INTRODUCCIÓN

En el mundo, las telecomunicaciones hacen parte importante de la vida cotidiana del hombre, y los medios de transmisión son una parte esencial pero como todo componente de un sistema de computación, solos no sirven mucho.

Aquí entra a formar parte distintos software y modelos de creación de redes quienes dan la configuración a los desarrollos que podemos hacer con las diferentes herramientas y dispositivos que encontramos en el mercado actual y que nos ofrecen una amplia gama de posibilidades.

El conocimiento oportuno y completo de los protocolos nos dan la ventaja de saber exactamente que podemos hacer con lo que tenemos y que es lo mejor que podemos sacar de ellos.

1. PROTOCOLOS DE COMUNICACIÓN.

Los protocolos son los que definen un conjunto de reglas para intercambiar información y cooperar. Son ellos, por ejemplo, los que definen la manera como compartimos información a través del Internet o incluso la manera como chateamos.

En la actualidad contamos con muchos protocolos de comunicación comerciales, incluso algunas empresas de telecomunicaciones tales como la AT&T han llegado a desarrollar sus propios protocolos, dependiendo de los servicios que ofrezcan a sus usuarios. Estos protocolos muchas veces aun sin darnos cuenta son usados por nosotros y nos ayudan a hacer tareas como los son el Internet, una transferencia por módem o una simple comunicación a un servicio en línea inteligente de algún banco.

Los protocolos que a continuación se relacionan son los más importantes y comerciales que existen hoy día, aunque la cantidad que se ha desarrollado es muy amplia pero también difícil de condensar en un solo trabajo y bastante complicado es ubicar, igualmente, información sobre ellas.

Los protocolos a tratar son:

- ftp.
- http.
- Ipx/spx.
- Nfs.
- pop3.
- Scp.
- Tcp/ip.

Cada uno esta hecho para diferentes tipos de tareas.

1.1. FTP.

El protocolo FTP o file transfer protocol (protocolo de transferencia de archivos) tiene como objetivo principal varios puntos, como son, promover el compartir archivos entre computadoras (programas y datos), alentar el uso remoto de las computadoras, y transferir datos de una forma segura y optima por computadora. FTP más que para ser usado por un usuario directamente es para que los programas lo usen entre ellos para comunicarse.

Con este tipo de forma de hacer las cosas le ayudamos al usuario para que no tenga que preocuparse por el tipo de computadora con la cual tiene contacto, sean microcomputadoras, micro, mini o simples computadores personales. Gracias a este tipo de protocolo no se necesita saber mucho y se pueden lograr muchas cosas.

El protocolo ha ido evolucionando demasiado en todos estos años desde que se creo, este empezó en 1.971 con un modelo de transferencia llamado RFC 141 en MIT. Fue hasta después de muchas revisiones que llegó a RFC 265 cuando ya se le considero como un protocolo de transferencia de archivos completo entre HOSTs (o servidores de archivos) de ARPHANET. Finalmente un documento declarando un FTP oficial se publicó cuando se llego a RFC 454.

El FTP cambio mucho pero al final de la edición de RFC 765 se incluyó alguno de los que son ahora los comandos de este protocolo:

- CDUP (change to parent directory).
- SMNT (structure mount).
- STOU (store unique).
- RMD (remove directory).
- MKD (make directory).
- PWD (print directory).
- SYST (system).

Existen tres tipos de datos en la transferencia por FTP, el tipo ASCII, EBCDIC, IMAGEN.

El tipo ASCII es el más común, se usa cuando se transfieren archivos de texto en el cual el SENDER debe convertir cualquiera que sea su estructura de archivos interna al formato genérico de 8 bits, y el RECEIVER a su propio formato.

El EBCDIC es el más eficiente cuando ambos equipos lo usan como formato propio, se representa también en 8 bits pero de forma EBCDIC, la diferencia se da en la forma de reconocer los códigos de los caracteres.

IMAGEN es cuando se empaca todo lo que se quiere enviar en cadenas seguidas de paquetes de 8 bits, esto es no importa el formato en que internamente se maneje la información, cuando se va enviar se tiene que hacer una conversión de 8 en 8 bits y cuando el que recibe tiene todo el paquete, el mismo debe codificarlos de nuevo para que la transmisión sea completada.

En FTP se consideran tres tipos deferentes de archivos. Estos son FILE−STRUCTURE (donde no hay estructuras internas y el archivos es considerado una secuencia continua de bytes), RECORD−STRUCTURE (donde los archivos contienen puros registros iguales en estructura) y PAGE−STRUCTURE (donde los archivos contienen paginas enteras indexadas separadas).

Al establecer una conexión por FTP se debe tomar en cuenta que el mecanismo de transferencia consiste en colocar bien la transferencia de datos en los puertos adecuados y al concluir la conexión estos puertos deben ser cerrados adecuadamente. El tamaño de transferencia es de 8 bits, en ambos. El que va a transferir, debe escuchar desde el puerto hasta que el comando enviado sea recibido y este será el que de la dirección de la transferencia. Una vez recibido el comando y establecido una transferencia del servidor a que solicita se inicializa la comunicación de la transferencia para verificar la conexión, esta es una cabecera con un formato específico, después de esto se comienza a enviar las tramas de 8 bits sin importar el tipo de datos que sea (antes mencionado), y al finalizar se envía otra trama cabecera ya establecida confirmando la transferencia completada.

Existen tres modos de transferencia en FTP como son el STREAM MODE, BLOCK MODE y COMPRESSED MODE.

