

Certamen Final

Tiempo 100 min.

1.- En una conexión punto a punto la secuencia de acuses de recibo en el transmisor es la siguiente :

0, 1, 2, 3, 4, 0, 1, 2, 5, 6, 7, 1

Para las preguntas de **a)** a **d)** responda con un escenario posible que genere la secuencia de acuses de recibo indicada.

Un posible escenario es el siguiente:

#sec Tx:	0	1	2	3	4	5	6	7	0	(Timeout 5)	5	1	2	3	4	5	6	7	0	1	(timeout 0)	0		
Acuse				0	1	2	3	4				0	1	2	-	-	5	6	7			1		
Rx			0	1	2	3	4	-	6	7	0		5	1	2	3	4	5	6	7	-	1		0

Nota: Hay muchos otros. Por ejemplo usted pudo suponer que sólo se perdían los ACK que se muestran pero nunca un paquete de datos. Este parece ser el escenario más simple.

a) ¿Cuáles son los números de secuencia asignados por el transmisor en los paquetes transmitidos?

0,1,2,3,4,5,6,7,0,(retransmisión)5,1,2,3,4,5,6,7,0,1,0 (retransmisión) (25 pts)

Nota: Dado que hubo otra interpretación (no basada en el escenario) 0,1,2,3,4,5,6,7, a quien respondió así le asigné 10 pts. Y la prueba fue evaluada sobre un total de 685 puntos en lugar de 700.

b) ¿Cuántos bits son requeridos para representar el número de secuencia?

3 bits (25 pts)

c) ¿Cuál es el tamaño de la ventana?

4 paquetes (25 pts)

d) ¿Se perdió algún paquete? De ser así, indique cuál.

En este escenario se perdieron el 6º y 17º paquetes y los acuses de recibo del 12º y 13º paquetes. (25 pts)

Nota: los ACK también son paquetes.

2.-Piense, y en no más de 5 líneas explique ¿por qué TCP intenta enviar segmentos no mayores al MTU (Maximum Transmission Unit)?.

Esto es conveniente para aumentar la utilización del enlace. El uso de segmentos mayores que la MTU obliga a fragmentar, y la pérdida de un fragmento IP implica descartar todo el segmento haciendo inútil el transporte de los fragmentos que si llegaron. (100 pts)

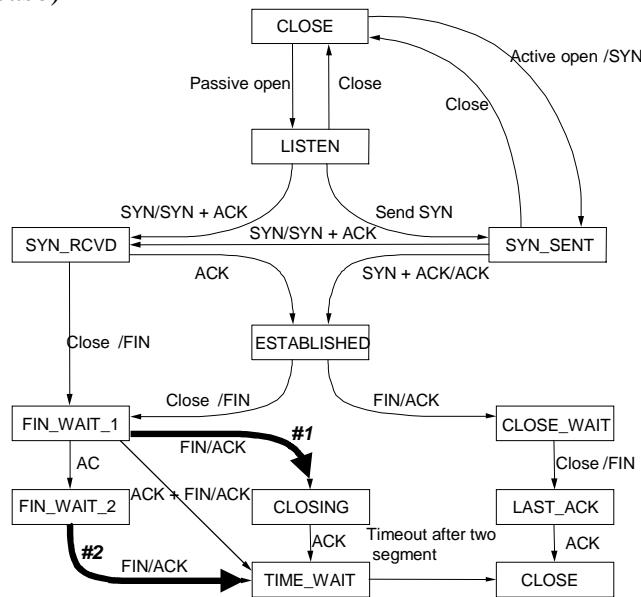
Nota: La fragmentación la hace IP localmente si es necesaria, luego la sigue haciendo IP en los routes si se requiere nuevamente. El reensamble también lo hace IP y sólo en el destino. TCP es un protocolo de extremo a extremo, no se percata de la fragmentación de IP. TCP pasa a IP un segmento, IP lo hace llegar a su par en la máquina destino y éste se lo pasa a TCP.

