Práctica 2

Capa de Aplicación - HTTP

2. ¿Cuál es la función de la capa de aplicación?

El **nivel de aplicación** o **capa de aplicación** es el séptimo nivel del modelo OSI y el cuarto de la pila TCP. Ofrece a las aplicaciones (de usuario o no) la posibilidad de acceder a los servicios de las demás capas y define los protocolos que utilizan las aplicaciones para intercambiar datos, como correo electrónico (POP y SMTP), gestores de bases de datos y protocolos de transferencia de archivos (FTP).

Capa de aplicación

La **capa de aplicación** define las aplicaciones de red y los servicios de Internet estándar que puede utilizar un usuario. Estos servicios utilizan la capa de transporte para enviar y recibir datos. Existen varios protocolos de capa de aplicación. En la lista siguiente se incluyen ejemplos de protocolos de capa de aplicación:

- Servicios TCP/IP estándar como los comandos `ftp`, `tftp` y `telnet`.
- Comandos UNIX "r", como `rlogin` o `rsh`.
- Servicios de nombres, como NIS o el sistema de nombre de dominio (DNS).
- Servicios de directorio (LDAP).
- Servicios de archivos, como el servicio NFS.
- Protocolo simple de administración de red (SNMP), que permite administrar la red.
- Protocolo RDISC (Router Discovery Server) y protocolos RIP (Routing Information Protocol).

3) ¿Cómo podrían comunicarse dos procesos si están en diferentes máquinas?
Cuando un programador desarrolla una aplicación de red, este requerirá que haya una comunicación entre los sistemas terminales que ejecutan el programa, ya sea para compartir datos o archivos que servirán para cumplir el propósito de la aplicación. Para esto, un proceso envía mensajes a la red y los recibe de ellas a través de una interfaz de software llamada socket. Cuando un proceso desea enviar un mensaje a otro proceso que se está ejecutando en otro host, envía el mensaje a través de la puerta.
Este proceso emisor supone que existe una infraestructura de transporte al otro lado de la puerta que llevará el mensaje hasta la puerta del proceso destino. Una vez que el mensaje llega al host de destino, este pasa a través de la puerta del proceso receptor, actuando entonces el proceso receptor sobre el mensaje.
Especificar el orden y contenidos de los mensajes enviados dependerá del protocolo utilizado por las aplicaciones.

3. Si dos procesos deben comunicarse:
a. ¿Cómo podrían hacerlo si están en diferentes máquinas?
Cuando un programador desarrolla una aplicación de red, este requerirá que haya una comunicación entre los sistemas terminales que ejecutan el programa, ya sea para compartir datos o archivos que

servirán para cumplir el propósito de la aplicación. Para esto, un proceso envía mensajes a la red y los recibe de ellas a través de una interfaz de software llamada socket. Cuando un proceso desea enviar un mensaje a otro proceso que se está ejecutando en otro host, envía el mensaje a través de la puerta.
Este proceso emisor supone que existe una infraestructura de transporte al otro lado de la puerta que llevará el mensaje hasta la puerta del proceso destino. Una vez que el mensaje llega al host de destino, este pasa a través de la puerta del proceso receptor, actuando entonces el proceso receptor sobre el mensaje.
Especificar el orden y contenidos de los mensajes enviados dependerá del protocolo utilizado por las aplicaciones.

4. Explique brevemente cómo es el modelo Cliente/Servidor. De un ejemplo de un sistema Cliente/Servidor en la “vida cotidiana” y un ejemplo de un sistema informático que siga el modelo Cliente/Servidor. ¿Conoce algún otro modelo de comunicación?

En una arquitectura cliente-servidor siempre existe un host activo, denominado servidor, que da servicio a las solicitudes de muchos otros hosts, que son los clientes.
Un ejemplo clásico es la web, en la que un servidor siempre activo sirve las solicitudes de los navegadores que se ejecutan en los hosts clientes.Otro paradigma arquitectónico predominante en Internet es el P2P. En una arquitectura P2P existe una mínima (o ninguna) dependencia de una infraestructura de servidores siempre activos. Los pares no son propiedad del proveedor del servicio. Una de las características más convincentes de las arquitecturas P2P es su autoescalabilidad. Cada peer genera una carga de trabajo solicitando archivos, también añade capacidad de servicio al sistema (por ejemplo distribuyendo archivos a otros pares). Por lo general, no requieren una infraestructura de servidores. Ejemplos: Utorrent, Skype.

