UNIVERSIDAD NACIONA DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

GESTION DE TRAFICO DE TUNELES IPV6/IPV4

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

GINO FRANCISCO ALANIA HURTADO

PROMOCIÓN

2002 – II

LIMA – PERÚ

2008

GESTION DE TRAFICO DE TUNELES IPV6/IPV4

Dedico este trabajo a:

A mis padres y ARUNI por estar justo en los momentos precisos de mi mayor esfuerzo.

SUMARIO

El presente trabajo es el desarrollo de una maqueta sobre un modelo de transición para acceder a redes comerciales soportados por IPv6 sobre nuestra actual Internet basada en IPv4.

Se presenta el trabajo con una introducción a la situación actual de la red y la necesidad de la migración y adecuar los servicios al cambio inminente, en el segundo capitulo se ingresa a la parte teórica acerca del protocolo, para luego en el tercer capitulo la implementacion de mismo para lo cual nos apoyamos en un estándar abierto que es el software libre el cual puede ser ejecutado por cualquier usuario domestico las pruebas realizadas pero que sirven como base para un ISP que desee apoyarse en este documento como migración.

Finalmente en el ultimo capitulo ejecutamos el core del trabajo presentado que es la gestión de trafico y una breve introducción a la migración de servicios.

Las configuraciones que se ejecutan para llegar a cumplir el objetivos requiere un nivel avanzado de Linux / Unix a fin de poder implementarlo con la sencillez con la que muestro en el presente trabajo.

Cabe recalcar que lo implementado se puede ejecutar en software propietario pero con los cambios de licencias , ya casi es lo mínimo que legalmente podríamos mostrar en pantalla a menos que copiemos temas de Internet que no es el motivo del trabajo, o con algún hardware propietario como Cisco o Junniper , pero el sentido del trabajo es mostrar la teoría pasos sencillos para su ejecución y aplicarlo a la investigación que es el principio de una universidad.

ÍNDICE

ÍNDICE DE ILUSTRACIONES

ÍNDICE DE TABLAS

ÍNDICE DE REFERENCIAS

(Soporte de graficos de Internet)

Rev 1 http://www.my.apan.net/ipv6

Rev 2 http://www.my.apan.net/ipv6/Papers

Rev 3 http://bgp.potaroo.net/ipv4/

Rev 4 http://www.nw.com

Rev 5 http://www.ripe.net/cgi-bin/ipv6allocs

Rev 6 http://staff.csc.fi/~psavola/ipv6/

Rev 7 http://www.optix.org/~dxy/solaris/ipv6/

Rev 8 http://staff.csc.fi/~psavola/ipv6/

Rev 9 http://www.ietf.org/rfc/

Rev10 http://en.wikipedia.com

PRÓLOGO

No cabe duda que día a día existe incremento de aplicaciones , servicios y lo que es mas incremento de hosts que ingresan a la red , no solo entendemos hosts como maquinas operadas por una persona , sino entendemos host como cualquier dispositivo ya sea desde un teléfono celular , un artefacto o hasta algún sensor que puede ubicarse físicamente en un lugar determinado o móvil , por lo que hace que una cambie nuestros paradigmas sobre una red local o conocida como red LAN y el concepto de una red WAN, estos paradigmas involucran replantear la red, pero una red que día a día apliquemos parches tiene un limite y ese limite es el protocolo que soporta a la red actual que ha cambiado el mismo concepto del mundo.

Estos cambios involucran el concepto de migración que se puede lograr de una manera escalonada o simplemente fijando un día CERO , el cual lógicamente sera imposible.

Desde ya ARIN (American Registry for Internet Numbers) ya no cuenta con rangos de direcciones IPv4 por lo que ya es un indicador de que estamos antes una migración evidente y paulatina.

