UTPL
La Universidad Católica de Loja

Modalidad Abierta y a Distancia

Arquitectura de Redes

Guía didáctica

UTPL

Departamento de Ciencias de la Computación y Electrónica

Sección departamental de Electrónica y Telecomunicaciones

Arquitectura de Redes

Guía didáctica

Autora:

Liliana Elvira Enciso Quispe

DRBD_4018

Asesoría virtual
www.utpl.edu.ec

Arquitectura de Redes
Guía didáctica
Liliana Elvira Enciso Quispe

Universidad Técnica Particular de Loja

Diagramación y diseño digital:

Ediloja Cía. Ltda.
Telefax: 593-7-2611418.
San Cayetano Alto s/n.
www.ediloja.com.ec
edilojainfo@ediloja.com.ec
Loja-Ecuador

ISBN digital - 978-9942-39-139-1

29 de marzo, 2021

Índice

1. Datos de información

1.1. Presentación de la asignatura

1.2. Competencias genéricas de la UTPL

- Orientación a la investigación e innovación.

1.3. Competencias específicas de la carrera

- Administrar los servicios de tecnologías de información de la organización utilizando buenas prácticas de la industria asegurando la continuidad operacional del negocio.

1.4. Problemática que aborda la asignatura

La asignatura aborda aspectos fundamentales sobre la gestión y seguridad de redes de comunicaciones de datos, así como también principales protocolos de comunicación que corren sobre la capa de aplicación y que hacen posible la aplicación y selección de esquemas de redes más idóneas para una situación específica.

2. Metodología de aprendizaje

1. **METODOLOGÍA DE APRENDIZAJE**

Estudiar a distancia es un reto que requiere esfuerzo, dedicación y sobre todo de organización, por ello debe hacer de esta actividad un trabajo continuo y sistemático, organice su tiempo para aprovechar los contenidos que contempla la asignatura y el aporte que le da a su formación profesional.

La metodología de aprendizaje de esta asignatura será ABP (Aprendizaje Basado en Problemas). Es una metodología centrada en el aprendizaje, en la investigación y reflexión que siguen los estudiantes para llegar a una solución ante un problema planteado por el docente. A través de esta metodología se promueve el desarrollo del pensamiento crítico, la capacidad de resolución de problemas, la empatía, la gestión de recursos y las habilidades de comunicación.

Lo innovador del ABP es el planteamiento de problemas como punto inicial para adquirir nuevos conocimientos y el protagonismo del estudiante como parte de la gestión de su aprendizaje. En este aprendizaje el estudiante debe construir su conocimiento sobre la base de problemas y situaciones de la vida real. El proceso que se sigue en el ABP es: primero se presenta el problema, luego se identifican las necesidades de aprendizaje, se busca la información necesaria y finalmente se vuelve al problema para su resolución.

Además, para el desarrollo de esta metodología ponemos a disposición los recursos educativos: guía didáctica, libro básico, páginas *web*, RFCs, videos tutoriales y ejemplos de herramientas para redes de comunicación.

3. Orientaciones didácticas por resultados de aprendizaje

Primer bimestre

Resultado de aprendizaje 1

- Discute las arquitecturas típicas de gestión de la red.

Estimado estudiante, durante este bimestre abordaremos el interesante tema de las aplicaciones por *Internet,* para ello hemos dispuesto de 6 unidades, la primera se refiere a la transferencia de datos que abordaremos a continuación:

Es obligatorio antes de comenzar que Ud. realice las siguientes actividades, a fin de que durante el desarrollo de la asignatura le sea más fácil el cumplimiento de las actividades programadas:

Informaciones Importantes

Contenidos, recursos y actividades de aprendizaje

Bienvenido al presente ciclo académico, donde le espera conocer mucho sobre el fascinante mundo de la arquitectura de redes. ¡Empecemos!

Semana 1

Unidad 1. Capa de aplicación y aplicaciones de transferencia de datos

1.1. Fundamentos de redes

Es necesario que realice una revisión de los contenidos abordados en los ciclos anteriores, por eso le recomendamos leer primero el texto básico sección 1.5. En donde se especifica las capas de los protocolos y sus modelos de servicio.

Las redes se estructuran en modelos de referencia en capas, entre los más conocidos tenemos: OSI y TCP/IP. En la Figura 1., se muestra las capas del modelo OSI y sus funciones principales de acuerdo a la capa que la compone. En la Figura 2., así mismo se visualiza el modelo TCP/IP con sus 5 capas de acuerdo a Kurose & Keith (2017), con sus funciones principales.

Figura 1.
Modelos de referencia de 7 capas (OSI)

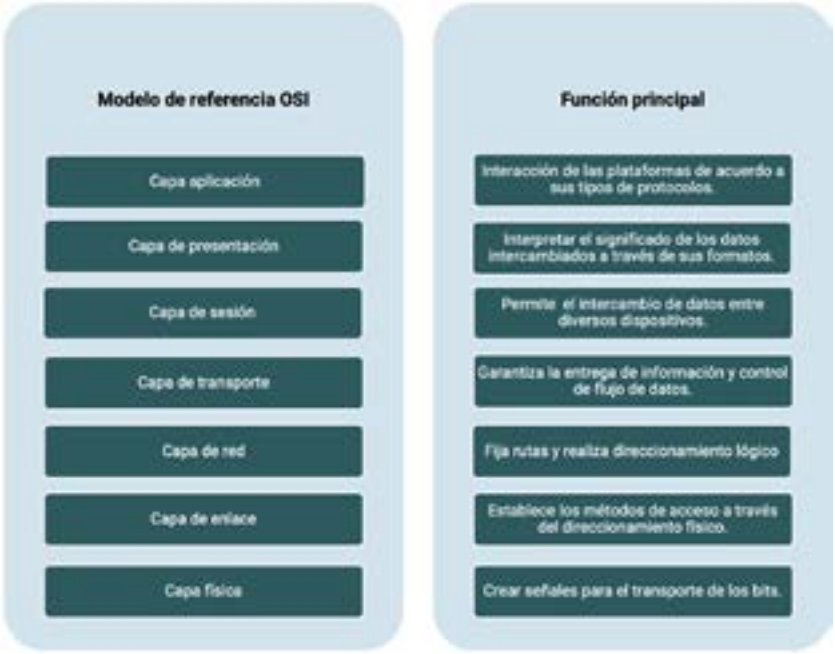
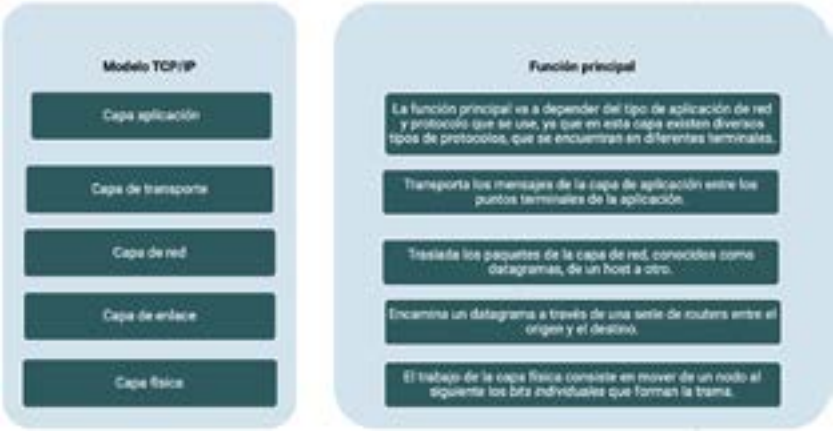


Figura 2.
Modelo TCP/IP con sus 5 capas



1.2. Principios de las aplicaciones de red

La gran red de redes es lo que conocemos como Internet (INTERnational NETwork of computer), como su nombre lo indica es un conjunto de redes que trabajan coordinadamente a través de herramientas y protocolos de acuerdo a cada capa establecida, comúnmente TCP/IP, con el objetivo de intercambiar información. Por ejemplo, en la aplicación *Web* se emplean dos programas diferentes que se comunican entre sí: el navegador que se ejecuta en el host del usuario (una computadora de escritorio, un portátil, una tableta, un teléfono inteligente, iPad, etc.) y el programa del servidor web que se ejecuta en el *host* que actúa como servidor *web*. Otro ejemplo sería el caso de un sistema de compartición de archivos P2P, en el que se emplea un programa en cada *host* que participa en la comunidad de compartición de archivos. En este caso, los programas instalados en los distintos *hosts* pueden ser similares o idénticos.

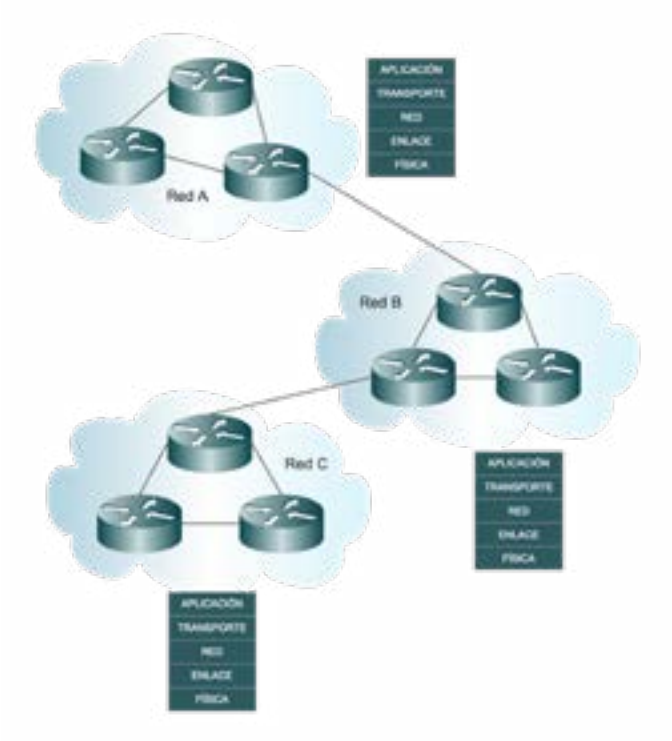
La administración de esta gran red es muy complicada, por lo que se estableció una administración distribuida a través de Proveedores de servicios de Internet o ISP (Proveedor de Servicios de Internet) y por sus siglas en inglés (Internet Services Provider). Los ISP son empresas dedicadas a dar servicio de *Internet* a los usuarios a través de diferentes tecnologías como: Red telefónica, ADSL, GSM, Wifi, etc.

Actividad de aprendizaje recomendada

Estimado estudiante, ahora vamos a realizar una actividad para averiguar cómo se comporta nuestro entorno y las variables que en él se encuentra, por ello le recomendamos identificar todos los proveedores de su región y revisar qué servicios a nivel de tecnología ofrecen para los clientes residenciales y comerciales.

1.2.1. Arquitectura de las aplicaciones de red

Figura 3.
Arquitectura de Internet con capas del modelo TCP/IP



Para abordar el presente tema es importante que lea la sección 2.1.1. del texto básico, como puede observar en la Figura 3, *Internet* a través de su modelo TCP/IP está estructurado en cinco capas, cada uno de estos niveles comprende un conjunto de protocolos que son capaces de realizar acciones relativas a la comunicación de los computadores en una relación entre pares de capas. En orden ascendente estas capas del modelo TCP/IP serían: física, enlace, red, transporte y aplicación.

1.2.2. Proceso de comunicación entre procesos

Para entender el proceso de comunicación entre procesos es importante entender el significado de: ¿Qué es un proceso?, pues podríamos decir que es un programa ejecutándose en un *host*. Se pueden tener dos tipos de comunicaciones entre procesos; una comunicación interna en el *host* denominada comunicación interproceso; y, la comunicación entre diferentes *hosts* o dispositivos que se da a través de protocolos de nivel de aplicación.

Los protocolos definen las reglas dentro del proceso de comunicación con la finalidad de controlar el intercambio de datos, dictar el formato de paquetes, evaluar tiempos y funciones adicionales requeridas durante la comunicación.

También es importante establecer que los procesos de dos sistemas terminales diferentes se comunican entre sí intercambiando mensajes a través de la red de computadoras.

En las asignaturas afines a esta asignatura dentro de la carrera hemos revisado algunos protocolos de comunicación en las capas inferiores a la capa de aplicación. Por favor, revise los conceptos expuestos en la sección 2.1.2 a fin de determinar todas las normativas que son orientadas por los protocolos de aplicación.

A continuación, se muestra algunos protocolos importantes por las 5 capas.

a. Capa física: DSL, ISDN, Ethernet entre otros.
b. Capa de enlace: Point to Point Protocol o PPP.
c. Capa de red: Internet Protocol o IP.
d. Capa de transporte: Transmission Control Protocol o TCP.
e. Capa de aplicación: HTTP, HTTPS, FTP, SMTP entre otros.

En este bimestre se revisará los principales protocolos de capa de aplicación que establecen servicios entre el cliente y el servidor, describiendo cómo debe ser el diálogo entre ellos. Entre las principales características que definen a los protocolos de aplicación están:

- Semántica establecida en cada uno de los campos.
- El tipo de mensaje intercambiado.
- La sintaxis de los mensajes.
- Las reglas de intercambio dependiendo del tipo de protocolo, en tiempos, acciones, etc.

1.2.3. Agente de usuario

El agente de usuario está representado por un tipo de navegador que está haciendo la solicitud al servidor, esta se encuentra en la capa de aplicación y es la interfaz entre el usuario y la capa de red. Un ejemplo de agente de usuario puede ser Mozilla, que es un navegador de Firefox.

Tabla 3.
Servicios populares asociados a agentes de usuario

Servicio	Protocolo	Agente de usuario
Web	HTTP	Navegador
E-mail	SMTP, POP3, IMAP, HTTP	Lector de correo
Streaming audio/video	RTSP	Media *player*

En la Tabla *3*, se presentan algunos servicios populares con los protocolos asociados a ellas y el agente de usuario respectivo. En la Figura 4 se observan los íconos de los principales navegadores

utilizados en diferentes sistemas operativos, a través de los cuales se puede acceder a un servicio.

1.2.4. Formato URL

Un archivo base HTML puede hacer referencia a muchos objetos contenidos en una *web* mediante los URL. Dicha URL tiene un formato que es una notación que expresa de manera uniforme los distintos recursos u objetos a los cuales se pueden acceder con los clientes *Web*. Dicho formato se encuentra descrito en RFC1738 y RFC1808.

Un URL consta de tres campos fundamentales: protocolo, dominio, nombre del servidor *Web*, camino de la información y nombre del documento o recurso (ver Figura 5).

Algunas URLs pueden también especificar puertos para establecer procesos más definidos.

Figura 4.
Campos fundamentales de una URL

Figura 5.
Agentes de usuario para servicio web (Depogi, 2020)

Ahora lo invitamos a revisar los conocimientos adquiridos, si su nota es baja por favor vuelva a leer y revisar los contenidos. Esta actividad es importante para determinar cuáles son los apartados que requieren una lectura adicional. El solucionario para este cuestionario lo encontrará al final de la guía.

Autoevaluación 1

Lea detenidamente cada enunciado y seleccione la respuesta correcta:

1. ¿Por cuántas capas está compuesta el modelo de referencia OSI?

   a. 5
   b. 4
   c. 7
   d. 8

2. Es la función principal de la capa de enlace en el modelo de referencia OSI:

   a. Garantiza la entrega de información sin errores.
   b. Direcciona mensajes intra y extra red.
   c. Establece los métodos de acceso.
   d. Datos de acceso al medio físico.

3. ¿Por cuántas capas está compuesta el modelo TCP/IP?

   a. 5
   b. 6
   c. 7
   d. 3

4. Es la función principal de la capa de red en el modelo TCP/IP.

   a. En esta capa un protocolo está distribuido entre varios sistemas terminales, constituye todos los servicios a los usuarios finales. Se encarga del formateo de información.
   b. Traslada los paquetes de la capa de red, conocidos como datagramas, de un host a otro.
   c. Encamina un datagrama a través de una serie de *routers* entre el origen y el destino.
   d. Transporta los mensajes entre los puntos terminales de la aplicación.

5. El protocolo PPP en el modelo TCP/IP es parte de la capa:

   a. Red
   b. Aplicación
   c. Enlace
   d. Transporte

6. ¿Qué significa ISP?

   a. Internet Services Protocol.
   b. Internet System Progress.
   c. Internet Services Provider.
   d. Internet Signal Programming.

7. El protocolo FTP en el modelo TCP/IP es parte de la capa:

   a. Red
   b. Aplicación
   c. Enlace
   d. Transporte

8. Un agente de usuario de un servicio web está asociado a:

   a. Navegador
   b. Streaming
   c. E-mail
   d. Audio/Video

9. En un formato URL http o https define a un:

   a. Elemento de red
   b. Protocolo
   c. Nombre de servidor
   d. Recurso

10. ¿En qué RFCs se encuentra descrito el formato URL?

   a. RFC1673 y RFC4536
   b. RFC2578 y RFC2114
   c. RFC1738 y RFC1808
   d. RFC2718 y RFC3528

Ir al solucionario

Semana 2

Unidad 2. Capa de aplicación y aplicaciones de transferencia de datos

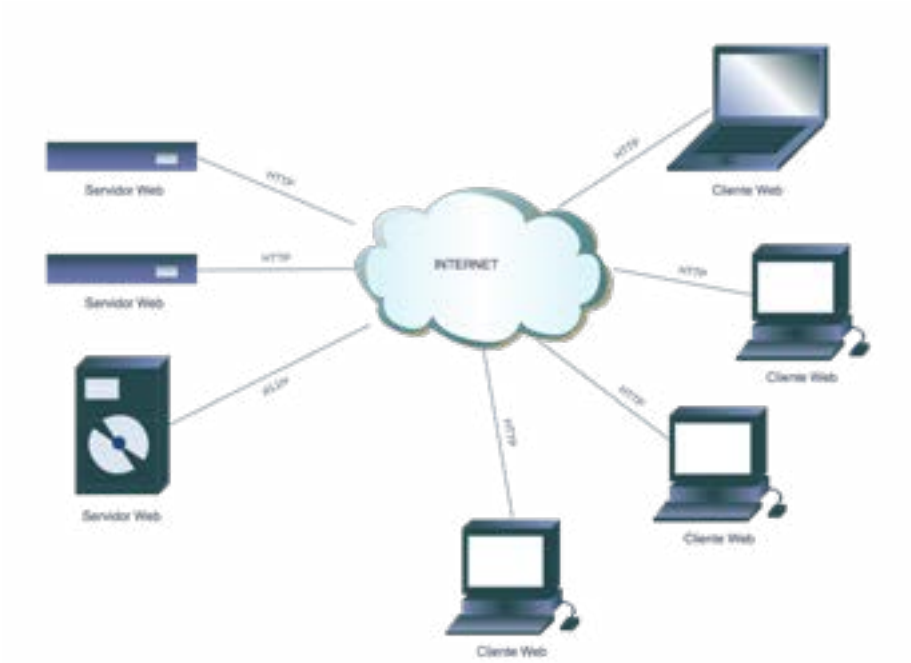
Estimado estudiante, esta semana revisaremos los protocolos HTTP y FTP. Para abordar el tema del protocolo HTTP es necesario que usted revise la sección 2.2 del texto básico.

2.1. Protocolo HTTP

El protocolo HTTP (en inglés, Hypertext Tranfer Protocol), es un protocolo de comunicación de la capa de aplicación que permite la transmisión de documentos hipermedia, como HTML. El modelo que maneja la red de redes es cliente-servidor; que como sabemos trata de un modelo de aplicación distribuido, en el que los clientes realizan diferentes tareas que son asignadas a proveedores de servicios, los cuales disponen de servidores para responder a cada uno de los requerimientos. También podemos afirmar que HTTP es un protocolo sin memoria del estado, puesto que no guarda ninguna información acerca de los clientes y las peticiones realizadas por estos.

En la Figura 6 se observa que los diversos dispositivos a través de las aplicaciones implementan servicios diferentes sobre el protocolo HTTP.

Figura 6.
Accesos web con HTTP desde diversos dispositivos y aplicaciones



2.1.1. Introducción a HTTP

El protocolo HTTP fue creado con el fin de intercambiar hipertexto mediante enlaces o comúnmente llamados páginas *web*, las cuales se encuentran alojadas en un servidor. Es importante señalar que el concepto *Web* fue creado en 1989 en Suiza por el inglés Tim Berners-Lee y el belga Robert Cailliau en la Organización Europea para la Investigación Nuclear y tuvo su crecimiento acelerado en 1993 (García, 2018).

La *Web* es un espacio de información global cuyo objetivo es organizar la información que se encuentra en Internet. La solución de esta organización consiste en hacer sencilla la comunicación entre los *hosts*. Esto es posible con un protocolo sencillo como es el protocolo HTTP. Pues hay que entender que los usuarios las 24/7 horas y días y 365 días al año hacen uso de la *Web* a través de una diversidad de consultas de información que se pueden realizar, estas consultas se realizan en modelo cliente-servidor, en específico clientes Web-servidores Web y el intercambio de información entre ellos se realiza mediante el Protocolo de Transferencia de Hipertexto o HTTP.

Es importante hacernos las siguientes preguntas: ¿a qué nos referimos con hipertexto?, ¿alguna vez escuchó este término?, ¿con qué procesos se lo puede relacionar?, ¿ha escuchado hablar de los hipervínculos? pues bien, hipertexto es el nombre que le asignamos al texto no lineal que permite a través de la pantalla acceder a referencias incrustadas, es decir a un texto relacionado y que luego puede ser accedido directamente. El hipervínculo es también hipertexto, pero con la condición especial de permitir relacionar automáticamente dos páginas *web*, o desde una página *web* a un archivo dentro de un mismo sitio o en otro sitio, basta con que esté conectado a *Internet*.

En consecuencia, la búsqueda y visualización de información por Internet requiere del uso de programas especiales o agentes de usuario denominados navegadores, exploradores o *browser* en inglés.

Para ampliar sus conocimientos, por favor, revise la sección 2.2 del texto básico, particularmente los numerales: 2.2.2 y 2.2.3.

Actividad de aprendizaje recomendada

Con el fin de que se familiarice con la Web, le proponemos revise qué navegadores existen en la actualidad y compare las características de cada uno, con el fin de que vea las ventajas que cada una ofrece en la búsqueda eficiente de la información.

2.1.2. Funcionamiento simplificado de HTTP

En relación al protocolo HTTP podemos establecer dos versiones importantes: HTTP 1.0 definida en RFC1945 y HTTP 1.1 definida en RFC2616. Es importante especificar que ambos realizan los siguientes pasos:

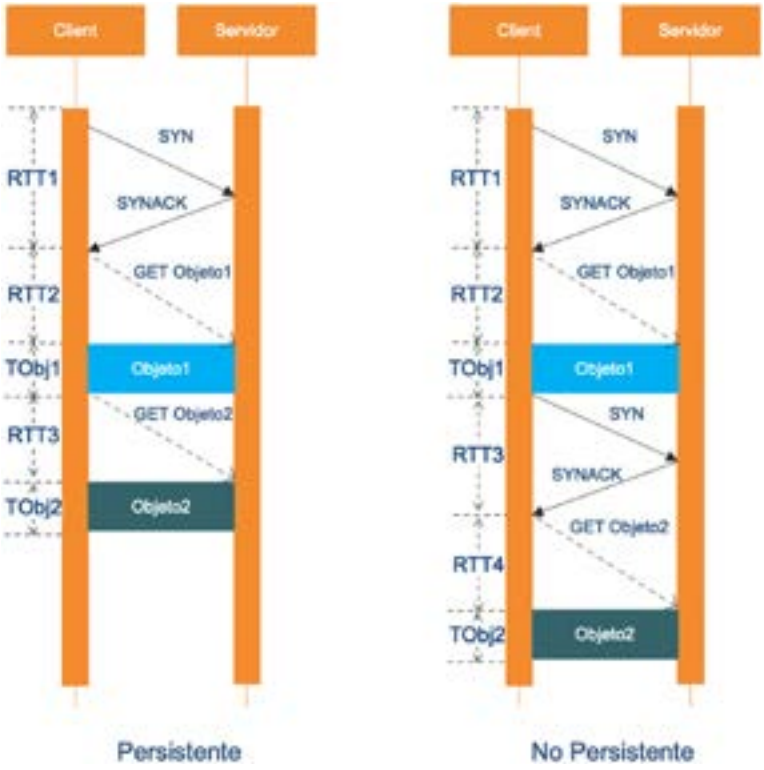
- El cliente inicia una conexión TCP al puerto 80 del servidor *web*.
- El servidor acepta la conexión TCP.
- El cliente solicita una página dentro de un sitio w*eb*.
- El servidor devuelve la página solicitada.
- El cliente la muestra y solicita los documentos incluidos.
- El servidor devuelve dichos documentos.
- El cliente muestra los documentos y la página solicitada.
- Finalmente, se procede a cerrar la conexión TCP.

2.1.3. Conexiones persistentes y no persistentes

Considerando que la comunicación entre el cliente y el servidor se hace durante un tiempo amplio, donde el cliente realiza una serie de solicitudes y el servidor responde. Cuando se da esta interacción sobre TCP, el desarrollador de aplicaciones debe tomar una decisión de enviar las solicitudes por separado o a través de una misma conexión. Para ahondar con estos detalles le invitamos a revisar

la sección 2.2.2 del texto básico, en esta sección se describe el problema que se tiene en HTTP 1.0 con las conexiones no persistentes, es decir, cuando se abre una nueva conexión para cada petición, por cuanto se sobrecarga la red con tráfico de control.

Figura 7.
Conexiones Persistente y No persistente de HTTP



En HTTP 1.1 para contrarrestar este problema se optó por mantener abierta la conexión hasta que el servidor o el cliente la cierren y a esto se denominó como conexión persistente, en la Figura 7 se esquematiza las diferencias entre una conexión persistente y no persistente secuencial.

El tiempo de respuesta, es decir, lo que se demora en enviar un paquete y recibir una respuesta asociada, se conoce como RTT por sus siglas en inglés *Round Trip Time*. Por lo que se utilizará un RTT para establecer la conexión TCP y un RTT para la petición HTTP del objeto o fichero. En la ecuación 1, podemos ver el tiempo total ($Tiempo_t$)de transmisión de un objeto, el cual será la suma de los dos RTT más el tiempo de transferencia del objeto o fichero. Esto es tanto para una conexión persistente y no persistente con un solo objeto; pero cuando se necesita transmitir 2 objetos la ecuación cambiaría. Ver Ecuaciones 2 y 3.

$$Tiempo_t = RTT1+RTT2+Tobj = 2RTT+Tobj \quad (1)$$

Para la transmisión de 2 objetos en conexión persistente sería:

$$Tiempo_t = RTT1+RTT2+RTT3+Tobj1+Tobj2 = 3RTT+2Tobj \quad (2)$$

Y para la conexión No persistente es con 2 objetos sería:

$$Tiempo_t = RTT1+RTT2+RTT3+RTT4+Tobj1+Tobj2 = 4RTT+2Tobj \quad (3)$$

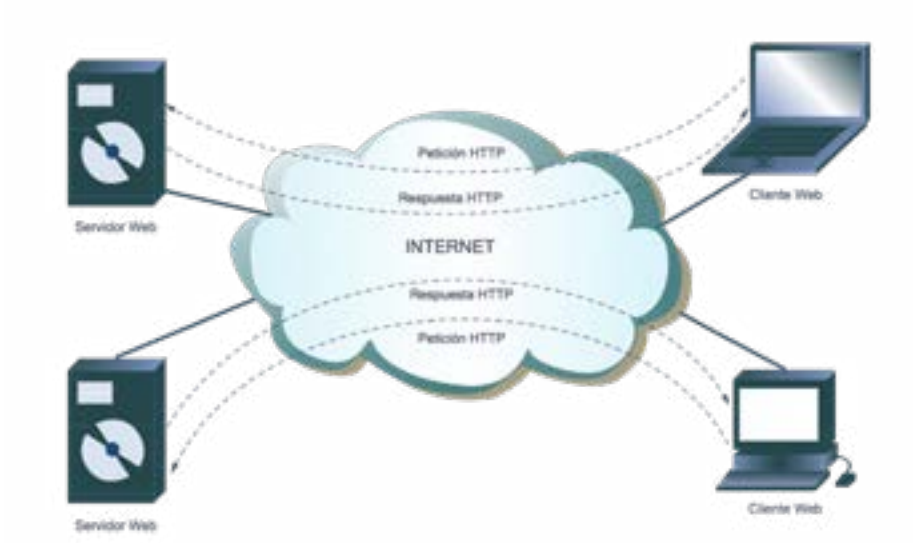
2.1.4. Formatos de los mensajes HTTP

Entre los métodos que tiene el protocolo HTTP se encuentran: solicitud/*request* y respuesta/response. En la Figura 8 se ejemplifica el comportamiento de ambos.

Figura 8.
Método Request/Response

Como todos sabemos en Internet se generan múltiples conexiones de este tipo al mismo tiempo. En la Figura 9 se muestra la interacción de varios clientes y servidores. La solicitud se realiza desde el cliente al servidor y las respuestas desde el servidor al cliente.

Figura 9.
Request/Response en Internet



Continuemos con el aprendizaje mediante la revisión de las solicitudes y respuestas HTTP.

Actividad de aprendizaje recomendada

Para ampliar sus conocimientos sobre este tema le recomendamos consultar todas las posibles cabeceras utilizadas por HTTP de acuerdo al agente de usuario de su preferencia.

Además, le invitamos a revisar el vídeo sobre Solicitudes, Respuestas y Cabeceras HTTP, del canal de YouTube Master IT, donde verá cuál es el proceso a seguir para visualizar el detalle de cada una de estas estructuras.

2.1.5. Almacenamiento caché Web

Normalmente es un ISP quien instala una caché web. Por ejemplo, una universidad podría instalar en su red LAN y configurar todos los navegadores del campus apuntando a la caché.

Figura 12.
Funcionamiento caché *Web*

La caché *Web* también se la conoce como servidor proxy, proxy *web* o simplemente *proxy*. Es considerada como una entidad que atiende las solicitudes HTTP en nombre del servidor *web* de origen. La caché *web* dispone de su propio almacenamiento en disco y mantiene en él copias de los objetos solicitados recientemente. En la Figura 12 se puede observar su funcionalidad.

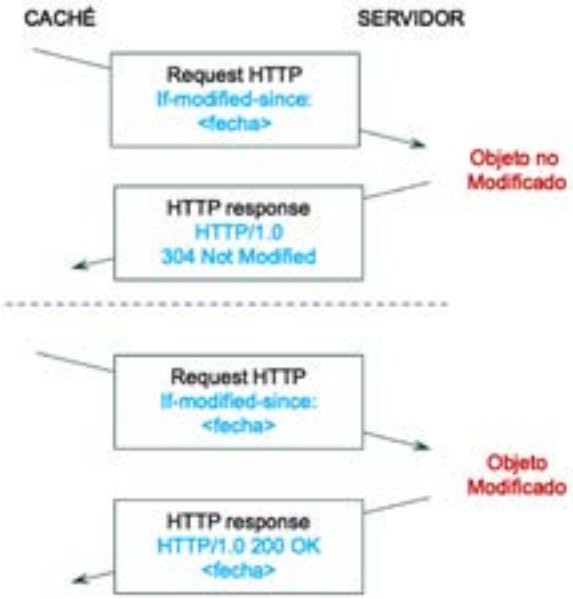
Entre las principales ventajas que tiene el servidor *proxy* están:

- Reducción de tiempos de respuesta a solicitudes de los clientes, o sea, eficiencia en las solicitudes de servicios *web*.
- Filtrado de información por lo que se reduce el tráfico en el enlace de acceso a Internet.
- Incremento en el rendimiento de las conexiones externas.

2.1.6. GET condicional

Aunque almacenar en caché reduce los tiempos de respuesta, este puede generar un problema: la copia de un objeto alojado en la caché y el cual podría estar desactualizado. Para este inconveniente HTTP dispone de un mecanismo de verificación de objetos en la caché. Este mecanismo es el GET condicional. Para ello basta con incluir en la cabecera de petición las siguientes proposiciones: If-Modified-Since, If-Unmodified-Since, If-Match, If-None-Match o If-Range. Es decir que el contenido de la respuesta solo será transmitido si se cumple las condiciones determinadas por esas líneas de cabeceras.

Figura 13.
Get condicional



El GET condicional también es conocido como caché del cliente, puesto que reduce el tráfico en las redes. En la Figura 13 se describe el funcionamiento de la Get condicional.

2.1.7. Lenguaje HTML

HTML (HyperText Markup Language) es el lenguaje de representación de documentos más común en la *Web* y que se referencia en RFC 1866. Como su nombre lo indica es un lenguaje de marcado que usa etiquetas para presentar y visualizar páginas multimedia que tienen hipertexto, por ejemplo: texto, hipervínculos, imágenes, vídeo, sonido, animaciones, objetos, entre otros.

En la Tabla *5* se encuentran algunas etiquetas básicas para crear documentos HTML.

Tabla 5.
Etiquetas o códigos básicos HTML

Etiquetas	Descripción
<html>	Define el inicio del documento HTML, le indica al navegador que lo que viene debe ser interpretado como código HTML.
<script>	Incrusta un script en una Web, o se llama a uno mediante src= "url del script".
<head>	Define la cabecera del documento HTML, puede contener información no visible al usuario.
<title>	Define el título de la página.
<link>	Sirve para vincular el sitio a hojas de estilo o iconos.
<style>	Sirve para colocar el estilo interno de la página; ya sea usando CSS, u otros lenguajes similares.
<body>	Define el contenido principal o cuerpo del documento. Dentro del cuerpo <body> podemos encontrar numerosas etiquetas. Por ejemplo:
h1> a <h6>	Encabezados o títulos del documento con diferente relevancia.
<table>	Define una tabla. <tr>: fila de una Tabla. <td>: celda de una Tabla.
	División de la página. Se recomienda, junto con *css*, para alinear contenido.

Etiquetas	Descripción
<a>	Hipervínculo o enlace, dentro o fuera del sitio *web*.
<img>	Imagen. Requiere del atributo *src*, que indica la ruta en la que se encuentra la imagen.
<b>	Texto en negrita
<i>	Texto en cursiva
<s>	Texto tachado
<u>	Texto subrayado

En la Figura 14 se muestra un ejemplo de código en HTML. El sitio *web* es de la página www.utpl.edu.ec; en esta captura se puede como HTML hace uso de los códigos o etiquetas descritas en la Tabla 5. Para su mejor comprensión trate de identificar estos códigos en algunas otras páginas *web*.

