

Nombre:

DNI:

Teoría y problemas (75 %)

1. (1 punto) Suponga una aplicación P2P de compartición de ficheros en la que existe un servidor central que ofrece un servicio de búsqueda de los contenidos compartidos por los peers. Explique de qué manera podrían comunicarse dos o más peers en el caso de que todos ellos estuvieran en redes privadas. Suponga que el servidor central tiene una dirección IP pública y que los peers no tienen acceso a su NAT para redireccionar “a mano” los puertos.

Para que una conexión entrante hacia un proceso peer supere el NAT dicho peer primero ha debido establecer una comunicación a través del NAT. Supongamos que cada proceso peer envía primero un datagrama al servidor central, y que éste anota el puerto utilizado por el NAT y la IP del mismo. Cuando otro peer quiere conectarse con el primer peer, obtiene del servidor central la IP del NAT y el puerto que va a permitir llegar la petición de establecimiento de conexión hasta el peer. Este razonamiento sería válido tanto para el UDP como para el TCP.

2. (0,5 puntos) Indique si es cierta o falsa la siguiente afirmación: “Dos procesos distintos pueden escuchar en el mismo puerto en el mismo host siempre y cuando ambos no utilicen el protocolo UDP.” Razone su respuesta.

Es cierta, siempre y cuando esos procesos sean los que establecen la conexión. En TCP se desmultiplexa utilizando la (IP origen, puerto origen, IP destino, puerto destino). Sólo hay una excepción, cuando los procesos que se conectan a los dos procesos que escuchan se ejecutan en la misma IP y además lo hacen a través del mismo puerto. Esto sin embargo es muy raro porque la API utiliza normalmente puertos distintos para procesos diferentes que no ‘‘escuchan’’, sino que son ‘‘clientes’’.

3. (0,25 puntos) ¿Qué indica el campo window de la cabecera TCP?

El número máximo de bytes que pueden ser transmitidos sin confirmación.

4. (0,25 puntos) ¿Es cierto que el campo ack de la cabecera TCP del último paquete recibido siempre vale seq + 1 + el tamaño del payload del último segmento enviado? Razone su respuesta.

Esto es falso. Sólo tenemos que pensar en qué ocurre cuando se pierde un segmento y comienzan a recibirse ACKs duplicados.

5. (0,25 puntos) ¿Puede un emisor controlar el flujo de datos usando el UDP? Si así es, ¿cómo?

Claro, limitando el número de paquetes que son enviados por unidad de tiempo. Este control de flujo se realizaría a nivel de aplicación.

6. (1 punto) Un servidor Web almacena una página HTML que referencia a 30 objetos Web. Suponiendo una capacidad infinita para los enlaces y que cada objeto tiene un tamaño inferior a un MSS, ¿cuántos RTTs son necesarios para descargar todos los objetos? Efectúe el cálculo en las siguientes situaciones:
- a) Conexiones HTTP no persistentes y un servidor Web que atiende secuencialmente a las peticiones.
 - b) Conexiones HTTP no persistentes y hasta 10 conexiones TCP paralelas.
 - c) Conexiones HTTP persistentes y un servidor Web secuencial.
 - d) Conexiones HTTP persistentes, un servidor Web que soporta pipelining y un cliente que realiza las peticiones en paralelo.

Apartado 1.

Todos los objetos son descargados secuencialmente.

$31 \text{ objetos} * 2\text{RTT/objeto} = 62 \text{ RTTs.}$

Recuerda que el establecimiento de la conexión tarda un RTT.

Apartado 2.

Descarga de la página: 2RTT.

Descarga de los 10 primeros objetos incrustados: 2RTT.

Descarga de los 10 siguientes objetos incrustados: 2RTT.

Descarga de los 10 últimos objetos incrustados: 2RTT.

Total: 8RTT.

Apartado 3.

Descarga de la página: 2RTT.

Descarga del resto de objetos: 30RTT.

Total: 32RTT.

Apartado 4.

Descarga de la página: 2RTT.

Descarga del resto de objetos: 1RTT.

Total: 3RTT.

7. (0,5 puntos) El ejercicio anterior está idealizado y por eso existe una diferencia tan grande entre algunos de los contextos analizados. ¿Qué factores provocan que en la práctica dichas diferencias sean menores?

La tasa de transmisión, el tamaño de los objetos y la multiplexación de los enlaces. Como es lógico aunque se soliciten 30 objetos la vez estos llegan al cliente en serie.

