

1. Sea una topología de red cliente/servidor, diseñada para distribuir ficheros entre un único servidor 3 y sus k clientes. El servidor dispone de un ancho de banda de subida y bajada de s_s y b_s kbits/s, respectivamente, y cada uno de los clientes de s_i y b_i kbits/s, con $i = 0, \dots, k$. En una red de estas características, responde razonadamente a las siguientes preguntas:

1. **(0,75 ptos)** Imagina que el servidor comienza a distribuir un fichero de tamaño T bytes entre todos sus clientes, simultáneamente. ¿Cuál debería ser su ancho de banda mínimo para no convertirse en el cuello de botella, o factor limitante, de la comunicación?

Imagina ahora que todos los clientes disponen del mismo ancho de banda de bajada, y que el ancho de banda del servidor se reparte equitativamente entre todos los clientes:

2. **(0,5 ptos)** En un momento cualquiera de la distribución del fichero, ¿qué proporción del ancho de banda de bajada de cada cliente queda sin utilizar? Supón que $s_s < kb_i$.
 3. **(0,25 ptos)** ¿Cuál sería el número máximo de clientes que el servidor podría atender?

Por último, trabajaremos con una topología de red P2P pura, diseñada para distribuir ficheros entre sus k nodos. Usaremos la misma notación que el caso anterior, de forma que cada uno de los nodos participantes dispone de un ancho de banda de subida y bajada de s_i y b_i , respectivamente, con $i = 0, \dots, k$. En una red de estas características,

4. **(1,5 ptos)** Existe un fichero de tamaño T bytes en un nodo concreto, k_0 , que desea distribuirse al resto de la red. ¿Cuáles son las condiciones que provocarían que el elemento potencialmente más lento (mayor tiempo mínimo de distribución) de la arquitectura fuera k_0 ?

NOTA: Recuadra claramente el resultado de cada uno de los apartados.

Solución:

En el caso CS, pregunta (1), se trata simplemente de comparar las expresiones que determinan los tiempos mínimos de cada elemento, por lo que podemos plantear:

$$\frac{kT}{s_s} < \frac{T}{b_{min}} \longrightarrow s_s > kb_{min}$$

Para las preguntas (2) y (3), los cálculos son muy sencillos:

(2) Si todos los clientes tienen en el mismo ancho de banda, y el servidor reparte el suyo equitativamente, cada cliente es atendido con ancho de banda s_s/k . Esto supone s_s/kb_i del ancho de banda de cada cliente, medido en tanto por 1 (puesto que nos piden proporción). La proporción no usada será, por tanto, $1 - (s_s)/(kb_i)$.

(3) De igual forma, el número máximo de clientes k que el servidor puede atender es simplemente:

$$s_s = kb_i \longrightarrow k = \frac{s_s}{b_i}$$

En el caso P2P, se trata de determinar las condiciones que provocarían que el servidor fuera el elemento más lento de la arquitectura. Para ello, simplemente debemos comparar la expresión que determina el mínimo tiempo que éste puede tardar en servir el fichero (a) con la del nodo más lento (b) o la de la red como un todo (c). Es decir,

$$\begin{aligned} \frac{T}{s_0} &> \frac{T}{b_{min}} \longrightarrow b_{min} > s_0 \\ s_0 + \sum_{i=1}^{k-1} s_i &> (k-1)b_{min} \end{aligned}$$

Simultáneamente, se debe cumplir también que $(a) > (c)$, es decir,

$$\frac{T}{s_0} > \frac{(k-1)T}{s_0 + \sum_{i=1}^{k-1} s_i} \longrightarrow s_0 + \sum_{i=1}^{k-1} s_i > (k-1)s_0 \longrightarrow \sum_{i=1}^{k-1} s_i > (k-2)s_0$$

