

UNIVERSIDAD DON BOSCO
FACULTAD DE INGENIERIA
ESCUELA DE INGENIERIA EN COMPUTACION



"Sistema de Simulación para el aprendizaje de Redes de Área Local (LAN) y Redes de Área Ampla (WAN) Básicas"

PRESENTADO POR:

Gilberto René Huezo Dueñas

Martín Alfredo Miranda Oliva

Roberto Mauricio Rodríguez Martínez

TRABAJO DE GRADUACION PARA OPTAR AL GRADO DE:
INGENIERO EN CIENCIAS DE LA COMPUTACION.

ASESOR:

ING. ERICK ALFREDO FLORES

SAN SALVADOR, EL SALVADOR, CENTROAMÉRICA.

UNIVERSIDAD DON BOSCO



RECTOR:

ING. FEDERICO MIGUEL HUGUET

VICERRECTOR:

Pbo. VICTOR BERMUDEZ

SECRETARIO GENERAL:

LIC. MARIO RAFAEL OLMOS

DECANO DE LA FACULTAD DE INGENIERIA:

ING. ERNESTO GODOFREDO GIRON

JURADO EVALUADOR:

LIC. MAURICIO COTO

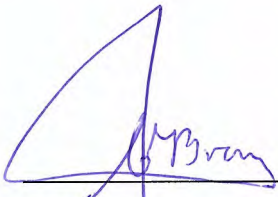
ING. CARLOS BRAN

UNIVERSIDAD DON BOSCO

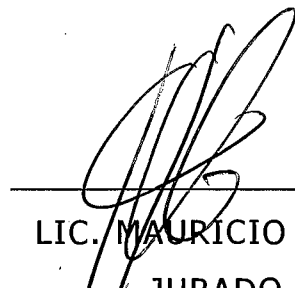
FACULTAD DE INGENIERIA

JURADO EVALUADOR DEL TRABAJO DE GRADUACIÓN

**SISTEMA DE SIMULACIÓN PARA EL APRENDIZAJE DE REDES DE
ÁREA LOCAL (LAN) Y REDES DE ÁREA AMPLIA (WAN) BÁSICAS**



ING. CARLOS BRAN
JURADO



LIC. MAURICIO COTO
JURADO



ING. ERICK ALFREDO FLORES
ASESOR

AGRADECIMIENTOS

Roberto Mauricio Rodríguez Martínez

Toma en cuenta a Dios en todas tus acciones, y él te ayudará en todo. Proverbios 3:6 (BLS)

Dios mío, enséñame a vivir como tú siempre has querido. Tú eres mi Dios y salvador, y en ti siempre confío. Salmos 25:5 (BLS)

Porque yo soy el Señor, tu Dios, que sostiene tu mano derecha; yo soy quien te dice: No temas, yo te ayudaré. Isaías 41:13

Encomienda al Señor tu camino; confía en él, y él actuará. Salmos 37:5

- Mi Dios, no tengo palabras para describir mi gratitud por haber enviado a tu Hijo Jesucristo a morir por mí, obsequiarme la salvación, vida eterna y por estar conmigo en esta etapa de mi vida en la cual he palpado Tu inagotable misericordia, provisión, sabiduría, amor y fuerza. Reconozco que sin Ti nada soy y este logro no tendría sentido si Tu no estuvieras en mi corazón.
- Gracias a mi papá Roberto Rodríguez por todo su amor, sus oraciones, apoyo incondicional, consejo acertado y su ejemplo que siempre ha retado mi vida.
- Gracias a mi mamá Emma Lidia Martínez de Rodríguez por todo ese amor de madre que me fortaleció, apoyo inmerecido, su cuidado en todo momento, sus oraciones sin cesar y paciencia que siempre me ha dado.
- Gracias a mi hermana Karla y mi cuñado David por sus oraciones y apoyo.
- Gracias a mi novia Lissette Calderón por todo su cariño, sus oraciones, por su ayuda para suavizar lo duro de este proceso, por sus palabras de aliento en los momentos más difíciles (El hombre prepara el caballo para entrar en batalla, pero el Señor es quien da la victoria. Proverbios 21:31(DHH)) y por su amor que alegro y dio colorido a mi vida.
- Gracias a todos mi amigos por sus palabras de animo, sus oraciones, amistad sincera y por todo el apoyo que han brindado a mi vida.

AGRADECIMIENTOS

- Gracias a mis compañeros de tesis Gilberto Huevo y Martín Miranda por haberme concedido la oportunidad desarrollar este proyecto con ellos, por su colaboración y amistad.
- Gracia a Roberto Peña por su gran apoyo y por poner la nota de humor en este proceso.
- Gracias a todos mis profesores, grupo evaluador y asesor que fueron parte fundamental para obtener este logro.
- Gracias a todos aquellos que de alguna forma estuvieron involucrados en mi carrera académica.

Mi deseo es que Dios a todos les recompense y prospere en grande por su apoyo.

Y ahora, gloria sea a Dios, que puede hacer muchísimo más de lo que nosotros pedimos o pensamos, gracias a su poder que actúa en nosotros.

Efesios 3:20 (DHH)

AGRADECIMIENTOS

Gilberto René Huevo Dueñas.

- Gracias en primer Lugar a Dios, por la guía, fortaleza, perseverancia, animo, esperanza, sabiduría, firmeza que me proporciono a lo largo de mi carrera, por su compañía en los momentos felices, por su soporte en mis momentos de angustia, por mantenerme al lado, a mis Ángeles guardianes, mis padres. Hay un pensamiento que resume mi sentir: Es una anécdota de una persona y Jesús que recorrieron juntos un camino, y al final, cuando la persona miró atrás le pregunta a Jesús: ¿Señor, cómo es posible que solo se vean unas huellas en algunos tramos, que paso con las otras? ¿Dónde estabas tú? Hijo-le respondió el Señor-, cuando aparecían unas solamente, eran las mías, porque yo te llevaba en mis hombros.
- Gracias a mi padre Gilberto Huevo Hernández, la persona que mas admiro, por inspirarme al transcurso de mi carrera, y de mi vida, por darme una guía a la cual apegarme, seguir, y desear sobrepasar, por su fe a ojos cerrados, puesta en mi a lo largo de todos mi estudios; por su esfuerzo incondicional para verme profesional; infinitas gracias.
- Gracias a mi madre, a mi amiga, Olga Patricia de Huevo, a la persona con mas créditos por el hecho de inculcarme lo que soy, para poder llegar a estas instancias, por hacerme crecer como persona y como profesional, a lo largo de todos mis niveles de estudio; no hay palabras para agradecerlo.
- Gracias a Ricardo Huevo, mi hermano, por apoyarme en las diferentes necesidades que tuve en mis años de estudio, por soportarme en mis momentos difíciles de estudio y por ser mi animo e inspiración en silencio.
- Gracias a Roberto Peña, por su apoyo incondicional, por su tiempo sacrificado, por su gran paciencia y por su contagiosa alegría con la cual me apoyo en mi proceso de trabajo de graduación, por lo cual pude aprender a ver las cosas de mi carrera desde otro punto de vista, en verdad, mil gracias.
- Gracias a Ricardo Alvarado, por ser mi soporte en los momentos que siempre necesite en mi proceso de trabajo de Graduación, por hacerme crecer profesionalmente, por depositar en mi su confianza, por poder depositar en él mi confianza plena, a lo largo de dicho proceso, y por poderme graduar de mi carrera con un nivel mayor de profesionalismo al que tenia al egresar de esta. Por siempre agradecido.

AGRADECIMIENTOS

- Gracias a Patricia Renderos, por ser un apoyo día a día, a lo largo del trabajo de graduación, por su preocupación, por su dedicación, por su ayuda, por animarme cuando lo necesite, por presionarme a trabajar duro cuando lo requerí, por sacrificar su tiempo, y todo esto por verme triunfar. Gracias, mil gracias.
- Gracias a Ing. Xenia Rosa, por su ayuda durante la parte de mi estudio de ingeniería que más lo necesite, por su apoyo y preocupación en el transcurso de mi trabajo de graduación. Muchas, pero muchas gracias por tu apoyo.
- Gracias a Ing. Daniel Alvarenga, por ser la otra persona que me apoyo cuando mas lo necesite en mi carrera, y seguir siendo un apoyo en mi crecimiento profesional; gracias por tu apoyo.
- Gracias a Martin Miranda, uno de mis dos compañeros, no solo de tesis, sino a lo largo de toda mi carrera, gracias por apoyarme, ayudarme incondicionalmente, animarme, soportarme durante todo este tiempo, durante las buenas y las malas de este largo camino, por tu tiempo, una de las personas que tengo que darle gracias y mil gracias.
- Gracias a Roberto Rodríguez, mi otro compañero durante toda mi carrera, gracias por enseñarme durante todo este tiempo tu mejor virtud en el campo profesional, La perseverancia, nunca rendirse; gracias por tu apoyo incondicional, por tu paciencia hacia mí, durante los momentos mas difíciles de mi carrera, por celebrar conmigo los triunfos de esta, y ser mi mejor influencia durante toda mi carrera.
- Gracias a todas las personas que de una forma u otra me apoyaron a lo largo de todo mi carrera Universitaria.
- Y una gratitud muy especial a mi tercer ángel guardián que me cuida desde el cielo, Blanca America Dueñas.

AGRADECIMIENTOS

Martín Alfredo Miranda Oliva.

- Primeramente, dar gracias a Dios por haberme dado la vida y llegar hasta este momento tan importante. Le estoy agradecido por darme la inteligencia, la fortaleza y la paciencia necesaria para poder culminar este proyecto y permitir que saliera adelante todos los días.
- A mi papá, Don Ernesto Miranda, por su ejemplo, por haberse esforzado para que yo terminara la universidad y salir adelante, por su apoyo y sus consejos y todo lo que me ha enseñado.
- A mi mamá, Teresa de Miranda Oliva, por todo su apoyo, sus oraciones, por confiar y creer en mí, y por guiarme en esta vida, estar conmigo.
- A mis hermanos, Carlos y Claudia, por preocuparse siempre por mí, y por su apoyo. Y sin faltar, mi sobrino Erick.
- A mi Tía Esperanza, por cuidarme y preocuparse por mí día a día, por sus atenciones, por su apoyo, consejos y por todo lo que ha hecho por mí. Le estaré agradecido toda la vida.
- A mi Tío Amadeo, por su paciencia, por haberme permitido estar en su casa durante todos estos años y así, yo pudiera culminar este proyecto.
- A todos mis amigos por sus muestras de apoyo, su amistad sincera y desinteresada, por hacer que mi vida no sea aburrida.
- A mis compañeros de tesis, Gilberto y Roberto, por haber compartido tantas alegrías y tristezas, y que mantuvieron esa perseverancia para continuar cada día.
- A Roberto Peña, por habernos ayudado en todo lo que pudo y por todos los sacrificios que tuvo que hacer por nosotros. Te estoy realmente agradecido.
- A la Ing. Mercedes Cáceres, por habernos ayudado en todo este proceso. Muchas gracias.

AGRADECIMIENTOS GRUPALES

- Tenemos un gran agradecimiento para nuestro Asesor Ing. Erick Flores, por su paciencia, disposición, recomendaciones, guía, confianza que nos transmitió a lo largo no solo de este proceso, sino en varias instancias de nuestra carrera en las cuales nos abocamos a él; así como también por ayudarnos a incrementar nuestros conocimientos aun mas allá de las fronteras que nuestro trabajo exigía.
- Agradecemos a Lic. Elisa de Sandoval, por su siempre disponibilidad de acceso a los recursos ideales para la realización de todo el proceso de nuestro trabajo de Graduación. Así como también su constante interés por los resultados de este. Mil Gracias de todo corazón.
- Agradecemos a Ing. Mercedes Cáceres, por su valiosísima ayuda de principio a fin, durante este largo proceso, y su interés en que este, lo lográramos de la mejor manera. Muchas, pero muchas gracias.
- Agradecemos al jurado Evaluador, por se críticos constructivos para con nosotros.

ÍNDICE

INTRODUCCIÓN

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1. OBJETIVOS.....	2
1.1. OBJETIVO GENERAL.....	2
1.2. OBJETIVOS ESPECIFICOS.....	2
2. ANTECEDENTES DEL TEMA O PROBLEMA.....	3
3. IMPORTANCIA Y JUSTIFICACIÓN DEL TEMA.....	6
4. PROYECCIÓN SOCIAL.....	7
5. DEFINICION DEL TEMA	8
6. ENFOQUE CON QUE SE ABORDARA EL TEMA	9
7. ALCANCES	10
8. LIMITACIONES	12

CAPITULO II

ESTUDIO DE FACTIBILIDAD

1. FACTIBILIDAD TÉCNICA.....	15
2. FACTIBILIDAD ECONÓMICA.....	51
3. FACTIBILIDAD OPERATIVA.....	54

CAPITULO III
METODOLOGÍA DE LA INVESTIGACION

1. METODOLOGIA DE LA INVESTIGACIÓN	56
2. TECNICAS A UTILIZAR EN LA INVESTIGACIÓN	59
3. INVESTIGACION PRELIMINAR Y RECOLECCION DE INFORMACIÓN ...	60

CAPITULO IV
DESCRIPCIÓN DEL PROYECTO

1. SITUACION ACTUAL	63
2. RESULTADOS ESPERADOS.....	64

CAPITULO V
MARCO TEORICO

1. MARCO HISTORICO	66
2. MARCO CONCEPTUAL	71

CAPITULO VI
DESARROLLO DEL DISEÑO

1. DISEÑO DEL SISTEMA	154
1.1. DISEÑO DE MENUS	154
1.2. DISEÑO DE PANTALLAS.....	158
1.3. DIAGRAMAS DE RED	162
1.4. DISEÑO DE LABORATORIOS	169
1.5. DESCRIPCIÓN DE COMANDOS	181
1.6. ESTRUCTURA DE ARCHIVOS.....	195
2. DESARROLLO DEL SISTEMA	198
2.1. COMPONENTES	198
2.1.1. VENTANA PRINCIPAL	198

2.1.2.	<i>HISTORIAL</i>	199
2.1.3.	<i>CONFIGURACIÓN DE IP DEL HOST</i>	200
2.1.4.	<i>CONSOLA DEL HOST</i>	203
2.1.5.	<i>CONSOLA DEL SWITCH</i>	204
2.1.6.	<i>CONSOLA DEL ROUTER</i>	206
2.1.7.	<i>ACERCA DE.....</i>	207
2.2.	MENUS.....	208
2.3.	DISPOSITIVOS	210
2.4.	LABORATORIOS.....	214
	PROCESOS	222
CONCLUSIONES		226
BIBLIOGRAFIA		227
GLOSARIO DE TERMINOS		229
ANEXOS		
1.	MANUAL DE USUARIO	243
2.	MANUAL DEL PROGRAMADOR.....	278

INTRODUCCIÓN

En la década de los '90 la Internet comenzó a crecer en tal medida que en estos días las personas tienen acceso a este servicio desde cualquier rincón en el que se encuentren, y es por eso que se ha convertido en una de las principales herramientas de comunicación y comercio.

La Internet se forma gracias a la interconexión de muchas redes de computadoras alrededor del mundo, con el fin de intercambiar información. La conexión física de estas computadoras se realiza a través de redes WAN (Wide Area Network) y LAN (Local Area Network).

El número de usuarios a este servicio cada vez crece mas, lo que ocasiona que se establezcan muchas más conexiones para satisfacer esta demanda. Es por eso que cada vez deben de haber más personas capacitadas para el desarrollo y mantenimiento de estas redes, y El Salvador no es la excepción; pero debido a la falta de solvencia económica de la mayoría de instituciones educativas de nivel superior, no es posible impartir de manera practica esta clase de conocimientos sino a través de centros técnicos especializados que cuentan con los recursos apropiados.

Para poder solucionar o mejorar esta situación se han buscado otras formas de aprender a manejar estos entornos sin la necesidad de equipo verdadero; pero sin alejarse de la realidad.

Una alternativa es el uso de la simulación por computadora, la cual se basa en modelos reales que proporcionan una enseñanza práctica y de bajo costo; y así poder estar al alcance de las instituciones que brindan este tipo de cátedra.

CAPITULO I

PLANTEAMIENTO DEL PROBLEMA

1. OBJETIVOS

1.1. OBJETIVO GENERAL

Desarrollar una herramienta informática (Simulador de Redes) abierta y dinámica que tenga la capacidad de simular los elementos y el funcionamiento de una red LAN para propósitos educativos.

1.2. OBJETIVOS ESPECIFICOS

- Desarrollar una interfase gráfica en la cual el usuario pueda encontrar los diferentes dispositivos (routers, switches, hubs, y hosts) y elementos de red (cable UTP estándar) mediante imágenes y así pueda diseñar una red LAN en una hoja electrónica; permitiendo configurar los dispositivos que lo requieran y simular las funciones de cada uno de estos.
- Desarrollar una herramienta abierta para que sea fácil de comprender por desarrolladores y así añadir nuevas características, funciones y dispositivos según los requerimientos vayan demandando.
- Desarrollar herramientas de prueba de las diferentes capas del modelo OSI como ping y trace, para determinar el buen funcionamiento de la red.
- Mostrar el flujo de información de los procesos de enrutamiento internos que realicen los routers.
- Implementar la selección de la mejor ruta para llegar a un destino.
- Desarrollar el proceso de actualización de las tablas de ruteo.

2. ANTECEDENTES DEL TEMA O PROBLEMA

La Universidad Don Bosco inició sus actividades en mayo de 1985 con las facultades de Ciencias y Humanidades e Ingeniería. La universidad estaba ubicada en el antiguo Instituto Internacional Don Rúa, luego y por un corto tiempo, pasó a la Escuela Domingo Savio; por motivos del terremoto del 10 de octubre de ese año, la universidad tuvo que regresar a su ubicación original.

Tanto el Colegio como la Universidad Don Bosco resultaron afectados por el terremoto de 1986, por lo que los salesianos se vieron en la necesidad de buscar un lugar mas apropiado.

Siguiendo el modelo de Don Bosco, la zona de Soyapango, por reunir los aspectos sociales, juveniles y laborales, fue la más apropiada para albergar al colegio y la universidad, además de situarse en un sector popular y de rápido crecimiento en el país, contribuyendo así al desarrollo humano y social de sus habitantes.

La Universidad ha venido expandiendo sus servicios en los últimos años, ofreciendo nuevas carreras y abriendo nuevas facultades como: la facultad de Economía, la de Estudios Tecnológicos y el nuevo departamento de Diseño Gráfico; con títulos que tienen un valor internacional, como lo es el departamento de Órtesis y Prótesis.

Una de las Fortalezas de la Universidad Don Bosco es la enseñanza practica, que la ha caracterizado sobre todo en la facultad de Ingeniería y Estudios Tecnológicos, donde cuenta con el mayor numero de alumnos.

La universidad cuenta con laboratorios con equipo sofisticado de los cuales se mencionan algunos de ellos a continuación:

En el laboratorio de **Electrónica** esta provisto de equipo digital para especializar a técnicos e ingenieros en el área de comunicaciones, con hardware y software para el área de redes y en el mantenimiento de computadoras de cualquier marca, equipo

electrónico básico, digital y de aplicación industrial para el área de mantenimiento eléctrico y mecánico, etc.

El laboratorio de **Mecánica** de precisión, se cuenta actualmente con tornos, fresadoras, Hornos para tratamientos térmicos, un moderno equipo de metrología, etc., con el fin de cubrir las siguientes áreas:

- Mecánica de Precisión
- Metalurgia
- Metrología
- Soldadura Industrial
- Hidráulica - Neumática

También cuenta con un **Centro de Cómputo** que se utiliza para las prácticas de laboratorio en el área de informática para las diferentes facultades y estudios tecnológicos y para dar capacitaciones a empresas. Dicho Centro de Cómputo cuenta 120 computadoras con programas utilitarios como: Microsoft Office, AutoCad, etc., sistemas operativos como: Windows 98, NT y 2000, Linux, etc. , Lenguajes de programación como: Visual Basic, Visual C++ y Visual FoxPro entre otros, y acceso a Internet, que cubren con los requisitos para todas las materias a las que se sirven, con la excepción de las materias electivas de Redes (Redes LAN y redes WAN) de la carrera de ingeniería en ciencias de la computación; por lo que se debe recurrir a otros laboratorios para realizar las practicas, pero que solo de manera parcial cubren los objetivos de dichas materias.

La academia Cisco en la Universidad Don Bosco nació en el año de 1999 mediante el convenio firmado entre esta universidad y Cisco Networking Academy, con el propósito de educar y certificar a todo aquel que desee adquirir los conocimientos acerca de las redes de computación en todo tipo de organización.

Cisco Systems es el líder mundial en redes para Internet. Las soluciones de Cisco conectan a las personas, las computadoras y las redes, permitiéndoles tener acceso o transmitir información sin importar el tiempo, el lugar o el tipo de equipo.

La certificación de Cisco es obtenida después de cursar cuatro módulos de clase, usando todo un programa pedagógicamente probado e interactivo, realizando practicas en un laboratorio dotado de los últimos en equipo, exámenes rendidos y evaluados por Internet, con la posibilidad de pasantías laborales; todo esto con la oportunidad de postularse en puestos de responsabilidad altamente demandados.

Al término de los módulos, bajo la guía de un instructor para apoyar las tareas de profundización teoría / práctica, en un laboratorio habilitado para uso de los estudiantes durante diferentes horarios, alcanzará los conocimientos necesarios en el diseño, implementación y administración de redes.

Conocimientos que incluyen temas tales como el modelo estándar en redes, direccionamientos físicos y lógicos, configuración de enrutadores, diseño de redes, detección y reparación de fallas, uso y análisis de resultados de software de administración de redes y mucho más.

La Universidad Don Bosco deseando que sus estudiantes obtengan una mejor capacitación en las áreas que están siendo exigidas en el mundo profesional ha pensado desarrollar un Simulador de redes LAN como una alternativa de solución a este problema, en el cual se pueda ir a realizar los laboratorios de las materias impartidas en la Universidad que así lo necesiten; contribuyendo así al desarrollo técnico y económico del país.

3. IMPORTANCIA Y JUSTIFICACIÓN DEL TEMA

El campo profesional en el área de informática cada vez va siendo más exigente, es por eso que cada estudiante universitario debe capacitarse más hoy en día. Uno de los obstáculos a esta capacitación en áreas específicas del conocimiento es que no se realizan prácticas en algunas de estas áreas, y por lo regular se debe a falta de equipo, pues su costo tiende a ser elevado. Una de las áreas que necesita mas desarrollo en un entorno practico es la de Redes de Comunicación. La solución más eficiente es la de comprar un equipo completo para desarrollar un entorno en el cual practicar, pero debido a la realidad de la situación económica de las instituciones educativas de este país, esto no es factible.

La manera más eficaz de llevar acabo practicas de este tipo es desarrollando un simulador de dichos entornos, en el cual se pueda realizar el proceso de configuración de una Red de Área Local (LAN); Dicha propuesta fue hecha por el Decano de Ingeniería de La Universidad Don Bosco.

4. PROYECCIÓN SOCIAL

A través de este proyecto se beneficiará a los estudiantes de la Escuela de Computación de la facultad de Ingeniería en Ciencias de la Computación y del Tecnológico en Ingeniería de la Computación de la Universidad Don Bosco y a otras personas interesadas en aprender el manejo de redes.

En promedio anualmente serán beneficiados 40 estudiantes por ciclo; con recursos limitados para pagar el costo de los equipos y elementos necesarios para poder diseñar y construir una red de laboratorio para cada practica de la materia.

5. DEFINICION DEL TEMA

Hoy en día las redes dentro de las empresas se han convertido en un elemento vital para la transferencia de información a nivel interno o externo, ya sea por medio de la red de redes (Internet) o a través de enlaces remotos a otros puntos de la empresa o institución.

Además el rápido crecimiento de Internet en los últimos ocho años obliga a las pequeñas y grandes instituciones a hacer uso de los servicios que el Internet ofrece, porque de no hacerlo muchas pueden quedar fuera del juego.

Por otro lado las instituciones se ven en la necesidad de enlazar sus computadoras para poder hacer uso de los servidores, bases de datos, sistemas, archivos y demás recursos.

Por ello es necesario contar con personal calificado que pueda operar, configurar y administrar todos los dispositivos y elementos necesarios para construir y dar mantenimiento a una red. Este personal requiere preparación técnica y académica, siendo la universidad la encargada de ofrecer dicha preparación. Sin embargo el alto costo de los equipos y elementos de redes como routers, switches, computadoras, etc. Unido a la falta de recursos económicos que muchas veces enfrentan las universidades dificulta la enseñanza práctica necesaria para producir el personal calificado que las instituciones demandan.

Una solución a dicha problemática es la creación de un Simulador de Redes LAN donde se encuentren todos los dispositivos y elementos de redes necesarios para poder simular una red real, simulando también la transmisión de los datos y así poder ofrecer una preparación adecuada a los estudiantes a un costo accesible.

6. ENFOQUE CON QUE SE ABORDARA EL TEMA

Investigación y Análisis de los conceptos, procesos, funcionamiento y configuración de todos los elementos involucrados en las redes LAN de datos. Con el fin de desarrollar una herramienta gráfica, sencilla al usuario, modular es decir se pueda ampliar y mejorar, que pueda simular todo estos elementos antes descritos y de esta manera permitir a la Universidad Don Bosco ofrecer una educación calificada de redes basándose en los recursos y oportunidades actuales con los que se cuentan.

7. ALCANCES

- Desarrollo e implementación de simulador gráfico de redes LAN para sistemas operativos Windows. Dicho Simulador será desarrollado en Visual Basic 6, y puede ser utilizado para enseñanza, entrenamiento y evaluación de diseños.
- Creación de manual del usuario y programador.
- El simulador contará con dos herramientas de prueba o diagnostico las cuales son: PING en consola de Router y PC y TRACERT solamente en consola de PC.
- Desarrollar una herramienta amigable, educativa y modular adaptada a las condiciones de la institución y a los requerimientos de redes.
- Entrenamiento del uso y administración del simulador al Decano de Ingeniería de la Universidad Don Bosco, instructores y profesores relacionados con materias de redes en el momento de implementar el proyecto.
- El máximo de alumnos que podrán beneficiarse en cada laboratorio donde se utilice el simulador de redes, será según la planificación de la Universidad.
- La red diseñada en el simulador podrá ser guardada para futuras modificaciones.
- Los resultados de las evaluaciones serán visualizados en pantalla y en papel.
- La configuración de cada dispositivo y el diseño en pantalla podrán ser impresos en papel.
- El historial de los procesos internos del router solo mostrara las siguientes acciones: Enviar, Cambiar métrica, Eliminar Ruta, Construcción de mensaje de actualización.
- Se llevara a cabo la simulación de procesos internos de los routers, como son, actualización, propagación de tablas, entre otros; los cuales se irán describiendo en una ventana de Historial que será mostrada cuando el usuario la invoque y se ocultara de la misma forma.
- Se podrán establecer ip secundarias en las interfaces del router.
- Implementación de búsquedas ARP, pero sin comandos de visualización.
- Se podrá eliminar los cables utilizados para la conexión entre dispositivos.

- Mediante comando se muestra las ip secundarias configuradas en las interfaces de router.
- Se podrá eliminar las Vlan creadas por el usuario. (Los puertos asociados a la Vlan eliminada, vuelen a asociarse a la Vlan1)

8. LIMITACIONES

- La herramienta se diseñara basándose en el equipo de redes con el que la Universidad Don Bosco y la Academia de Redes Cisco cuentan.
- La herramienta se desarrollará para la Universidad Don Bosco. La universidad tendrá toda la autoridad para distribuirla o no en otras instituciones educativas.
- Al Diseñar la red en el simulador solo se podrá utilizar cable UTP estándar y seriales para conexiones entre routers.
- La forma de evaluar el aprendizaje del estudiante consistirá en la realización de 8 laboratorios prediseñados en el simulador, los cuales serán calificados automáticamente y se mostrara su nota respectiva.
- La conexión WAN-Básica entre routers estará basada en el protocolo HDLC propietario de CISCO y no se utilizara PPP ni Frame Relay.
- Los protocolos de ruteo a utilizar serán IGRP y RIP versión 1.
- No se realizarán pruebas con la herramienta Telnet, ni con otra que no sean las dos antes mencionadas.
- Unión de dos switches, solo cuando sea implementación de trucking en VLAN.
- Solo se pueden crear y asignar 4 vlans adicionales a la Vlan por defecto (Vlan1), en cada switch y el rango de valores es de 1-1001.
- El sistema autónomo del protocolo IGRP acepta valores entre 1 a 65535.
- No se permite poner hubs en cascada, tampoco hubs cableados a un switch, ni viceversa.
- Los comandos ping y tracert no aceptan ningún parámetro
- El comando de ayuda (?) muestra la lista de comandos de los siguientes modos: usuario, privilegiado, configuración global, configuración específica de interfaz, y de router.
- Al poner en Shutdown una interfaz no funciona ninguna ip asociada a esta.
- Al eliminar un ip primaria de una interfaz se eliminan las ip secundarias.
- La salida del comando show interfaces en el switch es netamente descriptiva, solo para que el usuario conozca la salidas de este.

- Ninguna lista de control de acceso puede ser configurada en los routers.
- Ningún comando de switch o router permiten parámetros opcionales, excepto el comando para configuración de ip secundarias.
- El comando configure-register del router, acepta valores hexadecimales desde 0 hasta FFFF, y el registro solo es para que el usuario lo conozca.
- No se pueden conectar host, hub o switch a interfaces seriales del router.
- No se puede cablear un host a otro host directamente.
- Entre routers, no se pueden conectar una interfaz ethernet a una serial de otro router.
- El comando copy running-config startup-config solo es para familiarizarse con este.
- Los routers no envían mensajes de actualización por medio de sus interfaces ethernet.
- El comando Clock rate no esta implementado.
- Existe un límite de caracteres en las consolas con un promedio de dos millones.
- El switch no acepta configuraciones IP.
- Esta implementado en el switch solamente la Interfaz de Línea de Comandos (CLI).
- Aunque los dispositivos del simulador están basados en los modelos existentes en el laboratorio de CÍSCO. No todos los comandos soportados por estos dispositivos están implementados en el simulador. En el manual de usuario y en la ayuda del simulador se detallan los únicos comandos que están implementados.
- No esta implementada la administración remota de Router.
- Las contraseñas de seguridad en modo global implementadas son: Secret y password para el router y solamente password para el switch. Ningún otro tipo de contraseña ha sido implementada.
- No esta implementado la redistribución de rutas.
- Sistemas operativos en los que puede instalarse el simulador:
 - Windows (98/NT/2000/XP/2003)
- Limitación de dispositivos por diseño

Máximo de Routers	5 (Cuatro 2501 y Un 2514)
Máximo de Switch	6
Máximo de Hub	6
Máximo de LAN	6
Máximo de Hosts por LAN	4

Nota: El numero máximo de dispositivos (Switches o Hubs) por diseño es 6, esto significa que Puede haber 6 switches, o 6 hubs o la combinación entre ellos que hagan un total de 6.

CAPITULO II

ESTUDIO DE FACTIBILIDAD

1. FACTIBILIDAD TÉCNICA

Uso de la simulación por computadora en el aprendizaje.

Las simulaciones son la única estrategia de consulta debido a que ellas son representaciones de la realidad. Son como un vehículo para la adquisición de conocimiento y habilidades en un entorno de aprendizaje exploratorio, las simulaciones permiten la interacción del estudiante mediante el ingreso de respuestas, direcciones o decisiones y resolución de problemas.

Durante el proceso, el aprendiz esta activamente involucrado en la construcción y reconstrucción de su base del conocimiento. El poder es puesto en manos del estudiante, para proveerle con la habilidad de probar y comunicar sus propias ideas de cómo funcionan las cosas. El aprendizaje se adquiere "haciendo".

Las simulaciones por computadora no son como un programa didáctico en general ya que no pretenden reemplazar al maestro, sino que están diseñadas para proveer nuevas oportunidades de aprendizaje. Son ideales cuando es imposible recrear situaciones inaceptables en la realidad y son mejor utilizadas cuando son administradas por los estudiantes quienes han dominado un conjunto de conocimientos y están listos para aplicar el conocimiento adquirido. Debido a que el aprendizaje a través de la exploración se hace de forma cognitiva por parte del aprendiz, comportamientos de aprendizaje ineficientes e inefectivos pueden llevar a alumnos inhábiles, a hacer cambios al azar en vez de usar los valores apropiados en parámetros y variables.

Para que las simulaciones sean efectivas, ellas deben ser hechas en la presencia de un instructor o un entorno de aprendizaje por computadora que monitoree el rendimiento del estudiante y proporcione una guía en el uso de las simulaciones para desafiar el conocimiento existente.

1.1 Aspectos y Características.

Simulaciones por computadoras efectivas e interactivas proveen un conjunto verosímil de eventos y circunstancias, y opciones para respuestas continuas por parte del estudiante y una guía para el perfeccionamiento del escenario.

Los aspectos de la simulación basada en sistemas de entrenamiento son:

Los aprendices tienen varias experiencias del mundo objetivo en entornos imaginarios y gradualmente establecen inductivamente modelos mentales. El conjunto de circunstancias que definen la simulación son presentadas al estudiante en un escenario inicial. Este ofrece a los estudiantes suficiente información para fundamentar una respuesta bien formada y ellos deben elegir un curso de acción apropiado. El estudiante debería tener una clara expectativa a consecuencias anticipadas a las respuestas que dio.

Un programa de simulación no aburre al aprendiz por que "volar" un avión o hacer un diagnostico a un paciente simulado es más interesante que leer acerca de cómo hacerlo en un libro de texto.

Los aprendices comprenden los conceptos más fácilmente a través de la visualización sin necesidad de la comprensión de un complejo lenguaje natural.

Las condiciones deseadas de un entorno simulado son:

- Respuestas rápidas en acciones u operaciones del aprendiz.
- Modelos de simulación están bien integrados al curso.
- Cada paso de la situación es entendido fácilmente.

Bessemer y Kolosh (1992) examinaron el valor de usar los sistemas de simulación basados en computadoras para el entrenamiento de estudiantes. Ellos creían que la efectividad del uso de tal sistema instruccional yacía en proveer a los estudiantes con oportunidades realistas y seguras.

Una vez que el conocimiento / habilidad era adquirido, estos podían ser transferidos cuando eran aplicados a situaciones de la vida real.

Fuente: [http: //www.ascilite.org.au/ajet/ajet9/gatto.html](http://www.ascilite.org.au/ajet/ajet9/gatto.html) (esta en ingles)

1.2 Herramientas de aprendizaje y simulación usados en el mercado de servicios educativos nacional e internacional.

1.2.1 OMNeT++:

- Es in simulador de eventos discretos desarrollado por Andras Vargás.
- Esta diseñado principalmente para simular una red de computadoras, multi-procesos y otros sistemas distribuidos, pero puede ser útil para el modelado de otros sistemas también.
- Incluye una serie de presentaciones que introducen al OMNeT++. Desde Holger Karl, Technical University of Berlin, Telecommunication Networks Group.
- Su popularidad esta incrementándose en un largo rango de comunidades científicas y compite muy bien con herramientas ya establecidas. Esta siendo usado por un gran numero de universidades para la investigación en las redes de comunicación. Algunas de estas universidades también usan OMNet++ en cursos de simulación y/o redes de comunicación.
- A finales del 2000, se habían registrado descargas desde más de 40 universidades, entre las que se pueden mencionar:

Aalborg University, Aalborg, Denmark; Charles University, Faculty of Mathematics and Physics, Prague; Departement of Electrical and Electronic Engineering University of Pretoria - South Africa; Dept. EE, National Cheng Kung University, Taiwan; Dept. of

Electrical and Computer Eng., Elec.Info Insti of Wuhan Univ, P.R.China;
Electromechanical Faculty of Craiova - Romania; Department of Electrical Engineering
and Process Control in Mining; University of Virginia.

1.2.2 Harvard TCP/IP network simulator 1.0: basado en la metodología de simulación propuesta por S. Y. Wang y H. T. Kung en INFOCOM'99, usa el código BSD (Berkeley Software Distribution) existente (incluyendo la pila TCP/IP, programas de aplicación, utilidades, etc.) para proveer una simulación para una red TCP/IP extensible y de alta calidad. Esta versión es usada en más de 150 universidades, laboratorios de investigación industrial, e ISPs.

Fuente: <http://www.eecs.harvard.edu/networking/>

1.2.3 OPNET Modeler: Es un entorno de desarrollo para tecnologías de red, que permite diseñar y estudiar redes de comunicaciones, dispositivos, protocolos y aplicaciones con una flexibilidad incomparable. El Modeler es usado por las organizaciones de tecnología más prestigiosas para acelerar sus procesos R&D (Research And Development).

Las universidades que utilizan OPNET Modeler están:

1. Carnegie Mellon
2. MIT
3. Stanford
4. Univ. of California (Berkeley)
5. Univ. of Illinois (Urbana-Champaign)

fuelle: <http://www.opnet.com/products/modeler/home.html>

1.2.4 Otros simuladores.

1.2.4.1 Simulación de Osciloscopio.

Institución:

Laboratorio de Informática Educativa – Departamento de Computación,
Facultad de Ingeniería, Universidad de Buenos Aires.

- Este simulador tiene como objetivo presentar una introducción general al uso de un osciloscopio analógico y un generador de funciones básico. Debido a que estos dispositivos tienen un muy elevado costo, y quizás no se dispone de la cantidad suficiente para un alumnado numeroso, es una buena práctica realizar una introducción a su uso mediante la distribución de este programa, para aprovechar mejor las pocas horas de utilización de dichos aparatos que puede disponer cada uno de los alumnos.
- Este programa esta siendo utilizado como material de apoyo en escuelas de enseñanza técnica, donde los equipamientos son escasos y también en laboratorios previos al uso de equipos reales por parte de los alumnos.
- Esto se ha logrado debido al mínimo requerimiento de soporte informático para su funcionamiento. Hay que agregar a esto, que es fácilmente portable, de sencillo de uso y con una interfase amigable.
- Como en todo software educativo, se ha realizado una evaluación en el contexto educativo. De ella se desprende que los alumnos se sienten cómodos con el uso de esta herramienta, mientras los docentes de las materias conexas han informado que la simulación reduce los tiempos de aprendizaje del uso del osciloscopio.

Fuente: <http://enlaces.c5.cl/tise99/memoriatise99/html/software/simulacion/>

1.2.4.2 HYSYS

Institución:

Facultad de Química

Universidad Autónoma de México.

- El simulador cuenta con un amplio conjunto de módulos para simular las operaciones unitarias, además de una amplia base de datos que permite el cálculo de propiedades termodinámicas y cinéticas de compuestos reales e hipotéticos.
- Para el inicio de un caso en estudio, se seleccionan los paquetes de propiedades a utilizar, así como los componentes de la base de datos o se crean los hipotéticos. El diagrama de flujo de información se construye en la pantalla, instalando las figuras que representan los módulos de los equipos u operaciones necesarias, y las corrientes de entrada y salida de cada uno.
- A partir de conocimientos básicos de simulación es posible desarrollar proyectos para los que su solución requiera del uso de un simulador. La ubicación estará en dependencia de las operaciones unitarias que el proceso a simular incluya.
- Con el ejemplo que se presenta, el alumno, además de ejecutar las habilidades asociadas directamente con la simulación de procesos, refuerza la necesidad del cuidado del medio ambiente y el papel que el ingeniero químico debe jugar para su preservación, reduciendo los contaminantes atmosféricos. Por otra parte, profundiza en la utilización de los ciclos combinados para el incremento de la eficiencia de los procesos de generación de energía en centrales térmicas. La búsqueda independiente de información, el análisis de los posibles esquemas de simulación y la determinación de la influencia de las variables de operación sobre una considerada como objetivo, son las etapas que caracterizan la realización del proyecto y permiten el desarrollo de valores como la independencia, creatividad y la capacidad de análisis y síntesis.

Fuente: <http://www.fquim.unam.mx/eq/124/124-zuma.pdf>

1.2.4.3 SPICE

Institución:

Laboratorio de Electrónica,
Universidad Don Bosco.

- El nombre de SPICE es el acrónimo de SIMULATION PROGRAM WITH INTEGRATED CIRCUITS EMPHASIS, se desarrolló en el Laboratorio de Investigación en Electrónica, del Departamento de Energía Eléctrica y Ciencias de la Computación de la Universidad de Berkeley, California y fué puesto a disposición del dominio público en Julio de 1975, este programa es útil para resolver circuitos y sistemas simples o complejos, el SPICE es un simulador que ejecuta el análisis de circuitos de CD, de CA y de Fourier con afectos transitorios, así como también es posible hacer el análisis de respuestas en diagramas de bloques en el dominio de Laplace, gracias a esta ultima capacidad es posible la simulación de sistemas de tipo no eléctricos ni electrónicos, sino que es posible modelar cualquier sistema que sea posible trasladar al dominio de Laplace.
- Con el uso de esta herramienta se incrementa la percepción del funcionamiento de circuitos y su dependencia de los parámetros, así como su relación analítica y la interpretación gráfica alternativa acelerando el diseño y optimización de los mismos.

1.2.4.4 SIMULADORES EN LA UNIVERSIDAD FRANCISCO GAVIDIA.

<http://www.wisis.ufg.edu.sv/labvirtual/default.htm>

En la página Web de la universidad Francisco Gavidia, se pone a disposición del alumno, en la sección Laboratorios Virtuales ElectroLab, una serie de applets para la simulación de:

- Capacitares MOS
- Circuitos digitales, etc., para el área de ingeniería;
- Applet para el cálculo de vectores y sus diferentes operaciones y graficados, en el área de matemáticas.

Estos complementan la enseñanza tradicional haciéndola más activa y divertida, contando con más 124 *applets* insertados en sus páginas *webs*. Simuladores de sistemas físicos, prácticas de laboratorio, experiencias de gran relevancia histórica, problemas - juego, y otros temas complementarios.

OTROS VÍNCULOS A MÁS INFORMACIÓN:

<http://mip.ee.nus.edu.sg/links.html>

<http://www.topology.org/soft/sim.html>

Existe una gran cantidad de entornos de desarrollo de los cuales un programador de aplicaciones puede elegir, ya sea en el desarrollo de aplicaciones para Windows, Linux y otros sistemas operativos.

Estos entornos utilizan lenguajes para realizar tareas que el programador especifique. Para el lenguaje Basic existe Visual Basic, Realizer, GFA Basic; para C están Watcom C, Symantec C, Visual C++, Borland C, Zortech C. Para Pascal hay poco soporte. Por muchos años existió Microsoft Pascal, Turbo Pascal, FPK Pascal, Borland Pascal y por ultimo Delphi.

Debido a que el software de simulación a crear presentará elementos como computadoras, routers, hubs, etc., será necesario el uso de una herramienta visual que permita representarlos gráficamente y que además permita la creación de objetos personalizados para definir su comportamiento y sus características. Sin dejar a lado, un ambiente grafico donde pueda correr dicho simulador.

Por lo que la elección del software a utilizar no solo depende del lenguaje sino que también, que ofrezca un entorno en el que se puedan crear ventanas en las que se puedan desplegar los elementos representativos.

Este estudio se centra en 3 preguntas importantes para la elección de ese entorno visual de desarrollo a utilizar para la construcción del simulador.

Los aspectos son los siguientes:

¿Qué se esta tratando de hacer?

¿Qué clase de características se desean implementar?

Decidir *cuánto* se esta dispuesto a invertir en: ¿dinero? , ¿Entrenamiento? Y en ¿tiempo de desarrollo?

1.3 Desarrollo rápido de aplicaciones (Rapid Application Development).

El concepto de Desarrollo rápido de aplicaciones (RAD por sus siglas en ingles) hace referencia a que las aplicaciones pueden ser desarrolladas de forma más rápida y de alta calidad mediante:

La recolección de requerimientos de información usando talleres de trabajo o grupos.

La creación de prototipos para evaluación de los usuarios.

La reutilización de componentes de software.

Un diseño a partir del cual se pueden crear las nuevas mejoras para la siguiente versión de la aplicación.

Factores importantes para el desarrollo rápido:

Involucra intensivamente al usuario final en el diseño del sistema.

Uso de prototipos, que ayuda a los usuarios a visualizar y a hacer ajustes al sistema.

Involucra al usuario en la etapa de construcción, permitiendo que los detalles sean ajustados en caso de necesidad.

Desarrollo de una estructura de la tarea que permite realizar las actividades del proyecto en paralelo.

A continuación se expondrán dos de las herramientas RAD mas usadas por los programadores en el mundo.

1.4 Herramientas de programación visual.

Todas las herramientas de programación visual comparten una misma apariencia básica: un formulario, una caja de herramientas de componentes, una ventana de propiedades y una ventana de código.

1.4.1 VISUAL BASIC

Según el sitio de noticias informáticas Cnet.com, en el artículo "Microsoft to debut developer tools"¹ del 13 de Febrero del 2002, señala que de acuerdo a un reciente estudio hecho por investigadores de la firma IDC, dedicada al análisis de la industria y datos de mercado, los lenguajes mas utilizados son: C o C++, mas de 3 millones de desarrolladores de software lo usan como lenguaje primario, seguido de Visual Basic con 2.3 millones de usuarios.

Visual Basic es un entorno de desarrollo de aplicaciones para Windows con muchas herramientas útiles, sin mencionar el total acceso al entorno de Windows, por lo tanto un programa escrito en VB puede hacer casi todo lo que hace Windows. Además ofrece un conjunto de características orientadas al desarrollo de aplicaciones personalizadas de una forma fácil y rápida.

El lenguaje utilizado en Visual Basic es directo, con la revisión de sintaxis usando colores y con "compilador real" (sin usar el P-code) por lo que la ejecución de un programa es bastante razonable. Además de mantener el lenguaje BASIC fácil-de-aprender al que se le ha añadido la funcionalidad de Windows.

Un programa generado por Visual Basic está formado por una parte de código puro, y otras partes asociadas a los objetos que forman la interfase gráfica. Es por tanto un término medio entre la programación tradicional, formada por una sucesión lineal de código estructurado, y la programación orientada a objetos.

¹ Referencia: <http://news.com.com/2100-1001-836235.html>

La mayoría de las características nuevas de Visual Basic están orientadas más que todo al acceso a datos y de hacer a Visual Basic una opción viable como ambiente de desarrollo de aplicaciones integradas complejas. Muchas de las nuevas características son respuesta a las quejas comunes de desarrolladores propuestas en grupos de noticias y revistas.

Una de las razones de la popularidad de Visual Basic es el modelo de componente del que es pionero en la creación de pequeñas y medianas aplicaciones, ofreciendo mejoras en la productividad y la facilidad de uso. Además, Visual Basic maneja de mejor manera el ambiente grafico.

1.4.1.1 Características.

Programa con el más popular lenguaje de programación para crear rápidamente aplicaciones y componentes de alto rendimiento en un entorno RAD.

Construir aplicaciones y componentes de código nativo que usa la misma tecnología de compilador de clase mundial como en Visual C++. Las aplicaciones pueden ser optimizadas en velocidad y tamaño, y en muchas otras formas, mejorar la ejecución aun más.

Incrementa la escritura de programas y la productividad con la tecnología IntelliSense. Sintaxis al instante y terminación de sentencias usando características como Complete Word (completar palabra), Quick Info (Información rápida) y DataTips.

Creación fácil de un amplio rango de componentes re-usables que encapsulan propiedades o datos legales, incluyendo controles ActiveX.

Un entorno RAD muy intuitivo que promueve la productividad y ayuda a los desarrolladores a crear y desarrollar rápidamente aplicaciones cliente/servidor.

Creación de programas para Internet usando herramientas y técnicas de programación de Visual Basic, que pueden ser accedidas desde cualquier navegador en cualquier plataforma.

Incorpora el entorno de programación de datos mejorado y proveedores OLE DB. ADO (ActiveX Data Objects) para acceder a datos a través de la LAN o Internet.

Uso del Microsoft Data Engine como parte de las aplicaciones y obtener una compatibilidad completa con grandes bases de datos en SQL Server.

Construcción de aplicaciones con auto-reparación y bajos costos de distribución con el nuevo Visual Studio Installer.

1.4.2 DELPHI

Delphi es uno de los mejores entorno para desarrollo rápido de aplicaciones (RAD), con un potentísimo lenguaje el Object Pascal, un compilador rapidísimo que nos permite crear ejecutables con una velocidad cercana al C++, un lenguaje orientado a objetos, y con múltiples posibilidades: bases de datos, multimedia, web, etc.

Delphi, desciende por decirlo de alguna manera, del mítico lenguaje desarrollado por Niklaus Wirth en 1971, para facilitar la enseñanza de la programación estructurada, y que le puso el nombre del tan celebre filósofo y matemático francés Pascal.

El lenguaje Pascal es mas estricto que el de Visual Basic y requiere un poquito mas de prudencia al programar. Además, Delphi compila archivos .exe; su compilador es increíblemente rápido que al principio es difícil de creer que la compilación se realice tan rápida.

En aquella época, las fases de edición de código, compilación y enlazado, iban por separado, hasta que Philippe Khan, el fundador de Borland, actualmente Inprise, creo el Turbo Pascal, el primer entorno de programación DOS, con el editor, compilador y linker integrados, siguiéndole los famosos Turbo Basic y Turbo C/C++.

Luego vieron las versiones de Turbo Pascal para Windows, pero en 1995 ante la aparición del Windows a 32 bits, Borland decide dar un giro al producto, lo potencia y

mejora incorporando las nuevas tecnologías Windows, lo denomina Delphi, empieza a comerle terreno al Visual Basic. Mientras tanto, por la tardanza y algún que otro problema con los primeros xBase Windows muchos programadores Clipper y dBase, deciden pasarse a Delphi.

1.4.2.1 Características

Herramienta de desarrollo más rápida y productiva para la web y desarrollo de Bases de Datos.

Velocidad de ejecución, compilación y enlace cercano al C++, y por lo tanto mucho mejores que otros lenguajes existentes.

Verdadera Programación Orientada a Objetos, permite encapsulamiento, herencia y polimorfismo.

Componente integrados dentro del lenguaje, lo que reduce considerablemente la utilización de librerías y controles externos, por lo tanto menos problemas para los programadores y el usuario.

Tratamiento de errores mediante excepciones, lo que impide el típico error de programa que nos echa fuera.

Soporte avanzado de Bases de Datos mediante BDE (Borland Database Engine), ADO (ActiveX Database Objects), tecnología de Microsoft de acceso a Bases de Datos e Internet que incorporará el próximo Windows 2000, y finalmente InterBase Express, acceso nativo a InterBase, para desarrollo Cliente/Servidor off-line.

Modelo de datos y relaciones de forma visual.

Asistentes y componentes para Internet/Intranet.

Componentes compatibles con Microsoft Office.

Fácil integración de informes y gráficos de gestión.

Facilidad en la distribución de aplicaciones con el InstallShield Express.

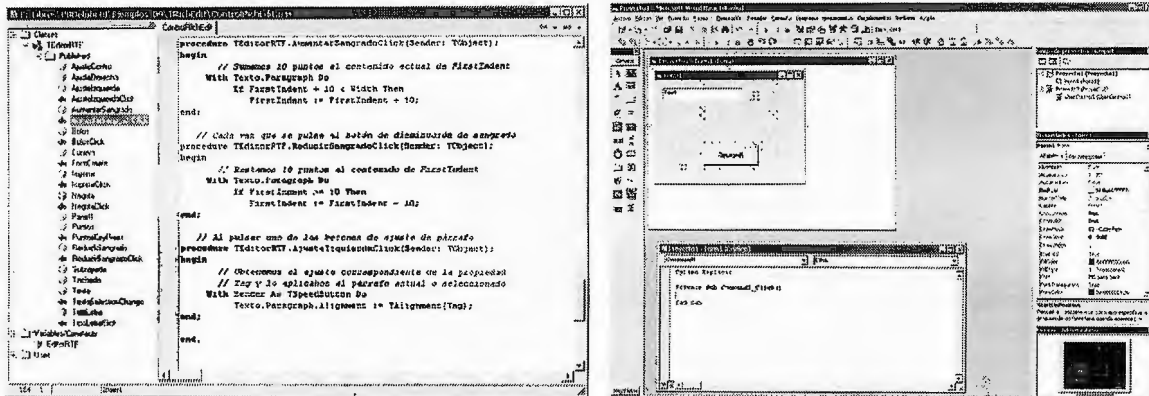
¿Delphi o Visual Basic?

Interfaz del entorno

Tanto Visual Basic como Borland Delphi cuentan con entornos de desarrollo basados en una interfaz de última generación, con múltiples ventanas acoplables entre sí, barras de botones configurables, conjuntos de componentes agrupados en páginas, menús emergentes sensibles al contexto y un impresionante número de opciones correctamente dispuestas en los correspondientes menús desplegables.

Trabajar con una herramienta de desarrollo RAD significa tener lo más a mano posible todos los objetos predefinidos, componentes o controles, que pueden utilizarse para el diseño de las aplicaciones. En el caso de Delphi existe la llamada Paleta de componentes, mientras que Visual Basic cuenta con su Caja de herramientas. La primera es una ventana multipágina en la que se encuentran los más de 170 componentes con que cuenta inicialmente Delphi, debidamente agrupados según su funcionalidad o categoría. La segunda es una ventana en la que tan sólo encontramos una veintena de controles, los más habituales, siendo preciso acceder a una ventana auxiliar cada vez que se quiere un nuevo control o grupo de controles. La Paleta de componentes de Delphi es totalmente configurable, permitiendo crear nuevas páginas, alterar el nombre y disposición de las existentes, mover componentes de unas páginas a otras, etc. En la Caja de herramientas de Visual Basic es posible añadir nuevas páginas, aunque de una forma mucho más rudimentaria. Además, cuando se añaden nuevos componentes éstos no van a la página que tenemos abierta sino a la página general, siendo preciso arrastrarlos después a la página que nos interese uno a uno.

Tanto el formulario como los diversos componentes insertables en él, son objetos genéricos que es preciso personalizar para adaptarlos a nuestras necesidades. Esto se consigue básicamente de dos formas: modificando propiedades y asociando código a los eventos. La ventana Propiedades de Visual Basic y el Inspector de objetos de



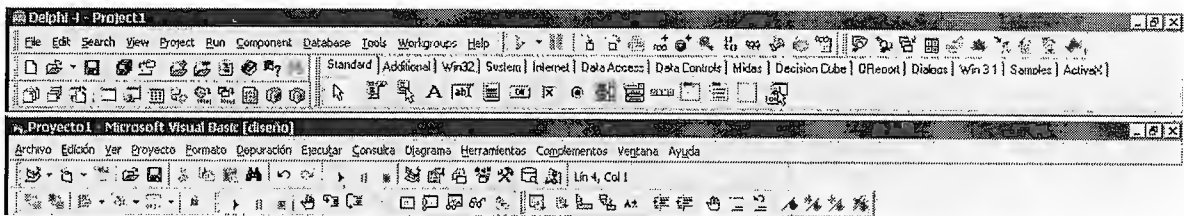
Delphi son ventanas muy similares, que facilitan la edición de los valores correspondientes a las propiedades con que cuenta el objeto seleccionado actualmente. Ambas cuenta con una lista desplegable que sirve tanto para saber qué objeto hay seleccionado como para alterar dicha selección. Delphi muestra la lista de propiedades en orden alfabético. En Visual Basic podemos elegir entre dicho orden o bien una lista por categorías. Esta última herramienta, además, puede mostrar en la parte inferior de la ventana una pequeña ayuda acerca de la finalidad de la propiedad elegida en cada momento.

Los formularios, así como los módulos de código, clases y otros recursos, son elementos que se combinan para crear proyectos. Para administrarlos es preciso disponer de la herramienta adecuada: el gestor de proyectos. Tanto Delphi como Visual Basic son capaces de manejar grupos de proyectos, de tal forma que es posible crear un control y una aplicación que lo utiliza, o una librería de enlace dinámico y un programa que usa sus funciones, por poner dos ejemplos habituales.

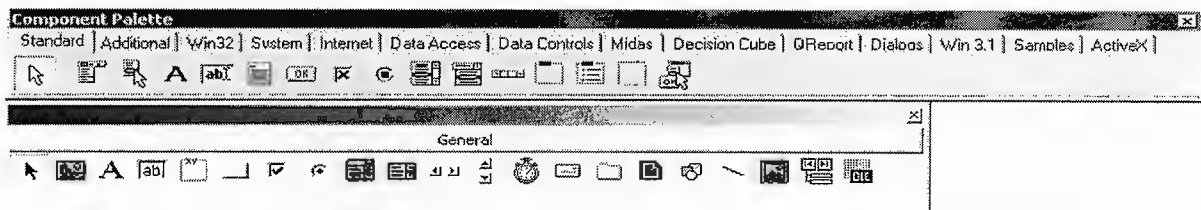
En general, las interfaces de Delphi y Visual Basic son eficientes, flexibles y cómodas. La mayoría de las ventanas, como las de edición de propiedades o gestión de proyectos, pueden ser marcadas como acoplables, lo cual facilita su disposición en el entorno. En principio Visual Basic usa un modelo MDI, de tal forma que las ventanas correspondientes a diseñadores y edición de código están contenidas en una ventana principal. Delphi, por el contrario, usa el modelo SDI que le es habitual desde su

Nacimiento. Los usuarios de Visual Basic podrán elegir el modelo que quieren, ya que es posible cambiar mediante una opción de MDI a SDI. Delphi, sin embargo, no dispone de una opción similar.

Las interfaces del entorno de trabajo de Visual Basic y Borland Delphi son, en general, cómodas y flexibles. Las distintas ventanas pueden acoplarse entre sí o quedar como ventanas flotantes, al igual que las paletas de botones.

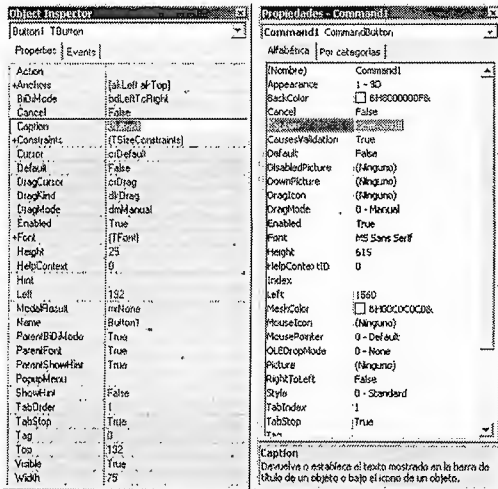


Las ventanas principales de Borland Delphi y Visual Basic alojan el menú de opciones y las diversas barras de botones. Delphi, además, también coloca inicialmente la Paleta de componentes en esta ventana.

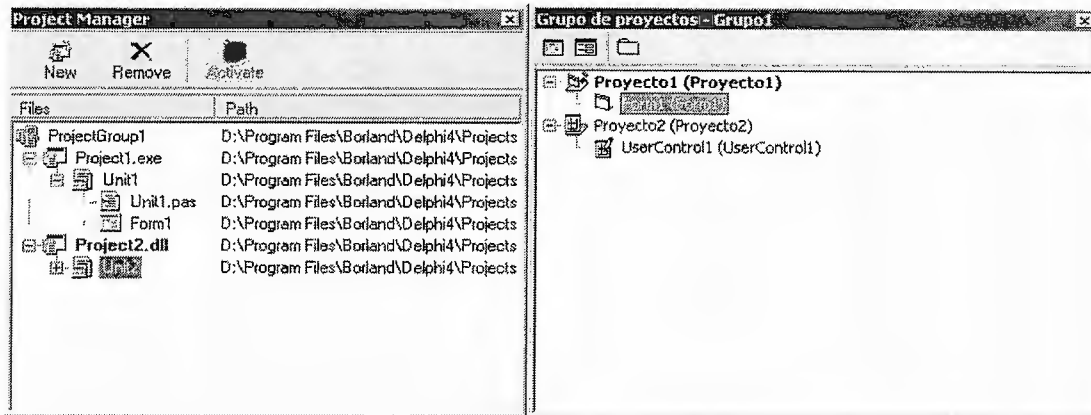


En la Paleta de componentes de Delphi (arriba) se encuentran todos los objetos disponibles, inicialmente más de 170, agrupados en páginas según su funcionalidad. La Caja de herramientas de Visual Basic en principio sólo contiene los controles más habituales.

El Inspector de objetos de Delphi (a la izquierda) y la ventana Propiedades de Visual Basic son muy similares, facilitando la edición de los valores que contienen actualmente las propiedades del objeto seleccionado.



Tanto Visual Basic (abajo) como Delphi cuentan con gestores de proyectos similares, que permiten trabajar con grupos de proyectos y manipular los elementos que los componen.



Componentes

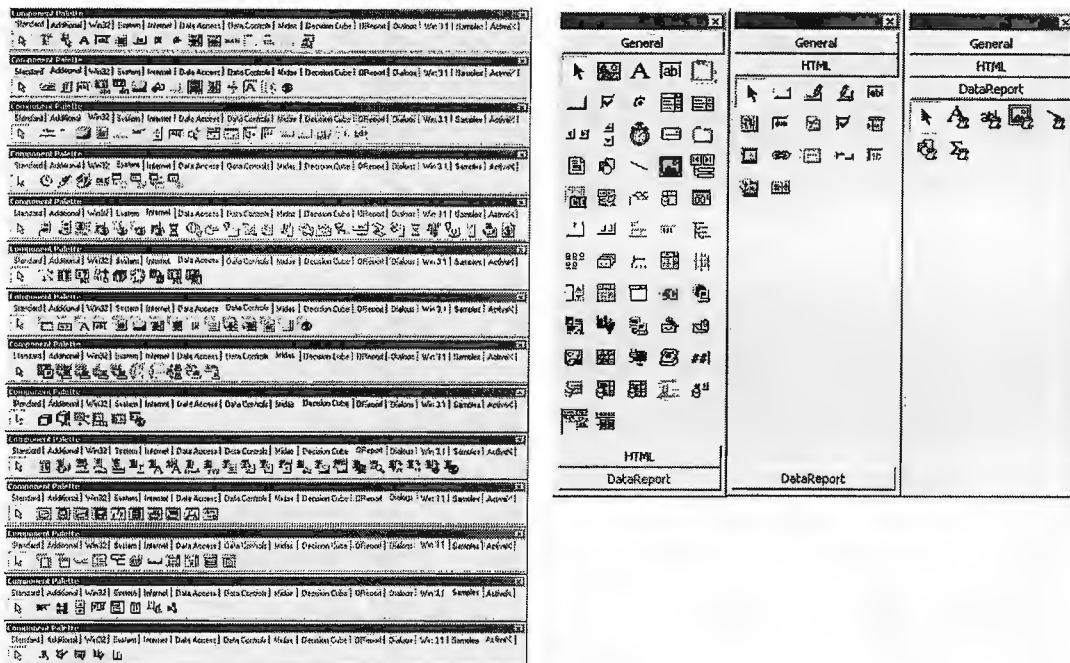
Uno de los apartados más importantes en una herramienta de desarrollo RAD es, sin duda, el número y la calidad de los componentes que incorpora. De ello depende que el usuario tenga más o menos posibilidades y que, en ocasiones, se vea en la necesidad o no de buscar componentes de terceros que satisfagan sus necesidades. También es importante conocer el modelo o modelos de componentes para los cuales está preparada dicha herramienta, ya que ello determinará la cantidad de componentes que podremos encontrar en el mercado.

La primera diferencia entre Visual Basic y Delphi la encontramos, precisamente, en los modelos de componentes que utilizan. Al ser un producto de Microsoft, Visual

Basic como es lógico usa controles ActiveX, el modelo *oficial* de la plataforma Windows. Delphi, por su parte, cuenta con un modelo propio llamado VCL pero, al tiempo, también contempla el uso de controles ActiveX.

La mayor especialización que tienen los componentes de Delphi contrasta con la generalidad de los controles de Visual Basic y también influye en la diferencia de número, con un saldo a favor de Delphi de prácticamente una centena de componentes.

