UNIVERSIDA NACIONAL AUTONOMA DE MEXICO

FACULTAD DE ESTUDIOS SUPERIORES Aragón

Propuesta de implementación del protocolo IS–IS con IPv6 en el backbone de la red CUDI

Tesis que presenta el alumno:

Rogelio Morales Galindo

Para obtener el titulo de:

Ingeniero Mecánico Electricista

Con asesor:

Ing. Narciso Acevedo Hernández

2006

Dedicatoria

Esta tesis esta dedicada especialmente a mi abuelita Félix, a mis papas; Alicia e Ignacio, a mis hermanas; Rocío, Esthela, Olga e Irene y a mis sobrinos; Bryan, Carlos, Sharon, Kenia, Axel y Fernanda.

Agradecimientos

A mis padres

Quisiera agradecerles por el apoyo incondicional que me han brindado, por ser los mejores padres que pude tener, por ser el mayor incentivo de superación en mi vida y por estar a m lado en mis errores y en mis aciertos.

A mi abuela

Quisiera haberle agradecido en persona pero desgraciadamente ya no esta físicamente a mi lado aunque si espiritualmente, y es que fue un gran ejemplo de superación y de lucha en la vida diaria, se que donde este disfrutara este logro tanto como lo esta disfrutando mi familia.

A mis hermanas

A ellas que siempre me han alentado en los estudios y que me han servido de ejemplo en la vida escolar y en la vida diaria ya que me han ayudado a formar un carácter de superación y de entrega para con mi familia.

A la Universidad Nacional Autónoma de México

Por permitirme ser uno de sus tantos egresados, por dejarme conocer una pequeña parte de la universidad y por dejarme sentir el orgullo de ser universitario.

A mis amigos

Agradezco por que participaron en mi formación académica dentro y fuera de la universidad y por alentarme cuando requerí de su amistad.

A mis compañeros de trabajo

Agradezco por permitirme aprender y orientarme respecto a la vocación laboral.

CONTENIDO

Objetivos ix

Introducción *xi*

Capítulo 1 Internet *1*

Historia de Internet *3*
ARPAnet *3*
Nacimiento de Internet *3*
NFSnet *4*
Internet actual *6*

Internet 2 *7*
Surgimiento *7*
Abilene *8*
Organización *9*

Internet 2 en México *11*
CUDI *11*
Organización *13*

Capítulo 2 Modelos de referencia *17*

Modelo OSI *19*
Descripción del modelo OSI *20*
Protocolo CLNP *26*
Direccionamiento CLNP *28*

Modelo TCP/IP *31*
Descripción del modelo TCP/IP *32*

Protocolo IPv4 *33*
Formato del paquete y descripción de los campos *34*

Direccionamiento IPv4 *37*
Tipos de dirección *38*
Subnetting *44*
VLSM *46*
CIDR *48*

Protocolo IPv6 *50*
Formato del paquete y descripción de los campos *51*
Encabezados de extensión *53*

Direccionamiento IPv6 *54*
Tipos de dirección *56*
Direcciones compatibles *59*

Capítulo 3 Enrutamiento *61*

Introducción *63*
Tabla de enrutamiento. *65*

Mecanismos de enrutamiento *66*
Directamente *conectado* *66*
Estático *67*
Predeterminado *68*
Dinámico *69*

Protocolos de enrutamiento _____ *70*
Protocolos Vector–distancia _____ *71*
Protocolos Estado–enlace _____ *73*

Dominios de enrutamiento _____ *74*

Protocolos de enrutamiento interior (IGP) _____ *76*
RIP _____ *77*
OSPF _____ *78*
IS-IS _____ *79*

Protocolos de enrutamiento exterior (EGP) _____ *80*
BGP _____ *80*

Capítulo 4 IS–IS _____ *83*

Historia de *IS–IS* _____ *85*

Conceptos básicos de *IS–IS* _____ *86*
Dominio de enrutamiento _____ *86*
Área y jerarquía de enrutamiento _____ *87*
Intermediate System y End System _____ *89*

Paquetes de *IS-IS* _____ *90*
Mensajes Hello _____ *92*
Paquetes LSP _____ *94*
Paquetes SNP _____ *99*
Soporte para IPv6 _____ *101*

Funcionamiento de *IS-IS* _____ *103*

Conclusiones _____ *105*

Bibliografía _____ *109*

Apéndices _____ *113*

Apéndice A
Extracto de la propuesta presentada en *CUDI* _____ *115*

Apéndice B
Configuración de los routers de la red *CUDI* _____ *119*

Apéndice C
Glosario _____ *123*

Objetivos

Elaborar una propuesta para implementar el protocolo de enrutamiento *IS–IS* en la red *CUDI,* lo cual permitirá soportar al nuevo protocolo de Internet versión 6*(IPv6)* con una eficiencia mayor para todas las Universidades e Institutos conectados a la red, este documento a su vez, será tomado como referencia para las Universidades que deseen implementar este protocolo en sus *Campus.*

Proporcionar la información necesaria para la comunidad estudiantil acerca de la red académica nacional destinada para la investigación y desarrollo educativo en las Universidades e Institutos dentro y fuera México, así como también mostrar el desarrollo que ha alcanzado el nuevo protocolo de Internet (IPv6), mostrar el soporte y la difusión que está alcanzando este nuevo protocolo.

Introducción

El rápido y continuo crecimiento de Internet ha provocado la investigación y el desarrollo de tecnologías emergentes que permitan mejorar el desempeño, la implementación y la operación de Internet.

La investigación y el desarrollo de las nuevas tecnologías se realiza mediante grupos de trabajo a nivel internacional, en los que colaboran un gran número de participantes que elaboran propuestas de tecnologías que posteriormente se convierten en estándares que deberán ser implementados para alcanzar el optimo desempeño de la red.

Las instituciones académicas han jugado un papel clave en la investigación y requieren de interactuar con mucha gente a nivel mundial para acelerar la investigación y el desarrollo tecnológico, debido a las necesidades de la investigación fue necesario implementar redes académicas que permitiesen la comunicación entre universidades e institutos de investigación.

En México se creo una red académica llamada *"red CUDI"*, esta red permite la interconexión de Universidades e institutos en México y debido al crecimiento de la misma se requirió llevarla al plano internacional estableciendo una conexión hacia la red académica de Estados Unidos (*Abilene*) y otra conexión hacia la red sudamericana (*Clara*).

Los inicios de la red *CUDI* se dieron en base a lo que en ese momento se podía implementar, pero debido a las nuevas tendencias tecnológicas, y al desarrollo de estas, es necesario llevar a cabo migraciones de unas tecnologías a otras para que se pueda mejorar el rendimiento y operación de la red

Algunas de las nuevas tecnologías que están comenzando a implementarse son:

- *Multiprotocol Label Switching (MPLS)*
- *Internet Protocol version 6 (IPv6)*
- *Traffic Engineering (TE)*

Actualmente la *red CUDI* no ha implementado *MPLS* ni *TE* en su infraestructura y el soporte para enrutar paquetes de *IPv6* dentro de la red y hacia las universidades o institutos conectados a la red no esta completo debido a limitaciones de *hardware* o de *software.*

La *red CUDI* cuenta actualmente con el protocolo *OSPF* para enrutar los paquetes de *IPv4* y con el protocolo *RIP* para enrutar los paquetes de *IPv6*, debido a la necesidad de enrutar paquetes de *IPv6* de una forma mas eficiente se realizo la búsqueda de un protocolo que pudiese cumplir con los requerimientos necesarios para llevar a cabo esa tarea y el protocolo elegido fue el protocolo *Integrated System to Integrated System (IS–IS).*

Para poder implementar un nuevo protocolo de enrutamiento interno en la *red CUDI* es necesario elaborar un documento que justifique el motivo de la migración del protocolo para ser expuesto ante las autoridades correspondientes de la *red CUDI*.

Para elaborar el documento mencionado en el párrafo anterior es necesario conocer el funcionamiento y las características del protocolo propuesto para determinar y justificar porque se esta proponiendo *IS–IS*, además de exponer las ventajas que traería al desarrollo de la red.

En esta tesis se redacta la información necesaria para llevar a cabo la elaboración de la propuesta que se presentara ante las autoridades pertinentes de la red CUDI.

La estructura de esta tesis se encuentra dividida en cuatro capítulos, a través de los cuales se podrán comprender desde los conceptos históricos hasta los conceptos que permitirán el funcionamiento del protocolo IS–IS.

A continuación se presenta una breve descripción de cada capitulo.

Capitulo 1

Este capitulo describe la historia del Internet desde su nacimiento, pasando por las diversas etapas de su desarrollo y hasta llegar al desarrollo de las redes académicas

Capitulo 2

En este capitulo se da un repaso desde los conceptos básicos como los modelos de referencia empleados para la transmisión de datos, además de la descripción de los tipos de protocolos empleados en cada modelo.

Se describe también el sistema de direccionamiento *NSAP* para el modelo *OSI* y el sistema de direccionamiento *IP* para el modelo *TCP/IP*, en este ultimo modelo se hace una referencia hacia el actual sistema de direccionamiento *(IPv4)* y del nuevo sistema de direccionamiento *(IPv6)* que esta siendo desarrollado e implementado en las redes académicas de todo el mundo.

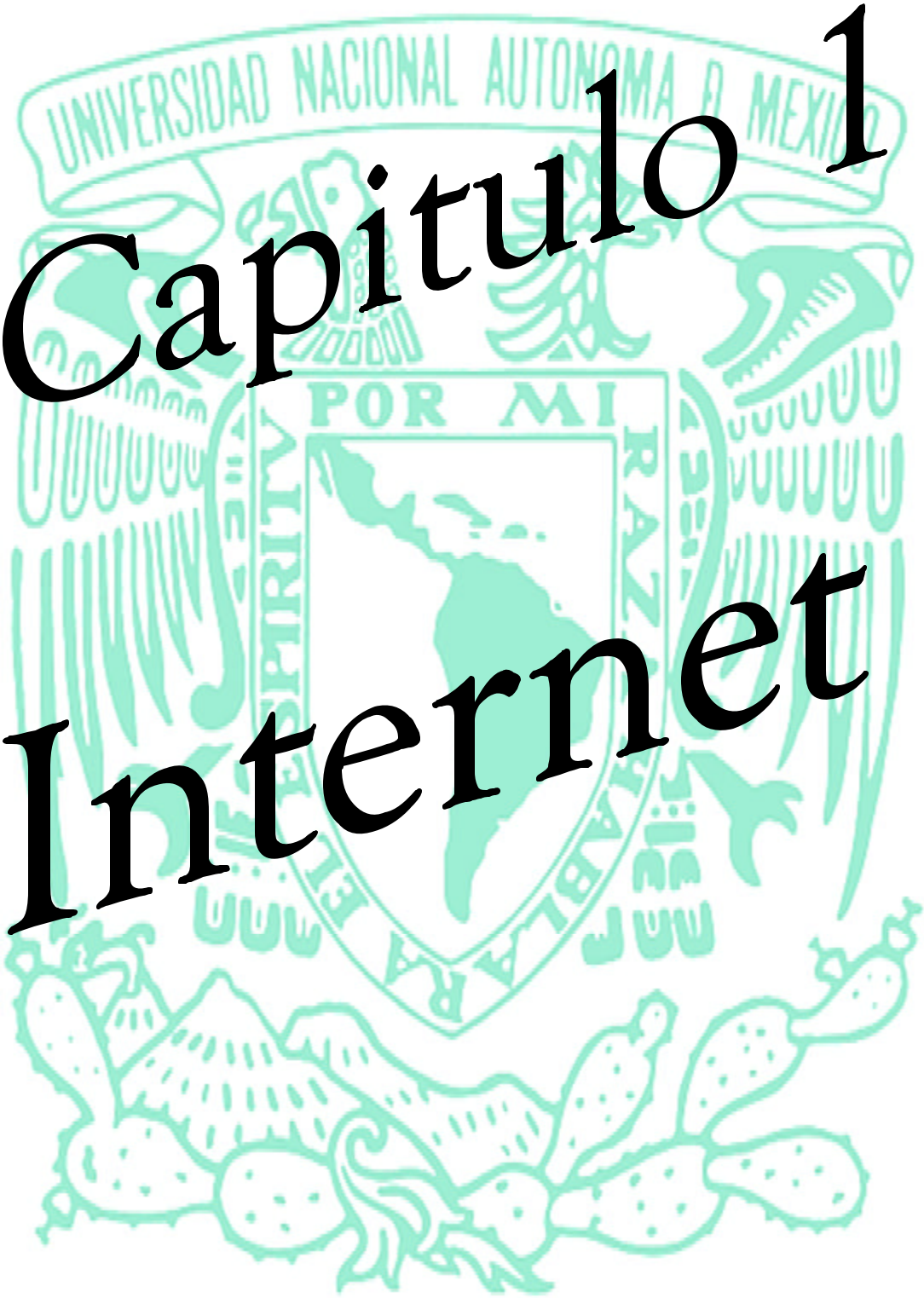
Capitulo 3

Es este capitulo detalla los conceptos del enrutamiento de paquetes, desde los inicios del enrutamiento de paquetes hasta las características de los protocolos de enrutamiento mas conocidos en el ámbito del enrutamiento.

Capitulo 4

Este ultimo capitulo esta dedicado al protocolo *IS–IS* y en el se describe desde su nacimiento, sus etapas de desarrollo y las modificaciones que le han sido agregadas para soportar las tecnologías que van emergiendo con el desarrollo de la comunicación de datos.

.



Capitulo 1

Internet

Historia de Internet

Como definición de Internet podemos encontrar dos significados distintos, el primero se refiere a *internet* (con i minúscula), se refiere a dos o más computadoras que se pueden comunicar entre ellas. Y la *internet* mas notable es la llamada Internet (con I mayúscula), que es una colaboración de mas de 150,000 redes interconectadas. Industria privada, además de organizaciones gubernamentales, instituciones educativas, institutos de investigación, corporaciones y bibliotecas en más de 100 países.

Pero como se fue desarrollando el Internet, a continuación es presentada una breve historia del Internet, desde sus orígenes y por lo cual es necesario explicar *ARPANET.*

ARPANET

A mediados de los 60's las computadoras en organizaciones de investigación funcionaban como independiente. Las computadoras de diferentes fabricantes eran incapaces de comunicarse entre ellas, fue entonces que la agencia de proyectos de investigación avanzada *(Advanced Research Project Agency, ARPA)* del departamento de la defensa *(Deparment of Defense, DoD)* se intereso en buscar una forma de poder comunicar las computadoras y así los investigadores poder compartir sus conocimientos, reduciendo costos y eliminando la duplicación de esfuerzo.

En 1967, dentro de una reunión de la Asociación de la industria de la computación (*Association for Computing Machinery, ACM*), *ARPA* presento un proyecto para interconectar un pequeño grupo de redes, llamado *ARPANET*, la idea era conectar varias computadoras a pesar de que estas fueran de diferente marca.

En 1969, el proyecto *ARPANET* fue hecho realidad, con cuatro nodos principales, la *Universidad de California en Los Angeles* (*UCLA*), la *Universidad de California en Santa Barbara (UCSB)*, el instituto de *Investigación de Stanford (SRI)* y la *Universidad de Utah*. El *ARPANET* continuo su crecimiento y agregando instituciones académicas e industriales.

Nacimiento de Internet

En 1972, *Vint Cerf* y *Bob Kahn*, quienes habían formado parte del grupo de investigación de *ARAPANET*, colaboraron en un proyecto llamado *Internetting*, ellos querían enlazar diferentes redes, para permitir la conexión de una computadora a otra pero en redes diferentes.

Hubo problemas que tuvieron que ser resueltos, respecto a : el tamaño de los paquetes, la diferencia de las interfaces y las tasas de transmisión. *Vint Cerf* y *Bob Kahn* tuvieron la idea de un dispositivo llamado gateway, para servir como de enlace intermedio para transferir paquetes de una red a otra.

Estos dos investigadores en 1973 publicaron un protocolo que permitiría la entrega de paquetes de extremo a extremo, lo llamaron protocolo de control de

transmisión.*(TCP)* Estos protocolos incluyen conceptos tales como, encapsulación, el datagrama y sus funciones.

En octubre de 1977, una *internet* consistía de tres redes diferentes (*ARPANET*, red de paquetes de radio y la red de paquetes vía satélite) fue exitosamente comprobada, con lo cual ya habría comunicación entre esas tres redes.

En corto tiempo la autoridades de la red tomaron la decisión de dividir *TCP* en dos protocolos: *TCP* e *IP (Internetworking Protocol)*. *IP* manejaría el enrutamiento de datagramas, mientras que *TCP* seria el responsable de funciones de más alto nivel, tales como la segmentación reensamblamiento y la detección de error.

Así, el sistema de protocolos que surgieron a partir de este proyecto serian conocidos como *"TCP/IP"*.

En 1981, la *Universidad de California Berkeley* dio a cada productor de hardware una versión abierta del sistema operativo *UNIX*, en el cual se habían establecido los protocolos *TCP/IP*, para que ellos incluyeran *TCP/IP* en sus sistemas.

En 1983, las autoridades establecieron los protocolos originales de *ARPANET*, y *TCP/IP* serian los protocolos oficiales de ARPANET. Mejor conocida como *"suite de protocolos TCP/IP"*, y todos aquellos que quisieran acceder a Internet desde una red diferente debían usar *TCP/IP*.

En 1983, *ARPANET* se dividió en dos redes; *MILNET* para fines militares y *ARPANET* para usuarios no militares.

CSNET es otra red que surgió poco después de *ARPANET*, en 1981, como iniciativa de las universidades interesadas en la comunicación de redes pero sin los mismos atributos que *ARPANET* y fue patrocinada por la Fundación Nacional de Ciencias *(National Science Fundation, NSF).* Esta red no era tan inteligente como la de *ARPANET*. Pero *CSNET* era mas barata, aunque no tenía redundancia y era mas lenta.

A mediados de los 80's muchas universidades de los Estados Unidos formaron parte de *CSNET* y otras instituciones y compañías formaron sus propias redes usando *TCP/IP* para interconectarse. El termino Internet fue asociado a las redes conectadas que fundo el gobierno. Ahora se refiere a las redes conectadas mediante *TCP/IP*.

NSFNET

A finales de los 80's y con el éxito de *CSNET*, la *NSF* se propuso conectar los seis centros de supercomputadoras que se encontraban esparcidos por todo el país, la idea era permitir el acceso a a toda la comunidad universitaria, centros educativos, agencias gubernamentales e inclusive la industria privada. Hasta ese momento solo las grandes universidades y las grandes corporaciones tenían acceso a las grandes supercomputadoras.

En un principio se intento utilizar la red *ARPANET*, pero surgieron muchos problemas burocráticos y por lo tanto decidieron crear su propia red, que seria llamada *NSFNET (National Science Foundation Network).*

NSFNET se baso en la tecnología *IP* de *ARPANET*, y fue creada en 1986 mediante circuitos punto a punto de 56 Kbps *(DS-0).* Para hacer rentable la red en lugar de hacer una red en estrella, conectando cada universidad con las supercomputadoras mediante extensos circuitos dedicados, se hicieron redes regionales, se conecto cada centro a la red regional mas cercana, y por ultimo se conectaron todas estas redes regionales entre si, figura 1.1..

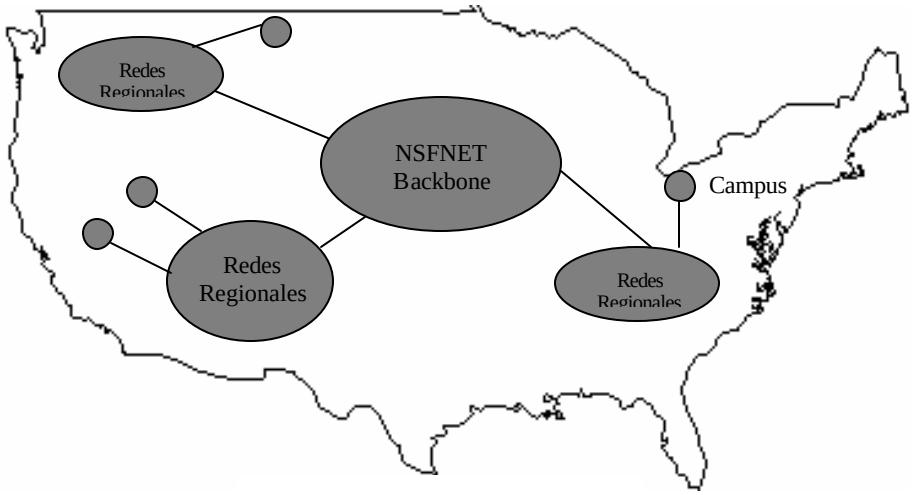


Figura 1.1

El soporte regional fue proporcionado pro consorcios de redes y el soporte local por institutos de investigación y educativos,.gran parte de ese apoyo vino de gobiernos estatales y federal, pero una contribución considerable fue hecha por la industria.

El comienzo de esta red fue tan espectacular, que pronto se vio saturada. En la primera mitad de 1988 la red transportaba 115 millones de paquetes por mes, esto ocasiono que en julio del mismo año se cambiaran los circuitos por unos de mayor capacidad, 1.544 Mbps *(T1)* y en diciembre de 1992 a 45 Mbps *(T3).* En noviembre de ese año, el trafico alcanzo magnitudes de 24,000 millones de paquetes por mes.

Desde 1997, la *NSF* había contratado la dirección, operación y desarrollo de *NSFNET* a la compañía *Merit Inc.*, la cual trabajaría en este proyecto junto con *MCI Corporation* e *IBM*. En septiembre de 1990, *Merit*, *MCI* e *IBM* crearon una nueva compañía llamada *ANS* (*Advanced Network & Services, Inc*), la cual se encargo de la red *NSFNET.*

Pero como había acontecido con *ARPANET, NSFNET* había llegado a su fin y dejo de dar soporte al tráfico de Internet en *EE.UU*. a partir de abril de 1995, las funciones de la red troncal de Internet fueron asumidas por la red Internet *MCI.* Posteriormente han surgido otras redes troncales y en la actualidad existen un gran número de redes regionales, también llamadas redes de nivel medio, que operan en el ámbito de estado o mediante consorcios con las universidades.

NSFNET tuvo el merito de haber sido la red que soportara Internet a todos los usuarios del ámbito universitario y de los centros gubernamentales norteamericanos durante los difíciles primeros años de introducción a la red.

.

Internet actual

El Internet actual consiste de redes de todo el mundo interconectadas mediante equipos que permiten esa conexión. Mas de 36 millones de están distribuidos en mas de 135, 000 redes. Gran parte del *backbone* o la espina dorsal de Internet trabajan a 155 Mbps *(STM-1),* la figura 1.2 muestra una parte del Internet; las nubes representan redes y las cajas representan los equipos de conexión.

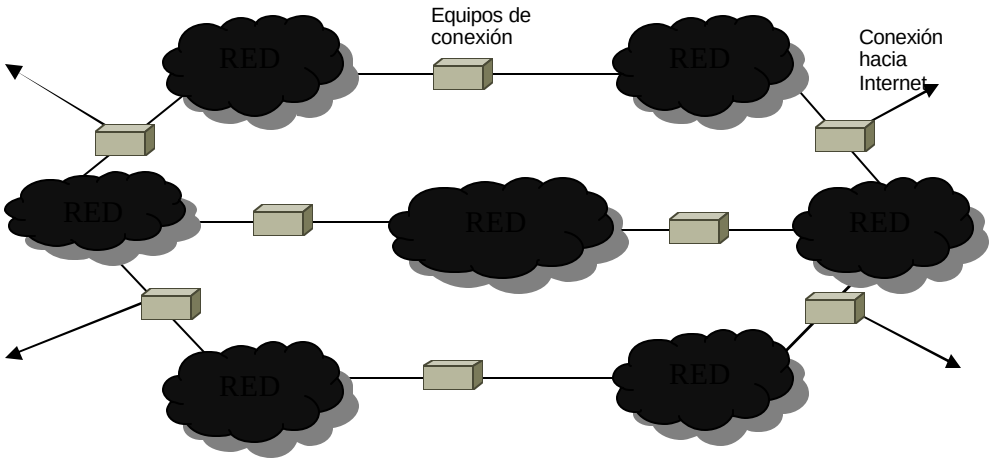


Figura 1.2

Internet2

Surgimiento

En octubre de 1996, un grupo de 34 universidades norteamericanas, encabezadas por la *National Science Foundation (NSF)* inicio un proyecto conocido como *Internet2*. a este proyecto se unieron posteriormente socios empresariales como: *Ameritech, Cisco, Digital, IBM, MCI, Sun,* y gubernamentales como el fondo para el Internet de nueva generación *(Next Generation Internet, NGI).*

Los objetivos de este proyecto son:

- Crear una red de alta velocidad para la investigación.
- Facilitar y coordinar el desarrollo, despliegue y operación de nuevas aplicaciones y servicios sobre Internet.
- Asegurar la transmisión de datos sobre esta red.

En 1997, un mensaje emitido por el presidente de los *EE.UU.*, definió a *Internet2* como una segunda generación de Internet que permitirá a las universidades y centros de investigación comunicarse a velocidades 1000 ve ces mayores a las actuales. El principal objetivo del gobierno norteamericano fue apoyar el mantenimiento del liderazgo de *EE.UU*. en esta industria emergente.

Desde el punto de vista del protocolo, uno de los avances ha sido la introducción del concepto *"Garantía de calidad de servicio"*, mejor conocido como *"Quality of Service guarantee"*. Esto significa que las aplicaciones demandaran una cierta cantidad de ancho de banda (velocidad mínima de transmisión garantizada) o una prioridad especifica que le permita dar respuesta con un mínimo de calidad (tiempo de respuesta). Expresado de otra forma, la red podrá dar prioridad a los paquetes de voz y video en tiempo real, a diferencia de otros paquetes de datos de otro tipo.

Actualmente esta es la distribución de las universidades de *EEUU* conectadas a *Internet2* , las cuales suman 206 universidades, figura 1.3.

Figura 1.3

Abilene

A medida que *Internet2* fue creciendo dentro de los *EE.UU.*, se hizo necesario construir un *backbone* de alto desempeño que permitiese la implementación de aplicaciones de red avanzadas y llevara los servicios de red a todas las universidades y laboratorios de investigación en todo el país. Esta red tomaría el nombre de *Abilene* y seria la red *IP* más avanzada disponible para universidades participantes en Internet2.

En abril de 1998, el vicepresidente *Al Gore* anuncio la red *Abilene* durante una ceremonia en la Casa Blanca. La operación inicial comenzó en febrero de 1999 y la implementación fue de 2.5 Gbps en sus enlaces y fue completada a finales de 1999. figura 1.4.

Figura 1.4

Esta red es netamente educativa y por lo tanto nos permite emplear un mayor ancho de banda y una mayor eficiencia para el desempeño de las conexiones, permite llevar a cabo de forma más efectiva la comunicación entre investigadores permitiendo compartir sus conocimientos y crear nuevas formas de enseñanza.

Abilene soporta el desarrollo de aplicaciones tales como: laboratorios virtuales, bibliotecas digitales, educación a distancia y tele inmersión, también como la capacidad para interconectarse con otras redes de investigación de alto desempeño de los *EE.UU.* y del plano mundial.

Abilene conecta redes regionales, mediante puntos de acceso llamados *"gigaPoPs"*, para proporcionar los servicios de redes avanzadas *(Internet2)* a mas de 220 universidades, corporaciones y miembros afiliados en los 50 estados y los distritos de *Columbia* y *Puerto Rico* .

La red actual esta formada actualmente por enlaces de fibra óptica de 10 Gbps *(OC-192c)* como tecnología de transporte y de *routers* de alto desempeño dentro del *backbone* y desempeña el protocolo *IPv6* en forma nativa, esta descripción actual fue alcanzada en el 2003.

Abilene es una colaboración de *Internet2, Qwest Communications, Nortel Networks, Juniper Networks, Cisco System* y la *Universidad de Indiana*.

Con el paso del tiempo, esta red ha venido creciendo día a día, no solo en los Estados Unidos, sino que comenzó a esparcirse por los países europeos, asiáticos y americanos. Después de crear infraestructuras por países se comenzó la creación de arquitecturas continentales.

Actualmente, existen un sinnúmero de redes de *Internet2* en todo el mundo y por consiguiente una gran cantidad de universidades e institutos que pueden tener entre si servicios como: videoconferencias de más alta velocidad, transmisiones de video en nuevos formatos y el desarrollo de nuevas tecnologías.

Entre las redes mas conocidos para nuestro país tenemos; la red *Geant* en Europa, red *Clara* en Latinoamérica, red *Iris* en España y la red *CANRIE* en Canadá.

Organización

Como ya se había mencionado, el proyecto de *Internet2* esta conformado por Universidades, Institutos de Investigación, Departamentos gubernamentales e Industria privada. Pero cada uno de estos grupos tiene una tarea diferente dentro de la organización y manutención del proyecto.

Actualmente, *Internet2* esta constituido por cuatro tipos de miembros diferentes, los cuales están divididos en los siguientes grupos:

- *Universidades*
- *Corporativos*
- *Afiliados*
- *Asociaciones*

En el grupo de Universidades, a diferencia de las 35 que comenzaron el proyecto, se puede apreciar un crecimiento significativo hasta 206 universidades como miembros del proyecto, aunque son muchas, no son todas las universidades de los *EE.UU*.

Los corporativos que ayudan en el patrocinio e implementación de nueva infraestructura o equipos que sean empleados para la investigación, están divididos en tres grupos que de acuerdo a su colaboración, se ubican dentro de cada uno :

- *Socios corporativos*
- *Patrocinadores corporativos*
- *Miembros corporativos*:

Actualmente hay 12 socios corporativos que ayudan a proveer la infraestructura a las universidades para probar, implementar y desplegar tecnologías avanzadas de red junto con las aplicaciones que estas implican.

En cuanto al grupo de patrocinadores corporativos, se tienen actualmente 13 patrocinadores.

Por ultimo, tenemos el grupo de miembros corporativos, y respecto a este grupo se cuenta actualmente con 42 miembros.

Por otro lado, este proyecto también tiene miembros afiliados, y respecto a la cantidad de ellos, actualmente hay 42 miembros afiliados.

En cuanto a las asociaciones pertenecientes a *Internet2*, solo hay 2 asociaciones colaborando con este proyecto.

Todos estos miembros contribuyen económicamente para actualizar y mejorar el desempeño de la red de *Internet2* para que permita una mayor eficiencia en la Investigación y desarrollo de proyectos.

