

UNIVERSIDAD NACIONAL AUTONOMA
DE MEXICO

FACULTAD DE INGENIERIA

DISEÑO Y DESARROLLO DE PRACTICAS PARA LA CREACION DEL LABORATORIO DE REDES DE DATOS

TESIS PROFESIONAL
QUE PARA OBTENER EL TITULO DE:

INGENIERO EN COMPUTACION
PRESENTAN:

YOLANDA CELIS MARES
JORGE ARTURO ACEVEDO CAÑEDO

DIRECTOR: ING. NOE CRUZ MARIN
CODIRECTOR: M.C. MA. JAQUELINA LOPEZ BARRIENTOS

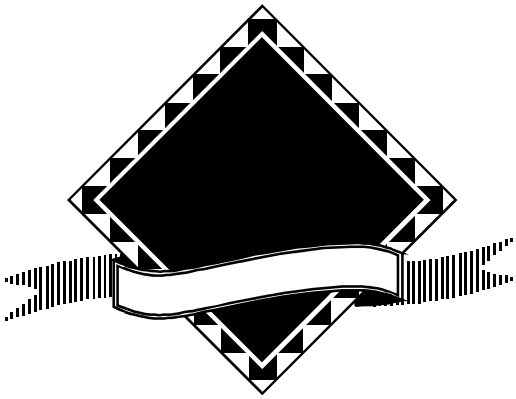


MEXICO, D.F. **2005**

DEDICATORIAS Y AGRADECIMIENTOS

A Dios:

Gracias por permitirme vivir, llenarme de tus bendiciones diariamente y por estar a mi lado siempre.



A mi Mamá:
Yolanda Mares.

Gracias por tu cariño, por ese gran apoyo y amor que nos brindas día a día a toda la familia, por tu esfuerzo y trabajo que has realizado para sacarnos adelante a mis hermanos y a mí, por darme esta carrera y ser lo que soy.
Dios te bendiga y ayude siempre.
Te quiero.

A mi Papá:
Roberto Celis.

Gracias por tu amor y por enseñarme que todo lo que hiciera en la vida lo realizara lo mejor posible.
Te quiero.

A mi hermano Beto:

Gracias por todo tu cariño, por estar siempre conmigo, por el apoyo que siempre me has brindado y por ser un ejemplo a seguir con lo emprendedor que eres.
Te quiero.

A mi hijo Dany:

Gracias por quererme tanto, eres una bendición de Dios, gracias también por ser tan bueno y comprensivo y hacerme tan feliz.
Dios te bendiga y ayude siempre.
Te quiero.

A mi hermano Oscar:

Gracias igualmente por tu cariño, por toda la ayuda que siempre me das y por estar siempre a mi lado en cada momento de mi vida. Dios Te bendiga y ayude siempre.
Te quiero.

A Familiares y Amigos.

Yolanda Celis Mares

DEDICATORIAS Y AGRADECIMIENTOS

A LA FACULTAD DE INGENIERIA DE LA UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO Y A TODOS MIS PROFESORES.

Por su formación como Ingeniero.
Gracias.

A mi profesor:
Ing. Noé Cruz Marín.

Por su tiempo, dedicación y gran apoyo profesional; como director de tesis y por su gran amistad y ayuda.
Gracias.



A mi profesora:
M.C. María Jaquelina López Barrientos.

Por su tiempo, gran apoyo, ayuda y comprensión en la dirección de este trabajo de tesis.
Gracias.

Yolanda Celis Mares

DEDICATORIAS

A Dios:

Por haberme brindado
la oportunidad de vivir,
Por darme padres con
una gran calidad moral,
Por darme la gracia de
llegar a este camino.

A mis padres:



Por ser la piedra
angular en mi vida
y mi ejemplo a seguir,
Por ser los cimientos
que me impiden caer,
Por ser la construcción
que me brinda resguardo,
Y ser lo más valioso
que tengo en mi vida.

A mis hermanos:

Por su apoyo y comprensión
a lo largo de mi vida.
Por estar a mi lado en los
buenos y malos momentos,
Por ser parte de mi vida.

A mi novia:

Por ser mi motivo para
seguir adelante.
Por darme los momentos
más hermosos de mi vida.
Por ser tu misma y
permitirme estar a tu lado.

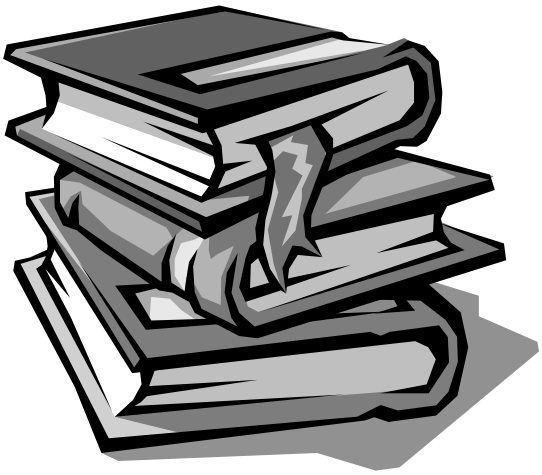
Acevedo Cañedo Jorge Arturo

A Familiares y Amigos.

AGRADECIMIENTOS

A la Universidad Nacional Autónoma de México:

Por ser mi alma mater,
Por ser la institución
que me hace sentir
orgulloso de ella.



A la Facultad de Ingeniería y mis profesores:

Por haber contribuido en
mi formación como persona,
Por compartir sus
conocimientos conmigo,
Por sus regaños que me hicieron
mejorar académicamente.

A mi Director y Codirector de tesis:

Por su apoyo, tiempo y dedicación
a lo largo de este trabajo de tesis,
Por sus regaños y consejos que
me permiten corregir mis errores,
Pero sobre todo por su paciecia
y por brindarme su amistad.

Índice

Índice Temático

Introducción 1

Capítulo 1 Conceptos Básicos

1.1. Definición de Redes 8
1.2. Objetivo de las Redes de Datos 8
1.3. Ventajas de usar Redes de Datos 8
1.4. Clasificación de Redes 9
1.4.1. Clasificación por Extensión Geográfica 9
1.4.1.1. Redes de Área Local (LAN – Local Area Network) 9
1.4.1.2. Redes de Área Metropolitana (MAN – Metropolitan Area Network) 10
1.4.1.3. Redes de Área Amplia (WAN – Wide Area Network) 10
1.4.2. Clasificación de Redes por su Topología 10
1.4.2.1. Topología Física 11
1.4.2.1.1. Topología en Estrella 11
1.4.2.1.2. Topología en Anillo 13
1.4.2.1.2.1. Token Ring 14
1.4.2.1.3. Topología en Bus 14
1.4.2.1.4. Topología de Árbol 16
1.4.2.1.5. Topología de Malla 17
1.4.2.2. Topología Lógica 18
1.4.2.2.1. Topología Lógica Bus 18
1.4.2.2.2. Topología Lógica Anillo 19
1.5. Esquemas de Acceso al Medio 20
1.5.1. Esquemas basados en Contienda 21
1.5.1.1. Redes Ethernet 22
1.5.2. Esquemas Token Passing 23
1.5.2.1. Redes Token Ring 24
1.5.3. Esquemas Polling 25
1.6. Variaciones de las Principales Topologías 25
1.7. Otras clasificaciones 26
1.8. Tipos de Conexiones 27

Capítulo 2 Modelo OSI

2.1. Antecedentes 28
2.2. Estandarización 28
2.2.1. IETF (Internet Engineering Task Force) 29
2.2.2. Estándares de Internet y RFC 30
2.3. Modelo OSI 30
2.3.1. Capa 7: La Capa de Aplicación 31
2.3.2. Capa 6: La Capa de Presentación 33
2.3.3. Capa 5: La Capa de Sesión 34
2.3.4. Capa 4: La Capa de Transporte 34
2.3.5. Capa 3: La Capa de Red 35
2.3.6. Capa 2: La Capa de Enlace de Datos 35
2.3.7. Capa 1: La Capa Física 35
2.4. Encapsulamiento 35
2.5. Desencapsulamiento 38
2.6. Nombres de los Datos en cada Capa del Modelo OSI 39
2.7. Modelo TCP/IP 40
2.7.1. Descripción del Modelo de Capas de TCP/IP 41
2.8. Modelo SNA Arquitectura de Sistemas de Red 43

Capítulo 3 Capa Física del Modelo OSI

3.1. Capa Física 44
3.2. Medios de Transmisión Terrestre 45
3.2.1. Cable Coaxial 45
3.2.2. Par Trenzado 46
3.2.2.1. Categorías del Cable UTP 51
3.2.2.1.1. Categoría 1 51
3.2.2.1.2. Categoría 2 51
3.2.2.1.3. Categoría 3 51
3.2.2.1.4. Categoría 4 51
3.2.2.1.5. Categoría 5 52
3.2.2.1.5.1. Categoría 5 Mejorada (5e) 53
3.2.2.1.6. Categoría 6 54
3.2.2.1.7. Categoría 7 54
3.2.3. Fibra Óptica 57
3.3. Medios de Transmisión Aéreos 59
3.3.1. Radio 59
3.3.2. Microondas 59

3.3.3. Infrarrojo 60
3.3.4. Enlace Vía Satélite 61
3.3.5. Redes Inalámbricas 61
3.3.5.1. Comunicación en Redes Inalámbricas 61
3.3.5.2. Tecnología Inalámbrica 62
3.4. Estándares de la Capa Física: RS-232, RS-422,RS-449 64
3.4.1. Definición 64
3.4.2. Módems 65
3.4.3. RS-232, RS-422, RS-449 68
3.4.3.1. Estándar de Capa Física RS-232 68
3.4.3.2. Estándar de Capa Física RS-449, RS-422-A 69
3.4.3.3. Estándar de Capa Física RS-422-A/ V.11 70
3.5. Cableado Estructurado 70
3.5.1 Estándar EIA/TIA 568 71
3.5.1.1. Subsistema Entrada del Edificio 72
3.5.1.2. Subsistema Cuarto de Equipo 72
3.5.1.3. Subsistema Cableado Horizontal 73
3.5.1.4. Subsistema Cuarto de Telecomunicaciones 74
3.5.1.5. Subsistema Cableado Vertical (Backbone) 74
3.5.1.6. Subsistema del Área de Trabajo 74
3.5.2. Diagrama de Subsistemas 75
3.6. Equipo 76
3.6.1. Repetidores 76
3.6.2. Hubs 77

Capítulo 4 Capa de Enlace de Datos

4.1. Capa de Enlace de Datos del Modelo OSI 79
4.2. Transmisión Síncrona y Asíncrona 81
4.2.1. Introducción al Modo de Transferencia 81
4.2.2. Red Digital de Servicios Integrados de Banda Ancha 82
4.2.3. Los Modos de Transferencia 84
4.2.4. El Modo Síncrono 84
4.3. Protocolos de Enlace 84
4.4. Tarjetas de Red (NIC) 85
4.4.1. Generalidades de la Tarjeta de Red 85
4.4.1.1. Definición 85
4.4.1.2. Funciones de la Tarjeta de Red 85
4.4.1.3. Velocidad de Conexión 86
4.4.1.4. Tipo de Conexión 87

4.4.1.5. Conectores ISA y PCI 87
4.4.1.6. NIC Especializadas 88
4.5. Switch 89
4.5.1. Introducción 89
4.5.2. Tipos de Switches 90
4.6. Ethernet 90

Capítulo 5 Capa de Red

5.1. Capa de Red del Modelo OSI 93
5.1.1. Introducción 93
5.2. Direcciones IP y Subneting 93
5.2.1. Conceptos Básicos 93
5.2.2. Subredes (Subnetting) 95
5.2.3. Funcionamiento 97
5.2.4. Recomendaciones 98
5.2.5. Beneficios del Subnetting 99
5.2.6. Ejemplos Prácticos 99
5.2.7. Resumiendo 102
5.3 Enrutamiento 106
5.3.1. Características de Diseño 107
5.3.2. Clasificación 109
5.3.2.1 Estáticos o Dinámicos 109
5.3.2.2 Single-Path o Multipath 109
5.3.2.3 Plano o Jerárquico 110
5.3.2.4 Link State o Distance Vector 110
5.3.2.5 Intadominio ó Interdominio 110
5.4 Jerarquía en capas 111
5.4.1. Capa de Acceso 111
5.4.2. Capa de Distribución 112
5.4.3. Capa de Core 112
5.5 Tipos de Backbone 112
5.5.1 Backbone Switcheado 113
5.5.2 Backbone Enrutado 113
5.6 Protocolo de enrutamiento OSPF 113
5.7 Protocolo de Hello 114

Capítulo 6 Capa de Transporte

6.1. Capa de Transporte 116

6.2. TCP 118
6.2.1. Establecimiento de una Conexión 119
6.2.1.1. Inicio una Conexión entre dos Aplicaciones 119
6.2.2. Paquetes TCP 119
6.2.3. Rendimiento 121
6.3. UDP 122
6.3.1. Paquetes UDP 122
6.4. Sockets 122
6.4.1. El Programa Sock 122
6.4.2. Introducción a los Sockets 123
6.4.3. Características de los Sockets 124
6.4.3.1. Tipo de Sockets 124
6.4.3.2. Dominio de un Socket 124
6.4.4. Proceso de Comunicación Cliente-Servidor 125
6.4.4.1. Servidor 125
6.4.5. Funciones del Cliente 129
6.4.6. Cliente Servidor 131

Capítulo 7 Capa de Sesión

7.1. Capa de Sesión 133
7.2. Uso de Puertos de Comunicación 137
7.2.1. Puertos 137
7.2.2. Firewall 139
7.2.2.1. Tipos de problemas que puede evitar un Firewall 140
7.3. Llamada de Procedimientos Remotos (RPC) 141
7.3.1. Modelo Cliente-Servidor 142
7.3.2. Realización de Llamadas de Procedimientos Remotos (RPC) 143

Capítulo 8 Capa de Presentación

8.1. Capa de Presentación 146
8.2. Representaciones comunes de los Datos 147
8.2.1. ASCII 7 Bits 147

8.2.2. ASCII 8 Bits 148
8.2.3. Unicode 148
8.3. Compresión de Datos 149
8.3.1. Antecedentes 149
8.3.2. Definición 151
8.4. Encriptación 151
8.4.1. Bases de la Encriptación 151
8.4.2. Características de SSH 152
8.4.3. Importancia de usar SSH 153
8.5. Autenticación 154
8.5.1. Kerberos 154
8.5.1.1. Funcionalidad de Kerberos 156
8.5.1.2. La Información en el Ticket 158
8.5.1.3. Autenticación mutua 159
8.5.1.4. Administración 160

Capítulo 9 Capa de Aplicación

9.1. Capa de Aplicación 164
9.2. Protocolos de Aplicación del Modelo OSI 166
9.2.1.TELNET 166
9.2.2. FTP (Protocolo de Transferencia de Archivos) 166
9.2.3. SMTP (Protocolo Simple de Transferencia de Correo) 166
9.2.4. SNMP (Protocolo Simple de Gestión de Red) 166
9.3. Establecimiento de una Conexión de Transporte 167
9.4. HTTP Hyper Text Transfer Protocol (Protocolo para la Transferencia de Hipertextos) 169
9.4.1. Características Principales 170
9.4.2. Etapas de una Transacción HTTP 171
9.4.3. Estructura de los Mensajes HTTP 172
9.4.4. Comandos del Protocolo HTTP 173
9.4.5. Las Cabeceras de los Mensajes HTTP 174
9.4.6. Tipos de Cabeceras 175
9.4.6.1.Cabeceras Comunes para Peticiones y Respuestas 175
9.4.6.2.Cabeceras sólo para Peticiones del Cliente 175
9.4.6.3.Cabeceras sólo para Respuestas del Servidor HTTP 175
9.4.6.4.Códigos de Estado del Servidor 177

9.4.6.5. Códigos de Estado más Comunes del Servidor HTTP 177
9.4.6.6. Cachés de Páginas y Servidores Proxy 180
9.4.6.7. Magic Cookies 180
9.4.6.8. Resumen de las Versiones del Protocolo 181

Conclusiones y Comentarios Finales 182

Glosario 184

Bibliografía 197

Anexo A Requerimientos de Prácticas

Anexo B Prácticas de Nivelación

Introducción

Introducción

Uno de los grandes recursos del hombre para mejorar sus condiciones de vida ha sido el hecho de poder comunicarse eficientemente con otros. Así en los inicios de la humanidad, el perfeccionamiento del lenguaje permitió una mejor coordinación entre individuos para obtener de manera comunitaria beneficios que de otra forma, no hubiera sido posible obtener. A partir de entonces las formas de comunicarse entre seres humanos han ido adaptándose a las necesidades de desarrollo de la sociedad, y la comunicación ha sido y seguirá siendo un factor de éxito para este punto. Tanto el uso del lenguaje como la propia sociedad, se han visto influidos por los avances tecnológicos en las comunicaciones o incluso han llegado a depender de ellos.

Con la creación e invención de los primeros aparatos electrónicos como el telégrafo y el teléfono primero y la radio y la televisión después, el desarrollo de los medios de comunicación comenzó a tener un avance vertiginoso de tal manera que motivó posteriores descubrimientos dentro del campo de la electrónica y las telecomunicaciones.

Hoy en día las redes de computadoras facilitan la comunicación entre individuos ya sea de forma personal, dentro de una organización o entre organizaciones; permitiendo explotar información que nos puede llevar a mejorar nuestras condiciones de vida e incrementar la productividad empresarial siendo así una herramienta de trabajo esencial. La información, el conocimiento y la comunicación, son palabras clave que han definido continuamente la historia del hombre. El tener un acceso adecuado a la información se ha convertido no sólo en una forma de adquirir conocimientos, sino en una cuestión de supervivencia por lo que la persona que conozca la forma de acceder y localizar información en esa inmensa biblioteca, tendrá acceso desde su mesa a prácticamente todos los hechos e ideas conocidos.

La *informática* y las *telecomunicaciones* nos permiten acceder a un volumen inmenso de información de forma instantánea, facilitándonos además, el proceso de búsqueda, por lo tanto, están destinados a la difusión, almacenamiento y tratamiento de la información en cualquiera de sus formas. Siendo la fusión y evolución

de estos dos campos de la tecnología factores determinantes del cambio en el modo de vida y las costumbres de la sociedad de nuestra época.

Debido a este rápido progreso, estas áreas están convergiendo y las diferencias entre juntar, transportar, almacenar y procesar información desaparecen rápidamente, esto, aunado a la creciente demanda de técnicas de procesamiento de información cada vez más avanzadas, y profesionales preparados, capacitados y hábiles para obtener, procesar y distribuir información, nos hace recapacitar en que si bien los estudiantes de esta Facultad cuentan con el conocimiento teórico es menester atacar el problema que implica la falta de práctica en lo referente a estos temas, como lo son entre otros la conexión y manejo de los diferentes dispositivos vistos en la asignatura de redes, punto clave y crítico que podemos mejorar; por ello se pretende dar solución a esta problemática a través del presente proyecto de tesis por medio de la creación de un conjunto de prácticas de redes mediante las cuales las futuras generaciones puedan acceder a un aprendizaje teórico-práctico en el cual, sea factible adquirir conocimientos en el aula y reafirmarlos mediante la realización de estas prácticas de laboratorio.

Por último, el desarrollar un programa de estudios que reconozca las tendencias demográficas y tecnológicas cambiantes de los estudiantes y de la población en edad laboral, la globalización de la economía, los cambios en los requisitos de conocimiento y aptitudes para la fuerza laboral y el papel que desempeña la tecnología en la educación, nos hace reflexionar y aportar material de apoyo para el aprendizaje de las redes que vaya más allá de la simple educación tradicional basada en la formación teórica y conceptual de las redes de datos para ayudar a los estudiantes a desarrollar un conocimiento y las aptitudes prácticas a través de la creación de prácticas que se pongan a prueba en un ambiente controlado, esto es, en un laboratorio que se asemeje mucho a un entorno real de trabajo que permita a los estudiantes trabajar con los diversos componentes de infraestructura que utiliza la tecnología de las redes de datos, dando como resultado estudiantes con una sólida formación teórica pero con la suficiente habilidad práctica en lo referente a la tecnología de redes de datos.

Objetivos

- Llevar a cabo un análisis del nuevo temario de la materia Redes de Datos, que permita diseñar y desarrollar una serie de prácticas de laboratorio que concedan la elaboración de un manual de prácticas acordes a la asignatura.

- Tomar como base las consideraciones, recomendaciones y modificaciones que se planteen para la asignatura en la actual revisión a los planes y programas de estudio de la carrera de Ingeniero en Computación.

Estructura de la Tesis

El nuevo temario para la materia Redes de Datos (antes Redes de Computadoras) de la carrera Ingeniero en Computación se comenzará a impartir desde el semestre 2006-2 en nuestra Facultad. En este contexto el marco teórico del presente trabajo de tesis, se desarrolló tomando como base el temario de la asignatura.

Para ello, fue preciso llevar a cabo un análisis del nuevo temario de la materia Redes de Datos, para así poder desarrollar cada uno de sus temas, entre los cuales se encuentra el modelo de referencia OSI, en el cual esta basada la tesis, en esta vía fue necesario confrontar cada una de las capas del modelo, para de esta manera atacar y proponer una solución a la problemática de la falta de práctica en el manejo y conexión de los diferentes dispositivos de comunicación. Dando como resultado la realización de las siguientes prácticas que a continuación se mencionan; y que pretenden ser una propuesta para el Laboratorio de Redes de Datos, que actualmente está creado.

La estructura básica de las prácticas propuestas para el laboratorio, se dio a lo largo del desarrollo y la prueba de cada una de ellas, es decir, que la actual versión de las mismas, es la adaptación final, la cual se ha logrado a través del perfeccionamiento que se ha venido dando en cada una de las pruebas realizadas dentro del laboratorio con la participación de los alumnos de la Facultad. Dicha estructura contempla los siguientes aspectos fundamentales:

- Titulo de la práctica
- Objetivo(s) de Aprendizaje
- Conceptos teóricos
- Equipo y material necesario:
  - Material que debe traer el alumno
  - Equipo de laboratorio
- Desarrollo
- Cuestionario
- Conclusiones u Observaciones
- Cuestionario Previo

Cabe citar que cada una de las prácticas cuenta con sus requerimientos para la realización de las mismas, los cuales se encuentran en el anexo A de la tesis.

Esta serie de prácticas vienen incluidas dentro de un CD, el cual está colocado en la pasta posterior de este ejemplar.

No. de práctica y nombre.	*Capa OSI*	*Objetivos de Aprendizaje del alumno al finalizar la práctica*
PRÁCTICA 1. Construcción de cables UTP para conexión directa y cruzada	1	Tendrá la capacidad de construir cables de conexión directa y cruzada con normas ANSI/EIA/TIA T568-A y T568-B.
PRÁCTICA 2 Componentes del Cableado Estructurado Norma ANSI/EIA/TIA 568	1	Se familiarizará con aspectos generales del cableado estructurado mediante la instalación de un *jack* y roseta, utilizando cable UTP Cat.5e ó superior.
PRÁCTICA 3 Instalación de una Red Básica en Plataforma Windows XP de Microsoft	2	Tendrá la capacidad de instalar y configurar una tarjeta de red y crear una Mini-LAN conectando la computadora a la red existente a través de la instalación ya configurada en el laboratorio. Y a través del cable crossover construido en la práctica No.1.
PRÁCTICA 4 Instalación de una Red	2	Tendrá la capacidad de instalar y configurar una tarjeta de red y crear

Básica en Plataforma LINUX		una Mini-LAN conectando dos computadoras utilizando un cable de conexión cruzada.
PRÁCTICA 5 Enrutamiento de paquetes	3	Se familiarizará con el manejo de algunas herramientas del Sistema Operativo Linux, como son route y traceroute, y sus similares en Windows, como son route y tracert, enfocadas al enrutamiento de paquetes a través de la red.
PRÁCTICA 6 Análisis de Paquetes	3	Conocerá y comprenderá los fundamentos de la monitorización de redes, encapsulado de unidades a diferentes niveles y demultiplexación de las mismas. Adquirirá la habilidad necesaria para realizar los filtros adecuados. Reafirmará conocimientos teóricos acerca del protocolo ARP mediante observación de casos reales.
PRÁCTICA 7 TCP y UDP	4	Podrá utilizar un programa que le permita enviar y recibir información utilizando los protocolos TCP y UDP y reafirmando conceptos teóricos. Podrá solicitar una página web a un determinado servidor desde un navegador. Tendrá la capacidad de conectarse al servidor SSH local haciéndole una petición de enviar la versión de éste. Será capaz de crear un socket servidor y un socket cliente.
PRÁCTICA 8 Filtrados de Puertos TCP/UDP mediante un Firewall	5	Se familiarizará con el manejo de los puertos. Será capaz de manejar uno de los firewalls. Creará filtros, permitiendo o negando el uso de puertos conocidos. Manipulará el uso del messenger y el navegador de Windows e Internet

		Explorer. Permitirá y negará paquetes del comando ping.
PRÁCTICA 9 SSH: Secure Shell	6	Conocerá la importancia de utilizar el protocolo SSH (Secure Shell) y su herramienta OpenSSH. Será capaz de iniciar una sesión remota a través de SSH, utilizando autenticación por contraseña. Iniciará una sesión remota con clave pública, generando las claves. Podrá transferir claves públicas al servidor.
PRÁCTICA 10 Aplicación Web Apache	7	Podrá instalar y configurar una aplicación típica de la capa 7 del Modelo OSI en Plataforma Linux. Será capaz de Instalar y configurar de forma segura un servidor web en su versión básica, pero con posibilidad de transferencias seguras mediante SSL (Secure Sockets Layer). Conocerá las principales directivas de configuración de un servidor web apache.

Prácticas de Nivelación

Estas prácticas se encuentran en el Anexo B de la tesis; y fueron desarrolladas con el propósito de dar una breve introducción y una serie de comandos básicos sobre Linux; para todos aquellos que no estén familiarizados o que deseen dar un repaso sobre dicho sistema operativo.

Las prácticas de nivelación son las siguientes:

PRÁCTICA 1 de Nivelación Introducción a LINUX

Los objetivos de aprendizaje son los siguientes:

Al finalizar la práctica el alumno conocerá las características básicas del sistema operativo *Linux* y utilizará los comandos básicos para la creación y uso del sistema de archivos de *Linux.*

PRÁCTICA 2 de Nivelación Shell: Comandos Básicos de Linux

Los objetivos de aprendizaje son los siguientes:

Al finalizar la práctica el alumno conocerá el funcionamiento de los comandos básicos de *Linux* y adquirirá otra alternativa de interacción con el sistema, además del ambiente gráfico.

Glosario y Bibliografía

El trabajo de tesis cuenta con un glosario de términos y una bibliografía de la cual fue extraída la información para el presente trabajo.

Capítulo 1

Conceptos Básicos

Capítulo 1
Conceptos Básicos

1.1. Definición de Redes

Una red de datos se puede definir como una colección de dispositivos interconectados entre sí, capaces de intercambiar algún tipo de información digital, tales como bases de datos bancarias, acervos bibliográficos, correos electrónicos, etc.

1.2. Objetivo de las Redes de Datos

Son muchas las organizaciones que cuentan con un número considerable de computadoras en operación y con frecuencia alejados unos de otros. Inicialmente cada una de estas computadoras puede haber estado trabajando en forma aislada de las demás pero, en algún momento, la administración puede decidir interconectarlos para tener así la capacidad de extraer y correlacionar información referente a toda la compañía.

Es decir el objetivo básico es *compartir recursos*; esto es, hacer que todos los programas, datos y equipos estén disponibles para cualquiera de la red que lo solicite, sin importar la localización del recurso y del usuario.

1.3. Ventajas de usar Redes de Datos

Hoy en día la mayoría de las organizaciones se encuentran ampliamente distribuidas de manera geográfica. Muchos de los departamentos requieren un flujo de información constante entre ellos, siendo la mejor opción cuando estos no se encuentran cerca geográficamente, una red de datos que les permita un fácil intercambio de información.

Las redes de computadoras permiten compartir recursos, tales como capacidad de procesamiento (procesamiento paralelo), espacio en disco, o disponibilidad de conexión a otras redes. Así mismo, permiten la integración del flujo de información requerido por empresas, y permiten que muchas de las tareas que típicamente

estaban atadas a una oficina, puedan hacerse desde prácticamente cualquier lugar que cuente con una línea telefónica.

1.4. Clasificación de Redes

Las redes pueden clasificarse bajo distintos parámetros:

- Por extensión geográfica,
- Por topologías,
- Otras clasificaciones.

1.4.1. Clasificación por Extensión Geográfica

Dicha clasificación obedece a la forma en que geográficamente están dispuestas las redes. Así tenemos:

1.4.1.1 Redes de Área Local (LAN - Local Area Network)

Una red de área local es una colección de 2 o más computadoras que están localizadas dentro de una distancia limitada una de la otra ya sea conectada directa o indirectamente, un ejemplo sencillo es la red instalada físicamente dentro de un edificio de oficinas. Las LAN permiten a las empresas que emplean tecnología informática compartir de forma eficiente elementos, tales como archivos e impresoras y posibilitar las comunicaciones, como el correo electrónico. Unen entre sí datos, comunicaciones, computadoras y servidores de archivos. Las LAN están diseñadas para lo siguiente:

- Operar dentro de una zona geográfica muy limitada,
- Permitir a muchos usuarios acceder a medios de alto ancho de banda,
- Proporcionar conectividad tiempo completo a los servicios locales,
- Conectar físicamente dispositivos adyacentes.

1.4.1.2 Redes de Área Metropolitana (MAN - Metropolitan Area Network)

Son redes de computadoras de tamaño superior a una LAN, soliendo abarcar el tamaño de una ciudad. Son típicas de empresas y organizaciones que poseen distintas oficinas repartidas en una misma área metropolitana, por lo que, en su tamaño máximo, comprenden un área de unos 10 kilómetros.

Se pueden clasificar en públicas o privadas. La primera de ellas se refiere a las de tipo privado, las cuales son implementadas en zonas de campus o corporaciones con edificios diseminados en un área determinada. Su estructura facilita la instalación de cableado de fibra óptica. El segundo tipo de redes MAN se refiere a las redes públicas de baja velocidad, las cuales operan a menos de 2 Megabits por segundo en su tráfico.

1.4.1.3 Redes de Área Amplia (WAN - Wide Area Network)

Una red de área amplia cubre una zona geográfica extensa conectando ciudades y países. Esto aplica a las redes corporativas con sedes en distintas ciudades y/o países. Un ejemplo claro de una red WAN es Internet, la cual une una diversidad de redes geográficamente dispersas.

1.4.2. Clasificación de Redes por su Topología

Topología es el diseño físico de los equipos, cableado y otros componentes de red. Sinónimos de topología son: Distribución física, Diseño, Diagrama, Mapa, etc.

La topología involucra directamente: El equipo requerido, La capacidad del Equipo, crecimiento de la red, Administración de red.

La interconexión de los equipos requiere de ciertos tipos de cables, tarjetas de red, sistema operativo, etc. La topología se ve ampliamente relacionada con el cableado estructurado.

Su definición contiene dos partes: la *topología física*, que es el diseño real del cableado (medios), y la *topología lógica*, que define cómo los hosts acceden a los medios.

1.4.2.1. Topología Física

1.4.2.1.1 Topología en Estrella

Estas redes se basan en un nodo central -computadora central o servidor de red-, que concentra todo el tráfico de información, y al que están conectadas todas las estaciones (ver figura 1.1). Normalmente el tamaño de la red depende de la capacidad de la computadora central.

Es decir; todos los equipos se conectan a través de un componente central llamado Hub o Concentrador, la utilización de cable es mayor que en un bus lineal.

La ventaja es que si un equipo falla sólo se interrumpe la actividad para ese equipo.

La desventaja es que, en caso de que el nodo central se "caiga", la red deja de funcionar. Es decir; si el concentrador es el que falla toda la red quedará incomunicada.

Puntos a considerar:

- *Instalación*. La topología estrella es moderadamente difícil de instalar. El diseño de una red es simple pero usted debe instalar un segmento de medio de transmisión por cada conexión que se desee hacer. El cableado de la topología estrella requiere mas cableado que otras topologías.
- *Reconfiguración*. La topología estrella es relativamente fácil de instalar y reconfigurar.
- Mover, agregar y cambiar no involucran más que la conexión entre el dispositivo cambiado y un puerto de un hub.

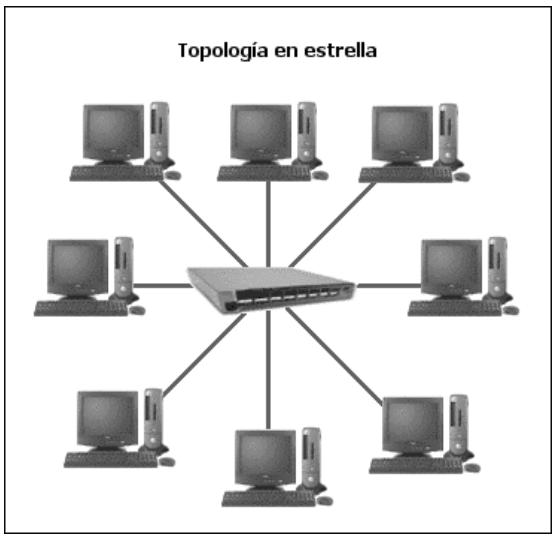


Figura 1.1 Topología en Estrella.

Protocolo por Sondeo

Sondeo/bit final. Protocolos de capa de enlace de datos síncronos bit a bit que indican la función de una trama. Si la trama es un comando, un 1 en este bit indica un sondeo. Si la trama es una respuesta, un 1 en este bit indica que la trama actual es la última trama de la respuesta.

Concentradores

Componente central de una topología estrella. La mayoría son Elementos Activos que regeneran la señal y la retransmiten de la misma forma que un repetidor, requieren corriente eléctrica para operar, son también llamados *repetidores multipuertos*. Algunos son Pasivos como por ejemplo Paneles de Cableado, o centrales de clavijas, solo actúan como punto de conexión.

Los concentradores Híbridos, permiten distintos tipos de cable, y permiten expandir las redes con el uso de más concentradores interconectados.

1.4.2.1.2 Topología en Anillo

En este caso la red se organiza de modo que cada nodo esté conectado solamente a otros dos, a través del medio de transmisión, conformando de esta manera un anillo físico y lógico. (Ver figura 1.2)

También el anillo puede ser con control centralizado, donde una de las estaciones tiene mayor jerarquía que las demás y controla las comunicaciones; o con control distribuido, donde todas las estaciones tienen la misma jerarquía.

Todos los equipos se encuentran interconectados en un círculo de cable, no hay terminadores, la información viaja alrededor del anillo en un solo sentido, cada equipo actúa como un repetidor y reconstruye la señal, es decir es una topología activa.



Figura 1.2 Topología en Anillo.

Puntos a considerar:

- *Instalación.* La topología anillo es moderadamente simple de instalar. Porque el anillo requiere un enlace cerrado, mas cable es requerido que la topología tipo de bus.
- *Reconfiguración*. La topología anillo viene a ser difícil de configurar. Los segmentos de anillo deben ser divididos o reemplazados con dos nuevos segmentos cada vez que un

segmento es cambiado. El anillo esta limitado a un número máximo de dispositivos.

1.4.2.1.2.1 Token Ring

Similar al Anillo, pero cada equipo recibe un Token, este se va enviando a cada uno de los clientes y solo lo recibe aquel cuya dirección coincida con la de destino del paquete, anuncia al cliente de envío que la computadora ha recibido el paquete y reenvía un nuevo Token.

Este proceso es llamado PASO DE TESTIGO. En un Anillo de 200 m el Token puede pasar hasta 10,000 veces por segundo. (Ver figura 1.3)

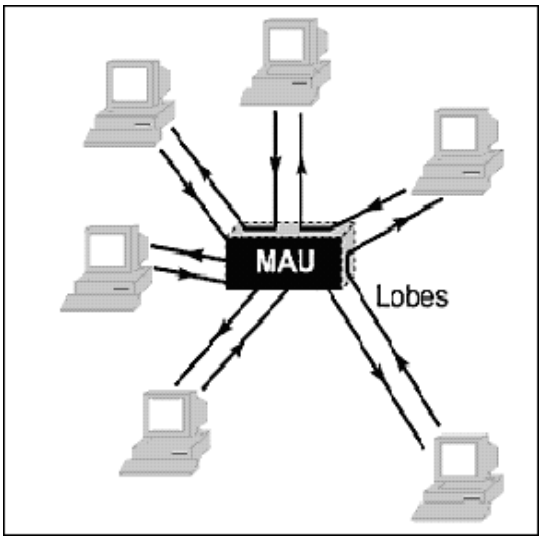


Figura 1.3 Topología Token Ring.

1.4.2.1.3 Topología en Bus

Esta es una de las topologías más utilizadas en las redes LAN, donde cada estación está conectada al medio de transmisión -canal o bus- y aunque alguna de ellas deje de funcionar la red puede seguir trabajando correctamente. (Ver figura 1.4)

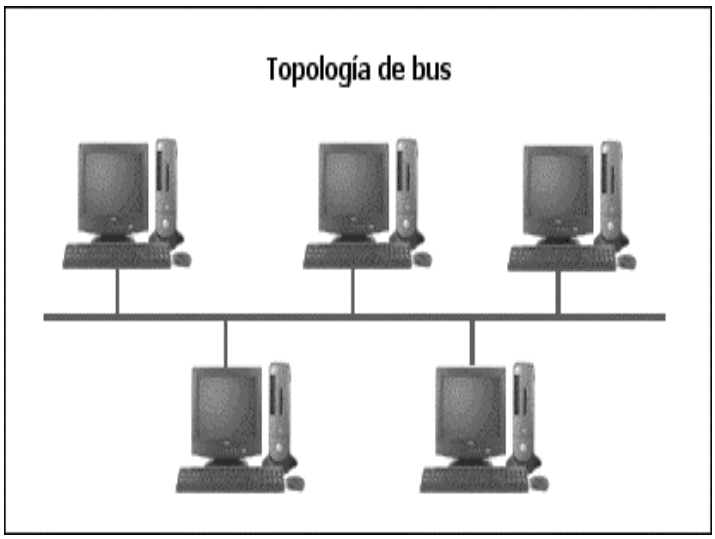


Figura 1.4 Topología en Bus

También conocida como Bus lineal, método más sencillo de comunicación, consiste de un solo cable principal que conecta a todos los equipos en una sola línea. Tecnología de comunicación pasiva, no es responsable de mover los datos solo los escucha.

Envío de la Señal:

Los datos se transmiten en forma electrónica a todos los equipos en el cable, solo un equipo puede enviar mensajes a la vez. Debido a este factor la fluida comunicación depende de la cantidad de equipos conectados al bus, la disminución de la velocidad de la información no sólo depende del número de equipos además de:

- Capacidad de Hardware
- Número de veces que un equipo transmite
- Tipo de aplicaciones de red
- Tipo de cable
- Distancia entre los equipos

Reflejo de la Señal

Como la información debe ser escuchada por todos los clientes de red debe viajar la señal de un extremo a otro por el cable de red, pero debe detenerse una vez que llega a un extremo.

Terminador

Es un dispositivo físico que se instala al final del cable para absorber las señales libres que llegan y liberar el bus para que otros cables puedan transmitir. Este Terminador es una resistencia que absorbe la corriente eléctrica del cable.

Desventajas:

- *Interrupción:* Si un equipo se desconecta o el cable se rompe la comunicación se inhabilita debido a que el cable se queda sin Terminador y la señal no se atenúa, a esto se le conoce Fallo de la Red.
- *Ampliación:* A medida que la red crece tanto en número de equipos como en extensión en ocasiones es necesario contar con elementos que nos permitan extender el bus estos pueden ser los siguientes:

    - *Conector Cilíndrico:* Atenúa la señal, no debe utilizarse en exceso.
    - *Repetidor:* Refuerza la señal y la regenera antes de reenviarla.

Puntos a considerar:

- *Instalación.* Una topología bus es relativamente fácil de instalar. Usted estira el cable desde un sitio a otro.
- *Reconfiguración*. Las topologías bus están construidas para minimizar la cantidad de cable requerido y mantener una distancia mediana entre los nodos, la reconfiguración tiende a ser moderadamente difícil.

1.4.2.1.4 Topología de Árbol

La topología de árbol combina características de la topología de estrella con la de bus. Consiste en un conjunto de subredes estrella conectadas a un bus. Esta topología facilita el crecimiento de la red. (Ver figura 1.5)

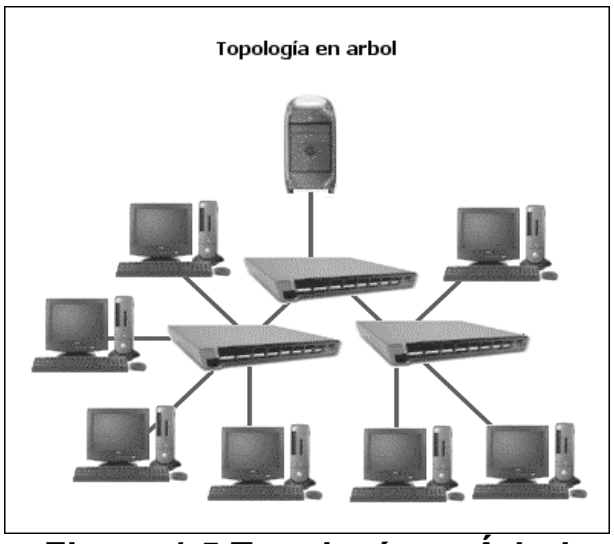


Figura 1.5 Topología en Árbol

1.4.2.1.5 Topología de Malla

En una topología de malla completa, cada nodo se enlaza directamente con los demás nodos (ver figura 1.6).

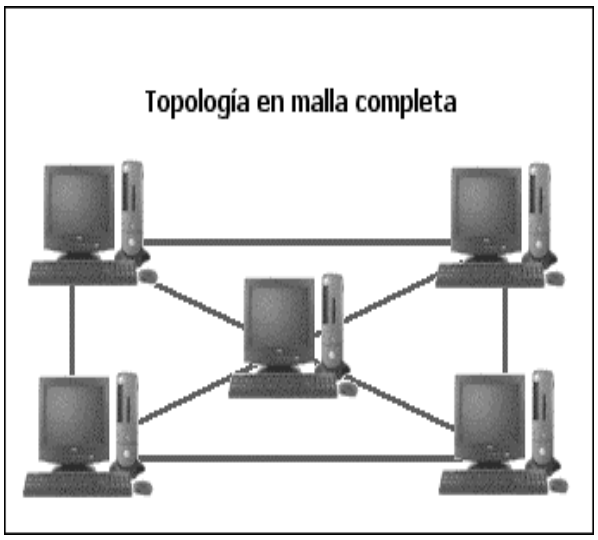


Figura 1.6 Topología en Malla Completa

Las ventajas son que, cada nodo se conecta físicamente a los demás, creando una conexión redundante, si algún enlace deja de funcionar la información puede circular a través de cualquier cantidad de enlaces hasta llegar a su destino. Además, esta topología permite que la información circule por varias rutas a través de la red.

La desventaja física principal es que sólo funciona con una pequeña cantidad de nodos, ya que de lo contrario la cantidad de medios necesarios para los enlaces, y la cantidad de conexiones con los enlaces se torna abrumadora.

1.4.2.2 Topología Lógica

Después de planear el esquema físico de la red. Se debe considerar el esquema lógico.

Hay dos topologías lógicas más comunes:

- Bus
- Anillo

La topología lógica de red es, en esencia, una estrategia para direccionar o encaminar el flujo de la señal.

Como se puede ver, el término *Tráfico de red y Colisiones* son comúnmente usadas en la terminología de red.

Las topologías lógicas son una parte fundamental de una red, porque las señales eléctricas deben mantenerse separadas y diferenciadas de las otras, para evitar choques y distorsiones.

Los dispositivos que envían la señal también deben mantener un orden. Los dispositivos deben hablar para tomar turnos, o mirar el tráfico de la red antes de enviar sus mensajes.

1.4.2.2.1 Topología Lógica Bus

En una topología lógica Bus, los dispositivos generan señales y las envían a través de la red.

Independientemente de la ubicación del receptor. Una topología lógica bus, puede ser solo usada con la topología física bus y estrella. (Ver figura 1.7)

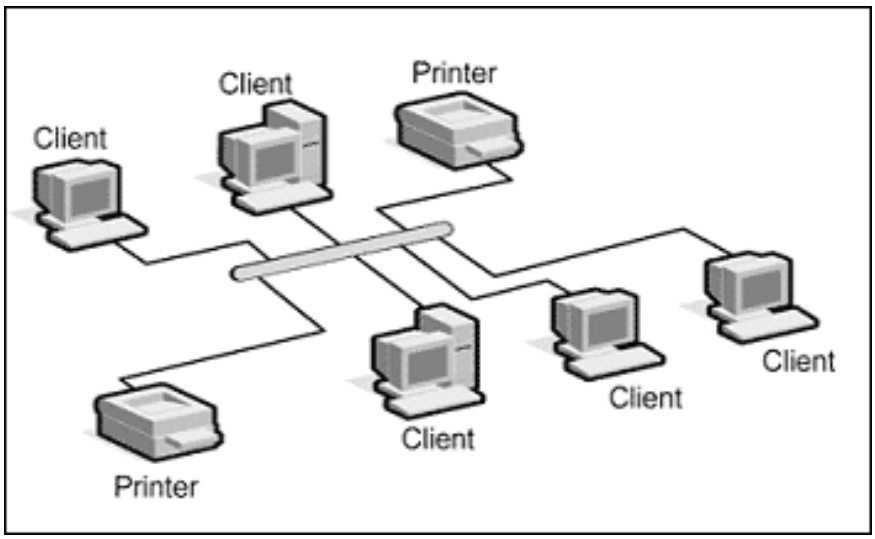


Figura 1.7 Topología Lógica Bus

El mensaje enviado a todos los dispositivos en una topología lógica bus contiene información que dice cuál dispositivo debe recibir el mensaje y los otros lo ignoran.

Una topología bus es una buena porque los dispositivos de red no están enterados de las ubicaciones físicas de los otros dispositivos.

No se puede dar a un dispositivo instrucciones para enviar mensajes directamente a otro dispositivo. Por ejemplo un dispositivo no puede conocer que otro dispositivo esta localizado a “tres nodos al sur sobre el lado derecho”. Por lo tanto, un dispositivo debe enviar el mensaje a todas las direcciones. Entonces cada dispositivo determina si el mensaje fue precisamente para ese dispositivo.

1.4.2.2.2 Topología Lógica Anillo

En una topología lógica anillo, la señal es generada y viaja a través de una ruta ya especificada en una simple dirección. (Ver figura 1.8).

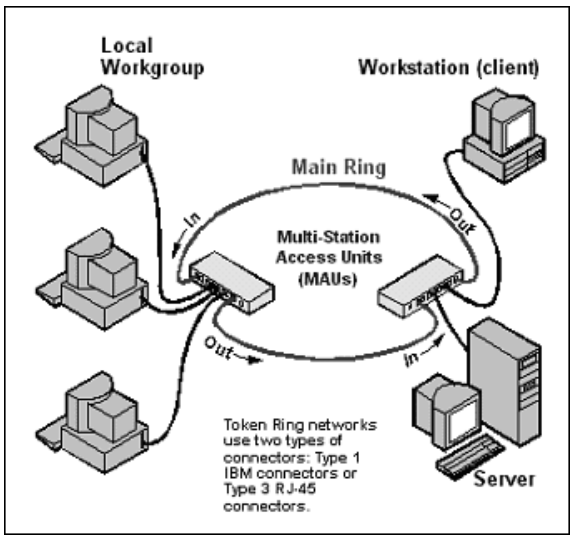


Figura 1.8 Topología Lógica Anillo

La topología lógica anillo puede ser usada con la topología física anillo y con la topología física estrella.

La diferencia entre el anillo lógico y bus lógico esta en que la señal enviada en una lógica bus va en todas direcciones. Las señales enviadas en un anillo lógico pueden solo ir en una dirección. Una topología física estrella puede manipular una topología lógica anillo, por que las señales entran en el hub y son enviados de regreso a los dispositivos de red en un orden predeterminado.

La topología física puede ser usada con cualquiera de las dos topologías. La estrella física es también relativamente fácil de instalar y fácil de reconfigurar. La topología física estrella es muy común.

1.5. Esquemas de Acceso al Medio

Un esquema de acceso al medio, es un juego de reglas que siguen las señales enviadas sobre el medio de transmisión en la red. Hay tres tipos de esquemas de acceso al medio:

- Contention
- Token Passing
- Polling

Estando en el medio de transmisión al mismo tiempo causan que las señales electromagnéticas colisionen, por eso el Esquema de Acceso al Medio (un juego de reglas para el control de tráfico de red), necesita estar en un lugar de control cuando los dispositivos de red están permitidos a transmitir señales de datos. Si los dispositivos de red operan sin un esquema de acceso al medio, los dispositivos transmitirán todas las veces que quieran, cuando ellos estén listos.

Algunas veces ellos transmiten al mismo tiempo, de manera que las señales se combinan y se dañan, al punto que hay pérdida en los datos de la señal, a lo cual se le llama colisión y ocasiona la destrucción de las comunicaciones.

No se puede operar una red a menos que se puedan controlar o eliminar las colisiones.

1.5.1. Esquemas basados en Contienda

Los esquemas basados en contienda permiten a los dispositivos de red transmitir datos cuando ellos deseen, independientemente de otros dispositivos en la red. Este esquema es simple y provee por igual derechos de acceso a todas las estaciones.

Desafortunadamente la estrategia de “transmita cuando esté listo” tiene una importante limitación: las estaciones algunas veces transmiten al mismo tiempo. Cuando esto sucede, el resultado es una mezcla de señales y la información se pierde.

Los esquemas basados en contención indican a todas las estaciones que deben escuchar el canal antes de transmitir. Si la estación que está escuchando detecta una señal ésta se abstiene de transmitir y trata de transmitir después.

Este esquema es llamado CSMA (Carrier Sense, Multiple Access), el cual controla y reduce las colisiones, pero éstas pueden continuar ya que si dos estaciones censan el cable, no detectan nada y transmiten al mismo tiempo, con toda seguridad se presentará una colisión, por lo tanto, este tipo de esquema debe también estar habilitado para detectar su ocurrencia. Si una colisión es detectada, la

señal es enviada otra vez, por lo que se refiere al protocolo CSMA/CD (CSMA con Collision Detection).

Los esquemas basados en contención, manejan en promedio condiciones de tráfico muy bien, pero pierden performance cuando el tráfico de la red llega a ser muy pesado y más colisiones podrían ocurrir.

1.5.1.1. Redes Ethernet

Ethernet es el estándar de red más popular que usa una topología lógica bus y puede ser conectada usando una topología física bus o estrella. (Ver figura 1.9)

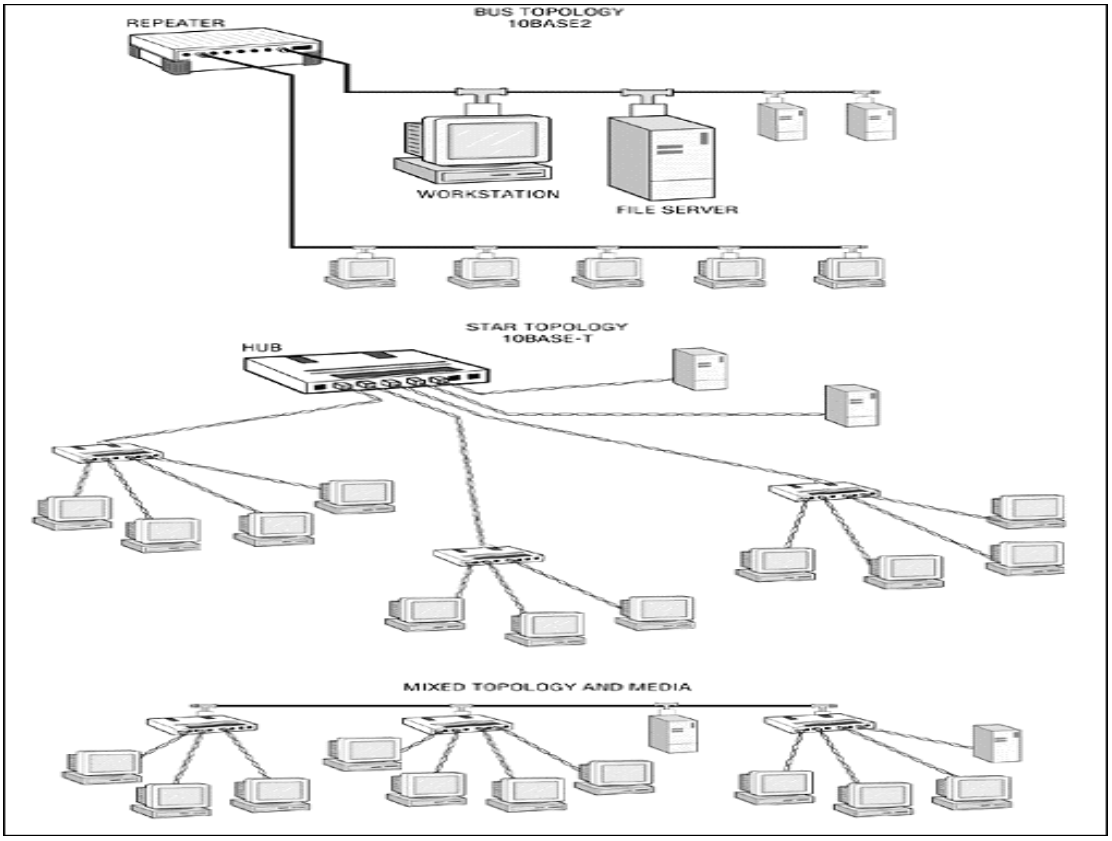


Figura 1.9 Redes Ethernet

Las redes Ethernet usan un esquema de acceso basado en contención. Las redes Ethernet mueven mensajes alrededor de la red en paquetes de información que incluyen la dirección de la estación origen, estación destino, señales de control y los datos mismos. Para

enviar paquetes, un dispositivo en la red debe primero escuchar y ver si algún otro dispositivo está haciendo uso del medio de transmisión. Cuando el cable (medio de transmisión) está libre de tráfico, el dispositivo envía sus paquetes.

Si dos dispositivos están intentando transmitir sobre el cable al mismo tiempo, el paquete puede colisionar físicamente con otro. El resultado puede ser el daño de los paquetes o paquetes no confiables.

Cuando la colisión ocurre, una señal es enviada para asegurar que la colisión ha sido reconocida alrededor de la red.

Los dispositivos compiten por el ancho de banda del cable para hacer sus transmisiones por una cantidad al azar de tiempo para asegurar que las colisiones sean eliminadas.

Cuando los dispositivos se enteran de la presencia de una señal en el cable, ellos revisan para asegurarse que el paquete no es un fragmento de otro paquete que ha sido dañado por una colisión.

1.5.2. Esquemas Token Passing

En esquemas Token Passing, una señal electrónica llamada Token viaja de un dispositivo a otro. Un Token es un mensaje especial que temporalmente da el control de acceso al medio al dispositivo que tiene el Token.

Moviendo el Token alrededor de todos los dispositivos, distribuye el control de acceso a todos los dispositivos de red.

Cada dispositivo conoce cual dispositivo recibirá el Token desde otra estación y sabrá a quien debe pasarlo. Cada dispositivo periódicamente consigue el control del Token, transmite sus datos y retransmite el Token al siguiente dispositivo para su uso.

Los protocolos limitan el tiempo que cada dispositivo puede controlar el Token.

Los esquemas Token Passing trabajan con las topologías físicas anillo y estrella.

Los esquemas Token Passing no permiten contienda. Un Token existe en el medio de transmisión y le da el turno de transmitir a la estación que tiene el Token. Sobre condiciones promedio de tráfico, el esquema Token passing es lento en relación al esquema basado en contención, pero en altas condiciones de tráfico el esquema de Token Passing tiene mayor rendimiento.

1.5.2.1. Redes Token Ring

Las redes Token ring combinan una topología física estrella y topología lógica anillo con el esquema de acceso al medio de Token passing. Esta es una configuración popular de red. (Ver figura 1.10)

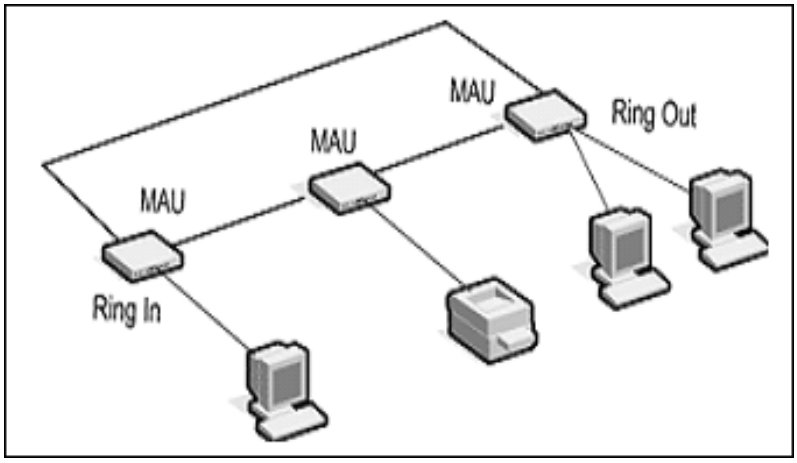


Figura 1.10 Redes Token Ring

Cuando una estación quiere transmitir al anillo, éste espera por el Token libre para comunicarse con otras estaciones. Cuando se hace eso, la estación fuente toma el Token libre y agrega datos a él.

La estación entonces envía el Token hacia el anillo. Desde ese momento el Token ocupado es pasado por cada estación alrededor del anillo, cada estación revisa sí la comunicación está dirigida hacia ella.

Si una estación no es la receptora del Token, ésta la reenvía al anillo; si una estación es la receptora, ésta copia los datos de la estación origen agregada al Token.

Finalmente éste agrega datos al Token para indicar que ha reconocido la dirección y que ha copiado los datos, entonces la estación envía el Token alterando al anillo. (Ver figura 1.11)

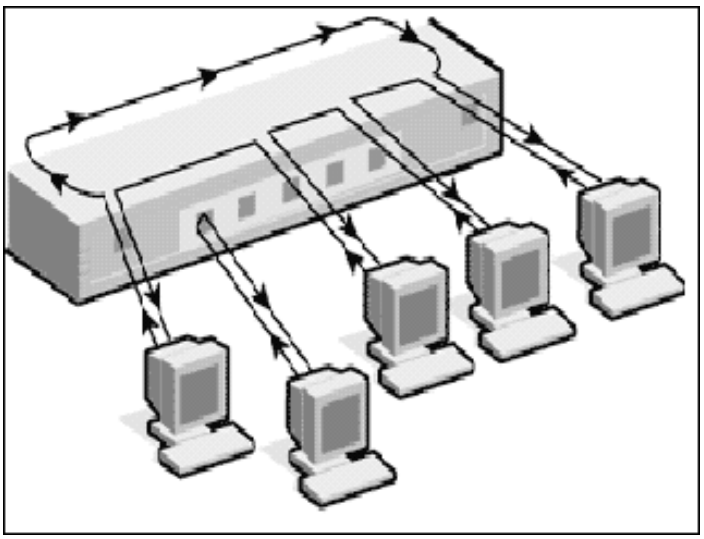


Figura 1.11 Token alterando el anillo.

El Token continúa alrededor del anillo hasta que éste alcanza la estación origen, cuando la estación ve que el Token ha sido recibido y los datos han sido copiados, ésta genera un nuevo Token libre, el cual buscará una estación que desee transmitir datos para repetir el proceso.

1.5.3. Esquemas Polling

Polling es un esquema de Acceso que designa un dispositivo (llamado controlador, primario o master) como administrador de acceso al medio.

Este dispositivo pregunta a los otros dispositivos (secundarios) en un orden predeterminado si ellos tienen información para transmitir.

1.6. Variaciones de las Principales Topologías

También existen topologías híbridas, las que surgen de combinar las topologías básicas anteriores con el fin de superar algunas limitaciones propias de estas últimas, por ejemplo, Red Jerárquica (estrella-estrella), Red Anillo-Estrella, Red Bus-Estrella, etc.

En la actualidad la mayoría de las topologías de red son combinaciones de topologías.

Bus Estrella

Varias redes estrella vinculadas con un bus, este bus lo forman concentradores conectados en cascada, si un equipo falla no interrumpe la comunicación de la red, si un concentrador es el que falla todos los clientes conectados al hub no podrán comunicarse, si el hub esta vinculado a otros concentradores también romperá la comunicación.

Anillo en Estrella

A veces llamado Anillo Cableado en Estrella, varios concentradores que contienen una topología estrella son conectados a un concentrador anillo.

1.7. Otras clasificaciones

Públicas

Internet: Internet es un conjunto de redes autónomas que se comunican a través del protocolo TCP/IP. Tiene una cobertura global, emplea una gran variedad de tecnologías que hacen posible que todos sus usuarios se comuniquen y usen los servicios que otras redes ofrecen.

Privadas

Intranet: Red de acceso restringido como lo es el uso exclusivo de empleados de una compañía.

Extranet: Es una intranet extendida donde la información puede ser accedida por empleados y clientes de una compañía (los externos requieren validación mediante una contraseña o permisos especiales).

1.8. Tipos de Conexiones

Existen dos maneras principales de conectar computadoras:

- **Punto a Punto**
  - Necesita un reloj común
  - Necesita un protocolo de comunicación.
- **Multipunto**:
  - Requiere un proceso llamado SONDEO (*Polling*).
  - Sondeo por llamado a Lista: (Ver figura 1.12).
  - Sondeo por secuencia: CPU manda un mensaje de atención a la 1a. terminal. Si no recibe respuesta, pasa a la segunda, y así sucesivamente: (Ver figura 1.13).

Direcc. de Terminales que habrían de intervenir	Prioridades de Atención	Atenciones
T1	1	2
T2	1	2
T3	2	1
T4	1	2

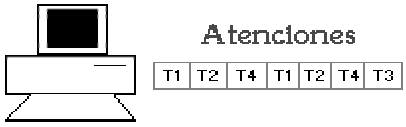


Figura 1.12. Sondeo por llamado a Lista.

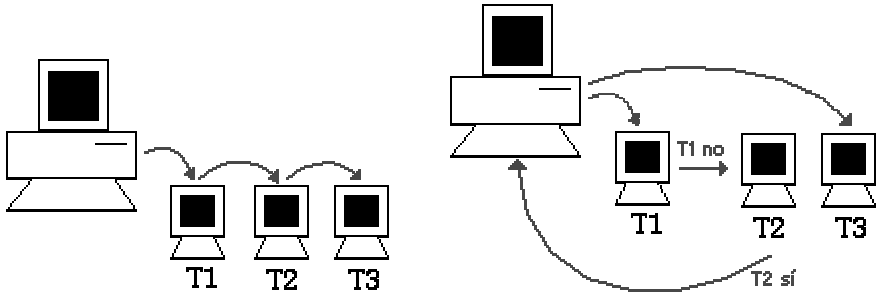


Figura 1.13. Sondeo por secuencia

Capítulo 2

Modelo OSI

Capítulo 2
Modelo OSI

2.1. Antecedentes

El primer desarrollo de las LAN, MAN y WAN fue en muchos sentidos caótico. A principios de los 80, se vieron notables aumentos en el número y tamaño de las redes. Mientras las compañías se daban cuenta que podían ahorrar dinero y ganar en productividad usando tecnologías de red, añadiendo redes y expandiendo las ya existentes casi tan rápidamente como se introducían las nuevas tecnologías de redes y los nuevos productos.

A mediados de los años 80, estas compañías comenzaron a experimentar las secuelas del crecimiento en todas las expansiones que realizaron. El problema fue incluso más agudo para las redes que emplearon especificaciones e implementaciones diferentes para comunicarse entre ellas. Comprendieron entonces que necesitaban apartarse de los sistemas de red propietarios.

Los sistemas patentados se desarrollan, poseen y controlan en privado. En la industria de las computadoras, patentado es lo opuesto a abierto. Patentado quiere decir que un grupo, o un pequeño grupo de compañías, controlan toda la evolución y utilización de una tecnología. Abierto significa el libre uso de la tecnología disponible para el gran público.

Para solucionar el problema de la incompatibilidad e incapacidad de comunicación entre los sistemas de red diferentes, la Organización Internacional de Normalización (ISO) investigó los esquemas de red, como DECnet, SNA y TCP/IP, para encontrar un conjunto de normas. Como resultado de la investigación, la ISO creó un modelo de red que podía ayudar a los fabricantes a crear redes que fuesen compatibles y que pudiesen interoperar con otras redes.

2.2. Estandarización

Conforme las primeras redes de datos fueron apareciendo, la interconexión entre ellas era sencilla cuando se trataba de los mismos

fabricantes, pero imposible cuando se trataba de fabricantes distintos. La necesidad de interconectarse por parte de los dueños de esas redes, y la búsqueda de nuevos mercados por parte de los fabricantes, hizo que estos últimos hicieran acuerdos en cuanto a la fabricación de hardware y software y así lograr una comunicación exitosa entre sus equipos. Pero no todos los fabricantes accedían a modificar sus tecnologías (tales como IBM). Pronto fue evidente que era necesario el establecimiento de organismos centrales que dirigieran el proceso de estandarización a nivel mundial para interfaces y protocolos de comunicación. Organismos como la ITU y la IEEE surgen y se comienza la estandarización de protocolos, haciendo posible la interconexión de equipos de diferentes proveedores. Algunos de los organismos de estandarización más reconocidos se listan a continuación:

- International Organization for Standardization (ISO)
- American National Standards Institute (ANSI)
- Electronic Industries Association (EIA)
- Institute of Electrical and Electronic Engineers (IEEE)
- International Telecommunication Union Telecommunication
- Standardization Sector (ITU-T, Antes CCITT)
- Internet Activities Board (IAB)

2.2.1. IETF (Internet Engineering Task Force)

El Grupo de Ingeniería de Internet (Internet Engineering Task Force), forma parte del Consejo de Arquitectura de Internet (IAB), que es un grupo asesor técnico que pertenece a la Sociedad de Internet (ISOC).

El IETF está dividido en grupos de trabajo, cada uno de ellos en directrices sobre un tema diferente relacionado con el establecimiento de normas de Internet. El número de miembros es abierto, pudiéndose unir cualquiera que esté interesado.

El objetivo principal de los grupos de trabajo del IETF es desarrollar y consensar los proyectos de Internet. Estos proyectos incluyen documentos oficiales de Petición de Comentarios o “Request

for Comments" RFC, que tienen que pasar por un proceso de aprobación previo para llegar a ser normas de Internet.

2.2.2. Estándares de Internet y RFC

En estos servicios y protocolos se incluyen elementos como: implementación del sistema de nombres DNS, las extensiones TCP/IP, las especificaciones de conversión de direcciones de red NAT.

Aunque muchas RFC tuvieron su origen en el IETF, cualquiera puede redactar propuestas de RFC. No todas las RFC describen normas, pero si un documento llega a ser una norma, ha tenido que pasar por estas tres fases:

- Propósito de norma
- Proyecto de norma
- Norma de Internet

El IEEE (Instituto de Ingenieros Eléctricos y Electrónicos) promueve el intercambio de información y desarrolla normas y especificaciones principalmente para las tecnologías de red del nivel más bajo (las que actúan en las capas de enlace y física). De particular interés para los profesionales de red son las especificaciones que constituyen el proyecto IEEE 802. El nombre se basa en la fecha en que se reunió el comité, 80 es el número del año (1980) y 2 el número del mes (febrero).

NOM – NORMA OFICIAL MEXICANA

Conjunto de normas jurídicas de carácter oficial, publicadas en el Diario Oficial de la Federación.

2.3. Modelo OSI

El modelo de referencia OSI, lanzado en 1984 por la ISO proporcionó a los fabricantes de equipo de red un conjunto de normas que podían facilitar una mayor compatibilidad e interoperabilidad entre los diferentes tipos de tecnologías de red producidos por muchas de las compañías de todo el mundo. El modelo OSI dividió el problema de

comunicación de red en tareas más sencillas y simples, pasando a entidades más pequeñas el poder desarrollar una actividad más específica.

El propósito del modelo de referencia OSI es acelerar el desarrollo de futuros productos de red. Aunque existen otros modelos, la mayoría de los fabricantes actuales relacionan sus productos con el modelo de referencia OSI.

El modelo de referencia OSI describe cómo la información en una computadora es transferida a una aplicación que reside en otra computadora. El modelo de referencia OSI es un modelo conceptual compuesto de 7 capas (Ver figura 2.1), cada una de ellas especificando funciones particulares, siendo cada capa razonablemente autocontenida, es decir que entre capas no existen funciones comunes. El modelo es considerado como un modelo de arquitectura para comunicaciones entre computadoras, y es usado como referencia en la comparación de diferentes tecnologías (por ejemplo ATM y HDSL).



Figura 2.1 Capas del Modelo OSI.

En el modelo de referencia OSI, existen siete capas numeradas. Cada capa muestra una función particular de la red. Esta separación de las funciones de la red se llama layering. Dividir la red en siete capas proporciona las siguientes ventajas:

- Divide la comunicación de red en partes más pequeñas, sencillas y fáciles de desarrollar y comprender.

- Facilita la normalización de los componentes de red, al permitir el desarrollo y el soporte de múltiples fabricantes.
- Permite que diferentes tipos de hardware y software de red se comuniquen entre sí.
- Impide que los cambios en una capa afecten a las otras, por lo que se pueden desarrollar más rápidamente.

La comunicación que procede de una capa del modelo, generalmente se da con otras tres capas: la capa inmediata superior, la capa inmediata inferior y la capa análoga (o peer) en la computadora a la que se comunicará.

Las tres capas superiores del modelo de referencia OSI se llaman capas de aplicación mientras que las restantes son llamadas de flujo de datos.(Ver figura 2.2)

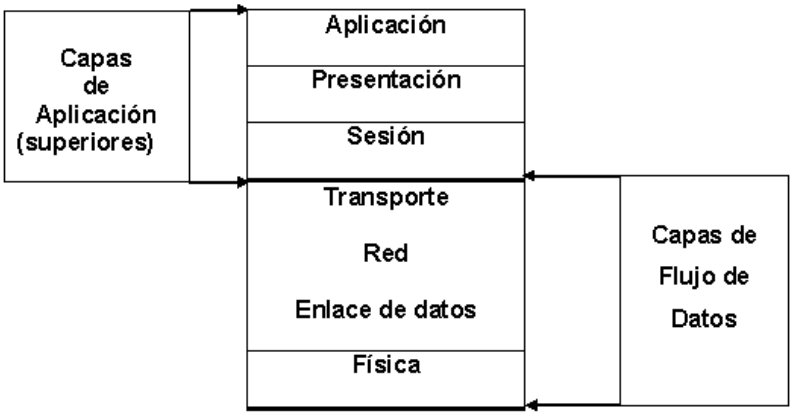


Figura 2.2 Capas de Aplicación y Capas de flujo de datos del Modelo OSI.

A continuación se muestran dichas capas (las 3 superiores con la información de su función así como algunos ejemplos).(Ver figura 2.3)

Las 4 capas inferiores del modelo OSI definen cómo los datos se transfieren por el cableado físico a través de los dispositivos de internetworking hasta la estación de destino que se desea, y finalmente hasta la aplicación.

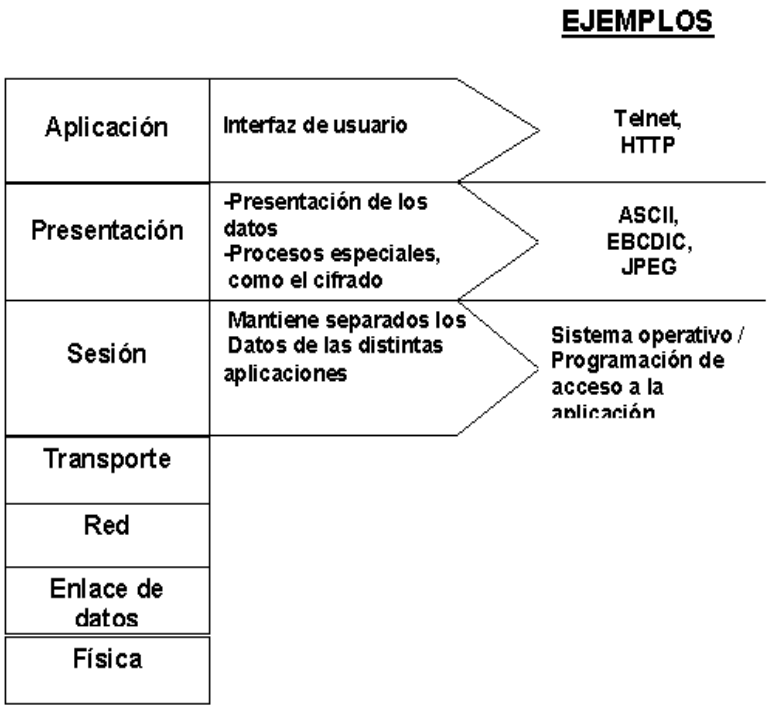


Figura 2.3 Ejemplos de las Capas Superiores del Modelo OSI.

2.3.1. Capa 7: La Capa de Aplicación

La capa de aplicación es la capa OSI más cercana al usuario. Proporciona los servicios de red como acceso e impresión de los archivos para las aplicaciones del usuario. Difiere de otras capas en que no proporciona servicio a ninguna otra capa OSI, sino sólo a las aplicaciones externas al modelo OSI. Ejemplos de dichas aplicaciones son los programas de hoja de cálculo, los procesadores de texto y los programas de las terminales bancarias. Sincroniza y establece un acuerdo con los procedimientos para la recuperación de errores e integridad en el control de datos. Para recordar a la capa hay que tener en mente a los navegadores.

2.3.2. Capa 6: La Capa de Presentación

La capa de presentación asegura que la información que se envía a la capa de aplicación de un sistema se pueda leer por la capa de aplicación de otro sistema. Si fuese necesario, la capa de aplicación traduce múltiples formatos de datos empleando un formato común. Esta capa es la responsable de la compresión y el cifrado.

Para recordar a la capa solo hay que recordar el formato de datos común.

2.3.3. Capa 5: La Capa de Sesión

Como su nombre implica, la capa de sesión establece, administra y finaliza las sesiones entre dos hosts. La capa de sesión proporciona su servicio a la capa de presentación. También sincroniza el diálogo entre las capas de presentación de los dos hosts y administra el intercambio de datos. Además de regular la sesión, la capa ofrece prevención para una eficiente transferencia de datos, clase de servicio y, excepcionalmente, informa de problemas en las capas de sesión, presentación y aplicación. Para recordar a la capa solo basta recordar los diálogos y conversaciones.

2.3.4. Capa 4: La Capa de Transporte

La capa de transporte segmenta los datos del sistema del host remitente y los reordena en un flujo de datos en el sistema del host receptor. El límite entre la capa de transporte y la capa de sesión puede imaginarse como el límite entre los protocolos de aplicación y los de flujo de datos. Mientras que las capas de aplicación, presentación y sesión se preocupan por los temas de la aplicación, las cuatro capas inferiores se preocupan por los temas del transporte de datos.

La capa de transporte intenta proporcionar un servicio de transporte de datos que proteja a las capas superiores de los detalles de la implementación del transporte. Específicamente, lograr un transporte fiable entre dos hosts ubicados en diferentes redes es el objetivo de la capa de transporte. Al proporcionar un servicio de comunicación, la capa de transporte establece, mantiene y finaliza adecuadamente los circuitos orientados a la conexión. Al suministrar un servicio fiable, se emplea la detección y recuperación de errores en el transporte y la información en el control del flujo. Para recordar la capa hay que tener en mente el control y la confiabilidad.

2.3.5. Capa 3: La Capa de Red

La capa de red es una capa compleja que proporciona conectividad y una selección de ruta entre dos sistemas host que pueden estar ubicados en redes geográficamente separadas. Si se quiere recordar a la capa, hay que acordarse de ruta, enrutamiento y direccionamiento lógico.

2.3.6. Capa 2: La Capa de Enlace de Datos

La capa de enlace de datos proporciona el tránsito de datos a través de un enlace físico. De este modo, la capa de enlace de datos se ocupa del enlace lógico entre nodos así como de su direccionamiento físico (lo contrario a lo lógico), el cual está directamente relacionado con la topología de la red (a veces se llama lógica), del acceso del medio a la red y de la detección de errores.

Para recordar la capa necesitamos tener en mente las tramas y el control del acceso al medio.

2.3.7. Capa 1: La Capa Física

La capa física define las especificaciones eléctricas, mecánicas, procedimentales y funcionales para activar, mantener y desactivar el enlace físico entre sistemas finales. Características como niveles de voltaje, cronometraje de los cambios de voltaje, velocidad de los datos físicos, distancia máximas de transmisión, conectores físicos y otros atributos similares, se definen mediante las especificaciones de la capa física. Es fácil recordar esta capa teniendo en cuenta los conceptos de las señales y medios.

2.4. Encapsulamiento

Todas las comunicaciones de la red se crean en un origen y se envían a un destino. La información que se envía por la red se denomina datos o paquetes de datos. Si una computadora (Host A) quiere enviar datos a otra computadora (Host B), los datos deben empaquetarse primero, este proceso es llamado encapsulación. La

encapsulación envuelve los datos con la información del protocolo necesaria antes de su tránsito por la red.

Por tanto, mientras el paquete de datos baja por las capas del modelo OSI, recibe las cabeceras, la información final y otra información.

Para ver cómo se produce la encapsulación, se debe examinar el modo en que los datos viajan a través de las capas (Ver figura 2.4)

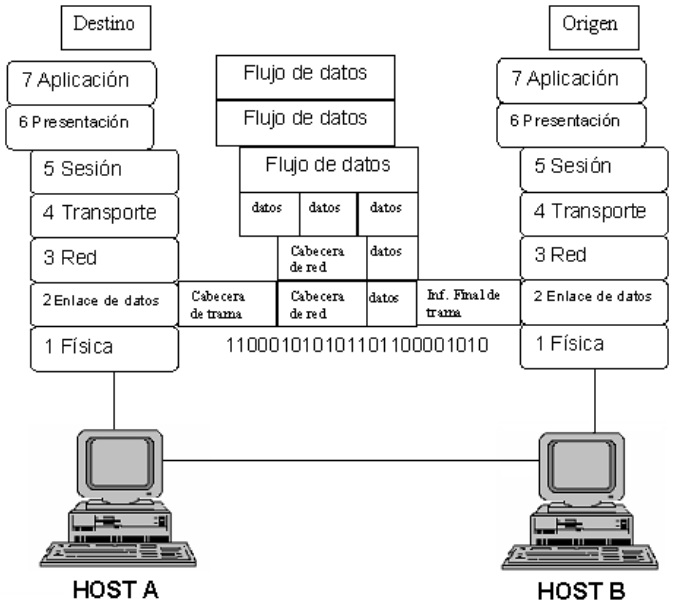


Figura 2.4 La capa inferior utiliza la encapsulación para poner la PDU de la capa superior en su campo de datos, y para añadir cabeceras e información final.

Cuando los datos se envían desde el origen, como se muestra en la figura 2.4, lo hacen a través de la aplicación y de otras capas. Como se observa, el empaquetamiento y el flujo de datos se intercambian mientras la red realiza su servicio para los usuarios finales.

Cada capa va agregando su PDU (unidad de datos de protocolo) hasta convertirse en señales eléctricas, las cuales deben viajar por el medio, sea cable o aire hasta la computadora de destino y, después, convertirse en su formato original para que puedan ser leídos por el destinatario. Como se puede imaginar, este proceso se realiza mediante varios pasos. Por este motivo, los diseñadores de hardware,

software y protocolos han reconocido que la manera más eficiente de implementar las comunicaciones de red sería un proceso de capas.

Las redes deben realizar los siguientes pasos de conversión para encapsular datos :(Ver figura 2.5)

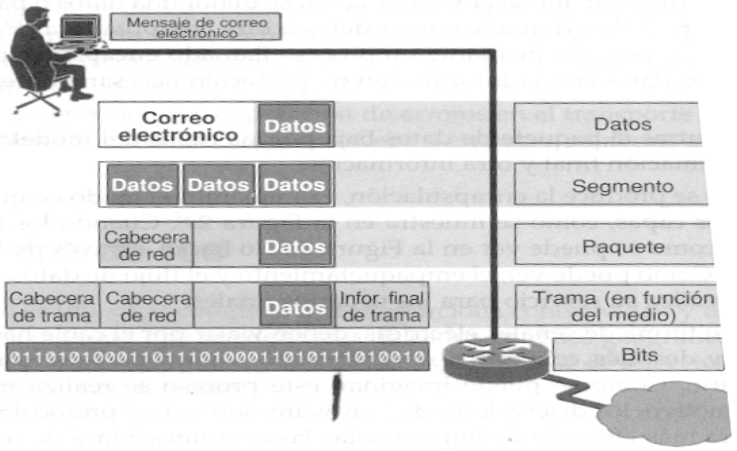


Figura 2.5 Las cabeceras y la información final se añaden mientras los datos se mueven a través de las capas del modelo OSI.

1) **Construir los datos:** Cuando un usuario envía un correo electrónico, sus caracteres alfanuméricos se convierten en datos (pdu de capas 7, 6 y 5) para que puedan viajar por la *internetwork*. (Red).
2) **Empaquetar los datos para el transporte de extremo a extremo**. Los datos se empaquetan para el transporte por la internetwork de redes. Usando segmentos (pdu de la capa 4), la función de transporte asegura que los hosts de ambos extremos del sistema de correo electrónico pueden comunicarse con total fiabilidad.
3) **Añadir la dirección de red a la cabecera.** Los datos se colocan en un paquete (pdu de la capa 3) o datagrama, que contiene una cabecera de red con direcciones lógicas de origen y destino. Dichas direcciones ayudan a los dispositivos de red a enviar paquetes a través de la red a lo largo de una ruta seleccionada dinámicamente.
4) **Agregar la dirección local a la cabecera de enlace de datos.** Cada dispositivo de red debe colocar el paquete en una trama (pdu de la capa2). La trama incluye una cabecera

con la dirección física del siguiente dispositivo conectado directamente en la misma ruta.

5) **Convertir los bits para la transmisión.** La trama se debe convertir en un modelo de 1 y 0 (bits – pdu de la capa 1) para su transmisión por el medio. Una función de cronometraje permite a los dispositivos distinguir a los bits mientras viajan por el medio. El medio de la internetwork física puede variar a lo largo de la ruta que se haya empleado. Por ejemplo, un correo electrónico puede originarse en una LAN, cruzar el backbone (Una línea de alta velocidad o una serie de conexiones que forman un mayor ancho de banda en una red) ó cable principal de un campus y salir a un enlace WAN hasta que alcance su destino en otra LAN remota.

2.5. Desencapsulamiento

Cuando el dispositivo remoto recibe una secuencia de bits, lo pasa por la capa de enlace de datos, recibe la trama y hace lo siguiente:

- Lee la dirección física y otras informaciones de control que proporciona la capa de enlace de datos conectada directamente.
- Interpreta la información de control de la trama, creando un datagrama. (Datagrama es una entidad de datos autocontenida e independiente que transporta información suficiente para ser encaminada desde su computadora de origen a su computadora de destino sin tener que depender de que se haya producido anteriormente tráfico alguno entre ambos y la red de transporte.)
- Pasa el datagrama a la siguiente capa, siguiendo las instrucciones que aparecen en la zona de control de la trama.

Este proceso se conoce como desencapsulamiento. Cada capa subsiguiente realiza un proceso de desencapsulamiento similar.

2.6. Nombres de los Datos en cada Capa del Modelo OSI

Para que los paquetes de datos viajen desde el origen hasta el destino, cada capa del modelo OSI del origen debe comunicarse con la misma capa del destino. Esta forma de comunicación se llama *comunicación de igual a igual.* Durante este proceso, cada protocolo de capa intercambia información, llamada **unidad de datos del protocolo** (pdu), entre las capas iguales. Cada capa de comunicación de la computadora origen se comunica mediante una PDU específica de cada capa, con su capa igual de la computadora destino.

Los paquetes de datos de una red se crean en el origen, y después viajan hasta el destino. Cada capa depende de la función del servicio de la capa OSI que esté debajo. Para proporcionar este servicio, la capa superior utiliza la encapsulación para colocar la PDU de la capa superior en su campo de datos, y añade las cabeceras e información final que la capa necesite para realizar su función. Después, mientras los datos se mueven entre las capas del modelo OSI, se añaden las cabeceras y la información final. Cuando las capas 7, 6 y 5 han añadido su información, la capa 4 añade más información. Esta agrupación de datos, la PDU de la capa 4, se llama **segmento.**

La capa de red, por ejemplo, proporciona un servicio a la capa de transporte, y la capa de transporte presenta los datos a la capa de red, que tiene la tarea de mover los datos por la internetwork. Realiza esta tarea encapsulando los datos y añadiendo una cabecera para crear un **paquete** o un **datagrama** (la PDU de la capa3). La cabecera contiene la información necesaria para completar la transferencia, como las direcciones lógicas de origen y destino.

La capa de enlace de datos proporciona un servicio a la capa de red. Encapsula la información de la capa de red en una **trama** (pdu de la capa 2); la cabecera de la trama contiene información (por ejemplo, las direcciones físicas) que se necesita para completar las funciones de enlace de datos. Asimismo, la información de la cabecera de enlace de datos es local, y solo es útil para los dispositivos conectados directamente.

La capa física también proporciona un servicio a la capa de enlace de datos. La capa física codifica la trama de enlace de datos en

un patrón de 1´s y 0´s (bits) para la transmisión por el medio (normalmente un cable) en la capa 1. Los bits se transmiten después al siguiente dispositivo conectado en la ruta de extremo a extremo.

2.7. Modelo TCP/IP

Tomando al modelo OSI (Open Systems Interconection) como referencia podemos afirmar que para cada capa o nivel que él define existen uno o más protocolos interactuando. Los protocolos son entre pares (peer-to-peer), es decir, un protocolo de algún nivel dialoga con el protocolo del mismo nivel en la computadora remota.

En 1973, la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA), de los Estados Unidos, inició un programa para la investigación de tecnologías que permitieran la transmisión de paquetes de información entre redes de diferentes tipos y características. El proyecto tenía por objetivo la interconexión de redes, por lo que se le denominó “Internetting”, y a la familia de redes de computadoras que surgieron de esta investigación se le denominó “Internet”. Los protocolos desarrollados se denominaron el Conjunto de Protocolos TCP/IP, que surgieron de dos conjuntos previamente desarrollados; los Protocolos de Control de Transmisión (Transmition Control Protocol) e Internet (Internet Protocol).

En la actualidad, las funciones propias de una red de computadoras pueden ser divididas en las siete capas propuestas por ISO para su modelo de sistemas abiertos (OSI). Sin embargo la implantación real de una arquitectura puede diferir de este modelo. Las arquitecturas basadas en TCP/IP proponen cuatro capas en las que las funciones de las capas de Sesión y Presentación son responsabilidad de la capa de Aplicación y las capas de Liga de Datos y Física son vistas como la capa de Interfase a la Red. Por tal motivo para TCP/IP sólo existen las capas Interfase de Red, la de Intercomunicación en Red, la de Transporte y la de Aplicación. Como puede verse TCP/IP presupone independencia del medio físico de comunicación, sin embargo existen estándares bien definidos para los niveles de Liga de Datos y Físico que proveen mecanismos de acceso a los diferentes medios y que en el modelo TCP/IP debe considerar la capa de Interfase de Red; siendo los más usuales el proyecto

IEEE802, Ethernet, Token Ring y FDDI. La relación del modelo TCP/IP con el modelo OSI esquemáticamente es el siguiente(Ver figura 2.6):

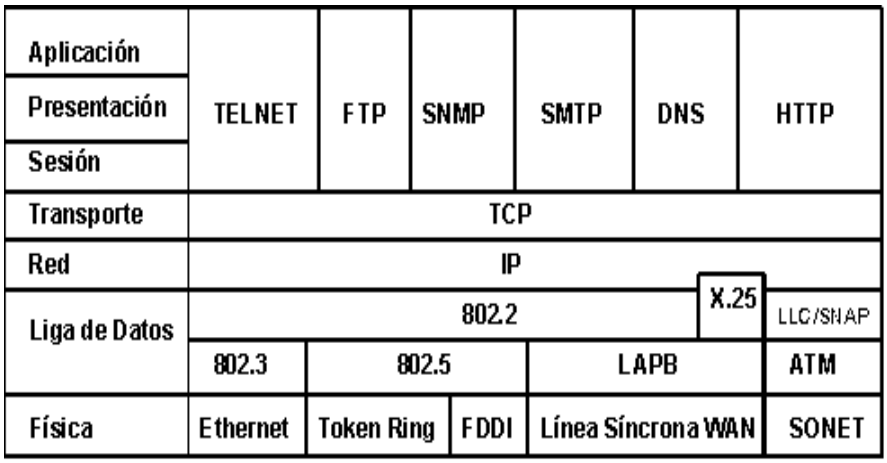


Figura 2.6 Relación esquemática del modelo TCP/IP con el modelo OSI

2.7.1. Descripción del Modelo de Capas de TCP/IP

El modelo de capas de TCP/IP es el siguiente:(Ver figura 2.7) y se muestra la analogía con el modelo OSI.

Capa de Aplicación: Invoca programas que acceden servicios en la red. Interactúan con uno o más protocolos de transporte para enviar o recibir datos, en forma de mensajes o bien en forma de flujos de bytes.

Capa de Transporte: Provee comunicación extremo a extremo desde un programa de aplicación a otro. Regula el flujo de información. Puede proveer un transporte confiable asegurándose que los datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente de tal manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota, esto lo hace añadiendo identificadores de cada una de las aplicaciones. Realiza además una verificación por suma, para asegurar que la información no sufrió alteraciones durante su transmisión.

Capa Internet: Controla la comunicación entre un equipo y otro, decide qué rutas deben seguir los paquetes de información para

alcanzar su destino. Conforma los paquetes IP que serán enviados por la capa inferior. Desencapsula los paquetes recibidos pasando a la capa superior la información dirigida a una aplicación.

Capa de Interfase de Red: Emite al medio físico los flujos de bit y recibe los que de él provienen. Consiste en los manejadores de los dispositivos que se conectan al medio de transmisión.

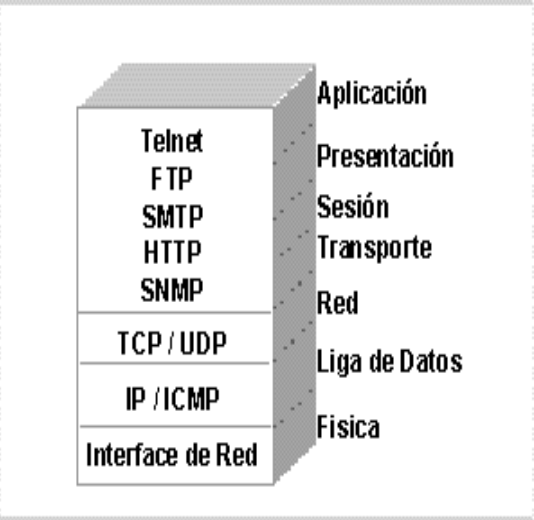


Figura 2.7 Muestra la analogía del Modelo TCP/IP (lado izquierdo) con el Modelo OSI (lado derecho)

Para entender el funcionamiento de los protocolos TCP/IP debe tenerse en cuenta la arquitectura que ellos proponen para comunicar redes. Tal arquitectura ve como iguales a todas las redes a conectarse, sin tomar en cuenta el tamaño de ellas, ya sean locales o de cobertura amplia.

Para que en una red dos computadoras puedan comunicarse entre sí ellas deben estar identificadas con precisión. Este identificador puede estar definido en niveles bajos (identificador físico) o en niveles altos (identificador lógico) dependiendo del protocolo utilizado. TCP/IP utiliza un identificador denominado dirección Internet o dirección IP, cuya longitud es de 32 bits. La dirección IP identifica tanto a la red a la que pertenece una computadora como a ella misma dentro de dicha red.

2.8. Modelo SNA Arquitectura de Sistemas de Red

Es una enorme y compleja arquitectura de red, con características, desarrollada en los 70's por IBM. Es similar en algunos aspectos al modelo de referencia OSI pero con una gran cantidad de diferencias: SNA está en esencia compuesto de siete capas:

- *Capa de Control de Flujo de Datos.*
- *Capa de Control de Enlace de Datos.*
- *Capa de Control de Trayectoria.*
- *Capa de Control Física.*
- *Capa de Servicios de Presentación.*
- *Capa de Servicios de Transacción.*
- *Capa de control de Transmisión.*

Capítulo 3

Capa Física

Capítulo 3
Capa Física del Modelo OSI

3.1. Capa Física

La capa física es la capa que define las especificaciones eléctricas, mecánicas, de procedimiento y funcionales para activar, mantener y desactivar los enlaces físicos entre sistemas externos.

La función de la capa física es transmitir los datos. Los datos, se representan por la presencia de pulsos eléctricos, a los que suele llamarse voltaje, en cables conductores de cobre o pulsos de luz dentro de la fibra óptica. Este proceso de transmisión; se realiza utilizando cables y conectores. Estos dispositivos de la Capa 1 ayudan a comprender los medios de una red.

Los *medios de transmisión* son los elementos utilizados para el transporte de la información entre dos o más nodos.Cuando hablamos de los medios de una red nos estamos refiriendo a los distintos entornos físicos a través de los que pasa la señal de transmisión. Para que las computadoras intercambien información codificada entre sí, el medio de red debe conectarlas físicamente, así tenemos dos clases principales de medios de comunicación para una red, los terrestres y los aéreos. (Ver figura 3.1)

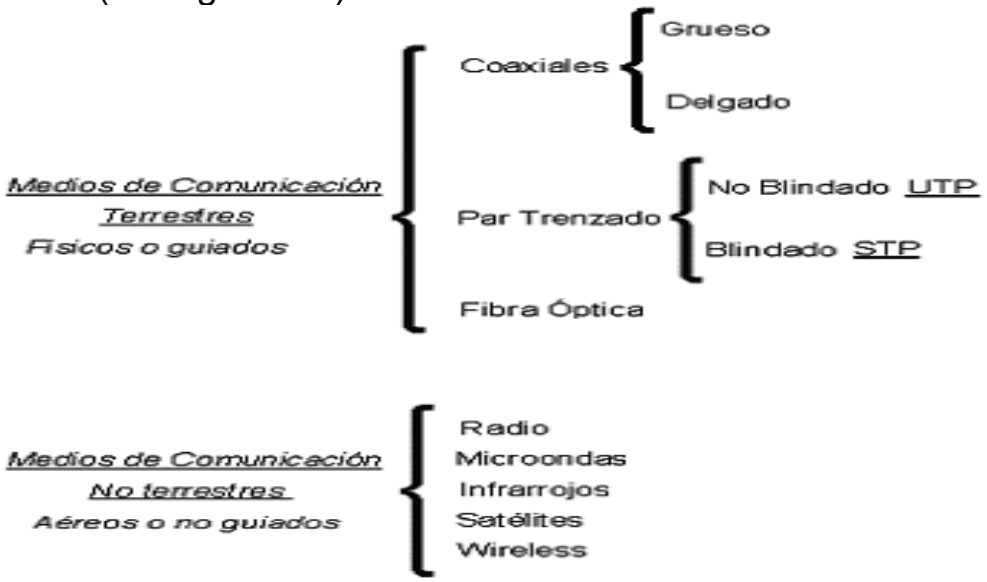


Figura 3.1. Medios de Comunicación de datos en las redes

3.2. Medios de Transmisión Terrestre

Dentro de los medios de transmisión terrestres se pueden emplear los siguientes tipos de cables para conectar computadoras:

- Cable coaxial (grueso y delgado),
- Par trenzado,
- Fibra Óptica.

3.2.1. Cable Coaxial

El *cable coaxial* está compuesto por un conductor cilíndrico externo hueco que rodea un solo alambre interno compuesto de dos elementos conductores. Uno de estos elementos (ubicado en el centro del cable) es un conductor de cobre. Está rodeado por una capa de aislamiento flexible. Sobre este material aislador hay una malla de cobre tejida o una hoja metálica que actúa como segundo alambre del circuito, y como blindaje del conductor interno. Esta segunda capa, o blindaje, ayuda a reducir la cantidad de interferencia externa. Este blindaje está recubierto por la envoltura del cable. (Ver figura 3.2)

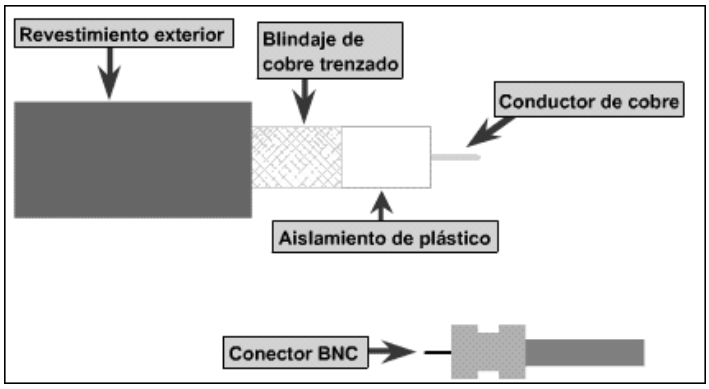


Figura 3.2 Cable Coaxial

Se utiliza en redes de comunicación de banda ancha y cables de banda base. El cable coaxial no se ve habitualmente afectado por interferencias externas, y es capaz de lograr altas velocidades de transmisión en largas distancias. Para las LAN, el cable coaxial ofrece varias ventajas.

Existen dos tipos principales de cable coaxial: el coaxial delgado o de banda angosta y el coaxial grueso o de banda ancha.

El cable coaxial grueso es un estándar en los sistemas de televisión con una impedancia clásica de 75 ohms. Se utiliza para señales analógicas con frecuencia modulada y en ciertas señales digitales de alta velocidad. Se especificó para su uso como cable de backbone de Ethernet (cableado principal) porque históricamente siempre poseyó mejores características de longitud de transmisión y limitación del ruido. Debido a su diámetro este tipo de cable puede ser demasiado rígido como para poder instalarse con facilidad en algunas situaciones. El cable coaxial delgado con un diámetro externo de solamente 0.35 cm (a veces denominado *thin* o de banda angosta). Se utiliza únicamente para señales digitales y regularmente su impedancia es de 50 ohms, además de requerir regularmente de codificación tipo Manchester alcanzando velocidades de transmisión de 10 Mbps.

Algunas características de este tipo de cables de muestran en la tabla 3.1

Características	Coaxial Grueso. RG-8	Coaxial Delgado. RG-58
Distancia máxima de segmentos	500 m	185 m
Ancho de Banda	10 Mbps	10 Mbps
Nodos por segmento	100	30
Impedancia	50 ohms	50 ohms
Diámetro	0.50 in	0.25 in
Conector	Transceiver (llamados Vampiros)	“T” y “BNC’s”
Usos	Transmisión Analógica y Cableado Principal (Backbone)	Transmisión Digital (Datos)

Tabla 3.1 Características de los cables coaxiales

3.2.2. Par Trenzado

En los cables de par trenzado, el trenzado de los pares individuales minimiza la interferencia electromagnética entre éstos y los cables en un par, tienen un espesor promedio de 0.016-0.036 in;

así mismo, el par puede utilizarse para transmitir señales analógicas y digitales.

El cable de par trenzado blindado (Con malla, STP –Shielded Twisted Pair) que se utiliza en conexiones de redes y estaciones de trabajo combina las técnicas de blindaje, cancelación y trenzado de cables. Según las especificaciones de uso de las instalaciones de red Ethernet, STP proporciona resistencia contra la interferencia electromagnética y de la radiofrecuencia sin aumentar significativamente el peso o tamaño del cable. El cable de par trenzado blindado tiene las mismas ventajas y desventajas que el cable de par trenzado no blindado. STP brinda mayor protección contra todos los tipos de interferencia externa, pero es más caro que el cable de par trenzado no blindado. (Ver figura 3.3)

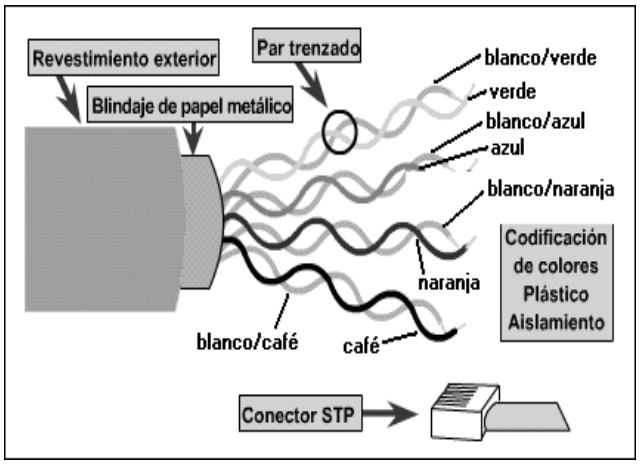


Figura 3.3 Par Trenzado STP

Por último, no es posible realizar tendidos de cable STP tan largos como con otros medios de interconexión de red (como, por ejemplo, cable coaxial) sin repetir la señal.

El cable de par trenzado no blindado (Sin malla; UTP Unshielded Twisted Pair) que se utiliza en las líneas telefónicas y protege muy poco o casi nada de las interferencias, es un medio compuesto por cuatro pares de hilos, que se usa en diversos tipos de redes. Cada par de hilos se encuentra aislado de los demás. Este tipo de cable se basa sólo en el efecto de cancelación que producen los pares trenzados de hilos para limitar la degradación de la señal que causan las *EMI (Electromagnetic Interferences o Interferencia*

Electromagnética). Éstas se definen como cualquier perturbación electromagnética que se manifieste en la degradación de la operación, el mal funcionamiento o la falla de un equipo eléctrico o electrónico y las RFI (Interferencias de radiofrecuencia). Para reducir aún más la diafonía entre los pares en el cable UTP, la cantidad de trenzados en los pares de hilos varía. Al igual que el cable STP, el cable UTP debe seguir especificaciones precisas con respecto a cuanto trenzado se permite por unidad de longitud del cable. (Ver figura 3.4).

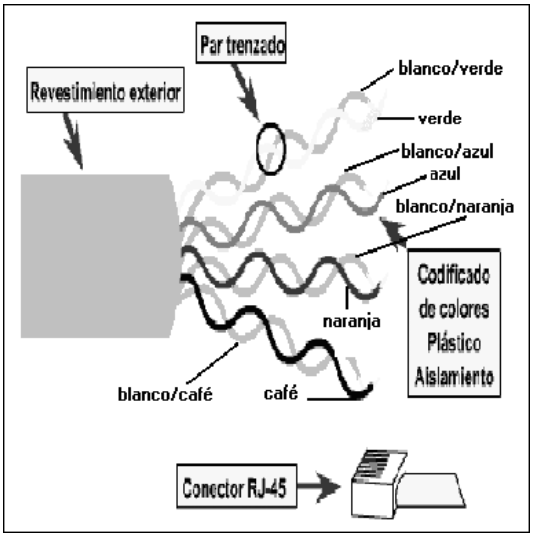


Figura 3.4 Par Trenzado UTP

Cuando se usa como medio de interconexión de redes el cable UTP tiene cuatro pares de hilos de cobre de calibre 22 ó 24. El UTP que se usa como medio de interconexión de red tiene una impedancia de 100 ohms. Esto lo diferencia de los otros tipos de cables de par trenzado, por ejemplo, los que se utilizan para los teléfonos. Como el UTP tiene un diámetro externo de aproximadamente 0.43 cm, el hecho de que su tamaño sea pequeño puede ser ventajoso durante la instalación. Como el UTP se puede usar con la mayoría de las arquitecturas de red principales, su popularidad va en aumento.

El cable de par trenzado no blindado presenta muchas ventajas. Es de fácil instalación y es más económico que los demás tipos de medios de interconexión de red. De hecho, el cable UTP cuesta menos por metro que cualquier otro tipo de cableado de LAN, sin embargo, la ventaja real es su tamaño. Como su diámetro externo es tan pequeño, el cable UTP no llena los conductos para el cableado tan

rápidamente como sucede con otros tipos de cables. Éste puede ser un factor sumamente importante para tener en cuenta, en especial si se está instalando una red en un edificio antiguo. Además, si se está instalando el cable UTP con un conector RJ-45, las fuentes potenciales de ruido de la red se reducen enormemente y prácticamente se garantiza una conexión sólida y de buena calidad.

Los conectores y jacks de uso común para el cable UTP son los RJ45. El Conector es una pieza de plástico transparente en donde se inserta el cable. El jack es también de plástico, pero en este se inserta el conector. Las siglas RJ significan Registro de Jack y el 45 especifica el esquema de numeración de pins. El cable se inserta en el conector, este se conecta al jack que puede estar en la pared, en la tarjeta de red de la computadora o en el concentrador.

Los conectores de 8 posiciones están numerados del 1 al 8, de izquierda a derecha cuando el conector es visto desde la parte posterior al gancho (la parte plana de los contactos). (Ver figura 3.5)

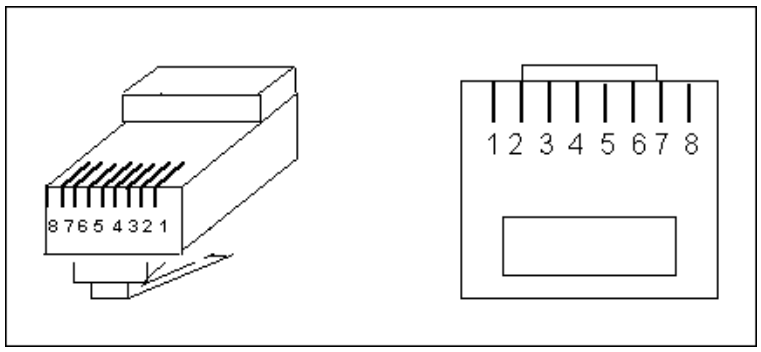


Fig.3.5 Conector RJ-45

Sin embargo, el cableado de par trenzado también tiene una serie de desventajas. El cable UTP es más sensible al ruido eléctrico y la interferencia que otros tipos de medios de interconexión de red. En la actualidad, se considera que el cable UTP es uno de los más rápidos entre los medios basados en cobre. La distancia entre los refuerzos de la señal es menor para UTP que para el cable coaxial.

El cable UTP se clasifica según su categoría. Este cable UTP permite la transmisión de grandes volúmenes de información. Estas propiedades están dadas por varios factores: el cobre con que está fabricado el conductor, el material de recubrimiento, tanto de cada

conductor como del cable total y finalmente el trenzado de cada par. Estas características hacen que el cable no requiera de blindaje para mantener la señal limpia y estable.

Las características generales del cable no blindado son:

- *Tamaño:* El menor diámetro de los cables de par trenzado no blindado permite aprovechar más eficientemente las canalizaciones y los armarios de distribución.
- *Peso:* El poco peso de este tipo de cable con respecto a los otros tipos de cable facilita el tendido.
- *Flexibilidad*: La facilidad para curvar y doblar este tipo de cables permite un tendido más rápido así como el conexionado de las rosetas y las regletas. El radio de doblado del cable no debe ser menor a cuatro veces el diámetro del cable. Para par trenzado de cuatro pares categoría 5 el radio mínimo de doblado es de 2.5 cm. Cabe mencionar que también existen categorías 6 y 7 del cable no blindado y que más adelante se mencionaran.
- *Impedancia característica:* La impedancia característica es igual a 100 ohms + 15% desde 1 Mhz hasta la frecuencia más elevada referida (16, 20 ó 100 Mhz). De una categoría particular.
- *Instalación*: Debido a la amplia difusión de este tipo de cables, existen una gran variedad de suministradores, instaladores y herramientas que abaratan la instalación y puesta en marcha.
- *Integración*: Los servicios soportados por este tipo de cable incluyen:
  - Red de Área Local ISO 802.3 (Ethernet) e ISO 802.5 (Token Ring),
  - Telefonía analógica,
  - Telefonía digital,
  - Terminales síncronos,
  - Terminales asíncronos,
  - Líneas de control y alarmas.

3.2.2.1. Categorías del Cable UTP

Una categoría de cableado es un conjunto de parámetros de transmisión que garantizan un ancho de banda determinado en un canal de comunicaciones de cable de par trenzado. Las categorías de cableado definen el estándar que debe cumplirse en la construcción extremo a extremo (end to end) del cableado estructurado.

A continuación se explicarán las categorías del cable UTP:

3.2.2.1.1. Categoría 1

La primera categoría responde al cable UTP, especialmente diseñado para redes telefónicas, el clásico cable empleado en teléfonos y dentro de las compañías telefónicas.

3.2.2.1.2. Categoría 2

El cable es también empleado para transmisión de voz y datos hasta 4Mbps.

3.2.2.1.3. Categoría 3

Define los parámetros de transmisión hasta 16 MHz. Entre las principales aplicaciones de los cables de categoría 3 encontramos: voz, Ethernet 10Base-T y Token Ring. El largo máximo que puede alcanzar un tramo sin considerar los cables de Panel de Parcheo (patch cord) según la norma dictada por el EIA/TIA es de 90 metros, o sea, 90 metros desde el punto de la pared al patch panel o panel de control, más 10 metros para conexiones.

3.2.2.1.4. Categoría 4

Tiene la capacidad de soportar comunicaciones en redes de computadoras a velocidades de 20Mbps. El largo máximo al igual y bajo las mismas premisas que la categoría 3 es de 90 metros.

3.2.2.1.5. Categoría 5

Es usado en redes LAN, con la capacidad de sostener comunicaciones a 100Mbps.

La categoría 5 define los parámetros de transmisión hasta 100 MHz. Inicialmente, la categoría 5 sólo definía atenuación y NEXT como parámetros importantes en la medición de las características del canal. A raíz de los trabajos en Gigabit Ethernet se agregaron nuevos parámetros a la definición de esta categoría puesto que había que garantizar una transmisión por los cuatro pares de manera simultánea en ambas direcciones (full dúplex).

Entre las principales aplicaciones de los cables de categoría 5 encontramos: voz, Ethernet 10Base-T, Token Ring, 100VG AnyLan, Fast Ethernet 100Base-TX, ATM 155 Mbps, ATM 622 Mbps y Gigabit Ethernet. (Ver tabla 3.2)

Parámetro de transmisión	Valor para el canal a 100 MHz
Longitud	90mts.
Atenuación	24 dB
NEXT	27.1 dB
Wiremap	N.A.

Tabla 3.2. Valores límites para las pruebas de certificación en categoría 5

El cable UTP para redes actualmente usado es el de 8 hilos categoría 5, es decir cuatro pares trenzados formando una sola unidad. Estos cuatro pares vienen recubiertos por un tubo plástico que mantiene el grupo unido, mejorando la resistencia ante interferencias externas. Es importante notar que cada uno de los cuatro pares tiene un color diferente, pero a su vez, cada par tiene un cable de un color específico y otro blanco con algunas franjas del color de su par. Esta disposición de los cables permite una adecuada y fácil identificación de los mismos con el objeto de proceder a su instalación. El número identificación de cada par referente a su color se muestra a continuación.(Ver figura 3.6)

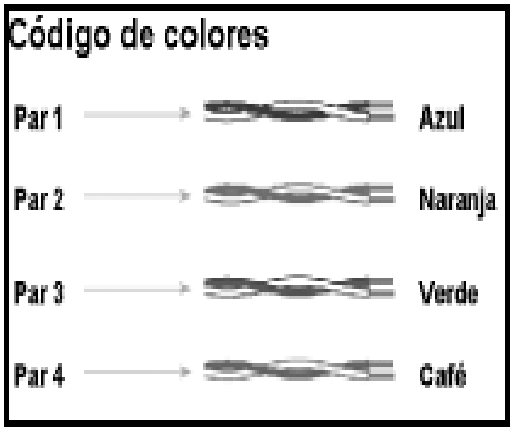


Figura 3.6 Código de colores para el cable UTP.

3.2.2.1.5.1. Categoría 5 Mejorada (5e)

La categoría 5 enhance define los parámetros de transmisión hasta 100 MHz. La diferencia fundamental con la categoría 5 normal es el agregar nuevas pruebas de certificación a manera de asegurar el soporte directo de la tecnología Gigabit Ethernet.

Entre las principales aplicaciones de los cables de categoría 5 mejorada encontramos: voz, Ethernet 10Base-T, Token Ring, 100VG AnyLan, Fast Ethernet 100Base-TX, ATM 155 Mbps, ATM 622 Mbps y Gigabit Ethernet. Estos valores fueron publicados como la adición 5 de la norma TIA/EIA-568A. (Ver tabla 3.3)

Parámetro de transmisión	Valor para el canal a 100 MHz
Atenuación	24.0 dB
NEXT	30.1 dB
PSNEXT	27.1 dB
ACR	6.1 dB
PSACR	3.1 dB
ELFEXT	17.4 dB
PSELFEXT	14.4 dB
Return Loss	10.0 dB
Delay	548 n.s.
Delay Skew	50 n.s.

Tabla 3.3. Valores límites para las pruebas de certificación en categoría 5e

Será el cable de categoría 5e el que se utilizará en el laboratorio.

3.2.2.1.6. Categoría 6

La categoría 6 fue liberada en el mes de junio de 2002 y define como pruebas de certificación las mismas que la categoría 5e pero siendo más estricta en sus valores límites, además por una petición de la IEEE las pruebas se aumentaron de 200Mhz que era la tasa de transmisión original a 250Mhz.

La categoría es tan estricta en sus pruebas que aún no hay soluciones para la construcción personal de cables de panel de parcheo los cuales sólo pueden ser fabricados en laboratorios (fábricas) especializadas.

La longitud máxima que puede alcanzar un tramo sin considerar los cables de Panel de Parcheo según la norma dictada por el EIA/TIA es de 90 metros más 10 metros para conexiones.

3.2.2.1.7. Categoría 7

La categoría 7 utiliza en su instalación cable par trenzado blindado, esta categoría no es utilizada en Sur y Norteamérica sino más bien en Europa.

Beneficios de la tecnología de categoría 7:

Las normas pendientes para componentes de categoría 7, enlaces y canales de Clase F ofrecerán un ancho de banda sobre cableado de par trenzado todavía no superado. Una solución del Sistema 7 de Siemon SM que emplea el conector TERA™ proporciona resultados excelentes hasta 1 GHz, muy por encima del límite de frecuencia superior de 600 MHz de las especificaciones actuales en borrador. Esta capacidad permite aplicaciones de demanda como vídeo de banda ancha, con un requisito superior de frecuencia de 862 MHz, para operar sobre un cableado de Clase F con conexiones simultáneas a otras aplicaciones de redes. Por ejemplo, un canal único de Clase F puede soportar conexiones a diversas aplicaciones tales

como voz análoga (1 par), una LAN de alta velocidad (2 pares) y vídeo de banda ancha (1 par).

Esta tecnología soporta todas las aplicaciones diseñadas para operar sobre cableado de par trenzado, así como aquellas que, de otro modo, podrían requerir fibra o coaxial. La inmunidad al ruido y el rendimiento de emisiones de la Clase F soportan aplicaciones avanzadas de LAN sin necesidad de una electrónica costosa para implementar una codificación compleja o procesamiento de señales.

Las normas pendientes de categoría 7 y clase F le permitirán a la instalación de cable blindado desarrollar todo su potencial en términos de ancho de banda, versatilidad y comodidad de uso. Los cables de la Categoría 7 serán "completamente blindados", con pares trenzados individualmente apantallados y un blindaje general. Este tipo de cable predomina en varios países Europeos. Sin embargo, su aceptación global se ha visto negativamente afectada por los componentes de conexión que son limitados en términos de rendimiento, comodidad de uso, adaptabilidad y tamaño. Una interfaz de Categoría 7 que diseñó específicamente TERA™ proporciona en forma consistente márgenes de alto rendimiento y respuesta de frecuencia lineal hasta por lo menos 1 GHz, con tiempo mínimo de terminación y en el mismo espacio que un conector de estilo modular (o "RJ") de ocho contactos.

Diferencias entre tecnología de categoría 6 y tecnología de categoría 7:

La categoría 6/clase E proporciona el nivel más alto de rendimiento de transmisión disponible sin pares individualmente apantallados. Para la inmensa mayoría de aplicaciones de negocios e institucionales, un ancho de banda de 250 MHz es más que apropiado para la vida útil del sistema de cableado, haciendo de la categoría 6/clase E la elección perfecta para cableado genérico de localidades.

El objetivo de la categoría 7/clase F es ser tan buena o mejor que cualquier otro tipo de medios balanceados para cada parámetro de transmisión. Por ejemplo, el canal proporcionará una ACR positiva de suma de potencia de hasta por lo menos 600 MHz.

Existen diferencias físicas significativas entre los componentes de categoría 6 y 7. La construcción "*completamente blindada*" del cable de categoría 7 tiene como resultado un diámetro exterior mayor y menos flexibilidad que el UTP o el ScTP. Estos atributos requieren un mayor cuidado en el diseño de trayectorias y espacios de terminación para dejar más espacio y mayores radios de curvatura.

Otra diferencia radica en los accesorios de conexión. La especificación pendiente de la categoría 7 exige que los conectores proporcionen por lo menos 60 dB de aislamiento de diafonía entre todos los pares a 600 MHz. Este requisito es 32 dB más estricto que la categoría 5 a 100 MHz, y 20 dB más estricto que la categoría 6 a 250 MHz.

Tecnología innovadora:

Independizarse de la interfaz modular de ocho contactos o de estilo "RJ" allanó el camino a muchas de las innovaciones de TERA™, como su rendimiento sin paralelo y la modularidad de 4, 2 y 1 par. La libertad de las restricciones del conector "RJ" le permitió a Siemon crear una nueva interfaz con aislamiento par-a-par, compactación, confiabilidad y comodidad de instalación como principales objetivos.

Además de su rendimiento, comodidad de terminación, tamaño pequeño y compatibilidad con los componentes actuales de montaje, el TERA™ incluye otras características intencionalmente previstas para brindar flexibilidad y aceptación por parte del cliente. La toma TERA™ está en capacidad de acoplarse con una clavija única de 4 pares o una combinación de clavijas de 2 pares y 1 par para soportar conexiones múltiples de equipos en cada extremo. Son apilables lateralmente, lo cual permite la misma densidad que el UTP o el ScTP, y puede encajar a presión desde la parte delantera o posterior de la abertura del tablero.

Se proveen también las mismas características de que se dispone con nuestros otros diseños de tomas, tales como un tabulador de íconos con código de colores y puerta. La clavija presenta un diseño libre de rebabas, codificación de colores, así como un mecanismo de aldaba robusto e intuitivo.

Una herramienta innovadora de preparación del cable y un sistema de unión contribuyen además a una aceptación positiva de parte del instalador del TERA™ y el Sistema 7.

Posición de Siemon en el desarrollo del cable de Categoría 7:

La Compañía Siemon ha asumido un papel de liderazgo en el desarrollo de la categoría 7 y se encuentra comprometido en forma activa en la normalización de la categoría 7/clase F de ISO/IEC.

3.2.3. Fibra Óptica

El *cable de fibra óptica* es un medio de interconexión de redes que puede conducir transmisiones de luz moduladas. Si se compara con otros medios de interconexión de red, es más caro, sin embargo, no es susceptible a la interferencia electromagnética y ofrece velocidades de datos más altas que cualquiera de los demás tipos de medios de interconexión de red. El cable de fibra óptica no transporta impulsos eléctricos, como lo hacen otros tipos de medios. En cambio, las señales que representan a los bits se convierten en haces de luz.

El cable de fibra óptica que se usa en interconexión de redes está compuesto por dos fibras envueltas en revestimientos separados. Si se observa una sección transversal de este cable, veremos que cada fibra óptica se encuentra rodeada por capas de material amortiguador protector, normalmente un material plástico como Kevlar, y un revestimiento externo. El revestimiento exterior protege a todo el cable. Generalmente es de plástico y cumple con los códigos aplicables de incendio y construcción. El propósito del Kevlar es brindar una mayor amortiguación y protección para las frágiles fibras de vidrio que tienen el diámetro de un cabello. Siempre que los códigos requieran que los cables de fibra óptica deban estar bajo tierra, a veces se incluye un alambre de acero inoxidable como refuerzo. (Ver figura 3.7)

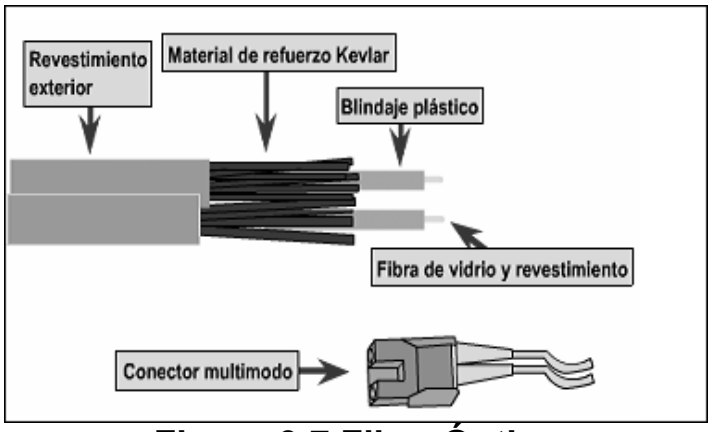


Figura 3.7 Fibra Óptica

Las partes que guían la luz en una fibra óptica se denominan *núcleo* y *revestimiento*. El núcleo es generalmente un vidrio de alta pureza con un alto índice de *refracción.* Cuando el vidrio del núcleo está recubierto por una capa de revestimiento de vidrio o de plástico con un índice de refracción bajo, la luz se captura en el núcleo de la fibra. Este proceso se denomina *reflexión interna total* y permite que la fibra óptica actúe como un "tubo de luz", guiando la luz a través de enormes distancias, incluso dando vuelta en codos.

Tipos de fibra óptica:

- ***Monomodo.*** Cuando la luz se propaga en línea recta, sin rebotar.
- ***Multimodo.*** Cuando un rayo de luz se refleja internamente y existe una gran cantidad de rayos rebotando a diferentes ángulos.

Características:

- Las fibras ópticas no se ven afectadas por alteraciones de voltaje o corriente, por interferencia electromagnética o por químicos corrosivos dispersos en el aire. En la actualidad los sistemas de fibra óptica son capaces de hacer transmisiones de datos de 1 Gbps en 1 Km.
- Especificación de la IEEE 10BaseFL: La máxima distancia entre los segmentos es de 2000 m.
- *Ventajas:* Posible transmisión de voz, datos y video por el mismo canal. Aplicaciones de alta velocidad. Es inmune a

interferencias del exterior pues no genera señales eléctricas o magnéticas. Puede propagar una señal sin necesidad de utilizar un amplificador en distancias de hasta 2000 m. Excelente tolerancia a factores ambientales.

3.3. Medios de Transmisión Aéreos

Las señales aéreas son ondas electromagnéticas, que pueden recorrer el vacío del espacio exterior y medios como el aire. Por lo tanto, no es necesario un medio físico para las señales inalámbricas, lo que hace que sean un medio muy versátil para el desarrollo de redes.

3.3.1. Radio

Son sistemas basados en la propagación de ondas electromagnéticas, rara vez forman parte de un sistema de comunicaciones de datos para computadora.

Este tipo de comunicaciones de datos se utiliza ocasionalmente debido a que la tasa de errores de datos es muy alta y se requieren medios complejos de detección y corrección de errores.

Otro inconveniente de este medio es que requiere de antenas muy grandes, además los rayos solares y las tormentas eléctricas introducen gran cantidad de ruido que a veces bloquean completamente la transmisión.

3.3.2. Microondas

Son ondas electromagnéticas transmitidas en el vacío por antenas especiales cuya forma es de plato y las cuales deben estar a la vista por lo que su transmisión se ve afectada por obstáculos como: bosques, edificios, mal tiempo, etc.

Esta transmisión se logra a través de la atmósfera entre torres de microondas, generalmente espaciadas a 40 y hasta 80 Km. entre sí.

Cada torre toma la señal transmitida de la torre anterior, la amplifica y retransmite a la siguiente torre de microondas. Pueden portar millares de canales a la vez y no requieren que se tienda cable.

Elementos de las microondas:

- Espacio aéreo como medio físico de transmisión,
- Antenas visibles de tipo plato,
- Terminal del usuario,
- Circuitos que interconectan la antena con la terminal del usuario.
- Las microondas permiten la transmisión de datos en redes de procesamiento de datos tal como:
  - Redes entre ciudades usando la red telefónica,
  - Redes cuyo canal de comunicación es privado y comprende un área metropolitana,
  - Redes de largo alcance por medio de satélites,
  - Antenas emisoras o repetidoras fijas,
  - Antenas emisoras o repetidoras móviles (barcos).

Desventajas:

- La instalación implica una gran dificultad técnica así como la utilización de cables en algunas conexiones.
- Las antenas deben contar con línea de vista, de lo contrario no se puede transmitir.

3.3.3. Infrarrojo

Las redes basadas en rayo infrarrojo operan usando un rayo de luz infrarroja el cual transporta los datos entre los dispositivos.

Este sistema trabaja con señales de una fuerte intensidad ya que una transmisión con señales de baja intensidad es susceptible a la luz como la que entra por las ventanas.

Este método puede transmitir señales a una alta velocidad ya que la luz infrarroja tiene un amplio ancho de banda. Una red que ocupa esta tecnología puede transmitir a 10 Mbps.

3.3.4. Enlace Vía Satélite

Es una forma especial de transmisión de relevo microondas.

Un satélite es un repetidor activo de señal en el espacio el cual filtra, amplifica y transmite una señal de tipo analógica a digital. Un satélite emplea antenas microondas para recibir señales de radio y las estaciones emisoras de la Tierra para devolver estas señales a otras estaciones terrenas.

Ventajas:

- Poseen una enorme capacidad de transmisión, cada satélite es capaz de soportar varios miles de canales telefónicos.
- Proporcionan una amplia cobertura territorial.

Desventajas:

- Las condiciones climatológicas tales como tormentas eléctricas pueden afectar a la señal durante su camino por los canales de subida y bajada.
- Plantean problemas de seguridad, ya que cualquier estación es capaz de captar cualquier transmisión con solo sintonizar la frecuencia del satélite, para evitarlo muchas empresas utilizan la codificación como medida de seguridad.
- Una señal recorre una gran distancia, unos 36,000 Km. de ida y otros tantos de vuelta.
- Las señales del satélite pueden verse interferidas también por otras señales de radio procedentes de sistemas terrestres.

3.3.5. Redes Inalámbricas

3.3.5.1. Comunicación en Redes Inalámbricas

El entorno inalámbrico

A medida que la tecnología avanza las redes inalámbricas empiezan a ganar terreno poco a poco, esto es las redes inalámbricas

comienzan a ganar popularidad gracias a la flexibilidad con la que es posible implementar nodos de comunicación donde el cableado por cuestiones técnicas no es posible realizarlo.

Capacidades Inalámbricas

Utilizables cuando:

- La conexión del equipo es temporal
- Agregar seguridad a una red ya instalada
- Cierto grado de portabilidad
- Extensión de la red mas allá del limite del cable

Aplicaciones de los sistemas inalámbricos

- Zonas con tránsito peatonal
- Áreas y edificios aislados
- Los puestos de trabajo se mueven constantemente
- Cuando es imposible cablear debido a la construcción

3.3.5.2. Tecnología Inalámbrica

Las redes inalámbricas son redes cuyos medios físicos no son cables de cobre de ningún tipo. Están basadas en la transmisión de datos mediante ondas de radio, microondas, satélites o infrarrojos.

Las redes inalámbricas de área local (WLAN) utilizan la tecnología de Radio Frecuencia (RF) para transmitir y recibir datos a través del aire, brindando características y beneficios de la tecnologías LAN tradicionales pero sin las limitaciones del cable. Una WLAN es un sistema flexible de comunicaciones de datos, implementado como una extensión de una red cableada o como una alternativa a la red cableada. Una WLAN es un sistema de comunicación de datos entre dos o más computadoras que transmite sin utilizar cables de por medio.

La tecnología wireless permite complementar el acceso a las redes de datos y dar soluciones alternativas al intercambio de información, los datos son enviados de un dispositivo a otro por medio del aire con ondas electromagnéticas.

La IEEE ha desarrollado varios estándares para las redes LAN inalámbricas y los ha clasificado en la familia de la especificación 802.11. Esta especificación clasifica a las redes wireless que funciona con tecnología de radiofrecuencia en bandas de 2.4 GHz y 5GHz. Las velocidades para este estándar van desde 1 y 2 Mbps hasta 54 Mbps y éstas van variando según el estándar utilizado. Los estándares más importantes para el mercado actual son el 802.11a, 802.11b y 802.11g (Ver tabla 3.4)

	802.11a	802.11b	802.11g
Ratificación de Estándar	2002	1999	2003
Frecuencia portadora	5.8 GHz	2.4 GHz	2.4 GHz
Velocidad Máxima de datos	54Mbps	11 Mbps	54 Mbps
Precio	Alto	Económico	Accesible
Área de Cobertura	50-80 metros	50-100 metros	50 – 125 metros
Popularidad	Relativamente nuevo	Amplia	Nuevo
Compatibilidad	No compatible	Comúnmente usado con 802.11g	Compatible con 802.11b

Tabla 3.4. Los estándares 802.11 a, 802.11b y 802.11g.

Ventajas:

- Principalmente la movilidad que ofrecen sin ningún cable de por medio.
- El ahorro de dinero para la instalación, comparándola con una red cableada.
- La rapidez y sencillez para la instalación y el mantenimiento.
- Se pueden trasladar a cualquier parte junto con el equipo móvil sin alterar la configuración y volverla totalmente portátil.

Desventajas:

- Una de las desventajas más importantes es que la tasa máxima de transferencia hasta el momento es de 54Mbps, cuando en una red cableada puede alcanzar los 1000Mbps.
- Se puede robar la señal de la red estando dentro de la misma área de cobertura.
- Falta mejorar la seguridad en las transferencias y cifrados de datos, para impedir que hackers accedan a los datos de la red.
- La velocidad especificada no es la real a la que indican las normas ya que a esto influye factores como la calidad del ambiente, los tipos de materiales utilizados en la construcción, las interferencias por las lluvias y con otro tipo de dispositivos inalámbricos e incluso hornos de microondas.