5. Describa la funcionalidad de la entidad genérica “Agente de usuario” o “User agent”.

Un agente de usuario es cualquier conexión que se realice con éxito en el servidor. Más frecuentemente, se utiliza para referirse a los motores de búsqueda de páginas Web, aunque puede darse también en otros contextos. Cada vez que el rastreador de un buscador o cualquier otro robot se conecta a tu servidor, deja también una identidad agente de usuario.

Es importante conocer este tipo de conceptos porque pueden darte mucha información acerca de los visitantes de tu página y qué tipo de máquinas están empleando. Un agente de usuario estándar incluye: la marca y versión del buscador, marca del sistema operativo y versión, nombre de la plataforma, máquina o tipo de procesador e idioma. Algunos navegadores permiten a los usuarios especificar qué información será enviada o prohibir el envío de cualquiera de estos datos.

Un agente de usuario es una aplicación informática que funciona como cliente en un protocolo de red; el nombre se aplica generalmente para referirse a aquellas aplicaciones que acceden a la World Wide Web. Los agentes de usuario que se conectan a la Web pueden ser desde navegadores web hasta los web crawler de los buscadores, pasando por teléfonos móviles, lectores de pantalla y navegadores en Braille usados por personas con discapacidades.
Cuando un usuario accede a una página web, la aplicación generalmente envía una cadena de texto que identifica al agente de usuario ante el servidor. Este texto forma parte del pedido a través de HTTP, llevando como prefijo User-agent: o User-Agent: y generalmente incluye información como el nombre de la aplicación, la versión, el sistema operativo, y el idioma.
Los bots, como los web crawlers, a veces incluyen también una URL o una dirección de

correo electrónico para que el administrador del sitio web pueda contactarse con el operador del mismo.
La cadena user-agent se empleó al principio de la Web para distinguir los navegadores entre sí y ofrecer un contenido diferente a cada uno.

6. ¿Qué son y en qué se diferencian HTML y HTTP?

HTML es un lenguaje de maquetado que utiliza etiquetas. Suele utilizarse para la vista en aplicaciones web estáticas o dinámicas junto con CSS. En cambio, HTTP es un protocolo de la capa de aplicación de Internet, que utiliza la web para recuperar documentos. Este protocolo define el formato y la secuencia de los mensajes que pasan entre el navegador web y el servidor web.

7. Utilizando la VM, abra una terminal. Investigue sobre el comando curl y analice para qué sirven los siguientes parámetros (-I, -H, -X, -s).

-I, --head

(HTTP FTP FILE) Fetch the headers only! HTTP-servers feature the command HEAD which this uses to get nothing but the header of a document. When used on an FTP or FILE file, curl displays the file size and last modification time only.

-H, --header <header/@file>

(HTTP) Extra header to include in the request when sending HTTP to a server. You may specify any number of extra headers. Note that if you should add a custom header that has the same name as one of the internal ones curl would use, your externally set header will be used instead of the internal one. This allows you to make even trickier stuff than curl would normally do. You should not replace internally set headers without knowing perfectly well what you're doing. Remove an internal header by giving a replacement without content on the right side of the colon, as in: -H "Host:". If you send the custom header with no-value then its header must be terminated with a semicolon, such as -H "X-Custom-Header;" to send "X-Custom-Header:".

curl -H "X-First-Name: Joe" http://example.com/

-X, --request <command>

(HTTP) Specifies a custom request method to use when communicating with the HTTP server. The specified request method will be used instead of the method otherwise used (which defaults to GET). Read the HTTP 1.1 specification for details and explanations. Common additional HTTP requests include PUT and DELETE, but related technologies like WebDAV offers PROPFIND, COPY, MOVE and more.