3.- Supongamos que dos alumnos trabajando en aragorn.elo.utfsm.cl ejecutan cada uno una sesión ssh hacia joshua.elo.utfsm.cl. Si en ambos casos los alumnos contactan mismo

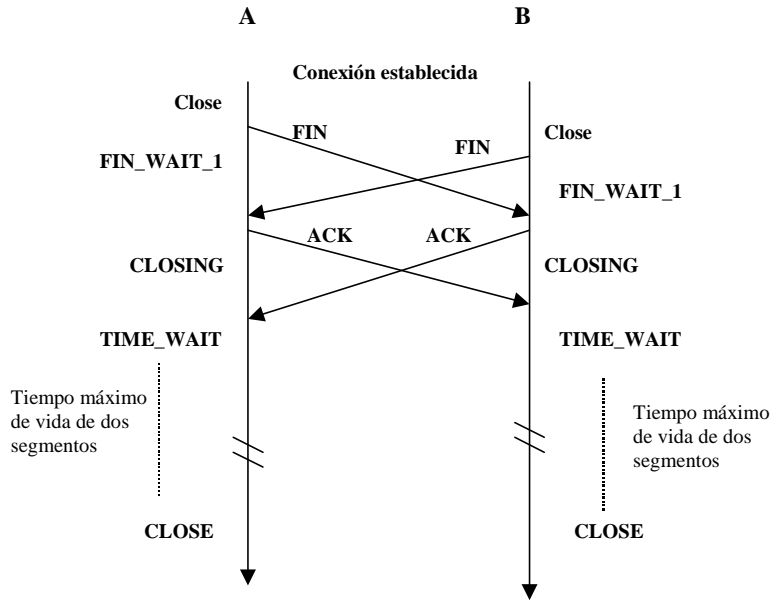
puerto ssh, ¿cómo joshua logra no confundirse y enviar las respuestas a cada usuario aun cuando ambos tienen la misma dirección IP origen?

La sesión ssh se mantiene gracias a una conexión TCP entre la máquina cliente y la servidora. joshua logra identificar cada usuario (en realidad la conexión) porque la conexión TCP queda definida por cuatro componentes (puerto local, IP local, puerto remoto, IP remota) En este caso todos los parámetros son comunes excepto el puerto remoto, el cual es diferente para cada cliente ssh saliendo de la misma IP. Es así como Joshua responde al puerto del cliente que corresponde. (100 pts)

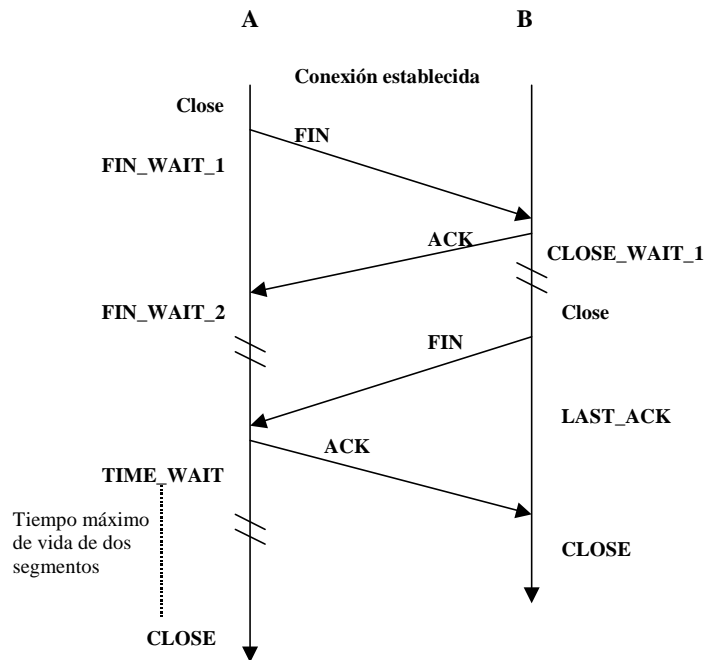
4.- Haga diagramas temporales indicando la secuencia de eventos que ocurren en ambos extremos de la conexión para pasar por las transiciones rotuladas como #1 y #2. (Un diagrama para cada caso)



Caso #1: Ambos cierran al mismo tiempo (50 pts)



Caso #2: A cierra primero (50 pts)



Nota: El requerimiento de close viene de la aplicación local hacia TCP, no del extremo opuesto de la conexión.

5.- Suponga que TCP calcula los timeout usando el algoritmo de Jacobson&Karels. Si en un instante $EstRTT = 4.0$, $Desviation=0.5$, y los siguientes RTT medidos son 1.0, ¿cuánto tiempo pasa hasta que el timeout llega a un valor inferior a **5.5**? Use $\delta = 1/8$.

