

Segundo Certamen (Tiempo: 100 min.)

1.- (5 pts por respuesta) Responda en forma breve y clara:

a) Señale una semejanza y una diferencia entre UDP y TCP.

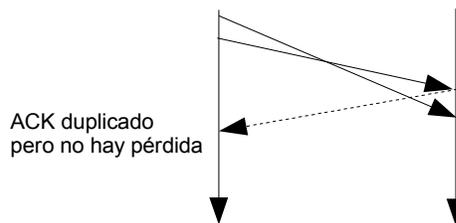
Semejanza: ambos son protocolos de la capa transporte y por ende ofrecen servicio de comunicación entre aplicaciones terminales.

Diferencia: UDP transporta bloques de datos, TCP transporta un flujo de bytes.

(más, UDP es no confiable, TCP sí; UDP no tiene control de flujo ni congestión, TCP sí; TCP tiene establecimiento y termino de conexión, UDP no, TCP es orientado a la conexión, UDP no)

b) En Go-Back-N ¿Qué impide declarar un paquete perdido ante la llegada del primer ACK duplicado?

El hecho que ese primer ACK duplicado no se deba a una pérdida de dato sino la llegada de dos paquetes fuera de orden.



c) Mencione dos diferencias entre Go-Back-N y Selective Repeat.

- Go-Back-N no requiere buffer en receptor, Selective Repeat sí lo usa.

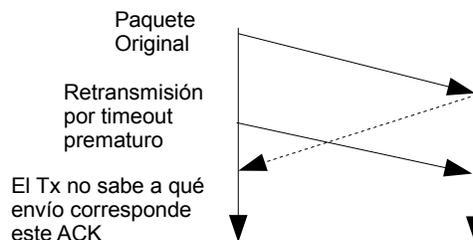
- Go-Back-N posee un solo timer, Selective Repeat posee uno por paquete.

(- Go-Back-N retransmite todos los paquetes posteriores a uno perdido, Selective Repeat retransmite sólo el perdido.

- Go-Back-N usa ACK acumulados, Selective Repeat no)

d) ¿Por qué TCP no actualiza el estimador para el RTT cuando llega un acuse de recibo de un segmento reenviado? Muestre un escenario que lo explique.

Porque TCP no puede asegurar que el ACK que recibe corresponde a la versión retransmitida; podría ser ser el ACK retrasado del paquete inicial.



e) Considerando el mecanismo usado por TCP para lograr confiabilidad, mencione una idea o mecanismo de TCP presente Go-Back-N (y no en Selective Repeat) y una idea o mecanismo de TCP tomado de Selective Repeat (y no en Go-Back-N).

Idea tomada de Go-back-N: Usar ACK acumulados.

Idea tomada de Selective Repeat: Almacenar en el receptor los paquetes fuera de orden.

2.- Considere una transmisión a través de un enlace de 1 Mbps, de 15 ms de retardo de propagación de un extremo al otro y tamaño de paquetes de 1000 Bytes.

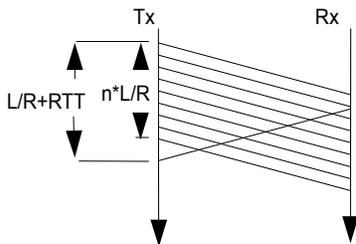
a) Determine el tamaño de la ventana del transmisor para garantizar una utilización del enlace superior a 90%.

b) Suponga números de secuencia en el rango 0 .. (N-1). Determine el valor mínimo para N si el enlace usa "Selective Repeat".

c) Suponga que el transmisor fija su timer igual $1,1 RTT$. En su respuesta considere Selective Repeat y para el diagrama pedido los paquetes P_i , paquete de dato con número de secuencia i , viajan de izquierda a derecha y ACK_i ,

acuse de recibo de paquete i, viajan de derecha a izquierda. Partiendo de P0 muestre el diagrama resultante cuando se pierde ACK0 y más tarde se pierde el tercer paquete de dato enviado. Termine su diagrama cuando $t=2RTT$.

