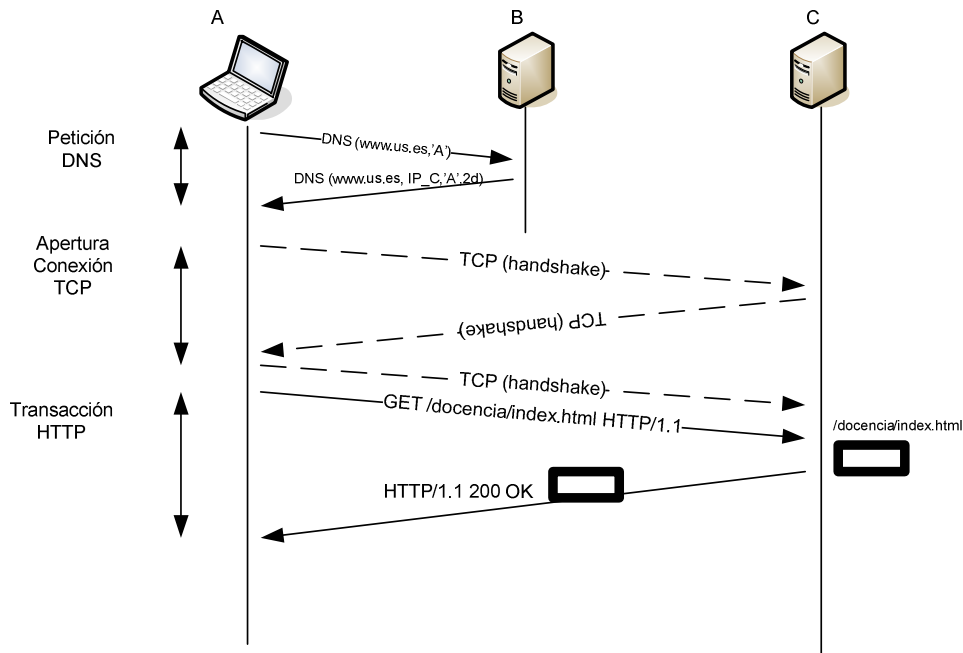


Solución:

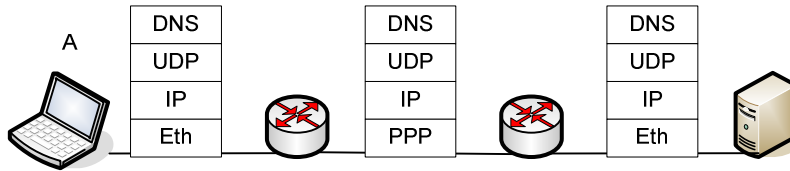
Apartado 1)

Dado que los protocolos pedidos son intercambio de mensajes entre hosts, utilizaremos un esquema con los tres hosts para ilustrar el intercambio de mensajes.



En primer lugar el host A debe averiguar la dirección IP del servidor web www.us.es para poder establecer una conexión TCP con el mismo. Por ello el host A lanza una petición de resolución de nombre DNS al servidor B (servidor local DNS principal). En el mensaje enviado A le indicará que necesita resolver la dirección IP de www.us.es y que quiere una respuesta con un registro de recurso tipo A. El servidor DNS le responde (suponiendo que tiene el registro de recurso almacenado en su caché). Una vez que A tiene la respuesta entonces intenta abrir una conexión TCP con el servidor web (para poder tener accesible un socket TCP desde el que enviar y recibir los mensajes de aplicación). El proceso de apertura de conexión TCP implica un intercambio de tres mensajes TCP entre A y C. Una vez enviado el último mensaje por parte de A, éste procede al envío del mensaje de aplicación (protocolo HTTP) enviando una petición GET al servidor web donde le indica la URL del recurso que solicita (en este caso /docencia/index.html). A la recepción de este mensaje el servidor web busca la página y devuelve su contenido dentro del cuerpo del mensaje de respuesta 200 OK. A la recepción de la respuesta el equipo A podría visualizarla en la pantalla siempre y cuando toda la página web este en el mensaje de respuesta.