Ya hace algunos años tenemos en el mercado procesadores con capacidad de proceso de 64 bits por lo que las condiciones externa ya están listas para poder adoptar y migrar muy a pesar que los ISP aplican NAT a todo y quieren salvar el mercado pero por el contrario los países europeos y asiáticos ya adoptan la migración en sus redes empleando el dual stack incluso China en un intento desesperado por asignación implantaron el IPv10 como estándar para su red lo cual provoco un aislamiento a Internet , lo cual dieron un paso atrás con Beijing en el cual aplicaron IPv6 para convivir con el mundo real , fue la primera vez que este nuevo protocolo es usado en un Mega evento mundial, mas de 1000 ingenieros estuvieron trabajando para

perfeccionar las funcionalidades, llevando a China a ser un espacio de alta tecnología con una velocidad de comunicaciones sin

precedentes.

Las aplicaciones que han trabajado sobre la plataforma IPv6 incluyen:

- Tecnología de banda ancha fija y móvil como GSM, EDGE, WIFI, WIMAX, TD-SCDMA y McWL.
- Televisión de alta definición, video streaming, Tv Móvil.

En el Perú INICTEL fue el primer promotor de la adopción de este protocolo , en el 2002 trabajé en el departamento de investigación del mismo el cual se elaboro una maqueta empleando la red de pruebas 6BONE y creando túneles para luego replicarlos entre las instituciones peruanas que lo soliciten brindando así el soporte necesario para su experiencia.

Posterior a esta experiencia llega le conexión a la red de CLARA y la formación de la primera red académica Peruana (RAAP) el cual es una red nativa en IPv6 y brinda una plataforma de experiencia real con IPv6 a las universidades conectadas a la misma via fibra óptica y equipos media converter a la mismas a través de telefónica , justamente mi experiencia aquí fue la de ser parte de la implementacion y configuración de la red MPLS en el operador mencionado.

Este trabajo esta resultando una actualización al trabajo que efectué en INICTEL para esta implementacion la ejecute desde mi servidor de mi casa la cual desde el 2003 ya corre en el dual Stack IPV6 y empleando Software Libre el cual te permite la libertad en configuración y manipulación de los programas y servicios y acondicionarlo a la situación que nosotros mismos queramos que trabaje.

CAPITULO I
SITUACION ACTUAL EN EL CONTEXTO IPv4 / IPv6

1.1 Contexto General

Dados el dinamismo de la tecnología y los requerimientos cambiantes en las empresas para tener una mayor cantidad de ancho de banda debido a las aplicaciones multimedia y aplicaciones de red que día a día deboran el ancho de banda de nuestros proveedores , IPv6 es crítico para que el futuro de las redes empresariales y las redes públicas de la Internet sigan creciendo. La creación así como el desarrollo e implementacion de IPv6 se ha llevado a cabo con mucha cautela y se han aplicado las lecciones aprendidas de los errores cometidos en IPv4.

El cambio de IPv4 a IPv6 se puede justificar de dos maneras o dos puntos de vista principalmente:

- Técnicamente en éste momento el sistema de direccionamiento ya no es suficiente para la gran cantidad de equipos conectados a la misma red, la demanda actual y futura no podrá ser satisfecha por la versión actual de IP, aunado a ésto, las tablas de enrutamiento actuales son demasiado grandes debido a la gran cantidad de direcciones que existen sin tener una autoconfiguración muy a pesar de la implementacion del NAT como solución temporal a ese déficit
- Socialmente las necesidades de los usuarios de la Internet están aumentado exponencialmente, exigiendo nuevas capacidades que la versión 4 no proporciona como lo son seguridad, privacidad, velocidad, VoIP, multimedia, teleconferencias y aplicaciones de gran demanda y capacidad para almacenamiento en redes SAN y NAS

Para contar con las bases suficientes para el cambio primero se debe de entender como se estructuró la Internet desde sus inicios, ver su crecimiento, ver las versiones de IP, recalcar los beneficios de IPv6 y ver porque se debe de cambiar a la nueva versión de IP.

1.2 Un poco de Historia de Internet.

La Internet como la conocemos actualmente es un medio de comunicación que ha revolucionado el mundo tanto de las comunicaciones (teléfonos, radio, prensa, televisión, etc) , como de las computadoras. Las bases que ayudaron a su desarrollo, si vemos desde los inicios más básicos, son desde el telégrafo hasta las computadoras personales pasando por el teléfono IP los streams de radio y TV, también debido a la cantidad de información que se maneja y transfiere actualmente en la Internet, es un recurso que revolucionó la manera de investigar y tener acceso a información mundial en segundos.

