UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES

CAMPUS ARAGON

"MANEJO DE INFORMACIÓN, DIRECCIONAMIENTO Y SEGURIDAD CON EL PROTOCOLO IPv6"

T E S I S

QUE PARA OBTENER EL TITULO DE:

INGENIERO MECANICO ELECTRICISTA

P R E S E N T A N:

PEDRO FERNANDO CAMORLINGA POSCH

RAFAEL QUIROZ PLATA

ASESOR DE TESIS:
ING. ELEAZAR PINEDA DIAZ

SAN JUAN DE ARAGON, EDO. DE MEXICO 2000

A mis padres, que me han dado un apoyo incondicional a todo lo largo de la carrera y de mi vida, que siempre han estado a mi lado para darme la fuerza, la entereza y el amor necesarios para salir adelante.

A mi hermana, que con su compresión, cariño y sinceridad me ha hecho ver siempre el lado humano de las cosas y luchar por ser cada vez mejor.

Pedro

A mis padres que siempre y en todo momento de mi vida me han sabido brindar su cariño. apoyo y comprensión, para salir adelante. y por que gracias ha ellos he logrado ser la persona y el profesionista que soy ahora

A mis hermanos David. Israel y Noé, que me han brindado su apoyo y amistad con quien he compartido grandes momentos de mi vida

Rafael

Indice

CAPÍTULO II

EL PROTOCOLO IPv4

Pag.

CAPÍTULO III

TRANSICIÓN ENTRE IPV4 E IPV6

Pag.

CAPÍTULO IV

MANEJO DE LA INFORMACIÓN EN IPV6

Pag.

CAPÍTULO V

DIRECCIONAMIENTO Y SEGURIDAD EN IPV6

Pag.

INTRODUCCION

La problemática o la necesidad de intercambiar información lo más rápida y confiablemente posible, ha propiciado el desarrollo de sistemas de comunicaciones para compartir recursos y datos La demanda de sistemas de comunicaciones basados en la transmisión de información a través de la red mundial Internet, ha tenido un incremento sin precedentes Cada vez se necesita una mayor velocidad y eficiencia en la transmisión de datos y en el intercambio de información entre dos puntos cualesquiera del mundo.

Por otro lado, debido a que la transmisión de información vía Internet está inmersa en un entorno tecnológico de rápida evolución, con tendencia a modificaciones inmediatas en los modos de operación, y demanda cada vez mayor de nuevos servicios, el protocolo y los sistemas que se diseñaron en un principio para cubrir las necesidades del momento, se han convertido con el paso del tiempo en insuficientes y obsoletos Es decir, los sistemas de hoy en día se han quedado al margen del desarrollo tecnológico. En estos momentos el principal problema a que se enfrentan estos sistemas y protocolos, es la insuficiencia de direcciones IP ya que se prevé que en un tiempo no muy largo se agoten por completo

Por lo anterior, se requiere implementar un nuevo protocolo de manera que no limite el crecimiento de la red y de los servicios, sino al contrario, que pueda incrementar su capacidad sin necesidad de efectuar un rediseño de toda la estructura, y debe ser diseñado bajo la premisa de que la red continuará expandiéndose como hasta ahora lo ha venido haciendo o incluso en una mayor proporción, y que, por lo tanto, debe ser capaz de cubrir la demanda de servicios a futuro.

Este trabajo se enfoca al análisis de las características del nuevo protocolo de Internet IPv6, que se ha planteado como una solución al problema cada vez mayor de escasez de direcciones para Internet, debido al constante crecimiento de esta red mundial, basándose en las necesidades que surgen de la incapacidad del protocolo actual IPv4, cada vez más obsoleto, para satisfacer la demanda de los usuarios en materia de comunicaciones, y sin pasar por alto los problemas inherentes a dicha transición.

Para el desarrollo de este trabajo en primer lugar debemos conocer el modelo sobre el que trabaja IPv4 e IPv6, para tener las bases necesarias y poder estudiar las posibles ventajas y desventajas que conlleva la implementación del nuevo protocolo IPv6 Después debemos analizar las mejoras y las novedades de IPv6 y, una cosa que no se puede pasar por alto, considerar el impacto que traerá la puesta en operación de este nuevo protocolo y los problemas que surgirán durante la transición entre los protocolos actual y nuevo

Por lo tanto, el objetivo general de este trabajo es hacer un análisis comparativo entre el protocolo actual de Internet, IPv4 y el protocolo de siguiente generación IPv6, y determinar en base a las características funcionales del nuevo protocolo las razones para cambiar IPv4

Este trabajo contiene los siguientes temas. a) en el tema I se da a conocer el modelo TCP/IP, su historia, su arquitectura y el modo de operación de sus capas, junto con los diferentes protocolos que cada una de ellas emplea. Para ello se tomará como referencia el modelo de comunicaciones OSI.

b) En el segundo tema se hará un repaso de las características del protocolo actual IPv4, las clases y tipos de direcciones que emplea, la forma de manejar la información y el formato que emplea para los datagramas, así como la manera en que los fragmenta y los reensambla, junto con el ruteo de la información y, algo que no se puede pasar por alto, la seguridad de ésta.

c) En el tema III se considerará la transición entre el protocolo actual IPv4 y el nuevo IPv6, viendo los diferentes tipos de componentes existentes durante esta transición, así como los mecanismos que ayudarán a resolverla en gran medida También se considerarán los aspectos negativos de IPv6, junto con los problemas derivados del cambio de protocolos

d) En el cuarto tema se estudiará el manejo de la información con el nuevo protocolo, el formato que tienen sus cabeceras y los tipos que existen de las llamadas cabeceras suplementarias El papel del control de flujo es también un aspecto muy importante que se tratará junto con el ruteo en IPv6.

e) Por último, se analizará el direccionamiento y la seguridad en el protocolo de nueva generacion. El direccionamiento se basa en las diferentes representaciones de las direcciones y en los diversos tipos que de ellas existen Este protocolo implementa la caracteristica de autoconfiguración de direcciones, lo cual permite un manejo más seguro de la información. Existen asimismo varios tipos de mecanismos de seguridad, donde juegan un papel muy importante algunos tipos de cabeceras y, una cosa muy importante, se pueden combinar los mecanismos de seguridad para obtener una mayor confiabilidad en el manejo de información.

Capítulo I

CARACTERISTICAS GENERALES DEL MODELO TCP/IP

1.1 HISTORIA DE TCP/IP

Las redes de computadoras han incrementado en gran medida su habilidad para comunicarse entre ellas, dado que la mayoría de las computadoras se emplean más para la comunicación de datos que para la computación. Muchas estaciones de trabajo y supercomputadoras están ocupadas realizando cálculos complejos para la ciencia y para los negocios, pero este número es mínimo en comparación con los millones de sistemas ocupados en el intercambio de información entre ellos. Además, si se toman en cuenta las computadoras portátiles que se emplean también para pasar información de una persona a otra, se puede comprender mejor por qué la mayoría de las computadoras son vistas como sistemas de comunicación.

El nombre "TCP/IP" se refiere a todo un conjunto de protocolos de comunicaciones de datos. Este conjunto toma su nombre de dos de los protocolos que pertenecen a él: el Protocolo de Control de Transmisión (Transmission Control Protocol) y el Protocolo Internet (Internet Protocol). Aunque existen muchos otros protocolos en el conjunto, los *protocolos TCP e IP son, ciertamente, los dos más importantes.*

El modelo TCP/IP (Protocolo de Control de Transmisión / Protocolo Internet) es un modelo de comunicación que permite la operación entre equipos de cualquier tipo de hardware con una gran gama de tecnología de redes

En 1969, la Agencia de Proyectos de Investigación Avanzada, ARPA (Advanced Research Project Agency) fundó un proyecto de investigación y desarrolló una red experimental de conmutación de paquetes. Esta red, llamada ARPANET, fue construida para estudiar las técnicas de confiabilidad, robustez e independencia en las comunicaciones de datos.

La red ARPANET tuvo tanto éxito, que muchas de las organizaciones enlazadas a ella empezaron a usarla para las comunicaciones de datos comunes. En 1975, ARPANET se convirtió de una red experimental a una red operacional, y la responsabilidad de su administración y manejo se le concedió al Departamento de la Defensa de los Estados Unidos, DARPA (Defense Advanced Research Project Agency). Los protocolos básicos TCP/IP fueron desarrollados después de que la red se convirtiera en operacional.

Los protocolos TCP/IP fueron adoptados como estándares militares (MIL STD) en 1983, posteriormente fueron adaptados para operar con redes locales. Todos los *servidores conectados a la red requirieron ser convertidos a los nuevos protocolos. Para* facilitar esta conversión, la agencia DARPA fundó Bolt, Beranek y Newman (BBN) para implementar TCP/IP como parte del sistema UNIX de Berkeley, y desde entonces comenzó su popularización.

Al mismo tiempo que el modelo TCP/IP fue adoptado como estándar, el término "Internet" se convirtió en un concepto de uso común. En 1983, la antigua red ARPANET fue dividida en una red militar (llamada Milnet), la parte no clasificada de la Red de Datos

de la Defensa, DDN (Defense Data Network), y en una nueva y más pequeña red ARPANET "Internet" fue usado para referirse a la red completa: Milnet junto con ARPANET

En 1985, la Fundación Nacional de Ciencia, NSF (National Science Foundation) creó la red NSFNet, conectándola a la red existente Internet. La red original NSFNet enlazó los cinco centros de supercómputo de la NSF. Fue más pequeña que la red ARPANET y no tan rápida (56 kbps). De cualquier manera, la creación de la red NSFNet fue un suceso muy significativo en la historia de Internet, porque la NSF proporcionó con ella una nueva visión del uso de Internet. La NSF quería extender la red de tal manera que todos los científicos e ingenieros de los Estados Unidos tuvieran acceso a ella. Para llevar a cabo esto, en 1987, la NSF creó un nuevo y más rápido backbone (estructura principal de la red) y una topología de red de tres capas que incluía el backbone, las redes regionales y las redes locales.

En 1990, la red ARPANET dejó formalmente de existir, y la NSFNet dejó su papel como backbone primario de Internet en 1995. Aún en nuestros días, la red Internet es la más grande que existe y abarca más de 100 mil redes en todo el mundo. Es una red tipo paranoico, donde sus protocolos tienen previsto que cualquier anomalía puede ocurrir. Esta red sigue el modelo *catenet*, en donde cada paquete de información (también conocido como datagrama) es una comunicación completa, y por lo tanto, dos paquetes pueden seguir trayectorias completamente diferentes.

Internet se ha desarrollado mucho más allá de su propósito original. Las agencias y redes originales que construyeron Internet la diseñaron para que desempeñara el simple papel de una red normal Internet fue convertida de un simple backbone, con una estructura jerárquica de tres capas, a una enorme red de nodos distribuidos e interconectados. Se ha desarrollado de manera exponencial desde 1983, duplicando su tamaño cada año A pesar de todos estos cambios, una característica se ha mantenido constante Internet está basada en el conjunto de protocolos TCP/IP

Actualmente TCP/IP es requerido para realizar una conexión en Internet, por lo que el crecimiento de ésta ha tenido un interés particular en dicho conjunto de protocolos. Con el auge de Internet nos encontramos con que la mayoría de las empresas en las que hasta hace un tiempo se utilizaban protocolos de red tradicionales con sistemas operativos como Novell NetWare, Lan Manager, Banyan Vines y otras redes pares (peer to peer) como Invisible, Power, Windows para Trabajo en Grupo, etc., se han familiarizado con TCP/IP, y se han dado cuenta de que su poder puede ser aplicado en otras aplicaciones diferentes de redes Los protocolos de Internet son usados muy frecuentemente para redes de área local, aun cuando la red local no esté conectada a Internet. TCP/IP tienen también un uso muy extendido en la implementación de redes dentro de las empresas. Las redes empresariales basadas en TCP/IP, que emplean técnicas de Internet y herramientas del World Wide Web para la distribución e intercambio de información interna, son llamadas *intranets*.

1.2 ARQUITECTURA DE TCP/IP

La popularidad de los protocolos TCP/IP se extendió rápidamente debido a que su uso era requerido para las conexiones de Internet. Existía, asimismo, la gran necesidad de

comunicación de datos a todo lo ancho del mundo, y por lo tanto, estos protocolos tenían unas características muy importantes que les permitían subsanar esta necesidad. Estas características son las siguientes:

- *Estándares de protocolos abiertos*, con disponibilidad libre de coste y con un desarrollo independiente de cualquier hardware específico o sistema operativo. Debido a que tiene un soporte muy amplio, TCP/IP es ideal para unificar hardware y software diferentes, aun cuando no exista una comunicación directa sobre Internet
- Independencia de cualquier hardware específico de red física. Esto permite a TCP/IP integrar muchas clases diferentes de redes TCP/IP puede operar sobre Ethernet, token ring, una línea dial-up, una red FDDI, y virtualmente sobre cualquier otro medio físico de transmisión.
- Un esquema de direccionamiento común que permite a cualquier dispositivo TCP/IP direccionar de manera única cualquier otro dispositivo en toda la red, aun si la red es tan grande como Internet.
- Protocolos estandarizados de alto nivel para servicios de usuarios robustos y de amplia disponibilidad

Estándares de Protocolos

Los protocolos son reglas formales de comportamiento. En relaciones internacionales, los protocolos minimizan los problemas causados por diferencias culturales cuando varias naciones trabajan conjuntamente. A través del convenio de un conjunto común de reglas que son internacionalmente conocidas e independientes de las costumbres de cualquier país, los protocolos minimizan los malos entendidos; cualquiera sabe cómo actuar y cómo interpretar las acciones de los demás. De forma similar, cuando las computadoras se comunican, es necesario definir un conjunto de reglas para dirigir la comunicación entre ellas

En las comunicaciones de datos, estos conjuntos de reglas también son llamados *protocolos* En redes homogéneas, un simple proveedor de computadoras especifica el conjunto de reglas de comunicaciones diseñadas para aprovechar toda la eficacia de los sistemas operativos y la arquitectura del hardware de los demás proveedores. TCP/IP intenta crear una red heterogénea con protocolos abiertos, que son independientes de las diferencias entre los sistemas operativos y las arquitecturas. Los protocolos TCP/IP están disponibles para cualquiera, y son desarrollados y modificados por consenso. Cualquier persona es libre de desarrollar productos que sean compatibles con las especificaciones de estos protocolos abiertos.

La naturaleza abierta de los protocolos TCP/IP requiere de documentos sobre estándares disponibles públicamente. Todos los protocolos del conjunto TCP/IP están definidos en una de tres publicaciones de estándares sobre Internet. Un número de protocolos han sido adoptados como Estándares Militares, *Military Standards* (MIL STD). Otros fueron publicados como Notas de Ingeniería de Internet, *Internet Engineering Notes* (IEN), aunque la forma de publicación IEN ahora ha sido abandonada. Las recomendaciones para nuevos protocolos o protocolos mejorados se emiten a través de documentos denominados Requerimientos para Comentarios, *Requests for Comments (RFCs)*.

Los RFCs se asignan a un grupo de trabajo de la Fuerza de Trabajo de Ingeniería de Internet, *Internet Engineering Task Force* (IETF). Los RFCs contienen las últimas versiones de las especificaciones de todos los protocolos estándares TCP/IP. Tal como el

título "Requerimientos para Comentarios" implica, el estilo y el contenido de estos documentos es mucho menos rígido que la mayoría de los documentos sobre estándares. Los RFCs contienen mucha información interesante y útil, y no están limitados a las especificaciones formales de los protocolos de comunicación de datos. Cuando se ha demostrado que los protocolos propuestos en estos documentos funcionan correctamente y son aceptados por los usuarios, se convierten de una manera natural en estándares *de facto*. De esta manera aparecieron y siguen apareciendo las numerosas aplicaciones que *se utilizan en el ambiente de Internet.*

1.2.1 Descripción General de la Familia de Protocolos TCP/IP

TCP/IP es una familia de protocolos desarrollados para permitir la comunicación entre cualquier par de computadoras de cualquier red o fabricante, respetando los protocolos de cada red individual.

La figura 1.1 muestra el modelo general de los protocolos TCP/IP con algunas de las aplicaciones normalizadas que los emplean, donde

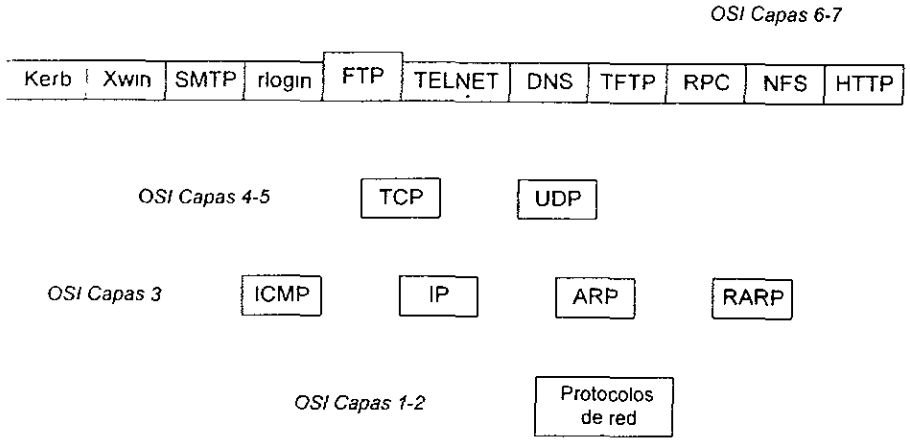


Figura 1.1 - *Protocolos y Aplicaciones TCP/IP*

Protocolos

Como ya lo explicamos anteriormente, los protocolos establecen una descripción formal de los formatos que deberán presentar los mensajes para poder ser intercambiados por equipos de cómputo; además, definen las reglas que ellos deben seguir para lograrlo.

Los protocolos están presentes en todas las etapas necesarias para establecer una comunicación entre equipos de cómputo, desde aquellas de más bajo nivel (por ejemplo,

la transmisión de flujos de bits a un medio físico) hasta aquéllas de más alto nivel (por ejemplo, el compartir o transferir información desde una computadora a otra en la red).

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

Conjunto de Protocolos TCP/IP

Los protocolos TCP/IP proporcionan a los usuarios unos servicios de comunicación, tales como:

- Transferencia de archivos
- Login remoto o Terminal Virtual
- Correo Electrónico
- Acceso a Archivos Distribuidos
- Administración de Sistemas
- Manejo de Ventanas

La familia de protocolos TCP/IP se puede estructurar en cuatro capas funcionales: Aplicación, Transporte, Internet y de Acceso al Medio, las cuales se describen de forma general a continuación

1.2.2 Descripción del Modelo de Capas de TCP/IP

Como vimos, en la actualidad, las funciones propias de una red de computadoras pueden ser divididas en las siete capas propuestas por ISO para su modelo de sistemas abiertos (OSI). Sin embargo, la implantación real de una arquitectura puede diferir de este modelo. Las arquitecturas basadas en TCP/IP proponen cuatro capas en las que las funciones de las capas de Presentación y Aplicación son responsabilidad de la capa de Aplicación, las capas de Transporte y Sesión se refieren a la capa de Transporte, la de Internet se relaciona con la de Red y las capas de Enlace de Datos y Física son vistas como la capa de Acceso al Medio o Interfaz de Red. Por tal motivo, para TCP/IP sólo existen las capas *Interfaz de Red*, *la de Intercomunicación en Red* (Internet), *la de* Transporte y la de Aplicación.

Capa de Aplicación.

La capa de aplicación corresponde a los niveles de presentación y de aplicación del Modelo OSI. Algunos autores la relacionan con los tres niveles superiores del Modelo OSI sesión, presentación y aplicación.

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

En esta capa se encuentran las aplicaciones disponibles para los usuarios. Una aplicación es un proceso de usuario que está cooperando con otro proceso de usuario en una misma máquina o en máquinas diferentes. Algunos ejemplos de tales aplicaciones son el Protocolo de Transferencia de Archivo (FTP), el Protocolo Simple de Transferencia de Correo (SMTP) y Conexión Remota (TELNET), entre otros.

Capa de Transporte.

Esta capa corresponde a los niveles de transporte y de sesión del Modelo OSI. Algunos autores la relacionan únicamente con el nivel de transporte de este modelo. Provee comunicación extremo a extremo desde un programa de aplicación a otro. Regula el flujo de información. Puede proveer un transporte confiable asegurándose de que los datos lleguen sin errores y en la secuencia correcta. Coordina a múltiples aplicaciones que se encuentren interactuando con la red simultáneamente, de tal manera que los datos que envíe una aplicación sean recibidos correctamente por la aplicación remota; esto lo hace añadiendo identificadores de cada una de las aplicaciones. Realiza además una verificación por suma, para asegurar que la información no sufrió alteraciones durante su transmisión.

La capa de transporte suministra a las aplicaciones servicios de comunicaciones de extremo a extremo utilizando dos tipos de protocolos: el Protocolo de Control de Transferencia (TCP), fiable y orientado a conexión, y el Protocolo de Datagrama de Usuario (UDP), no fiable y no orientado a conexión.

Capa Internet.

Esta capa corresponde al nivel de red del Modelo OSI. Controla la comunicación entre un equipo y otro, decide qué rutas deben seguir los paquetes de información para alcanzar su destino. Conforma los paquetes IP que serán enviados por la capa inferior. *Desencapsula los paquetes recibidos, pasando a la capa superior la información dirigida a* una aplicación.

La capa de Internet se superpone a la red física, creando un servicio de red virtual independiente de aquélla. No es fiable ni orientada a conexión Se esfuerza en entregar los paquetes, denominados datagramas, a su destino. Los datagramas pueden perderse, duplicarse o alterar su orden de secuencia.

Capa de Acceso al Medio.

Esta capa también es conocida como Interfaz de Red y corresponde a los niveles físico y de enlace del Modelo OSI. Emite al medio físico los flujos de bit y recibe los que de él provienen Consiste en los manejadores de los dispositivos que se conectan al medio de transmisión

Es la interfaz con la red real, física. Puede o no proporcionar fiabilidad en la distribución de datos, que pueden adoptar diferentes formatos. De hecho, TCP/IP no especifica ningún protocolo en esta capa, lo que manifiesta la flexibilidad de la capa Internet Como ejemplos de esta interfaz, tenemos la norma IEEE 802.2 (para redes de área local), X.25, Frame Relay o inclusive SNA.

Como puede verse TCP/IP presupone independencia del medio físico de comunicación. Sin embargo, existen estándares bien definidos a los niveles de Enlace de Datos y Físico que proveen mecanismos de acceso a los diferentes medios y que en el modelo TCP/IP deben considerarse en la capa de Interfaz de Red; siendo los más usuales el proyecto IEEE.802, Ethernet, Token Ring y FDDI.

La figura 1.2 muestra el modelo de capas de TCP/IP, describiendo la función que realiza cada una de las capas.

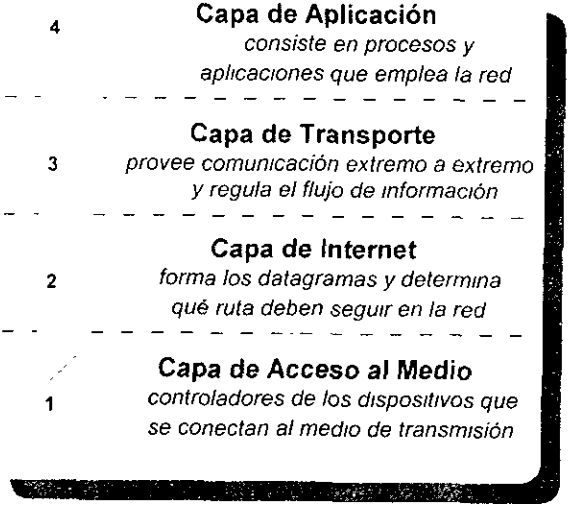


Figura 1.2 - *Modelo de Capas de TCP/IP*

Las aplicaciones se comunican entre sí mediante mensajes. No tiene nivel de presentación. Este nivel debe ser realizado por la aplicación. El nivel de transporte realiza también algunas funciones del nivel de sesión mediante paquetes. El nivel de Internet (IP) maneja los niveles de red mediante una interfaz de red, para acceder al medio. Las unidades de datos intercambiadas en el nivel IP son los datagramas. Finalmente, el nivel de acceso al medio manda los datos a través de la red en forma de tramas. La figura 1.3 ilustra los protocolos de cada capa indicando los datos manejados por cada una de ellas.

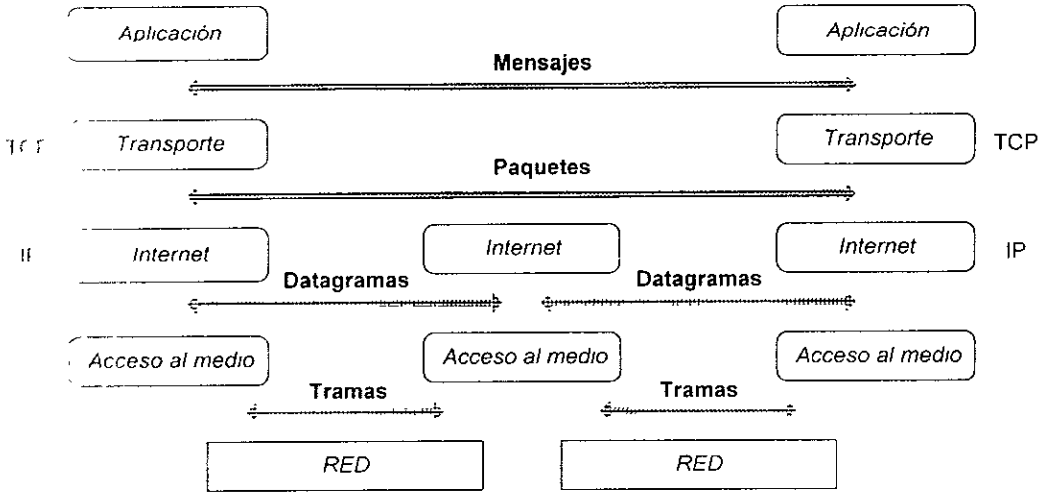


Figura 1.3 - *Protocolos de TCP/IP con sus mensajes*

1.2.3 Comparación de las capas de TCP/IP con las de OSI

Existen diferencias sustanciales entre la concepción del Modelo de Referencia OSI y la del conjunto de protocolos TCP/IP.

Históricamente, los protocolos TCP/IP fueron desarrollados antes que el Modelo de Referencia y, al igual que en muchas otras ocasiones, era muy difícil que los promotores y usuarios esperaran a que los comités de ISO completaran sus actividades. Por otra parte, TCP/IP tenía que contemplar realidades operativas de sistemas en producción, como la seguridad, la interoperabilidad entre redes, la fiabilidad o la gestión de red.

Conceptualmente, también existen diferencias, como son:

- El concepto de jerarquía en relación al de niveles o capas
- La interoperación de redes
- La fiabilidad extremo a extremo
- Los servicios no orientados a conexión
- La gestión de red

El concepto de *jerarquía con relación al concepto de capas* es extremadamente sutil y realmente es consecuencia de que en TCP/IP se aplicó posiblemente un mayor grado de pragmatismo que en los trabajos de ISO. En ambas arquitecturas, una tarea de comunicaciones se divide en módulos o entidades que se pueden comunicar con entidades pares en otro sistema. Una entidad dentro de un sistema proporciona servicios a otras entidades y, a su vez, emplea los servicios de otras. Estas entidades deben tener

una relación jerárquica, de tal manera que una entidad sólo pueda utilizar los servicios de las entidades jerárquicamente inferiores.

Para comunicar entidades similares, TCP/IP da libertad para definir y emplear múltiples protocolos con funcionalidades diferentes. Es decir, en el mismo nivel o capa pueden existir múltiples protocolos con distinta funcionalidad, dejando libertad al diseñador para la utilización de uno u otro. Lo único que realmente es común a todos los protocolos de un nivel o capa es que comparten el mismo conjunto de protocolos de la capa inferior, esto es, están en una jerarquía superior. De esta manera, en TCP/IP existen en la capa de transporte, protocolos de naturaleza muy distinta, como el UDP, no fiable ni *orientado a conexión y el TCP, fiable y orientado a conexión. Los diseñadores de los* niveles superiores tiene la opción de emplear el que desee. Se dice que el Modelo OSI es mas prescriptivo que descriptivo, en el sentido de que dicta los protocolos de un nivel determinado que deben realizar unas funciones determinadas.

Estas diferencias no suponen que haya funciones que se puedan realizar con OSI y no con TCP/IP Lo que sucede es que el conjunto de protocolos TCP/IP, al ser modular y jerarquico proporciona a los diseñadores mayor grado de libertad Podria decirse que estas diferencias son consecuencia de cómo se realiza el proceso de generación de normas En ISO, primero se especifican y posteriormente se realizan implementaciones; en Internet, se especifican al tiempo que se desarrollan, con lo que los ciclos se acortan y además tienen un carácter más orientado a la solicitud de requisitos reales. La aceptación *de facto* se consigue difundiendo por Internet tanto las especificaciones, como las implementaciones en los documentos denominados RFC, que ya tratamos en su *momento*

El modelo OSI se diseñó para sistemas conectados a una misma red Por su parte, los protocolos TCP/IP se han concebido desde su origen para *interconectar sistemas no conectados a la misma red.* Las caracteristicas del protocolo IP se derivan de esta caracteristica

Los protocolos TCP/IP proporcionan una *fiabilidad extremo a extremo* El protocolo IP de nivel de red no es fiable, es decir, no garantiza que los paquetes entregados sean correctos y que conserven la secuencia con que han sido enviados En otras palabras, IP supone que las redes son relativamente fiables y, en caso necesario, la fiabilidad debe garantizarse por los protocolos de transporte en los sistemas de usuario (TCP)

Por el contrario, X.25, por ejemplo, define un conjunto de protocolos, tanto a nivel de enlace como a nivel de red, para control de errores a manera de control de flujo, lo que es redundante con *funciones similares proporcionadas por otros niveles y disminuye la* eficiencia y capacidad de la red.

Como resultado de esto, *los servicios de IP son no orientados a conexión.* En cualquier nivel de TCP/IP se contempla la posibilidad de un servicio no orientado a conexión o datagrama. La conectividad extremo a extremo debe proporcionarse en los niveles superiores Es verdad, ciertamente, que OSI también contempla la posibilidad de utilización de datagramas, si bien se suelen considerar como una alternativa a la opción principal

En los primeros documentos de OSI no se contemplaban las *funciones de gestión.* Actualmente no es el caso, y es posible que los protocolos y servicios de gestión definidos por ISO alcancen un cierto nivel de aceptación, aunque no son tan populares como los definidos en TCP/IP.

Como consecuencia de todo lo anterior, la situación es que hay muy pocas aplicaciones definidas y realmente utilizadas dentro del Modelo de Referencia OSI. La más extendida

quizá sea posiblemente el sistema de correo electrónico basado en la recomendación X 400 Sin embargo, hay un amplio conjunto de aplicaciones TCP/IP que son estándares *de facto* y muy populares, como TELNET, FTP, SMTP, NFS, etc.

En la figura 1.4 aparece el conjunto de protocolos TCP/IP y su comparación con el modelo OSI.

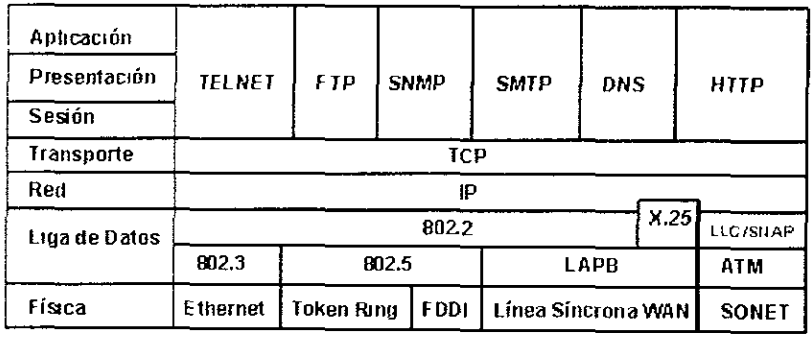


Figura 1.4 - *Protocolos TCP/IP y su relación con el Modelo OSI*

1.3 EL MODELO DE COMUNICACIONES OSI

1.3.1 Introducción

El Modelo de Referencia OSI tuvo su inicio a mediados de los años setenta cuando, debido al interés sobre el diseño de bases de datos distribuidas, surgió la necesidad de una arquitectura de comunicaciones estructurada y distribuida El grupo *Canepa* analizó algunas de las soluciones existentes en aquel momento, entre ellas la Arquitectura de Red de Sistemas, SNA (*System Network Architecture*) de IBM y los trabajos sobre protocolos de ARPANET. El resultado de este trabajo dio lugar en 1977 a una arquitectura de siete niveles, conocida internamente como Arquitectura de Sistemas Distribuidos, DSA (*Distributed Systems Architecture*). Al mismo tiempo, el Instituto Británico de Estandarización propuso a ISO la necesidad de un estándar de arquitectura que soportara la definición de la infraestructura de comunicaciones de los procesos distribuidos. ISO constituyó un comité sobre interconexión de sistemas abiertos (Comité técnico 97, Subcomité 16), siendo ANSI el encargado del desarrollo de las propuestas.

En 1978 se llegó a un consenso entre las propuestas de ANSI, en colaboración con el grupo Canepa, y las que presentó Honeywell. El resultado del acuerdo fue la elección de una arquitectura por niveles y la consideración de que ésta satisfaría la mayoría de los

requerimientos exigibles a la interconexión de sistemas abiertos, así como el reconocimiento de la capacidad de expansión del modelo para adecuarse a los requerimientos futuros. Ese mismo año se publicó la versión provisional del modelo. En junio de 1979 la siguiente versión, con muy pocos cambios, pasó a convertirse en estandar El modelo OSI definitivo es básicamente el mismo modelo DSA desarrollada en 1977.

En el ambito de las redes locales, las primeras tareas de normalización han corrido a cargo del Instituto de Ingenieros Eléctricos y Electrónicos, IEEE (*Institute of Electrical and Electronics Engineers*), cuyo proyecto 802 ofrece normas orientadas a guiar la fabricación de componentes y software para las redes locales, algunas de las cuales han sido adoptadas como estándares *de facto* en el mundo de la industria.

El proyecto IEEE 802 fue adoptado en 1985 por el Instituto Nacional de Estándares Norteamericano, ANSI (*American National Standards Institute*)[1], como conjunto de estándares americanos y, posteriormente, tras su revisión y reedición por la Organización Internacional para la Estandarización, ISO (*International Organization for Standardization*)[2] en 1987, pasó a convertirse en estándar internacional con la denominación de ISO 8802.

Por otra parte, ISO ha proporcionado lo que se denomina Modelo de Referencia para la Interconexión de Sistemas Abiertos, OSI (*Open Systems Interconnection*), el cual veremos a continuación

Este modelo, conocido por las siglas OSI, se creó a principios de 1977 y obtuvo el grado definitivo de estándar internacional en 1983, trata de establecer las bases para la definición de protocolos de comunicación entre sistemas informáticos. Su principal objetivo es la interconexión de sistemas de diferentes fabricantes, es decir, de sistemas abiertos Por ello, OSI constituye un marco para la coordinación de las actividades de normalización en los sistemas de telecomunicación e información

Cada sistema abierto está lógicamente formado por un conjunto ordenado de subsistemas (en concreto se han definido siete niveles) que, junto con el medio físico proporcionan un conjunto completo de servicios de comunicación. La funcionalidad de cada nivel está definida por los servicios OSI. La comunicación entre los niveles de sistemas diferentes se realiza mediante la definición de un protocolo, siendo éste independiente de los protocolos de los demás niveles

El enfoque modular es análogo al del diseño de sistemas software para ordenadores, particularmente en el área de los sistemas operativos. Los problemas de diseño e implementación se hacían más manejables mediante la división de los mismos en tareas mas pequeñas y concretas Asimismo, el mantenimiento y modificación es más eficiente,

---

[1] ANSI es la entidad normalizadora de los Estados Unidos. Fue fundada en 1918 y es una organización no gubernamental constituida por más de un millar de organizaciones comerciales, sociedades profesionales y corporaciones. ANSI por sí misma no crea estandares, sino que se dedica a coordinar y sincronizar las actividades de otras organizaciones que si desarrollan estándares, y a asegurar que todos los intereses afectados tienen una oportunidad de participar en el proceso.

[2] ISO es una organización no gubernamental, fundada en 1947. Su misión es coordinar el desarrollo y aprobación de los estándares como estándares internacionales. Es decir, promueve el desarrollo de la estandarización y actividades relacionadas en todo el mundo con el objetivo de facilitar el intercambio de bienes y servicios y la cooperación en los ámbitos intelectuales, científicos, tecnológicos y económicos Su ámbito de trabajo cubre todas las áreas, incluyendo la normalización de las redes de área local, a excepción de las áreas electrotécnicas, que son coordinadas por la Comisión Electrotécnica Internacional.

puesto que cada nivel, en teoria, puede ser reemplazado sin alteración de los niveles adyacentes.

El modelo, sin embargo, no está libre de críticas por parte de los usuarios, debido a que introduce una complicación innecesaria en los sistemas de comunicación más sencillos. El escepticismo radica en que el software necesario para realizar las funciones de los niveles puede llegar a ser considerable, consumiendo muchos recursos de manera excesiva, como la memoria, y haciendo más lentos los procesos, así como en la inclusión de redundancia en su realización

El Modelo de Referencia pretende ser suficientemente flexible como para que, a medida que se incrementen los avances tecnológicos y las demandas de los usuarios, *pueda ajustarse a tales exigencias.*

En general, el propósito del Modelo de Referencia es identificar áreas de desarrollo o mejora de estándares, y proporcionar una referencia común para el mantenimiento de la consistencia entre los mismos No es objetivo del Modelo de Referencia servir de especificación para la implementación, ni como base para la valoración de las implementaciones actuales, ni siquiera el de suministrar un nivel de detalle suficiente como para definir con precisión los servicios y protocolos de la arquitectura de interconexión En su lugar, el Modelo de Referencia proporciona un marco conceptual y funcional que permite a los equipos internacionales trabajar de manera productiva e independiente en el desarrollo de estándares para cada nivel del Modelo de Referencia OSI

1.3.2 El Modelo de Referencia

El modelo de referencia OSI es el modelo que se ha estructurado más recientemente, por lo que, a pesar de no existir muchas implementaciones OSI, se puede afirmar que se trata del modelo que proporciona un nivel más formal Por este motivo, es el que se emplea como referencia para desarrollar los conceptos de redes y sistemas teleinformáticos.

Dentro del modelo de referencia OSI se establecen tres niveles, que son

- La arquitectura OSI: define los elementos básicos de los sistemas abiertos, es decir, de qué manera debe verse un sistema desde el exterior
- Las especificaciones de servicio OSI. definen los servicios proporcionados a los usuarios en cada nivel; es decir, los servicios proporcionados por un nivel al *nivel superior.*
- Las especificaciones de protocolos OSI: definen la información de control transmitida entre los diferentes sistemas, así como los procedimientos para la interpretación de dicha información de control.

El modelo de referencia OSI es un modelo de redes estructuradas en capas o niveles. El objetivo de esto, es tratar de manera estructurada la totalidad de un sistema teleinformático. El conjunto de funciones del sistema se divide en niveles, facilitando su estudio y desarrollo, de manera que sean fácilmente controlables de manera individual y que en conjunto resuelvan satisfactoriamente las necesidades de comunicación.

Cada nivel se desarrolla sobre el anterior, de tal forma que recibe una serie de servicios sin conocer los detalles de cómo se realizan dichos servicios.

Las diferentes funciones de la arquitectura OSI han sido estructuradas en siete niveles o capas, siendo las funciones asignadas a cada una de ellas complementarias. Las características generales de las capas son las siguientes:

- *Cada una de las capas desempeña funciones bien definidas*
- Los servicios proporcionados por cada nivel son utilizados por el nivel superior.
- Existe una comunicación virtual entre 2 mismas capas, de manera horizontal.
- Existe una comunicación vertical entre una capa de nivel N y la capa de nivel N + 1.
- La comunicación física se lleva a cabo entre las capas de nivel 1

En la figura 1.5 se representa la arquitectura de una red basada en el Modelo OSI, donde se observa que.

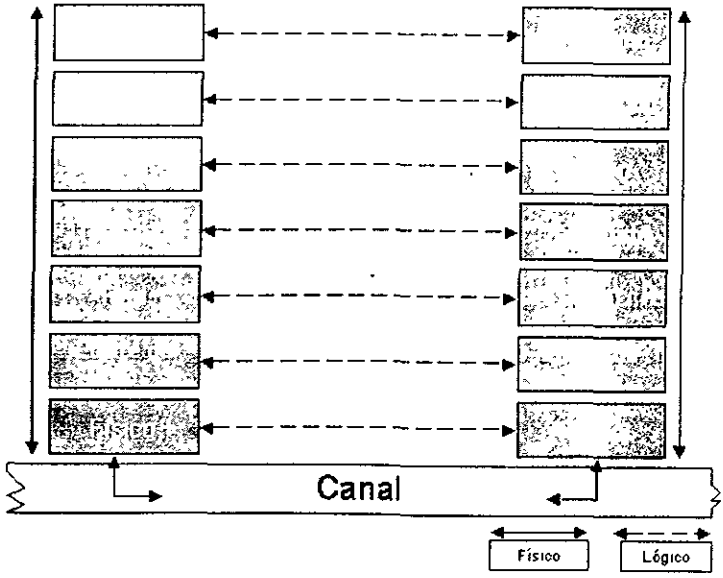


Figura 1.5 - *El Modelo OSI*

La estructura de una red de comunicaciones se compone de una serie de nodos que pueden estar formados por el sistema central, una unidad de control de comunicaciones o una terminal. En ella se define el término usuario final como el elemento que da origen o es receptor de información. Este usuario puede ser tanto un programa de aplicación como un dispositivo de entrada/salida.

Por el término nivel se entiende cada una de las particiones en que se ha dividido un sistema de comunicaciones. La unidad funcional o entidad es un proceso que se ejecuta dentro de un mismo nivel e implementa funciones de ese nivel. Un ejemplo de unidad funcional es un proceso, en un sistema multiproceso. Pueden existir varias unidades funcionales idénticas dentro de un nivel si se considera más eficiente para el sistema. También es posible la existencia de distintas entidades de las cuales cada una de ellas implementan funciones diferentes, por ejemplo, protocolos distintos.

Cada nivel se relaciona con el nivel inmediatamente superior e inferior a través del concepto de interfaz, que representa el conjunto de elementos lógicos y físicos existentes entre dos niveles adyacentes.

Los procesos que una unidad funcional realiza y cuyos resultados son ofrecidos o empleados por el nivel superior, se denominan servicios de nivel Estos servicios se proporcionan a través de los puntos de acceso al servicio (SAP, *Service Access Points*) de la interfaz

Los protocolos de niveles diferentes son independientes; es decir, sólo tienen que conocer la definición de servicios de su interfaz, y no tienen nada que ver con los protocolos de los restantes niveles ni con los servicios de sus interfaces

La figura 1.6 muestra la estructura interna de un nivel y su relación con los niveles adyacentes, donde se observa que:

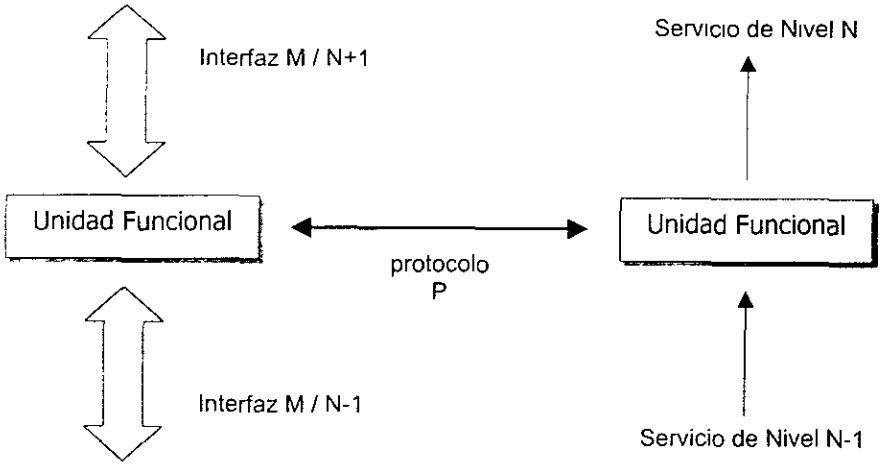


Figura 1.6 - *Estructura interna de un nivel*

1.3.3 Transmisión de datos en OSI

La comunicación entre dos nodos de una red significa que los correspondientes niveles de ambos nodos están "hablando" entre ellos. Para que dicha comunicación sea posible, *cada nodo debe tener idénticos protocolos de nivel. Esta comunicación se mantiene* mediante el intercambio de mensajes con un formato común denominados Unidades de Datos de Protocolo (PDU, *Protocol Data Unit*) Esto es, cada capa tiene su propia unidad de información y por tal motivo se les llama PDU de la capa correspondiente (por ejemplo, el PDU de la capa de enlace es la trama o frame).

Ahora veremos cómo se realiza la transmisión de datos a través de una red que sigue el Modelo de Referencia OSI. Para ello, debemos suponer dos nodos, uno emisor y otro receptor El nodo emisor pone a disposición de su nivel de aplicación los datos que desea transmitir El nivel de aplicación incorpora a los datos pasados por el nodo, informacion propia del nivel mediante datos de cabecera y cola (datos situados al inicio y al final del mensaje, respectivamente). La totalidad de la información, es decir, la cabecera más los datos, más la cola, es entregada al nivel de presentación, quien, a su vez, añade una nueva cabecera y cola propias del nivel, transfiriendo el resultado al nivel de sesión Este proceso se repite en el resto de los niveles por los cuales va pasando el mensaje hasta llegar al nivel físico.

El nivel de sesión únicamente regula el flujo de los datos y los envía al nivel de transporte, donde el mensaje es fragmentado en pequeñas unidades, y a cada unidad le es colocada una cabecera Esta cabecera incluye información de control, como la secuencia o la posición de las unidades para, posteriormente, poder reensamblar el mensaje original. En el siguiente nivel, en el de red, se les añade otra cabecera a cada uno de los segmentos del mensaje. Esta cabecera contiene el algoritmo de ruta, que determinará el camino por el que será enviado el mensaje.

El nivel de enlace de datos, recibe el mensaje y se lo envía al nivel físico, diciéndole la dirección del nodo receptor, y el camino por el cual lo debe mandar, camino que fue determinado por el algoritmo de ruta Finalmente, en el nivel físico es donde se realiza realmente la transmisión de la información. El medio físico típico por el cual se realiza la transmisión de datos es cable coaxial, par trenzado, microondas o fibra óptica, dependiendo de las características de la transmisión.

En el nodo receptor, el mensaje recibido sufre el proceso inverso al que se vio sometido en el emisor. A medida que el mensaje asciende por los niveles de la torre OSI del nodo receptor, se le quita la información de cabecera y de cola correspondiente a cada nivel. De esta forma, finalmente, los datos llegan al nodo receptor idénticos a como fueron enviados por el nodo emisor.

Como se puede observar, no existe realmente una comunicación directa entre los niveles, a excepción del nivel físico. Cuando se realiza la comunicación entre usuarios de diferentes sistemas, se establece una relación lógica entre los siete niveles de ambos, iniciando con el protocolo de nivel 7, el cual requiere los servicios del nivel 6, obligando, por lo tanto, a los dos niveles 6 a comunicarse a través de su propio protocolo de nivel 6 y así sucesivamente, hasta llegar al nivel 1 donde se realiza realmente la comunicación.

En la figura 1.7 se ilustra la estructura de un mensaje en una red con arquitectura OSI. Además cabe hacer notar que existe una clasificación de los niveles en dos grupos: de control y de transporte. Los niveles de aplicación, presentación y sesión corresponden a los de control, y son los relacionados con las necesidades de comunicación entre los usuarios finales; es decir, si dos usuarios no tuviesen necesidad de emplear una red de comunicación para comunicarse, sólo utilizarían estos niveles. Por su parte, los niveles de transporte, red, enlace y físico corresponden al grupo de transporte, y son los encargados de transferir los mensajes a través de la red.

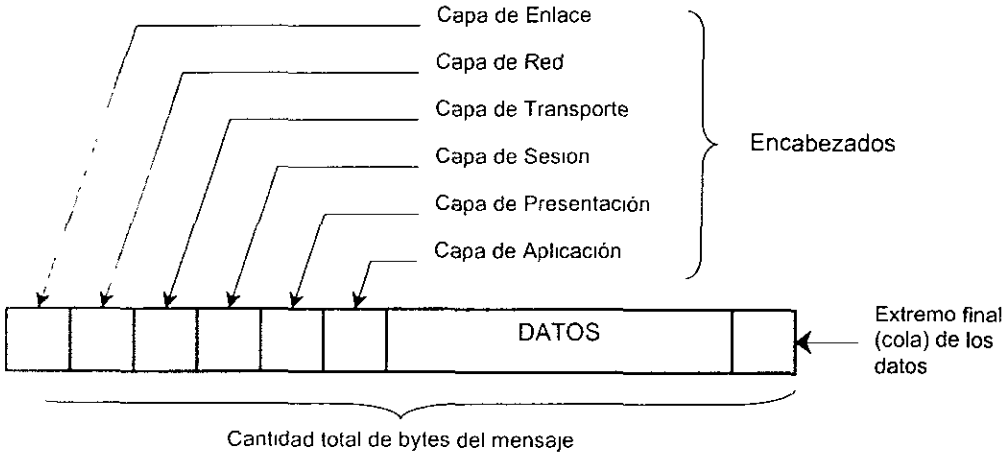


Figura 1 7 - *Estructura de un mensaje OSI*

1.3.4 Nivel de Aplicación

Este nivel es el número 7 en la arquitectura OSI y tiene como misión controlar y coordinar las funciones a realizar por los programas de usuarios, de manera que les permita el acceso al entorno OSI. Los procesos de aplicaciones se comunican entre sí por medio de las entidades de aplicación a las que están asociadas y controladas por protocolos de aplicación, utilizando servicios de presentación (de su nivel inferior inmediato, nivel 6)

Este nivel sirve como una ventana para las aplicaciones necesarias para acceder a los servicios de la red Maneja el acceso general a la red, el control de flujo y la recuperación de errores Representa los servicios que soportan directamente las aplicaciones de usuario, como el software para la transferencia de archivos

Las aplicaciones de este nivel son el correo electrónico (e-mail), aplicaciones internas de usuario, creación y aceptación de requerimientos, transferencia de archivos (FTP), login remoto (rlogin, telnet), acceso a base de datos y manejo de protocolos de alto nivel, como SNMP, NFS y recuperación de errores, los cuales serán tratados más adelante. Los protocolos FTAM y X.400 también son empleados en este nivel

Se pueden distinguir tres tipos de procesos de aplicación:

- Procesos del propio sistema. Son los que ejecutan funciones para controlar y supervisar operaciones de los sistemas conectados a la red de comunicación.
- Procesos de gestión de las aplicaciones. Son los encargados de controlar y supervisar las operaciones de los procesos de aplicación
- Procesos de aplicación de usuario. Son los que procesan la información real para los usuarios finales.

1.3.5 Nivel de Presentación

El nivel de presentación es el encargado de la transferencia de datos contenidos en los protocolos de aplicación. Establece una sintaxis y semántica de la información transmitida, es decir, intervienen los aspectos sintácticos de la información o, dicho en otras palabras, la forma o código en que se presentan los datos. En este nivel se define la *estructura de los datos a transmitir (define los campos de un registro: nombre, dirección,* teléfono, etc), y se define, asimismo, el código a usar para representar una cadena de caracteres (ASCII, EBCDIC, etc.)

Este nivel es el traductor de la red, es el responsable de la conversión de protocolos, de la traducción y el cifrado (encripción) de los datos y del manejo de la compresión de texto, imágenes y video Determina el formato usado para intercambiar datos entre computadoras conectadas a la red, o sea, es un traductor de protocolos de red. Todos los formatos diferentes de todas las fuentes de cualquier tipo son convertidos en un único formato comun, de manera que el resto del modelo OSI pueda comprenderlo

A través de este nivel, los procesos de aplicación adquieren independencia de la representación de los datos e incluyen en su entorno las posibles transformaciones de codigos.

La figura 1.8 representa la relación que existe entre los niveles de aplicación y de presentación, donde se puede ver que.

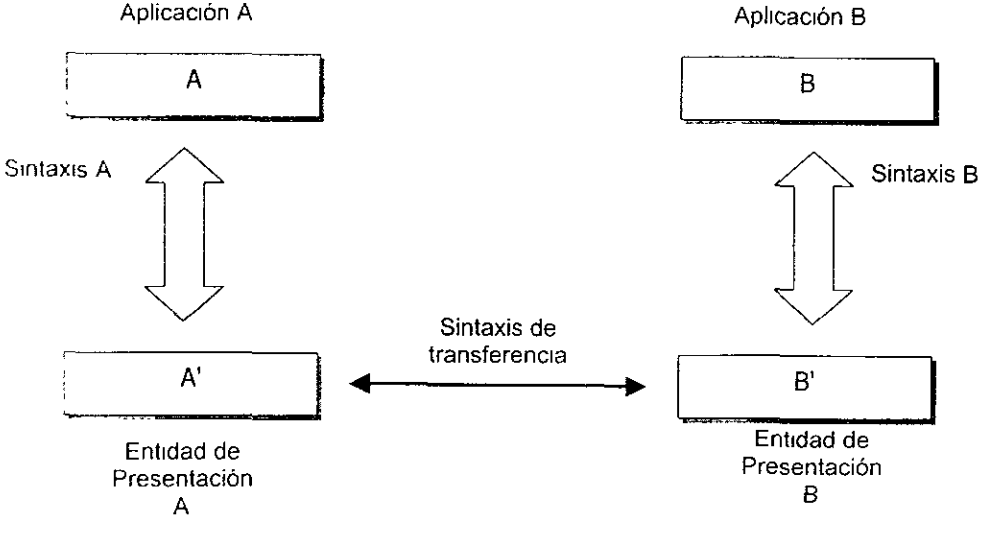


Figura 1.8 - *Relación entre los niveles de Aplicación y Presentación*

Algunos ejemplos de protocolos de presentación son, como ya mencionamos, la *compresión de datos, la criptografía (cifrado de la información para protegerla durante la* transmisión) y el protocolo de terminal virtual. Este último protocolo realiza la conversión

entre las características específicas de una terminal a las de un modelo virtual o genérico utilizado por los programas de aplicación

1.3.6 Nivel de Sesión

El propósito de este nivel es proporcionar los medios necesarios para controlar el diálogo entre entidades de presentación Este diálogo se realiza a través del establecimiento y uso de una conexión, denominada sesión Es decir, permite a usuarios en diferentes máquinas establecer una sesión Una sesión puede ser usada para efectuar un login (un registro) en un sistema de tiempo compartido remoto, para transferir un archivo entre dos máquinas, etc.

Los servicios proporcionados por el nivel de sesión son los siguientes:

- Establecimiento de la conexión de sesión. Se realiza la conexión de dos entidades de presentación a petición del usuario.
- *Controla el diálogo (quién habla, cuándo, cuánto tiempo, a qué velocidad*, half duplex o full duplex) durante el intercambio de datos Es el servicio que permite la transferencia de datos.
- Proporciona la función de sincronización y mantenimiento de la sesión. Se realiza la sincronización y control de la comunicación de manera que se produzca un intercambio ordenado de datos.
- Ejecuta el reconocimiento de nombres y contraseñas como medidas de seguridad para controlar el acceso a la información.
- Supervisa el control total de los paquetes TCP, llamados "pipes".
- Liberación de la conexión de sesión. Una vez finalizado el intercambio de datos, se procede a la desconexión.

Estos tres primeros niveles son los denominados niveles de control, que proporcionan las aplicaciones de los servicios de red a los usuarios.

1.3.7 Nivel de Transporte

El objetivo del nivel de transporte es proporcionar un mecanismo fiable para el intercambio de datos entre procesos en diferentes sistemas. Asegura una entrega libre de errores Este mecanismo independiza al nivel de sesión y niveles superiores de los elementos de comunicación que constituyen la red; es decir, oculta a los niveles superiores los detalles específicos de la red a través de la cual se transmite la información

El nivel de transporte establece conexiones punto a punto sin errores para el envío de mensajes, pasa los datos del nivel de sesión al nivel de red, fragmentándolos en unidades mas pequeñas si es necesario y asegurando que todos llegan correctamente a su destino. Los diversos protocolos tienen diferentes requerimientos para la longitud de datos por

paquete, por ejemplo, Token ring, Ethernet y ATM usan todos ellos, diferentes longitudes de datos para sus paquetes, y estos son distribuidos y redistribuidos en este nivel

Para realizar lo anterior emplea funciones de direccionamiento y multiplexión (permite multiplexar una conexión punto a punto entre diferentes procesos del usuario, puntos extremos de una conexión), se encarga del establecimiento de la conexión, desconexión y de la transferencia y control de flujo de los datos y, por último, provee la función de difusión de mensajes (broadcast) a múltiples destinos.

Este nivel garantiza que dentro del conjunto de paquetes que conforman el mensaje (PDU de esta capa), éstos estén formados en secuencia, sin omisiones ni duplicaciones.

El nivel de transporte ofrece, además de los servicios de detección y corrección de errores, para asegurar la integridad de los datos, niveles de calidad del servicio (QoS, *Quality of Service*). Por ejemplo, una entidad de sesión podría especificar tasas de errores aceptables, retardo máximo y prioridad.

La complejidad del protocolo de transporte dependerá del tipo de servicio ofrecido por el nivel de red. Cuanto más fiable sea el servicio proporcionado por el nivel de red, más sencillo o menos funciones incluirá el protocolo del nivel de transporte.

El equipo terminal que se emplea en los niveles hasta ahora tratados, es el protocolo propio de cada nivel (protocolo de aplicación, de presentación, etc.). Los protocolos de estos cuatro niveles sólo residen en los equipos de los usuarios finales y no en los equipos intermedios. En realidad no son de niveles de comunicaciones sino que tan sólo emplean los niveles inferiores para realizar la transmisión de información. Como ya se mencionó anteriormente, a estos cuatro niveles se les conoce como niveles de servicios. El equipo intermedio que emplea cada uno de los niveles es un gateway (los gateways son traductores entre modelos de comunicación incompatibles).

Ejemplo de protocolo de nivel de transporte es TCP.

1.3.8 Nivel de Red

La comunicación generalmente tiene lugar en el ámbito de una red, sea ésta pública o privada, compuesta por nodos. Este nivel es el responsable de asegurar que la información sea transmitida correctamente a través de la red, direccionándola, determinando rutas para enviarla, realizando la conmutación de paquetes y haciendo la traducción de direcciones lógicas en direcciones físicas.

Proporciona a las entidades del nivel de transporte una transferencia de datos transparente. En este sentido, libera al nivel de transporte de la necesidad de conocer los mecanismos de transmisión de datos o tecnologías utilizadas para conectar los sistemas.

Este nivel utiliza el nivel de enlace para el envío de paquetes. un paquete es encapsulado en una trama. Forma los paquetes basándose en los mensajes PDU de las capas superiores, añadiéndoles direcciones lógicas de fuente y de destino y realizando la macrofragmentación. Realiza el enrutamiento de paquetes, enviándolos de nodo a nodo como datagramas, o bien, empleando circuitos virtuales. Se encarga también del control de congestión, para resolver problemas de tráfico en la red.

El nivel de red tiene también como funciones la conexión y desconexión de las redes, sincronización y control del flujo de las transferencias y la detección de errores en la

transmisión, recuperándolos en caso necesario. En el caso de que hubiera más de una red implicada en la transmisión, también tiene como función el encaminamiento entre redes.

El equipo terminal de este nivel es el protocolo ruteado, el equipo intermedio lo conforman los ruteadores y los protocolos ruteador y ruteado. Cada modelo de comunicación tiene su propio esquema de direccionamiento y, por lo tanto, tiene sus propios protocolos ruteadores y ruteado.

Los protocolos que se emplean en el nivel de red son de alto nivel o de ruteo· IP, Apple, IPX, NDIS3, SPX y X 25 nivel 3.

1.3.9 Nivel de Enlace de Datos

El nivel de enlace de datos define el protocolo que las estaciones deben seguir para tener acceso a la red para la transmisión y recepción de información Es el responsable de mantener la integridad de los datos de una transmisión sobre un canal de comunicaciones. Es decir, proporciona un canal confiable entre dos puntos adyacentes de la red para la transmisión de datos sobre un medio físico, por lo general, no exento de ruido Para ello, entre sus funciones se encuentran las de detección y corrección de errores de transmisión que pudieran ocurrir en el medio físico. Proporciona servicios orientados a conexión a la capa de red.

Este nivel estructura el flujo de bits bajo un formato predefinido llamado trama. Para formar una trama (PDU del nivel 2), el nivel de enlace agrega a los paquetes (PDU del nivel 3) una secuencia especial de bits al principio y al final del flujo inicial de bits. Esta secuencial especial de bits contiene las direcciones físicas de origen y de destino, un campo que indica la longitud de la información a ser enviada, un campo que permite verificar que no se ganó o perdió información durante la transmisión para la verificación de integridad, información de control, banderas de sincronía y delimitadores.

Es a todo este conjunto de campos que forma la unidad de información a ser transmitida, al que se le conoce con el nombre de trama o frame La estructura de la trama tiene que ver con el método de acceso o protocolo del nivel 2, que especifica cómo evitar que dos o más estaciones transmitan al mismo tiempo y el resultado de dicha colisión distorsione las tramas. Este nivel transfiere tramas de una forma confiable libre de errores (utiliza reconocimientos y retransmisión de tramas), provee control de flujo y sincronización

En las redes de área amplia (WAN, *Wide Area Network*), el nivel de enlace de datos es una capa monolitica. En las redes locales (LAN, *Local Area Network*) y en las metropolitanas (MAN, *Metropolitan Area Network*), este nivel está compuesto por dos subcapas La función de la subcapa inferior es la de manejar el protocolo específico al tipo de red, y la de mantener un diálogo estándar con la subcapa superior. La función de la subcapa superior es la de mantener un diálogo estándar con cada una de las posibles subcapas inferiores y con el nivel de red.

A la subcapa superior se le conoce como Control de Enlace Lógico (LLC, *Logical Link Control*) y, como se mencionó arriba, es la subcapa homogeneizadora entre diferentes subcapas inferiores, para permitir la interconectividad entre diferentes tipos de redes. El

LLC controla el enlace y define los Puntos de Acceso a los Servicios (SAP, *Service Access Points*).

A las subcapas inferiores se les conoce con el nombre de Control de Acceso al Medio (MAC, *Media Access Control*) y manejan el protocolo y señalización específicos del tipo de red. Los tipos de red más populares que se emplean son los siguientes: Ethernet, Token Ring, FDDI, Fast Ethernet y ATM (como emulación de una LAN) Esta subcapa es la que está más relacionada con el medio físico, se comunica con la tarjeta adaptadora, maneja el método de acceso para la transmisión confiable libre de errores, realiza la microfragmentación y define los puntos de interfaz lógicos

El equipo terminal de este nivel es la tarjeta adaptadora y el protocolo propio; el equipo intermedio está formado por los puentes (bridges) y los conmutadores (switches).

Los protocolos del nivel de enlace definen el establecimiento y liberación de un enlace de datos, controlan la correcta transferencia de información y recuperación de anomalías, así como la gestión del propio nivel.

Los protocolos que se emplean en el nivel de enlace son protocolos de bajo nivel[3], a saber BSC (*Binary Synchronous Communications*), HDLC, LAP-B (*Link Access Protocol - D Channel*), X.25 nivel 2 y LLC.

1.3.10 Nivel Físico

El nivel físico es el responsable de la definición de las características mecánicas (cables y conectores), eléctricas (voltajes, niveles de corriente y técnicas utilizadas para modular la señal), funcionales y las especificaciones de procedimientos para la transmisión y recepción de la información utilizando un medio de comunicación específico. Esto es, describe la interfaz a nivel eléctrico, electromagnético o luminoso, tanto en lo mecánico como en lo funcional. Entre sus funciones básicas se encuentran la identificación de los circuitos de datos, el secuenciamiento de los mismos, la transmisión de datos libre de errores y la gestión de nivel.

En este nivel se realiza la transmisión de flujo de bits a través del medio. No existe estructura alguna. Maneja voltajes y pulsos eléctricos. Especifica cables, tarjetas, conectores y componentes de interfaz con el medio de transmisión, es decir, es el hardware. Define cómo el medio de transmisión (por ejemplo, cable o fibra óptica) debe ser unido con el adaptador de red y define, asimismo, la topología de las redes (comité IEEE 802).

La interfaz física es el punto de contacto entre el equipo y el medio de comunicación definida en el puerto del equipo, ahí es donde se generan los impulsos eléctricos, luminosos o electromagnéticos que se propagan por el medio. En este nivel se define la función de cada pin de los conectores, los voltajes y otras características requeridas para que la señal se propague eficientemente por el medio Para lograr que las señales se

---

Dentro de los niveles de comunicaciones, con frecuencia se hace referencia a los protocolos de bajo nivel y a los de alto nivel. Los protocolos del nivel de enlace de datos (nivel 2) son los que se conocen con el nombre de protocolos de bajo nivel o, en ocasiones, como los tipos de red. Por su parte, los protocolos del nivel de red (nivel 3) son los conocidos como protocolos de alto nivel o, simplemente, como los protocolos.

propaguen adecuadamente por el medio, se emplean diferentes equipos intermedios que afectan la señal en forma específica.

Los equipos intermedios del nivel físico son: repetidor o hub, amplificador, estrella pasiva, multiplexor, concentrador de terminales, módem, códec, CSU (*Channel Service Unit*), DSU (*Data Service Unit*), transceiver, transductor, balún/filtro[4], etc.

El equipo terminal está compuesto por las tarjetas adaptadoras y los puertos.

A estos dos últimos niveles juntos, se les conoce como nivel de hardware o nivel de servicios de red.

Ejemplos de protocolos de nivel físico son el V.24, el RS-232-C y el X.21.

Generalmente, cuando nos referimos al modelo OSI, pensamos en la representación de muchas aplicaciones diversas. Comúnmente, las diferencias en cualesquiera aplicaciones son los dispositivos o equipos terminales e intermedios, y los protocolos existentes en cada nivel. Veamos como ejemplo un sistema de comunicaciones basado en el Modelo de Referencia OSI, tal como la Red Digital de Servicios Integrados (ISDN, *Integrated Service Digital Network*).

Consideraremos los dos primeros niveles para mostrar que cada uno tiene su propio protocolo y que es diferente de los empleados por otros sistemas de comunicaciones.

El nivel físico de ISDN emplea un protocolo llamado 2 Binario 1 Cuaternario (2B1Q, *2 Binary 1 Quaternary*). Este protocolo es el método de señalización más común en las interfaces U. El 2B1Q es un protocolo para la transmisión de datos en este nivel y puede comparársele con el protocolo X.21.

Por otro lado, el protocolo empleado en el nivel de enlace de datos en ISDN, es el Protocolo de Acceso al Enlace - Canal D (LAP-D, *Link Access Protocol - D Channel*). Este protocolo es casi idéntico al empleado en el modelo X.25, el LAP-B, o al protocolo X.28, utilizado para enlazar terminales en una red X.25.

En la tabla 1.1 se presentan los servicios proporcionados por los diversos niveles del Modelo de Referencia OSI:

Nivel	Nombre OSI	Servicios
7	Aplicación	Terminal virtual - Correo - Transferencia de archivos
6	Presentación	Compresión - Códigos - Formatos
5	Sesión	Conexión - Desconexión - Control de flujo
4	Transporte	Extremo a extremo - Circuito virtual - Datagramas
3	Red	Transferencia de paquetes - Direcciones - Ruteo
2	Enlace	Formación de tramas - Sincronía - Corrección de errores
1	Físico	Transmisión de bits sin estructura - Voltajes - Hardware

Tabla 1.1 - *Funciones y servicios de los niveles OSI*

[4] Los balunes/filtros permiten enlazar cables metálicos con características diferentes, evitan que se produzcan reflejos y permiten el paso de ciertas frecuencias y bloquean el paso de otras.

1.3.11 Especificaciones de los servicios

Cada uno de los niveles del Modelo de Referencia OSI se define mediante el protocolo existente entre las entidades del mismo nivel en sistemas diferentes y por los servicios proporcionados por un nivel al inmediatamente superior en el mismo sistema.

El protocolo se define de manera precisa en los términos del formato de las Unidades de Datos de Protocolo (PDUs) intercambiadas, la sintaxis de cada campo en la PDU y la secuencia de acciones permitidas en el intercambio. La implementación del protocolo puede ser cualquiera, con tal que se conserve la sintaxis del mismo Lo esencial es asegurar que cada uno de los sistemas que intervienen en la comunicación tiene la misma percepción del protocolo que están empleando independientemente de las implementaciones particulares en los sistemas locales.

Los niveles equivalentes de distintos nodos se comunican mediante el intercambio de PDUs Estas unidades se anidan a medida que transitan por cada uno de los niveles de la torre OSI En cada uno de estos niveles, a los datos pasados por el nivel superior se les añade cierta información de control propia del nivel actual, denominada comúnmente "cabecera" (*header*), que posteriormente será eliminada por el mismo nivel en el nodo receptor En la figura 1.9 aparece la estructura de las PDUs.

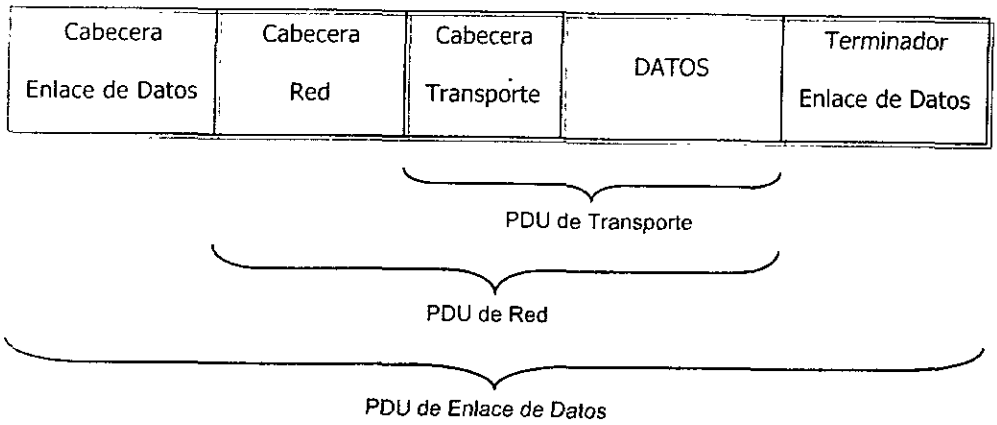


Figura 1.9 - *Unidades de Datos de Protocolo*

En el nivel de enlace (o en el de control de acceso al medio en las redes locales) se añade, además, la información consistente en bits para la detección de errores, denominada "terminador" (*trailer*) y, en el caso de algunas redes en anillo, se añade un campo de respuesta para indicar si la PDU transmitida fue aceptada en el destino.

En ocasiones, el protocolo de nivel, en el momento de transmitir, debe dividir la información pasada por el nivel superior en unidades más pequeñas. Esto dependerá de las características de la red, que no tienen por qué ser conocidas por el nivel superior. La información transferida a través de los Puntos de Acceso al Servicio (SAP) desde un nivel

superior a uno inferior, necesita, por tanto, ser diferenciada de la información intercambiada por el protocolo del nivel inferior (PDU); a la primera se le denomina Unidad de Datos de Servicio (SDU, *Service Data Unit*).

Los servicios proporcionados entre niveles adyacentes, se caracterizan mediante un conjunto de primitivas de servicio, que contienen el nombre del servicio y los parámetros o información necesaria que debe pasarse a través de los puntos de acceso al servicio.

Los servicios proporcionados a través de la interfaz de dos niveles adyacentes, se expresan en términos de primitivas y parámetros. Una primitiva especifica la función a realizar y los parámetros se utilizan para transferir información de datos y control. Algunos ejemplos de implementación de primitivas son las macros READ, WRITE, GET, PUT, CALL.

En el modelo OSI se emplean cuatro tipos de primitivas:

- *Petición*. esta primitiva se emite por el usuario del servicio con el objeto de solicitar un servicio y pasar los parámetros necesarios para realizar el servicio solicitado.

- *Indicación*: esta primitiva se emite por el suministrador del servicio para notificar al usuario del servicio de una acción iniciada por el proveedor o para indicar que ha sido invocado un procedimiento por parte del usuario del servicio del extremo remoto de la conexión y para proporcionar los parámetros necesarios

- *Respuesta*. esta primitiva se emite por el usuario del servicio para reconocer o completar algún procedimiento previamente indicado por una *Indicación* a dicho usuario.

- *Confirmación*: esta primitiva se emite por el suministrador del servicio para reconocer o completar algún procedimiento solicitado previamente por un usuario del servicio mediante un *Petición*.

Para describir la relación de las primitivas de servicio entre el usuario y el suministrador del servicio en un extremo de la comunicación, y el suministrador del servicio y el usuario en el otro extremo, se utilizan los diagramas de secuencias temporales.

En estos diagramas, los niveles de los usuarios del servicio se sitúan a la izquierda y a la derecha del nivel del suministrador. El usuario local se encuentra a la izquierda, donde se inician, por lo general, las actividades, y el usuario remoto en la derecha, representando cualquier nodo remoto de la red. Las líneas verticales representan los *SAP y el transcurso del tiempo* se muestra de arriba abajo del diagrama Las primitivas se representan mediante flechas, sin que exista ninguna relación temporal entre las primitivas del usuario local y las del usuario remoto, a no ser que se indique expresamente mediante líneas punteadas, como se indica en la figura 1.10.

