**Capítulo 2 Cuestiones de repaso**

**SECCIÓN 2.1**

**R.1.** **Enumere cinco aplicaciones de Internet no propietarias y los protocolos de la capa de aplicación que utilizan.**

La Web: HTTP;

Transferencia de archivos: FTP;

Login remoto: Telnet;

Network News: NNTP;

e-mail: SMTP.

**R.2**. **¿Cuál es la diferencia entre la arquitectura de red y la arquitectura de aplicación?**

La arquitectura de red refiere a la organización del proceso de comunicación en capas (por ejemplo, la arquitectura de Internet de cinco capas).

La arquitectura de la aplicación, por otra parte, está diseñada por el desarrollador de aplicaciones, el cual dicta la estructura general de la aplicación (por ejemplo, cliente-servidor o P2P)

**R.3. En una sesión de comunicación entre dos procesos, ¿qué proceso es el cliente y qué proceso es el servidor?**

El proceso que se inicia la comunicación es el cliente, el proceso que espera a ser contactado es el servidor.

**R.4. En una aplicación de compartición de archivos P2P, ¿está de acuerdo con la siguiente afirmación: “No existen los lados de cliente y de servidor en una sesión de comunicación”? ¿Por qué?**

No. Como se indica en el texto, todas las sesiones de comunicación tienen un lado cliente y un servidor. En un programa P2P para compartir archivos, el par que está recibiendo un archivo suele ser el cliente y el par que está enviando el archivo suele ser el servidor.

**R.5. ¿Qué información utiliza un proceso que se ejecuta en un host para identificar a un proceso que se ejecuta en otro host?**

La dirección IP del host de destino y el número de puerto del socket de destino.

**R.6. Suponga que desea realizar una transición <**transacción debería ser**> desde un cliente remoto a un servidor lo más rápidamente posible. ¿Qué utilizaría, UDP o TCP? ¿Por qué?**

Utilizaría UDP. Con UDP, la transacción se puede completar en un tiempo de ida y vuelta (RTT) - el cliente envía la solicitud de transacción en un socket UDP, y el servidor envía la respuesta de vuelta al socket UDP del cliente.

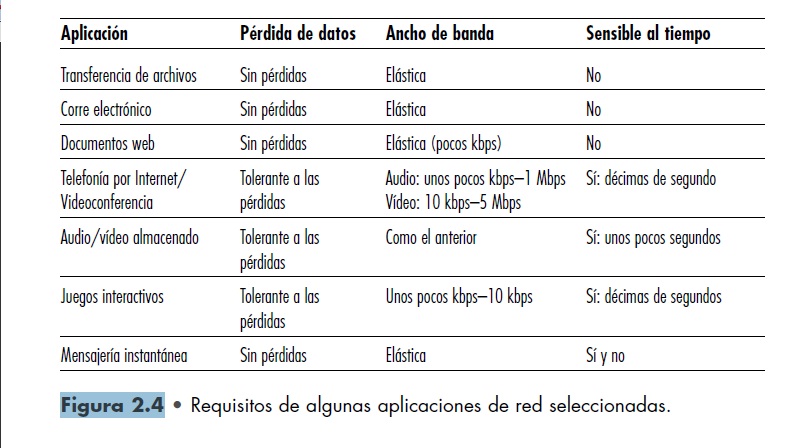
Con TCP, un mínimo de dos RTT son necesarios – uno para establecer la conexión TCP, y otro para que el cliente envíe la solicitud, y para que el servidor envíe de regreso la respuesta. **<**continua …**>**

<Agrego>

RTT Round-trip delay time

Tiempo que tarda un paquete enviado desde un emisor en volver a este mismo emisor habiendo pasado por el receptor de destino.

**R.7. Utilizando la Figura 2.4, podemos ver que ninguna de las aplicaciones indicadas en dicha figura presenta a la vez requisitos de temporización y de ausencia de pérdida de datos. ¿Puede concebir una aplicación que requiera que no haya pérdida de datos y que también sea extremadamente sensible al tiempo?**

****

No hay buenos ejemplos de una aplicación que requiera la ausencia de pérdida de datos y temporización.

