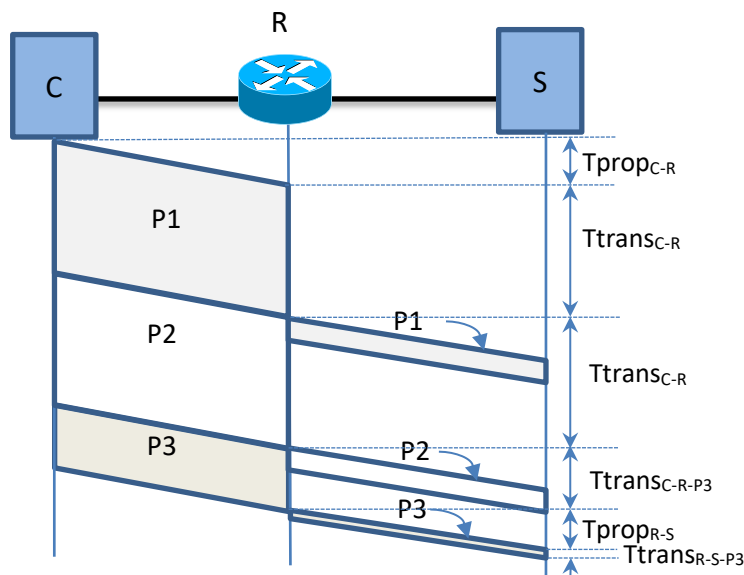
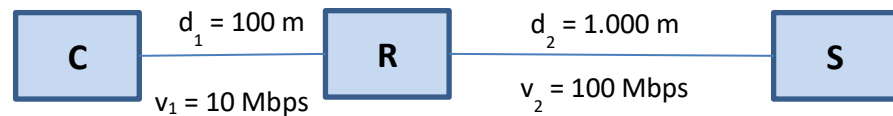




Apellidos y Nombre: _____

1. **(1,5 puntos)** Los computadores C y S están conectados a través del *router* R de conmutación de paquetes. La velocidad de transmisión en el enlace 1 es de 10 Mbps y en el enlace 2 de 100 Mbps. La longitud del enlace 1 es de 100 m y la del enlace 2 de 1.000 m. La velocidad de propagación en ambos enlaces es de 2×10^8 m/s. El tamaño máximo de paquete en ambas redes es de 500 bytes. C tiene que enviar un mensaje de 1.300 bytes. Se asume que no hay más paquetes en la red y que el tiempo de procesamiento en el *router* es despreciable. Si C empieza a enviar el primer paquete en $t = 0$:
 - i. Representa en un esquema gráfico el tránsito de los paquetes desde C hasta S.
 - ii. Calcula el tiempo total transcurrido hasta que S recibe todos los paquetes del mensaje.



Los dos primeros paquetes serán de tamaño máximo (500 bytes). Sin embargo, el último paquete llevará solo 300 bytes y, por tanto, su tiempo de transmisión será menor.

$$T_{propC-R} = \frac{100}{2 \times 10^8} = 0,5 \mu s$$

$$T_{transC-R} = \frac{500 \times 8}{10 \times 10^6} = 400 \mu s$$

$$T_{transC-R-P3} = \frac{300 \times 8}{10 \times 10^6} = 240 \mu s$$

$$T_{propR-S} = \frac{1000}{2 \times 10^8} = 5 \mu s$$

$$T_{transR-S-P3} = \frac{300 \times 8}{100 \times 10^6} = 24 \mu s$$

$$T_{total} = T_{propC-R} + 2 \times T_{transC-R} + T_{transC-R-P3} + T_{Router} + T_{propR-S} + T_{transR-S-P3} = (0,5 + 2 \times 400 + 240 + 0 + 5 + 24) = 1069,5 \mu s$$

2. **(1 punto)** A continuación, se muestra un diálogo HTTP entre un cliente y un servidor. Los caracteres <cr> <lf> representan los caracteres retorno de carro y salto de línea, respectivamente.