Normally you don't need this option. All sorts of GET, HEAD, POST and PUT requests are rather invoked by using dedicated command line options.

This option only changes the actual word used in the HTTP request, it does not alter the way curl behaves.

-s, --silent

Silent or quiet mode. Don't show progress meter or error messages. Makes Curl mute. It will still output the data you ask for, potentially even to the terminal/stdout unless you redirect it.

Use -S, --show-error in addition to this option to disable progress meter but still show error messages.

8. Ejecute el comando curl sin ningún parámetro adicional y acceda a www.redes.unlp.edu.ar. Luego responda:

a. ¿Cuántos requerimientos realizó y qué recibió? Pruebe redirigiendo la salida(>) del comando curl a un archivo con extensión html y abrirlo con un navegador.

Realizo un requerimiento y fue entregado el codigo html de la pagina solicitada.

b. ¿Cómo funcionan los atributos href de los tags link e img en html?

La etiqueta <link>, permite realizar la carga y establecer la relación existente entre el documento web actual y un recurso externo. Esta etiqueta es normalmente muy utilizada para realizar la carga de hojas de estilo para el documento web.

Href: este atributo permite establecer la URL en la que se encuentra el documento el cual se quiere enlazar, con el documento web.

c. Para visualizar la página completa con imágenes como en un navegador, ¿alcanza con realizar un único requerimiento? ¿Cuántos requerimientos serían necesarios para obtener una página que tiene dos CSS, dos Javascript y tres imágenes? Diferencie como funcionaría un navegador respecto al comando curl ejecutado previamente.

Se necesitarían 8 requerimientos, uno para el html, y uno para cada archivo que necesite conseguir para mostrar en la pagina. 1 HTML + 2 CSS + 2 Javascript + 3 imágenes=8

9. Ejecute a continuación los siguientes comandos:

curl -v -s www.redes.unlp.edu.ar > /dev/null
curl -I -v -s www.redes.unlp.edu.ar

-¿Qué diferencias nota entre cada uno?

Ambos muestran la misma informacion, sin embargo el primero trae todo el contenido de la pagina y lo redirecciona a /dev/null (por lo que no se muestra en la salida estandar), mientras que el segundo usa el argumento -I para utilizar el metodo HEAD de HTTP y solicitar solo las cabeceras de la pagina.
Ademas con -v se muestra mas informacion en la salida estandar, pero como en el primer caso la redireccion envia toda la salida estandar a otro archivo, en el segundo caso se muestra cierta informacion repetida por el -v que si se muestra.
El primero muestra el mensaje [data not show]

-¿Qué ocurre si en el primer comando quita la redirección a /dev/null? ¿Por qué no es necesaria en el segundo comando?

Mostraria todo el html y las cabeceras. No es necesario ya que solo se solicitan los headers de la pagina.

-¿Cuántas cabeceras viajaron en el requerimiento? ¿Y en la respuesta?

En el requerimiento se enviaron 3 cabeceras, mientras que en la respuesta viajaron 7.

10. Ejecute una vez más el comando curl www.redes.unlp.edu.ar pero sólo muestre los encabezados y luego responda:

a. ¿Es posible determinar qué servidor web se utiliza para servir la página?
Si, se indica en la cabecera Server: Apache/2.4.7 (Ubuntu)

b. ¿Cuál es el código de respuesta que devolvió el servidor? ¿Qué otros códigos existen y qué significan? Investigue genéricamente los tipos de error 2XX, 3XX, 4XX y 5XX.
Devolvió el código 200Ok

1XX Respuestas informativas

2XX Peticiones correctas

3XX Redirecciones

4XX Errores del cliente

5XX Errores de servidor

c. ¿Cuándo fue la última vez que se modificó la página?

Miercoles, 16 de marzo de 2016 a las 20:41:34

d. Solicite la página nuevamente con curl usando GET, pero esta vez indique que quiere obtenerla sólo si la misma fue modificada en una fecha posterior a la que efectivamente fue modificada. ¿Cómo lo hace? ¿Qué resultado obtuvo? ¿Puede explicar por qué y para qué sirve?

