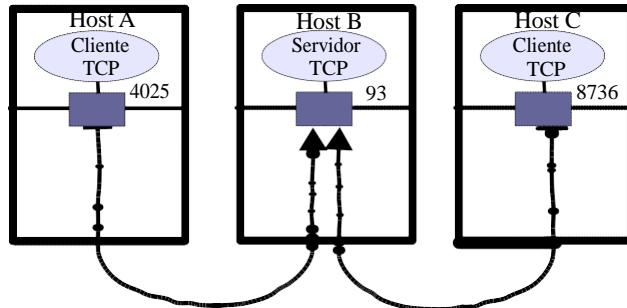


Ejercicios solucionados

Ejercicio 1:

Supongamos que tenemos 3 hosts: A, B y C. En los hosts A y C tenemos dos clientes que usan los puertos 4025 y 8736 respectivamente. En el host B tenemos un servidor TCP que ofrece su servicio en el puerto 93 (como muestra la figura).



El cliente en A se conecta con el servidor en B y posteriormente, el cliente en C se conecta también a este servidor en B.

- a) ¿Cómo puede el servidor en B diferenciar las dos conexiones, si ambas son gestionadas a través de un mismo puerto (el 93)?.
- b) ¿Y si el cliente en C hubiese obtenido como puerto local el 4025?
- c) ¿Produciría algún problema añadir en el host B un servidor UDP que ofrezca su servicio en el puerto 93? Justifica la respuesta.
- d) ¿Qué diferencia/s presenta un servidor concurrente frente a uno iterativo desde el punto de vista de los clientes?

Solución:

- a) Desde el punto de vista de la aplicación servidora, cada cliente está siendo servido por un socket distinto, a pesar de que ambos sockets compartan el mismo puerto. Desde el punto de vista de TCP, las conexiones se diferencian claramente puesto que una conexión TCP se identifica mediante cuatro parámetros: dirección IP origen, dirección IP destino, puerto origen y puerto destino. En ese caso, no existe ninguna ambigüedad, ya que las direcciones IP y los puertos de los clientes son diferentes.
- b) Tampoco se plantea ambigüedad ya que la dirección IP origen es distinta.
- c) El hecho de que se ofrezca un servicio UDP en el mismo puerto no causa ningún problema, ya que los paquetes de datos se demultiplexan en función del tipo de protocolo de transporte a nivel de IP, antes de que se inspeccionen los números de puerto.
- d) Desde el punto de vista de los clientes, el tiempo de respuesta de un servidor concurrente desde que el cliente solicita el servicio hasta que establece conexión con el servidor es siempre fijo y acotado, independientemente de cuantos clientes están accediendo al servicio. En servidores no concurrentes (secuenciales) un cliente debe esperar a que se sirvan todos los clientes que están encolados antes que él para conseguir conectar con el servidor.

Ejercicio 2: Encuentra los cinco errores de la siguiente cabecera TCP, sabiendo que corresponde al primer segmento de establecimiento de una conexión realizada por un cliente estándar de HTTP hacia el correspondiente servidor estándar, y que el campo de opciones no está vacío. Razona la respuesta.

Puerto TCP origen:	120	
Puerto TCP destino:	80	
Número de secuencia:	1400	
Número de reconocimiento:	0	
Longitud de la cabecera:	5	
Reservado:	0	
Código:	URG = 0	RST = 0

	ACK = 1	SYN = 1
	PSH = 0	FIN = 0
Tamaño de la ventana (en decimal)	86535	
Checksum (en hexadecimal)	9FB0	
Opciones (en bytes hexadecimales)	02- 04 - 05 – B4 (MSS = 1460)	
	04 – 02 (activar reconocimiento selectivo)	
Relleno:	(sin relleno)	

Ejercicio 3: El control de flujo TCP, basado en ventana deslizante, dispone de una indicación de ventana (buffer disponible en el otro extremo) que limita la inyección de segmentos en la conexión. El tamaño máximo que se puede indicar es de 64 KB. Esta limitación, ¿podría afectar a las prestaciones de TCP cuando se utilizan redes de alta velocidad (Ej.: Gigabit Ethernet ~1Gbps) con RTTs del orden de 2 ms?

