

UNIVERSIDAD AUTONOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS



**Simulación de Redes de Comunicaciones de Cobertura Amplia Basado en
Algoritmos de Enrutamiento Dinámico**

TESIS PROFESIONAL

Que para obtener el título de

Licenciado en Ciencias Computacionales

Presenta

Héctor Javier Jiménez Pensado

Ensenada, Baja California. Abril 2000

DEDICATORIA.

Todo el esfuerzo y trabajo de esta tesis lo dedico a mi familia con toda la admiración y amor que se pueda ofrecer.

A mi madre:

Kenia Pensado de Jiménez.

Esta persona es la que me mostró el perfecto ejemplo de salir adelante a pesar de cualquier circunstancia, de saber aprovechar las oportunidades, tomar decisiones y de gozar de la vida aún en situaciones difíciles. Gracias a su gran forma de pensar, de tomar las cosas y ese espíritu energético de siempre seguir, me ha hecho lograr metas y propósitos como el de la realización de esta tesis. Reconozco el esfuerzo, la paciencia y sobre todo la fe que me dió a través de mis estudios y mi vida, proporcionándome no sólo con cosas materiales sin importar cómo ó porqué, sino con lo que realmente importa: amor. Gracias a tí madre, por todo.

A mi padre:

Gustavo Francisco Jiménez y Lima

Me faltó conocerte más y que me conocieras más, quisiera poder demostrarte que estoy logrando metas en mi vida y que reconocieras el esfuerzo de mi madre en mí y en mi hermano. Pero sé que Dios se encarga de la última demostración en cada uno y que te toca lo que te mereces.

A mi hermano:

Gustavo Alfonso Jiménez Pensado

De un tiempo a otro, ahora más que antes, reconozco lo que pasaste, sé de tu capacidad y de tu indiferencia hacia las cosas que aunque no lo creas trae una indirecta influencia sobre mí en mi crecimiento y desarrollo. Esta tesis va por tí, para demostrarte que sí sé lo que significa romper barreras, tal y como tú lo haces y puedes hacer mejor que yo.

AGRADECIMIENTOS.

Agradezco a **Dios** sobre todas las cosas, por es EL quien me ha abierto el camino para lograr todo en mi vida aún cuando se olvida uno de las responsabilidades como ser humano ante EL y otras personas.

A mi director de tesis M.C David Covarrubias Rosales por su inmesurada atención a través de la elaboración de esta tesis, por permitir la realización de la misma, los consejos, la formación, la ayuda y la siempre buena disponibilidad que me mostró.

A los estudiantes de maestría de telecomunicaciones por sus valiosas aportaciones, consejos y amistad que me brindaron durante el desarrollo de esta tesis en especial a Miguel, Carlos e Iván.

A mi amigo M.C. Jorge Edgar Peña García por su amistad, disponibilidad y aportaciones de otro punto de vista en mi tesis lo cual daba una amplia gama de soluciones a escoger. Espero y se lleven a cabo todos tus ideales, estoy contigo, adelante.

A mi novia y compañera Elsa, por su paciencia y apoyo que me brindó en tiempos buenos y malos durante la tesis, además de su aportación en una de las figuras de la tesis.

A mi amigo M.C. Guillermo Díaz de Cosío, por sus consejos en los tiempos cuando todavía era estudiante, por su interés incondicional sobre mi futuro y planes así como mi bienestar y por tener el interés de guiarme en un camino que me lleve a la superación.

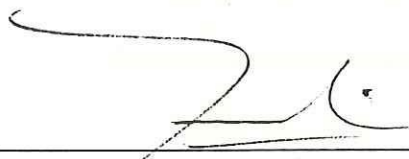
A la facultad de Ciencias y todas aquellas personas que me brindaron su apoyo durante mi carrera, en especial a mis amigos de carrera: Ray, Junior, Mendozas y Néstor por su indirecta influencia sobre mí en cuanto a superación tanto académica como personal y por su amistad a lo largo de la carrera.

Al Centro de Investigación Científica y de Educación Superior de Ensenada.

RESUMEN de la tesis de HECTOR JAVIER JIMENEZ PENSADO presentada como requisito parcial para la obtención de la Licenciatura en Ciencias Computacionales. Ensenada, Baja California, México. Abril 2000.

**SIMULACION DE REDES DE COMUNICACIONES DE COBERTURA
AMPLIA BASADO EN ALGORITMOS DE ENRUTAMIENTO DINAMICO**

Resumen Aprobado por: _____


M.C. David Covarrubias Rosales
Director de Tesis

La evolución de las redes locales de datos a redes de cobertura amplia ha sido posible en gran medida debido al empleo de la tecnología Interred en la cual es posible interconectar múltiples redes locales con diferentes características entre sí. Pero, existe una problemática asociada a la migración de redes locales a redes de cobertura amplia que es la interconectividad entre las mismas, específicamente en cuanto a enrutamiento.

Este trabajo de tesis analiza dicha problemática de interconectividad entre redes locales de datos y redes de cobertura amplia, enfatizando en el ingrediente esencial que viene siendo el enrutamiento. Para ello, se investigan los protocolos de las arquitecturas de red DoD y Novell, particularmente, el protocolo de enrutamiento RIP (Routing Information Protocol), el cuál es implementado a través de programática en una computadora personal conectada a la Interred TCP/IP, obteniendo con ésto último un registro con todas las rutas o trayectorias que son "observadas" por la computadora personal. Este registro se conoce como Tabla de Enrutamiento. Se propone además una solución al problema de interconectividad y se desarrolla la programática para enrutar paquetes desde una red Novell hacia una red TCP/IP y viceversa implementando un Sistema que se le denomina "Sistema de Canalización de Paquetes", dejando como problema abierto el desarrollo de programación de la capa de aplicación u otras aplicaciones, con el propósito de que surjan nuevas metodologías y tecnologías para el desarrollo de soluciones a los problemas de interconectividad existentes.

INDICE DE TRABAJO

CONTENIDO.	Pág.
I. INTRODUCCION	1
I.1.-Antecedentes	1
I.2.-Planteamiento del Problema	3
I.3.-Objetivo	4
I.4.-Metas	5
I.5.-Infraestructura Utilizada	6
I.6.-Organización del Trabajo	7
 II. ESTADO ACTUAL Y TENDENCIAS DE REDES LOCALES	
DE DATOS	8
II.1.-Historia	9
II.2-Estado Actual	12
 III. ARQUITECTURA DE RED DoD	15
III.1.-Introducción	15
III.2.-Antecedentes	15
III.3.-Modelo de Arquitectura de Red	17
III.4.-Protocolos de la Arquitectura de Red DoD	20
III.4.1.- Protocolos de capa de red	20
III.4.1.1.-Protocolo IP	20
III.4.1.2-Direccionamiento, Subredes y Broadcast en IP	23
III.4.1.3.-Protocolo ICMP	26
III.4.2.-Protocolos de capa de transmisión	28

CONTENIDO (continuación).	Pág.
III.4.2.1.-Protocolo UDP	28
III.4.2.2.-Protocolo TCP	31
III.5.-Enrutamiento IP	34
III.5.1-Protocolo IP	38
 IV. ARQUITECTURA DE RED NOVELL	 44
IV.1.-Introducción	44
IV.2.-Protocolos XNS	44
IV.3.-Ambiente de Red NetWare de Novell	44
IV.4.-Limitantes de una Red Novell para su Interconectividad RCA	47
IV.5-Protocolos de Capa de Red	52
IV.5.1.-Protocolo IPX	52
IV.5.2-Direcciones y Sockets	54
IV.6.-Enrutamiento IPX	57
 V. ALGORITMOS DE ENRUTAMIENTO DINAMICO	 62
V.1.-Introducción	62
V.2.-Algoritmos de Enrutamiento para Enrutadores	62
V.2.1.-Algoritmos de Vector-Distancia	63
V.2.2.-Algoritmos de Estado de Enlace	66
V.3.-Estado Actual e Implementaciones de Algoritmos de Enrutamiento Dinámico	67

CONTENIDO (continuación).	Pág.
V.4.-Problemas Presentados en el enrutamiento de una Interred	70
V.4.1.-Cambios de topología de red	70
V.4.2.-Inestabilidades. Conteo a infinito	71
V.4.3.-Soluciones a problemas de enrutamiento	72
V.5.-Selección de Protocolo de Enrutamiento	73
 VI. IMPLEMENTACION DE ALGORITMOS DE ENRUTAMIENTO	75
VI.1.-Introducción	75
VI.2.-Antecedentes	75
VI.2.1.-Tunel IP-IPX	75
VI.2.2.-Formato de paquetes	77
VI.2.3.-Desventajas de un esquema de interconectividad como el tunel IP-IPX	77
VI.2.4.-Diferencias entre Novell y TCP/IP	78
VI.3.-Solución Propuesta	79
VI.3.1.-Implementación de la propuesta en un ambiente real	82
VI.4.-Desarrollo de la Aplicación para un Ambiente Real	84
VI.4.1.-Sistema de Canalización de Paquetes (SCP)	85
VI.4.2.-Procesos de entrada y salida generales	87
VI.4.3.-Manejo de Temporizadores	90
VI.5.-Análisis de los Resultados Obtenidos a través de la Implementación en Ambiente Real	92

CONTENIDO (continuación).	Pág.
VI.6.-Validación del Protocolo RIP con el Paquete de Simulación COMNET III	100
VII. CONCLUSIONES, APORTACIONES Y RECOMENDACIONES	106
VII.1.-Conclusiones	106
VII.2.-Aportaciones del Trabajo	109
VII.3.-Recomendaciones	110
LITERATURA CITADA	112
APENDICE A	
Descripción de la Especificación PDS para el manejo de tarjeta de red.	
APENDICE B	
Organización, Descripción y Especificaciones de módulos en disco anexo.	

LISTA DE FIGURAS	Pág.
Fig. 1.-Escenario de Enrutamiento	4
Fig.2.-Arquitecturas de red	9
Fig.3.-Proyecto 802 de la IEEE	10
Fig.4.-Estructura en capas de las arquitecturas de red OSI y TCP/IP	17
Fig.5.-Escenario de una Interred TCP/IP	18
Fig.6.-Encabezado IP	21
Fig.7.-Clases y direcciones IP. Formato decimal de punto.	24
Fig.8.-Enmascaramiento de dirección IP	25
Fig.9.-Encabezado ICMP y datagrama asociado	27
Fig.10.-Encabezado UDP	29
Fig.11.-Proceso de travesía de datos hacia aplicaciones vía número de puerto	30
Fig.12.-Diagrama de bloque de la comunicación de TCP. Reconocimientos y secuenciación (Handshake).	32
Fig.13.-Proceso de Ventana Deslizante utilizado para la transmisión de datos en TCP	33
Fig.14.-Procesamiento de enrutamiento del protocolo IP	35
Fig.15.-Interacciones de enrutadores y tabla de enrutamiento	37
Fig.16.-Fases de inicialización, actualización y término del proceso RIP	40
Fig.17.-Encabezado RIP	41
Fig.18.-Procesamiento de enrutamiento del protocolo RIP	42
Fig.19.-Arreglo en capas de la arquitectura XNS y sus protocolos	45
Fig.20.-El modelo OSI y la arquitectura de red Novell	46
Fig.21.-Escenario de una red Novell	47
Fig.22.-Encabezado IPX	53

LISTA DE FIGURAS (continuación).	Pág.
Fig.23.-Proceso de enrutamiento IPX	58
Fig.24.-Escenario de Interred ejemplo para inestabilidades de enrutamiento	71
Fig.25.-Tunel IP-IPX	75
Fig.26.-Paquete IPX transportado como datos en un paquete UDP	77
Fig.27.-Topología empleada para la implementación de algoritmos de enrutamiento	80
Fig.28.-Mapeo desarrollado por el enrutador experimental pc-pruebas	81
Fig.29.-Enrutamiento de RIP (Novell) hacia RCA (TCP/IP)	82
Fig.30.-Organización interna del nodo pc-pruebas	83
Fig.31.-Funcionamiento del sistema de canalización de paquetes de la aplicación	86
Fig.32.-Proceso general de entrada de un paquete al SCP	88
Fig.33.-Proceso general de salida de la aplicación	89
Fig.34.-Procesamiento de temporizadores realizados por la rutina principal	91
Fig.35.-Metodología empleada para la simulación en COMNET III	103
Fig.36.-Topología de la RCA simulada en COMNET III	105

LISTA DE TABLAS.	Pág.
Tabla I.-Información de rutas para cada red en ejemplo de fig. 24	72
Tabla II.-Diferencias significativas entre Novell y TCP/IP	79
Tabla III.-Tabla inicial de pc-pruebas	93
Tabla IV.-Proceso de alta de una nueva ruta	93
Tabla V.-Tabla de enrutamiento completa de pc-pruebas	95
Tabla VI.-Tabla de enrutamiento de físicas antes de recibir las rutas de pc-pruebas	97
Tabla VII.-Tabla de enrutamiento de físicas después de recibir las rutas de pc-pruebas	98

I.- INTRODUCCION.

I.1.- Antecedentes.

El uso de las redes de comunicaciones va creciendo actualmente debido a la necesidad de intercambio de información, recursos y demás factores, resultando en una evolución natural de redes locales de datos (RLD) hacia redes de cobertura amplia (RCA). Las RLD originalmente surgieron para hacer más eficiente el trabajo dentro de organizaciones y de poder compartir información para su procesamiento, pero debido al advenimiento de nuevas tecnologías tanto de *hardware* como de *software* y al crecimiento exponencial de organizaciones e información, surge el concepto de RCA, las cuales permiten una interconectividad más amplia entre las organizaciones para sus requerimientos de comunicación y procesamiento informático.

Sin embargo, esta interconectividad engloba una problemática debido a factores como disimilitud de hardware (plataformas distintas) y software (sistemas operativos y protocolos de comunicación distintos) entre RLD y RCA, lo que provoca que un gran número de RLD se vayan aislando debido a su incompatibilidad hacia las RCA por los factores mencionados. Existe la tendencia a realizar estándares para que esos problemas sean resueltos. A pesar de ésto, la convergencia de una solución hacia la interconectividad ha sido lenta.

Actualmente, son muy pocos los trabajos que abordan de manera profunda la problemática asociada a la interconectividad, especialmente como temas de tesis. Lo anterior se refleja como muy pocas publicaciones y sin amplia investigación, a pesar de que el tema aborda un problema real.

La principal razón es la poca interacción de la problemática con el usuario ya que ésta es transparente a él, particularmente los aspectos de generación y actualización de las tablas de enrutamiento, las cuales permiten al paquete de datos, una vez fragmentado en pequeños paquetes, seguir trayectorias óptimas hasta alcanzar el nodo destino.

El desarrollo de una solución a la interconectividad, actualmente ha llegado a un punto de gran importancia dentro de varias ramas de la ciencia así como en los sectores administrativos, ya que hoy en día es un factor de impulso importante en la producción, aprendizaje y desempeño tanto de individuos como de organizaciones grandes a través del mundo.

Específicamente, dentro del área de las ciencias computacionales y de las telecomunicaciones, el problema de interconectividad va uniendo estas disciplinas más día con día, llamando la atención de las necesidades de otras. En el tema de computación distribuida, la interconectividad entre sistemas de alto cómputo a lo largo de un centro de investigación ó de varios centros de investigación dispersos en un país, viene siendo un punto crítico en su desarrollo, ya que gran parte del proceso de comunicación consiste en programática (protocolos implementados) que viene siendo la inteligencia de la comunicación, no solamente de la mecánica utilizada.

Los desarrollos actuales más fuertes se ven reflejados en las investigaciones originales realizadas por agencias militares del gobierno de los E.U.A. en la arquitectura de red conocida como DoD (Department of Defense).

Esta ha sido aprobada internacionalmente y ha causado innovaciones para el surgimiento de otras arquitecturas como el modelo de referencia estándar OSI, además de abrir el camino para otras compañías privadas y comerciales como Xerox, DEC, IBM y Novell en el desarrollo de nuevas metodologías para la interconectividad de redes de comunicaciones. Estas características hacen muy próspero el futuro desarrollo de interconectividad y de gran aplicación en la vida diaria de las personas, ya que existen tendencias de interconectividad total, en cualquier parte del mundo (compañía AT&T).

Por otro lado, la problemática anterior ha sido abordada por el grupo de investigación de redes de comunicaciones del CICESE, actualmente realizando trabajos relacionados a interconectividad. Tales temas abordan técnicas de codificación y compresión para voz en multimedios así como los protocolos asociados para su transporte, redes inalámbricas y protocolos de acceso a satélites. Este tema de tesis sigue parte de esa línea de investigación y desarrollo y se espera que los resultados de la misma aporten nuevas metodologías y técnicas de interconectividad.

I.2.- Planteamiento del problema.

El factor importante para la interconectividad es el enrutamiento de paquetes de datos a través de redes múltiples por dispositivos llamados enrutadores. Estos dispositivos manejan cierta información que es vital para el propio enrutamiento de paquetes: Tablas y algoritmos de enrutamiento. Por medio de los algoritmos, las tablas manejan rutas para que los paquetes de datos recorran un camino óptimo hacia un destino.

Además, las tablas deben ajustarse a cambios topológicos en una RCA actualizando de una manera dinámica sus rutas mediante la propagación de información de enrutamiento entre los enrutadores existentes en una RCA. Estos elementos de interconectividad se muestran en la figura 1.

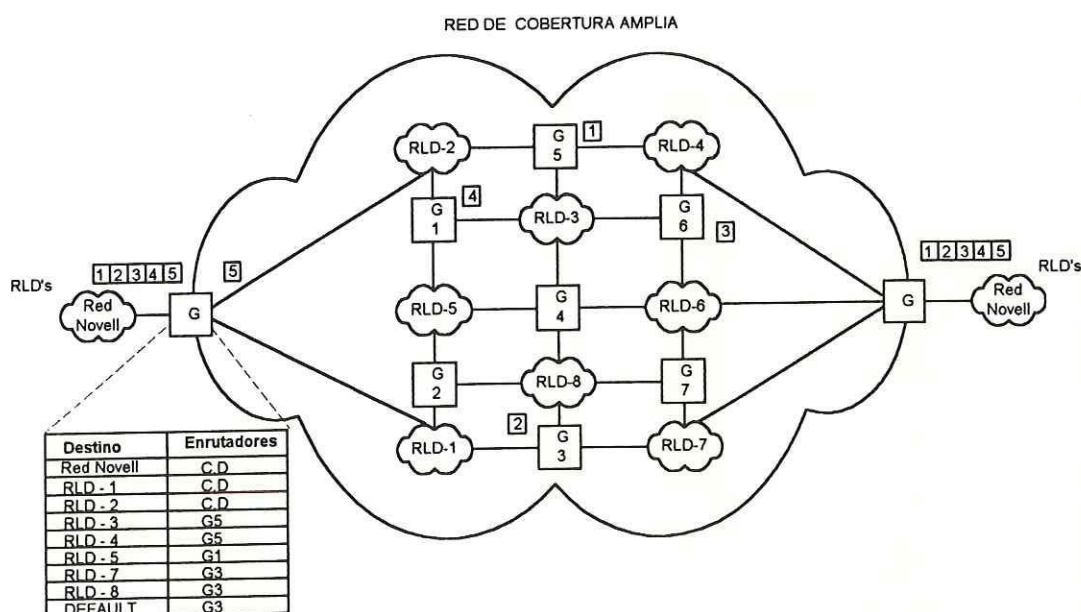


Figura 1.- Escenario de enrutamiento.

A partir de la figura anterior se puede ubicar de mejor manera la parte de análisis de este trabajo, planteando a partir de ello el siguiente objetivo.

I.3.- Objetivo.

El objetivo de esta tesis se concentra en la generación, operación y actualización de una manera dinámica de las tablas de enrutamiento por medio del desarrollo de los algoritmos de enrutamiento a través de la programación, siendo validada ésta en un ambiente real de comunicaciones, ya que esto provee los medios de interconectividad y solución a los problemas de compatibilidad de RLD a RCA.

I.4.- Metas.

- * Investigar el estado del arte de las redes locales de datos, en particular Ethernet.
- * Investigar la problemática asociada a la interconectividad de RLD's.
- * Investigar las arquitecturas de red OSI y DoD así como las arquitecturas de red de la compañías Xerox (XNS) y Novell, en particular los protocolos de enrutamiento y transporte de cada una de ellas.
- * Definir la problemática asociada a la compatibilidad entre Novell y TCP/IP para la identificación de disimilitudes y requerimientos de interconectividad.
- * Investigar los algoritmos de enrutamiento asociados a IPX y TCP/IP.
- * Definir un modelo de enrutamiento, basado en el algoritmo de enrutamiento contenido en el protocolo RIP de la arquitectura de red XNS, que permita un esquema de interconectividad y que cubra los aspectos de enrutamiento y coexistencia entre las arquitecturas de red DoD y Novell.
- * Desarrollo de software del modelo de enrutamiento.
- * Implementación del modelo de enrutamiento para su validación a través de una RLD Ethernet con salida a red de cobertura amplia (Internet TCP/IP).
- * Simulación del ambiente de interconectividad a través del paquete de simulación COMNET III.

I.5.- Infraestructura utilizada.

- * Red Local de Datos Ethernet
- * Red de Cobertura Amplia (Internet TCP/IP).
- * Compilador Turbo C++ ver. 3.0, Turbo C ver. 2.0 y Turbo Ensamblador.
- * Laboratorio de Redes Locales de Datos y Laboratorio de Simulación de Redes en el depto. de Electrónica y Telecomunicaciones del CICESE (Centro de Investigación Científica y de Educación Superior de Ensenada).
- * Paquete de simulación COMNET III.
- * Facilidades y servicios de Internet:
 - RFC's.
 - Gopher
 - FTP
 - Grupos de Interés
 - Telnet

I.6.- Organización del Trabajo.

Con los conocimientos en ciencias computacionales y aquellos adquiridos durante la realización de esta tesis, el trabajo se organiza de tal manera que se pueda profundizar en los aspectos de interconectividad de un ambiente de RLD (tomado en base de la arquitectura de red Novell) y un ambiente de RCA (tomado en base de la arquitectura de red TCP/IP), así como en su problemática de compatibilidad, para proveer un modelo de enrutamiento asociado a la programación en un ambiente real.

En los primeros capítulos de la tesis se crea una base para la mejor comprensión de las arquitecturas de red en cuestión. El capítulo II en particular, trata sobre las tendencias y el estado actual de las redes de comunicaciones.

El capítulo III realiza un estudio sobre la suite de protocolos TCP/IP, para observar el funcionamiento, los elementos y el enrutamiento de una RCA.

Al igual, el capítulo IV analiza tanto los protocolos de la compañía Novell como los de XNS ya que provee las características de una RLD y poder identificar y establecer la diferencias entre las dos arquitecturas de red para poder proponer una forma de interconectividad y coexistencia entre ellas. En ambos capítulos III y IV se enfatiza lo más posible en cuanto a sus aspectos de enrutamiento que son los procesos mismos de enrutamiento que se llevan a cabo en ambas arquitecturas y el manejo de tablas de enrutamiento. Se hace más énfasis en el capítulo III por tener un enrutamiento genérico bastante sólido y orientado a ambas RLD y RCA.

En el capítulo V se da una introducción a los algoritmos de enrutamiento dinámico existentes, sus tendencias y diferencias para poder establecer el más óptimo, aceptado y flexible de ellos. En este capítulo se refleja la validez de la utilización del algoritmo de enrutamiento del protocolo RIP, escogido para el estudio de esta tesis. Posteriormente, en el capítulo VI, se establece el desarrollo de un modelo de enrutamiento que permite interconectividad RLD - RCA así como los resultados obtenidos directamente de su implementación en un escenario real. El capítulo VII finaliza el trabajo con una serie de conclusiones, recomendaciones y aportaciones enfocadas a determinar el estado de arte y tendencias hacia nuevas metodologías y tecnologías en cuanto a la interconectividad de redes de comunicaciones a través del desarrollo de software.

II.- ESTADO ACTUAL Y TENDENCIAS DE LAS REDES LOCALES DE DATOS

II.1.- Antecedentes.

A fines de 1960 y a principio de 1970, las redes de computadoras se definían a partir de la computadora utilizada. Después de que una compañía cuidadosamente seleccionara dicha máquina de algún vendedor propietario como IBM, Honeywell, DEC u otro, los clientes o usuarios del sistema estaban sujetos a esa única arquitectura elegida hasta que se volvía obsoleta o era necesario un rediseño completo. Por otra parte, si el sistema requería de modificaciones o alteraciones, la compañía llamaba al vendedor que realizaría la actualización; la compañía casi nunca consideraba equipo de terceros fabricantes debido a la naturaleza de la arquitectura que se utilizaba desde un principio.

A mediados de 1970, se introdujo el concepto de sistemas abiertos los cuales se adherían a estándares públicos con interfaces definidas, abriendo la posibilidad de que una minicomputadora DEC accediera un mainframe IBM (Madron, 1990). El movimiento fue encabezado por usuarios que querían soluciones independientes de los vendedores propietarios, y de los vendedores mismos que querían desarrollar productos para el nuevo mercado abierto. La colaboración entre estos grupos formó estándares de desarrollo como el CCITT, ISO y ANSI. La creciente necesidad de comunicación de red a red, que hoy se llama *internetworking* o interconectividad, inspiró a la investigación que llevó a lo que se conoce como el modelo de referencia **OSI** en 1978 desarrollado por la **ISO** (International Standard Organization) (Santifaller, 1991).

Este modelo fue diseñado como estándar para permitir la comunicación de varios sistemas *abiertos*. Un sistema que cumple con los estándares (o protocolos) de comunicación con otros sistemas se define como abierto. El sistema abierto está basado en estándares en vez de estar orientado a propietarios. Por lo tanto, un sistema abierto se puede comunicar y cooperar con otro utilizando interfaces y protocolos que ambos sistemas entienden.

Debido al surgimiento del modelo OSI, se ha realizado un continuo desarrollo asociado a protocolos estándares. Estos protocolos describen la utopía de interconectividad, pero como aún predominan arquitecturas de red propietarias como SNA, DECnet, etc., la evolución hacia una arquitectura de red equivalente entre sistemas (como la de OSI) ha sido lenta.

OSI	XNS/Novell	DoD	DEC	SNA
Aplicación	Aplicación	Aplicación	Usuario	Usuario Final
Presentación			Manejo de red	Transacción
Sesión	Control	Transporte	Aplicación de red	Presentación
Transporte	Transporte Interproceso		Control de sesión	Flujo de datos
Red	Transporte Interred	Interred	Transporte end-end	Transmisión
Enlace			Enrutamiento	Control de camino
Física	Transmisión al medio	Red Local	Enlace	Control de enlace
			Física	Física

Figura 2.-Arquitecturas de red.