Figura 14.
Visualización de etiquetas básica de html

Actividad de aprendizaje recomendada

HTML no es el único lenguaje utilizado en Internet, también existen otros como: HTML dinámico, SGML, DocBook, MathML, Angular, Java, Perl, Ruby, etc. Como actividad complementaria es interesante que usted se familiarice con estos lenguajes y vea las potenciales que cada uno tiene.

2.1.8. Vulnerabilidad de HTTP

Todos conocemos la definición del protocolo HTTP, así como su uso. Este representa uno de los componentes esenciales de la arquitectura de la *web*. El cual a lo largo de los años se han ido lanzando diferentes versiones, y la última versión disponible es HTTP/3 que sucederá a la actual HTTP/2 que se utiliza ampliamente en la gran mayoría de *webs* de Internet.

El funcionamiento consiste cuando un cliente o usuario de la *web* intenta realizar una conexión mediante una solicitud, en el cual envía un mensaje con un formato determinado al servidor *web*. Ese servidor *web* es el que aloja al sitio o servicio *web* del cual necesitamos información. Este proceso se da todos los días de forma activa por todos los usuarios en la *web*. Gracias a los mismos, accedemos a la información que necesitamos. Sin embargo, existen varias actividades en la *web* que requieren el manejo de datos personales, o bien, cualquier tipo de dato que sea de carácter sensible, respetando por supuesto la ley de datos abiertos. En consecuencia, la seguridad de cualquier sitio *web*, incluyendo las aplicaciones *web*, son elementos importantes, tanto para los desarrolladores como para los usuarios finales. Existen diversas vulnerabilidades HTTP, las cuales pueden resultar casi tan severas como las XSS (Cross-Site Scripting), por citar un ejemplo.

A continuación, mostramos ocho vulnerabilidades descubiertas por los investigadores, Jonathan Looney de Neflix y Piotr Sikora de Google:

CVE-2019-9511 — HTTP/2 “Data Dribble”

CVE-2019-9512 — HTTP/2 “Ping Flood”

CVE-2019-9513 — HTTP/2 “Resource Loop”

CVE-2019-9514 — HTTP/2 “Reset Flood”

CVE-2019-9515 — HTTP/2 “Settings Flood”

CVE-2019-9516 — HTTP/2 “0-Length Headers Leak”

CVE-2017-9517 — HTTP/2 “Internal Data Buffering”

CVE-2019-9518 — HTTP/2 “Request Data/Header Flood”

Actividad de aprendizaje recomendada

Para esta actividad utilizaremos el programa Wireshark, el cual permite realizar capturas de tráfico de red para su posterior análisis. Usted puede descargar la herramienta desde Wireshark e instalarla de acuerdo al sistema operativo de su computador. Para realizar esta sencilla práctica demostrativa usted deberá seguir atentamente los siguientes pasos:

1. Ingrese al programa Wireshark.

2. En el menú superior vamos a la opción *capture*, y en interfaces seleccionamos nuestra dirección IP presionando el botón *start*; con ello se iniciará una escucha de nuestra interfaz de red para

posteriormente capturar los paquetes que se envíen a través de ella.

3. Ahora generaremos el tráfico que será analizado, para ello ingrese al navegador de su preferencia y accederemos a la página https://investigacion.utpl.edu.ec.

4. Regresemos al programa Wireshark, esperemos dos minutos y paremos la captura. En la ventana principal del programa tendremos algunas líneas. Las primeras líneas se refieren a la apertura y sincronización de la sesión TCP, esto lo puede comprobar revisando la columna *protocol*. Las siguientes líneas son las correspondientes al protocolo HTTP, le solicitamos realizar las siguientes actividades:

   - Identifique las etapas de sesión cliente-servidor.
   - Revise qué tipo de información ha sido transmitida, es decir, si es texto plano, imágenes, MIME, etc.
   - Compruebe el uso de comandos mencionados en la presente guía.
   - Revise en qué puerto se está realizando la comunicación.

2.2. Protocolo FTP

Previo a revisar los siguientes temas, le invitamos a que revise la sección 2.3 del texto básico. Es importante que reflexione sobre cada uno de los conceptos estudiados a través de esta lectura.

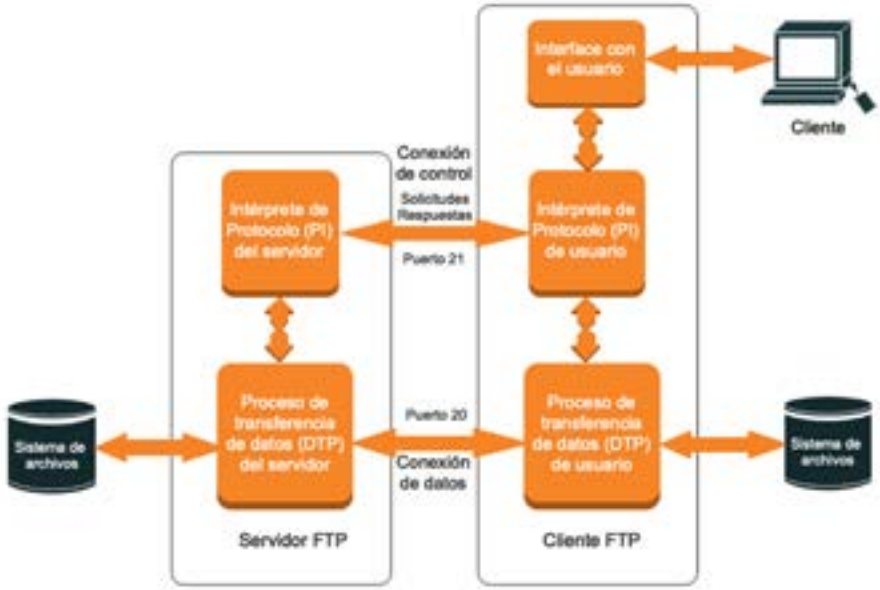
2.2.1. Introducción a FTP

Transferir grandes volúmenes de información es importante en la actualidad para los diversos tipos de usuarios. Como alternativa a esta necesidad tenemos al protocolo FTP.

El protocolo de transferencia de archivos o FTP por sus siglas en inglés (File Transfer Protocol) trabaja sobre TCP/IP, en modelo cliente-servidor y permite transferir grandes bloques de datos por la red, incluso cuando todavía no se tenía HTTP el FTP era muy utilizado. Entre los documentos técnicos que recogen las características relevantes del protocolo FTP tenemos:

- RFC 959 contiene información respecto a sus características y funcionamiento.
- RFC 1579 (Firewall-Friendly FTP). Describe el commando APSV de uso poco frecuente.
- RFC 2228 (FTP Security Extensions). Este documento describe el mecanismo para usar diferentes esquemas de autenticación y encriptación usando comandos como AUTH, PROT y nuevos comandos relacionados.
- RFC 2428 (FTP Extensions for IPv6 and NATs). Esta RFC hace que el protocolo FTP esté listo para IPv6.
- RFC 2640 (Internationalization of the File Transfer Protocol). Describe el uso de la codificación UTF-8 para nombres de archivo. Dado que el estándar FTP original solo permitía la codificación US-ASCII de 7 *bits*, esta extensión es totalmente compatible con versiones anteriores.
- RFC 4217 (Securing FTP with TLS). Esta RFC describe cómo proteger el FTP con TLS utilizando los comandos introducidos en RFC 2228. Cabe señalar que esta RFC en su mayor parte también se aplica al FTP sobre SSL obsoleto.

Figura 15.
Funcionamiento de protocolo FTP



En la Figura 15, se puede ver como FTP por defecto utiliza los puertos 20 y 21 con el protocolo TCP, el primero de ellos es para el flujo de datos entre el cliente y el servidor; y el segundo, es utilizado para el flujo de control; sin embargo, mientras se utiliza uno, el otro debe esperar. Para mayor detalle puede revisar el RFC 959.

Los datos en FTP admiten múltiples formatos de caracteres, entre ellos: ASCII y EBCDIC. Además, permite otras funcionalidades importantes, por ejemplo, compresión de ficheros, seguridad con autenticación, entre otras. En la Tabla *6* se puede visualizar tipos de ficheros asociados al tipo de transferencia.

Actividad de aprendizaje recomendada

Estimado estudiante, para ampliar su conocimiento sobre las redes le sugerimos consultar los códigos que pueden ser empleados por FTP. Para ello puede orientarse en el siguiente Lista de códigos de errores para FTP.

Tabla 6.
Tipos de archivos asociados al tipo de transferencia

Extensión de fichero	Tipo de transferencia
ARC (comprimido)	Binario
doc (documento	Binario
hqx (comprimido)	ASCII
html (página Web)	ASCII
pit (comprimido)	Binario
ps (poscript)	ASCII
shar (comprimido)	Binario
sit (comprimido)	Binario
tar (empaquetado)	Binario
txt (texto)	ASCII
uu (comprimido)	Binario
z (comprimido)	Binario
zip (comprimido)	Binario
zoo (comprimido)	Binario

2.2.2. Funcionamiento básico de FTP

Para que exista una comunicación es importante que participen dos elementos o entidades básicas, como son: el servidor y el cliente,

en los cuales existen *software* que permite dicha comunicación y que se ayuda de protocolos de comunicación de acuerdo al servicio requerido.

FTP es un protocolo de comunicación de carga y descarga de grandes volúmenes de información. El funcionamiento de dicho protocolo es:

1. En el servidor se tiene corriendo una aplicación denominada FTP daemon, quien ejecuta el protocolo FTP.

2. El cliente debe iniciar una conexión. Para ello el servidor le solicita su usuario y contraseña. Esta operación se conoce como conexión de control. Se puede tener seguridad adicional a través del uso de SSL y TLS. Los comandos usados en este paso son: *open, user, pass* y *site.*

3. El cliente entonces puede acceder al índice estructurado de archivos disponibles. El cliente puede navegar por este directorio para ubicar el archivo deseado si la operación que realizará es de tipo "*download*" descarga; o en su defecto para ubicar el directorio donde se pretende ubicar un nuevo archivo si la operación es de tipo "*upload*" subida de archivos. Los comandos para realizar esta navegación son: cd, lcd, ls y dir. En la Figura 16, podemos ver la estructura típica de visualización de ficheros.

4. El cliente realiza peticiones al servidor.

5. El servidor FTP gestiona las solicitudes que han sido realizadas por el cliente; para esta acción se debe establecer un control de los datos transferidos a través del comando *mode*, esto incluye:

- La forma en que los *bits* serán movidos desde un lugar a otro.
- Las representaciones de los datos en la arquitectura del sistema.
- La estructura de directorios en la cual los datos serán guardados.

6. El servidor envía los archivos, si fuera el caso.
7. Se termina la sesión.

Continuemos con el aprendizaje mediante la revisión de los comandos y respuestas de FTP.

2.2.3. Comandos de FTP

Los comandos más importantes en el protocolo FTP son:

FTP: este comando abre una sesión hacia el servidor FTP, a través de: ftp <dirección IP servidor>.

OPEN: sirve para abrir una sesión con un servidor ftp específico. Requisito indispensable para utilizar este comando es previamente haber establecido una conexión FTP.

CLOSE: este comando realiza la operación inversa al anterior, es decir, cierra la sesión.

GET: sirve para poder descargar los ficheros del servidor FTP, es posiblemente el comando más utilizado. Para aplicarlo basta con digitar: get <fichero>. Sin embargo, para utilizarlo es necesario estar en el directorio del servidor correcto.

MGET: permite descargar varios archivos a la vez desde el servidor FTP.

PUT: este comando sirve para subir ficheros al servidor FTP. Los ficheros serán tomados del directorio donde se ejecute: *put* <fichero>.

MPUT: sirve para subir varios archivos a la vez en el servidor FTP.

MODE: especifica el formato de la cadena de *bytes*.

LCD: este comando especifica el directorio local sobre el que se está trabajando.

CD: se utiliza para desplazarse entre los directorios del servidor FTP.

LS: sirve para listar los directorios y archivos que se encuentran en el servidor FTP.

DELETE: borra los archivos del servidor y por ello solo se puede aplicar en este.

APPEND: este comando reanuda una descarga que ha sido interrumpida. La interrupción puede ser de cualquier naturaleza, por ejemplo, corte de conexión, archivos demasiado pesados, etc.

USER: este comando permite cambiar la sesión de un usuario a otro distinto.

BYE: este comando cierra todas las sesiones que se tengan lanzadas con FTP.

En la Figura 16, se muestra un ejemplo de cómo se ve una estructura FTP.

Figura 16.
Estructura típica de un directorio FTP

Index of /ftp

Name	Last modified	Size	Description
Parent Directory		-	
ArmaSecreta.pdf	2019-11-28 07:21	228K	
Audioteca Service Ma..>	2019-11-28 07:34	-	
Buena-Idea.pdf	2019-11-28 07:34	2.2M	
Catalogo-SQI.pdf	2019-11-28 07:34	3.1M	
ClientesEnojados.pdf	2019-11-28 07:35	634K	
Coaching-de-Servicio..>	2019-11-28 07:35	3.7M	
Coaching.pdf	2019-11-28 07:35	3.7M	
Conexiones.pdf	2019-11-28 07:36	1.7M	
Cultura-de-Servicio.pdf	2019-11-28 07:36	196K	
EfectoWow2017.pdf	2019-11-28 07:36	316K	
Empowerment.pdf	2019-11-28 07:36	573K	
Empresas-Wow.pdf	2019-11-28 07:36	433K	
Experiencia-Increibl..>	2019-11-28 07:37	3.0M	
Healthcare.pdf	2019-11-28 07:37	2.3M	
Hoja-de-Inscripcion.pdf	2020-09-01 13:46	790K	
Hoja-de-inscripcion.pdf	2020-10-12 11:28	203K	
LET-y-Sentimientos.pdf	2019-11-28 07:37	572K	
LET.pdf	2019-11-28 07:37	1.2M	
Leales de por Vida.pdf	2019-11-28 07:37	1.2M	
Lealtad.pdf	2019-11-28 07:37	1.2M	
Libro Servicio al Cl..>	2019-11-28 07:39	-	
Liderazgo-de-Equipos..>	2019-11-28 07:40	1.1M	

En la Tabla 7 se puede revisar la sintaxis de los comandos más utilizados, para mayor detalle, por favor, consulte el Anexo 2.

Tabla 7.
Sintaxis de comandos FTP

Código	Descripción
USER	<SP> <username> <CRLF>
PASS	<SP> <password> <CRLF>
ACCT	<SP> <account-information> <CRLF>
CWD	<SP> <pathname> <CRLF>
CDUP	<CRLF>
SMNT	<SP> <pathname> <CRLF>
QUIT	<CRLF>
REIN	<CRLF>
PORT	<SP> <host-port> <CRLF>
PASV	<CRLF>
TYPE	<SP> <type-code> <CRLF>
STRU	<SP> <structure-code> <CRLF>
MODE	<SP> <mode-code> <CRLF>
RETR	<SP> <pathname> <CRLF>
STOR	<SP> <pathname> <CRLF>
STOU	<CRLF>
APPE	<SP> <pathname> <CRLF>
ALLO	<SP> <decimal-integer> [<SP> R <SP> <decimal-integer>] <CRLF>
REST	<SP> <marker> <CRLF>
RNFR	<SP> <pathname> <CRLF>
RNTO	<SP> <pathname> <CRLF>
ABOR	<CRLF>
DELE	<SP> <pathname> <CRLF>
RMD	<SP> <pathname> <CRLF>
MKD	<SP> <pathname> <CRLF>
PWD	<CRLF>
LIST	[<SP> <pathname>] <CRLF>
NLST	[<SP> <pathname>] <CRLF>
SITE	<SP> <string> <CRLF>
SYST	<CRLF>
STAT	[<SP> <pathname>] <CRLF>
HELP	[<SP> <string>] <CRLF>
NOOP	<CRLF>

2.2.4. Respuestas FTP

Los códigos de respuestas se organizan por grupos y constan de 3 dígitos, donde el primer dígito es el más significativo porque permite identificar a qué grupo pertenece la respuesta.

Tabla 8.
Respuestas FTP

Código	Descripción
1xx	Respuesta preliminar positiva
2xx	Respuesta finalización positiva
3xx	Respuesta intermedia positiva
4xx	Respuesta finalización negativa
5xx	Respuesta finalización permanente negativa

En la Tabla 8 se definen los grupos de respuestas. A continuación, se muestra la codificación y una breve descripción de algunas respuestas típicas:

331 Username OK, *password* requerida

125 conexión de datos lista; empezando transferencia

425 no se puede abrir conexión de datos

452 error en la escritura del archivo

220 servidor listo

200 comando ok

En la Figura 17 se tiene un ejemplo de sesión FTP con la utilización de comandos esenciales.

Figura 17.
Muestra de sesión FTP – Transferir un archivo a un host remoto *(Martínez-Diaz, s.f)*

```
[C:\SAMPLES]ftp host01.itsc.raleigh.ibm.com
Connected to host01.itsc.raleigh.ibm.com.
220 host01 FTP server (Version 4.1 Sat Nov 23 12:52:09 CST 1991) ready.
Name (rs60002): cms01
331 Password required for cms01.
Password: xxxxxx
230 User cms01 logged in.
ftp> put file01.tst file01.tst
200 PORT command successful.
150 Opening data connection for file01.tst (1252 bytes).
226 Transfer complete.
local: file01.tst remote: file01.tst
1285 bytes received in 0.062 seconds (20 Kbytes/s)
ftp> close
221 Goodbye.
ftp> quit
```

Actividad de aprendizaje recomendada

Existen muchos clientes FTP gratuitos en la red y los hay para todos los sistemas operativos conocidos; por ejemplo: FileZilla, CuteFTP, WinSCP, Cyberduck, CoreFTP LE, SmartFTP, CoffeCup, WSS FTP, FTP Now, WorldWide FTP, entre otros; por ello le invitamos a revisar dos de ellos y valorar sus potencialidades.

2.2.5. Modalidades de transferencia de datos

Existen dos formas de transferir datos: activa y pasiva, a continuación, las explicaremos.

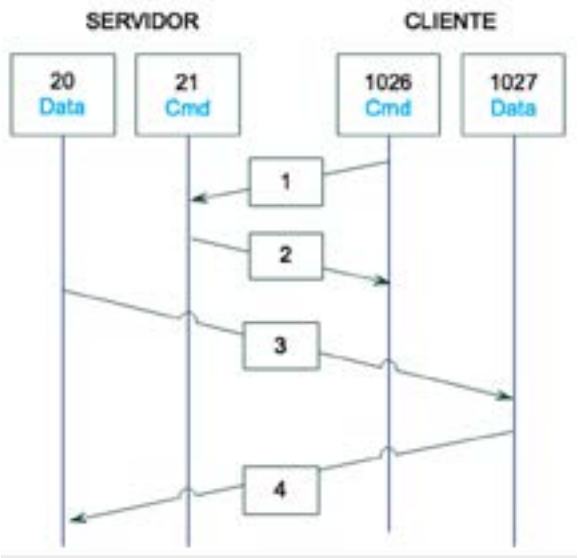
2.2.5.1. Transferencia activa de datos

En el tipo de *transferencia activa de datos* el servidor es quien se encarga de crear el canal de datos, esto se realiza por lo general

en el puerto 20 para el servidor, y del lado del cliente en cualquier número aleatorio mayor a 1024.

En este modo el cliente debe mandar el comando PORT al servidor a través de una conexión de control, indicándole el número de puerto; para que el servidor abra una conexión de datos lo que se confirma con el mensaje 200 (ver Figura 18). Este modo tiene graves problemas de seguridad, sobre todo porque el cliente se somete a aceptar cualquier conexión de entrada.

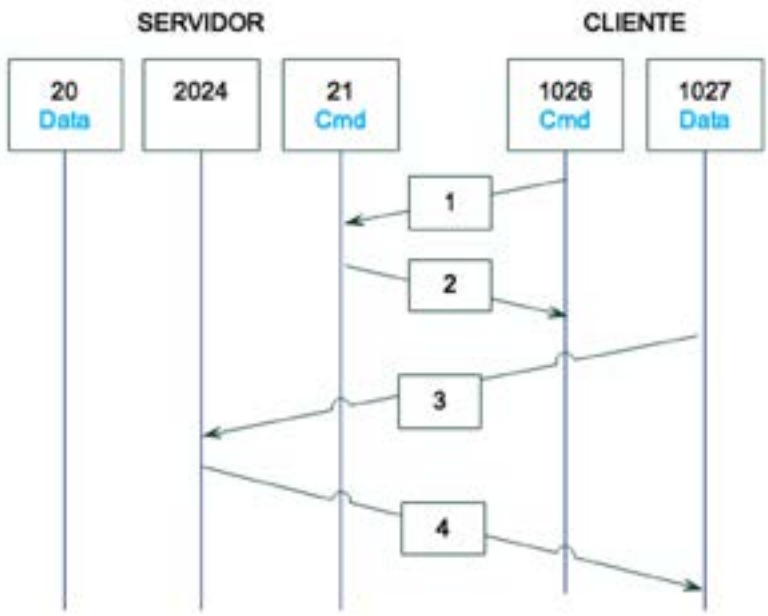
Figura 18.
Transferencia *activa* FTP



2.2.5.2. Transferencia pasiva de datos

En la *transferencia pasiva de datos* el cliente debe expresar su requerimiento de este tipo de conexión, explícitamente a través del comando PASV sobre una conexión de control, a esta solicitud el servidor le contesta el puerto asignado para esta conexión. El cliente inicia una conexión en dicho puerto (ver Figura 19).

Figura 19.
Transferencia pasiva FTP



2.2.6. Modos de un servidor de FTP

Entre los modos de un servidor FTP tenemos:

1. **FTP anónimo o de login anónimo**: es un servidor FTP abierto a todo público; aunque se pida usuario y contraseña, estos serían un mero formalismo. Los usuarios que accedan a este servicio podrán tener todos los privilegios, es decir, leer, subir o descargar archivos del servidor.

2. **FTP privado**: es un servidor que tiene las mismas funcionalidades que el anónimo; pero el cual solo pueden ingresar a través de una dupla usuario-contraseña válida y localizada en una base de datos, alojada en el sistema local del servidor.

Actividad de aprendizaje recomendada

Realice una consulta en internet de servidores FTP anónimos. Como recomendación le propongo la siguiente dirección: Servidores Anónimos FTP. Pero le sugiero revisar otras a fin de comparar las características que las definen.

2.2.7. Vulnerabilidades de FTP

FTP utiliza una comunicación simple por los enlaces, así pues, si se envía el usuario y contraseña en texto plano por redes no seguras, serían fácilmente capturados con técnicas como el *sniffing*.

Por ello, FTP permite conexiones anónimas a zonas restringidas donde solo se permiten descargas de archivos.

Lo invitamos a revisar este enlace, Vulnerabilidad en WebFTP en donde se revisa sobre el impacto de esta vulnerabilidad.

Actividad de aprendizaje recomendada

Una vez más le invitamos a usar el programa Wireshark. En esta ocasión utilizaremos la mencionada aplicación para revisar las sesiones FTP no seguras, a través de los siguientes pasos:

1. Ingrese al programa Wireshark.
2. En el menú superior vamos a la opción capture, y en interfaces seleccionamos nuestra dirección IP presionando el botón start; con ello se iniciará una escucha de nuestra interfaz de

red para posteriormente capturar los paquetes que se envíen a través de ella.

3. Ahora generaremos el tráfico que será analizado, para ello utilizaremos la conexión por línea de comando a un servidor FTP en Internet. Abra un terminal o *prompt* en su computador y digite: ftp.microsoft.com o ftp://ftp.microsoft.com, o cualquier otro servidor que usted conozca, como se explicó anteriormente bastará con definir un usuario y contraseña.

4. Realice ahora un requerimiento de archivo de acuerdo a lo aprendido.

5. Regresemos al programa Wireshark, esperemos dos minutos y paremos la captura. Se tendrán nuevamente unas entradas para la apertura de sesión TCP. Ahora identifique las entradas correspondientes al protocolo FTP y realice las siguientes actividades:

   a. Identifique las etapas de sesión cliente-servidor. Ponga especial cuidado en cómo se autentica el cliente ante el servidor y qué comandos utiliza.

   b. Revise qué tipo de información ha sido transmitida, es decir, si es texto plano, imágenes, MIME, etc.

   c. Compruebe el uso de comandos mencionados en la presente guía.

   d. ¿Cómo se realiza la descarga de archivos?, ¿qué comandos se utilizan?

   e. Observe con qué comando se da por finalizada la comunicación.

   f. Revise en qué puerto se está realizando la comunicación.

También debe revisar el vídeo del canal de YouTube, César Mancino, Taller Práctico Programación con Python. Operaciones con protocolo FTP?. En este video tutorial aprenderá a conectarse a un FTP con python mediante el uso del módulo ftplib. También se hará uso de los métodos retrbinary, ftplib.FTP y storbinary.

Ahora lo invitamos a revisar los conocimientos adquiridos, si su nota es baja por favor vuelva a leer y revisar los contenidos. Esta actividad es importante para determinar cuáles son los apartados que requieren una lectura adicional. El solucionario para este cuestionario lo encontrará al final de la guía.

Autoevaluación 2

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

1. ¿Qué es HTTP?

   a. Servicio de transferencia de información multimedia.
   b. Protocolo que emplea el servicio Web en cliente-servidor.
   c. Lenguaje de descripción de páginas Web.
   d. Aplicación cliente del servicio WWW.

2. ¿Qué acción ejecuta la etiqueta HEAD en el protocolo HTTP?

   a. Solicita un objeto.
   b. Indica los datos del objeto.
   c. Obtiene las cabeceras del objeto.
   d. Indica el estado de los objetos.

3. ¿Cuál de las siguientes órdenes es la correcta?

   a. GET / HTTP/1.0 <CR><LF><CR><LF>
   b. GET / HTTP/1.1 <CR><LF><CR><LF>
   c. GET / HTTP/1.0 <CR><LF><LF><CR>
   d. GET / HTTP/1.1 <CR><LF><LF><CR>

4. ¿Cuál de las siguiente es la mayor ventaja de utilizar proxy?

   a. Reducir el tráfico de redes.
   b. Encriptar información de autenticación.
   c. Limitar el número de conexiones por cliente.
   d. Permitir las conexiones a sitios *web* en Java.

5. ¿Qué acción ejecuta el comando RETR en el protocolo FTP?

   a. Obtienen un fichero del servidor.
   b. Especifica el puerto para la transferencia.
   c. Obtiene las cabeceras de un fichero.
   d. Lista los ficheros disponibles.

6. En un modo FTP pasivo, ¿cuál de los siguientes enunciados son ciertos?

   a. El servidor dispone el número de puerto para la comunicación de datos.
   b. El cliente siempre utiliza el mismo puerto para transferencia de datos.
   c. El cliente dispone el número de puerto para la comunicación de datos.
   d. Ninguna de las anteriores.

7. ¿Cómo se autentica un cliente en modo anónimo?

   a. Solo usuario
   b. Solo contraseña
   c. Usuario y contraseña única
   d. Usuario y contraseña registrada

8. ¿Cuál de las siguientes es una vulnerabilidad de FTP?

   a. Acceso a los mismos puertos siempre
   b. Envío de usuario y contraseña en texto plano
   c. Sesión anónima de cliente
   d. Número excesivo de servidores FTP en la *Web*

9. ¿En qué puerto funciona el servidor FTP?

   a. 250
   b. 120
   c. 80
   d. 21

10. ¿Qué significa la respuesta 331 en FTP?

   a. Error de escritura
   b. Empezando a transferir
   c. No puede abrirse la conexión
   d. Usuario listo

Ir al solucionario

Ahora le animamos a desarrollar los siguientes ejercicios:

Con el propósito de reforzar el nivel de conocimientos de la presente unidad resuelva los siguientes ejercicios propuestos en el texto básico.

Problemas: P1, P5, P7 y P10 del capítulo 2 texto básico.

Resultado de aprendizaje 2

- Diseña aplicaciones de red orientada a datos.

Contenidos, recursos y actividades de aprendizaje

Semana 3

Unidad 3. Capa de aplicación y aplicaciones de servicios

Estimado estudiante, esta semana nos centraremos en abordar las aplicaciones que se orientan a servicios de correo electrónico y servidor de nombre de dominio, pues en la actualidad no se puede concebir el uso de las redes sin estas herramientas.

3.1. Correo electrónico

El correo electrónico es una de las herramientas más utilizadas en la actualidad, que nos permite una comunicación asíncrona entre usuarios. En la actualidad existen diversos tipos de servidores de correos como, por ejemplo: Gmail, Hotmail, Zoho *mail*, etc. Los

cuales son escogidos por los usuarios, considerando las ventajas que estas ofrecen para el desenvolvimiento de sus actividades diarias.

En esta sección abordaremos las siguientes interrogantes: ¿qué protocolos se utilizan para el correo electrónico?, ¿cómo es el proceso para enviar y recibir correos electrónicos?; y, ¿qué herramientas se pueden utilizar para montar un servidor de correo electrónico? Para ampliar sus conocimientos le solicitamos leer la sección 2.4 del texto básico.

Para intentar responder a las interrogantes anteriormente planteadas es importante que usted diferencie entre un ENVÍO y una RECEPCIÓN de correo electrónico; y esto se debe realizar, ya que para estos tipos de transferencias se utilizan protocolos diferentes para cada operación. Para enviar correo se utiliza SMTP (Simple Mail Transfer Protocol) y para recibir se utiliza POP (Post Office Protocol) o IMAP (Internet Message Access Protocol).

Para disponer de un servicio de correo electrónico es necesario tener una dirección de correo, esto se puede realizar en cualquiera de los servidores que ofrecen este servicio, por ejemplo: www.gmail.com, www.hotmail.com, www.zohomail.com, etc.; o bien puede ir asociado a una institución como es el caso de www.utpl.edu.ec/mail de nuestra universidad. Las direcciones de correo tienen un formato específico:

buzón_usuario@dominio_de_correo

En esta sintaxis se pueden advertir los siguientes componentes:

1. Buzón de usuario: es cómo está identificado el usuario en el sistema dentro del servidor de correo o su alias.

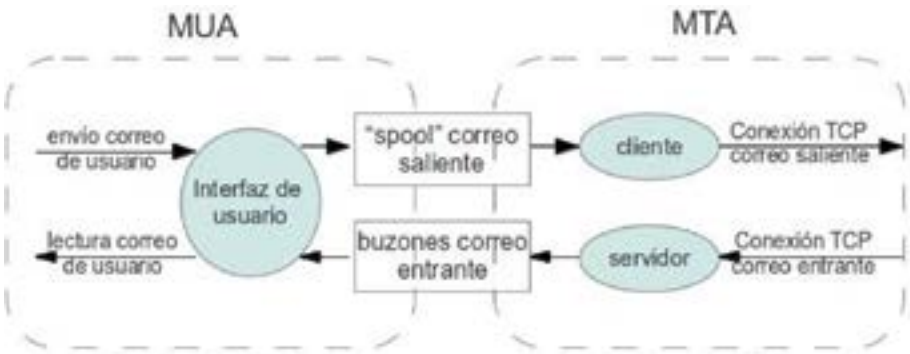
2. Dominio de correo: es cómo está identificado el servidor, por ejemplo, puede ser el nombre del servidor.

3. @: este es el separador por defecto de los componentes anteriores.

   Para poder transferir correo se necesitan dos agentes (ver Figura 20):

   - Agente de usuario (MUA- Mail User Agent); por ejemplo, Gmail y Hotmail.
   - Agente de transferencia de mensajes (MTA- Mail Transport Agent); por ejemplo, sendmail y postfix.

Figura 20.
Agentes utilizados en la transferencia de correo



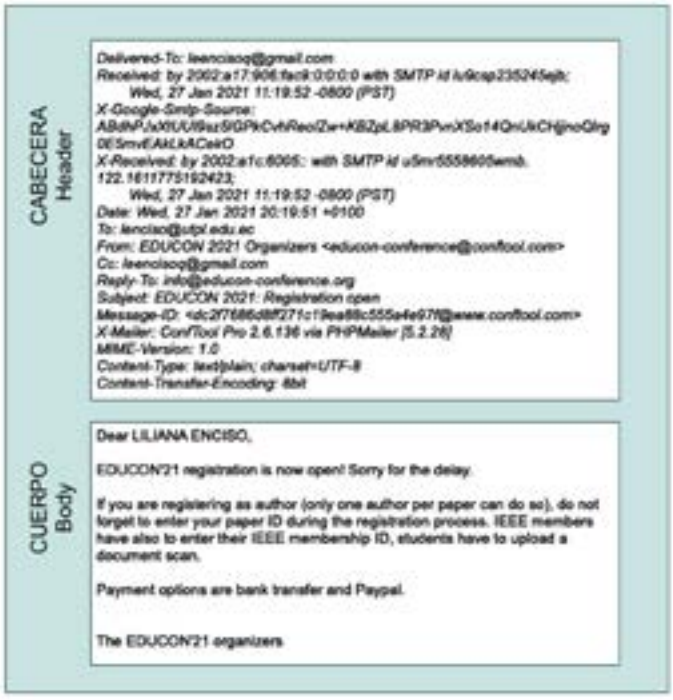
3.1.1. Mensaje de correo

En el documento RFC 2822, podemos encontrar el formato de los mensajes de correo electrónico, el cual consta de dos partes (ver Figura 21):

1. Cabecera (*header*): en la cabecera de un mensaje se dispone de la información respecto al mensaje, en la Tabla 9 puede revisar algunos campos básicos.

2. Cuerpo (*body*): contiene los datos que conforman el mensaje.

Figura 21.
Partes de un mensaje de correo



También se cuenta con un envoltorio (*envelope*) que se utiliza por los agentes de correo para entregar el mensaje, sin embargo, no se encuentra en el formato. En él se disponen comandos como: MAIL FROM y RCPT TO.

Tabla 9.
Campos de cabecera de un mensaje

Campo	Descripción
To:	Contiene las direcciones de correo electrónico de los principales destinatarios separados por comas.
Cc:	Destinatarios secundarios son las direcciones de correo que recibirán una "copia de carbón" del mensaje.
Bcc:	Copias a destinatarios ocultos o "copia de carbón ciega" del mensaje.
From:	Contiene las direcciones de correo electrónico del remitente.
Sender:	Dirección de correo del remitente.
Reveived:	Agentes que retransmitieron el mensaje.
Return-Path:	Trayectoria de regreso hacia el remitente.
Date:	Fecha y hora cuando se envió el mensaje.
Reply-To:	Dirección de correo donde se enviará la respuesta.
Message-Id:	Número del mensaje generado por el transporte de correo en el sistema remitente.
In-Reply-To:	Identificador del mensaje.
Subject:	Asunto del mensaje tratando de describir el contenido del mensaje.