Algunos de los comandos mas usados en FTP son los siguientes:

Comandos de acceso

USER NAME (USER)

PASSWORD (PASS

ACCOUNT (ACCT)

CHANGE WORKING DIRECTORY (CWD)

CHANGE TO PARENT DIRECTORY (CDUP)

REINITIALIZE (REIN)

LOGOUT (QUIT)

Comandos de transferencia

DATA PORT (PORT)

PASSIVE (PASV)

FILE STRUCTURE (STRU)

TRANSFER MODE (MODE)

Comandos de servicio

RETRIEVE (RETR)

STORE (STOR)

STORE UNIQUE (STOU)

APPEND (with create) (APPE)

ALLOCATE (ALLO)

RENAME TO (RNTO)

ABORT (ABOR)

DELETE (DELE)

REMOVE DIRECTORY (RMD)

MAKE DIRECTORY (MKD)

PRINT WORKING DIRECTORY (PWD)

LIST (LIST)

HELP (HELP)

Algunos de los códigos usados en la transferencia son los siguientes, estos códigos no son más que mensajes enviados por el protocolo:

Códigos normales

- 200 Command okay.
- 500 Syntax error, command unrecognized. This may include errors such as command line too long.
- 501 Syntax errors in parameters or arguments.
- 202 Command not implemented, superfluous at this site.
- 502 Command not implemented.
- 503 Bad sequence of commands.
- 504 Command not implemented for that parameter.
- 110 Restart marker reply. In this case, the text is exact and not left to the particular implementation; it must read:
- 211 System status or systems help reply.
- 212 Directory status.
- 213 File status.
- 214 Help message. On how to use the server or the meaning of a particular non−standard command. This reply is useful only to the human user.
- 215 NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.
- 120 Service ready in nnn minutes.
- 220 Service ready for new user.
- 221 Service closing control connection. Logged out if appropriate.
- 421 Service not available, closing control connection. This may be a reply to any command if the service knows it must shut down.
- 125 Data connection already open; transfer starting.
- 225 Data connection open; no transfer in progress.
- 425 Can't open data connection.
- 226 Closing data connection. Requested file action successful (for example, file transfer or file abort).
- 426 Connection closed; transfer aborted.
- 227 Entering Passive Mode (h1, h2, h3, h4, p1, p2).
- 230 User logged in, proceed.
- 530 not logged in.
- 331 User name okay, need password.
- 332 Need account for login.
- 532 Need account for storing files.
- 150 File status okay; about to open data connection.
- 250 Requested file action okay, completed.
- 257 "PATHNAME" created.
- 350 Requested file action pending further information.
- 450 Requested file action not taken. File unavailable (e.g., file busy).
- 550 Requested action not taken. File unavailable (e.g., file not found, any access).
- 451 Requested action aborted. Local error in processing.
- 551 Requested action aborted. Page type unknown.
- 452 Requested action not taken. Insufficient storage space in system.
- 552 Requested file action aborted Exceeded storage allocation (for current directory or dataset).
- 553 Requested action not taken. File name not allowed.

Códigos de mensajes con operaciones numéricas

- 110 Restart marker reply.
- 120 Service ready in nnn minutes.
- 125 Data connection already opens; transfer starting.
- 150 File status okay; about to open data connection.
- 200 Command okay.
- 202 Command not implemented, superfluous at this site.
- 211 System status or system help reply.
- 212 Directory status.
- 213 File status.
- 214 Help message. On how to use the server or the meaning of a particular non−standard command. This reply is useful only to the human user.
- 215 NAME system type. Where NAME is an official system name from the list in the Assigned Numbers document.
- 220 Service ready for new user.
- 221 Service is closing control connection. Logged out if appropriate.
- 225 Data connection open; no transfer in progress.
- 226 Closing data connection. Requested file action successful (for example, files transfer or file abort).
- 227 Entering Passive Mode (h1, h2, h3, h4, p1, p2).
- 230 User logged in, proceed.
- 250 Requested file action okay, completed.
- 257 "PATHNAME" created.
- 331 User names okay need password.
- 332 Need account for login.
- 350 Requested file action pending further information.
- 421 Service not available, closing controls connection. This may be a reply to any command if he service knows it must shut down.
- 425 can't open data connection.
- 426 Connection closed; transfer aborted.
- 450 Requested file action not taken. File unavailable (e.g., file busy).
- 451 Requested action aborted: local error in processing.
- 452 Requested action not taken. Insufficient storage space in system.
- 500 Syntax error, command unrecognized. This may include errors such as command line too long.
- 501 Syntax error in parameters or arguments.
- 502 Command not implemented.
- 503 Bad sequence of commands.
- 504 Command not implemented for that parameter.
- 530 Not logged in.
- 532 Need account for storing files.
- 550 Requested action not taken. File unavailable (e.g., file not found, no access).
- 551 Requested action aborted: page type unknown.
- 552 Requested file action aborted. Exceeded storage allocation (for current directory or dataset).

553 Requested action not taken. File name not allowed.

1.2. HTTP.

El protocolo HYPER TEXT TRANSFER PROTOCOL (protocolo para la transferencia de hipertextos) es para todos los sistemas de información distribuidos que tengas la necesidad de mostrar la información y pasarla por una comunicación normal haciendo uso de las ligas de este lenguaje. La primera versión de este lenguaje (http

0.9) se uso desde 1.990.

El protocolo fue implementado inicialmente para WWW en 1.991 como una iniciativa de software y se denominó http 0.9. El protocolo completo fue definido en 1.992 e implementado en marzo de 1.993.

- HTTP 1.0. esta especificación prevé las características básicas del protocolo.
- HTTP 1.1. la primera versión no está aun habilitada, pero las especificaciones son muy similares a la anterior.
- HTTP−NG next generation of HTTP, es un protocolo binario con nuevas características para un acceso más rápido usando TCP. Este es el último HTTP en la actualidad, es más complejo que un 0.9.

El protocolo encierra cierta terminología como:

- Conexión. Es el circuito virtual establecido entre dos programas en una red de comunicación con el proceso de una simple comunicación.
- Mensaje. Esta es la unidad básica, estos consisten en una secuencia estructurada que es transmitida siempre entre los programas.
- Servidor. El que presta el servicio en la red.
- Proxy. Un programa intermedio que actúa sobre los dos, el servidor y el cliente.

1.3. IPX/SPX

El internetwork packet exchange, sequence packet exchanged es un protocolo usado y registrado por la compañía mundial de redes NOVELL.