Respecto a las cabeceras cada mensaje (incluyendo las de aplicación), podemos destacar que los mensajes de aplicación DNS usan el protocolo de transporte UDP mientras que los mensajes del protocolo HTTP utilizan TCP como protocolo de nivel de transporte. Cualquier mensaje utiliza IP como protocolo de nivel de red. Los mensajes utilizarán el protocolo de nivel de enlace que corresponda al enlace por donde viajan. En el caso de la petición DNS las cabeceras que componen las tramas de nivel de enlace serían:



Para el caso de la apertura de la conexión web y la transacción http sería similar salvo que no existiría capa de aplicación en la apertura de la conexión y el transporte sería TCP en lugar de UDP en el caso del protocolo HTTP.

Apartado 2) Calcular el tiempo que tarda A en tener la página web en su pantalla.

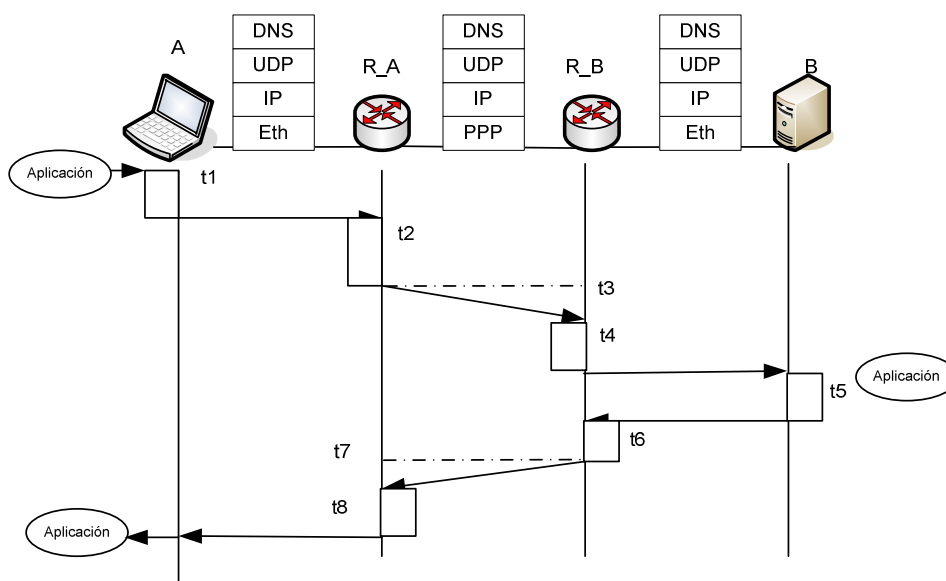
Para este apartado debemos sumar los tiempos que tardan los mensajes anteriores. Para ello debemos tener en cuenta el modelo de retardo de los mensajes extremo a extremo (y no sólo dentro de la red) visto en clase. Debido a que el problema indica que se puede despreciar el retardo de procesamiento y el de espera en cola, sólo nos queda calcular los retardos de transmisión de cada mensaje y el de propagación en aquellos enlaces para los que nos den la distancia (enlaces entre encaminadores en nuestro caso).

Para resolver la pregunta de este apartado debemos fijarnos en el intercambio de mensajes del hosts A; debemos calcular el tiempo que transcurre desde que A comienza a transmitir el primer mensaje hasta que recibe el mensaje 200 OK. Por ello podemos descomponer el tiempo total en tres sumandos:

- T_{DNS} tiempo empleado por A en obtener la dirección IP de C.
- T_{TCP} tiempo empleado por A en la apertura de la conexión TCP
- T_{HTTP} tiempo empleado en la transacción web.

La suma de los tres tiempos anteriores será la respuesta del apartado. A continuación procedemos a calcularlos:

- El tiempo empleado en obtener la dirección IP de B se muestra en la figura siguiente:



Donde t_1 es el tiempo que el host A tarda en transmitir el paquete DNS (nos indican que no hay retardo de propagación en la red de área local), t_2 es el tiempo de transmisión del encaminador R_A, t_3 es el tiempo de propagación del enlace que une R_A con R_B, t_4 es el tiempo de transmisión de R_B. En el camino de la respuesta encontramos que t_5 , t_6 , t_7 y t_8 son idénticos a t_1, t_2, t_3 y t_4 . Por lo tanto:

$$T_{DNS} = 2(t_1 + t_2 + t_3 + t_4)$$

A continuación procedemos a calcular estos valores:

- t_1, t_2, t_4 : suma del tamaño de la trama enviada a la red entre el régimen binario del enlace
- t_3 : longitud del enlace entre velocidad de propagación (2×10^5 km/s).