Sin embargo, en los mensajes actuales son muy frecuentes los datos multimedia dentro del correo, los denominados MIME (Multipurpose Internet Mail Extensions), en la Tabla 10 se tienen los tipos de MIME (RFC 2046). Para introducir datos tipo MIME en un correo se utilizan líneas adicionales en la cabecera definidas en RFC 2045 y 2056. En la Figura 22 se muestran algunos campos MIME.

Figura 22.
Campos adicionales para MIME

Tabla 10.
Tipos de MIME

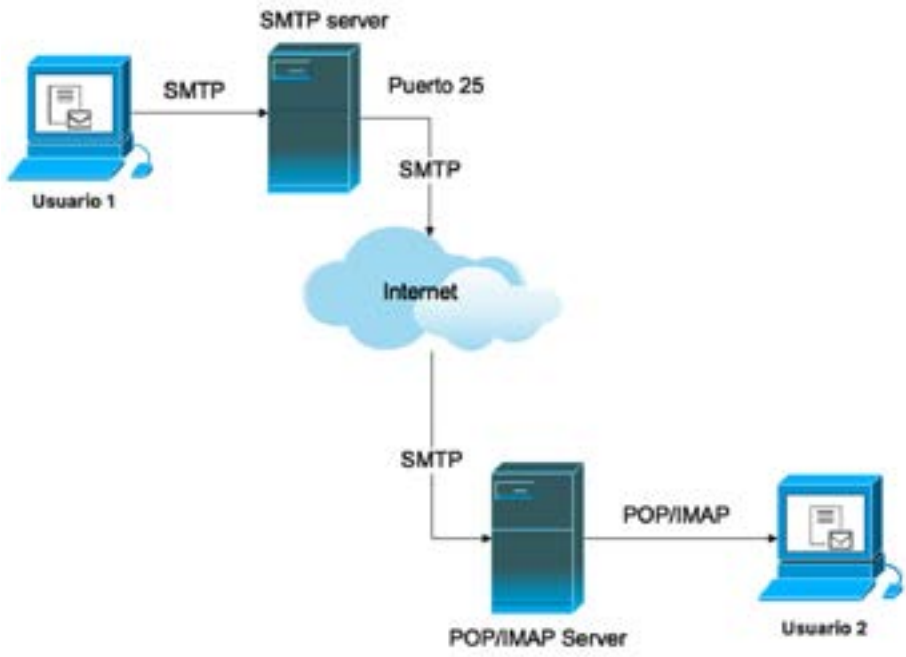
Tipo	Ejemplo
Text	Texto plano, html, richtext
Image	Jpeg, png, gif
Audio	Basic, 32kadpcm, mp3, wav
Video	Avi, mpeg, quicktime
Application	Msword, postscript, octet-stream

3.1.2. Protocolo SMTP

El Protocolo SMTP por sus siglas en inglés *Simple Mail Transfer Protocol* está definido en RFC 2821, y es un protocolo para el intercambio de mensajes de correo electrónico que usa TCP para transferir datos de correo de un cliente a un servidor, a través del puerto 25 (ver Figura 23). SMTP funciona a través de tres fases plenamente identificadas:

- Establecimiento (saludo)
- Transferencia de mensajes
- Cierre

Figura 23.
Funcionamiento SMTP



Continuemos con el aprendizaje con el tema SMTP.

Actividad de aprendizaje recomendada

Le invitamos a revisar otros protocolos y compararlos con el desempeño de SMTP, considerando que SMTP no es el único protocolo que se utiliza para transferir correo.

3.1.3. POP

El protocolo SMTP fue pensado para transferir mensajes de correo entre grandes computadores que tenían múltiples usuarios; sin embargo, con la masificación de las computadoras personales se hizo evidente la necesidad de tener un protocolo que permita a estos equipos conectarse y que acceder a las aplicaciones de correo, este es el contexto en el cual aparece POP en 1984.

Desde ese año a la actualidad se han tenido varias versiones del protocolo: POP1, POP2, la que actualmente se tiene en vigencia es POP3 que permite no solo recuperar/leer mensajes de correo, almacenarlos en el servidor y borrarlos; sino que además permite realizar acciones de gestión de correo.

Si revisamos la sección 3.1 de esta guía, podemos ver que se definieron dos agentes: agente de usuario MUA y Agente de transporte MTA; con estos nuevos conceptos identificamos que el primero corresponde a POP3 y el segundo enteramente a SMTP.

POP3 (Post Office Protocolo) está definido en RFC 1939 y utiliza conexión TCP en el puerto 110 del servidor de correo. En la Figura 24 se muestra la configuración de POP3 en Gmail.

Figura 24.
Configuración POP3

Le invito a complementar sus conocimientos sobre POP

3.1.4. IMAP

El IMAP (*Internet Message Access Protocol*) es una evolución POP3 y se referencia a RFC 1064. Este protocolo se ejecuta en el puerto 143 sobre TCP.

La característica diferenciadora con su predecesor es que IMAP, permite a los usuarios tener múltiples buzones de correo.

Entre las principales ventajas comparativas a POP3 tenemos:

- El servidor guarda información de estado de todos los correos dentro del buzón.
- Se pueden clasificar los correos dentro del buzón.
- Se puede realizar búsquedas dentro del buzón.
- Las sesiones IMAP pueden durar más tiempo.

3.1.4.1. Modos de operación IMAP

En el protocolo IMAP se permite tres modos de operación:

1. En línea (*online*): el cliente está en contacto permanente con el servidor.
2. Fuera de línea (*offline*): el cliente accede al servidor solo cuando debe manipular su buzón.
3. Desconectado (*disconnected*): es un híbrido entre los dos anteriores.

Actividad de aprendizaje recomendada

Como parte de ir mejorando en sus conocimientos, le sugerimos experimente con el agente de correo Outlook y realice dos pruebas, una con POP3 y otra con IMAP; evalúe las diferencias de ambos.

3.2. Servicio de directorio de Internet DNS

En esta sección abordaremos a una de las aplicaciones más funcionales dentro de Internet, se trata de DNS, para ello es necesario que usted lea detenidamente la sección 2.4 del texto básico.

También debe revisar el vídeo del canal de YouTube, NZNetwork, Qué es DNS y cuál es su funcionalidad?. En este video tutorial aprenderás a conectarte a conocer como un DNS funciona dentro de la red.

Conozca más sobre DNS accediendo al recurso:

Servicio de directorio de Internet DNS

Actividad de aprendizaje recomendada

En las secciones anteriores se realizaron prácticas con Wireshark, en esta oportunidad incluiremos una nueva herramienta *nslookup* que sirve para capturar consultas DNS. Le recomendamos seguir los pasos citados a continuación:

1. Ingrese al programa Wireshark.

2. En el menú superior vamos a la opción capture, y en interfaces seleccionamos nuestra dirección IP presionando el botón *start*; con ello se iniciará una escucha de nuestra interfaz de red para posteriormente capturar los paquetes que se envíen a través de ella.

3. Ejecute en un terminal o *prompt* la siguiente línea de comandos: nslookup www.utpl.edu.ec. En su terminal o prompt se verá la información correspondiente a la resolución del nombre de dominio. Si quiere más información puede ejecutar nslookup -type-NS utpl.edu.ec.

4. Regresemos al programa Wireshark, esperemos dos minutos y paremos la captura. Identifique las entradas del protocolo DNS y realice las siguientes actividades:

a. Revise qué puertos se utilizan para la ejecución de consultas DNS.
b. Revise las entradas standard query, ¿a qué se refieren estas entradas?
c. ¿Qué detalles puede revisar en las tramas correspondientes?

Le invitamos a resolver las siguientes preguntas que le servirán para determinar las áreas en las cuales ha tenido mayor dificultad y necesitan una nueva revisión.

Autoevaluación 3

Lea detenidamente y responda según corresponda:

1. ¿Qué protocolos se pueden utilizar para recoger un buzón de correo?

   a. POP3
   b. IMAP
   c. HTTP
   d. Todas las anteriores

2. ¿Qué acción ejecuta el comando HELO en SMTP?

   a. Muestra el estado del buzón
   b. Indica que los siguientes datos son los que se transmitirán
   c. Confirma el dominio de correo
   d. Establece el remitente del mensaje

3. ¿Qué comando inicia la fase de actualización en el protocolo POP3?

   a. START
   b. RESTART
   c. UPDATE
   d. QUIT

4. ¿Cuál de los siguientes protocolos permite organizar los correos en el buzón?

   a. SMTP
   b. POP3
   c. IMAP
   d. Todos los anteriores

5. ¿Qué puerto utiliza DNS para recibir solicitudes?

   a. TCP 53
   b. UDP 53
   c. TCP 17
   d. UDP 17

6. ¿Qué es TLC en el protocolo DNS?

   a. Servidores de dominio de primer nivel
   b. Servidores de dominio de segundo nivel
   c. Servidores de dominio principal
   d. Servidores de dominio jerarquizados

7. ¿Qué tipo de consultas DNS realiza habitualmente un *host*?

   a. Interactivas
   b. Recursivas
   c. Insistentes
   d. Programadas

8. Teniendo los siguientes URL: www.utpl.edu.ec/personal.htm y http://www.arcotel.gob.ec/personal.htm. ¿Cuál de las siguientes es la afirmación correcta?

   a. Direccionan a la misma página “personal.htm”
   b. Tienen el mismo dominio
   c. Ambas se alojan en el mismo servidor
   d. Se alojan en servidores distintos

9. ¿Cuántos *bytes* tiene la cabecera de mensajes DNS?

   a. 8 bytes
   b. 12 bytes
   c. 20 bytes
   d. 32 bytes

10. ¿Qué es un registrador?

   a. Entidad que maneja el registro único de dominios.
   b. Entidad que maneja las direcciones IP únicas para servidores.
   c. Entidad que ofrece servicios de *hosting*.
   d. Ninguna de las anteriores.

Ir al solucionario

Ahora a reforzar lo aprendido con la resolución de ejercicios

Se ha seleccionado los siguientes problemas propuestos en el texto, estos ejemplifican desde muchos aspectos lo que pretendemos usted conozca sobre esta Unidad, por favor, desarróllelos.

Problemas: P13, P17, P20 y P21 del capítulo 2 texto básico.

Semana 4

Unidad 4. Capa de aplicación y servicios especiales

En las unidades anteriores hemos analizado aplicaciones que utilizan el modelo cliente-servidor, a continuación, revisaremos aplicaciones que salen de este esquema y presentan nuevas maneras de intercambiar información en arquitecturas de redes heterogéneas.

Para la revisión de los siguientes puntos, por favor, lea primero el texto básico secciones 2.5 a 2.7; y luego la guía, puesto que solamente se aclaran aspectos puntuales.

4.1. Aplicaciones P2P

Las aplicaciones P2P (peer-to-peer) son servicios descentralizados y distribuidos; donde la información no se almacena en un servidor central sino en varios servidores tratados como iguales.

El primer servicio creado en este esquema fue Napster, con el fin de distribuir archivos de música. Esta filosofía de mercado global se

masificó hasta nuestros tiempos con los objetivos de colaboración e información global distribuida.

Las aplicaciones P2P son muy populares porque tienen varias ventajas en comparación con el modelo cliente-servidor; por ejemplo:

- No se tiene un servidor central que controle las transacciones.
- Sin servidores, los clientes pueden comunicarse directamente entre sí.
- Cada cliente se convierte en un nodo por sí mismo; esto tiene ventajas derivadas:

    a. Se puede realizar un balance de carga efectivo y mejorar la eficiencia de los recursos.

    b. Se puede optimizar el uso de las redes a través de la selección inteligente del tráfico.

Actividad de aprendizaje recomendada

Estimado estudiante, le sugerimos que revise el documento en formato pdf: "A survey and comparison of Peer-to-Peer Overlay Network Schemes", disponible en: P2P Esquemas de Red Superpuestos; en este documento se realiza un análisis de las ventajas de P2P en relación a otros esquemas, así como varias estructuras P2P. Haga un resumen de estos puntos para complementar sus conocimientos.

Antes de continuar con este tema, lo invitamos a ver el vídeo del canal de YouTube, Josias Castro, Cliente - Servidor y Peer-2-Peer. En

este video tutorial aprenderás a diferenciar la arquitectura P2P de la arquitectura cliente-servidor. En P2P, todos los nodos tienen los mismos beneficios. Por lo que si un nodo falla quedarían los otros activos y así siempre se garantizaría la transferencia del archivo, mientras que en un sistema cliente – servidor, si el servidor está caído las peticiones de los clientes no se ejecutarían.

4.1.1. Arquitectura P2P

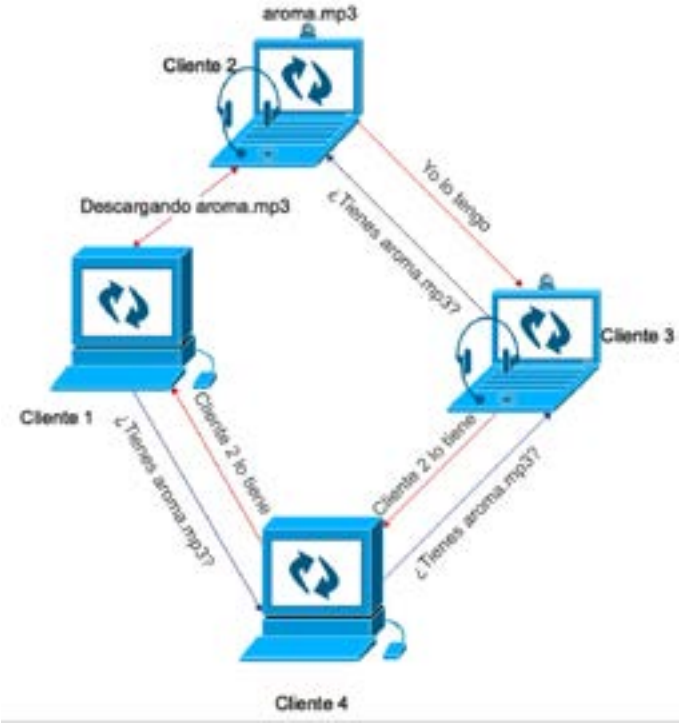
Como se referencia en la sección 2.5 del texto básico, P2P tienen dos arquitecturas: pura e híbrida.

4.1.1.1. Esquema puro o totalmente descentralizado

En una red P2P pura todos los participantes son iguales y cada nodo puede cumplir con tres funciones:

- Cliente: es la entidad que solicita información a un par.
- Servidor: es la entidad que brinda información de otro.
- Enrutador: es la entidad que sirve de intermediario entre otros dos.

Figura 25.
Arquitectura P2P puro



La localización de pares se realiza automáticamente por la aplicación P2P sin intervención de un servidor (ver Figura 25).

Gnutella y Freenet son ejemplos de esta arquitectura P2P. Por un lado, Gnutella es una gran red peer-to-peer. Fue la primera red peer-to-peer descentralizada de este tipo, lo que llevó a otras redes posteriores a adoptar el modelo. Este 14 de marzo de 2020 celebró dos décadas de existencia y tiene una base de millones de usuarios para el intercambio de archivos de igual a igual, así mismo Freenet es un *software* gratuito que permite compartir archivos de forma anónima, navegar y publicar "sitios gratuitos" (sitios web accesibles solo a través de Freenet) y chatear en foros, sin temor a la censura.

Freenet está descentralizado para hacerlo menos vulnerable a los ataques, y si se usa en el modo "darknet", donde los usuarios solo se conectan con sus amigos, es muy difícil de detectar.

4.1.1.2. Esquema híbrido o centralizado

Una red P2P híbrida cuenta con un servidor central que realiza funciones administrativas con el fin de facilitar los servicios entre pares. El nodo realiza una consulta previa al servidor, para luego poder establecer una conexión con su par. Es necesario que los nodos mantengan informado al servidor de las conexiones y desconexiones que realicen.

Figura 26.
Arquitectura P2P híbrido servidor de referencia

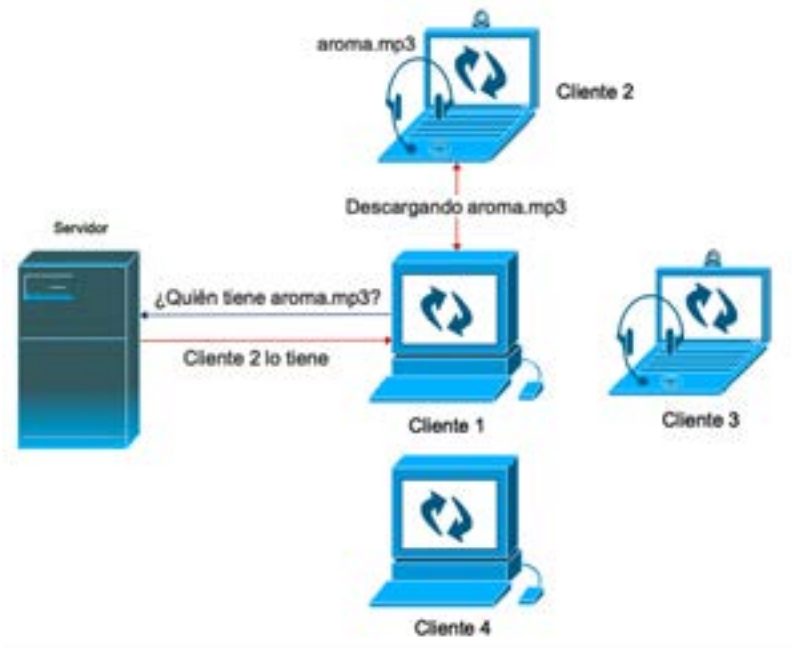
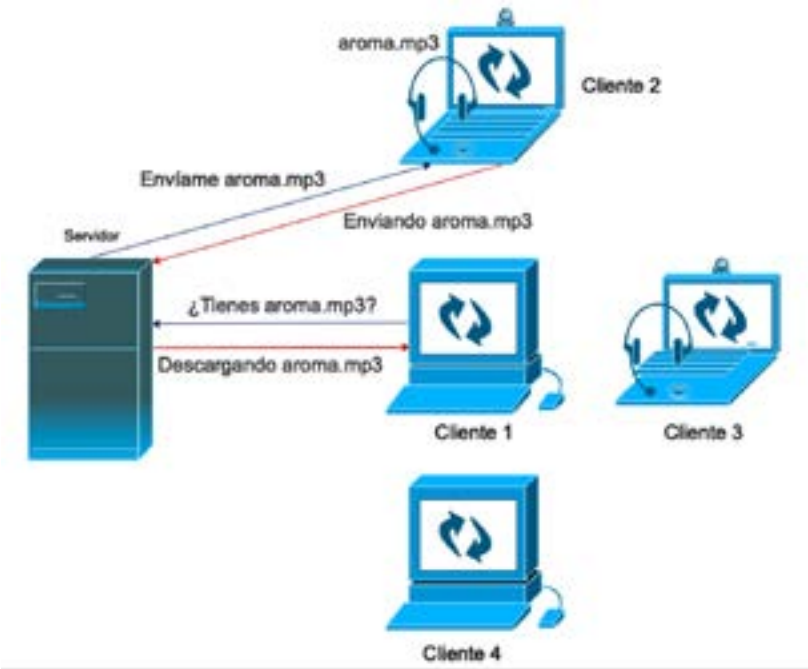


Figura 27.
Arquitectura P2P híbrido servidor de contenido



Ejemplos de esta arquitectura son: Napster y BitTorrent.

En este esquema se pueden tener dos casos: servidor de localización para nodos y operaciones de búsqueda de recursos; y; servidor de localización para nodos y operaciones y para repositorio de contenido de recursos, en las Figura 26 y Figura 27 se pueden referenciar las dos arquitecturas.

4.1.2. Elementos P2P

Las redes P2P tienen tres elementos fundamentales: pares, grupos de pares y servicios.

La unidad fundamental en aplicaciones P2P es el nodo o par; puesto que es quien realiza el procesamiento de la aplicación en sí. Existen dos tipos de pares:

1. *Pares simples*: se deben a un único usuario final, lo que le permite ofrecer servicios desde este dispositivo y al mismo tiempo servirse de los ofertados por otros pares de la red. Los pares simples son de naturaleza dinámica y heterogénea.

2. *Superpares*: asisten a los pares simples para que puedan encontrar otros pares o recursos de otros pares. Los superpares son de naturaleza estática.

Los pares pueden organizarse en grupo de pares que se reúnen para conseguir un objetivo común.

Todos los elementos que hemos mencionado hasta ahora buscan servicios. Los servicios ofrecen funcionalidades o información útil a través de la comunicación de pares, por ejemplo: transferir archivos, obtener información o simplemente comunicarse con otro usuario.

4.1.3. Tablas hash distribuidas

En la sección 7.5.4, página 647 del libro complementario (Tanenbaum & Wetherall, 2012), se introduce el tema de Tablas de *hash* distribuidas, por ello es necesario que lea detenidamente la sección.

Las Tablas *hash* distribuidas o DHT por sus siglas en inglés son esenciales para el funcionamiento de P2P.

Las DHT realiza dos funciones: almacena el par valor y clave en la Tabla *hash*, y dada una clave busca su valor, sin embargo, esto sea distribuidamente a múltiples máquinas.

Actividad de aprendizaje recomendada

Estimado estudiante, le sugerimos revise aplicaciones P2P como por ejemplo eMule, BitTorrrent, Ares, Vuze, uTorrent, Soulseek, entre otros. Haga un análisis comparativo de las ventajas y desventajas que tienen cada una de estas aplicaciones.

4.1.4. Vulnerabilidades de P2P

Dentro de las vulnerabilidades de P2P se puede mencionar:

- Las aplicaciones P2P abren demasiados puertos a la vez para establecer comunicación.
- Es muy fácil disipar virus y demás programas maliciosos por redes P2P que tiene como lógica la cooperación desinteresada.
- Un problema serio que tienen las redes P2P es la violación de derechos de autor y propiedad intelectual.
- Se han detectado varias fallas de seguridad en algunas aplicaciones P2P, en la actualidad se trabaja para corregirlas.

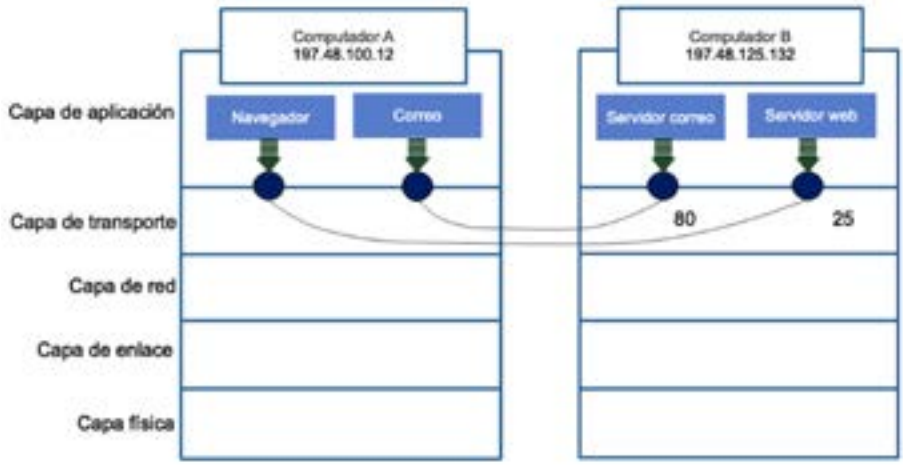
4.2. Programación de sockets

La programación de *sockets* es bastante interesante para desarrollar aplicaciones para Internet propietarias, usted puede incluso servirse de este tipo de aplicaciones para montar herramientas particulares. Le pedimos analice previamente la sección 2.7 del texto básico.

Iniciemos primeramente por la definición de un socket; frecuentemente hemos escuchado este término, pero no tenemos una definición clara de ello. Un socket es un punto final de un enlace de comunicación entre dos programas. El socket no es una unidad física, sino que es un objeto de *software* que conecta las aplicaciones a un protocolo de redes para transmitir datos y es controlado por el sistema operativo.

Una comunicación a través de sockets es una comunicación de dos vías; por ejemplo, cuando revisamos HTTP veíamos que era necesario enviar solicitudes y recibir respuesta para que el protocolo sea eficiente; en este caso HTTP abrirá un *socket* en el cual podrá escribir y leer datos, respectivamente. De la misma forma todos los sockets se asocian a un puerto dado, pues esto facilita que el protocolo de transporte pueda discriminar entre aplicaciones; siguiendo el mismo ejemplo para HTTP, el puerto asociado es 80 en TCP.

Figura 28.
Identificación de un socket



Para los protocolos TCP y UDP, un *socket* se identifica con la combinación de una dirección IP y un número de puerto, por ejemplo, en la Figura 28 visualizamos la interconexión desde un computador A con dirección 197.48.100.12 al computador B con dirección 197.48.125.132 con puertos 80 y 25 para aplicaciones *web* y de correo respectivamente.

Este concepto de *socket* nace con los Unix Domain Sockets que eran accesibles solo en el sistema Unix. Este tipo de sockets tienen la apariencia de un archivo común dentro del sistema de ficheros de Unix, sin embargo, se los puede identificar porque cuando se los lista (ls -l) se marcan con una "s" en la primera columna o columna del tipo de archivo.

En la capa de transporte existen dos tipos de socket: Sockets TCP y Sockets UDP. Y para el paquete java.net se tiene tres clases, que se muestran en la Tabla 16.

Tabla 16.
Tipos de Sockets del paquete java.net

Socket	Cliente	TCP
ServerSocket	Servidor	TCP
DatagramSocket	Cliente/Servidor	UDP

4.2.1. Sockets con TCP

Los sockets TCP o tipo Stream son orientados a la conexión, tienen una comunicación fiable y ordenada.

Los sockets TCP abren sesiones de comunicación en base al modelo cliente-servidor, y se ejecuta un proceso controlado de la siguiente forma (ver Figura 29):

1. El cliente crea un *socket* (sclient) acorde a la aplicación que lo requiere.

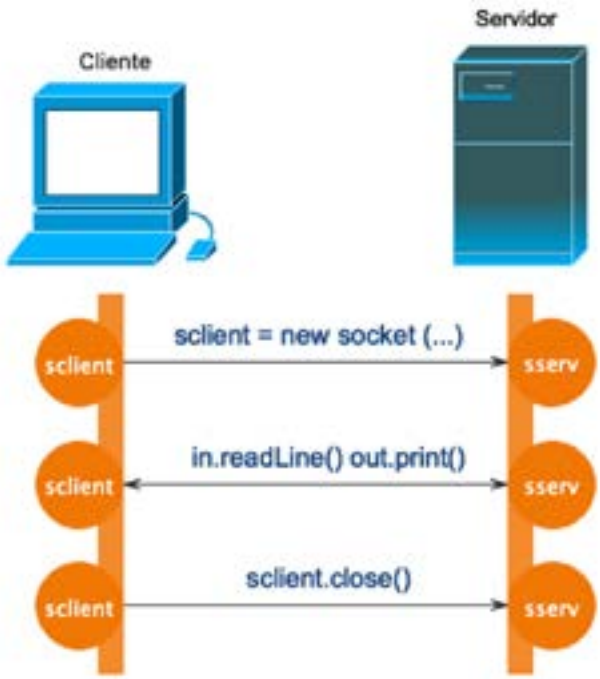
2. El servidor dispone un *socket* (sserv) para responder al cliente.

3. El cliente conecta su *socket* al del servidor.

4. Se realiza el intercambio de información entre cliente y servidor.

5. El cliente o el servidor cierra la conexión, pero cada uno se asegura de cerrar su *socket*.

Figura 29.
Tabla de ruteo de un router marca CISCO



4.2.2. Sockets con UDP

Los sockets UDP o tipo Datagram no son orientados a la conexión y se transfieren en bloques de datos; lo que es conveniente para poder realizar difusiones, sin embargo, no es una conexión fiable.

Cuando se utilizan *sockets* con UDP el emisor debe indicar explícitamente el identificador en cada datagrama.

Actividad de aprendizaje recomendada

Estimado estudiante, sería muy conveniente que se relacione con la programación de *sockets*, por eso le pedimos que dé lectura a la sección 2.7 del texto básico e intente aplicar programación básica en Java.

En esta Unidad nuestra preocupación ha sido describir algunas redes tipo, esperamos que usted haya tenido mucho éxito con este proceso; sin embargo, es necesario tratar de definir qué apartados necesitan una segunda revisión por su parte; por ello le planteamos el siguiente cuestionario como autoevaluación, recuerde que el solucionario de este se encuentra al final de la guía.

Autoevaluación 4

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

1. ¿Cuál de las siguientes características pertenece a UDP?

   a. Servicio orientado a conexión
   b. Permite difusiones
   c. Servicio fiable y ordenado
   d. Realiza control de flujo

2. ¿Qué rango de puertos utilizan los clientes?

   a. 0000-1024
   b. 1024-2048
   c. 0000-65535
   d. 1024-65535

3. ¿Según Java, cuáles son los tipos de socket?

   a. Client, server y datagram
   b. Client y server
   c. Client, server y conection
   d. Cliente, server y port

4. ¿Cuál de las siguientes afirmaciones es cierta?

   a. El cliente puede abrir dos sockets sobre el mismo puerto local y del mismo protocolo.
   b. El cliente puede abrir dos *sockets* sobre el mismo puerto local y de diferentes protocolos.
   c. El servidor puede tener dos *sockets* con el mismo puerto y del mismo protocolo.
   d. El servidor puede tener dos *sockets* con el mismo puerto y diferentes protocolos.

5. El establecimiento de conexión comienza con:

   a. Abriendo un par de *sockets*
   b. Acuerdo de tres vías
   c. Llamando a la aplicación en el servidor
   d. Ninguna de las anteriores

6. TCP es un protocolo:

   a. Best effort
   b. Fiable
   c. Se encarga del acceso al medio
   d. Crea sockets

7. El identificador de *sockets* consta de:

   a. Dirección IP e identificador de aplicación
   b. Dirección IP y Dirección MAC.
   c. Dirección MAC y número de puerto.
   d. Dirección IP y número de puerto.

8. Los primeros sockets fueron:

   a. Internet socket
   b. Domain sockets
   c. Unix sockets
   d. Network sockets

9. ¿Qué es el tiempo de distribución?

   a. Tiempo que demora cada par en tener una copia del archivo.
   b. Tiempo que demora un servidor para enviar copia del archivo.
   c. Tiempo que demora un *router* en actualizar sus rutas.
   d. Tiempo que demora un cliente en obtener una copia del archivo.

10. Cuando se tienen DHT circular se habla de:

    a. Red distribuida
    b. Red solapada
    c. Red centralizada
    d. Red unificada

Ir al solucionario

Ahora le animamos a desarrollar los siguientes ejercicios:

Con el propósito de reforzar el nivel de conocimientos de la presente unidad resuelva los siguientes ejercicios propuestos en el texto básico.

Problemas: P22, P23, P30 y P31 del capítulo 2 texto básico.

Resultado de aprendizaje 3

- Esquematiza estrategias de seguridad básica en redes de computadoras.

Contenidos, recursos y actividades de aprendizaje

Semana 5

Unidad 5. Seguridad de redes

En esta Unidad encontrará los conocimientos relacionados a seguridad de redes que son muy importantes en una gestión de red. Al finalizar esta unidad comprenderá la importancia e identificará vulnerabilidades y los diferentes mecanismos para tener una red segura. Le invitamos a que iniciemos con optimismo el desarrollo de esta unidad abordando temas sobre criptografía, integridad y autenticación, conexiones seguras, seguridad de redes LAN inalámbricas y seguridad operacional. En primer lugar, es necesario señalar algunas definiciones básicas para el desarrollo de los temas.

5.1. Introducción

Antes de introducirnos a la seguridad de redes le invito, estimado estudiante, a revisar algunos eventos pasados de ataques a redes muy importantes, realizadas en el mundo y recientes en el Ecuador, para entender la importancia de la seguridad de redes. Los Gobiernos, empresas y personas siempre han buscado la forma de proteger su información implementando e investigando nuevos mecanismos de seguridad, invirtiendo gran cantidad de recursos económicos. Pero existen personas que haciendo uso de sus habilidades y herramientas que disponen encuentran vulnerabilidades a sistemas seguros.

5.2. Definición

Le invitamos a desarrollar la lectura del capítulo 8, específicamente la sección 8.1: "¿Qué es la seguridad de red?" del texto básico, donde se realiza una introducción a la seguridad de redes. Para complementar sus conocimientos analice el siguiente vídeo: Seguridad a nivel de red local (segmentación, LAN, WiFi).

Realizada la lectura recomendada y el video; usted ya tiene las nociones fundamentales de seguridad de redes necesarias para la comprensión de los siguientes temas a estudiar.

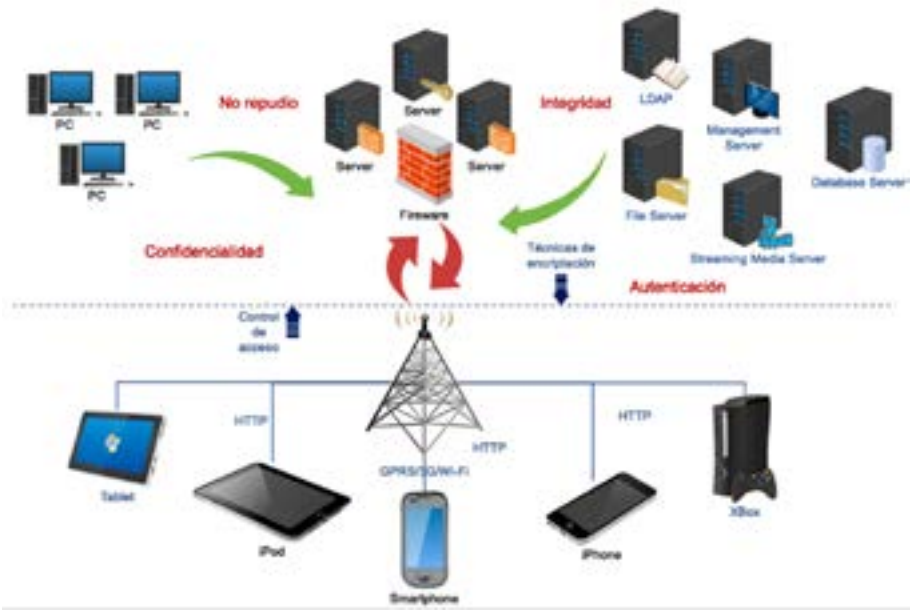
Con la lectura de esta sección y la observación del vídeo conviene que ponga énfasis en:

- Definición de seguridad de red.
- Propiedades deseables en una comunicación segura.
- Desventajas de un sistema de comunicación segura.

- La importancia de una red segura.
- Saber que los ataques también pueden ser desde dentro de la red LAN.