Podemos encontrar mayor información acerca de los grupos de trabajo, proyectos, congresos, y eventos de *abilene* en *abilene.internet2.edu* y sobre *Internet2* en *www.internet2.edu*

Internet2 en México

CUDI

Al igual que en el mundo de la investigación en todo el mundo, la creación de redes dedicadas única y exclusivamente a la investigación hizo eco en México, en la comunidad universitaria, la sociedad y el gobierno federal, para desarrollar una red de investigación que permitiese desempeñar aplicaciones en sobre Internet. A este proyecto se le llamo *"Internet2"* en México.

Pero el desenvolvimiento del proyecto tuvo varios sucesos importantes, a continuación mencionaremos solo algunos de ellos.

El 8 de abril de 1999 se oficializo en los Pinos la creación de la *Corporación Universitaria para el Desarrollo de Internet (CUDI).*

CUDI es una asociación civil sin fines de lucro que fue creada a petición de la sociedad científica del país de universidades publicas y privadas que fomentara e impulsara el desarrollo de aplicaciones e implementación entre sus miembros.

Sus objetivos fundamentales, al igual que la red de *Internet2* de *EE.UU.*, son:

- *Crear una red de alta velocidad para uso exclusivo de la investigación.*
- *Coordinar el desarrollo, despliegue y operación de nuevas aplicaciones y servicios sobre Internet.*
- *Garantizar la calidad de transferencia de datos sobre la red.*

El 20 de mayo, en San Diego California, representantes de *CUDI* firmaron 2 memorandums con 2 importantes corporaciones universitarias que promueven y coordinan la disponibilidad de redes avanzadas para aplicaciones de Investigación y educación en los *EE.UU.* Las 2 corporaciones son:

- *UCAID (Universitary Corporation for Advanced Internet Development)*
- *CENIC (Corporation of Education Network Initiatives in California)*

Estas importantes corporaciones acordaron mediante los memorandums colaborar con *CUDI* en el desarrollo de tecnologías y aplicaciones para la red de *Internet2* en México.

Para la creación de la infraestructura, había que buscar patrocinadores que ayudaran a las universidades a crear la infraestructura que con el tiempo se llamaría "*red CUDI"*. Para ello, se firmaron convenios con *TELMEX* , en primera instancia y tiempo después se agrego *AVANTEL.*

Haciendo uso de este convenio, *TELMEX* y *AVANTEL* han aportado sin costo alguno a la red *CUDI* 8,000 Km de infraestructura para formar un *backbone* de alta velocidad para la red de *Internet2* de México. A cambio de la donación, se estableció que la red debería ser de uso académico exclusivamente, es decir solo habría tráfico de carácter educativo o de investigación.

La distribución de los nodos o *PoPs* se dio de la siguiente forma, ver tabla 1.1.

NODO	TELMEX	AVANTEL
Cancún		X
Guadalajara	X	
Juárez	X	
México	X	X
Monterrey	X	X
Tijuana	X	

Tabla 1.1

De la asignación de nodos podemos ver que la distribución de los puntos de acceso, en el *backbone*, para *Internet2* será la siguiente, figura 1.5:

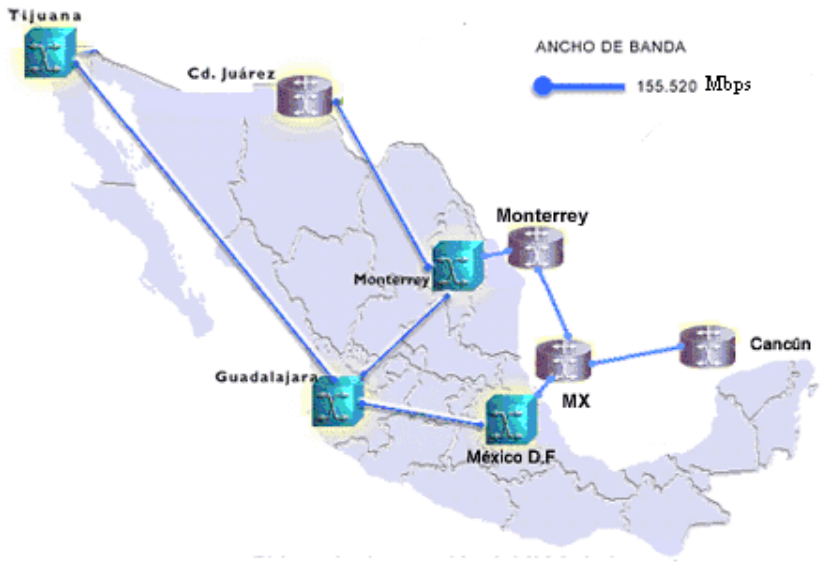


Figura 1.5

Mediante el diseño de este *backbone*, se permitirá la conexión de todas las universidades en institutos hacia *Internet2*, facilitando su ubicación geográfica.

Debido a la mayor congregación de Universidades e Institutos en ciertos sectores geográficos y a la factibilidad de la infraestructura, es por lo que existen dos nodos en México y Monterrey.

Actualmente, los enlaces que hay entre los nodos principales son de 155 Mbps *(STM-1 u OC3)*. Y hay enlaces hacia de la misma capacidad hacia *Abilene* en Juárez y Tijuana y hacia *Clara* en Tijuana.

Las conexiones hacia las universidades se dan mediante conexiones E3 *(34 Mbps)*.

Organización

La organización de este proyecto en México se dio por iniciativa de Universidades pioneras en la tecnología, así que cuando se creo el proyecto, una de las prioridades era implementar un esquema que permitiese a largo plazo implementar una red muy parecida a la red de *Internet2* en los *EE.UU*. pero adecuándola a México.

La organización se dividió en 4 principales grupos de miembros, haciendo la analogía a la red de *EE.UU*., los tipos de miembros quedo de la siguiente forma:

- *Asociados Académicos*
- *Afiliados Académicos*
- *Asociados Institucionales*
- *Afiliados Empresariales*

Los Asociados Académicos son las Universidades que colaboran económicamente para mantener el buen desempeño, la actualización y la operación de la red *CUDI*.

Los Afiliados Académicos son las Universidades que solo desean conectarse a la red, pero sin involucrarse en cuestiones de mantenimiento y operación del *backbone* de *Internet2* (red *CUDI*).

Los Asociados Institucionales son instituciones no universitarias, que colaboran económicamente para el mantenimiento y operación de la red, es decir son de la misma proporción que los Asociados Académicos.

Los Afiliados empresariales son instituciones no universitarias que realizan aportaciones económicas menores a las de los Asociados

En el grupo de Asociados Académicos, hay actualmente 17 universidades miembros y esta propensa a continuar su crecimiento, tabla 1.2.

	Asociados Académicos de CUDI
1	Benemérita Universidad Autónoma de Puebla (BUAP)
2	Centro de Investigación Científic a y de Educación Superior de Ensenada (CICESE)
3	Centros Públicos de Investigación CONACYT
4	Instituto Latinoamericano de Comunicación Educativa (ILCE)
5	Instituto Politécnico Nacional (IPN)
6	Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM)
7	Universidad Autónoma de Ciudad Juárez (UACJ)
8	Universidad Autónoma de la Laguna (UAL)
9	Universidad Autónoma de Nuevo León (UANL)
10	Universidad Autónoma de Tamaulipas (UAT)
11	Universidad Autónoma del Estado de Hidalgo (UAEH)
12	Universidad Autónoma del Estado de Morelos (UAEM)
13	Universidad Autónoma Metropolitana (UAM)
14	Universidad de Guadalajara (UDG)
15	Universidad de las Américas Puebla (UDLAP)
16	Universidad Nacional Autónoma de México (UNAM)
17	Universidad Veracruzana (UV)

Tabla 1.2

El grupo de Afiliados Académicos es mucho mayor que el de los Académicos, ya que cuenta con 33 miembros entre instituciones y universidades y 28 centros pertenecientes a el *CONACYT (Consejo Nacional de Ciencia y Tecnología),* ver tablas 1.3.

	Afiliados Academicos
1	Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV)
2	Colegio de Postgraduados (COLPOS)
3	Colegio Nacional (COLNAL)
4	Instituto de Investigaciones Eléctricas (IIE)
5	Instituto Mexicano del Petróleo (IMP)
6	Instituto Nacional de Salud Pública (INSP)
7	Instituto Tecnológico Autónomo de México (ITAM)
8	Instituto Tecnológico de Estudios Superiores de Irapuato (ITESI)
9	Instituto Tecnológico de Oaxaca (ITO)
10	Laboratorio Nacional de Informática Avanza da (LANIA)
11	Texas A&M University Center México (TAMU)
12	Universidad Autónoma de Aguascalientes (UAA)
13	Universidad Autónoma de Baja California (UABC)
14	Universidad Autónoma Chapingo (UChapingo)
15	Universidad Autónoma de Chihuahua (UACH)
16	Universidad Autónoma de Coahuila (UADEC)
17	Universidad Autónoma de San Luis Potosí (UASLP)
18	Universidad Autónoma de Tlaxcala (UATX)
19	Universidad Autónoma del Estado de México (UAEMEX)
20	Universidad Autónoma de Yucatán (UADY)
21	Universidad de Colima (UCOL)
22	Universidad de Guanajuato (UGTO)
23	Universidad de Sonora (USON)
24	Universidad del Valle de México (UVM)
25	Universidad Juárez Autónoma de Tabasco (UJAT)
26	Universidad Iberoamericana (UIA)
27	Universidad La Salle (ULSA)
28	Universidad Panamericana (UP)
29	Universidad Pedagógica Nacional (UPN)
30	Universidad Politécnica del Estado de Morelos (UPEMOR)
31	Universidad Popular Autónoma del Estado de Puebla (UPAEP)
32	Universi dad Regiomontana (UR)
33	Universidad Tecnológica de México (UNITEC)

Tabla 1.3

Dentro de este grupo, existen una gran cantidad de centros de investigación que dependen del *CONACYT* y son listados en la tabla 1.4.

	Centros CONACyT
1	Centro de Investigación en Alimentación y Desarrollo, A.C. (CIAD)
2	Centro de Investigación y Asesoría Tecnológica en Cuero y Calzado, A.C. (CIATEC)
3	Centro de Investigación y Asesoría en Tecnología y Diseño del Estado de Jalisco, A.C. (CIATEJ)
4	Centro de Investigación y Asesoría Técnica del Estado de Querétaro, A.C. (CIATEQ)
5	Centro de Investigaciones Biológicas del Noroeste, S.C. (CIBNOR)
6	Centro de Investigación Científica de Yucatán, A.C (CICY)
7	Centro de Investigación y Docencia Económicas, A.C. (CIDE)
8	Centro de Ingeniería y Desarrollo Industrial (CIDESI)
9	Centro de Investigación y Desarrollo Tecnológico en Electroquímica, S.C. (CIDETEQ)
10	Centro de Investigaciones y Estudios Superiores en Antropología Social (CIESAS)
11	Centro de Investigación en Geografía y Geomática "Ing. Jorge L. Tamayo", A.C. (CIGGET)
12	Centro de Investigación en Matemáticas, A.C. (CIMAT)
13	Centro de Investigación en Materiales Avanzados, S.C. (CIMAV)

Tabla 1.4

	Centros CONACyT
14	Centro de Investigaciones en Optica, A.C. (CIO)
15	Centro de Investigación en Química Aplicada (CIQA)
16	Colegio de la Frontera Norte, A.C. (COLEF)
17	Colegio de México, A.C. (COLMEX)
18	Colegio de Michoacán, A.C. (COLMICH)
19	Colegio de San Luis, A.C. (COLSAN)
20	Corporación Mexicana de Investigación en Materiales, S.A. de C.V. (COMIMSA)
21	Colegio de la Frontera Sur (ECOSUR)
22	Fondo para el Desarrollo de Recursos Humanos (FIDERH)
23	Facultad Latinoamericana de Ciencias Sociales (FLACSO)
24	Instituto Nacional de Astrofísica, Optica y Electrónica (INAOE)
25	Instituto de Ecología, A.C. (INECOL)
26	Fondo De Información y Documentación para la Industria (INFOTEC)
27	Instituto Potosino de Investigación IPICYT
28	Instituto de Investigaciones "Dr. José María Luis Mora" (MORA)

Tabla 1.4 (continuación)

El grupo de Asociados Institucionales esta compuesto por solo 4 miembros, los cuales se han ido incrementando, ver tabla 1.5.

	Asociados Institucionales
1	Avantel S.A. de C.V.
2	Cisco System de México S.A. de C.V
3	Consejo Nacional de Ciencia y Tecnología (CONACYT)
4	Telefonos de México S.A. de C.V.

Tabla 1.5

Y por ultimo, el grupo de Afiliados empresariales solo cuenta con un miembro, que es la empresa *VCON Inc*.

Entre todas estas instituciones se encargan de llevar a cabo reuniones y a partir de ellas se crearon grupos de trabajo para llevar a cabo la investigación referente a la operación, actualización, implementación y operación de la *red CUDI.*

Los grupos de trabajo que hay hasta este momento son:

- *QoS*
- *End to End*
- *Enrutamiento*
- *H.323*
- *HDTV*
- *IPv6*
- *MPLS*
- *Middleware*
- *Multicast*
- *NOC*
- *Seguridad*
- *Topologia*

Podemos encontrar mayor información acerca de los grupos de trabajo y de *CUDI* en general en *www.cudi.edu.mx*, y para mayor información acerca de *CLARA* se puede consultar la pagina *www.redclara.net*.

Capitulo 2

Modelos de referencia

Modelo OSI

En los principios de las telecomunicaciones, solo había sistemas propietarios de las tecnologías y propietarios de los protocolos. Muchos sistemas operativos fueron desarrollados por compañías, tales como: *IBM, Digital Equipment Corporations*. Esos sistemas operativos y sus protocolos correspondientes facilitaban la comunicación entre sus mismos sistemas, pero cuando *IBM* desarrollo *SNA* y *Digital Equipment Corporations* desarrollo *DECNet* no previeron la interconexión entre sistemas de diferente tecnología.

Como se podía prever en los siguientes años la compañías propietarias se dedicaron a desarrollar sistemas operativos que fueran capaces de interconectarse con otros sistemas de diferente manufactura. Tan pronto como esto comenzó a ocurrir seria necesario crear algún tipo de empresa estándar que permitiera a los sistemas de las compañías compartir información y comunicarse entre ellas.

A principios de los 80´s, la Organización Internacional de estandarización *(ISO)* y la Unión Internacional de Telecomunicación *(ITU–T)* liberaron un modelo estándar que llamarían *Open System Interconectión (OSI)* que poco a poco reemplazaría el modelo *TCP/IP*.

El modelo de referencia *OSI* consta de una arquitectura de siete capas (Figura 2.1); Aplicación, Presentación, Sesión, Transporte, Red, Enlace y Física, las cuales definen diferentes funciones de interconexión

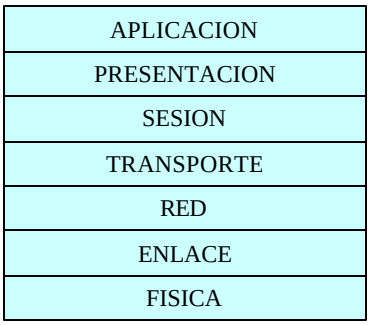


Figura 2.1

Capa de: aplicación. Proporciona servicios a aplicaciones de usuario.

Capa de presentación. Traducción, conversión, encriptación, desencriptación y compresión de datos.

Capa de sesión. Administración de sesión y control de dialogo.

Capa de transporte. Accesibilidad de conexión extremo a extremo entre programas y procesos.

Capa de red. Direccionamiento lógico y enrutamiento.

Capa de enlace. Transmisión y recepción de tramas.

Capa física. Codificación de señal, medios de transmisión y conectores.

El propósito del modelo de referencia *OSI* es permitir a sistemas similares y no similares comunicarse transparentemente mediante la creación de una arquitectura estándar para productores de hardware y software, aunque tal conexión no siempre es alcanzada.

Esas capas representan una serie de funciones y estándares que los vendedores deben seguir para alcanzar la interconexión, aunque los vendedores conservan la libertad de poder interpretar y decidir como desean agregar las especificaciones para las diferentes capas. A continuación vamos a describir con mayor profundidad las siete capas del modelo *OSI*.

Descripción del modelo OSI

Capa de aplicación

Este concepto puede confundir a la gente porque ellos creen que se refiere a aplicaciones de usuario tales como: *word, excel, power point*, etc. La capa de aplicación no se refiere a aplicaciones de software, pero más allá de un portal que facilita el acceso a recursos entre una aplicación y sistema final *(PC)* a través de una red. Este portal proporciona una ventana al modelo *OSI* para preparar los datos y ser empaquetados y enviados por el cable.

A la información contenida en esta capa se le llama *"datos"*.

La capa de aplicación permite a las aplicaciones de usuario enviar datos a través de la red, es decir, prepara los datos para que estos puedan acceder a las capas de nivel inferior y asi llegar a el usuario final.

El trabajo de la capa de aplicación es proporcionar una interfaz al conjunto de protocolos que transportaran los datos. A diferencia de otras capas, esta no proporciona servicios a cualquiera de las otras capas, solo proporciona acceso para servicios en la capa de aplicación.

Algunos de los servicios de la capa de aplicación son:

- *Aplicaciones con la red y servicios de interconexión.*
- *Servicios de archivo e impresión.*
- *Correo electrónico, Acceso Web y http.*
- *Acceso Telnet a usuarios remotos y transferencia de archivos FTP*

Capa de presentación

La funcionalidad de la capa de presentación es incluida generalmente en muchas implementaciones dentro del protocolo de la capa de aplicación o programa. Por lo tanto la capa de presentación esta integrada como un componente dentro de un programa de capa superior y no es vista como protocolo distinto o separado dentro del modelo.

Esta capa presenta un formato común de datos a través de diferentes plataformas y es responsable de los siguientes servicios:

- *Conversión de datos y traducción.*
- *Compresión y descompresión de archivos.*

Capa de sesión

Esta capa maneja y establece sesiones entre dos usuarios remotos. Una sesión consiste de un dialogo entre capas de presentación en los dos sistemas. Esta capa también maneja las respuestas a esos requerimientos entre sistemas y también controla el dialogo entre dos aplicaciones en diferentes sistemas y también la transferencia de datos.

La eficiencia del control de diálogos entre sistemas depende de el modo de comunicación, ya que este puede ser *half-duplex* o *full-duplex*. En una configuración *half-duplex* solo un dispositivo puede transmitir o comunicarse a la vez mientras que los otros deben esperar un estado de reposo para poder comunicarse. En una comunicación *full-duplex* el dispositivo puede enviar y recibir a la vez, por lo tanto *full-duplex* es mas eficiente que *half-duplex*.

A los datos que son enviados a la capa de sesión se les llama *"mensajes"*.

Capa de transporte

La capa de transporte es utilizada principalmente para proveer garantía para la entrega de datos entre dos procesos de comunicación o programas corriendo es sistemas remotos. Sin embargo esto solo es confiable si el vendedor decide implementar un protocolo orientado a conexión, tal como el Protocolo de control de transmisión *(TCP).*

La capa de transporte realiza las siguientes tareas:

- *Controla la comunicación extremo a extremo entre dos procesos corriendo en diferentes sistemas.*
- *Proporciona servicios orientados a conexión y sin conexión a capas superiores.*
- *Usa direcciones de puerto de cliente y servidor para identificar los procesos corriendo dentro del sistema.*

El término utilizado en esta capa para los datos es *"segmento"*

La capa de transporte maneja el direccionamiento con puertos. Esas direcciones identifican el programa o proceso de capa superior en un dispositivo particular. Los sistemas particulares pueden tener varias aplicaciones activas simultáneamente, las cuales se identifican por dirección de puerto.

Para el modelo *OSI* se establecen 2 tipos de servicios que son orientados a conexión y sin conexión que están agrupados en *TP0* a *TP4*.

En el servicio orientado a conexión están el *TP0, TP1, TP2* y *TP3* mientras que en los no orientados a conexión esta solamente *TP4*

Protocolo de transporte 0 (TP0): El más simple protocolo de transporte OSI, desempeña funciones de segmentación y ensamblamiento.

: *Protocolo de transporte 1 (TP1):* Desempeña segmentación y ensamblamiento, ofrece recuperación básica de error, secuencias de datos y retransmisión de datos o reiniciar la conexión si un excesivo numero de datos son desconocidos.

Protocolo de transporte 2 (TP2): Desempeña segmentación y ensamblamiento tan bien como multiplexaje y demultiplexaje de transmisión de datos sobre solo un circuito virtual.

Protocolo de transporte 3 (TP3): Ofrece recuperación básica de error, desempeña segmentación y ensamblamiento, multiplexaje y demultiplexaje de transmisión de datos sobre solo un circuito virtual y secuencias de datos y retransmisión de datos o reiniciar la conexión si un excesivo numero de datos son desconocidos.

Protocolo de transporte 4 (TP4): Ofrece recuperación básica de error, desempeña segmentación y ensamblamiento, multiplexaje y demultiplexaje de transmisión de datos sobre solo un circuito virtual y secuencias de datos y retransmisión de datos o reiniciar la conexión si un excesivo número de datos son desconocidos. *TP4* proporciona servicio y funciones de transporte confiable tanto con servicios de red orientados a conexión y sin conexión. Este esta basado en *TCP* y es la única clase de protocolo *OSI* que soporta el servicio de red sin conexión.

En el modelo *TCP/IP* encontramos sus equivalentes con el nombre de Protocolo de control de transmisión *(TCP)* y protocolo de datagrama de usuario *(UDP)*, los cuales serán explicados posteriormente.

Capa de red

La capa de red cubre las siguientes tareas:

- *Direccionamiento lógico.*
- *Entrega de paquetes.*
- *Enrutamiento.*

La principal responsabilidad de la capa de red es el direccionamiento lógico mediante la asignación de dirección origen y destino.

La siguiente responsabilidad es determinar la mejor ruta para el encaminamiento de datos entre redes.

En esta capa se proporcionan dos tipos de servicios para la capa de transporte, que son:

- *Orientados a conexión (Conection-Orientated Network Service).*
- *No orientados a conexión (Conectionless Network Service).*

Servicio orientado a conexión

Este servicio requiere del establecimiento explicito de una trayectoria o circuito entre las entidades de la capa de transporte antes de transmitir datos.

Servicio no orientado a conexión

Realiza el transporte de datagramas y no requiere que un circuito sea establecido antes de transmitir datos.

Cuando el soporte es proporcionado por el *CLNS*, el enrutamiento usa protocolos de enrutamiento para intercambiar información. *CLNS* desempeña la entrega del mejor esfuerzo, conocida como *"best.effort"* que significa que no existe garantía de que los paquetes sean perdidos o duplicados.

El servicio utilizado para el encaminamiento de datos es el no orientado a conexión, ya que este nos permite una entrega más rápida de datagramas, aunque no la garantía de la entrega.

El protocolo empleado en el modelo *OSI* para llevar a cabo el encaminamiento de los paquetes se llama *"Protocolo de red no orientado a conexión", (CLNP, Conectionless Network Protocol), CLNP* es el equivalente *OSI* de *IP*

El direccionamiento lógico que se implemento como estándar para el modelo *OSI* fue el de *"Punto de acceso al servicio de red" (Network Service Access Point, NSAP),* el cual será descrito con mayor detalle en el capitulo 2, sección *"Direccionamiento NSAP"*.

Capa de enlace

El término empleado para la descripción de la información en esta capa es *"trama"*.

Esta capa tiene responsabilidades como la transmisión y recepción de tramas y establecer el direccionamiento físico de los dispositivos empleados para la comunicación de datos.

En esta capa se agrega un encabezado en la parte frontal y un trailer de cuatro bytes en el final de cada trama antes de llevar a cabo la transmisión. Para de ese modo formar una trama alrededor de los datos. Solo en la capa de enlace se agrega un trailer a los datos.

La capa de enlace tiene las siguientes características y funciones:

- *Controlar el acceso al medio.*
- *Agrega direcciones de origen y destino al hardware.*
- *Asume la función de enviar y recibir datos sobre el cable.*
- *Calcula el CRC (Cyclic Redundancy check), también mencionado como el FCS (Frame Sequence Check).*
- *Bridges y Switches trabajan en esta capa.*

La capa de enlace prepara las tramas para la transmisión, insertando los datagramas o paquetes recibidos de la capa de red dentro de tramas.

Para llevar a cabo la tarea de poner datos en el cable y quitarlos antes de transmitir los datos se debe preparar la información recibida de la capa de red y armar la trama que permita su transmisión. Este proceso incluye agregar un encabezado que incluye la dirección física (MAC) origen y destino. Además un trailer que es agregado al final de la trama. Este trailer es referido como un CRC o FCS (Figura 2.2).

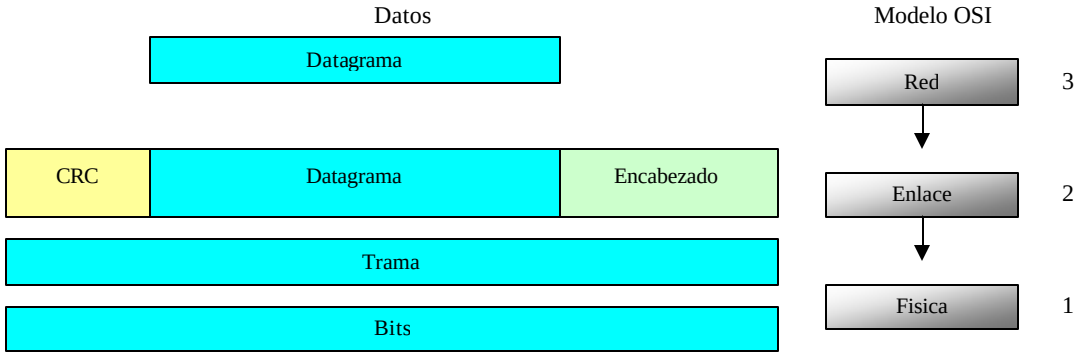


Figura 2.2

En la recepción, esta capa debe almacenar la trama, identificar su dirección, verificar el *CRC* y si todo esta correcto pasa la trama a la capa superior. Si el *CRC* no esta bien, el datagrama es descartado y no se realiza alguna otra acción.

Las direcciones físicas (mejor conocidas como direcciones *MAC*) son grabadas en cada interfaz de red y es suministrada por los fabricantes de dichas tarjetas. Cada dirección es de 6 bytes o 48 bits de longitud como fue especificado por el Instituto de Ingenieros Eléctricos Electrónicos *(IEEE).* Los primeros tres bytes representan un código, para identificar al fabricante, asignado por la *IEEE* y los siguientes tres bytes

los asigna el fabricante (Figura 23). Los dispositivos deben tener la capacidad de identificar esas direcciones.

.

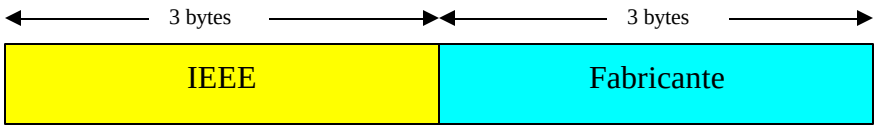


Figura 2.3

Ejemplo de dirección MAC: *000.32.09b948*

Capa Física

El término usado para describir la información en esta capa es el *"Bit"*. En la capa física los datos son representados por *1`s* y *0`s*. Los bits son codificados en pulsos eléctricos o pulsos de luz.

El objetivo principal de esta capa es transmitir los bits por un canal de comunicación y garantizar la entrega de la información sin alteración alguna.

Las principales funciones de esta capa son:

- *Definir las características físicas y eléctricas de los dispositivos que serán utilizados para la transmisión de datos.*
- *Definir las características funcionales de la interfaz*
- *Transmitir el flujo de bits a través del medio.*
- *Especificar cables, conectores y componentes de interfaz.*
- *Garantizar la conexión, aunque no la fiabilidad.*
- *Los Hubs trabajan en esta capa.*

De manera general, en la siguiente figura podemos ver cual es el funcionamiento para la transmisión de información tomando como referencia el modelo *OSI* (Figura 2.4).

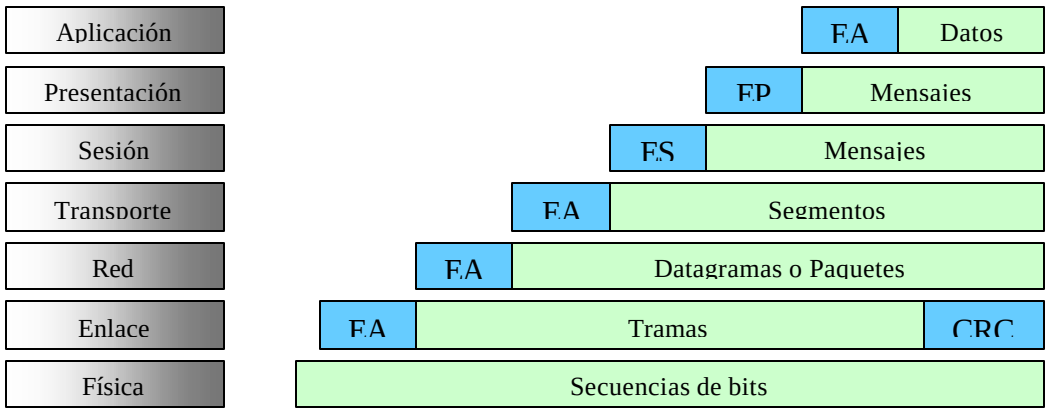


Figura 2.4

Protocolo CLNP

Este protocolo proporciona el servicio no orientado a conexión en la capa de transporte, de la misma forma que lo hace *IP*. *CLNP* proporciona mecanismos de fragmentación *(data unit identification, fragment/total length y offset).*

A continuación presentamos el formato del paquete *CLNP* (Figura 2.5) y la descripción de todos sus campos.