1. GET /cs453/index.html HTTP/1.1<cr><lf>
2. Host: gaia.cs.umass.edu<cr><lf>
3. User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gecko/20040804 Netscape/7.2 (ax) <cr><lf>
4. Accept:ext/xml, application/xml, application/xhtml+xml, text/html;q=0.9, text/plain;q=0.8,image/png,*/*;q=0.5<cr><lf>
5. Accept-Language: en-us,en;q=0.5<cr><lf>
6. Accept-Encoding: zip,deflate<cr><lf>
7. Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>
8. Keep-Alive: 300<cr><lf>
9. Connection: keep-alive<cr><lf><cr><lf>
10. <cr><lf>
11. <html><body><h1>404 Not Found</h1></body></html><cr><lf>

Contesta las siguientes preguntas, indicando (cuando proceda) en qué línea de las anteriores se apoya tu respuesta:

- i. ¿Cuál es la URL del documento solicitado por el navegador?
- ii. ¿Qué versión de HTTP está ejecutando el navegador?
- iii. ¿Qué tipo de conexión solicita el navegador, no-persistente o persistente? ¿El servidor está obligado a utilizar el tipo de conexión HTTP solicitada por el cliente?
- iv. ¿Qué objetivo tiene la línea 10?
- v. Indica porqué la respuesta del servidor no se ajusta al estándar.

i. [http:// gaia.cs.umass.edu/cs453/index.html](http://gaia.cs.umass.edu/cs453/index.html)

El campo Host indica el nombre del servidor (línea 2) y /cs453/index.html (línea 1) indica la ruta en el servidor y el nombre del objeto.

ii. HTTP 1.1. Se indica en la primera línea.

iii. Persistente (línea 9, Connection: keep-alive). El servidor no está obligado a seguir la solicitud del cliente, puede cerrar la conexión.

iv. Es la línea que indica la separación entre las cabeceras y el cuerpo del mensaje.

v. Falta la línea de estado inicial (HTTP/1.1 404 Not Found), que debería ir antes de la línea 10.

NOTA: se han marcado en amarillo en el enunciado el texto que permite responder a alguno de los apartados.

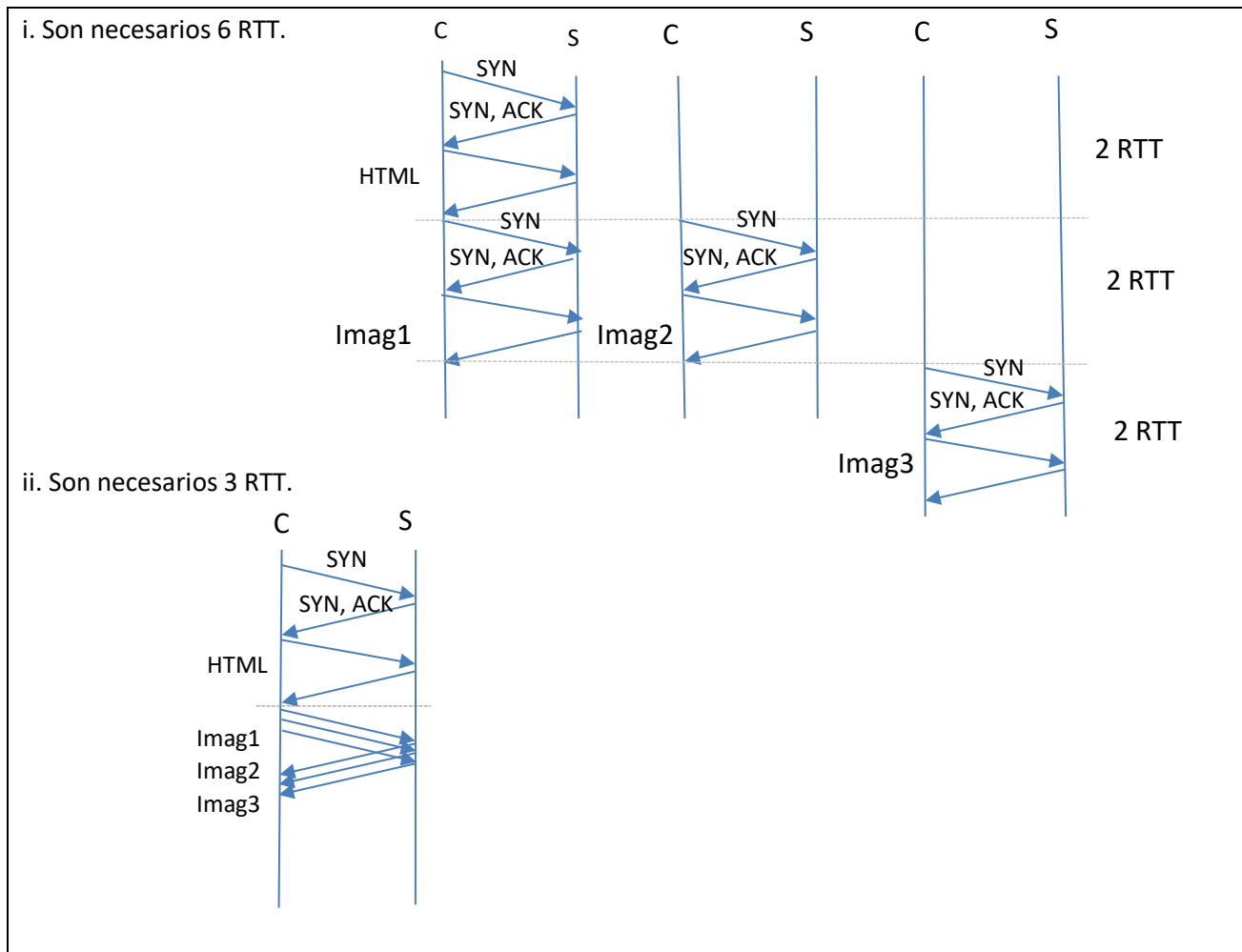
Recuperación 1^{er} parcial Redes de Computadores 12/02/2018

