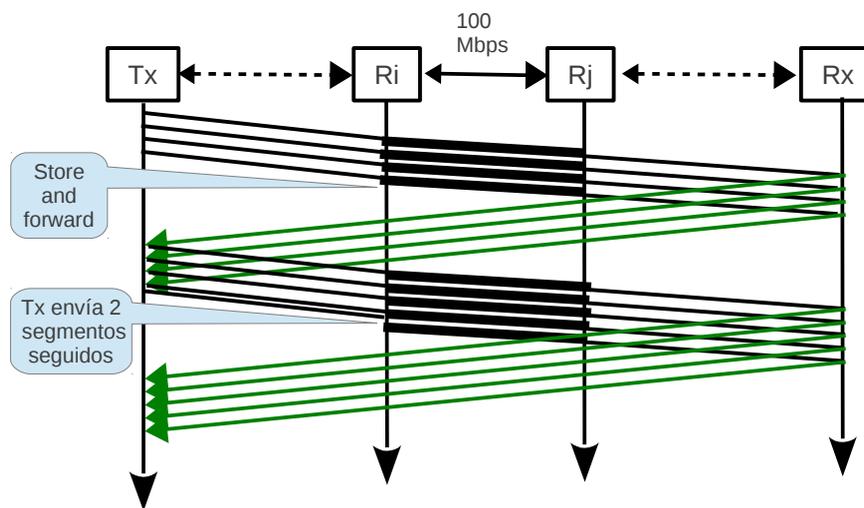
Una conexión TCP considera todos los enlaces desde la fuente a destino, luego supondré que se refiere a que el enlace más restrictivo para esta conexión le permite 100 Mbps de tráfico. En el desarrollo de esta pauta supondremos que el cuello de botella o enlace más restrictivo de la conexión posee buffer para dos segmentos.

(Nota: Esta pregunta es una variación de una planteada en el texto guía. Considerar que la conexión no tiene buffer de data implicaría que no hay almacenamiento y re-envío en toda la ruta y todos los enlaces deberían tener igual tasa de bits para que éstos salieran conforme llegan. En este caso el transmisor nunca logrará congestionar el enlace (no habrá pérdidas) pues su enlace de salida tiene la misma capacidad. En este caso se espera que la ventana crezca hasta lograr máxima utilización del enlace (100%) y se mantendría en esa condición. Este no es un escenario presente en Internet.)

Luego la conexión puede ser modelada como:



Donde supondremos que Ri tiene buffer para dos segmentos. En "congestion avoidance" el aumento de la ventana se implementa de modo que ésta crece en 1 segmento una vez que se reciben todos los ACKs de la ventana previa. La ventana crecerá hasta que el enlace congestionado tenga una utilización del 100%, es decir esté continuamente transmitiendo. Como éste es el más restrictivo, en esta condición el Tx aún podrá enviar más segmentos. Esto generará una pérdida aislada con lo cual la ventana de congestión baja a la mitad y la tasa baja a la mitad. Así tendremos una situación del tipo:



a) La ventana máxima se produce cuando alcanzamos la máxima utilización del enlace más restrictivo, esto es:

$$\widehat{W} * \left( \frac{MSS}{R} \right) \simeq RTT$$

15 pts

$$\widehat{W} \simeq \frac{RTT * R}{MSS} = 0.150 [s] * 100 * 10^6 \frac{[bit/s]}{8000 [bit/segmento]} = 1875 [segmento]$$

b) Tamaño promedio (en segundos) de una conexión, no tiene interpretación. Aquí hubo un error de escritura, debemos suponer que se pide el tamaño promedio de la ventana de congestión medida en segmentos.

$$\overline{W} = \frac{3}{4} \widehat{W} = 1406 [segmento]$$

7 pts.

$$throughput_{promedio} = \frac{\overline{W} * MSS}{RTT} = 1406 [seg] * 8000 \frac{[bit/seg]}{0.15 [s]} = 75 [Mbps]$$

c) La ventana crece aproximadamente 1 segmento en cada RTT, luego ésta aumentara desde  $\widehat{W}/2$  a  $\widehat{W}$  en:  $\widehat{W}/2 * RTT = 1875/2 * 0.15 [s] = 141 [s]$ . 8 pts.

3.- Considerando TCP y control de congestión. (20 puntos)

a) Dé el diagrama de un segmento y describa los campos usados.