3.4. *Estándares de la Capa Física: RS-232,RS-422,RS-449*

3.4.1. Definición

Cuando dos equipos (*DTE*) (*Equipo Terminal de Datos*) están cercanos entre sí y no usamos tasas de bits muy altas, podremos transmitir datos mediante líneas abiertas de dos alambres y circuitos de interfaz sencillos que conviertan los niveles de señal empleados dentro del equipo a un nivel compatible con el cable de interconexión. Pero conforme aumenta la separación física entre los DTE y se incrementa la tasa de bits, tendremos que usar circuitos y técnicas más avanzadas. Además, si los DTE están en diferentes puntos del país (o del mundo) y no se cuenta con recursos de comunicación de datos públicos, la única estrategia que ofrece efectividad de costos será usar las líneas que proporcionan las autoridades encargadas de la prestación de servicios telefónicos y de otros servicios de telecomunicaciones. Estas últimas se conocen como PTT. Con este tipo de medio de comunicación, usualmente es necesario convertir las señales eléctricas que produce el DTE de origen a una forma análoga a las señales empleadas para transmitir la voz. De manera similar, al recibir estas señales, es preciso convertirlas a una forma con la que el DTE de destino pueda trabajar. El equipo que realiza estas funciones se denomina *módem.* (Ver figura 3.8)



Figura 3.8 Módem

3.4.2. Módems

El módem es otro de los periféricos que con el tiempo se ha convertido ya en imprescindible y pocos son los modelos de computadoras que no estén conectados en red, que no lo incorporen. Su gran utilización viene dada básicamente por dos motivos: Internet y el fax, aunque también le podemos dar otros usos como son su utilización como contestador automático incluso con funciones de centralita o para conectarnos con la red local de nuestra oficina o con la central de nuestra empresa.

Aún en el caso de estar conectado a una red, ésta tampoco se libra de éstos dispositivos, ya que en este caso será la propia red la que utilizará el módem para poder conectarse a otras redes o a Internet estando en este caso conectado a nuestro servidor o a un router.

Lo primero que hay que dejar claro es que los módem se utilizan con líneas analógicas, ya que su propio nombre indica su principal función, que es la de modular-demodular la señal digital proveniente de nuestro ordenador y convertirla a una forma de onda que sea asimilable por dicho tipo de líneas.

Es cierto que se suelen oír expresiones como módem ADSL o incluso módem RDSI, aunque esto no es cierto en estos casos, ya que estas líneas de tipo digital no necesitan de ningún tipo de conversión de digital a analógico, y su función en este caso es más parecida a la de una tarjeta de red que a la de un módem.

Uno de los primeros parámetros que lo definen es su velocidad. El estándar más habitual y el más moderno está basado en la actual norma V.90 cuya velocidad máxima está en los 56 Kbps (Kilobits por segundo). Esta norma se caracteriza por un funcionamiento asimétrico, puesto que la mayor velocidad sólo es alcanzable "en bajada", ya que en el envío de datos está limitada a 33,6 Kbps.

Evidentemente, el módem que se encuentre al otro lado de la línea telefónica, sea nuestro proveedor de Internet o el de nuestra oficina debe ser capaz de trabajar a la misma velocidad y con la misma norma que el nuestro, ya que de lo contrario la velocidad que se establecerá será la máxima que aquel soporte.

Otras normas habitualmente utilizadas son (ver tabla 3.5):

Norma	Velocidad máxima	Otras velocidades
V.90 y X2*	56.000 bps	57.333, 54.666, 53.333, 52.000, 50.666, 49.333, 48.000, 46.666, 45.333, 44.000, 42.666, 41.333, 40.000, 38.666, 37.333, 36.000, 34.666 bps
V.34+	33.600 bps	31.200 bps
V.34	28.800 bps	26.400, 24.000, 21.600, 19.200, 16.800 bps
V.32bis	14.400 bps	12.000 bps
V.32	9.600 bps	7.200 bps
V.23	4.800 bps	
V.22bis	2.400 bps	
V.22 y Bell 212A	1.200 bps	
V.21 y Bell 103	300 bps	

Tabla 3.5. Normas utilizadas para el módem

*protocolo propietario de 3Com.

Otra funcionalidad ya considerada como obligatoria en cualquier módem es el soporte de funciones de FAX. Lo estándares son los siguientes (Ver tabla 3.6):

Norma	Velocidad máxima	Otras velocidades
V.17	14.400 bps	12.000 bps
V.29	9.600 bps	7.200 bps
V.27ter	4.800 bps	2.400 bps
V.21	300 bps	

Tabla 3.6. Estándares para el soporte de funciones de FAX en el módem.

Otros estándares considerados como imprescindibles son los de control de errores y compresión de datos. Los más habituales son: V.42, V.42bis y MNP 2-5.

No podemos dejar de comentar otros aspectos igualmente importantes como el de contar con una memoria de tipo flash que nos permita la actualización del firmware al igual que ocurre con las BIOS de las placas base.

Los módems pueden ser externos (un dispositivo de comunicación) o interno (dispositivo de comunicación interno o tarjeta de circuitos que se inserta en una de las ranuras de expansión de la computadora). (Ver figuras: 3.9, 3.10, 3.11 y 3.12 que ejemplifican algunos tipos módem)

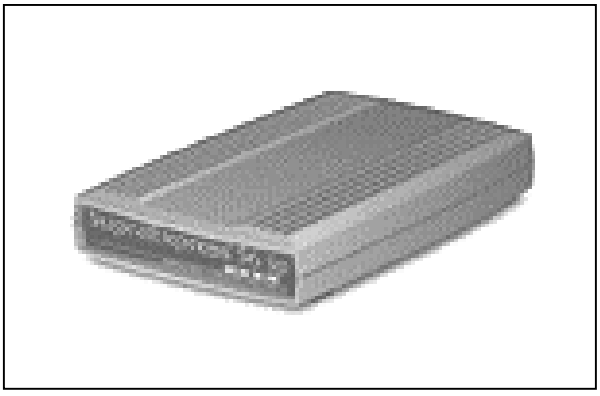


Figura 3.9. Módem externos para puerto serie.

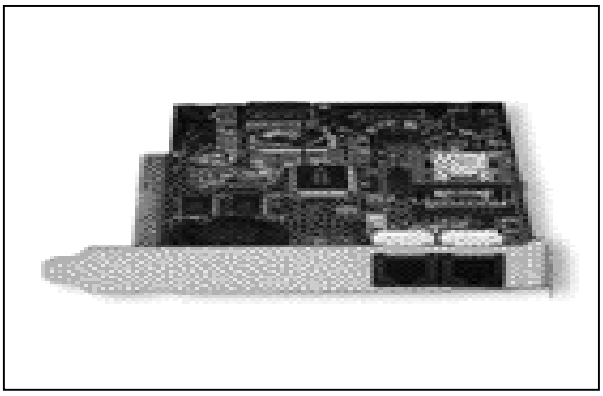


Figura 3.10. Módem internos

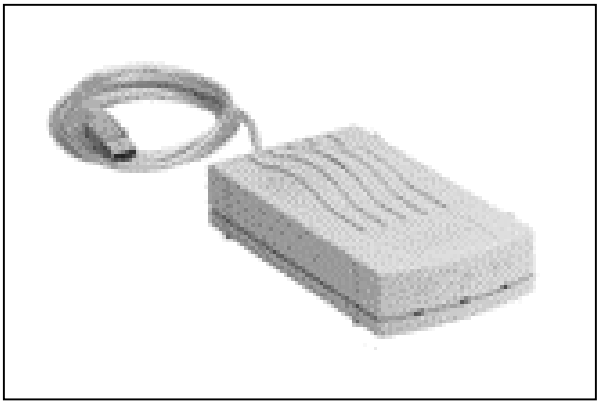


Figura 3.11. Módem externos para puerto USB

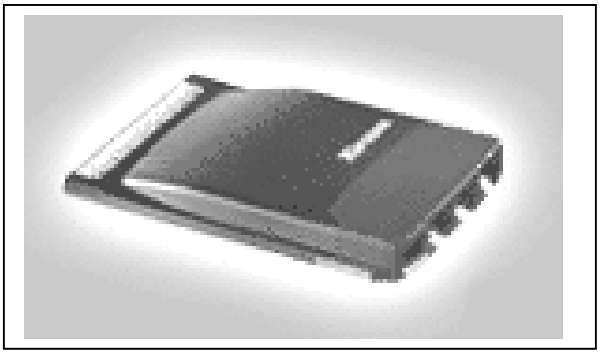


Figura 3.12 Módem PC-Card (PCMCIA)

3.4.3. RS-232,RS-422,RS-449

La interfaz entre la computadora o terminal y el módem es un ejemplo de protocolo de capa física, que debe especificar con detalle la interfaz mecánica, eléctrica, funcional y de procedimientos.

3.4.3.1. Estándar de Capa Física RS-232

Existe una amplia gama de módems para que los fabricantes de diferentes tipos de equipos puedan usar los recursos de transmisión

de la red telefónica conmutada. Estos módems trabajan con diferentes tasas de transmisión y se valen de diversos métodos de modulación. Hay muchos fabricantes de módems. Para garantizar que dos módems de diferentes fabricantes se puedan comunicar entre sí, se ha estandarizado el funcionamiento preciso de cada tipo de módem. Se ha definido una interfaz estándar con que podemos enlazar todos los tipos de módems y especificar tanto el número como la función de señales eléctricas, el tamaño físico del conector y el enchufe, y las configuraciones de contactos dentro de ellos.

A esta norma se le conoce como interfaz EIA-232-D (definida por la EIA Electronic Industries Association) o interfaz V.24 (definida por la ITU-T), y es la versión más reciente de las normas RS-232A, B y C, la primera de las cuales apareció a finales de la década de 1950.

En los estándares, la terminal o computadora se llama de manera oficial DTE (data terminal equipment, equipo terminal de datos), y el módem, DCE(data circuit-terminating equipment, equipo de comunicación de datos).

El estándar consta de varias especificaciones: *mecánica, eléctrica y funcional.*

3.4.3.2. Estándar de Capa Física RS-449 RS-422-A

El estándar llamado *RS-449*, en realidad es tres estándares en uno.

Las interfaces mecánica, eléctrica y de procedimientos se especifican en RS-449, pero la interfaz eléctrica se especifica en dos estándares diferentes. El primero de ellos, *RS-423-A,* es parecido al *RS-232-C* en que todos sus circuitos se comparten una tierra común. Esta técnica se llama *transmisión desbalanceada*. El segundo estándar eléctrico, *RS-422-A*, usa en contraste la *transmisión balanceada*, en la que cada uno de los circuitos principales requiere de dos hilos, sin tierra común. Esto permite usar la RS-422-A a velocidades de hasta 2 Mbps por cables de 60 metros.

3.4.3.3. Estándar de Capa Física RS-422-A/ V.11

Si incrementamos tanto la separación física como la tasa de bits, tendremos que optar por la definición de señal alternativa *RS-422-A/ V.11*, que se basa en el empleo de un cable de par trenzado y un par de circuitos transmisores y receptores diferenciales (también conocidos como balanceados o de extremo doble).

3.5. Cableado Estructurado

Un *sistema de cableado estructurado* (*SCE*) es un sistema de cableado que permite identificar, reubicar y cambiar en todo momento, fácilmente y de forma racional los diversos equipos que se conectan, basándose en una normativa completa de identificación de cables y de componentes, así como el empleo de cables y conectores de las mismas características para todos los equipos.

Un sistema de cableado estructurado es una forma ordenada y planeada de realizar cableados que permiten conectar teléfonos, equipo de procesamiento de datos, computadoras personales, conmutadores, redes de área local y equipo de oficina entre sí.

Dispositivos:

El cableado es normalmente el medio por el cual la información se mueve de un dispositivo de red a otro. Existen una serie de componentes que ayudan a organizar las grandes instalaciones UTP y facilitar su manejo.

Los armarios y racks de distribución ayudan a organizar una red que tiene muchas conexiones. Montando en ellos los dispositivos de red como routers, switchs, hubs, patch panels, etc. Los Paneles de Parcheo (Patch Panels) pueden admitir hasta 96 puertos de conexión.

En un cableado estructurado, la función principal de las canaletas es la de proteger el cableado. Las canaletas son conductos o tubos, normalmente de plástico, que rodean el cable; variando su diámetro de acuerdo a la cantidad y grosor de los cables que conducen y normalmente se colocan aprovechando las esquinas de las paredes de los edificios. Suelen ser más baratos que la ductería y

de acceso más fácil; pero están expuestos a las interferencias electromagnéticas (EMI).

Un jack o keystone RJ45 es un dispositivo modular de conexión mono-línea, hembra, apto para conectar plugs RJ45, que permite su inserción en rosetas y frentes de patch panels especiales mediante un sistema de encastre. Permite la colocación de la cantidad exacta de conexiones necesarias.

La roseta o faceplate es una pieza plástica de soporte que se fija a la pared y permite colocar hasta 2 jacks o keystones, formando así una roseta de hasta 2 entradas. Se conectan a los patch panels y placas de pared. Una de sus principales funciones es que pueden ser etiquetadas para facilitar el control de los puntos de conexión de nuestra red.

3.5.1. Estándar EIA/TIA 568

El estándar EIA/TIA 568 define un sistema genérico de alambrado de telecomunicaciones para edificios comerciales que pueden soportar un ambiente de productos y múltiples proveedores. El propósito de este estándar es permitir el diseño e instalación del cableado de telecomunicaciones contando con poca información acerca de los productos de telecomunicaciones que posteriormente se instalarán. La instalación de los sistemas de cableado durante el proceso de instalación y/o remodelación son significativamente más baratos e implican menos interrupciones que después de ser ocupado el edificio.

Ambos, el TIA (Telecomunications Industry Association) y el EIA (Electronic Industries Association) son miembros del ANSI (American National Standards Institute) el cual, es coordinado por grupos estándares voluntariados en Estados Unidos. El cableado de telecomunicaciones especificado en esta norma tiene por finalidad apoyar una gama muy amplia de diferentes edificios comerciales y de aplicaciones (por ejemplo voz, datos y video).

El Estándar EIA/TIA 568 lista los siguientes subsistemas de cableado estructurado**.**

I. *Subsistema Entrada del edificio.*
II. *Subsistema Cuarto de Equipo.*
III. *Subsistema Cableado horizontal.*
IV. *Subsistema Cuarto de telecomunicaciones*
V. *Subsistema Cableado vertical (Backbone).*
VI. *Subsistema del Área de trabajo.*

3.5.1.1. Subsistema Entrada del Edificio

Este subsistema consiste en una entrada de servicios de telecomunicaciones al edificio, la cual incluye el punto de entrada a través de la pared del edificio y continuando al cuarto o área de entrada. La entrada al edificio debe contener la ruta del backbone que interconecta con los otros edificios del campus; en caso de una comunicación a través de una antena, ésta también pertenece a la entrada del edificio.

3.5.1.2. Subsistema Cuarto de Equipo

El cuarto de equipo es un espacio centralizado para los equipos de telecomunicaciones (equipos de cómputo, hubs, switch, etc.) (Ver figura 3.13 Cuarto de Equipo), que sirven a los ocupantes del edificio. Este cuarto, únicamente debe guardar equipos directamente relacionados con el sistema de telecomunicaciones y sus sistemas de soporte. Los cuartos de equipo incluyen espacio de trabajo para el personal de telecomunicaciones. En la siguiente tabla 3.7 se muestra el tamaño recomendado de los armarios para el cableado.

Tamaño recomendado para el armario para cableado
(basado en 1 estación de trabajo por 10 metros cuadrados)

Área de servicio		Tamaño del armario para cableado	
$(m)^2$	$(ft)^2$	$(m)^2$	$(ft)^2$
1000	10000	3.0 x 3.4	10 x 11
800	8000	3.0 x 2.8	10 x 9
500	5000	3.0 x 2.2	10 x 7

Tabla 3.7 Tamaño de los armarios para el cableado.



Fig.3.13 Cuarto de Equipo

3.5.1.3. Subsistema Cableado Horizontal

El cableado horizontal es la porción del sistema de cableado que se extiende desde el clóset de telecomunicaciones (*rack*) hasta el usuario final (*hasta la roseta*).

La distancia horizontal máxima es de 90m. Independiente del cable utilizado. Esta es la distancia desde el área de trabajo hasta el cuarto de telecomunicaciones. La previsión máxima se hace de 10 m. adicionales para la distancia combinada de cables de empate (3 m.) y cables utilizados en conexión de equipos en el área de trabajo y el cuarto de telecomunicaciones. (Ver figura 3.14).

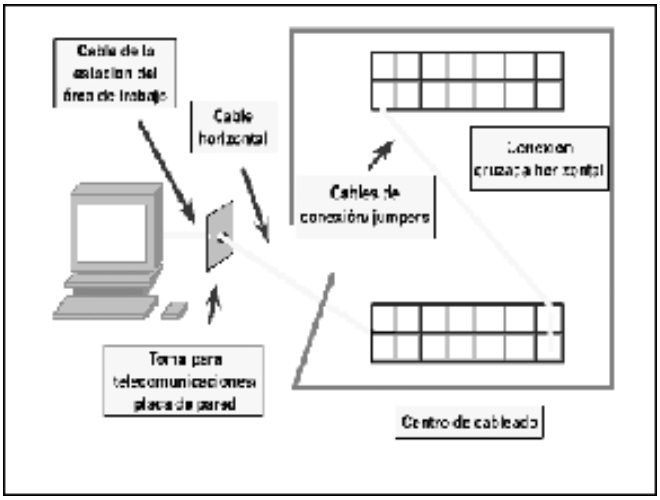


Fig.3.14 Componente del cableado horizontal TIA/EIA-568-A

3.5.1.4. Subsistema Cuarto de Telecomunicaciones

Un cuarto de telecomunicaciones es el área de trabajo en un edificio para el uso exclusivo de equipo asociado con el sistema de cableado de telecomunicaciones. El espacio del cuarto de comunicaciones no debe ser compartido con instalaciones eléctricas que no sean de telecomunicaciones. El cuarto debe ser capaz de albergar equipo de telecomunicaciones, terminaciones de cable y cableado de interconexión asociado.

3.5.1.5. Subsistema Cableado Vertical (Backbone)

El cableado vertical o de Backbone provee la interconexión entre el cuarto de equipos, cuarto de telecomunicaciones y la entrada al edificio.

El cableado vertical debe soportar todos los dispositivos que están dentro del rack y a menudo a todas las impresoras, terminales y servidores de archivos de un piso en un edificio. (Ver figura 3.15).

El cableado vertical consiste del cable backbone, de crossconnect intermedio y principal, de las terminaciones mecánicas y de los *patch cords.*

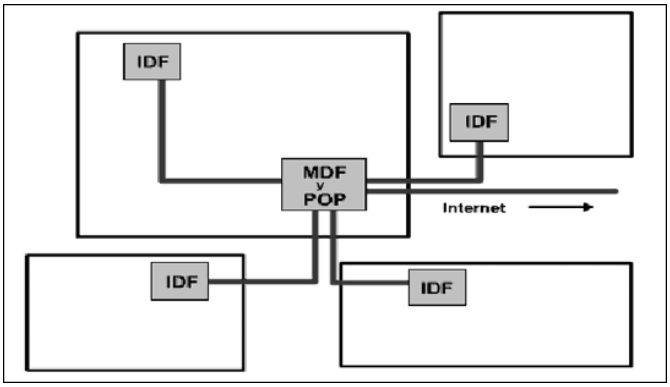


Figura 3.15 Cableado Vertical o del Backbone

3.5.1.6. Subsistema del Área de Trabajo

Comprende las inmediaciones físicas de trabajo habitual (mesa, silla, zona de movilidad, etc.) del o de los usuarios. El punto que marca

su comienzo en lo que se refiere a cableado es la roseta o punto de conexión.

En el ámbito del área de trabajo se encuentran diversos equipos activos del usuario como teléfonos, computadoras, impresoras, telefax, terminales, etc. (Ver figura 3.16.)

La naturaleza de los equipos activos condicionan el tipo de conectores existentes en las rosetas, mientras que el número de las mismas determinan si la roseta es: simple (1 conector), doble (2 conectores), triple (3 conectores), cuádruple (4 conectores), entre otros.

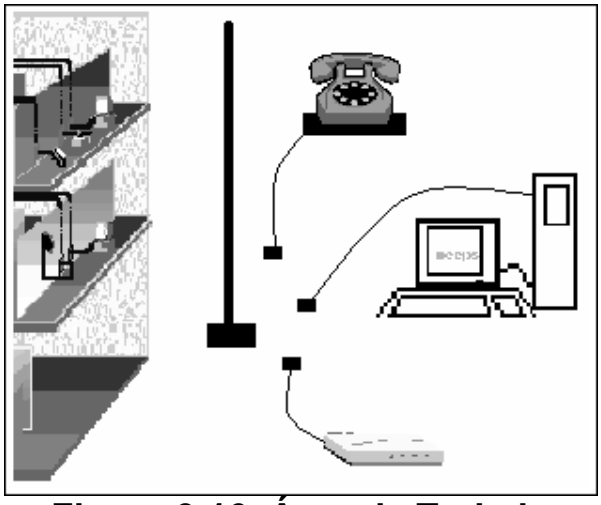


Figura 3.16. Área de Trabajo

3.5.2. Diagrama de Subsistemas

Existen otros estándares que junto con éste permiten contar con un cableado seguro y confiable entre los cuales están: (Ver tabla 3.8).

Estándar EIA/TIA 569	Estándar para ductos y espacios de telecomunicaciones en edificios comerciales.
Estándar EIA/TIA 606	Estándar de Administración para la infraestructura de telecomunicaciones de edificios comerciales.

Tabla 3.8 Estándares EIA/TIA 569 y EIA/TIA 606

3.6. Equipo

Los equipos utilizados para la capa Física del Modelo OSI son el repetidor y el hub.

3.6.1. Repetidores

Los repetidores son dispositivos de red que existen en la capa física, del modelo de referencia OSI.

El propósito del repetidor es regenerar y reenviar las señales de red a nivel de bits para hacer posible que éstas viajen largas distancias por los medios. Prolongan la longitud de la red uniendo dos segmentos y amplificando la señal, pero junto con ella amplifican también el ruido. La red sigue siendo una sola, con lo cual, siguen siendo válidas las limitaciones en cuanto al número de estaciones que pueden compartir el medio. (Ver figura 3.17 Repetidor).

Hay que tener cuidado con la regla de los cuatro repetidores para Ethernet de 10 Mbps, también conocida como la regla cinco segmentos de red de extremo a extremo usando cuatro repetidores, pero sólo tres segmentos podrán tener hosts. Para la regla de 5-4-3, solo se pueden conectar como límite 5 segmentos, con 4 repetidores y en donde solo en 3 pueden conectarse hosts.

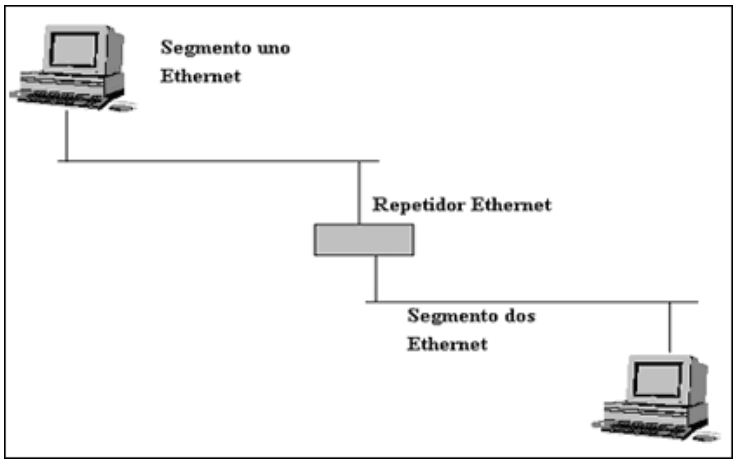


Figura 3.17 Repetidor

3.6.2. Hubs

En general, el término hub se emplea en lugar de repetidor cuando se refiere a un dispositivo que sirve como una central de la red. Aunque un hub opera en una topología física de estrella, crea el mismo entorno de conexión de un bus. Esto se debe a que cuando un dispositivo transmite, el resto de dispositivos le escuchan y la contención crea un bus lógico. El propósito de un hub es regenerar y reenviar señales de red. Esto se hace a nivel de bits con un gran número de hosts. Esta acción se conoce como concentración.

Las propiedades más importantes de los hubs son:

- Regenera y repite las señales
- Propaga las señales en la red
- No pueden filtrar el tráfico de la red
- No pueden determinar la mejor ruta
- Se utilizan como puntos de concentración de la red

En esencia, un hub es un repetidor multipuertos. Los hubs se emplean normalmente en las redes Ethernet 10BaseT o 100BaseTx. Dos motivos para emplear los hubs son el crear un punto de conexión central para la conexión eléctrica de los medios, y para aumentar la fiabilidad de la red. La fiabilidad de la red aumenta al permitir que cada cable pueda fallar sin afectar al resto de la red.

NOTA.- Para reafirmar algunos de los conceptos adquiridos en la teoría sobre la Capa Física del Modelo OSI; se recomiendan hacer las siguientes prácticas contenidas dentro de nuestra tesis:

PRÁCTICA 1. Construcción de cables UTP para conexión directa y cruzada

En donde el alumno al finalizar la práctica:

Tendrá la capacidad de construir cables de conexión directa y cruzada con normas ANSI/EIA/TIA T568-A y T568-B.

PRÁCTICA 2. Componentes del Cableado Estructurado Norma ANSI/EIA/TIA 568

En donde el alumno al finalizar la práctica:

Se familiarizará con aspectos generales del cableado estructurado mediante la instalación de un *jack* y roseta, utilizando cable UTP Cat.5e ó superior.

Capítulo 4

Capa de Enlace de Datos

Capítulo 4
Capa de Enlace de Datos

4.1. Capa de Enlace de Datos del Modelo OSI

La tarea principal de la **capa de enlace de datos** es tomar un medio de transmisión en bruto y transformarlo en una línea que parezca libre de errores de transmisión no detectados a la capa de red. Esta tarea la cumple al hacer que el emisor divida los datos de entrada en **marcos de datos** (unos cientos o miles de bytes, normalmente), que transmita los marcos en forma de secuencias y procese los **marcos de acuse de recibo** que devuelve el receptor. Puesto que la capa física solamente acepta y transmite una corriente de bits sin preocuparse por su significado o su estructura, corresponde a la capa de enlace de datos crear y reconocer los límites de los marcos. Esto se puede lograr añadiendo patrones especiales de bits al principio y al final del marco. Si estos patrones de bits ocurrieran en los datos por accidente, se debe tener cuidado especial para asegurar que estos patrones no se interpreten incorrectamente como delimitadores de marcos.

Una ráfaga de ruido en la línea puede destruir por completo un marco. En este caso, el *software* de la capa de enlace de datos de la máquina fuente puede retransmitir el marco. Sin embargo, las transmisiones repetidas del mismo marco introducen la posibilidad de duplicar marcos. Se podría enviar un marco duplicado si se perdiera el marco del acuse de recibo que el receptor devuelve al emisor. Corresponde a esta capa resolver el problema provocado por los marcos dañados, perdidos y duplicados. La capa de enlace de datos puede ofrecer varias clases de servicio distintas a la capa de la red, cada una con diferente calidad y precio.

Otra consideración que surge en la capa de enlace de datos (y también de la mayor parte de las capas más altas) es cómo evitar que un transmisor veloz sature de datos a un receptor lento. Se debe emplear algún mecanismo de regulación de tráfico para que el transmisor sepa cuánto espacio de almacenamiento temporal *(buffer)* tiene el receptor en ese momento. Con frecuencia esta regulación de flujo y el manejo de errores están integrados.

Si se puede usar la línea para transmitir, datos en ambas direcciones, esto introduce una nueva complicación que el *software* de la capa de enlace de datos debe considerar. El problema es que los marcos de acuse de recibo para el tráfico de A a B compiten por el uso de la línea con marcos de datos para el tráfico de *B* a A. Ya se inventó una solución inteligente (plataformas transportadoras).

Las redes de difusión tienen una consideración adicional en la capa de enlace de datos: cómo controlar el acceso al canal compartido. Una subcapa especial de la capa de enlace de datos se encarga de este problema, la subcapa de acceso al medio.

Proporciona el tránsito confiable de datos a través del enlace de red. Diferentes especificaciones de la capa de enlace de datos definen diferentes características de red y protocolo, incluyendo el direccionamiento físico, la topología de red, la notificación de error, la secuencia de tramas y el control de flujo. El direccionamiento físico (a diferencia del direccionamiento de red), define cómo se nombran los dispositivos en la capa de enlace de datos.

La topología de red consiste en especificaciones de la capa de enlace de datos, que con frecuencia definen la forma en que se conectarán físicamente los dispositivos, en topología bus o en topología anillo. La notificación de error alerta a los protocolos de las capas superiores cuando se presenta un error en la transmisión y la secuencia de tramas de datos reordena las que se han transmitido fuera de secuencia. Finalmente, el control de flujo regula la transmisión de datos para que el dispositivo receptor no se sature con más tráfico del que pueda manejar simultáneamente.

El **IEEE** *(Instituto de Ingenieros, en Electrónica y Electricidad)* ha subdividido la *capa de enlace de datos en dos subcapas:* **LLC** (*Control de Enlace Lógico*), y **MAC** (*Control de Acceso a Medios*). La figura 4.1 ilustra las subcapas IEEE de la capa de enlace de datos.

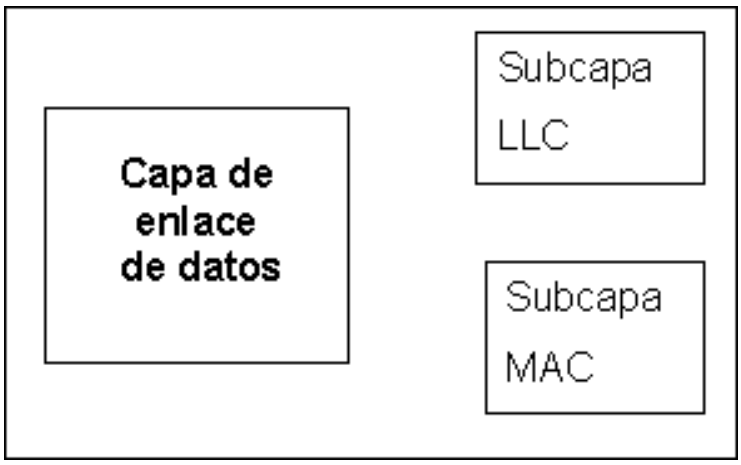


Figura 4.1. La capa de enlace de datos contiene dos subcapas.

*La **subcapa LLC** de la capa de enlace de datos administra las comunicaciones entre los dispositivos unidos por un enlace individual de red.* La subcapa LLC está definida en la especificación IEEE 802.2 y soporta los servicios orientados y no orientados a la conexión, utilizados por los protocolos de las capas superiores. El IEEE 802.2 define varios campos en las tramas de la capa de enlace de datos que permiten que varios protocolos de las capas superiores compartan un solo enlace físico de datos.

La **subcapa MAC** de la capa de enlace de datos *administra el protocolo de acceso al medio de transmisión físico de la red.* La especificación **IEEE MAC** define las *direcciones MAC*, las cuales permiten a múltiples dispositivos identificarse de manera única entre sí en la capa de enlace de datos.

4.2 Transmisión Síncrona y Asíncrona

4.2.1. Introducción al Modo de Transferencia

*El modo de transferencia asíncrono (**ATM**)* permitió la integración de voz, video y datos sobre una misma infraestructura de red.

A mediados de los años ochenta, se planteó la necesidad de integrar de una forma eficiente el soporte de aplicaciones tan diversas como la telefonía, la videoconferencia, la transferencia de datos, etc. sobre una misma infraestructura de red. Esta pretensión se materializó en la definición, por parte del CCITT, de la Red Digital de Servicios

Integrados de Banda Ancha (RDSI-BA). La tecnología de soporte de esta RDSI-BA fue propuesta a partir de la experiencia obtenida en el diseño y operación de las redes de conmutación de circuitos y de las redes de conmutación de paquetes. Finalmente en 1988 el CCITT definió las características del *Modo de Transferencia Asíncrono (Asynchronous Transfer Mode ATM)* y determinó que fuese el modo de transferencia para la RDSI-BA.

4.2.2 Red Digital de Servicios Integrados de Banda Ancha

A mediados de los años ochenta la realidad de la prestación de los servicios de telecomunicación era tal que cada uno de los servicios existentes era provisto mediante una red particular. De esta forma, para la prestación del servicio telefónico se debía diseñar, desplegar, operar y mantener una red telefónica conmutada y esto mismo era necesario para proveer el servicio de datos, para proveer el servicio de difusión de TV por cable y para proveer al incipiente servicio de videoconferencia.

Se imponía, por tanto, la necesidad de conseguir economías de escala para la provisión eficiente de los servicios de telecomunicación existentes y futuros. Ello debería traducirse en la consecución del soporte de todos los servicios de telecomunicación sobre una misma infraestructura de red. De este modo, sólo sería necesario el diseño, despliegue, operación y mantenimiento de una sola red.

Esta pretensión no era válida. Cada uno de los servicios identificados anteriormente muestran características de calidad de servicio y de ancho de banda necesario muy dispares. En la tabla 4.1 se muestra para cada uno de los servicios, si se trata de un servicio sin conexión u orientado a conexión, cuáles son los requisitos de retardo y de variabilidad de retardo, cuáles son los requisitos de ancho de banda y, finalmente, cuáles son las características del patrón de tráfico generado.

	Telefonía	Datos	TV por cable	Videoconferencia
Sin conexión		X	x	
Orientado a conexión	X	X		X
Retardo	Sensible	Insensible	Insensible	Sensible
Variabilidad de retardo	Sensible	Insensible	Sensible	Sensible
Ancho de banda	Bajo	Medio	Alto	Alto
Tasa de bit constante	X		x	x
Tasa de bit variable		x		x

Tabla 4.1. Requisitos de cada uno de los Servicios de telecomunicación.

Queda de manifiesto la diversidad de requisitos de cada uno de estos servicios de telecomunicación. Esta diversidad impuso un importante desafío a quienes pretendían encontrar un modo de transferencia que soportase la RDSI-BA.

Cuando se planteó la definición de la RDSI-BA, la normalización de la RDSI, que a partir de entonces pasó a denominarse RDSI-BE (Banda Estrecha), había alcanzado un grado de madurez aceptable. La pregunta que surge es: ¿por qué la RDSI-BE no satisfizo las necesidades que hicieron necesario el planteamiento de la RDSI-BA? Dos son las principales razones:

1. La RDSI-BE consiguió integrar distintos servicios de telecomunicación. Ahora bien, esta integración que se consiguió fue únicamente en el acceso: con la RDSI-BE se logró conseguir la prestación de diversos servicios de telecomunicación a través de un conjunto reducido de interfaces y de protocolos. Sin embargo, la infraestructura de red necesaria para la prestación de cada uno de ellos era específica del servicio. No se consiguió una economía de escala global.

2. La RDSI-BE fue diseñada para soportar accesos de hasta 2,048 Mbit/s. A mediados de los ochenta se preveía la aparición de nuevos servicios que estaban poco definidos aún, pero para los que se esperaba que necesitasen grandes anchos de banda, por encima de los 2 Mbit/s. Es en el escenario descrito hasta este punto, en el que se debe enmarcar la definición y normalización de ATM, que se describirá más adelante.

4.2.3 Los Modos de Transferencia

EL término Modo de Transferencia fue acuñado por el CCITT durante el proceso de definición de la tecnología de soporte de la RDSI-BA. El CCITT lo definió como: *"La técnica empleada en una red de telecomunicación y que cubre aspectos relacionados con la transmisión, el multiplexado y la conmutación".*

Al inicio del proceso de definición del modo de transferencia para la RDSI-BA, existían dos modos de transferencia en uso en las redes de telecomunicación: *el modo de transferencia síncrono y el modo de transferencia de paquete.*

4.2.4 El Modo Síncrono

El Modo de Transferencia Síncrono (Synchronous Transfer Mode, STM) era el empleado en la red telefónica conmutada digital y en la RDSI-BE, esto es, las redes digitales basadas en la conmutación de circuitos. Más específicamente el Modo Síncrono incluye la *técnica de multiplexado por división en el tiempo síncrono y la técnica de conmutación de circuitos.*

La técnica de multiplexado por división en el tiempo (*Time Division Multiplexing, TDM*) síncrono es la técnica utilizada en el *múltiplex MIC 30+2,* cuantiosamente empleado en telefonía digital.

4.3 Protocolos de Enlace

Los protocolos utilizados para determinar quien sigue en un canal multiacceso pertenecen a la subcapa MAC (Médium Access

control, control de acceso al medio). Esta subcapa tiene especial importancia en las LAN, casi todas las cuales usan un canal multiacceso como base de su comunicación.

4.4 Tarjetas de Red (NIC)

4.4.1 Generalidades de la Tarjeta de Red

4.4.1.1 Definición

Una tarjeta de interfaz de red (NIC - **N**etwork **I**nterface **C**ard) (también conocida como *adaptadora* o *tarjeta adaptadora*) es una placa de circuito instalada en un componente de equipo de informática, como un PC por ejemplo, que le permite conectar su PC a una red.

En la capa de enlace de datos podemos definir la tarjeta de red debido a que tiene y reconoce direcciones MAC (Subnivel de la capa de Enlace). Contienen un código único en todo el mundo, que se llama dirección de Control de Acceso al Medio (MAC, Media Access Control). Esta dirección se utiliza para controlar la comunicación de datos para el host de la red.

La NIC es el componente de hardware básico en las comunicaciones de red. Traduce la señal paralela producida por la computadora en un formato serie que se envía mediante el cable de red. La comunicación binaria (unos y ceros) se transforma en impulsos eléctricos, pulsos de luz, ondas de radio o cualquier esquema de transmisión de señales que usen los medios de comunicación en red, de manera que convierte el intercambio de señales a través de los medios de transmisión en una comunicación de datos efectiva.

4.4.1.2 Funciones de la Tarjeta de Red

Las funciones de la tarjeta de red son:

- Preparar los datos del equipo (formar tramas) para pasarlos a la capa física.

- En salida inyectar las tramas al medio de transmisión según el protocolo de comunicación.
- Recibir los datos que llegan por el cable y convertirlos en bytes para que puedan ser comprendidos por la unidad de procesamiento central del equipo (CPU).
- Controlar el flujo de datos entre el equipo y el sistema de cableado.

Todos los PC necesitan tarjetas de interfaz de red (***NIC***) para poder utilizarse en operaciones en red. Algunos se venden con la tarjeta NIC incorporada. Cuando se escoja una NIC (también conocida como tarjeta adaptadora) para instalar en un PC, se debe considerar lo siguiente:

- La velocidad del concentrador, conmutador, o servidor de impresora ***Ethernet*** (10Mbps) o ***Fast Ethernet*** (100Mbps).
- El tipo de conexión que se necesita: ***RJ-45*** para ***par trenzado o BNC*** para cable ***coaxial*** .
- El tipo de conector NIC disponible dentro de la PC: ***ISA*** o ***PCI***.

4.4.1.3 Velocidad de Conexión

Debe utilizarse una NIC de Ethernet con un conmutador Ethernet, y debe utilizarse una NIC de Fast Ethernet con un concentrador o conmutador Fast Ethernet.

Si se conecta la PC a un dispositivo dual speed que admite ambos valores, 10 y 100Mbps, se puede utilizar una NIC de 10Mbps o una NIC de 100Mbps. Un puerto en un dispositivo dual speed ajusta su velocidad automáticamente para que coincida con la velocidad más alta admitida por ambos extremos de la conexión. Por ejemplo, si la NIC soporta solamente 10Mbps, el puerto del concentrador dual speed que está conectado a dicha NIC pasará a ser un puerto de 10Mbps. Si la NIC soporta 100Mbps, la velocidad del puerto del concentrador será de 100Mbps.

De un modo semejante, si se tiene una NIC 10/100, se podrá conectar al concentrador Ethernet de 10Mbps o al concentrador Fast Ethernet de 100Mbps. La NIC 10/100 ajustará su velocidad para que

coincida con la velocidad más alta soportada por ambos extremos de la conexión.

4.4.1.4. Tipo de Conexión

El tipo de conector de la tarjeta de red dependerá de las características del medio físico (par trenzado, coaxial, fibra óptica, aire) al cual se conecte. (Ver figura 4.2).

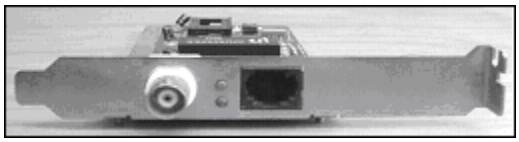


Figura 4.2. Tarjeta de Red con conectores BNC para cable coaxial y RJ-45 para UTP.

Si está instalando una red que utiliza cables de par trenzado UTP, se necesitará una NIC con un conector RJ-45. (Ver figura 4.3)



Figura 4.3. Tarjeta de Red con conector RJ-45 para cable UTP.

4.4.1.5. Conectores ISA y PCI

Hay dos tipos comunes de conectores de NIC para PC:

- Los zócalos ISA (Industry Standard Architecture. Arquitectura Estándar de la Industria). Son un tipo de slots o ranuras de expansión de 16 bits de longitud de palabra de datos para los ordenadores personales. Miden unos 14 cm. de longitud.
- Los zócalos PCI (Peripheral Component Interconnect Interconexión de Componentes Periféricos). Son un tipo de slots o ranuras de expansión de 32 bits de longitud de palabra de datos para los ordenadores personales. Los zócalos PCI tienen un mayor rendimiento que los ISA. Los zócalos PCI miden unos 9cm. de longitud.

Se debe consultar la guía del usuario de la PC para averiguar qué tipo de conector hay disponible en la PC.

4.4.1.6. NIC Especializadas

En algunos casos, es posible que necesite utilizar NIC especializadas. Por ejemplo, si el ordenador es un portátil, se necesitará utilizar una tarjeta ***PCMCIA.*** *(Personal Computer Memory Card International Association*. Asociación de la industria de fabricantes de hardware para ordenadores o computadoras portátiles encargada de la elaboración de estándares) y pueden ser de muy distintos tipos: memoria, disco duro, tarjeta de red, etc. Las tarjetas *PCMCIA* de 16 bits pueden recibir el nombre de *PC Card* y las de 32 bits el de *CARD BUS*. Una *tarjeta PCMCIA* es un dispositivo normalmente utilizado en computadoras portátiles para expandir las capacidades de este. (Ver figura 4.4).



Figura 4.4. Tarjeta PCMCIA.

Cuando elija una tarjeta PCMCIA, deberá considerar lo siguiente:

- La velocidad de su concentrador, conmutador o servidor de impresora - Ethernet (10Mbps) o Fast Ethernet (100Mbps).
- El tipo de conexión que necesita: RJ-45 para par trenzado o BNC para cable coaxial.
- Si tiene un puerto USB, podría considerar utilizar un Interfaz de red USB (*USB Network Interface*). (Ver figura 4.5).

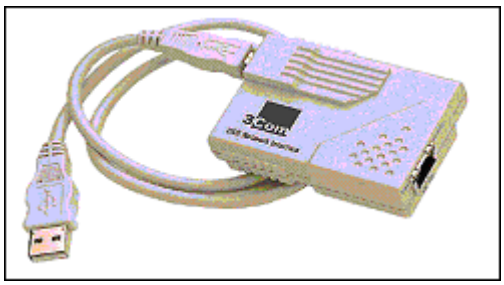


Figura 4.5 Interfaz de red USB.

En la actualidad se compran tarjetas de red UTP 10,100 y 1000 Mbps.

Algunos ejemplos de marcas comerciales de tarjetas de red se listan a continuación:

- 3COM
- ROBOTIC'S
- INTEL
- DELL
- ALTEC, etc.

4.5 Switch

4.5.1. Introducción

Una Red de Área Local (LAN) es el tipo más pequeño de red, y es el bloque básico para redes más grandes.

El uso de un concentrador o hub va a permitirnos crear una intranet de manera muy sencilla, permitiéndonos incorporar múltiples equipos a nuestra red con un esfuerzo mínimo, además de las innumerables ventajas que estas redes ofrecen a la hora de compartir archivos, impresoras y recursos, lo que aumenta la productividad de nuestras labores de manera considerable.

La mejor opción es crear una red por medio de un switch porque nos va a proporcionar unas velocidades de transmisión superiores al resto de conexiones existentes como cable serial, paralelo y hub. (Ver figura 4.6).



Figura 4.6. Concentrador o Switch.

A la hora de elegir el switch se deben tener en cuenta varios aspectos, entre ellos:

- Que soporte redes Ethernet, Fast Ethernet o superior (según los requerimientos del centro de cómputo, la empresa o institución donde se vaya a instalar).
- Que tenga suficientes conectores para comunicar a todos nuestros equipos.
- Que admita la conexión de otro switch más en una de sus conexiones por si deseamos ampliar la red en un futuro (cascadeo).

4.5.2. *Tipos de Switches*

Existen varios tipos de switchs entre ellos:

- ***Apilables***: Son concentradores independientes, que pueden apilarse para funcionar como uno sólo.
- ***De Chasis (Modulares)***: Son equipos con tarjetas modulares, que tienen que insertarse. Normalmente necesitan un módulo de administración.

4.6 Ethernet

Ethernet es la tecnología de red de área local **(LAN)** más ampliamente utilizada. Se diseñó para llenar el vacío que había entre las redes de larga distancia y baja velocidad y las redes especializadas, normalmente montadas en grandes salas de cómputo que transmitían datos a gran velocidad pero a distancias muy limitadas. Ethernet es la solución idónea para las aplicaciones en las que un medio de comunicación local debe transportar un tráfico esporádico, y a veces puede ser que intenso, con picos de velocidad alta.

Ethernet se consideró de alta velocidad cuando salió al mercado. Se diseñó para compartir recursos a nivel de grupo de trabajo local. Los objetivos de diseño incluyeron simplicidad, un bajo coste, la compatibilidad, la limpieza, un bajo retraso y una alta velocidad.

El Centro de investigación de la Xerox Corporation desarrolló el primer sistema experimental de Ethernet en los años 70**.** Se empleó como base para la especificación 802.3 del IEEE que se lanzó en 1980**.** Poco después, la Digital Equipment Corporation, la Intel Corporation y la Xerox Corporation se unieron para desarrollar una especificación Ethernet, la versión 2.0, que era sustancialmente compatible con la IEEE 802.3. Juntas, Ethernet y la IEEE 802.3, tienen actualmente la mayor cuota del mercado entre los protocolos LAN.

Las LAN Ethernet e IEEE 802.3 son redes broadcast. Esto significa que todas y cada una de las estaciones verá todas las tramas al mismo tiempo, sin importar si son el destino de los datos. Cada estación debe examinar las tramas recibidas para decidir si son el destino. Si lo son, la trama se pasa a un protocolo de capa superior en el interior de la estación para activar el proceso adecuado.

Las redes Ethernet utilizan el método de acceso CSMA /CD y corren sobre una gran variedad de tipos de cables a 10 Mbps. La red Ethernet es similar a los estándares de la serie IEEE 802.3

NOTA.- Para reafirmar algunos de los conceptos adquiridos en la teoría sobre la Capa de Enlace de datos del Modelo OSI; se recomiendan hacer las siguientes prácticas contenidas dentro de nuestra tesis:

Práctica 3. Instalación de una Red Básica en Plataforma Windows XP de Microsoft

En donde el alumno al finalizar la práctica:

Al finalizar la práctica tendrá la capacidad de instalar y configurar una tarjeta de red y crear una Mini-LAN conectando la computadora a la red existente a través de la instalación ya configurada en el laboratorio. Y a través del cable crossover construido en la práctica No.1.

Práctica 4. Instalación de una Red Básica en Plataforma LINUX

En donde el alumno al finalizar la práctica:

Al finalizar la práctica tendrá la capacidad de instalar y configurar una tarjeta de red y crear una Mini-LAN conectando dos computadoras utilizando un cable de conexión cruzada (crossover).

Capítulo 5

Capa de Red

Capítulo 5
Capa de Red

5.1. Capa de Red del Modelo OSI

5.1.1. Introducción

La ***capa de red*** se ocupa de controlar el funcionamiento de la subred. Una consideración clave de diseño es determinar cómo se encaminan los paquetes de la fuente a su destino. Las rutas se pueden basar en tablas estáticas que se "alambran" en la red y rara vez cambian. También se pueden determinar al inicio de cada conversación, por ejemplo en una sesión de terminal. Por último, pueden ser altamente dinámicas, determinándose de nuevo con cada paquete para reflejar la carga actual de la red.

Si en la subred se encuentran presentes demasiados paquetes a la vez, se estorbarán mutuamente, formando cuellos de botella. El control de tal congestión pertenece también a la capa de red.

En vista de que los operadores de la subred podrían esperar remuneración por su labor, con frecuencia hay una función de contabilidad integrada a la capa de red. Cuando menos, el *software* debe contar cuántos paquetes, caracteres o bits envía cada cliente para producir información de facturación. Cuando un paquete cruza una frontera nacional, con tarifas diferentes de cada lado, la contabilidad se puede complicar.

Cuando un paquete debe viajar de una red a otra para alcanzar su destino, pueden surgir muchos problemas. El tipo de direcciones que usa la segunda red puede ser diferente del de la primera; puede ser que la segunda no acepte en absoluto el paquete por ser demasiado grande; los protocolos pueden diferir y otras cosas. La capa de red debe resolver todos estos problemas para lograr que se interconecten redes heterogéneas

5.2. Direcciones IP y Subneting

5.2.1. *Conceptos Básicos*

Es importante saber que las direcciones IP están clasificadas acorde a un nivel por clase, siendo así que existen cinco clases de direcciones IP, las cuales son las siguientes:

- Clase A: Permite 16,777,214 hosts conectados a la red. Este tipo de direcciones son poco comunes, y se agotaron, ya que debido a sus características solo existían unas pocas miles de este tipo de direcciones.
- Clase B: Permite 65,534 hosts conectados a la red. Lamentablemente este tipo de direcciones ya no se ofrecen, y son sumamente costosas, por las comodidades que brinda (amplia gama de direcciones IP), pero representan un gran desperdicio de direcciones, ya que muy pocas redes Clase B llegan a conectar 65,534 hosts
- Clase C: Permite solo 254 hosts conectados a la red, y son actualmente las únicas ofrecidas a la venta. Se ha logrado implementar un método que permite fusionar varias direcciones Clase C, enmascarándolas como una sola red. Este método se conoce como CIDR (Classless InterDomain Routing).
- Clase D: Utilizada para propósitos de multicast.
- Clase E: Utilizada actualmente para fines experimentales.

Las direcciones IP son globalmente únicas, y tienen una estructura jerárquica de la forma <parte de red> + <parte de host>.

Tienen una notación de la siguiente manera:

- 10.3.2.4 (Clase A, pues el primer número está entre 0 y 127).
- 159.90.10.185 (Clase B, pues el primer número está entre 128 y 191).
- 192.12.69.77 (Clase C, pues el primer número está entre 192 y 223).
- 224.0.0.0 (Clase D, pues es en este número donde comienzan, hasta 239).
- 240.0.0.0 (Clase E, pues es en este número donde empiezan).

La idea principal de las direcciones IP era que cada parte de red identificara exactamente una red física. Pero resultó que esta meta tenía unos cuantos defectos. Por ejemplo, una red implementada en una universidad que tiene muchas redes internas decide conectarse a Internet. Para cada red, sin importar cuán pequeña sea, se necesita una dirección Clase C, por lo menos. Aún peor, para cada red con más de 255 hosts se necesitaría una dirección Clase B. Esto representa un gran desperdicio de direcciones, e ineficiencia en la asignación de direcciones IP, sin contar los altos costos. Peor aun, en dado caso que se llegaran muchas de las direcciones IP en una red Clase B, esto representa un aumento en el tiempo de envío de paquetes ya que la tabla de redireccionamiento de los *routers* aumentaría notablemente, y la búsqueda del destino en esta tabla tomaría mucho tiempo.

A medida que se agregan hosts se hace más grande la tabla de direccionamiento (*routing table*), lo que trae como consecuencia un aumento en los costos de los *routers* y una degradación en el rendimiento del *router*.

A medida que las redes crecen aumentando el número de segmentos, más direcciones de red (IP) son necesarias ya que cada segmento requiere un número propio. La InterNIC (Network Information Centers Cooperation), sin embargo, no puede manejar un número ilimitado de direcciones de red ya que se están acabando rápidamente debido a la alta demanda proveniente de la comunidad de Internet. Es por esto que los administradores de redes deberán trabajar con lo poco que tienen para acomodarse mejor a los requerimientos de la red y la reducida oferta de direcciones.

Una manera de lograrlo es tomar las direcciones que son asignadas a la red y expandir su capacidad con subredes. *Subnetting* (implementar subredes) permite incrementar el número de redes disponibles sin solicitar otra dirección IP.

5.2.2. Subredes (Subnetting)

Una gran solución a este problema es ofrecida por el s*ubnetting* (implementación de subredes), lo que permite reducir el número total de redes a ser asignadas. La idea es tomar una parte de red de una dirección de IP y asignar las direcciones IP de esa parte de red a

varias redes físicas, que serán ahora referidas como subredes. Pero hay que hacer ciertas cosas para que esto funcione. Primero, las subredes deben de estar cerca unas de otras, debido a que a un punto distante en el Internet todas lucirían igual a una sola red, teniendo solo una parte de red en común. Esto significa que un *router* solo estaría habilitado para seleccionar una sola ruta para llegar a cualquiera de las subredes, así que es mejor que se encuentren ubicadas en la misma dirección. No es que no vaya a funcionar si se encuentran muy separadas, solo que funcionará mucho mejor el sistema de subredes si logra tenerlas en la misma dirección general.

Un ejemplo práctico de utilización de subnetting es en una universidad con una red Clase B (como la USB) que tenga muchas redes físicas. Desde afuera de la universidad, todo lo que se necesita saber para alcanzar cualquier subred dentro de la red principal es la dirección del *router* que conecta a la universidad con el resto del Internet.

El mecanismo con el cual se puede lograr compartir un número de red (parte de red) entre distintas redes involucra la configuración de todos los nodos en cada subnet con una máscara de red, la misma para todos los nodos dentro de una subred. Con las máscaras de red se logra jerarquizar aun más la estructura de un IP, que como se dijo antes está constituida por <parte de red> + <parte de host>, incluyendo un nuevo nivel de jerarquía que llamaremos <número de subnet>. Como ya se sabe, todo los hosts en una misma red tienen la misma parte de red, pero ahora todos los hosts en la misma red física tendrán el mismo número de subnet, lo que hace que los hosts en la misma red, pero en distintas redes físicas, compartan la parte de red pero no el número de subnet, y esto como se puede notar ayuda notablemente en la transmisión de información, pues se complementan las tablas de direccionamiento con otro campo que ayudará a mejorar la eficiencia de envío de paquetes.

Para entender mejor el funcionamiento de las máscaras de red, supongamos que se quiere dividir una red Clase B en varias redes. Podríamos utilizar una máscara de red de la forma 255.255.255.0 (lo que pasado a binario son 1´s en los primeros 24 bits y 0´s en los últimos 8). Por lo tanto podríamos pensar que ahora los primeros 24 bits de una dirección IP representan la parte de red y los últimos 8 la

parte de host. Como los primeros 16 bits identifican una red Clase B, podemos pensar que la dirección no tiene dos sino tres partes: la <parte de red> + <parte de subnet> + <parte de host>.

5.2.3. Funcionamiento

Lo que subnetting significa para un host es que ahora está configurado con una dirección IP y una máscara de red para la subred a la cual se encuentra conectado. Cuando un host quiere enviar un paquete a una cierta dirección IP, lo primero que hace es realizar una operación de Y (AND) de bits entre su propia máscara de red y la dirección de destino. Si el resultado es igual al número de subnet del host que envía el paquete, entonces sabe que el host de destino está en la misma subred y el paquete de entregado directamente a través de la subred. Si el resultado no es igual, el paquete necesita ser enviado a un *router* para ser enviado desde este a otra subred.

Así como el trabajo de envío de un host cambia en una subred, también el trabajo de un *router* se ve afectado cuando se introduce la implementación de *subnetting*. Normalmente para satisfacer la estructura jerárquica de <parte de red> + <parte de host> el *router* tiene una tabla de direccionamiento formada por campos de la forma <Número_de_Red, Prox_Salto>. Para soportar *subnetting* la tabla ahora debe estar conformada por entradas de la forma <Número_de_Subred, Máscara_de_Subred, Próximo_Salto>. Para encontrar el lugar correcto en la tabla, el *router* aplica una operación AND entre la dirección de destino del paquete y la Máscara de Subred para cada una de las entradas, y cada vez revisa si el resultado es igual al Número de Subred de la entrada en turno. Si esto sucede, entonces esta es la entrada correcta a utilizar, y el *router* envía el paquete al *router* o a la interfaz especificada en el campo Próximo_Salto.

El algoritmo para el envío de *datagramas* se puede describir de la siguiente manera:

```
D = dirección IP de destino
para cada entrada en la tabla de direccionamiento
< Número_de_Subred, Máscara_de_Subred, Próximo_Salto >
D1 = Máscara_de_Subred & D
```

si (D1 = Número_de_Subred)
si (Próximo_Salto es una interfaz)
entrega el *datagram* directo a su destino
si no
entrega el *datagram* al *router* en Próximo_Salto

Normalmente en una tabla de direccionamiento se tiene un campo para el *router* predeterminado (default router), en dado caso que no se logre que algún *matching* ocurra. A pesar de ser funcional, este algoritmo resulta ineficiente pues realiza una búsqueda lineal a lo largo de toda la tabla, dándose los casos en que se puedan comparar repetidamente la dirección de destino con una máscara de subred que puede no ser diferente todas las veces.

5.2.4. Recomendaciones

Es bueno aclarar unos ciertos puntos sobre *subnetting*. No es necesario seleccionar una máscara con todos los bits 1s continuos, pero evitar esta estructura acarreara mayores complicaciones administrativas, ya que no será posible ver una parte de una dirección IP y decir ese es el número de subred.

También es importante saber que podemos poner múltiples subredes en una misma red física. El efecto que esto tendría es que se deberá forzar a los hosts en la misma red, pero en diferentes subredes, a hablar a través de *routers*. Esto puede ser útil para razones administrativas, como por ejemplo, separar distintos departamentos en una misma LAN, pero acarrea un aumento en los costos de la estructura o topología de la red.

Como habíamos mencionado antes, desde afuera de la red que se encuentra divida en subredes, los *routers* ven a la red como una red física sencilla y única, viendo las colecciones de subredes tan solo por la parte de red del IP, y guardan una entrada en su tabla de direccionamiento para saber como llegar a esa red. Dentro de la red los *routers* necesitan estar capacitados para direccionar paquetes a la correcta subred de destino. Es por esta razón que no todas las partes de la Internet ven exactamente la misma información de ruteo. Por ejemplo, un *router* en una red puede tener información de como llegar a otra subred cercana, pero no tiene ninguna entrada sobre otra

subred en la red. El envía el paquete al router correspondiente (o al default router), y este último se encarga de ver si sabe a donde enviarlo, si no, sigue el envío a otro *router* hasta que se consigue uno que sepa a donde enviarlo.

5.2.5. Beneficios del Subnetting

En conclusión, *subnetting* nos permite resolver los problemas de escalabilidad de dos maneras.

Primero, mejora nuestra eficiencia en la asignación de direcciones permitiendo no utilizar una nueva dirección Clase C o Clase B cada vez que necesitemos agregar una nueva red física. Segundo, nos ayuda a agregar información. Desde una distancia razonable, una colección compleja de redes físicas puede hacerse ver como una red sencilla, logrando así que la cantidad de información que los *routers* necesitan para enviar *datagramas* a esas redes sea reducida.

5.2.6. *Ejemplos Prácticos*

Las direcciones ip constan de 4 grupos de 8 bits, es decir que una red ip tiene un valor de 32 bits. Una dirección ip, por ejemplo 190.4.15.8, realmente sería:

10111110. 00000100. 00001111. 00001000

Esto viene dado por la siguiente tabla del sistema binario:

128 64 32 16 8 4 2 1
7 6 5 4 3 2 1 0
2 2 2 2 2 2 2 2 2exp0=1 2exp1=2... 2exp7=128
0 0 0 0 0 0 0 0 =0 (0)
0 0 0 0 0 0 0 1 =1 (1)
0 0 0 0 0 0 1 0 =2 (2)
0 0 0 0 0 0 1 1 =3 (2+1)
0 0 0 0 0 1 0 0 =4 (4)
0 0 0 0 0 1 0 1 =5 (4+1)
0 0 0 0 0 1 1 0 =6 (4+2)
0 0 0 0 0 1 1 1 =7 (4+2+1)

0 0 0 0 1 0 0 0 =8 (8)
0 0 0 0 1 0 0 1 =9 (8+1)
0 0 0 0 1 0 1 0 =10 (8+2)
0 0 0 0 1 0 1 1 =11 (8+2+1)
0 0 0 0 1 1 0 0 =12 (8+4)
0 0 0 0 1 1 0 1 =13 (8+4+1)
0 0 0 0 1 1 1 0 =14 (8+4+2)
0 0 0 0 1 1 1 1 =15 (8+4+2+1)
.....
.....
.....
1 1 1 1 1 1 1 1 =255 (128+64+32...+1)

Tenemos valores de 0 a 255 es decir 256 valores.

La dirección ip 127.0.0.1 es una dirección reservada que la podemos utilizar para hacer un ping a nuestra propia tarjeta de red para saber si el protocolo TCP/IP está bien instalado. Si este ping no nos responde habría que volver a instalar este protocolo.

Los valores 0 y 255, es decir 00000000 y 11111111 no se utilizan ya que están reservados para funciones de difusión y no pueden ser utilizados como direcciones de hosts.

Tenemos tres tipos de redes, las de clase A, clase B y clase C:

Clase A

El primer bit es 0 dejando 7 bits para crear las direcciones de red. Si 2 exponente 7 nos da como resultado 128 tenemos una dirección de 0 a 127 y como la dirección 0 y 127 no las *podemos* usar por ser de difusión nos quedan 126 redes.

De los cuatro octetos, el primero de ellos se refiere a la red y los tres restantes a los hosts:

1-127 (el 1er octeto) que indica la red. En este caso hasta ***126 redes*** (quitando 0 y 1)
El primer bit **1******* (128)**

256*256*256 (los tres restantes) que indican el nº de host que podemos tener, en este caso ***16,777,214 hosts*** (quitando 0 y 1)

red hosts
00000000 00000000 00000000 00000000
(1-127) 256 256 256

Clase B

Los dos primeros bits del primer octeto son 1 y 0, dejándonos 14 bits para crear las direcciones de red. 2 exponente 14 nos da como resultado 16384 redes.

De los cuatro octetos los dos primeros se refiere a la red y los dos restantes al nº de hosts:

128-191 (los dos primeros octetos) que indican la red. ***16,384 redes*** (quitando 0 y 1)

Los dos primeros bits **10****** **(128)**
11**** **(192)**

256*256 (los dos octetos restantes) que indican el nº de host en este caso ***65,534 hosts*** (quitando 0 y 1)

red hosts
00000000 00000000 00000000 00000000
(128-191) 256 256

Clase C

Los tres primeros bits del primer octeto son 1, 1 y 0 dejándonos 21 bits para crear las direcciones de red. 2 exponente 21 nos da como resultado 2097152 redes.

Los tres primeros octetos se refieren a la red y el octeto restante al nº de hosts:

192-223 (los tres primeros octetos) que indican la red. ***2,097,152 redes***(quitando 0 y 1)

Los tres primeros bits **110***** **(192)**
111*** **(224)**

256 (el último octeto) que indica el nº de hosts. ***254 hosts*** (quitando 0 y 1)

red hosts
00000000 00000000 00000000 00000000
(192 - 223) 256

5.2.7. Resumiendo

- CLASE A X.0.0.0 1-126.0.0.0
- CLASE B X.Y.0.0 128-191.0.0.0
- CLASE C X.Y.Z.0 192-223.0.0.0
- *Una dirección de red nunca puede ser 0 ya que representa la red local.*
- *Una dirección de red nunca puede ser 127 ya que está reservada.*
- *Una dirección de red nunca puede ser 255 ya que está reservada para broadcast.*
- *Una dirección de host nunca puede ser 0.*
- *Una dirección de host nunca puede ser 255 ya que está reservada para broadcast también.*

Cuando solicitamos una dirección de Internet a la **Network Information Center (NIC)**, estamos solicitando una dirección ip. Muchas redes son de clase C y lo que hace el NIC es darnos una dirección ip, por ejemplo 193.20.15.X donde X es el número de hosts que nosotros podemos tener para esa red desde 0 a 255. Las direcciones acabadas en 0 no se utilizan ya que son de broadcast y las 1 se utilizan para la puerta de enlace o gateway si instalamos un router.

La máscara de subred para la clase A es: 255.0.0.0
La máscara de subred para la clase B es: 255.255.0.0
La máscara de subred para la clase C es: 255.255.255.0

El **subnetting** trata de dividir una red en varios segmentos para descongestionar el tráfico de una red.

División de la red en dos segmentos

Por ejemplo tengo una red de clase B donde puedo tener hasta 65,534 hosts y quiero colocar un router para dividir esta red en dos segmentos.

Al ser de clase B sabemos que los dos primeros octetos pertenecen a la red y los dos siguientes a los hosts. Haciendo uso de un ejemplo válido, tendríamos:

190.4.X.0-------router-------190.4.X.0

¿Cómo sabemos aquí el valor de X?

Lo que si sabemos es que necesito hacer dos subredes, es decir, que del octeto de bits tomo dos, 00 000000, pero también sabemos que los rangos de direcciones 0 y 1 no se utilizan por lo que necesitaremos 4 direcciones.

¿Qué número de bits me va a dar 4 direcciones ips?

Sería 2 exponente n (nº de bits) será igual a 4 (2expn=4). Donde n es igual a 2. Necesito 2 bits como mínimo.

190.4.X.0 X=00(los dos bits que necesito) 000000

Calculando su submáscara de red

Sabemos que es de clase B ya que empieza por 190. Es decir de entrada sabemos que es 255.255.0.0, los dos bits que he tomado anteriormente los pongo en el tercer octeto para calcular la submáscara 11000000=192. Por lo que la submáscara será 255.255.192.0

Calculando las direcciones ip de cada segmento

Habíamos dicho que 2 exponente nº bits nos daba 4 posibles combinaciones y que ese número de bits es 2.

Aquí están las cuatro posibles combinaciones:

00 1ª combinación
01 2ª combinación
10 3ª combinación
11 4ª combinación

Sabemos que la primera y la última hay que desecharlas por ser de difusión (0 y 1) así que nos tenemos que quedar con la combinación segunda y tercera.

01 000000 = 64 190.4.64.0
10 000000 = 128 190.4.128.0

Los rangos por segmentos a utilizar son los siguientes:

De 190.4.64.0 sería de la 190.4.64.1 a la 190.4.127.254
De 190.4.128.0 sería de la 190.4.128.1 a la 190.4.254.254

División de la red en cuatro segmentos

Por ejemplo tengo una red de clase B donde puedo tener hasta 65,534 hosts y quiero colocar un router para dividir esta red en cuatro segmentos. Al ser de clase B sabemos que los dos primeros octetos pertenecen a la red y los dos siguientes a los hosts. Haciendo uso de un ejemplo válido, tendríamos:

190.4.X.0-------router-------190.4.X.0

¿Cómo sabemos aquí el valor de X?

Lo que si sabemos es que necesito hacer cuatro subredes, es decir que del octeto de bits tomo cuatro, 0000 0000, pero también sabemos que los rangos de direcciones 0 y 1 no se utilizan por lo que necesitaremos 6 direcciones.

¿Qué número de bits me va a dar 6 direcciones ips?

Sería 2 exponente n (nº de bits) será igual a 6 (2exp n=6). Donde n es como mínimo 3. Necesito 3 bits como mínimo.

190.4.X.0 X=000 (los tres bits que necesito) 00000

Calculando su submáscara de red

Sabemos que es de clase B ya que empieza por 190. Es decir de entrada sabemos que es 255.255.0.0, los tres bits que he tomado anteriormente los pongo en el tercer octeto para calcular la submáscara 11100000=224

Por lo que la submáscara será 255.255.224.0

Calculando las direcciones ip de cada segmento

Como 2 exponente 3 nos va a dar 8, vamos a tener un número mínimo de 8 direcciones ips. Ya que no hay ningún número más bajo que nos pueda dar como resultado 6.

Aquí están las ocho posibles combinaciones:

000 1ª combinación
001 2ª combinación
010 3ª combinación
011 4ª combinación
100 5ª combinación
101 6ª combinación
110 7ª combinación
111 8ª combinación

Sabemos que la primera y la ultima hay que desecharlas por ser de difusión (0 y 1) así que nos tenemos que quedar con las otras seis combinaciones.

001 00000 =32 190.4.32.0
010 00000 =64 190.4.64.0

011 00000 =96 190.4.96.0
100 00000 =128 190.4.128.0
101 00000 =160 190.4.160.0
110 00000 =192 190.4.192.0

Los rangos por segmentos que voy a utilizar son los siguientes:

De 190.4.32.0 sería de la 190.4.32.1 a la 190.4.63.254
De 190.4.64.0 sería de la 190.4.64.1 a la 190.4.95.254
De 190.4.96.0 sería de la 190.4.96.1 a la 190.4.127.254
De 190.4.128.0 sería de la 190.4.128.1 a la 190.4.159.254
De 190.4.160.0 sería de la 190.4.160.1 a la 190.4.191.254
De 190.4.192.0 sería de la 190.4.192.1 a la 190.4.223.254

Ahora solo necesito 4 direcciones ips por lo que despreciaríamos las siguientes, así que podríamos coger la 190.4.32.0 y la 190.4.192.0 ya que podríamos jugar con mas valores.

5.3. Enrutamiento

Internet está formado por una serie de Sistemas Autónomos (AS) (es decir; por un conjunto de redes bajo una administración común, que comparte políticas de enrutamiento estratégicas), que definen políticas de administración y de enrutamiento de diferentes organizaciones. Un Sistema Autónomo ocupa Protocolos de Compuerta Interna (IGPs *Interior Gateway Protocol*), tales como RIP, OSPF, ISIS, etc. para enrutar información del mismo Sistema Autónomo. Estos a su vez se interconectan vía un Protocolo de Compuerta Externa (*EGP Exterior Gateway Protocol*), tales como EGP y BGP para intercambio de información entre Sistemas Autónomos.

Un router es un dispositivo de red de propósito específico, que a través de los protocolos de enrutamiento IGPs y EGPs avanzan los paquetes de información de un lugar de origen a un destino, determinando la mejor ruta. Este equipo actúa sobre la capa 3 (Red) del modelo OSI. Los routers construyen tablas de enrutamiento conteniendo información de los mejores caminos (paths), a todos los destinos que conocen.

Existen dos tipos de enrutamiento: El directo y el indirecto.

Directo:

Es la transmisión directa de un paquete de información de una máquina a otra, siempre y cuando ambas máquinas pertenezcan al mismo segmento lógico y por lo tanto físico.

Indirecto:

Se presentan cuando las máquinas que se quieren comunicar no están en la misma red (lógica y física), lo cual se traduce en la necesidad reutilizar un equipo adicional, que se encargue de comunicar la red origen al destino.

Un router se comunica con otro router, con el fin de intercambiar tablas de enrutamiento, información de control y diversos mensajes. Este intercambio de información se da gracias a los algoritmos de enrutamiento.

Los algoritmos de enrutamiento comúnmente emplean una tabla de enrutamiento en la que almacenan información referente a los posibles destinos y la forma en como llegar a ellos. Las tablas se generan a partir de dos procesos: Iniciación del proceso de enrutamiento e intercambio de tablas con otros routers.

5.3.1. Características de Diseño

Existen diferentes algoritmos de enrutamiento y cada uno de ellos fue diseñado con un propósito en particular, es por eso que tienen diferente impacto sobre la red y sobre los recursos de los routers, cada uno de estos algoritmos utilizan una variedad de métricas (es una medida de confiabilidad asignada a un anuncio de enrutamiento), que afectan directamente en el cálculo de la mejor ruta. Los algoritmos de enrutamiento generalmente tienen una ó más de las siguientes características de diseño:

Robustez

La robustez nos indica que el algoritmo de enrutamiento debe de soportar una gran cantidad de rutas en las tablas de enrutamiento,

además debe ser tolerante a malas implementaciones y a fallas de hardware.

Estabilidad

Esta característica de diseño nos indica que los anuncios de enrutamiento que se encuentran en nuestros equipos, deben ser coherentes con lo que está aconteciendo en la red en todo momento.

Flexibilidad

Los algoritmos de enrutamiento deben adaptarse rápidamente a la gran variedad de circunstancias que ocurran en la red, es decir, los algoritmos deben ser programados para adaptarse a los cambios en la red como son el ancho de banda, tamaño de colas, retardos en la red, y otras variables.

Simplicidad y Bajo Overhead

El algoritmo de enrutamiento debe ser lo más simple posible, en otras palabras, debe ofrecer funcionalidad y eficiencia con un mínimo de recursos. La eficiencia es particularmente importante cuando la implementación del algoritmo de enrutamiento se ejecuta en un router con recursos físicos limitados. Por ejemplo el router no debe de utilizar mucho ancho de banda para transportar sus anuncios hacia otro router.

Rápida Convergencia

La convergencia es el proceso de conformidad por todos los routers sobre las rutas óptimas, es decir, cuando un anuncio de una red deja de ser anunciada por un router, se deben de mandar mensajes de actualización de rutas para recalcular rápidamente la ruta óptima, para poder llegar a ese destino por otra trayectoria. Si no existiera una convergencia rápida, se provocaría una inconsistencia en los anuncios de las redes: la convergencia se lleva a cabo una vez que los routers poseen la misma información de enrutamiento.

Optimización

Si tenemos varias rutas en nuestra tabla de enrutamiento para poder llegar a un destino determinado, el protocolo de enrutamiento debe ser lo suficientemente inteligente para poder seleccionar la mejor ruta.

5.3.2. Clasificación

Los algoritmos de enrutamiento pueden ser clasificados en base a diferentes criterios tales como:

- Estáticos o Dinámicos
- Single-Path o Multipath
- Plano o Jerárquico
- Link State o Distance Vector
- Intadominio ó Interdominio

5.3.2.1. Estáticos o Dinámicos

En los algoritmos de enrutamiento estático el administrador de la red crea manualmente una tabla de enrutamiento. Dicha tabla no cambia a menos que el administrador lo haga. Los algoritmos que usan rutas estáticas son fáciles de diseñar y trabajan bien en ambientes en donde el tráfico de la red es relativamente predecible y el diseño de la red es relativamente simple.

Actualmente este tipo de algoritmos no son utilizados debido a que no son capaces de detectar los cambios que acontecen en la red.

Los algoritmos de enrutamiento dinámico van ajustando las rutas dinámicamente en tiempo real, gracias a que analizan los mensajes de actualización de rutas. Este tipo de algoritmos también permiten la implementación de rutas estáticas cuando sean necesarias.

5.3.2.2. Single-Path o Multipath

Existen algoritmos de enrutamiento más sofisticados que aceptan múltiples rutas para un mismo destino. Estos algoritmos son

llamados Multipath, y pueden balancear cargas a través de múltiples líneas. Los algoritmos Single-Path, por el contrario, sólo aceptan una ruta por cada red.

5.3.2.3. Plano o Jerárquico

En un sistema de enrutamiento plano, todos los routers son parejas de todos. En un sistema jerárquico, existen diferentes niveles, en donde el nivel más alto es el nivel de backbone, los routers se encuentran interconectados con otros routers de backbone y con algunos de los routers en el siguiente nivel. Los routers en los siguientes niveles se conectan con otros routers en este nivel creando un grupo de routers o área, también se conecta con uno o varios routers del backbone pero no con todos. Para la comunicación con otras áreas es necesaria que la información primero sea enviada al nivel del backbone y después al área destino.

5.3.2.4. Link State o Distance Vector

Los algoritmos Link State (conocidos como algoritmos Shortest Path First), envían su información de enrutamiento a todos los nodos de la red. Sin embargo, cada router envía sólo una porción de su tabla de enrutamiento, la cual describe el estado de sus interfases. Los algoritmos Distance Vector (conocidos como algoritmos Bellman-Ford), envían toda la tabla de enrutamiento solamente a sus vecinos. Los algoritmos de tipo Link State son menos propensos a loops (bucles) de enrutamiento, porque convergen más rápidamente que los algoritmos Distance Vector. Una de las desventajas de los algoritmos Link State, es que su ejecución requiere del uso de más memoria y CPU.

5.3.2.5. Intadominio ó Interdominio

Algunos algoritmos de enrutamiento trabajan solo dentro de dominios (un dominio es un grupo de routers que hablan el mismo protocolo de enrutamiento, normalmente coincide con el Sistema Autónomo), otros trabajan dentro y entre dominios, es decir, los routers que trabajan con algoritmos Intradominio sólo intercambian tablas de enrutamiento dentro de su Sistema Autónomo, en cambio los

routers que utlizan algoritmos Interdominio pueden intercambiar sus tablas de enrutamiento dentro y fuera de su Sistema Autónomo.

5.4. Jerarquía en capas

El éxito de una red se basa en permitir un crecimiento sin que se comiencen a presentar problemas de saturación, por esta razón se vuelve indispensable un diseño jerárquico que divida sus funciones en capas.

Si una red es diseñada en forma jerárquica, cada una de las capas subsecuentes actuará como un filtro, permitiendo que la red pueda crecer de una manera eficiente. De esta forma se logra mantener el "tráfico local" de manera local, y sólo los datos e información acerca de recursos globales que necesiten viajar fuera del dominio inmediato sean quienes tengan que hacerlo, evitando que así el tráfico local consuma los recursos de la red de forma inadecuada. El número de capas que son necesarias depende del control de tráfico que se requiera. Para determinar el número de capas requeridas, se debe de identificar la función que tendrá cada una de ellas.

Cada capa en un diseño jerárquico es responsable de prevenir que tráfico innecesario sea enviado a las capas superiores. La meta es permitir que sólo el tráfico relevante pueda atravesar la red y con esto reducir la carga en la red. Si la meta es claramente conocida, la red puede escalar más eficientemente. Un esquema que normalmente es suficiente, es un modelo jerárquico en tres capas, tales capas son:

- La capa de Acceso
- La capa de Distribución
- La capa de Core

5.4.1. Capa de Acceso

La capa de Acceso es aquella en donde se conectan los dispositivos de orilla, los cuales conectan a las redes de los usuarios, en donde son necesarias grandes densidades de puertos. Los dispositivos de capa 3 (como los routers) son los responsables de la entrada y salida de tráfico, asegurando que el tráfico local no se

extienda por toda la red. En ocasiones dependiendo del tipo de usuario llegan a ser necesarias conexiones redundantes a la capa de distribución.

5.4.2. Capa de Distribución

La capa de Distribución es responsable de la conexión al Core filtrando todas las actualizaciones y seleccionando en forma granular el acceso a los usuarios. Una de sus funciones es la seguridad. Se recomienda tener conexiones redundantes al Core, aunque en ocasiones llegan a ser necesarias conexiones redundantes hacia los usuarios o entre dispositivos en la misma capa.

5.4.3. Capa de Core

La capa de Core tiene como función principal garantizar la conectividad a lo largo de toda la red. Su característica principal es la confiabilidad. Una falla a este nivel produce que grandes partes de la red queden incomunicados. Para asegurar una conectividad continua se debe de diseñar con altos niveles de redundancia tantos como sea posibles (preferentemente en una topología de full-mesh ó parcial mesh), también se debe de eliminar toda la latencia que sea posible, debido a esto, no es recomendable que se tengan que tomar decisiones complejas de enrutamiento como lo podrían ser los filtros. Por esto se recomienda que los filtros se implementen en las capas de Acceso y Distribución dejando al Core la responsabilidad de transportar los datos de la forma más rápida confiable. En algunas ocasiones se implementan ciertas políticas de QoS (Calidad de Servicio).

5.5. Tipos de Backbone

Una característica muy importante en el diseño del backbone, es definir los tipos de servicios que se van a proporcionar; voz, dato y video, y que tecnología se utilizará para soportarlos. Podemos hacer una clasificación de los backbone de acuerdo al modelo OSI, si va a ser construido para brindar servicios de capa2 se trata de un backbone “Switcheado” y cuando se construye para soportar servicios de capa3 se habla de un backbone “Enrutado”.

5.5.1. Backbone Switcheado

Este tipo de backbone se crea utilizando tecnologías como Frame Relay y/o ATM, el Core está formado por switches que a su vez se encuentran rodeados por Routers. Algunas de las principales ventajas de este tipo de backbone son: se puede tener un control más específico en cuanto asignación de ancho de banda y calidades de servicios (QoS), son tecnologías orientadas a circuitos. Sus principales desventajas son el enrutamiento y la solución de problemas complejos.

5.5.2. Backbone Enrutado

La infraestructura de este tipo de backbone está formada por routers, las conexiones entre ellos son realizas con enlaces PPP y/o HDLC. Las configuraciones de enrutamiento son más fáciles de realizar al igual que la solución de problemas, además también se obtiene un menor overhead (es información de control que requieren los equipos de comunicaciones, pero que no es parte de la información de los usuarios. Es un parámetro que se debe minimizar debido a que consume recursos de la red).

5.6.Protocolo de enrutamiento OSPF

OSPF (Open Shortest Path First) fue desarrollado por la IETF (Internet Engineering Task Force) como respuesta a la comunidad de Internet para introducir un nuevo protocolo de compuerta interna (IGP) de alta funcionalidad. Actualmente OSPF es el protocolo recomendado por la IETF para ser usado como protocolo intradominio.

OSPF es un protocolo de estado de enlace, que utiliza el algoritmo de Dijkstra SPF (Shortest Path First) y que es abierto (Open) esto significa que no es propietario de alguna organización o compañía.

La mayor ventaja del protocolo OSPF es su rápida convergencia, soporte para redes de gran tamaño y una baja susceptibilidad para conservar información errónea de enrutamiento.

OSPF define cinco tipos diferentes de red:

- Redes tipo Punto a Punto (Point to Point).
- Redes tipo Broadcast.
- Redes tipo Nonbroadcast Multi-acceso (NBMA).
- Redes tipo Punto Multi-punto.
- Enlaces Virtuales (Virtual Link).

5.7.Protocolo de Hello

Las principales funciones de este protocolo son:

- Ser el mecanismo para el descubrimiento de vecinos.
- El anuncio de varios parámetros con los cuales dos routers deben de estar de acuerdo para poder ser vecinos.
- Asegurar una comunicación bidireccional entre dos vecinos.
- La elección de un Router Designado (DR-Designated Router) y el respaldo del designado (Backup DR) en redes Broadcast y Nonbroadcast multiacceso.

Los routers que hablan OSPF envían periódicamente mensajes de Hello desde las interfases que tienen habilitado OSPF. Este período es conocido como HelloInterval y el tiempo puede ser configurado, un valor típico es de 10 segundos. Si un router no escucha un mensaje de Hello desde su vecino dentro de un cierto período de tiempo es conocido como DeadInterval, un valor para este tiempo es cuatro veces el valor del HelloInterval.

Cada paquete de Hello contiene la siguiente información:

- El router ID del router que originó el mensaje.
- El área ID de la interfaz del router que originó el mensaje.
- El tipo de autentificación y la información de autentificación para la interfaz que lo originó.
- El HelloInterval de la interfaz originante.
- El DeadInterval de la interfaz originante.
- La prioridad del router.
- El DR y el BDR.

- Cinco banderas de bits con un significado opcional.
- Los routers IDs de los routers vecinos originantes. Esta lista sólo contiene routers desde los cuales los mensajes de Hello fueron avanzados en las interfases originantes del último Router DeadInterval.

NOTA.- Para reafirmar algunos de los conceptos adquiridos en la teoría sobre la Capa de Red del Modelo OSI; se recomiendan hacer las siguientes prácticas contenidas dentro de nuestra tesis:

Práctica 5 Enrutamiento de paquetes

En donde el alumno al finalizar la práctica:

Al finalizar la práctica, se familiarizará con el manejo de algunas herramientas del Sistema Operativo Linux, como son route y traceroute, y sus similares en Windows, como son route y tracert, enfocadas al enrutamiento de paquetes a través de la red.

Práctica 6 Análisis de Paquetes

En donde el alumno al finalizar la práctica:

Conocerá y comprenderá los fundamentos de la monitorización de redes, encapsulado de unidades a diferentes niveles y demultiplexación de las mismas.

Adquirirá la habilidad necesaria para realizar los filtros adecuados.

Reafirmará conocimientos teóricos acerca del protocolo ARP mediante observación de casos reales.

Capítulo 6

Capa de Transporte

Capítulo 6
Capa de Transporte

6.1. Capa de Transporte

Todas las tecnologías de red establecen un tamaño máximo para las tramas que pueden ser enviadas a través de la red. Por ejemplo, Ethernet limita el tamaño del campo de datos a 1,500 bytes. Este límite es necesario por varias razones:

- Las tramas de tamaño reducido mejoran el rendimiento de una red compartida por muchos dispositivos. Si el tamaño de las tramas fuera ilimitado, su transmisión podría monopolizar la red durante un tiempo excesivo. Las tramas pequeñas permiten que los dispositivos se sumen a intervalos cortos de tiempo y tengan más opciones de acceder a la red.
- Al utilizar tramas pequeñas, es necesario volver a transmitir menos datos cuando se produce un error. Si un mensaje de 100 KB contiene un error en un sólo byte, es preciso volver a transmitir los 100 KB. Si el mensaje se divide en 100 tramas de 1 KB, basta con retransmitir una sola trama de 1 KB para corregir el error.

Una de las responsabilidades de la capa de transporte consiste en dividir los mensajes en fragmentos que coincidan con el límite del tamaño de la red. En el lado receptor, la capa de transporte reensambla los fragmentos para recuperar el mensaje original.

Cuando un mensaje se divide en varios fragmentos, aumenta la posibilidad de que los segmentos no se reciban en el orden correcto.

Al recibir los paquetes, la capa de transporte debe recomponer el mensaje reensamblando los fragmentos en el orden correcto. Para ello, la capa de transporte incluye un número de secuencia en la cabecera del mensaje.

Muchas computadoras son multitarea y ejecutan varios programas simultáneamente. Por ejemplo, la estación de trabajo de un usuario puede estar ejecutando al mismo tiempo un proceso para

transferir archivos a otra computadora, recuperando el correo electrónico y accediendo a una base de datos de la red. La capa de transporte debe entregar los mensajes del proceso de una computadora al proceso correspondiente de la computadora de destino.

Según el modelo OSI, la capa de transporte asigna una identificación de punto de acceso a servicio (SAP) a cada paquete (puerto es el término TCP/IP correspondiente a un punto de acceso a servicio). La ID de un SAP es una dirección que identifica el proceso que ha originado el mensaje. La ID permite que la capa de transporte del nodo receptor encamine el mensaje al proceso adecuado.

La identificación de mensajes de distintos procesos para posibilitar su transmisión a través de un mismo medio de red se denomina multiplexión. El procedimiento de recuperación de mensajes y de su encaminamiento a los procesos adecuados se denomina demultiplexión. Esta práctica es habitual en las redes diseñadas para permitir que varios diálogos compartan un mismo medio de red.

Dado que una capa puede admitir distintos protocolos, la multiplexión y demultiplexión puede producirse en distintas capas.

Algunos ejemplos:

- Transporte de distintos tipos de tramas Ethernet a través del mismo medio (capa de enlace de datos).
- Soporte simultáneo de NWLink y de TCP/IP en computadoras Windows NT (capa de enlace de datos).
- Mensajes de varios protocolos de transporte como TCP y UDP en un sistema TCP/IP (capa de transporte).
- Mensajes de distintos protocolos de aplicación (como Telnet, FTP y SMTP) en un host UNIX (capas de sesión y superiores).

Aunque las capas de enlace de datos y de red pueden encargarse de detectar errores en los datos transmitidos, además esta responsabilidad suele recaer sobre la capa de transporte. La capa de transporte puede realizar dos tipos de detección de errores:

- **Entrega fiable.** Entrega fiable no significa que los errores no puedan ocurrir, sino que los errores se detectan cuando ocurren. La recuperación puede consistir únicamente en notificar el error a los procesos de las capas superiores. Sin embargo, la capa de transporte suele solicitar que el paquete erróneo se transmita nuevamente.
- **Entrega no fiable**. No significa que los errores puedan producirse, sino que la capa de transporte no los verifica. Dado que la comprobación requiere cierto tiempo y reduce el rendimiento de la red, es frecuente que se utilice la entrega no fiable cuando se confía en el funcionamiento de la red. Éste es el caso de la mayoría de las redes de área local. La entrega no fiable es preferible cuando los mensajes constan de un alto número de paquetes. Con frecuencia, se denomina entrega de datagramas y cada paquete transmitido de este modo se denomina datagrama.

La idea de que siempre es preferible utilizar la entrega fiable puede constituir un error a la hora de diseñar la red. La entrega no fiable es aconsejable en al menos dos situaciones: cuando la red es altamente fiable y es necesario optimizar su rendimiento o cuando los paquetes contienen mensajes completos y la pérdida de un paquete no plantea un problema crítico.

En Internet existen dos protocolos principales, TCP y UDP. Ambos se encuentran dentro de la capa de transporte. Estos protocolos son diferentes entre sí, uno de ellos, TCP, es orientado a conexión (de entrega fiable) y UDP es un protocolo sin conexiones (de entrega no fiable).

6.2. TCP

El protocolo TCP, Transfer Control Protocol, o Protocolo de Control de Transmisión fue especialmente diseñado para realizar conexiones en redes inseguras, es decir, en redes de tipo C según la terminología de OSI. TCP es un protocolo de capa de transporte. TCP acepta bloques de datos de cualquier longitud, procedentes de las capas superiores o de los procesos de los usuarios, y los convierte en fragmentos de 64 KB como máximo, que pasan a la capa de red que, a su vez, puede volver a fraccionarlos para su transmisión efectiva.

Cada uno de los bloques de datos se transmite como si fuera un datagrama separado con entidad propia. TCP es el responsable de ensamblar los datagramas recibidos por el receptor, ya que la red IP puede desordenarlos al utilizar caminos diversos para alcanzar su destino. IP no garantiza que los datagramas lleguen a su destino, por lo que es necesaria una entidad superior, en este caso TCP, que se encargue de ello a través de un sistema de temporizadores y de retransmisiones en el caso de que haya problemas.

6.2.1. Establecimiento de una Conexión

6.2.1.1. Inicio una Conexión entre dos Aplicaciones

Siempre, antes de poder comunicarse ambas partes llaman a una subrutina que crea un bloque de memoria para almacenar los datos de TCP y de IP durante la conexión, como las direcciones de los conectores (sockets), los números actuales de secuencia, el valor inicial de IP para el tiempo de vida y otros. La aplicación servidora espera a los clientes. Un cliente que desee acceder al servidor lanza una solicitud de conexión mediante la dirección IP y el puerto del servidor.

6.2.2. *Paquetes TCP*

TCP (Protocolo de Control de Transmisión) es un protocolo de comunicación que ofrece un servicio orientado a conexión punto a punto, es decir, una conexión TCP tiene dos terminales. Define un servicio fiable de transmisión, asegurando que los datos enviados llegarán a su destino sin fallas, y lo harán completos y ordenados. TCP brinda un servicio dúplex donde una aplicación solicita una conexión, se crea la conexión, puede transmitir en ambos sentidos y termina la conexión, todo de manera confiable.

El encabezado de TCP cuenta con 12 campos (Ver figura 6.1):

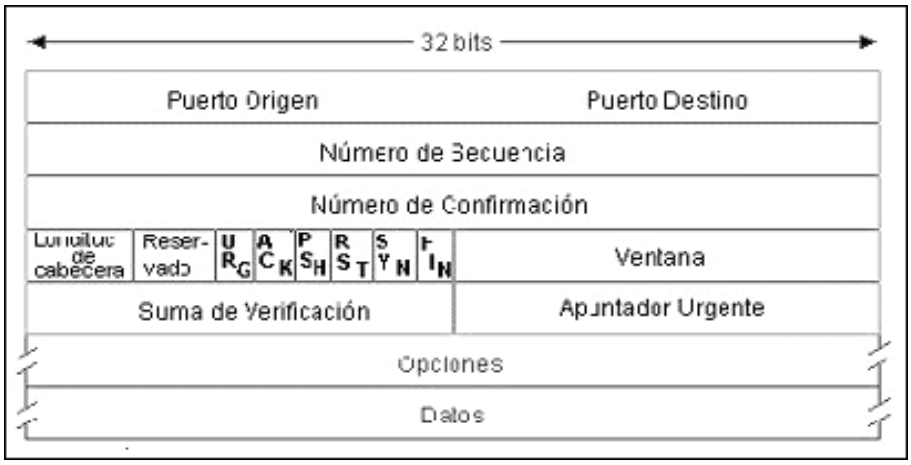


Figura 6.1. Cabecera TCP.

- Puerto de origen.
- Puerto de destino.
- Número de secuencia que especifica el número del primer byte del mensaje
- Número de confirmación que contiene el número de secuencia del siguiente byte.
- La longitud de cabecera TCP que indica la cantidad de palabras de 32 bits en la cabecera.
- Espacio reservado de 6 bits seguido por una serie de banderas de 1 bit: URG (urgente), ACK (número de acuse de recibo), PSH (indica datos empujados, con PUSH), RST (para restablecer una conexión o para rechazar un segmento no válido o un intento de conexión), SYN (para establecer conexiones) y FIN (para liberar una conexión).
- La Ventana especifica el espacio de almacenamiento disponible para los datos.
- La suma de verificación indica que el mensaje se daño en el trayecto.
- El apuntador de urgente marca el primer dato de urgente en el mensaje.
- Las opciones.
- Los datos que contienen información de las capas superiores.

Para que TCP pueda crear y terminar de manera confiable sus conexiones utiliza un método llamado "acuerdo de tres vías". Si un host 1 quiere establecer una conexión con un host 2, ejecuta una petición CONNECT con la dirección y el puerto del host 2, envía un

mensaje TCP con el bit SYN (segmento de sincronización) en encendido y espera una respuesta. Este mensaje TCP busca el puerto en el host 2, y si en el puerto no existe un proceso en estado de LISTEN, se manda un mensaje al host 1 rechazando la conexión. Pero si algún proceso está en estado LISTEN, entonces este puede decidir si acepta o no la conexión; si se acepta entonces manda un mensaje de ACK (acuse de recibo). Si no se acepta la conexión manda entonces un mensaje de rechazo. Y por último si el host 1 recibe el ACK manda entonces los datos y al final de estos añade un segmento FIN (o de terminación).

6.2.3. Rendimiento

Hay muchos factores que afectan el rendimiento de TCP. Los básicos son los recursos como la memoria y el ancho de banda. El ancho de banda y los retardos de la red subyacente imponen límites al rendimiento. Una baja calidad de transmisión implica un gran número de datagramas descartados. Al descartarlos se desencadenan retransmisiones con lo que se reduce el ancho de banda efectivo.

Un receptor que dispone de un gran buffer de entrada permite que el emisor continúe transmitiendo sin interrupción. Resulta muy importante en redes grandes con gran retardo.

Otro factor que afecta el rendimiento TCP son las características y la capacidad del servidor o hosts para reaccionar rápidamente ante eventos de alta prioridad y realizar cambios de contexto, es decir, dejar una tarea y comenzar otra de mayor importancia. Un servidor puede estar dando servicios a muchos usuarios locales interactivos, procesos de fondo y docenas de comunicaciones.

Se necesitan recursos suficientes de CPU para procesar rápidamente las cabeceras de TCP, un procesador que no pueda calcular de manera eficaz los valores de la suma de control de una cabecera TCP suele ralentizar la transmisión de datos. Por ultimo, los fabricantes deben ser capaces de implementar técnicas de configuración mas sencillos para los administradores de red y de esta manera puedan ajustarlos a su entorno.

6.3. UDP

6.3.1. *Paquetes UDP*

UDP (Protocolo de Datagramas de Usuario) es un protocolo de transporte sin conexión previa, ofrece un mecanismo para enviar Datagramas sin tener que establecer y liberar conexiones. Es un servicio no fiable ya que no agrega al mensaje IP funciones de confiabilidad, control de flujo y recuperación de errores. Por esto el UDP es mucho más ligero que TCP y como su encabezado es mucho mas pequeña genera menos carga en la red. El segmento UDP tiene una encabezado de 8 bytes seguida de los datos. El encabezado UDP, que se muestra en la figura 6.2 tiene cuatro campos: los puertos de origen y destino que identifican los puntos terminales, la longitud UDP que tiene el tamaño de la cabecera y los datos, y la suma de comprobación que es una verificación de la integridad de los datos, que es opcional (0 significa que no se calcula).

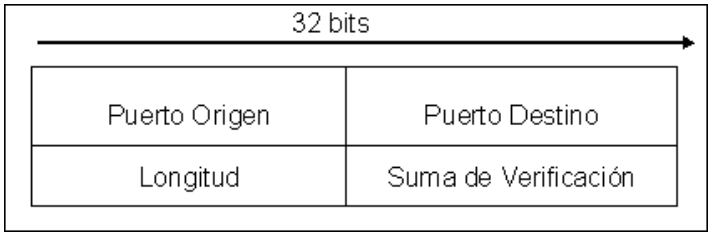


Figura 6.2 Cabecera UDP

6.4. Sockets

6.4.1. El Programa Sock

Para ejemplificar la Capa de Transporte, utilizaremos un pequeño programa en nuestra práctica que nos permita enviar y recibir información utilizando los protocolos TCP y UDP.

Este programa llamado sock ofrece un modo de acceder a la interfaz de los sockets sin tener que programar. El programa conecta la entrada/salida estándar (teclado/pantalla) con un socket cuyas características se especifican mediante parámetros al ejecutar la orden. Mediante la redirección de la entrada o la salida se puede

enviar el contenido de un archivo o almacenar en un archivo la información recibida.

Los sockets pueden ser de dos tipos: UDP o TCP, que corresponden con un servicio sin conexión, que no garantiza ni la entrega ni el orden de entrega de la información (UDP) y otro servicio que garantiza la entrega ordenada y sin errores de la información (TCP).

Además, se sabe que una aplicación puede comenzar iniciando la comunicación (enviando información) o bien puede esperar pacientemente hasta que la otra le solicite el inicio de la comunicación (espera petición).

El programa sock nos va a permitir imitar cualquiera de estas situaciones y alguna otra cosa más.

6.4.2. Introducción a los Sockets

Los sockets son puntos o mecanismos de comunicación entre procesos que permiten que dos procesos hablen (emita o reciba información) entre sí, incluso estando en distintas máquinas. Este mecanismo trabaja sobre la capa de transporte, que es la que establece una conexión entre dos hosts utilizando el protocolo TCP o UDP. Para establecer dicha conexión, se crea el socket para comunicar los procesos. Estos procesos pueden ser algún servicio en particular (TELNET, FTP, SMTP, Domain, etc.).

Los sockets surgen en el año de 1980, cuando la agencia ARPA (Agencia para la Investigación de Proyectos Avanzados) fundó un grupo de investigación en la Universidad de Berkeley (California) para llevar a cabo la implementación del software TCP/IP dentro del propio núcleo del sistema operativo UNIX. Como parte del proyecto, los diseñadores crearon una interfaz para permitir que las aplicaciones pudieran acceder a los servicios que brinda el software TCP/IP, y añadieron nuevas llamadas para dar soporte a las funciones TCP/IP que no se podían implementar adecuadamente con el conjunto de llamadas disponibles. Con estas modificaciones surge el sistema operativo conocido como UNIX Berckeley o BSD UNIX. Esta interfaz socket se convirtió en un estándar.

6.4.3. Características de los Sockets

Todos los sockets tienen tres características fundamentales: tipo, dominio y protocolo.

6.4.3.1. Tipo de Sockets

El tipo define las propiedades de las comunicaciones que se pueden dar entre el cliente y el servidor. Los tipos disponibles son los siguientes:

- **Tipo SOCK_DGRAM:** sockets para comunicaciones en modo no conectado, con envío de datagramas de tamaño limitado. En dominios de Internet se usa el protocolo UDP.
- **Tipo SOCK_STREAM:** para comunicaciones fiables en modo conectado, de dos vías y con tamaño variable de los mensajes de datos. En dominios de Internet se ocupa el protocolo TCP.
- **Tipo SOCK_RAW:** permite el acceso a protocolos de más bajo nivel como el IP (nivel de red).
- **Tipo SOCK_SEQPACKET:** tiene las características del SOCK_STREAM pero además el tamaño de los mensajes es fijo.

6.4.3.2. Dominio de un Socket

Indica el formato de las direcciones que podrán tomar los sockets y los protocolos que soportarán dichos sockets. Así se puede especificar si la conexión será a través de Internet o dentro de la misma máquina.

La estructura sockaddr empaqueta la dirección y algunos datos de los sockets. De manera general es como la siguiente:

sockaddr (family, address)

Los dominios pueden ser:

- **Dominio AF_UNIX (Address Family UNIX):** El cliente y el servidor deben estar en la misma máquina.
  sockaddr (family, address)
  Family: en este caso es AF_UNIX
- **Dominio AF_INET (Address Family INET):** El cliente y el servidor pueden estar en cualquier máquina de la red Internet.
  sockaddr (family, port, address, sin_zero[8])
  family: en este caso es AF_INET
  port: número de puerto
  address: dirección de Internet
  sin_zero[8]: campo de 8 ceros (reservado para otras aplicaciones)

Existen otros dominios no tan comunes como:

- **Dominio AF_NS:** El servidor y el cliente deben estar en una red XEROX.
- **Dominio AF_CCITT:** Para protocolos CCITT, protocolos X25, etc.

Esta característica se refiere al protocolo que se va a utilizar en la comunicación a través del socket, si se pone el valor 0 se toma el protocolo por defecto.

6.4.4. Proceso de Comunicación Cliente-Servidor

El proceso de comunicación cliente-servidor en modo conectado es utilizado por las aplicaciones estándar de Internet (telnet, ftp). Y se refiere al protocolo TCP. Las llamadas al sistema para crear la comunicación pueden separarse en dos partes: las llamadas del servidor y las llamadas del cliente.

6.4.4.1. Servidor

El servidor es el proceso que crea un socket con nombre y acepta las conexiones a él. El orden de las llamadas al sistema para la realización de esta función es la siguiente:

socket (dominio, tipo, protocolo)

Crea un socket sin nombre con un dominio, tipo y protocolo específico. Y con esto crea una estructura que contiene la información del socket, llamada sockaddr. Los valores que pueden tomar son los siguientes (Ver tabla 6.1):

ARGUMENTO	VALORES	SIGNIFICADO
Dominio	AF_INET, AF_UNIX	Protocolo o familia de direcciones
Tipo	SOCK__DGRAM, SOCK__STREAM	Tipo de servicio
Protocolo	0 (protocolo por defecto)	Número de protocolo a emplear o 0 para solicitar el puerto por defecto para un dominio y tipo dados

Tabla 6.1. Valores de la función socket.

La función socket regresa un descriptor de socket si no tiene problemas, y regresa el valor -1 si se produce algún tipo de error. Cuando se produce un error, la variable global errno contiene el código que especifica la causa del error. Los posibles errores se pueden ver en la tabla 6.2.

VALOR DE errno	CAUSA DEL ERROR
EPROTONOSUPPORT	Error en los argumentos: el protocolo especificado, o el servicio solicitado no es valido.
EMFILE	La tabla de descriptores de archivo de la aplicación está llena.
ENFILE	La tabla de archivos del sistema está llena.
EACCESS	Se ha denegado el permiso para crear el socket.
ENOBUFS	El sistema no dispone de espacio para el buffer.

Tabla 6.2. Errores de la función socket.

bind (descriptor-Server, sockaddr-Server, longDirecc)

La función bind, nombra un socket, es decir, asocia el socket no nombrado con el descriptor: descriptor-Server, con la dirección del socket almacenada en sockaddr-Server. La dirección depende de si estamos en un dominio AF_UNIX o AF_INET. (Ver tabla 6.3).

ARGUMENTO	SIGNIFICADO
Descriptor-server	Es el descriptor del socket.
Sockaddr-server	Es la dirección del socket en el servidor.
LongDirecc	Tamaño en bytes de la estructura dirección.

Tabla 6.3. Argumentos de la función bind.

La función de sistema bind retorna el valor 0 si realiza su labor sin problemas y -1 para indicar que se ha producido algún error. Cuando ocurre algún error, la variable global errno contiene un código que especifica la causa del error. Los posibles errores se pueden ver en la tabla 6.4.

VALOR DE errno	CAUSA DEL ERROR
EBADF	El argumento socket no especifica un descriptor válido.
ENOTSOCK	El argumento socket no especifica un descriptor de socket.
EADDRNOTAVAIL	La dirección especificada no es válida (por ejemplo, una dirección IP que no corresponde a ninguna interfaz local).
EADDRINUSE	La dirección especificada está en uso (por ejemplo, otro proceso ya ha reservado el puerto de Protocolo).
EINVAL	El socket ya tiene una dirección de red asignada.
ESCCES	La aplicación no tiene permisos para utilizar

	la dirección indicada
EFAULT	El puntero localaddr que se pasa como parámetro a la función no es válido.

Tabla 6.4. Errores de la función bind.

listen (descriptor-Server, longCola)

La llamada listen especifica el máximo número de peticiones de conexión pendientes. (Ver tabla 6.5).

ARGUMENTO	SIGNIFICADO
Descriptor-server	Es el descriptor del socket.
LongCola	Cantidad máxima de peticiones de conexión a ser encoladas (no suele pasar de 5).

Tabla 6.5. Argumentos de la función listen.

La llamada listen retorna 0 si puede llevar a cabo su labor sin problemas, retorna -1 en caso de que se produzca alguna situación de error. Cuando surge alguna situación de error, la variable global errno contiene uno de los siguientes valores (Ver tabla 6.6):

VALOR errno	CAUSA DEL ERROR
EBADF	El argumento socket no especifica un descriptor válido.
ENOTSOCK	El argumento socket no especifica un descriptor de socket.
EOPNOTSUPP	El tipo de socket no soporta llamadas a listen.

Tabla 6.6. Errores de la función listen.

Accept (descriptor-Server, sockaddr-Cliente, longDirecc-Cliente)

Esta función escucha al socket nombrado descriptor-Server y espera hasta que se reciba la petición de la conexión de un cliente. Al ocurrir esta incidencia, crea un socket sin nombre con las mismas

características que el socket servidor original, lo conecta al socket cliente y devuelve un descriptor que puede ser utilizado para la comunicación con el cliente. (Ver tabla 6.7).

ARGUMENTO	SIGNIFICADO
Descriptor-Server	Es el descriptor del socket del servidor.
Sockaddr-Cliente	Puntero a una estructura de dirección del socket en el cliente.
LongDirecc-Cliente	Puntero a un entero que especifica el tamaño en bytes del argumento sockaddr.

Tabla 6.7. Argumentos de la función accept.

Si todo va bien, la llamada accept retorna un descriptor socket con valor no negativo. Pero si falla retorna el valor -1 para indicar una situación de error. Cuando se produce un error, la variable global errno contiene uno de los siguientes valores (Ver tabla 6.8):

VALOR DE errno	CAUSA DEL ERROR
EBADF	El primer argumento no especifica un descriptor válido.
ENOTSOCK	El primer argumento no especifica un descriptor de socket válido.
EOPNOTSUPP	El socket no es del tipo SOCK_STREAM.
EFAULT	El puntero indicado en el segundo argumento no es válido.
EWOULDBLOCK	El socket esta marcado como de modo no bloqueante y no existen conexiones esperando a ser aceptadas, la llamada a accept provocaría el bloqueo del socket.

Tabla 6.8. Errores de la función accept.

6.4.5. *Funciones del Cliente*

Es el proceso encargado de crear un socket sin nombre y posteriormente enlazarlo con el socker servidor nombrado. El proceso

cliente es el que hace la solicitud de conexión al servidor. La secuencia de llamadas al sistema por el cliente es la siguiente:

socket (dominio, tipo, protocolo)

Al igual que el proceso servidor, el proceso cliente crea un socket sin nombre de un dominio, tipo y protocolo específico. Los argumentos utilizados en la declaración del socket son los mismos que se usaron en el socket del servidor.

connect (descriptor-Cliente, sockaddr-Server, longDirecc)

Intenta conectarse con un socket servidor cuya dirección se encuentra dentro de la estructura apuntada por sockaddr-Server. El descriptor: descriptor-Cliente, se utilizará para establecer la comunicación con el socket servidor. El tipo de estructura dependerá del dominio en que nos encontremos. (Ver tabla 6.9).

ARGUMENTO	SIGNIFICADO
Descriptor-Cliente	El descriptor de socket.
Sockaddr-Server	La dirección de red de la máquina remota.
LongDirecc	Tamaño en bytes del segunda argumento

Tabla 6.9. Argumentos de la función connect.

Si la función connect retorna el valor cero quiere decir que se puedo establecer la conexión y si retorna -1 quiere decir que se produjo algún error, en cuyo caso la variable global errno contendrá uno de los siguientes valores (Ver tabla 6.10):

VALOR DE errno	CAUSA DEL ERROR
EBADF	El primer argumento no especifica un descriptor válido.
ENOTSOCK	El primer argumento no especifica un descriptor de socket válido.
EAFNOSUPPORT	La familia de direcciones especificada en la dirección de red remota no puede ser

	utilizada con este tipo de socket.
EADDRNOTAVAIL	La dirección de red especificada no esta disponible.
EISCONN	El socket ya está conectado.
ETIMEDOUT	(Sólo TCP) El protocolo consumió su tiempo establecido sin conseguir establecer conexión con el servidor remoto.
ECONNREFUSED	(Sólo TCP) Conexión rechazada por la máquina remota.
ENETUNREACH	(Sólo TCP) No se puede conectar con la red.
EADDRINUSE	La dirección especificada ya está siendo utilizada.
EINPROGRESS	(Sólo TCP) El socket es de modo no bloqueante y un intento de conexión lo bloquearía.
EALREADY	(Sólo TCP) El socket es de modo no bloqueante y la llamada esperaría un intento de conexión previo para poder completarse.

Tabla 6.10. Errores de la función connect.

6.4.6. Cliente Servidor

El proceso servidor-cliente funciona de la siguiente manera: el servidor y el cliente crean un socket no nombrado independientemente. El servidor nombra al socket, es decir, le asigna un descriptor (el cual contiene la dirección del socket). Entonces el servidor se encuentra en estado de Listen, esperando que el cliente haga una petición de conexión. El cliente hace una petición de conexión y el servidor debe aceptarla, si la acepta se cierra entonces el socket del servidor y se establece el diálogo. Cuando el diálogo termina, se cierra el socket del cliente en ambos procesos, servidor y cliente. Este proceso se muestra con el siguiente pseudocódigo:

descrServer = socket (dominio, SOCK_STREAM, PROTOCOLO); (Crear el socket del Servidor)
descrClient = socket (dominio, SOCK_STREAM, PROTOCOLO); (Crear el socket del cliente)
bind (descriptor-Server, longServer); (Se nombra el socket)
do {
Listen (descrServer, longCola); (el servidor escucha las peticiones del cliente)
result = connect (descrClient, PuntSockServer, longserver); (el cliente trata de establecer la conexión)
DescrClient = accept (descrServer, PuntSockClient, longClient); (el servidor acepta la conexión)
} while (result == -1)
close (descrServer);
< DIALOGO >
Close (descrClient);

NOTA.- Para reafirmar algunos de los conceptos adquiridos en la teoría sobre la Capa de Transporte del Modelo OSI; se recomienda hacer la siguiente práctica contenida dentro de nuestra tesis:

Práctica 7 TCP y UDP

En donde el alumno al finalizar la práctica:

Podrá utilizar un programa que le permita enviar y recibir información utilizando los protocolos TCP y UDP y reafirmando conceptos teóricos. Podrá solicitar una página web a un determinado servidor desde un navegador.Tendrá la capacidad de conectarse al servidor SSH local haciéndole una petición de enviar la versión de éste. Será capaz de crear un socket servidor y un socket cliente.

Capítulo 7

Capa de Sesión

Capítulo 7
Capa de Sesión

7.1. Capa de Sesión

La capa de sesión organiza y sincroniza el intercambio de datos entre procesos de aplicación. Funciona con la capa de aplicación para proporcionar conjuntos de datos sencillos llamados puntos de sincronización, que le permiten a una aplicación conocer cómo está progresando la transmisión y recepción de datos. En términos sencillos, la capa de sesión puede pensarse como una capa cronometradora y controladora del flujo.

La capa de sesión interviene en la coordinación de las comunicaciones entre aplicaciones diferentes, permitiendo a cada una conocer el estado de la otra. Un error en una aplicación (sea en la misma máquina o al otro lado del país) lo maneja la capa de sesión para permitirle a la aplicación receptora saber que ha ocurrido un error. La capa de sesión puede resincronizar aplicaciones que están conectadas entre sí en ese momento. Esto puede ser necesario cuando comunicaciones se interrumpen en forma temporal o cuando ha ocurrido un error que resulta en la pérdida de datos.

El control de los diálogos entre distintos nodos es competencia de la capa de sesión. Un diálogo es una conversación formal en la que dos nodos acuerdan un intercambio de datos.

La comunicación puede producirse en tres modos de diálogo:

- **Simple (Simplex).** Un nodo transmite de manera exclusiva mientras otro recibe de manera exclusiva.
- **Semidúplex (Half-duplex).** Un solo nodo puede transmitir en un momento dado, y los nodos se turnan para transmitir.
- **Dúplex total (Full-duplex).** Los nodos pueden transmitir y recibir simultáneamente. La comunicación dúplex total suele requerir un control de flujo que asegure que ninguno de los dispositivos envía datos a mayor velocidad de la que el otro dispositivo puede recibir.

Las sesiones permiten que los nodos se comuniquen de manera organizada. Cada sesión tiene tres fases:

1. **Establecimiento de la conexión.** Los nodos establecen contacto. Negocian las reglas de la comunicación incluyendo los protocolos utilizados y los parámetros de comunicación.
2. **Transferencia de datos.** Los nodos inician un diálogo para intercambiar datos.
3. **Liberación de la conexión.** Cuando los nodos no necesitan seguir comunicados, inician la liberación ordenada de la sesión.

Los pasos 1 y 3 representan una carga de trabajo adicional para el proceso de comunicación. Esta carga puede no ser deseable para comunicaciones breves. Por ejemplo, considere la comunicación necesaria para una tarea administrativa de la red. Cuando una red administra varios dispositivos, éstos envían periódicamente un breve informe de estado que suele constar de una sola trama. Si todos estos mensajes se enviaran como parte de una sesión formal, las fases de establecimiento y liberación de la conexión transmitirían más datos que los del propio mensaje.

En estas situaciones, se comunica sin conexión. El nodo emisor se limita a transmitir los datos dando por sentado que el receptor está disponible.

Una sesión con conexión es aconsejable cuando la comunicación es compleja. Imagine la transmisión de una gran cantidad de datos de un nodo a otro. Si no se utilizaran controles formales, un solo error durante la transferencia obligaría a enviar de nuevo todo el archivo. Una vez establecida la sesión, los nodos implicados pueden pactar un procedimiento de comprobación. Si se produce un error, el nodo emisor sólo debe retransmitir los datos enviados desde la última comprobación. El proceso de gestión de actividades complejas se denomina administración de actividad.

La capa de sesión es la encargada de establecer el enlace de comunicación o sesión entre las computadoras emisora y receptora. Esta capa también gestiona la sesión que se establece entre ambos nodos. (Ver figura 7.1).

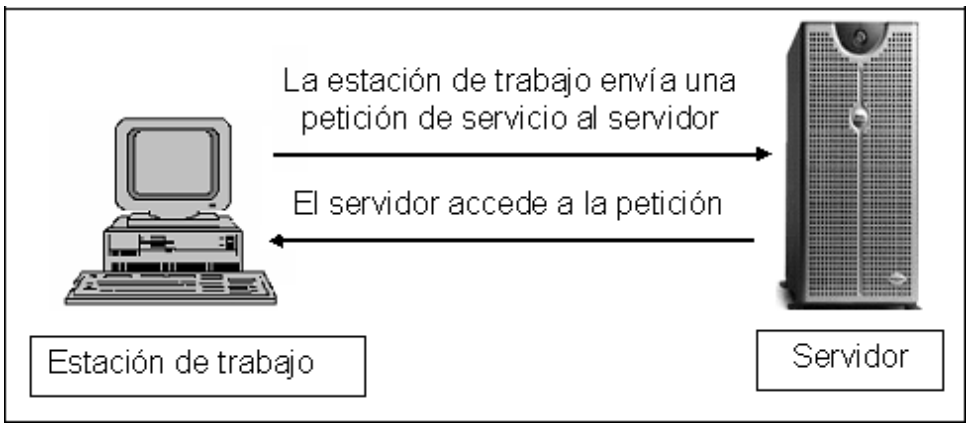


Figura 7.1 Comunicaciones en la capa de sesión.

La capa de sesión proporciona el enlace de comunicación entre dos computadoras que se están comunicando.

Una vez establecida la sesión entre los nodos participantes, la capa de sesión se encarga de ubicar puntos de control en la secuencia de datos. De esta forma, se proporciona cierta tolerancia a fallos dentro de la sesión de comunicación. Si una sesión falla y se pierde la comunicación entre los nodos, cuando después se restablezca la sesión sólo tendrán que volver a enviarse los datos situados detrás del último punto de control recibido. Así se evita el tener que enviar de nuevo todos los paquetes que incluía la sesión.

Los protocolos que operan en la capa de sesión pueden proporcionar dos tipos distintos de enfoques para que los datos vayan del emisor al receptor: La comunicación orientada a la conexión y la comunicación sin conexión.

Para comunicarse, los usuarios tienen que ejecutar el mismo conjunto de protocolos.

Por ejemplo se tomará el envío y recepción de un mensaje de correo electrónico, dando por sentado que tanto el remitente como el destinatario están ejecutando la misma pila de protocolos (la pila teórica OSI) en sus computadoras clientes. De hecho, las computadoras que ejecuten sistemas operativos distintos pueden comunicarse entre sí, si utilizan el mismo conjunto de protocolos de red. Esto es lo que explica que una máquina UNIX, un Macintosh o

una PC que esté ejecutando Windows utilicen TCP/IP para comunicarse en Internet. Un ejemplo en el que dos computadoras no podrían comunicarse, sería aquel en el que una computadora ejecutara TCP/IP y la otra IPX/SPX. Estos dos protocolos de red del mundo real utilizan reglas distintas y formatos de datos diferentes que hacen que la comunicación resulte imposible.

Los protocolos orientados a la conexión que operan en la capa de sesión proporcionan un entorno donde las computadoras conectadas se ponen de acuerdo sobre los parámetros relativos a la creación de los puntos de control en los datos, mantienen un diálogo durante la transferencia de los mismos, y después terminan de forma simultánea la sesión de transferencia.

Los protocolos orientados a la conexión operan de forma parecida a una llamada telefónica: en este caso, la sesión se establece con la persona que se desea hablar. La persona que llama y la que se encuentra al otro lado del teléfono mantienen una conexión directa. Y, cuando la conversación termina, ambos se ponen de acuerdo para dar por terminada la sesión y cuelgan el teléfono a la par.

El funcionamiento de los protocolos sin conexión, se parece más bien a un sistema de correo regular. Proporciona las direcciones pertinentes para el envío de los paquetes y éstos pasan a enviarse como si se echaran a un buzón de correos. Se supone que la dirección que incluyen permitirá que los paquetes lleguen a su destino, sin necesidad de un permiso previo de la computadora que va a recibirlos.

El TCP ejecuta funciones importantes en la capa de sesión. La capa de sesión proporciona sus servicios a la capa de presentación, proporcionando el medio necesario para que las entidades de presentación de dos host que se están comunicando por red organicen y sincronicen su diálogo y procedan al intercambio de datos.

Sus principales funciones son:

- Establecer, administrar y finalizar las sesiones entre dos hosts (máquinas en red) que se están comunicando.

- Si por algún motivo una sesión falla por cualquier causa ajena al usuario, restaurar la sesión a partir de un punto seguro y sin pérdida de datos o, si esto no es posible, terminar la sesión de una manera ordenada, chequeando y recuperando todas sus funciones, evitando así problemas en sistemas transaccionales.
- Sincronizar el diálogo entre las capas de presentación de los dos hosts y administrar su intercambio de datos, estableciendo las reglas o protocolos para él dialogo entre máquinas, regulando quien habla y por cuanto tiempo.
- Conseguir una transferencia de datos eficiente y un registro de excepciones acerca de los problemas de la capa de sesión, presentación y aplicación.
- Una sesión puede ser usada para efectuar un login a un sistema de tiempo compartido remoto, para transferir un archivo entre 2 máquinas, etc.
- Controla el diálogo (quién habla, cuándo, cuánto tiempo, half duplex o full duplex).
- Función de sincronización

7.2. *Uso de Puertos de Comunicación*

7.2.1. Puertos

Una computadora puede estar conectada con distintos servidores a la vez; por ejemplo, con un servidor de noticias y un servidor de correo. Para distinguir las distintas conexiones dentro de una misma computadora se utilizan los puertos.

Un puerto es un número de 16 bits, por lo que existen 65536 puertos en cada computadora. Las aplicaciones utilizan estos puertos para recibir y transmitir mensajes.

Los números de puerto de las aplicaciones cliente son asignados dinámicamente y generalmente son superiores al 1024. Cuando una aplicación cliente quiere comunicarse con un servidor, busca un número de puerto libre y lo utiliza.

En cambio, las aplicaciones servidoras utilizan números de puerto específicos: son los llamados puertos well-known (bien

conocidos). Estos puertos están definidos en la RFC 1700 y se pueden consultar en http://www.ietf.org/rfc/rfc1700.txt. A continuación se enumeran los puertos well-known más usuales: (Ver tabla 7.1).

Los puertos tienen una memoria intermedia *(buffer)* situada entre los programas de aplicación y la red. De tal forma que las aplicaciones transmiten la información a los puertos. Aquí se va almacenando hasta que pueda enviarse por la red. Una vez que pueda transmitirse, la información irá llegando al puerto destino donde se irá guardando hasta que la aplicación esté preparada para recibirla.

Palabra clave	**Puerto**	**Descripción**
	0/tcp	Reserved
	0/udp	Reserved
tcpmux	1/tcp	TCP Port Service Multiplexer
Rje	5/tcp	Remote Job Entry
Echo	7/tcp/udp	Echo
Discard	9/tcp/udp	Discard
systat	11/tcp/udp	Active Users
daytime	13/tcp/udp	Daytime
Qotd	17/tcp/udp	Quote of the Day
Chargen	19/tcp/udp	Character Generator
ftp-data	20/tcp	File Transfer [Default Data]
ftp	**21/tcp**	**File Transfer [Control]**
telnet	**23/tcp**	**Telnet**
smtp	**25/tcp**	**Simple Mail Transfer**
time	37/tcp/udp	Time
nameserver	42/tcp/udp	Host Name Server
nicname	43/tcp/udp	Who Is
domain	**53/tcp/udp**	**Domain Name Server**
Bootps	67/udp/udp	Bootstrap Protocol Server
Tftp	69/udp	Trivial File Transfer
gopher	70/tcp	Gopher
finger	79/tcp	Finger
www-http	**80/tcp**	**World Wide Web HTTP**
Dcp	93/tcp	Device Control Protocol
supdup	95/tcp	SUPDUP

hostname	101/tcp	NIC Host Name Server
Iso-tsap	102/tcp	ISO-TSAP
gppitnp	103/tcp	Genesis Point-to-Point Trans Net
Rtelnet	107/tcp/udp	Remote Telnet Service
pop2	109/tcp	Post Office Protocol – Version 2
pop3	**110/tcp**	**Post Office Protocol - Version 3**
Sunrpc	111/tcp/udp	SUN Remote Procedure Call
auth	113/tcp	Authentication Service
Sftp	115/tcp/udp	Simple File Transfer Protocol
nntp	**119/tcp**	**Network News Transfer Protocol**
Ntp	123/udp	Network Time Protocol
pwdgen	129/tcp	Password Generator Protocol
netbios-ns	137/tcp/udp	NETBIOS Name Service
netbios-dgm	138/tcp/udp	NETBIOS Datagram Service
netbios-ssn	**139/tcp/udp**	**NETBIOS Session Service**
snmp	161/udp	SNMP
snmptrap	162/udp	SNMPTRAP
Irc	194/tcp	Internet Relay Chat Protocol

Tabla 7.1. Puertos well-known más usuales.

7.2.2. Firewall

Un firewall es una herramienta que funciona como “cortafuegos” entre redes, permitiendo o denegando las transmisiones de una red a la otra. Un uso típico es situarlo entre una red local y la red Internet, como herramienta de seguridad para evitar que los intrusos puedan acceder a información confidencial.

Un firewall es simplemente un filtro que controla todas las comunicaciones que pasan de una red a la otra y en función de lo que sean permite o deniega su paso. Para permitir o denegar una comunicación el firewall examina el tipo de servicio al que corresponde, como pueden ser el web, el correo o el IRC. Dependiendo del servicio el firewall decide si lo permite o no. Además, el firewall examina si la comunicación es entrante o saliente y dependiendo de su dirección puede permitirla o no.

De este modo un firewall puede permitir desde una red local hacia Internet servicios de web, correo y ftp, pero no a IRC que puede

ser innecesario para nuestro trabajo. También podemos configurar los accesos que se hagan desde Internet hacia la red local y podemos denegarlos todos o permitir algunos servicios como el de la web, (si es que poseemos un servidor web y queremos que sea accesible desde Internet). Dependiendo del firewall que tengamos también podremos permitir algunos accesos a la red local desde Internet si el usuario se ha autenticado como usuario de la red local.

Un firewall puede ser un dispositivo software o hardware, es decir, un dispositivo que se conecta entre la red y el cable de la conexión a Internet, o bien un programa que se instala en la máquina que tiene el MODEM, que conecta con Internet. Incluso podemos encontrar computadoras muy potentes y con software específico que lo único que hacen es monitorizar las comunicaciones entre redes.

Dado que el firewall puede ser situado dentro de la capa de aplicación, transporte o red dependiendo del uso que se le de, es en este sentido que hemos empleado el firewall (software) dentro de esta capa, ya que para fines de la tesis el firewall a utilizar nos permite manipular los puertos de la computadora a modo de negar o permitir el uso de estos.

7.2.2.1. Tipos de peligros que puede evitar un Firewall

Algunos de estos pueden ser:

- Instalación y ejecución de programas en forma clandestina desde Internet, por ejemplo, vía aplicaciones Active X o Java que pueden llegar a transferir datos personales del usuario a sitios ajenos a él.
- Acceso de terceros por fallas o errores de configuración de windows (por ejemplo de NetBIOS).
- Instalación de publicidad (advertisers) o elementos de seguimiento (track) como las cookies.
- Troyanos: Aplicaciones ocultas que se descargan de la red y que pueden ser usadas por terceros para extraer datos personales. A diferencia de los virus, estos troyanos son activados en forma remota por un tercero.

- Reducción del ancho de banda disponible por el tráfico de *banners,* sitios no solicitados y otro tipo de datos innecesarios que vuelven lenta la conexión.
- Utilización de la línea telefónica por terceros por medio de *Dialers* (programas que cortan la actual conexión y utilizan la línea para llamadas de larga distancia).

7.3. *Llamada de Procedimientos Remotos (RPC)*

La capa de sesión tiene la responsabilidad de asegurar la entrega correcta de la información. Esta capa tiene que revisar que la información que recibe esté correcta; para esto, la capa de sesión debe realizar algunas funciones:

- La detección y corrección de errores.
- El controlar los diálogos entre dos entidades que se estén comunicando y definir los mecanismos para hacer las llamadas a procedimientos remotos (*Remote Procedure Control - RPC*).

El principal objetivo de la capa de sesión es la administración del diálogo y tratamiento de errores (por ejemplo, los relacionados con el fallo del sistema) que se generan por encima de la capa de transporte. En el Modelo de Referencia OSI, tiene poco sentido el concepto de sesión sin conexión. Sin embargo, ha habido mucha investigación, tanto en universidades como en industrias, sobre un modelo radicalmente diferente para el diálogo y el control de errores, basado en un modelo sin conexión. Este trabajo que lleva el nombre de **RPC (Llamada de Procedimiento Remoto)** se ha desarrollado extensamente en redes y (especialmente) en sistemas distribuidos.

La RPC no se adapta perfectamente al Modelo de Referencia OSI. Se ha diseñado para ser rápida y, por lo tanto, no contiene una estructura de múltiples capas. Lógicamente, la llamada de procedimientos remotos está interesada en aproximadamente los mismos temas que la capa de sesión (aunque desde una perspectiva diferente), razón por la cual se estudia en este capítulo. Sin embargo, la RPC también se puede realizar en la capa de aplicación (aunque de una manera menos eficiente), por lo que algunas veces se encontrarán estudios realizados al respecto en este contexto.

7.3.1. Modelo Cliente-Servidor

Un ejemplo común es una red de computadoras personales o estaciones de trabajo sin discos, llamadas **clientes**, que se están comunicando a través de una red con un **servidor de archivos** que tiene un disco sobre el cual se almacenan todos los archivos. En este sistema, los clientes acceden a sus datos mediante la transmisión de solicitudes al servidor, el cual se encarga de llevar a cabo el trabajo y devuelve respuestas. La comunicación siempre toma la forma de parejas de solicitud-respuesta, siempre iniciadas por el cliente y nunca por el servidor. A este modelo se le conoce con el nombre de modelo cliente-servidor, el cual se ilustra en la siguiente figura. (Ver figura 7.2.).