Curl -H “If-Modified-Since: <day-name>, <day> <month> <year> <hour>:<minute>:<second> GMT” www.redes.unlp.edu.ar
No se devolvió la pagina. Sucede porque le solicito que la devuelva en caso de que haya sido modificada desde una fecha posterior a la ultima vez que se modifico.

e. ¿Qué significa el encabezado Etag?

ETag o **entity tag** (*etiqueta de entidad* en español) es parte de HTTP, el protocolo para la World Wide Web. Es uno de los varios mecanismos que HTTP proporciona para la validación de caché web, y que permite a un cliente realizar peticiones condicionales. Esto permite que las cachés sean más eficientes y ahorra ancho de banda, puesto que un servidor web no necesita enviar una respuesta completa si el contenido no ha cambiado. Los ETags también pueden ser usados para el control de concurrencia optimista, como una manera de ayudar a prevenir que actualizaciones simultáneas de un recurso se sobrescriban entre sí.

Un ETag es un identificador opaco asignado por un servidor web a una versión específica de un recurso que se encuentra en una URL. Si el contenido del recurso en esa URL cambia, se le asigna un nuevo y diferente ETag. Usado de esta manera los ETags son similares a las huellas digitales, y se puede comparar de forma rápida para determinar si dos versiones de un recurso son la misma. Comparar ETags sólo tiene sentido con respecto a una URL - los ETags para los recursos obtenidos

a partir de diferentes URLs pueden o no ser iguales, por lo que no se puede inferir ningún significado de su comparación.

f. Investigue el encabezado If-Modified-Since. ¿Para qué cree que pueden servir los tres encabezados anteriores?

The **If-Modified-Since** request HTTP header makes the request conditional: the server will send back the requested resource, with a 200 status, only if it has been last modified after the given date. If the request has not been modified since, the response will be a 304 without any body; the Last-Modified header will contain the date of last modification. Unlike If-Unmodified-Since, If-Modified-Since can only be used with a GET or HEAD.
When used in combination with If-None-Match, it is ignored, unless the server doesn't support If-None-Match.
The most common use case is to update a cached entity that has no associated Etag.

Sirven para mejorar la velocidad de carga, aprovechando los datos ya almacenados en cache.

11. Utilizando la VM, realice las siguientes pruebas:

a. Ejecute el comando ’cat /home/redes/prueba-http-1-0.txt’ y copie la salida completa (incluyendo los dos enter que aparecen debajo del texto).

b. Desde la consola, ejecute el comando telnet www.redes.unlp.edu.ar 80 y luego pegue el contenido que tiene almacenado en el portapapeles. ¿Qué ocurre luego de hacerlo?

c. Repita el proceso anterior, pero copiando el contenido del archivo /home/redes/prueba-http-1-1.txt. Verifique que debería poder pegar varias veces el mismo contenido sin tener que ejecutar telnet nuevamente.

12. En base a lo obtenido en el ejercicio anterior, responda:
-¿Qué está haciendo al ejecutar el comando telnet?

The **telnet** command is used to communicate with another host using the TELNET protocol. If **telnet** is invoked without the *host* argument, it enters command mode, indicated by its prompt (**telnet**>). In this mode, it accepts and executes the commands listed below. If it is invoked with arguments, it performs an **open** command with those arguments.

Telnet (***Tel****ecommunication* ***Net****work*) es el nombre de un protocolo de red que nos permite acceder a otra máquina para manejarla remotamente como si estuviéramos sentados delante de ella. También es el nombre del programa informático que implementa el cliente. Para que la conexión funcione, como en todos los servicios de Internet, la máquina a la que se acceda debe tener un programa especial que reciba y gestione las conexiones. El puerto que se utiliza es el 23.

Al conectarse a un servidor mediante Telnet (comunmente utilizando el puerto 23), nuestra computadora pasa a ser una terminal del servidor al cual nos conectamos. Ésto significa que las

órdenes que demos se ejecutarán en él y no en nuestra máquina, por lo tanto nos es posible correr aplicaciones que el servidor posee sin utilizar recursos de nuestra computadora (solo los utilizados por el modem para la conexión).

