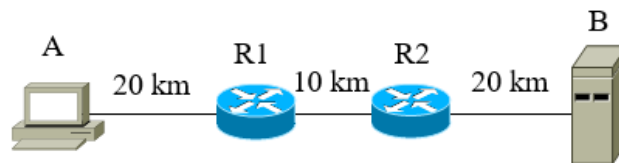




Apellidos y Nombre: _____

Preguntas:

1. El computador A desea transmitir al computador B un mensaje de 20.000 bytes mediante conmutación de paquete. La ruta entre ambos computadores atraviesa dos routers: R1 y R2. La longitud de cada enlace es la que se muestra en el dibujo (20 km y 10 Km). La velocidad de propagación es de 2×10^8 m/s. La velocidad de transmisión en todos los enlaces es 100 Mbits/s. La longitud máxima de los paquetes de datos es de 1.000 bytes y los paquetes de reconocimiento son de 50 bytes. Consideraremos despreciables los tiempos de procesamiento en hosts y routers. Suponemos que cada vez que se recibe un paquete en B, éste envía un paquete de reconocimiento de 50 bytes (ACK).
 - a. Calcule el tiempo que transcurre desde que sale el primer bit del mensaje hasta que se recibe el último ACK en A, en las siguientes condiciones:
 - i. Si el algoritmo empleado es de parada y espera. _____ (0,75 puntos)
 - ii. Si el algoritmo empleado es de ventana deslizante, con tamaño máximo de la ventana de transmisión de 5 paquetes. _____ (0,75 puntos)
 - b. Calcule el tamaño mínimo de la ventana de transmisión necesario para conseguir el envío continuo. _____ (0,50 puntos)



Número de paquetes transmitidos = $20.000 / 1.000 = 20$ paquetes

$$T_{\text{trans_datos}} = 1000 \times 8 / (100 \times 10^6) = 80 \times 10^{-6} \text{ s} = 80 \mu\text{s}$$

$$T_{\text{trans_ack}} = 50 \times 8 / (100 \times 10^6) = 4 \mu\text{s}$$

$$T_{\text{prop_1}} = 20 \times 10^3 / 2 \times 10^8 = 1 \times 10^{-4} \text{ s} = 100 \mu\text{s}$$

$$T_{\text{prop_2}} = 10 \times 10^3 / 2 \times 10^8 = 1 \times 10^{-4} \text{ s} = 50 \mu\text{s}$$

El RTT es el tiempo necesario para transmitir el segmento de datos y recibir su reconocimiento asociado:

$$\text{RTT} = 3(T_{\text{trans_datos}} + T_{\text{trans_ack}}) + 4 T_{\text{prop_1}} + 2 T_{\text{prop_2}} = 3(80 + 4) + 400 + 100 = 240 + 12 + 500 = 752 \mu\text{s}$$

a.i) Parada y espera

Al emplear parada y espera el tiempo de transmisión total serán 20 RTTs

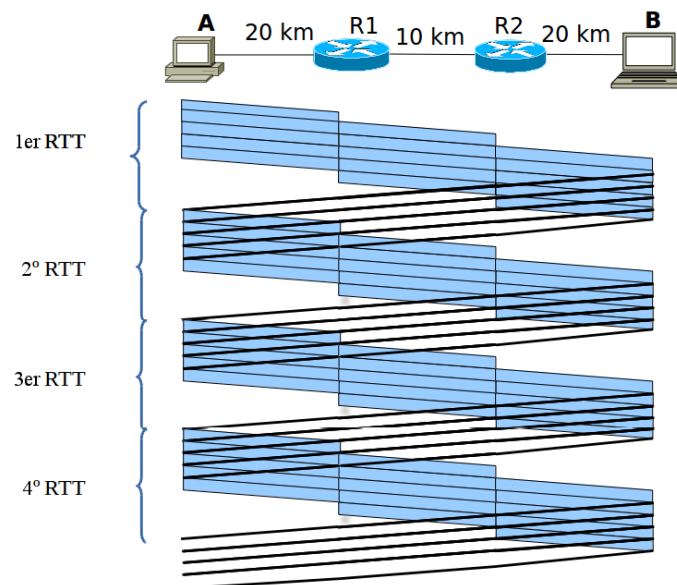
$$T_{\text{total}} = 20 \text{ RTT} = 20 \times 752 = 15040 \mu\text{s} = 15,040 \text{ ms}$$



a.ii) Ventana deslizante

Como A puede tener como máximo 5 paquetes pendientes de reconocimiento, puede enviar hasta 5 paquetes antes de recibir un ACK.