Por lo tanto:

$$t_1 = (\text{DNS} + H_{UDP} + H_{IP} + H_{Eth}) \text{ bits} / 100 \times 10^6 \text{ b/s} = 0,00592 \text{ ms}$$

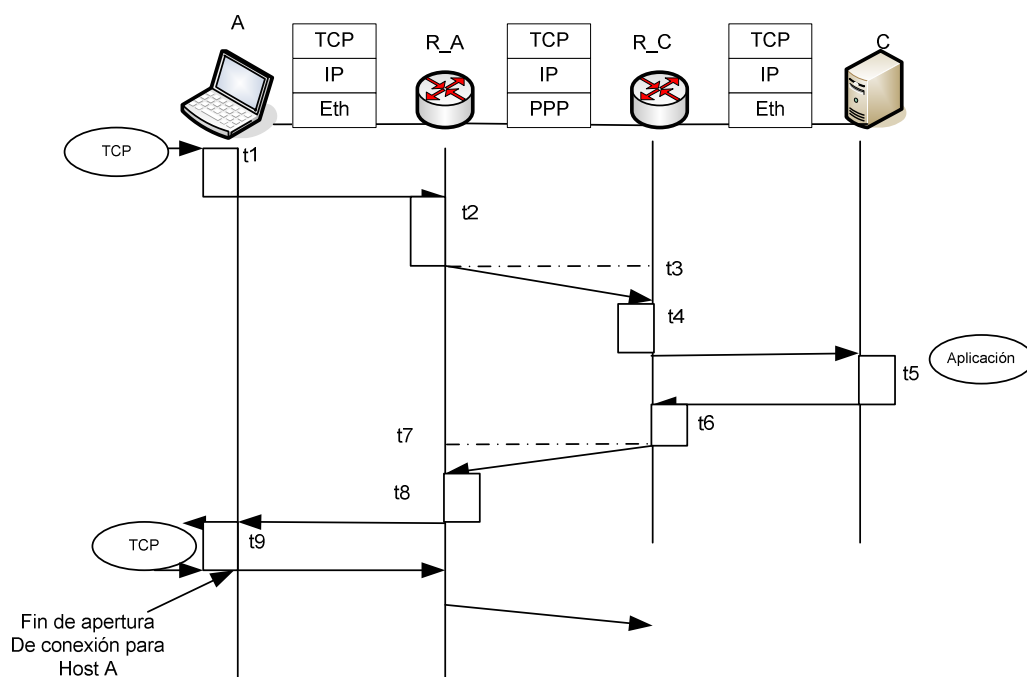
$$t_2 = (\text{DNS} + H_{UDP} + H_{IP} + H_{ppp}) \text{ bits} / 1 \times 10^6 \text{ b/s} = 0,456 \text{ ms}$$

$$t_3 = 100 \text{ km} / (2 \times 10^5 \text{ km/s}) = 0,5 \text{ ms}$$

$$t_4 = t_1$$

Por lo que tenemos $T_{DNS} = 2 (2t_1 + t_2 + t_3) = 1,93 \text{ ms}$

b) El tiempo que el host A emplea en la apertura de la conexión (establecimiento de su socket TCP asociado a las aplicaciones) es el tiempo desde que envía el primer mensaje TCP hasta que termina de enviar el segundo mensaje TCP.



Por lo tanto T_TCP es la suma desde $t1$ hasta $t9$. El cálculo es similar al caso anterior teniendo en cuenta que ahora la torre de protocolos usada (y por tanto la composición de las tramas) es diferente.

$$t1 = (H_TCP + H_IP + H_Eth) \text{ bits} / 100e06 \text{ b/s} = 0,0056 \text{ ms}$$

$$t2 = (H_TCP + H_IP + H_ppp) \text{ bits} / 1e06 \text{ b/s} = 0,424 \text{ ms}$$