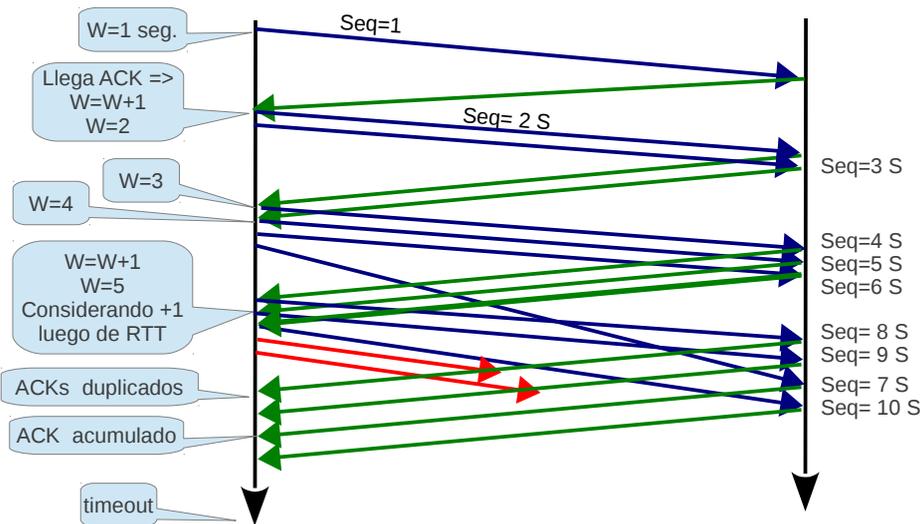
Supongo la pregunta no se restringe a control de congestión, sino a TCP en general:

Puerto Origen		Puerto Destino	
Número de secuencia			
Acuse de recibo			
Header Leng	U A P R S F	RecvWindow	
Checksum		Puntero Data Urgente	
Campos opcionales			
Datos			

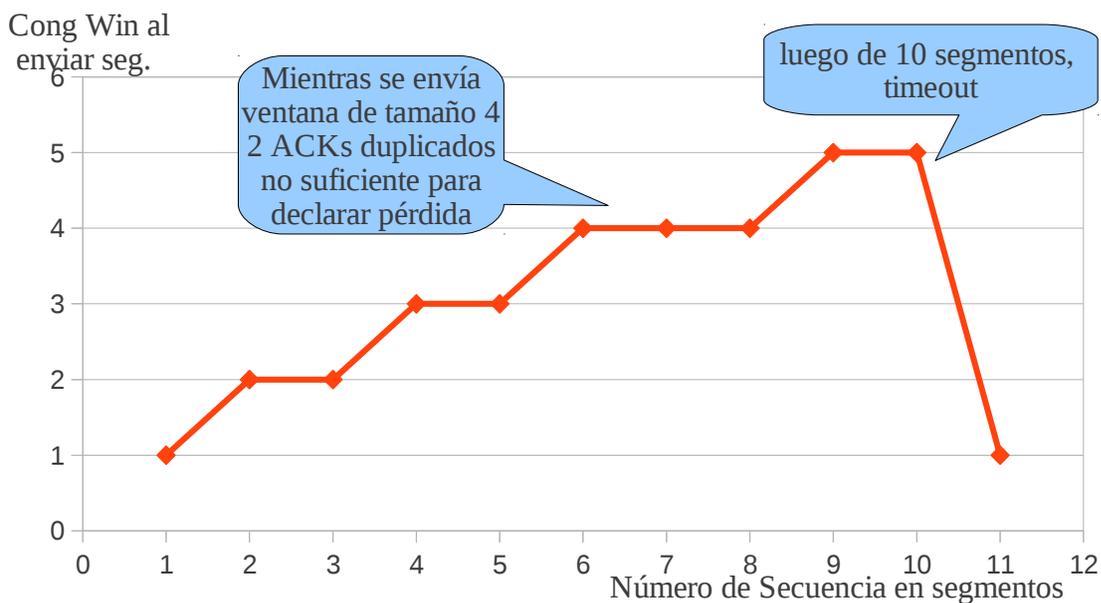
Se obtiene todo el puntaje si incluye los campos en azul (el orden no es estricto). 8 pts.

b) Asumiendo TCP Reno dé un gráfico de la ventana de congestión TCP en segmentos versus número de segmento transmitido. Asuma que se inicia con 1 segmento y que el umbral de congestión inicial es de 4 segmentos. Considere los siguientes eventos: 6 envíos exitosos, 2 ACKs duplicados, 4 envíos exitosos, un timeout.

Dos análisis parciales pueden conducir al gráfico pedido. Primero usaremos un diagrama de tiempo luego usaremos un gráfico Congestion Windows versus tiempo en RTT. Umbral de congestión inicial será entendido como umbral para partida lenta (Slow Star Threshold).