Como se puede ver en la figura anterior, existen varias disimilitudes entre las capas y funciones de las diferentes arquitecturas de red existentes, lo cual hace que la interconectividad entre dichos sistemas sea difícil, así como su evolución a una arquitectura estándar.

Para poder tener expectativas positivas en cuanto a que en las capas de las diferentes arquitecturas de red no se crearán conflictos, como en el caso del hardware de los diferentes sistemas, la **IEEE** (Institute of Electrical and Electronics Engineers) realizó un proyecto denominado 802 al reconocer las necesidades de estandarización y normalización en el mercado de las redes locales de datos (Madron, 1990).

Este proyecto hace referencia a estándares en las capas física y enlace del modelo

OSI:

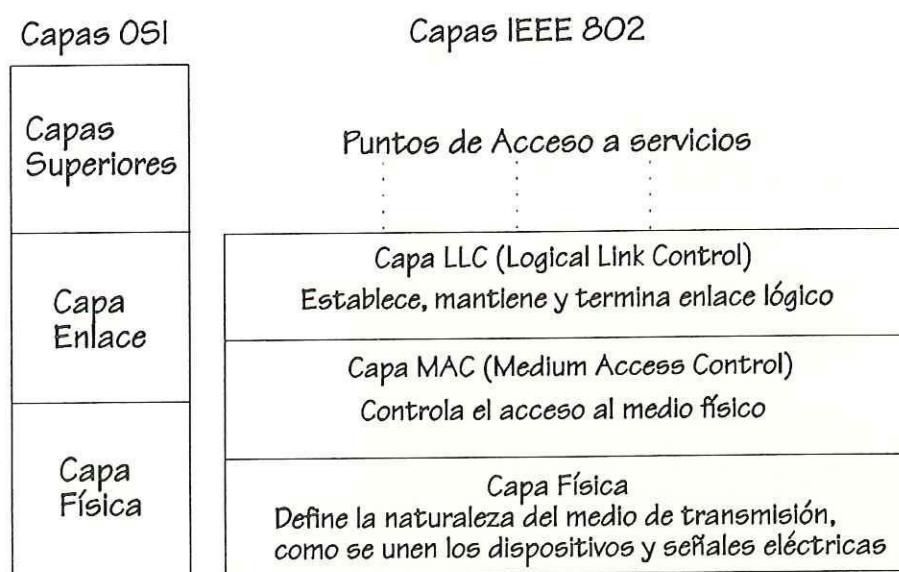


Figura 3.-Proyecto 802 de la IEEE

en donde la capa física del **OSI** es equivalente a la de la **IEEE**, la capa MAC (Medium Access Control) del **IEEE** entre la física y enlace **OSI**, y la LLC (Logical Link Control) en la de enlace.

Lo importante a notar aquí (diferencia entre capa de enlace y LLC) es que pueden existir múltiples puntos de acceso al enlace de datos en el modelo **IEEE**, es decir, una estación en una Red Local de Datos tiene una sola dirección MAC y se pueden comunicar con varios procesos diferentes (aplicaciones) de capas superiores.

El beneficio de esta característica es que se pueden tener diferentes procesos o protocolos en capas más altas en diferentes estaciones en la RLD con una sola dirección de hardware en cada estación en vez de tener varias para cada protocolo o proceso; la única condición es que ambos tengan este modelo en dichas capas.

La **IEEE** estableció varios protocolos para soportar la especificación 802; entre otras:

- * 802.1 - Interfaz Alto Nivel
- * 802.2 - Especificación de la capa LLC.
- * 802.3 - Especificación de protocolo CSMA/CD para Bus.
- * 802.4 - Especificación para paso de testigo en Bus (Token Bus).
- * 802.5 - Especificación para paso de terstigo en anillo (Token Ring).
- * 802.6 - Redes Metropolitanas
- * 802.7 - Grupo Consejero de Técnicas de Banda Amplia
- * 802.8 - Grupo Consejero de Técnicas de Fibra Optica
- * 802.9 - Redes de Datos y Voz Integrados
- * 802.10 - Seguridad en la interoperabilidad de RLD's
- * 802.11 - Redes Inhalámbricas

Con las normas establecidas por el modelo **IEEE**, se dió uno de los pasos importante a la individualización de las capas de arquitectura de red, como la de **OSI**, permitiendo tener diferente tipo de cableado como medio y diferente software o protocolos en las capas superiores, que fueran específicos para cada máquina o red, es decir, se aisló el área de las capas física y enlace del modelo de referencia **OSI**.

Para principios de 1980 las redes de computadoras ya se podían organizar en varias maneras; ya era posible diferenciar entre lo que se conoce como una Red Local de Datos (RLD) y una Red de Cobertura Amplia (RCA).

Muchas organizaciones, instituciones y campus universitarios requerían de compartir y procesar información entre usuarios dentro de las mismas organizaciones locales, de ahí el término Red *Local* de Datos ya que ésta cubre un área geográfica limitada (menos que 5 km.). Las RLD son administradas por una sola organización, privada, pero también existía la necesidad de compartir y procesar información entre las diferentes organizaciones, siendo éstas principalmente universidades. Por lo tanto, el término RCA surgió de la necesidad de *interconectividad*, casi al mismo tiempo que las RLD ya que la popularidad y crecimiento de éstas fue muy rápido.

II.2.-Estado Actual.

Actualmente, existen cientos de RLD interconectadas vía RCA, dispersas por todo el globo y cubriendo una gran diversidad no solo de redes (TCP/IP, X.25, SNA, etc.) sino además de tipos de computadoras (HP, IBM, Apple, Sun, etc.). Claramente se ve la necesidad de estandarización ó normalización entre RLD ó RCA.

Por otro lado, los *protocolos* son los medios por los cuales las aplicaciones de las computadoras en una RLD se pueden comunicar con otra, incluso de diferente arquitectura. Los protocolos son reglas que coordinan el intercambio de datos o mensajes entre parejas (Spragins, 1991).

Debido a la complejidad de comunicación entre sistemas de comunicaciones (RLD, RCA), no es sensato ejecutar todas las tareas de interconectividad en un solo protocolo, de aquí la división en capas de las arquitecturas de red.

Se aplican varios protocolos de comunicación simultáneos en la forma de capas *sobrepuestas* (Layering) de protocolos que llevan a cabo diferentes funciones para lograr comunicación, cooperación entre sí, para que juntos provean un servicio de usuario a usuario de manera transparente. Estas funciones van desde aplicaciones de usuario (interface, correo, transferencia de archivos, etc.) , enrutamiento (cómo enviar lo datos a través de RLD o de una RCA), hasta el medio físico (cableado, señales, etc.) (Feit, 1994).

En base a lo anterior, se puede establecer que el presente trabajo se concentra solamente en las funciones de enrutamiento; es decir, en los protocolos de enrutamiento más utilizados entre RLD y RCA. Se hará especial énfasis principalmente en las arquitecturas de red correspondientes a los protocolos TCP/IP y los de la compañía Novell, ya que éstas proveen implementaciones de RCA y RLD respectivamente, además de tener una gran popularidad y uso actualmente en diversos centros de investigación, universidades, organizaciones gubernamentales, empresas, etc. a nivel mundial.

Los protocolos por parte de la compañía Novell están orientados a un ambiente de RLD, aunque últimamente se ha ido incorporando a las RCA debido a las necesidades de interconectividad.

Debido a sus características de RLD, muchas organizaciones pequeñas y de tamaño moderado han implementado dicha arquitectura de red siendo entre las RLD una de las más populares. Como TCP/IP es lo más popular y cubre gran parte de las implementaciones a nivel global no solo en cuanto a RLD de diferentes tamaños, sino a RCA, el conjunto de protocolo de TCP/IP es de gran interés para este trabajo.

Por lo anterior, se realizará un análisis del funcionamiento de ambas arquitecturas por separado para la familiarización de conceptos requeridos, para posteriormente pasar a la parte medular de este trabajo de tesis, en donde se concentrará el interés en los protocolos de enrutamiento dinámico, tablas de enrutamiento, direccionamiento y el funcionamiento de los dispositivos de interconexión de RCA que son los enrutadores o gateways.

Se enfatizará en los aspectos mencionados de ambas arquitecturas para establecer un modelo que permita resolver la interconectividad entre una RLD como Novell en un ambiente de RCA como TCP/IP y viceversa. Todo esto a nivel de la capa de Interred o de red correspondiente al modelo de referencia OSI.

En los siguientes capítulos se realizará el estudio y análisis de las características de las arquitecturas de red de Novell tanto como de DoD, enfatizando en los aspectos de enrutamiento de ambas.

III.- ARQUITECTURA DE RED DoD

III.1.- Introducción.

Este capítulo examina los protocolos involucrados en lo que se conoce como la arquitectura DoD que comprende la suite de protocolos TCP/IP. Este tipo de protocolos es uno de los más utilizados actualmente. Existen muchas características importantes relacionadas a TCP/IP para este trabajo de tesis que incluyen enrutamiento dinámico, heterogeneidad y características de Red de Cobertura Amplia (RCA) genéricos. Los conceptos más importantes se incluyen en este capítulo.

III.2.- Antecedentes.

La arquitectura de red DoD, llamada *Internet* tuvo principios desde fines de 1960; existía gran demanda en diferentes universidades de Estados Unidos y centros de investigación para que una RCA permitiera la utilización a nivel nacional de los recursos de cada uno de ellos, ya que no existía arquitectura que permitiera esto por ser sistemas cerrados o propietarios.

Al mismo tiempo, una agencia del gobierno de los E.U.A llamada ARPA (Advanced Research Projects Agency) formó la ARPANET, una red con propósitos militares principalmente. Esta red consistía de 4 nodos principales ubicados en UCLA (Universidad de California en Los Angeles), UCSB (Universidad de California en San Bernadino), SRI (Instituto de Investigación de Stanford) y la Universidad del estado de Utah (Santifaller, 1991).

La compañía Bolt, Beranek and Neuman BBN obtuvo el contrato de la ARPANET y fué la que tuvo una fuerte influencia en el posterior desarrollo de la arquitectura DoD y los protocolos TCP/IP. Los diseños para los protocolos individuales y rutas específicas de lo que originalmente la ARPANET realizaba fueron desarrollados. Por estas raíces la suite de protocolos TCP/IP corresponde a lo que se llama la arquitectura DoD (Department of Defense).

La principal meta de TCP/IP es permitir la existencia de una comunicación abierta además de permitir la proliferación de interconectividad Red Local de Datos - Red Local de Datos (RLD) y RLD - RCA entre ambientes múltiples.

TCP/IP es uno de los protocolos de red más utilizados actualmente (Spragins, 1991). Antes de TCP/IP, los protocolos eran propietarios y conocidos solamente por pocos individuos u organizaciones. Usuarios y administradores de red estaban ligados y detenidos en cuanto a desarrollo de redes. TCP/IP aportó área de trabajo para protocolos futuros (OSI) y permitió acceso abierto para que usuarios eligieran este sistema sin necesidad de adaptarse a algún vendedor particular.

El estudio de esta arquitectura es de suma importancia para este trabajo de tesis por las características que provee de RLD así como de RCA. El protocolo está bien definido y ampliamente aceptado como un estándar de facto, justificando su funcionalidad y potencialidad.

III.3.- Modelo de arquitectura de red.

De acuerdo a la fig. 4, el modelo de la arquitectura DoD es diferente a la del modelo OSI.

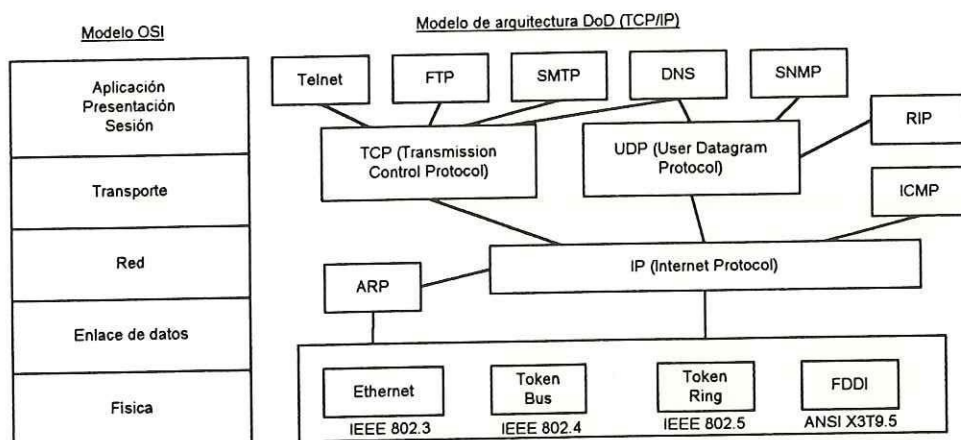


Figura No. 4.- Estructura en capas de las arquitecturas de red OSI y TCP/IP.

De la fig. 4 se puede observar que ambas arquitecturas difieren tanto en funciones por capas como en su organización, ya que la capa de interred IP DoD abarca parte de la capa de enlace y red de la de OSI, al igual que la capa de transmisión de DoD ocupa gran parte las capas superiores de la de OSI (Santifaller, 1991); es decir, no existe una correspondencia uno a uno entre las diferentes capas entre ambos modelos. esta idea es erróneamente manejada por muchas personas pensando que las funciones de TCP corresponden a las funciones de capa de transporte, y las de IP con las de red.

Por otro lado, la terminología utilizada en TCP/IP es diferente a la empleada en OSI. Esto se puede aclarar más de acuerdo a la figura 5.

RLD: Red Local de Datos
RCA: Red de Cobertura Amplia

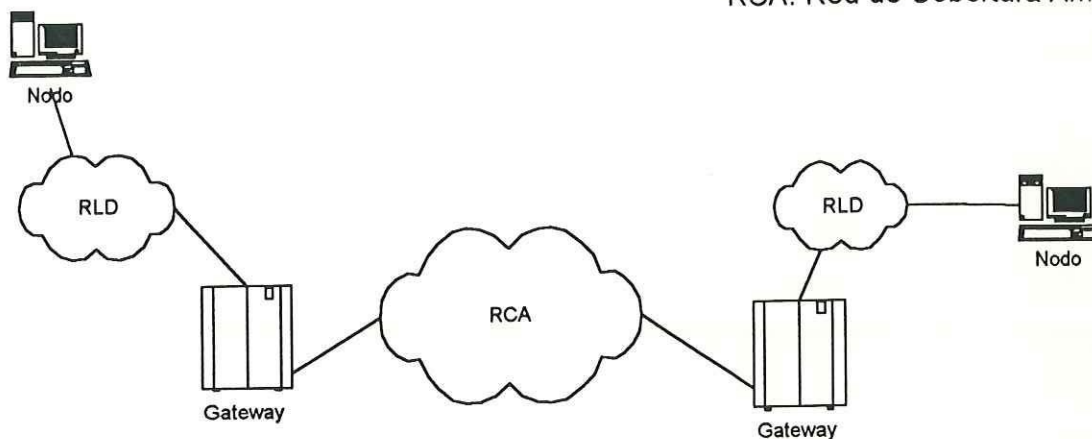


Figura 5.- Escenario de una Interred TCP/IP.

En la figura 5 se pueden apreciar los componentes más importantes en una Interred TCP/IP, los cuales son: Nodos ó computadoras que ejecutan tareas, aplicaciones, etc. que se conectan a una RLD ó a una RCA por medio de protocolos de acceso al medio físico (Ethernet); Gateways ó enrutadores IP que conectan varias RLD y a su vez a éstas con RCA (Comer, 1991).

La operación de enrutamiento de este escenario se debe principalmente a la característica de que cada nodo y enrutador contienen un módulo de implementación del protocolo IP. Estos módulos comparten reglas afines para interpretar información de los protocolos para su direccionamiento, fragmentación y reensamblaje (Feit, 1994). Estas reglas están dictadas por protocolos de enrutamiento dinámico en este caso.

Se requieren de protocolos adicionales a IP en los gateways para un enrutamiento más efectivo de la información que viaja a través de la Interred. Para ésto, los gateways mantienen una *tabla de enrutamiento* que especifica las rutas hacia otras redes. Para que las tablas se mantengan, actualicen y se verifiquen, los gateways las intercambian detallando la topología de toda la Interred. Estos intercambios y revisiones de tablas son realizados por *algoritmos de enrutamiento dinámico* (RIP, OSPF, etc.). De esta manera, cada paquete de datos es enrutado individualmente, basado en las tablas de enrutamiento que mantiene cada gateway, así como los algoritmos de enrutamiento que manejan dichas tablas.

El protocolo IP que reside en cada nodo en la Interred, contiene características que permiten un enrutamiento genérico muy adecuado para su adaptación en una RCA. Tales características incluyen direccionamiento, fragmentación y enrutamiento, además de la flexible adaptación de algoritmos de enrutamiento como lo son RIP y otros.

A continuación se describen los protocolos asociados a la arquitectura de red DoD, haciendo énfasis sólo en aquellos que son relevantes a esta tesis. Estos protocolos comprenden RIP, TCP, UDP, IP, ICMP de acuerdo a la relación en capas mostrada en la figura 4.

III.4.-Protocolos de la arquitectura de red DoD:

Los protocolos de la capa de red local de datos son totalmente independientes de las capas superiores, además de estar fuera del alcance de este trabajo, por lo que el interés se concentra más en las capas de red y de transmisión. El protocolo RIP se describe en otra sección ya que es de interés especial, en conjunto con características de enrutamiento de la arquitectura de red DoD.

III.4.1.- Protocolos de Capa de Red.

La capa de red consiste de los protocolos IP, ICMP y ARP, y para la capa de transmisión TCP y UDP. La capa de aplicación de esta arquitectura no es de interés en este trabajo de tesis, por tanto, no se profundiza tanto como la capa de red.

De la sección anterior se puede notar que el protocolo IP es el más importante para cuestiones de enrutamiento en una Interred, ya que provee mecanismos y estructuras genéricas y especiales para realizar adecuadamente las funciones de enrutamiento las cuales se describen a continuación.

III.4.1.1- Protocolo IP (Internet Protocol):

Este protocolo fué diseñado para interconectar redes de comunicación de conmutación de paquetes para formar una Interred. Transmite bloques de datos llamados *datagramas* que se reciben del software de capas superiores desde y hacia nodos. Provee un servicio de envío/recepción orientado a no-conexión (no establece una sesión con el destino antes de enviar los datos) desde un nodo origen hasta un destino. Este protocolo se encarga del direccionamiento en la Interred (Postel, 1981).

La meta principal de IP por tanto, es proveer un algoritmo básico de transferencia de datos desde y hacia una red. La unidad de transferencia (datagrama) de IP puede viajar sobre redes de alta velocidad (Ethernet, FDDI, Token Ring); cuando se transmite por éstas se llama *paquete* y no datagrama.

El protocolo IP agrega información, utilizada como control, a los datos que recibe de capas superiores (UDP ó TCP) llamado *encabezado IP* de longitud de 32 bits, siendo este la estructura utilizada para establecer un mecanismo de direccionamiento y enrutamiento. El encabezado IP se muestra en la fig. 6.

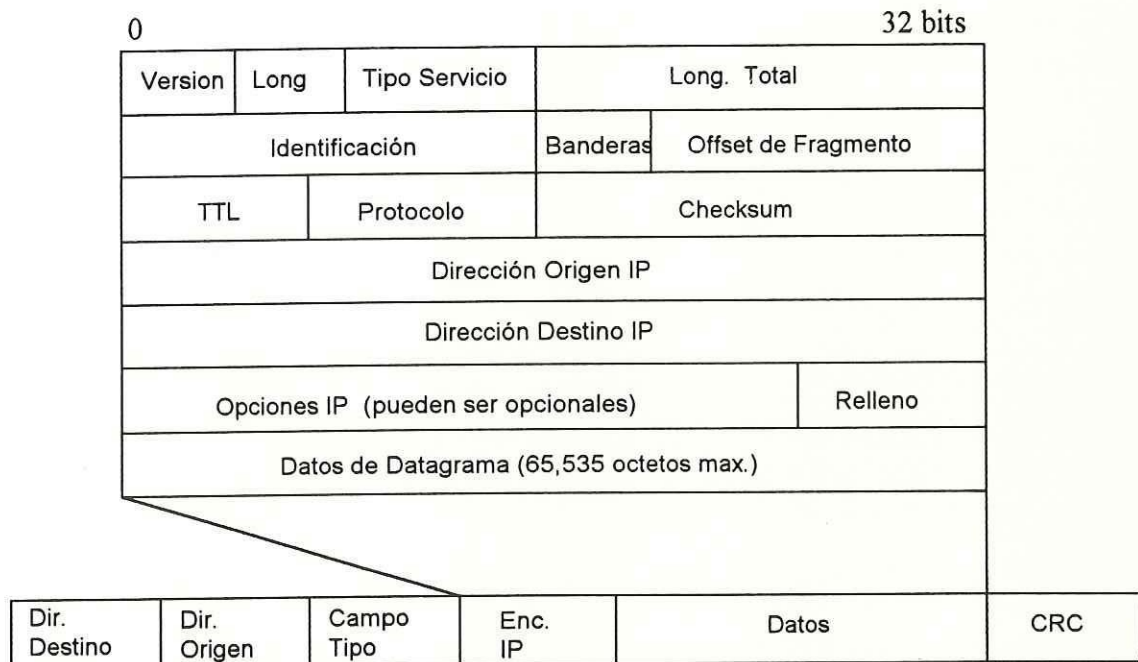


Figura 6.- Encabezado IP.

La información contenida en este encabezado permite la manipulación del datagrama para su enrutamiento utilizando los campos de *dirección IP* que el protocolo usa para la selección de un camino de transmisión. Las direcciones IP se explican con más detalle posteriormente. A continuación se muestra el significado de cada uno de los campos del encabezado IP.

Versión: Define la versión de implementación de IP del nodo.

Long: Longitud de Encabezado IP, sin incluir datos. Se mide en cantidades de palabras de 32 bits. La longitud mínima es de 20 octetos, o sea, este campo tendrá el valor de 5 (20 octetos = 160 bits/32bits = 5).

Tipo de servicio (TOS): Se subdivide en Precedencia que indica la prioridad de envío de datagrama; Retardo, Throughput y confiabilidad los cuales indican al enrutador IP cual ruta tomar. Este campo se le conoce como identificador TOS.

Identificación, banderas y offset: utilizados para la fragmentación de datagramas siendo el primer campo el indicador de cuales fragmentos (a su vez datagramas) pertenecen a un solo datagrama asignándoles un número de identificación; las banderas indican si existen más fragmentos ó si no se desea fragmentar el datagrama. Por último el offset indica la posición de este fragmento con respecto a los demás, utilizado en reensamblaje.

TTL: Indica el Tiempo de vida de un datagrama en la Interred, es decrementado por cada enrutador para ser descartado si éste llega a ser cero.

Protocolo: Indica cual protocolo de capa superior a IP envió el datagrama. Por lo regular siendo TCP ó UDP.

Checksum: Es el CRC de 16 bits. Utilizado para asegurar la integridad del encabezado. Calculado por el nodo transmisor y recalculado para su verificación por los demás nodos.

Opciones: Básicamente contiene información de enrutamiento fuente, rastreo de rutas, marcas de tiempo para paquetes mientras atraviesa la Interred, y seguridad. Estos campos son opcionales y pueden no aparecer en el encabezado.

Dirección origen y destino IP: Estos campos indican el destino final (nodo) al cual el paquete debe ser enviado y la dirección IP del nodo que originó la transmisión del paquete. Este esquema de direccionamiento requiere de más atención por lo que posteriormente se describe en la siguiente sección.

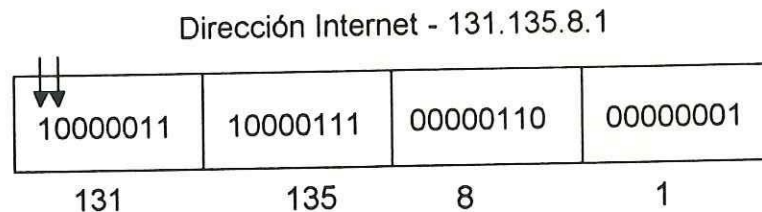
III.4.1.2- Direccionamiento, Subredes y Broadcast en IP.

Las ideas y conceptos de TCP/IP fueron establecidas antes que Ethernet y Token Ring se desarrollaran. Los controladores (tarjetas de red) de éstos últimos utilizan direcciones conocidas como direcciones MAC de 48 bits, las cuales identifican la tarjeta de la red del nodo como única. Por tanto, se implementó un esquema de direccionamiento para identificar a nodos IP y dónde se ubican en una Interred. Este esquema se llama *esquema de direccionamiento IP de 32 bits* (Naugle, 1994).

Las direcciones IP identifican a un número de nodo en particular así como al mismo tiempo un número de red donde ese nodo se encuentra. El formato generalizado de una dirección IP es el siguiente:

$$\langle \text{Número de Red, Número de nodo} \rangle = 128.4.70.9$$

Las direcciones IP se dividen en 5 clases: A, B, C, D y E. El formato, direccionamiento y la división en clases de una dirección IP se muestra en la figura 7.



En el sistema binario, el bit 0 y 1 son 1 y 0, indicando que la dirección es de clase B.
 En el sistema decimal, el primer octeto está en el intervalo 128-191, o sea, clase B.

Primer campo indica clase:

Clase A

1-126.nodo.nodo.nodo

Clase B

128-191.numred.nodo.nodo

Clase C

192-223.numred.numred.nodo

Figura 7.- Clases y Direcciones IP. Formato decimal de punto.

Por lo tanto, la dirección 158.97.0.0 denomina a una red de clase B; 158.97.10.0 denomina una *subred* de la red 158.97.0.0 con número 10; 158.97.10.20 es la dirección de un nodo (número 20) que está en la subred 10 de la red 158.97.

La dirección de clase A 127.0.0.0 contiene un significado especial, éste llamado *loopback* (Naugle, 1994) utilizada por procesos que deseen comunicarse con el protocolo TCP en el mismo nodo. Si cada octeto de la dirección IP contiene puros unos, a la dirección se le conoce como *broadcast limitado* y el paquete será recibido por todos los nodos en la red local. Los enrutadores no adelantan paquetes con este tipo de direcciones (255.255.255.255). Cualquier dirección con ceros en alguna parte denota "esta red".

El direccionamiento IP contiene a su vez un esquema que permite utilizar la parte del número de red de una dirección IP para reasignarlo como otro número de red; es decir, con una sola dirección IP se pueden representar varias subredes.

Este método se conoce como *enmascaramiento de subred* (Naugle, 1994) ya que se aplica una máscara a la dirección IP para saber si el nodo destino está en una red ó una subred. Este esquema es utilizado por los enrutadores y se muestra en la fig. 8.

Máscara de subred - 255.255.248.0

Clase B		30 subredes	Hasta 2046 nodos
10000001	00000001	00001001	00000001
Dirección de red		Sub-red	Dirección de nodo

Red 129.1

Subred 8

Nodo 257

- 1) 4 octetos de dirección escritos igual
- 2) Escribir dirección en binario
- 3) Escribir la máscara de subred en binario
- 4) Realizar la operación Y lógico
- 5) Los bits resultantes de la operación anterior muestran el número de red y subred.