Lógicamente, además de evaluar el número de componentes también sería preciso analizar su funcionalidad, reflejada en las propiedades, métodos y eventos con que



cuenta. Si bien los componentes comunes entre ambas herramientas funcionan de forma muy similar, habría que destacar la facilidad que ofrece Delphi a la hora de crear interfaces de usuario consistentes y flexibles. Para ello existen propiedades que permiten anclar los componentes de forma relativa a los márgenes del formulario,

Limitar sus dimensiones ya sea por exceso o por defecto, permitir que unos componentes se acoplen a otros o que generen acciones comunes, todo ello sin necesidad de escribir código alguno.

El número de componentes con que cuenta Borland Delphi es impresionante si lo comparamos con el existente en Visual Basic. También hay que valorar la forma en que se disponen los componentes en Delphi, agrupados según su finalidad, en lugar de ponerlos prácticamente todos en una sola página como hace Visual Basic. Algunos componentes de Visual Basic tan sólo están disponibles si tienen sentido en el módulo que hay abierto. Los componentes Delphi pueden utilizarse en cualquier formulario, por lo que siempre están activos.

Edición de código

A pesar de la potencia que supone la creación de una interfaz de usuario y parte de la funcionalidad de la aplicación simplemente disponiendo componentes en una ventana y modificando propiedades, está claro que un programa tendrá una utilidad muy limitada si no se gestionan adecuadamente los eventos generados por esos componentes. Dicha gestión se basa en la escritura de código, estructurado en métodos que se ejecutan al producirse el evento correspondiente.

Los editores de código de Visual Basic y Delphi se ajustan al funcionamiento asumido como estándar para este tipo de tareas en Windows. Ambos diferencian el texto sintácticamente usando diferentes colores y estilo en el texto, elementos que pueden ser configurados prácticamente en su totalidad tanto en Delphi como en Visual Basic. Es posible realizar operaciones de marcar y editar texto con el ratón, con la técnica de arrastrar y soltar. La indentación es automática, aunque puede desactivarse, y pueden especificarse los puntos de parada del tabulador.

El editor de Visual Basic sólo puede trabajar con un archivo al tiempo, lo que significa que tendremos tantas ventanas abiertas como módulos estemos editando en cada

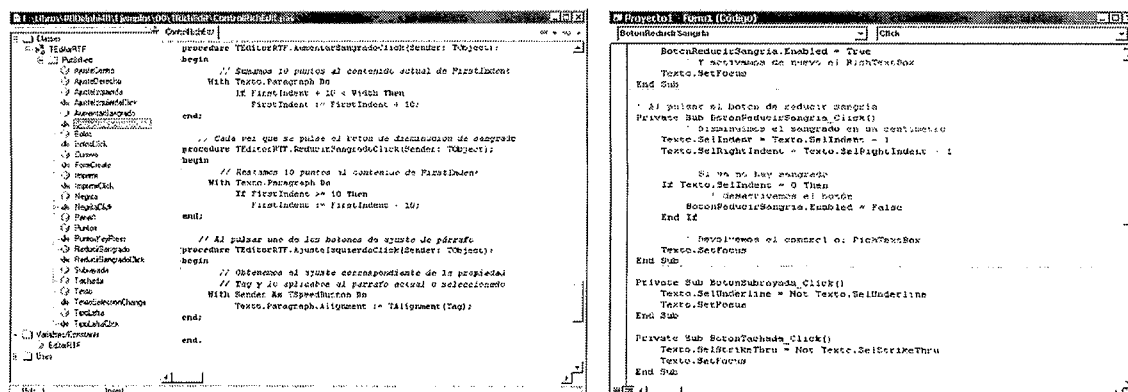
momento. El editor de Delphi, por el contrario, es de tipo multiarchivo, abriendo una nueva página por cada módulo que se esté editando. No obstante, también es posible abrir varias ventanas de edición, por lo que podemos elegir la forma de más trabajo que más se ajuste a nuestras preferencias.

La navegación por el código se ve asistida mediante dos mecanismos diferentes pero igualmente válidos. El editor de Visual Basic cuenta con dos listas desplegables en la parte superior, una en la que es posible seleccionar un objeto, como el formulario o un control, y otra que se utiliza para seleccionar el evento. Delphi cuenta con una ventana, inicialmente adosada al editor de código, llamada Explorador de código. Esa ventana muestra una lista jerárquica de todos los elementos existente en el módulo actual, desde clases, métodos y propiedades hasta variables y constantes. Esta lista, además, puede también ser usada para añadir nuevas declaraciones, módulos y otros elementos, basta con abrir el menú emergente y seleccionar la opción apropiada.

Forma parte de las funciones de una herramienta de desarrollo de este tipo la generación automática de código, básicamente el necesario para abrir los métodos que interesan al programador. En Visual Basic y Delphi basta con hacer doble clic sobre un componente para crear el método correspondiente al evento por defecto. Si estamos interesados en otros eventos, Visual Basic usa la lista existente en el propio editor de código para facilitar la selección. En Delphi, por el contrario, es el propio Inspector de objetos el que cuenta con una página en la que se muestran todos los eventos existentes así como los métodos de aquellos que ya estén codificados. En caso de que un mismo método de evento vaya a utilizarse para múltiples componentes, indistintamente de cuales se trate, basta con desplegar la lista adjunta a cada evento y seleccionar el método deseado, sin más. Visual Basic, sin embargo, no cuenta con esta posibilidad o una equivalente.

A la hora de comenzar a escribir código tanto Visual Basic como Delphi ponen todo su empeño en ayudarnos, abriendo listas de propiedades pertenecientes al objeto cuyo

nombre acabamos de escribir, listas de valores que es posible asignar o pequeñas ventanas de ayuda indicándonos los parámetros que precisa la función a la



que vamos a llamar. En muchos casos basta con seleccionar y aceptar para introducir directamente el nombre o valor adecuado, sin necesidad de terminar de escribirlo. La tecnología que hace esto posible se denomina *IntelliSense* en el caso de Visual Basic y *Code Insight* en el caso de Delphi. En este último existen posibilidades adicionales como las plantillas de código y una función que completa la declaración de nuevas clases.

Los editores de código de Delphi y Visual Basic utilizan colores y atributos en el texto para diferenciarlo sintácticamente y facilitar su lectura. Las listas desplegables existentes en el editor de Visual Basic facilitan parcialmente la funcionalidad que tiene el Explorador de código de Delphi.

Lenguaje

El lenguaje usado por Visual Basic es BASIC (*Beginner's All-purpose Symbolic Instruction Code*), un lenguaje desarrollado en la década de los sesenta por dos profesores estadounidenses, Kemeny y Kurtz, como una versión simplificada de FORTRAN (*FORmula TRANslator*), que fuese más accesible para los alumnos principiantes en computación.

Delphi utiliza el lenguaje Pascal, desarrollado poco después que BASIC, a principios de los setenta, por el profesor Niklaus Wirth en Zurich tomando como base los lenguajes ALGOL (*ALGO*rithms *LANG*uage) y PL/I. Pascal se llama así en memoria a Blaise Pascal, considerado el inventor de la primera máquina de cálculo.

El lenguaje Pascal que se utiliza en Delphi es conocido como Object Pascal, ya que es una versión de Pascal que cuenta con orientación a objetos, algo que, obviamente, no existía en el Pascal original. Muchos de los nuevos elementos existentes en Object Pascal están inspirados en el lenguaje C++, como la definición de clases o gestión de excepciones.

Ambos lenguajes, Visual Basic y Object Pascal, han evolucionado libremente según los deseos de sus respectivos fabricantes, a diferencia de otros lenguajes que están sujetos a comités de estandarización u organismos similares. Si en Visual Basic encontramos la experiencia de Microsoft en compiladores de BASIC, desde el ancestral GW-BASIC hasta las primeras versiones de Visual Basic pasando por QuickBasic, en Object Pascal está representada la experiencia de más de catorce años por parte de Inprise en el desarrollo de compiladores Pascal, primero en Turbo Pascal y después en Delphi.

Visual Basic es un lenguaje sencillo de aprender y, en general, bien estructurado, aunque en él aún permanecen elementos del pasado, como On Error Goto, que deberían haber sido sustituidos por estructuras más actuales y flexibles. A pesar de que es posible crear módulos de clases y manipular objetos de modo polimórfico, este mecanismo no se basa en un lenguaje totalmente orientado a objetos sino en el uso de interfaces COM. Es posible obtener direcciones de procedimientos, con el fin de usar llamadas *callback*, pero el manejo de punteros es oscuro e indocumentado por parte del fabricante, lo que redundará en un mayor esfuerzo por parte del desarrollador. La integración de Visual Basic con la tecnología ActiveX es sobresaliente. Puede utilizarse cualquier objeto que soporte automatización y, al

tiempo, cualquier objeto creado con Visual Basic se convierte ya en un objeto ActiveX. Para usar objetos COM simples, no automatizables, es necesario disponer de las librerías de tipos correspondientes, no existiendo modo alguno de crearlas desde el propio entorno de Visual Basic.

Object Pascal es un lenguaje totalmente orientado a objetos, muy estructurado, rico en tipos de datos, con total flexibilidad en la manipulación de punteros y la posibilidad de usar interfaces para ofrecer posibilidades como la herencia múltiple de otros lenguajes. Al igual que Visual Basic, Object Pascal está preparado para trabajar y crear objetos ActiveX, pero sin limitación alguna. Es posible crear clases que se comportan como objetos ActiveX, automatizables o no, y utilizar cualquier objeto de este tipo no supone problema alguno gracias al editor de librerías de tipos integrado en Delphi y la posibilidad de definir interfaces COM directamente en Object Pascal, sin necesidad de ningún elemento externo.

Es difícil convencer a un programador que utiliza habitualmente Object Pascal que el lenguaje Visual Basic es superior o viceversa. Sin embargo, analizando las posibilidades de ambos de una forma objetiva podemos obtener una conclusión clara: Object Pascal cuenta con todos los elementos y puede hacer todo lo que hace Visual Basic, mientras que lo contrario no es necesariamente cierto. No están muy claro qué beneficios obtendría un usuario de Visual Basic si éste tuviese herencia, por ejemplo, ya que la mayor parte de los programadores que utilizan esta herramienta no acostumbran a estructurar sus proyectos en clases que puedan derivar unas de otras ahorrando trabajo. Sí sería deseable, por el contrario, la posibilidad de utilizar objetos COM no automatizables sin tener que depender de terceras herramientas para la creación de las interfaces, o que el uso de punteros fuese más flexible y estuviese mejor documentado.

Depuración.

Difícilmente una aplicación llega a su estado definitivo sin pasar previamente por un proceso de depuración, ya que no todos los errores que se introducen pueden detectarse durante la compilación.

Ambas herramientas disponen de las habituales funciones de ejecución paso a paso, incluyendo las opciones de ejecución de procedimientos completos o instrucción a instrucción. Disponer o anular un punto de parada se efectúa de la misma forma: simplemente pulsando con el botón izquierdo del ratón en el margen izquierdo del editor de código, a la altura de la línea deseada. No obstante, en Delphi existe la posibilidad de crear puntos de parada que actúan al cargarse un determinado módulo o alterarse un cierto dato.

La modificación de valores durante la ejecución es posible en Delphi gracias a las opciones Evaluate/Modify e Inspect. En Visual Basic esta misma tarea puede realizarse de forma sencilla en la ventana de ejecución de inmediata. Esta ventana, además, permite ejecutar cualquier otro código que nos interese con algunas limitaciones lógicas, una posibilidad inexistente en Delphi. Es especialmente interesante para realizar pruebas durante el proceso de depuración Otra característica exclusiva de Visual Basic, muy útil en ocasiones, es la posibilidad de modificar el código que va a ejecutarse y continuar sin necesidad de reiniciar. En Delphi puede modificarse el código durante la depuración, pero los cambios no tienen efecto hasta que se vuelve a compilar y ejecutar.

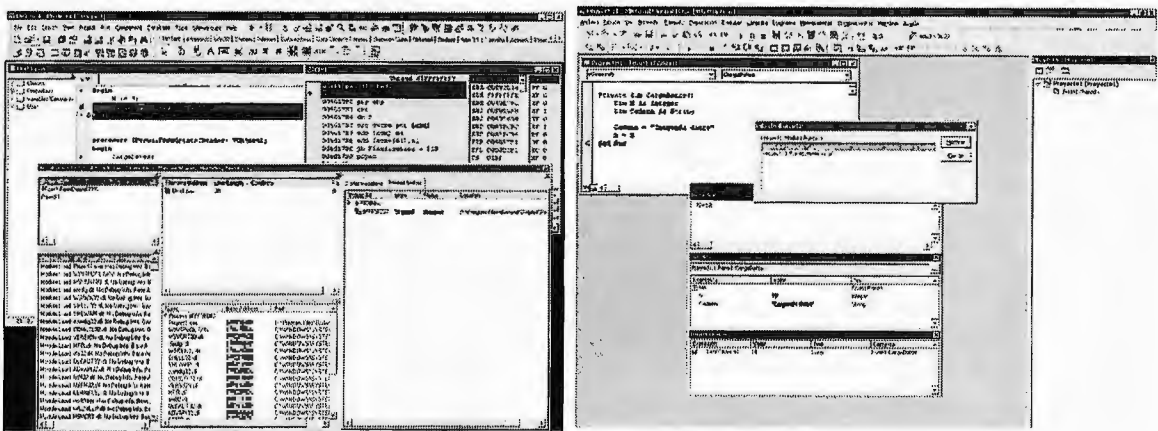
Tanto Delphi como Visual Basic cuentan con una pila de llamadas, que permite ver la secuencia que se ha seguido hasta llegar al punto actual, pero las posibilidades de Visual Basic prácticamente terminan ahí. Delphi, por el contrario, cuenta con ventanas que muestran el estado de los distintos hilos de ejecución y los módulos que hay cargados en cada momento. También dispone de un registro de eventos que permite analizar lo que sucede durante la ejecución. Hasta existe la posibilidad de ir a

la ejecución de más bajo nivel, viendo el estado del procesador, los registros, zonas de memoria e indicadores.

Tanto Visual Basic como Delphi cuentan con los elementos más habituales a la hora de depurar una aplicación, como las ventanas de variables locales, visores de identificadores, pila de llamadas o la ejecución paso a paso. En el entorno de Delphi pueden verse, además, ventanas con el estado de los hilos de ejecución, los módulos cargados por la aplicación, el visor de sucesos y la estado del procesador.

Creación y distribución de aplicaciones.

Todos los elementos vistos hasta ahora, entorno, componentes, editores y lenguaje, se utilizan con un fin primordial: la creación de una aplicación. Para conseguir este objetivo la fase de diseño y desarrollo se finalizará con la compilación y distribución del proyecto, tarea nada desdeñable si se tiene en cuenta la complejidad cada vez mayor de las aplicaciones.



Además de aplicaciones de tipo estándar, en forma de archivo ejecutable, tanto Visual Basic como Delphi disponen de muchas otras opciones que permiten crear proyectos de diversos tipos. Las posibilidades quedan reflejadas al seleccionar la opción New o Nuevo Proyecto del menú File/Archivo, momento en el que aparece la ventana conocida como repositorio o depósito de objetos. No existe, no obstante, una correspondencia directa entre el número de asistentes y elementos que aparecen en dicha ventana y los tipos de proyecto que es posible crear. Delphi, a excepción de

las aplicaciones HTML dinámico, es capaz de crear casi cualquier tipo de proyecto, mientras que Visual Basic tiene unas limitaciones mayores al estar fundamentado su funcionamiento en el mecanismo de automatización que, en parte, debe su existencia precisamente a Visual Basic.

Durante el proceso de desarrollo es habitual ejecutar innumerables veces el proyecto a fin de comprobar su funcionamiento. Tanto en Visual Basic como en Delphi basta una pulsación de tecla para poner en marcha la compilación y ejecución. La diferencia fundamental la encontramos en que Visual Basic realiza dicho proceso en memoria y a demanda, según se ejecuta el código, mientras que Delphi siempre crea un ejecutable y procesa todo el código del proyecto antes de iniciar la ejecución. Esto significa que en Visual Basic la ejecución es prácticamente inmediata, mientras que en Delphi es preciso esperar a que finalice la compilación.

Finalizado el proceso de desarrollo llega el momento de la compilación. Ambas herramientas disponen de múltiples opciones de optimización del código de objeto, la posibilidad de incluir información de versión en el ejecutable, establecer el título o el icono que se utilizará para representarlo. En el caso de Delphi podemos indicar si deseamos que los archivos intermedios de compilación sean en formato DCU, habitual en Delphi, o por el contrario queremos archivos OBJ con formato C o C++. Asimismo es posible generar un archivo de símbolos con información al nivel que se indique. En el caso de Visual Basic podemos elegir entre generar código nativo, directamente ejecutable, o bien *p-code* que es preciso interpretar posteriormente.

El seguimiento de los errores de compilación que efectúan las dos herramientas es también muy dispar. Visual Basic interrumpe la ejecución o la compilación, según el caso, en el momento en que encuentra el error, lo muestra en una ventana y desplaza el cursor hasta el punto adecuado en el editor de código. Tras realizar la corrección pertinente puede volver a ejecutarse o iniciar el proceso de compilación, repitiendo el proceso una y otra vez por cada error que se encuentre. El análisis de

errores en Delphi tiene lugar siempre durante la compilación, puesto que ésta es necesaria para poder ejecutar. El proceso no se detiene hasta que se ha analizado todo el código, indistintamente de si puede o no generarse el ejecutable. En caso de que se encuentren errores éstos aparecen en una ventana adosada al editor de código, facilitando una descripción junto con el nombre del módulo y la línea en la que se encuentra. Un doble clic sobre el mensaje o una pulsación de tecla son suficientes para llevar el cursor hasta el punto adecuado para realizar la corrección, tras lo cual puede continuarse con el siguiente error. Realizadas todas las correcciones el proyecto puede volver a compilarse. Una técnica que puede ser eficiente y hacer trabajar menos al programador.

Cuando la aplicación está preparada para utilizarse llega el momento de la distribución, cuya complejidad dependerá directamente de la complejidad del proyecto. Asumamos el caso más sencillo: una aplicación ejecutable con una o varias ventanas en la que utilizamos los componentes habituales en Windows. La aplicación generada por Visual Basic, con extensión EXE, podrá alojarse en cualquier carpeta del sistema del cliente. Además será preciso copiar la librería de ejecución de Visual Basic en esa misma carpeta o la carpeta de sistema de Windows. Por último, habrá que instalar y registrar todos los controles ActiveX que se hayan usado. El caso de Delphi es bastante más simple: basta con distribuir el archivo EXE generado, ya que contiene todo lo necesario. Aparte de ser un proceso más sencillo, en aquellas aplicaciones simples la inclusión de todo lo preciso en el mismo ejecutable genera habitualmente un paquete de distribución sensiblemente más pequeño.



La librería de ejecución de Visual Basic contiene todo lo necesario para el funcionamiento de una aplicación Visual Basic a excepción de los controles ActiveX. Ello explica el reducido tamaño que tienen los ejecutables generados por Visual Basic. Es un sistema muy eficiente cuando en un mismo sistema están ejecutándose múltiples aplicaciones creadas con la misma versión de Visual Basic, ya que parte del código es compartido entre ellas.

Integración con el sistema.

Tanto Delphi como Visual Basic son aplicaciones específicas para Windows, un sistema que cuenta con un gran conjunto de funciones (API) que permiten realizar todo tipo de operaciones, así como un importante número de componentes e interfaces ActiveX predefinidas. Un aprovechamiento total de las posibilidades del sistema implica que la herramienta de desarrollo sea flexible, permitiendo llamar a esas funciones, crear esos objetos y utilizar esas interfaces, ya sea implementándolas o usándolas para acceder a servicios del sistema.

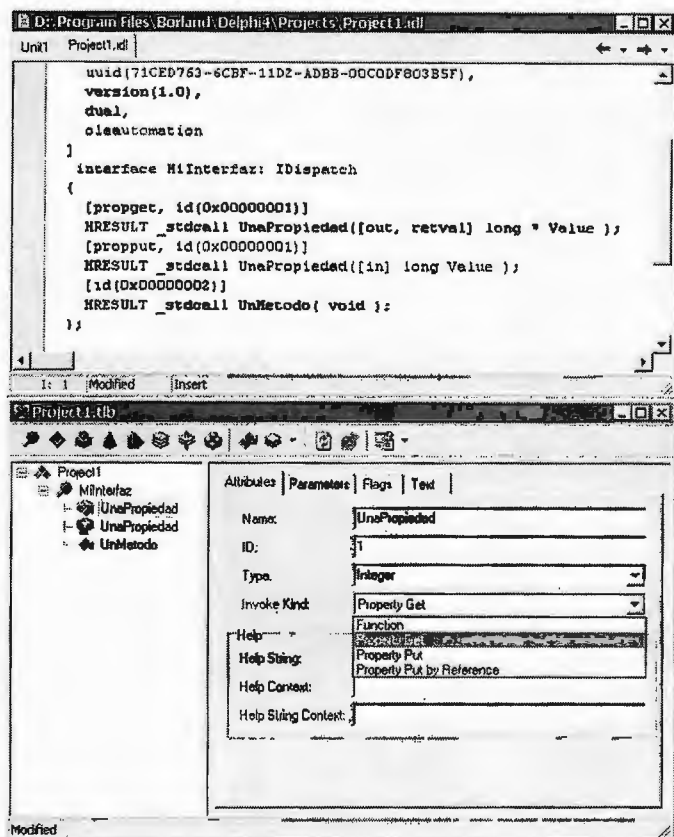
Para usar las funciones que componen la API de Windows es básico contar con tres elementos: un prototipo que permita conocer la localización, nombre, parámetros y tipo de retorno de la función, disponer de los tipos de datos adecuados y poder utilizar la convención de llamada que corresponda. Ambos lenguajes, Object Pascal y Visual Basic, cuentan con tipos de datos compatibles con los usados por las funciones de la API de Windows. También están preparados para usar la misma convención de llamada. La creación del prototipo es simple en ambos casos, basta con usar la sintaxis adecuada de propio lenguaje. La mayor diferencia es que Delphi dispone de múltiples módulos, equivalentes a los archivos de cabecera de C/C++, en los cuales se encuentran definidos los prototipos de prácticamente toda la API de Windows, conjuntamente con las estructuras de datos necesarias, los tipos y las constantes. Visual Basic, por el contrario, no cuenta con un mecanismo equivalente, lo que significa que antes de usar una función Windows siempre es preciso facilitar de antemano el prototipo que corresponda. Aunque en esta tarea se cuenta con una

utilidad, el Visor de texto de la API, lo cierto es que su uso no deja de ser tedioso, ya que hay que seleccionar de forma individual cada función, tipo de dato o constante, copiándolo al portapapeles y después pegándolo en nuestro módulo de código.

Desde Delphi es posible usar cualquier objeto ActiveX, independientemente de que sea o no automatizable. En caso de que no se disponga de una librería de tipos que describa las interfaces que implementa el objeto tampoco tendremos problemas. Podemos describir esa interfaz con código, usando la estructura Interfase que es similar a la definición de una clase. Más simple aún resulta el uso del editor de librerías de tipos ya que, de una forma visual, podemos ir definiendo los elementos de la interfaz y generando el código automáticamente. Incluso podemos elegir entre generar código Object Pascal o IDL (*Interface Definition Language*) estándar.

La implementación de interfaces ActiveX también es un campo ciertamente limitado en Visual Basic. Prácticamente las únicas interfaces estándar que es posible implementar son las de los controles ActiveX, algo que se produce automáticamente en el momento de crear un objeto de este tipo. Delphi, por el contrario, permite implementar cualquier interfaz en cualquier objeto, facilitando la creación de todo tipo de objetos COM, desde controles ActiveX, al igual que Visual Basic, hasta extensiones del sistema.

Mediante el editor de librerías de tipos ActiveX de Delphi la creación de librerías de tipos conteniendo definiciones de interfaces, objetos, enumeraciones y



tipos de datos es una tarea que se limita a algunas pulsaciones de ratón y la introducción de nombres para los diversos elementos. La pulsación de un botón genera automáticamente todo el código, ya sea en Object Pascal o IDL.

Documentación y ejemplos.

Para poder aprovechar todas las posibilidades de una herramienta de desarrollo es preciso disponer de la documentación adecuada, así como de la ayuda que nos permite resolver cualquier duda puntual de forma rápida. Todo ello puede complementarse con ejemplos demostrativos, ya sea pequeños fragmentos de código en la propia ayuda o aplicaciones completas. Si además todo ello lo encontramos en nuestra lengua materna, mejor que mejor.

Como es de esperar en productos de esta categoría, tanto Visual Basic como Delphi se acompañan de una correcta documentación impresa. En el caso de Visual Basic dicha documentación está en castellano, mientras que en caso de Delphi encontramos una combinación, puesto que la guía de inicio rápido y Object Pascal están en castellano pero el resto de la documentación está en inglés. Toda la documentación de Delphi puede encontrarse en formato electrónico, concretamente Acrobat Reader, en un directorio del CD-ROM en que se distribuye. Esto permite acceder a los manuales sin necesidad de manipular los volúmenes físicos.

La ayuda facilitada difiere tanto en formato como en contenido. Delphi sigue usando el tradicional formato de ayuda Windows, distribuyéndose los diferentes temas en otros tantos archivos independientes. Además de la ayuda relativa al entorno, el lenguaje Object Pascal, los componentes y otros temas concernientes al propio Delphi, también disponemos de archivos de ayuda con una referencia de la API general de Windows, así como temas más específicos relativos a OLE, OpenGL, MAPI, etc. Visual Basic usa el nuevo formato de ayuda HTML, disponiéndose de una interfaz

única para todos los archivos facilitados. No existe documentación sobre la API de Windows. A diferencia de la ayuda de Delphi, que siempre está disponible sin necesidad de insertar el CD-ROM del producto, para acceder a la ayuda de Visual Basic se precisan los dos CD-ROM adicionales en que se facilita este recurso. Al igual que ocurre con la documentación, la ayuda de Visual Basic está escrita en castellano, mientras que la de Delphi lo está en inglés.

Características exclusivas.

Una característica exclusiva de Visual Basic es la posibilidad de crear aplicaciones DHTML, consistentes en un documento HTML dinámico y una librería de código Visual Basic que es ejecutado desde dicho documento. De esta forma es posible crear documentos HTML dinámicos sin exponer el código, aunque su uso queda restringido a los usuarios de Internet Explorer. Es una opción muy válida sobre todo en *intranets*, donde el administrador tiene un total control sobre el *software* que utilizan los clientes.

En el haber de Delphi encontramos las opciones de creación de librerías de enlace dinámico con funciones y objetos, desarrollo de servicios NT, creación de aplicaciones multihilo sin necesidad de usar ejecutables ActiveX, creación de objetos COM automatizables y no automatizables y el desarrollo de objetos y módulos CORBA con servicios que pueden compartirse con otros lenguajes, como Java y C++. La inclusión de componentes de toma de decisiones y las amplias posibilidades de los componentes Internet también se han de tener en consideración, así como la facilidad para crear aplicaciones en múltiples idiomas.

Visual Basic cuenta con controles que facilitan tareas como las comunicaciones por puerto serie o la obtención de datos acerca del sistema, no existentes en Delphi. En contraposición, hay que decir que desde Delphi es posible utilizar cualquiera de los controles de Visual Basic, basta con añadirlo a la paleta de componentes, mientras que Visual Basic no puede usar los componentes de Delphi.

Conclusión.

No hay duda que tanto Delphi como Visual Basic son magníficas herramientas de desarrollo de aplicaciones para Windows que compiten prácticamente al mismo nivel en la mayoría de los aspectos tratados anteriormente, aunque hay aspectos en los que Delphi supera a Visual Basic.

Pero también, existen ciertos factores que tienen prioridad que nos hacen elegir a Visual Basic como la herramienta de programación para nuestro proyecto.

Retomando las preguntas hechas al inicio de este análisis, nos disponemos a contestarlas con lo expuesto anteriormente.

¿Qué se esta tratando de hacer?

Primeramente, lo que pretendemos hacer una aplicación para la simulación de una red de computadoras, con objetos programados que representen y se comporten como los elementos más comunes encontrados en una red, como lo son: router, computadoras, hubs, etc.....

Analizando un poco, nos hemos dado cuenta que la mayoría del tiempo que el usuario utilice el programa, será para la configuración de los elementos de la red, por lo tanto, un programa generado en Visual Basic ofrecerá el nivel de ejecución y rendimiento necesario que ofrece el código nativo. Además, el nivel de rendimiento del programa depende de la máquina donde se este ejecutando, ya que para realizar las otra tareas como hacer un *ping* o un *tracer*, se implementarán a través de funciones y procedimientos.

Delphi viene equipado una cantidad increíble de controles como ya se ha mencionado, pero el hecho es que los controles incorporados en Visual Basic son suficientes y funcionales, como lo son cajas de textos, botones de opciones, botones

de comandos, etc.; y en el caso de necesitar algún control, este puede ser construido o bien descargado de los muchos sitios que se dedican a dar soporte a la gran comunidad de programadores de Visual Basic.

¿Qué clase de características se desean implementar?

Delphi es una herramienta RAD, que debido a sus capacidades, esta mas orientada para la construcción de aplicaciones empresariales que necesitan manejar grandes cantidades de datos con el máximo rendimiento posible.

Una característica muy útil de mencionar de Visual Basic es la incorporación del **Control de Usuario**, que son controles ligeros que difieren de los controles normales por una razón muy significativa: no tienen un manejador de ventana (propiedad hWnd). Debido a esto, estos controles utilizan menos recursos del sistema.

Los controles de usuario pueden contener otros objetos ligeros, agregar nuevas propiedades y eventos personalizados. Un control como este es lo que necesitamos para representar los elementos de red, definiendo propiedades exclusivas de cada elemento.

¿Decidir *cuánto* se esta dispuesto a invertir en: dinero, entrenamiento y en tiempo de desarrollo?

Dinero:

Algo que hay que destacar es el bajo costo en que se tendrá que incurrir con la utilización de Visual Basic, ya que la Universidad Don Bosco ya cuenta con este software, subrayando que este ya esta instalado en el laboratorio de Biomédica, donde los alumnos reciben su laboratorio, por lo que no será necesario crear grandes discos de instalación ya se cuenta con todo lo necesario para poder ejecutar el programa, en las maquinas.

Entrenamiento:

Otro factor que influye es el tiempo con que contamos para realizar el proyecto. Como ya se mencionó el lenguaje que utiliza Visual Basic es muy fácil de aprender y codificar que el lenguaje Object Pascal que utiliza Delphi, también su IDE es intuitivo, simple y cómodo de usar, y los objetos que vienen con Visual Basic proporcionan cierta funcionalidad no presente en Delphi, que los hace más fáciles de utilizar.

Es de acentuar de que Visual Basic cuenta con ayuda totalmente en español, lo que facilitaría el entrenamiento y aprendizaje de esta herramienta RAD.

Tiempo de desarrollo:

Visual Basic hace muchas cosas por el programador. Por ejemplo, los objetos cuentan referencias y esto significa que, por decir algo, si creamos un objeto asignándolo a una variable local, el objeto será liberado automáticamente cuando la función o el procedimiento finalice (a menos que lo asignamos a una variable no local). Visual Basic tiene un sistema de administración sofisticado de memoria y utiliza un "colector de basura" (garbage collector) así que es rápido liberando memoria. En cambio en Delphi, los objetos no tienen referencias por lo que hay que estar pendientes de liberarlos de la memoria cuando se termine la función o el procedimiento.

Características como la tecnología *IntelliSense*, que aceleran significativamente el tiempo de desarrollo. Una cosa apreciable del IDE de Visual Basic es que cuando uno está depurando puede modificar una sentencia y continuar la ejecución con los cambios en efecto, sin tener que reiniciar y volver a compilar la aplicación. Otra característica agradable del Visual Basic es la ventana Inmediato, donde uno puede ejecutar sentencias interpretadas.

El compilador "real" con el que cuenta Visual Basic reduce el tiempo de desarrollo, al advertir de los errores en el momento que se escribe el código, sin tener que darse cuenta hasta el momento de la compilación (tomando en cuenta el tiempo que toma la compilación).

2. FACTIBILIDAD ECONÓMICA

Costo de Implementar el laboratorio de Redes a la UDB.

Costo de Implementación de Laboratorio de Redes a UDB

Cant	Dispositivo	Descripción	Unitario	Total
4	Router	Cisco Router 2610XM	\$2,420.00	\$9,680.00
1	Router	Cisco Router 2611XM	\$2,405.00	\$2,405.00
1	Switch	Cisco Catalyst 1912c	\$840.25	\$840.25
1	Switch	Cisco Catalyst 1924	\$935.00	\$935.00
3	Hub	BenQ de 8 puertos	\$33.05	\$99.15
5	Host	PC Clon / Windows XP Pro	\$667.26	\$3,336.30
			SubTotal	\$17,295.70
			IVA 13%	\$2,248.44
			Total	\$19,544.14

Nota: Cisco Router 2501 sustituido por Cisco 2610XM

Cisco Router 2514 sustituido por Cisco 2611XM

2.1 Costo de implementar el Simulador de Red para la UDB

2.1.1 Requerimientos

Sistema Operativo (Plataforma Windows)

Computadoras (200 MHZ o superior y 100 MB disponibles en disco duro, 32 MB de RAM)

Debido a que la Universidad Don Bosco ya cuenta con estos dos requerimientos para implementar el simulador, esta no incurrirá en gastos.



Costos para alumno por hacer uso de equipo real de redes

Toda persona que desea inscribirse en la Academia de Redes CISCO Universidad Don Bosco, tiene que inscribir el semestre o modulo 1 con un costo de \$110.00 para los hombres y \$100.00 para las mujeres. Para completar la academia cada persona tiene que completar 3 semestres o módulos más.

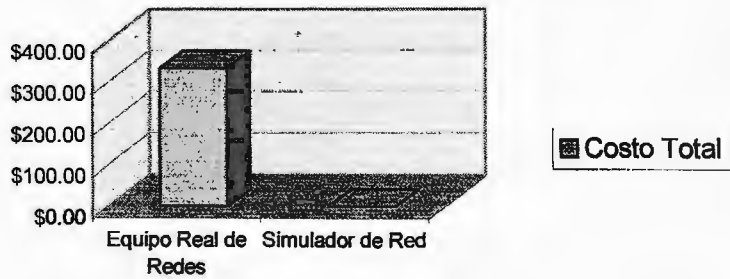
Con este pago el estudiante tiene derecho al curriculum online de cada semestre es decir a la teoría a través de un sitio web de CISCO, a las Instructorías de cada capitulo por parte de un Instructor Certificado por Cisco y acceso al laboratorio practico de redes para realizar los laboratorios de cada semestre o modulo guiado por el instructor.

Como los dispositivos implementados en el simulador están explicados en los primeros tres semestres de la academia, para fines de análisis se ha tomado como base este costo de \$330 para hombres y \$300 para mujeres como costo por hacer uso equipo real de redes.

2.1.2 Costo para alumno por hacer uso de Simulador de Redes

Algunas materias tienen laboratorios prácticos, al inscribirlas se cancela el costo de dicho laboratorio independientemente qué se utilice, es por eso que no tendría costos adicionales para que el alumno haga uso de dicha herramienta.

Comparativos de Costos para el Alumno



2.1.3 Costo del proyecto para grupo de investigadores

Papelería	\$ 110.00
Impresiones (Cartuchos de Tinta)	\$ 120.00
Internet	\$ 160.00 (\$ 20.00 mensuales)
Electricidad	\$ 126.25 (\$ 25.25 mensuales)
Varios	\$ 100.00

Total \$ 616.25

Nota: los costos de electricidad e Internet son estimados que se basan en los recibos mensuales de la residencia donde se van a efectuar las reuniones.

3. FACTIBILIDAD OPERATIVA

El Simulador de Red para el laboratorio de la materia de redes área local (LAN) de la Universidad Don Bosco, será una aplicación Windows completamente amigable y ejecutable en cualquier Sistema Operativo Windows (9x, Me, NT, 2000 y XP).

Su uso esta designado para los estudiantes de la materia de redes LAN y toda persona interesada de la Universidad Don Bosco en aprender el funcionamiento de las redes y configuración de los dispositivos que forman parte de las redes.

Para su instalación los archivos de instalación estarán en un CD con un archivo ejecutable (setup.exe), este cargará el asistente el cual instalará en el disco duro local todos los archivos, librerías y componentes necesarios para el simulador de red. Luego de instalado, el usuario será capaz de cargar el simulador de red y comenzar a diseñar las redes y configurar los dispositivos que necesite para esa red.

El usuario trabajará bajo un ambiente gráfico, de ventanas, menús y formularios. Para poder simular una red el usuario dispondrá de una pantalla de diseño donde se agregaran los dispositivos de red que se encuentran en el simulador los cuales son routers, switches, hubs, y hosts necesarios para diseñar la red de laboratorio y una vez agregados a la plantilla configurar cada uno según las limitantes de configuración que los dispositivos poseen. Después de unir cada uno de los dispositivos con los cables específicos, el simulador tiene dos herramientas de prueba que están disponibles en esta versión.

Las herramientas PING, TRACERT simularan lo que estas realizan en la realidad, utilizándolas como herramientas de prueba.

El simulador de red contará con un modulo para evaluar al estudiante de redes. Este modulo consiste en laboratorios prediseñados donde se evaluará por temas específicos los conocimientos y habilidades que el estudiante posee, luego de resuelto el laboratorio el simulador automáticamente lo calificara y presentara los resultados obtenidos en pantalla y/o impresos en papel para que el estudiante o usuario pueda ver como andan sus conocimientos y habilidades en los aspectos evaluados y así pueda mejorar.

Además el usuario del simulador tendrá la capacidad de ver un historial de los procesos internos de enrutamiento del router.

Estudio comparativo entre la implementación de un laboratorio real de redes y un simulador de redes

Laboratorio Real de Redes

Ventajas

- Acceso a todo el software del Router y Switch
- Acceso real al equipo (Manipulación real, configuración real etc.)

Desventajas

- Costo elevadísimo
- Posibilidad de dañar equipo o arruinarlos por completo
- Costo en reparación y mantenimiento alto
- Máximo de estudiantes por laboratorio: 5 estudiantes

Simulador de Redes

Ventajas

- Costo de elaboración e implementación baja
- Un promedio de estudiantes por laboratorio: 30
- No se incurren en gastos de reparación o mantenimiento
- No se existe la posibilidad de dañar equipo por ser un software

Desventajas

- No se tiene acceso a todo el software del router y switch, pero si a lo necesita un estudiante de redes
- No se puede manipular el equipo real

Por tanto operativamente es factible la implementación del simulador red en la Universidad.

La tesis contará con los manuales de usuario para poder utilizar el simulador y poder explotar al máximo todas sus características. También estará disponible el manual del programador, para futura expansión y mejoras de la herramienta.

Se capacitará a la persona encargada del laboratorio de redes LAN para que sepa usar el simulador.

CAPITULO III

METODOLOGÍA DE LA INVESTIGACION

1. METODOLOGIA DE LA INVESTIGACIÓN

Para el desarrollo de este proyecto se tomará como base el laboratorio de Cisco de la Universidad Don Bosco, haciendo las visitas necesarias a este laboratorio para recolectar información.

Dicho laboratorio ofrece el ambiente apropiado para conocer el funcionamiento de una red de área local (LAN por las siglas en ingles Local Area Network), al contar con los elementos básicos que se encuentran en una LAN.

Con la finalidad de llegar a entender tal funcionamiento, que servirá en el desarrollo del software de simulación, será necesario emplear el método de análisis y desarrollo de sistemas de simulación.

Este método plantea una serie ordenada de etapas que con llevan a una interacción entre analistas, diseñadores y usuarios.

Las etapas que pueden distinguirse en el desarrollo son:

1. *Formulación del problema.*

En esta etapa se establece cual será el objeto que se va a simular, para nuestro caso el objeto a simular es una **red de área local** (LAN). Además de definir algunos factores importantes como lo son:

- a. Los resultados que se esperan del simulador.
- b. Plan de experimentación.
- c. Tiempo disponible.

d. Variables de interés.

e. Complejidad de la interfaz del simulador.

En la etapa también se incluye si el usuario operará el simulador o simplemente recibirá resultados.

2. *Definición del sistema.*

El sistema a simular se define con mucho detalle. El cliente y el desarrollador acuerdan las fronteras del entorno a estudiar y cuales serán las interacciones del medio ambiente real que se tomarán en cuenta.

3. *Formulación del modelo.*

En esta etapa se desarrolla un modelo sencillo que encierra aspectos relevantes del entorno real.

4. *Colección de datos.*

La naturaleza y cantidad de datos necesarios para el desarrollo del programa de simulación están determinadas por las etapas anteriores de formulación del problema y del modelo. Estos datos pueden ser obtenidos mediante registros históricos, experimentos de laboratorio, observaciones directas o mediciones en el sistema real; estos serán procesados para darles el formato exigido por el modelo.

5. *Desarrollo del modelo en la computadora.*

Es la implementación del modelo planteado utilizando el lenguaje de programación seleccionado.

6. *Verificación.*

Se hacen experimentos para comprobar el funcionamiento del sistema y asegurarse de que no se cometió algún tipo de error en la etapa anterior.

7. *Validación y experimentación.*

En la etapa de validación se comprueba la exactitud del modelo desarrollado con datos verdaderos o datos similares para comparar los resultados con los recolectados en las observaciones.

8. *Interpretación.*

Se realiza un análisis de la sensibilidad del sistema con respecto a datos que tengan asociados una mayor incertidumbre y si es necesario, se recolectaran datos adicionales con el fin de refinar el sistema para el tratamiento de dichos datos.

9. *Implementación.*

Con esta etapa, el sistema es instalado a los usuarios para su uso y para que puedan evaluarlo y proporcionar una retroalimentación para posibles mejoras. Conviene acompañar al usuario en esta etapa para evitar el mal manejo del simulador o el mal empleo de los resultados.

10. *Documentación.*

Es la elaboración de la documentación técnica y manuales de uso. La documentación técnica incluye la descripción detallada del modelo. Esta documentación será de utilidad para el perfeccionamiento futuro del simulador.

2. TECNICAS A UTILIZAR EN LA INVESTIGACIÓN

Las técnicas de investigación a utilizar se listan a continuación:

- Análisis y diseño de sistemas.
- Técnicas de abstracción de datos.
 - Observación directa: será el primer contacto con el entorno a estudiar y con las personas involucradas a él. La observación directa también incluye conversaciones con cualquier otra persona implicada con el sistema a desarrollar.

Recopilación de información y de formatos que utilizan los usuarios para la construcción y configuración de una red, mediante preguntas a las personas conocedoras e involucradas con el laboratorio de Cisco.

3. INVESTIGACION PRELIMINAR Y RECOLECCION DE INFORMACIÓN

La materia de Redes LAN es impartida por la Universidad Don Bosco solo en el ciclo par de cada año; la cual consta de una clase teórica y un laboratorio practico que son realizados cada semana.

La clase teórica tiene lugar en un aula del edificio B ó C de la universidad, que son utilizados para las clases expositivas.

El laboratorio, por otra parte, es realizado en el edificio No. 3 de Electrónica, segunda planta, del Centro de Investigaciones y Transferencia de Tecnología (CITT), en el laboratorio de Biomédica.

Estudio del Entorno objetivo del proyecto.

Laboratorio de Biomédica.

Este laboratorio está equipado con 11 computadoras con procesador AMD, con 96 MB de memoria RAM y Windows 98 y un software para simular la configuración de un router utilizado por el laboratorio de Cisco. Se incluyeron varios hubs para conectar las computadoras.

Esta información fue proporcionada por Luis Orellana, instructor laboratorista, quien estuvo encargado de la materia en el año 2001 y según nos comentó, podría volver a ser el encargado de dicha materia.

Además, mencionó que se utilizó el laboratorio de telecomunicaciones, ubicado también en el edificio de Electrónica, que cuenta con tecnología de fibra óptica y microondas para llevar a cabo algunas prácticas.

El número de personas atendido en esa ocasión fue de aproximadamente 35 personas, distribuidas en dos grupos de laboratorio a horarios diferentes.

Entorno base del proyecto a estudiar.

Laboratorio de Cisco.

El laboratorio de Cisco esta ubicado en el edificio 5 del CITT, segunda planta y cuenta con 20 computadoras Pentium III con 128 MB de memoria RAM y Windows 2000 Professional conectadas a Internet, que son utilizadas por los alumnos para leer las lecciones teóricas de cada modulo.

Además, se dispone de 10 computadoras para el laboratorio práctico, 5 routers y 4 pares de cables DTDC para conectar los routers, 2 switches y otros elementos como hubs, cables, etc.

Equipo de la Academia de Redes de Cisco en la Universidad Don Bosco

Cantidad	Marca	Modelo	Dispositivo	Descripción
4	Cisco	2501	Router	2 Interfaces Seriales y 1 Interfaz Ethernet
1	Cisco	2514	Router	2 Interfaces Seriales y 2 Interfaz Ethernet
1	Cisco	Catalyst 1912C	Switch	12 puertos
1	Cisco	Catalyst 1924	Switch	24 puertos a 10MBPs y 2 puertos de 100MBPs
3			Hub	8 puertos cada uno de 10mpbs

En el laboratorio de Cisco se cuenta con un simulador de router, que es un programa que esta diseñado para complementar los trabajos de laboratorio, debido a que ayuda a los estudiantes a aprender la configuración del router y solucionar problema de la red sin necesidad de estar conectado a un router. Este simulador permite al estudiante probar sus configuraciones usando comandos como "Show Run", "Show IP Route", "Ping", "Trace" y "Telnet". Por ejemplo, el simulador generará información en la tabla de ruteo mostrando las rutas que los datos seguirán entre routers si estos estuvieran configurados correctamente. Se pueden utilizar comandos para probar si la conexión fue establecida, como lo es el *ping* ó *trace*, que fallará si los router están configurados incorrectamente. Así, aunque los estudiantes no posean acceso físico a los router, ellos pueden ganar una gran experiencia trabajando con los comandos del router.

Aunque este simulador no soporta todos los comandos del router, si soporta la mayoría de los que son usados en los laboratorios.

Esta información fue proporcionada por el Ingeniero Erick Flores, certificado de la academia de Cisco como CCNA y actual docente de la academia Cisco en la Universidad Don Bosco.

CAPITULO IV

DESCRIPCIÓN DEL PROYECTO

1. SITUACION ACTUAL

La situación actual del laboratorio de redes LAN es la siguiente:

Los alumnos llegan al laboratorio de Biomédica al horario establecido para realizar su práctica de redes LAN con el objetivo de comprobar los conocimientos vistos en clase.

Por el momento no se cuentan con guías de laboratorio que los estudiantes puedan seguir para desarrollar su laboratorio, por lo que el instructor tiene que explicar a los alumnos los conceptos adicionales y lo que tienen que hacer en la práctica.

Además, a los alumnos se les pondera una nota por los resultados obtenidos en las prácticas del laboratorio para alcanzar un promedio de la materia.

Básicamente, lo que los alumnos realizan en sus practicas son: configuración de computadoras para prepararlas en el entorno de red, que exista una correcta comunicación entre dichas computadoras, utilización de comandos para probar la configuración y comunicación (por ejemplo ping), verificar resultados (por ejemplo, utilización del comando *tracert* para conocer la ruta que siguieron los datos para llegar a otra computadora), etc.

2. RESULTADOS ESPERADOS

Los resultados que se pretenden alcanzar con el desarrollo del simulador de redes LAN:

- Facilitar a los estudiantes de Redes LAN de la Universidad Don Bosco de una herramienta grafica, amigable e interactiva por medio de la cual realicen sus prácticas.
- Con la herramienta se realizaran diferentes diseños de redes, y la configuración de cada uno de los dispositivos como hosts, switches y routers según los requerimientos de las guías de laboratorio o el diseño de cada uno de los usuarios y con la capacidad de comprobar por medio de dos herramientas si el diseño y la configuración funcionan o no.
- Dar la oportunidad a los estudiantes de la materia de redes LAN de la Universidad Don Bosco, de poder practicar los conceptos de redes en un laboratorio virtual.
- El usuario podrá simular una red LAN compuesta por los siguientes dispositivos: Routers, hosts, Switches, Hubs; de una manera dinámica, es decir, que no abra una red ya pre-establecida como la del simulador actual, sino que se podrá crear y modificar en tiempo de diseño.
- Mejorar la calidad de enseñanza de redes LAN y el desarrollo de habilidades en el manejo de dispositivos de red utilizados en el simulador, en los estudiantes de la Universidad Don Bosco que tenga acceso a la herramienta.
- Realizar un simulador que tenga la capacidad de expansión para futuras mejoras o requerimientos.
- Facilitar a la Universidad Don Bosco de una herramienta que pueda formar mejores profesionales de redes a un bajo costo.
- Verificar mediante una ventana de historial, los procesos internos de enrutamiento del router.

- Realizar pruebas de evaluación ya preestablecidas.
- Guardar diseño y configuración hecha por el usuario.
- Las configuraciones que el usuario podrá realizar son las siguientes:

Dispositivo	Configuración
Host	Dirección IP, Mascara de red, Default Gateway y Nombre de Host
Hub	Ninguna
Routers	Interfaces, Seguridad, Enrutamiento
Switches	VLAN's

CAPITULO V

MARCO TEORICO

1. MARCO HISTORICO

SIMULACIÓN

Inicialmente el uso de las ecuaciones y sistemas de ecuaciones diferenciales, han sido las herramientas más usadas para modelar sistemas. Para el siglo XIX el impresionante arsenal matemático, base de toda la ciencia clásica, empezó a mostrar su insuficiencia ante dos circunstancias que afectan al estudio de los sistemas: la incertidumbre y la imprecisión.

Esta dificultad se ha reducido en gran medida gracias al uso general de las llamadas técnicas de simulación, las cuales permiten el modelado y estudio de muchos sistemas de estructura compleja, obteniendo soluciones numéricas aproximadas.

A pesar de su utilidad la simulación no puede considerarse como una panacea capaz de resolver todo tipo de situaciones, aún contando con la ayuda de los lenguajes especializados para la simulación, o de los avances que han representado los entornos de software específico para simulación, con simuladores visuales, etc. .

Sin embargo la simulación, por sus características, y por los desarrollos tecnológicos en las computadoras que se han conseguido en los últimos años, sigue presentando una serie de ventajas que no solo la convierten en el procedimiento más adecuado en muchos casos, sino que hacen que sea la única alternativa tecnológica en muchos otros. Esto resulta especialmente obvio en aquellos casos en los que las características del sistema que se pretenden estudiar hacen inviable, por razones físicas o de costo, la experimentación directa sobre el sistema.

La construcción de modelos se inició de manera oficial desde la época del renacimiento, la historia de la simulación surge en los años cuarenta con los científicos Von Newman y Stanislaw Ulam que trabajaban en el Proyecto Manhattan, hicieron referencia a la simulación Monte Carlo, en el Laboratorio Nacional de los Alamos de California, durante la segunda guerra mundial, resolvieron problemas de

reacciones nucleares cuya solución experimental sería muy costosa y el análisis matemático demasiado complejo.

La simulación de eventos discretos por computadora digital comenzó hace unos treinta años, y como método de análisis, ha servido para estudiar una infinidad de problemas. Una gran parte de estos problemas son de los que se estudian por la llamada teoría de las colas o de las filas.

Desde los primeros años de la década del sesenta se realizaron estudios de simulación cuyos costos se medían en años-hombre y su duración en meses. Para este tiempo la simulación era uno de los muchos métodos usados para la ayuda en la toma de decisiones estratégicas dentro de la industria. Su principal fuerza radica en la habilidad para imitar problemas complejos del mundo real y analizar el comportamiento del sistema cuando el tiempo progresa. También para la década del sesenta las Técnicas Orientadas a Objetos fueron resultado del desarrollo de la simulación. Recientemente hay solo una aplicación de simulación Orientada a Objetos disponible en el mercado.

En los setenta aparecieron diversos lenguajes específicamente orientados a la simulación tales como SIMSCRIPT, etc. La Simulación Interactiva Visual (VIS por sus siglas en inglés) estuvo disponible desde los últimos años de la década del setenta. Antes estos modelos de simulación eran simples *cajas negras*, los datos entraban y resultados luego salían. En el cual un escenario establecía la credibilidad y la confianza en el modelo de simulación no podría ser fácil. La década de los ochenta supuso la adaptación sobre PC de productos ya existentes y la aparición de nuevos productos como SIMAN.

Los noventa han protagonizado hasta ahora una auténtica explosión de nuevos productos de manejo más intuitivo bajo entornos gráficos como Windows.

REDES

Las redes fueron el resultado de las aplicaciones de las empresas. En este entonces dichas aplicaciones fueron diseñadas para computadoras independientes y su funcionamiento era individual a las demás computadoras de la empresa.

Esto presento un problema para las empresas debido a que esta manera de funcionar no era muy rentable debido a que existía demasiada duplicación de los equipos y recursos.

Las empresas llegaron a la conclusión que podrían aumentar la productividad y reducir los costos utilizando la tecnología de redes. Se inicio añadiendo redes y expandiendo aquellas que ya existían, esto sucedió tan rápido como la introducción de nuevas tecnologías y productos de red. A principio de los años 80, las redes se extendieron rápidamente, pero este temprano crecimiento resulto en diversos problemas.

A mediados de los 80, la mayoría de las tecnologías de red fueron creadas con una diversidad de implementaciones de software y hardware distintas. Esto generó un problema de incompatibilidad. La comunicación entre redes de distintas especificación era casi imposible.

Entre las primeras soluciones a estos problemas, surgió la creación de la redes de área local (LAN). Ya que este tipo de redes conectan todos los hosts, dispositivos periféricos, terminales y otros dispositivos dentro de un mismo edificio, este tipo de redes permitió poder compartir archivos e impresoras.

Con el rápido crecimiento de las computadoras, aún las LAN no dieron a vasto ya que cada empresa o departamento, se convirtió en una isla electrónica.

El problema acá era la manera de poder unir estas islas de tal forma que la comunicación entre ellas fuera de una manera rápida y eficiente. Las redes de área metropolitana (MAN) y las redes de área amplia (WAN) fueron la solución a este problema. Ahora ya era posible comunicar dos o más redes entre sí a grandes distancias.

Dispositivos de red

1. En los años 40, las computadoras eran grandes dispositivos electromecánicos con posibilidades a fallar.
2. En 1947, se invento el transistor semiconductor el cual permitió computadoras más pequeñas y con menos errores.
3. En los años 50, las computadoras mainframe, que funcionaban con tarjetas perforadas, fueron utilizadas por las más grandes instituciones. En los últimos años de esta década, se creó el circuito integrado. En los años 60, los mainframes con terminales eran comunes y los circuitos integrados comenzaron a utilizarse.
4. En los últimos años de los años 60 y en los años 70 surgieron las computadoras más pequeñas denominadas *mini computadoras*.

La arquitectura abierta y la micro miniaturización de los circuitos integrados dio como resultado Las PCs.

5. En los últimos años de la década de los 80, los usuarios con sus computadoras individuales comenzaron a compartir archivos y recursos. Entonces surgió la inquietud de conectar las computadoras entre sí.

A la par los sistemas telefónicos avanzaban, llegando a la confiabilidad, especialmente en las áreas de conmutación y larga distancia.

6. Desde la década de los 60 hasta la década de los 90, el Departamento de Defensa de los Estados Unidos desarrolló las redes de área amplia. Parte de esta tecnología se utilizó en el desarrollo de las LAN y con el tiempo las WAN del Ministerio de Defensa origino a Internet.

Los siguientes avances tecnológicos fueron las tarjetas de red, el medio, los repetidores y hubs para comunicar a distancias más largas.

Después surgió la necesidad de que los grupos de trabajo se comunicaran entre si y como los hubs generaban mucho broadcast a todos los puertos, a medida crecía el numero de computadoras conectadas a los puertos se producían embotellamiento y entonces frente ha este problema se desarrollo el puente para así poder segmentar la red.

Los primeros switches LAN fueron desarrollados en 1990. Ellos fueron dispositivos de capa 2 llamados puentes dedicados a resolver problemas de ancho de banda. Recientemente los switches LAN involucran dispositivos multicapas capaces de manejar las asignaciones de los protocolos envueltos en las aplicaciones alto ancho de banda que históricamente tienen que ser resueltas por los routers. Ahora los switches LAN son usados para reemplazar los hubs en el armario porque las aplicaciones de usuario demanda gran ancho de banda.

La mejor característica del hub (concentración / conectividad) y la mejor característica del puente (segmentación) se mezclaron y crearon el switch.

En la mitad de los años 80, se crearon los gateways que eran computadoras para fines especiales y luego se llamaron routers. Estos interconectan LAN individuales, seleccionando la mejor ruta y usando conmutación para los datos.

7. Protocolo TCP/IP

El conjunto de protocolos de la suite TCP/IP se desarrollo como una parte de la investigación hecha por la Agencia de Proyectos de Investigación Avanzada para la Defensa (DARPA).

Originalmente fue para efectuar comunicaciones a través de DARPA.

Luego, TCP/IP se fue parte en la Distribución del Software Berkeley de UNIX. TCP/IP se ha convertido en el estándar para las comunicaciones de redes y funciona como protocolo de transporte para Internet, permitiendo que gran numero de computadoras en todo el mundo puedan estar comunicadas entre si.

2. MARCO CONCEPTUAL

SIMULACION

Simulación, del latín simulare, que significa "imitar" o "repetir".

Definición de simulación

Algunas de las definiciones más aceptadas y difundidas de la palabra simulación son las siguientes: Thomas H. Naylor, la define como *"Simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento de sistemas complejos del mundo real a través de largos períodos de tiempo"*.

La definición anterior está en un sentido muy amplio, pues puede incluir desde una maqueta, hasta un sofisticado programa de computadora. En sentido más estricto, H. Maisel y G. Gnugnoli, definen simulación como: *"Simulación es una técnica numérica para realizar experimentos en una computadora digital. Estos experimentos involucran ciertos tipos de modelos matemáticos y lógicos que describen el comportamiento de sistemas de negocios, económicos, sociales, biológicos, físicos o químicos a través de largos períodos de tiempo"*.

En este punto, es conveniente plantear las siguientes definiciones:

Sistema: Conjunto de objetos o ideas que están interrelacionados entre sí como una unidad para la consecución de un fin (Shannon, 1988). También se puede definir como la porción del Universo que será objeto de la simulación.

Modelo: Un objeto X es un modelo del objeto Y para el observador Z , si Z puede emplear X para responder cuestiones que le interesan acerca de Y (Minsky).

Simulación: Simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con él, con la finalidad de aprender el comportamiento del sistema o de evaluar diversas estrategias para el funcionamiento del sistema (Shannon, 1988).

Aplicaciones de la simulación

La simulación es conveniente cuando:

No existe una formulación matemática analíticamente resoluble. Muchos sistemas reales no pueden ser modelados matemáticamente con las herramientas actualmente disponibles.

Existe una formulación matemática, pero es difícil obtener una solución analítica.

No existe el sistema real. Es problema del ingeniero que tiene que diseñar un sistema nuevo. El diseño del sistema mejorará notablemente si se cuenta con un modelo adecuado para realizar experimentos.

Los experimentos son imposibles debido a impedimentos económicos, de seguridad, de calidad o éticos. En este caso el sistema real esta disponible para realizar experimentos, pero la dificultad de los mismos hace que se descarte esta opción.

El sistema evoluciona muy lentamente o muy rápidamente.

Entre las posibles desventajas de la simulación se pueden citar:

El desarrollo de un modelo puede ser costoso, laborioso y lento.

Existe la posibilidad de cometer errores. No se debe olvidar que la experimentación se lleva a cabo con un modelo y no con el sistema real; entonces, si el modelo está mal o se cometen errores en su manejo, los resultados también serán incorrectos.

No se puede conocer el grado de imprecisión de los resultados. Por lo general el modelo se utiliza para experimentar situaciones nunca planteadas en el sistema real, por lo tanto no existe información previa para estimar el grado de correspondencia entre la respuesta del modelo y la del sistema real.

Actualmente la simulación presta un invaluable servicio en casi todas las áreas posibles, algunas de ellas son:

Procesos de manufacturas: Ayuda a detectar cuellos de botellas, a distribuir personal, determinar la política de producción.

Plantas industriales: Brinda información para establecer las condiciones óptimas de operación, y para la elaboración de procedimientos de operación y de emergencias.

Sistemas públicos: Predice la demanda de energía durante las diferentes épocas del año, anticipa el comportamiento del clima, predice la forma de propagación de enfermedades.

Sistemas de transportes: Detecta zonas de posible congestionamiento, zonas con mayor riesgo de accidentes, predice la demanda para cada hora del día.

Construcción: Predice el efecto de los vientos y temblores sobre la estabilidad de los edificios, provee información sobre las condiciones de iluminación y condiciones ambientales en el interior de los mismos, detecta las partes de las estructuras que deben ser reforzadas.

Diseño: Permite la selección adecuada de materiales y formas. Posibilita estudiar la sensibilidad del diseño con respecto a parámetros no controlables.

Educación: Es una excelente herramienta para ayudar a comprender un sistema real debido a que puede expandir, comprimir o detener el tiempo, y además es capaz de brindar información sobre variables que no pueden ser medidas en el sistema real.

Capacitación: Dado que el riesgo y los costos son casi nulos, una persona puede utilizar el simulador para aprender por sí misma utilizando el método más natural para aprender: el de prueba y error.

Tipos de simulación

Por lo tanto en función de la naturaleza de los sistemas a simular, las técnicas de simulación convencional se dividen en cuatro grupos:

Simulación de sistemas continuos.

Simulación de procesos por lotes.

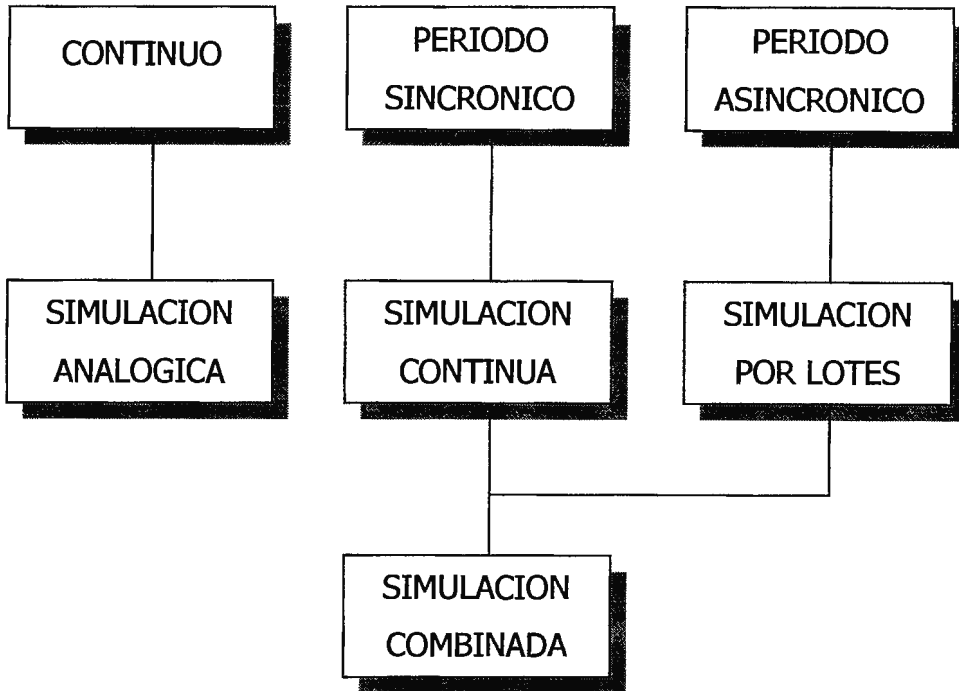
Simulación combinada.

Simulación de Monte Carlo.

El criterio que se toma para establecer esta división es la distinta utilización de la variable que se emplea para el control de la simulación, el tiempo.

En la figura siguiente se muestra un diagrama de los distintos tipos de simulación existentes y su relación con el tiempo.

El tiempo es intrínsecamente una variable continua; así, para realizar una simulación real de un proceso se debería manejar un tiempo que variase de forma completamente continua, lo que únicamente es posible en la simulación analógica.



