

Examen de junio de los temas 4 y 5 (teoría)

18 de junio de 2021

Apellidos, Nombre: _____

Titulación:

- ☐ Informática
- ☐ Software
- ☐ Computadores

Grupo:

- ☐ A (mañana)
- ☐ B (inglés)
- ☐ C (tarde)

EXAMEN → 1 hora

10 preguntas cortas (0.2) — 2P (solo teoría y muchas se responden con una frase)
Problema — 1P

TEMA 5 → No entra P2P ni multimedia

TEMA 4 → Solo TCP y UDP

Duración: **20 minutos**

Ejercicio (1 punto, 0.2 puntos por pregunta). Responda brevemente a las siguientes preguntas (responda justo debajo de la pregunta en el sitio reservado para ello):

1. ¿Puede tener un mismo equipo un servidor TCP en el puerto 15000 y un servidor UDP en el puerto 15000? Justifique su respuesta.

- Sí. El rango de puertos UDP están desacoplados de los puertos TCP. Cuando se usa UDP no se usa TCP (son completamente independientes)

2. Si un nodo tiene la ventana de recepción [(1000 - 2999) 3000 - 4999] (con corchete se indica los números de secuencia admitidos y con paréntesis los recibidos, pero no servidos a la capa de aplicación) y recibe un segmento con número de secuencia 4000 con 1000 B de datos. ¿Qué hace con esos datos? ¿Cambia la ventana? ¿Qué ACK responderá?

- Los puede recibir porque están dentro de la ventana.
- La ventana no se desplaza. simplemente se marcan como recibidos
- El ACK tendrá n° de secuencia = 3000.

3. ¿Cómo sabe TCP si el segmento lleva datos urgentes y dónde están? ★ CABECERA TCP

- Porque el bit "URG" de la cabecera se pone a 1 si el segmento lleva datos urgentes y el puntero a datos urgentes indica el inicio de los datos urgentes del segmento.

4. Nombra dos funcionalidades que pueda requerir el servicio de nivel de aplicación que sea conveniente utilizar UDP como capa de transporte.

- UDP es un protocolo no orientado a la conexión y si el servicio que queremos utilizar necesita velocidad o no importa la pérdida de datos, se utiliza UDP.

- No orientado a la conexión
- se puede hacer broadcast a varios destinos
- Menos fiable porque no está orientado a la conexión, en TCP, no.
- Más rápido

- TCP
 - Fiabilidad (control de errores, reenvío, retransmisión, etc.)
 - Control del flujo (asegura que no se satura al receptor)
 - Control de la congestión (asegura que no satura el medio)



☆ DIFERENCIAS UDP y TCP A NIVEL DE FUNCIONALIDAD

5. Si un servidor UDP está atendiendo de forma simultánea a 10 clientes, ¿cuántas colas de recepción tiene?

- Independientemente del número de clientes, la cola es siempre una.

☆ 6. Dada la siguiente afirmación indique si es cierta o falsa justificando la respuesta: "Para resolver un dominio siempre hay que consultar a un servidor autorizado".

- Si yo hago una consulta, la resolución se guarda en una caché DNS que se va actualizando conforme a un valor temporal, por lo que es falso. Primero se revisa la caché DNS y si no estuviera en ella, se hace la consulta.

La caché DNS sirve para no tener que estar haciendo continuamente consultas

A cada entrada se le asigna un valor temporal que, mientras esté el valor vigente, no se hace la consulta y se accede directamente

7. Si usando Wireshark observamos la comunicación TCP entre 193.78.2.1:20 y 37.54.1.98:60001 perteneciente al protocolo FTP, ¿cuál extremo está actuando de servidor? Y, ¿en qué modo de FTP se está utilizando para la comunicación? Justifique su respuesta

se usa en el modo activo

☆ 2 comunicaciones en FTP → Puerto 20 (datos) - 193.78.2.1:20
- Desde el punto de vista del cliente

☆ Comando PASV y PORT

☆ FTP utiliza el puerto 21 para el control

modo pasivo

modo activo

→ La comunicación se hace entre el puerto indicado por PORT y el puerto 20.

El servidor elige los puertos para la comunicación de datos

8. Si un correo va a múltiples destinatarios y uno de ellos usando el comando DELETE de POP3 lo borra, ¿el resto de los destinatarios puede consultar ese correo? Justifique la respuesta.

- El cliente borra su mensaje de su buzón, ya que los buzones son independientes y cada cliente tiene su propio buzón. El servidor, a cada destinatario se manda una copia del correo que es independiente.

9. ¿Qué diferencia existe entre las conexiones permanentes en HTTP/1.1 y la conexión HTTP/2?

- La versión 1.1, por defecto, utiliza la cabecera "Keep alive" que permite que cuando el cliente manda un GET, no se cierra la conexión tras recibir la primera información recibida, y luego se cierra. La transmisión de archivos se hace de manera secuencial.
- HTTP/2, tenemos flujos que permiten intercalar, en distintos flujos dentro de la misma conexión, la descarga y la transmisión de archivos. Es decir, se puede multiplexar la descarga de distintos ficheros

10. ¿Qué diferencia hay entre los protocolos RTSP, RTCP y RTP? NO ENTRA

11. En UDP, ¿qué pasa si transmito y no hay servidor activo al otro lado? ¿y en TCP?

- En UDP puede que no me de cuenta, al ser un protocolo no orientado a la conexión, mientras que en TCP, al ser necesario una fase de conexión, si no hay servidor activo, lo sé al instante.

