

## Ejercicios resueltos de Redes de Computadores – ETSINF – curso 2012-13

### Temas 2

1. A continuación se muestra un diálogo HTTP realizado mediante el programa nc:

```
1 GET /nube.gif HTTP/1.1
2 Pragma: no-cache
3 Cache-control: no-cache
4 Accept:image/png, image/jpeg, video/x-mng, image/jp2, image/gif;q=0.5, */*;q=0.1
5 Accept-Encoding: x-gzip, x-deflate, gzip, deflate
6 Accept-Charset: utf-8, utf-8;q=0.5, */*;q=0.5
7 Accept-Language: es, en
8 Connection: Keep-Alive
9
10 <html><body><h1>404 Not Found</h1></body></html>
```

Se pide:

- Identifica qué líneas son transmitidas por el cliente y cuáles por el servidor.
- Encuentra qué partes del diálogo no respetan el estándar de aplicación correspondiente.
- Corrige las líneas anteriores, añadiendo en su caso las líneas necesarias.

NOTA: Las líneas se han numerado como referencia.

#### **Solución:**

a) Transmitidas por el cliente: 1-9

Transmitidas por el servidor: 10

b) La respuesta del servidor no incluye línea de estado, ni línea de separación entre la cabecera y el cuerpo. Al ser el protocolo empleado por el cliente el HTTP 1.1 la petición del cliente debería también incluir la cabecera Host: <nombre del servidor destino>

c) Petición del cliente:

Incluirá las líneas 1-9 mostradas y además, antes de la línea en blanco (9) se necesita una cabecera Host: [www.ejemplo.com](http://www.ejemplo.com), suponiendo que [www.ejemplo.com](http://www.ejemplo.com) es el nombre del servidor destino.

La respuesta del servidor debería ser:

HTTP/1.0 404 Not Found

(línea en blanco)

<html><body><h1>404 Not Found</h1></body></html>

2. ¿Cuántos RTT's son necesarios para obtener una página web, consistente en un documento HTML y cuatro imágenes incrustadas, dos de ellas situadas en otro servidor, en cada uno de los siguientes casos?

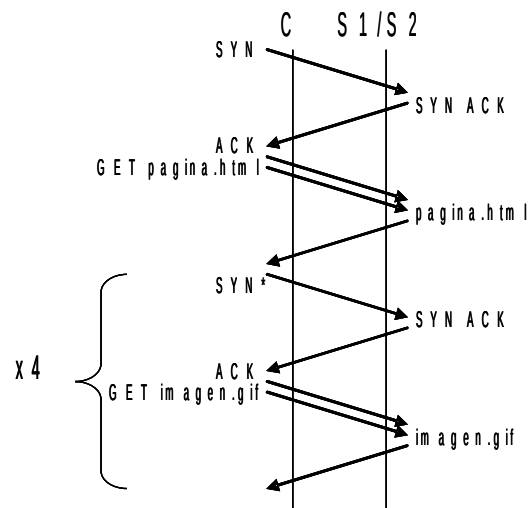
- Empleando conexiones no persistentes sin conexiones concurrentes.
- Empleando conexiones no persistentes con un máximo de dos conexiones concurrentes.
- Empleando conexiones persistentes, sin conexiones concurrentes y con pipelining

**Considera los tiempos de transmisión despreciables. Expresa de forma gráfica el intercambio de segmentos en cada uno de los casos**

En la solución presentada se ha interpretado que cuando se indica "sin conexiones concurrentes" es que no hay ningún tipo de concurrencia: ni en el mismo servidor ni entre servidores distintos. Sería conveniente que el alumno planteara la solución interpretando que cuando se indica "sin conexiones concurrentes" significara sobre el mismo servidor, es decir, permitiendo concurrencia siempre entre servidores distintos.

