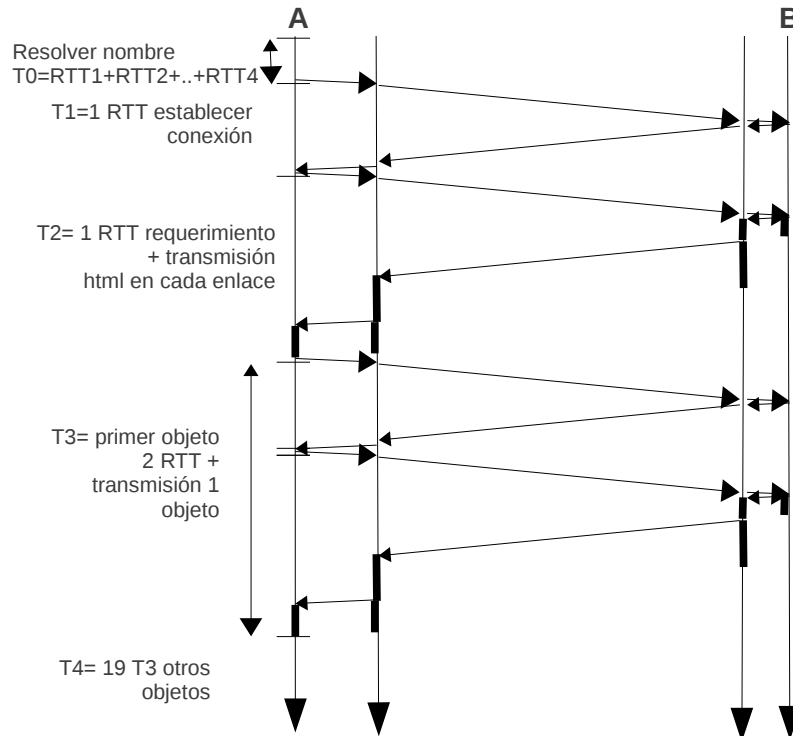


Primer Certamen

Pregunta 1 : Dos terminales A y B están conectados por tres enlaces: E1, E2 y E3. E1 (conectado a A) es de 100 [m] a 100 Mbps, E2 es de 1000[km] de largo a 10 Mbps, E3 es de 20 [m] a 1 Gbps. Considerando una velocidad de propagación $v=2.0 \times 10^8$ [m/s] e ignorando encolamiento y procesamiento. (40 pts)

a) Si un usuario en A baja una página web de B de 100KB. Considerando que se tiene que visitar 4 servidores DNS (RTT1,..., RTT4) para obtener la IP del servidor final y que la página web bajada tiene 20 objetos de 10 KB cada uno. Dé una expresión del tiempo que transcurre hasta obtener todos los objetos usando HTTP no persistente y sin conexiones paralelas.

Supongo que sólo el archivo html tiene tamaño de 100KB. (por diagrama 5 pts)



Sea TT: tiempo total desde que se inicia el requerimiento por parte del usuario hasta que se reciben todos los objetos de la página.

$$TT = RTT1 + RTT2 + RTT3 + RTT4 + 2 * RTT_{A-B} + t_{transmisión\ total\ html} + 20 * (2 * RTT_{A-B} + t_{transmisión\ total\ 1\ objeto})$$

Donde:

$$RTT_{A-B} = 2 * (E1_{largo} + E2_{largo} + E3_{largo}) / v_{prop.}$$

$$t_{transmisión\ total\ html} = HTML_{tamaño} * (1/E3 + 1/E2 + 1/E1)$$

$$t_{transmisión\ total\ 1\ objeto} = Objeto_{tamaño} * (1/E3 + 1/E2 + 1/E1)$$

(Por expresión 5 pts, si no hay diagrama y la expresión es correcta habría que dar 10 pts. En rigor el diagrama no se pedía. Lo mismo vale para otras preguntas similares a ésta).