La Internet comenzó siendo una idea de J.C.R. Licklider, quien en algunos memorandums en Agosto de 1962 en el MIT (Massachussets Institute of Technology), describía la idea de computadoras interconectadas entre si para tener acceso a la información de las mismas, él la describía como una Galactic Network o Red Galáctica. Por éstas ideas radicales en ese momento, a Licklider lo designaron director del programa encargado de desarrollar la DARPA (Defense Advanced Research Proyects Agency). Conjuntamente con Licklider trabajaban Ivan Sutherland, Bob Taylor y Lawrence G. Roberts.

La primera WAN (Wide Area Network) documentada fue la creada en 1965 por Lawrence G. Roberts y Thomas Merrill, quienes conectaron una TX-2 y un Q-32 desde el MIT en Massachussets hasta California mediante una línea telefónica. Para 1967 se había avanzado en el diseño de las redes, también Inglaterra aparte de los Estados Unidos había estado investigando por medio de otros grupos de trabajo como el RAND y el NPL. En el DARPA, acordaron interconectar todos sus centros de investigación por medio de una red a la que denominaron ARPANET (Pre-Internet).

Para finales de 1969 la ARPANET comenzaba a tomar forma, primero conectando el primer nodo en UCLA siendo éste el Network Measurement Center, después en el Stanford Research Institute instalando e interconectando el Network Information Center, posteriormente en la Universidad de California en Santa Barbara el Culler-Fried Interactive Mathematics, y finalmente ese mismo año, en Utah el Graphics.. Ese mismo año debido a que la red estaba en periodo de pruebas, se debía de tener un flujo de la comunicación de la información e ideas entre los grupos de investigación, por lo que se crearon un tipo de memorandums, los RFC (Request For Comments), cada RFC hace referencia a un protocolo o tipo de comunicación referente a la Internet.

Para 1970 se publica el primer protocolo de comunicación entre dos computadoras (host-to-host) de la ARPANET, el NCP (Network Control Protocol), su publicación corrió a

cargo del NWG (Network Working Group). Los avances en la ARPANET continuaban, para 1972, ya se tenía desarrollada la idea de la arquitectura abierta (Open Architecture) que significa que la red no dependa en si de ningún servidor o red en particular, lo cual permite una red de nodos heterogéneos en una red homogénea, ésto fue presentado en la "International Computer Communicaton Conference" (ICCC).

A la par del desarrollo de la ARPANET se desarrolló un programa capaz de leer y escribir mensajes a otras personas utilizando la red, Ray Tomlinson crea la primera versión de un correo electrónico, en versiones posteriores del e-mail se adoptaría el @ (at sign en inglés, arroba en español) como separador en las direcciones de correo electrónico, el reenvío de mensajes, el manejo de archivos po r medio de FTP (File Transfer Protocol), etc.

La capacidad de enviar y recibir mensajes rápidamente, el intercambio de información y el acceso a los RFC, se tornó en el éxito de la ARPANET ya que agilizó el contacto entre los grupos que trabajaban en el proyecto.

Muchas universidades quisieron adherirse a la ARPANET, mientras más crecía, se empezaron a hacer visibles las limitaciones que en ese momento tenía el protocolo NCP ya que se basaba en la propia comunicación de la ARPANET. No había un método de control de paquetes ni errores entre las computadoras que intercambiaban información y en un principio no se pensaba que se iban a conectar demasiadas computadoras a la red por lo que se asignaron pocos números para asignar los nodos. NCP no contaba con un sistema de direccionamiento dinámico (Por ejemplo DHCP actualmente).

