

Ejercicio P3 (Kurose)

- ❖ UDP y TCP utilizan complemento a 1s para sus sumas de comprobación (checksums). Supongamos que se tienen los siguientes bytes de 8 bits: 01010011, 01010100, 01110100.
 - (a) ¿Cuál es el complemento a 1s de la suma de estos bytes? (Tenga en cuenta que aunque UDP y TCP utilizan palabras de 16 bits en el cálculo de la suma de comprobación, para este problema se pide considerar sumas de 8 bits.) Razonar la respuesta.
 - (b) ¿cómo detecta los errores el receptor ?
 - (c) ¿Es posible que un error de 1 bit pase desapercibido? ¿Y un error de 2 bits?

Ejercicio P3 (Kurose)

- ❖ UDP y TCP utilizan complemento a 1s para sus sumas de comprobación (checksums). Supongamos que se tienen los siguientes bytes de 8 bits: 01010011, 01010100, 01110100.
(a) ¿Cuál es el complemento a 1s de la suma de estos bytes?

$$\begin{array}{r} 01010011 \\ + 01010100 \\ + 01110100 \\ \hline 00011011 \\ + 1 \\ \hline 00011100 \\ \text{Ca1 } \left(\begin{array}{l} 00011100 \\ \boxed{11100011} \end{array} \right. \text{Checksum} \end{array}$$

- (b) ¿cómo detecta los errores el receptor ?
- (c) ¿Es posible que un error de 1 bit pase desapercibido? ¿Y un error de 2 bits?

Ejercicio P3 (Kurose)

- ❖ UDP y TCP utilizan complemento a 1s para sus sumas de comprobación (checksums). Supongamos que se tienen los siguientes bytes de 8 bits: 01010011, 01010100, 01110100.
(b) ¿cómo detecta los errores el receptor ?

$$\begin{array}{r} 01010011 \\ + 01010100 \\ + 01110100 \\ + 100011011 \\ \hline 11111110 \\ + \quad \quad 1 \\ \hline 11111111 \end{array} \rightarrow \text{No hay error}$$

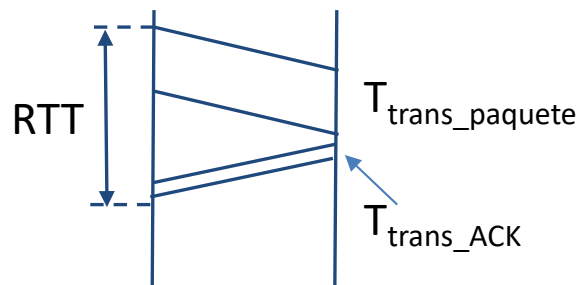
- (c) ¿Es posible que un error de 1 bit pase desapercibido? **No** ¿Y un error de 2 bits? **Sí**

Ejercicio

- ❖ ¿Cuánto tiempo se necesita para transmitir un mensaje de $1,5 \times 10^6$ bytes sobre un enlace con una velocidad de transmisión de 1 Mbps? Asumid que los paquetes de datos son de 1500B, el tamaño del ACK es de 50B, y el retardo de propagación de 10 ms. **NO** hay que considerar en los paquetes a transmitir las cabeceras de los protocolos empleados en los distintos niveles:
 - ❖ a) Si el control de flujo empleado es de parada y espera.
 - ❖ b) Si el control de flujo empleado es de ventana deslizante, con tamaño máximo de la ventana de transmisión de 7 paquetes.

- ❖ ¿Cuánto tiempo se necesita para transmitir un mensaje de $1,5 \times 10^6$ bytes sobre un enlace con una velocidad de transmisión de 1 Mbps? Asumid que los paquetes de datos son de 1500B, el tamaño del ACK es de 50B, y el retardo de propagación de 10 ms. **NO** hay que considerar en los paquetes a transmitir las cabeceras de los protocolos empleados en los distintos niveles:

- ❖ a) Si el control de flujo empleado es de parada y espera.



$$V_{\text{trans}} = 1 \times 10^6 \text{ bps}$$

$$T_{\text{prop}} = 10 \text{ ms} = 10 \times 10^{-3} \text{ s}$$

$$N^{\circ} \text{ paquetes a transmitir} = 1,5 \times 10^6 / 1500 = 1000 \text{ paquetes}$$

$$\text{Long_ACK} = 50 \text{ Bytes} = 50 \times 8 = 400 \text{ bits}$$

$$\text{Long_Paquete} = 1500 \times 8 = 12000 \text{ bits}$$

$$T_{\text{trans_paquete}} = 12000 / 1 \times 10^6 = 12 \times 10^{-3} \text{ s}$$

$$T_{\text{Trans_ACK}} = 400 / 1 \times 10^6 = 0,4 \times 10^{-3} \text{ s}$$

$$\text{RTT} = T_{\text{trans_paquete}} + T_{\text{trans_ACK}} + 2 \times T_{\text{prop}} =$$

$$= 12 \times 10^{-3} + 0,4 \times 10^{-3} + 2 \times 10 \times 10^{-3} = 32,4 \times 10^{-3} = 32,4 \text{ ms}$$

$$1000 \text{ paquetes} \rightarrow 1000 \times \text{RTT} = \mathbf{32,4 \text{ s}} \text{ (STOP \& WAIT)}$$

- ❖ b) Si el control de flujo empleado es de ventana deslizante, con tamaño máximo de la ventana de transmisión de 7 paquetes.