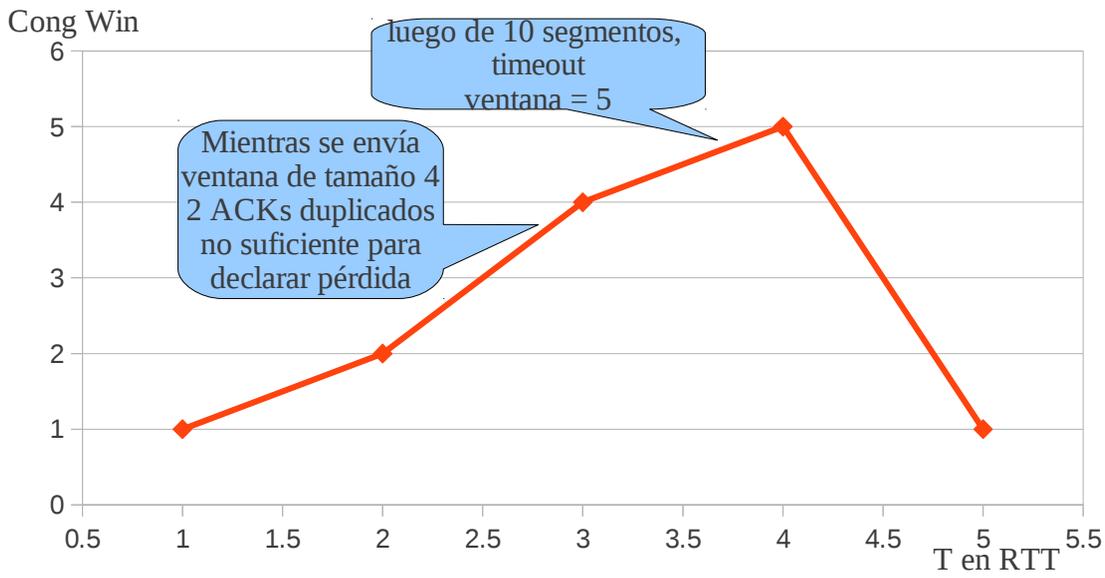
Sólo esto, 8 pts.



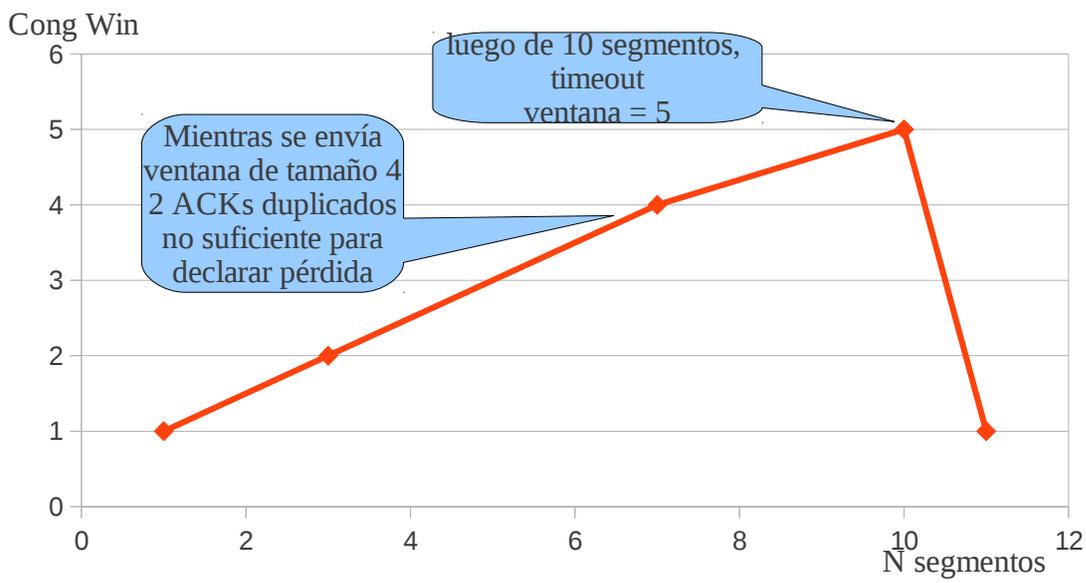
12 pts.

Lo importante es mostrar la tendencia, no reaccionar a los ACK duplicados y bajar a 1 luego de Seg. 10.

La gráfica habitual es Congestion Windows versus time medido en unidades de RTT. Aquí se pide ventana versus número de segmentos. Como paso intermedio, se desarrollará la gráfica Cong.Wind. Versus Tiempo en RTT.



Sólo esto, 10 pts.



12 pts.