Apellidos y Nombre: _____

3. **(1 punto)** Utilizando HTTP, se desea acceder a una página web que está formada por un fichero HTML y 3 imágenes embebidas ubicadas en el mismo servidor que el fichero HTML. Calcula el número de RTTs necesarios para todo el proceso en los casos siguientes:

- Empleando conexiones no persistentes con un máximo de dos conexiones concurrentes.
- Empleando conexiones persistentes, sin conexiones concurrentes y con *pipelining*.

Cada objeto transmitido cabe en un segmento, y los tiempos de transmisión se considerarán despreciables con respecto a los de propagación. Justifica brevemente la respuesta.



4. **(1 punto)** ¿Qué información contienen los registros MX de un servidor DNS? ¿Quién suele realizar consultas de tipo MX y con qué objetivo?

- El campo nombre del registro MX contendrá el nombre de un dominio de correo y el campo valor el nombre de dominio del servidor SMTP que atiende ese dominio de correo.
- Las consultas MX las suele realizar un servidor SMTP cuando ha de enviar un correo cuyo destinatario pertenece a un dominio de correo externo, y por tanto, necesita averiguar quién es el servidor SMTP que atiende ese dominio.

5. **(2 puntos)** La siguiente conexión TCP incluye 5 errores de concepto (segmentos erróneos puede haber alguno más para hacer más visible algún error). Suponemos que no se han producido pérdidas de segmentos y que todos los segmentos se han recibido en orden. Cada segmento de datos genera un reconocimiento. La secuencia aparece desde el punto de vista del servidor. Los mecanismos de control de la congestión actúan de la forma habitual (v_cong inicial = 2 MSS). Cuando detectes un error, simplemente indícalo **justificando** porqué el segmento o grupo de segmentos introduce ese error. Aunque el segmento sea erróneo se tendrá en cuenta en la línea siguiente como si fuera correcto. Es decir, si lleva datos afectarán a los reconocimientos, números de secuencia, etc. de las líneas posteriores.

	Origen (Cliente/Servidor)	Num. Secuencia	Flags	Num. ACK	Datos (byte inicial y final)
1	C	5.000	SYN	-	-
2	S	8.000	SYN, ACK	5.001	8.001-8.200
3	C	5.001	ACK	8.201	5.001-5.200
4	S	8.201	ACK	5.201	8.201-8.400
5	S	8.401	ACK	5.201	8.401-8.600
6	S	8.601	ACK	5.201	8.601-8.800
7	S	8.801	ACK	5.201	8.801-8.900
8	C	5.202	ACK	8.401	-
9	C	5.203	ACK	8.601	-
10	C	5.204	ACK	8.801	-
11	C	5.205	ACK	8.901	-
12	S	8.901	FIN, ACK	5.206	
13	C	5.206	ACK	8.901	-
14	S	8.901	ACK	5.207	8.902-9.000
15	C	5.207	FIN, ACK	9.001	-
15	S	9.001	ACK	5.207	

Nota: como parte de la solución se han marcado en colores los errores en el enunciado para que resulte más fácil verlos.