$$V_{\text{trans}} = 1 \times 10^6 \text{ bps}$$

$$T_{\text{prop}} = 10 \text{ ms} = 10 \times 10^{-3} \text{ s}$$

$$N^{\circ} \text{ paquetes a transmitir} = 1.5 \times 10^6 / 1500 = 1000$$

$$\text{Long_ACK} = 50 \text{ Bytes} = 50 \times 8 = 400 \text{ bits}$$

$$\text{Long_Paquete} = 1500 \times 8 = 12000 \text{ bits}$$

$$T_{\text{trans_paquete}} = 12000 / 1 \times 10^6 = 12 \times 10^{-3} \text{ s}$$

$$T_{\text{Trans_ACK}} = 400 / 1 \times 10^6 = 0.4 \times 10^{-3} \text{ s}$$

Comprobamos si se consigue envío continuo:

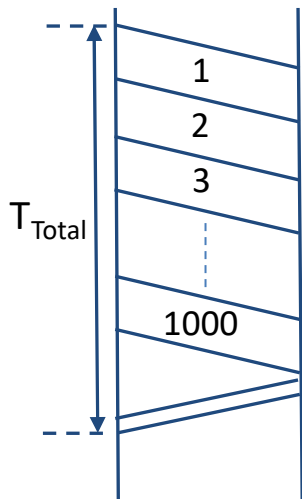
$$\text{¿ } 7 \times T_{\text{trans_paquete}} = 7 \times 12 \times 10^{-3} > \text{RTT} = 32.4 \times 10^{-3} \text{ s ?} \rightarrow \text{Se cumple}$$

Luego el tiempo para enviar N paquetes es de: $(N-1) \times T_{\text{trans_paquete}} + \text{RTT}$

$$\text{Como se envían 1000 paquetes: } (1000-1) \times T_{\text{trans_paquete}} + \text{RTT} = \mathbf{12.0204 \text{ s}}$$

O bien, observando el gráfico:

$$T_{\text{total}} = 1000 \times T_{\text{trans_paquete}} + 2 \times T_{\text{prop}} + T_{\text{trans_ACK}} = \mathbf{12.0204 \text{ s}}$$



Ejercicio P24 (Kurose)

Se requiere la transferencia de un archivo muy grande de L bytes del host A al host B. Suponga un MSS de 536 bytes.

- ❖ a. ¿Cuál es el valor máximo de L para que los números de secuencia TCP no se agoten? Recuerde que el campo de número de secuencia TCP tiene 4 bytes.
- ❖ b. Para el valor de L obtenido en (a), encuentre el tiempo que se necesita para transmitir el archivo. Suponga que cada segmento añade un total de 66 bytes para las cabeceras de transporte, red, y enlace de datos antes de enviar el paquete resultante a través de un enlace de 155 Mbps. No hay que tener en cuenta el control de flujo ni el de congestión, de modo que A puede enviar los datos seguidos y de manera continuada.

Ejercicio 2 (boletín)

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

ACK = 1

PSH = 0

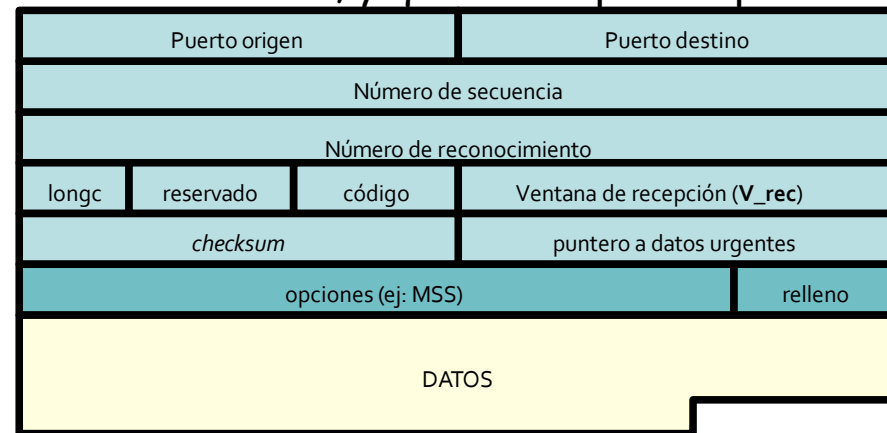
Tamaño de la ventana (en decimal) 86535

Checksum (en hexadecimal) 9FB0

Opciones (en bytes hexadecimales) 02- 04 - 05 – B4 (MSS = 1460)

04 – 02 (activar reconocimiento selectivo)

Relleno: (sin relleno)



RST = 0

SYN = 1

FIN = 0

Ejercicio 2 (boletín)

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

1º segmento. Bit ACK debería estar a 0

ACK = 1

PSH = 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)

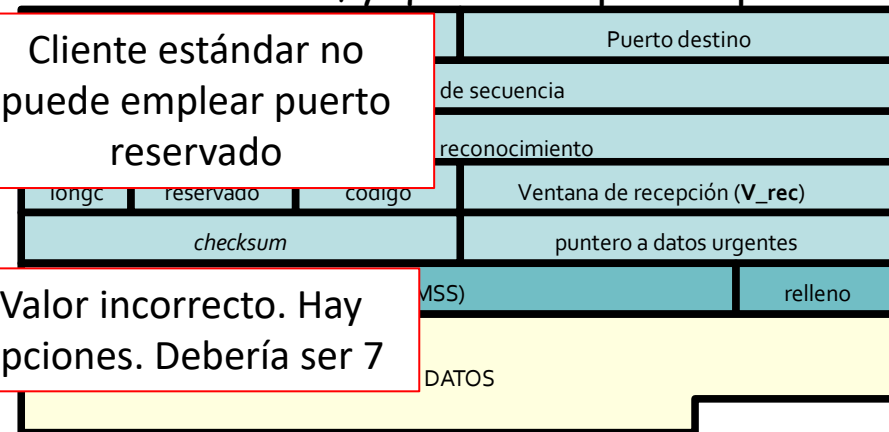
Relleno: (sin relleno)

Cliente estándar no puede emplear puerto reservado

Valor incorrecto. Hay opciones. Debería ser 7

Campo de 16 bits. Valor máximo $2^{16} - 1 = 65535$

Relleno sí es necesario: 00-00



RST = 0

SYN = 1

FIN = 0

Ejercicio 5 (boletín)

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 ha(n) 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?

