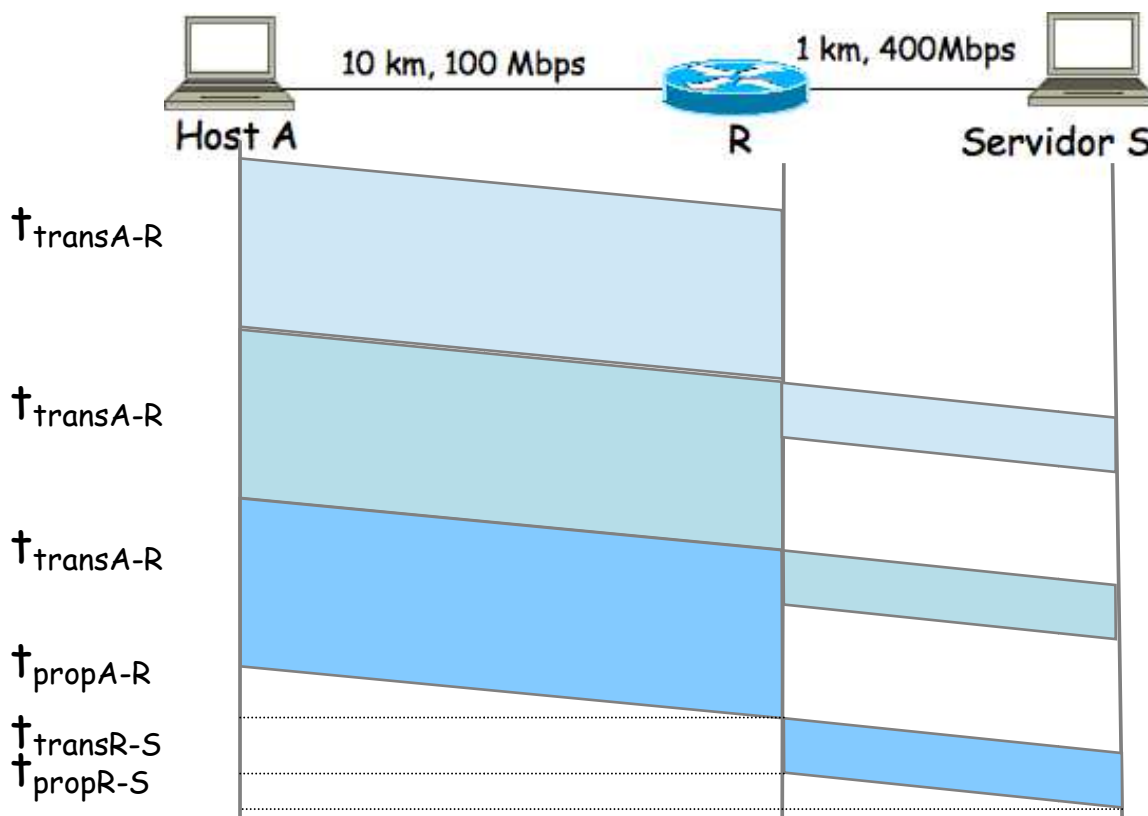




Apellidos, Nombre: _____ **Soluciones** _____ Grupo: _____

- El host A quiere enviar un mensaje de 6000 Bytes al servidor S mediante conmutación de paquete. La ruta entre ambos computadores atraviesa un único router: R. El enlace A-R tiene una longitud de 10 km y el enlace R-S es de 1 km. La velocidad de propagación es $2,5 \times 10^8$ m/s en ambos enlaces. La velocidad de transmisión es 100 Mbps en enlace A-R y de 400 Mbps en el enlace R-S. El tamaño máximo de los paquetes es de 2000 Bytes e incluyen una cabecera cuyo tamaño consideraremos despreciable. También consideraremos despreciables los tiempos de procesamiento en hosts y routers (no los de espera en colas). No hay errores en la transmisión. Se pide:
 - Representa en un esquema gráfico espacio-temporal el tránsito de los paquetes por la red de A a S. Indica en el mismo qué tiempo corresponde al t_{trans} y al t_{prop} en cada enlace.
 - Apoyándote en el esquema del apartado a), calcula el tiempo total transcurrido, desde que se inicia la transmisión del primer paquete hasta que el último paquete se acaba de recibir en el servidor S. **(2 puntos)**



$$t_{transA-R} = (2000 \cdot 8) / 10^8 = 16 \cdot 10^{-5} = 160 \mu s$$

$$t_{transR-S} = (2000 \cdot 8) / 4 \cdot 10^8 = 4 \cdot 10^{-5} = 40 \mu s$$

$$t_{propA-R} = 10^4 / 2,5 \cdot 10^8 = 0,4 \cdot 10^{-4} = 40 \mu s$$

$$t_{propR-S} = 10^3 / 2,5 \cdot 10^8 = 0,4 \cdot 10^{-5} = 4 \mu s$$

$$T_{total} = 3 \cdot t_{transA-R} + t_{propA-R} + t_{transR-S} + t_{propR-S} = 480 + 40 + 40 + 4 = 564 \mu s$$



2. Explica cómo influye el mecanismo de “Slow start” en el uso de conexiones persistentes en HTTP. (1 punto).

Al principio de una conexión TCP se comienza enviando segmentos lentamente según el mecanismo denominado “Slow start”. A medida que se comprueba que los segmentos llegan al otro extremo (por la recepción de los correspondientes ACKs) se va aumentando la velocidad de envío. Si se usan conexiones no persistentes, para cada objeto de una página web se abre una conexión diferente. Como la mayoría de objetos son pequeños, tan pronto la conexión alcanza una velocidad de transferencia alta, se cierra la conexión y se inicia una nueva que debe volver a aplicar la técnica de “comienzo lento” y, por tanto, sufrir un nuevo periodo de baja velocidad de transferencia. El uso de conexiones persistentes permite que la fase de “slow start” se aplique una sola vez y que la mayoría de objetos de una página se transfieren una vez se ha alcanzado una velocidad de transmisión más eficiente.