Segmento 2: un segmento con el bit SYN activado no puede llevar datos nunca.

Segmentos 4-7: el tamaño de la ventana de congestión no permite enviar 4 segmentos seguidos todavía. Se podrían enviar 2 segmentos de datos seguidos (segmentos 4-5) si se considera el valor inicial de la ventana de congestión (2). La otra posibilidad es considerar que se pueden enviar 3 segmentos (segmentos 4-6) si se considera que el reconocimiento incluido en el segmento 3 ha incrementado en 1 el valor inicial de la ventana de congestión. El segmento 3 reconoce unos datos que no debían haberse enviado, por eso se acepta como válida la posibilidad de ignorar el reconocimiento de esos datos. En cualquier caso, el envío de 4 segmentos de datos en ese punto vulnera el tamaño máximo de la ventana de congestión.

Recuperación 1^{er} parcial Redes de Computadores 12/02/2018

Apellidos y Nombre: _____

Segmentos 8, 10 y 11: Durante la fase de transferencia de datos, los segmentos que no llevan datos no incrementan el número de secuencia. Por tanto, el número de secuencia de los segmentos 8-11 debería ser el mismo. Además, el segmento enviado en 8 debería ser el 5.201.

Cierre (13, 15): los segmentos con el bit FIN activado consumen un número de secuencia. Por lo tanto, los reconocimientos de los segmentos 13 y 15 debería haberse incrementado en 1 y el nº de secuencia del segmento 14 debería ser 8.902.

Cierre (14): un extremo no puede enviar datos una vez enviado el segmento FIN. El segmento 14 sobra.

6. (1 punto) El siguiente fragmento muestra información sobre una captura realizada con wireshark:

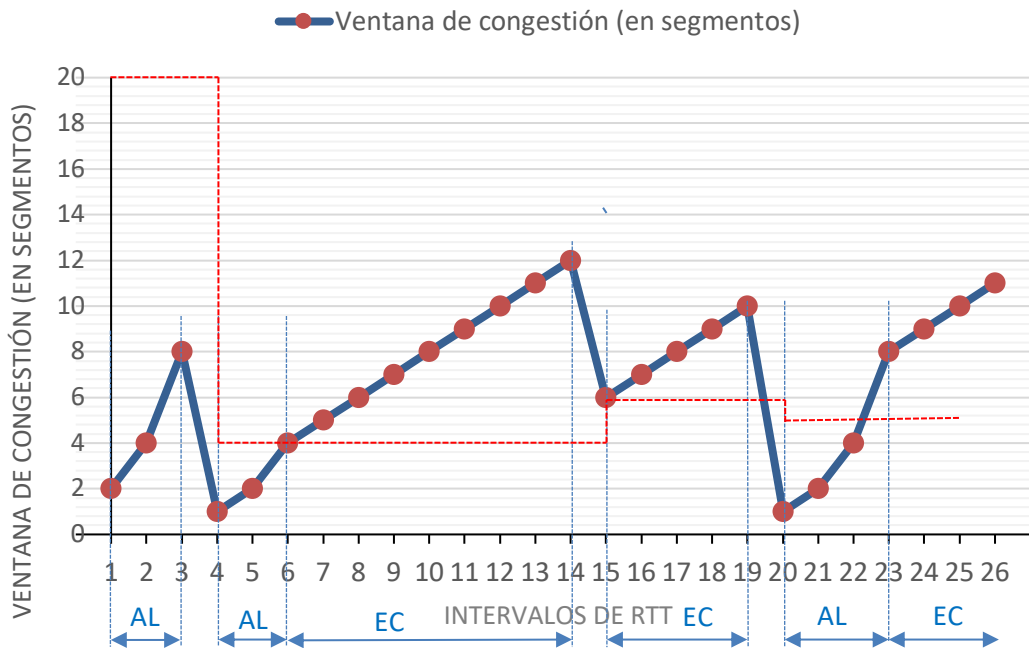
```
Internet Protocol Version 4, Src: 62.87.67.38, Dst: 192.168.1.7
Transmission Control Protocol, Src Port: 8080, Dst Port: 48306, Seq: 35, Ack: 43444, Len: 0
Source Port: 8080
Destination Port: 48306
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 35 (relative sequence number)
Acknowledgment number: 43444 (relative ack number)
1011 .... = Header Length: 44 bytes (11)
Flags: 0x010 (ACK)
Window size value: 928
[Calculated window size: 118784]
[Window size scaling factor: 128]
Checksum: 0xfada [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (24 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps, No-Operation (NOP), No-Operation (NOP), SACK
  TCP Option - No-Operation (NOP)
  TCP Option - No-Operation (NOP)
  TCP Option - Timestamps: TSval 740999910, TSecr 2472883681
  TCP Option - No-Operation (NOP)
  TCP Option - No-Operation (NOP)
  TCP Option - SACK 55028-56476
    Kind: SACK (5)
    Length: 10
    left edge = 55028 (relative)
    right edge = 56476 (relative)
    [TCP SACK Count: 1]
```