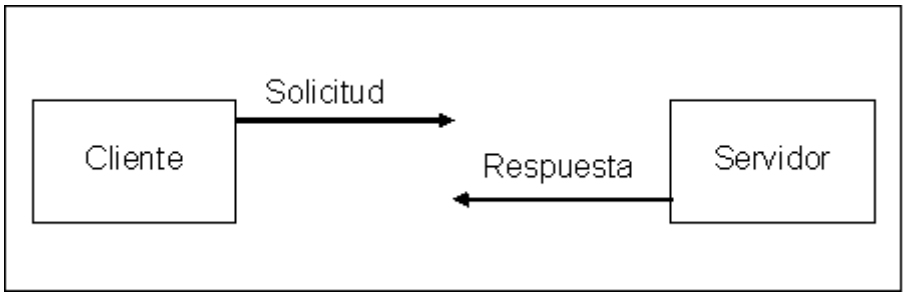


Figura 7.2. En el modelo cliente-servidor, el cliente transmite una solicitud y el servidor transmite una respuesta.

Mientras que resulta obvia la posibilidad de establecer sesiones entre clientes y servidores, y después utilizar comunicación semidúplex en esas sesiones, la gran sobrecarga ocasionada por múltiples capas de conexiones viene a ser, por lo general, poco atractiva para aplicaciones, como servidores de archivos, en donde el rendimiento es sumamente crítico. Una forma de comunicación totalmente sin conexión, construida justo encima del servicio de datagramas en bruto (especialmente en redes tipo LAN), resulta ser frecuentemente una mejor elección.

Aun cuando los problemas de rendimiento pueden resolverse mediante el empleo de un modo sin conexión, todavía existe un fallo importante en el modelo: la base conceptual de toda la comunicación es E/S (entrada/salida). Los programas se comunican con otros programas utilizando comandos como X.DATA.request y

X.DATA.indication, en donde el primero es una E/S, y el último una interrupción. Difícilmente vienen a ser las herramientas más apropiadas para construir aplicaciones bien estructuradas.

La escuela de pensamiento de la RPC desarrolla el modelo cliente-servidor desde una perspectiva completamente diferente. Desde este punto de vista, un cliente que transmite un mensaje a un servidor y obtiene una respuesta, es equivalente a un programa que llama a un procedimiento y obtiene un resultado. En ambos casos el que hace la llamada inicia una acción y espera hasta que se complete, y los resultados estén disponibles. Aunque en el caso ordinario (es decir, en el caso local), el procedimiento se ejecuta en la misma máquina del que hace la llamada, y con una RPC éste se ejecuta en una máquina diferente, el que hace la llamada no necesita preocuparse de esta diferencia.

7.3.2. *Realización de Llamadas de Procedimientos Remotos (RPC)*

En la siguiente figura (Ver figura 7.3.) se muestra una manera de realizar un sistema de llamadas de procedimientos remotos. En esta figura, la llamada remota toma diez pasos.

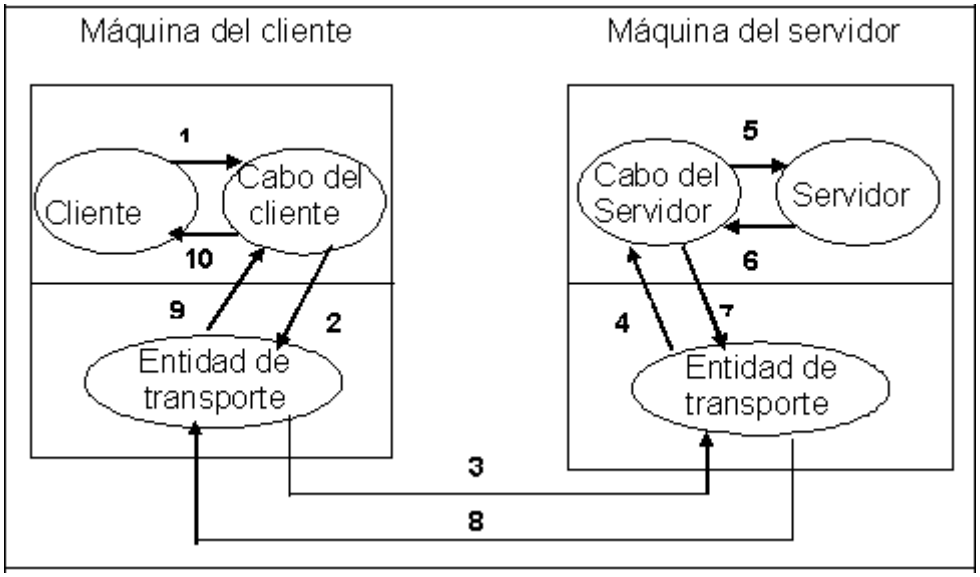


Figura 7.3. Los diez pasos necesarios para ejecutar una llamada de procedimiento remoto.

El paso 1 consiste en el programa (o procedimiento) cliente llamando al procedimiento cabo montado dentro de su propio espacio de direcciones. Los parámetros pueden pasarse de la manera normal. El cliente no nota ninguna cosa rara en esta llamada, porque se trata de una llamada normal de tipo local.

El procedimiento cabo del cliente, entonces, colecta los parámetros y los empaqueta en un mensaje. Esta operación se conoce como encapsulado de parámetros. Después de que el mensaje se haya construido, se pasa a la capa de transporte para su transmisión (paso 2). En un sistema de red tipo LAN sin conexión, la entidad de transporte probablemente sólo unirá una cabecera al mensaje y lo pondrá en la red (paso 3). En una red tipo LAN, la transmisión real puede ser más complicada. En muchos sistemas, el paso 2 es una llamada al sistema operativo.

Cuando el mensaje llega al servidor, la entidad de transporte lo pasa al procedimiento cabo del servidor (paso 4), el cual se encarga de desencapsular los parámetros. El procedimiento cabo del servidor entonces llama al procedimiento servidor (paso 5), pasando los parámetros en la forma normal. El procedimiento servidor no tiene ninguna manera de notar que está siendo activado a remoto, porque el que está haciendo la llamada, es un procedimiento local que obedece todas las reglas normales. Sólo el procedimiento cabo sabe que está sucediendo algo especial.

Después de que haya completado su trabajo, el procedimiento del servidor devuelve el control (paso 6), de la misma manera como cualquier otro procedimiento devuelve el control una vez que termina. También puede devolver un resultado al extremo que le hace la llamada. El procedimiento cabo del servidor, entonces, encapsula el resultado en un mensaje y lo entrega a la interfase de transporte (paso 7), posiblemente haciendo una llamada al sistema, como sucedió en el paso 2. Después de que la respuesta llega a la máquina del cliente (paso 8), se entrega al procedimiento cabo del cliente (paso 9). Por último, el procedimiento cabo del cliente devuelve el control al extremo que hace la llamada, es decir, al procedimiento cliente. Cualquier valor que devuelva el servidor en el paso 6, es entregado al cliente en el paso 10.

El propósito del mecanismo completo de la figura 7.3 es el de ofrecerle al procedimiento cliente la ilusión de que está haciendo una llamada directa al procedimiento servidor remoto. En la medida en que la ilusión sea exitosa y que el cliente no pueda decir que el servidor está remoto, se dice que el mecanismo es transparente.

NOTA.- Para reafirmar algunos de los conceptos adquiridos en la teoría sobre la Capa de Sesión del Modelo OSI; se recomienda hacer la siguiente práctica contenida dentro de nuestra tesis:

Práctica 8 Filtrados de Puertos TCP/UDP mediante un Firewall

En donde el alumno al finalizar la practica:

Se familiarizará con el manejo de los puertos, será capaz de manejar uno de los firewalls y creará filtros, permitiendo o negando el uso de puertos conocidos, entre otras cosas.

Capítulo 8

Capa de Presentación

Capítulo 8
Capa de Presentación

8.1. Capa de Presentación

La capa de presentación se responsabiliza de presentar los datos a la capa de aplicación. En ciertos casos, la capa de presentación traduce los datos directamente de un formato a otro. Las grandes computadoras IBM utilizan una codificación de caracteres denominada EBCDIC, mientras que las computadoras restantes utilizan el conjunto de caracteres ASCII. Por ejemplo, si se transmiten datos de una computadora EBCDIC a otra ASCII, la capa de presentación se encargara de traducir un conjunto de caracteres al otro. Además, la representación de los datos numéricos varía entre distintas arquitecturas de computadoras y debe convertirse cuando se transfieren datos de una máquina a otra.

Una técnica habitual para mejorar la transferencia de datos consiste en convertir todos los datos a un formato estándar antes de su transmisión. Puede que este formato estándar no sea el formato nativo de cualquier computadora, pero cualquiera de ellas puede configurarse para recibir datos en formato estándar y convertirlos en su formato nativo. Las normas OSI definen la Abstract Syntax Representation, Revision 1 (ASN.1 -Representación de sintaxis abstracta, revisión 1) como sintaxis estándar para los datos a nivel de la capa de presentación. Aunque el conjunto de protocolos TCP/IP no defina formalmente una capa de presentación, el protocolo External Data Representation (XDR -Representación de datos externos), utilizado por el sistema de archivos de red (NFS -Network File System), cumple una función similar.

Otras funciones que corresponden a la capa de presentación son la *encriptación/desencriptación* y *compresión/descompresión de datos*.

El papel principal de la capa de presentación es facilitar una comunicación segura entre dos hosts en el momento y después de la autenticación. La capa de presentación lleva esto a cabo manejando la encriptación y decodificación de datos, proporcionando protección de integridad de los paquetes de datos mientras son enviados y recibidos.

Además, la capa de presentación proporciona compresión de datos, lo que acelera la transmisión de información.

Al contactar un cliente a un servidor por medio del protocolo SSH, se negocian varios puntos importantes para que ambos sistemas puedan construir la capa de presentación correctamente. Durante el intercambio se producen los siguientes pasos:

- Intercambio de claves
- Se determina el algoritmo de encriptación de la clave pública
- Se determina el algoritmo de la encriptación simétrica
- Se determina el algoritmo autenticación de mensajes
- Se determina el algoritmo de hash que hay que usar

La capa de presentación es la que se implementa con menor frecuencia de las capas OSI. Se han definido pocos protocolos para esta capa. En la mayoría de los casos, las aplicaciones de red desempeñan las funciones asociadas con la capa de presentación.

8.2. Representaciones comunes de los Datos

8.2.1. ASCII 7 Bits

ASCII acrónimo de *American Standard Code for Information Interchange* (Código Normalizado Americano para el Intercambio de Información). En computación, un esquema de codificación que asigna valores numéricos a las letras, números, signos de puntuación y algunos otros caracteres. Al normalizar los valores utilizados para dichos caracteres, ASCII permite que las computadoras y programas informáticos intercambien información.

ASCII incluye 256 códigos divididos en dos conjuntos, estándar y extendido, de 128 cada uno. Estos conjuntos representan todas las combinaciones posibles de 7 u 8 bits, siendo esta última el número de bits en un byte. El conjunto ASCII básico, o estándar, utiliza 7 bits para cada código, lo que da como resultado 128 códigos de caracteres desde 0 hasta 127 (00H hasta 7FH hexadecimal).

En el conjunto de caracteres ASCII básico, los primeros 32 valores están asignados a los códigos de control de comunicaciones y de impresora —caracteres no imprimibles, como retroceso, retorno de carro y tabulación— empleados para controlar la forma en que la información es transferida desde una computadora a otra o desde una computadora a una impresora. Los 96 códigos restantes se asignan a los signos de puntuación corrientes, a los dígitos del 0 al 9 y a las letras mayúsculas y minúsculas del alfabeto latino.

8.2.2. ASCII 8 Bits

El conjunto ASCII extendido utiliza 8 bits para cada código, dando como resultado 128 códigos adicionales, numerados desde el 128 hasta el 255 (80H hasta FFH extendido).

Los códigos de ASCII extendido, del 128 al 255, se asignan a conjuntos de caracteres que varían según los fabricantes de computadoras y programadores de *software*. Estos códigos no son intercambiables entre los diferentes programas y computadoras como los caracteres ASCII estándar. Por ello, mientras que el conjunto de caracteres ASCII estándar es universal en el *hardware* y el *software* de las microcomputadoras, los caracteres ASCII extendido pueden interpretarse correctamente sólo si un programa, computadora o impresora han sido diseñados para ello.

8.2.3. Unicode

Básicamente Unicode proporciona un número único para cada carácter, sin importar la plataforma, ni el programa, ni el idioma, permitiendo un fácil traspaso entre distintos sistemas de codificación y plataformas.

Las computadoras sólo trabajan con números. Almacenan letras y otros caracteres mediante la asignación de un número a cada uno. Antes de que se inventara Unicode, existían cientos de sistemas de codificación distintos para asignar estos números. Ninguna codificación específica podía contener caracteres suficientes. Incluso para un solo idioma como el inglés, no había un único sistema de codificación que se adecuara a todas las letras, signos de puntuación y símbolos técnicos de uso común.

Además, estos sistemas de codificación presentan problemas entre ellos. Es decir, dos sistemas de codificación pueden utilizar el mismo número para dos caracteres distintos o bien utilizar números distintos para el mismo carácter. Toda computadora (especialmente los servidores) necesita ser compatible con muchos sistemas de codificación distintos; sin embargo, cada vez que los datos se traspasan entre distintos sistemas de codificación o plataformas, dichos datos siempre corren el riesgo de sufrir daños

Para ello, este método utiliza dos bytes por cada carácter. Cómo referencia, en el formato ASCII clásico es suficiente un solo byte para representar cada carácter. Esta mayor cantidad de espacio, normalmente está prevista por los programas y sistemas operativos que soportan esta codificación, y no debería representar problemas en circunstancias normales.

8.3. Compresión de Datos

8.3.1. Antecedentes

La primera razón para comprimir documentos es obvia: para que quepa en un espacio menor. Actualmente los discos duros tienen una gran capacidad de almacenamiento pudiendo guardar en ellos muchos datos de gran tamaño. Sin embargo, si queremos transportar los datos en otro medio (disquete, CD-ROM, entre otros) el espacio está más limitado, y tal vez no quepan los datos que necesitamos llevar con nosotros. Por ello hay que comprimirlos.

La segunda razón aparece fundamentalmente con Internet, y es el tiempo (que también se traduce en dinero) que necesitamos para acceder o descargar archivos de gran tamaño por la red. Por tanto, si esos datos están comprimidos tardaremos menos en enviarlos o recibirlos (por ejempl correo electrónico, web y ftp).

Lógicamente la compresión suele llevar asociada una descompresión, con la que podamos recuperar el formato original de esos datos para poder utilizarlos.

Una de las razones para no guardar los datos siempre comprimidos, es que algunos programas no se preocupan de que los datos que generan ocupen poco espacio ya que no están diseñados para ello, simplemente generan los archivos con los datos y ya está. Entonces para comprimir esos datos podemos utilizar unos algoritmos que sí están específicamente diseñados para hacer que esos datos generados por otros programas ocupen menos de lo que ocupan originalmente.

Esta compresión es posible porque normalmente en un archivo además de información hay también redundancia, es decir, datos que no aportan más información, en general porque pueden obtenerse a partir de los datos anteriores. Lo que suelen hacer los algoritmos de compresión es deshacerse de esa redundancia y optimizar así el espacio que ocupan los datos.

Otra de las razones es que para comprimir unos datos se requiere un tiempo de cálculo que no siempre estaremos dispuestos a esperar. Lo mismo ocurre cuando quiera ver o utilizar esos datos, ya que tendré que descomprimirlos para recuperar su formato original y eso vuelve a requerir más tiempo.

La compresión se puede hacer de dos formas:

- Manteniendo la integridad de los datos, es decir, que al descomprimir obtenga EXACTAMENTE lo que tenía en un principio, sin perder nada de información en el proceso.
- Sin mantener esa integridad, pudiendo perder ciertos datos que sean menos relevantes y que no podrán recuperarse más tarde al descomprimir.

Para comprimir los datos se utilizan distintos algoritmos, estando estos especialmente diseñados para cada tipo de dato. El resultado de la compresión será un archivo de menor tamaño (o igual, sino hemos conseguido comprimir nada) que el original y con un formato específico.

Todos los archivos tienen:

- Formato, normalmente definido por su extensión.

- Tamaño, normalmente expresado en bytes o Kb.

El formato del archivo (o su extensión) nos indican qué algoritmo se ha usado para comprimirlo. Hay programas que pueden comprimir y descomprimir distintos formatos de archivos (por ejemplo, WinZip puede manejar archivos del tipo Zip, TAR, gzip y CAB).

Podemos dividir los formatos de compresión (o algoritmos) según el tipo de datos que manejan. De esta forma tendremos formatos de compresión de archivos de datos como documentos, texto, programas, etc.; formatos de compresión de audio, de imágenes, de video…

8.3.2. Definición

Comprimir significa reducir el tamaño. Para comprimir archivos de datos, documentos o programas necesitamos una compresión SIN PÉRDIDAS de información, ya que cualquier alteración de la misma significará que al descomprimir no obtendremos el mismo resultado y seremos conscientes de ello (el documento no estará igual, o el programa no funcionará, etc).

El formato más extendido en entornos Windows es el ZIP, aunque hay muchos otros (TAR, gzip, CAB, ARJ, ARC, LZH, ISO, etc).

8.4. Encriptación

8.4.1. Bases de la Encriptación

Encriptación es una medida de seguridad utilizada para que al momento de almacenar y/o transmitir información sensible ésta no pueda ser obtenida con facilidad por terceros. La encriptación hace uso de diversas fórmulas matemáticas con el propósito de transformar el texto plano en un criptograma el cual es un conjunto de caracteres que a simple vista no tiene ningún sentido para el lector. La mayoría de los métodos de encriptación utilizan una clave como parámetro variable en las mencionadas fórmulas matemáticas de forma que a pesar de que un intruso las conozca, no le sea posible descifrar el criptograma si no conoce la clave, la cual solo se encuentra en

posesión de las personas que pueden tener acceso a la información en cuestión. Algunos métodos utilizan incluso dos claves, una privada que se utiliza para la encriptación y otra pública para la desencriptación.

SSH™ permite a los usuarios registrarse en sistemas de host remotamente. A diferencia de *FTP* o *Telnet*, *SSH* encripta la sesión de registro imposibilitando que alguien pueda obtener contraseñas no encriptadas.

SSH está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la shell de comando, tales como *telnet* o *rsh*. Un programa relacionado, el *scp*, reemplaza otros programas diseñados para copiar archivos entre hosts como *rcp*. El uso de métodos seguros para registrarse remotamente a otros sistemas hará disminuir los riesgos de seguridad tanto para el sistema cliente como para el sistema remoto.

8.4.2. Características de SSH

SSH (o *Secure SHell*) es un protocolo para crear conexiones seguras entre dos sistemas usando una arquitectura cliente/servidor.

El protocolo SSH proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- El cliente transmite su información de autenticación al servidor usando una encriptación robusta de 128 bits.
- Todos los datos enviados y recibidos durante la conexión se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremamente difícil de descifrar y leer.
- El cliente tiene la posibilidad de enviar aplicaciones X11 lanzadas desde el intérprete de comandos de la shell. Esta técnica proporciona una interfaz gráfica segura (llamada *reenvío por X11*) que proporciona un medio seguro para usar aplicaciones gráficas sobre una red.

Ya que el protocolo SSH encripta todo lo que envía y recibe, se puede usar para asegurar protocolos inseguros. El servidor SSH puede convertirse en un conducto para convertir en seguros los protocolos inseguros mediante el uso de una técnica llamada *reenvío por puerto*, como por ejemplo POP, incrementando la seguridad del sistema en general y de los datos.

Red Hat Linux contiene el paquete general de OpenSSH (openssh), el servidor de OpenSSH (openssh-server) y los paquetes de clientes (openssh-clients). Los paquetes OpenSSH requieren el paquete OpenSSL (openssl). OpenSSL instala varias librerías criptográficas importantes que ayudan a OpenSSH a proporcionar comunicaciones encriptadas.

Una gran cantidad de programas de cliente/servidor pueden usar el protocolo SSH. Muchas aplicaciones SSH cliente están disponibles para casi todos los principales sistemas operativos en uso hoy día.

8.4.3. Importancia de usar SSH

Los usuarios maliciosos tienen a su disposición una variedad de herramientas para interceptar y dirigir el tráfico de la red para ganar acceso al sistema. En términos generales, estas amenazas se pueden catalogar del siguiente modo:

- *Intercepción de la comunicación entre dos sistemas*: En este escenario, existe un tercero en algún lugar de la red entre entidades en comunicación que hace una copia de la información que pasa entre ellas. La parte interceptora puede interceptar y conservar la información, o puede modificar la información y luego enviarla al recipiente al cual estaba destinada. Este ataque se puede articular a través del uso de un paquete llamado sniffer — una utilidad de red muy común.
- *Personificación de un determinado host*: Con esta estrategia, un sistema interceptor finge ser el recipiente a quien está destinado un mensaje. Si funciona la estrategia, el cliente no se da cuenta del engaño y continúa la comunicación con el interceptor como si su mensaje hubiese llegado a su destino satisfactoriamente. Esto se produce con técnicas como el envenenamiento del DNS o spoofing de IP.

Ambas técnicas causan que se intercepte información, posiblemente con propósitos hostiles. El resultado puede ser catastrófico.

Si se utiliza SSH para inicios de sesión de shell remota y para copiar archivos, estas amenazas a la seguridad se pueden disminuir notablemente. Esto es porque el cliente SSH y el servidor usan firmas digitales para verificar su identidad. Adicionalmente, toda la comunicación entre los sistemas cliente y servidor es encriptada. No servirán de nada los intentos de falsificar la identidad de cualquiera de los dos lados de la comunicación ya que cada paquete está cifrado por medio de una clave conocida sólo por el sistema local y el remoto.

8.5. Autenticación

8.5.1. Kerberos

Internet es un lugar inseguro. Muchos de los protocolos utilizados actualmente carecen de seguridad. Además, existen crackers que con frecuencia interceptan contraseñas, razón por la cual aplicaciones que mandan una contraseña no encriptada en la red son extremadamente vulnerables. Peor aun, algunas aplicaciones cliente/servidor asumen que el cliente proveerá su identificación correctamente, y otras confían en que el cliente restringirá sus actividades a aquellas que están autorizadas sin ningún otro refuerzo del servidor.

Algunos sitios utilizan firewalls para solucionar los problemas de seguridad en redes. Pero los firewalls asumen que las personas que desean hacer daño están del lado de afuera, cuando en realidad esta aseveración suele ser falsa.

Kerberos es un personaje de la mitología griega que por ser quien cuidaba las puertas del infierno, representa seguridad. Se podría decir que como servicio de autenticación, ahora cuida las puertas de la red, impidiendo que entren personas indeseadas.

Kerberos es un servicio de autenticación desarrollado en MIT (Massachusetts Institute of Technology) y diseñado por Miller y

Neuman en el contexto del Proyecto Athena en 1987. Esta basado en el protocolo de distribución de claves presentado por Needham y Schroeder en 1978.

En una red con usuarios que solicitan servicios desde muchas terminales, hay tres enfoques básicos que se pueden utilizar para dar control de acceso:

1) No hacer nada. Confiar en que la máquina en la que el usuario está "en sesión" evite accesos no autorizados.
2) Requerir que el host pruebe su identidad, pero confiar en su palabra sobre quién es el usuario.
3) Requerir que el usuario pruebe su identidad para cada servicio solicitado.

En un ambiente cerrado, donde las máquinas tienen un control estricto, se podría utilizar el primer enfoque. En ambientes más abiertos, se podría usar la segunda opción. En ambientes como el de Athena, en el cual cada usuario debe probar su identidad a cada servicio que desee utilizar, el tercer enfoque es el más adecuado.

El entorno tendrá ciertos requerimientos sobre el sistema de identificación:

1) Seguridad. Un atacante no debe encontrar el punto débil en la autenticación. Alguien que observa la red no debe obtener la información necesaria para hacerse pasar por otro usuario.
2) Confiabilidad. El acceso a muchos servicios dependerá del servicio de autenticación y si éste no es confiable, el sistema de servicios tampoco lo será.
3) Transparencia. El usuario no debe percatarse de que se está autentificando.
4) Escalabilidad. El entorno no debe colapsar si aumenta el número de aplicaciones kerberizadas.

En una red de usuarios en la que se requieren servicios de distintas computadoras, si la red tiene miles de usuarios y decenas de servidores, no es deseable que cada servidor guarde las contraseñas de todos los usuarios. En ese caso, habría tantos puntos de ataque

como servidores. Es para evitar estos problemas que surge la idea de tener un servicio de autenticación.

Kerberos es el resultado de satisfacer estos requerimientos. La seguridad de Kerberos descansa en la seguridad de varios servidores de autenticación, pero no en el sistema que sé "loguea" o en los servidores finales que se utilicen.

8.5.1.1. Funcionalidad de Kerberos

Cada usuario tendrá una clave y cada servidor tendrá una clave, y Kerberos tiene una base de datos que las contendrá a todas. En el caso de ser un usuario, su clave será derivada de su contraseña y estará encriptada, mientras que en el caso del servidor, la clave se generará aleatoriamente. Los servicios de red que requieren autenticación y los usuarios que requieran estos servicios, se deben registrar con Kerberos. Las claves privadas se negocian cuando se registran.

Como Kerberos sabe todas las claves privadas, puede crear mensajes que convenzan a un servidor de que un usuario es realmente quien dice ser y viceversa. La otra función de Kerberos es generar las llamadas claves de sesión, que serán compartidas entre un cliente y un servidor, y nadie más. La clave de sesión podrá ser usada para encriptar mensajes que serán intercambiados entre ambas partes. El almacenamiento de la base de datos y la generación de claves, se lleva a cabo en un servidor que se denomina Servidor de Autenticación (AS por las siglas en inglés de Authentication Server). Kerberos provee tres niveles distintos de protección. El programador de la aplicación determinará cual es apropiado, de acuerdo a los requerimientos de la aplicación.

Autenticación:

Prueba que el usuario es quien dice ser. Puede ser que la autenticidad se establezca al inicio de la conexión de red y luego se asuma que los siguientes mensajes de una dirección de red determinada se originan desde la parte autenticada.

***Integridad de datos*:**

Asegura que los datos no se modifican en tránsito. Se requiere autenticación de cada mensaje, sin importar el contenido del mismo. Éstos se denominan mensajes seguros.

Privacidad de datos:

Asegura que los datos no son leídos en tránsito. En este caso no sólo se autentica cada mensaje sino que también sé encripta. Éstos son mensajes privados.

Hay tres fases básicas para el procedimiento de autenticación:

a) Solicitud de credenciales que se utilizarán al solicitar servicios.
b) Solicitud de credenciales para tener acceso a determinado servicio.
c) Presentación de credenciales al servidor deseado.

Hay dos tipos de credenciales que se utilizan en el modelo de autenticación de Kerberos: Tickets y Autenticadores. Aunque ambos se basan en encriptado de clave privada, sé encriptan con claves diferentes. El ticket se usa para pasarle al servidor final la identidad de la persona para la que fue emitido. El autenticador es una prueba de que el ticket fue creado para el usuario y no fue robado; contiene información que al ser comparada contra la que está en el ticket prueba que el usuario que lo presenta es el mismo al que le fue emitido.

Notación:

c = cliente
s = servidor
addr = dirección de red del cliente
K_x = Clave privada de x
$K_{x,y}$ = Clave de sesión de x e y
{info}K_x = Información encriptada con la clave privada de x
$T_{x,y}$ = Ticket de x para usar y
A_x = Autenticador para x

8.5.1.2. La Información en el Ticket

Un ticket sirve para un solo servidor y para un solo cliente pero una vez emitido, puede ser utilizado muchas veces por el cliente para tener acceso a ese servidor, hasta que el ticket expira. Como el ticket está encriptado con la clave del servidor no se pierde seguridad al permitir que el usuario le pase el ticket al servidor, ya que no podrá modificarlo.

1) El nombre del usuario es la información básica para pasarle al servidor. Así se le está informando quién está solicitando el servicio.
2) Si no estuviera el nombre del servidor, y solamente se enviara el nombre del usuario, entonces el servidor no sabría sí desencriptó bien la información. En cambio al ver su nombre en el ticket, se asegura que los datos son correctos.
3) Sin la dirección de red alguien podría copiar el ticket mientras viaja por la red, y luego, convencer a su estación de trabajo insegura de que el nombre de usuario del ladrón es el de la persona a quien se le está robando el ticket. Desde la estación envía el ticket al servidor, éste lo desencripta y ve que el nombre del que mandó el ticket es el mismo que está en el ticket, y por ende le da acceso. Agregar la dirección de red evita este problema.
4) El sello de tiempo se agrega para impedir lo que se llama ataques de “replay”. Esto ocurre cuando se roba un mensaje mientras viaja por la red y se intenta usar más tarde.
5) El tiempo de vida es el tiempo durante el cual el ticket es válido. Los tickets deben tener un tiempo de vida fijo porque si no expiraran nunca, representarían un riesgo de seguridad muy grande.
6) Aun con el sello de tiempo, es posible que un ataque de “replay” se lleve a cabo. Si alguien dejara su estación de trabajo antes de que el ticket expirara, sería posible que éste fuera utilizado durante el tiempo que resta. Con la clave de sesión, las partes interesadas comparten un secreto y se impide a través del autenticador que pueda ocurrir un ataque de “replay”.

Como un ticket sirve para un solo servidor, es necesario obtener un nuevo ticket para cada servicio que el usuario quiera utilizar. Los tickets para los distintos servidores se pueden obtener a través del TGS.

La idea del TGS surge para evitar que el usuario deba entrar su contraseña más de una vez. Aunque pueden convivir en la misma máquina, el TGS es lógicamente distinto del AS. A veces se refiere a ambos como KDC ("Key Distribution Center").

A diferencia del ticket, el autenticador sólo se puede utilizar una vez. De lo contrario, sería posible robar el ticket y el autenticador juntos y hacer el "replay" de todas formas. Se debe generar uno nuevo cada vez que el cliente quiere utilizar un servicio, pero esto no presenta un problema porque como el cliente tiene toda la información necesaria, puede construir el autenticador por sí mismo.

Cuando el cliente tiene el ticket para el servidor deseado lo envía al servidor junto con el autenticador. Una vez que el servidor recibió el ticket y el autenticador, desencripta el ticket con su clave privada. Si éste no expiró usa la clave de sesión que se encuentra dentro para desencriptar el autenticador y compara la información del ticket con la del autenticador. Si todo está bien, permite el acceso porque después de este intercambio, el servidor está seguro que de acuerdo a Kerberos, el usuario es quien dice ser.

Se asume que los relojes están sincronizados. Si el tiempo de la solicitud está muy lejos en el futuro o en el pasado, el servidor considera que la solicitud es un intento de hacer un "replay" de una solicitud anterior. Además, el servidor mantiene un registro de todas las solicitudes pasadas con sellos de tiempo aún válidos y si recibe una solicitud con el mismo ticket y el mismo sello de tiempo que una anterior, la descarta.

8.5.1.3. Autenticación mutua

Finalmente, el cliente puede querer que el servidor pruebe su identidad también. Entonces, el servidor le suma uno al sello de

tiempo que el cliente envió en el autenticador, encripta el resultado con la clave de sesión y envía el resultado de vuelta al cliente.

$$\{timestamp +1\}K_{c,s}$$

Esto demuestra que el servidor pudo leer el sello de tiempo del autenticador y para ello debió obtener la clave de sesión que se encontraba en el ticket encriptado con su clave privada. Entonces el cliente se cerciora de que el servidor es auténtico. Además, el cliente y el servidor comparten un secreto que nadie más sabe, y pueden asumir que un mensaje relativamente reciente encriptado con esa clave fue originado por el otro.

Para el usuario la presencia de Kerberos es casi imperceptible. El TGT se obtiene como parte del proceso de "login". Y los tickets son destruidos automáticamente cuando el usuario termina la sesión. El usuario notará la presencia de Kerberos sí su sesión dura más de 8 horas. En ese caso, si intenta acceder a algún servicio que utilice autenticación por medio de Kerberos el intento fallará porque el ticket habrá expirado.

8.5.1.4. Administración

Para el administrador del sistema, la presencia de Kerberos implica cierto mantenimiento. El administrador debe inicializar la base de datos de Kerberos al instalarlo. Además en caso de tener la base de datos replicada (con slaves), también será necesario administrarlas para que se mantengan consistentes. Esto implica tener que sincronizar las bases de datos replicadas con la base de datos principal cada cierto tiempo.

Se debe tener especial cuidado con la administración de las máquinas que tienen la base de datos Kerberos, pues el sistema podría volverse inseguro si alguien obtiene el control de alguna de estas máquinas.

El administrador también debe asegurarse de que la fecha y hora de todas las máquinas del sistema estén medianamente sincronizadas. Cuando una aplicación Kerberos es agregada al sistema, el administrador debe realizar varias operaciones para dejarla

funcional. El servidor kerberizado debe ser registrado en la base de datos, y se le debe asignar una clave privada. Luego algunos datos deben ser transferidos al nuevo servidor.

Además se aconseja tener la base de datos principal resguardada, en caso de que el disco en el que reside falle.

Cada sistema Kerberos contiene como parte de su nombre el realm (dominio) asociado. Esto permite que distintos sistemas Kerberos se comuniquen entre sí para brindarse servicios. Cuando un usuario se autentica con un sistema Kerberos, lo hace a un realm específico. Las aplicaciones kerberizadas generalmente reconocen credenciales de un realm particular; pero los usuarios pueden obtener credenciales de otros realms distintos del realm local. Para ello deben estar autenticados con el sistema Kerberos local, y los dos sistemas Kerberos (el local y el remoto) deben ponerse de acuerdo para brindar las credenciales necesarias en el sistema Kerberos remoto.

Los administradores de cada par de sistemas Kerberos que desean cooperar deben ponerse de acuerdo en la clave privada a utilizar para que los sistemas se comuniquen entre ellos.

Kerberos soporta la transitividad de la confianza; la idea es que si un sistema Kerberos confía en un segundo, y ese segundo confía en un tercer sistema Kerberos, entonces hay automáticamente una relación de confianza entre el primer y el tercer realm. La transitividad de la confianza asegura que los sistemas Kerberos sólo necesitan compartir una contraseña con los sistemas Kerberos de los realms inmediatamente encima y debajo de ellos en la jerarquía de realms (Kerberos se encarga del resto).

Kerberos también presenta ciertas limitaciones:

- Kerberos no tiene efectividad frente a ataques como el de diccionario. Si el usuario escoge una contraseña pobre, un atacante que la consiga tratando de adivinarla puede hacerse pasar por él.
- Kerberos requiere un camino seguro para entrar las contraseñas. Si un usuario entra una contraseña a un programa que ha sido modificado por un atacante, o si la

comunicación entre el usuario y el programa de autenticación inicial puede ser monitoreada, entonces el atacante puede llegar a obtener suficiente información para hacerse pasar por el usuario. Kerberos puede combinarse con otras técnicas para eliminar estas limitaciones.

- No hay un lugar seguro donde guardar las claves de sesión. De hecho, el lugar donde se guardan puede ser accedido por el root. Así es que un intruso que logre crackear el mecanismo de protección de la computadora local podrá robar las claves de sesión.
- El protocolo liga los tickets a las direcciones de red y esto es un problema en hosts con más de una dirección IP. En las estaciones de trabajo esto rara vez ocurre.
- Para que Kerberos sea útil debe integrarse con otras partes del sistema. No protege todos los mensajes que se envían entre dos computadoras. Sólo protege los mensajes desde el software que se ha escrito o modificado para usarlo. Aunque puede ser utilizado para intercambiar claves de encriptado cuando se establece un vínculo y niveles de seguridad de servicios de red, esto requeriría cambios en el software de los hosts involucrados.
- Tiempo de vida de un ticket. La elección del tiempo de vida de los tickets no es trivial. Si se elige un tiempo de vida para los tickets muy largo, y un usuario desprevenido olvida salir de sesión de una máquina, otra persona puede tomar su lugar. Por otro lado, si el tiempo de vida de los tickets es muy corto, el usuario va a ser molestado cada cierto tiempo para que ingrese nuevamente su contraseña.
- Manejo de proxies. Todavía no esta claro cómo permitir autenticación mediante proxies, es decir que un servidor utilice servicios de otros servidores en nombre de un usuario autenticado. Una solución tentativa es la de permitir que un usuario transfiera momentáneamente sus credenciales hacia otra máquina para poder obtener servicios de ella; pero esto no siempre es deseable pues el usuario podría no confiar en la máquina remota.

NOTA.- Para reafirmar algunos de los conceptos adquiridos en la teoría sobre la Capa de Presentación del Modelo OSI; se recomienda hacer la siguiente práctica contenida dentro de nuestra tesis:

Práctica 9 SSH: Secure Shell

En donde el alumno al finalizar la práctica:

Al finalizar la práctica, conocerá la importancia de utilizar el protocolo SSH (Secure Shell) y su herramienta OpenSSH.

Será capaz de iniciar una sesión remota a través de SSH, utilizando autenticación por contraseña.

Iniciará una sesión remota con clave pública, generando las claves.

Podrá transferir claves públicas al servidor.

Capítulo 9

Capa de Aplicación

Capítulo 9
Capa de Aplicación

9.1. Capa de Aplicación

La capa de aplicación proporciona los servicios utilizados por las aplicaciones para que los usuarios se comuniquen a través de la red. Algunos ejemplos de servicios, son:

- **Transporte de correo electrónico.** Gran variedad de aplicaciones pueden utilizar un protocolo para gestionar el correo electrónico. Los diseñadores de aplicaciones que recurren al correo electrónico no necesitan desarrollar sus propios programas para gestionar el correo. Además, las aplicaciones que comparten una misma interfaz de correo pueden intercambiar mensajes utilizando el gestor de correo electrónico.
- **Acceso a archivos remotos.** Las aplicaciones locales pueden acceder a los archivos ubicados en los nodos remotos.
- **Ejecución de tareas remotas.** Las aplicaciones locales pueden iniciar y controlar procesos en otros nodos.
- **Directorios.** La red puede ofrecer un directorio de recursos, incluyendo nombres de nodos lógicos. El directorio permite que las aplicaciones accedan a los recursos de la red utilizando nombres lógicos en lugar de identificaciones numéricas abstractas.
- **Administración de la red.** Los protocolos de administración de la red permiten que varias aplicaciones puedan acceder a la información administrativa de la red.

Es frecuente encontrar el término interfaz de programa de aplicación (API) asociado a los servicios de la capa de aplicación. Un API es un conjunto de reglas que permiten que las aplicaciones escritas por los usuarios puedan acceder a los servicios de un sistema de software. Los diseñadores de programas y protocolos suelen proporcionar varias API para que los programadores puedan adaptar fácilmente sus aplicaciones y utilizar los servicios disponibles en sus

productos. Un API habitual de UNIX es Berkeley Sockets; Microsoft lo ha implementado denominándolo Windows Sockets.

Llamamos ***procesos de aplicación*** a los ***programas/software*** que realizan las funciones de proceso reales asociadas a una aplicación (distribuida). Por ejemplo, un *proceso de aplicación* ***(AP)*** cliente puede ser un programa que se ejecuta en una computadora y que necesita tener acceso a un AP servidor de archivos que se ejecuta en otra computadora. Los protocolos de aplicación específicos proporcionan el soporte necesario para que un cliente pueda tener acceso a los distintos AP servidores en forma abierta, como si se estuviera ejecutando en la misma máquina que el servidor.

En el caso de un conjunto OSI, el enfoque consiste en definir, para cada servicio de aplicación distribuida, un ***dispositivo virtual*** junto con un conjunto definido de sus primitivas de servicio de usuario. Hay un almacén/servidor de archivos virtual, un servidor de correo virtual, etc. El protocolo de aplicación asociado a un servicio específico se implementa suponiendo que los AP cliente y servidor se comunican mediante primitivas iguales a las asociadas al dispositivo virtual definido.

Así, todas las primitivas de solicitud y respuesta en la interfaz con cada protocolo de aplicación *y por tanto en la interfaz con el entorno de interconexión de sistemas abiertos* tienen un formato estándar. Además, cada protocolo de aplicación opera en una forma definida que es independiente de las diferencias que pudiera haber en el modo de operación de los respectivos AP reales. Si los servicios asociados a los AP son diferentes de los asociados al dispositivo virtual, se añade otra capa de software dependiente del dispositivo entre cada AP y el protocolo de aplicación correspondiente para realizar todas las funciones de transformación necesarias para convertir las primitivas de servicio asociadas al dispositivo virtual a las que se usan con el dispositivo real y viceversa. De este modo, es posible utilizar el software de aplicación y los servidores existentes sin tener que modificarlos. El software de transformación se denomina ***elemento de usuario (UE: user element) o agente de usuario (UA: user agent).***

En la práctica, el UE se implementa ya sea como un proceso aparte o en forma de un conjunto de procedimientos (o funciones) de biblioteca que se enlazan al AP usuario. Casi todos los servicios tienen un AP iniciador (y por tanto una entidad de aplicación a la que se dice que están conectados) y un AP respondedor. El primero actúa como iniciador (u originador) de las solicitudes del dispositivo virtual y el segundo actúa como el respondedor (o destinatario) de dichas solicitudes. También, si la sintaxis de los datos que se intercambian dos AP usuarios es diferente, la entidad de presentación realiza la transformación de la sintaxis de transferencia (concreta) convenida a la sintaxis local, según sea necesario. El o los contextos de presentación específicos que se usarán se negocian en el momento en que se inicia el establecimiento de la asociación (conexión).

9.2. Protocolos de Aplicación del Modelo OSI

9.2.1. TELNET

Permite al usuario de una terminal (o a un AP usuario) de una máquina comunicarse interactivamente con un AP, por ejemplo, un editor de textos que se ejecuta en una máquina remota, como si la terminal de usuario estuviera conectada a él directamente.

9.2.2. *FTP (Protocolo de Transferencia de Archivos)*

Permite al usuario de una terminal (o a un AP usuario) tener acceso a un sistema de archivo remoto e interactuar con él.

9.2.3. SMTP (Protocolo Simple de Transferencia de Correo)

Ofrece un servicio de transferencia de correo a nivel de toda la red entre los sistemas de correo asociados a diferentes máquinas.

9.2.4. *SNMP (Protocolo Simple de Gestión de Red)*

Permite un usuario (por ejemplo, un gestor de red) recabar datos de rendimiento o controlar el funcionamiento de un elemento de red (por ejemplo, un puente o una pasarela) a través de la red misma.

Todas las interacciones cliente-servidor deben cumplir el requisito de establecer un camino de comunicación entre los dos protocolos/procesos de aplicación implicados; por tanto, antes de estudiar los diferentes protocolos de aplicación, examinemos cómo se establece un camino de comunicación.

9.3. Establecimiento de una Conexión de Transporte

Todos los AP servidores tienen asociado un nombre que se traduce en una dirección de red correspondiente. El procedimiento de traducción lo efectúa un proceso llamado ***servidor de nombres de dominio.*** La dirección de red resultante de un proceso servidor consta de dos partes: ***La dirección IP*** a nivel de red del sistema anfitrión en el que se está ejecutando el proceso y un ***número de puerto local.***

Un sistema abierto contiene múltiples clientes y servidores tanto de tipos distintos (FTP, SMTP, etc.) como del mismo tipo (varios servidores de archivo, servidores de correo, etc.). No obstante, a todos los servidores del mismo tipo se les asigna el mismo número de puerto en todo el sistema, de modo que un servidor específico es identificado por la dirección de IP del sistema anfitrión en el que se ejecuta. Los números de puerto asociados a los diferentes tipos de servicios se denominan ***puertos bien conocidos*****,** entre los cuales tenemos e incluyen los siguientes:

PUERTO	SERVICIOS
21	FTP
22	TELNET
25	SMTP

Así, cuando un proceso cliente inicia una llamada a un proceso servidor correspondiente, emplea como dirección de destino la dirección de IP del sistema anfitrión en el que el servidor se está ejecutando junto con el número de puerto bien conocido de ese servidor. Como dirección de origen el cliente utiliza la dirección de IP de su propio sistema anfitrión en conjunción con el siguiente número de puerto libre (no utilizado) de ese sistema anfitrión.

Los tres protocolos de aplicación TELNET, FTP y SMTP se ocupan de proporcionar servicios de aplicación de usuario en toda la red.

En contraste:

El protocolo simple de gestión de red (SNMP simple network management protocol) no se ocupa de servicios de usuario, sino más bien de la gestión de todos los protocolos de comunicación dentro de cada sistema anfitrión y de los diversos elementos de equipos de red que proveen estos servicios; en otras palabras, *del entorno de red total.*

Un entorno de redes de sistema abierto por lo regular comprende una diversidad de elementos de equipos de red diferentes: puentes para la interconexión de segmentos LAN similares, enrutadores para interconectar LAN disímiles, conmutadores de paquetes para las redes de conmutación de paquetes, diversos tipos de pasarelas de internet para interconectar redes, los enlaces de comunicación que conectan estos equipos, etcétera.

Por supuesto, si en cualquier entorno de red ocurre un fallo y se interrumpe el servicio, los usuarios esperarán que el fallo se corrija y se restablezca el servicio normal con un mínimo de retardo. A esta función suele llamársele *gestión de fallos***.** En forma análoga, si el rendimiento de la red por ejemplo, su tiempo de respuesta o su volumen de transmisión comienza a menguar como consecuencia de, digamos, un aumento en el nivel de tráfico en algunas regiones de la red, los usuarios esperarán que dichas regiones se identifiquen y se introduzca equipo (capacidad de transmisión adicional para aliviar el problema). Éste es un ejemplo de *gestión de rendimiento***.** El SNMP se definió para ayudar a un gestor de red a realizar las funciones de gestión de fallos y de rendimiento.

El enfoque de gestión de red estándar consiste en considerar todos los elementos de red que se deben administrar -protocolos, puentes, pasarelas, etc.- como *objetos gestionados***.** A cada objeto gestionado se asocia un conjunto definido de información relacionada con la gestión, la cual incluye variables; también llamadas *atributos* que el gestor de red puede leer o modificar a través de la red. Dicha

información contiene también, cuando sea apropiado, un conjunto de *informes de fallos* que un objeto gestionado envía cuando ocurre un fallo relacionado con él. En el caso del IP, por ejemplo, una variable de lectura podría tener que ver con, digamos, el número de datagramas de IP desechados cuando expira el parámetro de tiempo de vida, mientras que una variable de escritura podría ser el valor real del tiempo de vida que se usará. El SNMP es un protocolo de aplicación, así que debemos utilizar una plataforma de comunicación estándar para hacer posible la transferencia de los mensajes -PDU- asociados al mismo tiempo que se transfieren los mensajes relacionados con los servicios de usuario.

El papel del SNMP es permitir que el **proceso gestor** de la estación gestora intercambie mensajes relacionados con la gestión con los procesos de gestión que se ejecutan en los diversos elementos gestionados: sistemas anfitriones, pasarelas, etc. El proceso de gestión de estos elementos realiza las funciones administrativas asociadas a ellos; por ejemplo, responder a solicitudes de variables (contadores) especificadas, recibir variables operativas actualizadas y generar y enviar informes de fallo.

9.4. HTTP Hyper Text Transfer Protocol (Protocolo para la Transferencia de Hipertextos)

El protocolo HTTP se usa en los sistemas de información distribuidos que necesitan mostrar la información y pasarla por una comunicación normal haciendo uso de las ligas de este lenguaje ***articulando los intercambios de información entre los clientes Web y los servidores HTTP***.

El Protocolo fue implementado inicialmente por Tim Berners-Lee para WWW en 1991 como un protocolo rápido y sencillo que permite la transferencia de múltiples tipos de información de forma eficiente y rápida. Se denominó HTTP 0.9. El protocolo completo fue definido en 1992 e implementado en marzo de 1993.

Está soportado *sobre los servicios de conexión TCP/IP*, y funciona de la misma forma como los servicios comunes de los entornos UNIX: un proceso servidor escucha en un puerto de comunicaciones TCP (por defecto, el 80), y espera las solicitudes de

conexión de los clientes Web. Una vez que se establece la conexión, el protocolo TCP se encarga de mantener la comunicación y garantizar un intercambio de datos libre de errores.

HTTP se basa en sencillas operaciones de solicitud/respuesta. Un cliente establece una conexión con un servidor y envía un mensaje con los datos de la solicitud. El servidor responde con un mensaje similar, que contiene el estado de la operación y su posible resultado. Todas las operaciones ***pueden adjuntar un objeto o recurso*** sobre el que actúan. Estos objetos Web (documento HTML, archivo multimedia o aplicación CGI) es conocido por su URL y están clasificados por su descripción MIME (descargando al protocolo de este aspecto).

9.4.1. Características Principales

Algunas características principales se listan a continuación:

- Toda la comunicación entre los clientes y servidores se realiza a partir de caracteres de 8 bits. De esta forma, se puede transmitir cualquier tipo de documento: texto, binario, etc., respetando su formato original.
- Permite la transferencia de objetos multimedia. El contenido de cada objeto intercambiado está identificado por su clasificación MIME.
- Existen tres verbos básicos (hay más, pero por lo general no se utilizan) que un cliente puede utilizar para dialogar con el servidor: GET, para recoger un objeto; POST, para enviar información al servidor y HEAD, para solicitar las características de un objeto.
- Cada operación HTTP implica una conexión con el servidor, que es liberada al término de la misma. Es decir, en una operación se puede recoger un único objeto.
- No mantiene estado. Cada petición de un cliente a un servidor no es influida por las transacciones anteriores. El servidor trata cada petición como una operación totalmente independiente del resto.

- Cada objeto al que se aplican los verbos del protocolo está identificado a través de la información de situación del final de la URL.

9.4.2. *Etapas de una Transacción HTTP*

Cada vez que un cliente realiza una petición a un servidor HTTP, se ejecutan los siguientes pasos:

1. Un usuario accede a una URL, seleccionando un enlace de un documento HTML o introduciéndola directamente en el campo Location del cliente Web.
2. El cliente Web descodifica la URL, separando sus diferentes partes. Así identifica el protocolo de acceso, la dirección DNS o IP del servidor, el posible puerto opcional (el valor por defecto es 80) y el objeto requerido del servidor.
3. Se abre una conexión TCP/IP con el servidor, llamando al puerto TCP correspondiente.
4. Se realiza la petición. Para ello, se envía el comando necesario (GET, POST, HEAD,…), la dirección del objeto requerido (el contenido de la URL que sigue a la dirección del servidor), la versión del protocolo HTTP empleada (casi siempre HTTP/1.0) y un conjunto variable de información, que incluye datos sobre las capacidades del browser, datos opcionales para el servidor,…
5. El servidor devuelve la respuesta al cliente. Consiste en un código de estado y el tipo de dato MIME de la información de retorno, seguido de la propia información.
6. Se cierra la conexión TCP.

De forma natural al solicitar una página HTML se establecerían tantas conexiones como elementos incluyese (imágenes, sonidos, ...) además de la de la propia página. Últimamente se ha mejorado el proceso permitiendo que una misma conexión se mantenga activa de forma que sea utilizada en sucesivas transacciones (HTTP Keep Alive).

9.4.3. Estructura de los Mensajes HTTP

El diálogo con los servidores HTTP se establece a través de mensajes formados por líneas de texto, cada una de las cuales contiene los diferentes comandos y opciones del protocolo. Sólo existen dos tipos de mensajes, uno para realizar peticiones y otro para devolver la correspondiente respuesta. La estructura general de estos dos tipos de mensajes se puede ver en el siguiente esquema (organizado por grupos de líneas) (Ver tabla 9.1):

La separación entre cada línea del mensaje se realiza con un par CR-LF (retorno de carro más nueva línea). El final de las cabeceras se indica con una línea en blanco, tras la cual se pueden incluir los datos transportados por el protocolo, por ejemplo, el documento HTML que devuelve un servidor o el contenido de un formulario que envía un cliente.

	Mensaje de solicitud		***Mensaje de respuesta***	
1	Comando HTTP + parámetros.	Comando que se solicita al servidor HTTP (Una línea)	Resultado de la solicitud	Resultado de la operación (código numérico que permite conocer el éxito o fracaso de la operación). (Una línea)
2	Cabeceras del requerimiento	Unas obligatorias y otras opcionales	Cabeceras de la respuesta	Unas obligatorias y otras opcionales
3	En blanco.	(Una línea)	En blanco.	(Una línea)
4	Información opcional		Información opcional	

Tabla 9.1. Dos tipos de mensajes: uno para realizar peticiones y otro para devolver la correspondiente respuesta.

El conocimiento y empleo de los mensajes HTTP es necesario para:

- Diseñar módulos CGI, ya que es preciso construir una respuesta similar a la que el servidor HTTP proporciona al cliente.
- Diseñar aplicaciones independientes que soliciten información a un servidor (automatizar la recuperación de documentos o construir robots de búsqueda) se debe construir una cabecera http con la información de la petición al servidor.

9.4.4. *Comandos del Protocolo HTTP*

Los comandos o verbos de HTTP representan las diferentes ***operaciones que se pueden solicitar a un servidor HTTP***. El formato general de un comando es:

Nombre del comando, Objeto sobre el que se aplica, Versión de http utilizada.

Cada comando actúa sobre un objeto del servidor, normalmente un archivo o aplicación, que se toma de la URL de activación. La última parte de esta URL, que representa la dirección de un objeto dentro de un servidor HTTP, es el parámetro sobre el que se aplica el comando. Se compone de una serie de nombres de directorios y archivos, además de parámetros opcionales para las aplicaciones CGI.

El estándar HTTP/1.0 recoge únicamente tres comandos, que representan las operaciones de recepción y envío de información y chequeo de estado:

- ***GET:*** Se utiliza para recoger cualquier tipo de información del servidor. Se utiliza siempre que se activa un enlace (pulsando) o se teclea directamente a una URL. Como resultado, el servidor HTTP: Envía el documento correspondiente a la URL seleccionada, o bien activa un módulo CGI, que generará a su vez la información de retorno.
- ***HEAD:*** Solicita información sobre un objeto (archivo): tamaño, tipo, fecha de modificación… Es utilizado por los gestores de

cachés de páginas o los servidores proxy, para conocer cuándo es necesario actualizar la copia que se mantiene de un archivo.

- ***POST:*** Sirve para enviar información al servidor, por ejemplo los datos contenidos en un formulario. El servidor pasará esta información a un proceso encargado de su tratamiento (generalmente una aplicación CGI). La operación que se realiza con la información proporcionada depende de la URL utilizada. Se utiliza, sobre todo, en los formularios.

La última versión de HTTP, denominada 1.1, recoge otras novedades, como los siguientes comandos:

- ***PUT:*** Actualiza información sobre un objeto del servidor. Es similar a POST, pero en este caso, la información enviada al servidor debe ser almacenada en la URL que acompaña al comando. Así se puede actualizar el contenido de un documento
- ***DELETE:*** Elimina el documento especificado del servidor.
- ***LINK:*** Crea una relación entre documentos.
- ***UNLINK:*** Elimina una relación existente entre documentos del servidor.

9.4.5. *Las Cabeceras de los Mensajes HTTP*

Son un conjunto de ***variables*** que se incluyen en ***los mensajes HTTP***, ***para modificar su comportamiento*** o ***incluir información de interés***. En función de su nombre, pueden aparecer en los requerimientos de un cliente, en las respuestas del servidor o en ambos tipos de mensajes. El formato general de una cabecera es:

Nombre de la variable : Cadena ASCII con su valor

Los nombres de variables se pueden escribir con cualquier combinación de mayúsculas y minúsculas. Además, se debe incluir un espacio en blanco entre el signo ":" y su valor.

9.4.6. *Tipos de Cabeceras*

9.4.6.1. Cabeceras Comunes para Peticiones y Respuestas

Algunas de estas son las siguientes:

- Content-Type: descripción MIME de la información contenida en este mensaje. Es la referencia que utilizan las aplicaciones Web para dar el correcto tratamiento a los datos que reciben.
- Content-Length: longitud en bytes de los datos enviados, expresado en base decimal.
- Content-Encoding: formato de codificación de los datos enviados en este mensaje. Sirve, por ejemplo, para enviar datos comprimidos (x-gzip o x-compress) o encriptados.
- Date: fecha local de la operación. Las fechas deben incluir la zona horaria en que reside el sistema que genera la operación. Por ejemplo: Sunday, 12-Dec-96 12:21:22 GMT+01. No existe un formato único en las fechas; incluso es posible encontrar casos en los que no se dispone de la zona horaria correspondiente, con los problemas de sincronización que esto produce. Los formatos de fecha a emplear están recogidos en los RFC 1036 y 1123.
- Pragma: permite incluir información variada relacionada con el protocolo HTTP en el requerimiento o respuesta que se está realizando. Por ejemplo, un cliente envía un Pragma: no-cache para informar que desea una copia nueva del recurso especificado.

9.4.6.2. Cabeceras sólo para Peticiones del Cliente

A continuación se listan algunas:

- Accept: campo opcional que contiene una lista de tipos MIME aceptados por el cliente. Se pueden utilizar * para indicar rangos de tipos de datos; tipo /* indica todos los subtipos de un determinado medio, mientras que */* representa a cualquier tipo de dato disponible.

- Authorization: clave de acceso que envía un cliente para acceder a un recurso de uso protegido o limitado. La información incluye el formato de autorización empleado, seguido de la clave de acceso propiamente dicha.
- From: campo opcional que contiene la dirección de correo electrónico del usuario del cliente Web que realiza el acceso.
- If-Modified-Since: permite realizar operaciones GET condicionales, en función de sí la fecha de modificación del objeto requerido es anterior o posterior a la fecha proporcionada. Puede ser utilizada por los sistemas de almacenamiento temporal de páginas. Es equivalente a realizar un HEAD seguido de un GET normal.
- Referer: contiene la URL del documento desde donde se ha activado este enlace. De esta forma, un servidor puede informar al creador de ese documento de cambios o actualizaciones en los enlaces que contiene. No todos los clientes lo envían.
- User-agent: cadena que identifica el tipo y versión del cliente que realiza la petición. Por ejemplo, los browsers de Netscape envían cadenas del tipo User-Agent: Mozilla/3.0 (WinNT; I)

9.4.6.3. Cabeceras sólo para Respuestas del Servidor HTTP

Se listan algunas de estas a continuación:

- Allow: informa de los comandos HTTP opcionales que se pueden aplicar sobre el objeto al que se refiere esta respuesta. Por ejemplo, Allow: GET, POST.
- Expires: fecha de expiración del objeto enviado. Los sistemas de caché deben descartar las posibles copias del objeto pasada esta fecha. Por ejemplo, Expires: Thu, 12 Jan 97 00:00:00 GMT+1. No todos los sistemas lo envían. Puede cambiarse utilizando un <META EXPIRES> en el encabezado de cada documento.
- Last-modified: fecha local de modificación del objeto devuelto. Se puede corresponder con la fecha de modificación de un archivo en disco, o, para información generada

dinámicamente desde una base de datos, con la fecha de modificación del registro de datos correspondiente.

- Location: informa sobre la dirección exacta del recurso al que se ha accedido. Cuando el servidor proporciona un código de respuesta de la serie 3xx, este parámetro contiene la URL necesaria para accesos posteriores a este recurso.
- Server: cadena que identifica el tipo y versión del servidor HTTP. Por ejemplo, Server: NCSA 1.4.
- WWW-Autenticate: cuando se accede a un recurso protegido o de acceso restringido, el servidor devuelve un código de estado 401, y utiliza este campo para informar de los modelos de autenticación válidos para acceder a este recurso.

9.4.6.4. Códigos de Estado del Servidor

Ante cada transacción con un servidor HTTP, éste devuelve un código numérico que informa sobre el resultado de la operación, como primera línea del mensaje de respuesta. Estos códigos aparecen en algunos casos en la pantalla del cliente, cuando se produce un error (en ocasiones se presenta un mensaje de error más elaborado, en forma de documento HTML). El formato de la línea de estado es:

Versión de protocolo, HTTP utilizada, Código numérico de estado (tres dígitos), Descripción del código numérico

A continuación se muestra una tabla con los mensajes de estado más comunes:(Ver tabla 9.2)

9.4.6.5. Códigos de Estado más Comunes del Servidor HTTP

Existen ***cinco categorías*** de mensajes de estado, organizadas por el primer dígito del código numérico de la respuesta:

- 1xx : mensajes informativos. Por ahora (en HTTP/1.0) no se utilizan, y están reservados para un futuro uso.
- 2xx : mensajes asociados con operaciones realizadas correctamente.

- 3xx : mensajes de redirección, que informan de operaciones complementarias que se deben realizar para finalizar la operación.
- 4xx : errores del cliente; el requerimiento contiene algún error, o no puede ser realizado.
- 5xx : errores del servidor, que no ha podido llevar a cabo una solicitud.

Código	***Comentario***	***Descripción***
200	OK	Operación realizada satisfactoriamente.
201	Created	La operación ha sido realizada correctamente, y como resultado se ha creado un nuevo objeto, cuya URL de acceso se proporciona en el cuerpo de la respuesta. Este nuevo objeto ya está disponible. Puede ser utilizado en sistemas de edición de documentos.
202	Accepted	La operación ha sido realizada correctamente, y como resultado se ha creado un nuevo objeto, cuya URL de acceso se proporciona en el cuerpo de la respuesta. El nuevo objeto no está disponible por el momento. En el cuerpo de la respuesta se debe informar sobre la disponibilidad de la información.
204	No Content	La operación ha sido aceptada, pero no ha producido ningún resultado de interés. El cliente no deberá modificar el documento que está mostrando en este momento.
301	Moved Permanently	El objeto al que se accede ha sido movido a otro lugar de forma permanente. El servidor proporciona, además, la nueva URL en la variable Location de la respuesta. Algunos browsers acceden automáticamente a la nueva URL. En caso de tener capacidad, el cliente puede actualizar la URL incorrecta

		por ejemplo, en la agenda de *bookmarks*.
302	Moved Temporarily	El objeto al que se accede ha sido movido a otro lugar de forma temporal. El servidor proporciona, además, la nueva URL en la variable Location de la respuesta. Algunos browsers acceden automáticamente a la nueva URL. El cliente no debe modificar ninguna de las referencias a la URL errónea.
304	Not Modified	Cuando se hace un GET condicional, y el documento no ha sido modificado, se devuelve este código de estado.
400	Bad Request	La petición tiene un error de sintaxis y no es entendida por el servidor.
401	Unauthorized	La petición requiere una autorización especial, que normalmente consiste en un nombre y clave que el servidor verificará. El campo WWW-Autenticate informa de los protocolos de autenticación aceptados para este recurso.
403	Forbidden	Está prohibido el acceso a este recurso. No es posible utilizar una clave para modificar la protección.
404	Not Found	La URL solicitada no existe.
500	Internal Server Error	El servidor ha tenido un error interno, y no puede continuar con el procesamiento.
501	Not Implemented	El servidor no tiene capacidad, por su diseño interno, para llevar a cabo el requerimiento del cliente.
502	Bad Gateway	El servidor, que está actuando como proxy o pasarela, ha encontrado un error al acceder al recurso que había solicitado el cliente.
503	Service Unavailable	El servidor está actualmente deshabilitado, y no es capaz de atender el requerimiento.

Tabla 9.2. Mensajes de estado más comunes del servidor HTTP.

9.4.6.6. Cachés de Páginas y Servidores Proxy

Muchos clientes Web utilizan un *sistema para reducir el número de accesos y transferencias de información* a través de Internet, y así *agilizar la presentación de documentos* previamente visitados. Para ello, almacenan en el disco del cliente una copia de las últimas páginas a las que se ha accedido. Este mecanismo, denominado "*caché de páginas*", mantiene la fecha de acceso a un documento y comprueba, a través de un comando HEAD, la fecha actual de modificación del mismo. En caso de que se detecte un cambio o actualización, el cliente accederá, ahora a través de un GET, a recoger la nueva versión del archivo. En caso contrario, se procederá a utilizar la copia local.

Un sistema parecido, pero con más funciones es el denominado "*servidor proxy*". Este tipo de servidor, es una *mezcla entre servidor HTTP y cliente Web, realiza las funciones de caché y acceso a las páginas* para un gran número de clientes. Todos los clientes conectados a un proxy dejan que éste sea el encargado de recoger las URLs solicitadas. De esta forma, en caso de que varios clientes accedan a la misma página, el servidor proxy la podrá proporcionar con un único acceso a la información original.

La principal ventaja de ambos sistemas es la drástica reducción de conexiones necesarias a Internet, en caso de que los clientes accedan a un conjunto similar de páginas, como suele ocurrir con mucha frecuencia. Además, determinadas organizaciones limitan, por motivos de seguridad, los accesos desde su organización al exterior y viceversa. Para ello, se dispone de sistemas denominados "*cortafuegos*" (firewalls), que son los *únicos habilitados para conectarse con el exterior*. En este caso, el uso de un servidor proxy se vuelve indispensable.

9.4.6.7. *Magic Cookies*

Las *'galletas mágicas'* son una de las incorporaciones más recientes al protocolo HTTP, y son parte del nuevo estándar HTTP/1.1. Las cookies *son pequeños archivos de texto que se intercambian los clientes y servidores HTPP, para solucionar* una de las principales

deficiencias del protocolo: *la falta de información de estado entre dos transacciones*. Fueron introducidas por Netscape, y ahora han sido estandarizadas en el RFC 2109.

9.4.6.8. Resumen de las Versiones del Protocolo

- ***HTTP 1.0*** Esta especificación prevé las características básicas del protocolo. Fue desarrollado por Tim Berners-Lee, Roy T. Fielding, y Henrik Frystyk Nielsen.
- ***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, propuesta por Simón Spero. Es un protocolo binario con nuevas características para un acceso más rápido usando TCP. Este es el último HTTP en la actualidad.

NOTA.- Para reafirmar algunos de los conceptos adquiridos en la teoría sobre la Capa de Aplicación del Modelo OSI; se recomienda hacer la siguiente práctica contenida dentro de nuestra tesis:

Práctica 10 Aplicación Web Apache

En donde al finalizar la práctica el alumno:

Al finalizar la práctica podrá instalar y configurar una aplicación típica de la capa 7 del Modelo OSI en Plataforma Linux.

Será capaz de Instalar y configurar de forma segura un servidor web en su versión básica, pero con posibilidad de transferencias seguras mediante SSL (Secure Sockets Layer).

Conocerá las principales directivas de configuración de un servidor web apache.

Conclusiones y Comentarios Finales

Conclusiones

Se puede concluir que a través del desarrollo de la presente tesis se ha cumplido con los objetivos originalmente propuestos:

- Se llevó a cabo un análisis del nuevo temario de la materia Redes de Datos, que permitió diseñar y desarrollar una serie de prácticas de laboratorio que concedieron la elaboración de un manual de prácticas acordes a la asignatura.

- Se tomaron como base las consideraciones, recomendaciones y modificaciones que se plantearon para la asignatura en la actual revisión a los planes y programas de estudio de la carrera de Ingeniero en Computación.

La presente propuesta de este manual de prácticas para nuestro Laboratorio de Redes de Datos, permitirá a los nuevos egresados de la Facultad de Ingeniería de la Universidad Nacional Autónoma de México, complementar los conocimientos teóricos adquiridos dentro del aula a través del desarrollo de las diferentes prácticas.

El manual de prácticas se ha diseñado y elaborado de manera satisfactoria para cada una de las capas del Modelo OSI, de modo que puedan ser desarrolladas por los estudiantes a fin de ejercitar los conceptos teóricos de manera práctica. Abarca 10 prácticas basadas en este modelo y 2 prácticas de nivelación sobre introducción y comandos básicos sobre Linux.

Estamos convencidos que las prácticas probadas dentro del laboratorio con alumnos de la carrera de ingeniería en computación han sido y serán de gran ayuda para que los futuros egresados de ésta, nuestra máxima casa de estudios tengan más herramientas para su preparación y formación para su futuro como profesionistas.

Comentarios Finales

Cabe comentar que el presente manual de prácticas es considerado como una propuesta para el laboratorio de redes, sin

embargo, se espera que sean aprobadas en su totalidad y sean impartidas dentro del laboratorio a lo largo del semestre.

También es cierto que para lograr esta serie de prácticas fue necesario trabajar con el equipo disponible dentro del laboratorio, a tal grado que fue imprescindible la migración de los sistemas operativos empleados a otros más recientes (Windows XP y Linux Red Hat 9.0), lo cual hizo que se dificultará el avance y desarrollo de éstas.

Se espera que el laboratorio en un futuro cuente con el equipo necesario para ampliar la cantidad de prácticas propuestas y que estas a su vez, sean mejores y que vayan avanzando con las necesidades de los futuros alumnos de esta Facultad.

Las prácticas sugeridas ya han sido desarrolladas a manera de prueba piloto en los tres últimos períodos intersemestrales, así como en las Jornadas de Ingeniería de Computación celebradas en la Facultad de Ingeniería en los meses de Agosto y Septiembre del presente año, a través de las cuales logramos poner a prueba dichas prácticas, para de esta forma dar a los futuros egresados la última versión de las mismas.

Esta propuesta está abierta para ser ampliada y mejorada, o bien, ser complementada con otras prácticas.

Glosario

Glosario

Ancho de Banda

Es el máximo rango de frecuencias asignadas a un canal de transmisión. Usualmente se mide en bits por segundo (bps).

ANSI (American National Standards Institute)

Instituto Nacional de Normalización, miembro del ISO. Está coordinado por grupo de estándares en Estados Unidos.

API (Interfaz de Programa de Aplicación)

Es un conjunto de reglas que permiten que las aplicaciones escritas por los usuarios puedan acceder a los servicios de un sistema de software.

ARPA (Agencia para la Investigación de Proyectos Avanzados)

ASCII (American Standard Code for Information Interchange) (Código Normalizado Americano para el Intercambio de Información)

Es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y otras lenguas occidentales. Creado aproximadamente en 1963 por el Comité Estadounidense de Estándares (ASA) como una refundición o evolución de los conjuntos de códigos utilizados entonces en telegrafía. Normalmente el código ASCII se extiende a 8 bits (1 byte) añadiendo un bit de control, llamado bit de paridad.

Atenuación

Es la pérdida de potencia de una señal a lo largo del canal de transmisión. La atenuación es expresada en decibeles.

ATM (Modo de Transferencia Asíncrono)

Autenticación

La autenticación es un proceso por el cual el receptor de un mensaje puede mantener confidencial la identidad del que lo envía y/o la integridad del mensaje. Los protocolos de autenticación se basan en sistemas de encriptado convencionales con clave secreta (como en el caso de DES) o con clave pública (como en el caso del sistema RSA),

la autenticación en los sistemas con clave pública utiliza firmas digitales.

Backbone

Línea de alta velocidad. Serie de conexiones que forman un mayor ancho de banda en una red. Cable principal de una red.

Binario

Binario. Significa dos, y es el principio fundamental en que se basan los computadores digitales. Todo lo que se ingresa a la computadora se convierte en números binarios conformados por los dos dígitos 0 y 1 (bits).

Bit

Es la unidad de información más pequeña, proviene de la contracción de la palabra inglesa binary digit, sólo puede tomar los valores de 0 y 1.

BNC (British Naval Connector) Conector naval británico.

Conector utilizado comúnmente para cable coaxial. El enchufe se ve como un pequeño cilindro de metal sin la tapa y dos pequeñas agujas que salen del borde superior en ambos lados. Después de insertado, se hace girar el enchufe, apretando las agujas en el zócalo.

Browser

Se refiere a una aplicación que permite visualizar los contenidos de las páginas web en Internet. Algunos de estos navegadores son muy conocidos, por ejemplo: Netscape, Opera, Internet Explorer, entre otros. Pueden obtenerse fácilmente y de forma gratuita en Internet.

Buffer

(Memoria intermedia) Área de memoria utilizada por el módem o la computadora para almacenar la información que no puede procesar en el acto (según le va llegando).

Bus

Línea o canal de transmisión que permite transportar información y comúnmente es utilizado para compartir dispositivos.

Bus Lineal

Método más sencillo de comunicación, consiste de un solo cable principal que conecta a todos los equipos en una sola línea.

Byte

Es el conjunto de 8 bits u octeto y con el se puede representar un carácter.

Canal

1. Vía o ruta de alta velocidad, metálica o de fibra óptica, entre la computadora y las unidades de control de los dispositivos periféricos. Los canales se utilizan en mainframes y máquinas técnicamente complejas. Cada canal es una unidad independiente que puede transferir datos en forma concurrente con otros canales, así como con la CPU. Por ejemplo, en una computadora de 10 canales, se están transmitiendo de manera simultánea 10 flujos de datos hacia y desde la CPU. En contraste, el bus en un computador personal sirve como canal común, compartido entre todos los dispositivos, cada uno de los cuales debe esperar su turno en el bus.

2. En comunicaciones, cualquier vía entre dos computadoras o terminales. Puede referirse al medio físico, como un cable coaxial o a una frecuencia portadora específica (subcanal) dentro de un canal más amplio o un medio inalámbrico.

Canaleta

Son conductos o tubos, normalmente de plástico, que rodean el cable.

Cascadeo

En las Bases de Datos Orientadas a Objetos, se presenta la Transformación a Esquemas Orientados a Objetos y ésta a su vez contempla la Técnica de agrupación de clusters; en la que se consideran cuatro reglas adicionales que auxilian para preservar la secuencia lógica y natural de visualizar la base de datos a diferentes niveles de abstracción, para mantener toda la semántica de los datos, y para manejar algunas situaciones de conflicto, las cuales son; Agrupamiento Paso a Paso, Consistencia, Cascadeo y Visibilidad/Invisibilidad.

Si una entidad Ei es candidato tanto para la operación de formación de un cluster de cualquier tipo (absorción de entidad débil, agrupación dominante, generalización/categorización o por vínculos) como una entidad esclavo (i.e componente de una entidad compleja potencial E); así como también candidato para la operación de formación de cluster como una entidad maestro (i.e. una entidad cuyo nombre es idéntico al nombre del cluster potencial como son entidades dominantes/fuertes, entidades generalizadas y entidades del lado uno en un vínculo uno-a-muchos) entonces la operación de formación de cluster interna (i.e.la que involucra Ei como maestro) es aplicada antes del agrupamiento que involucra Ei como esclavo. Un caso especial, en la presencia de secuencias de entidades débiles con sus correspondientes vínculos, el agrupamiento de absorción empieza de la entidad y vínculo más dependientes, y formando iterativamente entidades complejas hasta que la entidad fuerte sea encontrada.

CCITT (International Consultative Committee for Telephony and Telgraphy)

Organización Internacional, dentro de la ITU (UIT en español) que a su vez lo es de la ONU, encargado de establecer recomendaciones referentes a las telecomunicaciones.

CGI (Computer Graphics Interface)
Interfaz gráfica para computadores.

Lenguaje independiente de gráficos para dispositivos, utilizado en pantallas de visualización, impresoras y plotters de GKS.

Codificar

Asignar un código para representar datos, como un código de partes.

Comando

Orden, mandato, comando. Instrucción para la computadora.

Comprimir

Compactar datos para ahorrar espacio.

Concentrador

Procesador programado para combinar datos transmitidos por varios dispositivos de baja velocidad y luego transmitirlos a una velocidad mucho mayor, generalmente a una computadora remota.

Concentradores Híbridos

Concentradores que permiten distintos tipos de cable, y permiten expandir las redes con el uso de mas concentradores interconectados.

Conector Cilíndrico

Atenúa la señal, no debe utilizarse en exceso.

Control Centralizado

Una de las estaciones tiene mayor jerarquía que las demás y controla las comunicaciones.

Control Distribuido

Todas las estaciones tienen la misma jerarquía.

Correo Electrónico

También denominado e-mail, es la transmisión de memos y mensajes a través de una red. Un sistema de correo electrónico requiere un sistema de mensajes, que provea la capacidad de almacenar y enviar, y un programa de correo que suministre la interfaz de usuario con sus funciones de envío y recepción.

Criptografía

Es el estudio de las técnicas y aplicaciones que tiene por objeto cifrar datos, aplicando algoritmos o transformaciones matemáticas, así como lograr la recuperación de los datos a partir del texto o datos cifrados.

CSMA (Carrier Sense, Multiple Access)

CSMA/CD (CSMA with Collision Detection).

dB

Es considerado como la base sobre la cual todos los diseñadores de telecomunicaciones comparan el rendimiento de los sistemas de cableado.

DCE (Equipo de Comunicación de Datos)

Demultiplexar

Reconvertir una transmisión que contiene varias señales entremezcladas en sus señales separadas originales.

DNS (Domain Naming System) Servidor de Nombres de Dominio

Sistema de nombrado de dominios. Sistema de direccionamiento de correo electrónico utilizado en redes como Internet y BITNET. Convierte una dirección de correo electrónico, como joe@clc.com a la dirección entre redes (IP) para transmisión.

Dominio

En comunicaciones, todos los recursos que están bajo el control de un solo sistema de computación. En una LAN, un dominio es una subred compuesta por un grupo de clientes y servidores bajo el control de una base de datos de seguridad. Al dividir las LAN en dominios se mejora el desempeño y la seguridad.

DTE (Data Terminating Equipment) Equipo Terminal de Datos

Dispositivo de comunicaciones que actúa como fuente o destino de señales en una red. Por lo general es una terminal o una computadora.

EBCDIC (Extended Binary Coded Decimal Interchange Code) Código ampliado de intercambio decimal codificado en binario.

Código binario para texto, comunicaciones y control de impresora de IBM. Este código se originó con el System/360 y aún se usa en mainframes IBM y en la mayoría de los computadores de medio rango de IBM. Es un código de 8 bits (256 combinaciones) que almacena un carácter alfanumérico o dos dígitos decimales en un byte.

EBCDIC y ASCII son los dos códigos de mayor uso para representar datos.

EMI (Electromagnetic Interference)

Interferencias Electromagnéticas

Firewall (Cortafuegos)

Actúa como una barrera de seguridad entre la INTRANET E Internet Para orevenir la intrusión de usuarios no autorizados. (Es un filtro).

Firmware

Categoría de chips de memoria que mantienen su contenido sin energía eléctrica; e incluye las tecnologías ROM, PROM, EPROM y EEPROM. Firmware se convierte en "software duro" cuando contiene código de programas.

HDSL (High-level Data Link Control) Control de enlace de datos de alto nivel

Protocolo de comunicaciones ISO usado en redes de conmutación por paquetes X.25. HDLC provee corrección de errores en el estrato de enlace de datos. SDCL, LAP y LAPB son subconjuntos de HDLC.

Hosts

Anfitrión. Computadora central o principal en un entorno de procesamiento distribuido. Por lo general se refiere a una gran computadora de tiempo compartido o una computadora central que controla una red.

HTML (HyperText Markup Languaje)
Lenguaje de Marcado de Hipertexto

Es un lenguaje para describir documentos estructurados.

HTTP (Protocolo para la Transferencia de HiperTexto)

IBM (International Business Machines Corporation, Armonk, NY)

Compañía de computadores más grande del mundo. Con el transcurso de los años, IBM ha ganado una reputación mundial por su tecnología y servicio. Fue la primera en desarrollar el disco flexible y el disco duro, y ha realizado muchas contribuciones a la industria de las computadoras. Como el líder mundial en mainframes, también ha tenido mucho éxito en las minicomputadoras y estaciones de trabajo con sus líneas AS/400 y RS/6000. En 1981, fijó el estándar para las computadoras personales, iniciando la industria de los

mismos; de los cuales IBM continúa siendo uno de los principales fabricantes.

Interfaz

Conexión e interacción entre hardware, software y el usuario. El diseño y construcción de interfaces constituye una parte principal del trabajo de los ingenieros, programadores y consultores

Internet

Inter (Entre) net (redes). Entre redes.

Internetworking

Interconexión de red.

Interrupción

Señal que capta la atención de la CPU y que usualmente se genera cuando se requiere una entrada/salida.

ISO (Internacional Standards Organization)
Organización Internacional de Normalización

Organismo cuya función es la de coordinar los trabajos de normalización realizados por los diferentes organismos internacionales.

KDC (Centro de Distribución de Llave)

Layering

En comunicaciones, protocolo que interactúa con otros protocolos para proveer todos los servicios de transmisión necesarios.

LLC (Control de Enlace Logico)

MAC (Control de Acceso al Medio)

Macintosh

Computadoras de la empresa Apple que se iniciaron en el 1984 tomando un gran auge, especialmente en USA. Es un sistema operativo multitareas que constituyó la primer interfase gráfica que obtuvo éxito comercial. La Mac fue un éxito inmediato en los campos

de producción gráfica y a la fecha tiene una parte importante de ése mercado.

Mbps Megabits por segundo

MIME Multi-purpose internet mail extender (extensor de correo en Internet de propósito general).

Es un protocolo usado para enviar mensaje de correo electrónico que contiene formas de datos de computadora distintos al texto. Algunos sistemas de correo electrónico tan solo están diseñados para mensajes textuales y el envío de otros datos de computadora provoca su corrupción MIME, que es una forma de codificar datos para evitar su corrupción. Al recibirse, los datos requieren ser descodificados antes de ser usados.

MNP 2-5 Microcom network protocolo (Protocolo de Red Microcom)

Es un conjunto de protocolos para corrección de errores y comprensión de datos.

NetBios (Net BEUI)

Es un protocolo pequeño, eficiente y rápido para pequeñas redes LAN de alta performance. Se configura solo y se autocontrola. Se instala y relaciona automáticamente con el adaptador de red cuando se instala Windows NT.

NFS (Sistema de Archivos de Red)

Nodo

Es el punto en donde se producen dos o más conexiones en una red de comunicaciones. No se trata de un elemento estrictamente físico, sino de una unidad funcional que exige hardware y software. Un nodo puede incluir controladores de comunicaciones, clusters, servidores, repetidores, etc.

NOM (Norma Oficial Mexicana)

Conjunto de normas jurídicas de carácter oficial, publicadas en el Diario Oficial de la Federación.

PDU (Unidad de datos de protocolo)

PIN (Personal identification number) número de identificación personal

Es un número usado como contraseña, en particular con las tarjetas bancarias y de crédito. Por ejemplo, una tarjeta bancaria es el dispositivo de identificación personal que ofrece la ID de usario y se usa en conjunto con un número de identificación personal para la obtención de efectivo de un cajero automático.

POP (Point of presence) punto de presencia

Es un punto de acceso a la red de un proveedor de servicios Internet. Por lo general, esto es la central telefónica donde los datos transmitidos abandonan la red telefónica pública e ingresan a la red del proveedor de servicios de Internet.

Protocolo

Se puede comparar con el lenguaje de habla. Es un conjunto de reglas para que las computadoras puedan comunicarse entre sí.

Proxy Servers

Habilita que se conecten diferentes computadoras de la red a Internet utilizando una simple dirección IP. Reemplazan direcciones IP privadas de computadoras enviando peticiones a un servidor de Internet con dirección IP pública asignada por algún ISP.

RDSI (Integrated Service Digital Network)

RDSI-BA (Red Digital de Servicios Integrados de Banda Ancha)

RDSI-BE (Red Digital de Servicios Integrados de Banda Estrecha)

Red Híbrida

En comunicaciones, una red compuesta por equipos de múltiples fabricantes.

RF (Radio Frecuencia)

RFC (Request for Comments) Petición de Comentarios

Documento que describe normas.

RPC (Control de Procedimientos Remotos)

Root

Administrador de un sistema Unix. // Root: Raíz, en inglés; se suele utilizar para designar el directorio (o carpeta) principal de una cierta unidad de disco.

Ruido

Interferencias en la línea que interfieren las comunicaciones**.**

SAP (Punto de Acceso a Servicio)

SCE (Sistema de Cableado Estructurado)

Señal Analógica

Utiliza una magnitud con una variación continua. representa la información como variaciones continuas de voltaje.

Señal Digital

Emplea valores discretos, predefinidos. Representa la información como pulsos de voltaje.

Servidor

En una red de ordenadores, un servidor es un equipo que pone ciertos recursos a disposición de otras computadoras (los clientes). Estos recursos pueden ser datos, aplicaciones, impresoras, etc.

Servidores de Archivos

En una LAN, computadora de alta velocidad que almacena los programas y archivos de datos compartidos por los usuarios en la red. También llamado network server (servidor de red), actúa como una unidad de disco remota.

Shell

(ó Intérprete de comandos) Nombre que recibe la parte de un sistema operativo que recibe nuestras órdenes. En MsDos suele ser el fichero COMMAND.COM, mientras que para los sistemas Unix hay varios disponibles, como el Bourne Shell (bsh), Korn Shell (ksh) o C Shell (csh).

SMTP*(*Simple Mail Transfer Protocol) Protocolo de Transferencia Simple de Correo

Es el protocolo usado para transportar el correo a través de Internet.

SNA (*System Network Arquitectura) Arquitectura de Sistemas de Redes*

Arquitectura de red exclusiva de IBM. Principalmente orientada a Mainframes.

Sniffer

Programa que monitorea y analiza el tráfico de una red para detectar problemas o cuellos de botella. Su objetivo es mantener la eficiencia del tráfico de datos. Pero también puede ser usado ilegítimamente para capturar datos en una red.

SNMP (Simple Network Management Protocol)

Protocolos para la gestión de redes basadas en TCP/IP.

Software

Todos los componentes informáticos de carácter no físico, sino lógico (se denomina también logical), como pueden ser Sistemas Operativos, programas dedicados a la gestión, de diseño, etc.

STM (Modo de Transferencia Sincrono)

STP

Cable de par trenzado apantallado (Shielded Twisted Pair).

Switch

Dispositivo que envía el paquete utilizando una ruta específica.

TDM (Multiplexado por Division en el Tiempo)

Token

Señal electrónica que viaja de un dispositivo a otro. Un Token es un mensaje especial que temporalmente da el control de acceso al medio al dispositivo que tiene el Token.

Topología

Es el diseño físico de los equipos, cableado y otros componentes de red.

Troyano

Aplicaciones ocultas que se descargan de la red y que pueden ser usadas por terceros para extraer datos personales.

UA (Agente de Usuario)

UDP (Protocolo de Datagramas de Usuario)

UE (Elemento de Usuario)

X.25

Define las relaciones para la transmisión por paquetes. Se refiere a los tres primeros niveles OSI. En cuanto al primer nivel, el físico, es importante el estándar RS-232-C (define número de pines o patillas, cómo han de ir los cables, etc.). En el nivel dos asegura que la comunicación sea fiable y el tres permite los circuitos virtuales permanentes.

Bibliografía

Bibliografía

Cisco Systems, Inc.
Academia de Networking de Cisco Systems
Guía del Primer Año
2ª. Edición
Pearson Educación, S.A.
Madrid, 2002

Cisco Systems, Inc.
Academia de Networking de Cisco Systems
Guía del Segundo Año
2ª. Edición
Pearson Educación, S.A.
Madrid, 2002

Comer Douglas E.
Interconectividad de Redes con TCP/IP. Vol I y II
Prentice Hall
3ª. Edición
México, 2000

EIA/TIA 568-A Norma para Cableado de Telecomunicaciones en Edificios Comerciales
Telecommunications Industry Association.
U.S.A. 1995

EIA/TIA 569 Norma para Vías de Telecomunicaciones y Espacios en Edificios Comerciales
Electronic Industries Association
Washington. 2000

Ford Merilee
Tecnologías de Interconectividad de Redes
Prentice Hall
Cisco Systems, Inc.

Halsall Fred
Comunicaciones de Datos, Redes de Computadores y Sistemas Abiertos
Pearson Educación
4ª. Edición
México 1998

Spurgeon Charles E.
Ethernet.The Definitive Guide
O'Reilly
First Edition
E.E.U.U. 2000

Stallings William
Comunicaciones y Redes de Computadores
Prentice Hall
6ª. Edición
España, 2000

Tanenbaum Andrew S.
Redes de Computadoras
Prentice Hall
3ª. Edición.
México, 1998

Tanenbaum Andrew S.
Redes de Computadoras
Prentice Hall Hispanoamericana
2ª. Edición.
México, 1997

UNAM
FACULTAD DE ING.
PROPUESTA DE IMPLEMENTACIÓN DE LOS PROTOCOLOS DE ENRUTAMIENTO DE LA RED DE ALTO DESEMPEÑO DEL BACKBONE DE CUDI.
TESIS QUE PARA OBTENER EL TITULO DE INGENIERO MECANICO ELECTRICISTA PRESENTA:
HANS LUDWIG REYES CHAVEZ
DIRECTOR DE TESIS: ING. ALFREDO HERNANDEZ MENDOZA
CIUDAD UNIVERSITARIA
JUNIO DE 2003

Información consultada en Internet en las siguientes direcciones electrónicas:

http://www.blyx.com/public/docs/pila_OSI.pdf
http://www.htmlweb.net/redes/tcp_ip/capa_1/fisica_1.html
http://www.htmlweb.net/redes/tcp_ip/capa_1/fisica_2.html
http://www.htmlweb.net/redes/tcp_ip/capa_1/fisica_3.html
http://www.htmlweb.net/redes/tcp_ip/capa_1/fisica_4.html
http://www.htmlweb.net/redes/tcp_ip/capa_1/fisica_5.html
http://www.microsoft.com/colombia/academico/comunidad/documentos/tips/tips1.asp
http://www.newsbook.tai.es/espjun03-5.html
http://personales.ciudad.com.ar/technical-support/topologías.doc
http://usuarios.lycos.es/bet zweb/photoalbum.html

Anexo A

Requerimientos de Prácticas

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 1** **Construcción de cables UTP para conexión directa y cruzada**

A. Pinzas de engarzadoras. (Mínimo 3)
B. Analizador de continuidad de cableado UTP ó TESTER de cableado UTP. (3)
   Se sugiere comprar el siguiente TESTER: DTX Cable Analyzer Modelo DTX-1800 de la Marca FLUKE. Se incorporan características técnicas al final de este anexo.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 2** **Componentes del Cableado Estructurado Norma ANSI/EIA/TIA 568**

A. Computadoras con Sistema Operativo Windows 98/NT/2000P/XP.
B. 1 Switch.
C. 1 Patch Panel ó Panel de Parcheo.
D. Pinzas para jack ó Pinzas de impacto.
E. Pinzas engarzadoras.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 3** **Instalación de una Red Básica en Plataforma Windows XP de Microsoft**

A. Computadoras con Sistema Operativo Windows 98/NT/2000P/XP.
B. Ranura de expansión PCI o ISA vacía.
C. Tarjeta de red PCI o ISA.

D. Controlador de la tarjeta de red.
E. Es necesario tener privilegios de administrador para realizar esta práctica.
F. Es posible que Windows le solicite al alumno que proporcione el nombre de la computadora y del grupo de trabajo. De ser así, se requiere proporcionar la siguiente información:
   Indique el nombre de la computadora:
   nombre_computadora
   Grupo: nombre_grupo
G. Cabe mencionar que después de instalar y configurar la tarjeta de red, se conectarán dos computadoras para probar la correcta construcción del cable de conexión cruzada *(crossover).* (Construido en la Práctica No.1).
   Se necesitará conocer los nombres e IP asociadas de las computadoras involucradas (nombre_computadora1 y su IP asociada, nombre_computadora2 y su IP asociada), así como el grupo al que pertenecen éstas deberá ser el mismo.

Nota.-En esta práctica se requiere contar con el controlador de la Tarjeta de Red (NIC) a emplear. Por supuesto esto va de acuerdo al equipo adquirido por el laboratorio, es decir, cada computadora cuenta con su propia tarjeta y con su controlador ó driver de fábrica.

Antes de realizar la práctica; se deberá desinstalar el controlador y retirar la tarjeta de red, de su ranura de expansión del equipo a utilizar.

En caso de que la tarjeta de red venga integrada en la tarjeta madre; sólo deberá desinstalar el driver ó bien revisar que tenga una ranura de expansión libre para insertar otra tarjeta de red. La instalación de dicho driver se explica paso a paso en dicha práctica.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 4** **Instalación de una Red Básica en Plataforma Linux**

A. Computadoras con Sistema Operativo Linux Mandrake 6.0.
B. Ranura de expansión PCI o ISA vacía.
C. Tarjeta de red PCI o ISA.
D. Controlador de la tarjeta de red.
E. Es necesario tener privilegios de administrador para realizar esta práctica. Por lo tanto se deberá iniciar una sesión como usuario root y el administrador o profesor deberá proporcionar dicha contraseña.
F. Cabe mencionar que después de instalar y configurar la tarjeta de red, se conectarán dos computadoras para probar la correcta construcción del cable crossover. (Construido en la Práctica No.1).
   Se necesitará conocer los nombres e IP asociadas de las computadoras involucradas (nombre_computadora1 y su IP asociada, nombre_computadora2 y su IP asociada).

Nota.-En esta práctica se requiere contar con el driver de la NIC a emplear. Por supuesto esto va de acuerdo al equipo adquirido por el laboratorio, es decir, cada computadora cuenta con su propia tarjeta y con su driver de fábrica.

Antes de realizar la práctica; se deberá desinstalar el controlador y retirar la tarjeta de red, de su ranura de expansión del equipo a utilizar.

En caso de que la tarjeta venga integrada en la tarjeta madre; sólo deberá desinstalar el driver ó bien revisar que tenga una ranura de expansión libre para insertar otra tarjeta de red. La instalación de dicho driver se explica paso a paso en dicha práctica.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 5** **Enrutamiento de paquetes**

A. Computadoras con Sistemas Operativos:
Linux Red Hat 9.0 ó Superior y Windows XP.
B. Creación de cuentas en cada computadora en donde el alumno trabajará para realizar la práctica:
Cuentas como usuario: root y contraseña: proporcionada por el profesor o administrador del Laboratorio. Para el Sistema Operativo Red Hat 9.0 ó Superior.
C. Cuentas como usuario: redes y contraseña: proporcionada por el profesor. Para el Sistema Operativo Windows.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 6** **Análisis de Paquetes**

A. Computadoras con Sistema Operativo Windows XP.
B. Creación de cuentas en cada computadora en donde el alumno trabajará para realizar la práctica:
Cuentas como usuario: redes y contraseña: proporcionada por el profesor ó administrador del Laboratorio.
C. Instalación del paquete Interfaz de red WinPcap_3_1_beta4.exe (Versión beta4 ó superior) en cada una de las computadoras donde se realice la práctica. Este paquete será la interfaz del Analizador de Paquetes (Analyzer), por lo tanto deberá instalarse primero.
D. Instalación del programa analyzer-setup.exe Analizador de Paquetes (Analyzer).

Nota:- En cada una de las máquinas el administrador del laboratorio deberá tener instalados los paquetes WinPCap_3_1_beta4 y analyzer-setup (este último a través de su instalador).

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 7** **TCP y UDP**

A. 1 Computadora por cada dos alumnos con Sistema Operativo *Linux Red Hat 9.0 ó Superior.*
B. Creación de cuentas de usuario: redes y contraseñas correspondientes, para cada una de las computadoras del laboratorio, que serán proporcionadas por el profesor ó administrador del mismo a los alumnos al momento de realizar la práctica.
C. Un programa instalado previamente en cada una de las computadoras del laboratorio; llamado sock (sock-1.1.tar.tar).

NOTA.- En esta práctica hacemos uso de un programa llamado sock-1.1.tar el cual es instalado a lo largo de la práctica. El profesor ó administrador deberá contar previamente con este programa.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 8** **Filtrados de Puertos TCP/UDP mediante un Firewall**

A. 1 Computadora por cada dos alumnos con Sistema Operativo Windows 98/NT/2000/XP ó superior.
B. Cuentas de usuario: redes y contraseña: proporcionada por el profesor.
C. Instalación del Firewall personal Kerio Versión 4.0 ó superior. (Kerio-pf-4.1.3-en-win.exe). Para su instalación se deberá contar con privilegios administrativos.

Nota.- El firewall a emplear deberá ser instalado por el administrador del laboratorio en cada una de las computadoras.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 9** **SSH: Secure Shell**

A. 1 Computadora por cada dos alumnos con Sistema Operativo LINUX Red Hat .9.0 ó Superior.
B. En cada computadora crear cuentas de usuario: **usuario: redes**, **contraseña: ingenieria.**
C. En cada computadora crear cuentas de usuario: **usuario: estudiante, contraseña: unamcu**.
D. **usuario: root, contraseña: proporcionada por el profesor**

Nota:- En esta práctica se emplearan los comandos preinstalados en el Sistema Operativo.

REQUERIMIENTOS ADMINISTRATIVOS PARA REALIZAR LA **PRÁCTICA 10** **Aplicación Web Apache**

A. 1 Computadora por cada dos alumnos con Sistema Operativo LINUX Red Hat .9.0 ó Superior.
B. El archivo apache_1.3.24.tar.gz
C. El archivo mod_ssl-2.8.8-1.3.24.tar.gz
D. El archivo openssl-0.9.6b.tar.gz
E. Creación de cuentas en cada computadora en donde el alumno trabajará para realizar la práctica:
   Cuentas como usuario: root y contraseña: proporcionada por el profesor ó administrador del Laboratorio.

Nota.-Para esta práctica deberán instalarse tres programas (apache_1.3.24.tar.gz, mod_ssl-2.8.8-1.3.24.tar.gz, openssl-0.9.6b.tar.gz), lo cual se detalla paso a paso en dicha práctica. El administrador del laboratorio solo deberá contar con dichos programas para la realización de esta práctica.

Aunque la última versión de apache disponible es la 2.0.36, para poder trabajar con SSL debemos utilizar la 1.3.24, que es la última para la que está disponible el módulo criptográfico mod_ssl.

Si queremos que tenga posibilidad de utilizar transferencias seguras con SSL necesitaremos al menos dos paquetes más: mod_ssl y OpenSSL, todos disponibles en un disco provisto por el administrador. En resumen:

Paquete: Apache
Versión: 1.3.24
Archivo: apache_1.3.24.tar.gz

Paquete: mod_ssl
Versión: 2.8.8-1.3.24
Descripción: la interfaz Apache con OpenSSL
Archivo: mod_ssl-2.8.8-1.3.24.tar.gz

Paquete: OpenSSL
Versión: 0.9.6b
Descripción: El Toolkit Open Source para SSL/TLS
Archivo: openssl-0.9.6b.tar.gz

Instalar sólo el paquete apache con SSL.

FLUKE networks.

NETWORKSUPERVISION™



DTX CableAnalyzer ™ reduce significativamente el tiempo total necesario para realizar la certificación.

La nueva serie de certificadores DTX CableAnalyzer™ de Fluke Networks es la plataforma de certificación para hoy y para mañana. Esta nueva y revolucionaria plataforma reduce de forma significativa el tiempo total necesario para la certificación mediante la mejora de todos los aspectos del proceso de comprobación. La revolución comienza con un tiempo de comprobación automática de categoría 6 tres veces menor que otros certificadores, y unas pruebas de fibra cinco veces más rápidas. Pero esto es sólo el principio. Además, DTX le proporciona una precisión de nivel IV, un sistema de diagnósticos para la solución de problemas sin precedentes, 900 MHz de ancho de banda para pruebas, 12 horas de duración de la batería y una configuración y generación de informes casi instantánea. Con DTX, estará listo para cualquier cosa que le depare el trabajo, tanto ahora como en el futuro.

DTX Certificador y Analizador de Cableado

El Tiempo es Oro.

FLUKE networks.

La ventaja DTX: velocidad y precisión.

Las potentes características, la velocidad y la extraordinaria precisión de DTX CableAnalyzer hacen que el personal sea más eficiente y productivo; y podrá ver fácilmente los resultados. Porque el tiempo el oro, y ningún otro certificador le ahorra más que DTX.

- **Aumente la productividad desde el primer día.** Una interfaz intuitiva implica un menor tiempo de formación y más tiempo disponible para las pruebas.
- **De Cero a Certificado en sólo 12 segundos.** Esta velocidad sin precedentes de las pruebas para la categoría 6 le permitirá moverse entre enlaces a velocidades tres veces más rápido que con los certificadores anteriores.
- **Precisión de nivel IV.** Consiga los resultados más precisos en el menor tiempo posible.
- **Intervalo de frecuencias de 900 MHz.** Le prepara para las futuras aplicaciones, como 10 GB Ethernet, clase F y CATV.
- **Diagnósticos avanzados en menos tiempo.** Detecta el lugar donde se ha producido un fallo y es el único certificador que sugiere una acción correctora, con el consiguiente ahorro de tiempo en la resolución de problemas
- **Certificación de fibra en tiempo récord.** Certifica un par de fibras en 12 segundos con los módulos de fibra DTX monomodo y multimodo.
- **12 horas de duración de la batería.** Le brinda la capacidad necesaria para completar cualquier trabajo.
- **Ahorre tiempo en la gestión de los resultados.** Desde la configuración hasta la generación de informes, la sencilla interfaz de usuario del software para gestión de pruebas de cable LinkWare™ y las características de ahorro de tiempo aumentan la productividad.

Reduce de forma significativa el tiempo total para certificar.

La serie DTX Digital CableAnalyzer le proporciona una completa solución que optimiza cada aspecto del trabajo de certificación, desde la configuración hasta las pruebas y la solución de problemas en tiempo récord, pasando por la generación de informes de resultados para el cliente. En pocas palabras, DTX puede ahorrarle de forma considerable tiempo y dinero, hasta cuatro horas diarias.

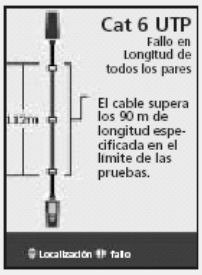
Certificación de Cat 6 en 12 segundos

Los DTX-1200 y DTX-1800 realizan las pruebas de certificación de Cat 6 en sólo 12 segundos, con total compatibilidad con los estándares de la industria y con una precisión extraordinaria. Es decir, tres veces más rápido que los certificadores existentes. Esta increíble velocidad significa que se pueden comprobar hasta 170 enlaces más en un turno de ocho horas.

Los fallos se corrigen con el doble de rapidez

Cuando falla un enlace, la serie DTX proporciona instrucciones sencillas para identificar el punto de fallo (distancia desde el certificador) y las posibles causas del fallo.

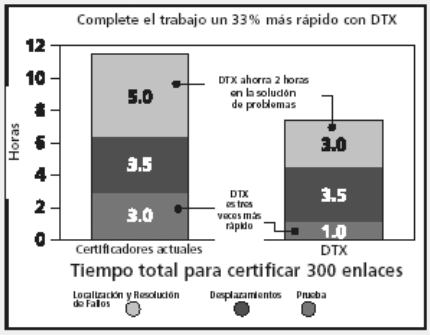
Estas instrucciones no sólo le indican el problema, sino que también *identifican las acciones correctoras* que pueden realizar los técnicos para solucionarlo, todo ello sin necesidad de consultar al responsable del proyecto. En lugar de gastar tiempo probando y equivocándose, y volver a certificar para verificar que se ha resuelto el problema, el técnico sabe exactamente



dónde mirar y qué se puede hacer para arreglar el enlace que falla. *Aunque sólo falle el dos por ciento de los cables certificados en un turno, ahorrará tanto como dos horas de mano de obra por día de certificación.*

Rápido incluso cuando no está realizando comprobaciones

La serie DTX no es sólo una forma rápida de realizar comprobaciones. También recorta los tiempos para realizar configuraciones y generar informes. La facilidad de uso reduce el tiempo necesario de formación. La duración ampliada de la batería significa que se pueden hacer más con una sola carga. Además, una pantalla en color brillante, la amplia memoria y los intercomunicadores incluidos en el DTX contribuyen a una productividad global a la vez que aumentan la satisfacción del usuario. **Todo esto contribuye a ahorrar tiempo y dinero cada día.**



DTX CableAnalyzer:
Un enfoque visionario de la certificación.

Precisión de nivel IV
Supera los requisitos de las especificaciones para la categoría 6 y superiores.

Fibra lista en todo momento
Los módulos de fibra opcionales se acoplan fácilmente y quedan protegidos e integrados en la cavidad del módulo DTX.



Diagnósticos avanzados que ahorran tiempo
Detectan problemas en cualquier lugar del enlace y sugieren la acción correctora para ayudarle a terminar el trabajo a tiempo.

Robusto Adaptador de Enlace Permanente
proporciona una precisión repetible e interoperabilidad de Cat 6.

Batería 12 horas de duración
La batería de ión de litio permite certificar durante todo el día.

Memoria interna
almacena hasta 250 resultados gráficos de las pruebas de Cat 6 o hasta 2.000 informes en formato texto.

Tarjeta de memoria
para transferencia de datos instantánea: cada múltiplo de 16 MB almacena 300 resultados gráficos de las pruebas de enlace de Cat 6.



El intercomunicador integrado ahorra tiempo
ya que permite ponerse en contacto con el compañero de pruebas en el otro extremo del enlace, en fibra y cobre.

Protector de goma integrado en la carcasa
para aguantar las duras condiciones de trabajo.

Ancho de banda de hasta 900 MHz - permite distribución de vídeo, clase F y 10GB Ethernet

Comprobación automática de Cat 6 en 12 segundos - efectúa el proceso de pruebas tres veces más rápido que cualquier otro certificador.

Gran pantalla en color con iluminación posterior brillante para facilitar la visualización.

Comprobación automática de fibra en 12 segundos - realiza la certificación de doble fibra y doble longitud de onda.

Guarde los resultados gráficos completos de las pruebas en el software LinkWare para PC de Fluke Networks.

Puerto USB para descargar datos de pruebas a alta velocidad.

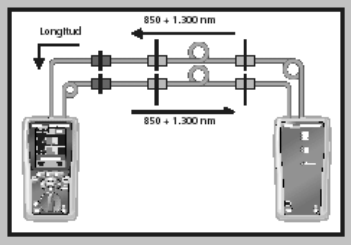
Diseño ergonómico, portátil y ligero para facilitar el uso sobre el terreno.

El mando rotario facilita el aprendizaje y simplifica el manejo: sabrá cuál es el modo de prueba seleccionado en todo momento.

Consiga realizar más trabajos en menos tiempo con los módulos de fibra integrados.

Certificación de fibra en tiempo récord

Los módulos de fibra de DTX aceleran el proceso de comprobación gracias a una exclusiva tecnología y una interfaz sencilla. Pulse el botón Autotest (comprobación automática) y disfrute de un sistema de certificación compatible con los estándares; compruebe dos fibras, cada una a dos longitudes de onda, mida la longitud y determine el estado PASA/FALLA; todo ello en unos 12 segundos. Nuestros módulos de fibra le permiten probar más fibras en menos tiempo, reduciendo los costes de las pruebas y permitiéndole realizar otras tareas. Puede ahorrar fácilmente más de 100 horas anuales.



Al contrario que la mayoría de los adaptadores de fibra, que miden longitud y dos mediciones de pérdidas por cada comprobación automática, DTX es la única solución de certificación de fibra óptica que mide la longitud y cuatro mediciones de pérdidas por cada comprobación automática, todo ello en unos 12 segundos. Con los módulos de fibra de DTX, se puede certificar fibra cinco veces más rápido.

Certifique cobre y fibra con sólo pulsar un botón

Sólo la plataforma DTX ofrece módulos internos de fibra opcionales, para que no pierda tiempo buscando un adaptador de fibra. Además, la certificación de fibra estará siempre lista cuando usted lo esté. No existe otra solución que le permita alternar entre cobre y fibra con sólo pulsar un botón.

Certifica fibra multimodo y monomodo

Los módulos de fibra de DTX usan LED y fuentes LÁSER para asegurar una certificación compatible con los estándares. Guarde, cargue, gestione e imprima informes completos de certificación utilizando el software LinkWare para PC.

Proporciona una certificación completa de nivel 1

Nuestra completa solución de certificación de nivel 1 incluye mediciones de pérdida, longitud y polaridad. Se puede validar el rendimiento de los enlaces de fibra y la calidad de la instalación. Mida la pérdida óptica en varias longitudes de onda. Mida la longitud de la fibra y verifique la polaridad. Además, realice pruebas bidireccionales de fibras en dos longitudes de onda sin intercambiar las unidades de extremo cercano y alejado.

Encuentre los fallos más rápidamente

Un Localizador Visual de Fallos (VFL) integrado en el módulo hace que la solución de los problemas simples de enlace sea rápida e increíblemente fácil. El VFL de láser brillante le ayuda a localizar muchos fallos en el extremo cercano de la fibra y se puede usar para verificar la continuidad y la polaridad. Nuestro exclusivo diseño integrado garantiza que el VFL esté siempre a mano cuando lo necesite.







FLUKE networks.

Frecuencia de 900 MHz para una inversión pensada para el futuro

El DTX-1800 puede certificar incluso a 900 MHz. Este ancho de banda ampliado es compatible con los enlaces ISO clase F, así como con los nuevos estándares, como el de 10 GB Ethernet sobre cable de par trenzado. También admite aplicaciones como la distribución de vídeo sobre cable de par trenzado. Ningún otro certificador del mercado le ofrece una inversión de futuro como ésta y que, al mismo tiempo, cumpla las especificaciones actuales de las categorías 5e y 6.

Precisión siempre: hoy y mañana

Los certificadores de la serie DTX son más precisos que cualquier otro certificador de cableado en el mercado actual. La precisión global se debe a dos factores cruciales: primero, nivel de precisión del certificador, que asegura que el certificador proporciona los resultados PASA/FALLA correctos, y, segundo, una tecnología de adaptador de enlace permanente que, a su vez, es función de la calidad y las características de los adaptadores que se conectan al enlace que se está comprobando.

Resultados	Enlace cumple especif.	Enlace no cumple especif.
Pasa	Aceptar	"Pasa" falso: ¿Cuánto cuesta una devolución de llamada?
Falla	"Falla" falso: Tiempo de solución de problemas y recertificación innecesario	Aceptar

El precio de la imprecisión: un certificador cuyo resultado "Pasa/Falla" es incorrecto puede hacer que falle un enlace correcto ("falla" falso) o que un enlace malo se dé por bueno ("pasa" falso). Ambos resultados falsos le cuestan tiempo y dinero.

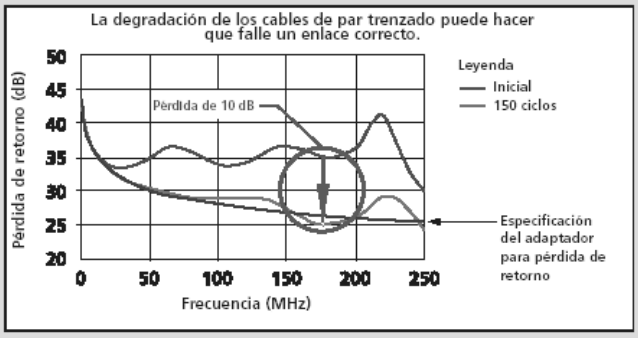
La precisión del certificador supera el nivel IV

Hasta ahora, los certificadores se han diseñado para cumplir el nivel III de precisión. La especificación más exigente en materia de precisión de nivel IV se ha propuesto en la ISO/IEC (estándares internacionales) para probar enlaces de clase F a 600 MHz. Los certificadores DTX superan esta nueva especificación, y le aseguran una mayor confianza en los resultados en toda la gama de frecuencias.

La tecnología de adaptador de enlace permanente elimina los fallos falsos.

Los latiguillos comerciales, incluidos los de Cat 5e, 6 y 7, que se emplean en las comprobaciones de enlace permanente, pueden provocar costosos fallos falsos, ya hacen que falle un buen enlace y aumentan las horas innecesarias para la solución de problemas y repetición de pruebas. El desgaste diario provocado por el enrollado-desenrollado y almacenamiento del cable produce una degradación invisible con el tiempo y contribuye a provocar pérdidas de retorno, que es con frecuencia la causa de un fallo.

En Fluke Networks, la precisión es lo primero. Por eso, hemos desarrollado unos adaptadores de enlace permanente de alto rendimiento en lugar de usar latiguillos comerciales. Nuestra tecnología patentada asegura la precisión y estabilidad extraordinarias que se precisan para medir la pérdida de retorno, lo que incrementa al mismo tiempo la exactitud del resto de parámetros medidos. **Precisión siempre: hoy y mañana.**



Como puede ver en este gráfico, un latiguillo comercial SSTP de Cat 7 tiene una pérdida de 10 dB tras 150 ciclos de enrollado-desenrollado. Esta degradación de 10 dB hace que falle un buen enlace, y acabará obteniendo un fallo falso. Estos 10 dB de pérdida no se producirán si utiliza la tecnología de adaptador de enlace permanente de Fluke Networks; con DTX dispondrá de una precisión repetible en la que puede confiar.

FLUKE networks.

El tiempo es dinero. DTX le ahorra ambos.

La serie DTX CableAnalyzer hace gala de un rendimiento superior que puede reducir los costes totales de certificación hasta en un 33% anual. Esta enorme reducción es el resultado no sólo de certificar más rápidamente, sino de una precisión mejorada, una capacidad de diagnóstico superior, una mayor duración de la batería, una interfaz de usuario sencilla y una rápida configuración y generación de informes. **DTX – El Tiempo es Oro.**



Serie DTX CableAnalyzer

Características del producto	DTX-1800	DTX-1200	DTX-LT
Tiempo de comprobación automática de Cat 6 (segundos)	12	12	30
Ancho de banda máximo (MHz)	900	350	350
Nivel de precisión	IV	IV	IV
Pantalla en color	•	•	•
Almacena datos gráficos de los resultados.	•	•	•
Capacidad de memoria interna (gráficos Cat 6)	250	250	250
Tarjeta de memoria extraíble Resultados gráficos de Cat 6 para cada múltiplo de 16 MB	300	300	n/d
Baterías de iôn de litio (duración)	12 Horas	12 Horas	12 Horas
Diagnósticos avanzados	•	•	•
Interfaz USB	•	•	•
Interfaz serie	•	•	n/d
Adaptador de enlace permanente Cat 6	•	•	•
Adaptador de canal Cat 6	•	•	Opción
Acepta módulos de fibra integrados	•	•	•
Inicio de comprobación automática en modo remoto	•	•	•
Intercomunicación entre unidades principal y remota	•	•	•
Accesorios estándares			
Módulo de certificación de fibra óptica (multimodo o monomodo)	Opción	Opción	Opción
Adaptador de enlace permanente Cat 7	Opción	n/d	n/d
Adaptador de canal Cat 7	Opción	n/d	n/d
Tarjeta de memoria multimedia	•	Opción	n/d
Cable USB	•	•	•
Cable de interfaz serie para ordenador	•	Opción	n/d
Auricular	•	•	Opción
Bolsa de transporte	•	•	•

Nota:
- • *La característica o el accesorio están disponibles en la configuración estándar del producto.*
- n/d *La característica no está disponible; el accesorio no es aplicable.*
- opción *El accesorio se puede adquirir aparte.*

La serie DTX protege el futuro de su inversión y le ofrece una ruta de actualización clara y sencilla desde cualquiera de los modelos a uno de prestaciones superiores, incluido el DTX-1800.

Experimente la potencia de DTX.

Le invitamos a comprobar las prestaciones exclusivas de nuestro DTX CableAnalyzer mediante una prueba virtual en www.flukenetworks.com/DTXlive.

Puede ponerse en contacto con nosotros en: Canada y USA: 1-800-508-040, España: +34 91 414 01 13 y le mostraremos cómo la tecnología DTX puede ayudarle. Si reside en otros países, entre en www.flukenetworks.com/contact para buscar un departamento comercial cercano. También puede suscribirse a un seminario gratuito de media jornada en el que se repasan los últimos avances en productos de comprobación y estándares, se realiza una demostración del DTX y una clase práctica. Para buscar un seminario en su zona, visite www.flukenetworks.com/workshops.

La serie DTX de Fluke Networks. La plataforma revolucionaria que reduce de forma significativa el tiempo total necesario para realizar una certificación, ahora y en el futuro.

Mejore su certificador DTX con servicio, soporte y formación.

Registre su DTX y reciba un regalo. Más acceso prioritario a nuestro Centro de Asistencia Técnica y una amplia base de datos de conocimiento en línea para obtener respuestas rápidas a las dudas. Inscríbase hoy mismo en www.flukenetworks.com/register.

La asistencia al cliente es fácil con nuestro programa de mantenimiento Gold NetworkSuperVision™. Asegura una gran tranquilidad con una calibración anual y unidades de préstamo gratis, asistencia ininterrumpida en el centro de asistencia técnica y descuentos especiales en formación y promociones. Visite www.flukenetworks.com/goldsupport para obtener más información. También puede ponerse en contacto con el departamento comercial en su país para obtener más información acerca de las modalidades, ventajas, precios y disponibilidad del programa Gold.

Forme a los instaladores con el programa CCTT de formación y certificación, de un día de duración. El curso, acreditado por BICSI, incluye formación teórica y práctica, y exámenes. Puede obtener más información en www.flukenetworks.com/CCTT. También puede ponerse en contacto con el departamento comercial más cercano para conocer la disponibilidad.

NETWORKSUPERVISION

Fluke Networks
P.O. Box 777, Everett, WA USA 98206-0777

Fluke Networks operates in more than 50 countries worldwide. To find your local office contact details, go to www.flukenetworks.com/contact.

©2004 Fluke Corporation. All rights reserved.
Printed in U.S.A. 4/2004 2141632 B-SPN-N A

Anexo B

Prácticas de

Nivelación

PRÁCTICA 1 DE NIVELACIÓN

Introducción a *Linux*

1.- Objetivo de Aprendizaje
El alumno:

- Al finalizar la práctica conocerá las características básicas del sistema operativo *Linux* y utilizará los comandos básicos para la creación y uso del sistema de archivos de *Linux*.

2.- Conceptos teóricos

La mayor parte de las computadoras que existen en la actualidad están diseñadas de forma que puedan ejecutar diversas tareas o programas. Estos programas pueden ir desde un procesador de textos, a un programa para la animación de gráficos tridimensionales o distintos tipos de juegos. Para su correcto funcionamiento deben ser además capaces de acceder a los recursos de que dispone la computadora, como por ejemplo escribir o leer datos en un disco duro, mostrar un gráfico por pantalla, etc. Es evidente, que si cada programa actuase de una forma independiente, existirían graves problemas y conflictos, puesto que, por ejemplo, tendrían libertad para escribir sus datos sobre los de otro.

Para solucionar este tipo de problemas se desarrollaron los *Sistemas Operativos*, los cuales aportan unos mecanismos y reglas básicas de funcionamiento, de forma que los programas puedan acceder a los recursos de la computadora de una forma adecuada.

En la actualidad existe una gran cantidad de sistemas operativos dependiendo del tipo de computadora en la que se va a ejecutar. Por ejemplo para los *PC* uno de los sistemas operativos más difundidos es *Microsoft Windows*. Otro de los sistemas operativos más poderosos y en rápida expansión para *PC, es LINUX.*

Usuarios y grupos

Linux es un sistema operativo *multitarea y multiusuario*. Esto quiere decir que es capaz de ejecutar varios programas (o tareas) de forma simultánea y albergar a varios usuarios de forma simultánea. Por lo tanto, todos los usuarios de *Linux* deben tener una cuenta de usuario en el sistema que establezca los privilegios del mismo. A su vez *Linux* organiza a los usuarios en *grupos* de forma que se puedan establecer privilegios a un determinado grupo de trabajo, para el acceso a determinados archivos o servicios del sistema.

Instalación

Uno de los conceptos principales a tener en cuenta antes de la instalación es el de *partición*. Cada sistema operativo organiza la información de los archivos que contiene de forma diferente, utilizando cada uno su propio sistema de archivos. Como referencia se indica a continuación el nombre del sistema de archivos de diferentes sistemas operativos(Ver Tabla No.1):

Sistema Operativo	**Sistema de archivos**
MS-DOS	FAT
MS Windows 95	VFAT
MS Windows 95 OSR2 y Windows 98	FAT32
MS Windows NT	NTFS
IBM OS/2 Warp	HPFS
Linux	Ext2 o Ext3

Tabla No.1

Esto en general impide que se puedan instalar varios sistemas operativos *mezclados* en un mismo disco duro. Para solucionar este problema existen las llamadas *particiones* con las que se divide un determinado disco duro de forma que pueda contener varios sistemas de archivos. A todos los efectos de realizar una partición es equivalente a que el disco duro se divida en más de una parte (aunque por supuesto no se divide de una forma física).

Existe un concepto fundamental a la hora de instalar y usar *Linux* que es el de *súper usuario* o usuario *root*. Este usuario es el administrador del sistema y se crea durante la instalación. Como administrador que es, puede acceder y modificar (así como destruir) toda la información del sistema.

Si se ha entrado como usuario *root* por primera vez tras instalar *Linux* el primer paso a seguir es crear una nueva cuenta de usuario que permita acceder al sistema como un usuario normal sin los privilegios del usuario root, de forma que se impida la modificación del sistema de forma inadecuada.

Iniciación a Linux

Existen tres formas de ingresar a un sistema *Linux*:

- A través de una consola de texto, el usuario se conecta directamente a la computadora que tiene instalado *Linux* y accede mediante un sistema no gráfico.
- Desde un gestor de sesiones gráfico *(X Window),* el usuario se conecta directamente a la computadora que tiene instalado *Linux* y accede al sistema mediante un programa gráfico.
- Desde una computadora remota mediante *telnet* o *secure shell*

Entrada y salida del sistema

El siguiente paso para usar su sistema *Linux* es iniciar una sesión. Cuando inicia una sesión, esta entrando al sistema (también llamado *Autenticación*). Sí escribe el nombre de usuario o contraseña incorrectamente, no estará autorizado para ingresar al sistema.

Nota: tanto *Linux*, como *UNIX*, distinguen entre letras mayúsculas y minúsculas. Por lo que root no es lo mismo que Root o ROOT.

Por motivos de seguridad la contraseña debe cumplir ciertas condiciones tales como:

- Contener al menos seis caracteres.
- Contener al menos un carácter numérico o especial y dos alfabéticos.
- Ser diferente del nombre de *login*.

Tipos de Archivos

La base del sistema de archivos de *Linux*, es obviamente el archivo, que no es otra cosa que la estructura empleada por el sistema operativo para almacenar información en un dispositivo físico como un disco duro, un diskette, un CD-ROM o un DVD. El sistema de archivos es la estructura que permite que *Linux* maneje los archivos que contiene.

Todos los archivos de *Linux* tienen un nombre, el cual debe cumplir unas ciertas reglas:

- Un nombre de archivo puede tener entre 1 y 255 caracteres.
- Se puede utilizar cualquier carácter excepto la barra inclinada */* y ***no*** es recomendable emplear los caracteres con significado especial en *Linux*, que son los siguientes: = \ ^ ~ ' " ` * ; - ? [ ] ( ) ! & ~ < >. Para emplear archivos con estos caracteres o espacios hay que introducir el nombre del archivo entre comillas.
- Se pueden utilizar números exclusivamente si así se desea. Las letras mayúsculas y minúsculas se consideran diferentes, y por lo tanto no es lo mismo *carta.txt* que *Carta.txt* ó *carta.Txt*

Como en *Windows*, se puede emplear un cierto criterio de "tipo" para marcar las distintas clases de archivos empleando una serie de caracteres al final del nombre que indiquen el tipo de archivo del que se trata. Así, los archivos de texto, HTML, las imágenes PNG o JPEG tienen extensiones *.txt, .htm (o .html), .png* y *.jpg (o .jpeg)* respectivamente.

Linux sólo distingue tres tipos de archivos:

- *Archivos* o archivos ordinarios, son los mencionados anteriormente.

- *Directorios* (o carpetas), es un archivo especial que agrupa otros archivos de una forma estructurada.
- *Archivos especiales*, son la base sobre la que se asienta *Linux*, puesto que representan los dispositivos conectados a una computadora, como puede ser una impresora. Para el usuario estos dispositivos tienen el mismo aspecto y uso que los archivos ordinarios.

Estructura del sistema de archivos

El sistema de archivos de *Linux* sigue todas las convenciones de *Unix*, lo cual significa que tiene una estructura determinada, compatible y homogénea con el resto de los sistemas *Unix*. Todo el sistema de archivos de *Unix* tiene un origen único, la raíz o root, representada por /. Bajo este directorio se encuentran todos los archivos a los que puede acceder el sistema operativo. Estos archivos se organizan en distintos directorios cuya misión y nombre son estándar para todos los sistemas *Unix*.

/ Raíz del sistema de archivos.

/dev Contiene archivos del sistema representando los dispositivos que estén físicamente instalados en el ordenador.

/etc Este directorio esta reservado para los archivos de configuración del sistema. En este directorio no debe aparecer ningún archivo binario (programas). Bajo este deben aparecer otros dos subdirectorios:

/etc/X11 Archivos de configuración de X *Window*

/etc/skel Archivos de configuración básica que son copiados al directorio del usuario cuando se crea uno nuevo.

/lib Contiene las librerías necesarias para que se ejecuten los programas que residen en */bin* (no las librerías de los programas de los usuarios).

/mnt Se refiere a sistemas de archivos montados temporalmente, tales como CD-ROMs y diskettes.

/opt Proporciona un área para almacenar habitualmente paquetes de software de una aplicación estática y amplia

/proc Contiene archivos especiales que o bien reciben o envían información al *kernel* del sistema (Se recomienda no modificar el contenido de este directorio y sus archivos).

/sbin Contiene programas que son únicamente accesibles al usuario root.

/usr Este es uno de los directorios más importantes del sistema puesto que contiene los programas de uso común para todos los usuarios. Su estructura suele ser similar a la siguiente:

/usr/X11R6 Contiene los programas para ejecutar X *Window*.

/usr/bin Programas de uso general, incluye el compilador de C/C++

/usr/doc Documentación general del sistema.

/usr/etc Archivos de configuración generales.

/usr/include Archivos de cabecera de C/C++ (.h).

/usr/info Archivos de información de GNU.

/usr/lib Librerías generales de los programas.

/usr/man Manuales accesibles con el comando *man.*

/usr/sbin Programas de administración del sistema.

/usr/src Código fuente de programas.

/usr/local Es para uso del administrador del sistema al instalar localmente el software. Necesita ser seguro para ser sobrescrito cuando el software del sistema es compartible entre un grupo de hosts, pero no se encuentra en /usr.

/var Este directorio contiene información temporal de los programas (lo cual no implica que se pueda borrar su contenido, de hecho, ¡no se debe hacer!).

Acceso a los diferentes sistemas de archivos

El sistema de archivos de *Linux* sólo tiene una raíz y su estructura es independiente de los dispositivos de almacenamiento existentes. Esto implica que el procedimiento a emplear para acceder a la información almacenada en los distintos sistemas de almacenamiento de una computadora no es tan sencillo como en *Windows*, y requiere un proceso llamado “montado”. Cuándo se ha terminado de trabajar con un determinado dispositivo hay que “desmontarlo” (¡No físicamente!).

Por ejemplo el proceso para leer un diskette sería el siguiente:

1. Introducir el diskette en el floppy.
2. Montar el sistema de archivos del mismo.
3. Leer, grabar, y manipular el contenido del diskette.
4. Desmontar el sistema de archivos del diskette.
5. Extraer el diskette del floppy.

El proceso puede parecer complejo pero es el precio a pagar por la seguridad, puesto que de esta forma se garantiza que no exista ninguna aplicación que esté usando el diskette cuando se extraiga. (En el caso de los CD-ROM *Linux* impide su extracción hasta que se desmonta).

Permisos

Linux, al igual que todos los sistemas *Unix*, mantiene un sistema de permisos de acceso a los archivos muy estricto, a fin de controlar qué es lo que se puede hacer con ellos, y quien lo puede hacer. Estos permisos se identifican con letras y son:

r	Permiso de lectura el archivo
w	Permiso de escritura en el archivo
x	Permiso de ejecución del archivo
s	Permiso para cambiar el propietario del archivo

A su vez cada uno de estos permisos se aplica: al dueño del archivo (*u*), al grupo de usuarios al que pertenece el dueño (*g*), ó al resto de usuarios (*a*). Así un archivo determinado puede tener permiso para ser leído, escrito y ejecutado por su dueño, leído y ejecutado por el grupo al que pertenece y no tener ningún tipo de acceso para los demás usuarios.

3.- Equipo y material necesario

Material que debe traer el alumno:

A. 1 diskette de 3 ½ (vació)
B. 1 Lápiz o pluma

Equipo del Laboratorio:

C. 1Maquina con Sistema Operativo *Linux Red Hat 9.0 ó superior.*

4.- Desarrollo

4.1 Modo de trabajar

Se trabajará en parejas o individualmente, dependiendo de la capacidad del grupo.

4.2 Realización de la práctica

4.2.1 Encienda el sistema y observe que en la pantalla aparecerá lo siguiente:

LILO boot:

dos linux (Queremos iniciar con linux)

4.2.2 Indique en el prompt la opción de cargar *Linux*, con la ayuda de las teclas de dirección, y oprima ENTER.

LILO boot: linux (Teclee <enter>)

4.2.3 Una vez que se han verificado todos los componentes del sistema, éste pasará del estado monousuario al estado X11. En este estado el sistema esta listo para aceptar sesiones de trabajo.

4.2.4 Aquí el sistema muestra la pantalla de inicio de sesión (The Graphical Login Screen) similar a la siguiente. Entre a sesión con una cuenta de usuario normal. (Ver figura No.1).

Nota: Para entrar a sesión el profesor ó administrador de la red le proporcionará la cuenta de usuario y contraseña correspondientes.



Figura No. 1

Teclee *redes* en el *login prompt*, presione <Enter>, teclee la contraseña *ingenieria* en el *password prompt*, y presione <Enter>.

Tras introducir el nombre del usuario y el password aparecerá una pantalla (The Graphical Desktop) similar a la siguiente: (Ver figura No.2)



Figura No. 2

4.2.5 Seleccione *Main Menu => Log Out*. Cuando aparezca la ventana de confirmación, seleccione la opción help.

Anote en las siguientes líneas la salida en pantalla y explique de donde proviene esta.

_____
_____
_____
_____
_____
_____
_____
_____
_____

4.2.6 Cierre la sesión. Seleccione *Main Menu => Log Out*.

Cuando aparezca la ventana de confirmación que se muestra a continuación, seleccione la opción *Log out* y de click en el botón *Yes*. Para guardar la configuración del escritorio, así como cualquier programa que esta corriendo, verifique la opción *Save current setup*. (Ver figura No.3).



Figura No.3

Cuando se sale del sistema vuelve a aparecer *The Graphical Login Screen*.