**R.8. Enumere las cuatro clases principales de servicios que puede proporcionar un protocolo de transporte. Para cada una de las clases de servicios, indique si UDP o TCP (o ambos) proporcionan un servicio así.**

a) Transferencia de datos fiable

TCP proporciona un flujo de bytes fiable entre el cliente y el servidor, pero UDP no lo hace.

b) Garantía de que un determinado valor para el rendimiento se mantendrá

Ninguno

c) La garantía de que los datos se entregarán dentro de un período especificado de tiempo

Ninguno

d) Seguridad

Ninguno

**R.9. Recuerde que TCP puede mejorarse con SSL para proporcionar servicios de seguridad proceso a proceso, incluyendo mecanismos de cifrado. ¿En qué capa opera SSL, en la capa de transporte o en la capa de aplicación? Si el desarrollador de la aplicación desea mejorar TCP con SSL, ¿qué tendrá que hacer?**

SSL opera en la capa de aplicación.

El socket SSL toma los datos no cifrados de la capa de aplicación, los encripta y después los pasa al socket TCP.

Si el desarrollador de aplicaciones TCP quiere mejorar su aplicación con SSL, debe incluir el código SSL en la misma.

**SECCIONES 2.2–2.5**

**R.10. ¿Qué quiere decir el término protocolo de acuerdo?**

Un protocolo utiliza handshaking si las dos entidades comunicantes primero intercambian los paquetes de control antes de enviar datos la una a la otra. SMTP utiliza handshaking en la capa de aplicación, mientras que HTTP no.

**R.11. ¿Por qué HTTP, FTP, SMTP y POP3 se ejecutan sobre TCP en lugar de sobre UDP?**

Las aplicaciones asociadas a estos protocolos requieren que todos los datos de la aplicación se reciben en el orden correcto y sin huecos. TCP proporciona este servicio, mientras que UDP no.

**R.12. Un sitio de comercio electrónico desea mantener un registro de compras para cada uno de sus clientes. Describa cómo se puede hacer esto utilizando cookies.**

Cuando el usuario visita el sitio por primera vez, el sitio devuelve un número de cookie. Este número de cookie se almacena en el host del usuario y está gestionado por el navegador. Durante cada visita posterior, el navegador envía el número de cookie de vuelta al sitio. Así, el sitio sabe cuando el usuario (más precisamente, este navegador) está de visita en el sitio.

**R.13. Describa cómo el almacenamiento en caché web puede reducir el retardo de recepción de un objeto solicitado. ¿Reducirá este tipo de almacenamiento el retardo de todos los objetos solicitados por el usuario o sólo el de algunos objetos? ¿Por qué?**

El almacenamiento en caché web puede llevar el contenido deseado "más cerca" al usuario, tal vez a la misma LAN a la que está conectado el host del usuario. Puede reducir el retardo para todos los objetos, incluso objetos que no están en caché, ya que el almacenamiento en caché reduce el tráfico en los enlaces. Causas para que objetos no sean cacheados son por ejemplo las normas administrativas para el tamaño máximo de objeto cacheable o cantidad máxima de objetos cacheables.

**R.14. Establezca una sesión Telnet en un servidor web y envíe un mensaje de solicitud de varias líneas. Incluya en dicho mensaje la línea de cabecera If-modified-since: para forzar un mensaje de respuesta con el código de estado 304 Not Modified.**

Se debe emitir el siguiente comando seguido por el mensaje GET de HTTP al servidor web "utopia.poly.edu".

Las primeras 4 líneas son el mensaje GET y las línea de la cabecera ingresadas por el usuario.

> telnet utopia.poly.edu 80

GET /index.html HTTP/1.1

Host: utopia.poly.edu

If-modified-since Fri, 18 May 2007 09:23:24 GMT

Si la página index.html en este servidor Web no se ha modificado desde viernes 18 May 2007 09:23:34 GMT, la siguiente salida se muestra cuando los comandos anteriores se ejecutaron el Lunes, 19 de mayo de 2007.