a)



$$U = \frac{n * L / R}{L / R + RTT} \geq 0.90$$

$$L = 8000 [bit]; R = 10^6 [bit/s]; RTT = 30 [ms]$$

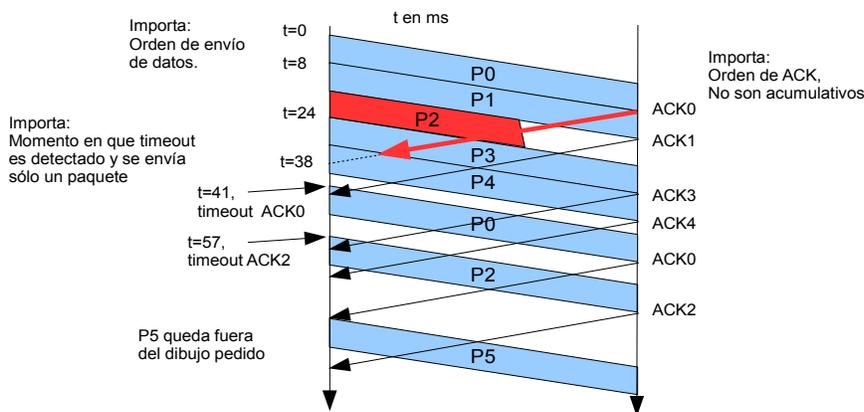
$$\text{tamaño de ventana} = n \geq 0.90 * \left(1 + \frac{R * RTT}{L}\right) = 0.90 * \left(1 + \frac{10^6 * 30 * 10^{-3}}{8000}\right) = 4,3$$

Luego el tamaño de ventana es 5 [paquetes]. 8 pts.

b) Con Selective Repeat se tiene:

$$\text{Rango de número de secuencias } N \geq 2 * \text{Tamaño de ventana} = 10 \quad 5 \text{ pts}$$

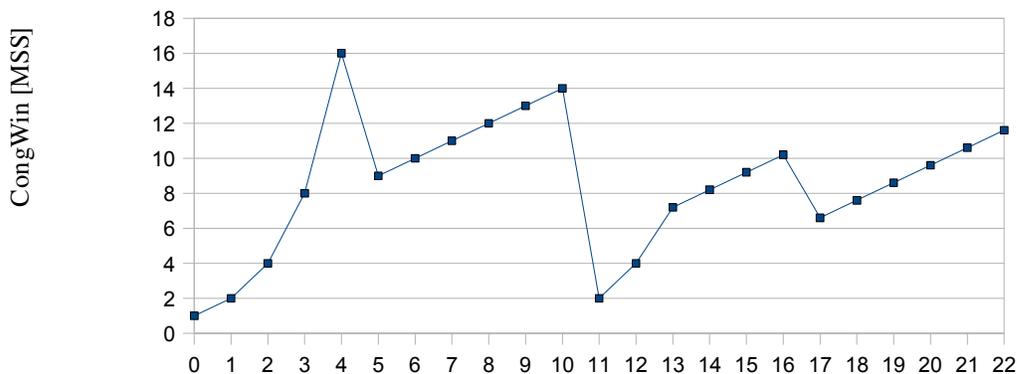
c)



12 pts

3.- Asumiendo que el receptor de una comunicación TCP tiene un buffer muy grande, grafique la ventana de congestión versus tiempo cuando se transfiere un archivo suficientemente grande. Considere el tiempo desde $t=0$ (justo se inicia la transferencia de datos) hasta $t=21 RTT$ y que en el proceso de transferencia ocurre lo siguiente:

- Justo después de $t = 4 RTT$ llegan 3 ACK duplicados.
- Justo después de $t = 10 RTT$ se produce un timeout en la espera de un ACK.
- Justo después de $t = 16 RTT$ llegan 3 ACK duplicados.