Solución:

A 1 Gbps el tiempo de transmisión de 64 KB será de $64 \times 8 \times 10^3 / 10^9 = 0.000512$ segundos, es decir, 0.512 ms. El emisor tardará este tiempo en transmitir toda su ventana de transmisión, mientras que el primer reconocimiento llegará, como pronto, 2 ms después. Eso significa que tras la transmisión de la ventana, habrá de esperar $2 - 0.512 = 1.488$ ms antes del primer ACK. Por tanto, sólo se está transmitiendo por la línea un 25% del tiempo, aproximadamente.

Otra forma de verlo es la siguiente: El funcionamiento óptimo en un protocolo de ventana deslizante se consigue cuando no es necesario esperar nunca a la recepción de un reconocimiento, es decir, que tras la transmisión del primer segmento de la ventana, el reconocimiento de dicho segmento, que llegará tras un intervalo de RTT (Round-Trip Time), se reciba durante la transmisión del último segmento de la ventana (a partir de este tamaño de ventana las prestaciones no mejorarán, ya que habremos obtenido una utilización del 100%). En el caso propuesto, si el RTT es de 2 ms, el tamaño de la ventana debería ser de

$$V_{tx} = 2 \text{ ms} \times 1 \text{ Gbps} = 2 \times 10^{-3} \times 10^9 = 2 \times 10^6 \text{ bits}$$

o lo que es lo mismo 250 KB. Esto significa que antes del primer reconocimiento tendríamos tiempo de haber enviado 250 KB. Puesto que la ventana máxima en TCP es de 64KB, la pérdida de prestaciones por las limitaciones del protocolo TCP resulta apreciable.

Ejercicio 4: Supongamos que un proceso P (en el host A1) quiere establecer una conexión TCP con el proceso Q (en el host E1):

a) Suponiendo que no se pierden datagramas ni se producen retransmisiones, indique los segmentos que se intercambiarían ambos TCP's en las siguientes situaciones:

1. Q realiza una apertura pasiva, y P una apertura activa (suponer que $NSI(P)=0$, $WIN(P)=20$, $NSI(Q)=13500$, $WIN(Q)=30$).
2. A continuación P envía 100 octetos a su entidad TCP para que los envíe a Q.
3. Después el proceso P le indica a su TCP el cierre de conexión. Supongamos que Q también está dispuesto a cerrar cuando le llegue el aviso de cierre de P.

b) Lo mismo que el apartado a), pero suponiendo que Internet pierde un segmento de cada 3 que envíe el TCP del proceso P. Los segmentos que envía el TCP del proceso Q no se pierden. El temporizador de retransmisión (RTO) de P tiene un valor igual a dos RTT's.

NOTA: NSI (P) representa el número de secuencia inicial del proceso P, mientras que WIN(P) refleja el tamaño de ventana inicial que escoge el TCP del proceso P. No se emplean reconocimientos retrasados, y el MSS=100 bytes. La respuesta ha de reflejarse en una tabla con formato:

Origen/ Nº secuencia / Flags / Nº reconocimiento / Datos (indicando byte inicial y final)

Solución:

a) 1)

Proceso	Nº secuencia	Flags	Nº reconocimiento	Datos
P	0	SYN	-----	-----
Q	13500	SYN,ACK	1	-----
P	1	ACK	13501	-----

2)

Proceso	Nº secuencia	Flags	Nº reconocimiento	Datos
P	1	ACK	13501	1..30
Q	13501	ACK	31	-----
P	31	ACK	13501	31..60
Q	13501	ACK	61	-----
P	61	ACK	13501	61..90
Q	13501	ACK	91	-----
P	91	ACK	13501	91..100
Q	13501	ACK	101	-----

3)

Proceso	Nº secuencia	Flags	Nº reconocimiento	Datos
P	101	FIN, ACK	13501	-----
Q	13501	FIN,ACK	102	-----
P	102	ACK	13502	-----

b) 1) Igual que en a). Ya se han enviado dos segmentos (SYN y ACK), por lo que el siguiente segmento se perderá.

