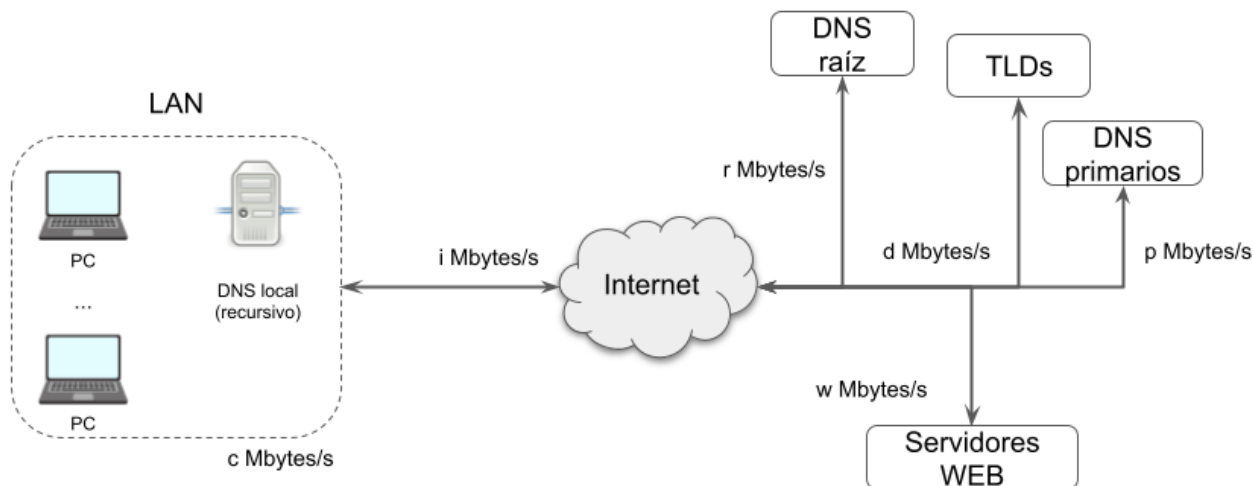


Las respuestas no razonadas, o no que utilicen estrictamente la notación indicada, no serán consideradas como válidas, aunque sean correctas. Recuadra CLARAMENTE tu respuesta a cada apartado. Consigna tu NIA, nombre y apellidos completos en todas las hojas que entregues.

1. [5 ptos.] Imagina el siguiente escenario:



En cada enlace, se muestra el ancho de banda disponible, tanto de subida como de bajada (p. ej.: el ancho de banda del acceso a los servidores DNS TLDs es d MBytes/s, y c MBytes/s en el interior de la LAN). Ésta se conecta a Internet con un *router* de acceso, cuyo enlace puede procesar i Mbytes/s (el tiempo de procesamiento de los paquetes por parte del propio *router* puede considerarse despreciable). Además,

- Las peticiones y respuestas DNS tienen un tamaño medio de t_d bytes. En este momento, el DNS local, de tipo recursivo, solo contiene los registros que se muestran en la tabla inferior.
- Las peticiones y respuestas WEB tienen un tamaño medio de t_w , a no ser que se indique otra cosa.
- El tiempo necesario para establecer una conexión TCP es t_c segundos.

En esta situación, calcula el tiempo medio de procesamiento o carga para las siguientes acciones:

- A. Un usuario visita la URL '<http://www.trust.net>' desde la LAN.
- B. Tras finalizar la carga del recurso anterior, el mismo usuario anterior descarga ahora el fichero '<http://www.trust.net/presentacion.wav>', de 50 MBytes de tamaño.
- C. A continuación, el usuario visita la página <http://www.uam.es/info.html>, que contiene a su vez 5 objetos embebidos de t bytes cada uno.

En las situaciones anteriores, considera siempre que el servidor utiliza HTTP/1.1, limita a 3 las conexiones concurrentes desde la misma dirección IP, y no soporta *pipelining* de recursos.