Como ya conoce, la seguridad de redes es primordial para evitar ataques y violaciones a los datos. Actualmente las empresas invierten más recursos en la seguridad de la información y en la integridad de la red. La importancia de la seguridad de redes tiene beneficios tanto para los usuarios de la red como para los proveedores de *hardware* y *software* mediante la aplicación de modelos y estándares.

Figura 30.
Propiedades de una red segura



En la Figura 30, se presenta las 4 propiedades más importantes que se debe tener una red para que esta sea segura, le invito a pensar cómo se deben aplicar estas propiedades en una red.

Para continuar le invitamos a revisar el siguiente recurso abierto:

Le invitamos a dar lectura al documento "Seguridad en TCP/IP", en el enlace: https://www.rediris.es/cert/doc/segtcpip/Seguridad_en_TCP-IP_Ed1.html

Basados en el recurso anterior debemos recalcar que la seguridad de redes involucra todas las capas del modelo de Internet. A continuación, señalamos una breve explicación de las vulnerabilidades de cada capa:

- Capa física: infraestructura física de la red.
- Capa de enlace: las debilidades de esta capa están asociadas al medio por donde se realizan las conexiones.
- Capa de red: en esta se pueden realizar cualquier ataque al datagrama IP.
- Capa de transporte: en esta capa se pueden encontrar problemas de autenticación, de integridad y de confidencialidad.
- Capa de aplicación: debido al gran número de protocolos definidos en esta capa, la cantidad de deficiencias presentes también será superior al resto de capas.

Después de realizar una introducción a la seguridad de redes e identificar su importancia y las vulnerabilidades de las redes, le invitamos a continuar con el mismo optimismo al estudiar las siguientes temáticas relacionadas a diferentes mecanismos de seguridad de una red.

Actividad de aprendizaje recomendada

Identifique 2 razones de por qué tener una red segura. Además, revise eventos importantes recientes en Internet de ataques a redes.

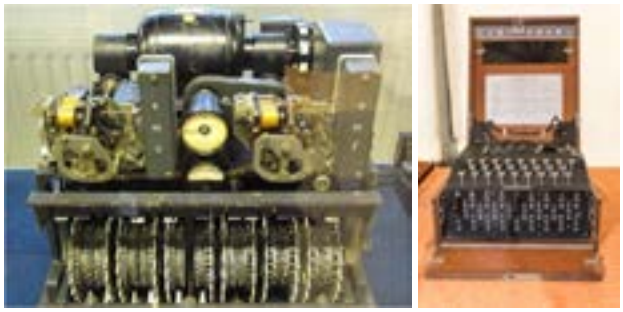
5.3. Principios de criptografía

Previo a continuar con este tema lo invitamos a revisar la sección 8.2 del texto básico: "Principios de la Criptografía", donde se especifica qué es la criptografía y se detalla algunas técnicas.

Una vez realizada la lectura recomendada, usted ya comprende la importancia de la criptografía y que no es un tema de reciente aparición, y que su origen es militar, teniendo conocimiento de que el emperador romano Julio César empleó una técnica de ocultamiento de mensajes para ser enviados. Adicionalmente, en la Figura 31 se presenta dos máquinas alemanas empleadas durante la segunda guerra mundial, con el propósito de enviar mensajes y órdenes sin que el enemigo lo entendiera, aunque interceptara el mensaje.

De los antecedentes anteriores se puede decir que la criptografía se originó como la ciencia de ocultar mensajes; es decir, encriptar para que únicamente el receptor pueda acceder a estos.

Figura 31.
Máquinas empleadas para criptografía: a) La máquina alemana de cifrado Lorenz, usada en la Segunda Guerra Mundial para el cifrado de los mensajes para los generales de muy alto rango. b) La máquina Enigma utilizada por los alemanes durante la II Guerra Mundial (Qwerty, 2021)



a) b)

Nota: a) La máquina alemana de cifrado Lorenz, usada en la Segunda Guerra Mundial para el cifrado de los mensajes para los generales de muy alto rango. b) La máquina Enigma utilizada por los alemanes durante la II Guerra Mundial (Qwerty, 2021)
Fuente: Giorgio Rossi|shutterstock.com

Adicionalmente, mencionar que todos los algoritmos criptográficos implican cambiar una cosa por otra. Estos se pueden resumir en dos métodos que a continuación hacemos mención:

- **Criptografía de clave simétrica**: esta técnica emplea la misma clave para cifrar y descifrar un mensaje. Entonces el principal problema de seguridad reside en el intercambio de claves entre el emisor y el receptor, ya que ambos deben usar la misma clave. Por lo tanto, se tiene que buscar un canal de comunicación que sea seguro para el intercambio de la clave.

- **Cifrado de clave pública**: es también conocida como criptografía de clave asimétrica, esta funciona con en el uso de una pareja de claves, pública y privada, de las cuales una se usa para cifrar y la otra para descifrar respectivamente.

Actividad de aprendizaje recomendada

Investigue si en el Ecuador existe una ley para la utilización de firmas digitales y de cómo puede hacer uso de una firma digital.

5.4. Integridad de los mensajes y autenticación

Le invitamos a desarrollar la lectura del capítulo 8, específicamente en las secciones 8.3 y 8.4: "Integridad de los mensajes y autenticación del punto terminal" del texto básico.

Después de haber realizado la lectura, conoció algunos mecanismos para brindar integridad a los mensajes o conocida también como autenticación de mensajes. A continuación, revisemos algunas características:

- *Funciones hash criptográficas*: estas son funciones que convierten una cadena de longitud arbitraria de un mensaje en una cadena de longitud fija.
- *Código de autenticación de mensaje*: también conocido como MAC (Message Authentication Code), está técnica emplea un fragmento de información para obtener una clave de autenticación, que se utiliza para autenticar un mensaje.
- *Firmas digitales*: esta es similar a las firmas que realizamos para certificar un determinado documento. Recuerde que esta firma tiene algunas características que nos permiten identificar y que no se puedan reproducir fácilmente por alguien más. En el mundo digital se emplea la firma digital para que los documentos enviados de forma digital tengan la misma validez que un documento firmado a mano. En

resumen, este es un método criptográfico que asocia una identidad, ya sea de una persona en particular o de un equipo a un mensaje enviado a través de la red, y es el resultado de aplicar a un documento en línea, un procedimiento matemático que requiere datos que exclusivamente conoce la persona que firma.

- *Autenticación del punto terminal*: este es un proceso para demostrar a alguien su propia identidad.

Actividad de aprendizaje recomendada

1. Revisar las aplicaciones que se encuentran en el siguiente curso abierto (OCW) de Seguridad de Redes de Comunicaciones, de la Universidad Politécnica de Cartagena. Para lo cual deberá descargar y descomprimir el curso completo. En él encontrará información relevante, prácticas y algunas aplicaciones de cifrado en flujo y cifrado en bloques que le permitirán entender la temática de criptografía.

2. Descargar el archivo: Herramienta Docente. Una aplicación que implementa una función Hash y su ataque por fuerza bruta Birthday Paradox, así como el archivo RFC 3174 – US Secure Hash Algoritm 1 (SHA1). Analizar cómo se aplican las funciones hash criptográficas.

Revisar el vídeo del canal de YouTube, denominado Hashing: introducción, ejemplos y funciones criptográficas, donde se explican los conceptos relacionados a las funciones *hash*.

Le invitamos a resolver las siguientes preguntas, que le servirán para determinar las áreas en las cuales ha tenido mayor dificultad y necesitan una nueva revisión.

Autoevaluación 5

Lea detenidamente cada enunciado y conteste con (V) en el caso de ser verdadero y con (F) si es falso:

1. ( ) La seguridad de redes tiene beneficios tanto para los usuarios de la red, como para los proveedores de *hardware* y *software* mediante la aplicación de modelos y estándares.

2. ( ) La autenticación es una propiedad para tener una red segura.

3. ( ) En la capa de enlace se pueden realizar cualquier ataque al datagrama IP.

4. ( ) En la capa de aplicación se pueden encontrar problemas de autenticación, de integridad y de confidencialidad.

5. ( ) El emperador romano Julio César empleó una técnica de ocultamiento de mensajes para ser enviados.

6. ( ) La criptografía se originó como la ciencia de visualizar mensajes; es decir, encriptar para que únicamente el receptor pueda acceder a estos.

7. ( ) La máquina alemana de cifrado Lorenz, usada en la Segunda Guerra Mundial para el cifrado de los mensajes para los generales de muy alto rango.

8. ( ) La seguridad de redes es primordial para evitar ataques y violaciones a los datos.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. La criptografía de clave simétrica:

   a. Emplea la misma clave para cifrar y descifrar un mensaje.
   b. Emplea una clave para cifrar y otra para descifrar un mensaje.
   c. Emplea una clave para cifrar y para descifrar un mensaje ninguno.
   d. No emplea una clave para cifrar, pero para descifrar un mensaje sí, la cual es enviada por medio de un canal seguro.

10. ¿Cómo también es conocido el cifrado de clave pública?

   a. Criptografía de clave simétrica.
   b. Criptografía de clave asimétrica.
   c. Criptografía de clave Isométrica
   d. Ninguno de los anteriores.

Ir al solucionario

Semana 6

Unidad 6. Seguridad de redes

Estimado estudiante, a continuación, le invitamos a desarrollar la lectura del capítulo 8, específicamente las secciones 8.5 del texto básico: "Asegurando correo electrónico" y del enlace: Seguridad de las aplicaciones.

6.1. Aplicaciones seguras

La seguridad de las aplicaciones se refiere a las medidas de seguridad, a nivel de aplicación, cuyo propósito es impedir el robo o el secuestro de datos o códigos dentro de la aplicación. Abarca las consideraciones de seguridad que se deben tener en cuenta al desarrollar y diseñar aplicaciones, además de los sistemas y los enfoques para proteger las aplicaciones después de distribuirlas.

Recordemos brevemente algunas aplicaciones seguras (Bermudez, 2018a) (Bermudez, 2018b):

- SSH (Secure Shell): es una aplicación que define un protocolo propio para la transmisión segura de la información, está

diseñada para substituir determinadas herramientas de acceso remoto; han sido usadas tradicionalmente en los sistemas Unix, como rsh (Remote Shell), rlogin (Remote Login) o rcp (Remote Copy), por nuevas versiones con servicios de seguridad.

- Correo electrónico seguro: emplea métodos para proteger el correo electrónico en el mismo nivel de aplicación, independientemente del sistema de transporte utilizado. La idea es aplicar las funciones criptográficas necesarias al mensaje antes de entregarlo a los agentes de transferencia del servicio de correo, y estos solo deben hacerlo llegar a su destino de forma habitual.

6.2. Seguridad de la capa de red: IPsec y redes privadas virtuales

Antes de continuar, desarrolle la siguiente lectura del capítulo 8, específicamente la sección 8.6: "Seguridad de la capa de red: IPSEC y redes privadas virtuales" del texto básico, donde se revisa la importancia de IPsec y las redes privadas virtuales.

Como usted ya debe conocer, IPsec, conocido como protocolo de seguridad IP, es una extensión al protocolo IP que proporciona seguridad a este y a los protocolos de capas superiores. Este fue desarrollado para el nuevo estándar IPv6 y después fue portado o aplicado a IPv4. La arquitectura IPsec se describe en el RFC2401. Una aplicación de IPsec muy importante es la creación de redes privadas virtuales conocidas como VPN, que funcionan sobre la red de Internet.

IPsec emplea dos protocolos diferentes el AH (Authentication Header) y ESP (Encapsulation Security Payload) para asegurar la autenticación, integridad y confidencialidad de la comunicación.

Puede proteger el datagrama IP completo o solo los protocolos de capas superiores. Estos modos se denominan, respectivamente, modo túnel y modo transporte.

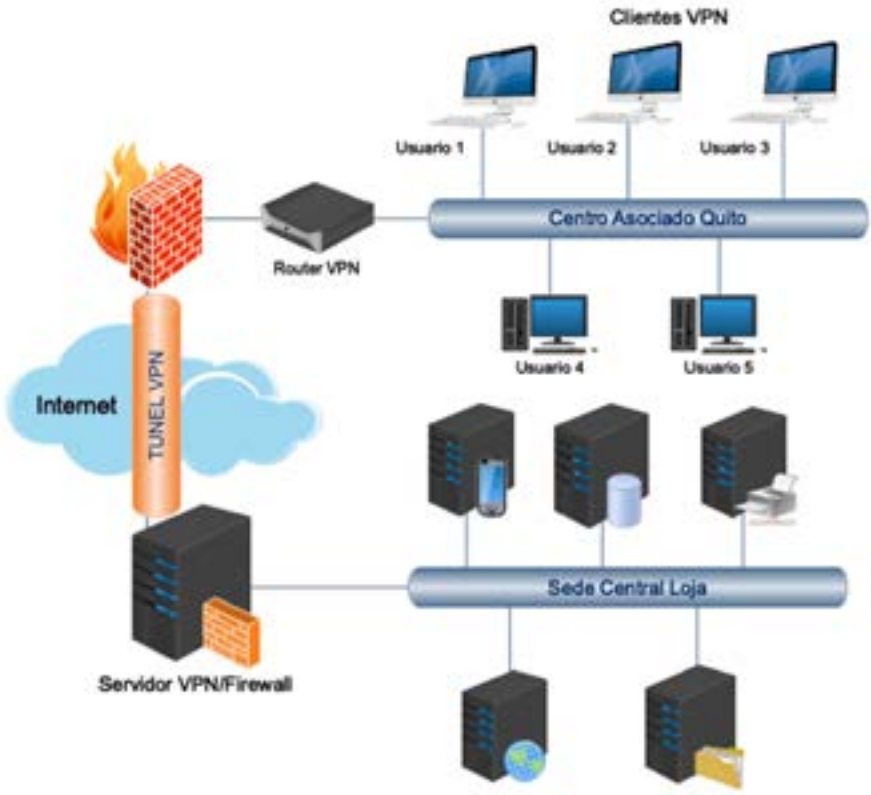
- En modo túnel el datagrama IP se encapsula completamente dentro de un nuevo datagrama IP que emplea el protocolo IPsec.
- En modo transporte IPsec solo maneja la carga del datagrama IP, insertándose la cabecera IPsec entre la cabecera IP y la cabecera del protocolo de capas superiores.

6.2.1. VPN (Virtual Private Network)

Una VPN no es más que una red privada que se extiende, mediante un proceso de encapsulación y en algún caso de encriptación, desde los paquetes de datos a diferentes puntos remotos, mediante el uso de infraestructuras públicas de transporte. Los paquetes de datos de la red privada viajan por un túnel definido en la red pública.

En la Figura 32 se puede ver un ejemplo de una VPN, donde diferentes dispositivos se conectan empleando Internet como una red privada.

Figura 32.
Ejemplo de una VPN



6.3. Seguridad de las redes LAN inalámbricas

Para este tema le sugerimos realizar la lectura del capítulo 8, específicamente la sección 8.8: "Asegurando las redes LAN inalámbricas" del texto básico, donde se revisa vulnerabilidades de este tipo de redes y mecanismos para salvaguardar la integridad de la red y de los datos.

Una red LAN inalámbrica (WLAN) al emplear un medio compartido como el aire, presenta vulnerabilidades, ya que cualquier dispositivo que se encuentre dentro de la cobertura del equipo emisor podrá interceptar la señal. Le invitamos a pensar en lo siguiente: ¿qué es lo que hace que las redes inalámbricas sean más vulnerables que las redes de cable?

Existen ataques particulares a las WLAN, que aprovechan algunas las debilidades de este tipo de redes como:

- Ataques por interferencia.
- Escucha pasiva de mensajes.
- Ataque por intercepción e inserción.

A continuación, revisemos los principales mecanismos para brindar seguridad a una WLAN, donde cada mecanismo logra un nivel diferente de seguridad y presenta ciertas ventajas y desventajas. Se hará a continuación un breve análisis de estos métodos:

- WEP (Wired Equivalent Privacy): este es el nivel más básico de seguridad para red inalámbrica, esta es una característica estándar que se incorpora en todas las redes WLAN certificadas con la norma WiFi (soportado por la gran mayoría de fabricantes de soluciones inalámbricas). WEP, creado por el Instituto de Ingenieros en Electricidad y Electrónica (IEEE), ha sido diseñado para proporcionar un nivel de seguridad, prevenir posibles escuchas de la información y proteger la red mediante la encriptación de todos los datos que se envíen de forma inalámbrica. Esta forma parte de la especificación 802.11, y se diseñó con el fin de proteger los datos que se transmiten en una conexión inalámbrica mediante cifrado. Pero presenta algunas vulnerabilidades que se encuentran especificadas en el texto básico como un cifrado relativamente débil. Actualmente existen herramientas gratuitas para romper la clave secreta de enlaces protegidos con WEP.

- 802.11i: está focalizada en la mejora de la seguridad para protocolos de autenticación y de codificación. El estándar abarca los protocolos 802.1x, TKIP (Protocolo de Claves Integra Seguras Temporales), y AES (Estándar de Cifrado Avanzado). Y para la autenticación emplea el protocolo ampliable de autenticación (EAP). Es decir, que proporciona varias formas de cifrado basado en AES y una versión más fuerte del cifrado WEP.

Para finalizar esta sección, comentar que la seguridad en las WLAN es una necesidad, que por ellas se transmite para no poner en peligro la confidencialidad e integridad de dicha información. Existen adicionalmente otros métodos que pueden incrementar la seguridad de la WLAN, le invito estimado estudiante, a realizar una investigación de ellos.

6.4. Seguridad operacional

Regresemos un momento al texto básico, y realice una lectura de la sección 8.8: "Seguridad operacional: cortafuegos y sistemas de detección de intrusos", donde se revisa las funcionalidades de un cortafuegos y sistemas de detección de intrusos.

Una vez que finalizó la lectura recomendada, usted ya tiene conocimiento de la importancia de incorporar a una red estos elementos adicionales para incrementar la seguridad de una red. A continuación, revisemos algunas funcionalidades y diferencias importantes de estos:

- Cortafuegos o también conocido como Firewall, cuya principal función es brindar una protección integra a una red de amenazas tanto entrantes como salientes. Estos se implementan en dispositivos especializados con *software* o mediante s*oftwar*e implementado en un computador.

- Sistemas de detección de intrusos (IDS: Intrusion Detection System), empleados para detectar accesos no autorizados a un computador o a una red. Es decir, es un sistema que intenta detectar y alertar sobre las intrusiones intentadas en un sistema o en una red, considerando como intrusión a toda actividad no autorizada o que no debería ocurrir en ese sistema. Esta definición muchos podrían pensar que ese trabajo ya se realiza mediante los cortafuegos o *firewalls*.

Ahora revisemos las diferencias entre los dos componentes y como un IDS es un buen complemento de los cortafuegos.

La principal diferencia es que un cortafuego es una herramienta basada en la aplicación de un sistema de restricciones y excepciones sujeta a muchos tipos de ataques, desde los ataques "*tunneling*" (saltos de barrera) a los ataques basados en las aplicaciones. Los cortafuegos filtran los paquetes y permiten su paso o los bloquean por medio de una Tabla de decisiones basadas en el protocolo de red utilizado. Las reglas se verifican contra una base de datos que determina si está permitido un protocolo determinado y permite o no, el paso del paquete, basándose en atributos tales como, las direcciones de origen y de destino, el número de puerto, etc... Esto se convierte en un problema cuando un atacante enmascara el tráfico que debería ser analizado por los cortafuegos o utiliza un programa para comunicarse directamente con una aplicación remota. Estos aspectos se escapan a las funcionalidades previstas en el diseño inicial de los cortafuegos. Es aquí donde entran los IDS, ya que estos son capaces de detectar cuando ocurren estos eventos.

Actividad de aprendizaje recomendada

Repaso del tema seguridad y gestión de red

Revisar el vídeo del canal de YouTube, denominado Red Privada Virtual (VPN), donde se explica las ventajas y desventajas de este tipo de redes.

Le invitamos a resolver las siguientes preguntas que le servirán para determinar las áreas en las cuales ha tenido mayor dificultad y necesitan una nueva revisión.

Autoevaluación 6

Lea detenidamente cada enunciado y conteste con (V) en el caso de ser verdadero y con (F) si es falso:

1. ( ) La seguridad de las aplicaciones se refiere a las medidas de seguridad, a nivel de aplicación, cuyo propósito es impedir el robo o el secuestro de datos o códigos dentro de la aplicación.

2. ( ) SSH es una aplicación diseñada para substituir determinadas herramientas de acceso remoto, usadas tradicionalmente en los sistemas Unix.

3. ( ) El correo electrónico seguro emplea métodos para proteger el correo electrónico en el mismo nivel de aplicación, independientemente del sistema de transporte utilizado.

4. ( ) Las VPN no emplea IPsec.

5. ( ) Las WLAN no presentan vulnerabilidades de seguridad.

6. ( ) 802.11i tiene las mismas vulnerabilidades de seguridad que WEP.

7. ( ) Firewall tiene como función brindar una protección integra a una red de amenazas tanto entrantes como salientes.

8. ( ) Los IDS tienen las mismas funciones que un Firewall.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. En la práctica, la seguridad en la red supone no solamente proteger, sino también:

    a. Detectar las brechas en la comunicación segura.
    b. Ataque a la infraestructura.
    c. Responder a los ataques.
    d. Todas las anteriores.

10. ¿Qué permiten las técnicas criptográficas?

    a. Que el receptor debe ser capaz de recuperar los datos originales a partir de los datos disfrazados.
    b. Que un emisor disfrace los datos de tal forma que un intruso no pueda obtener información de los datos interceptados.
    c. Ninguno de los anteriores.
    d. Todas las anteriores.

Verifique sus respuestas en el solucionario que se encuentra al final de la presente guía didáctica.

Ir al solucionario

Resultado de aprendizaje 1, 2 y 3

- Discute las arquitecturas típicas de gestión de la red.
- Diseña aplicaciones de red orientada a datos.
- Esquematiza estrategias de seguridad básica en redes de computadoras.

Contenidos, recursos y actividades de aprendizaje

Semana 7 y 8

Repaso de unidades 1- 6

Estimado estudiante, en esta semana lo invitamos a revisar los contenidos estudiados en el primer bimestre. Específicamente, deberá revisar los contenidos de las Unidades 1 a la 6. Esta revisión le permitirá reforzar los conocimientos adquiridos, lo cual lo preparará para la evaluación del primer bimestre.

También le recordamos que puede conectarse al chat de la tutoría para cualquier inquietud que tenga en el momento de revisar los contenidos del primer bimestre. Además, no olvide repasar las autoevaluaciones y ejercicios planteados en las Unidades antes mencionadas.

Segundo bimestre

Resultado de aprendizaje 4

- Describe el funcionamiento de redes multimedia y de tiempo real.

Contenidos, recursos y actividades de aprendizaje

Semana 9

Unidad 7. Aplicaciones de redes multimedia

Iniciamos esta Unidad revisando “Redes multimedia” que es la encargada de buscar, localizar y obtener información de la importancia de las bases de datos internas y los servidores de medios de comunicación, de depósitos mundiales, sea la NASA, la Biblioteca del Congreso de los Diputados, el Museo del Louvre y/o los múltiples foros electrónicos que ya existen sobre cualquier tema concebible. Los usuarios ansiosos de tener un almacén

de información mundial en sus manos pueden acceder sin complicaciones a estos depósitos y tener en su poder los resultados de sus búsquedas, gracias a las redes de comunicaciones de datos.

7.1. Ejemplos de aplicaciones multimedia

A continuación, es importante realizar una lectura de los principios generales de las redes de computadores y su evolución cronológica.

Esta Unidad en sí tratará de las "Aplicaciones de las Redes Multimedia". Le agradecería recordar de manera general o haciendo cuadros sinópticos sobre la clasificación de las aplicaciones multimedia, **y así tendrá una mayor visión de lo que estudiaremos.**

Para tener una visión general de esta Unidad, le invitamos a desarrollar la siguiente actividad:

Actividad de aprendizaje recomendada

Realizar un listado de al menos 10 sitios relevantes de Internet donde se suben información de audio y vídeo.

Una vez realizado el ejercicio anterior le invito a reflexionar sobre el tipo de soporte tecnológico que deben tener estas aplicaciones para su normal funcionamiento en una red, como es Internet.

Veremos enseguida que en muchas aplicaciones multimedia los paquetes que sufren un retardo emisor receptor de más de un poco de cientos de milisegundos resultan inútiles para el receptor. Por otro lado, las aplicaciones de red multimedia son casi siempre tolerantes a las pérdidas.

En este tema vamos a revisar primeramente el texto básico la sección 9.1.3, concerniente a las tres clases generales de aplicaciones multimedia como son: flujos de audio/vídeo almacenados y flujos de audio/vídeo en vivo.

También se señala en la bibliografía básica que existen aplicaciones de compartición de archivos P2P y que podemos leer la música antes de reproducirlos, entre algunas de estas aplicaciones tenemos: Ares, Lime Wire y Emule.

Una vez que ha leído las tres clases de aplicaciones multimedia podemos mencionarles algunos ejemplos prácticos, estos son:

- Vídeos de YouTube
- Televisión en vivo
- Radios en vivo
- Telefonía por Internet con vídeo incluido. Por ejemplo, las herramientas como: Skype, IONOS Video Chat, Spike, Zoom, ICQ, Jitsi, Tox, Viber, Facebook, WhatsApp, FaceTime, Google Duo, Google Hangouts, Line, WeChat y Wire, entre otros.

7.2. Obstáculos para la información multimedia en la Internet actual

Es importante revisar las características y ventajas de TCP y UDP para poder ver hasta qué punto estos protocolos de transporte proporcionan la adecuada garantía de retardo en las aplicaciones que la convocan. A continuación, veremos ejemplos de vídeos interactivos en tiempo real y cómo la telefonía por Internet ha encontrado un amplio uso en la vida común de las personas.

Algunos vídeos interactivos en tiempo real:

Vídeos interactivos semánticos. Por ejemplo, Semantic Gap. Consiste en enriquecer los documentos de vídeo con fuentes de datos externos o metadatos que pueden ser de tres tipos: datos descriptivos, anotaciones de texto y anotación semántica.

En estos vídeos el reto es conseguir que los usuarios puedan localizar la información que deseen cuando estén visualizando –en tiempo real- alguna sección particular del vídeo. Por ejemplo, supongamos que un usuario está visualizando un vídeo sobre la preparación de un plato típico narrado por un interlocutor en una lengua distinta a la lengua nativa del usuario. Imaginemos que en algún punto de la narración el usuario está interesado en obtener información sobre el lugar de procedencia del plato típico o el lugar de nacimiento y la nacionalidad del narrador. El usuario también podría sentir interés por obtener información acerca de las propiedades alimenticias de los ingredientes mencionados por el narrador o podría preguntarse dónde comprar tales ingredientes en su localidad. Toda esta información estaría disponible durante la visualización del vídeo.

Vídeos 3D interactivos. Es una de las experiencias perceptuales y táctiles que más ha impactado en el mundo actual; prueba de ello es que aún se está intentando conceptualizar esta manera de interactuar con la realidad a través de vídeos dispuestos en forma inteligente para generar la sensación de tridimensionalidad. Lo que verdaderamente genera ilusión, de que algún día - no muy lejano - esta tecnología llegue a nosotros en tiempo real.

El mayor de los obstáculos para este tipo de aplicaciones en tiempo real tal como lo dice el texto básico es la fluctuación de paquetes; es decir, el retardo y la fluctuación dentro del mismo paquete. El requerimiento para que este tipo de aplicaciones puedan funcionar es el ancho de banda, y eso muchas de las veces en empresas

pequeñas es un limitante, ya que no cuentan con recursos para delegar un ancho de banda exclusivo para este tipo de aplicaciones en su empresa.

7.3. Evolución de Internet para dar un mejor soporte a las aplicaciones multimedia

Como se puede leer en la sección 9.5, Internet no deja de evolucionar y eso lo vemos cada día, considerando las aplicaciones de multimedia subidas a la *Web* y a los servidores. Internet no solo será mucho más rápida gracias al desarrollo de las redes de muy alta velocidad, sino que además será cada vez más omnipresente, disponible en cualquier momento y en cualquier lugar. Esta comunicación se puede considerar como un paso preparatorio hacia la Internet del futuro y que exigirán las nuevas generaciones y aplicaciones en tiempo real. Las nuevas tendencias supondrán un desafío para la economía digital. En el recurso vemos la evolución que ha tenido la red desde su aparición hasta el presente año.
Es importante revisar el ANEXO 1 donde se encuentra tipos de servidores Web que le ayudarán a esclarecer mejor el tema.

Evolución de la red hasta el año 2021

El extendido uso de la banda ancha ha cambiado la forma en que los ciudadanos utilizan Internet. Si bien a mediados de los años 90 constituía una mera fuente de información, la "Web 2.0" fue más participativa e interactiva gracias a avances esenciales en servicios fáciles de utilizar. La cual también ha evolucionado, y en la actualidad ya estamos hablando de la Web 3.0 o Web Semántica, la cual apareció en el año 2006, pero se operativizó en el 2010. Se trata de una extensión de *World Wide Web*, por la que se pueden encontrar datos en cualquier lengua y en formatos aptos para todo tipo de *software*.

Web 3.0 incluye, la transformación de la red en una base de datos, un movimiento dirigido a hacer los contenidos accesibles por múltiples aplicaciones que no son solamente el navegador, el empuje de las tecnologías de inteligencias artificial, la web Geoespacial, la Web 3D. Otro posible camino para la Web 3.0 es la dirección hacia la visión 3D, liderada por el Web 3D Consortium.

La web 4.0, la cual ya la estamos usando en la actualidad y muchos ni siquiera la notamos. Esta *web* está sustentada en 4 pilares fundamentales:

- Comprensión del lenguaje natural.
- Comunicación máquina a máquina.
- Uso de la información de contexto. Se podrán usar *wearables* que monitoricen la tensión arterial o la temperatura corporal de una persona.
- Nuevas formas de interacción con la persona.

Así mismo se pueden distinguir algunas tendencias principales:

- Se prevé una evolución de las redes sociales para empresas que dará lugar a instrumentos de colaboración para estas (Enterprise 4.0). Este hecho, junto con la transformación del *software* en servicio, llevarán a una nueva generación de servicios informáticos fácilmente disponibles. Es lo que se conoce como la Internet de los servicios, pero en un entorno de transformación digital y en la nube.
- También se producirá el auge de la Internet de los objetos, que es la conexión sin fisuras de dispositivos, sensores, objetos, etc. a través de redes fijas e inalámbricas.

- El uso nómada a través de dispositivos portátiles transformará los modelos de organización de las empresas.
- Se necesitará un incremento de banda ancha como consecuencia del ingente tráfico de datos previsto, y, además, debido a la incorporación de Inteligencia Artificial en el manejo de grandes volúmenes de información, esto ralentizaría los procesos a ejecutar.

La presión de la competencia constituye el modo más eficaz de promover la migración a la banda ancha. No obstante, será esencial que Internet se mantenga abierta y los mercados de comunicaciones sigan siendo competitivos. Será preciso estimular la inversión en el acceso de banda ancha de alta velocidad dado el elevado coste de las obras de ingeniería civil necesarias, que supone un 80 % del total, así como por la incertidumbre de si los consumidores estarán dispuestos a pagar una cantidad suficiente por la obtención de servicios de banda ancha, de forma que las inversiones resulten rentables.

Una de las prioridades políticas será conseguir banda ancha para todos a un precio asequible en zonas rurales y urbanas. Lo que se analiza en el tema de discusión sobre "índice de eficacia de la banda ancha" en el informe de avance anual de Lisboa. El índice es un indicador compuesto, que pone de manifiesto la necesidad de mayor velocidad, cobertura, precios asequibles, innovación, servicios de calidad y un contexto socioeconómico favorable.

También existe un problema de competencia y convergencia. Dado que la convergencia está difuminando los límites entre los mercados de las telecomunicaciones, la electrónica de consumo, los servicios de difusión y las empresas de Internet, es importante garantizar que Internet se mantenga abierta a la competencia y la innovación. También es fundamental que los consumidores dispongan de

opciones reales y no se vean limitados a los mismos servicios y productos.

La arquitectura de Internet actual es insuficiente para afrontar los retos que plantean la informática nómada y la Internet de los objetos. Por lo tanto, es necesario iniciar un debate sobre el diseño y desarrollo de la Internet del futuro, ya que esta tendrá que responder a exigencias cada vez mayores de escalabilidad, movilidad, flexibilidad, seguridad, confianza y solidez.

También es fundamental preservar la privacidad y seguridad de la Internet del futuro desde una fase temprana. Para ello, la Comisión de Lisboa proporcionará directrices claras sobre la aplicación de la normativa en materia de protección de datos existentes y una estrategia coherente para una Internet del futuro segura.

En todos estos avances no hay que olvidar el papel fundamental que desempeñan los aspectos internacionales de la política, el diálogo en materia de reglamentación y la cooperación en materia de investigación.

En el texto básico se habla de tres enfoques para el tratamiento del tráfico multimedia que son:

- Mejorar el servicio del mejor esfuerzo
- QoS diferencial
- QoS garantizado

Estos enfoques son importantes al momento de describir algunas características de la red del futuro que deberemos trabajar muy fuerte en el tema de tráfico en la red donde se subirán múltiples aplicaciones de manera secuencial y paralela. Estos métodos o enfoques darán soporte a las aplicaciones multimedia. Revisar la Tabla 9.4. del texto básico.

Para entender mejor este tema sobre los enfoques para el tratamiento del tráfico multimedia, es importante que vea este vídeo del canal de YouTube, Curso MMInet, sobre Calidad de servicio en Internet para tráfico multimedia: best effort, IntServ, DiffServ). En este video tutorial aprenderá a diferenciar los enfoques de tratamiento del tráfico multimedia, así como la calidad de servicio como un elemento relevante en la transferencia de la información.

7.3.1. Servicio del mejor esfuerzo

Este enfoque está definido en el IntServ, considerado como servicios integrados que incorporan la provisión de QoS entre extremos de la red IP para determinados flujos de información. Un flujo es una secuencia de paquetes IP desde un único transmisor destinado a un único receptor. Para realizar la configuración de IntServ se usa el protocolo de señalización RSVP.

IntServ define tres grandes clases de servicios que una aplicación puede requerir: los servicios garantizados proveen condiciones seguras en comunicaciones de extremo a extremo. De carga controlada, provee la misma calidad de servicio que el flujo recibiría si la red está descongestionada, pero asegurando que el servicio se conservaría aun cuando la red estuviera sobrecargada. Servicios de mejor esfuerzo donde no se garantiza el éxito de un servicio.