1.4. NFS.

El network file system (sistema de archivos de red) es un sistema distribuido para archivos, este es para las redes heterogéneas, con este protocolo, el usuario solo ve un directorio cuando esta dentro de la red, claro que tiene ramas dentro pero no puede ver más arriba de el nivel en el que se entra, tal ves los archivos dentro esta estructura del directorio ni siquiera esta en la misma computadora.

1.5. POP3.

El protocolo Post office protocol versión 3 es netamente un protocolo para la administración de correo en Internet. En algunos nodos menores de Internet normalmente es poco práctico mantener un sistema de transporte de mensajes (MTS). Por ejemplo, es posible que una estación de trabajo no tenga recursos suficientes (hdd, entre otros) para permitir que un servidor de SMTP y un sistema local asociado de entrega de correo estén residentes y continuamente en ejecución. De forma similar, puede ser caro mantener una computadora personal interconectada a una red tipo IP durante grandes cantidades de tiempo.

A pesar de esto, a menudo es muy útil poder administrar correo sobre estos nodos, y frecuentemente soportan un user agent (agente de usuario) para ayudar en las tareas de manejo de correo. Para resolver este problema, un nodo que sí sea capaz de soportar un MTS ofrecerá a estos nodos menos dotados un servicio MAILDROP (es el lugar en el sistema con el MTS donde el correo es almacenado para que los otros nodos puedan trabajar con él sin necesidad de mantener su propio MTS. El protocolo de oficina de correos está destinado a permitir que una estación de trabajo acceda dinámicamente a un MAILDROP en un HOST servidor de forma útil y eficiente. Esto significa que el protocolo POP3 se usa para permitir a una estación de trabajo recobrar correo que el servidor tiene almacenado.

POP3 no está destinado a proveer de extensas operaciones de manipulación de correo sobre el servidor; normalmente, el correo es transmitido y entonces borrado. IMAP4 es un protocolo más avanzado y complejo.

De aquí en adelante el término host cliente se refiere a un host haciendo uso del servicio POP3 y host servidor al que ofrece este servicio. Inicialmente, el host servidor comienza el servicio POP3 leyendo el puerto 110 TCP. Cuando un host cliente desea hacer uso del servicio, establece una conexión TCP con el host servidor. Cuando la conexión se establece, el servidor POP3 envía un saludo. Entonces, el cliente y el servidor POP3 intercambian comandos y respuestas respectivamente hasta que la conexión se cierra o es abortada.

Los comandos en el POP3 consisten en una palabra clave (keyword), posiblemente seguida de uno o más argumentos. Todos los comandos terminan con un par CRLF. Las palabras clave y los argumentos consisten en caracteres ASCII imprimibles. Las palabras clave son de una longitud de tres o cuatro caracteres, mientras que cada argumento puede ser de hasta 40 caracteres de longitud.

Las respuestas en el POP3 consisten de un indicador de estado y una palabra clave posiblemente seguida de información adicional. Todas las respuestas acaban en un par CLRF. Las respuestas pueden ser de hasta 512 caracteres de longitud, incluyendo el CRLF de terminación. También existen dos indicadores de estado, positivo o afirmativo (+OK) y negativo (−ERR). Los servidores deben enviarlos en mayúsculas.

Las respuestas a ciertos comandos son multilínea (una respuesta compuesta de varias líneas). En estos casos después de enviar la primera línea de la respuesta y un CRLF, se envía cualquier línea adicional, cada una termina en un par CRLF. Cuando todas las líneas de la respuesta han sido enviadas, se envía una línea final, que consiste en un octeto de terminación y un par CRLF. Si alguna línea de la respuesta multilínea comienza con el octeto de terminación, se ponen bites de relleno precedidos por el byte de terminación en esa línea de la respuesta. De aquí en adelante una respuesta multilínea termina con los cinco bytes CRLF.CRLF. Al examinar una respuesta multilínea, el cliente comprueba si la línea comienza con el byte de terminación. Si es así y si siguen otros bytes a excepción del CRLF, el primer byte de la línea o de terminación es ignorado. De este modo se el CRLF sigue inmediatamente al carácter de terminación, entonces la respuesta desde el servidor POP termina y la línea conteniendo CRLF no es considerada como parte de la respuesta multilínea.

Una sesión POP3 progresa a través de una serie de estados a lo largo de su vida. Una vez la conexión TCP ha sido abierta y el servidor de POP3 ha enviado el saludo, la sesión entra en el estado de autorización. En este estado, el cliente debe identificarse al servidor de POP3. Una vez el cliente lo ha hecho satisfactoriamente, el servidor adquiere los recursos asociados al maildrop del cliente, y la sesión entra en el estado de transacción. En este estado, el cliente realiza una serie de solicitudes al servidor de POP3. Cuando el cliente ha emitido el comando de finalización (QUIT) la sesión entra en el estado de actualización. En este estado, el servidor de POP3 libera cualquiera de los recursos adquiridos durante el estado de transición, se despide y la conexión TCP se cierra.

Un servidor debe responder a comandos no reconocidos, no implementados, o sintáctica mente incorrectos con un indicador negativo de estado (respuesta negativa). También debe responder con un indicador negativo de estado cuando la sesión se encuentra en un estado incorrecto. No hay un método general para que el cliente distinga entre un servidor que no implementa un comando opcional y un servidor que no esta dispuesto o es incapaz de procesar el comando.

Un servidor de POP3 puede disponer de un temporizador o cronómetro de inactividad (autologout inactivity timer). Tal cronómetro debe ser de por lo menos 10 minutos de duración. La recepción de cualquier comando desde el cliente durante este intervalo reinicia la cuenta de este cronómetro. Cuando el cronómetro llega a los diez minutos, la sesión no entra en el estado de actualización. Entonces, el servidor debería cerrar la conexión TCP sin eliminar ningún mensaje y sin enviar ninguna respuesta al cliente.