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 ha(n) perdido? ¿Y cuáles puede considerar que se han recibido correctamente?

El primer ACK recibido con valor de 15873 indica que los 31 primeros segmentos han llegado correctamente. Los 3 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 – 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) 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?

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 de los mismos. 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 (podrían haber llegado fuera de orden y que todavía no se hubieran recibido los ACKs duplicados).

Ejercicio 3 (boletín)

- ❖ 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?

Ejercicio 7 (boletín)

- ❖ 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				

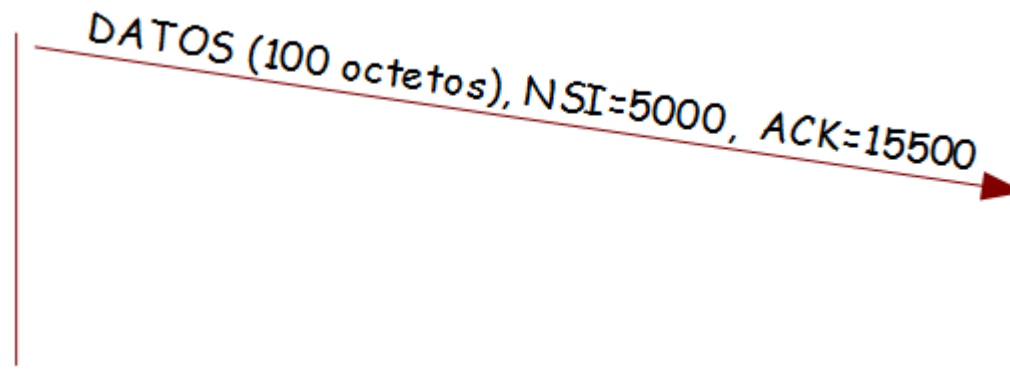
Ejercicio 7 (boletín)

- ❖ 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 <u>secuencia</u>	<u>Flags</u>	Nº <u>ACK</u>	Datos
A	51	SYN		---
B	0	RST, ACK	52	---

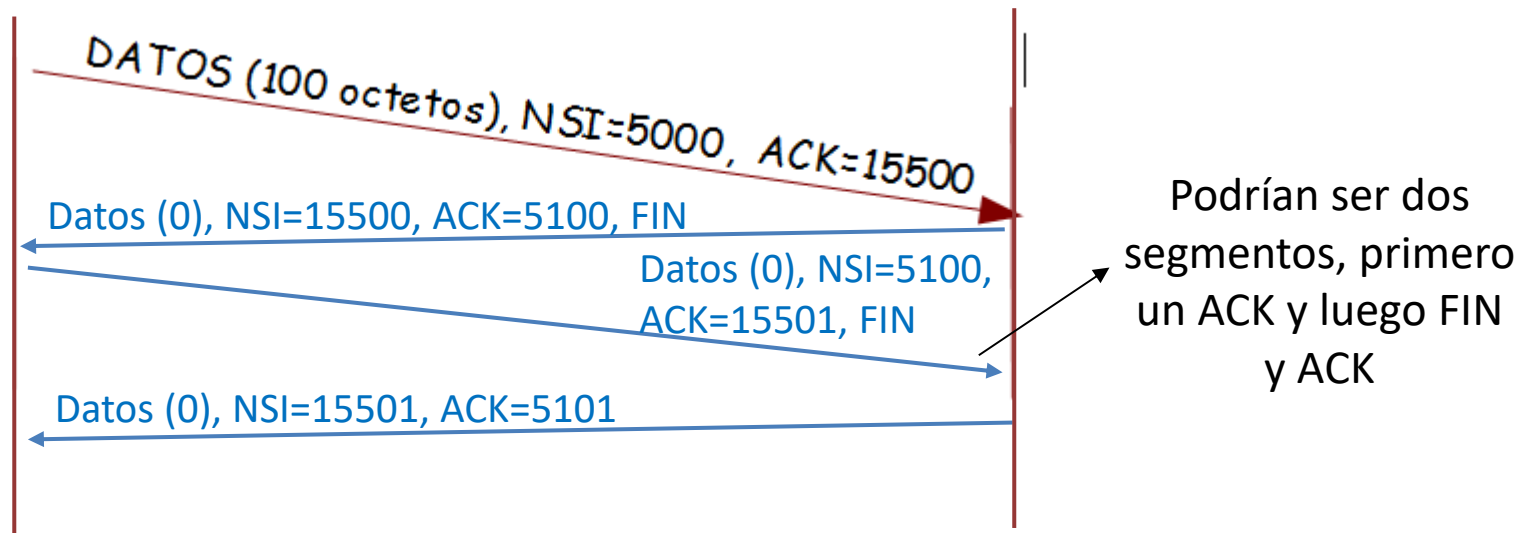
Ejercicio 8 (boletín)

- ❖ 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.