2)

Proceso	Nº secuencia	Flags	Nº reconocimiento	Datos
P	1	ACK	13501	1..30
Este segmento se pierde, por lo que tras cierto tiempo vencerá el temporizador				
P	1	ACK	13501	1..30
Q	13501	ACK	31	-----
P	31	ACK	13501	31..60
Q	13501	ACK	61	-----
P	61	ACK	13501	61..90
Este segmento se pierde, por lo que tras cierto tiempo vencerá el temporizador				
P	61	ACK	13501	61..90
Q	13501	ACK	91	-----
P	91	ACK	13501	91..100
Q	13501	ACK	101	-----

3)

Proceso	Nº secuencia	Flags	Nº reconocimiento	Datos
P	101	FIN, ACK	13501	-----
Este segmento se pierde, por lo que tras cierto tiempo vencerá el temporizador				
P	101	FIN, ACK	13501	-----
Q	13501	FIN,ACK	102	-----
P	102	ACK	13502	-----

Ejercicio 5: Un emisor ha enviado los segmentos 1 al 50. Cada uno de ellos con 512 bytes de datos. El emisor recibe un ACK con valor 15873 ($31 \times 512 = 15872$), y después 3 ACKs duplicados con valor 15873.

- a) Basándose en esta información, ¿qué segmento(s) puede suponer el emisor que se han perdido? ¿Y cuáles puede considerar que se han recibido correctamente?
- b) El mismo emisor reenvía el segmento supuestamente perdido y recibe como respuesta un ACK con valor 18433 ($36 \times 512 = 18432$). De los 50 segmentos enviados inicialmente, ¿puede suponer el emisor que se ha perdido alguno? En caso afirmativo, ¿cuál o cuáles? ¿Qué segmentos puede considerar como recibidos correctamente?

Solución:

- a) El primer ACK recibido con valor 15873 indica que los 31 primeros segmentos han llegado bien. Los tres ACKs duplicados con el valor 15873 indican que el siguiente segmento no ha llegado en orden, por lo que podemos asumir que el segmento 32 (con los bytes 15873 a 16384) se ha perdido. Respecto a los segmentos posteriores al 32 no podemos asumir nada, excepto que tres de ellos han llegado, pero no sabemos cuales.
- b) Tras recibir este nuevo ACK, sabemos que se había perdido el segmento 32 y que el 33, 34, 35 y 36 llegaron correctamente, puesto que se ha recibido el reconocimiento para ellos. Para el segmento 37 no hemos recibido ningún reconocimiento, por lo que podemos asumir que se ha perdido. Para los segmentos posteriores al 37 no tenemos ninguna información.

Ejercicio 6: El host B recibe peticiones de conexión TCP desde un host remoto llamado "A". Realiza, para cada uno de los escenarios siguientes, un esquema gráfico detallando qué segmentos se intercambiarían después de la llegada de los segmentos que se indican. No hay que detallar ningún segmento de la posible fase de transferencia de datos.

- a) Llega una petición de conexión desde "A", con número de secuencia 34, a un puerto donde no hay ningún servidor escuchando.
- b) Llega una petición de conexión desde "A", con número de secuencia 34, a un puerto donde hay un servidor escuchando. Antes de que se haya completado el establecimiento de la conexión en "B" llega, desde "A", una retransmisión de la petición anterior.
- c) "B" envía una petición de conexión a "A", con número de secuencia 75, a un puerto donde hay un servidor escuchando. Instantes después, "B" recibe una petición de conexión desde "A", con número de secuencia 34, a un puerto donde también hay un servidor escuchando.
- d) Llega una petición de conexión desde "A", con número de secuencia 34. Tras enviar "A" la petición de conexión, el computador A es reiniciado. Cuando "B" contesta esta petición de conexión, el computador A vuelve a estar totalmente operativo, ¿cuál sería su respuesta?

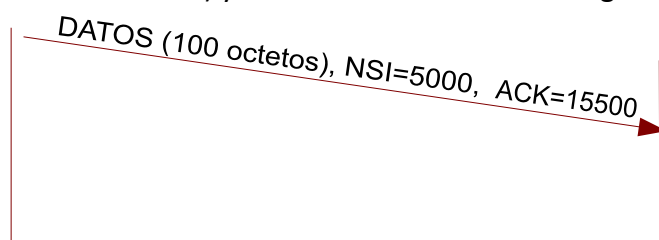
Ejercicio 7: Un usuario quiere conectarse al servidor web de un computador que en realidad no tiene ningún servidor web en ejecución, por lo que la conexión no se podrá establecer. Llamaremos "computador A" al host en el que el usuario está ejecutando el navegador, y "computador B" al host en el que supuestamente está en marcha el servidor web. Completa la siguiente tabla, especificando la secuencia de segmentos que tendrá lugar entre los dos computadores en el supuesto anterior.

