

# Control de Congestión

Resolución de ejercicios vistos en clase

17.10.2023

## 1. Primer ejercicio

### 1.1. Enunciado

Considere el efecto de usar Slow Start en una conexión TCP recién establecida ( $IW = 2 * SMSS$ ,  $SSTHRESH = 64KB$ ), que tiene un RTT de 10 mseg y sin congestión ni errores presentes en la red. La  $RWND$  es de 24KB y el  $SMSS$  es de 2KB. ¿Cuánto tiempo transcurre antes de que pueda ser enviada la primera ventana de recepción llena? (Asumir que el  $T_{tx}$  de una ventana es una componente despreciable del Delay total de la conexión)

### 1.2. Resolución

La idea acá es simular el intercambio de mensajes teniendo en cuenta que el  $T_{tx}$  de un ráfaga es mucho más chico que el  $T_{prop}$  de la conexión ( $T_{tx} \ll T_{prop}$ ). Esta es una condición fuerte pero asegura que los envíos se den de a ráfagas, lo cual simplifica la resolución del ejercicio, de manera de poder capturar la dinámica de intercambio de paquetes.

En el estado inicial, la conexión arranca con  $CWND = IW = 2 * SMSS$ . En el primer RTT se envía una ráfaga de dos segmentos llegando a un total de 4KB de datos en vuelo. Al momento de recibir cada ACK de vuelta, la ventana de congestión se incrementa como  $CWND+ = \min(N, SMSS)$  (Cabe aclarar que en este caso  $N=SMSS$ ). Esto deja la  $CWND = 4 * SMSS = 8KB$ . Así, la próxima ráfaga será de 8KB y volverán 4 ACKs de 2KB cada uno, por lo que la ventana llegará a 16KB al comienzo del próximo envío, que sería el tercero. El próximo envío (el tercero), se envían 16KB y vuelven 8 ACKs de 2KB cada uno  $\Rightarrow CWND=32KB$ . A partir de acá, en el 4to rtt, el envío continúa con  $FlightSize = \min(RWND, CWND) = \min(24, 32) = 24$ , momento en el cual se dice que se envía la primera ventana de recepción llena.

Esto puede quedar más claro usando una tabla como la siguiente, donde en cada línea está el estado de las variables de la conexión que necesitamos para resolver el ejercicio:

RTT	CWND	RWND	FlightSize
1	4KB	24KB	4KB
2	8KB	24KB	8KB
3	16KB	24KB	16KB
4	32KB	24KB	24KB
...			

## 2. Segundo ejercicio

### 2.1. Enunciado

En una conexión recién establecida con  $RTT=200ms$ , el host receptor siempre anuncia una *AdvertisedWindow* de 16KB. La red está cargada al punto que si una ráfaga fuera de 16KB o mas, se perderían todos los segmentos de la misma.

- a. ¿Cuánto vale la CWND luego de enviar un archivo de 40KB?
- b. 3 segundos después del envío del archivo, se envía otro archivo de 30KB ¿Cuánto tiempo tarda?

### 2.2. Resolución

Como se mencionó previamente, una forma sencilla de encarar estos problemas, es definir una tabla donde se lleva la cuenta de todas las variables que nos interesan para poder resolver el problema. Para esto, debemos decidir que variables involucrar. Primero, el RTT es el elemento principal que lidera la evolución de la conexión. Segundo, la ventana del control de flujo (RWND) y el control de congestión (CWND) que, el mínimo entre ambas, define la máxima ventana disponible para enviar. En tercer lugar, también tenemos que llevar el SSTHRESH dado que, ante la pérdida de una ráfaga, esta variable, se actualizará para marcar el fin de Slow Start, esta variable comienza valiendo un valor arbitrariamente alto, por ejemplo 64KB. Por último, llevamos cuenta de los datos en vuelo (FlightSize), dado que el control de congestión lo tiene en cuenta para ajustar la ventana y de los datos enviados hasta el momento (LastByteSent o LBS) que marca cuanto llevamos enviado con éxito hasta el momento.

RTT	CWND	RWND	SSTHRESH	FlightSize	LBS	Comentarios
1	4KB	16KB	64KB	4KB	4KB	La CWND inicia en $IW = 2 * SMSS$ y tomamos $SMSS = 2KB$ . Siempre, $RWND = 16KB$ porque, en este caso, la Advertised Window nunca cambia.
2	8KB	16KB	64KB	8KB	12KB	Al principio aumenta con Slow Start, dado que el SSTHRESH es arbitrariamente alto.
3	16KB	16KB	64KB	16KB	28KB	Aca los datos están en vuelo, y no hay respuesta de ACKs dado que se pierden todos los segmentos de la ráfaga.
4	16KB	16KB	64KB	16KB	28KB	TIME OUT, dado que $RTO=2*RTT$ .
5	2KB	16KB	8KB	2KB	14KB	La ventana se actualiza a $LW = 1 * SMSS = 2KB$ y el $SSTHRESH = \text{Max}(\text{FlightSize}/2, 2 * SMSS)$ .
6	4KB	16KB	8KB	4KB	18KB	Aca sigue con Slow Start hasta que la CWND alcance el SSTHRESH.
7	8KB	16KB	8KB	8KB	26KB	A partir de aca se aumenta con Congestion Avoidance a razón de un SMSS por RTT.
8	10KB	16KB	8KB	10KB	36KB	Aca se aproxima el aumento de la CWND. Tomamos 1 SMSS por RTT.
9	12KB	16KB	8KB	4KB	40KB	Solo faltan 4KB para llegar a los 40KB.