4.2.7 Abra una consola virtual. Oprima las teclas *<ctrl>+<alt>+<F1>.*

4.2.8 Inicie sesión desde la Consola Virtual con la misma cuenta de usuario normal anterior.

Fedora Core release 1 (Yarrow)
Kernel 2.4.22-1.2115.nptl on an i686
localhost login: (Teclee la cuenta "redes")
Password: (Teclea la contraseña "ingenieria")

4.2.9 Cierre la sesión
[redes@localhost redes]# exit

4.2.10 Vuelva al modo gráfico. Presione *<ctrl>+<alt>+<F7>.*

4.2.11 Entre a sesión como usuario root. El administrador le proporcionará la contraseña.

4.2.12 Abra un shell prompt seleccionando:
Main Menu => System Tools => Terminal.

Para salir del shell prompt teclee *exit* o presione *<ctrl>d* en el prompt.

4.2.13 Despliegue el tipo de CPU que tiene su equipo, teclee el siguiente comando:

[root@localhost root]# cat /proc/cpuinfo

A continuación anote un fragmento de la salida de pantalla y explique brevemente su contenido.

_____
_____
_____
_____
_____
_____
_____
_____

4.2.14 Inserte el disco de 3 ½ de alta densidad (vació) en la unidad de disco de la computadora. Teclee el siguiente comando:

[root@localhost root]# /sbin/mkfs -t ext2 /dev/fd0 1440

Anote y explique la salida en pantalla.

_____
_____
_____
_____
_____
_____
_____
_____
_____

4.2.15 Monte el sistema de archivos recién creado sobre el disco flexible en el subdirectorio /mnt. Y verifique que el kernel lo reconoce como parte del sistema de archivos. Teclee los siguientes comandos:

[root@localhost root]# /bin/mount -t ext2 /dev/fd0 /mnt
[root@localhost root]# /bin/mount

Apunte la salida y explique esta.

_____
_____
_____
_____
_____
_____
_____
_____

4.2.16 Copie el archivo /etc/passwd al disco flexible. Teclee el siguiente comando:

[root@localhost root]# cp /etc/passwd /mnt/passwd

4.2.17 Compruebe que el archivo se copió realmente. Teclee los siguientes comandos:

[root@localhost root]# cd /mnt
[root@localhost mnt]# ls –l

¿Cuál fue la salida de este último comando? ¿Realmente se copió el archivo?

_____
_____
_____
_____
_____
_____

4.2.18 Desmonte el sistema de archivos que se esta usando en el disco flexible. Teclee los siguientes comandos:

[root@localhost mnt]# cd
[root@localhost root]# umount /mnt

Una forma alternativa de hacer el último punto es:

umount /dev/fd0

4.2.19 Verifique la integridad del sistema de archivos creado en el disco flexible: Teclee el siguiente comando:

[root@localhost root]# fsck /dev/fd0

Anote la salida en pantalla y explique brevemente su contenido.

_____
_____
_____
_____
_____
_____

_____
_____
_____
_____

4.2.20 Monte nuevamente el sistema de archivos del disco flexible y ejecute el comando 'fsck' sobre éste sistema de archivos. ¿Qué sucede? ¿Por qué me pregunta si estoy seguro de continuar con la ejecución del comando?

_____
_____
_____
_____
_____
_____
_____
_____
_____

4.2.21 Desmonte el disco de 3 ½ y sáquelo del floppy.

4.2.22 Cierre el shell prompt y salga de la sesión.

4.2.23 Apague la computadora.

Desde *The Graphical Login Screen,* mostrado paginas atrás, seleccione *Shutdown* y de click en *OK* para confirmar.

Algunas computadoras se apagan automáticamente después de cerrar *Linux*. Si la computadora no lo hace, puede usted apagarla sin ningún riesgo después de ver el mensaje:

Power down.

5.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica

PRÁCTICA 1 DE NIVELACION

Introducción a *Linux*

Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. ¿Qué es *Linux*?

2. Mencione y explique las distribuciones más conocidas de *Linux*

3. ¿Cuáles son los diferentes tipos de entornos de trabajo en *Linux* y que diferencia existe entre ellos?

4. ¿Pueden existir varios usuarios root? ¿Por qué?

5. ¿Qué es GRUB?

6. ¿Es posible instalar *Linux* en un sistema de archivos de *Windows*?

7. Investigue que sucede si apaga la computadora sin cerrar la sesión

8. ¿Qué es un enlace y que es un path?

9. ¿Cuáles son las características del Sistema de Archivos?

10. ¿Cuáles son las características del Virtual File System (VFS)?

11. ¿Cuál es el propósito de los comandos *mount y umount*?

12. ¿Cuál es el propósito de los comandos *fdisk, mkfs y fsck*?

13. ¿En que situaciones es conveniente utilizar el comando *'fsck'*?.

14. Investigue como montar un disco flexible que tiene formato de DOS y poder copiar archivos de UNIX a DOS y viceversa.

15. Se puede verificar una partición montada vía NFS con el comando *fsck*. ¿Sí o no? ¿Por qué?

PRÁCTICA 2 DE NIVELACIÓN
Shell: Comandos Básicos de *Linux*

1.- Objetivo de Aprendizaje
El alumno:

- Al finalizar la práctica conocerá el funcionamiento de los comandos básicos de *Linux* y adquirirá otra alternativa de interacción con el sistema, además del ambiente gráfico.

2.- Conceptos teóricos

Existen una serie de nociones básicas que hay que tener en cuenta a la hora de introducir los comandos. En primer lugar citaremos las siguientes:

- Los comandos hay que teclearlos exactamente.
- Las letras mayúsculas y minúsculas se consideran como diferentes.
- En su forma más habitual (los *shells* de Bourne o de Korn), el sistema operativo utiliza un signo de *$* como *prompt* para indicar que está preparado para aceptar comandos, aunque este carácter puede ser fácilmente sustituido por otro u otros elegidos por el usuario.
- Cuando sea necesario introducir el nombre de un archivo o directorio como argumento a un comando, *Linux*, permite escribir las primeras letras del mismo y realiza un autorellenado al presionar la tecla del *tabulador*. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien.

Algunos Comandos Sencillos de LINUX

Para efectuar el cambio o la introducción de un password o contraseña se utiliza el comando *passwd*. El proceso a seguir es el siguiente:

Passwd nombreusuario

(current) UNIX password: (Se teclea la contraseña actual; no aparece en pantalla)
New UNIX password: (Se teclea la nueva contraseña; no aparece en pantalla)
Retype new UNIX password: (Se teclea de nuevo la nueva contraseña comprobando que se ha tecleado bien. Si no coincide no produce el cambio).

A continuación se describen algunos comandos sencillos que pueden ser útiles para familiarizarse con los comandos del sistema.

date Muestra por pantalla el día y la hora.
cal 1949 Muestra el calendario del año 1949.
cal 05 1949 Muestra el calendario de mayo de 1949.
who Indica qué usuarios tiene la computadora en ese momento, en qué terminal están y desde qué hora.
whoami Indica cuál es la terminal y la sesión en la que se está trabajando.
man comando Todos los manuales de *Linux* están dentro del propio sistema operativo, y este comando permite acceder a la información correspondiente al comando.
clear Este comando limpia la consola

Directorio Personal

El directorio personal es un directorio con un determinado nombre asignado a un usuario. Los directorios personales habitualmente son subdirectorios de */home* (en algunos casos se utiliza *mnt*, u otro subdirectorio de orden inferior).

Listado del contenido de directorios: comando ls

Una de las acciones más habituales a la hora de trabajar es mostrar el contenido de un directorio, existen herramientas gráficas con este fin, no obstante el *shell* incluye un programa con este mismo fin: *ls*,

ls Muestra los nombres de los archivos y subdirectorios contenidos en el directorio en el que se está. Sólo

se obtienen los nombres de los archivos, sin ninguna otra información (empleado el comando de esta forma, sin opciones).

Creación de subdirectorios. Comando mkdir

El comando *mkdir* (*make directory*) permite a cada usuario crear un nuevo subdirectorio:

mkdir subdir Donde *subdir* es el nombre del subdirectorio que se va a crear.

Borrado de subdirectorios. Comando rmdir

Este comando borra uno o más directorios del sistema (*remove directory*), siempre que estos subdirectorios estén vacíos.

rmdir subdir Donde *subdir* es el nombre del directorio que se va a eliminar.

Cambio de directorio. Comando cd

Este comando permite cambiar de directorio a partir del directorio actual de trabajo. Por ejemplo:

cd /home/redes En este ejemplo pasamos del directorio actual de trabajo al nuevo directorio */home/redes*, que será desde ahora nuestro nuevo directorio.

Situación actual. Comando pwd

El comando *pwd* (*print working directory*) visualiza o imprime la ruta del directorio en el que nos encontramos en este momento. Este comando es uno de los pocos que no tiene opciones y se utiliza escribiendo simplemente *pwd*.

Acceso a unidades de disco

Linux a diferencia de *Windows* no utiliza letras ("a:", "c:", "d:", ...) para acceder a las distintas unidades de disco de una computadora.

En *Linux* para acceder al contenido de una unidad de disco o de un CD-ROM este tiene que haber sido previamente "*montado*". El *montado* se realiza mediante el comando *mount*, con lo que el contenido de la unidad se pone a disposición del usuario en el directorio de *Linux* que se elija.

Copia de archivos. Comando cp

Este comando tiene la siguiente forma:

cp file1 file2 Hace una copia de *file1* y le llama *file2*. Si *file2* no existía, lo crea con los mismos atributos de *file1*. Si *file2* existía antes, su contenido queda destruido y es sustituido por el de *file1*.

Traslado y cambio de nombre de archivos. Comando mv

Este comando tiene una forma similar al anterior:

mv file1 file2 Realiza la misma función que el anterior (*cp*) pero además destruye el archivo original. En definitiva traslada el contenido de *file1* a *file2*; a efectos del usuario lo que ha hecho es cambiar el nombre a *file1*, llamándole *file2*.

Hay que recalcar que el comando *mv* sirve así mismo para cambiar el nombre de los archivos.

Enlaces a archivos. Comando ln

En *Linux* un mismo archivo puede estar repetido con más de un nombre, ya que con el comando *cp* se pueden realizar cuantas copias se desee del archivo. Pero, a veces, es más práctico tener un mismo archivo con varios nombres distintos, y lo que es más importante, poder acceder a él desde más de un directorio. En *Linux* esto recibe el nombre de enlaces múltiples a un archivo. La forma de este comando es,

ln file1 file2

A partir de este momento el archivo *file1* tiene dos nombres: *file1* y *file2*. A diferencia de los comandos *cp* y *mv*, este comando toma más precauciones, ya que advierte previamente si el nombre *file2* está ocupado, y en este caso no se ejecuta.

Borrado de archivos. Comando rm

Este comando tiene la forma siguiente:

rm file1 file2 Elimina uno o más archivos de un directorio en el cual tengamos permiso de escritura. Con este comando resulta facilísimo borrar archivos inútiles, y desgraciadamente, también los útiles.

Características de un archivo. Comando file

Este comando realiza una serie de comprobaciones en un archivo para tratar de clasificarlo. Su formato es:

file fich Muestra el tipo del archivo e información al respecto del mismo.

Cambio de modo de los archivos. Comandos chmod, chown y chgrp

Los permisos de cada archivo se pueden ver con el comando *ls -l*. Para cambiar los permisos de un archivo se emplea el comando *chmod*, que tiene el formato siguiente:

chmod [quien] oper permiso files

quien Indica a quien afecta el permiso que se desea cambiar. Es una combinación cualquiera de las letras *u* para el usuario, *g* para el grupo del usuario, *o* para los otros usuarios, y *a* para todos los anteriores. Si no se da él *quien*, el sistema supone *a*.

Oper Indica la operación que se desea hacer con el permiso. Para dar un permiso se pondrá un +, y para quitarlo se pondrá un -.

Permiso	Indica el permiso que se quiere dar o quitar. Será una combinación cualquiera de las siguientes letras: *r (read), w (write), x (executable), s (proprietor's change).*
files	Nombres de los archivos cuyos modos de acceso se quieren cambiar.

Por otra parte, el comando *chown* se emplea para cambiar de propietario ("*ch*ange *own*er") a un determinado conjunto de archivos. Este comando sólo le puede emplear el actual propietario de los mismos. Los nombres de propietario que admite *Linux* son los nombres de *usuario*, que están almacenados en el archivo */etc/passwd*. La forma general del comando *chown* es la siguiente:

chown newowner file1 file2 ...

Análogamente, el grupo al que pertenece un archivo puede ser cambiado con el comando *chgrp*, que tiene una forma general similar a la de *chown*,

chgrp newgroup file1 file2...

Espacio ocupado en el disco Comandos du y df

El comando *du* permite conocer el espacio ocupado en el disco por un determinado directorio y todos los subdirectorios que cuelgan de él. Para usarlo basta simplemente colocarse en el directorio adecuado y teclear, *du*, éste comando da el espacio de disco utilizado en bloques. Para obtener la información en bytes se debe emplear el comando con la opción *–h*: *du -h*

El comando *df* por el contrario informa del espacio usado por las particiones del sistema que se encuentren montadas.

Visualización sin formato de un archivo. Comando cat

Este comando permite visualizar el contenido de uno o más archivos de forma no formateada. También permite copiar uno o más

archivos como apéndice de otro ya existente. Algunas formas de utilizar este comando son las siguientes:

cat filename	Saca por pantalla el contenido del archivo *filename*.
cat file1 file2 >file3	El contenido de los archivos *file1* y *file2* es almacenado en *file3*.
cat >file1	Acepta lo que se introduce por el teclado y lo almacena en *file1* (se crea *file1*). Para cerrar la aplicación se emplea <*ctrl*>*d*

Comando head

head -7 filename Escribe las 7 primeras líneas del archivo filename

Visualización de archivos con formato. Comando pr

Este comando, a diferencia de *cat*, imprime por consola el contenido de los archivos de una manera formateada, por columnas, controlando el tamaño de página y poniendo cabeceras al comienzo de las mismas. Está muy en relación con el comando *lp* de salida por impresora. Las formas más importantes que admite son las siguientes:

pr file	Produce una salida estándar de 66 líneas por página, con un encabezamiento de 5 líneas (2 en blanco, una de identificación y otras 2 líneas en blanco).
pr -ln file	Produce una salida de n líneas por página (cuando el tamaño de papel de impresora, por ejemplo, tiene un número de líneas distinto de 66).
pr -p file	Hace una pausa para presentar la página, hasta que se pulsa <*return*> para continuar.

Visualización de archivos pantalla a pantalla. Comandos more y less

Estos comandos permiten visualizar un archivo pantalla a pantalla. El número de líneas por pantalla es de 23 líneas de texto y

una última línea de mensajes, donde aparecerá la palabra more. Cuando se pulsa la barra espaciadora (el espacio en blanco), se visualizará la siguiente pantalla. Para salir de este comando (terminar la visualización) se pulsa *<ctrl>d* o *q*. Por ejemplo: *more file.* El comando *less* es muy similar al anterior pero permite el desplazamiento a lo largo del texto empleando las teclas de cursores pudiendo desplazarse hacia arriba o abajo de un archivo.

Búsqueda en archivos. Comandos grep, fgrep y egrep

El comando *grep* localiza una palabra, clave o frase en un conjunto de directorios, indicando en cuáles de ellos la ha encontrado. Este comando rastrea archivo por archivo, por turno, imprimiendo aquellas líneas que contienen el conjunto de caracteres buscado. Si el conjunto de caracteres a buscar está compuesto por dos o más palabras separadas por un espacio, se colocará el conjunto de caracteres entre apóstrofes ('). Su formato es el siguiente:

grep 'conjuntocaracteres' file1 file2 file3

siendo 'conjuntocaracteres' la secuencia de caracteres a buscar, y *file1*, *file2*, y *file3* los archivos donde se debe buscar.

Comandos tar y gzip

Tanto el comando *tar* como *gzip* son ampliamente utilizados para la difusión de programas y archivos en *Linux*. El primero de ellos agrupa varios archivos en uno solo o "*archivo*", mientras que el segundo los comprime. En conjunto estos dos programas actúan de forma muy similar a programas como *Winzip*. Para crear un nuevo archivo se emplea:

tar –cvf nombrearchivo.tar archivo1 archivo2 …

donde archivo1, archivo2, etc. son los archivos que se van a añadir al archivo *tar*. Si se desea extraer los archivos se emplea:

tar –xpvf nombrearchivo.tar archivo1 …

Al contrario que *tar* que agrupa varios archivos en uno, *gzip* comprime un único archivo con lo que la información se mantiene pero se reduce el tamaño del mismo. El uso de *gzip* es muy sencillo

gzip archivo

con lo que se comprime *archivo* (que es borrado) y se crea un archivo con nombre *archivo.gz*. Si lo que se desea es descomprimir un archivo se emplea entonces:

gzip –d archivo.gz

recuperando el archivo inicial. Como se ha comentado al principio es típico emplear *tar* y *gzip* de forma consecutiva, para obtener archivos con extensión *tar.gz* o *tgz* que contienen varios archivos de forma comprimida (similar a un archivo *zip*). El comando *tar* incluye la opción z para estos archivos de forma que para extraer los archivos que contiene se teclea lo siguiente:

tar –zxf archivo.tar.gz

Comando de impresión lpr.

El comando *lpr* se emplea para imprimir una serie de archivos. Si se emplea sin argumentos imprime el texto que se introduzca a continuación en la impresora por defecto. Por el contrario,

lpr nombrearchivo Imprime en la impresora por defecto el archivo indicado.

3.- Equipo y material necesario

Material que debe traer el alumno:

- ***A.*** 1 diskette de 3 ½ (con al menos 2 archivos guardados sobre Windows)
- ***B.*** 1 Lápiz o pluma

Equipo del Laboratorio:

C. 1 Máquina con Sistema Operativo *Linux Red Hat 9.0 ó superior.*

4.- Desarrollo

4.1 Modo de trabajar

Se trabajará en parejas o individualmente, dependiendo de la capacidad del grupo.

4.2 Realización de la práctica

4.2.1 Encienda el sistema y elija la opción de cargar *Linux*.

4.2.2 Entre a sesión como usuario normal. La cuenta y la contraseña serán proporcionadas por el profesor.

4.2.3 Abra un shell prompt seleccionando:

Main Menu => System Tools => Terminal.

Para salir del shell prompt teclee exit o presione <ctrl>d en el prompt.

4.2.4 Ahora podemos emplear el sistema para ejecutar los comandos que necesitemos utilizar.

4.2.5 Liste los archivos del directorio actual.

[redes@localhost redes]$ ls (Lista el directorio actual "redes")

4.2.6 Cree el subdirectorio pract2 dentro del directorio actual y liste el directorio actual (observe que se ha creado el archivo pract2)

[redes@localhost redes]$ mkdir pract2 (Crea el subdirectorio pract2)

[redes@localhost redes]$ ls (Lista el directorio redes)

4.2.7Pase al subdirectorio pract2 y liste los archivos que contiene (observará que esta vació).

[redes@localhost redes]$ cd pract2 (Cambia al subdirectorio pract2)

[redes@localhost pract2]$ ls (Lista el directorio pract2)

4.2.8 Inserte el disco 3 ½ en el floppy y móntelo. Si el sistema no puede montarlo, llame al administrador, tal ves su disco este dañado.

[redes@localhost pract2]$ mount /mnt/floppy (Monta el disco 3 ½)

4.2.9 Cámbiese al directorio del disco 3 ½ y lístelo para ver sus archivos.

[redes@localhost pract2]$ cd /mnt/floppy (Cambia al directorio del disco 3 ½)

[redes@localhost floppy]$ ls (Lista los archivos del disco 3 ½)

4.2.10 Copie uno de sus archivos del disco al subdirectorio pract2.

[redes@localhost floppy]$ cp file /home/redes/pract2 (Copia file en el subdir pract2).

4.2.11 Cámbiese al subdirectorio pract2, desmonte el disco 3 ½ y sáquelo del floppy.

[redes@localhost floppy]$ cd /home/redes/pract2 (Cambia al subdir pract2)

[redes@localhost pract2]$ umount /mnt/floppy
(Desmonta el disco 3 ½)

4.2.12 Liste el subdirectorio pract2 (observe que se copio el archivo file), cambié el nombre al archivo file por prueba y vuelva a listar el subdirectorio pract2 (observe que cambio de nombre el archivo file).

[redes@localhost pract2]$ ls (Lista el subdir pract2)
[redes@localhost pract2]$ mv file prueba (Cambia de nombre a file por prueba)
[redes@localhost pract2]$ ls (Lista el subdir pract2)

4.2.13 Retroceda un nivel, es decir, pase al directorio redes, liste este directorio (observe los permisos del subdirectorio pract2), quítele el permiso de borrado al subdirectorio pract2, intente borrar el archivo prueba y observe el mensaje del prompt.

[redes@localhost pract2]$ cd .. (Pasa al directorio redes)
[redes@localhost redes]$ ls –la (Lista los archivos del directorio redes)
[redes@localhost redes]$ chmod –w pract2
(Quita el permiso de borrado al subdir pract2)
[redes@localhost redes]$ cd pract2 (Cambia al subdir pract2)
[redes@localhost pract2]$ rm –i prueba (“Borra” el archivo prueba)

4.2.14 Retroceda un nivel, liste el directorio redes (observe que paso con los permisos del subdirectorio pract2) y devuelva el permiso de borrado al subdirectorio pract2.

[redes@localhost pract2]$ cd .. (Pasa al directorio redes)
[redes@localhost redes]$ ls –la (Lista los archivos del directorio redes)

[redes@localhost redes]$ chmod +w pract2 (Da el permiso de borrado al subdir pract2)

4.2.15 Cree el archivo prueba2 en el subdir pract2 con el comando cat.

[redes@localhost redes]$ cd pract2 (Cambia al subdir pract2)
[redes@localhost pract2]$ cat >prueba2 (Crea el archivo prueba2)

4.2.16 Escriba en el prompt lo que desee y cuando haya finalizado teclee *<ctrl>d.*

4.2.17 Liste el subdirectorio pract2 (observe que se ha creado el archivo prueba2) y visualice el archivo prueba2 con el comando less.

[redes@localhost pract2]# ls (Lista el subdirectorio pract2)
[redes@localhost pract2]# less prueba2 (Visualiza el contenido del archivo prueba2)

Pulse la tecla q para salir del comando.

4.2.18 Agrupe los archivos prueba y prueba2 en un archivo llamado final.tar, liste el subdirectorio pract2 (observe que se creo el archivo final.tar), a este ultimo comprímalo, liste el subdirectorio pract2 (observe que se borro el archivo final.tar y en su lugar se creo el archivo final.tar.gz).

[redes@localhost pract2]$ tar –cvf final.tar prueba prueba2 (Agrupa los archivos prueba y prueba2 en uno solo nombrado final.tar)

[redes@localhost pract2]$ ls (Lista el subdir pract2)
[redes@localhost pract2]$ gzip final.tar (Comprime el archivo final.tar)

[redes@localhost pract2]$ ls (Lista el subdir pract2)

4.2.19 Borre los archivos prueba y prueba2, a continuación liste el subdirectorio pract2 (observe que solo queda el archivo final.tar.gz).

[redes@localhost pract2]$ rm prueba prueba2 (Borra los archivos prueba y prueba2)
[redes@localhost pract2]$ ls (Lista el subdir pract2)

4.2.20 Descomprima el archivo final.tar.gz y liste el subdirectorio pract2 (observe que se conserva el archivo final.tar.gz y además se extrae su contenido, prueba y prueba2).

[redes@localhost pract2]$ tar –zxf final.tar.gz
(Descomprime el archivo final.tar.gz)

[redes@localhost pract2]$ ls (Lista el subdir pract2)

4.2.21 Borre el subdirectorio pract2 (recuerde que no puede borrar un directorio si **NO** esta vació) y cierre el shell prompt.

*[redes@localhost pract2]$ rm –i ** (Borra todos los archivos del subdir pract2)
[redes@localhost pract2]$ cd .. (Cambia al directorio redes)
[redes@localhost redes]$ rmdir pract2 (Borra el subdir pract2)
[redes@localhost redes]$ exit (Cierra el shell prompt)

4.2.22 Cierre la sesión y apagué la computadora como lo hizo en la práctica 3.

5.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica

PRÁCTICA 2 DE NIVELACION
Shell: Comandos Básicos de Linux
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. Suponga una carpeta con los siguientes directorios:
   Programas
   Documentos_proyecto
   Documentos_privados

   ¿Qué sucede si escribe *cd Pr<tab>* y qué sucede si escribe *cd D<tab>*?

2. ¿Pueden varios usuarios trabajar dentro de un mismo directorio? ¿Por qué?

3. El comando *clear* limpia la consola, ¿Qué comando borra definitivamente la lista de comandos tecleados en la consola?

4. Investigue como funciona el comando *man*.

5. Además del comando *man,* ¿existen otros comandos que realicen la misma función?

6. Describa que información nos da de un archivo el comando ls –lac.

7. ¿Qué diferencia existe al teclear *rm file1* y *rm –i file1*?

8. ¿En *Linux* es posible combinar varias opciones en un mismo comando?
9. ¿Existe la forma *cd..* en el S.O *Linux*?

10.¿Es posible sacar la unidad de cd rom sin antes desmontarla?¿Por qué?

11. Tienen el mismo significado los permisos de lectura, escritura y ejecución cuando se aplican a directorios y no a archivos normales?

12.¿En qué archivo se encuentran los grupos de usuarios?

Prácticas Sugeridas

PRÁCTICA 1
Construcción de cables UTP para conexión directa y cruzada

1.- Objetivo de Aprendizaje
El alumno:

Al finalizar la práctica tendrá la capacidad de construir cables de conexión directa y cruzada con normas ANSI/EIA/TIA T568-A y T568-B.

2.- Conceptos teóricos

El cableado es normalmente el medio por el cual la información se mueve de un dispositivo de red a otro. El tipo de cable dependerá de diversos factores como la topología, la tecnología, el tamaño de la red, la velocidad de operación, etc.

*La construcción del cable de red UTP de conexión directa (*en inglés *straight-through),* se usa para conectar la tarjeta de red o *NIC* (en inglés *Network Interface Card*), de la estación de trabajo al *jack* de datos de la placa de pared o bien para conectar el *patch panel* a un *hub* o *switch Ethernet*. Las salidas de pin serán T568-B y los 8 hilos se deben terminar con conectores modulares RJ45. Sólo 4 de los 8 hilos se usan para el estándar Ethernet 10/100BASE-T. Los 8 hilos se usan para el estándar Ethernet 1000BASE-T.

Los cables se encuentran alambrados como cables de conexión directa, ya que el cable desde la estación de trabajo hasta el concentrador se cruza normalmente de forma automática en el concentrador. Se debe observar que los puertos en la mayoría de los concentradores tienen una X al lado. Esto significa que los pares de emisión y recepción se cruzarán cuando el cableado llegue al concentrador. (Ver figura No. 1).

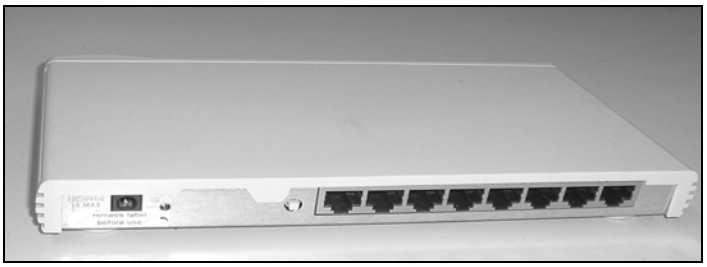


Figura No.1. Concentrador.

*Un cable de interconexión cruzada (*en inglés *crossover)* se puede utilizar como cable principal para conectar dos *hubs* o *switches* en una *LAN* y para conectar dos estaciones de trabajo aisladas para crear una *miniLAN*. Esto permite conectar dos estaciones de trabajo entre sí, o una estación de trabajo con un servidor sin que sea necesario que haya un hub entre ellos.

El cable cruzado (*crossover*), cruza las terminales de transmisión de un lado para que llegue a recepción del otro, y viceversa.

3.- Equipo y material necesario

Material que debe traer el alumno:

A. 12 metros de cable UTP Cat 5e ó superior.
B. 12 conectores RJ-45.
C. Pinzas de corte.
D. Pinzas de punta.
E. Flexómetro.
F. Multímetro.

Equipo del Laboratorio:

G. Pinzas de engarzadoras.
H. Analizador de continuidad de cableado UTP ó TESTER de cableado UTP que puede probar cables de tipo de conexión directa (*straight-through*) y de interconexión cruzada *(crosssover)* T568-A ó T568-B. (Ver figura No.2).

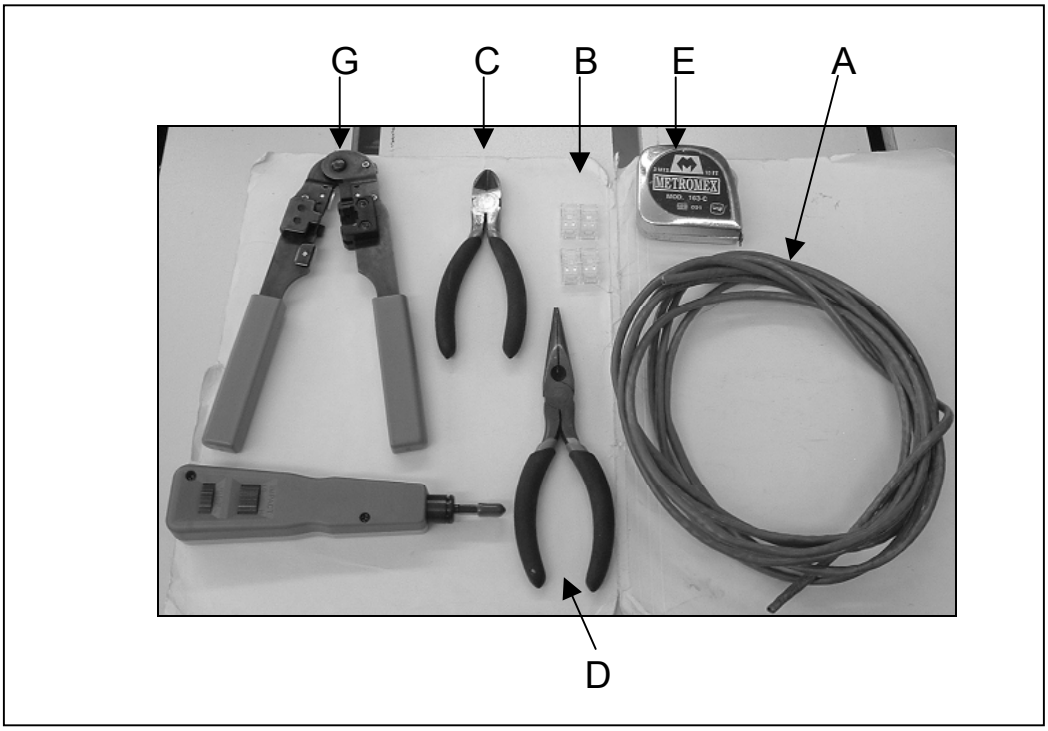


Figura No.2. Material y Equipo necesario.

4.- Desarrollo

4.1 Modo de trabajar

La construcción de los cables se realizará de manera individual.

4.2 Construcción de Cables

El cable categoría 5e o superior está formado de cuatro pares trenzados formando una sola unidad. Estos cuatro pares vienen recubiertos por un tubo de plástico que mantiene el grupo unido, mejorando la resistencia ante interferencias externas. Es importante notar que cada uno de los cuatro pares tiene un color diferente, pero a su vez, cada par tiene un cable de un color específico y otro cable blanco con algunas franjas del color de su par. Esta disposición de los cables permite una adecuada y fácil identificación de los mismos con el objeto de proceder a su instalación. El número de identificación de cada par referente a su color se muestra en la figura No.7.

4.3 Cable de Conexión Directa (straight through) Categoría 5e ó superior

A continuación se construirán cuatro cables de conexión directa: dos siguiendo la configuración T568-A y dos con la configuración T568-B. (Los cuales serán utilizados en la práctica No.2).

4.3.1 Construcción de cables UTP con código de colores T568-A

4.3.1.1 Corte un trozo de cable de par trenzado no blindado Cat. 5e o superior de una longitud de 2 metros. (Ver figura No.3).

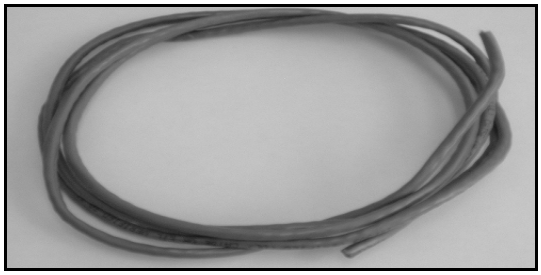


Figura No.3. Cable UTP Cat. 5e ó superior.

4.3.1.2 Retire 3 cm. de la envoltura de uno de los extremos del cable (Ver figura No. 4).

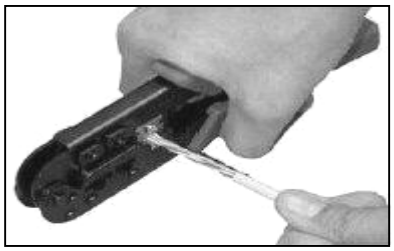


Figura No.4.

4.3.1.3 Sostenga la envoltura y el cable, destrence y ordene los pares de hilos (ver figuras No. 5 y 6) de modo que cumplan con el diagrama de color del cableado T568-A. (Ver figura No.7).

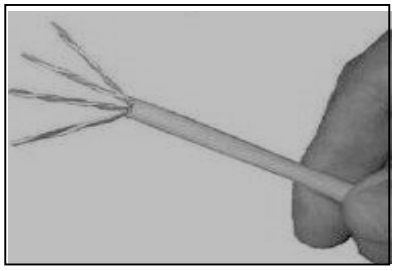


Figura No.5. Cable UTP con sus 4 pares de cables trenzados.

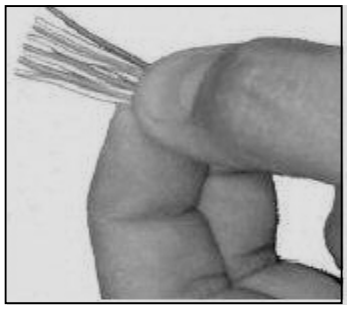


Figura No. 6. Pares ordenados para la conexión directa.

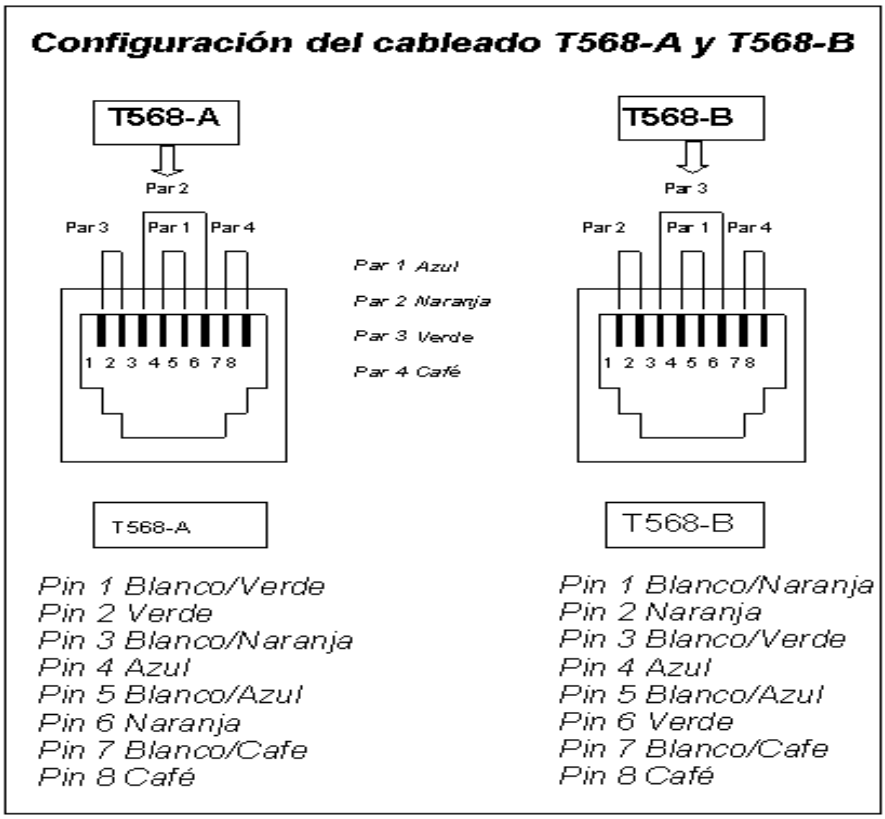


Figura No.7 Configuración del cableado T568-A y T568-B.

4.3.1.4 Aplane, enderece y haga coincidir los hilos empleando las pinzas de punta, luego recórtelos en línea recta alrededor de 1.20 cm. – 1.90 cm. del borde de la envoltura. (Ver figura No.8).

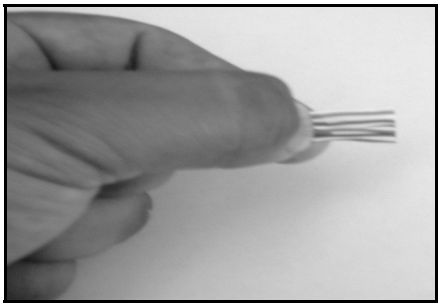


Figura No.8.

4.3.1.5 Coloque un conector RJ-45 en el extremo del cable, con la lengüeta hacia abajo. (Ver figura No. 9).

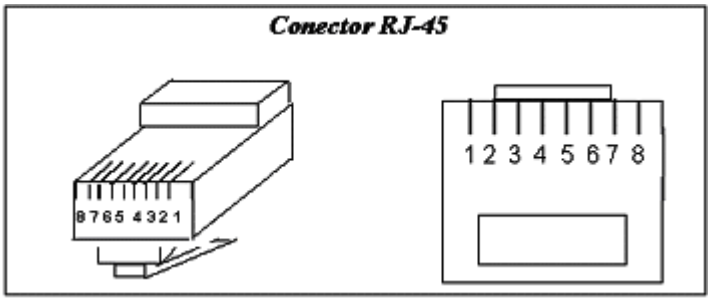


Figura No.9.

4.3.1.6 Empuje suavemente los hilos dentro del conector hasta que pueda ver los extremos de cobre de éstos a través del extremo del conector. Asegúrese de que el extremo de la envoltura esté ubicado dentro del enchufe y de que todos los hilos estén en el orden correcto. (Ver figura No.10).

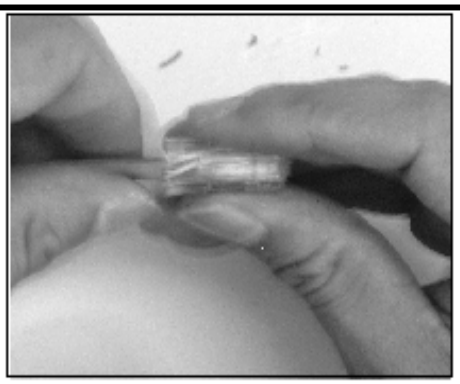
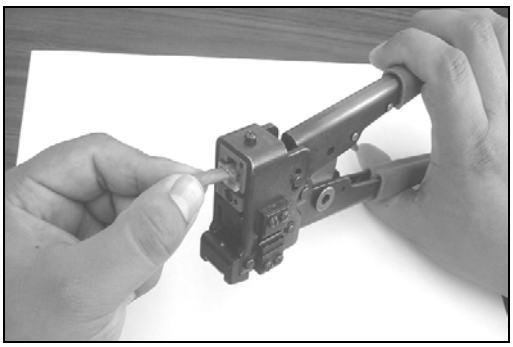
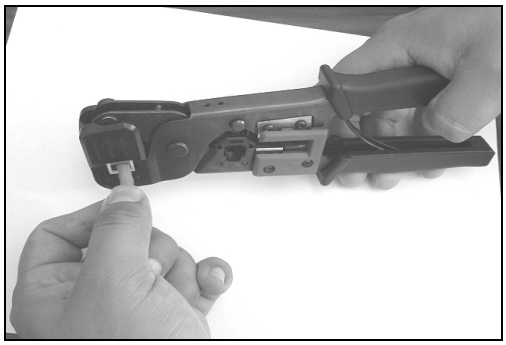


Figura No.10.

4.3.1.7 Utilice las pinzas engarzadoras y apriete el conector con suficiente fuerza como para forzar los contactos a través del aislamiento en los hilos, completando así el camino conductor. (Ver figuras No.11 y 12).





Figuras No.11 y 12. Utilización de pinzas para apretar el conector RJ-45.

Finalizando así la construcción de un extremo del cable. (Ver figura No.13).

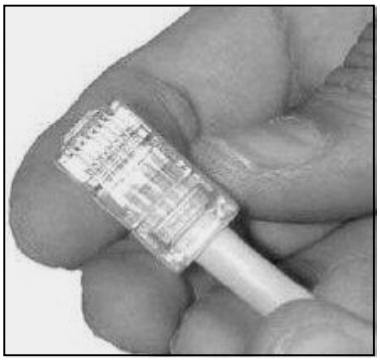


Figura No.13. Extremo del cable de conexión directa terminado.

4.3.1.8 Repita los pasos 4.3.1.2 a 4.3.1.7 para construir el otro extremo del cable de conexión directa con la configuración T568-A, ordénelos de acuerdo con la figura No 7. según se muestra. Finalizando así con el otro extremo del cable de conexión directa (*straight through*).

4.3.1.9 Repita los pasos 4.3.1.1 a 4.3.1.8 para la construcción de un cable más con la configuración T568-A.

4.3.2 Construcción de cables UTP con código de colores T568-B

4.3.2.1 Repita los pasos 4.3.1.1 a 4.3.1.9 empleados para la construcción de 2 cables de conexión directa, ahora utilizando la configuración T568-B en ambos extremos del cable.

4.3.3 Cable de Conexión Cruzada (crossover)

Se construirá un cable:

4.3.3.1 Corte un trozo de cable de par trenzado no blindado Cat. 5e ó superior de una longitud de 2 metros.

4.3.3.2 Retire 3 cm. de la envoltura de uno de los extremos del cable.

4.3.3.3 Sostenga la envoltura y el cable, destrence y ordene los pares de hilos (ver figuras No.14 y 15) de modo que cumplan con el diagrama de color de cableado de la figura No.16. (Conexión *crossover* para cable 1000Base-T).

Este *crossover* funcionará para redes 10/100/1000 Mbps. (Utilizaremos los 8 hilos).

Luego reorganice los pares del *Extremo No. 1* del cable según el orden de la conexión *crossover* de la figura No. 16. Note que el Extremo No.1 tiene la Configuración T-568-B.

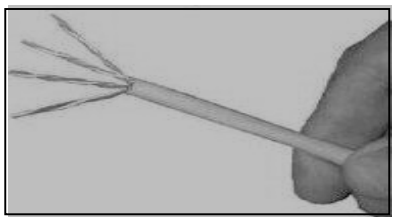


Figura No.14. Cable UTP con sus 4 pares de cables trenzados.

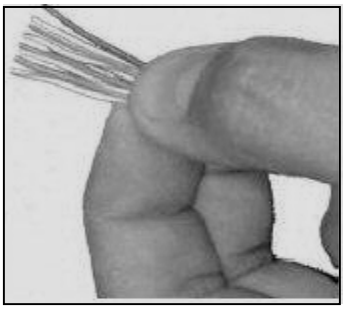


Figura No. 15. Pares ordenados para la conexión *crossover*.

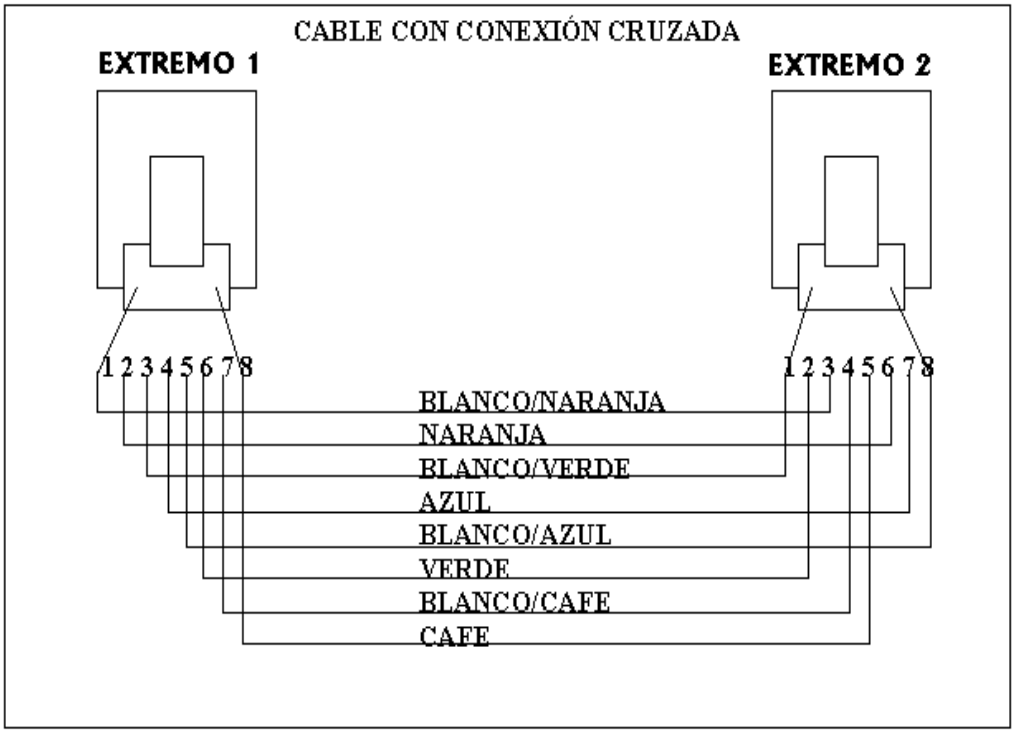


Figura No.16. Configuración de un cable *crossover* (10/100/1000 Mbps).

La siguiente tabla muestra el orden de los pines del cable *crossover* (Ver Tabla No.1).

Extremo No.1 (T-568-B)	*Extremo No.2*
1	3
2	6
3	1
4	7
5	8
6	2
7	4
8	5

Tabla No.1 Orden de los pines para el Extremo No.1 y el Extremo No.2 del cable *crossover.*

NOTA.- De manera práctica para redes 10/100Mbps. Se acostumbra rematar un extremo con el estándar de cableado T568-A y el otro con el estándar de cableado T568-B. (Recuerde que a 10/100 Mbps. Sólo se utilizan 4 hilos).

4.3.3.4 Repita pasos 4.3.1.4 a 4.3.1.7 del cable de conexión directa.

Finalizando así la construcción del extremo1 del cable. (Ver figura No.17).

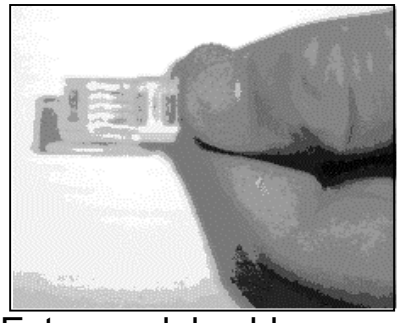


Figura No.17. Extremo del cable *crossover* terminado.

4.3.3.5 Repita los pasos 4.3.3.2 a 4.3.3.4 para construir el *Extremo No.2* del cable, ordénelos de acuerdo con la

figura No 16. según se muestra. Finalizando así el cable de conexión cruzada (*crossover*).

5.- Pruebas y Aplicaciones

5.1. Finalmente pruebe los cables terminados empleando el analizador de continuidad Ethernet.

5.2 Otra forma de probar los cables y en el caso de no contar con un analizador de continuidad, es empleando un multímetro midiendo la continuidad en cada uno de los hilos del par trenzado.

5.3. También mediante un escáner se puede hacer la prueba de los cables UTP ya sea par a par o alambre por alambre y verificar si tienen continuidad. El escáner hace varias pruebas, que indicará si las pasa o no .

En las pruebas de continuidad del multímetro o tester; si falla una conexión, el cable estará mal construido y habrá que repetir su construcción nuevamente.

6.- Cuestionario

1. ¿Cuál es la función de trenzar los filamentos en el cable UTP?

_____
_____
_____
_____

2. ¿Cuál es la diferencia que existe al emplear el código de colores T568-A y T568-B dentro del cableado estructurado?

_____
_____
_____
_____

3. Explique el fenómeno de la atenuación.

_____
_____
_____
_____

7.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica

_____
_____
_____
_____
_____
_____
_____

PRÁCTICA 1
Construcción de cables UTP para conexión directa y cruzada
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. ¿Qué son las Interferencias Electromagnéticas (EMI - Electromagnetic Interferences) y cómo afectan a la comunicación?

2. ¿Qué es un par trenzado: UTP (*UnShielded Twisted Pair*) Cable par trenzado no apantallado y STP (*Shielded Twisted Pair*) Cable de par trenzado apantallado?

3. ¿Cuántas categorías de cables UTP existen? Explique más a detalle las principales aplicaciones de los cables de la categoría UTP 5e (5 *enhance* - mejorada) y UTP 6.

4. Explique el fenómeno de la diafonía.

5. Explique el fenómeno de diafonía de extremo cercano. (*Near - End Crosstalk* -NEXT).

6. Explique el fenómeno de diafonía de extremo lejano (*Far-End Crosstalk* -FEXT).

PRÁCTICA 2
Componentes del Cableado Estructurado
Norma ANSI/EIA/TIA 568

1.- Objetivo de Aprendizaje
El alumno:

Al finalizar la práctica se familiarizará con aspectos generales del cableado estructurado mediante la instalación de un *jack* y roseta, utilizando cable UTP Cat. 5e ó superior.

2.- Conceptos teóricos

El *cableado* es normalmente el medio por el cual la información se mueve de un dispositivo de red a otro. Existen una serie de componentes que ayudan a organizar las grandes instalaciones UTP y facilitar su manejo.

Los *armarios* y *racks de distribución* ayudan a organizar una red que tiene muchas conexiones. Montando en ellos los dispositivos de red como routers, switchs, hubs, patch panels, etc.

En un cableado estructurado, la función principal de las canaletas es la de proteger el cableado. Las *canaletas* son conductos o tubos, normalmente de plástico, que rodean el cable; variando su diámetro de acuerdo a la cantidad y grosor de los cables que conducen y normalmente se colocan aprovechando las esquinas de las paredes de los edificios. Suelen ser más baratos que la ductería y de acceso más fácil; pero están expuestos a las interferencias electromagnéticas (EMI). (Ver figura No.1).

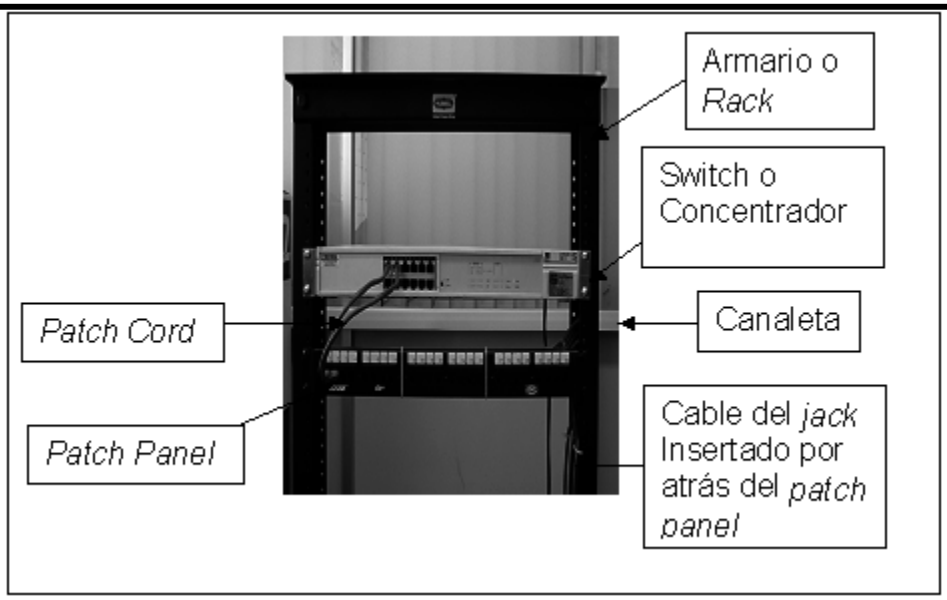


Figura No. 1 Dispositivos de red montados en un Armario o *Rack.*

Un *jack o keystone RJ45* es un dispositivo modular de conexión mono-línea, hembra, apto para conectar plugs RJ45, que permite su inserción en rosetas y frentes de "*patch panels*" especiales mediante un sistema de encaje ó empotre. Permite la colocación de la cantidad exacta de conexiones necesarias.

La *roseta o faceplate* es una pieza plástica de soporte que se fija a la pared y permite colocar *jacks* o *keystones*. Se conectan a los *patch panels* y placas de pared. Una de sus principales funciones es que pueden ser etiquetadas para facilitar el control de los puntos de conexión de nuestra red.

3.- Equipo y material necesario

Material que deberá traer el alumno:

A. 2 Conectores jack RJ45 Marca 3M con
No. de parte VOL-OCK6-U o similar.
B. 5 Metros de cable UTP Cat 5e ó superior.
C. Pinzas de corte.
D. 2 Cables de conexión directa configuración
T568- B construidos en la práctica 1.
E. Flexómetro.

F. Pinzas de punta.

Equipo del Laboratorio:

G. 1 Computadora con Windows 98/NT/2000P/XP.
H. 1 Switch.
I. 1 Patch Panel ó Panel de Parcheo.
J. Pinzas para jack ó Pinzas de impacto.
K. Pinzas engarzadoras. (Ver figura No. 2).

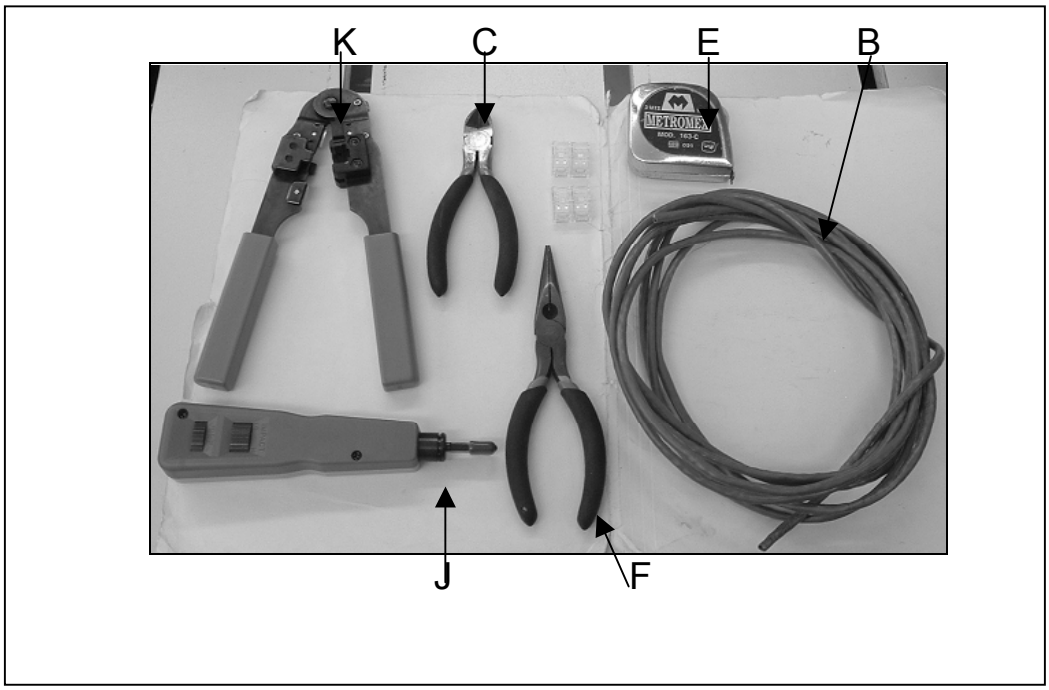


Figura No.2. Material y Equipo necesario.

4.- Desarrollo

4.1 Modo de trabajar

La instalación del jack RJ-45 se realizará de manera individual.

4.2 Instalación del Jack RJ-45

A continuación se explicará la instalación del jack RJ-45. Utilizando la configuración T568-B.

4.2.1 Corte un trozo de cable de par trenzado no blindado Cat. 5e ó superior de una longitud de 2 metros. (Ver figura No.3).

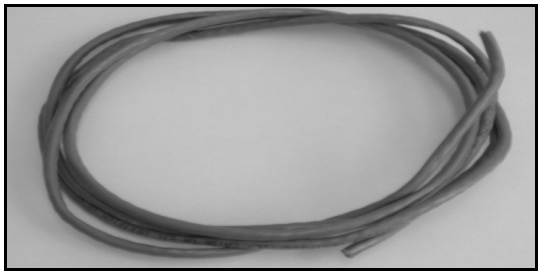


Figura No.3. Cable UTP Cat. 5e ó superior.

4.2.2 Retire 3 cm. de la envoltura de uno de los extremos del cable (Ver figura No. 4).

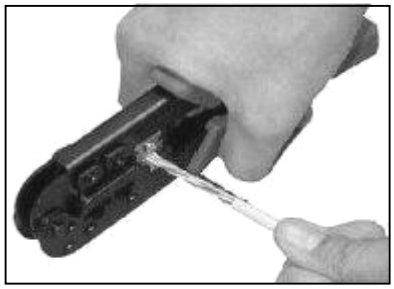


Figura No.4. Retirar la envoltura del Cable UTP.

4.2.3 Sostenga la envoltura y el cable, destrence y ordene los pares de hilos (Ver figuras No.5 y 6) de modo que cumplan con el diagrama de color del cableado T568-B. Como se muestra en la práctica 1.

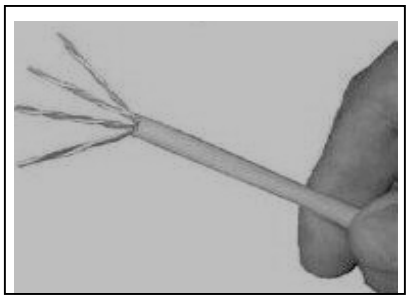


Figura No.5. Cable UTP con sus 4 pares de cables trenzados.

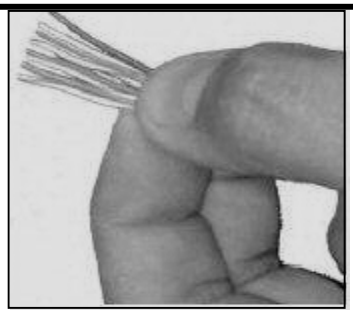


Figura No. 6. Pares ordenados para la conexión del jack RJ45. Siguiendo la configuración T568-B.

4.2.4 Aplane, enderece y haga coincidir los hilos empleando las pinzas de punta, luego recórtelos en línea recta alrededor de 3.00 cm. del borde de la envoltura. (Ver figura No.7).

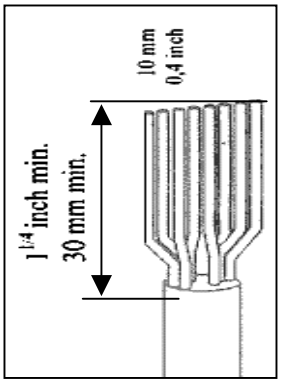


Figura No.7. Aplane, enderece y haga coincidir los hilos.

4.2.5 Inserte los hilos del cable en cada uno de los canales del jack RJ45 como se muestra a continuación y siguiendo la configuración T568-B. (Ver figura 8).

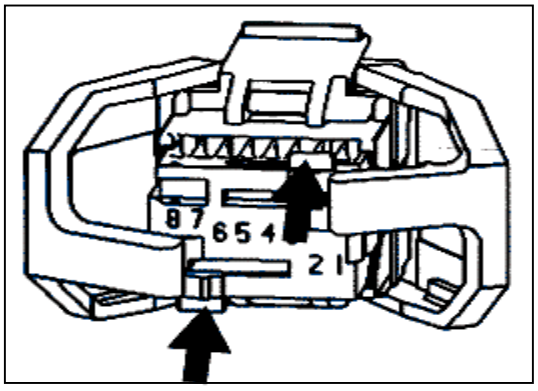


Figura No.8. Inserte los hilos del cable en cada uno de los canales del jack RJ45.

4.2.6 Corte al ras del jack RJ45 el excedente de cable con las pinzas de corte. (Ver figura No.9).

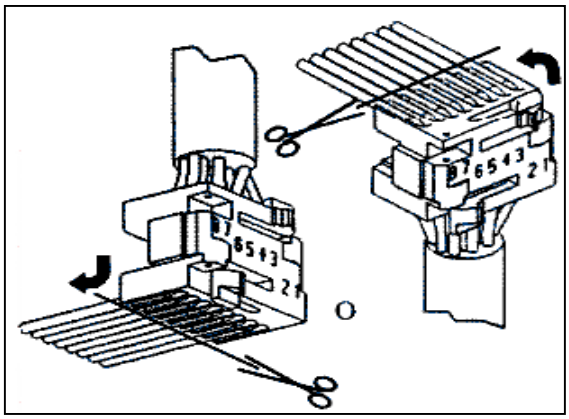
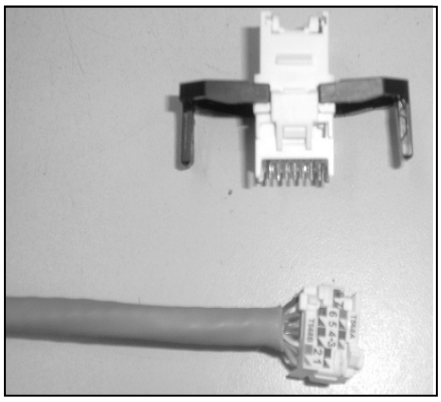
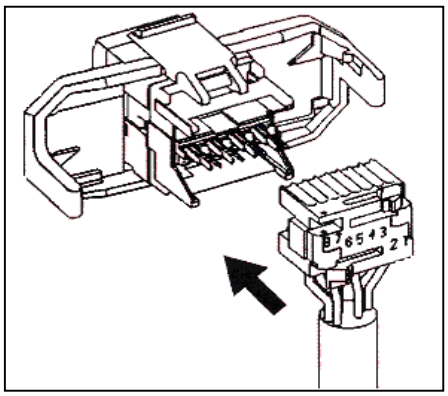


Figura No.9. Corte al ras del jack el excedente del cable.

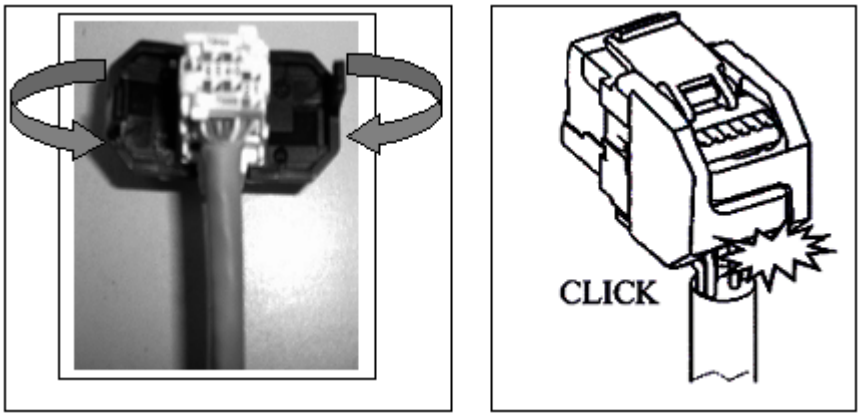
4.2.7 Inserte la tapa de terminación del jack en el outlet con firmeza y hasta el tope, teniendo cuidado de la alineación mencionada anteriormente. (Ver figuras No.10 y 11).





Figuras No. 10 y 11. Inserte la tapa de terminación del jack en el outlet con firmeza y hasta el tope.

4.2.8 Cierre y presione firmemente la tapa del jack RJ45 hasta escuchar un clic. (Ver figuras No.11 y 12).



Figuras No. 11 y 12. Cierre y presione firmemente la tapa del jack RJ45 hasta escuchar un clic.

4.2.9 Repita el procedimiento con el otro extremo. Finalizando así la instalación del jack RJ-45.

5.- Pruebas y Aplicaciones

5.1 Haga la siguiente conexión para cada una de las máquinas disponibles (Ver figura No.13):

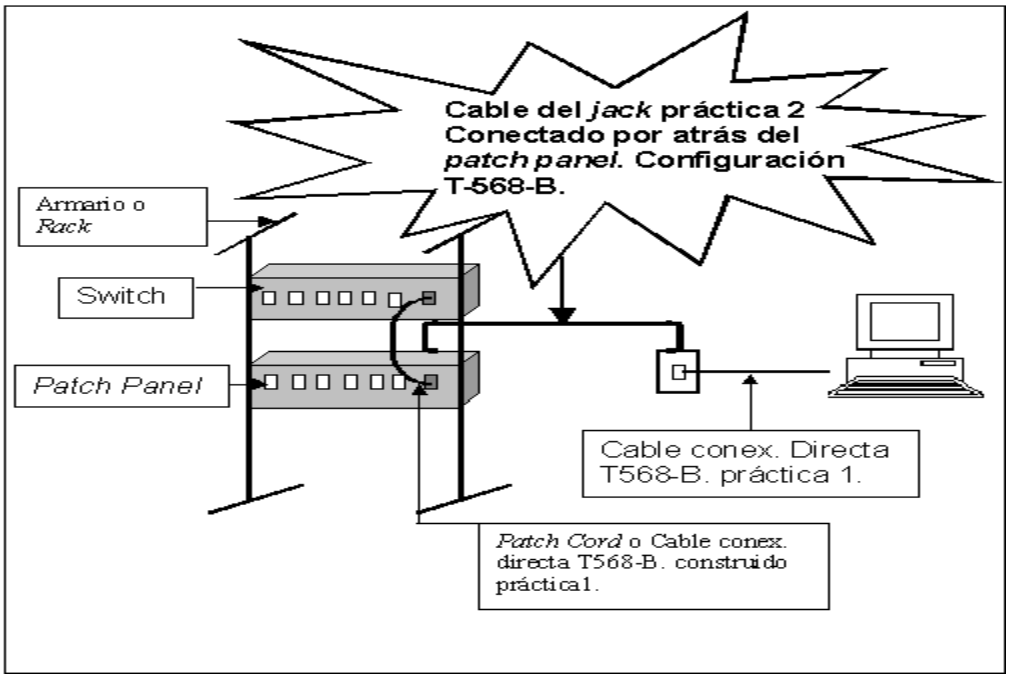


Figura No.13. Conexión de cables a dispositivos de la red.

5.2 Encienda la computadora y elija cargar Windows.

5.3 Verifique que en ambos extremos del cable horizontal este encendido el led, es decir, que el led de la trajeta de red de su computadora este encendido y que el led del puerto del switch al cual fue conectado el cable este encendido.

5.4 Verificar que las computadoras se puedan ver en el entorno de red utilizando el comando ping –t y conectándose al servidor 132.248.59.254 (Gateway de la DIE.) (Ver figura No.14).

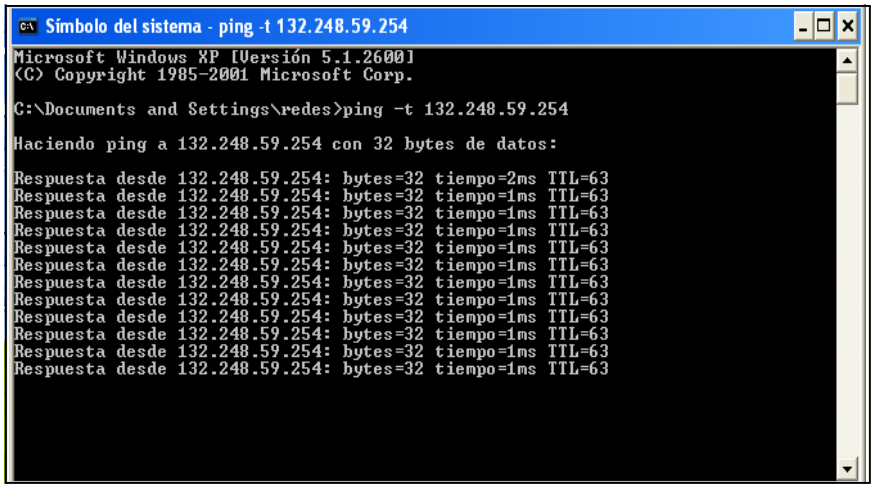


Figura No. 14. Entorno de red utilizando el comando ping –t.

5.5 Apague la computadora.

6.- Cuestionario

1. ¿Cuál es el estándar que regula a nivel internacional el sistema de cableado estructurado?

2. ¿A qué se refiere el concepto “cableado estructurado”?

3. ¿Cuál es la distancia máxima que puede tener el cableado horizontal?

7.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica

PRÁCTICA 2
Componentes del Cableado Estructurado
Norma ANSI/EIA/TIA T568-B
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. Mencione las partes que conforman el cableado estructurado. Explíque los subsistemas.

2. ¿Qué es el panel de parcheo?

3. Además del *jack* ó *keystone* RJ45, ¿qué otros tipos de conectores hembras existen?

4. ¿Cuántos tipos de canaletas existen?

5. ¿Cuál es la diferencia entre la ductería y las canaletas?

PRÁCTICA 3
Instalación de una Red Básica en Plataforma Windows XP de Microsoft

1.- Objetivos de Aprendizaje
El alumno:

- Al finalizar la práctica tendrá la capacidad de instalar y configurar una tarjeta de red.
- Será capaz de instalar una LAN básica conectando dos computadoras utilizando un cable de conexión cruzada (*crossover*).

2.- Conceptos teóricos

Una tarjeta de Red (NIC – *Network Interface Card*) es el dispositivo que conecta a una estación, PC u otro equipo de red con el medio físico. El tipo de conector de la tarjeta de red dependerá de las características del medio de comunicación de la red: (par trenzado, coaxial, fibra óptica, aire) al cual se conecte. (Ver figuras No.1 y No.2).

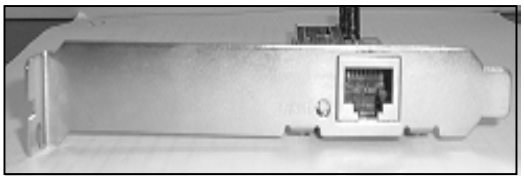


Figura No.1. Tarjeta de Red con conector RJ-45 para cable UTP.

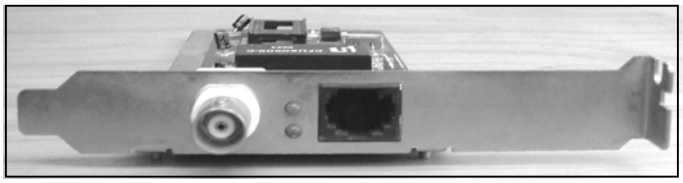


Figura No.2. Tarjeta de Red con conectores BNC para cable coaxial y RJ-45 para UTP.

Se define en la capa 2 del modelo OSI debido a que tiene y reconoce direcciones MAC (Subnivel de la capa de Enlace). Contienen un código único en todo el mundo, que se llama dirección de Control

de Acceso al Medio (MAC, *Media Access Control*). Esta dirección se utiliza para controlar la comunicación de datos para el host en la red.

La NIC es el componente de hardware básico en las comunicaciones de red. Traduce la señal producida por la computadora en un formato serie que se envía mediante el cable de red. La comunicación binaria (unos y ceros) se transforma en impulsos eléctricos, pulsos de luz, ondas de radio o cualquier esquema de transmisión de señales que usen los medios de comunicación en red, de manera que convierte el intercambio de señales a través de los medios de transmisión en una comunicación de datos efectiva.

Las funciones de la tarjeta de red son:

- Preparar los datos del equipo (formar tramas) para pasarlos a la capa física.
- En la salida transferir las tramas al medio físico de transmisión según el protocolo de comunicación.
- Recibir los datos que llegan por el cable y convertirlos en bytes para que puedan ser comprendidos por la unidad de procesamiento central del equipo (CPU).
- Controlar el flujo de datos entre el equipo y el sistema de cableado.

Tipos de Buses de una tarjeta de red:

Conectores ISA y PCI

Hay dos tipos comunes de conectores de NIC para PC:

- *Los zócalos ISA (Industry Standard Architecture. Arquitectura Estándar de la Industria).* Son un tipo de slots o ranuras de expansión de 16 bits de longitud de palabra de datos para los ordenadores personales.
- *Los zócalos PCI (Peripheral Component Interconnect Interconexión de Componentes Periféricos).* Son un tipo de slots o ranuras de expansión de 32 bits de longitud de palabra de datos para los ordenadores personales. Los zócalos PCI

tienen un mayor rendimiento que los ISA. Los zócalos PCI miden unos 9cm. de longitud.

NIC especializadas

En algunos casos, es posible que necesite utilizar *NIC especializadas.* Por ejemplo, si el ordenador es un portátil, se necesitará utilizar una *tarjeta PCMCIA (Personal Computer Memory Card International Association)*. Asociación de la industria de fabricantes de hardware para ordenadores o computadoras portátiles encargada de la elaboración de estándares) y pueden ser de muy distintos tipos: memoria, disco duro, tarjeta de red, etc.
Las tarjetas *PCMCIA* de 16 bits pueden recibir el nombre de *PC Card* y las de 32 bits el de *CARD BUS*.
Una *tarjeta PCMCIA* es un dispositivo normalmente utilizado en computadoras portátiles para expandir las capacidades de este. (Ver figura No.3).



Figura No.3.

Cuando elija una tarjeta PCMCIA, deberá considerar lo siguiente:

- La velocidad de su concentrador, conmutador o servidor de impresora - Ethernet (10Mbps) o Fast Ethernet (100Mbps).
- El tipo de conexión que necesita - RJ-45 para par trenzado o BNC para cable coaxial.
- Si tiene un puerto USB, podría considerar utilizar un Interfaz de red USB (*USB Network Interface*). (Ver figura No.4).

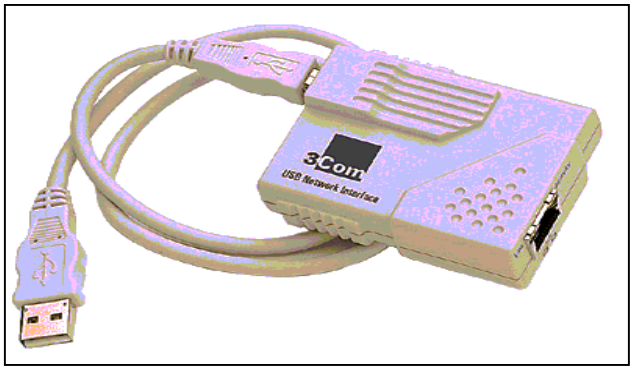


Figura No.4.

En la actualidad existen tarjetas de red UTP 10,100 y 1000 Mbps e inclusive de 10 Gbps.

Ejemplo de Marcas comerciales de Tarjetas de Red son:

- 3COM
- ROBOTIC'S
- INTEL
- DELL
- ALTEC, etc.

3.- Equipo y material necesario

Material que debe traer el alumno:

A. Desarmador de cruz.
B. Desarmador plano.
C. Cable cruzado (*crossover*) construido en la práctica 1.

Equipo del Laboratorio:

D. 2 Computadoras con Windows 98/NT/2000P/XP.
E. Ranura de expansión PCI o ISA vacía.
F. Tarjeta de red PCI o ISA.
G. Controlador de la tarjeta de red.

4.- Desarrollo

4.1 Modo de trabajar

La instalación de la LAN básica se realizará en equipos de dos personas por cada computadora del laboratorio.

El equipo donde se desarrolle la práctica deberá contar con privilegios de administrador.

4.2 Instalación de la Tarjeta de Red

4.2.1 Apague la computadora y desconecte el cable de alimentación.

4.2.2 Retire la placa contra el polvo del gabinete e identifique la ranura de expansión PCI o ISA vacía en la que desea instalar la tarjeta NIC (Ver figura No.5). Actualmente todas las tarjetas de red son del tipo PCI (A partir del procesador Pentium 2 ó 3). (Ver figura No. 6).

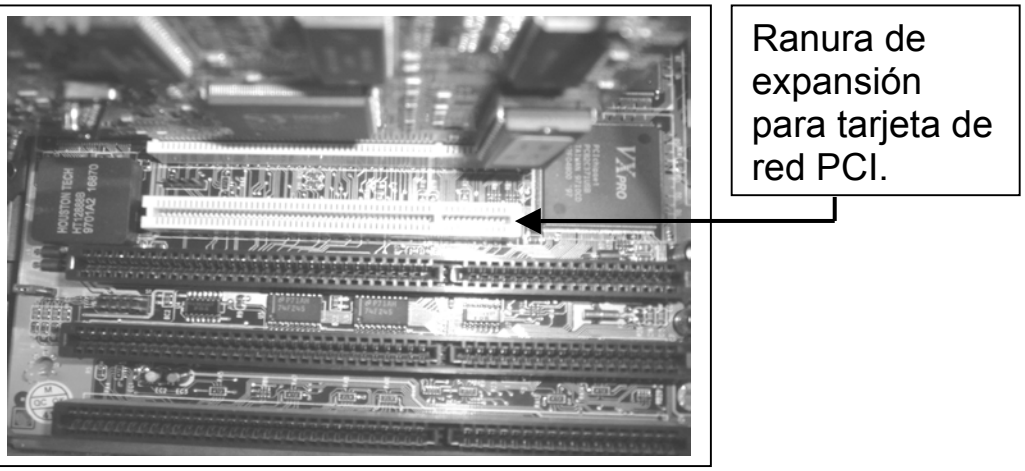


Figura No.5. Identificación de ranura de expansión para Tarjeta de Red PCI.

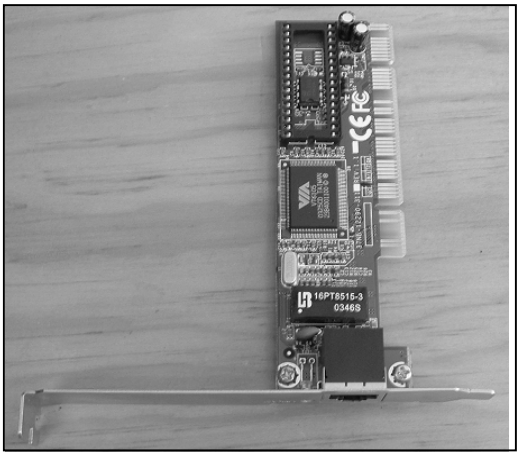


Figura No. 6 Tarjeta de Red PCI.

4.2.3 Sujete los bordes superiores de la tarjeta de red con ambas manos y coloque la tarjeta en la ranura de expansión. (Ver figura No.7).

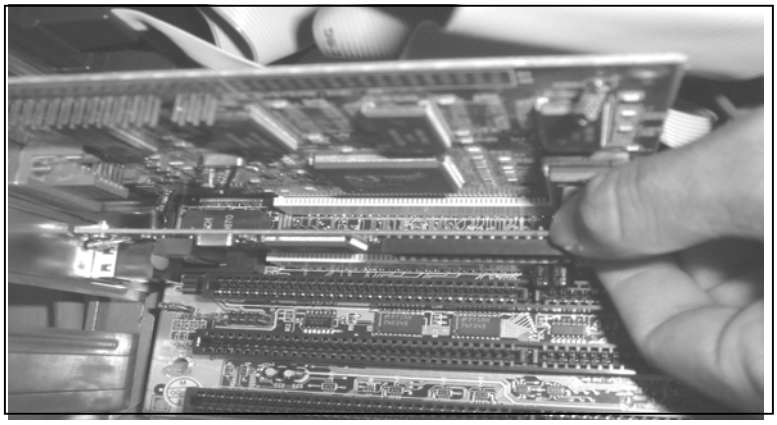
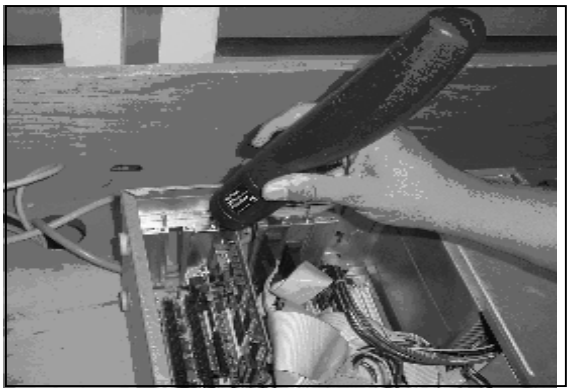


Figura No.7. Colocación de la Tarjeta de Red en la ranura de expansión.

4.2.4 Por último, use un tornillo para fijar la tarjeta en el gabinete. (Ver figuras No.8 y 9).

Figuras No.8 y 9. Fijación de Tarjeta de Red en el gabinete utilizando un tornillo y el desarmador.

4.2.5 Conecte el cable de alimentación, encienda la computadora y elija cargar Windows XP. El detector de hardware de Windows XP determina automáticamente el controlador del adaptador para la tarjeta de red y lo llevará paso a paso en la instalación de éste.
Es posible que Windows le solicite que proporcione el nombre de la computadora y del grupo de trabajo.

Indique el nombre de la computadora: nombre_computadora
Grupo: nombre_grupo

Nota.- Cabe mencionar que después de instalar y configurar la tarjeta de red, se conectarán dos computadoras para probar la correcta construcción del cable crossover. (Construido en la Práctica No.1). (Ver figura No.10).

Se necesitará conocer los nombres e IP asociadas de las computadoras involucradas (nombre_computadora1 y su IP asociada, nombre_computadora2 y su IP asociada), así como el grupo al que pertenecen éstas deberá ser el mismo.

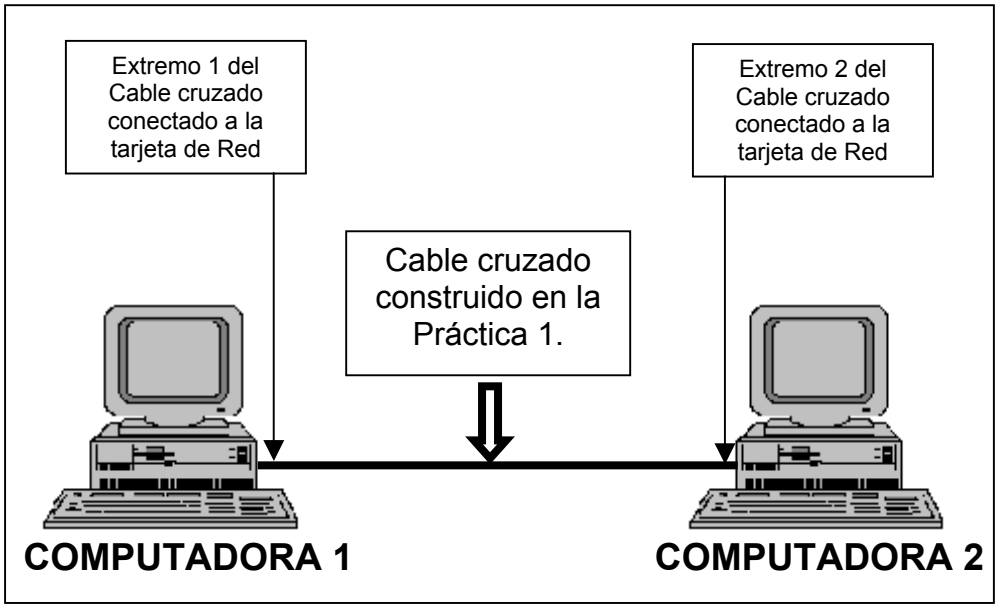


Figura No.10.

4.3 Para instalar el controlador de la tarjeta de red en el sistema operativo Windows XP Professional.

Realizar lo siguiente:

4.3.1 Al encender la computadora aparecerá *el asistente para hardware nuevo encontrado* (ver figura No.11), quien lo ayudará a instalar el software para el Controlador Ethernet y le dará dos opciones a elegir:

- 1. Instalar automáticamente.
- 2. Instalar desde una lista o ubicación específica.

4.3.2 Inserte el disco del controlador de la tarjeta en el floppy, elija la segunda opción y de clic en Siguiente.

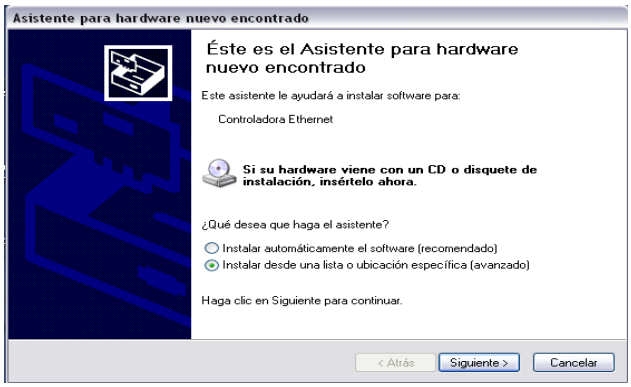


Figura No.11.

4.3.3 En seguida aparecerá la siguiente pantalla (figura No. 12) En donde se indicará que busque el controlador de la tarjeta de red en el disquete (caso de la práctica) o en el CD-ROM.

De clic en Siguiente.

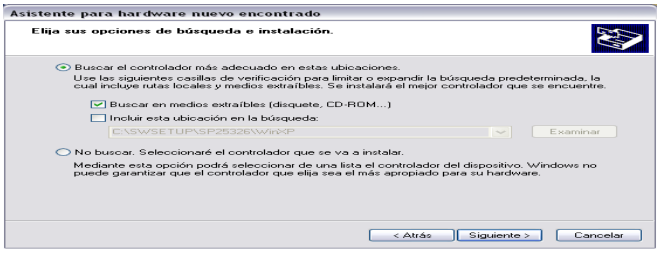


Figura No.12.

4.3.4 Al finalizar la instalación del controlador de la tarjeta de red aparecerá la siguiente pantalla (Ver figura 13).En donde se indicará el modelo de tarjeta instalada.
Haga clic en Finalizar.

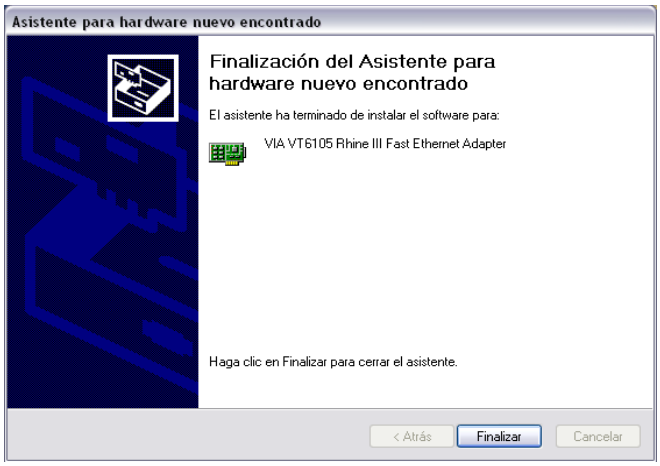


Figura No. 13.

4.3.5 Para cerciorarse que la tarjeta funciona correctamente vaya al Administrador de Dispositivos. Para ello realice lo siguiente:

a. Haga clic en el botón Inicio, seleccione Panel de control. y luego seleccione Sistema (Ver figuras No.14 y 15).



Figura No.14.

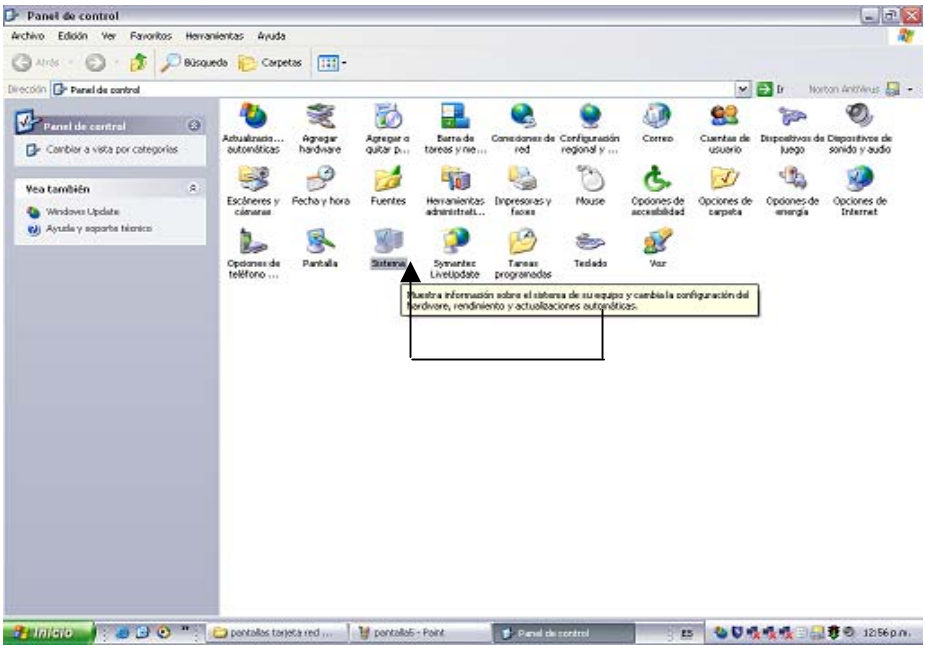


Figura No.15.

b. Haga doble clic en el icono Sistema. Aparece un cuadro de diálogo de Propiedades del Sistema. (Ver figura No.16).

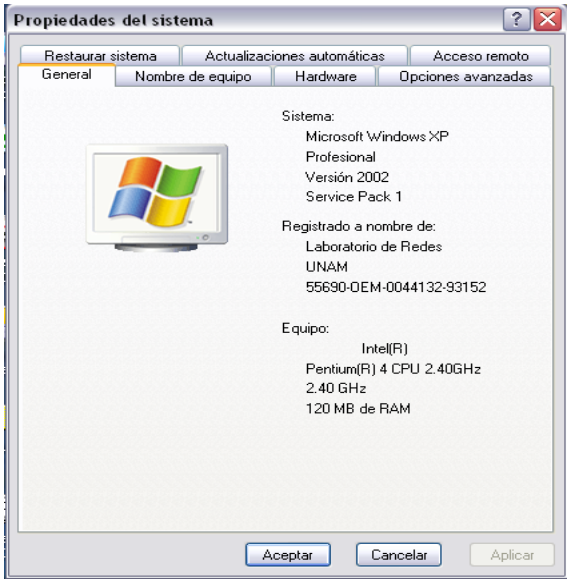


Figura No.16.

c. Haga clic en la pestaña Hardware y seleccione la opción Administrador de Dispositivos. Aparece un cuadro de

diálogo del Administrador de Dispositivos. Dé doble click (Ver figura No.17).



Figura No.17.

d. Haga clic en la opción de Adaptadores de Red. Si no aparece ningún signo de interrogación la Tarjeta de Red no presenta conflicto con otro dispositivo. (Ver figuras No.18 y 19).

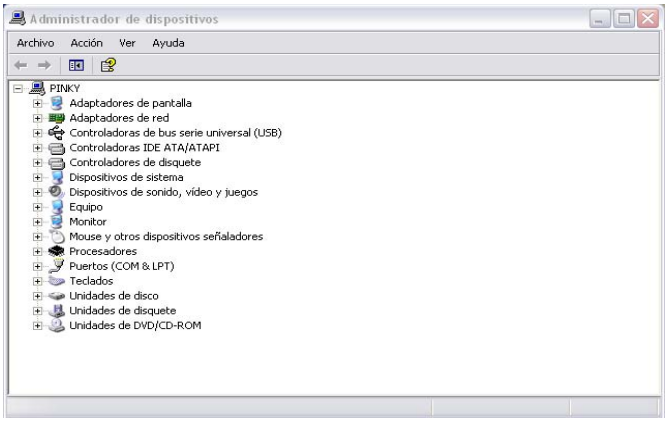


Figura No.18.

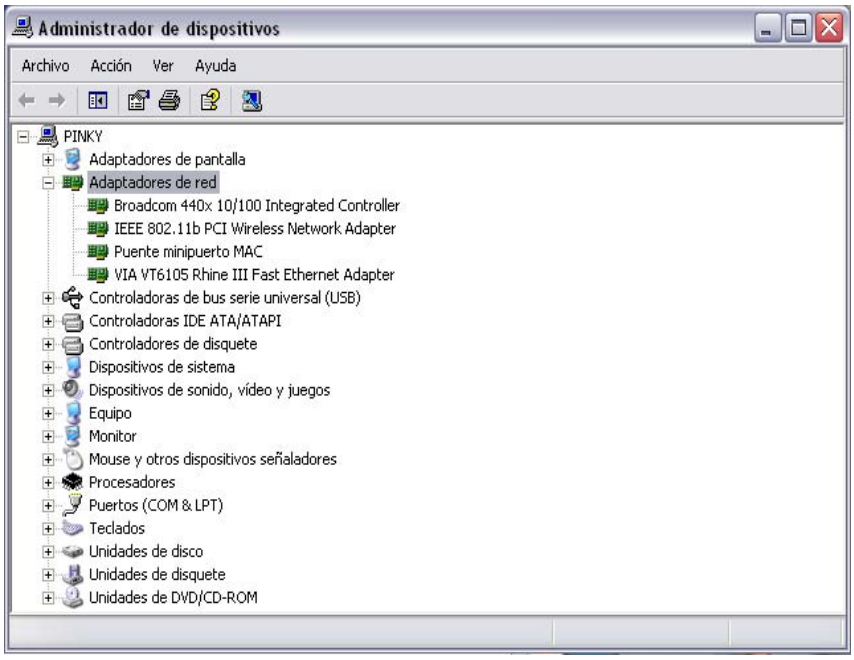


Figura No.19.

4.4 Configuración de la Tarjeta de Red

Los componentes de red son los clientes, servicios y protocolos que se utilizan para comunicar con los servidores de la red después de conectar con un servidor.

Es importante señalar que existen cuatro tipos de componentes representados cada uno por un icono distinto:



4.4.1 Para ver los componentes instalados en su computadora realice lo siguiente:

a) Haga clic en el botón Inicio, seleccione Panel de control y luego seleccione Conexiones de red. (Ver figura No.20).

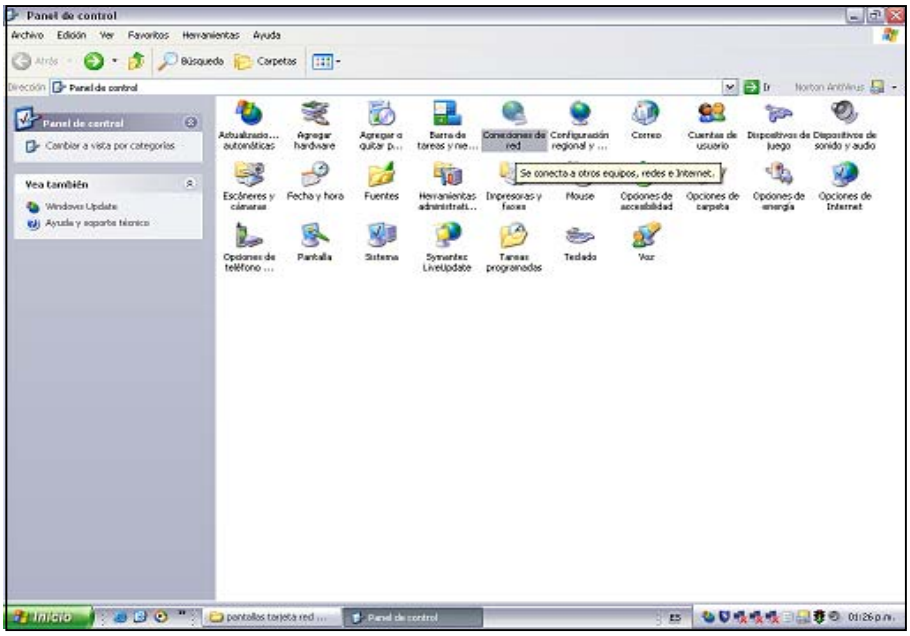


Figura No.20.

b) Dé doble clic en Conexiones de red. (Ver figura No.21).

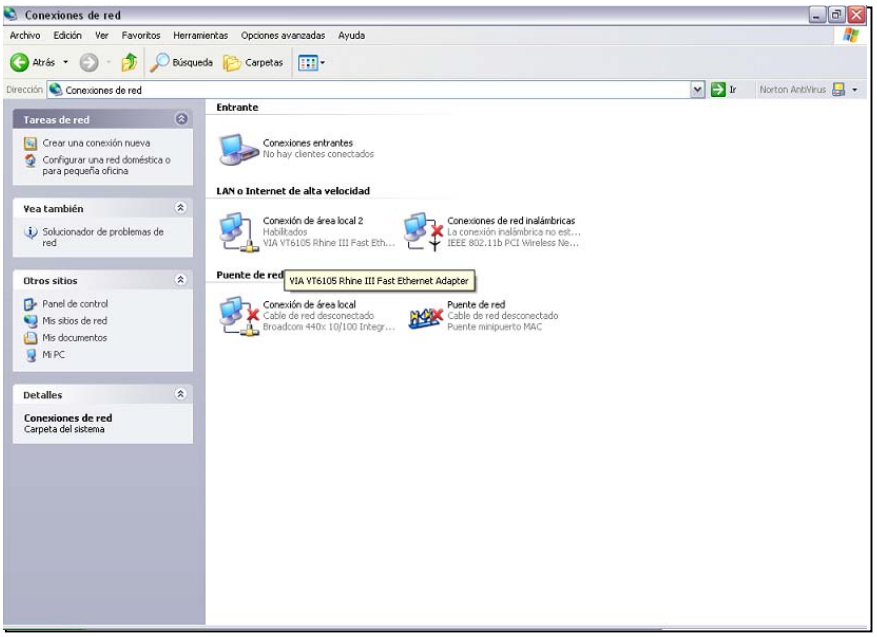


Figura No.21.

c) Haga un clic con el botón derecho del Mouse sobre el ícono en conexión de área local 2 (VIA VT6105 Rhine III Fast Ethernet Adapter ó el que aplique) adaptador de red en el caso de la práctica y seleccione la opción Propiedades. (Ver figura 22).

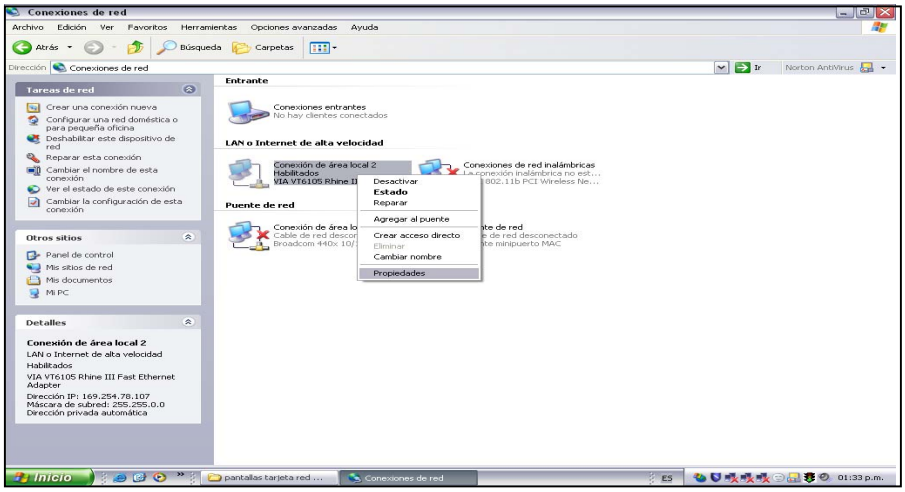


Figura No.22

d) Seleccione la carpeta general. Estos elementos aparecen por default. (Ver figura No.23).



Figura No.23

e) En el recuadro que dice:

Esta conexión utiliza los siguientes elementos, deben estar presentes los siguientes componentes de la red:

- Clientes para redes Microsoft (Cliente).
- Compartir impresoras y archivos para redes Microsoft (Servicio).

- Programador de paquetes Q o S.
- Protocolo de Internet (TCP/IP).
- IPX/SPX (Protocolo).
- Acceso telefónico a redes (adaptador). Opcional para está práctica.
- Algún adaptador de red que varía según la marca de la tarjeta de red (adaptador).

En caso de faltar alguno de estos componentes haga clic en el tipo de componente de red que desee instalar. Enseguida de Agregar.

Por ejemplo si desea agregar algún componente para cliente. (Ver figura No.24).

Un Cliente proporciona acceso a equipos y archivos en la red a la que se está conectando.

1. Cliente Seleccione fabricante Microsoft
   Cliente para red Microsoft

Hacer click en Agregar. Reinicie la computadora. (Ver figura No.24.a).

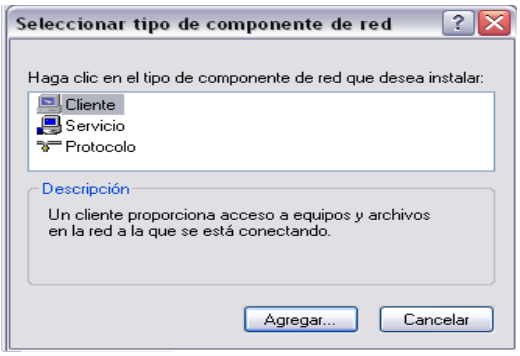


Figura No.24.

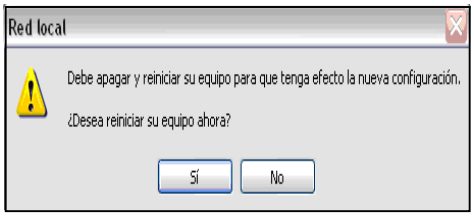


Figura No.24.a.

Si desea agregar algún componente para Servicios (Ver figuras No.25 y 26).

Los Servicios proporcionan características adicionales como el poder compartir archivos e impresoras.

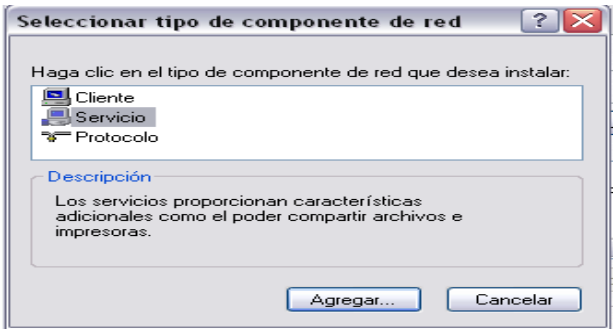


Figura No.25.



Figura No.26.

Si desea agregar algún componente para Protocolo (Ver figuras No.27 y 28).

2. Protocolo
   Seleccione fabricante Microsoft
   Seleccione el protocolo NETBEUI o IPX/SPX
   Hacer click en Agregar. Reinicie la computadora.

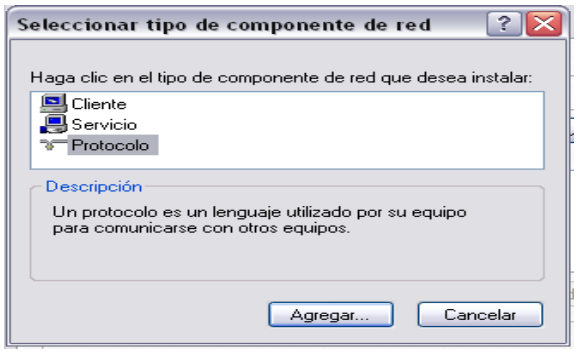


Figura No.27.

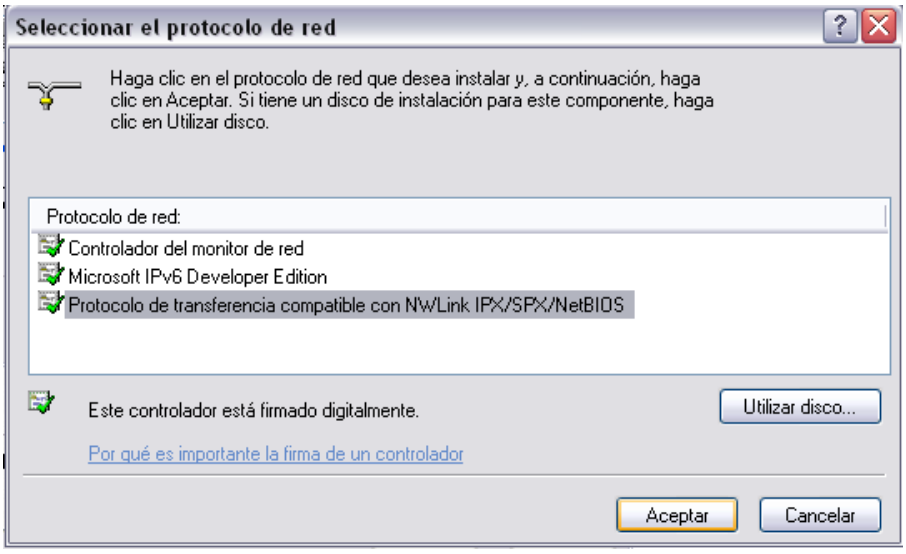


Figura No.28.

Aquí se verá el Protocolo que se instaló Nwlink IPX/SPX/ (Ver figura No.29).

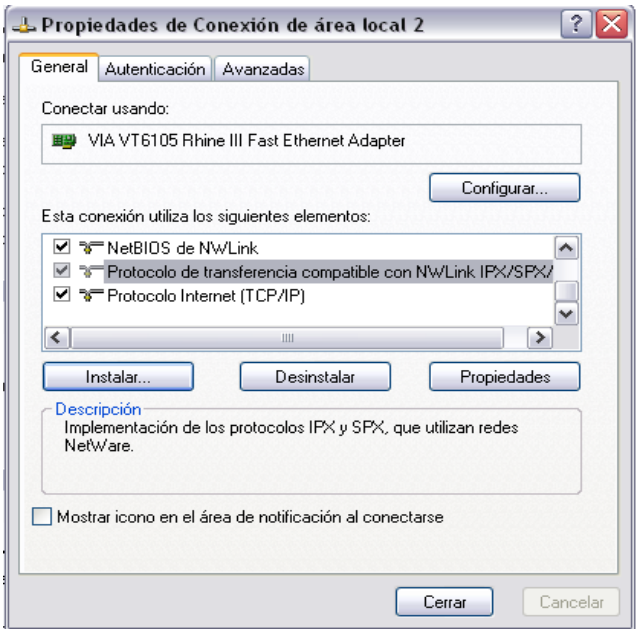


Figura No.29.



Figura No.30

NOTA:-Se puede hacer un IPCONFIG desde:
Inicio →símbolo de sistema ó
Inicio →Todos los programas→ Accesorios.
Para saber la dirección IP de nuestro equipo, ya que para configurar el Protocolo TCP/IP será necesario). (Ver figura No.31).

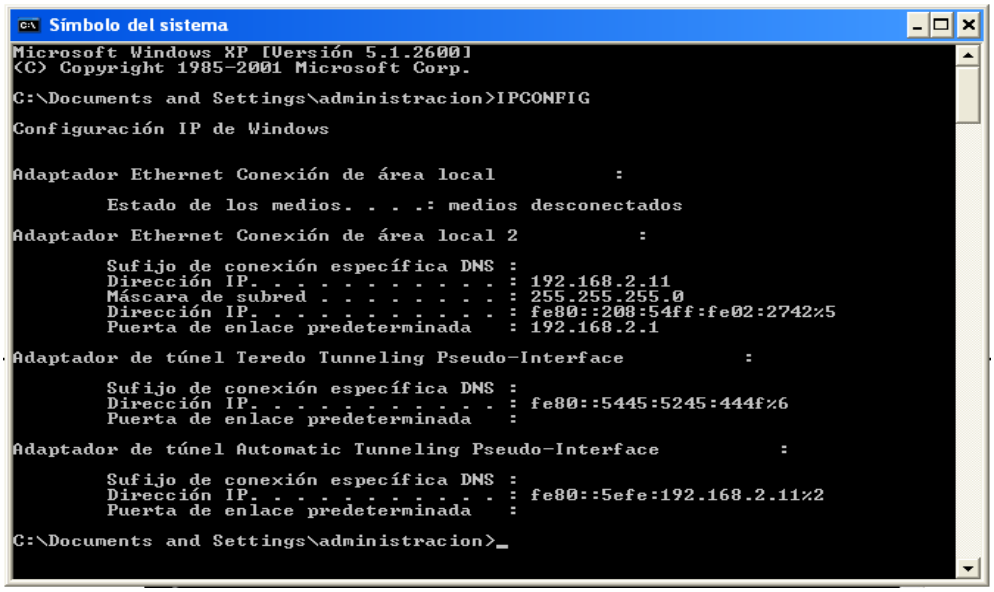


Figura No.31.

El Protocolo TCP/IP. Es un Protocolo de red de área extensa predeterminado que permite la comunicación entre varias redes conectadas entre sí. Es necesario configurarlo. Para ello de un clic sobre el protocolo. (Ver figura No.32).

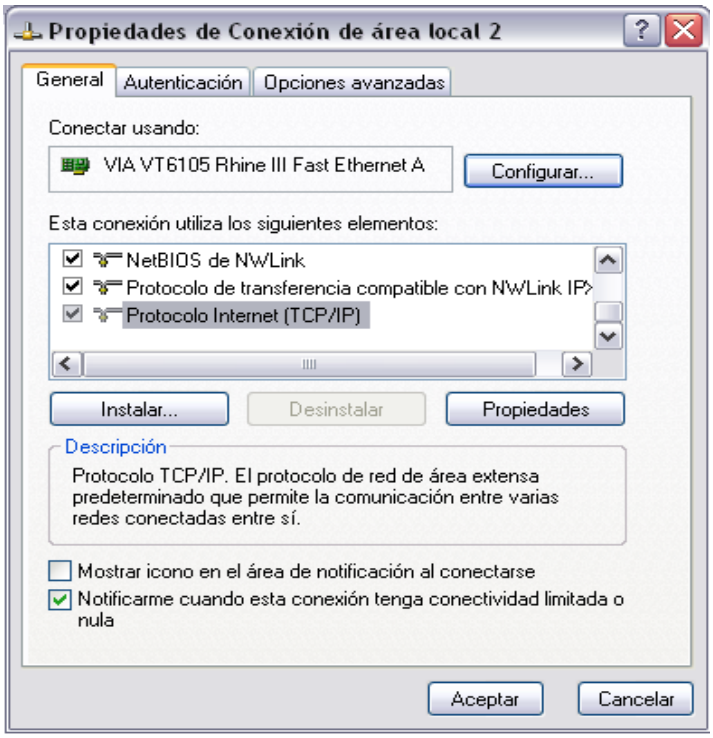


Figura No.32

En seguida de clic en Propiedades.
Aparecerá la pestaña general.
Seleccione el botón donde indica:

Usar la siguiente dirección IP.
Como se muestra en la figura No.33

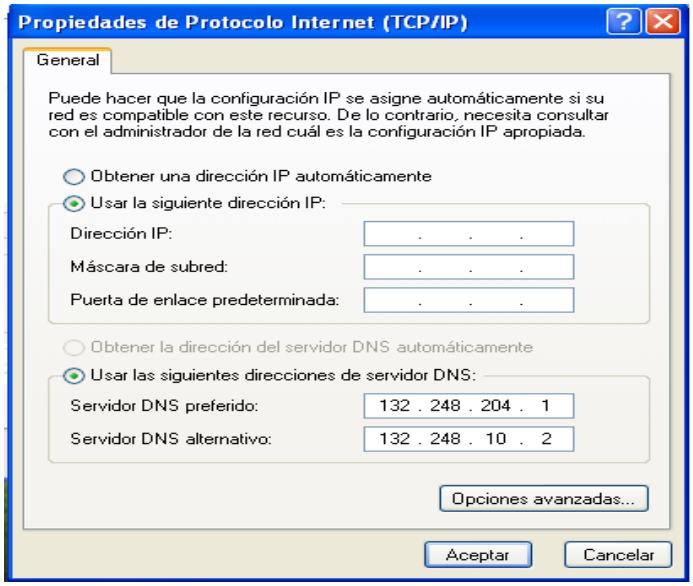


Figura No.33

Para configurar la Dirección IP es necesario consultar con el profesor del laboratorio para que nos proporcione la siguiente información:

La Dirección IP.
La Máscara de subred.
La puerta de enlace predeterminada.

Se debe asignar también la dirección IP de los siguientes servidores (la dirección IP de los servidores que se muestran en la pantalla son los de la DIE y DGSCA):

Servidor DNS preferido
Servidor DNS alternativo

Ya que la dirección IP de cada máquina es única y por lo tanto depende del equipo que se esté utilizando. (Ver Figuras No.34 y No.35). Donde **xx**, es la dirección IP asignada por el profesor del laboratorio. En seguida de un clic en Aceptar.

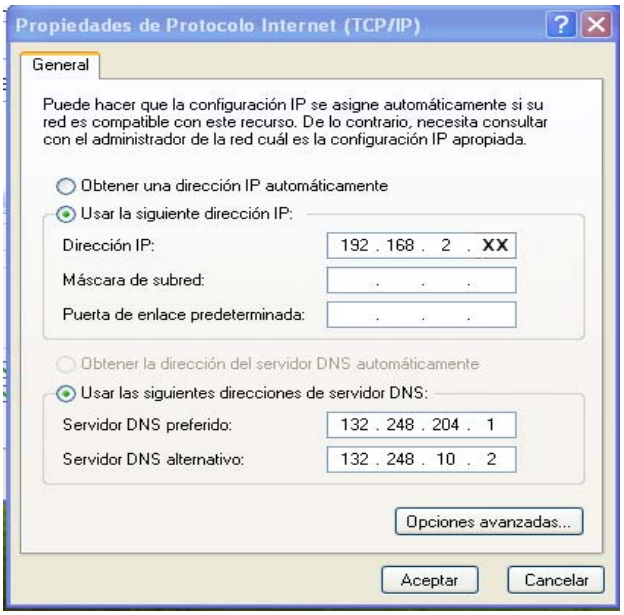


Figura No.34

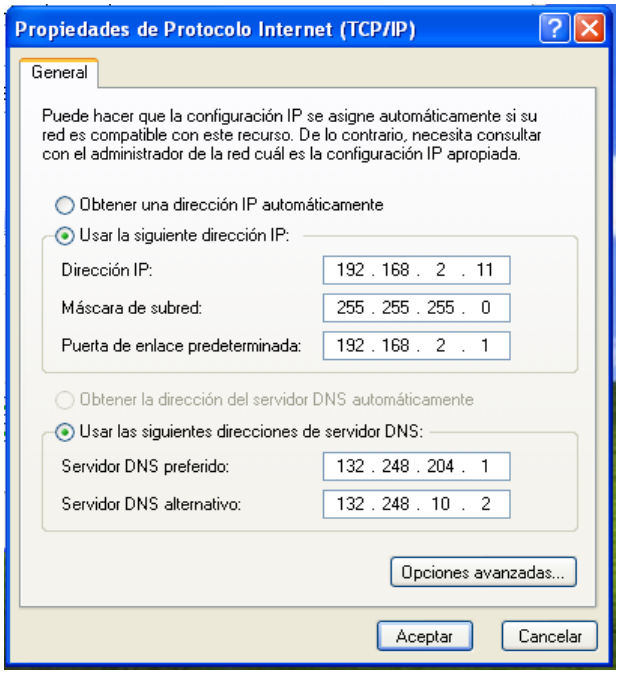


Figura No.35

De nuevo revise su conexión de red, en donde encontrará en la parte inferior izquierda que se configuró manualmente la Tarjeta de red (Ver figura No.36). Reinicie su equipo.

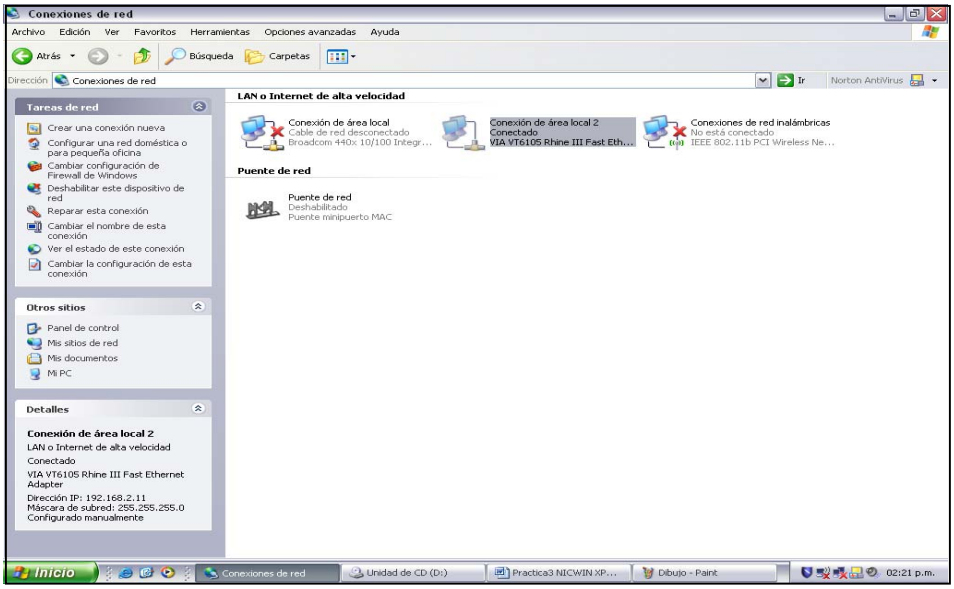


Figura No.36

Otro Componente que ofrece control del tráfico de la red, incluídos servicios de índice de flujo y prioritarización, es el programador de paquetes de calidad de servicio (Q o S). (Ver figura No.37).



Figura No.37.

3. Compartir Archivos o Impresoras

- Haga click en el botón Compartir impresoras y archivos. (Ver figura No.38).

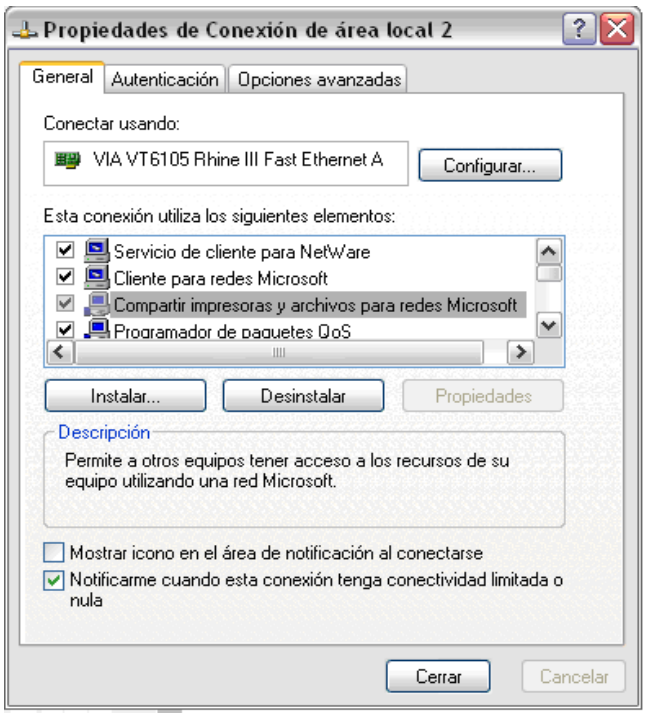


Figura No.38.

- Seleccionar ambas casillas que aparecen:

  - Permitir que otros usuarios tengan acceso a mis archivos.
  - Permitir que otros usuarios impriman en mis impresoras.

5.- Pruebas y Aplicaciones

5.1. Comprobar que la LAN básica funcione correctamente.

Aunque haya activado la compartición de recursos en su equipo, sus recursos no estarán disponibles en la red hasta que no haya indicado qué recursos desea compartir. Para compartir una unidad de disco o una carpeta:

5.1.1 Abra el explorador de Windows.

5.1.2 Haga click derecho sobre la unidad de disco o carpeta y seleccione compartir en el menú que se despliega. Esto despliega una ventana de Propiedades de la unidad o carpeta seleccionada. (Ver figura No.39).

NOTA.- Para hacer la carpeta accesible su nombre deberá contener 12 caracteres o menos.

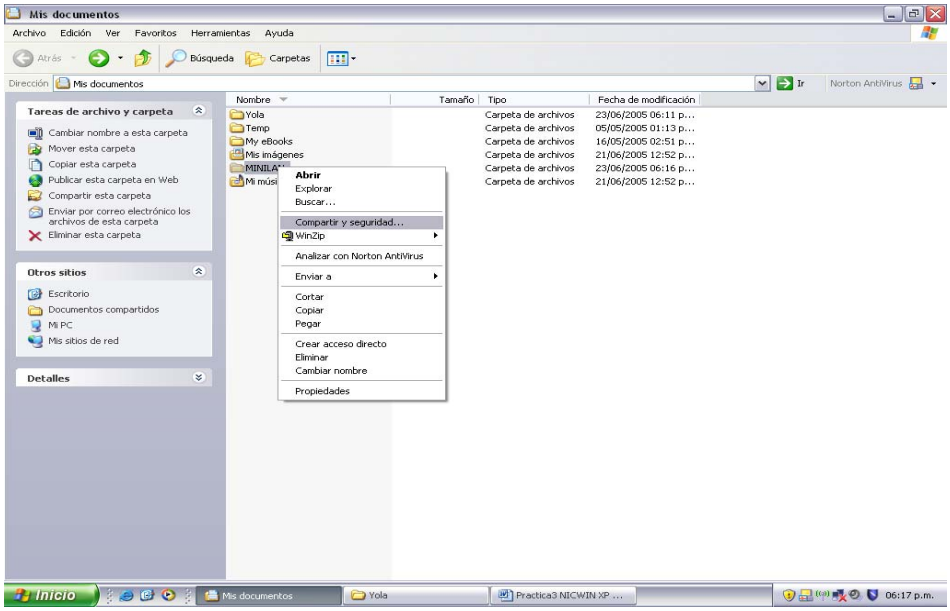


Figura No.39.

- Seleccione la pestaña Compartir. (Ver figura No.40).



Figura No.40.

- Habilitar la opción Compartir esta carpeta en la red. Y en seguida aparecerá en la Casilla del Recurso compartido el nombre de la carpeta o archivo que elegimos compartir. Dar clic en Aceptar (Ver figura No.41).

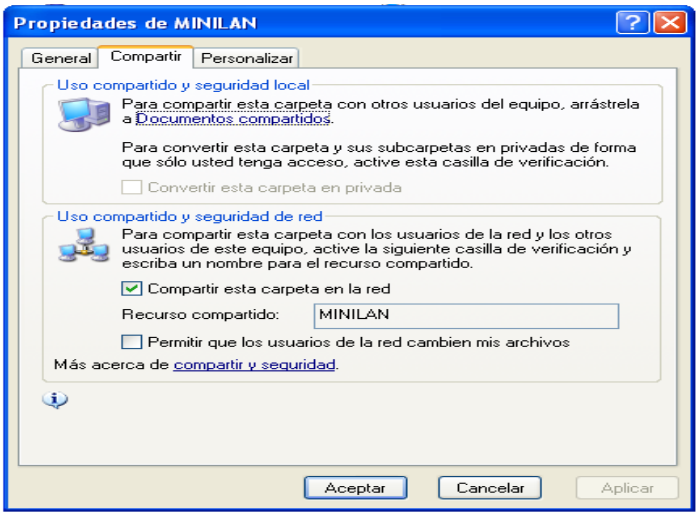


Figura No.41.

- Verifique en Mis documentos que la carpeta MINILAN aparece con un ícono diferente. (Ver figura No.42).

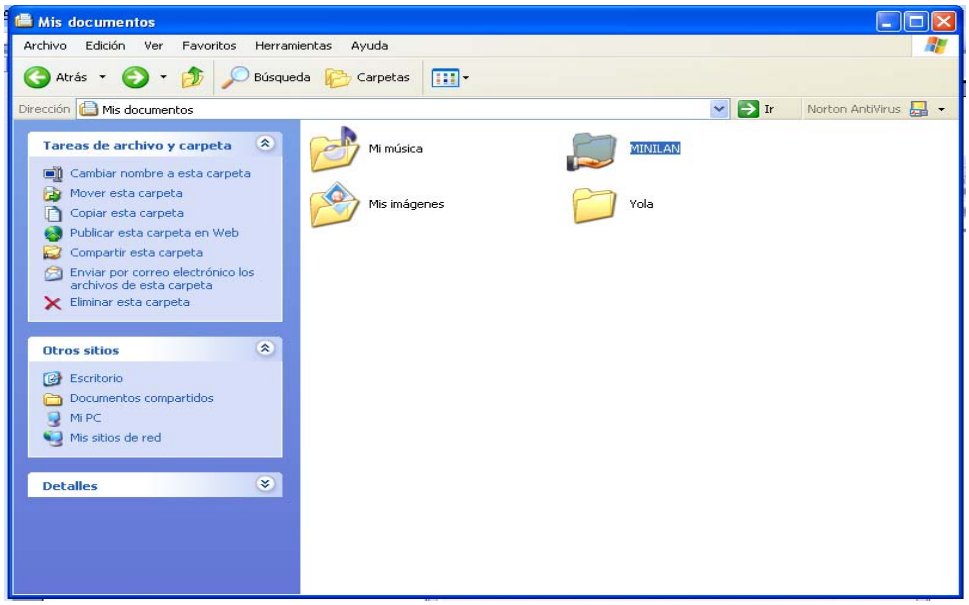


Figura No.42.

5.1.3 Otra forma de compartir una carpeta es la siguiente: Elija la carpeta a compartir y selecciónela, del lado izquierdo encontrará dentro de Tareas de archivo y carpeta, la opción Mover esta carpeta. Dar un clic (Ver figura No.43).

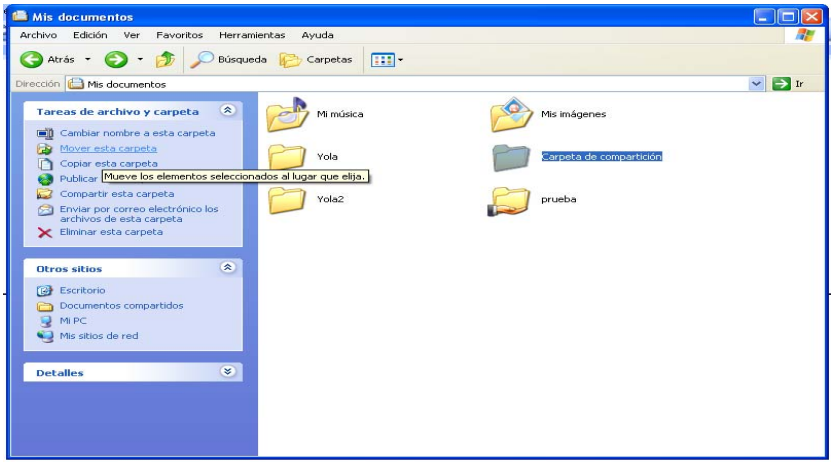


Figura No.43.

Aparecerá la siguiente pantalla, en donde tendrá que seleccionar el sitio donde desea mover la carpeta que desea Compartir “nombre de la carpeta”. (Ver figura No.44).

Seleccione la Carpeta de Documentos compartidos.

A continuación haga clik en el botón Mover.

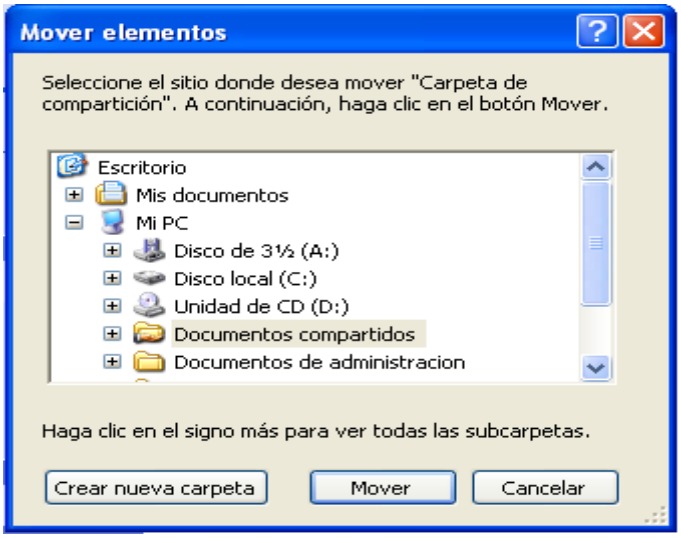


Figura No.44

En seguida se moverá la carpeta que eligió compartir del lugar en donde estaba. (Ver figura No. 45).

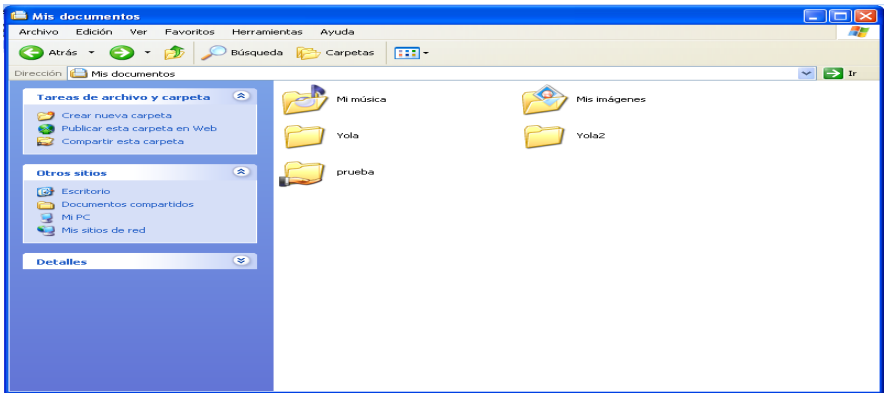


Figura No.45.

De un clic en la carpeta Documentos compartidos. Aquí se encontrará su carpeta que desea compartir. En nuestro ejemplo se llama carpeta de compartición. (Ver figura No.46).