Si usamos los datos del problema, el tiempo total en este caso sería:

$$RTT_{A-B} = \frac{2 * (100 + 1000 * 10^3 + 20) [m]}{2 * 10^8 [m/s]} \approx 10^{-2} [s] = 10 [ms]$$

$$t_{transmisión\ total\ html} = 800 * 10^3 [bit] * \left(\frac{1}{1 * 10^9 [bit/s]} + \frac{1}{10 * 10^6 [bit/s]} + \frac{1}{100 * 10^6 [bit/s]} \right)$$

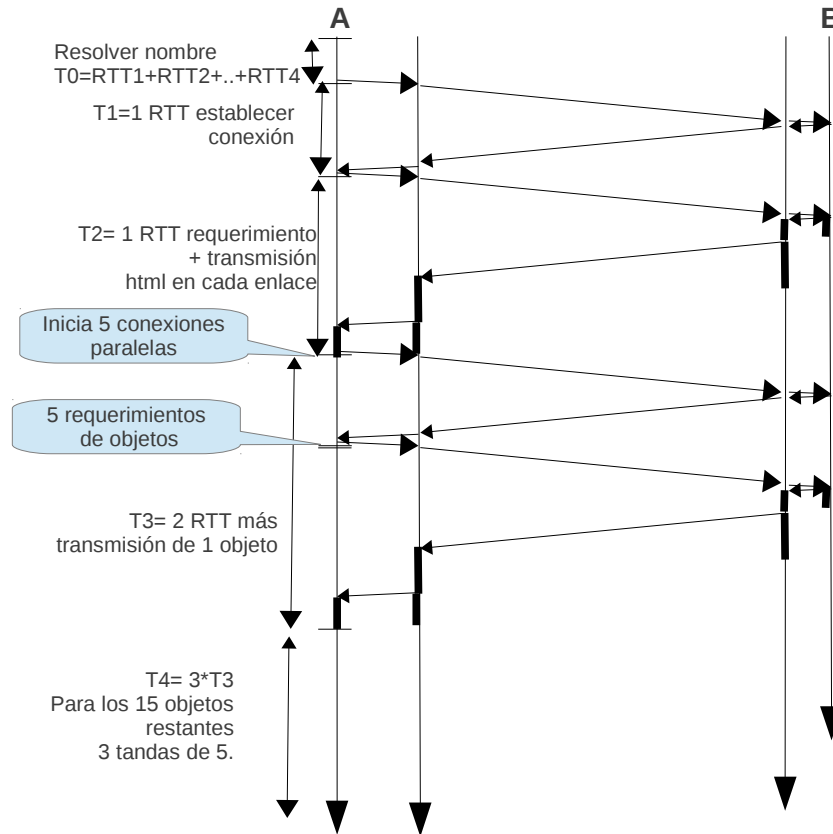
$$t_{transmisión\ total\ html} = 800 * 10^3 * 0.111 * 10^{-6} [s] = 0.888 * 10^{-1} [s] \approx 89 [ms]$$

$$t_{transmisión\ total\ 1\ objeto} = 80 * 10^3 [bit] * 0.111 * 10^{-6} [s] = 0.888 * 10^{-2} [s] \approx 9 [ms]$$

$$TT = RTT1 + RTT2 + RTT3 + RTT4 + 20 + 89 + 20 * (20 + 9) [ms] = RTT1 + RTT2 + RTT3 + RTT4 + 289 [ms]$$

b) Repita la pregunta 1ª con:

- HTTP no persistente y con 5 conexiones paralelas (Por diagrama 3 pts)



$$TT = RTT1 + RTT2 + RTT3 + RTT4 + 2 * RTT_{A-B} + t_{transmisión\ total\ html} + 4 * (2 * RTT_{A-B} + t_{transmisión\ total\ 1\ objeto})$$

Donde:

$$RTT_{A-B} = 2 * (E1_{largo} + E2_{largo} + E3_{largo}) / v_{prop.}$$

$$t_{transmisión\ total\ html} = HTML_{tamaño} * (1/E3 + 1/E2 + 1/E1)$$

$$t_{transmisión\ total\ 1\ objeto} = Objeto_{tamaño} * (1/E3 + 1/E2 + 1/E1)$$

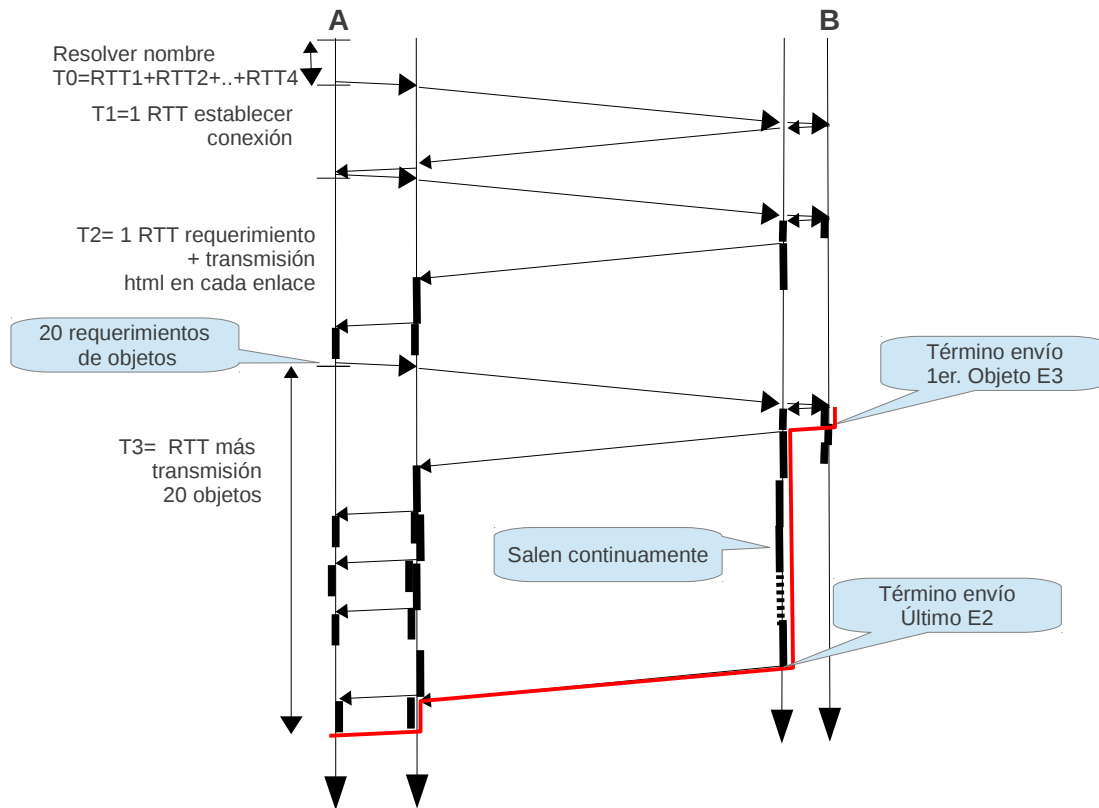
(Por expresión 2 pts)

Si usamos los datos del problema, el tiempo total en este caso sería:

$$RTT_{A-B} = 10 [ms] \quad t_{transmisión\ total\ html} \approx 89 [ms] \quad t_{transmisión\ total\ 1\ objeto} \approx 9 [ms] \quad \text{de a)}$$

$$TT = RTT1 + RTT2 + RTT3 + RTT4 + 20 + 89 + 4 * (20 + 9) [ms] = RTT1 + RTT2 + RTT3 + RTT4 + 225 [ms]$$

- HTTP persistente, con pipeline pero sin conexiones paralelas (Por diagrama 3 pts)



$$TT = RTT1 + RTT2 + RTT3 + RTT4 + 2 * RTT_{A-B} + t_{transmisión\ total\ html} + RTT_{A-B} + Objeto_{tamaño}/E3 + 20 * Objeto_{tamaño}/E2 + Objeto_{tamaño}/E1$$

$$TT = RTT1 + RTT2 + RTT3 + RTT4 + 2 * RTT_{A-B} + t_{transmisión\ total\ html} + RTT_{A-B} + Objeto_{tamaño} * (1/E3 + 20/E2 + 1/E1)$$

Donde:

$$RTT_{A-B} = 2 * (E1_{largo} + E2_{largo} + E3_{largo}) / v_{prop.}$$

$$t_{transmisión\ total\ html} = HTML_{tamaño} * (1/E3 + 1/E2 + 1/E1)$$

(Por expresión 2 pts)

Si usamos los datos del problema, el tiempo total en este caso sería:

$$RTT_{A-B} = 10 [ms] \quad t_{transmisión\ total\ html} \approx 89 [ms] \quad \text{de a)}$$