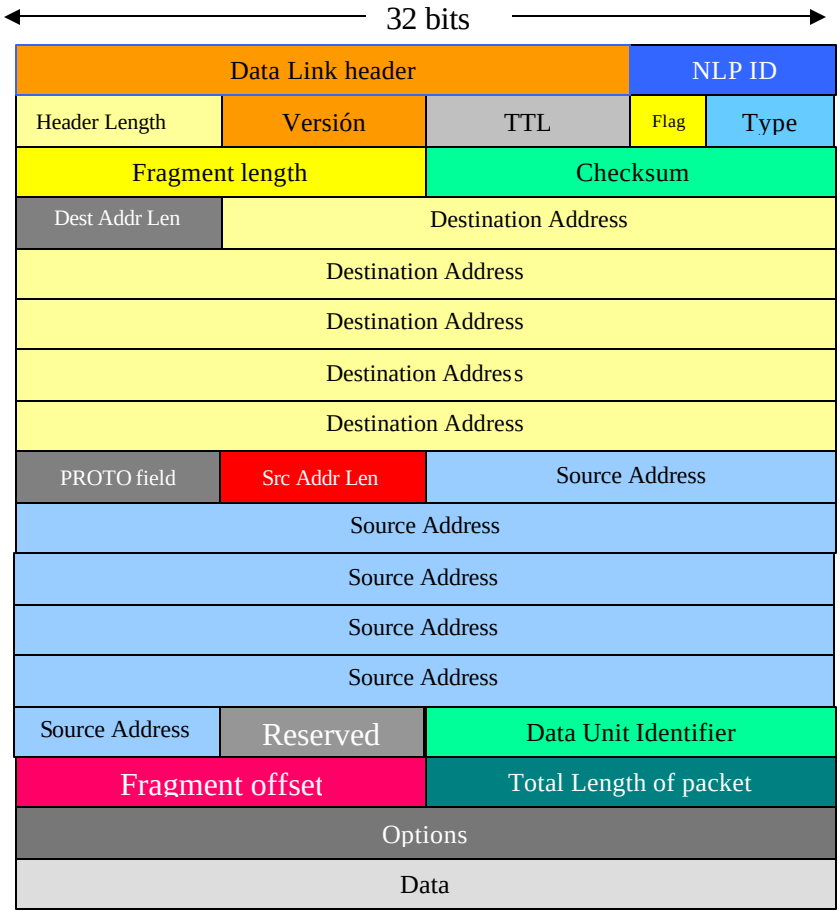


Figura 2.5

Network Layer Protocol Identifier (NLP ID) – Este un campo de 1 octeto se utiliza para identificar el protocolo de la capa d e red, el *ID* de *CLNP* es 1000 0001.

Header Length – Este campo indica la longitud del encabezado del paquete en octetos y es para señalar el comienzo de los datos.

Version – Este campo de un octeto identifica la versión del protocolo y debe ser puesto en el valor de 0000 0001.

LifeTime (Time to Live) – Este campo indica el tiempo máximo que el datagrama es permitido para permanecer en el Internet. Un datagrama no debe ser enviado con el valor de cero en este campo ya que de lo contrario es destruido. El tiempo es medido en unidades de 500 milisegundos

Flags – Hay tres tipos definidos de "*flags*", este campo ocupa 3 bits (Figura 2.6) y están distribuidos de la siguiente manera:

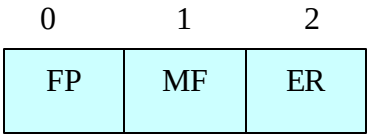


Figura 2.6

Fragment Permited (FP) – Cuando fragmentación es permitida, el valor es puesto en uno, y cuando el valor es puesto a cero, los campos "*Data Unit Identifier, Fragment offset y Total length*" no están presentes. Esto significa un solo fragmento del datagrama y la longitud total del fragmento es la longitud total del datgrama.

More Fragments (MF) – Cuando este campo tiene el valor de 1, indica que más datagramas continúan. Y cuando el valor es igual a 0, indica que es el último datagrama.

Error Report (ER) – Este campo es utilizado para suprimir la generación de un mensaje de error que haya sido detectado durante la transmisión de *datgramas*. Cuando este campo tiene el valor de 1 significa que la *PC* que genero el datagrama desea recibir todo lo concerniente a este tipo de errores.

Type – Este campo distingue los paquetes de datos *CLNP* de los reportes de error (Figura 2.7). y puede tener las siguientes opciones para la codificación:

3	4	5	6	7	
1	1	1	0	0	Datos
0	0	0	0	1	Reporte de error
1	1	1	1	0	Requerimiento de eco
1	1	1	1	1	Respuesta de eco

Figura 2.7

Fragment Length – Este campo contiene el valor de la longitud del fragmento en octetos, es decir la longitud del encabezado y los datos.

Checksum – Es el campo de revisión cuando un paquete es fragmentado y debe ser verificado en cada punto en que sea procesado el paquete.

Destination Address Length – Este campo indica la longitud de la dirección destino en octetos.

Destination Address – Este campo contiene la dirección NSAP. El octeto final de la dirección destino debe contener los valores del campo PROTO.

Protocol – Los 8 bits de este campo indican el protocolo que será usado en el campo de los datos..

Source Address Length – Este campo indica la longitud de la dirección origen en octetos.

Source Address – Este campo contiene la dirección *NSAP*. El octeto final de la dirección origen esta reservado. Puede ser utilizado para el valor del campo de protocolo en la transmisión y será ignorado en la recepción (el valor de cero no debe ser usado).

Data Unit Identifier – Este campo de 16 bits es usado para distinguir los segmentos de los paquetes originales, esto es para propósitos de reensamblamiento, este campo esta presente cuando se ha llevado a cabo la fragmentación.

Fragment Offset – Este campo de 16 bits es usado para identificar la posición relativa de los datos en este fragmento con respecto al comienzo del envió de datos.

Total Length – Este campo especifica la longitud completa del paquete original, incluyendo el encabezado y los datos. Este campo no debe ser cambiado en ningún fragmento del paquete original. Este campo esta presente solo cuando la fragmentación se ha realizado.

Options – Este campo se emplea para manejar características como:

- *Seguridad.*
- *Tipo de servicio.*
- *Padding.*
- *Origen de enrutamiento.*
- *Registro de ruta.*
- *Timestamp.*
- *Reporte de error y manejo de control de mensajes*.

Direccionamiento CLNP

El protocolo *CLNP* utiliza el formato de direcciones *NSAP (Network Service Access Point*, Figura 2.8), las cuales funcionan para la misma identificación y ubicación como una dirección IP, mas el seleccionador de protocolo con jerarquía adicional.

Las implementaciones de *CLNP* deben manejar direcciones de longitud variable arriba de 20 octetos, el formato de la dirección *NSAP* es el siguiente.

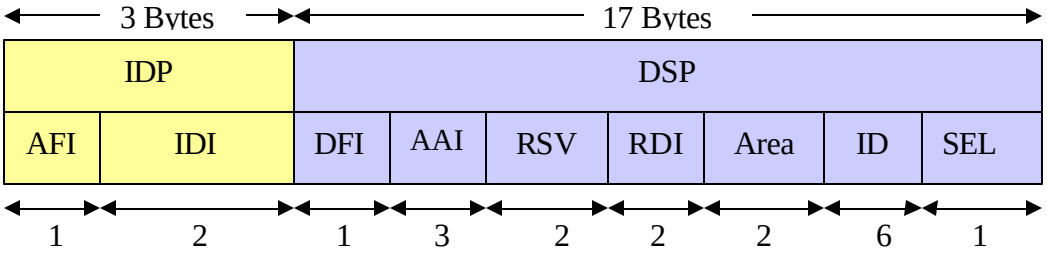


Figura 2.8

- *IDP (Inicial Domain Part).*
- *AFI (Authority and Format Identifier).*
- *IDI (Initial Domain Identifier).*
- *DSP (Domain Specific Part).*
- *DFI (DSP Format Identifier).*
- *AAI (Administration Autority Identifier).*
- *RSV (reservado).*
- *RDI (Routing Domain Identifier).*
- *Area (Area Identifier).*
- *ID (System Identifier).*
- *SEL (NSAP Selector).*

IDP.– Es la parte inicial del dominio.

AFI. – Este identificador es proporcionado por organismos internacionales y puede tener el valor de 49 en hexadecimal, cuando es una red privada y para equipos Cisco es permitido utilizar el valor de 47.

IDI – Este campo consta de 16 bits e identifica el dominio donde se encuentre la dirección.

DSP. – Es la porción especifica de la dirección.

DFI. – Especifica el formato del campo DSP y consta de 8 bits.

AAI. – En este campo se especifica el valor de la autoridad de administración, consta de 24 bits este campo.

RSV. – Este es un campo reservado para uso futuro.

RDI. – Este campo identifica el dominio de enrutamiento y consta de 16 bits.

Area – Este campo identifica el área de enrutamiento y consta de 16 bits.

ID. – Este campo es un identificador del sistema, consta de 48 bits.

SEL.– Este campo identifica cuando la dirección es utilizada por un *router* o por una PC, consta de 8 bits y tiene un valor de *00* para un *router* y cuando se una para una *PC* se omite dicho valor.

El formato de las direcciones *NSAP* es en *"Hexadecimal"*. El sistema hexadecimal comprende valores desde el número 0 hasta la letra E y por esta circunstancia existe un amplia gama de valores.

Aunque este es el formato general, como se había mencionado, es de longitud variable y se propusieron otros formatos alternativos, a continuación se presentan los otros formatos, empleando ejemplos de direcciones *NSAP* :

a. *47.0007.0000.3090.c7df.00*
b. *47.0004.0007.0000.3090.c7df.00*
c. *47.0005.80.0000a7.0000.ffdd.0007.0000.3090.c7df.00*

En el ejemplo a tenemos, figura 2.9:

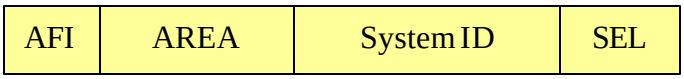


Figura 2.9

AFI=47, Area=0007, ID=0000.3090.C7DDF y SEL=00.

En el ejemplo b tenemos, figura 2.10:

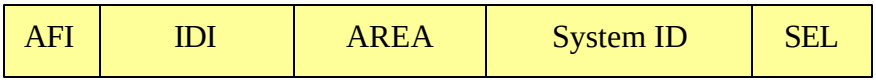


Figura 2.10

AFI=47, IDI=0004, Area=0007, ID=0000.3090.C7DDF y SEL=00.

Y finalmente en el ejemplo c tenemos, figura 2.11:

Figura 2.11

AFI=47, ICD=0005, DFI=80, AAI=0000a7, RSV=0000, RDI=FFDD, Area=0007, ID=0000.3090.C7DDF y SEL=00.

Modelo TCP/IP

A principios de los 70´s, el departamento de la defensa de Estados Unidos de América *(DoD)* por medio de la Agencia de proyectos de investigación avanzada *(ARPA)* se dedico a investigar un modelo de interconexión de redes que a la postre sería el modelo de referencia TCP/IP.

La idea principal para el desarrollo de este modelo era mantener la comunicación establecida entre origen y destino a pesar de que algún dispositivo intermedio fallara.

Este departamento fue de los primeros en implementar este modelo y por lo tanto también fue de los primeros en encontrar la necesidad de tener servicios universales y conectar entre si múltiples redes de manera transparente.

El modelo del departamento de la defensa consiste en cuatro capas funcionales (Figura 2.12); aplicación, transporte, Internet y acceso a la red. Cada una de las cuales tiene distintas responsabilidades. Las capas en el modelo definen protocolos, funciones de *hardware* y la interacción de las capas.

Figura 2.12

Capa de aplicación. – Esta capa incluye la capa de presentación y de sesión del modelo OSI. Incluye todos los procesos que involucren la interacción del usuario

Capa de transporte – Proporciona la transferencia de datos de extremo a extremo, asegurando que los datos lleguen en el mismo orden que han sido enviados, y sin errores. Esta capa puede incluir mecanismos de seguridad.

Capa de Internet. – Permite que los datos cruzar distintas redes interconectadas desde un origen hasta un destino mediante el direccionamiento lógico..

Capa de acceso a la red. – Esta capa incluye la capa de enlace y la capa física del modelo *OSI.* Define las características del medio físico *(hardware)* y es responsable del intercambio de datos entre sistemas conectados.

Descripción del modelo TCP/IP

A continuación presentamos una descripción mas detallada de cada una de las capas del modelo *TCP/IP*, incluyendo protocolos y algunas especificaciones de hardware.

Capa de aplicación

La parte mas alta de *TCP/IP* es la capa de aplicación. Esta capa incluye todos los procesos que usen los protocolos de la capa de transporte para entregar datos a la capa de Internet. Hay muchos protocolos de aplicación y son agregados nuevos protocolos frecuentemente. Los más ampliamente conocidos e implementados son:

- *TELNET (Protocolo de terminal de red).*
- *FTP (Protocolo de transferencia de archivos).*
- *SMTP (Protocolo de transferencia de correo electrónico).*
- *POP – 3 (Protocolo de oficina postal).*
- *DNS (Servicio de dominio de nombres).*
- *HTTP (Protocolo de transferencia de hipertexto).*
- *SMNP (Protocolo de administración de red).*

Capa de transporte

La capa de transporte cuenta con dos protocolos principales; El protocolo de control de transmisión *(TCP)* y el protocolo de datagramas de usuario *(UDP).*

TCP es un protocolo basado en conexión que provee detección y corrección de error con entrega confiable de paquetes mientras que *UDP* es un protocolo no orientado a conexión y con poca sobre carga de paquetes.

Un *software* de aplicación elige *TCP* o *UDP* basándose en si el asunto es importante o no, o que grado de importancia tiene para crear una conexión confiable con comunicación bi-direccional y administración de error, o si es mas importante implementar un bajo sobre encabezado para la transmisión de la aplicación.

Capa de Internet

La capa de Internet se ubica por encima de la capa de acceso y debajo de la capa de transporte. La principal tarea de esta capa es manejar la conexión de redes interconectadas para llevar a cabo la entrega de paquetes desde origen hasta el destino.

El protocolo encargado de hacer este tipo de tareas se llama: *Protocolo de Internet o Internet Protocol*, mejor conocido como IP por las siglas en ingles.

Las funciones que dicho protocolo desempeña son:

- *Definir un datagrama y un esquema de direccionamiento.*
- *Mover datos entre la capa de transporte y la de acceso a la red.*
- *Fragmentación y reensamblamiento de datagramas.*
- *Realizar operaciones de enrutamiento.*
- *Resolución de congestionamientos.*
- *Resolución de caídas de rutas.*

El otro protocolo utilizado en la capa de Internet es el Protocolo de control de mensajes de Internet *(ICMP*), un protocolo usado para comunicar mensajes de control en sistemas IP.

Cabe destacar que en esta capa se desempeña la tarea de encaminamiento o enrutamiento de paquetes, la cual recae en equipos llamados *"routers"* y es una parte fundamental de la comunicación a grandes distancias.

Capa de acceso a la red

Esta capa esta ubicada en el nivel mas bajo dentro de la jerarquía del modelo *TCP/IP*. Muchas veces es ignorada por los usuarios.

Las funciones desempeñadas en la capa de acceso son:

- *Encapsulación de datagramas en tramas, para ser transmitidas por la red.*
- *Mapeo de direcciones IP a direcciones físicas de hardware (MAC).*

Mucho del trabajo que se ubica en esta capa de acceso es manejado mediante software de aplicación y controladores que son únicos para las piezas individuales de hardware. La configuración consiste en simplemente la selección correcta del controlador y seleccionar TCP/IP como el protocolo a usar. Muchas computadoras ya vienen con sus controladores cargados y configurados los equipos o se pueden configurar ellos mismos mediante aplicaciones de *"plug-and-play"*.

En esta capa se encuentran ubicados para su utilización los estándares del Instituto de Ingenieros Electrónicos y Eléctricos *(IEEE)* para Ethernet, Token ring, X-25, etc.

Protocolo IPv4

El *Internet Protocol* o *IP* es el protocolo mas conocido dentro del modelo de referencia *TCP/IP*. Este protocolo proporciona una rápida pero no confiable intercambio de paquetes o datagramas de origen a destino. Proporciona la dirección y el mecanismo de entrega para todo el tráfico relacionado con *TCP/IP*. *IP* no es por si mismo confiable por lo que permite a los protocolos de capas superiores manejar esos aspectos de la transmisión.

Este protocolo de Internet permite llevar cabo tareas como:

- *Entrega de paquetes*
- *Direccionamiento lógico*
- *Fragmentación y reensamblamiento*

IP ha sufrido modificaciones desde su creación, actualmente esta en funcionamiento la versión del protocolo número 4, mejor conocida como *"IPv4"*. La versión más reciente es la versión 6, conocida como *"IPv6 o IPng"*, que será descrita posteriormente.

Este protocolo es un servicio de red no orientado a conexión, es un servicio de entrega de paquetes conocido como el del mejor esfuerzo, también conocido como *"Best-effort"*, que significa que *IP* no proporciona una forma de revisar la entrega. *IP*, este servicio asume la confiabilidad de las capas superiores y hace su mejor esfuerzo para transmitir los datos a su destino, pero sin ninguna garantía.

IP proporciona mecanismos para el manejo de paquetes muy grandes, mediante el uso del método de la fragmentación y ensamblamiento.

TCP/IP desarrollo un esquema de direccionamiento lógico para la capa de red, el cual permite asignar direcciones lógicas a los sistemas *(hosts*, *servers*, *routers*, etc.), a diferencia de las direccio nes físicas o direcciones *MAC*.

Esas direcciones facilitan la identificación y ubicación de equipos dentro y entre redes. Las direcciones son dependientes del protocolo y varían dependiendo del protocolo implementado en la capa de red.

Para comprender el funcionamiento de *IP*, es necesario describir el formato de su paquete y su direccionamiento.

Formato de paquete y descripción de los campos

En la figura 2.13 podemos ver el formato del paquete del protocolo *IP* versión *4*

<table>
<tr><td>Version</td><td>IHL</td><td>Type of Service</td><td colspan="2">Total Length</td></tr>
<tr><td colspan="3">Identification</td><td>Flags</td><td>Fragment Offset</td></tr>
<tr><td colspan="2">TTL</td><td>Protocol</td><td colspan="2">Header Checksum</td></tr>
<tr><td colspan="5">Source Address</td></tr>
<tr><td colspan="5">Destination Address</td></tr>
<tr><td colspan="4">Options</td><td>Padding</td></tr>
</table>

Figura 2.13

Version – Este campo identifica la versión actual del protocolo de Internet *(IP)* y emplea cuatro bits.

IHL (Internet Header Length) – Este campo (longitud del encabezado de Internet) identifica el tamaño del encabezado IP y esta constituido por cuatro bits.

Type of Service (ToS) – Este campo define el tipo de servicio que puede ser usado para la entrega de datos basándose en requerimientos de las aplicaciones tales como: *reliability, low delay, througput y cost.*

Este campo comprende 8 bits, pero esta distribuido en tres partes, las cuales se muestran en la figura 2.14

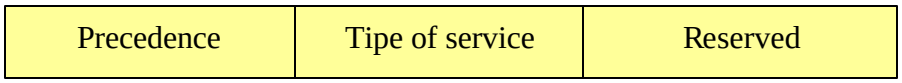


Figura 2.14

Precedence – Este campo es de 3 bits y es usado para proporcionar 8 niveles de de prioridad en el reenvió de paquetes en los *routers*,

Los datagramas con prioridad más alta, por ejemplo 5, son reenviados antes que los datagramas con prioridad mas baja, por ejemplo 4.

Este esquema de prioridad permite a los administradores manipular el reenvió de paquetes individuales salto a salto, dándoles prioridad en tráficos específicos. Los valores de prioridad están en el rango de 0 a 7, siendo 7 el valor más alto y 0 el más bajo.

A continuación, en la tabla 2.1, se establecen los diferentes tipos de prioridad, aunque algunos no has sido establecidos completamente.

Codigo	Prioridad
111	Network Control
110	Internetwork Control
101	Critical
100	Flash Override
011	Flash
010	Inmediate
001	Priority
000	Routine

Tabla 2.1

Type of Service – Este campo es de 4 bits e identifica el nivel del tipo de servicio que recibirá el paquete por el router.

Los bits de *ToS* deben ser reconocidos por el *router* receptor para reenviar los datagramas, cuando existen múltiples trayectorias, a lo largo de toda la trayectoria es deseable mantener el mismo nivel de servicio.

Por ejemplo cuando existen múltiples trayectorias a un mismo destino con características variables, tales como la capacidad del enlace y tazas de transferencia, los *routers* deben seleccionar la mejor trayectoria posible para llevar a cabo el reenvió.

Las aplicaciones pueden ayudar a los *routers* en esta selección para configurar uno de los bits que indican al *router* su preferencia por algún nivel particular de tipo de servicio. Con esta información el *router* reenviara el tráfico a lo largo de todo el camino con el nivel de tipo de servicio deseado.

En la tabla 2.2 podemos encontrar el valor numérico, en bits, asignado a cada tipo de servicio.

Codigo	Tipo de Servicio
1000	Low delay
0100	Throughput
0010	Reliability
0001	Cost
0000	Normal Service

Tabla 2.2

Reserved – Este campo esta reservado para uso futuro.

Total Length – Este campo identifica la longitud total del datagrama y esta compuesto por 16 bits.

Identification – Este campo de identificación esta compuesto por 16 bits y es usado por *IP* para la fragmentación y el ensamblamiento de datagramas. El *router* asigna a cada datagrama in identificador *(ID)*. Los datagramas relacionados tienen el mismo *ID*

Este *ID* permite al usuario identificar los datagramas que pertenecen al mismo envió de datos, pues los datagramas pueden ser enviados mediante diferentes rutas y ser recibidos en orden distinto, es importante que el receptor pueda identificar los datagramas relacionados al mismo *ID*. El uso del *ID* le permite al receptor reensamblar cualquier datagrama recibido independientemente del orden en que haya llegado.

Flags – Este campo esta formado por 3 bits y tendrá al menos una de cuatro opciones, las primeras dos son usadas para controlar la fragmentación de datagramas en los *routers*. Las últimos dos trabajan en conjunción con el campo de identificación para indicar si este es el único datagrama o pertenece a un flujo de datos con el mismo *ID*.

Fragment Offset – Este campo esta formado por 13 bits, e indica el sitio al cual pertenece el datagrama dentro de la transmisión general. Este valor el asignado por router origen.

TTL (Time to Live) – Este campo esta compuesto por 8 bits.El valor del TTL (Tiempo de vida) es un contador representado en segundos y determina el máximo tiempo de vida que un datagrama puede existir en una red.

Los *routers* decrecen este valor en al menos 1 segundo basándose en cuanto le toma al *router* procesar y reenviar un datgrama, cuando el *TTL* alcanza el valor de cero, los *routers* descartan el datagrama aunque el reenvió de los *routers* nunca tarda mas de in segundo, las reglas dictan que los *routers* deben decrecer este valor en 1 cuando procesen y reenvíen datagramas. Pro lo tanto cada segundo de *TTL* puede ser asociado con un simple salto.

El rango del valor de *TTL* es de 0 a 255 segundos.

Protocol – Este campo esta compuesto de 8 bits e identifica el protocolo que utilizara en la capa superior, el cual puede ser *TCP* o *UDP*. Los valores de este campo para seleccionar *TCP* es de *06 (00000110)* y para *UDP* es de *17 (00010001).*

Header Checksum – Este campo esta formado por 16 bits. IP emplea este campo para revisar que el encabezado no hay sido dañado durante la transmisión de algún paquete.

Source Address – Este campo esta constituido por 32 bits e identifica a la dirección lógica IP del origen .

Destination Address – Este campo esta constituido por 32 bits e identifica a la dirección que sera enviado el paquete.

Options – Este campo longitud variable y depende de la aplicación que sea requerida, las cuales pudiesen ser :

- *Seguridad.*
- *Origen de encaminamiento*
- *Timestamp*
- *Registro de ruta*
- *Identificación de secuencia*

Padding – Este campo es de longitud variable y se utiliza como relleno para el campo de opciones, para que este campo sea múltiplo de 4 bytes.

Direccionamiento IPv4

En el modelo *TCP/IP* se emplea un direccionamiento lógico, el cual tiene como objetivo fundamental establecer la mejor forma en que un paquete pueda llegar a su destino.

Este direccionamiento establece una jerarquía de dos bloques principales, figura 2.15, que es el bloque de red y el bloque del usuario.

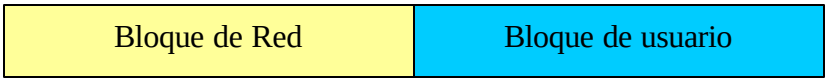


Figura 2.15

El bloque de red se utiliza para identificar a la red, también llamado *Net-ID* y el bloque de usuario es para identificar al usuario, también conocido como *USER-ID* o *HOST-ID.*

El bloque de usuario identifica a un dispositivo individual que se encuentra dentro de una red. Y el bloque de red es asignado a para identificar a compañías o instituciones.

Este sistema se entiende como el direccionamiento de una ciudad:

La dirección de red o bloque de red es como si fuera el nombre de la calle y el bloque de usuario o dirección de usuario es como si fuera el número de una casa en particular,

Bajo este formato, los dos bloques de direcciones antes mencionados no tienen una longitud fija, lo cual serà analizado y descrito en mayor detalle mas adelante.

El formato de las direcciones *IP* esta constituido por 32 bits, agrupados en 4 bytes y separados por un punto, asi se obtiene el siguiente formato general:

Dirección IP = A.B.C.D

Donde A, B, C y D representan 1 byte, entonces cada byte tendrá un valor mínimo de 0 y un máximo de 255, que en binario estaría representado por: 0=00000000 y 255=11111111.

Tipos de dirección

Cuando se creo el modelo de direccionamiento *IP* se establecieron dos bloques principales; *Net-ID (bloque de red)* y *User-ID (bloque de usuario).*

Con la creación del formato de dirección, también aparecieron 5 tipos de direcciones, las cuales están basadas en el tamaño del campo asignado para la red *(Net–ID).* A esta clasificación de direcciones se le llamo *"con clase o classful"*.

De los 5 tipos de direcciones, las 3 primeras son de uso para el público de Internet (A, B y C), la clase D es utilizada para la multidifusión o *multicast* y la clase E esta reservada.

A continuación se muestra esquemáticamente la clasificación, figura 2.16.

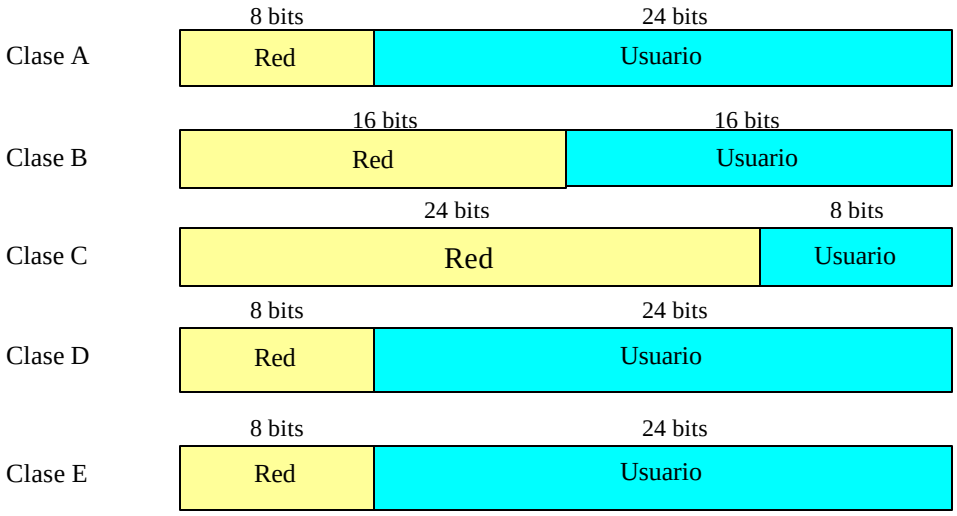


Figura 2.16

La designación de bits dentro del primer byte es la que va indicando a que clase pertenece la dirección.

Las direcciones de clase A comienzan con el *bit* "0" dentro del primer octeto, y los siguientes 7 bits para asignar las direcciones de red clase A, haciendo posible el rango de 0 a 127 (00000000 - 01111111) el cual se muestra en la figura 2.17, sin embargo el valor de 127 esta reservado para direcciones de *loopback*.

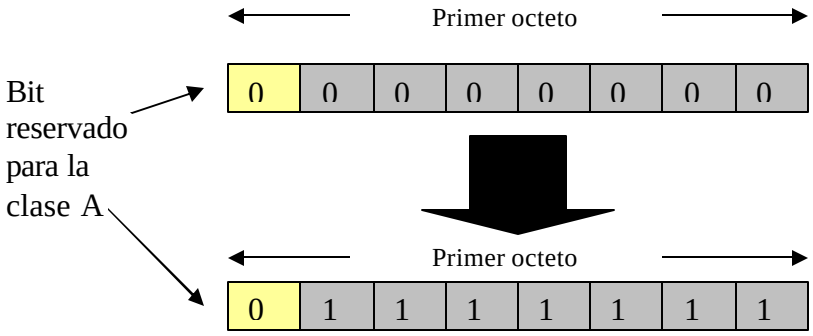


Figura 2.17

Los siguientes 24 bits son utilizados para definir a los usuarios o *host*.

El formato completo de una dirección clase A es presentado en la figura 2.18.

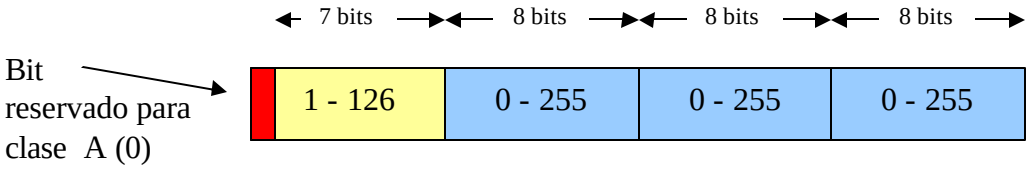


Figura 2.18

Las direcciones de clase B comienzan con los dos primeros bits en *"10"* dentro del primer octeto, y los siguientes 6 bits del primer octeto y los 8 del segundo octeto para asignar las direcciones de red clase B, haciendo posible el rango de 128 a 191 (10000000 - 10111111) en el primer octeto, el cual se muestra en la figura 2.19 y de 0 a 255 en el segundo octeto..

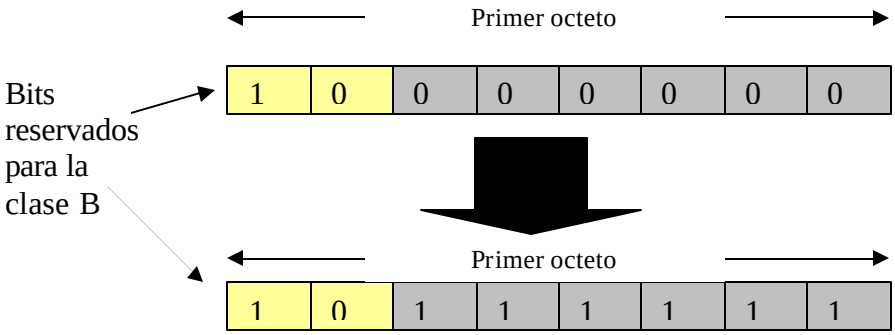


Figura 2.19

Los últimos 16 bits son usados para definir a los usuarios o hosts

El formato completo de una dirección clase B es presentado en la figura 2.20.