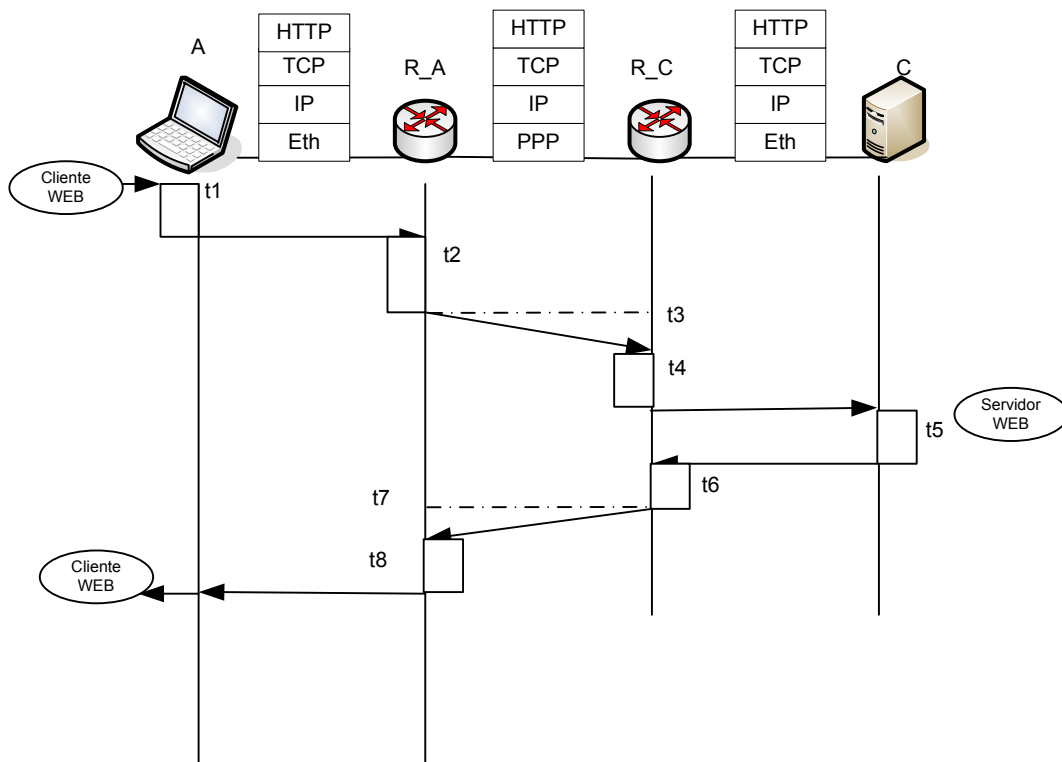
$$t3 = 200 \text{ km} / (2e05 \text{ km/s}) = 1 \text{ ms}$$

$$t4 = t1$$

El camino del segmento de respuesta TCP que viaja de C a A será idéntico al de ida. Pero para completar el cálculo es necesario añadir el tiempo que A tarda en transmitir el tercer mensaje del saludo o handshake TCP (que es $t1$). Por lo tanto:

$$T_TCP = 2(2t1+t2+t3) + t1 = 2,876 \text{ ms}$$

c) Una vez que la conexión TCP esta establecida el host A pasa a enviar el mensaje de aplicación. Por ello transmite un mensaje de petición HTTP con el método GET y espera a la respuesta tal y como se indica en la siguiente figura:



Ahora debemos calcular la suma de los tiempos anteriores. La principal diferencia con los apartados anteriores es la torre de protocolos y que el mensaje 200OK contiene no sólo la línea de estado si no el contenido del fichero solicitado.

$$t1 = (HTTP + H_TCP + H_IP + H_eth) \text{ bits} / 100e06 \text{ b/s} = 0,0072 \text{ ms}$$

$$t2 = (HTTP + H_TCP + H_IP + H_ppp) \text{ bits} / 1e06 \text{ b/s} = 0,584 \text{ ms}$$

$$t3 = 200 \text{ km} / (2e05 \text{ km/s}) = 1 \text{ ms}$$

$$t4 = t1$$

$$t5 = (\text{HTTP} + 28 \cdot 1024 \cdot 8 + H_{\text{TCP}} + H_{\text{IP}} + H_{\text{eth}}) \text{ bits} / 100 \text{e}06 \text{b/s} = 2,3 \text{ ms}$$

$$t6 = (\text{HTTP} + 28 \cdot 1024 \cdot 8 + H_{\text{TCP}} + H_{\text{IP}} + H_{\text{ppp}}) \text{ bits} / 1 \text{e}06 \text{b/s} = 230 \text{ ms}$$

$$t7 = t3$$

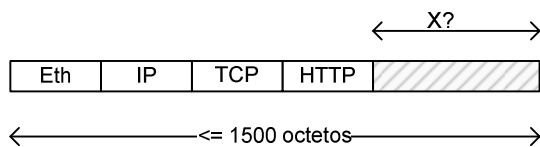
$$t8 = t5$$

$$\text{Por lo tanto } T_{\text{HTTP}} = 2t1 + t2 + 2t3 + 3t5 + t6 = 237 \text{ ms}$$