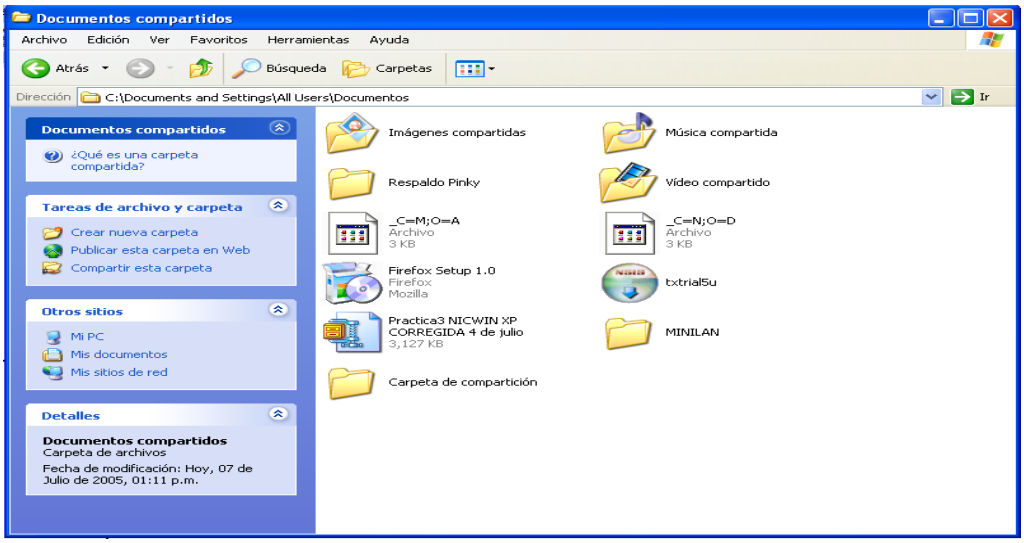


Figura No.46.

Verifique si desde su computadora puede tener acceso a la información compartida por la otra computadora. Para esto realizar lo siguiente:

Ir a Mis sitios de red, en donde encontrará todos los documentos compartidos por las computadoras que se encuentran en la Red local. (Ver figura No.47).

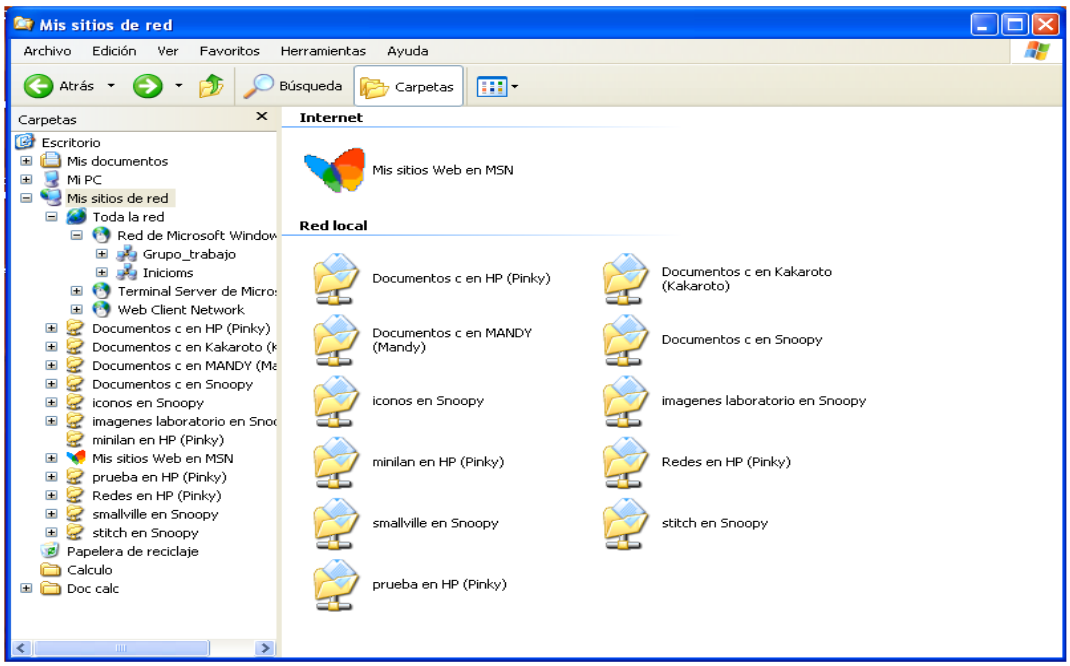


Figura No.47.

Seleccione la Carpeta de Documentos ("compartidos") de la Máquina que desea ver. (Figuras No.48 y 49).

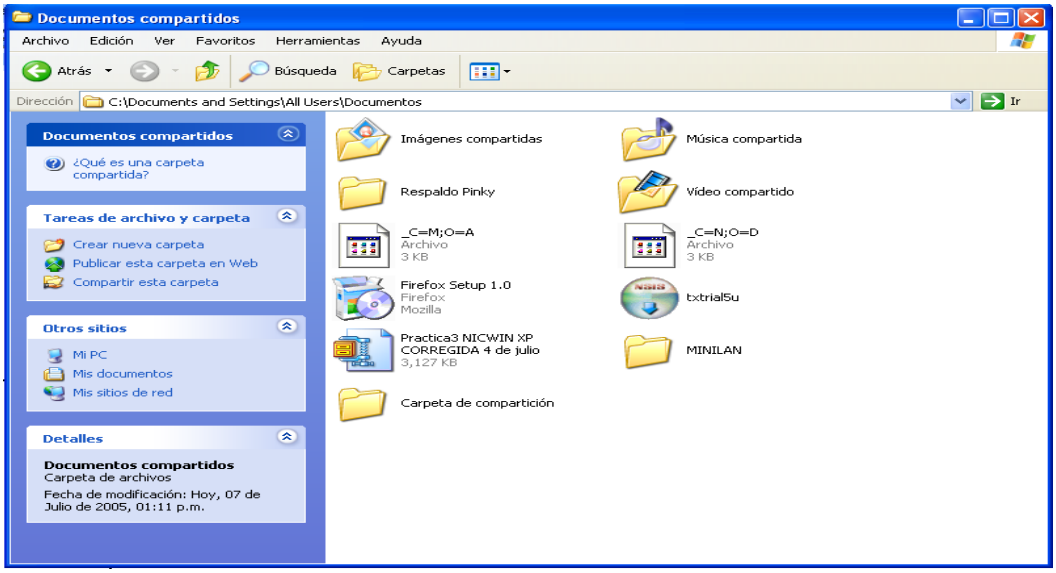


Figura No.48.



Figura No.49

i. Haga click sobre el icono de entorno de red.
ii. En la ventana del Entorno de red, seleccione Actualizar en el menú Ver.
iii. Seleccione Ver toda la red o presione F5.
iv. Verifique que puede "ver" a la otra máquina del laboratorio.
v. Haga doble click sobre ella y vea que recursos tiene compartidos.

5.2 *Conecte el cable cruzado (crossover) a las dos computadoras, creando así una LAN básica.* (Ver figura No.10).

5.2.1 Para comprobar el correcto funcionamiento de la red a través del cable crossover haga lo siguiente:

a. Ejecute el comando *ping -t* con su dirección IP como parámetro dentro del command prompt de Windows (Símbolo del sistema), una respuesta afirmativa le asegurará que su dispositivo está correctamente configurado, por lo que fallas de comunicación pueden ser atribuidas a otras partes de la red. (Ver figura No.50).

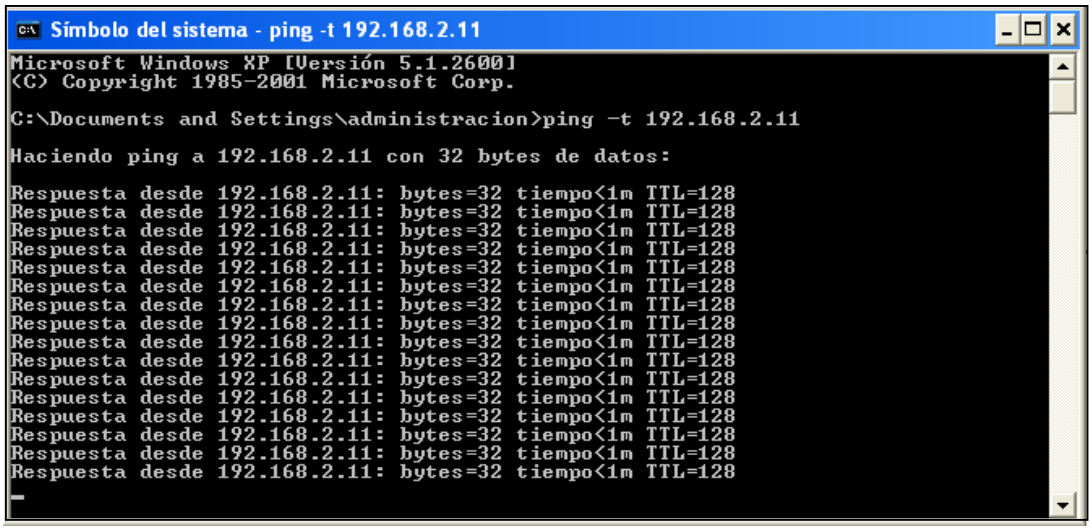


Figura No.50

b. Compruebe si la otra máquina a la que esta conectado a través del cable crossover es accesible mediante el comando *ping –t*. (Ver figura No.51).

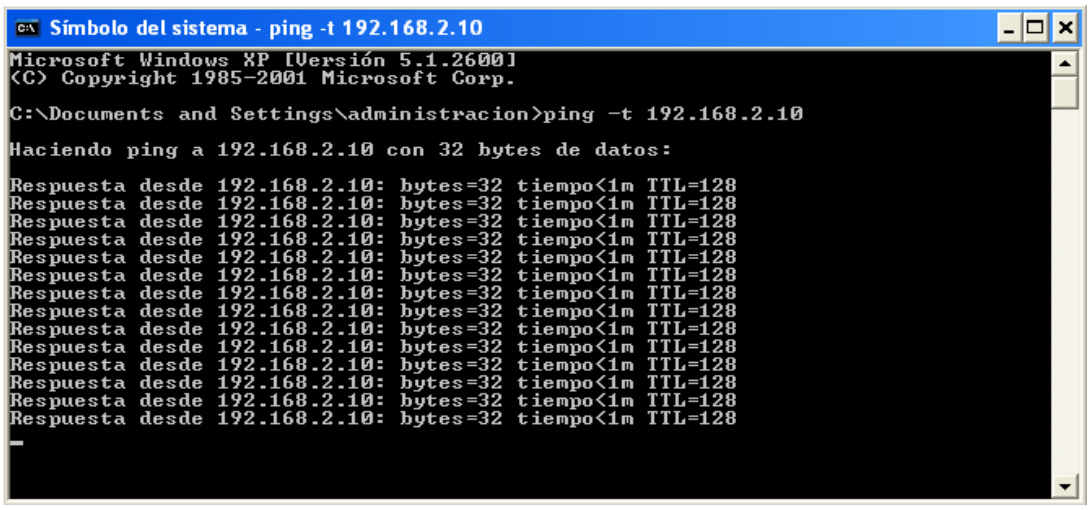


Figura No.21.

c. Desconecte su cable crossover de sus máquinas y conecte el cable de cada una al servidor del laboratorio de redes.

Apague la Computadora.

6.- Cuestionario

1. ¿Qué debería ser considerado cuando se selecciona una NIC al instalar en una computadora?

_____
_____
_____
_____

2. En el ambiente de las redes Microsoft ¿Qué es un grupo?

_____
_____
_____
_____

3. ¿Qué es Plug & Play?

_____
_____

4. ¿Qué parámetros se configuran en una Tarjeta de Red?

7.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica

PRÁCTICA 3
Instalación de una Red Básica en Plataforma Windows XP de Microsoft
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. Investigue cómo se forma una dirección MAC y para qué sirve

2. Investigue las diferencias entre un tarjeta tipo ISA y una PCI

3. ¿Qué es un dispositivo dual speed?

4. Si conecta su computadora a un dispositivo dual speed, ¿debe utilizar una NIC de 10Mbps o una NIC de 100Mbps?

5. ¿Qué tipo de NIC debe emplear si esta instalando una red que utiliza fibra óptica?

6. ¿Qué es un cliente, protocolo, adaptador y servicio en una red?

7. ¿Qué características tiene el protocolo NETBEUI?

PRÁCTICA 4
Instalación de una Red Básica en Plataforma LINUX

1.- Objetivo de Aprendizaje:
El alumno:

- Al finalizar la práctica tendrá la capacidad de instalar y configurar una tarjeta de red.
- Será capaz de instalar una LAN básica conectando dos computadoras utilizando un cable de conexión cruzada (*crossover*).

2.- Conceptos teóricos:

Una tarjeta de Red (NIC – *Network Interface Card*) es el dispositivo que conecta a una estación, PC u otro equipo de red con el medio físico. El tipo de conector de la tarjeta de red dependerá de las características del medio de comunicación de la red: (par trenzado, coaxial, fibra óptica, aire) al cual se conecte. (Ver figuras No.1 y No.2).



Figura No.1. Tarjeta de Red con conector RJ-45 para cable UTP.

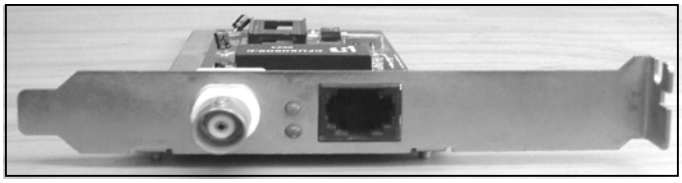


Figura No.2. Tarjeta de Red con conectores BNC para cable coaxial y RJ-45 para UTP.

Se define en la capa 2 del modelo OSI debido a que tiene y reconoce direcciones MAC (Subnivel de la capa de Enlace). Contienen un código único en todo el mundo, que se llama dirección de Control

de Acceso al Medio (MAC, Media Access Control). Esta dirección se utiliza para controlar la comunicación de datos para el host en la red.

La NIC es el componente de hardware básico en las comunicaciones de red. Traduce la señal producida por la computadora en un formato serie que se envía mediante el cable de red. La comunicación binaria (unos y ceros) se transforma en impulsos eléctricos, pulsos de luz, ondas de radio o cualquier esquema de transmisión de señales que usen los medios de comunicación en red, de manera que convierte el intercambio de señales a través de los medios de transmisión en una comunicación de datos efectiva.

Las funciones de la tarjeta de red son:

- Preparar los datos del equipo (formar tramas) para pasarlos a la capa física.
- En salida transferir las tramas al medio físico de transmisión según el protocolo de comunicación.
- Recibir los datos que llegan por el cable y convertirlos en bytes para que puedan ser comprendidos por la unidad de procesamiento central del equipo (CPU).
- Controlar el flujo de datos entre el equipo y el sistema de cableado.

3.- Equipo y material necesario.

Material que debe traer el alumno:

- Desarmador de cruz.
- Cable cruzado (*crossover*) construido en la práctica 1.

Equipo del Laboratorio:

- 1 Computadoras con Sistema Operativo Linux Mandrake 6.0.
- Ranura de expansión PCI o ISA vacía.
- Tarjeta de red PCI o ISA.
- Controlador de la tarjeta de red.

4.- Desarrollo:

4.1 Modo de trabajar

La instalación de la LAN básica se realizará en equipos de dos personas por cada computadora del laboratorio.

El equipo donde se desarrolle la práctica deberá contar con privilegios de administrador.

4.2. Instalación de la Tarjeta de Red.

4.2.1 Apague la computadora y desenchufe el cable de alimentación.

4.2.2 Retire la placa contra el polvo del gabinete e identifique la ranura de expansión PCI o ISA vacía en la que desea instalar la tarjeta NIC (Ver figura No.3). Actualmente todas las tarjetas de red son del tipo PCI. (Ver figura No. 4).

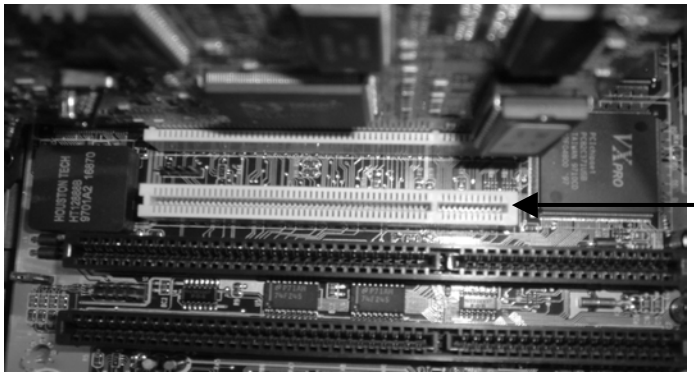


Figura No.3. Identificación de ranura de expansión para Tarjeta de Red PCI.

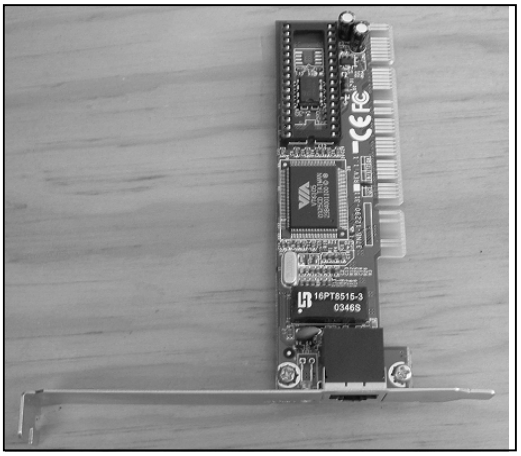


Figura No. 4 Tarjeta de Red PCI.

4.2.3 Sujete los bordes superiores de la tarjeta de red con ambas manos y coloque la tarjeta en la ranura de expansión. (Ver figura No.5).

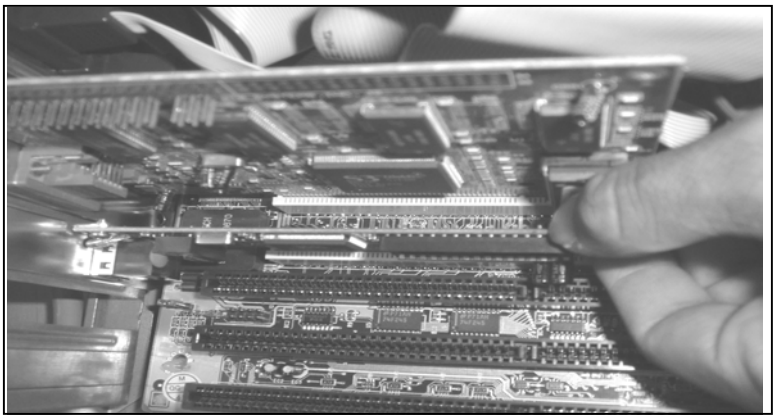
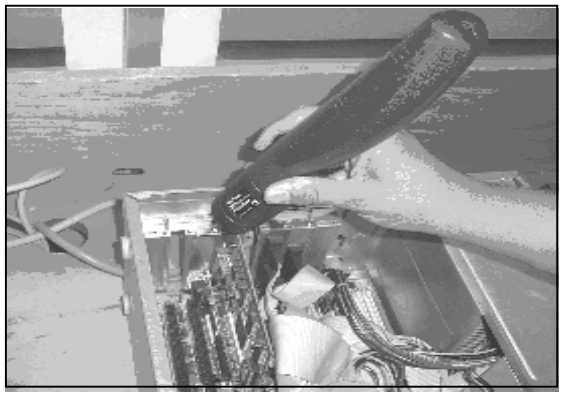


Figura No.5. Colocación de la Tarjeta de Red en la ranura de expansión.

4.2.4 Por último, use un tornillo para fijar la tarjeta en el gabinete. (Ver figuras No.6 y 7).





Figuras No.6 y 7. Fijación de Tarjeta de Red en el gabinete utilizando un tornillo y el desarmador.

4.2.5 Enchufe el cable de alimentación, encienda la computadora y elija cargar Linux. El detector de hardware Kudzu de Linux determina automáticamente el controlador del adaptador para la tarjeta de red y lo llevara paso a paso en la configuración de la NIC. (Ver figuras No.8 y No.8a).

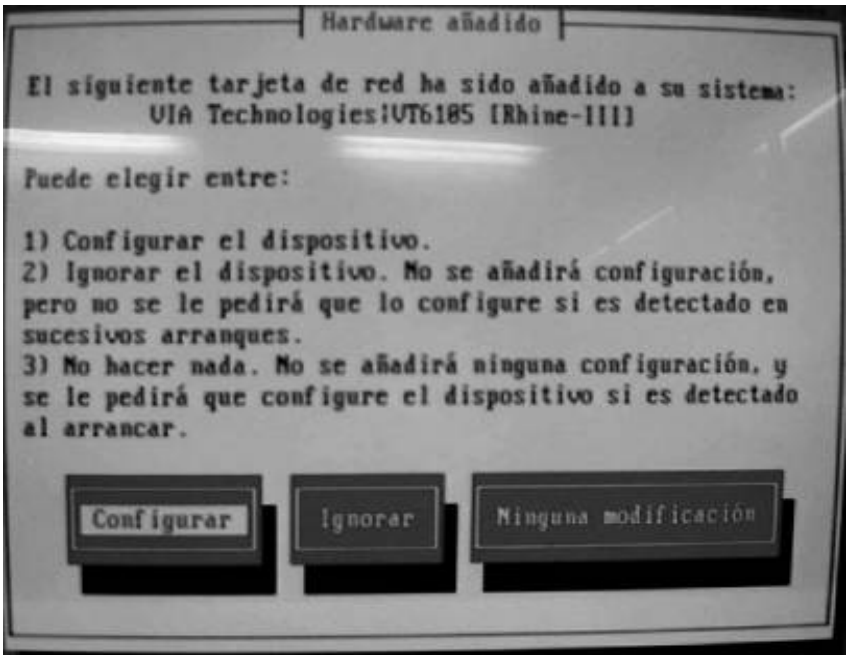


Figura No.8

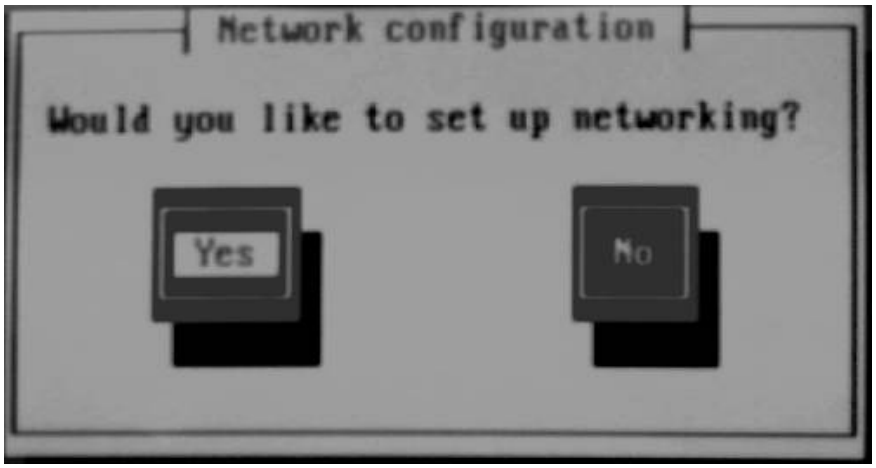


Figura No.8ª

4.2.6 En caso de que el propio sistema no reconozca la tarjeta de red y usted no posea el disco del driver de la NIC haga lo siguiente, si posee el disco pase al punto 4.2.7.

a. Inicie una sesión como root. El administrador le proporcionará la contraseña. (Ver figura No.9).



Figura No.9.

b. Abra un shell prompt. (Ver figura No.10).

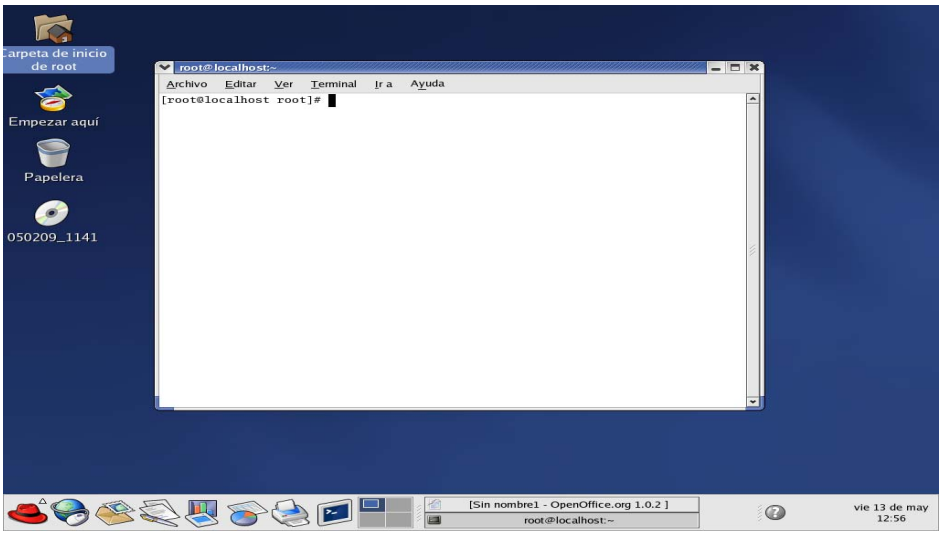


Figura No.10.

c. Obtenga el modelo del chip de la tarjeta de red. Esto puede determinarse examinando físicamente la tarjeta de red o bien teclee el siguiente comando:

[root@localhost root]# less /proc/pci | grep Ethernet

Lo cual devuelve una salida similar a la siguiente: (Ver figura No.11).

Ethernet controller: PCI device 1106:3106 (VIA Technologies, Inc) (rev 134).

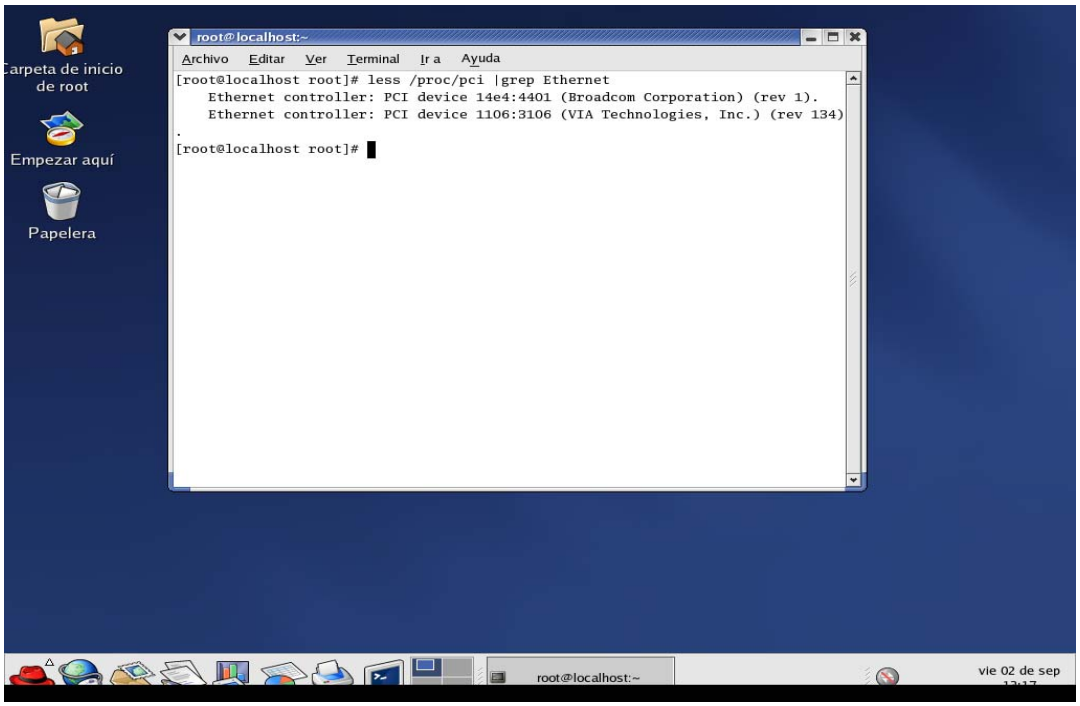


Figura No.11.

d. Verifique la versión del kernel de su máquina. Teclee el siguiente comando: (Ver figura No.12).

[root@localhost root]# uname –r

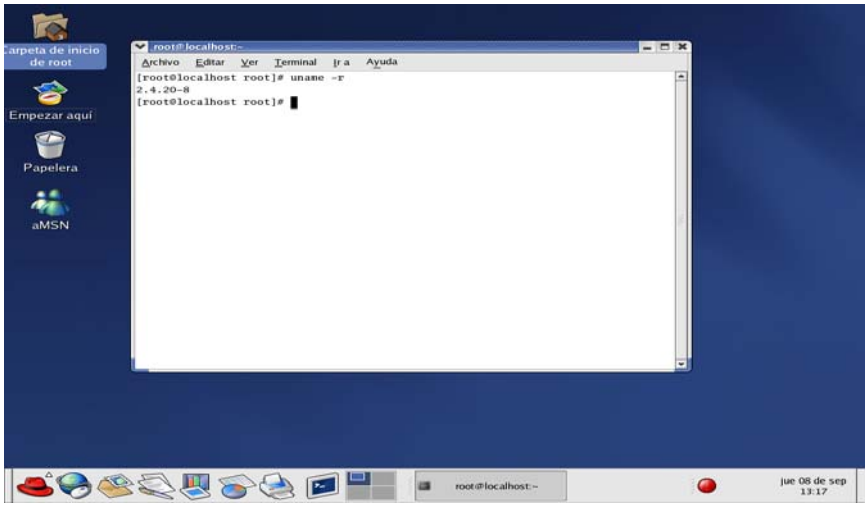


Figura No.12.

e. Si su versión de kernel es 2.4.x, liste el contenido del directorio /lib/modules/versión_kernel/kernel/drivers/net/ y si es 2.2.x liste el directorio /lib/modules/versión_kernel/kernel/net/ y busque el modulo adecuado para su NIC. (Ver figura No.13).

[root@localhost root]# ls /lib/modules/version_kernel/kernel/drivers/net
ó
[root@localhost root]# ls /lib/modules/version_kernel/kernel//net

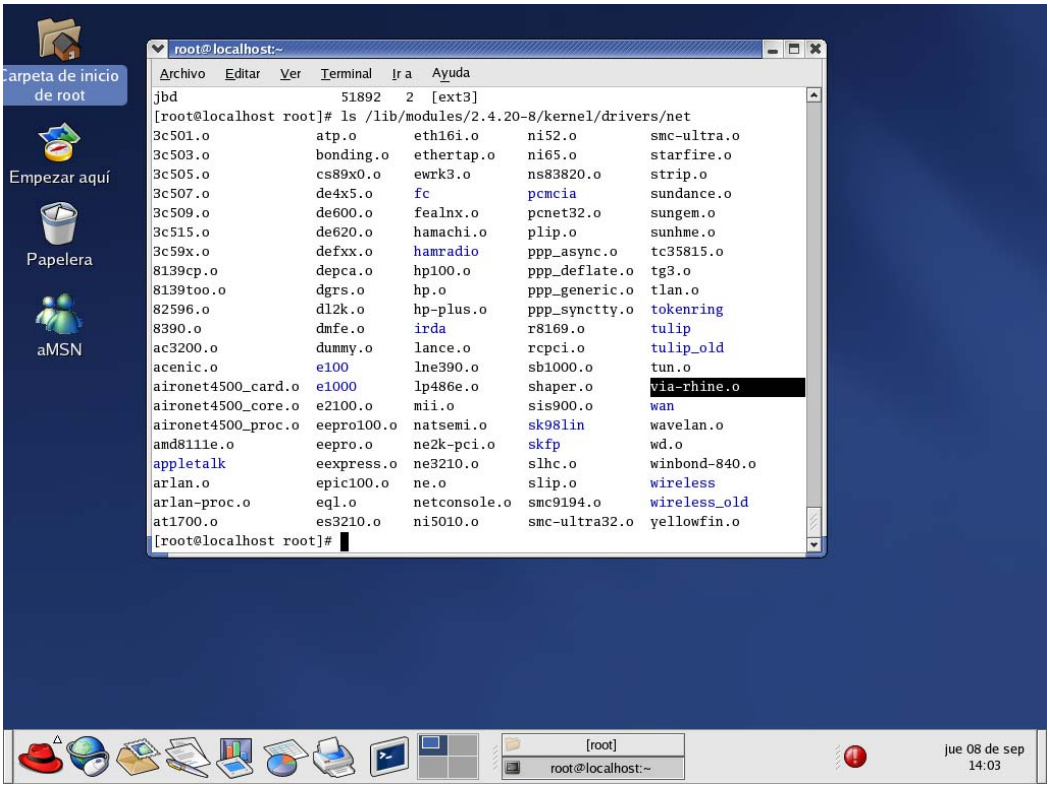


Figura No.13.

f. Edite con el procesador de textos *vi* o *vim* el archivo */etc/modules.conf* o *modprobe.conf* (según su distribución de *Linux*). (Ver figura No.14).

[root@localhost nombredriver]# cd
[root@localhost root]# vi /etc/modules.conf
ó
[root@localhost root]# vi modprobe.conf

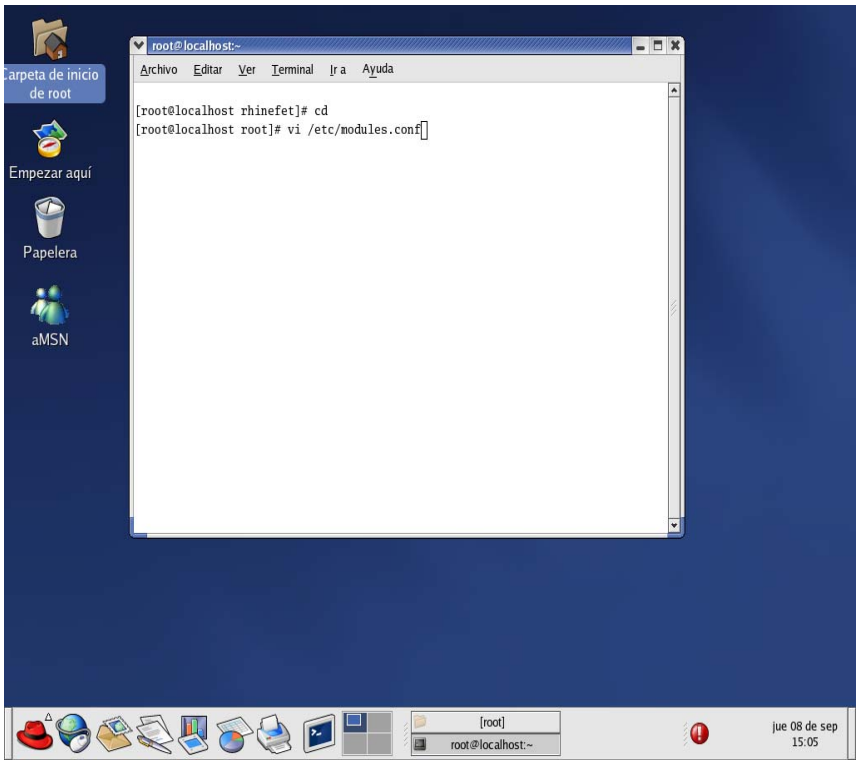


Figura No.14.

Dicho archivo debe contener lo siguiente: alias eth0 nombredriver. (Ver figura No.15).

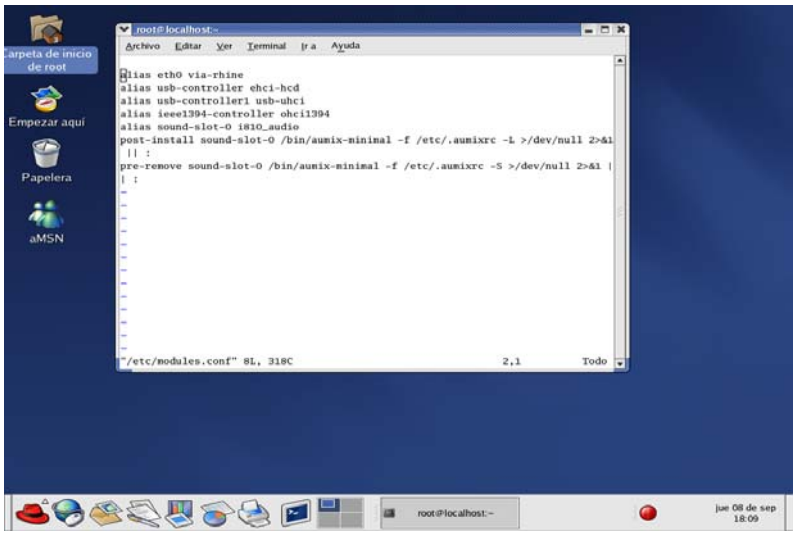


Figura No.15.

g. Salga de sesión y reinicie la computadora.

h. Inicie una sesión como usuario root y abra un shell prompt.(Ver figura No.16).

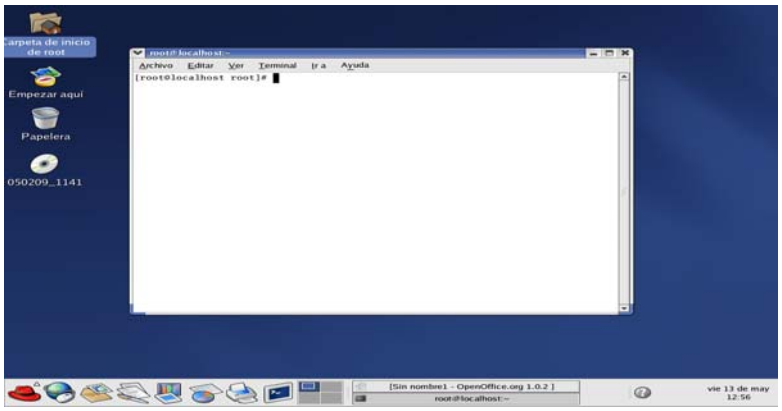


Figura No.16

i. Instale el modulo del driver (si el módulo del driver está en el lugar incorrecto un mensaje de error aparecerá y dirá que no puede encontrar el modulo del driver) y liste los módulos cargados para asegurarnos que está funcionando correctamente: (Ver figura No.17).

[root@localhost root]# insmod nombredriver
[root@localhost root]# lsmod

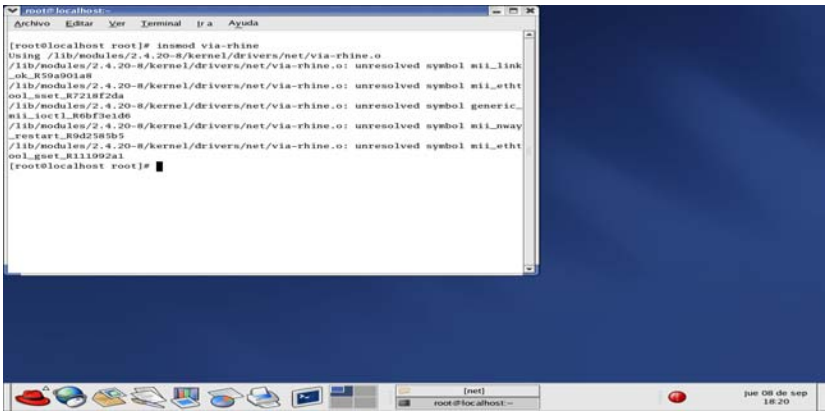


Figura No.17.

De esta manera hemos cargado el módulo para la NIC. Si el sistema indica algún error, deberá asegurarse de que ese es el módulo adecuado para su tarjeta de red o probar otro similar.

4.2.7 En caso de que el propio sistema no reconozca la tarjeta de red pero usted posea el disco del driver de la NIC, haga lo siguiente:

a. Inicie una sesión como root. El administrador le proporcionará la contraseña. (Ver figura No.18).



Figura No.18.

b. Abra un shell prompt. (Ver figura No.19).

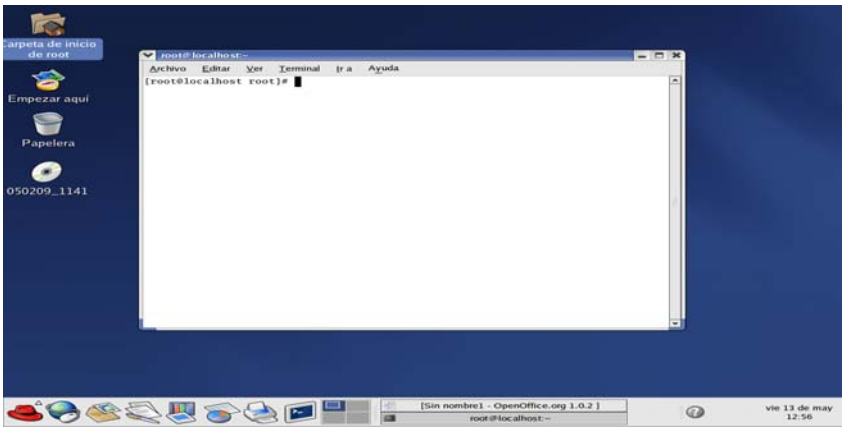


Figura No.19.

c. Cree un directorio temporal *(/temp*): (Ver figura No.20).

[root@localhost root]# mkdir /temp

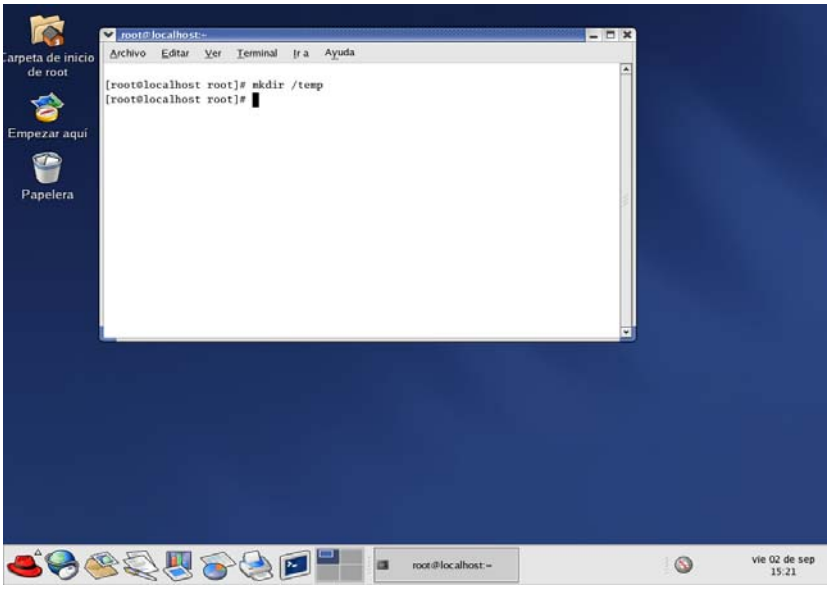


Figura No.20.

d. Cámbiese al directorio /temp: (Ver figura No.21).

[root@localhost root]# cd /temp

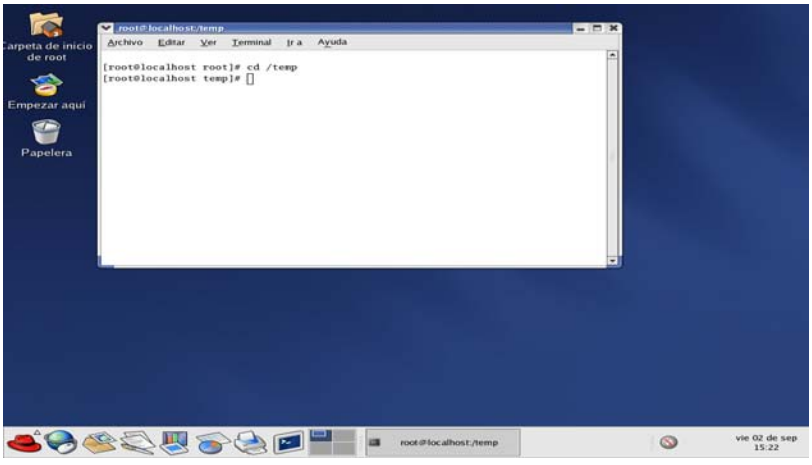


Figura No.21.

e. Monte el disco del controlador, liste su contenido, busque el controlador (usualmente son .tgz) y copie el driver en el directorio */temp*: (Ver figura No.22).

[root@localhost temp]# mount -t msdos /dev/fd0 /mnt
[root@localhost temp]# ls –lia /mnt/linux
[root@localhost temp]#cp /mnt/linux/nombredriver.tgz /temp

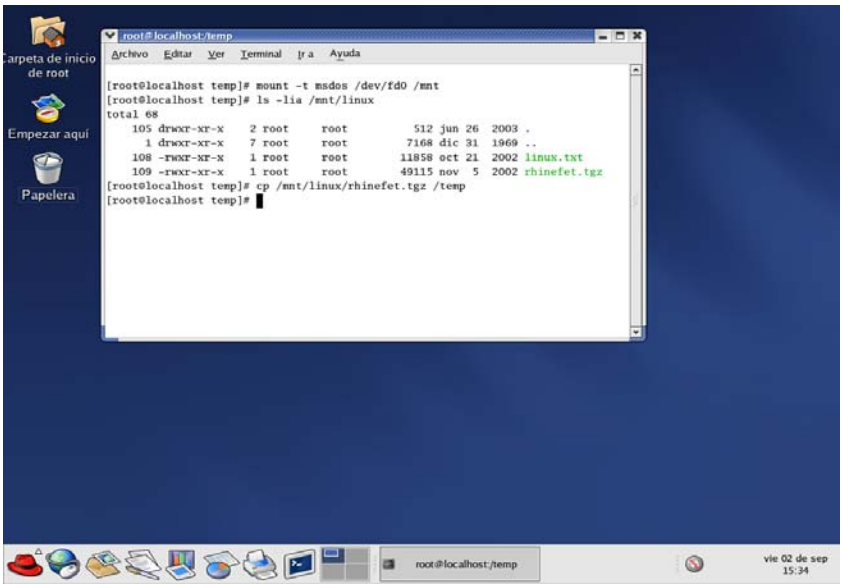


Figura No. 22.

f. Descomprima el archivo del controlador y cámbiese al directorio del driver: (Ver figura No.23).

[root@localhost temp]# tar xzvf nombredriver.tgz
[root@localhost temp]# cd nombredriver

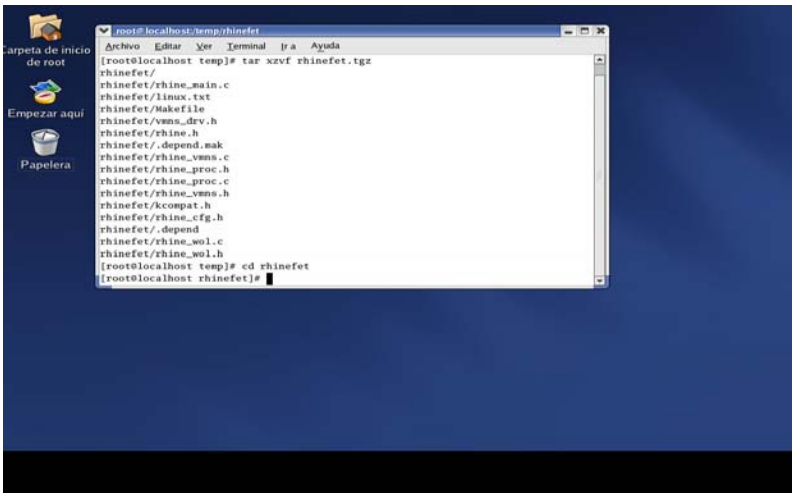


Figura No.23

g. Compile el archivo fuente del driver, eso generará el archivo .o, y copiará éste en el path correcto de instalación. (Ver figura No.24).

   (/lib/modules/versiónkernel/kernel/drivers/net/ o /lib/modules/versiónkernel/kernel/net/).
   [root@localhost nombredriver]# make install

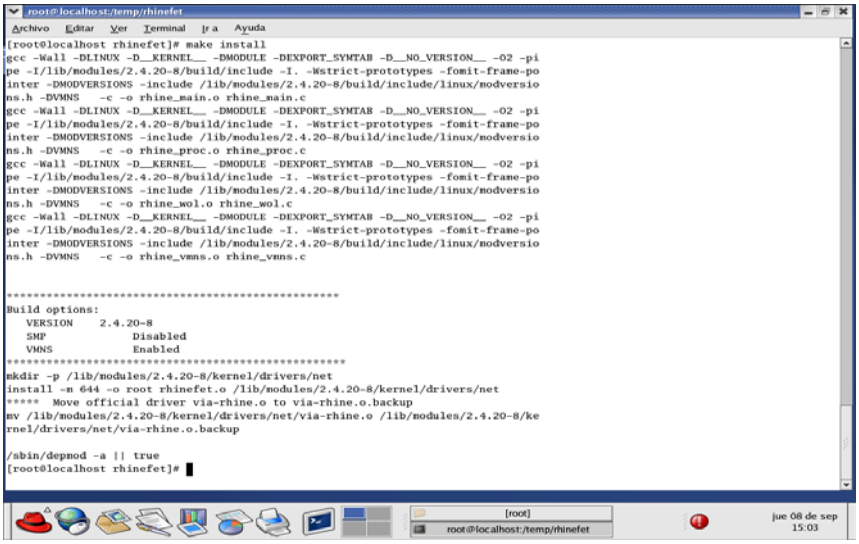


Figura No.24.

h. Edite con el procesador de textos *vi* o *vim* el archivo */etc/modules.conf* o *modprobe.conf* (según su distribución de *Linux*). (Ver figura No.25).

[root@localhost nombredriver]# cd
[root@localhost root]# vi /etc/modules.conf
ó
[root@localhost root]# vi modprobe.conf

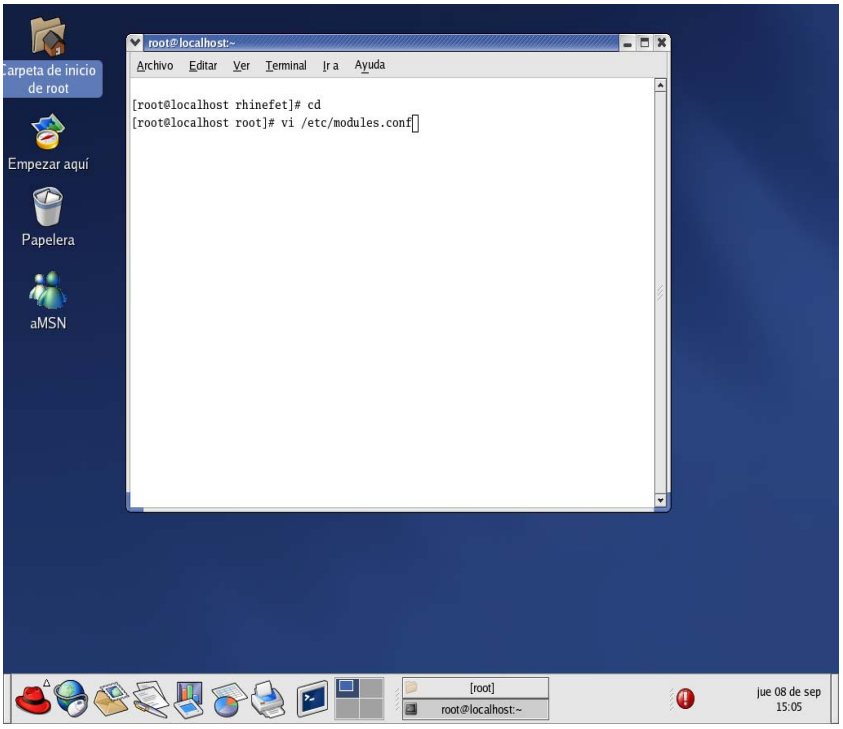


Figura No.25.

Dicho archivo debe contener lo siguiente: alias eth0 nombredriver. (Ver figura No.26).

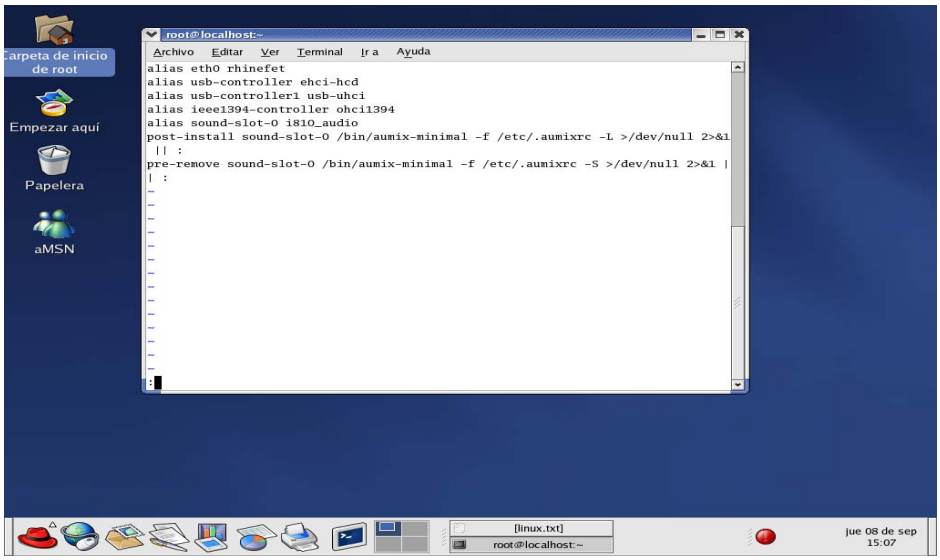


Figura No.26.

i. Salga de sesión y reinicie la computadora.

j. Inicie una sesión como usuario root y abra un shell prompt. (Ver figura No.27).

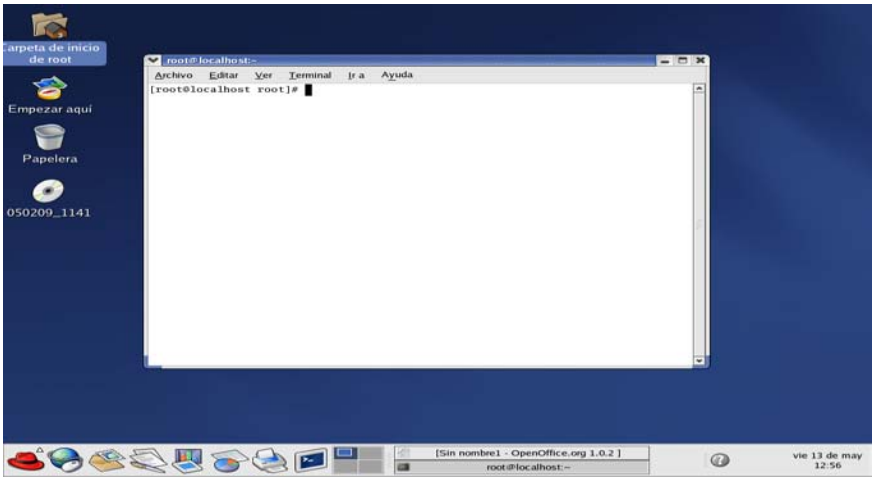


Figura No.27.

k. Instale el modulo del driver (si el módulo del driver está en el lugar incorrecto un mensaje de error aparecerá y dirá que no puede encontrar el módulo del driver) y liste los módulos cargados

para asegurarnos que esta funcionando correctamente: (Ver figura No.28).

[root@localhost root]# insmod nombredriver
[root@localhost root]# lsmod

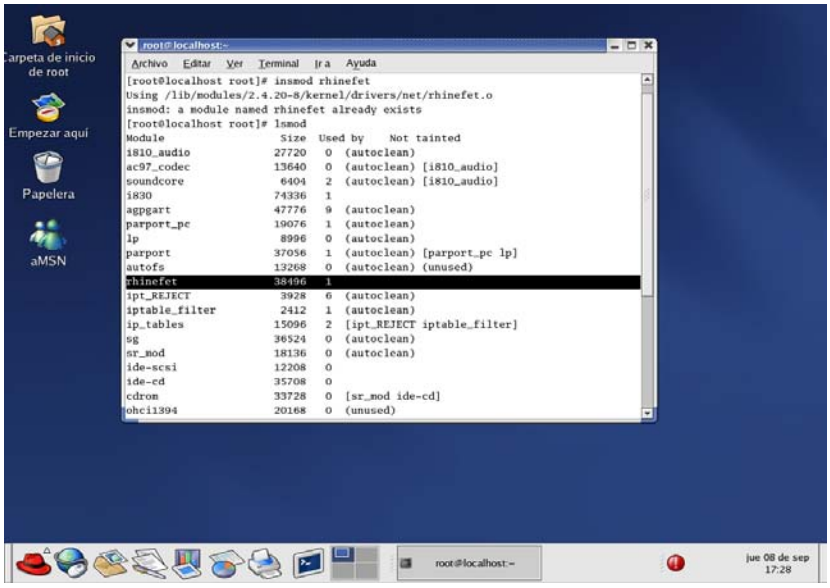


Figura No.28.

Conecte dos computadoras a través de la tarjeta de red empleando el cable cruzado (*crossover)* construido en la práctica 1.

Nota.- Cabe mencionar que después de instalar y configurar la tarjeta de red, se conectarán dos computadoras para probar la correcta construcción del cable cruzado. (Construido en la Práctica No.1). (Ver figura No.29).

Se necesitará conocer los nombres e IP asociadas de las computadoras involucradas (nombre_computadora1 y su IP asociada, nombre_computadora2 y su IP asociada).

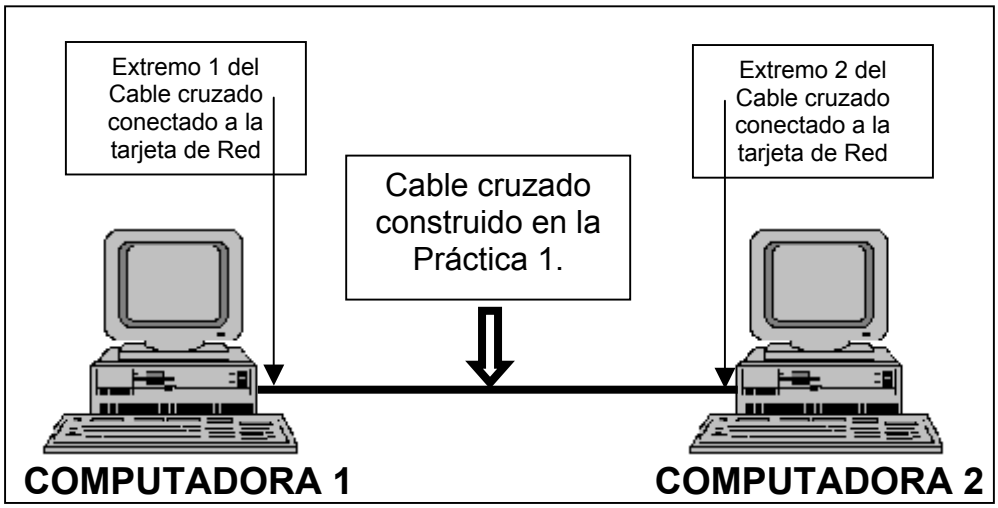


Figura No.29

4.3 Configuración de la Tarjeta de Red.

Una vez detectada la tarjeta de red, deberá configurarla. Para esta práctica se explicaran dos formas de realizarlo:

- Haciendo uso de scripts.
- A través de comandos desde el prompt (nuestro caso).

Nota: Los parámetros para la configuración de la NIC serán proporcionados por el administrador.

4.3.1Configuración de la NIC usando scripts.

4.3.1.1 Edite el archivo */etc/sysconfig/network-scripts/ifcfg-eth0* y coloque lo siguiente (Ver figuras No.30 y No.31).

[root@localhost root]# vi /etc/sysconfig/network-scripts/ifcfg-eth0

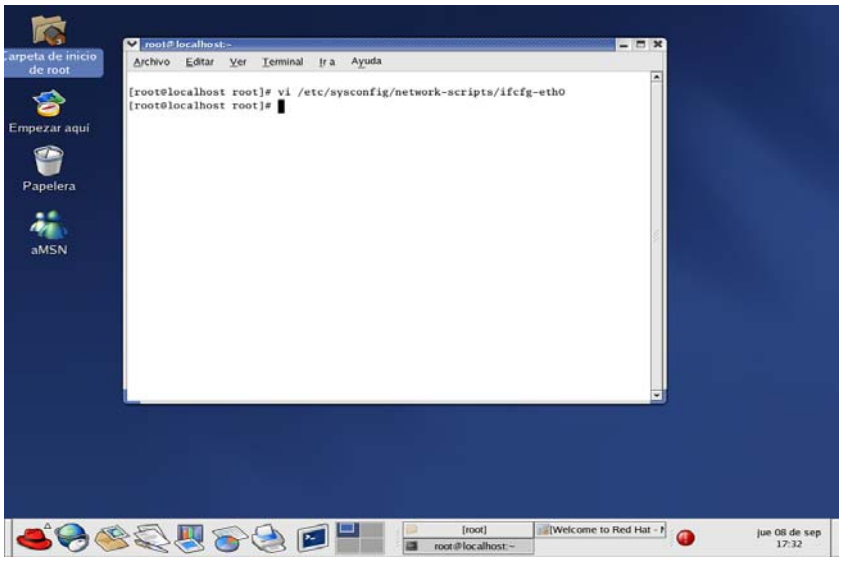


Figura No.30

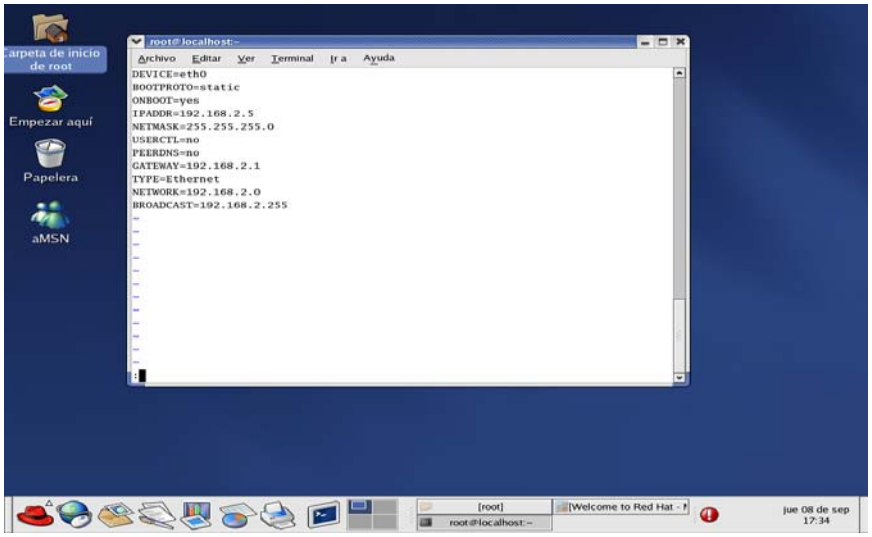


Figura No.31

DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=192.168.2.5.
NETMASK=255.255.255.0
GATEWAY=(colocar ip gateway)
NETWORK=192.168.2.0
BROADCAST=192.168.2.255

Si no esta el archivo, créelo.

4.3.1.2 En el archivo */etc/sysconfig/network* coloque lo siguiente:(Ver figuras No.32 y No.33).

[root@localhost root]# vi /etc/sysconfig/network

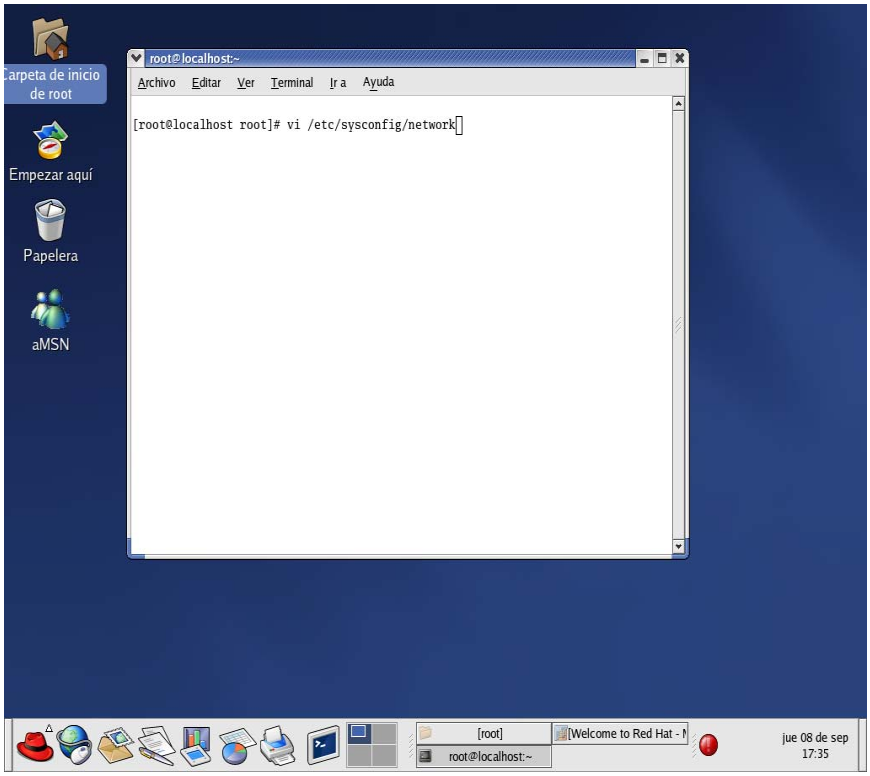


Figura No.32

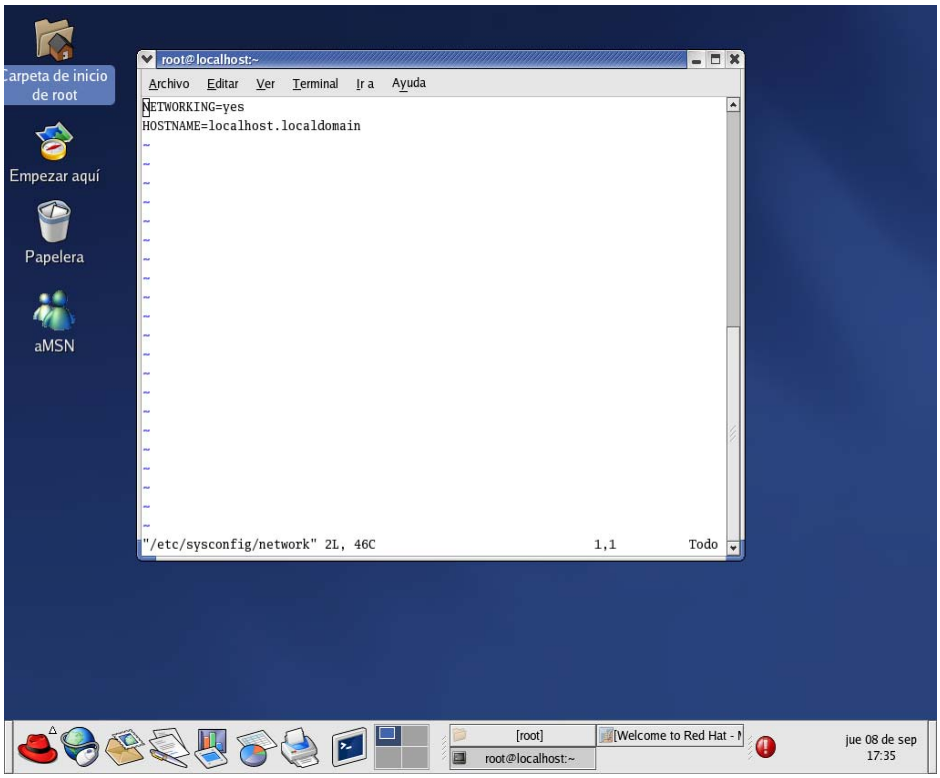


Figura No.33

NETWORKING=yes
HOSTNAME=nombre_maquina.unam.mx

4.3.1.3. Dentro del archivo */etc/resolv.conf* coloque los DNS (Ver figuras No.34 y No.35).

[root@localhost root]# vi /etc/resolv.conf

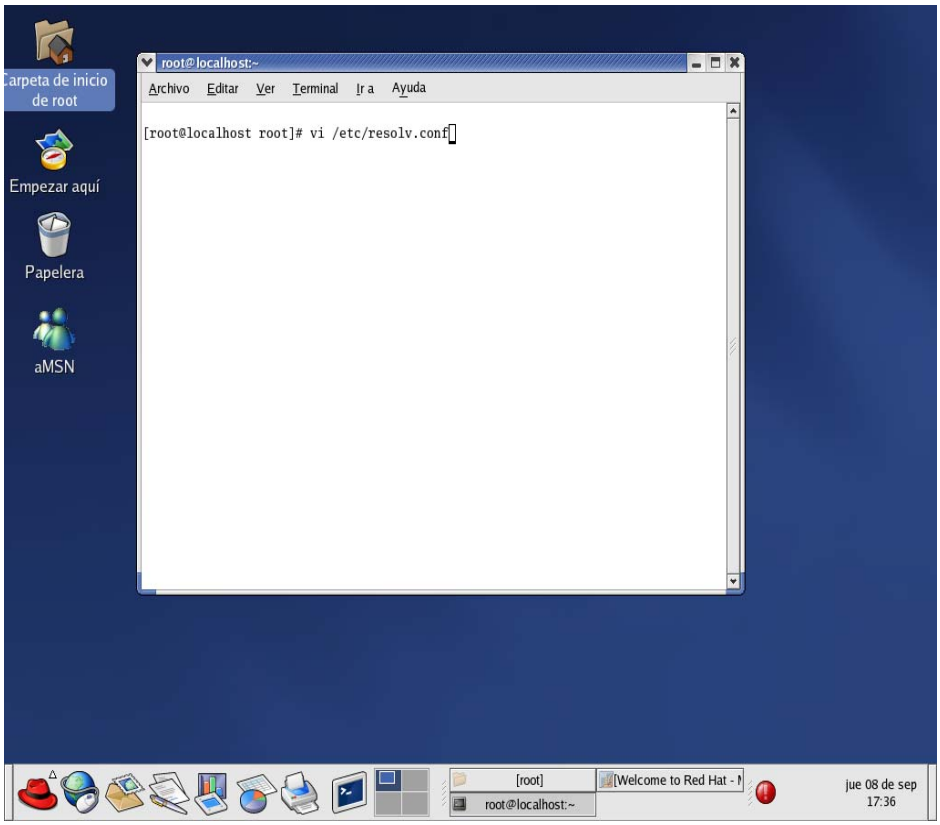


Figura No.34

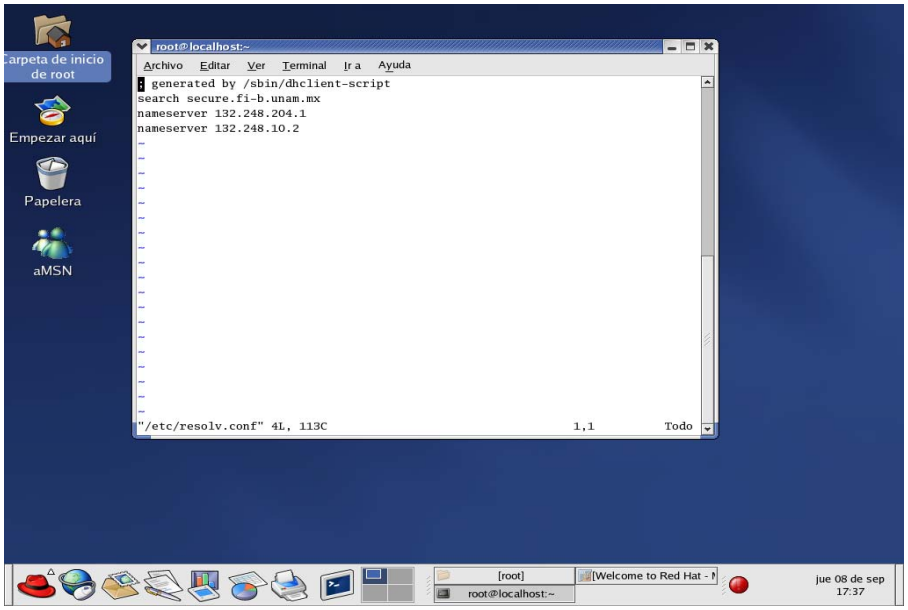


Figura No.35

search fi-b.unam.mx
nameserver 132.248.204.1
nameserver 132.248.10.2

4.3.1.4. Finalmente teclee: (Ver figura No.36).

[root@localhost root]# ifconfig eth0 up

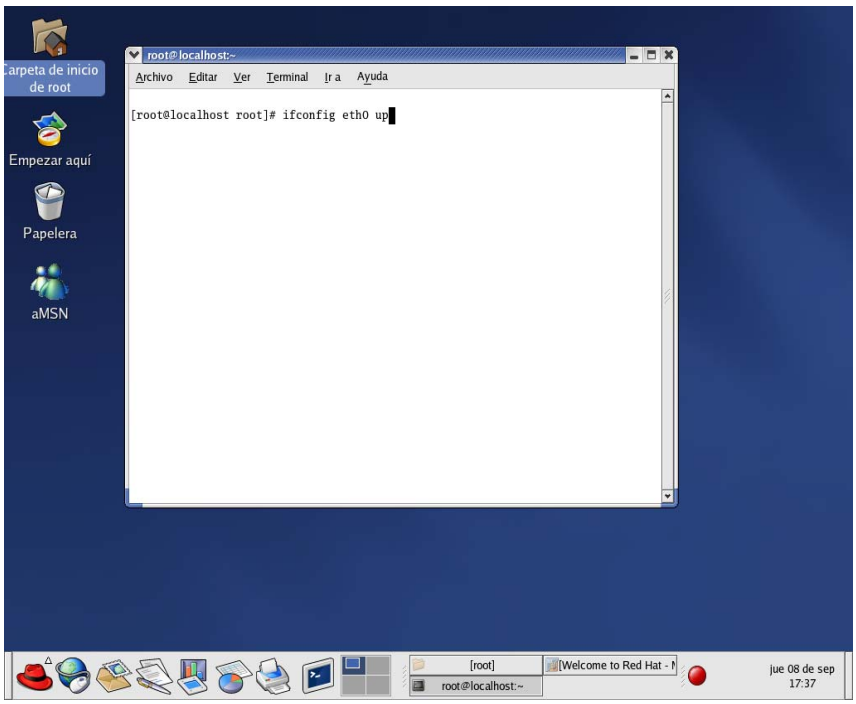


Figura No.36

4.4. Configuración de la NIC a través de comandos.

4.4.1. El comando utilizado para esta tarea es *ifconfig*. Utilice el siguiente comando para configurar la NIC (Ver figura No.37).

[root@localhost root]# ifconfig eth0 192.168.2.X netmask 255.255.255.0 pointtopoint ip_maquina_crossover

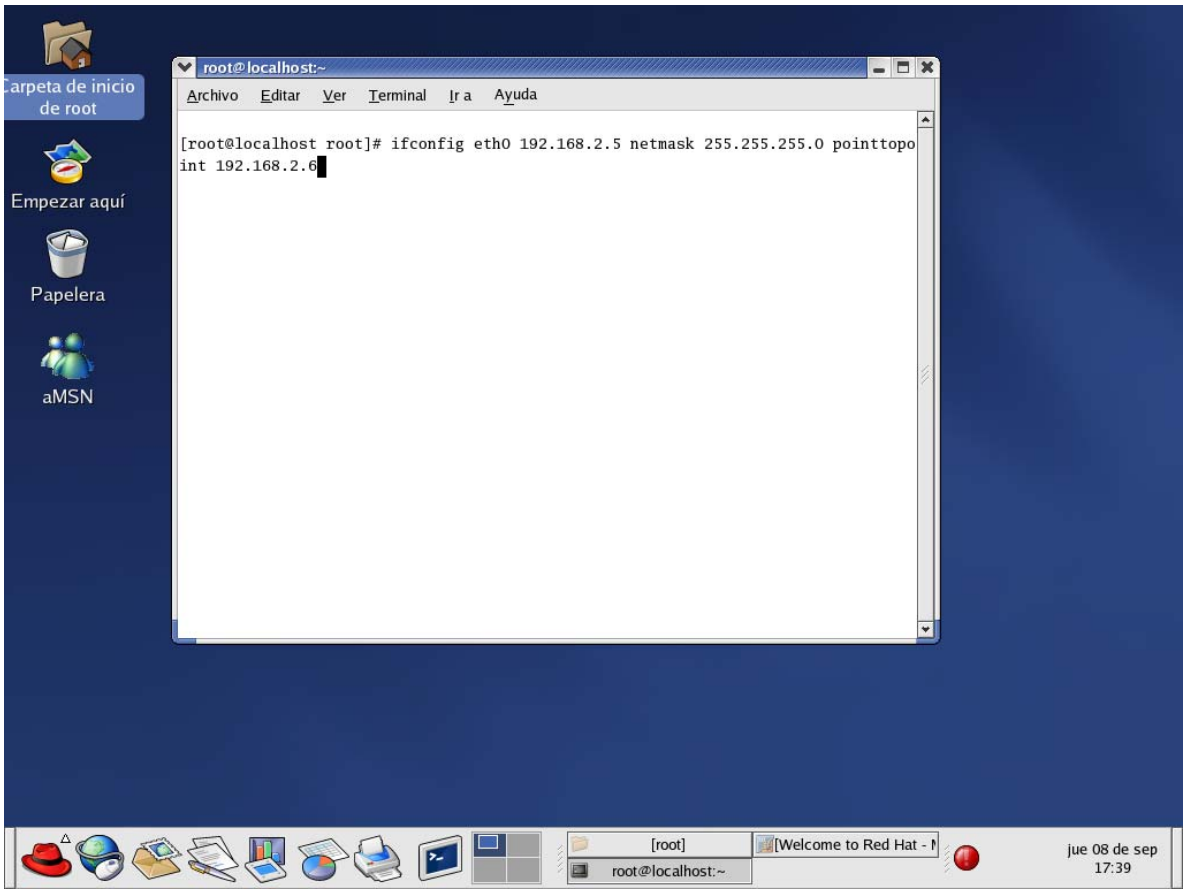


Figura No.37

Hasta este momento se ha instalado y configurado la tarjeta de red correctamente.

5.- Pruebas y Aplicaciones:

5.1 Comprobar que la LAN básica funcione correctamente.

5.1.1 Para comprobar la configuración actual de la NIC, utilice el siguiente comando (Ver figura No.38).

[root@localhost root]# ifconfig

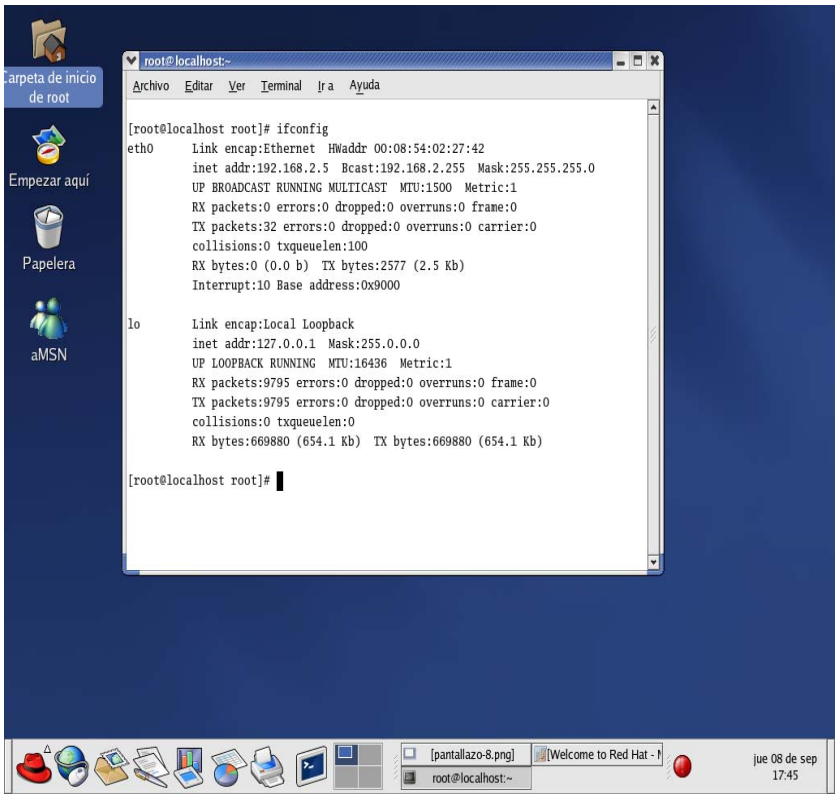


Figura No.38

El comando devuelve una salida similar a la siguiente:

lo Link encap:Local Loopback
inet addr:127.0.0.1 Bcast:127.255.255.255
Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING
MTU:3584 Metric:1
RX packets:57 errors:0 dropped:0 overruns:0
frame:0
TX packets:57 errors:0 dropped:0 overruns:0
carrier:0

eth0 Link encap:Ethernet Hwaddr 00:60:97:05:42:7E
inet addr:192.168.2.5 Bcast:192.168.5.255
Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST
MTU:1500 Metric:1
RX packets:2406345 errors:620 dropped:0

overruns:0
TX packets:2464404 errors:0 dropped:0
overruns:0
Interrupt:10 Base address:0x7f00

5.1.2 Verifique la salida de la máquina hacia aquella a la cual está conectada a través del cable crossover. Teclee el siguiente comando: (Ver figura No.39).

[root@localhost root]# ping ip_maquina_crossover

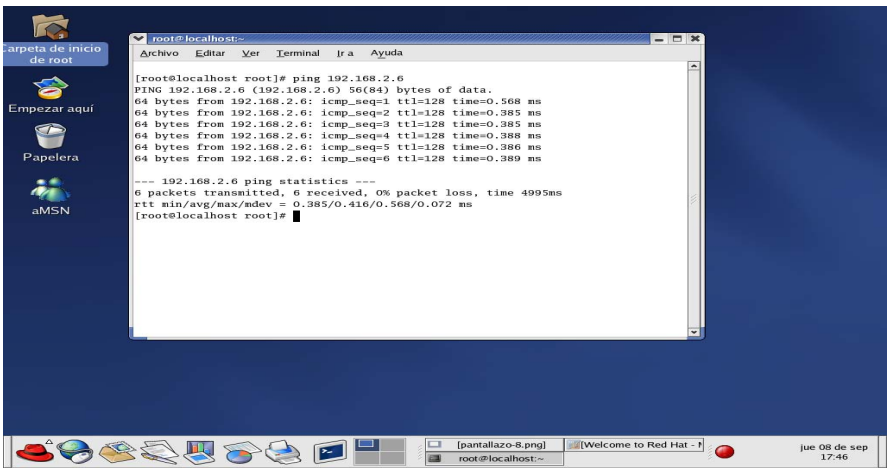


Figura No 39

Una respuesta afirmativa le asegurará que su dispositivo está correctamente configurado, por lo que fallas de comunicación pueden ser atribuidas a otras partes de la red. Anote textualmente la salida del comando:

_____
_____
_____
_____
_____
_____

5.1.3 Salga de sesión y apague la computadora.

6.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica:

PRÁCTICA 4
Instalación de una Red Básica en Plataforma LINUX
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. ¿Para qué nos sirve el protocolo TCP/IP?

2. ¿Qué es *Kudzu* y que función realiza?

3. ¿Por qué es importante conocer el modelo del chipset de la NIC?

4. ¿Qué significan cada uno de los parámetros en la versión del kernel (ejemplo: kernel 2.4.6)?

5. ¿Para qué sirve el comando ping?

PRÁCTICA 5
Enrutamiento de paquetes

1.- Objetivos de Aprendizaje
El alumno:

Al finalizar la práctica, se familiarizará con el manejo de algunas herramientas del Sistema Operativo Linux, como son route y traceroute, y sus similares en Windows, como son route y tracert, enfocadas al enrutamiento de paquetes a través de la red.

2.- Conceptos teóricos

Route

Este comando se utiliza para configurar las tablas de ruteo del núcleo de nuestro sistema. Generalmente en todo equipo de una red local tenemos al menos tres rutas: la de *loopback*, utilizando el dispositivo de bucle interno (lo, lo0...), la de red local (*localnet*), que utiliza la tarjeta de red para comunicarse con equipos dentro del mismo segmento de red, y una *default* que también utiliza la tarjeta para enviar a un *router* o *gateway* paquetes que no son para equipos de nuestro segmento.

Si *route* nos muestra una configuración sospechosa (esto es, las tablas no son las que en el sistema hemos establecido como administradores, aunque todo funcione correctamente) esto puede denotar un ataque de simulación: alguien ha desviado el tráfico por un equipo que se comporta de la misma forma que se comportaría el original, pero que seguramente analiza toda la información que pasa por él. Hemos de recalcar que esto suele ser transparente al buen funcionamiento del equipo (no notamos ni pérdida de paquetes, ni retardos excesivos, ni nada sospechoso), y que además el atacante puede modificar los ficheros de arranque del sistema para, en caso de reinicio de la máquina, volver a tener configuradas las rutas a su gusto; estos ficheros suelen del tipo /etc/rc.d/rc.inet1 o /etc/rc?.d/Sinet.

También es posible que alguien esté utilizando algún elemento utilizado en la conexión entre nuestro sistema y otro (un *router*, una pasarela...) para amenazar la integridad de nuestro equipo; si queremos comprobar el camino que siguen los paquetes desde que salen de la máquina hasta que llegan al destino, podemos utilizar la orden *traceroute*. Sin embargo, este tipo de ataques es mucho más difícil de detectar, y casi la única herramienta factible para evitarlos es la criptografía.

Traceroute

La orden traceroute se utiliza para imprimir la ruta que los paquetes siguen desde nuestro sistema hasta otra máquina, realizar pruebas, medidas y administración de una red; introduce mucha sobrecarga, lo que evidentemente puede acarrear problemas de rendimiento, llegando incluso a negaciones de servicio por el elevado tiempo de respuesta que el resto de aplicaciones de red pueden presentar.

Traceroute es una herramienta que combina muy inteligentemente, dos características de los protocolos que hacen posible Internet. Éstos son:

TTL o expiración de los paquetes

Para proteger a Internet del efecto de paquetes atrapados en ciclos de enrutamiento, los diseñadores de TCP/IP dotaron a cada datagrama IP de un contador que llamaron TTL por las siglas de *Time To Live*. Esto es un número que limita cuántos *saltos* puede dar un datagrama, antes de ser descartado por la red.

Cuando se introduce un datagrama IP a la red, el campo TTL es poblado con el número máximo de saltos que define la vida de ese datagrama. Cada enrutador por el que ese datagrama transita, resta uno a ese número. Cuando éste llega a cero, el datagrama es descartado.

Internet Control Message Protocol ó ICMP

ICMP sirve para manejar mensajes de control. Esto son mensajes administrativos entre nodos de Internet.

Los paquetes ICMP sirven para muchas cosas: Avisar que un enlace o que un dispositivo están congestionados, que se escogió un camino sub-óptimo para enviar un paquete, que no se puede acceder a un sitio en particular, etc. Uno de esos avisos es particularmente útil para traceroute: El aviso de que se excedió la vida útil del paquete.

Combinando estas dos herramientas, traceroute permite construir un mapa de la red de acuerdo como es vista desde un nodo en particular. A continuación, un ejemplo del comando traceroute hacia el servidor de yahoo.

[root@localhost root]$ traceroute www.unam.mx
traceroute to dragon.dgsca.unam.mx (132.248.10.7), 30 hops max, 38 byte packets

1 192.168.2.1 (192.168.2.1) 0.808 ms 0.551 ms 0.573 ms
2 fe3-15-iimas-core.fi-b.unam.mx (132.248.59.254) 1.244 ms 2.477 ms 1.222 ms
3 vel96-dgsca-core-ge.unam.mx (132.148.255.197) 2.276 ms 1.327 ms 1.513 ms
4 gb4-3-dgsca-core.ge.umam.mx (132.248.255.109) 1.318 ms 1.284 ms 2.387 ms
*5 dragon.dgsca.unam.mx (132.248.10.7) 1.569ms * 1.879ms*

Aquí se muestra cada uno de los saltos que tiene que dar un paquete al recorrer el camino desde la computadora hasta www.unam.mx. La dirección del recorrido es muy importante, porque en Internet no necesariamente el camino de ida es igual al de regreso.

Para cada salto, que traceroute numera en la primera columna, se envían tres paquetes con un TTL que se va incrementando progresivamente. El nombre del nodo que responde, bien sea indicando que se llegó al destino o que el paquete expiró en tránsito, se muestra junto a los tiempos transcurridos entre el envío del paquete

y la recepción de la respuesta. Eso nos da una idea de cómo es la condición de la red en ese salto o en alguno de los anteriores.

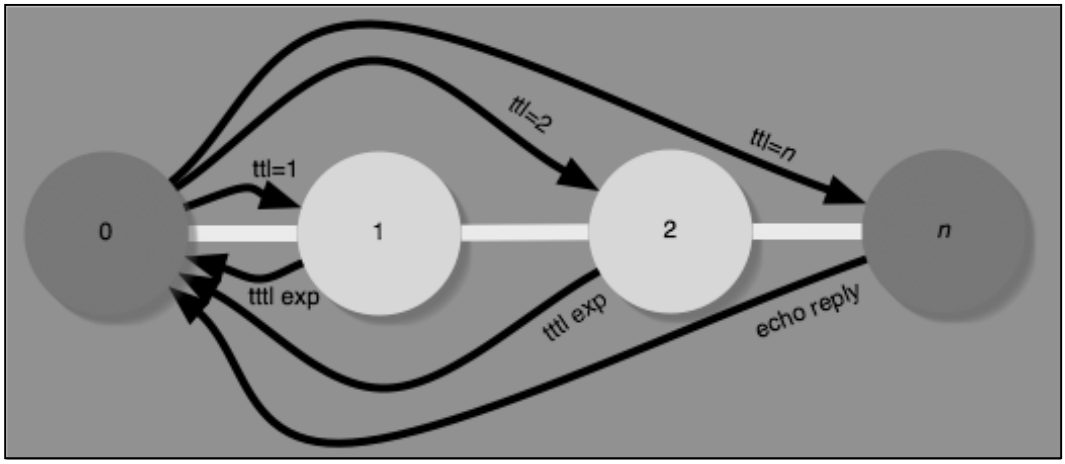


Figura No.1

El ejemplo anterior permite ver mejor cómo funciona la herramienta. (Ver figura No.1). En el primer salto, hacia el nodo 1, traceroute pone el valor ttl en 1 y envía el paquete hacia el nodo de destino. Cuando el nodo 1 decrementa el valor del ttl y obtiene un cero, devuelve al nodo de origen un mensaje de error que dice que el ttl expiró mientras el paquete iba en tránsito. Este proceso se repite varias veces y los tiempos se registran.

Para el siguiente salto, traceroute aumenta en uno el valor del ttl y lo envía de nuevo hacia su destino. El nodo 1 decrementa el valor del ttl a uno y pasa el paquete hacia el nodo 2. El nodo 2 recibe el paquete con ttl uno y al decrementarlo, obtiene un ttl cero, enviando el correspondiente mensaje de error hacia el nodo de origen. Este proceso se va repitiendo con valores progresivamente más grandes de ttl, para ir encontrando los saltos cada vez más lejanos o hasta que se llega a un ttl muy grande. Típicamente este valor máximo es 30, aunque puede ser de hasta 255.

3.- Equipo y material necesario

Material que deberá traer el alumno:

A. Lápiz ó pluma.

Equipo del Laboratorio:

B. 1 Computadora con Sistema Operativo Linux Red Hat 9.0 ó Superior y Windows XP.

4.- Desarrollo

4.1 Modo de trabajar

Se trabajará en equipos de dos personas por computadora.

4.2 Realización de la práctica

4.2.1 Encienda el sistema y elija la opción de cargar *Linux*.

4.2.2 Entre a sesión como usuario root. El profesor le proporcionará la contraseña. (Ver figura No.2).



Figura No.2

4.2.3 Abra un shell prompt seleccionando (Ver figura No.3):

Main Menu => System Tools => Terminal.

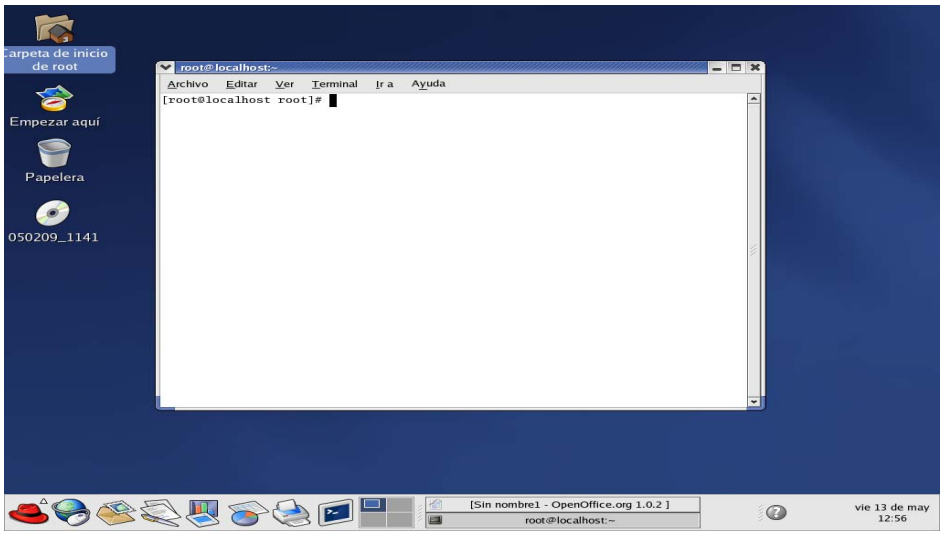


Figura No.3

Para salir del shell prompt teclee exit o presione <ctrl>d en el prompt.

4.2.4 Ahora podemos emplear el sistema para ejecutar los comandos que necesitemos utilizar.

4.2.5 Monitorice la interfaz eth0. (Ver figura No.4): Teclee lo siguiente:

[root@localhost root]$ tcpdump -i eth0

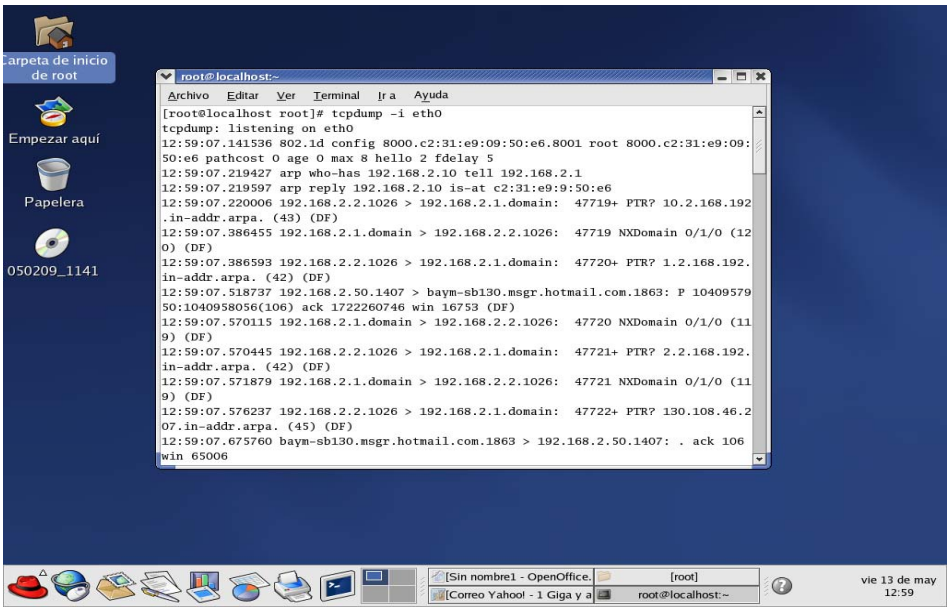


Figura No.4

Anote la salida en pantalla y coméntela.

4.2.6 Capture 3 paquetes salientes de la interfaz eth0. Teclee lo siguiente y anote la salida (Ver figura No.5):

[root@localhost root]$ tcpdump -c 3 -i eth0

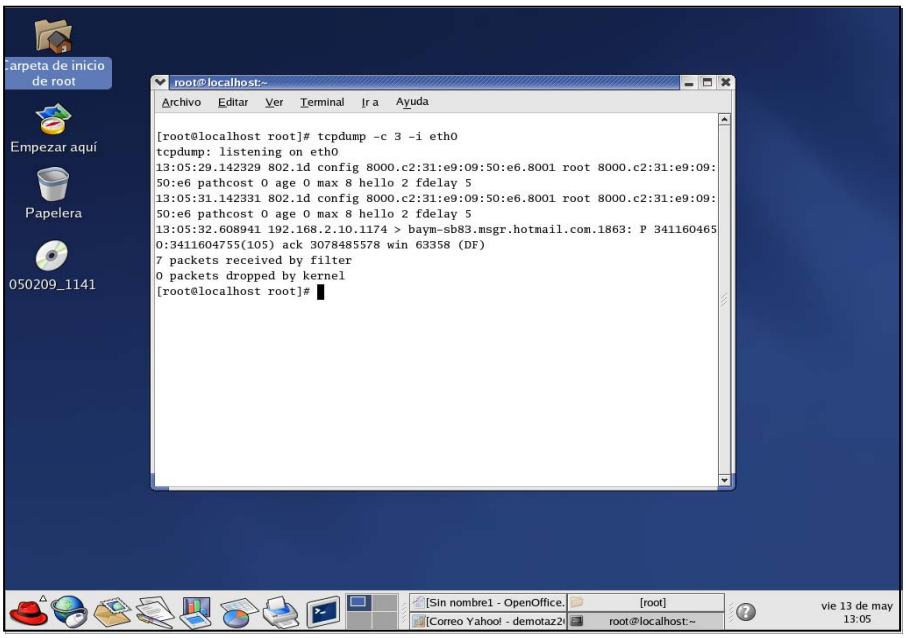


Figura No.5

4.2.7 Capture toda la trama Ethernet. (Ver figura No.6). Teclee lo siguiente:

[root@localhost root]$ tcpdump -s 1500

Teclee <ctrl>c *para detener la captura.*

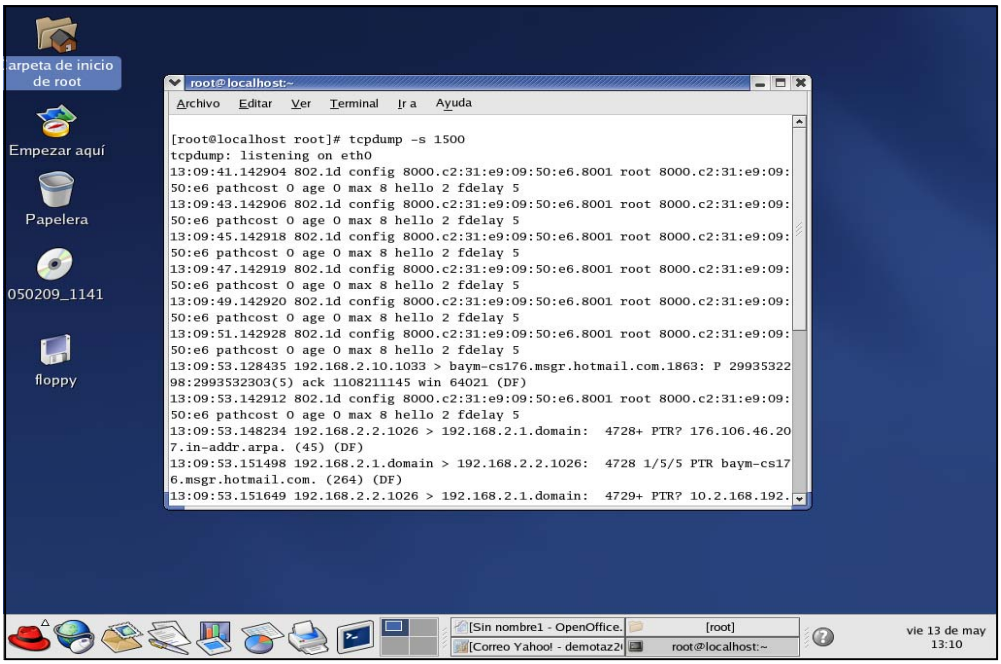


Figura No. 6

4.2.8 Escriba un fragmento de la trama y explique que significan cada una de las partes visualizadas.

_____

_____

_____

_____

_____

_____

_____

_____

_____

4.2.9 Visualice la configuración actual de las tablas de ruteo. (Ver figura No.7). Teclee lo siguiente:

[root@localhost root]$ route

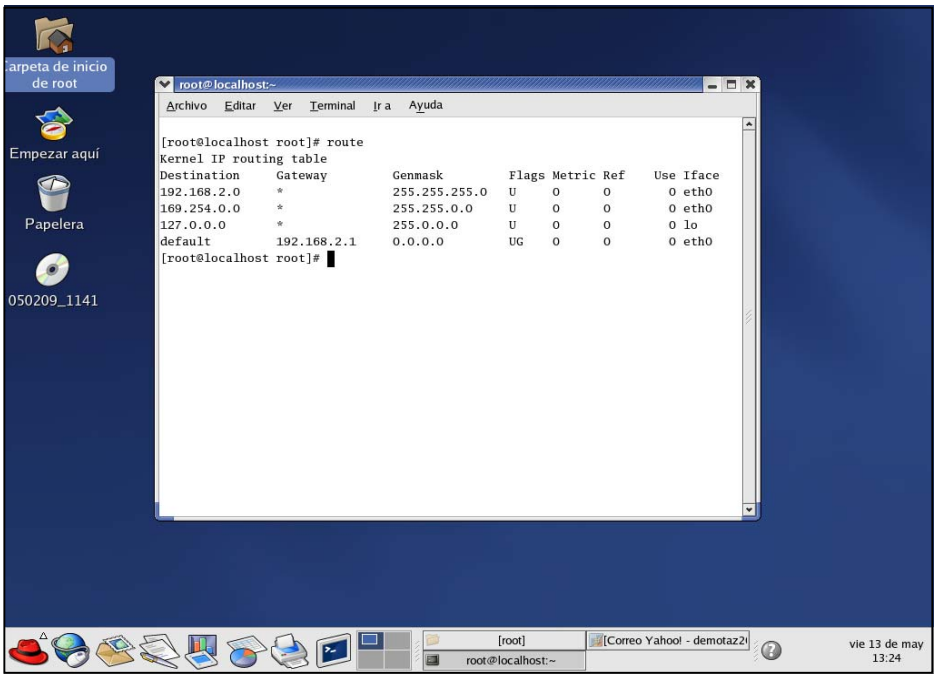


Figura No.7

Anote textualmente lo obtenido en pantalla y explíquelo.

_____
_____
_____
_____
_____
_____
_____
_____
_____

4.2.10 Observe la ruta que sigue un datagrama hacia el servidor de google. (Ver figura No.8).Teclee lo siguiente:

[root@localhost root]$ traceroute www.google.com

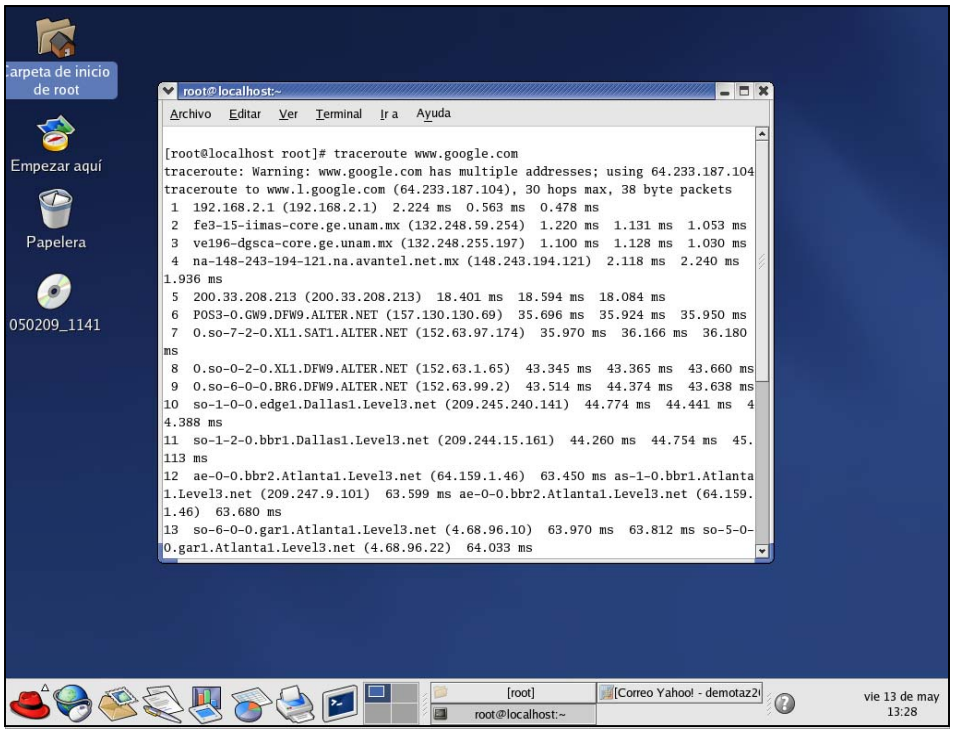


Figura No.8

Anote el resultado a continuación.

_____

_____

_____

_____

_____

_____

_____

_____

_____

4.2.11 Realice el paso anterior pero ahora hacia el servidor de altavista. (Ver figura No.9).

[root@localhost root]$ traceroute www.altavista.com

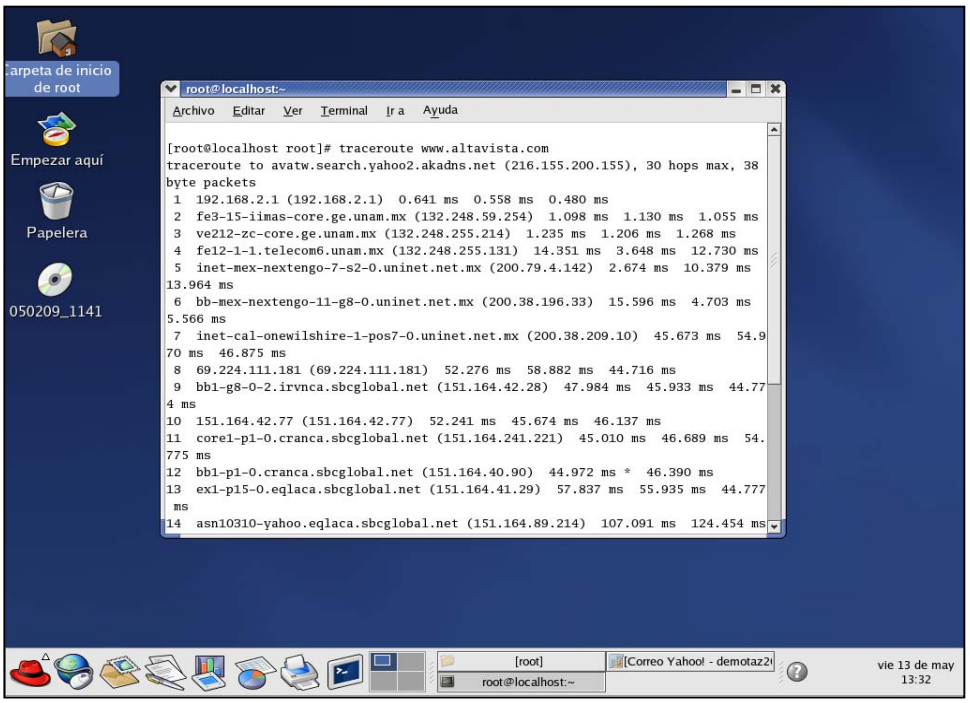


Figura No.9

Anote el resultado.

_____

_____

_____

_____

_____

_____

_____

_____

_____

4.2.12 Realice nuevamente el punto 4.2.10. Observe con cuidado si hay alguna diferencia con dicho punto. (Ver figura No.10).

[root@localhost root]$ traceroute www.google.com

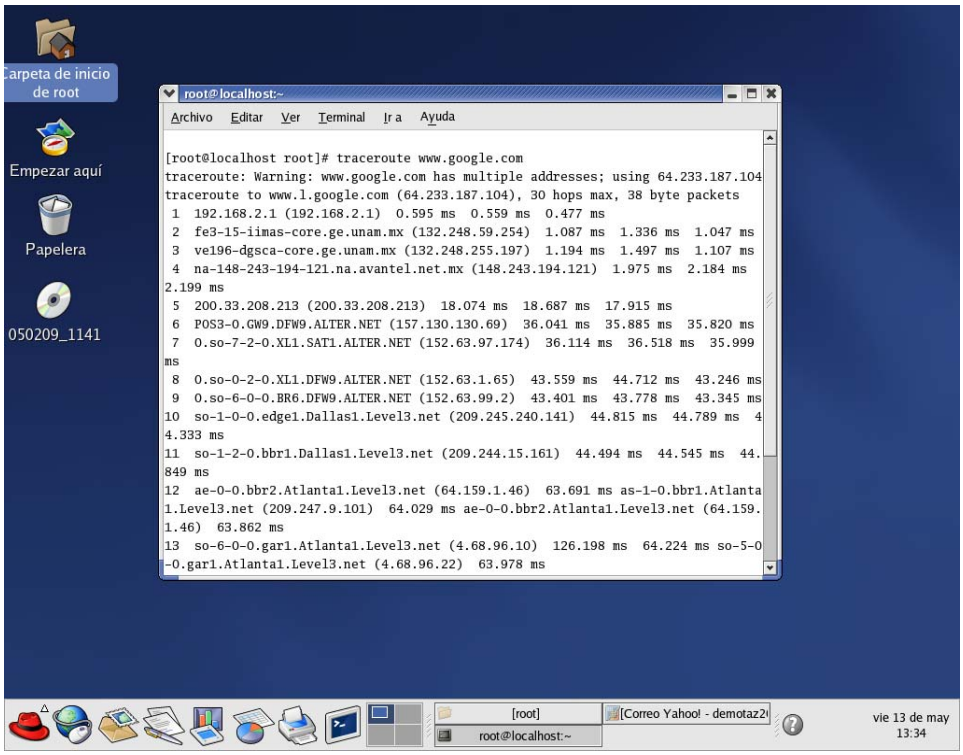


Figura No.10

Anote la salida en pantalla y coméntela.

_____

_____

_____

_____

_____

_____

_____

_____

_____

4.2.13 Salga de sesión y reinicie la computadora. (Ver figura No.11).



Figura No.11

4.2.14 Elija la opción de cargar Windows XP.

4.2.15 Entre como usuario redes, el profesor le proporcionará la contraseña.

4.2.16 De clic en Inicio, seleccione Todos los programas y de clic sobre el Símbolo del sistema. (Ver figura No.12).

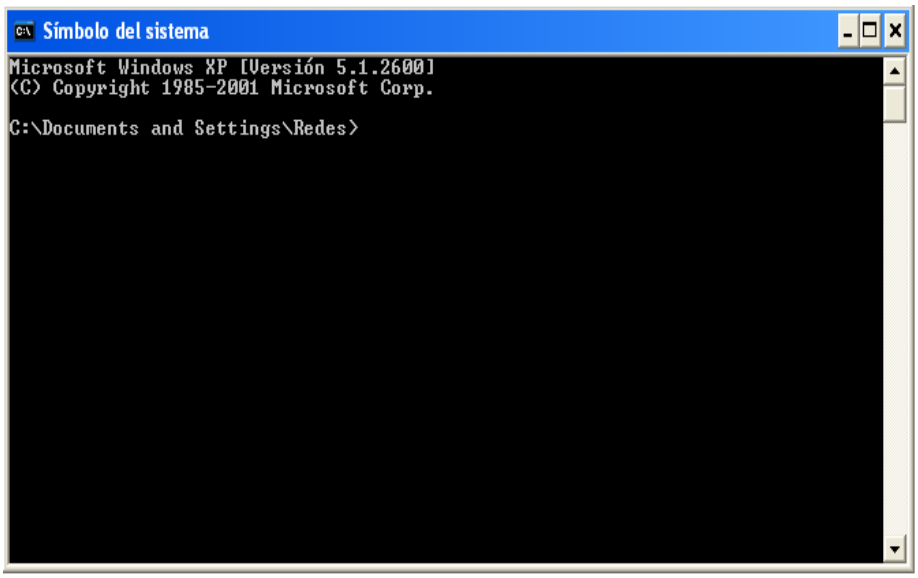


Figura No.12

4.2.17 Empleando el comando route, visualice la configuración actual de las tablas de ruteo. Teclee route print. (Ver figura No.13).

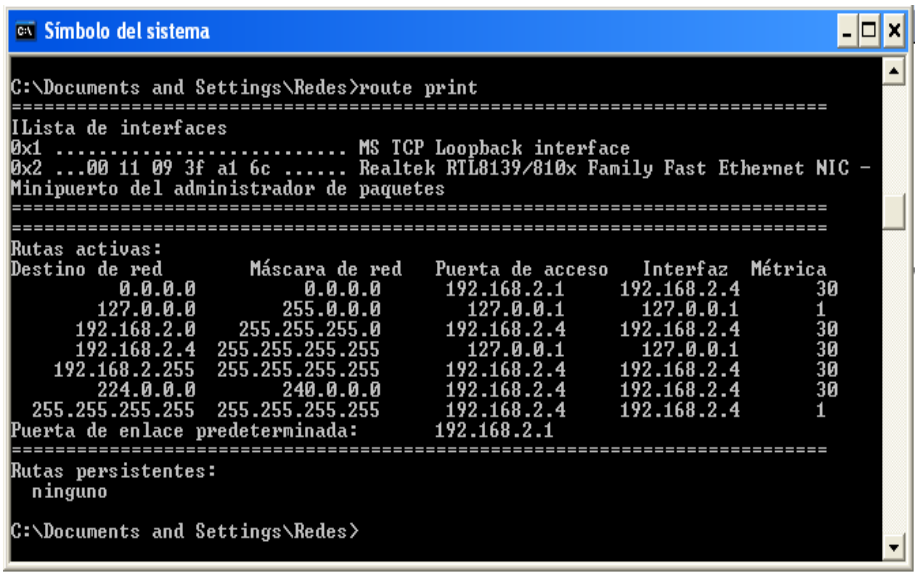


Figura No.13

Compárelo con lo obtenido en el punto 4.2.9 y concluya.

_____
_____
_____
_____
_____
_____

4.2.18 Observe el encaminamiento que sigue un paquete hacia el servidor de hotmail. (Ver figura No.14). Teclee tracert www.mineria.unam.mx

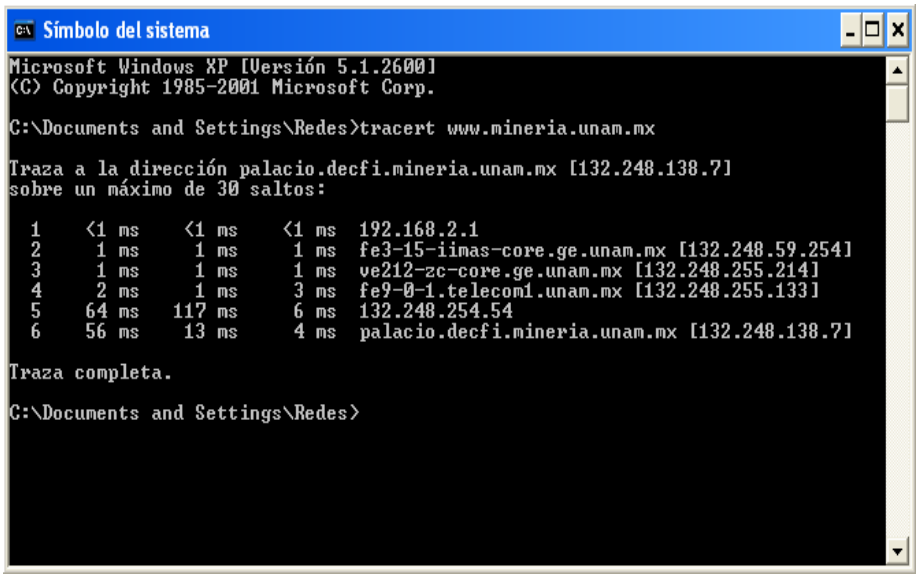


Figura No.14

Anote el resultado a continuación.

4.2.19 Cierre todas las aplicaciones y apague la computadora.

5.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica:

6.- Cuestionario

1. ¿En qué situaciones utilizaría los comandos vistos en clase?

_____
_____
_____
_____

2. ¿Cuál es la real importancia del comando *traceroute*?

_____
_____
_____
_____

3. ¿Qué sucedería si un paquete fuese enviado a un nodo inexistente sin TTL?

_____
_____
_____
_____

4. ¿Porque en Internet no necesariamente el recorrido de ida de un datagrama es igual al de regreso?

_____
_____
_____
_____

5. ¿Porque la cantidad de saltos de realiza un datagrama puede darnos una idea de la complejidad de la red?

_____
_____
_____
_____

PRÁCTICA 5
Enrutamiento de paquetes
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. ¿Cuál es el formato de trama Ethernet?

2. ¿Cuáles son los principales campos que forman la trama Ethernet?

3. Investigue cuales son los diferentes parámetros del comando *tcpdump*.

4. Mencione y explique tres algoritmos de enrutamiento estático.

5. Mencione y explique tres algoritmos de enrutamiento adoptivo.

6. Investigue cuales son los diferentes parámetros del comando *route*.

7. ¿Si no se especifica ningún parámetro, que muestra el comando *route*?

8. ¿Qué herramienta podemos utilizar para proteger la integridad de nuestro equipo?

PRÁCTICA 6
Análisis de Paquetes

1.- Objetivos de Aprendizaje
El alumno:

- Conocerá y comprenderá los fundamentos del monitoreo de redes, encapsulado de unidades a diferentes niveles y demultiplexación de las mismas.
- Adquirirá la habilidad necesaria para realizar los filtros adecuados.
- Reafirmará conocimientos teóricos acerca del protocolo ARP mediante observación de casos reales.

2.- Conceptos teóricos

El monitoreo de redes resulta una herramienta fundamental en dos sentidos. Por un lado, permite apreciar de forma realista muchos de los conceptos fundamentales de las redes en general, y de los protocolos TCP/IP en particular (encapsulación, fragmentación, secuenciación de mensajes, etc). Por otro lado, permite realizar un diagnóstico muy preciso de las redes en funcionamiento, desde la detección de errores, la verificación de los mecanismos de seguridad y la evaluación de prestaciones de la red.

Es por ello que en esta práctica se estudiará una herramienta gratuita de monitoreo de redes, denominada *Analyzer*, que trabaja sobre un interfaz de red denominado *WinPCap*.

El monitoreo de red, o captura de tramas, consiste en la obtención directa de tramas tal y como aparecen a nivel de LAN. Puesto que el medio de transmisión es, generalmente, una línea de difusión, el monitoreo permite observar la totalidad de las comunicaciones que tienen lugar a través de la red, y por tanto resulta una herramienta muy potente, tanto desde el punto de vista positivo (diagnóstico de red) como el negativo (compromete la confidencialidad de las comunicaciones).

La cantidad de información obtenida de un monitoreo es enorme. Por tanto, es necesario establecer filtros de aceptación, que permiten que las tramas no consideradas relevantes no se almacenen ni muestren al usuario.

El paquete Analyzer

Analyzer es una aplicación completamente configurable para el análisis mediante monitoreo de redes locales en entornos TCP/IP sobre cualquiera de las tecnologías soportadas por la interfaz *WinPCap*. Antes de instalar el Analizador de paquetes Analyzer; se deberá instalar y ejecutar la interfaz de red *WinPCap*.

Como puede apreciarse en la figura No. 1, tras la pantalla de presentación, la pantalla principal de la aplicación está encabezada por un menú, así como varias barras de iconos que resultan equivalentes a la misma.

Captura de tramas

Para comenzar la captura emplearemos la opción Capture Ø Start Capture, o bien el botón correspondiente en la barra de tareas, ilustrado en la imagen como una tarjeta con una flecha hacia arriba.

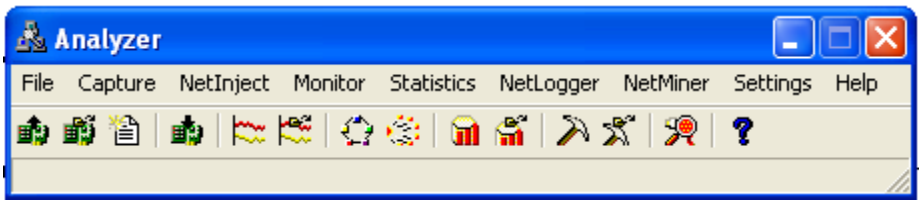


Figura No.1.

En ambos casos se abre la ventana de selección de filtros. En ella se encuentran varios parámetros importantes:

- *Select Adapter*: Permite, en caso de que nuestra computadora disponga de diversos interfaces de red, indicar sobre cuál de ellos se realizará la captura. Es destacable que en muchos casos se detecta una interfaz de red genérico NDIS que corresponde a un acceso alternativo a la tarjeta de red, o bien

las conexiones de acceso telefónico. En tal caso, es conveniente seleccionar la interfaz de más bajo nivel (por ejemplo, antes la tarjeta que la interfaz NDIS), y tener en cuenta que el acceso telefónico a redes no es totalmente compatible con *WinPCap.*(Ver figura No.2).

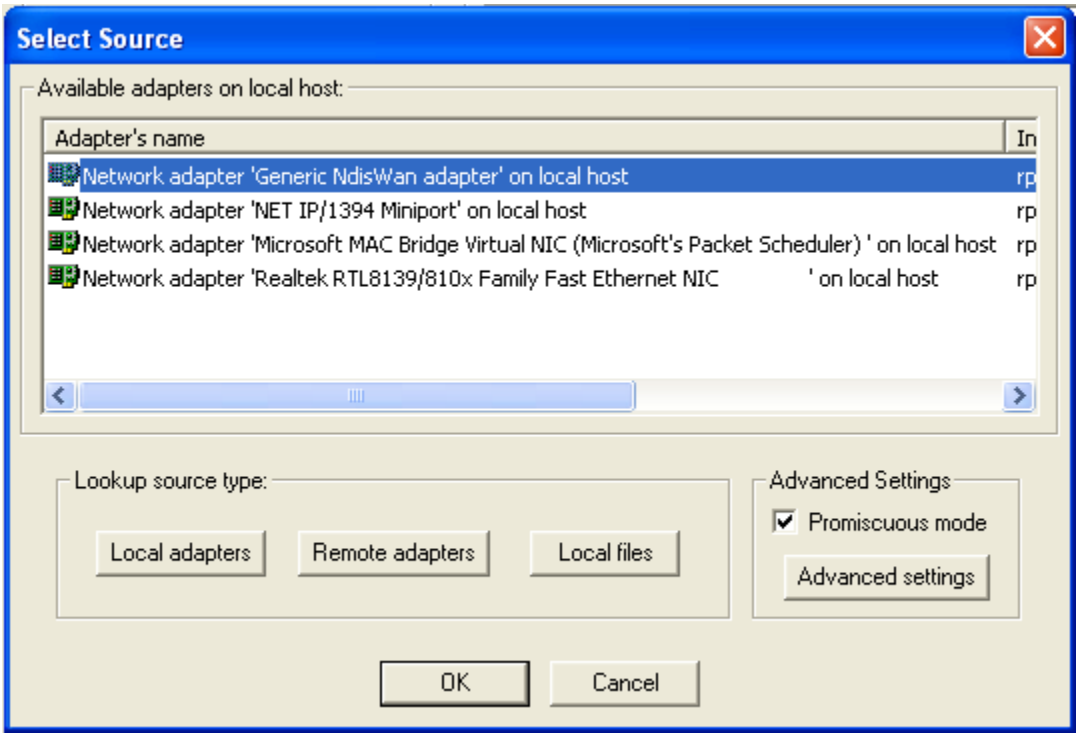


Figura No.2.

- *Promiscuos Mode*: En modo normal, las tarjetas de red únicamente almacenan aquellas tramas que han sido dirigidas explícitamente a su dirección MAC, aquellas dirigidas a grupos suscritos o difusiones. Para la monitorización entre dos máquinas desde una tercera es imprescindible recibir tramas que no cumplen ninguna de estas tres condiciones, y que por lo tanto serían ignoradas en condiciones normales. Mediante esta casilla se indica al adaptador de red que pase a modo promiscuo, en el cual debe recoger todas las tramas que aparezcan por la red sin considerar las condiciones anteriores.

- *Select Filter*: Permite seleccionar cuáles de las tramas recibidas por el adaptador deben ser almacenadas por el programa. En

nuestro caso, comenzaremos probando el protocolo ARP. (Ver figura No.3).

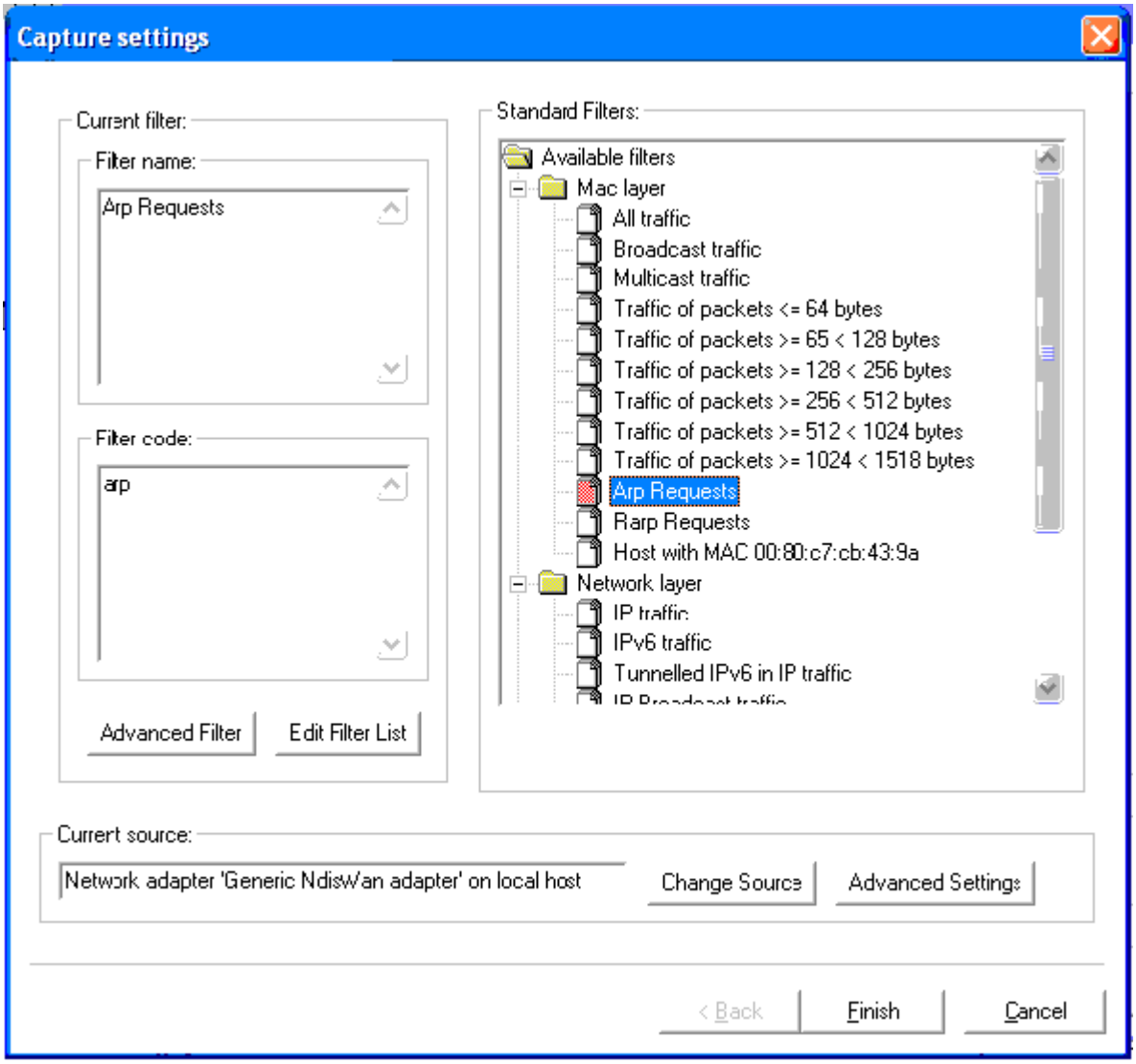


Figura No.3.

En este paso existen varias posibilidades. Una de ellas consiste en seleccionar alguno de los filtros predefinidos en el árbol de la derecha, que representan a las capturas más habituales. Por otro lado, también es posible especificar un filtro mediante un método similar al empleado por el programa *tcpdump*.

La visibilidad de tramas. En un medio de difusión, como por ejemplo ethernet sobre coaxial o con hubs, todas las tramas son

visibles por todas las estaciones. Sin embargo, si existen en la topología puentes, conmutadores o pasarelas, estos elementos actúan como divisores de dominios de colisión, y por tanto impiden la visibilidad de las tramas que no entran en el dominio de colisión del monitor.

El siguiente paso consiste en lanzar la captura de tramas, aplicando el filtro seleccionado, oprimiendo Finish en la interfaz Capture settings.

Cuando se considere realizada la captura, se detendrá mediante el botón *Stop* y la aplicación pasará a mostrar los paquetes capturados con el siguiente formato: (Ver figura No.4).

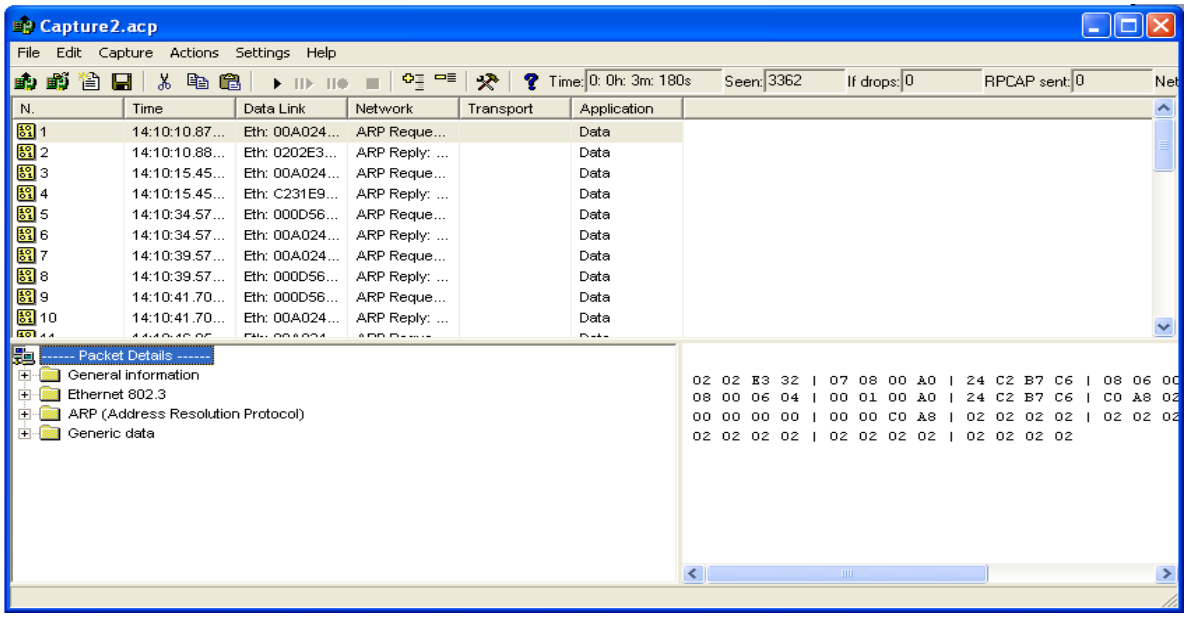


Figura No.4.