Como $5 \times T_{\text{trans_datos}} = 5 \times 80 = 400 \mu\text{s}$ es un tiempo inferior al RTT, A puede enviar 5 paquetes hasta alcanzar el tamaño máximo de la ventana, después debe detenerse hasta que reciba el primer ACK. Tras lo cual por cada ACK que reciba podrá enviar un nuevo paquete. Los ACKs siguientes llegarán con una diferencia de tiempo de $T_{\text{trans_datos}}$ cada uno. Por lo tanto, una vez recibido el primer ACK, A puede transmitir los restantes 5 paquetes sin detenerse. Cuando hayan transcurrido 4 RTTs habrá transmitido los 20 paquetes que tiene que transmitir en total y recibirá el ACK del décimo sexto paquete. El tiempo necesario para que lleguen los cuatro ACKs restantes es $4 \times T_{\text{trans_datos}}$. El 4º RTT coincide con el fin de la 4ª llave, que es cuando llega el 16º ACK). El tiempo necesario para que lleguen los cuatro ACKs restantes es $4 \times T_{\text{trans_datos}}$, como puede verse en la figura.



$$T_{\text{total-vdes}} = 4 \times \text{RTT} + 4 \times T_{\text{trans_datos}} = 4 \times 752 + 4 \times 80 = 3008 + 320 = 3328 \mu\text{s} = 3,328 \text{ ms}$$

b). Envío continuo

Para conseguir envío continuo se requiere que $n \times T_{\text{trans_datos}} > \text{RTT}$, donde n es el tamaño máximo de la ventana de transmisión. En nuestro caso, el valor mínimo de n será:

$$n \times 80 > 752; n > 752/80 = 9,4$$

Como n debe ser un valor entero, el valor mínimo para conseguir envío continuo será $n = 10$.



2. Considere la siguiente cadena de caracteres ASCII capturados por Wireshark durante el envío de un mensaje GET HTTP mediante un navegador (es decir, éste es contenido real de un mensaje GET de HTTP). Las combinaciones `<cr>` `<lf>` representan los caracteres retorno de carro y salto de línea.

```
GET /cs453/index.html HTTP/1.1<cr><lf>Host: gaia.cs.umass.edu<cr><lf>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>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>Accept-Language: en-us,en;q=0.5<cr><lf>Accept-Encoding: zip, deflate<cr><lf>Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>Keep-Alive: 300<cr><lf>Connection:keep-alive<cr><lf><cr><lf>
```

Conteste las siguientes preguntas, además subraye la parte del mensaje GET HTTP anterior donde se encuentra la respuesta. _____ (1 punto)

Las respuestas se han subrayado en amarillo en el mensaje HTTP.

¿Cuál es el URL del documento solicitado por el navegador?

http://gaia.cs.umass.edu/cs453/index.html

- a. ¿Qué versión de HTTP está ejecutando el navegador?

HTTP/1.1

- b. ¿Qué tipo de conexión solicita el navegador, no-persistente o persistente?

Conexión persistente

- c. ¿Se puede saber cuál es la dirección IP del host donde se está ejecutando el navegador? En caso afirmativo indique cómo averiguarla o cuál es.

En este nivel (aplicación) no se dispone de esta información. Se necesitaría información de los datagramas IP (que llevaba el segmento TCP que lleva la solicitud HTTP GET) para responder a esta pregunta.

3. Indique qué tipo de información contienen cada uno de los siguientes tipos de registro DNS:

NS, A, CNAME y MX _____ (1 punto)

Tipo NS: asocia un nombre_de dominio al nombre_del servidor DNS de ese dominio

Tipo A: asocia un nombre_de dominio de host (nombre-canónico) a una dirección IP

Tipo CNAME asocia un alias a un nombre-canónico



Tipo MX asocia un nombre_de un dominio de correo con el nombre_de su servidor de_correo