Origen (computador)	Nº secuencia	Flags	Nº ACK	Datos
A	51	SYN		
B				

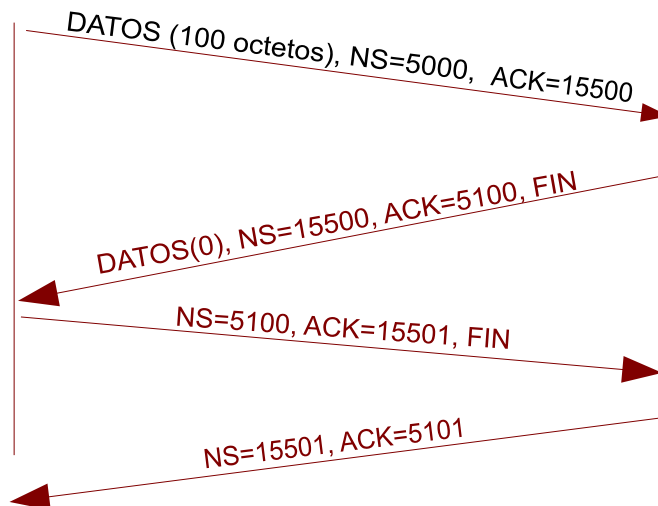
Solución:

Origen (computador)	Nº secuencia	Flags	Nº ACK	Datos
A	51	SYN		
B	xxxxxxx	RST SYN	52	-

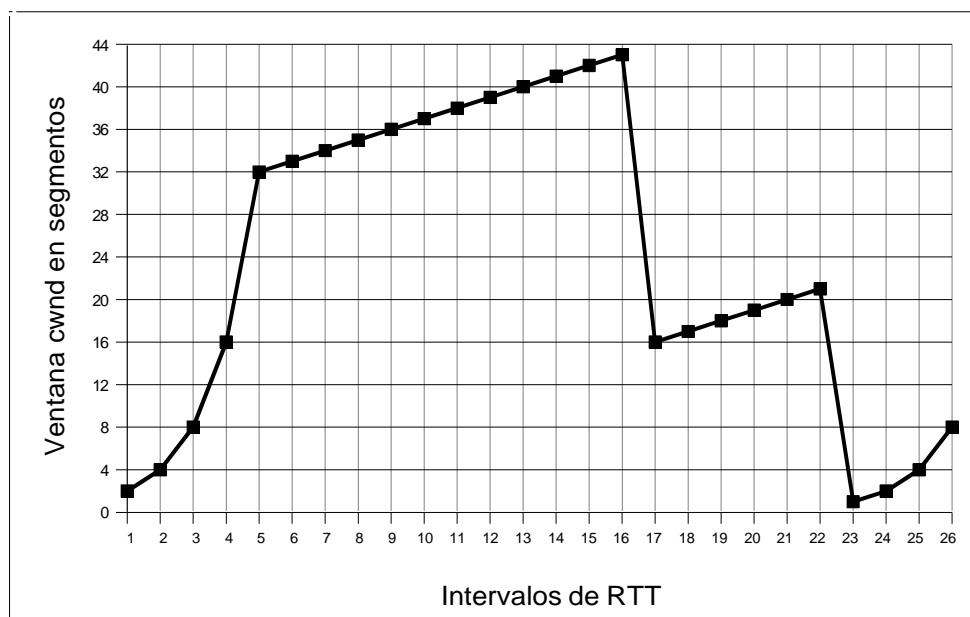
Ejercicio 8: Suponiendo que no se producen errores y que ninguno de los dos extremos envía ya más datos, completa la siguiente secuencia de segmentos TCP hasta que la conexión quede cerrada en los dos extremos. Indica para cada segmento: *flags* activados, valor del campo ACK (sólo si el flag ACK está activado) y número de secuencia del segmento.