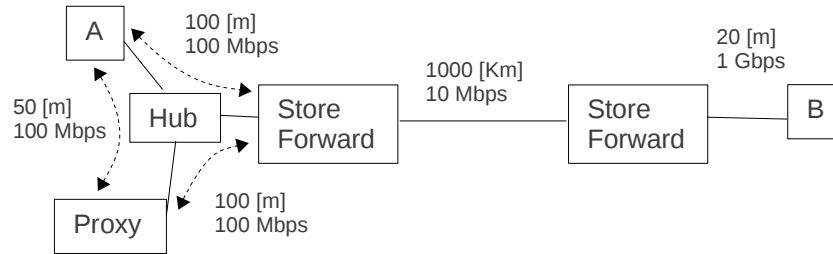
$$Objeto_{tamaño} * \left(\frac{1}{E1} + \frac{20}{E2} + \frac{1}{E1} \right) = 80 * 10^3 [bit] * \left(\frac{1}{1000 * 10^6} + \frac{20}{10 * 10^6} + \frac{1}{100 * 10^6} \right) \left[\frac{1}{bit/s} \right] = 80 * 10^3 * (2.011 * 10^{-6}) [s] \approx 161 [ms]$$

$$TT = RTT1 + RTT2 + RTT3 + RTT4 + (20 + 89 + 10 + 161) [ms] = RTT1 + RTT2 + RTT3 + RTT4 + 280 [ms]$$

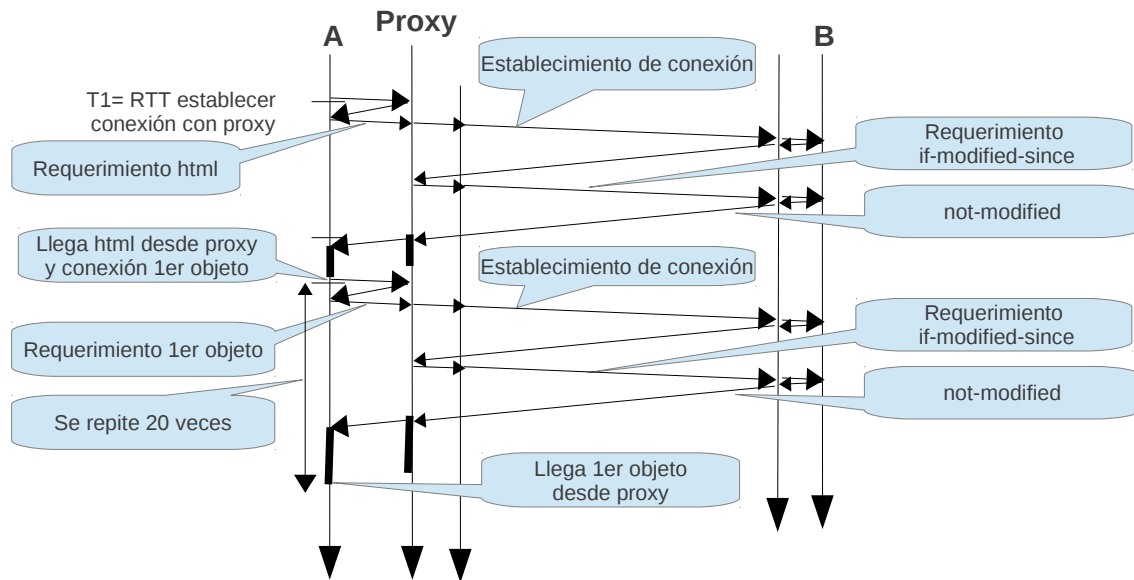
c) Repita la pregunta 1ª pero considerando que un servidor proxy responde por B y que está a 50 [m] de A en su mismo LAN.

En este caso el proxy responde por B, luego la consulta DNS al servidor final la hizo antes B. Supondré que no ha expirado el tiempo para la asociados nombre-IP, luego no es necesario hacer consultas DNS en este caso. Si un alumno incluye el tiempo de resolución DNS, está bien también.

No hay información sobre cómo están conectados los equipos de la red LAN. Supondré una situación como la figura adjunta:



(Por diagrama 5 pts, puede haber más opciones válidas según supuesto de alumno)



$$TT = 2 * RTT_{A-p} + 2 * RTT_{p-B} + HTML_{\text{tamaño}} / E1 + 20 * (2 * RTT_{A-p} + 2 * RTT_{p-B} + Objeto_{\text{tamaño}} / E1)$$

(Por expresión 5 pts)