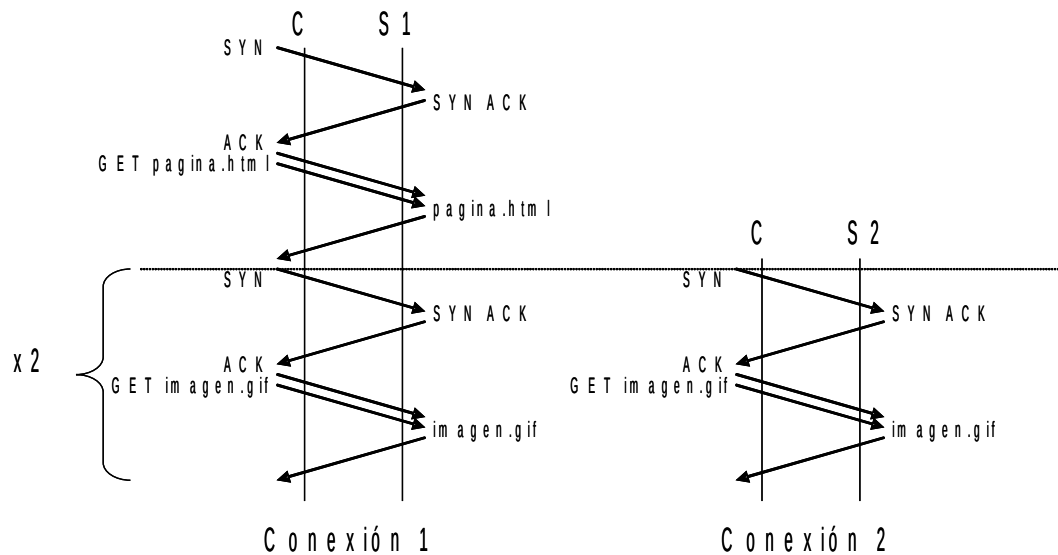
**Solución:**

a)  $RTTs_a = 2 \text{ (HTML)} + 4 \times 2 \text{ (4 imágenes, 2 RTT's por imagen)} = 10 \text{ RTTs}$

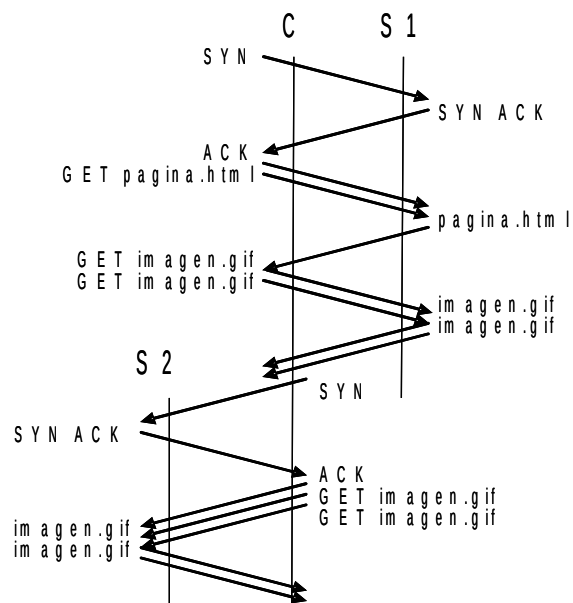


\* Dos de estas conexiones serán al servidor 2

b)  $RTTs_b = 2 \text{ (HTML)} + 2 \times 2 \text{ (Imágenes desde S1 ó S2, en paralelo)} = 6 \text{ RTTs}$



c)  $RTTs_c = 2 \text{ (HTML)} + 1 \text{ (Imágenes S1)} + 2 \text{ (Imágenes de S2)} = 5 \text{ RTTs}$