Se puede decir que la CWND termina en 12KB luego de enviar todos los datos pero, la transmisión con éxito, termina luego de la llegada de los ACKs que faltan. Por aproximación tomamos que este último RTT, la CWND, aumentó otro SMSS y termina valiendo **14KB**.

Para el segundo inciso, la idea es acordarse que, luego de un RTO sin transmitir, la CWND, se reinicia a  $RW = \min(IW, CWND) = 4KB$ . Una vez tenido esto en cuenta, se puede ver cuanto tardaría en enviarse los proximos 30KB, usando la tabla:

RTT	CWND	RWND	SSTHRESH	FlightSize	LBS	Comentarios
1	4KB	16KB	8KB	4KB	4KB	Comienzo a contar en términos de RTT desde 1 nuevamente. SSTHRESH se mantiene como venía hasta ahora porque no se aclara que deba cambiar.
2	8KB	16KB	8KB	8KB	12KB	Como $CWND < SSTHRESH$ aumento con Slow Start.
3	10KB	16KB	8KB	10KB	22KB	Aca comienzo Congestion Avoidance.
4	12KB	16KB	8KB	8KB	30KB	Al final de este RTT termina con éxito la transmisión.

Hicieron falta 4 RTTs para enviar los 30KB. La respuesta sería:  $4 * 200ms = 0,8seg$

### 3. Tercer Ejercicio

#### 3.1. Enunciado

Dada una conexión TCP recién establecida entre dos host para la cual el RTT es de  $50ms$ . Los dos host están separados por un sólo router que también conecta otras redes y está cargado a tal punto que cada vez que una ráfaga de paquetes es de 20KB o más, se descartan todos los paquetes de la ráfaga. El host emisor tiene que enviar un archivo bastante grande que se está transmitiendo por horas y el host receptor siempre anuncia una *AdvertisedWindow* de 28KB.

- Si se define que una conexión alcanza el *estado estacionario* en el momento que el SSTHRESH converge a un valor a partir del cual ya no cambia más. ¿Cuánto tiempo tarda la conexión en alcanzar el estado estacionario? ¿Cuál es el valor del SSTHRESH es dicho momento?
- Finalizada la transferencia, se cierra la conexión, y se inicia una nueva en la que el host receptor siempre anuncia una *AdvertisedWindow* de 18KB. Si esta nueva conexión tuviera que transferir el mismo archivo, ¿tardaría más o menos tiempo que la anterior? (*Suponer las mismas condiciones de congestión en el router*)

### 3.2. Resolución

RTT	CWND	RWND	SSTHRESH	FlightSize	LBS	Comentarios
1	4	28	64	4	4	-
2	8	28	64	8	12	-
3	16	28	64	16	28	-
4	32	28	64	28	56	-
5	32	28	64	28	56	TIME OUT
6	2	28	14	2	30	-
7	4	28	14	4	34	-
8	8	28	14	8	42	-
9	16	28	14	16	58	-
10	18	28	14	18	76	-
11	20	28	14	20	96	-
12	20	28	14	20	96	TIME OUT
13	2	28	10	2	78	ACA EL SSTHRESH ADOP- TA EL ULTIMO VALOR Y NO CAMBIA MAS
14	4	28	10	4	82	-
15	8	28	10	8	90	-
16	12	28	10	12	102	-
17	14	28	10	14	116	-
18	16	28	10	16	132	-
19	18	28	10	18	150	-
20	20	28	10	20	170	-
21	20	28	10	20	170	TIME OUT
22	2	28	10	2	152	ACA EL SSTHRESH YA DA LO MISMO

Bajo este diagrama la respuesta es, a partir del RTT 13  $\rightarrow 13 * 50ms = 650ms$

Para el segundo ítem, la *MaxWindow* del host emisor se mantendrá constante en 18KB regida por la RWND. El host receptor mantiene al emisor enviando datos justo por debajo del umbral de congestión. Esto hace que no se pierdan paquetes por congestión y que en cada RTT se envíen 18KB. En promedio, se enviarán más datos por RTT que en la conexión anterior, donde la red si llega a congestionarse y TCP ejecuta los mecanismos de recuperación que enlentecen el envío de datos.