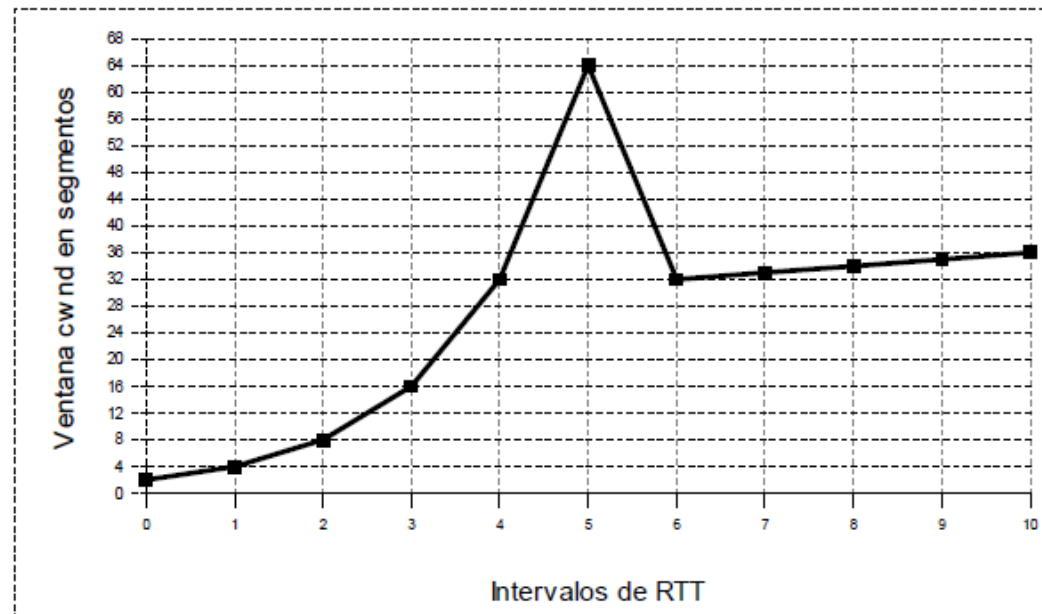
Ejercicio 8 (boletín)

- ❖ 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.



Ejercicio 10 (boletín)

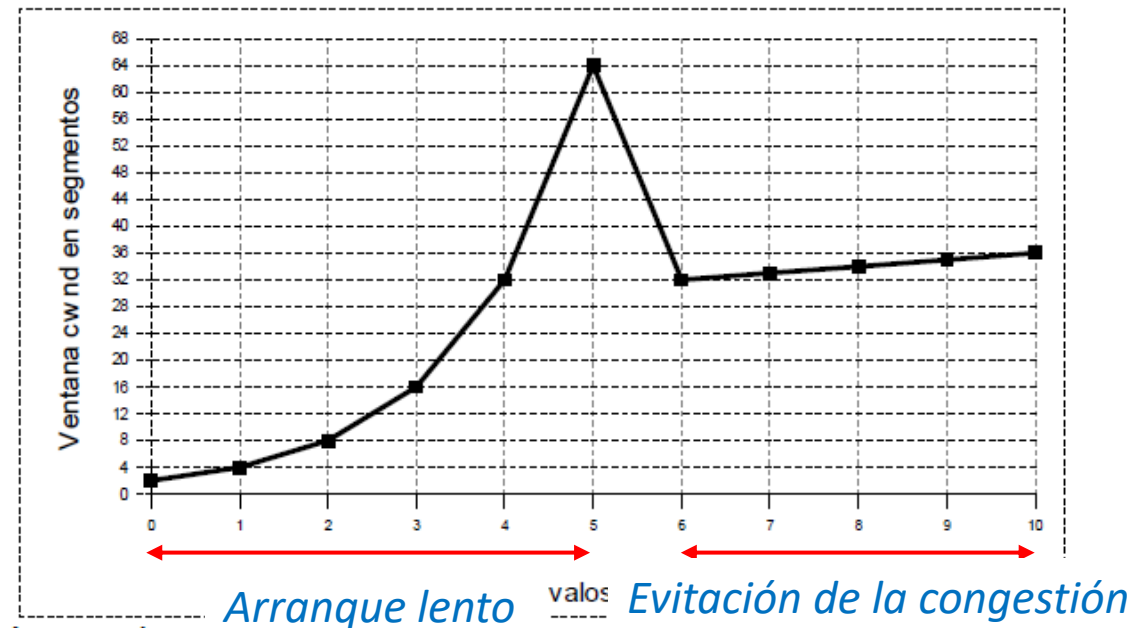
La gráfica siguiente representa la evolución de una ventana de congestión TCP en función del tiempo. La ventana de permisos del receptor (WIN) es siempre igual a 100 segmentos ($100 \cdot \text{MSS}$ bytes). Responde a las siguientes preguntas:



- ❖ a) Indica qué mecanismos están actuando y durante qué intervalos.
- ❖ b) ¿Qué ocurre durante el RTT 5? ¿Cómo se detecta?

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:

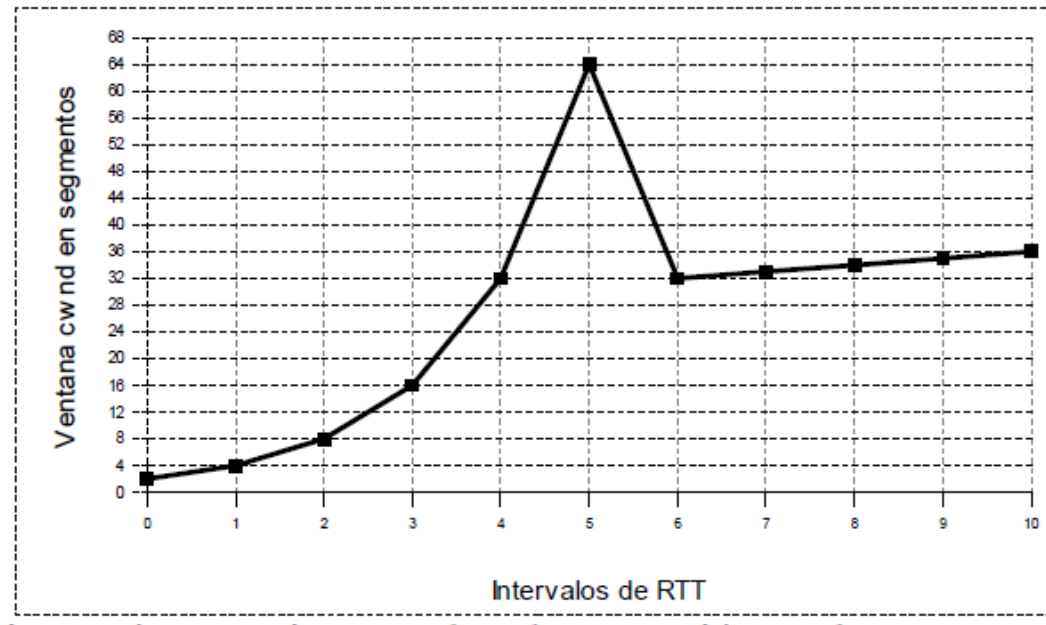
- ❖ a) Indica qué mecanismos están actuando y durante qué intervalos.