Tipos de simulación y su relación con el tiempo

Existe otra serie de procesos, denominados por lotes, en los cuales no es precisa una evolución continua del tiempo, ya que las modificaciones que existen en el sistema a simular se produce cada ciertos intervalos de tiempo, siendo éstos de valor elevado y espaciados no uniformemente en el tiempo. En este caso se precisa un reloj asíncrono, cuya actualización depende de los eventos producidos en el sistema.

La simulación combinada contempla procesos mixtos en los que existen subprocesos de tipo continuo, junto con subprocesos en lotes.

Uno de los tipos de simulación de mayor novedad es la Simulación Cualitativa, en la que se utilizan modelos simbólicos de sistemas continuos que son capaces de predecir la evolución de los sistemas. En este tipo de simulación no se manejan valores numéricos del tiempo, utilizándose los conceptos de instantes e intervalos de

tiempo entre los que se puede fijar únicamente una relación de orden (anterior-posterior).

Por último, existen una serie de sistemas en los que el factor tiempo no influye significativamente, produciéndose las técnicas de simulación mediante el método de Monte Carlo.

Sistemas

Un sistema es una sección de la realidad que es el foco primario de un estudio y está compuesto de componentes que interactúan con otros de acuerdo a ciertas reglas dentro de una frontera identificada para el propósito del estudio. Un sistema puede realizar una función que no es realizable por sus componentes individuales.

Los objetos o componentes que forman parte del sistema se denominan *entidades*. Estas entidades poseen propiedades denominadas *atributos*, por y se relacionan entre sí a través de *relaciones* o *funciones*. Estas relaciones pueden ser:

Estáticas o estructurales: un auto posee cuatro ruedas.

Dinámicas o funcionales: un auto consume nafta si se enciende el motor.

Los valores asumidos por los atributos de las entidades en un momento dado determinan el *estado* del sistema. El estado puede ser *estático* o *estacionario*, esto significa que se mantiene constante en el tiempo; o por el contrario, puede ser *dinámico* o *transitorio* si evoluciona con el tiempo. Un sistema puede presentar los dos tipos de conductas; generalmente, cuando inicia su funcionamiento pasa por un estado dinámico y luego alcanza un estado estacionario o de régimen.

Un estado estacionario es *estable* si el sistema retorna a él luego de una perturbación. Por el contrario, un estado estacionario es *inestable* si el sistema se aleja de él luego de una perturbación. Este alejamiento puede dar lugar a una respuesta *acumulativa* (crece o decrece continuamente, o alcanza otro estado estacionario) o a una respuesta *oscilatoria* (crece y decrece continuamente).

Clasificación de sistemas

De acuerdo a su naturaleza, un sistema puede ser (Law and Kelton, 1991):

Determinístico: Si el sistema no contiene ningún elemento aleatorio es un sistema Determinístico.

Estocástico: En este caso algún elemento del sistema tiene una conducta aleatoria.

Continuo: Se tiene un sistema continuo cuando las relaciones funcionales entre las variables del sistema sólo permiten que el estado evolucione en el tiempo en forma continua (basta que una variable evolucione continuamente).

Discreto: Se tiene un sistema discreto cuando las relaciones funcionales del sistema sólo permiten que el estado *varíe* en un conjunto finito (contable) de puntos temporales. Las causas instantáneas de los cambios de estados se denominan *eventos*.

Modelado de sistemas

Concepto de Modelo: Un modelo es una abstracción de la realidad que captura la esencia funcional del sistema, con el detalle suficiente como para que pueda utilizarse en la investigación y la experimentación en lugar del sistema real, con menos riesgo, tiempo y coste.

Un modelo es un conjunto formado por otros dos conjuntos:

Un conjunto de variables.

Un conjunto de relaciones entre las variables del modelo.

Las variables del modelo representan habitualmente magnitudes físicas del sistema que se modela; las relaciones describen su comportamiento ante una cierta clase de situaciones.

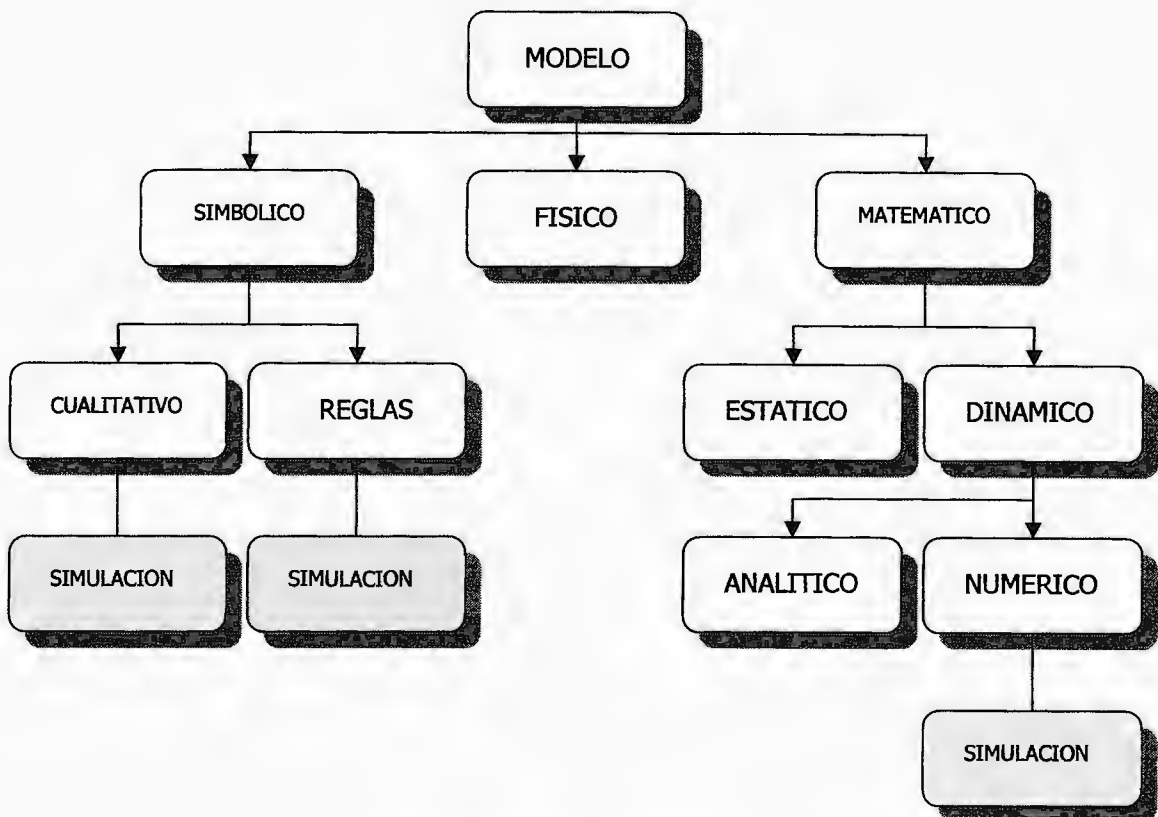
Es posible realizar distintos tipos de clasificaciones de modelos. Por ejemplo según la naturaleza del sistema o del uso que se vaya a dar al modelo. Un ejemplo de clasificación es la que califica a los modelos de:

- Físico.
- Mental.
- Gráfico.

- Matemático.
- Computerizado.

Tipos de modelos.

Existe una gran cantidad de técnicas de modelado, y, por ello es posible construir una gran cantidad de modelos para un sistema dado. En la siguiente figura se recoge en un pequeño diagrama algunas de las posibilidades de modelado más comunes:



Algunos Tipos de Modelos

El simulador por computadora

Un simulador por computadora está compuesto por las siguientes partes:

Un modelo: Es un modelo simbólico. Puede ser un conjunto de ecuaciones, reglas lógicas o un modelo estadístico.

El evaluador: Es el conjunto de procedimientos que procesarán el modelo para obtener los resultados de la simulación. Puede contener rutinas para la resolución de sistemas de ecuaciones, generadores de números aleatorios, rutinas estadísticas, etc.

La interfaz: Es la parte dedicada a interactuar con el usuario, recibe las acciones del mismo y presenta los resultados de la simulación en una forma adecuada. Esta unidad puede ser tan compleja como la cabina utilizada en los simuladores de vuelos profesionales.

Lenguajes de programación

Los lenguajes de alto nivel o lenguajes de propósito general tales como C, Fortran, Visual Basic, Cobol, Lisp, Algol, Pascal, etc. normalmente alejan al programador de las tareas de bajo nivel del computador y suelen ir apoyados en un conjunto de librerías que en el caso de la simulación facilitan mucho la tarea de modelar los sistemas y reducen normalmente el tiempo de ejecución del programa.

REDES

Modelo de Referencia OSI

El modelo de referencia OSI es un modelo de red dividido en capas y un esquema que se utiliza para entender como viaja la información a través de la red.

La división de la red se hace en siete capas con las siguientes ventajas:

Estandariza las interfaces

Reduce la complejidad

Facilita la técnica modular

Acelera la evolución

Asegura la interoperabilidad

Simplifica la enseñanza y el aprendizaje

El encapsulamiento se encarga de rodear los datos con la información de protocolo necesario antes que se una al transito de red.

Las siete capas son las siguientes:

Capa 7. Aplicación

Capa 6. Presentación

Capa 6. Sesión

Capa 4. Transporte

Capa 3. Red

Capa 2. Enlace de Datos

Capa 1. Física

Protocolos

Para que la comunicación en las redes pueda ser una realidad es necesario que todos los dispositivos implicados en la red hablen el mismo lenguaje para que puedan entenderse el uno con el otro a este lenguaje se le llama protocolo.

Un protocolo es una serie de reglas o normas que hacen que la comunicación en la redes sea más eficiente, técnicamente se dice que un protocolo es un conjunto de normas, o un acuerdo que determina la transmisión y formatos de los datos. La capa n de una computadora se comunica directamente con la misma capa n en la otra

computadora. Los acuerdos y normas que se utilizan en esta comunicación se llaman protocolo de capa n.

Entre los protocolos más utilizados para las redes se encuentra el TCP/IP.

TCP/IP es un conjunto de protocolos desarrollados para permitir la colaboración entre computadoras y así compartir recursos a través de una red.

TCP/IP es una suite de protocolos de Internet. TCP e IP son dos protocolos de la suite como los protocolos TCP e IP son los más comunes se usa el término TCP/IP para referirse a toda la familia. A continuación el gráfico del protocolo TCP/IP.

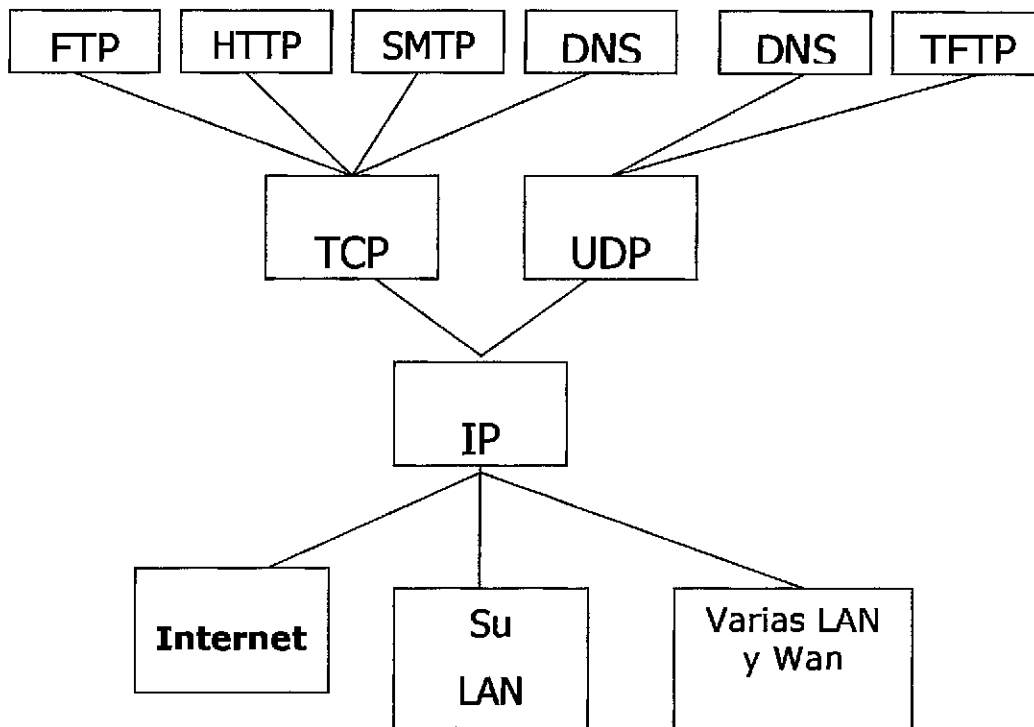


Gráfico de Protocolo TCP/IP.

Medios

Los medios son componentes de la capa física y por donde se transmite la información en forma de bits y bytes a través de una LAN. Los medios de red limitan las señales de red a un cable o fibra. Para escoger el tipo de medio que mejor convenga es necesario tomar en cuenta las ventajas y desventajas entre ellos, algunas de ellas son:

Costo
Facilidad de instalación
Longitud del cable
Cantidad total de computadoras

Acceso Múltiple con Detección de Portadora y Detección de Colisiones (CSMA/CD)

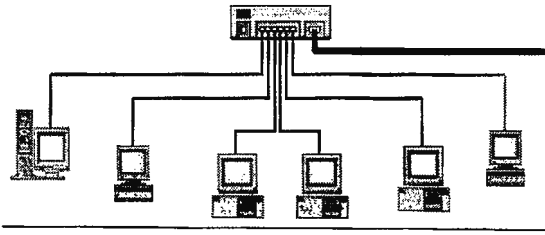
- En las LAN Ethernet se denomina red de acceso múltiple con detección de portadora y detección de colisiones porque solo se permite una transmisión a la vez, esto quiere decir que cuando se transmite desde un punto origen esta atraviesa toda la red, luego es recibida y examinada por cada nodo, los terminadores la absorben evitando que regrese al segmento.
- Cuando un hosts desea transmitir este censa el cable o medio para determinar si alguna otra estación esta transmitiendo, si esta libre transmite, pero cabe la posibilidad de que dos hosts en el mismo momento examinen el medio y determine que esta libre y empiecen a transmitir y este punto se produce una colisión.
- Cuando un hosts que esta transmitiendo reconoce que existe una colisión este envía una señal de congestión que hace que la colisión dure lo suficiente como para que los demás host se den cuenta. Los hosts transmisores dejan de enviar tramas durante un periodo de tiempo seleccionado al azar antes de retransmitir. Si los intentos subsecuentes producen colisiones, el hosts retransmite hasta quince veces antes dejarlo de hacer.

Hosts

Se le denomina hosts a las PCs, impresores y servidores entre otros, para este proyecto en particular, cuando se haga una referencia a host significara una PC.

Hub

- La función de un hub es regenerar y retemporizar las señales de red. Utilizando un proceso llamado concentración.



- El hub también es llamado repetidor multipuerto.
- El hub ofrece un punto de conexión central para los medios de cableado y aumentar la confiabilidad de la red.
- Los Hub se encuentran entre los dispositivos de capa 1 de Modelo OSI, ya que sólo regeneran la señal y la envían por medio de broadcast de ella a todos los puertos.

Switch

- El switch es un dispositivo de capa de enlace de datos del Modelo Osi. También es llamado puente multipuerto.
- La diferencia entre un Hub y un Switch es que los switches toman decisiones basándose en las direcciones MAC, utilizando una tabla de direcciones MAC para esto.
- Los switches hacen un proceso llamado *conmutando* datos solo entre los puertos donde están los hosts origen y destino, proporcionando ancho de banda total utilizando la microsegmentación. Esto hace que la LAN sea más eficiente.
- El propósito del switch es concentrar la conectividad, permitiendo que la transmisión de datos sea más eficiente.
- Los switches aprenden las direcciones MAC en sus tablas de la siguiente manera:
Aprenden el lugar de la estación examinando la dirección MAC origen y guardándola en la tabla correspondiente

Manda fuera a todos los puertos (excepto el puerto donde esta el hosts origen) cuando la dirección destino es una broadcast, un multicast o una dirección desconocida

Pasa cuando la dirección destino esta localizada en una interfaz diferente

Filtra cuando la dirección destino esta localizada en la misma interfaz

- Algunos beneficios al usar switches son:
 - Se reducen el numero de colisiones
 - Comunicación simultanea y múltiple
 - Enlaces de alta velocidad
 - Dominios de colisión libre
 - Mejora la repuesta de red
 - Incrementa la productividad del usuario

Router

- El router trabaja en la capa de red de modelo OSI. Esto permite que un router tome las decisiones basándose en grupos de direcciones de red (Direcciones IP) a diferencia de las direcciones MAC que son individuales. Los routers pueden conectar distintas tecnologías como por ejemplo: Ethernet, Token-Ring y FDDI. Como los routers enrutan paquetes estos se han convertido en el backbone de Internet, ejecutando el protocolo IP.

- Una de las funciones del router es determinar cual ruta debe utilizar cuando envía datagrama dado. Un frame es recibido en una interfaz del router. El frame de la capa de enlace de datos es quitado y descartado y el frame de la capa de red es enviado al proceso apropiado de la capa de red. Allí el encabezado del protocolo de red es examinado para determinar el destino del paquete. El proceso de la capa de red luego lo refiere a la tabla de enrutamiento. Luego la tabla indica cual interfaz conecta al próximo salto mas apropiado hacia el destino. El paquete es luego pasado de regreso a la capa de enlace de datos, donde es encapsulado en un nuevo frame y es encolado para ser entregado de la interfaz apropiada. Finalmente el frame es colocado en la red y procesado al siguiente salto donde el proceso se repite.

Las tablas de ruteo incluyen:

- Todas las rutas posibles que el router puede alcanzar
- Información en la cual lidera para un grupo de direcciones en particular

- Priorizar las conexiones a ser usadas
 - Reglas para manejar ambas rutinas y en casos especiales de tráfico
-
- Las tablas de ruteo son dinámicas, ellas se actualizan por protocolos de enrutamiento tal como Routing Information Protocol (RIP) o Protocolo de enrutamiento de gateway interior (IGRP) que constantemente se intercambian mensajes entre los routers. El router los consultan para determinar si existe o no una ruta a una dirección destino en particular.
 - Los routers se encargan regular el tráfico en las redes, permiten también que cualquier computador se pueda comunicar con otro computador en cualquier parte del mundo.

Suite de Protocolos TCP/IP

La función fundamental del protocolo TCP/IP es la transferencia de información desde un dispositivo a otro asemejándose a la función del modelo de referencia OSI en las primeras capas y soporta todos los protocolos físicos y de enlace de datos.

Direccionamiento IP

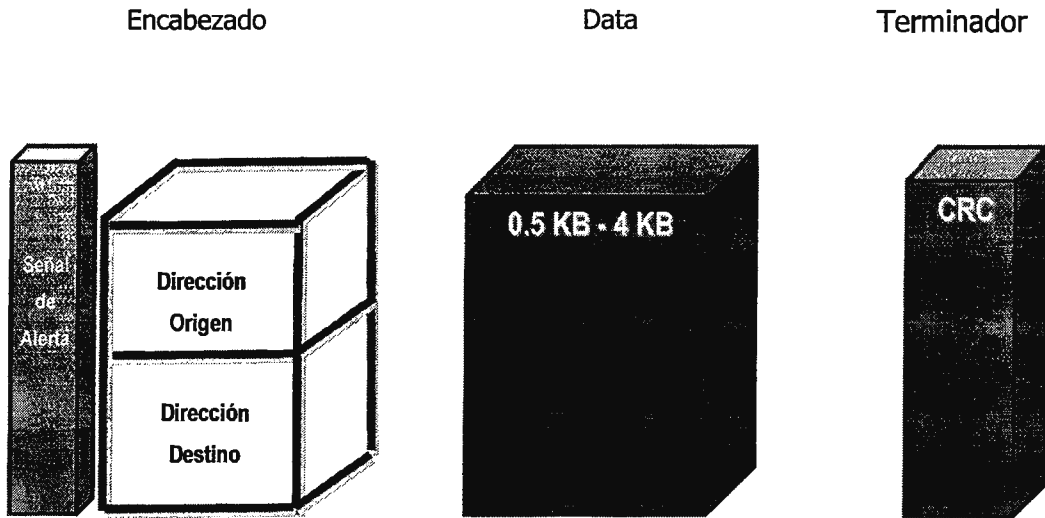
1. Direcciones de capa de red

El protocolo Internet (IP) es el protocolo que usa Internet y es la implementación más popular de un esquema de direccionamiento de red jerárquico.

En la capa de red, los datos se encapsulan en paquetes o datagramas. IP determina el encabezado del paquete IP incluyendo direccionamiento e información de control, pero no se ocupa de datos ya que los datos se ocupan las capas superiores.

Encabezado IP

Esta formado por:



2. Direcciones IP como numero binario de 32 bits

Una dirección IP se representa mediante un número binario (0 ó 1) de 32 bits.

Las direcciones IP se expresan como decimales. Se divide los 32 bits en cuatro octetos (grupo de 8 bits)

El numero de red identifica la red a la que se conecta un dispositivo y la otra parte identifica a la parte que identifica al un host específico dentro de esa red. Para su recordatorio rápido las direcciones están en notación decimal separadas por cuatro puntos que representan los cuatro octetos.

Los hosts de una red solo pueden comunicarse directamente con los dispositivos que tengan el mismo ID de red y pueden compartir el mismo segmento físico, pero si tienen diferentes ID de red entonces es necesario un router para poder conectar las dos redes.

3. Clases de dirección IP

Existen tres clases de direcciones IP:

Clase A

Se representa porque el primer BIT de la izquierda es siempre 0 y el primer octeto representa la parte de red y los tres octetos restantes son para host. En una dirección

de Clase A se pueden asignar hasta 2^8 direcciones (menos 2) o 16,777,214 direcciones IP posibles para los dispositivos que están conectados a la red.

Las direcciones van desde 1 hasta 127 en el primer octeto.

Clase B

Los primeros dos bits de la izquierda son siempre 10 (uno y cero) y los primeros dos octetos representan la parte de la dirección de red y los dos restantes la parte de hosts. En una dirección de Clase B se pueden asignar hasta 2^{16} menos dos o 65,534 direcciones IP posibles para los dispositivos conectados a la red.

Las direcciones van desde 128 hasta 191 en el primer octeto.

Clase C

Los primeros tres bits de la izquierda son siempre 110 (uno, uno y cero) y los primeros tres octetos representan a la dirección de red y el octeto restante a la dirección IP. En una dirección de Clase C se puede asignar hasta 2^{24} menos 2 o 254.

Las direcciones van desde 192 a 223

4. Subredes

Cuando las redes son muy grandes es bueno hacer subredes para que sea más flexible.

Las direcciones de subred incluyen la porción de cualquiera de las tres clases además de un campo de subred y un campo de host. El campo de Subred + Host se crea a partir del Host. Es decir se piden prestados bits a la parte de host para asignarlo como campo de subred.

La razón principal para usar una subred es reducir el tamaño del dominio de broadcast.

5. Mascara de subred

La mascara de subred formalmente llamada prefijo de red extendida le indica a los dispositivos de red cual es la parte de una dirección que corresponde a la parte de

red y la parte de host. La mascara de subred tiene una longitud de 32 bits y tiene 4 octetos.

Las mascaras de subred tienen todos unos en las posiciones de bit de red y tienen todos ceros en las posiciones de bit restantes asignándolas en la porción de host de una dirección.

Cuando se piden bits prestados al campo de host es necesario prestar por lo menos dos en este punto cabe mencionar que siempre existen dos subredes no utilizables / reservadas de subred y de broadcast.

Clase	Tamaño por defecto campo de host	Cantidad máx. de bits de subred	Mascara por defecto
A	24	22	255.0.0.0
B	16	14	255.255.0.0
C	8	6	255.255.255.0

6. Propósito de las direcciones IP

Comunicar computadoras entre si y con servidores esto es posible debido a que cada nodo que utiliza el protocolo TCP/IP tiene una dirección IP, esta es una dirección lógica exclusiva de 32 bits y se especifica en un formato decimal separado por puntos de 32 bits. Si un router enruta IP es necesario configurarle una dirección IP en cada interfaz.

Cada organización conectada a Internet tiene una dirección de red y los hosts que están dentro de la organización son parte de la misma dirección de red, pero con la diferencia que cada hosts se identifica por medio de una dirección exclusiva.

Enrutamiento IP

Explicación de Enrutamiento

Cuando un router toma un paquete y lo envía cerca de su destino final, se dice que el paquete ha sido enviado en términos sencillos, esto es lo que un router hace; envía paquetes hacia su destino y lo trata de hacer al menor costo posible. El costo para un router, no es medido en dólares o en centavos, pero sí en "Saltos". Cada vez que un paquete es ruteado a través de un router a otro, un número en el paquete de dato llamado *conteo de saltos* se incrementa a 1. Si el contador de saltos alcanza ciertos límites preestablecidos (Por ejemplo RIP permite un máximo de 16 saltos), el paquete podría ser descartado como no entregado desde los 16 saltos, los router no son capaces de entregar el paquete a la dirección destino.

Para los routers, el costo no es la variable absoluta en el mundo físico porque no es mucho más caro irse por "la ruta larga" desde un origen a un destino donde realiza una transacción con el Internet. Esto es porque:

- Los datos se mueven a la velocidad de la luz (o muy cerca de ella sobre los alambres de cobre), así ninguna distancia adicional no es muy relevante
- El Internet fue diseñado para ser redundante. Si primera ruta falla, prueba una segunda o tercera.

El diseño de Internet es sólido porque este reenruta paquetes de datos todo el tiempo como un resultado de eventos como desastres naturales y fallas en la energía.

Determinación de ruta

La determinación de ruta se realiza en la capa de red (capa 3) Esta función de determinación le permite al router evaluar todas las rutas disponibles hacia un destino y así establecer la mejor ruta posible para poder manejar los paquetes. Los servicios de enrutamiento usan la información de la topología de red al evaluar las rutas de una red. Esta información se puede recolectar dinámicamente por procesos de red o estáticamente configurada por un administrador de red.

La capa 3 o capa de red realiza la entrega de paquetes de máximo esfuerzo y de extremo a extremo a través de redes conectadas entre sí. La capa de red utiliza la

tabla de enrutamiento IP para enviar paquetes desde la red origen a la red destino. Cuando el router determina la mejor ruta para el destino del paquete, luego envía los paquetes por la interfaz de salida o el puerto donde este conectada la mejor ruta tomándolo desde la interfaz de entrada donde fue aceptado.

Enrutamiento de paquetes del origen al destino por parte de los routers.

Para que una red sea realmente práctica esta debe representar de manera coherente las rutas disponibles entre los routers. La dirección de red es el número que esta entre dos routers y proporciona información que un proceso de enrutamiento utiliza para transportar paquetes desde un origen a un destino. Por medio de estas direcciones, la capa de red una transmisión que interconecta redes aisladas.

La coherencia de las direcciones de red, mejora el uso de ancho de banda evitando broadcast innecesarios.

Al utilizar un direccionamiento de extremo a extremo coherente para representar la ruta de las conexiones. La capa de red es capaz de encontrar la mejor ruta sin generar tráfico innecesario de broadcast los dispositivos de red.

Selección de ruta y conmutación de paquetes

Selección de ruta

- El router usa la porción de red de la dirección para la selección de ruta para transferir el paquete al siguiente router y esta función permite al router seleccionar la mejor interfaz para enviar un paquete.
- El router final (conectado a la red destino) usa la porción de host de la dirección para entregar el paquete al host específico.

Conmutación de paquetes

Este proceso permite al router aceptar un paquete en una interfaz y enviarlo a través de una segunda interfaz.

Protocolo Enrutado vrs Protocolo de enrutamiento

Protocolo enrutado

Es aquel que proporciona suficiente información en su dirección de red para permitir que un paquete se envíe desde un host a otro tomando como base el esquema de direccionamiento. Los protocolos enrutados definen los formatos de campo dentro de un paquete. Los paquetes generalmente se transmiten de un sistema final a otro.

Protocolo de enrutamiento

Soportan los protocolos enrutados, ofreciendo mecanismos para compartir información de enrutamiento. Los protocolos de enrutamiento permiten a los routers intercambiar información para actualizar y mantener las tablas.

Protocolos de enrutamiento

- RIP (Routing Information Protocol o Protocolo de Información de enrutamiento)
- IGRP (Interior Gateway Routing Protocol o Protocolo de enrutamiento de gateway interior)

Protocolo de ruteo más antiguo y el más sencillo, es conocido como *RIP (Routing Information Protocol)*. RIP fue desarrollado por Xerox para su protocolo de red XNS. Luego fue adoptado por la comunidad Internet para la norma TCP/IP, donde ganó popularidad. Tanto fue así que varios otros protocolos posteriormente lo utilizaron y se adaptaron a sus necesidades.

Algoritmos de enrutamiento

Vector distancia

Este algoritmo determina la dirección (vector) y la distancia hacia cualquier enlace en la red

Para actualizar la información este algoritmo envía copias periódicas de una tabla completa de enrutamiento de un router a otro.

Estado - enlace

Este tipo de enrutamiento recrea la topología de la porción de red donde se localiza el router a este tipo de enrutamiento también se le denomina primero la ruta libre más corta.

Híbrido

Es la combinación de ciertos aspectos del algoritmo vector distancia y el estado – enlace.

RIP(Routing Information Protocol)

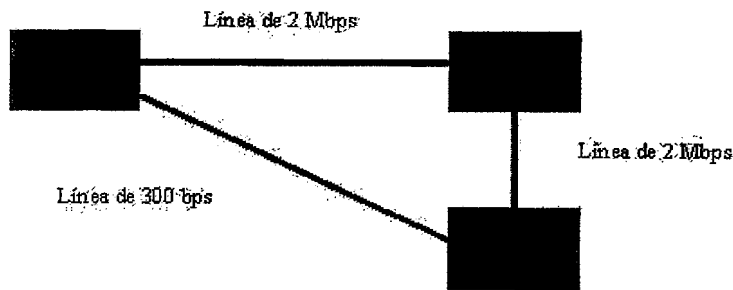
El protocolo de ruteo más antiguo, y por ende el más sencillo, es conocido como *RIP (Routing Information Protocol)*. RIP fue desarrollado por Xerox para su protocolo de red XNS. Luego fue adoptado por la comunidad Internet para la norma TCP/IP, donde ganó notoriedad. Tanto fue así que varios otros protocolos posteriormente lo utilizaron y se adaptaron a sus necesidades.

RIP elige el mejor camino en función de la cantidad de saltos o *hops* (es decir, la cantidad de redes atravesadas) entre la red origen y la red destino. Se denomina "métrica" a cualquier parámetro que sirva para establecer una comparación entre rutas.

- Es un protocolo broadcast para redes pequeñas y de mediano tamaño, el cual toma sus decisiones de enrutamiento o la mejor ruta basado en la información de la distancia que existe entre el host origen y el host destino. Posee un contador de saltos.
- RIP mantiene las tablas de enrutamiento y envía periódicamente anuncios a otros routers RIP informando acerca de cuales redes él puede alcanzar y cuales no.
- RIP tiene una limitación de 15 saltos o hops lo cual resulta un problema en redes de gran tamaño, además es incapaz de encontrar lazos de rutas y RIP anuncia el contenido de sus tablas de

enrutamiento a través de broadcast a nivel de MAC a todas las redes adjuntas cada 30 segundos, generando abundante tráfico.

- Este protocolo funcionaba bien mientras todos los enlaces WAN eran de velocidad similar, en cuyo caso se habla de una red homogénea. Sin embargo, al incorporarse vínculos WAN de diferente velocidad se producían casos como el presentado en la siguiente figura.

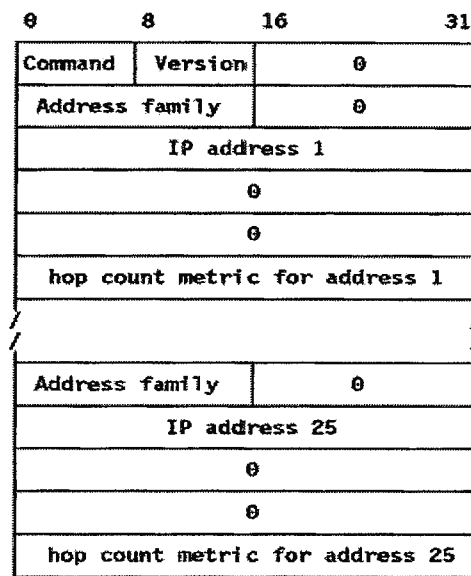


Utilizando el protocolo RIP, el router 1 le envía al router 3 todo el tráfico por el enlace de 300 bps (1 salto) en vez de emplear los dos enlaces de 2 Mbps (2 saltos, pero aproximadamente 10000 veces mas rápidos)

Hay dos versiones de RIP. La versión 1, llamada RIP a secas, es un protocolo muy extendido con un número de limitaciones conocidas. La versión 2 es una versión mejorada diseñada para aliviar estas limitaciones siendo al mismo tiempo muy compatible con su predecesor.

RIP versión 1

RIP opera en uno de dos modos: *activo* (normalmente usado por "routers") y *pasivo* (normalmente usado por hosts). Los mensajes RIP se envían en datagramas UDP y cada uno contiene hasta 25 pares de números como se muestra en la siguiente figura:



Mensaje RIP - En un mensaje RIP se pueden listar entre 1 y 25 rutas. Con 25 rutas el mensaje tiene 504 bytes ($25 \times 20 + 4$) que es el tamaño máximo que se puede transmitir en un datagrama UDP de 512 bytes.

- **Command:** Es 1 para una petición RIP o 2 para una respuesta.
- **Version:** Es 1.
- **Address Family:** Es 2 para direcciones IP.
- **IP address:** Es la dirección IP de para esta entrada de encaminamiento: un host o una subred (caso en el que el número de host es cero).
- **Hop count metric:** Es el número de saltos hasta el destino. La cuenta de saltos para una interfaz conectada directamente es de 1, y cada "router"

intermedio la incrementa en 1 hasta un máximo de 15, con 16 indicando que no existe ruta hasta el destino.

Tanto el modo activo como el pasivo escuchan todos los mensajes de broadcast y actualizan su tabla de encaminamiento

Operaciones básicas

- Cuando RIP se inicia envía un mensaje a cada uno de sus vecinos pidiendo una copia de la tabla de enrutamiento del vecino. Los routers vecinos devuelven una copia de sus tablas de encaminamiento.
- Cuando RIP está en modo activo envía toda o parte de su tabla de enrutamiento a todos los vecinos (por broadcast y/o con enlaces punto a punto). Esto se hace cada 30 segundos. La tabla de enrutamiento se envía como respuesta.
- Cuando RIP descubre que una métrica ha cambiado, la difunde por broadcast a los demás routers.
- Cuando RIP recibe una respuesta, el mensaje se valida y la tabla local se actualiza si es necesario.
- Para mejorar el rendimiento y la fiabilidad, RIP especifica que una vez que un "router" (o host) ha aprendido una ruta de otro, debe guardarla hasta que conozca una mejor (de costo estrictamente menor). Esto evita que los "routers" oscilen entre dos o más rutas de igual costo.
- Cuando RIP recibe una petición, distinta de la solicitud de su tabla, se devuelve como respuesta la métrica para cada entrada de dicha petición fijada al valor de la tabla local de enrutamiento. Si no existe ruta en la tabla local, se pone a 16.
- Las rutas que RIP aprende de otros routers expiran a menos que se vuelvan a difundir en 180 segundos (6 ciclos de broadcast). Cuando una ruta expira, su métrica se pone a infinito, la

invalidación de la ruta se difunde a los vecinos, y 60 segundos más tarde, se borra de la tabla.

Limitaciones

RIP no está diseñado para resolver cualquier posible problema de enrutamiento. Sin embargo, RIP está muy extendido y es probable que permanezca sin sustituir durante algún tiempo. Tiene las siguientes limitaciones:

- El costo máximo permitido en RIP es 16, que significa que la red es inalcanzable. De esta forma, RIP es inadecuado para redes grandes (es decir, aquellas en las que la cuenta de saltos puede aproximarse perfectamente a 16).
- RIP no soporta máscaras de subred de longitud variable (*variable subnetting*). En un mensaje RIP no hay ningún modo de especificar una máscara de subred asociada a una dirección IP.
- RIP carece de servicios para garantizar que las actualizaciones proceden de routers autorizados. Es un protocolo inseguro.
- RIP sólo usa métricas fijas para comparar rutas alternativas. No es apropiado para situaciones en las que las rutas necesitan elegirse basándose en parámetros de tiempo real tales como el retardo, la fiabilidad o la carga.
- El protocolo depende de la *cuenta hasta infinito* para resolver algunas situaciones inusuales. RIP especifica mecanismos para minimizar los problemas con la cuenta hasta infinito que permiten usarlo con dominios mayores, pero eventualmente su operatividad será nula. No existe un límite superior prefijado, pero a nivel práctico este depende de la frecuencia de cambios en la topología, los detalles de la topología de la red, y lo que se

considere como un intervalo máximo de tiempo para que la topología de enrutamiento se estabilice.

- Cada enrutador envía a sus vecinos su vector de distancias cada 30 segundos.
- Los mensajes RIP se encapsulan en datagramas UDP.
- En un mensaje RIP pueden enviarse hasta 25 entradas del vector de distancias.
- Para transportar vectores grandes se utilizan varios mensajes.
- Cuando un enrutador A recibe de un vecino B su vector de distancias, actualiza la entrada de su tabla de enrutamiento correspondiente a la red K si:

A no conocía a K

$$B_k < A_k + 1$$

A enruta por B hacia K y B_k cambió

- La actualización de la tabla de enrutamiento del enrutador A modifica el renglón correspondiente a la red K:
 - la nueva distancia es $B_k + 1$
 - el siguiente salto es B
- Una entrada de la tabla de enrutamiento se vuelve inválida si pasan 180 segundos sin que sea refrescada.
- Ejemplo de construcción de las tablas de enrutamiento.
- El destino se considera inalcanzable si la distancia es mayor a 15.
- Una solución complementaria para acelerar la convergencia se conoce con el nombre de *split horizon*.

– si S piensa que puede llegar a 10.0.0.0 vía R, sus mensajes a R no deben incluir a esta red.

- El esquema simple del ***split horizon*** omite las rutas aprendidas de un vecino en el vector que se le envía a ese vecino.
- El esquema con "**veneno en reversa**" utilizado en RIP incluye esas rutas en el vector pero establece la distancia a infinito.
- Para acelerar aún más la convergencia, un enrutador envía a todos sus vecinos su vector de distancias inmediatamente después de haber considerado inalcanzable un destino.
- Un enrutador también puede enviar su vector de distancias cuando cambia su distancia a un destino.

Algoritmo RIP

Actualización: A intervalos regulares cada router envía un mensaje de actualización describiendo su base de datos de enrutamiento a todos los otros routers que están directamente conectados.

El router envía estos mensajes cada 30 segundos, así que la red siempre tiene una información actualizada para rápidamente adaptarse a los cambios.

El mensaje de Actualización que un router envía a sus vecinos, no contiene rutas que ese vecino le ha proporcionado antes.

Propagación: Cuando un router X encuentra que un router Y tiene una ruta mas corta y mas rápida hacia un router Z, luego de actualizar su propia base de datos de enrutamiento para indicar el cambio.

Cualquier ruta más rápida es rápidamente propagada a los routers vecinos a través de un proceso de actualización, hasta que es propagado totalmente en la red RIP. Una descripción matemática de este algoritmo es la siguiente.

- Let $D(i, j)$ ser la métrica para la mejor ruta desde el router i a el router j .
- Let $d(i, j)$ representa la distancia desde el router i a el router j , fije a infinito si i y j son lo mismo o si i y j no son vecinos inmediatos.
- La mejor distancia es la siguiente

$$D(i, j) = 0, \text{ para todo } i$$

$$D(i, j) = \min (d(i, j) + D(k, j)), \text{ for } i \neq j, \text{ sobre todo } k$$

Envío de Registro: Se encarga de formar el datagrama que será enviado por la interfaz serial hacia un router vecino.

Expiración de Ruta: Proceso en el cual el router esta monitoreando si una ruta no ha sido actualizada después de 180 segundos, de ser así, la pone como invalida al darle el valor de 16 a la métrica de dicha ruta; después de eso sigue revisando si alguna ruta llega a los 240 segundos si actualizarse, de no hacerlo, es eliminada.

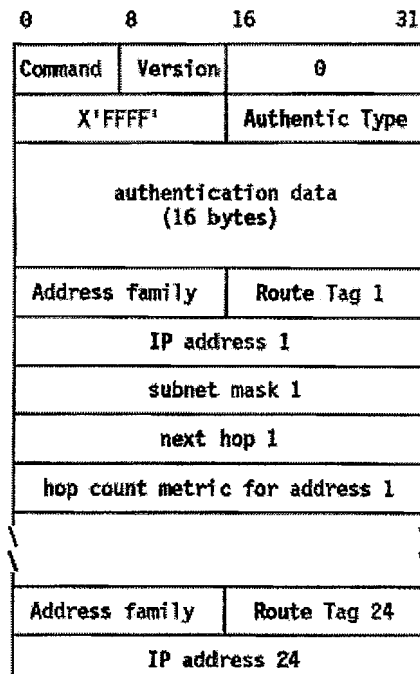
Inicio de Rip: Es el primer proceso que se llama acabo, al iniciar el router utilizando este protocolo, lee sus interfases Seriales y así guardar las direcciones IP que estas posean. También crea una datagrama con ID=2 el cual significa que es una solicitud de tablas de ruteo, la cual envía a los routers vecinos; y a su vez se inicializan los timers de propagación.

Escucha (Listener): Este proceso se encarga de saber cuando un mensaje a sido enviado a alguna interface del router, verifica que ID trae, para definir si es una solicitud de sus tablas de ruteo, un envío de actualización de otro router o si es una solicitud de ruteo.

RIP versión 2

RIP-2 extiende RIP-1. Es menos potente que otros IGP's recientes tales como OSPF de IS-IS, pero tiene las ventajas de una fácil implementación y menores factores de carga. La intención de RIP-2 es proporcionar una sustitución directa de RIP que se pueda usar en redes pequeñas y medianas, en presencia de subnetting variable o supernetting y, sobretodo, que pueda interoperar con RIP-1.

RIP-2 aprovecha que la mitad de los bytes de un mensaje RIP están reservados (deben ser cero) y que la especificación original estaba diseñada con las mejoras en la mente de los desarrolladores, particularmente en el uso del campo de versión. Un área notable en la que este no es el caso es la interpretación del campo de métrica. RIP-1 lo especifica con un valor de 0 a 16 almacenado en un campo de 4 *bytes*. Por compatibilidad, RIP-2 preserva esta definición, lo que significa en que interpreta 16 como infinito, y desperdicia la mayor parte del rango de este campo.



Mensaje RIP-2 - La primera entrada del mensaje puede ser una entrada de autenticación, como se muestra aquí, o una ruta como en el mensaje RIP. Si la primera entrada es de autenticación, sólo se pueden incluir 24 rutas en el mensaje; de otro modo, el máximo es 25, como en RIP.

Los campos del mensaje RIP-2 son los mismos que en RIP excepto los siguientes:

- **Version:** Es 2. Le dice al "router" RIP-1 que ignore los campos reservados, los que deben ser cero (si el valor es 1, los "routers" deben desechar los mensaje con valores distintos de cero en estos campos, ya que los originó un "router" que dice ser RIP, pero que envía mensajes que no cumplen el protocolo).
- **Address Family:** Puede ser X'FFFF' sólo en la primera entrada, indicando que se trata de una entrada de autenticación.
- **Authentication Type:** Define como se han de usar los restantes 16 bytes. Los únicos tipos definidos son 0, indicando ninguna autenticación, y 2 indicando que el campo contiene datos de password.

- **Authentication Data:** El password es de 16 bytes, texto ASCII plano, alineado a la izquierda y rellenado con caracteres nulos ASCII (X'00').
- **Route Tag:** Es un campo dirigido a la comunicación de información acerca del origen de la información de encaminamiento. Está diseñado para la interoperabilidad entre RIP y otros protocolos de encaminamiento. Las implementaciones de RIP-2 deben conservarlo, aunque RIP-2 no especifica como se debe usar.
- **Subnet Mask:** La máscara de subred asociada con la subred a la que se refiere esta entrada.
- **Next Hop:** Una recomendación acerca del siguiente salto que el "router" debería usar para enviar datagramas a la subred o al host dado en la entrada.

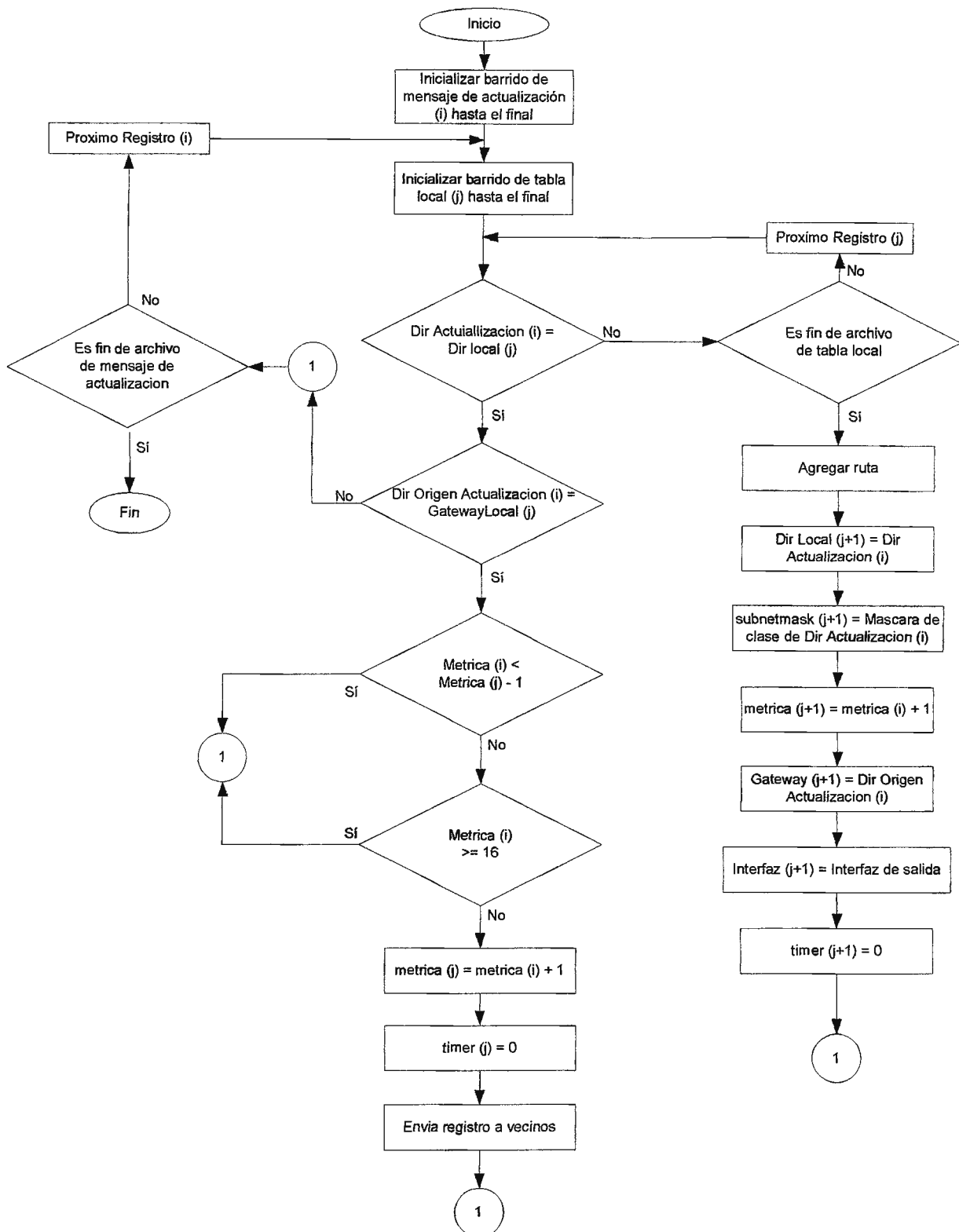
Limitaciones

- La información interna a una red nunca se debe anunciar a otra red.
- La información acerca de una subred más específica no se debe anunciar donde los "routers" vean una ruta de host.
- Las rutas a superredes (rutas con una máscara de subred más corta que la máscara natural de la red) no se deben anunciar en los sitios en los que puedan ser malentendidas por los "routers" RIP.

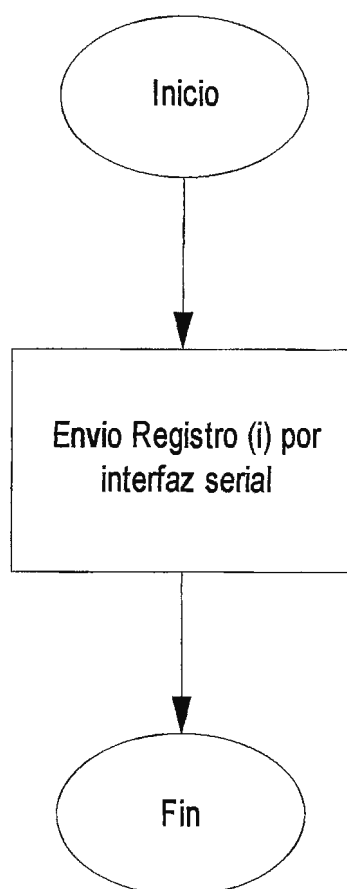
RIP-2 soporta además el multicast con preferencia al broadcast. Esto puede reducir la carga de los host que no están a la escucha de mensajes RIP-2. Esta opción es configurable para cada interfaz para asegurar un uso óptimo de los servicios RIP-2 cuando un "router" conecta redes mixtas RIP-1/RIP-2 con redes RIP-2. Similarmente, el uso de la autenticación en entornos mixtos se puede configurar para adecuarse a los requerimientos locales.

Flujogramas del protocolo RIP

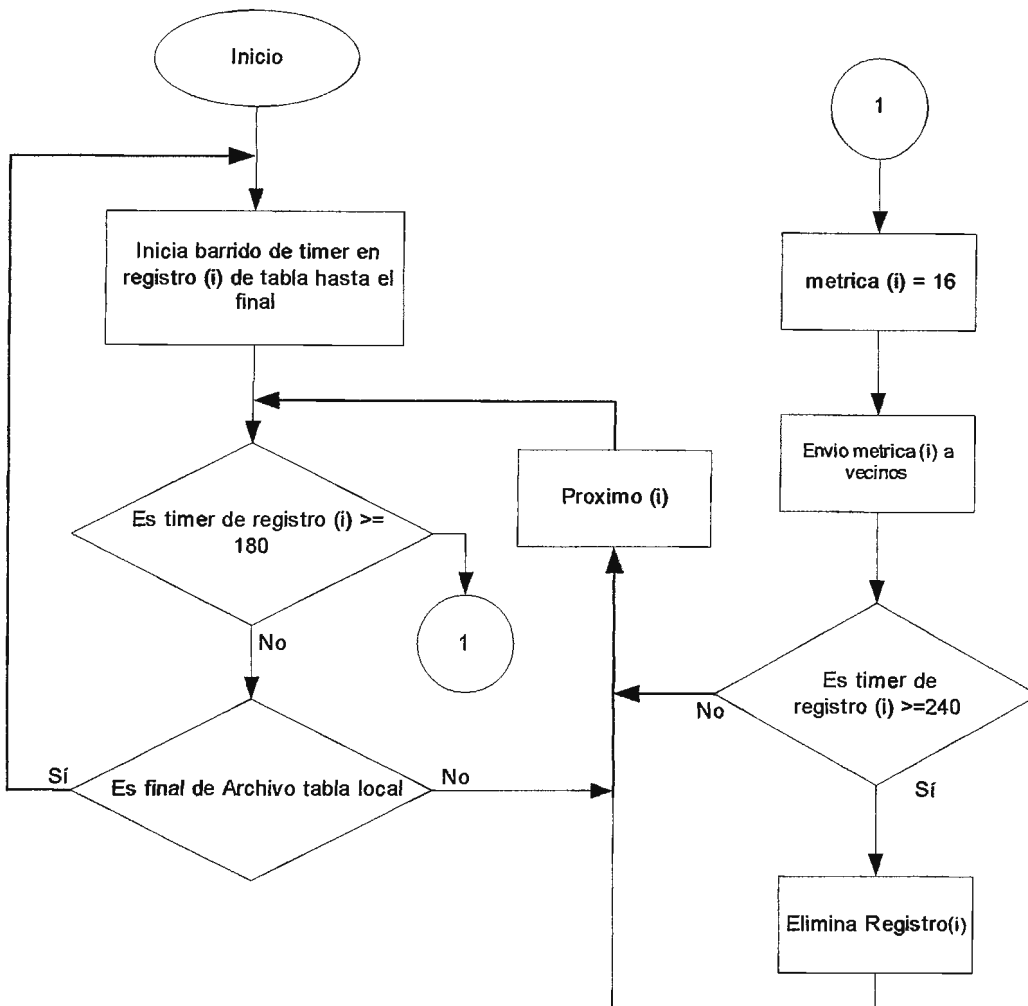
Proceso de Actualización

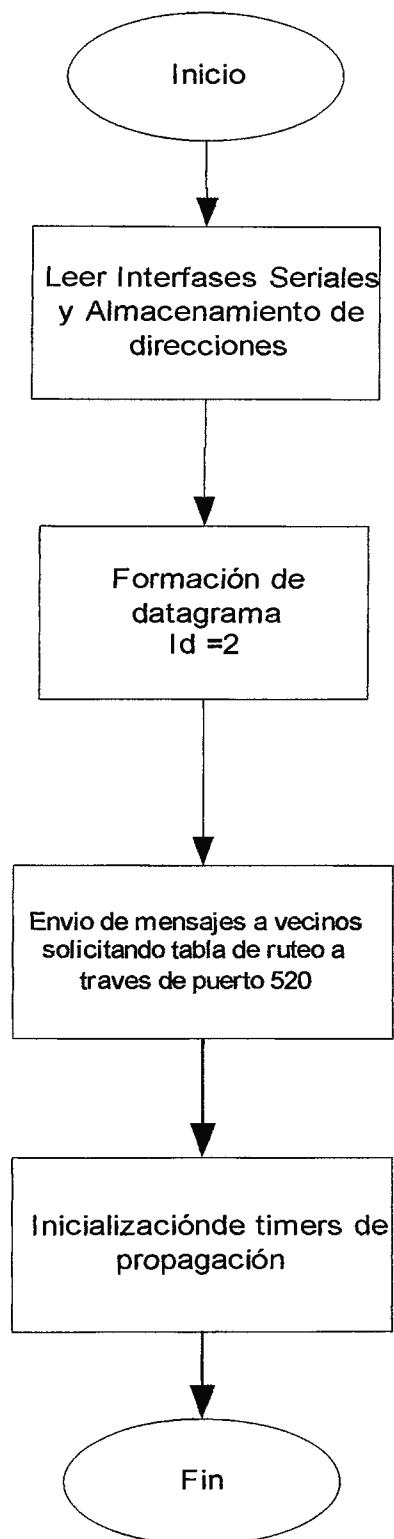


Proceso de Envío de Registro

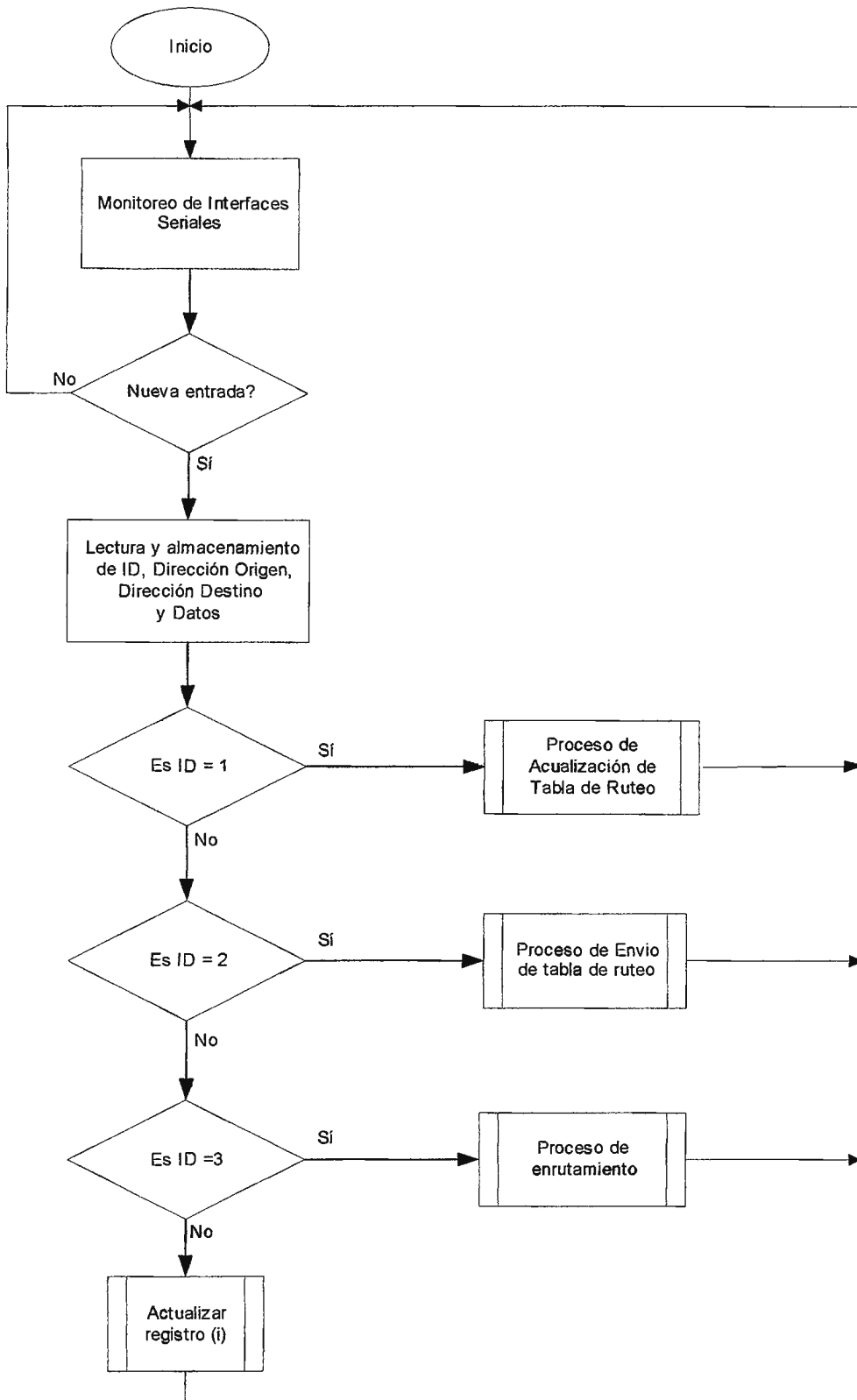


Proceso de Expiración de Ruta

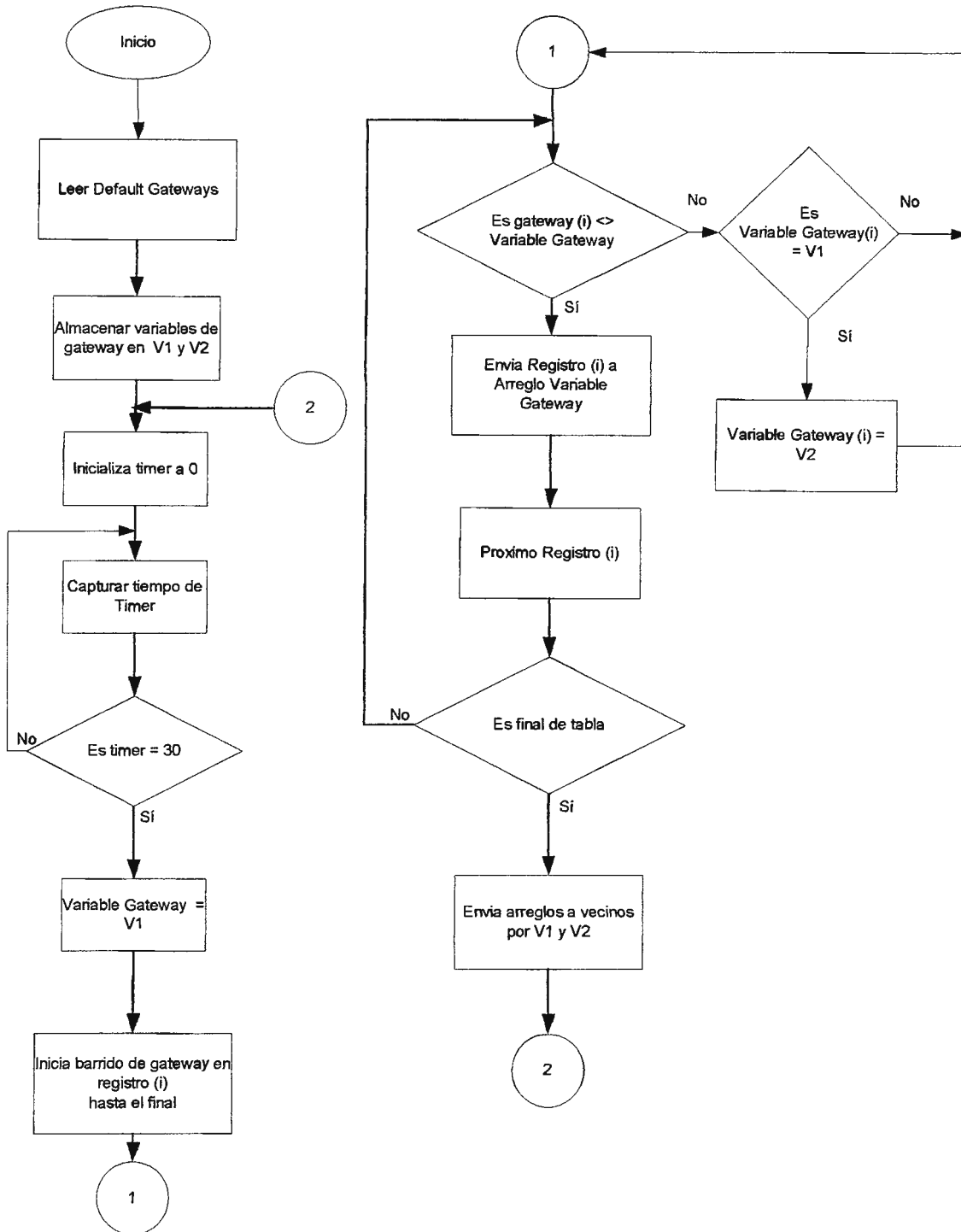


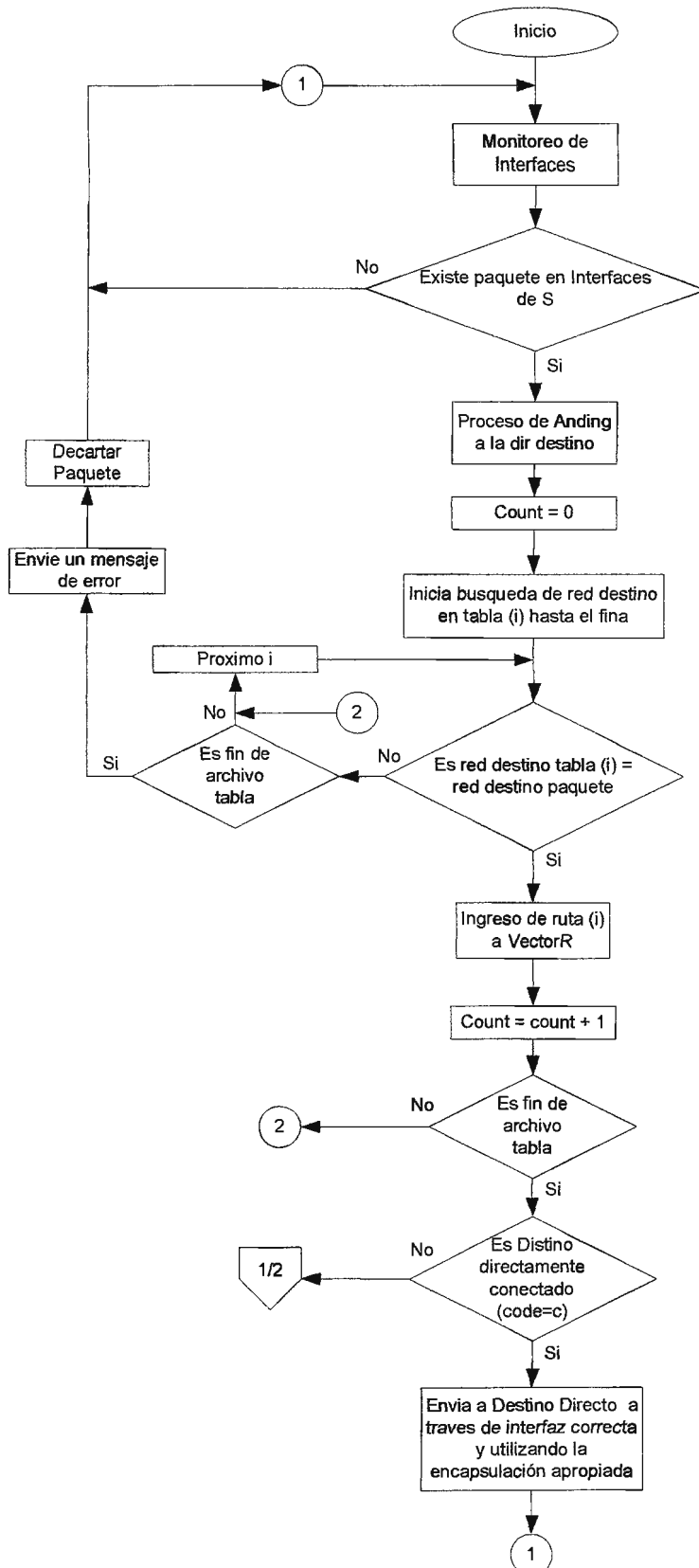
Proceso de Inicio de RIP

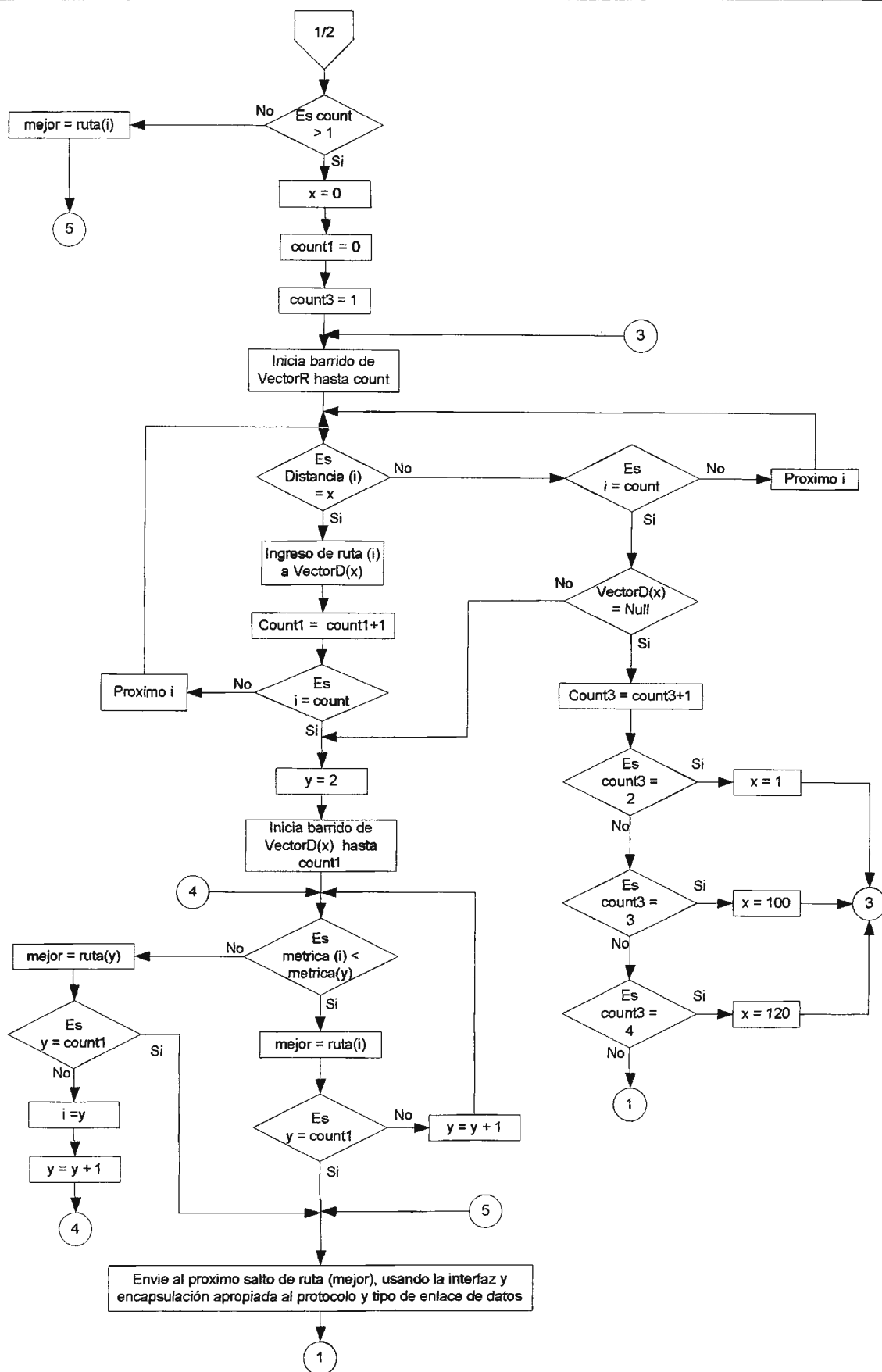
Proceso de Listener



Proceso de Propagación cada 30 segundos







IGRP(Interior Gateway Routing Protocol)

IGRP es un protocolo de ruteo interno utilizado en TCP/IP y OSI. La versión original de IP fue diseñada y desarrollada con éxito en 1986. Se considera un IGP (Interior Gateway Protocol) pero también ha sido utilizado como un protocolo de ruteo externo para el ruteo inter-domain. IGRP utiliza el algoritmo del vector distancia. El concepto es que cada router no necesita conocer todas las rutas/enlaces de la red entera. Cada router, informa acerca de los destinos y su distancia correspondiente. Cada router escuchando información, ajusta las distancias y las propaga a los routers vecinos. La información sobre la distancia en IGRP está representada como una combinación de ancho de banda disponible, retardo, carga y fiabilidad del enlace. Esto permite conseguir rutas óptimas.