USER nombre

Argumentos: una cadena identificando un mailbox, el cual solo tiene significado para el servidor

Restricciones: solo puede darse en el estado de autorización después del saludo o de los comandos USER o PASS sin éxito.

Definición: Para autentificar usando la combinación de los comandos USER y PASS, el cliente debe primero emitir el comando USER. Si el servidor responde afirmativamente (+OK), entonces el cliente puede responder con el comando PASS para completar la autentificación, o el comando QUIT para finalizar con la conexión. Si el servidor responde negativamente (−ERR) al comando USER, el cliente puede emitir un nuevo comando de autenticación o bien el comando QUIT.

El servidor puede devolver una respuesta afirmativa incluso a pesar de que no exista ningún mailbox. El servidor puede devolver una respuesta negativa si el mailbox existe, pero no permitir la autenticación.

PASS cadena

Argumentos: palabra de acceso al mailbox

Restricciones: solo puede darse en el estado de autorización inmediatamente después de un comando USER satisfactorio.

Definición: Cuando el cliente el comando PASS, el servidor utiliza el par de argumentos de los comandos USER y PASS para determinar si al cliente se le debe dar acceso al maildrop apropiado.

Ya que el comando PASS tiene exactamente un argumento, un servidor de POP3 puede tratar los espacios como parte del password en lugar de cómo separadores de argumentos.

APOP nombre digest

Argumentos: una cadena identificando un mailbox y una cadena digest MD5

Restricciones: solo puede darse en el estado de autorización después del saludo o de los comandos USER o PASS sin éxito.

Definición: Normalmente, cada sesión POP3 comienza con intercambio USER/PASS. Esto tiene como resultado una clave de acceso específica enviada a través de la red. Para un uso intermitente del POP3, no conlleva un riesgo considerable. Sin embargo, muchas implementaciones de cliente POP3 conectan al servidor regularmente para comprobar si hay correo nuevo. Además, el intervalo de iniciación de la sesión puede ser del orden de 5 minutos. Por lo tanto, el riesgo de que la clave de acceso sea capturada es alto.

Se requiere un método alternativo de autenticación que no implique el envío de claves de acceso a través de la red. Esta funcionalidad la proporciona el comando APOP.

Un servidor que implemente el comando APOP incluirá una marca de tiempo (timestamp) en sus "saludos". La sintaxis de la marca de tiempo corresponde al "msg−id" en la RFC 882 (actualizada por RFC 973 y después por RFC 1982), y debe ser diferente cada vez que el servidor envía un saludo. Por ejemplo, en una implementación UNIX en la cual un proceso UNIX separado es el encargado de cada instancia de servidor, la sintaxis de la marca de tiempo podría ser: process−ID.clock@hostname, donde process ID es el valor decimal

del PID del proceso, clock es el valor decimal del reloj del sistema, y hostname es el nombre de dominio del host donde el servidor está funcionando.

El cliente recibe esta marca de tiempo y emite un comando APOP. El parámetro nombre tiene el mismo significado que el parámetro nombre del comando USER. EL parámetro digest se calcula aplicando el algoritmo MD5 (RFC 1321) a una cadena consistente en una marca de tiempo (incluyendo <) seguido de un secreto compartido. Este secreto compartido es una cadena conocida solo por el cliente y el servidor. Se debe tener un gran cuidado para prevenir una revelación no autorizada del secreto, ya que su conocimiento puede permitir a cualquier entidad hacerse pasar por el usuario. El parámetro digest es un valor de 16 bytes que se envía en formato hexadecimal, utilizando caracteres ASCII en minúsculas.

Cuando el servidor recibe el comando APOP, verifica el digest proporcionado. Si el digest es correcto, el servidor envía una respuesta afirmativa y la sesión entra en el estado de transacción. Si no, envía una respuesta negativa y la sesión permanece en el estado de autorización.

Notar que conforme incrementa la longitud de los secretos compartidos, aumenta la dificultad de derivarlos. Como tales, los secretos compartidos deben ser cadenas largas (considerablemente más largas que el ejemplo de 8 caracteres mostrado abajo).

AUTH mecanismo

Argumentos: una cadena que identifique un mecanismo de autenticación IMAP4 (definición en IMAP4−AUTH).

Restricciones: sólo puede darse en el estado de autorización.

Definición: El comando AUTH se refiere a un mecanismo de autenticación al servidor por parte del cliente. Si el servidor soporta este mecanismo, lleva a cabo el protocolo para la identificación del usuario. Opcionalmente, también procede con un mecanismo de protección para las subsiguientes interacciones del protocolo. Si este mecanismo de autentificación no es soportado, el servidor debería rechazar el comando AUTH enviando una respuesta negativa.

El protocolo de autentificación consiste en una serie de cuestiones por parte del servidor y de unas respuestas del cliente, específicas de este mecanismo de autentificación. Una pregunta del servidor, es una línea que consiste en un carácter "+" seguido de un espacio y una cadena codificada en base 64. La respuesta del cliente es una línea que contiene otra cadena codificada en base 64. Si el cliente desea cancelar la autentificación, debe emitir una línea con un único "*". Si el servidor la recibe, rechazará el comando AUTH.

Un mecanismo de protección proporciona integridad y privacidad a la sesión del protocolo. Si se utiliza un mecanismo de protección, este será aplicado a todos los datos que se envíen en la conexión. El mecanismo de protección tiene efecto inmediatamente después de que un CLRF concluya con el proceso de autenticación del cliente y de la respuesta positiva del servidor. Una vez el mecanismo de protección se hace efectivo, el flujo de bytes de comandos y respuestas se procesa en buffers de ciphertext (texto cifrado). Cada buffer es transferido en la conexión como un flujo de bytes seguidos de un campo de 4 bytes que representan la longitud de los siguientes datos. La longitud máxima de los búferes de ciphertext se define en el mecanismo de protección.