Las ventajas de IntServ son:

- La simplicidad conceptual, que facilita que toda la red mantenga una política de red integrada.
- La posibilidad de crear reglas de QoS para flujos discretos, generación de llamada de voz, control de admisión de llamadas (CAC), lo que permite conocer a los nodos extremos sobre la disponibilidad de ancho de banda.

Las desventajas:

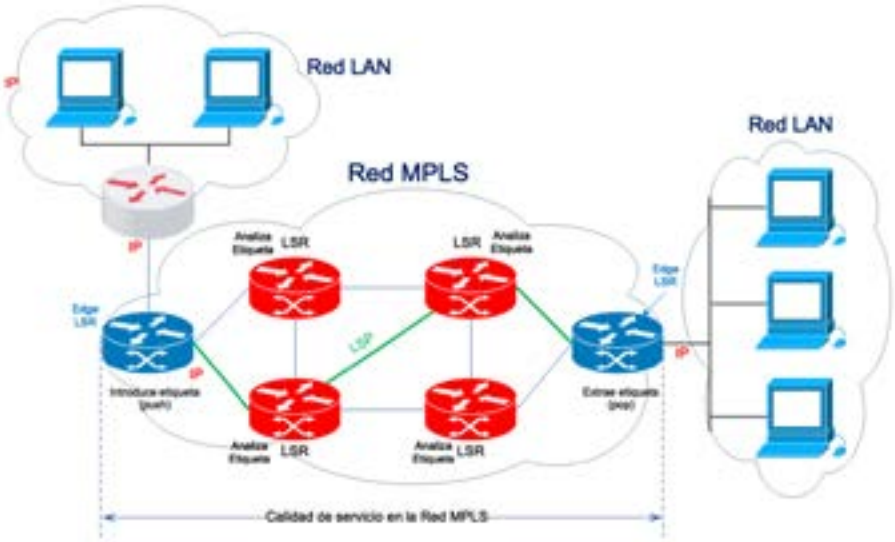
- Todos los elementos deben mantener el estado e intercambiar mensajes de señalización por cada flujo. Se necesitan mensajes periódicos de actualización para mantener la sesión, lo que aumenta el tráfico en la red y es susceptible a perdidas de paquetes. Todos los nodos intermedios deben tener RSVP en sus funciones.
- Se necesita de nuevo *software* tanto en el envío de paquetes y en el control de todos los *routers* a lo largo del camino de la red concerniente.

7.3.2. QoS diferencial

Un ejemplo de este tipo de enfoques IP/VPN con acceso satelital, el cual es un servicio de interconexión de redes locales sobre infraestructura similar a la que tiene Telefónica. Permite la creación de redes privadas virtuales sobre dicha infraestructura compartida manteniendo las mismas prestaciones que si fuera una red privada, reduciendo costos y aumentando el rendimiento.

Tiene como base la Red IP-MPLS Figura 33, la cual ofrece calidad de servicio de extremo a extremo para la transmisión de voz, datos y vídeo.

Figura 33.
Red IP-MPLS



Entre uno de los beneficios está que ofrece calidad de servicio (QoS) diferencial por tipo de aplicación. De esta manera cada cliente puede personalizar el grado de prioridad que van a tener cada una de sus aplicaciones a través de la red IP MPLS.

7.3.3. QoS garantizado

Significa máximo ancho de banda para que las aplicaciones multimedia funcionen de forma correcta.

En organizaciones grandes con oficinas remotas en el mundo la priorización del tráfico WAN es crítica. Los enlaces WAN son costosos y muy limitados en ancho de banda. Muchas aplicaciones críticas como ERP (Entreprise Resourse Planing, voz sobre IP, servidores remotos de aplicaciones, consultas a información crítica, etc., requieren de anchos de banda definidos y garantizados. Sin una priorización de los servicios y una repartición adecuada del ancho

de banda estos servicios colapsan y los tiempos muertos o fuera de servicio son cada vez más frecuentes e interminables.

Utilizando servicios con QoS, el ancho de banda de la red puede ser garantizado para los servicios esenciales durante los períodos de alta congestión. Utilizando esquemas de priorización de tráfico que se modifiquen en el tiempo se logra una mejor administración y uso de los recursos de ancho de banda limitados.

Cuando hay excesivo tráfico y congestión se priorizan los servicios esenciales y se les entrega la mayor disponibilidad del ancho de banda; luego al disminuir la carga o cuando los servicios esenciales no están en uso, el ancho de banda se retorna automáticamente al resto de los solicitadores de recursos.

Como podemos ver, los tres enfoques son muy esenciales para dar soporte a las aplicaciones multimedia que permitan versatilidad al momento de hacer tráfico en la red.

7.4. Compresión de audio y vídeo

En las páginas del texto básico sección 9.1.1 y 9.1.2, está una explicación bien detallada de lo que significa la compresión de audio y vídeo en Internet y de como una señal analógica se convierte en digital. Además, se explica sobre la técnica de codificación básica y técnicas de compresión de audio populares. En cuanto a la compresión de vídeo lo más relevante a resaltar son los tipos de redundancia en los vídeos que son: redundancia espacial y redundancia temporal. A continuación, se mencionan otras técnicas para audio (Ver Tabla 17) y video:

Audio:

- G.711 (64kbps). Codec “base”, utiliza dos posibles leyes de compresión: μ-law y A-law (Recommendation G.711, 1988).
- G.722 (164 kbps, 256 kbps y 348 kbps). Inicialmente diseñado para audio y videoconferencias, actualmente utilizado para servicios de telefonía de banda ancha en VoIP (Recommendation G.722, 2010).
- GSM-900: 124 canales en dos sub-bandas
- DCS-1800: 374 canales en dos sub-bandas de 75 Mhz
- PCS-1900:374 canales en dos sub-bandas de 75 Mhz
- DPCM (Diferential Pulse Code Modulation)
- PAC - (Perceptual Audio Coding).

Tabla 17.
Comparación entre TCP y UDP

Formato del fichero de audio	Extensión del fichero	Estándar	Aplicación
Digital Theater Systems	DTS	ETSI TS 102 114	DVD, Audio CD
Dolby Digital	AC-3	ATSC Standard A/52A	DVD
MPEG-1 Layer I	MPA	ISO/IEC-11172-3	
MPEG-1 Layer II	MP2	ISO/IEC-11172-3	VCD, SVCD
MPEG-1 Layer III	MP3	ISO/IEC-11172-3	
MPEG-2 Layer I	MPA	ISO/IEC-13818-3	
MPEG-2 Layer II	MP2	ISO/IEC-13818-3	5.1 SVCD
MPEG-2 Layer III	MP3	ISO/IEC-13818-3	

Vídeo:

- H.261
- H.264 o MPEG-4
- OGG THEORA
- M-JPEG
- MPEG-7
- MPEG-21
- DivX

Estimado estudiante, Le invitamos a resolver las siguientes preguntas que le servirán para determinar las áreas en las cuales ha tenido mayor dificultad y necesitan una nueva revisión.

Autoevaluación 7

Lea detenidamente cada enunciado y conteste con (V) en el caso de ser verdadero y con (F) si es falso:

1. ( ) Las aplicaciones multimedia de red nunca son tolerantes a las pérdidas.

2. ( ) Son características diferenciadoras de flujos de audio y vídeo almacenado: medios almacenados, flujos, reproducción continua.

3. ( ) La distribución de audio/vídeo en vivo a muchos receptores no puede llevarse a cabo eficientemente a través de técnicas de multidifusión IP.

4. ( ) El audio interactivo en tiempo real a través de Internet suele referirse como telefonía por Internet.

5. ( ) En el caso de voz los retardos menores de 50 milisegundos no son percibidos por el oído humano.

6. ( ) El protocolo IP implantado en la red Internet proporciona un servicio de mejor esfuerzo a todos los datagramas que transporta.

7. ( ) Según el método de laissez-faire en el tráfico de flujos en vivo no pueden implantarse redes solapadas de multidifusión.

8. ( ) Antes de poder transmitir a una red de computadoras el audio y vídeo es necesario digitalizarlos y comprimirlos.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. ¿Cuáles son los enfoques para el tratamiento de tráfico multimedia?

   a. Mejorar el servicio del mejor esfuerzo
   b. QoS diferencial
   c. QoS garantizad
   d. Todos los anteriores

10. ¿Cuál de las siguientes son técnicas de compresión de vídeo?

   a. MPEG4
   b. GSM
   c. G.72
   d. MP3

Ir al solucionario

Semana 10

Unidad 8. Flujos de audio y video almacenado

Estimado estudiante, para tener una visión general de esta unidad, le invitamos a desarrollar la siguiente actividad:

Actividad de aprendizaje recomendada

Elabore un listado de servidores, reproductores y tecnologías de protocolos con sus respectivas ventajas y desventajas. Estos se encuentran en la sección 9.2. texto básico.

Para entender mejor este tema es importante que vea este vídeo del canal de YouTube, UPV, sobre Transmisión de flujos de video. En este vídeo tutorial aprenderá a diferenciar como se realiza la transmisión de video con UDP y TCP. Así como también tipos de *streaming* de vídeo.

8.1. Acceso al audio y vídeo a través de un servidor Web

Primeramente, para saber cómo se accede a un servidor *Web* es importante establecer qué tipos de servidores existen y saber diferenciar su uso de las mismas.

Esta lista categoriza los diversos tipos de servidores del mercado actual:

- Plataformas de servidor (*Server Platforms*): un término usado a menudo como sinónimo de sistema operativo, la plataforma es el *hardware* o *software* subyacentes para un sistema, es decir, el motor que dirige el servidor.
- Servidores de aplicaciones (*Application Servers*): designados a veces como un tipo de middleware (*software* que conecta dos aplicaciones), los servidores de aplicaciones ocupan una gran parte del territorio entre los servidores de bases de datos y el usuario, y a menudo los conectan.
- Servidores de audio/vídeo (*Audio/Video Servers*): los servidores de audio/vídeo añaden capacidades multimedia a los sitios *Web* permitiéndoles mostrar contenido multimedia en forma de flujo continuo (*streaming*) desde el servidor.
- Servidores de *chat* (*Chat Servers*): los servidores de *chat* permiten intercambiar información a una gran cantidad de usuarios ofreciendo la posibilidad de llevar a cabo discusiones en tiempo real.
- Servidores de fax (*Fax Servers*): un servidor de fax es una solución ideal para organizaciones que tratan de reducir el uso del teléfono, pero necesitan enviar documentos por fax.

- Servidores FTP (*FTP Servers*): uno de los servicios más antiguos de Internet, File Transfer Protocol permite mover uno o más archivos a sitios remotos.
- Servidores groupware (*Groupware Servers*): un servidor *groupware* es un *software* diseñado para permitir colaborar a los usuarios, sin importar la localización, vía Internet o vía Intranet corporativo y trabajar juntos en una atmósfera virtual.
- Servidores IRC (*IRC Servers*): otra opción para usuarios que buscan la discusión en tiempo real. Internet Relay Chat consiste en varias redes de servidores separadas que permiten que los usuarios conecten el uno al otro vía una red IRC.
- Servidores de listas (*List Servers*): los servidores de listas ofrecen una manera mejor de manejar listas de correo electrónico, bien sean discusiones interactivas abiertas al público o listas unidireccionales de anuncios, boletines de noticias o publicidad.
- Servidores de correo (*Mail Servers*): casi tan ubicuos y cruciales como los servidores Web. Los servidores de correo mueven y almacenan el correo electrónico a través de las redes corporativas (vía LANs y WANs) y a través de Internet.
- Servidores de noticias (*News Servers*): los servidores de noticias actúan como fuente de distribución y entrega para los millares de grupos de noticias públicos actualmente accesibles a través de la red de noticias USENET.
- Servidores proxy (*Proxy Servers*): los servidores proxy se sitúan entre un programa del cliente (típicamente un navegador) y un servidor externo (típicamente otro servidor Web) para filtrar peticiones, mejorar el funcionamiento y compartir conexiones.

- Servidores telnet (*Telnet Servers*): un servidor telnet permite a los usuarios entrar en un ordenador huésped y realizar tareas como si estuviera trabajando directamente en ese ordenador.

- Servidores web (*Web Servers*): básicamente, un servidor *Web* sirve con contenido estático a un navegador, carga un archivo y lo entrega a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante http. No olvide revisar el ANEXO 1 sobre este tema.

En el texto básico en la sección 9.2.1 y Figura 9.1, se explica detalladamente cómo es el flujo de la información desde un servidor *Web* hacia un reproductor multimedia. En este proceso podemos ver cuán importante es el metarchivo y como ayuda a la comunicación directa al servidor *Web*.

8.2. Envío de información multimedia desde un servidor de flujos a una aplicación de ayuda

En la Figura 9.2. del texto básico se explica una transmisión de datos desde un servidor de flujos a un reproductor multimedia y una explicación detallada de cómo la arquitectura de este tipo de transmisión de información requiere de dos servidores: servidor *Web* y servidor de flujos.

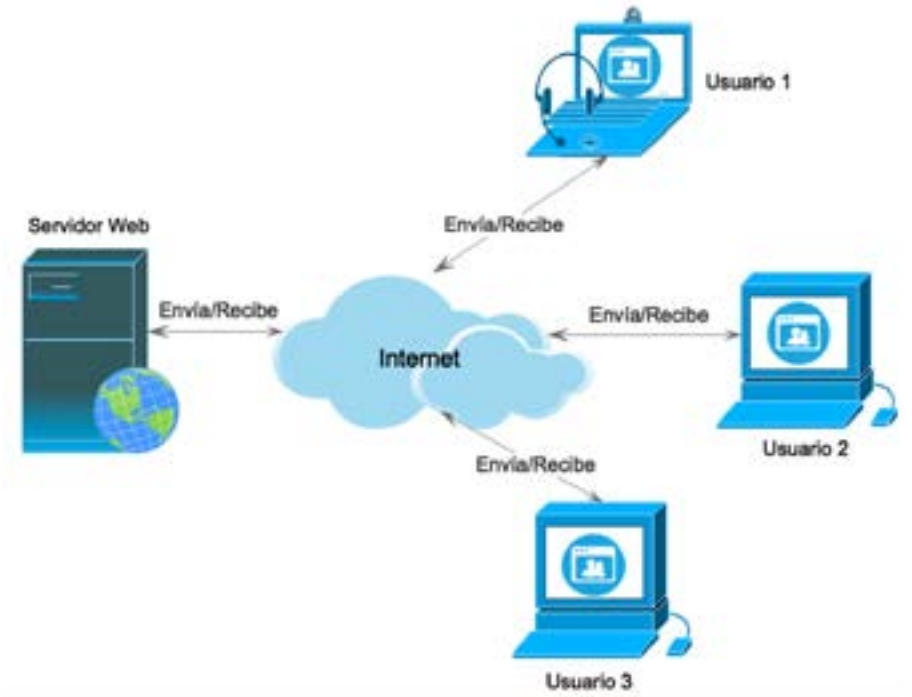
Para tener una visión general de esta Unidad, le invitamos a desarrollar la siguiente actividad:

Actividad de aprendizaje recomendada

Elaborar un cuadro sinóptico de las opciones para la entrega de audio y vídeo desde el servidor de flujos al reproducir multimedia.

A continuación, en la Figura 34, se muestra cómo se envía y recibe datos de audio y vídeo desde un servidor *Web* en concordancia con los equipos personales de los usuarios finales.

Figura 34.
Envío y recepción de audio y vídeo desde un servidor Web



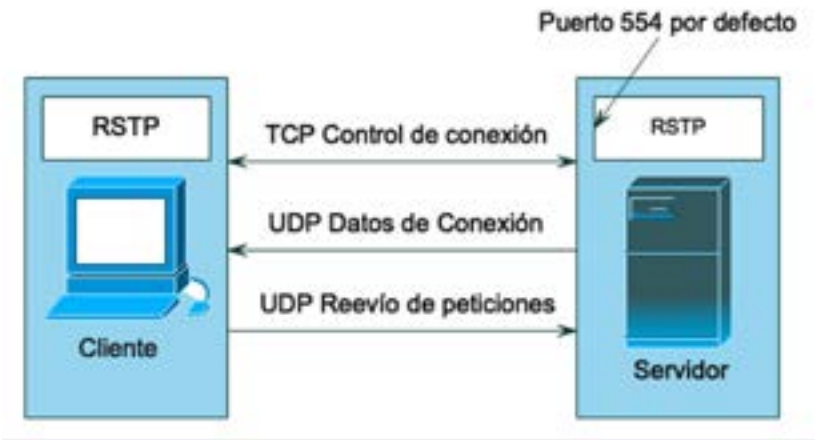
8.3. Protocolos de transmisión de flujos en tiempo real

En esta sección se analiza un sistema que proporciona interactividad con medio como es el RTSP – Real Time Streaming Protocol. Es un protocolo no orientado a la conexión, en lugar de esto el servidor mantiene una sesión asociada a un identificador, en la mayoría de los casos RTSP usa TCP para datos de control del reproductor y UDP para los datos de audio y vídeo, aunque también puede usar TCP en caso de que sea necesario. En el transcurso de una sesión RTSP, un cliente puede abrir y cerrar varias conexiones de transporte hacia el servidor con tal de satisfacer las necesidades del protocolo (Ver Figura 35).

El protocolo soporta las siguientes operaciones:

- Recuperar contenidos multimedia del servidor.
- Invitación de un servidor multimedia a una conferencia.
- Adición multimedia a una presentación existente.

Figura 35.
Operaciones básicas del protocolo RTSP



Ahora revisaremos los requerimientos de servicios en tiempo real y cómo se transmiten a través de las redes, para lo cual es necesario leer la sección 9.2.1., 9.2.2 y 9.4.1 previamente. En este contexto se puede controlar la reproducción, el reproductor multimedia y el servidor y como necesitan de un protocolo para intercambiar la información de control de la reproducción.

Por otra parte, lo que no hace RTSP está detalladamente explicado en el texto básico sección 9.2.1. También, es importante indicar que usted extraiga estas funciones para poder analizar detenidamente el proceso de interacción entre un cliente y un servidor utilizando RTSP.

Para una mejor comprensión de la información es relevante establecer que el servidor responde con códigos de respuestas estandarizadas.

Sin embargo, para poder evaluar las ventajas que ofrece RTSP debemos ver la similitud y la diferencia entre HTTP y RTSP.

Similitudes:

- Formato de las peticiones/respuestas
- Línea de petición + cabeceras + cuerpo
- Códigos de estado
- Mecanismos de seguridad
- Formato de la URL
- Negociación de los contenidos
- Su sintaxis es muy similar
- Tanto los servidores como los clientes RTSP pueden realizar peticiones

Diferencias:

- Los datos son transportados mediante un protocolo diferente (datos transportados fuera de banda)
- La Request-URI siempre contiene una URI absoluta
- RTSP introduce nuevos métodos y tiene un identificador de protocolo diferente.
- Un servidor RTSP necesita mantener el estado de la conexión al contrario de HTTP.

Revisar el ejemplo de la página 585 del texto básico, donde se explica lo que es capaz de hacer el RTSP.

Autoevaluación 8

Ahora lo invitamos a revisar los conocimientos adquiridos, si su nota es baja por favor, vuelva a leer y revisar los contenidos. En esta Unidad nuestra preocupación ha sido describir cómo se hace el flujo de la información desde diversos tipos de servidores; por ello le planteamos el siguiente cuestionario como autoevaluación, recuerde que el solucionario de este se encuentra al final de la guía.

Lea detenidamente cada enunciado y conteste con (V) en el caso de ser verdadero y con (F) si es falso:

1. ( ) El protocolo de transmisión de flujos en tiempo real (RTSP) es un protocolo de dominio público que no proporciona interactividad con el usuario.

2. ( ) Cuando un archivo de audio reside en un servidor *Web*, el archivo es un objeto ordinario dentro del sistema de archivos del servidor.

3. ( ) No es tarea del navegador solicitar información acerca del archivo multimedia que va a ser transmitido mediante HTTP.

4. ( ) Un servidor de flujos pude ser un servidor de flujos propietario.

5. ( ) RTSP define esquemas de compresión para audio y vídeo.

6. ( ) RTSP no permite que un reproductor multimedia controle la transmisión de un flujo multimedia.

7. ( ) FTP utiliza el concepto fuera de banda.

8. ( ) RTSP no es un protocolo fuera de banda.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. ¿Cuál de las siguientes funciones desempeña un reproductor de medios?

   a. Compresión
   b. No eliminación de fluctuaciones
   c. Encriptación
   d. Descompresión

10. ¿Cuál de las siguientes funciones hace RTSP?

   a. Define los esquemas de compresión para audio y vídeo.
   b. Permite a los protocolos de medios controlar la transmisión del flujo.
   c. Define cómo el audio/vídeo es encapsulado en paquetes para ser transmitido sobre una red.
   d. Restringe el transporte de los medios.

Ir al solucionario

Semana 11

Unidad 9. Utilización óptima del servicio de entrega del mejor esfuerzo

Una vez revisado el protocolo RSTP (CISCO, 2019c), ahora abordaremos el tema de protocolo de mejor esfuerzo.

9.1. Limitaciones de un servicio de entrega de mejor esfuerzo

Para profundizar en esta Unidad, le invitamos previamente a desarrollar la siguiente actividad:

Actividad de aprendizaje recomendada

Elaborar un cuadro sinóptico de las limitaciones de un servicio de entrega de mejor esfuerzo. Para ayudarse con esta actividad debe revisar el texto básico desarrolladas en la sección 9.3.1.

Para entender mejor este tema es importante que vea este vídeo del canal de YouTube, Pro Amperos, sobre Telefonía IP y VoIP. En este

vídeo tutorial aprenderá a conocer las ventajas de VoIP, además conocer cómo se puede transmitir telefonía a través de las redes de datos.

Las limitaciones por las que tiene que pasar este tipo de aplicación es:

- Pérdida de paquetes
- Retardo terminal a terminal
- Fluctuación de los paquetes

A continuación, describimos un ejemplo de cómo funciona la telefonía por Internet (VoIP).

La VoIP (Voice Over Internet Protocol – Voz sobre Protocolo de Internet) convierte la señal de voz de su teléfono en una señal digital que puede viajar a través de Internet. Si llama a un número telefónico regular, la señal se reconvierte en el otro extremo. Dependiendo del tipo de servicio de VoIP, usted puede hacer llamadas de VoIP desde una computadora, un teléfono especial para VoIP o un teléfono tradicional con o sin adaptador. Además, la existencia de nuevos puntos de acceso a Internet de alta velocidad o "*hotspots*" en lugares públicos como aeropuertos, parques y cafés le permiten conectarse a Internet y usar el servicio de VoIP. Si su proveedor de servicio de VoIP le asigna un número de teléfono regular, entonces podrá recibir llamadas de teléfonos regulares que no necesitan ningún equipo especial y seguramente podrá marcar como siempre lo ha hecho.

En la Figura 36, se muestra un ejemplo de cómo funciona el servicio de VoIP.

Figura 36.
Funcionamiento de VoIP

9.2. Eliminación de las fluctuaciones al reproducir el audio en el receptor

En este ítem es importante tener en cuenta que las aplicaciones de vídeo tienen requisitos similares a la voz. Analizar los tres mecanismos especificados en la página 574 del texto básico, con

el fin de eliminar los efectos de fluctuaciones con el apoyo de las estrategias de reproducción tales como:

- Retardo de reproducción fijo.
- Retardo de reproducción adaptativo.

Actividad de aprendizaje recomendada

Elaborar un algoritmo asumiendo que el receptor pueda determinar si un paquete es el primero de su correspondiente período de conversación.

9.3. Recuperación frente a pérdidas de paquetes

En este capítulo se analizarán dos tipos de esquemas de anticipación de pérdidas, estos son:

- Corrección de errores hacia delante (FEC).
- Intercalado.

Considerando las características de estos tipos de esquemas podremos explicar cuál es el funcionamiento de estas técnicas que permitirán añadir información redundante al flujo original de paquetes, tal como están descritos en la sección 9.3.3 del texto básico.

Así mismo, es importante señalar que FEC se utiliza en sistemas sin retorno o sistemas en tiempo real, donde no se puede esperar a la retransmisión para mostrar los datos. Este mecanismo de corrección de errores se utiliza, por ejemplo, en las comunicaciones vía satélite, en las grabadoras de DVD y CD o en las emisiones de TDT para terminales móviles (estándar DVB-H), concretamente en

este último caso se trata de un tipo especial de FEC, el denominado MPE-FEC (Multiprotocol Encapsulation–Forward Error Correction).

La corrección de errores se puede conseguir añadiendo al mensaje original unos *bits* de redundancia. La fuente digital envía la secuencia de datos al codificador, encargado de añadir dichos *bits* de redundancia. A la salida del codificador obtenemos la denominada palabra código. Esta palabra código es enviada al receptor y este, mediante el decodificador adecuado y aplicando los algoritmos de corrección de errores, obtendrá la secuencia de datos originales. Los dos principales tipos de codificación usados son: FEC reduce el número de transmisiones de errores, así como los requisitos de potencia de los sistemas de comunicación e incrementa la efectividad de los mismos, evitando la necesidad de reenvío de los mensajes dañados durante la transmisión.

9.4. Distribución multimedia en Internet actual: redes de distribución de contenido

Tal como se revisó en la sección 2.6.3. del texto básico, CDN - Content Delivery Network es un término que se acuñó a finales de los 90 para describir un sistema de ordenadores conectados a Internet y que colaboran de forma transparente para distribuir contenidos, especialmente contenidos multimedia muy grandes, a los usuarios finales, formando una red que cuenta con una estructura eficiente para este tipo de servicios. El número de nodos y de servidores que forman estas redes varía, dependiendo de la arquitectura, pudiendo alcanzar miles de nodos con decenas de miles de servidores conectados.

Entre algunos ejemplos prácticos tenemos: redes basadas en jerarquías de almacenamiento en cachés o en servidores completamente distribuidos. Los sistemas completamente distribuidos, altamente redundantes, se diseñan para no requerir

la modificación de sus contenidos, y más importante aún, para no tener puntos únicos de fallo. La otra alternativa ofrecida es el *multicast*, que permite lograr más eficiencia en las transmisiones de una misma información a varios destinos. Esta funcionalidad, más eficiente para la distribución simultánea de información a un grupo de destinatarios, usa la estrategia de enviar solo una vez los mensajes a través de cada enlace de la red y crear copias solo cuando el camino hacia los destinos se ramifica minimizando así la carga en la red.

9.4.1. La problemática de la distribución de contenidos

Considerando la problemática de la distribución de contenidos es importante distinguir dos casos específicos:

- *Distribución de contenidos "en bloque"*: se trata de la situación en la que un fichero, que puede ser de gran tamaño está accesible para su descarga por parte de los usuarios. Ejemplo típico es la descarga de programas o de películas. En este modelo no existe relación entre el tiempo de descarga y el tiempo de consumo. Habitualmente la descarga puede requerir mucho más tiempo que el consumo, es decir, la velocidad de descarga en cada momento no es la velocidad de consumo.

- *Distribución de contenidos "en streaming"*: engloba aquellos servicios en los que se envía un flujo (*stream*) continuo, generalmente de vídeo o audio, que es recibido en tiempo real por el usuario que lo solicita. En este caso la velocidad de entrega de datos debe ser exactamente la velocidad de consumo. Si recibo un *stream* de una película de 60 minutos el *stream* durará 60 minutos, y en cada instante ofrecerá el caudal de datos exacto necesario para visualizar ese instante de película. En este caso, se trata de un flujo de datos constante por usuario, sostenido en el tiempo, lo que lo hace más restrictivo en cuanto a exigencias de red, puesto

que la pérdida de un paquete no se puede solucionar con una retransmisión. Por ejemplo, se produciría un salto en el partido que estemos viendo.

Así mismo, dentro de los servicios de *streaming* se pueden describir dos casos:

- ***Contenidos bajo demanda***: el contenido, que está pregrabado, se envía a cada usuario cuando lo solicita de forma independiente. Usualmente se permite a cada usuario un control individualizado del flujo, es decir, se le permite enviar comandos a la fuente para avanzar a mayor velocidad, retroceder, pausar o saltar dentro del contenido.

- ***Eventos en vivo*:** el contenido se genera en ese instante, y por tanto no hay posibilidad de controlar el flujo. Es habitual ofrecer este contenido en el mismo momento en el que se genera a un número grande de usuarios que reciben todos exactamente lo mismo.

Dificultades de las redes actuales y varias soluciones

En la Figura 37, se muestran distintos tipos de atascos que se pueden encontrar en una red global como Internet, están en estos lugares:

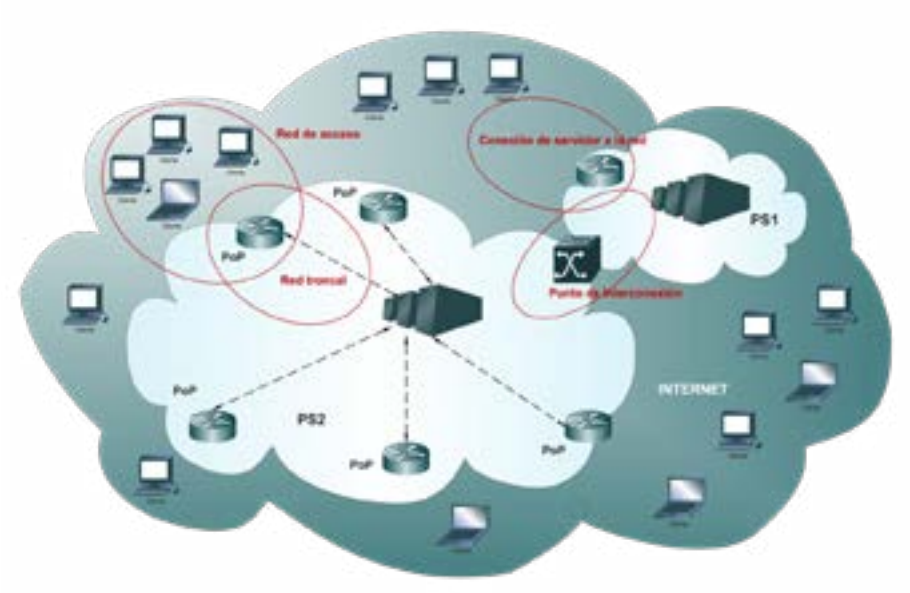
- En la conexión del servidor a la red (segmento de conexión de servidor),

- En la red troncal del servidor *backbone* de la red autónoma, llamadas en Internet AS (sistemas autónomos),

- En los puntos de interconexión *peering points*: puntos en los que se interconectan los AS,

- En la conexión del usuario (red de acceso).

Figura 37.
Tipos de atascos en la WAN



Las redes de distribución de contenidos se presentan como alternativas técnicas para minimizar la problemática que se ha descrito, permitiendo la descarga de grandes archivos a múltiples terminales de la red de una manera eficiente y escalable. Las aproximaciones que se siguen para la implementación de estas redes, y que se exponen en los siguientes apartados, son las siguientes:

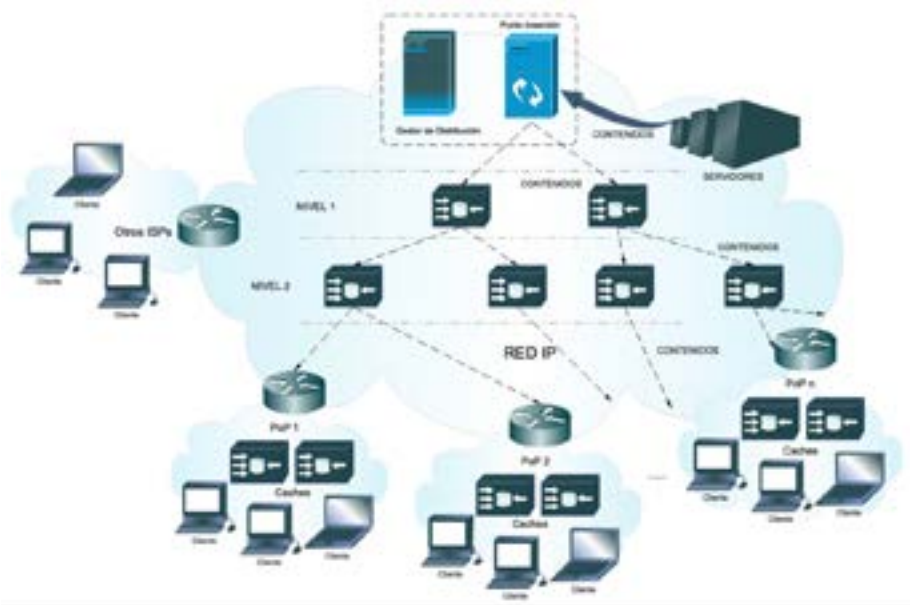
- Las redes de distribución de contenidos basadas en cachés: este escenario requiere la introducción en la red de una jerarquía de cachés, de manera que cada equipo replica los contenidos hacia los equipos de jerarquía inferior, apoyándose en protocolos *unicast*.

- Las redes de distribución basadas en *multicast*. Se trata de una solución eficiente, si bien este tipo de solución requiere disponer de redes TCP/IP que soporten *multicast*.

Distribución de contenidos en tiempo real

Esta alternativa es adecuada para la distribución de contenidos pregrabados sin exigencias de tiempo real, siendo contenidos que estarán accesibles para que los usuarios se los descarguen individualmente, como el caso de películas, archivos de música, programas u otros similares. La arquitectura típica de las redes de distribución de contenidos basadas en cachés está formada por un Gestor de Distribuciones, uno o varios puntos de inserción de contenidos, una serie de elementos replicadores intermedios agrupados jerárquicamente y finalmente, unos dispositivos de borde.

Figura 38.
Distribución de contenidos en tiempo real



- *El Gestor de Distribución*: es el dispositivo central desde donde se controla la red de distribución de contenidos. Es el elemento encargado de gestionar la replicación selectiva de los contenidos. Para ello, mantiene información de la jerarquía de la red, así como el estado de cada uno de los elementos y evolución de las distribuciones. Estos elementos permiten programar y controlar las distribuciones empleando diferentes políticas, tanto en los servidores origen como en los dispositivos destino.

- *Puntos de inserción*: son los dispositivos por los cuales los contenidos son introducidos en la red de distribución de contenidos y desde los que se inician las replicaciones al resto de elementos de la jerarquía.

- *Elementos replicadores intermedios*: están agrupados en niveles jerárquicos de forma que cada elemento conoce los miembros de su nivel y los siguientes, a los cuales se deben inyectar los contenidos. Por otro lado, el gestor de distribución determina la composición de los niveles, así como qué contenidos se deben replicar y a quién. De esta forma, cada elemento establece una conexión para recibir los contenidos de su superior cercano (padre) y otra con cada uno de los miembros del nivel inferior a los que debe inyectar los contenidos (hijos).