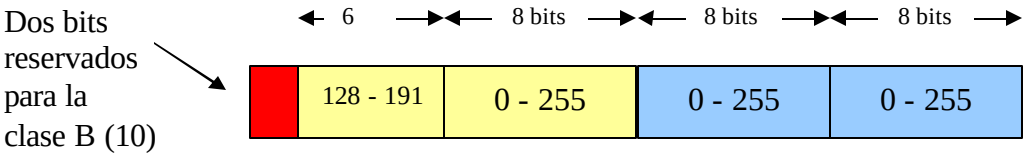


Figura 2.20

Las direcciones de clase C comienzan con los tres primeros bits en "110" dentro del primer octeto, y asigna los siguientes 5 bits del primer octeto, los 8 del segundo octeto y los 8 del tercero para las direcciones de red clase C, haciendo posible el rango de 192 a 223 (11000000 - 11011111) en el primer octeto, que se muestra en la figura 2.21, de 0 a 255 en el segundo octeto y de 0 a 255 en el tercer octeto.

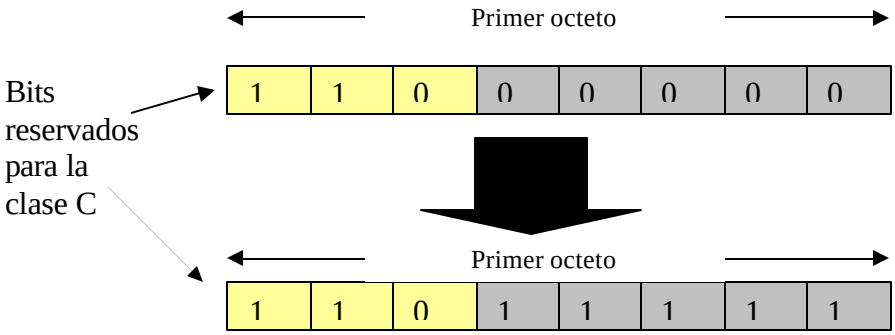


Figura 2.21

Los últimos 8 bits son utilizados para definir a los usuarios o *hosts*.

El formato completo de una dirección clase C es presentado en la figura 2.22

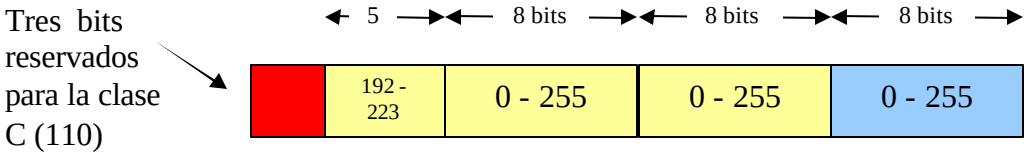


Figura 2.22

Las direcciones de clase D comienzan con los cuatro primeros bits en "1110" dentro del primer octeto. Y los demás bits (27) quedan disponibles son utilizados para formar grupos de multidifusión y no para usuarios particulares dentro de una red.

El rango de direcciones clase D es de 224 a 239 (11100000 - 11101111) en el primer octeto, que se muestra en la figura 2.23, de 0 a 255 en el segundo octeto, de 0 a 255 en el tercer octeto y finalmente de 0 a 255 en el cuarto octeto.

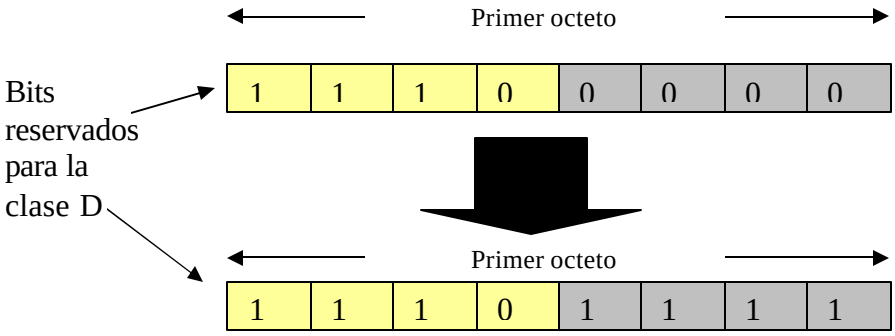


Figura 2.23

El formato completo de una dirección clase D es presentado en la figura 2.24.

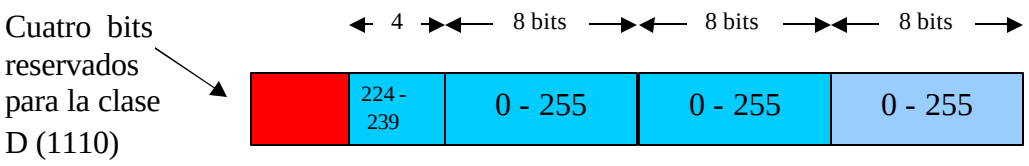


Figura 2.24

Las direcciones de clase E siempre comienzan con los cinco primeros bits en "11110" dentro del primer octeto, y los demás quedan libres. El rango de direcciones de la clase E es de 240 a 247 (11100000 - 11101111) en el primer octeto, mostrado en la figura 2.25.

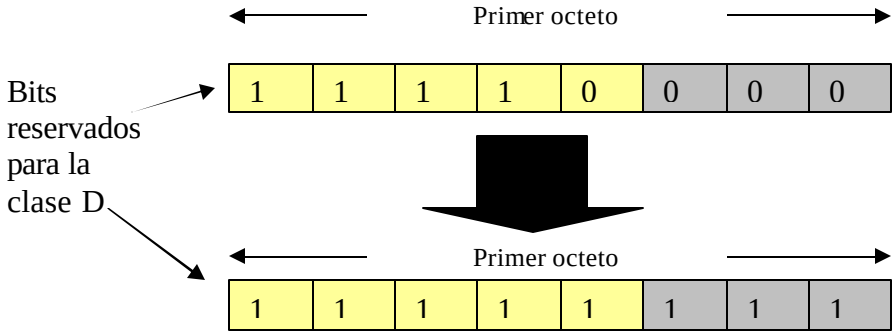


Figura 2.25

El formato completo de una dirección clase E es presentado en la figura 2.26.

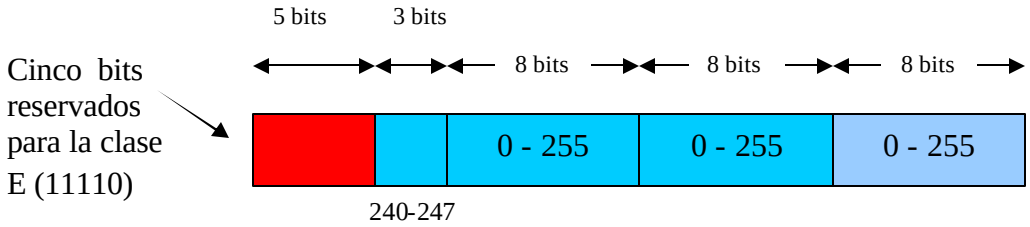


Figura 2.26

Para saber que tipo de clase de red se esta utilizando, podemos ayudarnos de la asignación de bits en el primer octeto, pero otra forma es referenciando la mascara de red que le sea asignada a un usuario.

Existen tres mascaras de red asignadas a una clase de red, que son los siguientes:

- *Clase A – 255.0.0.0*
- *Clase B – 255.255.0.0*
- *Clase C – 255.255.255.0*

De la clasificación anterior podemos obtener las características de las 3 principales clases de direcciones de red, Tabla 2.3.

Atributos	Clase A	Clase B	Clase C
Tamaño	(Diagonal /8) Es para definir un pequeño numero de redes, tales como gubernamentales, universidades, militares e instituciones de investigación en las que existen un gran numero de ususarios	(Diagonal /16) Rodea a un gran número de redes con igual número de usuarios (host), como compañías de tamaño mediano.	(Diagonal /24) Esta clase pretende abordar las necesidades de pequeñas redes proporcionando un gran número de redes con un pequeño número de usuarios (host) por red.
Designaciones de bits	El primer bit es cero (0).	Sus dos primeros bits son uno y cero (10).	Sus primeros tres bits son uno, uno y cero (110).
Valor del rango	1-126	128-191	192-223
Bytes asignados para red	1 byte	2 bytes	3 bytes
Mascara asignada por omision.	255.0.0.0	255.255.0.0	255.255.255.0
Bytes disponibles para usuarios (host)	3 bytes	2bytes	1 byte
Numero de redes	126	16,384	2,092,152
Numero de usuarios	16,777,214	65,534	254

Tabla 2.3

Además de las clases de direcciones antes mencionadas, también existen bloques de direcciones que se han reservado para aplicaciones especificas, así que existen dos tipos mas de direcciones que son:

- *Direcciones reservadas*
- *Direcciones privadas*

Direcciones reservadas

Las direcciones reservadas pueden ser utilizadas para un gran numero de propósitos especiales. A continuación se presentan los tipos de estas direcciones:.

- **255.255.255.255**. Una dirección *IP* es puesta en su totalidad en 1`s y significa que un mensaje en la red es enviada a todos los nodos y todas las redes, es usada para propósitos de amplia difusión, mejor conocida como *dirección de broadcast*.

- **0.0.0.0** Una dirección puesta en ceros representa una red o usuario desconocido y típicamente es empleada para definir la puerta de enlace por omisión o ultimo recurso *(default gateway o last resort).*

- **127.0.0.1** Esta dirección en especial es empleada para pruebas internas (*loopback*) solo designa el nodo local y no generara ningún tipo de trafico en la red.

Aunque las direcciones 255.255.255.255 son consideradas direcciones de *broadcast* en cualquier equipo, los *routers* no reenvían este tipo de mensajes a todas las redes. Los *routers* aíslan estos mensajes solo a las subredes. Para enviar los mensajes de *broadcast* a todos los usuarios de una subred se ponen todos los bits del bloque de usuario en uno.

Por ejemplo, si se quiere enviar un mensaje de *broadcast* a todos los usuarios de una red clase B 132.107.0.0 con una mascara de red estándar de 255.255.0.0, se tendría que especificar la dirección destino como: 131.107.255.255.

Direcciones privadas

Las direcciones privadas son utilizadas para las redes privadas, es decir que no serán validas globalmente, no se pueden usar en Internet. Las compañías usualmente emplean este tipo de direcciones de manera interna.

Existen tres bloques dentro de cada clase de direcciones, que son:

- *Clase A, 10.0.0.0 – 10.255.255.255*
- *Clase B, 172.16.0.0-172.31.255.255*
- *Clase C,192.168.0.0-192.168.255.255*

Subnetting

El término de *subnetting* se refiere a la división de una red mayor en varias redes más pequeñas. Los diseñadores de IP desarrollaron un esquema jerárquico para facilitar la división de redes y organización de *hosts* dentro de esas subredes, pera además, para limitar o segmentar la cantidad de tráfico en toda la red. El *subnetting* permite llevar a cabo una mejor asignación y aprovechamiento de direcciones.

Al dividir una red mayor, el tráfico que se intercambia entre dos *hosts* en el mismo segmento no afectaría a otros *hosts* en otros segmentos dentro de la misma red. Cuando los *hosts* en diferentes subredes quieren comunicarse, los *routers* facilitan el reenvió de trafico entre esas subredes. El *router* debe tener claro un mapa de las subredes para poder llevar a cabo en forma efectiva el reenvió de datagramas.

La dirección asignada es de 4 bytes o 32 bits de longitud, de los cuales en el primer byte se determina la clase, y dependiendo de la clase, se asignaran 1, 2 ò 3 bytes para el área de red, esta porción no puede ser modificada de ninguna manera. Los siguientes bits sin asignación dentro de la dirección de 32 bits podrá ser empleada en cualquier forma que el administrador lo desee, en términos de asignación de subredes y *hosts.*

Así también se diseño una formula que nos permite saber que cantidad de subredes o hosts pueden obtenerse a partir del numero de bits.

 # De subredes = 2^n
 # De hosts = 2^n

Donde n es el número de bits tomados para subredes o para *hosts.*

Las subredes necesitan un identificador de subred y una dirección de *broadcast*, la cual permitirá identificar a la subred y la dirección de *broadcast* es para reenviar trafico única y exclusivamente e esa subred. por lo cual se descuentan 2 subredes, que son la primera y la ultima, es decir que la formula quedara de la siguiente manera:

 # De subredes validas = 2^n-2 ò # de hosts = 2^n-2

Una nueva forma de describir la mascara de subred es mediante la notación ya conocida, que es *A.B:C:D* en forma decimal, se introduce el empleo de una diagonal al final de la dirección IP, es decir; *A.B.C:D/n* donde n es el numero de bits que están destinados para el *NET-ID*, incluyendo el *SUBNET-ID.*

A continuación se presenta un ejemplo para explicar el proceso de *subnetting*.

Ejemplo 2.1, figura 2.27.

Una empresa de tamaño medio esta constituida por 5 departamentos (A, B, C, D y E) y desea llevar a cabo un direccionamiento que permita una administración optima, para ello cuenta con una red clase C 200.23.60.0 con mascara 255.255.255.0.

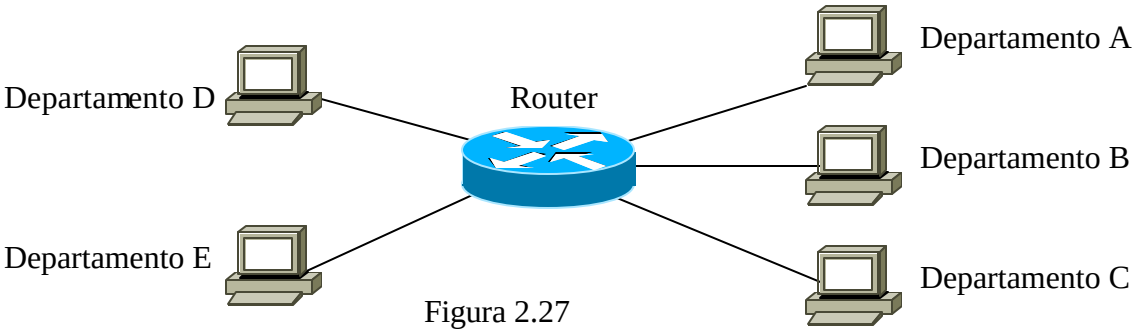


Figura 2.27

Solución:

Para cubrir la necesidad de 5 subredes, se deberán tomar 3 bits, Pero ¿porque no elegir 2 ò 4 bits? La razón es porque si elegimos 2, solo se crearían 4 subredes, las cuales resultarían insuficientes y si elegimos 4, se crearían 16 subredes, las cuales serian demasiadas.

Por lo tanto elegimos 3 bits y obtenemos la siguiente solución, ver tabla 2.4:

 # De subredes = 2^3-2 = 6 subredes validas
 # De hosts = 2^5-2 = 30 hosts .por subred

Subred	Rango de hosts	Uso
200.23.60.0/27	No utilizado	Identificador de la subred
200.23.60.32/27	200.23.60.33 – 200.23.60.63	Departamento A
200.23.60.64/27	200.23.60.35 – 200.23.60.95	Departamento B
200.23.60.96/27	200.23.60.97 – 200.23.60.127	Departamento C
200.23.60.128/27	200.23.60.129 – 200.23.60.159	Departamento D
200.23.60.160/27	200.23.60.161 – 200.23.60.191	Departamento E
200.23.60.192/27	200.23.60.193 – 200.23.60.223	No utilizada
200.23.60.224/27	No utilizado	Dirección de broadcast

Tabla 2.4

De la tabla anterior se tiene que la subred 0 será utilizada como identificador de la subred *(SUBNET-ID)*, y las subredes 1, 2, 3, 4 y 5 serán asignadas a los departamentos A, B, C, D y E. respectivamente, cada una de los cuales tendrá a su disposición 30 direcciones para hosts, mientras que la subred 6 no será empleada a ningún departamento y por lo tanto estará reservada para uso futuro. Por ultimo la subred 7 es asignada como dirección de *broadcast*.

VLSM

El empleo del *subnetting* fue utilizado para dividir una red mayor en un número mas pequeño de redes, llamándolas "*subredes*". Pero estas subredes deberían tener un mismo tamaño, es decir, si se tuviera una red *200.0.204.0* con mascara de *255.255.255.0* y se requerían 10 subredes, se tomarían 4 bits y los 4 restantes corresponderían a los hosts.

Es decir se tendría:

 # de subredes = 2^4 – 2 = 14 subredes
 # de hosts validos = 2^4 –2 = 14 hosts por subred.

Pero este esquema no siempre resolvería al *100 %* el aprovechamiento de las direcciones *IP* y una mejor administración de las mismas.

Cuando se trata de asignar direcciones a diferentes departamentos o asignar subredes de diversos tamaños, es necesario llevar a cabo el menor desperdicio de direcciones.

La técnica de asignación de subredes de diferentes tamaños es llamada enmascaramiento de subred de longitud variable (*Variable Length subnet masking, VLSM).*

La idea de *VLSM* es dividir la red y entonces volver a crear una nueva división, es decir crear subredes de las subredes. Esto puede hacerse tantas veces como sea necesario, y depende de cuantos bits se tengan disponibles en el área de *HOST-ID.*

Para comprender mejor este concepto, se presenta el ejemplo 2.2:

Ejemplo 2.2

Se tiene un departamento que tiene a su cargo una red clase C 200.0.204.0/24 y necesita tener 5 subredes *(A, B, C, D y E)* con el siguiente número de *hosts* respectivamente: 60, 60, 60, 30, 30. el administrador del departamento no puede usar una mascara de subred con 2 bits, ya que este solo permitiría 4 subredes con 62 *hosts.* Tampoco puede usar una mascara de subred con 3 bits porque esta crearía 8 subredes con 30 *hosts* cada una.

Para resolver este problema, el administrador usara el método de VLSM. Es decir, creara en un principio 4 subredes, empleando 2 bits para ello. Ver tabla 2.5.

 # De subredes = 2^2 = 4 subredes
 # De hosts = 2^6 – 2 = 62 hosts por subred

Subred	Rango de hosts	Uso
200.0.204.0	200.0.204.1 – 200.0.204.63	Subred A
200.0.204.64	200.0.204.65 – 200.0.204.127	Subred B
200.0.204.128	200.0.204.129 – 200.0.204.191	Subred C
200.0.204.192	200.0.204.193 – 200.0.204.255	VLSM

Tabla 2.5

Como se aprecia en la tabla 2.4, ase satisfacen los requerimientos de las 3 primeras subredes *(A, B y C),* ya que se les proporciona una cantidad de 62 *hosts* aun cuando estas requieren solo 60 y a la ultima, se le aplicara el método de *VSLM*.

Pero ahora para satisfacer a las redes D y E se empleara el método de *VLSM*: Se lleva a cabo un nuevo *subnetting*, es decir subdividir lo ya dividido.

Se tiene una subred clase C *200.0.204.192/26* y se requieren de ella 2 sub-subredes con 30 hosts cada una. Por lo tanto se elige 1 bit para llevar acabo el *subnetting*, figura 2.28.

Figura 2.28

Donde:

- *Net-ID* es el espacio natural reservado para las redes clase C (24 bits).

- *S* es el espacio tomado para aplicar el *subnetting* (2 bits).
- *A* es el espacio tomado para aplicar *VLSM* o *sub-subnetting* (1 bit).
- *Host-ID* es el espacio reservado para hosts en el primer *sub-subnetting* (5 bits).

 # De subredes = 2^1 = 2 subredes.
 # De hosts = 2^5 -2 =30 hosts por subred.

Quedando la asignación de la siguiente manera, ver tabla 2.6:

Subred	Rango de hosts	Uso
200.0.204.192/27	200.0.204.193 – 200.0.204.223	Subred D
200.0.204.224/27	200.0.204.225 – 200.0.204.254	Subred E

Tabla 2.6

De la aplicación de *VLSM* podemos resumir el procedimiento a la figura 2.29 como referencia de la distribución de las subredes.

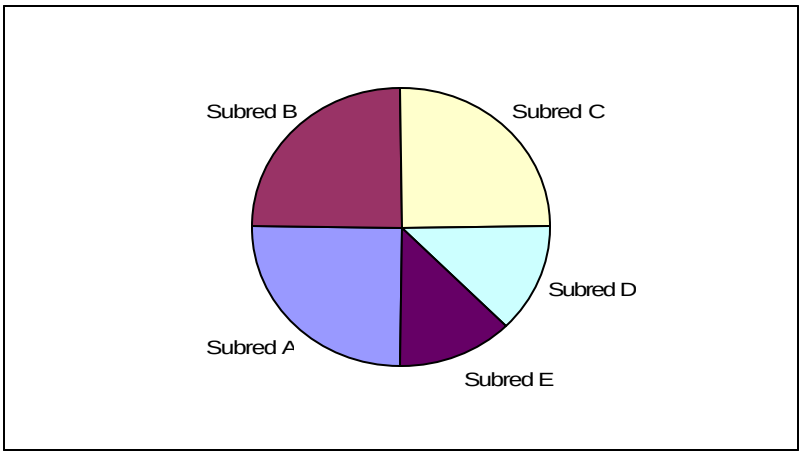


Figura 2.29

CIDR

A finales de los 80`s surgió, como consecuencia del *subnetting* y *VLSM*, un crecimiento acelerado en la cantidad de redes y subredes en el mundo de Internet.

El conflicto que comenzó a ocasionar fue que al haber tantas subredes, estas debían ser anunciadas de manera individual, lo que traería un crecimiento en las tablas de enrutamiento de los *routers*, ocasionando un consumo excesivo de procesamiento y memoria.

A principios de los años 90 se desarrollo un esquema, para contrarrestar el crecimiento de las tablas de enrutamiento, el cual fue llamado *Supernetting*, *Sumarización*, *Agregación* o *CIDR (Classless Inter-Domain Routing).* Cualquier término que se decida utilizar, establece un método para reducir el tráfico en la actualización de rutas y de las tablas de enrutamiento.

Este método establece una forma más flexible para asociar grupos de direcciones *IP* contiguos o adyacentes dentro de una sola dirección que representa a un grupo de direcciones o subredes.

CIDR especifica un rango de direcciones *IP* por la combinación de una dirección *IP* y su mascara de red asociada, la notación *CIDR* utiliza el siguiente formato:

xxx.xxx.xxx.xxx/n

Donde n es el número de bits con valor de "1", contados de izquierda a derecha, de la mascara de red.

A continuación se describen dos ejemplos que emplean *CIDR.*

Ejemplo 1

S tiene la siguiente dirección *IP 132.248.120.0* con una mascara de *255.255.254.0.*

Dirección IP = 132.248.120.0/23
Mascara de red = 11111111.11111111.11111110.00000000

Que significaría tener dos subredes clase C; *132.248.120.0 y 132.248.121.0* con una mascara de red de *255.255.254.0*

Entonces al presentar o anunciar el bloque o prefijo *132.248.120.0/23* hacia Internet se dará por entendido que se están englobando las dos subredes que se obtienen de ella.

Ejemplo 2:

Y si se tuviera el bloque *IP 200.0.204.0/22*.

Dirección IP = 200.0.204.0/22
Mascara de red = 11111111.11111111.11111100.00000000

Esto significa que se tendrá una mascara de *255.255.252.0* para la red *200.0.204.0* y permitirá la creación de 4 redes clase C *200.0.204.0, 200.0.205.0, 200.0.206.0* y *200.0.207.0*, las cuales emplearan una mascara por omisión de *255.255.255.0.*

Se dijo que las subredes, para ser agregadas, deben ser contiguas en el espacio de dirección. *CIDR* no podría agregar *200.0.204.0* y *200.0.207.0* dentro de un mismo bloque, a menos que *200.0.205.0* y *200.0.206.0* sean incluidas.

La solución para este ejemplo seria, anunciar el prefijo *200.0.204.0/24*, lo que significa que se estarían anunciando las 4 subredes contiguas *(200.0.204.0, 200.0.205.0, 200.0.206.0* y *200.0.207.0),* en lugar de anunciar una por una.

Para poder llevar a cabo la implementación de este esquema, se requiere que los equipos soporten este tipo de convenciones. Hay protocolos de enrutamiento como: *BGP*, *OSPF*, *ISIS* fueron modificados para soportar *CIDR*, pero algunos otros siguen sin soportarlo.

Los *routers* en el *backbone* de Internet generalmente soportan *CIDR* en los enlaces *WAN* entre los proveedores de servicio de Internet, *ISP's*. El soporte de *CIDR* en el *backbone* es esencial para manejar tablas de enrutamiento de menor tamaño, ya que mientras mayor sea el numero de redes contenidas en una tabla de enrutamiento, mayor será el consumo de procesamiento y de memoria del equipo encargado de realizar esa tarea.

Protocolo IPv6

Esta nueva versión del protocolo *IP* nace como necesidad de corregir las deficiencias del protocolo *IPv4*, a su vez también se llevaron a cabo modificaciones en *ICMP* y en los protocolos de enrutamiento como *RIP*, *BGP* y *IS–IS*, pero las principales cambios que se implementaron en *IPv6* fueron los siguientes:

- Una dirección *IPv6* es de 128 bits de longitud, por lo tanto es mas grande comparada con una de *IPv4* que es de 32 bits.

- *IPv6* usa un nuevo formato para el encabezado en el que las opciones son separadas del encabezado base y son insertados solo cuando son necesarios.

- *IPv6* introduce nuevas opciones que permiten funciones adicionales.

- *IPv6* esta diseñado para permitir la extensión del protocolo si es requerido por nuevas tecnologías o aplicaciones.

- En *IPv6* el campo de tipo de servicio ha sido removido para dar paso al campo *"Flow Label"*, que nos permite asignar tratamientos especiales a paquetes.

- Las opciones de autenticación y encripción en *IPv6* proporcionan confidencialidad e integridad al paquete.

Como parte principal de la nueva versión del protocolo es importante conocer el formato del paquete del protocolo *IPv6.*

Formato del paquete y descripción de los campos

El formato del paquete *IPv6* ha cambiado considerablemente en comparación con el formato de la versión anterior. En la siguiente figura se introduce el formato del paquete *IPv6* y ahí se aprecian la nueva estructura del mismo, figura 2.30.

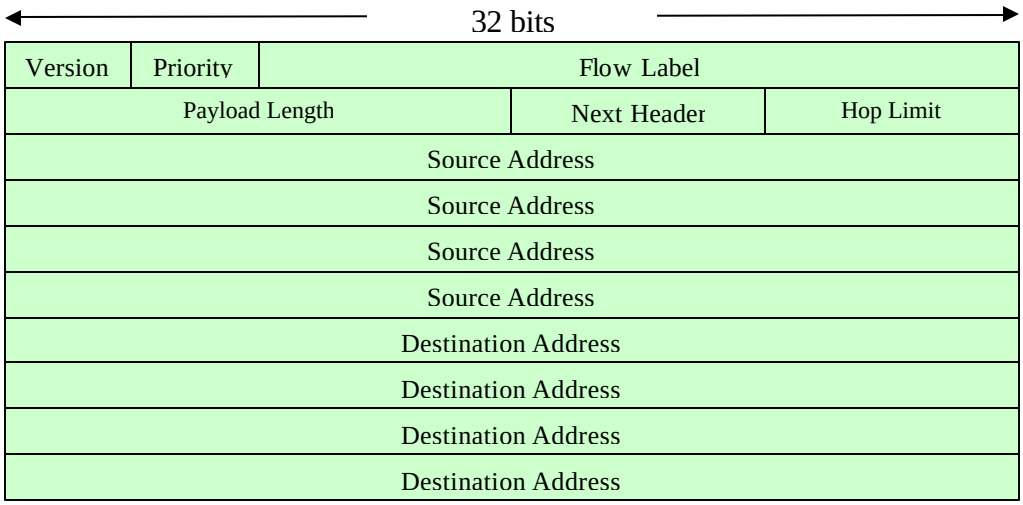


Figura 2.30

Version – Este campo de 4 bits define el número de versión del protocolo *IP*. Para *IPv6* el valor es de 6.

Priority – Este campo de 4 bits define la prioridad del paquete respecto a la congestión de tráfico de otros paquetes del mismo origen. *IPv6* divide el tráfico en dos categorías:

- *Congestión controlada*
- *Congestión sin control*

La congestión de tráfico controlado adapta el tráfico al tipo de congestión que exista. Para llevar a cabo una asignación de diferentes niveles de prioridad, el cual se muestra en la tabla 2.7.

Prioridad	Significado
0	No specific traffic
1	Background data
2	Unattended data traffic
3	Reserved
4	Atended bulk data traffic
5	Reserved
6	Interactive traffic
7	Control traffic

Tabla 2.7

No specific traffic. Este valor es asignado a un paquete cuando no se ha definido la prioridad.

Background data. Este valor es asignado a paquetes que ya han sido entregados con anterioridad.

Unattended data traffic. Este valor es asignado a paquetes que no son esperados por el usuario y que pueden ser muy grandes, un ejemplo de esto es el correo electrónico.

Attended bulk data traffic. Este valor se asigna a paquetes muy grandes, por ejemplo la transferencia de archivos mediante FTP.

Interactive traffic. Esta prioridad se asigna a protocolos que necesitan de la interacción con el usuario, un ejemplo de esto es: *TELNET*.

Control traffic. Este es el nivel mas alto de prioridad, y es asignado a aplicaciones de enrutamiento y de administración como: *RIP, OSPF, IS-IS; SNMP*, etc.

La congestión del tráfico sin control se refiere a un tipo de tráfico que no espera un mínimo retraso. Descartar estos paquetes no es deseable ya que la retransmisión en muchos casos es imposible.

La prioridad para estos paquetes va desde el 8 hasta el 15 y aunque aun no hay un estándar para dichos valores, la prioridad se asigna en base a la cantidad de datos pueden ser afectados.

Los datos que contienen menos redundancia como audio y video de baja fidelidad podrían tener una prioridad de 15, mientras que los datos con mas redundancia como el audio y el video de alta fidelidad deberían tener la prioridad mas baja. A continuación se muestra la distribución de prioridad en la congestión de tráfico no controlada en la tabla 2.8.

Prioridad	Significado
8	Datos con la mayor redundancia
.	.
.	.
.	.
15	Datos con la menor redundancia

Tabla 2.8

Flow Label – Esta etiqueta de flujo es un campo de 24 bits ó 3 bytes que es diseñado para proporcionar un manejo especial para un flujo particular de datos.

Payload Length. – Este campo define la longitud de los datos esta formado por 16 bits ó 2 bytes. Este campo define la longitud total del datagrama *IP*, excluyendo la base del encabezado.

Next Header – Este campo esta compuesto por 8 bits y define al encabezado que seguirá al encabezado base en el datagrama. Cada "*next header*" puede adquirir los siguientes valores, tabla 2.9.

