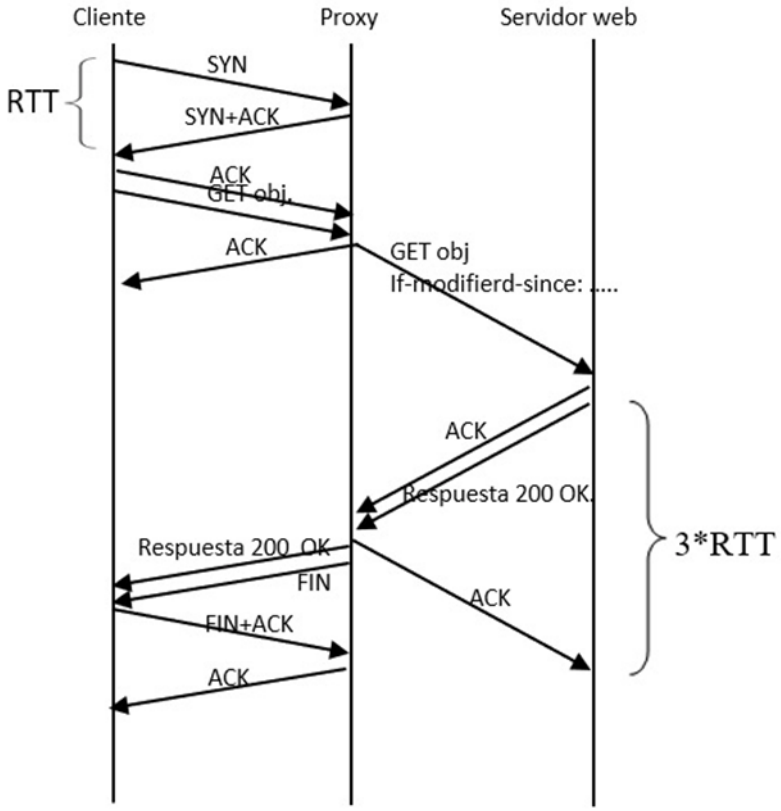
1er parcial Redes de Computadores 21/01/2021

APELLIDOS: _____ NOMBRE: _____

DNI: _____GRUPO: _____

1) (2,5 Puntos) En el escenario planteado en la siguiente figura, un navegador se conecta a un servidor web por medio de un proxy, que ya tiene una conexión establecida con el servidor web.



NOTA: Observa que, al estar el servidor más distante, el RTT de la conexión entre el proxy y el servidor es el triple que el RTT entre el cliente y el proxy.
Responde **justificadamente** las siguientes cuestiones:

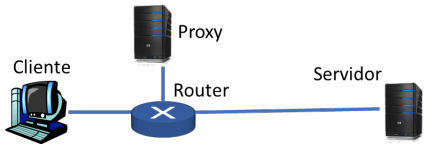
a) Cuando llega la petición del cliente, ¿el objeto solicitado se encuentra en la caché del proxy?

Sí, se encuentra en la caché. Se realiza un GET condicional al servidor para saber si el proxy mantiene una versión actualizada.

b) ¿Podemos saber si en algún momento el objeto ha sufrido cambios en el servidor?

Sí, la respuesta **200 OK** del servidor indica que ha cambiado desde la última versión que tenía el proxy.

c) Asumiendo que no hay más tráfico en la red y que todos los tiempos de transmisión se aproximan a cero. Sabiendo además que, tal y como se muestra en la siguiente figura, a nivel de red los tres dispositivos están conectados mediante un único router.



Si la velocidad de propagación en los tres enlaces es de 2,5 x 10^8 m/s, la longitud del enlace router-proxy es de 10 metros y la del cliente-router de 100 metros, ¿cuánto tiempo tardaría el navegador en obtener el objeto?

tiempo_respuesta=5*RTT

$tprop_{proxy\_router}$ = 10 / (2,5*10^8) =0,04 µs

$tprop_{cliente\_router}$ = 0,4 µs

RTT = (0,04+0,4)*2= 0,88 µs → tiempo_respuesta=5*0,88 = 4,4 µs

d) ¿Cuál es la longitud del enlace del router con el servidor web?

