

LABORATORIO

FTP

**Nombre: -** ​Camilo Claure Díaz  **Código:** ​ S5899-8

**-** ​Alexander Vargas Villagómez ​ **Código:** ​S5506-9

**Materia:** Redes y Telecomunicaciones I​

**Semestre:** ​ 5to semestre

**Carrera:** ​ Ing. De sistemas

**Docente:** ​Ing. Ariel Mendoza I.

**Santa Cruz 29 de marzo de 2018**

# 1. ANTECEDENTES

En la comunicación que se pueden entablar entre dispositivos conectados en una red se tienen los que utilizan protocolos del modelo TCP/IP.

Entre estos protocolos se tienen los que son utilizados para la transferencia de archivos de texto plano utilizados en las páginas web, los que se utilizan para el envío y recepción de correos electrónicos o lo que simplemente se utiliza para la transferencia de archivos. Este último se le llama FTP por su sigla en inglés, ​*File Transfer Protocol.*

# ​ 2. OBJETIVO

# ​2.1. OBJETIVO GENERAL

Establecer un servidor FTP entre dos computadoras con el uso de cables UTP y servidores

FTP para el envío de datos entre usuarios

# ​ 2.2. OBJETIVOS ESPECÍFICOS

▪ Recopilar información acerca del funcionamiento de los servidores FTP

▪ Comprar los materiales acordes a las limitantes económicas

▪ Armar el UTP de acuerdo a la estructura

▪ Comprobar la correcta conexión entre las computadoras con el comando “ping”.

▪ Se levanta la conexión FTP entre el usuario y el servidor

▪ Utilizar los comandos del FTP para la manipulación de archivos

▪ Capturar el comportamiento de todos estos comandos en el Wireshark

# 3. MARCO TEÓRICO

Este protocolo (File Transfer Protocol – FTP) permite acceder a un servidor de archivo remoto, el cual es un requisito fundamental en muchas aplicaciones distribuidas. En algunas instancias, múltiples clientes pueden acceder a un solo servidor de archivo, mientras que varias copias del mismo archivo pueden estar guardadas en varios servidores.

A un cliente FTP pueden acceder tanto un usuario en un terminal como un proceso de aplicación de usuario. Normalmente, un solo cliente puede soportar muchos de estos usuarios concurrentemente. Este proporciona a cada usuario un conjunto similar de servicios disponibles en los sistemas de archivos. Entonces, cada usuario puede hacer listas de directorios, obtener contenido de archivos, crear nuevos archivos, actualizarlos o eliminarlos y otras actividades. igualmente, un servidor FTP puede contestar a las peticiones de varios clientes concurrentemente. Al recibir cada petición, el servidor actúa con su sistema local de archivos para atender como si esta se hubiese generado localmente.

El cliente FTP permite al usuario especificar la estructura y el tipo de datos del archivo involucrado. Estas estructuras son similares a las de los sistemas de archivos. Se soportan 3 tipos de estructura de archivos.

* No estructurados
* Estructurados
* De acceso aleatorio

Se soportan cuatro tipos de formato de datos

● Binarios de 8 bits

* texto ASCII
* texto EBCDIC
* archivo binario de longitud variable

El servidor FTP accede a cada archivo desde su sistema de archivos locales y los transfiere al cliente FTP con la estructura definida con anterioridad. Como un archivo no estructurado puede contener cualquier tipo de dato, binario o texto, este se transfiere entre dos entidades de protocolo FTP en un tren de bits transparentes. El usuario interpreta entonces la secuencia de bits de acuerdo a los tipos de datos dentro de este.

Los archivos estructurados consisten en una secuencia de registros de tamaño fijo de un tipo definido. De aquí el contenido de tales archivos se transfiere normalmente en una cadena de bloques de tamaño fijo. Alternativamente, los contenidos pueden transferirse en modo comprimido, cuando los archivos son grandes. Los tipos de redundancia pueden ser explotados por cada entidad de protocolo FTP aplicando un algoritmo de compresión, previo acuerdo, al contenido de los archivos antes de transmitirlos. E el modelo OSI, la capa de presentación ejecutaría esto.

Los archivos de acceso aleatorio están compuestos por registros de tamaño variable, y son referidos como páginas, es decir, como un archivo paginado. Cada registro/página tiene un encabezamiento asociado, el cual incluye una longitud, el tipo de campo total. Cada página se transfiere entre las dos entidades del protocolo de la misma forma.

En el caso de transferencias de modo bloque y modo comprimido, las dos entidades FTP controlan la transferencia de los archivos grandes con controles especiales. Esta acción es similar a los servicios de sincronización de la capa de sesión del modelo OSI.