Hop Limit – Este campo sirve para contar el límite de saltos máximo del paquete y esta compuesto por 8 bits. Tiene la misma función que el campo *TTL* en *IPv4.*

Code	Next Header
0	Hop by Hop option
2	ICMP
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
59	Null (no next header)
60	Destination option

Tabla 2.9

Source Address – Este campo esta compuesto por 128 bits ó 16 bytes e identifica la dirección origen del paquete.

Destination Address – Este campo esta compuesto por 128 bits ó 16 bytes e identifica la dirección destino del paquete.

Encabezado de extensión

A continuación serán descritos en forma general los encabezados de extensión, ya que su análisis se encuentra fuera del alcance de esta tesis, de acuerdo a la actividad que ellos desempeñan dentro de *IPv6*:

- *Hop by hop option*
- *Source routing*
- *Fragmentation*
- *Authentication*
- *Encrypted security payload*
- *Destination option*

Hop by hop option – Este encabezado define un conjunto arbitrario de opciones que son preparadas para ser examinadas por todos los equipos entre la trayectoria origen y el destino.

Source Routing – Este encabezado define el método para permitir al equipo origen especificar la ruta para un datagrama. En este campo se definen múltiples tipos de enrutamiento

Fragmentation – Este campo es incluido cuando se ha fragmentado el paquete y es requerida información referente a la fragmentación y al ensamble del paquete.

Authentication –Este encabezado lleva consigo información usada para autenticar los datos encriptados.

Encrypted security payload – Este encabezado lleva los datos encriptados para asegurar la comunicación.

Destination option – Este encabezado define un conjunto arbitrario de opciones que son previstas para ser examinadas por el destino.

Direccionamiento IPv6

El tamaño de una dirección IPv6 es de 128 bits divididos en dos bloques de 64 bits cada uno, el primer bloque identifica a la subred y el segundo identifica a la interfaz, figura 2.31.

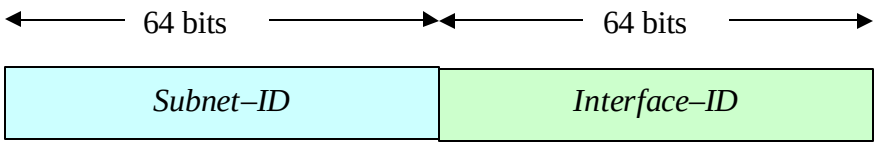


Figura 2.31

El bloque de *Sudnet–ID* se encarga de identificar a la red del enlace y es asignada por alguna organización internacional encargada de la asignación de identificadores de subred..

El bloque de *Interface–ID* se encarga de identificar a la interfaz dentro de la red o al usuario, mediante la asignación automática de una dirección *MAC* (48 bits) o de un identificador asignado manualmente por el administrador.

El bloque *Sudnet–ID* esta subdividido en 5 campos. Esos campos son mostrados en la figura 2.32.

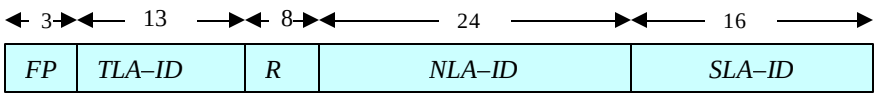


Figura 2.32

Format prefix (FP) – Este campo identifica el valor de una dirección única con el valor de 001 en binario.

Top–Level Aggregation Identifier (TLA–ID) – Este campo es fundamental en la comprensión del direccionamiento jerárquico de las direcciones *IPv6*. Las direcciones globales *IPv6* serán asignadas a *ISPs* u organizaciones encargadas de su distribución.

Reserved (R) – Este campo es reservado para uso futuro.

Next–Level Aggregation Identifier (NLA–ID) – Este campo identifica la siguiente asignación de direcciones globales *IPv6*. Este método de asignación promueve la reducción del tamaño de las tablas de enrutamiento.

Site–Level Aggregation Identifier (SLA–ID) – Este campo permite hacer la asignación de las subredes, este campo es delegado al administrador de la red.

La notación que utiliza *IPv6* para describir su dirección es mediante números hexadecimales (los números hexadecimales están compuestos por cuatro bits), que son agrupados en 8 bloques de 4 números hexadecimales y separados por dos puntos (**:**).

Ejemplo de una dirección *IPv6*:

2001:FDEC:BA98:0074:3210:000F:0000:FFFF

Aunque la direcciones *IPv6* son muy largas, muchos de los dígitos que son ceros pueden ser abreviados u omitidos, según sea el caso, siguiendo algunas reglas.

1) Se pueden omitir los ceros que están a la izquierda de algún número diferente de cero.

Por ejemplo:

2001:FDEC:BA98:0074:3210:000F:0000:FFFF

En esta dirección se puede abreviar *0074* por *74*, *000F* por *F* y *0000* por *0*, el numero 3210 no puede abreviarse ya que el cero se encuentra a la derecha de un número diferente de cero. De acuerdo a la primera regla, la dirección quedara de la siguiente manera:

2001:FDEC:BA98:74:3210:F:0:FFFF

2) Otra forma de llevar a cabo la omisión de ceros es cuando estos se encuentran de manera consecutiva, es decir podemos reemplazar los ceros consecutivos con solo poner dos veces los dos puntos.

Por ejemplo:

FDCE:0:0:0:0:BBFF:0452:FFFF

Al llevar a cabo la nueva abreviatura en base a la primera y segunda regla, tendremos:

FDCE::BBFF:452:FFFF

3) En una tercera regla que hace referencia a la omisión de ceros, nos encontramos con una restricción y es que no puede haber un doble existencia de dobles puntos.

Por ejemplo:

3FEE:0:0:0:FABB:0:0:1234

En este ejemplo se podrían omitir los 3 ceros que están consecutivos y después los otros dos ceros. Sin embargo, atendiendo la ultima regla mencionada, solo se puede llevar a cabo un omisión en los ceros consecutivos. Por lo tanto la nueva dirección podría quedar de alguna de las dos siguientes formas:

3FEE::FABB:0:0:1234 ó 3FEE:0:0:0:FABB::1234

Una forma de abreviar algún prefijo de dirección *IPv6* es empleando la diagonal (/) para señalar hasta que bit es la misma dirección. Es un modelo parecido a *CIDR.*

El siguiente ejemplo muestra este tipo de abreviación:

FDEC:0:0:0:0:BBFF:0:FFFF  ***FDEC::BBFF/96***

Tipos de dirección

El direccionamiento *IPv6* elimina las clases de direcciones, ahora *IPv6* solo define tres tipos de direcciones a diferencia de los cinco tipos de *IPv4*, estos son:

- *Direcciones Unicast*
- *Direcciones Anycast*
- *Direcciones Multicast*
- *Direcciones Reserved*

Las direcciones *Unicast* definen a usuario *(PC o host).* El paquete que es enviado a una dirección *unicast* debe ser entregado solamente a este usuario.

Las direcciones *Anycast* definen a un grupo de usuarios que tienen el mismo prefijo. Por ejemplo, todas los usuarios que están conectados a la misma red física o comparten el mismo prefijo. Por lo tanto un paquete que es enviado a una dirección *Anycast* debe ser entregado al usuario mas cercano del grupo.

Las direcciones *Multicast* definen a un grupo de usuarios que pueden o no compartir el mismo prefijo o la misma red física. Un paquete enviado a una dirección *multicast* debe ser entregado a cada usuario del grupo.

Las direcciones *reserved* o reservadas están destinadas para aplicaciones especificas, a continuación se listan los tipos de direcciones y las aplicaciones a las cuales se han destinado.

- Dirección de *Loopback*, esta dirección es utilizada para hacer pruebas hacia el mismo usuario, el formato que presenta esta dirección es el siguiente:

::1

Esto quiere decir que hay 31 ceros antes del uno.

- Direcciones de *Link–Local*, este tipo de direcciones son usadas si una *LAN* va a usar protocolos de Internet, pero no va a usar Internet por razones de seguridad, esto es algo parecido a las direcciones no homologadas en *IPv4*. este tipo de direcciones usan un prefijo inicial de 1111111010.

Las direcciones de enlace local presentan el siguiente esquema, figura 2.33.

10 bits	54 bits	64 bits
1111111010	Todos ceros	Interfaz–ID

Figura 2.33

- Direcciones de *Site–Local*, estas direcciones son utilizadas si un sitio tiene varias redes que usen protocolos de Internet pero no esta conectada a Internet. Este tipo de direcciones usan el prefijo inicial de 1111 1110 11.

El esquema de las direcciones de *Site–Local* es presentado en la figura 2.34.

10 bits	38 bits	16 bits	64 bits
1111111011	Todos ceros	Subset address	Interfaz–ID

Figura 2.34

- Direcciones *multicast* – Estas direcciones son usadas para definir un grupo de usuarios a los cuales se les desea enviar la misma información. Todas las direcciones *multicast* usan el prefijo 1111 1111 ó FF en el primer campo.

El formato de la dirección de *multicast* es presentado en la figura 2.35.

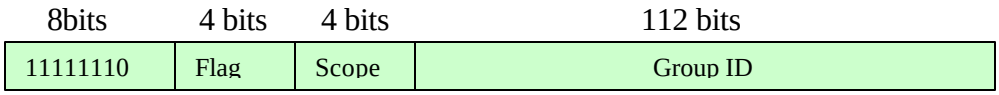


Figura 2.35

El segundo campo de esta dirección es un *"flag"* que define la dirección del grupo como permanente o transitorio, ver tabla 2.10. Una dirección permanente es definida por las autoridades de Internet y puede ser accedida en cualquier momento. Una dirección transitoria es usada solo temporalmente.

Código	Flag
0000	Permanente
0001	Transitorio

Tabla 2.10

El tercer campo define el alcance o *"scope"* de la dirección del grupo, ver tabla 2.11.

Codigo	Alcance
0000	Reservado
0001	Nodo local
0010	Enlace local
0101	Sitio local
1000	Organización local
1110	Global
1111	Reservado

Tabla 2.11

La tabla 2.12 muestra tipos de prefijos reservados y su aplicación para *IPv6.*

Prefijo	Tipo
0000 0000	Reservado
0000 0001	Sin asignar
0000 001	NSAP (Network Service Access Point)
0000 010	IPX (Novell)
0000 011	Sin asignar
0000 1	Sin asignar
0001	Sin asignar
001	Direcciones *Unicast* validas globalmente
010	Sin asignar
011	Sin asignar
100	Sin asignar
101	Sin asignar
110	Sin asignar
1110	Sin asignar
1111 0	Sin asignar
1111 10	Sin asignar
1111 110	Sin asignar
1111 1110 0	Sin asignar
1111 1110 10	Direcciones de enlace local
1111 1110 11	Direcciones de sitio local
1111 1111	Direcciones Multicast

Tabla 2.12

Direcciones compatibles

Para llevar a cabo la migración del protocolo *IPv4* al protocolo *IPv6* se requiere de una etapa de transición y para ello se han implementado formatos de direcciones compatibles entre ambos protocolos.

Existen dos esquemas que nos permites utilizar direcciones de *IPv4* embebidas en direcciones de *IPv6*, uno es conocido como direcciones compatibles y el otro como mapeo de direcciones.

- *Direcciones compatibles*

El esquema de direcciones compatibles permite establecer una dirección *IPv4* dentro de una dirección *IPv6*, pero dicha dirección tendrá un formato distinto al antes mencionado para *IPv6*. El prefijo de ocho ceros es reservado para esta tarea, los primeros 96 bits son puestos a cero y los siguientes 32 bits son referidos a la dirección *IPv4*, los ceros y la dirección *IPv4* son separados por dos puntos (:), esta técnica es usada cuando un usuario de *IPv4* quiere enviar paquetes a otro usuario que este usando *IPv6*.

Ejemplo:

Se tiene una dirección *132.248.120.211*, la dirección compatible seria:

132.248.120.211 → ***::132.248.120.211***

- *Mapeo de direcciones*

El mapeo de direcciones se auxilia del bloque reservado que tiene los primeros ocho bits puestos a cero y designa los siguientes 72 bits puestos a cero también, los siguientes 16 serán puestos a uno y por ultimo viene la dirección *IPv4* separada por dos puntos (:).

Ejemplo:

Se tiene una dirección 132.248.120.212 y al aplicar el método de mapeo de direcciones quedaría de la siguiente forma.

132.248.120.212 → ***::FFFF:132.248.120.212***

Capitulo 3

Enrutamiento

Introducción

Para llevar a cabo la comunicación comunicar a usuarios que pertenezcan a redes diferentes es necesario el movimiento de paquetes entre esas redes. Esa tarea es conocida como enrutamiento de paquetes o encaminamiento de paquetes.

El enrutamiento de paquetes es definido como el reenvió de paquetes de un origen a un destino, basándose en la dirección lógica destino, que es definida en la capa de red del modelo de referencia *OSI*.

Los equipos que permiten la conexión entre redes y que llevan a cabo la tarea de enrutamiento de paquetes son conocidos como *routers*, estos equipos trabajan con el direccionamiento lógico y se auxilian de protocolos de enrutamiento, que serán definidos mas adelante.

Sin el empleo de los *routers* y sin los protocolos de enrutamiento seria imposible poder comunicar a usuarios en diferentes redes, la comunicación estaría limitada solamente a usuarios en el mismo segmento, figura 3.1.

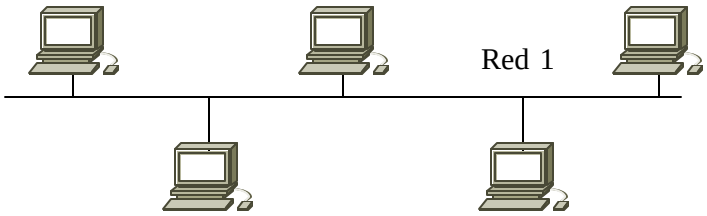


Figura 3.1

En la figura 3.2 muestra la comunicación de usuarios en 3 redes diferentes conectadas entre si mediante el empleo de *routers* y protocolos de enrutamiento.

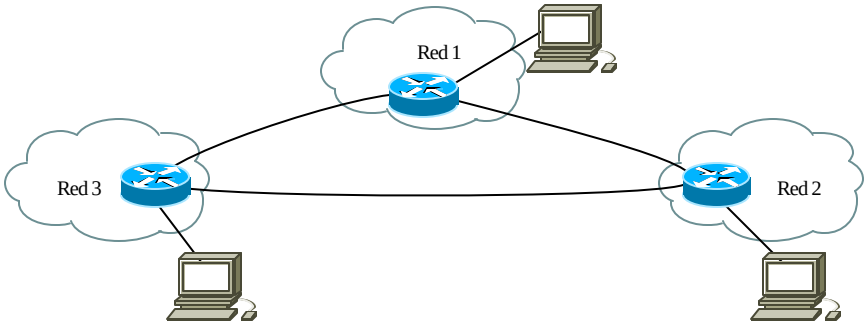


Figura 3. 2

El enrutamiento de paquetes es una tarea muy importante que es encomendada a los *routers, ¿Pero como se lleva a cabo el enrutamiento?* A continuación se describirá la tarea de enrutamiento.

Cuando un usuario1 intenta comunicarse con un usuario2 que esta ubicado en una red distinta a la del usuario1, es necesario enviar los paquetes a los *routers* para que estos se encarguen de enviarlos hacia otra red.

Los *routers* deben saber conocer la ubicación de las otras redes y conocer cual es la mejor ruta para que los paquetes puedan llegar a ellas.

Cuando un paquete es enviado hacia otra red, primero se envía hacia el *router* local para que este defina si la dirección destino se encuentra dentro de la misma red o es una red externa, si esta dentro de la misma red, el *router* local se encarga de encaminar el paquete hacia su destino final, si el destino final no es la misma red entonces el *router* local envía el paquete hacia el siguiente *router*.

El siguiente *router* determina si conoce la dirección destino y si esta está dentro de las conexiones del *router* entonces el paquete es enviado hacia el destino, pero si la dirección no se encuentra conectada a ese router, entonces el paquete es reenviado hacia otro *router*.

Al *router* al que son reenviados los paquetes que no han podido ser entregados es conocido como el *router* predeterminado o *"Default Gateway"*.

En la figura 3.3 podemos comprender el comportamiento de un paquete dentro de una red.

El usuario A desea enviar un paquete hacia el usuario B, el cual no se encuentra en la misma red, entonces el paquete es enviado al *router* W y como no encuentra la dirección destino dentro de sus interfaces, entonces envía el paquete hacia el siguiente *router*, que es el *router* X. El *router* X solo tiene conectada la red D y ahí no esta el usuario B, así que reenvía el paquete hacia el siguiente *router*, que es el *router* Y, y así sucesivamente hasta que el paquete llegue al *router* Z, el cual si encuentra dentro de su red conectada al usuario B y le envía el paquete del usuario A.

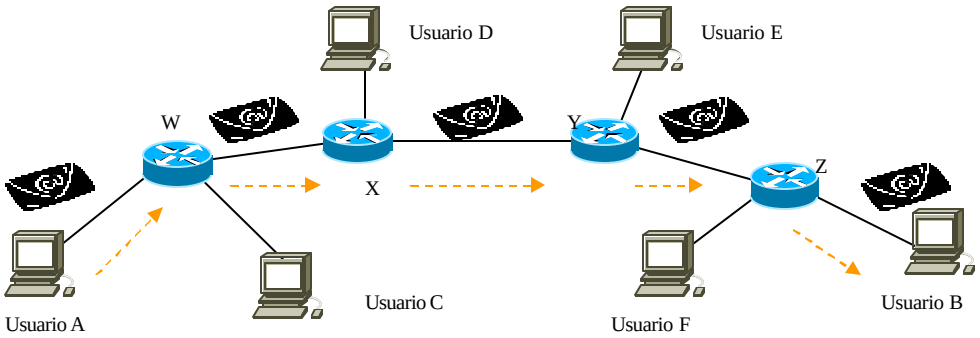


Figura 3.3

Como se puede apreciar, la tarea de los *routers* es muy importante en la comunicación de redes.

Los *routers* necesitan conocer las redes que tienen conectadas, para con ello poder crear una base de datos de las redes conectadas y de la ruta predeterminada en caso de no tener la red. La base de datos es conocida como tabla de enrutamiento.

Las tareas que debe realizar un *router* son:

- *Creación de la tabla de enrutamiento*
- *Reenvío de paquetes o datagramas.*

Tabla de enrutamiento

La tabla de enrutamiento contiene toda la información acerca de las interfaces que están conectadas al *router*, de las redes que están conectadas en cada interfaz y si esta conectada hacia otros *routers*.

La figura 3.4 es una muestra de una tabla de enrutamiento, en la cual se puede apreciar las redes que están conectadas en el router (127.0.0.1 y 191.72.1.0) y las que están más allá del *router (192.72.2.0 y 132.248.0.0).*

Destination	Gateway	Netmask	Flags	Metric	Ref	Use
127.0.0.1	*	255.255.255.255	UH	1	0	
191.72.1.0	*	255.255.255.0	U	1	0	
191.72.2.0	191.72.1.1	255.255.255.0	UGN	1	0	
132.248.0.0	191.72.1.1	255.255.0.0	UGN	1	0	

Figura 3.4

Loa *routers* utilizan diferentes mecanismos de enrutamiento para construir y mantener sus tablas de enrutamiento, también conocidas como tabla de reenvío, existen varios mecanismos de enrutamiento: dinámico, estático, directamente conectado y predeterminado, que serán vistos con mayor detalle mas adelante.

Esos mecanismos sirven como fuente de entrada de datos para las tablas, le proporcionan al *router* la información de las redes y subredes conectadas. Las tablas son utilizadas por los *routers* para determinar cual es la mejor ruta hacia entre el origen y el destino cuando se reenvían *datagramas.*

Estas tablas incluyen toda la información relacionada con las redes y subredes conocidas para un *router* y la dirección del siguiente *router*, también conocido como *"next hop"*, usada para alcanzar otras redes.

Cuando el *router* recibe un datagrama, la dirección destino es determinada y entonces comparada con cada ruta dentro de la tabla de enrutamiento hasta que una ruta exacta o la mejor ruta es encontrada. Si una ruta exacta coincide con el destino dentro de la tabla de enrutamiento, el *router* encamina el datagrama usando la dirección MAC pero si ninguna ruta es encontrada entonces el paquete es encaminado hacia la interfaz que lo conecta con el siguiente *router*

En una tabla de enrutamiento pueden ser incluidas múltiples rutas hacia un destino. Si este fuese el caso, una ruta debe ser seleccionada como la mejor ruta (*best path*) por un protocolo de enrutamiento y debe ser colocada en la tabla de enrutamiento como la primera ruta que se debe tomar. Esta ruta debería ser la principal ruta, pero algunos protoc olos de enrutamiento soportan balanceo de carga a través de múltiples rutas o caminos. Ambas rutas estarán disponibles para el destino y estarán en la tabla de enrutamiento. Ambas rutas pueden ser usadas alternativamente para el reenvío de datagramas por los *routers* y así balancear la carga a través de ambas rutas.

Una vez que el router ha construido su tabla de enrutamiento, este debe mantener actualizada toda la información contenida en ella.

La manutención de la tabla debe incluir las rutas configuradas manualmente y la información aprendida a través de protocolos de enrutamiento. La precisión que tenga el router para reenviar paquetes es la clave del éxito en el reenvío de tráfico.

El contenido de una tabla de enrutamiento es tan bueno como los datos que hayan entrado en ella. La comunicación exitosa entre sistemas remotos depende de la manutención de esta información. Mala información en la tabla de enrutamiento llevara al *router* a tomar malas decisiones en el reenvío de paquetes y buena información lo llevara a la buena selección de rutas.

Teniendo conocimiento sobre el proceso de enrutamiento de paquetes, la creación de la tabla de enrutamiento, manutención de la tabla y sabiendo como se lleva a cabo el reenvío de paquetes, el siguiente paso es comprender cuales son los mecanismos de enrutamiento que servirán como fuente de datos para la tabla de enrutamiento y como funcionan cada uno de ellos.

Mecanismos de enrutamiento

Existen varios mecanismos que ayudan al *router* a aprender las rutas hacia algún destino y así construir con esa información su tabla de enrutamiento, los mecanismos son los siguientes:

- *Directamente conectado*
- *Enrutamiento predeterminado*
- *Enrutamiento estático*
- *Enrutamiento dinámico*

Comúnmente los *routers* usan una combinación de estos mecanismos para construir la tabla de enrutamiento.

Directamente conectado

Este método de aprendizaje es muy sencillo, ya que las redes que están directamente conectadas al *router* son las primeras que aprende el *router* por están directamente conectadas.

La figura 3.5 es una muestra de la tabla de enrutamiento y presenta a las redes que están directamente conectadas al *router*.

```
Pruebas-I2>show ip route | include C
Codes: C - connected, S - static, R - RIP, M - mobile, B – BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2
C       137.164.27.88/31 is directly connected, GigabitEthernet0/1.2
C       200.0.204.132/30 is directly connected, GigabitEthernet0/1.1
C       200.23.60.240/30 is directly connected, ATM1/0.1
C       200.23.60.248/30 is directly connected, ATM1/0.4
C       200.23.60.104/30 is directly connected, Loopback0
C       200.23.60.108/30 is directly connected, ATM1/0.3
C       200.23.60.112/30 is directly connected, Tunnel0
C       200.23.60.118/32 is directly connected, Multilink1
C       200.23.60.116/30 is directly connected, Multilink1
C       200.23.60.1/32 is directly connected, Loopback1
```

Figura 3.5

Enrutamiento estático

El enrutamiento estático debe ser establecido manualmente en una tabla de enrutamiento estático. El enrutamiento estático o las rutas estáticas establecen cual debe ser el siguiente salto que los paquetes deben tomar de manera forzada.

Debido a que este tipo de enrutamiento es de naturaleza estático, no tiene la posibilidad de ajustarse a los cambios que ocurran en la red. Si una interfaz o el router fallan o tiene problemas de disponibilidad, la ruta hacia el destino también falla.

Este tipo de enrutamiento tiene la ventaja de eliminar todo el tráfico relacionado con las actualizaciones de enrutamiento. El enrutamiento estático tiende a ser ideal donde los enlaces son temporales o hay especificaciones de ancho de banda, y se requiere usar el método para redes dial-up o enlaces *WAN*.

Se puede implementar rutas estáticas en conjunción con otros métodos de enrutamiento para proporcionar rutas alternas o de respaldo hacia el destino, cuando los enlaces primarios vía protocolos de enrutamiento hayan fallado.

Si se deseara diseñar una red completa con solo este mecanismo de enrutamiento, se requeriría una establecer una ruta estática en cada Router para cada red que no estuviera directamente conectada, entonces este mecanismo no resulta práctico. Si un Router fallara o se agregara un Router, se tendría que reestablecer de nueva cuenta cada rutar, removiendo la ruta que fallo o agregar la nueva ruta. Mientras tanto los otros *routers* no pueden reenviar paquetes a esa ubicación pues la ruta es inválida hasta que se establezca la nueva.

Si se desea configurar una *LAN* estableciendo rutas estáticas, debe tomarse en cuenta que en el diseño de la red estas no deben pasar de 10 o 15 en total.

La figura 3.6 es una muestra de la base de datos de un router que presenta las rutas estáticas que están contenidas dentro de ella.

```
Pruebas-I2>show ip route | i S
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2

S       198.32.8.182 is directly connected, Tunnel0
S       132.248.4.0/24 [1/0] via 158.97.0.0
S       132.248.5.0/24 [1/0] via 158.97.0.0
S       132.248.3.0/24 [1/0] via 158.97.0.0
```

Figura 3.6

Enrutamiento predeterminado (default)

Este tipo de enrutamiento se establece en los *routers*, para que cualquier red que no se encuentren en su tabla de enrutamiento, la reenvíen hacia la ruta que se ha sido establecida.

La figura 3.7 es una muestra de la tabla de enrutamiento de un router que presenta las rutas predeterminadas que están dentro de ella.

```
Pruebas-I2>show ip route | include 0.0.0.0
Codes: C - connected, S - static, I - IGRP, R - RIP, M-mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS
level-2       ia - IS-IS inter area, * - candidate default

Gateway of last resort is 132.248.255.252 to network 0.0.0.0
S*   0.0.0.0/0 [1/0] via 132.248.255.252
```

Figura 3.7

La figura 3.8 muestra como es configurada una ruta predeterminada.

```
Pruebas-I2#show running-configuration | include ip route 0.0.0.0

ip route 0.0.0.0 0.0.0.0 132.248.255.252 name default->Pruebas
ip route 0.0.0.0 0.0.0.0 132.247.255.254 2 name default->backup
```

Figura 3.8

Enrutamiento dinámico

Este mecanismo de enrutamiento le permite al *router* conocer rutas hacia redes que estén conectadas mas allá de sus interfaces, es decir, una red puede ser conocida por un *router* aunque esa red no este directamente conectada al *router.*

La figura 3.9 nos presenta un escenario de donde se requeriría más que un enrutamiento estático y predeterminado a fin de poder comunicar a un usuario en la red A con un usuario en la red D

En la figura 3.9 existe un usuario en la red A que necesita enviar datos a un usuario en la red D, el *router* A es el que se encarga de reenviar paquetes hacia otras redes para la red A y este debe conocer el camino hacia la red D.

Para hacer el reenvío de paquetes hacia la red A resulta complejo realizarlo mediante el uso del enrutamiento estático, predeterminado y directamente conectado, así que este tipo de escenarios, en la actualidad muy comunes, originaron el desarrollo del enrutamiento dinámico, el cual mejoraria el enrutamiento de paquetes

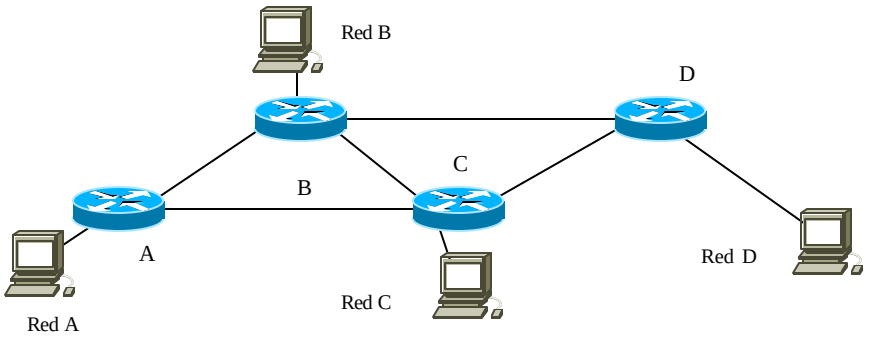


Figura 3.9

El enrutamiento dinámico se lleva a cabo mediante procesos de aprendizaje de redes. En cada uno de los *routers* se establece un proceso de descubrimiento de redes, así que cuando un *router* anuncia las redes que tiene conectadas a el, el *router* vecino es capaz de conocer mediante el proceso de aprendizaje las redes que están conectadas a el y entonces conocer el camino hacia ellas. Además de conocer la ruta hacia esas redes se realiza una selección de la mejor ruta en caso de que exista más de un camino.

Para llevar a cabo este tipo de enrutamiento de desarrollaron diferentes protocolos que involucran las técnicas fundamentales para llevar a cabo el aprendizaje de rutas, estos protocolos se clasificaron en base al mecanismo de aprendizaje y al algoritmo empleado para hacer la selección de rutas y entonces se tuvo la siguiente clasificación:

- *Protocolos Distance–Vector*
- *Protocolos Link-State*

Algunos ejemplos de protocolos para este tipo de enrutamiento son: *RIP, IS-IS, OSPF y BGP*, estos serán abarcados mas ampliamente en la siguiente sección.

Protocolos de enrutamiento

Como fue mencionado en la sección anterior, los protocolos de enrutamiento llevan a cabo el aprendizaje de nuevas rutas mediante la ejecución de algoritmos que permiten llevar a cabo el descubrimiento de redes, para después seleccionar la ruta adecuada y establecerla como la mejor ruta, mejor conocida como *"best–path",* .

Para poder diferenciar las rutas fue necesario establecer una medida en la que pudiese basarse el algoritmo para calcular y definir cual era la mejor ruta.

La unidad de medida se llamo *"métrica"* y es definida como el valor de costo usado para determinar la mejor ruta hacia un destino, es decir que el mejor valor de métrica implicaría el mejor camino hacia un destino.

Para algunos protocolos de enrutamiento, esas métricas pueden ser estáticas y no pueden ser cambiadas por el usuario y para otros protocolos esos valores de costo pueden ser asignados dinámicamente y asignados por el administrador de la red.