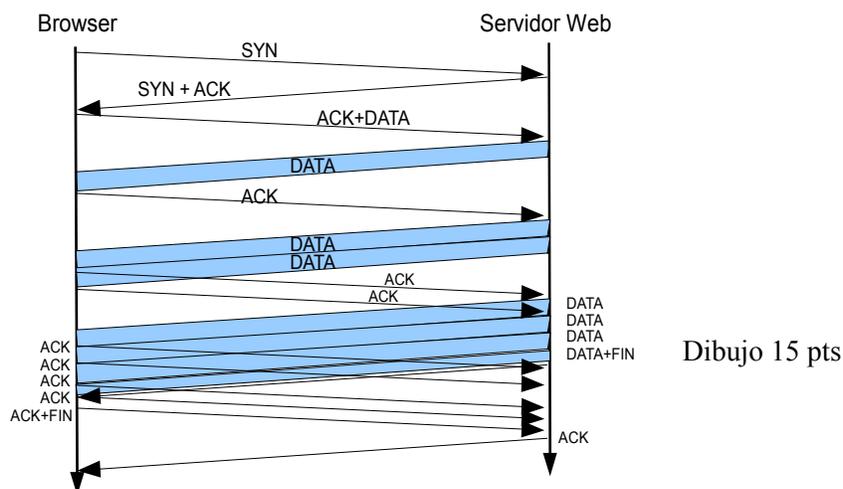


Tiempo [RTT]

Nota: Lo relevante es que para $t=0$ la ventana parte en 1 MSS y se duplica cada RTT (3+3 pts); luego en $t=4$ RTT y $t=16$ RTT cae a la "mitad" (3+3 pts) y se incrementa en 1 MSS cada RTT (3+3 pts); y para $t=10$ RTT vuelve a partir desde 1 RTT (3 pts) y se duplica cada RTT hasta superar 7, a partir de aquí aumenta linealmente (4 pts). Los números exactos no son tan críticos.

Análisis más detallado: para $t=0$, slow start, ventana = 1 MSS; para $t=4$ [RTT]+delta, se detecta una pérdida por 3 ACK duplicados, de inmediato la ventana baja a la mitad (8 MSS), y comienza a subir un MSS por cada RTT. Por ello para $t=5$ RTT, ventana =9MSS. Si usted consideró ventana de 8 para $t=5$ RTT será considerando bien (es un detalle menos relevante). Similar situación fue considerada para $t=10$ RTT + delta y $t=16$ RTT + delta. Notar que para $t=10$ RTT+delta ocurre el timeout, la pérdida del paquete debió comenzar entre $t=8$ RTT y $t=9$ RTT, pues el timeout debe ser al menos igual a 1 RTT; sin embargo, el control de congestión sólo reacciona cuando el timeout ocurre. Este análisis no es requeridos para optar al puntaje máximo.

4.- Desde mi casa uso un navegador para bajar una foto de 10.000 bytes desde un sitio web. La foto es el único objeto. Considere RTT para la conexión de 150 [ms], "ancho de banda" de 600.000 [bps], y que el MSS (Maximun Segment Size) de TCP es 1500 bytes (desprecie el tamaño de los encabezados). Asumiendo que no se pierden paquetes de ningún tipo, complete el diagrama adjunto desde el inicio de la conexión hasta la llegada completa de la foto. Estime el tiempo desde que mi navegador pide la foto hasta que la logro ver. En su diagrama rotule los segmentos con SYN, ACK, FIN, y DATA según corresponda.



En dibujo importa:

- Segmentos SYN, SYN+ACK y ACK. En este caso el último ACK ya lleva datos porque el cliente inicia la transferencia con mensaje GET de HTTP. Si usted consideró dos segmentos distintos, igualmente su estimación de tiempo conduce a lo mismo.
- Por slow start, el servidor parte enviando un segmento de tamaño máximo.
- Ante la llegada del primer ACK el servidor aumenta su ventana en un segmento por ACK.
- El último segmento es de tamaño menor (1000 bytes) pues no hay más datos que enviar.
- Para efectos del diagrama y cálculo pedido no se requiere incluir los mensajes FIN y ACK del FIN. La foto se puede ver antes que se cierre la conexión.

Tiempo total estimado hasta bajar foto:

$$t = 2 * RTT + RTT + (RTT - MSS / R) + L / R = 2 * 150 + 150 + (150 - 1500 * 8 / 600) + 10000 * 8 / 600 [ms]$$

$$t = 450 [ms] + (150 - 20) [ms] + 133 [ms] = 713 [ms] = 0,71 [s]$$

Expresión y cálculo 10 pts.