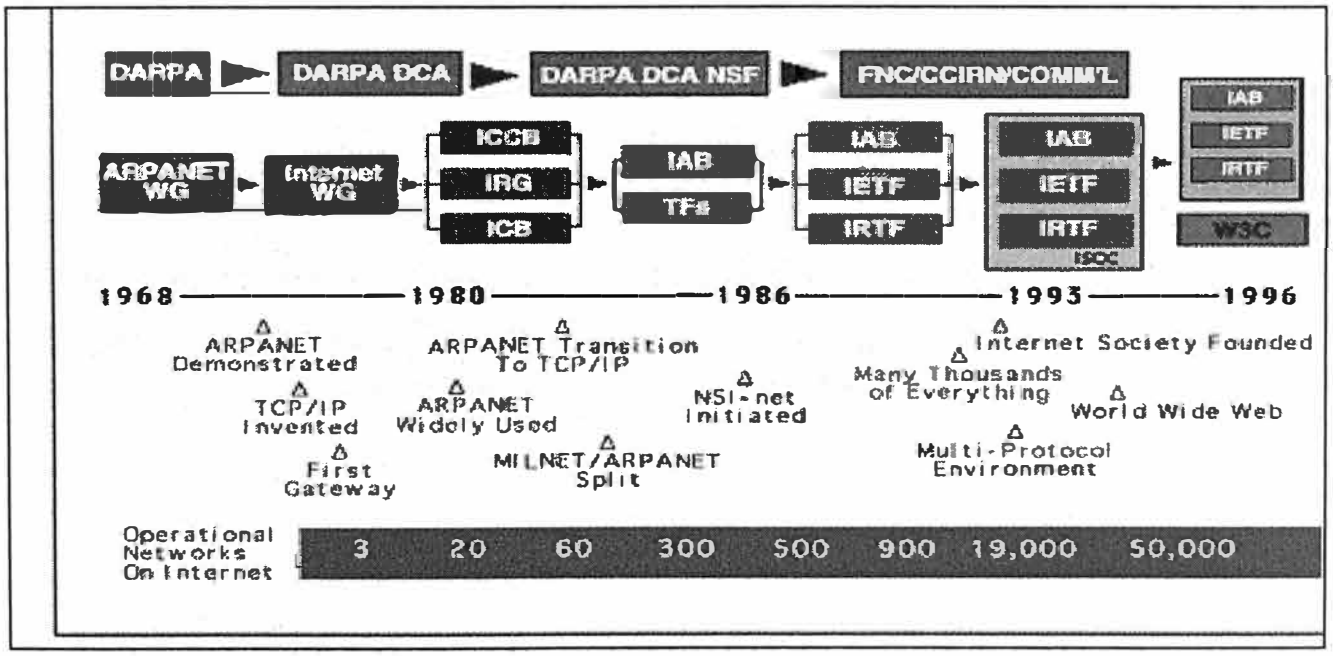


Fig. 1.1 Evolución de Arpanet / Internet (Ref 1)

Debido a esto hubo la necesidad de crear un nuevo protocolo de comunicación entre computadoras, siguiendo con el concepto de Open Architecture, se evolucionó el protocolo NCP a TCP/IP (Transmission Control Protocol / Internet Protocol) con los siguientes principios:

1. Cada una de las redes conectadas debe ser independiente del resto, y no deberán realizarse cambios en estas para su conexión a ARPANET.
2. Si un paquete no alcanza su destino, deberá ser retransmitido por el origen.
3. Se utilizarán unas cajas negras para la interconexión de redes (más tarde se les denominó gateways o ruteadores) que no mantendrán información referente a cada una de las conexiones que se estén produciendo en cada momento, permitiendo una cierta tolerancia a fallos. Estas cajas negras tendrán la función de conducir los paquetes hacia los nodos de destino, lo que implica un direccionamiento dentro de la red.
4. Se deben permitir simultáneamente diferentes comunicaciones entre las computadoras (pipelining) facilitando la interactividad. Ésto posibilita la existencia de varias conexiones simultáneas en un mismo equipo y obliga a la adopción de un sistema para diferenciarlas.
5. Es necesario un sistema de direccionamiento global para todos los nodos que forman parte de la red.