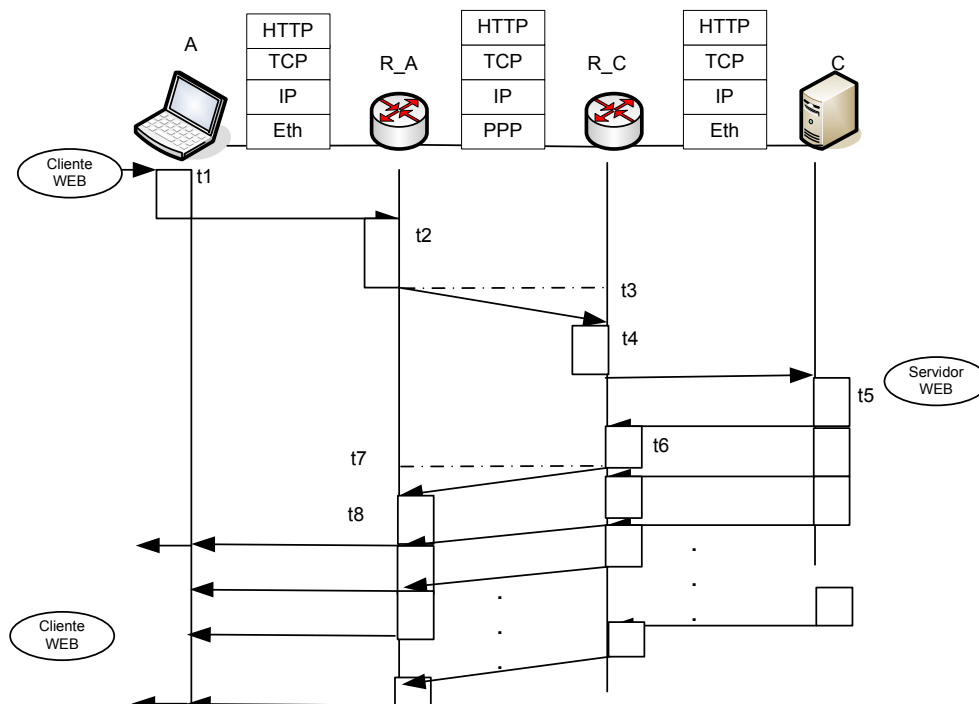
La respuesta a la pregunta es $T_{\text{DNS}} + T_{\text{TCP}} + T_{\text{HTTP}} = 242 \text{ ms}$

Apartado 3) ¿Cómo cambiaría el apartado anterior si el fichero tuviera que ser transferido en segmentos de un tamaño tal que las tramas de nivel de enlace nunca superasen el tamaño de 1500 octetos?

En este caso cambiaría sólo el cálculo del factor T_{HTTP} ya que ahora el fichero pedido no puede ser transmitido en un solo mensaje 200 OK sino en varios. La primera pregunta que hay que responder es ¿Cuántos mensajes necesita C enviar a A para completar la descarga del archivo?. Para ello utilizamos la restricción impuesta: el nivel de enlace nunca podrá manejar tramas de un tamaño mayor a 1500 octetos. Si tenemos en cuenta la composición de las tramas, es sencillo calcular el número de mensajes necesarios. La siguiente figura ilustra el razonamiento.



De donde resulta $X = 1410$ octetos y por lo tanto es necesaria la transmisión de 20 tramas de tamaño 1500 octetos y una última trama de 562 octetos. Hemos utilizado Ethernet en el nivel de enlace ya que impone la mayor de las cabeceras de este nivel.



Todos estos segmentos sería transmitidos secuencialmente desde C hasta A. Se deja al alumno el cálculo del tiempo que A tardaría hasta recibir el último de estos mensajes.

Apartado 4) si el documento tuviera referencias a otros objetos, éstas serían leídas a la llegada del documento html. Después el cliente puede optar por enviar una petición GET para cada objeto de forma secuencial con parada y espera (hasta que no recibe la respuesta de la primera petición no se envía la segunda), o enviarlas consecutivas (con pipelining). En este último caso el cliente envía la segunda petición GET inmediatamente después de la primera. Se deja al alumno el cálculo del tiempo requerido para este apartado en ambos casos tanto sin restricción sobre el tamaño máximo de las tramas como con la restricción del apartado 3.