3. Para poder traducir cualquier nombre de dominio a su correspondiente dirección IP, ¿qué información mínima necesita mi computador? ¿Qué información mínima deben almacenar los servidores de nombres locales para poder resolver consultas externas a su propio dominio? (1 punto)

Un computador correctamente configurado (que disponga de dirección de red, máscara de red, puerta de enlace predeterminada, etc.) únicamente necesita la dirección IP del servidor DNS local para comenzar una resolución de nombres.

Un servidor de nombres local (correctamente configurado) únicamente necesita la dirección IP de un servidor raíz (root-server) para resolver consultas externas a su propio dominio.

4. Indique las sentencias Java mínimas necesarias para ejecutar las siguientes acciones (no use "import", no defina la clase, no defina el método, no use excepciones, etc. Escriba, por tanto, sólo las sentencias imprescindibles).

a) El programa escucha en el puerto TCP 444, cada vez que se conecta un cliente muestra por pantalla la dirección y el puerto desde donde se ha producido la conexión. A continuación, cierra la conexión. (1,5 punto).

```
ServerSocket ss = new ServerSocket(444);
while(true) {
    Socket s = ss.accept();
    System.out.println("Dirección IP: "+s.getInetAddress().getHostAddress());
    System.out.println("Port: " + s.getPort());
    s.close();
}
```

b) El programa se conecta a un servidor web cuyo nombre se le ha pasado por línea de órdenes, envía una petición GET para obtener la pagina por omisión ("/) e imprime por pantalla la línea de estado obtenida en la respuesta (usa HTTP 1.1). (1,5 punto).

```
Socket s = new Socket(args[0],80);
PrintWriter alServidor = new PrintWriter(s.getOutputStream());
Scanner delServidor = new Scanner(s.getInputStream());
alServidor.print("GET / HTTP/1.1\r\n");
alServidor.print("Host:"+args[0]+"\r\n\r\n");
alServidor.flush();
System.out.println(delServidor.nextLine());
```



5. El control de flujo TCP utiliza una indicación de ventana (buffer disponible en el otro extremo) que limita la inyección de segmentos en la conexión. Como sabes, el tamaño máximo que se puede indicar es de 64 KB. Esta limitación, ¿podría afectar a las prestaciones de TCP cuando se utilizan redes de Ethernet ($V_{trans}=10\text{Mbps}$) con RTTs de 2 ms? (1 punto)

El problema de eficiencia en TCP se produce cuando el tiempo para la transmisión de la ventana es considerablemente menor que el RTT, puesto que en este caso la línea se encuentra libre la mayor parte del tiempo mientras se esperan los ACK's, de forma similar al caso de parada y espera (Stop&Wait).

*Dadas las prestaciones de la red, el emisor TCP sería capaz de enviar en el tiempo correspondiente a un RTT un total de **BpR** bits en cada RTT, donde*

$$BpR = RTT \times V_{trans} = 2 \times 10^{-3} \times 10 \times 10^6 = 2 \times 10^4 \text{ bits} = 2500 \text{ Bytes}$$

Puesto que la ventana máxima es de 64KB, en este caso el tamaño de la ventana no es ninguna limitación para las prestaciones de TCP, ya que el primer ACK llega antes de enviar toda la ventana.

6. El computador A mantiene una conexión TCP con el computador B. A acaba de recibir un segmento de B sin datos, con el flag ACK activado, con número de reconocimiento 5001 y valor del campo WIN=400. Tras recibir ese ACK, el valor de la ventana de congestión de A es de 500 bytes. El MSS utilizado por ambos extremos es de 100 bytes. (2 puntos)
- a) ¿Debe A enviar un ACK a B en respuesta al segmento recibido? ¿Por qué?

No, solo se reconocen los segmentos que llevan datos.

- b) Si el último segmento enviado por A contenía datos en el rango 4901 a 5000 ¿Cuántos segmentos de datos puede enviar A antes de recibir un nuevo segmento de B? Justifica la respuesta.

Puede enviar 400 bytes (el mínimo entre el valor de WIN y el valor de la ventana de congestión). Para el envío de estos 400 bytes debe usar como mínimo 4 segmentos de 100 bytes (MSS=100bytes). Esta sería la solución óptima, pero por distintos motivos A puede decidir enviar esos 400 bytes usando un número mayor de segmentos (por ejemplo, 8 segmentos de 50 bytes)

- c) Si esta conexión usa la opción de escala de ventana con $k=3$ ¿Cuántos bytes de datos puede enviar A antes de recibir un nuevo segmento de B? Justifica la respuesta.

Si se usa la opción TCP de escala de ventana con $k=3$, la ventana anunciada por B con el valor de WIN=400, se debe interpretar como $400 \cdot 2^3 = 3200$ bytes. Este es el máximo número de bytes que puede enviar A antes de recibir un nuevo reconocimiento de B, en el que B podría autorizar nuevos envíos.