No es necesario que el servidor soporte algún mecanismo de autentificación, y tampoco es necesario que los

mecanismos de autentificación soporten mecanismos de protección. Si un comando AUTH falla, la sesión permanece en el estado de autorización y el cliente puede probar con otro AUTH o bien con otro mecanismo como la combinación USER/PASS, o el comando APOP. En otras palabras, el cliente puede pedir tipos de autentificación en orden decreciente de preferencia, con USER/PASS o APOP como últimos recursos.

SI el cliente completa la autentificación satisfactoriamente, el servidor de POP3 emite una respuesta afirmativa y se entra en el estado de transacción.

TOP mensaje

Argumentos: un número de mensaje, que si aparece no se puede referir a ningún mensaje marcado como borrado; y un número no negativo de líneas.

Restricciones: solo puede darse en el estado de transacción.

Definición: Si el servidor emite una respuesta positiva, entonces ésta es multilínea. Después del +OK inicial, el servidor envía las cabeceras del mensaje, la línea en blanco separando las cabeceras del cuerpo, y luego el número de líneas del cuerpo del mensaje.

Si el número de líneas requeridas por el cliente es mayor del número de líneas del cuerpo, el servidor envía el mensaje entero.

UIDL [mensaje]

Argumentos: un número de mensaje opcional. Si está presente no debe referirse a un mensaje marcado como borrado.

Restricciones: solo puede darse en el estado de transacción.

Definición: Si se da un argumento, el servidor emite una respuesta afirmativa con una línea que contiene información del mensaje. Esta línea se llama unique−id listing.

Si no se da ningún argumento y el servidor emite una respuesta afirmativa, la respuesta dada es multilínea. Después del +OK inicial, por cada mensaje en el maildrop, el servidor responde con una línea con información de ese mensaje.

Para simplificar el análisis, todos los servidores deben tener un mismo formato de unique−id listing, que consiste en el número de mensaje, un espacio y el unique−id del mensaje. Después no hay más información.

El unique−id listing de un mensaje es una cadena arbitraria determinada por el servidor, que consiste en 70 caracteres entre 0x21 y 0x7E (hexadecimal), los cuales identifican únicamente un mensaje en el maildrop y los cuales permanecen a lo largo de las distintas sesiones. Esta persistencia es requerida incluso si la sesión termina sin entrar en el estado de actualización. El servidor nunca debería rehusar el unique−id en un maildrop dado a lo largo de todo el tiempo de existencia de la entidad que usa el unique−id.

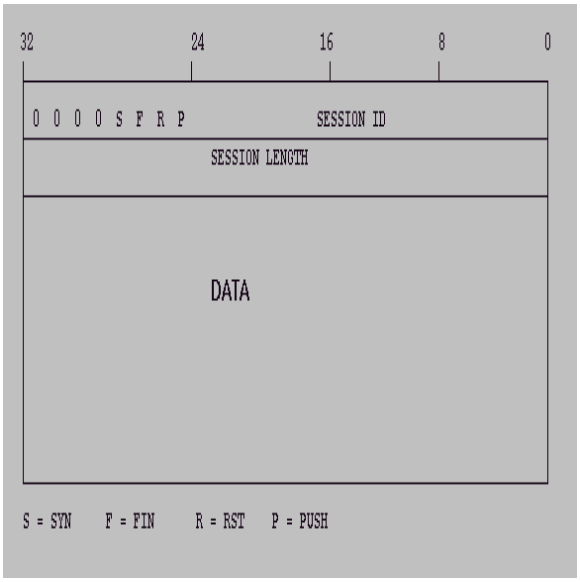
Mientras que generalmente es preferible para implementaciones de servidor almacenar los unique−id en el maildrop, la especificación tiene la intención de permitir que los unique−id sean calculados como trozos del mensaje. Los clientes deberían de ser capaces de manejar una situación en la que se den dos copias idénticas de un mensaje en un maildrop con el mismo unique−id.

1.6. SCP.

El modo SCP o simple communication protocol, es un protocolo simple que deja al servidor y al cliente tener múltiples conversaciones sobre una TCP normal, esto como es evidente declara que el protocolo SCP necesita montarse sobre es SCP. Este protocolo esta diseñado para ser simple de implementar.

El servicio principal de este protocolo es el control del dialogo entre el servidor y el cliente, administrando sus conversaciones y agilizadas en un alto porcentaje, este protocolo le permite a cualquiera de los dos establecer una sesión virtual sobre la normal.

La descripción de un formato de comunicación en las cabeceras enviadas por la red es la siguiente.



1.7. TCP/IP.

Este protocolo, el transfer communication protocol/Internet protocol, es el más usado actualmente en lo que a Internet se refiere. El TCP/IP es un conjunto de protocolos de comunicación, es decir convenciones particulares, creadas para permitir la colaboración y la partición de recursos entre más ordenadores conectados entre sí en la que está definida como red o network. Internet es en absoluto la más grande entre todas las redes que existen, debido a que logra conectar entre sí ordenadores personales y redes de menor amplitud en todo el mundo. Sobre Internet, de hecho, puede usted encontrar en conexión los ordenadores de instituciones del gobierno, militares, universidades y empresas privadas. Lo que permite a máquinas tan distintas por hardware y por prestaciones, comunicarse entre sí de manera casi transparente es el TCP/IP. Este constituye un tipo de lenguaje universal comprendido y utilizado por todas las máquinas que cooperan en red.

Estas son algunas definiciones de base. El nombre más apropiado para indicar este conjunto de protocolos, es Internet protocol suite, es decir colección de protocolos de Internet. El TCP y el IP son dos protocolos que pertenecen a esta colección.

Puesto que éstos son también los protocolos más conocidos, ha entrado en el uso común Llamar TCP/IP a toda la familia, aunque en algunas ocasiones una generalización parecida puede resultar un error. Cualquiera sea su nombre el TCP/IP representa una familia de protocolos, proveen a la gestión de las funciones de bajo nivel, que son necesarias para la mayoría de las aplicaciones. El TCP y el IP pertenecen a los protocolos de bajo nivel. Sobre esta base, se desarrollan otros protocolos que gestionan funciones particulares como enviar correo electrónico o conexión remota.