- Justifica hasta que byte de datos se ha recibido correctamente y en secuencia.
- Justifica si puede haberse perdido algún segmento y si el receptor ha recibido algún segmento fuera de orden.

- Se ha recibido correctamente y en secuencia hasta el nº de secuencia 43.443 inclusive. En el momento de enviar el segmento mostrado el receptor estaba esperando recibir un segmento con el nº de secuencia 43.444.
- El segmento incluye una opción de reconocimiento selectivo, lo que indica que se ha recibido al menos un segmento fuera de orden. En concreto, se han recibido sin error de transmisión, pero fuera de secuencia, los bytes de datos 55.028-56.475. Por tanto, el segmento (o segmentos) que llevaban los datos 43.444 – 55.027 no han llegado aún al receptor y pueden haberse perdido.

7. (1,5 puntos) La figura muestra la evolución de la ventana de congestión TCP al principio de cada RTT. Se supone que TCP envía una ráfaga de paquetes igual al tamaño de su ventana de congestión al comienzo de cada RTT. El resultado del envío de la ráfaga de paquetes es una de las siguientes posibilidades (i) todos los paquetes están reconocidos al final del RTT, (ii) se produce el timeout del primer paquete del RTT, o (iii) se reciben tres ACKs duplicados del primer paquete del RTT. Se pide:
- Indica la secuencia de eventos (ACKs duplicados, *timeouts*) y los instantes en que se producen que generen la evolución de la ventana de congestión TCP mostrada. Justifica cómo deduces el tipo de evento que se ha producido (una vez es suficiente para cada tipo de evento).

- b. Indica los algoritmos de control de la congestión que actúan en cada intervalo.
- c. Indica en la gráfica los valores del umbral desde el RTT 1 hasta el RTT 25 (valor inicial del umbral = 20 segmentos).



Se ha supuesto que las modificaciones de las variables en todo el ejercicio actúan al final del RTT donde se produce el evento.

- a. Timeouts en RTT 3 y 19. Se detectan porque en el RTT siguiente la ventana de congestión pasa a valer 1.
Recepción de 3 ACKs duplicados en RTT 14. La ventana de congestión en el RTT 15 se reduce a la mitad del valor que tenía la ventana de transmisión al inicio del RTT 14 (12 segmentos).
- b. Respondido sobre el enunciado (en azul). AL = Avance Lento. EC = Evitación de la Congestión.
- c. Representado en rojo en el enunciado.

8. **(1 punto)** Indica cómo obtener al enviar un mensaje las propiedades de seguridad indicadas en la tabla cuando se utiliza criptografía de clave pública.

Confidencialidad	Cifrando el mensaje con la clave pública del receptor del mensaje antes de enviarlo.
Autenticación	Cifrando el mensaje con la clave privada del emisor del mensaje (firma digital) antes de enviarlo.