- *Dispositivos de borde*: constituyen el último nivel de elementos de la red de distribución de contenidos. Se trata de dispositivos de caché situados en puntos estratégicos de la red próximos al usuario, genéricamente denominados PoPs (Point of Presence). Los PoPs constituyen localizaciones geográficas donde se hospedan contenidos de la red de distribución de contenidos y desde donde la información es servida de forma distribuida hasta los usuarios, en lugar de hacerse desde el punto centralizado. El envío de la información

desde estos puntos es transparente para el usuario, para el que aparentemente es servida desde el servidor central.

La funcionalidad requerida para estos dispositivos es:

- Cacheo de contenido estático
- Facilidades de *streaming*
- Splitting de flujos en vivo
- Distribución de vídeo en vivo

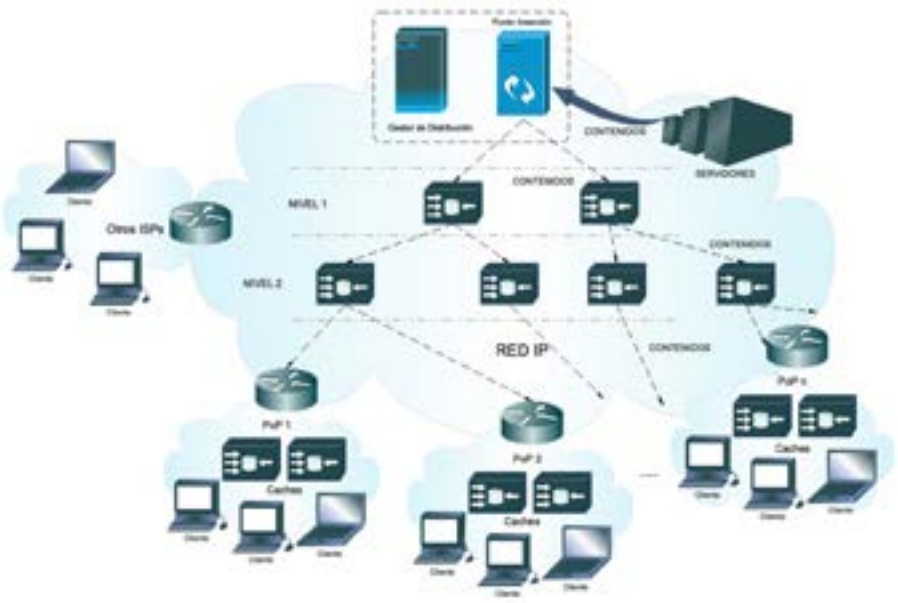
Figura 39.
Distribución de POPs en la red

Distribución de contenidos basada en multicast

En la solución anterior existe un problema fundamental, sobre todo cuando se trata de un mismo contenido que se distribuye a la vez a muchos destinos: la transmisión física de los ficheros desde un repositorio principal hasta elementos extremos.

Multicast es un modo de comunicación entre un emisor y varios receptores. Se caracteriza porque permite transmitir de manera única un mismo paquete a varios receptores simultáneamente, sin necesidad de repetir el envío para cada uno de ellos. Frente al *broadcast*, que es la otra posibilidad de transmisiones punto a multipunto que soportan los protocolos IP, en *multicast* los envíos se pueden realizar a un grupo determinado de receptores, no necesariamente a todos los que existan en la red. Por tanto, esta funcionalidad se hace interesante para aplicaciones de difusión controlada de información (transmisión de una misma información a un grupo determinado de receptores).

Figura 40.
Distribución de contenidos general



La distribución de contenidos tiene fuertes requisitos de calidad, un contenido que se descarga a un servidor debe estar íntegro e incorrupto, puesto que cualquier defecto heredado de la fase de distribución afectará a todos los usuarios que accedan a él posteriormente (Ver Figura 40). Por tanto, si se quiere utilizar *multicast* en la distribución de contenidos parece lógico pensar en la necesidad de disponer de capas superiores que implementen un cierto control para garantizar las transmisiones.

9.5. Dimensionamiento de las redes con servicio de entrega de mejor esfuerzo para proporcionar QoS

En esta sección se analiza cuánta capacidad se debe proporcionar en los enlaces de la red con cierta topología. Por ellos se revisará aspectos relevantes como:

- Aprovisionamiento de ancho de banda
- Dimensionamiento de la red

En estos temas abordados en el texto básico páginas 591 y 592, se trata de resolver problemas que permitan predecir el rendimiento de nivel de aplicación entre dos puntos terminales de una red, como son:

- Modelos de demanda de tráfico entre puntos terminales de la red.
- Requisitos de rendimiento bien definidos.
- Modelos para predecir el rendimiento terminal a terminal para un modelo de carga de trabajo determinado, junto con técnicas para encontrar una asignación de coste mínimo del ancho de banda que permita satisfacer los requisitos de todos los usuarios.

En esta Unidad hemos visto cuán importante es el análisis del servicio de entrega del mejor esfuerzo y el dimensionamiento para proporcionar QoS; el siguiente cuestionario como autoevaluación le ayudará a recoger temas conceptuales importantes de este capítulo.

Autoevaluación 9

Lea detenidamente cada enunciado y conteste con (V) en el caso de ser verdadero y con (F) si es falso:

1. ( ) El servicio de entrega de mejor esfuerzo puede conducir a la pérdida de paquetes.

2. ( ) La mayoría de aplicaciones de telefonía por Internet se ejecutan sobre UDP y se preocupan de retrasmitir los paquetes perdidos.

3. ( ) Se pueden tolerar tasas de pérdida de paquetes entre el 1 y 20% sin depender de cómo se codifique y transmita la voz.

4. ( ) El retardo terminal a terminal es la suma de los retardos de transmisión, de procesamiento y de puesta en cola de los *routers*.

5. ( ) Un componente crucial del retardo terminal a terminal son los retardos aleatorios de puesta en cola dentro de los *routers*.

6. ( ) Las fluctuaciones no pueden eliminarse utilizando números de secuencia, marcas de tiempo y un retardo de reproducción.

7. ( ) Si un fragmento tiene una marca de tiempo que indica que fue generado en el instante “t”, el receptor reproduce dicho fragmento en el instante “t+q”.

8. ( ) El ajuste adaptativo de los retardos de reproducción al principio de los períodos de conversación no hará que los períodos de silencio del emisor se compriman y estiren.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. ¿Cuál de las siguientes son limitaciones del servicio de mejor esfuerzo?

   a. Pérdida de paquetes
   b. Retardo entre extremos
   c. Fluctuación de paquetes
   d. Todas las anteriores

10. ¿Qué es *multicast*?

   a. Es un modo de comunicación entre un emisor y varios receptores, permitiendo transmitir de manera única un mismo paquete a varios receptores simultáneamente.
   b. Es un modo de comunicación entre un emisor y todos los receptores.
   c. Ninguno de los anteriores.
   d. Todos los anteriores.

Ir al solucionario

Semana 12

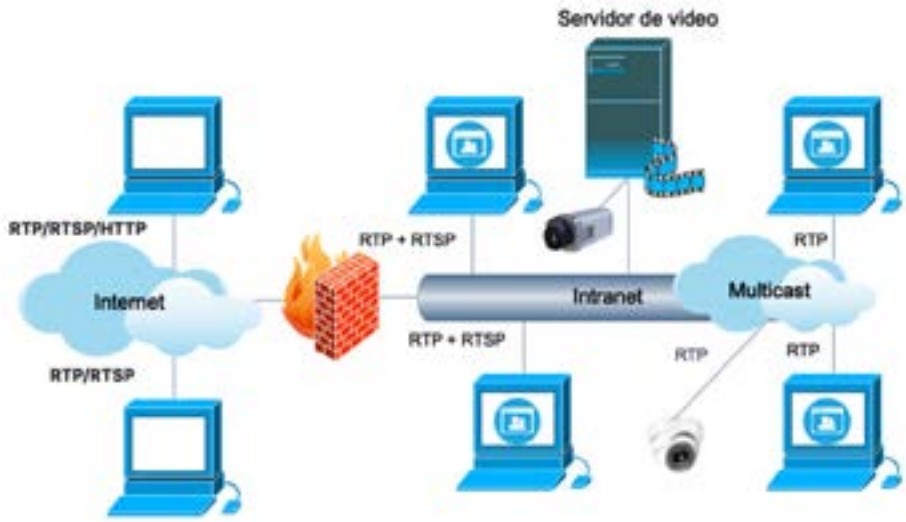
Unidad 10. Protocolos para aplicaciones interactivas en tiempo real

Estimado estudiante, una vez revisado el servicio de entrega del mejor esfuerzo, en esta Unidad procederemos a revisar los Protocolos RTP y RTCP. Estos protocolos fueron creados para transmitir contenidos multimedia.

10.1. RTP (Real-time Transport Protocol)

La mayoría de los protocolos de aplicaciones existentes que usan *multicast* lo hacen sobre UDP. Otras aplicaciones, sobre todo aquellas que tienen que transmitir contenidos multimedia lo hacen usando el protocolo RTP; además de la característica importante de reservar el ancho de banda necesario para la distribución del contenido.

Figura 41.
RTP en una red multiservicios



10.2. Protocolo de control de RTP (RTCP)

Para profundizar más sobre el estudio de este protocolo de control remítase a las páginas 582, 583, 584 y 585 del texto básico, allí se encuentra debidamente explicado sobre su aplicación y las características que lo definen.

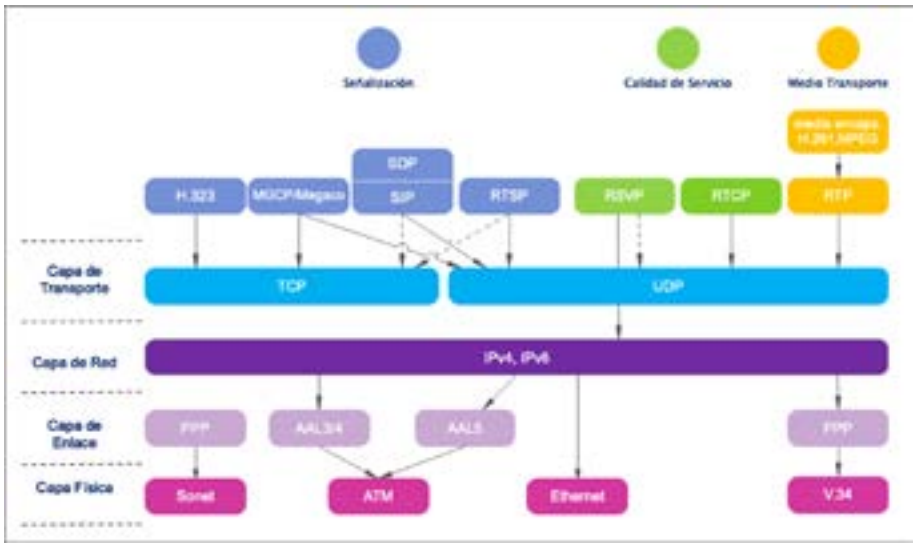
10.3. SIP y H.323

SIP se desarrolla siguiendo los procedimientos del IETF, mientras que H.323 es una recomendación de la ITU-T.

Objetivos de SIP.

- Más integrado con las aplicaciones y servicios Internet.
- Mayor flexibilidad para incorporar nuevas funciones.
- Implementación más simple.

Figura 42.
Funcionamiento del protocolo RTP Y RTCP



La arquitectura IETF es:

- Distribuida.
- Basada en un conjunto de protocolos independientes e intercambiables.
- Flexible, Escalable, Abierta. Compatible con sistemas basados en H.323.
- Funciones de establecimiento, modificación y finalización de sesiones: protocolo SIP.

Arquitectura de los sistemas SIP:

- Integrada en la infraestructura *Web.*
- Modelo cliente–servidor.
- Mensajes de petición y respuesta.
- Reutiliza conceptos de otros servicios (*Web*, correo, dns).
- Agentes de usuario: agentes de usuario clientes (UAC). Agentes de usuario servidores (UAS).
- Servidores: proxys, de registro y de redirección.

Protocolo SIP:

- Sintaxis similar a HTTP o SMTP. Uso de URIs.
- Métodos básicos: INVITE, ACK, BYE, CANCEL, REGISTER, OPTIONS.
- Los mensajes se agrupan en transacciones y llamadas.
- Generalmente, el cuerpo de los mensajes contiene descripciones de sesiones multimedia.
- Códigos de respuesta similares a los de HTTP. Ejemplo: 200–OK.
- Localización basada en el DNS. Cabeceras como método de ampliación.

Diferencias entre SIP y H.323

- Las diferencias entre ambos son consecuencia de las diferencias entre el IETF y la ITU-T.
- Las diferencias en cuanto a servicios soportados se reducen a medida que se desarrollan nuevas versiones.
- Mucha propaganda cuando menos inexacta, incluso desde organizaciones aparentemente rigurosas.
- Errores frecuentes, por ejemplo: SIP es más simple.
- Los análisis comparativos existentes son erróneos o no están actualizados.
- Las comunidades existentes en torno a SIP y H.323 tienen tradiciones distintas.
- H.323 especifica servicios, mientras que SIP es solo un protocolo de señalización para dar base a servicios.
- H.323 engloba un amplio conjunto de protocolos de implementación obligatoria.
- Negociación de capacidades más completa y compleja en H.323.
- H.323 define mecanismos de gestión y administración de la red.
- En la arquitectura SIP, funciones y servicios como garantía de calidad, directorio o descripción de sesiones son ortogonales.
- SIP está integrado en la infraestructura *Web* y proporciona servicios de mensajería instantánea.

- SIP tiene mejores mecanismos de detección de bucles, espirales y otros errores de configuración de la red.
- El 3gpp ha adoptado SIP como protocolo de señalización.
- Desde las primeras versiones, el inicio de llamadas es más rápido con SIP.

Para complementar sus conocimientos es importante que vea este vídeo del canal de YouTube, Rafa Sebastián, sobre Protocolo H.323 y códecs audio y vídeo. En este vídeo tutorial aprenderá a conocer el funcionamiento del protocolo H.323 para telefonía IP, así como el uso de *gatekeepers* y *gateways*.

10.4. Múltiples clases de servicio

10.4.1. Escenarios

Dentro de los escenarios existen muchos mecanismos para dar soporte a múltiples clases de servicio, por lo que antes de analizar los escenarios especificados en el texto básico es importante hacer una síntesis del ítem 9.5.2 de las páginas 592 y 593. Luego de ello debe elaborar un cuadro sinóptico que recoja toda la funcionalidad de los escenarios 1, 2 y 3. No olvidar leer y analizar los principios de cada uno de los escenarios.

10.4.2. Mecanismos de planificación y vigilancia

Un mecanismo de planificación debe:

- Favorecer a los trabajos cortos.
- Favorecer a los trabajos limitados por la E/S para lograr un mejor aprovechamiento de los dispositivos de E/S.

- Determinar la naturaleza de un trabajo lo más pronto posible y planificarlo de acuerdo con su naturaleza.

10.4.3. DiffServ

Diffserv son servicios diferenciados que proporcionan un método que intenta garantizar la calidad de servicio en redes de gran tamaño, como es Internet.

Diffserv analiza varios flujos de datos en vez de conexiones únicas o reservas de recursos. Esto implica que una negociación será hecha para todos los paquetes que envía una organización, ya sea una universidad, un proveedor de servicios de Internet o una empresa. Los contratos resultantes de esas negociaciones son llamados Acuerdos de Nivel de Servicio (SLA), e inevitablemente implican un intercambio oneroso.

Estos SLA especifican qué clases de tráfico serán provistos, qué garantías se dan para cada clase y cuántos datos se consideran para cada clase.

La esencia principal de DiffeServ consiste en dividir el tráfico en múltiples clases y tratarlas de diferente forma. DiffServ renombra al campo ToS (IP) como DS Field (Differented Services Field). Por ejemplo, una aplicación de los servicios diferenciados es de utilidad en un Proveedor de Servicios de Internet (ISP), donde el cliente debe tener un SLA (Service Level Agreement) con su ISP. En este caso los servicios son:

- Expedited Forwaring Serveces (Premium): servicio con confiabilidad, baja demora y bajo Jitter (versión de la demora).
- Gold, Silver y Bronze (Assured): servicio con confiabilidad y cierto tiempo de transmisión.
- Best Effort: servicio tradicional de Internet.

Autoevaluación 10

Ahora lo invitamos a revisar los conocimientos adquiridos, si su nota es baja por favor vuelva a leer y revisar los contenidos.

Le invitamos a resolver las siguientes preguntas que le servirán para determinar las áreas en las cuales ha tenido mayor dificultad y necesitan una nueva revisión.

Lea detenidamente y responda según corresponda. Conteste con una V si es verdadero o F si es falso.

1. ( ) Una aplicación multimedia añade campos de cabecera a los fragmentos de audio/ vídeo antes de pasarlos a la capa de transporte.
2. ( ) RTP proporciona mecanismos para garantizar la entrega a tiempo de los datos.
3. ( ) Para un flujo de vídeo , el tipo de carga útil se utiliza para indicar el tipo de codificación de vídeo.
4. ( ) La API de UDP requiere que el proceso emisor establezca, para cada segmento UDP que envía, la dirección de destino IP.
5. ( ) Los paquetes RTCP encapsulan fragmentos de audio y vídeo.
6. ( ) Cada flujo RTP que transmite un emisor, crea y transmite paquetes de descripción del origen.
7. ( ) RTCP no tiene un potencial problema de escalado.

8. ( ) Los mensajes SIP son mensajes legibles ASCII y son parecidos a los mensajes HTTP.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. ¿De los siguientes estándares cuál es empleado en aplicaciones de tiempo real?

   a. 802.11
   b. SIP
   c. Ninguno de los anteriores
   d. Todos los anteriores

10. ¿De las siguientes funciones cuál está relacionado con SIP?

   a. Proporciona mecanismos para establecer llamadas sobre una red IP.
   b. Proporciona mecanismos para que el que llama determine la dirección IP actual del que es llamado.
   c. Proporciona mecanismos para la gestión de llamadas.
   d. Todas las anteriores.

Ir al solucionario

Resultado de aprendizaje 3

- Discute las arquitecturas típicas de gestión de la red.

Contenidos, recursos y actividades de aprendizaje

Semana 13

Unidad 11. Gestión de redes

Estimado estudiante, en esta Unidad encontrará los conocimientos relacionados a la gestión de redes que son necesarios para comprender cómo supervisar, gestionar y controlar los componentes de *hardware* y *software* de una red conociendo su complejidad. Comenzaremos esta Unidad analizando los temas relacionados a la infraestructura para la gestión de redes y el entorno de gestión estándar de Internet.

11.1. Definición de la gestión de redes

Antes de realizar una definición sobre la gestión de redes vale recalcar que el principal objetivo de la gestión de redes es reducir los

costos y riesgos de operación asociados con sistemas distribuidos grandes y complejos.

Así mismo, mencionar que la gestión de redes se centra en la monitorización, interpretación y control de los comportamientos de la red, empleando tecnologías para la gestión de redes y protocolos que permiten integrar todas estas funcionalidades. Actualmente en los últimos años estas herramientas han evolucionado incrementando más características.

Le invito además a revisar la sección 5.7 en la página 348: ¿qué es la gestión de red? del texto básico, donde se realiza una introducción a gestión de redes y su definición.

Una vez que usted ha revisado en el texto básico la definición de gestión de redes y aclarado con algunos ejemplos prácticos, la importancia de esta temática, revisemos nuevamente la definición "La gestión de redes incluye el despliegue, integración y coordinación del *hardware*, *software* y los elementos humanos para monitorizar, probar, sondear, configurar, analizar, evaluar y controlar los recursos de la red para conseguir los requerimientos de tiempo real, desempeño operacional y calidad de servicio a un precio razonable".

Por lo tanto, se requiere de una verdadera necesidad de gestión en entornos multifabricante, multiprotocolo y multitecnología. Y, finalmente de una convergencia de la gestión de redes puramente informáticas y la gestión de redes de telecomunicaciones, las plataformas de gestión pueden proporcionar gestión en ambos ámbitos de redes.

Considerando los requerimientos mencionados anteriormente y la definición, se desprende que existen diversos modelos de gestión de redes. Pero antes que aparecieran estos modelos, en un inicio surgieron aplicaciones que posibilitaban la supervisión remota de

redes. Sin embargo, cada aplicación solo servía para redes que estuvieran compuestas por equipos de un mismo fabricante. Esta es la denominada gestión homogénea.

Dada la evolución de las redes, la heterogeneidad de los recursos se hizo mayor, por lo que se desarrollaron sistemas de gestión heterogénea. Más tarde fue necesario evolucionar hacia los sistemas de Gestión Integrada, que permiten la utilización de un único centro de gestión válido para llevar el control de entornos heterogéneos. Para llegar a estos sistemas era necesaria una estandarización previa de la gestión de red. En la actualidad existen tres modelos fundamentales de gestión integrada:

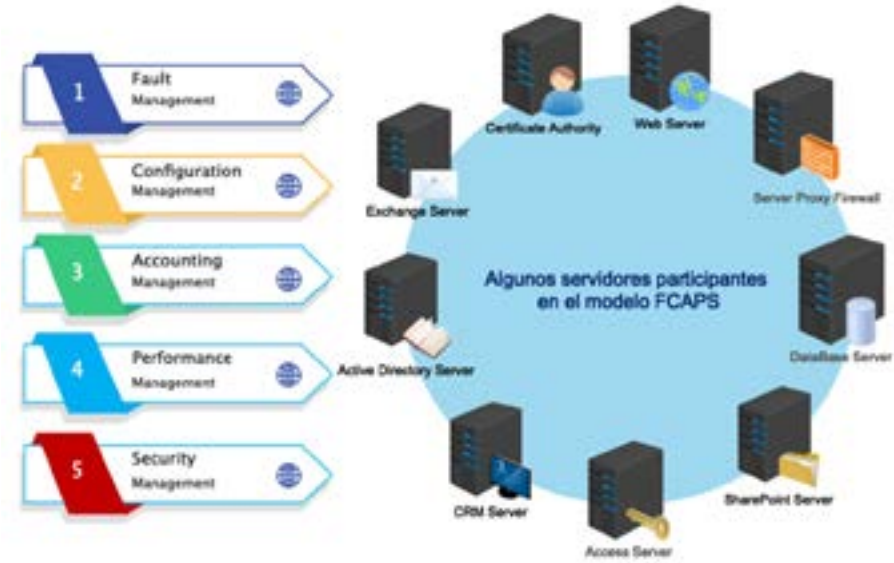
- ***Arquitectura ITU-T***: emplea TMN (Telecommunications Management Network), proporciona funciones de gestión y comunicaciones para la operación de la administración y del mantenimiento de una red de telecomunicaciones y sus servicios en un entorno de múltiples fabricantes.
- ***Arquitectura ISO***: gestión de la torre de protocolos OSI mediante el empleo de CMISE/CMIP (Common Management Information Services Element/ Common Management Information Protocol).
- ***Arquitectura en Internet***: gestión de redes TCP/IP o de Internet, empleando el protocolo SNMP (Simple Network Management Protocol). Hoy en día, SNMP es el entorno de gestión de red más ampliamente utilizado y desarrollado. Y este ha evolucionado actualmente en su versión SNMPv3 con capacidades adicionales de seguridad y administración.

A continuación, se muestra el modelo (arquitectura) de gestión de redes propuesto por la organización Internacional para la Estandarización (ISO), que establece unos componentes, los cuales

se encuentran en la Figura 43 y se detalla la funcionalidad de cada uno:

- Gestión de fallos, en inglés *Fault Management*: alarmas y eventos, diagnóstico y reparación, supervisión de estado de conexiones y equipos.
- Gestión de configuración, en inglés *ConFiguration Management*: control de inventario, configuraciones de *hardware* y *software*, servicio de localización, licencias de *software*.
- Gestión de cuentas, en inglés *Accounting Management*: permite al administrador de red especificar, registrar y controlar el acceso a los usuarios y dispositivos a los recursos de la red.
- Gestión de rendimiento, en inglés *Performance Management*: cuantificar, medir, informar, analizar y controlar el rendimiento de los distintos elementos de la red.
- Gestión de seguridad, en inglés *Security Management*: autenticación de usuarios y acceso a recursos.

Figura 43.
Modelo de gestión de redes funcionales FCAPS definidos por la ITU y OSI



11.2. Componentes para la gestión de una red

Le invitamos a desarrollar la lectura de la sección 5.7.1." El marco conceptual de la gestión de red" del texto básico, donde se expone los componentes para la gestión de redes.

En el modelo FCAPS, uno de sus componentes es la gestión de seguridad. En este vídeo tutorial del canal de YouTube, HTML Rules, sobre Factores que promueven la inseguridad informática, podrá conocer lo fundamental de la seguridad de la información y, en concreto, sobre ciberseguridad y las amenazas a las que nos exponemos hoy en día.

Entre los componentes necesarios de una arquitectura de gestión de redes tenemos:

- ***Entidad gestora***: es el punto central de la administración de una red, también es conocido como el centro de operaciones de red (NOC: Network Operations Center) o servidor de gestión, donde se visualiza la información de los dispositivos gestionados. En la Figura 9.3 se presenta un ejemplo de un NOC de AT&T. También se puede decir que son elementos que interaccionan con los operadores humanos, y desencadenan las acciones pertinentes para llevar a cabo las operaciones solicitadas.

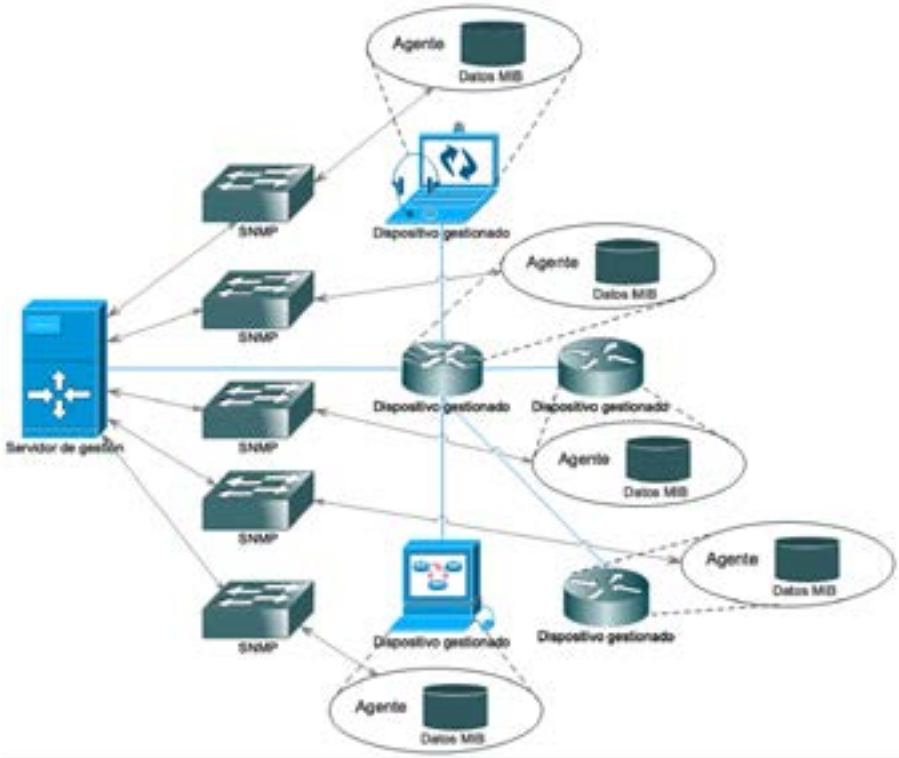
- ***Dispositivos gestionados***: estos son dispositivos activos (*host*, *router*, impresora, modem, hub, etc.) conectados a la red (incluye también el *software*), y contiene uno o más objetos de gestión (tarjeta de red, memoria, pila de protocolo IP, etc.). Estos objetos de gestión tienen información que puede ser adquirida por la entidad gestora y llevan a cabo las operaciones de gestión invocadas por los gestores de la red.

- ***Management Information Base***: conocidos también como Base de la información de gestión, estos recogen a los objetos de gestión, quienes disponen de fragmentos de información y están disponibles para la entidad gestora o servidor de gestión.

- ***Agente de gestión***: en cada dispositivo gestionado existe un agente de gestión de red, que es un proceso residente que se ejecuta en estos dispositivos y se comunica con la entidad gestora, realizando acciones locales bajo el control de los comandos enviados por la entidad gestora o servidor de gestión.

- ***Protocolo de gestión de red***: este se ejecuta entre el servidor de gestión y los dispositivos gestionados. Provee además las reglas de comunicación entre la entidad gestora y los agentes de gestión. Define también tipos de mensaje, seguridad (autenticación, privacidad), manejo de secuencias.

Figura 44.
Componentes de una arquitectura de gestión de red



En la Figura 44 se presentan los componentes necesarios para una arquitectura de gestión de red y cómo están relacionados.

En este mismo contexto se encuentra relacionado el concepto de NOC – (Network Operations Center), que es un centro de operaciones de red, el cual también es conocido como Centro de Control de Red - CCR. La función de estos centros es de monitorizar las redes y atender averías que puedan presentarse. Entre las averías que pueden ser consideradas están: fallos en la red de energía, parámetros de rendimiento y otros que estén relacionados con el mantenimiento de las redes. Estos centros pueden estar

repartidos en diferentes partes de un país de manera estratégica. Por ejemplo, el NOC de AT&T en Bedminister, comenzó en el otoño de 2017 el proyecto de NOC virtual, algunos meses después de probar escritorios remotos en realidad virtual utilizando la pila de escritorio (Ver Figura 45 y Figura 46).

Figura 45.
NOC de AT&T en Bedminister, Nueva Jersey

Extraído de la página web: https://hackmd.io/@XR/noc *[12-12-20]*

Figura 46.
Área de trabajo en el NOC de AT&T en Bedminster

Extraído de la página web: https://www.business.att.com/about/att-corporate-briefing-center.html *[2-01-21]*

Le invitamos a resolver las siguientes preguntas que le servirán para determinar las áreas en las cuales ha tenido mayor dificultad y necesitan una nueva revisión.

Autoevaluación 11

Lea detenidamente cada enunciado y conteste con (V) en el caso de ser verdadero y con (F) si es falso:

1. ( ) La gestión de redes se centra en la monitorización, interpretación y control de los comportamientos de la red, empleando tecnologías para la gestión de redes y protocolos que permiten integrar todas estas funcionalidades.

2. ( ) Para considerar que se está realizando una verdadera gestión de red se requiere de una verdadera necesidad de gestión en entornos multifabricante, multiprotocolo y multitecnología.

3. ( ) Una gestión homogénea es cuando una aplicación solo sirve para redes que estuvieran compuestas por equipos de diferentes fabricantes.

4. ( ) La gestión integrada permite la utilización de un único centro de gestión válido para llevar el control de entornos heterogéneos.

5. ( ) Arquitectura ISO, proporciona funciones de gestión y comunicaciones para la operación de la administración y del mantenimiento de una red de telecomunicaciones y sus servicios en un entorno de múltiples fabricantes.

6. ( ) Arquitectura ITU-T, es la gestión de la torre de protocolos OSI mediante el empleo de CMISE/ CMIP (Common Management Information Services Element/ Common Management Information Protocol).

7. ( ) En la gestión de cuentas, se realiza el control de inventario, configuraciones de *hardware* y *software*.

8. ( ) En la gestión de rendimiento se cuantifica, mide, informa, analiza y controla el rendimiento de los distintos elementos de la red.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. Es un componente necesario de una arquitectura de gestión de redes:

   a. Entidad gestora
   b. MIB
   c. Protocolo de gestión
   d. Todas las otras opciones

10. En cada dispositivo gestionado existe:

    a. Entidad gestora
    b. Agente de gestión
    c. Protocolo de gestión
    d. Ninguna de las otras opciones

Ir al solucionario

Semana 14

Unidad 12. Entorno, estándares y MIB en la gestión de redes

Antes de iniciar con este tema, lo invitamos a revisar el vídeo del canal de YouTube, Net Faculty, sobre SNMP - MIB. En este vídeo tutorial podrá conocer cómo funcionan los MIB (management information base) de SNMP, y porque son importantes. Se analiza además lo que son los OID's, el árbol de jerarquía utilizado, y como están construidos.

12.1. Entorno de gestión estándar de Internet

Este modelo de gestión estándar de Internet tiene sus orígenes en SGMP (Simple Gateway Monitoring Protocol), Protocolo Simple de Monitorización de Pasarela. Posteriormente pasaría a llamarse SNMP.

A finales de la década 1980 el SNMP se convirtió en el estándar de las redes TCP/IP y de Internet. Este ha evolucionado en tres revisiones principales que actualmente se denomina SNMPv3 y se encuentra especificado en el documento RFC 3410.

Le invitamos a dar lectura la sección 9.3 del texto básico: "*Entorno de gestión estándar de Internet"*, donde se define los objetos de gestión de red, un lenguaje definición de datos, un protocolo para transmitir información entre una entidad gestora y un agente que se ejecuta en un dispositivo de red gestionado, y finalmente capacidades de administración y seguridad.

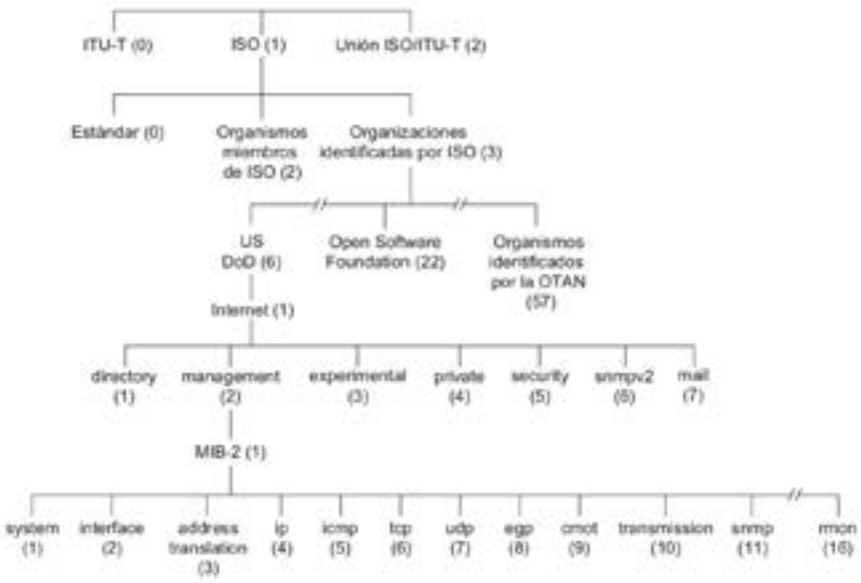
Luego de haber realizado la lectura anterior, a continuación, reforzaremos las partes que conforman el entorno de gestión estándar de Internet:

- ***Objetos de gestión de red***, conocidos como objetos MIB. En la arquitectura de gestión de Internet, la información de gestión se representa mediante un conjunto de objetos conocido como MIB. Por lo tanto, los objetos MIB definen la información de gestión que mantiene un dispositivo gestionado.
- ***Lenguaje de definición de datos***, denominado SMI (estructura de la información de gestión: Structure of Management Information), este es un lenguaje de definición de datos que elimina la ambigüedad en la sintaxis y semántica de los datos. Y define los tipos de datos, modelo de objetos y reglas para escribir y comprobar la información de gestión. Es decir, que los objetos MIB con SMI.
- ***Protocolo SNMP***, para transmitir información y comandos entre la entidad gestora y un agente que se ejecuta en un dispositivo de red.
- ***Capacidad de administración y seguridad***. Este componente es adicional con la mejora de SNMPv3.