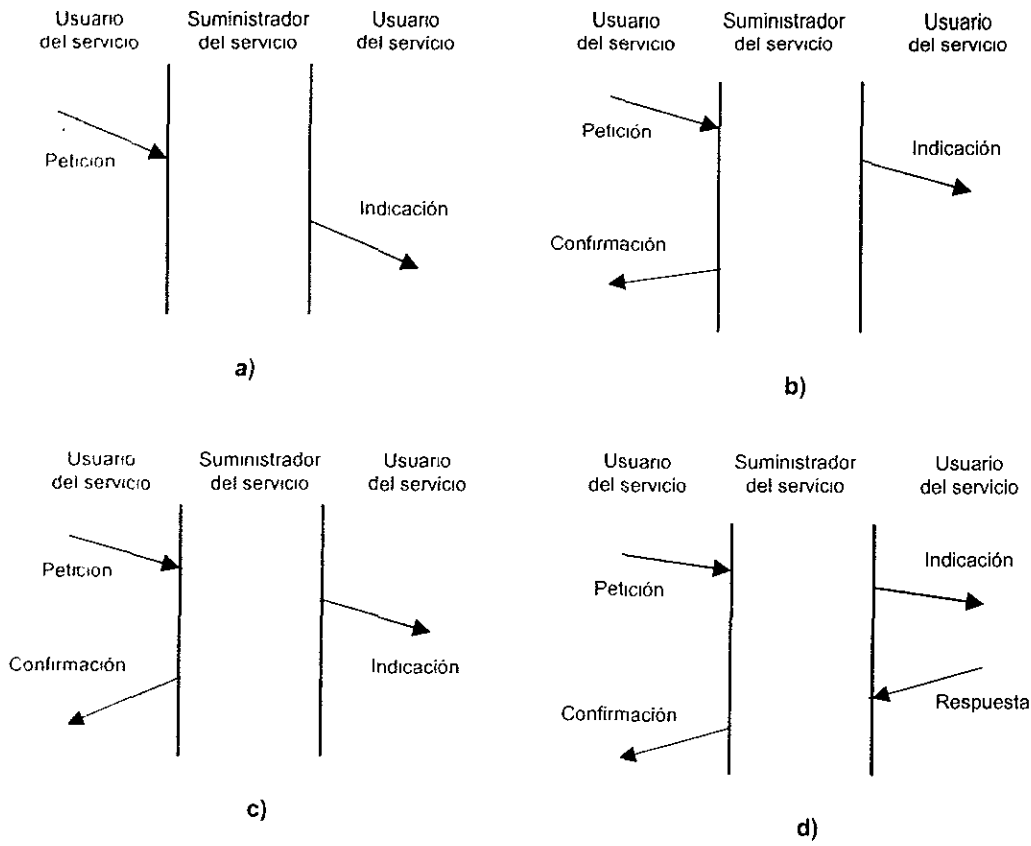


Figura 1.10 - *Relaciones entre primitivas*

El diagrama más clásico está representado por el diagrama 1.10.d, para cuando se logra realizar una conexión completa (servicio orientado a conexión); o bien, por 1.10.a, cuando no hay confirmación. Por su parte, los diagramas 1.10.b y 1.10 c se utilizan fundamentalmente en los estándares para redes locales

Para ejemplificar lo anterior, consideraremos el establecimiento de una conexión entre el usuario A ubicado en un sistema A y el usuario B en otro sistema B. La secuencia de eventos sería la siguiente.

a) El usuario A solicita los servicios del nivel inferior para el establecimiento de una conexión mediante la primitiva <conexión>.*Petición*. Junto con la primitiva incluye los parámetros necesarios, tales como la dirección del emisor y la dirección del receptor.

b) Con la información pasada por el usuario A, el suministrador del servicio del sistema A genera la unidad de datos de protocolo PDU, que transferirá al suministrador de servicio del sistema B.
c) El suministrador de servicio del sistema B comunica al usuario B la solicitud de una conexión por parte del usuario A, mediante la primitiva <conexión>.*Indicación*, incluyendo la información sobre la dirección del emisor.
d) El suministrador de servicio de B transmite al suministrador de servicio de A, a *través de una PDU, el reconocimiento del usuario B, mediante la primitiva* <conexión>.*Respuesta*
e) El reconocimiento es enviado al usuario A por su suministrador de servicio mediante la primitiva <conexión> *Confirmación.*

1.4 CAPA DE APLICACION

Como ya lo explicamos en el subtema 1.2.2, la capa de aplicación tiene como función invocar programas que acceden servicios en la red y que interactúan con uno o más protocolos de transporte para enviar o recibir datos, en forma de mensajes o bien en forma de flujos de bytes

Se debe indicar que todas las aplicaciones TCP/IP se basan en el modelo cliente/servidor. En esta capa se encuentran las aplicaciones disponibles para los usuarios

Las aplicaciones que proporciona esta capa son las siguientes:

- Llamadas a procedimientos remotos (RPC)
- Conexión remota (TELNET)
- Correo electrónico (SMTP)
- Protocolos de transferencia de archivos (FTP, TFTP)
- Sistema de Archivos de Red (NFS)
- Sistema de Nombre de Dominios (DNS)
- Manejo de hipertexto (HTTP)

A continuación, veremos cada una de las aplicaciones de esta capa en forma detallada, explicando cómo funcionan y qué protocolos emplean.

1.4.1 Llamadas a Procedimientos Remotos (RPC)

Un RPC es una llamada a un procedimiento que se ejecuta en un sistema diferente del que realiza la llamada, esta es la razón por la que el procedimiento se denomina "remoto". Es la manera más sencilla que posee un cliente para solicitar un servicio. El servidor ofrece uno o más conjuntos de procedimientos remotos y el cliente realiza llamadas a los procedimientos que le ofrece el servidor. Se trata de una simple llamada a un

procedimiento con su lista de parámetros (si fuera necesario). Los RPC son uno de los procedimientos utilizados para la realización de aplicaciones distribuidas, junto con el Método Conversacional[5] y la Cola de Mensajes[6].

Estos tres métodos conforman los denominados servicios de comunicación entre procesos, los cuales son necesarios para que los procesos cliente emitan sus peticiones a los procesos servidores, y éstos envíen los resultados a sus clientes

En una RPC, el código de programa que realiza la llamada y el procedimiento llamado se comunican a través de una "interfaz RPC", que consiste en un conjunto de operaciones y datos que sirven de "contrato" para un conjunto de procedimientos remotos.

RPC sigue el esquema cliente/servidor. El proceso petitorio (cliente) envía un mensaje al proceso servidor y espera una respuesta Por otra parte, el proceso servidor se encuentra sumido en un estado de espera de peticiones y, al recibir un mensaje de un cliente, estudia los parámetros del procedimiento solicitado, obtiene los resultados y los envía de vuelta al proceso cliente mediante un mensaje de respuesta.

Existen dos tipos de servidores:

- El **servidor iterativo**: es un servidor que inicialmente se encuentra a la espera de peticiones. Cuando le llega un mensaje de petición procedente de un cliente, abandona el estado de espera, proporciona el servicio que le ha sido requerido (devuelve los resultados) y vuelve al estado de espera de nuevas peticiones.
- El **servidor concurrente**: es un servidor que, al igual que el anterior, inicialmente se encuentra en estado de espera de peticiones. Si le llega un mensaje de petición, contesta al mensaje enviando al cliente un número de puerto (a través del cual el cliente recibirá su servicio) e inmediatamente arranca un proceso paralelo que presta el servicio requerido al cliente. Tras el inicio del proceso paralelo, el servidor vuelve al estado de espera en el que se encontraba inicialmente

1.4.2 Conexión Remota (TELNET)

Principios de operación

Este tipo de aplicación permite que un usuario en una terminal pueda acceder a recursos y aplicaciones de varias computadoras, a través de redes, apareciendo para el

---

En el Método Conversacional, tanto el cliente como el servidor pueden abrir, cerrar o usar el enlace de comunicación. Seguidamente, se establece un protocolo de petición/respuesta entre cliente y servidor de tal manera, que si uno toma el control de la conversación, el otro extremo debe escuchar Esta situación puede cambiar durante el transcurso de la conversación.

La comunicación entre el cliente y el servidor se consigue de forma indirecta mediante colas de mensajes El cliente coloca su petición en la cola asociada al servidor y cuando el servidor está disponible, atiende la petición dejando los resultados en la cola del cliente. Con ello se consigue un paralelismo mayor que con los otros métodos. Se utiliza en sistemas transaccionales de tipo no conversacional y de difusión (broadcasting).

usuario de la terminal como si estuviese conectado localmente al ordenador remoto. Esto es, emula una conexión remota como local. Para poder conectar terminales heterogéneas a computadoras también heterogéneas es necesario definir lo que se denomina un Protocolo de Terminal Virtual (VTP, *Virtual Terminal Protocol*). Un VTP es un protocolo que realiza básicamente las siguientes funciones.

- Establecimiento y mantenimiento de conexiones.
- Control del diálogo para negociar las acciones permitidas durante la conexión.
- Creación y mantenimiento de una estructura que representa el estado de la terminal
- *Traslación de las características* de la terminal real a la representación normalizada

El objetivo principal de una terminal virtual consiste en transformar las características de una terminal real en una terminal normalizada. Es prácticamente imposible definir una única terminal virtual que pueda realizar todas las funciones de las terminales existentes, por lo que normalmente se definen las funciones básicas, como modo de línea para terminales sin inteligencia local, tipo teclado - pantalla / impresora, modo de página en la que los caracteres pueden direccionarse a nivel de página mediante un cursor o modo gráficos

En general, un VTP tiene las siguientes fases de operación:

- Establecimiento y liberación de la conexión.
- Negociación.
- Control
- Transferencia de datos

La negociación se utiliza para determinar el conjunto de características del diálogo entre los extremos de la conexión. El control realiza el intercambio de información de control y mandatos. En la figura 1.11 se representa la arquitectura de niveles de la terminal virtual, donde:

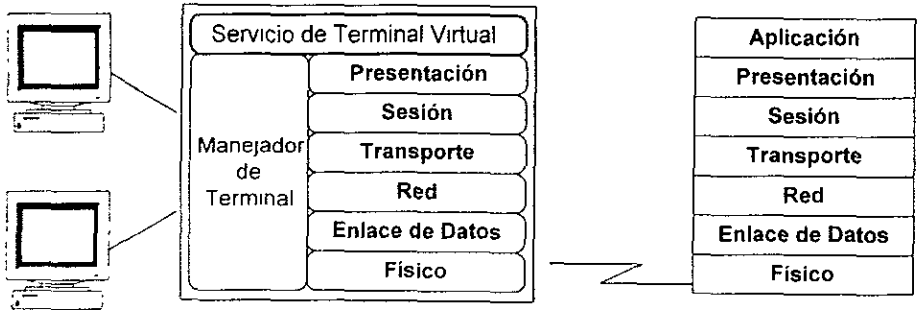


Figura 1.11 - *Arquitectura de Terminal Virtual*

Entre las VTP más conocidas está TELNET, que fue desarrollada sobre los protocolos TCP/IP ISO quiso definir una VTP más generalizada y uno de los conjuntos básicos que adoptó fue precisamente TELNET:

TELNET se fundamenta en tres principios:

- La arquitectura de Terminal Virtual (NVT, *Network Virtual Terminal*).
- Simetría entre terminales y procesos.
- Opciones negociadas.

En la práctica, el servidor es más complejo de lo que se muestra en la figura 1.11, debido a que maneja múltiples conexiones concurrentes. Normalmente, un proceso servidor-maestro espera nuevas conexiones y genera procesos esclavo o hijos que manejan cada una de ellas. El servidor de TELNET que se muestra en la figura 1.11, representa al proceso esclavo que maneja una conexión en particular.

Cuando un usuario invoca la aplicación TELNET, un proceso de usuario se convierte en la aplicacion cliente. Este cliente establece una conexión TCP con el servidor, a través de la cual se comunicarán ambas entidades. Una vez establecida la conexión, el cliente va aceptando los datos que teclea el usuario y se los envía al servidor.

El procedimiento de operación es similar al siguiente:

telnet nombre_de_host	Ejemplo.	**telnet asc.unam.mx**
Username: **xxxxx**		Username: **einstein**
Password: **yyyyy**		Password **E=mc2**

El puerto estándar del servidor TELNET es el puerto 23 (RFC-1700, *Assigned numbers*). Se pueden utilizar submandatos. Para ello, se introduce únicamente el mandato telnet.

Para permitir que TELNET interopere con diferentes sistemas, se debe considerar la heterogeneidad de los mismos, así como de sus sistemas operativos. Por ejemplo, algunos sistemas requieren que las líneas de texto finalicen con el carácter "Retorno de Carro" (*Carriage Return* - CR); otros, sin embargo, requieren el carácter "Alimentación de Línea" (*Linefeed* - LF), mientras que otros requieren ambos.

Para permitir esa heterogeneidad de los sistemas, TELNET define cómo han de ser los datos y las secuencias de mandatos que han de circular por la red, definición conocida como *Terminal Virtual de Red* (NVT). En la figura 1.12 se muestra el mecanismo con el cual se consigue este propósito.

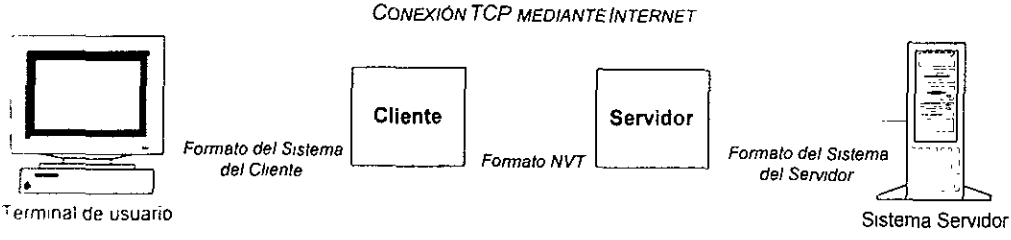


Figura 1.12 - *Mecanismo de Terminal Virtual de Red utilizado por TELNET*

La comunicación puede ocurrir entre dos terminales, dos procesos o bien entre una terminal y un proceso.

Mandatos de Control

El NVT de TELNET provee funciones de control para pasar del cliente al servidor. Por ejemplo, NVT define una tecla conceptual que posibilita la interrupción de un programa La lista de las funciones de control que proporciona NVT se muestra en la tabla 1 2

Señal	Significado
IP	Interrupción de proceso
AO	Interrumpir salida (descartar búfer de salida)
AYT	Verificar si el servidor responde
EC	Borrar carácter anterior
EL	Borrar línea
SYNCH	Sincroniza
BRK	*Interrupción*

Tabla 1.2 - *Funciones de control de NVT*

En la práctica, la mayoría de las terminales no tienen definidas teclas extras para realizar determinados mandatos El diseño de NVT separa estos mandatos del conjunto de los caracteres ASCII, por dos razones

1 La definición de funciones de manera individual le otorga gran flexibilidad. De este modo puede transferir cualquier secuencia de caracteres entre el cliente y el servidor
2 Mediante la separación de las señales de datos, NVT permite que el cliente defina *señales de manera no ambigua.*

Para transferir funciones de control mediante conexiones TCP, TELNET las encapsula en secuencias de escape, que son octetos reservados. Existe un octeto reservado, conocido por las siglas IAC (Interpretar Como Mandato - *Interpret As Command*), como comienzo de una secuencia de escape.

1.4.3 Correo Electrónico (SMTP)

El correo electrónico es probablemente el servicio más popular entre los usuarios de red, debido principalmente a que proporciona una transferencia de información de manera rapida, barata y eficiente Un usuario puede enviar correo a otro o mantener una conversación con un grupo de personas.

El correo no es interactivo. Cuando un usuario tiene un mensaje que enviar a otro usuario que no está conectado, el sistema del correo debe tomar ese mensaje y guardarlo en una cola

Existen, por tanto, dos partes conceptualmente diferentes en un sistema de correo: por un lado, el proceso de la sección de entrada (llamado *front-end*) que acepta correo de un usuario y lo coloca en un área de almacenamiento temporal (llamada *spool*), mientras que, por otro, existe un proceso que extrae esos mensajes del área de spool y los envía al destino

De esta forma, un usuario puede comunicarse con otros aunque éstos no estén activos El área en donde se depositan los mensajes recibidos hasta que el destinatario los recibe, se denomina buzón.

Las funciones básicas de un correo electrónico son las siguientes:

- *Creación* El usuario crea y edita un mensaje, generalmente utilizando medios locales de edición.
- *Emisión*: Se envia el mensaje a los destinatarios y se almacena en los correspondientes buzones
- *Recepción*: El destinatario accede al mensaje almacenado para efectuar su lectura
- *Almacenamiento* Tanto el emisor como el destinatario pueden almacenar el mensaje en un archivo

El Protocolo SMTP

TCP/IP define un estándar para el intercambio de correo entre dos máquinas, denominado Protocolo Simple de Transferencia de Correo (SMTP, *Simple Mail Transfer Protolcol*). Este estándar especifica el formato exacto de los mensajes que un cliente debe enviar en una máquina al servidor en la otra. El protocolo SMTP especifica qué mensajes deben intercambiar las máquinas, pero no especifica cómo debe almacenarse el correo o con qué frecuencia se debe intentar enviar mensajes.

El objetivo de SMTP es transferir correo electrónico segura y eficazmente. SMTP es independiente del subsistema de transmisión y requiere sólo de un canal de datos fiable y ordenado Un rasgo importante de SMTP es su capacidad para retransmitir correo a través de entornos del servicio del transporte. Un servicio de transporte provee e interprocesa un entorno de comunicación (IPCE). Un IPCE cubriría una red, varias redes, o un subconjunto de una red. Es importante hacer notar que el sistema del transporte (o IPCE) no es punto-a-punto con redes. Un proceso se puede comunicar directamente con otro proceso por cualquier IPCE conocido. El correo electrónico es una aplicación o uso de comunicación entre procesos. El correo se puede comunicar entre procesos en IPCEs diferentes por transmisión de un proceso que ya se conectó con dos o más IPCEs. Más específicamente, el correo se puede transmitir entre hosts en diferentes sistemas de transporte o por un host en ambos sistemas del transporte

La comunicación entre el cliente y el servidor se lleva a cabo mediante mensajes legibles Aunque SMTP define un formato para los mandatos, una persona puede leer fácilmente las interacciones entre cliente y servidor.

Inicialmente, el cliente establece una conexión al servidor con la orden "MAIL", que indica la identificación del emisor, y espera que le devuelva el mensaje con el formato "READY FOR MAIL". Si el servidor está sobrecargado, debe retardar el envío de los datos, mediante el mensaje "550". Una vez que el cliente recibe este mensaje, envía un mandato con el formato "HELO". El fin de línea marca el fin del mandato. El servidor responde identificándose con el mensaje "OK 250". Una vez que la comunicación ha sido establecida, el emisor puede transmitir uno o más mensajes de correo, finalizar la conexion (con el comando "QUIT") o interrogar al servidor para intercambiar papeles de emisor y receptor (utilizando el comando "TURN") y permitir el intercambio de mensajes en sentido opuesto. El receptor debe enviar la conformidad con la recepción de cada mensaje, mediante el mensaje "OK 250".

Las transacciones de correo comienzan con un mandato que tiene el formato "MAIL", que envia el cliente junto con un identificador de campo con el formato "FROM", que

contiene la dirección a donde deben ser enviados los mensajes de error. Se preparan estructuras de datos para recibir nuevos mensajes de correo, respondiendo el servidor al mensaje "MAIL" con el mensaje "OK". Esta respuesta significa que todo salió bien.

Tras un mandato MAIL llevado a cabo con éxito, el emisor envía una serie de mandatos RCPT que identifican los recipientes del mensaje de correo El receptor recibe cada uno de ellos y contesta con un mensaje del tipo "OK" si está bien o con un mensaje de error "550 *No such user here*".

Tras esto, se envían un mandato con el formato "DATA". Si es aceptado, el receptor devuelve una respuesta intermedia del tipo 354 (comienzo de la entrada de datos) y toma todas las líneas sucesivas como texto del mensaje. El receptor contesta con mensajes "*Start mail input*" y especifica una secuencia de caracteres usados para finalizar los mensajes de correo La secuencia de finalización consiste en 5 caracteres: retorno de carro, line feed, punto, retorno de carro y line feed.

Hay que señalar que los sistemas de correo normalmente ocultan al usuario las interacciones descritas, proporcionando interfaces mucho más amigables.

A continuación presentamos un ejemplo de un procedimiento SMTP. Este ejemplo de utilizacion del SMTP, muestra un mensaje enviado por Jorge, del host Alpha.ARPA, a Juan, Martin y Luis, del host Beta.ARPA. Aquí se asume que el host Alfa está conectado con el host Beta directamente.

E: MAIL FROM:<Jorge@Alfa ARPA>
R: 250 OK
E: RCPT TO.&lt,Juan@Beta ARPA>
R 250 OK
E: RCPT TO:<Martin@Beta.ARPA>
R: 550 No such user here
E RCPT TO:<Luis@Beta.ARPA>
R: 250 OK
E: DATA
R 354 Start mail input; end with <CRLF>.&lt,CRLF>
E: ...etc. etc. etc.
E: <CRLF>.<CRLF>
R: 250 OK

El mensaje ha sido aceptado por Juan y Luis. Martín no tenía una cuenta de correo (mailbox) en el host Beta.

Los comandos están compuestos del código del comando (nombre) seguidos por un campo de argumento. El código de comando se compone de cuatro caracteres alfabéticos. El campo de argumento consiste en una cadena de longitud variable *terminada con la secuencia de caracteres <CRLF>. El receptor no realiza la acción* hasta que no recibe esta secuencia.

1.4.4 Protocolo de Transferencia de Archivos (FTP)

Acceso a archivos

El acceso a archivos compartidos se puede ver desde dos perspectivas: acceso ***en línea***, en tiempo real, o mediante ***copia completa***. Compartir mediante acceso en línea

significa que múltiples programas acceden a un archivo frecuentemente, por lo que los cambios que se efectúan en el archivo se llevan a cabo en el momento y están disponibles para todos los procesos que acceden al archivo. El acceso por copia completa significa que cuando un programa quiere acceder a un archivo realiza una copia del mismo en el sistema local. Este último mecanismo es usado frecuentemente para datos de sólo lectura, pero si el archivo es modificado, el programa realiza el cambio en la copia local y transfiere la modificada al sitio original.

El esquema de transferencia de archivos requiere un proceso de dos pasos: el primero consiste en obtener una copia en el sistema local del archivo La mayoría de los mecanismos de transferencia de archivos operan fuera del sistema de archivos local. El usuario invoca a un programa cliente para que transfiera el archivo, debiendo especificar el sistema remoto dónde está el archivo, llevándose a cabo una autorización de dicha operación El cliente se conecta con el servidor remoto y pide una copia del archivo. Una vez que la transferencia ha sido realizada, el usuario finaliza su conexión como cliente y mediante un programa accede a ese archivo en su sistema local para leerlo y actualizarlo.

Este mecanismo, como ocurre con el de compartición de archivos en línea, puede resultar bastante complejo. El cliente y el servidor deben estar de acuerdo en autorizaciones, propiedad de archivos, protecciones y formato de datos. Este último aspecto es muy importante, ya que si no existe el mismo formato, al realizar la transferencia del archivo, se pueden perder o alterar algunos datos Por ejemplo, consideremos dos máquinas A y B, que emplean diferentes formatos para la representación de números con punto flotante y para archivos de texto Para el primer caso, es imposible llevar a cabo esta conversión del formato de una máquina al de la otra sin perder precisión. Supongamos que la máquina A almacena textos en líneas de longitud variable y el sistema B lo realiza mediante líneas de longitud fija La transferencia de archivos de A a B se puede llevar a cabo rellenando las líneas a esa longitud fija, haciendo que la copia final difiera de la original.

El Protocolo FTP

La transferencia de archivos con el protocolo FTP (*File Transfer Protocol*) es una de las mas empleadas. FTP proporciona facilidades para las funciones de transferencia, como son:

- **Acceso interactivo**: si bien FTP está diseñado para ser usado por programas, la mayoría de las implementaciones proporcionan al usuario una interfaz con servidores remotos para importar o exportar archivos.
- **Especificaciones de formato**: FTP permite al cliente especificar el tipo y el formato de los datos. Por ejemplo, puede especificar que los datos sean ASCII o binarios.
- ***Control de autenticación***. *FTP exige al cliente que se identifique mediante su* nombre de usuario y su contraseña. El servidor puede negarle el acceso en caso de que ese usuario no esté autorizado.

Al igual que otros servidores, la mayoría de las implementaciones de FTP permiten el acceso concurrente de varios clientes. Los clientes emplean TCP para conectarse al servidor. En general, en este tipo de servidores, el proceso maestro del servidor genera un esclavo para atender a cada uno de los clientes. En FTP, el proceso maestro acepta y lleva a cabo las peticiones de conexión del cliente, pero emplea otro proceso para manejar la transferencia de datos.

El proceso cliente se conecta al servidor mediante una conexión TCP, mientras que la transferencia de datos emplea sus propias conexiones TCP. En general, el proceso de control y la conexión de control, permanecen activos mientras el usuario mantenga su sesión FTP abierta. Sin embargo, FTP establece una nueva conexión de transferencia para cada archivo que se vaya a transmitir.

La conexión para transferencia de datos y los procesos de transferencia de datos se crean dinámicamente según se van necesitando, mientras que la conexión de control permanece activa mientras perdure la sesión FTP. Una vez que la conexión de control desaparece, la sesión finaliza y los procesos de ambos extremos finalizan la transferencia de datos.

Asignación de número de puerto TCP

Cuando un cliente se conecta al servidor, el cliente emplea un puerto aleatorio, pero el servidor se conecta en el puerto 21. Cuando el proceso de control crea una nueva conexion TCP para la transferencia de datos, no puede emplear los mismos números de puertos empleados en la conexión de control El cliente obtiene un puerto no usado de su máquina y lo emplea para el proceso de transferencia de datos. El proceso de transferencia de datos en el servidor se lleva a cabo mediante el puerto 20 (puerto reservado para la transferencia de datos).

Los usuarios ven a FTP como un sistema interactivo. Una vez que se ha invocado, el cliente ejecuta una serie de submandatos para efectuar la conexión y la transferencia de archivos Para las conexiones vía Internet a otros *hosts* de la red para la transferencia de archivos, existen dos nombres de usuarios que suelen estar definidos en las máquinas con acceso a Internet y que no necesitan contraseña. Estos nombres son "anonymous" y "ftp"

Por último, cabe señalar que los mensajes de control y de errores se llevan a cabo mediante mensajes de 3 dígitos seguidos de texto, de manera que son perfectamente legibles, como por ejemplo: *150 Modo Binario, 200 Comando exitoso, 226 Transferencia completa*

1.4.5 FTP Trivial (TFTP)

A pesar de que FTP es el protocolo más utilizado para la transferencia de archivos, también se emplea otro protocolo llamado TFTP, ya que muchas aplicaciones no necesitan de todas las funcionalidades que ofrece FTP; por ejemplo, FTP obliga a que el cliente y el servidor empleen múltiples conexiones concurrentes.

TFTP (*Trivial File Transfer Protocol*) es un protocolo para transferir archivos entre distintas máquinas conectadas a través de una red de comunicaciones Se implementa sobre un servicio de comunicaciones no fiable y no orientado a conexión. Consiste fundamentalmente en la lectura o escritura por parte de un cliente de un archivo de un servidor

TFTP fue definido para aplicaciones que no necesitan tanta interacción entre cliente y servidor Está restringido a operaciones de transferencia de archivos en los que no es necesaria una autenticación. Por estas razones, los protocolos TFTP son más sencillos.

La simplicidad es particularmente importante para algunas aplicaciones. Por ejemplo, los diseñadores de estaciones de trabajo sin disco (terminales tontas) pueden incluir el software TFTP en memorias de sólo lectura (ROM). El programa incluido en las memorias

ROM se llama **bootstrap**. La ventaja de TFTP es que permite el arranque de un sistema de manera remota. Así es posible que una computadora arranque desde un servidor

TFTP emplea el puerto 69 que previamente debe estar asignado. Emplea el protocolo UDP en vez de TCP. La corrección de errores se realiza a nivel TFTP, es decir, utiliza un mecanismo de parada y espera para controlar el flujo de información El emisor envía los archivos por bloques (paquetes DATA) de tamaño fijo (512 bytes) y espera la confirmación por parte del receptor. A su vez, el receptor tiene que enviar la conformidad de la recepción de cada bloque de datos

Existe un mecanismo de negociación de opciones donde se incluye una opción para especificar el tamaño del bloque En este caso, el cliente añadirá a su solicitud una opción de nombre "blksize", y con un valor igual al número de bytes propuesto como tamaño de bloque, entre 8 y 65464. Si el servidor acepta dicha opción, enviará una confirmación incluyéndola y con un valor igual o menor al propuesto. El cliente entonces dará por bueno dicho valor, o abortará la transferencia enviando un paquete ERR con código de error 8 Las reglas para terminar la transferencia normalmente son las mismas que en el caso de tamaño 512 bytes: el paquete DATA con menos bytes de los negociados como tamaño de bloque, se considera el último.

Las reglas para TFTP son muy simples. En el envío del primer paquete se establece una interacción entre el cliente y el servidor. Se empieza una numeración de bloques, comenzando por el 1. Cada paquete de datos contiene una cabecera que especifica el bloque que contiene, y por cada confirmación contiene el número de bloque que confirma su recepción. Un bloque de menos de 512 bytes implica que es final del archivo.

Existen plazos de tiempo para la recepción de los paquetes. Si un paquete se pierde en la comunicación, a su destinatario le vencerá un plazo y deberá retransmitir el último paquete transmitido (de datos o de confirmación), lo que causará que el emisor del paquete perdido retransmita dicho paquete. Se utilizan los plazos tanto para el cliente como para el servidor.

Existen tres tipos de situaciones que causan un error:

- cuando no es posible aceptar una solicitud de transferencia (archivo inexistente, violación de permisos)
- cuando se recibe un paquete con formato incorrecto
- cuando se pierde el acceso a un recurso (disco lleno) en mitad de la transferencia

Los errores causarán la terminación de la transferencia. Un error se indicará enviando un paquete de error. Este paquete ni se confirma ni se retransmite, así que el otro extremo de la comunicación puede no recibirlo nunca.

Una vez que se ha efectuado una petición de lectura o escritura, el servidor emplea la dirección IP y el protocolo UDP del cliente para las operaciones siguientes. Ni los mensajes de datos ni los de confirmación de recepción (ACK) necesitan incluir el nombre del archivo

En la figura 1.13 se representa el flujo de transferencia TFTP, donde:

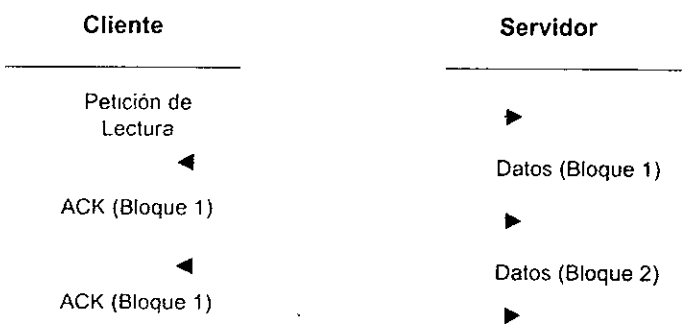


Figura 1.13 - *Transferencia TFTP*

Cada extremo implicado en la conexión implementa un temporizador (o tiempo de plazo) y una retransmisión. Si vence el temporizador del emisor, éste retransmite el último bloque de datos. Si vence el temporizador del receptor, éste retransmite la última confirmación.

A pesar de que la retransmisión simétrica garantiza robustez, puede resultar costosa tal retransmisión El problema puede surgir si un paquete "n", en lugar de perderse, simplemente se demora. El emisor retransmite el paquete de datos mientras el receptor emite la confirmación. Ambas confirmaciones llegan y se transmite el paquete "n+1". El receptor confirma ambas copias del paquete "n+1" y tales confirmaciones hacen que el emisor transmita el paquete "n+2" El ciclo continúa y provoca que cada paquete de datos sea transmitido dos veces.

1.4.6 Sistema de Archivos de Red (NFS)

El Sistema de Archivos de Red (NFS, *Network File System*) ha sido desarrollado por Sun Microsystems Incorporated y autoriza a los usuarios el acceso "en línea" a archivos que se encuentran en sistemas remotos; de esta forma, el usuario accede a un archivo remoto como si éste fuera un archivo local. Desde la perspectiva del usuario, NFS es casi invisible.

Cuando un programa de aplicación se ejecuta, un proceso cliente realiza una llamada al sistema operativo, ya sea para abrir un archivo, o bien para almacenar datos en el archivo. El mecanismo que controla el acceso a los archivos acepta la petición y automáticamente la pasa, dependiendo de si el archivo se encuentra en el disco local o en el sistema remoto, al sistema de archivos local o al cliente NFS Cuando se recibe la petición, el proceso cliente utiliza el protocolo NFS para contactar con el proceso servidor adecuado en la máquina remota y así realizar el servicio requerido. Una vez que el proceso servidor devuelve los resultados correspondientes, el cliente reenvía estos resultados al programa de aplicación, finalizando así la cooperación entre el cliente y el servidor

Para llevar a cabo este servicio, son necesarios dos protocolos·

- El **protocolo MOUNT**, que especifica el host remoto y el sistema de archivos al que se va a acceder.
- El **protocolo NFS**, que realiza las tareas de entrada/salida del archivo remoto y que significa Sistema de Archivos de Red.

Ambos protocolos son aplicaciones de RPC (Llamada a Procedimientos Remotos, vista anteriormente) y utilizan el servicio de transporte UDP.

El protocolo MOUNT

Se trata de un servidor RPC que realiza cinco funciones:

- **NULL**. operación nula. Sirve para que el sistema se autoverifique.
- **MOUNT**: monta una unidad de disco, un directorio o un conjunto de archivos *externos, en el sistema de archivos local del host, de esta forma, el host los puede* tratar como si formaran parte de uno de sus subdirectorios
- **DUMP**. devuelve una lista de los archivos montados en el sistema.
- **EXPORT**: proporciona información sobre los sistemas de archivos que se encuentran disponibles.

A continuación, describimos brevemente un ejemplo de la función MOUNT. El cliente llama al procedimiento MOUNT, especificándole que quiere tomar del servidor un directorio y que quiere tenerlo en su árbol de directorios, como un subdirectorio, con un nuevo nombre. Este mandato lo ejecuta el cliente a través de la interfaz API, que es una aplicación RPC. Como respuesta a la invocación del procedimiento MOUNT, el cliente recibe un bloque de control procedente del servidor que contiene información relativa al servidor en el que se encuentra el archivo requerido, la biblioteca, etc

El protocolo NFS

Una vez que se ejecuta el mandato MOUNT, el protocolo NFS se encarga de realizar *todas las operaciones básicas de entrada/salida de archivos. NFS da soporte a 18* procedimientos que cubren todas estas operaciones básicas. A continuación se exponen algunas de ellas:

- **LOOKUP**: busca un archivo en el directorio actual de trabajo
- **READ/WRITE**. son las dos primitivas básicas de acceso a un archivo para leer o escribir
- **RENAME**: renombra un archivo.
- **REMOVE**: borra un archivo.
- **MKDIR/RMDIR**: creación y borrado de directorios.

Una vez que se monta el directorio remoto en un host, el sistema operativo local del host debe encargarse de reencaminar estas primitivas de entrada/salida al host remoto. De esta forma, el servicio proporcionado por el protocolo NFS resulta totalmente transparente al usuario.

1.4.7 Sistema de Nombre de Dominios (DNS)

El Sistema de Nombre de Dominios (DNS, *Domain Name System*) es una base de datos distribuida, que nos permite, a partir de un nombre de máquina, obtener su dirección IP correspondiente y viceversa. EL DNS está formado por servidores de nombres que contienen la información de los ordenadores que están declarados en esa organizacion y que hace disponible esa información a clientes llamados "resolvers".

La estructura de DNS es jerárquica, existiendo un número de servidores para el dominio raiz que son los responsables de tener la información actualizada de los dominios superiores y de los que hay por debajo de ellos

Un dominio, identifica su posición en la base de datos, al igual que un camino absoluto en un sistema de ficheros UNIX. En DNS un dominio es una secuencia de nombres separados por puntos. En DNS, cada dominio puede estar administrado por una organización, la cuál puede, a su vez, subdividir ese dominio en subdominios y delegar la administración a otras organizaciones.

Es interesante distinguir entre dos conceptos que pueden llevar a confusión: dominio y zona El dominio es el conjunto completo de máquinas que forman parte de una organización, mientras que una zona es el área del DNS para el que es responsable un servidor Por ejemplo, el dominio de la UNAM es "unam.mx", el cuál contiene todas las maquinas que forman parte de la Universidad Nacional Autónoma de México. Mientras nosotros podemos delegar la autoridad de una zona de ese dominio (de la cuál es responsable un servidor) a un servidor en concreto.

También cabe hacer notar que DNS no sólo se utiliza para la resolución de nombres de máquina, sino también para el correo electrónico (mediante unos registros llamados MX)

BIND

La primera realización de DNS recibió el nombre de JEEVES y fue escrita por Paul Mockapetris. Una posterior implementación se llamó BIND y fue escrita para el sistema operativo Unix 4.3BSD por Kevin Dunlap. El Dominio de Nombres de Internet de Berkeley, BIND (*Berkeley Internet Name Domain*) es la implementación más utilizada de DNS en la actualidad, y ha sido realizado para la mayoría de las plataformas UNIX, formando parte del sistema operativo.

Existen versiones de dominio público de BIND (que son las más utilizadas en el ámbito de Internet). El software de BIND está formado por un servidor de DNS, así como por algunas utilidades de resolución de nombres (nslookup, dig).

La última versión de BIND es la 8.1.1. Esta versión ha supuesto un cambio sustancial con respecto a las versiones anteriores (BIND 4.9.x). En esta nueva versión, además de soportar IPv6, se incluyen bastantes novedades en el momento de configurar los servidores de DNS.

1.4.8 Protocolo de Transferencia de Hipertexto (HTTP)

El Protocolo de Transferencia de Hipertexto (HTTP, *Hypertext Transfer Protocol*) es un protocolo de la capa de aplicación con la agilidad y velocidad necesarias para sistemas de información distribuidos y que manejen multimedia (es el conjunto de tipos de información, como son los gráficos, texto, video y sonidos juntos; también se le conoce como hipermedia)

Es un protocolo genérico, orientado a objetos, que puede ser empleado para muchas aplicaciones, como servidores de nombres y sistemas de manejo de objetos distribuidos, a traves de la extensión de los métodos de requerimiento (comandos).

Una caracteristica de HTTP es la escritura de la representación de datos, que permite a los sistemas ser construidos independientemente de los datos que están siendo transmitidos

El protocolo HTTP ha estado en uso en la iniciativa de información global World Wide Web desde 1990. Esta especificación refleja el empleo común de este protocolo conocido como "HTTP/1 0" y describe las características que parecen ser implementadas consistentemente en la mayoría de los servidores y clientes HTTP/1.0.

Los sistemas de información práctica requieren una mayor funcionalidad que la proporcionada por la recuperación simple de datos, incluyendo la búsqueda, la actualización y la marcación o anotación de datos. HTTP proporciona un conjunto abierto de métodos que son empleados para indicar el propósito de un requerimiento. Está basado en la referencia proporcionada por el Identificador Uniforme de Recursos (URI, *Uniform Resource Identifier*), como una localidad (URL, *Uniform Resource Location*) o nombre (URN), para indicar los recursos en los cuales son aplicados estos métodos. Los mensajes son transmitidos en un formato similar al empleado por el correo de Internet y por las Extensiones Multipropósito de Correo de Internet (MIME, *Multipurpose Internet Mail Extensions*)

El protocolo HTTP es usado también como un protocolo genérico para la comunicación entre agentes de usuarios o gateways/proxies y otros protocolos de Internet, tales como el Protocolo Simple de Transferencia de Correo (SMTP, *Simple Mail Transfer Protocol*), el Protocolo de Transferencia de Noticias en la Red (NNTP, *Network News Transfer Protocol*), el Protocolo de Transferencia de Archivos (FTP, *File Transfer Protocol*), Gopher (que es un servicio de acceso a bancos de documentos mediante menús) y WAIS (que es un servicio de búsqueda de información en bancos de datos con opciones, esto es, con números, sin menús). Este protocolo también permite a los recursos básicos multimedia acceder a los recursos disponibles desde diversas aplicaciones y simplificar la implementación de los agentes de usuarios.

1.5 CAPA DE TRANSPORTE

Como ya lo explicamos en el subtema 1.2.2, la capa de transporte tiene como función suministrar a las aplicaciones servicios de comunicaciones de extremo a extremo empleando dos protocolos: el Protocolo de Control de Transferencia (TCP), que es fiable y orientado a conexión, y el Protocolo de Datagrama de Usuario (UDP), el cual es no fiable y no orientado a conexión. Esta capa regula, además, el flujo de información y

coordina a múltiples aplicaciones que se encuentren interactuando con la red simultaneamente.

A continuación trataremos el protocolo TCP y, posteriormente, daremos lugar al protocolo UDP

1.5.1 El Protocolo de Control de Transferencia (TCP)

Características del protocolo TCP

El Protocolo de Control de Transferencia (TCP, *Transfer Control Protocol*) es un protocolo orientado a conexión que utiliza los servicios de la capa de Internet. Al igual que en todo protocolo orientado a conexión, ésta consta de tres fases:

- Establecimiento de la conexión.
- Transferencia de datos.
- Liberación de la conexión

TCP permite la multiplexión, esto es, la capacidad de que una conexión TCP pueda ser utilizada simultáneamente por varios usuarios

La unidad de datos que maneja TCP se denomina segmento y la longitud de un segmento se mide en caracteres (octetos) Los canales de comunicación establecidos mediante TCP son dúplex (aunque el enlace sea semidúplex) y se mantiene la secuencia de entrega de datos transferidos.

La transmisión que ofrece TCP es fiable, es decir, permite la recuperación de datos perdidos, erróneos o duplicados y garantiza la secuencia de entrega, para lo que se asigna al segmento de datos un número de secuencia (información de control) y un *checksum* (código de control). La fiabilidad de la transmisión se consigue mediante tres mecanismos diferentes:

- Confirmación de recepción.
- Temporizadores de espera de confirmación.
- Retransmisión de segmentos.

Para disponer de control de flujo, el receptor mantiene una ventana que indica al emisor la cantidad de datos que puede enviar a partir de cada confirmación recibida.

Puertos

Lo más normal es que, en un momento determinado, haya más de un proceso de usuario o aplicación utilizando TCP simultáneamente. Por ello, es necesario un método que identifique los datos asociados a cada proceso Un puerto es una palabra de 16 bits que identifica hacia qué aplicación o proceso deben dirigirse los datos. Se trata de un mecanismo a través del cual las distintas aplicaciones contactan con TCP/IP.

Sockets o Zócalos

Es un par de números que identifica de manera única cada aplicación. Cada socket se compone de dos campos:

- La dirección IP del host en el que la aplicación se está corriendo.
- El puerto a través del cual la aplicación se comunica con TCP/IP. Este número de puerto identifica el proceso.

Un socket es una interface que permite que las aplicaciones escritas para una especificación se ejecuten sobre distintas implementaciones TCP/IP. U.C. Berkeley Sockets es el estándar de *facto* y fue adoptado en la forma de Windows Sockets (el muy nombrado winsock.dll) en las implementaciones de Windows. Es una Interfaz de Programación de Aplicaciones (API, *Application Program Interface*) para la que se escriben las aplicaciones Hay implementaciones TCP/IP que corren protocolos superiores y aplicaciones del mismo proveedor, sin hacer uso de esta interface. La tendencia es que todo lo que corre sobre TCP/IP se «relaciona» a través de esta API, permitiendo mayor independencia de proveedor y facilidad en el desarrollo de aplicaciones

Algunas de las aplicaciones que realizan la función de servidores normalizados empleando los servicios TCP/IP, son TELNET para conexión remota o FTP para transferencia de archivos. Por ello, en todas las realizaciones TCP/IP, estas aplicaciones tienen siempre asignado el mismo número de puerto, concretamente la aplicación de transferencia de archivos FTP tiene asignado el puerto 21 para control y el puerto 20 para datos y TELNET tiene el 23 Estos puertos reservados se denominan "puertos conocidos" (*well_known ports*) y no deben utilizarlos otras aplicaciones que no sean las previamente asignadas (SMTP, Whois, TFTP, etc.). Los números entre el 1 y el 255 (ambos inclusive) corresponden a puertos conocidos.

En el entorno UNIX estas asignaciones de puertos se encuentran ubicadas en un archivo (/etc/services) en el que se encuentran los puertos bien conocidos, así como los puertos para aplicaciones desarrolladas con TCP/IP (como la denominada *sgenserv*). En este archivo también se especifica el protocolo del nivel de transporte que se va a emplear (TCP o UDP).

El mecanismo de ventanas deslizantes

En su versión más elemental, un protocolo de transporte simple utiliza como control de flujo el mecanismo de "Parada y Espera", en el que se envía un paquete y cuando es recibido correctamente por el receptor, éste envía un ACK. En cambio, cuando se envía un paquete y se pierde, y se vence el temporizador sin recibir respuesta, el emisor lo retransmite

Con este protocolo, el emisor siempre espera un ACK (confirmación) de que el segmento (tratándose de TCP) ha llegado correctamente por parte del receptor. Ahora bien, este mecanismo desaprovecha gran parte del ancho de banda de la red. Por esta razón, el protocolo TCP utiliza un mecanismo de ventanas para controlar el flujo de informacion

La idea del mecanismo de ventana deslizante es que el emisor pueda transmitir tantos paquetes de información sin recibir la confirmación de recepción como tenga en la ventana El rendimiento de este mecanismo depende del tamaño de la ventana y de la velocidad a la que la red transmite los paquetes. Si el tamaño de la ventana es 1, entonces el rendimiento es el mismo que con el mecanismo de parada y espera.

TCP divide las series de caracteres en segmentos y la longitud de cada segmento se mide en caracteres Así pues, se pueden distinguir cuatro áreas en el búfer donde, en la primera, se incluyen los caracteres transmitidos y confirmados; en la segunda, están

incluidos los caracteres enviados y no confirmados; la tercera incluye los caracteres no enviados pero que pueden enviarse sin esperar a recibir la confirmación y, por último, en la cuarta se incluyen los caracteres que no pueden enviarse en ese instante

Formato de los Segmentos TCP

En la figura 1.14 se muestra el formato de los segmentos TCP Cada segmento está dividido en dos partes una cabecera seguida de datos La cabecera TCP contiene la información de identificación y control.

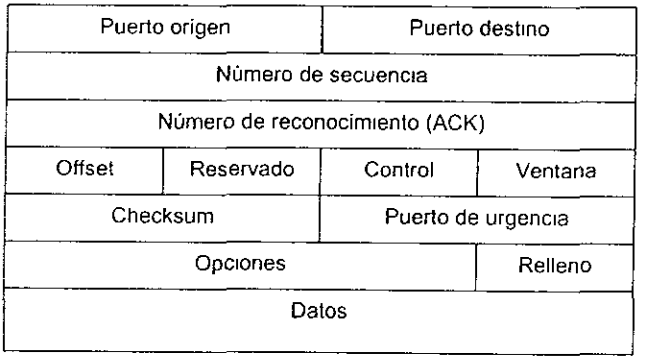


Figura 1 14 - *Formato de los Segmentos TCP*

- **Puerto origen**: Puerto a través del cual una aplicación invoca a TCP. Su tamaño es de 16 bits.
- **Puerto destino**: Puerto de la aplicación en destino. Su tamaño también es de 16 bits.
- **Número de secuencia**: Es el número de secuencia del primer byte de ese segmento que se espera recibir
- **Offset**: Contiene un entero que especifica la longitud de la cabecera del segmento en múltiplos de 32 bits. Su longitud es de 4 bits.
- **Reservados**. Estos 6 bits están reservados para usos futuros
- **Control**: Indica el tipo de segmento. Estos 6 bits indican cómo deben interpretarse algunos campos de la cabecera. Los bits están especificados según el orden en el que se enumeran, de manera que, si el segundo bit tiene el valor de 1, es un segmento de confirmación. La interpretación de cada bit es la siguiente.
  - **URG**: Segmento urgente
  - **ACK**. Segmento de confirmación.
  - **PSH**: En TCP, tanto el emisor como el receptor, disponen de un búfer para almacenar los datos a enviar o a recibir. Cuando el receptor recibe un segmento con el bit de "PUSH" activado, entiende que debe enviar todo lo que tiene almacenado en su búfer al proceso del cual acaba de recibir el segmento de "PUSH".
  - **RST**: Segmento de Reset o reinicio de conexión.
  - **SYN**: Segmento que sincroniza el número de secuencia.
  - **FIN**: Segmento que indica que no hay más datos para el receptor.

- **Ventana** Indica el tamaño de la ventana.
- **Checksum**: Es un campo de 16 bits. Está formado por el complemento a 1 de la suma (en complemento a 1) de todas las palabras que componen el segmento TCP.
- **Puntero urgente**: Aunque TCP está orientado a conexión, a veces es importante enviar datos fuera de banda. Esto puede ocurrir cuando, en una conexión remota, el usuario decide enviar una secuencia de teclado que interrumpa o aborte el programa. Estas señales deben ser enviadas sin esperar a que el programa esté listo para recibir datos

  Para acomodar el ancho de banda a las señales, TCP permite identificar estos datos como urgentes, haciendo que lleguen al destino lo antes posible. El protocolo TCP especifica que, cuando unos datos son urgentes, el programa TCP del receptor debe procesarlos de inmediato y, una vez procesados, volver al modo normal Cuando un bit URG está activo, el puntero urgente especifica en la ventana la posición donde acaban los datos urgentes.
- **Opciones**. Similar al campo de opciones de IP (que se tratará con detalle en el capítulo II) En ellas se especifica el máximo tamaño del segmento
- **Relleno** Es como en IP el campo *Padding* Son bits a cero que se utilizan para rellenar la cabecera TCP, de manera que ésta alcance una longitud total que sea multiplo de 32

Para manejar los datos, la información de los protocolos superiores se encapsula en la información de los protocolos inferiores. Esto es, a los datos se les agrega una cabecera TCP, una cabecera IP y, por último, una cabecera de subred.

1.5.2 El Protocolo de Datagrama de Usuario (UDP)

El Protocolo de Datagrama de Usuario (UDP, *User Datagram Protocol*) es un protocolo de la capa de transporte que se basa en el intercambio de datagramas Como su nombre lo dice, las unidades de información que maneja se llaman datagramas. UDP permite el envio de datagramas a través de la red sin que se haya establecido previamente una conexion (ofrece un servicio no orientado a conexión), para lo que el propio datagrama incorpora la suficiente información de direccionamiento. Esto simplifica notablemente el protocolo, pero, a cambio, no se confirman los datagramas recibidos ni se garantiza su orden, debiendo ser la capa de aplicación la que se encargue de su control.

El protocolo UDP maneja también los conceptos de puertos y sockets, ya que este protocolo es utilizado simultáneamente por varias aplicaciones (al igual que TCP). UDP básicamente proporciona acceso a los servicios de la capa IP, incorporando multiplexión/demultiplexión. No proporciona control de flujo ni fiabilidad en las transmisiones o recuperación de algunos tipos de errores. Funciona como multiplexor/demultiplexor en el envío y recepción de datagramas IP a través de los puertos.

Formato de los datagramas UDP

El formato de los datagramas UDP puede observarse en la figura 1.15, donde:

- **Puerto origen**· Puerto del proceso emisor u origen (a este puerto deben dirigirse las respuestas requeridas) y que tiene una longitud de 16 bytes
- **Puerto destino**. Especifica el puerto del proceso destino (en el host destino) y ocupa los 16 bytes restantes de la primera palabra.
- **Longitud** Tiene 16 bytes y es la longitud en bytes del datagrama UDP (incluida la cabecera)
- **Checksum** Es el complemento a 1 de la suma (en complemento a 1) de todos los bits que forman el datagrama UDP, más unos bits adicionales constituidos a partir de la cabecera IP Tiene una longitud de 16 bytes.

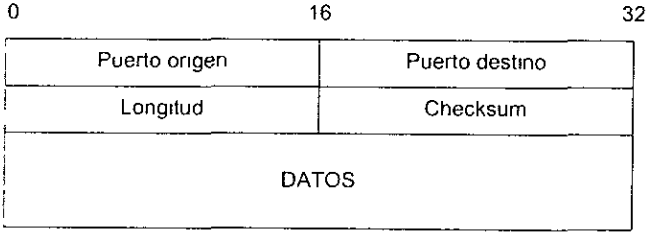


Figura 1.15 - *Formato de los Datagramas UDP*

Multiplexión, Demultiplexión y Puertos

El software de UDP acepta datagramas UDP de múltiples programas de aplicación y los pasa a la capa IP para su transmisión, a la vez que acepta datagramas de IP y se los pasa a los correspondientes programas de aplicación.

Conceptualmente, toda la multiplexión y demultiplexión entre el software de UDP y los programas de aplicación se realiza mediante puertos. En la práctica, cada programa de aplicación debe negociar con el sistema operativo para obtener un puerto de protocolo y un número de puerto antes de que pueda enviar datagramas UDP. Una vez que el puerto ha sido asignado, cualquier datagrama que envíe la aplicación pondrá ese número en el Numero de Puerto UDP

Mientras se procesa la entrada, UDP acepta datagramas de software IP y los demultiplexa dependiendo del puerto destino UDP. En la figura 1.16 se puede ver la demultiplexión (o multiplexión en el caso de emisión) de datagramas por parte de UDP.

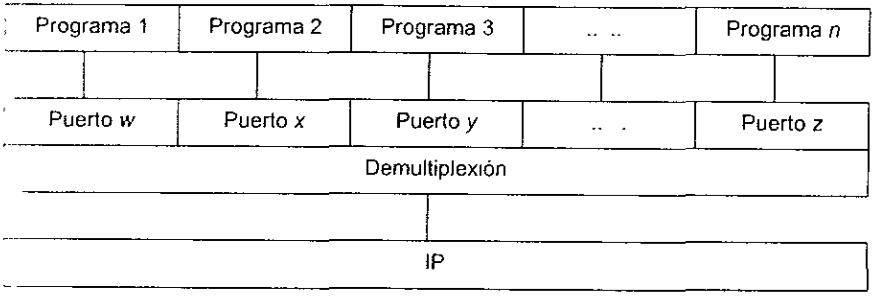


Figura 1.16 - *Demultiplexión de Datagramas UDP*

Por último, cabe hacer mención que este protocolo suele ser usado para tipos de direcciones omnidireccionales (multicast) y bidireccionales (broadcast), ya que no asegura la entrega de paquetes, ni su secuencia (semejante a IP). Es adecuado para redes de alta fiabilidad (física) sin requerir un alto sobre encabezado (overhead), ahorrando ancho de banda

1.6 CAPA DE INTERNET

Como ya lo explicamos anteriormente, esta capa controla la comunicación entre un equipo y otro, decide qué rutas deben seguir los paquetes de información para alcanzar su destino y conforma los paquetes IP que serán enviados por la capa inferior. Desencapsula los paquetes recibidos, pasando a la capa superior la información dirigida a una aplicación.

La capa de Internet crea un servicio de red virtual, no es fiable ni orientada a conexión. Se esfuerza en entregar los paquetes, denominados datagramas, a su destino. Los datagramas pueden perderse, duplicarse o cambiar de orden de secuencia

Los protocolos más importantes de esta capa son los siguientes

- Protocolo IP (Protocolo Internet)
- Protocolo ICMP (Protocolo de Mensajes de Control Internet)
- Protocolo ARP (Protocolo de Resolución de Direcciones).
- Protocolo RARP (Protocolo Inverso de Resolución de Direcciones).

1.6.1 El Protocolo IP

La característica principal es que no es fiable ni orientado a conexión, lo que significa que:

- No garantiza el control de flujo.
- No garantiza la recuperación de errores.
- No garantiza que los datos lleguen a su destino.

El protocolo IP define un encabezamiento que sólo sirve para verificar la integridad de sí mismo y se utiliza para dar soporte a protocolos superiores de la familia TCP/IP. El protocolo IP siempre trabaja con entregas de datagramas (sin conexión previa) que viajan de extremo a extremo de la red El estándar no define capacidad para verificar la integridad del contenido de los paquetes ni tampoco el orden de entrega de los mismos. Los datagramas enviados por IP pueden perderse, llegar desordenados o duplicados. IP no se responsabiliza de estas situaciones que tendrán que ser contempladas por el nivel superior (TCP). No obstante, la red siempre intenta que los datagramas alcancen su destino.

Este protocolo se encarga de seleccionar la trayectoria a seguir por los datagramas, es decir, por dónde se deben encaminar los datagramas salientes, pudiendo realizar las operaciones de fragmentación y reensamblado. El Protocolo Internet posee un muy bajo sobre encabezado (overhead) y es especialmente apto para el ruteo y, por ende, para Redes de Area Amplia (WAN)

El mecanismo de direcciones IP

Cada host posee una dirección IP, que es la encargada de identificar la red y el host. Cada dispositivo en la red debe poseer una dirección IP única e irrepetible. Las direcciones IP son direcciones de 32 bits de longitud, es decir, se compone de cuatro bytes y se representa con cuatro números decimales separados por puntos dentro del rango de números representables por cada uno de esos bytes Existen cinco tipos de formatos diferentes para las direcciones IP que las dividen en cinco clases Estas direcciones se clasifican como A, B, C, D y E. D es para Multicast (un tipo de direccionamiento) y E es reservada, pero, por ser usuarios comunes, sólo nos ocuparemos de las primeras tres, y las veremos en el segundo capítulo.

Cada dirección IP consta de dos direcciones lógicas.

Dirección IP = *<Dirección de la Red> <Dirección del host>*
Ejemplo. 132.208.159.041
dirección de la red: 132 248 dirección del host: 159 041

Hay que tener en cuenta que el número de hosts en realidad debe ser repartido en la cantidad de dispositivos que requieran su propia dirección IP. Esto incluye ruteadores y, por ejemplo, múltiples placas de red de una misma computadora

Se puede diferenciar a qué tipo pertenece una dirección IP por el número que define el primer byte. Una dirección IP identifica una red y dentro de esa red al dispositivo específico y que, por ende, estas direcciones no pueden estar repetidas. Por esto, las direcciones visibles en Internet están reguladas por un organismo específico a tales fines: el Centro de Información de Redes, NIC (Network Information Center)

Por ejemplo, las siguientes dos redes132.168 10.1 y 132.168 11.200 pertenecen a la misma red B; en cambio, una red con una dirección132.167.10.1 no pertenece a la red de las dos primeras aunque sea una red de la misma clase. A priori los dos primeros dispositivos se ven y no ven al tercero sin un gateway en el medio

En algunos sistemas también puede identificarse la subred en la que está ubicado el host, como en el ejemplo anterior:

Dirección IP = *<Dirección de la Red> <Dirección de la Subred> <Dirección del host>*

Esta segunda forma de direccionamiento surge como consecuencia del enorme crecimiento de Internet. El hecho de asignar direcciones IP a los host llegó a ser demasiado inflexible en el momento de realizar pequeños cambios en las configuraciones de las redes locales que estaban conectadas a Internet. Estos cambios se debían principalmente a que el número de hosts que estaban conectados a una red llegaba a ser muy grande y había que realizar una división de la red en dos o más redes de menor tamaño Debido a esto, surgió el término *Subred*, al particionar la red lógica en redes

menores No obstante, la subred tiene existencia propia dentro de la red original, pero no respecto al mundo exterior, que ve una única red, Internet.

La utilización de subredes se realiza mediante la división del/los byte/s pertenecientes al campo host, para identificar a una subred y a un host dentro de la subred mediante la aplicacion de una «máscara de subred» La mayoría de los que emplean el protocolo TCP/IP deberían optar por no usar subredes (esto se logra ubicando 255 en cada campo de red y 0 en cada campo host), salvo que lo necesiten específicamente.

Las subredes, si bien agregan una carga administrativa adicional, permiten segmentar una única red IP para dar servicio a varios segmentos de redes, algo muy útil en el caso de recibir una dirección IP específica de Internet También permite simplificar las tablas de enrutamiento, ahorrando memoria y velocidad de procesamiento de direcciones. A diferencia de la clase de red que puede ser deducida, una dirección IP por sí misma no indica si se está usando el esquema de subredes o no.

Para la aplicación de máscaras de subred, es necesario realizar conversiones binario-decimales Veamos el siguiente ejemplo: para una dirección IP = 200.x.x.182 y Máscara de Subred = 255.255.255.248 El enmascaramiento funciona poniendo en uno los bits que queremos que definan la subred y en cero los que deseamos reservar para dirección de host

Si ahora vemos las direcciones del ejemplo anterior, 132.168 10 1 y 132 168.11.200, al usar una máscara 255 255 255.0, estas direcciones están en distintas subredes y no se ven en forma directa. Ahora 132.168.0.0 es la red y.132.168.10.0 y 132.168.11.0 son distintas subredes En este caso, todos los bits del primer byte del campo Host, están reservados para definir subredes

Es muy importante comprender que las máscaras no deben diferir nunca en hosts en una misma red física ya que esto produciría errores por diferencia en la interpretación de las direcciones.

Los datagramas IP

Los datagramas IP contienen una cabecera con información para el nivel IP y datos. Los datagramas se encapsulan en tramas que, dependiendo de la red física utilizada, tienen una longitud determinada. Por ejemplo, en Ethernet, la longitud máxima es de 1500 bytes.

El formato de la cabecera se puede observar en la figura 1.17

Version	IHL	TOS	Total lenght
Identification		Flags	Offset
TTL	Protocol	Checksum	
Source IP Address			
Destination IP Address			
Options			Padding
DATA			

Figura 1.17 - *Cabecera de Datagrama IP*

donde, los campos que componen a la cabecera IP son los siguientes:

- **Version**: Es la versión del protocolo IP. La versión actual es la 4.
- **IHL** Longitud de cabecera IP en palabras de 32 bits.
- **TOS** Tipo de Servicio (*Type of Service*). Indica las prioridades deseadas. Está compuesto por dos subcampos
  - Los tres bits más altos indican la prioridad
  - Los cinco bits siguientes indican el tipo de servicio Normalmente no se emplean, pero algunas aplicaciones, como el control de encaminamiento y los algoritmos de colas en las pasarelas, utilizan este campo
- **Total lenght** Longitud total del datagrama (cabecera y datos) expresada en bytes.
- **Identification**. A todos los fragmentos en que se puede dividir un datagrama se les asigna el mismo identificador Contiene un entero que identifica el datagrama. Cuando se produce una fragmentación de un datagrama, el campo del identificador se copia en todos los datagramas ya fraccionados. De esta manera, el receptor puede identificar los datagramas que componen el datagrama fragmentado.
- **Flags** Son identificadores de control:
  - Reservado.
  - Se permite / no fragmentación.
  - Último fragmento / más fragmentos
- **Fragment Offset**. Se utiliza en el reensamblaje de los datagramas previamente segmentados. Especifica la posición (offset) en bytes de cada fragmento del datagrama original. El campo de offset se va incrementando en cada fragmento del datagrama que se envía empezando con cero.
- **TTL** Tiempo de vida del datagrama (*Time to Live*). Especifica en segundos el tiempo que puede viajar por una red un datagrama. El tiempo de vida está limitado a 255 segundos. Cada vez que un datagrama pasa a través de un ruteador, resta de este campo el tiempo que tarda en procesar el datagrama (1 como mínimo, aunque el tiempo de proceso sea menor). Cuando este campo alcanza el valor cero antes de alcanzar su destino, se supone que el datagrama está perdido en un bucle cerrado y se descarta
- **Protocol**: Indica el protocolo de nivel superior para el cual el nivel IP está realizando el servicio de transporte de datos en el datagrama. Especifica el formato del área de datos Como ejemplos de protocolos superiores están los siguientes:
  - ICMP
  - TCP
  - UDP
  - IGP
- **Checksum**: Son unos bytes de verificación que afectan a la cabecera y no a los datos El *checksum* se calcula como el complemento a uno de la suma (en complemento a uno) de todos los bits que componen la cabecera. Normalmente, hay que recalcular el checksum de cada nodo por el que pasa el datagrama ya que, al ir atravesando los diferentes gateways, el campo TTL (tiempo de vida) va variando.
- **Source IP Address**: Dirección IP del host origen.
- **Destination IP Address**: Dirección IP del host destino.
- **Options**: Una implementación IP no está obligada a generar diversas opciones para los datagramas que ella misma crea, pero lo que sí debe hacer, es procesar los datagramas que la contengan. Ejemplos de opciones son.
  - *Opción de seguridad*: utilizada por aplicaciones seguras

- *Opción de ruta prefijada*: en el campo Options se especifica una lista de direcciones Internet que componen el camino que deberá seguir el datagrama.
- *Opción de registrar la ruta*: el host fuente crea una lista vacía de direcciones Internet en el campo Options y cada máquina que manipula el datagrama, ha de grabar su dirección en esta lista
- *Opción de registrar la hora*: es similar a la opción anterior Cada máquina graba la hora en la que manipuló el datagrama y, opcionalmente, graba también su dirección

- **Padding** Son bits de relleno. Cuando se utilizan opciones en el campo Options, los datagramas se rellenan con bits a cero, para ajuste a frontera de 4 octetos
- **Data** Son los datos contenidos en el datagrama que pasan al protocolo superior indicado en el campo Protocol. Por definición, el tamaño máximo de un datagrama IP es de 65535 bytes y, suponiendo que la longitud de la cabecera es de 20 bytes, quedan 65515 bytes para datos

1.6.2 El Protocolo ICMP

El protocolo IP se utiliza para poder encontrar una ruta a través de la cual los datagramas viajen por la red y alcancen su destino. En ocasiones, el host destino y los ruteadores necesitan comunicarse con el host fuente, por ejemplo, para que les informe de los errores encontrados al procesar los datagramas. Para esta función y para otras de información de errores o de control, se utiliza el Protocolo de Mensajes de Control de Internet (ICMP, *Internet Control Message Protocol*) ICMP se emplea sólo para los fines expuestos, pero no para hacer fiable el protocolo IP; es un protocolo de mantenimiento de IP y no esta relacionado con el transporte de datos en sí mismo Los datagramas pueden no ser entregados sin ningún tipo de confirmación La fiabilidad debe ser proporcionada por los niveles superiores que utilicen IP.

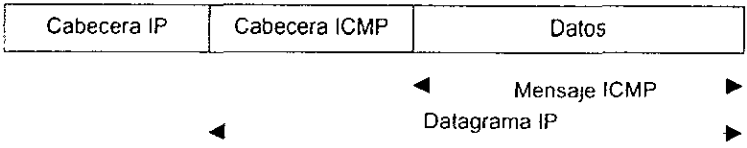
El objetivo principal de este protocolo es proporcionar la información de estados y errores entre dos nodos IP. La implementación del protocolo ICMP es obligatoria como un subconjunto lógico del protocolo IP.

Los mensajes de este protocolo normalmente los genera y los procesa el software TCP/IP de la red y no el usuario. Por ello, no es necesario ningún número de puerto en la cabecera del mensaje ICMP para indicar hacia dónde se dirigen los mensajes

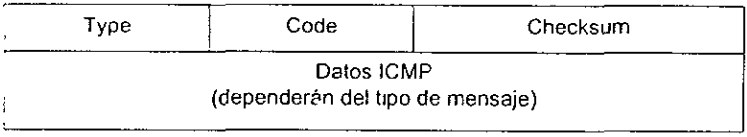
ICPM es utilizado, por ejemplo, en PING, que es una utilidad de diagnóstico de Unix y que permite ver si una dirección IP está viva o visible, es decir, si una computadora está conectada a la red.

Los mensajes ICMP

Estos mensajes se envían encapsulados en datagramas IP. El protocolo IP considera como datos los mensajes ICMP Como ya lo vimos en su momento, los datagramas IP se componen de una cabecera IP y del campo de Datos que, en este caso, consta de una cabecera ICMP y de un campo de datos, como se muestra en la figura 1.18.



A) *Encapsulamiento del mensaje ICMP*



B) *Formato del mensaje ICMP*

Figura 1.18 - *Estructura del mensaje ICMP*

donde los campos del formato son los siguientes.

- **Type** Este campo está formado por 8 bytes y especifica el tipo de mensaje Los tipos se expondrán en el siguiente apartado.
- **Code** Tiene una longitud de 8 bytes. Contiene el código del error que afecta al datagrama al que se refiere el mensaje IP.
- **Checksum**. Consta de 16 bits que son el resultado de realizar el complemento a uno del resultado obtenido al efectuar la suma en complementos a uno del mensaje ICMP, comenzando desde el campo "Type".
- **Datos ICMP**· Normalmente, el campo de datos contiene una parte del mensaje IP original (mensaje causante de generar el mensaje ICMP).

Tipos de Mensaje ICMP

Existen los siguientes tipos de mensaje, que se denominan **Mensajes de error**, a excepción del último, que es de control:

- ***Mensajes de Destino no alcanzable***. Estos mensajes los utilizan los ruteadores y, en algunas ocasiones, los hosts destino
  En un ruteador, este mensaje se envía (de acuerdo con las tablas de ruteo del ruteador) cuando la distancia a la red es infinita o cuando el ruteador no puede enviar el datagrama a la red destino por cualquier motivo.
  Un host envía este mensaje si el protocolo de nivel superior o el puerto especificado no se encuentran en estado activo en el host.
- ***Mensajes de Control de congestión***: Cuando un host destino tiene los búfers llenos, envía este mensaje al host origen, indicándole este suceso. Este envío suele producirse antes de que el búfer se llene al 80% para que el emisor ralentice los mensajes

- ***Mensajes de Redireccionamiento*** Estos mensajes los envían los ruteadores al host emisor. Indican si el datagrama IP se enviará a través de otro ruteador diferente. La nueva ruta será más óptima
- ***Mensaje de Tiempo Excedido***: Es el mensaje que se envían los ruteadores cuando el campo TTL del datagrama IP es cero, o si el temporizador de reensamblado expira antes de que se hayan recibido todos los fragmentos del datagrama inicial

  Existen determinadas circunstancias en las cuales no deben generarse esta clase de mensajes
  - Para responder a otro mensaje de error ICMP.
  - Para responder a un datagrama IP cuya dirección de destino es una dirección IP de difusión (broadcast).
- ***Mensajes de Petición/Respuesta de Eco***: Son mensajes que utilizan los hosts para comprobar que el enlace funciona correctamente (mensaje que envía la aplicación PING de Unix).

1.6.3 El Protocolo ARP

El Protocolo de Resolución de Direcciones (ARP, *Address Resolution Protocol*) es un protocolo que se utiliza para convertir las direcciones IP en direcciones de la red física (por ejemplo, direcciones MAC). El protocolo ARP, al igual que ICMP, es un protocolo de mantenimiento de IP y no está relacionado con el transporte de datos. ARP es utilizado para conversiones de direcciones IP/ MAC, ya que toda dirección IP debe estar asignada a una dirección de hardware específica. Las especificaciones de ARP están descritas en el RFC 826

Los protocolos TCP/IP direccionan los diferentes hosts de Internet mediante direcciones IP, pero, al intentar enviar un datagrama a su destino (por ejemplo, 132 248 10 21), es necesario encontrar la dirección de la red física. Por ello, en cada host existe un módulo ARP cuya misión es convertir las direcciones IP en direcciones físicas que puedan entender los manejadores. Para poder realizar esta conversión, este módulo utiliza una tabla, denominada **Tabla de Direcciones ARP**, que la mayoría de los sistemas trata como si fuera una memoria intermedia (memoria caché), de manera que la información que lleva mucho tiempo sin utilizarse, se borra La duración de cada entrada en la tabla de direcciones va de 1 a 30 minutos y dependerá de la implementación (ver RFC 1122) Además, el emisor de la petición ARP incluye la pareja correspondiente al mapeo de su propia dirección IP y dirección física, el receptor a quien va dirigida dicha petición, actualiza su tabla con esta información antes de enviar una respuesta.

Una máquina que utiliza el protocolo ARP, mantiene en memoria caché una lista de parejas (dirección IP, dirección física) de las direcciones recién adquiridas. Cuando recibe una respuesta ARP, actualiza su tabla de direcciones. Cuando se envía un datagrama IP a un host destino, el módulo ARP busca en la Tabla de Direcciones la correspondencia entre la dirección IP y la dirección física. Si existe la entrada en la tabla, se procede a la transmisión y el protocolo ARP no es efectuado.

Si, por el contrario, la dirección IP del host destino no se encuentra en la tabla de direcciones, se genera una petición ARP que se difunde a través de toda la red. El paquete que engloba esta petición se compone de los siguientes 3 campos:

- Dirección IP del host Origen
- Dirección IP del host Destino
- Dirección Física del host Origen

Si alguna de las máquinas de la red reconoce su propia dirección IP en el paquete de petición envía un mensaje de respuesta al host origen. A su vez, la respuesta se compone de los 2 campos siguientes:

- Dirección IP del host Destino
- Dirección física del host Destino

donde la dirección física del host destino se introduce en la tabla de direcciones del host origen.

En la figura 1.19 puede observarse un ejemplo de cómo el host A quiere resolver la dirección IP de D, es decir, conocer su dirección física. Primero consulta la dirección IP de D y después, A difunde a través de la red (efectúa un broadcast) una petición ARP para pedir al host con la dirección consultada, que responda dando a conocer su dirección física. Todos los hosts, incluyendo a D, reciben la petición. D reconoce su dirección IP y envía una respuesta (respuesta ARP) conteniendo su dirección física. Cuando A recibe la respuesta, utiliza la dirección física para comunicarse directamente con D. En este ejemplo, las computadoras están conectadas a una Ethernet con dirección IP de subred: 131.3.1.