Todo esto está generalmente simplificado en un modelo cliente/servidor, en el cual el servidor se identifica con el ordenador que proporciona un servicio específico, a través del network y en el cual el término cliente se

identifica con el ordenador que explota este servicio, aunque con la palabra cliente incluya también aquellos programas que uno utiliza para tener acceso a estos mismos servicios (netscape).

El TCP/IP es un conjunto de protocolos a capas o niveles. Por ejemplo, cuando se quiere enviar un correo a través de Internet, lo primero que se necesita es definir un protocolo específico para el correo, o sea, un conjunto de reglas unívocamente reconocidas por todos los ordenadores conectados en red, y el cual tendrá la tarea de coger la carta que hay que enviar, añadirle el emisor y el destinatario enviarla a quien corresponda. Esto último es la tarea del protocolo específico de gestión del correo, que podría ser comparado al de una persona a la que un amigo muy ocupado le deja una carta y ella se encarga de ponerla en el sobre, escribir los datos de expedición y echarla al correo.

Evidentemente, si sólo existiese esta figura la carta se quedaría eternamente en el buzón sin que nadie se preocupase de hacerla llegar a su destino. Sin embargo, nuestro amigo muy ocupado tendría suerte ya que existe una camioneta del servicio de correos que dos veces al día vacía el buzón y transporta las cartas que allí encuentra a un lugar donde serán clasificadas y diferenciadas; allí su preciosísima carta será cuidada y mimada hasta que llegue al buzón del destinatario.

Para continuar con el paralelismo del ejemplo, diremos que el TCP/IP representa el sistema de transporte de Internet. En particular, el TCP se preocupa de 'empaquetar' bien todos los datos que le son suministrados por los protocolos de nivel superior; es posible que los subdivida en más partes si resultasen demasiado largos para un solo envío en red; asimismo recuerda lo que ha sido enviado, se acuerda de volver a enviarlo en el caso en que se hubiera perdido y controla que todo se realice de forma transparente para el usuario.

Ya que este tipo de operaciones es de uso general y es necesario tanto para enviar correo como para enviar ficheros u otras cosas, se ha pensado en hacer un protocolo propio, que pueda ser utilizado por muchos otros. Es precisamente por este motivo por lo que hemos definido protocolo de bajo nivel.

El TCP, sin embargo, no es el protocolo de nivel más bajo desde el momento en que éste utiliza el IP para realizar determinadas acciones. De hecho, a pesar de que el TCP sea muy utilizado, existen protocolos que prefieren no usarlo y que para funcionar sólo necesitan las funciones que puede ofrecer incluso el más humilde IP.

Este tipo de organización 'a capas' permite una gran eficiencia y un menor gasto de recursos.

Para terminar con un ejemplo, el envío de un mensaje de correo electrónico a través de Internet utiliza un sistema compuesto por cuatro capas:

Un protocolo de alto nivel específico para el correo 2.El protocolo TCP que es utilizado también por otros protocolos de alto nivel 3.El protocolo IP que se ocupa de la específica tarea de tomar los paquetes y enviarles a su destino 4.El protocolo del hardware específico, que se utiliza para la transmisión y la recepción de los datos

A este punto se nos aparece claro el motivo por el que el conjunto de los protocolos de Internet es llamado genéricamente TCP/IP. De hecho, estos son los protocolos más utilizados y de los que sólo pueden prescindir muy pocos protocolos de un nivel más alto.

Antes de terminar esta exposición general sobre el funcionamiento del TCP/IP es necesario introducir el concepto de data grama (datagram), que representa cada uno de los paquetes de informaciones que es enviado a través de la red. Como ya hemos dicho antes, un conjunto de informaciones demasiado largo que es subdividido en paquetes más pequeños, precisamente llamados data grama, que viajan individualmente en la red. Esto significa que si un fichero que se debe enviar es subdividido en 10 data gramas secuénciales, no está dicho que el cuarto llegue antes que el séptimo, desde el momento en que éste puede perderse o tomar un

camino equivocado. Será una tarea de los diversos protocolos el hacer que dicho paquete sea enviado nuevamente y colocado en el correcto orden secuencial a su llegada a destinación.

Y ahora, para evitar los ataques de los "puristas" diremos que a pesar de que los términos, data grama y paquete son muy a menudo utilizados como sinónimos, en realidad existe una diferencia. Mientras el data grama es específico del TCP/IP y representa la mínima unidad lógica utilizable por los diversos protocolos, el paquete es una entidad física bien presente para quien administra una red de tipo Ethernet. En el caso, por lo demás muy frecuente, que en un paquete viaje un solo data grama, la diferencia es sólo teórica pero existen también específicas configuraciones hardware de red que utilizan paquetes de dimensión menor respecto al del data grama individual. Entonces sucede que una data grama se descompone en más paquetes durante el envío a la red específica y que sea recompuesto a su llegada, de forma absolutamente transparente respecto al mismo data grama que... 'no se da cuenta' de haber sido descompuesto y luego recompuesto. Es evidente cómo en dicha situación los términos paquete y data grama no coinciden. Es una buena medida, por tanto, acostumbrarse a utilizar el término data grama cuando se habla del TCP/IP.