Para manejar peticiones múltiples concurrentemente, cada servidor FTP y cliente –si es que éste también puede soportar múltiples peticiones– crea una nueva instancia de protocolo FTP o proceso FTP para cada nueva petición recibida. Usualmente hay un solo proceso principal, que es el número de la puerta bien conocida, al cual se pasan todas las nuevas peticiones. Este proceso maestro crea un nuevo proceso para esta conexión, realizando varias funciones de control asociadas con la sesión, que incluyen el procedimiento de derecho de acceso (LOGON con password) y la descripción de la estructura y los tipos de datos asociados de los archivos que han de transferirse. Éste también define si se usará algún tipo de compresión y qué algoritmo de compresión. Muchos fabricantes han creado otro proceso para manejar los datos transferidos actualmente asociados con la sesión. En tales casos, una sesión simple FTP

involucra dos conexiones de transporte: una para intercambiar mensajes de control y la otra para transferir contenidos de archivos. Esto, por supuesto, es totalmente transparente para el usuario y el software del servidor. El formato de los mensajes asociados con el contenido de los archivos se determina por la estructura del archivo. El formato de los mensajes intercambiados entre dos procesos de control FTP –las unidades de protocolo de datos FTP– debe también ser acordado en sintaxis para asegurar que éstas tengan el mismo significado (y que, por tanto, serán interpretadas de la misma manera) en ambas computadoras. Con este propósito se emplea el formato NVT.

Ahora, para la conexión entre computadoras se creó​ una​ red de área local con un cable de ethernet cruzado. Este cable se constituye en un cable UTP y conectores RJ-45. El término cruzado se refiere a la disposición interna de los pequeños cables internos.

Para que sea de cruzado el cable los extremos deben tener las dos normas existentes la 568A y 568B, uno en cada punta (ver anexo).

# ​ 4. MARCO PRÁCTICO

Primero necesitaremos fabricar el cable de ethernet que utilizaremos para la conexión para ello ocuparemos dos conectores RJ45 un cable utp y una crimpadora se alista los materiales que se van a usar luego se empieza a cortar las puntas de los cables se debe cortar una medida adecuada ya que si cortamos mucho el cable con el pasar del tiempo se dañara y si dejamos muy corto el cable el sellado con la crimpadora será mucho más difícil realizar , una vez hecho el corte procedemos a seguir el código de colores (el código de colores depende el tipo de red que estemos armando) una vez seguido los anteriores pasos antes de realizar el último paso que es el de sellar con la herramienta se verifica por última vez el código de colores para evitar fallos y una vez echo en un extremo se lo realiza en el otro extremo y cuando tenemos el cable listo se verifica el cable para ver que todo esté bajo control. **ver anexo**​

Para poder entender el funcionamiento de esta conexión lo explicaremos mediante el modelo de protocolo TCP\IP

**Capa de acceso a la red**​: En este laboratorio serán el cable utp y los conectores RJ45 que poseen 9 pines.

**Capa de Internet**​: En esta capa se utilizan las direcciones IP privadas de la red local

**Capa de transporte**​: En esta capa utilizaremos el protocolo http para el envío de datos

**Capa de aplicación:** ​Aquí ocuparemos el protocolo de transferencia FTP

para el creado del servidor ftp seguimos los siguientes pasos

-primero debemos activar el servidor FTP ya que el Windows lo tiene desactivado por defecto - Luego partiendo del panel de control nos dirigimos a herramientas administrativas y posteriormente a internet service information aquí procederemos a crear y configurar lo que sería el servicio ftp

* una vez creado el sitio FTP necesitaremos brindarle los permisos necesarios al usuario para que pueda realizar el envío de datos ya que la cuenta que se le asigna automáticamente es una cuenta que tiene restricciones de acceso por ende necesitamos crear una cuenta personalizada al usuario que contenga el nombre del usuario y su contraseña y le habilitaremos la lectura y escritura.
* Después de haber configurado correctamente el servidor y haber creado la contraseña algo que debemos tomar en cuenta es de que el firewall de la computadora no permite conexiones remotas por cuestiones de seguridad

Terminada la parte de configuración ahora procederemos a verificar la conexión en este caso utilizaremos el comando ping seguido de la ip destino y veremos si los paquetes están siendo enviados correctamente luego de esto procederemos realizar la conexión primero abrimos la consola de Windows “cmd’ en este pondremos el siguiente comando

ftp seguido de la ip de la computadora que será el servidor. una vez la consola nos indique que la fue completada con éxito ya tendremos acceso a una carpeta específica del servidor y podremos usar los siguientes comandos