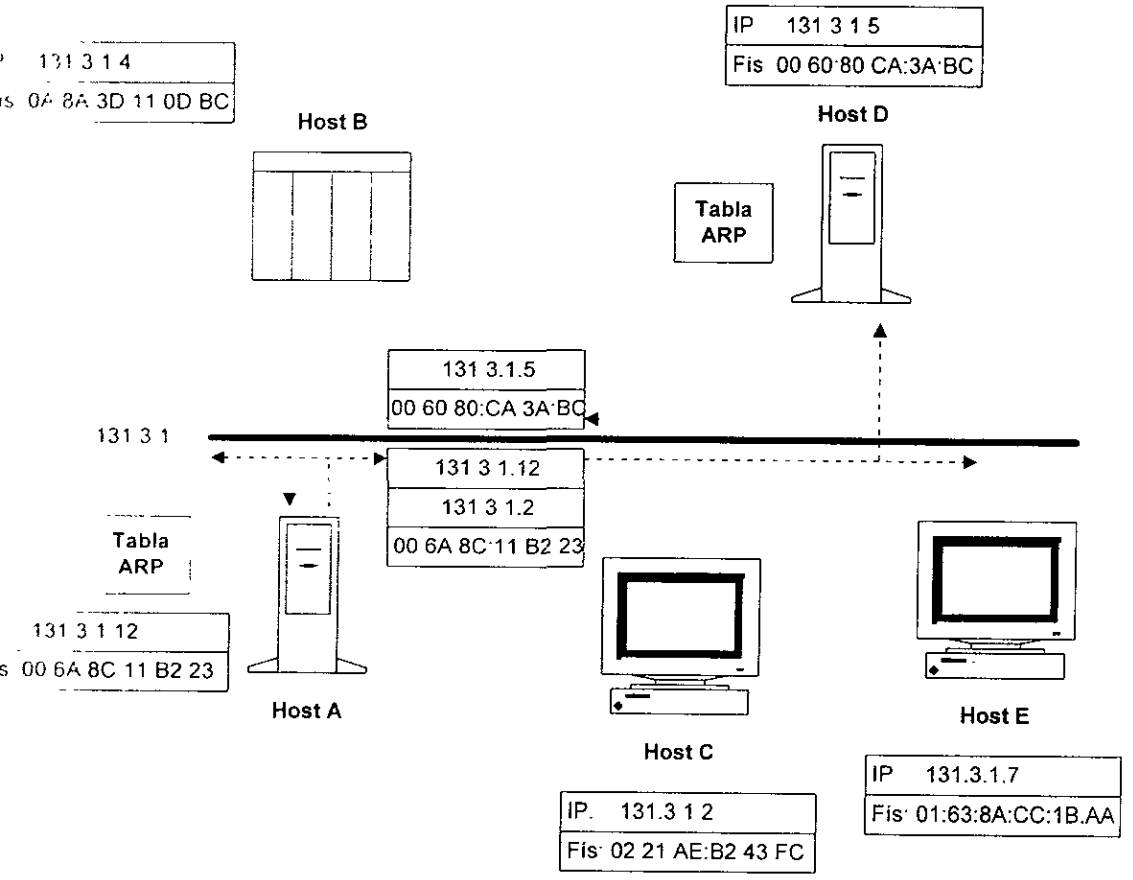


Figura 1.19 - *Petición/respuesta de mensajes ARP*

1.6.4 El Protocolo RARP

El Protocolo Inverso de Resolución de Direcciones (RARP, *Reverse Address Resolution Protocol*) se utiliza cuando, al producirse un arranque inicial, los hosts no conocen su dirección IP. Es un mecanismo similar al mecanismo ARP, con la diferencia de que la dirección física de cada host, en el caso de RARP, es un parámetro conocido, mientras que la dirección IP es desconocida.

El protocolo RARP, al igual que el protocolo ARP, se emplea en redes de difusión (broadcast) Requiere que existan en la red uno o varios servidores RARP. El formato de

los paquetes en este protocolo es el mismo que en ARP Cuando un host desea conocer su dirección IP envía un paquete, difundiéndolo por la red, que contiene su propia dirección física. Todas las máquinas de la red reciben la demanda. Los servidores de RARP, al recibir esta información, buscan en la tabla RARP la dirección de red (dirección IP) correspondiente a la dirección física inicial indicada en el paquete y, al encontrarla, envían un paquete al host origen con esta información. Esto quiere decir que únicamente las máquinas autorizadas para atender mensajes RARP envían una respuesta. La red debe contar con, por lo menos, un servidor RARP La máquina fuente recibe respuesta de todos los servidores RARP, sólo el primero es tomado en cuenta

Así como el protocolo ARP se incorpora normalmente en los productos TCP/IP, al protocolo RARP sólo se incorpora en un número reducido de productos Se utiliza algún tipo de terminales X, si bien su uso es cada vez menos frecuente.

1.7 CAPA DE ACCESO AL MEDIO

Esta capa es la más baja de la jerarquía de protocolos TCP/IP También es conocida como Interfaz de Red y proporciona los medios para que el sistema emita al medio físico los flujos de bits y reciba los que de él provengan Consiste en los manejadores de los dispositivos que se conectan al medio de transmisión. Esta capa define cómo usar la red para transmitir un datagrama IP a otros dispositivos conectados directamente a la red.

A diferencia de los protocolos de alto nivel, los protocolos de la Capa de Acceso al Medio deben conocer los detalles de la red primaria (la estructura de sus paquetes, el direccionamiento, etc), para dar formato correctamente a los datos que serán transmitidos y cumplir con los requerimientos de la red

La Capa de Acceso al Medio es generalmente ignorada por los usuarios. El diseño de TCP/IP oculta la función de la capa más baja y, los protocolos mejor conocidos (IP, TCP, UDP, etc), son todos de alto nivel

Las funciones desempeñadas en esta capa incluyen el encapsulamiento de datagramas IP en tramas que son transmitidas a través de la red y la conversión de direcciones IP a direcciones físicas usadas por la red. Una de las características más robustas de TCP/IP es su esquema de direccionamiento universal Como ya mencionamos, una dirección IP debe ser convertida en una dirección apropiada para la red física sobre la cual serán transmitidos los datagramas.

Esta capa está formada típicamente por módems para líneas analógicas y en Unidades de Servicio de Datos (DSU, *Data Service Unit*) para líneas digitales. El módem o la DSU están conectados a un medio físico implementado por la compañía telefónica, como cable de par trenzado, por ejemplo.

La conexión entre el dispositivo de usuario y el módem o DSU, generalmente tiene lugar a través de un conector EIA 232-E y uno de los estándares apropiados de módems de la serie V (V 42 o V.90, por ejemplo) de la ITU.

Como se ve, es la interfaz con la red real, física y puede o no proporcionar fiabilidad en la distribución de datos, los cuales pueden adoptar diferentes formatos. Debido a que

cada dia aparecen nuevas tecnologías de hardware, los protocolos de acceso a la red deben ser desarrollados de manera que las redes basadas en TCP/IP puedan usarlas. Consecuentemente, TCP/IP no especifica ningún protocolo en esta capa, es decir, existen muchos protocolos de acceso (uno por cada estándar de red), lo que manifiesta la flexibilidad de la capa Internet. Como ejemplos de esta interfaz tenemos la norma IEEE 802 2 (para redes de área local), X.25, Frame Relay o inclusive SNA.

TCP/IP presupone independencia del medio físico de comunicación. Sin embargo, existen estándares bien definidos a los niveles de Enlace de Datos y Físico que proveen mecanismos de acceso a los diferentes medios y que en el modelo TCP/IP deben considerarse en la capa de Interfaz de Red; siendo los más usuales el proyecto IEEE.802, Ethernet, Token Ring y FDDI.

La figura 1 20 muestra la forma de un conector EIA 232-E de 25 terminales y el intercambio de señales que se realiza entre sus terminales cuando se comunica con otro conector, en el extremo final del canal de comunicación. Se señala el número de terminal y el tipo de señal que maneja cada una de ellas; los acrónimos son los siguientes: TxD - Transmisión de datos; RxD - Recepción de datos, CTS - Borrar para enviar; RTS - Peticion para enviar; DSR - Serie de datos listos; CD - Detección de la señal de línea recibida, DTR - Terminal de datos lista, RI - Indicador de anillo; Ground - Señal de referencia (tierra). DTE - Equipo de terminal de datos.

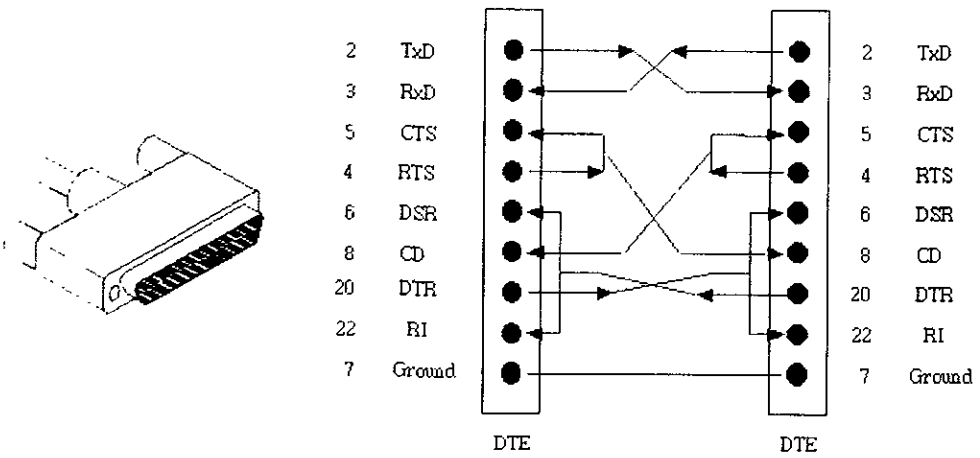


Figura 1.20 - *Conector EIA 232-E y comunicación entre sus terminales*

Capítulo II

EL PROTOCOLO IPv4

2.1 EL DATAGRAMA IP

2.1.1 Introducción

El Protocolo Internet fue desarrollado para "proporcionar las funciones necesarias para entregar un bloque de bits (un datagrama Internet), desde una fuente hasta un destino sobre un sistema interconectado de redes". IP ha proporcionado fielmente esta función durante al menos dos décadas IP se relaciona, principalmente, con la entrega de datagramas, sin embargo, hay que hacer notar que IP no proporciona un servicio de distribución de paquetes de información confiable extremo a extremo, ni una entrega secuencial Esto es, el servicio es orientado a no conexión, lo que significa que los paquetes de información, que serán emitidos a la red, son tratados independientemente, pudiendo viajar por diferentes trayectorias para llegar a su destino. La no fiabilidad se refiere a que no se garantiza la recepción del paquete. IP confía estas características a la capa superior y a las implementaciones del Protocolo de Control de Transmisión (TCP) y al Protocolo de Datagrama de Usuario (UDP) que residen allí.

El termino *datagrama* se refiere a la unidad de información intercambiada por IP, que es un bloque de datos transmitidos sobre una red sin conexión *Sin conexión* significa que no existe una conexión previa entre el origen y el destino, para la transmisión de datos. Tomando como analogía los marcos intercambiados por una red física, los datagramas contienen un encabezado y una área de datos. IP no especifica el contenido del área de datos, ésta será utilizada arbitrariamente por el protocolo de transporte

De forma general, las características del protocolo IP, son las siguientes:

- Protocolo orientado a no conexión.
- Fragmenta paquetes si es necesario.
- Direccionamiento mediante direcciones lógicas IP de 32 bits
- Si un paquete no es recibido, este permanecerá en la red durante un tiempo finito.
- Realiza el "mejor esfuerzo" para la distribución de paquetes
- Tamaño máximo del paquete de 65635 bytes.
- Sólo se realiza verificación por suma al encabezado del paquete, no a los datos que éste que contiene

Otro tipo de transmisión de datos es la conexión por circuito virtual, la cual emplea una red orientada a conexión. Un circuito virtual es análogo a una llamada telefónica, donde la dirección destino es contactada y una ruta es definida a través de la red, previamente a la transmisión de datos. IP es un ejemplo de protocolo basado en datagrama, TCP de un protocolo basado en circuito virtual.

En el proceso de entrega de datagramas, IP debe emplear el direccionamiento y la fragmentación El direccionamiento asegura que el datagrama llegue al destino correcto, ya sea dentro de una misma red, o bien, que cruce todo el mundo. La fragmentación es necesaria porque las redes LAN y WAN, que cualquier datagrama puede atravesar, podrían manejar diferentes tamaños de tramas, y el datagrama IP debe siempre adecuarse a ellas, como se muestra en la figura 2.1.

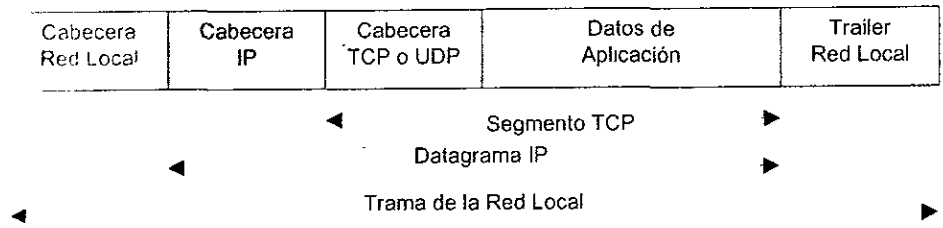


Figura 2.1 - *Trama de Transmisión con IPv4*

Algunos campos específicos dentro de la cabecera IP, que veremos a continuación, se refieren a las funciones de direccionamiento y de fragmentación. En la figura 2.2 se puede ver que cada grupo horizontal de bits (llamado "palabra") tiene una anchura de 32 bits. Como una nota histórica, las palabras de 32 bits fueron empleadas con IPv4 porque los procesadores originales que implementaban este protocolo, manejaban palabras con una longitud de 32 bits.

2.1.2 La Cabecera IP

Los datagramas IP contienen una cabecera con información para el nivel IP y datos. Los datagramas se encapsulan en tramas que, dependiendo de la red física utilizada, tienen una longitud determinada. Por ejemplo, en Ethernet, la longitud máxima es de 1500 bytes, mientras que en FDDI es de 4470 bytes.

El formato de la cabecera se puede observar en la figura 2.2, donde:

Version	IHL	TOS	Total lenght
Identification		Flags	Offset
TTL	Protocol	Header Checksum	
Source IP Address			
Destination IP Address			
Options			Padding
DATA			

Figura 2.2 - *Cabecera de Datagrama IP*

La cabecera IP contiene un mínimo de 20 bytes con información de control. Los campos que contiene la cabecera IP son los siguientes

- **Version** Consta de 4 bits y define la versión del protocolo IP La versión actual es la 4

- **IHL** Longitud de Cabecera Internet (*Internet Header Lenght*). Está formada por 4 bits y mide la longitud de la cabecera IP en palabras (bloques) de 32 bits de longitud. Este campo también proporciona una medición de dónde empieza la información de la capa superior, tal como la cabecera TCP, dentro del datagrama

- **TOS** Tipo de Servicio (*Type of Service*). Consta de 8 bits e Indica las prioridades deseadas, es decir, la calidad del servicio requerido por el datagrama. Está compuesto *por dos subcampos:*
  - Los tres bits más altos indican la prioridad
  - Los cinco bits siguientes indican el tipo de servicio Normalmente no se emplean, pero algunas aplicaciones, como el control de encaminamiento y los algoritmos de colas en las pasarelas, utilizan este campo.

  Los valores son los siguientes:

Bits 0 - 2:	Precedencia (o importancia relativa de este datagrama) 111 - Control de Red 110 - Control Interno de Red 101 - CRITIC / ECP 100 - Flash Override 011 - Flash 010 - Inmediato 001 - Prioridad 000 - Rutina
Bit 3:	Retardo. 0 = Retardo normal, 1 = Retardo bajo
Bit 4	Eficiencia. 0 = Eficiencia normal, 1 = Eficiencia alta
Bit 5	Confiabilidad. 0 = Confiabilidad normal, 1 = Confiabilidad alta
Bits 6 - 7	*Reservados para uso futuro (valor de 0)*

- **Total lenght**: Mide la longitud total del datagrama (cabecera, datos e información de la capa superior), expresada en bytes. Este campo tiene 16 bits y permite el manejo de datagramas de hasta 65,535 bytes, aunque todos los hosts deben ser capaces de manejar datagramas de al menos 576 bytes.

La siguiente palabra de 32 bits contiene tres campos que se refieren a la fragmentación y al reensamble de los datagramas. Como acabamos de decir, un datagrama IP puede tener un tamaño de hasta 65,535 bytes. Pero, *¿qué sucede cuando el punto final de una* WAN, que maneja uno de estos datagramas, es empalmado con una LAN, con un tamaño máximo del campo de datos de 1,500 bytes? IP fragmenta este datagrama grande en pedazos más pequeños, de manera que puedan ser manejados por todas las redes y, para que el receptor pueda volver a ensamblarlo, existe el campo de Identificación.

- **Identification**: El emisor asigna el campo de Identificación (de 16 bits), para ayudar a reensamblar los fragmentos dentro del datagrama. A todos los fragmentos en que se puede dividir un datagrama se les asigna el mismo identificador. Este contiene un

entero que identifica el datagrama. Cuando se produce la fragmentación de un datagrama, el campo del identificador se copia en todos los datagramas ya fraccionados. De esta manera, el receptor puede identificar los fragmentos que componen el datagrama fragmentado.

- **Flags** Este campo, de 3 bits, es un identificador de control Existen tres indicadores (flags), que señalan cómo será llevado a cabo el proceso de fragmentación.
  - Bit 0. Reservado (valor de 0)
  - Bit 1 (DF) 0 = Se permite, 1 = No fragmentación
  - Bit 2 (MF) 0 = Último fragmento, 1 = Más fragmentos

- **Fragment Offset**. Este campo tiene 13 bits y se utiliza en el reensamble de los datagramas previamente segmentados. Especifica la posición (offset) en bytes de cada fragmento del datagrama original. El campo de offset se va incrementando en cada fragmento del datagrama que se envía empezando con cero

- **TTL** Tiempo de vida del datagrama (*Time to Live*). Este campo, que consta de 8 bits, *especifica en segundos o en saltos, el tiempo que puede viajar por una red un* datagrama El tiempo de vida está limitado a 255 segundos o 4 25 minutos, un tiempo bastante largo para las redes de alta velocidad que se emplean hoy en día. Cada vez que un datagrama pasa a través de un ruteador, resta de este campo el tiempo que tarda en procesar el datagrama (1 como mínimo, aunque el tiempo de proceso sea menor) Cuando este campo alcanza el valor cero antes de alcanzar su destino, se supone que el datagrama está perdido en un bucle cerrado y se descarta. El documento de "Números Asignados" (RFC 1700) especifica un valor, por default, de TTL = 64

- **Protocol** Indica el protocolo de nivel superior para el cual el nivel IP está realizando el servicio de transporte de datos en el datagrama. Esta compuesto por 8 bits. Especifica el formato del área de datos. Como ejemplos de protocolos superiores estan los siguientes:

Decimal	Protocolo	Descripción
1	ICMP	Protocolo de Mensajes de Control de Inernet
6	TCP	Protocolo de Control de Transmisión
17	UDP	Protocolo de Datagrama de Usuario

- **Checksum**: La tercera palabra de la cabecera, se completa con este campo, que está compuesto por 16 bits. Estos son dos bytes de *verificación que afectan a la cabecera* y no a los datos. El *checksum* se calcula como el complemento a uno de la suma (en complemento a uno) de todos los bits que componen la cabecera. Normalmente hay que recalcular el checksum de cada nodo por el que pasa el datagrama, ya que al ir atravesando los diferentes gateways, el campo TTL (tiempo de vida) va variando.

- **Source IP Address**: Esta palabra contiene la dirección IP del host origen.

- **Destination IP Address**. Dirección IP del host destino. Las direcciones dentro de la cabecera IP son direcciones de la Capa de Acceso al Medio. Las direcciones de Internet son direcciones lógicas que dirigen al datagrama, a través de Internet, hacia el host y la red correcta (LAN, MAN o WAN).

- **Options**: Una implementación IP no está obligada a generar diversas opciones para los datagramas que ella misma crea, pero lo que sí debe hacer es procesar los datagramas que la contenga. Ejemplos de opciones son:
  - *Opción de seguridad* utilizada por aplicaciones seguras
  - *Opción de ruta prefijada*: en el campo Options se especifica una lista de direcciones Internet que componen el camino que deberá seguir el datagrama.
  - *Opción de registrar la ruta* el host fuente crea una lista vacía de direcciones Internet en el campo Options y cada máquina que manipula el datagrama ha de grabar su dirección en esta lista.
  - *Opción de registrar la hora.* es similar a la opción anterior. Cada máquina graba la hora en la que manipuló el datagrama y opcionalmente graba también su dirección

- **Padding** Son bits de relleno Cuando se utilizan opciones en el campo Options, los datagramas se rellenan con bits a cero, para ajuste a frontera de 4 octetos.

- **Data** Son los datos contenidos en el datagrama que pasan al protocolo superior indicado en el campo Protocol. Por definición, el tamaño máximo de un datagrama IP es de 65535 bytes y, suponiendo que la longitud total de la cabecera sea de 24 bytes, quedan 65511 bytes para datos.

2.1.3 El Segmento TCP

El segmento TCP se encuentra en la capa de transporte, la cual tiene como función suministrar a las aplicaciones, servicios de comunicación de extremo a extremo. TCP permite la multiplexión, esto es, la capacidad de que una conexión pueda ser utilizada por varios usuarios al mismo tiempo. La transmisión que ofrece TCP es fiable, es decir, permite la recuperación de datos perdidos, erróneos o duplicados y garantiza la secuencia de entrega.

La unidad de datos que maneja TCP se denomina segmento Como se puede ver en la figura 2 1, cada segmento está dividido en dos partes: una cabecera seguida de datos. La cabecera contiene la información de identificación y control.

La cabecera TCP contiene los siguientes campos (mismos que se pueden consultar con más detalle en el capítulo I, sección 5): puertos origen, destino y de urgencia, numeros de secuencia y de reconocimiento, offset, control, ventana, checksum, opciones, relleno y reservado. Después de esta cabecera se encuentran, finalmente, los datos de aplicacion, que constituyen la información que se va a transmitir. Para manejar los datos, la información de los protocolos superiores se encapsula en la información de los protocolos inferiores.

Desde el punto de vista de un datagrama IP, el segmento TCP puede contener una cabecera tanto TCP, como UDP. La diferencia entre ellas es que UDP permite el envío de datagramas a través de la red, sin que se haya establecido previamente una conexión (ofrece servicio orientado a no conexión), para lo que el propio datagrama incorpora la suficiente información de direccionamiento. Esto simplifica notablemente la cabecera

UDP, que contiene únicamente los campos de puertos origen y destino, longitud y checksum, pero, a cambio, no se confirman los datagramas recibidos ni se garantiza su orden, debiendo ser la capa de aplicación la que se encargue de su control.

2.2 FRAGMENTACION Y REENSAMBLE

2.2.1 Introducción

En el proceso de entrega de datagramas, IP debe emplear la fragmentación y el reensamble La fragmentación es necesaria porque las redes LAN y WAN, que cualquier datagrama puede atravesar, podrían manejar diferentes tamaños de tramas, y el datagrama IP debe siempre adecuarse a ellas.

Como lo expusimos en el apartado anterior, la longitud total del un datagrama IP (cabecera, datos e información de la capa superior), puede ser de hasta 65,535 bytes. Pero, cuando un tipo de red (una WAN), se enlaza con otro (una LAN), que maneja otro tamaño máximo del campo de datos, existen problemas Por ello, IP divide este datagrama grande en fragmentos más pequeños, de manera que puedan ser manejados por todas las redes y, para que el receptor pueda volver a ensamblarlo, existe el campo de Identificación

El emisor asigna el campo de Identificación (de 16 bits), para ayudar a reensamblar los fragmentos dentro del datagrama. A todos los fragmentos en que se puede dividir un datagrama se les asigna el mismo identificador. Este contiene un entero que identifica el datagrama. Cuando se produce la fragmentación de un datagrama, el campo del identificador se copia en todos los datagramas ya fraccionados De esta manera, el receptor puede identificar los fragmentos que componen el datagrama fragmentado.

2.2.2 Unidad Máxima de Transferencia, MTU

Todas las redes tienen un tamaño máximo de Unidad de Datos de Protocolo, PDU (*Protocol Data Unit*), llamada Unidad Máxima de Transferencia, MTU (*Maximum Tansfer Unit*) La MTU tiene las siguientes características:

- Indica la longitud de una trama que podrá ser enviada a una red física en particular
- Es determinada por la tecnología de la red física.
- Para el caso de Ethernet es de 1500 bytes

La Unidad Máxima de Transferencia determina la longitud máxima, en bytes, que podrá tener un datagrama para ser transmitido por una red física. Obsérvese que este parámetro está determinado por la arquitectura de la red: para una red Ethernet el valor de la MTU es de 1500 bytes, mientras que para una red FDDI, es de 4470 bytes.

Dependiendo de la tecnología de la red los valores de la MTU pueden ir desde 128 hasta unos cuantos miles de bytes.

La arquitectura de interconexión de redes propuesta por TCP/IP, indica que éstas deben ser conectadas mediante una computadora o equipo de procesamiento denominado gateway, sin obligar a que la tecnología de las redes físicas que se conecten sea homogenea Por tal motivo, si para interconectar dos redes, se utilizan medios con diferente MTU, los datagramas deberán ser fragmentados para que puedan ser transmitidos Y, por supuesto, IP debe contar con un mecanismo de reensamble en el destino final, una vez que los paquetes hayan alcanzado la red extrema, que vuelva a ordenar los fragmentos en el orden en que fueron transmitidos originalmente.

Curiosamente, IP maneja cada operación de fragmento independientemente. Esto es, los fragmentos pueden atravesar diferentes ruteadores hacia su destino, y pueden volver a sufrir fragmentaciones más adelante, si pasan a través de redes que usen MTUs más pequeñas Cada ruteador usa el valor de posición (offset) del fragmento entrante para determinar los valores de posición de los datagramas fragmentados Si ocurre una *fragmentación posterior en otro ruteador, el valor de posición del fragmento es ajustado al* lugar que este fragmento ocupa en relación al datagrama original, y no al paquete fragmentado precedente. La figura 2.3 muestra un ejemplo de operaciones de fragmentación múltiple al cruzar dos ruteadores.

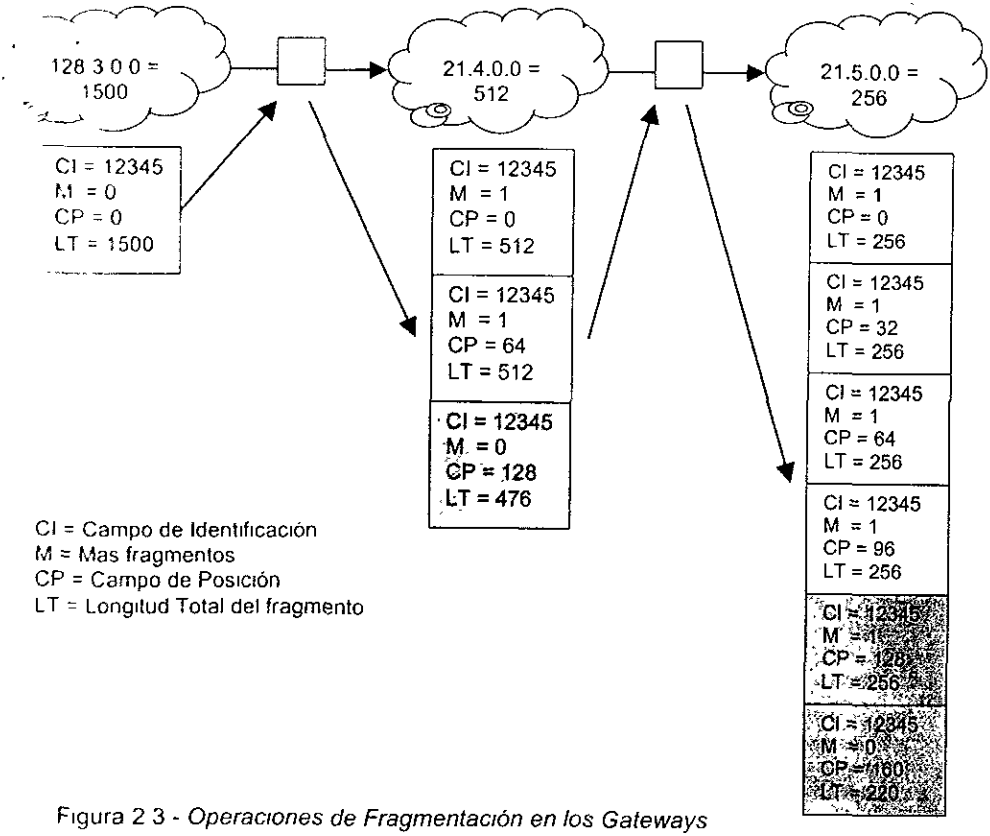


Figura 2 3 - *Operaciones de Fragmentación en los Gateways*

La subred 128.3.0.0 usa un tamaño de PDU de 1500 bytes; pasa esta unidad de datos al ruteador A, el cual decide enrutar la PDU a la subred 21 4.0.0, la cual, a su vez, soporta un tamaño de PDU de 512 bytes. El ruteador fragmenta la unidad de datos de 1500 bytes en tres unidades de datos más pequeñas: 512, 512 y 476 bytes El último segmento, que contiene 476 octetos, es rellenado con ceros, hasta igualar un tamaño total que sea múltiplo de 8. De esta manera, este campo de datos es de 480 bytes.

El ruteador A pasa los datos a la subred 21.4 0.0, la cual los manda al ruteador B. Este ruteador determina que los fragmentos del datagrama serán transferidos a la subred 21 5 0 0 Debido a que el ruteador sabe que esta red usa una MTU de 256 bytes, efectúa nuevamente una fragmentación: divide los fragmentos de 512 bytes en unidades de datos mas pequeñas aún. Usando los valores de posición de los tres fragmentos entrantes, ajusta, correspondientemente, los valores de posición de las unidades de datos salientes. Hay que hacer notar que los valores de posición son reiniciados en el ruteador B, y sus valores se derivan de los valores de posición contenidos en los fragmentos precedentes.

La figura 2.4 ilustra el reensamble de los paquetes, el cual ocurre en el host receptor. El modulo IP configura un espacio de búfer, cuando los primeros fragmentos son recibidos Un búfer es reservado para cada fragmento, y éste es situado en una área dentro del búfer, relativa a su posición en el datagrama original Tal como van llegando los fragmentos, son colocados en el lugar adecuado dentro del búfer. Cuando todos los fragmentos han sido recibidos, el módulo IP pasa los datos a un protocolo de nivel *superior, en el mismo orden como fue originalmente enviado por el emisor.* En esta figura se asume que los fragmentos de datagrama, mostrados en la figura anterior, fueron enviados a otros ruteadores: digamos C y D.

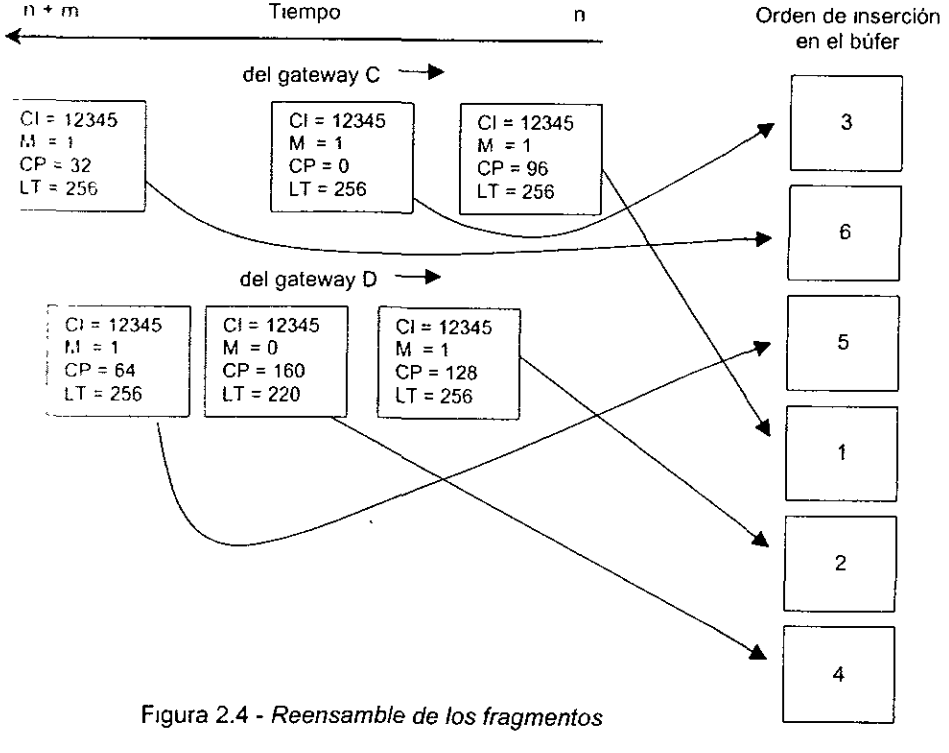


Figura 2.4 - *Reensamble de los fragmentos*

La figura 2.4 muestra que los fragmentos llegan desde los ruteadores C y D en el orden señalado por la flecha de tiempo, donde la primera llegada es en *n* y el último tiempo de llegada es *n* + *m* Los fragmentos, por consiguiente, llegaron en el siguiente orden (usando los valores de posición en la figura para identificar el fragmento):

Primero. fragmento con valor de posición de 96
Segundo fragmento con valor de posición de 128
Tercero fragmento con valor de posición de 0
Cuarto: fragmento con valor de posición de 160
Quinto fragmento con valor de posición de 64
Sexto: fragmento con valor de posición de 32

La maquina receptora tiene el trabajo relativamente fácil, de descifrar donde serán ubicados los fragmentos El módulo IP simplemente multiplica el valor de posición por 8, para determinar qué ranura del búfer debe recibir al fragmento Por ejemplo, la posición relativa en el búfer, del primer fragmento que llegó, se calcula como 96 x 8 = 768, o la dirección de memoria 768.

En la figura 2.4, el host que efectúa el reensamble no conoce la longitud completa del datagrama IP, sino hasta que recibe el cuarto fragmento, el cual contiene el bit M = 0 (no mas fragmentos), el valor de posición y la longitud del fragmento. Debido a que el valor de posición es 160 y la longitud es de 220 octetos, el host ya sabe ahora que el datagrama total es de 1500 bytes, ya que (valor de posición 160) x (8 octetos por valor) + (220 octetos en el fragmento final) = 1500 bytes.

Ahora se puede ver por qué el bit M tiene tanta importancia. Debido a que el campo de longitud en el fragmento no se refiere al tamaño del datagrama original, sino al tamaño del fragmento, el único método para determinar la longitud original (y el fragmento final) es el indicador M = 0.

Si algunos fragmentos no llegan o han sido descartados, debido a que excedieron el parametro TTL, IP desecha los fragmentos del datagrama parcialmente reensamblado. Además, cuando se realiza la detección de la llegada del primer fragmento, la computadora receptora activa un contador de reensamble Este contador, que es configurado por el administrador de la red, es empleado para asegurar que todos los fragmentos lleguen de manera oportuna. Si el contador expira antes de que todos los fragmentos hayan llegado, los fragmentos recibidos son desechados.

Si el usuario no desea que exista fragmentación, el indicador de fragmentos (flag) debe ser puesto en 1, lo cual indica que no se efectuará ninguna fragmentación. Esto puede ser ventajoso, si la fragmentación provoca un sobreencabezado excesivo, lo cual es debido a que el contador de reensamble continúa descartando fragmentos que requieren ser retransmitidos desde protocolos de capas superiores. Esta situación, sin embargo, debe ser tomada en cuenta contra el hecho de que la activación del indicador "No fragmentar", *significa que el datagrama será desechado por los ruteadores si la MTU excede el tamaño* de la capacidad de la red

2.2.3 Paquetes reensamblados en el ruteador

Los ruteadores orientados a no conexión que usan el ruteo dinámico (como IP), no reensamblan los datagramas. Esta tarea es imposible, puesto que todos los fragmentos

pertenecientes al datagrama original, pueden no ser procesados por los mismos ruteadores. Como consecuencia, el ruteador no sabe cómo calcular los valores de posicion para los fragmentos que él no recibe.

En contraste, los ruteadores orientados a conexión pueden efectuar un reensamble intermedio, ya que, debido a la naturaleza del sistema, todas las PDUs pasan a través de los mismos ruteadores. Por ejemplo, supongamos que el ruteador A en la figura 2.3, que ilustra la fragmentación, es un ruteador orientado a conexión. Este puede reensamblar los fragmentos para su transmisión sobre una subred de mayor capacidad, a través de la detección del indicador M = 0 en un fragmento, así como también el valor de posición de 128 y longitud de 476, lo que da como resultado 1500 bytes (valor de posición 128 x 8 octetos por valor + 476 octetos en el fragmento final)

2.3 RUTEO EN IPv4

El enrutamiento se refiere al proceso de determinar la trayectoria que un datagrama debe seguir para alcanzar su destino. A los dispositivos que pueden elegir las trayectorias se les denomina ruteadores. En el proceso de enrutamiento intervienen tanto los equipos como las compuertas que conectan redes, hay que recordar que el término compuerta es impuesto por la arquitectura TCP/IP de conexión de redes, sin embargo una compuerta puede realizar diferentes funciones a diferentes niveles, una de esas funciones puede ser la de enrutamiento y por tanto recibir el nombre de ruteador.

La operación de los ruteadores se basa en los siguientes planteamientos.

- Los datos se envían a los ruteadores y no a través de los ruteadores.
- Los ruteadores solamente analizan la información a ellos dirigida.
- Los ruteadores basan su decisión en la dirección de nivel de red.
- Los ruteadores mantienen una tabla de rutas por cada protocolo que soportan.

En base a estos planteamientos, los ruteadores minimizan los mensajes de difusión del nivel de enlace y permiten la utilización de la misma dirección de enlace en diferentes segmentos y subredes.

Para que los ruteadores puedan operar, los protocolos de nivel de red deben ser lo que se denominan protocolos enrutables, es decir, que están preparados para transportar información de ruta en sus tramas. Normalmente, como en el caso del protocolo IP utilizado en Internet, la información de ruta está proporcionada mediante unos campos específicos que indican la red y el nodo de destino. Algunos ejemplos de protocolos enrutables son: TCP/IP, APPN (*Advanced Peer to Peer Networking*), DecNet (de la arquitectura DNA, desarrollada por DEC), IPX (*Internetwork Packet Exchange*),y OSI (*Open Systems Interconnection*).

Los protocolos no enrutables no transportan información de ruta, por lo que no pueden operar con ruteadores. Algunos protocolos no enrutables son: DEC LAT, Netbios y SNA jerárquico

La operación básica de los ruteadores es la siguiente:

Cada nodo mantiene una tabla de ruteo en la que se almacena, entre otra información, los siguientes datos por cada entrada correspondiente a una red de destino:

- Dirección de la red destino.

- Dirección del próximo ruteador para alcanzar la red destino
- Interfaz puerto por donde debe dirigirse la información
- Métrica: parámetro asociado a la red destino, como el tiempo de tránsito o el número de ruteadores que se deben atravesar para alcanzarla

Una vez que la información llega al ruteador, éste consulta los datos y especificaciones de su tabla de ruteo y, con base en éstos, decide por dónde dirigir la información hacia su destino.

Los tipos de enrutamiento para alcanzar la red de destino pueden ser:

- Enrutamiento Directo

  Cuando la red de destino está conectada al ruteador. Existe una transmisión de datagramas IP entre dos equipos de la misma red física sin *la intervención de gateways El emisor encapsula el datagrama en la trama* de la red, efectuando la vinculación entre la dirección física y la dirección IP, y envía la trama resultante en forma directa al destinatario.
- Enrutamiento Indirecto

  Cuando la red de destino se alcanza mediante otros ruteadores. Los gateways forman una estructura cooperativa, interconectada. Envían los datagramas hasta que se alcanza al gateway que puede distribuirlos en forma directa a la red destino
- Enrutamiento por Defecto

  Ruta directa o indirecta a seguir en caso de que la dirección de la red de destino no se encuentre en la tabla

Debido a que en el enrutamiento directo los datagramas se transmiten de un equipo a otro, en la misma red física, el proceso es muy eficiente. La vinculación entre la dirección física y la IP se realiza mediante el ARP. En el indirecto la transmisión del datagrama se efectua mediante la intercesión de los gateways. Aquí el gateway que actúa como *ruteador debe estar provisto de mecanismos para conocer, y por tanto decidir, la* trayectoria de la red que se desea alcanzar.

Enrutamiento Indirecto: En este direccionamiento, un equipo debe enviar a un gateway el datagrama con destino a una red física distante. El gateway de la red física envía el datagrama a otros gateways hasta alcanzar a aquel que puede emitirlo en forma directa a la red destino. El gateway debe conocer las rutas hacia las diferentes redes externas, ellas pueden utilizar a su vez un enrutamiento indirecto en el caso de no conocer la ruta a una red específica. Los gateways conocen las trayectorias a otra red mediante Tablas de Enrutamiento.

2.3.1 Tablas de ruteo IP

Este es el algoritmo comúnmente utilizado para el enrutamiento de IP. Las tablas de enrutamiento están presentes en todo equipo que almacene información de cómo alcanzar posibles destinos. En las tablas no se almacena la ruta específica a un equipo, sino aquella de la red donde se encuentre. Cada puerto de comunicación del gateway debe poseer una dirección IP.

- Si cada tabla de ruteo conservara información sobre todos los destinos posibles, el espacio sería insuficiente.
- Es necesario que con un mínimo de información, el equipo pueda tomar decisiones de ruteo
- Una técnica para mantener tablas de ruteo pequeñas consiste en enviar los datagramas a destinos predeterminados (redes predeterminadas)

Para que en los equipos no exista una tabla excesivamente grande, que contenga todas las rutas a las redes que se interconecta el equipo, es de gran utilidad definir una ruta por omisión. A través de esta ruta se deberán alcanzar todas las redes destino.

La ruta por omisión apunta a un dispositivo que actúa como compuerta de la red donde se encuentre ubicado el equipo que la posee

Las tablas de ruteo pueden ser estáticas o dinámicas. Las tablas estáticas se establecen por configuración, mientras que las tablas dinámicas se obtienen mediante el intercambio de información entre los ruteadores. A su vez, las tablas dinámicas se derivan, bien del conocimiento de la topología de la red o bien del conocimiento del ruteador siguiente y de las distancias a las redes destino

Hasta 1990, el diseño de las tablas de ruteo IP estaba basado en la percepción individual de los vendedores sobre la necesidad de los parámetros que se debían incluir en ellas. Pero con la publicación de la Base de Información de Administración de Internet (MIB, *Internet Management Information Base*), está disponible ahora una definición más formal de las tablas de ruteo. La tabla 2.1 muestra una tabla de ruteo IP, tal como está *definida en el estándar de la MIB, que fue publicado en el RFC 1213*

Cada fila de la tabla de ruteo contiene una entrada para cada ruta conocida por el ruteador. Las columnas representan la información disponible en cada ruta.

	Destino	Indice de Interfaz	Métrica 1	Métrica 5	Siguiente Salto	Tipo de ruta	Protocolo de ruteo	Edad de la ruta	Máscara de ruteo	Información de la ruta
Ruta 1										
Ruta 2										
Ruta 3										
Ruta n										

Tabla 2.1 - *Tabla de ruteo*

La columna *Destino* contiene la dirección IP del destino de la ruta. Si esta columna esta determinada como 0.0.0.0, la ruta es considerada como ruta por default.

La columna *Indice de Interfaz* contiene la ubicación de la interfaz local (más comúnmente conocida como puerto físico), a través de la cual puede ser alcanzado el siguiente punto (salto) en la ruta.

Las siguientes cinco columnas contienen la etiqueta de *Métrica*. Estas entradas contienen información acerca de la métrica de costo usada para determinar la ruta. La métrica de costo es el número de saltos necesarios para alcanzar el destino. Con la evolucion de protocolos de descubrimiento de ruta más sofisticados (como OSPF), es posible, sin embargo, que más de una métrica de costo pueda ser usada en el cálculo de una ruta Como la tabla lo indica, pueden ser almacenadas hasta cinco métricas de costo.

El parametro *Siguiente salto* contiene la dirección IP que identifica el siguiente punto para una ruta

La siguiente columna de la tabla es el *Tipo de Ruta*. Este es uno de cuatro posibles valores para proporcionar la siguiente información

1 = ninguna de las siguientes
2 = ruta inválida
3 = ruta conectada directamente (una subred conectada ecuante)
4 = ruta indirecta (una conexión indirecta para alcanzar otro)

La siguiente columna se refiere al *Protocolo de ruteo* entrada identifica el protocolo de descubrimiento de ruta por medio del cual aprendida una ruta (recordemos que los ruteadores tienen la capacidad de aprender las rutas después de determinarlas la primera vez).

La columna de la *Edad de la Ruta* se refiere al tiempo transcurrido, desde que la ruta fue actualizada o verificada por última vez.

La siguiente columna contiene la *Máscara de la Ruta* aliza la función AND entre esta máscara y la dirección destino del datagrama antes de ser comparada con la primer columna de esta tabla (para mayor información sobre máscaras, véase el apartado 2 4 3)

La última columna, denominada *Información de ruta* permite hacer una referencia a una definición MIB para un protocolo de ruteo particular. El valor de esta columna depende del tipo de protocolo de ruteo usado.

Ruteadores

Hay que señalar que en Internet se suele denominar *gateway* lo que en otros casos se denomina *ruteador* o compuerta. Existen dos tipos de ruteadores o gateways. los internos y los que conectan redes autónomas (es decir, administradas separadamente). La diferencia fundamental es que los que pertenecen a distintas redes han de seguir protocolos estándares *de facto*. Los ruteadores internos pueden seguir un protocolo determinado Atendiendo a esto, Internet clasifica los protocolos de ruteo en los siguientes tipos:

1 ***Core Gateways***: ruteadores principales
Los principales protocolos son:

- GGP (*Gateway-Gateway Protocol*), para comunicar dos ruteadores principales. El protocolo GGP utiliza los servicios de los datagramas IP. Cada mensaje GGP tiene una cabecera con formato fijo que identifica el tipo de mensaje y el formato de los campos
- EGP (*Exterior Gateway Protocol*), para comunicar ruteadores exteriores de sistemas autónomos e intercambiar información de alcance. Cuando dos ruteadores intercambian información de ruta se denominan vecinos. EGP soporta un procedimiento de "adquisición de vecino" que permite a un ruteador pedir a otro que participe con él en el intercambio de ruta. Un ruteador está verificando

continuamente si los EGP vecinos responden, ya que éstos deben intercambiar frecuentemente información de red mediante el paso de mensajes de actualización de ruta.

2 ***Non-Core Gateways***: ruteadores internos
Los principales protocolos son:

- EGP, con un Core Gateway
- IGP (*Interior Gateway Protocol*), para enlazar con otro ruteador interno. Para automatizar la tarea de mantener actualizada la información de ruta, los ruteadores internos se comunican normalmente con otros intercambiando datos de accesibilidad o información de ruta. La mayoría de los sistemas autónomos emplean un único protocolo para propagar información de ruta interna. Algunos sistemas emplean EGP como IGP. Entre los protocolos IGP se pueden mencionar: HELLO, RIP y OSPF. El primero ya no es usado, por lo que no lo trataremos; los otros dos los presentaremos más adelante.

2.3.2 Ruteadores principales

Este tipo de ruteadores está controlado por la Internet Activities Board (IAB). Fueron instalados en un primer momento en ARPANET. El sistema de ruteadores principales está diseñado para proveer rutas consistentes para todos los posibles destinos. Estos ruteadores se comunican entre sí para garantizar que la información compartida sea consistente

En la arquitectura de interconexión de redes de TCP/IP cada par de redes se conecta mediante ruteadores. Para que los paquetes alcancen sus redes destino los ruteadores deben contar con mecanismos mediante los cuales intercambien la información de las redes que conecta cada uno.

Los ruteadores principales se encargan de conectar sistemas autónomos. Se entiende por sistema autónomo un conjunto de redes y ruteadores bajo una misma administración. Para hacer posible la aplicación de algoritmos de ruteo automático, cada sistema autónomo tiene un número asociado y suministrado por la autoridad central que asigna todas las direcciones Internet. Cada sistema autónomo es libre de escoger la arquitectura de ruteo, pero debe recoger información de todas las redes y dársela a uno o más gateways exteriores por donde pasará información de otros sistemas autónomos.

En la Arquitectura de Enrutamiento por Ruteador Principal existe un ruteador que centraliza las funciones de enrutamiento entre redes, a este ruteador se le denomina principal o núcleo. Cada ruteador en las redes a conectar tiene como ruteador por omisión al principal. Varios ruteadores principales pueden conectarse para formar una gran red; *entre ellos se intercambiará información concerniente a las redes que cada uno alcanza.*

Debido a que Internet emplea la arquitectura *Core*, cada sistema debe pasar información al ruteador principal de Internet. El sistema *Core* fue diseñado para permitir la conexión de nuevos ruteadores principales (core) sin modificar los existentes. Cuando se añade un nuevo ruteador al sistema Core, se le asignan uno o más ruteadores principales vecinos con los que se comunica. Los vecinos, también miembros del Core, propagan información de ruta entre ellos mismos. De esta manera, los nuevos ruteadores sólo necesitan informar a sus vecinos acerca de las redes a las que pueden acceder; ellos actualizan sus tablas de ruteo y propagan esta nueva información.

Las características más importantes de la Arquitectura de Enrutamiento por Ruteador Principal son las siguientes:

- Es el primer esquema de enrutamiento que existió.
- Los ruteadores de diferentes redes se conectan a uno principal
- El ruteador principal es el ruteador por omisión de las redes locales.
- Los ruteadores principales no pueden contar con ruteadores por omisión

Entre las desventajas que presenta esta arquitectura, podemos mencionar las siguientes

- Es conveniente sólo para redes administradas centralizadamente.
- Los ruteadores principales deben almacenar toda la información de las rutas hacia las redes que conectan
- Complejidad de administración de acuerdo a la complejidad o cambios en la red.

La arquitectura centralizada de enrutamiento fue la primera que existió. Sus principales problemas radican no tanto en la arquitectura en sí, sino en la forma en que se propagaban las rutas entre las compuertas núcleo.

Propagación automática de rutas

Conforme las complejidades de las redes aumentaron se debió buscar un mecanismo que propagase la información de rutas entre las compuertas. Este mecanismo debía ser automático, lo cual está obligado por el cambio dinámico de las redes. De no ser así, las transiciones entre las compuertas podían ser muy lentas y no reflejar el estado de la red en un momento dado.

La propagación automática de rutas establece algoritmos como el de vector-distancia, o GGP (Protocolo de Gateway a Gateway), para el intercambio de información entre ruteadores, y así poder mantener la información de rutas siempre actualizada. Toma en consideración el hecho de que las redes son dinámicas y siempre están sufriendo modificaciones y no obliga a tener un esquema centralizado de ruteo

Los ruteadores principales emplearon originalmente un protocolo de vector-distancia (metrica de los ruteadores) conocido como GGP, que ya mencionamos. El término vector-distancia se refiere a una clase de algoritmo que emplean los ruteadores para propagar la información de rutas. La información de ruta intercambiada con el protocolo GGP consiste en una pareja de números (R, D), donde R es una dirección IP de una red y D es la distancia a dicha red, expresada en saltos o número de ruteadores que debe atravesar. No siempre un número de saltos menor implica una mayor velocidad.

Algoritmo de Vector-Distancia

- Se asume que cada ruteador comienza su operación con un conjunto de reglas básicas de cómo alcanzar las redes que conecta.
- Las rutas son almacenadas en tablas que indican la red y los saltos para alcanzar esa red.
- Periódicamente cada ruteador envía una copia de las tablas que alcanza directamente.

- Cuando un ruteador recibe el comunicado de otro, actualiza su tabla incrementando en uno el número de saltos.

Aunque los algoritmos vector-distancia son fáciles de implementar, tienen desventajas significativas. En un entorno completamente estático, estos algoritmos propagan rutas a todos los destinos Cuando las rutas cambian rápidamente debido a nuevas conexiones o al fallo de las existentes, por ejemplo, la información se propaga lentamente de un ruteador a otro. En este intervalo, los ruteadores pueden tener información de ruta incorrecta Por otra parte, los mensajes de actualización de ruta contienen una entrada por cada red posible, con lo que la longitud del mensaje es proporcional al número total de redes Dado que, adicionalmente se requiere la participación de todos los ruteadores en el intercambio de información de ruta, el volumen de información intercambiado es muy elevado

El concepto de este vector ayudó a definir a través de cuántos ruteadores debería viajar un paquete para alcanzar su red destino Mediante el vector, un ruteador podía saber a qué otro ruteador enviar el paquete de información, sabiendo que éste podría no ser el último por el que el paquete tendría que viajar Este esquema permite tener varios caminos a una misma red, eligiendo el camino más corto, es decir aquel ruteador que con menos saltos conduzca a la red destino.

2.3.3 El Protocolo RIP

Ruteadores internos

Estos ruteadores son internos a los sistemas autónomos. Están controlados por los administradores de los mismos Por tanto, no tienen por qué seguir unas normas estándar, a excepción de los que se comunican con ruteadores principales. Como ya lo mencionamos líneas arriba, los protocolos más empleados por los ruteadores internos son el EGP y el IGP. Dentro de éste último, se encuentran los protocolos RIP y OSPF, que describimos a continuación, haciendo hincapié en que el primero de ellos ya no es muy empleado, la tendencia es hacia la utilización cada vez mayor del segundo.

Protocolo RIP

El Protocolo de Información de Ruteo (RIP, *Routing Information Protocol*) fue uno de los protocolos interiores más empleados anteriormente en los sistemas Unix. Tiene dos versiones, en las que la diferencia principal es que en la primera existe un intercambio de tablas completas, mientras que en la segunda versión el intercambio es únicamente de actualizaciones. Este protocolo fue inicialmente creado por la Universidad de Berkeley y está basado en las investigaciones de Xerox, generalizadas para cubrir diferentes familias de redes

El protocolo RIP está basado en el algoritmo vector-distancia. Los ruteadores activos informan de sus rutas a otros ruteadores; los pasivos escuchan y actualizan sus tablas de ruteo, pero no las notifican. Generalmente, los ruteadores operan RIP en modo activo, mientras que los hosts o nodos lo realizan en modo pasivo.

Un ruteador en modo activo envía un mensaje de difusión cada 30 segundos. El mensaje contiene información de la tabla del ruteador. Cada mensaje consiste en un par, que contiene la dirección IP y la distancia a dicha red En la métrica RIP, un ruteador tiene distancia "uno" para redes que están directamente conectadas, dos para redes en las que hay que atravesar un ruteador y así sucesivamente.

Tanto los ruteadores RIP pasivos como los activos, escuchan todos los mensajes de difusión y actualizan sus tablas de acuerdo al algoritmo del vector-distancia descrito anteriormente Por ello, no es un algoritmo que ofrezca resultados óptimos cuando existen redes con mucha diferencia de velocidades

Para evitar oscilaciones en la información de rutas entre dos o más caminos de igual coste (distancia), RIP especifica que un ruteador que aprende una ruta, ha de mantenerla hasta que aprende otra de menor coste Además, para evitar la inestabilidad, RIP emplea un valor bajo para la máxima distancia posible. La distancia máxima contemplada en este protocolo es 16 Por ello, RIP no permite más de 15 ruteadores, con lo que, para efectos prácticos, si la métrica tiene el valor 16, es considerada como infinito.

El algoritmo del vector de distancia genera una gran cantidad de información de ruta cuando desaparece un enlace Escogiendo un infinito pequeño (15) se limita dicha congestión pero no se elimina.

Cuando un ruteador actualiza una ruta en su tabla, activa un temporizador para dicha ruta Si a los 180 segundos no ha recibido nueva información de la misma, dicha entrada queda invalidada. El temporizador es iniciado cada vez que recibe información sobre la ruta El inconveniente de este algoritmo es *que no detecta bucles en la transmisión de* información de ruta.

El protocolo RIP también borra rutas de la tabla de ruteo. Esto lo lleva a cabo de dos maneras Primero, si el ruteador hacia el destino dice que el coste de la ruta es mayor que 15, la ruta es borrada. Segundo, RIP asume que un ruteador que no envía actualizaciones esta caido o muerto Todas las rutas a través del ruteador son borradas si no se reciben actualizaciones de ese ruteador en un periodo determinado de tiempo. Por lo general, en muchas implementaciones, si un ruteador no envía actualizaciones de ruteo durante 180 segundos, entonces todas las rutas que involucren a ese ruteador serán eliminadas de la tabla de ruteo.

RIP es fácil de implementar y sencillo de configurar, pero tiene tres serios inconvenientes, a saber:

1 *Diametro de red limitado* La ruta más larga es de 15 saltos. Un ruteador RIP no *puede mantener una tabla de ruteo completa de una red que tiene destinos más allá* de 15 saltos El contador de saltos no puede ser incrementado por el segundo inconveniente

2 *Convergencia lenta* La eliminación de una ruta errónea requiere, a veces, del intercambio de múltiples paquetes de actualización de ruta, hasta que el coste de la ruta alcance un valor de 16. Esto es llamado "cuenta hacia infinito", porque RIP se mantiene incrementando el coste de la ruta hasta que alcanza un valor más allá del permitido por su métrica (en este caso, 16 es infinito). Adicionalmente, RIP debe esperar 180 segundos antes de eliminar las rutas inválidas. Estas condiciones retrasan la convergencia del ruteo; es decir, a la tabla de ruteo le toma un tiempo muy grande reflejar el estado actual de la red.

3 *Ruteo de clase* RIP interpreta todas las direcciones utilizando las condiciones descritas en el primer apartado de este capítulo. Para RIP todas las direcciones son clase A, B o C, lo cual hace a RIP incompatible con las superredes de Ruteo de Inter-dominio sin Clase (CIDR, *Classless Inter-Domain Routing*) e incapaz de soportar subredes de longitud variable

Para evitar la cuenta hacia infinito, existen dos características llamadas "Horizonte dividido" y "Regreso envenenado". Con la primera característica, el ruteador no da a conocer rutas en el enlace desde el cual dichas rutas fueron obtenidas. El Regreso envenenado dice que un ruteador debe dar a conocer una distancia infinita para las rutas en ese enlace

Tambien existe la característica de Actualizaciones Disparadas, las cuales, en lugar de esperar el tiempo normal de 30 segundos como intervalo entre actualizaciones, envía inmediatamente una actualización disparada Por lo tanto, cuando un ruteador se descompone o un enlace local se cae, el ruteador actualiza inmediatamente su tabla de ruteo y envia los cambios a sus vecinos. Esta característica emplea también eficientemente el ancho de banda de la red, ya que no incluye la tabla de ruteo completa, sino unicamente las rutas que han cambiado

2.3.4 El Protocolo OSPF

El protocolo OSPF (*Open Shortest Path First Protocol*) es un protocolo de estado de enlace, al contrario que RIP, el cual es un protocolo de vector de distancia. El OSPF se derivo del protocolo de enrutamiento IS-IS de la OSI Es una alternativa al RIP entre los protocolos para ruteadores internos, corrigiendo todas las limitaciones que tenia éste; está basado en el algoritmo SPF (*Shorter Path First*). Este algoritmo reemplaza al algoritmo de vector-distancia, y requiere que cada ruteador tenga una información completa de la topologia de la red En vez de enviar mensajes que contengan una lista de destinos, un ruteador participa en el algoritmo SPF realizando dos tareas. en primer lugar, un ruteador no intercambia distancias con sus vecinos, sino que activa una verificación de todos los ruteadores vecinos (en términos de redes, dos ruteadores vecinos están conectados a una misma red). En segundo lugar, propaga periódicamente a través del sistema autónomo, la información de estado de sus enlaces al resto de los ruteadores. De esta manera, cada ruteador captura esta información y construye su tabla de enrutamiento, de forma tal, que todos los ruteadores involucrados tendrán la misma tabla.

Desde un punto de vista práctico, la diferencia más importante es que un protocolo de estado del enlace converge con más rapidez que un protocolo de vector de distancia. Por convergencia se entiende que la estabilización después de cambios en la red, como caidas de ruteadores o de enlaces, será notablemente más rápida.

OSPF se diferencia de RIP (y de otros muchos protocolos de encaminamiento) en que utiliza directamente IP, o sea, no utiliza UDP ni TCP.

Además de ser un protocolo de enlace en vez de distancia, OSPF tiene otras muchas características que lo hacen superior a RIP, como son:

1. OSPF puede calcular un conjunto separado de rutas para cada tipo de servicio IP. Esto quiere decir que para un mismo destino puede haber varias entradas en la tabla de enrutamiento, una por cada tipo de servicio.
2. A cada interfaz se le asigna un coste. Este puede asignarse en función del ancho de banda de salida, seguridad, fiabilidad, etc. Pueden asignarse distintos costes para distintos servicios.
3. Cuando existen varias rutas a un mismo destino, con idénticos costes, OSPF distribuye el trafico por ambas rutas de forma equitativa.

4 OSPF soporta subredes: una máscara de subred es asociada con cada ruta notificada Esto permite que una única dirección IP de cualquier clase pueda ser dividida en múltiples subredes de varios tamaños Las rutas a un host son notificadas mediante una máscara de subred con todos los bits a 1. Una ruta por defecto es notificada como una dirección IP de 0.0.0.0 con una máscara con todos los bits a 0.

5 Los enlaces punto a punto entre ruteadores no necesitan una dirección IP a cada extremo, lo que se conoce como redes no numeradas. De esta forma se ahorran direcciones IP

6 Es posible emplear un pequeño mecanismo de autenticación, por lo que se puede enviar una contraseña (password), de forma parecida a la empleada en RIP.

7 OSPF emplea multicast en vez de broadcast, para reducir la carga en los sistemas que no emplean OSPF.

El protocolo OSPF incluye dos tipos de rutas de servicio. Los administradores pueden definir varias rutas para un mismo destino, cada una para un tipo de servicio. Cuando se enruta un datagrama, un ruteador bajo OSPF emplea tanto el campo dirección de destino, como el campo del tipo de servicio. De hecho, es el primer protocolo TCP/IP que emplea el campo de tipo de servicio de ruta, es decir, tiene enrutamiento basado en un tipo de nivel superior de solicitudes del servicio (TOS, *Type of Service*). Por ejemplo, una aplicación puede especificar que ciertos datos son urgentes y si OSPF tiene enlaces de alta prioridad a su disposición, ellos pueden ser utilizados para transportar un datagrama urgente. OSPF soporta una o más métricas.

Como ya lo dijimos, OSPF puede equilibrar la carga; si el administrador especifica más de una ruta con el mismo coste, OSPF distribuye el tráfico entre éstas. De nuevo, OSPF es el primer IGP abierto que ofrece equilibrio de carga. RIP, por ejemplo, sólo admite una única ruta

El protocolo OSPF permite dividir la red y los ruteadores en subconjuntos llamados áreas, para permitir el crecimiento de la misma y hacer que su administración sea más sencilla Cada área es autocontenida; el conocimiento de la topología de un área permanece transparente al resto de las áreas. Por otro lado, muchos grupos de una zona dada pueden cooperar en el empleo de OSPF para el enrutamiento, incluso si cada grupo se reserva la decisión de cambiar la topología interna.

OSPF especifica que todos los intercambios entre ruteadores deben ser autentificados, permite una variedad de esquemas e, incluso, permite a un área escoger un esquema distinto de otra. Con la autenticación sólo los ruteadores autorizados propagan información de ruta, evitándose problemas de seguridad, como el que s describe a continuación, utilizando un protocolo RIP.

Un pirata informático (*hacker*) podría simplemente, por medio de su computadora personal, propagar mensajes RIP informando que su dirección corresponde a la ruta de mas bajo coste, con lo que otros nodos y ruteadores comenzarán a enviarle información.

La tabla 2.2 muestra una comparación de las características de OSPF y de RIP en su segunda versión, donde se puede ver que:

Función	OSPF	RIP
Técnica	Estado de enlaces	Vector-distancia
RFC	1245, 1583	1058, 1387-89
Carga de red	controlada	rápido crecimiento
Selección de ruta	tipo de servicio	no

Equilibrio de cargas	sí	no
Métrica	SPF	saltos
Seguridad	autenticación	no
Convergencia	rápida	lenta
Complejidad de los ruteadores	mayor	menor

Tabla 2.2 - *Características de OSPF y RIP*

2.4 LAS DIRECCIONES EN IPv4

Interconexión de Redes

Para entender el funcionamiento de los protocolos TCP/IP debe tenerse en cuenta la arquitectura que ellos proponen para comunicar redes. Tal arquitectura ve como iguales a todas las redes a conectarse, sin tomar en cuenta el tamaño de ellas, ya sean locales o de cobertura amplia. Define que todas las redes que intercambiarán información deben estar conectadas a una misma computadora o equipo de procesamiento dotado con dispositivos de comunicación; a tales computadoras se les denomina compuertas (gateways), pudiendo recibir otros nombres como ruteadores o puentes

La Arquitectura de Interconexión de Redes en TCP/IP pretende las metas siguientes:

- La tecnología de conexión de ser independiente de la arquitectura de la computadora.
- Conectividad Universal a través de la red.
- Reconocimientos de extremo a extremo.
- Protocolos de Aplicación Estandarizados.

Las características de la arquitectura de interconexión de redes son las que se mencionan a continuación:

- Protocolos de no conexión en el nivel de red.
- Conmutación de paquetes entre nodos.
- Protocolos de transporte con funciones de seguridad
- Conjunto común de programas de aplicación

Direcciones IP

Para que en una red dos computadoras puedan comunicarse entre sí, deben estar identificadas con precisión. Esta identificación puede estar definida en niveles bajos (identificador físico) o en niveles altos (identificador lógico), dependiendo del protocolo utilizado. TCP/IP utiliza un identificador denominado dirección Internet o dirección IP, cuya longitud es de 32 bits. Las direcciones IP están divididas en dos secciones, una que identifica la red a la que pertenece una computadora (Network ID) y otra que la identifica a ella misma dentro de dicha red (Host ID). Es decir, las direcciones Internet tienen las siguientes características:

- Longitud de 32 bits.
- Identifican a las redes y a los nodos conectados a ellas.
- Especifican la conexión entre redes.
- Se representan mediante cuatro octetos escritos en formato decimal, separados por puntos

Las direcciones IP pueden tener cinco formatos diferentes. Los formatos difieren en el numero de bits que son empleados tanto para la identificación de la red, como para el host, éstos son identificados por los primeros tres bits; es decir, existen cinco clases de direcciones, las cuales abordaremos en el apartado 2.4.2. A continuación veremos los tipos de direcciones.

2.4.1 Tipos de Direcciones

Para IPv4, están definidos tres tipos diferentes de direcciones, los cuales son: unidireccional (*unicast*), bidireccional (*broadcast*) y multidireccional (*multicast*).

Unidireccional

Para este tipo de direccionamiento, tenemos que existe un identificador para una interfaz simple. Un paquete enviado a una dirección unidireccional es entregado a la interfaz identificada por esa dirección. La figura 2.5 representa este tipo de dirección, donde se puede observar que:

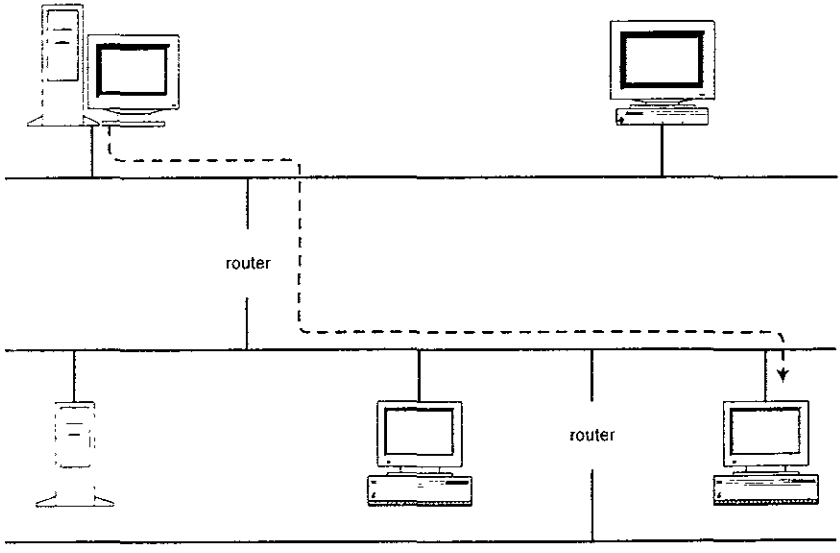


Figura 2.5 - *Direccionamiento Unidireccional*

Bidireccional

En este tipo de direccionamiento existe un identificador para un conjunto de interfaces, generalmente pertenecientes a nodos diferentes. Un paquete enviado a una dirección bidireccional, es entregado en una de las interfaces identificadas por esa dirección, la más cercana, de acuerdo con la medición de distancia del protocolo de ruteo, como se ve en la figura 2.6

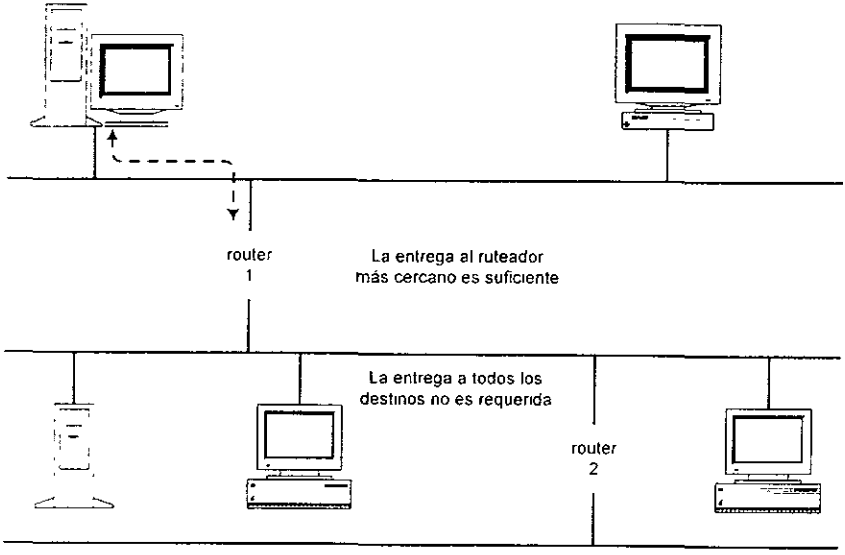


Figura 2.6 - *Direccionamiento Bidireccional*

Omnidireccional

Este tipo de direccionamiento casi no es empleado porque genera mucho tráfico en la red. Está reservado únicamente para direcciones Clase D, como veremos en el siguiente apartado. En el direccionamiento omnidireccional, hay un identificador para un conjunto de interfaces que, generalmente, pertenecen a nodos diferentes. Un paquete enviado a *una direccion de este tipo es entregado a todas las interfaces identificadas por esa* dirección, como se muestra en la figura 2.7

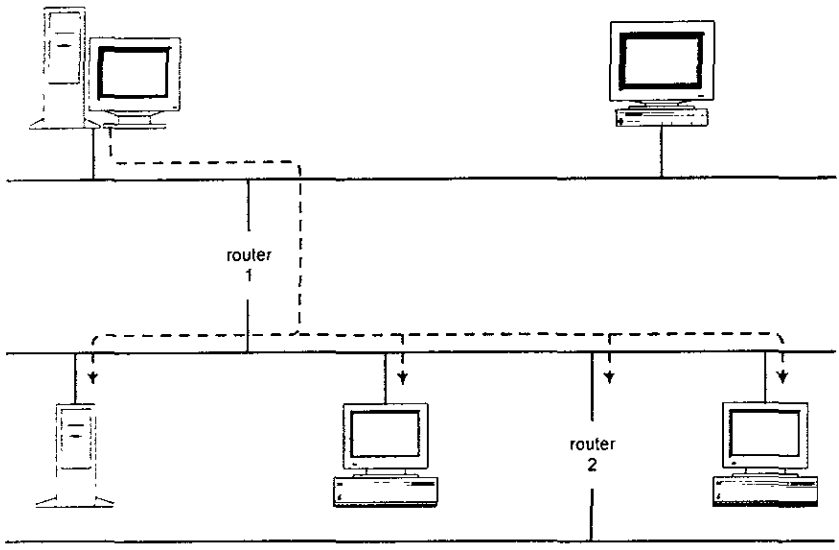


Figura 2.7 - *Direccionamiento Multidireccional*

2.4.2 Clases de Direcciones IP

Tomando tal cual está definida una dirección IP, podría surgir la duda de cómo identificar qué parte de la dirección se refiere a la red y qué parte al nodo en dicha red. Lo anterior se resuelve mediante la definición de las "Clases de Direcciones IP". Para clarificar esto, se puede ver que una red con dirección clase A queda precisamente definida con el primer octeto de la dirección, la clase B con los dos primeros y la C con los tres primeros octetos. Los octetos restantes definen los nodos en la red específica.

Todas las direcciones IP son escritas de la forma conocida como *notación decimal*, en la que cada octeto está dado como un número decimal del 0 al 255. En la tabla 2.3 presentamos la capacidad de cada una de las clases de redes, con los números máximos de redes y de nodos que pueden contener. El número máximo teórico ha sido reducido en dos considerando los valores reservados, cuando son todos ceros y todos unos. De esta manera el número máximo de redes de la Clase A es, *teóricamente, 128 ó* 2^7, *y el máximo* es 128 -2 = 126; mientras que el número de nodos es de 16,777,214 y el rango de direcciones va de la 1.0.0.0 a la 127.0.0.0.

Clases	Número de Redes	Número de Nodos	Rango de Direcciones IP
A	126	16,777,214	1.0.0.0 a la 127.0.0.0
B	16,382	65,534	128.0.0.0 a la 191.255 0.0
C	2,097,150	254	192 0.0 0 a la 223.255.255.0

Tabla 2.3 - *Capacidad de las Clases de Redes*

Las direcciones Clase A son identificadas por el primer bit, que es el 0. Los bits del 1 al 7 identifican la red y los bits del 8 al 31 identifican un nodo específico dentro de esa red. Las direcciones Clase A están diseñadas para redes muy grandes, que tienen muchos hosts. Con un identificador de red de siete bits (1 a 7), están disponibles únicamente 128 direcciones Clase A y de éstas, las direcciones 0 y 127 están reservadas.