Cuando se selecciona en la lista de tramas capturadas en el recuadro inferior aparece un árbol con el detalle de los campos de cada nivel (información general, información de ethernet, etc.), y a la derecha una numeración hexadecimal y otra ASCII del campo de datos de la trama.

La orden arp

En una computadora que está conectada a una red de área local Ethernet que, a su vez, se conecta a Internet a través de un *router*, cuando las aplicaciones en red (Netscape, por ejemplo) generan peticiones para otras computadoras de Internet, crean paquetes (también llamados datagramas) que contienen la dirección IP de la máquina destino. El uso de direcciones IP (y de los protocolos TCP/IP) crea la ilusión de que todas las máquinas que se comunican pertenecen a una única (e inmensa) red común: Internet. Si la dirección IP destino corresponde a una máquina de esa misma subred (198.162.2.xx), el paquete puede ser entregado directamente a su destino sin más intermediarios. Sin embargo, cuando la dirección IP corresponde a una red o subred externa, la entrega de la información debe realizarse a través del *router*. En primer lugar, habrá que entregar la información al *router* de nuestra red y éste será el encargado de encaminar el paquete para hacerlo llegar a la red destino donde se encuentra la computadora referenciada. Como ve, en cualquiera de los dos casos, en una primera instancia se realiza una transmisión de información a través de la red de área local.

Desgraciadamente, las direcciones IP no son, por sí mismas, válidas para transmitir una trama a través de la red de área local. Las tarjetas adaptadoras de red que conectan las estaciones con el medio no entienden las direcciones IP, sólo entienden direcciones físicas. Por tanto, para que un datagrama IP viaje por la red de área local, este debe encapsularse dentro de una trama. Una trama Ethernet contiene la dirección física del siguiente destino que puede tratarse de la computadora final a la que van dirigidos los paquetes (origen y destino en la misma red local) o del *router* que encaminará el paquete hacia el exterior (origen y destino distintas redes o subredes).

En TCP/IP se utiliza un protocolo para la obtención de direcciones físicas a partir de direcciones IP dentro de una red de área local. Este protocolo se conoce con el nombre ARP (*Address Resolution Protocol - Protocolo de Resolución de Direcciones*). En esta práctica nos limitaremos a comprobar la existencia de este protocolo a través de la orden arp a nivel comando. Esta orden permite ver (y modificar) la caché ARP de su computadora. La caché ARP es

una tabla que almacena temporalmente las relaciones entre direcciones IP y direcciones físicas que ha conseguido averiguar su computadora utilizando el protocolo ARP. Es importante destacar, que la mayoría de estas entradas se generaran automáticamente (y de forma transparente al usuario) cuando se ejecuta una aplicación Internet (ping, cliente web, cliente ftp, etc.). Por tanto, muy rara vez (fuera de esta práctica) es necesario que el usuario modifique manualmente esta tabla.

Más concretamente, la orden arp a nivel comando permite:

- Ver la caché local de ARP (arp –a)
- Eliminar entradas manualmente de la caché (arp –d dirección_IP o arp –d *)
- Añadir entradas manualmente a la caché (arp –s dirección_IP dirección_Física)

3.- Equipo y material necesario

Equipo del Laboratorio:

A. 1 Computadora con Sistema Operativo Windows 98/NT/2000/XP*.*
B. La interfaz de red WinPCap.
C. Un analizador de paquetes (Analyzer). (Versión actual)

4.- Desarrollo:

4.1 Modo de trabajar

Para realizar la práctica se trabajará en equipos de dos personas por computadora.

4.2 Utilización de la aplicación Analyzer

Como primera aproximación a la utilización de la aplicación Analyzer para captura de paquetes y análisis de pro tocolos, a

continuación realizará una captura de paquetes IP y analizará los formatos de las tramas generadas que contengan datagramas IP.

4.2.1 Encienda el sistema y elija la opción de cargar *Windows*.

4.2.2 Haga doble clic sobre el acceso directo que se encuentra en el escritorio. (Ver figura No.5).



Figura No.5.

4.2.3 Inicie una captura (“Capture” Ø “Start Capture...”), o bien mediante el icono correspondiente. (Ver figura No.6).



Figura No.6.

4.2.4 Oprima el botón Change Source en la interfaz Capture settings, seleccione el adaptador de red 'Realtek RTL8139/810x Family Fast Ethernet NIC’ (El profesor indicará el tipo de adaptador de red propio del equipo. El adaptador mencionado es solo un ejemplo), desactive el modo promiscuo y oprima ok. (Ver figura No.7).

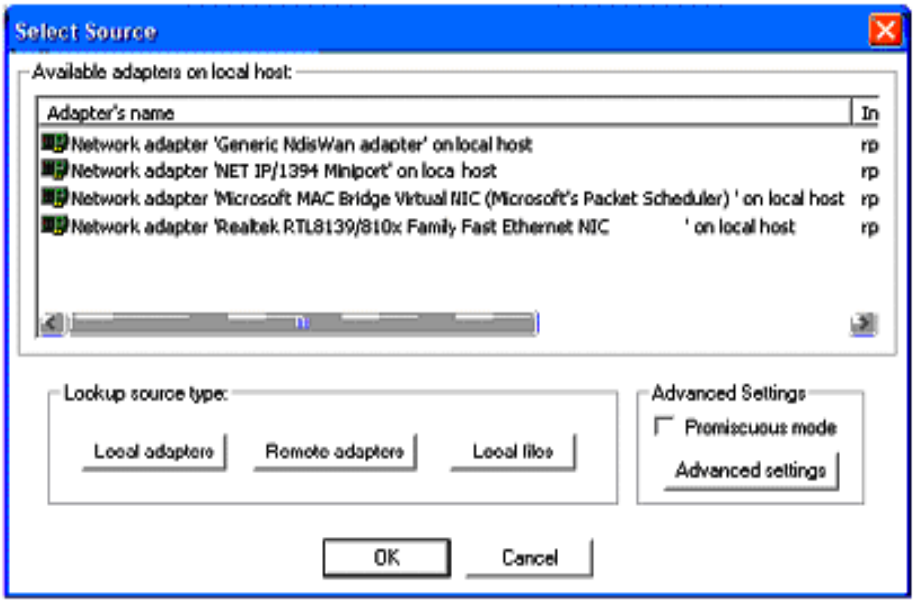


Figura No.7.

4.2.5 Defina un filtro que capture todas las tramas que contengan datagramas IP que entren o salgan de su computadora. Para realizar esta selección seleccione el protocolo IP del árbol de la derecha (dentro del apartado Network Layer) o bien escriba en el campo Filter code el filtro “ip”. No obstante, para evitar sobrecargar la captura con paquetes de difusión, afine más el filtro indicando “ip and not ip broadcast and not ip multicast”. (Ver figura No.8).

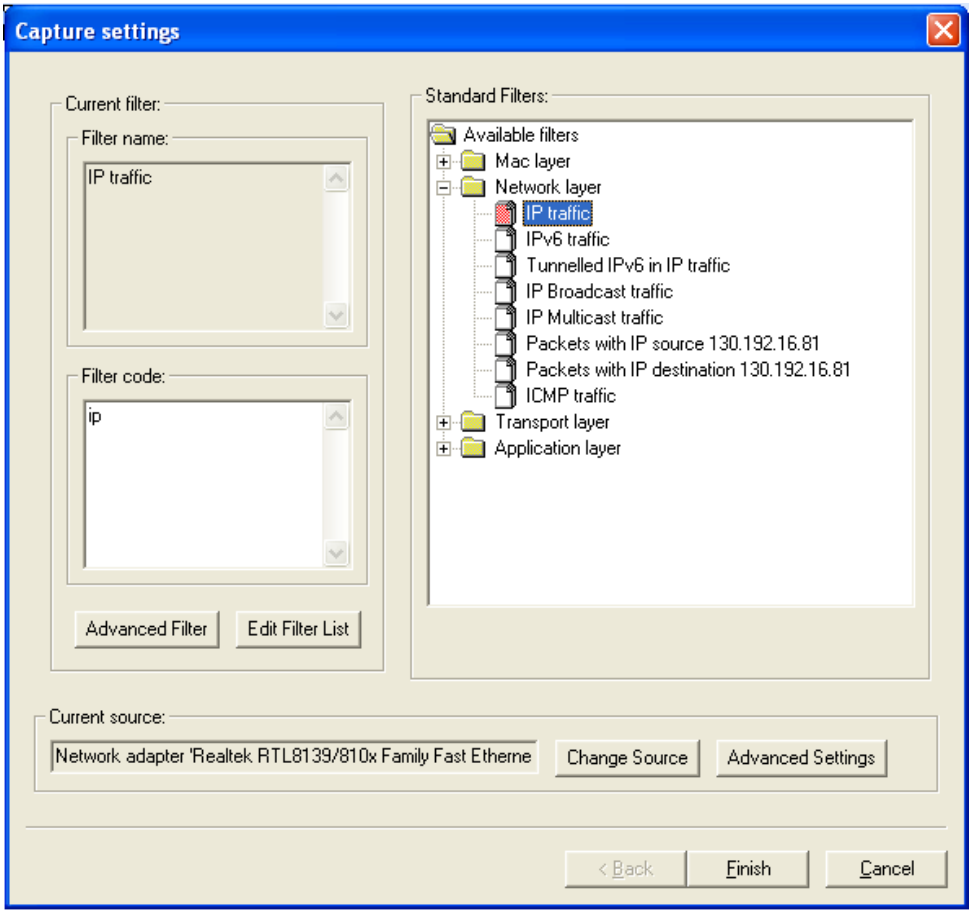


Figura No.8.

NOTA: Asegúrese de que el adaptador de red escogido es el correcto y que no está recibiendo en modo promiscuo.

4.2.6 Una vez definido el filtro, comience la captura de paquetes pulsando Finish.

4.2.7 Para generar algo de tráfico en su máquina, abra un cliente web (Netscape, Explorer) y acceda a la página *www.fi-b.unam.mx*. (Ver figura No.9).



Figura No.9.

4.2.8 Cuando haya finalizado la descarga de la página seleccionada, detenga la captura pulsando el botón stop en la gráfica obtenida. Obtendrá un resultado similar al de la figura 10:

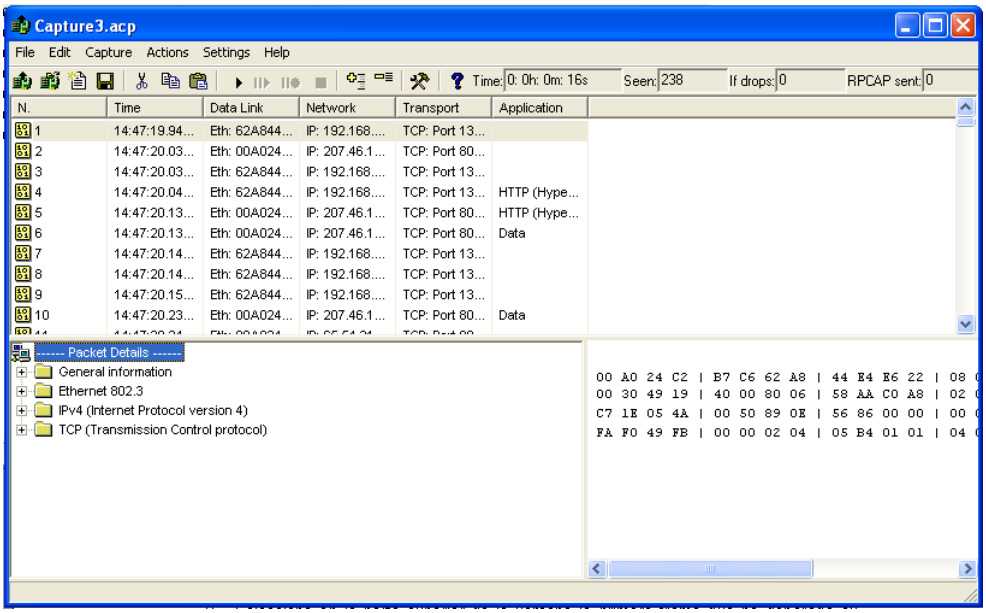


Figura No.10.

4.2.9 Seleccione en la parte superior de la ventana la primera trama que ha generado su computadora. (Ver figura No.11).

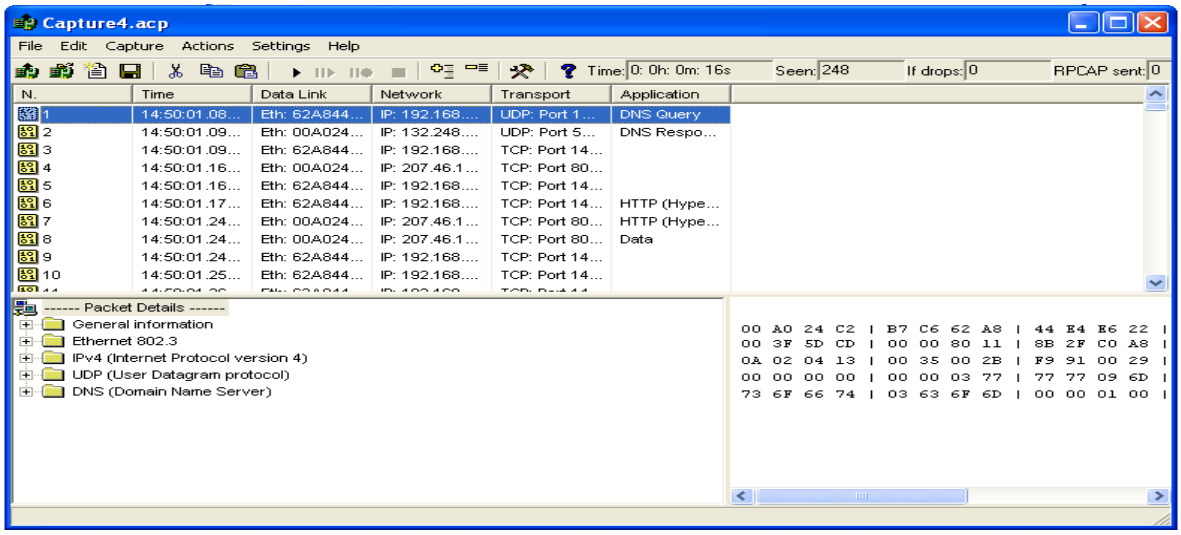


Figura No.11.

4.2.10 Analice los diferentes campos de la cabecera de la trama Ethernet y de la cabecera IP. Para ello abra las carpetas correspondientes en la sección Packet Details. (Ver figura No.12).

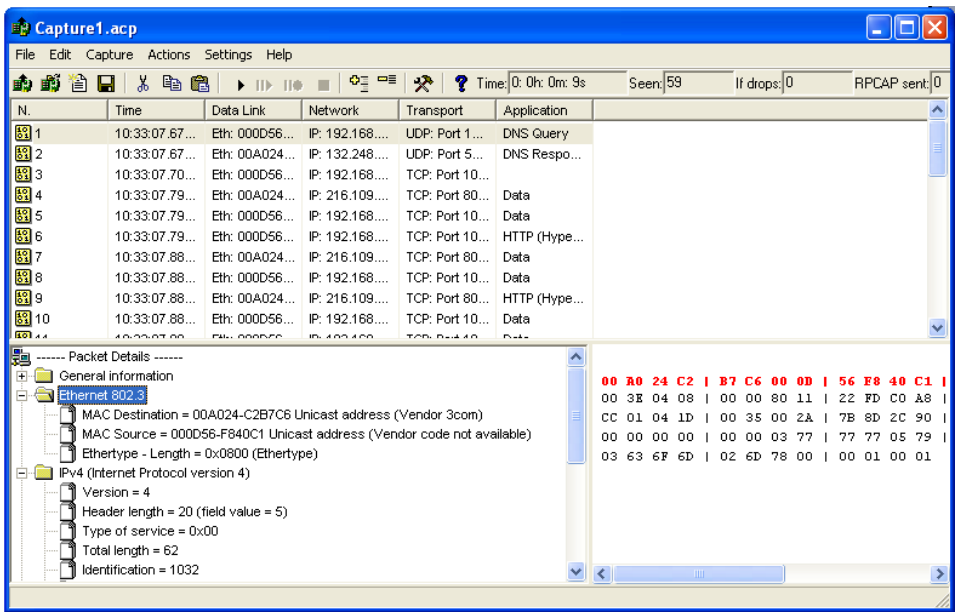


Figura No.12.

4.2.11 A partir de esta información llene las siguientes tablas: (Ver tabla No.1)

Cabecera de la trama Ethernet capturada:

Dirección física destino	Dirección física origen	tipo

Tabla No.1.

NOTA: Estos campos corresponden como necesarios para toda trama ethernet. El resto de los campos (CRC, preámbulo, etc.) son procesados y eliminados por el adaptador de red antes de entregar la trama al procesador.(Ver tabla No.2).

Cabecera del paquete IP capturado:

Versión	longc	tipo servicio	long total	
Identificación			flags	desplaz. fragmento
tiempo vida	protocolo		checksum de la cabecera	
dirección IP fuente				
dirección IP destino				
opciones (variable)				

Tabla No.2.

4.2.12 Desde una ventana de comandos ejecute la orden arp –a para comprobar que la caché ARP está vacía. (Ver figura No.13).

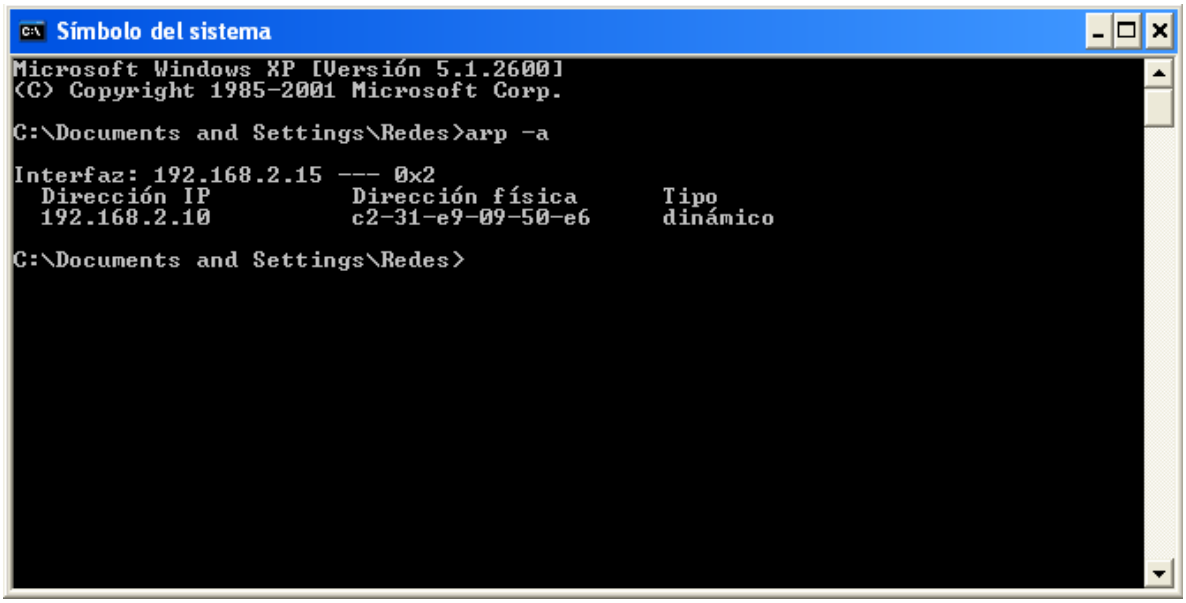


Figura No.13.

4.2.13 Si no lo está, cierre todas las aplicaciones que hagan uso de la red y elimine las entradas de la caché ARP usando la orden arp –d *. Si sale la siguiente pantalla simplemente espere un par de minutos (sin ejecutar nuevas aplicaciones en red) y las entradas desaparecerán de la caché. (Ver figura No.14).

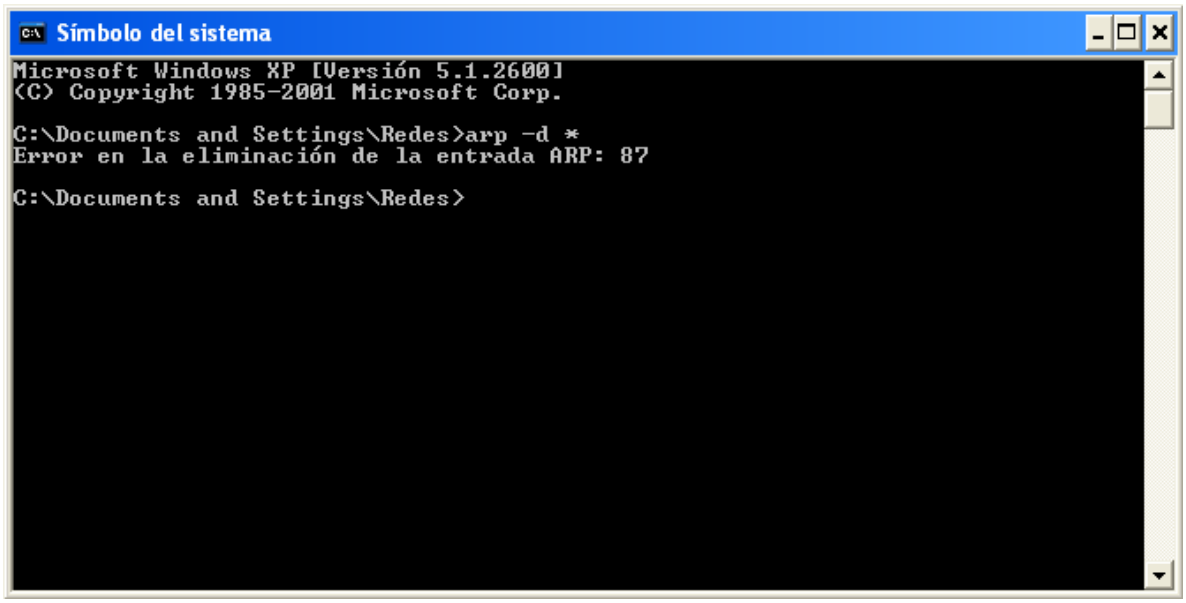


Figura No.14.

4.2.14 A continuación ejecute un navegador WEB para acceder a la máquina 132.248.59.6 (es la dirección de la máquina www.fi-b.unam.mx) mediante el URL http://132.248.59.6, espere a que se complete la descarga de la página y examine de nuevo la caché ARP. (Ver figura No.15).

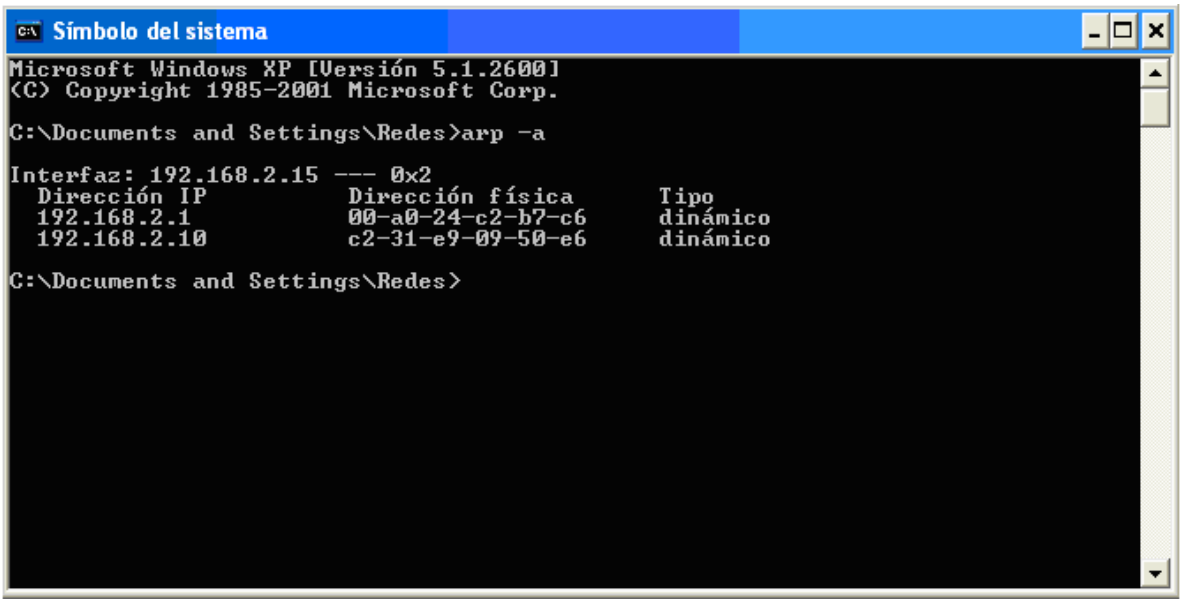


Figura No.15.

4.2.15 Anote la información obtenida en la tabla siguiente:(Ver tabla No.3).

Dirección IP	Dirección Física

Tabla No.3.

4.2.16 Cierre el navegador WEB, elimine manualmente las entradas de la caché ARP (o espere un par de minutos) y utilice el mismo navegador para acceder al servidor www.unam.mx. (Ver figura No.16).



Figura No.16.

4.2.17 Anote la información obtenida en la tabla siguiente:(Ver tabla No.4).

Dirección IP	Dirección Física

Tabla No.4.

¿A qué máquina corresponde la dirección almacenada en la caché?. Consulte la información ofrecida por la orden ipconfig /all para resolver este apartado. (Ver figura No.17).



Figura No.17.

¿Cree que esta dirección corresponde a la máquina www.unam.mx? ¿Por qué?

_____
_____
_____
_____
_____
_____

A continuación empleará el *Analyzer* para observar la secuencia de pasos de una petición ARP y los efectos en la caché ARP. Para ello necesitara ejecutar simultáneamente:

- Una ventana de comandos donde empleara el comando arp para ver la caché ARP y la orden ping para generar las peticiones correspondientes.
- El *Analyzer* para observar el detalle de las tramas que se transmiten y reciben.

4.2.18 En primer lugar, prepare la captura con el *Analyzer* de forma análoga a la empleada anteriormente: El objetivo consiste en la captura de todas las tramas ARP relativas a su computadora. No lance aún la captura. (Ver figura No.18).

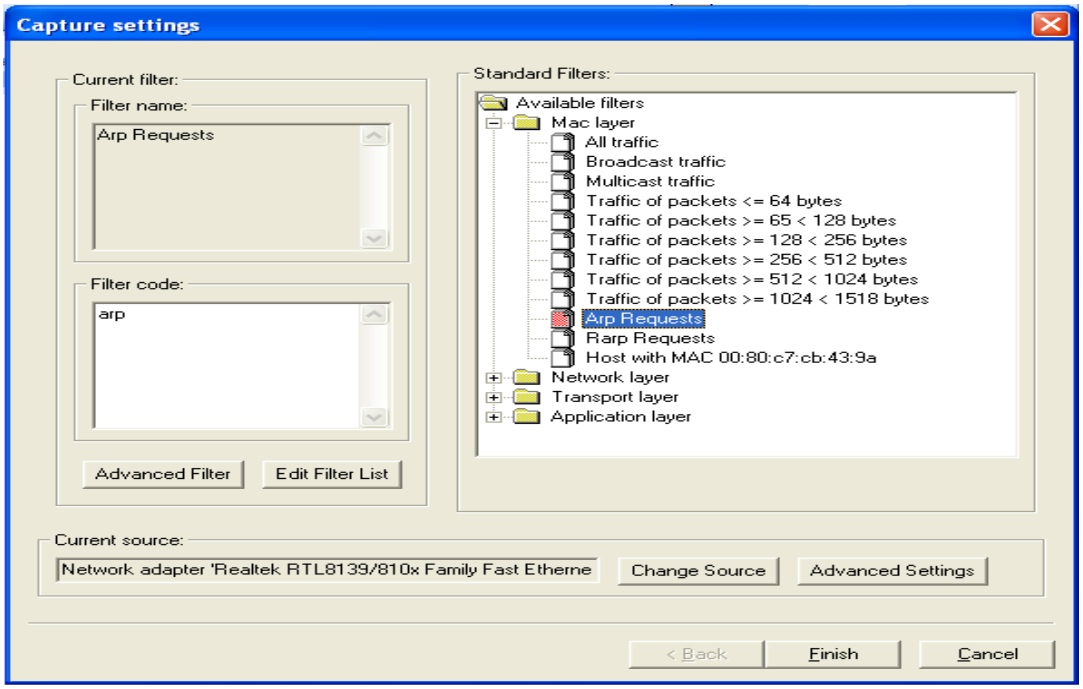


Figura No.18.

4.2.19 Una vez dispuesto el monitor, en la ventana de comandos vacíe la caché ARP mediante el comando arp –d * (o espere unos minutos). (Ver figura No.19).

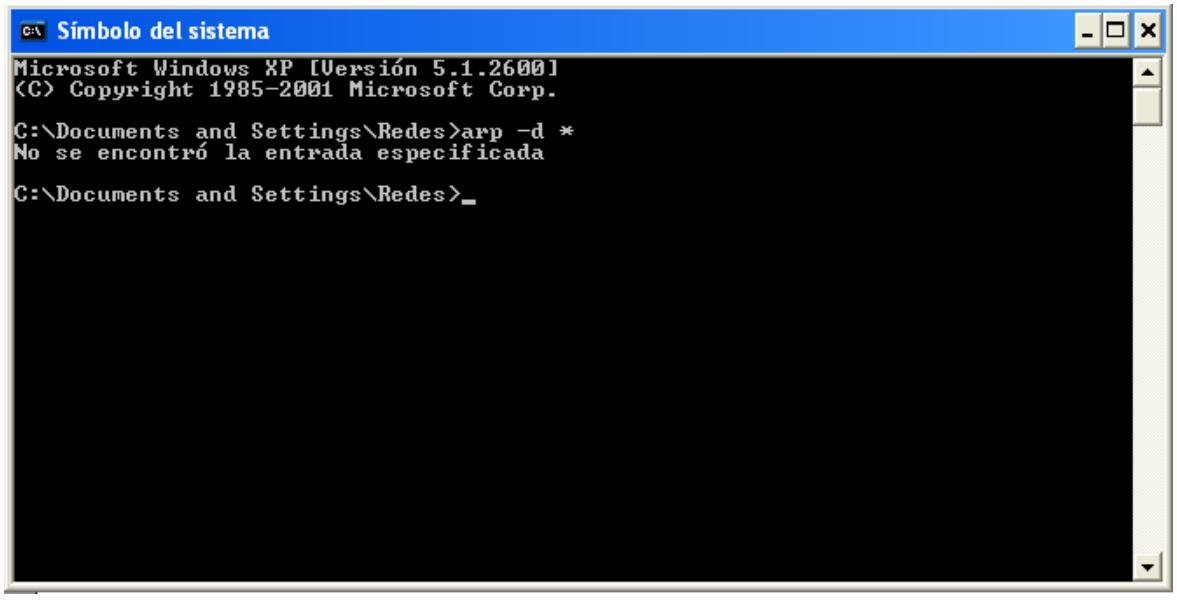


Figura No.19.

4.2.20 Tras ello, ponga en marcha la captura. (Ver figura No.20).

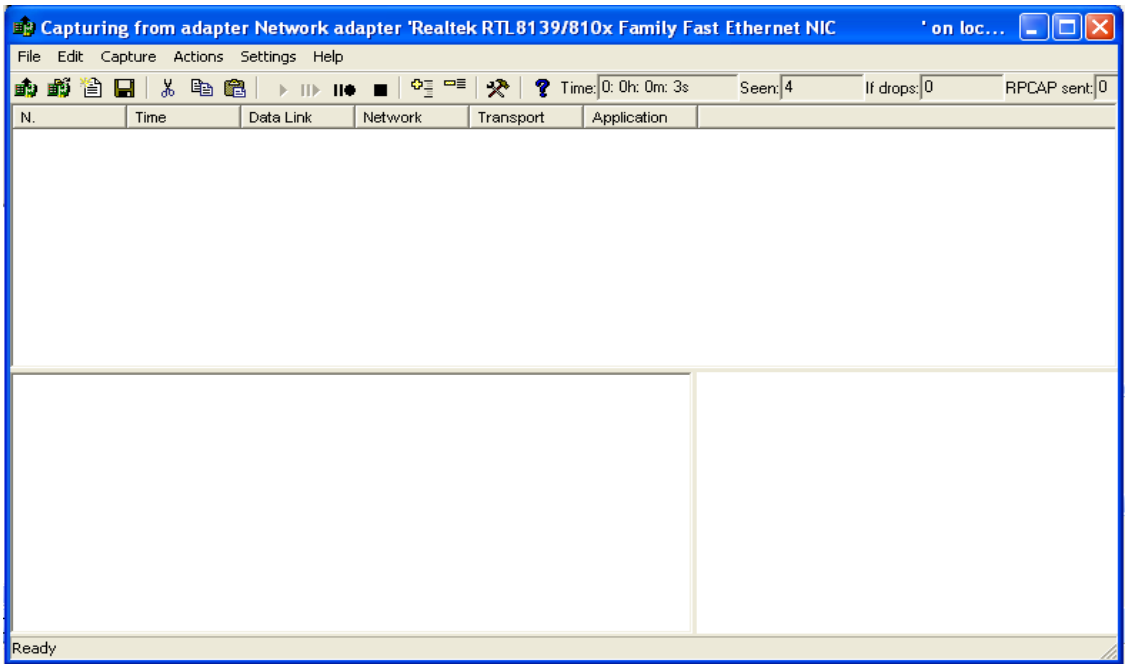


Figura No.20.

4.2.21 Vuelva a la ventana de comandos y ejecute la orden ping www.fi-b.unam.mx. Esto causa la transmisión y recepción de una serie de datagramas IP sobre la máquina destino. (Ver figura No.21).

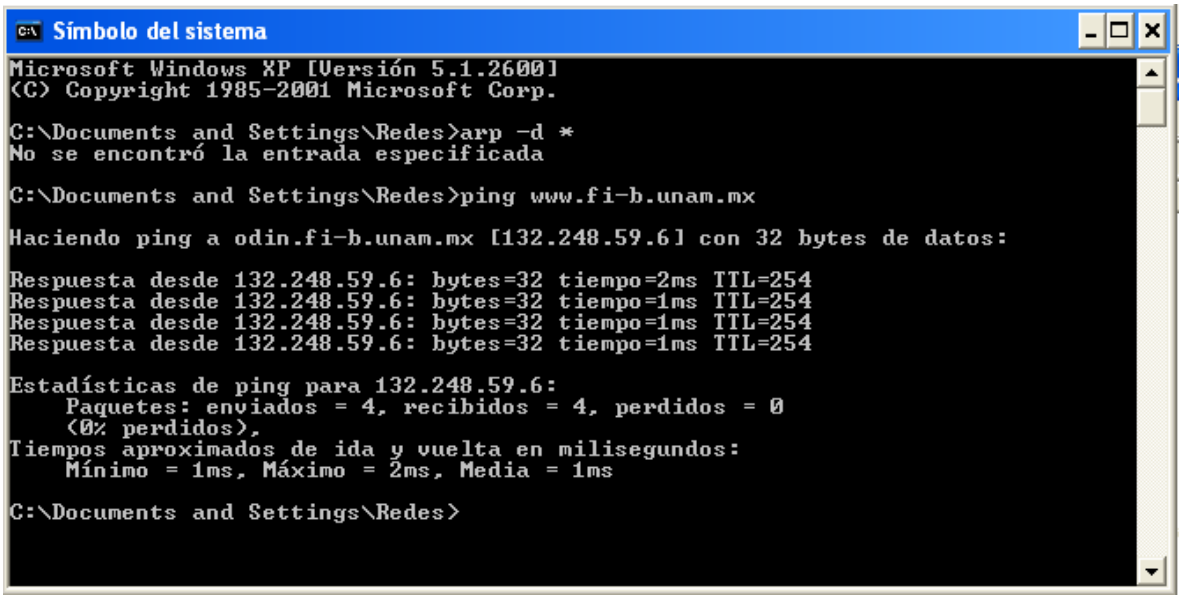


Figura No.21.

4.2.22 A continuación muestre la caché ARP con la orden arp –a. (Ver figura No.22).

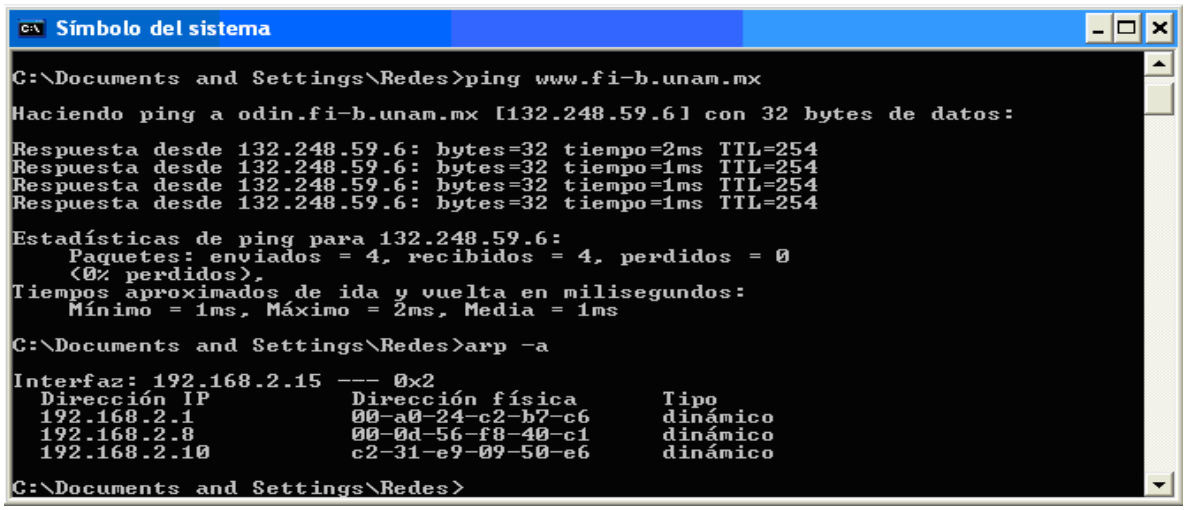


Figura No.22.

4.2.23 Detenga la captura en el *Analyzer*

4.2.24 Llene la siguiente tabla con el contenido de dicha caché. (Ver tabla No.5).

Nº	Dirección IP	Dirección Física
1		
2		
3		

Tabla No.5.

4.2.25 Vuelva al *Analyzer* y observe las tramas recibidas. (Ver figura No.23).

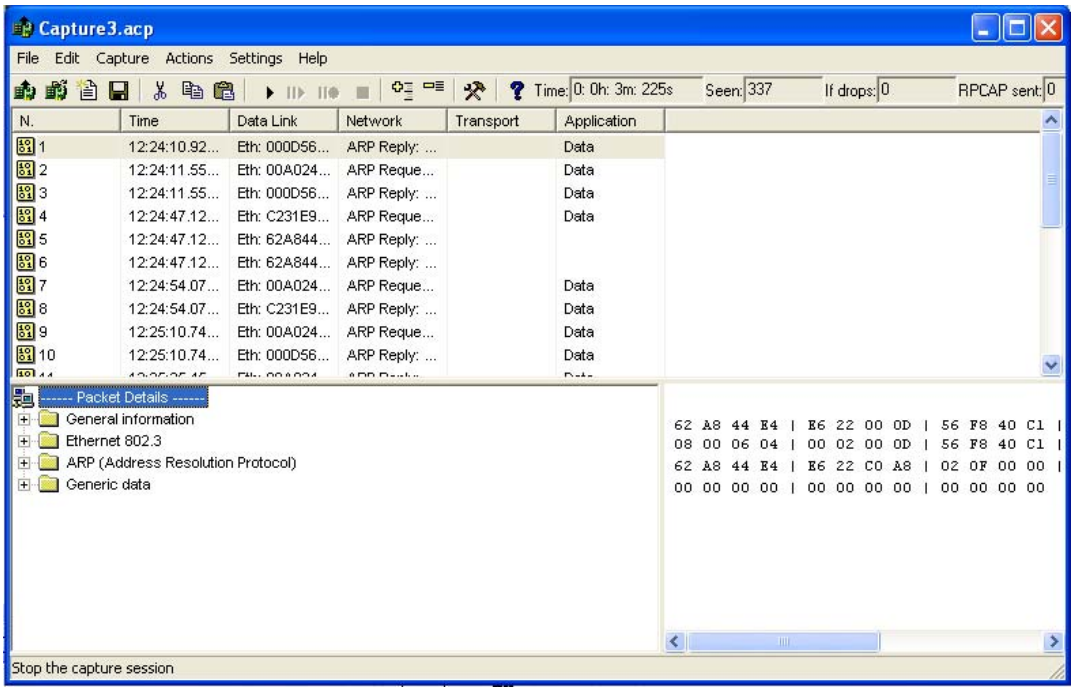


Figura No.23.

En un caso óptimo, la captura contendría únicamente dos tramas, un ARP REQUEST y su correspondiente ARP RESPONSE. No obstante, es habitual que también se hayan capturado ARP REQUEST que no están relacionados con el ping ejecutado. A efectos de este ejercicio, no se consideraran estas tramas. (Ver figura No.24).

4.2.26 Rellene la tabla siguiente con la información extraída del ARP REQUEST. (Ver tabla No.6).

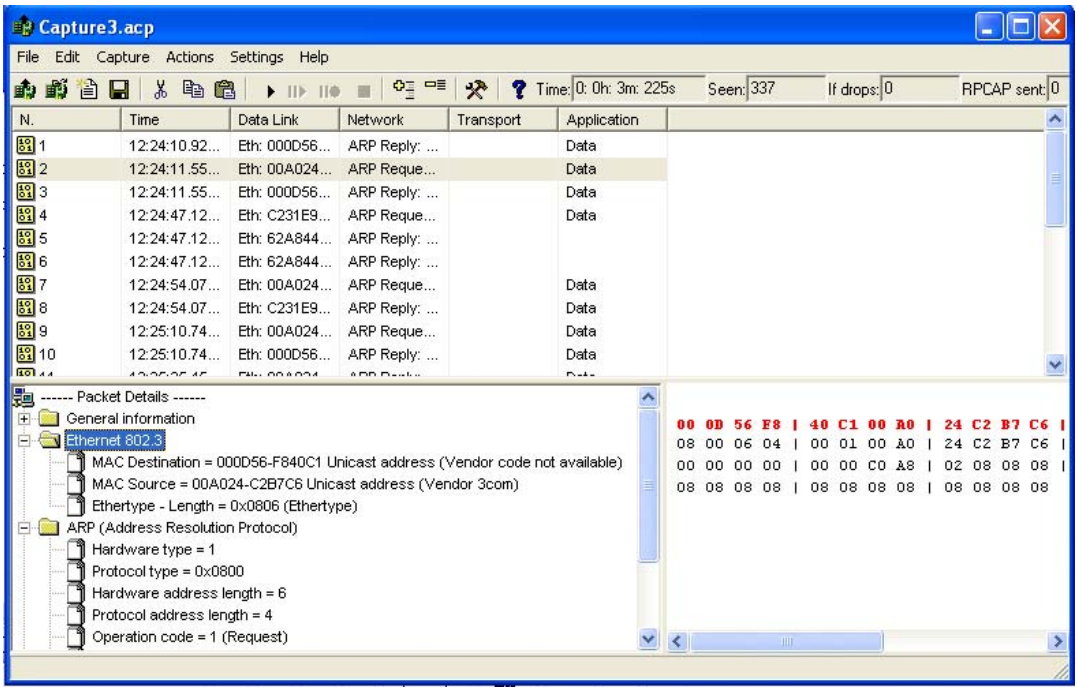


Figura No.24.

Cabecera de la trama:

Dirección física destino	Dirección física origen	tipo

Tabla No.6

Mensaje ARP REQUEST:

Tipo de hardware		Tipo de protocolo
Longitud dirección física	Longitud dirección protocolo	Operación
Dirección física fuente		
dirección IP fuente		
Dirección física 2		
dirección IP 2		

Tabla No.7

4.2.27 Rellene la tabla siguiente con la información acerca de la trama ARP RESPONSE. (Ver figura No.25). (Ver tabla No.8).

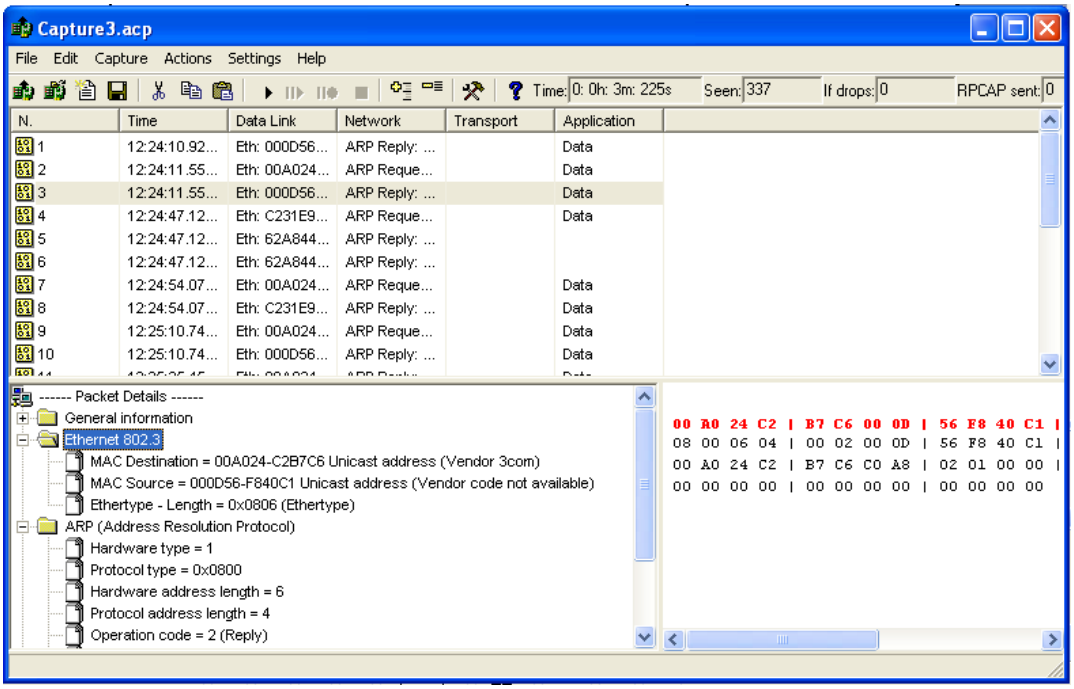


Figura No.25.

Cabecera de la trama:

Dirección física destino	Dirección física origen	tipo

Tabla No.8.

Mensaje ARP RESPONSE:

Tipo de hardware		Tipo de protocolo
Longitud dirección física	Longitud dirección protocolo	Operación
Dirección física fuente		
dirección IP fuente		
Dirección física 2		
dirección IP 2		

Tabla No.9.

¿Qué entrada de la caché ARP observada en el paso 4.2.24 corresponde con estas tramas?

_____

_____

Si existen otras entradas en la caché ¿A qué tramas de las capturadas corresponden?

De todas estas entradas en la caché ¿Cuál crees que se eliminará antes? ¿Por qué?

4.2.28 Monitoree cantidad de paquetes ARP que recibe su PC. Para ello inicie el *Analyzer* y seleccione la opción *Monitor Ø Start graphical Monitor*. Elija el filtro arp y oprima Next. (Ver figura No.26).

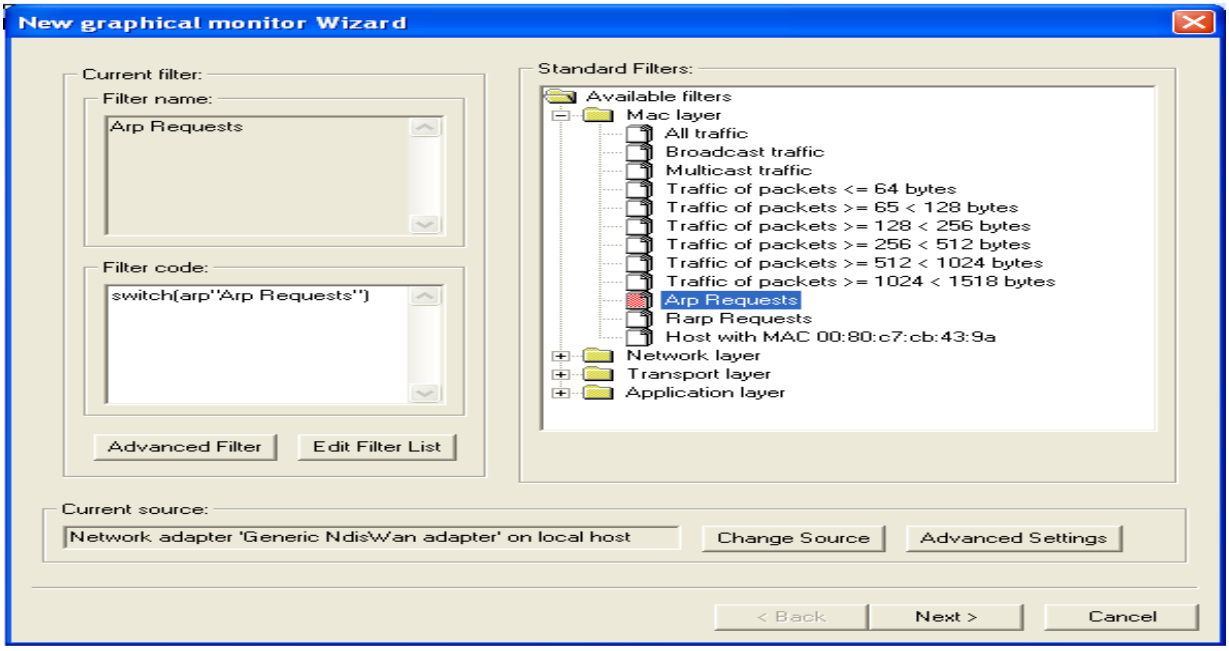


Figura No.26.

4.2.29 En el ajuste de las características del monitoreo seleccione la opción Packets/s en el campo Representation, dejando el resto de los parámetros con los valores por omisión, oprima Next. (Ver figura No.27).

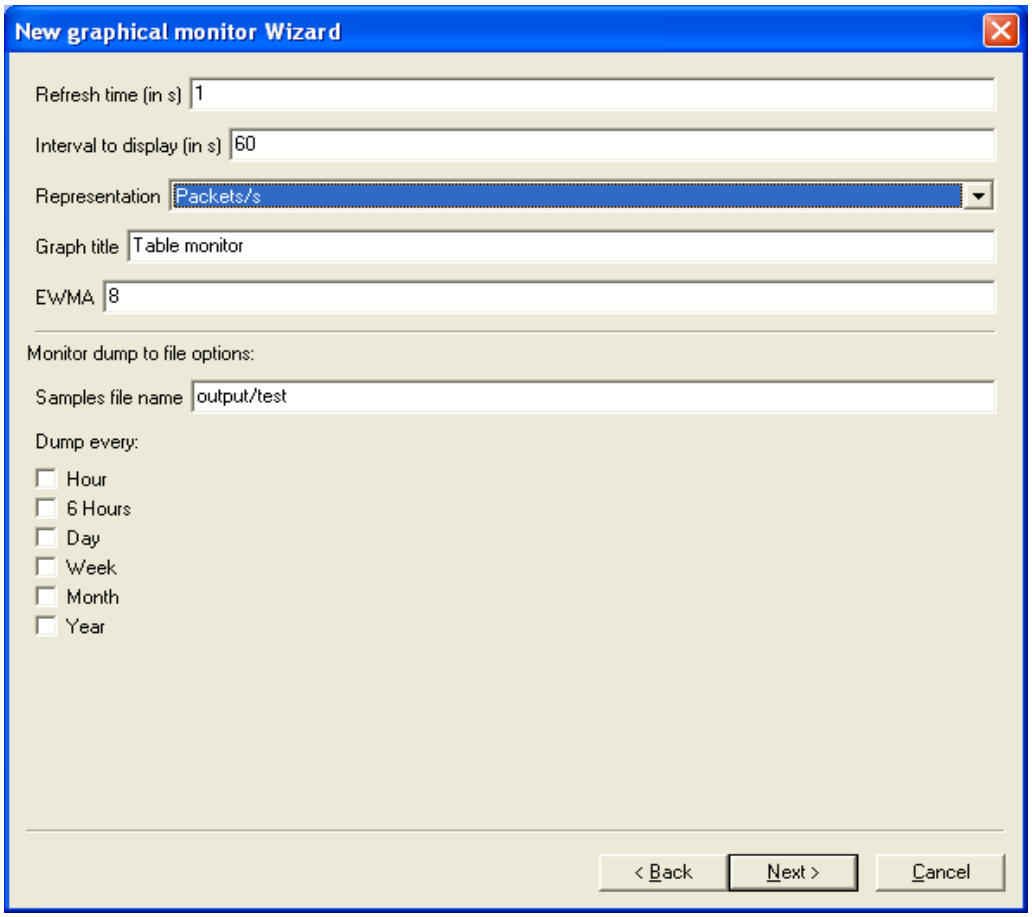


Figura No.27.

4.2.30 A continuación se presentan cuatro tipos de visualización: Displays data as vertical bars, Displays data as a series of polylines, Displays data a pie chart o Displays data a simple table. Seleccione líneas y proceda a la captura. (Ver figura No.28).

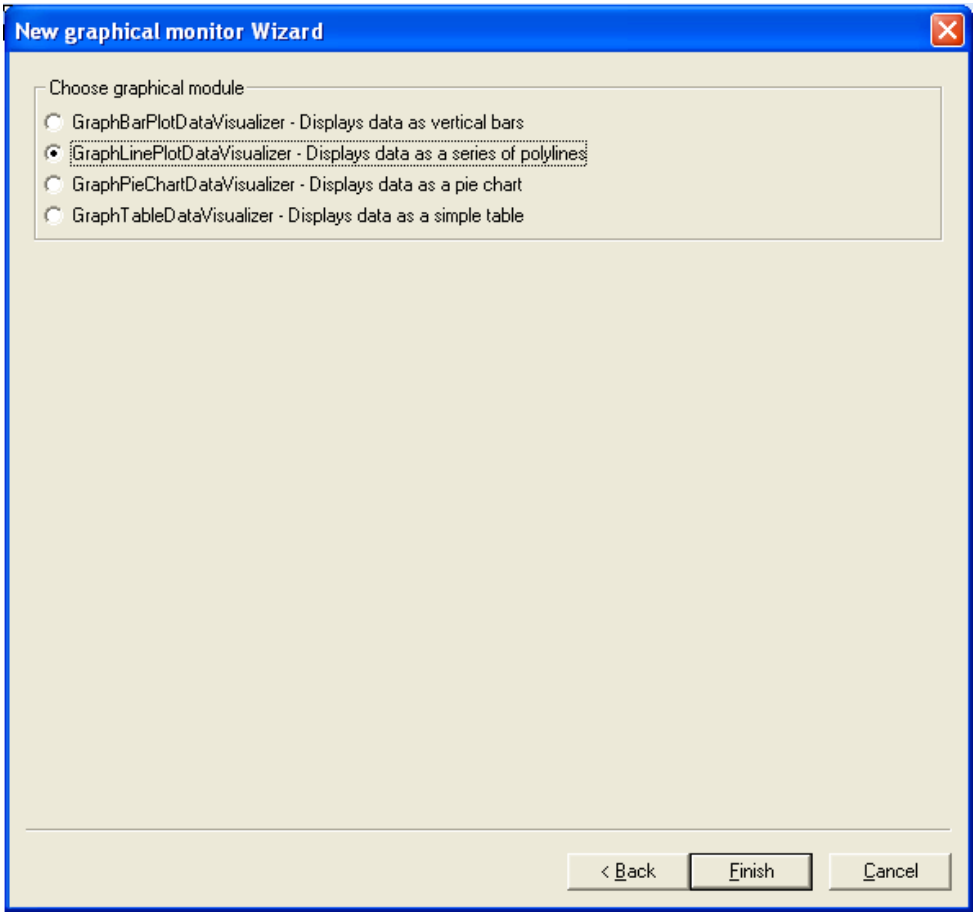


Figura No.28.

4.2.31 Observe el monitor en marcha durante algunos minutos, y comente lo visto en pantalla. (Ver figura No.29).

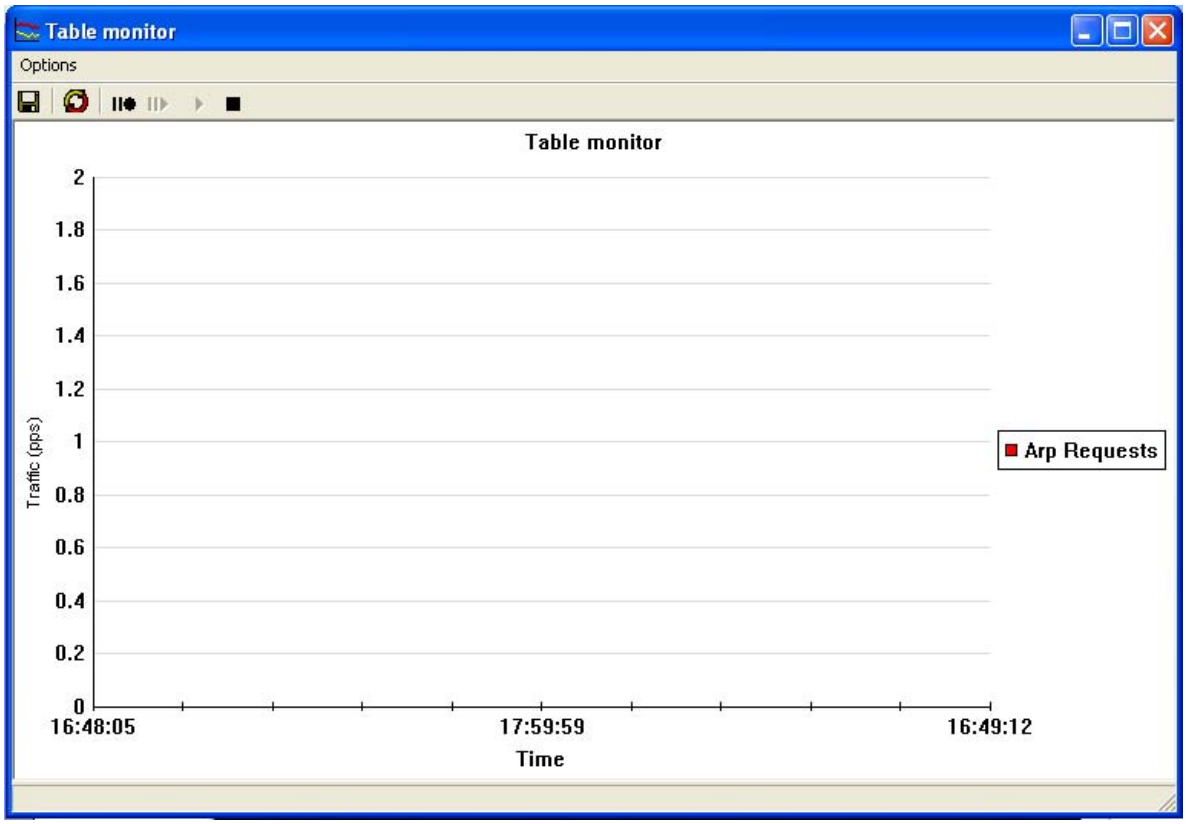


Figura No.29.

5.-Cuestionario

1. ¿Qué es la encapsulación de paquetes?

2. ¿Qué es la fragmentación de paquetes?

3. ¿Qué es una trama?

6.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica:

PRÁCTICA 6
Análisis de Paquetes
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. ¿Qué es la secuenciación de mensajes?

2. Dibuje la cabecera de una trama Ethernet

3. ¿Qué es una línea de difusión?

4. ¿A que se refiere una interfaz de red NDIS?

5. Investigue la diferentes opciones del comando ARP a nivel de comandos

PRÁCTICA 7
TCP y UDP

1.- Objetivos de Aprendizaje
El alumno:

- Podrá utilizar un programa que le permita enviar y recibir información utilizando los protocolos TCP y UDP y reafirmando conceptos teóricos.
- Podrá solicitar una página web a un determinado servidor desde un navegador.
- Tendrá la capacidad de conectarse al servidor SSH local haciéndole una petición de enviar la versión de este.
- Será capaz de crear un socket servidor y un socket cliente.

2.- Conceptos teóricos

El programa sock

El programa sock ofrece un modo de acceder a la interfaz de los sockets sin tener que programar. El programa conecta la entrada/salida estándar (teclado/pantalla) con un socket cuyas características se especifican mediante parámetros al ejecutar la orden. Mediante la redirección de la entrada o la salida se puede enviar el contenido de un archivo o almacenar en un archivo la información recibida.

Los sockets pueden ser de dos tipos: UDP o TCP, que se corresponden con un servicio sin conexión, que no garantiza ni la entrega ni el orden de entrega de la información (UDP) y otro servicio que garantiza la entrega ordenada y sin errores de la información (TCP).

Además, se sabe que una aplicación puede comenzar iniciando la comunicación (enviando información) o bien puede esperar pacientemente hasta que la otra le solicite el inicio de la comunicación (espera petición).

El programa sock nos va a permitir imitar cualquiera de estas situaciones y alguna otra cosa más. Vamos a ir viendo, mediante una serie de ejercicios sus posibilidades.

3.- Equipo y material necesario

Material que deberá traer el alumno:

A. Lápiz ó pluma.
B. 1 diskette de 3 ½ (con una imagen .png por lo menos)

Equipo del Laboratorio:

C. 1 Computadora con Sistema Operativo *Linux Red Hat 9.0 ó Superior.*
D. Un programa sock (sock-1.1.tar.tar).

4.- Desarrollo

4.1 Modo de trabajar

Se trabajará en equipos de dos personas por computadora.

4.2 Realización de la práctica

4.2.1 Encienda el sistema y elija la opción de cargar *Linux.*

4.2.2 Entre a sesión como usuario normal. La cuenta es redes y la contraseña la proporcionará el profesor. (Ver figura No.1).



Figura No.1.

4.2.3 Abra un shell prompt seleccionando:
Main Menu => System Tools => Terminal.

Para salir del shell prompt teclee exit o presione <ctrl>d en el prompt. (Ver figura No.2).

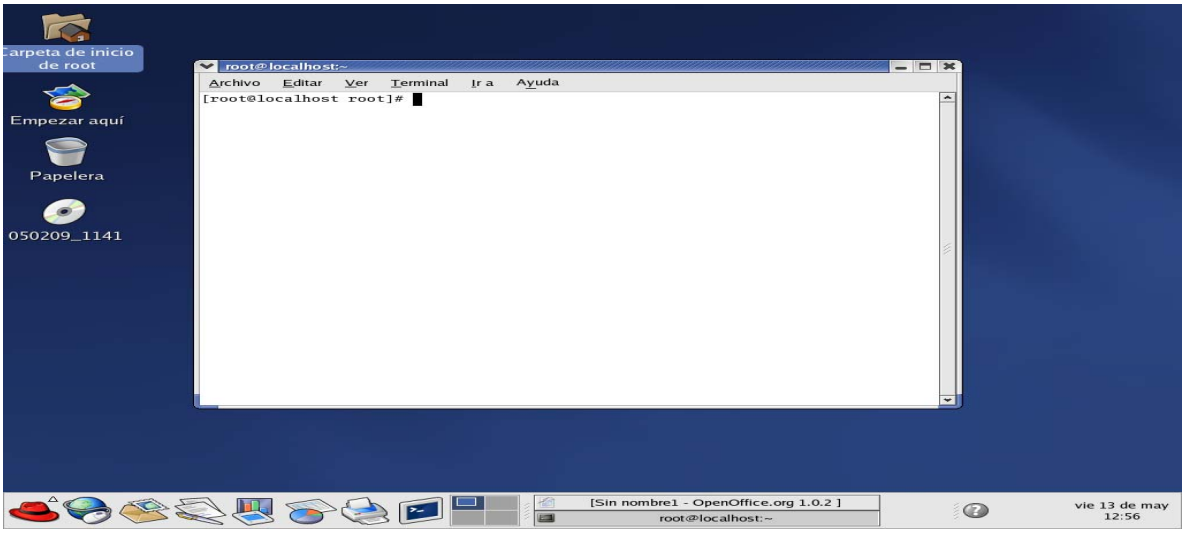


Figura No.2.

4.2.4 Ahora podemos emplear el sistema para ejecutar los comandos que necesitemos utilizar.

4.2.5 Cree el subdirectorio práctica dentro del directorio actual. (Ver figura No.3).

[redes@localhost redes]$ mkdir practica

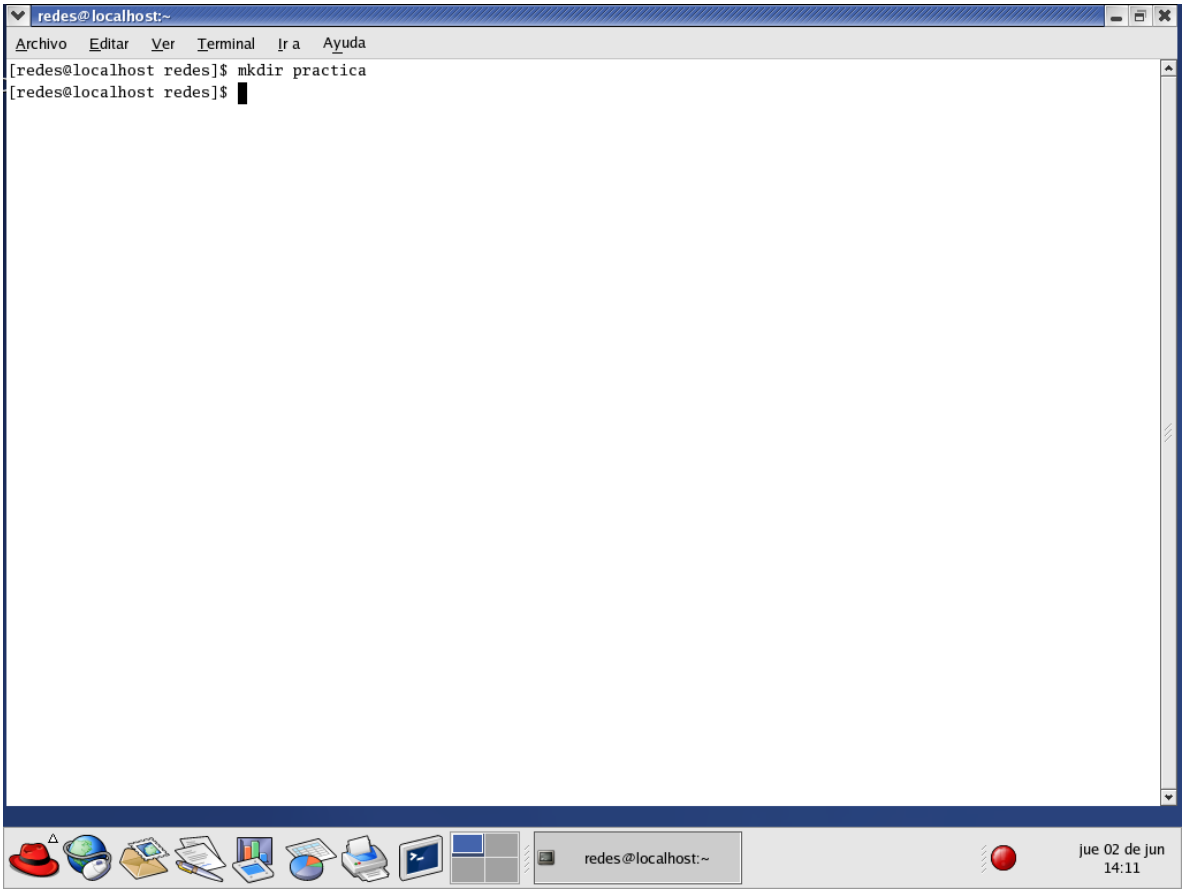


Figura No.3.

4.2.6 Copie el archivo sock-1.1.tar.tar dentro del subdirectorio practica. (Ver figura No.4).

[redes@localhostredes]$ cp sock-1.1.tar.tar /home/redes/practica

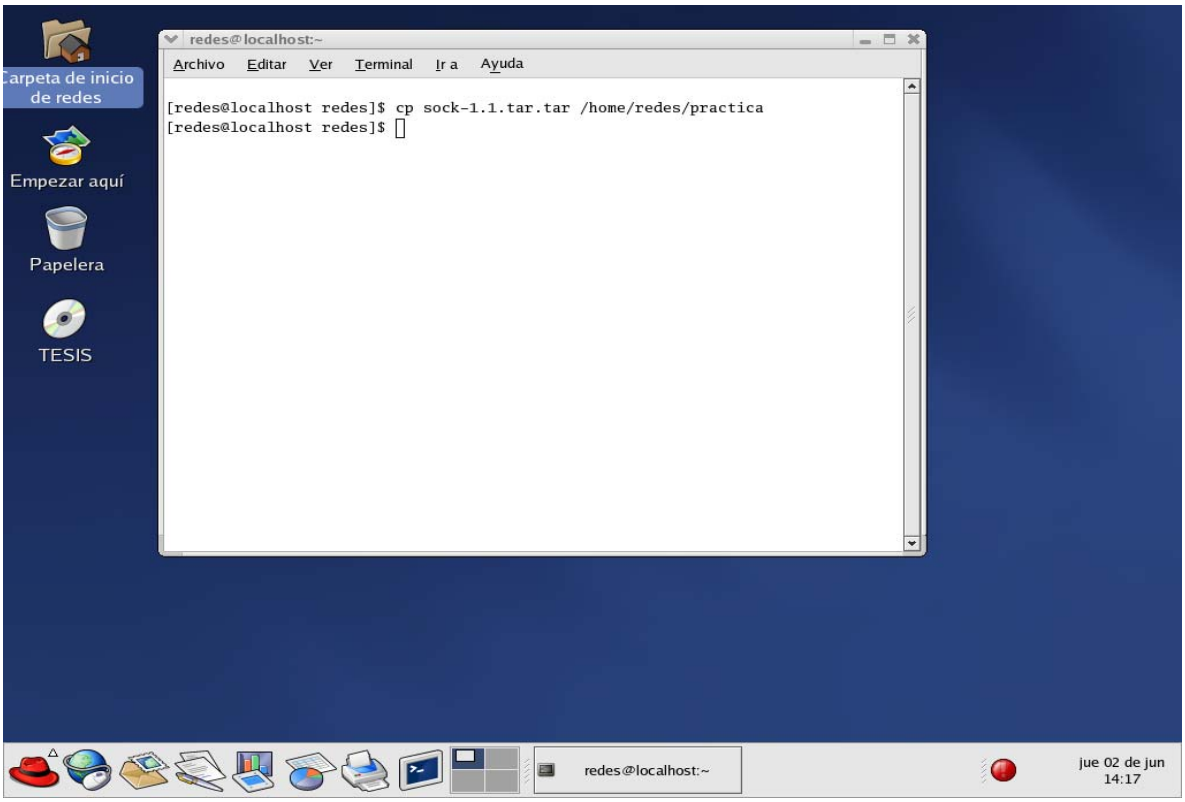


Figura No.4.

4.2.7 Cámbiese el subdirectorio practica y descomprima el archivo sock-1.1.tar.tar (Ver figura No.5).

[redes@localhost redes]$ cd practica
[redes@localhost practica]$ tar xvf sock-1.1.tar.tar

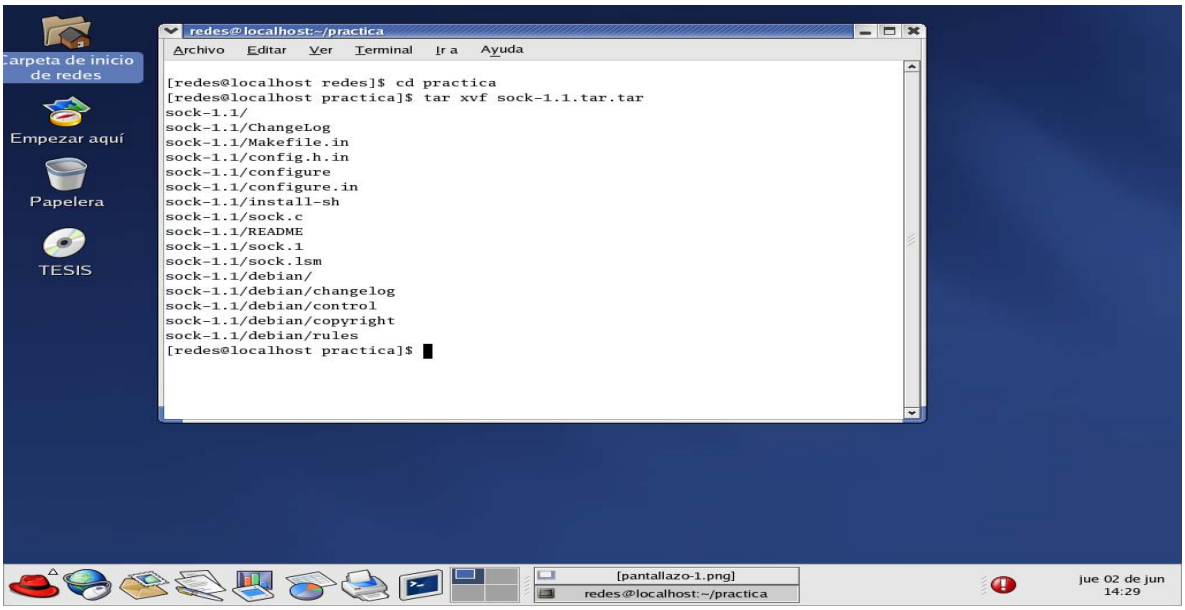


Figura No.5.

4.2.8 Sitúese dentro del subdirectorio sock-1.1 y ejecute la orden ./configure con la que el programa quedará preparado para su compilación y montaje. (Ver figura No.6).

[redes@localhost practica]$ cd sock-1.1
[redes@localhost sock-1.1]$ ./configure

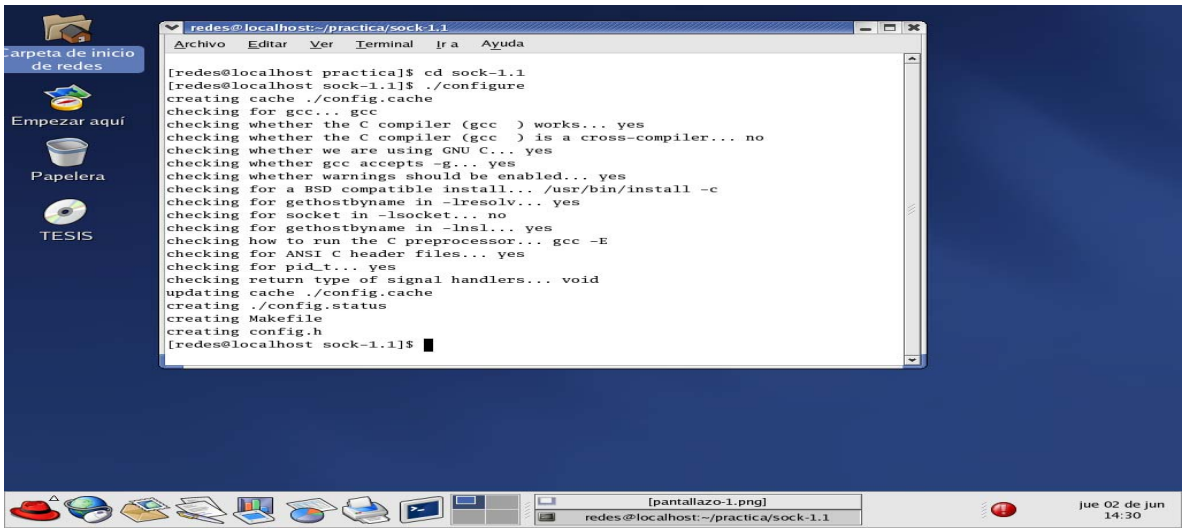


Figura No.6.

4.2.9 Compile el programa. Ahora ya disponemos del programa sock ejecutable. (Ver figura No.7).

[redes@localhost sock-1.1]$ make

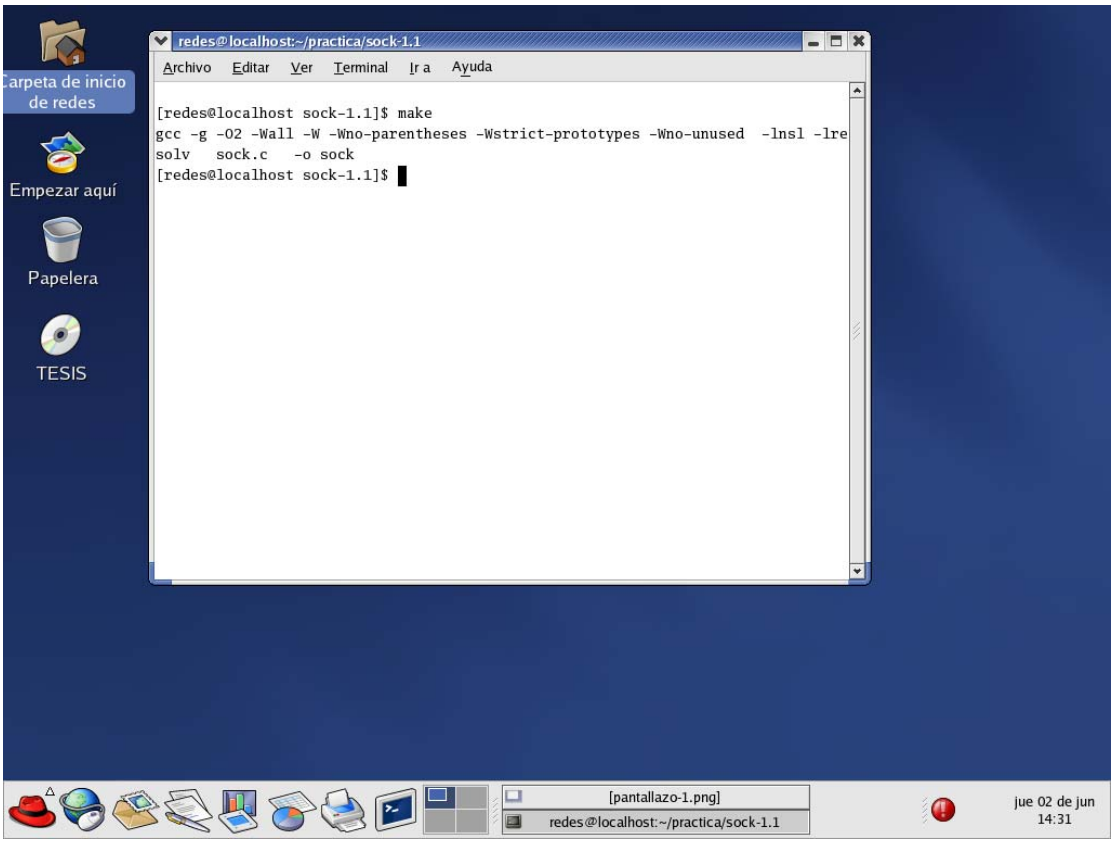


Figura No. 7.

4.3 Clientes TCP

Vamos a comenzar viendo qué sucede cuando un navegador se dirige a un servidor de web y le solicita una página.

4.3.1 En el shell teclee lo siguiente y después de pulsar la tecla retorno de carro, escriba el texto GET / HTTP/1.0
(Ver figura No.8).

[redes@localhost sock-1.1]$. /sock –e www.fi-b.unam.mx:80

GET / HTTP/1.0

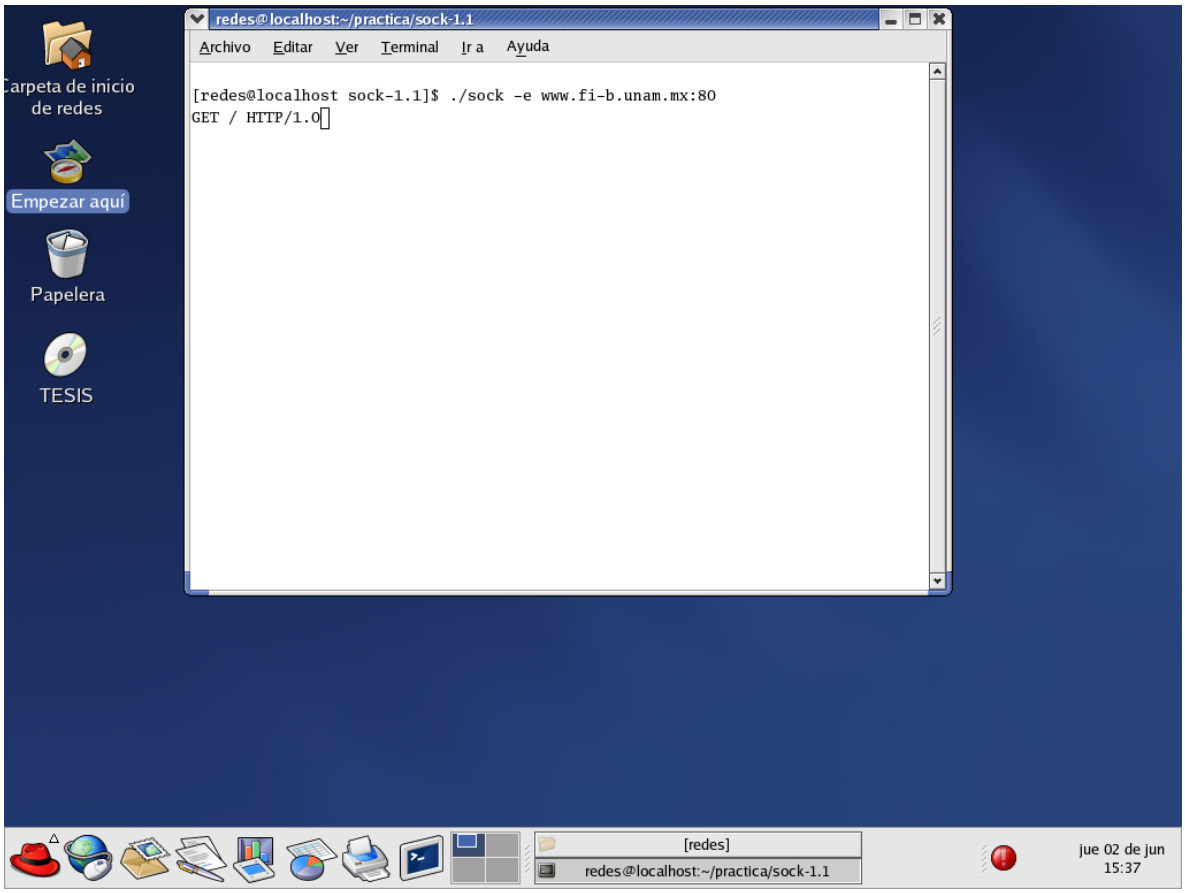


Figura No.8.

NOTA: Hay que pulsar la tecla de retorno de carro dos veces después de la segunda línea.

Con esto se está conectando al servidor www.fi-b.unam.mx (que es el servidor web de la DIE) al puerto 80, que es donde se encuentra este servicio habitualmente (*well-known port*) y se utiliza el protocolo TCP.

Como resultado de esta acción se recibirá una pantalla parecida a la siguiente: (Ver figura No.9).

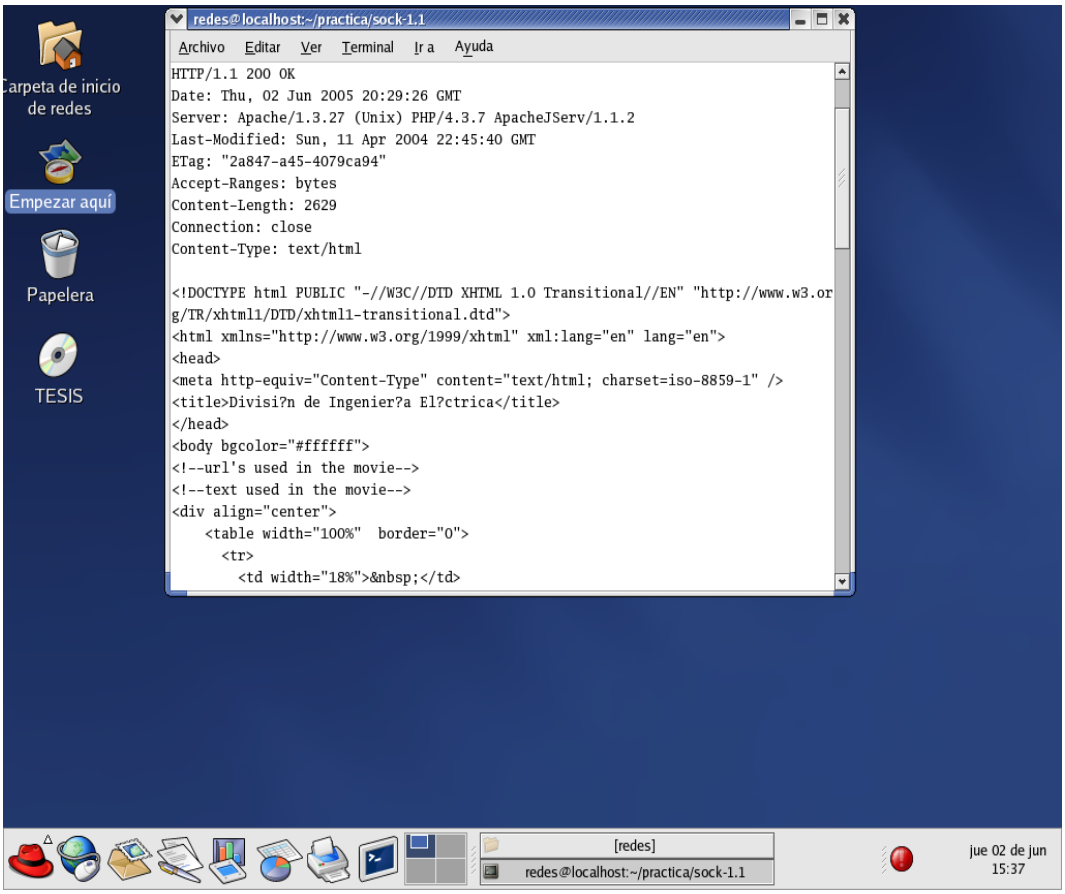


Figura No.9.

Lo que estamos haciendo es crear un socket en nuestra computadora. Ese socket, que actúa como cliente, lo conectamos al servidor de web de la DIE y le solicitamos que nos envíe el contenido de su página web inicial.

La conexión iniciada por el programa sock se realiza al puerto 80 del servidor www.fi-b.unam.mx y dura sólo lo indispensable hasta que se entrega la página web solicitada.

Es importante destacar que la respuesta del servidor contiene una información del protocolo HTTP (o cabecera) a la que sigue, después de una línea en blanco, el código HTML de la página solicitada. Tras enviar esa información el servidor cierra la conexión, con lo cual la ejecución de la orden sock finaliza.

4.3.2 En el shell teclee lo siguiente: (Ver figura No.10).

[redes@localhost sock-1.1]$ ./sock :22

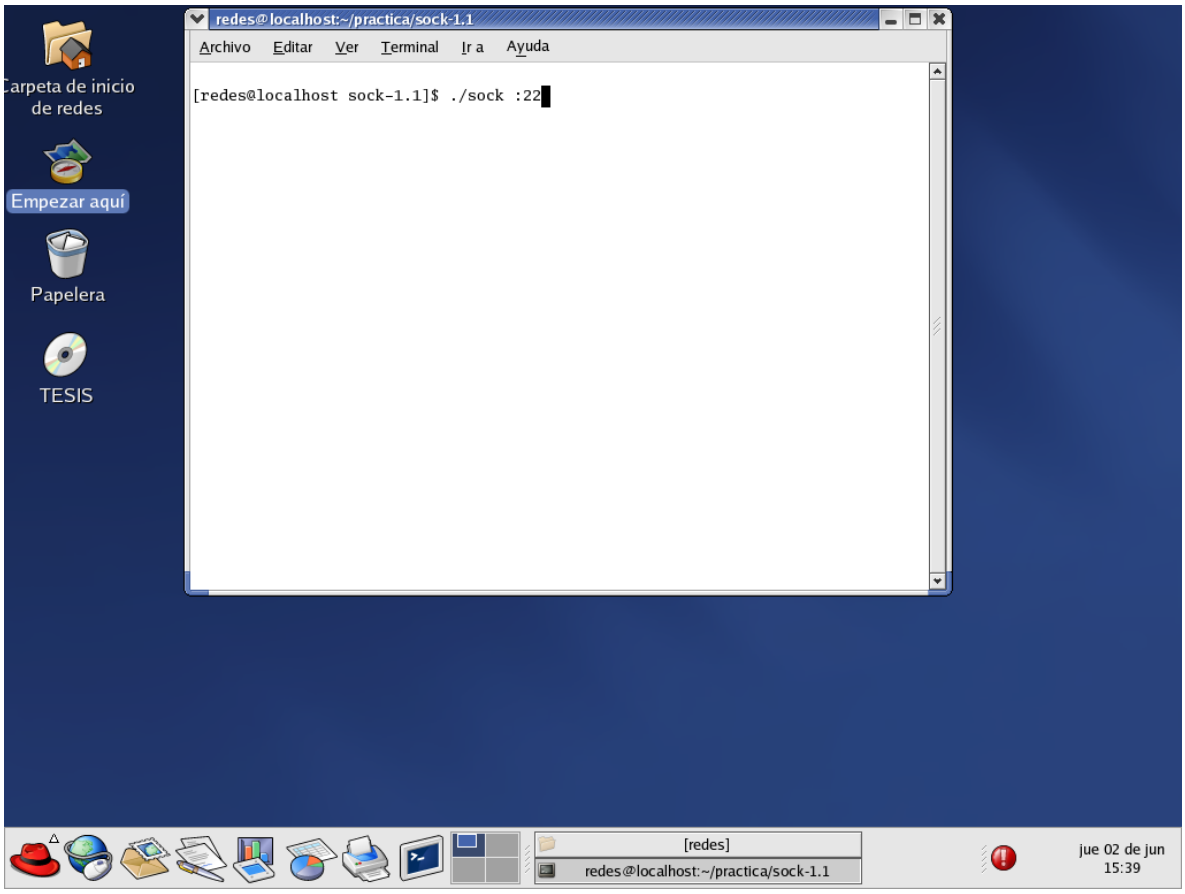


Figura No.10.

Deberá obtener como resultado algo así: (Ver figura No.11).

SSH-1.99-OpenSSH_2.9p1

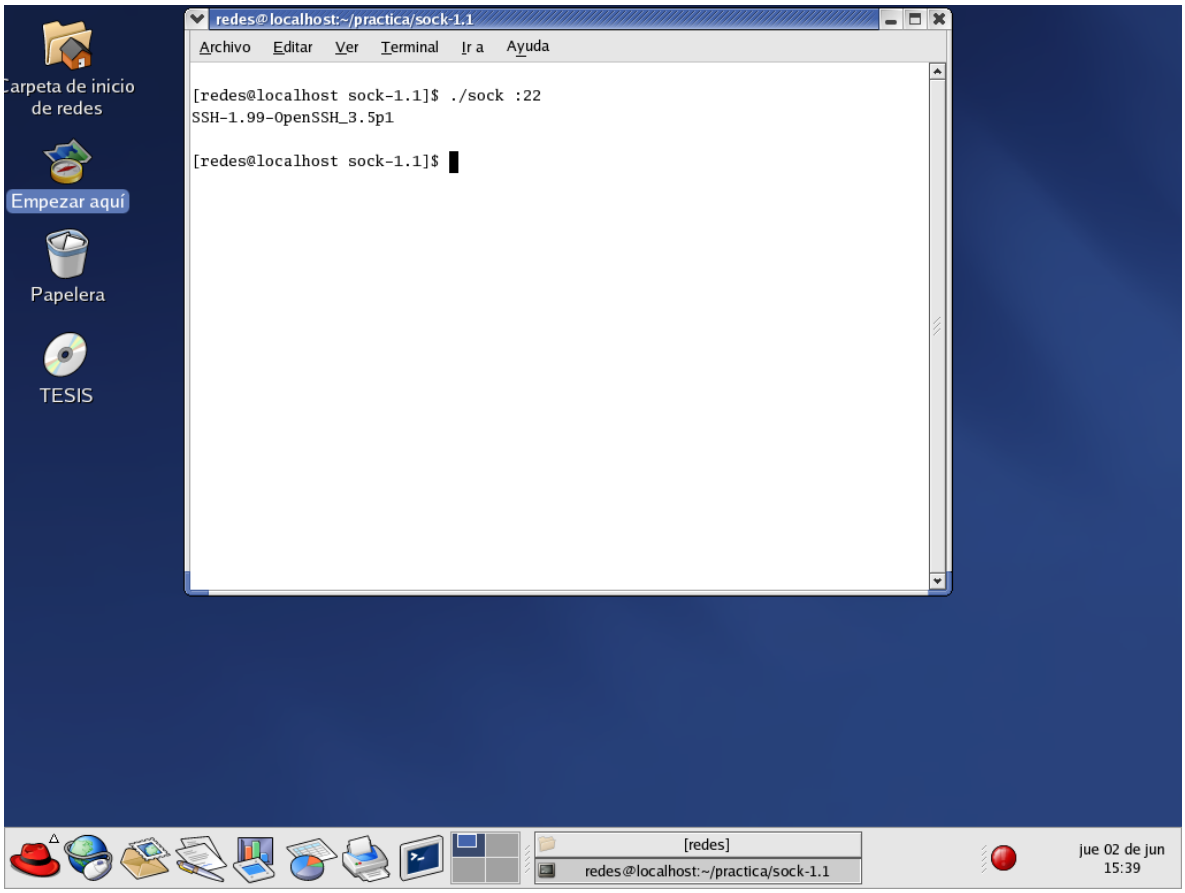


Figura No. 11.

4.3.3 Observará que el programa no finaliza, para que lo haga pulse las teclas <CTRL>+<c>.

En este ejercicio se está conectando con el servidor SSH local que se está ejecutando en la misma computadora desde el que ejecuta la orden. Esto es así porque al no especificar un servidor y sólo un puerto (:22) se entiende que nos referimos a la computadora local.

El servidor SSH comienza enviando una cadena que identifica la versión del programa, y eso es lo que obtenemos como resultado.

4.4 Servidor TCP

Los programas pueden esperar pacientemente a que se les solicite algo antes de enviar alguna información. Este es el

comportamiento de muchos servidores. Utilizando el programa sock va a crear un servidor cuya única función es esperar a que un cliente se conecte y luego conecta la entrada y salida estándar con ese cliente.

4.4.1 Para crear un socket servidor, teclee lo siguiente en el shell: (Ver figura No. 12).

[redes@localhost sock-1.1]$ ./sock -le :7701

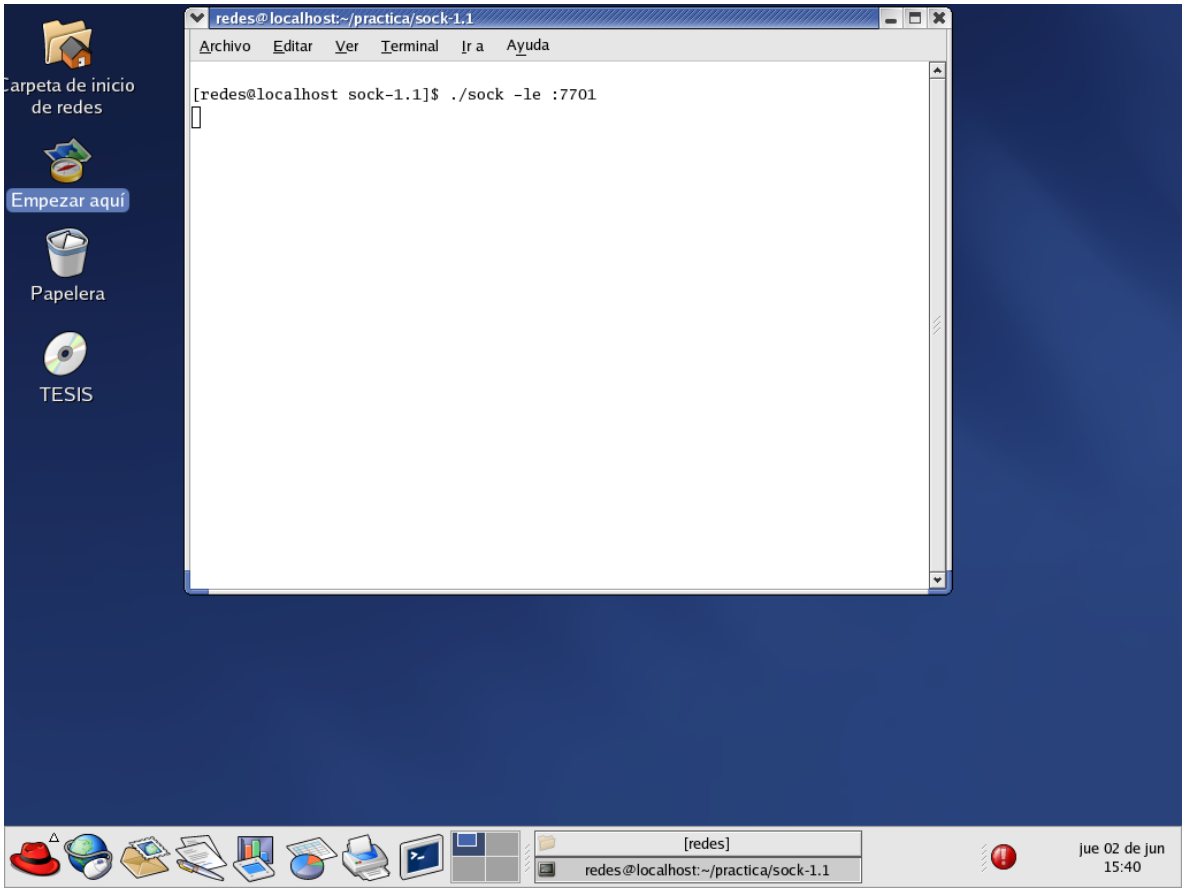


Figura No.12.

4.4.2 Ahora, abra un nuevo shell, sitúese en el subdirectorio sock-1.1 y ejecute la siguiente orden: (Ver figura No.13).

[redes@localhost sock-1.1]$ ./sock -e :7701

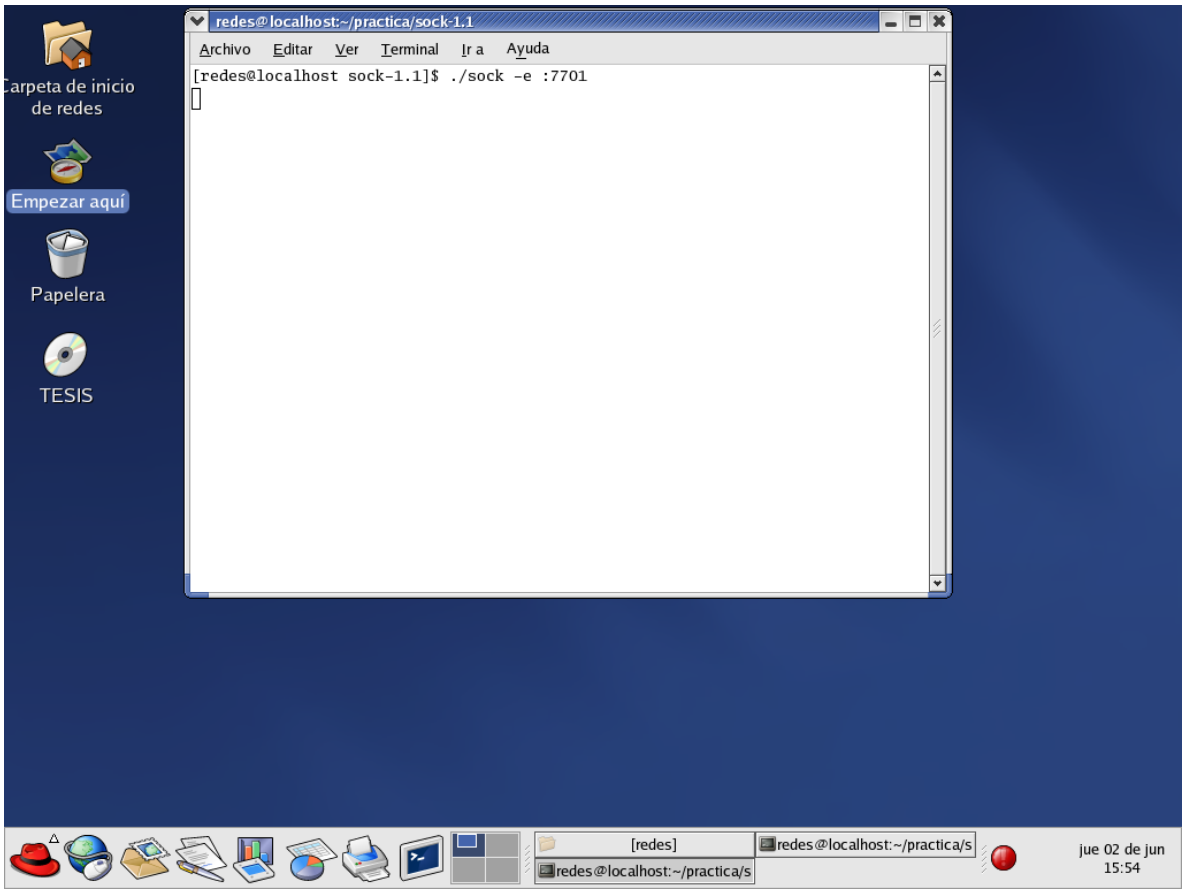


Figura No.13.

4.4.3 Escriba en este segundo shell y después del retorno de carro observe los que sucede en el primer shell. Seguidamente escriba en el primer shell, ¿qué sucede en el segundo shell? (Ver figura No.14).

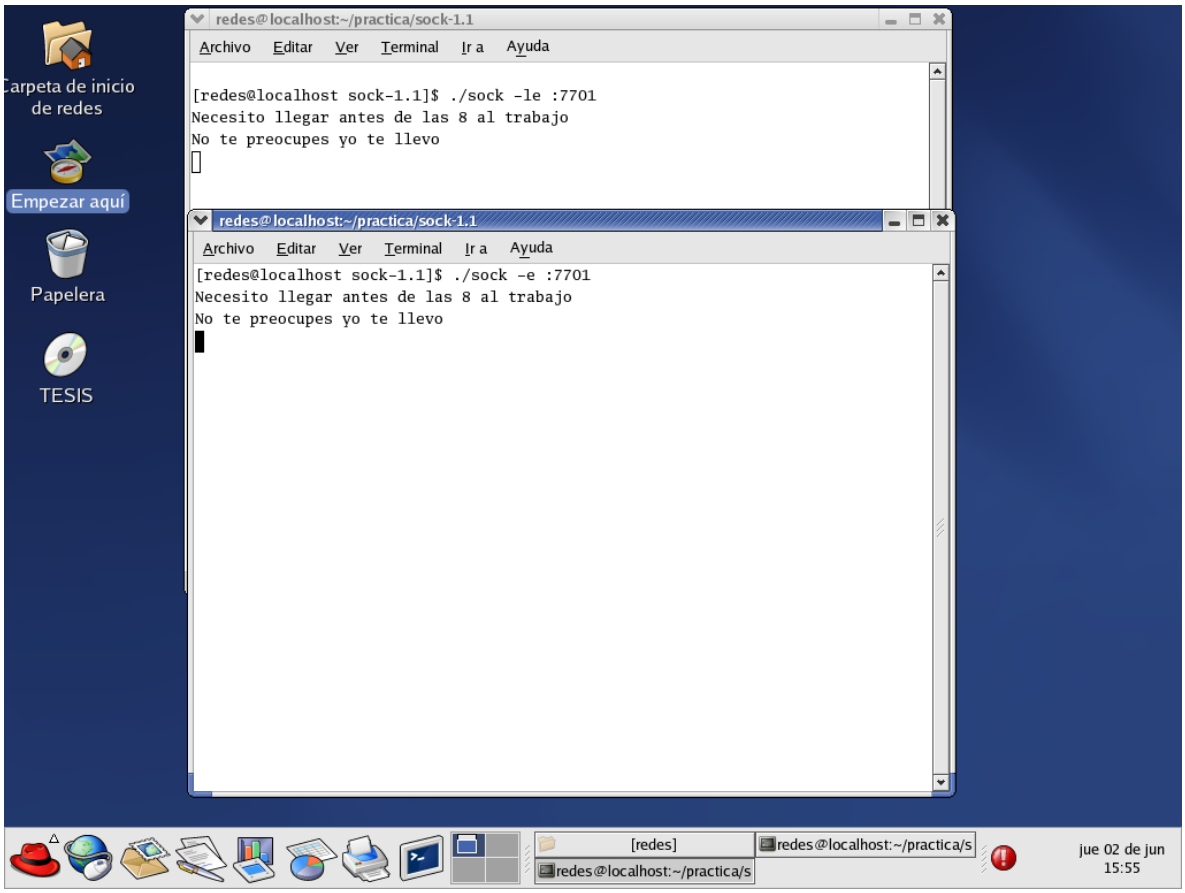


Figura No.14.

4.4.4 Cierre ambos shell.

La orden del punto 4.4.2 es equivalente a:

telnet localhost 7701

El parámetro -l hace que la aplicación configure el socket en modo escucha (*listen*) y acepte peticiones. Por tanto, en el punto 4.4.1 ha puesto en marcha, en su computadora, un servidor que escucha en el puerto 7701. Mientras que las órdenes de los pasos 4.4.2 y 4.4.3 han arrancado clientes TCP que se han conectado a ese puerto.

4.4.5 Abra un shell, sitúese en el subdirectorio sock-1.1 y teclee lo siguiente: (Ver figura No.15).

[redes@localhost sock-1.1]$ ./sock -l :7701 -d ls

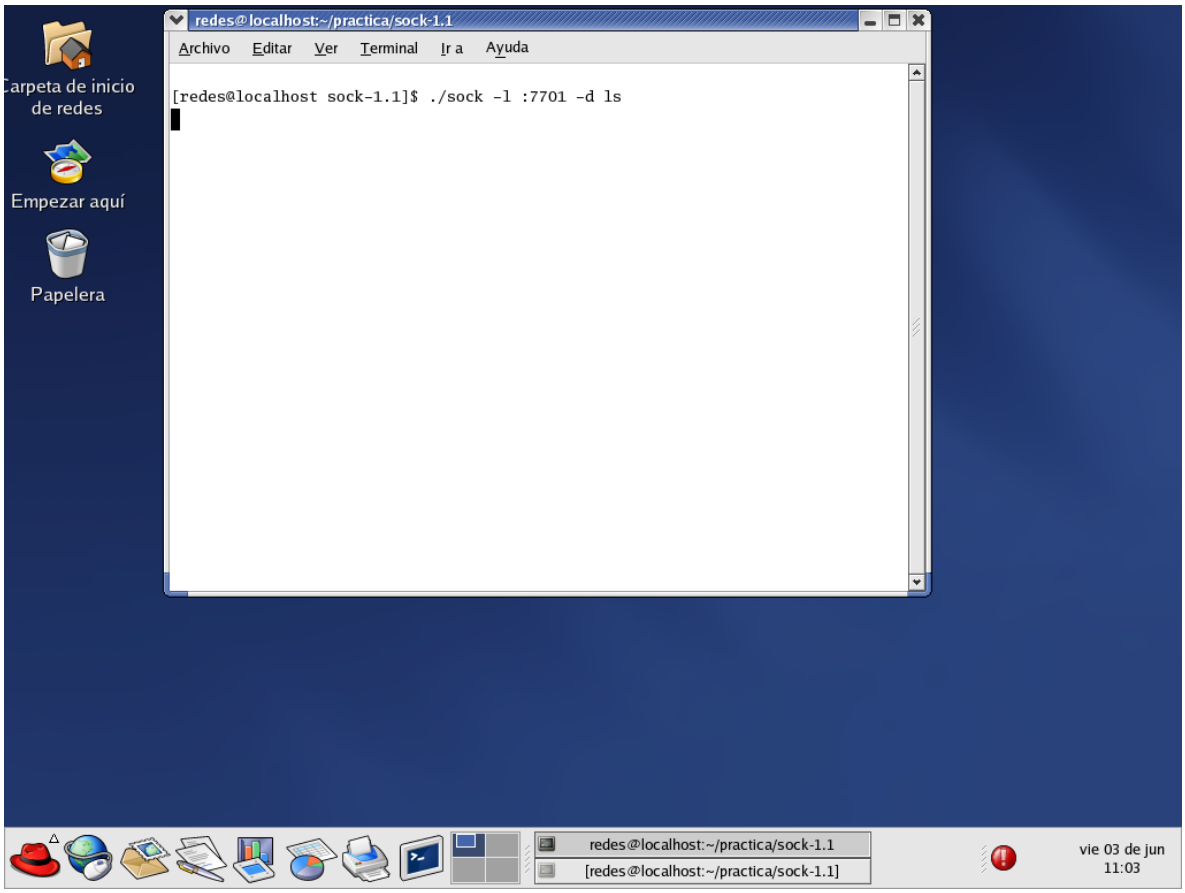


Figura No.15.

4.4.6 Ahora, en otro shell, sitúese en el subdirectorio sock-1.1 y ejecute la orden: (Ver figura No.16).

[redes@localhost sock-1.1]$ ./sock -e :7701

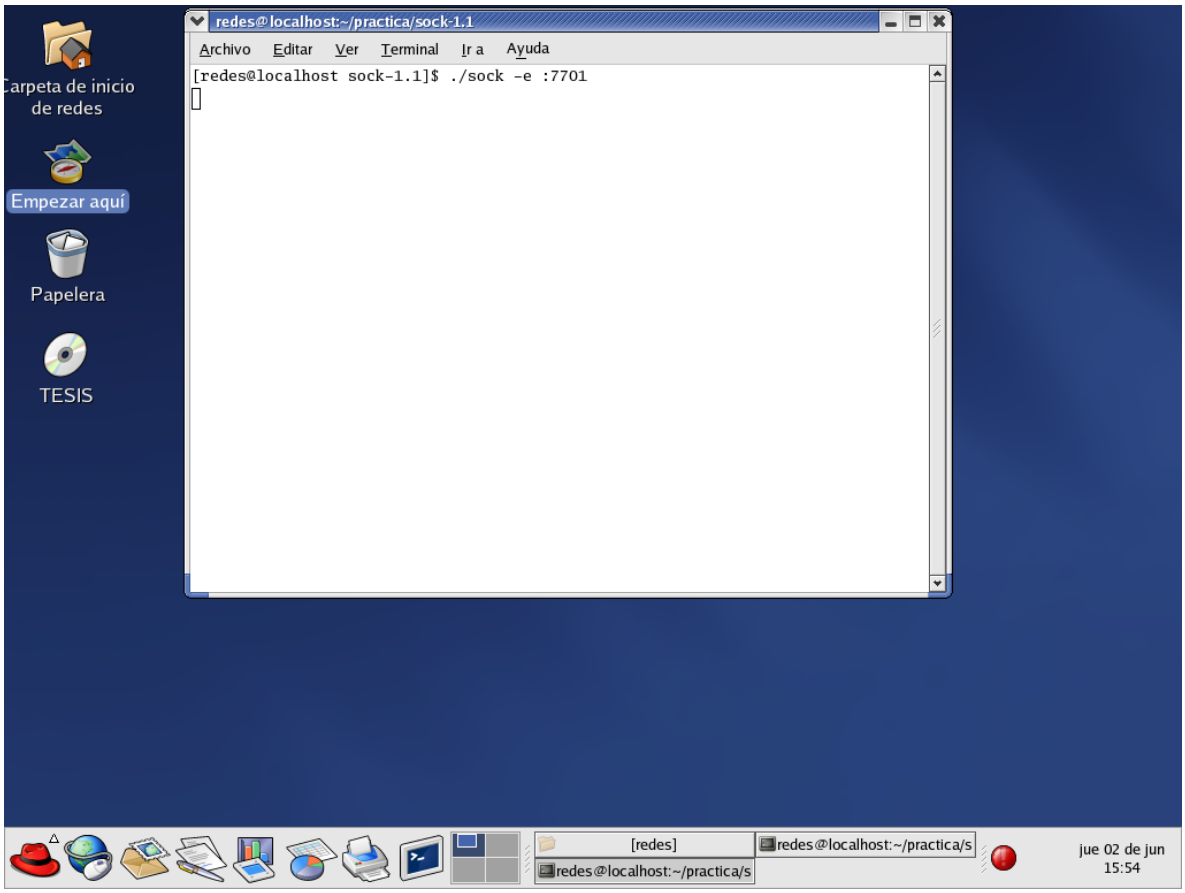


Figura No.16.

4.4.7 Observe lo que sucede. (Ver figura No.17).

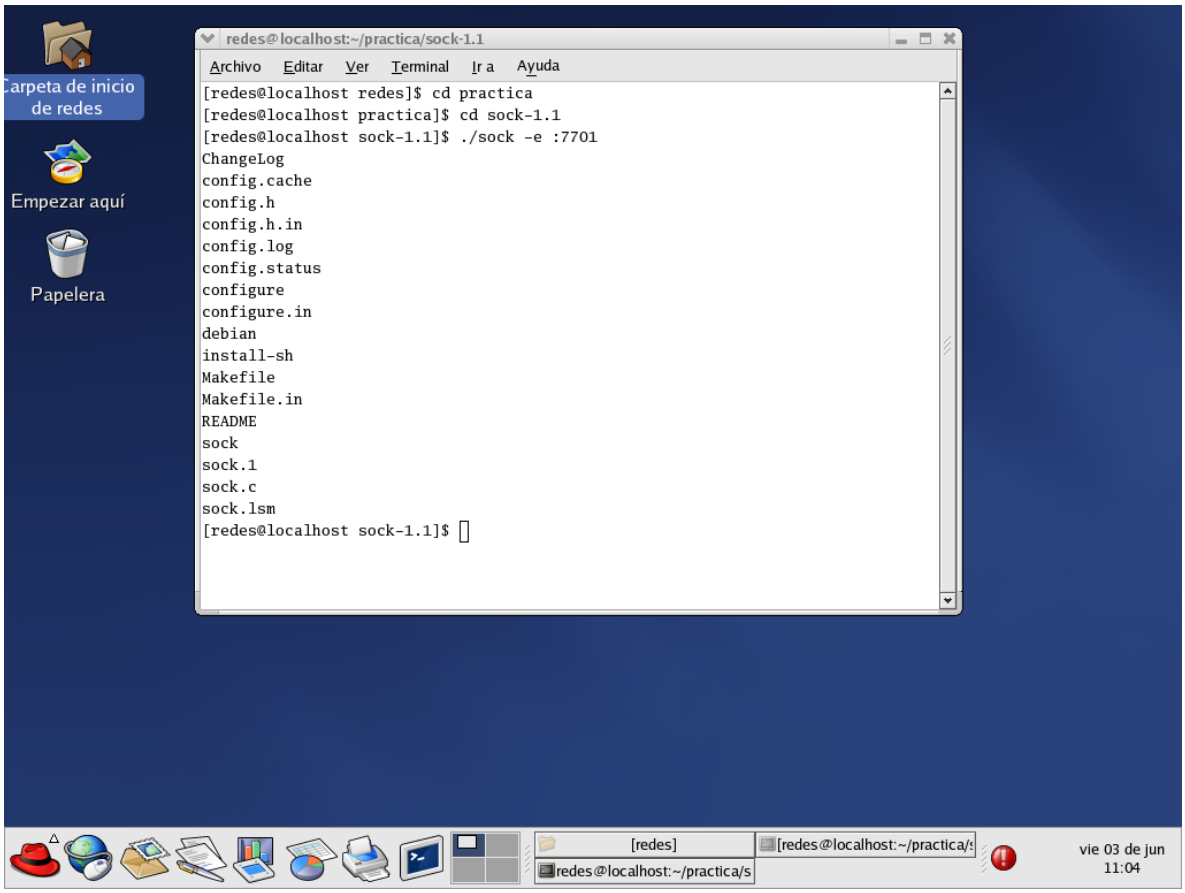


Figura No.17.

En este experimento se ha construido un “miniservidor”. Lo que hace el programa es esperar la conexión de un usuario al puerto indicado (7701 en este caso) y cuando el cliente se conecta (mediante la orden sock o el programa telnet) entonces ejecuta la orden ls que lista el contenido del directorio y lo envía a través del socket. Una vez finalizada la orden ls el servidor corta la conexión del cliente telnet, pero sigue escuchando en el puerto para atender nuevas peticiones de otros clientes.

Si se sustituye la orden ‘ls’ por la orden ‘date’ en el punto 4.4.5 tendrá un miniservidor de fecha y hora.(Ver figura No.18).

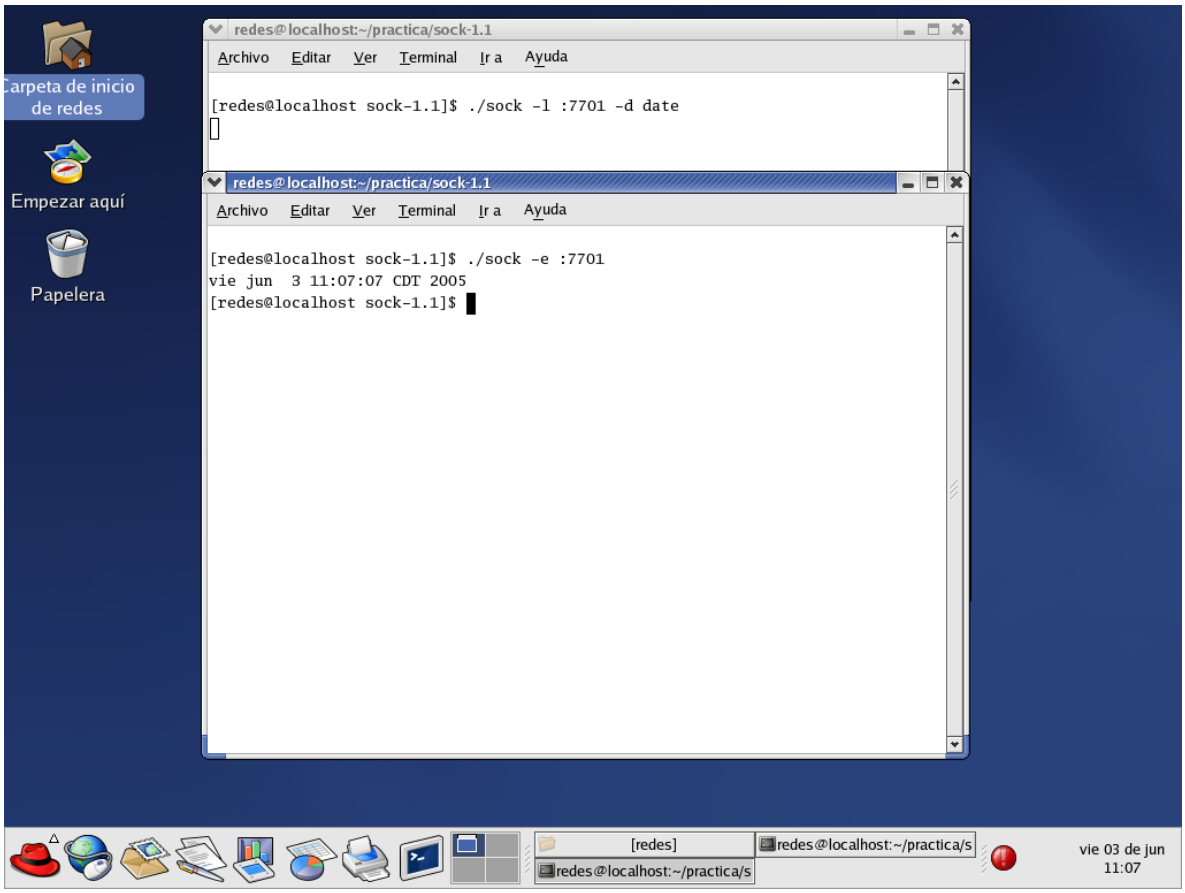


Figura No.18.

4.5 El protocolo UDP

Del mismo modo que en los ejemplos anteriores ha utilizado el protocolo TCP, ahora va a ver cómo se puede enviar información mediante el protocolo UDP. Para ello mantendrá los dos shell que tiene abiertos.

4.5.1 En un shell teclee lo siguiente: (Ver figura No.19).

[redes@localhost sock-1.1]$ ./sock -ul :7701

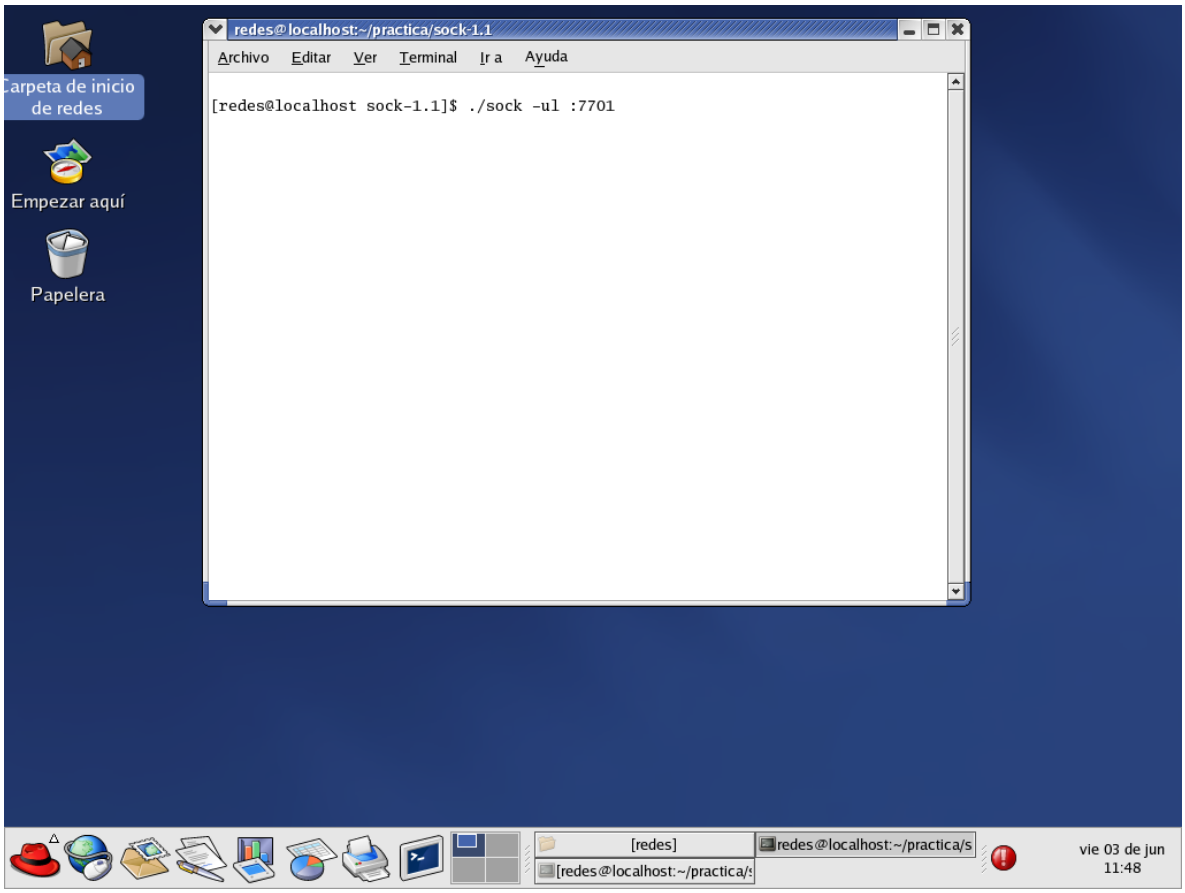


Figura No.19.

4.5.2 Y en otro shell ejecute la orden: (Ver figura No.20).

[redes@localhost sock-1.1]$ ./sock -u :7701

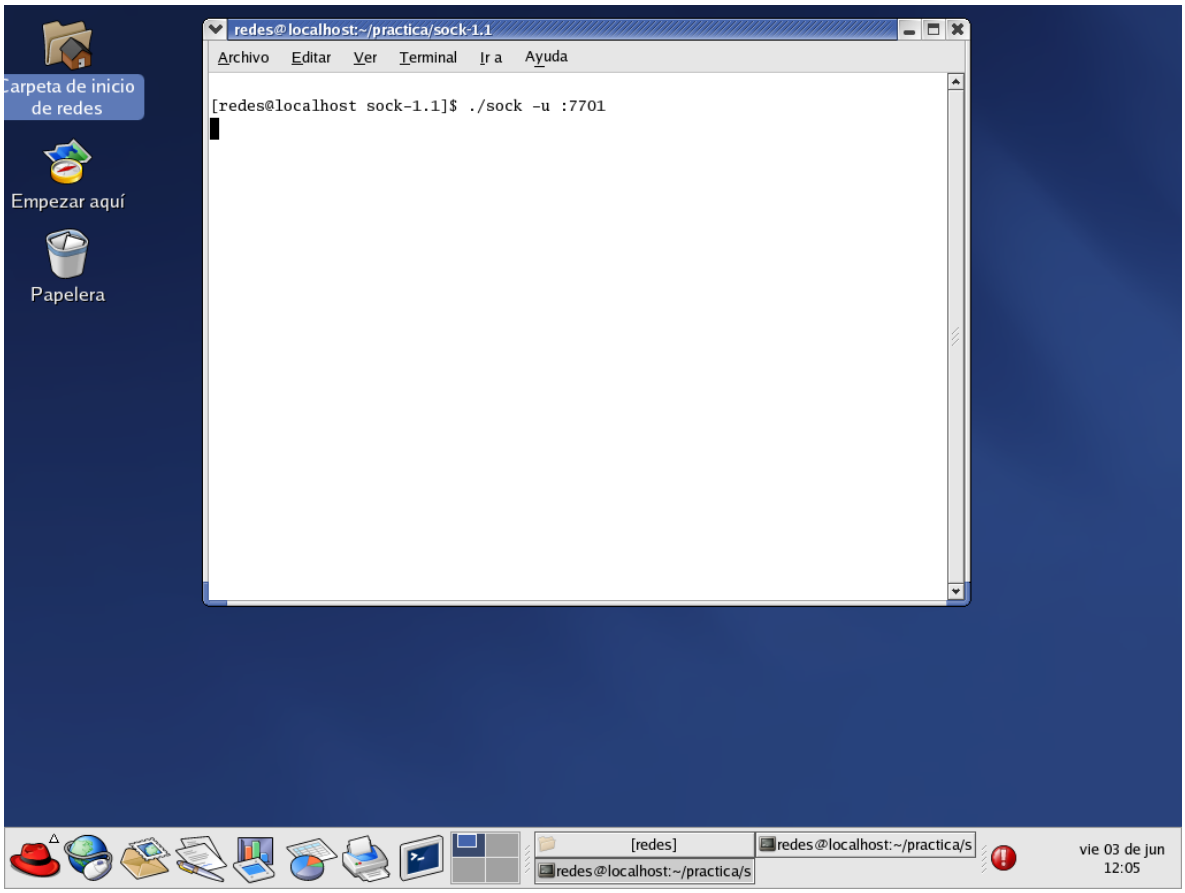


Figura No.20.

4.5.3 Escriba en este segundo shell y después del retorno de carro observe los que sucede en el primero. (Ver figura No.21).

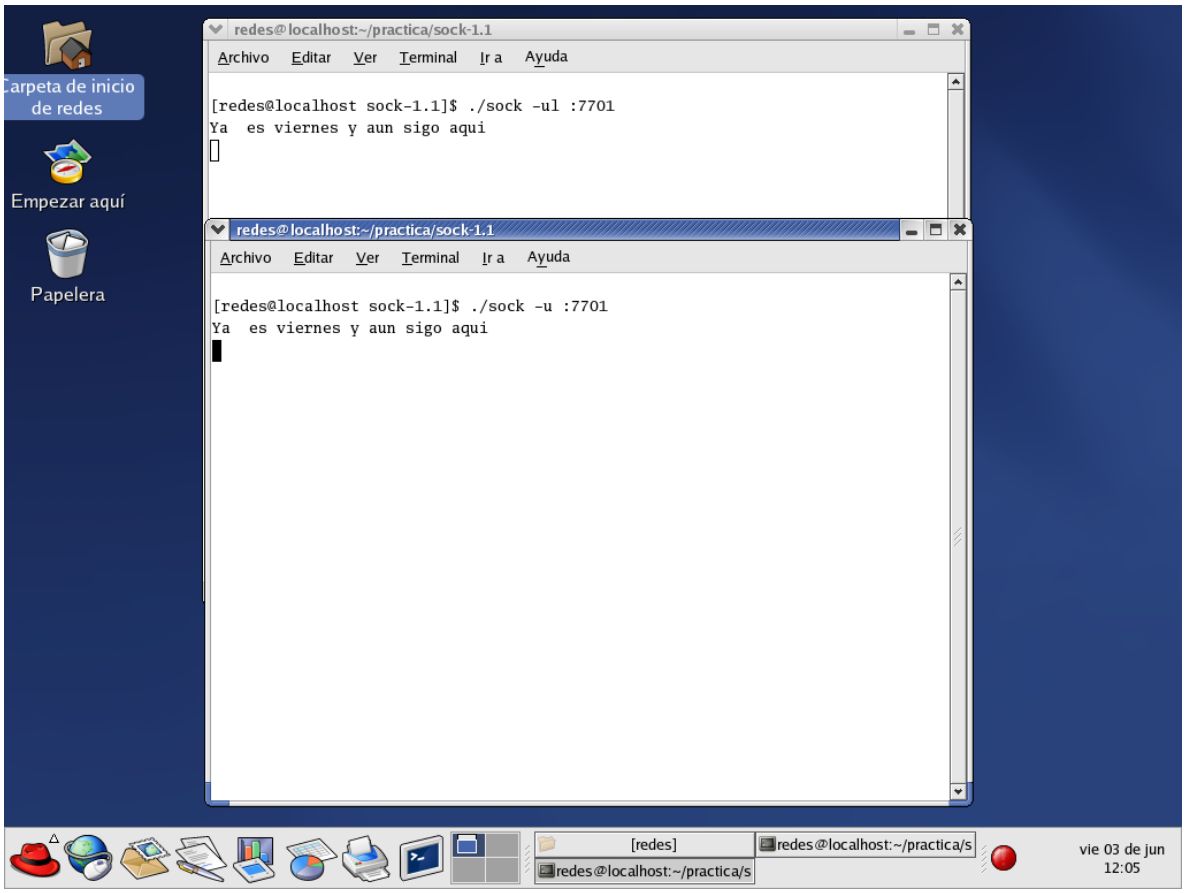


Figura No.21.