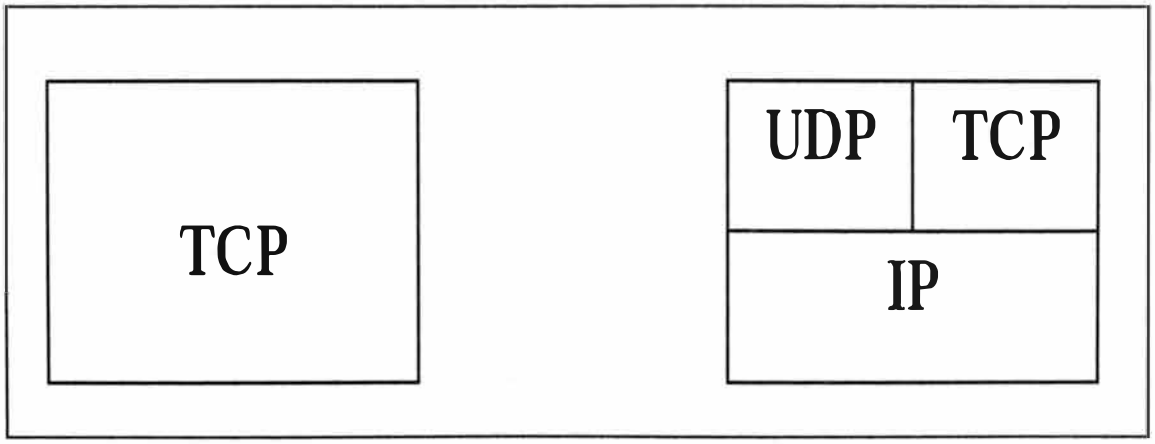


Fig. 1.2 Modelo del protocolo TCP/IP

Tras varios estudios, se asignaron roles a los protocolos TCP/IP, el protocolo IP solamente se encargaría de enviar paquetes a través de la red hacia el destino. Para el control de flujo o asegurar que los paquetes lleguen al destino se tienen 2 protocolos, el TCP y el UDP

(User Datagram Protocol), en esencia son el mismo pero el segundo no asegura que todos los paquetes lleguen a su destino, es decir, no es confiable. Los diseños de las capas que utiliza TCP/IP se pueden ver en las figuras 1.2 respectivamente, siendo el primero el diseño general de TCP y el segundo la estructura basada en jerarquías que utiliza TCP/IP.

Los grupos encargados para el desarrollo del nuevo protocolo se encontraban en las universidades de Stanford y UCLA, también se incluyó a la empresa Bolt, Beraneck and Newman (BBN), designados por la DARPA.

La introducción de las computadoras personales en los 80's y el desarrollo de las redes locales como Ethernet influyeron al desarrollo de la ARPANET, como se introducían mas y mas redes a la ya tan mencionada ARPANET fue necesario el replanteamiento del direccionamiento en las redes tanto locales como globales (LAN y WAN), dando como resultado el direccionamiento que se conoce actualmente.

Cabe mencionar que Ethernet fue desarrollada por Bob Metcalfe en los laboratorios XEROX PARC (Palo Alto Research Center) en 1973. "Ether" se supone según los griegos que es la materia presente en todas partes y "net" es red en inglés, por lo tanto la Ethernet se suponía debería de ser la red existente en todas partes.

La figura 1.3 muestra el esquema pensado por Bod para el diseño del protocolo.