Solución:



Ejercicio 9: Considera la siguiente representación del tamaño de la ventana de congestión en función del tiempo. Para el control de la congestión de TCP, responde a las siguientes cuestiones: (Cuando se pregunte tamaño de ventanas, indícalo en segmentos).

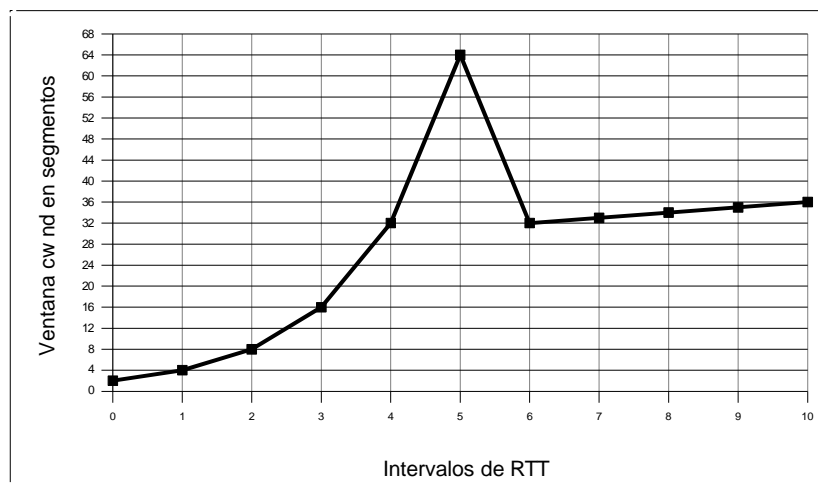


- Identifica los intervalos en los que actúa el mecanismo arranque lento (slow-start).
- Identifica los intervalos en los que actúa el mecanismo de evitación de la congestión (congestion avoidance).
- A continuación del RTT 16, ¿cómo se detecta la pérdida del segmento? ¿Y a continuación del RTT 22?
- ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 2?
- ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 18?
- ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 24?
- ¿Durante que RTT se envía el segmento 70?
- Suponiendo que tras el RTT 26 se detecta la pérdida de un paquete por la recepción de tres ACKs duplicados, ¿cuáles serán los valores de la ventana de congestión y del umbral?

- a) Desde el RTT 1 hasta el 5 y desde el 23 al 25.
- b) Desde el RTT 5 hasta el 16, desde el 17 al 22 y parte del 25 al 26.
- c) En el RTT 16 se detecta la recepción de tres ACK's duplicados. En el RTT 22 vence un temporizador.
- d) El valor en RTT 2 es 32 segmentos.
- e) El valor en RTT 18 es 16 segmentos.
- f) El valor en RTT 24 es 10'5 segmentos.
- g) En el RTT 5 hemos enviado $2+4+8+16+32=62$ segmentos, luego el 70 se enviará en el RTT 6
- h) En el RTT 26 se envían 8 segmentos. La ventana de transmisión, por tanto, tendrá tamaño 8. al recibirse los tres reconocimientos duplicados, y aplicando las fórmulas vistas en clase:

$$\text{Umbral} = V_{\text{trans}}/2 = 8/2 = 4 ; V_{\text{cong}} = \text{Umbral}=4.$$

Ejercicio 10: La gráfica siguiente representa la evolución de una ventana de congestión TCP en función del tiempo. La ventana de permisos del receptor (WIN) es siempre igual a 100 segmentos (100*MSS bytes). Responde a las siguientes preguntas:



- a) Indica qué mecanismos están actuando y durante qué intervalos.
- b) ¿Qué ocurre durante el RTT 5? ¿Cómo se detecta?
- c) ¿Cuál es el valor del umbral durante el RTT 3? ¿Y durante el RTT 8?
- d) Si durante el RTT 10 vence un temporizador, ¿cuáles serán los valores de la ventana de congestión y del umbral en el RTT 11?