3. En el siguiente diálogo cliente/servidor:

```
220 out.telefonica.net ESMTP Service (7.2.056.6) ready
EHLO [192.168.1.33]
250-out.telefonica.net
250-DSN
250-8BITMIME
250-PIPELINING
250-HELP
250-X-CP-DELIVER-AFTER
250-DELIVERBY 300
250 SIZE 52428800
MAIL FROM:<pepu@telefonica.net> SIZE=376
250 MAIL FROM:<pepu@telefonica.net> OK
RCPT TO:<luis@telefonica.net>
250 RCPT TO:<luis@telefonica.net> OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
Message-ID: <47985FC8.30504@telefonica.net>
Date: Thu, 24 Jan 2008 10:52:08 +0100
From: <pepu@telefonica.net>
User-Agent: Mozilla Thunderbird 2.0.2 (Linux/20070317)
X-Accept-Language: en-us, en
MIME-Version: 1.0
To: luis@telefonica.net
Subject: prueba
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit

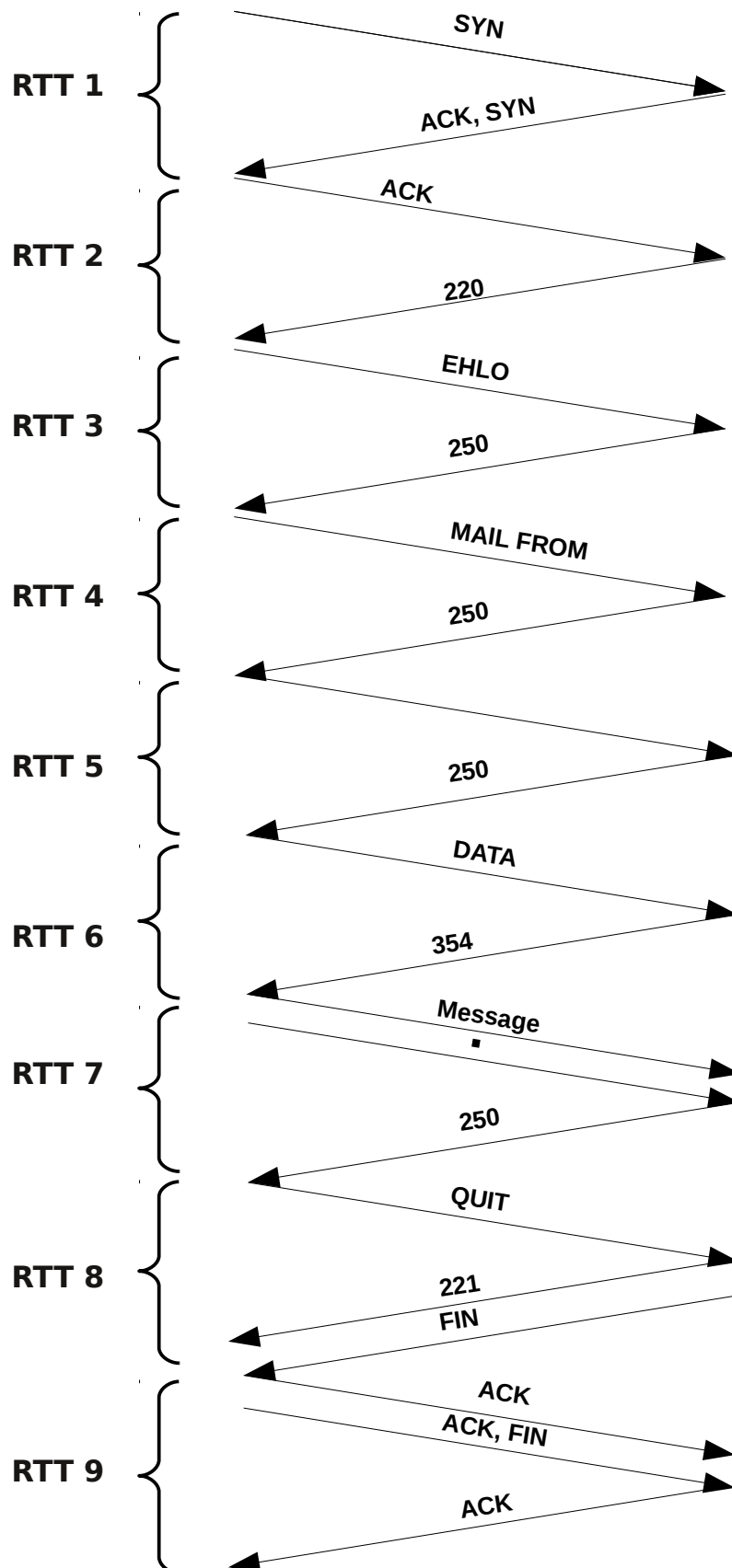
prueba 2
.
250 <4797C7CB0000E186> Mail accepted
QUIT
221 out.telefonica.net QUIT
```