El establecimiento del valor de *métrica* se realiza en base a los factores implicados en las conexiones físicas, los factores implicados son:

- *Saltos*
- *Ancho de banda*
- *Retardo*
- *Confiabilidad*
- *Carga*

Saltos

Los *Saltos o Hops,* es un valor de métrica usado para medir la distancia que un paquete debe cruzar para llegar a su destino, es decir cada vez que un paquete es reenviado por un router es considerado como un salto.

Ancho de banda

El *Ancho de banda o Bandwidth*, es la medida que se emplea para determinar la capacidad de un enlace. La unidad de medida de esta métrica es en bits por segundo. Los enlaces que soportan la transmisión en magnitudes de gigabit *(1000 Mbps)* son preferidas por encima de enlaces dedicados *(56 kbps).*

Retardo

El *Retardo* o *Delay* es medido en centésimas de microsegundos *(µs)* y representa la cantidad de tiempo en que un router proceso y envía un paquete. Los protocolos que emplean esta métrica deben determinar los valores de retardo a lo largo de toda la ruta para todos los enlaces. Después de hecha la determinación de retardo elige la de menor retardo acumulado y es seleccionada como la mejor ruta.

Confiabilidad

La *Confiabilidad o Reliability,* esta métrica mide la confiabilidad de los enlaces por los cuales debe pasar el paquete y esta relacionado con el desempeño de los enlaces, es decir, si el enlace tiene fallas, errores de interfaz o perdidas de paquetes. Los enlaces que presenten mayores problemas serán considerados menos confiables y por lo tanto se convierten en rutas no deseables.

Carga

La *Carga o Load,* esta métrica se encarga de medir la cantidad de tráfico que esta pasando por el enlace en forma de porcentaje. Siendo el valor de 255 equivalente al 100%. Debido a la peculiaridad de esta métrica, la carga puede ser variable.

Después de conocer los factores que están implicados en la determinación de la métrica para los enlaces, es momento de comenzar a describir el funcionamiento de los mecanismos de aprendizaje de rutas o protocolos de enrutamiento.

Protocolos Vector–Distancia

Este protocolo es mejor conocido como *"Distance-Vector Protocol"* fue el primero de los protocolos de enrutamiento dinámico y emplea un algoritmo conocido como *"Algoritmo Bellman–Ford "*

Este algoritmo realiza la selección de la mejor ruta en base a la distancia que se tenga hacia alguna red La métrica que emplea este tipo de protocolos es en base a la variable denominada como *"Saltos"*.

Algunos de los protocolos *Distance–Vector* son *RIP* e *IGRP*

A continuación se presentan los conceptos fundamentales para conocer el funcionamiento de los protocolos de enrutamiento que emplean el algoritmo *Distance-Vector*:

- *Actualización de rutas*
- *Métricas*
- *VLSM*
- *Balanceo del tráfico*
- *Alcance*
- *Convergencia.*

Actualización de rutas

La actualización de la tabla de enrutamiento bajo este algoritmo se lleva a cabo mediante el envió de mensajes broadcast hacia todos los nodos dentro de la red. Esto provoca consumo de procesamiento innecesario en nodos que no requieren de esa información.

Cuando se inicia un proceso dinámico, todos los *routers* envían la tabla de enrutamiento completa hacia todos sus vecinos, por lo tanto estos protocolos envían actualizaciones de forma periódica o cuando un evento ocurre, sea una falla en algún enlace o se agrega uno nuevo. Al enviar estas actualizaciones completas sin que un cambio haya sido efectuado en la red, provoca un desperdicio de ancho de banda.

El tiempo establecido para el reenvió de las actualizaciones en *RIP* es de 30 segundos y en *IGRP* es de 90 segundos

Métricas

La métrica que emplea el protocolo *RIP* es la de salto (*hop*) y la métrica empleada por IGRP es una combinación de ancho de banda y retardo (*bandwidth y delay*).

VLSM

El término de *VLSM* se refiere al subneteo o subdivisión de las redes, para lo cual estos protocolos no han sido facultados para emplear las condiciones del *subnetting*. Estos protocolos son considerados con clase o *classful* y no permiten enrutar redes más pequeñas que las ya establecidas *(clase A, B y C).*

Balanceo del tráfico

El balanceo del tráfico se refiere a la distribución de los paquetes, es decir si se tiene dos enlaces, se puede repartir los paquetes pos las dos rutas disponibles. Algunos protocolos soportan hasta para seis rutas disponibles.

Alcance

Este alcance se refiere a la máxima distancio o numero máximo de saltos que un paquete de este protocolo puede alcanzar. Para este tipo de protocolos, el alcance máximo es de 15 saltos. Debido a esta característica, estos protocolos no son recomendables para redes grandes

Convergencia

La convergencia se refiere al tiempo en que todos los *routers* dentro de un dominio han llegado a un estado en que todos tienen las mismas tablas de enrutamiento.

Después de conocer el funcionamiento, las características y las limitaciones del protocolo *Distance –Vector,* es momento de pasar a la descripción del protocolo que resolvería dichas limitaciones, el protocolo *Link –State*.

Protocolos Estado–Enlace

Este protocolo es mejor conocido como *"Link–State Protocol"* y fue desarrollado específicamente para corregir las limitaciones que presentaban los protocolos *Distance–Vector*. Este protocolo emplea el algoritmo *Dijkstra* o *Short Path First* para determinar la ruta mas corta hacia el destino.

Este tipo de protocolos emplean una métrica que involucra a todas la variables posibles, a diferencia de los protocolos *Distance –Vector* que solo emplean los *"Saltos"*.

Los protocolos *estado-enlace* proporcionan una mayor flexibilidad y son más sofisticados que los *distance–vector*. Algunos de los protocolos que mas se han desarrollado son: *IS–IS* y *OSPF*, el primer protocolo será explicado mas ampliamente en el siguiente capitulo ya que es la fuente principal de esta tesis.

Estos protocolos también llevan a cabo de forma dinámica y automática el descubrimiento de vecinos (*routers*) para establecer las nuevas redes en la tabla de enrutamiento y determinan la mejor ruta *"Best Path"* hacia el destino mediante la ejecución del algoritmo *Dijkstra*.

La selección de la mejor ruta para estos protocolos depende de los costos que sean asignados a los enlaces por los protocolos. La asignación de costos se realiza en base a parámetros de las conexiones físicas que pueden ser: ancho de banda, retardo, confiabilidad y carga.

Algunas de las diferencias mas notables entre los protocolos *estado-enlace* y los protocolos *vector–distancia* es que los primeros reducen el tráfico *broadcast*, debido a que no envían mensajes *broadcast* periódicamente y tampoco realizan el envían de la tabla de enrutamiento completa. Los protocolos *estado-enlace* realizan un intercambio completo de la tabla de enrutamiento solamente cuando se esta inicializando el proceso, después las actualizaciones de la tabla solo se dan mediante mensajes multicast, es decir, solo son enviados a los *routers* en la red y no a los host.

Estos protocolos solo envían el cambio que se ha efectuado para que sea sustituido en la tabla de enrutamiento ya establecida y no se envía toda la tabla como lo hace la contraparte.

Los cambios que han sido efectuados comienzan a inundar toda la red rápidamente y si ningún cambio ha ocurrido, no generan ninguna actualización. Esta rápida inundación permite tener una rápida convergencia entre los *routers.* Aunque a veces el inundado de actualizaciones puede provocar exceso de tráfico.

Estos protocolos soportan el *subnetting*, es decir pueden reenviar información de subredes y basan su funcionamiento en la construcción y manutención de tres bases de datos, que son:

- *Tabla de vecinos (base de datos de adyacencias)*
- *Mapa topológico (base de datos del estado de los enlaces)*
- *Tabla de enrutamiento (base de datos de reenvió)*

Para establecer la tabla de vecinos, todos los *routers* deben identificar a sus vecinos, enviando mensajes *hello*, es decir a los *routers* que estén directamente conectados a ellos, estableciendo una relación con el otro, conocida como adyacencia.

Una vez que la adyacencia ha sido establecida, la información de enrutamiento puede ser intercambiada mediante el inundado de actualizaciones en todo el dominio de enrutamiento. Cualquier cambio que ocurra será inmediatamente anunciado a todas las rutas excepto a la que origino el mensaje. Todos los *routers* pueden procesar toda la información en forma paralela y así llevar a cabo una convergencia más rápida.

Al estar recibiendo todas las actualizaciones, los *routers* construyen y mantienen el mapa topológico del dominio.

Cada *router* construye un árbol lógico, ubicándose en la posición de raíz, con todas las redes y subredes emanando de este.

Todos los protocolos *Link–State* tienen como origen de la tabla de enrutamiento la ejecución del algoritmo *Dijkstra* contra el mapa topológico, ubicando las rutas de más bajo costo en la tabla de enrutamiento.

Los principales inconvenientes de estos protocolos son la cantidad de *CPU* que involucran en la calculación de cambio de ruta y los recursos de memoria que se requieren para almacenar la tabla de vecinos, tabla de enrutamiento y el mapa topológico completo.

Con el paso del tiempo los protocolos han ido sufriendo cambios y la tendencia se inclino básicamente hacia los protocolos basados en los algoritmos *Dijkstra* y en una mezcla de *Dijkstra* y *Bellman-ford*.

Dominios de enrutamiento

El termino *"Dominio de enrutamiento"* es empleado para referirse a *routers*, redes y subredes que emplean el mismo protocolo de enrutamiento para comunicarse entre ellos y que también comparten información dentro de la misma área común, este dominio de enrutamiento debe ser controlada por una entidad administrativa única.

Los dominios de enrutamiento pueden estar conectados por protocolos de enrutamiento *Distance–vector o Link–state*, por enrutamiento estático o solamente por redes directamente conectadas, dependiendo de las necesidades requeridas.

La figura 3.10 muestra un dominio de enrutamiento en el que los *routers* se conectan mediante diferentes tecnologías como: *FDDI, Ethernet* y enlaces seriales, pero el protocolo de enrutamiento es el mismo.

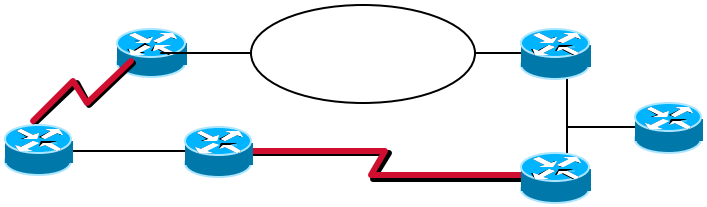


Figura 3.10

Puede existir la necesidad de unir a dos o mas dominios de enrutamiento que ocupen diferente protocolo de enrutamiento, y a la unión de estos dominios se les llama *"Sistemas autónomos"*, denominado como *"Autonomous System"* (AS).

La identificación de los sistemas autónomos es mediante números decimales y su asignación realizada por organismos internacionales ubicados geográficamente en el mundo, estos organismos son: *AFRINIC*, *APNIC*, *ARIN*, *LACNIC* y *RIPE.*

Un sitio para consultar acerca de la distribución de sistemas autónomos es *http://www.iana.org/assignments/as-numbers*.

En la figura 3.11 se muestra un sistema autónomo 100 que permite la unión del dominio de enrutamiento A, B y C y que se basan en el protocolo de enrutamiento *RIP*, *OSPF* y *IS-IS* respectivamente.

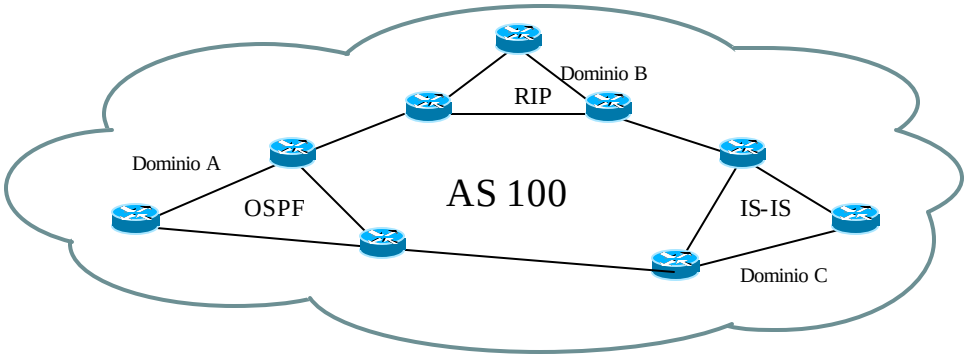


Figura 3.11

Debido a la necesidad de comunicación entre usuarios de redes en diferentes lugares, fue necesario desarrollar la comunicación entre sistemas autónomos.

La comunicación entre sistemas autónomos se realiza mediante un protocolo llamado *Border Gateway Protocol (BGP),* el cual será descrito brevemente mas adelante.

Dadas las funciones realizadas por un protocolo de enrutamiento dentro de los sistemas autónomos y los dominios de enrutamiento, se puede realizar la siguiente clasificación:

- *Interior Gateway Protocols (IGP)*
- *Exterior Gateway Protocols (EGP)*

Los *"Interior Gatreway Protocols"* o *"Protocolos de enrutamiento interior"* son los que realizan funciones de enrutamiento exclusivamente dentro de una red local o sistema autónomo, estas funciones permiten conectar varios dominios de enrutamiento, ejemplos de estos protocolos son: *IS–IS, OSPF y RIP.*

Los *"Exterior Gateway Protocols* o *"Protocolos de enrutamiento externo"* son los que realizan funciones de enrutamiento solamente para interconectar sistemas autónomos, en este campo no ha habido el desarrollo en comparación con los *IGPs* y el único protocolo que es implementado para llevar a cabo estas tareas es: *BGP*

En la figura 3.3, se muestra la ubicación de los protocolos de enrutamiento de acuerdo a las funciones que realice dentro de un sistema autónomo. El sistema autónomo 100 tiene dos dominios de enrutamiento, el primero trabaja con el protocolo *IS–IS* y el segundo con el protocolo *OSPF*, para el sistema autónomo 200 se tiene dos dominios de enrutamiento, el primero trabaja con *RIP* y el segundo con *OSPF*, estos a su vez son unidos mediante el protocolo *BGP*.

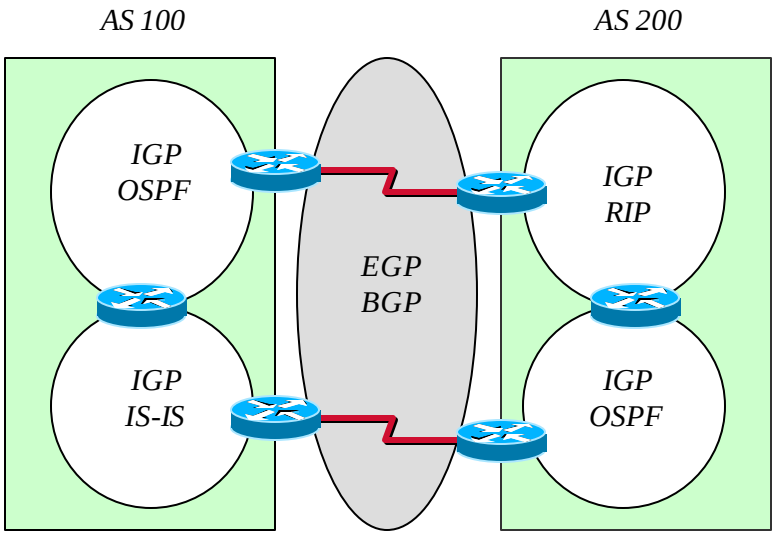


Figura 3.3

Protocolos de enrutamiento interior (IGP)

A continuación se describirá brevemente el funcionamiento, las características y la operación de los protocolos de enrutamiento interior para comprender el porque de la selección de *IS-IS* como protocolo propuesto.

Los protocolos descritos serán:

- *RIP*
- *OSPF*
- *IS–IS*

RIP

El protocolo *RIP* (*Routing Information Protocol*) puede correr en hosts o en *routers*, utiliza la comunicación *UDP*, puerto 520 y esta clasificado como un protocolo de red debido a que su función y operación están basadas en el reenvió de paquetes de capa de red.

RIP proporciona determinación de ruta dentro de un sistema autónomo y realiza un mejor desempeño en redes pequeñas ya que en redes mas grandes esta limitado por características de funcionamiento.

RIP presenta las siguientes características en su funcionamiento:

- Es un protocolo basado en *broadcast*, es decir que los *routers* que se encuentran dentro del mismo segmento llevan a cabo sus actualizaciones mediante mensajes *broadcast*.

- Funciona como *IGP* y conserva el contacto de las rutas internas a alguna organización de Internet.

- Trabaja mejor en redes pequeñas debido a que esta basado en *broadcast* y a que su diámetro es limitado y no es escalable hacia redes medianas o grandes.

- Es un protocolo *distance–vector* y usa los saltos como métrica de distancia para hacer la mejor selección de ruta a el destino.

- Sus actualizaciones son llevadas a cabo en forma periódica, la cual se realiza cada 30 segundos mediante el envió de *broadcast*.

- Cuando es necesaria la actualización de la tabla de enrutamiento, envía toda la tabla hacia todos los *routers* a pesar de que a ellos no les afecte el cambio.

- Este protocolo esta limitado a un máximo diámetro de red de 15 saltos y cualquier valor arriba de este valor es considerado como inalcanzable.

- No soporta *VLSM*

El protocolo *RIP* basa su selección de ruta al destino mediante la ruta mas corta, medida en saltos y con un valor máximo de 15. Las actualizaciones son enviadas utilizando mensajes periódicos incluyendo toda la información contenida en la tabla de enrutamiento a pesar de que ningún cambio haya sido efectuado.

Aunque existen otros protocolos actualmente, *RIP* se mantiene como uno de los más populares debido a la menor complejidad en su configuración.

Este protocolo esta descrito en los *RFCs 1058* y *2453*.

OSPF

El protocolo *OSPF* emplea el algoritmo de enrutamiento *Link–state* y hace más inteligente la determinación de la mejor ruta que los protocolos *distance–vector*. Cuado determina la mejor ruta hacia el destino, como todo protocolo *link–state*, puede considerar cualquiera de las siguientes métricas: ancho de banda, retardo, carga y confiabilidad.

Todos los *routers OSPF*, redes y subredes están lógicamente agrupadas dentro de áreas. Las redes *OSPF* pueden consistir de una sola área o múltiples áreas ordenadas jerárquicamente. Si un área única o múltiples áreas son contenidas dentro de una red completa de *OSPF* es denominada dominio de enrutamiento (*routing domain* o sistema autónomo)

Esta división jerárquica aísla los cambios y el trafico de actualización de ruta hacia las áreas, además de reducir el sobre encabezado (*overhead*) involucrado en el mantenimiento de grandes tablas de enrutamiento y en la recalculaciòn de ruta cuando algún cambio ha ocurrido.

Este protocolo tiene ventajas sobre los protocolos *distance–vector*, como son:

- *OSPF* usa mensajes *multicast* para distribuir la información de ruta en lugar de mensajes *broadcast*, esto quiere decir que solo envía mensajes de actualización a los *routers* vecinos.

- Las actualizaciones de tablas de enrutamiento solo se lleva a cabo al menos que un cambio haya ocurrido en la red y no periódicamente como lo hacen los protocolos *distance – vector*.

- Las actualizaciones inundan todos los *routers OSPF*, permitiendo una actualización paralela e incrementando el tiempo de convergencia.

- Cada cambio solo efectúa la actualización del cambio en la tabla y no de la tabla completa.

- Las actualizaciones son aisladas a los *routers* que se encuentran en el área donde se ha llevado a cabo el cambio.

- *OSPF* es capaz de soportar redes de mediano y mayor tamaño debido a que no tienen un máximo conteo de saltos limitando su diámetro

- *OSPF* soporta *VLSM*, es decir puede reenviar redes y subredes, esto también es conoc ido como sin clase o *classless*.

- Para la calculación de costo emplea variables como: ancho de banda, retardo, confiabilidad y carga, dentro de estas variables no se encuentran los saltos.

El protocolo *OSPF* esta descrito en el *RFC 1583*.

IS–IS

El protocolo *IS-IS* (*Intermediate System to Intermediate System*) fue diseñado para el modelo *OSI* originalmente, pero después al ver los beneficios de este y que podía modificarse para trabajar con redes *OSI* y redes *IP* se creo una nueva versión llamada *Integrated IS–IS* o *Dual IS–IS*.

Este protocolo esta basado en el algoritmo *Dijkstra* y es muy parecido en su comportamiento a *OSPF*, sin embargo presenta alguna diferencias que permiten establecer cual de los dos proporcionan un mejor funcionamiento.

A continuación se mencionan las principales características de *IS-IS*, ya que en el siguiente capitulo esta enfocado a este protocolo exclusivamente:

- *IS-IS* usa mensajes *multicast* para distribuir la información de ruta, esto quiere decir que solo envía mensajes de actualización a los *routers* vecinos

- Las actualizaciones de las tablas solo se envían cuando un cambio ha ocurrido.

- Las actualizaciones comienzan a fluir de forma paralela, provocando con esta una inundación en todos los *routers* con IS-IS más rápida y con ello menor tiempo de convergencia.

- Cuando algún cambio ha ocurrido solo se envía la actualización de la tabla de enrutamiento y no la tabla completa.

- Las actualizaciones son exclusivas para las áreas de enrutamiento.

- La implementación de *IS-IS* no esta limitada por el tamaño de la red, es un protocolo diseñado para operar en redes de gran y menor tamaño.

- *IS–IS* soporta el reenvió de redes *IP* y de redes *OSI*, en las redes *IP* se incluye también a las subredes.

- *IS–IS* también establece áreas, es decir mantiene una división jerárquica para distribuir el tráfico de las actualizaciones.

- Para la calculación de costo, actualmente este es definido por el administrador de la red, es decir estable el valor de la métrica. En versiones posteriores se pretende involucrar como: retardo, costo, error y el valor predeterminado.

- En la nueva versión de este protocolo ya se puede soportar en enrutamiento de redes *IPv6*.

Este protocolo es descrito en el *RFC 1195* para el soporte de redes *IP* y en el *draft "is-is-ipv6-05"* se establece lo referente al soporte de *IPv6*.

Protocolos de enrutamiento externo (EGP)

El protocolo desarrollado para realizar enrutamiento externo es *BGP* y es el único protocolo desarrollado levar a cabo dicha tarea.

La descripción de este protocolo será demasiado breve, pues su análisis es muy amplio y queda fuera del alcance de esta tesis.

BGP

Este protocolo esta diseñado para permitir la comunicación entre sistemas autónomos, sin importar la estructura que tenga cada sistema autónomo en particular.

BGP es un protocolo basado en el algoritmo *Bellman–ford* y también en el algoritmo *Dijkstra*, es decir es un protocolo hibrido que conjuga algunas características de los protocolos *distance–vector* y los protocolos *link–state*.

Se basa principalmente en el conteo de saltos, pero la métrica es definida por varios parámetros que son establecidos por el protocolo y pueden ser modificados por el administrador.

BGP es un protocolo muy robusto que ha sido la columna vertebral de Internet. El propósito principal es anunciar la presencia de redes y estructuras a otros *routers* que también hablen *BGP* en Internet (*routers* de los *ISP*s). *BGP* provee una estructura de enrutamiento jerárquico para enlazar lugares distantes en todo el Internet.

Aunque *BGP* es conocido por su capacidad de interconectar sistemas autónomos, también puede establecer conexiones internas con los *routers* que estén ubicados dentro de su sistema autónomo, debido a la característica arriba mencionada, existen dos estados de funcionamiento para este protocolo y son:

- *External Border Gateway Protocol (EBGP)*
- *Internal Border Gateway Protocol (IBGP)*

IBGP es empleado para realizar las conexiones de *BGP* internas y *EBGP* es usado para conectar sistemas autónomos diferentes.

A continuación se mencionan las características principales de este protocolo:

- *BGP* proporciona selección inteligente de ruta basándose en prefijos específicos y en la ruta mas corta hacia los sistemas autónomos (*AS-Path*).

- Soporta *CIDR* (*Classless Inter Domain Routing*).

- Es un protocolo orientado a conexión, es decir establece primeramente una sesión de *TCP* (puerto 179) con los *routers* vecinos antes de llevar a cabo el intercambio de mensajes.

- Aprende rutas externas e internas de los *routers* que hablen *BGP*.

- La selección de rutas involucra atributos que los *routers* establecen para las redes, los atributos puedes ser modificados por el administrador y pueden ser los siguientes:

    - *Peso (Weight)*
    - *Preferencia local (Local preferente)*
    - *Métrica (MED)*
    - *Ruta hacia el sistema autónomo (AS–Path)*
    - *Siguiente salto (Next–Hop)*
    - *Comunidad (Community)*

- Cuando existen dos o más caminos por los que un paquete puede llegar a su destino, el *router* ejecuta un mecanismo para seleccionar la mejor ruta y establecerla en la tabla de enrutamiento como la mejor , *"best"*.

- Los criterios de desempate para dos caminos hacia un destino son:

    - Si en la ruta se encuentra un siguiente salto inaccesible, entonces esa ruta es descartada.
    - Prefiere la ruta que tenga un mayor peso definido.
    - Si el peso es el mismo, entonces escoge la que tenga una mayor preferencia local.
    - Si la preferencia local es la misma, entonces elije la ruta que haya sido originada por el *router* local.
    - Si la ruta no fue originada por el *router* local, entonces elije la ruta mas corta hacia el sistema autónomo (*AS–Path*).
    - Si los *AS–Path* tienen la misma longitud, entonces elige el que tenga el *AS–Path* origen menor.
    - Si los códigos de origen son iguales, entonces elije el que tenga la métrica menor.
    - Si aun las rutas aun están empatadas, entonces elije la que venga del vecino más cercano.
    - Y si después de todo siguen igual, entonces elije la que tenga el identificador del *router* con menos valor (*Router–ID*).

Como puede apreciarse en la descripción del funcionamiento de este protocolo, se entiende que es mucho más complejo que los que se mencionaros anteriormente y es por ello que para su total comprensión se necesitara de lectura adicional.

Este protocolo es descrito en el *RFC 1771* y se pueden encontrar tutórales en la red como los de las siguientes ligas: *http://www.academ.com/nanog/feb1997/BGPTutorial/* o *http://www.ittc.ku.edu/EECS/EECS_800.ira/bgp_tutorial/* .

Capitulo 4

IS-IS

Historia de IS–IS

Este protocolo fue desarrollado por la organización internacional de estándares (*International Standards Organization, ISO*) basado en un protocolo conocido como *DECnet fase V. ISIS* se desarrollo para trabajar con un protocolo de red no orientado a conexión (*Conectionless Network Protocol, CLNP*) del modelo *OSI.*

La empresa *Digital Equipment Corporation (DEC)* desarrollo su propio conjunto de protocolos en 1975 llamado *DNA (Digital Network Architecture)*. El protocolo *DECnet* es su principal protocolo para la comunicación de computadoras. La clasificación y crecimiento del protocolo se distingue por fases.

La arquitectura *DNA* esta compuesta de 8 capas que son similares a las capas del modelo *OSI*, figura 4.1.

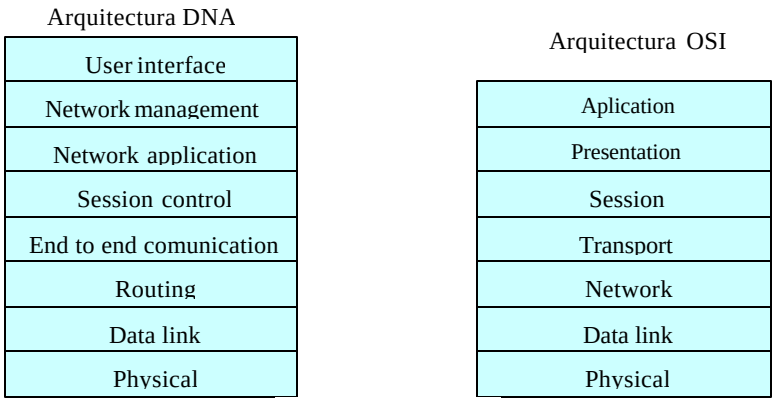


Figura 4.1

Las primeras cuatro fases del protocolo *DECnet* no agregaron el soporte para la arquitectura *OSI*, ya que solo estaban basadas en la arquitectura *DNA*.

Cuando se comenzó a desarrollar la *fase V* de *DECnet*, la empresa decidió adoptar el modelo *OSI* para ser agregado en el soporte de *DECnet*, así *DECnet* soportaría el modelo *OSI* y *DNA*. Este protocolo fue conocido en el medio como *DECnet fase V* o *DECnet OSI/DNA.*

Debido a la similitud entre el modelo *OSI* y la arquitectura *DNA*, la compatibilidad de ambos sistemas fue exitosa. Sin embargo, ahora se necesitaba un protocolo de enrutamiento que pudiera enrutar arquitecturas *OSI* o *DNA* y también *IP*. La solución a este problema fue el protocolo de enrutamiento *Intermediate System to Intermediate System (IS-IS)*.

Cuando *DECnet fase V* estaba siendo desarrollado, *ISO* también desarrollaba su servicio de red no orientado a conexión (*Conectionless Network Service, CLNS*) para trabajar con le protocolo *CLNP* (*Conectionless Network Protocol*). *CLNP* es un protocolo puramente *OSI* que maneja direccionamiento en la capa de red y permite la transferencia de datos no orientada a conexión entre dos sistemas. *CLNP* es rápido, relativamente pequeño en tamaño y es el equivalente de *OSI* para el protocolo *IP*, *CLNP* fue introducido en el capitulo 2.

Cuando *DECnet* comenzó a desarrollar la *fase V OSI/DNA*, requería de un protocolo de enrutamiento *OSI*. El protocolo encontrado fue *CLNP*, sin embargo *CLNP* no permitía la compatibilidad con las fases anteriores de *DECnet* y en ese momento no había un protocolo de enrutamiento suficientemente flexible para manejar las arquitecturas mezcladas de *OSI* y *DNA*, y el que se tenia era muy robusto para trabajar en grandes ambiente son orientados a conexión.

Como *DECnet* necesitaba un protocolo flexible y *OSI* ya había desarrollado uno para ambientes no orientados a conexión. *DEC* decidió ayudar al desarrollo de *IS-IS* para ambientes *OSI/DNA* con *CLNP*.

ISO desarrollo el protocolo *IS-IS* como un rápido y escalable protocolo de enrutamiento basado en *link–state* para *CLNP*. *IS-IS* cubrió las limitaciones del protocolo *DECnet fase V* y debido a que estaba basado en servicios no orientados a conexión solo era necesario hacer pequeñas modificaciones para que soportara *IP* además dar soporte a *CLNP*.