Figura 8.- Enmascaramiento de Dirección IP.

Operación:

```

10000001 00000001 00001001 00000001 - Dirección 129.1.9.1
11111111 11111111 11111000 00000000 - Máscara de subred 255.255.248.0
-----
10000001 00000001 00001000 00000000 - Red 129.1
                                         - Subred 8
  
```

Dirección verdadera: 129.1.8.0.

Posteriormente se describirán más aspectos involucrados con el protocolo IP, en la sección de descripción del protocolo RIP y la sección de enrutamiento IP.

III.4.1.3.- Protocolo ICMP (Internet Control Message Protocol).

En conjunto con el protocolo IP, se debe tener una implementación del protocolo ICMP. Ocasionalmente, un gateway o nodo destino se deberá comunicar con el origen para reportes de error en el procesamiento de comunicación principalmente. Para estos propósitos se diseñó ICMP.

Este protocolo utiliza el soporte básico de IP como si fuera un protocolo de capa superior, aunque en realidad es una parte integral de IP. Los mensajes de ICMP son enviados en casos tales como: un datagrama no puede llegar a su destino, el gateway necesita redireccionar el datagrama hacia otra ruta, etc.

El protocolo IP no fue diseñado para ser absolutamente confiable (Postel, 1981), así, los mensajes que proporciona ICMP proveen una retroalimentación acerca de problemas en el ambiente de comunicación. El propósito de ICMP no es hacer el protocolo IP confiable.

El formato de mensajes de ICMP se envían con el encabezado IP y en el primer octeto del área de datos del datagrama se encuentra información relevante al protocolo ICMP como lo muestra la fig. 9.

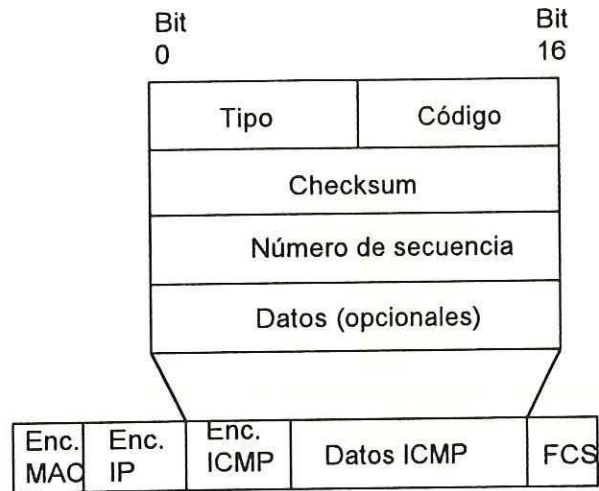


Figura 9.- Encabezado ICMP y datagrama asociado.

III.4.2.- Protocolos de capa de Transmisión.

Esta sección describe los protocolos de transporte UDP y TCP, que residen en la capa inmediata superior a IP. Debido a la orientación de este trabajo de tesis, el protocolo TCP sólo se definirá y se hará más profunda la descripción del protocolo UDP.

Se ha descrito anteriormente la capacidad de los protocolos de TCP/IP de transferir datagramas IP entre nodos, donde cada datagrama es enrutado a través de la Interred basándose en la dirección IP destino. En la capa del protocolo Interred, una dirección identifica a un nodo único, no se hace distinción adicional de qué usuario ó qué aplicación recibirá el datagrama. Los protocolos de transporte, añaden un mecanismo que permite distinguir entre múltiples destinos dentro de un nodo dado mediante asignaciones de números de puerto. De esta manera se pueden comunicar varias aplicaciones entre sí de un nodo a otro.

III.4.2.1.- Protocolo UDP (User Datagram Protocol)

El protocolo UDP provee acceso a la red a programas de aplicación utilizando un servicio de capa de transmisión orientada a no conexión (datagrama). Permite la transferencia de datos entre nodos origen y destino sin tener que establecer una sesión antes de transmitir. Este protocolo no usa la revisión de error de reconocimiento de nodos que utiliza TCP. Básicamente la funcionalidad de capa de transmisión existe en UDP pero el sobreprocesamiento es más bajo. Por estas razones, UDP es un protocolo ligero a comparación con el protocolo TCP.

El protocolo UDP especifica *números de puerto* (Postel, 1980) para identificar a las aplicaciones que lo utilizan, éstos y otros datos viajan en el encabezado.

Cada mensaje UDP contiene un número de puerto origen y destino haciendo posible que el software en el destino entregue el mensaje a la aplicación correcta. La transferencia es realizada empleando una comunicación orientada a no conexión como en el caso de IP para el protocolo UDP. No hay revisión de errores ni reconocimientos de envío y recepción. Lo único que UDP provee es la asignación y manejo de números de puertos que identifican aplicaciones individuales que están en los nodos en la Interred.

Así como los demás protocolos, UDP maneja un encabezado que se muestra en la figura 10.

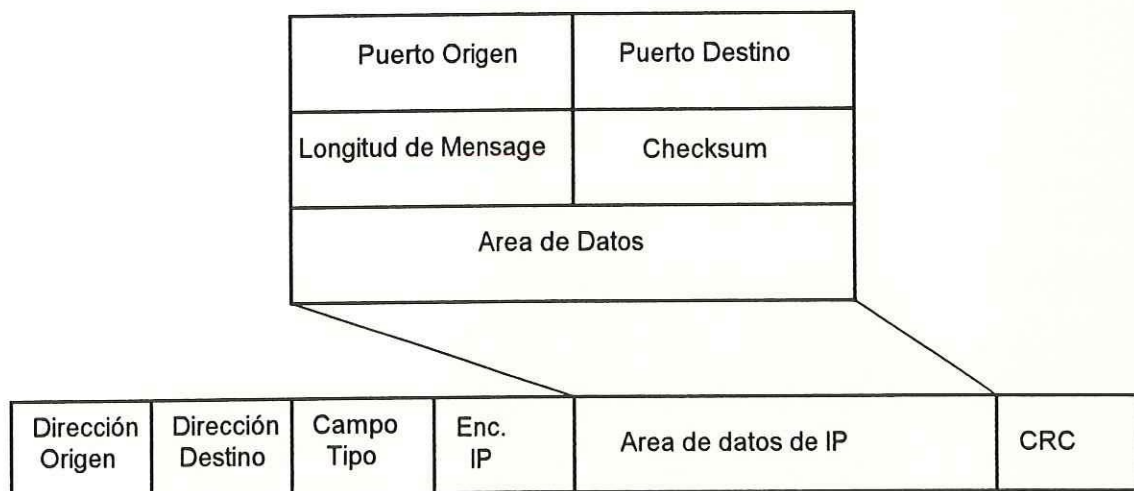


Figura 10.- Encabezado UDP.

Como se puede apreciar de la figura anterior, el procesamiento de un encabezado UDP es mínimo por eso que lo hace un protocolo ligero. Los únicos campos de interés vienen siendo los puertos origen y destino que son los que un protocolo o aplicación maneja directamente con el encabezado.

El protocolo RIP utiliza este protocolo como medio de transporte por el puerto definido con el número 520. Otros protocolos y aplicaciones utilizan un número de puerto diferente. La figura 11 muestra cómo se realiza la multicanalización de datos hacia las capas superiores a través de el sistema de transporte UDP (Naugle, 1994).

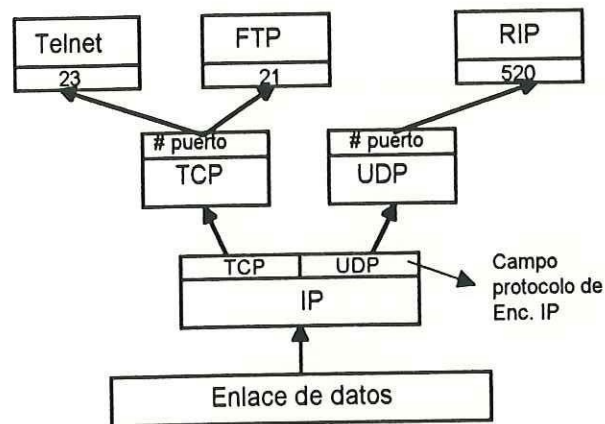


Figura 11.- Proceso de travesía de datos hacia aplicaciones vía número de puerto.

Este protocolo de transporte es utilizado para cuestiones de enrutamiento en lugar de TCP, ya que implica un procesamiento menor y el tamaño de datos adicionales a la información de enrutamiento que se agrega es mínima (tamaño de encabezado UDP es de 8 octetos) por lo que el protocolo es bastante funcional. UDP permite que la aplicación se comunique directamente con él para el transporte de datos, así UDP utiliza los recursos de IP para realizar el envío de datos de la aplicación. Por tanto, el protocolo RIP, central a este trabajo, hace uso de esta característica para hacer una combinación flexible de comunicación RIP - IP para enrutamiento sin necesidad de sobreprocesamiento adicional.

III.4.2.2.-Protocolo TCP (Transmission Control Protocol)

Debido a que IP provee un servicio datagrama para datos, el protocolo TCP permite el acceso a red para programas de aplicación utilizando un servicio de transporte orientado a conexión confiable. Este protocolo es responsable de establecer sesiones entre procesos de usuarios en la Interred, además asegura la comunicación confiable entre dos o más procesos.

Las funciones que permite incluyen:

- * Escuchar peticiones de establecimiento de sesiones que lleguen al nodo
- * Realizar una petición de una sesión a otro nodo.
- * Enviar y recibir datos de una manera confiable
- * Cerrar sesiones de una manera adecuada.

La meta principal del protocolo TCP es la detección de errores para asegurar la integridad de datos, debido a ésto contiene una serie de algoritmos de revisión de errores, así como reglas muy estrictas en el manejo de datos. TCP utiliza números de secuencia y reconocimiento para la comunicación entre nodos a través de la Interred. Estos números identifican el ordenamiento de datos en los paquetes, además para encontrar paquetes perdidos de una manera confiable. El proceso de comunicación de TCP con reconocimientos y números de secuencia se muestra en la fig. 12.

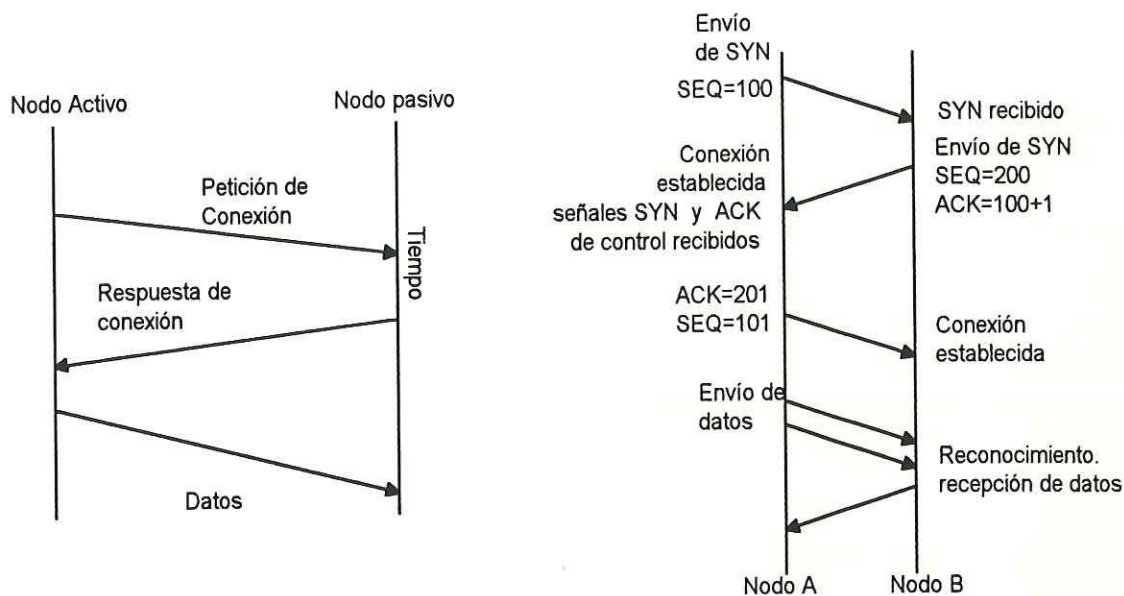


Figura 12.- Diagrama de bloque de la comunicación de TCP. Reconocimientos y secuenciación. (*handshake*).

TCP calcula un número de secuencia (SEQ) para cada octeto de datos, incrementándolo por uno cada octeto que se transmitirá. Este número de secuencia se inicializa como el primer octeto del paquete que se envía. Además, por cada paquete que se transmita debe haber un reconocimiento bidireccional el cual se mantiene con un número (ACK).

Estos números de secuenciación y reconocimiento viajan junto con los datos en un encabezado especial para TCP al igual que los demás protocolos de diferentes capas. No se incluye en este trabajo porque no es relevante, pero el funcionamiento de este protocolo es importante para una descripción de los protocolos TCP/IP.

La transmisión de datos en TCP utiliza un mecanismo llamado *ventana deslizante* (Comer, 1991) que consiste en "recorrer una ventana" sobre los datos a transmitir para que el protocolo identifique cuales octetos han sido transmitidos y reconocidos, cuales octetos han sido enviados pero no reconocidos y de los octetos que estan en espera de ser transmitidos. Este proceso se muestra en la fig. 13.

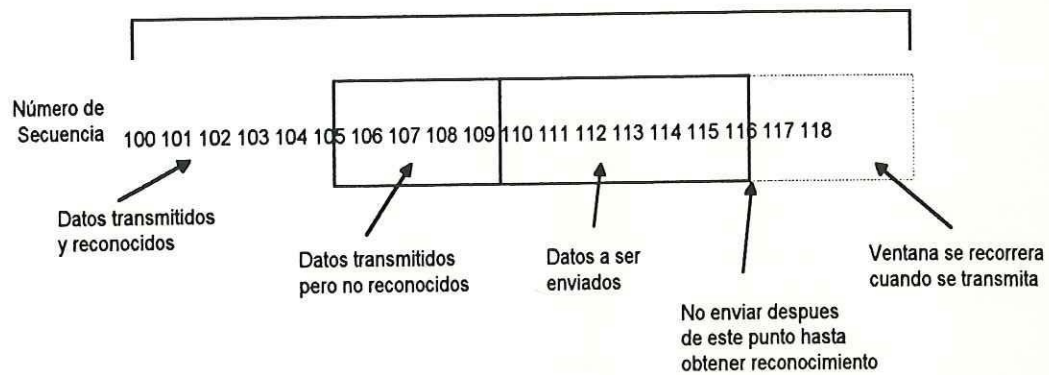


Figura 13.- Proceso de ventana deslizante utilizada para transmision de datos en TCP.

Con estos mecanismos además de los algoritmos de revisión estricta de errores, el protocolo TCP provee un servicio de transporte bastante íntegro para la comunicación a través de una RCA así como una RLD.

III.5.- ENRUTAMIENTO IP.

Esta sección describe los mecanismos utilizados para enrutar datagramas IP a través de una Interred. Se comenzará describiendo el proceso general de enrutamiento IP para resaltar las características genéricas de comunicación RLD de una Interred TCP/IP. Posteriormente se formaliza el protocolo asociado a IP, RIP, sus interacciones y tipos de algoritmos de enrutamiento de la suite de protocolos TCP/IP que permiten la comunicación a nivel RCA.

En una Interred TCP/IP los gateways se agrupan entre sí para formar lo que se conoce como un *sistema autónomo* (Braden, 1987). Los protocolos que rigen el enrutamiento dentro de un sistema autónomo se conocen como IGP's y los manejan el enrutamiento entre los sistemas autónomos se les llama EGP's.

Existen muy pocas implementaciones de EGP's (Mills, 1984) en la actual Interred, todos los demás gateways contienen IGP's. El más popular y aceptado como un estándar de facto es el protocolo RIP que en conjunto con IP proveen el ambiente de RCA para TCP/IP.

En el protocolo IP los paquetes se pueden enrutar en una RLD ó hacia una RCA a través de un gateway, la base para esto está en la dirección destino IP (número de red) contenida en el encabezado IP de cada datagrama (Postel, 1981). El proceso de enrutamiento de cada nodo con el protocolo IP se muestra en la fig. 14.

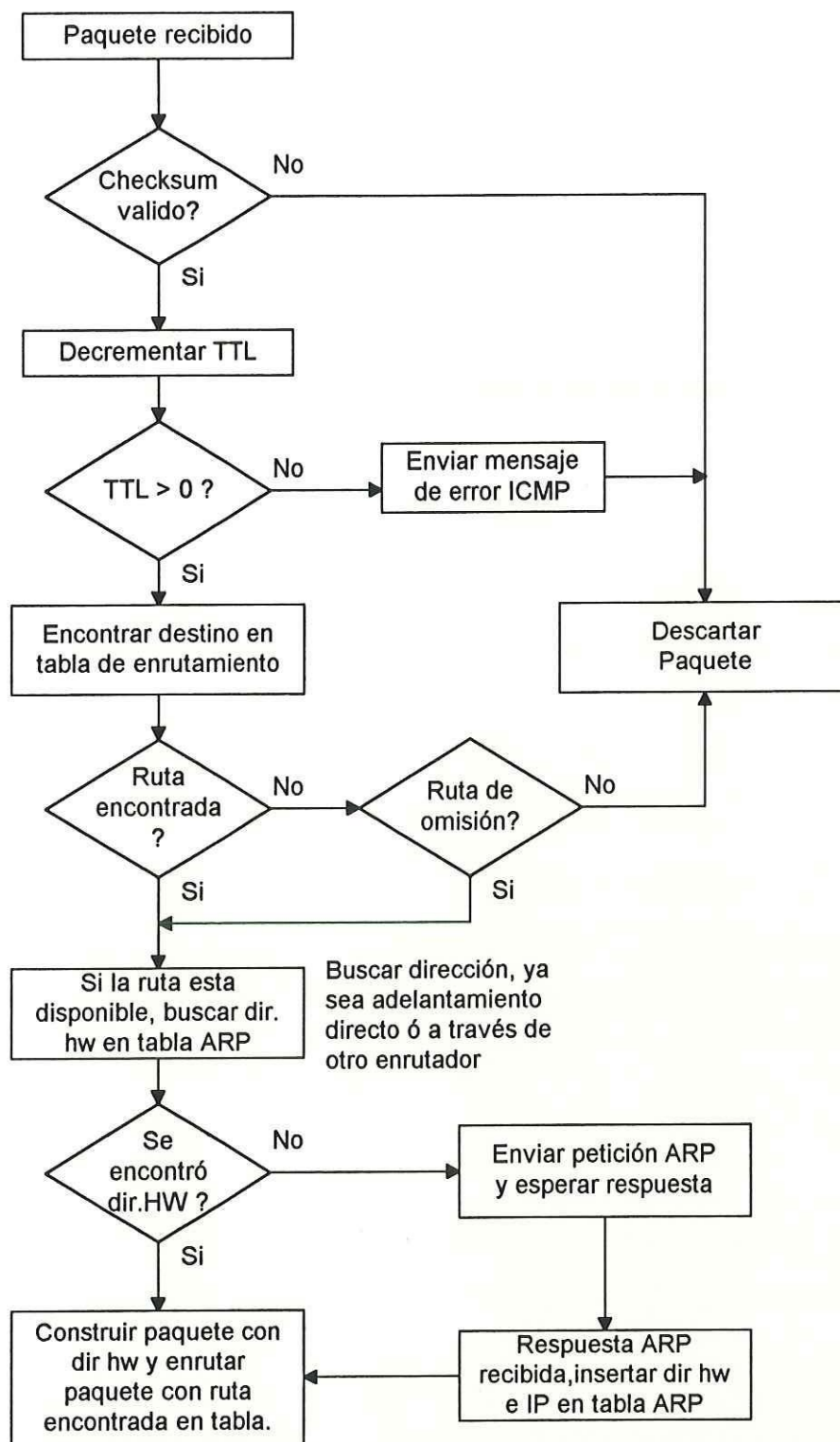


Figura 14.- Procesamiento de enrutamiento del protocolo IP.

Al recibir un paquete, el nodo revisa si el destino es una ruta local, siendo así el nodo busca un mapeo de dirección IP a dirección hardware mediante el protocolo ARP para su envío directo. En caso de no ser resuelto el mapeo, se manda una petición de resolución de dirección a otro nodo. Si el destino no se encuentra en la red local (numero de red destino) el nodo enrutará el paquete hacia otro nodo que pueda alcanzar el destino. Esto es posible mediante la tabla de enrutamiento local del nodo, realizando una búsqueda del siguiente nodo en el camino al destino.

Pero, para el enrutamiento a nivel RCA, se necesita un mejor procedimiento de manejo de rutas, direcciones y hasta errores que puedan ocurrir en una Interred de gran tamaño. Para esto, IP se apoya en algoritmos de enrutamiento dinámico, en sí, protocolos de tipo IGP descritos en un capítulo posterior. En particular, el protocolo RIP.

Algunos protocolos de enrutamiento permiten a los enrutadores la determinación de caminos o rutas hacia redes basados en tablas de enrutamiento que contienen información acerca del número de red destino, la cantidad de enrutadores que se necesitan cruzar para llegar a ella (conocido como *metric*) y el siguiente enrutador a enviar el datagrama.

La tabla de enrutamiento se muestra en interacción con los enrutadores en un escenario de Interred en la figura 15.

Tabla de enrutamiento de Enrutador A.

Número de Red	Siguiente Enrutador	Hops	Tiempo de vida de ruta	Dirección Hw. (Puerto)
132.2.0.0	Ruta directa	1	xxx	1
133.3.0.0	Ruta directa	1	xxx	2
130.1.0.0	Ruta directa	1	xxx	3
134.4.0.0	133.3.0.4	2	xxx	2

Tabla de Enrutamiento de Enrutador B.

Número de Red	Siguiente Enrutador	Hops	Tiempo de vida de ruta	Dirección Hw. (Puerto)
132.4.0.0	Ruta directa	1	xxx	1
133.3.0.0	Ruta directa	1	xxx	2
132.2.0.0	133.3.0.3	2	xxx	2
130.1.0.0	133.3.0.3	2	xxx	2

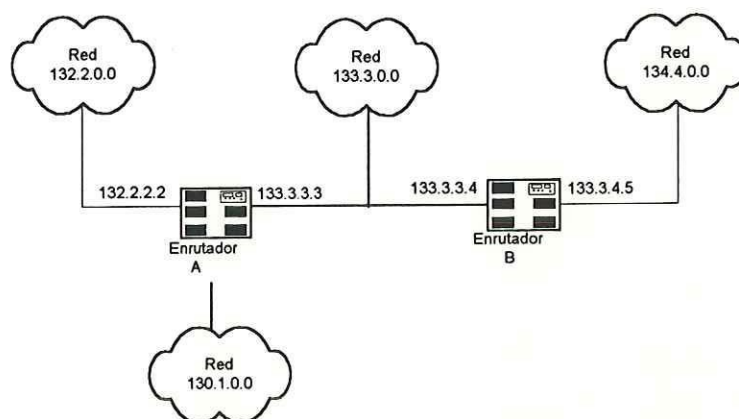


Figura 15.- Interacciones de enrutadores y tablas de enrutamiento.

Los enrutadores intercambian esta información con otros en la Interred detallando así la topología de ésta. El protocolo RIP dicta el proceso de intercambio de tablas de una manera dinámica entre los enrutadores de una Interred, además de realizar una elección de rutas óptima para los paquetes que desean llegar a destinos remotos.

El conjunto de protocolos IP - RIP hacen que la arquitectura de red DoD contenga un motor de enrutamiento bastante flexible y fuerte no sólo para RLD sino para RCA también. El protocolo RIP se describe a continuación para detallar más profundamente el enrutamiento dinámico de TCP/IP.

III.5.1.- Protocolo RIP (Routing Information Protocol).

Este protocolo está basado en un algoritmo de vector-distancia originalmente implementado en la arquitectura de red XNS. Este tipo de algoritmos se conocen también como algoritmos de Bellman-Ford y consiste en mensajes de actualización por pares de (vector,distancia) para determinar el camino más corto hacia una red alcanzable a través de algún enrutador vecino. El protocolo RIP se basa en conteos de pasos ó *hops* hacia sus destinos (Hedrick, 1988).

Este protocolo intercambia la información de rutas hacia destinos, los cuales pueden ser nodos individuales, redes ó un destino especial. La información más importante que maneja RIP es el hop (también llamado *metric*). Cualquier enrutador que utilice RIP tendrá interfaces de red por cada red que pueda alcanzar, a estas se les conoce como redes conectadas directamente a él.

Cada enrutador mantendrá toda esta información en una tabla de enrutamiento como aquellas mostradas en la sección anterior por la figura 3.12. Esta tabla contiene una entrada para cada destino que es alcanzable a través del sistema descrito por RIP. La tabla debe contener por lo menos la siguiente información:

- * Dirección IP de red destino
- * Un costo ó metric asociado a ese destino que representa el costo total de llevar un datagrama a cierto destino. Esta medida es la suma de las medidas asociadas con las redes que deberían atravezarse para llegar al destino. Cuando este valor llega a ser 16, la ruta se marca como inválida. Esto significa que RIP solamente puede recorrer 16 enrutadores como máximo.
- * La dirección IP del siguiente enrutador que pueda adelantar el datagrama en caso de que no se pueda llegar al destino directamente.
- * Temporizador de envejecimiento de entrada de ruta en la tabla para evitar la saturación de rutas en la tabla que no se usen.

Las entradas iniciales en la tabla de enrutamiento consisten en una ruta de omisión que es utilizada en los casos que no se tenga una ruta específica en la tabla y el envío pueda ser adelantado por esa ruta, además de todas las redes que estén conectadas directamente al enrutador.

El protocolo RIP es ampliamente aceptado por ser implementado como parte del sistema operativo UNIX de Berkeley, ya que existe una gran cantidad de computadoras que lo utilizan (Naugle, 1994).

Una de sus principales características es el de poder compartir de una manera dinámica la información (tabla de enrutamiento) y ajustarse a cambios de topología en la Interred como lo podrían ser fallas de enrutadores, etc.

El proceso que lleva a cabo RIP para estas funciones consiste en varias fases: inicialización, actualización y propagación periódica de tablas, y el término adecuado de funcionamiento del protocolo. Para ilustrar este proceso se tiene la figura 16.

