

Ejercicios

TEMA 4: NIVEL DE TRANSPORTE

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

a1

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

a2

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	-----

a3

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

Ejercicio 4

- b1. Igual que en a). Ya se han enviado dos segmentos (SYN y ACK), por lo que el siguiente segmento se perderá.
- b2

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	-----

- b3

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	-----

- 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?

- 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.*

- 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.

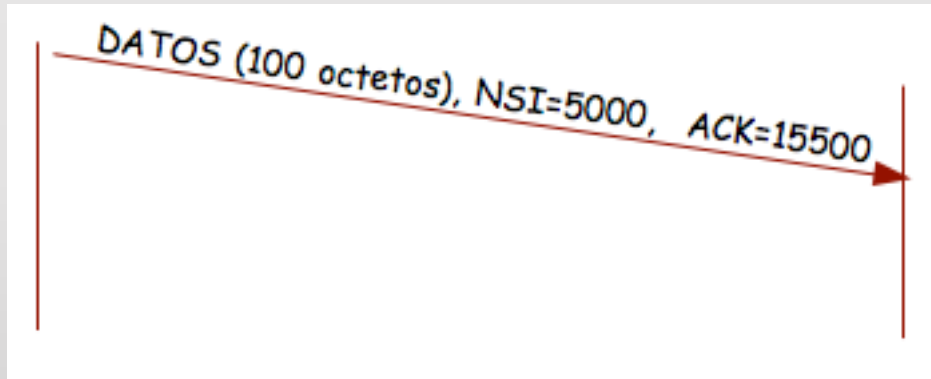
Origen (computador)	Nº secuencia	Flags	Nº ACK	Datos
A	51	SYN		
B				

Solución:

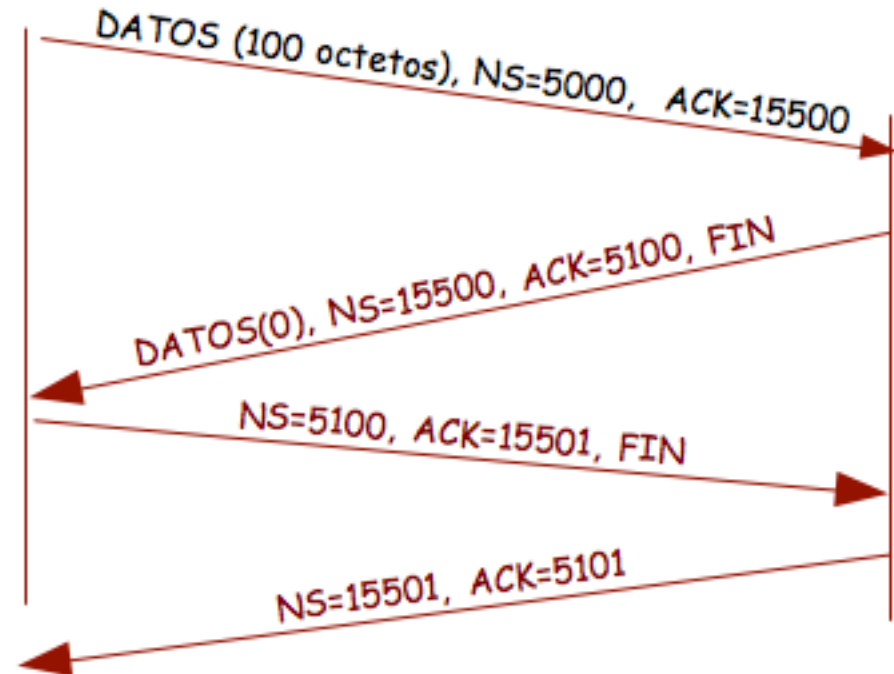
Origen (computador)	Nº secuencia	Flags	Nº ACK	Datos
A	51	SYN		
B	xxxxxxx	RST ACK	52	-

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.