- a) Sabiendo que la caché DNS del computador del usuario está vacía, indica la/s consulta/s DNS necesaria/s antes de iniciar la conexión TCP (tipo de consulta, a quién y para qué).
- b) Sabiendo que cada envío del cliente y cada respuesta del servidor ocupan un único segmento y todo el mensaje de correo otro, representa gráficamente el intercambio de segmentos y calcula los RTTs transcurridos desde el instante que se inicia la conexión hasta que se envía el último segmento de la misma, indicando a qué corresponde cada uno. Ambos extremos emplean ACKs retardados (si dos segmentos de datos se reciben en instantes muy próximos se reconocen con un único ACK los dos. Se estudiará en el tema 4).

### **Solución:**

- a) El cliente necesita establecer una conexión TCP con el servidor out.telefonica.net, para lo que necesita conocer la dirección IP de ese servidor. Por lo tanto, realiza una consulta de tipo **A** (*Address*) a su servidor DNS local. Si el servidor DNS dispone de la información, bien porque dadas las direcciones de correo del emisor y el destinatario parece que toda la interacción tiene lugar dentro del dominio telefonica.net y por lo tanto, el cliente y el servidor probablemente tengan registradas sus nombres en el mismo servidor DNS, o bien porque la tenga almacenada en caché, ya no serían necesarias más consultas.

b)



4. Seguidamente se muestra el diálogo entre un cliente y un servidor. ¿Qué protocolo están empleando para comunicarse?. ¿Cuál es la utilidad de este protocolo?. Indica sobre el diálogo qué líneas envía el servidor (S) y cuáles el cliente (C) (La solución a este apartado está incluida ya en el enunciado).

(S) 220 hamburger.edu  
(C) HELO crepes.fr  
(S) 250 Hello crepes.fr, pleased to meet you  
(C) MAIL FROM: <alice@crepes.fr>  
(S) 250 alice@crepes.fr... Sender ok  
(C) RCPT TO: <bob@hamburger.edu>  
(S) 250 bob@hamburger.edu ... Recipient ok  
(C) DATA  
(S) 354 Enter mail, end with "." on a line by itself  
(C) Te apetece ir al cine?  
(C) A ver dos policias rebeldes II?  
(C) .  
(S) 250 Message accepted for delivery  
(C) QUIT  
(S) 221 hamburger.edu closing connection

### **Solución:**

El protocolo empleado es el SMTP (*Simple Mail Transfer Protocol*).

Este protocolo permite el envío de mensajes de correo desde un cliente hasta el servidor de correo donde esté el buzón del destinatario. Como paso intermedio puede utilizarse algún otro servidor SMTP. Es un protocolo de envío de mensajes, pero no sirve para acceder a nuestro buzón y descargarnos los mensajes recibidos, para eso deben emplearse otros protocolos como POP3 o IMAP.

Las respuestas del servidor son todas las que empiezan con un código numérico de 3 cifras, las líneas restantes han sido enviadas por el cliente.

5. El usuario [emisor@upv.es](mailto:emisor@upv.es) tiene su aplicación de correo configurada para gestionar el correo a través de dos servidores: [imap.upv.es](mailto:imap.upv.es) y [smtp.upv.es](mailto:smtp.upv.es), situados en la red de la UPV. Tras encender su ordenador ([pc.upv.es](http://pc.upv.es)), el usuario ejecuta la aplicación de correo para enviar un correo al usuario [receptor@ya.com](mailto:receptor@ya.com) (perteneciente a otro dominio de correo).

Describe la secuencia de interacciones cliente/servidor necesarias hasta que el mensaje enviado al usuario receptor salga de la red de la UPV. Teniendo en cuenta que:

- Cada conexión TCP, desde el establecimiento hasta su cierre, se indicará como una única interacción (no hay que indicar el diálogo Cliente/Servidor realizado).