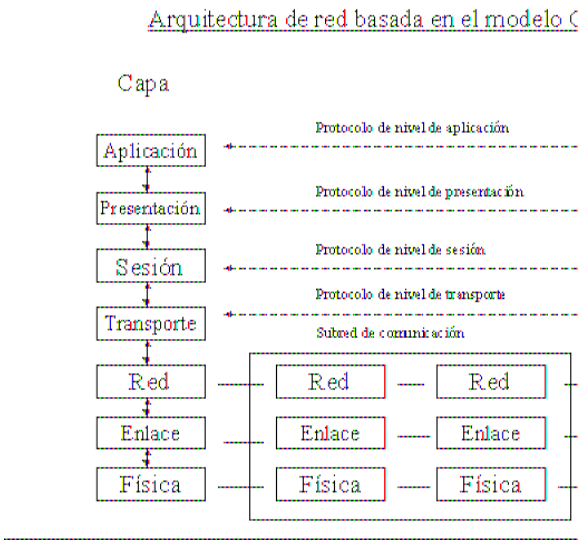
2. ELMODELO OSI

En 1.984, la organización internacional de estandarización (ISO) desarrolló un modelo llamado open systems interconection (OSI, interconexión de sistemas abiertos), el cual es usado para describir el uso de datos entre la conexión física de la red y la aplicación del usuario final. Este modelo es el mejor conocido y el más usado para describir los entornos de red. La arquitectura por capas que presenta el modelo OSI proporciona las siguientes ventajas.

- Reduce la complejidad. El entendimiento de cómo se realiza la interconexión y operación entre dos computadores se hace mucho más sencillo cuando el modelo se presenta por capas, esta división trae consigo sencillez en el aprendizaje de cada uno los procesos involucrados en esta comunicación y transferencia de información.
- Estandariza las interfaces. El estándar OSI plantea un modelo en el cual un dato pasa de un host a otro a través de varios niveles o capas, estas se encargan de una parte específica tanto en la parte de codificación como transporte y envió. Bajo este esquema una debe proveer servicios a la capa superior e inferior, para lo cual se debe establecer una interfaz única y estándar entre cada una de las capas. No importa el trabajo o la tecnología bajo la cual la capa opere, siempre habrá una interfaz estándar para interactuar con las diferentes capas.
- Facilita la ingeniería modular. Este modelo trae una gran ventaja cada vez más aprovechada, la posibilidad de diseñar equipos de comunicación divididos en módulos, cuya tarea esté orientada a cada una de las funciones de los niveles OSI. Se logra entonces una modularidad que facilita el desarrollo de la tecnología independientemente en cada una de las partes que la componen.
-

- Asegura la tecnología ínter operable. El hecho que las interfaces Sean III estándar entre cada una de las capas y la misma modularidad, permite que diferentes tecnologías se desarrollen en las capas, sin que se presente incompatibilidad entre éstas. Lo que logra, es entonces, una alta interoperabilidad entre cada tecnología, permitiendo el desarrollo por diferentes cambios tecnológicos.
- Acelera la evolución. La ingeniería modular es fuerte en este sentido, es decir, provee la forma para que cada ente que la compone se desarrolle por separado. Aquí en los niveles OSI, también se presenta este hecho. Esta división a la que nos hemos referido, ha permitido que cada capa se desarrolle vertiginosamente.
- Simplifica la enseñanza y el aprendizaje. Este esquema también provee una forma fácil de enseñar y aprender el proceso de comunicación Inter redes. Algo un poco complicado en los esquemas anteriores.

El modelo OSI se presenta en 7 capas, enumeradas desde la inferior (capa No. 1 física) hasta la superior (No.7 aplicación). A continuación la explicación de cada una de ellas. La grafica que se da a continuación ayudara a dar una idea del funcionamiento de las capas:



2.1. CAPA FISICA.

La capa física del modelo OSI es la que se encarga de las conexiones físicas de la computadora hacia la red, en este nivel están, por ejemplo, los estándares de cable de par trenzado que se deben usar para conectar una red, la forma en que las antena de microondas deben estar orientadas para comunicarse, y las características de propagación de ondas radiales. Define la conexión física de la red.

2.2. CAPA DE ENLACE DE DATOS.

Entrega los datos entre un nodo y otro en un enlace de red. La capa de enlace de datos, provee la transmisión de los bits en frames de información, es quien checa que los bits lleguen libres de errores a su destino y controla las secuencias de transmisión y los acuses de recibo de los mensajes recibidos. También se encarga de retransmitir los paquetes o frames que no han sido acusados por el otro extremo.

También este nivel controla el flujo de información entre dos nodos de la red.

Este nivel solo se encarga de la transmisión y recepción de datos entre dos nodos colindantes, y no es quien redirige o re−enruta paquetes (ese es el siguiente nivel, el nivel de red).

La capa de enlace provee la transmisión física a través del medio. Maneja el control de errores, la topología de la red, y el control del flujo. Esta capa se encarga de preparar los datos antes de enviarlos a través del medio físico.

Un ejemplo del nivel de enlace de datos es el estándar de ETHERNET o el de ATM.

2.3. CAPA DE RED

Las capas tres y cuatro manejan lo que comúnmente conocemos como networking o manejo de red. Es aquí donde se definen las rutas, destinos y caminos de llegada de un punto a otro de la red. Esto es comúnmente lo que manejan las capas de TCP/IP. Todo lo referente a los ROUTERS, BRIDGES, IP ADDRESS, IP MASK, ETC pertenecen a este nivel.

Un destino es un punto valido en la red donde los mensajes pueden llegar y ser enviados. Para llegar a un destino, debe de existir una ruta de comunicación, por lo general los puntos aislados de la red solo apuntan a una dirección default (que se llama el default gateway). En una red pequeña esto significa el ROUTER más cercano. Este Router tiene las direcciones más conocidas de la red y el enlace que conduce a ellas. Si la dirección que se le manda no es conocida por el ROUTER, este también tiene un default gateway que es un ROUTER en una red más grande, así se va pasando de ROUTER a ROUTER MAYOR, hasta llegar al Internet backbone, que es una red de SUPER ROUTERS que tienen todas las direcciones de Internet y el SUPER ROUTER más cercano a ellas.

Las funciones de esta capa también pueden ser capaces de reconfigurar la red para que los datos fluyan por un camino u otro si es que un enlace se cae.

Esta capa finalmente determina el mejor camino para mover los datos de un lugar a otro.

Maneja el direccionamiento de los dispositivos y supervisa la ubicación de los dispositivos en la red. Los enrutadores operan en esta capa.