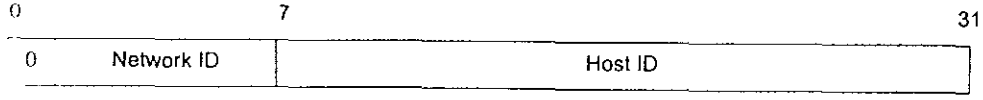
Las direcciones Clase B son identificadas por los primeros dos bits, con un valor de 10 (binario). Los siguientes 14 bits, identifican la red y los restantes 16 identifican el nodo. 16,384 direcciones Clase B están disponibles, con las direcciones 0 y 16,383 reservadas.

Las direcciones Clase C comienzan con un número binario igual a 110. Los siguientes *21 bits identifican la red, y los restantes 8 bits se emplean para identificar al nodo. Existe* un total de 2,097,152 direcciones Clase C posibles, estando reservadas las direcciones 0 y 2,097,151. Las direcciones Clase C son usadas generalmente para redes pequeñas, tales como LAN.

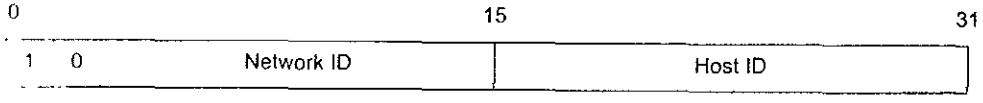
Las direcciones Clase D tienen los primeros bits con un valor binario de 1110 y se emplean para propósitos de direccionamiento omnidireccional (multicasting).

Si los cuatro primeros bits de una dirección son 1111, se trata de una dirección reservada. Esta clase de direcciones son llamadas Clase E, pero no se refieren, realmente, a una red específica. Están reservadas para uso futuro.

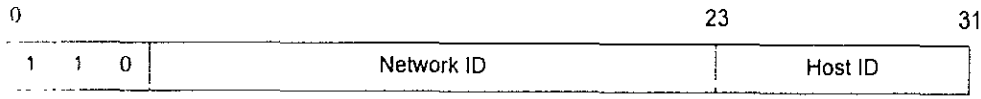
En la figura 2.8 presentamos un esquema de los formatos de las clases de direcciones IP mostrando los grupos de bits que se utilizan para identificar la clase de red, la red misma y los nodos.



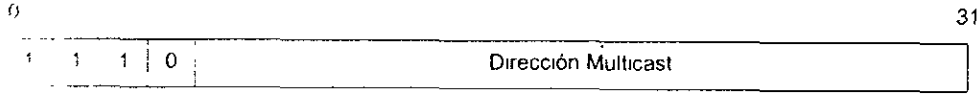
A) **Dirección Clase A**



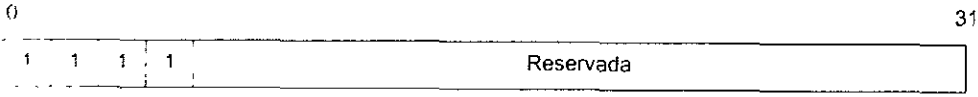
B) **Dirección Clase B**



C) **Dirección Clase C**



D) **Dirección Clase D**



E) **Dirección Clase E**

Figura 2.8 - *Clases de Direcciones IP*

2.4.3 Máscaras

Una dirección de red tiene asociada una máscara (un conjunto de bits) que indica los bits que se ocupan para la identificación de la red (y la subred en caso de que exista), y la identificación del nodo.

- La máscara tiene un conjunto de bits encendidos (unos) en forma ininterrumpida para identificar el número de bits que se emplean para la red y la subred.
- El resto de la máscara es un conjunto de bits apagados (ceros) en forma ininterrumpida que indica el número de bits que se utilizan para identificar el nodo.

Ejemplo

	Notación decimal	*Notación binaria*
Dirección IP	150.102.245.142	10010110.01100110.11110101.10001110
Máscara:	255.255.0.0	11111111.11111111.00000000.00000000

Cada clase de red tiene una máscara por default:

- Clase A 255.0.0.0
- Clase B 255 255.0.0
- Clase C 255.255.255.0

2.4.4 Subredes

La estructura de una dirección IP puede ser modificada localmente, usando algunos bits de redes adicionales que identifican a los nodos. Esencialmente, la línea divisoria entre los bits de la dirección de las redes y los bits de la dirección de los nodos, es movida creando redes adicionales, pero reduciendo el número máximo de nodos que puede contener cada red. Estos bits de la red reasignados, definen, dentro de una red grande a una red pequeña llamada subred.

Las subredes son para el mundo exterior, como una sola red; sin embargo, en forma interna son un conjunto de subredes que simplifican la administración y control, así como también reducen el tráfico entre ellas.

Las organizaciones deciden, generalmente, crear subredes para eliminar problemas topológicos u organizacionales Las subredes permiten descentralizar la administración del direccionamiento de los hosts. Con un esquema de direccionamiento estándar, un administrador central es el responsable del manejo de las direcciones de los hosts para la red completa Con las subredes, el administrador puede delegar la asignación de direcciones a organizaciones más pequeñas, dentro de la organización global.

Las subredes pueden ser usadas también para vencer diferencias de hardware y limitaciones debidas a las distancias. Los ruteadores IP pueden enlazar redes físicas diferentes, pero únicamente si cada red física tiene su propia dirección Las subredes dividen una dirección de red simple, en muchas direcciones de subredes únicas, de tal manera que cada red física tiene su propia dirección única

Algunas características de las subredes son las siguientes.

- Las Subredes son redes físicas distintas que comparten una misma dirección IP.
- Deben identificarse una de otra usando una máscara de subred.
- La máscara de subred es de cuatro bytes y para obtener el número de subred se *realiza una operación AND lógica entre ella y la dirección IP de algún equipo.*
- La máscara de subred deberá ser la misma para todos los equipos de la red IP.

Como sabemos, el enrutamiento sirve para alcanzar redes distantes, también sabemos que las direcciones IP se agrupan en clases Ahora bien para cada clase se puede contar con un número determinado de subredes. Las subredes son redes físicas independientes que comparten la misma dirección IP (es decir, aquella que identifica a la red principal). La pregunta entonces es ¿cómo se logra que equipos que comparten el mismo identificador de red pero que se sitúan en redes físicas diferentes puedan comunicarse usando compuertas? La solución a este problema es determinando una mascara de dirección.

Máscaras de subredes

Una subred está definida por el cambio de la máscara de una dirección IP. Una mascara de subred funciona de la misma manera que una máscara normal: un bit "encendido" es interpretado como un bit de red; un bit "apagado" pertenece a la parte del host de la dirección. La diferencia estriba en que la máscara de subred sólo es usada

localmente. En el mundo exterior, la dirección sigue siendo interpretada como una dirección IP estándar

Cuando se especifican subredes se ocupan más bits para la identificación de la red y la subred que en la máscara por default.

Al utilizar algunos bits para la identificación de la subred, se reduce el número de bits para la identificacion del nodo, y por lo tanto, se reduce el número de nodos que pueden existir por cada subred

Para realizar la transferencia de información utilizando subredes, se debe ejecutar la función AND que extrae los campos de la dirección IP de la siguiente manera: la función AND se ejecuta en la dirección IP y en la máscara de subred, el resultado de esta operacion es comparado con la dirección destino en una tabla de ruteo y si los resultados son iguales, la siguiente dirección IP (relativa a la dirección destino) es usada para determinar el próximo salto en la ruta

La mascara se convierte en una parte del enunciado condicional del algoritmo de ruteo. "*Si la direccion IP destino y la máscara de subred son iguales a mi dirección IP y a mi mascara de subred, entonces se envía el datagrama a una red local, de lo contrario, se* envia el datagrama al gateway correspondiente a la dirección destino". Ciertamente, el uso de mascaras manipula las rutas hacia convenciones directas, rutas específicas de hosts y rutas por default

Para implementar una subred, se deben tener en cuenta los siguientes puntos:

- El algoritmo IP debe ser implementado en todas las máquinas que existan dentro de una subred
- Las máscaras de subred deben ser las mismas para todas las máquinas.
- Si una o más máquinas no soportan máscaras, se debe emplear el proxy ARP para poder implementar una subred.

Ejemplo

- Supóngase que la dirección IP de una equipo es 148.206. 247.2
- La máscara de subred es 255 255 255.0

Si los 3 tres octetos de orden superior definen la subred, entonces el equipo se encuentra dentro de la subred 148.206 247.0

2.5 SISTEMA DE NOMBRES DE DOMINIO

2.5.1 Historia

Aunque las direcciones IP proveen una representación conveniente y compacta, para especificar la fuente y el destino de los paquetes que se envían a través de Internet, los usuarios prefieren identificar a las máquinas con nombres fáciles de recordar. Aunque es necesario proveer una forma de convertir los nombres a direcciones numéricas

En un principio, cuando la red era muy pequeña, hallar un nombre era fácil. El Centro de Información de la Red, NIC (*Network Information Center*), estableció un registro manteniendo un archivo de nombres y direcciones llamado archivo hosts, el cual se

distribuia a cada computadora en la red. Los nombres eran simples palabras, cada una escogida de manera que fuera única.

Desafortunadamente, a medida que iba creciendo la red, también iba creciendo el archivo Existían muchas dificultades con los nombres, además, se utilizaba mucho tiempo de red en distribuir ese enorme archivo a cada máquina contenida en él. Era obvio que se necesitaba una base de datos distribuida alrededor del mundo para administrar los nombres y direcciones y además proveer la traducción de un nombre a una dirección IP y viceversa, este sistema se llama Sistema de Nombres de Dominio, DNS (*Domain Name System*)

Este sistema nació en la década de los 80's, creado por Paul Mockapetris en colaboración con Jon Postel, de la Universidad del Sur de California y posteriormente Paul Vixie Juntos desarrollaron lo que hasta ahora conocemos como el DNS, un sistema cliente/servidor, distribuido y jerárquico, con características muy parecidas a un sistema de archivos de UNIX, pero distribuido.

Originalmente, el uso del DNS involucró solamente instituciones académicas, de investigacion y por supuesto, la milicia de los EEUU. Eran los tiempos en que las universidades empezaban a realizar su conexión a las múltiples redes, entre ellas BitNet. Algo empezaba a trascender y era importante establecer un orden en cuanto a los equipos que ingresaban a la red.

Se crearon entonces los nombres de dominio genéricos de primer nivel, gTLD (*generic Top-level Domain*), .com, .net y .org, es decir, se habían creado estos tres nombres con el fin de ubicar el tipo de entidades que buscaban tener presencia en Internet. Además de estos gTLD se empezó por delegar los sufijos nacionales, nTLD (*national Top-level Domain*) a los países que se fueran conectando a la red. De esta forma, a México se le asigno el .mx a finales de 1988, cuando el ITESM Campus Monterrey se conectó de manera dedicada a Internet. Este nTLD empezó a operar desde el 1 de Febrero de 1989. Así, cada país obtuvo su propio nTLD, incluso EEUU, el cual tiene el .us También existen unos nombres de dominio especiales sTLD, que son sólo para los EEUU: .mil, .edu y .gov

Las organizaciones que administran los nTLD son, por lo general, instituciones académicas, como es el caso del .mx y el ITESM. Sin embargo, el caso de los gTLD es diferente porque originalmente fueron administrados por el Stanford Research Institute Network Information Center (SRI-NIC), de la Universidad de Stanford en Menlo Park, California, pero pronto cambiaría a InterNIC.

En 1992, la Fundación Nacional de Ciencias de los EEUU, NSF (*National Science Foundation*), quien administraba el backbone de Internet (en ese entonces NSFNET), decide licitar la operación del InterNIC y en 1993, a través de un convenio de cooperación le otorga esta función a la empresa Network Solutions Inc (NSI). No obstante, en 1994, el grupo SAIC compra esta empresa y su experiencia en contratos federales le ayuda a re-negociar el contrato previo. De esta forma, logra que se empiece a cobrar $50 USD anuales por cada nombre de dominio, estableciendo que el 30% de estas cuotas se irían a un fondo de infraestructura administrado por la NSF.

Para mediados de 1996, Jon Postel, el director del Internet Assigned Numbers Authority (IANA), organismo administrador de las direcciones de IP y nombres de dominio, realizó una propuesta en la que contemplaba la creación de 150 nuevos nombres de dominios genericos gTLD, así como el .com, .net y .org. De esta forma, en Noviembre de 1996 nació el Internet-International Ad Hoc Committee (IAHC), impulsado por la Internet Society (ISOC), el cual generóun reporte final con el nombre de "Memorando de Entendimiento para los Nombres de Dominio genéricos de Nivel Superior".

El IAHC se disolvio en 1997, para dar paso al generic Top level Domain Memorandum of Understanding (gTLD-MoU), documento respaldado por organizaciones de todo el mundo, entre ellas la Organización Mundial de la Propiedad Industrial (WIPO), Union Internacional de Telecomunicaciones (ITU), Internet Society , MCI y por Latino America, sólo NIC-México. En este documento se contemplaba la propuesta de un esquema que permitiera cambiar de registro, es decir, portabilidad de los nombres de dominio Esto aseguraba según el gTLD-MoU, que todos los registros dieran un servicio de calidad

Esta propuesta necesitaba un nuevo esquema distribuido de DNS (new DNS Shared Registry System), que incluyera siete nuevos nombres de dominio genéricos: .firm, .store, web, arts, rec, .info, nom

A finales de enero de 1998, el gobierno de los Estados Unidos publicó un documento, conocido como Green Paper, en el que se desconocía la autoridad y el consenso del gTLD-MoU y por lo tanto de las organizaciones que lo representaban. Debido a la cantidad tan grande de comentarios en contra de este documento, en junio de 1998, el gobierno a través de la Cámara de Comercio, emitió otro documento conocido como White Paper, en el cual se presentaban los planteamientos finales para realizar la transicion en la administración de Internet

DNS

El Servidor de Nombres de Dominio hace referencia al servicio que permite la traduccion de direcciones textuales a direcciones IP. En Internet, las redes que la forman, así como las máquinas conectadas a las mismas, están identificadas por una dirección, que consiste en 4 números separados por puntos y que se denomina dirección IP. Esta direccion nos permitirá identificar univocamente cualquier recurso dentro de Internet.

Por ejemplo, la dirección IP 132.248 159.41 corresponde al dominio aketzali super unam mx, que es un servidor de la DGSCA, en la UNAM Gracias a este mecanismo de direcciones es posible no sólo identificar, sino también localizar cualquier red subred y máquina dentro del universo de Internet. Estos números son fáciles de manejar y de procesar por una computadora, pero difíciles de recordar y poco *significativos para las personas, ya que nos resulta mucho más sencillo tratar con* direcciones textuales, con algún significado relacionado al servicio que se está usando.

En el lenguaje de Internet, se denomina "resolución directa" a la traducción de una direccion textual a su correspondiente dirección IP. La operación contraria se denomina "resolucion inversa".

DNS supera las dos debilidades principales de las tablas del host:

- DNS no se atiene a una simple tabla grande; es un sistema de datos distribuidos que no se bloquea con el crecimiento de la base de datos y que proporciona actualmente, información sobre 16 millones de hosts aproximadamente, mientras que en la tabla del host únicamente están enlistados 10 mil.
- DNS garantiza que la nueva información del host será diseminada al resto de *la red, según sea requerido.*

La información es automáticamente difundida, pero únicamente a aquéllos que están interesados en ella. La manera en como trabaja DNS es la siguiente: si un servidor DNS recibe un requerimiento de información acerca de un host sobre el cual no tiene ningún dato, transfiere el requerimiento a un *servidor autoritario*. Un servidor autoritario es

cualquier servidor responsable de mantener la información correcta acerca de los dominios requeridos. Cuando este servidor responde, el servidor local guarda la respuesta (cache) para uso futuro. La siguiente vez que el servidor local reciba un requerimiento de esta información, responderá por si mismo sin necesidad de consultar otro servidor. La habilidad de controlar la información del host desde un servidor autoritario y de difundir automáticamente la información correcta, hace a DNS superior a la tabla del host, aún para redes no conectadas a Internet.

2.5.2 La jerarquía de dominios

DNS es un sistema jerárquico distribuido para la resolución de nombres de hosts en direcciones IP. Bajo DNS, no existe una base de datos central con toda la información de los hosts de Internet. La información es distribuida entre miles de servidores de nombres organizados jerárquicamente, de forma similar a los archivos de sistema de Unix. DNS tiene un dominio raiz (*root domain*), ubicado en la parte superior de la jerarquía de dominios y que es atendido por un grupo de servidores de nombres, llamados *servidores raiz*.

La información sobre un dominio es encontrada a través de apuntadores, desde el dominio raiz, pasando por los dominios subordinados, hasta el dominio final. Directamente bajo el dominio raíz, se encuentran los dominios de primer nivel (*top-level domains*). Existen dos tipos básicos de dominios de primer nivel: los geográficos y los organizacionales. En Estados Unidos, los dominios de primer nivel más populares son los organizacionales; esto es, una agrupación en un dominio, está basada en el tipo de organización (comercial, militar, etc.) a la cual el sistema pertenece. Los dominios de primer nivel usados en los Estados Unidos son:

com	una compañía, institución u organización comercial
edu	una institución educativa
gov	una empresa de gobierno
mil	una organización militar
net	servidores administrativos de una red
int	una organización gubernamental internacional
org	organizaciones privadas que no caen dentro de las otras clases.

Por otro lado, los dominios geográficos han sido establecidos para cada país del mundo y son identificados por un código de dos letras, por ejemplo

au	Australia
ca	Canadá
fr	Francia
uk	Reino Unido
mx	México

Estos, además, pueden tener subdominios. Por ejemplo: ac.uk para instituciones académicas y co.uk para las comerciales. Cabe mencionar que Estados Unidos tiene su propio codigo de pais, sin embargo, no se usa mucho. Muchas redes en los Estados Unidos usan dominios "organizacionales" como edu, en lugar de dominios "geográficos" como va.us - Virginia.

Se han hecho numerosas propuestas para incrementar el número de dominios de primer nivel Los dominios propuestos son llamados "dominios genéricos de primer nivel" o gTLD Las propuestas sugieren la creación de dominios de primer nivel adicionales y de nuevos registros para su administración. Todos los dominios actuales son manejados por un registro simple InterNIC. Una motivación para esta propuesta es el enorme tamaño del dominio *com*, es tan grande, que se dificulta el mantenimiento de una base de datos. Por ello, se propusieron los siguientes dominios genéricos de primer nivel:

firm	negocios o firmas
store	tiendas y negocios de ventas de bienes
web	organizaciones del World Wide Web
arts	organizaciones culturales y de entretenimiento
rec	organizaciones recreativas y de entretenimiento
info	sitios que proporcionan servicios de información
nom	individuales u organizaciones que desean definir una nomenclatura personal

La figura 2 9 ilustra la jerarquía de dominios, usando los dominios de primer nivel organizacionales En la parte superior está el dominio raiz, abajo se encuentran los dominios de primer nivel. Los servidores raíz tienen información completa únicamente sobre estos dominios No existen servidores que tengan información completa sobre todos los dominios, pero los servidores raíz tienen apuntadores para los dominios de segundo nivel

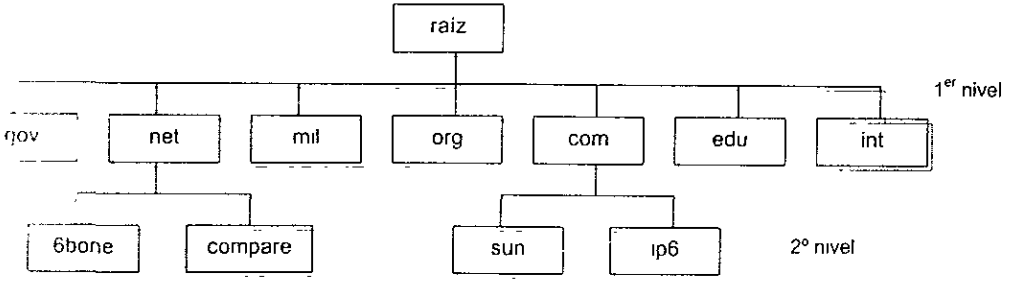


Figura 2.9 - *Jerarquía de Dominios*

2.5.3 Creación de Dominios y Subdominios

La finalidad del DNS es la de permitir el crecimiento, tanto administrativo como técnico, del sistema de nombres de Internet, por medio de una distribución jerárquica de dominios asignados Los dominios son entidades administrativas cuyo propósito es subdividir la carga de gestión de un administrador central repartiéndola entre distintos

subadministradores. Estos, a su vez, pueden repetir el proceso, si el tamaño del dominio a administrar así lo aconseja. De esta forma, se pueden crear distintos niveles de dominios asignados, donde cada administrador asigna nombres unívocos a su nivel, garantizando así que sea único cualquier nombre del DNS que se forma juntando los distintos nombres de dominio (separados por puntos "."), de abajo a arriba en la jerarquía hasta llegar al último, denominado raíz del DNS o "."

Para obtener un dominio, se requiere la autorización del NIC, para crear un dominio de segundo nivel bajo uno de los dominios de primer nivel. Una vez que la autorización es concedida se pueden crear dominios adicionales (subdominios), bajo el primer dominio. Esto es, cada grupo puede crear o cambiar lo que está dentro de su dominio. Si un grupo decide crear un subgrupo, podría hacerlo sin necesidad de pedirle permiso a nadie. Todo que tiene que hacer es añadir el nuevo nombre a su parte de la base de datos, y cada persona que lo necesite, encontrará el nuevo grupo Si cada grupo sigue las reglas y se asegura de que los nombres que asigna son únicos, entonces no habrá dos sistemas con el mismo nombre en todo Internet. Podrían haber dos máquinas con el mismo nombre, pero solamente si estan en diferentes dominios

Nombres de Dominios

Los nombres de dominio reflejan la jerarquía de dominios. Estos nombres son escritos del mas especifico (nombre del host) hasta el menos especifico (dominio de primer nivel), en el cual, cada parte está separada por un punto Un nombre de dominio totalmente calificado, FQDN (*fully qualified domain name*) empieza con un host específico y termina con un dominio de primer nivel. Por ejemplo, *servidor.unam.mx* es el FQDN de la estación de trabajo *servidor*, dentro del dominio *unam*, que pertenece al dominio geográfico *mx*.

Los nombres de dominios no son siempre escritos como totalmente calificados. Tambien pueden ser escritos como relativos al dominio por default DNS añade el dominio por default cuando el usuario elabora el requerimiento para un servidor de nombres. Por ejemplo, si el dominio por default es *unam mx*, el usuario puede omitir la extensión *unam mx* para cualquier nombre de host dentro de ese dominio.

La manera como será usado el dominio por default y cómo serán elaborados los requerimientos, depende de la implementación del software. Algunas veces se añade la extension a cualquier nombre de host, a menos que termine en punto (lo cual indica que esta fuera de la raíz). Otras veces, añade la extensión únicamente si no existe algún punto en el nombre del host requerido.

2.5.4 Resolución de Nombres de Dominios

Para determinar nombres amigables para los usuarios en direcciones IP, se debe trabajar con el concepto de resolución de nombre de dominio. Afortunadamente, la tarea del usuario es muy simple para determinar (resolver) estos nombres. El usuario necesita únicamente proporcionar un conjunto de argumentos a un agente local, llamado resolvedor de nombres (*name resolver*), el cual recupera la información basada en un nombre de dominio o bien, transfiere la petición a un servidor de nombres (*name server*). El usuario únicamente debe elaborar el requerimiento adecuado y proporcionar ciertas

indicaciones de cómo será ejecutada la operación. La figura 2 10 muestra la estructura de la resolución de nombres de dominios

El usuario ve a la estructura de dominios como un simple nombre (un simple espacio de información) Por su parte, el resolvedor asume la tarea de resolver el nombre, o de enviarlo a sistemas cooperativos independientes (los servidores de nombres) para la resolucion nombre / dirección. Como se muestra en la figura 2.10, el servidor de nombres atiende un requerimiento del resolvedor de nombres. Así, el resolvedor actúa como un proveedor de servicio para el programa de usuario Por su parte, el resolvedor actúa como usuario del servidor de nombres.

El servidor de nombres puede almacenar una parte de la misma información contenida en el resolvedor de nombres, para proporcionar mayor eficiencia y respaldo A pesar de que la información es almacenada en el servidor de nombres, el resolvedor debe conocer el nombre de al menos un servidor, para iniciar el requerimiento. Este requerimiento es pasado al servidor de nombres desde el resolvedor para que proporcione una respuesta o haga una referencia a otro servidor de nombres. Con esta característica, el resolvedor puede conocer más acerca de la identidad de otros servidores de nombres y de la información que él maneja.

La determinación de los servidores de nombres que participarán en la operación de resolucion, está basada en la jerarquía de dominios. Cada parte del árbol jerárquico corresponde a un servidor de nombres Cada servidor en un subdominio, sabe qué otros servidores están bajo su dominio y puede escoger el servidor adecuado para responder un requerimiento

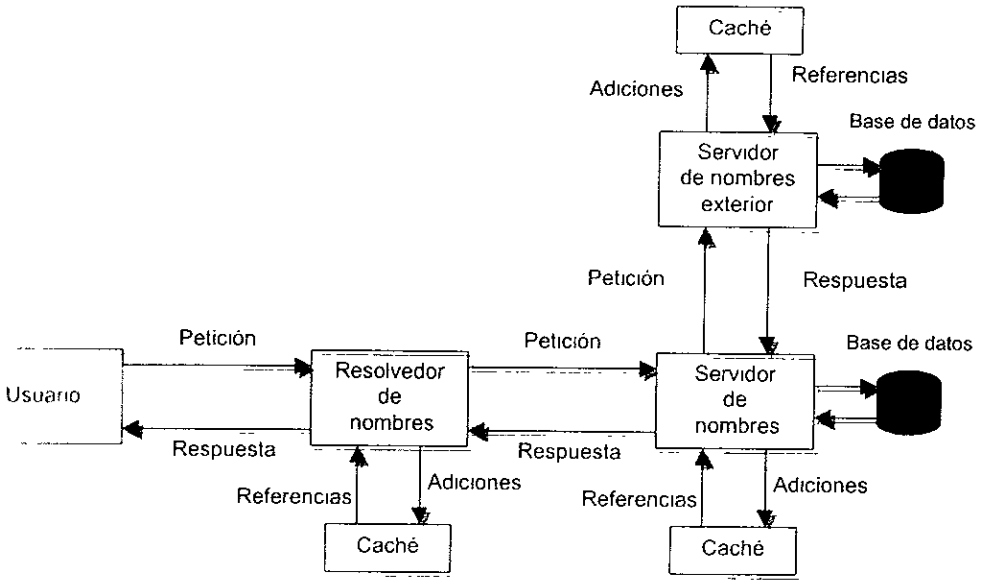


Figura 2.10 - *Resolución de Nombres de Dominios*

En la figura 2 10 también aparece otro elemento incluido en el servidor, llamado *cache de nombres*. Cuando se recibe un requerimiento, el servidor verifica este almacenamiento local para ver si la respuesta está disponible localmente; si es así, la respuesta es enviada al cliente; si la respuesta no está disponible en el caché, el servidor debe determinar cuáles son los mejores servidores de nombres para proporcionar la respuesta

El caché de nombres generalmente está incompleto, pero proporciona la información requerida más frecuentemente, para agilizar el proceso de resolución de nombres. La información del cache es borrada eventualmente, por medio del uso de contadores.

Operaciones del servidor de nombres

Existen dos tipos de requerimientos: los no-recursivos y los recursivos En un requerimiento no-recursivo, el servidor de nombres manda una de las siguientes contestaciones al servidor local la respuesta, la identificación de un error o una referencia hacia otro servidor. El servidor local debe seguir los apuntadores por sí mismo. En un requerimiento recursivo, el servidor de nombres sigue los apuntadores hasta resolver la petición y regresa la respuesta final al servidor local. Si el servidor de nombres no retorna la direccion IP requerida, manda, entonces, una respuesta negativa, ya que no le está permitido retornar una referencia. Este tipo de búsqueda reduce la carga de trabajo del host del usuario, pero incrementa la del servidor remoto El servidor raíz, generalmente ejecuta sólo búsquedas no-recursivas

El efecto de los requerimientos no-recursivos y recursivos, asegura que el usuario sepa que, al menos un servidor, está en la dirección requerida. También asegura que el servidor de nombres conozca la dirección IP de, al menos, otro servidor de nombres.

El servidor es el responsable del mantenimiento de una porción del árbol, llamada *zona*, que es una sección contigua del espacio de dominios y típicamente, existe una base de datos separada para cada zona. El servidor de nombres tiene como función, verificar periódicamente que la información de su zona sea correcta y si no, actualizarla. Una zona puede ser actualizada únicamente por la autoridad apropiada El servidor de nombres emplea un Protocolo de Transferencia de Zona (*Zone Transfer Protocol*) para permitir que más de un servidor almacene datos sobre una zona.

Si el servidor de nombres de un dominio falla por cualquier razón, deben estar disponibles copias redundantes de la información de direccionamiento y de nombres, en otros servidores de nombres. Un servidor de nombres es clasificado como *primario*, o bien, *secundario* La función de un servidor de nombres primario puede ser duplicada en otras maquinas, que en ese momento son llamadas servidores de nombres secundarios. Esta caracteristica proporciona fiabilidad y eficiencia en la atención de los requerimientos.

Los mensajes de petición y de respuesta, transmitidos entre dos servidores de nombres pueden usar tanto el Protocolo de Control de Transmisión (TCP), como el Protocolo de Datagrama de Usuario (UDP). Generalmente es usado el protocolo no orientado a conexión UDP, por permitir un mejor desempeño En cambio, para actividades que requieren la actualización de bases de datos, como las operaciones de actualización (refresco) de zona, es preferible TCP, para obtener una transferencia confiable; de cualquier manera, los servidores de nombres pueden usar ambos protocolos.

2.6 SEGURIDAD EN IPv4

2.6.1 Introducción

"Los ataques a maquinas conectadas a Internet se incrementaron en un 260% desde 1994, se calcula una perdida de 1 290 millones de dólares anuales sólo en los EEUU".
NCSA (National Computer Security Agency), New York 1996

En la era de la informática, las ideas, los datos y los archivos en una red, son probablemente lo más valioso que un usuario posee Se puede pensar, por ejemplo, en una empresa que tiene listas de clientes y registros de accionistas, transacciones comerciales y material de marketing, estrategias de comercialización y diseño de productos, etc., que debe proteger a toda costa, para lo cual requiere de sistemas de seguridad que le garanticen la integridad de su información.

Es decir, todas las redes requieren de diferentes niveles de seguridad y la arquitectura de seguridad del Modelo de Referencia para Interconexión de Sistemas Abiertos ISO 7498/2 define cinco servicios de seguridad autenticación, control de acceso, confidencialidad de los datos, integridad de los datos y no-repudiación. Estos servicios se proporcionan por los niveles superiores, a través de la apropiada aplicación de uno o más mecanismos de seguridad Se identifican ocho mecanismos de seguridad específicos. cifrado, firma digital, control de acceso, integridad de los datos, intercambio de autenticación, protección del tráfico (*traffic padding*), control de enrutamiento y notarización, y cinco mecanismos de seguridad generales: funcionalidad fiable, etiquetas de seguridad, detección de eventos, auditoría de seguridad y recuperación de la seguridad

Los mecanismos de seguridad deben satisfacer los siguientes requisitos:

- Proporcionar servicios criptográficos (codificación) de seguridad.
- No interferir con el funcionamiento de sistemas no protegidos
- Soportar modos de operación transparentes en los sistemas protegidos.
- Proporcionar comunicación opcional con sistemas no protegidos.

TCP/IP no incluye seguridad dentro de su modelo, con excepción de lo que se refiere a la integridad de los datos en el campo de checksum. TCP/IP deja a los niveles superiores la tarea de especificar el nivel de seguridad deseado para la transmisión de los datos, de acuerdo con las características de la red.

Los diversos mecanismos de seguridad que existen son, todos ellos, ajenos al modelo TCP/IP en si mismo; estos mecanismos operan en las capas superiores de aplicación y de transporte

La información que está almacenada o que fluye en una red debe tener fácil acceso e intercambio, para que pueda ser utilizada eficientemente. Pero es aquí donde se plantea el dilema de qué es preferible· accesibilidad o seguridad. Los mecanismos de seguridad deben brindar la posibilidad de manejar libremente la información, sabiendo que está debidamente protegida

A continuación mencionaremos el servicio de autenticación, debido a que es el empleado por la mayoría de los sistemas actuales de comunicación, los demás servicios de seguridad son manejados en niveles superiores y no son manejados por el nivel en el que se encuentra el protocolo IPv4, por lo que solamente se hace mención de ellos.

2.6.2 Autenticación

En lo que se refiere a la autenticación, hay que tener en cuenta que la mayoría de los sistemas actuales de computadoras proporcionan "autenticación en un solo sentido"; es decir, el usuario que desee tener acceso a una computadora, primero debe probar su identidad, de manera que los mecanismos de control puedan decidir sobre lo que el usuario esta autorizado o no a realizar. Con el gran desarrollo de las redes, más y más instalaciones, aplicaciones y usuarios demandan una "autenticación en doble sentido", donde las partes involucradas en la comunicación deben probar sus respectivas identidades ante la otra parte

Ademas de la exigencia de la autenticación en doble sentido, la rápida expansión de las redes ha dado lugar a la necesidad de autenticación, de un tercer elemento fiable. A medida que los usuarios utilizan más servicios de red, a veces simultáneamente, no desean tener que identificarse una y otra vez para cada servicio individual o programa de aplicacion con el que se comuniquen. En su lugar, es deseable poder utilizar cualquier servicio de la red a través de un único procedimiento de registro ante una autoridad reconocida por toda la red, llamado servicio de autenticación Este servicio emite *tokens* (tambien llamados *tickets*, certificados o credenciales), reconocidos por todos los servicios de red como garantías fiables y no falsificables de la identidad del usuario.

Las técnicas de firma digital garantizan la autenticación en un solo sentido, como la integridad de los mensajes firmados. Se asegura que el mensaje procede de quien lo firmó y que, una vez firmado, no puede ser cambiado por el emisor, un posible interceptor o por el destinatario, sin que el intento de fraude sea detectado.

Además de ofrecer la facilidad de firma digital, un servicio de autenticación puede ofrecer también otras facilidades, como son la notarización y el no-repudio. La *notarización* es el registro de la información (mensajes) por el servicio de autenticación, de forma que éste pueda ser consultado posteriormente para atestiguar la exactitud de las características del mensaje, tales como el contenido, origen, destino, fecha de emisión, etc El *no-repudio* es una forma particular de notarización que permite a cualquier usuario, emisor, receptor o un tercero, obtener una prueba de que el mensaje fue correctamente enviado, como en el caso del correo certificado. Estas facilidades desempeñan un papel importante en los aspectos legales de los sistemas de información utilizados para realizar transacciones comerciales, como el Intercambio de Datos Electrónicos, EDI (*Electronic Data Interchange*)

Ahora mencionaremos un elemento que es muy importante en la seguridad de las redes y que proporciona algunos de los mecanismos de seguridad, tales como control de acceso, protección del tráfico, control de enrutamiento e intercambio de autenticación.

Firewall

Cuando se tiene una red interna conectada a Internet o a una Intranet corporativa, se necesita un mecanismo de seguridad, tal como un firewall, para mantener las normas de seguridad entre ellas El firewall mantiene separada la red interna (de la cual se tiene control) de diferentes tipos de redes externas (de las cuales no se tiene control). El firewall controla la entrada y salida de tráfico, protegiendo la red de intromisiones no deseadas La función del firewall es la de ser una sólida barrera entre la red y el mundo exterior Este permite habilitar el acceso a usuarios (modulo optativo) y servicios aprobados

Algunas de las prestaciones que proporciona un firewall, son:

- Prevenir que usuarios no autorizados obtengan acceso a la red.
- Proveer acceso transparente hacia Internet a los usuarios habilitados
- Asegurar que los datos privados sean transferidos en forma segura por la red pública
- Ayudar a los administradores a buscar y reparar problemas de seguridad.
- Proveer un sistema de alarmas advirtiendo intentos de intromisión a la red.

Algunas de las características técnicas de un firewall son las siguientes.

- Tiene dos tipos de configuración: local y remota.
- Configuración remota por medio de una interface gráfica (GUI)
- Configuración local por medio de una interface que se utiliza desde la consola del firewall
- Permite el uso de aplicaciones que se utilizan en la medición de tiempos de conexión y uso de servicios.
- Conexiones de todos los servicios comunes de TCP/IP a través del firewall, de manera totalmente transparente.
- Soporta servicios multimedia
- Auto configuración de servidores que proveen servicios hacia el exterior de la red interna por medio de normas de seguridad.
- Múltiples alarmas de intentos de ingreso fallidos hacia la red.
- Filtro de acceso de conexiones permitidas por interfaces no permitidas. Este filtro es importante para contrarrestar técnicas de piratería (*spoofing*).
- La configuración del firewall se puede hacer mediante el mismo servidor o desde un servidor remoto, corriendo un sistema de administración específico que utiliza para esta tarea, una interface dedicada o *tunneling* (comunicación encriptada).

La figura 2 11 muestra el diagrama de conexión de un firewall, donde se puede ver que actua como unión y a la vez como filtro entre los usuarios y los sistemas de información, aplicando mecanismos de seguridad para permitir únicamente el acceso a determinados usuarios

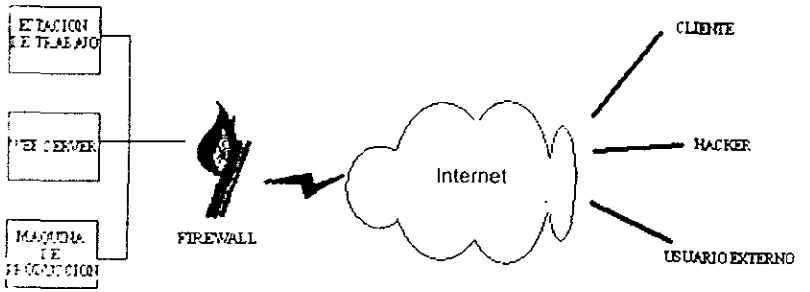


Figura 2.11 - *Conexión de un firewall*

Los firewalls representan un componente muy importante en la seguridad de las redes, pero en muchas ocasiones se presupone que se tiene una seguridad adecuada, tan solo por emplear un firewall. Sin embargo, un firewall es tan bueno como su configuración. Múltiples reglas para la filtración de paquetes y proxies[1] para servicios distintos pueden ser mal configurados, creando otros riesgos. La configuración de un firewall, su integridad y durabilidad, deben ser examinadas periódicamente para confirmar que esté proporcionando el nivel de seguridad esperado

[1] Un *Proxy* es un servidor intermedio que sirve de puente entre una computadora y el servidor Web al que queremos acceder. El proxy almacena en su caché (en su disco duro) las páginas solicitadas por los clientes, de modo que si un usuario solicita una página que ya ha sido visitada por otro usuario, el proxy le devuelve la información contenida en su caché. El resultado es una mayor velocidad en el acceso a páginas Web, al obtener la información de un lugar más cercano. El proxy dispone de mecanismos para que la información que contiene esté permanentemente actualizada

Capítulo III

TRANSICION ENTRE IPv4 E IPv6

3.1 EL SURGIMIENTO DE IPv6

El desarrollo del nuevo estándar IP (*Internet Protocol*) empezó en 1992, seguido del primer congreso de la Internet Society, en Boston. Entre 1992 y 1994, surgieron varias propuestas que pretendían situarse como estándares, para reemplazar al cada vez más insuficiente IPv4

El Protocolo Internet versión 6 (IPv6, *Internet Protocol version 6*) es la designación de la nueva versión del estándar IP, y fue conocido previamente como Protocolo Internet de próxima generación (IPng, *IP next generation*) Existen varias razones para el desarrollo de un nuevo estándar IP, la más importante de las cuales fue el agotamiento del espacio de direcciones de 32 bits que proporciona el estándar actual IPv4 (*Internet Protocol version 4*). Aunque un espacio de direcciones de 32 bits provee alrededor de 4 billones de direcciones, sin tomar en cuenta el rango de direcciones reservado, ya se están agotando, debido, principalmente, a la mala distribución de direcciones que se hizo desde un principio (por ejemplo, una dirección clase A tiene demasiadas direcciones para una sola entidad). Debido a todo esto, se pronostica que para principios del próximo milenio ya no existirán direcciones IPv4 disponibles, ya que la escasez es cada vez mayor con el estandar actual.

Otro factor muy importante que contribuyó a la propuesta de desarrollar un nuevo estandar, es la necesidad cada vez mayor del soporte de nuevas aplicaciones, tales como audio y video en tiempo real. Con el nuevo protocolo IPv6 se pretende dar solución a estas necesidades y a otros aspectos problemáticos de IPv4, como son los campos de las cabeceras, que ocupan mucho espacio y que no son imprescindibles, y el manejo de tablas de ruteo.

A continuación, se presentan algunas de las propuestas más importantes.

TUBA (TCP and UDP over Bigger Address)

La propuesta conocida como TUBA sugería adoptar el protocolo ISO/OSI 8473 CLNP, para sustituir a IPv4, tratando, con esto, de fusionar drásticamente el mundo OSI y el mundo de Internet (TCP/IP). La propuesta podría haber permitido tener una plataforma común con OSI y con TCP.

En los dos primeros años, la propuesta de TUBA fue discutida y analizada, aceptándola con la propuesta original de CLNP, y rechazándola después, por no incorporar aspectos innovadores como el direccionamiento multicast, la movilidad, la Calidad de Servicio (QoS, *Quality of Service*) y por razones de incompatibilidad con la base instalada de OSI (que fue de importancia secundaria). Por esto las propuestas de TUBA y de OSI CLNP se designaron incompatibles, considerándolas fallidas

IPv7, TP/IX, CATNIP

En 1992, Robert Ullman lanzó la propuesta de un nuevo protocolo, llamado IPv7. La propuesta fue reelaborada en 1993 y asume el nombre de TP/IX para indicar el cambio del protocolo IP y el protocolo TCP al mismo tiempo. La propuesta contenía ideas interesantes para el procesamiento rápido de los paquetes y un nuevo protocolo de ruteo llamado RAP En 1994, la propuesta considerada como una nueva evolución, trataba de definir un formato único para los paquetes de IP, CLNP e IPX, y asumir el nuevo nombre CATNIP CATNIP podría haber sido una plataforma común, ya que soportaba varios protocolos de transporte, como OSI, TCP, UDP y SPX

IP en IP, IPAE (IP Address Encapsulation)

La propuesta de IP en IP, elaborada en 1992, fue diseñada considerando la utilización de dos capas de IPv4, para solucionar la falta de direcciones del nivel de Internet. Una capa se usaría para implementar el backbone mundial y la otra dentro de las áreas de frontera En 1993, se desarrolló otra propuesta, llamada Encapsulación de Direcciones IP (IPAE, *IP Address Encapsulation*) y fue aceptada como solución para la transición hacia SIP

SIP (Simple IP)

La propuesta conocida como SIP (*Simple IP*), fue realizada por Steve Deering, en noviembre de 1992 Esta se basa en la idea de incrementar las direcciones IP hasta 64 bits y eliminar los detalles obsoletos de IP. Esta propuesta fue aceptada inmediatamente por muchas compañías, quienes apreciaron su simplicidad.

PIP (Paul`s Internet Protocol)

La propuesta del PIP fue realizada por Paul Francis, quien introdujo innovaciones significantes sobre el ruteo, permitiendo una política eficiente de ruteo y una buena implementación de movilidad. En 1993, la propuesta del PIP se unió con la propuesta del SIP, dando origen al SIPP.

SIPP (Simple IP Plus)

Esta propuesta trata de combinar la simplicidad de implementación del SIP y la flexibilidad de ruteo del PIP SIPP fue diseñado para trabajar eficientemente tanto en redes de alto rendimiento, por ejemplo ATM, como también en redes de bajo rendimiento, como las redes inalámbricas. SIPP tiene un tamaño pequeño de cabecera y una dirección de 64 bits.

Con SIPP, la cabecera es elaborada eficientemente por ruteadores y pueden ser extendidas para insertar nuevas opciones en el futuro.

Para lograr lo que hoy es IPv6 se tuvo que realizar una comparación de las propuestas antes mencionadas y otras más, para obtener un protocolo estándar, considerando más las propuestas CATNIP, SIPP y TUBA, para elegir cuál se retomaría para realizar el estándar.

A continuación se muestra la tabla 3 1, en donde se evalúan las propuestas CATNIP, SIPP y TUBA.

Características	CATNIP	SIPP	TUBA
Especificaciones completas	No	Si	En su mayor parte
Simplicidad	No	No	No
Escalabilidad	Si	Si	Si
Flexibilidad de Topología	Si	Si	Si
Performance	Contradictorio	Contradictorio	Contradictorio
Robustez de servicio	Contradictorio	Contradictorio	Si
Mecanismos de transición	Contradictorio	No	Contradictorio
Independencia del medio	Si	Si	Si
Datagrama	Si	Si	Si
Simplicidad de Configuración	Contradictorio	Contradictorio	Contradictorio
Seguridad	Desconocido	Si	Contradictorio
Acceso Estándar	Si	Si	Contradictorio
Soporte de Multicast	Desconocido	Si	Contradictorio
Disponibilidad de Clases de Servicio	Desconocido	Si	Contradictorio
Soporta movilidad	Desconocido	Contradictorio	Contradictorio
Protocolo de control	Desconocido	Si	Contradictorio
Soporta Tunelado	Desconocido	Si	Contradictorio

Tabla 3.1 - *Evaluación de propuestas*

La decisión final se tomó en junio de 1994 y fue la de adoptar SIPP como base para diseñar IPv6, solo que con algunas modificaciones, entre ellas la longitud de la dirección, que es de 128 bits.

Nota: El término de IPv5 fue usado experimentalmente para tiempo real, pero ya es obsoleto

En la figura 3 1 se puede observar como fue desarrollándose el protocolo IPv6.

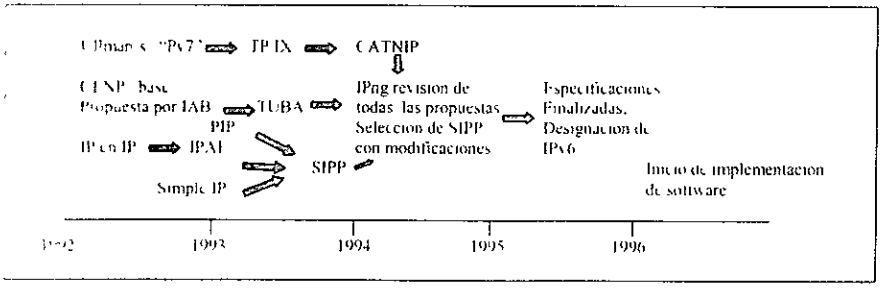


Figura 3.1 - *Evolución de IPv6*

3.2 MEJORAS CON EL PROTOCOLO IPv6

El protocolo IPv4 no fue diseñado para trabajar en lo que hoy en día es Internet, ya que tiene algunas características que se han quedando obsoletas y que son ineficientes o simplemente no sirven, debido a que en el diseño de IPv4 no se previó que en menos de dos décadas el mundo de las computadoras fuera a crecer demasiado, donde ahora existen miles de dispositivos conectados a Internet. Ahora el protocolo IPv6 fue diseñado viendo hacia el futuro de Internet, tomando en cuenta no solo el mundo de las computadoras sino también otro tipo de dispositivos, para que en un futuro este protocolo también sea usado en dispositivos que no necesariamente sean computadoras, como por ejemplo, teléfonos celulares, es por eso el gran número de direcciones que soporta IPv6, entre otras de sus características.

Uno de los problemas ligado a IPv4 es el fenómeno conocido como Explosión de Tablas de Ruteo (*Router Table Explosion*) , ya que el ruteo en IPv4 requiere tablas que van creciendo gradualmente hasta ser inmanejables. En un principio se manejaba el ruteo basándose en el tamaño geográfico de la red (tipos A, B y C, basados en anticipar el tamaño de las redes, teniendo muy pocas redes gigantescas y un gran numero de pequeñas redes). Debido a que algunas redes eran tan grandes, que las tablas de ruteo eran inmanejables, surgió posteriormente lo que es CIDR (*Classless Interdomain Routing*) que es usado para hacer frente al fenómeno de la explosión de tablas de ruteo. El objetivo de CIDR es reemplazar el concepto de ruteo IPv4, creando una estructura de ruteo basada en jerarquías de ruteadores y de redes dentro de Internet, sin tomar en cuenta la jerarquía en base al tamaño geográfico de las redes como en IPv4. Así que IPv6 se basa en CIDR para tener jerarquías de ruteo.

Otro de los aspectos negativos de IPv4 es la ineficiencia en la fragmentación del paquete en la transmisión, ya que el límite del tamaño del paquete conocido como Unidad de Transferencia Máxima (MTU, *Maximum Transfer Unit*) puede variar según el camino que tome el paquete para llegar a su destino, encontrándose con diferentes medios físicos de transmisión, topologías o restricciones de software (por ejemplo, en las redes Ethernet el tamaño máximo del paquete es de 1518 octetos, incluyendo su propia cabecera) por lo que ocurre la fragmentación más de una vez durante el viaje del paquete, hasta que llega a su destino final. En IPv6 la fragmentación se hace sólo cuando es necesario y siempre se realiza por el nodo que envía el paquete y no por los ruteadores, como se hace en IPv4 disminuyendo el rendimiento de los ruteadores.

Una de las opciones que tiene mucho impacto en el rendimiento de la red y provee poco beneficio, es el campo *Checksum* de la cabecera IPv4, ya que este campo sirve para verificar el tamaño del paquete y para confirmar si el paquete llegó completo a su destino, pero este cálculo se hace tomando en cuenta el tiempo de vida (TTL,*Time to Live*), el cual va cambiando durante todo el recorrido del paquete hasta que llega a su destino, ya que cada ruteador donde llega el paquete disminuye en uno el campo TTL, por lo que tiene que volver a realizar el cálculo nuevamente para así obtener el nuevo valor, viéndose afectado en gran parte el rendimiento de la red. En IPv6 se eliminó el campo Checksum, aunque en los mecanismos de transición para la traducción de cabeceras, se usa el campo Checksum sólo que con algunas modificaciones; una de ellas es que ya no se toma en cuenta el TTL (en IPv6 es conocido como Hop-Limit) para

realizar el cálculo de la longitud del paquete, con lo cual los ruteadores ya no necesitan calcular del tamaño del paquete, teniendo con esto un mayor rendimiento en las redes.

IPv6 provee algunos beneficios inmediatos, como es la autoconfiguración, mayor rendimiento en la red, el incremento de la seguridad, y el soporte de aplicaciones de tiempo real (multimedia)

En la tabla 3.2 se presenta una comparación entre IPv4 e IPv6, donde se muestran algunas de sus características más importantes.

CARACTERISTICAS	IPv4	IPv6
Direcciones	• 32 bits (4 octetos)	• 128 bits(16 octetos)
Espacio de direcciones	• alrededor de 10^9 posibles *direcciones*	• alrededor de 10^{38} posibles direcciones
Cabecera del paquete	• Tamaño variable, provoca el consumo de tiempo máquina para su procesamiento	• Tamaño fijo (40 octetos) es más eficiente
Campos Especiales en la Cabecera	• Muchos tipos de estos campos, que normalmente no son soportados por los frabicantes de los equipos *ya que afecta el* rendimiento de los equipos	• Son eliminados para mayor eficiencia o remplazados por otras características.
Tamaño del paquete	• 65536 octetos como máximo	• paquete normal 65536 octetos • Paquete tipo "jumbogram", soporta arriba de 4 billones de octetos, para computadoras LAN de alto rendimiento.
Asignación de direcciones	• por clases de redes A , B, C (Grandes, medianas, pequeñas redes)	• Compatibilidad con IPv4 • Jerarquía por registro, proveedor, subscriptor, y subred • Jerarquía por región geográfica • Arriba de 70% de direcciones reservadas para una expansión futura
Tipo de notación numérica de direcciones	• Notación decimal	• Notación Hexadecimal
Tipo de direcciones	• Punto a punto • Broadcast (local) • Multicast limitado • Anycast experimental	• Anicast (punto a punto) • Multicast • Anycast (llega a un grupo de interfaces) • Direcciones de casos especiales de IPv4 (Loopback)
Fragmentacion	• Posibles múltiples fragmentaciones, hechas por ruteadores, impactando en el desempeño de ruteo	• Hecha más de una vez por el host, no por el ruteador, después del descubrimiento del MTU de la ruta a seguir.

Calidad de servicio (QoS)	• No prevista	• Etiqueta de Flujo (*Label Flow*) • Clase de Trafico (Traffic Class) • Soporte de datos de *tiempo-real* y *distribución* de multimedia
Seguridad	• No prevista (depende de protocolos de niveles más altos, vulnerabilidad para repudio de servicio, ataques (spoofing) y posible engaño de direcciones)	• Autenticación (validación del origen del paquete) • Encriptación o codificación (Privacidad del contenido)
Configuración	• manual	• Configuración automática de redes locales basada en su dirección física
Ruteo	Uso de BGP-4 entre subdominos • Uso de un alto overhead (sobre encabezado, referido a los bits cuales deben ser enviados adicionalmente con el mensaje) TCP • Diseñado para direcciones de 32 bits • Una sola familia de direcciones • Uso de OSPF, RIP	Uso de DIRP entre subdominios • Datagrama IP basado en un bajo overhead (sobre encabezado, son bits los cuales deben ser enviados adicionalmente al mensaje) • Es rediseñado para soportar direcciones de 128 bits • Múltiple tipos de direcciones • Uso de actualizaciones de *OSPF y RIP*

Tabla 3.2 - *Comparación entre IPv4 e IPv6*

3.3 PROBLEMAS DERIVADOS DEL CAMBIO DEL PROTOCOLO IPV4 POR IPV6

El problema principal del cual se derivan otros es que IPv6 no es compatible con IPv4, por lo que requiere de un cambio radical en todos los aspectos (aplicaciones, sistemas operativos, equipo de comunicaciones, etc)

Para que el proceso de transición sea exitoso es necesario un protocolo que permita comunicar IPv4con IPv6 y viceversa. Aún se está trabajando en varios mecanismos de transicion pero el problema que surge de esto, es que si no se encuentra un mecanismo adecuado para realizar la transición que sea reconocido como un estándar para que todas las entidades usen un solo tipo de mecanismo, podría producir un caos total ya que cada entidad puede usar un mecanismo de transición diferente a otras entidades, lo cual podría producir una incompatibilidad de comunicación, ya que si no se diseña un mecanismo adecuado podría ser que IPv6 no sea la mejor opción para reemplazar a IPv4,

provocando que se utilicen otras opciones, como por ejemplo el Traductor de Direcciones de Red (NAT, *Network Address Translation*), que su objetivo es seguir usando las mismas caracteristicas de IPv4 sólo con la diferencia en el uso de direcciones, proponiendo que no es necesario tener una dirección única por cada host en todo internet, sino que sea una dirección única por cada segmento de red y así se realice la comunicación

El estándar de transición debe ser accesible en la comunicación entre IPv4 e IPv6, ya que no se puede cambiar radicalmente la infraestructura IPv4, pues la etapa de transición *se plantea a largo plazo porque significa un costo alto*

Otro de los problemas al que se enfrenta IPv6 es que, dependiendo del mecanismo de transicion que se adopte como estándar, éste se debe ser implementado en todos los nuevos dispositivos que salgan al mercado, para que permita la comunicación entre IPv4 e IPv6 Aunque se tome un sólo mecanismo de transición como estándar, no sería un problema, ya que todos los fabricantes de equipo de comunicaciones tendrían que implementarlo en sus equipos; pero en la actualidad cada fabricante puede usar un mecanismo de transición diferente, lo que conlleva a la problemática que se mencionó anteriormente.

Otro problema es que IPv6 podría tener problemas más adelante con el modelo actual TCP/IP, por lo cual requiere de un periodo de pruebas para prever los posibles problemas que pueden surgir. Para ello se tiene el 6bone, que es el backbone de IPv6, en el cual se tienen conectadas redes IPv6, los cuales pueden comunicarse con el protocolo IPv4 a traves de un mecanismo de transición.

Con respecto a los sistemas operativos, cómo los protocolos, forman parte del kernel de los sistemas operativos (como computadoras que tienen implementado Unix, así como las últimas versiones de Windows NT y OS/2), el cambio del protocolo significa actualizar el sistema operativo. Aunque algunas computadoras solo son de tipo IPv4, esto no afectaria ya que algunos mecanismos de transición se basan en que primeramente se realicen modificaciones en los ruteadores para que sean de tipo IPv6/IPv4, lo cual no afectaria en el cambio de sistemas operativos; pero en otros que involucran la conversión de los Host a IPv6/IPv4 sí se tendría que hacer el cambio de sistema operativo. Mirando hacia el futuro se tienen que realizar la modificaciones en el kernel de los sistemas operativos, para realizar la actualización de los sistemas operativos.

Asi como se necesita el cambio de sistema operativo, así también se necesita la migración de aplicaciones, por lo que los Sockets de la Interface programada de aplicación (API, *Application Program Interface*) deben sufrir cambios con respecto a IPv4, la estructura de los sockets API son definidas para contener campos de 32 bits para direcciones IPv4. Para IPv6 estas estructuras deben sufrir cambios para ahora contener campos de 128 bits para las direcciones de IPv6.

3.4 COMPONENTES DURANTE LA TRANSICIÓN

El actual proceso de transición de IPv4 a IPv6, en principio es un proceso de migración a pequeña escala durante un largo plazo con el fin de desaparecer poco a poco equipos que soporten IPv4, así como sus aplicaciones para que toda la infraestructura y

aplicaciones tiendan hacia IPv6. En el lapso de transición tendrá que existir una interoperabilidad entre los dispositivos de IPv4 e IPv6, para poder realizar las comunicaciones entre dispositivos que trabajen con uno de los dos protocolos o ambos. En la práctica, el concepto de transición significa que por muchos años la red mundial "internet" contendrá ambos tipos de protocolos IPv4 e IPv6. Por esta razón, en el tiempo que dure la transición, la red mundial contendrá varios tipos de componentes como:

Host

Durante la etapa de transición este tipo de componentes tendrán implementado el protocolo IPv4, el protocolo IPv6 o ambos. Así que en la transición existen host IPv4, host IPv6 y host IPv6/IPv4

Ruteadores y Protocolos de ruteo.

Los ruteadores al igual que los hosts pueden tener implementado el protocolo IPv4, IPv6 o ambos Por lo cual en la transición existen Ruteadores IPv4, Ruteadores IPv6 y Ruteadores IPv6/IPv4

Los protocolo de ruteo se consideran como otro tipo de componente que va de la mano con los ruteadores, ya que estos protocolos de ruteo también sufren un cambio aunque no es tan radical, ya que para poder rutear los paquetes de IPv6 se necesita de un nuevo protocolo de ruteo, o incluso los que se están usando solo que con algunas modificaciones como lo son OSPFv6, RIPv6 entre otros, por lo cual se tienen componentes de protocolos de ruteo para IPv4 (OSPF, RIP, etc) y los protocolos de ruteo para IPv6

Sistema de Nombres de Dominio (DNS, *Domain Name System*)

Durante la fase de transición el DNS soportará hosts IPv4 con direcciones de 32 bits y host IPv6 con direcciones de 128 bits, así el DNS debe ser capaz de responder para ambas direcciones según sea el caso

Este nuevo DNS registrará nombres llamados "AAAA" que son usados para direcciones de 128 bits de IPv6, las cuales son cuatro veces mas grandes que las de IPv4, y nombres "A" para direcciones de 32 bits de IPv4 Para nodos IPv6/IPv4 el servicio de nombres también contiene la tradicional dirección IPv4 de 32 bits "A" y la dirección "AAAA" de IPv6, para que responda con la dirección adecuada según sea el caso.

Para las direcciones IPv4 compatibles con IPv6 no es necesario responder con "AAAA" y con "A" ya que las bibliotecas de IPv6 son capaces de extraer la dirección IPv4 de la dirección IPv6 compatible.

En una forma general los componentes de transición se clasifican como sigue:

Nodos sólo IPv4 (only-IPv4)

Este tipo de nodo puede ser un host o un ruteador que sólo tenga implementado el protocolo IPv4. Este tipo de nodo solo-IPv4 no es capaz de comunicarse con nodos IPv6 ya que no entiende la información que le es enviada por los nodos IPv6.

Nodos sólo IPv6 (only-IPv6)

Este nodo puede ser un host o un ruteador que sólo tenga implementado el protocolo IPv6 En nodos sólo-IPv6 se implementan un mínimo de mecanismos de transición, pero no se puede implementar el mecanismo de tunelado. En la sección 3 5 se explican los mecanismos de transición. Estos nodos no se pueden comunicar con nodos IPv4.

Nodos IPv6/IPv4

Este tipo de nodo puede ser un host o ruteador que tenga implementado ambos protocolos. IPv4 e IPv6. Estos nodos también pueden tener implementado algún mecanismo de transición como lo es el tunelado.

Nodos IPv4

Este tipo de nodo es cualquier host o ruteador que tenga implementado IPv4. Los nodos IPv6/IPv4 y nodos solo-IPv4 están dentro de esta clasificación de nodos IPv4.

Nodos IPv6

Este tipo de nodo es cualquier host o ruteador que tenga implementado IPv6. Los nodos IPv6/IPv4 y nodos solo-IPv6 están dentro de esta clasificación de nodos IPv6.

Area completa de IPv4

Una región de infraestructura que sólo puede enrutar paquetes IPv4

Area completa de IPv6

Una region de infraestructura que sólo pueden enrutar paquetes IPv6

3.5 MECANISMOS DE TRANSICIÓN

El IETF (*Internet Engineering Task Force*) ha invertido mucho tiempo en el diseño de *mecanismos de transición que aseguren una evolución segura de IPv4 a IPv6. Si esta* transicion no se hace cuidadosamente, el costo y complejidad del cambio podrían hacer retractarse a los usuarios de cambiar al nuevo protocolo. Una de las primeras metas de los diseñadores de los mecanismos de transición fue permitir tanta flexibilidad como fuera razonablemente posible.

La estrategia de transición evita dependencias entre los varios elementos de una red durante el proceso de actualización, esto es, por ejemplo, si un usuario necesita esperar a que otra máquina se actualizara antes de poder actualizarse; con esto la transición podría alentarse. Para combatir esto, IPv6 puede debe ser introducido primero en los hosts o *primero en los ruteadores, o en ambos a la vez.*

Existen infraestructuras edificadas en IPv4 tras un periodo grande de tiempo y a un alto costo; como consecuencia, los usuarios no aceptarán tan fácilmente el cambio a IPv6.

Para ayudar a las redes ya instaladas con los costos de arranque, la estrategia de transición permite a IPv6 explotar todos los recursos de IPv4, lo cual es benéfico para todos los sitios.

3.5.1 Mapeo Simplificado de Direcciones

Este mecanismo consiste en usar direcciones IPv4 compatibles con IPv6, reemplazando direcciones IPv4 de 32 bits por direcciones IPv6 de 128 bits, y sustituyendo el prefijo de 96 bits de la dirección.

En la transición se usan dos formatos especiales de direcciones IPv6, ambos están reservados para direcciones IPv4 compatibles con IPv6 y direcciones IPv4 que son mapeadas o sustituidas por direcciones IPv6

Las direcciones IPv4 compatibles con IPv6 son asignadas para nodos IPv6/IPv4 que soportan túneles automáticos y tiene la estructura que se muestra en la figura 3.2

80 bits (10 octetos)	*16 bits* (2 octetos)	*32 bits* (4 octetos)
0:0:0:0	0000	Dirección IPv4

Fig. 3.2 - Formato de la dirección IPv4 compatible con IPv6

Las direcciones de los nodos sólo-IPv4, son representadas como direcciones IPv4 mapeadas con formato IPv6. Estas direcciones tienen la estructura que se muestra en la figura 3 3

80 bits (10 octetos)	16 bits (2 octetos)	32 bits (4 octetos)
0:0:0:0	FFFF	Dirección IPv4

Fig. 3.3 - *Formato de la dirección IPv4 mapeada con IPv6*

Las direcciones IPv4 compatibles con IPv6 son designadas para ser usadas por nodos IPv6 que desean interoperar con nodos IPv4. Este tipo de direcciones son escuchadas por el DNS en IPv6 "AAAA" y IPv4 "A".

Las direcciones IPv4 mapeadas con IPv6 sólo son usadas para representar las direcciones de nodos IPv4, estas nunca son asignadas para nodos IPv6. Este tipo de direcciones solo son escuchadas por el DNS "A".

3.5.2 Protocolo Traductor de Cabeceras

Este protocolo de traducción consiste de un simple mapeo o sustitución entre los dos protocolos, con algunas reglas especiales para la fragmentación y el descubrimiento del MTU La operación básica es la de remover la cabecera IP original y remplazarla con la nueva cabecera de la otra versión IP, este protocolo también hace la traducción del protocolo ICMP En la figura 3 4 se muestra como se realiza la traducción de cabeceras.

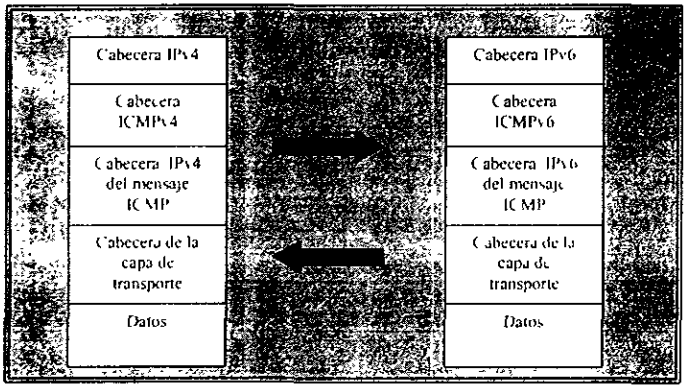


Fig 3 4 - *Función del Protocolo Traductor*

La traducción de cabeceras es un mecanismo opcional que es usado cuando se desea permitir que nodos sólo-IPv6 interoperen con nodos sólo-IPv4, aunque en la práctica este concepto ha sido demasiado complejo.

El traductor de cabeceras es ejecutado por los ruteadores, el cual interconecta áreas completas de IPv4 e IPv6 Todo él tráfico que cruza por la frontera de estas áreas debe ser traducido por el protocolo traductor. Este tráfico puede venir de distintas formas:

a) Tráfico IPv4 de término - Paquetes IPv4 que son direccionados hasta un nodo dentro de una área de direcciones IPv6

b) Tráfico IPv4 de tránsito - Paquetes IPv4 que son direccionados hasta un nodo que esta fuera de un área IPv6, pero que debe pasar por el área IPv6.

c) Tráfico IPv6 de término– Paquetes IPv6 que son direccionados hasta un nodo que esta dentro de un área IPv4

d) Tráfico IPv6 de tránsito - Paquetes IPv6 que son direccionados hasta un nodo fuera de una área IPv4, pero debe pasar por un área de IPv4.

e) Encapsulamiento de tráfico IPv6- Paquetes IPv6 encapsulados con paquetes IPv4.

Los traductores de cabeceras son ruteadores IPv6/IPv4, estos se encargan traducir las cabeceras de paquetes *IPv4 en IPv6, y cabeceras IPv6 en IPv4. Los ruteadores* requieren alguna información de configuración para conocer cuáles paquetes podrían ser traducidos, y cuáles podrían ser simplemente reenviados sin ninguna modificación.

En la figura 3.4 se muestra el caso donde el traductor de cabecera está siendo usado para comunicar área IPv4 con áreas IPv6 El protocolo traductor debe traducir todos los paquetes IPv4 que son direccionados hasta los nodos localizados en el área de IPv6 o que deben pasar por el área IPv6.

En la figura 3 5 se puede observar cómo el ruteador traductor conecta un área IPv4 con un area IPv6

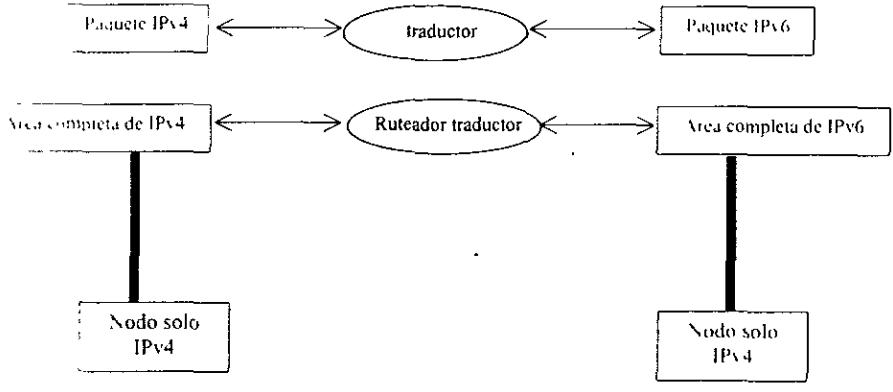


Figura 3.5 - *Interoperabilidad con el traductor de cabeceras*

Cuando se traducen paquetes IPv6 a IPv4, el ruteador traductor usa los 32 bits menos significativos de la dirección origen y destino de IPv6 para generar la dirección para el paquete IPv4. La direcciones origen y destino deben ser direcciones compatibles IPv4 para que el paquete pueda ser traducido.

Cuando se traducen paquetes IPv4 a IPv6, los ruteadores traductores, agregan el prefijo 0.0.0.0:0:0 para la dirección IPv4 origen para generar la dirección origen para el paquete IPv6. Ellos agregan el prefijo 0:0:0:0:0:0 o 0:0:0:0:0 FFFF para generar la direccion destino. Para que el ruteador sepa cuál prefijo va a agregar requiere de una información de configuración. Los traductores usan el prefijo 0:0:0:0:0:0 si el destinatario es localizado dentro de un área IPv6, y agrega el prefijo 0:0:0:0:0.FFFF si el destinatario es localizado fuera del área IPv6

Las cabeceras IPv6 e IPv4 tienen algunas similaridades, algunos campos que son familiares para un protocolo IP, son extraños para otro, ya que difiere el tamaño o tiene diferente significado. El traductor directamente copia, traduce, ignora o pone campos en la cabecera IP con valores por default cuando traduce de una versión IP a otra. En la figura 3.6 se muestra la acción que toma el traductor para cada campo de la cabecera.

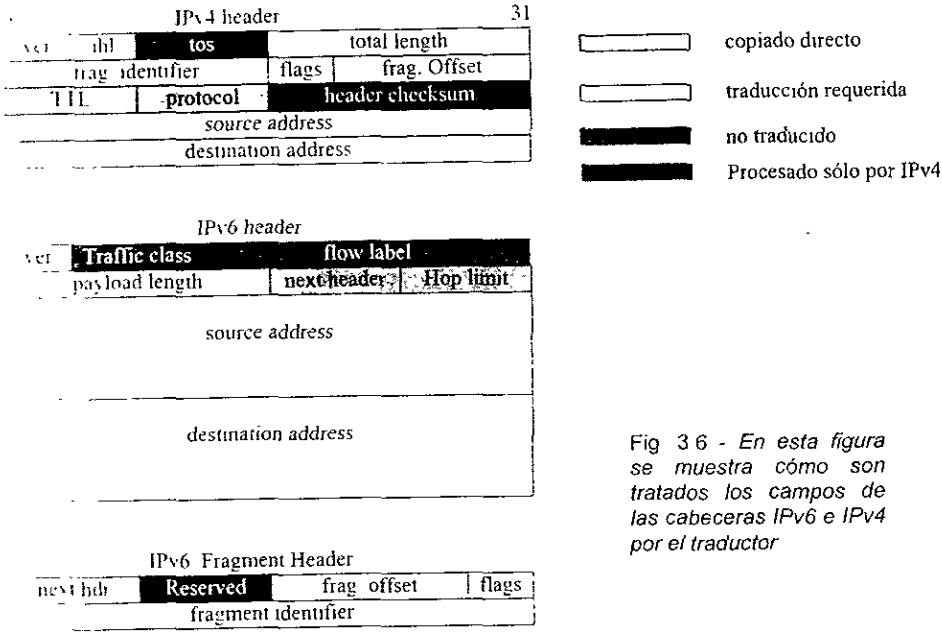


Fig 3.6 - *En esta figura se muestra cómo son tratados los campos de las cabeceras IPv6 e IPv4 por el traductor*

Muchos de los campos requieren un simple ajuste. El campo *checksum* es calculado cuando se realiza la traducción de las cabeceras de IPv6 a IPv4 , y es ignorado cuando se realiza la traducción de IPv4 a IPv6. El campo *total-length* de IPv4 incluye el tamaño de la cabecera IPv4 mientras que el campo pay-load de IPv6 no la incluye. El *hop limit* (límite de saltos) y el TTL son copiados y decrementados en uno. El campo *protocol* puede ser directamente copiado desde una versión IP a otra Con excepción de la cabecera de fragmentación de IPv6, todos las demás cabeceras suplementarias de IPv6 y las opciones de IPv4 son ignoradas por el traductor. El campo tipo de servicio (ToS, *type-of-service*) de IPv4 y el campo *traffic-class* y *flow-label* son también ignorados por el traductor de cabeceras IP.

Cuando el ruteador traductor recibe el paquete de fragmentación, la traducción se hace directa solo con la excepción en los campos de fragment-identifier, ya que son de tamaño distinto ya que en IPv6 el campo es de 32 bits, el doble del tamaño del campo en IPv4, por lo que cuando se traduce de IPv6 a IPv4 sólo se copian lo 16 bits menos significativos del campo

Traducción de Cabeceras de IPv6 a IPv4

Si no existe cabecera de fragmentación, los campos de la cabecera IPv4 son como se muestra a continuación:

Version 4

Internet Header Lenght Puesta con el valor cero.