|  |  |
| --- | --- |
| **Orden** | **Descripción** |
| help | Muestra todos los comandos admitidos por el servidor FTP. |
| status | Se usa para mostrar algunas de las configuraciones de la máquina cliente. |
| binary | Este comando cambia del modo ASCII (envío de documentos de texto) al modo binario (envío de archivos en modo binario, es decir, archivos sin texto, como lo son por ejemplo las imágenes o los programas). |
| ascii | Cambia de modo binario a modo ASCII. Este es el modo predeterminado |
| type | Muestra el modo de transferencia actual (binario o ASCII) |
| user | Permite reiniciar una sesión en el sitio FTP actual con un nombre de usuario diferente. Luego, se le pedirá que introduzca una contraseña nueva. |
| ls | Crea una lista de todos los archivos que se encuentran en el directorio actual. El comando ​**Is-I** brinda información adicional de los archivos. |
| pwd | Muestra el nombre completo del directorio actual. |
| CD | El comando significa ​*change directory*​ (cambiar el directorio) y se usa para pasar a un directorio diferente. El comando ​**cd...**​ se utiliza para tener acceso al directorio principal. |

|  |  |
| --- | --- |
| mkdir | El comando ​**mkdir**​ (en UNIX, o ​**md**​ en Microsoft) se utiliza para crear un directorio dentro del directorio actual. El uso de este comando se reserva para los usuarios que tengan acceso permitido. |
| rmdir | El comando ​**rmdir**​ (en UNIX, o ​**md**​ en Microsoft) se utiliza para eliminar un directorio dentro del directorio actual. El uso de este comando se reserva para los usuarios que tengan acceso permitido. |
| get | Este comando permite recuperar un archivo que se encuentra en el servidor.    Si el comando aparece seguido del nombre de un archivo, el archivo remoto se transfiere a la máquina local, dentro del directorio local actual. Si aparece seguido de dos nombres de archivos, el archivo remoto (el primer nombre) se transfiere a la máquina local en el directorio local actual con el nombre del archivo especificado (el segundo nombre). Si el nombre del archivo contiene espacios, será necesario introducido entre comillas. |
| put | Este comando se utiliza para enviar un archivo local al servidor.    Si el comando aparece seguido del nombre de un archivo, el archivo local se transfiere al servidor en el directorio remoto actual. Si el comando aparece seguido de dos nombres de archivos, el archivo local (el primer nombre) se transfiere al servidor en el directorio remoto actual, con el nombre del archivo especificado (el segundo nombre). Si el nombre del archivo contiene espacios, será necesario introducido entre comillas. |
| open | Cierra la sesión actual y abre una sesión nueva en otro servidor FTP. |
| close | Cierra la sesión actual, pero deja al ​*software*​ FTP activo. |
| bye | Desconecta el ​*software*​ cliente del servidor FTP y lo coloca en modo inactivo. |
| quit  .........................................................  .. | Desconecta el ​*software*​ cliente del servidor FTP y lo coloca en modo inactivo. |

y luego de esto utilizaremos el wireshark para observar el comportamiento de la conexión desde que se inicia la conexión hasta su finalización

# ​ 5. CONCLUSIONES Y RECOMENDACIONES

## 5.1. CONCLUSION

Al finalizar el proyecto se llegó a las siguientes conclusiones

* Se pudo armar satisfactoriamente el cable de red
* Las pruebas de envío de información fueron exitosas
* El ftp ​es muy rápida y directa en el acceso de la conexión que tiene el servidor
* Permite transferir archivos de un sistema operativo a otro, transferir archivos en una sola​ conexión y transferir archivos en ambos sentidos.
* Permite ​copiar archivos de gran tamaño
* Se pudo establecer exitosamente la comunicación cliente-servidor mediante FTP • ​Los datos e información de inicio de sesión se envían sin encriptación a través de la red.
* ​No se puede automatizar procesos.
* Las capturas de WireShark fueron exitosas

# ​ 5.2. RECOMENDACIONES

Las recomendaciones para el proyecto son las siguientes:

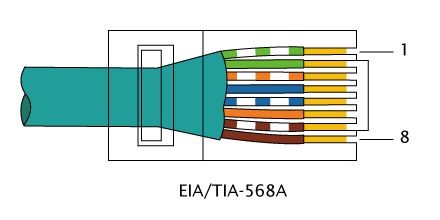
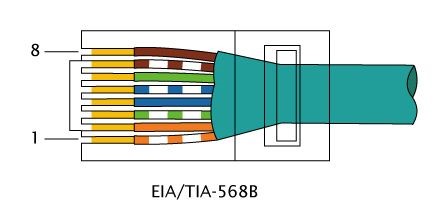
* Se requiere tener conocimiento sobre la consola de comandos del sistema operativo
* Son necesarias ciertas habilidades para armar el cable
* ​Las conexiones FTP no son confiables, por lo tanto, se necesita un certificado de seguridad.

## ​ 6. Fe de erratas

Los problemas que se presentaron fueron los siguientes:

* Al principio no se tomaron en cuenta los usuarios, por lo tanto, regresaba error 550
* Desconocimiento del uso correcto de los comandos
* Firewall de Windows no permite el envío de paquetes

# 7. ANEXOS



**Anexo 1**