Si usamos los datos del problema, el tiempo total en este caso sería:

$$RTT_{A-p} = \frac{2 * 50 [m]}{2 * 10^8 [m/s]} = 0.5 [us]$$

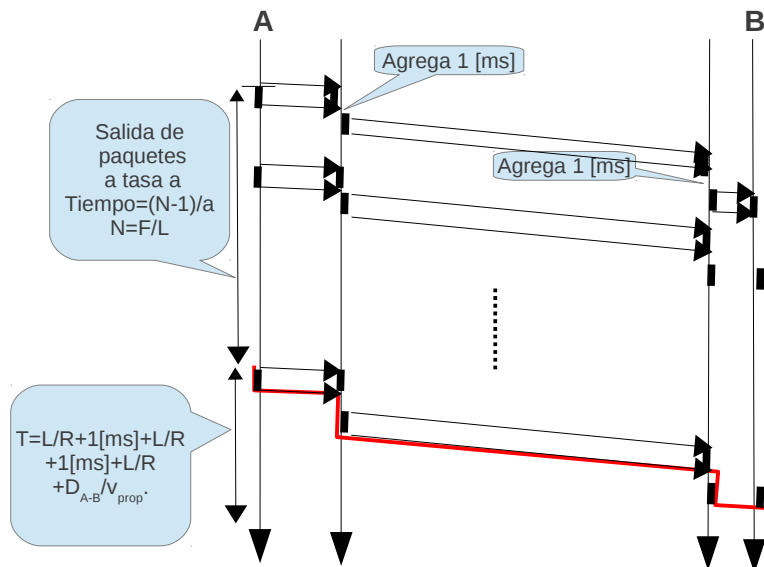
$$RTT_{p-B} = 10 [ms]$$

$$TT = 2 * 0.5 [us] + 2 * 20 [ms] + \frac{800 * 10^3 [bit]}{100 * 10^6 [bit/s]} + 20 * (2 * 0.5 [us] + 2 * 10 [ms] + \frac{80 * 10^3 [bit]}{100 * 10^6 [bit/s]})$$

$$TT \approx 40 [ms] + 8 [ms] + 20 * (20 + 0.8) [ms] = 464 [ms]$$

d) Ahora considere tiempos de encolamiento, propagación y transmisión pero ignorando procesamiento. Si A envía un archivo de F bits a B. Puede asumir paquetes de L bits, a es la tasa de recepción de paquetes (paquetes/s), R es tasa de transmisión (bps). Considere que $L a / R \ll 1$ y que el tiempo mínimo de encolamiento es de 1 [ms]. Obtenga una expresión para el retardo de origen a destino. ¿Qué pasa con retardo si $L a / R = 1$?

Supondremos que cada conmutador posee los parámetros dados.



Si alguien agrega RTT inicial para establecer conexión, OK. Aquí se supuso protocolo tipo UDP.

- i) Retardo = $(F/L - 1)/a + 3 * L/R + 2[ms] + Distancia_{A-B}/v_{prop}$ (6 pts.)
- ii) Como no se ha considerado otros tráficos y éste es determinista; es decir no hay multiplexación estadística, la expresión sigue siendo válida.
 Retardo = $(F+2L)/R + 2[ms] + Distancia_{A-B}/v_{prop}$ (4 pts.)

Pregunta 2: Considerando el retardo de encolamiento en un buffer de un router. Suponga que se tienen paquetes de $L=500$ bits, la tasa de transmisión R es de 100 Mbps, que en promedio llegan $N=1000$ paquetes simultáneamente al buffer cada LN/R segundos. Determine el retardo promedio de un paquete. (Considerando que el retardo para el primer paquete es cero, para el segundo el L/R , etc) (20 pts)

Sea R el retardo promedio en la cola.

Retardo del paquete i ésimo = $R_i = (i-1) * \frac{L}{R}$, Luego

$$R = \frac{1}{N} * \sum_{i=1}^N R_i = \frac{L}{NR} \sum_{i=1}^N (i-1) = \frac{L}{NR} \sum_{i=0}^{N-1} i = \frac{L}{R} * \frac{(N-1)}{2} = \frac{500}{100 * 10^6} * \frac{999}{2} [s] = 2,5 * 999 [us] \approx 2.5 [ms]$$

Hasta aquí 10 pts. /\

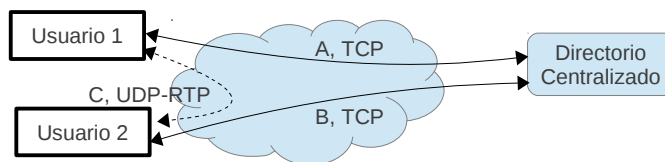
Hasta aquí + 6 pts. /\

Hasta aquí + 4 pts. /\

Pregunta 3: Usted está siendo entrevistado por Skype, para este ejercicio asuma un protocolo llamado RTP que funciona sobre UDP para el envío de la voz. (30 pts)

- a) Explique y diagrame cómo se podría armar una comunicación para hablar usando Voz Sobre IP entre dos usuarios usando un esquema con directorio centralizado. Muestre usuarios, servidores, terminales, red sobrepuesta (si aplica) protocolos.