-¿Qué comando HTTP utilizó? ¿Qué recurso solicitó?

Se utilizo el comando GET. Se solicito /http/HTTP-1.1/

-¿Qué diferencias notó entre los dos casos? ¿Puede explicar por qué?
El primero utiliza la version 1.0 de HTTP que no permite establecer conexiones persistentes, mientras que el segundo usa la version 1.1 que deja abierta la conexión para realizar otros pedidos.

Con curl no es necesario enviar cabeceras para hacer el requerimiento , mientras que con telnet si.
Curl no es interactivo y telnet si.
Con curl la orden se ejecuta en nuestra maquina y con telnet en el servido

-¿Cuál de los dos casos le parece más eficiente? Piense en el ejercicio donde analizó la cantidad de requerimientos necesarios para obtener una página con estilos, javascripts e imágenes. El caso elegido, ¿puede traer asociado algún problema?
El segundo caso es mas eficiente para realizan multiples solicitudes, pero tenerlo abierto durante mucho tiempo puede impedir que otros realicen solicitudes.

13) La página www.redes.unlp.edu.ar/http/idioma.php tiene soporte para visualizarse en inglés y en español. Manipule los encabezados de HTTP para visualizar la página en los diferentes idiomas.

Curl -H “Accept-Language:es-AR” www.redes.unlp.edu.ar

Para poder recuperar páginas en inglés hay que enviar el encabezado de solicitud “Accept-Language: en” para poder recibir la página en el idioma inglés, o cualquier otro idioma deseado. Si mandamos una solicitud con un idioma no soportado entonces se nos enviará la versión del documento por defecto, en el caso del ejercicio en español.

14. En el siguiente ejercicio veremos la diferencia entre los métodos POST y GET. Para ello, será necesario utilizar la VM y la herramienta Wireshark.

El método HTTP utilizado para enviar los datos varía según la página. Esto se especifica como parámetro method dentro de la etiqueta HTML form.
La diferencia entre estos métodos es que si se utiliza el POST los datos que el usuario ingresa en el formulario viajarán en el cuerpo de la entidad. Si no se utiliza el post y se utiliza el GET entonces los datos viajarán en la URL solicitada, lo cual es riesgoso en caso que se envíe información privada.
curl -X POST -F “materia: redes” -F “tema:curl” http://localhost

GET
http://www.redes.unlp.edu.ar/http/metodos-lectura-valores.phpform_nombre=pepe&form_apellido=pepa&form_mail=pepe%40hotmai.com&form_sexo=sexo_masc&form_pass=123456

POST
http://www.redes.unlp.edu.ar/http/metodos-lectura-valores.php

Ejercicio de parcial

curl -X ?? www.redes.unlp.edu.ar/??
> HEAD /metodos/ HTTP/??
> Host: www.redes.unlp.edu.ar
> User-Agent: curl/7.54.0
< HTTP/?? 200 OK
< Server: nginx/1.4.6 (Ubuntu)
< Date: Wed, 31 Jan 2018 22:22:22 GMT
< Last-Modified: Sat, 20 Jan 2018 13:02:41 GMT
< Content-Type: text/html; charset=UTF-8
< Connection: keep-alive

a. ¿Qué versiones de HTTP podría estar utilizando el servidor?
Debe ser la version 1.1 o mayor para soportar el keep-alive.

b. ¿Qué método está utilizando? Dicho método, ¿retorna el recurso solicitado?
Esta usando el metodo HEAD. Si, se solicito la cabecera y esta es la que se muestra.

c. ¿Cuál es el recurso solicitado?
Se solicita el header de la pagina.

d. El recurso solicitado, ¿fue obtenido exitosamente?
Si, la respuesta es positiva (200 OK)

e. Si la solicitud hubiera llevado un encabezado que diga: If-Modified-Since: Sat, 20 Jan 2018 13:02:41 GMT ¿Cuál habría sido la respuesta del servidor web? ¿Qué habría hecho el navegador en este caso?
Ubiese devuelto el header con *304 Not Modified* en el codigo de respuesta indicando que la pagina no fue modificada.