4.5.4 Ahora en este último shell cambie la orden del paso número 4.5.2 por la siguiente: (Ver figura No.22).

[redes@localhost sock-1.1]$ date | ./sock -u :7701

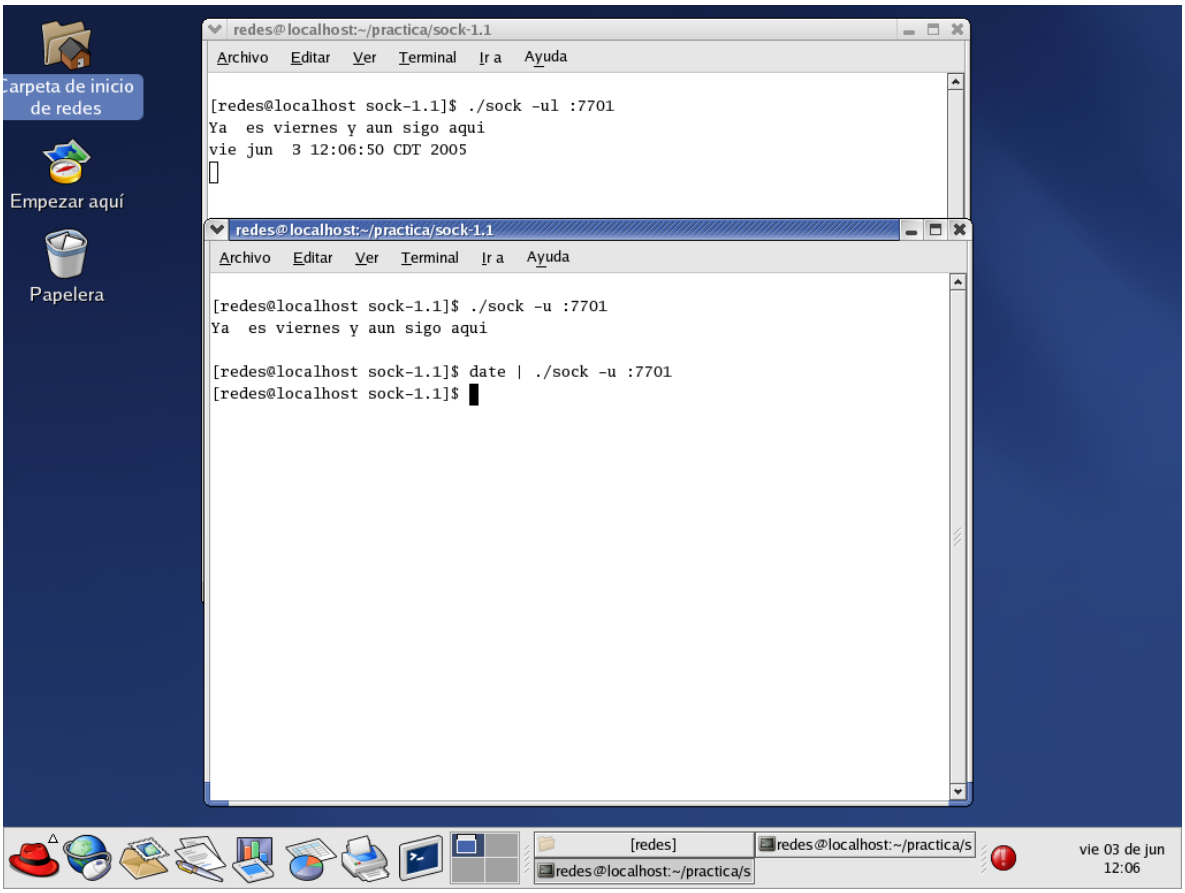


Figura No.22.

Como ve el funcionamiento es bastante similar, pero al carecer UDP del concepto de conexión no se puede construir un servidor de manera tan sencilla.

Pero la razón que hace que UDP tenga utilidad para muchas aplicaciones es su capacidad para hacer difusiones (enviando a la dirección 255.255.255.255 realmente se envía un datagrama que será recibido por todas las computadoras de la misma red IP). Sin embargo, y por motivos de seguridad, el uso de esta característica está restringido y no se empleará en esta práctica.

Una forma de evitar esta restricción es emplear la dirección IP de multicast que esté configurada en todos sus equipos como si se tratara de una dirección de difusión.

4.6 Transferencia de archivos

En los ejercicios anteriores ha visto algunos de los usos que nos permite un socket. Ahora va a utilizar los servicios de TCP y UDP para el envío de archivos entre dos computadoras.

En el siguiente ejercicio se mostrará cómo transferir un archivo empleando el programa sock:

4.6.1 Conéctese a la máquina de al lado con la cuenta redes desde uno de los shell tecleando: (Ver figura No.23).

[redes@localhost sock-1.1]$ ssh –l redes labrdXX

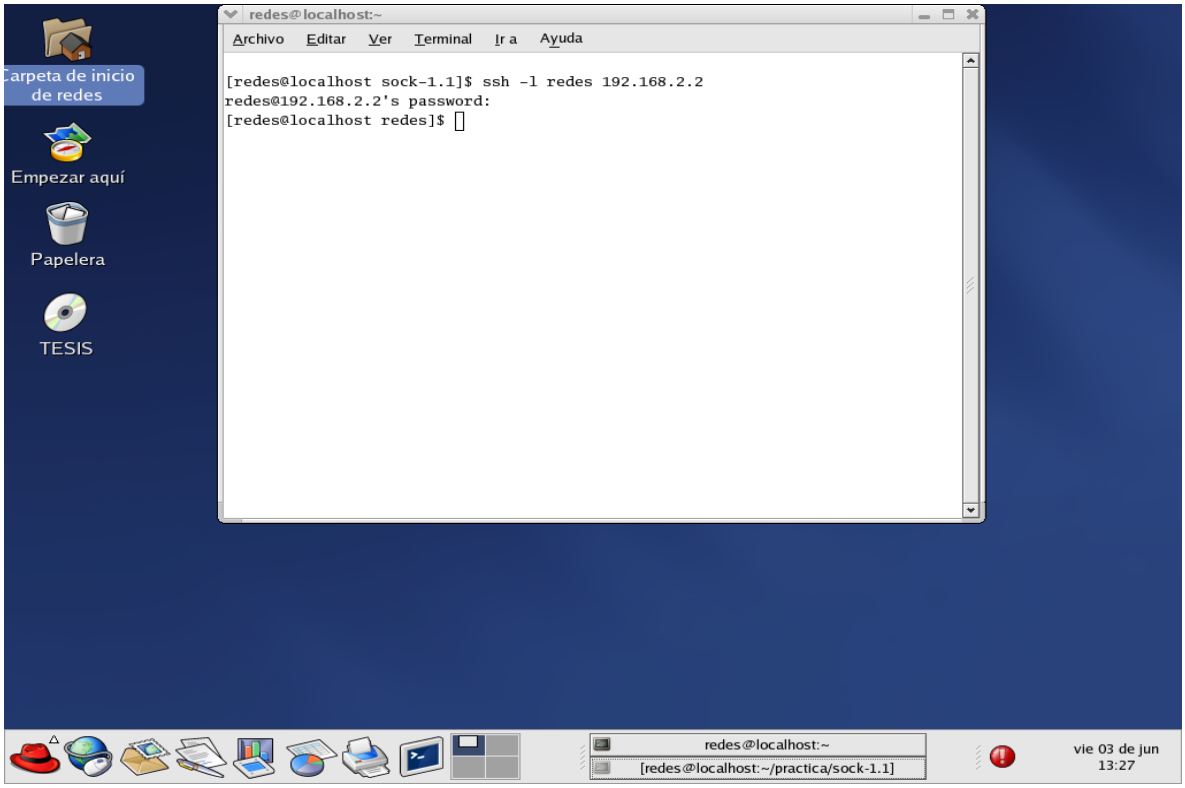


Figura No.23.

NOTA: labrdcXX se sustituirá por la IP de la computadora de al lado.

4.6.2 En otro shell monte su disquette y copie la imagen de la dirección /mnt/floppy en el subdirectorio

/home/redes/practica/sock-1.1 (Ver figura No.24).

[redes@localhost sock-1.1]$ mount /mnt/floppy
[redes@localhost sock-1.1]$ cp /home/redes/redhat.gif /home/redes/practica/sock-1.1

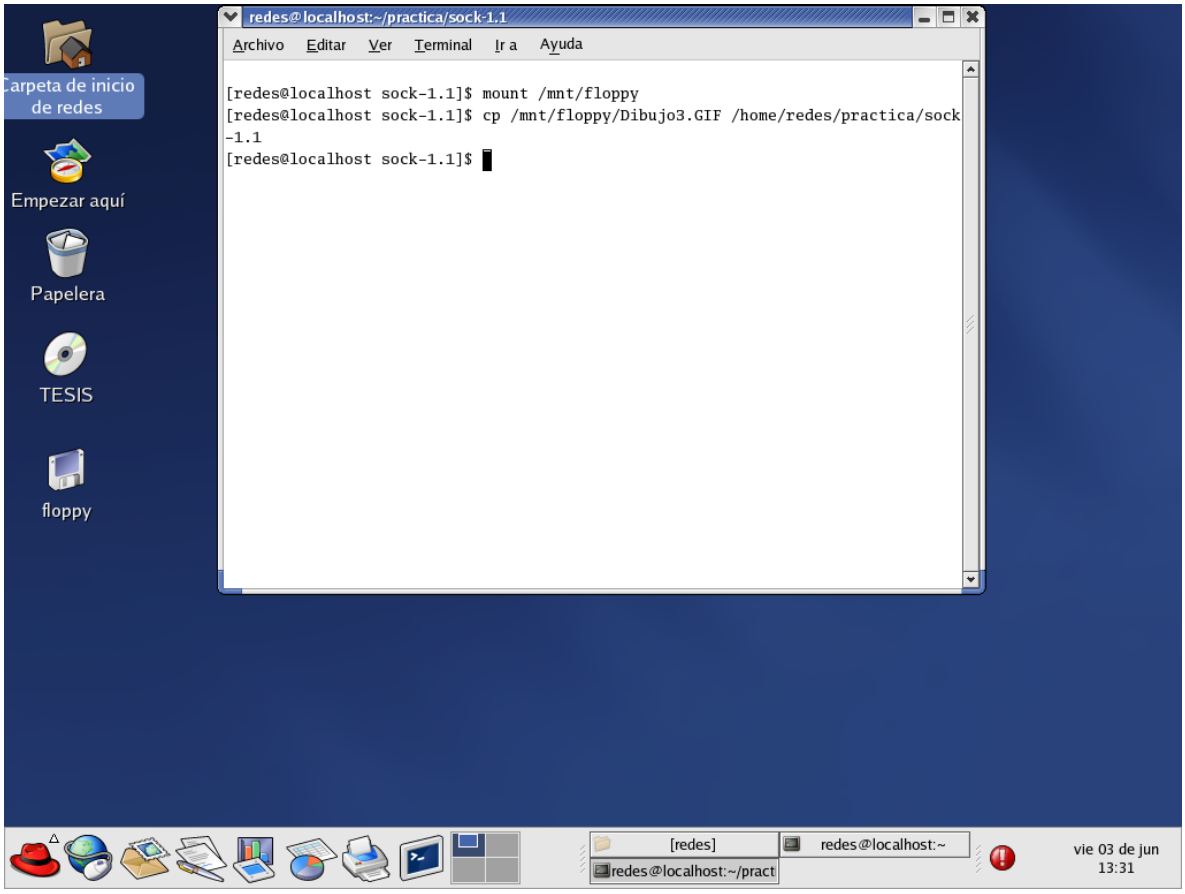


Figura No.24.

NOTA: redhat.gif se sustituirá por el nombre de la imagen de su disquete.

4.6.3 Ahora va a enviar la imagen tecleando en el segundo shell: (Ver figura No.25).

[redes@localhost sock-1.1]$ ./sock -l :8888 -d 'cat redhat.gif'

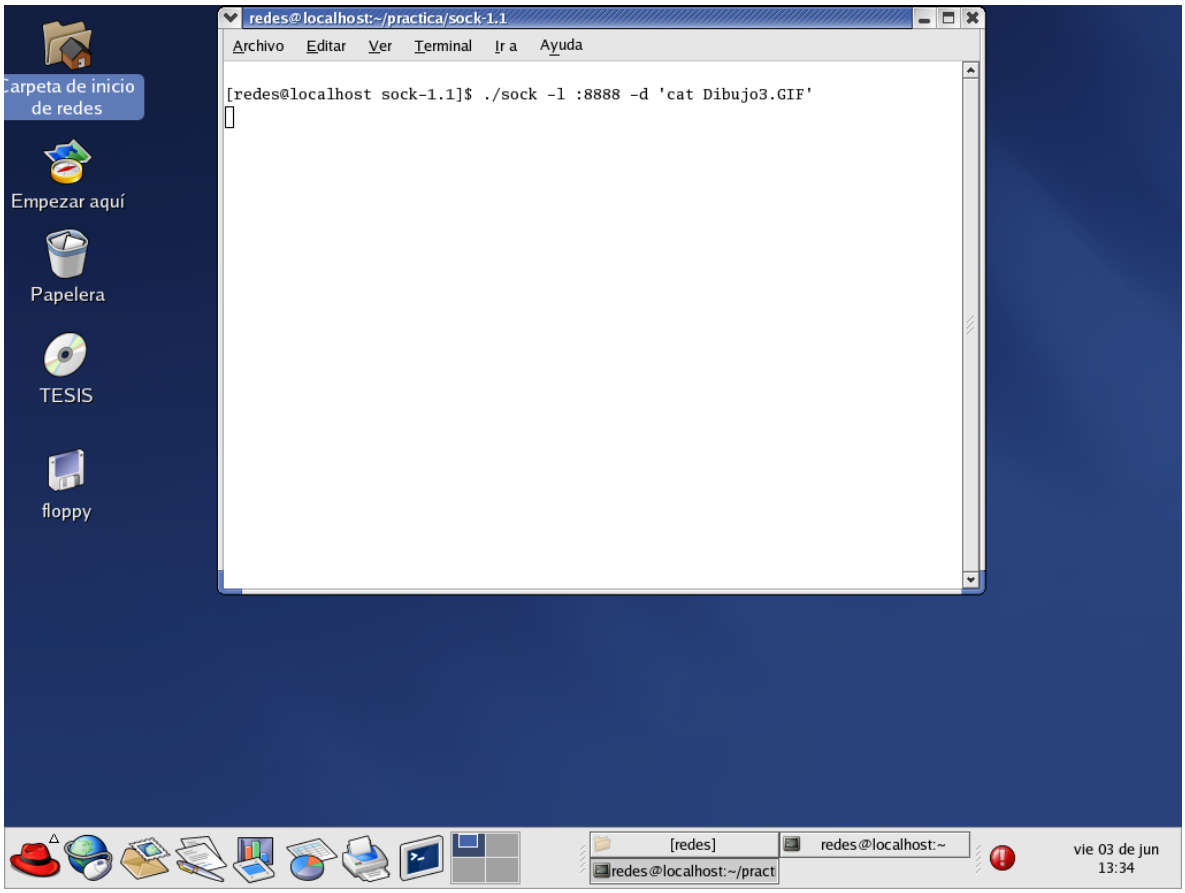


Figura No.25.

4.6.4 En la cuenta redes de la máquina de al lado, sitúese en el subdirectorio sock-1.1 y teclee: (Ver figura No.26).

[redes@localhost redes]$ cd practica/sock-1.1
[redes@localhost sock-1.1]$ ./sock –e rdcXX:8888>nube.gif

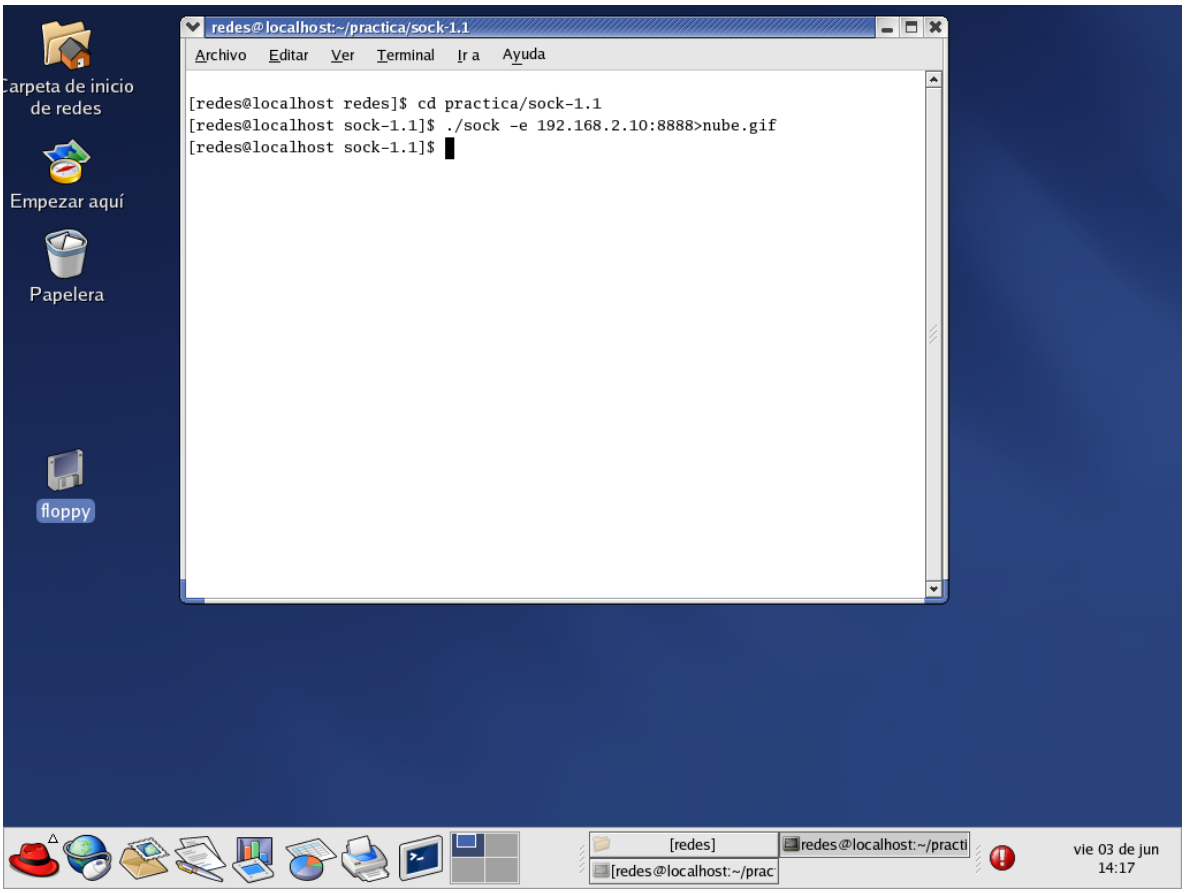


Figura No.26.

NOTA: rdcXX se sustituirá por la IP de su computadora (escrito en la etiqueta pegada en su monitor)

4.6.5 Compruebe que el archivo recibido en la máquina de al lado tiene el mismo tamaño que el original, utilice el comando: ls–la. (Ver figura No.27).

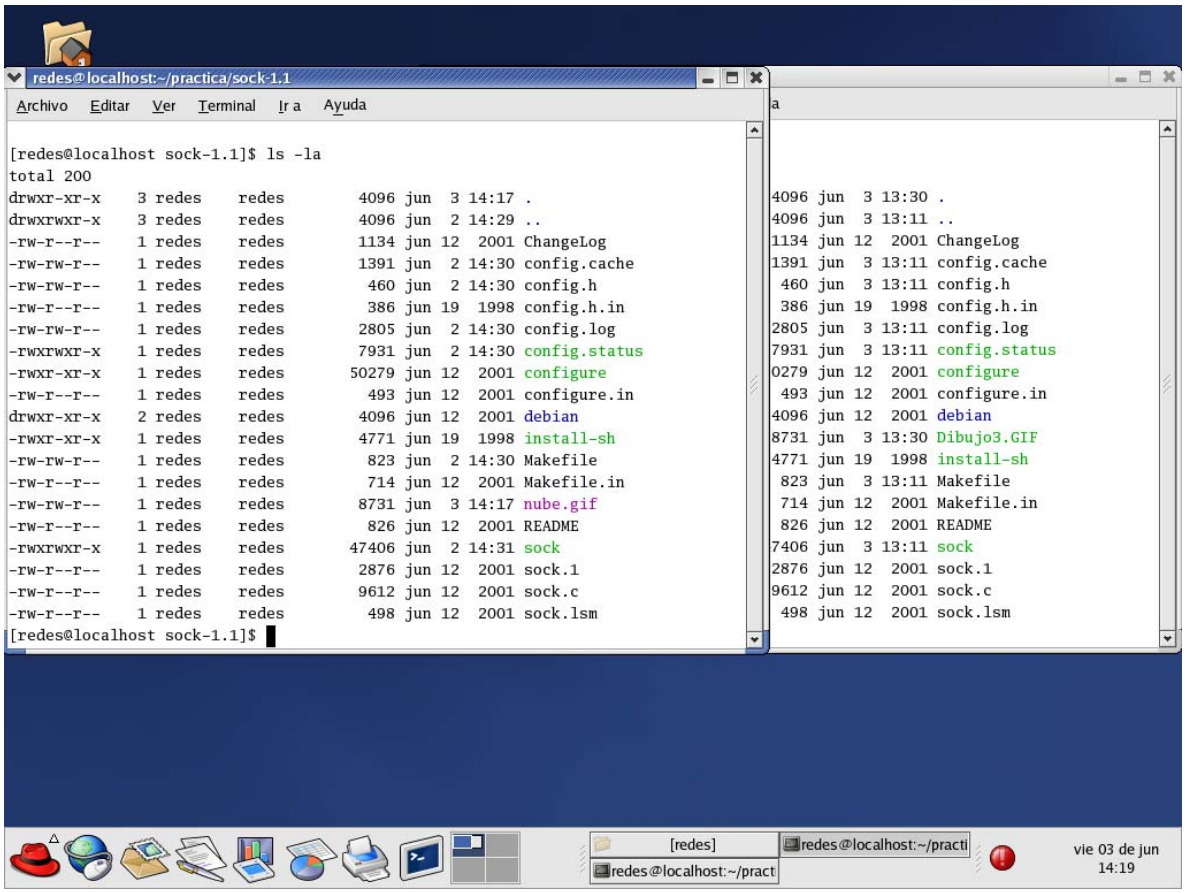


Figura No.27.

4.6.6 Cierre ambos shell.

En este ejercicio se ha realizado la transferencia del archivo mediante el protocolo TCP. Su computadora ha quedado a la espera de un cliente en el paso 4.6.3. Y desde la máquina de al lado se ha conectado como tal cliente en el paso 4.6.4.

Es interesante resaltar que aunque el archivo resultante tenga el mismo tamaño, eso no garantiza que la transferencia ha tenido éxito (¿y sí el contenido fuera diferente?). Ahora enviará el archivo de vuelta para poderlo comprobar, pero empleando el protocolo UDP.

4.6.7 Conéctese a la cuenta redes de la máquina de al lado desde un shell tecleando: (Ver figura No.28).

[redes@localhost redes]$ ssh –l redes lbrdXX

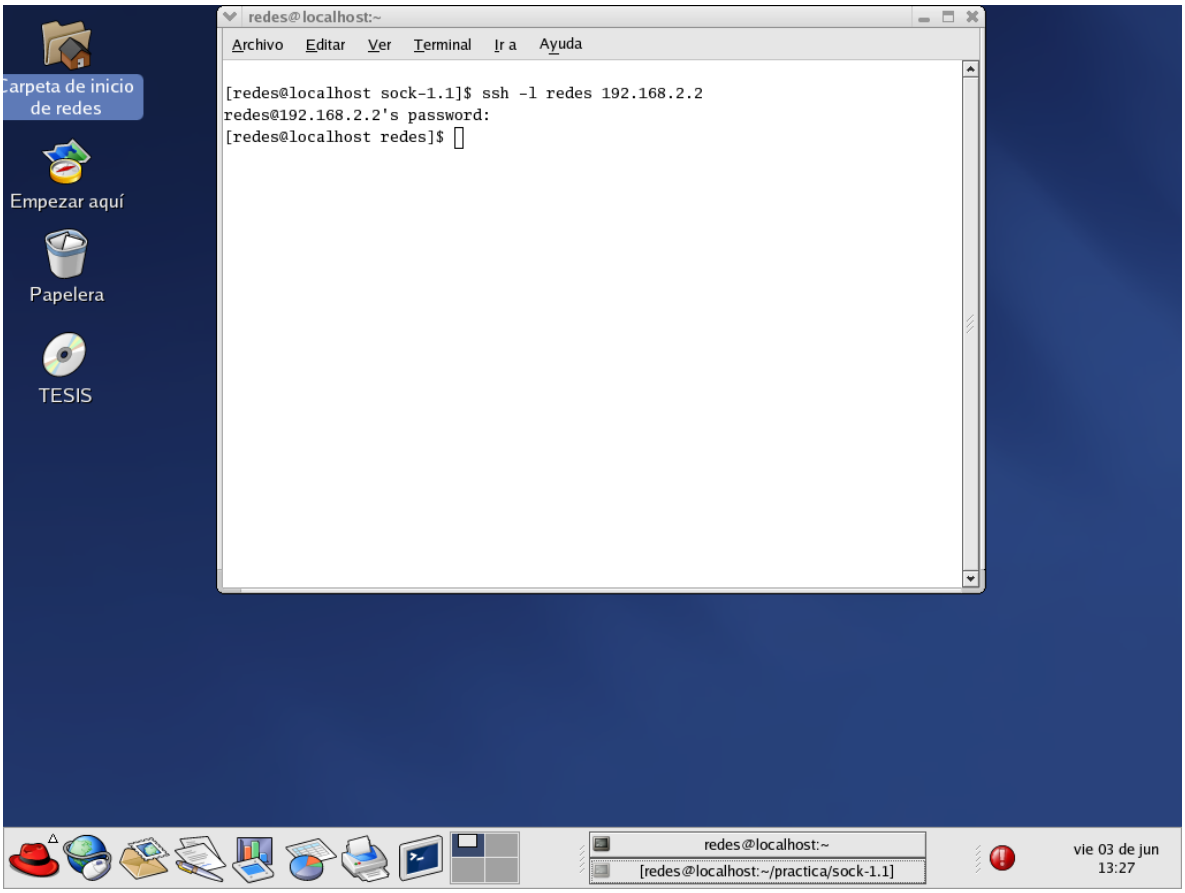


Figura No.28.

4.6.8 En otro shell, sitúese en el subdirectorio sock-1.1 y teclee: (Ver figura No.29).

[redes@localhost redes]$ cd practica/sock-1.1
[redes@localhost sock-1.1]$ ./sock -ul :8888>red.gif

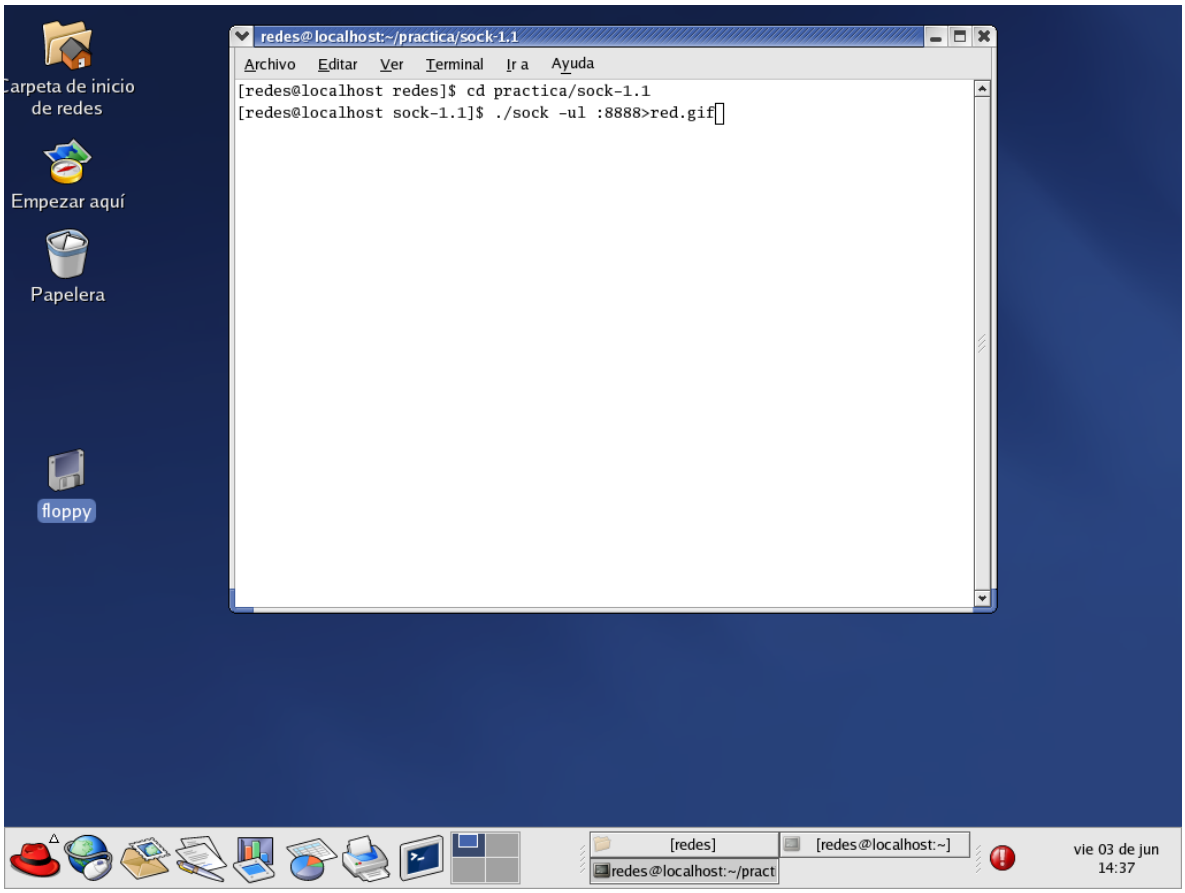


Figura No.29.

Lo que le prepara para recibir el archivo, -u indica UDP

4.6.9 En la cuenta redes de la máquina de al lado, sitúese en el subdirectorio sock-1.1y teclee lo siguiente para enviar el archivo: (Ver figura No.30).

[redes@localhost redes]$ cd practica/sock-1.1
[redes@localhost sock-1.1]$ cat nube.gif | sock –u rdcXX:8888

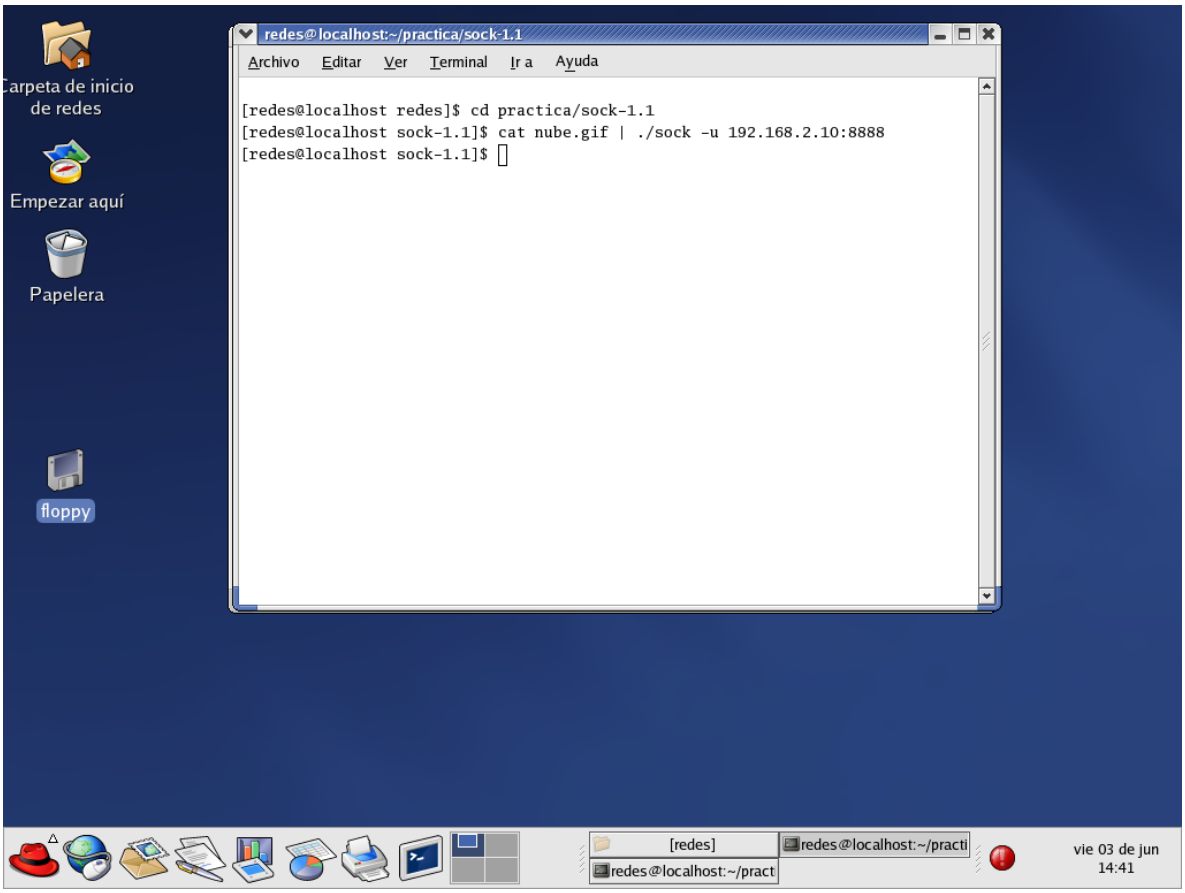


Figura No.30.

4.6.10 A continuación, finalice la orden del paso 4.6.8 pulsando <Ctrl>+<c> en esa ventana (asegúrese de que la ha seleccionado primero, haciendo click con el ratón). (Ver figura No.31).

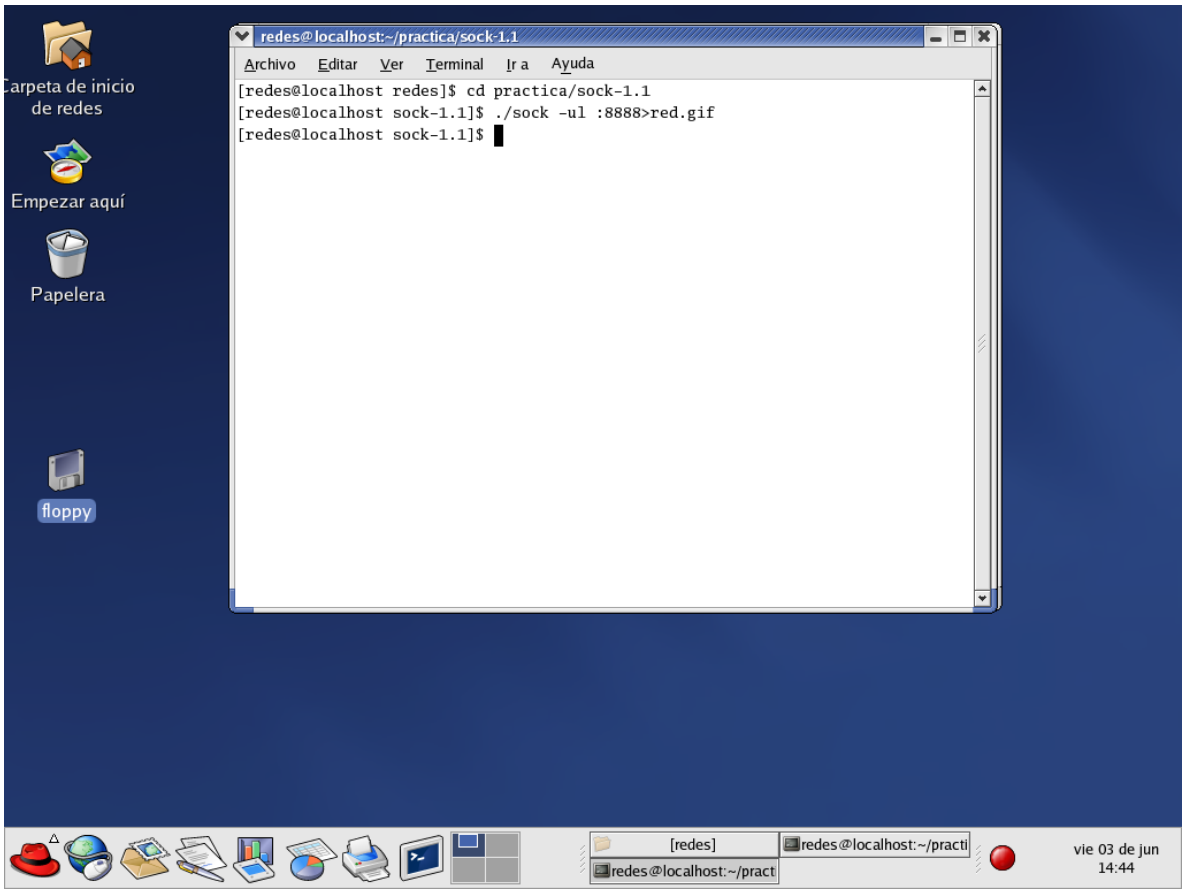


Figura No.31.

4.6.11 Compruebe que los archivos “nube.gif” y “red.gif” son iguales con la orden ls –la. (Ver figura No.32).

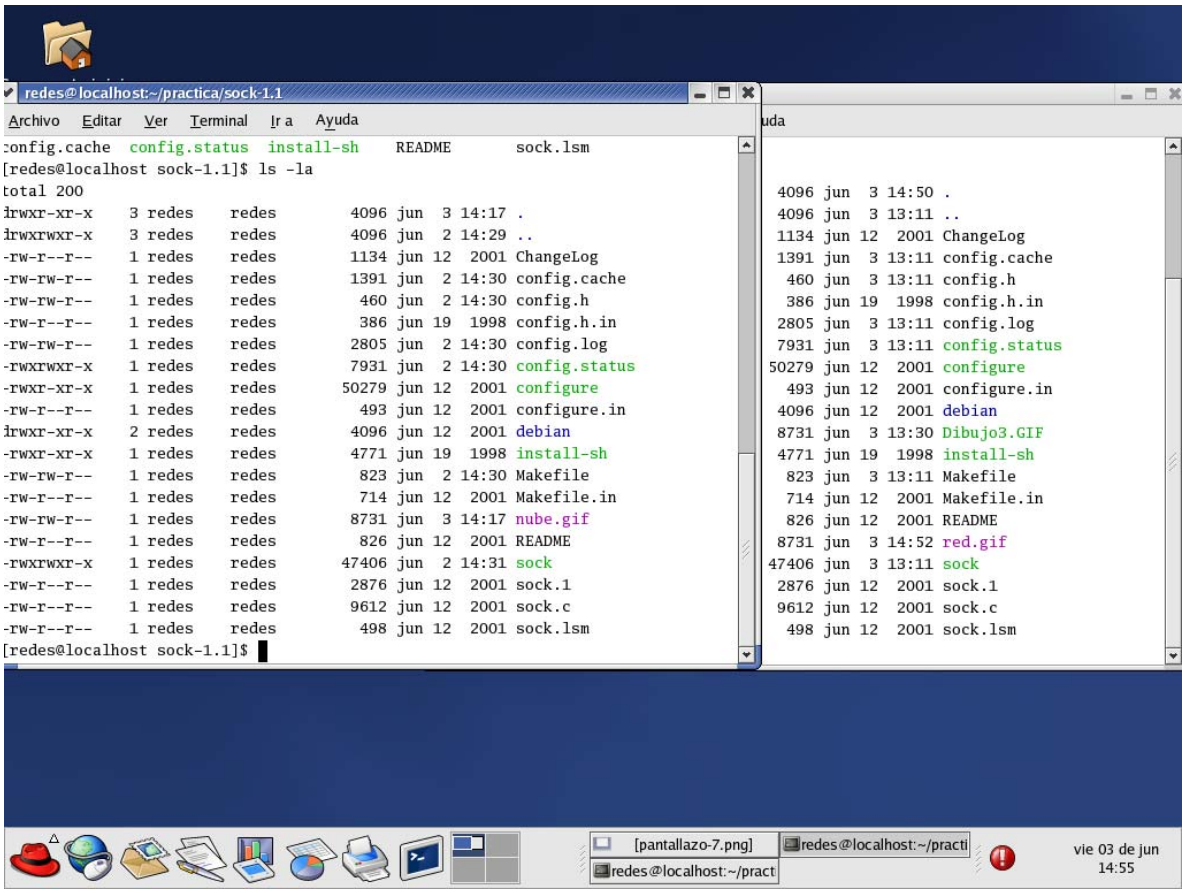


Figura No.32.

Si ambos archivos son iguales entonces podrá concluir que tanto la transmisión desde su computadora a la de al lado, empleando TCP, como la vuelta, empleando UDP, no han sufrido errores. Si repite la operación con un archivo mayor (por ejemplo el enunciado de esta práctica en pdf) encontrará que la transmisión por TCP no tiene problemas pero la de UDP fallará eventualmente, aunque es te punto no se realizará.

4.6.12 Cierre el shell que está conectado a la cuenta de la máquina de al lado. (Ver figura No.33).

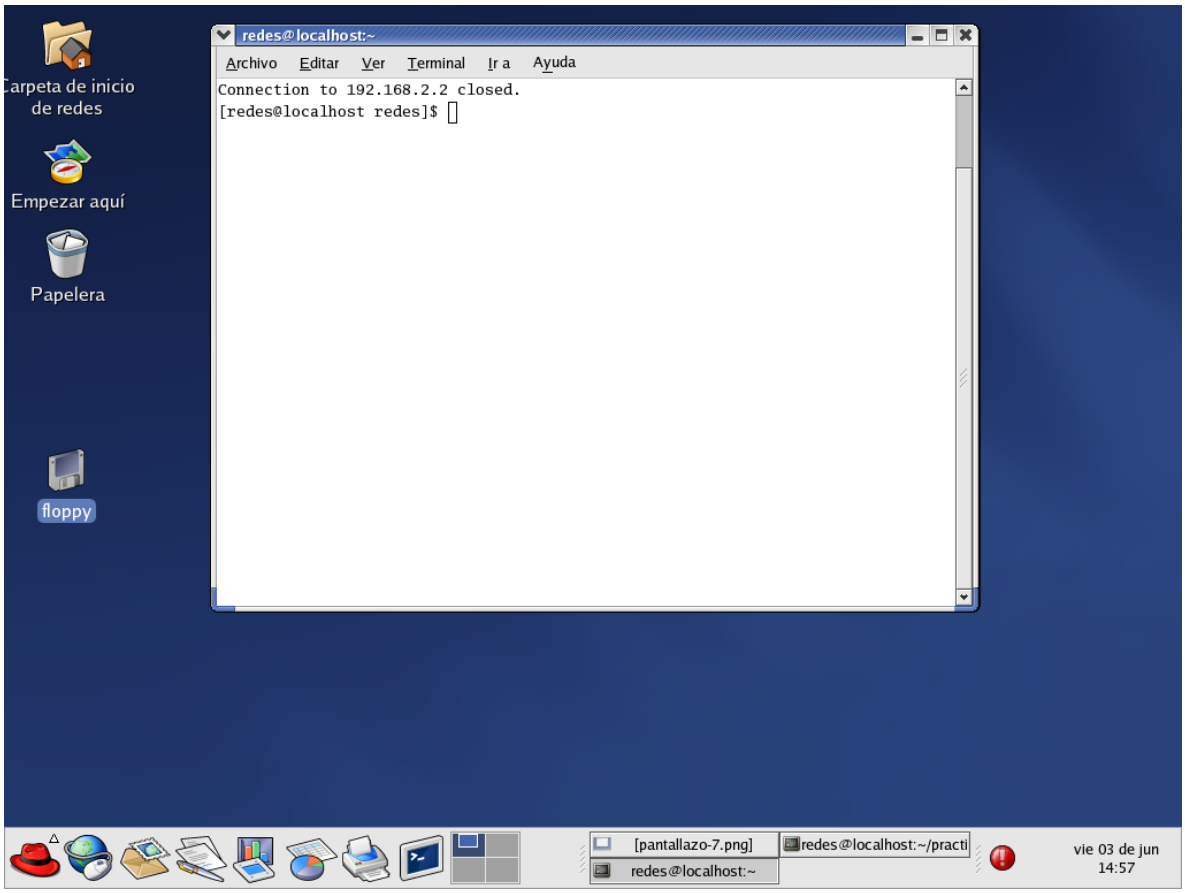


Figura No.33.

4.6.13 Borre el subdirectorio sock-1.1, el subdirectorio practica y cierre el shell (recuerde que no puede borrar un directorio si **NO** esta vació). (Ver figuras No.34 y 35).

*[redes@localhost sock-1.1]$ rm –i **

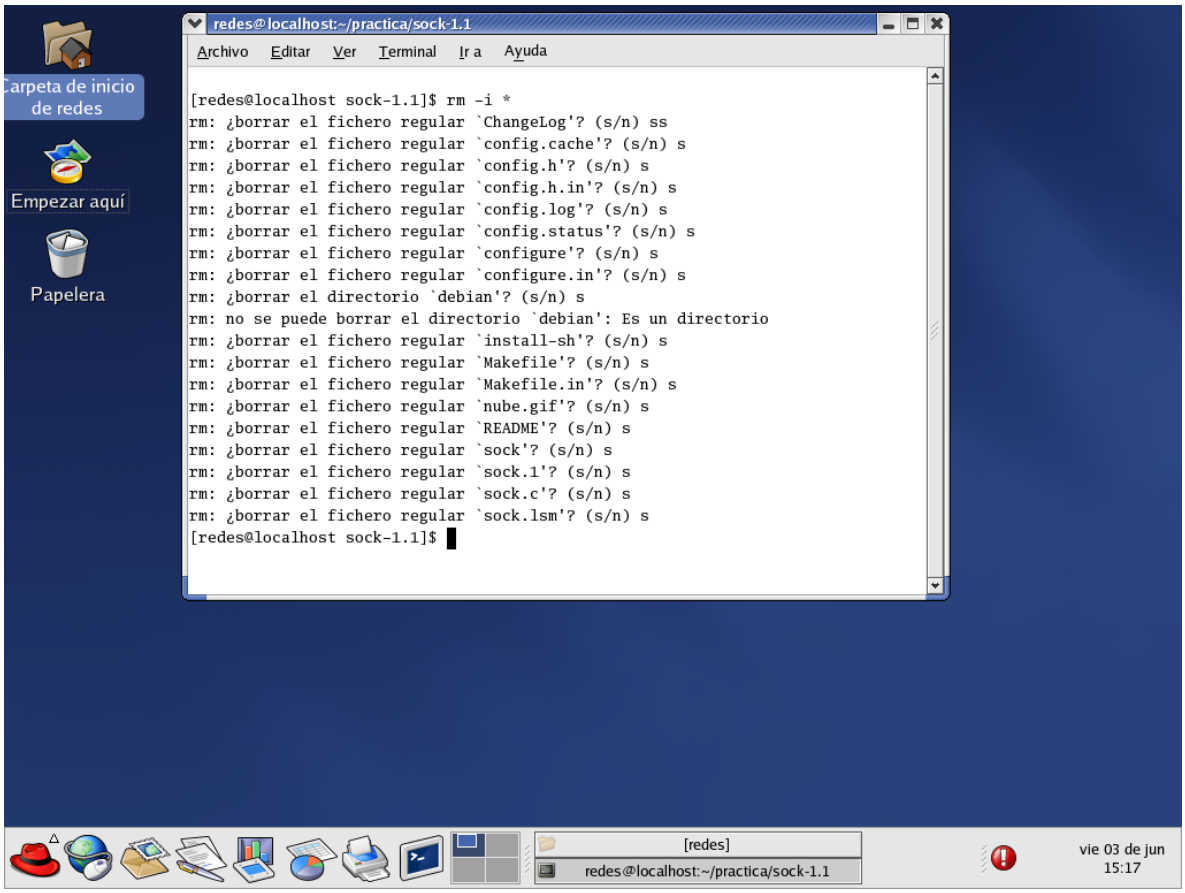


Figura No.34.

[redes@localhost sock-1.1]$ cd debian
*[redes@localhost debian]$ rm –i **
[redes@localhost debian]$ cd ..
[redes@localhost sock-1.1]$ rmdir debian
[redes@localhost sock-1.1]$ cd ..
*[redes@localhost practica]$ rm –i **
[redes@localhost practica]$ rmdir sock-1.1
[redes@localhost practica]$ cd..
[redes@localhost redes]$ rmdir practica
[redes@localhost redes]$ exit

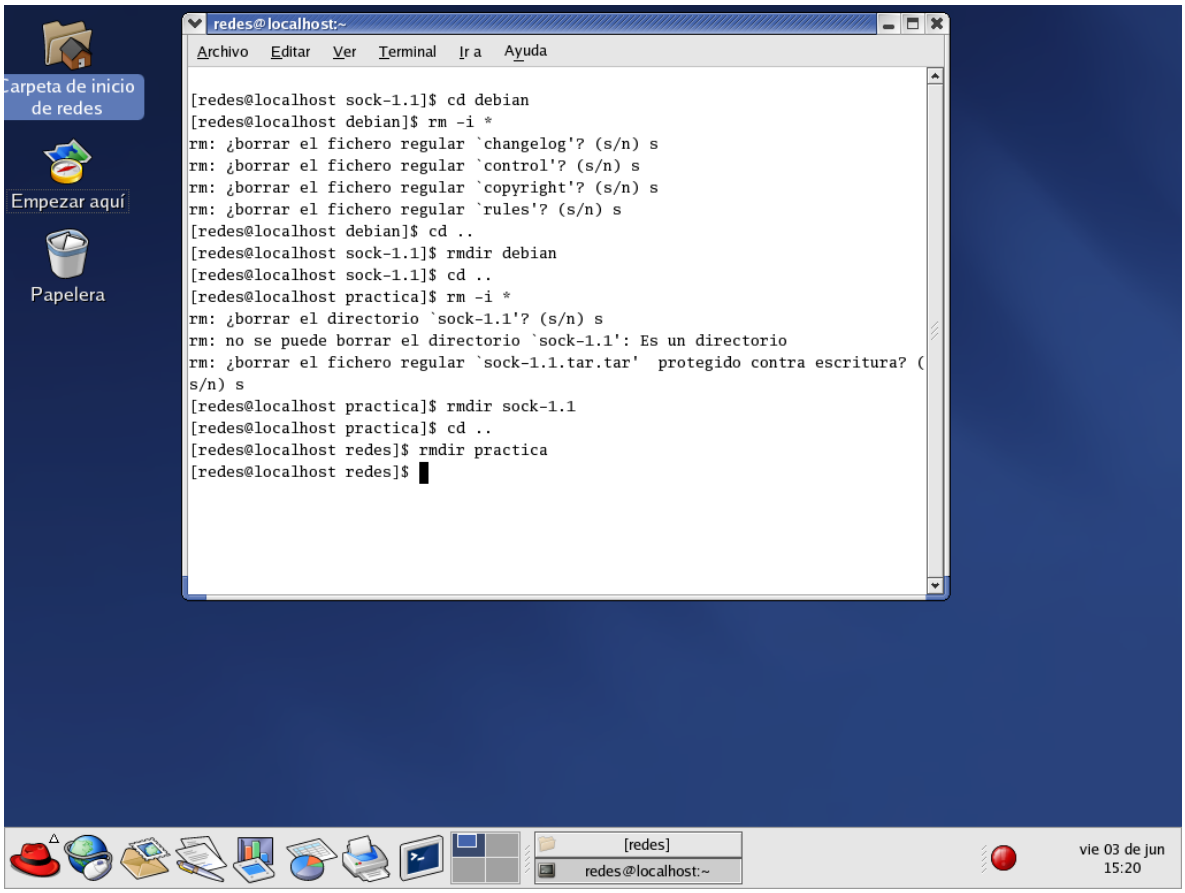


Figura No.35.

4.6.14 Cierre la sesión y apagué la computadora.

5.- Cuestionario

1. ¿Qué diferencias sustanciales existen entre TCP y UDP?

2. ¿Por qué la conexión iniciada por el socket al servidor sólo dura lo necesario para recibir la información requerida?

6.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica:

PRÁCTICA 7
TCP y UDP
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. ¿Qué es un socket?
2. Mencione algunos protocolos de transporte.
3. ¿Qué es el protocolo de transporte TCP?
4. ¿Qué es el protocolo de transporte UDP?
5. ¿Qué es una dirección IP de multicast?
6. ¿A qué nos referimos con difusión?
7. Explique qué hace el programa sock.

PRÁCTICA 8
Filtrados de Puertos TCP/UDP mediante un Firewall

1.- Objetivos de Aprendizaje
El alumno:

- Al finalizar la práctica se familiarizará con el manejo de los puertos.
- Será capaz de manejar uno de los firewalls.
- Creará filtros, permitiendo o negando el uso de puertos conocidos.
- Manipulará el uso del messenger y el navegador de Windows e Internet Explorer.
- Permitirá y negará paquetes del comando ping.

2.- Conceptos teóricos

La capa de red transfiere datagramas entre dos computadoras a través de la red utilizando como identificadores las direcciones IP. La capa de transporte añade la noción de *puerto* para distinguir entre los muchos destinos dentro de un mismo *host*. No es suficiente con indicar la dirección IP del destino, además hay que especificar la aplicación que recogerá el mensaje. Cada aplicación que esté esperando un mensaje utiliza un número de puerto distinto; más concretamente, la aplicación está a la espera de un mensaje en un puerto determinado (escuchando un puerto).

Pero no sólo se utilizan los puertos para la recepción de mensajes, también para el envío: todos los mensajes que envíe una computadora debe hacerlo a través de uno de sus puertos. La siguiente figura (Ver figura No.1), representa una transmisión entre la computadora 194.35.133.5 y la 135.22.8.165. La primera utiliza su puerto 1256 y la segunda, el 80.

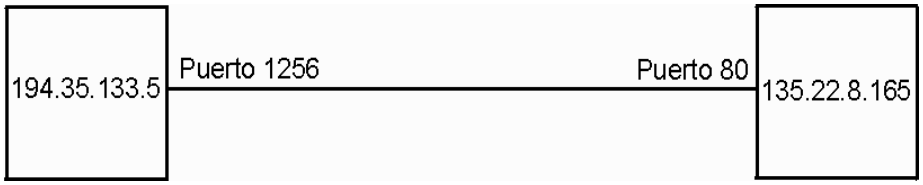


Figura No.1

La capa de transporte transmite mensajes entre las aplicaciones de dos computadoras. Por ejemplo, entre nuestro navegador de páginas web y un servidor de páginas web, o entre nuestro programa de correo electrónico y un servidor de correo. (Ver figura No.2).

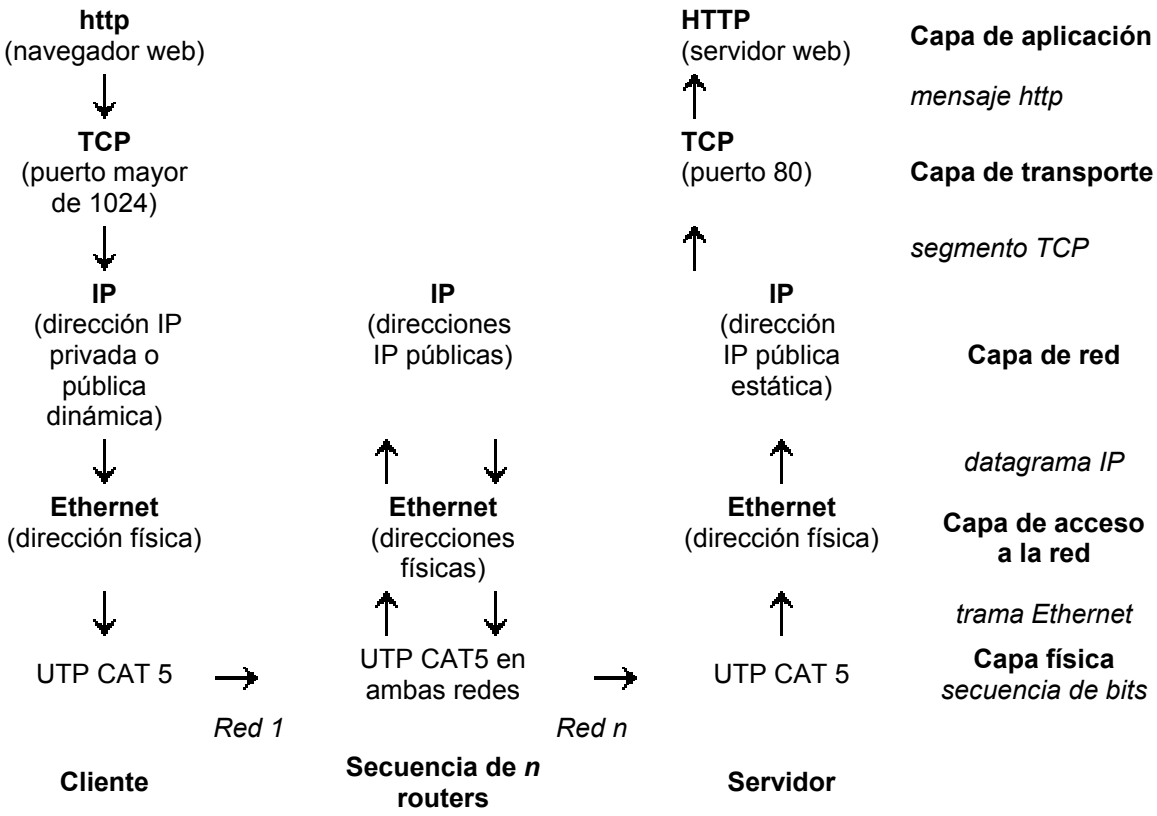


Figura No.2.

Puertos

Una computadora puede estar conectada con distintos servidores a la vez; por ejemplo, con un servidor de noticias y un

servidor de correo. Para distinguir las distintas conexiones dentro de una misma computadora se utilizan los puertos.

Un puerto es un número de 16 bits, por lo que existen 65536 puertos en cada computadora. Las aplicaciones utilizan estos puertos para recibir y transmitir mensajes.

Los números de puerto de las aplicaciones cliente son asignados dinámicamente y generalmente son superiores al 1024. Cuando una aplicación cliente quiere comunicarse con un servidor, busca un número de puerto libre y lo utiliza.

En cambio, las aplicaciones servidoras utilizan números de puerto específicos: son los llamados puertos well-known (bien conocidos). Estos puertos están definidos en la RFC 1700 y se pueden consultar en http://www.ietf.org/rfc/rfc1700.txt. A continuación se enumeran los puertos well-known más usuales: (Ver Tabla No.1).

Palabra clave	Puerto	Descripción
	0/tcp	Reserved
	0/udp	Reserved
tcpmux	1/tcp	TCP Port Service Multiplexer
Rje	5/tcp	Remote Job Entry
Echo	7/tcp/udp	Echo
Discard	9/tcp/udp	Discard
systat	11/tcp/udp	Active Users
daytime	13/tcp/udp	Daytime
qotd	17/tcp/udp	Quote of the Day
chargen	19/tcp/udp	Character Generator
ftp-data	20/tcp	File Transfer [Default Data]
ftp	**21/tcp**	**File Transfer [Control]**
telnet	**23/tcp**	**Telnet**
smtp	**25/tcp**	**Simple Mail Transfer**
time	37/tcp/udp	Time
nameserver	42/tcp/udp	Host Name Server
nicname	43/tcp/udp	Who Is
domain	**53/tcp/udp**	**Domain Name Server**
bootps	67/udp/udp	Bootstrap Protocol Server
Tftp	69/udp	Trivial File Transfer
gopher	70/tcp	Gopher
finger	79/tcp	Finger
www-http	**80/tcp**	**World Wide Web HTTP**
Dcp	93/tcp	Device Control Protocol

supdup	95/tcp	SUPDUP
hostname	101/tcp	NIC Host Name Server
Iso-tsap	102/tcp	ISO-TSAP
gppitnp	103/tcp	Genesis Point-to-Point Trans Net
rtelnet	107/tcp/udp	Remote Telnet Service
pop2	109/tcp	Post Office Protocol – Version 2
pop3	**110/tcp**	**Post Office Protocol - Version 3**
sunrpc	111/tcp/udp	SUN Remote Procedure Call
auth	113/tcp	Authentication Service
Sftp	115/tcp/udp	Simple File Transfer Protocol
nntp	**119/tcp**	**Network News Transfer Protocol**
Ntp	123/udp	Network Time Protocol
pwdgen	129/tcp	Password Generator Protocol
netbios-ns	137/tcp/udp	NETBIOS Name Service
netbios-dgm	138/tcp/udp	NETBIOS Datagram Service
netbios-ssn	**139/tcp/udp**	**NETBIOS Session Service**
snmp	161/udp	SNMP
snmptrap	162/udp	SNMPTRAP
Irc	194/tcp	Internet Relay Chat Protocol

Tabla No.1

Los puertos tienen una memoria intermedia *(buffer)* situada entre los programas de aplicación y la red. De tal forma que las aplicaciones transmiten la información a los puertos. Aquí se va almacenando hasta que pueda enviarse por la red. Una vez que pueda transmitirse, la información irá llegando al puerto destino donde se irá guardando hasta que la aplicación esté preparada para recibirla.

Firewall

Un firewall es un dispositivo que funciona como “cortafuegos” entre redes, permitiendo o denegando las transmisiones de una red a la otra. Un uso típico es situarlo entre una red local y la red Internet, como dispositivo de seguridad para evitar que los intrusos puedan acceder a información confidencial.

Un firewall es simplemente un filtro que controla todas las comunicaciones que pasan de una red a la otra y en función de lo que sean permite o deniega su paso. Para permitir o denegar una comunicación el firewall examina el tipo de servicio al que corresponde, como pueden ser el web, el correo o el IRC. Dependiendo del servicio el firewall decide si lo permite o no. Además,

el firewall examina si la comunicación es entrante o saliente y dependiendo de su dirección puede permitirla o no.

De este modo un firewall puede permitir desde una red local hacia Internet servicios de web, correo y ftp, pero no a IRC que puede ser innecesario para nuestro trabajo. También podemos configurar los accesos que se hagan desde Internet hacia la red local y podemos denegarlos todos o permitir algunos servicios como el de la web, (sí es que poseemos un servidor web y queremos que sea accesible desde Internet). Dependiendo del firewall que tengamos también podremos permitir algunos accesos a la red local desde Internet si el usuario se ha autentificado como usuario de la red local.

Un firewall puede ser un dispositivo software o hardware, es decir, un dispositivo que se conecta entre la red y el cable de la conexión a Internet, o bien un programa que se instala en la máquina que tiene el modem que conecta con Internet. Incluso podemos encontrar computadoras muy potentes y con software específico que lo único que hacen es monitorizar las comunicaciones entre redes.

Dado que el firewall puede ser situado dentro de la capa de aplicación, transporte o red dependiendo del uso que se le de, es en este sentido que hemos empleado el firewall (software) dentro de esta capa, ya que para fines de la tesis el firewall a utilizar nos permite manipular los puertos de la computadora a modo de negar o permitir el uso de estos.

Tipos de peligros que puede evitar un Firewall

- Instalación y ejecución de programas instalados clandestinamente desde Internet, por ejemplo, vía aplicaciones Active X o Java que pueden llegar a transferir datos personales del usuario a sitios.
- Acceso de terceros por fallas o errores de configuración de windows (por ejemplo de NetBIOS).
- Instalación de publicidad (advertisers) o elementos de seguimiento (track) como las cookies.

- Troyanos: Aplicaciones ocultas que se descargan de la red y que pueden ser usadas por terceros para extraer datos personales. A diferencia de los virus, estos troyanos son activados en forma remota por un tercero.
- Reducción del ancho de banda disponible por el tráfico de *banners,* sitios no solicitados y otro tipo de datos innecesarios que vuelven lenta la conexión.
- *Spyware*.
- Utilización de la línea telefónica por terceros por medio de *Dialers* (programas que cortan la actual conexión y utilizan la línea para llamadas de larga distancia).

3.- Equipo y material necesario

Material que deberá traer el alumno:

A. Lápiz ó pluma.

Equipo del Laboratorio:

B. 1 Computadora con Sistema Operativo Windows 98/NT/2000/XP ó superior.
C. Firewall Personal Kerio v4 ó superior.

4.- Desarrollo

4.1 Modo de trabajar

Se trabajará en equipos de dos personas por computadora.

4.2 Realización de la práctica

4.2.1 Encienda el sistema y elija la opción de cargar *Windows*.

4.3 Familiarizándose con Kerio.

La Interfaz Gráfica de Usuario de Kerio Personal Firewall es ejecutada automáticamente por el motor del cortafuegos cada vez que éste se ejecuta o detecta que el GUI no está funcionando. Cuando está en marcha el GUI está representado por un icono en forma de escudo en la Bandeja del Sistema, como se muestra en la siguiente figura No.3:



Figura No.3.

4.3.1 Haga clic con el botón derecho del ratón sobre este icono para abrir el diálogo de configuración. (Ver figura No.4).



Figura No.4.

- **Disable Firewall**.- Esta opción desactiva el cortafuegos.
- **Stop all traffic**.- Esta opción bloquea todo el tráfico en el ordenador en el cual Kerio Personal Firewall está instalado.
- **Register**.- Muestra la ventana para registrar el programa.
- **About**.- Muestra información sobre la versión del programa y sus elementos, así como el tiempo de expiración de la versión trial y tipo de licencia.
- **Exit**.- Esta opción cierra el programa.
- **Configuration.-** Abre la ventana de configuración de Kerio Personal Firewall.

4.3.2 De un clic en la opción Configuration, abrirá la sección Overview por defecto.

4.3.3 Seleccione la pestaña Connections, aparecerá la siguiente figura: (Ver figura No.5).

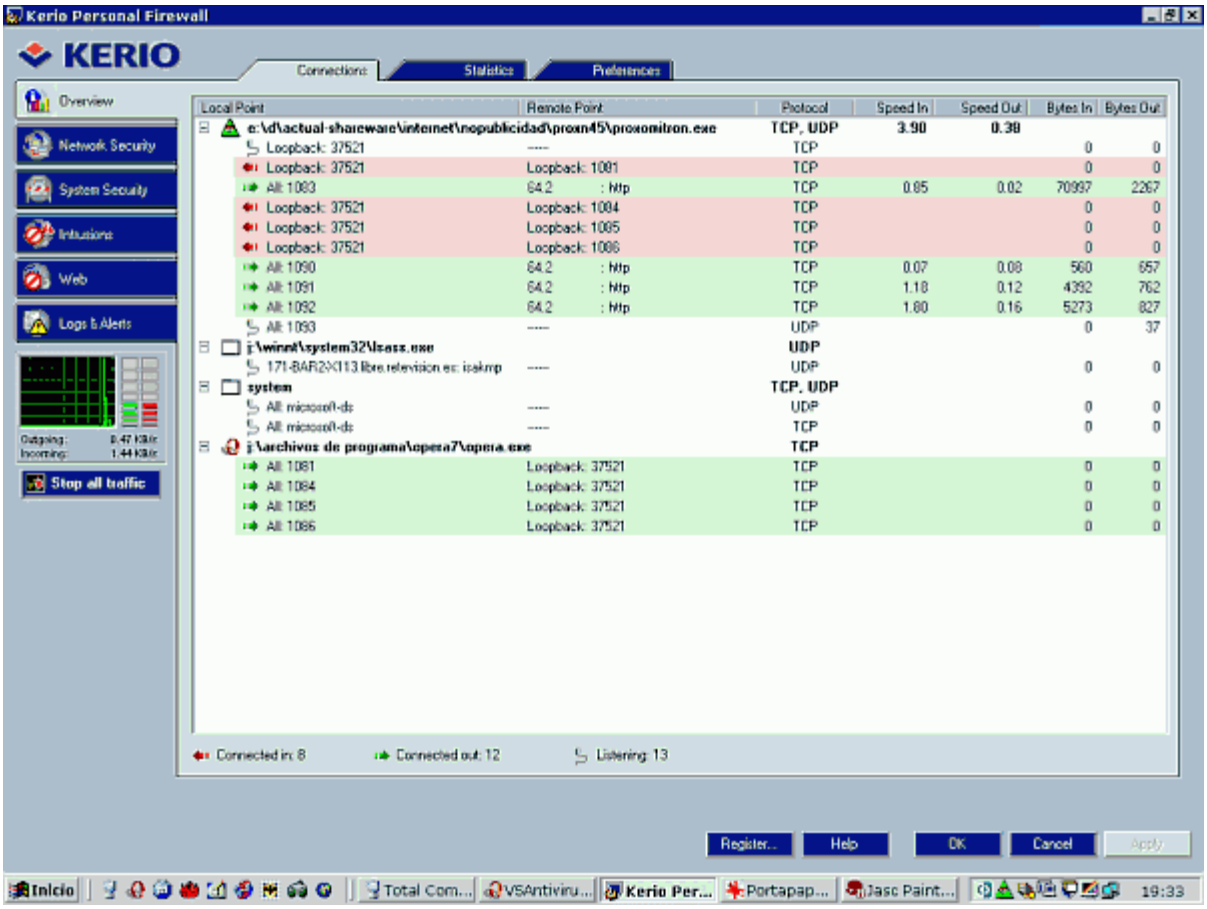


Figura No. 5.

Connections.- Proporciona información de las conexiones establecidas y puertos abiertos por cada aplicación.

4.3.4 Seleccione la pestaña Statistics, aparecerá la siguiente pantalla:(Ver figura No.6).

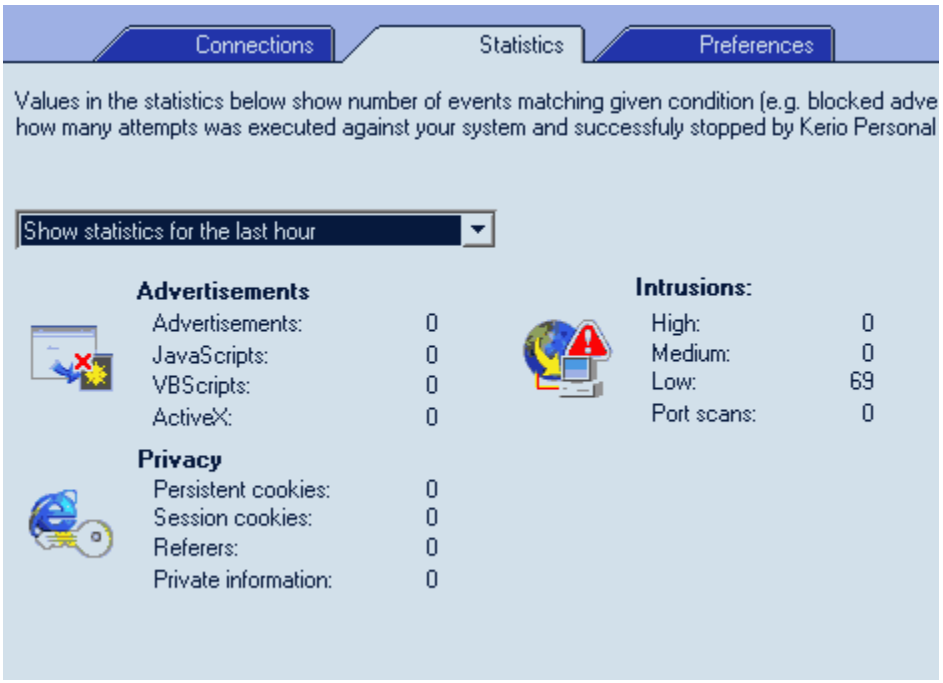


Figura No.6.

Statistics.- Informa al usuario del número de objetos bloqueados por el filtro de contenidos Web y el número de intentos de intrusión detectados para un cierto periodo de tiempo.

4.3.5 De clic en la pestaña Preferences, se mostrará la siguiente pantalla: (Ver figura No.7).

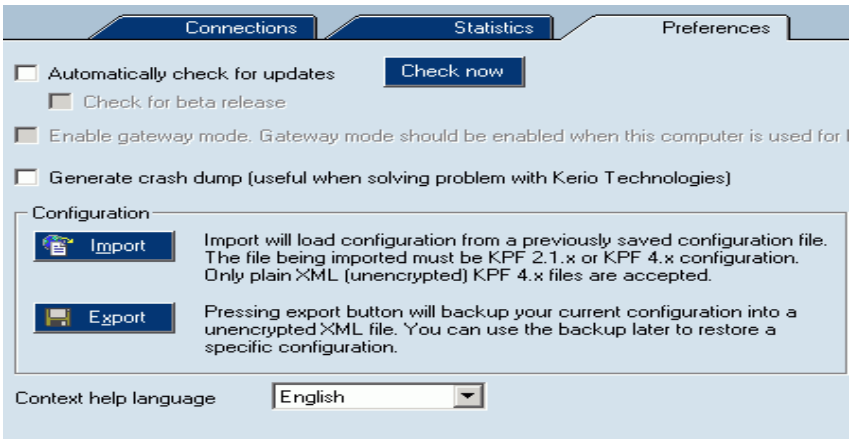


Figura No.7.

Preferences.- Permite importar/exportar configuraciones del cortafuegos, además de otras opciones.

4.3.6 Abra el módulo Network Security dando un clic sobre éste. (Ver figura No.8).

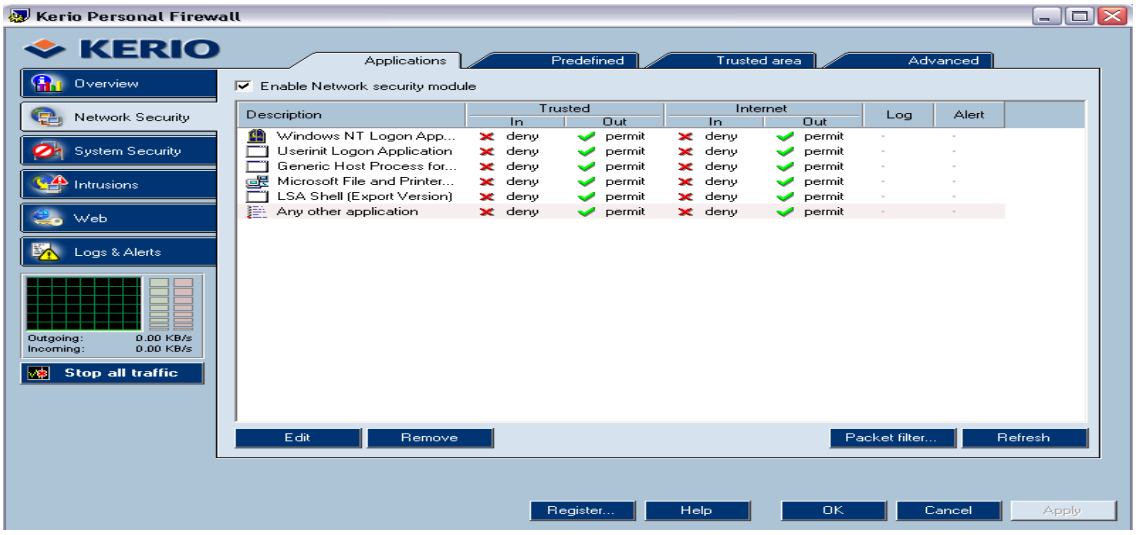


Figura No.8

En el momento que Kerio Personal Firewall detecta tráfico que no corresponde con ninguna regla, el usuario será preguntado para permitir o denegar la comunicación. Opcionalmente, se puede crear la correspondiente regla para la aplicación o comunicación teniendo en cuenta esta decisión.

- **Application rules**.- Es posible negar/permitir comunicaciones de red para cada aplicación o fijar que Kerio Personal Firewall pregunte al usuario.
- **Packet filter rules**.- Para definir reglas de paquetes avanzadas para el tráfico de red (especificación de direcciones IP, protocolos, puertos, etc.). Estas reglas pueden crearse especificando tanto una aplicación en particular como una regla general para cualquier aplicación (any applicaction).
- **Predefined**.- Kerio Personal Firewall incluye por defecto un grupo de reglas de red (p.e. para DNS, DHCP, etc.). Estas reglas están separadas de las reglas definidas por el usuario y pueden ser activadas o desactivadas.

4.3.7 De clic sobre el módulo System Security. (Ver figura No.9).

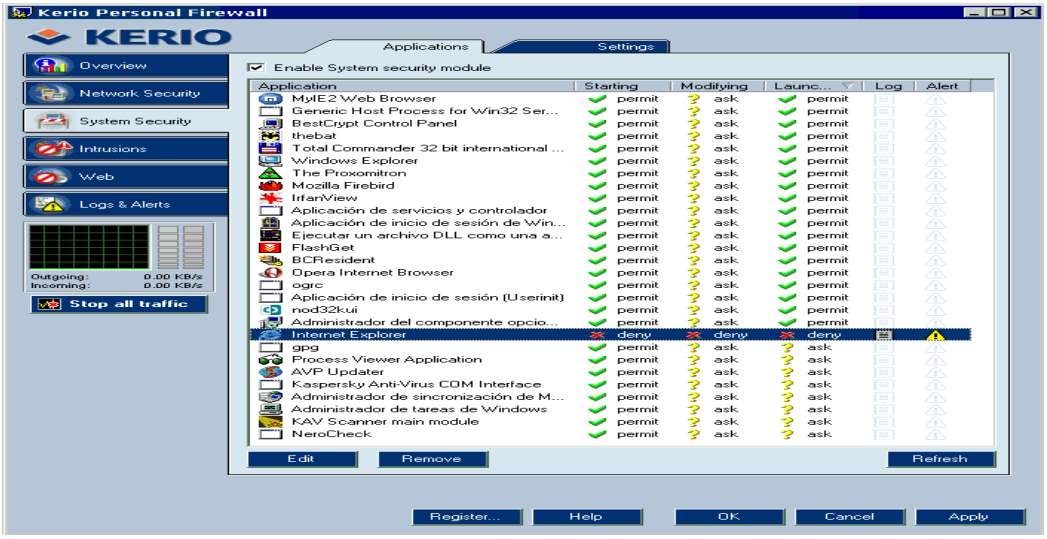


Figura No.9.

El módulo System Security controla las aplicaciones que se ejecutan en el sistema operativo. Los siguientes eventos son controlados:

- Ejecución de aplicaciones.
- Reemplazamiento de la aplicación desde la última vez que esta se ejecutó.
- Ejecución de una aplicación por otra aplicación.

Como en el caso del tráfico de red, se pueden definir reglas para cada aplicación de manera individual. Estas reglas pueden negar o permitir el evento o preguntar al usuario. Si el evento no corresponde con ninguna regla, Kerio Personal Firewall preguntará automáticamente al usuario para permitir o denegar el evento.

En caso de infección, el cortafuegos puede resultar más fiable que un antivirus (sí el virus es nuevo y no está incluido en la base de virus, el antivirus podría no detectarlo; Kerio Personal Firewall detectará el reemplazamiento de cualquier ejecutable y avisará al usuario).

4.3.8 De clic en el módulo Intrusión Detection. (Ver figura No.10).

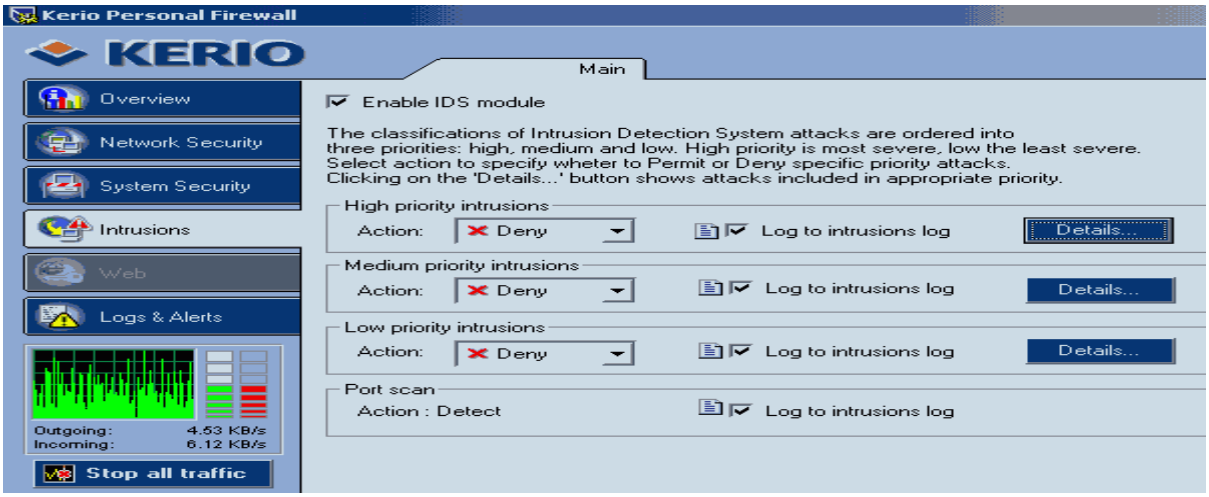


Figura No.10.

Intrusión Detection System (IDS) puede distinguir, bloquear y loguear tipos de intrusión conocidos. Para este propósito Kerio Personal Firewall usa una base de datos de tipos de intrusión conocidas. Esta base de datos es actualizada regularmente.

4.3.9 Abra el modulo Web Content Filtering. (Ver figura No.11).



Figura No.11.

Este módulo activa las siguientes funciones:

- Boqueo de publicidad, scripts y otros objetos Web.
- Bloqueo de ventanas emergentes (pop-up windows).
- Bloqueo de scripts (JavaScript, VB Script).
- Protección de cookies no deseadas y revelación de datos personales en formularios de aplicaciones Web.

4.4 Configuración de Kerio Personal Firewall

Kerio Personal Firewall es una aplicación que protege las computadoras personales de ataques externos (típicamente desde Internet), bichos (virus, gusanos, troyanos, etc.) y de las propias aplicaciones que se ejecutan en nuestro PC.

Es por ello importante definir nuestras áreas de confianza, es decir una dirección IP en concreto de nuestra subred o un rango de direcciones IP's de estas. Para ello definiremos el rango de direcciones IP's de nuestra subred.

4.4.1 De clic en la pestaña Trusted Área de la sección Network Security. (Ver figura No.12).

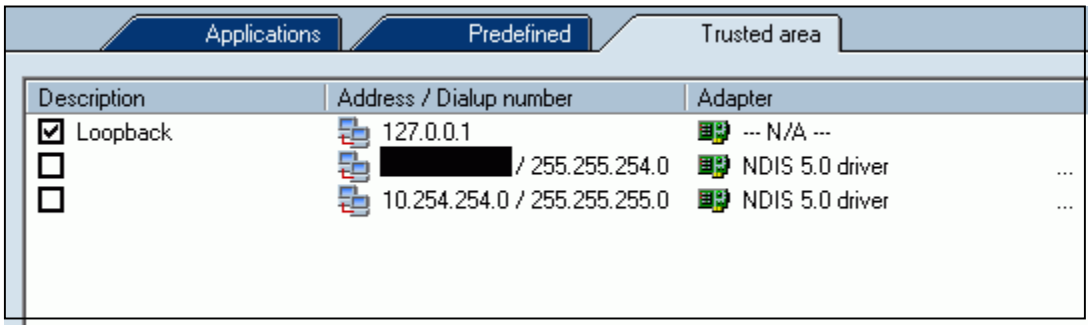


Figura No.12.

4.4.2 Oprima la opción Add, aparecerá la siguiente figura (Ver figura No.13). Llene el recuadro descripción con la leyenda Lab Redes, en la parte de Address type seleccione la opción IP address/ range, coloque la dirección IP 192.168.2.2 en First IP address y la dirección IP 192.168.2.254 en Last IP address. Por último oprima ok y después el botón Apply.

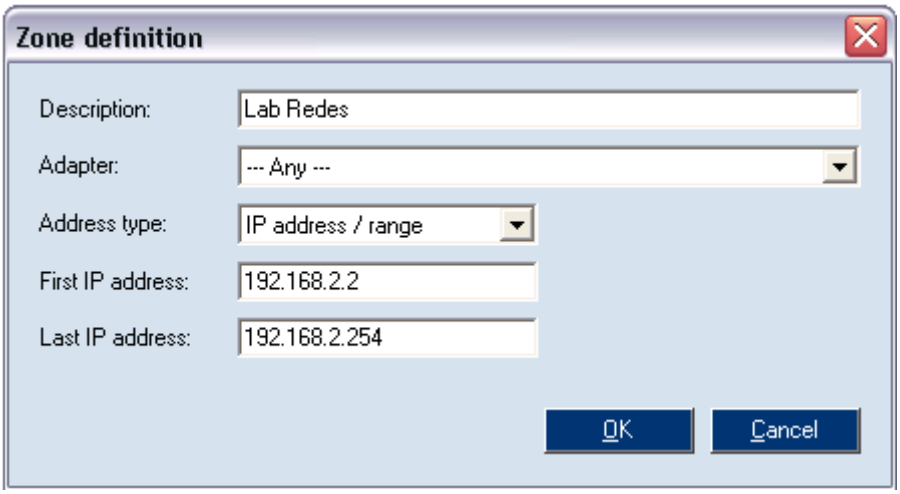


Figura No.13.

Ya definió en primera instancia el rango de direcciones IP's de confianza, ahora creará algunas reglas de filtrado a emplear en esta práctica para la subred del laboratorio.

4.5 Reglas de filtrado para el messenger

Con este par de reglas hará que el messenger inicie sesión pero impedirá que nadie, ni siquiera usted mismo pueda enviar o recibir archivos y fotos. Es decir, solo chatear, correo y se acabo, ni *netmeeting* a través de messenger, ni conferencias de voz.

Regla para permitir al messenger abrir sesion, chating y mailing (Regla 1).

4.5.1 De clic en el botón Packet filter desde la pestaña Aplications en la sección Network Security. (Ver figura No.14).

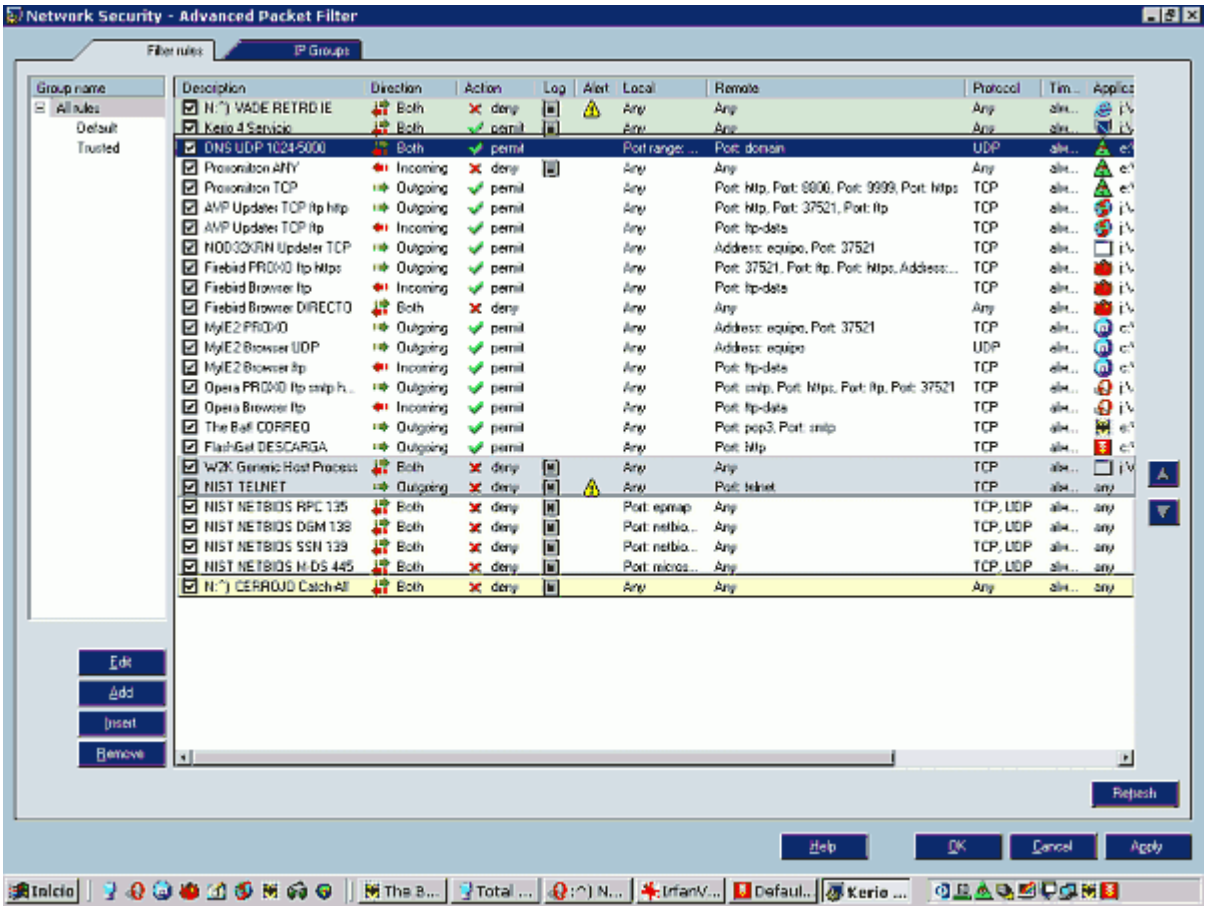


Figura No.14.

4.5.2 Oprima el botón Add, se abre el diálogo para la definición del filtro avanzado. (Ver figura No.15).

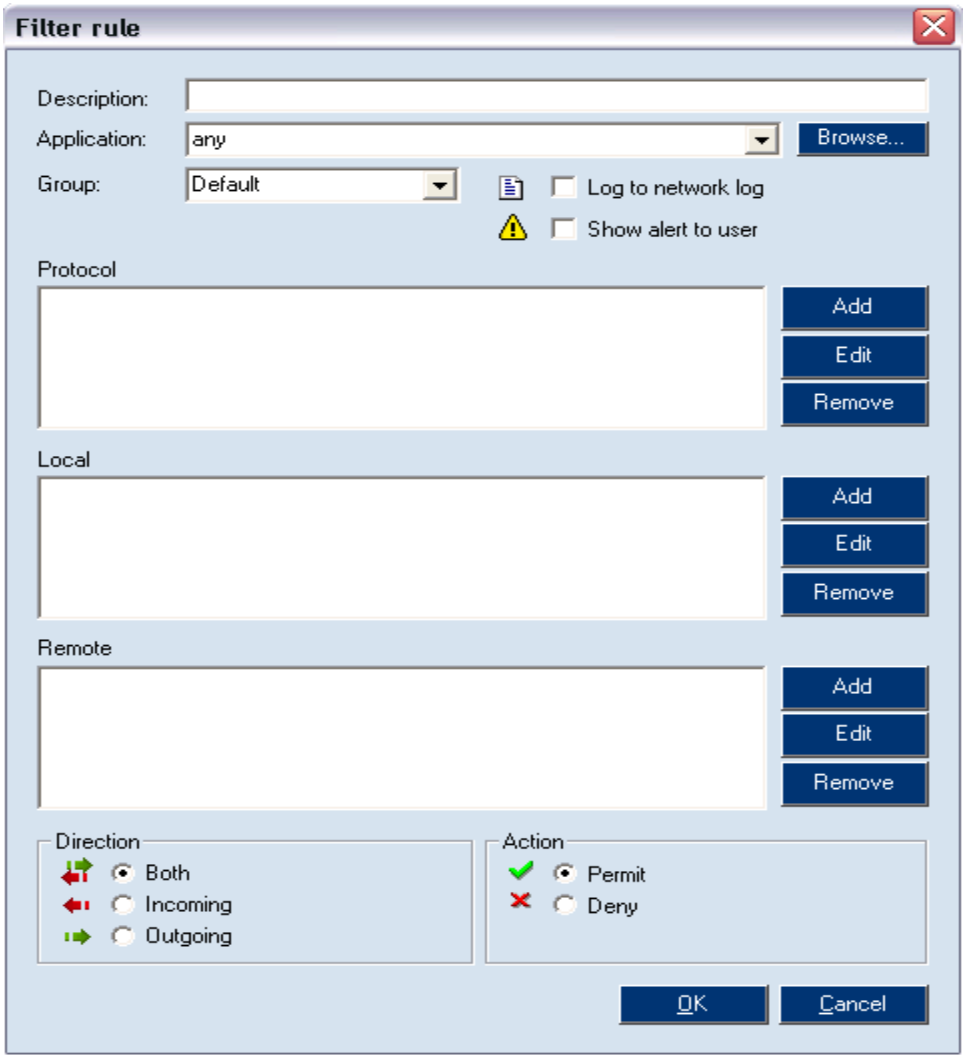


Figura No.15.

4.5.3 En el recuadro Description coloque la leyenda Filtro 1 Messenger. (Ver figura No.16).



Figura No.16.

4.5.4 En el recuadro Application mediante el botón Browse elija el archivo msmsgs.exe de su disco duro.
C:\Program Files\Messenger\msmsgs.exe
(Ver figura No.17).



Figura No.17.

4.5.5 En Group y las dos casillas frente a éste, déjelas como están. (Ver figuras No.18 y No.18a).



Figuras No.18 y 18ª

4.5.6 En la sección Protocolo elija la casilla Add, para abrir el diálogo para seleccionar el protocolo. Seleccione el protocolo TCP en el campo Name y oprima ok. (Ver figuras No.19 y 20).

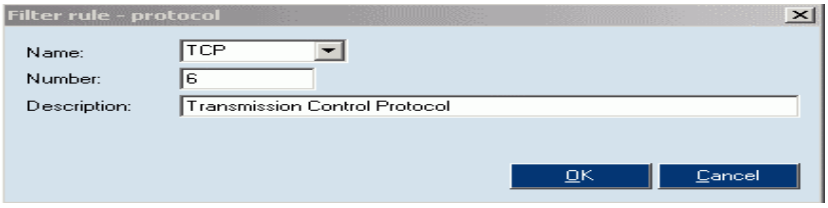


Figura No.19.



Figura No.20.

4.5.7 En la sección Local oprima la casilla Add y elija la opción Add port range, en la parte First port en el campo Number coloque 1030 y en Last port en el mismo campo 14000. Oprima ok. (Ver figuras No.21 y 22).

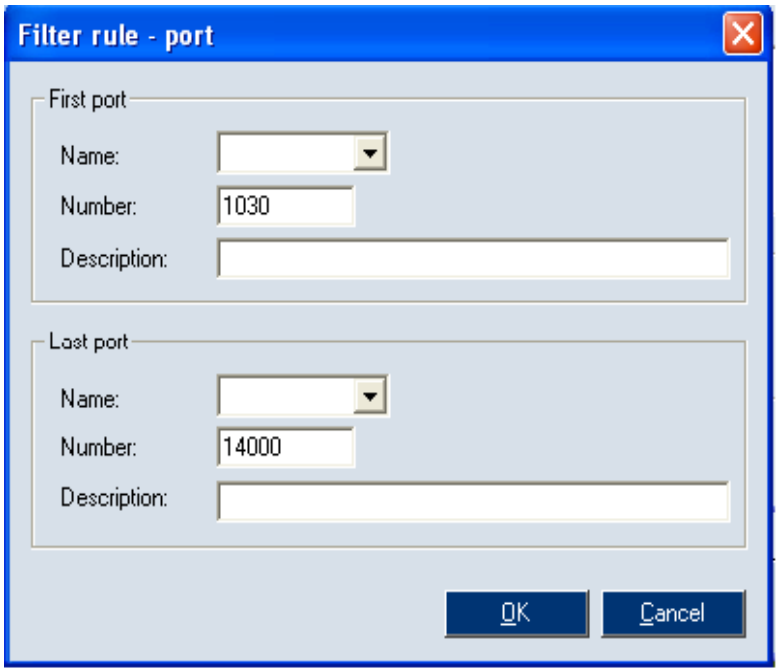


Figura No.21.



Figura No.22.

4.5.8 En la sección Remote oprima la casilla Add y elija la opción Add port, en el campo Number coloque 1863 y oprima ok. (Ver figuras No.23 y 23a).

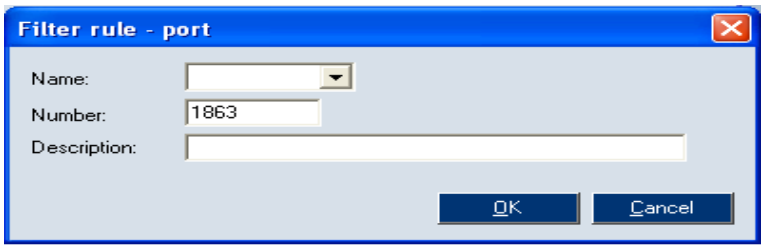


Figura No.23.



Figura No.23a.

4.5.9 Nuevamente oprima Add en la sección Remote y elija la opción Add address range, en el campo First IP address coloque 64.4.12.1 y en el campo Last IP address 64.4.13.254. Oprima ok. (Ver figuras No.24 y 24a).

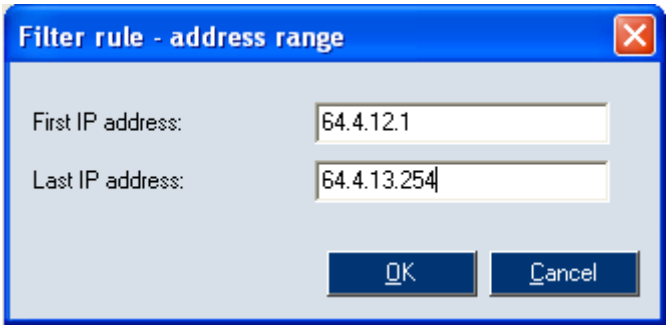


Figura No.24.

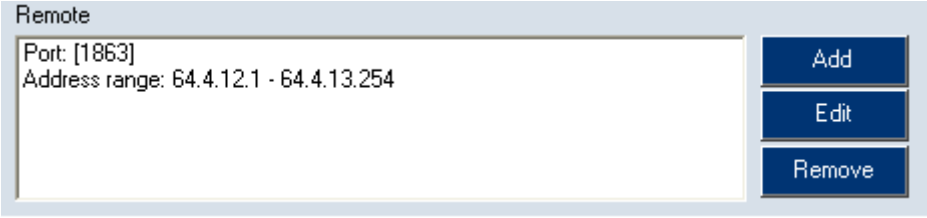


Figura No.24a.

4.5.10 En la sección Direction seleccione la opción Outgoing y en la sección Action la opción Permit. (Ver figura No.25).

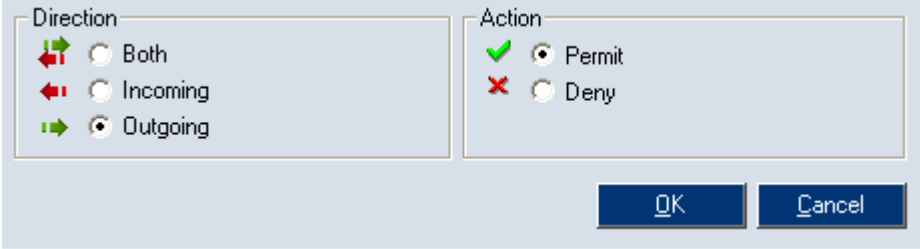


Figura No. 25.

4.5.11 La pantalla final deberá quedar así. Por último oprima ok y la regla se creará. (Ver figura No.26).

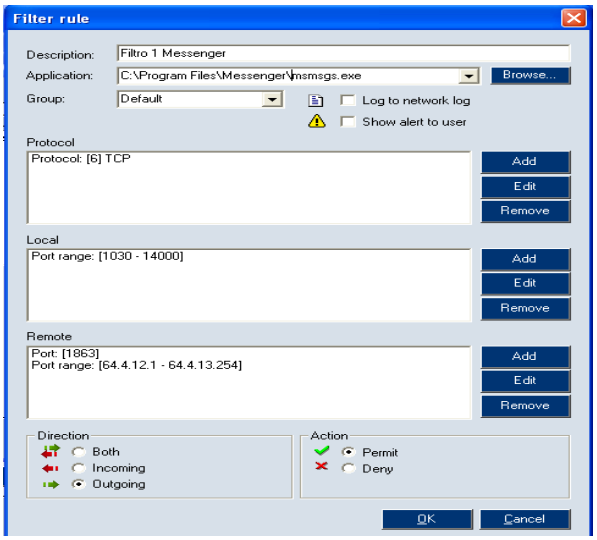


Figura No.26

4.5.12 Oprima el botón Apply para que la regla aplique.

Para que Kerio sepa que no puede hacer otra cosa más que solamente lo especificado en la regla anterior, cree una nueva regla genérica que denegará cualquier intento de conexión de messenger o de alguien, hacia su propio messenger.

Regla para que messenger quede aislado a excepción de iniciar sesión y punto (Regla 2).

4.5.13 Realice los pasos necesarios para obtener el siguiente filtro. (Ver figura No.27).

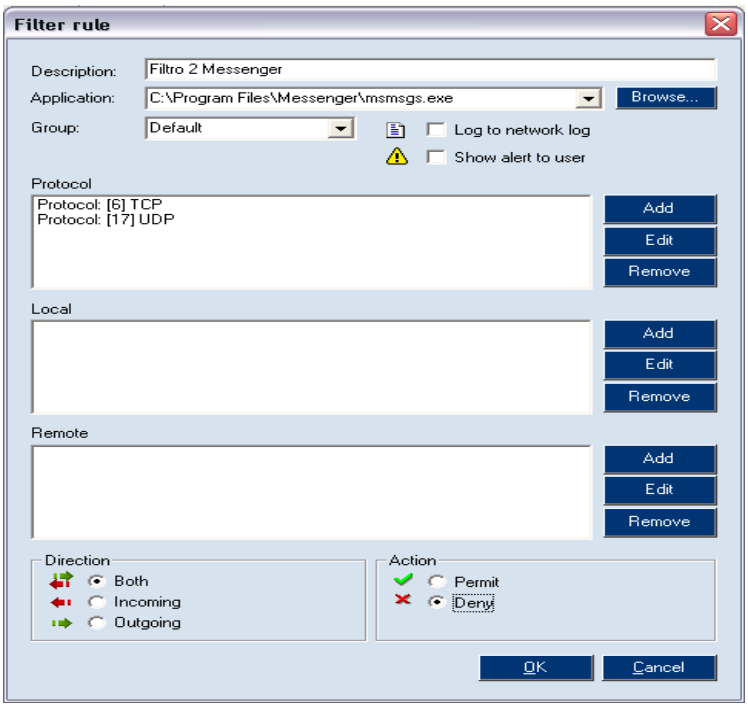


Figura No.27.

Con esta regla, al quedar situada debajo de la anterior, impedirá que aparte de iniciar sesión como quedó especificado anteriormente, messenger no pueda hacer nada. Al elegir el TCP más el UDP en ambos sentidos, hacia y desde cualquier puerto o IP remota ha logrado aislar el programa, el cual solo puede conectar a una IP y puerto.

Nota: Si los campos Protocol, Local y Remote, se dejan en blanco el filtro se aplicará para todos los protocolos, todas los puertos locales y todos los puertos y direcciones IP remotas, respectivamente. Todo filtro específico debe ir colocado arriba de cualquier genérico de lo contrario, de nada servirá dicho filtro.

4.5.14 Desde la Bandeja del Sistema de clic derecho sobre el icono del messenger y elija la opción iniciar sesión.(Ver figura No.28):



Figura No.28

4.5.15 Inicie sesión con la cuenta proporcionada por el administrador.

4.5.16 Trate de enviar un archivo a través del messenger a la cuenta de la computadora de al lado.

Comente lo ocurrido.

_____
_____
_____
_____
_____
_____
_____
_____

Ahora nos interesa que la computadora de al lado pueda enviar archivos a su computadora. Para ello hará otra regla que deberá quedar por encima siempre de la genérica (la anterior), no importa si queda en primer lugar o en segundo pero jamás lógicamente, por debajo de la genérica, pues quedaría nula. Para ello utilice las flechas de dirección en la pantalla Filter Rules.

Regla para que el messenger solo pueda conectar con la máquina de al lado (Regla 3).

4.5.17 Realice los pasos necesarios para obtener el siguiente filtro. (Ver figura No.29).

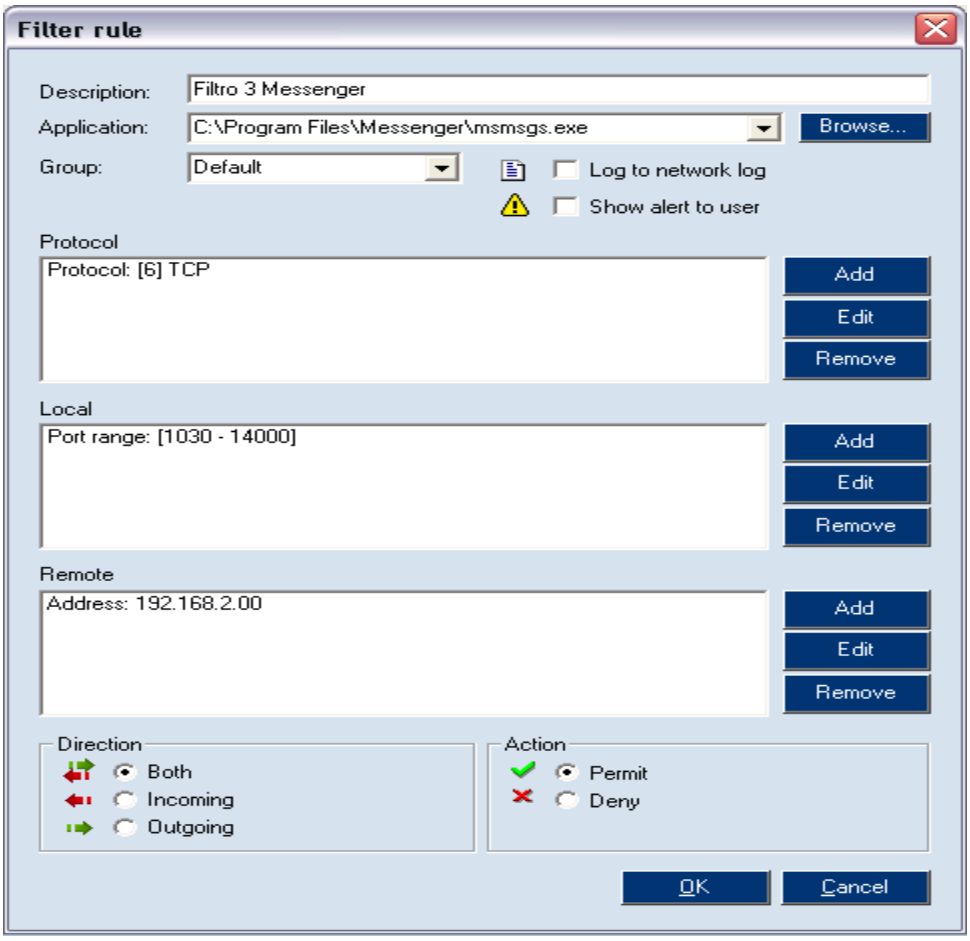


Figura No.29.

Con este filtro permite que messenger pueda establecer conexión por el protocolo TCP a la IP 00 (cambiar 00 por número de host de la máquina de al lado), desde un puerto local comprendido entre el 1030 y el 1400 a cualquier puerto de la máquina de al lado.

4.5.18 Trate nuevamente de enviar un archivo a través del messenger a la computadora de al lado.

¿Qué sucede? Coméntelo

_____

_____

_____

_____

_____
_____
_____
_____

4.5.19 Cierre la sesión del messenger desde la Bandeja del Sistema dando clic derecho sobre el icono del messenger y eligiendo la opción cerrar sesión.

4.5.20 Ahora desde la pestaña Filter rules con la ayuda del botón Remove, borre los tres filtros que acaba de realizar.

4.6 Reglas de filtrado para el Internet Explorer

4.6.1 Abra el explorador de Internet y vea si existe comunicación con la página google.com. Y ciérrelo.

¿Existió o no comunicación con google? Coméntelo.

_____
_____
_____
_____
_____
_____
_____

4.6.2 Realice el siguiente filtro. (Ver figura No.30).

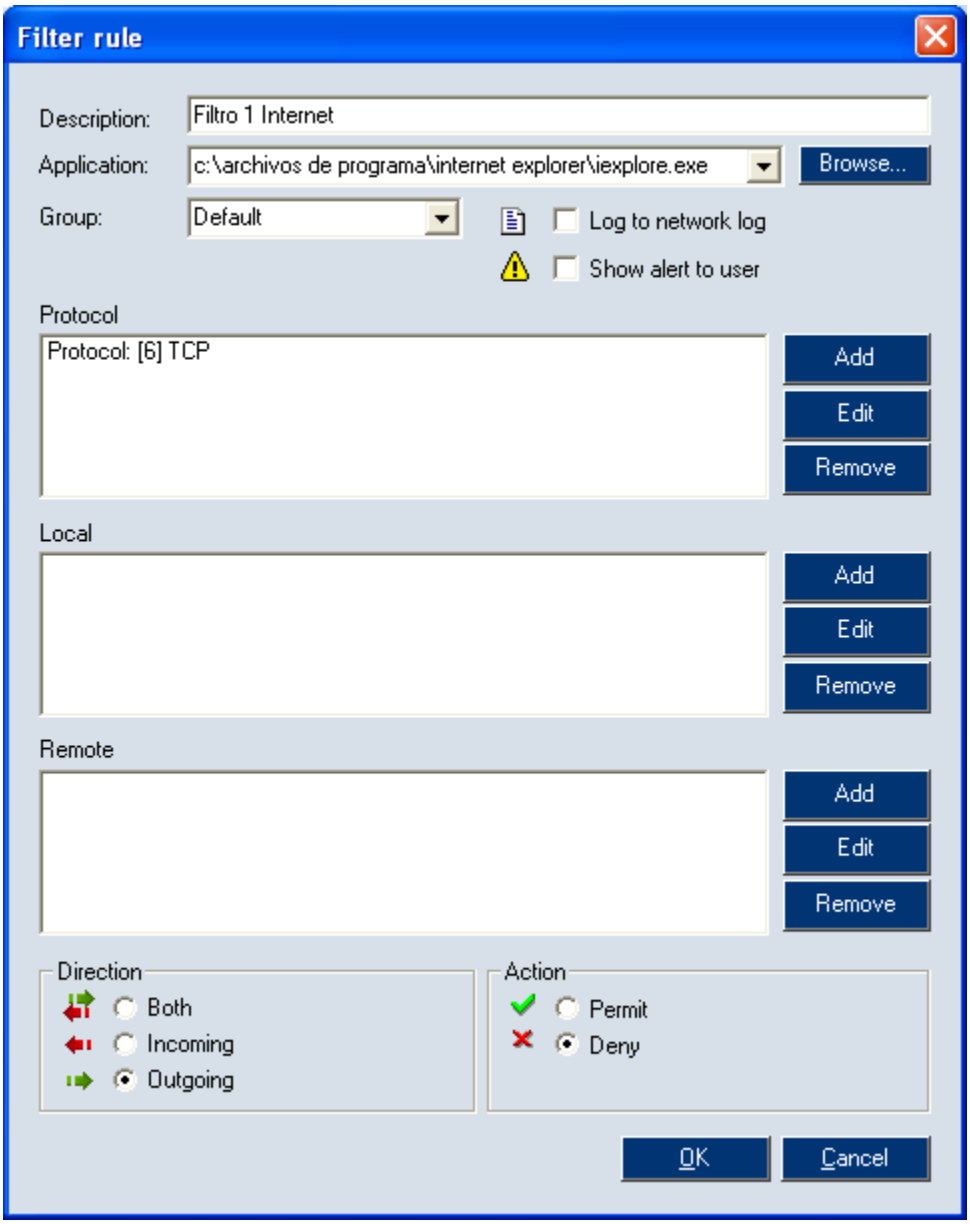


Figura No.30

Nuevamente abra el explorador de Internet y trate de tener comunicación con google.com. Y ciérrelo.

4.6.3 Realice este segundo filtro para el explorador de Internet. (Ver figura No.31).

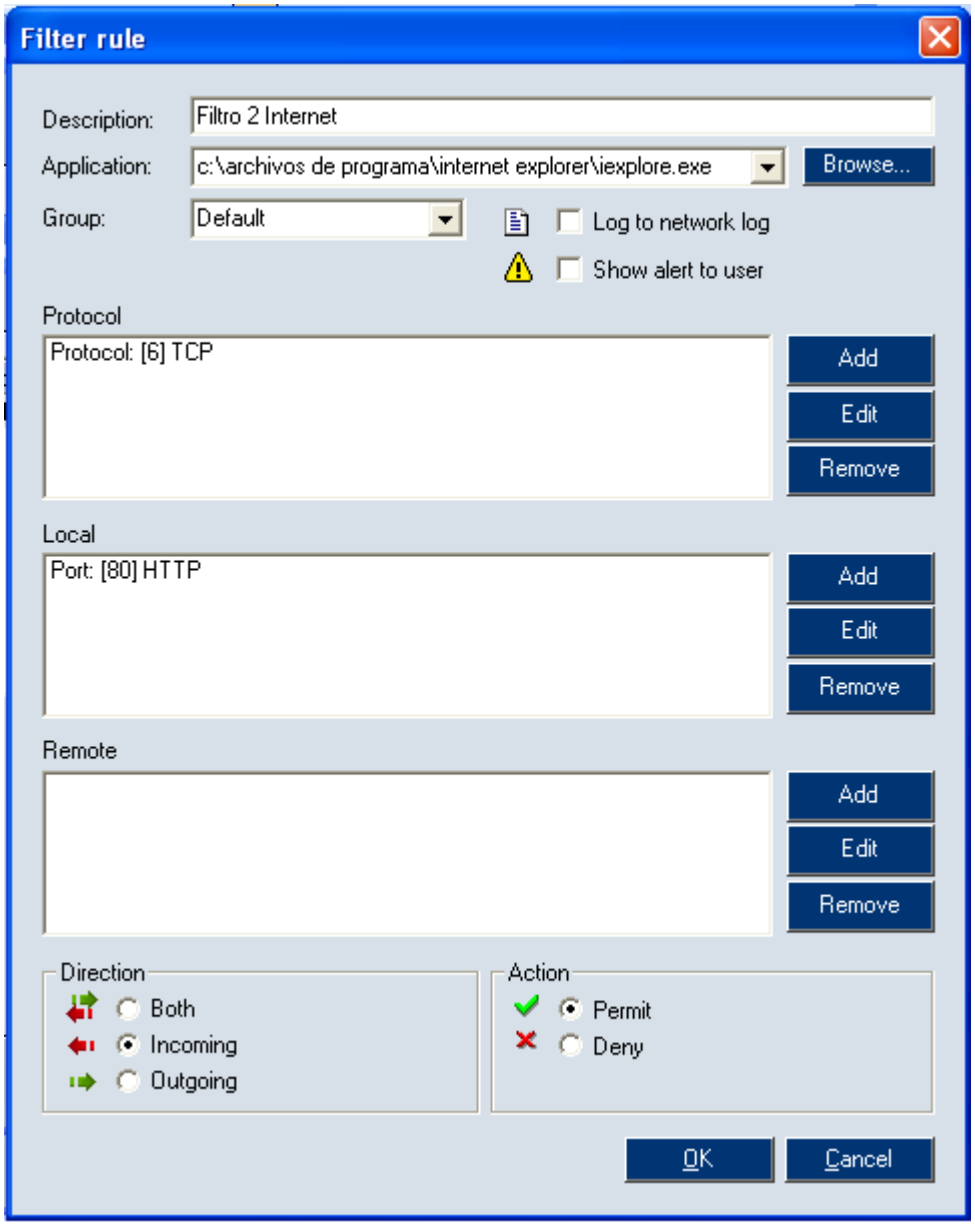


Figura No.31

Nota: Recuerde que las reglas genéricas deben ir por debajo de las específicas.

4.6.4 Abra el explorador de Internet e intente tener comunicación con google.com.

¿Hay comunicación con el sitio requerido? ¿Por qué? Comente que hacen los dos filtros anteriores.

4.6.5 Cierre el explorador de Internet y borre los dos filtros anteriores.

4.7 Filtros para el protocolo ICMP

El protocolo ICMP sirve para comunicaciones de comprobación entre varios host en la red. Se suele usar mucho para establecer distancias, tiempos y todo tipo de comprobación entre dos host. En esta práctica solo se tendrá presente del protocolo ICMP, el echo request y el echo reply. Conocidos como el ping in y el ping out, respectivamente.

4.7.1 Desde la consola de comandos de Windows realice un ping a la computadora de al lado.

4.7.2 Seleccione la pestaña Predefined del módulo Network Security. Aparecerá la siguiente figura. (Ver figura No.32).

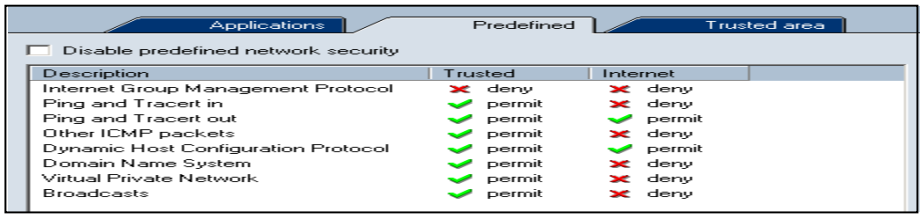


Figura No.32.

4.7.3 De un clic sobre la sección Trusted del Ping and Tracert in, cambiará el permiso de permit a deny. Oprima Apply.

4.7.4 Realice nuevamente un ping a la computadora de al lado.

4.7.5 Vuelva a permitir el Ping and Tracert in de la sección Trusted y niege el Ping and Tracert out de la misma sección. Se verá algo similar a la siguiente figura. (Ver figura No.33).

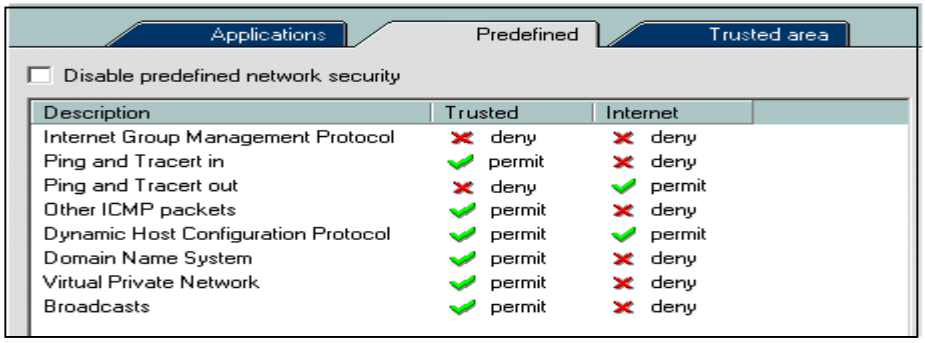


Figura No.33

4.7.6 Oprima ok para que se aplique el cambio.

4.7.7 Intente realizar un ping a la computadora de al lado.

Comente lo acontecido en esta última parte de la práctica y explique el por qué de esto.

_____
_____
_____
_____
_____
_____
_____
_____

4.7.8 Realice los pasos necesarios para dejar la sección Predefined del módulo Network Security tal como estaba en un inicio.

4.7.9 Apague la computadora.

5.- Cuestionario

1. ¿Qué es una regla de filtrado?

_____
_____
_____
_____

2. ¿Qué son los paquetes?

_____
_____
_____
_____

3. ¿Qué es el rango de red?

_____
_____
_____
_____

6.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica:

_____
_____
_____
_____
_____
_____
_____

PRÁCTICA 8
Filtrado de Puertos TCP/UDP mediante un Firewall
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1.- ¿Mencione los diferentes tipos de firewall existentes?

2. ¿Qué es un host y que un localhost?

3. ¿Qué es un protocolo?

4. ¿Qué es un ISP?

5. ¿A qué nos referimos al decir “host local” y “host remote”?

6. ¿Qué es la máscara de red?

7. ¿Qué son las IRC?

8. ¿Qué es un Spyware?

PRÁCTICA 9
SSH: Secure Shell

1.- Objetivos de Aprendizaje
El alumno:

- Al finalizar la práctica, conocerá la importancia de utilizar el protocolo SSH (Secure Shell) y su herramienta OpenSSH.
- Será capaz de iniciar una sesión remota a través de SSH, utilizando autenticación por contraseña.
- Iniciará una sesión remota con clave pública, generando las claves.
- Podrá transferir claves públicas al servidor.

2.- Conceptos teóricos

SSH™ permite a los usuarios registrarse en sistemas de host remotamente. A diferencia de *FTP* o *Telnet*, *SSH* encripta la sesión de registro imposibilitando que alguien pueda obtener contraseñas no encriptadas.

SSH está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la shell de comando, tales como *telnet* o *rsh*. Un programa relacionado, el *scp*, reemplaza otros programas diseñados para copiar archivos entre hosts como *rcp*. Ya que estas aplicaciones antiguas no encriptan contraseñas entre el cliente y el servidor, evite usarlas mientras le sea posible. El uso de métodos seguros para registrarse remotamente a otros sistemas hará disminuir los riesgos de seguridad tanto para el sistema cliente como para el sistema remoto.

Características de SSH

SSH (o *S*ecure *SH*ell) es un protocolo para crear conexiones seguras entre dos sistemas usando una arquitectura cliente/servidor.

El *protocolo SSH* proporciona los siguientes tipos de protección:

- Después de la conexión inicial, el cliente puede verificar que se está conectando al mismo servidor al que se conectó anteriormente.
- El cliente transmite su información de autenticación al servidor usando una encriptación robusta de 128 bits.
- Todos los datos enviados y recibidos durante la conexión se transfieren por medio de encriptación de 128 bits, lo cual los hacen extremamente difícil de descifrar y leer.
- El cliente tiene la posibilidad de enviar aplicaciones X11 lanzadas desde el intérprete de comandos de la shell. Esta técnica proporciona una interfaz gráfica segura (llamada *reenvío por X11*) que proporciona un medio seguro para usar aplicaciones gráficas sobre una red.

Ya que el protocolo *SSH* encripta todo lo que envía y recibe, se puede usar para asegurar protocolos inseguros. El servidor *SSH* puede convertirse en un conducto para convertir en seguros los protocolos inseguros mediante el uso de una técnica llamada *reenvío por puerto*, como por ejemplo POP, incrementando la seguridad del sistema en general y de los datos.

Red Hat Linux contiene el paquete general de *OpenSSH* (*openssh*)*, el servidor de OpenSSH (openssh-server)* y *los paquetes de clientes (openssh-clients).* Los paquetes OpenSSH requieren el paquete OpenSSL (openssl). OpenSSL instala varias librerías criptográficas importantes que ayudan a OpenSSH a proporcionar comunicaciones encriptadas.

Una gran cantidad de programas de cliente y servidor pueden usar el protocolo SSH. Muchas aplicaciones SSH cliente están disponibles para casi todos los principales sistemas operativos en uso hoy día.

¿Por qué usar SSH?

Los usuarios maliciosos tienen a su disposición una variedad de herramientas para interceptar y dirigir el tráfico de la red para ganar acceso al sistema. En términos generales, estas amenazas se pueden catalogar del siguiente modo:

- *Intercepción de la comunicación entre dos sistemas*: En este escenario, existe un tercero en algún lugar de la red entre entidades en comunicación que hace una copia de la información que pasa entre ellas. La parte interceptora puede interceptar y conservar la información, o puede modificar la información y luego enviarla al recipiente al cual estaba destinada. Este ataque se puede articular a través del uso de un paquete sniffer — una utilidad de red muy común.
- *Personificación de un determinado host*: Con esta estrategia, un sistema interceptor finge ser el recipiente a quien está destinado un mensaje. Si funciona la estrategia, el cliente no se da cuenta del engaño y continúa la comunicación con el interceptor como si su mensaje hubiese llegado a su destino satisfactoriamente. Esto se produce con técnicas como el envenenamiento del DNS o spoofing de IP.

Ambas técnicas causan que se intercepte información, posiblemente con propósitos hostiles. El resultado puede ser catastrófico.

Si se utiliza SSH para inicios de sesión de shell remota y para copiar archivos, estas amenazas a la seguridad se pueden disminuir notablemente. Esto es porque el cliente SSH y el servidor usan firmas digitales para verificar su identidad. Adicionalmente, toda la comunicación entre los sistemas cliente y servidor es encriptada. No servirán de nada los intentos de falsificar la identidad de cualquiera de los dos lados de la comunicación ya que cada paquete está cifrado por medio de una clave conocida sólo por el sistema local y el remoto.

Capa de Presentación

El papel principal de la capa de presentación es facilitar una comunicación segura entre los dos hosts en el momento y después de la autenticación. La capa de presentación lleva esto a cabo manejando la encriptación y decodificación de datos y proporcionando protección de integridad de los paquetes de datos mientras son enviados y recibidos. Además, la capa de presentación proporciona compresión de datos, lo que acelera la transmisión de información.

Al contactar un cliente a un servidor por medio del protocolo SSH, se negocian varios puntos importantes para que ambos sistemas puedan construir la capa de presentación correctamente. Durante el intercambio se producen los siguientes pasos:

- Intercambio de claves.
- Se determina el algoritmo de encriptación de la clave pública.
- Se determina el algoritmo de la encriptación simétrica.
- Se determina el algoritmo autenticación de mensajes.
- Se determina el algoritmo de hash que hay que usar.

3.- Equipo y material necesario

Material que deberá traer el alumno:

A. 1 Lápiz o pluma.
B. 1 diskette de 3 ½ (vació).

Equipo del Laboratorio:

C. 1 Computadora con Sistema Operativo *Linux Red Hat 9.0* ó Superior.

4.- Desarrollo:

4.1 Modo de trabajar

La realización de la práctica se hará por equipos de dos personas por computadora y se trabajara conjuntamente con la maquina de al lado, que hará la función de servidor.

4.2 Ejercicio

NOTA: Para ejemplificar el siguiente ejercicio se muestra la siguiente figura No.1.

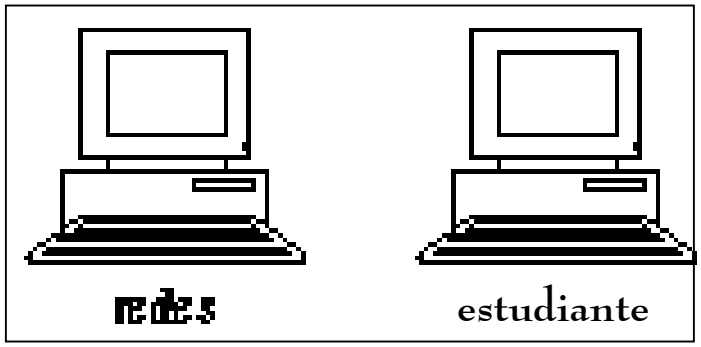


Figura No.1

4.2.1 Encienda el sistema y elija la opción de cargar *Linux*.

4.2.2 Entre a sesión como usuario normal. La cuenta y la contraseña serán proporcionadas por el profesor del laboratorio.

4.2.3 Abra un shell prompt seleccionando (Ver figura No.2):

Main Menu => System Tools => Terminal.

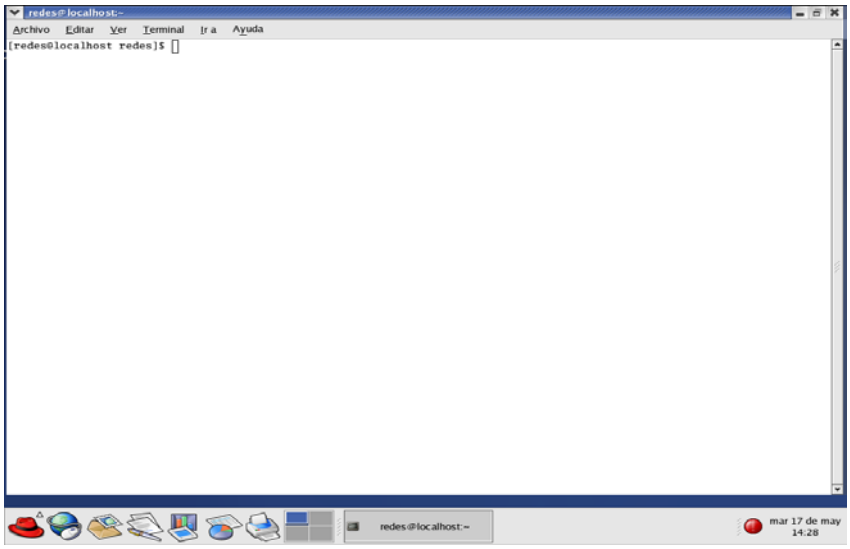


Figura No.2

Para salir del shell prompt teclee exit o presione <ctrl>d en el prompt.