12.2. Base de Información de gestión (MIB)

Como es de su conocimiento una MIB es un tipo de base de datos que contiene información jerárquica, estructurada en forma de árbol, de todos los dispositivos gestionados en una red de comunicaciones (Ver Figura 47).

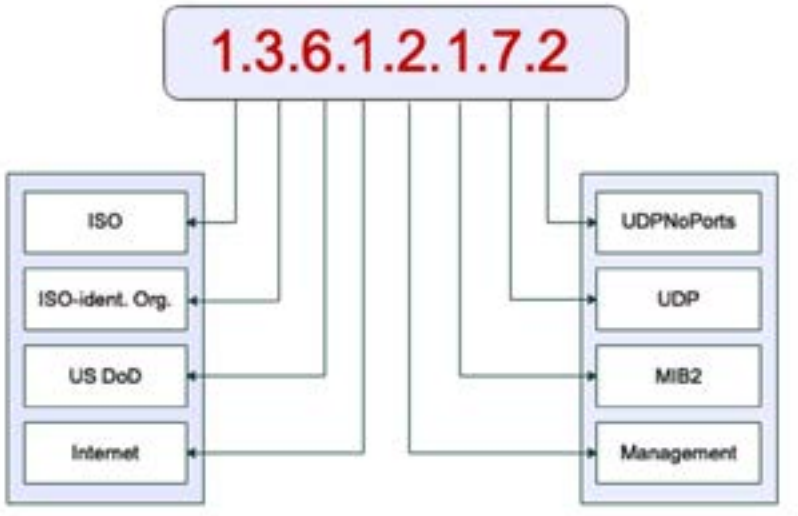
Figura 47.
Árbol de identificación de objetos ASN.1



Kurose & Ross, 2010.

Es parte de la gestión de red definida en el modelo ISO. Define las variables usadas por el protocolo SNMP para supervisar y controlar los componentes de una red. Está compuesta por una serie de objetos que representan los dispositivos (*routers*, *host*, *switch*, *hub,* etc.) en la red. El entorno de identificación de objetos adoptado por ISO es parte del lenguaje de definición ASN.1 (Notación de sintaxis abstracta uno: Abstract Syntax Notation One).

Figura 48.
Ejemplo de árbol de objeto identificador ISO



En la Figura 47 los objetos se nombran en forma jerárquica. Observe que cada punto de ramificación tiene un nombre y un número (identificado entre paréntesis), por lo tanto, cualquier punto del árbol puede ser identificado mediante la secuencia de nombres o números como en la Figura 48, que especifica el camino desde la raíz hasta un punto determinado del árbol de identificación.

12.3. Seguridad y Administración

El SNMPv3 actualmente proporciona mecanismos de seguridad como cifrado, autenticación, protección contra ataques por reproducción y control de acceso. A continuación, revisemos estos mecanismos:

- Cifrado: para cifrar la PDU SNMP emplea el estándar DES.

- Autenticación: emplea la técnica MAC para proporcionar tanto autenticación como protección frente a falsificaciones.
- Protección frente ataques por reproducción: SNMP asegura que un mensaje recibido no es una reproducción de algún mensaje anterior, el receptor requiere que el emisor incluya un valor en cada mensaje.
- Control de acceso: emplea un mecanismo basado en vistas, que permite controlar qué información de gestión de red puede ser consultada y/o definida por qué usuario.

Actividad de aprendizaje recomendada

Conteste las siguientes preguntas: ¿qué herramientas usar para gestionar mi red?, ¿cuáles son las herramientas de mayor relevancia para gestionar una red?, ¿identificar la importancia de la gestión de redes?

Le invitamos a resolver las siguientes preguntas que le servirán para determinar las áreas en las cuales ha tenido mayor dificultad y necesitan una nueva revisión.

Autoevaluación 12

Lea detenidamente cada enunciado y conteste con (V) en el caso de ser verdadero y con (F) si es falso:

1. ( ) La gestión de redes incluye el despliegue, integración y coordinación del *hardware*, *software*.
2. ( ) Existen tres componentes principales en una arquitectura de gestión que son: entidad gestora, dispositivos gestionados y protocolo.
3. ( ) Un dispositivo gestionado solo tiene un objeto de información.
4. ( ) Actualmente los protocolos de gestión como el SNMP3 no integra seguridad.
5. ( ) Para la interacción entre entidad gestora y dispositivo gestionado emplean un protocolo.
6. ( ) Las arquitecturas de gestión de redes actuales permiten realizar una gestión homogénea.
7. ( ) Mediante un árbol de identificación de objetos ASN.1 nos permite acceder a un específico objeto información.
8. ( ) SNMP es el protocolo más difundido y utilizado en la actualidad.

Lea detenidamente cada enunciado y seleccione la respuesta correcta según corresponda:

9. ¿Qué RFC determina los módulos MIB estandarizados?

   a. 3000
   b. 3001
   c. 2700
   d. 300

10. ¿Qué capacidades adicionales integra SNMPv3?

   a. Seguridad
   b. Administración
   c. Ninguna de las anteriores
   a. Todas las anteriores

Verifique sus respuestas en el solucionario que se encuentra al final de la presente guía didáctica.

Ir al solucionario

Resultado de aprendizaje 1, 3 y 4

- Discute las arquitecturas típicas de gestión de la red.
- Esquematiza estrategias de seguridad básica en redes de computadoras.
- Describe el funcionamiento de redes multimedia y de tiempo real.

Contenidos, recursos y actividades de aprendizaje

Semana 15 y 16

REPASO DE UNIDADES 7-12

Estimado estudiante, en esta semana lo invitamos a revisar los contenidos estudiados en el segundo bimestre. Específicamente, deberá revisar las unidades 7 a la 12. Esta revisión le permitirá reforzar los conocimientos adquiridos, lo cual lo preparará para la evaluación del segundo bimestre.

También le recordamos que puede conectarse al chat de la tutoría para cualquier inquietud que tenga en el momento de revisar los contenidos del segundo bimestre. Además, no olvide repasar las autoevaluaciones y ejercicios planteados en las Unidades antes mencionadas.

4. Solucionario

Primer bimestre

Autoevaluación 1	
Pregunta	**Respuesta**
1	c
2	c
3	a
4	b
5	c
6	c
7	b
8	a
9	b
10	c

Ir a la autoevaluación

Autoevaluación 2	
Pregunta	**Respuesta**
1	b
2	c
3	a
4	a
5	a
6	c
7	d
8	b
9	d
10	d

Ir a la autoevaluación

Autoevaluación 3	
Pregunta	**Respuesta**
1	d
2	b
3	d
4	c
5	b
6	a
7	b
8	d
9	b
10	a

Ir a la autoevaluación

Autoevaluación 4	
Pregunta	**Respuesta**
1	b
2	d
3	a
4	c
5	b
6	b
7	d
8	c
9	a
10	d

Ir a la autoevaluación

Autoevaluación 5	
Pregunta	**Respuesta**
1	V
2	V
3	F
4	F
5	V
6	F
7	V
8	V
9	a
10	b

Ir a la autoevaluación

Autoevaluación 6	
Pregunta	**Respuesta**
1	V
2	V
3	V
4	F
5	F
6	F
7	V
8	F
9	d
10	d

Ir a la autoevaluación

Segundo bimestre

Autoevaluación 7	
Pregunta	**Respuesta**
1	F
2	V
3	F
4	V
5	V
6	V
7	F
8	V
9	d
10	a

Ir a la autoevaluación

Autoevaluación 8	
Pregunta	**Respuesta**
1	F
2	V
3	F
4	V
5	F
6	F
7	V
8	F
9	d
10	b

Ir a la autoevaluación

Autoevaluación 9	
Pregunta	**Respuesta**
1	V
2	F
3	F
4	V
5	V
6	F
7	V
8	F
9	d
10	a

Ir a la autoevaluación

Autoevaluación 10	
Pregunta	**Respuesta**
1	V
2	F
3	V
4	V
5	F
6	V
7	F
8	V
9	b
10	d

Ir a la autoevaluación

Autoevaluación 11	
Pregunta	**Respuesta**
1	V
2	V
3	F
4	V
5	F
6	F
7	F
8	V
9	d
10	b

Ir a la autoevaluación

Autoevaluación 12	
Pregunta	**Respuesta**
1	V
2	V
3	F
4	F
5	V
6	F
7	V
8	V
9	a
10	d

Ir a la autoevaluación

5. Referencias bibliográficas

Bermudez, D. (2018a). *Correo electrónico seguro.* Recuperado de: https://aprender-libre.com/2018/06/correo-electronico-seguro/# [7-01-21]

Bermudez, D. (2018b). *El protocolo SSH*. Recuperado de: https://aprender-libre.com/2018/12/el-protocolo-ssh/#

CISCO. (2019c). CCNA 3: Scaling Networks v6.0

Depogi. (2020). *Web navegador UC Browser Logotipo de Internet*. *Freepng.* Recuperado de: https://www.freepng.es/png-36x509/ [01-02-21]

García, F. (2018). *El rating de la información en internet: de la sociedad del conocimiento al big data*. Universidad Complutense de Madrid. España.

Jimenez, J. (2021). *Por qué es esencial corregir las vulnerabilidades de DNSPooq y similares. RZ-Redes Zone*. Recuperado de: https://www.redeszone.net/tutoriales/seguridad/dnspooq-vulnerabilidades-ataques-dns/ [21-02-21]

Kurose, J. & Ross, K. (2017). *Redes de computadoras: Un enfoque descendente*. In PEARSON Educación (7ma ed.). Pearson, Madrid – España.

Kurose, J. & Ross, K. (2010). *Redes de computadoras: Un enfoque descendente*. Quinta edición. Pearson Educación, España.

Martínez Diaz, R. (s.f.). *Protocolo de transferencia de ficheros (FTP).* Universidad Politécnica de Valencia. Recuperado de: http://personales.upv.es/rmartin/TcpIp/cap04s04.html [01-01-21]

Qwerty (2021). Cifrado de Lorenz – Lorenz cipher [En línea]. Disponible en: https://es.qaz.wiki/wiki/Lorenz_cipher [Consulta 10-02-2021]

Recommendation G.711: *"Pulse Code Modulation (PCM) of voice frequencies".* ITU-T. Noviembre 1988.

Recommendation G.722 Amendment 1: "7 kHz audio-coding within 64 kbit/s Amendment 1: New Annex B with superwideband embedded extension". ITU-T. noviembre 2010.

Tanenbaum, A. & Wetherall, D. (2012). *Redes de computadoras*. Quinta edición. Pearson Educación, México.

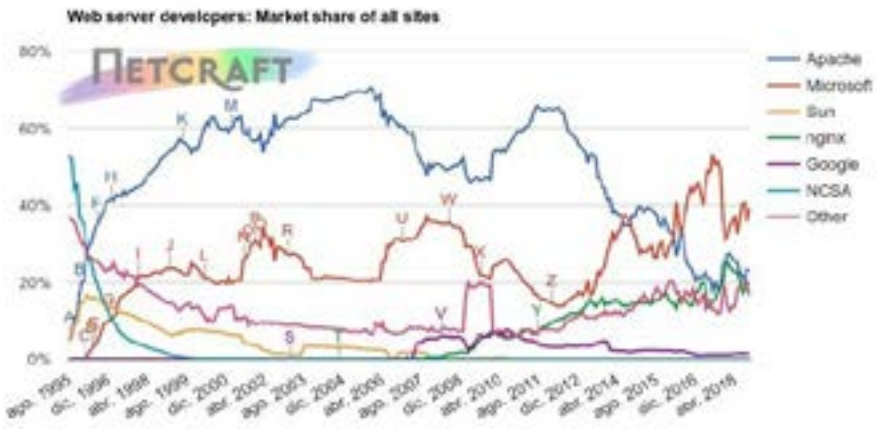
6. Anexos

Anexo 1. Servidores *web* más utilizados

Dentro de todos estos servidores *web* que mencionamos antes, existen varios que gozan de mayor popularidad, tanto por antigüedad, como por rendimiento o tecnologías que soportan; vamos a ver ahora cuáles son los servidores *web* más usados.

Este listado de servidores *web* más utilizados lo hemos obtenido del reporte oficial de Netcraft de octubre de 2018 (el más reciente al momento de escribir este artículo), en él se describen cuáles son los servidores más usados, y qué cuota de uso tienen en el mercado global.

Figura 49.
Estadísticas de los servidores más utilizados hasta el 2018



Developer	September 2018	Percent	October 2018	Percent	Change
Microsoft	585,892,927	35.67%	656,395,388	39.22%	3.55
Apache	384,521,189	23.41%	384,514,944	22.96%	-0.44
nginx	319,330,263	19.44%	330,074,074	19.72%	0.28
Google	22,210,093	1.35%	23,620,555	1.41%	0.06

De acuerdo a la Figura 55, los servidores más utilizados, el veredicto es el siguiente.

Los servidores más utilizados en el 2018 [1]:

- Microsoft IIS
- Apache
- Nginx
- Google GWS

1 https://blog.infranetworking.com/tipos-de-servidores-web/#Servidores_web_mas_utilizados

Como vemos en esta gráfica y la correspondiente tabla, el claro dominante del mercado hoy por hoy es Microsoft IIS, que ha ganado mucho terreno últimamente frente a su rival clásico Apache, el cual se mantuvo como líder indiscutido durante décadas como el más usado.

Nginx sigue creciendo lentamente, y Google tiene un *market share* realmente pequeño, pues hay que tener en cuenta que es usado primordialmente en las plataformas de Google *search.*

Ir al contenido

Índice
Primer bimestre
Segundo bimestre
Solucionario
Referencias bibliográficas
Anexos
Recursos

Anexo 2. RFC 959_PROTOCOLO DE TRANSFERENCIA DE FICHEROS (FTP)

Network Working Group J. Postel

Request for Comments: 959 J. Reynolds ISI

Obsoletes RFC: 765 (IEN 149) octubre 1985

Traducción al castellano: febrero 2000

Gonzalo Paniagua Javier <gpanjav@jazzfree.com>

PROTOCOLO DE TRANSFERENCIA DE FICHEROS (FTP)

Estado de este Documento

Este documento es la especificación oficial del Protocolo de Transferencia de Ficheros (File Transfer Protocol, FTP). Se permite la distribución ilimitada de este documento.

Las siguientes nuevas órdenes opcionales se incluyen en esta edición de la especificación:

CDUP (cambiar al directorio padre), SMNT (montar estructura), STOU (guardar con nombre único), RMD (borrar directorio), MKD (Make Directory), PWD (mostrar directorio actual), y SYST (sistema).

Esta especificación es compatible con ediciones anteriores.

1. INTRODUCCIÓN

Los objetivos del FTP son 1) promocionar el uso compartido de ficheros (programas y/o datos), 2) animar al uso indirecto o implícito (a través de programas) de servidores remotos, 3) hacer transparente al usuario las variaciones entre la forma de almacenar

ficheros en diferentes ordenadores, y 4) transferir datos fiable y eficientemente. El FTP, aunque puede ser utilizado directamente por un usuario en un terminal, está diseñado principalmente para ser usado por programas.

Con esta especificación se intentan satisfacer las diversas necesidades de los usuarios de maxi-hosts, mini-hosts, estaciones de trabajo personales y TAC's con un diseño de protocolo simple y fácil de programar.

En este documento se asumen conocimientos del Protocolo de Control de Transmisión (TCP, Transmision Control Protocol) [2] y del Protocolo Telnet [3]. Estos documentos se encuentras en el manual de protocolos de ARPA-Internet.

:::::

:::::

4. FUNCIONES DE TRANSFERENCIA DE FICHEROS

El canal de comunicación entre el user-PI y el server-PI se establece como una conexión TCP desde el usuario al puerto estándar. El intérprete de protocolo de usuario es responsable de enviar órdenes FTP e interpretar las respuestas recibidas; el server-PI interpreta las órdenes, envía respuestas y controla su DTP para establecer la conexión de datos y transferirlos. Si el proceso de transferencia pasiva es el user-DTP, entonces está controlado a través del protocolo interno del ordenador donde se ejecuta el user-DTP; si es otro server-DTP, está controlado a través de órdenes recibidas en su PI desde el user-PI. Las respuestas FTP se tratan en la siguiente sección.

Al describir algunas de las órdenes en esta sección, viene bien ser explícito con las posibles respuestas.

1.1. ÓRDENES FTP

4.a.1. ÓRDENES DE CONTROL DE ACCESO

Las siguientes órdenes especifican identificadores de control de acceso (los códigos de las órdenes están entre paréntesis).

NOMBRE DE USUARIO (USER)

El argumento es una cadena Telnet que identifica al usuario. Esta identificación es la que requiere el servidor para acceder a su sistema de ficheros. Normalmente esta será la primera orden a transmitir una vez establecida la conexión de control (algunos ordenadores lo pueden requerir). El servidor puede requerir información adicional como una contraseña y/o cuenta. Los servidores pueden permitir una nueva orden USER en cualquier momento para cambiar el control de acceso y/o la información de la cuenta. Esto tiene el efecto de descartar cualquier información anterior sobre usuario, contraseña y cuenta, y comienza la secuencia de acceso otra vez. Todos los parámetros de la transferencia permanecen sin cambios, y cualquier transferencia de fichero en curso se completa bajo los anteriores parámetros de control de acceso.

CONTRASEÑA (PASS)

El argumento es una cadena Telnet especificando la contraseña del usuario. Esta orden debe ir inmediatamente precedida por la orden USER y, para algunos ordenadores, completa la identificación del usuario para el control de acceso. Como la información de la contraseña es un dato confidencial, es preferible, en general, "enmascararla" o evitar mostrarla en pantalla. Parece que el servidor no tiene un medio a prueba de tontos para conseguir esto. Por tanto, es responsabilidad del proceso user-FTP el ocultar la información sobre la contraseña.

CUENTA (ACCT)

El argumento es una cadena Telnet identificando la cuenta del usuario. Esta orden no está necesariamente relacionada con la orden USER, ya que algunos ordenadores pueden requerir una cuenta para acceder y otros solo para cierto tipo de acceso, como almacenar ficheros. En este último caso, la orden se puede enviar en cualquier momento.

Hay códigos de respuesta para diferenciar automáticamente estos casos: cuando se requiere información de la cuenta, la respuesta a una orden PASS correcta es el código 332. Por otra parte, si NO se requiere esta información, la respuesta a una orden PASS correcta es 230; y si la cuenta se requiere para una orden enviada más tarde, el servidor debería devolver una respuesta 332 o una 532 dependiendo de qué almacene (esté pendiente de recibir el comando ACCT) o descarte la orden, respectivamente.

CAMBIO DE DIRECTORIO DE TRABAJO (CWD)

Esta orden permite al usuario trabajar en un directorio o conjunto de datos [data set] diferente, para almacenar o recuperar información sin alterar su información de entrada o de cuenta. Los parámetros de transferencia permanecen sin cambios. El argumento es un nombre de ruta especificando el directorio o alguna otra agrupación de ficheros dependiente del sistema.

CAMBIAR AL DIRECTORIO PADRE (CDUP)

Esta orden es un caso especial de CWD, y se incluye para simplificar la implementación de programas para transferir árboles de directorios entre sistemas operativos que tienen diferentes formas de nombrar al directorio padre. Los códigos de respuestas deberán ser idénticos a los de CWD. Vea el Apéndice II para más detalles.

MONTAR ESTRUCTURA (SMNT)

Esta orden permite al usuario montar un sistema de ficheros diferente sin alterar su información de entrada o de cuenta. Los parámetros de transferencia permanecen sin cambios. El argumento es un nombre de ruta especificando un directorio o alguna otra agrupación de ficheros dependiente del sistema.

REINICIALIZAR (REIN)

Esta orden termina una orden USER, descargando todos los datos de entrada/salida y la información de cuenta, excepto que si hay alguna transferencia en proceso permite que termine. Todos los parámetros se inician con sus valores por defecto y la conexión de control se deja abierta. El estado alcanzado es idéntico al que se tiene inmediatamente después de abrir la conexión de control. Posiblemente se espere una orden USER a continuación de esta.

DESCONECTAR (QUIT)

Esta orden termina una orden USER, y si no hay en proceso ninguna transferencia, cierra la conexión de control. Si hay una transferencia de fichero en proceso, la conexión permanecerá abierta hasta que el servidor envíe una respuesta con el resultado de la transferencia y luego se cierra. Si el proceso de usuario está transfiriendo ficheros para varios usuarios (USERs) pero no quiere cerrar la conexión y reabrirla para cada usuario, se debería usar el comando REIN en lugar de QUIT. Un cierre inesperado de la conexión de control provoca que el servidor actúe como si hubiera recibido las órdenes interrumpir (ABOR) y desconectar (QUIT).

4.a.2. ÓRDENES DE PARÁMETROS DE TRANSFERENCIA

Todos los parámetros de transferencia de datos tienen valores por defecto, y las órdenes que especifican valores para ellos solo

se deben utilizar si se van a cambiar los parámetros por defecto. El valor por defecto es el último valor especificado o, si no se ha especificado ninguno, el valor por defecto estándar es el que se indica aquí. Esto implica que el servidor debe "recordar" los valores por defecto aplicables. Las órdenes pueden ir en cualquier orden, pero deben preceder a la transferencia. Las siguientes órdenes especifican parámetros de transferencia de datos:

PUERTO DE DATOS (PORT)

El argumento es una especificación ordenador-puerto, para el puerto que será usado en la conexión de datos. Hay valores por defecto tanto para el puerto de usuario como para el del servidor y, bajo circunstancias normales, esta orden y su respuesta no son necesarias. Si se usa esta orden, el argumento es la concatenación de una dirección IP (32 bits) y un puerto TCP (16 bits). Esta información está repartida en campos de 8 bits y el valor de cada campo se transmite como un número decimal (representado como una cadena de caracteres). Los campos están separados por comas. Una orden PORT podría ser algo así:

PORT h1, h2, h3, h4, p1, p2; donde h1 es el número decimal correspondiente a los 8 bits más altos de la dirección IP del ordenador. PASIVO (PASV)

Esta orden solicita al server-DTP que "escuche" en un puerto de datos (que no es el puerto por defecto) y espere a recibir una conexión en lugar de iniciar una al recibir una orden de transferencia. La respuesta a este comando incluye la dirección IP y el puerto donde este servidor está esperando a recibir la conexión.

TIPO DE REPRESENTACIÓN (TYPE)

El argumento especifica un tipo de representación tal y como se describió en la sección Representación de Datos y Almacenamiento.

Algunos tipos requieren un segundo parámetro. El primer parámetro es un único carácter Telnet, y el segundo, para ASCII y EBCDIC, también; el segundo parámetro para tamaño de byte local es un entero decimal. Los parámetros están separados por un <SP> (espacio, código ASCII 32).

Se asignan los siguientes códigos para tipos:

\ /

A - ASCII | | N - No para imprimir

|-><-| T - Formateo con caracteres Telnet

E - EBCDIC| | C - Control de carro (ASA)

/\

I - Imagen

L <tamaño de byte> - Tamaño de byte local

La representación por defecto es ASCII no para imprimir. Si se cambia el parámetro de formato y más tarde solo el primer argumento, el formato vuelve a ser el inicial por defecto (no para imprimir).

ESTRUCTURA DEL FICHERO (STRU)

El argumento es un único carácter Telnet especificando una estructura de fichero de las descritas en la sección “Representación de datos y almacenamiento”.

Existen los siguientes códigos para la estructura:

F - Fichero (sin estructurar en registros) R - Estructurado en registros P - Estructurado en páginas. La estructura por defecto es Fichero.

MODO DE TRANSFERENCIA (MODE)

El argumento es un único carácter Telnet especificando un modo de transferencia de los descritos en la sección “Modos de transmisión”.

Los posibles códigos son los siguientes:

S - Flujo

B - Bloque

C - Comprimido

El modo por defecto es Flujo.

4.a.3. ÓRDENES DE SERVICIO FTP

Las órdenes de servicio FTP definen la transferencia del fichero o la función del sistema de ficheros que requiere el usuario. El argumento normalmente será un nombre de ruta. La sintaxis del nombre de ruta debe seguir las convenciones del servidor (con valores estándar por defecto aplicables) y las convenciones del lenguaje usado en la conexión de control. Se sugiere que por defecto se utilice el último dispositivo, directorio o nombre de fichero o el valor por defecto para los usuarios locales. Las órdenes pueden enviarse en cualquier secuencia, excepto que la orden “renombrar” debe ir seguida por una “renombrar a” y la orden “recomenzar” debe ir seguida por la orden de servicio interrumpida (por ejemplo, STOR o RETR). Cuando se transfieren datos se debe hacer siempre a través de la conexión de datos, excepto para algunas respuestas informativas. Las siguientes órdenes especifican peticiones de servicio FTP.

RECUPERAR (RETR)

Esta orden hace que el server-DTP transfiera una copia del fichero especificado en el nombre de ruta al proceso que está al otro lado de la conexión de datos. El estado y el contenido del fichero en el servidor debe permanecer tal y como estaba.

ALMACENAR (STOR)

Esta orden hace que el server-DTP lea los datos transferidos por la conexión de datos y los guarde en un fichero en el servidor. Si el fichero especificado en el nombre de ruta existe en el servidor, su contenido se debe reemplazar con los datos recibidos. Se crea un fichero nuevo en el servidor si el indicado no existía ya.

ALMACENAR ÚNICO (STOU)

Esta orden se comporta igual que STOR, solo que el fichero resultante se crea en el directorio actual con un nombre único para ese directorio. La respuesta 250 Transferencia iniciada debe incluir el nombre generado.

AÑADIR (con creación) (APPE)

Esta orden hace que el server-DTP reciba datos a través de la conexión de control y los guarde en un fichero en el servidor. Si el fichero especificado en el nombre de ruta existe, los datos se añaden a ese fichero; si no, se crea un fichero nuevo en el servidor.

SOLICITAR ESPACIO (ALLO)

Esta orden puede ser necesaria para que algunos servidores reserven suficiente espacio de almacenamiento para recibir el nuevo fichero. El argumento debe ser un entero decimal, indicando el número de bytes (usando el tamaño de byte lógico) de almacenamiento que se deben reservar. Para ficheros enviados

con estructura en registros o páginas, puede ser necesario enviar el tamaño máximo de registro o página; esto se indica con un entero decimal como segundo argumento de la orden. Este segundo argumento es opcional, pero cuando se utilice, debe estar separado del primero por los caracteres Telnet <SP> R <SP>. A continuación de esta orden se deberá indicar una orden STOR o APPE. La orden ALLO debería tratarse como la orden NOOP (no operación) por los servidores que no necesitan conocer de antemano el tamaño del fichero y aquellos servidores que solo interpreten el tamaño máximo de registro o página deberían admitir un valor inútil como primer argumento e ignorarlo.

RECOMENZAR (REST)

El argumento representa un marcador del servidor, a partir del cual debe recomenzar la transferencia. La orden no realiza la transferencia del fichero, pero hace que el puntero de lectura o escritura del fichero se sitúe a continuación del punto indicado. A continuación de esta orden se debe enviar la orden de servicio FTP apropiada que hará que continúe la transferencia del fichero.

RENOMBRAR DE (RNFR)

Esta orden indica el fichero que queremos cambiar de nombre en el servidor. Debe ir inmediatamente seguida de la orden “renombrar a” con el nuevo nombre para el fichero.

RENOMBRAR A (RNTO)

Esta orden especifica el nuevo nombre para el fichero indicado mediante el comando RNFR. Las dos órdenes seguidas hacen que el fichero cambie de nombre.

INTERRUMPIR (ABOR)

Este comando pide al servidor que interrumpa la orden de servicio FTP, previa y cualquier transferencia de datos asociada. La orden de interrupción puede requerir alguna "acción especial", tratada más adelante, para forzar el reconocimiento de la orden por el servidor. No se hará nada si la orden anterior ha finalizado (incluyendo la transferencia de datos). El servidor no cierra la conexión de control, pero puede que sí cierre la conexión de datos. Hay dos posibles casos para el servidor al recibir esta orden: (1) la orden de servicio FTP está ya terminada, o (2) aún está en ejecución.

En el primer caso, el servidor cierra la conexión de datos (si está abierta) y devuelve una respuesta 226 indicando que la orden de interrumpir se ha procesado correctamente. En el segundo caso, el servidor interrumpe el servicio FTP en proceso y cierra la conexión de datos, devolviendo una respuesta 426 para indicar que la solicitud de servicio terminó anormalmente. Luego, el servidor envía una respuesta 226 para indicar que la orden de interrumpir se ha procesado correctamente.

BORRAR (DELE)

Esta orden borra en el servidor el fichero indicado en el nombre de ruta. Si se quiere tener un nivel extra de protección (del tipo "¿Seguro que quiere borrar el fichero?"), la debería proporcionar el proceso user- FTP.

BORRAR DIRECTORIO (RMD)

Esta orden borra en el servidor el directorio indicado. Vea el apéndice II. CREAR DIRECTORIO (MKD)

Esta orden crea el directorio indicado en el servidor. Vea el apéndice II. MOSTRAR EL DIRECTORIO DE TRABAJO (PWD)

Esta orden hace que el servidor nos devuelva en la respuesta el nombre del directorio actual. Vea apéndice II.

LISTAR (LIST)

Esta orden hace que el servidor envíe un listado de los ficheros a través del proceso de transferencia de datos pasivo. Si el nombre de ruta u otra agrupación de ficheros, el servidor debe transferir una lista de los ficheros en el directorio indicado. Si el nombre de ruta especifica un fichero, el servidor debería enviar información sobre el fichero. Si no se indica argumento alguno, implica que se quiere listar el directorio de trabajo actual o directorio por defecto. Los datos se envían a través de la conexión de datos con tipo ASCII o EBCDIC. (El usuario se debe asegurar del tipo con TYPE). Como la información sobre un fichero puede variar mucho de un sistema a otro, es muy difícil que esta pueda ser procesada automáticamente, pero puede ser útil para una persona.

LISTAR NOMBRES (NLST)

Esta orden hace que se envíe un listado de directorio desde el servidor. El nombre de ruta indica un directorio u otra agrupación de ficheros específica del sistema; si no hay argumento, se asume el directorio actual. Los datos se transfieren en formato ASCII o EBCDIC a través de la conexión de datos separados unos de otros por <CRLF> o <NL>. (Una vez más el usuario se debe asegurar con TYPE). La función de esta orden es devolver información que pueda ser usada por un programa para procesar posteriormente los ficheros automáticamente. Por ejemplo, implementando una función que recupere varios ficheros.

PARÁMETROS DEL SISTEMA (SITE)

Esta orden la usa el servidor para proporcionar servicios específicos propios de su sistema que son fundamentales para transferir

ficheros, pero no lo suficientemente universales como para ser incluidos como órdenes en el protocolo. La naturaleza de este servicio y la especificación de su sintaxis se puede obtener como respuesta a una orden HELP SITE.

SISTEMA (SYST)

Esta orden devuelve el tipo de sistema operativo del servidor. La respuesta debe tener como primera palabra uno de los nombres de sistema listados en la versión actual del documento Números Asignados [4].

ESTADO (STAT)

Esta orden hace que el servidor nos envíe una respuesta con su estado a través de la conexión de control. La orden se puede enviar durante la transferencia de un fichero (junto con las señales Telnet IP y Synch, vea la sección Órdenes FTP) en cuyo caso el servidor responderá con el estado de la operación en progreso, o se puede enviar entre transferencias de ficheros. En este último caso, la orden puede llevar un argumento. Si el argumento es un nombre de ruta, la orden es similar a LIST, excepto que los datos se devolverán por la conexión de control. Si no hay ningún argumento, el servidor devolverá información general del estado del proceso servidor FTP. Esto debería incluir los valores actuales de los parámetros de transferencia y el estado de las conexiones.

AYUDA (HELP)

Esta orden hace que el servidor envíe información sobre la implementación del FTP a través de la conexión de control. La orden puede tener un argumento que debe ser un nombre de orden y así devuelve información más específica como respuesta. La respuesta es de tipo 211 o 214. Se sugiere que se permita el uso de HELP antes del comando USER. El servidor puede usar esta respuesta

para especificar parámetros dependientes del sistema, por ejemplo, en respuesta a HELP SITE.

NO OPERACIÓN (NOOP)

Esta orden no afecta a ningún parámetro ni orden introducida previamente. No hace nada más que provocar que el servidor envíe una respuesta OK.

:::::

:::::

1.2. RESPUESTAS FTP

Las respuestas a órdenes del FTP están pensadas para asegurar la sincronización entre peticiones y acciones en el proceso de transferencia de ficheros y para garantizar que el proceso de usuario siempre conoce el estado del servidor. Cada orden debe generar por lo menos una respuesta, aunque puede haber más de una; en este último caso, las diferentes respuestas se deben distinguir fácilmente. Además, algunos comandos deben ocurrir en secuencia, como USER, PASS y ACCT o RNFR y RNTO. Las respuestas muestran la existencia de un estado intermedio si todos los comandos anteriores han ido correctamente. Un error en cualquier punto de la secuencia implica que se debe iniciar de nuevo desde el principio.

Los detalles de la secuencia orden-respuesta se muestran en un conjunto de diagramas de estado más abajo.

Una respuesta FTP consiste en un número de tres cifras (transmitido como tres caracteres alfanuméricos) seguidos de texto. El número se proporciona para su uso por autómatas que deben determinar el próximo estado; el texto va dirigido a las personas. Se pretende que el número contenga suficiente información codificada como para que el intérprete de protocolo de usuario no necesite examinar

el texto y pueda o bien descartarlo o bien mostrarlo al usuario. En particular, el texto puede depender del servidor y, por tanto, puede variar en cada código de respuesta.