3*RTT= 2,64 µs = 2*$tprop_{proxy\_router}$ + 2*$tprop_{servidor\_router}$ =2*0,04 + 2* $tprop_{servidor\_router}$

$tprop_{servidor\_router}$=1,28 µs →

$d_{servidor\_router}$ = $tprop_{servidor\_router}$*vprop= 1,28*10^{-6}*2,5*10^8= 320 metros

2) (1 punto) Un cliente desea conectarse con el servidor www.redes.upv.es. Teniendo en cuenta que el servidor existe y se encuentra operativo, las siguientes líneas muestran el diálogo entre los mismos:

Línea	
0	% nc www.redes.upv.es 80
1	GET / HTTP/1.1
2	Connection: keep-alive
3	
4	HTTP/1.1 400 Bad Request
5	Content-Type: text/html; charset=us-ascii
6	Server: Microsoft-HTTPAPI/2.0
7	Date: Tue, 05 Jan 2021 15:28:35 GMT
8	Connection: close
9	Content-Length: 334
10	
11	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN""http://www.w3.org/TR/html4/strict.dtd">
12	<HTML><HEAD><TITLE>Bad Request</TITLE>
13	<META HTTP-EQUIV="Content-Type" Content="text/html; charset=us-ascii"></HEAD>
14	<BODY> Bad Request - Invalid Hostname
15	<hr> HTTP Error 400. The request hostname is invalid.
16	</BODY></HTML>
17	%

Contesta a las siguientes cuestiones **razonando las respuestas**:

a) ¿Se está empleando una conexión persistente?

La conexión es no persistente. Aunque el cliente indica en la solicitud que desea emplear una conexión persistente, el servidor, al contestar, le indica que será no persistente (Connection: close). El servidor es quien determina, en última instancia, el tipo de conexión.

b) En caso de que haya ocurrido algún error, ¿cuál ha sido el motivo? ¿Cómo se solucionaría para obtener el contenido correctamente?

Efectivamente se ha producido un error que impide la descarga solicitada. El error es debido a no cumplir con la especificación de HTTP 1.1 en el que se define la necesidad de usar la cabecera Host en la petición. La solución sería, por tanto

GET / HTTP/1.1
Host: www.redes.upv.es
Connection: keep-alive

c) ¿Para qué sirven las líneas 3 y 10?

Las líneas delimitan el final de las cabeceras tanto en las peticiones como en las respuestas. En el caso de la línea 3 indica el final de las cabeceras y dado que no hay bloque de datos, el servidor ya puede procesar la petición. En el caso de la línea 10 indica el final de las cabeceras y separa las mismas del inicio de los datos que envía el servidor.

3) (1 punto) Completa la tabla siguiente suponiendo que un servidor DNS realiza una consulta iterativa al servidor TLD.com. El objetivo de la consulta es averiguar el servidor SMTP asociado al dominio de correo everis.com. Se supone que los servidores utilizan nombres asociados al servicio que ofrecen. Por ejemplo, el servidor DNS de everis sería dns.everis.com.
Si alguna casilla debe quedar en blanco **justifica el porqué**.

P/R	Tipo/s de registro/s	Nombre	Valor
Pregunta	MX	everis.com (dominio de correo)	(es lo que se pretende averiguar)
Respuesta/s	NS	everis.com	dns.everis.com
Respuesta/s (registro/s con posible información adicional)	A [también AAAA]	dns.everis.com	dir IP del servidor dns.everis.com

4) (1 punto) Indica las sentencias Java mínimas necesarias para ejecutar las acciones que se solicitan a continuación (no usar "import", no definir la clase, no definir el método main, no usar excepciones, etc.). Sí que es necesario definir los sockets que vayan a utilizarse.