4.2.4 Ahora podemos emplear el sistema para ejecutar los comandos que necesitemos utilizar.

4.2.5 Visualice el archivo */etc/ssh/ssh_config* en el cliente redes. (Ver figura No.3).Teclee lo siguiente:

[redes@localhost redes]$ vi /etc/ssh/ssh_config

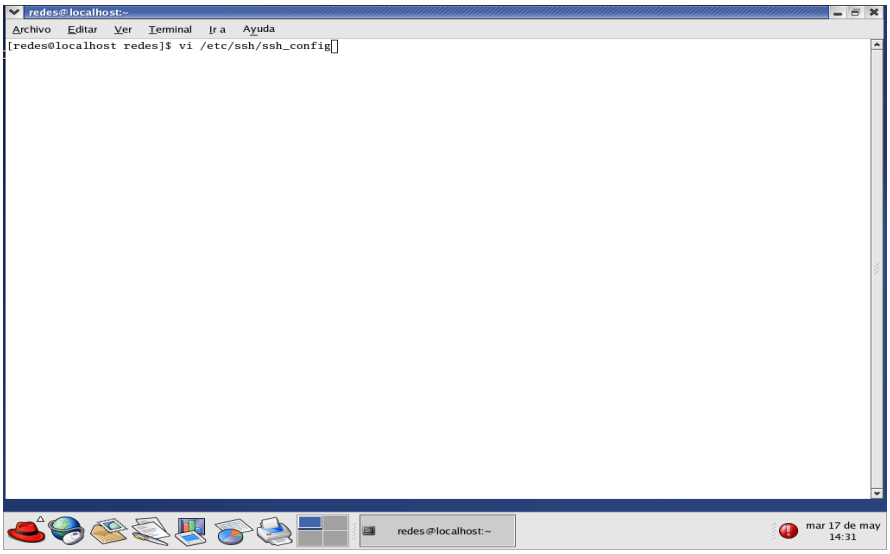


Figura No.3

La salida del comando le dará algo similar a lo siguiente (Ver figura No.4):

*Host **
Port 22
Protocol 2
StrictHostKeyChecking ask
PasswordAuthentication yes
PubkeyAuthentication yes
IdentityFile ~/.ssh/id_rsa
ForwardX11 yes

NOTA: El símbolo ~ significa el directorio home del usuario.

Anote la información obtenida en la pantalla.

_____
_____
_____
_____
_____
_____

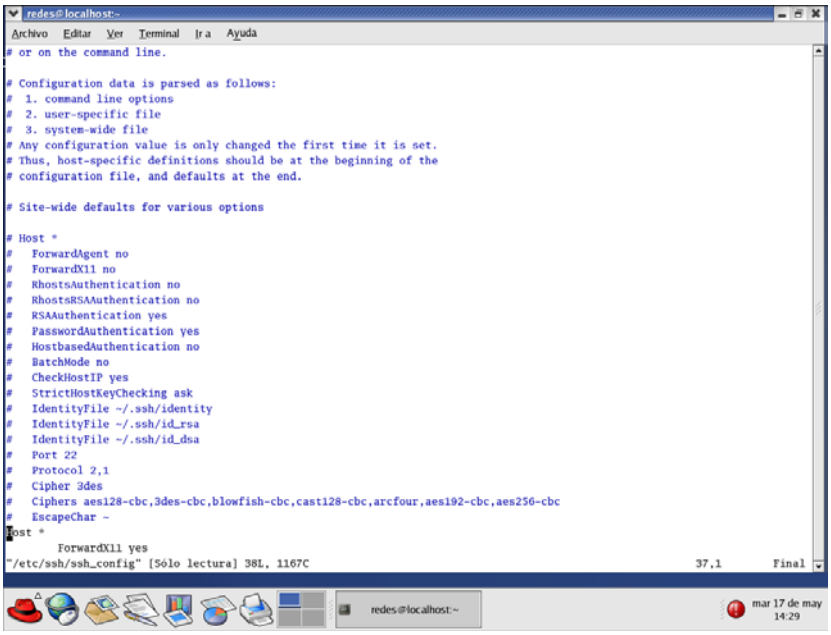


Figura No.4

4.3 Iniciando una sesión remota con contraseña

4.3.1 El primer ejemplo que analizaremos es el inicio de una sesión remota a través de SSH, utilizando autenticación por contraseña. Para ello, ingrese como usuario “estudiante” en el servidor, es decir, la computadora de al lado. (Ver figura No.5).

Introduzca el siguiente comando:

[redes@localhost redes]$ ssh estudiante@192.168.2.x

NOTA: El valor x será proporcionado por el profesor de acuerdo a la máquina que esté utilizando.

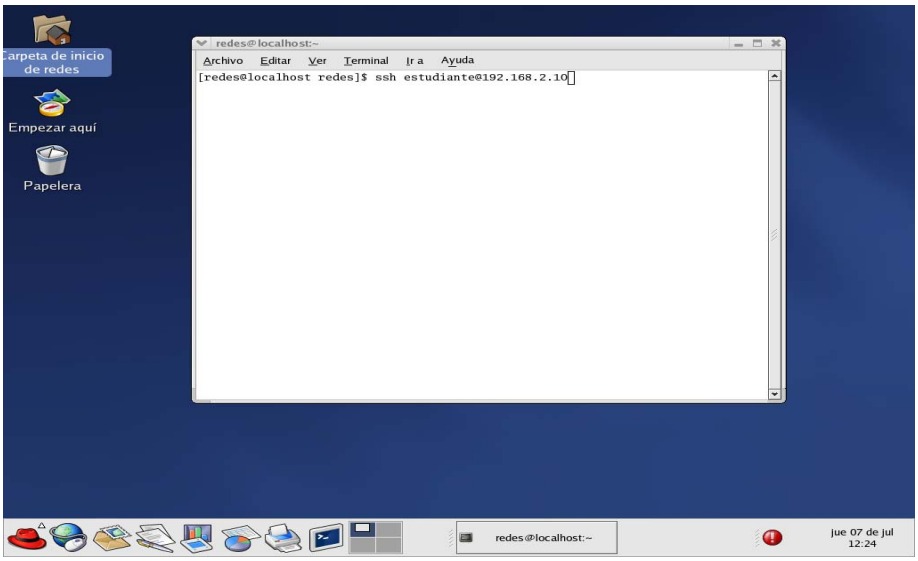


Figura No.5

Al ser la primera vez que se conecta al servidor, si previamente no ha agregado la clave pública del mismo en */home/redes/.ssh/known_hosts* (o en el archivo global */etc/ssh/ssh_known_hosts*), aparecerá un mensaje similar al siguiente: (Ver figura No.6).

The authenticity of host 'redes3 (192.168.2.x)' can't be established.
RSA key fingerprint is d7:7d:c9:b4:09:56:50:bd:9e:78:b8:93:8c:8d:ed:5c.
Are you sure you want to continue connecting (yes/no)?

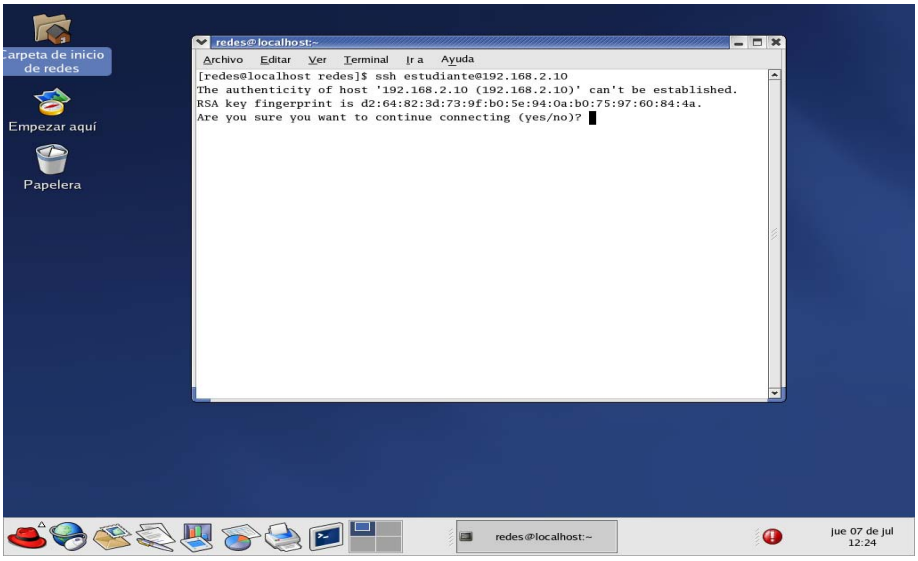


Figura No.6

4.3.2 Debido a que confiamos que esa es la verdadera clave pública del servidor. Teclee yes. Luego el cliente informará algo similar a lo siguiente:

Warning: Permanently added 'redes3,192.168.2.x' (RSA) to the list of known hosts.

Lo que significa que se ha agregado la clave pública del servidor en */home/redes/.ssh/known_hosts*. (Ver figura No.7). Luego el cliente solicitará el ingreso de la contraseña o password:

estudiante@servidor's password:

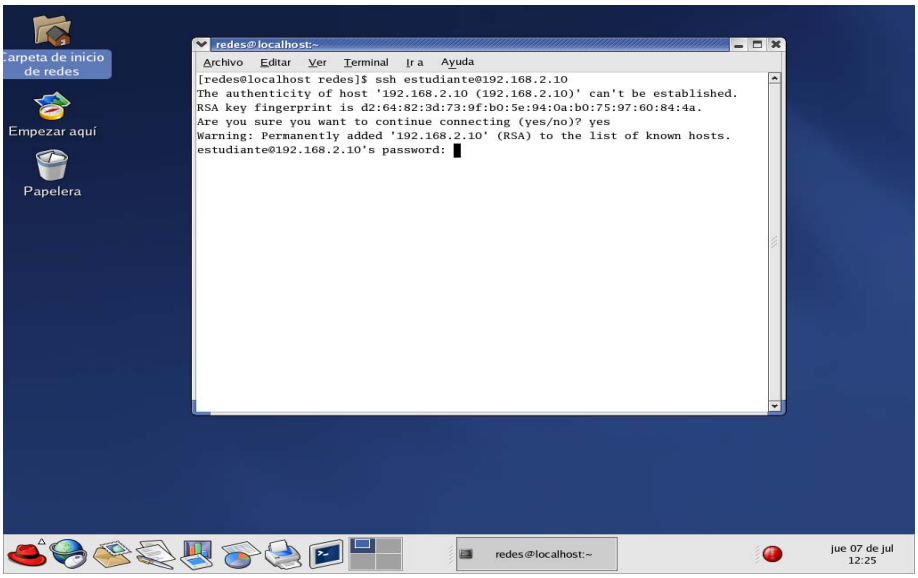


Figura No.7

4.3.3 Teclee la contraseña de la cuenta estudiante, que será proporcionada por el profesor. Finalmente, si la contraseña ingresada es correcta, aparecerá el mensaje de bienvenida del servidor, similar a la siguiente: (Ver figura No.8).

Bienvenido a servidor (Debian GNU/Linux)
[estudiante@servidor:estudiante]$

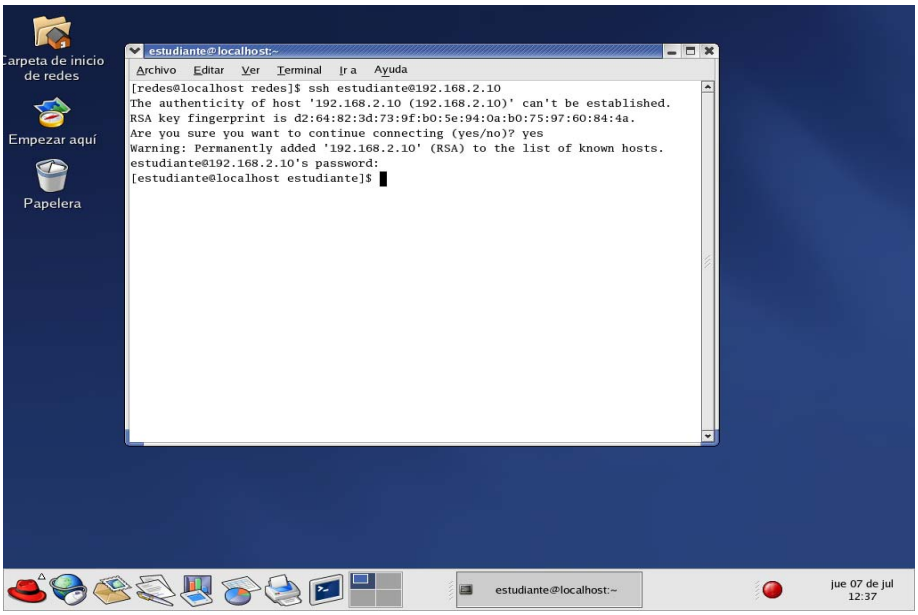


Figura No.8

Con lo cual hemos iniciado una sesión en el servidor como el usuario estudiante.

4.3.4 Cierre la sesión remota. Teclee exit ó logout (ó presione <CTRL>+D): (Ver figura No.9).

[estudiante@servidor:estudiante]$ exit
logout
Connection to servidor closed.
[redes@localhost redes]$

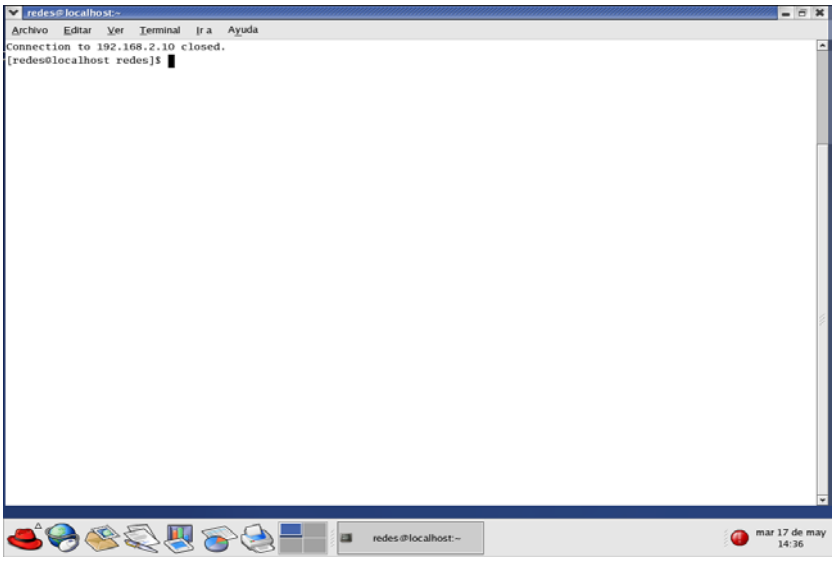


Figura No.9

4.3.5 Intente conectarse nuevamente al servidor con la misma cuenta, si recibe una clave pública diferente a la que tiene almacenada en *known_hosts*, aparecerá un mensaje de advertencia similar al siguiente y se abortará la conexión:

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED! @@
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
The fingerprint for the RSA key sent by the remote host is d7:7d:c9:b4:09:56:50:bd:9e:78:b8:93:8c:8d:ed:5c.
Please contact your system administrator.
Add correct host key in /home/redes/.ssh/known_hosts to get rid of this message.
Offending key in /home/redes/.ssh/known_hosts
RSA host key for 192.168.2.x has changed and you have requested strict checking.

4.3.6 Si la clave pública no es diferente, realice la conexión y después salga de ésta.

4.4 Iniciando una sesión remota con clave pública
Generando las claves

4.4.1 El primer paso para utilizar la autenticación mediante clave pública es generar el par de claves de RSA que se utilizarán.

Para ello, ejecute el siguiente comando: (Ver figura No.10).

[redes@localhost redes]$ ssh-keygen -t rsa

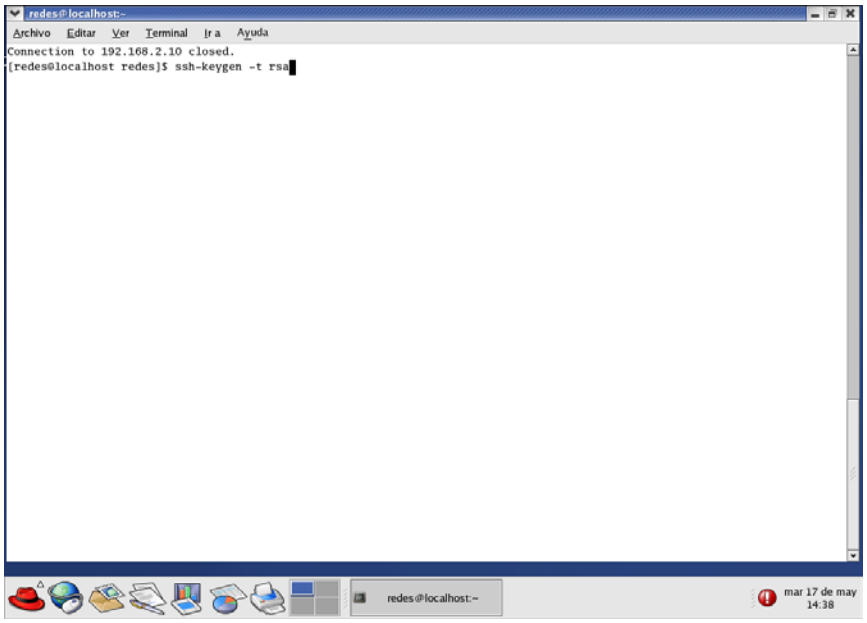


Figura No.10

El programa responderá algo similar a lo siguiente: (Ver figura No.11).

Generating public/private rsa key pair.
Enter file in which to save the key (/home/redes/.ssh/id_rsa)

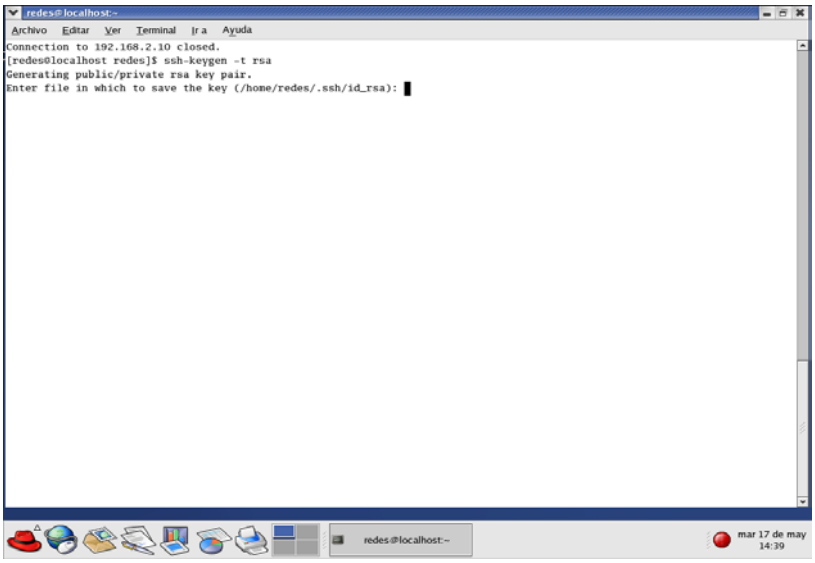


Figura No.11

4.4.2 Nos solicita que ingresemos el nombre del archivo en donde se almacenará la clave privada, proponiéndonos */home/redes/.ssh/id_rsa*, lo cual concuerda con la configuración del cliente SSH. Presione <*Enter*> para aceptar el valor por defecto. Luego nos solicitará una frase clave: (Ver figura No.12).

Enter passphrase (empty for no passphrase):
Enter same passphrase again:

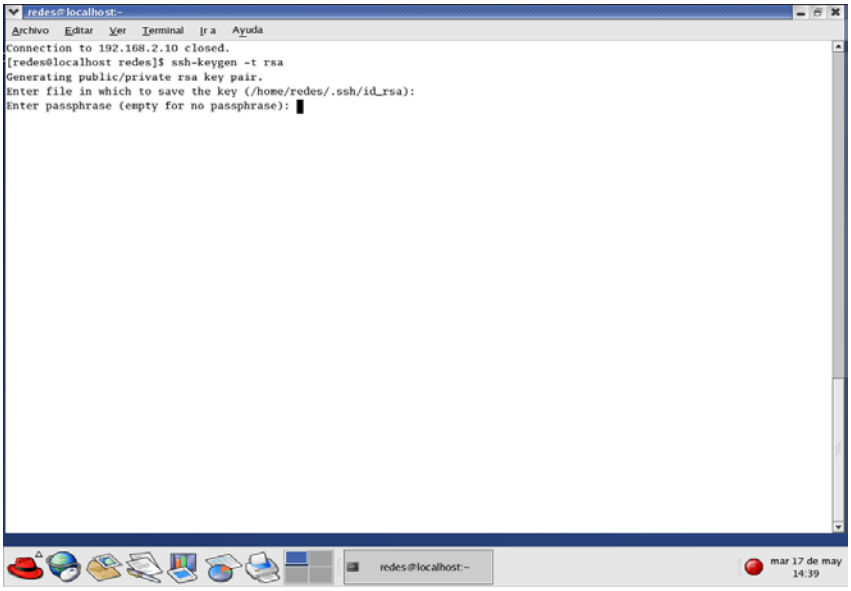


Figura No.12

4.4.3 Presione dos veces <*Enter*> para omitir el uso de una frase clave. Más adelante nos ocuparemos de esto. Finalmente nos informa: (Ver figura No.13).

Your identification has been saved in /home/redes/.ssh/id_ rsa.
Your public key has been saved in /home/redes/.ssh/id_ rsa.pub.
The key fingerprint is:
13:8b:23:74:53:e4:0f:b3:16:49:1b:79:64:60:7c:38 redes@cliente
[redes@localhost redes]$

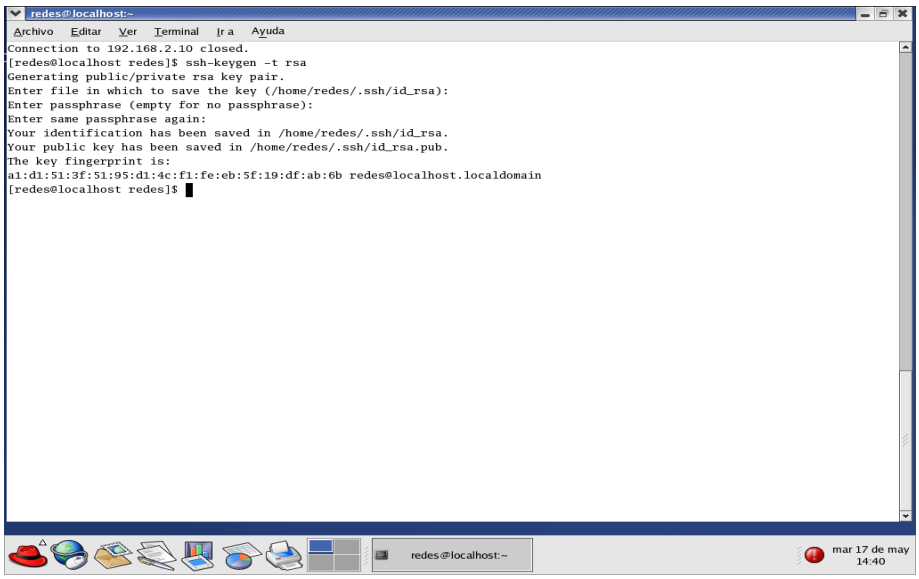


Figura No.13

4.5 Transfiriendo la clave pública al servidor

Luego, debemos transferir la clave pública del usuario redes *(/home/redes/.ssh/id_rsa.pub*) al directorio *home* del usuario estudiante en servidor y añadirla al final del archivo */home/estudiante/authorized_keys*.

4.5.1 Para ello, cierre la sesión redes y abra una sesión como usuario root, la contraseña será proporcionada por el profesor.

4.5.2 Desde una terminal teclee: (Ver figura No.14).

[root@localhost root]$ cat /home/redes/.ssh/id_rsa.pub > > /home/estudiante/authorized_keys

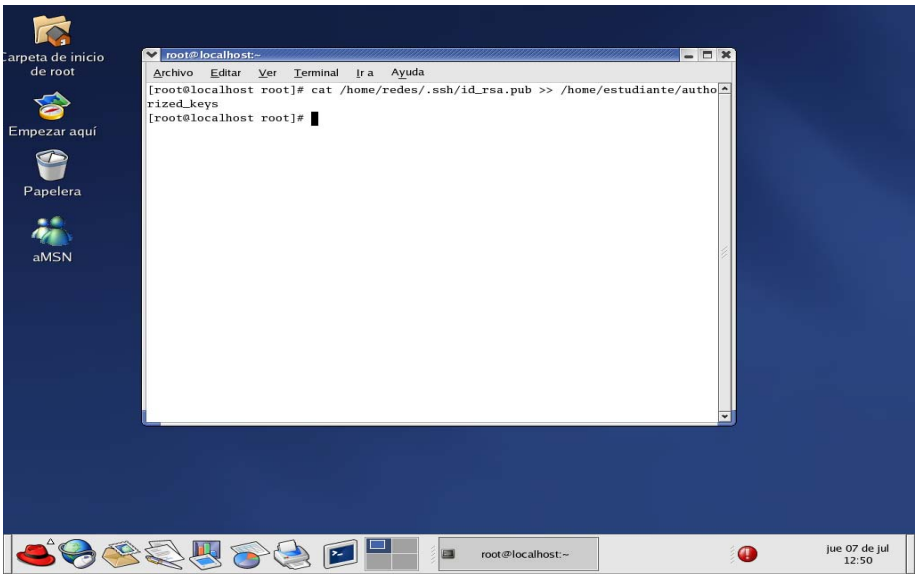


Figura No.14

4.5.3 Cierre la cesión root y vuelva a la sesión redes.

4.6 *Iniciando la sesión*

4.6.1 Ingrese el siguiente comando:

[redes@localhost redes]$ ssh estudiante@192.168.2.x

El servidor nuevamente envía su clave pública de RSA, la cual es comparada con la almacenada en *known_hosts*, y si coincide el proceso continúa.

El cliente de SSH, al encontrar el archivo */home/redes/.ssh/id_rsa*, primero intentará la autenticación con clave pública. El servidor le enviará el *challenge* cifrado con la clave pública encontrada en */home/estudiante/authorized_keys* (en el directorio *home* del usuario estudiante) y el cliente deberá devolverla descifrada (usando la clave */home/redes/.ssh/id_rsa* en el directorio *home* del usuario redes).

NOTA: Esto se realiza automáticamente, sin la intervención del usuario.
Si esto se realiza correctamente, se iniciará la sesión remota (Ver figura No.15).

Bienvenido a servidor (Debian GNU/Linux)
[estudiante@servidor:estudiante]$

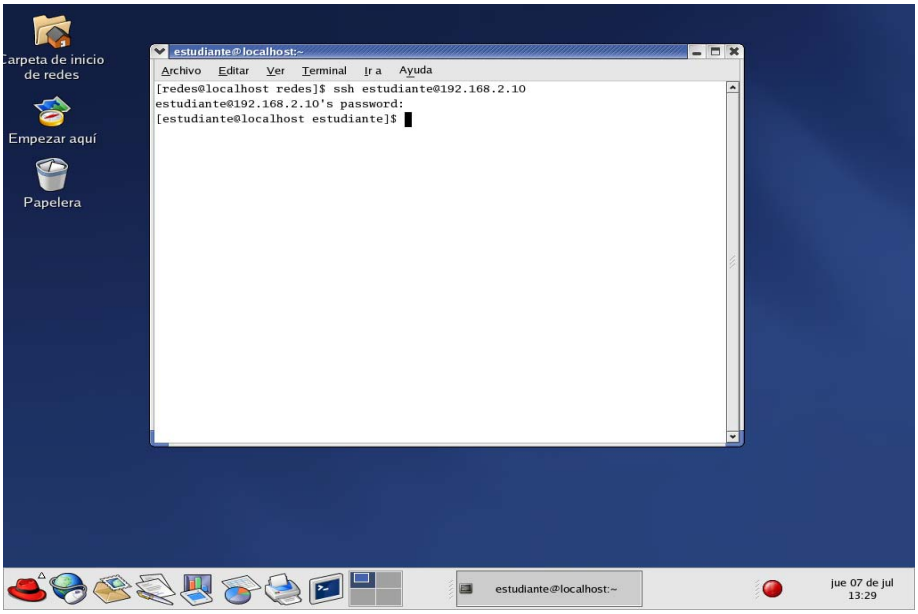


Figura No.15

4.6.2 Si la autenticación con clave pública hubiera fallado, el cliente intentará con la autenticación con contraseña. Después de conectarse al servidor, sálgase de este. (Ver figura No.16).

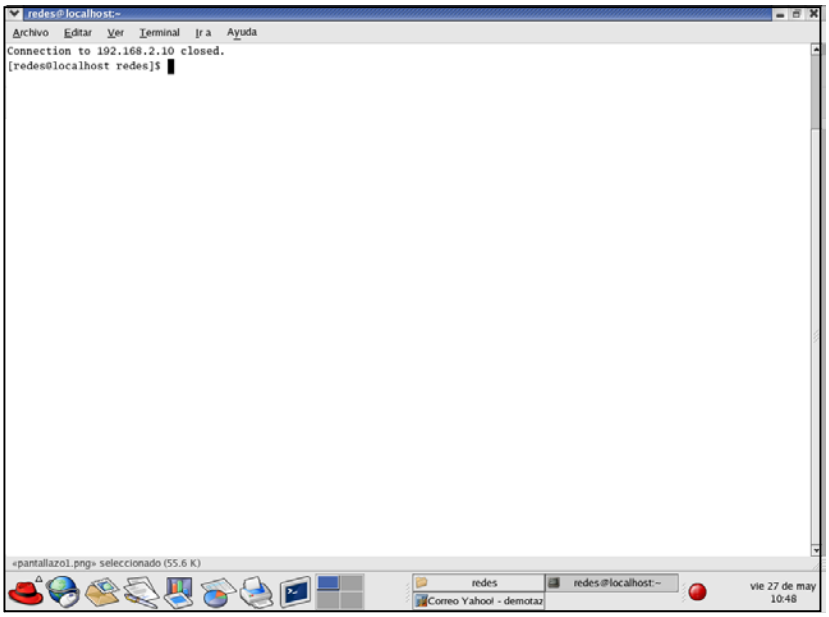


Figura No.16

4.7 Asegurando la clave privada

4.7.1 Cuando creo el par de claves usando ssh-keygen, omitimos especificar la frase clave que se usaría a tal efecto. Usando nuevamente ssh-keygen asignaremos una nueva. Teclee lo siguiente:

[redes@localhost redes]$ ssh-keygen -p -f /home/redes/.ssh/id_rsa

Nos pedirá ingresar la nueva frase clave: (Ver figura No.17).

Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
[redes@localhost redes]$

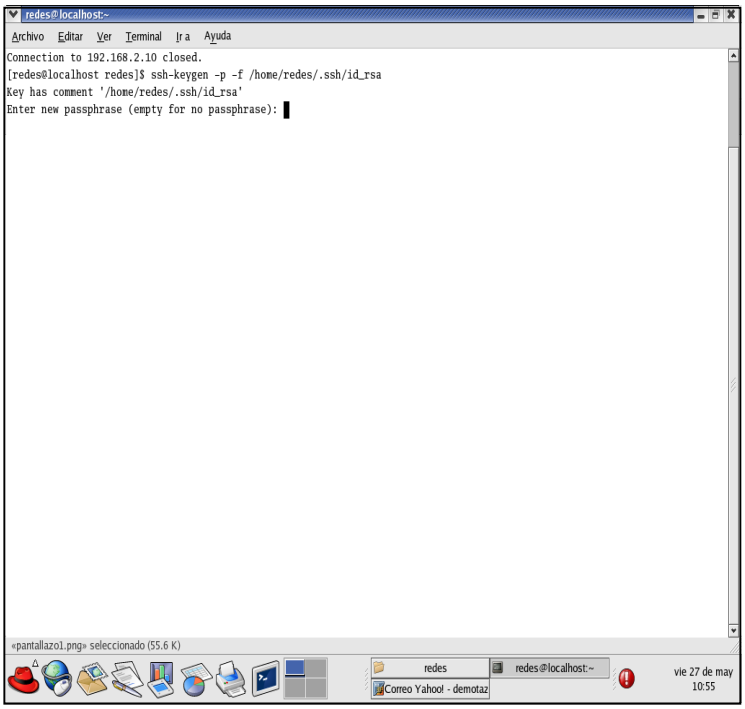


Figura No.17

4.7.2 Ingrese la frase clave, para nuestro caso gilthoniel, en ambas ocasiones. Finalmente nos informa: (Ver figura No.18).

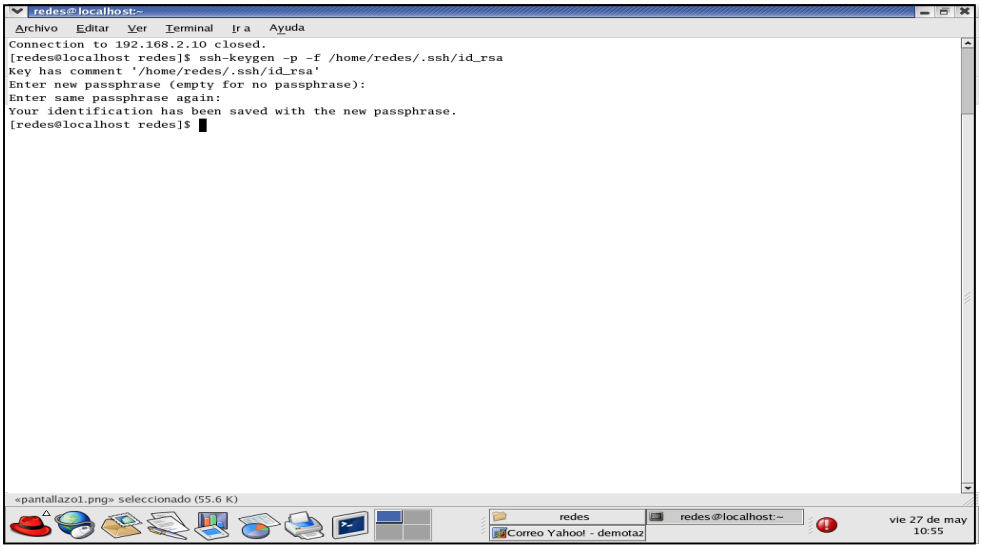


Figura No.18

Una vez finalizado, la clave privada se encuentra cifrada con 3DES, razón por la cual cada vez que necesitemos usarla (por

ejemplo, en cualquiera de los ejemplos anteriores al intentar conectarnos con el servidor), se nos solicitará la frase clave para descifrarla. Por ejemplo:

[redes@localhost redes]$ ssh estudiante@192.168.2.x
Need passphrase for /home/redes/.ssh/id_rsa
Enter passphrase for /home/redes/.ssh/id_rsa

El procedimiento continuará como en los casos anteriores.

4.8 Usando ssh-agent en el shell

4.8.1 En la sesión redes, ejecute el ssh-agent de la siguiente forma: (Ver figura No.19).

[redes@localhost redes]$ eval 'ssh-agent'
Agent pid 3156
[redes@localhost redes]$

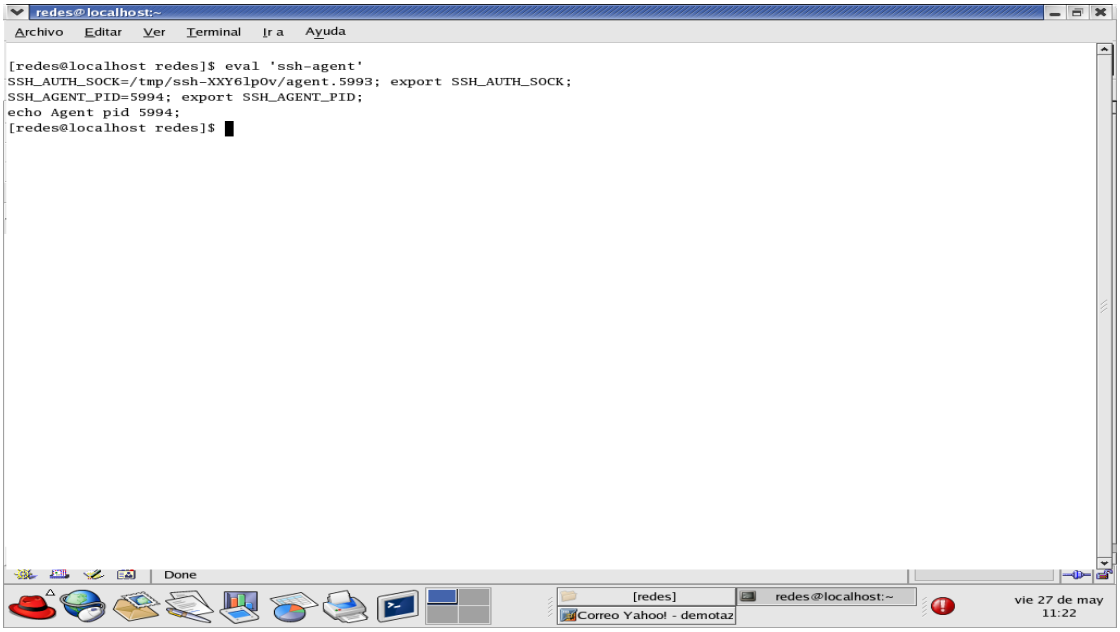


Figura No.19

4.8.2 Agregue su clave privada de RSA, recuerde que es gilthoniel. (Ver figura No.20). Para ello use el comando *ssh-add*:

[redes@localhost redes]$ ssh-add /home/redes/.ssh/id_rsa
Enter passphrase for /home/redes/.ssh/id_rsa
Identity added: /home/redes/.ssh/id_rsa (/home/redes/.ssh/id_rsa)
[redes@localhost redes]$

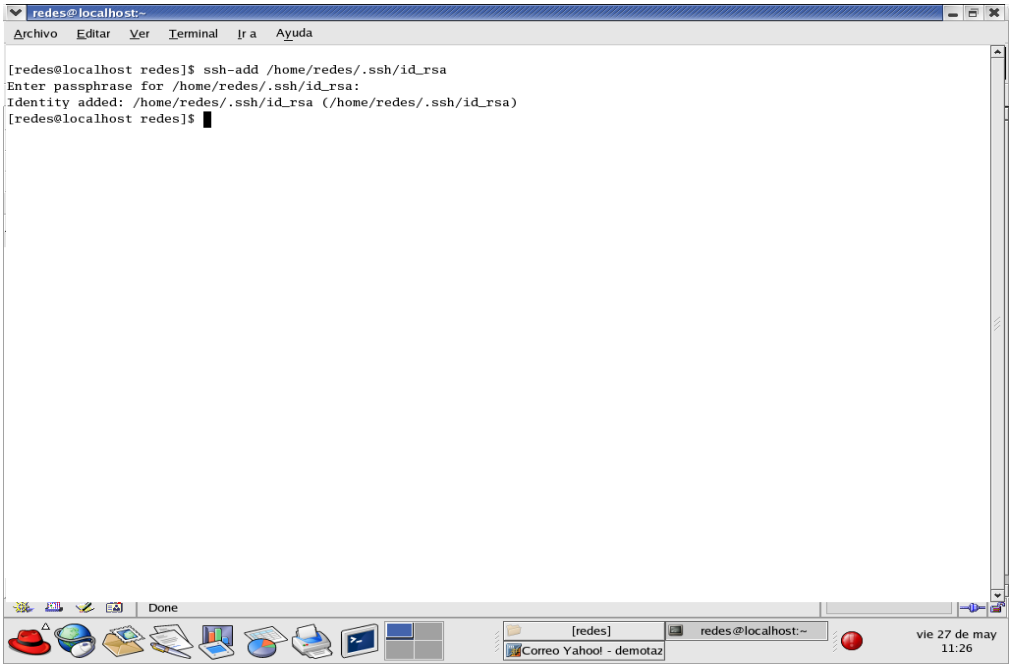


Figura No.20

Este procedimiento puede repetirse si tiene varias claves privadas. Luego, siempre y cuando nos encontremos en el mismo shell en donde hemos iniciado el ssh-agent, al ejecutar ssh (o scp) este le solicitará al ssh-agent nuestra clave privada, con lo cual no deberemos introducir la frase clave.

4.8.3 Inicie una sesión remota como usuario root al servidor. (Ver figura No.21).

[redes@localhost redes]$ ssh root@192.168.2.x

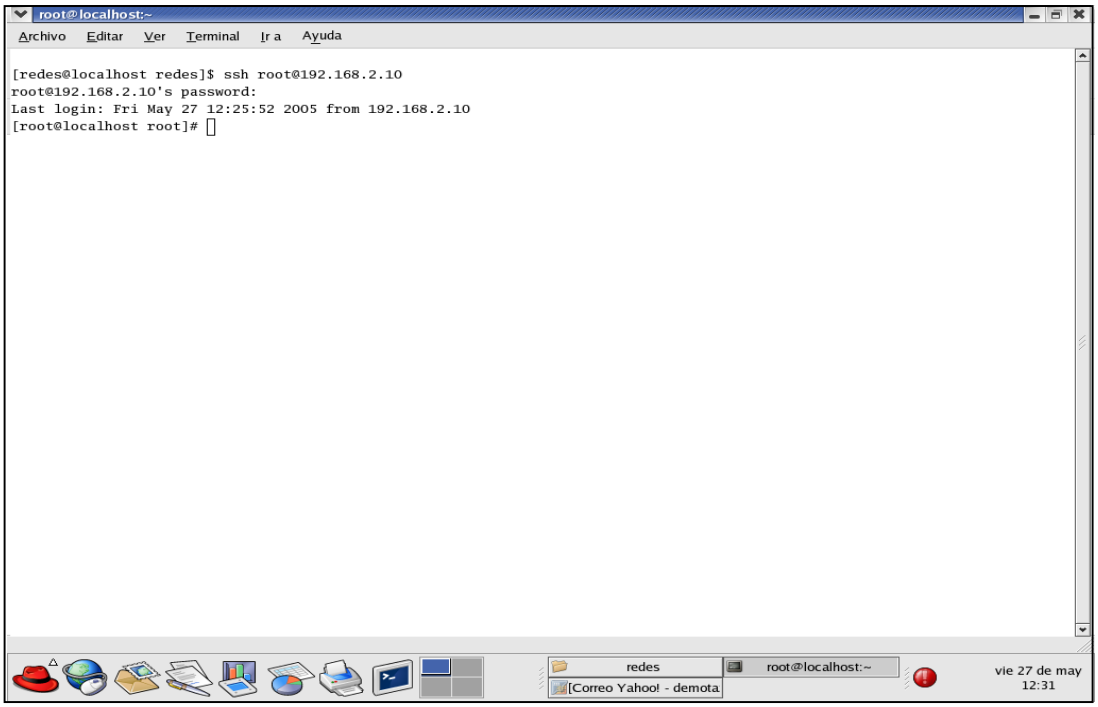


Figura No.21

4.8.4 Monte el diskette de 3 ½ en el floppy. (Ver figura No.22).

[root@servidor root]$ mount /mnt/floppy

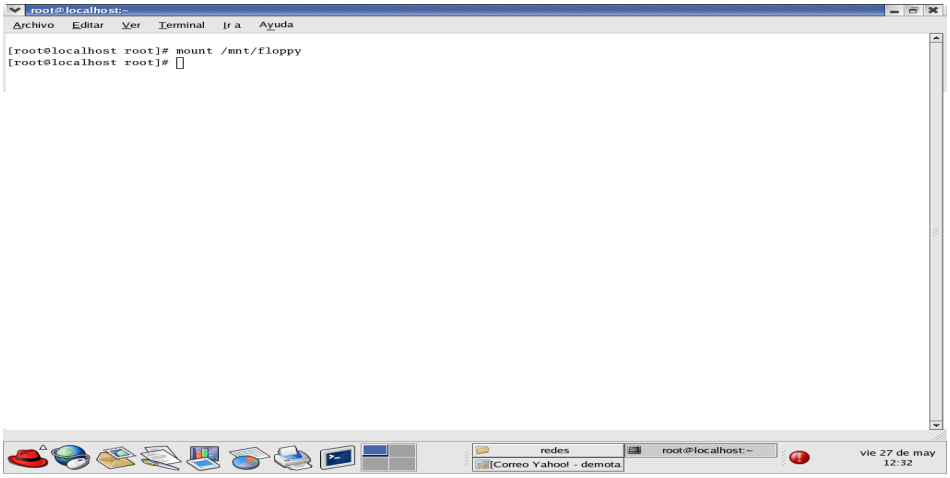


Figura No.22

4.8.5 Dentro del diskette cree un archivo llamado ssh con el comando *cat* y escriba un breve texto. (Ver figura No.23).

[root@servidor root]$ cat > /mnt/floppy/ssh

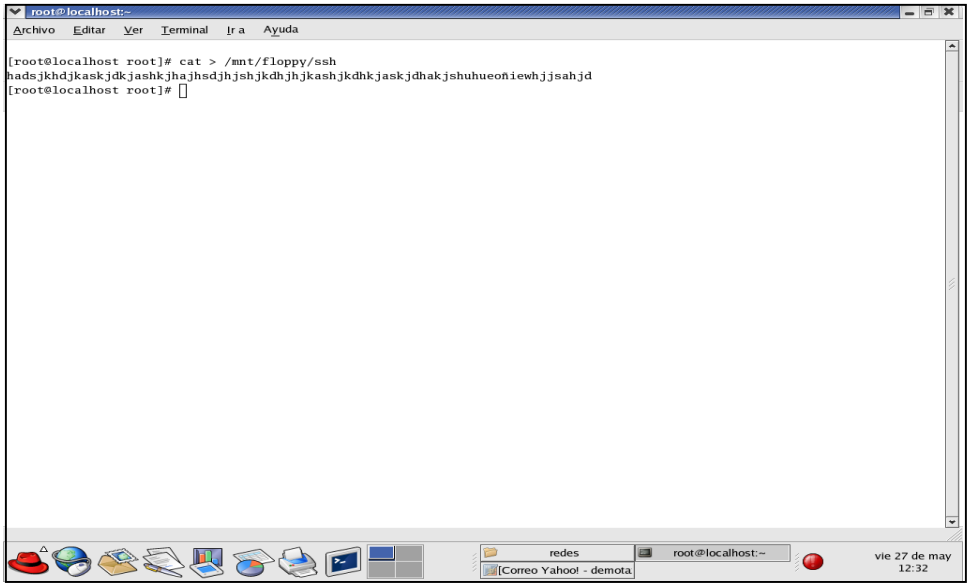


Figura No.23

4.8.6 Cierre la sesión remota. Ahora revise el archivo creado con el comando *vi.* (Ver figura No.24).

[redes@localhost redes]$ vi /mnt/floppy/ssh

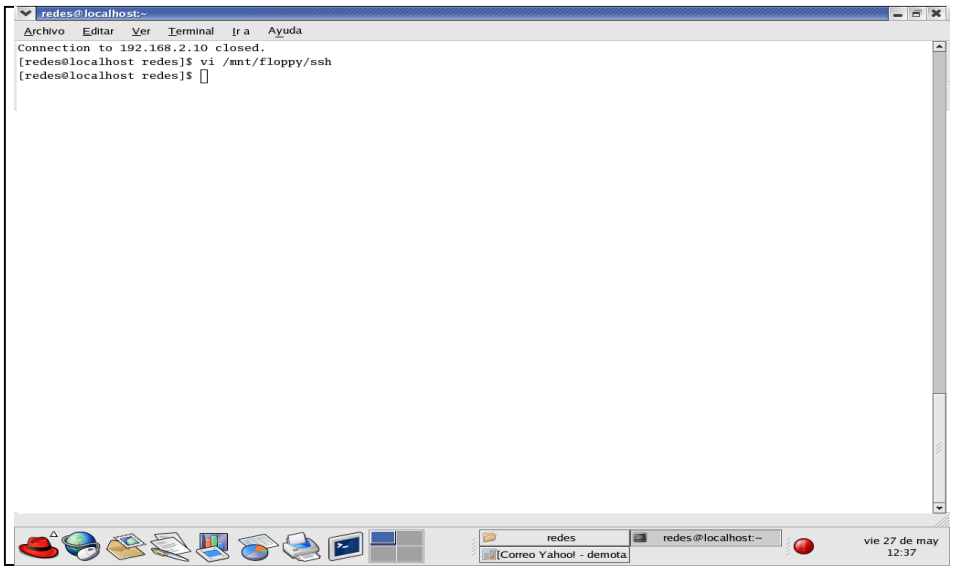


Figura No.24

Escriba la información visualizada y diga si es lo que usted escribió. (Ver figura No.25).

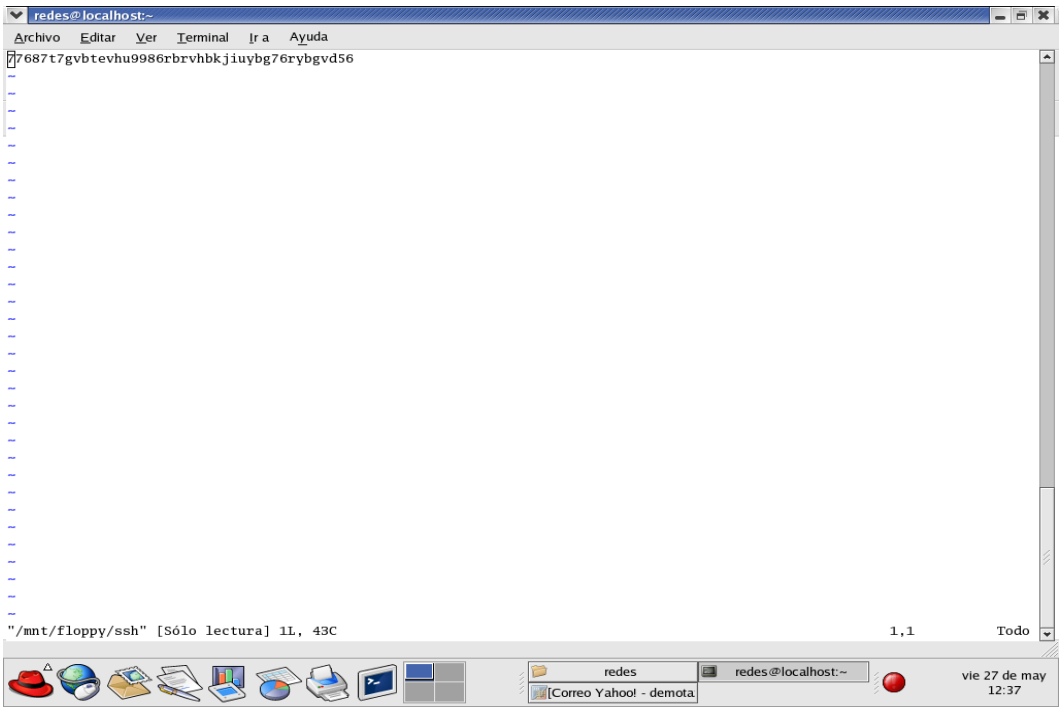


Figura No.25

4.8.7 Inicie una sesión remota como usuario root al servidor. (Ver figura No.26).

[redes@localhost redes]$ ssh root@192.168.2.x

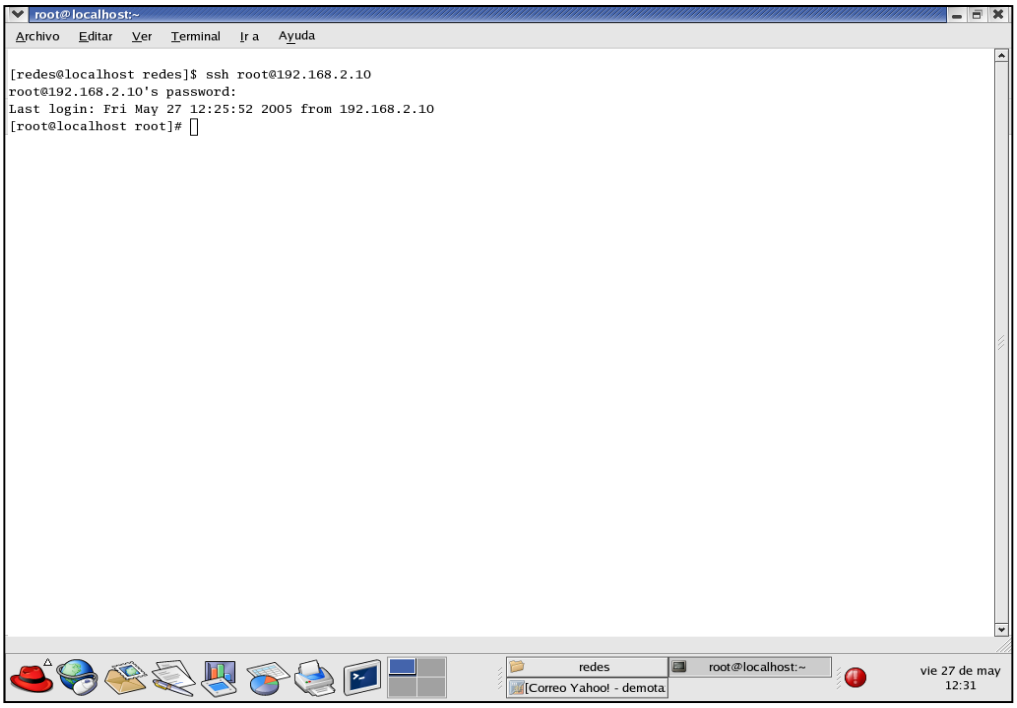


Figura No.26

4.8.8 Desmonte el diskette de 3 ½ del floppy. (Ver figura No.27).

[root@servidor root]$ umount /mnt/floppy

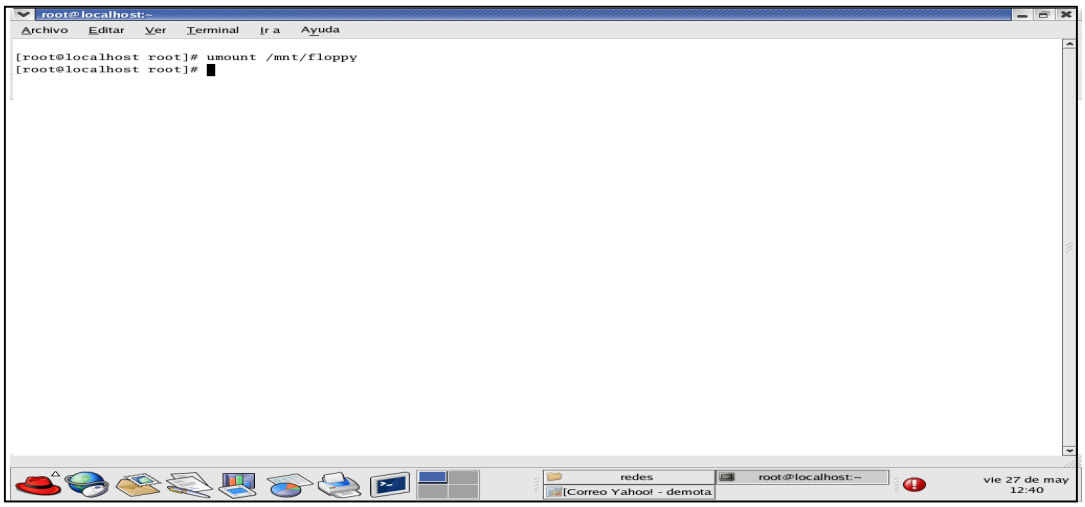


Figura No.27

4.8.9 Cierre la sesion remota, el shell, la sesion redes y apague la computadora.

5.-Cuestionario

1. Explique la información obtenida en apantalla en el punto 4.2.5

2. ¿Qué sucedería si escribiera mal la contraseña al querer hacer una conexión remota con el ssh?

3. Explique porque el diskette en el punto 4.8.6 contenía información distinta a la tecleada por usted.

6.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica:

_____
_____
_____
_____
_____
_____
_____
_____

PRÁCTICA 9
SSH: Secure Shell
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1. ¿Qué es el reenvió por X11?

2. ¿Qué es un sniffer?

3. ¿En qué consiste el envenenamiento de la caché del DNS?

4. ¿Qué es un spoofing de IP?

5. Mencione cuales son las versiones del protocolo SSH y explique sus características.

6. ¿Cuáles son las secuencias de eventos a llevar a cabo en una conexión SSH?

7. ¿A qué nos referimos con la Autenticación?

8. Explique detalladamente los pasos que se producen cuando un cliente contacta a un servidor a través del protocolo SSH.

9. ¿Qué algoritmo de codificación emplea el protocolo SSH?

10. Investigue cuales son los archivos de configuración de OpenSSH y explíquelos detalladamente.

11. ¿Para que nos sirve el comando *ssh-keygen*?

12. ¿Qué es SSH-Agent?

PRÁCTICA 10
Aplicación Web Apache

1.- Objetivos de Aprendizaje
El alumno:

- Al finalizar la práctica podrá instalar y configurar una aplicación típica de la capa 7 del Modelo OSI en Plataforma Linux.
- Será capaz de Instalar y configurar de forma segura un servidor web en su versión básica, pero con posibilidad de transferencias seguras mediante SSL (Secure Sockets Layer).
- Conocerá las principales directivas de configuración de un servidor web apache.

2.- Conceptos teóricos

La capa de aplicación del modelo OSI proporciona los servicios utilizados por las aplicaciones para que los usuarios se comuniquen a través de la red.

El servidor HTTP apache es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 (RFC 2616) y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que originalmente apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor *parcheado*).

El servidor apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

En la actualidad (2005), apache es el servidor HTTP más usado, siendo el servidor HTTP del 68% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft.

Versión 2.x

El núcleo 2.x de apache tiene varias mejoras clave sobre el núcleo de apache 1.x. Estas mejoras incluyen *threads* de UNIX, mejor soporte para plataformas no Unix (como Windows), un nuevo API, y soporte de IPv6.

Módulos

El servidor de base puede ser extendido con la inclusión de módulos entre los cuales se encuentran:

- mod_perl - Páginas dinámicas en Perl.
- mod_php - Páginas dinámicas en PHP.
- mod_python - Páginas dinámicas en Python.
- mod_jk - Conector para enlazar con el servidor Jakarta Tomcat de páginas dinámicas en Java (servlets y JSP).
- mod_ssl - Comunicaciones Seguras.
- mod_rewrite - reescritura de direcciones servidas.

HTTP

HTTP es el protocolo de la Web (WWW), usado en cada transacción. Las letras significan **H**yper **T**ext **T**ransfer **P**rotocol, es decir, protocolo de transferencia de hipertexto. El hipertexto es el contenido de las páginas web, y el protocolo de transferencia es el sistema mediante el cual se envían las peticiones de acceder a una página web, y la respuesta de esa web, remitiendo la información que se verá en pantalla. También sirve el protocolo para enviar información adicional en ambos sentidos, como formularios con mensajes y otros similares.

HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores. Al finalizar la transacción todos los datos se pierden. Por esto se popularizaron las

cookies, que son pequeños ficheros guardados en el propio ordenador que puede leer un sitio web al establecer conexión con él, y de esta forma reconocer a un visitante que ya estuvo en ese sitio anteriormente. Gracias a esta identificación, el sitio web puede almacenar gran número de información sobre cada visitante, ofreciéndole así un mejor servicio.

La versión actual de HTTP es la 1.1, y su especificación está en el documento RFC-2616.

HTTP dispone de una variante cifrada mediante SSL llamada HTTPS.

HTTPS

Versión segura del protocolo HTTP. El sistema HTTPS utiliza un cifrado basado en las Secure Socket Layers (SSL) para crear un canal cifrado (cuyo nivel de cifrado depende del servidor remoto y del navegador utilizado por el cliente) más apropiado para el tráfico de información sensible que el protocolo HTTP.

Es utilizado principalmente por entidades bancarias, tiendas en línea, y cualquier tipo de servicio que requiera el envío de datos personales o contraseñas.

El puerto estándar para este protocolo es el 443.

NCSA HTTPd

El NCSA HTTPd era un Servidor web desarrollado originalmente en el National Center for Supercomputing Applications por Robert McCool y una lista de colaboradores.

El desarrollo del NCSA HTTPd se suspendió en 1998, pero el código sobrevivió durante un tiempo en manos del Proyecto Apache, el cual es utilizado actualmente por dos terceras partes de los servidores web de Internet. Prácticamente todo el código de NCSA se ha ido reescribiendo progresivamente en versiones de Apache.

Estructura básica de apache

Cuando el servidor de páginas (apache) recibe la requisición para "x" página éste reconoce cuando debe enviar un documento estático (HTML) o ejecutar algún tipo de aplicación, en el diagrama A se puede observar que la solicitud de "x" página invoca (llama) un programa en *Perl* y este a su vez solicita información a una base de datos, por lo tanto para llevar acabo esta operación debieron iniciarse *2 procesos* nuevos.

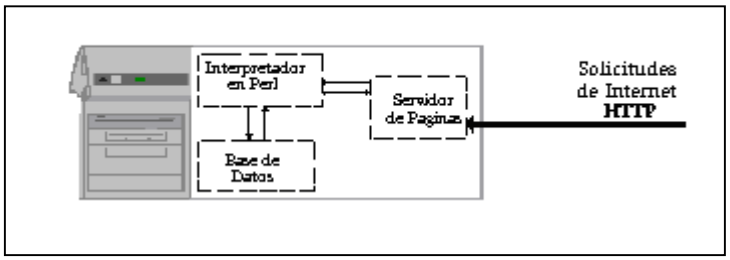


Diagrama A.

Ventajas de apache

- Es capaz de utilizar otros interpretadores y lenguajes como "Tcl", "PhP" y "Python".
- Puede conectarse directamente a una Base de datos.
- Posee diversos módulos que le permiten utilizar una gran gamma de lenguajes y desarrollar funcionalidades avanzadas.

Existen otros servidores web además de apache como, por ejemplo:

- AOLServer
- IIS
- Zope
- Tomcat o Jakarta Apache

El servidor web apache, es uno de los más utilizados actualmente en Internet desde hace años.

Además, posee la ventaja de estar disponible para varios sistemas operativos, como Unix, Windows NT, Windows 2000, Windows XP, Windows.NET SERVER 2003.

Para proporcionar la posibilidad de SSL utilizaremos el paquete mod_ssl, por su mayor facilidad de instalación y mejor documentación que el paquete Apache-SSL.

Generalmente, apache se inicia como usuario root, y posteriormente atiende las peticiones recibidas como el usuario establecido por la directiva User del fichero de configuración httpd.conf. Como con cualquier orden que ejecuta root, hay que tener cuidado de que esté protegido frente a modificaciones por otros usuarios. No sólo los ficheros deben ser modificables sólo por root, sino también los directorios y los padres de estos directorios. Por ejemplo, si situamos la raíz del servidor en /usr/local/apache, este directorio debe crearse como root.

Arranque y parada del servidor

Si se desea que el servidor escuche en el puerto 80 (el de default), que es un puerto reservado, habrá que lanzarlo como root. Tenemos dos posibilidades.

Ejecutar httpd (usr/local/apache/bin/httpd,), o la orden apachectl star. (usr/local/apache/bin/apachectl start) que es la opción más recomendable.

Esto buscará el fichero de configuración (*httpd.conf*) en el lugar compilado en el código (por defecto */usr/local/apache/conf/httpd.conf*). Si está en otro sitio se puede indicar con la opción -f. :

[root@localhost root]# /usr/sbin/httpd -f /etc/httpd/httpd.conf

Una vez arrancado se puede acceder a la documentación utilizando el navegador:

[root@localhost root]# http://localhost

Cuando el servidor arranque creará varios procesos hijos para atender las solicitudes. Si se inició apache como usuario root, el proceso apache continuará ejecutándose como root mientras los hijos cambiarán al usuario indicado en el httpd.conf.

Si se desea que el servidor continúe ejecutándose tras reiniciar el sistema debe incluirse en uno de los ficheros de inicio (por ejemplo: rc.local), en nuestro caso no es necesario.

Configuración del servidor

Entre los archivos de nuestro directorio de configuración *(/usr/local/apache/conf*) tenemos:

- **httpd.conf**: Establece los atributos generales del servidor: número de puerto en el que escucha, usuario que lo ejecuta, raíz del árbol de documentos, etc. En la actualidad se recomienda establecer todas las opciones de configuración necesarias en este fichero, y dejar los otros dos ficheros de configuración vacíos, esto simplifica la administración del servidor. Su configuración está bastante bien documentada en el propio archivo mediante comentarios.
- **srm.conf**: Se utilizaba para establecer la raíz del árbol de documentos, que ahora se establece en httpd.conf, por lo tanto no hay que utilizarlo.
- **access.conf**: Establecía la política de acceso. Ahora también puede establecerse desde httpd.conf, por lo que no es necesario utilizar este fichero.

Además de estos tres ficheros el comportamiento del servidor puede configurarse directorio a directorio mediante ficheros, .htaccess en los directorios – que contienen documentos html - a los que el servidor accede.

Es muy importante recordar que **cuando se cambia la configuración de los ficheros es necesario re-iniciar el apache** (apachectl restart) o enviarle una señal de SIGHUP con kill para que los cambios tengan efecto. Hay que estar seguro de que se envía la señal al proceso padre y no al hijo. El padre en general es el que tiene

un número de proceso menor. El id del proceso del padre está también en el fichero httpd.pid en el directorio log.

En esta práctica no entraremos en los aspectos de configuración de servidores virtuales, sino únicamente del servidor principal.

3.- Equipo y material necesario

Equipo del Laboratorio:

A. 1 Computadora con Sistema Operativo Linux Red Hat 9.0 ó superior.
B. El archivo apache_1.3.24.tar.gz
C. El archivo mod_ssl-2.8.8-1.3.24.tar.gz
D. El archivo openssl-0.9.6b.tar.gz

4.- Desarrollo

4.1 Modo de trabajar

Se trabajará en equipos de dos personas por computadora.

4.2 Instalación del servidor.

Aunque la última versión de apache disponible es la 2.0.36, para poder trabajar con SSL debemos utilizar la 1.3.24, que es la última para la que está disponible el módulo criptográfico mod_ssl.

Si queremos que tenga posibilidad de utilizar transferencias seguras con SSL necesitaremos al menos dos paquetes más: mod_ssl y OpenSSL, todos disponibles en un disco provisto por el administrador. En resumen:

Paquete: Apache
Versión: 1.3.24
Archivo: apache_1.3.24.tar.gz

Paquete: mod_ssl
Versión: 2.8.8-1.3.24
Descripción: la interfaz Apache con OpenSSL
Archivo: mod_ssl-2.8.8-1.3.24.tar.gz

Paquete: OpenSSL
Versión: 0.9.6b
Descripción: El Toolkit Open Source para SSL/TLS
Archivo: openssl-0.9.6b.tar.gz

Instalar sólo el paquete apache con SSL.

4.2.1 Encienda el sistema y elija la opción de cargar *Linux.*

4.2.2 Entre a sesión como usuario root. El administrador le proporcionará el password. (Ver figura No.1).



Figura No.1.

4.2.3 Abra un shell prompt seleccionando:

Main Menu => System Tools => Terminal.

Para salir del shell prompt teclee *exit* o presione *<ctrl>d* en el prompt. (Ver figura No.2).

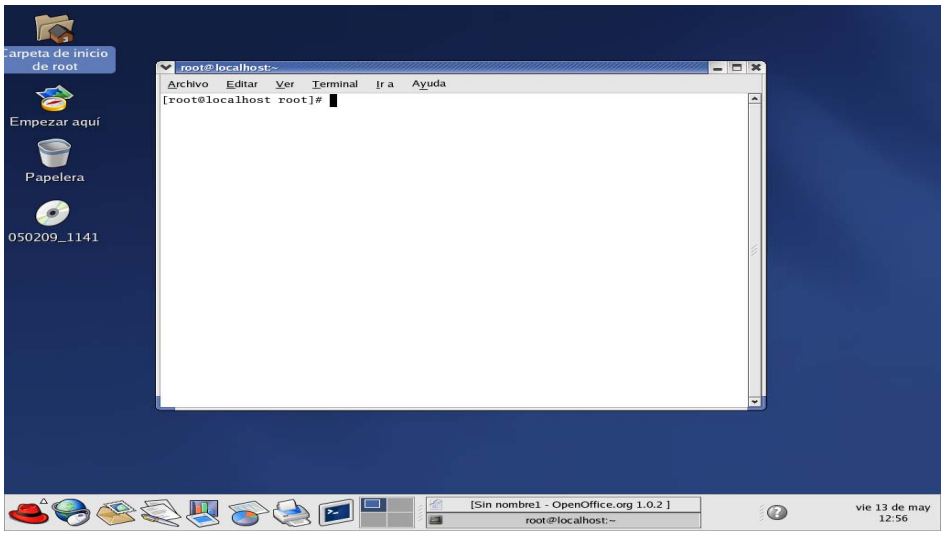


Figura No.2.

4.2.4 Descomprima los paquetes y ejecute el tar para situar los archivos en sus directorios.

Nota: Se recomienda hacer la instalación en */usr/local/apache* (Ver figura No.3).

[root@localhost root]# mkdir /usr/local/apache
[root@localhost root]# cd /usr/local/apache

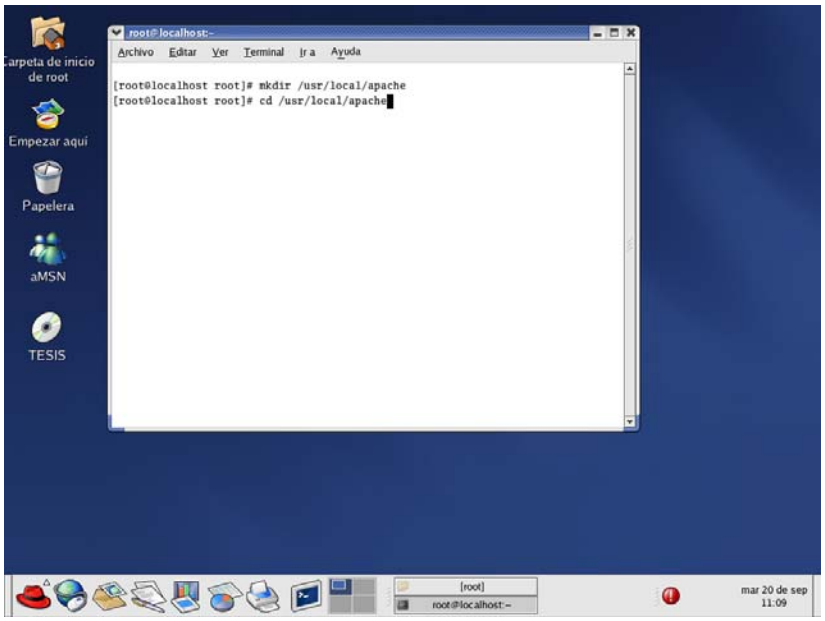


Figura No.3.

[root@localhost apache]# cp /mnt/cdrom/ apache_1.3.24.tar.gz /usr/local/apache
[root@localhost apache]# cp /mnt/cdrom/ mod_ssl-2.8.8-1.3.24.tar.gz /usr/local/apache
[root@localhost apache]# cp /mnt/cdrom/ openssl-0.9.6b.tar.gz /usr/local/apache
(Ver figura No.4).

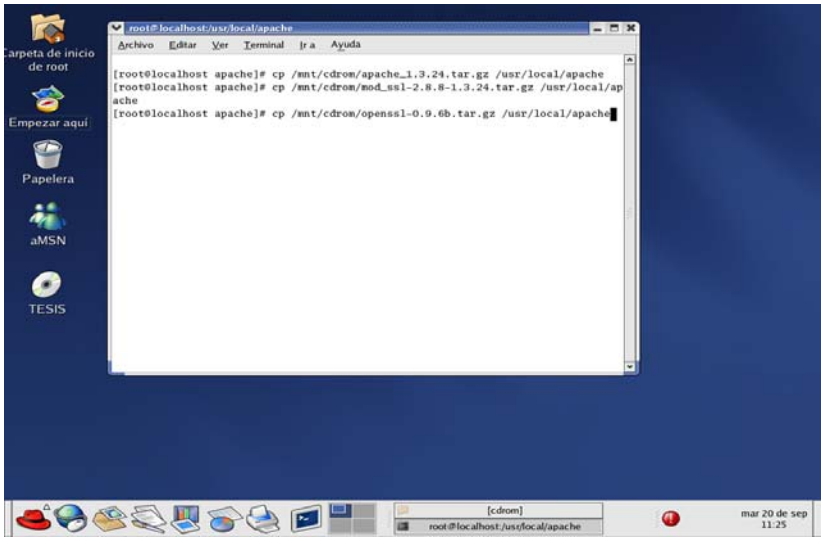


Figura No.4.

[root@localhost apache]# tar xzvf apache_1.3.24.tar.gz
(Ver figuras No.5 y 6).

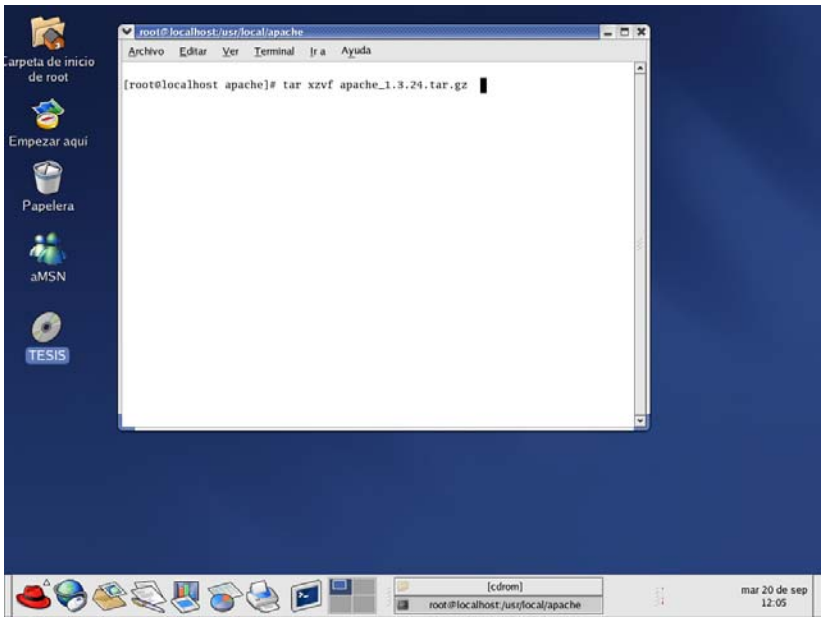


Figura No.5.

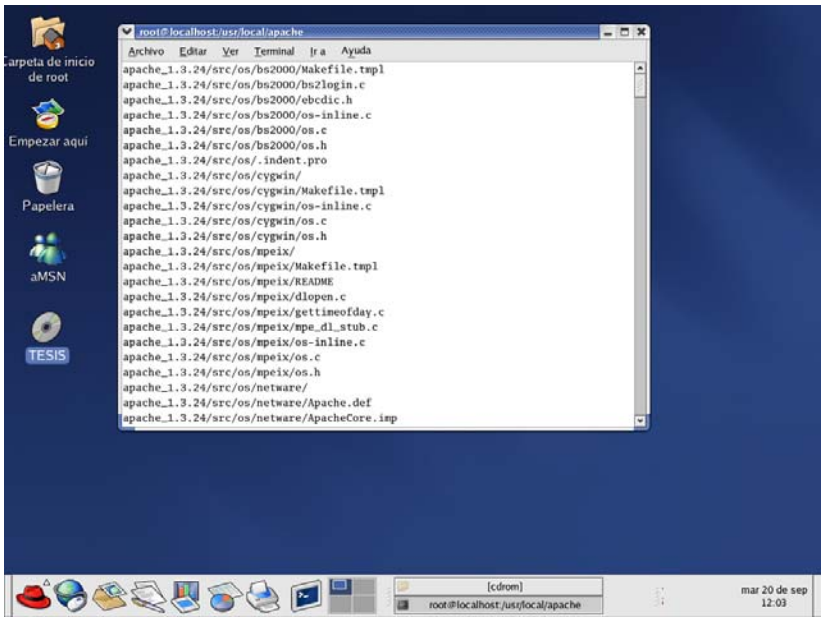


Figura No.6.

[root@localhost apache]# tar xzvf mod_ssl-2.8.8-1.3.24.tar.gz
(Ver figura No.7).

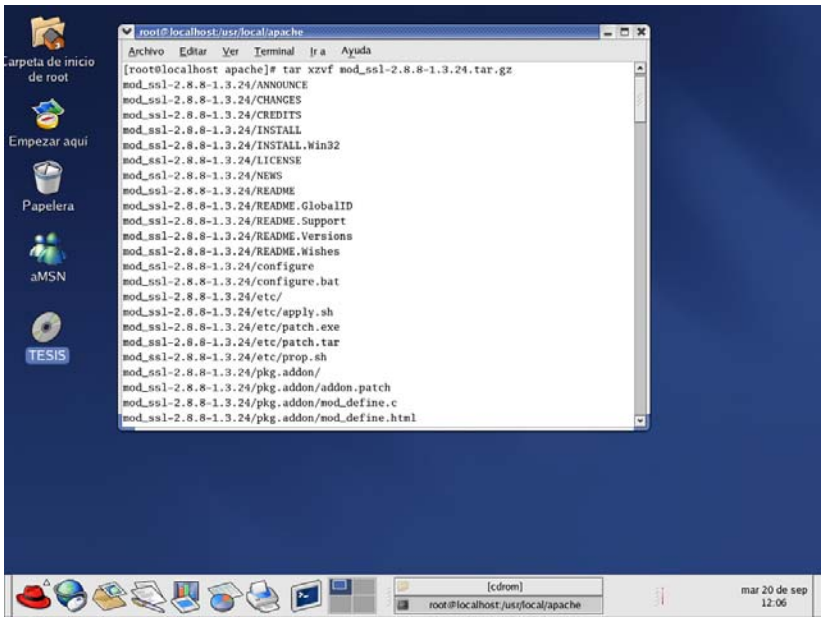


Figura No.7.

[root@localhost apache]# tar xzvf openssl-0.9.6b.tar.gz
(Ver figuras No.8 y 9).

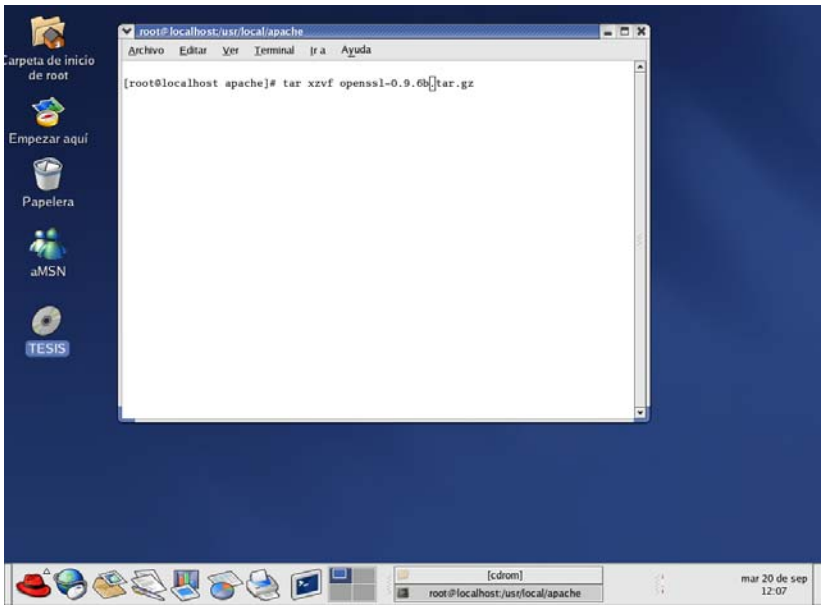


Figura No.8.

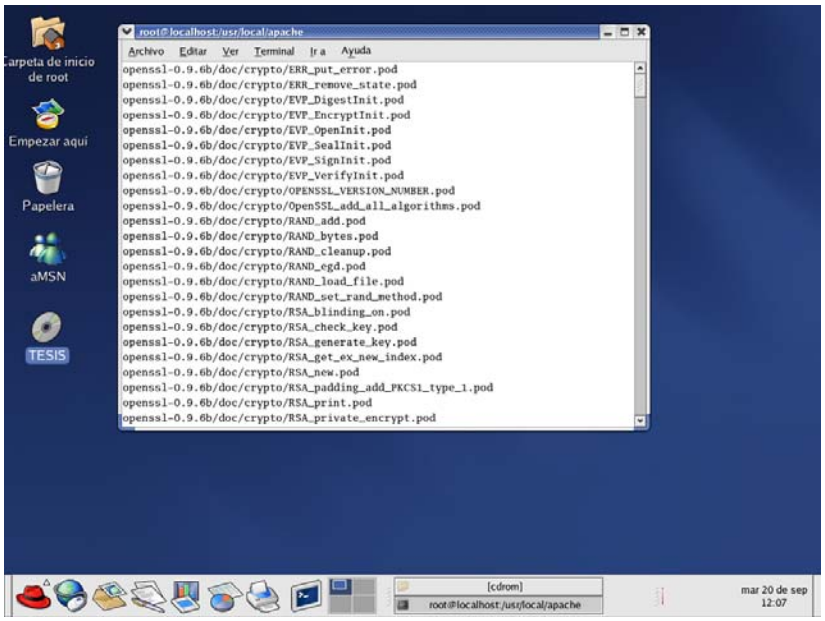


Figura No.9.

4.2.5 Configure y compile la librería *OpenSSL* :
(Ver figuras No.10 y 11).

[root@localhost apache]# cd openssl-0.9.6b
[root@localhost openssl-0.9.6b]# ./config

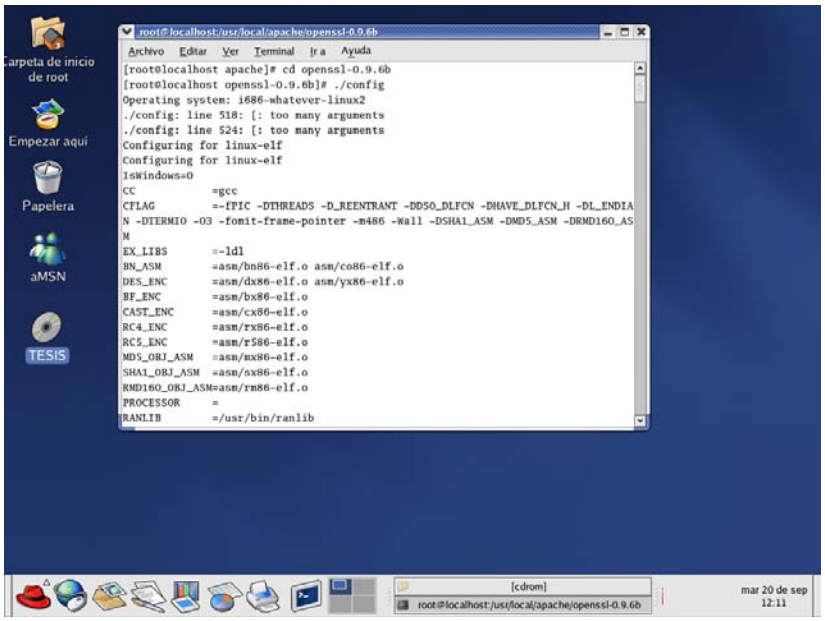


Figura No.10.

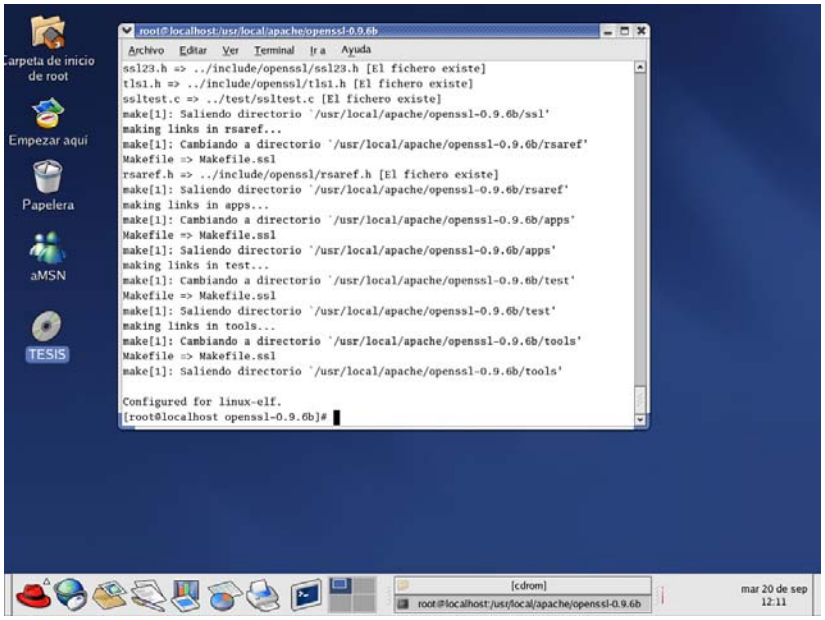


Figura No.11.

[root@localhost openssl-0.9.6b]# make (que tarda unos minutos)(Ver figuras No.12 y 13).

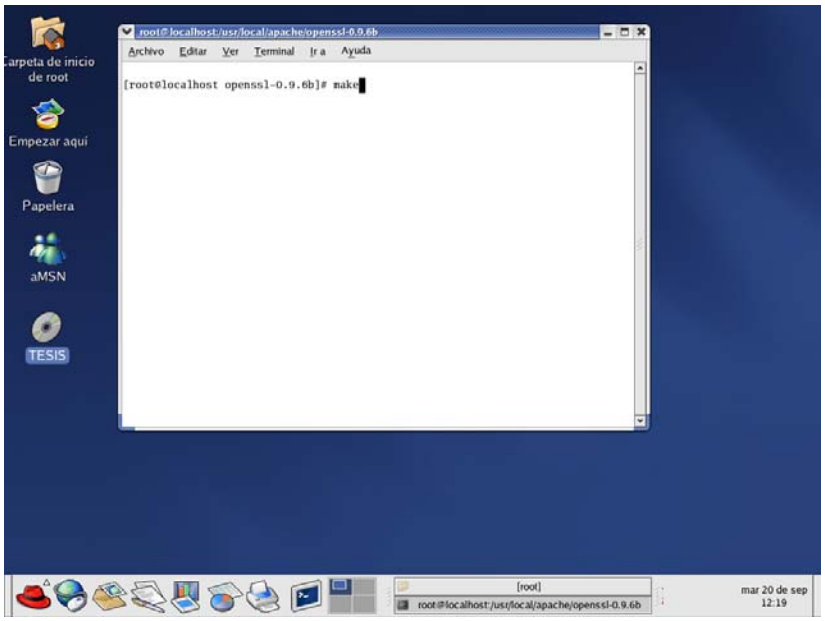


Figura No.12.

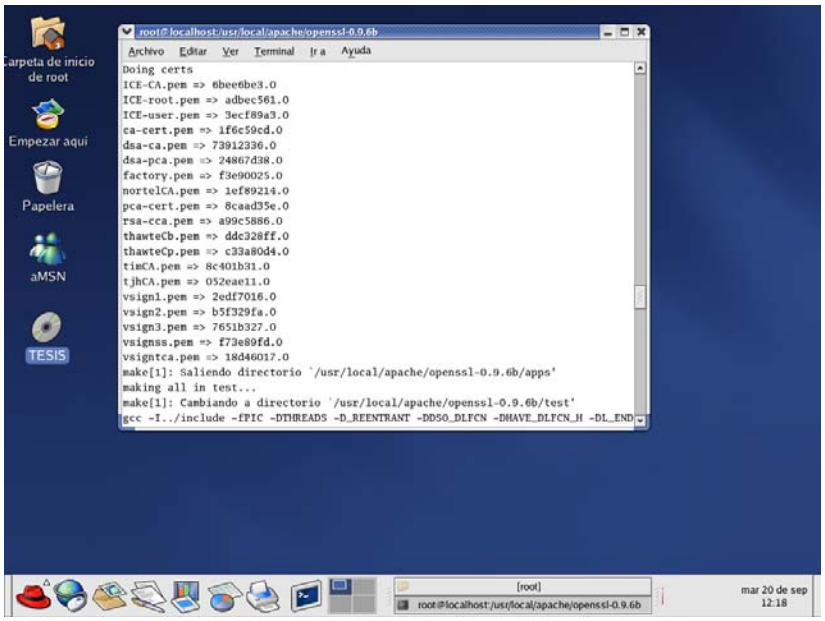


Figura No.13.

[root@localhost openssl-0.9.6b]# cd ..
(Ver figura No.14).

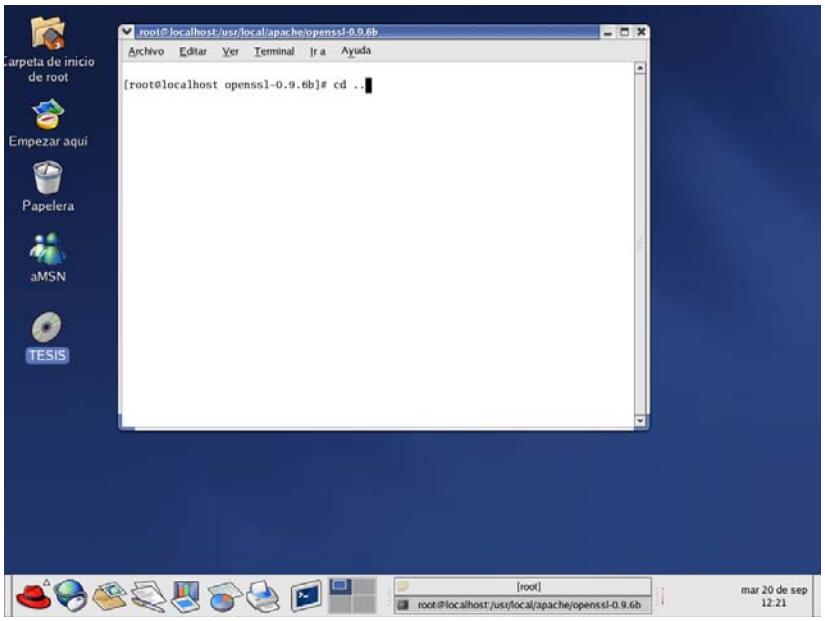


Figura No.14.

4.2.6 Por último aplique las extensiones fuente *mod_ssl* y los parches al Apache y compile Apache con *mod_ssl* y *OpenSSL* (Ver figura No.15):

[root@localhost apache]# cd mod_ssl-2.8.8-1.3.24

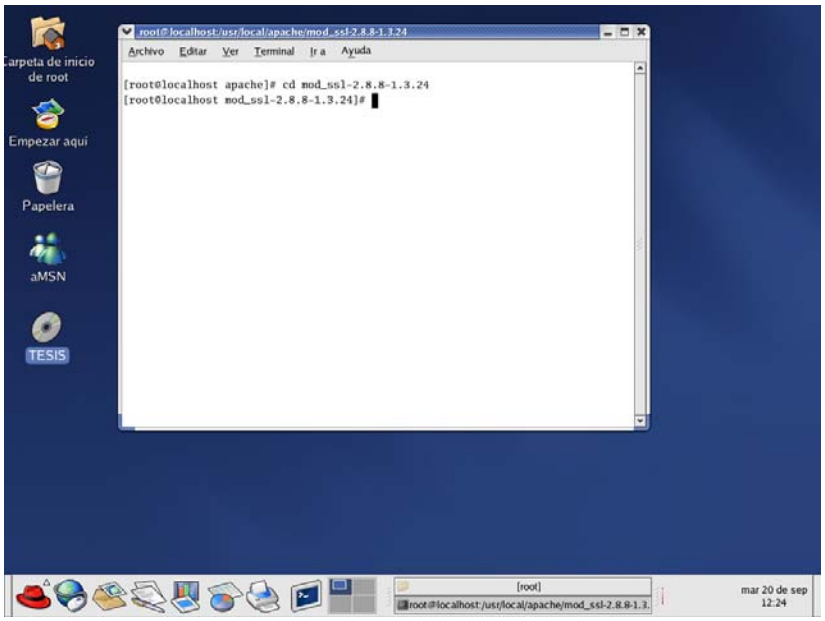


Figura No.15.

[root@localhost mod_ssl-2.8.8-1.3.24]# ./configure --with-apache=/usr/local/apache/apache_1.3.24 --with-ssl=/usr/local/apache/openssl-0.9.6b--prefix=/usr/local/apache
(Ver figura No.16).

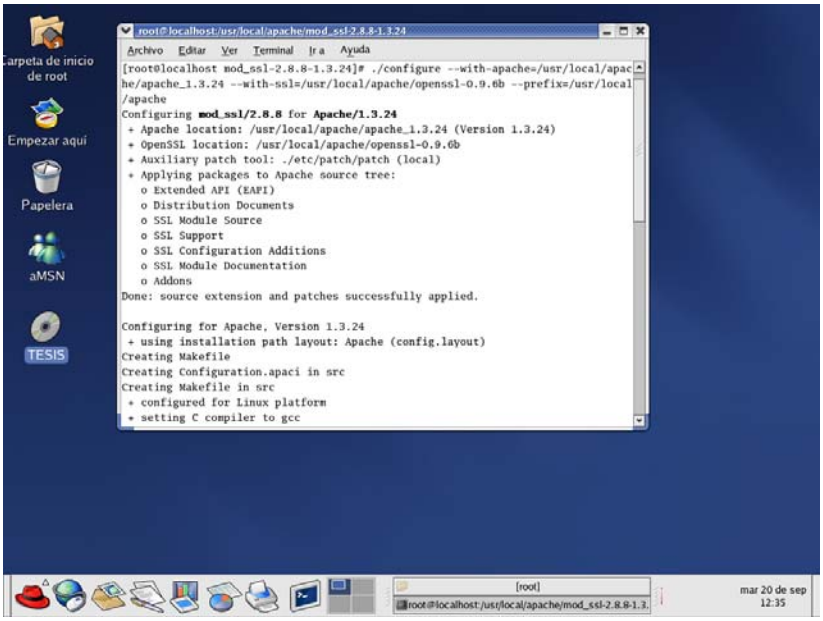


Figura No.16.

[root@localhost mod_ssl-2.8.8-1.3.24]# cd ..
[root@localhost apache]# cd apache_1.3.24
(Ver figura No.17).

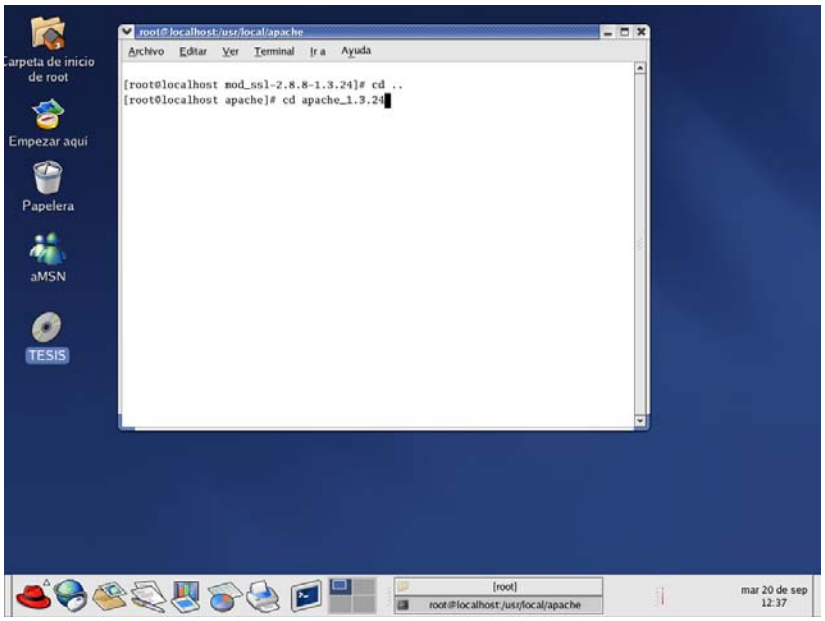


Figura No.17.

[root@localhost apache_1.3.24]# make (tarda unos minutos)
(Ver figuras No.18 y 19).

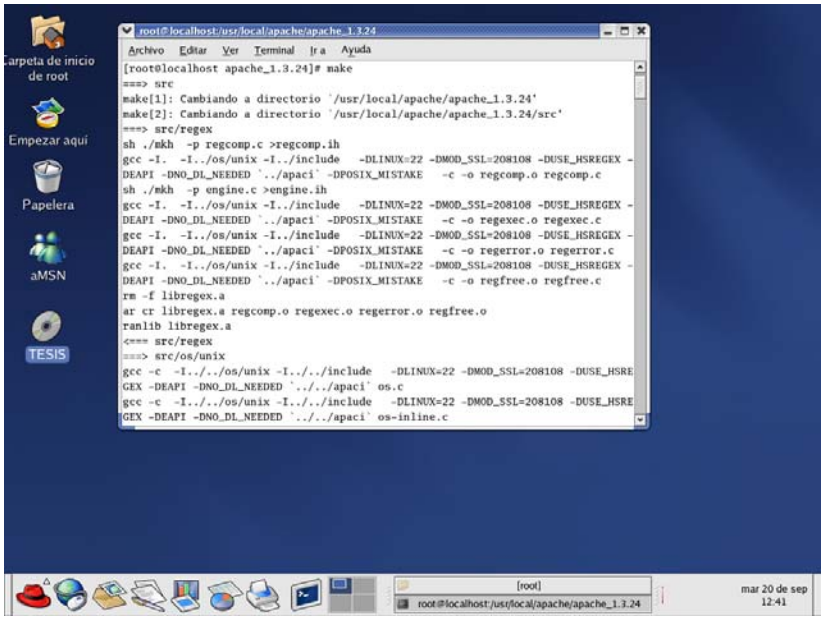


Figura No.18.

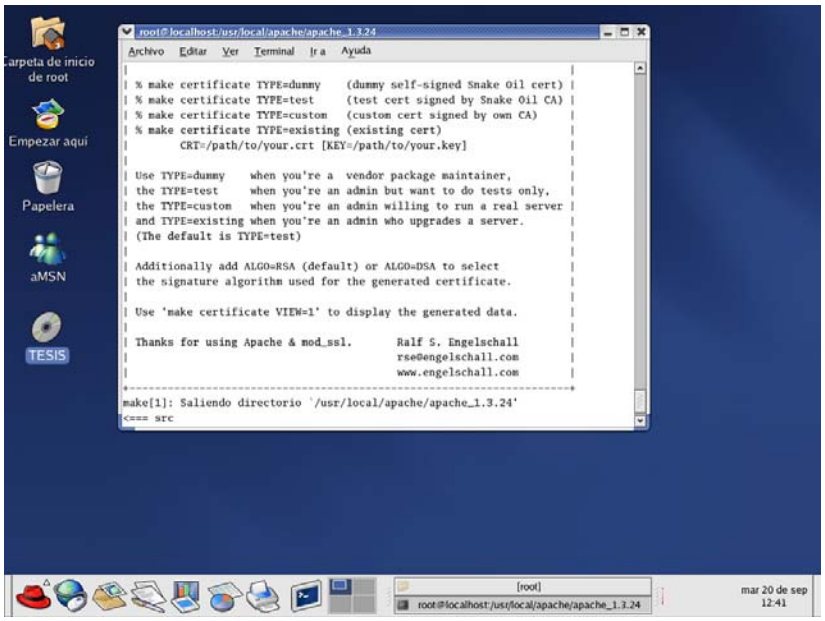


Figura No.19.

Al hacer el "*make certificate*" (que es una autenticación de la identidad del servidor) se pueden especificar varios tipos, en nuestro caso el adecuado es "*test*", que es el que toma por defecto si no le indicamos ninguna opción. Los valores que va pidiendo a continuación para generar el certificado no son críticos, con excepción del "*Common Name*" para el que es conveniente especificar el nombre de dominio del servidor, en nuestro caso www.ingenieria.unam.mx, en la parte de la frase clave PEM coloque "*ingenieria*" y confirmela. Para las demás opciones solo oprima Enter.

Una vez obtenido el certificado, la orden "*make certificate VIEW=1*" permite ver los datos generados. (Ver figuras No.20, 21 y 22).

[root@localhost apache_1.3.24]# make certificate

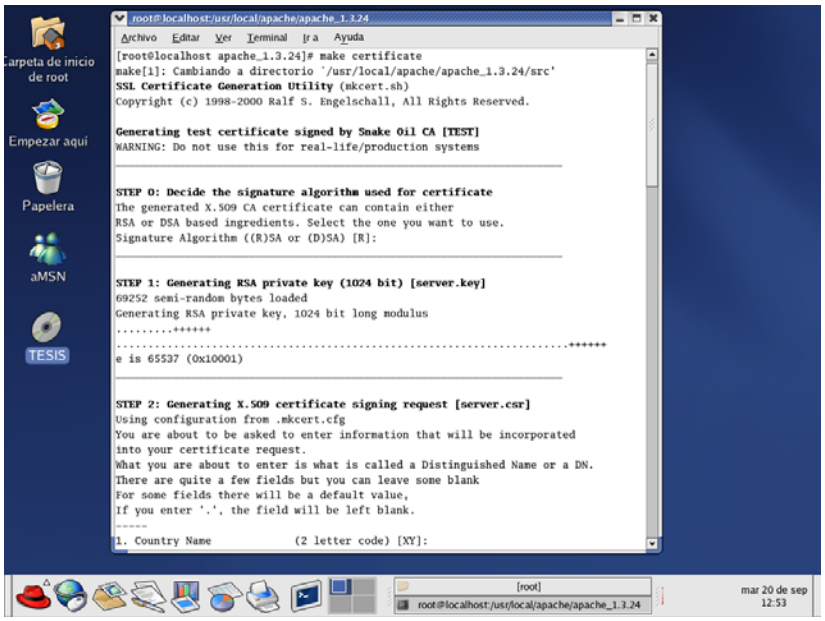


Figura No.20.

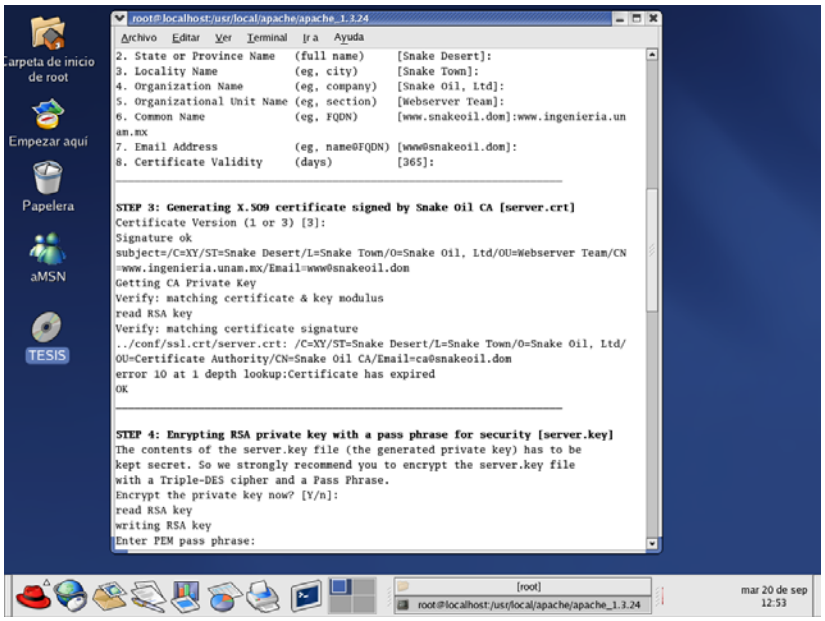


Figura No.21.

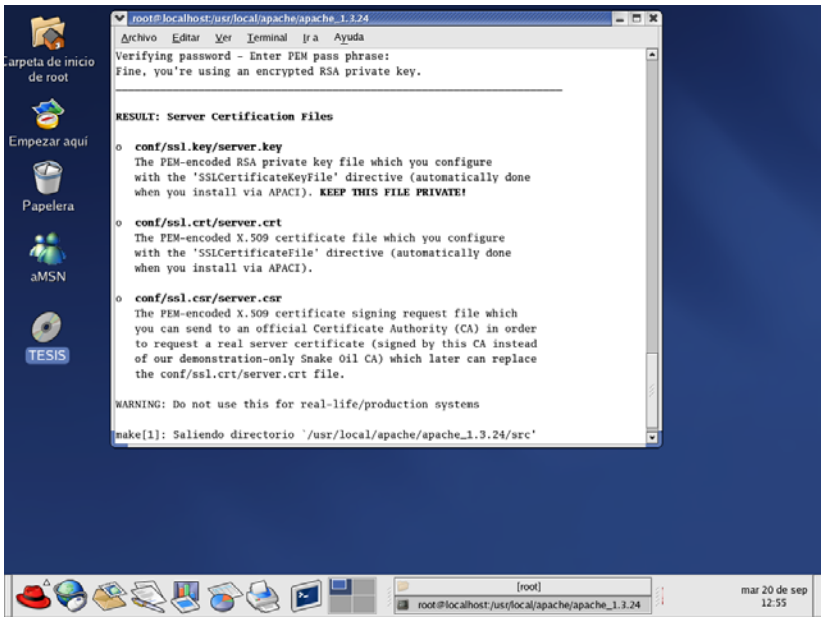


Figura No.22.

[root@localhost apache_1.3.24]# make install
(Ver figuras No.23 y 24).

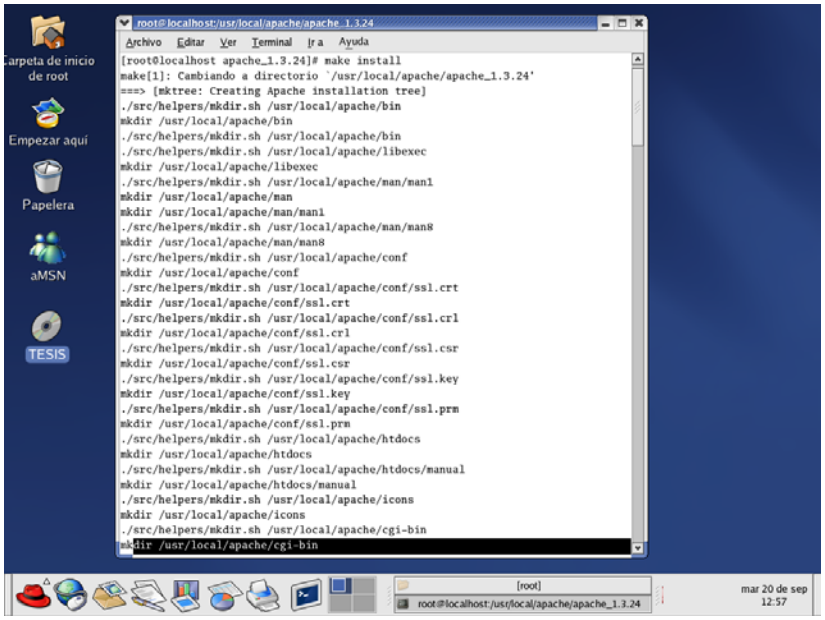


Figura No.23.

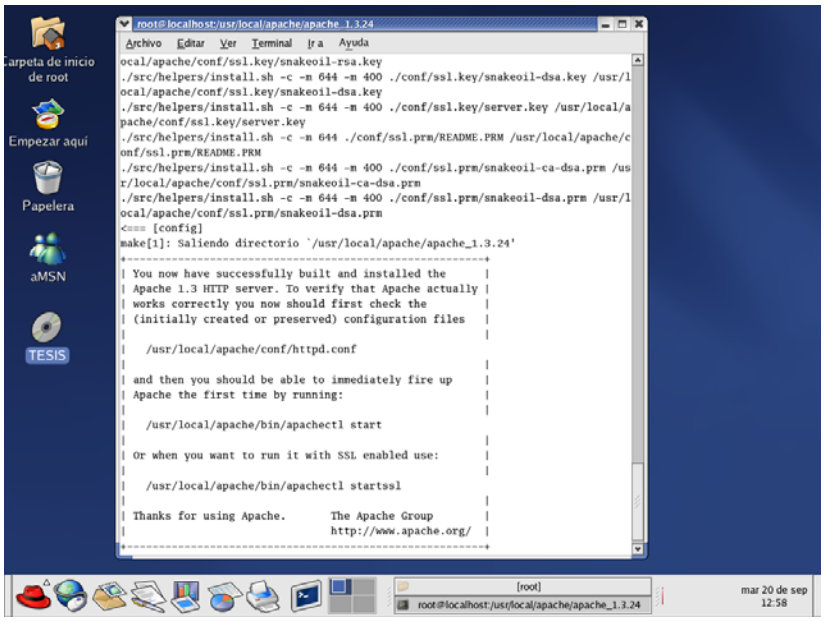


Figura No.24.

[root@localhost apache_1.3.24]# cd
(Ver figura No.25).

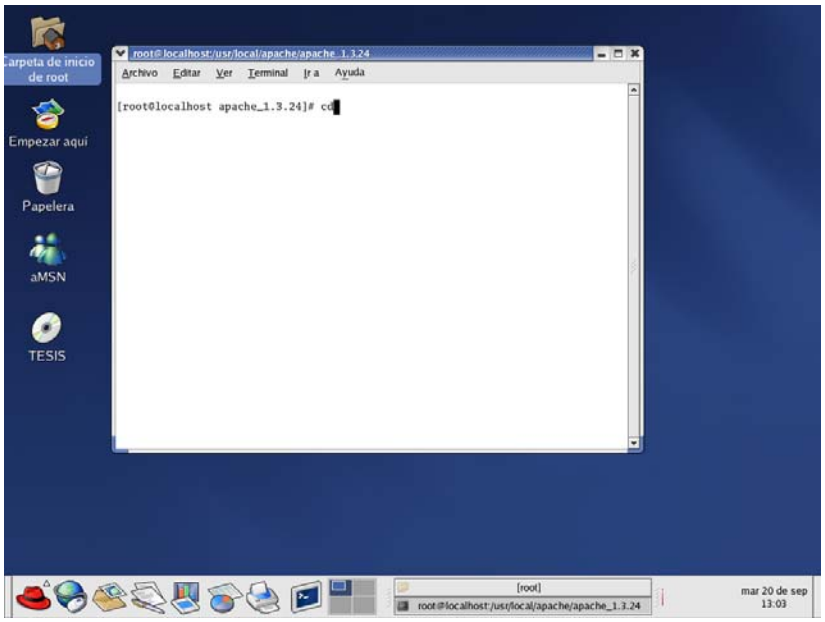


Figura No.25.

El paso siguiente será comprobar que los permisos de los directorios y ficheros del apache están como dicen las recomendaciones de seguridad:

4.3 Verificación de la instalación

Advertencias:

- Cuando se utiliza https, conectarse mediante localhost puede ocasionar problemas (o por lo menos una advertencia de seguridad) porque el nombre de dominio del ordenador (o alguno de sus alias) debe coincidir con el indicado al crear el certificado para el servidor.
- Si en un ordenador ya existe un servidor web ejecutándose -porque se ha lanzado en el arranque del sistema- tendrá reservado el puerto 80 y no dejará arrancar uno nuevo sobre el mismo puerto. El problema se detecta porque la conexión a través de SSL no funciona, mientras que la que utiliza sólo http trabaja con el servidor que ya estaba instalado.
- En caso de problemas una buena medida es consultar los registros de log (*/usr/local/apache/log*).
- Para reiniciar el servidor:

  [root@localhost root]# /usr/local/apache/bin/apachectl restart

4.3.1 Arranque el servidor sin SSL (sólo puede utilizarse http) (Ver figura No.26):

[root@localhost root]# /usr/local/apache/bin/apachectl start

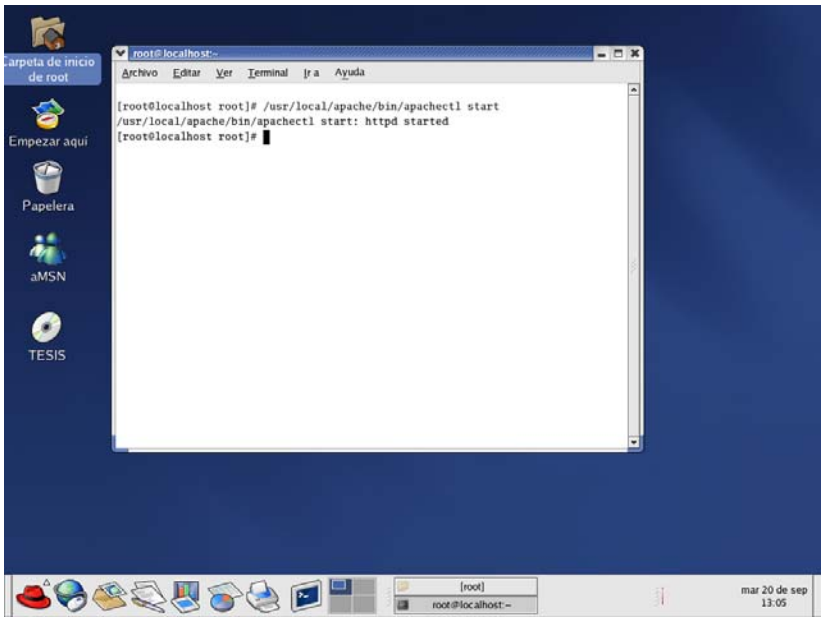


Figura No.26.

4.3.2 Ejecute Mozilla con http://localhost (Ver figura No.27).

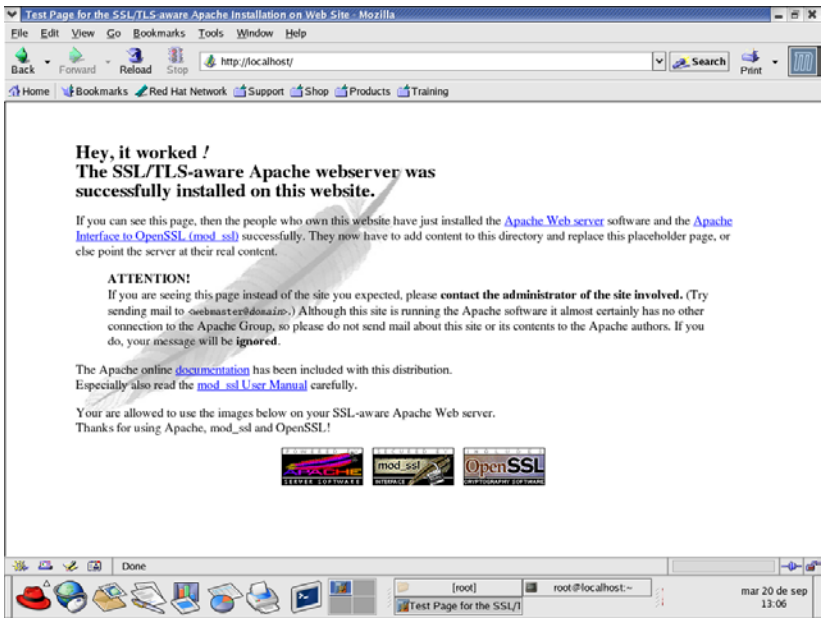


Figura No.27.

4.3.3 Detenga el servidor, antes de lanzar el servidor que incluye SSL (Ver figura No.28):

[root@localhost root]# /usr/local/apache/bin/apachectl stop

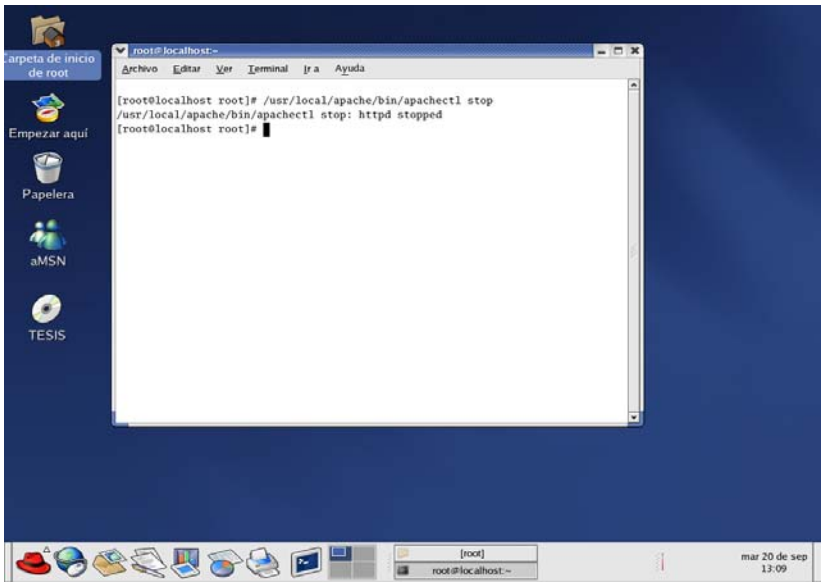


Figura No.28.

4.3.4 Arranque el servidor con SSL (pueden utilizarse http y https), cuando pida la frase clave recuerde que es *ingenieria* (Ver figura No.29):

[root@localhost root]# /usr/local/apache/bin/apachectl startssl

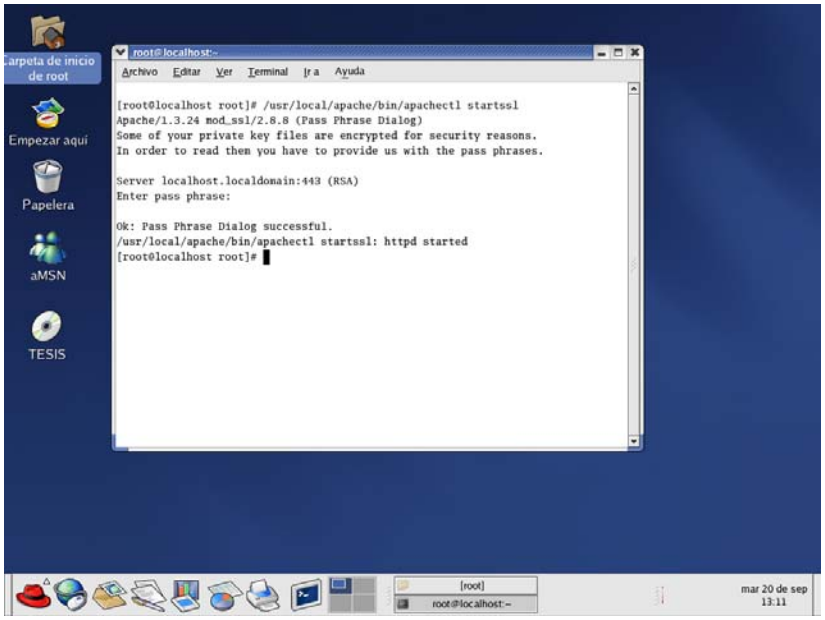


Figura No.29.

4.3.5 Ejecute Mozilla con http://www.ingenieria.unam.mx (Ver figura No.30).



Figura No.30.

4.3.6 Ejecute Mozilla con https://www.ingenieria.com.mx (Ver figura No.31).



Figura No.31.

El servidor advertirá que no conoce la autoridad certificadora que emitió el certificado, si en la computadora solo existe el servidor web que hemos instalado. Pero para nuestro caso, como podrá ver, la conexión a través de SSL no funciona porque en la computadora ya existe un servidor web ejecutándose.

4.3.7 Detenga el servidor (Ver figura No.32):

[root@localhost root]# $ /usr/local/apache/bin/apachectl stop

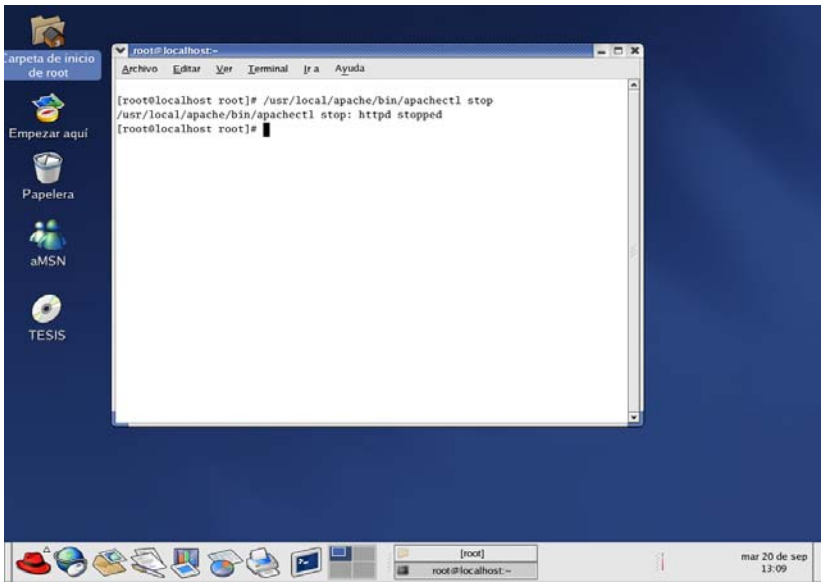


Figura No.32.

4.4 Configuración del servidor principal

4.4.1 Abra el archivo de configuración httpd.conf con el editor de texto vi, y modifíquelo de manera que las siguientes directivas (tanto las del entorno global como las del servidor principal, las demás déjelas tal cual) queden de este modo (Ver figuras No.33, 34 y 35):

[root@localhost root]# $ vi /usr/local/apache/conf/httpd.conf

httpd.conf -- Apache HTTP server configuration file

Section 1: Global Environment
ServerType standalone

ServerRoot "/usr/local/apache"

PidFile /usr/local/apache/logs/httpd.pid

ScoreBoardFile /usr/local/apache/logs/httpd.scoreboard

```
Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 15

MinSpareServers 5
MaxSpareServers 10

StartServers 5

MaxClients 150

MaxRequestsPerChild 0

### Section 2: 'Main' server configuration
Port 80

<IfDefine SSL>
Listen 80
Listen 443
</IfDefine>

User nobody
Group nobody

ServerAdmin root@localhost.localdomain

DocumentRoot "/usr/local/apache/htdocs"

<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

<Directory "/usr/local/apache/htdocs">
```

```
    Options Indexes FollowSymLinks MultiViews

    AllowOverride None

    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_userdir.c>
    UserDir public_html
</IfModule>


<IfModule mod_dir.c>
    DirectoryIndex index.html
</IfModule>

AccessFileName .htaccess

<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy All
</Files>

UseCanonicalName On

<IfModule mod_mime.c>
    TypesConfig /usr/local/apache/conf/mime.types
</IfModule>

DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile /usr/local/apache/conf/magic
</IfModule>

HostnameLookups Off
```

ErrorLog /usr/local/apache/logs/error_log

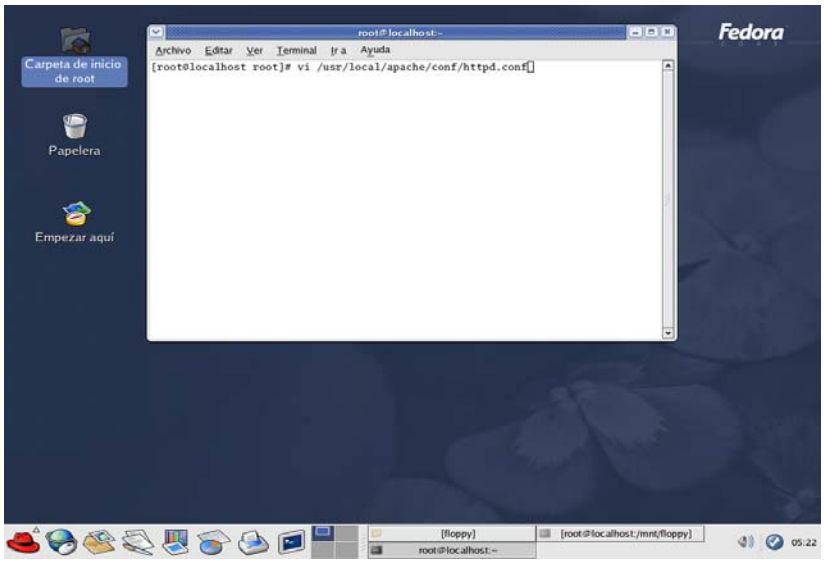


Figura No.33.

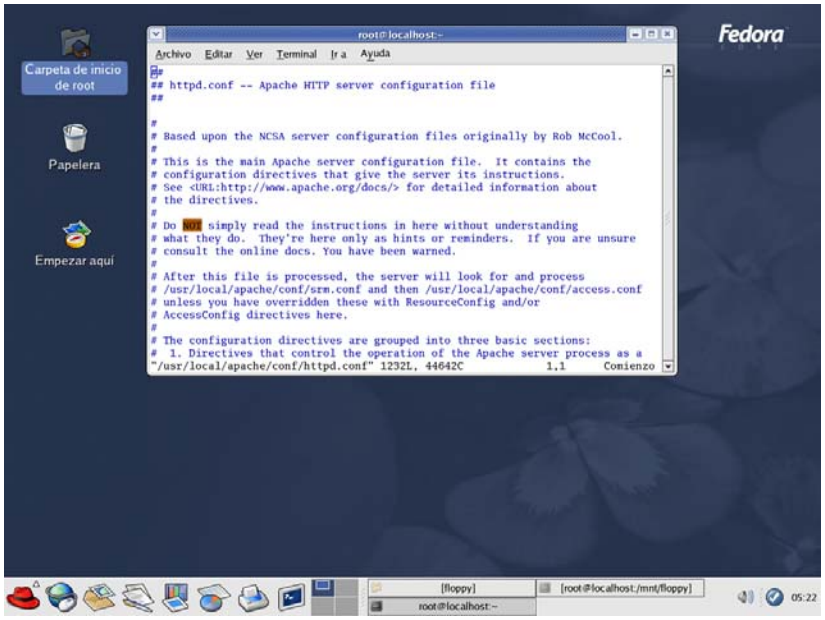


Figura No.34.

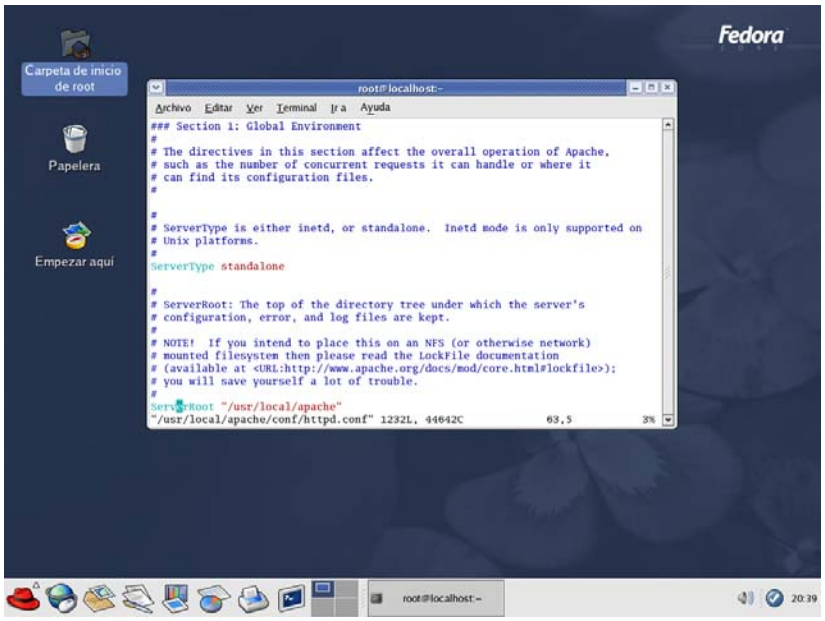


Figura No.35.

4.4.2 Reinicie el servidor apache para que los cambios sean efectuados (Ver figura No.36).

[root@localhost root]# $ /usr/local/apache/bin/apachetl restart

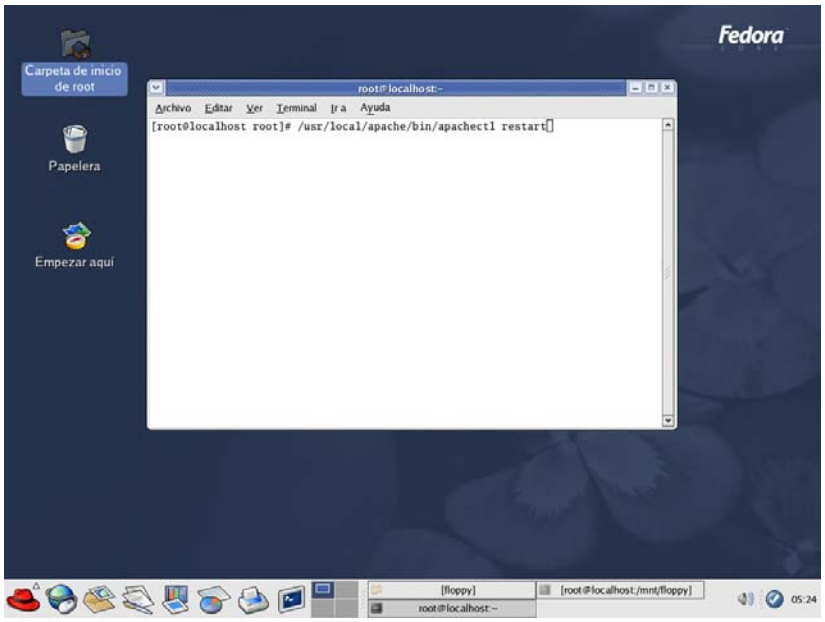


Figura No.36.

5.- Anote sus Conclusiones u Observaciones; revisando los objetivos planteados al inicio de la práctica:

PRÁCTICA 10
Aplicación Web Apache
Cuestionario Previo

Nombre del alumno: _____
Gpo. de Laboratorio: _____ ***Gpo. de Teoría:*** _____
Calificación: _____

1.-¿Cuáles son las aplicaciones típicas de la capa 7 del Modelo OSI?

2.- Explique ¿qué hace http y https?

3.- Investigue ¿qué es SSL?

4.- Investigue las características generales del servidor Web IIS (*Internet Information Services*).

5.- Investigue las características generales de otros servidores de Web como, Tomcat, Zope, etc.