Una respuesta debe contener el código de 3 dígitos, seguidos de espacio <SP>, seguido de una línea de texto (en la que hay definida una longitud máxima), y termina con el código Telnet de fin de línea. Habrá casos en los que el texto es mayor que una línea. En estos casos el texto debe ir marcado para que el proceso de usuario sepa cuando ha terminado la respuesta (i.e., deje de leer de la conexión de control) y haga otras cosas. Esto requiere de un formato especial en la primera línea para indicar que hay más, y otro en la última línea para indicar lo propio. Al menos una de estas debe contener el código de respuesta apropiado para indicar el estado de la transacción. Para satisfacer a todos, se ha decidido que ambas, la primera y la última línea, deben contener el mismo código.

Por tanto, el formato para respuestas multilínea consiste en que la primera línea empieza con el código de respuesta requerido, seguido inmediatamente de un guion, "-" (también es el signo menos), seguido del texto. La última línea empezará con el mismo código, seguido inmediatamente por un espacio <SP>, opcionalmente texto, y el código Telnet de fin de línea. Por ejemplo:

123-Primera línea

Segunda línea

234 una línea que comienza con números

123 la última línea

El proceso de usuario solo necesita buscar la segunda ocurrencia del mismo código de respuesta, seguido de <SP> (espacio), al principio de una línea. Si una línea intermedia comienza con un

número de tres dígitos, el servidor debe desplazar ese número para evitar confusiones.

Este esquema permite que se usen los procedimientos estándar del sistema para la información de respuesta (como, por ejemplo, para STAT), con las líneas primera y última añadidas "artificialmente". En casos raros en los que estos procedimientos generen tres dígitos y un espacio al principio de cualquier línea, el principio de cada línea de texto se debería desplazar con algún texto neutral, como espacios.

Este esquema asume que las respuestas multilínea no pueden estar anidadas.

Cada tres dígitos de la respuesta tienen un significado especial. Se pretende permitir un rango de respuestas desde muy simple a muy sofisticado para el proceso de usuario. El primer dígito denota si la respuesta es buena, mala o incompleta. (Refiriéndonos al diagrama de estado), un proceso de usuario poco sofisticado podrá determinar su próxima acción simplemente examinando el primer dígito. Un proceso de usuario que quiera conocer aproximadamente el tipo de error ocurrido (por ejemplo, error del sistema de ficheros o error de sintaxis) puede examinar el segundo dígito, reservando el tercero para una mayor precisión en la información (por ejemplo, orden RNTO sin ser precedida por una RNFR).

Hay cinco valores para el primer dígito del código de respuesta:

1. yz Respuesta preliminar positiva. Se ha iniciado la acción requerida; se espera otra respuesta antes de seguir con una nueva orden. (El proceso de usuario que envía otra orden antes de que la respuesta de finalización llegue, estará violando el protocolo, pero el proceso servidor debería encolar cualquier orden que reciba mientras está ejecutando una orden anterior.). Este tipo de respuesta se puede usar para

indicar que se ha aceptado la orden y que el proceso de usuario debería ocuparse ahora de la conexión de datos, en implementaciones donde controlar ambas conexiones a la vez es difícil. El proceso servidor puede enviar, a lo sumo, una respuesta 1yz por orden recibida.

2. yz Respuesta de finalización positiva. La acción requerida se ha completado satisfactoriamente. Se puede iniciar una nueva orden.

3. yz Respuesta intermedia positiva. La orden se ha aceptado, pero se está pendiente de recibir más información para completarla. El usuario debería enviar otra orden indicando esta información. Esta respuesta se utiliza en órdenes que deben ir en secuencia.

4. yz Respuesta de finalización negativa transitoria. La orden no se ha aceptado y la acción requerida no se ha llevado a cabo, pero la condición de error es temporal y se puede solicitar la acción de nuevo. El usuario debería volver al principio de la secuencia de comandos, si es que la hay. Es difícil dar un significado a "transitoria", particularmente cuando dos sistemas diferentes de (el servidor y el usuario) deben estar de acuerdo en la interpretación. Cada respuesta de la categoría 4yz puede tener un valor diferente, pero se pretende con ella que el proceso de usuario reintente la operación. Una regla para determinar si una respuesta pertenece a la categoría 4yz o a la 5yz (permanente negativa) es que las respuestas son 4yz, si la orden se puede repetir sin cambios en la forma o en las propiedades del usuario o del servidor (por ejemplo, la orden y sus argumentos son los mismos; el usuario no cambia su cuenta ni su nombre; el servidor no lo interpreta de diferente forma).

5. yz Respuesta de finalización negativa permanente. La orden no se ha aceptado y la acción requerida no ha tenido lugar. El proceso de usuario no debería repetir la misma petición (en la misma secuencia). Incluso algunas condiciones de error “permanente” se pueden corregir, por eso el usuario (la persona) puede hacer posteriormente que su proceso de usuario repita posteriormente la orden (por ejemplo, se corrige el argumento, o el usuario cambia el estado de su directorio).

Las siguientes agrupaciones de funciones se codifican en el segundo dígito: x0z Sintaxis

Estas respuestas se refieren a errores de sintaxis, órdenes correctas sintácticamente, pero que no encajan en ninguna otra categoría, órdenes no implementadas o superfluas.

x1z Información

Estas son respuestas a solicitudes de información como STATUS o HELP.

x2z Conexiones

Respuestas referidas a las conexiones de control y de datos.

x3z Autenticación y cuenta

Respuestas para el proceso de entrada al sistema y procedimientos de cuenta.

x4z Sin especificar aún.

x5z Sistema de ficheros

Estas respuestas indican el estado del sistema de ficheros en el servidor según se realizan transferencias u otras acciones sobre el sistema de ficheros.

El tercer dígito afina más en el significado de cada una de las categorías indicadas por el segundo. La lista de respuestas de la siguiente sección mostrará esto. El texto asociado con cada respuesta es solo una recomendación, no una obligación, y puede incluso cambiar según la orden asociada. Los códigos de respuesta, por otra parte, deben seguir estrictamente las especificaciones; es decir, las implementaciones de servidores no deben inventarse nuevos códigos para situaciones que son ligeramente diferentes a las descritas aquí, sino que deberían adaptarse a los códigos ya definidos.

Una orden como TYPE o ALLO cuya correcta ejecución no proporciona al usuario ninguna nueva información debería devolver un código 200. Si la orden no se ha implementado porque no tiene sentido para un determinado sistema, por ejemplo, ALLO en un TOPS-20, una respuesta de finalización positiva es conveniente para no confundir al proceso de usuario. En este caso se usa una respuesta 202 con, por ejemplo, el siguiente texto: “No es necesario solicitar espacio para el fichero.” Si, por otra parte, la orden no es específica a ningún sistema y no está implementada, la respuesta debe ser 502. Un caso particular de esto es la respuesta 504 para una orden que está implementada pero que se solicita con un argumento no implementado.

:::

REFERENCIAS

[1] Feinler, Elizabeth, *Libro de trabajo sobre la transición de protocolo en Internet,* Network Information Center, SRI International, marzo de 1982.

[2] Postel, Jon, *Protocolo de control de la transmisión - Especificación del protocolo del programa DARPA Internet*, RFC 793, DARPA, septiembre de 1981.

[3] Postel, Jon, and Joyce Reynolds, *Especificación del Protocolo Telnet*, RFC 854, ISI, mayo de 1983.

[4] Reynolds, Joyce, and Jon Postel, *Números asignados*, RFC 943, ISI, abril de 1985.

Ir al contenido

Índice

Primer bimestre

Segundo bimestre

Solucionario

Referencias bibliográficas

Anexos

Recursos

Anexo 3. RFC 1869_EXTENSIONES DEL SERVICIO SMTP

Network Working Group	J. Klensin, presidente del grupo de trabajo
Request for Comments:	1869 MCI
STD: 10	N. Freed, Editor
Obsoleta: 1651	Innosoft International, Inc.
Categoría:	Seguimiento de Estándar M. Rose Dover Beach Consulting, Inc.
	E. Stefferud
	Network Management Associates, Inc.
	D. Crocker
	Brandenburg Consulting noviembre 1995

Traducción: Rubén Afonso Francos <rujofa2@terra.es>

Extensiones del Servicio SMTP

3. Estructura para las extensiones SMTP

Para las extensiones de los servicios SMTP, el mensaje SMTP es una unidad de correo formada por envoltura y contenido.

(1) La envoltura SMTP es clara y se transmite como una serie de unidades del protocolo SMTP. está formada por una dirección de origen (a donde se deberían remitir los errores); un modo de entrega (por ejemplo, entregar a las cuentas de correo del receptor); y una o más direcciones de recepción.

(2) El contenido SMTP se envía dentro de la unidad de DATOS del protocolo SMTP, y consta de dos partes: las cabeceras y el cuerpo. Las cabeceras forman un conjunto de pares campo/valores estructurados de acuerdo con el RFC 822 [2], mientras que el cuerpo, si se encuentra estructurado, está definido de acuerdo con MIME [3]. El contenido es textual por naturaleza, representado utilizando el repertorio US ASCII (ANSI X3.4-1986). A pesar de que las extensiones (como MIME) pueden suavizar esta restricción referente al cuerpo, las cabeceras siempre se codifican utilizando el repertorio US ASCII. El algoritmo definido en [4] se utiliza para representar valores de la cabecera que no están en el repertorio US ASCII, mientras se sigue usando dicho repertorio para codificarlos.

A pesar de que SMTP se encuentra amplia y sólidamente extendido, algunos sectores de la comunidad Internet desean ampliar los servicios que proporciona. Este memorándum define un método que permite que un cliente y un servidor extendidos puedan reconocerse como tal y en el que un servidor pueda informar al cliente de las extensiones que soporta.

Debe resaltarse que ninguna extensión del servicio SMTP debe ser considerada a la ligera. La robustez de SMTP proviene principalmente de su sencillez. Las experiencias con otros protocolos han demostrado que: los protocolos con pocas opciones tienden a la ubicuidad, mientras que los que tienen muchas opciones tienden a la oscuridad.

Esto significa que cada extensión, sin importar las ventajas que aporte, debe ser cuidadosamente analizada a lo que su implementación, desarrollo, y costes de interoperabilidad se refiere. En muchos casos, el coste derivado de la extensión del servicio SMTP superará a los beneficios.

Dada esta situación, las extensiones descritas en este memorándum consisten en:

(1) un nuevo comando SMTP (sección 4),

(2) un registro de las extensiones del servicio SMTP (sección 5),

(3) parámetros adicionales para los comandos MAIL FROM y RCPT TO del protocolo SMTP (sección 6).

4. El comando EHLO

Un cliente SMTP que soporte las extensiones del servicio SMTP debería comenzar una sesión SMTP enviando el comando EHLO en lugar del comando HELO. Si el servidor SMTP también las soporta devolverá una respuesta satisfactoria (ver sección 4.3), una respuesta de fallo (ver sección 4.4), o bien una respuesta de error (4.5). Si el servidor SMTP no soporta ninguna extensión del servicio SMTP generará una respuesta de error (ver sección 4.5).

1.3. Modificaciones al STD 10, RFC 821

Esta especificación está destinada a extender el STD 10, RFC 821, sin influir de ninguna manera en los servicios ya existentes. Los cambios menores requeridos se enumeran debajo.

4.a.1. Primer comando

El RFC 821 establece que el primer comando en una sesión SMTP debe ser HELO. Por la presente se modifica este requisito de forma que se permite iniciar una sesión con HELO o con EHLO.

4.a.2. Longitud máxima de la línea de comandos

Esta especificación amplía los comandos MAIL FROM y RCPT TO del protocolo SMTP para permitir parámetros y valores de parámetro adicionales. Es posible que las líneas MAIL FROM y RCPTTO resultantes excedan el límite de 512 caracteres impuesto por el RFC 821. Por la presente se modifica dicho límite para que se aplique únicamente a las líneas de comandos que no contengan ningún parámetro. Las especificaciones que definan nuevos parámetros para MAIL FROM o RCPT TO deben además especificar la longitud máxima de los mismos, para que los implementadores de las extensiones conozcan el tamaño del bloque de memoria que deben reservar. La longitud máxima de comando que una implementación SMTP con extensiones debe soportar es 512, más la suma de la longitud máxima de todos los parámetros de cada una de las extensiones soportadas.

1.4. Sintaxis del comando

La sintaxis de este comando, usando la notación ABNF de [2], es:

ehlo-cmd ::= “EHLO” SP dominio CR LF

Si hay éxito, el servidor SMTP responde con el código 250. En caso de fallo, el servidor responde con el código 550. Si hay un error, responde con el código 500, 501, 502, o 421.

Este comando es enviado en lugar del comando HELO, y puede ser transmitido en cualquier momento en el que fuese apropiado enviar el comando HELO. Es decir, si se envía el comando EHLO, y se devuelve una respuesta de éxito, otro comando HELO o EHLO posterior hará que el servidor SMTP responda con el código 503. El cliente SMTP no debe almacenar ninguna información que se haya devuelto si el comando EHLO se responde con éxito. Esto es, el

cliente SMTP debe enviar el comando EHLO al inicio de cada sesión SMTP si se necesita información sobre los recursos extendidos.

1.5. Respuesta satisfactoria

Si el servidor SMTP implementa y es capaz de utilizar el comando EHLO, devolverá el código 250. Esto indica que tanto el cliente como el servidor SMTP se encuentran en el estado inicial, es decir, no existe transacción en curso y todas las tablas de estado y buffers se encuentran vacías.

Normalmente, esta respuesta ocupará varias líneas. Cada línea de la respuesta contendrá un comando y, opcionalmente, uno o más parámetros. La sintaxis para una respuesta positiva, utilizando la notación ABNF de [2], es:

respuesta-ehlo-ok ::= “250” dominio [SP saludo] CR LF

/ (“250-” dominio [SP saludo] CR LF

*(“250-” linea-ehlo CR LF)

“250” SP linea-ehlo CR LF)

; la conversación EHLO usual

saludo ::= 1*<cualquier carácter distinto de CR o LF>

linea-ehlo ::= clave-ehlo *(SP parámetro-ehlo)

clave-ehlo ::= (LETRA / DÍGITO) *(LETRA / DÍGITO / “-”)

; la sintaxis y los valores dependen de la

; clave-ehlo

parámetro-ehlo	::= 1*<cualquier CARACTER excluyendo SP y todos los caracteres de control (US ASCII 0-31 inclusive)>
LETRA	::= <cualquiera de los 52 caracteres alfabéticos (A hasta Z en mayúsculas, y, a hasta z en minúsculas)>
DÍGITO	::= <cualquiera de los 10 caracteres numéricos (0 hasta 9)>
CR	::= <el carácter de retorno de carro (código ASCII decimal 13)>
LF	::= <el carácter de nueva línea (código ASCII decimal 10)>
SP	::= <el carácter de espacio (código ASCII decimal 32)>

Aunque los comandos EHLO estén escritos en mayúsculas, minúsculas, o una mezcla de ambas, siempre deben ser reconocidos y procesados de forma insensible a las mayúsculas o minúsculas. Esto solo es una extensión de las prácticas empezadas en el RFC 821.

La IANA mantiene un registro de las extensiones del servicio SMTP. Asociado con cada extensión existe un valor clave EHLO. Cada extensión de servicio registrada por la IANA debe estar definida en un RFC. Dichos RFC deben ser un seguimiento de estándar o definir un protocolo experimental aprobado por el IESG. La definición debe incluir:

(1) el nombre textual de la extensión del servicio SMTP;

(2) el valor de la clave EHLO asociada con la extensión;

(3) la sintaxis y los posibles valores de los parámetros asociados con el valor de la clave EHLO.

(4) cualquier palabra SMTP adicional asociada con la extensión (las palabras adicionales generalmente, aunque no necesariamente, coinciden con el valor de la clave EHLO);

(5) cualquier parámetro nuevo que la extensión asocie con las claves MAIL FROM o RCPT TO;

(6) la forma en que el hecho de soportar la extensión afecta al comportamiento del cliente y del servidor SMTP; y,

(7) el incremento que produce la extensión en la longitud máxima de los comandos MAIL FROM, RCPT TO, o ambos, por encima de lo especificado en el RFC 821.

Adicionalmente, cualquier valor de la clave EHLO que comience con "X", ya sea mayúscula o minúscula, hará referencia a una extensión SMTP local, la cual será utilizada a través de un acuerdo bilateral en lugar de uno estandarizado. No se utilizarán claves que comiencen con "X" en una extensión de servicio registrada.

Cualquier valor de clave presentado en la respuesta EHLO que no comience con "X" debe corresponder con un estándar, seguimiento de estándar, o extensión de servicio SMTP experimental aprobada por el IESG y registrada por la IANA. Un servidor conforme con esto no debe ofrecer claves que no comiencen por "X" que no estén descritas en una extensión registrada.

Las claves adicionales están sometidas a las mismas normas que las claves EHLO; en concreto, las palabras que comiencen con "X" son extensiones locales que pueden no estar registradas o estandarizadas y las palabras que no comiencen con "X" siempre deben estar registradas.

1.6. Respuesta de fallo

Si por alguna razón el servidor SMTP es incapaz de mostrar las extensiones de servicio que soporta, devolverá el código 554.

En el caso de una respuesta de fallo, el cliente SMTP debería enviar el comando HELO o QUIT.

1.7. Respuestas de error generadas por servidores extendidos

Si el servidor SMTP reconoce el comando EHLO, pero no acepta los argumentos que lo acompañan, devolverá el código 501.

Si el servidor SMTP lo reconoce, pero no lo implementa, devolverá el código 502.

Si el servidor SMTP determina que no es posible seguir proporcionando el servicio SMTP (por ejemplo, debido a un apagado inmediato), devolverá el código 421.

El cliente SMTP debería enviar el comando HELO o QUIT en el caso de que reciba cualquier respuesta de error.

1.8. Respuestas de servidores sin extensiones

Un servidor SMTP que es conforme con el RFC 821, pero no soporta las extensiones especificadas aquí, no reconocerá el comando EHLO y consecuentemente devolverá el código 500, tal y como se especifica en el RFC 821. El servidor SMTP debería permanecer en el mismo estado después de devolver este código (ver sección 4.1.1 del RFC 821). El cliente SMTP puede entonces enviar el comando HELO o QUIT.

1.9. Respuestas de servidores mal implementados

Se sabe que algunos servidores SMTP cortan la transmisión SMTP al recibir el comando EHLO. La desconexión puede ocurrir inmediatamente o después de devolver una respuesta. Este comportamiento infringe la sección 4.1.1 del RFC 821, que establece claramente que la desconexión solo debe producirse después de que se envíe el comando QUIT.

Para conseguir una mejor interoperabilidad, en ningún caso se sugiere que los clientes SMTP extendidos utilizando EHLO sean implementados para comprobar si el servidor ha desconectado después de enviar el comando EHLO, ya sea antes o después de haber devuelto una respuesta. Si esto ocurre el cliente debe decidir si la operación puede ser realizada sin utilizar ninguna extensión SMTP. Si es así, se puede abrir una nueva conexión y utilizar el comando HELO.

Otros servidores mal implementados no aceptarán un comando HELO después de que haya sido enviado el comando EHLO y se haya rechazado.

En algunos casos, este problema puede ser evitado enviando un RSET después de devolver la respuesta fallida al EHLO, y enviar posteriormente HELO. Los clientes que hagan esto deberían ser advertidos de que muchas implementaciones devolverán un código de fallo (por ejemplo, 503 secuencia errónea de comandos) en respuesta al RSET. Este código puede ser ignorado con tranquilidad.

::::

8. Ejemplos de uso

(1) Una interacción del estilo de:

S: <espera una conexión TCP en el puerto 25>

C: <establece una conexión con el servidor>

S: 220 dbc.mtview.ca.us servicio SMTP preparado

C: EHLO ymir.claremont.edu

S: 250 dbc.mtview.ca.us saluda

...

Indica que el servidor SMTP implementa únicamente aquellos comandos SMTP que están definidos como obligatorios en [5].

(2) En cambio, una interacción de la forma:

S: <espera una conexión TCP en el puerto 25>

C: <establece una conexión con el servidor>

S: 220-dbc.mtview.ca.us servicio SMTP preparado

C: EHLO ymir.claremount.edu

S: 250-dbc.mtview.ca.us saluda

S: 250-EXPN

S: 250-HELP

S: 250-8BITMIME

S: 250-XONE S: 250-XVRB

...

Indica que el servidor SMTP además implementa los comandos EXPN y HELP, una extensión de servicio estándar (8BITMIME), y dos extensiones de servicio no estándares sin registrar (XONE y XVRB).

(3) Finalmente, un servidor que no soporte extensiones del servicio SMTP, debería actuar de la siguiente manera:

S: <espera una conexión TCP en el puerto 25>

C: <establece una conexión con el servidor>

S: 220 dbc.mtview.ca.us servicio SMTP preparado

C: EHLO ymir.claremont.edu

S: 500 Comando no reconocido: EHLO

...

La respuesta 500 indica que el servido SMTP no implementa las extensiones aquí especificadas. Normalmente, el cliente debería entonces enviar un comando HELO y continuar como se especifica en el RFC 821(ver la sección 4.7 para un análisis adicional).

11. Referencias

[1] Postel, J., *Protocolo Simple de Transferencia de Correo*, STD 10, RFC 821, USC/Information Sciences Institute, agosto 1982.

[2] Crocker, D., *Estándar para el formato de los mensajes de texto de la Internet* ARPA, STD 11, RFC 822, UDEL, agosto 1982.

[3] Borestein, N., y N. Freed, *Extensiones Multipropósito del Correo de Internet*, RFC 1521, Bellcore, Innosoft, septiembre 1993.

[4] Moore, K., *Representación del Texto No-ASCII en las cabeceras de los Mensajes de Internet, RFC 1522*, Universidad de Tennessee, septiembre 1993.

[5] Braden, R., *Requisitos de los Anfitriones de Internet - Aplicación y Apoyo, STD 3, RFC 1123*, USC/Instituto de Ciencias de la Información, octubre 1989.

Ir al contenido

Índice

Primer bimestre

Segundo bimestre

Solucionario

Referencias bibliográficas

Anexos

Recursos

7. Recursos

Informaciones Importantes

Instale en su PC el programa Wireshark. Verifique la versión de su plataforma para la instalación. Wireshark es un *software* que permite capturar tráfico de red para analizar protocolos, paquetes y tramas que circulan por una red. Siga las instrucciones de la página de descarga para la instalación de Wireshark.

Ingrese al portal de curso de Packet Tracer de NetAcad de Cisco®, que es un curso gratuito sobre el manejo de una herramienta llamada Packet Tracer, que permite la simulación de redes de datos, una vez inscrito podrá descargar e instalar esta herramienta. Se debe seguir el curso y aprobarlo para que adquiera la habilidad necesaria para elaborar las tareas.

También se recomienda seguir el curso Introduction to Cybersecurity de NetAcad de Cisco®,

Le invitamos a revisar el vídeo de Capa de Aplicación del modelo TCPIP | Aprende Redes desde CERO, del canal de YouTube Master IT, donde podrá recordar los conceptos básicos sobre el modelo OSI y generalidades sobre las redes de datos en la capa de aplicación.

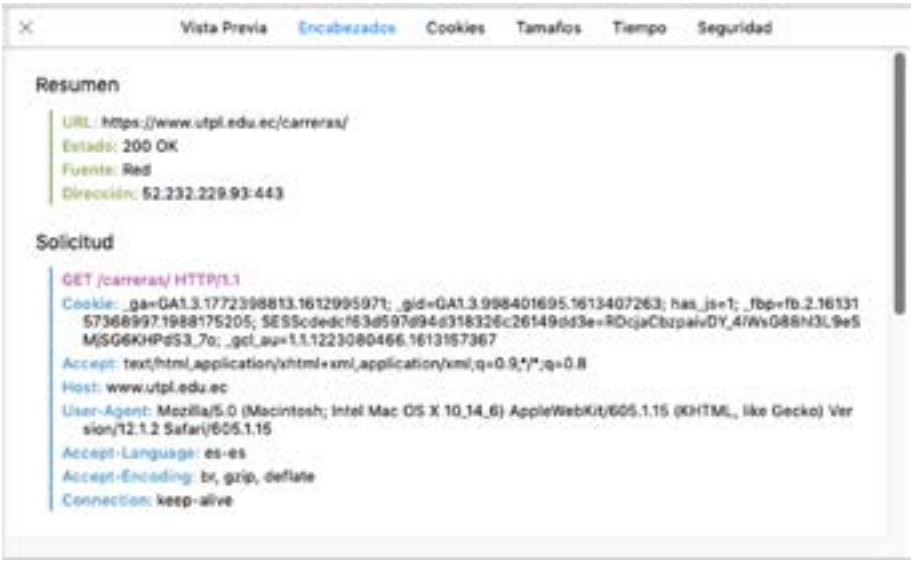
Ir al contenido

Solicitudes y respuestas HTTP

2.1.4.1. Solicitudes HTTP

En la Figura 10, se visualiza el detalle de una cabecera de solicitud típico de HTTP, en el cual se puede ver por ejemplo el tipo de agente de usuario utilizado en la solicitud.

Figura 10.
Cabecera de solicitud típico HTTP



Existen métodos que se utilizan en las solicitudes con HTTP 1.0:

- GET: solicita un recurso u objeto.
- HEAD: es similar a GET, la diferencia es que con HEAD solo se solicitan las cabeceras del recurso. Con este método es fácil conocer las características del recurso u objeto sin tener que acceder a él para su descarga.

- POST: se utiliza para enviar datos a un servidor. Los datos se transfieren a través del cuerpo de la solicitud.

Los métodos con HTTP 1.1 son:

- GET: solicita un recurso u objeto.
- HEAD: es similar a GET, la diferencia es que con HEAD solo se solicitan las cabeceras del recurso. Con este método es fácil conocer las características del recurso u objeto sin tener que acceder a él para su descarga.
- POST: se utiliza para enviar datos a un servidor. Los datos se transfieren a través del cuerpo de la solicitud.
- PUT: se utiliza para almacenar el fichero del cuerpo del mensaje en la ruta especificada en el URL.
- DELETE: borra un fichero indicado con el URL.

El formato inicial de la solicitud es:

- Método <URL> versión

Ejemplo: GET www.utpl.edu.ec HTTP/1.0

A continuación, se define el formato de una cabecera:

Nombre-Cabecera: valor <CR><LF>

HTTP 1.0 define 16 cabeceras ninguna de ellas es obligatoria, mientras que HTTP 1.1. define 46 cabeceras de las cuales solo Host es obligatoria.

Las cabeceras pueden ser diferentes y esto depende del tipo de agentes de usuario:

- User-Agent: el navegador o agente de usuario que usa el cliente.
- Accept: tipo de contenidos que son aceptados (solo para HTTP 1.1).

Le invito a complementar sus conocimientos con el apartado de las respuestas HTTP.

2.1.4.2. Respuestas HTTP

En la Figura 11, se muestra el detalle de un mensaje de respuesta típico de HTTP. A continuación, describiremos su estructura.

Los mensajes de respuesta inician con la línea de estado:

Protocolo <código de estado> frase de estado

Por ejemplo: HTTP/1.1 200 OK

Figura 11.
Cabecera de respuesta típico HTTP

Tabla 4.
Códigos de respuestas HTTP

Código	Descripción
1xx	Informativo. La solicitud se recibe y sigue en proceso
2xx	Éxito. La acción solicitada ha sido recibida, entendida y aceptada
3xx	Redirección. Informa que se deben realizar acciones adicionales
4xx	Error del cliente, sobre todo en errores de sintaxis
5xx	Error del servidor

En la Tabla *4*, se muestran los códigos de estado de respuestas HTTP, entre los principales tenemos:

- 200 OK: la solicitud se ha ejecutado con éxito.
- 301 moved Permanently: el objeto solicitado está en una nueva posición especificada en este mensaje (Location:).
- 400 Bad Request: la petición no fue entendida por el servidor.
- 404 Not Found: el documento solicitado no se encuentra en el servidor.
- 505 HTTP Version Not Supported. La versión de protocol HTTP no es soportado por el servidor.

Las respuestas HTTP también utilizan cabeceras, entre las más importantes son:

- Content-Type: el tipo MIME de los datos en el cuerpo.
- Content-Length: el tamaño en bytes del cuerpo.

Ir al contenido

SMTP

3.1.2.1. Mensajes SMTP

Antes de abordar el presente tema, invitamos a leer detenidamente la sección 2.4.3 del texto básico.

Los mensajes SMTP permiten realizar la transferencia de correo a través de modelo solicitud/respuesta. En el SMTP actual todos los mensajes se estructuran en formato ASCII de 7 *bits* y están delimitados por el carácter <CR><LF>. Basta con disponer de 8 comandos para realizar un intercambio de correo, y son: HELO, MAIL, RCPT, DATA y QUIT.

En la Tabla 11 se encuentran algunos comandos básicos utilizados.

Tabla 11.
Comandos SMTP

Comando	Descripción
HELO <nombre de dominio> <CR><LF>	Especifica el origen de la conexión.
MAIL FROM: <dirección origen> <CR><LF>	Identifica al remitente.
RCPT TO: <dirección destino> <CR><LF>	Identifica al destinatario.
DATA <CR><LF>	Introducción de datos.
RSET <CR><LF>	Aborta la conexión con el servidor SMTP.
QUIT <CR><LF>	Cierra la conexión con servidor SMTP.
HELP <CR><LF>	Presenta ayuda sobre órdenes.
EXPN <dirección de correo> <CR><LF>	Lista el correo.
VRFY <dirección de correo> <CR><LF>	Comprueba la existencia de una dirección de correo.

Las respuestas SMTP del servidor por lo general se estructuran en líneas que comienzan con un código de tres dígitos, seguidos por información descriptiva sobre el resultado.

Tabla 12.
Respuestas SMTP

Código	Descripción
2xx	Respuesta satisfactoria
3xx	Respuesta temporal afirmativa
4xx	Respuesta de error pasajera
5xx	Respuesta de error permanente

En la Tabla 12 se encuentran las respuestas SMTP de acuerdo a un código agrupado, para luego detallarse considerando las posibles respuestas. Algunas respuestas frecuentes se muestran en la Tabla 13.

Tabla 13.
Ejemplos de Respuesta SMTP

220	Servidor listo
211	Estado del sistema
421	Servidor no disponible
250	OK
500	Error de sintaxis
503	Secuencia errónea de comandos
550	Usuario no encontrado
554	Transacción fallida
354	Empezar entrada de correo

3.1.2.2. Funcionamiento simplificado de SMTP

Como mencionamos anteriormente se debe establecer una conexión entre cliente y servidor antes de intercambiar un correo. Para ello es necesario que:

1. El servidor envíe un mensaje confirmando o negando su disponibilidad 220 y 421, respectivamente.

2. Luego el cliente envía el comando HELO con su identificación.

3. Con la orden MAIL el cliente notifica al servidor que quiere enviar un correo e incluso se puede enviar una dirección de notificaciones. A lo cual el servidor envía una confirmación de ello a través del comando 250.

4. Se debe especificar el destino del mensaje, para ello se dispone de la orden RCPT TO: destino. El servidor contestará 250 si dispone de ese receptor, de lo contrario informará con el comando 550.

5. A continuación, el cliente informa que desea enviar datos mediante el comando DATA. Si el servidor está listo para recibir el *mail* responde 354, informándole además al cliente que debe notificar el fin del envío de datos cuando termine.

6. El cliente envía todos los datos del cuerpo del mensaje, y cuando termina envía una línea <CL><RF> seguido de una línea que solo contendrá un punto para identificar que el mensaje finaliza. Si se reciben todos los datos del correo el servidor envía una confirmación de ello a través del comando 250.

7. Se cierra lo conexión con la orden QUIT.

3.1.2.3. SMTP Extendido

El ESMTP (Extended SMTP) es una extensión del protocolo SMTP definido en RFC 1425. La principal diferencia con el SMTP es que se agregan dos comandos:

- EHLO <nombre de dominio> <CR><LF>: sirve para que el servidor realice una consulta al DNS sobre el dominio destino del correo, de tal forma de verificar que este exista.
- ETRN <nombre de dominio> <CR><LF>: con este comando el cliente le solicita al servidor se le envíe todos los mensajes que van dirigidos a él.

En el anexo 3 puede consultar más detalles al respecto de las modificaciones funcionales de las extensiones SMTP en el RFC 1869.

Ir al contenido

POP

3.1.3.1. Funcionamiento simplificado de POP3

El protocolo POP3 recupera los mensajes de correo en base a sesiones. Estas sesiones constan de:

1. *Autorización*: cuando el cliente POP3 establece la conexión TCP, el servidor lanza una línea de saludo, es entonces cuando se encuentra en fase de autorización, donde el servidor identifica al usuario. Esto se puede realizar en base a las órdenes USER y PASS o a la orden APOP.

2. *Transferencia*: es cuando el usuario ya ha sido identificado y ha podido acceder a su cuenta, entonces el usuario manipula la información contenida en su buzón de correo.

3. *Actualización*: luego de que el cliente ha realizado todas sus transacciones, este debe enviar la orden QUIT para entrar en la fase de actualización, puesto que todas las modificaciones se realizan cuando el cliente finaliza el servicio.

3.1.3.2. Comandos usados por POP3

Los comandos usados por POP3 son similares a los de SMTP, en la Tabla 14 se encuentran los comandos usados por el cliente POP3. Sin embargo, existen algunas órdenes opcionales, por ejemplo: TOP

<mensajes líneas> <CR> <LF> que permite listar la cabecera de los mensajes más las líneas del cuerpo que se indican (ver Tabla 15).

Tabla 14.
Comandos POP3

Comando	Descripción
USER<usuario> <CR><LF>	Sirve para identificar al usuario
PASS <password> <CR><LF>	Sirve para identificar la contraseña
STAT <CR><LF>	Devuelve el número de mensajes y tamaño en *bytes*
LIST <CR><LF>	Lista los mensajes en buzón
RETR <mensaje> <CR><LF>	Devuelve un mensaje específico
DELE <mensaje> <CR><LF>	Marca un mensaje para borrar
QUIT <CR><LF>	Termina y ejecuta todas las modificaciones
RSET <CR><LF>	Abandona la sesión sin borrar mensajes

El servidor responderá a cada uno de los comandos con dos tipos de mensajes: "+OK" o "-ERR", más un espacio en blanco y a continuación la descripción del resultado.

Tabla 15.
Comandos opcionales POP3

Comando	Descripción
APOP <usuario> resu <CR><LF>	Permite autenticación
TOP <mensaje> n <CR><LF>	Se envía cabecera más n líneas del mensaje (solo en transacción)
UIDL <mensaje> <CR><LF>	Devuelve el identificador de mensaje (solo en transacción)

Ir al contenido