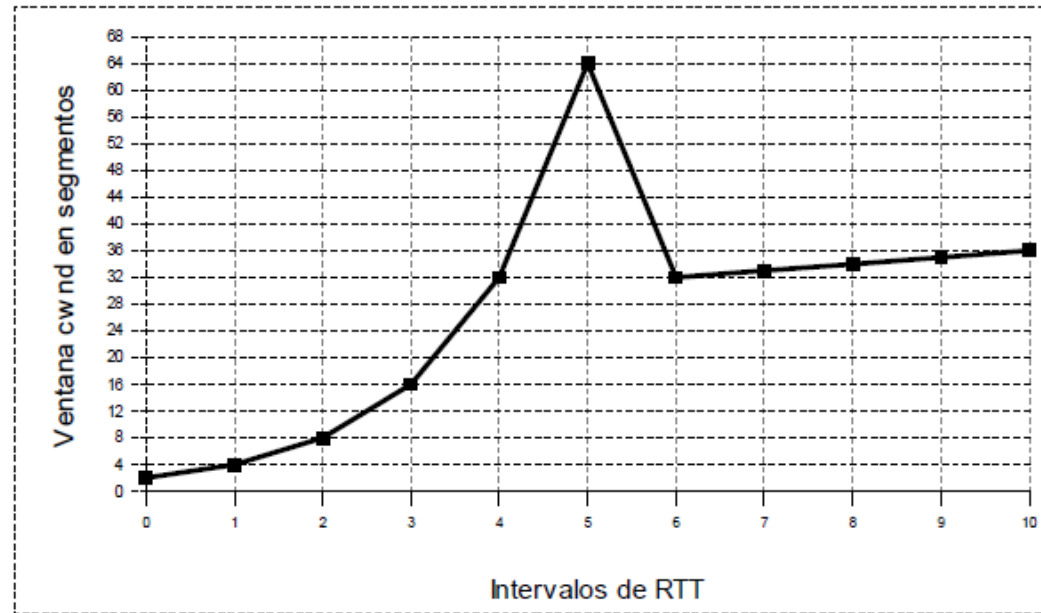
- ❖ b) ¿Qué ocurre durante el RTT 5? ¿Cómo se detecta?

Pérdida de un segmento (3 ACKs duplicados)

Ejercicio 10 (boletín)



- ❖ 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?



- ❖ c) ¿Cuál es el valor del umbral durante el RTT 3? ¿Y durante el RTT 8?

En el RTT=3 el Umbral es el inicial igual a 100 MSS Bytes

En el RTT=8 el Umbral es de 32 MSS Bytes

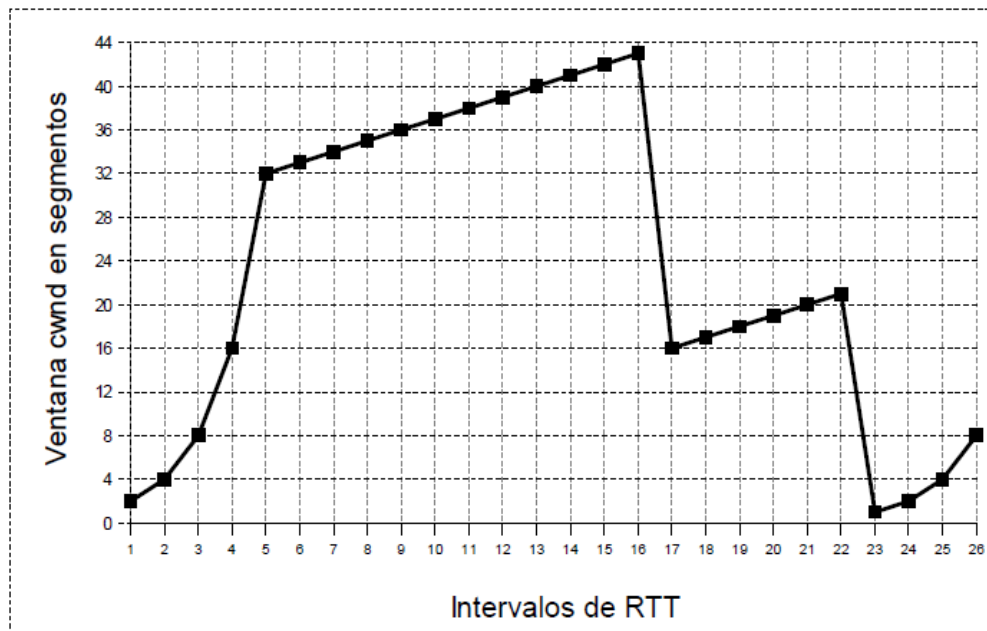
- ❖ d) Si durante el RTT 10 vence un temporizador, ¿cuáles serán los valores de la ventana de congestión y del umbral en el RTT 11?