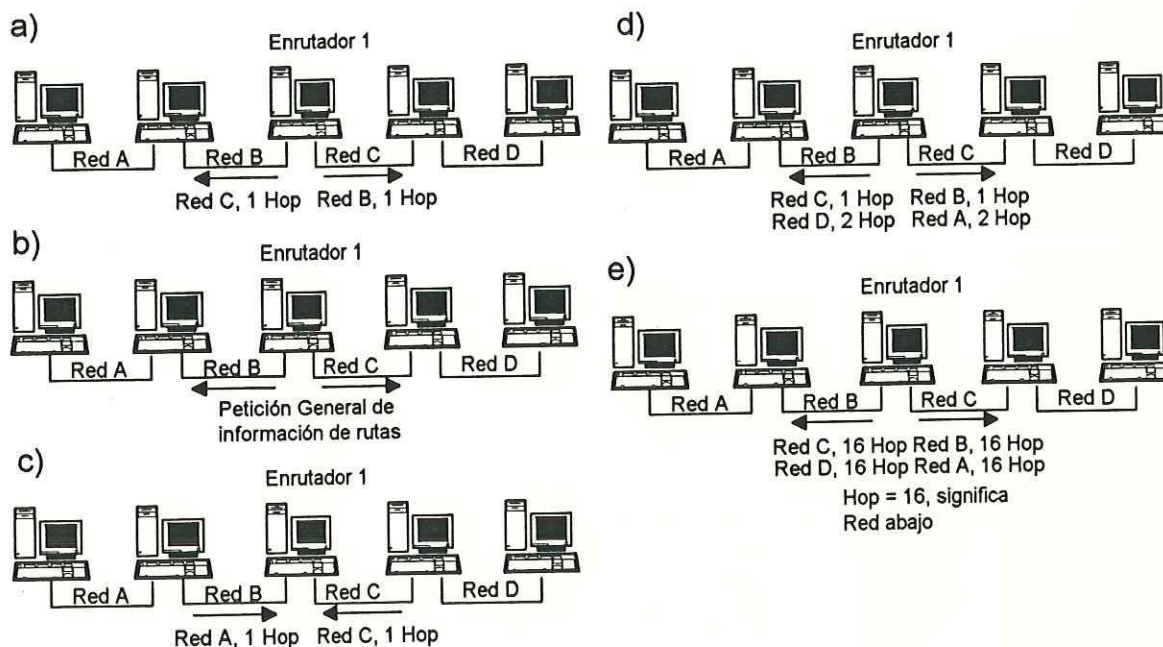


Figura 16.- Fases de inicialización, actualización y término del proceso RIP.

Como se puede observar en a), el enrutador 1 propaga sus redes conectadas directamente a sus enrutadores vecinos para el informe de alcance de estas nuevas redes vía él, inmediatamente después en b), se realiza una petición acerca de otras rutas para establecer su propia tabla de enrutamiento.

En c) llega tal petición y en d) periódicamente (cada 30 ó 60 segundos) se envía la información de rutas del enrutador. Al finalizar en e), el enrutador propaga sus rutas con una medida de 16, lo cual indica que esas rutas ya no serán disponibles a través de él.

El intercambio de tablas de enrutamiento es llevado a cabo por un mecanismo que incluye un encabezado RIP y como servicio de transporte utilizando a UDP. Cada enrutador mantiene un proceso que envía y recibe paquetes por el puerto número 520 vía UDP. Todas las comunicaciones se realizan a través de este puerto enviando las rutas e información adicional en las estructuras mostradas en la figura 17.

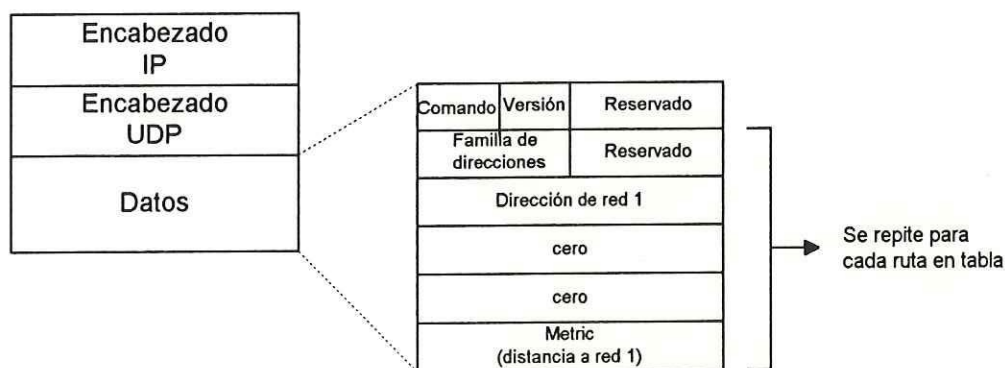


Figura 17.- Encabezado RIP.

Existen 2 tipos de paquetes RIP que son identificados por el campo *comando*: respuesta y petición. Los primeros se utilizan para el envío de la tabla misma y los otros para las actualizaciones periódicas. La versión de RIP que se maneja es la 1, aunque existe la versión 2 de RIP recientemente. Los demás campos definen la tabla de enrutamiento que se envía o se recibe: número de red destino IP, familia de direcciones (actualmente solo existe la núm. 2 que corresponde a las direcciones IP) y la medida o metric asociada a la ruta. Se pueden enviar/recibir hasta 24 rutas debido al protocolo UDP, siendo el máximo tamaño de paquete 512 octetos. El procesamiento de paquetes por el módulo de implementación del protocolo existente en los enrutadores se muestra en la figura 18.

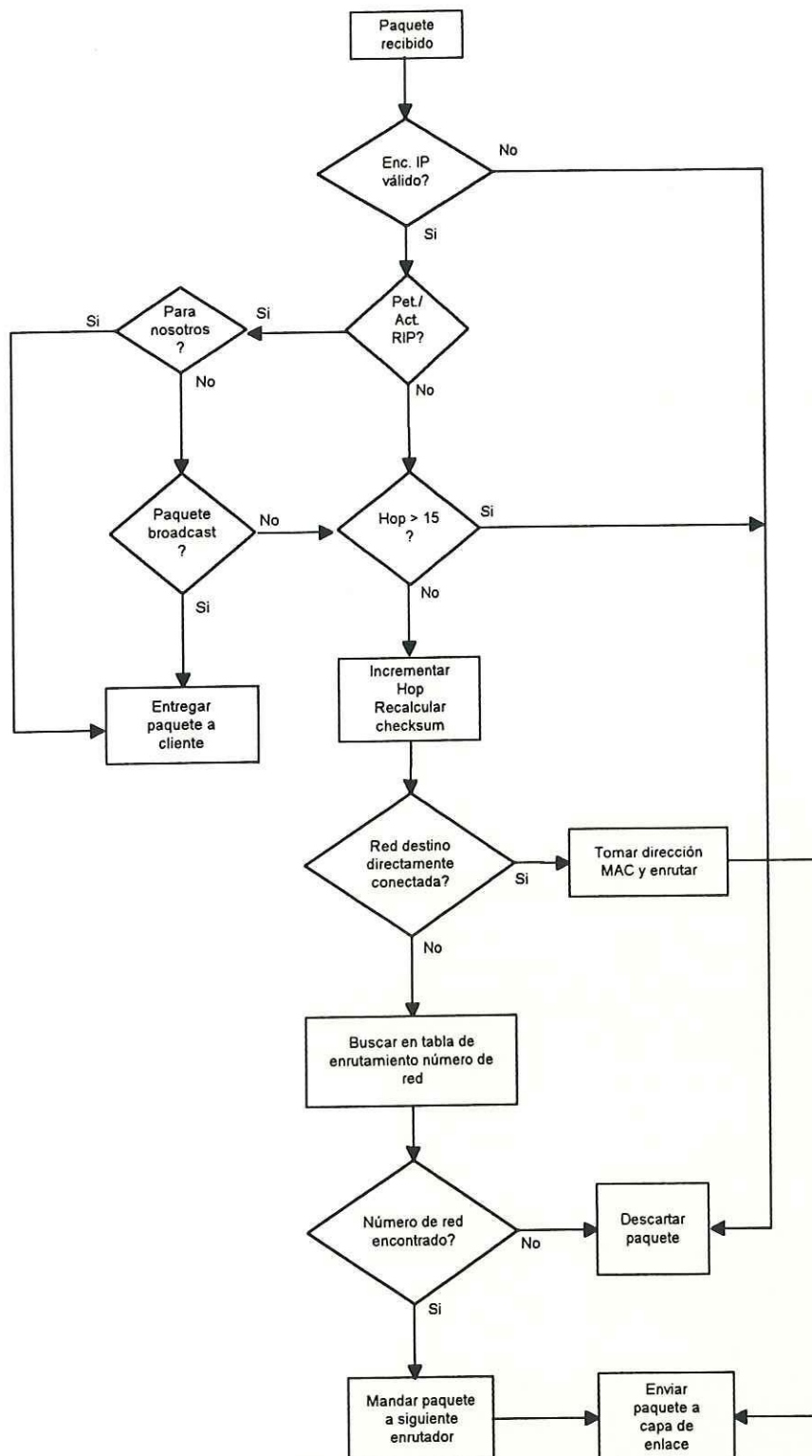


Figura 18.- Procesamiento de enrutamiento del protocolo RIP.

Cada 30 segundos el protocolo RIP genera una respuesta completa a cada enrutador vecino, por tanto la implementación debe manejar un temporizador para cada ruta en la tabla. La tabla de enrutamiento se inicializa cuando se recibe una ruta y al transcurrir 180 segundos de la última vez que se inició una entrada de ruta en la tabla se expira el temporizador y comienza el borrado de la ruta en la tabla así como la propagación de que la ruta es inválida para que otros enrutadores se enteren.

El protocolo RIP maneja genéricamente unas reglas para prevenir inestabilidades que ocurran en la topología de la Interred conocidas como *split horizon* y *actualizaciones disparadas*. Estas reglas se definen en el capítulo V que trata el tema de enrutamiento dinámico más profundamente. En tal capítulo se tomará la descripción de los aspectos más importantes que aporta RIP al enrutamiento en una Interred tanto para esta arquitectura de red como para otras.

Este protocolo reúne muchas características genéricas para el manejo de un enrutamiento adecuado, con un sistema de transporte muy sencillo y eficaz. En este capítulo se expuso la funcionalidad de enrutamiento que tiene TCP/IP gracias a RIP y su combinación con IP. Esto permite analizar el funcionamiento de enrutamiento de una arquitectura de red a nivel de red local y poder comparar, diferenciar e identificar los puntos claves donde se presenta la problemática de interconectividad entre un ambiente de TCP/IP a uno como el de Novell. El siguiente capítulo describe los aspectos de enrutamiento y conceptos básicos necesarios del ambiente de red Novell para posteriormente poder identificar la problemática asociada con la interconectividad RLD - RCA - RLD y poder establecer el manejo de las tablas de enrutamiento que permitan la coexistencia de estos ambientes a su plenitud.

IV.- ARQUITECTURA DE RED NOVELL

IV.1.- Introducción.

Este capítulo examina el funcionamiento de los protocolos de la arquitectura de red definida por la compañía Novell, ya que ésta provee características de enrutamiento en una Red local de Datos (RLD), a diferencia de TCP/IP que incluye dicho ambiente además de cubrir Redes de Cobertura Amplia (RCA). Se incluye una conceptualización de los protocolos XNS ya que de ahí se derivó la arquitectura de red de Novell. A partir de ello se definen conceptos y estructuras utilizados para la comunicación en red con los protocolos de Novell. El análisis sobre Novell es necesario ya que cubre aspectos de enrutamiento a nivel de RLD para poder establecer los problemas que surgen cuando se interconecta una RLD a una RCA.

IV.2.- Protocolos XNS (Xerox Network Systems).

Estos protocolos fueron desarrollados por la compañía Xerox a fines de los años 1970 en conjunto con investigaciones de las compañías Intel y DEC, que en ese tiempo trabajaban con Ethernet. De esto resultó lo que se conoce como la arquitectura de red XNS que salió en 1980. XNS se inició como arquitectura abierta y se diseñó con la idea de interconectividad de vendedores múltiples, por lo que compañías como Novell, 3Com y Banyan han implementado estos protocolos dentro de sus respectivas arquitecturas (Naugle, 1994). Los protocolos XNS suponen que la inteligencia en la conexión entre canales de comunicación reside en un dispositivo llamado *enrutador* (router). La unidad de información transferida en la red se llama *paquete internet* (internet packet) y debe contener direccionamiento adicional para asegurar el envío del paquete a la red destino.

El paquete se envía en un servicio orientado a no conexión también llamado servicio *datagrama*. La arquitectura XNS se define en 5 niveles o capas numeradas del 0 al 4 parecidas a las del modelo de referencia OSI, tal como lo muestra la Fig. 19.

Nivel 4 ---	Aplicación	Protocolos de Aplic.
Nivel 3 ---	Control	Courier, Clearinghouse
Nivel 2 ---	Transporte	RIP, SPS, PEP
Nivel 1 ---	Direccionamiento	IDP
Nivel 0 ---	Transmisión (medio)	Ethernet

Figura 19.- Arreglo en capas de la Arquitectura XNS y sus protocolos.

En el nivel 1, se encuentra el protocolo IDP que al igual que el protocolo IP se encarga de enrutar, direccionar y enviar paquetes internet con servicio datagrama. En el nivel 2 esta el protocolo RIP que es de interés en este trabajo. En este protocolo, cada enrutador tiene una tabla y realiza revisiones para determinar la mejor ruta para cada paquete internet a un destino. La tabla se mantiene o se actualiza con información transmitida o recibida entre aplicaciones mientras los diversos enrutadores se informan entre ellos de los cambios de la topología de la inter-red.

IV.3.- Ambiente de Red NetWare de Novell.

La implementación de protocolos de la compañía Novell en sus productos del sistema operativo NetWare han sido muy extensos en ambas capas de transporte y de red de la arquitectura XNS. El protocolo IPX de Novell es casi idéntico que el IDP de XNS (Naugle, 1994). La arquitectura de red de Novell se muestra en la figura 20 con un análisis comparativo entre las capas del modelo OSI.

<i>OSI</i>	<i>Novell</i>		
Presentacion	Protocolos	de	Aplicacion
Aplicacion	NCP	SAP	RIP
Sesion	NetBIOS		
Transporte	SPX		
Red	IPX		
Enlace	Protocolos	de	acceso
Fisica	Ethernet,	Token ring,	Etc..

Figura 20.- El modelo OSI y la arquitectura de red Novell.

Existen diferentes componentes en el ambiente de red NetWare que son los básicos y claves para acceder y comunicarse en la RLD. La comunicación se define como *cliente/servidor*. En este ambiente, existen 2 entidades físicas primarias: La estación de trabajo o host (Cliente) y el servidor (Server) conectados vía una RLD (como Ethernet, Token Ring, ARCnet, ProNET-10, FDDI) como muestra la figura 21.

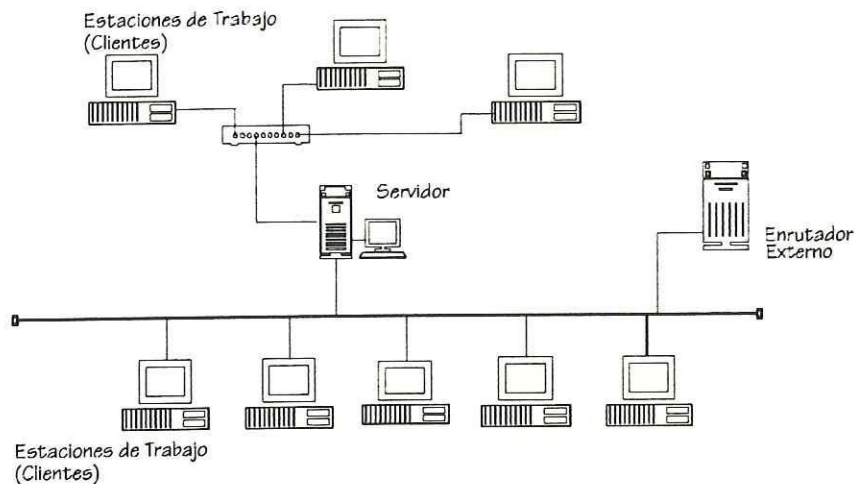


Figura 21. - Escenario de una red Novell.

Los clientes realizan peticiones al servidor para servicios de manejo y transferencia de archivos e impresión. La capacidad de intercambiar información cliente-cliente o cliente-servidor se debe al protocolo IPX derivado del IDP. El modelo cliente/servidor de Novell establece que cada estación envía una sola petición y ésta debe esperar una respuesta del servidor para poder mandar otra, es decir, un cliente depende totalmente del servidor.

IV.4.- Limitantes de una RLD Novell para su interconectividad RCA.

Existen varias desventajas en el ambiente NetWare cuando se trata de enrutamiento o interconectividad en una RCA: NetWare fué diseñado originalmente para un ambiente de RLD y no de RCA por lo que no provee ninguna función genérica de enrutamiento en una inter-red como en el caso de TCP/IP.

El sistema operativo NetWare puede abarcar casi cualquier grupo de trabajo pequeño de RLD fácilmente, pero cuando se manejan cientos de nodos, se empiezan a tener ciertos problemas debido al funcionamiento de los 4 protocolos principales de Novell: IPX, SAP, RIP y NCP, siendo el más crítico de ellos IPX (Thomas, 1993).

- * NCP (Network Core Protocol) - Acarrea mensajes de aplicaciones entre clientes y servidores. (Login, Lectura/Escritura a disco, impresión, etc.)
- * SAP (Service Advertising Protocol) - Utilizado por los servidores para propagar su disponibilidad a otros nodos de otra red.
- * RIP (Routing Information Protocol) - Permite que los enrutadores NetWare se mantengan informados de caminos a otras redes Novell ó a otro servidor remoto en el caso de que el servidor de una red falle.

En un ambiente de RLD, estos protocolos manejan bien sus tareas, el ancho de banda es suficiente para el tráfico generado por pocos clientes y pocos servidores. En una ambiente de RCA, donde varios segmentos de RLD se interconectan vía enrutadores, se suscitan varios problemas:

- * **Direccionamiento:** En el contexto del formato de direcciones de RCA de TCP/IP y otros protocolos, IPX difiere. Además, en una RLD Novell no se tiene control sobre las asignaturas de direcciones de redes que deben ser únicas; es decir, existen números de red duplicados, lo cual presenta un problema ya que se desea interconectar 2 redes con un mismo número de red.

En TCP/IP la asignatura de direcciones es administrada globalmente y es flexible en cuanto a que presenta un esquema para poder crear subredes a partir de un mismo numero de red. La ventaja de direcciones de red IPX es que sólo se asigna a un servidor y no a todos los clientes en la RLD.

* **Rendimiento de enrutadores IPX:** Existen muchos enrutadores externos para una RLD Novell, pero representan sólo 1/5 del rendimiento que ofrecen los enrutadores IP.

* **Tamaño de paquete:** El manejo de paquetes grandes en una RCA es favorable ya que no se requiere de sobrecargar la RCA con mucho tráfico (su ancho de banda no es tan grande como el de una RLD); es decir, se reduce el número de encapsulación de paquetes y de reconocimientos que se necesitan enviar. Las implementaciones de IPX son en base al protocolo IDP que maneja 576 octetos como máximo, o sea, paquetes pequeños. Esta restricción baja el rendimiento de aplicaciones que necesitan un tiempo de respuesta rápido cuando se lleva a cabo una comunicación vía una RCA.

* **Rendimiento General:** Novell creó al protocolo NCP como protocolo de RLD, funciona esperando que el ancho de banda entre cliente y servidor sea suficiente. Pero en el modelo de Novell, cada paquete que envía el cliente es una petición al servidor y el cliente debe esperar una respuesta. Por lo tanto, si el servidor está remoto (el reconocimiento debe cruzar una RCA) se introduce un *retardo de serialización* (Thomas, 1993) que baja el desempeño de comunicación en un 30 a un 40 % (Leibig, *et al*, 1990).

A partir de la version 3.11 de NetWare, Novell incluyó servicios de enrutamiento vía TCP/IP sólo en los servidores y no en los clientes para poder incorporar estaciones de trabajo UNIX en su ambiente de RLD y no sólo computadoras personales (IBM PC's); de esta manera los servidores NetWare pueden hablar con clientes TCP/IP. Si una estación desea comunicación con un nodo TCP/IP, éste deberá cargar la pila de protocolos de TCP/IP (adicional a la de IPX necesaria para cualquier comunicación) pero el servidor aún maneja el tráfico y permisos de la comunicación.

Así, NetWare se comunica con enrutadores IP vía RIP, estableciendo una lista de enrutamiento interna despues de cargar toda la pila de protocolos de TCP/IP en el servidor.

Pero, dado que las tablas de enrutamiento no pueden cambiarse dinámicamente el servidor con TCP/IP debe quitar los módulos de TCP/IP y cargarse de nuevo; ésto significa que se terminarán sesiones TCP. El enrutamiento en NetWare es estático en vez de dinámico. Otro punto a notar es que todos los clientes dependen del servidor para comunicación RCA incluso RLD, centralizando el enrutamiento y las tablas en el servidor de la RLD, sobrecargando a éste por el procesamiento adicional de enrutamiento.

El enrutamiento a través de NetWare en una RCA se realiza con lo que se conoce como *tunelaje* (Provan, 1991). Este esquema utiliza una inter-red TCP/IP para llegar a otro servidor NetWare, es decir, la información de IPX viaja dentro de un paquete TCP/IP (encapsulación). Esto agrega una sobrecarga mínima de un 20 % al procesamiento del servidor (enrutador) y por lo tanto la rapidez de la comunicación decae.

De acuerdo a lo anterior, Novell se inclinó hacia la utilización de protocolos que sí puedan soportar comunicación inter-red tal como lo realiza TCP/IP (enrutamiento dinámico, direccionamiento, etc.) por ser éstos los más utilizados y aceptados mundialmente, son los que esta compañía adaptó. Por lo tanto, se concluye que se debe realizar mayor énfasis en la pila de protocolos que permita ese ambiente de interconectividad en ambas partes: RLD y en una RCA, para realizar una adaptación al ambiente de NetWare y a sus servicios centralizados de red local.

Posteriormente se propondrá una alternativa a la solución que existe de interconectividad inter-red de RLD y RCA con los protocolos IPX y TCP/IP. Para esto es necesario analizar a fondo los detalles del funcionamiento del protocolo IPX de Novell, así como en la adaptación que se llevó a cabo en el protocolo RIP.

IV.5.- Protocolos de Capa de Red

Para este trabajo de tesis no es necesario involucrarse con los protocolos superiores a esta capa, ya que intervienen en un mínimo para cuestiones de enrutamiento. El motor de transporte viene definido por IPX, que es utilizado para toda comunicación ya sea servidor ó cliente.

IV.5.1.-Protocolo IPX (Internetwork Packet Exchange):

Esta sección describe el mecanismo de enrutamiento genérico de comunicación RLD de Novell dictada por el protocolo IPX. IPX es la interface entre el sistema operativo NetWare (compuesto principalmente por el protocolo NCP) y los métodos de acceso a la RLD. IPX provee direccionamiento para transmisión en red además de ser el sistema de envíos y recepciones de datos. Debido a que IPX es una implementación casi directa de IDP de XNS, y éste a su vez fue construido para estar específicamente sobre Ethernet, IPX se diseñó para un ambiente de RLD y no contiene las características necesarias como IP para el manejo de direcciones ni mecanismos de transporte en RCA. IPX se encarga también de enrutamiento de paquetes de datos a través de una RLD, la cual permite que redes de tipo Ethernet, Token ring, ARCnet, etc. intercambien información transparente e independiente del medio que se utilice.

El paquete de datos IPX consiste en dos partes: un encabezado y una porción de datos. El tamaño mínimo de un paquete IPX sin incluir la longitud del marco MAC es de 30 octetos (solo el enc.) y como máximo 65,535 octetos, aunque históricamente se especificó como 576 octetos máximo, a causa de IDP, incluyendo el encabezado dejando 546 octetos para datos. El medio de transmisión es el que realmente limita (Naugle, 1994).

El paquete IPX es puesto inmediatamente después del marco MAC y antes de los datos. El paquete de datos puede incluir el encabezado de un protocolo de capas superiores. El encabezado de un paquete IPX es el paquete de registro de datos que se pone ya sea dentro del marco Ethernet o Token Ring para transmisión, o extraído del marco de recepción. Este encabezado consiste de 30 octetos y se muestra en la figura 22.

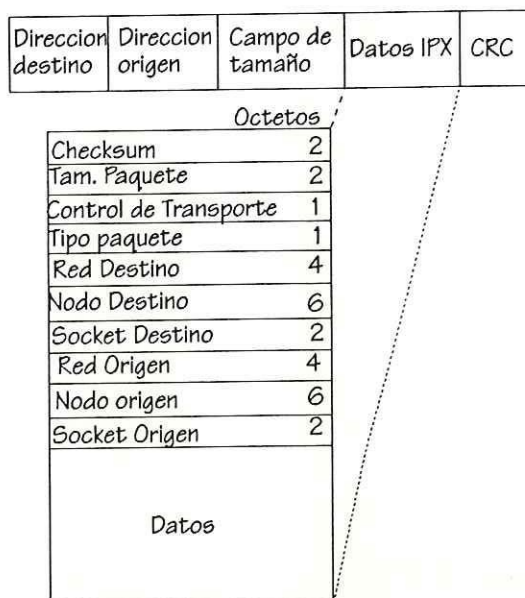


Figura 22.- Encabezado IPX.

La información contenida en el encabezado es la siguiente:

Checksum - (2 octetos) Xerox define este campo en su protocolo del XNS correspondiente a IPX, es por eso que aparece en su encabezado. Utilizado para cuestiones de verificación de errores. No es utilizado por Novell.

Tamaño de paquete - Tamaño del paquete completo, incluyendo datos y encabezado. Paquete mínimo es de 30 octetos y el máx. es de 576 octetos.

Control de Transporte - Un paquete puede cruzar hasta 16 enrutadores. IPX pone un valor de 0 cuando el paquete es transmitido originalmente y es incrementado cada vez que se cruza un enrutador. Si llega a tomar el valor de 16 el paquete se descarta.

Tipo de Paquete - Define varios tipos de paquetes para distinguir el protocolo de capa superior que lo envió. Para SPX este valor es de 0x05; para NCP es 17; para RIP 0x01; para SAP es 0x04. Si este campo es cero se desconoce que tipo de paquete es.

Red Destino, Nodo Destino, Socket Destino - El primero de éstos identifica a la red en donde se encuentra el nodo destino. El segundo indica a dónde se enviará un paquete, es decir, identifica el nodo objetivo por su dirección física única. El tercero es la aplicación en cuestión llamada *socket*. Si el valor 0x-00-00-00-00 está en el campo de red destino, el destino del paquete es la misma red que la origen, aquí no hay dirección broadcast. Un enrutador no pasará el paquete. Para el campo de nodo destino, la dirección 0xFF-FF-FF-FF-FF-FF es broadcast y la 0x00-00-00-00-00-01 indica al paquete que se dirija al servidor NetWare.