Hay algunas áreas en las que la descripción de este protocolo puede esperarse que sea diferente de la implementación de Cisco. Estas áreas son:

- Cisco no implementa todavía múltiples tipos de servicio, no comprueba todo lo que depende de la cuenta de los saltos. De cualquier modo hacen mantenimiento y propagan la información necesaria.
- Cisco tiene un número de controles administrativos, permitiendo filtros y modificaciones de varias clases en la información de enrutamiento.
- Cisco proporciona varios caminos para definir las rutas por defecto.

OBJETIVOS DE IGRP

IGRP es un protocolo que asigna un número de routers para coordinar su enrutamiento. Sus metas son:

- Ruteo estable incluso en redes muy grandes y complejas. No deben producirse lazos, incluso si son transitorios.
- Rápida respuesta a cambios en la topología de la red.
- Mensajes con información para el chequeo de errores, IGRP no usa más ancho de banda que lo que necesita para su tarea.

- Reparte el tráfico entre rutas paralelas diferentes cuando éstas son en términos generales igual de buenas.
- Toma en cuenta la tasa de errores y el nivel de tráfico en diferentes caminos.
- La capacidad de manejar múltiples "tipos de servicio" con un conjunto simple de información.

La actual implementación de IGRP maneja ruteo para TCP/IP. De todos modos, el diseño básico está propuesto para ser capaz de manejar una variedad de protocolos. Durante los últimos años, el ruteo se ha convertido de repente en un problema más difícil al que solíamos tener. Hace pocos años, protocolos como RIP eran suficientes, pero el crecimiento de la Internet, y la descentralización del control de su estructura, ha resultado en un sistema de redes que está muy lejos de nuestra capacidad de manejarlo. IGRP es una herramienta propuesta para atacar este problema.

IGRP es un protocolo basado en el algoritmo del vector distancia. Los routers intercambian información de ruteo solo con sus routers vecinos. Esta información de ruteo contiene un resumen de información sobre el resto de la red. Cada router solo necesita resolver parte del problema, y solo tiene que recibir una porción de los datos totales.

La principal alternativa es una clase de algoritmos referidos a SPF (shortest-path first). Que están basados en la técnica de "flooding"(inundación), donde todo router debe mantener información del estado de toda interfase en todos los otros routers. Cada router independientemente resuelve el problema desde su punto de vista usando información de toda la red. En algunas circunstancias SPF puede ser capaz de responder a cambios más rápidamente. Para prevenir los lazos, IGRP tiene que ignorar nuevos datos durante unos pocos minutos después de fijar los cambios. Porque SPF tiene información directamente de cada uno de los routers, es posible evitar estos lazos en el ruteo. Puede actuar con la nueva información inmediatamente. De todos modos, SPF tiene más información que IGRP, tanto en las estructuras de datos internas y como en los mensajes que intercambian los routers. Las implementaciones de SPF tienen más mensajes de

verificaciones de errores que las implementaciones de IGRP, en otras cosas son iguales.

EL PROBLEMA DEL RUTEO

IGRP esta diseñado para usarse en routers que conectan distintas redes. Asumiendo que las redes usan la tecnología basada en paquetes. De hecho los routers actúan como conmutadores de paquetes. Cuando un equipo conectado a una red quiere enviar un paquete a otro equipo en una red diferente, dirige el paquete al router. Si el destino se encuentra en una de las redes conectadas al router, el router mandará el paquete al destino. Sino lo enviará a otro router que se encuentre cerca del destino. Los routers utilizan las tablas de ruteo para ayudarse a decidir qué hacer con el paquete.

La principal propuesta de IGRP es permitir a los routers construir y mantener las tablas de ruteo.

RESUMEN DE IGRP

IGRP es un protocolo que permite a los routers construir las tablas de ruteo a partir del intercambio de información con otros routers. Un router comienza con entradas en sus tablas para todas las redes que están directamente conectadas a él. En el caso más simple, el router encontrará una ruta que representa la mejor para llegar a cada red. Un camino se caracteriza por el próximo router al que deben ser enviados los paquetes, la interfase de red que debe utilizarse e información de la métrica. La métrica es un conjunto de números que determinan cuánto de buena es una ruta. Esto permite al router comparar rutas y elegir la mejor. Hay a menudo casos donde hace sentir que se reparte el tráfico entre 2 o más rutas. IGRP hará esto cuando 2 o más rutas sean igualmente buenas. El usuario puede configurarlo para repartir el tráfico cuando las rutas sean igualmente buenas.

La métrica utilizada por IGRP incluye:

- El retardo de la topología (topological delay time)

- El ancho de banda (ancho de banda del segmento, el más estrecho ancho de banda de la ruta)
- La ocupación de la línea (ocupación del canal en la ruta)
- La fiabilidad (confiabilidad de la ruta)

El retardo de la topología es la cantidad de tiempo que pasa hasta llegar al destino a través de la ruta, asumiendo una red no cargada. Desde luego hay un retardo adicional cuando la red está cargada.

De todos modos, la carga se mide por la ocupación del canal, no intentando medir el retraso actual.

El ancho de banda de la ruta es simplemente el ancho de banda en bits por segundo del enlace más lento de la ruta.

La ocupación del canal indica cuánto de este ancho de banda está actualmente en uso. Éste es medido y cambiará con la carga.

La fiabilidad indica la actual tasa de error. Es una fracción de los paquetes que llegan al destino sin error.

Aunque no son usadas como parte de la métrica, dos piezas de información adicionales son pasadas con ella: la cuenta de saltos y la MTU (Maximun Transfer Unit). El contador de saltos es simplemente el número de routers que el paquete debe atravesar para llegar al destino deseado. Y la MTU es el máximo tamaño de paquete que puede ser enviado a lo largo de todo el trayecto sin fragmentación. Es la mínima de las MTUs de todas las redes incluidas en la ruta al destino.

Basado en la información de la métrica, una simple "métrica compuesta" es calculada para la ruta. Esta métrica compuesta combina el efecto de varios componentes métricos en un número simple que representa lo buena que es la ruta. Esta métrica se usa para decidir la mejor ruta.

Cuando un router es por primera vez encendido, su tabla de Ruteo es inicializada. Esto debe ser hecho por un operador desde un terminal, o bien leyendo la información desde los archivos de configuración. Se proporciona una descripción de

cada red conectada al router, incluyendo el retraso a través del enlace (cuánto le cuesta a un bit atravesar el enlace) y el ancho de banda del enlace.

Periódicamente cada router emite por broadcast su tabla entera de ruteo a los routers vecinos cada 90 segs.

Cuando un router recibe esta información de otro router, compara la tabla con la suya. Cualquier nuevo destino o ruta es añadida a la tabla de Ruteo del router. Las rutas en el broadcast son comparadas con las rutas existentes. Si una nueva ruta es mejor, reemplazará la que tenía por la nueva. La información en el broadcast es también utilizada para actualizar la ocupación del canal y otra información sobre las rutas existentes.

El proceso básico de construcción de las tablas de ruteo por intercambio de información con los vecinos es descrito por el algoritmo de Bellman - Ford.

En IGRP, el algoritmo general de Bellman-Ford es modificado en tres aspectos críticos:

1.- En lugar de una métrica simple, un vector de métricas es utilizado para caracterizar la ruta. Una simple métrica compuesta puede ser calculada a partir de este vector de acuerdo con la ecuación 1. El uso de un vector permite al router acomodar diferentes tipos de servicio utilizando coeficientes distintos en la ecu.1.

2.- En lugar de escoger la ruta con la métrica más pequeña, el tráfico es repartido entre diferentes rutas, cuyas métricas caen dentro de un determinado rango. Esto permite distintas rutas para ser utilizadas en paralelo, proporcionando un ancho de banda efectivo mayor que con una solo ruta. Una varianza V es especificada por el administrador de red. Todas las rutas con métrica mínima se mantienen. También, todas las rutas cuya métrica es menor que $V \times M$ se mantienen. El tráfico es distribuido a través de múltiples rutas en una proporción inversa a las métricas compuestas.

3.- Diferentes características son introducidas para proporcionar estabilidad en situaciones donde la topología está cambiando. Estas características han sido propuestas para prevenir lazos en la topología y el problema de la cuenta a infinito. Las principales características de estabilidad son: "holddowns", "triggered updates", "split horizon", and "poisoning".

El reparto de tráfico (punto 2.) tiene un peligro. La varianza V está designada para permitir al router usar rutas paralelas de diferente velocidad. Si la varianza es 1, solo la mejor ruta será usada. Subiendo la varianza podemos permitir al tráfico ser repartido entre la mejor ruta y otras rutas que están cerca de ser tan buena como la mejor. Pero existe el peligro de que con una varianza suficiente grande, rutas que no solo son más lentas sino que actualmente van en la dirección equivocada, se vuelvan válidas. No se envía tráfico a través de caminos cuya métrica remota (la métrica calculada en el siguiente salto) sea mayor que la métrica calculada en el router. En general, los administradores de sistema han llegado al acuerdo de utilizar una varianza de valor 1, excepto en situaciones específicas donde se necesita usar rutas paralelas.

La mejor ruta es elegida según una métrica compuesta (composite metric) descrita a continuación:

$$[(K1 / Be) + (K2 * Dc)]^r \quad \text{ecuación 1}$$

Donde:

$K1, K2$: constantes \rightarrow indican el peso asignado al ancho de banda y al delay.

Dependerán del "tipo de servicio"

Be : ancho de banda efectivo. Ancho de banda cuando la red no está cargada $\times (1 - \text{ocupación del canal})$

Dc : delay

r : (confiabilidad) fiabilidad \rightarrow % de transmisiones que son recibidas con éxito en el siguiente salto

En principio, Dc (composite delay), puede ser definido como:

$$Dc = Ds + Dcir + Dt$$

Donde:

Ds = retardo de conmutado

$Dcir$ = delay del circuito (retardo de propagación de 1 bit)

D_t =retardo de transmisión

La ruta que minimice esta métrica será la mejor.

Cuando existe más de una ruta para un mismo destino, el router puede enrutar los paquetes por más de una ruta.

Se dan 2 ventajas por utilizar un vector de información métrica:

1.- Proporciona capacidad de soportar múltiples "tipos de servicio" desde el mismo conjunto de datos.

2.- Precisión

Cuando se utiliza una métrica simple, normalmente se trata como si fuera un delay. Cada enlace en el camino es añadido a la métrica total. Si hay un enlace con un bajo ancho de banda, normalmente se representa por un gran delay.

IGRP proporciona un sistema para la interconexión de redes de calculadoras que pueden de forma estable manejar un grafo de la topología incluyendo lazos. El sistema mantiene mucha información métrica de rutas, o sea, conoce los parámetros de ruta de todas las otras redes a las cuales algún router está conectado. El tráfico puede ser distribuido sobre caminos paralelos y múltiples parámetros del camino pueden ser simultáneamente calculados sobre la red entera.

IGRP está definido para manejar múltiples tipos de servicio y múltiples protocolos. El "tipo de servicio" es una especificación en un paquete de datos que modifica las rutas a ser evaluadas. Por ejemplo, en TCP/IP el protocolo permite al paquete especificar la importancia relativa de un gran ancho de banda, bajo retardo, o alta fiabilidad. Generalmente, las aplicaciones interactivas especificarán un bajo retardo y las aplicaciones de transferencia especificarán un gran ancho de banda. Estos requerimientos determinan los valores de K_1 y K_2 que son utilizados en la ecuación 1. Cada combinación de especificaciones en el paquete que va a ser soportada se refiere a un "tipo de servicio". Para cada tipo de servicio, un conjunto de parámetros

K1 y K2 puede ser elegido. Una tabla de ruteo es mantenida para cada tipo de servicio. Esto se hace porque las rutas son elegidas y ordenadas de acuerdo con la métrica compuesta definida por la ecuación 1. Esto es diferente para cada tipo de servicio. La información procedente de todas estas tablas de ruteo es combinada para producir mensajes de actualización de la información de ruteo que son intercambiados por los routers.

EJEMPLO:

El router S está conectado a las redes 2 y 3 mediante las interfases correspondientes. Inicialmente, el router S solo sabe que puede alcanzar cualquier destino en las redes 2 y 3. Todos los routers son programados para periódicamente transmitir a sus vecinos tanto su propia información como la recogida de otros routers.

El router S recibiría actualizaciones de los routers R y T, y aprenderá que puede alcanzar equipos en la red 1 a través del router R y equipos de la red 4 a través del router T. Cuando el router S envía su tabla entera de ruteo, en el próximo ciclo, el router T aprenderá que puede llegar a la red 1 a través del router S.

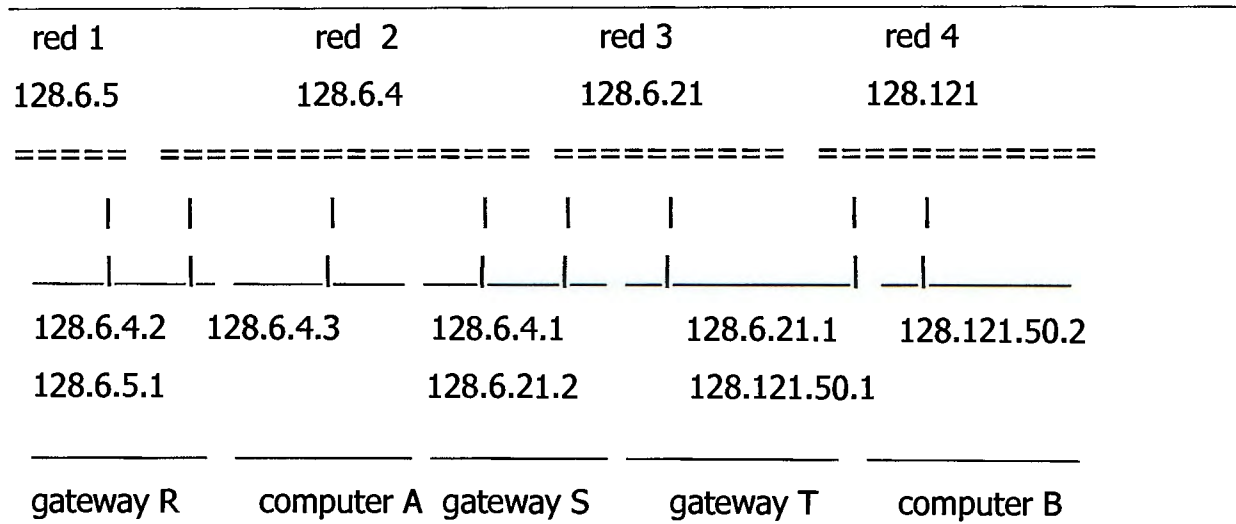


Fig 1. Un ejemplo de red

Cada router calcula una métrica compuesta para determinar lo buenos que son los caminos a un destino.

En la siguiente figura (2), para un destino en la red 6, el router A calcularía las funciones métricas para 2 caminos, vía el router B y C. Notar que los caminos son definidos solamente por el próximo salto. Hay 3 posibles rutas desde A hasta red 6:

- Directamente a B
- a C y luego a B
- a C y luego a D

El router A necesita elegir entre las 2 rutas que incluyen a C. La tabla de ruteo en A tiene una única entrada representando el camino a C. Su métrica representa el mejor camino para alcanzar el destino a través de C. Si A envía un paquete a C, éste decidirá si usa B o D.

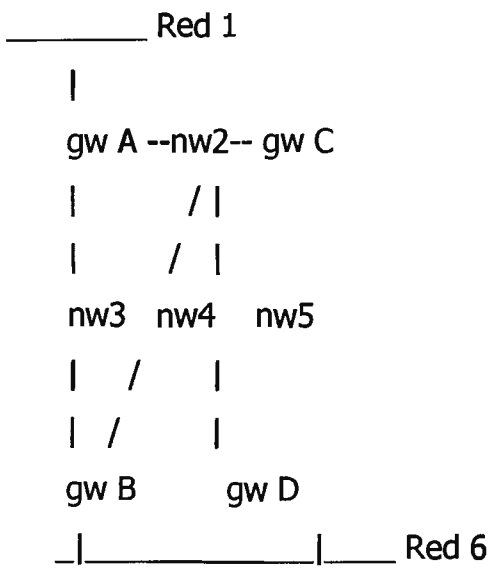


Fig 2. Ejemplo de caminos alternativos

Aquí, mostramos un ejemplo de cómo podría ser la tabla de ruteo de S. Notar que los componentes individuales del vector métrico no son mostrados, por simplicidad. Para construir la tabla a partir de la información de los vecinos se utiliza, como ya dijimos, el algoritmo de Bellman-Ford.

	interfaz	next gateway	metric
	-----	-----	-----
Red 1	nw 1	ninguna	directamente conectada
Red 2	NW 2	ninguna	directamente conectada
Red 3	NW 3	ninguna	directamente conectada
Red 4	NW 2	C	1270
	NW 3	B	1180
Red 5	NW 2	C	1270
	NW 3	B	2130
Red 6	NW 2	C	2040
	NW 3	B	1180

Fig 3. Un ejemplo de tabla de ruteo

CARACTERÍSTICAS DE LA ESTABILIDAD

Estas características son definidas para prevenir a los routers elegir rutas erróneas. Como se describe en el RFC 1058, esto puede ocurrir cuando una ruta no puede ser utilizada por un fallo en un router o en la red. En principio, son los routers vecinos los que detectan el fallo. Entonces éstos envían las actualizaciones necesarias que muestran la vieja ruta como inutilizada. Sin embargo, es posible que las actualizaciones no alcancen algunas partes de la totalidad de la red, o que sean retrasadas en determinadas routers. Un router que todavía piensa que la vieja ruta es buena puede continuar propagando esta información, tal que vuelve a entrar la ruta fallida en el sistema. Finalmente, esta información será propagada a través de la red y volverá al router que la reinyectó. El resultado es una ruta circular.

De hecho, hay cierta redundancia en los contadores de medida. En principio, holddowns y triggered updates deberían ser suficientes para prevenir las rutas erróneas. Sin embargo, en la práctica, los fallos de comunicaciones de varios tipos pueden hacer que éstos sean insuficientes. Split horizon y route poisoning han sido ideadas para evitar los lazos en cualquier caso.

Normalmente, las nuevas tablas de rutas son enviadas a los routers vecinos regularmente (cada 90 seg. Por defecto, aunque puede ser fijado por el administrador del sistema).

Un ***triggered update*** es una nueva tabla de ruteo que es enviada inmediatamente, en respuesta a algún cambio. El cambio más importante es el de una ruta. Esto puede ocurrir porque un timeout ha expirado (probablemente un router vecino o una línea ha caído), o porque un mensaje de actualización del siguiente router en el camino muestra que el camino no es el más largo de los utilizables. Cuando un router G detecta que una ruta no es la más larga, dispara una actualización inmediatamente. Esta actualización mostrará la ruta como inutilizada. Considerando lo que ocurre cuando esta actualización llega a los routers vecinos, si la ruta del

vecino apunta a G, el vecino de borrar la ruta. Esto hace que el vecino dispare una actualización, etc.

Así, un fallo disparará una ola (serie) de mensajes de actualización, que se propagarán a través de la parte de la red en cuyas rutas había un fallo en un router o en una red.

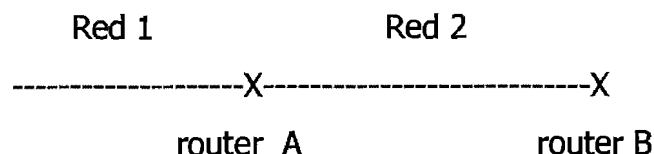
Triggered updates serían suficientes si pudiéramos garantizar que la ola de actualizaciones alcanzase todo router implicado inmediatamente. Sin embargo, hay dos problemas:

- 1.- los paquetes que contienen el mensaje de actualización pueden ser descartados o corrompidos por algún enlace en la red
- 2.- los triggered updates no ocurren instantáneamente

Los **Holddowns** se han definido para solucionar estos problemas. Cuando una ruta es eliminada, no se aceptará una nueva ruta para el mismo destino hasta pasado un tiempo. Esto da tiempo a los triggered updates para llegar al resto de routers, así estamos seguros de que cualquier nueva ruta que tengamos no sea algún router reinsertando al viejo. El período del holddown debe ser suficientemente largo para permitir la ola de triggered updates a través de la red. Además suele incluir, una pareja de ciclos regulares broadcast, para manejar los paquetes descartados. Consideremos que pasa si, un triggered update es descartado o corrompido, el router que emitió esa actualización emitirá otra actualización en la próxima actualización. Esto empezará la ola de triggered updates en los vecinos que no dieron con la ola inicial.

La combinación de triggered updates y holddowns debería ser suficiente para eliminar las rutas inutilizadas y prevenir que sean reinsertadas. Sin embargo, algunas precauciones adicionales se han desarrollado, éstas son el split horizon y el route poisoning.

Split horizon aparece de la observación de que nunca siente que envía una ruta de vuelta en la dirección de donde viene. Consideremos la siguiente situación:



El router A le dirá a B que tiene una ruta a la red1. Cuando B envíe actualizaciones a A, nunca se hará mención a la red1. Como A está más cerca de red1, no se debe considerar ir vía B. La regla del Split horizon dice que un mensaje de actualización independiente debe ser generado para cada vecino. La actualización para un vecino dado debería omitir las rutas que apunten a este vecino. Esto previene lazos entre routers vecinos. Supongamos que la interfase de A con la red1 falla, sin la regla de split horizon, B diría a A que puede llegar a red1. Ya no tiene una ruta real, A podría escoger esta ruta. En este caso, ambos A y B tienen ambas rutas a 1. Pero A apuntaría a B y B apuntaría a A. Desde luego, los triggered updates y holddowns deberían prevenir que esto sucediera. Pero no hay motivo para mandar información de vuelta al lugar del que vino. Además de su papel en la prevención de lazos, el split horizon mantiene bajo el tamaño de los mensajes de actualización.

Split horizon debería evitar los lazos con routers vecinos.

Route poisoning es propuesto para romper lazos más grandes. Su papel es que cuando una actualización muestre la métrica para una ruta existente y tenga que aumentar suficientemente, hay un lazo. La ruta debería ser eliminada y aplicar holddown. Actualmente la regla es, una ruta se elimina si la métrica compuesta aumenta más de un factor de 1.1.

Deshabilitar Holddowns

Cisco proporciona la opción de deshabilitar los holddowns. La desventaja de los holddowns es que éstos retrasan la adopción de la nueva ruta cuando la vieja ruta ha

fallado. Con parámetros por defecto, esto puede tardar varios minutos antes de que el router adopte la nueva ruta después del cambio. Sin embargo, por las razones antes explicadas, no es seguro evitar los holddowns. El resultado sería una cuenta a infinito. Se presupone, pero no se puede asegurar, que con una versión más fuerte de route poisoning, los holddowns no son necesarios para evitar la cuenta a infinito, y se podrían deshabilitar.

Esta implementación más fuerte del route poisoning se basa en un contador de saltos. Si el contador de saltos para un camino aumenta, la ruta se elimina. Eliminará obviamente las rutas que son todavía válidas. Si algo en la red cambia, como que el camino vaya ahora por un router más, el contador de saltos aumentará. En este caso, la ruta es todavía válida. Sin embargo, no hay una forma completamente segura de distinguir este caso de los lazos (cuenta a infinito). La aproximación más segura es eliminar la ruta siempre que el contador de saltos aumente. Si la ruta es todavía válida, será reinstalada en la próxima actualización, y esto causará un triggered update que reinstalará la ruta en el sistema.

En general, los algoritmos de vector distancia adoptan nuevas rutas fácilmente. El problema es solamente eliminar las viejas del sistema.

PROCESO DE ACTUALIZACIÓN

Un router puede procesar datos de diferentes protocolos. Cada protocolo tiene diferentes estructuras de direccionamiento y de formato de paquete, por lo que la implementación para cada uno será también diferente. La principal diferencia de un protocolo a otro será el formato del paquete de actualización de ruteo que debe ser diseñado para ser compatible con el protocolo utilizado.

La definición de destino puede variar de un protocolo a otro. El método que describiremos aquí puede ser usado para enrutar a host individuales, redes o a esquemas de direcciones jerárquicos más complejos. El tipo de ruteo utilizado dependerá de la estructura de direcciones del protocolo. La actual implementación de

TCP/IP soporta solamente ruteo a redes IP. Así, el destino significa una red IP o un número de subred.

Al inicio del código del programa, será necesario definir los protocolos aceptables y los parámetros que describen cada interfase.

Los datos de entrada son:

- Redes a las que el router está conectado
- Descarga de ancho de banda de cada red
- Retardo topológico
- Confiabilidad
- Canal ocupado de cada red
- MTU de cada red

Los tres primeros elementos son más o menos permanentes. No dependen de la carga. Pueden ser establecidos desde un archivo de configuración o directamente como entrada. Hay que notar que IGRP no usa un retraso (delay) medido, pues resultaría muy difícil mantener un ruteo estable. Hay dos parámetros de medida, confiabilidad (basada en la tasa de error reportada por la interfase de red hardware o firmware) y channel occupancy.

El algoritmo de Ruteo también requiere el valor de varios parámetros: valor de los timers, varianza (en Cisco siempre vale 1), holddowns (sí están o no habilitados)

Una vez la información inicial ha sido introducida, las operaciones en el router se disparan por eventos, llegada de un paquete de datos por una interfase, o expiración de un timer.

- Cuando un paquete llega, se procesa según la siguiente figura 1 (las figuras aparecen después).
- Cuando un paquete es aceptado por un router, es analizado según el protocolo específico.

Si es un paquete de actualización de ruteo se procesa según la figura 2.

- Muestra los eventos disparados por el timer. El timer está puesto tal que genere una interrupción por segundo. Cuando se produce una interrupción, se ejecuta la figura 3.
- Subrutina de actualización del ruteo. Figura 3.
- Muestra los detalles de calculación de la métrica, figura 4.

Hay 4 constantes de tiempo críticas que controlan la propagación de la ruta y la expiración. Estas constantes las fija el administrador del sistema. Son:

- *broadcast time*: las actualizaciones son broadcast por todos los routers en todas las interfases conectadas.
- *invalid time*: si no se ha recibido la actualización para un camino dado durante este tiempo, se produce timeout.
- *Hold time*: cuando un destino se ha vuelto desconocido (o la métrica ha aumentado suficiente para causar poisoning) el destino va a holddowns. Durante este estado, ningún nuevo camino será aceptado para el mismo destino durante este período de tiempo.
- *Flush time*: si no se ha recibido la actualización para un camino dado durante este tiempo, la entrada para él, se elimina de la tabla de ruteo.

Hay que notar que un mensaje de actualización de ruteo IGRP (actualiza mensajes) tiene tres partes:

- Interior: para rutas a subredes.
- Sistema (que significa, este "sistema autónomo"), y no interior.
- Exterior.

La sección interior, no incluye toda la información de la subred. Solo las subredes de una red son incluidas. Esto es, la red asociada con las direcciones a las cuales la actualización está siendo enviada. Normalmente las actualizaciones son broadcast en cada interfase. Las redes más grandes (no subredes) se colocan en la sección de

sistema del mensaje de actualización, a menos que éstas sean marcadas como exteriores.

Una red será marcada como exterior si fue instruida por otro router y la información le llega de la sección exterior del mensaje de actualización. La implementación de Cisco permite al administrador del sistema declarar determinadas redes como exteriores. Las rutas exteriores son candidatas por defecto. Hay rutas que van a, o a través de routers que son considerados apropiados por defecto, para ser usados cuando no hay una ruta explícita para un determinado destino. La implementación de Cisco elige una ruta por defecto seleccionando la ruta exterior con la métrica menor.

FIGURAS

1.- RUTEO DE PAQUETES:

Este proceso consta de una serie de pasos:

A.- este proceso utiliza la lista de protocolos soportados y la información sobre las interfases registradas cuando el router es inicializado. Los detalles de proceso de los paquetes dependen del protocolo utilizado por el paquete, asimismo el contenido del paquete está especificado en el protocolo.

B.y C.- Las especificaciones del protocolo incluyen procedimientos para determinar destino del paquete, compara el destino con las direcciones propias del router o determina si éste es broadcast.

D.- Prueba de búsqueda de los destinos listados en la tabla de ruteo. Esta prueba se satisface si hay una entrada en la tabla de ruteo para ese destino dado, y el destino lleva asociado al menos una ruta válida.

E.- Calcula la ruta a utilizar. Las rutas cuya métrica compuesta remota no es menor que su métrica, no son consideradas. Si más de un camino es aceptable, estos caminos serán utilizados alternativamente, round-robin. La frecuencia con la que un camino es usado es inversamente proporcional a su métrica.

Procesando los paquetes entrantes (Processing incoming packets)

Llegada de paquetes de datos usando la interfase I.

A. Determinar el protocolo usado por el paquete

SI el protocolo no es soportado ENTONCES descartar el paquete

B. SI la dirección destino concuerda con cualquiera de las direcciones de una puerta de enlace O dirección de broadcast ENTONCES procesar el paquete en el modo especificado para el protocolo.

C. SI el destino esta en una red directamente conectada ENTONCES enviar los paquetes directamente a su destino usando la encapsulación apropiada para el protocolo y el tipo de enlace.

D. SI no hay rutas al destino en la tabla de ruteo O todas las rutas upstream ENTONCES enviar un mensaje de error específico del protocolo y descartar el paquete

E. Elige la siguiente ruta a usar. SI hay más de una ruta, un round-robin alternativo con una frecuencia proporcional al inverso de la métrica compuesta.

Obtener el siguiente salto desde la ruta seleccionada en el próximo paso.

Enviar el paquete al siguiente salto, usando la encapsulación apropiada para el protocolo o el tipo de enlace de datos.

2.- RECEPCIÓN DE ACTUALIZACIONES DEL RUTEO:

Las actualizaciones son recibidas de los routers vecinos. Estas actualizaciones consisten en una lista de entradas, cada una de las cuales de información de un

destino. Puede haber más de una entrada para el mismo destino en una sola actualización, para acomodar múltiples tipos de servicio. Cada una de estas entradas es procesada individualmente.

El proceso de la figura 2, debe ser repetido una vez por cada tipo de servicio soportado por el router (lazo), utilizando la información asociada a este tipo de servicio. Nota: la actual implementación de IGRP no soporta múltiples tipos de servicio, por lo que este lazo no se implementa.

Pasos:

A.- Se realizan pruebas de aceptabilidad al camino. Las actualizaciones son rechazadas si el destino al que se refieren, está en holddown, es decir, el tiempo de expiración del holddown no es cero y mayor que el tiempo actual.

B.- La tabla de ruteo es analizada para ver qué entrada describe un camino que ya es conocido. Un camino en la tabla de ruteo está definido por el destino con el que está asociado, el próximo salto en la lista como parte del camino, la interfase de salida a ser utilizada por el camino, y la información de la fuente (S) (la dirección de donde viene la actualización). La entrada del paquete de actualización describe un camino cuyo destino está listado en la entrada, cuya interfase de salida en la interfase por la que llega la actualización, y cuyo próximo salto y la información de la fuente son la dirección del router que envió la actualización.

H. y T.- El proceso de actualización descrito en la figura 4 esta programado. Este proceso actualizará después de que, este proceso (figura 2) haya acabado. El proceso de actualización solo ocurrirá una vez.

K.- Se realiza si el destino descrito por la entrada actual en el paquete de actualización ya existe en la tabla de ruteo. K compara la nueva métrica calculada a partir de los datos del paquete de actualización con la mejor métrica para el destino.

La mejor métrica no es recalculada en este momento, por lo que si el camino que está siendo considerado está ya en la tabla de ruteo, esta prueba puede comparar la nueva y la vieja métrica para el mismo camino.

L.- Es ejecutado para los caminos que son peores que la mejor métrica existente. Incluye nuevos caminos, el peor de los existentes y los caminos existentes cuya métrica ha aumentado. L examina cual de los nuevos caminos es aceptable. Este examen implementa, una prueba para determinar cual de los nuevos caminos es suficiente bueno para mantenerlo, y también route poisoning. Para ser aceptables, el valor del delay no debe indicar un destino desconocido, y la métrica debe ser aceptable (se comparará con las métricas del resto de caminos al mismo destino). Siendo M la mínima de estas métricas, el nuevo camino es aceptable si es VxM , donde V es la varianza (introducida cuando el router es inicializado, siempre vale 1 en Cisco)

V.- Se realiza cuando la nueva información para el camino indica que la métrica disminuirá. Las métricas de todos los caminos al destino D se comparan. En esta comparación, la nueva métrica para P es utilizada, mejor que una que parezca en la tabla de ruteo. La métrica mínima M es calculada. Entonces todos los caminos a D son examinados de nuevo. Si, la métrica para cualquier camino es mayor que VxM , ese camino es eliminado.

Proceso de actualización del ruteo entrante (Processing incoming ruteo updates)

La actualización de ruteo llega de la fuente S.

Por cada tipo de servicio soportado por la puerta de enlace
Utilice los datos del ruteo asociados a este tipo de servicio.

Para cada destino D mostrado en la actualización

A. Si D es inaceptable o el HoldDown ENTONCES ignorar esta entrada y continúe el lazo con el siguiente destino a D

B calcula la métrica para la ruta de P a D vía S (ver figura 8)

SI el destino a D no está ya en la tabla de ruteo ENTONCES INICIAR

Agregue la ruta a P en la tabla de ruteo, establecer los últimos tiempos de la actualización para P y D al tiempo actual

H Dispara una actualización

Establecer la Métrica compuesta para D y P para la nueva métrica compuesta calculados en el paso B.

FIN

SINO ir al INICIO (de sino. D ya esta en la tabla de ruteo)

K compra la nueva métrica compuesta para P con la mejor métrica existente para D.

Nuevo > Viejo:

L SI D es mostrada como inalcanzable en la actualización, o los HoldDowns están disponibles y la nueva métrica compuesta > (métrica existente para D) * V [Usa 1.1 en ves de V SI V=1, tal como esta en Cisco Release 8.2] O los HoldDowns estan deshabilitados Y P tiene una nueva cuenta de salto > que la vieja cuenta de salto ENTONCES INICIAR

Remover P de la tabla de ruteo if existe

SI P era la última ruta a D ENTONCES a menos que IHoldDowns estén deshabilitados
Establecer el tiempo del HoldDown para D con tiempo actual + tiempo T del
HoldDown y disparar una actualización

FIN

SINO ir al INICIO

Calcular la nueva mejor métrica compuesta para D

Poner la nueva información de la métrica en la entrada para P en la tabla de ruteo

Agregar la ruta P a la tabla de ruteo si no estaba presente.

Establecer los últimos tiempos de la actualización para P y D al tiempo actual.

FIN

NUEVO \leq VIEJO:

V Establecer la métrica compuesta para D y P para la nueva métrica compuesta
calculado en el paso B.

SI cualquiera de las otras rutas a D ahora están fuera de la variación, removerlas

Poner la nueva información de la métrica en la entrada para P en la tabla de ruteo

Establecer los últimos tiempos de la actualización para P y D al tiempo actual.

FIN

FIN DEL LAZO "PARA CADA"

FIN DEL LAZO "PARA CADA"

3.- PERIODIC PROCESSING :

Este proceso se dispara una vez por segundo. Examina varios timers en la tabla de ruteo, para determinar si alguno ha expirado.

Pasos:

U.- Activa el proceso descrito en la figura 4

R. y S.- Son necesarios porque las métricas almacenadas en la tabla de ruteo dependen de la ocupación del canal (channel occupancy). Periódicamente, la ocupación del canal es recalculada, haciendo media del tráfico medido a través de la interfase. Si el último valor calculado difiere del existente, todas las métricas de esta interfase deben ser ajustadas. Todos los caminos en la tabla de rutas son examinadas. Todo camino cuyo próximo salto utilice la interfase "I" recalcula su métrica. Esto se hace de acuerdo con la ecuación 1, usando como ocupación del canal el máximo valor almacenado en la tabla de ruteo y el último valor calculado de ocupación del canal de la interfase.

Proceso periódico (Periodic processing)

El proceso es activado por el reloj regular, por ejemplo, un segundo

Para cada ruta P en la tabla de ruteo (excepto las interfasas directamente conectadas)

SI tiempo actual < Ultimo tiempo de ACTUALIZACIÓN de p + TIEMPO NO VÁLIDO
ENTONCES CONTINÚA CON LA RUTA SIGUIENTE P

Remover P from ruteo table

Si P era la ruta anterior a D entonces establecer la métrica para D a inaccesible a menos que los HoldDown esten deshabilitados, iniciar el HoldDown para D y disparar una actualización

SINO Recalcular la mejor métrica para D

FIN DEL LAZO "PARA CADA"

Para cada destino a D en la tabla de ruteo

SI las métricas de D son inaccesibles entonces INICIAR

Limpiar todas las rutas a D

SI tiempo actual \geq ultimo tiempo de actualización para D + tiempo de limpieza
ENTONCES remover entrada para D

FIN

FIN DEL LAZO "PARA CADA"

Para cada interfaz de la red I unida a la puerta de enlace

R Recalcular el canal de ocupación y la tasa de error

S SI el canal de ocupación O la tasa de error ha cambiado ENTONCES recalcular las métricas.

Fin del lazo "Para Cada"

En los intervalos del tiempo de broadcast

U Disparar una actualización

4.- GENERACIÓN DE LOS MENSAJES DE ACTUALIZACIÓN:

Aquí se describe cómo el router genera las actualizaciones para ser enviadas a los routers vecinos. Un mensaje independiente es enviado por cada interfase del router.

J.- El mensaje es enviado a todos los routers que son accesibles a través de la interfase. Se envía normalmente como un mensaje broadcast. Si la tecnología de red o el protocolo no permite broadcast, puede ser necesario enviar el mensaje individualmente a cada router.

G.- El mensaje se construye añadiendo una entrada por cada destino en la tabla de ruteo. El destino de la información sobre el camino asociada con cada tipo de servicio, debe ser utilizado. En el peor caso, una entrada nueva es añadida a la actualización para cada destino para cada tipo de servicio. De todos modos, antes de añadir una entrada al mensaje en el paso G., las entradas ya añadidas son examinadas. Si una nueva entrada está ya presente en el mensaje de actualización no se añade otra vez. Una nueva entrada duplica la que ya existe cuando los destinos y los routers siguientes (próximo salto) son los mismos.

El pseudo código omite el hecho de que los mensajes de actualización IGRP tienen 3 partes: interior, sistema y exterior. Así, hay actualmente 3 lazos según el destino.

El primero incluye solo subnets de una red.

El segundo incluye todas las grandes redes (no subredes) no marcadas como exterior.

El tercero incluye todas las grandes redes, no exteriores.

E.- Este paso implementa la prueba de Split horizon. En el caso normal, esta prueba falla en rutas cuyo mejor camino salga a la misma interfase por la que la actualización está siendo enviada. Sin embargo, si la actualización está siendo enviada a un destino determinado, (Ej, en respuesta a una solicitud IGRP, desde otro router, o como parte de point-to-point IGRP), split horizon falla solo si el mejor camino viene desde este destino (su información de fuente es la misma que la del destino) y su interfase de salida es la misma por la que viene la solicitud.

Generar los mensajes de la actualización (Generate update messages)

El proceso es causado por la "actualización del disparador"

Para cada interfase de la red I unida a la puerta de actualización

Crear un mensaje vacío de actualización

Para cada tipo de servicio S soportado

Utilice los datos de la ruta/destino para S

Para cada destino a D

E SI cualquier ruta a D tiene un salto siguiente alcanzado a través I ENTONCES continuar con el destino siguiente.

SI cualquier ruta a D con la mínima métrica compuesta está ya en el mensaje de actualización ENTONCES continúe con el destino siguiente

G crea una entrada para D en el mensaje de actualización, usando la información de la métrica de una ruta con la mínima métrica compuesta (véase fig. 8)

FIN DEL LAZO "PARA CADA"

FIN DEL LAZO "PARA CADA"

J SI hay alguna entrada en el mensaje de actualización ENTONCES enviar a la interfase I

FIN DEL LAZO "PARA CADA"

5.-CALCULO DE LA INFORMACIÓN MÉTRICA:

Aquí se describe el procedimiento para calcular las métricas y la cuenta de saltos con la llegada de una actualización en el ruteo.

La entrada a esta función es una entrada para cada destino específico en un paquete de actualización de ruteo.

La salida es un vector de métricas que puede ser usado para calcular la métrica compuesta y la cuenta de saltos.

Si este camino es añadido a la tabla de ruteo, el vector entero de métricas se coloca en la tabla. Los parámetros de la interfase usados en las definiciones siguientes son fijados cuando el router es inicializado, para la interfase por la que llega la actualización de ruteo, espera que la ocupación del canal y la confiabilidad estén basadas en una media del tráfico medido a través de la interfase.

Delay = retardo de paquete + retardo topológico de interfaz

Bandwidth = max (ancho de banda de paquete, ancho de banda de interfaz)

Confiabilidad = min (confiabilidad de paquete, interfaz confiabilidad)

Channel occupancy = max (channel occupancy from packet, interface channel occupancy)

Lo siguiente no forma parte del vector métrico pero si se mantiene en la tabla de ruteo como características del camino:

Hop count = hop count from packet

MTU = min (MTU from packet, interfase MTU)

Remote composite metric = calculada con la ecuación 1, usando los valores del paquete

Composite metric = calculados con la ecuación 1, usando los valores métricos calculados según se describe en esta sección.

Esta sección describe el procedimiento para calcular las métricas y el hop count para las actualizaciones de ruteo que deben ser enviadas.

La entrada está basada en un determinado camino a un destino. Si hay más de un camino a ese destino, se elige el camino cuya métrica sea menor.

SI el destino es inaccesible, esto es indicado usando un valor específico en el campo del delay. Este valor se elige para sea más grande que el delay válido más grande. Para la implementación de la IP todos estos son unos en un campo 24-bit.

Si el destino está directamente accesible desde una de las interfases, utilice el delay, ancho de banda, confiabilidad, y ocupación del canal de la interfase. Establezca la cuenta del salto a 0.

SINO, utilice el vector de la métrica asociada a la ruta en la tabla de ruteo. Agregue uno a la cuenta del salto de la ruta en la tabla de ruteo.

A continuación aparece cómo se obtiene la métrica compuesta que es utilizada actualmente en la versión 8.0 de Cisco:

$$\text{metric} = [K1 * \text{bandwidth} + (K2 * \text{bandwidth}) / (256 - \text{load}) + K3 * \text{delay}] * [K5 / (\text{confiabilidad} + K4)]$$

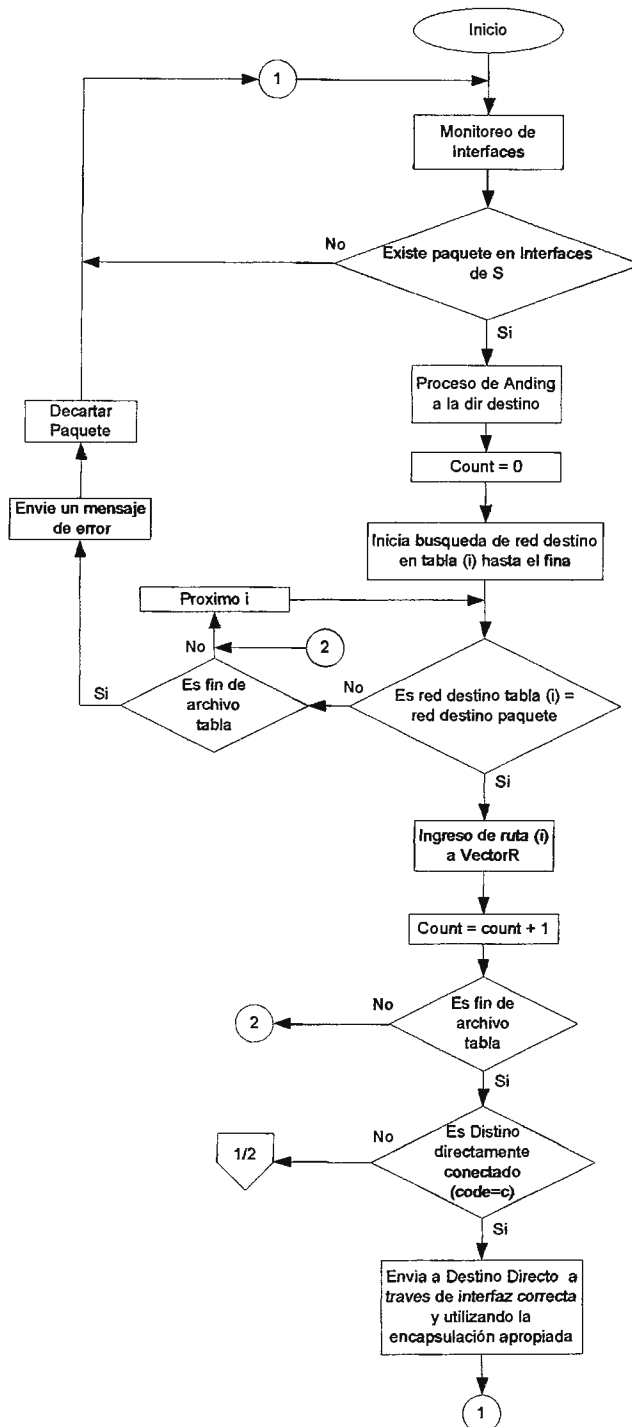
If K5 == 0, El término confiabilidad no está incluido

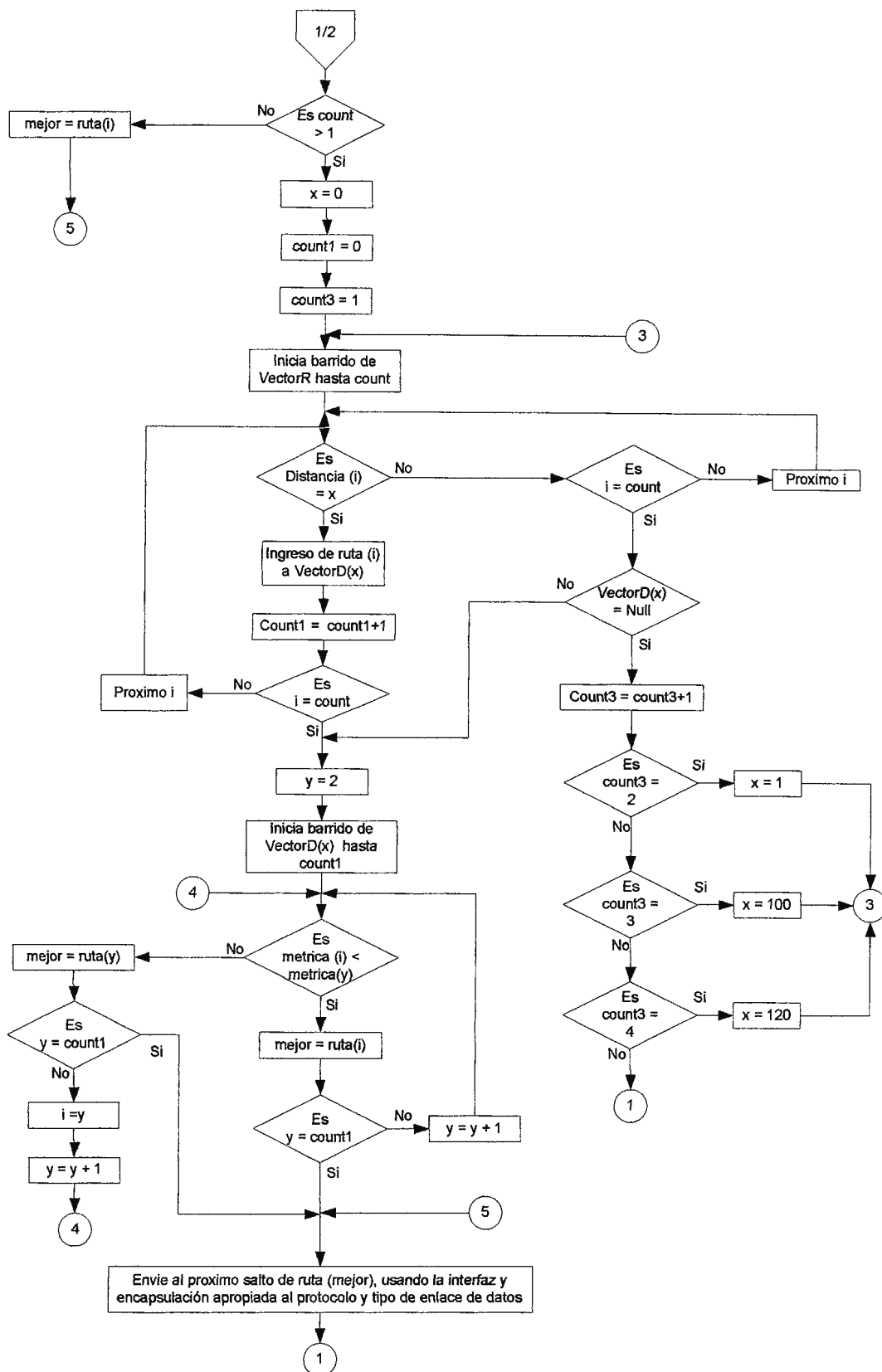
La versión por defecto de IGRP tiene:

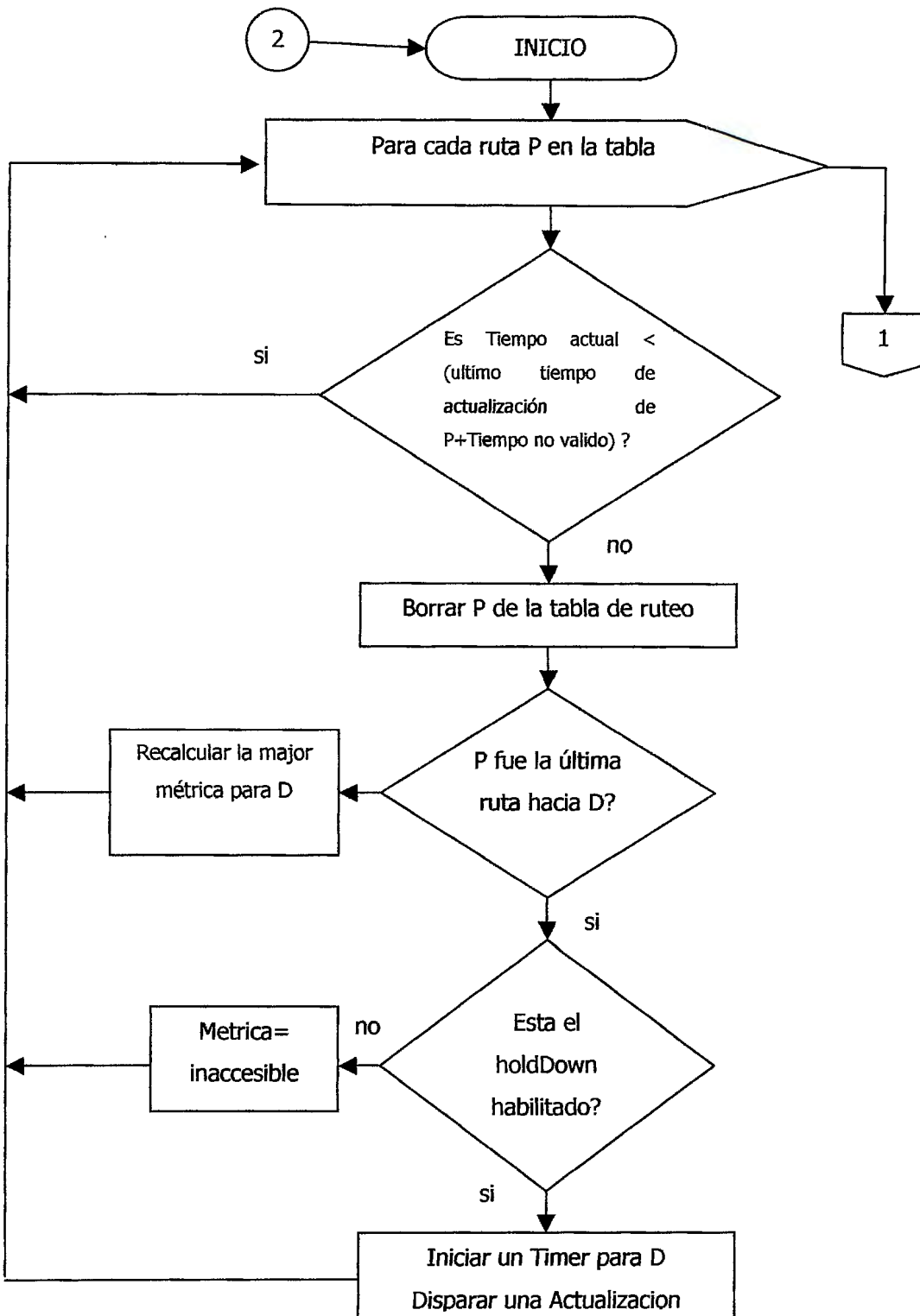
K1 == K3 == 1, K2 == K4 == K5 == 0

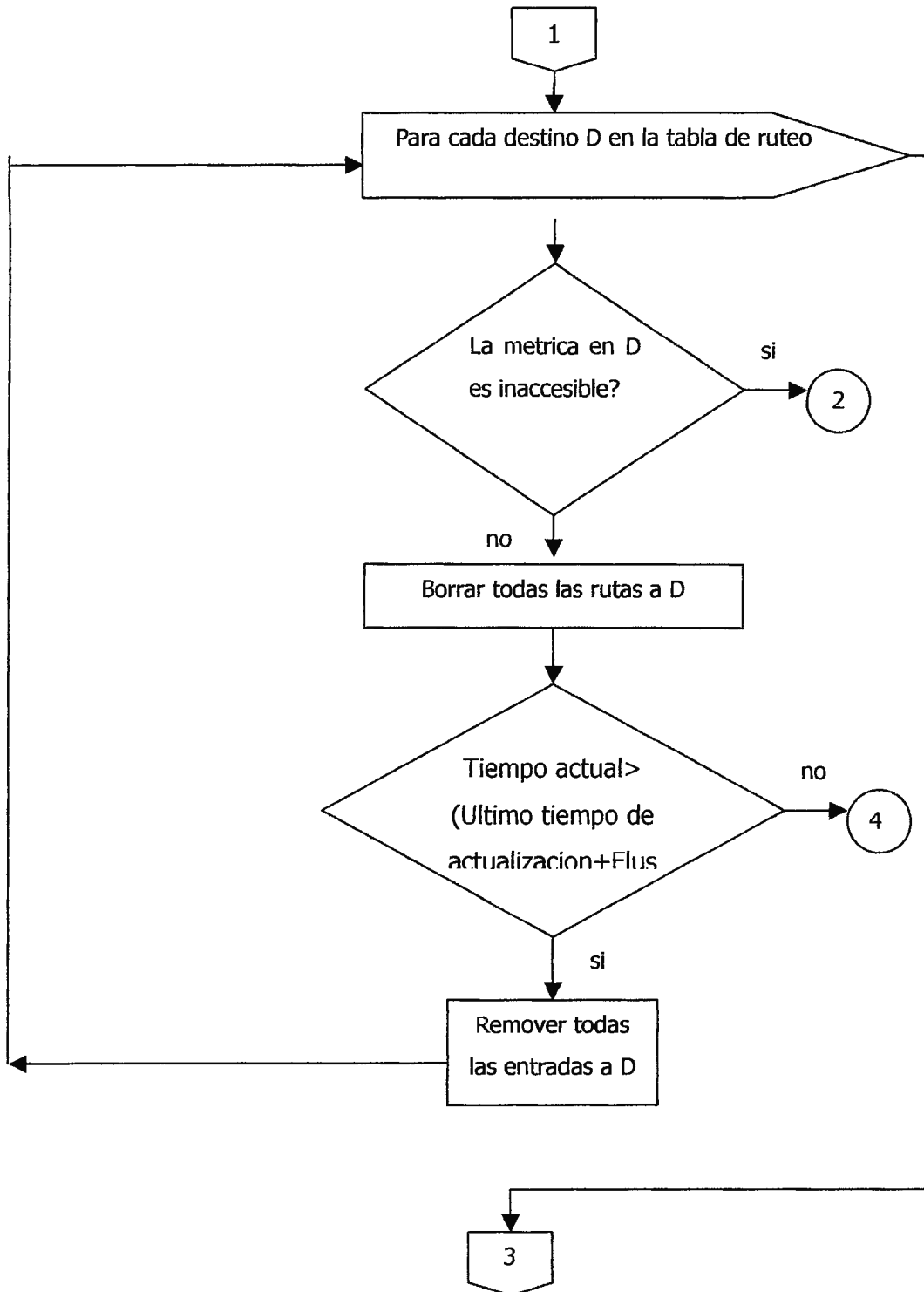
Flujogramas del protocolo IGRP

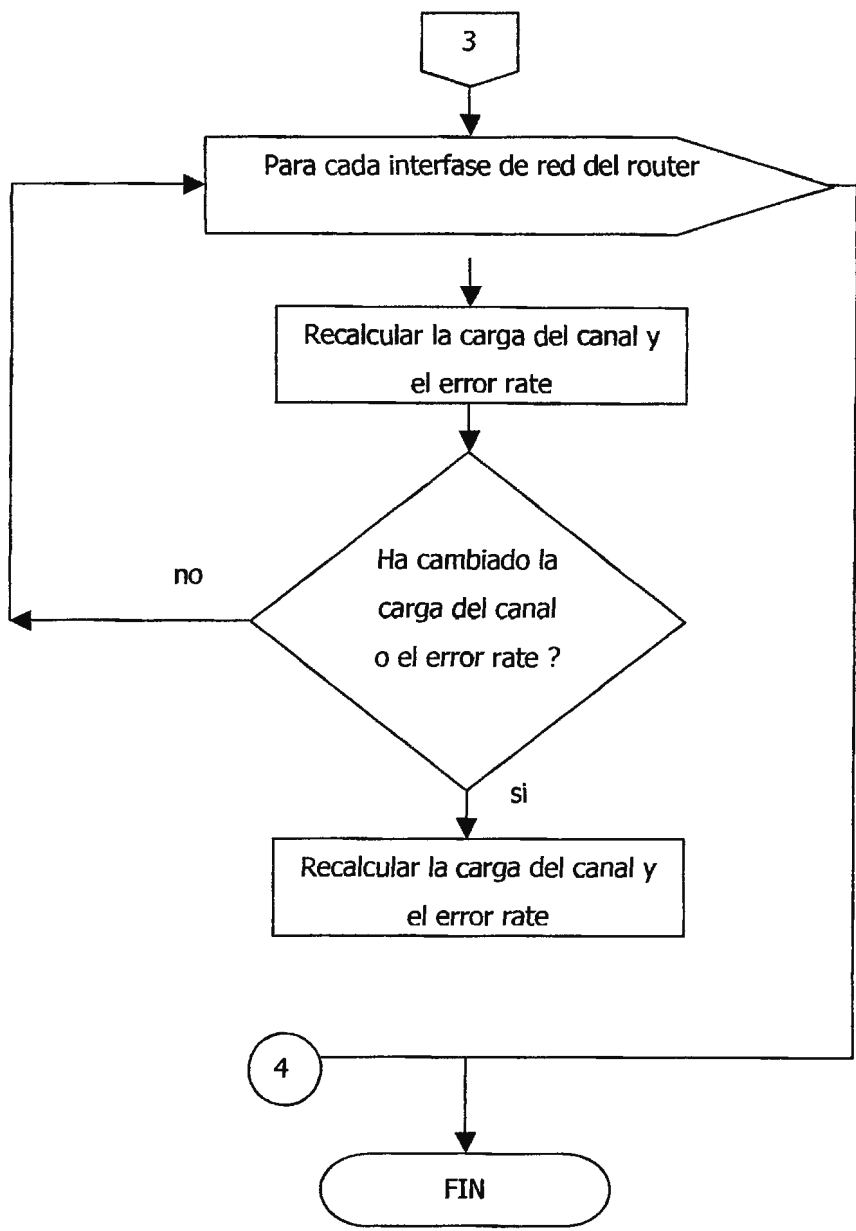
Processing Incoming Packets

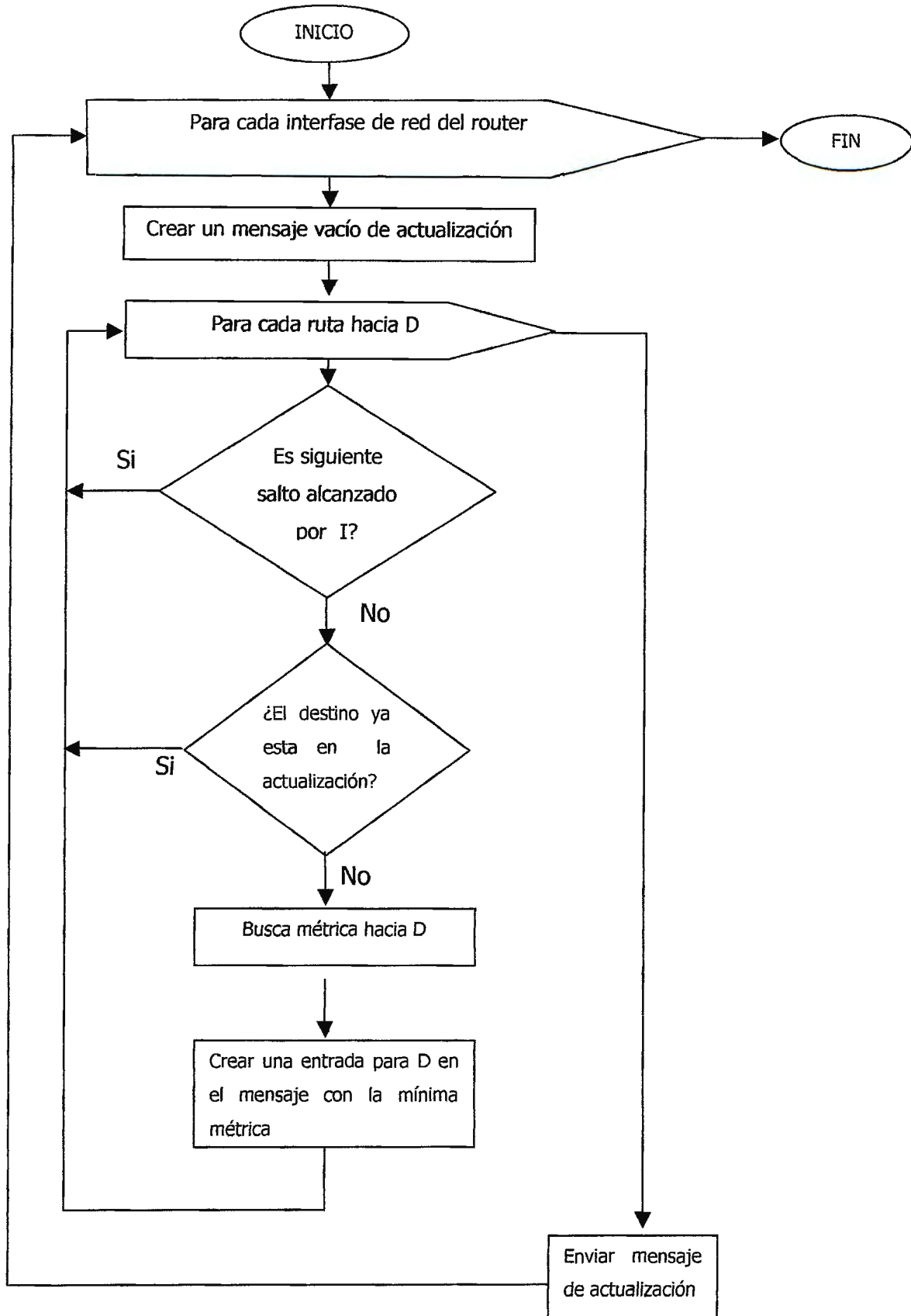




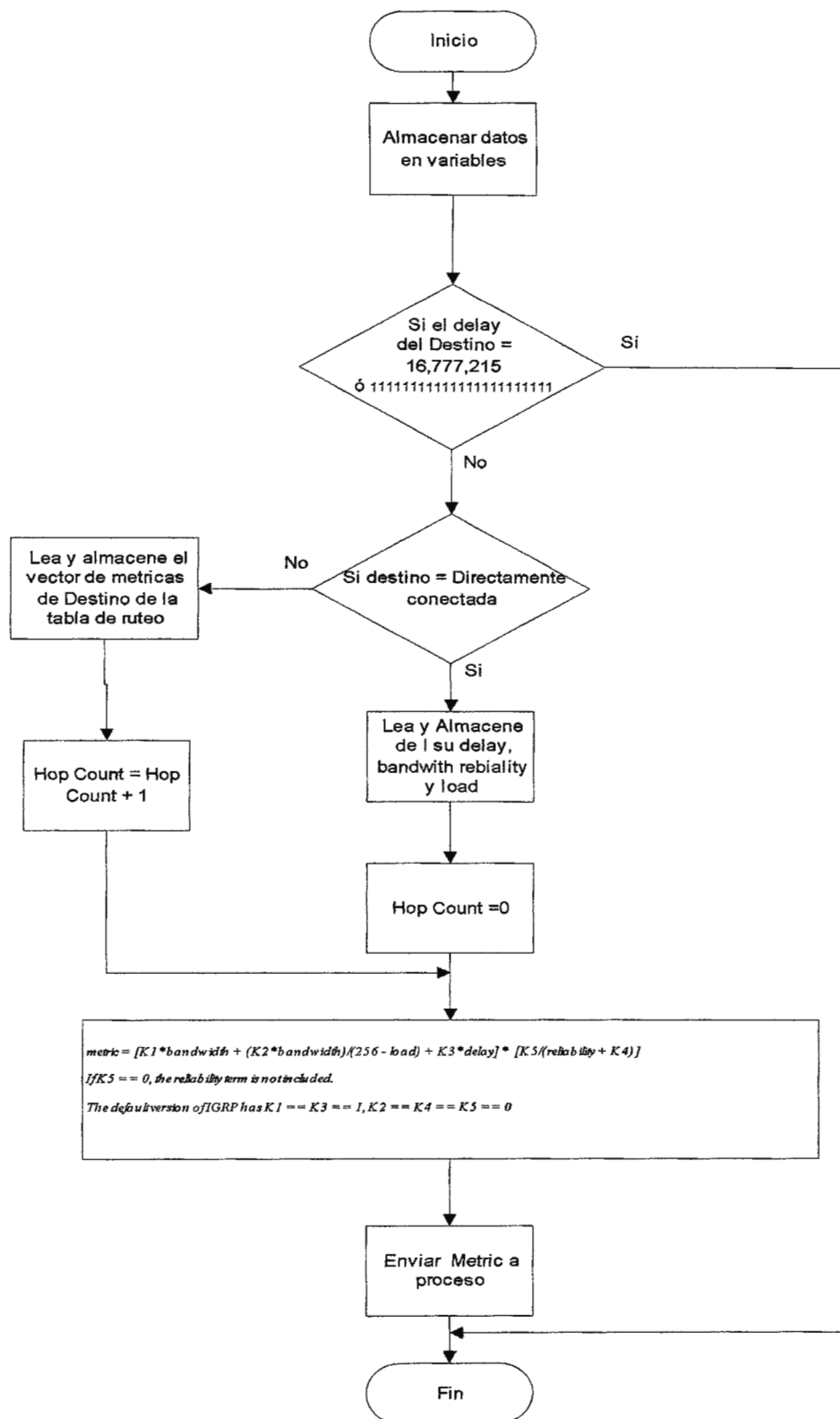
Proceso de Periodic Processing





Proceso de Generación de Actualización

Proceso de Procesamiento de métricas



Operaciones de protocolo de capa de red0

Cuando se necesita enviar un paquete por parte de una aplicación a un destino en una red distinta, el host direcciona la trama de enlace de datos hacia el router, usando una de las interfaces del router es decir su dirección. El proceso de la capa de red del router examina el encabezado del paquete entrante para determinar la red destino y después poder consultar la tabla de enrutamiento que relaciona las interfaces de salida con redes. Luego el paquete se encapsula nuevamente en la trama de enlace de datos optima para la interfaz seleccionada y se localiza en la cola para su entrega al siguiente salto en la ruta.

Rutas estáticas

Son configuradas manualmente por un administrador de red el cual debe de conocer la ruta y cuando existe cualquier cambio en la topología de la red es necesario realizar la actualización.

Rutas dinámicas

Una vez realizada la configuración de un router de manera dinámica este conoce las rutas dinámicamente a través de un proceso de enrutamiento y cuando existe un cambio este se actualiza automáticamente. Este conocimiento dinámico se intercambia entre routers.

Determinación de las distancias en las rutas en la red a través de diversas métricas.

El principal objetivo de un algoritmo de enrutamiento el cual se encarga de actualizar una tabla de enrutamiento es determinar cual es la mejor información que debe incluir en la tabla. El algoritmo genera un número, llamado métrica para cada ruta a través de la red. Usualmente cuando menor sea la métrica, mejor será la ruta.

Vista preliminar sobre conmutación LAN

- Los switches alcanzan esta transferencia de alta velocidad leyendo la dirección MAC de la capa de enlace de datos destino de un paquete. El paquete es enviado al host receptor antes que todo el paquete llegue al switch, esto da como resultado una latencia baja y una alta tasa de velocidad para el envío de paquetes.
- La conmutación Ethernet consiste en segmento de red dedicado ósea punto a punto dentro de switch. Es decir se forma un circuito virtual cuando dos host necesitan comunicarse.
- Aunque el switch elimina los dominios de colisión, aun todos los hosts conectados al switch pertenecen al mismo dominio de broadcast.

Como aprende direcciones un switch LAN

- Leyendo la dirección de origen de cada paquete transmitido
- Escribiendo el puerto donde se escucho el frame
- Las direcciones se aprenden en forma dinámica. Cuando se lee un origen y no se encuentra en la memoria direccionable por contenido (CAM) se aprende y almacena para ser utilizado después.
- Cuando se almacena una dirección se escribe la fecha. Esto permite que las direcciones puedan almacenarse por un periodo establecido de tiempo. Las direcciones que no se utilizan durante un periodo establecido de tiempo se eliminan de la lista.

Beneficios de la conmutación

- Permite a muchos usuarios comunicarse en paralelo mediante el uso de circuitos virtuales y segmentos de red dedicados en un entorno libre de colisión
- Maximiza el ancho de banda disponible en el medio

Conmutación simétrica y asimétrica

Simétrica

Brinda conexiones conmutadas entre puertos con el mismo ancho de banda

Asimétrico

Se optimiza para flujos de red cliente – servidor. En este tipo de switch se necesita buffering de memoria para permitir el paso del tráfico por el puerto

Método de conmutación

Método de corte

EL switch lee la dirección de destino antes de recibir todo el paquete, después el paquete se envía antes de que llegue completo.

Ventaja: Latencia corta debido a al solo leer el paquete se comienza a enviar

Desventaja: Detección de errores baja debido a poco tiempo disponible

Almacenamiento y envío

El frame completo es recibido antes de que el envío tome lugar. Este método implica latencia mientras todo el frame es recibido. Se produce una mayor latencia con frames más grandes, pero una alta detección de errores porque el frame puede ser examinado completamente.

VLAN (LAN VIRTUAL)

Las LAN se dividen cada vez más en grupos de trabajo conectados por medio de backbones comunes para formar topologías de LAN virtual (VLAN). Las VLAN segmentan lógicamente la infraestructura física de la LAN en subredes diferentes (dominios de broadcast para Ethernet) de modo que los frames de broadcast sólo se conmutan entre puertos de la misma VLAN, esto con el fin de resolver problemas asociados con movimientos, adiciones y cambios de ubicación y además ofrece mayor seguridad con la creación de grupos de usuarios seguros, con una mejor gestión y control de la actividad broadcast.

Un componente esencial en la VLAN es el switch, ya que estos switches brindan la inteligencia para agrupar usuarios, puertos, o direcciones lógicas en comunidades de intereses comunes.

Los switches Ethernet están diseñados para segmentar físicamente una LAN en dominios de colisión individuales, pero cada segmento es aun parte de un mismo dominio de broadcast. La cantidad de segmento es un switch es igual a un dominio de broadcast, es decir un broadcast en un segmento de colisión es visto en todos los demás segmentos.

Las redes de área local virtuales (VLAN) son un grupo de usuarios o dispositivos de red que no se limitan a un segmento físico. Una VLAN es un grupo de lógico de usuarios o dispositivos. Estos usuarios o dispositivos se agrupan por función, departamento, aplicación, etc. No importa la ubicación física en un segmento. Una VLAN crea un dominio de broadcast único que no esta limitado físicamente y recibe el tratamiento de una subred.

La configuración de la VLAN se realiza por el software del switch.

Considerando el diseño de una red que consiste solamente dispositivos de capa 2. Por ejemplo, este diseño podría ser un segmento simple de ethernet. Una red conmutada totalmente de capa 2 es llamada como una red de topología plana. Una red plana es un simple dominio de broadcast, tal que cada dispositivo conectado puede ver cada paquete de broadcast que es transmitido. Como el número de estaciones aumenta en la red, así también el número de broadcast.

Debido a los fundamentos de la capa 2, las redes planas no pueden contener rutas redundantes para el balanceo de carga o la tolerancia a fallos.

Un ambiente conmutado ofrece la tecnología de superar las limitaciones de las redes planas. Las redes conmutadas pueden ser subdividas en redes virtuales (VLANs). Por definición, una VLAN es un simple dominio de broadcast. Todos los dispositivos conectados a la VLAN reciben broadcast desde otros miembros de la VLAN. Sin embargo, los dispositivos conectados a una VLAN diferente no podrán recibir el mismo broadcast.

Una VLAN esta compuesta de miembros definidos comunicándose como un segmento lógico de red. En contraste, un segmento físico, consiste de dispositivos que tienen estar conectados al segmento de cable físico. Una VLAN puede tener miembros conectados y localizados en cualquier lugar de la red, tan largo como la conectividad en la VLAN lo permita entre los miembros de la VLAN.

Membresía de VLAN

Cuando una VLAN es proporcionada en una capa de acceso del switch. Un usuario final tiene que tener algunos significados para ganar la membresía. Existen dos tipos de membresía: VLANs Estáticas y VLANs Dinámicas.

VLANs Estáticas

Las VLANs Estáticas ofrecen membresía basada en puertos, cuando los puertos del switch están asignados a una VLAN específica. Los dispositivos del usuario final vienen a ser miembros en una VLAN basada en puertos físicos del switch a los que están conectados.

Los puertos del switch están asignados a las VLANs por la intervención manual de administrador de red.

VLANs Dinámicas

Las VLANs dinámicas son usadas para proveer membresía basada en direcciones MAC de un dispositivo de usuario final. Cuando un dispositivo esta conectado a un puerto del switch, el switch tiene que consultar una base de datos para establecer la membresía VLAN. Un Administrador de red tiene que asignar la dirección MAC del usuario a una VLAN en la base de datos de un Servidor de Políticas de Membresía de VLANs.

VLAN Trunks

En la capa de acceso, dispositivos de usuario final conectado a los puertos del switch que provee conectividad a cada VLAN. Los dispositivos colgados son inconscientes de

cualquier estructura de VLAN y simple adjunte a lo que aparece en un segmento de red físico normal. Recuerde, enviando información desde un enlace en una VLAN a otra VLAN no es posible sin la intervención de un dispositivo adicional. Como uno de capa 3 (router) o un puente de capa 2 externo.

Un trunk link, Sin embargo, puede transportar más de una VLAN a través de un simple puerto del switch. Los Trunk Links son mas beneficiosos cuando switches están conectados a otros switches o switches están conectados a routers.

Un trunk link no esta asignado a una VLAN especifica, En vez, una, muchas o todas las VLANs activas pueden ser transportadas entre los switches usando un simple enlace trunk físico.

Cuantas VLANs sean agregadas a una red, así el número de enlaces pueden crecer rápidamente. Un más eficiente uso de las interfaces físicas y cableado es la razón del uso del trunking.

VLAN Frame Identification

Debido que un trunk link puede ser usado para transportar muchas VLANs, un switch tiene que identificar los frames de las VLANs, como cuando son enviadas y recibidas sobre un trunk link. Frame identification o tagging, asigna un único ID para cada frame transportado en un trunk link. Este ID puede ser pensado como el numero de VLAN o el "color" de la VLAN.

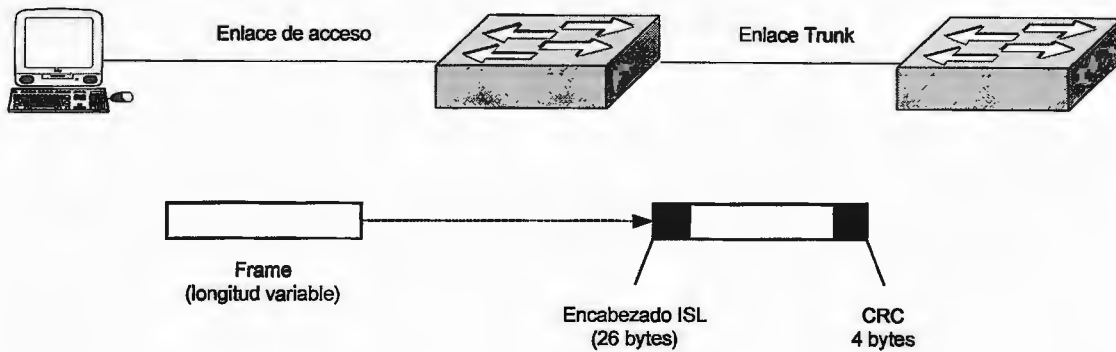
Protocolo Inter - Switch Link (ISL)

El Protocolo Inter-Switch Link es un método propietario de Cisco que sirve para preservar la identificación origen de la VLAN, en los frames que están pasando sobre un trunk link. ISL realiza identificación de frame en la capa 2 por encapsulación de cada frame entre un header (encabezado) y un trailer (terminador).

Cuando el destino de un frame es fuera de un trunk link a otro switch o router. ISL agrega un encabezado de 26 bytes y un trailer de 4-bytes al frame. La VLAN origen es identificado con un identificador de VLAN de 10-bits en el encabezado.

El trailer contiene un chequeo de redundancia cíclica (CRC) para asegurar la integridad de la data del nuevo frame encapsulado.

Figura : Identificación del Frame ISL



Si un frame esta destinado para un link de acceso. La encapsulación ISL (ambas encabezado y trailer) son quitados antes de la transmisión.

Herramientas.

PING (*Packet InterNet Groper*). Esta aplicación TCP/IP envía datagramas ECHO_REQUEST del protocolo ICMP hacia un host destino determinado solicitando una respuesta y mide el tiempo en que se tarda en retornarla todo esto por medio de un datagrama llamado ECHO_RESPONSE.

Esta aplicación ayuda a probar si existe un enlace entre un host origen y un host origen al evaluar las respuestas.

Respuestas ECHO_RESPONSE:

- Respuesta desde 168.243.144.130: bytes=32 tiempo=70ms TTL=121 (El datagrama ICMP llego a su destino con éxito)
- Host de destino inaccesible

- Tiempo de espera agotado para esta solicitud (El tiempo de vida expiro y no encontró el destino)

También hace un reporte de las estadísticas de ping:

- No. Paquetes: enviados
- No. de paquetes recibidos
- No. de paquetes perdidos

Tiempos aproximados de recorrido redondo en milisegundos:

- Tiempo mínimo
- Tiempo máximo
- Tiempo promedio

TRACE Se basa en el protocolo ICMP. Esta aplicación consiste en rastrear la ruta o los saltos que realiza un paquete en todos los gateways antes de llegar hacia el host destino. Esto lo realiza enviando un datagrama IP con un tiempo de vida (TTL) obteniendo así un tiempo de respuesta de cada gateway.

CAPITULO VI

DESARROLLO DEL DISEÑO

1. DISEÑO DEL SISTEMA

1.1. DISEÑO DE MENUS

Menú Archivo: Opciones para trabajar con un archivo de almacenamiento del simulador

Nuevo : Abre un nuevo proyecto colocando una nueva plantilla de diseño

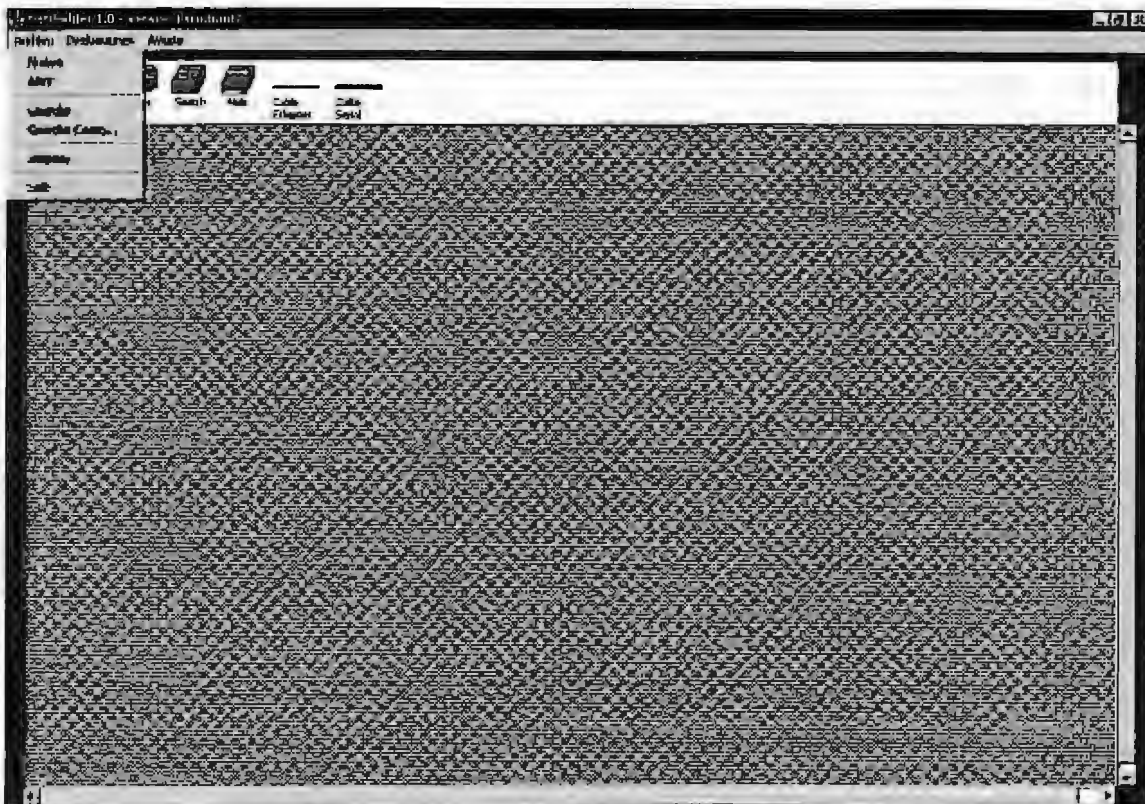
Abrir: Carga un diseño y configuraciones guardadas en disco

Guardar: Guarda en disco el diseño y configuraciones de un proyecto en memoria

Guardar Como...: Guarda con otro nombre un proyecto ya guardado en disco

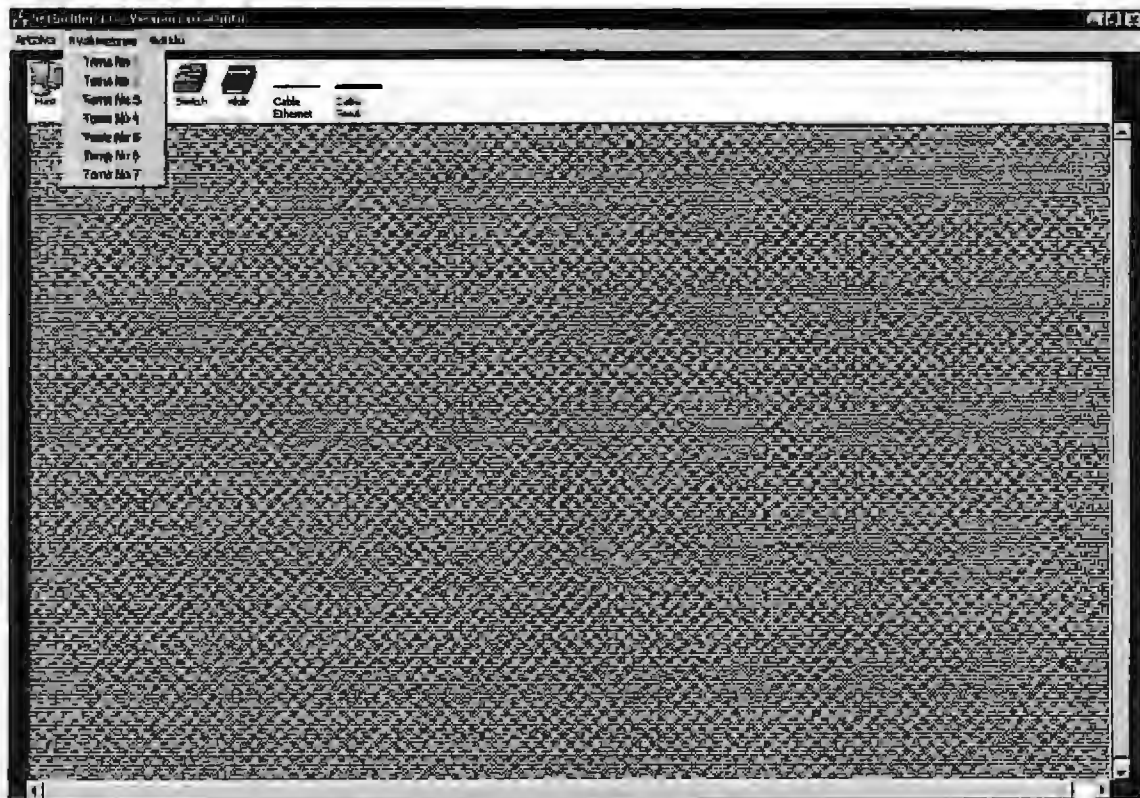
Imprimir: Imprime diseño y configuraciones de proyecto en memoria

Salir: Salir completamente del simulador

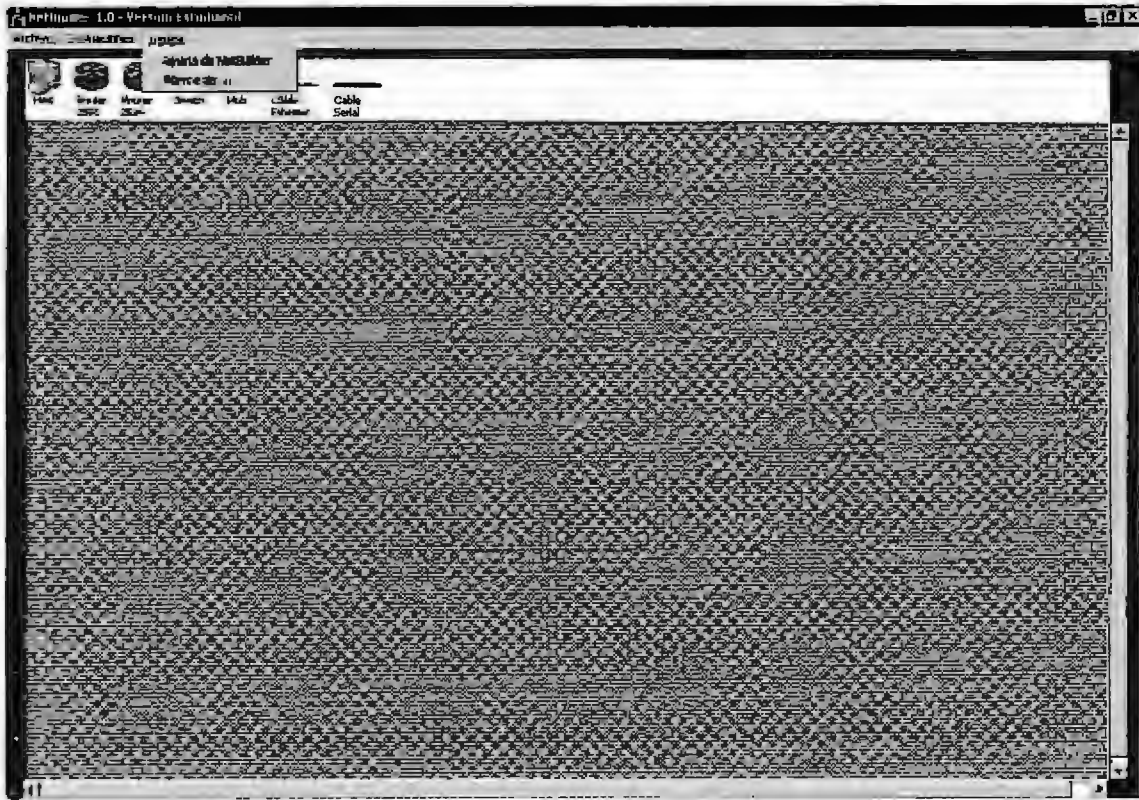


Menú Evaluaciones: Opciones para seleccionar un Laboratorio Evaluado entre ocho

1. Router Básico
2. Router y LAN
3. Router y WAN
4. RIP Y LAN
5. IGRP Básico
6. IGRP Avanzado
7. VLAN Básico
8. VLAN Avanzado



Menu Ayuda: Opciones de ayuda básica del simulador y el Acerca De... Simulador
Ayuda de NETZBuilder
Acerca de ...



Menú contextual: Opciones según el dispositivo donde se haga click derecho

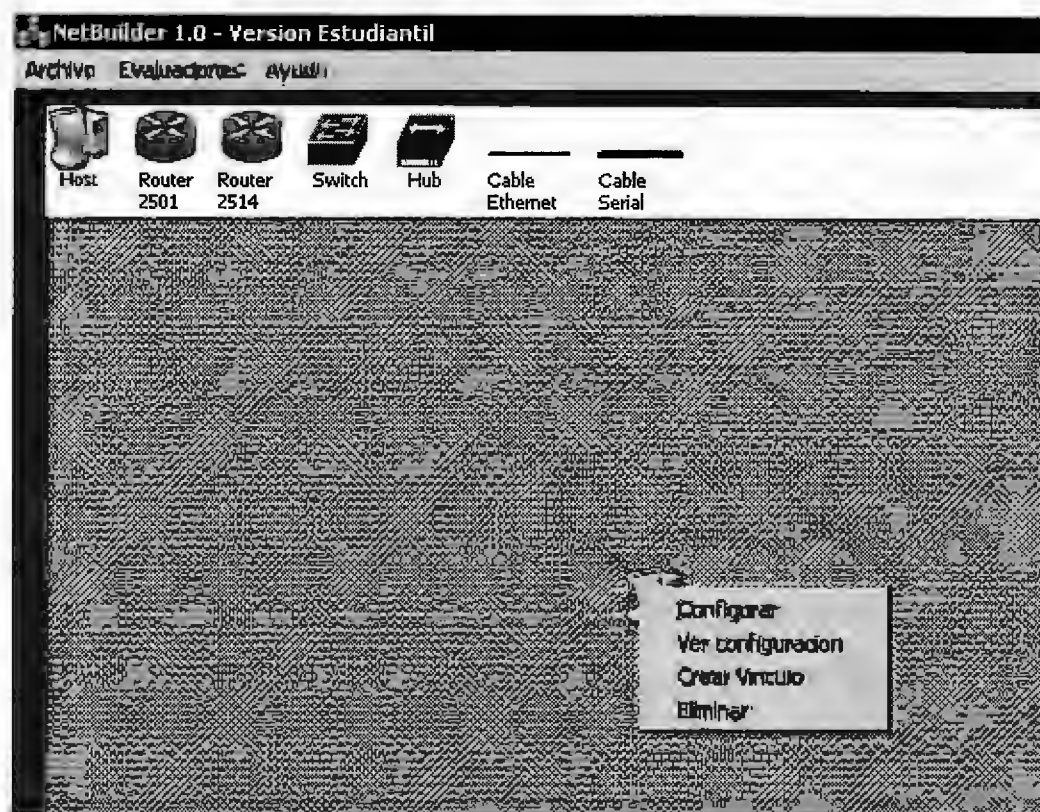
Configurar: Configurar dispositivos (host, switch o router)

Ver Configuración: Muestra la configuración de un dispositivo

Crear Vínculo: Establece la conexión entre un dispositivo y otro por medio de un cable

Eliminar: Elimina dispositivos (host, switch, router o hub)

Ver Historial: Muestra la ventana de historial de los proceso internos del router



1.2. DISEÑO DE PANTALLAS

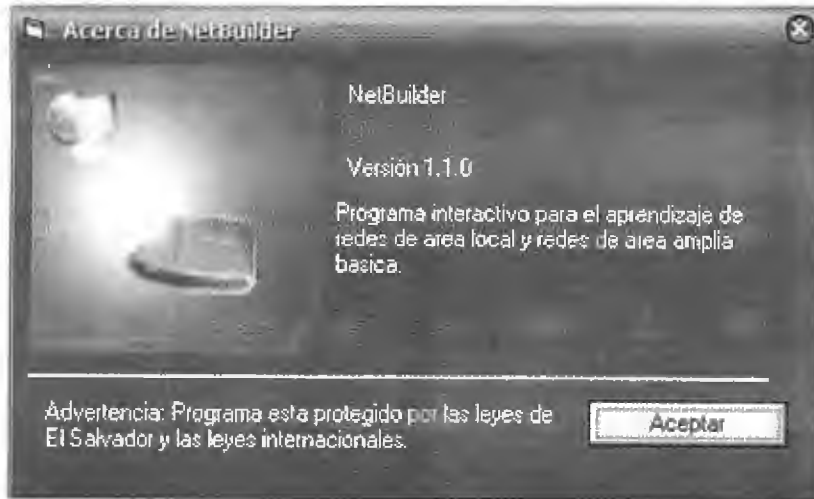
El simulador será diseñado en un entorno grafico, a continuación se mostraran las pantallas que se utilizaran para la configuración y verificación de los dispositivos que así lo requieran.

Pantalla Principal: Diseño de pantalla principal de simulador. Contiene:
a. Barra de menús

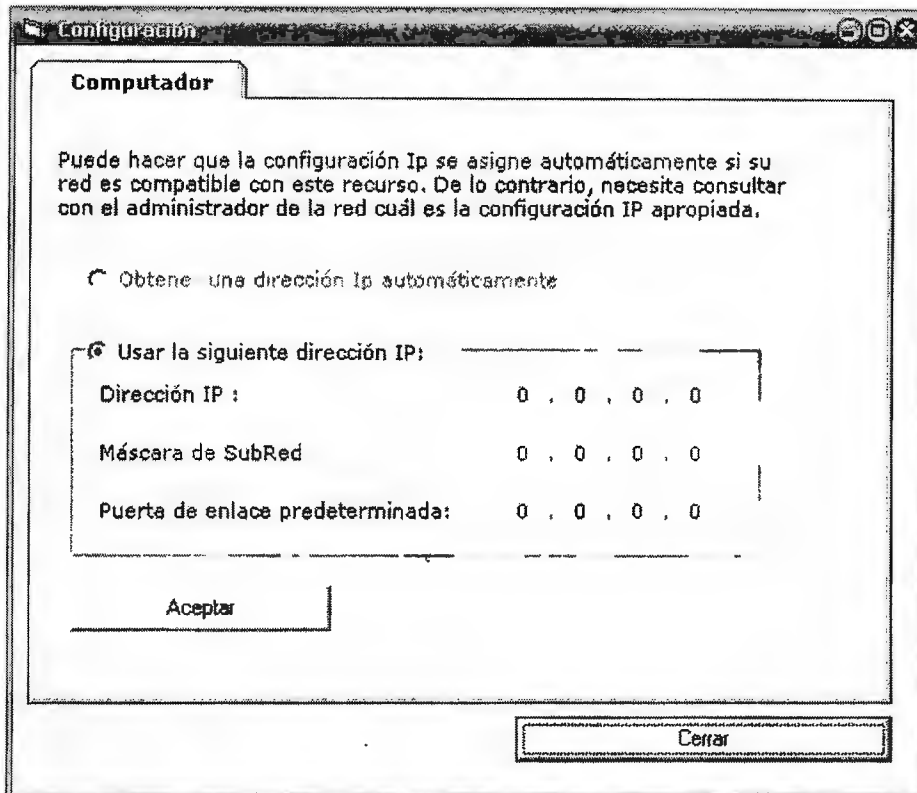
- b. Barra de dispositivos
- c. Plantilla de diseño



Ventana acerca de...: Ventana que muestra la Información General acerca del simulador



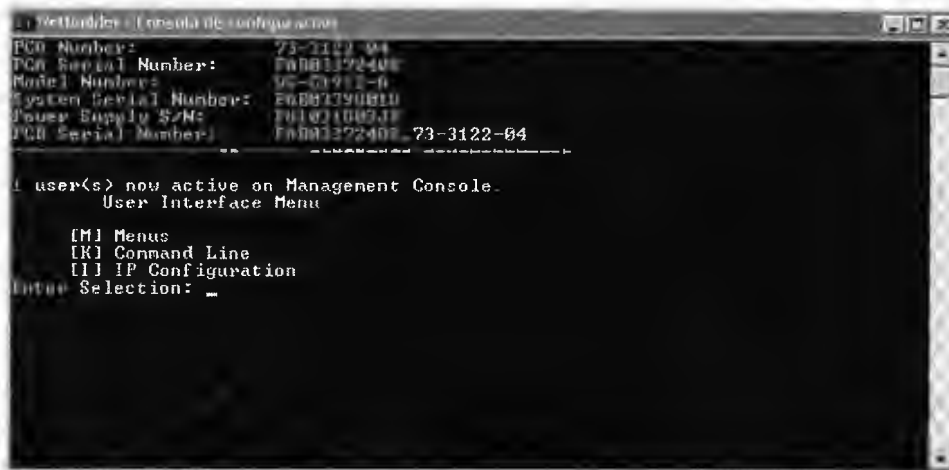
Ventana de configuración del host: Ventana para configurar la IP, Mascara de Subred y Default Gateway de un host



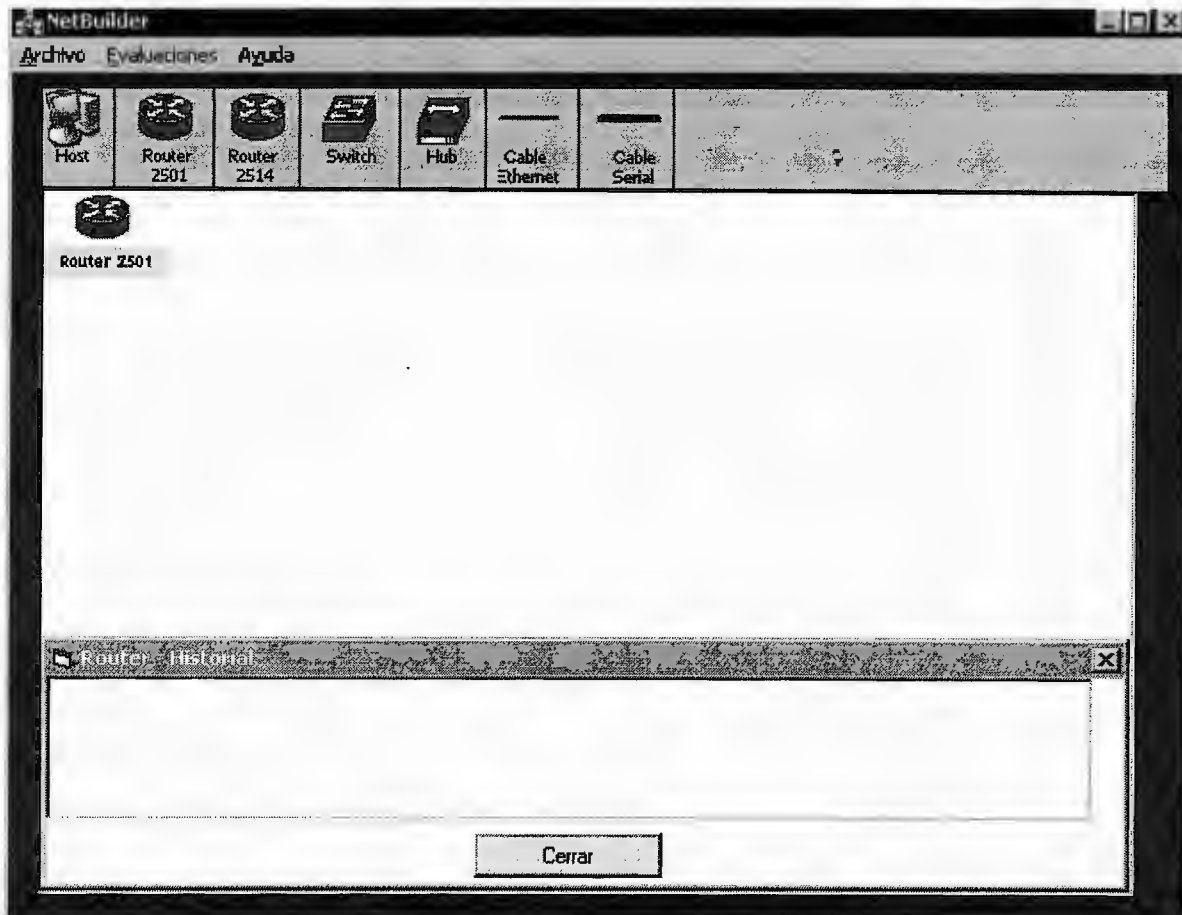
Consola de configuración del router: Ventana de consola para configuración de router



Consola de configuración del switch: Ventana de consola para configuración de switch



Ventana de Historial: Muestra mensajes de lo que esta sucediendo internamente en el router, cuando sus procesos de enrutamiento se están simulando.



1.3. DIAGRAMAS DE RED

El simulador será creado con fines de enseñar a como configurar algunos dispositivos, los cuales son necesarios para la comunicación por medio de redes, teniendo en cuenta los siguientes aspectos:

1. Tamaño del área de trabajo: Debido a que el área de trabajo esta delimitada al tamaño de la pantalla, se ha tenido que restringir el número de dispositivos.
2. Propósitos de aprendizaje: Para enseñar a configurar una red, solo es necesario tener cierta cantidad de dispositivos para evaluarla.

Limitaciones

Retomando lo que se ha mencionado anteriormente, a continuación se explicara detalladamente las restricciones en cuanto al número de un mismo dispositivo en la red.

- No se permitirá la unión de Hubs en cascada, ya que para fines de aprendizaje del proceso de comunicación con un hub se puede demostrar, la unión de estos solo seria una extensión.
- Solo se permitirá tener cuatro hosts por LAN, porque aunque los dispositivos como los hubs tengan 8 puertos o mas, con 3 se puede probar el funcionamiento y así no saturar el área de trabajo con estos.
- Las demás limitaciones se deben al área de trabajo y se decidió por ese número de dispositivos basándose en el laboratorio de la Academia Cisco en la Universidad Don Bosco.
- El número de LAN permitidas será de 6, ya que 4 de los routers que se pueden utilizar tienen una interfase Ethernet y solo 1 tiene 2.
- Unión de dos switches solo cuando sea implementación de trucking en VLAN y es una unión por diseño. Tampoco se permite la unión de dos hubs en cascada.

- Limitación de dispositivos por diseño.

Máximo de Routers	5 (Cuatro 2501 y Un 2514)
Máximo de Switch	6
Máximo de Hub	6
Máximo de LAN	6
Máximo de Hosts por LAN	4

Nota: El número máximo de dispositivos (Switches o Hubs) por diseño es 6, esto significa que puede haber 6 switches, o 6 hubs o la combinación entre ellos que hagan un total de 6.

Detalle de posibles combinaciones entre # máximo de hubs y switches por diseño

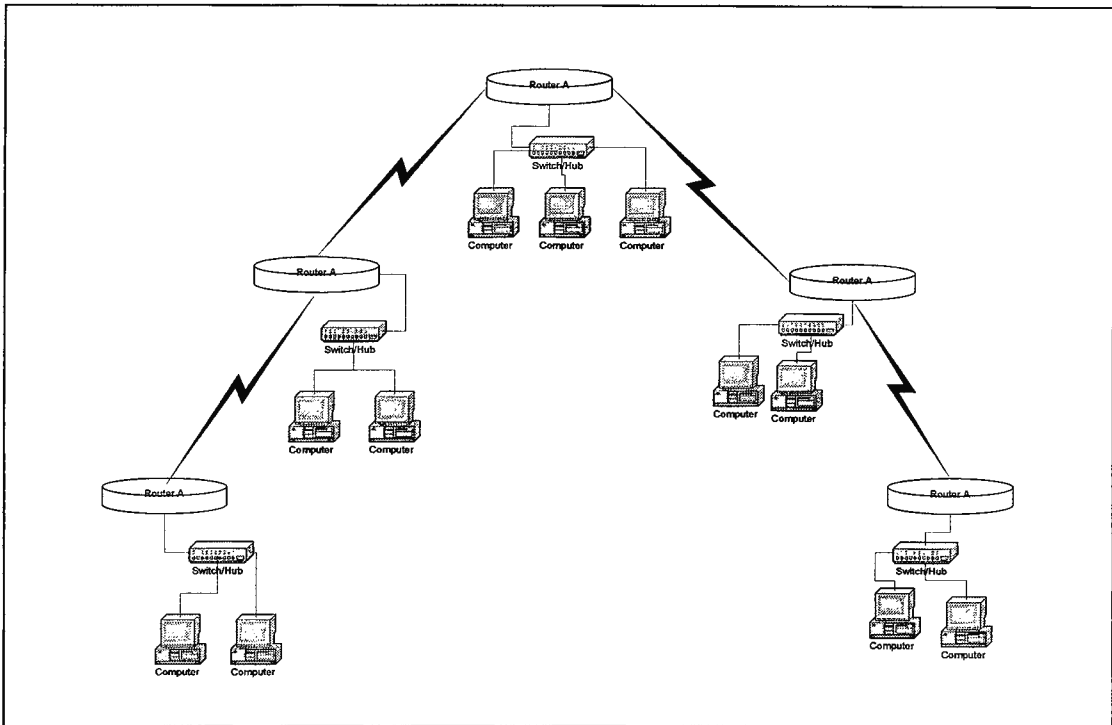
Combinación	# de Hubs	# de Switches	#Total de Dispositivos
A	6	0	6
B	5	1	6
C	4	2	6
D	3	3	6
E	2	4	6
F	1	5	6
G	0	6	6

Detalle de posibles combinaciones entre # máximo hubs y switches por diseño cuando existe trunking

Combinación	# de Hubs	# de Switches	#Total de Dispositivos
A	4	0	4
B	3	1	4
C	2	2	4
D	1	3	4
E	0	4	4

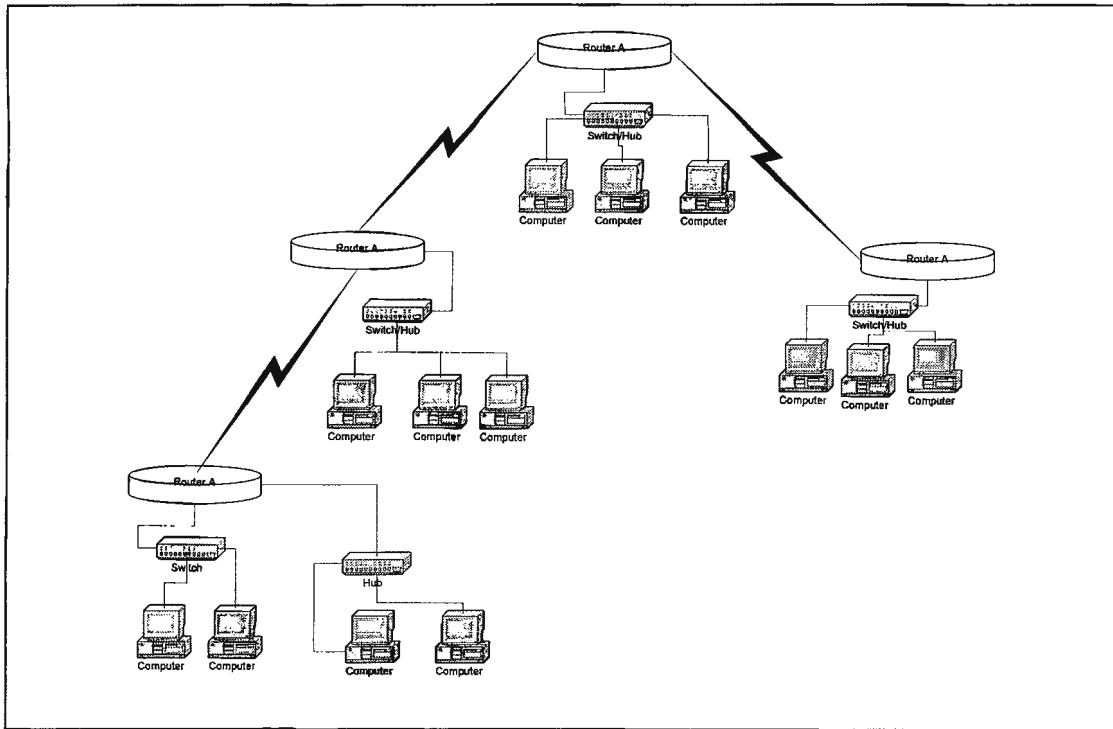
Nota: Tomándose en cuenta que en Trunking ya hay dos switch utilizados

A continuación se definirán algunos ejemplos de Diseños de red, que se podrán probar en el simulador.



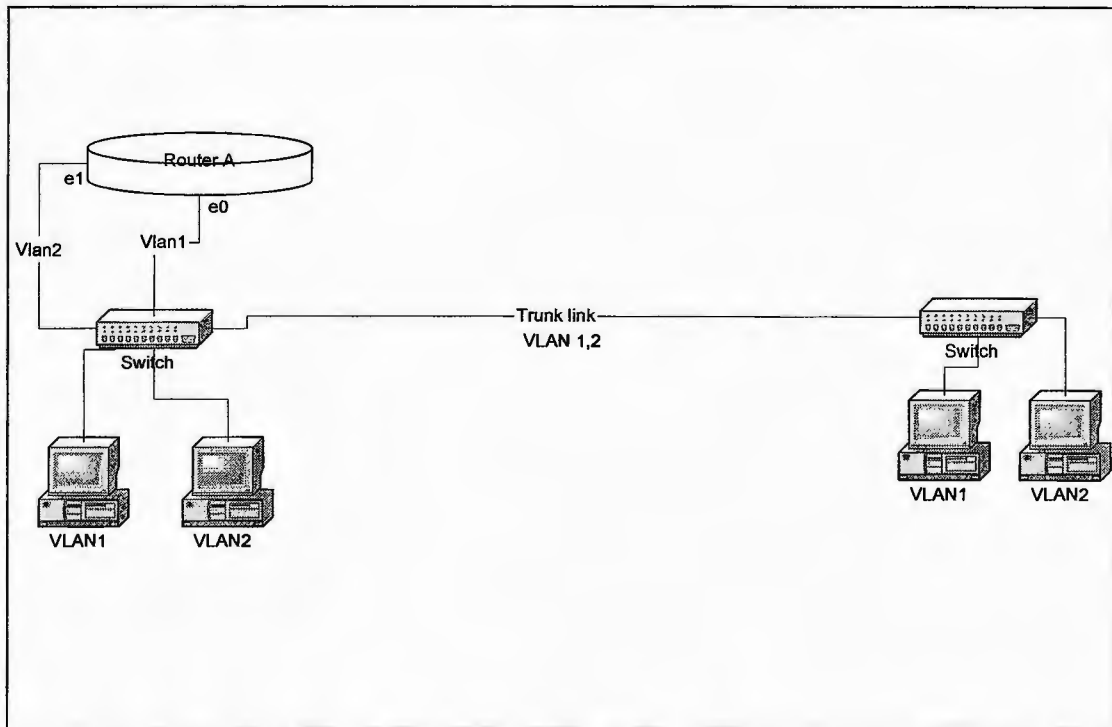
Diseño 1.

5 Routers en cadena, representan el numero máximo de routers por diseño, a cada uno de estos se le ha conectado un hub el cual tiene dos o tres computadoras unidas a este, pues da lo mismo en cuestión de configuración tener dos computadoras en un hub, que tener 8 a este.



Diseño 2.

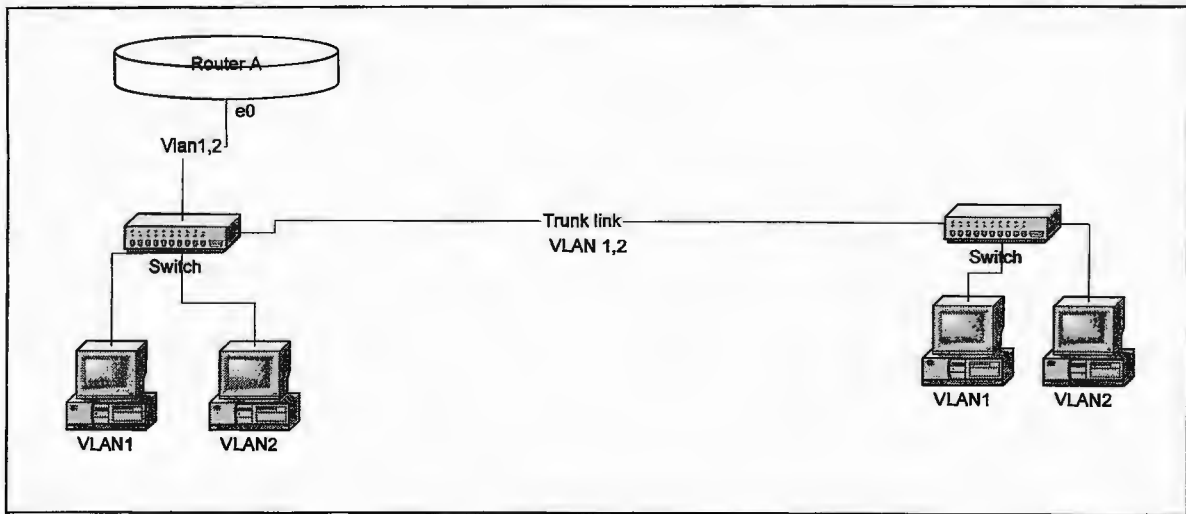
3 Routers con una interfase Ethernet y 1 con dos interfaces Ethernet, con un switch o hub conectados a dichas interfaces.



Diseño 3.

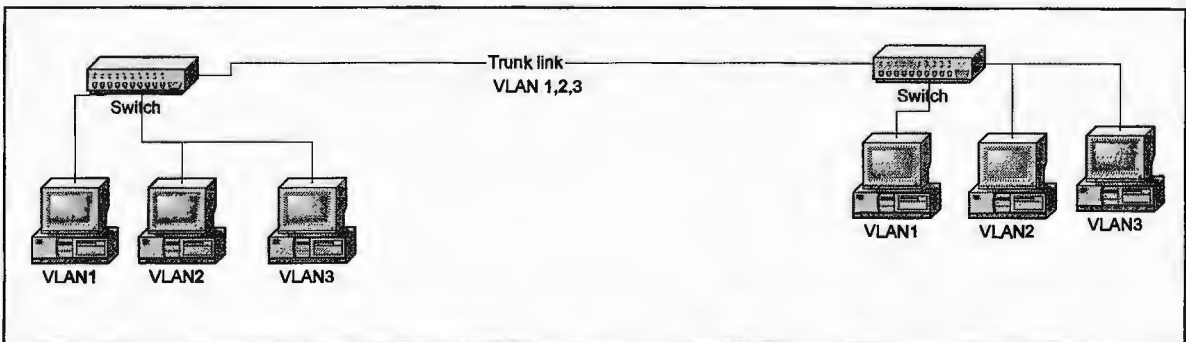
1 Router con dos interfaces Ethernet, el cual comunicara a dos VLANs que están distribuidas en 2 Switch unidos en Trunking. La unión de Switches por Trunking solo se podrá hacer una por diseño.

Este diseño plantea la forma de que dos VLANs (las cuales no se pueden comunicar entre si mediante los switch) puedan comunicarse utilizando un router.



Diseño 4.

Este diseño comunica dos switches con tres VLANs configuradas solo ocupando un puerto en cada switch.



Diseño 5.

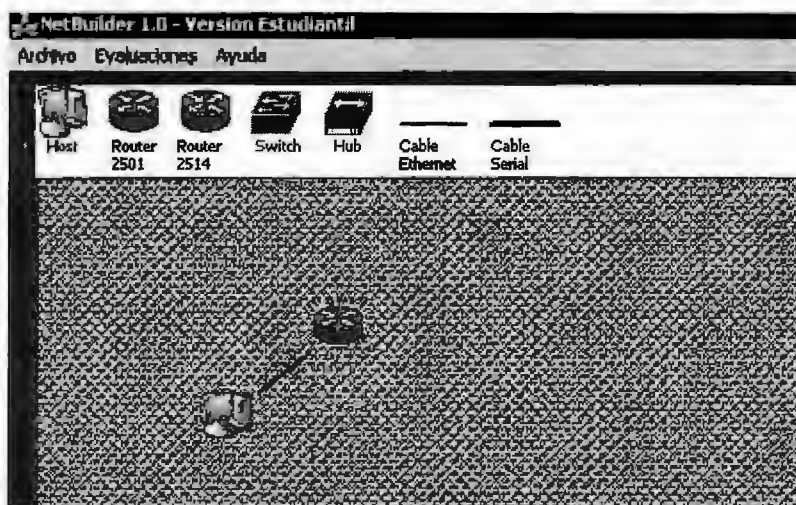
Este diseño comunica dos switches con dos VLANs configuradas solo ocupando un puerto en cada switch.

1.4. DISEÑO DE LABORATORIOS

Lab 1. Router Básico

Objetivos

- Configuración de contraseña encriptada y sin encriptación
- Cambio de nombre del router
- Configuración de Interfaz Ethernet
- Configuración de NIC de Host
- Revisión de versión de Router
- Uso de Ping para probar conexión de host a Interfaz Ethernet de Router
- Familiarizarse con el comando Help
- Entrar en los diferentes nodos de Router (Usuarios, Privilegiado, Configuración Global, Configuración específica de Interfaz E0 y configuración específica de enrutamiento)
- Guardar configuración en NVRAM

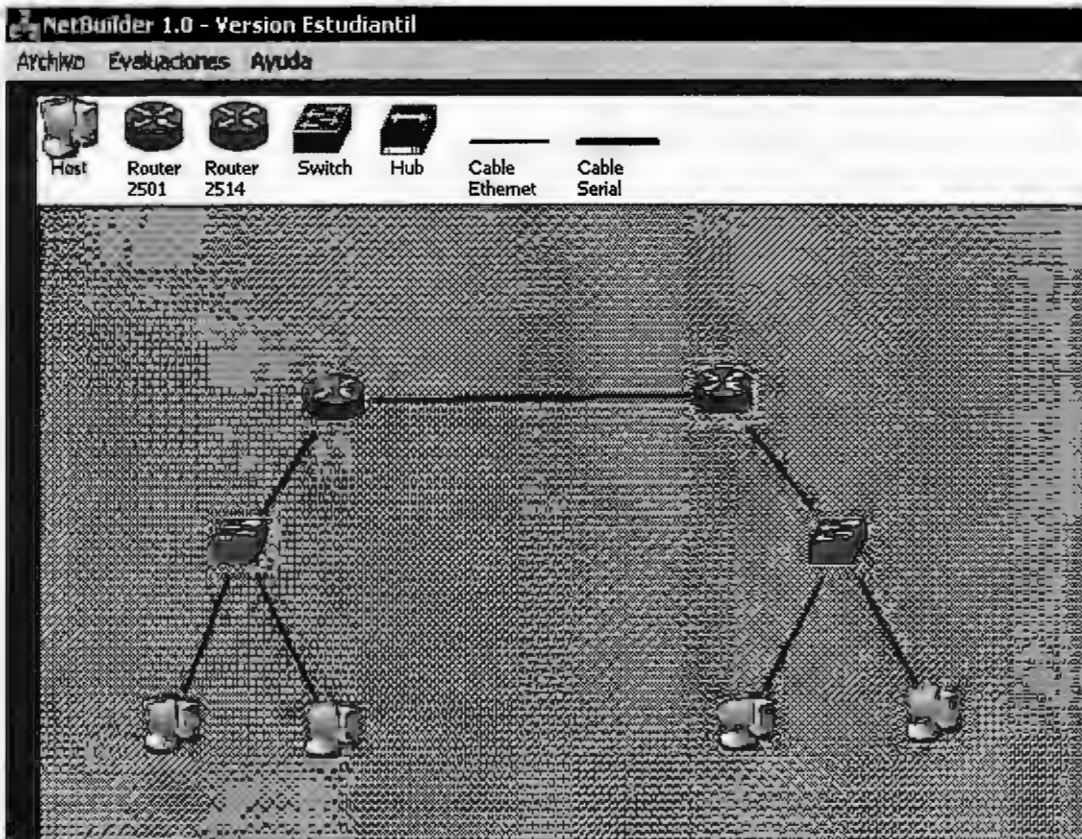


Diseño de red a Utilizar para el Lab 1.