Ventana de congestión = 1 MSS Bytes

Umbral = $36/2 = 18$ MSS Bytes

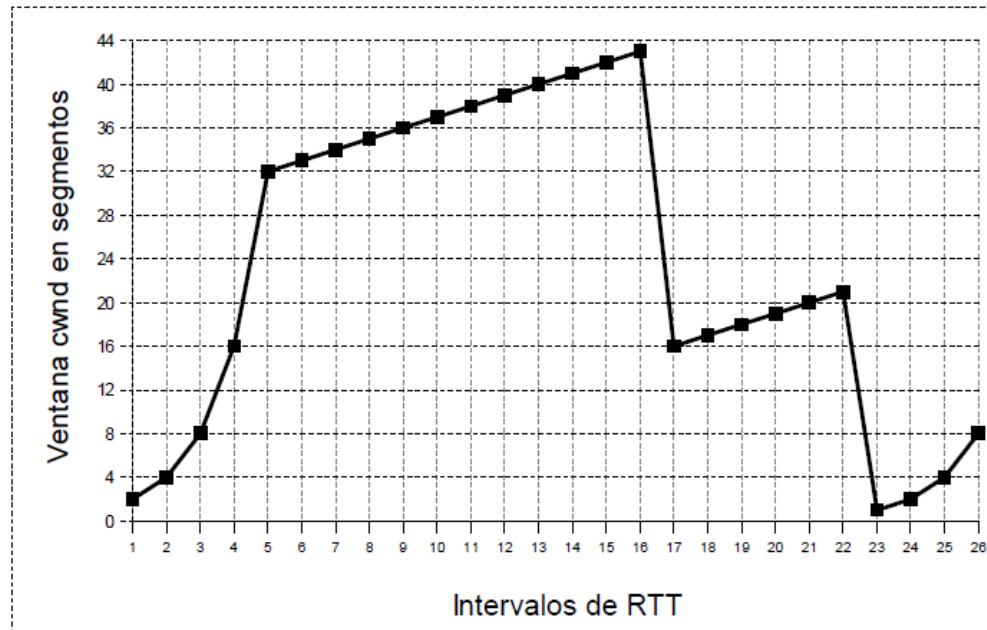
Ejercicio 9 (boletín)

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).



- ❖ a) Identifica los intervalos en los que actúa el mecanismo arranque lento (slow-start).
- ❖ b) Identifica los intervalos en los que actúa el mecanismo de evitación de la congestión (congestion avoidance).

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).



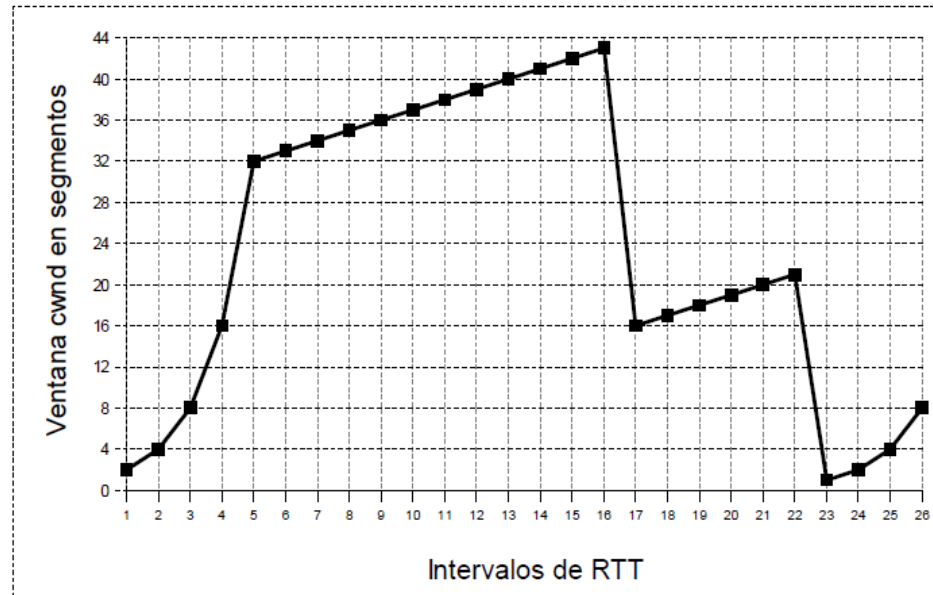
- ❖ a) Identifica los intervalos en los que actúa el mecanismo arranque lento (slow- start).

Del RTT 1 al 5 y desde el RTT 23 al 25

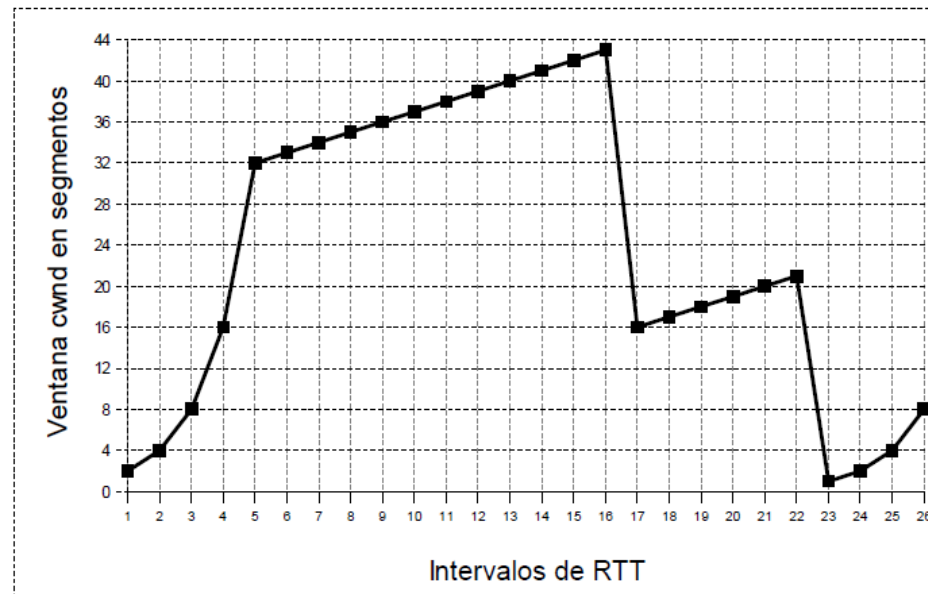
- ❖ b) Identifica los intervalos en los que actúa el mecanismo de evitación de la congestión (congestion avoidance).