Solución:

- a) *Slow-start* actúa desde RTT=0 hasta RTT=5 y evitación de congestión desde RTT=6 hasta RTT=10
- b) En RTT=5 se produce la pérdida de un segmento. Se detecta mediante la llegada de tres reconocimientos duplicados.
- c) En RTT=3, **umbral**=100*MSS. En RTT=8, **umbral**=32*MSS.
- d) Tras un *Timeout*, $v_{\text{cong}} = 1 \cdot \text{MSS}$, **umbral** = $V_{\text{trans}}/2 = 18 \cdot \text{MSS}$.

Ejercicio 11: El proceso pA en el computador A y el proceso pB en el computador B se disponen a establecer una conexión TCP. A lo largo de toda la conexión, los tamaños de la ventana que pA y pB declararán en sus segmentos permanecerán constantes con los valores siguientes: WIN(pA) = WIN(pB) = 300 bytes. Suponiendo que no se pierden paquetes ni se producen retransmisiones, describe el intercambio de segmentos entre pA y pB en las situaciones siguientes (véase la nota al final del enunciado con el formato a seguir):

- a) Establecimiento de conexión entre pA y pB (pA realiza una apertura activa y pB una apertura pasiva). Suponed que durante este proceso se intercambian los siguientes números de secuencia iniciales: NSI(pA) = 1000 y NSI(pB) = 5000.
- b) A continuación, pA envía 700 bytes a pB. Para este apartado supondremos, que durante la fase de establecimiento de la conexión, ambos procesos han acordado intercambiar

segmentos de tamaño máximo 100 bytes (MSS = 100 bytes). Siempre que sea posible pA envía segmentos del tamaño MSS. Se aplica una política de reconocimientos retrasados. La transferencia de información debe seguir el protocolo de arranque lento (slow-start). No se produce la pérdida ni el cambio de orden de ningún segmento.

NOTA: La respuesta ha de reflejarse en una tabla con formato:

Origen/ N° secuencia / Flags / N° reconocimiento / Datos (indicando byte inicial y final)

Solución:

a)

Origen	N° secuencia	Flags	N° reconocimiento	Datos
p _A	1000	SYN	-----	-----
p _B	5000	SYN,ACK	1001	-----
p _A	1001	ACK	5001	-----

b)

Origen	N° secuencia	Flags	N° reconocimiento	Datos
p _A	1001	ACK	5001	1001..1100
p _A	1101	ACK	5001	1101..1200
p _B	5001	ACK	1201	-----
p _A	1201	ACK	5001	1201..1300
p _A	1301	ACK	5001	1301..1400
p _A	1401	ACK	5001	1401..1500
p _B	5001	ACK	1401	-----
p _A	1501	ACK	5001	1501..1600
p _A	1601	ACK	5001	1601..1700
p _B	5001	ACK	1501	-----
p _B	5001	ACK	1701	-----

Ejercicio 12: Un cliente y un servidor se comunican mediante el protocolo TCP. La aplicación cliente envía una petición de 30 bytes al servidor. La respuesta del servidor es un mensaje de 1476 bytes, tras el que iniciará el cierre de la conexión. Sabemos que el MSS que emplean los dos extremos es de 512 bytes, NSI(C) = 7.000, NSI(S) = 15.000 (NSI es el número de secuencia inicial), WIN(C) = WIN(S) = 2048 y que, para este ejercicio, el tamaño inicial de la ventana de congestión es dos segmentos (2*MSS bytes). Ambos extremos emplean reconocimientos retrasados. Describe la evolución de la conexión TCP, desde el establecimiento hasta el cierre de la conexión.

La respuesta ha de reflejarse en una tabla con formato:

Origen/ N° secuencia / Flags / N° reconocimiento / Datos (indicando byte inicial y final)

Solución:

Origen (C o S)	N° secuencia	Flags	N° ACK	Datos
C	7000	SYN	---	---
S	15000	SYN, ACK	7001	---
C	7001	ACK	15001	---
C	7001	ACK	15001	1..30 (7001..7030)
S	15001	ACK	7031	1..512 (15001..15512)
S	15513	ACK	7031	513..1024 (15513..16024)
C	7031	ACK	16025	---
S	16025	ACK	7031	1025..1476 (16025..16476)
C	7031	ACK	16477	---
S	16477	FIN, ACK	7031	---
C	7031	FIN, ACK	16478	---
S	16478	ACK	7032	---