DNS local Corp	
TLD .com	x.x.x.x
TLD .es	y.y.y.y
elpais.es	z.z.z.z
aquejarre.net	a.a.a.a

Solución:

- A. La conexión a la URL, implicará primero su resolución DNS, por lo que podemos distinguir dos grandes tiempos:

$$t_{TOTAL} = t_{DNS} + t_{CARGA}$$

Calculémoslos por partes. El primero de los sumandos anteriores implica la comunicación con el DNS local, quien comprueba que no dispone del registro asociado al dominio solicitado, ni el TLD del dominio .net. Por tanto, es necesario acudir a los DNS raíz, para solicitar la dirección IP de dicho TLD.

El tiempo total será entonces los tiempos de ida y vuelta al servidor local (t_{DNSl}^1 y t_{DNSl}^2 , respectivamente), servidores raíz, TLD .net y servidor primario:

$$t_{DNS} = t_{DNSl}^1 + 2 \cdot t_{root} + 2 \cdot t_{TLD} + 2 \cdot t_{PRI} + t_{DNSl}^2$$

Podemos calcular cada tiempo de la siguiente forma:

1. Comunicación entre el cliente y el servidor DNS local, que se realiza en la LAN:

$$t_{DNSl}^1 = t_d/c$$

2. El servidor DNS local es recursivo, por lo que el resto de peticiones las realiza él mismo. Primero, consulta a los servidores raíz:

$$t_{root} = t_d/i + t_d/r + t_d/r + t_d/i = 2(t_d/i + t_d/r)$$

3. Con la IP recibida, se consulta al TLD.net para solicitar el servidor primario del dominio:

$$t_{TLD} = t_d/i + t_d/d + t_d/d + t_d/i = 2(t_d/i + t_d/d)$$

4. Con la respuesta recibida, el DNS local consulta ahora al servidor primario del dominio .trust.net:

$$t_{PRI} = t_d/i + t_d/p + t_d/p + t_d/i = 2(t_d/i + t_d/p)$$

5. Finalmente la respuesta debe volver del DNS local a la máquina del cliente

$$t_{DNSl}^2 = t_d/c$$

Por lo que, finalmente, tenemos que:

$$t_{DNS} = 2t_d/c + 6t_d/i + 2t_d/r + 2t_d/d + 2t_d/p$$

Una vez que la máquina del usuario dispone de la dirección IP adecuada, puede empezar la conexión con el servidor Web y la descarga del recurso. El tiempo de conexión es t_c , luego pasamos a calcular el tiempo de carga del recurso 'http://www.trust.net'. Como no se nos dice lo contrario, entendemos que tendrá un tamaño t_w . Por tanto, tenemos que:

$$t_{CARGA} = t_c + \left(\frac{t_w}{c} + \frac{t_w}{i} + \frac{t_w}{w} \right)$$

Finalmente, ya disponemos de todos los sumandos necesarios, y obtenemos que:

$$t_{TOTAL} = t_{DNS} + t_{CARGA}$$

- B. Una vez finalizada la carga del recurso anterior, el DNS local ya dispone de la dirección IP del dominio 'trust.net', por lo que no necesitaremos volver a pedir su resolución en este caso. El tiempo de carga será, ahora, simplemente el tiempo de petición del recurso y su descarga (consideramos la conexión establecida).

El tiempo de petición es, como en el apartado anterior, $t_{peticion} = t_w/c + t_w/i + t_w/w$, por lo que:

$$t_{TOTAL} = t_{peticion} + \frac{50}{w} + \frac{50}{i} + \frac{50}{c} = \frac{50 + t_w}{w} + \frac{50 + t_w}{i} + \frac{50 + t_w}{c}$$