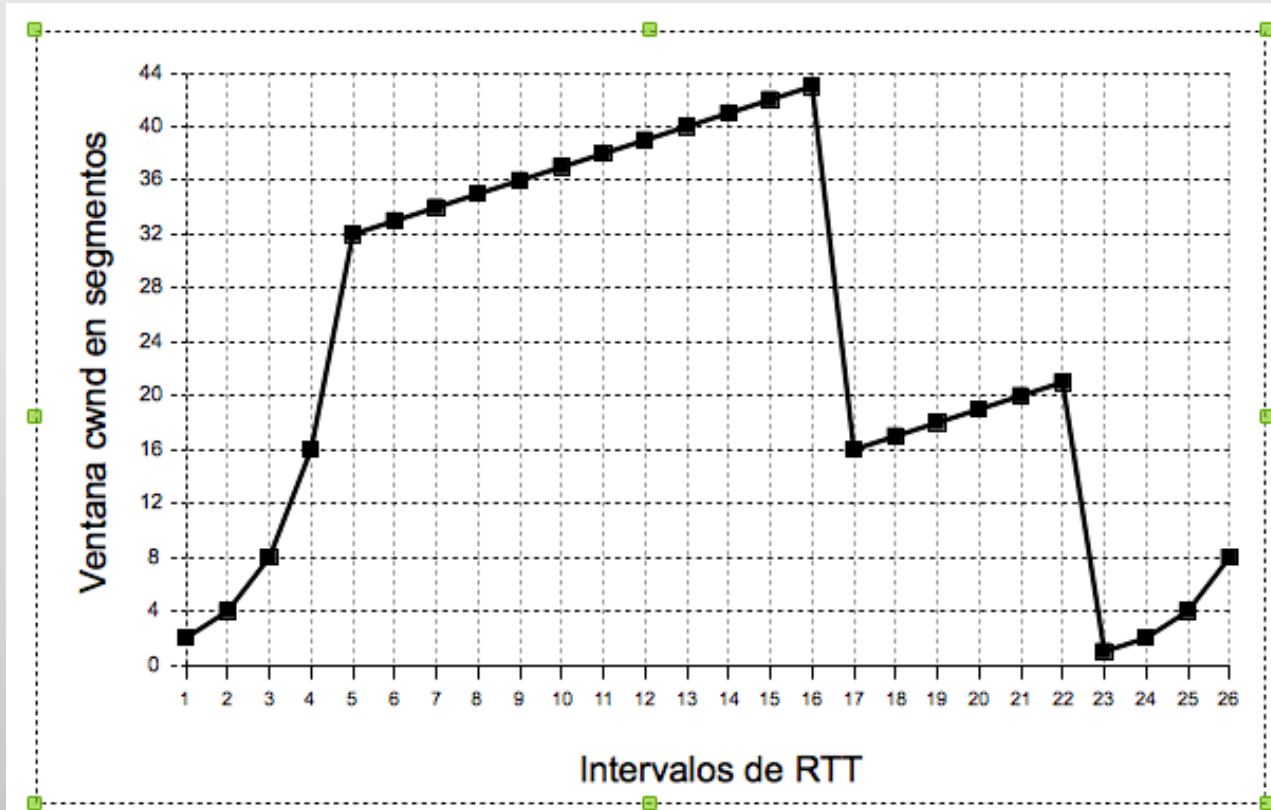


Solución:



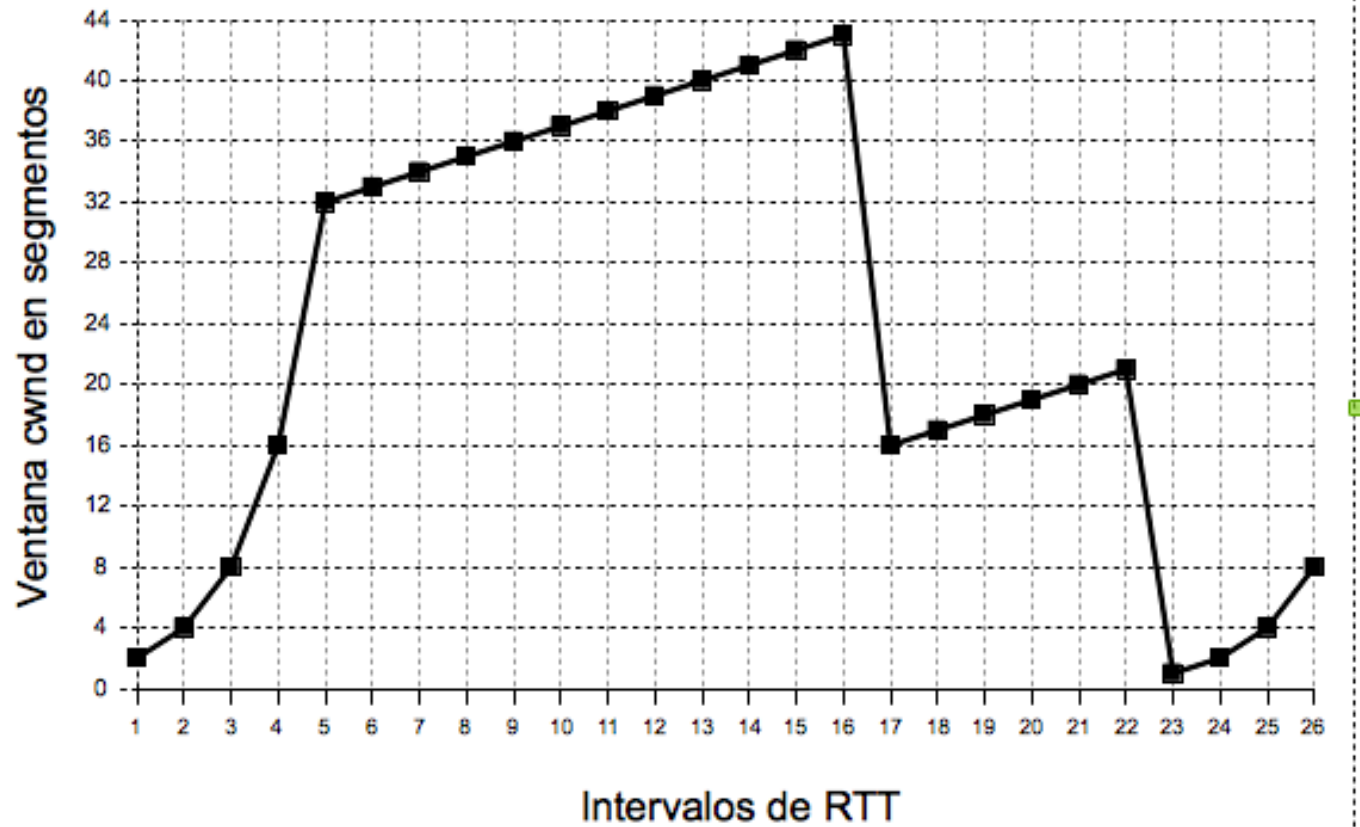
Ejercicio 9

- 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?