HTTP/1.1 304 Not Modified

Date: Mon, 21 May 2007 15:20:05 GMT

ETag: “1631-3a3-3e6d478b”

**R.15. ¿Por qué se dice que FTP envía la información de control “fuera de banda”?**

FTP utiliza dos conexiones TCP paralelas, una conexión para el envío de información de control (tal como una solicitud para transferir un archivo) y otra conexión para realmente transferir el archivo. Debido a que la información de control no se envía a través de la misma conexión que el archivo, se dice que FTP envía información de control fuera de banda

**R.16. Suponga que Alicia, que dispone de una cuenta de correo electrónico web (como por ejemplo Hotmail o gmail), envía un mensaje a Benito, que accede a su correo almacenado en su servidor de correo utilizando POP3. Explique cómo se transmite el mensaje desde el host de Alicia hasta el de Benito. Asegúrese de citar la serie de protocolos de la capa de aplicación que se utilizan para llevar el mensaje de un host al otro.**

El mensaje se envía desde el host de Alicia hacia su servidor de correo a través de HTTP. El servidor de correo de Alicia envía el mensaje al servidor de correo de Benito a través de SMTP. Luego el servidor de correo Benito transfiere el mensaje hacia su host a través de POP3.

**R.17. Imprima la cabecera de un mensaje de correo electrónico que haya recibido recientemente.**

**¿Cuántas líneas de cabecera Received: contiene? Analice cada una de las líneas de cabecera del mensaje.**

|  |  |
| --- | --- |
| Received: | from 65.54.246.203 (EHLO bay0-omc3-s3.bay0.hotmail.com) (65.54.246.203) by mta419.mail.mud.yahoo.com with SMTP; Sat, 19 May 2007 16:53:51 -0700 |
| Received: | from hotmail.com ([65.55.135.106]) by bay0-omc3-s3.bay0.hotmail.com with Microsoft SMTPSVC(6.0.3790.2668); Sat, 19 May 2007 16:52:42 -0700 |
| Received: | from mail pickup service by hotmail.com with Microsoft SMTPSVC; Sat, 19 May 2007 16:52:41 -0700 |
| Message-ID: | <BAY130-F26D9E35BF59E0D18A819AFB9310@phx.gbl> |
| Received: | from 65.55.135.123 by by130fd.bay130.hotmail.msn.com with HTTP; Sat, 19 May 2007 23:52:36 GMT |
| From: | "prithula dhungel" <prithuladhungel@hotmail.com> |
| To: | prithula@yahoo.com |
| Bcc: | |
| Subject: | Test mail |
| Date: | Sat, 19 May 2007 23:52:36 +0000 |
| Mime-Version: | 1.0 |
| Content-Type: | Text/html; format=flowed |
| Return-Path: | prithuladhungel@hotmail.com |

Received: Este campo de encabezado indica la secuencia en la que los servidores SMTP envían y reciben el mensaje de correo incluyendo las marcas de tiempo correspondientes.

En este ejemplo hay 4 "Received:" líneas de cabecera. Esto significa que el mensaje de correo electrónico pasa a través de 5 diferentes servidores SMTP antes de ser entregado al buzón de correo del receptor. El último (cuarto) "Received:" el encabezado indica el flujo de mensajes de correo electrónico desde el servidor SMTP del remitente al segundo servidor SMTP en la cadena de servidores. El servidor SMTP del remitente está en la dirección 65.55.135.123 y el segundo servidor SMTP en la cadena es by130fd.bay130.hotmail.msn.com.

El tercer " Received:" la cabecera indica el flujo de mensajes de correo electrónico desde el segundo servidor SMTP en la cadena al tercer servidor, y así sucesivamente.

Finalmente, el primer " Received:" la cabecera indica el flujo del mensaje electrónico desde el cuarto servidor SMTP hasta el último servidor SMTP en la cadena (es decir, el servidor de correo del receptor).

Message -id: El mensaje esta dado por este numero BAY130-F26D9E35BF59E0D18A819AFB9310@phx.gbl (por bay0-omc3-s3.bay0.hotmail.com. Mensaje-id es una cadena única asignada por el sistema de correo cuando el mensaje es creado.