a) Un programa cliente establece una conexión TCP con el servidor "serv1.uji.es", al puerto 1524, le envía el texto "Hola\r\n" y muestra por pantalla la línea de texto con la respuesta recibida del servidor.

```
Socket s = new Socket(serv1.uji.es, 1524);

PrintWriter esc = new PrintWriter(s.getOutputStream());

esc.printf("Hola\r\n");

esc.flush();

Scanner lee = new Scanner (s.getInputStream());

System.out.println(lee.nextLine());
```

b) Un programa cliente envía un paquete UDP con el contenido "Hola\r\n" a todos los computadores de su misma red, usando la dirección de difusión "255.255.255.255" y el puerto destino 1524. Ayuda: DatagramPacket (byte buf[ ], int longitud, InetAddress dirIP, int puerto) throws SocketException

```
DatagramSocket ds = new DatagramSocket();
InetAddress dir = InetAddress.getByName("255.255.255.255");
String mens = "Hola\r\n";
byte[] buffer = mens.getBytes();
DatagramPacket paq = new DatagramPacket(buffer, buffer.length, dir, 1524);
ds.send(paq);
```

5) (1,5 puntos) Un cliente (C) y un servidor (S) se comunican mediante el protocolo TCP. La aplicación cliente envía una petición de 524 bytes al servidor. El servidor responderá con un mensaje de 1550 bytes, tras el cual iniciará el cierre de la conexión. El MSS que emplean los dos extremos es de 512 bytes, NSI(C) = 1.000, NSI(S) = 3.000 (NSI es el número de secuencia inicial). Supondremos que el tamaño de la ventana de recepción tanto del cliente como del servidor se mantiene constante igual a 2000 bytes y que 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 la tabla siguiente:

Origen (C/S)	Núm. de secuencia	Flags	Núm. de ACK	Datos (bytes inicial y final)
C	1000	SYN	---	---
S	3000	SYN,ACK	1001	---
C	1001	ACK	3001	1-512 (1001-1512)
C	1513	ACK	3001	513-524 (1513-1524)
S	3001	ACK	1525	1-512 (3001-3512)
S	3513	ACK	1525	513-1024 (3513-4024)
C	1525	ACK	4025	---
S	4025	ACK	1525	1025-1536 (4025-4536)
S	4537	ACK	1525	1537-1550 (4537-4550)
C	1525	ACK	4551	---
S	4551	FIN,ACK	1525	---
C	1525	FIN,ACK	4552	---
S	4552	ACK	1526	---

6) (1,25) Tras establecer una conexión entre un proceso en el host A y otro en el host B, la tabla refleja la evolución de la ventana de recepción de B en cada RTT.
Suponiendo que A tiene infinitos segmentos para enviar, que en el RTT=6 se produce un TimeOut y en el RTT=10 se detectan 3 ACK's duplicados (eventos que se detectan al final del RTT, y por tanto afectan al siguiente RTT), completa la tabla siguiente. No se producen otros errores ni se utilizan reconocimientos retardados. Todos los elementos se miden en segmentos.

RTT	1	2	3	4	5	**6**	7	8	9	**10**	11	12	13	14
V_rec(B)	16	30	32	12	30	16	16	24	26	24	10	10	12	12
Umbral (A)	16	16	16	16	16	16	8	8	8	8	4	4	4	4
V_cong(A)	2	4	8	16	17	18	1	2	4	8	4	5	6	7
V_trans(A)	2	4	8	12	17	16	1	2	4	8	4	5	6	7

7) (0,75 puntos) ¿Por qué no se aplica la técnica de reconocimientos retrasados cuando se envían reconocimientos duplicados?

Para que el otro extremo detecte cuanto antes la pérdida del segmento y lo pueda retransmitir

8) (1 punto) ¿Cuál es el principal problema de los cifrados de clave secreta o simétrica para su uso en las aplicaciones de red? ¿Cómo se soluciona actualmente en las comunicaciones informáticas?

El principal problema con los sistemas de cifrado simétrico no está ligado a su seguridad, sino al intercambio/distribución de claves. Para solucionar este problema se emplea criptografía de clave pública o asimétrica para el intercambio de las claves que luego se emplearán para el cifrado simétrico.