Total Length. Contiene el valor de la longitud de la cabecera IPv6, más el tamaño de la cabecera IPv4

Identification Puesta con el valor cero.

Flags Si no hay cabecera de fragmentación es puesta en verdadera con el valor de 1.

Fragment Offset: Puesta con el valor cero

Time to Live El valor del campo Hop limit de la cabecera IPv6 es copiado a este campo y decrementado en uno.

Protocol Es copiado desde el campo Payload desde la cabecera IPv6

Header Checksum Es calculado cuando se crea la cabecera IPv4.

Source y Destination Addresses. Ese depende del mecanismo de mapeo de direcciones, que se vio en el punto 3.5.1

Si el paquete IPv6 contiene cabecera de fragmentación los campos de fragmentación son puestos como a continuación se muestra:

Total Lenght: Es el valor del campo Payload, restándole 8, por la cabecera de fragmentación, más el tamaño de la cabecera IPv4.

Flags El valor es copiado desde la cabecera de fragmentación

Fragment Offset: Copiado desde el campo Fragment Offset del campo de fragmentacion

Traducción de Cabeceras de IPv4 a IPv6.

Si no hay fragmentación, entonces la cabecera de IPv6 es como sigue:

Versión. 6

Traffic-Class. 0

Flow ID. 0

Payload Lenght: Valor del total de longitud de la cabecera IPv4 menos el valor del campo Internet Header Length de la cabecera IPv4.

Next Header. Copiado desde el campo Protocol, si el valor del campo Protocol es 1 (ICMPv4), entonces es substituido con 58(ICMPv6)

Hop Limit. Es el valor del campo Time to Live de la cabecera IPv4 y es decrementado en uno

Source y destination Addresses. Depende el mecanismo de mapeo de direcciones, que se vio en el punto 3.5 1

Si existe la necesidad de agregar una cabecera de fragmentación en IPv6, las cabeceras involucradas se ponen como sigue:

Payload Lenght: Es el valor total de la longitud de la cabecera IPv4 menos el valor del campo Internet Header Lenght de la cabecera IPv4, más 8 de la cabecera de fragmentación.

Next Header: 44, que es el valor de la cabecera de fragmentación

Los campos de la cabecera de fragmentación son puestos como sigue

Next Header: Copiado desde el campo de Protocol, si el valor del campo Protocol es 1 (ICMPv4), entonces es substituido con 58(ICMPv6)

Fragment Offset: el valor es copiado desde el campo Fragment Offset de IPv4

M flag el valor es copiado desde el campo de more Fragment de IPv4

Fragment Identification: Los 16 bits del campo Fragment Identification son copiados a este campo y los 16 bits restantes son puestos en cero.

Traducción de Cabecera ICMP

Los formatos de ICMPv4 e ICMPv6 son semejantes, sólo cambia el valor de los campos A continuación se muestra cómo se realiza la traducción de las cabeceras de ICMP

Traducción de ICMPv4 a ICMPv6

EL campo Type para el mensaje de Solicitud de eco (Echo request) y contestación de eco (Echo Reply) (con el valor 8 y 0 respectivamente en el campo Type para un mensaje de tipo ICMPv4): se pone el valor de 128 y 129 respectivamente para el mensaje ICMPv6.

El mensaje de destino inalcanzable (Destination Unreachable) con valor en el campo Type de 3 para ICMPv4: este se pone con el valor 1 para el mensaje de destino inalcanzable ICMPv6.

Para el mensaje de tiempo excedido con valor de 11 en el campo Type del mensaje ICMPv4, se pone el valor 3 en el campo Type de mensaje de tiempo excedido de ICMPv6, el valor de campo Code no sufre modificaciones

Para el mensaje de problemas de parámetro con valor de 12 en el campo Type del mensaje ICMPv4, se coloca el valor 4 en el campo Type del mensaje de problemas de parámetro ICMPv6, y se traducen los valores del campo Pointer de 0a 0, de 2 a 4, de 8 a 7 de 9 a 6, de 12 a 8,de 16 a 24, y para otros valores de este campo se pone el valor de 1

El valor del campo *code* para mensajes ICMPv4 se traduce como se muestra a continuacion

- Para los valores 0,1,6,7,8,11 y 12 de campo Code del mensaje ICMPv4, se pone el valor 0 en el campo Code del mensaje ICMPv6 , ya que es para un tipo de mensaje de destino inalcanzable, que significa que no existe ruta para llegar a el destino.
- Para el valor 2· Se traduce como un mensaje de problema de parámetro (con valor de los campos de ICMPv6 Type =4 y Code=1) y se pone el valor del campo Pointer en 6
- Para el valor 3 de ICMPv4 se pone el valor 4 de puerto inalcanzable, que es para un mensaje ICMPv6 de destino inalcanzable
- Para el valor 4 del campo Code de ICMPv4: se traduce como un mensaje ICMPv6 de tipo paquete demasiado grande (con valores de campo Type=2 y Code=0) y el campo MTU necesita ser ajustado por la diferencia entre el tamaño de cabecera de IPv4 e IPv6
- Para el valor 9 y 10 del campo Code de ICMPv4: éste se traduce como un mensaje de ICMPv6 de destino inalcanzable, por una prohibición administrativa, por lo que se coloca el valor 1.

Traducción de ICMPv6 a ICMPv4

El campo Type para el mensaje de Solicitud de eco (Echo request) y contestación de eco (Echo Reply), con el valor en el campo Type 128 y 129 respectivamente para un mensaje de tipo ICMPv6. Se pone el valor de 8 y 0 respectivamente para el mensaje ICMPv4

El mensaje de destino inalcanzable (Destination Unreachable), con valor en el campo Type de 1 para ICMPv6. Este campo se pone con el valor 3 para el mensaje de destino inalcanzable del mensaje ICMPv4. Los valores del campo Code se traducen como a continuación se muestra:

- Code 1 se pone el valor 10 en el campo Code (comunicación del destino administrativamente prohibida)
- Code 2. Se pone el valor 5 (origen de ruta fallida)
- Code 3: se pone el valor 1 (de host inalcanzable)
- Code 4 Se pone le valor 3 (de puerto inalcanzable)

Para el mensaje de tiempo excedido con valor 3 en el campo Type del mensaje ICMPv6, se pone el valor 11 en el campo Type de mensaje de tiempo excedido de ICMPv4, y el valor de campo Code no sufre modificaciones.

Para el mensaje de problemas de parámetro de ICMPv6 con valor de 4 en el campo Type, se pone el valor 2 en el campo Code y 12 en el campo Type, y valor del campo Pointer con –1 Si el valor es 1, se traduce como un mensaje ICMPv4 de tipo protocolo irreconocible, con valores de campo type=3 y code=2. Si el valor del campo Code es 0, entonces se pone un valor 12 en el campo Type y un valor 0 en el campo Code y se traducen los valores del campo Pointer como sigue, de 0 a 0, de 4 a 2, de 7 a 8, de 6 a 9, de 8 a 12, de 24 a 16 y para otros valores se pone el valor de –1.

Para un mensaje de paquete demasiado grande con valor 2 en el campo Type se traduce hasta un mensaje ICMPv4 de tipo destino inalcanzable con valor 4 en el campo Code y el campo MTU debe ser ajustado por la diferencia de tamaño de los cabeceras IPv6 e IPv4

3.5.3 Doble Capa IP (Dual IP Layer)

La forma más directa para que los nodos IPv6 sean compatibles con nodos sólo-IPv4 es disponer de una implementación completa de IPv4. Los nodos IPv6 que proveen esta implementación en adición con su implementación IPv6 son llamados "nodos IPv6/IPv4", esto es llamado un esquema de doble capa IP, que permite que a IPv6 se le añadan hosts, servidores DNS y ruteadores, sin *ningún cambio o ruptura en el soporte actual* protocolo IPv4. Los nodos IPv6/IPv4 tienen la habilidad de enviar y recibir paquetes de IPv4 e IPv6 Ellos pueden interoperar con nodos IPv4 usando paquetes IPv4, y también interoperar directamente con nodos IPv6 usando paquetes IPv6 En la figura 3.7 se muestra cómo es que se encuentra la doble capa IP.

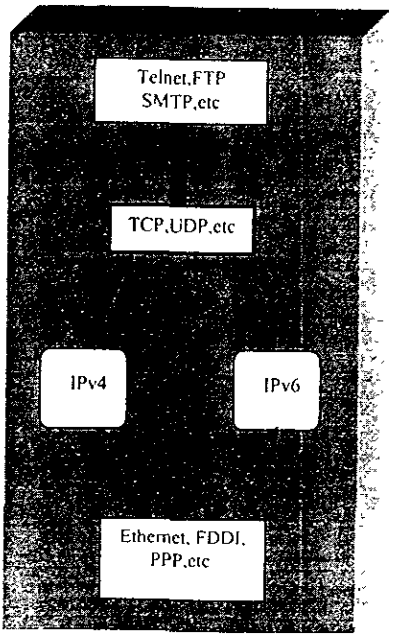


Fig 3.7 - *Estructura general de un nodo IPv6/IPv4 en una red de doble capa IP*

En el mecanismo de doble capa IP, para que el DNS pueda servir como un cliente de doble capa IP, debe ser actualizado para manejar el nuevo tipo de direcciones IPv6 AAAA" Los servidores que proveen soporte AAAA no necesitan ser actualizados para ser uso de las direcciones IPv6 para la transferencia de datos entre los servidores DNS. Para sitios que no han actualizado sus servidores de DNS, los nodos IP pueden resolver direcciones de redes usando tablas estáticas de hosts definidas manualmente

La transición de la doble capa IP permite que paquetes IPv4 e IPv6 sean enrutados independientemente. Mucha de la flexibilidad de la estrategia de la transición IPv6 proviene del hecho de que los ruteadores se ocupen ya de protocolos múltiples Cuando las redes de doble capa IP sean actualizadas, las aplicaciones tradicionales de IPv4 pueden ser ejecutadas sin ningún cambio El tunelado provee una manera de utilizar la infraestructura ya establecida de IPv4 para llevar él trafico IPv6, mientras hay recursos nativos de IPv6 que puedan ser explotados.

El tunelado aligera el sistema de ruteo de IPv6, si tenemos una red con IPv6, en parte de ella el ruteo será más eficiente si se realiza el tunelado de IPv6, en lugar de cambiar de golpe toda la topologia.

El mecanismo de doble capa IP puede o no puede ser usada en conjunción con el mecanismo tunelado IPv6 sobre IPv4 (IPv6 over IPv4 tunneling) Un nodo IPv6/IPv4 que soporta tunelado, puede soportar solo el tunelado configurado, o ambas configuraciones, tunelado configurado y tunelado automático Los mecanismos de tunelado se describen en la seccion 3 5.4

Existen tres configuraciones posibles:

- Nodo IPv6/IPv4 que no ejecuta el tunelado
- Nodo IPv6/IPv4 que ejecuta solo el tunelado configurado por default
- Nodo IPv6/IPv4 que ejecuta tunelado configurado por default y el tunelado automatico.

3.5.4 Construcción de Túneles IPv6 sobre IPv4

En la mayoría de los ambientes de desarrollo, la infraestructura de ruteo IPv6 será lanzada en un cierto plazo, la construcción de un túnel proporciona una manera de utilizar una infraestructura existente del enrutamiento IPv4 para controlar el tráfico IPv6 a pesar de que apenas están surgiendo recursos nativos de IPv6 Mientras que se está desarrollando IPv6, la infraestructura existente de IPv4 puede seguir existiendo y ser funcional con IPv6, ya que es muy difícil, por no decir imposible cambiar radicalmente el protocolo IPv4 por IPv6. La construcción de túneles puede simplificar los procesos de transicion para los usuarios, así como proporcionar otras ventajas, pues al realizar la construcción de túneles fortalece el sistema de enrutamiento IPv4 para construir un sistema de enrutamiento IPv6.

La construcción de túneles ayuda a activar pronto un servicio global IPv6 en la transición, pues quienes adopten IPv6, aunque pueden estar dispersos geográficamente, pueden construir un túnel para proporcionar conectividad global con IPv6 sin esperar que Internet se cambie totalmente a IPv6. La construcción de túneles utiliza una estrategia de transición en la cual la infraestructura del enrutamiento IPv6 pueda crecer gradualmente.

Para enviar un paquete en un túnel, un nodo primero crea una cabecera de encapsulamiento IPv4, y enseguida transmite el paquete encapsulado. Cuando el paquete llega a su destino, se desencapsula el paquete quitándole la cabecera IPv4, como se muestra en la figura 3 8

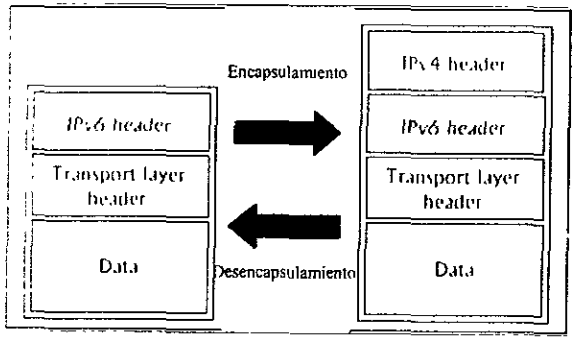


Figura 3 8 - *Encapsulado y desencapuslado del paquete IPv6*

La dirección destino del paquete de encapsulamiento IPv4 especifica el extremo del tunel, que es el nodo que recibe el paquete encapsulado, este extrae la cabecera de encapsulado IPv4, actualiza la cabecera IPv6, y después procesa el paquete IPv6 como cualquier otro paquete recibido.

Existen dos métodos importantes para la construcción de túneles el hacer un túnel configurado por default y el hacer un túnel automático

Debido a que los hosts y los ruteadores pueden desempeñar el papel de extremo en un túnel, hay un número de configuraciones factibles para realizar un túnel. Algunas de estas configuraciones se prestan para hacer un túnel automático, mientras que otras requieren el método de configuración por default. Las configuraciones son las siguientes:

Ruteador a Ruteador. Los ruteadores IPv6/IPv4 interconectados por una infraestructura IPv4 pueden transmitir paquetes IPv6 por túneles entre sí mismos. En este caso, el túnel atraviesa un segmento del camino punto a punto que el paquete IPv6 toma.

Host a Ruteador. Los hosts IPv6/IPv4 pueden transmitir paquetes IPv6 por un túnel hacia un ruteador intermediario IPv6/IPv4 que sea accesible por medio de una infraestructura IPv4. Este tipo de túnel atraviesa el primer segmento de la trayectoria punto a punto del paquete.

Host a host. Los host IPv6/IPv4 que están interconectados por una infraestructura pueden transmitir paquetes IPv6 por un túnel entre sí mismos En este caso, el túnel atraviesa el camino entero punto a punto que el paquete tiene que recorrer.

Ruteador a Host Los ruteadores IPv6/IPv4 pueden transmitir paquetes IPv6 con el host de destino final IPv6/IPv4. Este túnel atraviesa solamente el último segmento de la trayectoria punto a punto

En las dos primeras configuraciones de tunelamiento descritos anteriormente, la de ruteador a ruteador y de host a ruteador, el paquete IPv6 esta siendo tunelado a un ruteador. El extremo del túnel es como un ruteador intermediario el cual debe desencapsular el paquete IPv6 y direccionarlo a su destino final. Estas configuraciones usan comunmente túneles de configuración por default.

En las últimas dos configuraciones, host a host y ruteador a host, los paquetes IPv6 usan el tunelado tipo automático.

Las dos técnicas de tunelado difieren principalmente en cómo determinan la dirección del tunel del punto final, aunque el funcionamiento es el mismo.

A la entrada del primer nodo del túnel (nodo de encapsulamiento) se crea un encapsulamiento con una cabecera IPv4 y se transmite el paquete encapsulado.

A la salida del último nodo del túnel (nodo de desencapsulamiento) se recibe el paquete encapsulado; entonces se remueve la cabecera IPv4, se actualiza la cabecera IPv6 y se procesa el paquete IPv6.

El nodo de encapsulamiento puede necesitar mantener un estado de captura de información por cada túnel para así registrar parámetros como el MTU del túnel para procesar los paquetes de IPv6 y enviarlos dentro del túnel. Aunque esta información puede descartarse cuando no se use.

Túnel Configurado por Default

Los nodos que están conectados con infraestructuras de ruteo IPv4 pueden utilizar un tunel configurado para obtener un backbone IPv6. Si la dirección IPv4 de un ruteador IPv6/IPv4 de frontera del backbone es conocida, el túnel puede ser configurado a este ruteador. Este túnel puede ser configurado dentro de la tabla de ruteo como ruteador por *default o ruta por default* (default router). Es decir, todas las direcciones destino IPv6 coincidirán con la ruta y podrán potencialmente atravesar el túnel. Puesto que la longitud de la mascara de tal ruta por default es cero, será utilizada únicamente si no hay otras rutas con una máscara más larga que correspondan con el destino.

La dirección del extremo del túnel por default podrá ser la dirección IPv4 de un ruteador IPv6/IPv4 en la frontera del backbone IPv6. Alternativamente, el extremo del tunel podrá ser una dirección IPv4 de tipo "anycast". Con este acercamiento, los ruteadores fronterizos múltiples IPv6/IPv4 informan sobre la escalabilidad o alcance de IPv4 en la misma dirección IPv4. Todos estos ruteadores aceptan los paquetes de esta dirección como propios y desencapsulan los paquetes IPv6 tunelados a esta dirección. Cuando un nodo IPv6/IPv4 envía paquetes encapsulados a esta dirección, será entregado solo a uno de los ruteadores fronterizos, pero el nodo que envía el paquete no sabrá cuál es. El sistema de ruteo IPv4 transportará, generalmente trafico al ruteador más cercano.

El usar un túnel por default a una dirección IPv4 anycast de cualquier tipo de transmisión proporciona un alto grado de robustez. Puesto que los ruteadores múltiples fronterizos son de este tipo de direcciones y con los mecanismos normales de retraso de ruteo IPv4, el trafico cambiará automáticamente a otro ruteador cuando uno de ellos se caiga.

Túnel de Configuración Automática

En el tunelado automático la dirección del extremo del túnel es determinado desde que el paquete está en el túnel Cuando un host es el extremo del túnel (en el tunelamiento host a host y ruteador a host), los paquetes IPv6 son tunelados en todo su camino hasta *u destino final* Estas configuraciones por lo general usan tunelado automático. En este aso, el extremo del túnel es el nodo por el cual el paquete IPv6 es direccionado o rocesado A partir de que un extremo del túnel es el último destino del paquete IPv6, el *xtremo del túnel puede ser determinado por la dirección* IPv6 destino, si la dirección es ompatible con IPv4 entonces los 32 bits menos significativos guardarán la dirección IPv4 el nodo destino, y eso puede ser usado como la dirección destino del extremo del túnel. *Por lo tanto, el tunelamiento automático elimina la necesidad de configurar explícitamente* configuración del extremo del túnel. Los paquetes IPv6 que no son direccionados a una irección compatible con IPv4 no pueden usar tunelamiento automático

Los nodos IPv6/IPv4 necesitan determinar cuáles paquetes pueden ser enviados por medio del tunelamiento automático Una manera es usar una tabla de enrutamiento IPv6 para dirigir el tunelamiento automático. La implementaron puede tener un registro en una tabla especial fija de ruteo para el prefijo 0:0:0:0.0·0/96 (esto es, una ruta al prefijo con solo ceros con una máscara de 96 bits). Los paquetes que concuerden con este prefijo son enviados a un controlador de pseudo-interface que realiza el tunelado automático. A partir de que todas las direcciones IPv6 compatibles con IPv4 concordarán con este prefijo, todos los paquetes dirigidos a esos destinos serán auto-tunelados (al menos que exista una mejor ruta de concordancia).

En ambos tipos de tunelado (automático y configurado), la dirección origen de la cabecera IPv4 del paquete tunelado son los 32 bits menos significativos de una dirección IPv6 compatible con IPv4, del nodo que realiza el encapsulamiento La dirección destino IPv4 son los 32 bits menos significativos de la dirección IPv6 compatible con IPv4 del extremo del túnel

En la tabla 3.3 se muestra cómo se obtienen la direcciones IPv4 a partir de una irección IPv6 compatible con IPv4.

Tipo de Tunelado	Nodo de Encapuslamiento	Nodo de Desencapsulamiento	Dirección IPv4 del extremo del túnel.
Automático	Host origen	Host destino	Los 32 bits menos significativos de la dirección IPv6 del host destino
Automático	Ruteador	Host destino	Los 32 bits menos significativos de la dirección IPv6 del host destino
Configurado por default	Host origen	Ruteador	Los 32 bits menos significativos de la *dirección IPv6 del ruteador* De desencapsulamiento

Configurado por default	Ruteador	Ruteador	Los 32 bits menos significativos de la dirección IPv6 del ruteador De desencapsulamiento

Tabla 3 3 - *Obtención de la dirección IPv4 del nodo extremo del túnel*

Tunelamiento y DNS

Cuando una dirección IPv6 compatible con IPv4 es asignada a un host IPv6/IPv4 que soporta tunelamiento automático, el registro A y los registros AAAA correspondientes pueden ser listados en el DNS. Los registros AAAA guardan la dirección IPv6 compatible con IPv4 mientras que el registro A guarda los 32 bits menos significativos de esta dirección El registro AAAA será utilizado por solicitudes de hosts IPv6, mientras que el registro A será utilizado por solicitudes de hosts IPv4 únicamente.

La decisión de establecer un registro AAAA en el DNS que guarde una dirección IPv6 compatible con IPv4 para un host IPv6, puede ser utilizada como una política de control de trafico que llega a ese host Si una dirección compatible con IPv4 es listada, entonces otros hosts originarán tráfico tunelado a ese host Si sólo un registro A es listado, entonces el host aparecerá para los otros como un host IPv4 únicamente y solo será enviado trafico IPv4.

Cuando una petición de un host IPv6/IPv4 localiza un registro AAAA que guarda una direccion IPv6 compatible con IPv4, así como cuando un registro A guarda la dirección IPv4 correspondiente, las librerías de resolución no requieren necesariamente devolver las dos direcciones a la aplicación. Tiene tres opciones:

- Devolver sólo la dirección IPv6 a la aplicación
- Devolver sólo la dirección IPv4 a la aplicación
- Devolver las dos direcciones a la aplicación.

La determinación de cuáles direcciones devolver puede ser utilizada como una política de selección en el host para controlar el tráfico originado por ese host. Si el administrador *del sistema desea prevenir que su host origine tráfico tunelado, puede configurar* las librerías de resolución para devolver sólo direcciones IPv4 a la aplicación. Si se permite el trafico tunelado, entonces el administrador permite que direcciones IPv6 (compatibles con *IPv4) sean devueltas a la aplicación.*

Características de la Implementación del Tunelamiento

Los nodos que realizan tunelamiento necesitan ajustarse a una serie de características *que son comunes para el tunelamiento automático y configurado. Muchas* de estas características tratan el hecho que la topología y operaciones de ruteo de un túnel IPv4 es demasiado transparente para los nodos IPv6.

- La MTU (Maximum Transfer Unit) del túnel y la fragmentación: Es técnicamente factible para un nodo de encapsulamiento manejar a un túnel como un conexión

virtual IPv6, con un MTU largo dependiendo de una capa IPv4 de fragmentación y del reemsamble para liberar paquetes IPv6 que son más grandes que el MTU de las conexiones subyacentes de la ruta entre el nodo de encapsulamiento y de desencapsulamiento. Pero esto nos lleva a una fragmentación interna dentro del túnel, lo cual es ineficiente en el rendimiento de la red. Una mejor perceptiva es hacer que el nodo de encapsulamiento interprete rutas IPv4 en la ruta del tunel y después usarla junto con la ruta IPv6 para reportar el tamaño del MTU de regreso al host origen.

- Mantenimiento del estado de información: Si el nodo de encapsulamiento interpreta rutas IPv4 del MTU de sus túneles, necesitará mantener un estado de información para cada túnel Debido a que el número de túneles que un nodo puede estar usando puede crecer gradualmente, dicho nodo debe emplear un esquema de refrescamiento del estado de información que necesita y periódicamente descartar la información que no este siendo usada.

- *Límite de salto de IPv6 a IPv4 TTL (**Hop Limit**)*: Los túneles IPv6 en IPv4 son tratados como conexiones de salto sencillo desde el punto de vista de IPv6 Esto es, el límite de saltos de IPv6 es decremetado en uno cuando un paquete IPv6 atraviesa un túnel. Pero el nodo de encapsulamiento debe utilizar un valor TTL en la cabecera del encapsulado IPv4, que es bastante grande como para garantizar que el paquete encapsulado no expirará en el túnel ruteado, hasta que llegue al nodo de desencapsulamiento. Este valor normalmente lo establece el administrador de la red.

- *Mensaje de error ICMPv4 para IPv4 y tunelamiento* Los paquetes tunelados pueden fallar al ser liberados en el extremo del túnel porque un extremo puede ser incomunicable. El TTL de IPv4 no es lo suficientemente grande o el paquete es demasiado grande (sobrepasa el MTU) Estas fallas mandarán mensajes de error ICMPv4 dirigidos de vuelta al punto de entrada del túnel Algunos de estos mensajes de error del ICMPv4 pueden no contener bastante información del paquete original IPv6 para poder identificar su origen, puesto que muchos ruteadores IPv4 devuelven solamente 8 bytes de datos más allá que la cabecera IPv4 del paquete de error. Si es posible, el nodo de encapsulado debe procurar recuperar el paquete original IPv6, y generar el mensaje de error apropiado ICMPv6 y enviarlo al nodo origen.

Para lograr un ruteo dinámico en un ambiente tan híbrido, se requieren mecanismos globales para distribuir la capa de red IPv6. El uso de tunelamiento requiere el estado coherente de rutas entre IPv4 e IPv6. Por ejemplo, consideremos un paquete que comienza como paquete IPv6 y que después pasa a través de un tunel IPv4, es decir, se encapsula en un paquete IPv4 en el centro de su camino desde el origen a su destino. Este paquete debe ser enrutado (con enrutamiento IPv6) al extremo del túnel, atravesar el tunel como paquete IPv4 y después atravesar el resto del camino otra vez como paquete IPv6 Este paquete tiene que seguir claramente una ruta constante en todo el camino desde el origen hasta el destino. Las implicaciones de este proceso en el ruteo se discuten por separado para los túneles de ruteador a ruteador y los túneles de host a host, host a ruteador y ruteador a host.

Túneles de ruteador a ruteador

Los túneles de ruteador a ruteador se basan en la configuración manual de ambos extremos del túnel. Se debe configurar manualmente para saber las direcciones

asociadas al otro final de la conexión. Tales túneles también se refieren como túneles completamente manual configurados", puesto que ambos extremos de la conexión deben ser configurados manualmente.

En los túneles de ruteador a ruteador que manejan IPv6 sobre IPv4, los ruteadores tratan la conexión igual que si fuera una conexión normal punto a punto. Por ejemplo los protocolos dinámicos de ruteo como OSPF o BGP/IDRP pueden enviar la información de escalabilidad a través de esta conexión igual que en cualquier otro tipo de conexión. La decisión parta remitir un paquete a través de un túnel ruteador a ruteador configurado manualmente se hace de manera semejante a la decisión de remitir un paquete a través de cualquier otro tipo de conexión Específicamente, los paquetes se remiten basados en las rutas calculadas por protocolos estándares de ruteo. Estas rutas pueden utilizar conexiones normales o conexiones de túnel en cualquier combinación

El uso de los túneles ruteador a ruteador manualmente configurados tiene la ventaja de que la infraestructura subyacente es transparente a los protocolos que se remiten a través del túnel Por ejemplo, si IPv6 es tunelado a través de IPv4, después la infraestructura IPv4 se utiliza para la expedición del paquete IPv6, pero los detalles internos de la infraestructura IPv4 no importan a los ruteadores IPv6 ni a los protocolos de ruteo IPv6 También, todos los tipos de las direcciones IPv6 sin excepción pueden ser anunciadas en el ruteo IPv6 y tunelado a través de las redes IPv4. Puesto que se encapsulan los paquetes IPv6 solamente cuando viajan a través de los segmentos de la red que no utilizan IPv6, y se remiten según sus cabeceras nativas a otra parte, este método no obliga a no usar de políticas de ruteo, las cuales pueden emplearse a través de la porcion IPv6 del camino de datos.

Sin embargo, un túnel de ruteador a ruteador manualmente configurado difiere de una conexion normal en un aspecto importante: en muchos casos es probable tener un funcionamiento más bajo, tal como rendimiento de procesamiento más bajo o más retardado El uso de un protocolo de ruteo tal como RIP, que trata cada conexión por igual, podría conducir a las rutas subóptimas. Sin embargo este no es problema con protocolos de ruteo más flexibles como OSPF, que permite un rango dinamito ancho en la métrica asignada a cada conexión. Los túneles en los cuales se configuran manualmente ambos extremos de la conexión pueden en un principio, también ser utilizados con la configuración host a ruteador o host a host. Sin embargo, cuando un los hosts están implicados como extremos de un túnel, el requisito de cada extremo del túnel es que la configuración este asignada por default, por lo que hace que los túneles completamente manuales sean menos útiles para los ruteadores.

Tunelamiento Automático de Host a Host

Si el host origen y host destino hacen uso de direcciones IPv6 compatible con IPv4, entonces es posible que el tunelado automático sea utilizado para la trayectoria entera del host de origen al host destino. En este caso, el paquete IPv6 es encapsulado en un paquete IPv4 por el host origen, y remitido por los ruteadores como un paquete IPv4 todo el camino hasta el host destino Esta característica permite la implementación inicial de los hosts IPv6 antes de que cualquier ruteador sea actualizado.

Un host de origen puede hacer uso de tunelamiento automático de host a host a condicion de que todo lo que a continuación se indica sea cumplido:

- La dirección de origen es una dirección IPv6 compatible con IPv4
- La dirección destino es una dirección IPv6 compatible con IPv4
- El host origen no tiene conocimiento de ningún ruteador IPv6 vecino
- El host origen tiene conocimiento de uno o más ruteadores IPv4 vecinos

Si todos estos requisitos se cumplen el host de origen puede encapsular el paquete IPv6 en un paquete IPv4, usando una dirección IPv4 de origen extraída de la dirección IPv6 origen, y una dirección IPv4 destino extraída de la dirección IPv6 destino. Donde se utilice tunelamiento automático de host a host, el paquete se remite como paquete normal IPv4 por todo su camino, y es desencapsulado solamente por el host destino.

Túneles de Host a Ruteador

En algunos casos un host de doble capa IP (IPv6/IPv4) puede llegar a necesitar transmitir un paquete IPv6, pero puede no tener ningún ruteador IPv6 local que pueda utilizar para este propósito. En lugar de esto, el host puede utilizar tunelamiento a un ruteador IPv6. En algunos, el tunelado configurado por default sirve para poder ser usado para encapsular paquetes de IPv6 para transmitirlos desde el origen hasta un backbone de IPv6. El tunelado configurado por default es particularmente útil si el host origen no conoce a ningún ruteador IPv6 (esto implica que el paquete no puede ser enviado directamente como un paquete normal IPv6), y cuando el host destino no tiene una dirección IPv6 compatible con IPv4 (implica que la configuración host a host no puede ser usada).

El tunelado configurado de host a ruteador puede ser usado como una opción cuando el host origen no conoce ningún ruteador local IPv6.

Los túneles host a ruteador pueden ser logrados manualmente, configurando el host de doble capa IP con un dirección IPv4 que pueda utilizar para llegar hasta el backbone IPv6.

Para que la comunicación trabaje en ambas direcciones es necesario que un túnel ruteador a host tenga camino de vuelta. Esto requiere que cualquiera de ambos extremos del túnel se configuren manualmente, o que se utilice el tunelamiento automático en la dirección posterior (ruteador a host). Este último tipo de túnel se puede referir como "túnel semimanualmente configurado", puesto que la configuración manual se utiliza en una sola dirección, pero el tunelamiento automático se utiliza en la otra dirección.

Un túnel semimanualmente configurado ocurre cuando el host es configurado para que sepa encontrar el ruteador, pero el ruteador no se configura con ningún conocimiento específico del host y tunelamiento automático para encontrarlo. Esto, por supuesto, requiere que el host tenga una dirección IPv6 compatible con IPv4 y que el host esté configurado con una dirección IPv4 para utilizar un túnel al ruteador IPv6.

Un host origen puede hacer uso del túnel semimanualmente configurado de host a ruteador a condición de que se cumpla lo siguiente:

- La dirección de origen es una dirección IPv6 compatible con IPv4
- El host origen no tiene conocimiento alguno de ningún ruteador IPv4 vecino
- El host origen es configurado manualmente con una dirección IPv4 de un de un ruteador de doble capa, con la cual puede servir como extremo del túnel

Si todos los requisitos se cumplen, entonces el host origen puede encapsular el paquete IPv6 en un paquete IPv4, usando una dirección IPv4 de origen que se extraiga de la dirección IPv6 asociada al origen, y una dirección destino que corresponda a la dirección configurada del ruteador de doble capa que sirva como extremo del túnel.

Cuando el tunelado de host a ruteador es usado, el paquete es enviado como un paquete normal de IPv4 desde el host origen hasta el ruteador IPv6/IPv4, que sirve como un extremo del túnel, el paquete es desencapsulado por el ruteador de doble capa y es entonces enviado como un paquete de IPv6 por el extremo del túnel

Tunelamiento Automático de Ruteador a Host

Si se utiliza el tuneleamiento de un host a un ruteador para llegar al backbone IPv6, es tambien necesario utilizar tuneleamiento del ruteador al host. En este caso (a condición de que el host destino tenga una dirección IPv6 compatible con IPv4) la expedición normal IPv6 se puede utilizar para conseguir que el paquete de un ruteador de doble capa sea encapsulado al host destino

La expedición normal del paquete es directa en este caso: El ruteador de encapsulamiento crea la cabecera de encapsulamiento IPv4, usando una dirección IPv4 asignada a si mismo como la dirección IPv4 de origen, y con una dirección IPv4 destino extraida de la dirección IPv6 destino compatible con IPv4 El paquete encapsulado se emite del ruteador de encapsulamiento al host destino usando el ruteo normal IPv4.

El tuneleamiento de ruteador a host por lo general ocurre cuando uno o más ruteadores de doble capa están localizados entre el límite de una la red IPv4 y una red de doble capa IP En este caso los ruteadores fronterizos necesitan avisar en el ruteo IPv6 (red de doble capa) que ellos pueden interpretar ciertas direcciones IPv6 compatible con IPv4 que corresponden a las direcciones que existen en la red IPv4 En general esto requiere la configuración manual de los ruteadores fronterizos.

3.5.5 NAT-PT(Network Address Translation – Protocol Translation)

Este mecanismo basa su funcionamiento en el mecanismo conocido como Traducción de direcciones de red (NAT, *Network Address Translation*) y en el protocolo traductor de cabeceras (*Protocol translation*), aunque el funcionamiento del NAT que se maneja en este mecanismo, tiene alguna modificaciones al original ya que el NAT solo se refiere al trasladar direcciones IPv4 a IPv4 y en NAT-PT es traducir direcciones IPv4 dentro de direcciones IPv6 y viceversa.

El NAT-PT ofrece una directa solución punto a punto que es transparente al ruteo y a los protocolos de traducción de direcciones o mapeo de direcciones y de cabeceras (Address and Protocol Translation), que fueron analizados anteriormente. La desventaja es que usan métodos de traducción de direcciones y cabeceras, que se ha visto que no es muy facil de implementar en la practica.

El NAP-PT es también limitado por el factor que este puede solo traducir semánticas compartidas de las cabeceras de ambos protocolos, pero características especificas solo de IPv6 o características no soportadas por IPv6 no las soporta el NAT-PT.

Sesión de salida NAT-PT (IPv6 a IPv4)

En la figura 3.9 se muestra como trabaja el mecanismo de NAT-PT. El NAT-PT tiene banco de direcciones incluyendo subredes IPv4 (p. e. 120.130.26/24)

Digamos que un host A IPv6 se quiere comunicar con un host C IPv4, como se muestra en la figura 3.12

El host A crea un paquete con:

Dirección Origen FEDC BA98::7654.3210
Dirección Destino: PREFIx::132.146.243 30

Nota el prefijo PREFIX ::/96 es un prefijo preconfigurado que sirve solo para ser ruteable solo dentro un dominio representado por direcciones IPv4.

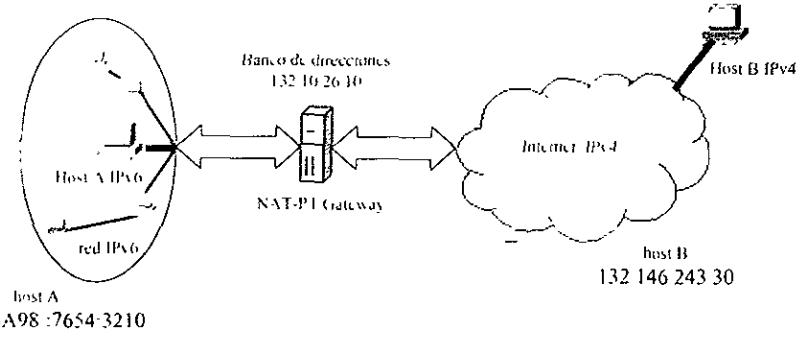


Fig. 3.9 - *El NAT-PT*

El paquete es ruteado hasta el gateway NAT-PT, donde la cabecera IPv6 tiene que ser traducida en una cabecera IPv4.

Digamos que si un paquete de salida no es de iniciación de sesión, el NAT-PT puede tomar algunas medidas como, almacenar algunos estados acerca de la sesión relacionada, incluyendo la asignación de dirección IPv4 y la traducción del paquete a IPv4, sino el paquete puede ser descartado.

Si el paquete es un paquete de iniciación de sesión el NAT-PT asigna una dirección localmente (por ejemplo 120.130.26.10) desde su banco de direcciones y el paquete es traducido a IPv4, mientras los parámetros de traducción son almacenados para ser utilizados durante la sesión, y la dirección IPv6 es retenida por el NAT-PT.

Resultando un paquete IPv4 con dirección origen 120.130.26.10 y con dirección destino 132.146.243.30 El trafico de regreso será reconocido por la sesión que se usa y por la información almacenada para la traducción, mientras que las direcciones después de la traducción serán, la dirección origen: 132.146.243.30 y la dirección destino: FEDC.BA98::7654:3210, y ahora este paquete puede ser ruteable dentro de la red de IPv6 normalmente.

Sesión de salida NAPT-PT (IPv6 a IPv4) con una sola dirección IPv4.

El NAPT-PT, puede permitir la comunicación de host IPv6 con host IPv4 usando una sola dirección IPv4. Lo que hace aquí para diferenciar las diferentes sesiones es usar distintos puertos de TCP/UDP según lo requiera la aplicación, por lo que es llamado " *Network Address Port Translation + Protcolo Translation*" (NAPT-PT)

Los puertos TCP/UDP de los host IPv6 son traducidos en puertos registrados TCP/UDP de direcciones IPv4.

Con el NAPT-PT se resuelve el problema que se tiene con el NAT-PT de que este podría caer cuando hay falta de direcciones IPv4 en el banco de direcciones, el NAPT-PT permite 63 000 sesiones de TCP y 6300 UDP antes de que falle .

Retomando la figura 3.9 del NAPT-PT supongamos que tenemos los mismos hosts que se presentan en ella.

El host A que desea comunicarse con el host B que se encuentra en una red de tipo IPv4, pero ahora el host A necesita que se habrá una sesión con un puerto TCP/UDP según la aplicación que se este manejando, tomemos que es de tipo TCP, entonces el host A crea un paquete con:

Dirección Origen: FEDC:BA98::7654.3210	Puerto origen TCP. 3017
Dirección Destino: PREFIx..132.146.243 30	Puerto destino TCP: 23

Cuando el paquete llega hasta el NAPT-PT que esta en un ruteador fronterizo, el NAPT-PT asignará uno de los puertos de TCP a ese enlace para realizar la comunicaciones y realizara el asignamiento de la dirección IPv4 para traducir el paquete quedando como se muestra a continuación

Dirección Origen:132 130.26 10	Puerto origen TCP. 1025
Dirección Destino: 132.146.243.30	Puerto destino TCP: 23

El trafico de regreso del host B será reconocido como el perteneciente a la misma sesión y será traducido como un paquete IPv6 teniendo lo siguiente

Dirección Origen: PREFIx.:132.146.243 30	Puerto origen TCP: 3017
Dirección Destino: FEDC:BA98::7654:3210	Puerto destino TCP: 3017

3.5.6 EL SIIT (Stateless IP/ICMP Translation)

El mecanismo conocido como SIIT evita el protocolo de traducción de direcciones, ya que usa direcciones mapeadas y direcciones IPv4 compatibles con IPv6, así como el protocolo traductor de cabeceras.

Con este mecanismo, nodos de tipo solo IPv6 pueden interoperar con nodos de tipo solo IPv4, mediante la adquisición de una dirección IPv4 temporal. Esta dirección IPv4 será usada como una dirección IPv4 compatible con IPv6, y los paquetes pasaran a través del protocolo traductor de cabeceras el cual traducirá las cabeceras de los paquetes entre IPv4 e IPv6 y traducirá las direcciones en estas cabeceras de una

direcciones IPv4 en una dirección IPv4 compatible con IPv6 o una dirección mapeada IPv6

El nodo IPv6 adquiere temporalmente una dirección IPv4 a través de un banco de direcciones IPv4 y es registrada temporalmente en el DNS. El protocolo DHCP, se utiliza para adquirir la dirección temporal con algunas modificaciones.

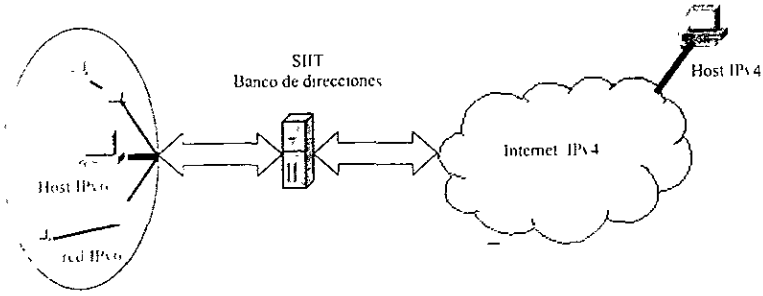


Fig. 3.10 - *Implementación del SIIT*

En la figura 3.10 se visualiza el mecanismo del SITT como es que se realiza la comunicación entre una red IPv6 y la red actual IPv4, y se observa que el mecanismo SITT se implementa en redes IPv6

El SIIT tiene algunas desventajas, pues cuando la etapa de transición de IPv6 a IPv4 este por finalizar, las redes en su mayoría serán de tipo IPv6 y la minoría será de tipo IPv4, y este mecanismo es solo para implementarse en redes de tipo IPv6, por lo cual sería complicado usarlo cuando la transición de IPv6 a IPv4 este por finalizar, por lo que el mecanismo SITT tendria que implementarse en todas las redes IPv6, que serian la mayoría, lo cual no es adecuado.

3.5.7 IPv6/IPv4 Network Address and Protocol Translation

El problema de los mecanismos de transición es que algunos solo se enfocan a proveer interoperabilidad entre sitios IPv6 con el internet IPv4, que es el internet actual, como lo es el mecanismo SIIT y NAT-PT, y otros se enfocan en la interoperabilidad entre sitios IPv4 con el internet IPv6 que es el futuro internet.

El mecanismo de IPv6/IPv4 Network Address and Protocol Translation , permite la comunicación en ambas formas, entre nodos de un sitio IPv4 con nodos de una red IPv6, y entre nodos de un sitio IPv6 con nodos de una red IPv4.

Cuando un sitio IPv6 se comunica con nodos IPv4, el sitio IPv6 debe ser configurado para que este pueda reconocer los paquetes IPv4 que llegan desde redes IPv4. Esto se logra a traves el protocolo de traducción de cabeceras que ya antes se analizo en el punto 3.5.2 además se utiliza un banco de direcciones IPv4 para asignar una dirección IPv4 al paquete IPv6, para que esta ocupe el papel de la dirección origen esto se hace a través del servidor traductor, asi cuando el paquete ya regresa desde el host IPv4 y llega al

servidor traductor, el traductor realiza la relación de direcciones IPv4 con su correspondiente dirección IPv6, ya sea una dirección IPv6 compatible con IPv4 o realiza el mapeo de la dirección. Con respecto a la asignación de direcciones IPv4 para las direcciones IPv6 se debe tener cuidado ya que esta asignación es dinámicamente y tan pronto como se termine una sesión, esta dirección se podrá utilizar en otro enlace, una de las desventajas es que si no existe un banco de direcciones lo suficientemente grande para la asignación de direcciones IPv4 esto podría ser ineficiente debido al gran numero de direcciones IPv6 que se tiene en comparación con IPv4, esto claro también va ligado al número de nodos que se tenga en el sitio IPv6.

En el otro caso donde el sitio IPv4 quiere comunicar con el internet IPv6, el paquete llegara al servidor traductor y en este se realizará la traducción de las cabeceras y así como el mapeo de las direcciones IPv4 en direcciones IPv4 compatibles con IPv6. Mandando asi el paquete hacia el sitio IPv6 con la dirección origen como la dirección IPv4 compatibles con IPv6, y cuando el paquete va de regreso del host IPv6 al IPv4, y llega al traductor este se encarga de colocar en el paquete la dirección destino solo los 32 bits menos significativos de la dirección IPv4 compatible con IPv6 que se coloco primeramente y la dirección IPv4 del servidor traductor como dirección origen.

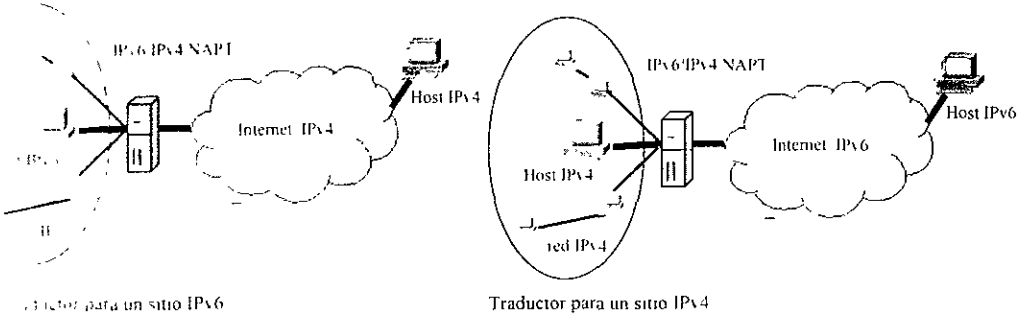


Fig. 3.11 - *Implementación del Mecanismo IPv6/IPv4 Network Address and Protocol Translation*

En la figura 3.11 se observa como el mecanismo IPv6/IPv4 Network Adrress and Protocol Translation, puede implementarse en redes IPv6 o en redes IPv4, por lo cual este mecanismo es mas flexible para la etapa de transición de IPv6 a IPv4.

3.6 EL PROTOCOLO ICMPV6

EL Protocolo IPv6 usa el protocolo ICMP (Internet Control Message Protocol) como el que es definido para IPv4 (RFC-792), solo que con algunas modificaciones, por lo que el resultado es el protocolo ICMPv6 (Internet Control Message Protocol version 6) y para el campo de Next Header de IPv6 tiene el valor de 58.

ICMPv6 es usado por nodos IPv6 para reportar errores encontrados en el procesamiento de paquetes, para eficientar las funciones de otras capas de internet, así como también para funciones de diagnostico como lo es la aplicación de ping ICMPv6 es una parte integral del protocolo IPv6 por lo que debe de ser implementada en el transcurso de la transicion en cada uno de los nodos IPv6.

3.6.1 Formato general del Mensaje

Los mensajes ICMPv6 son agrupados en dos clases mensajes de error y mensajes de tipo informativos Los mensajes de error se identifican por tener un valor cero en los bits de valor más significativo en el campo Type. Así que los mensajes de error van del rango 0 a 127, estos valores los puede tomar el campo Type según el tipo de mensaje de error que corresponda, los valores del campo Type para mensajes de tipo informativos toman los valores de 128 a 255

Aquí se definirán los siguientes tipos de mensajes ICMPv6:

Mensajes de Error ICMPv6

Destino inalcanzable (Destination Unreachable)
Paquete demasiado grande (Packet Too Big)
Tiempo excedido (Time Exceeded)
Problema de parámetro (Parameter Problem)

Mensajes de tipo informativo ICMPv6:

Solicitud de Eco (Echo Request)
Contestacion de Eco (Echo Reply)
Consulta de los miembros del grupo (Group Membership Query)
Reporte de los miembros del grupo (Group Membership Report)
Reducción de los miembros del grupo (Group Membership Reduction)

Existen más mensajes de tipo informativo como se dijo anteriormente, en este trabajo solo estudiaremos solo de 128-132, y al igual que los mensajes de error solo estudiaremos del 1-4

Cada mensaje ICMPv6 es precedido por una cabecera IPv6 y cero o mas cabeceras anexas de IPv6. La cabecera ICMPv6 es identificado con el valor 58 en el campo Next Header de la cabecera anterior.

El formato de la cabecera general de ICMPv6 se muestra en la figura 3.12.

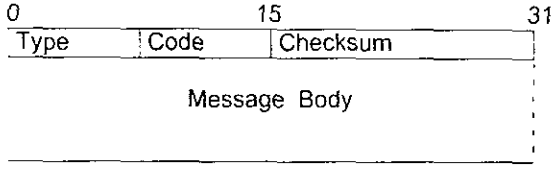


Fig. 3 12 - *Formato general del Mensaje ICMPv6*

El campo Type indica el tipo de mensaje. Su valor determina el formato de los datos restantes

El campo Code depende de tipo de mensaje, este es usado para crear un nivel adicional de mensaje.

El campo checksum es usado para detectar corrupción de datos en el mensaje ICMPv6 y partes de la cabecera IPv6

El checksum tiene una longitud de 16 bits, es uno de los complementos de la suma total del mensaje ICMPv6; para realizar el cálculo, el campo checksum se pone en cero.

El campo Message Body son características especificas de cada tipo de mensaje ICMPv6 que se encuentre usando

3.6.2 Determinación de la dirección origen del mensaje

Un nodo que envía mensajes ICMPv6 tiene que determinar las direcciones IPv6 origen y destino en la cabecera IPv6 antes de realizar el calculo para el campo checksum. Si el nodo tiene mas de una dirección unicast, el nodo debe elegir la dirección origen de las formas que se explica a continuación

Si el mensaje es una respuesta, para un mensaje enviado por una de las direcciones de un nodo, la dirección origen deberá ser la misma dirección del nodo

Si el mensaje es una respuesta para un mensaje enviado por una dirección multicast ó por un grupo anycast, en el cual el nodo es miembro, la dirección origen de la contestación deberá ser la dirección unicast correspondiente a la interface en la cual el paquete multicast o anycast fue recibido.

Si el mensaje es una respuesta para un mensaje enviado por una dirección que no corresponde a un nodo, la dirección origen podría ser la dirección unicast correspondiente al nodo que será mas útil para diagnosticar el error. Por ejemplo, si el mensaje de error es la respuesta a un paquete enviado que no se pudo completar exitosamente, la dirección origen seria la dirección unicast correspondiente a la interface en la cual el paquete falló

De lo contrario, la tabla de ruteo del nodo deberá ser examinada para determinar cual interface será usada para transmitir el mensaje a su destino, y la dirección unicast correspondiente a la interface para que sea usada como la dirección origen del mensaje.

3.6.3 Reglas de procesamiento del mensaje

Para la implementación completa de IPv6, debe tenerse en cuenta las siguientes reglas cuando se procesa los mensajes ICMPv6:

Si el mensaje de error ICMPv6 de tipo desconocido es recibido, éste debe ser pasado a capas superiores.

Si un mensaje de tipo informativo ICMPv6 de tipo desconocido es recibido este debe ser descartado

Cada mensaje de error ICMPv6 con el valor del campo Type menor a 128 incluye el paquete IPv6 involucrado (El paquete que cause el error), como colocando fuera el paquete de mensaje de error que exceda el minimo de MTU de IPv6

En estos casos donde el protocolo de la capa de internet es requerido para pasar un mensaje de error ICMPv6 a capas superiores, el tipo de protocolo de capa superiores es extraido desde el paquete original (contenido en el cuerpo del mensaje de error ICMPv6) , usado para seleccionar un proceso apropiado de capas superiores para manejar el error

Si el paquete original tiene una cantidad inusual de cabeceras suplementarias, es posible que el tipo de protocolo de capas superiores no este presente en el mensaje ICMPv6 por lo que el mensaje de error es desechado

Un mensaje de error ICMPv6 no debe ser enviado como resultado del recibimiento de:

- Un Mensaje de error ICMPv6
- Un paquete destinado a una dirección multicast IPv6, aunque existen dos excepciones para esta regla: 1) El mensaje de paquete demasiado grande, para permitir descubrir el MTU para el multicast IPv6, 2) El mensaje de problema de parámetro, con el valor 2 en el campo code, reporta una opción irreconocible de IPv6 teniendo en la opción Type los dos bit más significativos de este campo puestos en 10 respectivamente.
- Un paquete enviado como un broadcast
- *Paquete cuya dirección origen no es identificada como única de un solo nodo, como* una dirección multicast IPv6 o anycast.

Finalmente para él limite de ancho de banda y el costo que incurre al enviar mensajes de error ICMPv6, ya que un nodo IPv6 debe limitar el índice de mensajes de error ICMPv6 que envia Esto puede ocurrir cuando un nodo envía un torrente de mensajes de erroneos, por lo que incurrirá en demasiados mensajes de error , esto se puede implementar limitando el rango de transmisión de mensajes de error

3.6.4 Mensaje de Destino Inalcanzable (Destination Unreachable Message)

El mensaje de destino inalcanzable ICMPv6, es usado cuando el paquete no puede llegar hasta el nodo destino, ya sea por que no existe ruta para llegar o por una prohibicion del administrador de la red El formato de este tipo mensaje se muestra en la figura 3 13

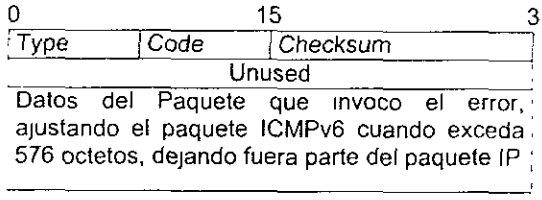


Fig. 3.13 - *Formato del Mensaje de destino Inalcanzable ICMPv6*

En la cabecera IPv6 la dirección destino(campo Destination Adrdress) a donde va dirigido el mensaje de error ICMPv6 debe de ser copiado desde el campo Source Address de la cabecera IPv6 del paquete que origino el error.

El campo Type identifica este tipo de mensaje ICMPv6 con el valor 1

El campo Code identifica cual fue la razón por la cual que se origino este tipo de mensaje de error, este campo tiene 5 valores definidos los cuales son:

- Code = 0 No hay ruta para el llegar al destino (este caso son para nodos que tienen una configuración por default y en su tabla de ruteo no pueden encontrar el ruteador por default)
- Code = 1 La comunicación con el destino esta administrativamente prohibida (Cuando se usan filtros como por ejemplo un Firewall)
- Code = 2 No asignado
- Code = 3. Dirección inalcanzable (este podria ser un el protocolo de la capa de transporte no este escuchando a la capa IP)
- Code = 4 Puerto inalcanzable

El campo Unused esta sin usar por los valores del campo code y debe ser inicializado con cero por el que realiza el envío y es ignorado por el que lo recibe.

Por ultimo se tiene un campo en el cual se incluyen datos del paquete original que provoco el mensaje, para poder identificar cuál paquete fue el que originó el mensaje de error, aunque sólo se toman 576 octetos en el caso de que el paquete exceda este tamaño

Un mensaje de destino inalcanzable podría ser generado por un ruteador o por un nodo IPv6, en respuesta de que un paquete no puede ser entregado a su dirección destino por razones de tránsito en la red.

3.6.5 Mensaje de Paquete demasiado grande (Packet Too Big Message)

Este tipo de mensaje de error ICMPv6, se origina cuando el paquete excede el MTU de una red por lo cual requiere que se fragmente por el nodo origen. Este mensaje de paquete demasiado grande, es enviado al nodo que origino el paquete que invoco el mensaje de error. El formato de este tipo de mensaje de error se observa en la figura 3.14

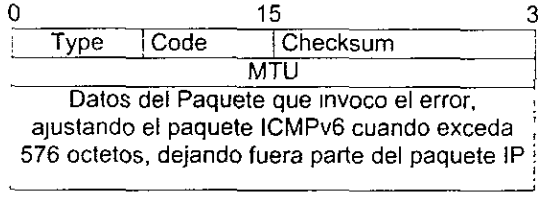


Fig. 3.14 - *Formato del Mensaje Paquete Demasiado Grande ICMPv6*

En la cabecera IPv6 la dirección destino (campo Destination Adrdress) a donde va dirigido el mensaje de error ICMPv6 debe de ser copiado desde el campo Source Address de la cabecera IPv6 del paquete que originó el error.

El campo Type identifica el tipo de mensaje con el valor 2.

El campo Code identifica la razón por la que se surge el mensaje de error, que es por exceder el MTU y lo identifica con el valor 0.

El campo MTU contiene la Unidad máxima de transmisión del segmento de red donde se origino este tipo de mensaje de error ICMPv6.

Por ultimo se tiene un campo en el cual se incluyen datos del paquete original que provoco el mensaje, para poder identificar cuál paquete fue el que originó el mensaje de error, aunque sólo se toman 576 octetos en el caso de que el paquete exceda este tamaño

Este tipo de paquete debe ser enviado por un ruteador en respuesta de que el paquete no puede ser enviado, porque su tamaño es más grande que el MTU permitido en la red. La información es usada para descubrir el MTU de la ruta, por dónde va ser enviado el paquete Este tipo de paquete debe ser pasado al proceso de capas superiores

3.6.6 Mensaje de tiempo excedido (Time Exceeded Message)

Este tipo de mensajes puede ser originado, debido a que el numero de saltos permitidos en el campo hop limit de la cabecera IPv6 (sección 4.1) para que el paquete pueda llegar a su destino fue excedido. Otra razón por la que puede originarse este tipo de mensaje ICMPv6, puede ser debido a que se excede el tiempo permitido para el reemzamble del paquete. El formato de este tipo de mensaje de error ICMPv6 se puede mostrar en la figura 3.15.

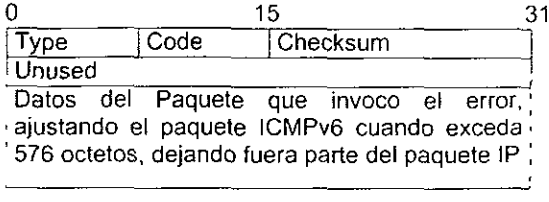


Fig. 3 15 - *Formato del mensaje de tiempo excedido ICMPv6*

En la cabecera IPv6 la dirección destino(campo Destination Adrdress) a donde va dirigido el mensaje de error ICMPv6 debe de ser copiado desde el campo Source Address de la cabecera IPv6 del paquete que originó el error.

El campo Type identifica el tipo de mensaje con el valor 3.

El campo Code identifica cuál fue la razón por la cual que se origino este tipo de mensaje de error, existen dos valores.

- Code = 0 Limite de saltos excedido en el transcurso de camino del paquete para llegar a su destino.
- Code = 1 Tiempo excedido en el reemsamble

El campo Unused esta sin usar por los valores del campo code y debe ser inicializado en cero por el que realiza el envio y es ignorado por el que lo recibe.

Por ultimo se tiene un campo en el cual se incluyen datos del paquete original que invoco el mensaje, para poder identificar cual paquete fue el que origino el mensaje de error aunque solo se toman 576 octetos en el caso de que el paquete exceda este tamano.

Si un ruteador recibe un paquete con el valor 0 del campo Hop Limit de la cabecera IPv6, o un ruteador lo decrementa hasta cero, el ruteador descarta el paquete y envía un mensaje de error ICMPv6 de tiempo excedido con código 0 (Campo code =0) hasta el nodo origen Este tipo de paquete debe ser pasado al proceso de capas superiores.

3.6.7 Mensaje de Problema de Parámetro (Parameter Problem Message)

Este tipo de mensaje es originado cuando al procesar el paquete se encuentra un parametro erróneo, en la cabecera IPv6 o en una de las cabeceras suplementarias de IPv6, el formato del mensaje de problema de parámetro se muestra en la figura 3.16

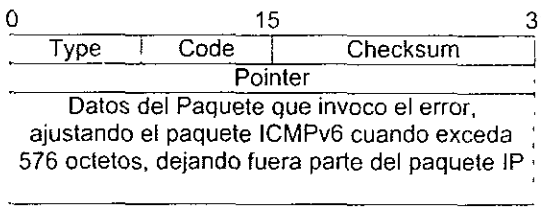


Fig 3 16 - *Formato del Mensaje de problema de parámetro ICMPv6*

En la cabecera IPv6 la dirección destino(campo Destination Adrdress) a donde va dirigido el mensaje de error ICMPv6 debe de ser copiado desde el campo Source Address de la cabecera IPv6 del paquete que origino el error.

El campo Type identifica el tipo de mensaje con el valor 4.

El campo Code identifica cual fue la razón por la cual que se origino este tipo de mensaje de error, existen dos valores.

- Code = 0 Algún campo de la cabecera IPv6 fue encontrado erróneo
- Code = 1 Tipo de cabecera siguiente encontrado irreconocible (No reconoce el valor del campo Next Header de IPv6)
- Code = 2 Opción IPv6 encontrada irreconocible

El campo Pointer identifica el octeto en el paquete donde fue detectado el error.

Por ultimo se tiene un campo en el cual se incluyen datos del paquete original que provoco el mensaje, para poder identificar cuál paquete fue el que originó el mensaje de error, aunque sólo se toman 576 octetos en el caso de que el paquete exceda este tamaño

Si un nodo IPv6 procesando un paquete encuentra algún problema con un campo de la cabecera IPv6 o en alguna de las cabeceras suplementarias, y el nodo no puede completar el proceso, el nodo manda un error de este tipo al que origino el paquete con valor en el campo code igual a 0, indicando el tipo y la localización del problema en el campo Pointer

3.6.8 Mensaje de Solicitud de Eco (Echo Request Message)

Este tipo de mensaje es de tipo informativo, y sirve para solicitar a un nodo una respuesta, este tipo de mensaje junto con el mensaje de contestacion de eco, son usadas en aplicaciones como Ping. *El formato de este mensaje se muestra en la figura* 3.17.

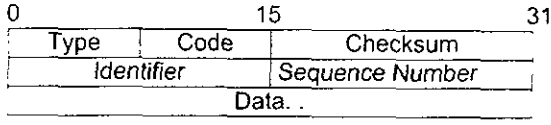


Fig. 3 17 - *Formato del mensaje de Solicitud de Eco ICMPv6*

En el campo de dirección destino de la cabecera IPv6 que va junto con este tipo de mensaje ICMPv6, se pone cualquier dirección IPv6 permitida.

El campo Type identifica el tipo de mensaje con el valor 128.

El campo Code contiene el valor cero que identifica la causa por el que se envía el mensaje, que es para solicitud de eco a un nodo.

El campo Identifier es un identificador para auxiliar a la contestación de eco correspondiente a la previa solicitud de eco. El valor puede ser cero

El campo Sequence Number contiene una secuencia de numeros para auxiliar a la contestación de eco correspondiente a la previa solicitud de eco. El valor puede ser cero.

Cada nodo debe implementar una función de respuesta de Eco ICMPv6, que reciba solicitudes de eco y envíe la contestación correspondiente de Eco

3.6.9 Mensaje de Contestación de Eco (Echo Reply Message)

Este tipo de mensaje es enviado, en base a un mensaje previo de solicitud de eco, que como ya se dijo se usa en alguna aplicaciones como lo es Ping El formato del mensaje se puede muestra en la figura 3.18.

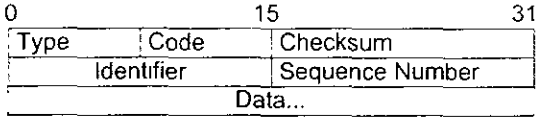


Fig 3 18 - *Formato del Mensaje de Contestación de Eco ICMPv6*

El campo Destination Adrdress de la cabecera IPv6, es copiado desde el campo Source Address de la cabecera IPv6 del paquete que origino el mensaje de contestacion de eco.

El campo Type identifica el tipo de mensaje con el valor 129.

El campo Code contiene el valor cero que identifica la causa por el que se envía el mensaje, que es para contestación de eco a un nodo.

El campo Identifier, contiene un numero identificador que es copiado del campo Identifier del mensaje de solicitud de eco previamente recibido, quien solicita el mensaje de contestación de eco.

Campo Sequence Number, copia el numero de secuencia del campo Sequence Number del mensaje de solicitud de eco previamente recibido, quien solicita el mensaje de contestación de eco.

Cada nodo debe implementar una función de respuesta de Eco ICMPv6, que reciba solicitudes de eco y envíe la contestación correspondiente de Eco. Este tipo de mensajes como el de solicitud de eco es funcional para pruebas de diagnostico, como lo es la aplicacion de Ping o Tracerouter.

La dirección origen de una contestación de eco que se envía en respuesta de una solicitud de eco de tipo unicast debe ser la misma dirección destino del mensaje de solicitud de eco.

Una contestación de eco puede ser enviada en respuesta para un mensaje de solicitud de eco enviado por una dirección multicast IPv6 La dirección origen de la contestación *debe ser la dirección unicast correspondiente a la interface por la cual el* mensaje de solicitud de eco multicast fue recibido.

Los datos recibidos en el mensaje de solicitud de eco deben ser regresados, completamente y sin modificaciones en el mensaje de contestación de eco.

3.6.10 Mensajes de miembros de grupo(Group Membership Messages)

Estos mensajes forman parte de la implementación de IP multicasting y consta de los siguientes tres tipos de mensajes

Los mensajes de consulta de miembros del grupo (Group Membership Query) , son enviados para descubrir cual grupo de host es miembro en su red local, los host generan un mensaje de reporte de miembros del grupo (Group Membership Report), cuando un *host termina su estancia en el grupo envía un mensaje de reducción de miembros del* grupo (Group Membership Reduction).

Los mensajes están definidos por el valor en el campo type; 130 para el mensaje de consulta de miembros del grupo (Group Membership Query), para el mensaje de reporte de miembros del grupo (Group Membership Report) e el valor de 131 y para el mensaje de reducción de miembros del grupo (Group Membership Reduction) e el valor de 132.

El formato del mensaje se muestra en la figura 3.19, donde se pueden ver los campos que contiene

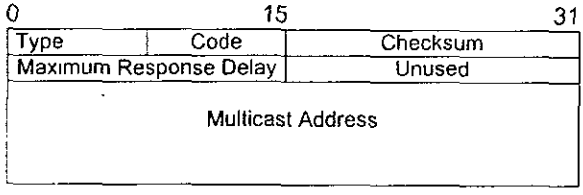


Fig. 3.19 - *Formato del Mensaje Miembros de Grupo ICMPv6*

Para mensajes de consulta, el campo de *Maximum Response Delay* tiene una longitud de 16 *bits y contiene el tiempo máximo* de retraso de la respuesta de un mensaje de reporte de miembros de un grupo, para mensajes de Reporte y de reducción este campo es inicializado en cero e ignorado por el que recibe el paquete El campo *Unused* puede ser inicializado en cero por el que envía el paquete e ignorado por el que lo recibe.

El campo de Multicast Address contiene la dirección de grupo del cual se está enviando el mensaje, en el mensaje de consulta de miembros del grupo este campo puede ser inicializado en cero, lo cual implica consultar todos los grupos Este campo tiene una longitud de 128 bits.

3.7 6BONE

La actual red del protocolo IPv6 es conocido comúnmente con el nombre de 6bone, donde el principal interés es de la implementación de IPv6, donde aplicaciones IPv6 están en prueba *corriendo en paralelo con el actual internet de IPv4. El 6bone esta en* crecimiento constante, actualmente el 6bone llega a más 30 diferentes paises de Norte America, Europa y Asia

Algunos de los paises donde llega el 6bone son los siguientes

Austria	República Checa	Hungría	Polonia
Australia	Alemania	Irlanda	Portugal
Belgica	Dinamarca	Italia	Rumania
Bulgaria	España	Japón	Federación Rusa
Brasil	Finlandia	Corea	Suecia
Canada	Francia	Lituania	Singapur
Suiza	Gran Bretaña	México	Eslovenia
Camerun	Grecia	Los Países Bajos	Eslovaquia
China	Hong Kong	Noruega	Taiwan
Estados Unidos		Sudáfrica	

El 6bone está centrado en proveer una serie de políticas y procedimientos necesarios para llevar a cabo pruebas para proveer un transporte IPv6 El 6bone es operado bajo investigación y desarrollo de políticas aceptables de uso.

Con respecto a las aplicaciones nativas de IPv6 ya se tienen algunas como lo es FTP, telnet, ping, que pueden correr bajo IPv4 o IPv6, pero muchas empresas como lo es Apple Computer, IBM, Digital Equipment Corporation, *FTP Software*, Linux, *Microsoft, Silicon* Graphics, Sun Microsystems, entre muchas otras se encuentran desarrollando aplicaciones nativas de IPv6. Algunas aplicaciones que ya se dieron a conocer son el cliente de *HTTP, NFS* entre muchos otros, e *incluso sistemas operativos que ya soportan* IPv6, como lo es la version de Linux 2.1.8, IRIX, y se esta en prueba en Solaris así como en Windows NT.

También algunas compañías ya están implementando en sus equipos el protocolo IPv6, como lo es 3com, CISCO, y otras empresas, sólo que aún está en la etapa de *pruebas en el 6bone.*

Los enlaces del backbone del 6bone, pueden operar usando rutas estáticas o uno de los *protocolos soportados por IPv6, como lo es RIPng* (Routing Information Protocol-Next Generation), BGP4+ (Border Gateway Protocol version 4), OSPF .

En la figura 3.20 se muestra de qué compañías está formado el backbone del 6bone, así como los protocolos que usan o si usan rutas estáticas, aunque cabe mencionar que el 6bone sigue teniendo un gran crecimiento.

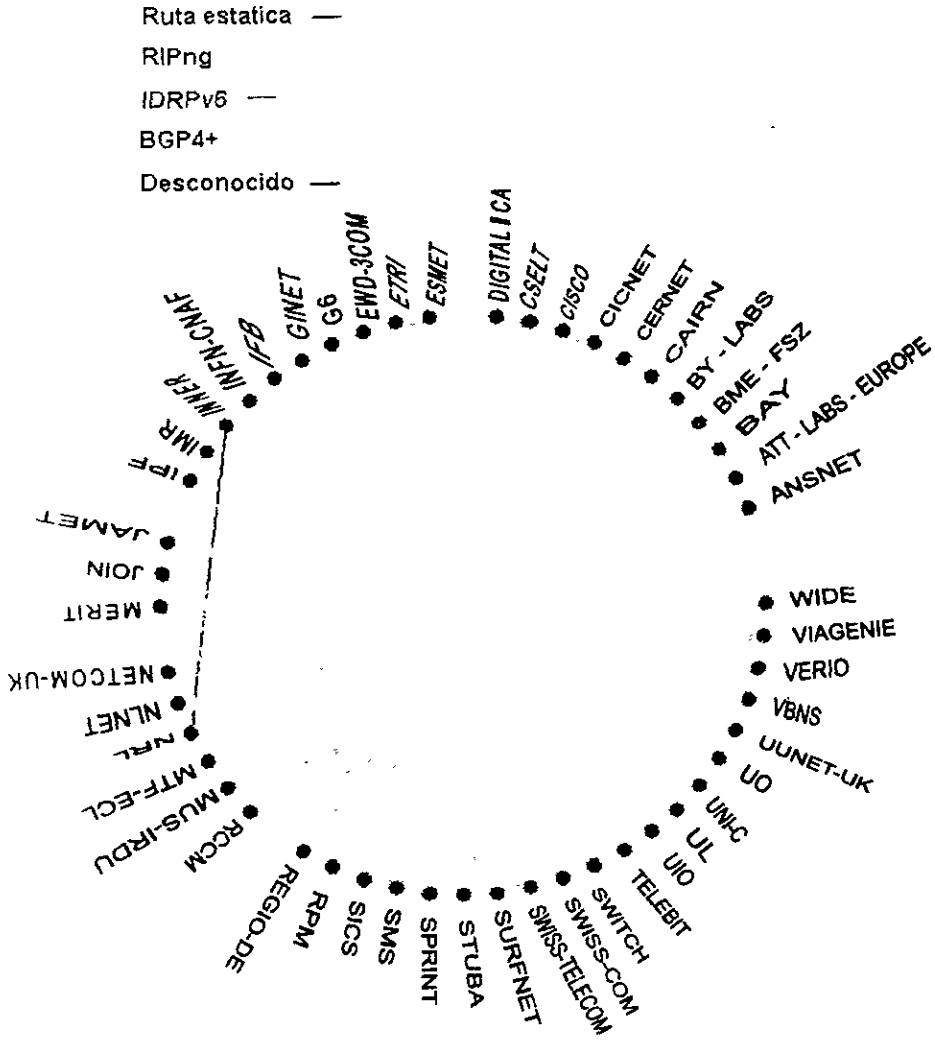


Fig. 3.20 - *Estructura del Backbone del 6bone*

Capítulo IV

MANEJO DE LA INFORMACIÓN EN IPV6

4.1 FORMATO DE CABECERA IPV6

El paquete IPv6 es transmitido dentro del frame de la red local como en el caso de IPv4, sin embargo las cabeceras de IPv6 consisten de dos partes:

- La cabecera base IPv6
  Las cabeceras suplementarias adicionales, que pueden ser opcionales.

Sin embargo con o sin las cabeceras suplementarias, el tamaño del frame de la red local debe ser respetado, por ejemplo, la máxima cantidad de datos que permite ser transmitidos en un frame Ethernet es de 1500 octetos, tomando en cuenta si se añaden cabeceras suplementarias en al paquete, un numero menor de aplicaciones podran ser enviadas, para asi no sobrepasar el numero de octetos permitidos.