8. (0,5 puntos) Imagine que es un hacker y va a acceder ilegalmente a un host que está en tu misma red local. El servicio que va a usar es el ssh que está disponible en todas las máquinas de su red local. ¿Cómo haría para que la dirección física del interface de red de la estación a "hacker-ar" no apareciera en la tabla ARP de su host mientras permanece conectado a ella? Nota: no tiene privilegios de administrador en ninguna de las máquinas y por tanto no puede manipular las tablas ARP de ninguna de ellas.

Conectándonos al host destino a través de un host intermedio.

9. (2 puntos) Considere dos hosts conectados por un enlace con una capacidad de 10^6 bits/segundo y con un retraso de propagación de 10×10^{-3} segundos. Se desea transmitir un fichero de 4000 bytes. Cada paquete tiene un tamaño máximo de 1000 bytes (incluyendo 40 bytes de cabecera). Suponiendo que los paquetes con los ACKs tienen 125 bytes, que se tarda 2×10^{-3} segundos en comprobar si un paquete recibido contiene errores de transmisión, que dichos errores afectan a cada paquete cuyo índice es múltiplo de 3 y que se utiliza un protocolo ARQ con repetición selectiva que sólo usa reconocimientos positivos y un tamaño de ventana de 3 paquetes. ¿Cuánto tiempo pasaría, como mínimo, desde que el emisor envía el primer paquete hasta que el último ACK es recibido? En sus cálculos utilice un time-out acorde con los datos proporcionados.

El número de paquetes que como mínimo hay que enviar es:

$$\text{floor}(4000/(1000 - 40)) = 5$$

Por tanto, los 4 primeros paquetes tienen 1000 bytes de longitud y el último $4000 - 4 \times 960 + 40 = 200$ bytes.

Un paquete de 1000 bytes tarda en enviarse:

$$1000 \times 8 \text{ bits} / 10^6 \text{ bits/segundo} = 8 \text{ ms}$$

El paquete de 200 bytes tarda en enviarse:

$$200 \times 8 \text{ bits} / 10^6 \text{ bits/segundo} = 1,6 \text{ ms}$$

Un ACK tarda en enviarse:

$$125 \times 8 \text{ bits} / 10^6 \text{ bits/segundo} = 1 \text{ ms}$$

El time-out tiene que ser mayor o igual que

$$\text{RTT} + \text{tiempo_de_chequeo_errores} + \text{envío_ACK} = 10 \times 2 + 2 + 1 = 23 \text{ ms}$$

ms	Emisor	Receptor
0	.. Comienza en el envío del paquete P1	
8	.. Finaliza envío P1 y comienza envío P2	
10	Comienza recepción P1
16	.. Finaliza envío P2 y comienza envío P3	
18	Finaliza recepción P1 y comienza recepción P2
20	Comienza envío ACK1
21	Finaliza envío ACK1
24	.. Finaliza envío P3	
26	Finaliza recepción P2 y comienza recepción P3
28	Comienza envío ACK2
29	Finaliza envío ACK2
30	.. Comienza recepción ACK1	
31	.. Finaliza recepción ACK1 y comienza envío P4	
34	Finaliza recepción P3 (con errores)
38	.. Comienza recepción ACK2	
39	.. Finaliza recepción ACK2, finaliza envío P4 y comienza envío P5	
40,6	Finaliza el envío de P5
41	Comienza recepción P4

47 .. Comienza reenvío P3
49 Finaliza recepción P4 y comienza recepción P5
50,6 Finaliza recepción P5
55 .. Finaliza reenvío P3
57 Comienza recepción P3
65 Finaliza recepción P3
67 Comienza envío ACK5
68 Finaliza envío ACK5
77 .. Comienza recepción ACK5
78 .. Finaliza recepción ACK5

10. (0,25 puntos) Indique si es cierta la siguiente afirmación: Para que un host pueda pertenecer a un grupo multicast debe existir un router multicast en su sub-red. Si así fuera, ¿a qué es debido esto?

Totalmente cierta. El router multicast gestiona la llegada del tráfico multicast a la sub-red.

11. (0,5 puntos) Explica cómo conseguir que una transmisión multicast no salga de la red local aunque nuestro router sea multicast.

Poniendo a 1 el TTL de los datagramas enviados al grupo multicast.

12. (0,5 puntos) Si todos los enlaces en Internet proporcionaran un servicio de entrega fiable, ¿sería el servicio de entrega fiable del TCP redundante?