Red Origen, Nodo Origen, Socket Origen - Correspondientes a la dirección del transmisor completa. (Igual que en Destino). El campo nodo origen no puede contener una dir. broadcast, si fue enviado con 0x00-00-00-00-00-01 el paquete lo transmitió un servidor NetWare.

Porción de Datos - Campo de 0 - 546 octetos de longitud e involucra los datos actuales a enviar.

IV.5.2.- Direcciones y Sockets:

Una dirección IPX consiste de número de red, nodo y socket. Para paquetes IPX la dirección destino puede contener un grupo de direcciones (*broadcast*) .

Los paquetes se transmiten especificando la dirección IPX, pero la transferencia es a través de un *socket*. Los números de socket deben ser diferentes para cada aplicación, es decir, el socket especifica cuál aplicación se va comunicar con cuál al igual que el significado de números de puertos de TCP/IP.

Un ejemplo de una dirección de inter-red Novell sería de la siguiente manera:

00000004:040660390000:4002 donde:

00000004=Número de red, 040660390000=Número de nodo, 4002=Número de socket

Como se puede apreciar del formato de dirección IPX, éste difiere en su totalidad del formato de direcciones IP, por lo que se debe emplear una especie de mapeo de direcciones para su manejo durante la transferencia de RLD a RCA y viceversa.

Cada participante en la RLD es identificado por un *nombre de usuario* que debe ser único. Esta identificación es usada por los usuarios en cualquier estación de trabajo de la RLD. Aquí se hace una analogía de dirección IP única en una Interred con el nombre de usuario único en la RLD, con la excepción de que una dirección IP no puede cambiarse y asignarse a otro nodo como en el caso de un nombre de usuario que puede estar asignado a cualquier estación en cualquier instancia de su existencia en la RLD.

Este nombre de usuario es asociado con un número de conexión hacia el servidor. Un nombre de usuario puede tener múltiples instancias en la RLD al mismo tiempo. El protocolo IPX, en conjunto con servicios de conexión que mantiene a través del protocolo NCP, utiliza este número de conexión para realizar comunicación con otras estaciones en la RLD e inclusive con el servidor mismo (Rose, 1990).

Los servicios de conexión pueden ser utilizados por alguna aplicación, creando una interface de programación flexible para la comunicación con el protocolo IPX. Estos servicios son numerosos, pero los más utilizados para interés de enrutamiento son:

- * `IpxGetLocalTarget` - Toma la dirección Hardware de un nodo que pueda enrutar el paquete a ser enviado.
- * `IpxSendPacket` e `IpxListenForPacket` - Utilizadas para el envío y recepción de paquetes a través de sockets.
- * `GetConnectionNumbers` - Toma los números de conexión asociados a un nombre de usuario de la base de datos del servidor llamada *bindery*.
- * `IpxGetInternetAddress` - A partir de un número de conexión el servidor regresa la dirección IPX completa al cliente.
- * `GetConnectionInformation` - Toma datos de enlace con el servidor de algún nodo a partir de su número de conexión.
- * `IpxOpenSocket` y `IpxCloseSocket` - Realizan la apertura de un socket para comunicación entre aplicaciones así como su cierre.

El enrutamiento IPX se describe en la siguiente sección mostrando sus diferentes fases de inicialización y el proceso mismo en conjunto con la utilización de estos servicios y conceptos vistos. Además, se incluyen algunos conocimientos adicionales para establecer el enrutamiento realizado por una RLD Novell.

IV.6.- Enrutamiento IPX

Esta sección describe el enrutamiento genérico que realiza IPX en su comunicación a través de una RLD. La descripción de esta infraestructura es necesaria ya que se utilizará para su incorporación dentro de los algoritmos de enrutamiento para la adaptación del ambiente RLD a RCA.

El protocolo IPX utiliza la dirección nodo directamente de la dirección hardware de la tarjeta de red (no es necesario una resolución de dirección como en el caso de TCP/IP) de tal manera que IPX necesita saber este tipo de direcciones tanto para estaciones de origen como para destinos (Rose, 1990).

El proceso de enrutamiento de IPX es llevado a cabo con la combinación de los servicios de conexión que ofrece NetWare y el protocolo IPX. Este proceso se muestra en la figura 23.

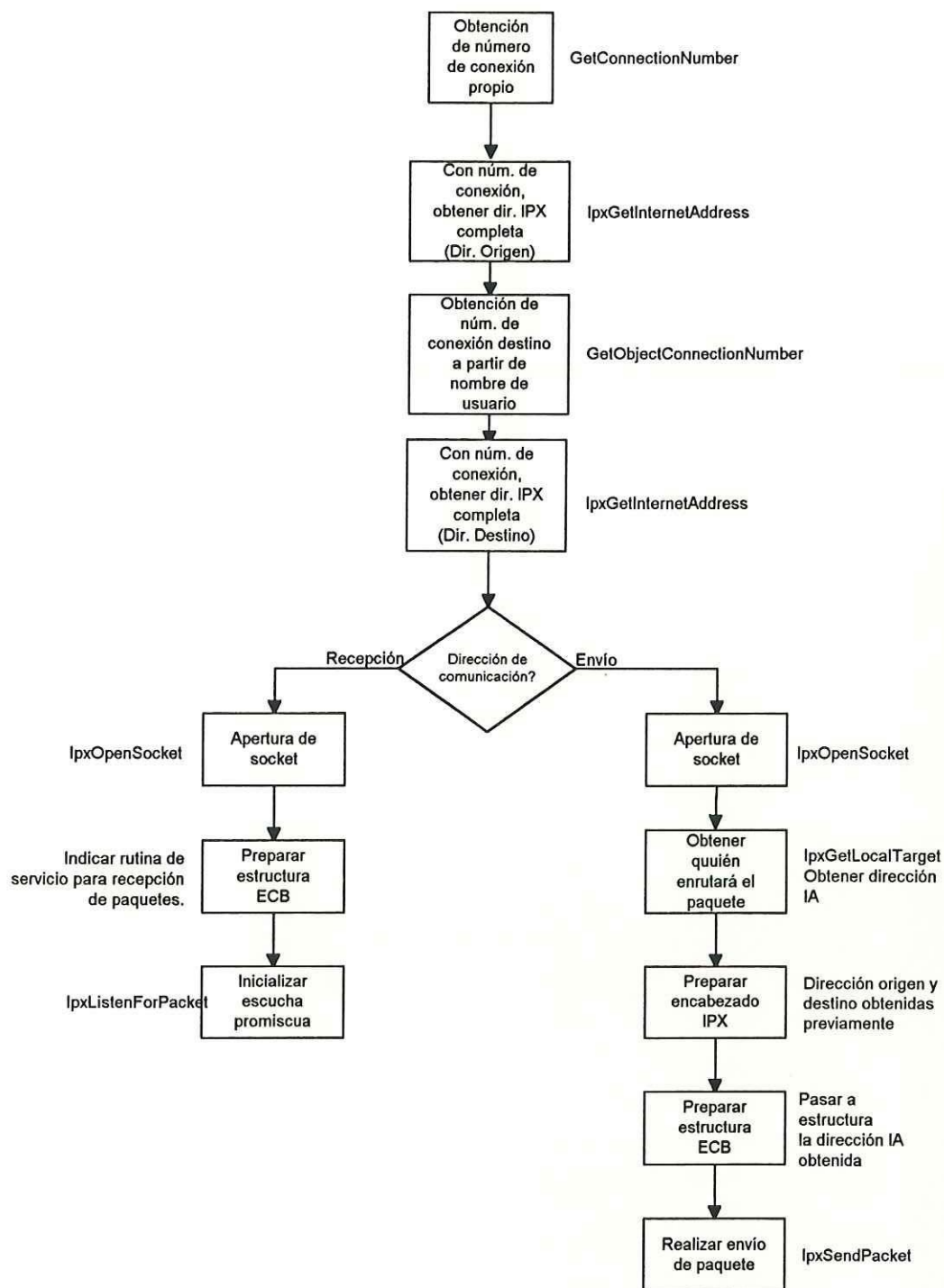


Figura 23.- Proceso de Enrutamiento IPX.

Este proceso se divide en recepción y envío de paquetes. Para ambos casos se debe realizar una búsqueda de la dirección IPX del origen tanto como destino a través de servicios de conexión. Además, se debe realizar una apertura de socket de aplicación para poder establecer una comunicación.

Existen otros requisitos necesarios para iniciar la comunicación de estación - estación para IPX (Davis, 1990), estos son:

- * Inicialización de una estructura llamada ECB (Event Control Block) que controla los eventos (recepción y envío).
- * Inicialización del encabezado IPX con las direcciones obtenidas en pasos iniciales del proceso.
- * Establecer una rutina de servicio de recepción de paquetes para que ésta sea promiscua.

La estructura ECB se mantiene para cada evento que ocurra de comunicación, además contiene información relevante a ésta como sockets utilizados, áreas de recepción (buffers), etc.

Existe información importante en el ECB para enrutamiento que se le conoce como IA (Immediate Address). Este campo del ECB es una dirección de hardware del siguiente servidor que pueda llevar a su destino al paquete que una estación desee enviar.

Por lo tanto, al servidor se le envía a través de la función `IpxGetLocalTarget`, la dirección destino completa IPX y éste responde con la dirección hardware (de nodo) ya sea de él ó de otro servidor. El IA es puesto en un campo de la estructura ECB para que el protocolo IPX a través de la función `IpxSendPacket` realice el envío adecuado apoyado del ECB.

Para cuestiones de enrutamiento de RCA este esquema no es tan flexible como el genérico de TCP/IP. Para la adaptación hacia una RCA, Novell implementó en los servidores un servicio de enrutamiento utilizando el protocolo RIP, pero éste, como se mencionó en secciones anteriores, es para propósitos de propagación de información entre servidores para sus servicios en vez de proporcionar interconectividad a través de una RCA.

Esto limita al enrutamiento de Novell como si fuera un sistema cerrado, sólo el protocolo IPX podrá fluir en una Interred con este tipo de implementación de RIP. Además, esta implementación no es adecuada por las características del modelo de comunicación de Novell, es decir, el protocolo RIP presenta un bajo desempeño debido a los reconocimientos entre cliente - servidor, aquí RIP no está en un ambiente distribuido como para el que fue diseñado para su funcionalidad.

Claramente se ve que el ambiente de una RLD como en el caso de Novell contiene varias limitaciones en cuanto a enrutamiento para una RCA. En los capítulos posteriores se concentra en la adaptación de las características de enrutamiento de los protocolos TCP/IP así como IPX para establecer un modelo que permita la interoperabilidad a nivel capa de red entre ellos.

Hasta este punto se han analizado dos grandes arquitecturas que engloban características genéricas de ambientes de RLD y RCA por lo que falta definir más a fondo los algoritmos de enrutamiento. El siguiente capítulo describe esto para definir con precisión los elementos necesarios para dicha interconectividad y poder tener los conceptos necesarios para proponer una solución a la problemática en cuestión.

V.- ALGORITMOS DE ENRUTAMIENTO DINAMICO

V.1.- Introducción.

Para obtener una solución efectiva de la problemática asociada a la interconectividad RLD - RCA que lleva al manejo de tablas de enrutamiento, es necesario entender lo que significa enrutamiento dinámico en una Interred y qué tipos de algoritmos están relacionados, además de un análisis funcional de ellos (requerimientos de implementación).

Este capítulo profundiza en estos temas enfocando principalmente el interés al modelo funcional de enrutamiento de TCP/IP y no al de Novell, ya que este último no brinda un panorama tan bien definido como TCP/IP para una RCA, como se ha visto en capítulos anteriores. Se definen los problemas asociados en una ambiente de RCA, los tipos de metodologías y algoritmos utilizados para dar solución, así como también los requisitos que deben reunir los enrutadores para llevar a cabo un enrutamiento de una manera adecuada.

V.2.- Algoritmos de enrutamiento para enrutadores.

Los enrutadores utilizan algoritmos o protocolos de enrutamiento para conocer la topología de la Interred principalmente, de esta manera, pueden elegir una ruta óptima que puede estar basada en diferentes factores dependiendo del algoritmo utilizado para enrutar. Los algoritmos de enrutamiento se clasifican en dos clases: Algoritmos de Vector-Distancia y Algoritmos de Estado de Enlace (Narten, 1989). Los primeros intercambian tablas de enrutamiento unos con otros, mientras que los otros propagan el estado de la topología actual de la Interred.

Este trabajo de tesis hace uso de un algoritmo de vector-distancia por lo que se su análisis será más completo.

V.2.1.- Algoritmos de Vector-Distancia.

Este tipo de algoritmos se les conoce con el nombre de Bellman-Ford, utilizados originalmente en la ARPANET (Spragins, 1991). Los mensajes de actualización consisten en pares (destino,metric). Conceptualmente, un enrutador que genera mensajes de actualización, propaga rutas de destinos que éste puede alcanzar, con un metric ó medida asociada a ellas.

De las actualizaciones recibidas, el enrutador selecciona de cada destino el siguiente enrutador (llamado hop ó paso) que propaga ese destino con el metric más bajo. Existe un valor especial de metric llamado *infinito* que denota caminos que no son alcanzables y que no se deben utilizar.

Los detalles individuales de algoritmos de vector-distancia dependen de sus implementaciones. En algunos casos, el metric asociado representa costos de enlace fijos y otros lo varían para reflejar carga de red ó retardos.

Estos algoritmos están basados en una tabla que contiene los pares (destino,metric), lo cual da la mejor ruta a cada destino en el sistema. A ésta se le llama *tabla de enrutamiento* (Comer, *et al*, 1991). Claro está que para definir cual es la mejor ruta se debe poder medir ese "mejor", y ésto se refiere al metric ó medida.

En redes simples, es común usar medidas que simplemente cuenta cuantos enrutadores debe pasar un mensaje. En redes más complejas una medida se escoge para representar la cantidad total de retardo que el mensaje sufre, el costo de enviarlo, u otra cantidad que pueda ser minimizada. El requerimiento principal es que pueda ser posible representar la medida como una suma de costos para hops individuales.

Formalmente (Hedrick, 1988), si es posible llegar de una entidad i a una entidad j directamente, entonces se asocia un costo $d(i,j)$ al paso (hop) entre i y j . En el caso normal donde todas las entidades en una red dada son consideradas iguales, $d(i,j)$ es el mismo para todos los destinos en una red dada, y representa el costo de utilizar la red. Para conseguir la medida de una ruta completa, solo se suman los costos de pasos individuales que hace la ruta.

Sea $D(i,j)$ la medida de la mejor ruta de la entidad i a la j . Se debe definir para cada par de entidades. $d(i,j)$ representa los costos de pasos individuales. Formalmente, sea $d(i,j)$ el costo de ir de la entidad i directamente a j . Es infinito si i y j no son vecinos inmediatos ($d(i,i)$ es infinito, esto es, no se considera esto como una conexión directa). Como los costos son aditivos, se puede mostrar que la mejor medida debe ser descrita por:

$$\begin{aligned} D(i,i) &= 0 && \text{para toda } i \\ D(i,j) &= \min_k [d(i,k) + D(k,j)] && \text{de otra manera} \end{aligned}$$

y que las mejores rutas empiecen yendo de i a aquellos vecinos k para los cuales $d(i,k) + D(k,j)$ tenga un valor mínimo. Se puede limitar la segunda ecuación a aquellas k que son inmediatos vecinos de i . Para los demás, $d(i,k)$ es infinito, por lo que el término que las involucra no puede ser el mínimo nunca.

Se puede calcular la media por un algoritmo sencillo basado en ésto. La entidad i hace que sus vecinos k envíen sus estimaciones de la///s distancias al destino j . Cuando i obtiene las estimaciones de k , le agrega $d(i,k)$ a cada estima. Esto simplemente es el costo de atravesar la red entre i y k . De vez en cuando i compara los valores de todos sus vecinos y elige la más pequeña.

El algoritmo converge a las estimaciones correctas de $D(i,j)$ en tiempo finito en ausencia de cambios topológicos. Con este algoritmo se pueden enviar actualizaciones asíncronas de acuerdo al reloj de cada entidad. Las actualizaciones se pueden descartar, pero no todas. Como no hay especificación para condiciones iniciales para $D(i,j)$, el algoritmo se puede ajustar a cambios, cuando el sistema cambia, el algoritmo se mueve hacia otro equilibrio utilizando el último como punto de comienzo. Es importante que el algoritmo converja en un tiempo finito sin importar el punto de comienzo, ya que de otra manera con ciertos cambios se puede llegar a un comportamiento no-convergente.

Los nodos que no funcionan como enrutadores (nodos que sólo están conectados a una sola red) no necesitan mandar mensajes de actualización. La única información que pueden proporcionar es $D(i,i) = 0$ que por lo visto, no se necesita.

Básicamente, la especificación para el algoritmo de vector-distancia utilizado para enrutamiento que estará activo en cada entidad que participe en enrutamiento, queda de la siguiente manera:

1. Mantener una tabla con una entrada para cada destino posible en el sistema. La entrada contiene la distancia D al destino y el primer Gateway en la ruta hacia esa red.

2. Periódicamente enviar actualización de enrutamiento a cada vecino. La actualización es enviada como un conjunto de mensajes que contienen toda la información de la tabla de enrutamiento. Contiene una entrada para cada destino y su distancia a él.
3. Cuando una actualización llega de un vecino k , se agrega el costo asociado con la red que es compartida con k (la red por la cual la actualización llegó). Llamar la distancia resultante d . Comparar las distancias resultantes con las de las entradas de la tabla. Si la nueva distancia d para n es menor que el valor existente, adopta la nueva ruta, esto es, cambiar la entrada de la tabla para n con medida d y gateway k . Si k es el gateway por el cual la ruta existente vino, utilizar la nueva medida aún cuando sea mayor que la existente. (puede que sea demasiado baja la medida existente).

La simplicidad de algoritmos de vector-distancia contribuye a su amplia popularidad. Utilizan únicamente información local y los enrutadores sólo propagan información a sus vecinos. Aunque existen problemas con este tipo de algoritmos como en el caso del conteo a infinito detallado posteriormente así como también ciclos, se tienen métodos para dar solución a este tipo de problemas.

V.2.2.- Algoritmos de Estado de Enlace.

Este tipo de algoritmos propagan el estatus de la topología física actual de la Interred a cada enrutador. Los mensajes de actualización consisten en pares de (enlace, metric), donde el enlace identifica a un par de enrutadores adyacentes, y el metric indica el costo de utilizar ese enlace. El valor del metric puede ser fijo o dar una estimación dinámica de su carga, transmisión, retardo, etc. (Narten, 1989).

Periódicamente, un enrutador que controla los broadcasts de enlaces actualiza los mensajes a todos los enrutadores que participen en la Interred. De la base de datos de costos de enlaces, cada enrutador construye un árbol de caminos más cortos que lleguen a cada destino.

Debido a que los mensajes son realizados por broadcasts a cada enrutador, éstos construyen sus tablas de enrutamiento de la misma información, reduciendo la probabilidad de que se formen ciclos. Por lo tanto, este tipo de protocolos es menos vulnerable a la formación de ciclos que los de vector-distancia.

Pero, en la práctica esto no es totalmente verdadero ya que un broadcast puede propagar lentamente causando que los enrutadores reciban información de actualización en diferentes tiempos.

Además, el costo de envío de mensajes es significativo y no es apropiado a una Interred de miles de redes y enrutadores.

V.3.- Estado Actual e Implementaciones de Algoritmos de enrutamiento dinámico.

Esta sección establece la validez de utilizar un algoritmo de vector-distancia a preferencia de los de estado de enlace, realizando un análisis de las implementaciones actuales de ellos. Esta sección establece el porqué de la utilización del protocolo de vector-distancia RIP en este trabajo de tesis.

Como se ha mencionado en capítulos anteriores, TCP/IP contiene los elementos necesarios para un enrutamiento dinámico no sólo a nivel de capa física y de enlace como en el caso de Novell, sino además cubre las capas superiores permitiendo que un protocolo como RIP permita enrutamiento a nivel RCA. Por esto último, se analizan las implementaciones realizadas sobre esa arquitectura.

En el capítulo III, se describió un sistema autónomo y los protocolos asociados al manejo de éstos: EGP's e IGP's en conjunto con el protocolo IP ya que son necesarios para la comunicación en una Interred TCP/IP. Los EGP's intercambian información entre sistemas autónomos y los últimos dentro de un sistema autónomo.

Los EGP's son protocolos de alcanzabilidad (Mills, 1984) más que de enrutamiento porque la especificación de EGP no incluye el concepto de metric y sus implementaciones son y están limitadas en la Interred actual TCP/IP.

Por otro lado, los IGP son algoritmos de enrutamiento dinámico ó distribuido que engloban las siguientes características:

- * Algoritmo para asignar una longitud a cada camino en la Interred (hop).
- * Algoritmo para calcular el camino más corto a un destino basado en hop.
- * Algoritmo de intercambio de información entre enrutadores.

De los más utilizados son GGP (Gateway to Gateway Protocol), SPF (Shortest Path First), RIP (Routing Information Protocol) y últimamente OSPF (Open Shortest Path First).

Los dos últimos representan la mayoría en utilización, los demás tienden a ser obsoletos. Existe la tendencia a la utilización del algoritmo contenido en OSPF, a preferencia de RIP.

Las siguientes características son consideradas como necesarias para que un algoritmo de enrutamiento opere exitosamente en un enrutador (Braden, 1987):

1. El algoritmo debe saber la falla ó recuperación de un enlace u otro enrutador y cambiar caminos apropiados.
2. El algoritmo debe suprimir ciclos de enrutamiento entre enrutadores vecinos, además debe evitar ciclos que se formen en enrutadores que no sean vecinos.
3. El control de tráfico necesario para operar el algoritmo no debe degradar ó interrumpir la operación normal de la red. Cambios en una red local no debe causar disturbios en otras redes.
4. El tamaño de la tabla de enrutamiento no debe exceder de una constante independientemente de la topología de la red, por el número de nodos ó por el promedio de enlaces incidentes.
5. Debe incluir rutas por omisión y usarlas con cuidado ya que pueden llevar a ciclos ó caminos múltiples a un sólo destino.

Con estos conceptos, la siguiente sección analiza los problemas que se presentan en el funcionamiento de enrutamiento en una Interred así como sus soluciones para posteriormente enfatizar la implementación genérica de estas soluciones en el algoritmo de enrutamiento contenido en RIP, de ahí demostrando su amplia aceptación y utilización en este trabajo de tesis.

RIP reúne los requisitos principales de un buen algoritmo de enrutamiento probado y justificado en miles de implementaciones actuales alrededor del mundo. En la actualidad, el protocolo OSPF ha tendido a establecerse como estándar IGP, pero ha demostrado ser ineficiente en redes complejas o de gran tamaño debido al desacuerdo de una implementación estándar, lo cual realiza cálculos diferentes en su propagación dentro de un sistema autónomo; es decir, la implementación de RIP ha sido estándar gracias a la implementación de la universidad de Berkeley y su gran uso.

V.4.- Problemas presentados en el enrutamiento de una Interred.

V.4.1.- Cambios de topología de red.

Se supone que la topología de una red es fija, pero en la práctica las líneas y enrutadores fallan y vuelven a operar. Para ajustar estos cambios, el algoritmo debe modificarse. Este calcula mínimos entre enrutadores vecinos inmediatos. Si la topología cambia, los vecinos también, por lo tanto, la siguiente vez que se realice el cálculo, el cambio será reflejado. Si el enrutador falla, no tiene manera de avisar a sus vecinos acerca de un cambio ya que si esto pasa los cambios en la topología no se ven reflejados.

Para manejar estos problemas, el algoritmo debe proveer alguna manera de expirar rutas en un tiempo dado. En RIP, las actualizaciones se realizan cada 30 segundos. Si una ruta N utiliza el enrutador G y G no responde en 180 segundos, la ruta N se marca como inválida hasta que aparezca una nueva ruta N'. Esta ruta se marca con un valor de 16 en el metric que significa infinito.

V.4.2.- Inestabilidades. Conteo a infinito.

Existen casos cuando el tiempo no garantiza que será lo suficientemente pequeño para la convergencia de rutas, o sea, cuando las medidas hacia las redes destino se vuelven inaccesibles. Como ejemplo se ilustra la figura 24.

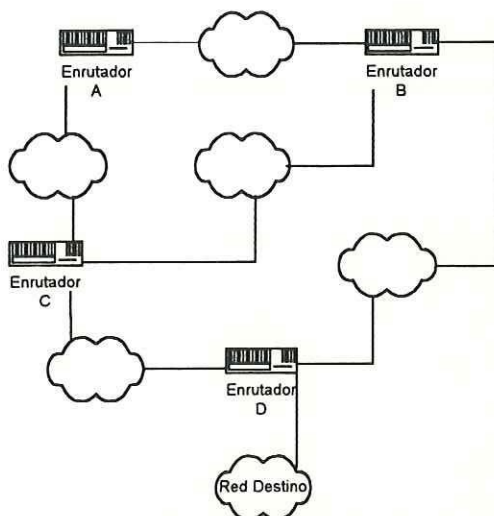


Figura 24.- Escenario de Interred ejemplo para inestabilidades de enrutamiento.

En este escenario se supone que la red destino se vuelve inaccesible y los enrutadores vecinos D y B expiran sus rutas a ellas (medida = 16). Los enrutadores a un hop de ellos (A y C) tendrán medidas de 17 para la red destino y los enrutadores a 2 hops las tendrán a 18 etc. Estos valores son mayores que 16 y se ve por lo tanto que se converge a este número. Pero esto depende del valor de infinito, si éste es muy grande, la convergencia es mayor también. Este fenómeno se conoce como conteo a infinito y es ilustrado en la tabla I.

Tabla I.- Información de rutas para cada red en ejemplo de figura 24.

	Tiempo 1	Tiempo 2	Tiempo 3		Tiempo 18
Red					
D:	dc - m=1	dc - m=1	dc - m=1	...	dc - m=1
B:	-----	vía C - m=4	vía C - m=5	...	vía C - m=18
C:	vía C - m=4	vía A - m=4	vía A - m=4	...	vía A - m=18
A:	vía B - m=3	vía C - m=4	vía C - m=4	...	vía C - m=18

dc - red directamente conectada

m - metric asociado a ruta

B expira ruta, y A,C piensan que pueden llegar a D vía B, en el tiempo 2 siguen enviando actualizaciones con medida de 3. Para el tiempo 3, B pensará que todavía puede llegar a D vía A ó C. Cada uno piensa que se puede llegar a través del otro hasta que se llega a 16 ó más en la medida lo cual tarda un tiempo, mientras creando un ciclo entre ellos. Por lo tanto si una red se vuelve inaccesible, las medidas convergen lentamente hasta llegar al infinito.

V.4.3.- Soluciones a problemas de enrutamiento.