- C. Por último, tenemos la descarga de 5 objetos embebidos en la página 'info.html' del mismo dominio, de t bytes cada uno. Sea t_{INFO} el tiempo de carga de dicha página, que sería igual que t_{CARGA} , sin incluir el tiempo de conexión, puesto que ésta ya está establecida:

$$t_{INFO} = \left(\frac{t_w}{c} + \frac{t_w}{i} + \frac{t_w}{w} \right)$$

Ahora, comenzará la descarga de los objetos. El servidor soporta 3 conexiones simultáneas desde la misma dirección IP, por lo que los objetos serán servidos en lotes. Sea t_{OBJ} el tiempo de descarga de un objeto:

$$t_{OBJ} = \left(\frac{t}{c} + \frac{t}{i} + \frac{t}{w} \right)$$

Por tanto, los 5 objetos se descargarán en dos lotes, cada uno de los cuales tomará t_{OBJ} . Así que finalmente:

$$\begin{aligned} t_{TOTAL} &= t_{INFO} + 2t_{OBJ} \\ &= \left(\frac{t_w}{c} + \frac{t_w}{i} + \frac{t_w}{w} \right) + 2 \left(\frac{t}{c} + \frac{t}{i} + \frac{t}{w} \right) \\ &= \left(\frac{2t + t_w}{c} + \frac{2t + t_w}{i} + \frac{2t + t_w}{w} \right) \end{aligned}$$

2. [2 ptos.] Imagina que Alicia desea enviar un correo electrónico a Bernardo. ¿Qué protocolos a nivel de red y aplicación podrían utilizarse entre las siguientes partes? Copia y rellena una tabla como la mostrada más abajo, indicando entre paréntesis junto a cada protocolo su puerto asociado.



	Alicia > Servidor A	Servidor A > Servidor B	Servidor B > Bernardo
Aplicación			
Transporte			

Solución:

	Alicia > Servidor A	Servidor A > Servidor B	Servidor B > Bernardo
Aplicación	SMTP (25)	SMTP (25)	POP3 (110), IMAP (143), HTTP (80,443)
Transporte	TCP	TCP	TCP

3. [3 ptos.] Imagina una serie de objetos HTTP, O_1, \dots, O_n , embebidos en una página Web, cada uno de tamaños s_1, \dots, s_n . Además, sabemos que todos los objetos son de igual tamaño, excepto el primero, que es muy superior al resto. Explica de **forma clara y concisa** cómo gestionaría un navegador la descarga secuencial de dichos objetos si el servidor Web:
1. Solo soporta HTTP/1.1, y no permite pipelining ni conexiones simultáneas.
 2. Solo soporta HTTP/1.1, no permite pipelining pero sí n conexiones simultáneas por cliente.
 3. Un servidor Web que soporta HTTP/2, y una única conexión simultánea.

Responde a la siguientes preguntas:

1. Para cada uno de los escenarios anteriores, ¿cuál sería el tiempo total de descarga? Se considera que la descarga finaliza solo cuando se han recibido todos los objetos.
2. ¿Cuál es la gran ventaja que ofrece HTTP/2 en la descarga de estos objetos respecto a HTTP/1.1?

Solución:

1. **HTTP/1.1, sin pipelining ni conexiones simultáneas**

En este caso, el objeto O_1 actúa como un 'tapón' en la comunicación (llamado *efecto HOL*), haciendo que no pueda comenzar la descarga del resto de objetos hasta que no finalice ésta. Por tanto, si t_i el tiempo de descarga del objeto O_i , tenemos que el tiempo total de descarga es:

$$t_{total} = t_1 + t_2 + \dots + t_n = \sum_{i=1}^n t_i$$

2. **HTTP/1.1, sin pipelining, n conexiones simultáneas**

Ahora, el servidor permite n conexiones simultáneas, luego todos los objetos pueden comenzar su descarga a la vez. Por tanto, el tiempo total será el mayor de ellos, que claramente corresponde al primer objeto. Por tanto:

$$t_{total} = \max\{t_i\} = t_1$$

3. HTTP/2 con una única conexión simultánea

En este caso, la gran diferencia radica en que HTTP/2 soporta la subdivisión de los objetos (*framing*), dividiéndolos en frames más pequeños, que pueden intercalarse entre sí. Supongamos que el tamaño del frame es el de cualquiera de los objetos, excepto el primero.

Así, la ventaja del protocolo HTTP/2 es que ahora no se produce el bloqueo HOL, puesto que los objetos pequeños se descargarán primero (típicamente, en un esquema Round-Robin). Aún así, es necesario esperar a la descarga completa de O_1 , por lo que el tiempo total es el mismo que en el escenario 1.