En algún tiempo se pensó en establecer *CLNP* como el protocolo de Internet, pero fue clara la superioridad de *IP* en el Internet. Por lo tanto el soporte de *IP* ha sido agregado a *IS-IS*.

Cuando se implemento el soporte de *IP* a *IS-IS* aun mucha gente usaba ambientes puramente *OSI* y se decidió nombrar de forma distinta a la nueva versión de IS-IS y esta se llamo *Integrated IS-IS*, la cual ofrecía soporte para múltiples protocolos de red simultáneos, como *CLNP* e *IP*.

Como ya se había mencionado anteriormente, *IS-IS* es un protocolo basado en link–state (algoritmo shortest path. first, también conocido como dijkstra) para el establecimiento de las mejores rutas hacia el destino.

Conceptos básicos de IS–IS

Es importante estar familiarizado con los conceptos que maneja *IS-IS*, ya que existen nuevos conceptos que son introducidos.

Conceptos fundamentales en el lenguaje del protocolo *IS-IS* son:

- *Dominios de enrutamiento*
- *Áreas y jerarquías de enrutamiento*
- *Intermediate System y End system*

Dominio de enrutamiento

Un dominio de enrutamiento es una red en la que todos los routers corren el mismo protocolo de enrutamiento *(IS-IS)* para soportar el intercambio de información de enrutamiento el protocolo *IS-IS* fue inicialmente diseñado para soportar dominios solo *CLNP*, y después en el *RFC 1195* se adaptaron las especificaciones para que *IS-IS* soportara *IP*.

Los siguientes requerimientos s de implementación fueron especificaciones en el *RFC 1195* para los dominios de *IS-IS*:

- Dominios solo de *IP*, estos dominios solo enlutan trafico *IP* pero soportan reenvió y procesamiento de paquetes OSI, necesarios para la operación de *IS-IS*.
- Dominios solo de *OSI-CLNP*, estos dominios llevan trafico *CLNP* incluyendo los requerido para la operación de *IS-IS*.
- Dominios duales, estos dominios enlutan tanto trafico *IP* como *CLNP* simultáneamente.

Es posible diseñar dominios duales, donde algunas áreas trabajen con tráfico *IP* y otras con tráfico *CLNP*, en la figura 4.2 se muestra un dominio de enrutamiento mixto.

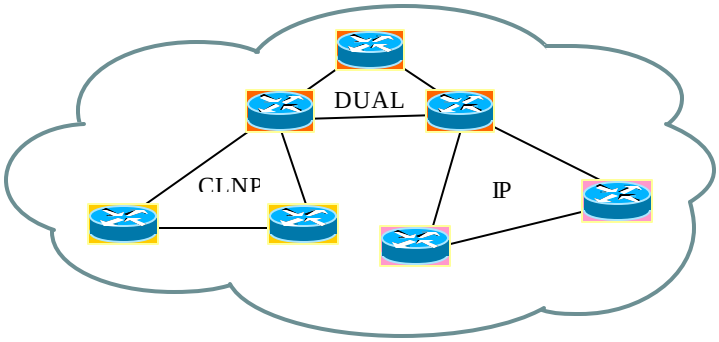


Figura 4.2

En la figura 4.1 se muestran un dominio de enrutamiento mixto, ya que este dominio esta compuesto por un área de enrutamiento que transporta trafico *IP*, por otra área que transporta trafico *CLNP* y por un área muy importante que permite el transporte de trafico *IP* y *CLNP*.

Áreas y jerarquías de enrutamiento

Un dominio de enrutamiento puede ser segmentado de acuerdo a las necesidades de escalamiento y administración de tráfico, a los segmentos del dominio se les conoce como áreas de enrutamiento. En la figura 4.3 se muestra un dominio que ha sido dividido en 3 áreas, el área 0001, el área 0002 y el área 0003.

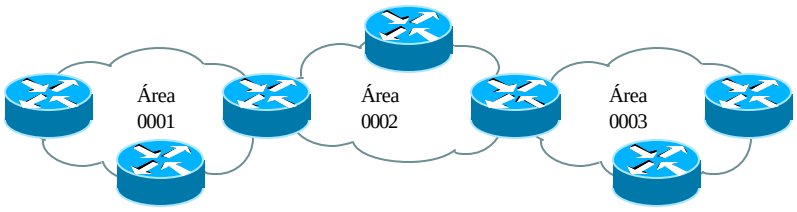


Figura 4.3

Para asignar las direcciones de área, *IS-IS* se auxilia del direccionamiento *CLNP* para identificar a los nodos o *routers* dentro del área o dominio de enrutamiento.

Las direcciones *CLNP*, son conocidas como direcciones *NSAP (Network Service Access Point)*, estas direcciones fueron vistas anteriormente en la sección del capitulo 2 llamada *"Direccionamiento CLNP"*. La figura 4.4 muestra el formato empleado por el protocolo *IS–IS* para las direcciones *NSAP*, las cuales básicamente se componen de dos partes, la primera es la parte inicial del dominio (*Initial Domain Part, IDP*) y la segunda es la parte especifica del dominio (*Domain Specific Part*).

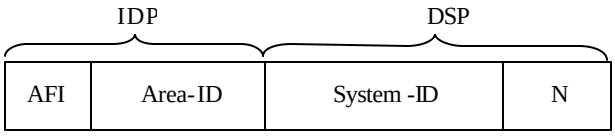


Figura 4.4

El campo *AFI* es utilizado para identificar al formato de la dirección y la autoridad que usara estas direcciones, existe un valor para redes privadas, concepto similar en *IP*.

El campo *Area-ID* es utilizado para identificar el área de enrutamiento.

El campo *System-ID* es utilizado para identificar al sistema, que en este caso se refieren a los routers como sistema, cada *router IS-IS* solo debe tener un *System-ID*.

El campo *N* es para seleccionar el servio de red que requiere el usuario.

Los valores de la dirección *NSAP* son escritos con nomenclatura hexadecimal y la dirección *NSAP* también es conocida como *"Titulo de entidad de red" (Network Entity Title, NET).*

Para la implementación de *IS–IS* en un dominio segmentado es necesario tener un área principal, conocida como *backbone*, la cual debe tener comunicación directa con todas las otras áreas existentes dentro del dominio estableciendo una división jerárquica de áreas de enrutamiento.

La figura 4.5 muestra una red segmentada en 4 áreas denominadas como: A, B, C y D, donde el área de *backbone* es el área A ya que es la que esta directamente conectada con las otras 3 áreas.

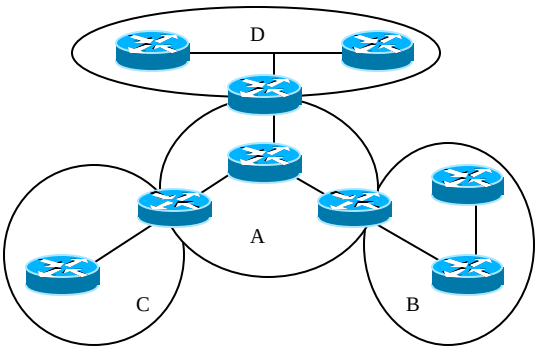


Figura 4.5

La comunicación para intercambiar información sobre las bases de datos y tablas de enrutamiento entre *routers* es clasificada de acuerdo a las áreas que intercambien información, existen tres niveles de comunicación, por cuestiones de literatura, los niveles de enrutamiento serán denotados como *"Level"* y son los siguientes:

- *Level 2*
- *Level 1-2*
- *Level 1*

La comunicación *"Level 2"* es la de más alto nivel de comunicación y se da solamente entre *routers* que estén dentro de área de *backbone*. El la figura 4.5 este nivel de comunicación se daría entre *routers* dentro del área A

La comunicación *"Level 1-2"* es la de un nivel mas bajo pues que esta se da entre *routers* del área de *backbone* y *routers* en áreas mas allá del *backbone*, es decir se da entre *routers* que conectan ambas áreas. En la figura 4.5 este nivel de comunicación se establece entre los *routers* que conectan el área B con la A, en la C con la A y en la D con la A.

La comunicación *"Level 1"* es la de mas bajo nivel y se da solamente entre *routers* que no estén conectados con el *backbone*, solo *routers* dentro de la misma área. En la figura 4.5 este nivel de comunicación se daría entre los dos *routers* que están dentro del área B

Intermediate System y End System

El modelo *OSI* empleo el término *End System (ES)* para referirse a una *PC* o *host* dentro de una red y el término *Intermediate System (IS)* para referirse a un *router* que permita la comunicación entre varias redes. De acuerdo a esta terminología *OSI* desarrollo el protocolo *ES–IS* para la comunicación entre *host* y *router* y el protocolo *IS–IS* para la comunicación entre *routers*.

Ahora que han sido definidos los conceptos básicos del protocolo *IS–IS*, se presenta la figura 4.6, de donde se aprecia la ubicación de cada uno de los conceptos que fueron mencionados con anterioridad para el protocolo *IS–IS*.

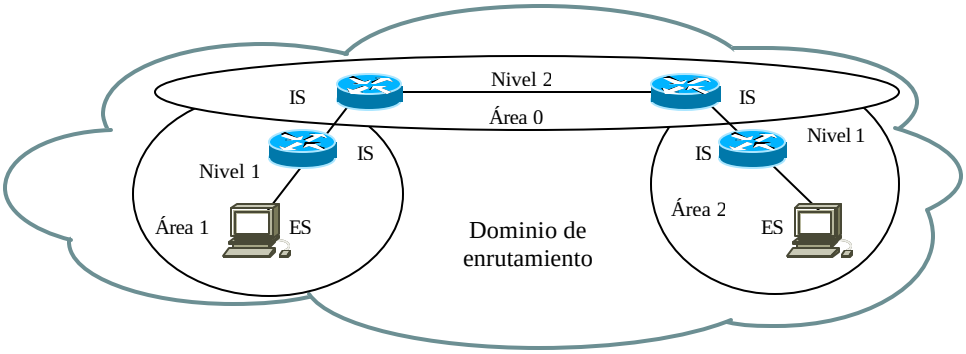


Figura 4.5

Paquetes IS-IS

Los protocolos no orientados a conexión como *CLNP* e *IP*, transmiten datos en forma de paquetes. En el estándar *ISO10598* referido a *IS-IS*, los paquetes son mencionados como unidades de datos del protocolo (*Protocol Data Unit*, *PDU*) pero en esta tesis se refiere a ellos como *"paquetes"* para tener una mejor concordancia con la literatura de *IP*, ya que múltiples tipos de paquetes son usados en ambientes no orientados a conexión.

La parte fundamental y el alma de *IS–IS* son sus paquetes y en esta sección se describirán detalladamente los tres paquetes que requiere *IS–IS* para su funcionamiento:

- *Mensajes Hello*
- *Paquetes Link–State*
- *Paquetes Sequence Number*

Una parte del encabezado de los paquetes *IS–IS* se presenta en forma constante en todos los paquetes *IS–IS* y es llamada *"Parte común del paquete IS–IS"*. Existe una parte del paquete que varia dependiendo del tipo de mensaje que requiera ser enviado y es llamada *"Parte especifica del paquete IS–IS"*. Una última parte del paquete *IS–IS* es una parte que varia en cuanto a la longitud que ocupara dentro del paquete y es llamada *"Parte variable del paquete IS–IS"*. La figura 4.6 muestra la composición del paquete *IS–IS* de acuerdo a las partes arriba mencionadas.

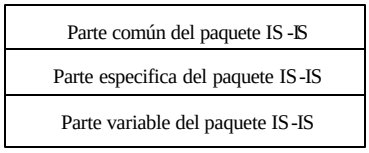


Figura 4.6

La figura 4.7 presenta la estructura que aparece en forma constante dentro del paquete *IS–IS* denominada como "Parte común del paquete *IS–IS"*.

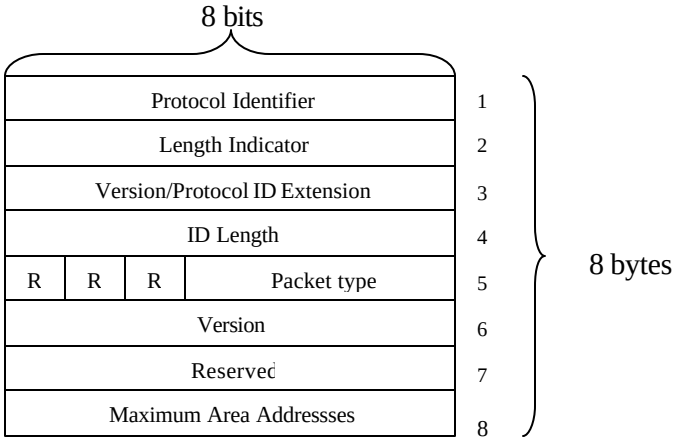


Figura 4.7

A continuación se describen los campos que componen la parte común del pa quete.

Protocol Identifier – Identifica al protocolo que será empleado para el envío de paquetes, ISO asigno el valor de *0x83* en hexadecimal para el protocolo *IS-IS*,

Length Indicator – Indica la longitud que tendrá la parte común del paquete, este es un valor fijo de 8, lo cual significa que tiene un tamaño de 8 bytes.

Version – Específica la versión de *IS-IS*, la versión actual es 1.

ID length – Específica la longitud del campo *System–ID*. Si el valor esta entre 1 y 8, el campo *System–ID* adquirirá el valor asignado en bytes. Si el valor es de cero, el valor del campo *System–ID* será de 6 bytes.

Reserved – Los campos denominados con la letra *R* están reservados para uso futuro y por lo tanto siempre tendrán el valor de cero

Packet Type – Específica el tipo de paquete IS-IS que será utilizado.

Maximum Address Area – Específica el número máximo de áreas.

La parte especifica del paquete depende directamente del valor que sea asignado en el campo *"Packet Type"* ya que ahí se especifica el tipo de paquete que esta siendo enviado, en la tabla 4.1 se muestran la clasificación de los distintos tipos de paquetes que pueden ser enviados, así como sus valores respectivos.

.

Paquete	Tipo de Paquete	Valor
Hello (Hello message)	Level 1 LAN	15
	Level 2 LAN	16
	Point to point	17
Link State Packet (LSP)	Level 1	18
	Level 2	20
Sequence Number Packets (SNP) Complete	Level 1 complete	24
	Level 2 complete	25
Sequence Number Packets (SNP) Partial	Level 1 partial	26
	Level 2 partial	27

Tabla 4.1

La parte variable del paquete es mejor conocida como *"Type Length Value"* debido a la estructura que sigue este campo del paquete. La arquitectura que es empleada para la parte variable es un formato general, solamente modificado por el tamaño del paquete, la figura 4.8 muestra el formato de la parte variable del paquete *IS–IS*.

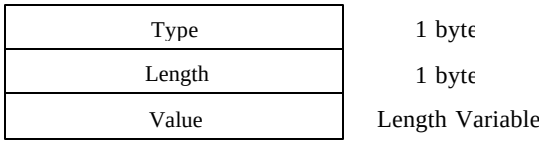


Figura 4.8

De la figura 4.8 se obtiene la siguiente descripción de los campos *Type*, *Length* y *Value* respectivamente.

Type – Este campo identifica al tipo de parte variable que complementara al paquete *IS–IS* de acuerdo a las necesidades requeridas.

Length – En este campo se especifica la longitud del campo variable.

Value – En este campo son asignados los valores requeridos para llevar a cabo establecer un correcto y eficaz enrutamiento de paquetes, el tamaño de este campo puede variar de 0 a 255 bytes.

A continuación se presenta la tabla 4.2, dentro de la cual se pueden apreciar los diferentes paquetes *TLV* empleados en el enrutamiento de paquetes OSI de acuerdo a las especificaciones *ISO 10589*.

TLV	Type
Area Address	1
Intermediate System Neighbors	2
End System Neighbors	3
Partition Designated Level 2 IS	4
Prefix neighbors	5
Intermediate System Neighbors	6
Not specified	7
Padding	8
LSP Entries	9
Authentication Information	10

Tabla 4.2

La tabla 4.3 muestra los tipos de *TLV's* empleados en el enrutamiento de paquetes IP de acuerdo al *RFC 1195*.

TLV	Type
IP Internal Reachiability Information	128
Protocols Soported	129
IP External Reachiability Information	130
Interdomain Routing Protocol Information	131
IP Interface Address	132
Authentication Information	133

Tabla 4.3

Después de la descripción de las partes que componen a un paquete *IS–IS*, es necesario describir con mayor detalle las estructuras y funcionamiento de cada paquete especifico, además de los paquetes *TLV* que serán requeridos en cada caso particular.

Mensajes Hello

Estos paquetes conocidos como mensajes de "*Hola*" son usados por los *routers* para informar de la presencia de un nuevo *router* dentro de la red a otros *routers* y con ello establecer adyacencias o una vecindad con los otros *routers*.

Estos paquetes dependen del tipo de conexión física que sea empleada, para la conexión mediante tecnología "*Punto a punto*" se emplea el formato de la figura 4.4.

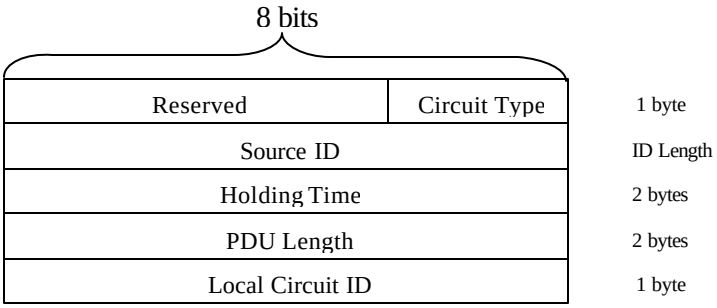


Figura 4.4

Existe un campo reservado de 6 bits al comienzo de los mensajes de *"Hola"*.

Circuit Type – Indica al nivel de comunicación que maneja el *router* fuente del mensaje de *hola*, el valor de *01* es para los *routers "Level 1"*, el valor *10* es para los *routers "Level 2"* y por ultimo, el valor *11* es para los *routers "Level 1-2"*.

Source ID – Identifica al *System–ID* que esta originando el mensaje.

Hold time – Específica el tiempo de espera que un mensaje de *"Hola"* permanecerá en la red antes de ser descartado.

PDU Length – Indica la longitud total del paquete.

Local Circuit ID – Identificador único del circuito o enlace.

Para las conexiones mediante *"Ethernet"*, el último campo es remplazado por 3 campos, estas modificaciones se pueden apreciar en la figura 4.5.

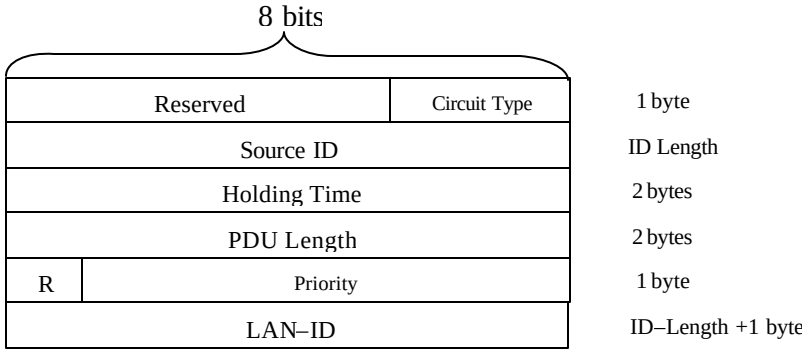


Figura 4.5

Se reserva el primer *bit* después del campo *PDU Length* para uso futuro.

Priority – Este campo designa la prioridad que tendrá el *router* fuente para ser empleado como nodo principal o *Pseudonodo* dentro de una red conectada vía *Ethernet*.

LAN-ID – Identifica la conexión mediante el *System–ID* más un numero único.

Para complementar el mensaje de *"Hola"* se requiere agregar algún paquete *TLV*. Los paquetes *TLV* utilizados para estos mensajes son los siguientes:

- *Area Address*.
- *Intermediate System Neighbors*
- *Padding*
- *Authentication*.

A Continuación se describe cada paquete TLV usados en el envío de mensajes de *"Hola"* y la composición del mismo:

Area Address TLV – Contiene las direcciones de área configuradas en el *router*

Type =1
Length = Tamaño total del campo *Value*.
Value = 1 byte tamaño de la dirección + dirección del área variable.

IS–Neighbors TLV – Contiene las direcciones MAC de los vecinos que requieran iniciar la comunicación.

Type = 2
Length = 1 Tamaño total del campo *Value*...
Value = 1 byte virtual + *n* múltiplos de (4 bytes de información de métricas de los vecinos + *System–ID* del vecino + 1 byte del *Pseudonode–ID*).

Authentication TLV – Contiene información confidencial

Campo Type = 10
Campo Length = Tamaño total del campo *Value*.
Campo Value = Este campo esta compuesto de dos partes:

- *Tipo de autenticación* – El valor 0 y 2 – 254 están reservados, el valor 1 indica contraseña no encriptada y el valor de 255 indica un amplio dominio de autenticación.
- *Contraseña de autenticación* – Para la autenticación no encriptada significa que la contraseña puede tener un valor de hasta 254 bytes.

.Paquetes Link–State (LSP)

Los paquetes de este tipo, son la parte fundamental de los protocolos *link–state*, una vez que los *routers* han descubierto a sus vecinos y establecido las adyacencias satisfactoriamente se comienza el intercambio de información de rutas.

La información de enrutamiento es enviada a toda la red por medio de estos paquetes. Como resultado del intercambio de información entre *routers* esta la actualización de las bases de datos, la actualización del estado de los enlaces y la actualización del mapa topológico de la red.

Todos los paquetes *link–state* (*LSP*) deben contener la información necesaria para que el *router* (*IS*) sea capaz de mover información correctamente a través de la red

El formato de un *LSP* es presentado en la figura 4.6 y sus campos son descritos a continuación:

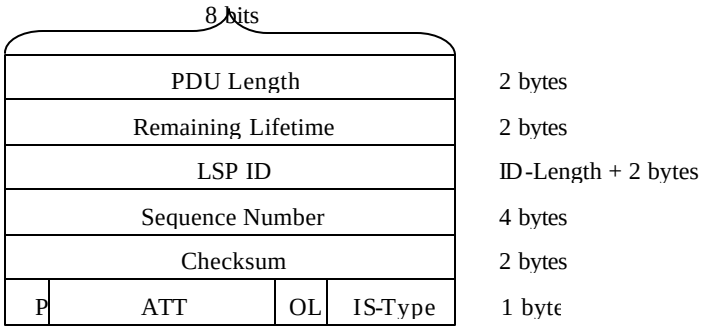


Figura 4.6

PDU Length – Específica el tamaño del paquete en bytes.

Remaining Lifetime – Especifica el tiempo que el paquete permanece en la red antes de expirar.

LSP-ID – Identifica al paquete *LSP* en relación al *router* fuente, la figura 4.7 muestra la conformación del campo *LSP–ID*.

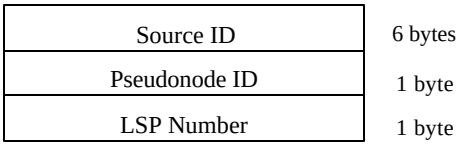


Figura 4.7

Source–ID – Es el identificador o *System–ID* del *router* fuente.

Pseudonode–ID – Es el identificador del nodo principal en redes *Ethernet*.

LSP Number – Este número se refiere a los fragmentos de un paquete *LSP*.

Sequence Number – Indica el numero de secuencia del paquete.

Checksum – Revisa que el paquete no haya sido dañado durante el transcurso del camino.

P (Partition) – Este *bit* solo es usado para especificar si el *router* soporta la reparación de paquetes.

ATT (Attachment) – Indica si el *router* origen esta conectado a una o múltiples áreas e indica las métricas que soporta, la tabla 4.4 muestra las métricas soportadas.

Métrica	Bit
Predeterminado	4
Retardo	5
Costo	6
Error	7

Tabla 4.4

OL (Overload) – Este campo indica si la base de datos del *router* fuente esta llena.

IS-Type – Identifica el nivel de comunicación del *router* fuente, el valor de *01* es para los *routers* "*Level 1*", el valor *10* es para los *routers* "*Level 2*" y por ultimo, el valor *11* es para los *routers* "*Level 1-2*".

Los *LSP* requieren que se les agregue un paquete *TLV*, los *LSP* soportan dos clasificaciones de paquetes *TLV*, una es para los paquetes *"Level 1"* y la otra es para los paquetes *"Level 2"*. Los paquetes *"Level 1"* son presentados en la tabla 4.5 y descritos posteriormente.

TLV	Type
Area Address	1
Intermediate System Neighbors	2
End System Neighbors	3
Authentication Information	10
IP Internal Reachability Information	128
Protocols Soported	129
IP Interface Address	132

Tabla 4.5

Area Address TLV – Contiene las direcciones de área configuradas en el *router*

Type =1
Length = Tamaño total del campo *Value*.
Value = 1 byte tamaño de la dirección + dirección del área variable.

IS–Neighbors TLV – Contiene las direcciones *MAC* de los vecinos que requieran iniciar la comunicación.

Type = 2
Length = 1 Tamaño total del campo *Value*...
Value = 1 byte virtual + *n* múltiplos de (4 bytes de información de métricas de los vecinos + *System–ID* del vecino + 1 byte del *Pseudonode–ID*).

ES–Neighbors TLV – Contiene información acerca de las estaciones de trabajo conectadas a los *routers*.

Type = 3
Length = Tamaño total del campo *Value*.
Value = Métrica común +múltiplos del *System–ID*.

Authentication TLV – Contiene información confidencial

Type = 10
Length = Tamaño total del campo *Value*.
Value = Este campo esta compuesto de dos partes:

- *Tipo de autenticación* – El valor 0 y 2 – 254 están reservados, el valor 1 indica contraseña no encriptada y el valor de 255 indica un amplio dominio de autenticación.
- *Contraseña de autenticación* – Para la autenticación no encriptada significa que la contraseña puede tener un valor de hasta 254 bytes.

IP Internal reachability TLV – Este paquete almacena únicamente los prefijos *IP* que están directamente conectados

Type = 128
Length = Tamaño total del campo *Value* (múltiplos de 12 bytes)
Value = Requiere de 12 bytes por prefijo para llevar información de la métrica del prefijo; 4 bytes son para información acerca de la métrica, los siguientes 4 bytes son para información del prefijo *IP* y los últimos 4 bytes son para indicar la mascara de subred del prefijo *IP*.

Protocols Soported TLV – Este paquete identifica los protocolos que están siendo soportados por *IS–IS*.

Type = 129
Length = Tamaño total del campo *Value*.
Value = *A*signa un identificador para el protocolo con el que vaya a trabajar, para identificar al protocolo *CLNP* se tiene el identificador 0x81 y para el protocolo *IP* se tiene el identificador 0xCC

IP Interface Address TLV – Este paquete puede tener una o más direcciones *IP* configuradas.

Type = 132
Length = Tamaño total del campo *Value*.
Value = requiere de 4 bytes como mínimo para una dirección *IP*.

La figura 4.8 presenta una muestra de la salida de un comando en un *router* donde se pueden apreciar los parámetros referentes a una *LSP*.

```
router-1>show isis database detail

IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
router-2.00-00        0x0000C836   0x86FF        887               0/0/0
  Area Address: 49.0205
  NLPID:        0xCC 0x81
  Hostname: router-2
  IP Address:   200.0.205.5
```

Figura 4.8

Los paquetes *TLV* empleados con los *LSP "Level 2"* son mostrados en la tabla 4.6 y descritos a continuación.

TLV	Type
Area Address	1
Intermediate System Neighbors	2
Partition Designated Level 2 IS	4
Prefix Neighbors	5
Authentication Information	10
IP Internal Reachability Information	128
Protocols Soported	129
IP External Reachability Information	130
Interdomain Routing Protocol Information	131
IP interface Address	132

Tabla 4.6

Area Address TLV – Este paquete es el mismo que se usa para los *LSP "Level 1"*.

IS Neighbors TLV – Este paquete es el mismo que se usa para los *LSP "Level 1"*.

Partition Designated Level 2 IS TLV – Este paquete puede establecer conexiones virtuales de áreas fuera del backbone hacia el mismo.

Type = 4
Length = Tamaño del *System–ID*.
Value = *System–ID* designada por el *router* "*Level 2*".

Prefix Neighbors TLV – Reúne información acerca de los prefijos alcanzables.

Type = 5
Length = Tamaño total del campo *Value*.
Value = Consta de 4 bytes para información de métrica + múltiplos del tamaño de la dirección del prefijo + la dirección del prefijo.

Authentication Information TLV – Este paquete es el mismo que se usa para los LSP *"Level 1"*.

IP Internal reachability TLV – Este paquete es el mismo que se usa para los *LSP "Level 1"*.

Protocols Soported TLV – Este paquete es el mismo que se usa para los *LSP "Level 1"*.

IP External reachability TLV –Este paquete reúne información acerca de rutas obtenidas de otro protocolo de enrutamiento distinto a *IS–IS*.

Type =130
Length = *n* x 12, donde *n* es el numero de rutas externas.
Value = Múltiplos de 12 bytes para cada ruta externa, 4 bytes para información de métrica. 4 bytes para información del prefijo *IP* y 4 bytes para la mascara de subred del prefijo *IP*.

Interdomain routing protocol Information TLV – Este paquete permite la interacción de *IS–IS* con otros protocolos de interdominio, como *BGP*.

Type = 131
Length = Tamaño total del campo *Value*.
Value = Contiene información referente al número de sistema autónomos.

IP Interface Address TLV – Este paquete es el mismo que se usa para los *LSP "Level 1"*.

La figura 4.9 presenta una muestra de la salida de un comando en un *router* donde se pueden apreciar los parámetros referentes a una *LSP "Level 2"*.

```
router-2>show isis database detail

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
router-1.00-00        0x0000C89D   0xB767        654               0/0/0
  Area Address: 49.0205
  NLPID:        0xCC 0x81
  Hostname: router-1
  IP Address:   200.0.205.5
```

Figura 4.9

Paquetes Sequence Number (SNP)

Mientras que los mensajes de *"Hola"* se encargan de establecer y mantener las adyacencias entre los *routers* vecinos y los paquetes *LSP* son los vehículos para compartir la información de enruta miento, bs *SNP* son los paquetes utilizados como mecanismos auxiliares para asegurar la integridad de los paquetes que llevan la información de enrutamiento.

En los ambientes *Link–State*, los *routers* reciben todas las actualizaciones necesarias de los vecinos que estén dentro de la misma área a través de la interfaz que los tenga directamente conectados, a este proceso de envío de paquetes hacia todos los *routers* conectados se le llama *"Flooding" o" Inundado"*.