Se podría usar una arquitectura tipo Napster. (Diagrama 5 pts)

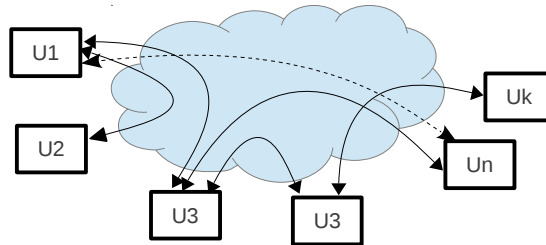


Los usuarios contactan al servidor centralizado para conocer su lista de contacto y usuarios activos en el sistema. Aquí se ha usado TCP para acceder a esta información en forma confiable, conexiones A y B en diagrama. Estos enlaces se pueden usar también para iniciar el llamado (Ring) y al final para señalar que se ha colgado.

C representa el intercambio de paquetes cuando se inicia la transferencia de voz. En C se usa protocolo RTP sobre UDP. (Explicación 5 pts)

b) Como en el problema anterior, explique y diagrame cómo se podría armar una comunicación para hablar usando Voz Sobre IP entre dos usuarios usando un esquema P2P sin directorio centralizado. ¿Cuál sería el rol de la red sobrepuesta? ¿Cómo se generaría y actualizaría esta red? ¿Cuál sería el rol de un ruteo reverso en este esquema?

Se podría usar una arquitectura tipo Gnutella. (Diagrama 5 pts)



La red sobrepuesta aquí permite difundir mensajes de búsqueda de un usuario para llamarlo.

La red se puede generar partiendo con enlaces TCP con un conjunto de otros usuarios seleccionados de una lista de candidatos. Luego a través de éstos enviando mensajes tipo ping y pong es posible ubicar otros usuarios.

El ruteo inverso permitiría enviar las respuestas de búsqueda de usuarios.

Una vez ubicado un usuario se intercambian paquetes de voz entre llamador y llamado usando RTP sobre UDP, línea punteada en diagrama. (Explicación 5 pts)

c) Contraste ambas soluciones anteriores en términos de escalabilidad, seguridad, cuellos de botella, posibles problemas de fallas del sistema y posibles costos de mantenimiento.

La solución b es más escalable. En la solución a) es más fácil proveer seguridad, pues allí se puede validar usuarios.

En a) hay un cuello de botella, el directorio centralizado.

La solución a) es menos robusta. En caso b) si falla una máquina o si se cae, las otras vías pueden ser usadas.

En la opción b) los costos están distribuidos. En a) alguien debe hacerse cargo del servidor central.

(Algo sobre escalabilidad 3 pts, seguridad 3 pts., cuellos de botella 3 pts, falla o costo de mantención 1 pt.)

Pregunta 4: Describa cómo funciona un servidor web (detalle paso a paso el proceso de como se envían páginas web al usuario e incluya el uso de HTML, HTTP, GET, POST) (10 pts)

Un servidor WEB es una aplicación que corre permanentemente y espera conexiones TCP hacia su puerto 80 (por omisión, si bien puede correr en otros puertos). El protocolo capa aplicación que reconoce es HTTP. Cuando los clientes se conectan, este servidor espera la llegada de un requerimiento al cual dar respuesta. En HTTP los requerimientos pueden ser variados, entre ellos el método GET y el POST. Ante la llegada de un requerimiento con método GET, el servidor obtiene el archivo solicitado desde los datos del método, lo busca en disco, prepara la respuesta con el archivo y la envía hacia el cliente. El cliente recibe e interpreta el archivo enviado. HTML es el lenguaje usado para componer una página web. Ante la llegada de un POST, el servidor obtiene los datos subidos por el cliente en la sección datos, por ejemplo los campos de un formulario. Al usar método GET es posible incluir datos de formularios en campo URL del GET.

(Servidor corre permanentemente 2 pts., explicación HTTP 2 pts, explicación GET 2 pts y POST 2 pts., explicación HTML 2 pts.)