Algoritmo de Jacobson&Karels:

$$\begin{aligned} \text{Diff} &= \text{sampleRTT} - \text{EstRTT} \\ \text{EstRTT} &= \text{EstRTT} + (\delta \times \text{Diff}) \\ \text{Dev} &= \text{Dev} + \delta (|\text{Diff}| - \text{Dev}) \\ \text{TimeOut} &= \text{EstRTT} + 4 \times \text{Dev} \end{aligned}$$

SampleRTT	Diff	EstRTT	Dev	Timeout
-	-	4,00	0,50	6,00
1,00	-3,00	3,63	0,81	6,88
1,00	-2,63	3,30	1,04	7,45
1,00	-2,30	3,01	1,20	7,79
1,00	-2,01	2,76	1,30	7,95
1,00	-1,76	2,54	1,36	7,96
1,00	-1,54	2,35	1,38	7,86
1,00	-1,35	2,18	1,37	7,68
1,00	-1,18	2,03	1,35	7,43
1,00	-1,03	1,90	1,31	7,14
1,00	-0,90	1,79	1,26	6,83
1,00	-0,79	1,69	1,20	6,49
1,00	-0,69	1,60	1,14	6,15
1,00	-0,60	1,53	1,07	5,81
1,00	-0,53	1,46	1,00	5,47

14,00 Unidades de RTT

(90 pts)

Pasan 14 unidades de tiempo (igual unidad que las de RTT) hasta que el timeout llega a un valor inferior a 5.5. (10 pts)

6.-a) Explique por qué la curva de potencia sube a la izquierda de a carga óptima y luego baja a su derecha.

La potencia sube en la zona izquierda porque al aumentar la carga inicialmente hay un aumento de throughput y con RTT más o menos constantes, dado que aun estamos lejos de congestión y por ello los routers mantienen sus colas muy cortas. (40 pts)

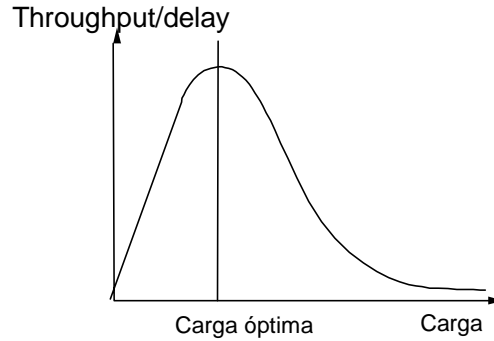
La potencia baja en zona de la derecha porque al seguir aumentando la carga el throughput ya no experimenta aumentos significativos ya que nos acercamos a la capacidad de los enlaces. Por otro lado el RTT aumenta por el crecimiento de las colas en los routers. Esto hace bajar el cuociente. (40 pts)

b) ¿En qué zona el retardo es mayor?

Derecha (10 pts)

c) ¿En qué zona el Throughput es mayor?

En la derecha del carga máxima (10 pts)



Nota: Debe ser más o menos claro para ustedes que el retardo aumenta en la medida que la carga aumenta por las razones dadas arriba. En un comienzo aumenta lentamente y luego lo hace más rápidamente cuando se inicia la congestión. La potencia alcanza su máximo cuando el cociente throughout/delay se mantiene constante al aumentar la carga. Dado que el retardo siempre está creciendo, para que el cociente sea constante, el throughput debe crecer de la misma forma. Luego en la vecindad de la carga óptima el throughput aun aumenta. Esto explica por qué el valor máximo para el throughput debe estar en algún punto de la derecha de la carga óptima (presumiblemente no muy lejos).

7.-

a) Mencione una ventaja de Encolamiento Equitativo (Fair Queuing) frente al FIFO.

FIFO maneja sólo una cola y por lo tanto no distingue flujos distintos, por lo cual no es posible distribuir la capacidad a cada flujo en forma equitativa. FQ no permite que flujos con tráfico excesivo dañen a flujos con tráfico moderado.

40 pts

b) Mencione una ventaja de RED frente a DECbit

DECbit requiere de manipulaciones adicionales (mayor procesamiento) a las realizadas por los protocolos de extremo a extremo. Se debe monitorear el bit de congestión. En RED no hay efecto adicional al ya considerado en las máquinas de los extremos.

30 pts.

Nota: DECbit no incorpora un nuevo bit, sino hace uso de uno ya existente cuyo uso no estaba definido (ver encabezado IP)

c) Mencione una ventaja de TCP Vegas frente a RED.

TCP Vegas es una solución independiente de si los routers intermedios en el camino fuente destino implementan o no algún esquema para abolir congestión. RED es una solución que requiere la participación de routers y hosts extremos.

30 pts.

Éxito en sus resultados de fin de semestre !