Los problemas de enrutamiento se deben principalmente, siguiendo la figura 24, a que A y C están mutuamente decepcionándose diciéndose cada uno que puede llegar a D vía el otro. El esquema de solución ante este problema omite rutas aprendidas de un vecino en actualizaciones enviadas a ese vecino. Este proceso se le conoce como *Split Horizon* (Hedrick, 1988). La ventaja es que mejora el comportamiento dinámico, si cambia la topología, el mencionar rutas que no deben pasar por el enrutador así como también aquellas que sí aceleran la convergencia.

Para la aceleración de la convergencia se utiliza un proceso conocido por su término en inglés *triggered updates* (actualizaciones disparadas) (Hedrick, 1988).

Cuando un enrutador cambia la medida de una ruta, éste manda mensajes de actualización inmediatamente, aún cuando no es tiempo de mandar la actualización periódica de cada 30 segundos. Así, los demás enrutadores reciben el nuevo cambio antes del conteo a 16.

V.5.- Selección del protocolo de enrutamiento.

Para este trabajo de tesis, se encontró el protocolo RIP muy adecuado, ya que reúne las características necesarias para un enrutamiento a nivel de RCA, incluyendo en su algoritmo de vector-distancia las soluciones descritas en la sección anterior. Además, por tener un vasto uso de él a nivel internacional, hace de su elección una opción muy adecuada.

Además, para el trabajo de esta tesis, se necesitará la modificación del algoritmo contenido en RIP para la implementación de la solución propuesta especificada en el siguiente capítulo. El protocolo RIP contiene la modularidad y flexibilidad necesaria para poder incorporar una nueva metodología en su algoritmo de enrutamiento para permitir su uso en el ambiente de RCA - RLD. Tales aspectos útiles en cuanto a poder modificar de una manera adecuada a RIP para este trabajo incluyen:

- * Independencia de otras capas de arquitectura. Si se realiza una modificación a RIP, no se altera el funcionamiento de ninguna de las capas de red de ambas arquitecturas de red TCP/IP y Novell.
- * Manejo de estructuras flexible. El manejo de las tablas de enrutamiento en un lenguaje de programación como lo es C, resulta de una manera fácil, rápida y eficiente para su implementación, en conjunto con metodologías aplicadas para la eficiencia de búsqueda de rutas en dicha tabla.

- * Disponibilidad de documentación bastante amplia debido a su gran número de implementaciones. Esto hace que la implementación de la solución propuesta en RIP sea igualmente aceptada y confiable.
- * Por último, sus características de enrutamiento dinámico genérico pueden ser aprovechadas de una manera sencilla. Esto permite que la aplicación sea implementada con mínimo requerimiento de hardware y software de desarrollo (PC XT y Compilador de lenguaje C Turbo C versión 2.0).

Este capítulo complementó al enrutamiento descrito en el capítulo III, en cuanto a la validez del protocolo RIP como algoritmo de enrutamiento principal en la solución propuesta, así como también los problemas y soluciones que se presentan en la puesta en marcha de la implementación de este protocolo con la solución. Como se puede determinar, el enrutamiento RIP - IP es más adecuado que el enrutamiento IPX por todos los aspectos descritos en los capítulos anteriores.

En el siguiente capítulo se describe la solución propuesta en este trabajo, ya con los conceptos manejados anteriormente en este y otros capítulos.

VI.- IMPLEMENTACION DE ALGORITMOS DE ENRUTAMIENTO

VI.1.- Introducción.

En los capítulos anteriores se realizó un análisis de operabilidad del enrutamiento del protocolo RIP, sus funciones, alcances y limitaciones, con la finalidad de acotar su campo de trabajo y posteriormente poder realizar un desarrollo formal del mismo.

En este capítulo, se desarrollarán los mecanismos necesarios para la implementación del protocolo RIP para validar su desempeño de enrutamiento existente en RCA. Además, se propone una solución a la problemática de interconectividad RLD - RCA, que se ve contemplada dentro de la implementación del protocolo RIP mediante la programación. La validación del protocolo de enrutamiento RIP será a través de programación en lenguaje C y Ensamblador, validado en un ambiente real (Red CICESE) y posteriormente en un ambiente de simulación a través del paquete COMNET III, con estos desarrollos será posible obtener el objetivo planeado originalmente.

VI.2.- Antecedentes.

VI.2.1.-Tunel IP-IPX.

Las redes Novell fueron creadas como una tecnología de RLD, por ello, en sus primeras versiones no proveían medios de interconectividad a RCA. Con el advenimiento de nuevas tecnologías y la difusión de la Inter-red TCP/IP, Novell y otras compañías desarrollaron esquemas para la interconectividad entre Novell y TCP/IP.

De los anteriores esquemas, el que es mas utilizado es el propuesto por Novell, este esquema es conocido como TUNEL IP-IPX y a partir de la versión 3.11, forma parte del Sistema Operativo de Red ofrecido por Novell.

El protocolo IPX es el protocolo de red usado por la Suite de protocolos de Novell. A continuación se describe el método empleado para encapsular datagramas IPX dentro de paquetes IP. Para la formación de los paquetes que habrán de atravesar la Inter-red TCP/IP se emplea el protocolo UDP en la capa de Transporte.

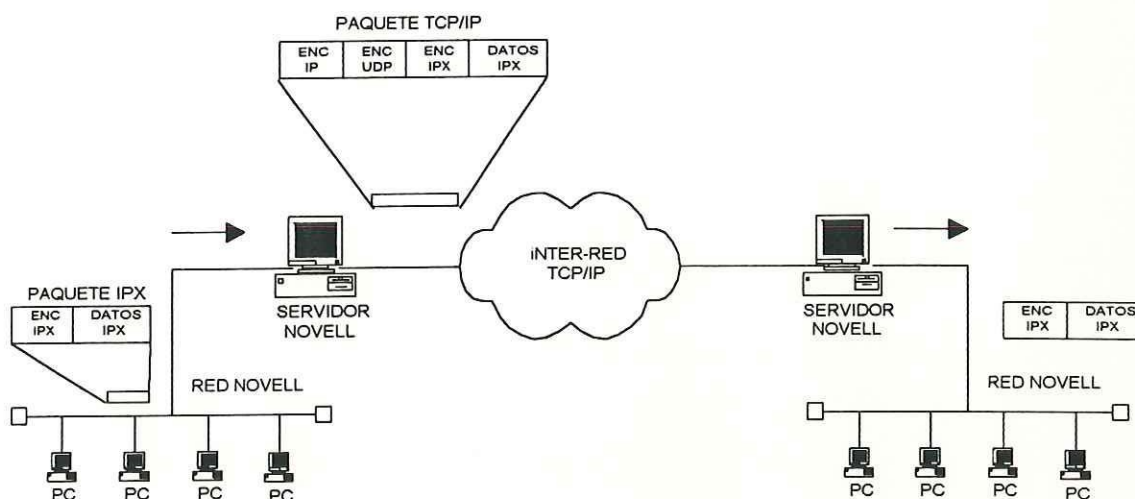


Figura 25.- Tunel IP/IPX.

En la figura 25 se observa el proceso de tunelaje desarrollado por los servidores novell, estos tienen la capacidad de encaminar paquetes de una red Novell hacia una red TCP/IP, teniendo como destino ya sea un nodo IP o un servidor Novell. En el servidor destino se desarrolla la función contraria, adecuando los paquetes para su entrega a la red Novell.

VI.2.2.- Formato de paquetes.

Como se observa en la figura 26, cada uno de los datagramas IP es transportado en la porción de datos de un paquete UDP (Cap III). Los campos IP y UDP son configurados normalmente, a los puertos fuente y destino se les asigna el valor 520. Como con cualquier aplicación UDP, la parte transmisora tiene la opción de evitar el sobreprocesamiento, configurando a CERO el campo de "chequeo" del encabezado UDP. Dado que las implementaciones IPX nunca usan el campo de prueba, es recomendable el uso del chequeo UDP para encapsulación de IPX.

Encabezado IP (20 ó más octetos)	Encabezado UDP (8 Octetos)	Encabezado IPX (30 Octetos)	Area de Datos IPX (546 Octetos)
---------------------------------------	---------------------------------	----------------------------------	--------------------------------------

Figura. 26.- Paquete IPX transportado como datos en un paquete UDP.

VI.2.3.- Desventajas de un esquema de interconectividad como el tunel IP-IPX.

El tunel IP-IPX es un esquema de interconectividad que aunque es ampliamente utilizado, tiene desventajas que limitan la operación y funcionamiento de las redes Novell. Este y otros esquemas presentan las siguientes desventajas para cuestiones de enrutamiento en una RCA:

1).- La asignación de Números de Red.

La asignación de números de Red es completamente arbitraria, es decir, cuando el administrador de una red Novell enciende el servidor asigna una dirección de Red, sin tener conocimiento de las direcciones de otras Redes. De esta manera pueden existir direcciones de red duplicados. Lo anterior, provoca problemas ya que Novell no permite la coexistencia de segmentos con el mismo número de Red.

2).- La sobrecarga en el servidor.

Los desarrollos a enrutadores Novell son muy pobres, es decir, la programación que permite a los servidores desarrollar el tunel IP-IPX no es adecuada, de manera que un enrutador Novell tiene sólo el 20% de la eficiencia de un enrutador IP. Además si sólo existe un servidor en un segmento (caso típico) y éste se adecúa como enrutador, se desempeño se ve afectado hasta en un 40%.

3).- Modelo de comunicación Cliente-Servidor.

Los nodos de una Red Novell dependen completamente del servidor, por esta razón, en una Red Novell, los servidores son los únicos que tienen la capacidad de compartir recursos con otros servidores, los nodos no pueden compartir sus recursos (disco duro, aplicaciones, etc) con otros nodos.

4).- Sobreprocesamiento.

En la figura 26, se puede observar el sobreprocesamiento de los paquetes. Los encabezados que se agregan al paquete IPX provocan que el caudal eficaz disminuya considerablemente.

VI.2.4.- Diferencias entre Novell y TCP/IP.

El estudio desarrollado durante el trabajo de tesis permite identificar diferencias muy significativas entre las tecnologías en cuestión, estas diferencias se citan en la tabla II.

Tabla II. Diferencias significativas entre Novell y TCP/IP.

Novell	TCP/IP
Modelo de Comunicación	
Sólo RLD, por su modelo de Cliente - servidor.	Ambiente de RLD y RCA. Cada nodo tiene capacidad de enrutar.
Direccionamiento	
Formato de direcciones es: 00000034:080635590000:4003	Direcciones de la forma: hjimenez@cicese.mx
Se asignan direcciones de red sólo a servidores. Las direcciones de nodo se asignan dinámicamente	Cada nodo en la Interred tiene su propia dirección IP.
La asignación de números de red es arbitraria.	La asignación de números es realizada por un organismo centralizado llamado NIC (Network Information Center)
Tiene problemas de direccionamiento hacia subredes.	Provee mecanismos como el de enmascaramiento de subredes.
Enrutamiento	
Ubicado sólo en los servidores (30 a 40 % de pérdida de la eficiencia)	Localizado en cada uno de los nodos de la RCA.
Las tablas de enrutamiento manejan sólo direcciones IPX.	Manejo de "familia de direcciones" en RIP para dar soporte a otras direcciones.
Tamaño de paquete y fragmentación	
Tamaño máximo de 576 Octetos en datos y encabezado.	Tamaño máximo es variable.
Fragmentación se realiza en el nodo.	Fragmentación se realiza en los enrutadores

VI.3.-Solución Propuesta.

De acuerdo a los análisis realizados en los capítulos anteriores, la solución propuesta hace énfasis en los siguientes problemas identificados por las diferencias entre los ambientes de red:

- * Mapeo de direcciones.
- * Sobreprocesamiento de red.
- * Manejo de tablas de enrutamiento genéricas.

Puntualizando estos tres aspectos se propone un esquema como el de la figura 27 para la implementación de este trabajo. Aunque este esquema es específico para la subred de física aplicada de la red CICESE, éste se puede migrar fácilmente a cualquier otra subred o red.

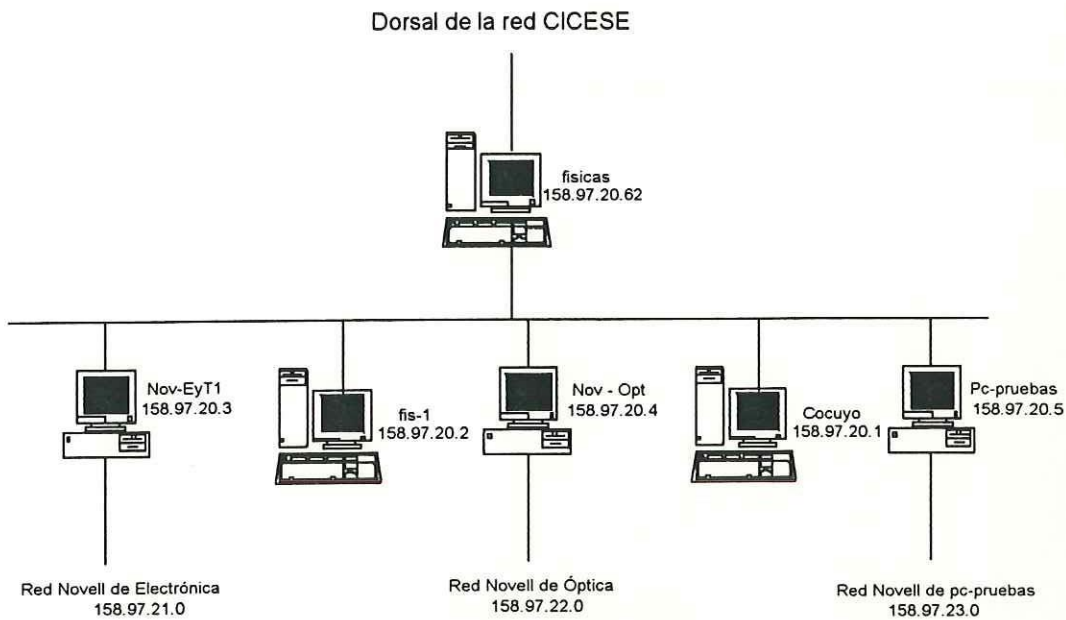


Figura 27.- Topología empleada para la implementación de algoritmos de enrutamiento.

Esta topología está constituida por una RLD Ethernet, en la cual se tienen en existencia 5 nodos, uno de ellos (físicas) siendo el enrutador hacia la dorsal de la red CICESE (fibra óptica).

La solución propuesta sugiere que el direccionamiento principal a utilizar sea el de IP, ya que contiene características y mecanismos para RCA.

Por lo tanto, se propone que a cada usuario en una RLD Novell se le asigne una dirección IP, así la tabla de enrutamiento será la misma para las dos tecnologías de red. De acuerdo a ésto, el direccionamiento de la RLD Novell se adapta al direccionamiento IP mediante el mapeo de dirección IP a nombre de usuario. Este mapeo se maneja únicamente por el nodo enrutador (pc-pruebas de la fig. 27) mediante una tabla de mapeo "nombre de usuario" - "dirección IP". La tabla de mapeo se muestra en la figura 28.

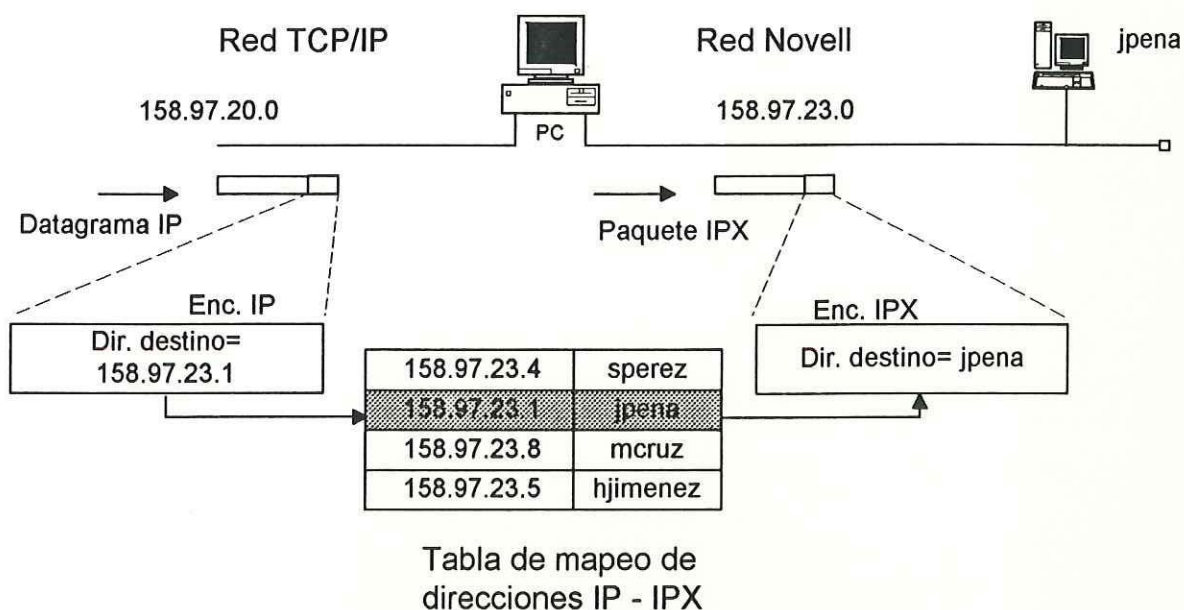


Figura 28.- Mapeo desarrollado por el enrutador experimental pc-pruebas.

De acuerdo con la figura 28, si se envía un paquete desde una red TCP/IP hacia una subred Novell, con la dirección destino 158.97.23.1 previamente asignada, el datagrama es recibido por pc-pruebas, el proceso de revisión de direcciones con ayuda de un identificador interno de la tabla de enrutamiento local de la PC, identificará la dirección IP como un destino hacia la red Novell y lo cambiará al formato usado por Novell ya con el nombre de usuario final "jpena".

Una vez realizado este mapeo existosamente, se encuentra dinámicamente al usuario "jpena" para el envío de un paquete IPX puro. Esto elimina la sobrecarga al servidor, además de eliminar información adicional al paquete; es decir, no viaja información TCP/IP en la red Novell y viceversa aislando el tráfico en ambos lados de las redes.

Para el enrutamiento hacia la RCA (TCP/IP) desde una RLD Novell, el paquete IPX generado en cualquier estación de la RLD contendrá la dirección destino IP en su área de datos, la cual será extraída por el enrutador pc-pruebas y enviado por la RCA por el proceso IP. Este funcionamiento lo muestra la figura 29.

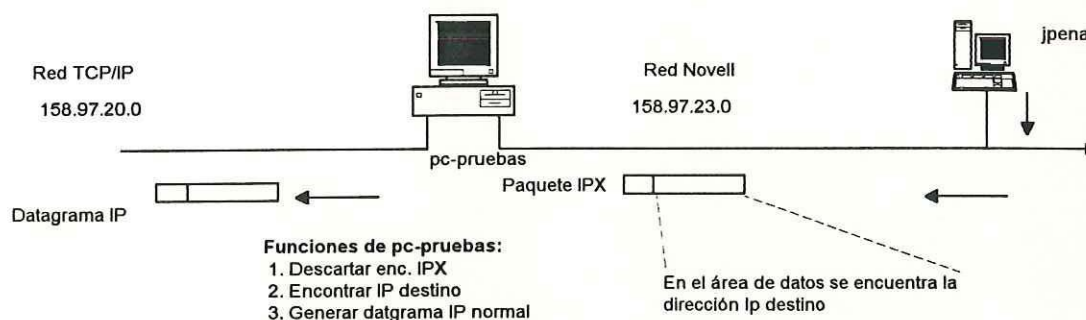


Figura 29.-Enrutamiento de RLD (Novell) hacia RCA (TCP/IP).

En la siguiente sección se presenta la especificación formal de todos estos procesos mediante una implementación realizada a través de la programación.

VI.3.1.- Implementación de la propuesta en un ambiente real.

El algoritmo de enrutamiento RIP en conjunto con la solución propuesta se implementó en la subred de física aplicada de la red CICESE.

Dicha subred maneja únicamente tráfico IP, por lo tanto, para implementar a RIP, fué necesario desarrollar protocolos adicionales como UDP, IP, ICMP y ARP. Como se observa en la figura 27, se adecuó una PC AT (80286) (llamada pc-pruebas) con programación para desarrollar funciones de enrutamiento hacia la red Novell (asignado el número de red 158.97.23.0). A dicha PC se le asignó un número IP (158.97.20.5) y se conectó al segmento TCP/IP. Con este esquema, pc-pruebas tiene la capacidad de escuchar tráfico de ambos segmentos.

En la figura 30, se muestra internamente a pc-pruebas. Como se puede apreciar, el programa de aplicación proporciona la interface entre las dos tecnologías de red, por esta razón se hace necesaria una explicación más formal del mismo.



Figura 30.- Organización interna del nodo pc-pruebas.

VI.4.- Desarrollo de la aplicación de la solución propuesta.

La aplicación consiste en módulos de programas realizados en lenguaje C y ensamblador, los cuales realizan funciones de enrutamiento de los protocolos IP y RIP, además de incluir los protocolos adicionales UDP, ICMP y ARP, cubriendo así el manejo de enrutamiento de RCA. Además, se tienen módulos adicionales que se encargan del manejo de enrutamiento Novell (protocolo IPX). Los módulos contienen rutinas que describen el funcionamiento de los protocolos de las dos arquitecturas de red, éstos están lógicamente divididos en procesos de entrada y salida de cada uno de ellos.

Todos los procesos de los protocolos implementados son manejados por una rutina principal (main()) a cuyo funcionamiento se le denominó "sistema de canalización de paquetes". Este organiza a los protocolos de acuerdo a la información contenida en los paquetes que se manejen y controla todas los procesos de entrada/salida de los protocolos. La relación de éstos últimos están regidos por el sistema de canalización que contiene varios mecanismos específicos de la implementación como lo son:

- * Sistema de canalización de paquetes mismo, implementado como una estructura de cola.
- * Información adicional a cada paquete para su control interno de la aplicación, denominado "pseudo-encabezado".
- * Procesamiento de todos los temporizadores asociados a rutas de las tablas de enrutamiento y demás protocolos, implementados como una estructura de cola como lista ligada doble.

Para el manejo de la capa física de TCP/IP, se utilizó un manejador de tarjeta de red (packet driver) que provee una interface de programación a través de funciones activada por interrupciones. Esta interface es dictada por la especificación PDS de la compañía FTP (ver Apéndice A). Para el manejo de la capa física por parte de Novell, se utilizó el programa genérico IPX.COM y NETx.COM, ya que éstos manejan directamente la tarjeta de red, aprovechando solamente sus servicios también activados por interrupciones. De cualquier manera, para la recepción y envío de paquetes por ambas arquitecturas de red se utiliza un manejador de paquetes.

VI.4.1.- Sistema de canalización de paquetes. Rutina principal.

El sistema de canalización de paquetes (SCP) depende principalmente de la capa física para el procesamiento de paquetes. Al ejecutarse una interrupción de hardware se activa una rutina (*recibe_paquete()* para TCP/IP y *llego_ipx()* para IPX) que es la que proporciona servicio al paquete recién obtenido, básicamente, la función de esta rutina es de proveer un área de almacenamiento (buffer) en memoria para guardar ahí el paquete. Esta área es la Cola Principal de Paquetes; es decir, al recibir un paquete, ya sea por IPX ó por TCP/IP, al paquete se le agrega inmediatamente el pseudo-encabezado interno que identifica la arquitectura de red destino y se ubica espacio en memoria para él.

El paquete se guarda internamente según un formato conocido como mbuf (Comer, *et al*, 1991), el cual permite el manejo de la información de una manera muy eficiente. Este proceso se muestra en la figura 31.

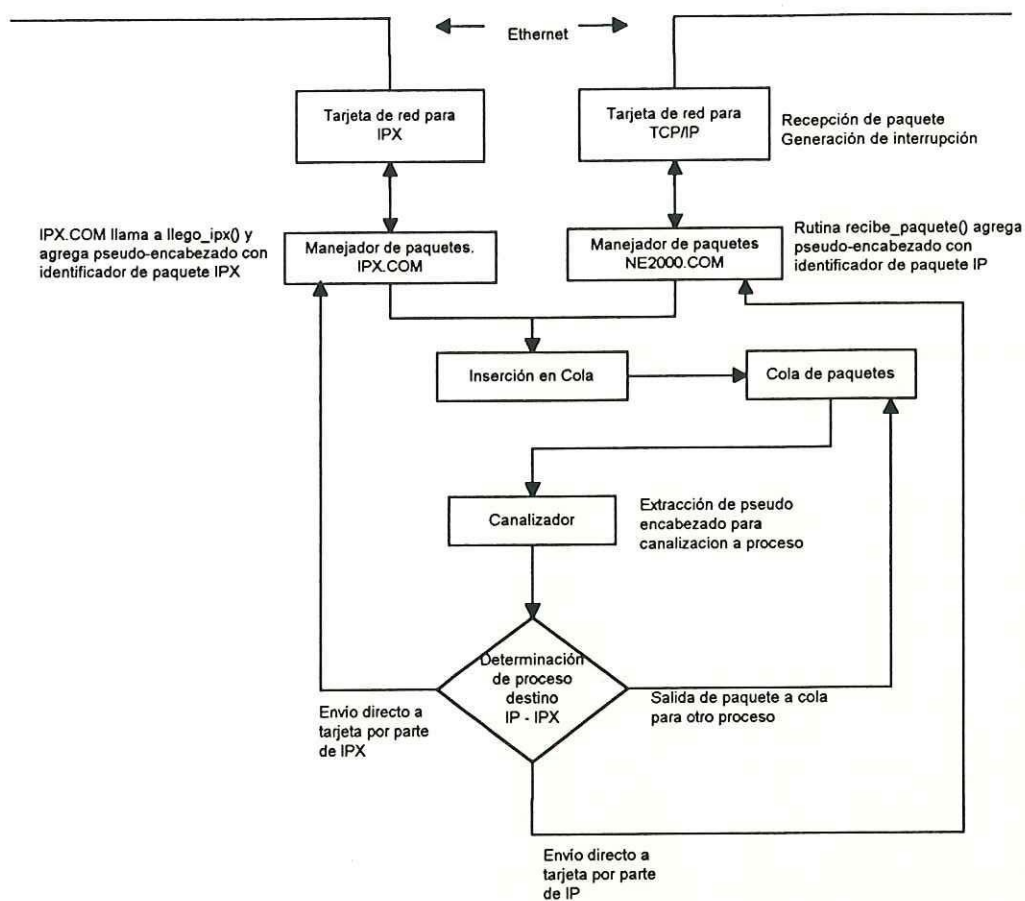


Figura 31.- Funcionamiento del Sistema de Canalización de Paquetes de la aplicación.

Además, el SCP maneja los temporizadores de algunos procesos. El SCP procesa todos los paquetes mediante su extracción de la estructura de cola, así como su extracción del pseudo-encabezado para la determinación del proceso destino (TCP/IP ó IPX).