- Cada datagrama UDP se expresará como otra interacción.
- Se supone que el servidor DNS de la UPV (dns.upv.es) tiene en caché los servidores TLD necesarios (y sus IPs) para las consultas que tenga que realizar. Sin embargo, las cachés DNS de los otros ordenadores no disponen de la información DNS necesaria.
- En caso de que sea necesario emplear más servidores, se pueden nombrar según el protocolo que empleen y el dominio al que pertenezcan, como protocolo.dominio (por ejemplo, protocolo: dns, dominio: upv.es, nombre del servidor: dns.upv.es).

**Solución:**

Computador Origen	Computador Destino	Protocolo de aplicación	Objetivo (Si es UDP aclarar si es petición o respuesta, si es petición indicad el tipo de registro)
1. pc.upv.es	dns.upv.es	DNS	- Petición DNS para obtener la IP del servidor smtp.upv.es - Reg A
2. dns.upv.es	pc.upv.es	DNS	- Respuesta DNS con la IP del servidor smtp.upv.es - Reg A
3. pc.upv.es	smtp.upv.es	SMTP	- Envío del mensaje de correo
4. smtp.upv.es	dns.upv.es	DNS	- Petición DNS para obtener el nombre del servidor smtp asociado al dominio ya.com - Reg MX
5. dns.upv.es	TLD de dominio .com	DNS	Petición DNS para obtener el nombre del servidor smtp asociado al dominio ya.com - Reg MX
6. TLD de dominio .com	dns.upv.es	DNS	- Respuesta DNS con el nombre y la IP del servidor DNS asociado al dominio ya.com -Reg NS + reg A
7. dns.upv.es	dns.ya.com	DNS	- Petición DNS para obtener el nombre del servidor smtp asociado al dominio ya.com - Reg MX
8. dns.ya.com	dns.upv.es	DNS	- Respuesta DNS con el nombre y la IP del servidor smtp asociado al dominio ya.com - Reg MX + reg A
9. dns.upv.es	smtp.upv.es	DNS	- Respuesta DNS con el nombre y la IP del servidor smtp asociado al dominio ya.com - Reg MX + reg A
10. smtp.upv.es	smtp.ya.com	SMTP	Envío del mensaje de correo

6. Describe la secuencia de mensajes DNS (de petición y de respuesta) que se generarán cuando un usuario de ONO, cuyo DNS local es dns.ono.com, intenta descargarse la página principal del servidor web www.redes.upv.es. Dicha página contiene una imagen con URL <http://www.upv.es/images/logo.gif>.

Se supone que la caché de dns.ono.com no contiene información relativa al dominio upv.es pero contiene la lista de servidores TLD del dominio es.

El servidor de nombres de la UPV es dns.upv.es y resuelve el dominio upv.es y los subdominios derivados de upv.es.

Si necesitas añadir nuevos servidores de nombres utiliza un nombre adecuado (por ejemplo, el TLD del dominio .es debería llamarse TLD.es).

### **Solución:**

Origen	Destino	Tipo registro DNS	Recursiva/ Iterativa (Rec/It)	Objeto
usuario.ono.com	dns.ono.com	A	Recursiva	Pregunta la dirección IP del servidor www.redes.upv.es
dns.ono.com	TLD.es	A	Iterativa	Pregunta la dirección IP del servidor www.redes.upv.es
TLD.es	dns.ono.com	NS A	Iterativa	Devuelve servidores de nombres del dominio upv.es y sus IPs
dns.ono.com	dns.upv.es	A	Iterativa	Pregunta la dirección IP del servidor www.redes.upv.es
dns.upv.es	dns.ono.com	A	Iterativa	Responde la dirección IP del servidor www.redes.upv.es
dns.ono.com	usuario.ono.com	A	Recursiva	Responde la dirección IP del servidor www.redes.upv.es
usuario.ono.com	dns.ono.com	A	Recursiva	Pregunta la dirección IP del servidor www.upv.es
dns.ono.com	dns.upv.es	A	Iterativa	Pregunta la dirección IP del servidor www.upv.es
dns.upv.es	dns.ono.com	A	Iterativa	Responde la dirección IP del servidor www.upv.es
dns.ono.com	usuario.ono.com	A	Recursiva	Responde la dirección IP del servidor www.upv.es