Lab 2. Router y LAN

Objetivos

- Configuración de password encriptado
- Cambio de nombre del router
- Configuración de interfaces Ethernet
- Configuración de interfaces Seriales
- Dar de alta a interfaces
- Uso de Ping para probar conexión de host a Interfaz Ethernet de Router
- Configuración de Rip
- Configuración de la NIC de los host para cada LAN
- Crear Subredes
- Guardar configuración en NVRAM

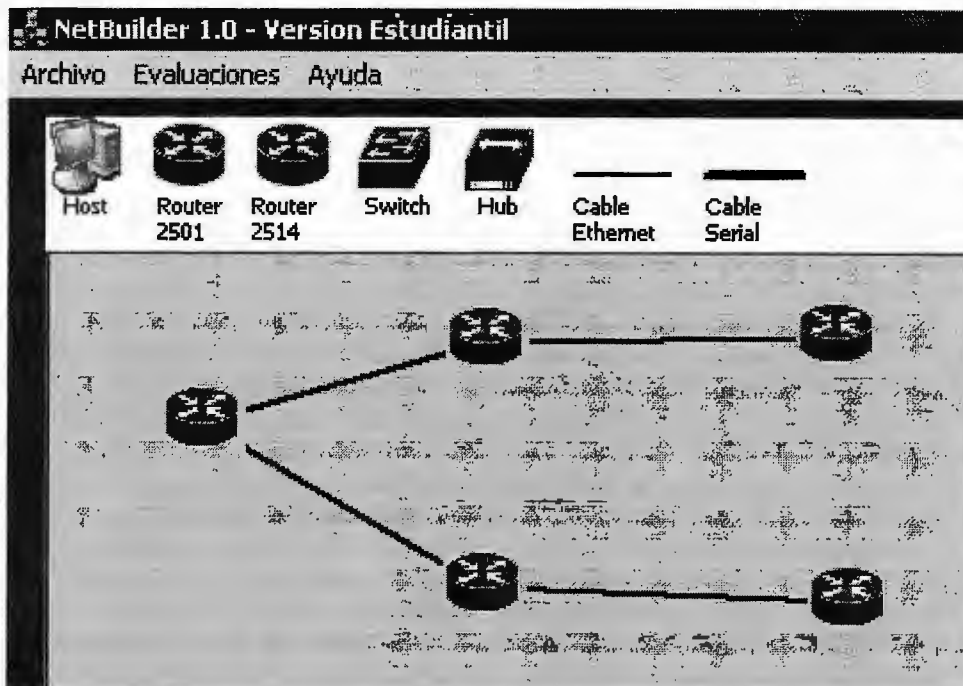


Diseño de red a Utilizar para el Lab 2.

Lab 3 Router y WAN

Objetivos

- Configuración de password encriptado
- Cambio de nombre del router
- Configuración de Interfaces Seriales
- Dar de alta a interfaces
- Uso de Ping para probar conexión de host a Interfaz Ethernet de Router
- Uso de Tracert
- Configuración de Rip
- Crear Subredes
- Guardar configuración en NVRAM
- Revisar estado de Interfaz

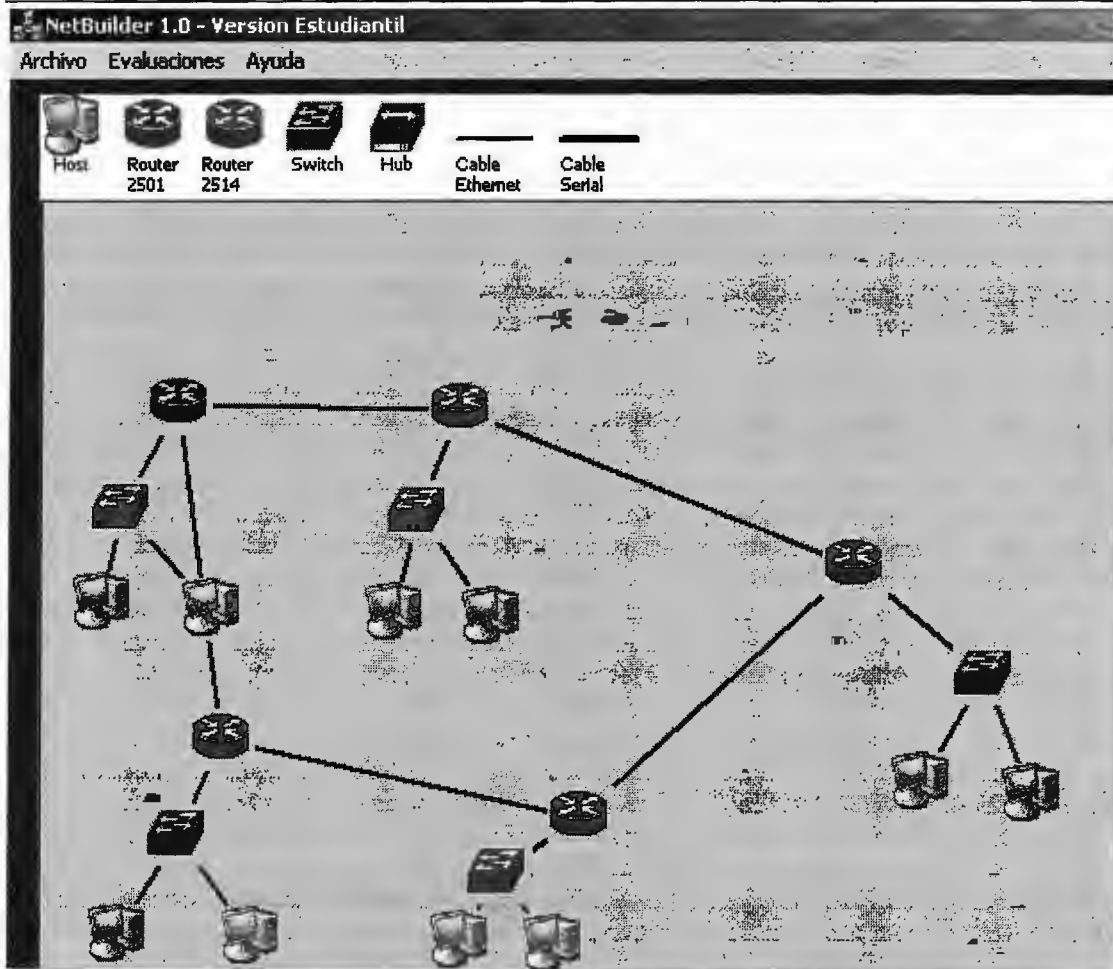


Diseño de red a Utilizar para el Lab 3.

Lab 4. RIP y LAN

Objetivos

- Configuración de password encriptado
- Cambio de nombre del router
- Configuración de interfaces Ethernet
- Configuración de interfaces Seriales
- Dar de alta a interfaces
- Uso de Ping para probar conexión de host a Interfaz Ethernet de Router
- Configuración de Rip
- Visualización y determinación de mejor ruta
- Configuración de la NIC de los host para cada LAN
- Crear Subredes
- Guardar configuración en NVRAM

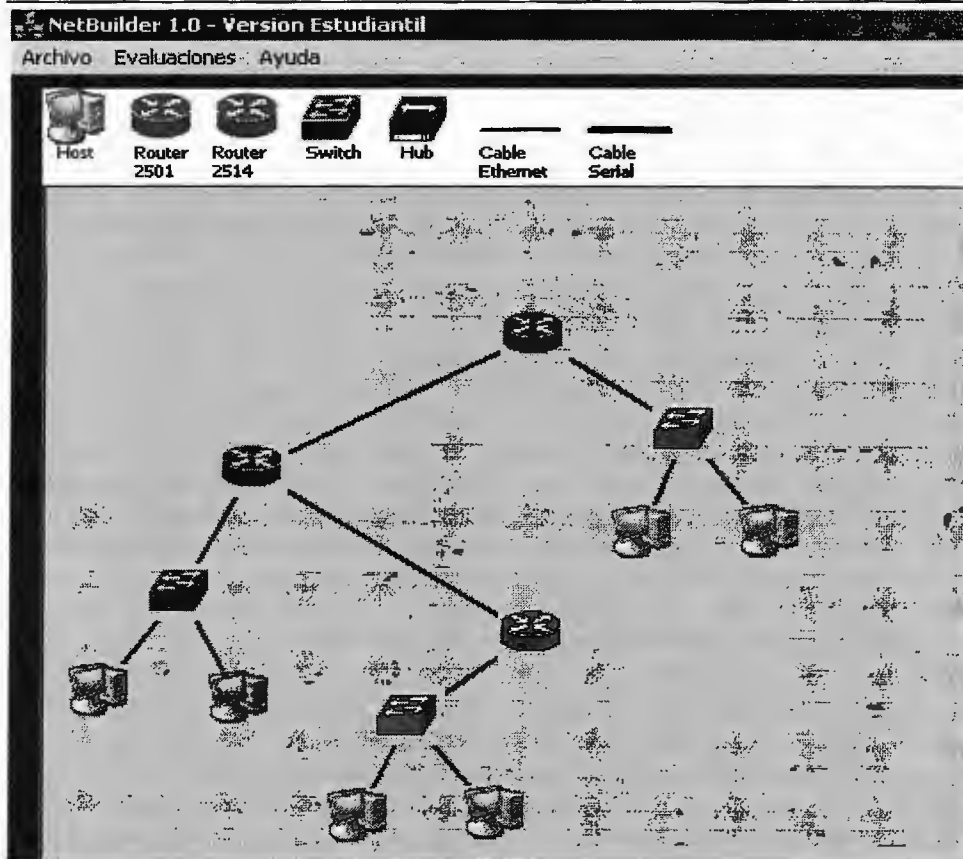


Diseño de red a Utilizar para el Lab 4.

Lab 5. IGRP básico

Objetivos

- Configuración de password encriptado
- Cambio de nombre del router
- Configuración de interfaces Ethernet
- Configuración de interfaces Seriales
- Dar de alta a interfaces
- Configuración de IGRP
- Uso de Ping para probar conexión de host a Interfaz Ethernet de Router
- Configuración de la NIC de los host para cada LAN
- Crear Subredes
- Guardar configuración en NVRAM

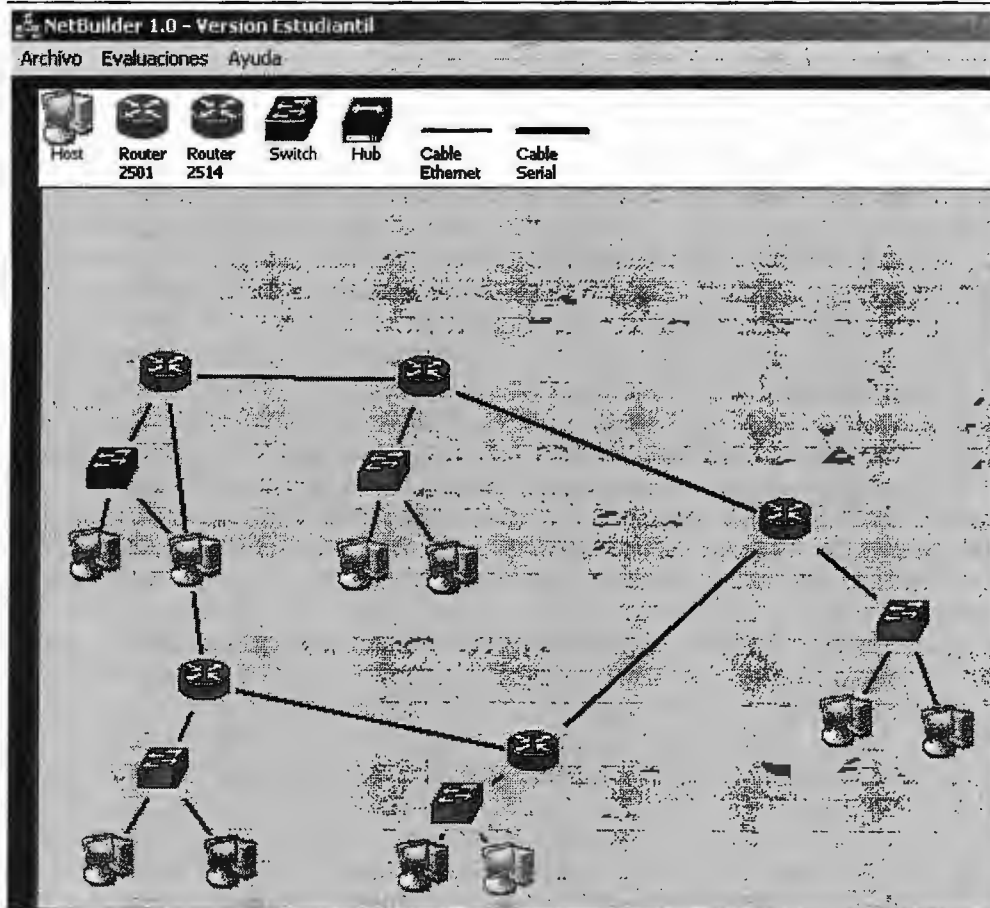


Diseño de red a Utilizar para el Lab 5.

Lab 6. IGRP Avanzado

Objetivos

- Configuración de password encriptado
- Cambio de nombre del router
- Configuración de interfaces Ethernet
- Configuración de interfaces Seriales
- Dar de alta a interfaces
- Configuración de IGRP
- Visualización y determinación de la mejor ruta
- Uso de Ping para probar conexión de host a Interfaz Ethernet de Router
- Uso de Tracert
- Configuración de la NIC de los host para cada LAN
- Crear Subredes
- Guardar configuración en NVRAM

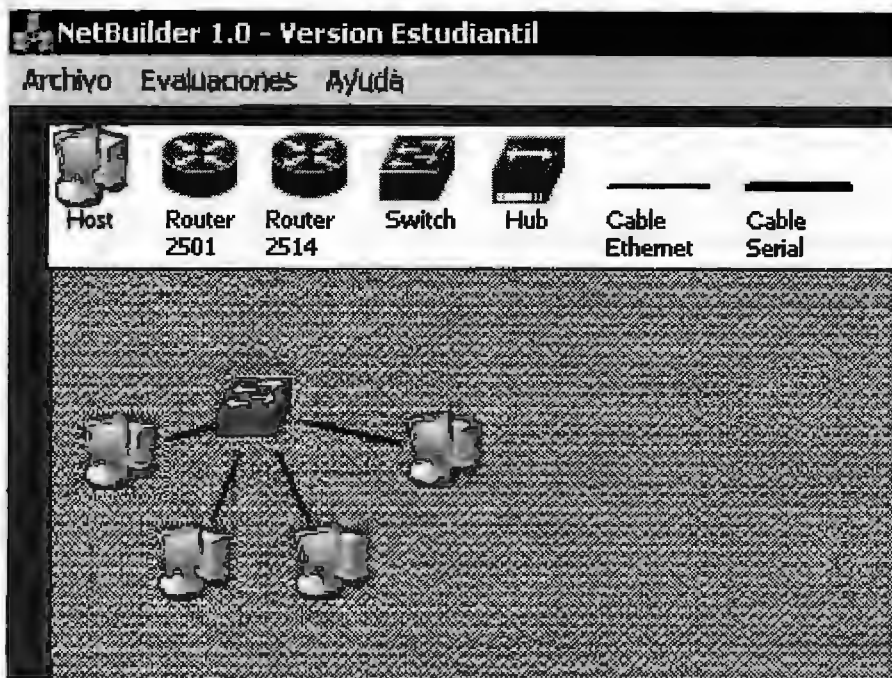


Diseño de red a Utilizar para el Lab 6.

Lab 7. VLAN Básico

Objetivos

- Configuración de Interfaz de cada Host
- Configuración básica de VLAN
- Demostrar la comunicación de VLANs
- Guardar configuración en NVRAM
- Demostrar que las VLANs son dominios de Broadcast diferentes.

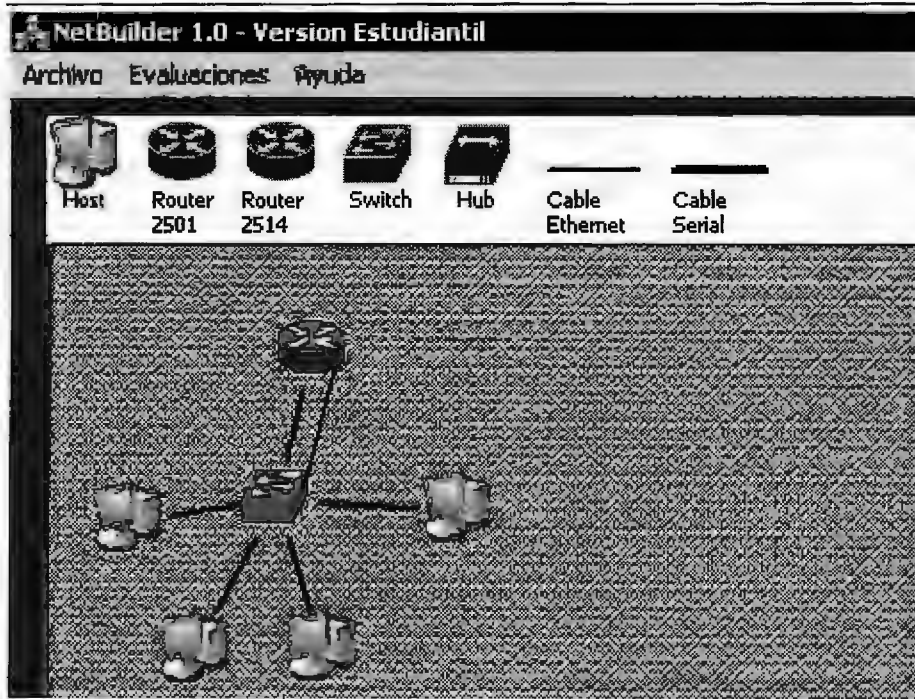


Diseño de red a Utilizar para el Lab 7.

Lab 8. VLAN Avanzado

Objetivos

- Configuración de contraseña encriptada y sin encriptación
- Cambio de nombre del router
- Configuración de Interfaz Ethernet
- Configuración de NIC de Host
- Configuración básica de VLAN
- Demostrar la comunicación de VLANs
- Guardar configuración en NVRAM
- Demostrar que dos VLANs diferentes se pueden comunicar por medio de un dispositivo de capa 3 (Router).



Diseño de red a Utilizar para el Lab 8.

1.5. DESCRIPCIÓN DE COMANDOS

?

Se utiliza el comando **?** en cualquier modo para obtener la ayuda o listado de comando con sus respectivas descripciones según el modo en que se encuentre.

Sintaxis

?

Modo

Usuario, Privilegiado, Configuración global y configuración específica

CLOCK (RATE)

Se utiliza el comando **clockrate** en el modo de configuración específica de interfases seriales para sincronizar las interfases seriales a una velocidad determinada, haciendo esto la interfaz serial se convierte en DCE. Se utiliza el comando **no clock rate** para quitar la sincronización de la interfaz.

Sintaxis

Clock rate *bps*

Speed Asignar velocidad en Bits por segundo: 1200, 2400, 4800, 9600, 19200, 38400, 56000, 64000, 72000, 125000, 148000, 250000, 500000, 800000, 1000000, 1300000, 2000000, 4000000

Modo

Configuración específica de interfaz

CONFIGURE

Se utiliza el comando **configure** en el modo privilegiado para entrar al modo de configuración global. Use la opción **terminal** para una terminal específica.

Sintaxis

Config [*terminal*]

terminal Nombre de la terminal destino

Modo

Privilegiado

CONFIGURE-REGISTER

Se utiliza el comando **config-register** en el modo de configuración global para cambiar los valores del registro de configuración por defecto

Sintaxis

Config-register <0x0-0xFFFFFFFF>

<0x0-0xFFFFFFFF> Numero Hexadecimal de registro de configuración. Por defecto es el 0x2102

Modo

Configuración Global

COPY (RUNNING-CONFIG STARTUP-CONFIG)

Se utiliza el comando **copy running-config startup-config** o en el modo privilegiado para guardar la configuración actual de la RAM (la que se esta ejecutando) en la NVRAM (como la configuración de inicio o copia de respaldo) y viceversa para el segundo caso.

Sintaxis

Copy running-config startup-config

copy startup-config running-config

Modo

Privilegiado

DISABLE

Se utiliza el comando **disable** en el modo privilegiado para salir de este nivel y entrar al modo usuario.

Sintaxis

Disable

Modo

Privilegiado

ENABLE

Se utiliza el comando **enable** para entrar del modo usuario al modo privilegiado.

Sintaxis

Enable

Modo

Usuario y Privilegiado

ENABLE (PASSWORD)

Se utiliza el comando **enable password** en el modo de configuración global para configurar una contraseña de usuario privilegiado no encriptada. Utilice el comando **no enable password** para remover la contraseña.

Sintaxis

Enable password *<password>*

No enable password

password Una cadena de noncase – sensitive entre 4 y 8 caracteres, espacios y puntuación (excepto comillas dobles). Contraseña con espacios en blanco debe ser encerrada en comillas dobles.

Modo

Configuración Global

ENABLE (SECRET)

Se utiliza el comando **enable secret** en el modo de configuración global para configurar contraseñas encriptadas de modo privilegiado. Utilice el comando **no enable secret** para remover la contraseña.

Sintaxis

Enable secret *<password>*

No enable secret

password

Una cadena de noncase – sensitive entre 4 y 8 caracteres, espacios y puntuación (excepto comillas dobles). Contraseña con espacios en blanco debe ser encerrada en comillas dobles.

END

Se utiliza el comando **end** en el modo de configuración de interface (config-if) y del modo de configuración de línea (config-line) para salir de cualquiera de estos modos y entrar al modo privilegiado.

Sintaxis

End

Modo

Todos los modos de configuración

EXIT

Se utiliza el comando **exit** para salir del sistema o de cualquier modo de configuración.

Sintaxis

Exit

Modo

Todos los modos de configuración y Modos Exec

HOSTNAME

Se utiliza el comando **hostname** en el modo de configuración global para asignarle un nombre al sistema. Utilice el comando **no hostname** para quitar el nombre.

Sintaxis

Hostname *name*

No hostname

Name Nombre de sistema entre 1 y 255 caracteres alfanumericos

Modo

Configuración Global

INTERFACE

Se utiliza el comando **interface** en el modo de configuración global para seleccionar un tipo de interfase y entrar a su modo de configuración.

Sintaxis

Interface type *port*

Type Tipo de interface: Ethernet, Serial.

port Número de puerto rango de 1 a 27:

1 a 25 Ethernet (fixed)

26,27 Fast Ethernet (fixed)

Modo

Configuración Global

IP (ADDRESS)

Se utiliza el comando **ip address** en el modo de configuración específica de interfaz para configurar la dirección IP y máscara de subred. Utilice el comando **no ip address** para asignarle la dirección IP y máscara de subred los valores por defecto.

Sintaxis

Ip address *ipaddress mask*

No ip address

Ipaddress Dirección IP

Mask Máscara de subred

Modo

Configuración Global

IP (ROUTE)

Se utiliza el comando **ip route** en el modo de configuración global para definir una ruta hacia una red o subred IP destino.

Sintaxis

Ip route network *[mask / interface] [distance]*

Network red o subred destino

Mask Mascar de subred

Address Direccion IP del router del salto siguiente

Interface Nombre de la interfaz que se debe utilizar para llegar a la red destino

Distance Distancia administrativa

Modo

Privilegiado

NETWORK

Se utiliza el comando `network network_address` en el modo de configuración específica de router para enlistar las redes en los procesos de enrutamiento de los protocolos de enrutamiento RIP e IGRP. Utilice `no network network_address` para eliminar una red.

Sintaxis

`Network network_address`

Modo

Configuración específica de router

PING

Se utiliza el comando **ping** en el modo usuario y privilegiado para enviar mensaje de eco ICMP a una dirección IP o host específico.

Sintaxis

Ping *[ip-address / hostname]*

Ip-address Dirección IP del host

Hostname Nombre del host

Modo

Usuario y privilegiado

ROUTER (IGRP AUTONOMOUS-SYSTEM)

Se utiliza el comando **router igrp autonomous-system** en el modo de configuración global para iniciar el protocolo IGRP como un proceso de enrutamiento IP.

Sintaxis

Router igrp autonomous-system

Autonomous-system Sistema Autonomo que identifica los procesos del router IGRP que compartirán la información de enrutamiento.

Sub commando: Router(config-router)# network network-number

Network-number Especifica una red directamente conectada

Modo

Router igrp autonomous-system – Configuración Global

Network – Configuración específica

ROUTER (RIP)

Se utiliza el comando **router rip** en el modo de configuración global para iniciar el proceso de enrutamiento RIP.

Sintaxis

Router rip

Sub commando: Router(config-router)# network network-number

Network-number Especifica una red directamente conectada

Modo

Router rip – Configuración Global

Network – Configuración específica

SET TRUNK

Se utiliza el comando **set trunk** en el modo de configuración global para configurar al switch al modo trunking y cualquier modo de negociación. También identifica las VLANs que van a ser transportadas sobre el trunk link.

Sintaxis

Set trunk *port [on / off / desirable / auto / nonegotiate] vlan-range [ISL]*

port: Puerto del switch

On: Este establece al puerto en modo trunking permanentemente

Off: Este establece al puerto en modo de no-trunking permanentemente

Desirable: Seleccionando este puerto activamente convertirá el enlace en modo trunking

Auto: Para que el puerto este dispuesto a convertir el enlace en modo trunking

Nonegotiate: El puerto es colocado en modo trunking permanentemente, pero no son generados frames DTP para la negociación

Vlan-range: Para configurar el rango de VLANs que van a pasar por el trunk link. Por default transporta todas las VLANs (1-1000)

ISL: Las VLANs son etiquetadas por encapsulación en cada frame usando el protocolo ISL de Cisco. Este protocolo esta por defecto, sino se especifica el valor

Modo

Configuración global del switch

SET VLAN

Se utiliza el comando **set vlan** *vlan-num* en el modo de configuración global para crear o configurar en el switch LAN Virtuales.

Sintaxis

Set vlan *vlan-num* [*name-name*]

Vlan-num: Para crear el numero de la VLAN

[name-name]: Para asignarle un nombre descriptivo

Modo

Configuración Global del switch

SET VLAN VLAN-NUM

Se utiliza el comando **set vlan** *vlan-num port-list* en el modo de configuración global para asignar una VLAN ya numerada a uno o mas puertos del switch, identificados con el número de modulo del switch. Por ejemplo el comando set vlan 101 3/1, 3/3, 3/4, 3/5, 3/6 y 3/7 a la VLAN 101.

Sintaxis

set vlan *vlan-num mod-num/port-list*

vlan_num: Número asignado a la vlan

Modo

Configuración global del switch

SHOW (FLASH)

Se utiliza el comando **show flash** en los modos usuario y privilegiado para visualizar la información acerca del dispositivo de memoria flash.

Sintaxis

Show flash

Modo

Usuario y privilegiado

SHOW (HISTORY)

Se utiliza el comando **show history** en modo usuario y privilegiado para mostrar los comandos usados en esa sesión.

Sintaxis

Show history

Modo

Usuario

SHOW (INTERFACES)

Se utiliza el comando **show interfaces** en modo privilegiado para mostrar las estadísticas y estado de todas las interfaces del dispositivo.

Sintaxis

Show interfaces [*type*] [basic / secondary]

<i>Type</i>	Tipo de interfase
	Serial – Estadísticas y estado para las interfases seriales
	Ethernet - Estadísticas y estado para las interfases ethernet
	Fastethernet - Estadísticas y estado para las interfases FastEthernet

Modo

Privilegiado

SHOW (IP)

Se utiliza el comando **show ip** en el modo usuario para mostrar la información global del Protocolo Internet (IP).

Sintaxis

Show ip

Modo

Usuario

SHOW (IP INTERFACE)

Se utiliza el comando **show ip interface** en los modos usuario y privilegiado para listar un resumen de la información y estado de una interfaz IP.

Sintaxis

Show ip interface [*brief*] [*type*] [*number*]

[<i>brief</i>]	Muestra un resumen del estado y configuración de todas las interfaces
------------------	---

<i>[type]</i>	Muestra la información de una interfaz específica por ejemplo (Serial o Ethernet).
<i>[Number]</i>	Numero de la interfaz

SHOW (IP PROTOCOLS)

Se utiliza el comando **show ip protocols** en los modos usuario y privilegiado para ver los protocolos de enrutamientos activos.

Sintaxis

Show ip protocols

Modo

Usuario y privilegiado

SHOW (IP ROUTE)

Se utiliza el comando **show ip route** en los modos usuario y privilegiado para visualizar las entradas de la tabla de enrutamiento.

Sintaxis

Show ip route

Modo

Usuario y privilegiado

SHOW (MAC-ADDRESS-TABLE)

Se utiliza el comando **show mac-address-table** en el modo privilegiado para mostrar las direcciones que están en la tabla MAC address por los puerto o módulo conmutados.

Sintaxis

Show mac-address-table [permanent / dynamic / restricted static] [address *mac-address*]

Modo

Privilegiado

SHOW (PROTOCOLS)

Se utiliza el comando **show protocols** en los modos usuario y privilegiado para mostrar los protocolos configurados en el router. Este comando muestra el estado global y específico para la interfaz de cualquier protocolo de capa 3 configurado.

Sintaxis

Show protocols

Modo

Usuario y Privilegiado

SHOW (RUNNING-CONFIG)

Se utiliza el comando **show running-config** en el modo privilegiado para mostrar la configuración que se está ejecutando.

Sintaxis

Show running-config

Modo

Privilegiado

SHOW TRUNK

Se utiliza el comando **show trunk** en el modo privilegiado para ver y verificar la configuración de trunk en un switch.

Sintaxis

Show trunk *port*

port: Puerto del switch

Modo

Privilegiado del switch

SHOW (VERSION)

Se utiliza el comando **show version** en el modo de usuario y privilegiado para mostrar la información hardware básico y la versión de firmware.

Sintaxis

Show versión

Modo

Usuario y privilegiado

SHOW (VLAN)

Se utiliza el comando **show vlan** en el modo privilegiado para mostrar la configuración de los parámetros de VLAN.

Sintaxis

Show vlan [*vlan*]

Vlan Número desde 1 a 1005

Modo

Privilegiado

SHOW (VLAN-MEMBERSHIP)

Se utiliza el comando **show vlan-membership** en el modo privilegiado para mostrar las asignaciones VLAN y tipo de membresía para todos los puertos del switch.

Sintaxis

Show vlan-membership

Modo

Privilegiado

SHUTDOWN

Se utiliza el comando **shutdown** en el modo de configuración de interfaz para deshabilitar un interfase. Utilice el comando **no shutdown** para reiniciar la interfase deshabilitada.

Sintaxis

Shutdown

No shutdown

Modo

Configuración de Interfase

TRACERT

Se utiliza el comando **tracert** en modo de usuario o privilegiado en el router y en la consola del host para rastrear la ruta hacia una dirección destino. Mostrando cada uno de los saltos que el paquete tiene que pasar.

Sintaxis

Tracert IP Addresss

IP Address Dirección IP Destino

Modo

Usuario y Privilegiado

1.6. ESTRUCTURA DE ARCHIVOS

Estructura de los archivos:

El archivo de almacenamiento esta dividido en secciones. Cada sección contiene claves y las claves tienen valores. Los nombres de las secciones se escriben entre corchetes. *Los nombres de las claves pueden nombrarse con cualquier carácter, excepto !@#\$%^=, etc., ni espacios en blanco.*

El nombre de cada sección es dado de la siguiente manera:

- Si el objeto dentro del programa es un Host, el nombre de la sección está formado por la dirección MAC del Host, por ésta identifica de manera única al Host de los demás Hosts.
- Si el objeto es un Hub, router o switch, el nombre de la sección está formado por el SerialNumber que identifica de manera única al Hub, router o switch de los demás elementos.

Los nombres de las claves están separados de su valor por el signo '=' y que por lo tanto todo lo que se encuentre a su derecha será considerado el valor de esa clave.

Estructura de las secciones según el elemento

Para un Host:

[DIRECCION MAC]	Inicio de la sección
Type=valor	Tipo de dispositivo, para este caso, PC.
Top=valor	Coordenadas en la pantalla.
Left=valor	
IpAddress=valor	Dirección Ip del host.
Mask=valor	Máscara de subred del host.
Gateway=valor	Puerta de enlace del host.

Para un Switch:

[DIRECCION MAC]	Inicio de la sección
Type=valor	Tipo de dispositivo, para este caso, SWITCH.
Top=valor	Coordenadas en la pantalla.
Left=valor	
Hostname=valor	Nombre de host del Switch.
Puerto1=valor	El valor de esta clave es el número de la vlan al que esta configurado el puerto.
Puerto2=valor	
Puerto3=valor	
Puerto4=valor	
Trunk=valor	Puerto utilizado como Trunk.

Para un Router:

[DIRECCION MAC]	Inicio de la sección
Type=valor	Tipo de dispositivo, para este caso, ROUTER.
Top=valor	Coordenadas en la pantalla.
Left=valor	
Hostname=valor	Nombre de host del Router.
Password=valor	Contraseña administrativa del router.
Secret=valor	Contraseña encriptada.
Ethernet0=valor	Los valores son los identificadores de los objetos que están conectados a las interfaces.
Ethernet1=valor	
Serial0=valor	
Serial1=valor	
Igrp=valor	True o False. Indica si se esta usando IGRP
Rip	True o False. Indica si se esta usando RIP.
Config-Register	

Para un Hub:

[SERIALNUMBER]	Inicio de la sección
Type=valor	Tipo de dispositivo, para este caso, HUB.
Top=valor	Coordenadas en la pantalla
Left=valor	
SerialNumer=valor	Serial Number del hub que lo identifica.

Para un cable:

[SERIALNUMBER]	Inicio de la sección
Type=valor	Tipo de elemento, para este caso, un MEDIO (cable).
X1=valor	Coordenada de inicio en X de la Línea.
Y1=valor	Coordenada de inicio en Y de la Línea.
X2=valor	Coordenada final en X de la Línea.
Y2=valor	Coordenada final en Y de la Línea.
Conexión=valor	Contiene los identificadores de los elementos a los que conecta la línea.

2. DESARROLLO DEL SISTEMA

2.1. COMPONENTES

2.1.1. VENTANA PRINCIPAL

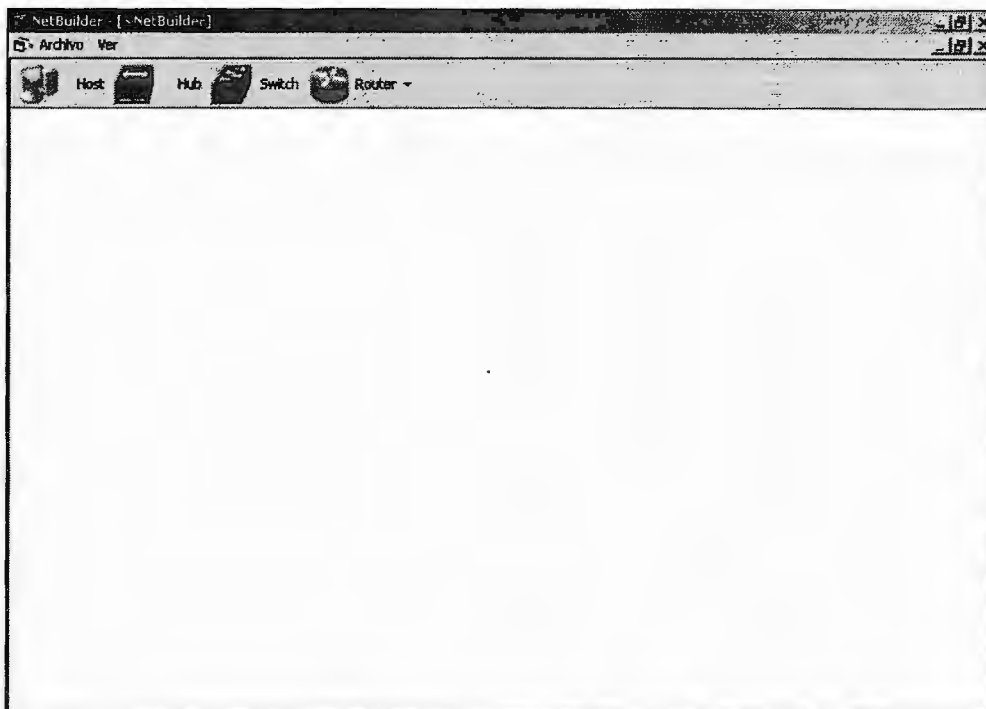
Ventana inicial en la cual se crean los diseños de red antes explicados.

Componentes:

Un formulario y 5 botones de comando.

Los botones de comandos son los que crean los elementos como Enrutadores, Switches, Hub o Host haciendo una copia de la clase antes definida para cada uno de estos, y hasta que se le da clic se crea la imagen en el formulario

El formulario detecta y guarda las posiciones en las cuales se coloca los elementos a agregar.

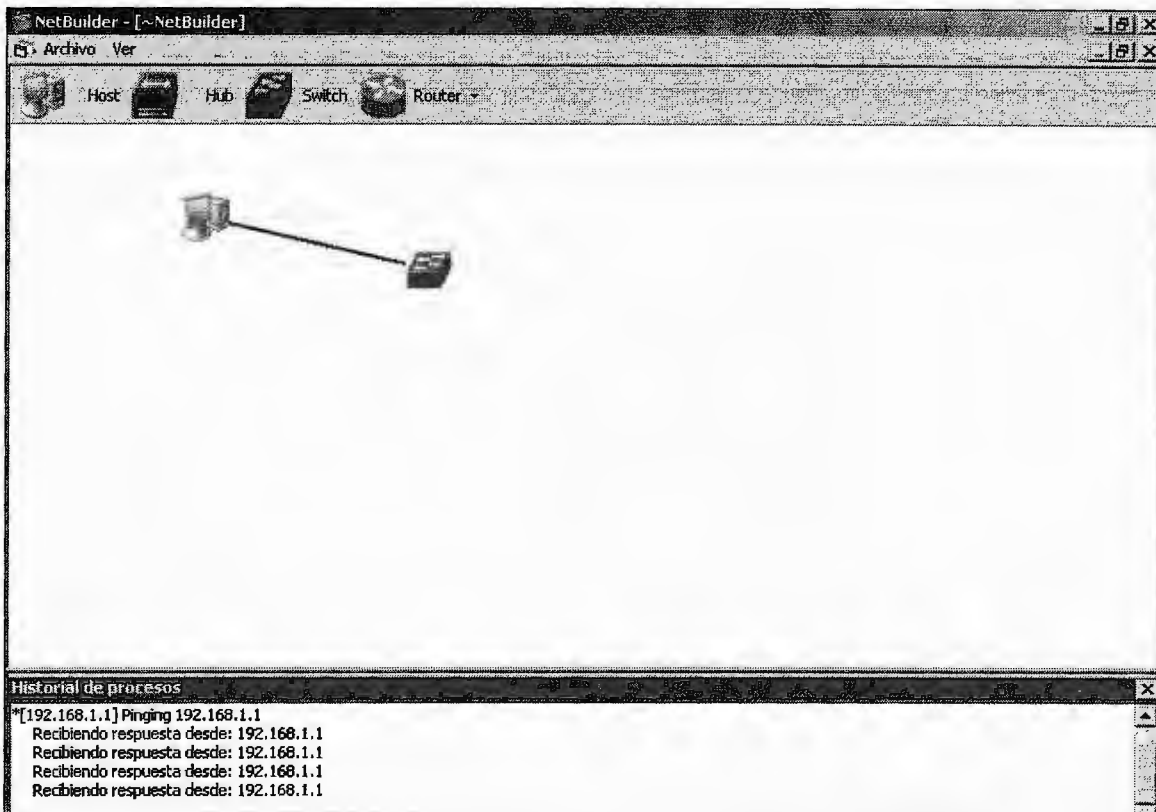


2.1.2. HISTORIAL

Consola que lleva el registro de los procesos más relevantes al momento de configurar como también al momento de la comunicación entre dispositivos.

Procesos de los que lleva Bitácora:

- a. Resolución de direcciones MAC
- b. Proceso del Ping
- c. Configuración de Trunk en el Switch
- d. Configuración de Vlans
- e. Configuración de interfaces del router
- f. Procesos internos del router
- g. Cableado entre dispositivos



2.1.3. CONFIGURACIÓN DE IP DEL HOST

Pantalla en la que se le configura los siguientes parámetros al host:

- h. Dirección Ip
- i. Mascara de Red
- j. Default Gateway

Cuando se ingresan los parámetros se los asigna a las propiedades del elemento creado con los botones de comando, pero antes hace las siguientes validaciones:

Dirección IP

- Los bits del Host ID, no pueden ser todos 0s, ni todos 1s.
- El primer octeto tiene que ser un valor entre 1-223
- Una dirección ip no puede tener el mismo valor de la red a la que esta pertenece.

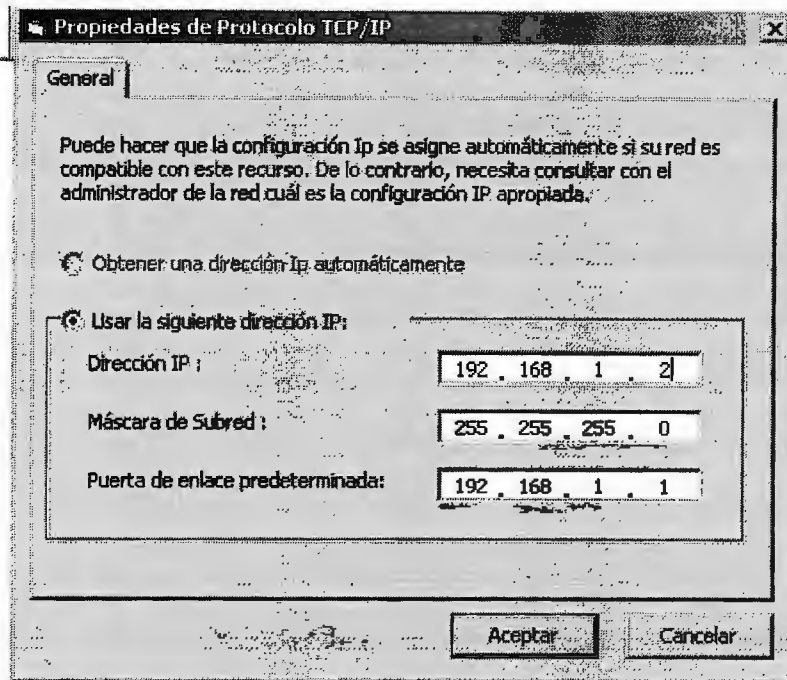
Mascara de Red

- Las mascara de red se compone de 1s consecutivos y 0s consecutivos. Debe comenzar con 1s de izquierda a derecha y al poner un cero todos los demás valores a la derecha deben de ser 0s.
- Valores que se pueden utilizar en el primer octeto:
 - 224
 - 240
 - 248
 - 252
 - 254
 - 255

- Valores que se pueden utilizar en el primer octeto:
 - 128
 - 192
 - 224
 - 240
 - 248
 - 252
 - 254
 - 255
- Valores que se pueden utilizar en el cuarto octeto:
 - 128
 - 192
 - 224
 - 240
 - 248
 - 252

Default Gateway

Las mismas validaciones que la Dirección Ip.



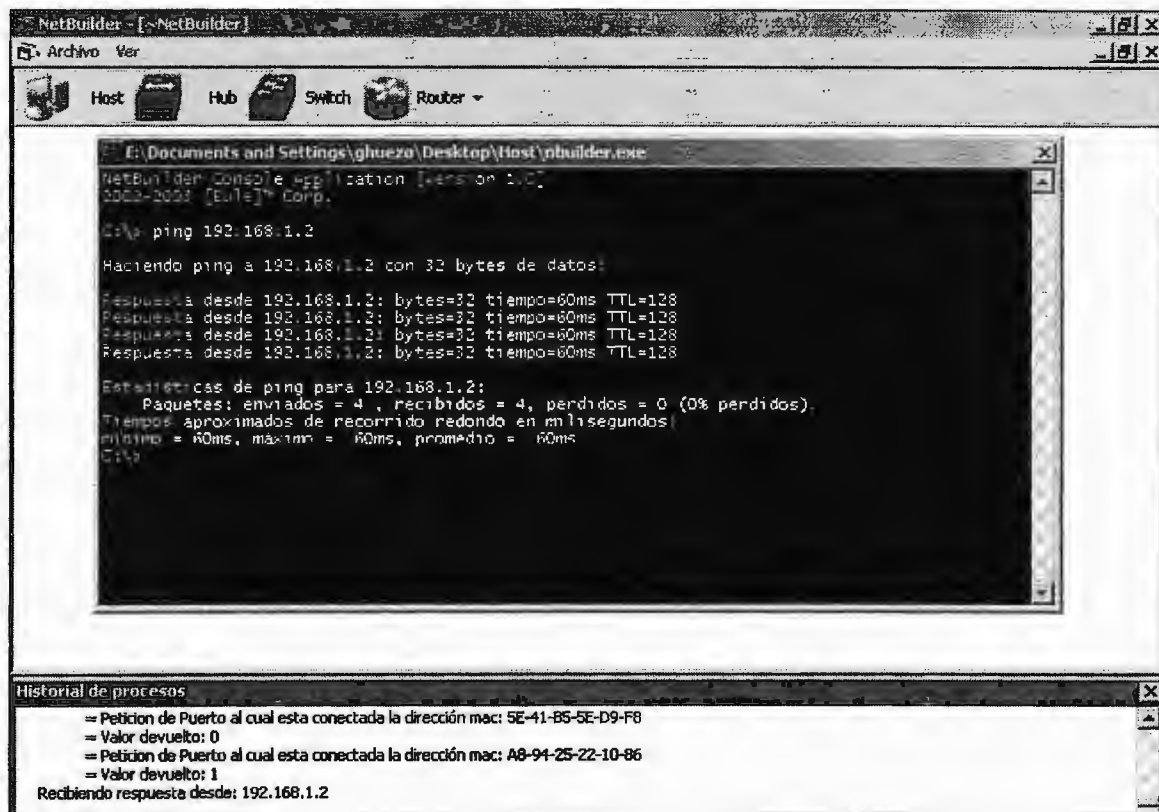
2.1.4. CONSOLA DEL HOST

Consola creada de un formulario y una caja de texto.

Ya se tiene un tamaño fijo del cual ya no puede retroceder mas; los comandos los interpreta dividiendo lo escrito cada vez que encuentra un espacio y compara los comando, para saber que acción debe realizar.

Los comandos que se pueden realizar desde esta consola son:

- Ping
- Tracert



2.1.5. CONSOLA DEL SWITCH

Consola en la cual se configura a nivel de comandos el Switch, se compone de un formulario y una caja de texto, con la misma metodología de interpretación de comandos que la consola de host.

Desde esta consola se configuran propiedades del Switch y de sus respectivos puertos.

Propiedades configurables desde dicha consola:

Vlan

Trunk

Password

Hostname

También se puede visualizar las propiedades del switch, mediante la lectura de las propiedades configuradas en cada switch, así como también los mensajes ya establecidos, los cuales son almacenados en archivos de texto.

Dicha configuración de cada propiedad del switch es almacenada en archivos de texto, para poder volver a cargar esa configuración ya hecha.


```
[SWITCH]
PCA Number: 73-3122-04
PCA Serial Number: FAB0337240K
Model Number: WS-C1912-A
System Serial Number: 4696BC42884F
Power Supply S/N: PHI031803JK
PCB Serial Number: FAB0337240K,73-3122-04
-----
1 user(s) now active on Management Console.
User Interface Menu
  [M] Menus
  [K] Command Line
  [I] IP Configuration
Enter Selection: k
Switch> enable

Switch# configure terminal

Switch(config)# exit
Switch#
```

2.1.6. CONSOLA DEL ROUTER

Consola en la cual se configura a nivel de comandos el Router, se compone de un formulario y una caja de texto, con la misma metodología de interpretación de comandos que la consola de host.

Desde esta consola se configuran propiedades del Router y de sus interfaces, tanto seriales como Ethernet.

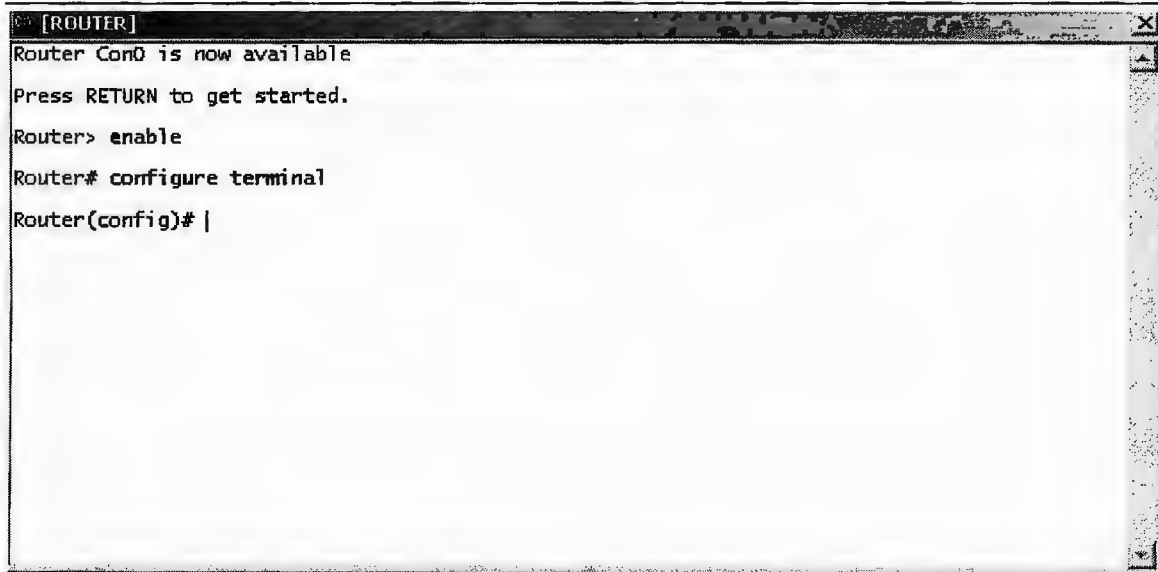
Propiedades configurables desde dicha consola:

- Dirección Ip de Interfaces
- Mascara de Red
- Configuración de registro
- Hostname
- Protocolo de enrutamiento
- Habilitar interfaces
- Secret y Password

También se puede ver la siguiente información acerca del router:

- Configuración y Estado de interfaces
- Estado de protocolo IP
- Tablas de ruteo
- Estado de los procesos de protocolos de enrutamiento
- Versión de software, configuración del sistema de hardware

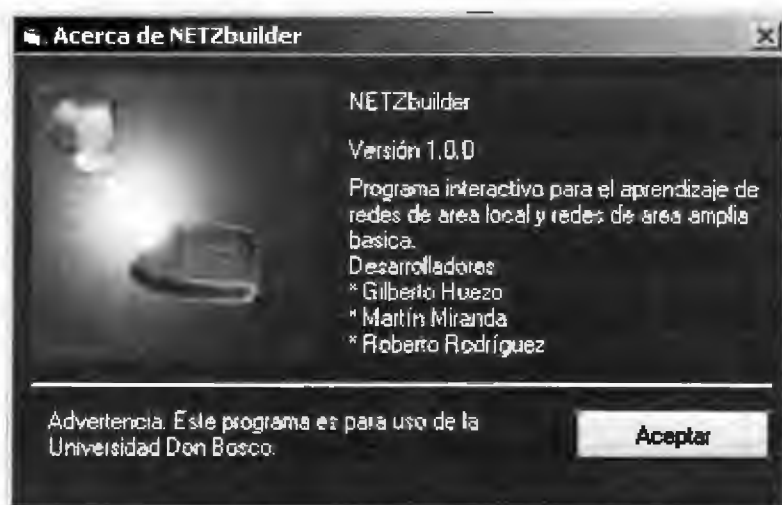
Dicha configuración de cada propiedad del router es almacenada en archivos de texto, para poder volver a cargar esa configuración ya hecha.



```
[ROUTER]
Router Con0 is now available
Press RETURN to get started.
Router> enable
Router# configure terminal
Router(config)# |
```

2.1.7. ACERCA DE...

Formulario para información acerca de la versión del programa



2.2. MENUS

Archivo

Nuevo: Crea una nueva ventana de diseño

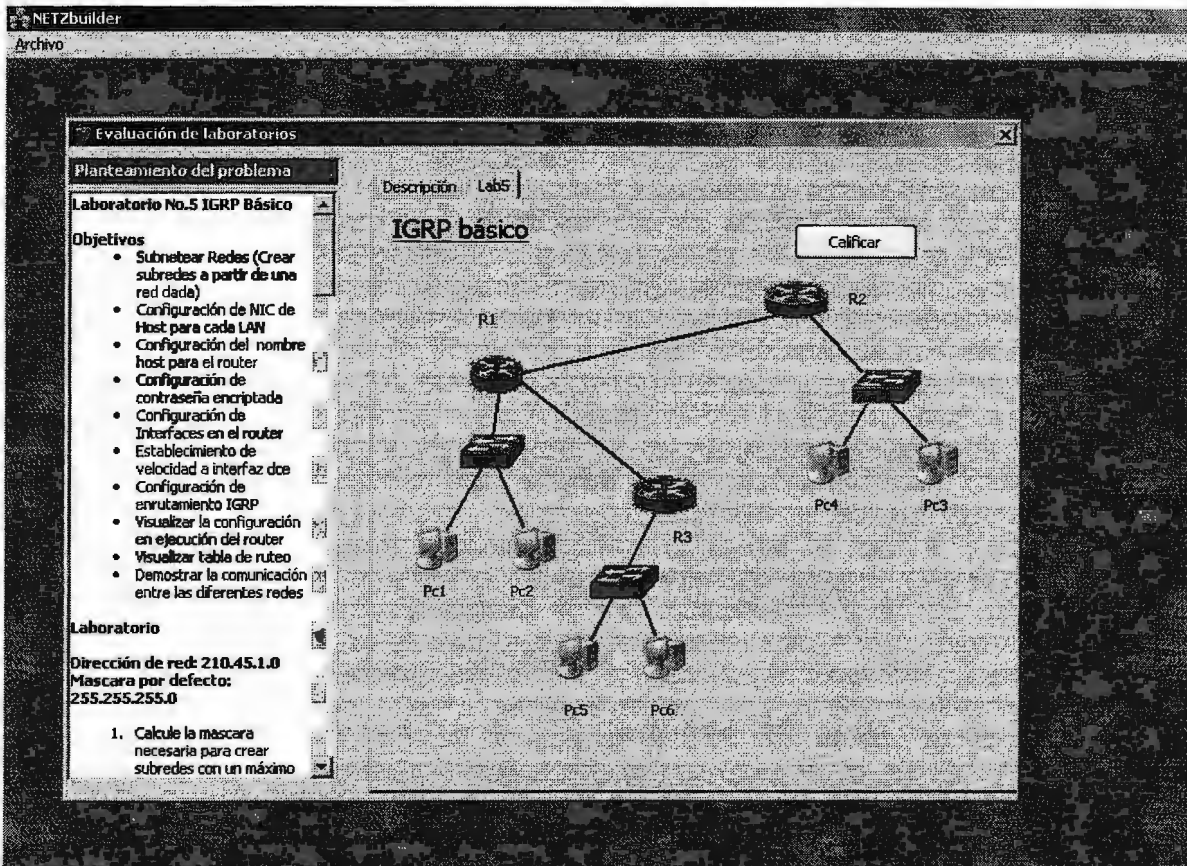
Abrir: Visualizar diseños antes hechos y que fueron guardados, para su posterior seguimiento; guarda la configuración de cada dispositivo y de cada conexión entre ellos.

Guardar: Guarda los diseños ya creados para poder seguir trabajando en este, o para revisión.

Imprimir: Permite imprimir el diseño grafico de la red, así como también algunas configuraciones de cada dispositivo.

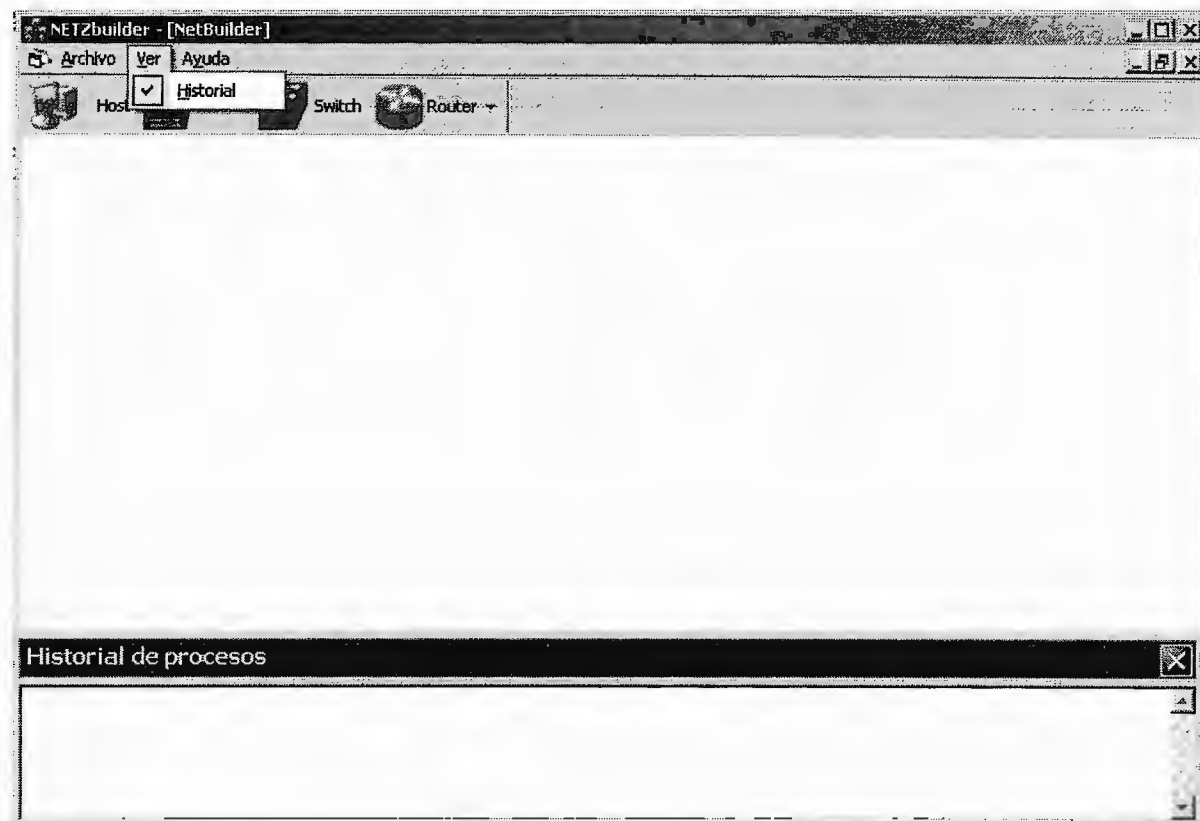
Evaluaciones

Muestra laboratorios prediseñados.



Ver

Barra de Historial: Opción para mostrar u ocultar la barra de Historial



2.3. DISPOSITIVOS

1. HOST

Los Host son representados por una clase la cual tiene una propiedad que representa la Tarjeta de Red.

Posee 3 Funciones:

- **Scanla:** realiza la búsqueda de una IP en el ARP local.
- **Ping:** realiza el proceso de hacer ping.
- **SetupNIC:** configura una NIC con una dirección IP, máscara de subred y una puerta de enlace

2. HUB

Son representados por una Clase dentro del programa:

Propiedades:

- **Jackets:** variable de tipo Collection que representa los puertos del Hub.
- **cablesIn y cablesOut:** mantienen referencias a las líneas que representan el medio de conectividad.
- **SerialNumber:** es un número aleatorio que identifica al hub dentro del programa.

Funciones y métodos:

- **Link_plug:** realiza el proceso vinculación de una PC a un puerto del hub.
- **Link_unplug:** desvincula un host de un puerto del hub.
- **Implementa la clase link.cls para usar los métodos Link_scan, Link_send y link_route.**
- **newSerial:** genera un número de serie aleatorio.

3. SWITCH

Son representados por una Clase dentro del programa:

Propiedades:

- **Jackets:** arreglo del tipo Tport, que representa los puertos del switch.
- **ArpTable:** maneja la tabla ARP del Switch.
- **cablesIn y cablesOut:** mantienen referencias a las líneas que representan el medio de conectividad.
- **SerialNumber:** es un número aleatorio que identifica al hub dentro del programa.
- **Hostname:** representa el nombre del switch. Este puede ser cambiado mediante la consola de comandos.
- **password:** contraseña dada al switch.
- **MTrunker:** contiene el número del puerto configurado para el trunk.

Funciones y métodos:

- Implementa la clase link.cls para usar los métodos Link_scan, Link_send, link_route y link_unplug.
- **PlugSwitch:** crea una referencia a otro objeto switch.
- **UnplugSwitch:** borra la referencia al objeto switch.
- **TrunckedScan:** hace la búsqueda de un dispositivo en el switch conectado.
- **TrunckedSend:** envía mensajes a un dispositivo en el lado del switch conectado.
- **Setvlan:** crea una vlan.
- **setvlan2:** establece el puerto a una vlan.
- **ShowvLan:** devuelve información acerca de todas las vlans creadas.
- **show_vlan_membership:** devuelve la información de membresía de las vlans creadas (que puertos pertenecen a una vlan)
- **ShowTable:** devuelve información de la tabla MAC del switch.
- **ShowInterface:** devuelve la información de las interfaces del switch

4. ROUTER

Son representados por una Clase dentro del programa:

Propiedades:

- Ethernet0: representa una interfaz ethernet (de tipo routenic).
- Ethernet1: representa una interfaz ethernet.
- Serial0: representa una interfaz serial (de tipo serialnic).
- Serial1: representa una interfaz serial.
- Model: indica el modelo que se esta usando, ya sea el modelo 2501 o el 2514.
- Secret: devuelve o establece la contraseña secreta.
- Autonomous_system: representa el sistema autonomo del protocolo IGRP.
- Code, net, subnetmask, metric, defaultgateway y ninterface: representa las propiedades de la tabla de ruteo.

Métodos:

- Riptmr: representa timer de RIP
- Igrptmr: representa timer de IGRP
- addRipNetwork: agrega una red usando el protocolo RIP.
- addIgrpNetwork: agrega una red usando el protocolo IGRP.
- update: Simula el proceso de actualización de una tabla de ruteo.
- Implementa la clase link.cls para usar los métodos Link_scan, Link_send y link_route.
- Link_unplug: desconecta un dispositivo de una interfaz.
- Anding: realiza el proceso de comprobación ANDing.
-

Representa la interfaz ethernet de un router.

Propiedades:

- NIC_mask: implementación de la clase NIC.cls. Es utilizada como máscara de subred de la interfaz.
- NIC_MacAddress: implementación de la clase NIC.cls. Es utilizada como dirección mac de la interfaz.

- **NIC_IPAddress**: implementación de la clase **NIC.cls**. Es utilizada como dirección IP de la interfaz.
- **NIC_gateway**: implementación de la clase **NIC.cls**. Es utilizada como puerta de enlace predeterminada en un router.

Métodos:

- Implementa los métodos **NIC_scan**, **NIC_send** y **NIC_route** de la clase **NIC.cls**.
- **newMac**: genera una nueva y única dirección MAC para la interfaz.
- **Anding**: realiza el proceso de comprobación ANDing.

Representa la interfaz serial de un router.

Propiedades:

- Implementa las propiedades **NIC_mask**, **NIC_MacAddress**, **NIC_IPAddress** de la clase **NIC.cls**.

Métodos:

- Implementa los métodos **NIC_scan**, **NIC_send** y **NIC_route** de la clase **NIC.cls**.
- **newMac**: genera una nueva y única dirección MAC para la interfaz.
- **Anding**: realiza el proceso de comprobación ANDing.

2.4. LABORATORIOS

Los Laboratorios son prediseñados, contienen una parte en la cual se describe los pasos a realizar en dicho laboratorio y otra que contiene el diseño grafico, en el cual se tendrá que realizar las configuraciones respectivas a los diferentes dispositivos que así lo pida el laboratorio.