Del RTT 5 hasta el 16 y desde el 17 al 22

Ejercicio 9 (boletín-cont.)



- ❖ c) A continuación del RTT 16, ¿cómo se detecta la pérdida del segmento? ¿Y a continuación del RTT 22?
- ❖ d) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 2?
- ❖ e) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 18?
- ❖ f) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 24?
- ❖ g) ¿Durante que RTT se envía el segmento 70?
- ❖ 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?



- ❖ 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 3 ACKs duplicados.

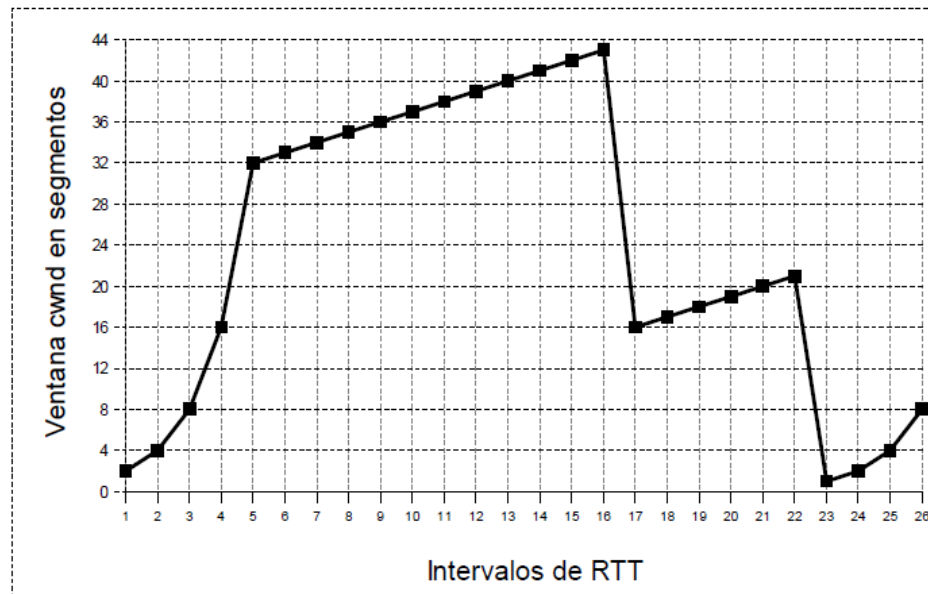
En el RTT=22 vence un temporizador

- ❖ d) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 2?

Umbral = 32 segmentos

- ❖ e) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 18?

Umbral = $\text{Ventana_transmisión} / 2 = 32 / 2 = 16$ segmentos



- ❖ f) ¿Cuál es el valor del umbral (slow-start threshold) en el RTT 24?

$$\text{Umbral} = \text{Ventana_transmisión} / 2 = 21 / 2 = 10.5 \text{ segmentos}$$

- ❖ 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

- ❖ 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?

$$\text{Ventana_congestión} = \text{Umbral} = \text{Ventana_transmisión} / 2 = 8 / 2 = 4 \text{ segmentos}$$

Ejercicio (no boletín)

Refleja en la siguiente tabla la evolución del umbral y de las ventanas de congestión y transmisión con respecto al tiempo (en RTT's), teniendo en cuenta los eventos que se indican (suceden durante el RTT, y se detectan al final del mismo). Todos los valores se expresan en segmentos. El valor de V_{rec} indicado en cada RTT es válido desde el principio del mismo. El emisor siempre tiene nuevos datos que transmitir. No se emplean ACKs retrasados.

(**) = recepción de 3 ACK's duplicados (*) = *TimeOut*

[illegible]

Ejercicio (solución)

Refleja en la siguiente tabla la evolución del umbral y de las ventanas de congestión y transmisión con respecto al tiempo (en RTT's), teniendo en cuenta los eventos que se indican (suceden durante el RTT, y se detectan al final del mismo). Todos los valores se expresan en segmentos. El valor de V_{rec} indicado en cada RTT es válido desde el principio del mismo. El emisor siempre tiene nuevos datos que transmitir. No se emplean ACKs retrasados.

(**) = recepción de 3 ACK's duplicados (*) = *TimeOut*

RTT	1	2	3	4	5	6	7**	8	9*	10	11	12
V_{rec}	64	64	64	64	64	64	64	30	30	40	30	20
Umbral	64	64	64	64	64	64	64	32	32	15	15	15
V_{cong}	2	4	8	16	32	64	65	32	33	1	2	4
$V_{trans_m\acute{a}x}$	2	4	8	16	32	64	64	30	30	1	2	4

Ejercicio 12 (boletín)

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)

Origen	N° secuencia	Flags	N° ACK	Datos

Solución

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.