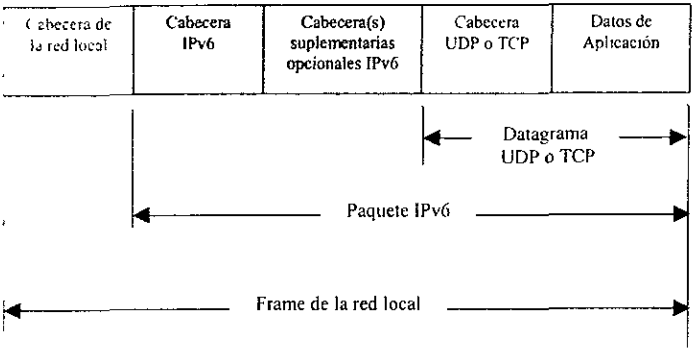


Fig. 4.1 -*Transimisión de frame con IPv6*

En la figura 4.1 podemos observar el formato del frame de una red local, el cual contiene el paquete IPv6, y el datagrama de TCP o UDP según corresponda.

La cabecera IPv6 tiene una longitud de 40 octetos, con 8 campos y dos direcciones, comparado esta con la cabecera IPv4, la cual tiene una longitud de 20 octetos, tiene 10 campos dos direcciones y opciones adicionales.

El formato base de la cabecera IPv6 se muestra en la figura 4.2

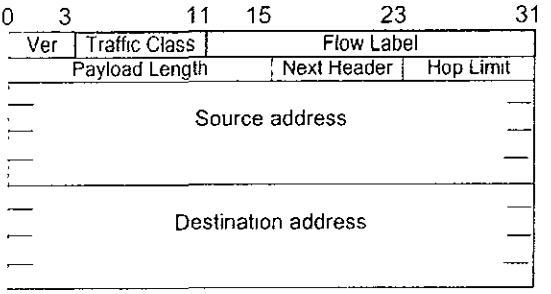


Fig. 4.2 - *Cabecera IPv6*

El campo Version (Ver) tiene una longitud de 4 bits, identifica la versión del protocolo IP, para IPv6 el valor es de 6

El campo Traffic Class tiene una longitud de 8 bits y está destinado para nodos originadores del paquete y/o ruteadores que reenvían el paquete para identificar y distinguir entre diferentes clases o prioridades de paquetes IPv6 Este campo primeramente fue llamado Priority y sólo constaba de una longitud de 4 bits, ahora este campo se conoce como Traffic Class y tiene una longitud de 8 bits, pues se le disminuyó 4 bits al campo Flow Label quedando con sólo 20 bits de longitud y con 8 bits el campo Traffic Class Con respecto a la clasificacion de los tipos de tráfico, aun no se publican debido a que se están haciendo experimentos para definir claramente las diferentes clases de trafico

El Campo Flow Label, se menciono anteriormente que constaba de 24 bits pero debido a que se aumento la longitud del campo Traffic Class, este campo se disminuyo a 20 bits de longitud.. Este campo se usa para cuando un host requiera un especial manejo por parte de los ruteadores para ciertos paquetes, tales como, no tener una calidad de servicio por default o un servicio de tiempo real.

Un flujo es una secuencia de paquetes enviados desde un origen en particular hasta un destino particular ya sea unicast o multicast, para el cual el nodo origen desea un manejo especial por los ruteadores que intervienen en el envío del paquete. La naturaleza de este especial manejo del paquete podrá ser comunicada a los ruteadores por un protocolo de control, como por ejemplo un protocolo de reserva de recursos, o por información contenida dentro de los mismo paquetes pertenecientes al flujo, por ejemplo la cabecera Hop by Hop.

Un flujo es identificado por la combinación de la dirección fuente y el identificador del campo Flow Label tomando en cuenta que este debe ser distinto de cero ya que los paquetes que no pertenezcan a un flujo llevan el valor de cero en este campo, por lo que los paquetes que pertenezcan a un mismo flujo de trafico deben ser enviados con la misma dirección origen, dirección destino, mismo valor del campo Flow Label, misma cabecera de Ruteo y cabecera Hop by Hop (con excepción del valor del campo next header en las cabeceras), entonces todos deben ser originados con el mismo contenido de esta cabecera. El valor de este campo es asignado por el nodo emisor, y se escoge de

forma (pseudo) aleatoria del rango 1 hasta FFFFF (hexadecimal) para formar un conjunto de bits Esta etiqueta de flujo es usada como una llave para que los ruetadores busquen el estado asociado con el flujo. Los tipos de estados de calidad de servicio del flujo, aun se estan redefiniendo debido al cambio de tamaño del campo de 24 a 20 bits.

El Campo Payload Length consta de 16 bits de longitud, el cual se encarga de medir la longitud del paquete, obteniéndola en octetos, considerando las cabeceras suplementarias parte de la información útil para tomarla en cuenta en la realizacion del cálculo junto con protocolos de capas superiores como TCP, FTP, etc.

Este campo Payload Length es similar al campo Total Length de IPv4, excepto que se realizan dos diferentes medidas en IPv6, El campo Payload Length en IPv6 mide los datos y después las cabeceras, mientras que el campo Total Length en IPv4 mide los datos y la cabecera al mismo tiempo

Aquí se permiten paquetes mayores que 65,535 y son llamados jumbo payloads. Para indicar un paquete jumbo payload, el valor del campo Payload Length es puesto con cero y la longitud del paquete es especificada dentro de las opciones que son transmitidas en la cabecera Hop by Hop

El campo Next Header consta de 8 bits de longitud, identifica la cabecera que sigue inmediatamente, este campo usa los mismo valores que se usan en el campo Protocol de IPv4

En la Tabla 4.1 se muestran algunos valores de las cabeceras que se usan en este campo

Valor	Tipo de Cabecera
0	Hop by Hop Options
1	ICMPv4
4	IP en IP (encapsulamiento)
6	TCP
17	UDP
43	Ruteo (Routing)
44	Fragmentación (Fragment)
50	Encapsulating Security Payload
51	Autentificación (Authentication)
58	ICMPv6
59	Ninguna cabecera siguiente
60	Opciones de Destino (Destination Options)

Tabla 1.1 – *Valores del Campo Tipo de Cabecera*

En el paquete IPv6, las cabeceras suplementarias también emplean el campo Next Header

La demultiplexación del campo Next Header de la cabecera IPv6 invoca un modulo para procesar la primera cabecera suplementaria o las cabeceras de los protocolos superiores, en el caso de que no existan cabeceras suplementarias

El Campo Hop Limit tiene una longitud de 8 bits, y es decrementado uno por uno, por cada nodo que reenvia el paquete durante el transcurso del camino hasta al destino. Cuando el valor de este campo es igual a cero el paquete es descartado y un mensaje ICMPv6 es regresado hasta el origen de donde se emitio el paquete. Este campo es similar al campo TTL de IPv4, con la diferencia de que el campo Hop Limit mide el numero maximo de saltos que pueden ocurrir cuando el paquete es reenviado por varios nodos durante su viaje para llegar a su destino, en cambio el campo TTL de IPv4 puede ser medido en número de saltos o en base a segundos.

El campo Source Address tiene una longitud de 128 bits, contiene la dirección origen que identifica al nodo originador del paquete. El formato del tipo de direcciones se define en el capitulo 5

El campo Destination Address tiene una longitud de 128 bits, contiene la dirección destino que identifica el destino donde debe de llegar el paquete. Una distinción importante es que el destinantario puede no ser el ultimo destinatario, como en la cabecera de ruteo (Routing Header) puede ser empleado para especificar el camino que el paquete debe tomar desde el origen, a través de destinos intermediarios, hasta su final destino El formato del tipo de direcciones se define en el capitulo 5.

4.2 CABECERAS SUPLEMENTARIAS

Con el protocolo IPv6, la información opcional que no es requerida que se coloque en la cabecera base IPv6, es codificada en cabeceras por separado llamadas cabeceras suplementarias, que pueden estar colocadas entre la cabecera IPv6 y la cabecera(s) de capas superiores del frame Cada cabecera siplementaria tiene un número de identificacion, que va en el campo de Next Header de cada una de las cabeceras.
Un paquete IPv6 puede transmitir cero, uno o más cabeceras suplementarias.

En la Fig. 4 3 se muestra un ejemplo de cómo pueden ir colocadas las cabeceras suplementarias en el paquete IPv6.

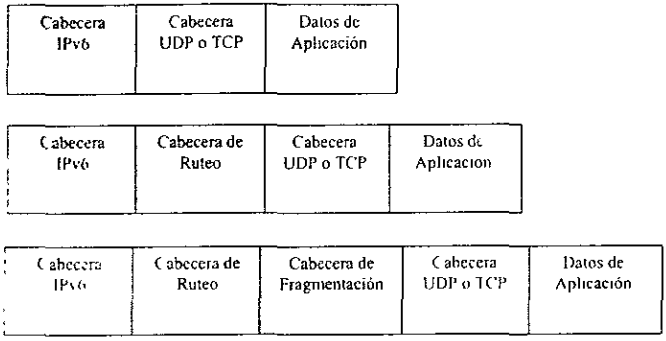


Fig. 4.3. - *Secuencia de las cabeceras IPv6*

Con excepción de la cabecera Hop by Hop, las cabeceras suplementarias no son examinadas o procesadas nodo a nodo, durante el transcurso del paquete a su destino Sino hasta que el paquete arriva a su destino o destinos finales(dependiendo del tipo de direccion si es Unicast, Anycast o Multicast, que es identificado por la direccion destino que va en el campo Destination Address).

La cabecera Hop by Hop la cual transmite información que debe ser examinada y procesada por cada nodo a lo largo del camino del paquete, incluyendo los nodos origen y destino La cabecera Hop by Hop cuando se presenta debe ir inmediatamente despues a la cabecera base IPv6, su presencia se identifica por el numero cero en el campo Next Header del paquete IPv6

El contenido y semántica de cada cabecera suplementaria determina si se procesa o no la cabecera siguiente. Por lo tanto las cabeceras suplementarias deben ser procesadas estrictamente en el orden que deben de aparecer en el paquete

La implementación completa de IPv6 consta de las siguientes cabeceras suplementarias.

- Hop by Hop (options)
- Ruteo (tipo cero)
- Fragmentación
- Destination Options
- Autenticación
- Encapsulating Security Payload

Cuando más de una cabecera suplementaria es usada en el mismo paquete, deben *aparecer en el siguiente orden*

- Cabecera IPv6
- Cabecera Hop by Hop (options)
- Cabecera Destination Options (Nota 1)
- Cabecera de Ruteo
- Cabecera de Fragmentación
- Cabecera de Autenticación
- Cabecera Encapsulating Security Payload
- Cabecera Destination Options (Nota 2)
- Cabeceras de capas superiores

Nota 1 Son Opciones contenidas en la cabecera Destination Option para ser procesadas por el primer destino que aparece en el campo Destination Address de la cabecera IPv6 además de los subsecuentes destinos listados en la cabecera de Ruteo

Nota 2 Son Opciones contenidas en la cabecera Destination Option para ser procesadas solo por el destino final.

Cada cabecera suplementaria sólo puede aparecer una vez, con excepción de la cabecera *Destination Options, la cual puede aparecer dos veces.*

Si las cabeceras de capas superiores es otra cabecera IPv6 (en el caso de cuando IPv6 este siendo tunelado o encapsulado en IPv6), este puede ser seguido por sus propias cabeceras suplementarias, las cuales son sujetas al mismo orden antes citado.

Opciones

Dos de las cabeceras suplementarias, Hop by Hop y Destination Options, transmiten un numero de variables que pueden llevar una o más opciones *que identifican parámetros* u operacion de la red. Estas opciones son codificadas usando un formato de TLV (Type Length Value), con el formato que se muestra en la figura 4.4.

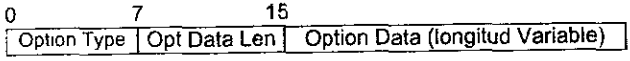


Fig. 4.4 - *Formato TLV*

El campo *Option Type tiene una longitud* de 8 bits la cual identifica el tipo de opción, el campo Option Data Length tiene una longitud de 8 bits, la cual especifica la longitud del campo Option Data, medida en octetos. El campo Option Data especifica el tipo de opción de datos

Dos de los bits más significativos del campo Option Type especifica la accion a tomar,de com manejar las opciones que son irreconocibles para el procesamiento del paquete, como se muestra en la tabla 4.2

Valor	Acción a tomar
00	Ignorar la opción y continuar procesando el paquete
01	Descarta el paquete
10	Descarta el paquete y enviar un Mensaje ICMP hasta el destino de tipo Problema de parámetro
11	Descarta el paquete y enviar un Mensaje ICMP hasta el destino(solo sí el destino no fue de tipo multicast) de tipo Problema de parámetro

Tabla 4.2 – *Option Type primero y segundo bit*

El tercer bit más significativo del campo Option Type especifica si puede ó no cambiar el campo Option data durante el transcurso a su destino fina,como se muestra en la tabla 4.3

Valor	Acción a tomar
0	La Opción de datos no puede cambiar en el transcurso del paquete a su destino
1	La Opción de datos puede cambiar en el transcurso del paquete a su destino

Tabla 4.3 - *Option Type tercer bit*

Existen dos opciones de relleno que son usadas para los requerimientos de alineación, para asegurar la alineación en múltiplos de 8 octetos del campo Option en la cabecera que se este manejando y este va dentro de estas opciones. La opción llamada Pad1 es usada para insertar un octeto en el área de opciones de la cabecera, esta se especifica con el valor cero en el campo type. El formato de la opción Pad1 se muestra en la figura 4.5

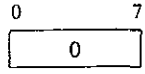


Fig. 4.5 - *Formato de la opción Pad1*

Existe la opción PadN que es usada para insertar dos o más octetos de relleno dentro del área de opciones de la cabecera que sé este manejando, ya sea el Hop by Hop o Desination Options, esta opción tiene el valor de 1. El formato de la opción PadN se muestra en la figura 4.6.

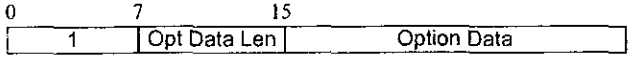


Fig. 4.6 - *Formato de la opción PadN*

Por ejemplo digamos que se tiene un paquete con una opción Y que requiere tres campos de datos, uno de longitud de 4 octetos otro de longitud de 2 octetos y otro con longitud de 1 octeto; este paquete tendrá el formato que se muestra en la figura 4.7.

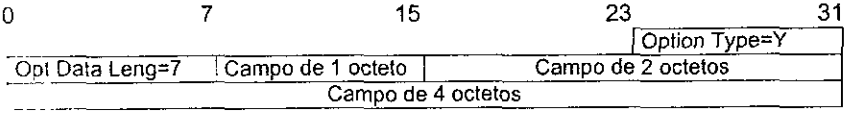


Fig. 4.7 - *Formato de opciones.*

Digamos que estas opciones irán en un encabezado Hop by Hop o Destination Options, quedando con el formato de la figura 4.8.

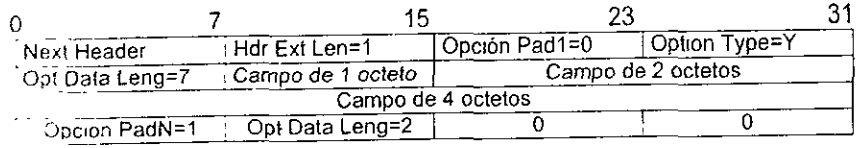


Fig 4.8 - *Cabecera Hop by Hop, con sus respectivas opciones.*

En el formato de la figura 4.8, podemos observar que se necesitaron una opción Pad1 , una Opción PadN con longitud de dos octetos para que el formato de la cabecera quedara con un múltiplo de 8 octetos, que en este caso quedo con una longitud de 16 octetos

4.2.1 Cabecera Hop by Hop.

Este tipo de cabecera transporta información opcional que debe ser examinada por cada nodo durante el trayecto del paquete hasta su destino final Cuando este tipo de cabecera se presenta debe estar inmediatamente despues de la cabecera base IPv6; esta cabecera es identificada por el valor 0 del campo Next Header de la cabecera anterior y tiene el formato que se muestra en la figura 4.9.

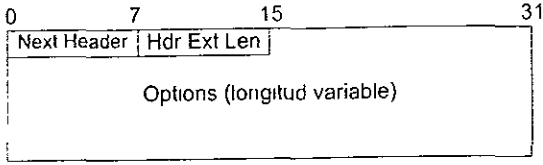


Fig. 4.9 - *Formato de la Cabecera Hop by Hop*

En este formato se puede observar :

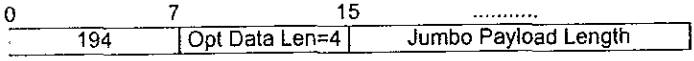
El campo next header tiene una longitud de ocho bits e identifica la cabecera siguiente, este campo usa los mismos valores usados para IPv4.

El campo Header Extension Length (Hdr Ext Len) tiene una longitud de 8 bits y mide la longitud de la cabecera Hop by Hop en unidades de 8 octetos, no tomando en cuenta los primeros 8 octetos

El campo Options tiene una longitud variable y consta de opciones que pueden servir para configuracion o comunicación entre los nodos, algunas opciones fueron antes

citadas Esta cabecera debe ser completada con rellenos, para que tenga una longitud que sea múltiplo de 8 octetos(64 bits).

La opción Jumbo Payload, la cual es usada para enviar paquetes IPv6 más grandes que 65535 octetos, es definida por Option Type=194, Opt Data Len = 4(octetos) y el campo Jumbo Payload Length de 4 octetos de longitud, el cual transmite la longitud del paquete Jumbo Packet en unidades de octetos (excluyendo la cabecera base, pero incluyendo la cabecera Hop by Hop Options y otras cabeceras). El campo Payload Length de la cabecera base IPv6, tiene el valor cero para indicar una especial condición cuando se usa el paquete Jumbo Payload; este se muestra en la figura 4.10.



4.10 - *Formato de las opciones para Jumbo Payload*

4.2.2. Cabecera Destination Options

La cabecera de Destination Options transmite información opcional que necesita ser examinada por los nodos destinatarios que aparecen en el paquete. La presencia de la cabecera de Destination Options es identificada por el valor 60 en el campo Next Header de la cabecera anterior. Esta cabecera tiene dos campos y opciones extra, como se muestra en el la figura 4.11.

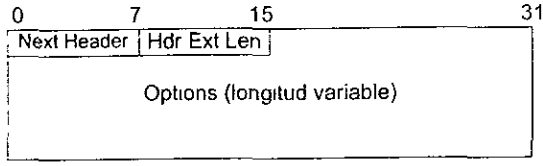


Fig. 4 11 - *Formato de la cabecera Destination Options.*

En este formato se puede observar.

El campo Next Header tiene una longitud de 8 bits e identifica la cabecera siguiente a la cabecera de Destination Options.

El campo Header Extension Length (Hdr Ext Len) tiene una longitud de 8 bits y mide la longitud de la cabecera Destinations Options en unidades de 8 octetos, no tomando en cuenta los primeros 8 octetos.

El campo Options tiene una longitud variable así que la longitud total debe ser una longitud de un valor múltiplo de 8 octetos, para lograr esto se agregan campos de relleno (Pad)

4.2.3. Cabecera de Ruteo.

La cabecera de ruteo es usada por el nodo IPv6 origen para listar una o más nodos intermedios, por los cuales el paquete tendra que pasar para que el paquete llehue hasta su destino final. La cabecera de ruteo es identificada por el valor 43 en el campo Next Header de la cabecera anterior

El formato General de esta cabecera se muestra en la figura 4.12.

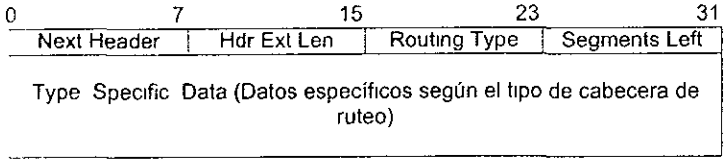


Fig. 4.12 - *Formato general de la cabecera de Ruteo.*

En este formato se puede observar:
El campo Next Header tiene una longitud de 8 bits y sirve para identificar la cabecera siguiente

El campo Header Extension Length, se encarga de medir la longitud de la cabecera en unidades de 8 octetos, no incluyendo los primeros 8 octetos

El campo Routing Type tiene una longitud de 8 bits e identifica la parte variante de la cabecera de ruteo

El campo Segment Left tiene una longitud de 8 bits, contiene el número de nodos intermedios por los que el paquete necesita pasar antes de arrivar a su destino final.

El campo Type Specific Data tiene una longitud variable y es de una valor entero múltiplo de 8 octetos, su formato está determinado por el tipo de ruteo del campo Routing Type

Aquí sólo se considerá la cabecera de ruteo tipo cero, cuyo formato se muestra en la figura 4.13

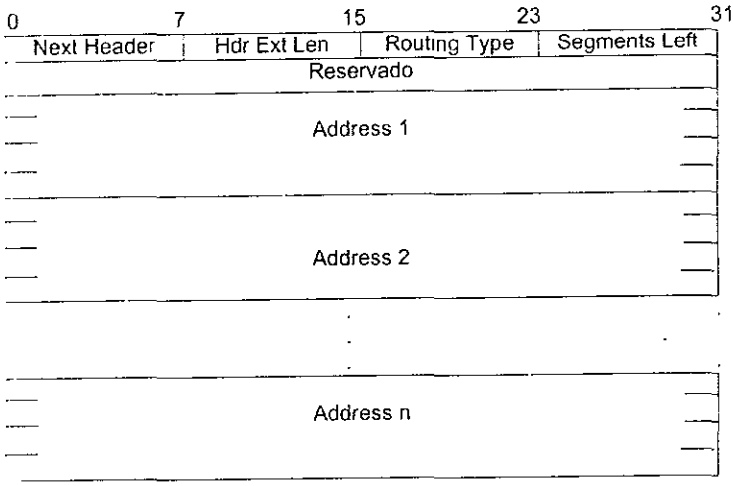


Fig. 4.13 - *Formato de la cabecera de Ruteo tipo 0.*

En este formato se puede observar

El campo Next Header tiene una longitud de 8 bits y sirve para identificar la cabecera siguiente

El campo Header Extension Length, se encarga de medir la longitud de la cabecera en unidades de 8 octetos, no incluyendo los primeros 8 octetos. En este tipo de cabecera tiene un valor de 2 veces el numero de direcciones

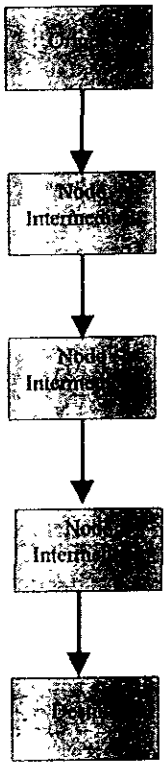
El campo Routing Type tiene una longitud de 8 bits e identifica la parte variante de la cabecera de ruteo.

El campo Segment Left tiene una longitud de 8 bits, contiene el número de nodos intermedios por los que el paquete necesita pasar antes de arrivar a su destino final.

El campo Reservado tiene una longitud de 32 bits, se inicializa en cero en la transmisión y es ignorada en la recepción

Cada uno de los campos de direcciones (Address) tiene una longitud de 128 y es numerado de 1 hasta n.

A continuación se da un ejemplo de la función de la cabecera de ruteo, a través de la figura 4.14



Cabecera IPv6	Cabecera de Ruteo
DO = O DD = I1	HEL = 6 SL = 3 A[1] = I2 A[2] = I3 A[3] = D
DO = O DD = I2	HEL = 6 SL = 2 A[1] = I1 A[2] = I3 A[3] = D
DO = O DD = I3	HEL = 6 SL = 3 A[1] = I1 A[2] = I2 A[3] = D
DO = O DD = D	HEL = 6 SL = 3 A[1] = I1 A[2] = I2 A[3] = I3

A[n] Dirección IPv6
DD Dirección Destino
DO Dirección Origen
HEL Campo Header Extension Length
In Nodo Intermedio n
SL Campo Segments Left
O Origen
D Destino

Fig. 4.14 - *Uso de la Cabecera de Ruteo*

En dicha figura se puede observar que hay tres nodos intermedios entre el nodo origen y el nodo destino.

Para que el paquete viaje desde el nodo origen hasta el nodo intermedio 1, la cabecera base IPv6 usa como dirección origen la dirección del nodo que emitió el paquete, y como dirección destino la dirección del nodo intermedio 1 . La cabecera de ruteo especifica una longitud de 6 unidades de 8 octetos en el campo Header Extension Length, debido a que se usan tres direcciones en la cabecera de ruteo y como se dijo anteriormente en

esta cabecera la longitud es el número de direcciones multiplicado por dos, ya que no se toman en cuenta los primeros 8 octetos, solo las direcciones que va en la cabecera de ruteo

En este caso, en la cabecera de ruteo el campo Segments Left tiene un valor de 3, ya que necesita pasar por 3 nodos antes de llegar a su destino final el Nodo intermedio 1, despues Nodo intermedio 2, Nodo intermedio 3 y por ultimo el nodo de destino final.

Para que el paquete viaje del nodo intermedio 1 al nodo intermedio 2, el algoritmo de ruteo intercambia la dirección destino, que antes tenia el paquete IPv6 con la primera direccion de la lista de direcciones de la cabecera de ruteo, que en este caso es la direccion del nodo intermedio 2, también se debe de notar que la dirección origen siempre es la misma en todos los segmentos.

Para que el paquete IPv6 viaje del nodo intermedio 2 al nodo intermedio 3, el algoritmo de ruteo intercambia la dirección destino que antes tenia el paquete IPv6, con la segunda direccion de la lista de direcciones de la cabecera IPv6.

Para que el paquete viaje del nodo intermedio 3 al destino final, el algoritmo de ruteo intercambia la dirección destino que antes tenia el paquete IPv6, por la tercera dirección de la lista de direcciones, que es la dirección del destino final. Note que ahora la cabecera Base IPv6 tiene como dirección origen la dirección del nodo que origino el paquete y como dirección destino la dirección del destino final, y la cabecera de ruteo tiene la lista de direcciones de los nodos intermedios en orden por donde el paquete fue pasando.

4.2.4 Cabecera de Fragmentación

La cabecera de fragmentación es usada por hosts IPv6 origen, cuando es necesario enviar paquetes de tamaño mayor al MTU(Unidad Máxima de Transferencia) permitido en las distintas tecnologías de redes (Token Ring, Ethernet, FDDI), durante el transcurso a su destino final. La fragmentación en IPv6 es distinta a IPv4, ya que en IPv6 la fragmentación es realizada solo por los nodos origen y no por los ruteadores a lo largo del camino a su destino, como se realiza en IPv4, con esto se tiene una mayor eficiencia en el ruteo La cabecera de fragmentación es identificada por el valor 44 del campo Next Header de la cabecera anterior a la de fragmentación.

La cabecera de Fragmentacion se muestra en la figura 4.15

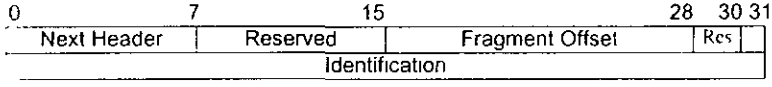


Fig. 4 15 - *Formato de la Cabecera de Fragmentación*

En este formato se puede observar:

El campo Next Header tiene una longitud de 8 bits e identifica la cabecera que le sigue a la cabecera de fragmentación, que será la cabecera del paquete inicial de la *fragmentacion del paquete original.*

El campo de reserva (Reserved) es un campo reservado para usos futuros y tiene una longitud de 8 bits, este campo es inicializado en cero e ignorado en la recepción.

El campo Fragment Offset tiene una longitud de 13 bits, este campo tiene la función de medir el desplazamiento en unidades de octetos de los datos siguientes a este encabezado.

El campo de reserva (Reserved) tiene una longitud de 2 bits y es reservado para usos futuros. este campo se inicializa en cero en la transmisión y es ignorado por el receptor.

El campo M tiene una longitud de un bit y determina si más fragmentos están llegando o si es el ultimo fragmento. Cuando es el ultimo fragmento, el campo M tiene el valor de cero y cuando no es el ultimo tiene el valor de uno.

El campo Identification tiene una longitud de 32 bits y sirve para identificar los fragmentos del paquete. durante el proceso de reemsamble. Para cada paquete que es fragmentado el nodo origen genera un valor de identificación para este campo, que debe ser diferente a otro paquete que ha sido fragmentado recientemente.

El paquete original, al ser fragmentado consta de dos partes como se ilustra en la figura 4 16

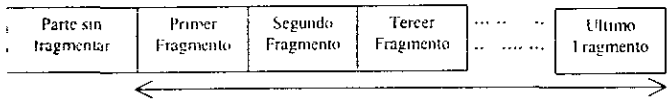


Fig 4 16 - *Partes del Paquete Fragmentado*

La parte sin fragmentar consiste de la cabecera base IPv6 y cabeceras suplementarias (en el caso de que se presenten) que deben ser procesadas por los nodos durante el trayecto del paquete hasta su destino, estas pueden ser la cabecera de ruteo y la cabecera Hop by Hop.

La parte fragmentada consiste del resto del paquete, la cual incluye cualquier otra de las cabeceras suplementarias que se presentan en el paquete, ya que estas solo necesitan ser procesadas por el nodo o nodos (en el caso de dirección Multicast o Anycast) de destino final, también contiene las cabeceras de capas superiores y los datos.

La parte fragmentable del paquete es dividida en fragmentos que tienen una longitud de un numero múltiplo a 8 octetos, excepto para el ultimo fragmento. Cada fragmento del paquete consiste de 3 partes: La parte sin fragmentar del paquete original, la cabecera de fragmentación y los datos fragmentados.

- La parte sin fragmentar contiene la misma información que se menciono anteriormente, solo que cambia el valor del campo Payload Length que ahora tendrá la longitud del fragmento, asi como el valor del campo Next Header que tendrá el valor 44 que es el de la cabecera de fragmentación
- La cabecera de fragmentación contiene el valor 44 en el campo Next Header, señalando con esto que hay mas fragmentos, excepto para el ultimo fragmento que

contendrá el valor de la cabecera correspondiente a la cabecera que este después del ultimo fragmento.

- Los datos fragmentados corresponden a la información de cabeceras de capas superiores, cabeceras suplementarias que solo son procesadas por el nodo de destino final y los datos.

En la figura 4 17 se muestra como es que se compone un fragmento de un paquete.

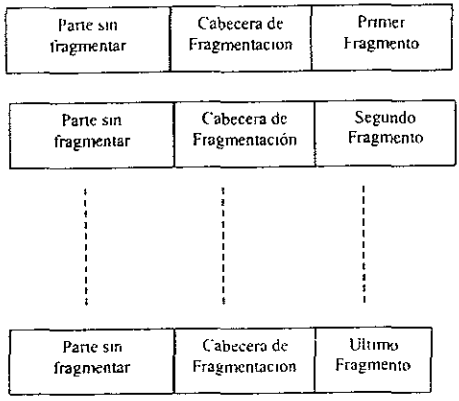


Fig.4.17 - *Partes del Fragmento*

El paquete original es reemsambleado solo con los paquetes que tiene la misma Dirección origen, Dirección Destino y mismo identificador.

4.2.5 Cabecera de Autenticación.

La cabecera de autenticación provee de una conexión integra y autenticación de los datos de origen para datagramas IP y provee proteccion contra reenvios de paquetes que podrian ser interceptados por un nodo.

La presencia de la cabecera de autenticación es identificada por el valor 51 del campo Next Header de la cabecera anterior. Esta cabecera tiene seis campos como se muestra en la figura 4 18.

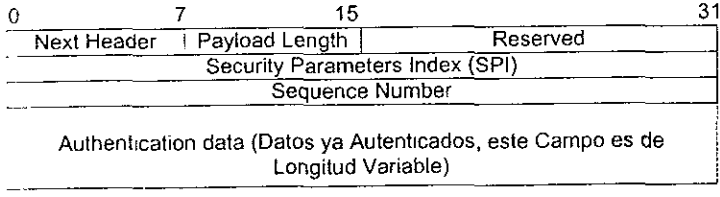


Fig 4.18 - *Formato de la Cabecera de Autenticación.*

El campo Next Header tiene una longitud de 8 bits, e identifica la cabecera inmediatamente siguiente a la cabecera de Autenticación.

El campo Payload Length tiene una longitud de 8 bits y el objetivo de este campo es medir la longitud de la cabecera de Autenticación en unidades de 32 bits (4 bytes), no tomando en cuenta lo dos primeros 64 bits (8 bytes) de la cabecera El minimo valor de este campo es 1, siendo asi su longitud total de 96 bits, pero como se dijo que no se tomaban en cuenta los primeros 64 bits, por lo que son 3 unidades de 32 bits menos 2 unidades de 32 bits queda una unidad de 32 bits (3 -2=1, 96-64= 32 bits). Este valor mínimo es solo usado en el caso de un algoritmo de autenticación nulo, empleando esta cabecera para suprimir errores.

EL campo de reserva (Reserved) tiene una longitud de 16 bits, y es reservado para usos futuros Este campo es inicializado en cero en la transmisión, pero es ignorado en la recepcion

El campo Security Parameters Index (SPI), tiene una longitud de 32 bits. El SPI es un valor arbitrario de 32 bits, que junto con la dirección IP destino y el protocolo de seguridad usado en la cabecera de Autenticación, identifica la asociación de seguridad para este datagrama El conjunto de valores SPI en el rango 1 hasta 255 son reservados por la organizacion Internet Assiganed Numbers Autority (IANA) para un uso futuro El valor de 0 es reservado para uso local, uso de implementaciones especificas y no debe ser enviado en la red.

El campo Sequence Number tiene una longitud de 32 bits, conteniendo un numero de 32 bits que gradualmente va incrementando el valor del contador del numero de secuencia. Este es obligatorio y esta siempre presente en esta cabecera. El procesamiento de este campo es a juicio del receptor, por ejemplo el nodo que envía el paquete debe transmitir siempre este campo, pero el receptor no necesita hacer alguna acción sobre este campo.

El contador del remitente y del receptor son inicializados hasta cero cuando la asociacion de seguridad es establecida.

El campo Authentication Data tiene una longitud variable, contiene el valor de verificacion de integridad (ICV, *Integrity Check Value*). La longitud de este campo debe ser un múltiplo de 32 bits, por lo que el campo puede incluir bits de relleno.

El calculo del valor de verificacion de integridad (ICV) se aplica sobre los siguientes campos

- Los campos de la cabecera IPv6 que son inalterables durante el camino a su destino, o que pueden ser predecibles, como lo es el campo de Destination Address cuando viene con la cabecera suplementaria de Ruteo Un ejemplo de campos alterables durante el transcurso del camino del paquete son los campos Hop Limit, Traficc Class y Flow Label.
- La cabecera de Autenticación
- Datos de protocolos de capas superiores

4.2.6 Cabecera Encapsulating Security Payload (ESP)

La cabecera de ESP es diseñada para proveer confidencialidad, autenticación de datos de origen, integridad orientada a conexión, servicio detección de datagramas duplicados, confidencialidad en el flujo del trafico. Esta cabecera puede emplearse solo o junto con la cabecera de autenticación. El conjunto de servicios provistos depende en las opciones seleccionadas cuando se establece la asociación de seguridad

La opcion de confidencialidad en el flujo de trafico requiere la selección de forma de modo tunel para poder aplicarse.(Esto se estudia mas a fondo en el capitulo 5)

La cabecera ESP tiene el formato que se muestra en la figura 4.19.

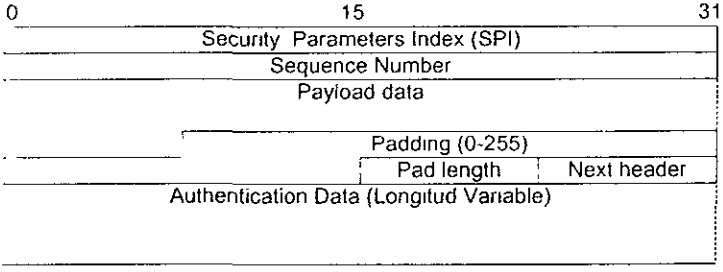


Fig. 4 19 - *Formato de la Cabecera de Encapsulation Security Payload*

El campo Security Parameters Index (SPI), tiene una longitud de 32 bits. El SPI es un valor arbitrario de 32 bits, que junto con la dirección IP destino y el protocolo de seguridad usado en la cabecera de Autenticación, identifica la asociación de seguridad para este datagrama. El conjunto de valores SPI en el rango 1 hasta 255 son reservados por la organización Internet Assiganed Numbers Autority (IANA) para un uso futuro. El valor de 0 es reservado para uso local, uso de implementaciones especificas y no debe ser enviado en la red.

El campo Sequence Number tiene una longitud de 32 bits, conteniendo un numero de 32 bits que gradualmente va incrementando el valor del contador del numero de secuencia Este es obligatorio y esta siempre presente en esta cabecera. El procesamiento de este campo es a juicio del receptor, por ejemplo el nodo que envía el paquete debe transmitir siempre este campo, pero el receptor no necesita hacer alguna acción sobre este campo.
El contador del remitente y del receptor son inicializados hasta cero cuando la asociacion de seguridad es establecida.

El campo Payload tiene una longitud variable, contiene los datos descritos por el campo Next Header. Este campo es obligatorio.

El campo Padding es de longitud variable, puede contener opcionalmente de 0 a 255 octetos de información de relleno como según se requiera de la opción de seguridad.

El campo Pad Length tiene una longitud de 8 bits e indica el numero de octetos de relleno (0 a 255) que ocupa el campo anterior. Este campo es obligatorio.

El campo Next header tiene una longitud de 8 bits e identifica la cabecera siguiente a esta cabecera

El campo Authentication Data tiene una longitud variable, contiene un valor de verificación de integridad (ICV, *Integrity Check Value*), que es calculado sobre el paquete ESP, no tomando en cuenta el campo Authentication Data. La longitud de este campo debe ser un múltiplo de 32 bits, por lo que el campo puede incluir bits de relleno. Este campo es opcional, solo es incluido si el servicio de autenticación ha sido seleccionado por la asociación de seguridad

4.3. TAMAÑO DE LOS PAQUETES EN IPV6

En el diseño de IPv6 se tomo como tamaño máximo del paquete IPv6, el mismo tamaño que en IPv4, que es de 65575 octetos. Tomando en cuenta los 40 octetos del *tamaño de la cabecera base IPv6, la información útil es de 65535 En el campo Payload* Length de la cabecera IPv6 es donde identifica el tamaño del paquete.

IPv6 requiere que cada tecnología de enlace en Internet tenga un MTU de 1280 octetos o mayor, en cualquier enlacé o medio físico que no tenga estas características, la fragmentacion y el reemsamble del paquete debe hacerse en capas inferiores a la perteneciente a IPv6 (Capa de Internet).

Para cuando un paquete necesita transportar una cantidad mayor de 65535 octetos de información útil, existe en IPv6 lo que se conoce como los paquetes Jumbogram que soporta paquetes de tamaño mayor a 4GB, este tipo de paquetes se diseño para uso de Supercomputadoras para hacer una transferencia mas eficiente con pocas interrupciones, estos paquetes solo pueden ser utilizados dentro de una misma red local

Los paquetes Jumbograms son soportados por la opción Jumbo Payload dentro de la cabecera Hop by Hop, en este caso el campo Payload Length en la cabecera IPv6 es puesta en cero, y en el calculo del tamaño de la longitud de paquete no se toma en cuenta la cabecera IPv6 pero si la cabecera Hop by Hop, y esta longitud del paquete va contenida en el campo Jumbo Payload Length dentro de la opción Jumbo Payload de la cabecera Hop by Hop.

4.4 PROTOCOLO DE DESCUBRIMIENTO DEL NODO VECINO (NEIGHBOR DISCOVERY FOR IPV6)

El *protocolo de descubrimiento del nodo vecino* (Neighbor Discovery for IPv6), corresponde a una combinación de protocolos IPv4, ARP, mensajes de descubrimiento de

Ruteadores ICMP (RDISC, *ICMP Router Discovery*) y Redireccionamiento ICMPv4 (*ICMPv4 Redirect*).

Este Protocolo solventa los problemas relacionados entre la interacción de los nodos pertenecientes a una misma red. Este protocolo provee mecanismo para resolver los siguientes problemas.

- Descubrimiento de ruteadores: Como los host localizan a los ruteadores que residen en una misma red
- Descubrimiento del Prefijo Como los host descubren el conjunto de prefijos de direcciones, que definen cuales direcciones de posibles host destinos se encuentran conectados a la red
- Descubrimientos de parámetros: Como los Host aprenden los parámetros de la red, como lo es el MTU y él número de saltos para llegar a su destino (Hop Limit).
- Autoconfiguración de direcciones. Como los nodos automáticamente configuran una dirección para una interface.
- Resolución de direcciones: Como los nodos determinan la dirección física de la interface del host destino en base a su dirección IP
- Determinación del Siguiente Salto: Como el nodo determina el siguiente nodo por donde se enviara los paquetes para que lleguen a su destino. Es siguiente salto del paquete puede ser un ruteador o el mismo nodo destino.
- Detección de un Nodo Vecino Inalcanzable: Como el nodo determina que el nodo vecino es inalcanzable ya sea por problemas en el nodo, en el medio físico u otros problemas. Para el caso donde un ruteador vecino es inalcanzable, se puede probar con otros ruteadores por default que se encuentren en la misma red
- Detección de una Dirección Duplicada: Como un nodo determina que una dirección IP que este nodo quiere usar no esta siendo ocupada por otro nodo.
- Redireccionamiento. Como un ruteador informa a un host de una mejor ruta para el primer salto, para llegar a su destino final.

El protocolo de descubrimiento del nodo vecino (Neighbor Discovery for IPv6) proporciona varias mejoras sobre el conjunto de protocolos IPv4, como:

- El descubrimiento del ruteador es parte de la base del conjunto de protocolos de IPv6, así los host no necesitan involucrar a los protocolos de ruteo.
- Los mensajes de Notificación del ruteador transmiten las direcciones físicas, y no en lugar de usar el paquete para conocer la dirección física de la red. También este mensaje habilita la autoconfiguración de direcciones.
- Los ruteadores pueden divulgar el MTU a los host

El Protocolo de descubrimiento del Nodo Vecino define 5 diferentes tipos de mensajes ICMPv6. Solicitud de Ruteador (Router Solicitation), Notificación del ruteador (Router Advertisement), Solicitud de Host Vecino (Neighbor Solicitation), Notificación del nodo vecino (Neighbor Advetisement), el mensaje de direccionanmiento

- Solicitud al Ruteador (Router Solicitation): Cuando una interface es habilitada, el host puede emitir mensajes de Solicitación al Ruteador, solicitando así ruteadores que generen un mensaje de Notificación del Ruteador, para saber que ruteadores de la misma red esta disponibles.

- Notificación de ruteador (Router Advertisement): Este mensaje lo mandan los ruteadores para advertir su presencia en la red, así como para dar a conocer algunos parámetros de la red (MTU, Prefijos, límite de saltos, etc.) o también en respuesta a un Mensaje de Solicitud de Ruteador.
- Solicitud al nodo Vecino (Neighbor Solicitation) Este mensaje sé envía por un nodo para determinar la dirección física del host vecino, o para verificar que el host vecino aun esta disponible con la dirección física guardada en su tabla. También estos mensajes son utilizados para la detección de direcciones duplicadas
- Notificación de nodo vecino (Neighbor Advertisement). Este mensaje lo envía un nodo en repuesta al mensaje de Solicitud de Host Vecino. Un nodo puede también enviar este tipo de mensajes sin previa solicitud, para informar su cambio de dirección física de la interface.
- Redireccionamiento: Este tipo de mensajes es usado para informar a los host de un mejor primer salto para que el paquete llegue a su destino

4.4.1 Modelo de la Base de Datos de un Host

Un Host incluye varias estructuras de datos que son mantenidas para facilitar la interacción con los nodos vecinos, la información que radica en el host incluye lo siguiente

- Información de nodos vecinos: Un conjunto de información de nodos vecinos , a los *cuales se ha tenido comunicación recientemente* Esto incluye, las direcciones físicas de los nodos vecinos, tipo de nodo vecino (Host o Ruteador), indicadores para los paquetes que están haciendo cola de espera para resolución de dirección.
- Información de destino: Es la información de los destinos donde el trafico ha sido enviado. Esta información puede incluir destinos dentro del mismo segmento de red o *fuera del segmento. La información es actualizada con los mensajes* de redireccionamiento. También información como el MTU y tiempo de viaje pueden ser almacenados aquí.
- Lista de prefijos: Una lista de prefijos que definen un conjunto de direcciones que se encuentra en el mismo segmento de red. Esta información es creada a partir de la *información recibida por los mensajes de notificación de ruteador.*
- Lista de Ruteadores por default: Una lista de ruteadores por default, a los cuales se les puede enviar los paquetes.

Cuando un paquete esta para ser enviado, el nodo usa la información de destino, de la lista de prefijos y de la lista de ruteadores por default, para determinar la dirección IP del salto siguiente del paquete, ya que la dirección IP del paquete es conocida, es consultada la información de nodos vecinos para obtener información de este nodo (dirección física, si es ruteador o host, etc.).

Además de la información antes citada el host mantiene un numero de variables como lo son

- Link MTU. El MTU de la red.
- CurHoplimit: El límite de saltos por default que se usa para cuando paquetes unicast IPv6 son enviados.
- BaseReachableTime. Un valor base usado para calcular aleatoriamente el valor de tiempo de alcanzabilidad del destino
- ReachableTime El tiempo en que un nodo vecino es alcanzable por el paquete despues de recibir una confirmación de alcanzabilidad.
- RetransTimer El tiempo entre la retransmisión de mensajes de solicitud al nodo vecino cuando se está probando la alcanzabilidad del vecino

4.4.2 Mensaje de Solicitud de Ruteador (Router Solicitation)

El formato del mensaje de Solicitud de Ruteador se puede observar en la figura 4.20.

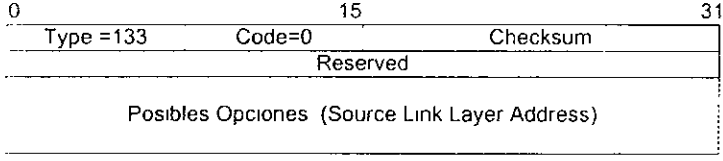


Fig. 4.20 - *Mensaje ICMPv6 de Solicitud al Ruteador*

El mensaje de solicitud de ruteador es transmitido por un host, para que los ruteadores generen rápidamente el mensaje de notificación del ruteador.

En la cabecera IPv6 cuando se envía este tipo de mensaje, el campo de Source Address identifica la dirección de la interface por la cual se envio el paquete y el campo Destination Address identifica la dirección multicast de todos los ruteadores.

La cabecera de Autenticación es incluida si la asociación de seguridad existe entre el nodo origen y la dirección destino.

En el mensaje ICMPv6; el campo type identifica el tipo de mensaje con el valor 133.

El campo Code tiene el valor 0, el campo Reserved puede ser inicializado en cero y ignorado por el receptor.

Este mensaje tiene como posible opcion: Dirección Física Origen (Source Link Layer Address)

4.4.3 Mensaje de Notificación de ruteador (Router Advertisement).

Estos mensajes son transmitidos por ruteadores periódicamente o en respuesta a un mensaje Solicitud de Ruteador hecha por un host

El formato del mensaje puede observarse en la figura 4 21

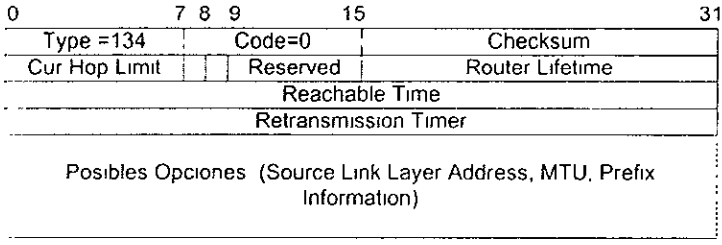


Fig. 4.21 - *Mensaje ICMPv6 de Notificación de Ruteador.*

En la cabecera IPv6, cuando se envía este tipo de mensaje, el campo de Source Address identifica la dirección de la interface por la cual este paquete fue enviado, el campo Destination Address identifica la dirección origen del mensaje que invoco el mensaje de solicitud de ruteador, o la dirección multicast de todos los rutedores.

El campo type tiene el valor 134 que identifica este mensaje

El campo code tiene el valor cero. El campo Current Hop Limit tiene el valor default que deberia ser puesto en el campo Hop Limit de la cabecera IPv6 de los paquetes IPv6, el valor cero significa que no es especifica por el ruteador. Dos banderas de un solo bit cada una identifican el manejo de la configuración de direcciones (bandera M) y otras configuraciones (bandera O).

Cuando la bandera M es igual a uno, el host usa protocolos como Protocolo de Configuración de Host Dinámicos versión 6 (DHCPv6, *Dynamic Host Configuration Protocol*) para la configuración de direcciones, y cuando la bandera O es igual a uno, el host usa protocolos para la autoconfiguración de otra información diferente a direcciones

El campo Reserved se plantea para usos futuros y se inicializa en cero y es ignorado por el receptor.

El campo Router Lifetime especifica el tiempo de vida asociado con el ruteador por default en unidades de segundos, el máximo valor corresponde a 18.2 horas, el valor cero especifica que no es un ruteador y que pueda ser tomado como un ruteador por default.

El campo Reachable Time especifica el tiempo en milisegundos, que asume un nodo hacia el alcance de un host vecino, después de haber recibido la confirmación de alcanzabilidad

El campo Retransmision Timer especifica el tiempo de retransmisión en milisegundos, entre el mensaje retransmitido de solicitud al nodo vecino.

El campo de posibles opciones tiene: Dirección Física Origen (Source Link Layer Address), MTU e Información de Prefijo (Prefix Information).

4.4.4 Mensaje de Solicitud al Nodo Vecino (Neighbor Solicitation)

El mensaje de Solicitud al Nodo Vecino es enviado por nodos que requieren la dirección física de la interface del nodo, mientras también provee su propia dirección física de su interface. Los mensajes de solicitudes al nodo son multicast cuando el nodo necesita conocer por primera la dirección, y es unicast cuando el nodo busca verificar la alcanzabilidad del nodo vecino

El formato de este tipo de mensaje se puede observar en la figura 4.22.

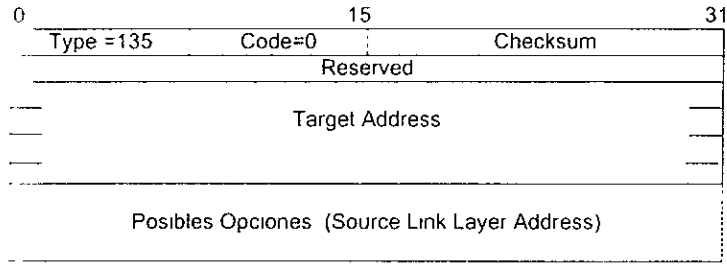


Fig. 4 22 - *Mensaje ICMPv6 de Solicitud al Nodo Vecino.*

Con respecto a la cabecera IPv6, el campo de Source Address contiene la dirección física de la interface del nodo que envía este mensaje, y el campo de Destination Address es la dirección multicast correspondiente al nodo para resolución de direcciones.

La cabecera de Autenticación es incluida si la asociación de seguridad existe entre el transmisor y la dirección destino.

En el mensaje ICMPv6, el campo Type tiene un valor a 135 y el campo code tiene el valor de cero

El campo Reserved se plantea para usos futuros y se inicializa en cero y es ignorado por el receptor.

El campo Target Address es la dirección IPv6 del nodo especifico a quien va destinado el mensaje para realizar la resolución de direcciones Esta dirección no debe ser un direccion Multicast

En el campo de posibles opciones se tiene la opción de dirección física origen (Source Link Layer Address).

4.4.5 Mensaje de Notificación de Nodo Vecino (Neighbor Advertisement)

Este tipo de mensajes son enviados por los nodos, en respuesta de un mensaje de solicitud al nodo vecino También son enviados sin una previa solicitud para propagar nueva información, por ejemplo cuando la dirección física de su interface ha cambiado, el nodo envía un mensaje de notificación de nodo vecino a todos los nodos para que los nodos actualicen sus tablas y conozcan su nueva dirección física.

En la figura 4.23 se muestra el formato de este tipo de mensaje.

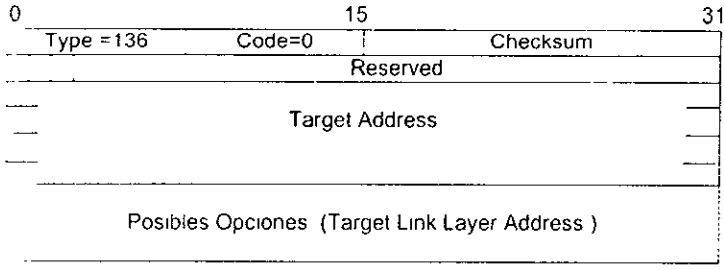


Fig. 4.23 - *Mensaje ICMPv6 de Notificación de Nodo Vecino.*

Con respecto a la cabecera IPv6, el campo de Source Address contiene la dirección física asignada a la interface del nodo que envía este mensaje, el campo de Destination Address es la dirección del nodo que invoco la solicitud al nodo vecino.

La cabecera de Autenticación es incluida si la asociación de seguridad existe entre el transmisor y la dirección destino.

En el mensaje ICMPv6, el campo Type contiene el valor 136 y el campo code contiene el valor cero, estos campos son los que identifican este mensaje

El campo con la bandera R cuando tiene el valor de uno indica que el mensaje de notificación fue enviado por un ruteador

El campo con la bandera S cuando tiene el valor de uno indica que el mensaje de Notificacion de Nodo Vecino fue enviado en respuesta a un mensaje Solicitud al Nodo Vecino desde la dirección destino que fue solicitada.

El campo con la bandera O indica que el nodo que envío la Solicitud al Nodo Vecino, puede hacer caso omiso de la dirección física que tiene en su tabla de direcciones y actualizarla con la dirección física que viene en el paquete de notificación Si el bit del campo O no se pone en uno, el nodo no podrá actualizar su tabla con la dirección física que se envio, sino que la actualización de la tabla se hará con la información de que no se conoce la dirección física del nodo a la que se le hizo la solicitud

El campo Reserved se plantea para usos futuros y se inicializa en cero y es ignorado por el receptor

El campo Target Address es la dirección especificada en el campo target address del mensaje Solicitud al Nodo Vecino, o la dirección IPv6 del nodo que sufrio el cambio de la direccion física (Esto es para mensajes que no fueron solicitados previamente). Esta direccion no debe ser un dirección Multicast.

En el campo de posibles opciones se tiene la opción deTarget Link Layer Address, identificando la dirección física del nodo al que se realizo la reasignacion de su direccion IPv6, o direccion fisica en el caso de cambio de trageta de red.

4.4.6 Mensaje de Redireccionamiento (Redirect Message)

El mensaje de direccionamiento es enviado por los ruteadores para informar a los hosts de un mejor nodo como primer salto en su camino hacia el destino final

El formato del mensaje de direccionamiento se muestra en la figura 4.24.

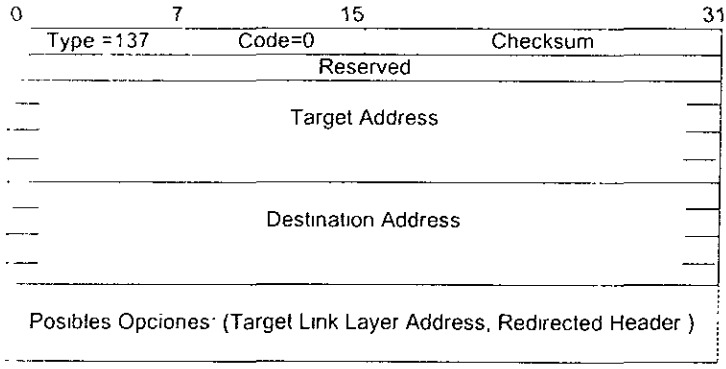


Fig 4 24 - *Mensaje ICMPv6 de Redireccionamiento*

En la cabecera IPv6, el campo Source address contiene la dirección física de la interface del nodo que envía este mensaje, y el campo Destination Address es la dirección origen del mensaje que provoco el direccionamiento

La cabecera de Autenticación es incluida si la asociación de seguridad existe entre el transmisor y la dirección destino.

El campo Type del mensaje contiene el valor 137 que identifica este mensaje, el campo Code tiene el valor de cero.

El campo Reserved se plantea para usos futuros y se inicializa en cero y es ignorado por el receptor

El campo Target Address es la dirección IPv6 del nodo que es mejor como primer salto del paquete, asi como también usarla como dirección destino del mensaje ICMP.

El campo Destination Address es la dirección IPv6 del nodo que estaba como primer salto hacia el camino del paquete

En el campo de posibles opciones se tiene: la opción de Target Link Layer Address, identificando la dirección física del nodo al que se realizo la resolución de direcciones y la cabecera de Redireccionamiento

4.4.7 Opciones de los Mensajes de descubrimiento de Nodo vecino(Neighbor Discovery Message Options)

Los mensajes, como se vio anteriormente pueden incluir o no opciones, algunas de las cuales pueden aparecer varias veces en el mismo mensaje Se han definido 5 opciones que son

- Dirección física Origen (Source Link Layer Address)
- Dirección física del nodo al que se realizo la reasignacion de direcciones (Target Link Layer Address)
- Información Prefijo (Prefix Information)
- Cabecera de Redireccionamiento (Redirect Header)
- MTU

Dirección física Origen (Source Link Layer Address)

La opción de Dirección física origen, contiene la dirección física de la interface del nodo que emite el paquete. Esta es usada en los mensajes de: solicitud al nodo vecino, Solicitud al Notificación de Nodo Vecino, Solicitud al Ruteador y Notificación de Ruteador. Esta opción debe ser ignorada por los otros mensajes del protocolo de descubrimiento de nodo vecino En la figura 4.25 se muestra el formato de esta opción

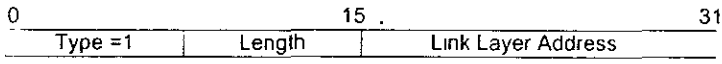


Fig. 4.25 - *Opción de Dirección Física Origen*

Esta opción es identificada por el valor 1 en el campo Type. El campo Length mide la longitud de opción en unidades de 8 octetos (64 bits).

Dirección física del nodo al cual se le realizo la reasignacion de dirección (Target Link Layer Address)

Esta opción contiene la dirección física del nodo a quien se le efectúa la reasignacion de direcciones IPv6. Esta es usada en los mensajes. Notificación de Nodo Vecino y Redireccionamiento. Esta opción debe ser ignorada por los otros mensajes del protocolo de descubrimiento de nodo vecino. En la figura 4.26 se muestra el formato de esta opción.

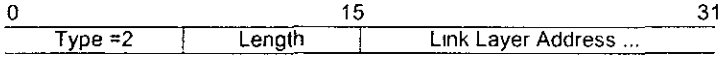


Fig. 4.26 - *Opción de Dirección Física del nodo al que se le realizó la resolución de direcciones.*

Esta opción es identificada por el valor 2 en el campo Type. El campo Length mide la longitud de opción en unidades de 8 octetos (64 bits).

Información Prefijo (Prefix Information)

La opción Prefijo provee de un prefijo a las direcciones de un segmento de red y de otro para la autoconfiguración de direcciones. Esta opción aparece en los mensajes de Notificación de Ruteador Esta opción debe ser ignorada por los otros mensajes del protocolo de descubrimiento de nodo vecino En la figura 4.27se muestra el formato de esta opcion

La información del prefijo provee a los host con un prefijo adecuado de las direcciones pertenecientes a un segmento de red y de un prefijo para la autoconfiguración de direcciones

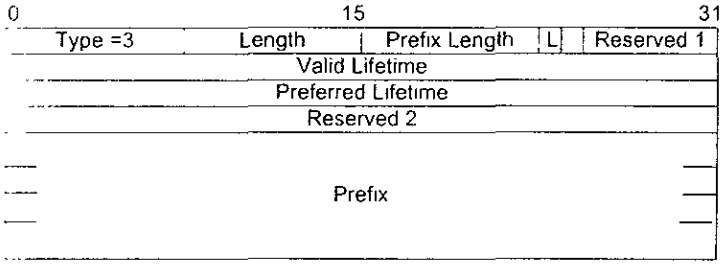


Fig. 4.27 - *Información del Prefijo*

Esta opción es identificada por el valor 3 en el campo Type y el campo Length tiene un valor de 4(en unidades de 64 bits)

El campo Prefix Length tiene una longitud de 8 bits, se encarga de indicar el numero de bits principales que son validos en el prefijo, este puede tomar un rango de 0 hasta 128.

El campo con la bandera L de un solo bit de longitud, cuando es puesta en uno indica que el prefijo puede ser usado para determinar las direcciones de los nodos que se encuentra en un segmento. Cuando no se pone en uno esto indica que se puede usara para determinar la dirección de nodo que puede pertenecer o no al mismo segmento de red

El campo con la bandera A (autónoma autoconfiguración de direcciones) indica que el prefijo puede ser usado para una autónoma autoconfiguración de direcciones.

Los campos Reserved 1 y 2 son reservados para un uso futuro, estos se inicializan en cero y son ignorados por el receptor.

El campo Valid Lifetime, tiene una longitud de 32 bits y tiene el tiempo en segundos que el prefijo es valido para el propósito de la determinación de la dirección.

El campo Preferred Lifetime, tiene una longitud de 32 bits y tiempo en segundos que las direcciones generadas desde el prefijo mediante la autoconfiguración de direcciones pueden permanecer en primera estancia.

El campo Prefix tiene una longitud de 128 bits y contiene una dirección IPv6 o un prefijo para una dirección IPv6.

Cabecera de Redireccionamiento (Redirect Header).

La cabecera de redireccionamiento es usada en un mensaje de redireccionamiento; esta contiene toda o parte del paquete que esta siendo redireccionado. Esta opción debe ser ignorada por los otros mensajes del protocolo de descubrimiento de nodo vecino. En la figura 4 28 se muestra el formato de esta opción.

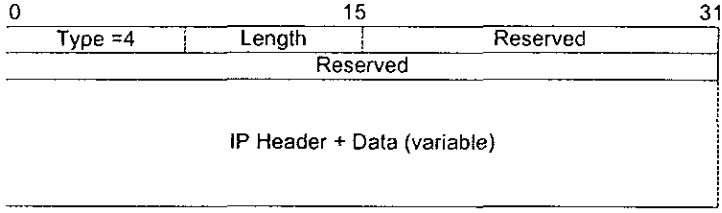


Fig. 4.28 - *Cabecera de Redireccionamiento*

Esta opción es identificada por el valor 4 en el campo Type. El campo Length mide la longitud de esta opción en unidades de 64 bit.

Los campos reserved, son inicializados en cero e ignorado por el receptor.

El Campo IP Header + Data contiene el paquete original truncado, para poder asegurar que el tamaño del mensaje de redireccionamiento no exceda los 576 octetos.

Opción MTU.

La opción MTU es usada en el mensaje de notificación de ruteador para asegurar que todos los nodos en la red usen el mismo valor de MTU. Esta opción debe ser ignorada por los otros mensajes del protocolo de descubrimiento de nodo vecino. En la figura 4.29 se muestra el formato de esta opción.

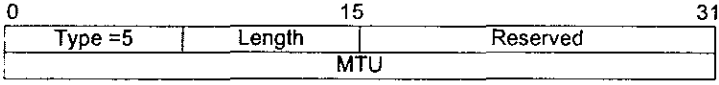


Fig. 4.29 - *Opción MTU*

Esta opción es identificada por el valor 5 en el campo Type. EL campo Length se encarga de la longitud de la opción la cual es 1 (unidades de 64 bits).

El campo reserved es inicializado en cero e ignorado por el receptor.

El campo MTU tiene una longitud de 32 bits y especifica el MTU recomendado para la red en que se esta comunicando.

4.4.8 Proceso de descubrimiento del MTU de la ruta

El proceso del descubrimiento del MTU se muestra en el figura 4.30, este proceso inicia desde el nodo origen asumiendo el MTU de la red donde se encuentra el nodo que inicia el camino del paquete a su destino (El primer salto) Por ejemplo si el nodo origen conoce que el primer salto esta dentro de una red FDDI, con un MTU de 4,352 octetos, entonces asume que el camino completo tiene un MTU de 4,352, entonces el nodo transmite un paquete para verificar el tamaño del MTU, si no se recibe un mensaje ICMPv6 de paquete demasiado grande, entonces el nodo origen asume que el valor del MTU es el mismo en todo el camino y el nodo sigue transmitiendo, pero si se recibe un mensaje de paquete demasiado grande, entonces el MTU de la ruta debe ser reducido por el valor que viene en el mensaje ICMPv6, y nuevamente se transmite el paquete de prueba con el valor del MTU que viene en el mensaje ICMPv6 y se hace la misma prueba anterior Este proceso continua hasta que ningún mensaje ICMPv6 de paquete demasiado grande es regresado Ya que se termino este proceso, el MTU ya es conocido por el nodo origen y se puede proceder a la transmisión. En un periodo base se hace una nueva verificacion del valor del MTU de la ruta, para checar si el valor del MTU puede ser incrementado.

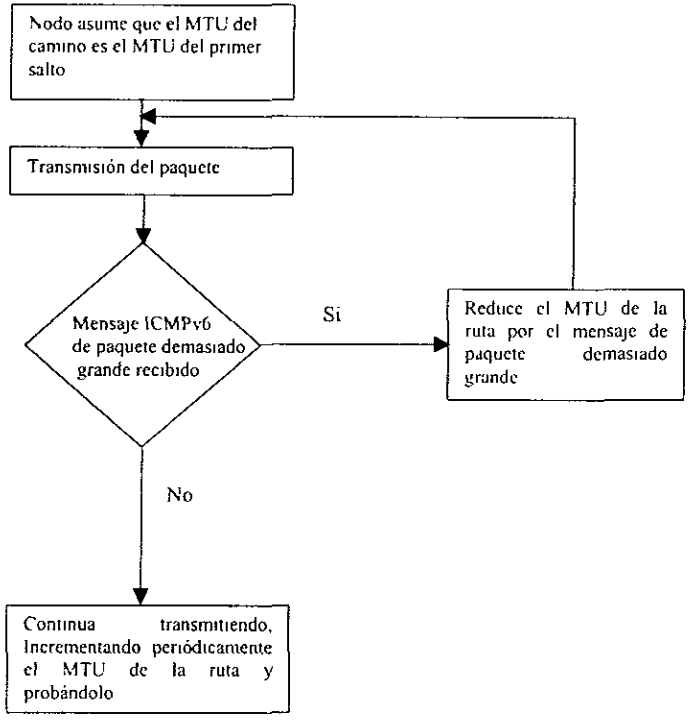


Fig. 4 30 - *Descubrimiento de MTU.*

4.4.9 Procesos de descubrimiento del prefijo y del ruteador

El descubrimiento del ruteador es un proceso para localizar ruteadores pertenecientes a mismo segmento, y también para aprender prefijos y parámetros de configuración para así guardarlos en su base de datos. El proceso de descubrimiento de prefijo es mediante el cual el host aprende los rangos de las direcciones IP que residen en el mismo segmento y son por eso directamente alcanzables, sin pasar por el ruteador. Cuando ruteadores transmiten mensajes de notificación de ruteador, los cuales contienen la información de los prefijos y del ruteador, los procesos de descubrimiento de ruteador y del prefijo son satisfechos

4.4.10 Proceso de resolución de direcciones y detección de la inalcanzabilidad del nodo vecino

Los mensajes de solicitud al nodo vecino y notificación de nodo son usados para transmitir información con respecto a las direcciones y alcanzabilidad de los nodos vecinos Estos mensajes también son usados para el proceso de detección de direcciones duplicadas, el cual es parte del algoritmo de autoconfiguración

La resolución de direcciones es el proceso a través del cual un nodo determina la dirección física del nodo vecino, cuando solo conoce su dirección IPv6. El proceso de resolución de direcciones es solo para ser ejecutado en las direcciones que pertenecen al mismo segmento.

La Detección de Inalcanzabilidad del nodo vecino es el proceso en el que una ruta de comunicación es identificada como fallida para que sea corregida. El especifico procedimiento de recuperación variara dependiendo del tipo de nodo donde surgió el problema

El estado de alcanzabilidad de un nodo vecino puede tomar uno de 5 posibles valores, que son

- Incompleto: El proceso de resolución de direcciones esta en progreso, y la dirección física del nodo vecino no ha sido determinada.
- Alcanzable. Se conoce que el nodo vecino ha sido recientemente contactado.
- Stale: No se conoce lo suficiente la alcansabilidad del nodo vecino (no se ha enviado reciente mente paquetes hacia el nodo)como para determinar el tiempo de la variable de alcanzabilidad (ReachableTime)del nodo vecino, pero hasta que no envíe un paquete que llegue al host vecino, no se podrá determinar el valor de la variable ReachebleTime
- Delay. No se conoce lo suficiente la alcansabilidad del nodo vecino como para determinar el tiempo de la variable de alcanzabilidad (ReachableTime)del nodo vecino, aunque se ha enviado trafico hasta aquel nodo recientemente.
- Sondeo: No se conoce lo suficiente, la alcansabilidad del nodo vecino , se envían pruebas de sondeo en periodos de tiempo, para verificar su alcanzabilidad.

Cuando un paquete está por ser enviado, el nodo emisor consulta la información de los nodos vecinos, con respecto al nodo vecino con que se quiere comunicar. Si no existe información con respecto al nodo requerido, se hace uso del proceso de resolución de direcciones, y este proceso crea una entrada de información de estado incompleto y envía un mensaje de solicitud de nodo vecino. Mientras espera la respuesta el nodo emisor retransmite estos mensajes de solicitud al nodo vecino en periodos de tiempo iguales a la variable de Retrans Timer, cuando llega la respuesta la información del mensaje se guarda en la tabla de información de nodos vecinos

4 5 AUTOCONFIGURACION DE DIRECCIONES

El proceso de autoconfiguración de direcciones incluye la creación de la dirección de la red local y verificar que la dirección sea única en la red, así como también determinar que información deberia ser autoconfigurada (direcciones, otra información, o ambas). Existen tres mecanismos para obtener una dirección IPv6: mecanismos sin estado, un mecanismo de estado completo, o ambos Los mecanismos de autoconfiguración sin estado y de estado completo pueden ser usados simultáneamente. Los mensajes de notificacion de ruteador son los que se encargan de especificar el mecanismo que se esta usando y si se están usando ambos. Por ejemplo un host puede usa la autoconfiguración sin estado para configurar su propia dirección, pero podría usar la configuración de estado completo para obtener otra información de configuración.

4.5.1 Autoconfiguración sin estado

La autoconfiguración sin estado no requiere de una configuración manual del host , una mínima configuración de ruteadores (puede o no requerirse), no requiere servidores para la autoconfoiguración. Este mecanismo es usado cuando un sitio le interesa una direccion en especifico, siempre y cuando la dirección sea única y pueda ser ruteable(este mecanismo no lleva un control fuerte en la asignación de direcciones).

Con el mecanismo de autoconfiguración sin estado, un host genera su propia dirección usando dos elementos de información: Información localmente disponible, incluso información del mismo host y información adicional de las notificaciones de los ruetadores La información que es parte del host es llamada un identificador de interface, la cual identifica una interface en una subred La parte de la información que llega desde el ruteador, es una prefijo de dirección el cual identifica la asociación de la subred con un enlace Si un ruteador no existe en la subred, el host puede continuar generando un tipo especial de direcciones llamada, dirección de enlace local La dirección de enlace local solo puede ser usada para la comunicación entre nodos pertenecientes al mismo enlace.

La autoconfiguración sin estado, aplica solo a host, ya que los ruetadores usan otros medio para la autoconfiguración , aunque los ruteadores pueden generar su propia dirección de enlace local, y puede verificar que no esta duplicada en la red, cuando inicializan.

Las direcciones IPv6 son concedidas a una interface para un periodo de tiempo en particular, el cual puede ser infinito. Este tiempo de vida es el tiempo que la dirección IPv6 puede ser relacionada a la interface. Cuando el tiempo de vida expira, la asociación entre la interface y la dirección IPv6 son inválidos y la dirección puede ser reasignada a otra interface. En soporte a la culminación de la asociación de la dirección y la interface por la expiración del tiempo de vida, la asignación de direcciones puede tener dos fases: fase *preferida*, significando que el uso de la dirección no tiene restricciones; y la fase *desaprobada*, indicando que ya no se fomenta uso de esta dirección, anticipando que esta dirección llegara a ser invalida y si una comunicación usara esta dirección podría fallar.

Una dirección que se encuentra en la fase preferida, se encuentra en la fase desaprobada cuando el tiempo de vida de la fase preferida expira. Una dirección que se encuentra en fase desaprobada, podría continuar siendo usada como dirección origen en comunicaciones existentes, pero no puede ser usada en nuevas comunicaciones con otros nodos, si una dirección sustituta esta disponible y tiene suficiente alcance o tiempo de vida. El protocolo IP y protocolos de capas superiores como TCP o UDP deben continuar aceptando datagramas destinadas a la dirección que se encuentra en fase desaprobada, siendo aun una dirección valida para la interface.

Para cuando la dirección pasa a una fase desaprobada, el host puede generar una nueva dirección para la interface, o hacer una solicitud al servidor DHCP solicitando que actualice el tiempo de vida de la dirección (esto solo se realiza cuando existe un servidor DHCP en la red).

La dirección de enlace local es generada por la combinación de un prefijo de la dirección de enlace local y un identificador de interface que comúnmente es de una longitud de 64 bits aunque puede variar. El identificador de interface es específico para la topología LAN o WAN que se este usando, en muchos casos este identificador se deriva de la dirección de hardware (dirección física) que reside en la ROM de la tarjeta de red.

