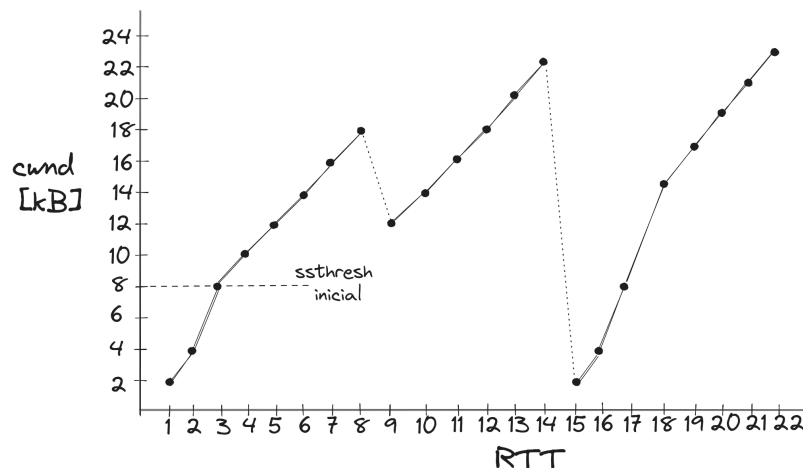


Guía de Ejercicios - 4

Control de Congestión

1. Responder las siguientes preguntas:
 - a. ¿Es necesario adoptar mecanismos de control de congestión en redes de conmutación de circuitos?
 - b. En el caso de disponer de *buffers* con memoria infinita en los routers de una red, ¿es posible evitar la congestión en dicha red?
 - c. ¿En qué situaciones la ventana de congestión de TCP puede no ser representativa del estado de congestión de la red?
2. `cwnd` y `ssthresh` son las dos variables que deben regularse en el control de congestión de TCP.
 - a. ¿Qué valores deberían tener cuando la conexión recién comienza?
 - b. ¿Cómo se modifican ante un *timeout*?
 - c. ¿Por qué Slow Start no inicia con una `cwnd` arbitrariamente grande?
 - d. ¿Cuánto debe incrementarse la ventana de congestión por cada RTT durante la fase de Congestion Avoidance?
3. Recurrir al [RFC 5681](#) para contestar las siguientes preguntas:
 - a. ¿A qué tipo de ataques es susceptible el control de congestión de TCP?
 - b. Profundizar en las referencias indicadas en qué consiste el ataque de división de ACKs. ¿Cómo puede mitigarse?
4. En una conexión TCP recién establecida se envían cuatro segmentos con los siguientes tamaños: 100 bytes, 150 bytes, 200 bytes y 300 bytes. Todos los segmentos llegan al receptor sin errores, aunque el segmento de 150 bytes llega último (fuera de orden).
 - a. Indicar hasta qué byte reconoce el receptor en cada ACK.
 - b. ¿Cuál es el valor de `ssthresh` en el emisor luego de recibir el ACK que indica que los cuatro segmentos llegaron a destino?
 - c. Repetir la pregunta anterior para `cwnd`.

-
5. En una conexión TCP, un interlocutor recibe cuatro ACKs con los siguientes RTTs muestreados en cada caso: 80 ms, 20 ms, 100 ms y 90 ms. Asumiendo que `EstimatedRTT` se inicializa con la primera medición (80 ms),
- Programar en Python una función que, dado un nuevo muestreo del RTT, permita calcular el RTO tal como vimos en clase.
 - Calcular el RTO luego de la llegada de cada ACK tomando $\alpha = 0.125$ y $\alpha = 0.4$.
 - ¿Se produce algún *timeout* en alguno de estos dos escenarios? Sacar conclusiones al respecto.
6. Se tiene una conexión TCP recién establecida en la que el RTT es de 10 ms. Uno de los interlocutores anuncia a su par una ventana de recepción de 24 KB. Asumiendo que no hay congestión ni demoras en la red, calcular cuánto tiempo transcurre antes de que pueda ser enviada la primera ventana de recepción completa (tener en cuenta el uso de Slow Start con una `cwnd` inicial de 1 MSS = 2 KB y `ssthresh` inicial de 64 KB).
7. Se tiene una conexión TCP recién establecida en la que uno de los interlocutores siempre anuncia una ventana de recepción `rwnd` = 16 KB. La red está sufriendo congestión al punto de que, si una ráfaga de segmentos fuese de por lo menos 16 KB, se perderían todos los segmentos que la conforman. Indicar cuánto vale la `cwnd` luego de enviar 40 KB de datos hacia dicho interlocutor (asumir una `cwnd` inicial de 1 MSS = 2 KB).
8. El siguiente gráfico muestra cómo evoluciona la ventana de congestión `cwnd` durante los primeros RTTs de una conexión de cierto emisor corriendo TCP Reno. Como se distingue en la figura, la `cwnd` toma un valor inicial de 2 kB = 1 MSS, mientras que la variable `ssthresh` se inicializa en 8 kB.



Describir todos los estados atravesados por el algoritmo de control de congestión, indicando en qué RTTs empieza y termina cada uno de ellos y qué eventos causan las transiciones (asumir que en todo momento la ventana de emisión efectiva se deriva únicamente de la ventana de congestión).

9. Consideremos un host que acaba de establecer una conexión TCP con otro host al que desea enviarle 20 kB de datos. En dicho interlocutor, la ventana de recepción es siempre de 8 kB. Además, el host inicializa su ventana de congestión en 1 MSS = 1 kB y pone la variable `ssthresh` en 6 kB. Sabemos, por otro lado, que en el quinto RTT los segmentos del host emisor atravesaron un router congestionado y en consecuencia fueron descartados por falta de espacio en sus *buffers*.
 - a. Completar la siguiente tabla donde se resume la evolución de las distintas variables del emisor durante los primeros siete RTTs de la conexión. ¿Pudo el host enviar la totalidad de los datos al cabo de estos RTTs?
 - b. Indicar qué impacto tiene inicializar la variable `ssthresh` en valores bajos. ¿Cómo repercute esto en este ejercicio?

RTT	estado	rwnd	cwnd	ssthresh	en vuelo	total enviado
0	Slow Start	8 kB	1 MSS = 1 kB	6 kB		
1						
2						
3						

Licenciatura en Tecnología Digital
TD4: Redes de Computadoras

4						
5						
6						