No. El TCP soluciona además los siguientes problemas:

1. Los errores en los routers (por modificación de datos y por routing incorrecto).
2. Los errores de reordenación y duplicado de los datos.
3. El control de la congestión.

Prácticas (25 %)

- 0,1 puntos por pregunta.
- Una mal resta una bien.

V F Verdadero/Falso.

- En una red como la del laboratorio de redes (actualmente una red Ethernet 100 BaseT), con Ethereal podemos capturar todos los frames que circulan por toda la red.
- Cuando Ethereal se configura para capturar los frames desde el interface de red Ethernet, todos los frames capturados son Ethernet.
- Hay una diferencia fundamental entre el pinger UDP creado en prácticas y el programa estándar ping. El primero utiliza paquetes UDP y el segundo paquetes ICMP.
- Cuando se ejecutan los programas PingServer y PingClient en la misma computadora y el parámetro LOSS_RATE es cero, todavía es frecuente que algunos paquetes no puedan ser devueltos por el servidor.
- La clase SMTPConnection desarrollada en la práctica del agente de correo se encarga de realizar la interacción SMTP entre el agente y el servidor de SMTP.
- La consulta nslookup -type=NS ual.es se realiza siempre a uno de los servidores de nombres autorizados para el dominio ual.es.
- Cuando descargamos un conjunto de objetos Web desde un servidor a nuestro host, el número de consultas realizadas al DNS es igual al número de objetos Web solicitados.
- En su versión inicial, el proxy Web desarrollado en prácticas no cachea los objetos Web. Por tanto, su uso va a ralentizar en la mayoría de las ocasiones nuestra navegación por la Web.
- El proxy Web desarrollado en prácticas no utiliza conexiones TCP paralelas. Por tanto, la descarga de una página Web con múltiples objetos incrustados va a ser generalmente lenta.
- Cuando usamos el protocolo HTTP para transmitir un fichero de datos, dependiendo de la longitud de dicho fichero serán generados uno o varios paquetes UDP.
- En TCP, cuando un segmento tiene el flag ACK activado significa que no transporta datos.
- Cuando accedemos a una página Web mediante nuestro navegador por primera vez se produce un establecimiento de conexión TCP. Tras capturar los frames con Ethereal y establecer el filtro tcp deberíamos ver que los tres primeros segmentos enviados pertenecen a dicho establecimiento de conexión.
- Para calcular la tasa de transmisión real de una conexión TCP podemos dividir el número total de bytes transmitidos entre el tiempo que transcurre desde que se establece la conexión TCP hasta que se recibe el último ACK del último segmento transmitido.
- En una gráfica donde se muestra el número de secuencia de los segmentos transmitidos en función del tiempo podemos apreciar la acción del control de la congestión realizada por el TCP.
- El periodo de arranque lento en el TCP se reconoce porque el número de segmentos enviados sin confirmación es menor que el tamaño de la ventana especificada por el receptor.
- Sabemos que un datagrama IP ha sido fragmentado porque el flag MF (More Fragments) es 0 y el campo Fragment Offset es distinto de 0.
- La aplicación traceroute se aprovecha de que los routers envían un datagrama ICMP con el mensaje TTL excedido hacia el host emisor del datagrama que ha sido destruido por dicho motivo.

- Si el campo `Fragment offset` de un datagrama IP es distinto de cero entonces dicho datagrama es un fragmento de otro datagrama mayor.
- Cuando un datagrama es fragmentado, el tamaño de todos los fragmentos es siempre el mismo.
- Todos los datagramas ICMP generados por `pingplotter` deberían tener la misma IP destino cuando estamos calculando la ruta entre nuestro host y `www.ace.ual.es`.
- Tras deshabilitar el análisis del protocolo IP en Ethereal, todas las direcciones físicas fuente y destino que aparecen en la lista de frames capturados pertenecen a interfaces de nuestra red local.
- Todos los frames que encapsulan datagramas que van dirigidos a nuestro host y que vienen desde otro que está fuera de nuestra red tienen como dirección física origen la de un router de nuestra red.
- Si mostramos el contenido de nuestra tabla ARP y sólo aparece una entrada para el router por defecto, entonces no existe ninguna computadora más que utiliza la red en nuestra sub-red.
- Si en una petición HTTP aparece entre las líneas de cabecera la entrada `IF-MODIFIED-SINCE`, entonces la caché de nuestro navegador Web está vacía.
- Si una página Web contiene dos objetos el número de peticiones HTTP es igual a dos independientemente del tamaño de los objetos.