En la figura 4.31 se muestra un diagrama donde se ven los pasos que se siguen para la autoconfiguración de direcciones sin estado.

Una vez que se genera la dirección tentativa con el prefijo de la red y con el identificador de la interface, ahora se determina que esta dirección tentativa sea única en la red, transmitiendo un mensaje de Solicitud al nodo vecino con la dirección tentativa como Address Target. Si otro no esta usando esta dirección, un mensaje de Notificación de Nodo vecino es regresado hasta el nodo que generó el mensaje de solicitud; en este caso la autoconfiguración se detiene y se requiere de una intervención manual.

Si ninguno de los mensajes de Notificación de nodo vecino es regresado, con respecto a la solicitud hecha, la dirección tentativa es considerada como única en la red y en estos momentos el nivel IP tiene la posibilidad de comunicación.

Host y ruteadores pueden usar hasta aquí este proceso de autoconfiguración, para generar direcciones de enlace local.

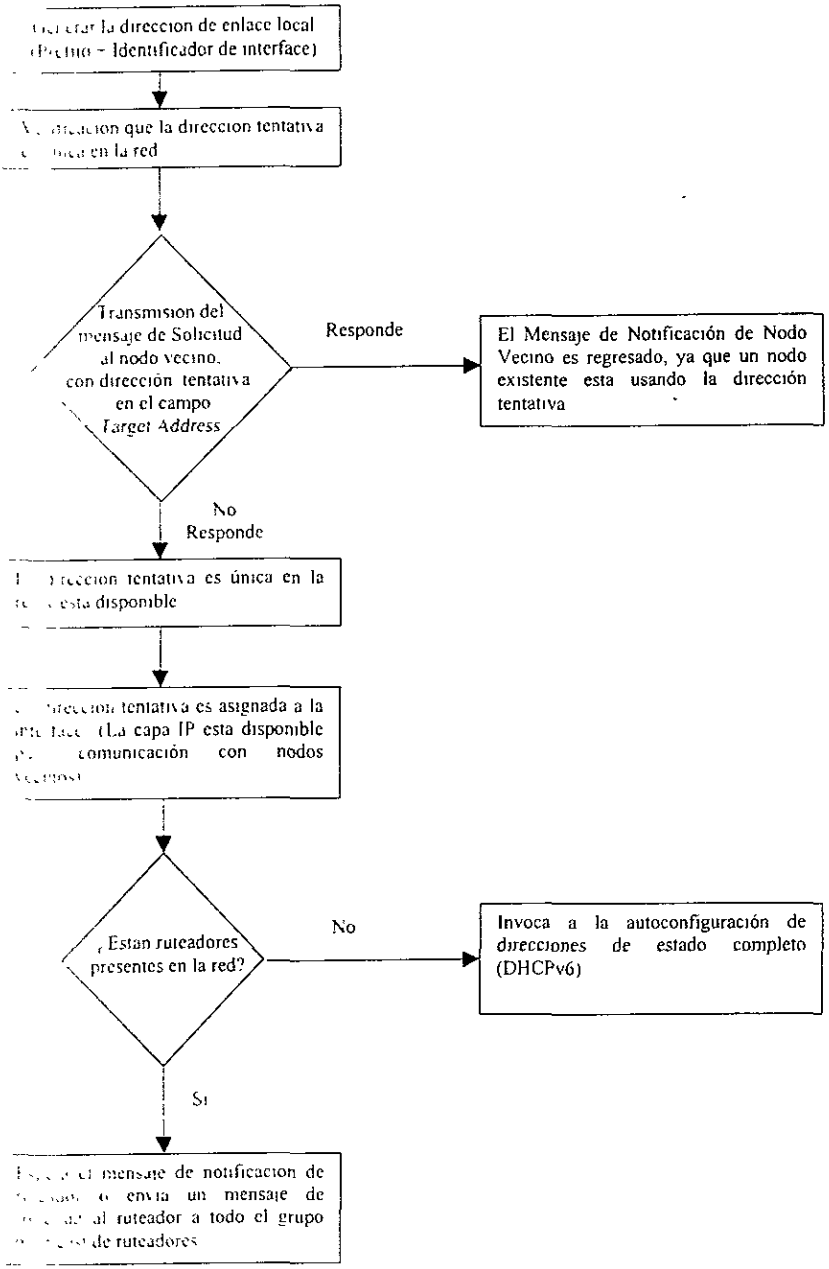


Fig 4.31 - *Diagrama del proceso de Autoconfiguración de direcciones sin estado*

El siguiente paso es ejecutado solo por los host, este involucra escuchar mensajes de Notificación de ruteador, que periódicamente son transmitidos, o se puede forzar a que estos mensajes sean transmitidos, transmitiendo un mensaje de solicitud de ruteador. Si no se reciben mensajes de notificación de ruteador, significa que rueteadores no existen en la red, por lo cual se hace uso de un método de autoconfiguración de estado completo como lo es DHCPv6, que es usado para completar el proceso de autoconfiguración, dando a conocer alguna otra información de autoconfiguración.

Si hay ruteadores presentes, los mensajes de Notificación de ruteador serán enviados periodicamente Estos mensajes incluyen dos banderas M y N, las cuales son usadas en proceso de autoconfiguración

- La bandera M (administración de configuración de direcciones) es habilitada cuando es puesta a 1. Esta bandera indica que los host deben usar un protocolo de administración de estado completo, para la autoconfiguración de direcciones, adicionalmente a la autoconfiguración de direcciones sin estado usada para obtener la direccion

- La bandera O (Configuración de estado completo) es habilitada cuando es puesta 1. Esta bandera al estar habilitada indica que los host deberán de usar protocolos de autoconfiguración de estado completo, para la configuración de información adicional.

Los mensajes de notificación de ruteador, como se analizo anteriormente pueden incluir una o mas opciones como, entre ellas la opción de Información de prefijo que incluye dos banderas L y A, que pueden ser usadas con la autoconfoiguración de direcciones.

- La bandera L es habilitada cuando es puesta en 1, esta bandera indica que el prefijo que es enviado en el mensaje de notificación de rutreador puede ser usado para la determinación de direcciones que esta asignadas a una interface en una misma red.

- La bandera A es habilitada cuando es puesta en 1, esta bandera indica que el prefijo que es enviado en el mensaje de notificación de rutreador puede ser usado para la configuración de una dirección autónoma.

4.5.2 Protocolo de Configuración Dinámica de Host versión 6 (DHCPv6, *Dynamic Host Configuration Protocol versión 6*)

En casos en que una dirección duplicada exista o no existan ruteadores en la red, debe usarse un proceso de autoconfiguración de estado completo. La propuesta de DHCPv6 provee estos parámetros de configuración para los nodos. DHCPv6 consiste de dos elementos: un protocolo que entrega información de configuración a nodos específicos. desde un servidor DHCPv6 hasta un cliente; y un mecanismo para asignamiento de direcciones de red y otros parámetros a nodos IPv6.
DHCP es un mecanismo que permite a los administradores de sistemas locales un control sobre los parámetros de configuración., por ejemplo un administrador podría ser capaz de

hacer cumplir políticas locales de referentes a la asignación y acceso de los recursos locales

DHCP no requiere de una configuración manual de clientes DHCP, excepto para los requerimientos de seguridad. DHCP no requiere de un servidor por cada segmento de red, aunque se puede dar el caso que se tenga mas de uno para asegurar la fiabilidad y rendimiento. Los servidores DHCP son capaces de realizar actualizaciones dinámicas al DNS

Existen varios tipos de nodos funcionales DHCPv6, que son definidos como:

- Cliente DHCPv6: Un nodo que inicia solicitudes en una red, para obtener parámetros de configuración.
- Servidor DHCPv6 Un nodo que responde a las solicitudes de los clientes para proveer direcciones, longitud de prefijo, o otro tipo de parámetros de configuración.
- Difusor DHCPv6 Un nodo que actúa como intermediario para entregar mensajes DHCPv6 entre clientes y servidores.
- Agente DHCPv6. Es un nodo que es un servidor o difusor DHCPv6

DHCPv6 es construido en un modelo cliente/servidor, el cual cuenta con un total de 6 mensajes de solicitud y respuestas para la comunicación de los parámetros:

- Mensaje de Solicitud DHCPv6. EL Mensaje de solicitud DHCPv6 es un mensaje de tipo multicast, enviado por un cliente a uno o más agentes, o es un mensaje que es retransmitido a uno o más servidores, el formato se puede observa en la figura 4.32.

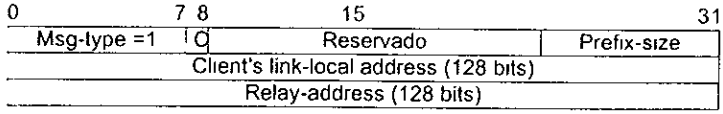


Fig. 4.32 - *Mensaje de solicitud DHCPv6.*

El campo msg-type identifica al mensaje con el valor de 1, tiene una longitud de 8 bits.

El campo con la bandera C cuando se pone en 1 el cliente solicita, que todos los servidores que reciban el mensaje designen los recursos asociados con el cliente.

El campo prefix-size dice el tamaño del prefijo de la dirección IPv6 del agente, tiene una longitud de 8 bits.

El campo Client's link-local address tiene una longitud de 128 bits, contiene la dirección IP de enlace local de la interface del cliente, desde la cual se hace la petición.

El campo Relay-address si no es cero, contiene la dirección IP del nodo difusor DHCPv6, en la cual este mensaje será recibido.

Un cliente deberá enviar un mensaje de solicitud DHCP para el grupo multicast de los agentes DHCP, poniendo el campo Relay-address en cero Cualquier nodo difusor que reciba este mensaje deberá retransmitir el mensaje hacia el grupo multicast de servidores DHCP

- Mensaje de Notificación DHCPv6 El mensaje de notificación es un mensaje de tipo unicast enviado por un agente DHCP en respuesta a un mensaje de solicitud DHCP realizada por un cliente DHCP, el formato se puede observa en la figura 4.33.

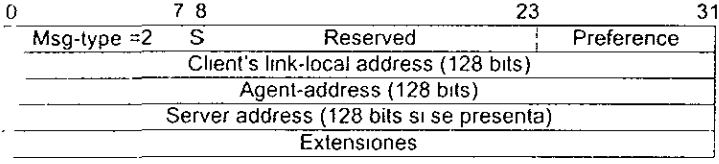


Fig. 4 33 - *Mensaje de Notificación DHCPv6*

El campo msg-type identifica al mensaje con el valor de 2, tiene una longitud de 8 bits.

El campo con la bandera S sirve para saber si el mensaje de Notificacion DHCPv6 contedra la direccion del servidor. Cuan la bandera S tiene el valor 1 el mensaje contedra la dirección del servidor en el campo server address y cuando tenga el valor 0 se omitira la direccion del servidor.

El campo Client's link-local address tiene una longitud de 128 bits, contiene la dirección IP de enlace local de la interface del cliente, desde la cual hace la petición

El campo Agent-address tiene una longitud de 128 bits, contiene la dirección IP del agente DHCP perteneciente a la misma red del cliente.

El campo Server address tiene una longitud de 128 bits, contiene la dirección IP del servidor DHCP en el caso se que exista en la misma red.

Supongamos que el servidor en la misma red usa un mensaje de notificación DHCPv6 en respuesta del mensaje de solicitud DHCPv6 enviado a la dirección multicast del grupo de agentes, entonces la dirección del agente será una dirección IP de uno de las interfaces del servidor de la misma red del cliente, y la bandera S será puesta en cero, indicando la ausencia del campo del server/address.

En situaciones donde no existen ruteadores en la red , el servidor DHCPv6 debe ser configurado en la misma red.

- Mensaje de Petición DHCPv6: El mensaje de petición DHCP es un mensaje de tipo unicast, enviado por un cliente a un servidor, para pedir parámetros de configuración de la red, el formato se puede observa en la figura 4.34.

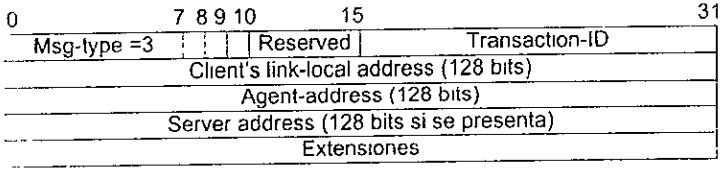


Fig. 4 34 - *Mensaje de petición DHCPv6*

El campo msg-type identifica al mensaje con el valor de 3, tiene una longitud de 8 bits.

El campo con la bandera C cuando se pone en 1, indica que el cliente pide al servidor que remueva todos los recursos asociados con el cliente, excepto los recursos provistos como extensiones; la bandera S cuando se pone en 1, el campo server/address con la dirección del servidor es puesta en el mensaje; la bandera R cuando se pone en 1 indica que el cliente fue inicializado y que requiere que todos sus identificadores de transacciones anteriores sean borrados y este disponible para reusar los identificadores,

El campo transaction-ID, es un entero que se va incrementando gradualmente, que se usa para identificar las peticiones del cliente, y es copiado en el mensaje de contestación DHCPv6

El campo Client's link-local address tiene una longitud de 128 bits, contiene la dirección IP de enlace local de la interface del cliente, desde la cual hace la solicitud.

El campo Agent-address tiene una longitud de 128 bits, contiene la dirección IP del cliente DHCP perteneciente a la misma red del cliente.

El campo Server address tiene una longitud de 128 bits, contiene la dirección IP del servidor DHCP en el caso de que exista en la misma red.

En el campo de extensiones van los parámetros opcionales que pide para la configuración

- Mensaje de Contestación DHCPv6: El mensaje de contestación DHCPv6 es un mensaje de tipo unicast enviado por un servidor en respuesta para un mensaje de petición DHCP enviado por un cliente. Las extensiones de este mensaje describen los recursos que el servidor esta asignando a un cliente, y puede contener otra información que puede ser usada por el cliente, el formato se puede observa en la figura 4 35

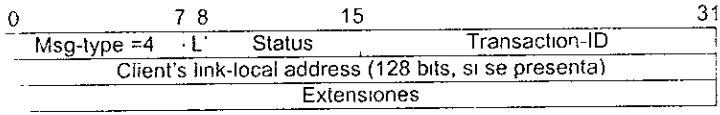


Fig 4.35 - *Mensaje de contestación DHCPv6*

El campo msg-type identifica al mensaje con el valor de 4, tiene una longitud de 8 bits.

El campo con la bandera L cuando se pone en 1, indica que la dirección de enlace local de cliente esta presente

El campo status (estado), maraca el estado del paquete y contiene los siguientes valores

0	Exitoso
16	Fallido, Sin razón especificada
17	Autenticación fallida o no presente
18	Petición pobremente formada, sin suficiente información
19	Recursos no disponibles
20	Registro de cliente no disponible
21	Dirección IP del cliente, es invalida
22	El agente de difusión no puede encontrar la dirección del servidor
64	Servidor DHCPv6 inalcanzable

El campo Transaction/ID, es un entero que se va incrementando gradualmente , que se usa para identificar el mensaje de contestación DHCPv6, y es copiado desde el mensaje de petición DHCPv6

El campo Client's link-local address tiene una longitud de 128 bits, contiene la dirección IP de enlace local de la interface del cliente, desde la cual hace la petición.

En el campo de extensiones van los parámetros opcionales que pide para la configuración

Si la bandera L es puesta en 1, y de esta manera la dirección de enlace local del cliente esta presente en el mensaje de contestación DHCPv6, el servidor envía el mensaje hasta un nodo de difusión DHCPv6, el cual fue especificado en el campo agent/address del mensaje de petición DHCPv6y el nodo difusor usa la dirección del cliente para enviarle el mensaje de contestación del servidor, (esto es cuando el servidor no se encuentra en la misma red del cliente).

- Mensaje de Liberación DHCPv6: EL mensaje de liberación DHCP es un mensaje de tipo unicast, enviado por un cliente para informar al servidor que el cliente esta liberando recursos, el formato se puede observa en la figura 4.36.

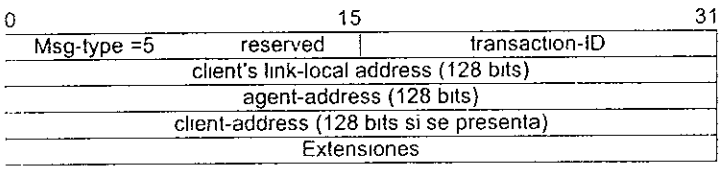


Fig. 4.36 *Mensaje de liberación DHCPv6*

El campo msg-type identifica al mensaje con el valor de 5, tiene una longitud de 8 bits.

El campo con la bandera D cuando se pone en 1, indica que el cliente da instrucciones al servidor para enviar directamente de regreso la contestación DHCP, en lugar de usar la dirección del agente (agent-address) o la dirección de enlace local, para que un nodo difusor mande el mensaje de contestación

El campo transaction/ID, es un entero que se va incrementando gradualmente, que se usa para identificar el mensaje de liberación, y es copiado en el mensaje de contestación DHCPv6

El campo Client's link-local address tiene una longitud de 128 bits, contiene la dirección IP de enlace local de la interface del cliente, desde la cual envía el mensaje de liberación DHCPv6

El campo Agent-address tiene una longitud de 128 bits, contiene la dirección IP del agente DHCPv6.

Client-address tiene una longitud de 128 bits, contiene la dirección IP global del cliente, desde donde se emitió el mensaje de liberación DHCPv6. El campo Agent-address solo estara presente si la bandera D es puesta en 1, aun si es la misma dirección que la direccion de enlace local.

El campo de extensiones contiene los parámetros de configuración que quiere el cliente que sean liberados.

- Mensaje de reconfiguración DHCPv6: El mensaje de reconfiguración DHCP, es un mensaje de tipo unicast o multicast, enviado por un servidor para informar a uno o mas clientes, que el servidor tiene nueva información de configuración importante para el cliente, para poder reconfigurarlos los parámetros de configuración que se indican en las extensiones, cada cliente deberá iniciar una nueva transacción de solicitud/contestación El formato se puede observa en la figura 4.37.

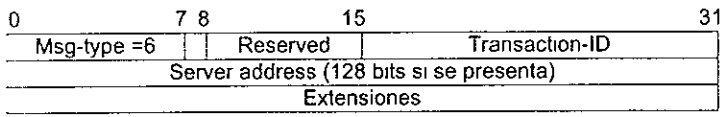


Fig. 4.37 - *Mensaje de reconfiguración DHCPv6*

El campo msg-type identifica al mensaje con el valor de 6, tiene una longitud de 8 bits.

El campo con la bandera N cuando se pone en 1, indica que el cliente no podría esperar una contestación DHCP en respuesta a la petición dhcpv6, y el cliente en respuesta envia un mensaje de reconfiguración DHCPv6.

El campo transaction-ID, es un entero que se va incrementando gradualmente, que se usa para identificar las peticiones del cliente, y es copiado del mensaje de petición hecha por el cliente.

El campo server-address, indica la dirección del servidor que emitió el mensaje de reconfiguración.

El campo de Extensiones indican los parámetros que deben ser reconfigurados.

Todos los mensajes excepto el de solicitud DHCP, tiene un campo conocido con el nombre de extensiones, en el cual tiene varias opciones, las cuales son los parámetros de configuración, como lo es la dirección IP, Tiempo de vida de la dirección IP, el DNS especificando el DNS por default que el cliente puede usar, parámetros de aplicación y de servicio, parámetros de TCP, Autenticación cliente/servidor, entre otros.

Para encontrar un server, un cliente envía un mensaje de solicitud DHCPv6, desde la interface de red que quiere configurar El cliente entonces espera un mensaje de notificación DHCPv6, donde contendrá la dirección IP de un servidor DHCP. La transacción empieza cuando el cliente envia un mensaje de petición DHCP y El servidor envia un mensaje de contestación DHCP de tipo unicast (posiblemente mediante un difusor DHCPv6) En este punto todos los datos han sido transmitidos y se presume que han sido recibidos.

Para proveer de un mecanismo de recuperación, si el cliente o el servidor no recibe su mensaje, el cliente retransmite cada mensaje de petición DHCP hasta que este mismo mensaje convoca al correspondiente mensaje de contestación DHCP, o hasta que el cliente este seguro que el servidor DHCP deseado no esta disponible, o que el cliente no quiere una respuesta (como que aborte la transacción).

El protocolo DHCP usa como protocolo de transporte a UDP, para la comunicación entre el cliente y el servidor, aunque UDP no es confiable, la retransmisión DHCP antes mencionada provee fiabilidad ente el cliente y el servidor.

Las siguientes direcciones multicast son usadas por agentes y clientes DHCP:

FF02:0:0:0:0:0:1:2
: Esta dirección multicast pertenece a los agentes DHCP (servidores o difusores DHCPv6)

FF05 0:0:0:0:0:1:3
: Esta dirección multicast pertenece al grupo de servidores DHCPv6.

FF05:0:0:0:0.0:1:4
: Esta dirección multicast pertenece al grupo de difusores DHCPv6.

4.6. RUTEO EN IPV6

Los protocolos de ruteo se dividen en dos categorías: Protocolos de puerta Interior (IGP, *Interior Gateway Protocol*), y Protocolos de puerta Exterior (EGP, *Exterior Gateway Protocol*) Un IGP es usado para transmitir información de ruteo dentro de un mismo sistema autonomo(AS, *Autonomous System)*, es una red que es administrada por una

sola entidad Un EGP es usado para transmitir información de ruteo entre varios sistemas autónomos.

El protocolo de información de ruteo (RIP, *Routing Information Protocol*) y el Protocolo OSPF (Open Shortest Paht First), son protocolo de tipo IGP, y el protocolo BGP (Border Gateway Protocol) es un protocolo de tipo EGP.

El ruteo en el 6bone es estructurado en base a una jerarquía. La jerarquía esta conformada de sitios seudo TLA(pseudo Top Level Aggregator), que son sitios de mayor jerarquia en base a la dirección global Unicast que esta dividida en proveedores , también en sitios seudo NLA (pseudo Next Level Aggregator) que es otra jerarquía de la dirección Unicast y sitios locales. Tomando en cuenta de que las jerarquías TLA y NLA, pueden estar conformadas de varios niveles. Esto se estudia mejor en el punto 5.1.6.

El backbone de la red 6bone esta formada únicamente de sitios pTLA, Y cada sitio pTLA conecta a sitios pNLA y proveedores. Los sitios pTLA usan el protocolo BGP4+ Border gateway Protocol) que es un protocolo de tipo EGP, los sitios pNLA pueden usar también BGP4+ o un protocolo de tipo IGP.

4.6.1 Routing Information Protocol para IPv6

RIP es uno de los protocolos IGP mas comúnmente usados, la versión de RIP con IPv6 es nombrado RIPng (Routing Information Protocol next Generation).

RIP es un protocolo basado en un algoritmo de vector de distancia, con una historia que data desde los días de ARPANet. RIP fue diseñado primeramente para redes de tamano moderado, con pocas limitaciones como:

- El protocolo esta limitado para redes , cuyo camino o ruta no exceda los 15 saltos.
- El protocolo depende de un proceso llamado " cuenta hasta infinito (counting to infinity) cuando llega a 16 saltos "para resolver algunos problemas tales como rutas de vuelta (loop). Este proceso puede consumir una gran cantidad de ancho de banda de la red antes de la resolución de que el paquete esta dando vueltas por la red.
- El protocolo depende en una métrica fija para comparar rutas alternativas, sin considerar parámetros de tiempo real tales como retrasos, alcanzabilidad o exceso de trafico

RIPng es el protocolo que permite a los ruteadores intercambiar información, para calcular rutas a través de una red interna IPv6. Cada ruteador que tiene implementado RIPng tiene una aval de ruteo que tiene una entrada para cada uno de los destinos IPv6 que son alcanzables

Cada entrada tiene lo siguiente:

- El prefijo IPv6 del destino
- Una métrica que indica el numero total en saltos , para que el datagrama llegue a su destino desde el ruteador

- La dirección IPv6 del siguiente ruteador a lo largo del camino, hasta el destino, llamado salto siguiente (Next Hop).
- Una bandera llamada cambio de ruta (Route Change), que indica si la información de la ruta ha cambiado recientemente.
- Varios tiempos, tal como el tiempo para transmitir la información de la tabla de ruteo a ruteadores vecinos el cual es cada 30 segundos.

RIPng esta basado en UDP, con el cual envía y recibe paquetes desde el puerto 521.

El formato del paquete RIPng consta de 3 campos como se puede observar en la figura 4.38

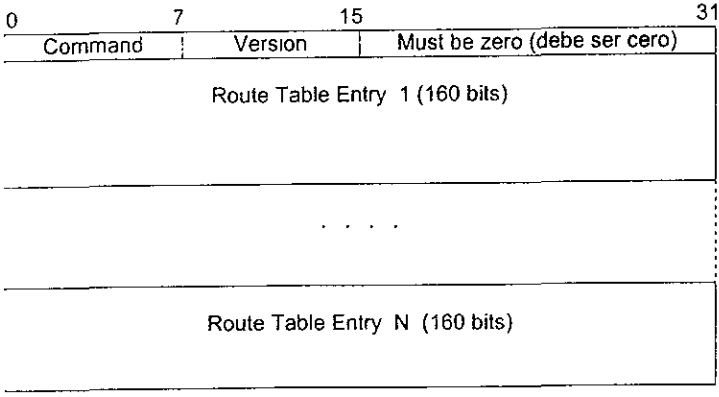


Fig. 4.38 *Formato del paquete de RIPng*

El campo Command contiene informacion sobre si el paquete es una solicitud a ruteadores vecinos o una respuesta a ruteadores vecinos.

La solicitud es usada para pedir una respuesta donde un paquete contenga todas las entradas de una tabla de ruteo o parte de la tabla de ruteo de los ruteador. Normalmente las solicitudes enviadas por lo ruteadores, son emitidas como un paquete multicast, para que el paquete les llegue a todos los ruteadores vecinos, para poder completar su tabla de ruteo lo mas pronto posible, aunque pude darse el caso de que la solicitud solo se le haga un solo ruteador.

Si el campo Command contiene que es una respuesta a ruteadores vecinos, el paquete contendrá las entradas de la tabla de ruteo de los nodos vecinos. La respuesta puede darse por una previa solicitud o por una actualización regular sin previa solicitud, y esta puede ser por cambio de ruta o de algún nodo que ya no esta en la red.

El campo versión contiene la versión del protocolo RIPng.

Los campos Route Table Entry (RTE), contiene la información de cada nodo.

Cada RTE tiene un formato 4 campos como se muestra en la figura 4.39.

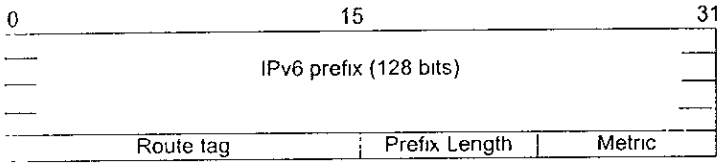


Fig 4.39 - *Formato de la entrada de la tabla de ruteo (RTE, Route Table Entry)*

El campo IPv6 prefix contiene el prefijo contiene el prefijo de la dirección IPv6 de las redes que son alcanzables.

El campo route tag, contiene un atributo asignado a una ruta, el cual debe ser preservado , difundido. La intención de este campo, es de proveer un método para separar rutas internas RIPng (rutas que se encuentran dentro de un dominio RIPng) de rutas externas RIPng las cuales puede haber sido importadas desde un EGP o un IGP.

El campo prefix length, contiene la longitud en bits de la parte significativa del prefijo que puede ser un valor entre 0 y 128.

El campo metric contiene el valor de 1 a 15, especificando la métrica (numero de saltos) para que el paquete llegue a su destino, o el valor de 16 de inalcanzable

RIPng también tiene la habilidad de proveer la dirección IPv6 del siguiente salto inmediato para los paquetes El salto siguiente es especificado por una especial RTE, conocida como RTE de salto siguiente, esta se muestra en la figura 4 40

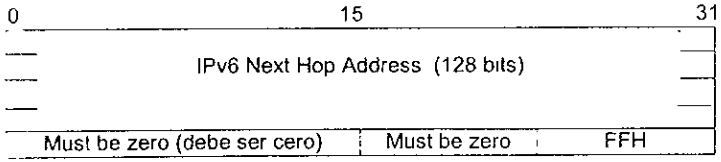


Fig 4 40 - *Formato de la entrada de la tabla de ruteo de siguiente salto (RTE Next Hop)*

En esta entrada de tabla de ruteo, el campo IPv6 Next Hop address especifica la dirección IPv6 del salto siguiente del paquete.

En el campo metric es identificado por el valor FFH.

4.6.2 Protocolo OSPF para IPv6(Open Shortest Paht First Protocol for IPv6)

El protocolo OSPF usa un algoritmo de estado de enlace (LSA, *Link State Algorithm*), este protocolo es de tipo IGP. EL algoritmo LSA tiene varias ventajas en comparación con el algoritmo usado para RIPng, como lo son: adaptación más rápida a cambios dentro de la red, configuración jerárquica de topología, calcula el costo mínimo de rutas múltiples para balancear el trafico sobre varios caminos, también permite el uso de máscaras de longitud variable

El protocolo OSPF distribuye información de ruteo en sólo sistema autónomo.

Cada ruteador OSPF mantiene una base de datos del sistema autónomo, que es idéntica a la de los demás ruteadores del mismo sistema autónomo. OSPF también usa direcciones IP multicast para enviar y recibir actualizaciones, para así calcular rapidamente las rutas mas optima.

El protocolo OSPF envía notificaciones de estado de enlace, a los demás ruteadores en base a la jerarquía del área.

El protocolo OSPF sufrio algunos cambios para que soportara el protocolo IPv6, los más importantes son los siguientes:

- Soportaran prefijos para la identificación de los enlaces en lugar de usar las máscaras, también los registro de estado de enlace soportan la dirección de 128 bits en lugar de 32 bits
- Se removió la autenticación de la cabecera del paquete OSPF, como esta función es ahora cubierta con las cabeceras de autenticación y ESP de IPv6.
- La versión del protocolo cambia a la numero 3.
- Los ruteadores son identificados por in identificador de ruteador en un mismo enlace
- Se removieron las semánticas de tipo de servicio soportado en OSPF, ya que IPv6 soporta este servicio con el campo Flow Label de la cabecera IPv6.

El protocolo OSPF tiene definidos cinco tipos de paquetes, todos estos paquetes tienen una cabecera OSPF común. El formato se puede observar en la figura 4.41.

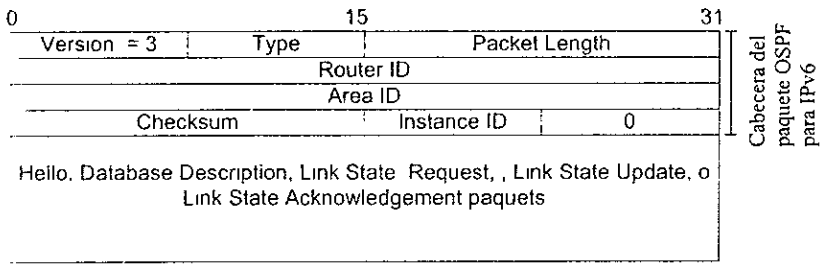


Fig. 4.41 - *Cabecera común del Formato OSPF*

EL campo versión contiene la versión del protocolo OSPF que en este caso es la versión 3

El campo Type contiene el numero que identifica el tipo de paquete, que son: 1 para el paquete Hello, 2 para el paquete Databse Description, 3 para el paquete Link state request, 4 para el paquete Link state Update, 5 para el paquete Link state acknowledgement

El campo Packet Length contiene la longitud del paquete en unidades de bytes.

El campo Router ID contiene el identificador del ruteador

El campo Area ID, contiene el identificador del área, a la cual el paquete pertenece.

El campo Checksum contiene la

El paquete Hello, se encarga de establecer la relación entre los vecinos. El ruteador envía un paquete Hello por todas sus interfaces periódicamente a los ruteadores vecinos para establecer y mantener las relacione entre ellos.

El paquete Database Description, describe el contenido de la base de datos del estado del enlace, estos paquetes son intercambiados cuando un ruteador adyacente es inicializado

El paquete Link State Request, después de que un ruteador intercambia la descripción de la base de datos con un ruteador vecino, el ruteador podría darse cuenta que parte de su base de datos del estado del enlace están sin actualizarse, por hace una solicitud del estado del enlace, usando este paquete para pedir fragmentos de la base de datos del los ruteadores vecinos.

El paquete Link State Update, este paquete se encarga de actualizar el estado del enlace a los demás ruteadores, este tipo de paquete solo puede ir un salto mas allá del origen, este paquete es de tipo multicast.

El paquete Link State Acknoledgment, sirve para realizar un reconocimiento del estado del enlace, cuando se envían torrentes de estos paquetes se realiza el reconocimiento del enlace ya que en estos paquete van notificaciones del estado del enlace.

Existen varios tipos de notificaciones de estado del enlace (LSA, *link State advertisement*), como:

Notificaciones del enlace del ruteador (RLA, *Router Link Advertisements*), estos describen los estados de enlace de los ruteadores de una área especifica. Un ruteador envía un RLA para cada área a la cual pertenece, describiendo el funcionamiento de sus interfaces, o los enlaces del área a los que puede accesar

Notificaciones del enlace de la red (Network Link Advertisiments)estas notificaciones describen a todos los ruteadores que pertenece a una misma red, esta notificación debe realizarla el ruteador que es designado como maestro en toda la red. Estas notificaciones las hace para todos los ruteadores.

Notificaciones de resúmen de enlace (Summary Links Advertisements), esta notificación describe las rutas internas de un sistema autónomo, el rango de direcciones del sistema autónomo y el ruteador fronterizo del sistema autónomo

Notificación del enlace externo al sistema autónomo, esta notificación describe rutas a destinos externos al sistema autónomo.

4.6.3 BGP-4 (Border Gateway Protocol version 4)

El protocolo BGP es un protocolo de tipo EGP. Los sistemas autónomos intercambian su información de ruteo usando un protocolo EGP, el protocolo BGP-4 es usado para intercambiar información de ruteo entre distintos sistemas autónomos del backbone IPv6.

BGP-4 usa TCP para hacer mas fiable la comunicación entre los sistemas autónomos Este protocolo consta de cuatro tipos de mensajes que son:

- Mensaje de Abertura (Open Message). Este mensaje se encarga de inicializar la conexión BGP
- Mensaje de Actualización (Update Message): Este mensaje es usado para transferir información de ruteo entre los sistemas autónomos
- Mensaje de conservación de la conexión (Keeplive Message) Este mensaje se encarga de verificar durante periodos de tiempo que el destino se encuentre en un estado de alcanzabilidad.
- Mensaje Notificación (Notification Message) Este mensaje se envia cuando una condición de error es detectada, por lo que causa que la conexión BGP sea cerrada.

Después que la conexión TCP es establecida, el primer mensaje que se envía es el mensaje de abertura, si el mensaje de abertura es aceptado en el otro lado de la conexion, un mensaje de conservación de la conexión (Keeplive Message) es regresado como confirmación, hasta donde se origino el mensaje de abertura. Ya que se confirmo el enlace de comunicación los mensajes de Actualización, de conservación de la conexión y de notificación pueden ser intercambiados entre los sistemas autónomos para mandar la información de ruteo.

Antes de enviar información a otro sistema autónomo es necesario que el ruteador BGP asegure la alcanzabilidad del ruteador BGP del sistema autónomo con que se quiere tener la comunicación.

En el protocolo BGP-4 se han hecho cambios para que soporte el protocolo IPv6, el principal cambio fue el tamaño de la dirección IP, de 32 bits a 128 bits, soportando así toda la arquitectura de direccionamiento de IPv6

4.7 Movilidad en IPv6

El protocolo IPv6 soporta morbilidad de los nodos, esto significa que un nodo puede conectarse a Internet mediante un medio inalámbrico.

La movilidad en Ipv6 maneja algunos términos que no hemos manejado anteriormente, por lo que a continuación explicaremos esos términos:

Direccion local (home address) Es aquella dirección IPv6 que identifica a un nodo movil en su enlace local.

Enlace local (home Link): Es el enlace en el cual el prefijo de la subred del nodo móvil es definido

Nodo movil (mobile node): Un nodo que puede cambiar el punto de unión de un enlace local a otro

Enlace Exterior (Foreign Link) Cualquier otro enlace que no sea el enlace local del nodo movil

Agente local (Home agent) Es un ruteador en un enlace local con el cual el nodo movil registra su dirección de enlace exterior activa. Este ruteador debe soportar ser agente local en un enlace móvil.

Direccion de enlace local (home address): Dirección del nodo móvil en su enlace local.

Dirección de enlace exterior (care-of address). Es una dirección IPv6 asociada con un nodo móvil mientras se encuentra en un enlace exterior.

Un nodo móvil es siempre direccionable por su dirección local, ya sea que esté en su enlace local o en un enlace exterior. Mientras un nodo móvil se encuentre dentro del enlace local al que pertenece originalmente, todos los paquetes son direccionados de la misma manera como si el nodo estuviera estático en un solo sitio

Cuando un nodo móvil se encuentre en un enlace exterior, los paquetes del nodo serán direccionados por una o mas direcciones de enlace exterior además de su dirección local El prefijo de la subred de la dirección de un enlace exterior del nodo móvil, es el prefijo del enlace exterior donde se encuentra el nodo móvil. Un nodo móvil típicamente adquiere una dirección de enlace exterior mediante los mecanismos de autoconfiguración de direcciones sin estado y DHCPv6, otro método de adquirir la dirección de enlace exterior es mediante una asignación estática que es realizada por el administrador de la red

Cuando un nodo móvil se encuentra en un enlace exterior, el nodo móvil registra una de sus direcciones de enlace exterior con un ruteador en su enlace local, solicitando al ruteador que haga la función de agente local para el nodo móvil. La asociación de la dirección de enlace exterior con la dirección local del nodo móvil es hecha por el nodo móvil al enviar un paquete con la cabecera de destination option hasta el agente local, la opción contenida, es la opción de actualización de dicha asociación llamada binding Update. ya que se realizo la asociación entre la dirección de enlace exterior con la direccion local del nodo móvil, la dirección de enlace exterior ahora se conoce como dirección primaria de enlace exterior ya que es la dirección primaria del enlace exterior

que esta siendo utilizada. Después el agente local del nodo móvil intercepta cualquier paquete IPv6 que es dirreccionado a la dirección local del nodo móvil en el enlace local y cada paquete interceptado es enviado por un túnel mediante encapsulación IPv6 hasta la direccion primaria de enlace exterior

Al estar un nodo móvil en un enlace exterior, algunos nodos de su enlace local pueden haber cambiado su configuración, como lo es el ruteador que estaba operando como su agente local, por lo que en este caso el nodo móvil puede no conocer la dirección IPv6 de un agente local, por lo que el nodo móvil usa un mecanismo conocido como Descubrimiento de la dirección del agente local (Dynamic home agent address discovery', que le permite a un nodo móvil conocer la dirección IP de su agente local, con el cual podra registra su dirección primaria de enlace exterior

La movilidad en IPv6 tiene cuatro tipos de opciones que se transmiten en la cabecera Destination Option, estas opciones son las siguientes:

- ***Binding Update:*** Esta opción es utilizada por el nodo móvil, para notificar a un nodo movil o fijo o incluso al mismo agente local su asociación actual, entre la dirección de enlace exterior y la dirección de enlace local del nodo móvil. La opción Binding Update es enviada en la cabecera de Destination Option hasta el agente local del nodo móvil, para registrar su dirección primaria de enlace exterior. Cualquier paquete que incluya esta opción debe incluir una cabecera de autenticación o una cabecera Encapsulating Security Payload.

- ***Binding Acknoledgement:*** Esta opción es usada para ser enviada como contestación cuando un nodo recibe un paquete con la opción Binding Update, notificando así si la asociación entre las direcciones de enlace exterior y de enlace local fue aceptada o si no lo fue. Cualquier paquete que incluya esta opción debe incluir una cabecera de autenticación o una cabecera Encapsulating Security Payload

- ***Binding Request:*** Esta opción es usada para solicitar a un nodo móvil información acerca de la actualización de la asociación de su dirección primaria de enlace exterior y su dirección de enlace local, para que así el nodo móvil envíe un paquete con la opción Binding Update. Esta opción es usada para cuando un nodo necesita renovar su infamación acerca de la asociación de las direcciones del nodo móvil, ya que la información de dicha asociación ha expirado su tiempo de vida.

- ***Home Address:*** Esta opción es usada para informar la dirección local del nodo móvil. Cuando el nodo móvil se encuentra en un enlace exterior usa la dirección de enlace exterior como dirección origen en el paquete, pero envía la dirección de enlace local en la opción Home Address. Para el nodo que recibe el paquete, el procesamiento es transparente ya que es capaz de substituir la dirección de enlace local por la dirección primaria de enlace exterior que la que actualmente utiliza el nodo móvil.

Cuando un nodo móvil envía un paquete con la opción Binding Update, hasta una direccion anycast de los agentes locales de su enlace local, así con esto alcanza uno de los ruteadores que están actuando como agentes locales; este agente local rechaza el paquete con la opción Binding Update pero envía un paquete con una opción Binding Acknowledgement, conteniendo una lista de todos las direcciones IP de los agentes locales pertenecientes a su enlace local.

Siempre que un paquete es enviado con la opción Binding Update, el paquete también debe incluir la opción Home address, para así indicara la asociación entre las direcciones de enlace local y la dirección primaria de enlace exterior

Cuando un nodo móvil o fijo desea enviar un paquete hasta un nodo móvil, primeramente examina la información contenida en memoria de la dirección destino del nodo para quien va destinado el paquete, si el nodo emisor tiene alguna asociación entre las direcciones de enlace externo y enlace local del nodo móvil, el nodo emisor envía el paquete usando una cabecera de ruteo colocando como dirección destino la dirección primaria de enlace externo del nodo móvil

Si en cambio, el nodo que enviará el paquete no tiene ninguna información en memoria de alguna asociación entre las direcciones de enlace externo y enlace local del nodo movil, entonces el paquete es enviado sin una cabecera de ruteo, sólo como dirección destino, la dirección de enlace local del nodo móvil; si el nodo móvil se encuentra en el enlace local el paquete será entregado directamente al nodo, pero si el nodo móvil no se encuentra en el enlace local, entonces el agente local del nodo móvil se encargara de enviar el paquete a través de un túnel hasta la dirección primaria de enlace exterior, con encapsulación IPv6. Después de esto del nodo móvil enviara un paquete con la opción Binding Update hasta el nodo que emitió el paquete, permitiendo con esto que el nodo emisor tenga en memoria la asociación de la dirección de enlace local con la dirección primaria de enlace exterior.

Cuando un nodo desea enviar un paquete a un nodo móvil y éste se encuentra en su enlace local, el paquete es enviado normalmente como si el nodo móvil fuera un nodo fijo.

Capítulo V

DIRECCIONAMIENTO Y SEGURIDAD EN IPv6

La red Internet representa un ejemplo paradigmàtico de las redes de *datagramas* o, tambien llamadas como *no orientadas a conexión.* En ella, cada fragmento (paquete) de informacion es transmitido por la red de manera no fiable y sin mantener vínculo alguno, ni siquiera de orden, con el resto de los paquetes de la unidad de información intercambiada. Para poder efectuar esto, cada paquete contiene en su inicio (o cabecera), la identificación de su origen, junto con la de su destino. Esta caracteristica del protocolo IP presenta la ventaja de facilitar la interconexión de redes, en contraposición con otras tecnologías orientadas a conexión, como X.25 o ATM, basadas en el concepto de circuitos virtuales o canales fiables, que asigna la red para la comunicación entre extremos de manera ordenada e integra.

El protocolo IP gobierna la estructura y la transmisión de los datagramas a través de los nodos de la red, sean éstos sistemas finales (hosts, con un punto de conexión a red) o intermedios (ruteadores, con más de un punto de conexión). Dentro del protocolo están definidas características tales como la fragmentación de la información en paquetes de tamaño adecuado para su transmisión por un medio determinado, el formato de las direcciones de red o las opciones de enrutamiento de los datagramas Quizás el recurso mas valioso que posee la red Internet, es su *espacio de direcciones* o el conjunto de todos los identificadores que pueden ser asignados a los puntos de conexión a red y que está estrechamente ligado a los conceptos administrativos de direccionamiento y enrutamiento.

La asignación de direcciones comenzó a hacerse de manera centralizada por un único centro de registro (NIC), satisfaciendo casi todas las solicitudes, sin necesidades de mayor trámite. Cuando la red empezó a crecer exponencialmente, este modelo de asignación de direcciones originó algunos problemas, tales como mal aprovechamiento de direcciones, peligro de agotamiento de las direcciones de clase B y síntomas de saturación en los ruteadores de los backbones, entre otros

Debido al carácter puramente académico de Internet en sus comienzos, los asuntos relativos a la seguridad fueron relegados a estudios posteriores, hasta que se realizaron los primeros ataques globales y empezó a realizarse un notable esfuerzo para incorporar mecanismos de seguridad a las aplicaciones ya existentes. Sin embargo, el problema de la seguridad en el nivel de red siguió sin ser tenido en cuenta y comenzaron a producirse una serie de ataques cada vez más complejos, basados en la suplantación de la identidad de maquinas conectadas a la red, dando la posibilidad de violar un acceso prohibido y de desviar la información a intrusos. Como respuesta, surgieron mecanismos de barrera, como los firewalls, aunque los protocolos continuaron sin incorporar medidas específicas de seguridad.

Todos los problemas mencionados anteriormente, han sido tomados en cuenta por la comunidad Internet, en el desarrollo del nuevo protocolo IPv6. Las direcciones que maneja son más amplias, además de que incluye una mejora en el enrutamiento; es decir, no sólo aumenta el espacio de direcciones, sino que se estructura se hace más adecuada para un enrutamiento óptimo, mediante una jerarquía de prefijos que implica distintas condiciones de asignación.

5.1 DIRECCIONAMIENTO EN IPv6

Con IPv6, las direcciones que se manejan son ampliadas a 128 bits y las facilidades de enrutamiento son mejoradas. El espacio de direcciones no sólo aumenta, sino que su estructura se hace más adecuada para un enrutamiento óptimo, mediante una jerarquía de prefijos

Otras características del nuevo esquema de direccionamiento son las siguientes:

- distintos modelos de asignación, como el basado en el proveedor, el geográfico y el local.
- Un nuevo tipo de dirección: *anycast*
- Las direcciones de *broadcast* no existen en IPv6. Son casos particulares de direcciones *multicast*, las cuales incorporan un campo de ámbito, en sustitución del parámetro TTL usado en la actualidad para determinar el rango de actuación de una sesión multicast. .
- Autoconfiguración. Uno de los aspectos fundamentales de IPv6 es la incorporación de mecanismos que permitan la conexión automática (modelo *plug and play*) de equipos a la red. pueden construirse direcciones globales usando como parte local la dirección MAC de un equipo y obteniendo el prefijo a través de un servidor de la red.

Las cabeceras de los paquetes han sido simplificadas en IPv6 eliminando los campos no utilizados y añadiendo el concepto de cabeceras de extensión Estas permiten seleccionar facilidades especiales de encaminamiento, fragmentación y seguridad, así como el manejo de opciones, que han sido eliminadas de la cabecera IPv4. Cada cabecera incluye un campo que define el tipo de cabecera que le sigue a continuación, hasta llegar a la de transporte, con lo que se agiliza el proceso de los paquetes.

5.1.1 Representación de las Direcciones

Existen tres formas convencionales de representar las direcciones IPv6 como cadenas de texto la forma preferente (la dirección IPv6 completa en valores hexadecimales), la forma comprimida (con la sustitución de las cadenas de ceros) y la forma mixta (conveniente para ambientes mixtos con nodos IPv4 e IPv6).

La forma preferente es **x:x:x:x:x:x:x:x**:, donde "x" se refiere a los valores hexadecimales de las 8 partes de 16 bits de la dirección. Algunos ejemplos son los siguientes:

FEDC:2B5F:709C:216:AEBD 97:3154:3D12
1030.2D9C:0.0:0:500:200C.3B4

En esta forma de representación de las direcciones, no es necesario escribir todos los ceros en un campo individual, pero debe haber al menos un número en cada campo.

Debido al método de localización de las direcciones IPv6, serán comunes las direcciones que contengan largas cadenas de ceros. Para hacer más sencilla la escritura

de direcciones con muchos ceros, se puede usar el método comprimido. El uso de un par de dos puntos "::", indica que existen grupos múltiples de 16 bits con ceros únicamente. Este doble símbolo "::" puede aparecer solamente una vez en una dirección. Como ejemplos, tenemos los siguientes

FF08 0:0.0:0:0:209B 61	FF08::209B 61
1030:2C9D·0.0.0.500 200B:3B4	1030.2C9D .500:200B:3B4
0 0:0.0:0:0.0:1	1

En ambientes mixtos de nodos IPv4 e IPv6, se puede emplear una forma más conveniente de representar las direcciones: la forma mixta. En esta forma, los direcciones de los nodos son expresadas como x·x·x:x:x:x:d.d.d.d, donde "x" es el valor hexadecimal de las seis partes de 16 bits de mayor orden de la dirección, y la "d" se refiere al valor decimal de las cuatros partes de 8 bits de menor orden de la dirección en la representación estándar de IPv4. Algunos ejemplos son los siguientes:

0.0:0.0.0:0:193 136 239 163
0·0.0·0:0:FFFF 129.234.71.52

o en su forma comprimida.

::193.136.239 163
::FFFF:129.234.71.52

Cuando las direcciones IPv6 son expresadas en texto, es común indicarlas tanto por la dirección, como por la longitud del prefijo:

dirección IPv6 / longitud del prefijo

donde la dirección IPv6 es expresada en una de las formas mencionadas arriba, y la longitud del prefijo es un valor decimal que especifica el número de bits más significativos de la dirección, comprimiendo el prefijo Por ejemplo

12BC:0000·0000:CD30:0000:0000:0000:0000 / 60

lo cual indica que el prefijo es de 60 bits y es el siguiente (en hexadecimal):

12BC00000000CD3

5.1.2 Dirección Unicast

En una dirección unicast, existe un identificador para una interfaz simple. Un paquete enviado a una dirección unicast, es entregado en la interfaz identificada por dicha dirección, como se muestra en la figura 5.1.

Actualmente, existen varios modelos de asignación de direcciones unicast en IPv6:

- basado en el proveedor global (direcciones unicast de equipos que se conectan a través de un proveedor)
- geográfico (direcciones unicast para puntos neutros de interconexión)
- local (sin conexión a la red global)
- direcciones compatibles con IPv4

- direcciones de Punto de Acceso al Servicio de Red (NSAP, *Network Service Access Point*), usadas en el esquema de direccionamiento de OSI
- direcciones jerárquicas del Protocolo de Intercambio de Paquetes (IPX, *Internetwork Packet Exchange Protocol*)

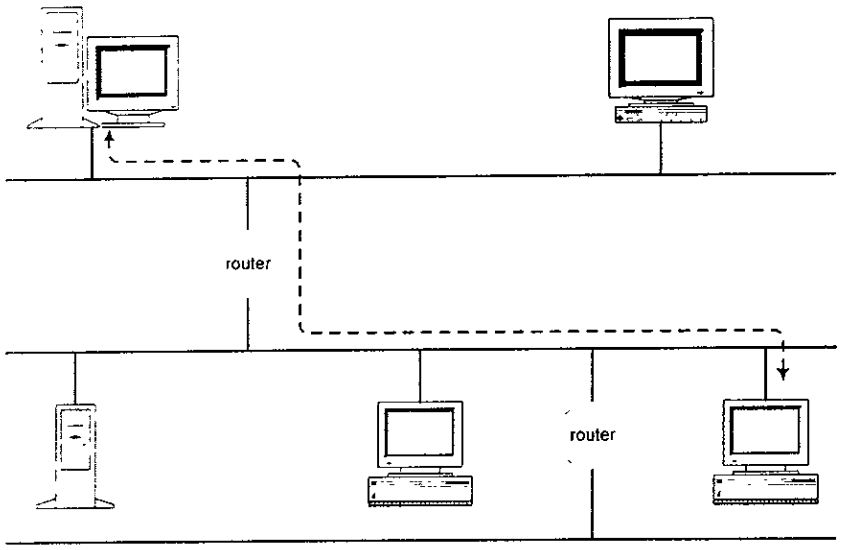


Figura 5.1 - *Dirección Unicast*

Los nodos IPv6 pueden tener mucho o poco conocimiento de la estructura interna de una dirección IPv6, dependiendo del papel que juega el nodo (host vs. ruteador).

Como mínimo, un nodo puede considerar que la dirección unicast (incluyendo la suya propia) no tiene una estructura interna, como se muestra en la figura 5.2.

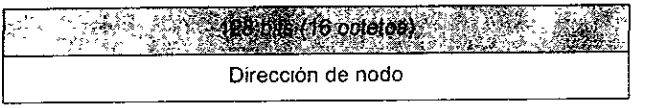


Figura 5.2 - *Dirección unicast sin estructura jerárquica*

Algunos nodos más complejos (por ejemplo, ruteadores), pueden estar enterados de las fronteras jerárquicas de una dirección unicast. Las fronteras conocidas difieren de nodo a nodo, dependiendo del papel del nodo y de su posición en la jerarquía estructural.

La figura 5.3 ilustra el caso en el que una organización o sitio requiere de varias capas de jerarquía interna.

s bits		n bits	
Prefijo	ID de área	ID de subred	ID de interfaz

Figura 5.3 - *Dirección unicast con varias capas de jerarquía interna*

La dirección 0:0:0:0.0:0:0:0 es llamada dirección no especificada y nunca debe ser asignada a algún nodo, ya que indica la ausencia de una dirección. Un ejemplo de su uso es en el campo de Dirección Origen de cualquier datagrama IPv6 enviado por un host, antes de que aprenda su propia dirección.

La dirección 0:0:0:0:0:0:0:1 es llamada dirección de lazo cerrado. Puede ser empleada por un nodo para enviar un datagrama IPv6 a él mismo. Nunca debe ser asignada a alguna interfaz y no debe ser usada como una dirección origen de algún datagrama IPv6 que sea enviado a la red.

Direcciones basadas en el proveedor

Las direcciones unicast basadas en el proveedor son utilizadas para comunicaciones globales. Los primeros tres bits identifican la dirección como una dirección basada en el proveedor Los campos siguientes son asignados respectivamente a las autoridades de registro, quienes asignan las porciones del espacio de direcciones a los proveedores de servicio, los cuales a su vez, asignan un espacio de direcciones a los suscriptores. La parte de la dirección que le corresponde al suscriptor interno (intra-suscriptor) está organizada de acuerdo con la topología local de internet del suscriptor. El formato de este tipo de direcciones se ilustra en la figura 5.4.

3 bits	n bits	m bits	p bits	125-n-m-p bits
010	ID de registro	ID de proveedor	ID de suscriptor	Intra-suscriptor

Figura 5.4 - *Dirección unicast basada en el proveedor*

Direcciones de uso local

Una dirección de uso local es una dirección unicast que tiene únicamente un propósito de ruteo local. Existen dos tipos de direcciones de uso local: de enlace local y de sitio local

Las direcciones de enlace local están diseñadas para ser usadas para el direccionamiento en un enlace simple, para propósitos tales como la autoconfiguración de direcciones, el descubrimiento de vecino, o cuando no hay ruteadores presentes.

Las direcciones de sitio local pueden ser empleadas para sitios u organizaciones que no se encuentran conectadas a Internet.

La parte de orden más bajo de los dos tipos de direcciones de uso local contienen un campo de identificación de interfaz, que debe ser único en el dominio en el cual se esté usando

Direcciones compatibles con IPv4

Han sido definidas dos direcciones de transición para redes de transición IPv4/IPv6. La primera de estas direcciones es llamada dirección IPv6 compatible con IPv4 (figura 5.5) y es usada cuando dos dispositivos IPv6 (ya sean hosts o ruteadores) se necesitan comunicar a través de una infraestructura de ruteo de IPv4. Los dispositivos en el borde de una infraestructura IPv4, deben emplear esta dirección unicast especial, que transporta una dirección IPv4 en los 32 bits menos significativos. Este proceso es llamado *tunelado automático*. Hay que hacer notar que el prefijo es de 96 bits, todos igual a cero.

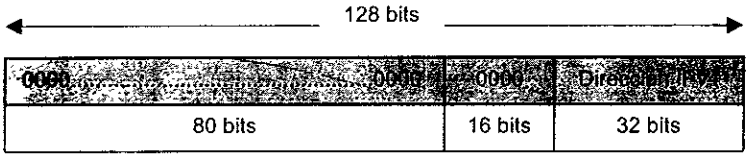


Figura 5.5 - *Dirección IPv6 compatible con IPv4*

El segundo tipo de dirección de transición es llamado dirección IPv6 mapeada en IPv4 (figura 5.6). Esta dirección es usada por nodos solo-IPv4 que no soportan la estructura IPv6. Por ejemplo, un host IPv6 debe emplear una dirección IPv6 mapeada en IPv4 para comunicarse con otro host que sólo soporte IPv4. El prefijo es de 80 bits, todos igual a cero, seguidos de 16 bits igual a uno.

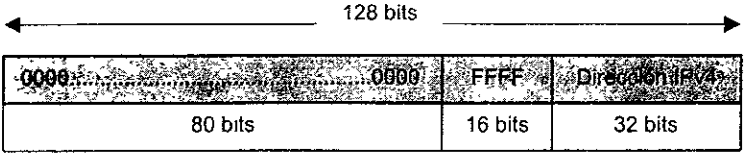


Figura 5.6 - *Dirección IPv6 mapeada en IPv4*

Direcciones unicast que soportan la arquitectura OSI

Muchas redes incorporan en su direccionamiento y arquitectura de ruteo, elementos derivados de los protocolos de Interconexión de Sistemas Abiertos (OSI). Un ejemplo es el Protocolo de Red sin conexión (ISO 8473) y su esquema de direccionamiento, basado en la direcciones de Punto de Acceso al Servicio de Red (NSAP). Otros ejemplos son los

protocolos de ruteo Sistema Final a Sistema Intermedio (ES-IS, *End System to Intermediate System*), o Sistema Intermedio a Sistema Intermedio (IS-IS). Debido a que las direcciones NSAP (llamadas NSAPA) tienen una longitus de 20 octetos, se deben proporcionar mecanismos para adaptar este formato a la estructura de direccionamiento de 16 octetos de IPv6. Las direcciones que soportan NSAPA tienen un prefijo de siete bits, igual a 0000001, como se puede observar en la figura 5.7

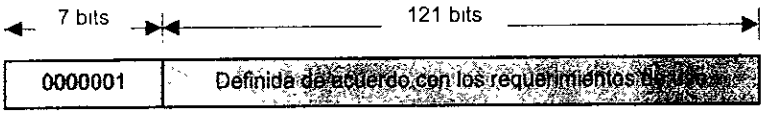


Figura 5.7 - *Dirección NSAP*

Direcciones IPX

Las direcciones de Intercambio de Paquetes (IPX, *Internetwork Packet Exchange*) deben ser mapeadas en direcciones IPv6, con un formato que empiece con el prefijo de siete bits 0000010, como se muestra en la figura 5.8. el balance de la dirección se encuentra todavía en estudio.

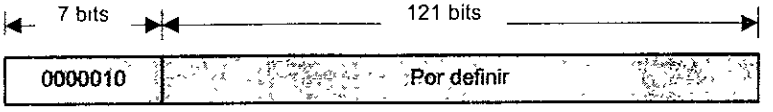


Figura 5.8 - *Dirección IPX*

5.1.3 Dirección Anycast

Una dirección anycast, es una dirección que está asignada a múltiples interfaces, típicamente en nodos diferentes. Un paquete enviado a una dirección anycast es dirigido a la interfaz más cercana a esa dirección, de acuerdo con las mediciones de distancia de los protocolos de ruteo, como se puede observar en la figura 5 9

Uno de los usos esperados de una dirección anycast es identificar un grupo de ruteadores pertenecientes a un proveedor de servicios de Internet. Algunos otros posibles usos son la identificación de un conjunto de ruteadores unidos a una subred específica o de un grupo de ruteadores que proveen acceso a un dominio de ruteo particular.

Las direcciones anycast están derivadas del espacio de direcciones unicast, usando cualquiera de los formatos definidos para este tipo de direcciones. Así, las direcciones anycast no son distinguibles en lo que a sintaxis se refiere, de las direcciones unicast. Cuando una dirección unicast es asignada a más de una interfaz, se convierte en una

dirección anycast y los nodos a los cuales es asignada esta dirección deben ser configurados explícitamente para saber que se trata de una dirección anycast.

Las siguientes restricciones son impuestas a las direcciones anycast:

- no deben ser usadas como la dirección origen de un paquete IPv6
- no deben ser asignadas a un host IPv6; esto es, sólo pueden ser asignadas a un ruteador IPv6

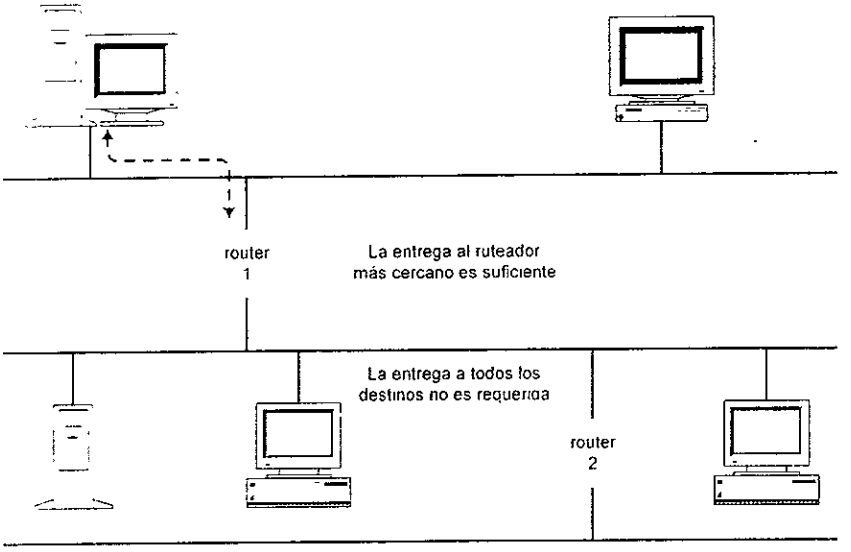


Figura 5.9 - *Dirección Anycast*

5.1.4 Dirección Multicast

Una dirección multicast es un identificador para un conjunto de interfaces, típicamente pertenecientes a nodos diferentes. Un paquete enviado a una dirección multicast es entregado en todas las interfaces identificadas por esa dirección, como se muestra en la figura 5.10.

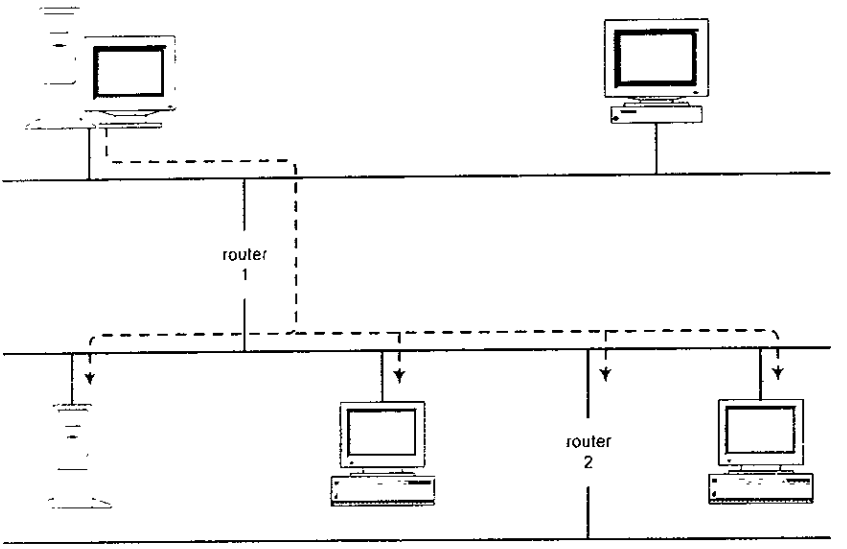


Figura 5.10 - *Dirección Multicast*

Una dirección multicast IPv6 tiene el formato que se muestra en la figura 5.11.

8 bits	4 bits	4 bits	112 bits
11111111	flgs	scop	ID de grupo

Figura 5.11 *Formato de una dirección multicast*

Los campos señalados en esta figura tienen los siguientes significados:

- los bits 11111111 en el inicio de la dirección, la identifican como una dirección multicast
- *flgs* es un conjunto de 4 banderas o identificadores (flags). Los tres indicadores de mayor orden están reservados y deben ser puestos en cero. El valor 0 para el indicador de menor orden, indica una dirección multicast permanentemente asignada ("bien conocida"), asignada por la autoridad global de Internet. El valor de 1 para el indicador de menor orden indica una dirección multicast que no está permanentemente asignada ("transitoria").
- *scop* es un número de 4 bits usado para limitar el alcance de un grupo multicast. Los valores posibles de alcance se muestran en la tabla 5.1.

0	Reservado
1	Nodo local
2	Enlace local
3, 4	(no asignado)
5	Sitio local
6, 7	(no asignado)
8	Organización local
9 - D	(no asignado)
E	Alcance global
F	Reservado

Tabla 5 1 - *Valores de alcance de un grupo multicast*

- *ID de grupo* identifica el grupo multicast

Las direcciones multicast no deben ser usadas como direcciones origen en datagramas IPv6 o aparecer en ninguna cabecera de ruteo

Direcciones Multicast Predefinidas

En la tabla 5 2 se presentan algunas direcciones multicast permanentemente asignadas ("bien conocidas"), que están reservadas o predefinidas

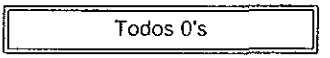
FF00:0:0:0:0:0:0	Reservada
FF01:0:0:0:0:0:0	Reservada
FF02:0:0:0:0:0:0	Reservada
. ..	....
FF0F:0:0:0:0:0:0	Reservada
FF01:0:0:0:0:0:1	Todos los nodos (nodos locales)
FF02 0:0:0:0:0:1	Todos los nodos (enlace local)
FF01:0:0:0:0:0:2	Todos los ruteadores (nodos locales)
FF02:0:0:0:0:0:2	Todos los ruteadores (enlace local)

Tabla 5.2 - *Direcciones multicast predefinidas*

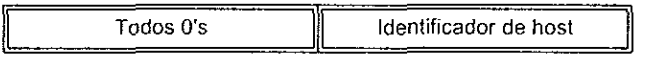
Las direcciones reservadas nunca deben ser asignadas a ningún grupo multicast. Las demas direcciones identifican un grupo de todos los nodos IPv6 dentro del alcance 1 (nodo local) o 2 (enlace local). Por ejemplo, la dirección FF02:0:0:0:0:0:1 (ó FF02::1) significa "todos los nodos en este enlace".

5.1.5 Direcciones Especiales

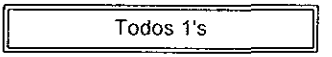
Existen algunas combinaciones de ceros y unos que no se asignan como dirección IP, sino que tienen asociado un significado especial. Las diferentes combinaciones son las que se indican a continuación:



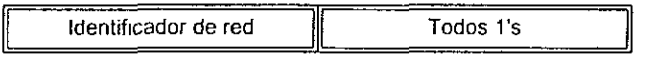
Identifica al propio host



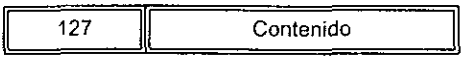
Identifica al host en su red



Multidifusión limitada en la propia red



Multidifusión a todos los hosts de la red indicada



Bucle local

Los dos primeros casos sólo pueden ser usados al arrancar el sistema (por ejemplo, en máquinas sin unidad de almacenamiento fijo) y nunca se usan como una dirección de destino válida. En cualquier caso, sólo se usan de forma temporal, mientras el host *aprende* su dirección IP.

El tercer caso es la denominada dirección multicast de red local o dirección de multidifusión limitada, que permite difundir un mensaje a toda la red local independientemente de su dirección IP asignada. Un host puede utilizar esta dirección como parte de un procedimiento de comienzo antes de conocer su dirección IP o la dirección IP de su red.

La dirección multicast dirigida a una red permite enviar un mensaje a todas las estaciones situadas en una red determinada. Es una herramienta muy potente, ya que

permite enviar un solo paquete que será difundido en toda la red Esta dirección se usa de forma restringida, ya que supone una gran carga de trabajo en redes grandes

La dirección de bucle local está diseñada para pruebas y comunicación entre procesos en la máquina local Si un programa envía un mensaje a esta dirección, el módulo internet le devolverá los datos sin enviar nada a la red. De hecho, nunca debe haber en la red un paquete de este tipo, ya que no es una dirección de red válida.

5.1.6 Reglas para la Asignación de Direcciones

En este apartado veremos las reglas que se han propuesto para facilitar el direccionamiento dentro de Internet y proporcionar un ruteo escalable de las direcciones basadas en el proveedor global (direcciones de equipos que se conectan a través de un proveedor) Para ello, se han creado los Identificadores de Acceso al Nivel Superior (TLA ID, *Top-Level Aggregation Identifiers*), los Identificadores de Acceso al Nivel Próximo (NLA ID, *Next-Level Aggregation Identifiers*) y los Identificadores de Acceso al Nivel Local (SLA ID *Site-Level Aggregation Identifiers*).

El formato de las direcciones basadas en el proveedor global está diseñado para soportar tanto el acceso actual (basado en el proveedor), como un nuevo tipo de acceso basado en la conmutación. Esta combinación permite un ruteo eficiente para sitios que están conectados directamente al proveedor y para sitios conectados a un conmutador .

Debido a que este formato de direcciones está diseñado para soportar enrutamientos basados en la conmutación (además del acceso actual basado en el proveedor), no es dependiente de la conmutación de todas las propiedades de acceso de su ruta. Esto proporciona un ruteo eficiente con tan solo el acceso basado en el proveedor.

El formato de una dirección de acceso global es como el que se puede ver en la figura 5 12, donde se muestran los identificadores de acceso:

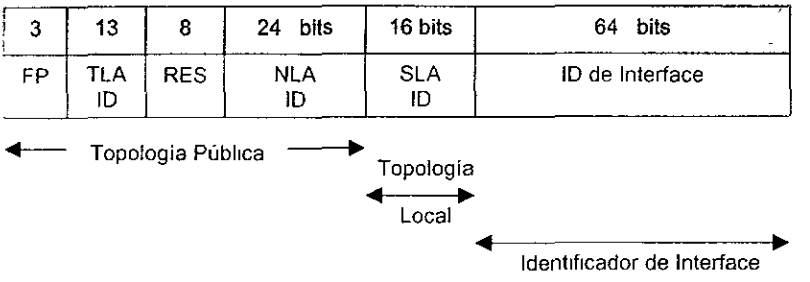


Figura 5.12 - *Formato de una dirección de acceso global*

Donde

FP	Prefijo de Formato
TLA ID	Identificador de Acceso al Nivel Superior
RES	Reservado para uso futuro
NLA ID	Identificador de Acceso al Nivel Próximo
SLA ID	Identificador de Acceso al Nivel Local
ID de Interface	Identificador de Interface

El tipo especifico de una dirección IPv6 es indicado por los primeros tres bits en la dirección. El campo que comprende estos bits es llamado *Prefijo de Formato* (FP).

El tamaño del *Identificador de Acceso al Nivel Superior* es de 13 bits, lo cual permite tener hasta 8192 TLA ID's. Este tamaño fue escogido para asegurar que la tabla de ruteo en los ruteadores de nivel superior de Internet, su mantuvieran dentro de los límites. El margen es importante porque los ruteadores pueden tener también un número de prefijos mas largos (más específicos) para optimizar las rutas internas de una TLA y entre TLAs.

El punto más importante de TLA no es únicamente el tamaño de la tabla de ruteo, sino tambien la complejidad de la topología que determina el número de veces que el ruteador debe buscar un ruta mientras calcula el direccionamiento. En IPv4 es común ver un prefijo anunciado cincuenta veces en diferentes rutas. Además, la complejidad de la topología de Internet se incrementará dentro de poco tiempo.

En el futuro, la tecnología de ruteo se incrementará para soportar un número más grande de rutas de nivel superior en las tablas de ruteo, por lo cual existirán dos alternativas para incrementar el número de identificadores TLA. La primera es expandir el campo TLA ID utilizando el campo Reservado (RES). Esto incrementará el número de TLA ID hasta aproximadamente dos millones. La segunda opción es colocar otro prefijo de formato (FP) para usarlo con este formato de direccionamiento.

El tamaño del campo *Reservado* (RES) es de 8 bits. Este tamaño fue escogido para permitir un crecimiento significativo ya sea del campo TLA ID o bien, del campo NLA ID.

El tamaño del campo *Identificador de Acceso al Nivel Próximo* es de 24 bits. Esto permite tener aproximadamente 16 millones de NLA ID's si es usado de manera simple. Usado jerárquicamente, permite tener una complejidad equivalente al espacio de direcciones de IPv4. Si en el futuro es necesario incrementar la complejidad del NLA ID, puede efectuarse extendiendo el campo NLA ID dentro del campo Reservado.

El tamaño del campo *Identificador de Acceso al Nivel Local* es de 16 bits. Este soporta 65,535 subredes individuales por sitio. El objetivo del tamaño de este campo es que sea suficiente para todas las organizaciones más grandes. El campo SLA ID es de un tamaño fijo para forzar a todos los prefijos identificadores de un sitio en particular, a ser de la misma longitud, lo cual facilita el movimiento de sitios dentro de la topología de Internet.

El campo *ID de Interface* es de 64 bits de longitud. Este tamaño fue escogido para cumplir con los requerimientos especificados para los identificadores de interfaces.