From: Indica la dirección de correo electrónico del remitente del correo. En el ejemplo dado, el remitente es [prithuladhungel@hotmail.com](mailto:prithuladhungel@hotmail.com)

To: Este campo indica la dirección de correo electrónico del receptor del correo. En el ejemplo, el receptor está "prithula@yahoo.com"

Subject: Indica el tema del correo electrónico (si la hay especificado por el remitente). En el ejemplo, el tema especificado por el remitente es "Test mail"

Date: La fecha y hora en que el mensaje fue enviado por el remitente. En el ejemplo, el remitente envió el correo el 19 de mayo de 2007, a las 23:52:36 GMT.

MIME-Version: versión MIME utilizada para el correo. En el ejemplo, que es de 1,0.

Content-Type: El tipo de contenido en el cuerpo del mensaje de correo. En el ejemplo, se trata de "text / html".

Return-Path: Especifica la dirección de correo a la que se enviara el correo si el receptor de este mail desea responder al remitente. Es también utilizado por el servidor de correo del remitente al momento de rebotar mensajes que no se pueden entregar, los cuales son enviados al Return-Path como mensajes de error de mailer-daemon. En el ejemplo, el Return-Path es "prithuladhungel@hotmail.com".

**R.18. Desde la perspectiva de un usuario, ¿cuál es la diferencia entre el modo “descargar y borrar” y el modo “descargar y mantener” en POP3?**

Con descargar y borrar, luego que un usuario obtiene sus mensajes de un servidor POP, los mensajes se eliminan. Esto plantea un problema para el usuario nómada, que quiere acceder a los mensajes desde diferentes máquinas (PC de oficina, PC, etc.) En la configuración descargar y mantener, los mensajes no se eliminan cuando el usuario los obtiene. Esto también puede ser un inconveniente, ya que cada vez que el usuario obtiene los mensajes almacenados desde una nueva máquina, todos los mensajes que no son eliminadas se transfieren a la nueva máquina (incluyendo mensajes muy viejos).

**R.19. ¿Pueden el servidor web y el servidor de correo electrónico de una organización tener exactamente el mismo alias para un nombre de host (por ejemplo, foo.com)? ¿Cuál sería el tipo especificado en el registro de recurso (RR) que contiene el nombre de host del servidor de correo?**

Si el servidor de correo y el servidor Web de una organización pueden tener el mismo alias para un nombre de host. El registro MX se utiliza para asociar el nombre de host del servidor de correo a su dirección IP.

<Agrego>

Un registro MX o Mail eXchange record (registro de intercambio de correo) es un tipo de registro, un recurso DNS que especifica cómo debe ser encaminado un correo electrónico en internet. Los registros MX apuntan a los servidores a los cuales envían un correo electrónico, y a cuál de ellos debería ser enviado en primer lugar, por prioridad.

**SECCIÓN 2.6**

**R.20. En BitTorrent, suponga que Alicia proporciona fragmentos a Benito a intervalos de 30 segundos. ¿Devolverá necesariamente Benito el favor y proporcionará fragmentos a Alicia en el mismo intervalo de tiempo? ¿Por qué?**

No necesariamente Benito proporcionará también fragmentos a Alicia. Alicia debe estar en los 4 primeros vecinos de Benito para que él le envíe fragmentos a ella, lo que puede no ocurrir aun cuando Alicia le proporcione fragmentos a Benito a intervalos de 30 segundos.

**R.21. Suponga que un nuevo par Alicia se une a BitTorrent sin tener en su posesión ningún fragmento. Dado que no posee fragmentos, no puede convertirse en uno de los cuatro principales suministradores de ninguno de los otros pares, ya que no tiene nada que suministrar. ¿Cómo obtendrá entonces Alicia su primer fragmento?**

Alicia obtendrá su primer fragmento a causa de ser seleccionada por uno de sus vecinos como resultado de un "optimistic unchoke", para enviarle fragmentos.