4. Implemente en Java un servidor TCP iterativo de echo que se ejecute de forma permanente. El servidor escuchará en el puerto 7, y responderá al cliente línea por línea, hasta que el cliente envíe una línea cuyo único contenido sea la palabra FIN. _____ (1,5 puntos)

```
import java.net.*; import java.util.*; import java.io.*;

public class ServidorTCP {
    public static void main(String args[]) {
        try {
            ServerSocket ss = new ServerSocket(7);
            while(true){
                Socket sc = ss.accept();
                Scanner entrada = new Scanner(sc.getInputStream());
                PrintWriter salida = new PrintWriter(sc.getOutputStream());
                String lee = entrada.nextLine();
                while (!lee.equals("FIN")) {
                    salida.println(lee);
                    salida.flush();
                    lee = entrada.nextLine();
                }
                sc.close();
            }
        } catch(IOException e) {
            System.out.println(e)
        }
    }
}
```

5. Un cliente y un servidor se quieren comunicar mediante el protocolo TCP. Sabemos que:
- El MSS que emplean los dos extremos es de 1000 bytes.
 - NSI(C) = 60.000, NSI(S) = 10.000 (NSI es el número de secuencia inicial).
 - WIN(C) = WIN(S) = 4000 bytes.
 - El tamaño inicial de la ventana de congestión es dos segmentos (2*MSS bytes).
 - Ambos extremos emplean reconocimientos retrasados.

Suponiendo que no se pierden datagramas ni se producen retransmisiones, indique los segmentos que se intercambiarían ambos TCP's en las siguientes situaciones. Para rellenar las tablas, hágase desde la óptica del cliente.

- a. El servidor realiza una apertura pasiva, y el cliente una apertura activa. _____ (0,5 puntos)

Origen (C o S)	Nº secuencia	Flags	Nº ACK	Datos (indicando byte inicial y final)
Cliente	60000	SYN		
Servidor	10000	SYN, ACK	60001	
Cliente	60001	ACK	10001	



- b. La aplicación cliente envía una petición de 1000 bytes al servidor. La respuesta del servidor es un mensaje de 9000 bytes _____ (1 punto)

Origen (C o S)	Nº secuencia	Flags	Nº ACK	Datos (indicando byte inicial y final)
Cliente	60001	ACK	10001	1 - 1000
Servidor	10001	ACK	61001	1-1000
Servidor	11001	ACK	61001	1001-2000
Cliente	61001	ACK	12001	
Servidor	12001	ACK	61001	2001-3000
Servidor	13001	ACK	61001	3001-4000
Cliente	61001	ACK	14001	
Servidor	14001	ACK	61001	4001-5000
Cliente	61001	ACK	15001	
Servidor	15001	ACK	61001	5001-6000
Servidor	16001	ACK	61001	6001-7000
Cliente	61001	ACK	17001	
Servidor	17001	ACK	61001	7001-8000
Servidor	18001	ACK	61001	8001-9000
Cliente	61001	ACK	19001	

Al emplear reconocimientos retrasados, el cliente cada vez que recibe 2 segmentos de datos suficientemente próximos (enviados en el mismo RTT) los reconocerá con el mismo reconocimiento. En la respuesta de la tabla se asume que el RTT es \gg de 500 ms. Esto implica que si el número de segmentos de datos enviados por el servidor en un RTT es impar, el cliente generará un ACK para reconocer el último segmento de datos del RTT en solitario. Debido a las restricciones de la ventana de congestión, el servidor enviará los 9 segmentos de datos en 3 RTTs. Enviando en el primer RTT, 2 segmentos (generan 1 ACK); en el segundo RTT, 3 segmentos (generan 2 ACK) y en el tercero, 4 segmentos (generan 2 ACK). Se recuerda que la tabla se ha completado desde la óptica del cliente tal como indica el enunciado.

- c. Después de la transmisión de datos la aplicación cliente le indica a su TCP el cierre de la conexión. Se supone que el servidor también está dispuesto a cerrar cuando le llegue el aviso de cierre del cliente _____ (0,5 puntos)

Origen (C o S)	Nº secuencia	Flags	Nº ACK	Datos (indicando byte inicial y final)
Cliente	61001	FIN, ACK	19001	
Servidor	19001	FIN,ACK	61002	
Cliente	61002	ACK	19002	