2.4. CAPA DE TRANSPORTE.

Ahora, si mandamos un archivo grande, este archivo deberá de ser dividido en pedazos que puedan ser transmitidos por la red, estos pedazos viajan por la red y al llegar al destino deben ser acomodados de la manera en que fueron enviados (pueden desacomodarse por que pueden tomar diferente ruta se acaso una se congestiona o se cae). La forma de reacomodar los paquetes, cuanto tiempo y como esperar por ellos son las funciones de la capa 4.

Esta capa segmenta y reensambla los paquetes de datos en un bloque de datos en un bloque de datos. Se encarga de la interconexión de los equipos. Aquí es donde se negocia el inicio y terminación de una comunicación y la cantidad de paquetes a enviar. Algunas de sus funciones más importantes son las siguientes:

- Segmenta las aplicaciones de las capas superiores.
- Establece una conexión extremo−extremo.
- Envía segmentos de un host extremo a otro.
- Opcionalmente, asegura la confiabilidad de los datos.
- Se encarga de la conexión, reconocimiento, transmisión.

2.5. CAPA DE SESIÓN.

La capa de sesión es la encargada de ordenar o decidir a donde deben de ir los datos, además, indicar cuantos datos se enviaran o recibirán en cierto destino de la red.

Una comunicación en la red tiene dos tipos. Con conexión lógica (conection oriented, como el TCP) o sin conexión lógica entre los nodos (conection less como el UDP).

En la comunicación con conexión primero se establece una conexión lógica (una serie de mensajes se envían para saber primero si es que podemos establecer la comunicación entre los nodos). Una vez que la comunicación ha sido establecida, entonces los datos fluyen entre los dos destinos de la red. Cuando la comunicación ya no es necesaria entonces la conexión se libera.

La capa de sesión establece, mantiene y maneja las sesiones entre las aplicaciones. Es una comunicación interhost.

2.6. CAPA DE PRESENTACIÓN.

Un protocolo de comunicaciones debe ser diseñado para que diferentes versiones y sistema lo puedan usar, de modo que los datos se deben de tener en un formato definido y documentado. Por ejemplo una página de HTML debe tener campos como el puerto, la dirección URL, y el texto del mensaje. Esos campos serán transmitidos como bits y bytes y hay un documento (el estándar de HTML) que me indica en que parte del mensaje va cada pedazo de la pagina.

Precisamente de esto es lo que se encarga la capa de presentación, recibe bits y bytes de las aplicaciones y las formatea de modo que sean octetos entendibles en una red. Recibe un mensaje con octetos de una red y los decodifica para que se conviertan en bits y bytes de una aplicación.

Esta capa provee la representación de datos y el formateo del código. Asegura que los datos que recibe de la red puedan ser utilizados por la aplicación, y asegura que la información enviada por la aplicación pueda ser transmitida en la red.

2.7. CAPA DE APLICACIÓN.

Todas las capas anteriores en el modelo OSI sirven como infraestructura de telecomunicaciones. Por si solas no hacen nada más que mantener en buen estado el camino para que fluyan los datos, la capa que hace posible que una red se pueda usar es la capa de aplicación.

Es aquí donde lo visible y lo más orientado a es usuario se genera. A esta capa pertenecen por ejemplo los Web browser, el FTP, el e−mail, el telnet, las presentaciones de shockwave, los java applets y demás.

Una aplicación en java solo tiene que saber en que dirección y en que puerto se localiza el nodo remoto y ordenar a las demás capas (por medio de un TCP API) que vayan a ese nodo remoto y le envíen información.

La capa de aplicación provee servicios de red a las aplicaciones de los usuarios. Por ejemplo, una aplicación de un procesador de palabras es servida por los servicios de transferencia de archivos en esta capa. La aplicación es lo que es tangible para el usuario en el monitor, es el programa que se ejecuta.

CONCLUSIONES

Los protocolos son una herramienta importante en el diseño de las redes ya que nos dan las reglas y/o normas para sacarle el mejor provecho a una red.

Cada una esta destinada para algo en especial o para hacer las cosas de una manera característica, es por eso que tiene tanta importancia para el futuro ingeniero de sistemas y computación su conocimiento.

Dentro de los protocolos definimos los más comerciales y/o importantes que se encuentran en la actualidad, y son:

- ftp.
- http.
- Ipx/spx.
- Nfs.
- pop3.
- Scp.
- Tcp/ip.

Cada uno fue descrito con sus características más importantes.

El modelo OSI representa un importante desarrollo en la manera de construir redes y transportar datos a través de ellas.

En este trabajo se le dio definición a sus 7 capas, que son desde la más alta, las siguientes:

- Capa de aplicación. Es el nivel último de las capas del sistema OSI, el que aloja el programa de red interactúa con el usuario.
- Capa de presentación. Maneja los datos de la aplicación, los acomoda en un formato que pueda ser transmitido en una red.
- Capa de sesión. Establece conexiones lógicas entre puntos de la red.
- Capa de transporte. Maneja la entrega entre un punto y otro de la red, de los mensajes de una sesión.
- Capa de red. Maneja destinos, rutas, congestión de rutas, alternativas de rutas.
- Capa de enlace de datos. Entrega los datos entre un nodo y otro en un enlace de red.
- Capa física. Define la conexión física de la red.

En este modelo, solo las capas que tengan otra capa equivalente en el nodo remoto podrán comunicarse, esto es, solo las capas que son iguales entre si se comunican entre sí.

El protocolo de capa solo se interesa por la información de su capa y no por la de los demás.

La información se pasa a las capas de abajo hasta que la información llega a la red. En el nodo remoto, la información es entonces pasada hacia arriba hasta que llega a la aplicación correspondiente. Cada capa confía en que las demás harán su trabajo, y lo único que le interesa es la forma en como los datos serán pasados hacia arriba o hacia abajo.

BIBLIOGRAFIA

http://www.gratisweb.com/alricoa/contenido.htm

http://www.cybercursos.net/cursos−online/protocolos.htm.

Http://www.geocities.com/txmetsb/index.htm.