**R.22. ¿Qué es una red solapada? ¿Contiene routers? ¿Cuáles son las fronteras en una red solapada? ¿Cómo se crea y se mantiene la red solapada que se encarga de distribuir las consultas?**

Una red solapada en un sistema de intercambio de archivos P2P consiste en los nodos que participan en el sistema de intercambio de archivos y los vínculos lógicos entre los nodos. Existe una relación lógica (un "borde" en términos de teoría de grafos) del nodo A al nodo B si existe una conexión semi-permanente TCP entre A y B. Una red solapada no incluye routers. Con Gnutella, cuando un nodo quiere unirse a la red Gnutella, este descubre primero ("fuera de banda") la dirección IP de uno o más nodos que ya están en la red. A continuación, les envía mensajes a estos nodos para unirse a la red. Cuando el nodo recibe confirmaciones, se convierte en un miembro de la red Gnutella. Los nodos mantienen sus vínculos lógicos con mensajes periódicos de actualización.

**R.23. ¿En qué sentido la mensajería instantánea con un índice centralizado es un híbrido de las arquitecturas cliente-servidor y P2P?**

Se trata de un híbrido de cliente-servidor y arquitecturas P2P debido a que:

-Hay un componente centralizado (el índice) como en el caso de un sistema cliente-servidor.

-Otras funciones (excepto la indexación) no utilizan ningún tipo de servidor central. Esto es similar a lo que existe en un sistema P2P.

**R.24. Considere una DHT con una topología de red solapada en malla (es decir, cada par controla a todos los demás pares del sistema). ¿Cuáles son las ventajas y desventajas de un diseño de este tipo? ¿Cuáles son las ventajas y desventajas de una DHT circular (sin atajos)?**

Malla DHT: La ventaja es para encaminar un mensaje al par más cercano a la Key, sólo se requiere un salto, la desventaja es que cada par debe rastrear todos los otros pares en el en la DHT.

Circular DHT: la ventaja es que cada par tiene que rastrear sólo unos pocos otros pares, la desventaja es que O(n) saltos son necesarios para enrutar un mensaje a un compañero responsable de la Key.

**R.25. Skype utiliza técnicas P2P para dos funciones importantes. ¿Cuáles son dichas funciones?**

a) Ubicación del Usuario

b) NAT transversal

**R.26. Cite al menos cuatro aplicaciones distintas que se adapten de forma natural a las arquitecturas P2P. (*Sugerencia:* la distribución de archivos y la mensajería instantánea son dos de ellas.)**

a) Distribución de archivos

b) Mensajería instantánea

c) Video Streaming

d) Computación Distribuida

**SECCIONES 2.7–2.8**

**R.27. El servidor UDP descrito en la Sección 2.8 sólo necesitaba un socket, mientras que el servidor TCP descrito en la Sección 2.7 necesitaba dos. ¿Por qué? Si el servidor TCP tuviera que soportar *n* conexiones simultáneas, cada una procedente de un host cliente distinto, ¿cuántos sockets necesitaría el servidor TCP?**

Con el servidor UDP, no hay ningún socket de bienvenida, y todos los datos provenientes de los diferentes clientes entran al servidor a través de este único socket.

Con el servidor TCP, hay un socket de bienvenida, y cada vez que un cliente inicia una conexión con el servidor, un nuevo socket se crea. Por lo tanto, para soportar n conexiones simultáneas, el servidor necesitara n+1 sockets.

**R.28. En la aplicación cliente-servidor sobre TCP descrita en la Sección 2.7, ¿por qué tiene que ser ejecutado el programa servidor antes que el programa cliente? En la aplicación cliente-servidor sobre UDP descrita en la Sección 2.8, ¿por qué el programa cliente puede ejecutarse antes que el programa servidor?**

Para la aplicación TCP, tan pronto como el cliente se ejecuta, intenta iniciar una conexión TCP con el servidor. Si el servidor TCP no se está ejecutando, el cliente no podrá establecer una conexión.

Para la aplicación UDP, el cliente no inicia conexiones (o intenta comunicarse con el servidor UDP) inmediatamente después de su ejecución.