Origen	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	---

Ejercicio examen (Rec. Febrero 2018)

La única de diferencia entre dos computadores es que en uno de ellos el valor inicial de la ventana de congestión en el algoritmo de arranque lento está configurado a 8 MSS (computador A) y en el otro a 4 MSS (computador B). Indique, mostrando los cálculos realizados, qué máquina conseguirá enviar antes (en menor número de intervalos de RTT) los siguientes mensajes de distinto tamaño teniendo en cuenta que la ventana de transmisión está únicamente limitada por la de congestión, no por la de recepción (el receptor tiene suficiente espacio):

- a. Si el mensaje requiere 4 MSS:
- b. Si el mensaje requiere 6 MSS:
- c. Si el mensaje requiere 20 MSS:
- d. Si el mensaje requiere 60 MSS:
- e. Si el mensaje requiere 64 MSS:

La única de diferencia entre dos computadores es que en uno de ellos el valor inicial de la ventana de congestión en el algoritmo de arranque lento está configurado a 8 MSS (computador A) y en el otro a 4 MSS (computador B). Indique, mostrando los cálculos realizados, qué máquina conseguirá enviar antes (en menor número de intervalos de RTT) los siguientes mensajes de distinto tamaño teniendo en cuenta que la ventana de transmisión está únicamente limitada por la de congestión, no por la de recepción (el receptor tiene suficiente espacio):

- a. Si el mensaje requiere 4 MSS:
- b. Si el mensaje requiere 6 MSS:
- c. Si el mensaje requiere 20 MSS:
- d. Si el mensaje requiere 60 MSS:
- e. Si el mensaje requiere 64 MSS:

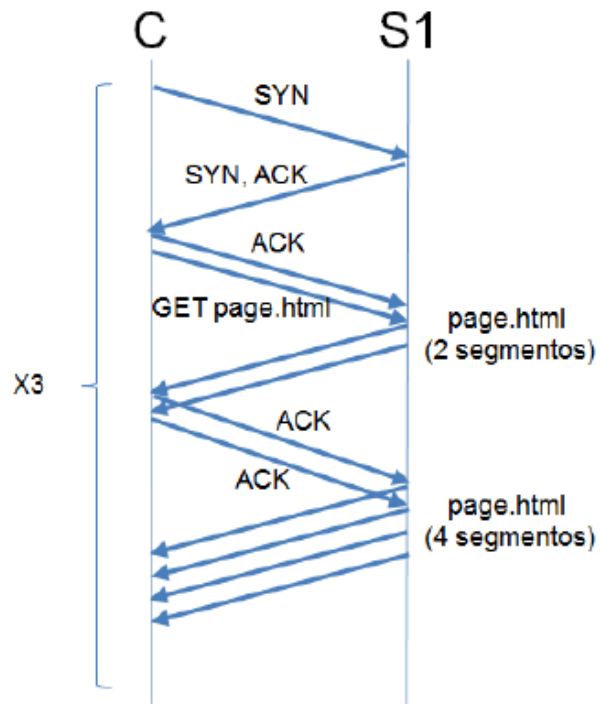
Hay que calcular en qué RTT termina de realizar cada computador el envío de todos los segmentos que se indican.

- a. Los dos tardan 1 RTT
- b. El computador A tarda menos (1 RTT) mientras que B necesita 2 RTT
- c. El computador A realiza 8 envíos en el primer RTT y puede realizar hasta 16 en el siguiente. Por lo tanto terminará los envíos en el RTT número 2. B envía 4 en el primero, 8 en el segundo y los restantes 6 en el RTT número 3.
- d. Computador A: $8 + 16 + 32 = 56$. Terminará el envío de los 4 restantes en el RTT nº 4
Computador B: $4 + 8 + 16 + 32 = 60$. Necesitará el mismo nº de RTT (4)
- e. Computador A: según el apartado anterior, necesitará 4 RTT mientras que B necesitará uno más (5 RTT)

Ejercicio examen (Rec. Febrero 2018)

Explica cómo afecta el uso del **arranque lento** TCP a las conexiones no persistentes HTTP. Para apoyar tu explicación muestra gráficamente la diferencia en el tiempo necesario para descargarse del mismo servidor un documento html con dos objetos adicionales. Cada respuesta ocupa 6 segmentos TCP. Cada una de las peticiones ocupa 1 segmento. Compara dos situaciones: con conexiones no persistentes y con conexiones persistentes sin *pipelining*.

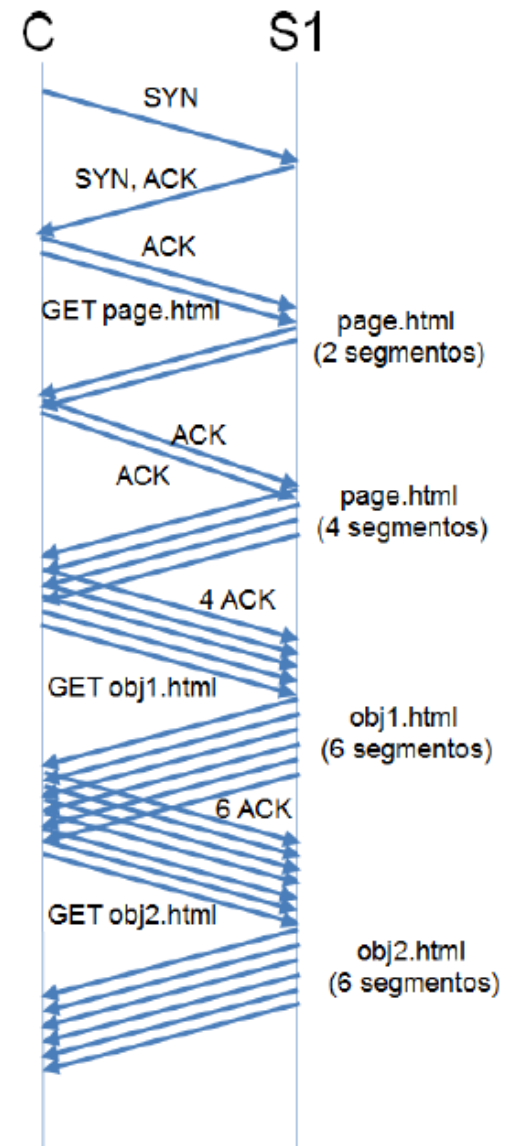
Nota: supóngase que NO se utilizan reconocimientos retrasados, y que la ventana de congestión nunca supera al umbral.



a) Total RTT : 9

conexiones no persistentes

b) Total RTT : 5



conexiones persistentes sin pipelining