Al terminar el laboratorio, este será evaluado y mostrara un porcentaje de dicha configuración.

Laboratorio No.1 Router Básico

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Conocer los diferentes modos del router (Usuario, Privilegiado, Configuración Global y Configuración Especifica)
- Familiarizarse con el uso de la ayuda en el Router
- Configuración de NIC de Host
- Revisión de Información variada en el Router
- Configuración del nombre host para el router
- Configuración de contraseña encriptada y sin encriptación
- Configuración de Interfaces en el router
- Visualizar la configuración en ejecución del router
- Guardar la configuración almacenada en la NVRAM del router
- Demostrar la comunicación entre el host y la interfaz del router

Laboratorio No.2 Router y LAN

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Configuración de NIC de Host para cada LAN
- Configuración del nombre host para el router
- Configuración de contraseña encriptada
- Configuración de Interfaces en el router
- Visualizar la configuración en ejecución del router
- Visualizar la información de las interfaz Ethernet
- Visualizar tabla de ruteo
- Guardar la configuración almacenada en la NVRAM del router
- Visualizar la tabla MAC del switch
- Demostrar la comunicación entre host de la misma subred y diferentes subredes

Laboratorio No.3 Router y WAN

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Configuración del nombre host para el router
- Configuración de contraseña encriptada
- Configuración de Interfaces en el router
- Configuración de rutas estáticas
- Configuración de protocolo de enrutamiento RIP
- Visualizar la configuración en ejecución del router
- Visualizar la información de las interfaz Ethernet y Seriales
- Visualizar tabla de ruteo
- Guardar la configuración almacenada en la NVRAM del router
- Demostrar la comunicación entre las diferentes redes
- Uso de la herramienta Trace

Laboratorio No.4 RIP Y LAN

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Configuración de NIC de Host para cada LAN
- Configuración del nombre host para el router
- Configuración de contraseña encriptada
- Configuración de Interfaces en el router
- Configuración de enrutamiento RIP
- Visualizar la configuración en ejecución del router
- Visualizar la información de las interfaz Ethernet y Seriales
- Visualizar el estado del protocolo de enrutamiento activo
- Visualizar el estado global y específico de las Interfaz del protocolo enrutado activo
- Visualizar tabla de ruteo
- Guardar la configuración almacenada en la NVRAM del router
- Demostrar la comunicación entre las diferentes redes

Laboratorio No.5 IGRP Básico

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Configuración de NIC de Host para cada LAN
- Configuración del nombre host para el router
- Configuración de contraseña encriptada
- Configuración de Interfaces en el router
- Configuración de enrutamiento IGRP
- Visualizar la configuración en ejecución del router
- Visualizar tabla de ruteo
- Guardar la configuración almacenada en la NVRAM del router
- Demostrar la comunicación entre las diferentes redes

Laboratorio No.6 IGRP Avanzado

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Configuración de NIC de Host para cada LAN
- Configuración del nombre host para el router
- Configuración de contraseña encriptada
- Configuración de Interfaces en el router
- Configuración de enrutamiento IGRP
- Visualizar la configuración en ejecución del router
- Visualizar la información de las interfaz Ethernet y Seriales
- Visualizar el estado del protocolo de enrutamiento activo
- Visualizar el estado global y específico de las Interfaz del protocolo enrutado activo
- Visualizar tabla de ruteo
- Uso de herramienta tracert
- Guardar la configuración almacenada en la NVRAM del router
- Demostrar la comunicación entre las diferentes redes

Laboratorio No. 7 VLAN Básico

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Configuración de Interfaz de cada Host
- Configuración básica de switch (Nombre de Switch y Contraseña de acceso)
- Verificación de Tabla de MAC del switch
- Configuración básica de VLAN
- Demostrar la comunicación de VLANs
- Guardar configuración en NVRAM
- Demostrar que las VLANs son dominios de Broadcast diferentes

Laboratorio No. 8 VLAN Avanzado

Objetivos

- Subnetear Redes (Crear subredes a partir de una red dada)
- Configuración de NIC de Host para cada LAN
- Configuración de Interfaces en el router
- Configuración básica de switch (Nombre de Switch y Contraseña de acceso)
- Verificación de Tabla de MAC del switch
- Configuración básica de VLAN
- Demostrar la comunicación de VLANs
- Guardar configuración en NVRAM
- Demostrar que las VLANs son dominios de Broadcast diferentes
- Demostrar que dos VLAN diferentes o dominios de Broadcast diferentes se pueden comunicar mediante un dispositivo de capa 3 (Router)

PROCESOS

- **Resolución de Direcciones MAC:**

Una computadora conectada a un dispositivo como un Switch, Hub o hasta el mismo router, cuando se quiere comunicar con otro dispositivo que tenga una dirección Ip.

Los pasos que realiza son los siguientes:

- ✓ Fijar su dirección Mac en tu tabla Arp
- ✓ Preguntar en todos los puertos del Switch, del Hub o directamente a la interfase del router, respecto a que dispositivo tiene la Dirección IP que anda buscando.
- ✓ Al detectar un dispositivo que este tiene la id que se anda buscando, agrega las Direcciones IP y MAC del dispositivo que la busca, a su tabla de Arp Local.
- ✓ Envía una respuesta al Dispositivo origen, proporcionándole su MAC, la cual el dispositivo origen hará lo mismo y la guardara en su tabla ARP Local.

- **Ingreso de Direcciones MAC en las Tablas de Switch**

Cuando el proceso anterior se lleva a cabo con dispositivos conectados a un Switch, este va ingresando en que puerto esta conectado cada dispositivo con su respectiva MAC.

- **Validación de Direcciones IP**

Al ingresar la dirección IP de un Host, este valida si la dirección no esta siendo utilizada por otro dispositivos y si cumple con los requerimientos antes mencionados. Ver Pág. 200

- **Validación de Direcciones de la Mascara**

Al ingresar la dirección de la mascara de red de un host, este evalúa si pertenece al rango de valores validos antes mencionados. Ver Pág.200

- **Configuración de Trunking entre Switch**

Configuración hecha para poder habilitar que dos switch transmitan toda la información de uno hacia otro a través de un mismo puerto, no importando a que VLAN pertenezcan los puertos a los que estén conectados los host.

Para eso se debe configurar a un puerto específico en cada switch para que este sea ocupado para trunking y luego cablear dichos switch, al cablear este automáticamente evalúa si hay un puerto en cada extremo configurado para trunk, de no ser así, muestra un error, el cual indica que se deben configurar dichos puertos antes de cablear.

- **Configuración de Vlans**

En cada switch se podrán configurar Vlans, con valores entre 1-1001, al tratar de comunicarse dos computadoras conectadas a un switch, examinarán a que vlan pertenece cada puerto en los que estén conectadas dichas computadoras, de no pertenecer a la misma inmediatamente responderá que no se pueden comunicar, y solo lo permitirá cuando pertenezcan a la misma.

- **Configuración de Password**

A los únicos dispositivos que se les podrá asignar una contraseña para que sea solicitada al ingresar son: El Switch y el Router.

Será solicitada en la consola respectiva de cada dispositivo, al cambiar hacia modo Privilegiado.

- **Actualizaciones de Tablas de Ruteo**

En este proceso el router realiza dos tareas, Escuchar las actualizaciones de otros routers, y la otra mandar sus actualizaciones a otros. Cada router posee Timers, los cuales medirán la frecuencia con la cual se enviarán las actualizaciones respectivas, y también cuanto tiempo permanecerá una ruta en cada router sin ser actualizada.

Los tiempos son los siguientes:

RIP

- ✓ Actualizar: 30 seg.
- ✓ Invalida: 180 seg.
- ✓ Expirar: 240 seg.

IGRP

- ✓ Actualizar: 90 seg.
- ✓ Invalida: 270 seg.
- ✓ Expirar: 630 seg.

Cuando una ruta se vuelva inválida, ya no es tomada en cuenta al mandar las actualizaciones, y cuando expira es borrada de la tabla.

- **Habilitación de Interfaces de Routers**

Para iniciar el funcionamiento del router se deben habilitar administrativamente las interfaces que se van a utilizar, esto se realiza a través de consola.

- **Publicación de Redes**

Para poder publicar redes se realiza antes el proceso de habilitar los protocolos de enrutamiento, en este caso solo se pueden habilitar RIP e IGRP, a través de la consola.

Después, también a nivel de comandos, se realiza la publicación de las redes directamente conectadas en el router hacia los demás routers.

- **Selección de mejor Ruta**

El router realiza un proceso interno, en el cual elige la mejor ruta para llegar a un destino.

Los parámetros que considera son los siguientes en su orden respectivo:

- ✓ Distancia Administrativa

Directamente Conectada (C): 0

Estática (S): 1

IGRP (I): 100

RIP(R): 120

Si hay varias rutas con la misma distancia administrativa, evalúa según la métrica.

✓ Métrica

RIP: La calcula en base a cuantos saltos realiza para llegar a un destino. (HOPCOUNT)

IGRP: La calcula en base a la siguiente formula

$$\text{Métrica} = [K1 * \text{Bandwidth} + (K2 * \text{Bandwidth}) / (256 - \text{load}) + K3 * \text{Delay}] * [K5 / (\text{reliability} + K4)]$$

Los valores de las constantes son $K1 = K3 = 1$ y $K2 = K4 = K5 = 0$.

Debido a que las redes de este sistema son homogéneas, también ocupa el HOPCOUNT para elegir la mejor ruta.

CONCLUSIONES

- El proyecto planteado facilitara el aprendizaje de un usuario en lo que respecta al diseño de una red LAN sin tener que exponerse a deteriorar o arruinar un dispositivo de cuantioso valor.
- El Proyecto permitirá diseñar una red LAN de manera personalizada, es decir no estará ya establecida y solo de configurar como lo hace el actual simulador.
- El Proyecto se realizará con suficiente documentación para poder ser mejorado, según lo requieran las exigencias en el futuro.
- Se podrán realizar comprobaciones de conectividad entre los dispositivos que formen parte del diseño de la red mediante utilitarios antes especificados.
- Se podrán llevar a cabo evaluaciones, ya pre-diseñas, sobre temas específicos.
- Tanto el diagrama en pantalla como la configuración de cada dispositivo, podrá ser impresa para la comodidad del usuario.
- Con la versión impresa de los diagramas, el usuario podrá analizar con más detalle su trabajo. Además de tener un respaldo de su trabajo.
- La versión impresa puede servir como base para implementaciones reales en un futuro.
- El proyecto tendrá disponibles 8 laboratorios evaluados por el simulador para que el estudiante pueda medir sus conocimientos y habilidades sobre redes.
- El proyecto soporta una serie de comandos de router y switch que sirven para la configuración y administración de cada dispositivo.
- Se podrá visualizar un historial de lo que ocurre en los procesos de enrutamiento interno de los routers.

BIBLIOGRAFIA

Libros de Texto

- Análisis y Diseño de Sistemas de Información, James A. Senn, Segunda Edición.
- Conceptos y métodos en la simulación digital de eventos discretos, Fishman G. S., Limusa.
- Teoría General de Sistemas, Angel A. Sarabia, Publicaciones de Ingeniería de Sistemas, ISDEFE, 1995.
- Course Implementing a Microsoft 2000 Network Infrastructure - Microsoft Official Curriculum, E. Microsoft Corporation, Colombia, 2000.
- Course Internetworking with Microsoft TCP/IP on Microsoft Windows NT 4.0 – Microsoft Official Curriculum, E. Microsoft Corporation, Colombia, 2000.
- Computer Professional's Dictionary, Allen L. Wyatt, Osborne McGraw-Hill
- CCNP Routing Study Guide, Todd Lammle, Sean Odom, Kevin Wallace, Sybex, Inc
- Cisco CCNP Switching Exam Certification Guide, Tim Boyles, Dave Hucaby, Cisco Press

Sitios Web

URL	Nombre de Empresa	Giro	Pais	Año
www.cisco.com	Cisco Systems, Inc	Comercio/Servicio/Educacion	USA	1984
www.outsights.com	OUTSIGHTS, INC.	Servicio	USA	1997
www.solutionbase.co.uk	SolutionBase	Servicio	Inglaterra	2003
www.decisioneering.com	Decisioneering, Inc	Comercio/Servicio	USA	1986
www.webopedia.com	Jupitermedia Corporation	Servicio	USA	1999
www.whatis.com	TechTarget, Inc	Servicio	USA	1996
www.about.com	About, Inc	Servicio	USA	1997
gsyc.escet.urjc.es	Universidad Rey Juan Carlos	Educación	España	
iiio.ens.uabc.mx	Universidad Autonoma de Baja California	Educación	Mexico	2001
www.uady.mx	Universidad Autonoma de Yucatan	Educación	Mexico	1996
www.topology.org/soft/sim.html	Site de Alan Kennington	Servicio	Australia	1999
medusa.unimet.edu.ve	Universidad Metropolitana	Educación	Venezuela	1976
profesores.elo.utfsm.cl	Universidad Técnica Federico Santa María	Educación	Chile	1930
elqui.dcsc.utfsm.cl	Universidad Técnica Federico Santa María	Educación	Chile	1930
www ldc.lu.se	Unversidad Lunds	Educación	Suecia	1966
www.ucbcba.edu.bo	Unversidad Católica Boliviana San Pablo	Educación	Bolivia	1966
www.redaccionvirtual.com	Cisco Systems, Inc	Comercio/Servicio/Educacion	USA	1984
www.tele.sunyit.edu	SUNYIT	Educación	USA	
www.sybex.com	Sybex Inc	Comercio	USA	1976

GLOSARIO DE TERMINOS

ACTIVEX: Es una consecuencia de otras dos tecnologías de Microsoft llamadas OLE(Object Linking Embedding) y COM(Component Object Model). Conjunto de tecnologías que permiten a un componente de software interactuar con otro en un entorno de red, sin importar el lenguaje en el cual hayan sido creados.

ANSI: American National Standards Institute o Instituto Nacional Americano de Estándares, Una organización dedicada al desarrollo de los estándares de la industria americana.

APPLICATION PROGRAM INTERFACE (API): Abreviatura del Application Program Interface, de un conjunto de rutinas, de protocolos, y de herramientas para construir aplicaciones de software. Un buen API hace más fácil desarrollar un programa proporcionando todos los bloques para la construcción de dicho programa. Un programador pone los bloques juntos. La mayoría de los ambientes operativos, tales como Ms-Windows, proporcionan un API de modo que los programadores puedan escribir aplicaciones consistentes con el ambiente operativo. Aunque las APIs se diseñan para los programadores, son en última instancia son buenos para los usuarios porque garantizan que todos los programas que usan un API común tendrán interfaces similares. Esto hace más fácil para que los usuarios aprendan nuevos programas.

BACKBONE : Núcleo estructural de la red, que conecta todos los componentes de la red de manera que se pueda producir la comunicación.

BROADCAST: Paquete de datos enviado a todos los nodos de una red. Los broadcasts se identifican por una dirección broadcast.

BUFFER: Un área de almacenamiento temporal de datos.

BUFFERING: Almacenamiento de datos en un buffer mientras realiza algún proceso.

CARGA: indica cuánto de este ancho de banda está actualmente en uso.

CLOCKRATE: Comando utilizado en las interfaces seriales del router para sincronizar a una velocidad en bits por segundo específica.

COMPILADOR: Un recopilador es un programa especial que procesa las declaraciones escritas en un lenguaje de programación particular y los convierte en lenguaje de maquina o "código" que el procesador de una computadora utiliza. Típicamente, un programador escribe declaraciones en un lenguaje tal como PASCAL o C en una línea a la vez que usando un editor. El archivo que se crea contiene lo que se llama código fuente. El programador luego ejecuta el compilador apropiado del lenguaje, especificando el nombre del archivo que contiene el código fuente del programa.

Al ejecutarse, el compilador primero analiza todas las declaraciones del lenguaje sintácticamente una después de la otra y entonces, en unas o más etapas o "pasos sucesivos", construye el código de salida, cerciorándose de que las declaraciones que refieren a otras declaraciones están referidas correctamente en el código final. Tradicionalmente, la salida de la compilación se ha llamado *código de objeto* o a veces *un módulo de objeto*. El código de objeto es el código automático que el procesador puede procesar o "ejecute" una instrucción a la vez.

Más recientemente, el lenguaje de programación de Java, un lenguaje usado en la programación orientada a objeto, ha introducido la posibilidad de compilar la salida (llamada bytecode) que puede funcionar en cualquier plataforma la cual proporcionen una máquina virtual de Java o un intérprete del bytecode que traduzca las instrucciones del bytecode para que puedan ser ejecutadas por el procesador real del hardware.

Tradicionalmente en algunos sistemas operativos, un paso adicional fue requerido después de la compilación que de resolver la localización relativa de instrucciones y de datos cuando más de un módulo de objeto debía ser ejecutado al mismo tiempo y crear referencias cruzadas a la secuencia de instrucciones de cada uno o a los datos. Este proceso fue llamado a veces edición de enlace y la salida conocida como módulo cargable.

COMPONENT OBJECT MODEL (COM): El modelo de componente del objeto (COM) es un marco de trabajo de Microsoft para programas con componentes de objetos de desarrollo y soporte. El objetivo es proporcionar capacidades similares a éstos objetos definidos en el Common Object Request Broker Architecture (CORBA), un marco de trabajo para la interoperación de objetos distribuidos en una red que sea apoyada por otras compañías importantes en la industria de la computación. Mientras que el Object Linking and Embedding (OLE) de Microsoft provee servicios para documentos compuesto que los usuarios ven en su pantalla, COM proporciona los servicios subyacentes en la negociación del interfaz, administración del ciclo de vida (que se determina cuando un objeto se puede quitar de un sistema), licenciando, y los servicios a eventos (que ponen un objeto en servicio como resultado un acontecimiento que ha sucedido a otro objeto). Esto permite a los programadores desarrollar objetos que puedan ser accesados por cualquier aplicación con soporte COM.

CONTROL ACTIVEX: Control usando tecnología ActiveX. Un control ActiveX puede ser automáticamente descargado y ejecutado en un navegador web. ActiveX no es un lenguaje de programación, sino un conjunto de reglas de cómo las aplicaciones deberían compartir información. Estos controles pueden ser desarrollados en varios lenguajes, incluyendo C/C++, Visual Basic y Java.

Un control ActiveX es similar a un Java applet; pero a diferencia de estos, los controles ActiveX tienen acceso completo al sistema operativo Windows. Esto les da mucho mas poder que los Java applet, pero con un cierto riesgo de dañar programas o datos en la maquina. Para controlar este riesgo, Microsoft ha desarrollado un sistema de registro para que los navegadores puedan identificar y autenticar un control ActiveX antes de descargarlo.

Los controles ActiveX solo pueden ser ejecutados en entornos Windows.

CRC: (Cyclic Redundancy Check) Chequeo de redundancia Ciclica. Técnica común utilizada para detectar errores en la transmisión de datos.

CSMA/CD: (*Acceso múltiple con detección de portadora y detección de colisiones*): Mecanismo de acceso a medios dentro del cual los dispositivos que están listos para transmitir datos primero verifican el canal en busca de una portadora. El dispositivo puede transmitir si no se detecta ninguna portadora durante un período de tiempo determinado. Si dos dispositivos transmiten al mismo tiempo, se produce una colisión que es detectada por todos los dispositivos que colisionan. Esta colisión subsecuentemente demora las retransmisiones desde esos dispositivos durante un período de tiempo de duración aleatoria. El acceso CSMA/CD es utilizado por Ethernet e IEEE 802.3.

CUENTA AL INFINITO: Característica de estabilidad de enrutamiento para resolución de lazos de enrutamiento. Al enviar un paquete, cada router de la ruta reduce el campo TTL en al menos una unidad cada vez. De este modo, si el campo TTL alcanza el valor 0 se descarta el paquete. Para prevenir que esta cuenta tienda al infinito, los protocolos de vector distancia definen infinito como un número máximo. Este número se refiere a una métrica de enrutamiento como un número de saltos.

DATAGRAMA O FRAME: Es un paquete en cual se transmite información.

DCE: (Data Communications Equipment) Equipo de Comunicación de Datos. Dispositivo que se comunica con un Equipo Terminal de Datos (DTE) en comunicaciones RS-232C.

DNS: (*Sistema de denominación de dominio*): Sistema utilizado en Internet para convertir los nombres de los nodos de red en direcciones.

DOMINIO DE BROADCAST: Conjunto de todos los dispositivos que reciben tramas de broadcast que se originan en cualquier dispositivo dentro de ese conjunto. Los dominios de broadcast normalmente se encuentran limitados por routers porque los routers no envían tramas de broadcast.

ENCAPSULAMIENTO: Mediante esta técnica conseguiremos que cada clase sea una caja negra, de tal manera que los objetos de esa clase se puedan manipular como unidades básicas. Los detalles de la implementación se encuentran dentro de la clase, mientras que desde el exterior, un objeto será simplemente una entidad que responde a una serie de mensajes públicos.

ENTORNO INTEGRADO DE DESARROLLO: Un entorno integrado de desarrollo (IDE) es un entorno de programación que se ha empaquetado como programa de uso, consistiendo en típicamente un editor del código, un compilador, un depurador, y un constructor del entorno gráfico de usuario (GUI). El IDE puede ser un uso independiente o puede ser incluido como parte de unos o más usos existentes y compatibles. El lenguaje de programación BASIC, por ejemplo, se puede utilizar dentro de Microsoft Office, que permite escribir un programa de WordBasic dentro Microsoft Word. Los IDEs proporcionan un marco amigable para el usuario de uso fácil para muchos lenguajes de programación modernos, tales como Visual Basic, Java, Delphi, etc.

ETHERNET: Especificación de LAN de banda base, inventada por Xerox Corporation y desarrollada conjuntamente por Xerox, Intel y Digital Equipment Corporation. Las redes Ethernet utilizan CSMA/CD y funcionan con una variedad de tipos de cable a 10 Mbps. Ethernet se asemeja a la serie de estándares IEEE 802.3.

ETHERNET 802.3: Es un protocolo de la subcapa de acceso al medio perteneciente al nivel de enlace de datos de modelo OSI, la cual se ocupa principalmente de la transferencia de datos de la capa de red en el host origen a la capa de red del host destino.

FDDI: Abreviación de Fiber Distributed Data Interface o Interfase de Datos Distribuida de Fibra. Conjunto de Protocolos ANSI para envío de datos digitales sobre fibra óptica. Las redes FDDI son redes de pasando la ficha o token-passing y soportan rangos de datos arriba de 100 Mbps (100 millones de bits por segundo). Las redes FDDI son típicamente usadas como backbones de redes de área amplia.

FIABILIDAD: indica la actual tasa de error. Es una fracción de los paquetes que llegan al destino sin error.

FIRMWARE: Software (programas o datos) que se han escrito sobre la memoria solo de Lectura (ROM). El firmware es una combinación del software y del hardware.

FLOODING: (Inundación), Técnica donde todo router debe mantener información del estado de toda interfase en todos los otros routers.

FORMULARIO: Una colección de unas o más páginas que se comporta como una ventana estándar o una caja de diálogo. Una forma contiene los controles para exhibir y corregir datos. Las formas se pueden contener dentro de un sistema de la forma.

Una parte de la pantalla que puede contener su propio documento o mensaje. En programas basados en ventanas, la pantalla puede dividirse en varias ventanas, cada una de las cuales tiene sus propios límites y puede contener un documento diferente (o una presentación distinta del mismo documento). Cada ventana puede contener su propio menú u otros controles, y el usuario puede ampliarla o reducirla mediante un dispositivo señalador (puntero), que se acciona con el ratón o *mouse*.

FTP: (*Protocolo de Transferencia de Archivos*): Protocolo de aplicación, parte de la pila de protocolo TCP/IP, utilizado para transferir archivos entre nodos de red. FTP se define en la RFC 959.

GATEWAY: Un nodo en una red que funciona como una entrada a otra red y es la computadora o router que enruta el tráfico desde un host a fuera de la red.

HARDWARE: Se refiere a los objetos que usted puede realmente tocar, como discos, pantallas, teclados, circuitos, impresoras y tableros. En contraste, el software no se puede tocar. El software existe como ideas, conceptos, y símbolos, pero no tiene ninguna sustancia.

HERENCIA: Es el mecanismo que nos permite crear clases derivadas (especialización) a partir de clases bases (generalización). Es decir, podríamos tener la clase "Empleado" (clase base) y la clase "Vendedor" derivando de la anterior. Una librería de clases (como la MFC) no es más que un conjunto de definiciones de clases interconectadas por múltiples relaciones de herencia.

HOLDDOWNS: Evitan que los paquetes que contienen el mensaje de actualización puedan ser descartados o corrompidos por algún enlace en la red y evitan que los triggered updates no ocurran instantáneamente.

HOP: Una conexión intermedia en una cadena de conexiones que enlazan dos dispositivos de red. Es decir los saltos o routers que un paquete tiene que pasar hasta llegar a su destino.

HTTP: (*Protocolo de Transferencia de Hipertexto*): Protocolo utilizado por los navegadores y servidores de la Web para transferir archivos, como archivos de texto y de gráficos.

ICMP: Internet Control Message Protocol, Protocolo de Mensajes de Control de Internet, una extensión del protocolo de Internet (IP). ICMP soporta paquetes conteniendo error, control y mensajes de información. Por ejemplo el comando PING usa ICMP para probar una conexión de red.

ICONO: En entornos gráficos, pequeña imagen gráfica mostrada en la pantalla que representa un objeto manipulable por el usuario. Por ejemplo, una papelera representa un comando para borrar textos o archivos no deseados. Los iconos permiten controlar ciertas funciones de la computadora sin tener que recordar comandos ni escribirlos con el teclado. Son un elemento importante de las interfaces gráficas de usuario, ya que facilitan el manejo de las distintas funciones.

IGP: Protocolo de Gateway Interior. Protocolo de Internet utilizado para Intercambiar información de enrutamiento dentro de un Sistema Autónomo. Entre los ejemplos de IGP de Internet comunes se incluyen IGRP, OSPF y RIP.

INTERFAZ GRÁFICA: Tipo de visualización que permite al usuario elegir comandos, iniciar programas y ver listas de archivos y otras opciones utilizando las representaciones visuales (iconos) y las listas de elementos del menú. Las selecciones pueden activarse bien a través del teclado o con el ratón. Para los autores de aplicaciones, las interfaces gráficas de usuario ofrecen un entorno que se encarga de la comunicación con la computadora. Esto hace que el programador pueda concentrarse en la funcionalidad, ya que no está sujeto a los detalles de la visualización ni a la entrada a través del ratón o del teclado.

También permite a los programadores crear programas que realicen de la misma forma las tareas más frecuentes, como guardar un archivo, porque la interfaz proporciona mecanismos estándar de control como ventanas y cuadros de diálogo. Otra ventaja es que las aplicaciones escritas para una interfaz gráfica de usuario son independientes de los dispositivos: a medida que la interfaz cambia para permitir el uso de nuevos dispositivos de entrada y salida, como un monitor de pantalla grande o un dispositivo óptico de almacenamiento, las aplicaciones pueden utilizarlos sin necesidad de cambios.

INTERNET: La internetwork de redes más grande del mundo, que conecta decenas de miles de redes de todo el mundo y con una cultura que se concentra en la investigación y estandarización basada en el uso real. Muchas tecnologías de avanzada provienen de la comunidad de la Internet. La Internet evolucionó en parte de ARPANET. En un determinado momento se la llamó Internet DARPA, y no debe confundirse con el término general Internet.

INTERNETWORK DE REDES: Agrupamiento de redes interconectadas por routers y otros dispositivos que funciona (de modo general) como una sola red.

IP: (*Protocolo Internet*): Protocolo de capa de red de la pila TCP/IP que ofrece un servicio de internetwork de redes no orientado a conexión. El IP brinda funciones de direccionamiento, especificación del tipo de servicio, fragmentación y reensamblaje, y seguridad. Se define en RFC 791. IPv4 (Protocolo Internet versión 4) es un protocolo de conmutación no orientado a conexión de máximo esfuerzo.

ISL: Protocolo Inter.- Switch Link (ISL) El Protocolo Inter-Switch Link es un método propietario de Cisco que sirve para preservar la identificación origen de la VLAN, en los frames que están pasando sobre un trunk link.

LAN (*red de área local*): Red de datos de alta velocidad y bajo nivel de errores que cubre un área geográfica relativamente pequeña (hasta unos pocos miles de metros). Las LAN conectan estaciones de trabajo, periféricos, terminales y otros dispositivos en un solo edificio u otra área geográficamente limitada. Los estándares de LAN especifican el cableado y señalización en las capas físicas y de enlace de datos del modelo OSI. Ethernet, FDDI y Token Ring son tecnologías LAN ampliamente utilizadas.

LENGUAJE DE PROGRAMACIÓN: Cualquier lenguaje artificial que puede utilizarse para definir una secuencia de instrucciones para su procesamiento por una computadora. Es complicado definir qué es y qué no es un lenguaje de programación. Se asume generalmente que la traducción de las instrucciones a un código que comprende la computadora debe ser completamente sistemática. Normalmente es la computadora la que realiza la traducción.

MAC: Dirección física de la tarjeta de red.

MÉTODOS: Una acción que un objeto es capaz de ejecutar.

METRICA: es cualquier parámetro que sirva para establecer una comparación entre rutas.

MICROSOFT WINDOWS: Una familia de sistemas operativos para computadoras personales. Windows domina el mundo de la computadora personal, funcionamiento, por algunas estimaciones, en el 90% de todos los computadores personales. Los 10% restantes son sobre todo computadoras de Macintosh. Como el ambiente de funcionamiento de Macintosh, Windows proporciona una interfaz gráfica de usuario (GUI), gestión de la memoria virtual, tareas múltiples, y la ayuda para muchos dispositivos periféricos.

MODULO: En software, un módulo es una parte de un programa. Los programas se componen de unos o más módulos independientemente desarrollados que no se combinan hasta que el programa es enlazado. Un solo módulo puede contener una o varias rutinas.

MTU: Unidad de Transmisión Máxima. El tamaño físico más grande del paquete, medido en bytes que una red puede transmitir. Cualquier mensaje más grande que el MTU se divide en paquetes más pequeños antes de ser enviado.

MULTIPLE DOCUMENT INTERFACE(MDI): La interfaz múltiple de documento es una interfaz de programación de Microsoft Windows para crear una aplicación que permite a los usuarios trabajar con múltiples documentos al mismo tiempo. Cada documento está en un espacio separado con sus propios controles. El usuario puede ver y trabajar con diversos documentos tales como una hoja electrónica, un documento del texto, o un espacio de dibujo simplemente moviendo el cursor de un espacio a otro.

Una aplicación MDI es algo como la interfaz de escritorio de Windows puesto que ambos incluyen espacios múltiples de visualización. Sin embargo, las vistas del MDI se confinan al área de la ventana o del cliente de la aplicación. Dentro del área del cliente, cada documento se exhibe dentro de una ventana "hija" separada. Las aplicaciones MDI se pueden utilizar para una variedad de propósitos - por ejemplo, trabajar en un documento mientras que lo que se refiere a otro documento, ver diversas presentaciones de la misma información, viendo sitios múltiples en el web, y cualquier tarea que requiera puntos de referencia múltiples y áreas de trabajo en el mismo tiempo.

NVRAM: (Non-Volatile Random Access Memory). Un tipo de memoria que retiene su contenido cuando el dispositivo se apaga. Utilizada en el router y switch para guardar la configuración.

OSI (*internetwork de sistemas abiertos*): Programa internacional de estandarización creado por ISO e UIT-T para desarrollar estándares de networking de datos que faciliten la interoperabilidad de equipos de varios fabricantes.

P-CODE: Es un paso intermedio entre las instrucciones de alto nivel de un programa en Basic y el código nativo de bajo nivel que el procesador de una computadora ejecuta (Visual Basic no genera código de maquina como los compiladores). En tiempo de ejecución, Visual Basic traduce cada declaración del p-code a código nativo. Compilando directamente al formato del código nativo, se elimina el paso intermedio del p-code.

Básicamente, el pcode es un conjunto compilado de instrucciones escritas de tal forma que el ambiente en tiempo de ejecución de Visual Basic pueda traducir las instrucciones para hacer funcionar un programa.

PING (*búsqueda de direcciones de internet*): Mensaje de eco ICMP y su respuesta. A menudo se usa en redes IP para probar el alcance de un dispositivo de red.

POISONING: Característica de estabilidad en el proceso de enrutamiento el cual es propuesto para romper lazos más grandes. Su papel es que cuando una actualización muestre la métrica para una ruta existente y tenga que aumentar suficientemente, hay un lazo. La ruta debería ser eliminada y aplicar holddown.

POINT TO POINT: Enlace de comunicación de punto a punto.

POLIMORFISMO: Esta característica nos permite disponer de múltiples implementaciones de un mismo método de clase, dependiendo de la clase en la que se realice. Es decir, podemos acceder a una variedad de métodos distintos (con el mismo nombre) mediante el mismo mecanismo de acceso. En C++ el polimorfismo se consigue mediante la definición de clases derivadas, funciones virtuales y el uso de punteros a objetos.

PROPIEDADES: Atributo de un control que define una de las características de un objeto o un aspecto de su comportamiento.

RETARDO: es la cantidad de tiempo que pasa hasta llegar al destino a través de la ruta, asumiendo una red no cargada.

RFC: Request For Comments, una serie de notas sobre el Internet, comenzada en 1969 (cuando el Internet era el ARPANET). Un documento del Internet se puede someter al IETF por cualquier persona, pero el IETF decide esta bien el documento

para un RFC. Eventualmente, si gana bastante interés, él puede convertir en un estándar del Internet.

ROUND-ROBIN: Algoritmo que en su versión más simple dado un conjunto de elementos selecciona uno a partir de haberlos ordenado en forma circular, permitiendo que siempre que se complete un ciclo, todos hayan sido elegidos la misma cantidad de veces.

SHORTER PATH FIRST: Primera Ruta más Corta. Utilizada en el proceso de enrutamiento para despachar los paquetes.

SPLIT HORIZON: Característica de estabilidad en el proceso de enrutamiento que consiste en partir el horizonte (Split horizon): Omite la información de distancia que fue obtenida del nodo al cual se le envía la actualización.

SMTP: Simple Mail Transport Protocol (Protocolo de transporte de correo simple)

SUBNETTING: Implementación de subredes, lo que permite reducir el numero 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.

TABLA DE ENRUTAMIENTO: Tabla almacenada en un router o en algún otro dispositivo de internetwork que realiza un seguimiento de las rutas hacia destinos de red específicos y, en algunos casos, las métricas asociadas con esas rutas.

TCP: (*Protocolo de Control de Transmisión*): Protocolo de capa de transporte orientado a conexión que provee una transmisión confiable de datos de dúplex completo. TCP es parte de la pila de protocolo TCP/IP.

TCP/IP: (*Protocolo de Control de Transmisión /Protocolo Internet*): Nombre común para el conjunto de protocolos desarrollados por el DoD de EE.UU. en los años '70 para promover el desarrollo de internetwork de redes a nivel mundial. TCP e IP son los dos protocolos más conocidos del conjunto.

TELNET: Protocolo de emulación de terminal estándar de la pila de protocolo TCP/IP. Telnet se usa para la conexión de terminales remotas, permitiendo que los usuarios se registren en sistemas remotos y utilicen los recursos como si estuvieran conectados a un sistema local. Telnet se define en RFC 854.

TIMERS: Temporizador que controla el tiempo de algún proceso.

TFTP: (*Protocolo de Transferencia de Archivos Trivial*): Versión simplificada de FTP que permite la transferencia de archivos de un computador a otro a través de una red.

TOKEN – RING: Un tipo de redes de computadoras en el cual todas las computadoras están esquemáticamente en un círculo. Una ficha o token, el cual es un patrón de un bit especial, este viaja a través del círculo. Para enviar un mensaje una computadora toma el token, adjunta el mensaje al y luego lo deja que siga el viaje por la red.

TOPOLOGIA: La forma de la red área local (LAN) o de otro sistema de comunicaciones. Las topologías son físicas o lógicas.

TRACEROUTE: Programa disponible en varios sistemas que rastrea la ruta que recorre un paquete hacia un destino. Se utiliza a menudo para depurar los problemas de enrutamiento entre hosts. Existe también un protocolo traceroute definido en RFC 1393.

TRAILER: Terminador de datagrama o frame. En esta parte del datagrama o frame se encuentra CRC.

TRIGGERED UPDATE: Característica de estabilidad en el proceso de enrutamiento que establece una nueva tabla de Ruteo que es enviada inmediatamente, en respuesta a algún cambio.

TRUNKING: Compartir un canal donde se puedan transmitir múltiples VLANs.

TRUNK LINK: Es un enlace capaz de transportar tráfico de múltiples VLANs entre dos switches o entre un switch y un router.

UDP: (*Protocolo de Datagrama de Usuario*): Protocolo no orientado a conexión de la capa de transporte de la pila de protocolo TCP/IP. UDP es un protocolo simple que intercambia datagramas sin confirmación o garantía de entrega y que requiere que el procesamiento de errores y las retransmisiones sean manejados por otros protocolos. UDP se define en la RFC 768.

UIT-T: Sector de Normalización de las Comunicaciones de la UIT (Unión Internacional de Telecomunicaciones), organismo especializado de Naciones Unidas en el campo de las comunicaciones.

UPSTREAM: Ancho de banda en subida o carga.

VLAN: LAN Virtual, grupo de dispositivos de una LAN que están configurados (usando el software de administración) de tal modo que se pueden comunicar como si estuvieran conectados al mismo cable, cuando, en realidad, están ubicados en una serie de segmentos de LAN distintos. Debido a que las LAN virtuales están basadas en conexiones lógicas en lugar de físicas, son extremadamente flexibles.

WAN (*Red de área amplia*): Red de comunicación de datos que sirve a usuarios dentro de un área geográfica extensa y a menudo usa dispositivos de transmisión suministrados por carriers comunes. Frame Relay, SMDS y X.25 son ejemplos de WAN.

ANEXOS

1. Manual de Usuario

Acerca de este manual

Objetivos de este documento

Esta publicación provee la información necesaria que le ayudará a utilizar correctamente el Simulador de Redes NETZBuilder 1.2 y así obtener el mejor provecho de esta herramienta, poner en practica y mejorar los conocimientos teóricos de redes, prepararte para la configuración de pcs, switches y routers, poner a prueba sus conocimientos, probar y construir diseños instalaciones y configuraciones de redes a un bajo costo.

Dirigido

Esta publicación esta dirigida para dar el soporte necesario sobre el simulador NETZBuilder a los estudiantes de redes de la Universidad Don Bosco quienes tienen conocimientos sobre redes LAN y WAN, TCP/IP, Configuración y funcionamiento de swiches, routers.



Organización de este documento

INTRODUCCIÓN	246
¿QUÉ ES NETZBUILDER?.....	246
EL PROGRAMA SIMULA LOS SIGUIENTES PROCESOS Y/O ELEMENTOS DE RED	247
LIMITACIONES DE DISEÑO	253
REQUERIMIENTOS DE SISTEMA	256
INICIANDO	257
COMO INSTALAR NETZBUILDER 1.2	257
COMO INICIAR NETZBUILDER 1.2	258
CÓMO UTILIZAR NETZBUILDER	258
CONOZCA NETZBUILDER 1.2	258
MODO DISEÑO.....	259
MODO EVALUACIÓN.....	259
MODO CONSOLA	259
MENÚ PRINCIPAL	260
<i>Menú Archivo</i>	260
<i>Menú Ver</i>	260
<i>Menú Evaluaciones</i>	261
<i>Menú Ayuda</i>	261
MENÚ CONTEXTUAL	261
<i>Menú Contextual de Hub</i>	263
<i>Menú Contextual de Switch</i>	263
<i>Menú Contextual de Router</i>	263
PANTALLA PRINCIPAL DE DISEÑO	264
PANTALLA DE LABORATORIOS	265
<i>Ventana de Calificación</i>	266
BARRA DE DISPOSITIVOS	267

BARRA DE HISTORIAL.....	268
CONSOLA.....	268
<i>Consola de Host</i>	269
<i>Consola de Switch</i>	270
<i>Consola de Router</i>	271
VENTANA DE CONFIGURACIÓN DE HOST	272
VENTANA DE SELECCIÓN DE PUERTOS DE SWITCH.....	272
VENTANA DE SELECCIÓN DE INTERFAZ DE ROUTER	273
CÓMO COLOCAR LOS DISPOSITIVOS EN EL ÁREA DE DISEÑO	273
CÓMO CONFIGURAR UN HOST	274
CÓMO MOVER LOS DISPOSITIVOS DENTRO DEL ÁREA DE DISEÑO.....	274
CÓMO CABLEAR UN DISPOSITIVO CON OTRO	274
CÓMO ELIMINAR UN DISPOSITIVO	275
CÓMO IMPRIMIR UN PROYECTO	275
CÓMO CALIFICAR UN LABORATORIO.....	276
CÓMO UTILIZAR LAS CONSOLAS.....	277
<i>Consola de Host</i>	277
<i>Consola de Switch</i>	277
<i>Consola de Router</i>	277

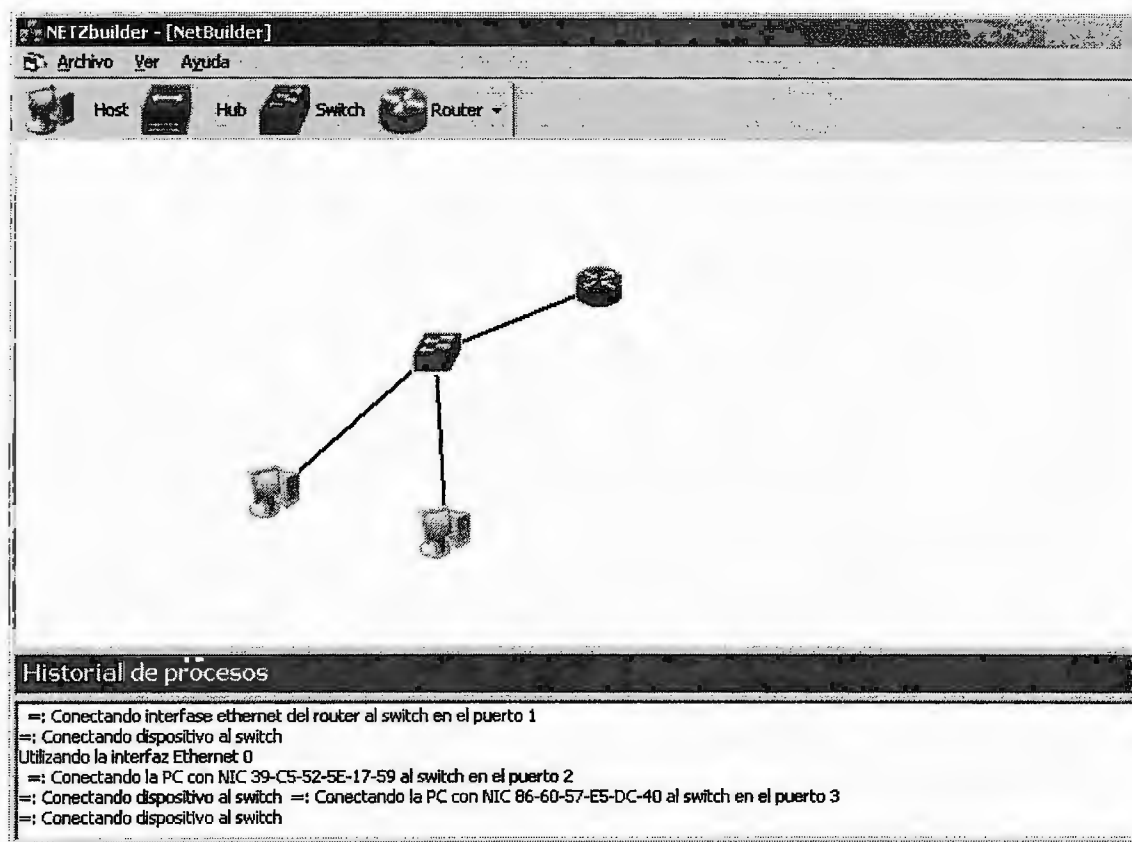
Introducción

¿Qué es NETZBuilder?

Sea usted bienvenido NETZBuilder 1.2 este es un Simulador interactivo y gráfico de Redes LAN y WAN Básica especializado en la enseñanza de redes. En este simulador usted será capaz de construir sus propios diseños de red utilizando los siguientes dispositivos de red:

- Host o PC
- Hub o Concentrador
- Switch LAN o Conmutador LAN
- Router o Enrutador

Todos los dispositivos son configurables menos el hub y cada uno pueden ser interconectado para formar cada red.



Vista de NETZBuilder 1.2

Características

- El simulador contará con dos herramientas de prueba o diagnóstico las cuales son: PING en consola de Router y PC y TRACERT solamente en consola de PC.
- La configuración de cada dispositivo y el diseño en pantalla podrán ser impresos en papel.
- La red diseñada en el simulador podrá ser guardada para futuras modificaciones.
- Los resultados de las evaluaciones serán visualizados en pantalla y en papel.
- La configuración de cada dispositivo y el diseño en pantalla podrán ser impresos en papel.
- El historial de los procesos internos del router solo mostrara las siguientes acciones: Enviar, Cambiar métrica, Eliminar Ruta, Construcción de mensaje de actualización.
- Se llevara a cabo la simulación de procesos internos de los routers, como son, actualización, propagación de tablas, entre otros; los cuales se irán describiendo en una ventana de Historial que será mostrada cuando el usuario la invoque y se ocultara de la misma forma.
- Se podrán establecer ip secundarias en las interfaces del router.
- Implementación de búsquedas ARP, pero sin comandos de visualización.
- Se podrá eliminar los cables utilizados para la conexión entre dispositivos.
- Mediante comando se muestra las ip secundarias configuradas en las interfaces de router.
- Se podrá eliminar las Vlan creadas por el usuario. (Los puertos asociados a la Vlan eliminada, vuelen a asociarse a la Vlan1)

Router

- Procesos de Protocolo de Enrutamiento RIP e IGRP
 - Actualización de tabla de ruteo
 - Expiración de ruta
 - Construcción y envío de mensaje de actualización implementando split horizon
 - Enrutamiento de paquetes
- Sumarización hacia fuera
- Construcción y actualización de Tabla de Ruteo automáticamente
- Consola de configuración de router

Comandos Disponibles

Modo Usuario

1er Nivel	2do Nivel	3er Nivel	4rto Nivel	5to Nivel
?				
exit				
enable				
ping				
show	interface	serial	0	
			1	
		ethernet	0	
			1	
	ip	interface	serial	0
				1
			ethernet	0
				1
	protocols			
	version			

Comandos Disponibles

Modo Usuario

1er Nivel	2do Nivel	3er Nivel	4to Nivel	5to Nivel
?				
exit				
enable				
show	interface	fastethernet0/1		
		fastethernet0/2		
		fastethernet0/3		
		fastethernet0/4		
	version			

Modo Privilegiado

1er Nivel	2do Nivel	3er Nivel	4to Nivel	5to Nivel
?				
exit				
disable				
show	interface	fastethernet0/1		
		fastethernet0/2		
		fastethernet0/3		
		fastethernet0/4		
	version			
	vlan			
	vlan-membership			
	trunk			
	running-config			
	mac-address-table			
configure	terminal			

Modo de Configuración Global

1er Nivel	2do Nivel	3er Nivel	4to Nivel	5to Nivel
?				
exit				
hostname				
enable	password			
set	vlan			
		vlan-num		
	trunk	port	on/off	
no	hostname			
	enable	password		
	set	vlan	vlan-num	

El programa simula los siguientes procesos y/o elementos de red

- Direccionamiento IP
- Herramienta ping
- Herramienta tracert

Host

- Configuración TCP/IP de la Network Interface Card (NIC) de una PC
- Resolución de Direcciones (ARP) para la comunicación
- Consola DOS para PC

Propiedades TCP/IP Disponibles

Nombre Propiedad	Ejemplo
Dirección IP	192.168.1.2
Mascara de subred	255.255.255.0
Puerta de enlace predeterminada	192.168.1.1

Switch

- Construcción y actualización de Tabla MAC del switch automáticamente
- Conmutado de paquetes en base a Tabla MAC del switch
- Configuración y funcionamiento de VLAN en switch
- Configuración y funcionamiento de Trunk en switch
- Consola de configuración de switch

Router

- Procesos de Protocolo de Enrutamiento RIP e IGRP
 - Actualización de tabla de ruteo
 - Expiración de ruta
 - Construcción y envío de mensaje de actualización implementando split horizon
 - Enrutamiento de paquetes
- Sumarización hacia fuera
- Construcción y actualización de Tabla de Ruteo automáticamente
- Consola de configuración de router

Comandos Disponibles

Modo Usuario

1er Nivel	2do Nivel	3er Nivel	4rto Nivel	5to Nivel
?				
exit				
enable				
ping				
show	interface	serial	0	
			1	
		ethernet	0	
			1	
	ip	interface	serial	0
				1
			ethernet	0
				1
	protocols			
	version			

Modo Privilegiado

1er Nivel	2do Nivel	3er Nivel	4to Nivel	5to Nivel
?				
exit				
disable				
ping				
copy	running-config	startup-config		
show	interface	serial	0	
			1	
		ethernet	0	
			1	
	ip	interface	serial	0
				1
			ethernet	0
				1
		protocols		
		route		
	protocols			
	version			
	running-config			
configure	terminal			

Modo de Configuración Global

1er Nivel	2do Nivel	3er Nivel	4to Nivel	5to Nivel
?				
exit				
hostname				
configure-register				
enable	secret			
	password			
ip	route			
interface	serial	0		
		1		
	Ethernet	0		
		1		
router	rip			
	igrp	SA		
no	hostname			
	enable	secret		
		password		
	ip	route		
	router	rip		
		igrp		

Modo de Configuración Especifica de Interfaz

1er Nivel	2do Nivel	3er Nivel	4rto Nivel	5to Nivel	6to Nivel
?					
exit					
end					
ip	address			secondary	
shutdown					
no	ip	address			secondary
	shutdown				

Modo de Configuración Especifica de Router

1er Nivel	2do Nivel	3er Nivel	4rto Nivel	5to Nivel
?				
exit				
end				
network				
no	network			

Limitaciones de Diseño

- Al Diseñar la red en el simulador solo se podrá utilizar cable UTP estándar y seriales para conexiones entre routers.
- Los protocolos de ruteo a utilizar serán IGRP y RIP versión 1.
- No se realizarán pruebas con la herramienta Telnet, ni con otra que no sean las dos antes mencionadas.
- Unión de dos switches, solo cuando sea implementación de trucking en VLAN.
- Solo se pueden crear y asignar 4 vlans adicionales a la Vlan por defecto (Vlan1), en cada switch y el rango de valores es de 1-1001.
- El sistema autónomo del protocolo IGRP acepta valores entre 1 a 65535.
- No se permite poner hubs en cascada, tampoco hubs cableados a un switch, ni viceversa.
- Los comandos ping y tracer no aceptan ningún parámetro
- El comando de ayuda (?) muestra la lista de comandos de los siguientes modos: usuario, privilegiado, configuración global, configuración específica de interfaz, y de router.
- Al poner en Shutdown una interfaz no funciona ninguna ip asociada a esta.
- Al eliminar un ip primaria de una interfaz se eliminan las ip secundarias.
- La salida del comando show interfaces en el switch es netamente descriptiva, solo para que el usuario conozca la salidas de este.
- Ninguna lista de control de acceso puede ser configurada en los routers.
- Ningún comando de switch o router permiten parámetros opcionales, excepto el comando para configuración de ip secundarias.
- El comando configure-register del router, acepta valores hexadecimales desde 0 hasta FFFF, y el registro solo es para que el usuario lo conozca.
- No se pueden conectar host, hub o switch a interfaces seriales del router.
- No se puede cablear un host a otro host directamente.
- Entre routers, no se pueden conectar una interfaz ethernet a una serial de otro router.

- El comando `copy running-config startup-config` solo es para familiarizarse con este.
- Los routers no envían mensajes de actualización por medio de sus interfaces ethernet.
- El comando `Clock rate` no esta implementado.
- Existe un límite de caracteres en las consolas con un promedio de dos millones.
- El switch no acepta configuraciones IP.
- Esta implementado en el switch solamente la Interfaz de Línea de Comandos (CLI).
- Aunque los dispositivos del simulador están basados en los modelos existentes en el laboratorio de CISCO. No todos los comandos soportados por estos dispositivos están implementados en el simulador. En el manual de usuario y en la ayuda del simulador se detallan los únicos comandos que están implementados.
- No esta implementada la administración remota de Router.
- Las contraseñas de seguridad en modo global implementadas son: `Secret` y `password` para el router y solamente `password` para el switch. Ningún otro tipo de contraseña ha sido implementada.
- Sistemas operativos en los que puede instalarse el simulador:
 - Windows (98/NT/2000/XP/2003)
- Limitación de dispositivos por diseño

Máximo de Routers	5 (Cuatro 2501 y Un 2514)
Máximo de Switch	6
Máximo de Hub	6
Máximo de LAN	6
Máximo de Hosts por LAN	4

Nota: El número máximo de dispositivos (Switches o Hubs) por diseño es 6, esto significa que puede haber 6 switches, o 6 hubs o la combinación entre ellos que hagan un total de 6.

- No esta implementado la redistribución de rutas.

Detalle de posibles combinaciones entre # máximo de hubs y switches por diseño

Combinación	# de Hubs	# de Switches	#Total de Dispositivos
A	6	0	6
B	5	1	6
C	4	2	6
D	3	3	6
E	2	4	6
F	1	5	6
G	0	6	6

Detalle de posibles combinaciones entre # máximo hubs y switches por diseño cuando existe trunking

Combinación	# de Hubs	# de Switches	#Total de Dispositivos
A	4	0	4
B	3	1	4
C	2	2	4
D	1	3	4
E	0	4	4

Nota: Tomándose en cuenta que en Trunking ya hay dos switch utilizados

Requerimientos de sistema

Requerimientos de Hardware

Hardware	Mínimo	Optimo
Procesador (Mhz)	450	2000
Memoria (MB)	64	256
Disco Duro (MB)	300	400

Requerimientos de Sistema Operativo

- Windows 98
- Windows NT
- Windows 2000
- Windows XP
- Windows 2003

Iniciando

Como Instalar NETZBuilder 1.2

1. Desde el directorio raíz del CD de instalación, haga doble sobre setup.exe. Aparecerá el Asistente de Instalación.
2. Seleccione la carpeta destino donde desea instalar NETZBuilder y luego haga click en el botón de siguiente.
3. Seleccione la carpeta de menú de inicio en donde desea instalar los iconos de acceso, luego haga clic en el botón de siguiente.
4. Seleccione las tareas adicionales como "Crear icono en el escritorio" y "Crear icono en Quick launch" y luego haga clic en siguiente.
5. Aparecerá la ventana de resumen, indicando los procesos seleccionados anteriormente. Haciendo clic en Instalar iniciará la copia de archivos necesarios y registro controles activeX y dlls. Haciendo clic en el botón de atrás usted puede modificar alguna selección.
Nota: Si alguno del los componentes de sistema ya existe, aparecerá una ventana indicando error que no puede registrarse, en este caso ignorar los componentes.
6. Al finalizar la instalación, aparecerá la ventana para completar la instalación y haciendo clic en el botón terminar reiniciara el equipo para que la instalación sea completada. En el caso de que ya estén registrados los componentes aparecerá una ventana seleccionado la opción de Cargar la aplicación

En este momento usted habrá instalado satisfactoriamente NETZBuilder en su computadora.

Como iniciar NETZBuilder 1.2

1. Desde su escritorio, haga click en Inicio, apunte a Programas, apunte a NETZBuilder y haga click en NETZBuilder. Se cargará la aplicación NETZBuilder 1.2

Como iniciar un nuevo proyecto

- a. Haga click en Archivo y luego apunte a Nuevo o la combinación de teclas CTRL+N. Cargara la pantalla principal de NETZBuilder con un nuevo proyecto.

Como abrir un proyecto guardado

- a. Haga click en Archivo y luego apunte a Abrir o la combinación de teclas CTRL+A. Cargara la ventana de Abrir.
- b. Seleccione la ubicación del archivo con extensión .ats donde se encuentra el proyecto que desea abrir
- c. Haga click en el botón abrir. Cargara el proyecto guardado en pantalla

Cómo utilizar NETZBuilder

Conozca NETZBuilder 1.2

Construya sus propios diseños de red, configure dispositivos como pcs, routers y switches, evalúe su conocimiento, observe el funcionamiento interno de los dispositivos, pruebe sus diseños y configuraciones, imprima sus proyectos y guárdelos en disco. El entorno del simulador NETZBuilder 1.2 consiste en una barra de menú, menús contextuales, pantalla de diseño, pantalla de laboratorios, barra de dispositivos, barra de historial, consolas y ventana de selección de puertos.

Modo Diseño

Para iniciar el modo diseño de NETZBuilder, abra un nuevo proyecto o abra uno guardado desde las opciones Nuevo y Abrir respectivamente del Menú Archivo.

En modo diseño usted puede construir todos los diseños que desee, con los dispositivos disponibles y luego interconectarlos o cablearlos entre sí, siempre en base las limitantes para esta versión de NETZBuilder.

Nota: No podrá pasar al Modo Evaluación cuando este en Modo Diseño, sino hasta que cierre este modo

Modo Evaluación

Para iniciar el modo evaluación de NETZBuilder, Seleccione cualquier Laboratorio propuesto del menú Evaluaciones.

En este modo usted será capaz de realizar todos los laboratorios prediseñados y luego de completado el laboratorio calificarlo.

Nota: No podrá pasar al modo Diseño cuando este en Modo Evaluación, sino hasta que cierre este modo

Modo Consola

Para iniciar el modo consola de NETZBuilder, previamente tendrá que haber iniciado cualquiera de los modos anteriores Diseño o Evaluación y luego en el menú contextual de host, switch y router tiene la opción para cargar la consola de cada dispositivo.

En este modo puede configurar, utilizar comandos y herramientas de conectividad.

Al tener iniciado este modo, no podrá pasar a los otros modos sino hasta que cierre la consola que ha cargado.

Menú Principal



Vista de Menú Principal

La barra del menú principal se muestra en la parte superior de la pantalla principal del simulador.

Los 4 submenús desplegables de Archivo, Ver, Evaluaciones y Ayuda, son con los que cuenta el simulador para poder realizar los proyectos de red.

Menú Archivo

Las opciones de Nuevo, Abrir e Impresión se realizan desde el menú Archivo.

Nuevo: Crea un nuevo proyecto en blanco.

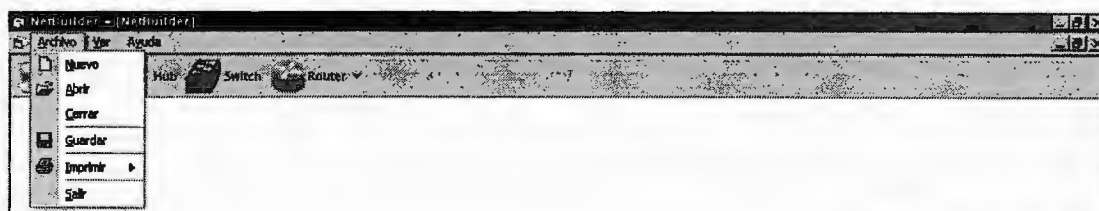
Abrir: Abre o busca un proyecto guardado.

Guardar: Guarda el proyecto activo con su nombre, ubicación y configuraciones.

Imprimir: Imprime el diseño y/o las configuraciones del proyecto activo.

Cerrar: Cierra proyecto actual.

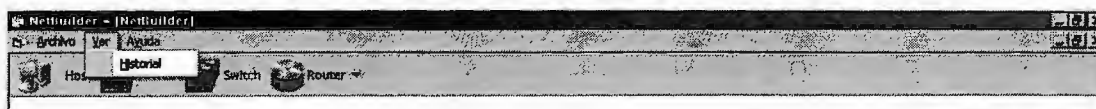
Salir: Cierra el simulador.



Vista de Menú Archivo

Menú Ver

La opción de Ver historial se realiza desde el Menú Ver. La barra de historial muestra los eventos internos del simulador en tiempo real.



Vista de Menú Ver

Menú Evaluaciones

Las opciones de Lab 1. Router Básico, Lab 2. Router y LAN, Lab 3. Router y WAN, Lab 4. RIP y LAN, Lab 5. IGRP Básico, Lab 6. IGRP Avanzado, Lab 7. VLAN Básico y Lab 8. VLAN Avanzado se realizan desde el Menú Evaluaciones. Cada opción de laboratorio lo llevara a realizar un laboratorio prediseñado y luego de completarla podrá obtener una calificación.



Vista de Menú Evaluaciones

Menú Ayuda

Las opciones de Ayuda de NETZBuilder y Acerca de... se realizan desde el Menú Ayuda.

Ayuda: El Ayudante de NETZBuilder ofrece temas de ayuda y sugerencias para construir los proyectos de red.

Acerca de: Muestra el número de versión del programa, Qué es NETZBuilder y los avisos de Copyright.

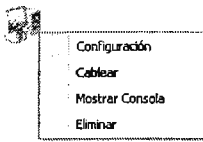


Vista de Menú Ayuda

Menú Contextual

Cada dispositivo de red posee su propio menú contextual al hacer clic derecho sobre el dispositivo.

Menú Contextual de Host



Una vez colocado un host en el área de la pantalla de diseño, haga click derecho sobre el icono de host y se mostrara el menú contextual con las opciones de Configuración, Cablear,

Eliminar y Mostrar Consola.

Configuración: Carga la ventana de configuración básica de TCP/IP para la tarjeta de red del host.

Cablear: Habilita el host para ser cableado en un puerto de hub, switch o router.

Eliminar: Elimina el dispositivo host con todo y su cable si lo tuviera.

Mostrar Consola: Carga la consola de DOS para utilizar las herramientas de prueba ping y trace.

Menú Contextual de Hub

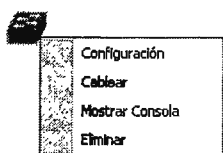


Una vez colocado un Hub en el área de la pantalla de diseño, haga click derecho sobre el icono de hub y se mostrara el menú contextual con las opciones de Cablear y eliminar.

Cablear: Habilita el hub para recibir el cableado de un host o router.

Eliminar: Elimina el dispositivo hub con todo y su(s) cable(s) si los tuviera.

Menú Contextual de Switch



Una vez colocado un Switch en el área de la pantalla de diseño, haga click derecho sobre el icono de Switch y se mostrara el menú contextual con las opciones de

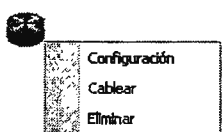
Configuración, Cablear y Eliminar.

Configuración: Carga la consola de switch.

Cablear: Habilita el switch para recibir el cableado de un host, router o de otro switch en caso especial de trunk.

Eliminar: Elimina el dispositivo switch con todo y su(s) cable(s) si lo tuviera.

Menú Contextual de Router



Una vez colocado un Router en el área de la pantalla de diseño, haga click derecho sobre el icono de Router y se mostrara el menú contextual con las opciones de

Configuración, Cablear y Eliminar.

