

Guia 1: Nivel de Aplicación

Redes de Comunicaciones (TB067) - 2C2024 - FIUBA

Martin Klöckner - mklockner@fi.uba.ar

1. Para una sesión de comunicación entre un par de procesos, ¿qué proceso es el cliente y cuál es el servidor?

El proceso cliente es aquel que inicia la comunicación entre un par de procesos, mientras que el proceso servidor es aquel que espera a que otro proceso inicie la comunicación.

2. Para una aplicación de intercambio de archivos P2P, ¿está de acuerdo con la afirmación: “No existe la noción de los lados cliente y servidor de una sesión de comunicación”? ¿Por qué o por qué no?

No sé está de acuerdo, ya que existe la noción de cliente y servidor en una sesión de transferencia de archivos P2P, solo que cualquier proceso puede ser servidor o cliente, es decir, cualquier proceso puede iniciar la comunicación con otro, o esperar a que otro proceso inicie la comunicación.

3. ¿Qué es un “socket”?

Un “socket” es un conjunto de datos que permite la comunicación entre dos procesos. Cuando se establece una conexión entre dos procesos, cada proceso debe asignar un socket a esa comunicación.

En el modelo TCP/IP, se habla de un socket de internet, el cual permite la comunicación entre dos procesos, por lo general pertenecientes a dos computadoras distintas. Los sockets de internet se identifican por su número de socket, el cual se crea a partir de el protocolo de transporte, la dirección y el número de puerto.

4. Mencione una aplicación que requiera que no haya pérdida de datos y que también sea extremadamente sensible al tiempo.

Un ejemplo puede ser el protocolo SMTP utilizado para la comunicación de correos electrónicos, en este caso es extremadamente importante que no ocurra pérdida de datos, ya que pueden transportar información sensible y/o importante; en cuanto al tiempo no es tan importante ya los usuarios pueden permitirse que se demore unos segundos.

Existen diversos ejemplos siendo la mayoría correspondiente a servicios interactivos en tiempo real, como la telefonía por internet (VoIP), las teleconferencias, o los juegos multijugador.

5. ¿Cuáles son algunas diferencias entre TCP y UDP?

La principal diferencia entre TCP y UDP es que TCP es más confiable ya que es mas robusto debido a varios mecanismos que lo diferencian de UDP y que lo hacen mas seguro, por ejemplo el proceso de 3 pasos que se utiliza en TCP para iniciar una sesión, o el procedimiento

6. ¿Por qué TCP y UDP no tienen mecanismos de cifrado?

Porque son protocolos relativamente viejos, los cuales en un principio no estaban pensados en términos de seguridad.

7. ¿Por qué HTTP, SMTP e IMAP se ejecutan sobre TCP en lugar de UDP?

Porque son protocolos que requieren que no haya perdida de datos. En el caso de SMTP o IMAP que se utilizan para la transmisión de correo electrónicos, la perdida de datos implicaría perdida de la información, la cual puede ser importante.

8. ¿Cuál es la diferencia entre una conexión HTTP persistente y una conexión no persistente?

La diferencia radica en que la conexión HTTP persistente una vez finalizada la transferencia de datos continua esperando por datos del usuario hasta que se termine el tiempo de conexión, mientras que la conexión HTTP no persistente termina una vez finalizada la transferencia.

9. ¿El almacenamiento en caché web reducirá la demora para todos los objetos solicitados por un usuario o solo para algunos de los objetos? ¿Por qué? ¿En qué casos un almacenamiento en caché web no mejora el tiempo de respuesta?

El almacenamiento en caché web (también llamado servidor proxy) siempre reduce la demora en la carga de archivos, ya que por lo general están mas cerca de los usuarios; en tal caso se evita la transferencia desde el servidor, que por lo general suele estar mas lejos al cliente.

Puede que la copia de datos almacenada en el servidor caché sea obsoleta con respecto a los datos del servidor, en ese caso el servidor caché tendrá que obtener los datos desde el servidor, y en ese caso existe una demora mayor que si no hubiera servidor caché.