7. a) Utilizando un cliente ftp en modo texto como el de prácticas se quiere: abrir una sesión con un servidor FTP, descargar tres ficheros almacenados en él y, a continuación, cerrar la sesión. ¿Cuántas conexiones TCP serán necesarias? ¿De qué tipo será cada una? Indica para qué se emplea cada tipo de conexión.

b) Si los tres ficheros se encuentran almacenados en el mismo directorio del servidor y disponibles a través de un servidor HTTP, ¿cuántas conexiones TCP se requerirían para descargarlos? Justifica la respuesta, teniendo en cuenta las distintas posibilidades que hay (a

nivel de HTTP).

**Solución:**

- a) Se necesitan cuatro conexiones: una de control para el diálogo con el servidor y tres de datos, una por cada fichero que hay que descargarse del servidor.
- b) Depende de que las conexiones sean persistentes o no. Con conexiones persistentes (comportamiento por defecto en HTTP/1.1) una única conexión sería suficiente. Si la conexión es no persistente (comportamiento por defecto en HTTP/1.0) y el servidor cierra la conexión después de enviar el objeto solicitado se necesitarían tres. Una para descargarse cada uno de los ficheros.

**Cuestiones**

- 1) a) ¿Para qué sirve MIME? b) ¿Cómo es posible enviar varios objetos MIME distintos dentro del mismo correo electrónico?

- a) Para poder enviar cualquier tipo de datos en binario (imágenes, ficheros de audio, ...) a través del protocolo SMTP que soporta sólo codificación ASCII de 7 bits. Para ello MIME define tres cabeceras adicionales a incluir en el mensaje: "MIME-Version:", "Content-Type" y "Content-Transfer-Encoding".
- b) Con el tipo MIME "Multipart". Luego cada uno de los objetos irá precedido por una cabecera "Content-type" indicando el tipo del objeto que sigue.

- 2) Suponiendo el comportamiento por defecto para las conexiones HTTP de la versión 1.1 a) ¿Cuántas conexiones serán necesarias para obtener, mediante HTTP 1.1 con *pipelining*, una página que está formada por un fichero HTML y tres imágenes incrustadas si todos los objetos se encuentran en el mismo servidor? b) ¿Y si una de dichas imágenes se encuentra en un servidor distinto? **Justifica la respuestas.**

- a) Una conexión. En HTTP 1.1 el tipo de conexión HTTP por defecto es persistente, al encontrarse todos los objetos en el mismo servidor se enviarán por la misma conexión.
- b) Dos conexiones. Una para la página y las imágenes que están en el mismo servidor y otra para la imagen que está en el otro servidor.

- 3) a) ¿Qué información contienen los registros MX de un servidor DNS? b) ¿Quién suele realizar consultas de tipo MX y con qué objetivo?

- a) MX – Nombre: dominio de correo; Valor: servidor de correo de ese dominio.
- b) Las realizan los servidores SMTP cuando tienen que enviar un mensaje de correo a un destinatario de otro dominio de correo distinto del suyo.

- 4) ¿Qué es un servidor proxy? ¿Qué ventajas proporciona?

Un servidor proxy es un elemento intermedio entre el cliente y el servidor que se emplea normalmente para hacer de memoria caché. Permite acortar los tiempos de respuesta cuando se reciben varias peticiones del mismo recurso en intervalos de tiempo próximos.



- 5) Explica de qué forma se puede reducir el tiempo de respuesta en los accesos a páginas web al usar un proxy web.

Al usar un proxy web, todas las peticiones de los clientes se canalizan por dicho proxy. Este proxy podría guardar una caché de las páginas consultadas por los clientes, de forma que cuando un cliente accede a una página web determinada, si antes otro cliente ha accedido a esa misma página, el proxy no necesita solicitar la página al servidor web correspondiente sino que la puede sacar de su propia caché, ahorrándose el tiempo de la consulta y de esta forma reduciendo el tiempo de respuesta que observa el cliente.