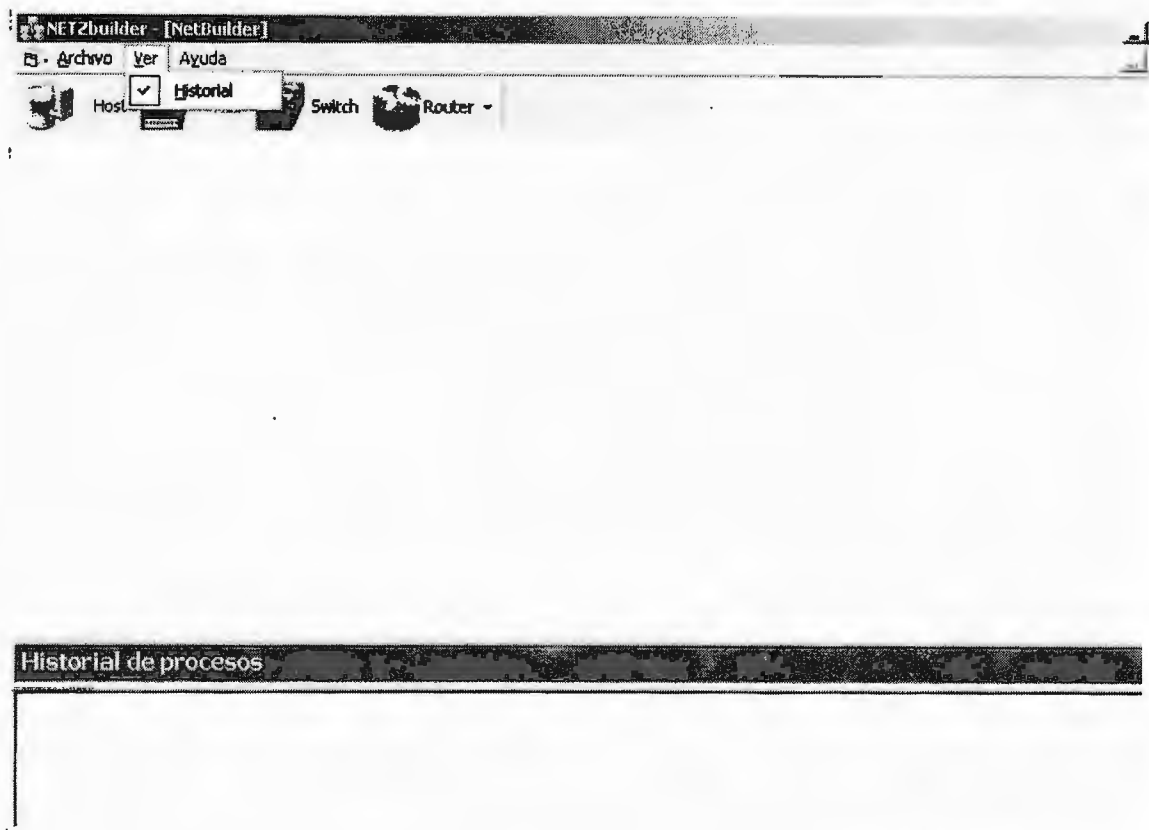
Configuración: Carga la consola de Router.

Cablear: Habilita el router para recibir el cableado de un host, router o de un switch.

Eliminar: Elimina el dispositivo router con todo y su(s) cable(s) si lo tuviera.

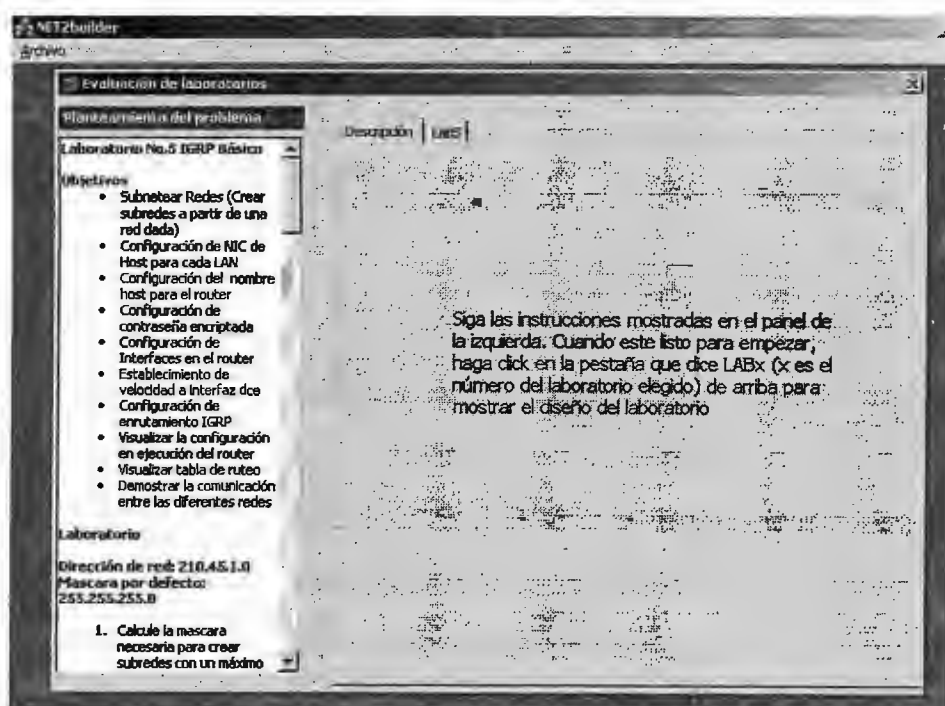
Pantalla Principal de Diseño

La pantalla principal de diseño es el área donde se puede construir los diseños de red que usted desee con los elementos de la barra de dispositivos de red (host, hub, switch y router) basándose en las restricciones mencionadas anteriormente, además de los diseños usted puede configurar y observar el funcionamiento de los dispositivos host, switch y router como si estuviera con uno real, hacer las pruebas de conectividad necesarias y medir sus conocimientos.

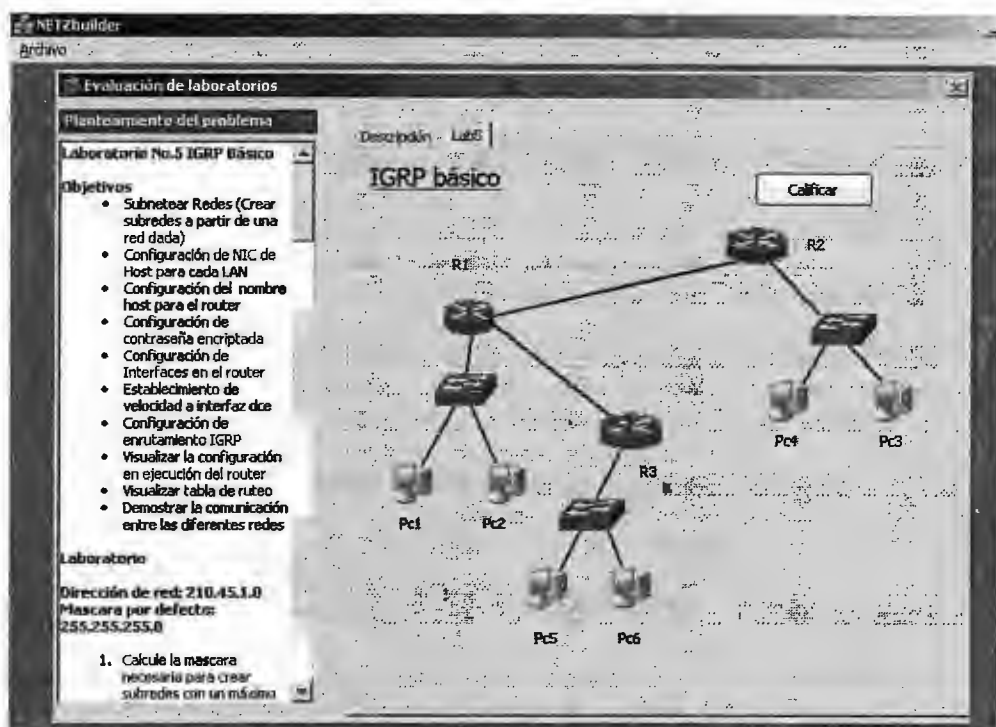


Vista de Pantalla Principal

Pantalla de Laboratorios



Vista de Pantalla de Descripción de Laboratorios



Vista de Pantalla de Diseño de Laboratorios

La pantalla de laboratorios se carga cuando usted selecciona cualquier opción de laboratorio del menú de evaluaciones.

En esta pantalla usted encontrará en la parte derecha un navegador de Laboratorios el cual cargara el laboratorio seleccionado y en el lado izquierdo de la pantalla encontrara un tabstrip con dos opciones.

Descripción: Detalla los objetivos de cada laboratorio y describe los pasos que tiene que seguir para realizar el laboratorio.

Lab#. Muestra el diseño prediseñado de cada laboratorio y el botón Calificar que califica la configuración que usted realizo en el laboratorio mostrándole una Ventana de Calificación donde aparece los porcentajes obtenidos.

Ventana de Calificación

Esta ventana muestra un reporte de los porcentajes obtenidos en el Subneteo y Uso de comandos, además presenta su respuesta comparada con la respuesta correcta y un porcentaje para cada ítem evaluado.






Evolución de laboratorios			
Planteamiento del problema			
Laboratorio No.5 IGRP Básico			
Resultados			
Subneteo : 100 %			
Interfaz	Su Respuesta	Respuesta correcta	Porcentaje
R1 Serial 0	Incorrecta1	210.45.1.33/27	5.26%
R1 Serial 1	Incorrecta2	210.45.1.65/27	5.26%
R1 Ethernet 0	Incorrecta3	192.168.1.1/24	5.26%
R2 Serial 1	Incorrecta4	210.45.1.62/27	5.26%
R2 Ethernet 0	Incorrecta5	192.168.100.1/24	5.26%
R3 Serial 0	Incorrecta6	210.45.1.94/27	5.26%
R3 Ethernet 0	Incorrecta7	192.168.20.1/24	5.26%
NIC1	Incorrecta8	192.168.1.2/24	5.26%
DNIC1	Incorrecta9	192.168.1.1	5.26%
NIC2	Incorrecta10	192.168.1.3/24	5.26%
DNIC2	Incorrecta11	192.168.1.1	5.26%
NIC3	Incorrecta12	192.168.100.2/24	5.26%
DNIC3	Incorrecta13	192.168.100.1	5.26%
NIC4	Incorrecta14	192.168.100.3/24	5.26%

Vista de Ventana Calificación

Barra de Dispositivos

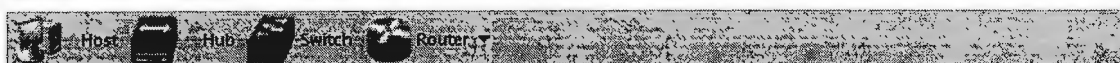
Esta barra se encuentra en la parte superior de la pantalla principal de diseño, donde usted dispone de 5 iconos que representan los dispositivos implementados en el simulador.

Dispositivos de Red

Icono	Nombre de Dispositivo	No de Interfaces		
		Ethernet	FastEthernet	Serial
	Host		1	
	Hub	4		
	Switch		4	
	Router 2501	1		2
	Router 2514	2		2

Para seleccionar un dispositivo de la barra y colocarlo en el área de diseño simplemente haga click sobre el dispositivo y luego haga otro click en el punto del área de diseño donde lo desea ubicar.

En el caso de seleccionar un router directamente desde el botón con el icono de router en la barra de dispositivos, el simulador agregará un router 2101, cuando se desea agregar un 2514 entonces seleccionarlo desde la flecha a la derecha de ese icono.



Vista de Barra de Dispositivos

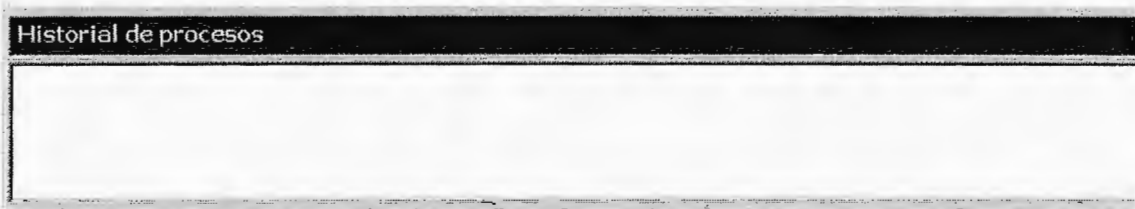
Barra de Historial

La barra de historia se habilita desde el Menú Ver, en la opción Barra de Historial. Cuando esta se carga se ubica en la parte inferior de la pantalla principal de diseño.

Una vez cargada usted podrá monitorear los eventos y procesos internos del simulador.

Detalle de eventos y procesos internos:

- Resolución de direcciones MAC
- Proceso de ping
- Proceso de trace
- Configuración de vlans en el switch
- Configuración de trunk en el switch
- Proceso de cableado de dispositivos
- Configuración de interfaz de router
- Procesos internos de router como ruteo de paquetes, actualización de tabla de ruteo, construcción de mensajes de actualización, envío de mensaje de actualización y expiración de ruta

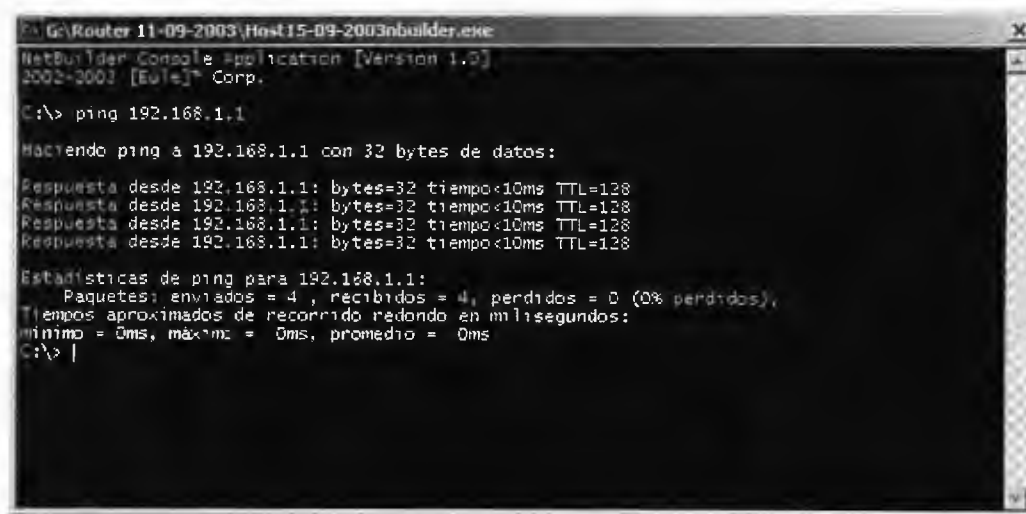


Vista de Barra de Historial

Consola

NETZBuilder te ofrece tres tipos de consolas en las cuales puedes probar conectividad con las herramientas ping y trace, configurar y ejecutar comandos de switch y de router.

Consola de Host



```
G:\Router 11-09-2003\Host15-09-2003nbuilder.exe
HostBuilder Console Application [Version 1.0]
2002-2003 [Edu] Corp.

C:\> ping 192.168.1.1

Haciendo ping a 192.168.1.1 con 32 bytes de datos:

Respuesta desde 192.168.1.1: bytes=32 tiempo<10ms TTL=128
Respuesta desde 192.168.1.1: bytes=32 tiempo<10ms TTL=128
Respuesta desde 192.168.1.1: bytes=32 tiempo<10ms TTL=128
Respuesta desde 192.168.1.1: bytes=32 tiempo<10ms TTL=128

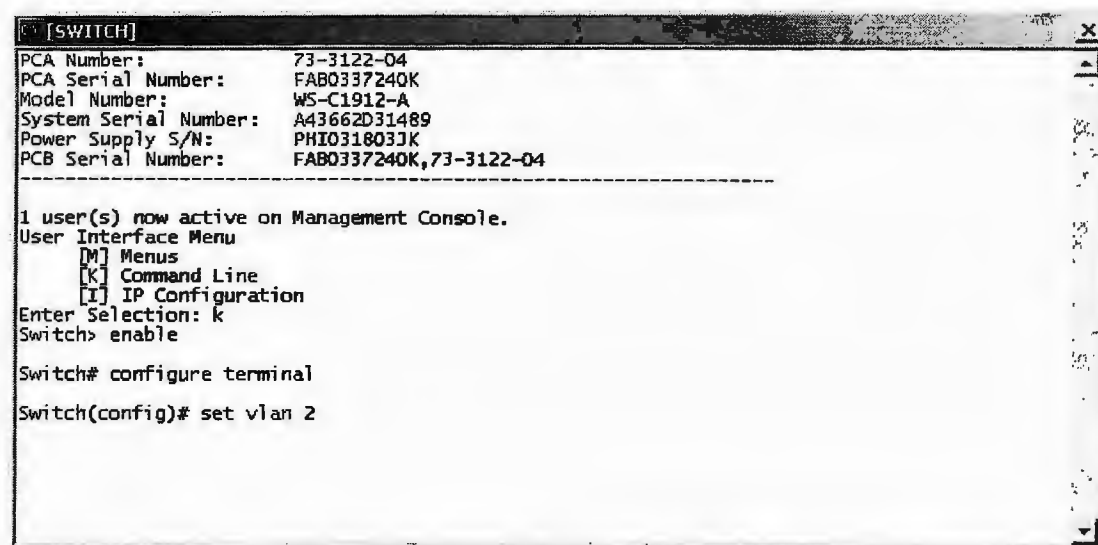
Estadísticas de ping para 192.168.1.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0 (0% perdidos),
    Tiempos aproximados de recorrido redondo en milisegundos:
    mínimo = 0ms, máximo = 0ms, promedio = 0ms
C:\> |
```

Vista de Consola de Host

La consola de host se inicia desde el menú contextual de host, en la opción **Mostrar Consola**.

Desde esta consola usted puede utilizar las herramientas ping y trace para probar la conectividad de la red.

Consola de Switch



```
[SWITCH]
PCA Number:      73-3122-04
PCA Serial Number: FAB0337240K
Model Number:    WS-C1912-A
System Serial Number: A43662D31489
Power Supply S/N:  PH1031803JK
PCB Serial Number: FAB0337240K,73-3122-04
-----
1 user(s) now active on Management Console.
User Interface Menu
  [M] Menus
  [K] Command Line
  [I] IP Configuration
Enter Selection: k
Switch> enable

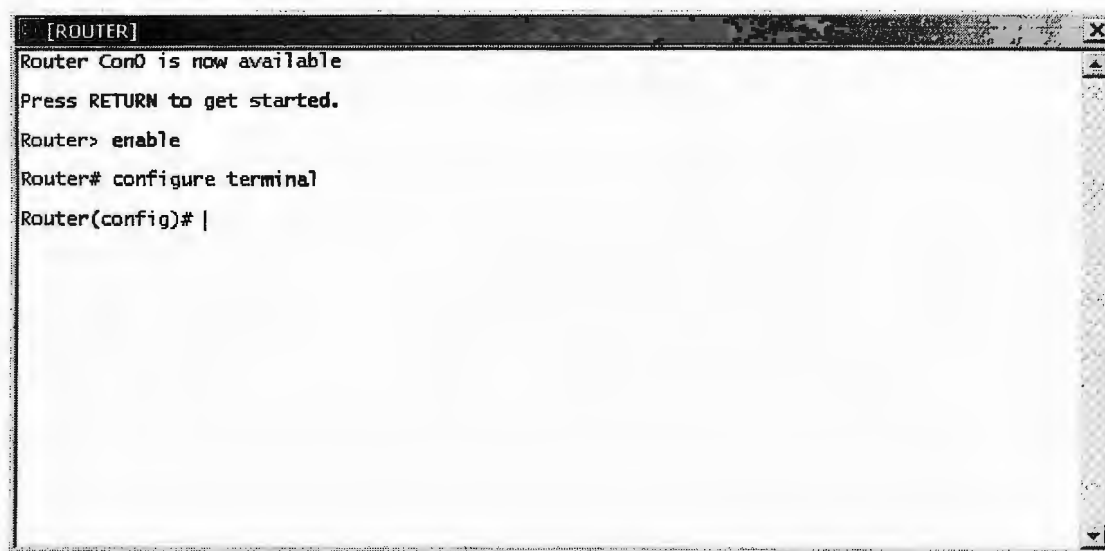
Switch# configure terminal
Switch(config)# set vlan 2
```

Vista de Consola de Switch

La consola de switch se inicia desde el menú contextual de switch, en la opción Configuración.

Desde esta consola usted puede utilizar todos los comandos disponibles para el switch, comandos anteriormente descritos. A través de esta consola se configura y observar el funcionamiento de cada switch.

Consola de Router



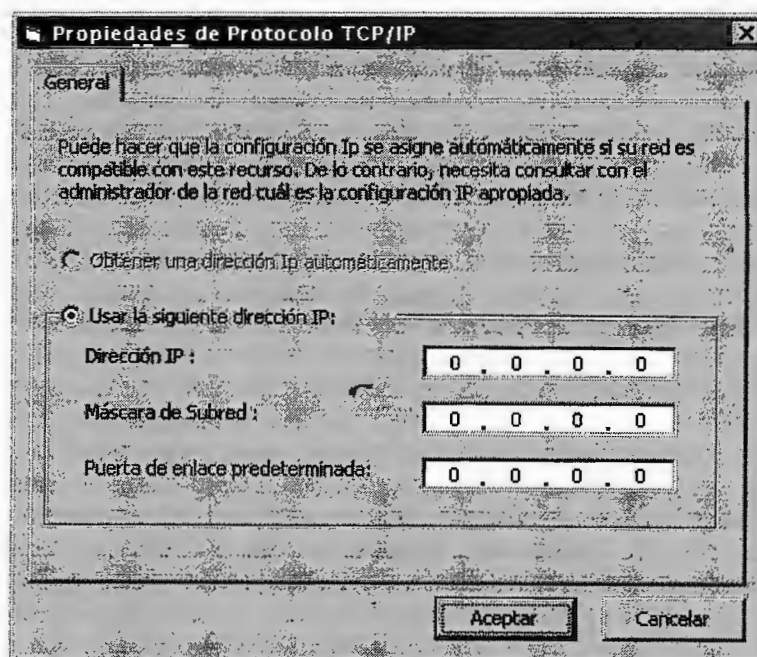
```
[ROUTER]
Router Con0 is now available
Press RETURN to get started.
Router> enable
Router# configure terminal
Router(config)# |
```

Vista de Consola de Router

La consola de router se inicia desde el menú contextual de router, en la opción Configuración.

Desde esta consola usted puede utilizar todos los comandos disponibles para el router, comandos anteriormente descritos. A través de esta consola se configura y observar el funcionamiento de cada router como la tabla de ruteo.

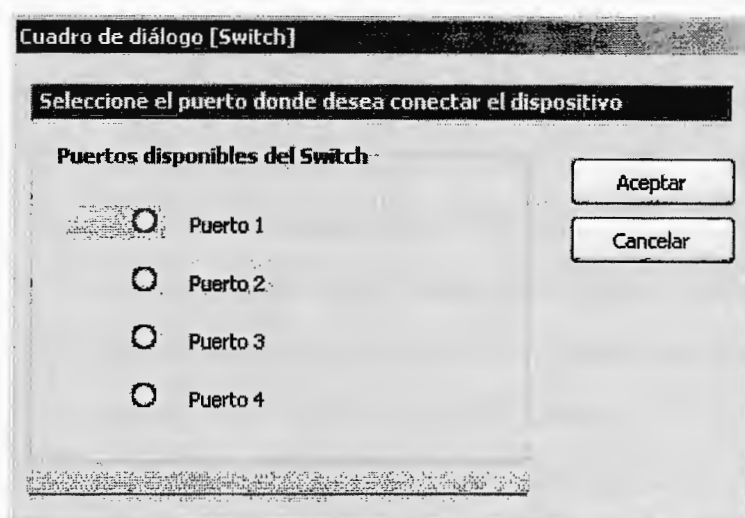
Ventana de Configuración de Host



Vista de Ventana de Configuración de Host

Esta ventana permite configurar un host con una Dirección IP, Mascara de Subred y Puerta de Enlace por Defecto. Con esta configuración el host esta en la capacidad de ser parte de la red.

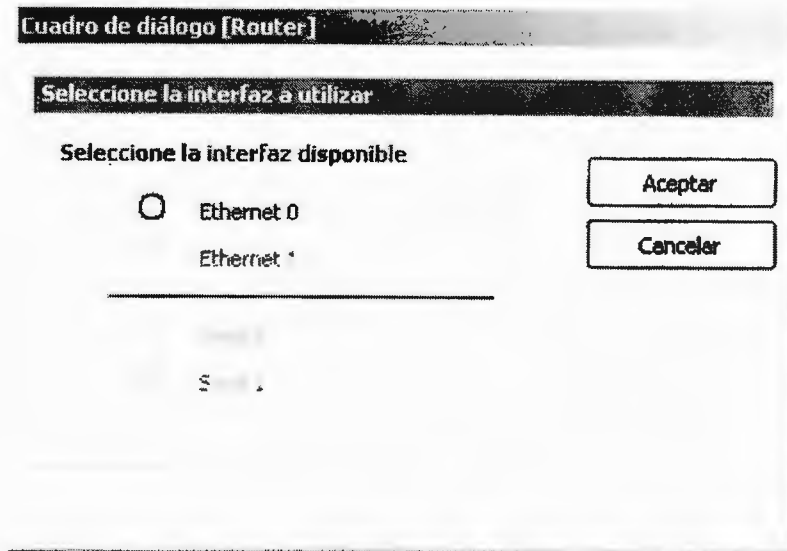
Ventana de Selección de Puertos de Switch



Vista de Ventana de Selección de Puertos

Esta ventana permite seleccionar el puerto donde desee conectar el cable que viene de la interfaz ethernet de un router y de la tarjeta de red de un host.

Ventana de Selección de Interfaz de Router



Vista de Ventana de Selección de Interfaz

Esta ventana permite seleccionar la interfaz de router donde desee conectar el cable que viene de un dispositivo LAN como una PC, Hub o Switch o también de otro router a través de sus seriales.

Cómo colocar los dispositivos en el área de diseño

Para colocar un dispositivo en el área de diseños, simple haga clic en el dispositivo que desea colocar de la barra de dispositivos, luego haga otro clic en la posición del área de diseño donde lo desee.

Cómo configurar un host

Para configurar un host en el modo de diseño, haga click derecho sobre el icono que representa al host, se mostrara el menú contextual, seleccione la opción Configurar.

Se mostrara la ventana de configuración de host con los campos Dirección IP, Mascara de Subred y Puerta de Enlace Predeterminada. Llene apropiadamente y haga click en Aceptar.

Cómo mover los dispositivos dentro del área de diseño

Para mover un dispositivo dentro del área de diseño, haga un simple drag and drop o arrastre y suelte en la nueva posición donde desee.

Cómo cablear un dispositivo con otro

Para cablear un dispositivo con otro, haga click derecho sobre el dispositivo que desea cablear. Aparecerá el menú contextual, seleccione la opción Cablear y luego mueva el mouse al otro dispositivo haciendo un click cuando este sobre el. Automáticamente aparecerá el cable conectando los dos dispositivos.

Para el caso cuando se conecte un host o router a un switch en el momento de hacer click sobre el dispositivo se cargara la ventana de selección de puertos de switch donde usted seleccionara un puerto disponible y luego haga clic en aceptar.

Al cablear Switch con Switch lo debe de hacer a través de puertos habilitados para trabajar con trunk, es por eso que primero debe configurar un puerto en cada switch a trunk y luego cablearlos sin necesidad de escoger en que puertos.

Cuando se conecten dispositivos como PC, Hub o Switch con un router aparecerá la Ventana de Selección de Interfaz para que usted seleccione la interfaz Ethernet 0 ó 1 del router según el modelo de router.

Al conectar dos routers aparecerá la Ventana de Selección de interfaz para seleccionar entre la interfaz Serial 0 ó 1 del router; para este caso en particular tome el siguiente ejemplo:

R1—R2 si va a conectar del router R1 al R2 le da clic derecho y la opción cablear en el R1, luego da clic izquierdo sobre el R2, la pantalla que aparecerá le pedirá en que interfase de R2 ira la conexión, al dar aceptar, le pedirá en que interfase del R1 conectara.

Cómo eliminar un dispositivo

Para eliminar un dispositivo del área de diseño, haga click derecho sobre el dispositivo que desea eliminar. Aparecerá el menú contextual, seleccione la opción Eliminar.

Nota: Al eliminar un dispositivo también se elimina el cable que esta conectado a el.

Cómo eliminar solo el cable

Para eliminar un cable, haga clic derecho sobre el cable; aparecerá el menú contextual, seleccione la opción eliminar.

Cómo Imprimir un proyecto

Para imprimir un proyecto usted tiene disponibles dos opciones:

- Imprimir Diseño: Imprime el diseño grafico activo.
- Imprimir Configuración: Imprime las configuraciones de los dispositivos activos.

Estas opciones están disponibles en el submenú Imprimir del Menú Archivo

Como guardar un proyecto

Una vez cargado un proyecto, usted tiene la opción de guardar su proyecto (Diseño y configuraciones de dispositivos) para abrirlo posteriormente, el proceso es el siguiente: Hacer clic en el Menú Archivo, luego apuntar a la opción Guardar. Aparecerá una ventana guardar donde puede seleccionar la ubicación del disco donde almacenará el archivo y luego coloque el nombre respectivo al archivo con extensión .ats (Area de Trabajo NETZBuilder). Finalmente hacer clic en el botón guardar.

Como abrir un proyecto

Si usted desea abrir un archivo de Area de Trabajo NETZBuilder (.ats), donde esta guardado un proyecto construido en Simulador de Red NETZBuilder. Haga clic en el Menú Archivo, luego apunte y haga clic en la opción Abrir. Aparecerá una ventana Abrir donde usted ubicara y seleccionara el archivo guardado. Una vez seleccionado, haga clic en Abrir e inmediatamente se cargara en pantalla.

Cómo Calificar un Laboratorio

Para calificar usted tiene que estar seguro que ya completo su laboratorio y luego hacer clic sobre el botón Calificar en Tab de Lab#. Se ejecutara el proceso de calificación automática, luego se presentara un reporte de Calificación donde usted encontrar los porcentajes obtenidos en subneteo y usos de comandos, la respuesta que usted escribió y la respuesta correcta para saber cual fue su error.

Cómo Utilizar las consolas

Las consolas de NETZBuilder fueron diseñadas para que simulen una consola real.

Consola de Host

Para hacer pruebas con herramientas ping y tracert.

Consola de Switch

Una vez cargada la consola, digite la tecla k para entrar al modo usuario de switch.

Los comandos disponibles para esta versión se mencionan en la Introducción de este manual.

Consola de Router

Una vez cargada la consola, oprima ENTER para entrar al modo usuario de router.

Los comandos disponibles para esta versión se mencionan en la Introducción de este manual.

2. MANUAL DEL PROGRAMADOR.

SISTEMA OPERATIVO.

El simulador de red NETZBuilder esta diseñado para ejecutarse bajo las siguientes especificaciones

Hardware	Mínimo	Optimo
Procesador (Mhz)	450	2000
Memoria (MB)	64	256
Disco Duro (MB)	300	2000

Requerimientos de Sistema Operativo

- Windows 98
- Windows NT
- Windows 2000
- Windows XP
- Windows 2003

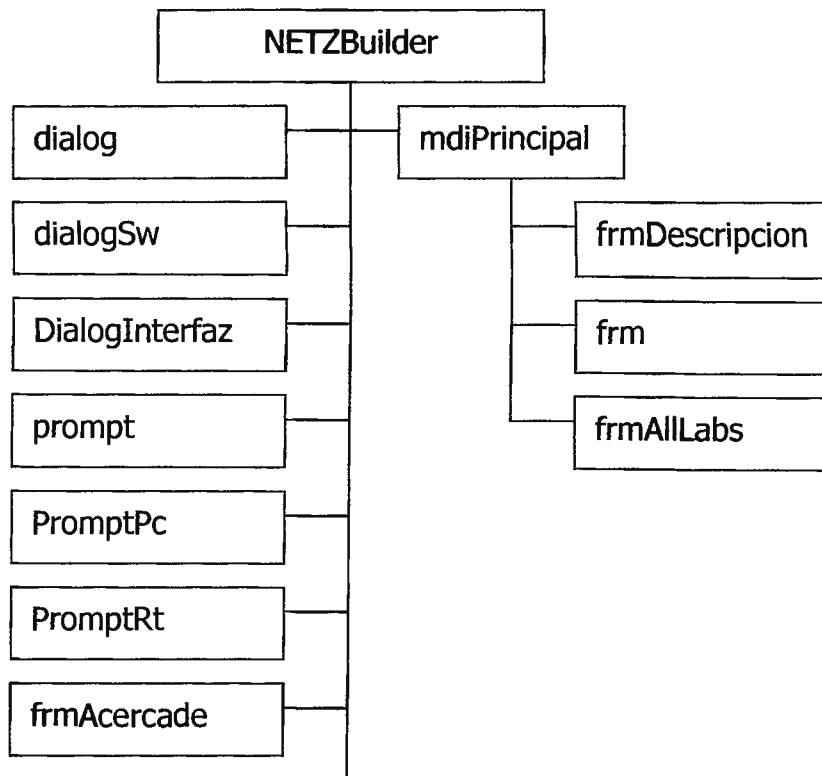
CODIGO FUENTE.

NETZBuilder esta desarrollado completamente en Visual Basic 6.0.

La programación está orientada en el modelo basado en objetos (clases) que representan los elementos de una red y de los componentes de los que están conformados. Los elementos gráficos son implementados mediante cuadros de imágenes y líneas.

Las consolas para la PC, Switch y Router son formularios que capturan los caracteres en un cuadro de texto.

Estructura de los formularios.



Los Objetos creados son los siguientes:

Objeto	Clase	Descripción
NIC o tarjeta de red	NIC.cls	<p>Clase que sirve de plantilla para la implementación de la tarjeta de red o NIC.</p> <p><u>Propiedades de la clase NIC:</u></p> <ul style="list-style-type: none"> • Mask: máscara de subred. • IPAddress: dirección IP. • Gateway: puerta de enlace. • MacAddress: Dirección MAC <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • Scan: hace una búsqueda en la red a través de un dispositivo. • Send: envía un mensaje a la red a través de un dispositivo. • Route: simula el proceso de ruteo. • Msg: simula la recepción de un paquete TCP/IP.

	PcNic.cls	<p>Clase para implementar la tarjeta de red en un host.</p> <p><u>Propiedades:</u></p> <ul style="list-style-type: none"> • Implementa las propiedades Mask, IPAddress, Gateway y macAddress de la clase NIC.cls • CablesIn y CablesOut: mantienen referencias a las líneas que representan el medio de conectividad. <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • Configure: verifica si la pcNIC esta configurada, de ser así, llama la función registerMac para registrar la NIC en el ARP local. • Anding: realiza la operación de ANDing para verificar a que red pertenece la dirección IP de la NIC configurada. • Implementa los métodos Scan, Send, Router y Msg de la clase NIC.cls • NewMac: genera una dirección MAC aleatoria para la NIC. • RegisterMac: registra la NIC en el ARP local.
	SerialNIC.cls	Implementa la interfaz serial del router.

	RouterNIC.cls	Implementa la interfaz ethernet del router.
Elementos de red	NetworkElements.cls	<p>Clase que se utiliza para crear y manejar los objetos de clase PC, Hub, Switch y Router.</p> <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • addPc: crea una instancia de la clase PC. • addHub: crea una instancia de la clase Hub. • addSwitch: crea una instancia de la claseSwitch. • addRouter: crea una instancia de la clase Router. • GetNode: devuelve la referencia de una instancia de cualquier clase. • RemoveNode: borra la instancia de una clase. • CountNodes: cuenta el número de instancias de clase creadas. • RemoveAll: Borra todas las instancias de clase creadas. • RetrieveAll: Devuelve todas las instancias de todos los objetos creados. • Exist: Verifica si se ha creado una instancia de un objeto.
Enlace	Link.cls	Clase que sirve de plantilla para la

		<p>implementación del hub.</p> <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • Plug: conectar un dispositivo • Scan: hace una búsqueda en la red a través de un dispositivo. • Send: envía un mensaje a la red a través de un dispositivo. • Unplug: desconecta un dispositivo. • Route: simula el proceso de enrutamiento.
Tabla ARP.	ArpRoutingTable.cls	<p>Implementación de la tabla ARP en una NIC.</p> <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • GetMac: Devuelve la dirección MAC de la dirección IP pasada como parámetro. • SetIp: Agrega una dirección MAC a la tabla de ruteo, estableciendo como clave de búsqueda al dirección IP. • RemoveIp: remueve una dirección Mac de la tabla ARP.
Tabla ARP del un switch	SwitchArpTable.cls	<p>Implementación de la tabla ARP en un switch.</p> <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • Getport: devuelve el puerto al que está conectado un host a

		<p>partir de la dirección Mac de esta pasada como parámetro.</p> <ul style="list-style-type: none"> • Setport: Establece el puerto al que se vinculará una dirección MAC. • CleanPort: borra una MAC de la tabla ARP para el puerto especificado. • GetData: devuelve todo el contenido de la tabla ARP.
Host	Pc.cls	<p>Clase que representa un host dentro del programa.</p> <p><u>Propiedades:</u></p> <ul style="list-style-type: none"> • card: variable de tipo NIC para representar la tarjeta de red. <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • Scan: realiza la búsqueda de una IP en el ARP local. • Ping: realiza el proceso de hacer ping. • SetupNIC: configura una NIC con una dirección IP, máscara de subred y una puerta de enlace.
Hub	Hub.cls	<p>Clase que representa a un hub o concentrador dentro del programa:</p>

		<p><u>Propiedades:</u></p> <ul style="list-style-type: none"> • Jackets: variable de tipo Collection que representa los puertos del Hub. • cablesIn y cablesOut: mantienen referencias a las líneas que representan el medio de conectividad. • SerialNumber: es un número aleatorio que identifica al hub dentro del programa. <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • Link_plug: realiza el proceso vinculación de una PC a un puerto del hub. • Link_unplug: desvincula un host de un puerto del hub. • Implementa la clase link.cls para usar los metodos Link_scan, Link_send y link_route. • newSerial: genera un número de serie aleatorio.
Switch	Switch.cls	<p>Representa un switch dentro del programa.</p> <p><u>Propiedades:</u></p> <ul style="list-style-type: none"> • Jackets: arreglo del tipo Tport, que representa los puertos del

		<p>switch.</p> <ul style="list-style-type: none"> • ArpTable: maneja la tabla ARP del Switch. • cablesIn y cablesOut: mantienen referencias a las líneas que representan el medio de conectividad. • SerialNumber: es un número aleatorio que identifica al hub dentro del programa. • Hostname: representa el nombre del switch. Este puede ser cambiado mediante la consola de comandos. • password: contraseña dada al switch. • MTrunk: contiene el número del puerto configurado para el trunk. <p><u>Funciones y métodos:</u></p> <ul style="list-style-type: none"> • Implementa la clase link.cls para usar los métodos Link_scan, Link_send, link_route y link_unplug. • PlugSwitch: crea una referencia a otro objeto switch. • UnplugSwitch: borra la referencia al objeto switch. • TrunckedScan: hace la
--	--	--

		<p>búsqueda de un dispositivo en el switch conectado.</p> <ul style="list-style-type: none"> • TrunckedSend: envía mensajes a un dispositivo en el lado del switch conectado. • Setvlan: crea una vlan. • setvlan2: establece el puerto a una vlan. • ShowvLan: devuelve información acerca de todas las vlans creadas. • show_vlan_membership: devuelve la información de membresía de las vlans creadas (que puertos pertenecen a una vlan) • ShowTable: devuelve información de la tabla MAC del switch. • ShowInterface: devuelve la información de las interfaces del switch
Router	Router.cls	<p>Representa un router dentro del programa.</p> <p><u>Propiedades:</u></p> <ul style="list-style-type: none"> • Ethernet0: representa una interfaz ethernet(de tipo routenic).

		<ul style="list-style-type: none"> • Ethernet1: representa una interfaz ethernet. • Serial0: representa una interfaz serial (de tipo serialnic). • Serial1: representa una interfaz serial. • Model: indica el modelo que se esta usando, ya sea el modelo 2501 o el 2514. • Secret: devuelve o establece la contraseña secreta. <p><u>Métodos:</u></p> <ul style="list-style-type: none"> • addRipNetwork: agrega una red usando el protocolo RIP. • addIgrpNetwork: agrega una red usando el protocolo IGRP. • update: Simula el proceso de actualización de una tabla de ruteo. • Implementa la clase link.cls para usar los métodos Link_scan, Link_send y link_route. • Link_unplug: desconecta un dispositivo de una interfaz. • Anding: realiza el proceso de comprobación ANDing.
Routernic	Routernic.cls	Representa la interfaz ethernet de un router.

		<p><u>Propiedades:</u></p> <ul style="list-style-type: none"> • NIC_mask: implementación de la clase NIC.cls. Es utilizada como máscara de subred de la interfaz. • NIC_MacAddress: implementación de la clase NIC.cls. Es utilizada como dirección mac de la interfaz. • NIC_IPAddress: implementación de la clase NIC.cls. Es utilizada como dirección IP de la interfaz. • NIC_gateway: implementación de la clase NIC.cls. Es utilizada como puerta de enlace predeterminada en un router. <p><u>Métodos:</u></p> <ul style="list-style-type: none"> • Implementa los métodos NIC_scan, NIC_send y NIC_route de la clase NIC.cls. • newMac: genera una nueva y única dirección MAC para la interfaz. • Anding: realiza el proceso de comprobación ANDing.
RoutingTable	RoutingTable.cls	Representa los procesos de actualización de tabla de ruteo, construcción de mensaje de

		actualización y envío de mensaje de actualización.
RoutingTableRecord	RoutingTableRecord.cls	Se definen las propiedades de la tabla de ruteo, se inicializan los timers, definición de mensaje de actualización y expiración rutas.
serialnic	Serialnic.cls	<p>Representa la interfaz serial de un router.</p> <p><u>Propiedades:</u></p> <ul style="list-style-type: none"> • Implementa las propiedades NIC_mask, NIC_MacAddress, NIC_IPAddress de la clase NIC.cls. <p><u>Métodos:</u></p> <ul style="list-style-type: none"> • Implementa los métodos NIC_scan, NIC_send y NIC_route de la clase NIC.cls. • newMac: genera una nueva y única dirección MAC para la interfaz. • Anding: realiza el proceso de comprobación ANDing.
Control Tooltip	CTooltip.cls	Módulo de clase que genera un toolTip con múltiples líneas para los controles, ya que Visual Basic no permite mostrar más de una línea.
CcrpTimer	Ccrptmr6.dll	Control ActiveX DLL de un contador Timer mejorado

Controles de usuario.

Nombre	Descripción
IPAddress	Control de usuario utilizado para capturar una dirección IP.
LiteBtnCtrl	Control de usuario utilizado como botón de comando.

API

Se utilizaron las siguientes API's de Windows para el manejo de archivos de la aplicación:

Escribe una cadena en una sección del archivo especificada:

```
Public Declare Function WritePrivateProfileString Lib "kernel32" Alias
"WritePrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As
Any, ByVal lpString As Any, ByVal lpFileName As String) As Long
```

Recupera una cadena de la sección del archivo especificada:

```
Public Declare Function GetPrivateProfileString Lib "kernel32.dll" Alias
"GetPrivateProfileStringA" (ByVal lpApplicationName As String, ByVal lpKeyName As
Any, ByVal lpDefault As String, ByVal lpReturnedString As String, ByVal nSize As
Long, ByVal lpFileName As String) As Long
```

Recupera todos los nombres de las secciones en un archivo:

```
Public Declare Function GetPrivateProfileSectionNames Lib "kernel32.dll" Alias
"GetPrivateProfileSectionNamesA" (ByVal lpzReturnBuffer As String, ByVal nSize As
Long, ByVal lpFileName As String) As Long
```


FORMULARIOS

El programa cuenta con los siguientes formularios, con los que el usuario interactúa durante la ejecución de la aplicación:

Nombre del formulario	Descripción
DialogConf	Ventana de TCP/IP.
DialogInterfaz	Ventana utilizada para establecer la interfaz donde se conectará en dispositivo.
DialogConfSw	Ventana de configuración del puerto a ocuparse en un switch.
Frm	Formulario donde se podrán crear las redes personalizadas.
FrmAcercaDe	Formulario que muestra la información acerca del programa.
FrmAllLabs	Formulario de donde se realizan todos los laboratorios prediseñados.
FrmDescripcion	Muestra los resultados de la evaluación en un laboratorio prediseñado.
MDIPrincipal	Formulario inicial. A partir de este formulario principal se llamarán el resto de formularios.
Prompt	Representa la consola comandos del switch.
PromptPc	Representa la consola de comandos de un host.
PromptRt	Representa la consola de comandos del router.

FUNCIONES

Función	Descripción	Referencia
----------------	--------------------	-------------------

openFile	Función para abrir un archivo ats.	Frm.frm
CreateElement	Función para crear un nuevo elemento en el pool de elementos.	Frm.frm
CurrentElement	Devuelve la referencia de un objeto.	Frm.frm
GetIniSections	Recuperar todas las claves del archivo ats.	Frm.frm
saveFile	Función para guardar un archivo del área de trabajo.	Frm.frm
setFree	Reestablece todas las variables globales a nivel de formulario, descarga todos las imágenes en pantalla y borra la referencia de todos los objetos.	Frm.frm, mdiPrincipal.frm
LoadLab	Carga un laboratorio prediseñado.	FrmAllLabs.frm
EjecutarComando	Función que se utiliza para verificar y ejecutar un comando.	Prompt.frm, promptPc.frm y promptRt.frm
OpenFile	Función que se utiliza para abrir el archivo de mensajes y devolver el texto correspondiente al mensaje solicitado.	MdDeclaracion.bas

Delay	Función que se utiliza para hacer una pausa.	MdDeclaracion.bas
Ping	Función que simula en comando ping.	Pc.cls
Send	Simula el envío de un paquete por TCP/IP.	NIC.cls
scan	Simula un broadcast dentro de la red.	NIC.cls
Msg	Simula la recepción de un paquete por TCP/IP.	NIC.cls
addPC	Crea en memoria un nuevo host.	NetworkElements.cls
addHub	Crea en memoria un nuevo hub.	NetworkElements.cls
addSwitch	Crea en memoria un nuevo Switch.	NetworkElements.cls
addRouter	Crea en memoria un nuevo Router.	NetworkElements.cls

CODIGO FUENTE

Procedimiento CreateElement

```
Public Sub CreateElement(Index As Integer, Optional idElement As String)
```

```
    Select Case Index - 1
```

```
        Case 0 ' AGREGAR PC
```

```
            If idElement = "" Then
```

```
                Set pivotElement = pool.addPc
```

```
            Else
```

```
Set pivotElement = pool.addPc(idElement)
```

```
End If
```

```
cmdElemento.Buttons(1).Tag = True
```

```
PCCounter = PCCounter + 1
```

```
If PCCounter = 24 Then cmdElemento.Buttons(1).Enabled = False
```

```
Creating = True
```

Case 1 ' AGREGAR HUB

```
If idElement = "" Then
```

```
Set pivotElement = pool.addHub
```

```
Else
```

```
Set pivotElement = pool.addHub(idElement)
```

```
End If
```

```
hubSwitchCounter = hubSwitchCounter + 1
```

```
cmdElemento.Buttons(2).Tag = True
```

```
Creating = True
```

```
If hubSwitchCounter = 6 Then
```

```
cmdElemento.Buttons(2).Enabled = False
```

```
cmdElemento.Buttons(3).Enabled = False
```

```
End If
```

Case 2 ' AGREGAR SWITCH

```
If idElement = "" Then
```

```
Set pivotElement = pool.addSwitch
```

```
Else
```

```
Set pivotElement = pool.addSwitch(idElement)
```

```
End If
```

```
hubSwitchCounter = hubSwitchCounter + 1
```

```

cmdElemento.Buttons(3).Tag = True
If hubSwitchCounter = 6 Then
    cmdElemento.Buttons(2).Enabled = False
    cmdElemento.Buttons(3).Enabled = False
End If
Creating = True

```

Case 3 ' AGREGAR ROUTER

```

If idElement = "" Then
    Set pivotElement = pool.addRouter
Else
    Set pivotElement = pool.addRouter(idElement)
End If

```

```

pivotElement.Model = "2501"
RouterCounter = RouterCounter + 1

```

```

cmdElemento.Buttons(4).Tag = True
If RouterCounter = 6 Then
    cmdElemento.Buttons(4).ButtonMenus(1).Enabled = False
    cmdElemento.Buttons(4).ButtonMenus(2).Enabled = False
    'cmdElemento.Buttons(5).Enabled = False
End If
Creating = True

```

Case 4 ' AGREGAR ROUTER

```

If idElement = "" Then
    Set pivotElement = pool.addRouter
Else
    Set pivotElement = pool.addRouter(idElement)

```

End If

RouterCounter = RouterCounter + 1

pivotElement.Model = "2514"

cmdElemento.Buttons(4).Enabled = False

cmdElemento.Buttons(4).Tag = True

If RouterCounter = 6 Then

cmdElemento.Buttons(3).Enabled = False

cmdElemento.Buttons(4).Enabled = False

End If

Creating = True

End Select

End Sub

Procedimiento MouseUp

Private Sub Form_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)

Dim i As Integer

' AGREGAR UN HOST

If Creating Then

Creating = False

m_Index = m_Index + 1

Load imagesPool(m_Index)

With imagesPool(m_Index)

.Top = Y

.Left = X

.Enabled = True

.Visible = True

```
If CBool(cmdElemento.Buttons(1).Tag) Then
    .Picture = LoadPicture(App.Path & "\computer.jpg")
    .Tag = pivotElement.card.NIC_MacAddress
    cmdElemento.Buttons(1).Tag = False
    Set pivotElement = Nothing
    cmdElemento.Buttons(1).Enabled = True
End If
```

'AGREGAR UN HUB

```
If CBool(cmdElemento.Buttons(2).Tag) Then
    .Picture = LoadPicture(App.Path & "\hub.jpg")
    .Tag = pivotElement.SerialNumber
    cmdElemento.Buttons(2).Tag = False
    Set pivotElement = Nothing
    'cmdElemento.Buttons(1).Enabled = True
End If
```

' AGREGAR UN SWITCH

```
If CBool(cmdElemento.Buttons(3).Tag) Then
    .Picture = LoadPicture(App.Path & "\switch.jpg")
    .Tag = pivotElement.SerialNumber
    cmdElemento.Buttons(3).Tag = False
    Set pivotElement = Nothing
    'cmdElemento.Buttons(1).Enabled = True
End If
```

' AGREGAR UN ROUTER

```
If CBool(cmdElemento.Buttons(4).Tag) Then
    .Picture = LoadPicture(App.Path & "\router.jpg")
```

```
.Tag = pivotElement.SerialNumber  
cmdElemento.Buttons(4).Tag = False  
Set pivotElement = Nothing
```

```
End If
```

```
End With
```

```
If PCCounter < 24 Then cmdElemento.Buttons(1).Enabled = True
```

```
If hubSwitchCounter < 6 Then
```

```
    cmdElemento.Buttons(2).Enabled = True
```

```
    cmdElemento.Buttons(3).Enabled = True
```

```
End If
```

```
If RouterCounter < 5 Then
```

```
    cmdElemento.Buttons(4).Enabled = True
```

```
    'cmdElemento.Buttons(4).Enabled = True
```

```
End If
```

```
cmdElemento.Enabled = True
```

```
End If
```

```
If Cabling And LineCount > 1 Then
```

```
    Unload Medio(LineCount)
```

```
    Cabling = False
```

```
End If
```

```
End Sub
```


La consola.

La consola esta formada por un textBox que captura los comandos. El texto digitado es almacenado en la propiedad TAG en el evento KeyPress del textbox.

Existe una variable llamada LastEnter que mantiene la última posición del cursor. A partir de esa posición, se empieza a capturar los nuevos comandos.



Los pasos son:

- Cada vez que se digita un carácter se llama el evento KeyPress. Al presionar la tecla Enter, se extrae el texto digitado desde la posición indicada por lastEnter hasta el final del textbox (a excepción de los caracteres Cr y Lf); y se establece la nueva posición del cursor (lastEnter es igual a la longitud del textBox).

```
Private Sub Text1_KeyPress(KeyAscii As Integer)
    erasing = False
    If KeyAscii = vbKeyReturn Then
        text1.Tag = Mid(text1, lastEnter + 1) 'Extrae el texto digitado a partir de
        ' la ultima posición indicada por lastEnter
        text1.SelStart = Len(text1) 'mueve el cursor hasta el final del textBox
        lastEnter = Len(text1) + 2 'establece la nueva posicion y
    ElseIf KeyAscii = vbKeyBack Then ' si esta borrando un caracter
        If Len(text1) = lastEnter Then KeyAscii = 0
        erasing = True
    End If
End Sub
```

- Después de escribir un carácter y haber ejecutado el evento KeyPress, se ejecuta el evento Change del textbox, si lastEnter es igual a la longitud del textBox (valor establecido al presionar Enter) entonces, indica que se ha terminado de digitar un comando y hay que ejecutarlo, llamando a la function EjecutarComando.

```
Private Sub Text1_Change()  
    Dim result As Integer  
    curpos = text1.SelStart  
    If lastEnter = Len(text1) And Not erasing Then  
        result = EjecutarComando  
        text1 = text1 + prompt  
        lastEnter = lastEnter + Len(prompt) + result  
        text1.SelStart = Len(text1)  
        curpos = text1.SelStart  
    End If  
End Sub
```

- en la función EjecutarComando, se evalúa el texto digitado y comprueba si lo que se digitó fue un comando válido.

```
Public Function EjecutarComando() As Integer  
    Dim CommandLine() As String, strMessage As String  
    Dim ipAddress() As String, count As Integer  
    Dim sInput As String, i As Integer  
    sInput = Trim(text1.Tag)  
    CommandLine = Split(sInput) ` Divide la cadena, cada indice es una palabra  
  
    If UBound(CommandLine) >= 1 Then
```

```

If LCase(CommandLine(0)) = "ping" Then ` Si la primera palabra digitada
` fue ping... entonces, comprobar si se digito correctamente la IP.
    ipaddress = Split(CommandLine(1), ".")
    'If UBound(ipAddress) = 3 Then
    count = 0
    For i = 0 To UBound(ipaddress)
        If Val(ipaddress(i)) >= 0 And Val(ipaddress(i)) <= 255 Then
            count = count + 1
        End If
    Next
    If count = 4 Then ` si la IP es valida
        text1.Locked = True
        aux.ping CommandLine(1) ` llamar la función Ping
        text1.Locked = False
    Else ` Si la IP no fue valida, escribir un mensaje de error.
        Delay (2)
        strMessage = "La solicitud de ping no pudo encontrar el host " &
        CommandLine(1) & ". Compruebe el nombre " _
        & "y vuelva a intentarlo." & vbCrLf & vbCrLf
    End If
    'ElseIf CommandLine(0) = "tracert" Then
    Else
        ` si el comando, no es un comando conocido, escribir el error
        strMessage = "" & CommandLine(0) & "" no se reconoce como un
        comando interno o externo," & vbNewLine _
        & "programa o archivo por lotes ejecutable." & vbCrLf & vbCrLf
    End If
    ElseIf UBound(CommandLine) = 0 Then
        If LCase(CommandLine(0)) = "exit" Then
            Unload Me

```

Exit Function

End If

End If

text1.Tag = ""

text1 = text1 & strMessage

EjecutarComando = Len(strMessage)

End Function

Función PING

Public Sub ping(target As String)

Dim i As Integer, recibidos As Integer, A() As String

TTL = 128

If target = card.NIC_IPAddress Then

mdiPrincipal.txtHistorial = mdiPrincipal.txtHistorial & "[" + card.NIC_IPAddress + "]" + "Pinging " + target & vbNewLine

frmConsolaPc.text1 = frmConsolaPc.text1 & vbNewLine & "Haciendo ping a " & target & " con 32 bytes de datos:" & vbNewLine & vbNewLine

recibidos = 0

For i = 0 To 3

mdiPrincipal.txtHistorial = mdiPrincipal.txtHistorial & " Recibiendo
respuesta desde: " + target & vbNewLine

Delay 1

frmConsolaPc.text1 = frmConsolaPc.text1 & "Respuesta desde " & target & ":
bytes=32 tiempo<10ms TTL=" & TTL & vbNewLine

recibidos = recibidos + 1

frmConsolaPc.text1.SelStart = Len(frmConsolaPc.text1)

Next i

```
frmConsolaPc.text1 = frmConsolaPc.text1 & vbCrLf & "Estadísticas de ping
para " & target & ":" & vbCrLf
```

```
frmConsolaPc.text1 = frmConsolaPc.text1 & "    Paquetes: enviados = 4 ,
recibidos = " & recibidos & ", perdidos = " & 4 - recibidos & " (" & Format(1 -
recibidos / 4, "#0%") & " perdidos)," & vbCrLf
```

```
frmConsolaPc.text1 = frmConsolaPc.text1 & "Tiempos aproximados de recorrido
redondo en milisegundos:" & vbCrLf & _
```

```
    "mínimo = 0ms, máximo = 0ms, promedio = 0ms" & vbCrLf
```

```
frmConsolaPc.lastEnter = Len(frmConsolaPc.text1)
```

```
frmConsolaPc.text1.SelStart = Len(frmConsolaPc.text1)
```

```
frmConsolaPc.curpos = frmConsolaPc.lastEnter
```

```
mdiPrincipal.txtHistorial.SelStart = Len(mdiPrincipal.txtHistorial)
```

```
Exit Sub
```

```
End If
```

```
A = Split(target, ".")
```

```
For i = 0 To UBound(A)
```

```
    If Not IsNumeric(A(i)) Then
```

```
        frmConsolaPc.text1 = frmConsolaPc.text1 & "Host desconocido " & target &
vbCrLf
```

```
        frmConsolaPc.lastEnter = Len(frmConsolaPc.text1)
```

```
        frmConsolaPc.text1.SelStart = Len(frmConsolaPc.text1)
```

```
        frmConsolaPc.curpos = frmConsolaPc.lastEnter
```

```
        Exit Sub
```

```
    End If
```

```
Next
```

```
mdiPrincipal.txtHistorial = mdiPrincipal.txtHistorial & "[" + card.NIC_IPAddress +
"] " + "Pinging " + target & vbCrLf
```

```

frmConsolaPc.text1 = frmConsolaPc.text1 & vbNewLine & "Haciendo ping a " &
target & " con 32 bytes de datos:" & vbNewLine & vbNewLine
recibidos = 0

For i = 0 To 3
    If card.NIC_send(target) Then
        mdiPrincipal.txtHistorial = mdiPrincipal.txtHistorial & "          Recibiendo
respuesta desde: " + target & vbNewLine
        Delay 1
        frmConsolaPc.text1 = frmConsolaPc.text1 & "Respuesta desde " & target & ":
bytes=32 tiempo=60ms TTL=" & TTL & vbNewLine
        recibidos = recibidos + 1
    Else
        Delay 1
        frmConsolaPc.text1 = frmConsolaPc.text1 & "Tiempo de espera agotado para
esta solicitud." & vbNewLine
    End If
    frmConsolaPc.text1.SelStart = Len(frmConsolaPc.text1)
    mdiPrincipal.txtHistorial.SelStart = Len(mdiPrincipal.txtHistorial)
Next i
frmConsolaPc.text1 = frmConsolaPc.text1 & vbNewLine & "Estadísticas de ping
para " & target & ":" & vbNewLine
frmConsolaPc.text1 = frmConsolaPc.text1 & "    Paquetes: enviados = 4 , recibidos
= " & recibidos & ", perdidos = " & 4 - recibidos & " (" & Format(1 - recibidos / 4,
"#0%") & " perdidos)," & vbNewLine
frmConsolaPc.text1 = frmConsolaPc.text1 & "Tiempos aproximados de recorrido
redondo en milisegundos:" & vbNewLine & _
    "mínimo = 60ms, máximo = 60ms, promedio = 60ms" & vbNewLine
frmConsolaPc.lastEnter = Len(frmConsolaPc.text1)
frmConsolaPc.text1.SelStart = Len(frmConsolaPc.text1)

```

```
frmConsolaPc.curpos = frmConsolaPc.lastEnter
```

```
End Sub
```

Función Send.

```
Public Function NIC_send(ByVal target As String) As Boolean
```

```
    Dim TargetMac As String
```

```
    NIC_send = False
```

```
    If Enabled Then
```

```
        If Anding(target) Then
```

```
            TargetMac = RoutingTable.getMac(target)
```

```
            If TargetMac = "" Then
```

```
                TargetMac = device.scan(target, p_IPAddress, p_macAddress)
```

```
            End If
```

```
            If TargetMac <> "" Then
```

```
                RoutingTable.setIp target, TargetMac
```

```
                If device.send(TargetMac, p_macAddress) Then
```

```
                    NIC_send = True
```

```
                Else
```

```
                    mdiPrincipal.txtHistorial = mdiPrincipal.txtHistorial & "      Tiempo de  
respuesta excedido [No Encontrada]" & vbNewLine
```

```
                End If
```

```
            Else
```

```
                mdiPrincipal.txtHistorial = mdiPrincipal.txtHistorial & "      Tiempo de  
respuesta excedido [No Encontrada]" & vbNewLine
```

```
            End If
```

```
        Else
```

```
            If routable Then
```

```
                If Anding(p_Gateway) Then
```

```
                    TargetMac = RoutingTable.getMac(p_Gateway)
```

```

        If TargetMac = "" Then TargetMac = device.scan(p_Gateway,
p_IPAddress, p_macAddress)
        If TargetMac <> "" Then RoutingTable.setIp p_Gateway, TargetMac
        'Funcion de Ruteo!!!
        mdiPrincipal.txtHistorial = mdiPrincipal.txtHistorial & " [" +
p_IPAddress + "]" Enviando mensaje a Router" & vbNewLine
        device.route p_macAddress, TargetMac, target
    End If
End If
End If
End If
End Function

```

Función MSG.

```

Public Function NIC_msg(ByVal target As String) As Boolean
    NIC_msg = Enabled And (target = p_macAddress)
End Function

```

Función addPC

```

Public Function addPc(Optional strPcMac As String) As PC
    Dim newPc As PC
    Set newPc = New PC
    If strPcMac = "" Then
        While Nodes.Exists(newPc.card.NIC_MacAddress)
            newPc.card.newMac
        Wend
    Else
        newPc.card.NIC_MacAddress = strPcMac
    End If

```


End Function

TIPOS DE DATOS, VARIABLES Y CONSTANTES.

Tipo/Variable/Constante	Descripción	Referencia
Scripting dictionary	Un dictionary es el equivalente a un arreglo asociativo en Perl. Los elementos del diccionario pueden ser de cualquier tipo y están asociados a una clave única.	ArpRoutingTable.cls, SwitchArpTable.cls, NetworkElements.cls,
Collection		Hub.cls, PcNic.cls, router.cls, routernic.cls, routingtable.cls, switch.cls
vLanStruct	Tipo de dato creado para manejar la información de la vlan	Switch.cls
TPort	Tipo de dato creado para manejar la información de un puerto del switch.	Switch.cls
pool	De tipo networkElements, contiene información de todos los elementos creados (hosts, hubs, switches y routers)	Frm.frm
NType = "Fast Ethernet"	Variables que especifica el tipo de puerto que utiliza el switch	Switch.cls
CommandLine	Arreglo que contiene el	Consolas

	comando y sus parámetros digitados en la consola.	
curElement	Variable que mantiene el identificador único del objeto actual objeto.	Fmr.frm y frmAllLab.frm
aux	Variable temporal que mantiene la referencia del objeto actual extraído del pool.	Todo el proyecto.
tmp	Variable temporal que mantiene la referencia del anterior objeto al actual que fue seleccionado.	Frm.frm
vbKeyLeft	Constante de Visual Basic que representa la tecla de dirección la izquierda.	
vbKeyRight	Constante de Visual Basic que representa la tecla de dirección la derecha.	
vbKeyUp	Constante de Visual Basic que representa la tecla de dirección hacia arriba.	
vbKeyDown	Constante de Visual Basic que representa la tecla de dirección hacia abajo.	
vbKeyEnd	Constante de Visual Basic que representa la tecla END.	

vbKeyHome	Constante de Visual Basic que representa la tecla HOME	
vbKeyReturn	Constante de Visual Basic que representa el Enter.	Consolas
vbKeyBack	Constante de Visual Basic que representa la tecla BackSpace	
vbCrLf	Constante de Visual Basic que representa un salto de línea con retorno de carro.	Todo el proyecto.