Los motivos técnicos específicos para las reglas propuestas para la asignación de TLA ID y NLA ID descritas en este apartado, son los siguientes:

- Limitar el número de prefijos de nivel superior en Internet a un tamaño manejable Es importante asegurar que las tablas de ruteo en los ruteadores de nivel superior de Internet, se mantengan dentro de los límites, con un margen razonable, establecido por la tecnología actual de direccionamiento.
- Deben estar disponibles públicamente todas las asignaciones de las organizaciones que tienen TLA ID's asignados Esto es necesario para que los registros tengan la información exacta de las de las asignaciones y para facilitar la solución de problemas de Internet.
- Disposición de prefijos que señalen el formato de direccionamiento. Específicamente la disposición de prefijos no mayores que 48 bits, para no alterar los campos del Identificador de Acceso al Nivel Local (SLA) y del Identificador de Interface Esto es para facilitar el movimiento de sitios dentro de la topología de Internet

Etapas para la asignación de TLA

Las asignaciones de TLA son hechas en dos etapas. La primera de ellas es colocar un Sub-TLA ID Este identificador de Sub-TLA es asignado fuera del TLA ID como se muestra en la figura 5.13, donde se puede ver que se emplea el campo Reservado y parte del campo NLA ID para crear el campo de Sub-TLA. sin afectar el campo TLA ID.

3 bits	13 bits	13 bits	19 bits
FP	TLA ID	Sub-TLA ID	NLA ID

Figura 5.13 - *Asignación del campo Sub-TLA*

Donde

FP = Prefijo de Formato
Este es el Prefijo de Formato usado para identificar el tipo de direcciones.

TLA ID = Identificador de Acceso al Nivel Superior
Este es el TLA ID asignado por la Autoridad de Números Asignados de Internet (IANA; *Internet Assigned Numbers Authority*) para la colocación de Sub-TLA.

Sub-TLA ID = Identificador de Sub-TLA
Este campo es usado por los registros para la colocación inicial de las organizaciones proveedoras de servicios de Internet. La IANA asigna bloques pequeños (por ejemplo, un par de cientos) de Sub-TLA ID's a los registros. Cuando estos registros hayan asignados todos los identificadores a las organizaciones, pueden hacer el requerimiento a la IANA para que les proporcione otros bloques de identificadores.
IANA puede asignar también Sub-TLA ID's directamente a las organizaciones. Esto incluye la asignación temporal de TLA para prueba y uso experimental de actividades desarrolladas dentro del 6bone o para accesos diferentes (por ejemplo, conmutados).

NLA ID = Identificador de Acceso de Nivel Próximo

Este identificador es usado por organizaciones asignadas a un TLA ID para crear un direccionamiento jerárquico y para la identificación de sitios Las organizaciones pueden asignar la parte superior del NLA ID de tal manera que se establezca el direccionamiento jerárquico más apropiado a sus redes

Como parte de la asignación del TLA ID a una organización, la IANA puede inicialmente asignar solo una fracción del espacio del NLA ID Cuando la organización haya asignado más del 90% de este espacio del NLA ID, puede pedir. un espacio adicional para su identificador

5.2 SEGURIDAD EN IPV6

La seguridad en IPv6 es proporcionada por el protocolo IPsec (IPsec, Internet Protocol Secure) En IPv6 el protocolo IPsec se encuentra situado dentro de la implementación de IPv6, por lo que IPsec se encuentra dentro de los protocolos nativos de IPv6, lo que no sucede en IPv4 ya que IPsec es protocolo adicional a la implementación de IPv4. El conjunto de servicios de seguridad ofrecidos por IPsec, incluyen el control de sucesos, integridad orientada a conexión, autenticación de la información, protección contra reenvío de paquetes, confidencialidad (encripción de la información), y confidencialidad limitada en el flujo de datos. Estos servicios son proporcionados a través de dos cabeceras de seguridad que son: Cabecera de Autenticación y la Cabecera de Encapsulación de Seguridad de la Información (ESP, Encapsulating Security Payload)

La cabecera de autenticación, provee integridad orientada a conexión, autenticación de la información, y un servicio opcional de anti-reenvio.

La cabecera ESP puede proveer confidencialidad por medio del uso de la encripción de los paquetes.

Ambas cabeceras son vehículos para el control de acceso, ya que se basan en la distribución de llaves de Criptografía.

5.2.1 Asociaciones de seguridad

La asociación de seguridad es una simple conexión entre dos nodos, que ofrece los servicios de seguridad para todo el trafico que sea transmitido dentro de la asociación de seguridad. Los servicios de seguridad ofrecidos en una asociación de seguridad son: la cabecera de autenticación o de la cabecera ESP, pero no es posible que en una misma asociación de seguridad se ofrezcan ambos servicios. Si se requiere que ambas cabeceras de seguridad sean aplicadas para un mismo flujo de trafico, entonces es necesario el uso de dos o más asociaciones de seguridad para la protección del trafico de la información. En la asociación de seguridad se tendrá la información referida a los parámetros de seguridad que se requieran, como el tipo de
algoritmo de encripción o de autenticación que se debe de usar.

Una asociación de seguridad es identificada por el valor del campo Security Parameter Index (SPI), la dirección destino y un identificador del protocolo de seguridad (Cabecera de Autenticación, o de la cabecera Encapsulating Security Payload).

Existen dos tipos de asociaciones de seguridad: modo transporte y modo túnel.

La asociación de seguridad de modo transporte solo puede existir entre dos host. Para el modo transporte, el protocolo de seguridad (La cabecera de autenticación o la cabecera ESP) debe aparecer después de la cabecera base IPv6 y cualquier otra cabecera complementaria opcional y antes de las cabeceras de los protocolos de capas superiores, tales como TCP o UDP. Cuando la cabecera de autenticación se usa en modo transporte, la seguridad sólo se provee para una porción de la cabecera IPv6 y para los protocolos de capas superiores; cuando la cabecera ESP es empleada en modo transporte la seguridad sólo se provee para las capas superiores a IP

La asociación de seguridad en modo túnel la puede ser de host a host, de host a un gateway de seguridad, de un gateway de seguridad a un host o de un gateway de seguridad a un gateway de seguridad. Siempre que el extremo de un túnel sea un gateway de seguridad, debe utilizarse el modo túnel en la respectiva asociación de seguridad

Para el modo túnel el paquete contiene dos cabeceras de IPv6, una cabecera IPv6 exterior que especifica el destino donde se realizara el procesamiento de IPsec, y una cabecera IPv6 interior que especifica el destino final del paquete Cuando la cabecera de autenticación es empleada en modo túnel, la seguridad proveída es solo para una porción de la cabecera IPv6 exterior y para toda la parte restante del paquete tunelado, que contiene la cabecera IP interior, cabeceras suplementarias y las capas superiores. Cuando la cabecera ESP es empleada en modo túnel, la seguridad proveída es solo para la cabecera interior IP y capas superiores del paquete tunelado

El host debe soportar el modo transporte y el modo túnel en las asociaciones de seguridad, y el gateway de seguridad es requerido que soporte solo el modo túnel, con la excepción de cuando el gateway de seguridad este actuando como host, por ejemplo cuando reciba comandos SNMP (SNMP, Simple Network Management Protocol), el gateway de seguridad podrá usar el modo transporte.

Combinación de Asociaciones de Seguridad

Como se dijo antes una sola asociación de seguridad sólo puede ofrecer protección relacionada a un solo tipo de protocolo de seguridad, ya sea el de la cabecera de autenticación o el de la cabecera ESP, pero no ambos. Algunas veces las políticas de seguridad pueden requerir de una combinación de servicios de seguridad para mayor proteccion de un flujo de trafico, donde con una sola asociación de seguridad es imposible que pueda llevarse a cabo, por lo es necesario emplear mas de una asociación de seguridad para la implementación del requerimiento de las políticas de seguridad.

El Conjunto de asociaciones de seguridad pueden ser combinadas en dos formas: Transporte adyacente y tunelado repetido.

- La forma de transporte adyacente se refiere a aplicar mas de un protocolo de seguridad (Cabecera de autenticación o cabecera ESP) en modo transporte, para el

mismo datagrama IPv6, dejando fuera el modo túnel, este tipo de combinación de asociaciones de seguridad se muestra en la fig. 5.14.

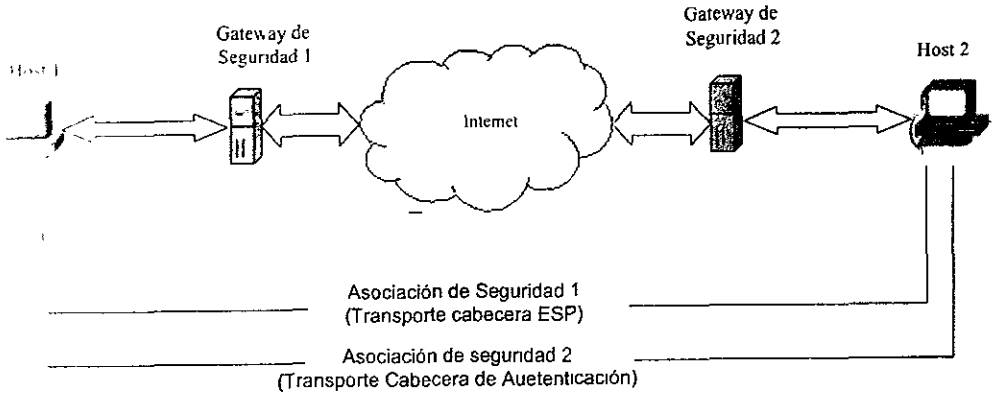


Fig 5 14 - *Combinaciones de Asociaciones de Seguridad en Modo Transporte Adyacente*

- La forma de tunelado repetido se refiere a la aplicación de múltiples capas de protocolos de seguridad, que es realizado a través del tunelado IP. Esto permite distintos niveles de seguridad, ya que cada túnel puede originarse o terminar en con una distinta cabecera de seguridad de IPv6. Existen 3 diferentes casos de tunelado repetido
- Los extremos de las asociaciones de seguridad en modo túnel son las mismas, un túnel puede utilizar la cabecera de autenticación mientras que el otro estará usando la cabecera ESP. Este tipo de combinación se ejemplifica en la figura 5.15.

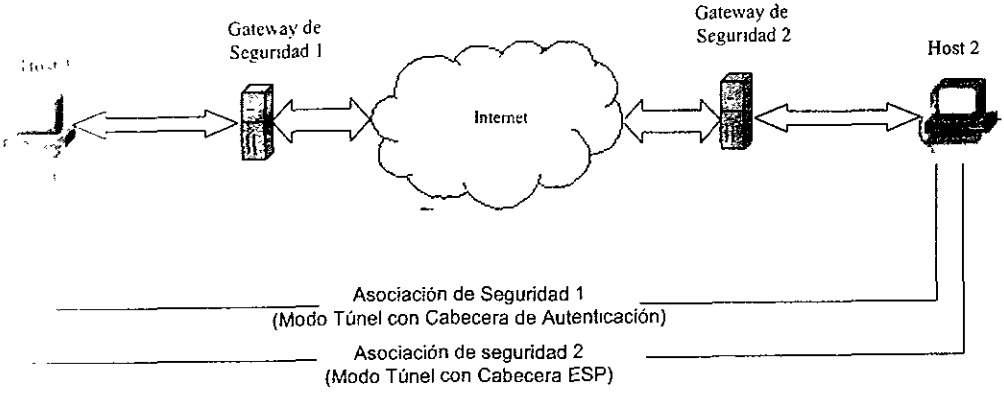


Fig.5.15 - *Tunelado Repetido cuando los extremos del túnel son los mismos*

- Uno de los nodos extremos de los túneles es el mismo en ambas asociaciones de seguridad, ya sea un gateway de seguridad o un host, por ejemplo se puede observar en la figura 5 16 que el extremo en común de los túneles es el host. Un túnel puede estar usando la cabecera de autenticación mientras que el otro usara la cabecera ESP

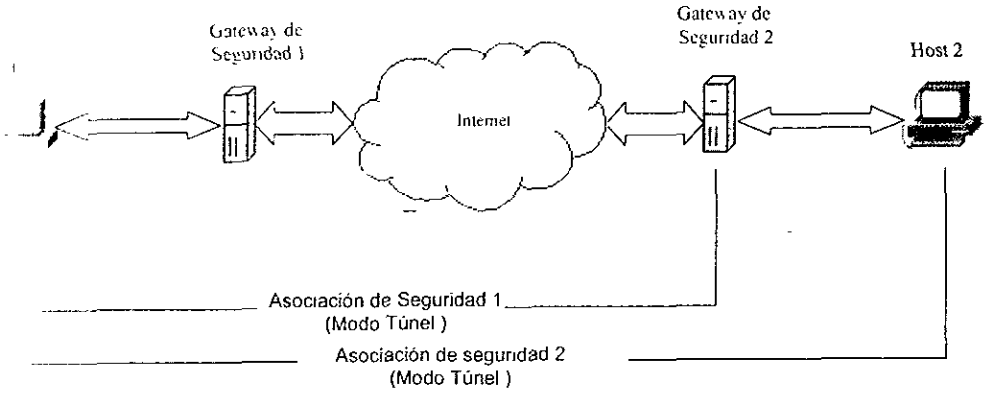


Fig 5 16 - *Tunelado Repetido cuando uno de los extremos de los túneles es el mismo*

- Otra de las formas es cuando ninguno de los extremos de los túneles es el mismo, como se muestra en la figura 5 17 Un túnel puede estar usando la cabecera de autenticación, mientras que el otro usara la cabecera ESP.

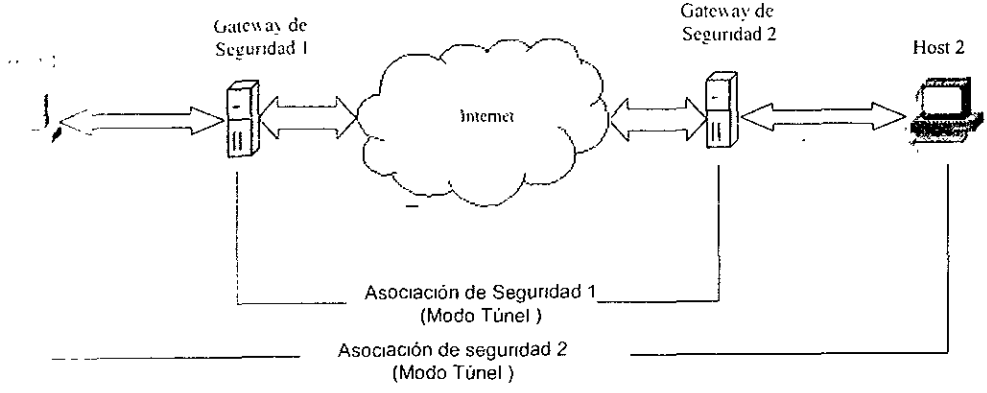


Fig 5.17 - *Tunelado repetido cuando los extremos de los túneles son diferentes*

Estas combinaciones antes citadas de cómo es que se puede dar un conjunto de asociaciones de seguridad, generan cuatro formas básicas de posibles combinaciones

que se pueden dar al implementar la seguridad con IPv6, estas cuatro posibles combinaciones se dan en base al análisis anterior.

- La primera combinación provee seguridad de extremo a extremo entre dos host, Este tipo de combinación se da cuando se necesitan que los paquetes crucen a través de Internet o una intranet, esto se puede observar en la figura 5.18. En esta configuración se puede seleccionar el modo túnel o el modo transporte.

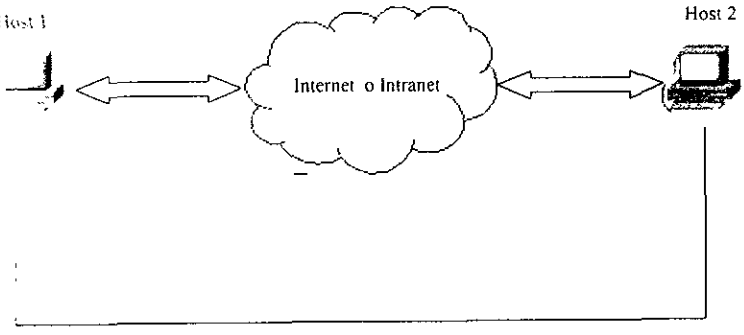


Fig.5.18 - *Asociaciones de seguridad entre dos Host a Host*

- El segundo caso es cuando se desea interconectar dos redes en forma segura, a traves de internet. En este caso solo se debe usar el modo túnel, este ejemplo se puede observar en la figura 5.19

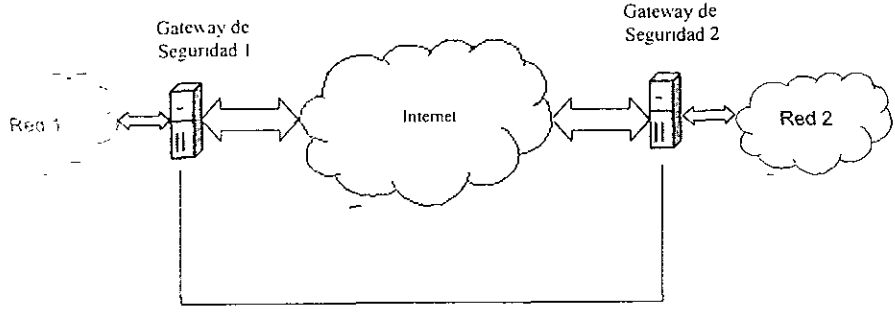


Fig. 5.19 - *Conexión de dos redes por medio de dos gateway de seguridad*

- El tercer caso se combina los dos casos anteriores, esto permite mas flexibilidad en la implementación ya que los host que tengan implementada seguridad en IPv6, podrá usar una o un conjunto de asociaciones de seguridad en modo transporte para la

Todo el trafico enviado entre dos nodos, puede ser enviado por una sola asociación de seguridad o por varias, esto dependerá de la aplicación que se este utilizando, ya que algunas aplicaciones exigirán distintas características de seguridad, con respecto a otras.

Bases de datos de las Asociaciones de seguridad

Las bases de datos de las asociaciones de seguridad, sirven para guardar la información acerca de los parámetros de las asociaciones de seguridad que se encuentran en un estado activo, asi como también las políticas de seguridad de la red. Algunos de estos parámetros son el modo de la asociación de seguridad que se esta usando que tipo de cabecera se esta usando (cabecera de autenticación o la cabecera ESP), que algoritmo de autenticación o de encriptación se esta usando

Existen dos tipos de bases de datos: Base de datos de las políticas de seguridad y la base de datos de las asociaciones de seguridad. Las dos bases de datos especifican las políticas que determinan la disposición de todo el trafico IPv6 que entra y sale de la red. La base de datos de la asociación de seguridad, contiene los parámetros relacionados a cada una de las asociaciones de seguridad que se encuentran activas.

Cada interface de red en la cual se tenga implementado IPsec en IPv6, debe separar las bases de datos (base de datos de políticas de seguridad y base de datos de las asociaciones de seguridad) para el trafico entrante y para el tráfico saliente, esto es para dar un mejor manejo con respecto a la seguridad al trafico entrante y al trafico saliente.

Base de datos de políticas de seguridad

Esta base de datos debe ser consultada durante el procesamiento de todo el trafico (incluyendo trafico no IPsec), ya sea trafico que este entrando o saliendo de la red; en el caso de que no se encuentre ninguna información en la base de datos acerca de cómo debe manipular un cierto tipo de paquete, este debe ser desechado

Existen tres posibles acciones que se pueden ejecutar sobre el datagrama IPv6, al consultar esta base de datos, ya sea para trafico de entrada o trafico de salida, esta son :

- Descartar el datagrama: Se refiere a descartar el datagrama ya que para el nodo, el datagrama IPv6 es de tipo desconocido y no sabe que hacer con el.
- No aplicar IPsec. Se refiere a que el datagrama IPv6 no requiere que se le aplique ningún tipo de seguridad de IPv6.
- Aplicar IPsec: esto se refiere a que se aplique algún tipo de seguridad en el datagrama IPv6 esto dependerá del tipo de aplicación a que este asociada el datagrama y las políticas de seguridad contenidas en la base de datos de políticas de seguridad; también dependerá si se aplica una sola asociación de seguridad o un conjunto de asociaciones de seguridad.

Esta base de datos debe ser configurada de manera tal para el que trafico que no este relacionado con el protocolo IPsec de IPv6, lo deje pasar para que sea procesado normalmente En esta base de datos se contiene la información del orden que deben de seguir la asociaciones de seguridad, para cuando se requiere de mas de una asociación de seguridad para un trafico especifico.

Base de datos de la asociación de seguridad

Esta base de datos se definen los parámetros asociados con cada una de las asociaciones de seguridad que se encuentran activas; como lo son la dirección IP destino, el protocolo de seguridad usado (autenticación o encriptación), y el valor contenido en el campo security parameter index (SPI) de la cabecera de autenticacion o de la cabecera ESP segun que cabecera se este usando.

Para todo el trafico que es emitido y requiere de un tipo de asociación de seguridad, primeramente se analiza en la base de datos de políticas de seguridad; posteriormente se ve requiere de algún tipo de seguridad; se busca en la base de datos de las asociaciones de seguridad, si se tiene alguna asociación de seguridad activa que cumpla los requerimientos de trafico emitido, si existe esta asociación de seguridad los paquetes se envian por la asociación de seguridad activa; en el caso de que ninguna asociación de seguridad activa, cumpla los requerimientos de seguridad para el trafico que se desea enviar, entonces se crea una o un conjunto de asociaciones de seguridad, que se indican en la base de datos de las asociaciones de seguridad

Para todo trafico entrante siempre se verifica en la base de datos de las asociaciones de seguridad, a que asociación de seguridad pertenece el trafico que viene entrando de acuerdo a su dirección destino, el protocolo de seguridad que se esta usando y el valor del campo SPI.

5.2.2 Gateway de Seguridad

El gateway de seguridad es un sistema que provee seguridad a una red, también actúa como el nodo intermediario para una comunicación segura entre una red externa y la red local donde se encuentra el gateway de seguridad. Los host's o redes que no pertenezcan a la misma red donde se encuentra el gateway de seguridad, son vistas por este como sistemas intrusos, mientras que los host's que se encuentran dentro de la misma red a la que pertenece el gateway de seguridad, son vistos como host confiables.

El gateway de seguridad establece asociaciones de seguridad para los host, así como también provee servicios de seguridad entre el gateway y destinos externos a su red local.

Un host o un gateway de seguridad, debe tener una interface de administración que permita al usuario o al administrador de la red, configurar la dirección de un gateway de seguridad, al que puedan accesar para realizar la conexión a nivel IPsec., así el host o el gateway de seguridad podrá localizar el gateway de seguridad cuando se requiera. El gateway de seguridad debe soportar los protocolos de la cabecera de Autenticación o la cabecera ESP, o ambas.

5.2.3 Función de la Cabecera de Autenticación

El papel de la cabecera de autenticación, dentro de IPv6 es proveer integridad orientada a conexión, autenticación de datos de origen, y un servicio opcional de anti-reenvios. La cabecera de Autenticación puede ser implementado en dos maneras: modo transporte cuando la comunicación es entre dos host y modo túnel para cuando la

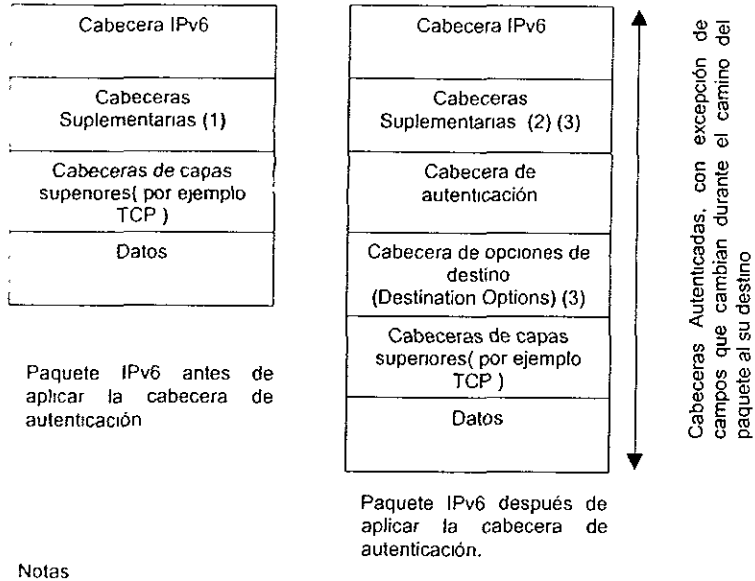
comunicacion es entre un host y un gateway de seguridad, o entre gateway's de seguridad.

La cabecera de autenticación es apropiada para cuando no se requiere confidencialidad, o en el caso de que no sea permitida (quizá por regulaciones gubernamentales)

La cabecera ESP al igual que la cabecera de autenticación, también provee autenticacion, aunque la diferencia es que la cabecera de autenticación usa un algoritmo de autenticacion más poderoso comparado con el algoritmo que usa la cabecera ESP.

En lo que se refiere a la autenticación en una comunicación punto a punto, los algoritmos de autenticación incluyen llaves de mensajes de código de autenticación (HMAC. Message Atutentication Codes), para la creación de estas llaves se basa en algoritmos de encripción simétrico, como lo es Estándar de Encripción de Datos (DES, Data Encryption Standar) o con funciones hash, que son algoritmos para poder crear llaves de seguridad (llaves de mensaje de código); los algoritmos hash que se usan en IPsec para la autenticación son. MD5 (Message Digest Algorithm 5) que usa y SHA-1 (Secure Hash Algorithm), En el punto 5.2.5 se retoma este punto explicándolo mas ampliamente.

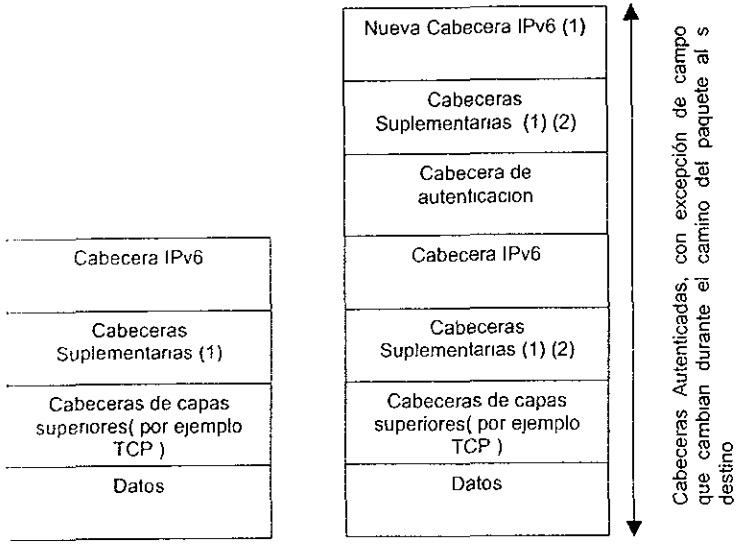
En la figura 5.22 se puede observar como la cabecera de autenticación maneja el paquete con una asociación de seguridad de modo transporte.



Notas
(1) Si se presenta
(2) Cabeceras Hop by Hop, de Opciones de destino (Destination Options), de ruteo, de fragmentacion si se presentan
(3) La Cabecera de Opciones de destino puede presentarse (si se presenta) antes o despues de la cabecera de autenticación.

Fig. 5.22 - *Cabecera de Autenticación en modo Transporte*

En el modo túnel, el paquete contiene una cabecera IPv6 interna (que contiene la dirección del ultimo destino) y una cabecera IPv6 externa, la cual puede ser direccionada hasta un gateway de seguridad. En el modo túnel la cabecera de autenticación protege a la cabecera IPv6 interna, a las capas superiores y para la cabecera IPv6 externa, como se puede observar en la figura 5.23.



Notas
(1) Si se presenta
(2) Sujetas a modificacion durante el procesamiento del paquete, debido a los campos que sufren modificaciones durante el transcurso del paquete hasta su destino

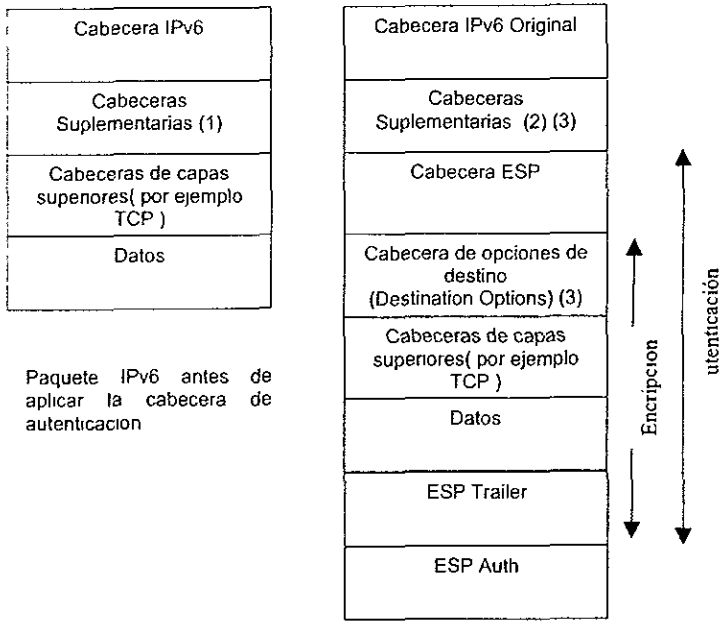
Fig. 5.23 - *Cabecera de Autenticacion en modo Túnel*

5.2.4 Función de la Cabecera ESP

La cabecera Encapsulating Security Payload (ESP) provee confidencialidad por medio de la encripción de la información, autenticación de los datos de origen, integridad orientada a conexión, servicio de anti-reenvio de paquetes y limitada confidencialidad contra el análisis de trafico. Tanto la cabecera de autenticación y la cabecera ESP pueden ser usadas para el control de acceso, basados en una llave de distribución para poder descifrar los paquetes. El alcance de la autenticación ofrecida por la cabecera ESP no es

tan extensa comparada con la que ofrece la cabecera de autenticación; si se desea confidencialidad pero no gran fuerza en lo que respecta a la autenticación solo se puede usar la cabecera ESP, pero si se desea confidencialidad y gran fortaleza en la autenticación debe de usarse la cabecera de autenticación y la cabecera ESP.

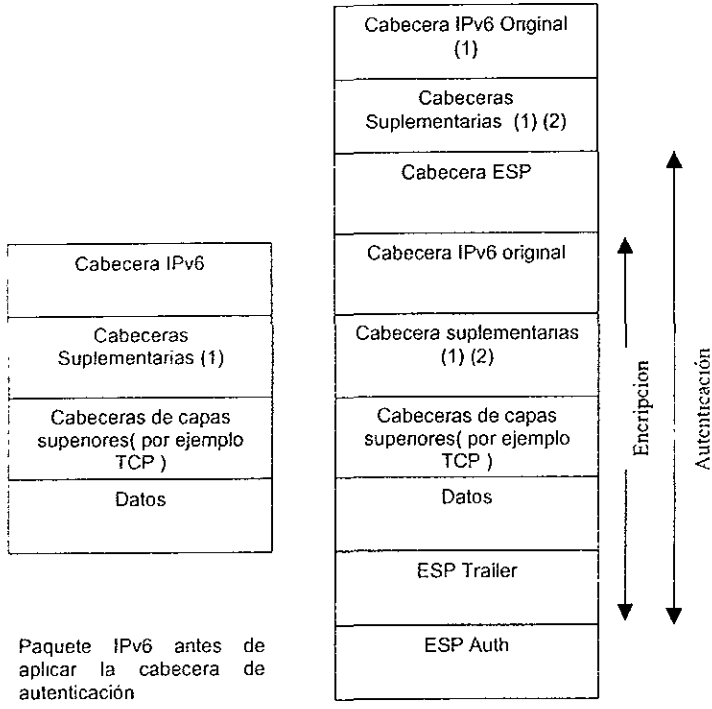
La cabecera ESP al igual que la cabecera de autenticación puede usarse en modo transporte o en modo túnel En modo transporte la autenticación se da solo para los datos, los protocolos de capas superiores, la cabecera de Destinations Opcion (en el caso de que se presente después de la cabecera ESP) y para la cabecera ESP. La encripción se da solo para la cabecera de Destinations Opcion (si se presenta después de la cabecera ESP), protocolos de capas superiores y los datos. La forma en que la cabecera ESP maneja el paquete en modo transporte se puede observar en la figura 5.24



Notas:
(1) Si se presenta
(2) Cabeceras Hop by Hop, de Opciones de destino (Destination Options), de ruteo, de fragmentación si se presentan
(3) La Cabecera de Destination Options puede presentarse (si se presenta) antes o después de la cabecera ESP.

Fig. 5.24 - *Cabecera de ESP en modo Transporte*

En el modo túnel la cabecera IPv6 original (cabecera IPv6 interna) entra en la parte que es autenticada y encriptada La forma en que se maneja el paquete después de aplicar la cabecera ESP en modo túnel se muestra en la figura 5.25.



Notas.
(1) Si se presenta
(2) Sujetas a modificación durante el procesamiento del paquete, debido a los campos que sufren modificaciones durante el transcurso del paquete hasta su destino

Fig 5.25 - *Cabecera de ESP en modo Transporte*

En la cabecera de Autenticación se usa el algoritmo Estandar de Encripción de Datos (DES, data Encription Estandar) y el CAST con una llave de 128 bits, esto se explica mejor en el punto 5.2 5.

5.2.5 IPsec y el manejo de llaves

Como se menciono anteriormente el protocolo IPsec en IPv6 utiliza varios algoritmos de Criptografia para poder generar la llave de autenticación y/o de encripción según se requiera Primeramente daremos una breve explicación de lo que es una algoritmo de cifrado y su llave seguridad.

Un algoritmo es una función matemática usada para la encriptación y desencriptacion de la información, generalmente se tiene dos funciones una para la encriptación y otra para la desencriptación. Para realizar el proceso de encriptación y desencriptacion se utilizan llaves.

Una llave es una clave que es usada para poder cifrar o descifrar un paquete, se puede usar una sola llave para cifrar el paquete y descifrarlo, o se puede usar una llave para cifrarlo y otra llave para descifrarlo.

Existen algoritmos de una sola llave, estos algoritmos se conocen como algoritmos simétricos, la llave que se usa para cifrar y descifrar la información es la misma, la desventaja que se tiene con este tipo de algoritmos es que es necesario transmitir la llave por un canal seguro, es muy difícil el intercambio seguro de infamación con nodos desconocidos

También existen los algoritmos de llaves públicas, estos algoritmos también son conocidos como algoritmos asimétricos o de dos llaves, en este tipo de algoritmos usan dos tipos de llaves, la llave privada y la llave publica. La llave publica se usa para cifrar los mensajes y la llave privada se usa para descifrarlos. La llave privada solo la tiene una sola entidad, ya sea un host o un gateway, en cambio la llave publica la tienen varias entidades En el uso llaves publicas no se requiere de un canal seguro para la transmisión de las llaves publicas, cada entidad solo requiere de sus dos tipos de llaves, el uso de algoritmos asimétricos son mas lentos comparados con los algoritmos simétricos. Por ejemplo cuando un host (host 1) necesita enviarle información a un nodo (host 2) en forma segura lo que hace es que cifra la información con la llave publica del nodo a quien va dirigida la información (host 2), con esto se asegura que solo el host 2 podrá descifrar la información mediante el uso de la llave privada Esto se puede observar en la figura 5 26

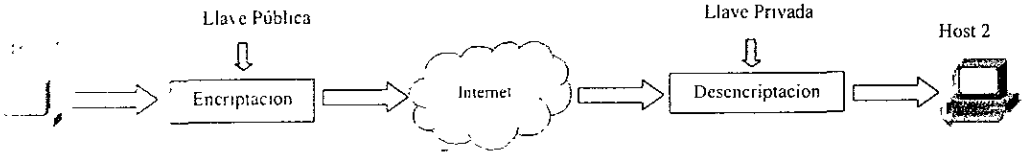


Fig. 5.26 - *Uso de las Llaves Públicas*

Existen funciones hash de una sola vía los cuales generan un valor de tamaño fijo a partir de una entrada de longitud variable, este valor se les conoce como huellas digitales. El calculo realizado por la función hash es muy fácil de obtener, pero muy difícil de encontrar los valores de la entrada que generaron el valor hash. Por lo que se dice que es calculable por un lado pero incalculable por el otro, ya que el cambio de un bit en la entrada aproximadamente cambia la mitad de los bis de salida.

Como se menciono anteriormente IPsec en IPv6, usa los algoritmos DES (Data Encription Standar), CAST que son las siglas de los nombres quien desarrollaron este algoritmo MD5 (Mesage Digest Algoritm 5), SHA (Secure Hash Algoritm).

EL Algoritmo DES fue desarrollado en Estados Unidos por IBM y la NSA, este algoritmo es de una sola llave este algoritmo usa una llave cuya longitud es de 56 bits, que hoy en dia es una vulnerabilidad usar una llave de esa longitud, debido a que con la tecnología actual es posible descifrar la información.

El algoritmo CAST fue desarrollado por C. Adams y S. Tavares de Northern Telecom en Canada Este algoritmo es de una sola llave cuya longitud puede ser de 64 bits o de 128 bits, la ventaja de este algoritmos es que es de uso gratuito sin restricciones.

MD5 es una función hash de una sola vía , fue desarrollado por RSADS, este tipo de función hash genera un valor de una longitud de 128 bits.

SHA al igual que MD5 es una función hash de una sola vía, fue desarrollado por NIST y la NSA, como parte del Estándar de la Firma Digital (DSS, Digial Signature Standar), esta funciona genera un valor de 160 bits la cual la hace mas poderosa en comparación con las antes citadas, trabaja muy similar a MD5.

El manejo de llaves en IPv6 puede ser en banda o fuera de banda. El manejo de llaves en banda implica transmitir los datos relacionados al manejo de las llaves en la cabecera IPv6. le manejo en fuera de banda significa en transmitir los datos relacionados al manejo de las llaves en protocolos de capas superiores tales como UDP o TCP. El manejo de las llaves incluye información como lo es el establecimiento de llaves y métodos de verificación, métodos de transferencia de las llaves, autenticación, simetría.
La forma mas simple de administración de las llaves es la administración manual, donde una persona configura manualmente cada nodo con su propia llave y con las llaves de los demás nodos, esto es común en entornos estáticos, pequeños, pero no de gran escala. También existe protocolos de distribución de llaves automáticos los cuales permiten que las actualización de llaves se hagan de una forma automática, estos administradores de llaves son útiles para redes de gran tamaño

5.2.6 Seguridad en IPv6 conjunta con el uso de Firewall's

La seguridad en IPv6 puede ser usada en conjunto con otros mecanismo de seguridad existentes, uno de ellos puede ser Firewall.

En firewall puede ser configurada para que soporte lo que son las cabeceras de seguridad de IPv6, para una mayor protección de la organización a la que se encuentra conectada el firewall.

Un ejemplo seria que una organización quiera usar encripción con el firewall, para proteger el trafico sensible modificación o a posibles robos de información. Teniendo en cuenta de que la organización tenga varios sitios que son interconectados mediante un proveedor comercial de servicio IP, la organización puede colocar en cada uno de sus sitios un firewall que soporte la configuración de la seguridad IPv6, para así mediante el uso de túneles enviar la información encriptada a través de internet.

El uso de túneles permite el manejo de la seguridad de IPv6, ya que la información al ser encapsulada (debido al tunelado) con el protocolo IPv6, le coloca las cabeceras de IPv6 entre ellas las cabeceras de seguridad, permitiendo que la información viaje encriptada a través de internet, al llegar los paquetes al firewall destino, este desencapsula la información y permite el acceso de los paquetes a su red local.

CONCLUSIONES

En un inicio, el conjunto de protocolos TCP/IP surgió como una necesidad para comunicarse eficientemente entre varios centros militares del Departamento de la Defensa de los Estados Unidos. Poco tiempo despues adoptaron estos protocolos como estandares, para convertirse en un modelo de comunicación confiable y robusto que permite la operación entre equipos de cualquier tipo de hardware y que utilicen cualquier sistema operativo. Esto es lo que lo ha hecho el más popular de los estándares de comunicaciones y ha hecho que se desarrollara mucho más allá de su propósito original.

Los protocolos TCP/IP se han diseñado, desde un principio, para interconectar sistemas no conectados a la misma red. El modelo de comunicaciones debe ser suficientemente flexible como para que, a medida que se incrementan los avances tecnológicos y las necesidades de los usuarios, pueda ajustarse a tales exigencias.

Las necesidades y exigencias del mundo moderno son cada vez mayores, demandando el soporte de nuevas aplicaciones, tales como audio y vídeo en tiempo real, voz sobre IP, telefonía celular, etc., lo cual ha propiciado que el modelo de comunicación TCP/IP ya no sea lo suficientemente flexible y capaz de proporcionar la infraestructura requerida para solucionar tales demandas. Todo esto ha derivado en una propuesta para desarrollar un nuevo estándar de comunicaciones dentro del modelo de TCP/IP que elimine todos los aspectos problemáticos de IPv4 que han surgido conforme al gran crecimiento de la red conocida como internet, así como al surgimiento de nuevas necesidades, además de proporcionar un margen considerablemente grande para evitar que se vuelva obsoleto.

Esta nueva propuesta ha sido llamada IP de Nueva Generación o IPv6 y pretende solucionar la cada vez más crítica ineficiencia de IPv4, ya que este protocolo fue diseñado en base a la problemática actual que afecta a IPv4, así como previendo a un futuro para que el nuevo protocolo no sea obsoleto a las necesidades futuras. En el protocolo IPv6 se han hecho varios cambios referentes a las problemáticas actuales del protocolo IPv4, entre ellos se encuentran: el empleo de direcciones con una longitud cuatro veces mayor, una distribución jerárquica y ordenada de las direcciones, incorporación de protocolos de seguridad como parte inherente al nuevo conjunto de protocolos y no de manera opcional, como es considerado en la versión actual, además de permitir la comunicación entre nodos móviles (redes inalámbricas) y nodos fijos, el reemplazo de direcciones de tipo broadcast por multicast aumentando así el desempeño de la red, el empleo de la autoconfiguracion de direcciones. Otra de las características de la nueva versión es el uso de una cabecera base de longitud fija y cabeceras suplementarias que son empleadas únicamente cuando se requiere; por ejemplo, en el caso de que sea necesaria la fragmentacion, se añadirá una cabecera de fragmentación, y a diferencia con el protocolo IPv4 que la fragmentación se hacia durante el transcurso del paquete hacia su destino por los ruteadores, bajando el desempeño de la red, ahora solo la fragmentación será hecha

por el nodo origen, mediante un descubrimiento previo del MTU de la red, dando así un mejor desempeño a la red ya que con esto los ruteadores no serán los encargados de realizar la fragmentación, agilizando así el envío de los paquetes

Debido a todo lo anterior, podemos decir que la nueva versión del protocolo IP cumple con las expectativas originadas por los avances tecnológicos del mundo moderno y se presenta como la solución más viable a las necesidades de intercambio global de información a través de Internet, con la única desventaja que plantea la transición de la versión actual a la nueva. Aunque, si tomamos en cuenta que cada vez que exista una actualización o modificación debe existir una etapa de transición, queda únicamente definir cuál será la mejor opción.

El principal problema de la migración de una versión a otra radica en la incompatibilidad entre los dos protocolos debido a la longitud de las direcciones y al formato de las cabeceras. Actualmente se está trabajando en varios mecanismos de transición, de los que analizamos en este trabajo, creemos que el que menos dificultades puede causar es IPv6/IPv4 Network Address and Protocol Translation, debido a que este mecanismo puede implementarse tanto en redes IPv4 como en redes de IPv6, mientras que otros mecanismos únicamente se pueden aplicar en redes IPv4 (como NAT-PT) o en redes IPv6 (como SIIT), pero no en ambas.

De lo que se trata es de hacer un mecanismo de transición estándar, que evite la interaccion simultánea de varios de ellos, con la consecuente problemática y que haga de Internet un sistema homogéneo que permita una comunicación transparente entre diversos sistemas autónomos Otro de los aspectos más fiable dentro del proceso de transición es el uso del tunelado (es la forma en como se transportan paquetes IPv6, encapsulandolos con una cabecera IPv4) para que mediante este mecanismo se permita la comunicación de dos sitios IPv6, mediante el uso de la infraestructura de la red actual de IPv4, con el proceso de encapsulamiento que brinda el tunelado.

Dentro del proceso de evolución, debe trabajarse a nivel aplicación para que estas aplicaciones sufran los cambios requeridos para que puedan funcionar con las caracteristicas del protocolo IPv6, entre las aplicaciones mas comunes es el DNS, para que se pueda hacer la resolución de nombres por las direcciones IPv6, así como también por las de IPv4 durante el proceso de transición. A nivel aplicación no le afectará demasiado, pues solo las nuevas aplicaciones deberán trabajar con direcciones de IPv6.

Con este trabajo nos pudimos dar cuenta del problema cada vez más apremiante de la escasez de direcciones de Internet, así como la insuficiencia de la red actual de IP de no soportar nuevas aplicaciones tales como voz y video en tiempo real, y si no se propone una solución real en el corto plazo, pueden llegar a colapsarse las comunicaciones a nivel mundial en un período no muy largo

GLOSARIO

A

ANSI(American national Standars Institute)

Es la entidad normalizadora de los Estados Unidos. Fue fundada en 1918 y es una organizacion no gubernamental constituida por más de un millar de organizaciones comerciales. sociedades profesionales y corporaciones. ANSI por si misma no crea estándares. sino que se dedica a coordinar y sincronizar las actividades de otras organizaciones que si desarrollan estándares, y asegurar que todos los intereses afectados tienen una oportunidad de participar en el proceso

API (Application Programming Interface)

Especificación las características de las funciones que define una interfase para un servicio

APPN(Advanced Peer to Peer Networking)

APPN maneja el establecimiento de la sesión, entre nodos pares, dinámicamente realiza el calculó de la ruta.

ARP (Address Resolution Protocol)

Este protocolo es usado por IP (Internet Protocolo) para mapear una dirección IP con una direccion MAC.

ARPA (Advanced Research Projects Agency)

La agencia del Departamento de Defensa de los Estados Unidos. Es responsable de numerosos avances en rede y comunicaciones.

ARPANET.

Red desarrollada por ARPA que utiliza técnicas de "packet-switching". Fue la primera red que trabajó con estas técnicas.

ASCII

American Standard Code for Information Interchange; Código de caracteres de siete bits estandarizado por ANSI y está designado como x3.4-1977 en donde 1977 es el año de la ultima revisión. ASCII también fue estandarizado por ISO y CCITT y es conocido internacionalmente como Alfabeto #5 Internacional de Telégrafos. Es casi el código

universal de representar caracteres en las computadoras, a excepción de algunas máquinas de IBM que emplean aún EBCDIC y BCD.

ATM (Asynchronous Transfer Mode)

ATM es un estándar internacional de conmutación de celdas, la cual soporta servicios tales como voz, datos y video, esta tecnología maneja de una longitud de celdas de 53 bytes ATM es una tecnología de alta velocidad.

Autenticación

La verificación de la identidad de una persona o un proceso.

B

Backbone (espina dorsal de red)

Es la infraestructura de conexión principal de una red y está constituida por los enlaces de mayor velocidad dentro de dicha red.

Balún

Los balunes/filtros permiten enlazar cables metálicos con características diferentes, evitan que se produzcan reflejos y permiten el paso de ciertas frecuencias y bloquean el paso de otras.

BCD.

Siglas en inglés de "Binary-Coded Decimal" (decimal codificado en binario). Técnica para representar los dígitos de un número en decimal (0-9) cada uno mediante una secuencia de cuatro bits.

BROADCAST.

Paquete de datos que es enviado a todos los nodos de una red.

BSC (*Binary Synchronous Communications*)

Protocolo de enlace de datos para aplicaciones half-duplex.

C

Cabecera (header)

Información de control colocada antes de los datos cuando encapsulas estos datos para enviarlos en la red

CCITT

International Telegraph and Telephone Consultative Committee (Comité Consultativo Internacional de Teléfonos y Telégrafos) Agencia de la Unión Internacional de Telecomunicaciones.

Circuito virtual

Circuito lógico creado para asegurar una comunicación entre dos dispositivos de una red Un circuito virtual es definido por un par de VPI/VCI y puede ser un circuito virtual permanente (PVC) o switcheado (SVC). Los circuitos virtuales son usados en Frame Relay y X 25. En ATM un circuito virtual es llamado Canal virtual (Virtual Channel).

CSU (Channel Service Unit)

Dispositivo de interfase digital que conecta un equipo de usuario final , hasta una red digital.

D

Datagrama

Es un grupo lógico de información enviado como una capa de la red sobre el medio de transmisión , sin ningún establecimiento previo de algún circuito virtual Los datagramas IP son la unidad primaria de información en internet.

DDN (Defense Data Network)

Es red militar de Estados Unidos compuesta de la red conocida como MILNET y varias redes secretas, DDN es operada y mantenida por DISA

DecNET

Es un grupo de productos de comunicaciones (incluyendo una suite de protocolos), desarrollado y soportado por Digital Equipment Corporation. DECnet/OSI (también llamada DECnet fase V) soporta protocolos de OSI y protocolos propietarios de Digital.

DES (DATA ENCRYPTION STANDARD)

Es un algoritmo estándar criptográfico de datos.

DNS (Domain Name Service)

Base de Datos distribuida que mapea un sistema o nombres de usuario con direcciones IP.

DSU (DATA SERVICE UNIT)

Unidad de servicio de información. Dispositivo usado en DDS para interconectar un DTE con el loop local.

E

EBCDIC

"Extended Binary-Coded Decimal Interchange Code". Es un código de carácter desarrollado por IBM. Es un código de ocho bits, algunas de las 256 combinaciones no se utilizan.

ENCRYPTION (CODIGO CIFRADO)

Proceso mediante el cual la información se convierte en un otra aparentemente sin sentido, pero transformada mediante un código generalmente algorítmico, para protegerla de ser recibida por usuarios sin autorización.

EGP (exterior Gateway Protocol)

Es un protocolo que permite el intercambio de información de ruteo, entre sistemas autónomos. Este protocolo esta documentado en el RFC 904. EGP es un protocolo obsoleto que fue remplazado por el protocolo BGP (Border Gateway Protocol).

Ethernet

Es una tecnología LAN que fue inventada por Xerox Corporation y desarrollado por Xerox Intel y Digital Equipment Corporation. Las redes Ethernet usan CSMA/CD y corre una variedad de tipo de cables a una velocidad de 10 Mbps. , Ethernet es similar al estándar IEEE 802.3

F

FDDI (Fiber Distributed Data Interface)

Es una tecnología de redes LAN, definida por ANSI X3T9.5, especifica una red de token passing a 100 Mbps, usando fibra óptica, FDI usa un doble anillo, proveyendo redundancia.

FLAG

Los protocolos de bits usan una secuencia de ocho bits llamada "flag" o bandera para determinar el principio y el final de un "frame" o marco. Se utiliza una técnica de "relleno de bits" para que no ocurra dentro del "frame" una secuencia de "flags" inadvertida.

Frame

Procedimiento mediante el cual un protocolo le añade a los datos originales un encabezado ("header") y una cola ("trailer").

Frame Relay

Es un protocolo de switcheo en la capa de enlace de datos que maneja circuitos virtuales, usando encapculación HDLC entre la conexión de los dispositivos. Frame Relay es más eficiente que X.25.

FTP (File Transfer Protocol)

Soporta la transmisión de caracteres de texto o ficheros binarios de una red a otra.

G

Gateway

En la comunidad de IP, es un termino viejo referido a un dispositivo de ruteo Hoy en día el termino ruteador es usado para describir los nodos que realizan esta función, y el termino gateway se refiere a un dispositivo especial que ejecuta una conversión de un protocolo a otro.

GGP (Gateway –Gateway Protocol)

Es un protocolo que especifica como core-routers (gateways) podrían intercambiar información de ruteo.

H

HDLC

Siglas para High Level Data Link Control; estándar comprensivo desarrollado por la ISO Es en protocolo de bit de la capa de conexión.

HELLO

Es un protocolo de ruteo interior usado principalmente por nodos NSFnet. Hello permite el switcheo de paquetes para descubrir un retardo mínimo en las rutas.

Host

Es una computadora en una red. Es similar a nodo, excepto que el host usualmente implica un sistema de computo, y un nodo aplica a cualquier sistema de red, incluyendo servidores y ruteadores

Hub

En Ethernet y IEEE 802.3 es un repetidor multipunto, algunas veces llamado concentrador

I

IEEE (Institute of Electrical and Electronics Engineer's)

Instituto de Ingenieros Eléctricos y Electrónicos Es una entidad que ha generado muchísimos estándares en telecomunicaciones.

IEEE802

Estos son los estándares para la conexión física y eléctrica de LAN's desarrollado por IEEE (Institute of Electronic and Electrical Engineers).

IEEE 802.1D

Estandar de la IEEE para el nivel de acceso de control para los puentes o "bridges", entrelazando redes IEEE 802.3, 802 4 y 802.5.

IEEE 802.2

Estandar para la capa de conexión lógica, para usarse con redes IEEE 802.3, 802.4 y 802 5

IEEE 802.3 1Base5

Especificación de la IEEE que iguala el antiguo producto de AT&T StarLAN. Este designa un rate de 1 mbps, técnica de base de banda y un máximo de distancia de cable de 500 metros

IEEE 802.3 10Base2

Esta especificación de la IEEE iguala el cableado estrecho de Ethernet. Este designa un rate de señal de 10 mbps, técnica de base de banda, y un máximo de distancia de cable de 185 (casi 200) metros.

IEEE 802.4

Aquí se describe un LAN que usa un rate de señales de 10 megabits por segundo, control de acceso para "token-passing" y una topología de bus física. Este es típicamente usado como parte de redes que siguen a MAP (Manufacturing Automation Protocol) desarrollado por General Motors. Este es a veces confundido con ARCnet pero no es el mismo

IEEE 802.5

Esta especificación de la IEEE describe un LAN que usa 4 o 16 megabits por segundo, MAC "token-passing" y una topología física de anillo. Es utilizado por los sistemas IBM de Token-Ring.

IGP (Interior Gateway Protocol)

Es un protocolo usado en internet para realizar el intercambio de información de ruteo dentro de un sistema autónomo. Algunos ejemplos de este tipo de protocolos son IGRP, OSPF, y RIP.

INTERFACE

Una interfase provee los medios para la interconexión de equipo (o procesos) localizados en un lugar específico. Ejemplos de interfaces lo son el RS232-C, RS449, X-21, etc

InterNIC

Es una organización que sirve a la comunidad de Internet para el suministro de asistencia a usuarios, documentación, capacitación, registro de servicio de nombre de dominios en internet y otros servicios. Formalmente llamado NIC.

Intranet

Red de uso privado que emplea los mismos estándares y herramientas de Internet. Es uno de los segmentos del mercado de computación que más impulso está cobrando.

IPv4 (Internet Protocolo Version 4)

Es el protocolo que se encuentra en la capa de red dentro del modelo de TCP/IP. Es el protocolo usado actualmente en Internet.

IPv6(internet Protocol Version 6)

Es el protocolo de internet que pretende remplazar al actual protocolo IP, este protocolo es también llamado IPng (Internet Protocolo Next Generation).

IPX(Internetwork Packet Exchange)

Protocolo de comunicaciones de NetWare de Novell utilizado para la transferencia de datos entre los nodos de una red.

ISDN (Integraed Service Digital Network)

Red Digital de Servicios Integrados, tecnología en plena evolución que es ofrecida por las compañías telefónicas más importantes. ISDN combina servicios de voz y digitales a través de la red en un solo medio, haciendo posible ofrecer a los clientes servicios digitales de datos así como conexiones de voz a través de un solo "cable". Los estandares de la ISDN los especifica la ITU-TSS.

IS-IS

Es el protocolo de ruteo de interdominio. El protocolo de OSI especifica como los ruteadores se comunican con otros ruteadores de diferentes dominios.

ISO

Es una organización no gubernamental, fundada en 1947. Su misión es coordinar el desarrollo y aprobación de los estándares internacionales. Es decir, promueve el desarrollo de la estandarización y actividades relacionadas en todo el mundo con el objetivo de facilitar el intercambio de bienes y servicios y la cooperación en los ámbitos intelectuales, científicos, tecnológicos, y económicos.
Su ámbito de trabajo cubre todas las áreas, incluyendo la normalización de las redes de area local, con excepción de las áreas electrotécnicas, que son coordinadas por la Comision Electrotécnica Internacional.

ITU (International Telecommunications Union) (Unión Internacional de Telecomunicaciones)

Agencia de las Naciones Unidas que coordina los diversos estándares nacionales de telecomunicaciones de forma que las personas pueden comunicarse entre sí independientemente del país donde vivan.

ITU-TSS (International Telecommunications Union-Telecommunications Standards Sector)

(Union Internacional de Telecomunicaciones- Sector de Estándares de Telecomunicaciones) Nuevo nombre del CCIT tras la reorganización de la ITU Su función es la misma, habiendo cambiado solo el nombre.

L

LAN (LOCAL AREA NETWORK)

Red de Area Local, o más brevemente Red Local de Computadoras. Se refiere a una red de computadoras conectadas bajo un mismo protocolo y tipo de conexión física, sin modulación de la señal y en distancias cortas (menores generalmente a los 10 KM, por ejemplo el diámetro de un campus universitario).

M

MAN (Metropolitan Area Network)

Red de área metropolitana. Red de datos la cual se extiende en un área de varios Kilometros

MTU (Maxium Transfer Unit)

Es la unidad máxima de transferencia, esto es el tamaño máximo en bytes de un paquete que puede transferirse por una tecnología de red

N

NetBEUI:

Este termino proviene de NetBIOS Extended User Interface. Se trata de un controlador de dispositivo de red Es el controlador de transporte proporcionado con LAN Manager (Administrador de Red Local de Microsoft), y es el protocolo de comunicación entre redes LAN

NIC (Network Information Center)

Organización cuyas funciones han sido asumidas por el InterNIC.

NSFNET (National Science Foundation Network)

Es una red que fue controlada por la organización National Science Foundation, ahora provee servicios de red en ayuda a la educación y la investigación en los Estados Unidos.

O

Octeto

Un octeto esta formado por 8 unidades de información (llamadas "bits"). Este término se usa a menudo en vez de "byte" en la terminología de redes porque algunos sistemas tienen "bytes" que no están formados por 8 bits.

OSI(Open Systems Interconection)

Esta es una recomendación de la ISO que describe una estructura de siete capas para la particion de comunicación de datos y funciones de telecomunicaciones en capas.

P

Payload

Porción de una célula, frame o paquete, que contiene información de protocolos de capas superiores.

PROTOCOLO

Este es el procedimiento (conjunto de pasos, mensajes, forma de los mensajes y secuencias) que se utiliza para mover la información de una localización a otra sin errores

PROTOCOLO DE ACCESO

Estas son las reglas de tráfico a las que se sostienen estaciones de trabajo LAN para evitar la colisión de datos cuando se envían señales a través de un medio de red compartido También conocido como MAC o "Media Acces Control Protocol". Ejemplos comunes de esto lo es el CSMA o "carrier sense multiple access" .

Puentes (bridges)

Los puentes son dispositivos que tienen usos definidos Primero, pueden interconectar segmentos de red a través de medios físicos diferentes; por ejemplo, no es poco común ver puentes entre cable coaxial y de fibra óptica. Además, pueden adaptar diferentes protocolos de bajo nivel (capa de enlace de datos y física de modelo OSI).

R

RS-232

En este estándar se define las características físicas y eléctricas para poder conectar un equipo de comunicaciones de datos DTE y DCE. Este provee la conectividad desde el codec permitiendo las entradas de datos para transmitir desde 300 bits por segundo hasta 19 2 Kbps

Ruteadores (routers)

Los ruteadores determinan la trayectoria más eficiente de datos entre dos segmentos de red Operan en la capa superior del modelo OSI a la de los puentes -la capa de red- no estan limitado por protocolos de acceso o medio

S

Segmento

Es una sección de una red que esta separada por un ruteador, bridge o switch.

Sistema Autónomo

Es una red que es administrada por una sola entidad o protocolo

SMTP (Simple Mail Transfer Protocol)

Un protocolo básico de Correo electrónico.

SNA (system Network Architecture)

Es una arquitectura de red desarrollada por IBM en los años 70's, esta arquitectura es similar a la arquitectura del modelo OSI

SNMP (Simple Network Management Protocol)

Este protocolo de administración de redes es usado solo en redes TCP/IP, SNMP proveo mecanismos para monitorear y controlar los dispositivos de una red tales como ruteadores, switches, concentradores, etc.

Socket

Numero de identificación compuesto por dos números: la dirección IP y el número de puerto TCP En la misma red, el número IP es el mismo, mientras que el número de puerto es el que cambia En máquinas de distintas redes, pueden tener el mismo número de puerto sin llevar a confusión, pues el número IP las distingue

T

TCP/IP (Transmission Control Protocol/Internet Protocol)

Desarrollado por la "Defense Advanced Research Projects Agency" en USA, es el protocolo básico de Internet o Intranet.

TELNET

Sesión que realiza una conexión directa y altamente insegura entre dos máquinas.

TFTP (Trivial File Transfer Protocol)

Es una versión mas simple de FTP que permite enviar archivos de una red a otra pero sin necesidad de usar, login y password para el inicio de una sesión.

Token Ring

Es una tecnología de red desarrollada por IBM, esta tecnología corre a una velocidad de 4 hasta 16 Mbps en una topología tipo anillo. Es similar al estándar IEEE 802.5

U

UDP (User Datagram Protocol)

Acronimo de User Datagram Protocol (Protocolo de datagrama a nivel de usuario), perteneciente a la familia de protocolos TCP/IP Este protocolo no es tan fiable como TCP pues se limita a recoger el mensaje y enviar el paquete por la red. Para garantizar el éxito de la transferencia, UDP hace que la máquina de destino envía un mensaje de

vuelta Si no es así, el mensaje se envía de nuevo. Con este protocolo no se establece una conexion entre las dos máquinas.

UNIX

Sistema operativo diseñado para ser usado por mucha gente al mismo tiempo (es multiusuario) y tiene TCP/IP. Es el sistema operativo más común para servidores en Internet

V

V.24

Estandar de la ITU-T para una interfase de la capa física entre un DTE y un DCE. V24 es esencialmente el mismo que el estándar EIA/TIA-232.

VCI (virtual Channel)

Este es un campo de una longitud de 16 bit en la cabecera de una celda ATM, el VCI junto con el VPI, es usado para identificar el destino siguiente.

VPI (Virtual Path Identifier)

Este es un campo de una longitud de 8 bits en la cabecera de una celda ATM, el VPI junto con el VCI, es usado para identificar el destino siguiente.

W

WAIS (Wide Area Information Servers)

Acrónimo de Wide Area Information Servers (Servidores de Información de Área Extendida) Paquete de software comercial que permite indicar grandes cantidades de información y hacer que esos índices puedan buscarse a través de Internet. Una característica primordial de WAIS es que los resultados de búsqueda están medidos de acuerdo a lo relevantes que son, y otras búsquedas subsiguientes.

WAN (Wide Area Network)

Es una red de área amplia, que es de grandes dimensiones como lo es Internet.

X

X.21

Es el estándar de comunicaciones de ITU-T para comunicaciones seriales sobre enlaces digitales sincronos.

X.25

Estándar internacional que define protocolos de comunicación de conmutación de paquetes ("packet-switched comunication") para redes privadas o públicas.

X.28

La recomendación de ITU-T que define la interfase de la terminal PAD en redes X.25

X.400

Es la recomendación de ITU-T que especifica un estándar la transferencia de correo electrónico.

GLOSARIO DE RFC's

RFC 1981 Path MTU Discovery for IP version 6

En este documento se describe el proceso del descubrimiento del MTU con el protocolo IPv6

RFC 2185 Routing Aspects Of IPv6 Transition

En este documento se señalan las características pincipales de ruteo usadas con IPv6 durante la transición

RFC 2374 AN IPV6 AGGREGATABLE GLOBAL UNICAST ADDRESS FORMAT

En este documento se describe el formato de la dirección unicast.

RFC 2373 IP VERSION 6 ADDRESSING ARCHITECTURE

Este documento define la arquitectura de direccionamiento de IPv6.

RFC 2375 IPV6 MULTICAST ADDRESS ASSIGNMENTS

En este documento se define el asignamiento inicial de las direcciones multicast IPv6.

RFC 2402 IP Authentication Header

En este documento se describen las características y funciones de la cabecera de autenticación.

RFC 2406 IP Encapsulating Security Payload (ESP)

En este documento se describen las caracteristicas y funciones de la cabecera ESP.

RFC 2450 PROPOSED TLA AND NLA ASSIGNMENT RULES

Este documento provee información acerca de las reglas para la asignación de direcciones sobre la base de la agregación de identificadores de alto nivel (Top-Level Aggregation Identifiers, TLA ID), así como la asignación de los identificadores del siguiente nivel (Next-Level Aggregation Identifiers, NLA ID)

RFC 2460 INTERNET PROTOCOL, VERSION 6 (IPV6) SPECIFICATION

Este documento describe todas las cabeceras base y las cabeceras complementarias usadas por el protocolo IPv6.

RFC 2461 Neighbor Discovery for IP Version 6 (IPv6)

Este documento describe el funcionamiento del protocolo de descubrimiento del nodo vecino (Neighbor Discovery Protocol), el cual es usado por los nodos para poder determinar la dirección física de sus nodos vecinos

RFC 2462 IPV6 STATELESS ADDRESS AUTOCONFIGURATION

En este documento se describe el mecanismo de autoconfiguración de dirección sin estado (IPv6 Stateless Address Autoconfiguration).

RFC 2463 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification

En este documento se describen los mensajes del protocolo de control de mensaje de internet (Internet Control Message Protocol, ICMP) que son usados por el protocolo IPv6.

RFC 2464 TRANSMISSION OF IPV6 PACKETS OVER ETHERNET NETWORKS

En este documento se describe el frame para poder transmitir paquetes IPv6 sobre redes Ethernet

RFC 2467 TRANSMISSION OF IPV6 PACKETS OVER FDDI NETWORKS

En este documento se describe el frame para poder transmitir paquetes IPv6 sobre FDDI, así como también el direccionamiento que se usa sobre esta tecnología.

RFC 2470 TRANSMISSION OF IPV6 PACKETS OVER TOKEN RING NETWORKS

En este documento se describe el frame y el MTU para poder transmitir paquetes IPv6 sobre Token Ring, así como también el direccionamiento que se usa sobre esta tecnología

RFC 2471 IPv6 Testing Address Allocation

Este documento describe el plan de asignación de direcciones IPv6 para ser usado en pruebas de prototipos de software que soporten IPv6.

RFC 2472 IP VERSION 6 OVER PPP

Este documento describe los métodos para transmitir paquetes IPv6 sobre enlaces de tipo PPP, así como el protocolo de control de red (Network Control Protocols , NCPs) para el establecimiento y configuración de IPv6 sobre PPP.

RFC 2474 Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers

Este documento describe las caracteristicas y funciones del campo que se encarga de los diferentes tipos de servicio en las cabeceras IPv4 e IPv6.

RFC 2491 IPV6 OVER NON-BROADCAST MULTIPLE ACCESS (NBMA) NETWORKS

En este documento se describe la arquitectura de IPv6 sobre redes de arquitectura de tipo de acceso multiple sin uso de broadcast tales como lo son ATM y Frame Relay.

RFC 2492 IPV6 OVER ATM NETWORKS

Describe los mecanismos usados para transmitir paquetes de IPv6 sobre ATM.

RFC 2590 TRANSMISSION OF IPV6 PACKETS OVER FRAME RELAY NETWORKS SPECIFICATION

Este documento describe los mecanismos para transmitir paquetes IPv6 sobre Frame Relay

RFC 2772 6BONE BACKBONE ROUTING GUIDELINES

Este documento proporciona información acerca de las políticas de ruteo manejadas en el 6bone, plantea los prefijos que se deben manejar para enlaces locales, loop-back, multicast, rutas de default, etc

BIBLIOGRAFIA

LIBROS:

Black, Uyless, *TCP/IP and related protocols*, McGraw-Hill, USA, 1995.

Bradner, Scott O and Mankin, Allison, *Ipng Internet Protocol Next Generation*, Adisson-Wesley, USA, 1996.

Collazo, Javier L., *Encyclopedic Dictionary of Technical Terms*, McGraw-Hill, USA, 1993.

CSO Latinoamerica Servicios Educacionales, *Tecnología Básica de Redes*, 3Com Corporation, México, 1995

García Tomas, Jesús, *Redes para Proceso Distribuido*, Ra-Ma, Madrid, 1997.

Goncalves, Marcus and Niles, Kitty, *Ipv6 Networks*, McGraw-Hill, USA, January 1998.

Hunt, Craig, *TCP/IP Network Administration*, O'Reilly & Associated, USA, 1998.

Miller, Mark A., *Implementing Ipv6, Migrating to the Next Generation Internet Protocol*, M&T Books, USA, 1998.

Sacristan, Eduardo, *Criptogrqafia y sus usos actuales*, Congreso General de Computo (computo 98@mx), México, Noviembre, 1998.

REQUEST FOR COMMENTS (RFC's):

Hinden, R (Nokia), M. O'Dell (UUNET) and Deering, S. Cisco *An Ipv6 Aggregatable Global Unicast Address Format*, RFC 2374, July 1998.

Hinden, R (Nokia), *Proposed TLA and NLA Assignement Rules*, RFC 2450, December 1998.

Hinden, R (Nokia), Fink, R. (LBNL) and Postel, J. (ISI), *Ipv6 Testing Address Allocation*, RFC 2471, December 1998.

Hinden, R (Nokia) and Deering, S. (Cisco Systems), *IP Version 6 Addressing Architecture*, RFC 2373, July 1998.

Nordmark, Erik (Sun Microsystems), *Stateless IIP/ICMP Translator (SIIT)*, Internet Draft <draft-ietf-ngtrans-siit-03 txt>, November 1998

Tsirtsis, George (BT Laboratories) and Srishuresh, Pyda (Lucent Technologies), *Network Address Translation – Protocol Translation (NAT-PT)*, Internet Draft <draft-ietf-ngtrans-natpt-03.txt>, November 1998

Egevang, K (Cray Comunications) and Francis, P (NTT), *The Ip Network Address Translator (NAT)*, RFC 1631, May 1994.

Conta, A (Lucent Technologies) and Deering, S. (Cisco Systems), *Generic Packet Tunneling in Ipv6*, Internet Draft <draft-ietf-ipngwg-ipv6-tunnel-08.txt>, January 1998.

Conta, A (Lucent Technologies) and Deering, S. (Cisco Systems), *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (Ipv6) Specification*, Internet Draft <draft-ietf-ipngwg-icmp-v2-02.txt>, September 1998

Bound, J (Compaq Computer Corporation) and Perkins, C. (Sun Microsystems), *Dynamic Host Configuration for Ipv6 (DHCPv6)*, Internet Draft <draft-ietf-dhc-dhcpv6-13.txt>, June 1998

Durand Alain (IMAG) and Buclin, Bertrand (AT&T Laboratories), *6-Bone Routing Practice*, Internet Draft <draft-ietf-ngtrans-6bone-routing-01.txt>, May 1998.

Thomson, S. (Bellcore) and Narten, T. (IBM), *Ipv6 Stateless Address Configuration*, RFC 2462, December 1998

Malkin G (Xylogics) and Minnear, R. (Ipsilon), *RIPng for Ipv6*, RFC 2080, January 1997.

McCann, J. (Digital Equipment Corporation), Deering, S. (Xerox PARC) and Mogul, J. Digital Equipment Corporation), *Path MTU Discovery for Ipv6*, RFC 1981, August 1996.

Gilligan R and Nordmark, E. (Sun Microsystems), *Transition Mechanisms for Ipv6 Hosts and Routers*, RFC 1933, April 1996.

Crawford, Matt (Fermilab), *Router Renumbering for Ipv6*, Internet Draft <draft-ietf-ipngwg-router-renum-05.txt>, September 1998.

Narten, T. (IBM), Nordmark, E. (Sun Microsystems) and Simpson, W. (Daydreamer), *Neighbor Discovery for IP version 6 (Ipv6)*, RFC 2461, December 1998.

Deering, S (Cisco Systems) and Hinden, R. (Nokia), *Internet Protocol Version 6 Specification*, RFC 2460, December 1998

Johnson David B (Carnegie Mellon University) and Perkins, Charles (Sun Microsystes), *Mobility Support in Ipv6*, Internet Draft <draft-ietf-mobileip-ipv6-07.txt>, November 1998.

Kent S (BBN Corporation) and Atkinson, R. (Home Network), *IP Authentication Header*, RFC 2402 November 1998.

Kent, S (BBN Corporation) and Atkinson, R. (Home Network), *IP Encapsulating Security Payload (ESP)*, RFC 2406, November 1998.

Nichols, K and Baker, F. (Cisco Systems), Blake, S. (Torrent Networking Technologies) and Black, D. (EMC Corporation), *Definition of the Differentiated Service Field (DS Field) in the Ipv4 and Ipv6 Headers*, RFC 2474, December 1998.

Kent, S (BBN Corporation) and Atkinson, R. (Home Network), *Secure Architecture for the Internet Protocol*, RFC 2401, November 1998.

Callon, R. (Cascade Communication) and Haskin, D. (Bay Network), *Routing Aspect of Ipv6 Trasition*, RFC 2185, September 1997.

DIRECCIONES IP

www.ewos.be/coexist/etg071/gintrd.htm#INICIO
Tipos de direcciones y métodos de transición

telecom noc.udg.mx/~kenia/ipv6-27.html
Tunelado en Ipv6

tangle seas gwu.edu/~reto/ipv6/comprar.htm
Seguridad en IP comparada con SSL

www ccsf.caltech.edu/~rfire/ipv6/tutorial.htm
Tutorial General de Ipv6

www cs-ipv6.lamcs ac.uk/ipv6/documents/papers/bound
Comparación de Ipv6 contra Ipv4

www.ip6.com/us/paper/migr/migr.htm
La migración de IPv4 a IPv6

www.pz-oekosys uni-kiel.de/~friedel/sont/ipng.htm
Información sobre la IETF y algunos RFC's

www 6bone.com
Pagina con información sobre el 6Bone