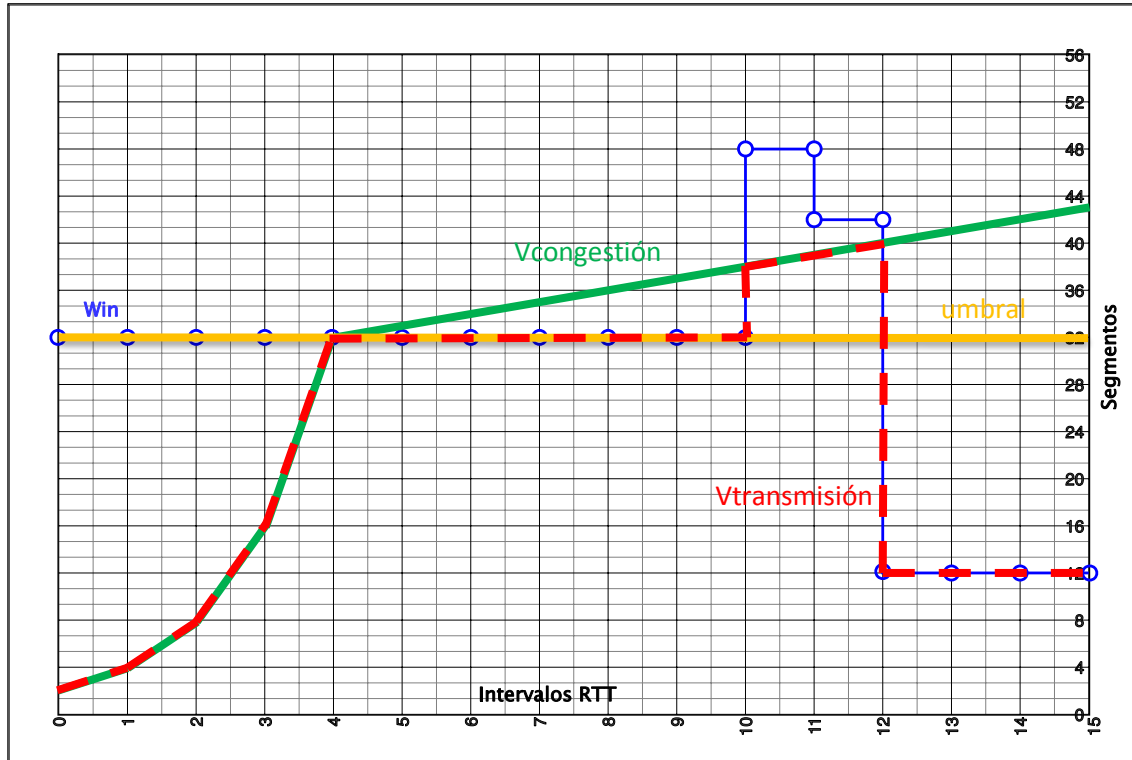
1^{er} parcial Redes de Computadores 09/01/2017



Origen (C o S)	Nº secuencia	Flags	Nº ACK	Datos (indicando byte inicial y final)
Cliente	61001	FIN, ACK	19001	
Servidor	19001	ACK	61002	
Servidor	19001	FIN, ACK	61002	
Cliente	61002	ACK	19002	



6. La siguiente gráfica muestra la evolución de la ventana de recepción (win) en una conexión TCP



- a. Dibuje sobre la gráfica anterior los valores que toma el “umbral” del algoritmo de control de congestión teniendo en cuenta que no se produce ningún tipo de error en la comunicación. _____ (0,5 puntos)
- b. Dibuje sobre la gráfica anterior cuál sería la evolución de las ventanas de transmisión y de congestión _____ (0,5 puntos)
- c. Suponiendo que en el último RTT se detecta la pérdida de un paquete por la recepción de 3 ACKs duplicados, ¿cuáles serán los valores de la ventana de congestión y del umbral en el siguiente RTT? ¿qué mecanismo (*slow-start* o evitación de congestión) se utilizará en ese RTT (el siguiente)? _____ (0,5 puntos)

Umbral = 6

Ventana de congestión = 6

El siguiente mecanismo será el de evitación de congestión



7. ¿Cuáles son las diferencias entre la confidencialidad de un mensaje y la integridad de un mensaje?
¿Puede haber confidencialidad sin integridad? ¿Puede haber integridad sin confidencialidad?
Justifique su respuesta. _____ (1 punto)

Confidencialidad: el contenido del mensaje debe ser inteligible únicamente para el emisor y el receptor

Integridad: cualquier alteración (accidental o intencionadamente) del mensaje durante la transmisión debe ser detectada.

Puede haber integridad sin confidencialidad, por ejemplo, cuando se emplean aplicaciones de firma digital, la firma es anexada al documento o mensaje sin alteración de la legibilidad del mismo, pero protegiéndolo de posibles alteraciones.