Ejercicio 9

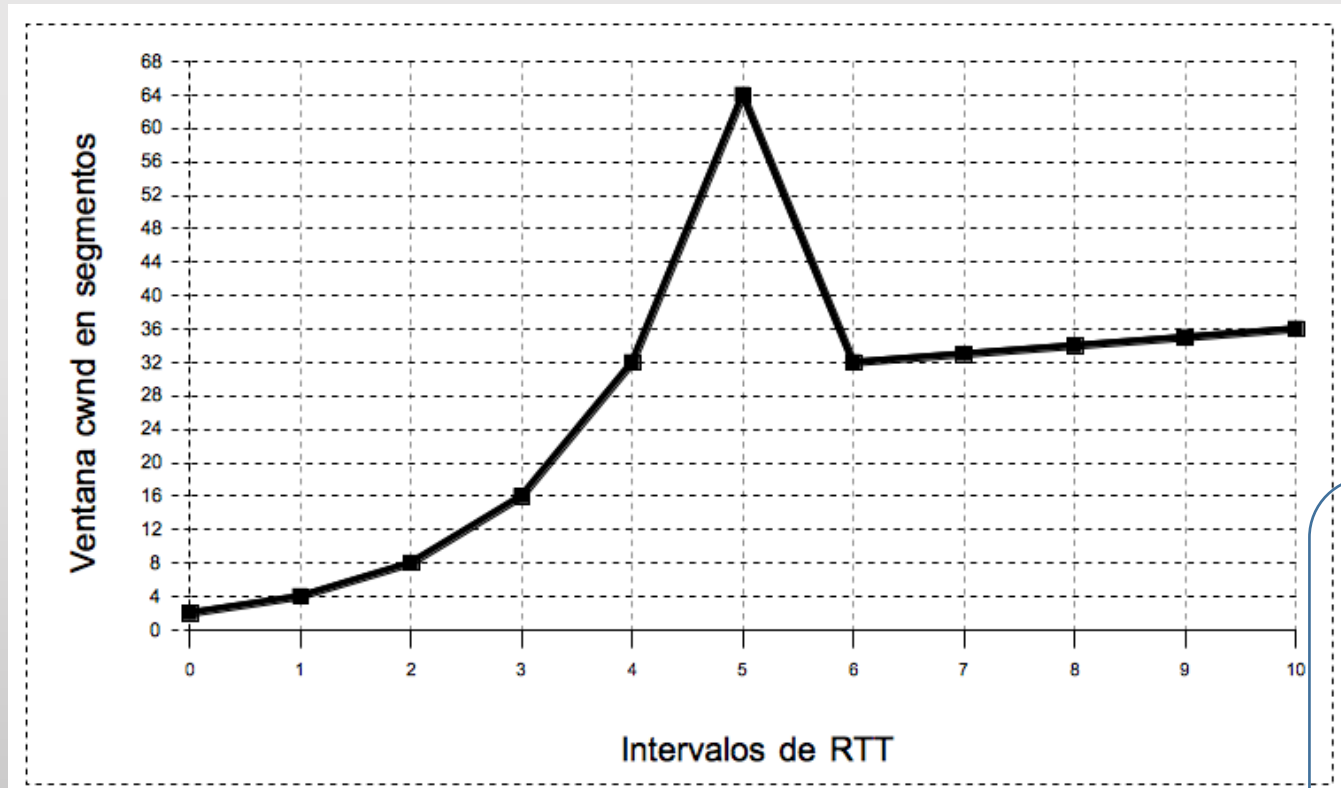


- h) Suponiendo que tras el RTT 26 se detecta la pérdida de un paquete por la recepción de tres ACKs duplicados, ¿cuáles serán los valores de la ventana de congestión y del umbral?
En el RTT 26 se envían 8 segmentos. La ventana de transmisión, por tanto, tendrá tamaño 8. al recibirse los tres reconocimientos duplicados, y aplicando las fórmulas vistas en clase:
$$\text{Umbral} = V_{\text{trans}} / 2 = 8 / 2 = 4 ; V_{\text{cong}} = \text{Umbral} = 4.$$

- a) Identifica los intervalos en los que actúa el mecanismo arranque lento (slow-start).
Desde el RTT = 1 hasta el RTT=5 y desde RTT= 23 a RTT=26
- b) Identifica los intervalos en los que actúa el mecanismo de evitación de la congestión (congestion avoidance).
Desde el RTT 5 hasta el 16, desde el 17 al 22.
- c) A continuación del RTT 16, ¿cómo se detecta la pérdida del segmento? ¿Y a continuación del RTT 22?
En el RTT 16 se detecta la recepción de tres ACK's duplicados. En el RTT 22 vence un temporizador.
- d) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 2?
El valor en RTT 2 es 32 segmentos.
- e) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 18?
El valor en RTT 18 es 16 segmentos.
- f) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 24?
El valor en RTT 24 es 10'5 segmentos.
- g) ¿Durante que RTT se envía el segmento 70?
En el RTT 5 hemos enviado 2+4+8+16+32=62 segmentos, luego el 70 se enviará en el RTT 6

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 ($100 \cdot \text{MSS}$ bytes). Responde a las siguientes preguntas:



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

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

- 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)*

a)

Origen	Nº secuencia	Flags	Nº reconocimiento	Datos
p _A	1000	SYN	-----	-----
p _B	5000	SYN,ACK	1001	-----
p _A	1001	ACK	5001	-----

b)

Origen	Nº secuencia	Flags	Nº 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	1601	-----
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)

Ejercicio 12

Origen (C o S)	Nº secuencia	Flags	Nº 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	---