VI.4.2.- Procesos de entrada y salida generales.

Todos los protocolos tiene un proceso de entrada y otro de salida, los más importantes para este trabajo son los del protocolo IP y RIP. El SCP utiliza el pseudo-encabezado agregado por la rutina de recepción para identificar cuál proceso es el que va a intervenir en el procesamiento del paquete que se esté manejando. Los procesos de entrada que controla el SCP son el de ARP, IP e IPX. El flujo de procesamiento de un paquete recibido lo muestra la figura 32.

El proceso IP revisa la dirección IP contenida en el encabezado IP del paquete recibido, si es la dirección local (de pc-pruebas) el proceso RIP actualiza la tabla de enrutamiento a través de los procesos UDP y RIP (el proceso UDP llama a su vez al proceso RIP). Si no es la dirección local, se realiza el enrutamiento. Se efectúa una búsqueda en la tabla de enrutamiento para ver si la dirección destino es una red IP ó IPX. En el último caso, se realiza el mapeo de direcciones especificado anteriormente y se invoca al proceso de salida IPX para el envío de un datagrama IPX puro. De otra manera, se llama al proceso de salida IP normal.

El proceso de entrada IPX, extrae la dirección IP del área de datos del datagrama IPX recibido y forma un encabezado apropiado para el envío del nuevo datagrama IP a través de la RCA invocando al proceso de salida IP normal.

El proceso ARP sólo actualiza su tabla de mapeo de direcciones IP - direcciones hardware y procesa peticiones de otros nodos para la resolución de direcciones.

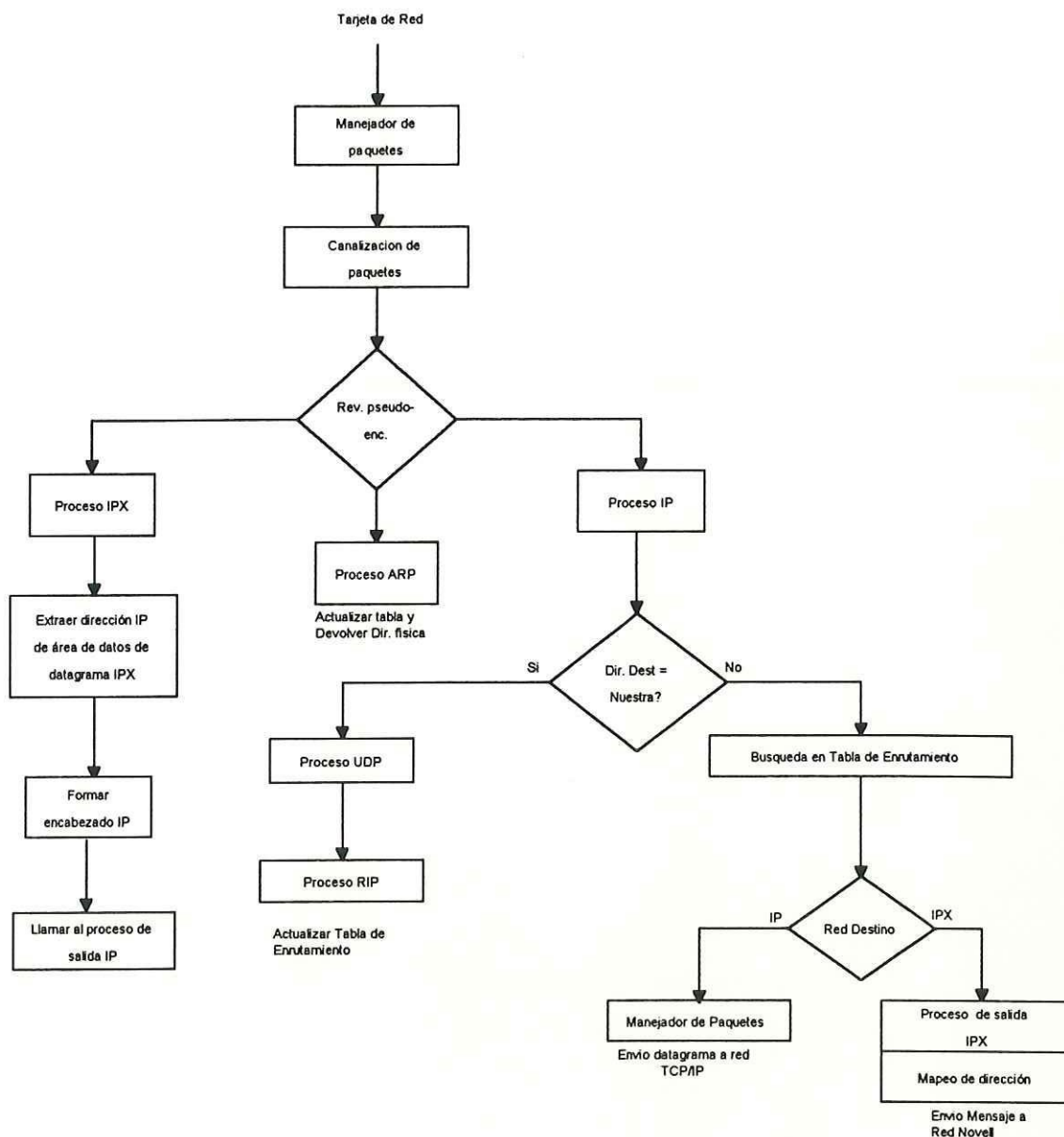


Figura 32.- Proceso General de Entrada de un paquete al SCP.

Básicamente, el proceso de salida más importante es el de la actualización periódica de la tabla de enrutamiento de pc-pruebas hacia la RCA.

Este proceso viene siendo el de salida de RIP. Existen salidas de los otros protocolos como en el caso de un mensaje de error, éste siendo el proceso de salida del protocolo ICMP. El protocolo RIP utiliza al proceso de salida UDP a su vez para el envío de la tabla de enrutamiento. Otras salidas son provocadas por una actualización disparada de una ó más rutas. Todo es realizado a través del SCP como en el caso de las entradas. Este procedimiento se muestra en la figura 33.

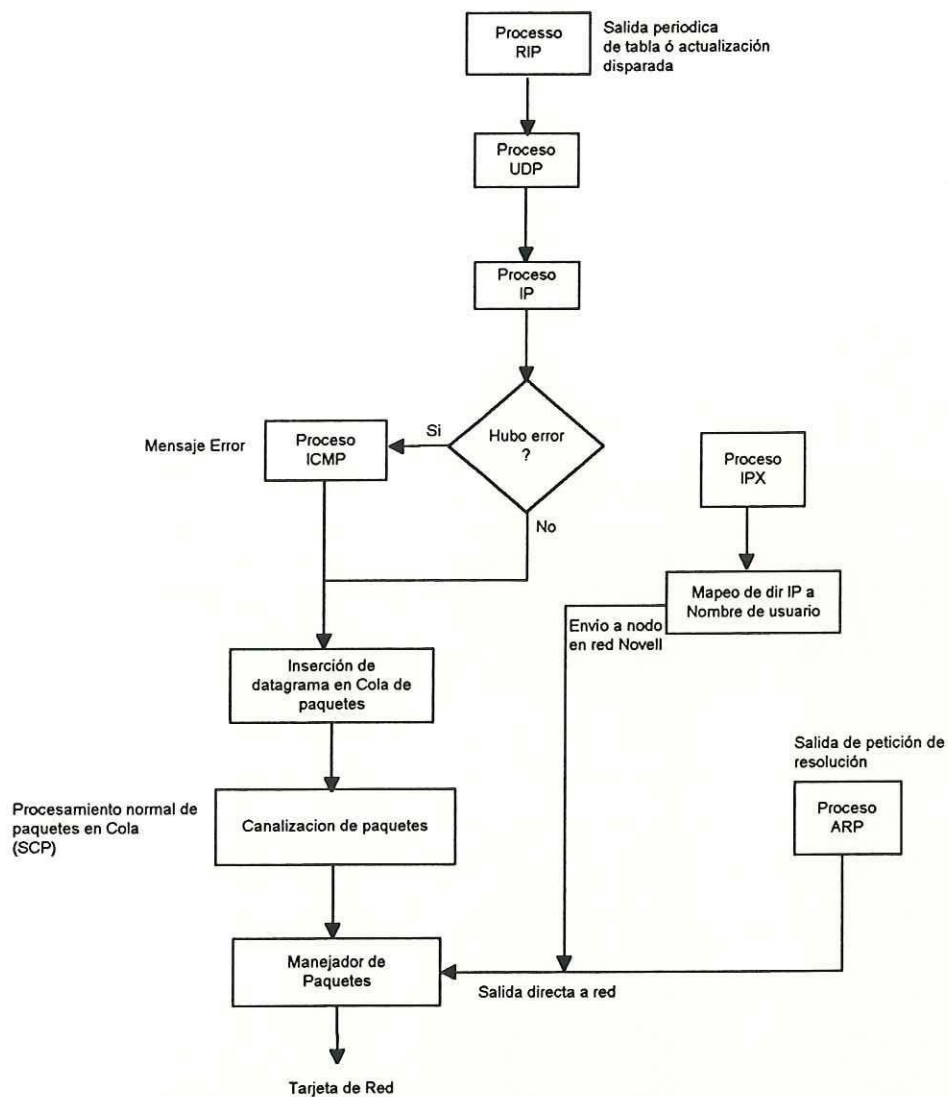


Figura 33.- Proceso General de Salida de la aplicación.

En capítulos anteriores se mostró en diagramas el procesamiento que se lleva a cabo en el proceso IP y RIP, siendo éstos los más importantes.

VI.4.3.-Manejo de temporizadores.

Además de funcionar como canalizador de paquetes, la rutina principal de la aplicación maneja los temporizadores de algunos procesos. Los temporizadores se mantienen en una estructura de cola implementada como una lista doble ligada, en la cual cada tick de reloj decrementa la cuenta de tiempo del primer elemento de la estructura. Finalmente, los temporizadores que se expiren se mantienen en una lista ligada sencilla para ser procesados por su respectivo proceso de manejo de expiración de tiempo. Este proceso se muestra en la figura 34.

Para el manejo adecuado de temporizadores, se utilizó el reloj del sistema de la PC al cual se tiene acceso a través de la instalación de una rutina de servicio de la interrupción 0x1C. Esta rutina se llama *reloj()* y lo único que establece es un contador llamado *Tick* que es incrementado cada vez que el reloj del sistema realiza un "tick" interno. De esta manera, el SCP se fija si el contador *Tick* se incrementó, si esto sucede, se decrementa éste mismo y se procesan los temporizadores para ese tick de reloj ocurrido.

Al finalizar la ejecución de la aplicación en la PC, se debe volver a reestablecer el vector de interrupción 0x1C, de otra manera el funcionamiento de ésta será afectado y no se podrá utilizar hasta reinicializar (reboot) la PC por completo.

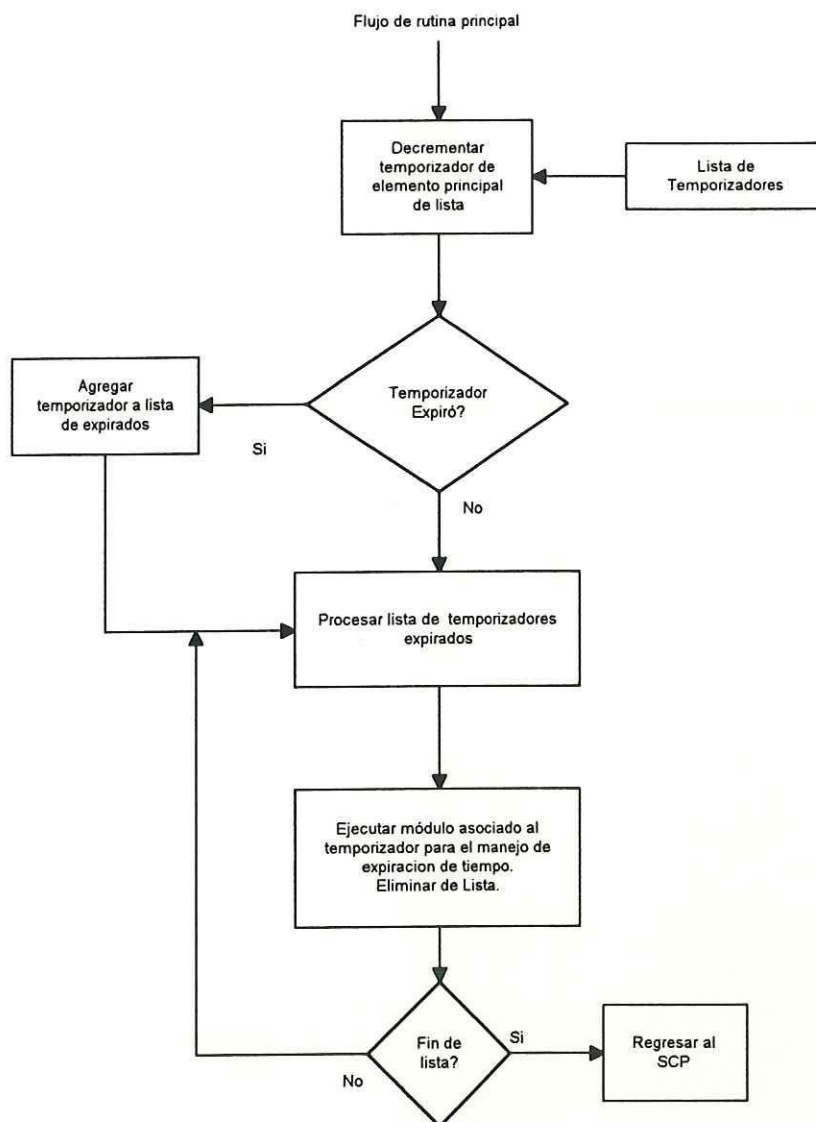


Figura 34.- Procesamiento de temporizadores realizado por la rutina principal.

Los módulos de implementación contienen documentación en el código fuente incluídos en el disco que acompaña a este trabajo. La distribución de los módulos en el disco y otros detalles se encuentran en el apéndice B.

VI.5.- Análisis de los resultados obtenidos a través de la implementación en ambiente real.

Con la implementación desarrollada se obtuvo el objetivo planteado originalmente, es decir, a través de programación se habilita una pc como enrutador, proporcionándole para ello los algoritmos necesarios para la creación y actualización de una **Tabla de Enrutamiento**, que proporciona mecanismos para que la red Novell logre enrutar paquetes hacia la Red TCP/IP y viceversa. El programa elaborado, tiene inicialmente una Tabla de enrutamiento que consta únicamente de dos rutas, la cual se muestra en la Tabla III. Como puede observarse, en las diferentes columnas de dicha tabla, existen parámetros asociados con los mecanismos de enrutamiento. Cada uno de los renglones corresponde a una ruta, las columnas de izquierda a derecha; **Destino**: indica la dirección IP de la red destino. Para el caso particular de la implementación, a la subred Novell, se le asignó el número 158.97.23.0., como es una red que no forma parte del sistema, se configura para que su enrutamiento se realice a través del enrutador experimental (pc-pruebas). **Longitud**: lo que indica este parámetro es el número de bits de la dirección IP que son significativos para los mecanismos de enrutamiento (su función es semejante a las máscaras). **Interface**: indica el nombre de la interface a través de la que la pc se comunica con la red destino. **Gateway**: indica el dispositivo de salida (enrutador ó gateway) a través del cual es posible alcanzar la red destino. **Metric**: es un valor numerico que refleja el costo de usar la ruta. **Reloj**: Es empleado para eliminar rutas que se consideren obsoletas. **Uso**: es el número de paquetes que han sido encaminados a través de la ruta.

Todos estos atributos que han sido implementados en el enrutador experimental son los que se obtienen en la operación normal de enrutador comercial; es decir, la implementación experimental ha buscado una total compatibilidad con sistemas comerciales, buscando además cumplir con las expectativas de desarrollo a corto plazo.

Tabla. III.- Tabla inicial de pc-pruebas.

Red destino	Long	Interface	Gateway	Metric	Reloj	uso
158.97.23.0	24	irq9	158.97.20.5	1	0	0
default	0	irq9	158.97.20.62	3	0	0

Posteriormente, con la ayuda de otro programa de aplicación que permite observar el tráfico TCP/IP, se logró observar paso a paso la generación de la Tabla de Enrutamiento de pc-pruebas. El primer paquete recibido, se muestra a continuación en el mismo formato que lo despliega la aplicación.

```
Fri May 06 01:58:11 1994 - irq9 recv:
Ether: len 66 00:00:c0:14:65:2c->ff:ff:ff:ff:ff:ff type IP
IP: len 52 158.97.20.3->255.255.255.255 ihl 20 ttl 128 prot UDP
UDP: 520->520 len 32
RIP: RESPONSE vers 1 entries 1:
158.97.21.0 1
```

Como se muestra en este paquete, el enrutador 158.97.20.3, envía una ruta en un paquete broadcast, notificando que a través de el, es posible alcanzar la red 158.97.21.0 con un costo unitario (metric = 1). Pc-pruebas, captura el paquete y agrega entonces la nueva ruta a su tabla de enrutamiento, incrementando en una unidad el metric de la ruta. La tabla se actualiza de la siguiente manera.

Tabla IV.- Proceso de alta de una nueva ruta.

Red destino	Long	Interface	Gateway	Metric	Reloj	uso
158.97.21.0	24	irq9	158.97.20.3	2	113	0
158.97.23.0	24	irq9	158.97.20.5	1	0	0
default	0	irq9	158.97.20.62	3	0	0

De manera similar, son recibidas las rutas de los demás enrutadores en el segmento TCP/IP. A continuación se muestran los paquetes recibidos por los distintos enrutadores.

Fri May 06 01:58:17 1994 - irq9 recv:
Ether: len 66 00:00:c0:f4:89:28->ff:ff:ff:ff:ff:ff type IP
IP: len 52 158.97.20.4->255.255.255.255 ihl 20 ttl 128 prot UDP
UDP: 520->520 len 32
RIP: RESPONSE vers 1 entries 1:
158.97.22.0 1

Fri May 06 01:58:36 1994 - irq9 recv:
Ether: len 306 08:00:20:0a:33:7f->ff:ff:ff:ff:ff:ff type IP
IP: len 292 158.97.20.62->158.97.20.0 ihl 20 ttl 60 prot UDP
UDP: 520->520 len 272
RIP: RESPONSE vers 1 entries 13:
0.0.0.0 2 158.97.32.0 3 158.97.33.0 2
158.97.1.0 1 158.97.34.0 3 158.97.35.0 3
158.97.10.0 2 158.97.11.0 3 158.97.12.0 3
158.97.13.0 4 158.97.14.0 3 158.97.30.0 3 158.97.31.0 4

Una vez que ya se han recibido el total de rutas, pc-pruebas tiene la Tabla de Enrutamiento completa. Esta tabla, producto final del trabajo, se muestra a continuación.

Tabla V.- Tabla de Enrutamiento completa de pc-pruebas.

Red destino	Long	Interface	Gateway	Metric	Reloj	uso
158.97.30.0	24	irq9	158.97.20.62	4	103	0
158.97.10.0	24	irq9	158.97.20.62	3	103	0
158.97.21.0	24	irq9	158.97.20.3	2	113	0
158.97.31.0	24	irq9	158.97.20.62	5	103	0
158.97.14.0	24	irq9	158.97.20.62	4	103	0
158.97.11.0	24	irq9	158.97.20.62	4	103	0
158.97.1.0	24	irq9	158.97.20.62	2	103	0
158.97.32.0	24	irq9	158.97.20.62	4	103	0
158.97.22.0	24	irq9	158.97.20.4	2	115	0
158.97.12.0	24	irq9	158.97.20.62	4	103	0
158.97.33.0	24	irq9	158.97.20.62	3	103	0
158.97.34.0	24	irq9	158.97.20.62	4	103	0
158.97.35.0	24	irq9	158.97.20.62	4	99	0
158.97.13.0	24	irq9	158.97.20.62	5	103	0
158.97.21.0	24	irq9	158.97.20.3	2	113	0
158.97.23.0	24	irq9	158.97.20.5	1	0	0
default	0	irq9	158.97.20.62	3	0	0

Finalmente, pc-pruebas debe compartir su tabla de enrutamiento, como lo hacen normalmente los sistemas comerciales, con los enrutadores vecinos. Para realizar esta función, el programa de aplicación, envía cada 30 segundos un paquete RIP (de respuesta) con todas las rutas existentes en su tabla de enrutamiento, el paquete es de tipo multicast y dirigido al segmento de la subred a la cual el enrutador experimental pc-pruebas pertenece.

A continuación se muestran los mensajes de envíos realizados por pc-pruebas mediante el proceso de salida RIP implementado en la aplicación.

Fri May 06 01:58:39 1994 - irq9 sent:
Ether: len 366 08:06:33:44:00:00->08:00:20:0a:33:7f type IP
IP: len 352 158.97.20.5->158.97.20.0 ihl 20 ttl 15 prot UDP
UDP: 520->520 len 332
RIP: RESPONSE vers 1 entries 16:
0.0.0.0 3 158.97.30.0 4 158.97.10.0 3
158.97.34.0 4 158.97.31.0 5 158.97.14.0 4
158.97.11.0 4 158.97.35.0 4 158.97.1.0 2
158.97.21.0 2 158.97.32.0 4 158.97.12.0 4
158.97.33.0 3 158.97.22.0 2 158.97.13.0 5 158.97.23.0 1

Una vez que la Tabla de pc-pruebas es recibida por los enrutadores vecinos, ellos deben agregar a su tabla, las rutas existentes en pc-pruebas, siempre y cuando, tengan un valor de metric menor o simplemente sea una ruta que no exista en su tabla.

Este último es el caso de la ruta hacia la red 158.97.23.0, la cual es agregada en el enrutador "FISICAS" de la red de Física Aplicada.

Para ilustrar lo anterior se muestra la tabla de enrutamiento de "fisicas" antes de recibir las rutas de pc-pruebas en la Tabla VI, posteriormente se muestra la misma información en la Tabla VII una vez que llegaron las rutas de pc-pruebas.

Tabla VI.- Tabla de Enrutamiento de físicas antes de recibir las rutas de pc-pruebas.

Destination	Gateway	Flags	Refcnt	Use	Int
localhost	localhost	UH	2	4658	lo0
default	cicese-gw	UG	1	61789	le0
subred-Sismologia	cicese-gw	UG	0	0	le0
subred-Computo	cicese-gw	UG	4	193308	le0
red-cicese-dorsal	fisica	U	26	173648	le0
subred-Ciencias-Comp	cicese-gw	UG	0	0	le0
subred-Oceanologia1	oceano	UG	0	10318	le0
subred-Informatica1	oceano	UG	0	0	le0
subred-Informatica2	oceano	UG	0	0	le0
subred-Fisica-Aplica	fisicas	U	9	673546	le1
subred-Electronica	nov-EYT1.cicese.mx	UG	2	74973	le1
subred-Oceanologia2	oceano	UG	0	0	le0
subred-Geofisica	cicese-gw	UG	1	133	le0
subred-Optica	nov-Opt.cicese.mx	UG	0	38986	le1
subred-Est.Esp.	cicese-gw	UG	0	0	le0

Tabla VII.- Tabla de Enrutamiento de físicas despues de recibir las rutas de pc-pruebas.

Destination	Gateway	Flags	Refcnt	Use	Int
localhost	localhost	UH	2	4658	lo0
default	cicese-gw	UG	1	61789	le0
subred-Sismologia	cicese-gw	UG	0	0	le0
subred-Computo	cicese-gw	UG	4	193308	le0
red-cicese-dorsal	fisica	U	26	173648	le0
subred-Ciencias-Comp	cicese-gw	UG	0	0	le0
subred-Oceanologia1	oceano	UG	0	10318	le0
subred-Informatica1	oceano	UG	0	0	le0
subred-Informatica2	oceano	UG	0	0	le0
subred-Fisica-Aplica	fisicas	U	9	673546	le1
subred-Electronica	nov-EYT1.cicese.mx	UG	2	74973	le1
subred-Oceanologia2	oceano	UG	0	0	le0
subred-Geofisica	cicese-gw	UG	1	133	le0
subred-Optica	nov-Opt.cicese.mx	UG	0	38986	le1
158.97.23.0	pc-pruebas.cicese.mx	UG	0	0	le1
subred-Est.Esp.	cicese-gw	UG	0	0	le0

Como se puede observar en la Tabla VII, el enrutador físicas, ya tiene la ruta para poder encaminar un paquete hacia la red Novell (158.97.23.0), a través del enrutador experimental pc-pruebas.

Nótese que las Tablas de enrutamiento de "físicas", aunque poseen un formato diferente a la tabla obtenida en el enrutador experimental, tiene en esencia la misma información. La razón por la que la tabla observada en "físicas" es diferente a la tabla de pc-pruebas, se debe a que es desplegada con el comando NETSTAT -r , el cual pertenece al sistema operativo UNIX para SUN Sparc.

Una observación adicional. Como se puede observar, la ruta que aparece en la Tabla VII remarcada, tiene la columna de red destino = 158.97.23.0 y no un nombre pronunciable, la razón de esto es que no está dada de alta con algún nombre en el Domain Name System (DNS) de la red CICESE.

En base a los resultados obtenidos, mostrados en las tablas 5, 6 y 7, se pueden concluir los siguientes aspectos:

- * El enrutador experimental ha sido implementado con todos los atributos que se requieren para la generación y actualización dinámica de las tablas de enrutamiento.

- * El enrutador experimental ha demostrado su total compatibilidad con enrutadores comerciales, coexistiendo con éstos en un ambiente real de redes de cobertura amplia.

* El enrutador experimental cumple las expectativas iniciales de diseño en cuanto a:

- Compartición de rutas.
- Escucha de rutas.
- Alta y baja de rutas.
- Asignación dinámica de costos de enlaces.

En base a lo anterior se puede decir que el objetivo planteado originalmente se ha cumplido y se deja como una fuerte aportación el enrutamiento de paquetes desde un ambiente TCP/IP hacia una red Novell.

La aplicación desarrollada es lo bastante flexible y modular a pesar de su complejidad, ya que las implementaciones originales de enrutamiento están bajo el sistema operativo UNIX el cual provee la generación de multitarea y multiprocesamiento de una manera genérica y no en un ambiente de PC. A pesar de esto, la aplicación resultó eficiente.

Se deja un campo de investigación y desarrollo sobre este trabajo bastante amplio. Los problemas abiertos son para el desarrollo de programas de aplicación que interactúen entre Novell y TCP/IP y que permitan principalmente: 1).- El enrutamiento de paquetes de Novell hacia TCP/IP, 2).- La implementación del Domain Name System (DNS), 3).- La transferencia de archivos, 4).- Login remoto, etc., obviamente esta programación deberá seguir el esquema aquí presentado.

Además, un campo de investigación y desarrollo en el cual se puede aplicar este trabajo puede ser en el área de computación distribuida.