10. La siguiente cadena de caracteres ASCII ha sido capturada por Wireshark cuando el navegador enviaba un mensaje GET HTTP. Responda a las siguientes cuestiones, indicando en qué parte del siguiente mensaje GET HTTP se encuentra la respuesta.

```
GET /cs453/index.html HTTP/1.1<cr><lf>
Host: gaia.cs.umass.edu<cr><lf>
User-Agent: Mozilla/5.0 (Windows;U; Windows NT 5.1; en-US; rv:1.7.2)
Gecko/20040804 Netscape/7.2 (ax)<cr><lf>
Accept:ext/xml, application/xml, application/xhtml+xml, text/html;q=0.9,
text/plain;q=0.8, > image/png,*/*;q=0.5<cr><lf>
```

```
Accept-Language: en-us,en;q=0.5<cr><lf>
Accept-Encoding: zip,deflate<cr><lf>
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7<cr><lf>
Keep-Alive: 300<cr><lf>
Connection:keep-alive<cr><lf><cr><lf>
```

- a. ¿Cuál es el URL del documento solicitado por el navegador?
- b. ¿Qué versión de HTTP se está ejecutando en el navegador?
- c. ¿Qué tipo de conexión solicita el navegador, persistente o no persistente?
- d. ¿Cuál es la dirección IP del host en el que se está ejecutando el navegador?
- e. ¿Qué tipo de navegador inicia este mensaje? ¿Por qué es necesario indicar el tipo de navegador en un mensaje de solicitud HTTP?

- a. La URL del documento solicitado es `www.gaia.cs.umass.edu/cs453/index.html` ya que se indica en la sección del encabezado `Host: .`
- b. El navegador está ejecutando la versión `1.1` de HTTP, esto se puede ver en la sección `GET /cs453/index.html HTTP/1.1`.
- c. El navegador solicita una conexión de tipo persistente, la cual es la acción por defecto del protocolo HTTP. Esto se puede ver en la última línea del encabezado la cual indica: `Connection:keep-alive`.
- d. La dirección IP del host en que se está ejecutando el navegador no se puede saber ya que no se indica en el encabezado HTTP.
- e. El tipo de navegador se puede ver en la parte `User-Agent` del encabezado HTTP, en este caso es el navegador Netscape versión 7.2 de escritorio, en particular corriendo sobre el sistema operativo Windows NT 5.1; `Mozilla/5.0` indica que es compatible con ese navegador, y se incluye por razones históricas.

11. El siguiente texto muestra la respuesta devuelta por el servidor al mensaje de solicitud GET HTTP del problema anterior. Responda a las siguientes cuestiones, indicando en qué parte del siguiente mensaje se encuentran las respuestas.

```
HTTP/1.1 200 OK<cr><lf>Date: Tue, 07 Mar 2008 12:39:45GMT<cr><lf>Server:
Apache/2.0.52 (Fedora) <cr><lf>Last-Modified: Sat, 10 Dec2005 18:27:46
GMT<cr><lf>ETag: "526c3-f22-a88a4c80"<cr><lf>Accept- Ranges:
bytes<cr><lf>Content-Length: 3874<cr><lf> Keep-Alive:
timeout=max=100<cr><lf>Connection: Keep-Alive<cr><lf>Content-Type:
text/html; charset= ISO-8859-1<cr><lf><cr><lf><!doctype html public "-
//w3c//dtd html 4.0transitional//en"><lf><html><lf> <head><lf> <meta
http-equiv="Content-Type" content="text/html; charset=iso-8859-1"><lf>
<meta name="GENERATOR" content="Mozilla/4.79 [en] (Windows NT 5.0; U
Netscape]"><lf> <title>CMPSCI 453 / 591 / NTU-ST550ASpring 2005
homepage</title><lf></head><lf> <aquí continúa el texto del documento (no
mostrado)>
```

- a. ¿Ha podido el servidor encontrar el documento? ¿En qué momento se suministró la respuesta con el documento?
- b. ¿Cuándo fue modificado por última vez el documento?
- c. ¿Cuántos bytes contiene el documento devuelto?
- d. ¿Cuáles son los primeros cinco bytes del documento que se está devolviendo?
- e. ¿Ha acordado el servidor emplear una conexión persistente?

- a. Si lo ha podido encontrar ya que el código de respuesta es `200 OK`, el momento en que se suministra la respuesta fue en la fecha `Tue, 07 Mar 2008 12:39:45 GMT`.
- b. El documento fue modificado por última vez en la fecha `Sat, 10 Dec 2005 18:27:46 GMT`.
- c. El documento devuelto contiene `3874` bytes, esto se puede ver en la etiqueta `Length` del encabezado.
- d. Los primeros 5 bytes son `<!doc`, la secuencia `<cr><lf><cr><lf>` indica el término del encabezado HTTP y luego comienza el documento devuelto (recordemos que cada carácter ocupa 1 byte).
- e. Si, se puede ver en la etiqueta `Connection:` del encabezado HTTP, la cual indica `Keep-Alive`.

12. Haga Telnet a un servidor web y envíe un mensaje de solicitud multilínea. Incluya en el mensaje de solicitud la línea de encabezado `If-modified-since:` para forzar un mensaje de respuesta con el código de estado 304 No modificado.