- 6) a) ¿Para qué sirve el protocolo POP3? b) ¿Qué otras alternativas existen para conseguir el mismo objetivo?

a) Para descargarse los mensajes de correo electrónico que un usuario tiene en su buzón.

b) Los mensajes también pueden descargarse mediante el protocolo IMAP o empleando acceso web (con el protocolo HTTP).

- 7) ¿Qué servicio ofrece el protocolo de correo POP3 que no realiza SMTP? ¿Cuáles son las fases de una sesión POP3? Transcribe de la forma más precisa posible (y en el orden correcto) las órdenes que un cliente POP3 envía al servidor a lo largo de una sesión en la que se realiza la transferencia de un mensaje. (0,75 puntos)

Nota: algunas órdenes de diversos protocolos de correo son: STAT, UIDL, HELO, MAIL FROM, USER, RCPT TO, LIST, DATA, DELE, RSET, QUIT, PASS, HELP, RETR, VRFY, TOP.

El protocolo de correo POP3 permite descargar los mensajes desde el servidor de correo local a la máquina del cliente. El protocolo SMTP sirve para enviar correo y llevarlo hasta el servidor de correo del destinatario. POP3 cubre el último paso.

Fases de una sesión POP3:

- Autenticación: identificación del usuario
- Manipulación de los mensajes: leerlos, listarlos, marcarlos para borrar, ...
- Actualización: se termina la sesión realizando (o no) los cambios indicados en la fase anterior (borrado de mensajes marcados).

USER pepe

PASS \*\*\*\*

LIST

RETR 1

DELE 1

QUIT

- 8) ¿Qué papel tiene el cliente en el modelo cliente-servidor? Explícalo con un ejemplo de una aplicación cliente-servidor.

El cliente tiene un papel activo. El cliente es el que:

1-Inicia la conexión al servidor.

2-Realiza peticiones al servidor (que el servidor se encargará de atender y contestar).

- 9) Explica por qué motivo “GET / HTTP/1.0 <CRLF><CRLF>” es una petición HTTP correcta y sin embargo “GET / HTTP/1.1 <CRLF><CRLF>” no lo es.

Falta la cabecera Host para especificar el host donde reside el objeto. Esta cabecera es obligatoria en el protocolo HTTP 1.1.

- 10) El protocolo SMTP no requiere la autenticación del cliente. ¿Debe permitirse que clientes ajenos a la organización que gestiona un servidor SMTP abran una sesión SMTP en el servidor (téngase en cuenta que estos clientes pueden estar incluidos en agentes de usuario u otros servidores)? ¿Por qué? ¿Qué acciones deberían permitirse a cada uno de estos tipos de clientes?

Sí debe permitirse que clientes ajenos a la organización abran una sesión SMTP en el servidor. Es necesario para que otros servidores SMTP hagan llegar sus mensajes a los destinatarios de la organización. Estos clientes “ajenos” sólo deberían tener la posibilidad de entregar mensajes dirigidos a destinatarios de la propia organización. No se les debería permitir que utilizarasen a nuestro servidor para reenviar mensajes a otros destinatarios externos.

- 11) ¿Qué tipo de información obtenemos cuando hacemos una consulta al servidor DNS con el nombre de dominio upv.es y el tipo MX?, ¿y si repetimos la consulta con el tipo NS?

Cuando hacemos una consulta del tipo MX al servidor DNS acerca del dominio upv.es obtenemos los nombres de dominio de los servidores de correo electrónico que atienden el dominio de correo upv.es.

Si hacemos la misma consulta, pero con el tipo NS, obtenemos nombres de dominio de los servidores de nombres para el dominio upv.es.

- 12) Indica cómo puede un servidor web saber qué cliente (Mozilla, Internet Explorer, etc.) está solicitándole una página web. ¿Lo puede saber siempre? Justifica la respuesta.

Mediante una cabecera en la petición HTTP. En concreto, la cabecera User-Agent:

No lo puede saber siempre ya que esta cabecera no es obligatoria.