VI.6.- Validación del protocolo RIP con el paquete de simulación COMNET III.

Como una herramienta adicional, para la validación del algoritmo de enrutamiento RIP, se exploró el paquete de simulación COMNET III.

COMNET III es un paquete de simulación, que no requiere programación, puede ser empleado para analizar RLD's, RCA y Redes de Area Metropolitana (MAN's). A través de COMNET III, es posible modelar redes con tecnología TCP/IP, además posee los mecanismos de enrutamiento de paquetes mediante el protocolo de enrutamiento RIP, por todo esto, se consideró pertinente modelar una Red de Cobertura Amplia basada en TCP/IP en éste paquete de simulación.

COMNET III, posee una gran variedad de atributos que lo convierten en una poderosa arma para el análisis de Redes de Comunicaciones, entre los cuales se pueden mencionar los siguientes:

- **Simulación Interactiva.**- Este atributo permite hacer cambios mientras la simulación se está ejecutando, de esta manera, es posible alterar los parámetros de algún nodo o cambiar la topología de la red y observar los cambios generados con estos cambios.

- **Subredes.**- Es posible definir objetos que representan una subred, la cual puede tener su propia estructura y protocolos, independientemente del resto de la Inter-red.

- **Elementos del modelaje.**- Como se puede observar en la figura 35, la topología de la red está definida por nodos y enlaces. Los nodos desarrollan funciones de

procesamiento y contienen puertos para conectar enlaces, los puertos, proveen entrada o salida a buffer y procesamiento. Hay dos tipos de nodos: Los nodos de aplicación y los nodos de comunicaciones. Los primeros ejecutan aplicaciones y contienen almacenamiento para archivos, aceptan procesamientos de lectura, escritura, etc. Los nodos de comunicación, desarrollan conmutación o funciones de enrutamiento y son usados para interconectar nodos de aplicación y subredes.

La carga de la red es producida por aplicaciones ejecutandose en los nodos, por mensajes y generadores de conexión, los cuales demandan directamente el transporte de datos. Las aplicaciones pueden ser disparadas de acuerdo a una distribución probabilística o de acuerdo a una petición de llegada. Cada uno de los nodos tiene una lista ó una distribución de los posibles destinos a los cuales enviar paquetes de información, de ésta manera, se simula lo que representa el tráfico en la red.

Por último, los mensajes creados por las aplicaciones requieren ser transportados desde el origen hasta el destino, usando protocolos de Transporte, de Enrutamiento, de Enlace de Datos y de Control de acceso al medio. Los protocolos de Transporte son provistos por los nodos fuente y destino, desarrollando la fragmentación de paquetes en el nodo fuente y el posterior reensamblaje en el nodo destino.

Los paquetes son transportados de extremo a extremo, usando enrutamiento orientado a conexión o a no conexión (connectionless). Cada nodo selecciona el puerto de salida para cada uno de los paquetes usando un protocolo de enrutamiento. Los servicios de Enlace de Datos en los puertos, crean marcos de transmisión y entonces usan

Los paquetes son transportados de extremo a extremo, usando enrutamiento orientado a conexión o a no conexión (connectionless). Cada nodo selecciona el puerto de salida para cada uno de los paquetes usando un protocolo de enrutamiento. Los servicios de Enlace de Datos en los puertos, crean marcos de transmisión y entonces usan el controlador de acceso al medio para contender por el enlace. Los servicios de la capa física son provistos por los enlaces.

En la figura 35, se observa la metodología para la configuración de parámetros para la simulación de redes de comunicaciones, cada uno de éstos parámetros se selecciona de las opciones que provee COMNET III, sin embargo, algunos de ellos, es posible crearlos, es decir, el usuario tiene la posibilidad de definir sus propios esquemas de aplicaciones y de transporte de datos.

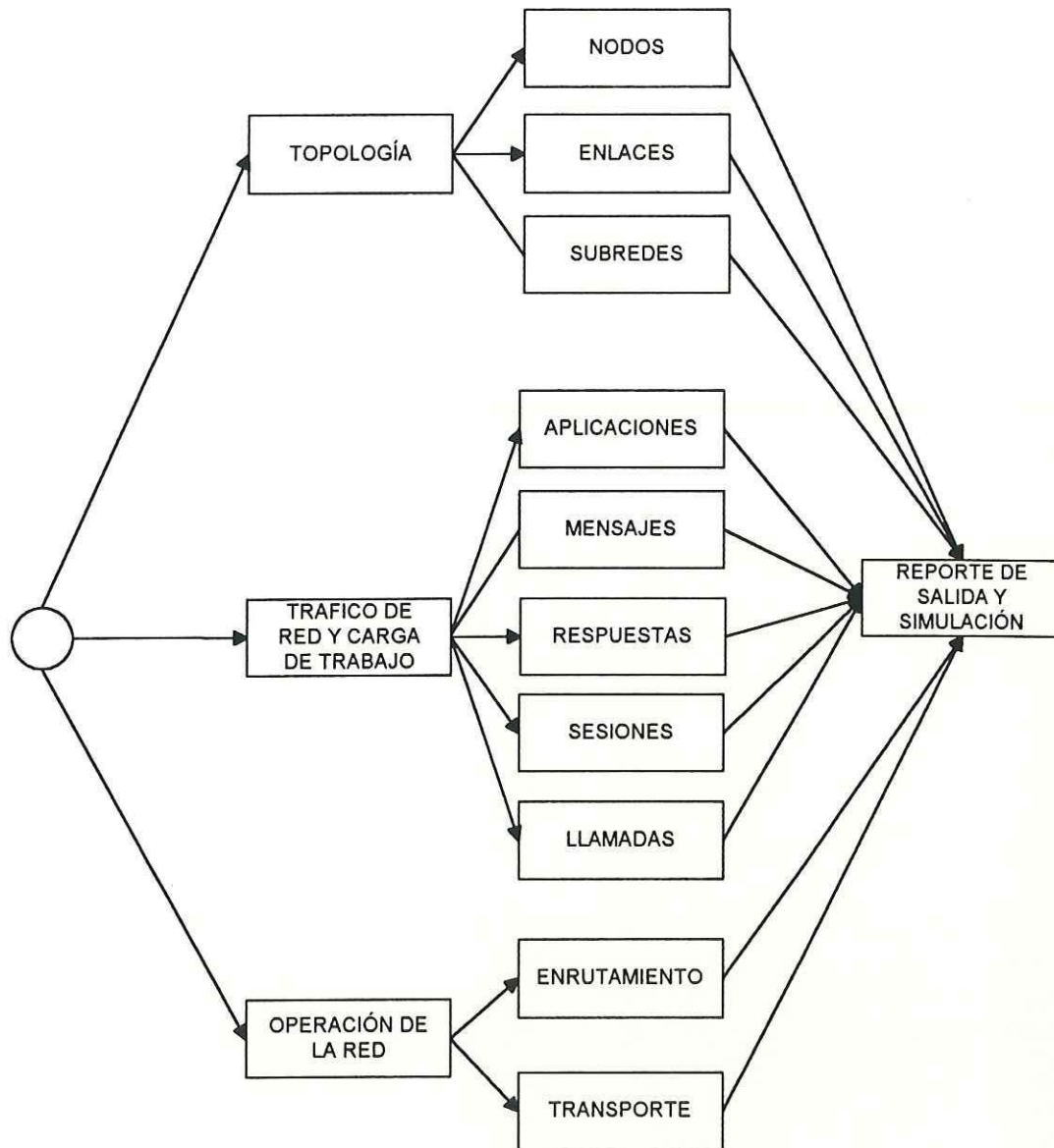


Figura. 35.- Metodología empleada para la simulación en COMNET III.

En el caso particular del trabajo de tesis, se simuló una red de cobertura amplia, mostrada en la figura 36, con 6 nodos de comunicaciones, cada uno de ellos con su propio generador de mensajes.

A cada uno de los nodos se le configuró el algoritmo de enrutamiento RIP, de ésta manera, el tráfico de paquetes entre nodos es realizado bajo el esquema de enrutamiento RIP, el cual se implementó en las secciones anteriores. Aunque se tiene la certeza que el protocolo de enrutamiento RIP genera tablas de enrutamiento en cada uno de los nodos de la simulación y que eventualmente, éstas tablas son modificadas por cambios en la topología, no es posible observarlas, por ésta razón no se documenta más formalmente el proceso de simulación desarrollado para la RCA, es decir no se documenta la elección de parámetros para cada nodo, enlace, generador de mensajes, etc, sin embargo el trabajo realizado con COMNET III prueba su factibilidad en la simulación de parámetros de Redes de comunicaciones menos específicos.

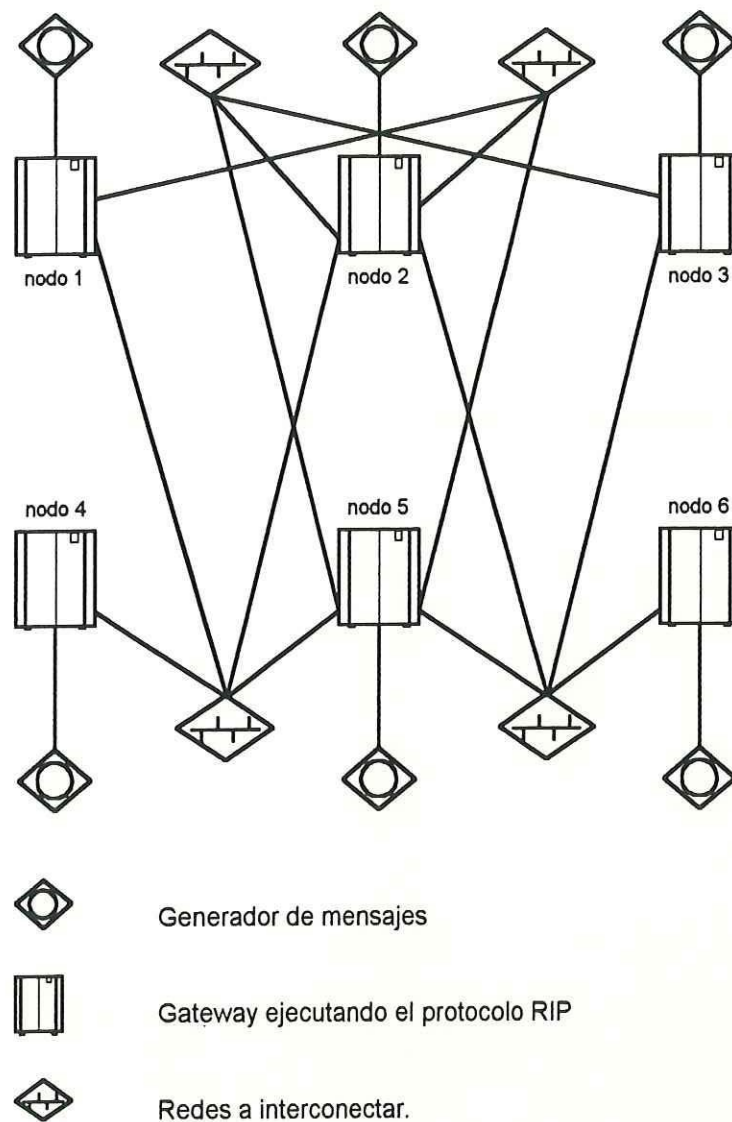


Figura. 36.- Topología de la RCA simulada en COMNET III.

En el siguiente capítulo, se concluye finalmente sobre el trabajo de tesis, haciendo una serie de aportaciones y recomendaciones en base al estudio realizado.

VII.- CONCLUSIONES, APORTACIONES Y RECOMENDACIONES.

VII.1.- Conclusiones.

Tomando en cuenta el desarrollo de este trabajo de tesis, el cual incluye consideraciones en cuanto a:

- * La investigación de las arquitecturas de red más empleadas actualmente como son TCP/IP y Novell.
- * La investigación de mecanismos de enrutamiento de paquetes en la actual interred.
- * La interconectividad de RLD's a través de la interred.
- * La investigación, desarrollo y tendencias de dispositivos enrutadores y
- * El desarrollo de software de un enrutador experimental con características genéricas y además comerciales, ejecutando el protocolo de enrutamiento RIP para el encaminamiento de paquetes en ambientes Novell - TCP/IP, se pueden concluir los siguientes aspectos:

1.- El proceso de interconectividad de RLD's tales como Ethernet, ARCnet, Token-ring, etc. regidas por sistemas operativos como Novell, se apoya fuertemente en la suite de protocolos TCP/IP para enrutamiento más poderoso ó hacia RCA's, ya que la confiabilidad y madurez de éstos proporcionan esquemas bajo los cuales se solucionan las principales diferencias existentes de RLD's, concluyendo en la importancia del estudio y utilización de la arquitectura de red DoD.

2.- El direccionamiento usado por TCP/IP, es sumamente funcional pero dado el crecimiento desmesurado de la Interred, requerirá migrar de una longitud de 32-bits a una de mayor longitud a corto o mediano plazo.

3.- Las redes bajo Novell, desde su inicio, fueron planeadas como RLD's, por esta razón, existen parámetros que dificultan su conexión a la actual Interred, entre los cuales se encuentran: el direccionamiento, el enrutamiento, el modelo de comunicación y la fragmentación. De estos aspectos se puede concluir que son de suma importancia en el estudio de problemas de interconectividad y que de éstos se pueden obtener soluciones y nuevos esquemas de interconexión de redes de comunicaciones.

4.- Existen diferentes esquemas de interconexión entre redes Novell y redes TCP/IP, el más empleado de ellos, conocido como tunel IP/IPX, tiene fuertes desventajas en cuanto a su aplicación en RCA, la principal de ellas es el sobreprocesamiento en los paquetes, el cual limita la eficiencia de la red Novell. Claramente se puede concluir que la sobrecarga, más en las aplicaciones existentes hoy en día, es un factor importante de estudio y cuidado en cuanto al desarrollo de software de enrutamiento.

5.- Si bien, el protocolo de enrutamiento RIP, es capaz de solucionar los problemas existentes en la Interred en materia de cambios topológicos, es inminente su desplazamiento hacia mejores protocolos como el protocolo Open Shortest Path First (OSPF), el cual no ha sido implantado a gran escala aún, ni aceptado ampliamente.

6.- Los paquetes con información de enrutamiento no deben ser muy grandes, ya que éstos son transmitidos periódicamente y podrían afectar el desempeño de la red. Por esta misma razón, el tamaño de las tablas de enrutamiento debe restringirse.

7.- Se concluye que las ciencias computacionales dentro del área de telecomunicaciones, así como en otras, es de suma importancia, ya que aporta una nueva metodología de estudio y un punto de vista diferente para atacar los problemas que se presenten. Además, provee la inteligencia de comunicación, la mejora de técnicas de enrutamiento y proporciona un medio para la unión de varias ciencias de una manera adecuada sin perder de vista el objetivo en común.

8.- La estrategia empleada en el desarrollo del presente trabajo de tesis, demuestra la factibilidad de estudiar, desarrollar e implementar protocolos de comunicaciones en ambientes de redes de datos reales, basándose para ello en especificaciones formales de los mismos; es decir, seguir una línea de estándares para que un desarrollo ó aplicación constituya una base sólida para los sistemas que requieran este tipo de soporte.

9.- Tomando en consideración la estrategia de éste trabajo, se puede decir que el objetivo de la tesis quedó cubierto en su totalidad, investigando, desarrollando e implementando las funciones de los enrutadores interred, vistos generalmente como una "caja negra", que es empleada transparentemente en la interconectividad de RLD's y que proporciona los mecanismos necesarios para el enrutamiento de datagramas en una red de conmutación de paquetes.

VII.2.- Aportaciones del Trabajo.

Las principales aportaciones que proporciona este trabajo de tesis son:

- * El análisis de la estructura e interrelación entre los diferentes protocolos de las arquitecturas de red DoD y Novell, desde el punto de vista de las ciencias computacionales y no sólo de telecomunicaciones.

- * La definición de los principales tipos, funciones y problemas asociados al manejo de aspectos de enrutamiento, como lo son las tablas y algoritmos de enrutamiento, para la interconectividad de RLD's a RCA.

- * El análisis, desarrollo e implementación del protocolo RIP, IP en conjunto con el protocolo IPX.

- * La obtención de un enrutador experimental con las siguientes características:

- Compatible con enrutadores comerciales
- Basado en RIP, el cual provee actualizaciones dinámicas de sus tablas de enrutamiento.
- Su implementación puede ser en una PC-XT debido a la programación desarrollada, haciéndolo un enrutador "barato" de implementar.
- Requerimientos de memoria y espacio en disco son mínimos, aísla tráfico en red y produce menos sobrecarga en la red Novell como enrutador.

- * Se definieron las bases y procedimientos para desarrollar un esquema de interconexión entre redes más eficiente a través de la programación.

* Desarrollo de una metodología para la simulación en redes de cobertura amplia del protocolo RIP mediante el paquete de simulación COMNET III.

* La aportación en el enrutamiento de paquetes desde una red TCP/IP hasta una Novell y viceversa, atravesando para ello el enrutador experimental desarrollado.

* Este trabajo de tesis deja una fuerte aportación para el desarrollo de nuevas tecnologías en cuanto a las capas de aplicación de red.

VII.3.- Recomendaciones.

Se recomienda profundizar más en la programática de la arquitectura de red Novell, para poder interactuar más dinámicamente con el programa de aplicación obtenido y de esta manera, poder desarrollar aplicaciones que vayan más allá del enrutamiento de paquetes.

En base al análisis desarrollado de la Interred y dado que el protocolo de enrutamiento RIP limita a 16 el número de enrutadores que un paquete puede atravesar, se recomienda que las redes basadas en TCP/IP estén estructuradas con el mismo número de estratos ó subredes.

Se recomienda también a los administradores de las redes TCP/IP, que pongan atención especial al enrutamiento IP, ya que un esquema de enrutamiento deficiente en cuanto a topología y configuración de parámetros de enrutamiento, puede afectar en gran medida el desempeño de la red.

Por último, se recomienda que las implementaciones de algún protocolo en especial sean desarrolladas basándose en especificaciones formales existentes de organismos adecuados, porque cualquier incompatibilidad entre las implementaciones afecta directamente la operabilidad de la interred, evitando los fracasos en implementaciones como en el caso de OSPF.

LITERATURA CITADA.

Braden, R. 1987. "Requirements for Internet Gateways". RFC 1009. USC/Information Sciences Institute.

Chappel, Laura. 1993. Novell's Guide to NetWare Analysis. Novell Press. San Jose. 412 p.

Comer, Douglas E. 1991. Internetworking with TCP/IP. Vol. I. Prentice Hall. New Jersey. 116-299 pp.

Comer, Douglas E. 1991. Internetworking with TCP/IP. Vol. II. Prentice Hall. New Jersey. 32-124 pp.

Davis, Ralph. 1990. NetWare Programmer's Guide. Addison-Wesley publishing Co. New York. 453-496 pp.

Feit, Sidnie. 1993. TCP/IP Architecture, Protocols and Implementation. McGraw Hill Inc. San Francisco. 446 p.

Hedrick, C. 1988. "Routing Information Protocol". RFC 1058. USC/Information Sciences Institute. 33p.

Liebing, Edward y Neff, Ken. 1990. NetWare Server Handbook. McGraw Hill Inc. New York. 458-471 pp.

Mills, D. L. 1984. "EGP Formal Specification". RFC 904. USC/Information Sciences Institute. 30 p.

Narten, Thomas. 1989. "Internet Routing". ACM Computing. Indiana. 271-282 p.

Naugle, Matthew G. 1994. Network Protocol Handbook. McGraw Hill Inc. San Francisco. 510 p.

Novell. Inc. 1992. "IPX Router Specification". Novell Inc. Utah. 49 p.

LITERATURA CITADA (continuación).

- Plummer, David C. 1982. "An Ethernet Address Resolution Protocol (ARP)". RFC 826. USC/Information Sciences Institute. p.
- Postel, J. B. 1981. "Internet Protocol - DARPA Internet Program Protocol Specification". RFC 791. USC/Information Sciences Institute. 45 p.
- Postel, J. B. 1981. "Internet Control Message Protocol - DARPA Internet Program Protocol Specification". RFC 792. USC/Information Sciences Institute. 21p.
- Postel, J. B. 1980. "User Datagram Protocol". RFC 768. USC/Information Sciences Institute. 4p.
- Provan, D. 1991. "Tunneling IPX Traffic through IP Networks". RFC 1234. USC/Information Sciences Institute. 5p.
- Rose, Charles G. 1990. Programmer's Guide to NetWare. McGraw Hill Inc. New York. 261-472 pp.
- Santifaller, Michael. 1991. TCP/IP and NFS, Internetworking in a UNIX environment. Addison-Wesley publishing Co. New York. 12-30 pp.
- Schwaderer, W. David. 1992. C Programmer's Guide to NetBios, IPX and SPX. SAMS Publishing Co. Carmel IN. 505 p.
- Thomas, Richard. 1993. "Building Up NetWare, Brick by Brick". Data Communications. 22(13):83-88 pp.

APENDICE A

Este apéndice formaliza la especificación de la interface de programación para tarjetas de red, conocida como Packet Driver Specification (PDS) de la compañía FTP, que se utilizó en este trabajo. Se basa en la especificación original Rev. 1.09 (1989).

Esta especificación fue obtenida mediante los servicios que ofrece Internet de FTP anónimo. El PDS se encuentra en varios servidores FTP, se extrajo de la dirección *omingate.clarkson.edu*.

La compañía FTP provee varios manejadores de tarjetas desarrollados según la especificación PDS, los que fueron utilizados son: NE2000.COM para tarjetas de red de 16 bits y NE1000.COM para tarjetas de red de 8 bits.

A través de éstos, una aplicación se puede comunicar directamente con la tarjeta de red por medio de ciertas funciones que se activan por una interrupción de software, por el otro lado, al recibir un paquete, se activa una interrupción de hardware que "llama" al manejador localizado en alguna dirección de memoria RAM. Estos parámetros se especifican de la siguiente manera en el *prompt* del sistema operativo de disco:

```
A:\>ne2000 0x60 9 0x340
```

en donde el primer parámetro es la interrupción de software por donde la aplicación se comunicará con el manejador, el segundo indica la interrupción de hardware y el último es la dirección base de RAM para el manejador.

Con el manejador en memoria RAM, la aplicación puede realizar las siguientes operaciones por medio de las funciones que proporciona el manejador:

* Identificación de tarjeta de red (Ethernet, Token Ring, ProNET-10, etc.)

Nombre de función: driver_info

Número de función: 1

* Mecanismo para recepción de paquetes:

Nombre de función: access_type

Número de función: 2

Nombre de función: release_type

Número de función: 3

* Envío directo de paquetes:

Nombre de función: send_pkt

Número de función: 4

* Obtención de información de la tarjeta (Dirección, tipo, clase, etc.):

Nombre de función: get_address

Número de función: 6

Existen más funciones como la colección de estadísticas, etc. pero no fueron utilizadas en este trabajo. Especificaciones particulares del uso del PDS para esta aplicación, se encuentra documentada en los módulos codificados en lenguaje C que acompañan a esta tesis (MANEJAP.C y MANEJAP.H).

APENDICE B

Los módulos de implementación están codificados en lenguaje C principalmente, éstos comprenden un total de 18 archivos (extensión .c) con 13 archivos de encabezados y definiciones (extensión .h). Además, se tienen 3 archivos que contienen código en lenguaje Ensamblador para la PC (XT ó AT). Todos se incluyen en el disco que acompaña a este trabajo de tesis.

Los compiladores utilizados para la generación del módulo ejecutable *enruta.exe* fueron: Turbo C++ ver. 3.0 y Turbo C ver 2.0, Turbo Ensamblador ver. 2.0 de la compañía Borland. Todos los módulos fueron compilados en el modelo de memoria *small* utilizando un "project" definido en el archivo *enruta.prj*. Para obtener el módulo .obj del código en ensamblador se utilizaron los siguientes parámetros en el Turbo Ensamblador:

```
C:\>tasm -mx -t módulo.asm
```

Para la puesta en marcha del enrutador experimental se debe ejecutar el archivo batch *enruta.bat* ó manualmente se pueden ejecutar los siguientes pasos (que viene siendo el contenido del archivo batch):

```
ipx  
net5  
f:\login pc-pruebas  
ne2000 0x7e 9 0x340  
enruta
```

Este enrutador es experimental, por lo tanto, fué desarrollado para un escenario específico: 2 tarjetas de red máximo, una para un segmento de red Novell y otra para un segmento de red TCP/IP. Esto puede interpretarse como una limitación del trabajo, pero el diseño de software que se realizó, su estructuración y modularidad provee una flexibilidad suficiente para que con mínimas modificaciones al código, se puedan manejar varias tarjetas de red con el fin de que el enrutador conecte a más de 2 redes directamente. Los módulos de enrutamiento no se modificarían mucho (excepto en 2 ó 3 líneas de código) y no se ven afectados por modificaciones en las capas inferiores a las de enrutamiento (enlace y física).

Cada protocolo viene implementado en un archivo con el mismo nombre de éste, por ejemplo, el protocolo RIP está contenido en el archivo *rip.c*. Existen archivos para los protocolos arp, icmp, ip, rip y udp. El protocolo IP es el único que encuentra dividido en 3: *ip1.c*, *ip2.c* e *ip3.c*. Los demás módulos contienen la siguiente información:

<i>enruta.c:</i>	Contiene el SCP y procesamiento de temporizadores, main().
<i>mbuf.c:</i>	Manejo de estructuras mbuf (extracción, inserción en cola).
<i>if.c:</i>	Manejo de estructura para la interface con el manejador de paquetes
<i>manejap.c:</i>	Rutinas para el control de la interface con el manejador.
<i>varios2.c:</i>	Rutinas de propósito general (conversiones de tipos).
<i>tiempo.c:</i>	Manejo de temporizadores (inicio y terminación).
<i>recibe.c:</i>	Rutinas para la recepción de paquetes por IPX.
<i>envia.c:</i>	Rutinas para el envío de paquetes IPX.
<i>manda.c:</i>	Rutina para el envío de mensaje a nodo en red Novell.
<i>ethernet.c:</i>	Rutinas para el manejo de paquetes Ethernet (Blue Book).
<i>manejip.asm:</i>	Rutina de servicio de interrupción para paquetes IP.
<i>manejipx.asm:</i>	Rutina ESR para paquetes IPX.
<i>varios.asm:</i>	Rutinas de propósito general.

La ejecución de la aplicación se ve reflejada en el monitor de la PC sólo con el propósito de mostrar las actualizaciones de la tabla de enrutamiento, aunque esto no es requisito, ya que las capas de red son implementadas sin interface al usuario. La pantalla muestra de una manera aproximada los datos que se muestran en la Tabla V del capítulo VI.

Para el término de ejecución del módulo ejecutable sólo se tiene que presionar cualquier tecla y se regresará al *prompt* del sistema operativo.