12. El cliente envía 10000B, pero el buffer de recepción del servidor es de 7000B, ¿qué pasa?

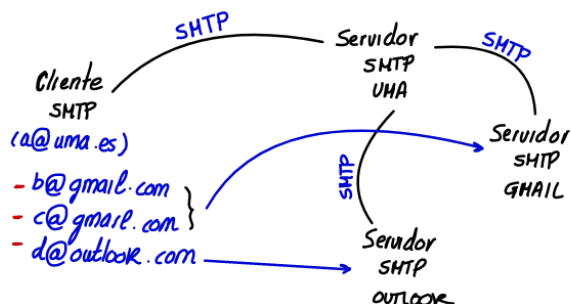
- Se aceptan los 7000B primeros y los 3000B restantes se descartan para siempre.

★ DIAGRAMA SMTP:

- ¿cuál es el protocolo

SMTP. El servidor se presenta al cliente, cuando el cliente va a enviar manda "HELO", ...

★ HTTP es un protocolo a nivel de aplicación SIN ESTADO, mientras SMTP es uno CON ESTADO, ya que, en función a la secuencia, pasa de un estado a otro. FTP es otro protocolo con estado.



PROBLEMA

★ Entre 25 y 30 segmentos (se puede simplificar)
 Importante: comunicación (MSS y tam. ventana)
 primero establecer la comunicación y luego rellenar las tramas (nº secuencia, ack y bits)

PUNTA
MÁS

★ El envío del correo con 3 destinatarios desencadena 3 sesiones

- envío al servidor de la UHA desde el cliente
- envío al servidor GMAIL desde el de la UHA
- envío al servidor OUTLOOK desde el de la UHA

★ PROBLEMA: NO CAE FTP!!

NO PLANIFICACIÓN HUFFMAN

- HTTP/2.0: ¿qué problema resuelve? → Multiplexación de archivos mediante flujos, aumenta la eficiencia (las cabeceras se comprimen) y otra ventaja son los mensajes PUSH-PROMISE (Mensajes que el servidor sabe que el cliente los va a pedir y empieza a enviarlos antes de que el cliente los pida).
- HTTP/1.1: Es secuencial

★ RELACIÓN PROBLEMAS (T5)

4. NO CAE FTP

★ Solución del síndrome de la ventana efímera (Silly window): Tras cerrar una ventana, el receptor no debe aceptar una ventana distinta de 0 hasta que no

CONTROL CONGESTIÓN (Ventana congestión): Posible pregunta

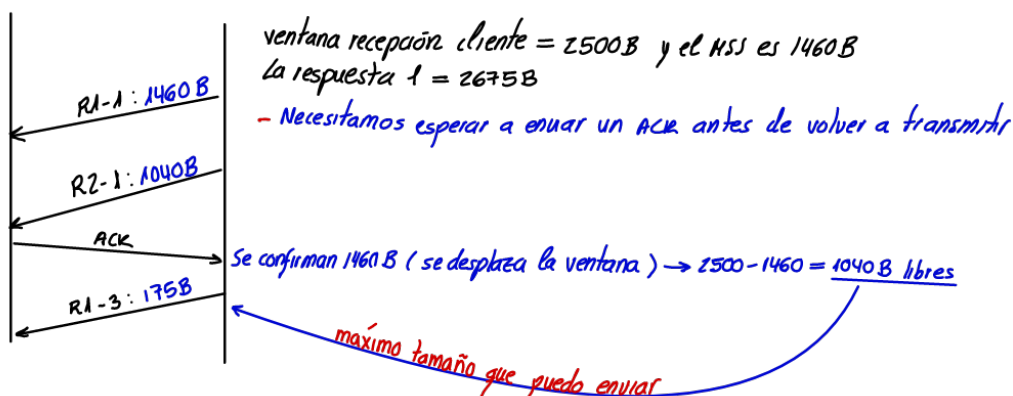
- Tamaño que se está considerando en cada etapa (RTT) tam-max que hay que considerar para transmitir por encima de cualquier límite que existe. (en función de cuantos RTT hayan pasado)

Mínimo el 50% del buffer o mínimo un MSS → lado del receptor
 Algoritmo de Nagle: si está todo confirmado, transmito y si no, me espere a enviar → lado emisor

★ Si tenemos "keep alive", hay una sola conexión en toda la transmisión.

★ El MSS no cuenta las cabeceras → Si el MSS es de 1000, se puede transmitir una trama de 1020 B (1000 B datos + 20 B cab)

★ HAY TAMAÑO MÁXIMO DE VENTANA EN AMBOS EXTREMOS (EXAMEN)



★ NO HAY CONTROL DE LA CONGESTIÓN: Envío todo lo que pueda teniendo en cuenta el MSS y el tamaño de la ventana de recepción