Por medio del *flooding*, todos los *routers* dentro de la misma área construyen en forma sincronizada una base de datos idéntica, esto quiere decir, que las actualizaciones se deben ir haciendo en forma sincronizada mediante algún mecanismo de sincronización y es ahí donde aparecen los paquetes *SNP* para jugar un papel muy importante en la sincronización de actualizaciones. Existen dos tipos de paquetes *SNP*:

- *Complete Sequence Number Packet (CSNP)*
- *Partial Sequence Number Packet (PSNP)*

Estos paquetes lleva n la información de enrutamiento resumida. La diferencia básica entre los dos tipos de paquetes es que los *CSNP* llevan completa la información de la base de datos de los paquetes *LSP* mientras que los *PSNP* solo llevan parcialmente la información de los paquetes *LSP*.

Para construir los paquetes *SNP* es necesario extraer información clave de los paquetes LSP, la información extraída es:

- *Link–State Packet Identifier (LSP–ID)*
- *Sequence Number*
- *Checksum*
- *Remaining Life Time*

La estructura del paquete *CSNP* es el mismo, para la comunicación *"Level 1"* y *"Level 2"* y es mostrada en la figura 4.10.

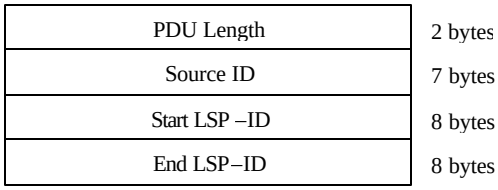


Figura 4.10

PDU Length – Indica el tamaño del paquete.

S*ource–ID* – Identifica al *router* origen mediante el *System–ID*.

Start LSP–ID – Identifica al primer paquete *LSP*.

End LSP–ID – Identifica al ultimo paquete *LSP.*

La estructura para los paquetes *PSNP* es mostrada en la figura 4.11 y el formato del paquete es el mismo sin importar el tipo de comunicación.

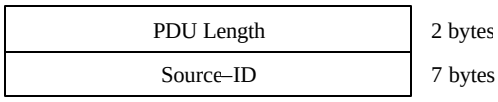


Figura 4.11

PDU Length – Indica el tamaño del paquete.

S*ource–ID* – Identifica al *router* origen mediante el *System–ID*.

Los paquetes *TLV* empleados para el envió de paquetes *CSNP* son los mostrados en la tabla 4.7.

TLV	Tipo
LSP Entries	9
Authentication Information	10

Tabla 4.7

LSP Entries TLV – En este paquete se reúne la información resumida de los *LSP* de todos los *LSP* conocidos por el router que esta enviando actualizaciones.

Type = 9
Length = Tamaño total del campo *Value*.
Value = asigna 2 bytes para *LSP remainning life time*, 8 bytes para el *LSP–ID*, 4 bytes para el *LSP Sequence Number* y 2 bytes para el *LSP Checksum*.

Authentication TLV – Contiene información confidencial

Type = 10
Length = Tamaño total del campo *Value*.
Value = Este campo esta compuesto de dos partes:

- *Tipo de autenticación* – El valor 0 y 2 – 254 están reservados, el valor 1 indica contraseña no encriptada y el valor de 255 indica un amplio dominio de autenticación.
- *Contraseña de autenticación* – Para la autenticación no encriptada significa que la contraseña puede tener un valor de hasta 254 bytes.

Los paquetes *TLV* empleados para el envió de paquetes *PSNP* son los mostrados en la tabla 4.8.

TLV	Tipo
LSP Entries	9
Authentication Information	10

Tabla 4.8

LSP Entries TLV – Este paquete es el mismo que fue presentado para los *CSNP*.

Authentication TLV – Este paquete es el mismo que fue presentado para los *CSNP*.

Soporte para IPv6

Con el crecimiento de Internet y el desarrollo de un nuevo protocolo para Internet llamado *"IPv6"*, descrito en el capitulo 2, se requería que se tuviera el soporte necesario para que los protocolos de enrutamiento pudieran enrutar los paquetes de *IPv6*.

La *IETF* se encargo de elaborar un documento que permitiera el soporte para enrutar paquetes de *IPv6* mediante el protocolo *IS–IS*, la *IETF* elaboro un documento llamado *"Routing IPv6 with IS–IS"*.

Ya que IS –IS es un protocolo *Link–State* y cada *router* genera paquetes *LSP* que contienen información de cada *router*, además de generar paquetes *TLV* que llevan información adicional. Para que *IS–IS* pudiese soportar *IPv6* fue necesario crear dos nuevos paquetes *TLV* que permitieran mover la información necesaria para *IPv6*, así como un identificador para el nuevo protocolo, los nuevos *TLV* son mostrados en la tabla 4.9 y también la modificación al *TLV 129*.

TLV	Tipo
Protocols Soported	129
IPv6 Interface Address	232
IPv6 Reachability	236

Tabla 4.9

Protocols Soported – Identifica al protocolo con el que va a trabajar, en este caso ha sido solamente modificado para tener el soporte de *IPv6* mediante el numero *0x8E*.

IPv6 Interface Address TLV – Este paquete solo es ampliado para soportar direcciones de 128 bits.

La estructura de este *TLV* se muestra en la figura 4.12.

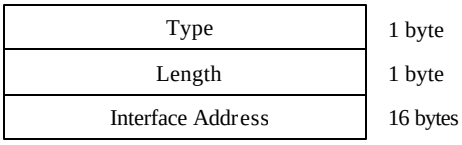


Figura 4.12

Type = 232
Length = Tamaño total del campo *Value*.
Value = En este campo esta la dirección *IPv6* de la interfaz, si la interfaz tiene mas de una dirección *IPv6*, esta también esta incluida en este campo (16 bytes por dirección *IPv6*).

IPv6 Reachability TLV – Este paquete resume los *TLV* usados en *IPv4 "IP Internal Reachability"* e *"IP External Reachability"* en un mismo paquete.

La estructura de este *TLV* es presentada en la figura 4.13.

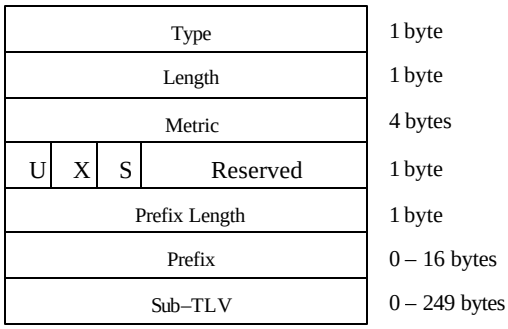


Figura 4.13

Type = 236
Length = Tamaño total del campo *Value*
Value = Este campo se descompone en 8 campos, descritos a continuación.
Metric – Contiene información de la métrica (4 bytes).
U – Indica si el prefijo esta siendo anunciada de un nivel más alto o bajo (1 bit).
X – Indica si el prefijo es externo (1 bit).
S – Indica si existen otros *Sub–TLV*.
Reserved – Es un campo reservado (5 bits).
Prefix Length – Indica la longitud del prefijo (1 byte).
Prefix – Contiene el prefijo (0 – 16 bytes).
Sub–TLV – Indica si algún *Sub–TLV* esta presente, si alguno existe, entonces el primer byte indica la longitud del *Sub-TLV* y los demás son para el *Sub-TLV*.

Un *Sub–TLV* tiene la misma estructura que un *TLV* y su función es proporcionar información adicional.

La figura 4.14 muestra la salida de el comando *"show isis database detail"*, donde se puede apreciar la información de *IPv6* contenida en las bases de datos de los *routers*.

```
router-3>show isis database detail

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime      ATT/P/OL
router-4.00-00        0x0000C9F9   0xFBC5        824               0/0/0
  Area Address: 49.0205
  NLPID:        0xCC  0x8E
  Hostname: router-4
  IP Address:   200.0.205.5
  IPv6 Address: 2001:1348::1005

  Metric: 10         IP 200.0.205.5/32
  Metric: 10         IP 200.0.206.128/28
  Metric: 0          IPv6 2001:798:2015:10AA::4/126
  Metric: 10         IPv6 2001:1348::4/126
  Metric: 10         IPv6 2001:1348::24/126
  Metric: 10         IPv6 2001:1348::1005/128
  Metric: 0          IPv6 2001:1348:1:1::/64
```

Figura 4.14

Funcionamiento de IS–IS

El protocolo *IS–IS* puede trabajar en ambientes *"Punto a punto"* o en ambientes *"Ethernet"*, la figura 4.15 muestra un ambiente *"Punto a punto"*, a partir e la cual se describe el funcionamiento de *IS–IS*.

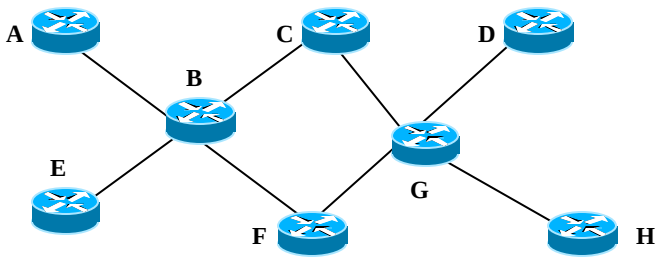


Figura 4.15

Todos los *routers* de la figura 4.13 deben inundar toda la red con mensajes de *Hola* hacia las interfaces que tengan activas para anunciar al *router* dentro de la red y con ello establecer las adyacencias necesarias para comenzar a recabar información y así llenar sus bases de datos y crear con ello el mapa topológico de la red.

Ya que se hayan establecido las adyacencias con los demás *routers* entonces se envían los datos de cada base de datos de cada *router* hacia los demás *routers* mediante los paquetes *Link–State*.

Con el comienzo de las actualizaciones de las bases de datos hacia todos los *routers* también se comienza la sincronización de todas las bases de datos hasta que todos los *routers* tengan una idéntica base de datos de toda la red.

En la figura 4.16 se muestra un ambiente *Ethernet*, a partir del cual se describe el funcionamiento de IS-IS en ese tipo de ambiente.

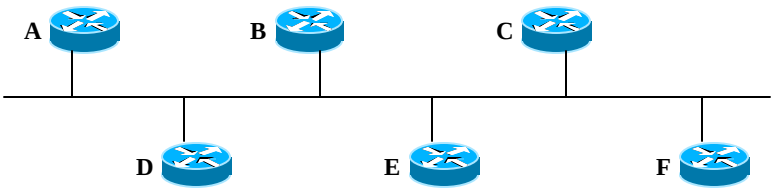


Figura 4.16

En el ambiente de la figura 4.16 existe una diferencia en comparación con el ambiente *punto a punto* y es que después de la inundación de mensajes de *Hola* se selecciona un *pseudonodo* que haría tener el siguiente ambiente lógico, figura 4.17.

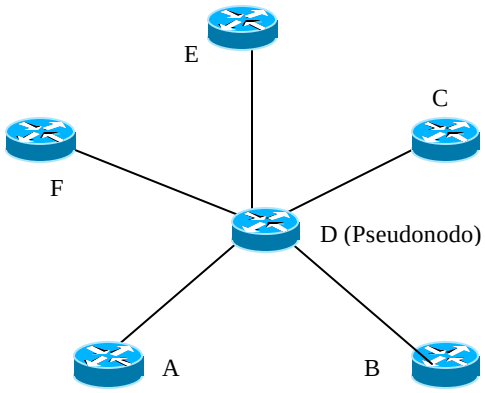


Figura 4.17

Al tener el esquema lógico de la figura 4.17 traería como consecuencia la creación del mapa topológico teniendo como raíz al *router D* y siendo este el origen de todos los paquetes que requieran ser enviados a toda la red.

Conclusiones

Conclusiones

Se concluyo la propuesta para implementar el protocolo *IS–IS* en la red *CUDI,* la cual fue presentada ante el grupo de enrutamiento de *CUDI* para ser analizada y aprobada para su posterior publicación en la página del grupo.

La implementación del protocolo *IS-IS* actualmente se encuentra postergada a requerimientos de software, se espera que la adquisición de éste sea en un lapso aproximado de un mes y posteriormente llevar a cabo la migración de *OSPF* hacia *IS–IS* dentro del *backbone* y con ello tener una mayor cobertura para *IPv6*

Al elaborar esta tesis se logro realizar un documento en el cual se puede compartir la experiencia en *IS–IS* con estudiantes e ingenieros que deseen implementar este protocolo o simplemente comprender su mecanismo para la transmisión de información a través de la red académica de México.

Algo relevante que se observo al realizar este trabajo de investigación es que existen muchos temas que pueden emanar del mismo y que pueden ser investigados, documentados y publicados para la comunidad de las telecomunicaciones.

Bibliografia

Libros

IP Routing Primer PLUS
Heather Osterloh

CCNP Routing
Eric McMasters
Brian Morga
Mike Shroyer

Cisco Router Configuration & Troubleshooting
Mark Tripod

IS –IS Network Design Solutions
Abe martey
Scott Sturguest

Cisco Router Handbook
George C Sackett

Cisco Router Networking
Paul Ammann

Routing TCP/IP
Jeff Doyle

IP Switching and Routing Essentials
Stephen Tomas

IP Routing Fundamentals
Mark Sportack

IPng, Internet Protocol next generation
Scott O. Bradner
Allison Mankin

IPv6 Clearly explained
Peter Loshin

IPv6 Essentials
Silvia Hagen

TCP-IP and related protocols
Uyless Black

Internetworking with TCP/IP
Douglas Comer

Paginas web

www.abilene.iu.edu
www.internet2.edu
www.cudi.edu.mx
www.unam.mx
www.noc.cudi.edu.mx
www.noc.redclara.net
www.cisco.com
www.juniper.net
wgrouting.internet2.unam.mx
www.ciscopress.com
www.6bone.net
www.ipv6.org
www.geant.net
www.rediris.es
www.consulintel.es
www.rfc-editor.org
www.ietf.org
www.nanog.org

Apendices

Apéndice A: Extracto de la propuesta presentada en CUDI.

Apéndice B: Configuración de los routers de la red CUDI.

Apéndice C: Glosario.

Extracto de la propuesta presentada en *CUDI*

Introducción

Para el enrutamiento de redes *IPv6* dentro del *backbone* de la red *CUDI*, se tenía que implementar un protocolo que diera soporte a estas redes. Para el inicio de la red, solamente existía el protocolo *RIP* con soporte para redes *IPv6*. *RIP* es un protocolo *vector–distancia* que debido al crecimiento de las redes, es un protocolo no es escalable.

Para inicios de este año se comenzó a buscar un protocolo que respondiera a las necesidades actuales de la red y que permita en crecimiento de la red educativa.

En este documento se describe el motivo de la propuesta de migración del protocolo de enrutamiento interno, la selección de un nuevo protocolo que permita el enrutamiento de redes *IPv6* mediante un protocolo *link–state*, las características del protocolo seleccionado.

Este documento también describe la forma de configuración de *IS-IS* para soportar el enrutamiento de redes *IPv6*, así como una muestra de cómo quedarían las configuraciones de los *routers* con el protocolo *IS-IS* ya funcionando.

Mapa topológico

El siguiente mapa nos muestra la topología actual de la red *CUDI*. Esta cuenta actualmente con 9 nodos distribuidos en todo el país.

Motivo de migración

. En la actualidad se cuenta con el protocolo *OSPF* para enrutar redes *IPv4* y con el protocolo *RIP* para enrutar redes *IPv6*. Debido al crecimiento de las redes *IPv6*, los fabricantes están desarrollando nuevas tecnologías para el soporte de *IPv6* y algunas redes educativas a nivel mundial ya las están implementando.

El funcionamiento del protocolo *RIP* se basa en un algoritmo *vector–distancia* (conteo de saltos). *RIP* no es escalable para redes grandes además de que el desarrollo

de los protocolos estado–enlace han superado a los protocolos vector–distancia por la rápida convergencia y mayor escalabilidad.

Dadas las características del protocolo *RIP*, es preferible implementar un protocolo de enrutamiento interno basado el algoritmos *estado–enlace*. Actualmente existen dos opciones de protocolos a implementar:

- *IS-IS con IPv6*
- *OSPFv3*

Protocolo propuesto

El protocolo propuesto se llama *IS-IS (Integrated System to Integrated System)*, también conocido como *Integrated IS-IS* para el enrutamiento de redes *IP* y redes *OSI*, el cual incluye nuevos estándares para el soporte de *IPv6*, *MPLS* e *Ingeniería de tráfico*.

Para la selección de *IS-IS* cabe señalar que se tomaron en cuenta aspectos operativos además de los aspectos técnicos.

Los aspectos que han sido tomados en cuanta para la selección del protocolo *IS-IS* sobre *OSPFv3* son las siguientes:

- *IS–IS* desempeña un proceso para el para el establecimiento de adyacencias, ya sean de *IPv4* o *IPv6* mientras que *OSPF* mantiene dos procesos, uno para redes *IPv4* y otro para redes *IPv6*.
- Existe mayor literatura acerca de la implementación y soporte para el protocolo *IS-IS* que para *OSPFv3*.
- Existe experiencia en cuanto a soporte del protocolo *IS-IS* en redes educativas que colaboran con la red *CUDI*, como son: *ABILENE, GEANT, CLARA* y también de algunos proveedores de Internet.
- *IS-IS* maneja dos niveles de enrutamiento (*Nivel 1 y Nivel 2*), lo cual permite tener un área de *backbone* mas flexible a diferencia del manejo de áreas en *OSPF*.

Características de IS–IS

La configuración de *IS-IS* es sencilla, para llevar a cabo su proceso de configuración hay que determinar la dirección *NSAP (Network Service Access Point)* que será configurada en cada *router*, ya que esta dirección permitirá identificar a cada nodo o *"Intermediate System"* como es conocido en el protocolo *IS-IS*..

Las direcciones *NSAP* tienen un formato principal, que será el que seguiremos para la conformación de las direcciones *NSAP* necesarias.

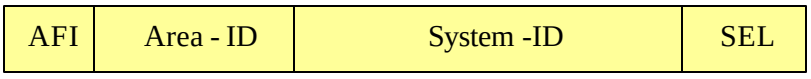


Figura 1

El campo *AFI* esta determinado por organizaciones internacionales, pero se tiene un valor asignado para las redes privadas, 49.

El valor del *Area–ID* es determinado por el administrador de la red y es definido en 4 valores hexadecimales. Este campo identifica el área de enrutamiento, para la red *CUDI* hemos propuesto el valor de 1999 en alusión al año de fundación de *CUDI*.

El valor de *System–ID* es determinado igualmente por el administrador y no debe ser repetido, ya que identifica a cada IS como único dentro del área.

El campo *SEL* es para identificar si el equipo hará tareas de enrutamiento, para nuestro caso adquirirá el valor de 00, ya que este identifica a los *routers*.

El direccionamiento para los routers de la red *CUDI* se hará tomando como referencia el identificador de cada *router* y quedara de la siguiente forma, tabla 1.1.

NODO	Router-ID	System-ID
México (Telmex)	200.23.60.5	2000.2306.0005
Monterrey (Avantel)	200.23.60.33	2000.2306.0033
México (Avantel)	200.23.60.65	2000.2306.0065
Tijuana	200.23.60.105	2000.2306.0105
Monterrey (Telmex)	200.23.60.129	2000.2306.0129
Juárez	200.23.60.161	2000.2306.0161
Guadalajara	200.23.60.201	2000.2306.0201
Cancún	200.23.60.94	2000.2306.0094

Tabla 1.1 *Systems –ID's*

Configuración en el router

La activación de *IS-IS* con *IPv6* en un *router* se realiza mediante los siguientes comandos:

1. enable
2. configure terminal
3. router isis backbone
4. net 49.1999.2000.2306.0xxx.00
5. is-type level-2-only
6. metric-style wide level-2
7. passive-interface [type number]
8. exit
9. interface [type number]
10. ip router isis backbone
11. ipv6 address ipv6-prefix/prefix-length [eui-64]
12. ipv6 router isis backbone
13. exit

La versión completa de la propuesta de implementación de IS-IS puede ser encontrado en la pagina *http://wgrouting.internet2.unam.mx*.

Configuración de los *routers* de la red *CUDI*

A continuación se muestra la configuración de los *routers* de la red *CUDI* para que puedan soportar *IPv6* nativo y el protocolo *IS–IS* en el *backbone*.

```
!
interface ATM1/0.1 point-to-point
 description PVC A GDL (POP)
 mtu 9180
 ip address 200.23.60.225 255.255.255.252
 ip pim sparse-dense-mode
 atm pvc 3 10 210 aal5snap inarp
 ip router isis backbone
 ipv6 address 2001:448:3:65::2/64
 ipv6 router isis backbone
!
!
interface ATM1/0.10 point-to-point
 description AVANTEL-MEX (POP)
 ip address 200.23.60.229 255.255.255.252
 ip pim sparse-mode
 atm pvc 12 8 212 aal5snap inarp
 ip router isis backbone
 ipv6 address 2001:448:3:62::1/64
 ipv6 router isis backbone
!
router isis backbone
net 49.1999.2000.2306.0005.00
is-type level-2-only
metric-style wide level-2
```

Configuración del router Telmex–México

```
!
interface POS1/0/0
 description ENLACE A MEXICO AVANTEL
 ip address 200.23.60.246 255.255.255.252
 ip router isis backbone
 ipv6 address 2001:448:3:61::1/64
 ipv6 router isis backbone
!
!
interface ATM2/0/0.1 point-to-point
 description ENLACE ROUTER MTY-TELMEX
 ip address 200.23.60.238 255.255.255.252
 no ip directed-broadcast
 pvc 7/209
 encapsulation aal5snap
 ip router isis backbone
 ipv6 address 2001:448:3:69::2/64
 ipv6 router isis backbone
 !
 !
router isis backbone
net 49.1999.2000.2306.0033.00
is-type level-2-only
metric-style wide level-2
```

Configuración del router Avantel–Monterrey

```
!
interface POS1/0/0
 description Enlace a MTY-AVANTEL (POP)
 ip address 200.23.60.245 255.255.255.252
 ip pim sparse-mode
 ip router isis backbone
 ipv6 address 2001:448:3:61::2/64
 ipv6 router isis backbone
!
interface POS1/0/1
 description Enlace Cancun
 ip address 200.23.60.93 255.255.255.252
 ip router isis backbone
 ipv6 address 2001:448:3:68::1/64
 ipv6 router isis backbone
!
interface ATM2/0/0.1 point-to-point
 description Enlace TELMEX-MEXICO (POP)
 ip address 200.23.60.230 255.255.255.252
 no ip directed-broadcast
 ip pim sparse-mode
 ip router isis backbone
 ipv6 address 2001:448:3:62::2/64
 ipv6 router isis backbone
!
router isis backbone
 net 49.1999.2000.2306.0065.00
 is-type level-2-only
 metric-style wide level-2
```

Configuración del router Avantel–México

```
!
interface ATM1/0.1 point-to-point
 description PVC TELMEX-MEXICO (GIGAPOP)
 mtu 9180
 ip address 200.23.60.226 255.255.255.252
 ip pim sparse-mode
 atm pvc 3 10 210 aal5snap inarp
 ip router isis backbone
 ipv6 address  2001:448:3:65::1/64
 ipv6 router isis backbone
!
interface ATM1/0.2 point-to-point
 description PVC CON TELMEX-MONTERREY (GIGAPOP)
 mtu 9180
 ip address 200.23.60.234 255.255.255.252
 ip pim sparse-mode
 atm pvc 5 10 211 aal5snap inarp
 ip router isis backbone
 ipv6 address 2001:448:3:64::1/64
 ipv6 router isis backbone
!
interface ATM1/0.3 point-to-point
 description PVC CON TELMEX-TIJUANA (GIGAPOP)
 mtu 9180
 ip address 200.23.60.242 255.255.255.252
 ip pim sparse-mode
 atm pvc 6 10 212 aal5snap inarp
 ip router isis backbone
 ipv6 address 2001:448:3:63::1/64
 ipv6 router isis backbone
!
router isis backbone
net  49.1999. 2000.2306.0201.00
is-type level-2-only
metric-type wide level-2
```

Configuración del router Guadalajara

```
!
interface ATM1/0.1 point-to-point
 description PVC A GDL (GIGAPOP)
 ip address 200.23.60.241 255.255.255.252
 ip router isis backbone
 ipv6 address 2001:448:3:63::2/64
 ipv6 router isis backbone
!
ip router isis backbone
net 49.1999.2000.2306.0105.00
is-type level-2-only
metric-style wide level-2
```

Configuración del router Tijuana

```
!
interface ATM2/0.1 point-to-point
 description PVC A MTY (GIGAPOP)
 ip address 200.23.60.146 255.255.255.252
 ip pim sparse-dense-mode
 encapsulation aal5snap
 ip router isis backbone
 ipv6 address 2001:448:3:67::2/64
 ipv6 router isis backbone
 !
 router isis backbone
 net 49.1999.2000.2306.0161.00
 is-type level-2-only
 metric-type wide level-2
```

Configuración del router Juárez

```
!
interface POS2/0
 ip address 200.23.60.94 255.255.255.252
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 pos framing sdh
 ip router isis backbone
 ipv6 address 2001:448:3:68::2/64
 ipv6 router isis backbone
!
router isis backbone
net 49.1999.2000.2306.0094.00
is-type level-2-only
metric-type wide level-2
```

Configuración del router Cancún

```
!
interface ATM1/0.1 point-to-point
 description PVC A TELMEX-GDL (POP)
 mtu 9180
 ip address 200.23.60.233 255.255.255.252
 ip pim sparse-mode
 atm pvc 3 10 211 aal5snap inarp
 ip router isis backbone
 ipv6 address 2001:448:3:64::2/64
 ipv6 router isis backbone
!
interface ATM1/0.5 point-to-point
 description PVC A Juarez (GIGAPOP)
 ip address 200.23.60.145 255.255.255.252
 ip access-group NO_SNMP_Juarez out
 ip pim sparse-dense-mode
 atm pvc 7 7 207 aal5snap inarp
 ip router isis backbone
 ipv6 address 2001:448:3:67::1/64
 ipv6 router isis backbone
!
interface ATM1/0.7 point-to-point
 description PVC AVANTEL-MTY(POP)
 ip address 200.23.60.237 255.255.255.252
 no ip redirects
 ip pim sparse-mode
 atm pvc 9 7 209 aal5snap inarp
 ip router isis backbone
 ipv6 address 2001:448:3:69::1/64
 ipv6 router isis backbone
!
 router isis backbone
 net 49.1999. 2000.2306.0129.00
 is-type level-2-only
 metric-type wide level-2
```

Configuración del router Telmex–Monterrey

Glosario

A

Abilene – Red educativa y de investigación de los EE.UU.

AFRINIC –dependencia encargada de la asignación de direcciones IP, direcciones IPv6 y Sistemas Autónomos en el continente Africano.

APNIC – Dependencia encargada de la asignación de direcciones IP, direcciones IPv6 y Sistemas Autónomos en el continente Asiático

ARIN – Dependencia encargada de la asignación de direcciones IP, direcciones IPv6 y Sistemas Autónomos en el continente Americano.

ARPANET – Advanced Research Project Agency Network

AVANTEL – Proveedor de servicios de Internet en México.

B

Backbone – Arquitectura de la red principal.

Bandwidth – Capacidad de transmisión de un enlace.

Best path –Mejor ruta conocida hacia un destino o red.

BGP – Borde Gateway Protocol.

C

Canarie – Red educativa y de investigación de Canadá.

CENIC – Corporation of Education Network Initiatives in California.

CIDR – Clasless Interdomain Routing

Clara – Red educativa y de investigación de Sudamérica.

Clasful – Nomenclatura usada para denominar a las redes que se dividen de acuerdo a la jerarquía establecida.

Clasless – Nomenclatura usada para denominar las redes que pueden dividirse libremente, sin ninguna jerarquia.

CSNET – Computer Science network

CUDI – Corporación Universitaria para el desarrollo de Internet..

D

Datagrama – Nombre asignado a los datos en la capa 3 del modelo OSI, también es conocido como paquete.

Default Gateway– Router predeterminado.

Delay – Retardo en el envio de paquetes.

DoD – Deparment of Defense

E

EGP – Exterior Gateway Protocol.

G

Gateway–Dispositivo que realiza funciones de enrutamiento de paquetes.

Gbps – Giga bits por segundo
Geant– Red educativa y de investigación de Europa

H

H-323 – Tecnología para la transmisión de voz mediante redes de datos.
HDTV – Televisión de alta definición.
Host – Computadora personal o *router*

I

IETF – Internet Engineering Task Force.
IGP – Interior Gateway Protocol.
IGRP – Inter Gateway Routing Protocol.
internet – Conexión de redes de datos ubicadas en distintos lugares geográficos.
Internet – Red mas grande del mundo, interconecta redes de todo el mndo.
Internetting – Interconexion de varias redes de datos
IP – Internet Protocol, protocolo que permite la conexión entre computadoras.
IPng – Internet Protocol next generation.
IPv6 Internet Protocol version 6
IS–IS –Intermediate System to Intermediate System Protocol.

K

Kbps – Kilo bits por segundo

L

LACNIC – Dependencia encargada de la asignación de direcciones *IP*, direcciones *IPv6* y Sistemas Autónomos en Latinoamérica.
LAN – Red de datos de área local.

M

Mbps – Mega bits por segundo
Métrica –Unidad de medida para los protocolos de enrutamiento.
MILNET – Military Network
MPLS – Multiprotocol Label Switching.
Multicast – Tecnología para propagar datos de un punto a muchos.

N

Next Hop – Siguiente router al cual se reenvían los paquetes.
NGI – Next Generation Internet
NSF –National Science Fundation
NSFNET –National Science Fundation Network

O

OSPF – Open Shortest Path First Protocol.

P

PDU – Protocol Data Unit.
PoP – Punto de presencia de un equipo de acceso a la red.
Pseudonode – También llamado *Pseudonodo*, es un nodo virtual que se establece en las redes unidas mediante tecnología *Ethernet*.

Q

QoS – Quality of Service, calidad de servicio

R

RFC – Request For Comments
RIP – Routing Information Protocol.
RIPE – Dependencia encargada de la asignación de direcciones IP, direcciones IPv6 y Sistemas Autónomos en el continente Europeo.
Route table –Tabla de enrutamiento, base de datos que contiene las rutas hacia las redes.
Routers – Dispositivo que realiza funciones de enrutamiento de paquetes.

T

T1 – Unidad de medida del sistema Americano (1.544 Mbps)
T3 – Unidad de medida del sistema Americano (1.544 Mbps)
TCP – Transmission Control Protocol, protocolo de control de la transmisión de datos.
TELMEX – Proveedor de servicios de Internet en México.
TTL –Time to Live

U

UCAID – Universitary Corporation for Advanced Internet Development

V

VLSM – Variable Length Subnetting Mask

W

WAN – Red de datos de área amplia.