2. Imagina un usuario de una red corporativa donde existe un DNS local recursivo, y cuyas tablas de traducción se encuentran inicialmente vacías. Considera que el establecimiento de una conexión TCP consume un tiempo t_c , la consulta a cualquier servidor DNS, t_d , y el envío de cualquier otra cantidad de datos, $f \cdot t_m$, donde f es la cantidad de Kbytes a transmitir. Considera que el resolver de la máquina no cachea información. En esta situación,
1. **(1 pto)** ¿Cuánto tiempo tardaría el DNS local en resolver las siguientes peticiones? Éstas se reciben y resuelven secuencialmente.
 - a) `http://www.uam.es`
 - b) `ftp://asterix.uam.es`
 - c) `https://www.atari.com`
 - d) `mail.nextspectrum.com`
 2. **(0,5 ptos)** ¿Cómo queda la tabla de traducción del DNS local tras todas las resoluciones?
 3. **(1,5 ptos)** Después de las resoluciones anteriores, un usuario desea visitar la página `http://www.atari.com/launch/index.html`, de tamaño 1 Kbytes, que contiene los siguientes elementos, en ese orden:
 - a) 8 fotografías de 2 Kbytes
 - b) 3 vídeos de 500 Mbytes
 Considera que el servidor Web de Atari utiliza HTTP/1.1, y permite 3 conexiones simultáneas desde el mismo cliente. Considera tanto los casos en los que el navegador del usuario utilice y no pipelining de recursos.

Solución:

Denotaremos como $t_d(x)$ el tiempo de resolución de cierto elemento x de la arquitectura DNS (DNS local, servidores raíz, etc.).

1. Es importante tener en cuenta que nos preguntan por el **tiempo de resolución** de las peticiones **por parte del DNS local**, por lo que no debemos incluir tiempos de conexión o transmisión de contenidos, que entrarán en juegos en los apartados siguientes:
 - a) **www.uam.es**: Todas las cachés están vacías, por lo que hay que elevar la consulta hasta los servidores raíz:

$$t_{UAM} = t_d(\text{raíz}) + t_d(TLD.es) + t_d(NS) = 3t_d$$

- b) **asterix.uam.es**: Gracias a la consulta anterior, el DNS local ya dispone de la IP del TLD .es y del primario (NS) del dominio .uam.es, por lo que puede preguntarle directamente por el subdominio *asterix*:

$$t_{asterix} = t_d(NS) = t_d$$

- c) **www.atari.com**: Se trata ahora del primero dominio .com a resolver, por lo que habrá que subir de nuevo hasta los raíz para conseguir el TLD correspondiente, quedando finalmente:

$$t_{atari} = t_d(\text{raíz}) + t_d(TLD.es) + t_d(NS) = 3t_d$$

- d) **mail.nextspectrum.com**: Ahora, igual que en el caso b), el DNS local ya dispone de la IP de un TLD .com, por lo que puede interrogarle directamente:

$$t_{mail} = t_d(TLD.com) + t_d(NS) = 2t_d$$

2. El DNS local guardará todas las traducciones que han pasado por él, quedando:
3. Veamos primero el caso sin pipelining. El tiempo total de carga será la suma de:

a) El tiempo de carga de la página *index.html*, que será:

$$t_{index} = t_d(local) + t_c + 1Kbytes \cdot t_m = t_d + t_c + t_m$$

b) Sin pipelining, cada conexión (hasta un máximo de 3), solo puede descargar un recurso simultáneamente, por lo que (consideramos que los recursos se cargan en el orden en el que aparecen en la página HTML):

- 1) 3 primeras fotografías (t_c por el establecimiento de las conexiones, que se hace simultáneamente): $t_c + 2t_m$
- 2) 3 siguientes fotografías: $2t_m$
- 3) 3 siguientes fotografías y uno de los vídeos: $2t_m$
- 4) 2 vídeos: $5 \cdot 10^5 t_m$

Por tanto, el tiempo total final es:

$$t_{carga} = (7 + 5 \cdot 10^5)t_m + 2t_c + t_d$$

4. Con pipelining, y puesto que se define ninguna restricción de ancho de banda máximo, cada conexión puede cargar varios recursos a la vez, por lo que el tiempo de carga de los recursos será el del más lento (un vídeo). Por tanto, el tiempo total de carga será:

$$t_{carga} = (1 + 5 \cdot 10^5)t_m + 2t_c + t_d$$