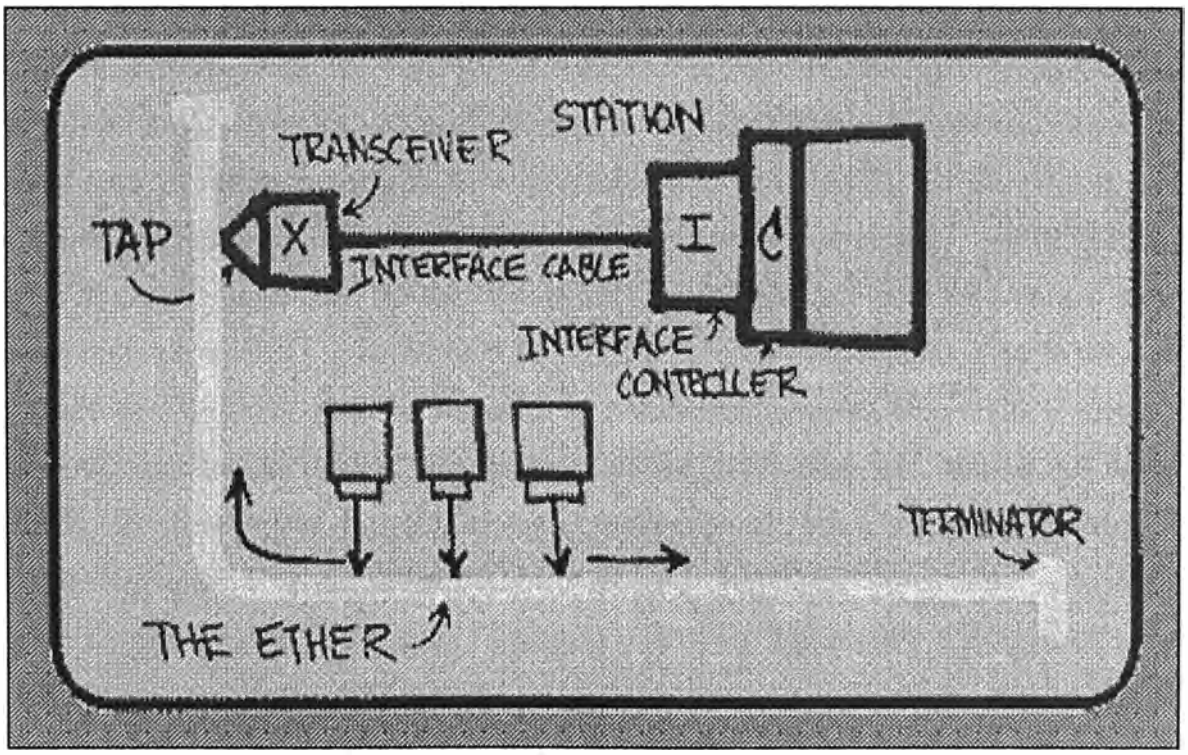


Fig. 1.3 Diseño del modelo Ethernet (Ref 2)

Cuando el Departamento de Defensa estadounidense adoptó los protocolos TCP/IP en 1980, dicho departamento se integró a la topología de ARPANET, creando dos redes, la MILNET, para propósitos militares y la ARPANET para propósitos científicos.

A principios de 1983 se substituyó el protocolo NCP por el protocolo TCP/IP en las dos redes y para 1985 también se suma la red NSFNET a ARPANET, dicha red era la red del NFS (Nacional Science Fundation) haciendo crecer más aún la ARPANET. El crecimiento de la ARPANET, también empezó a interesar a las empresas con redes locales (PSI,UUNET, ANS CO+RE), debido a ésto las personas responsables del NSFNET empezaron una campaña de limitar dicha red solamente a propósitos científicos, lo cual impulsa una privatización controlada de la misma. Por la gran demanda y el control de la integración de redes particulares a la red ARPANET, se determina terminar o dar de baja la ARPANET y dar de alta la Internet en 1990.

Internet crece aún más cuando el CERN (Organización Europea de Investigación Nuclear) en 1991 da a conocer el lenguaje HTML (Hyper Text Markup Languaje), que permite de una manera sencilla compartir texto e imágenes en las computadoras locales para que sean accesados por medio de Internet desde cualquier otra máquina conectada en la red. El lenguaje HTML es la base del WWW (World Wide Web). El correo electrónico también evoluciona añadiendo capacidades de seguridad y privacidad.

En 1994 el NRCC (National Research Council Comitee) en colaboración con la NSF publica el informe "Towards a National Research Network". Este informe impulsó y sentó las bases para las futuras autopistas de la información.

Finalmente, en 1995 culmina la política de privatización impulsada por NSF, lo que provoca la disolución de NSFNET en sub-redes locales, gracias a esto en 8 años y medio se pasó de tener 8 nodos conectados a 56 Kbps a 21 nodos conectados a 45 Mbps, con ésto finalmente Internet se hizo accesible a prácticamente todo el mundo y elevó su crecimiento exponencialmente como se puede ver en las figuras 1.4 , el cual describe la evolución del crecimiento de los hosts en comparación al incremento en el tiempo , se puede observar que el crecimiento de los hosts se considera desde el fecha en que este adquiere un nivel considerable o de referencia que es en el año 1994 y a partir de ahi su crecimiento se traduce en exponencial.

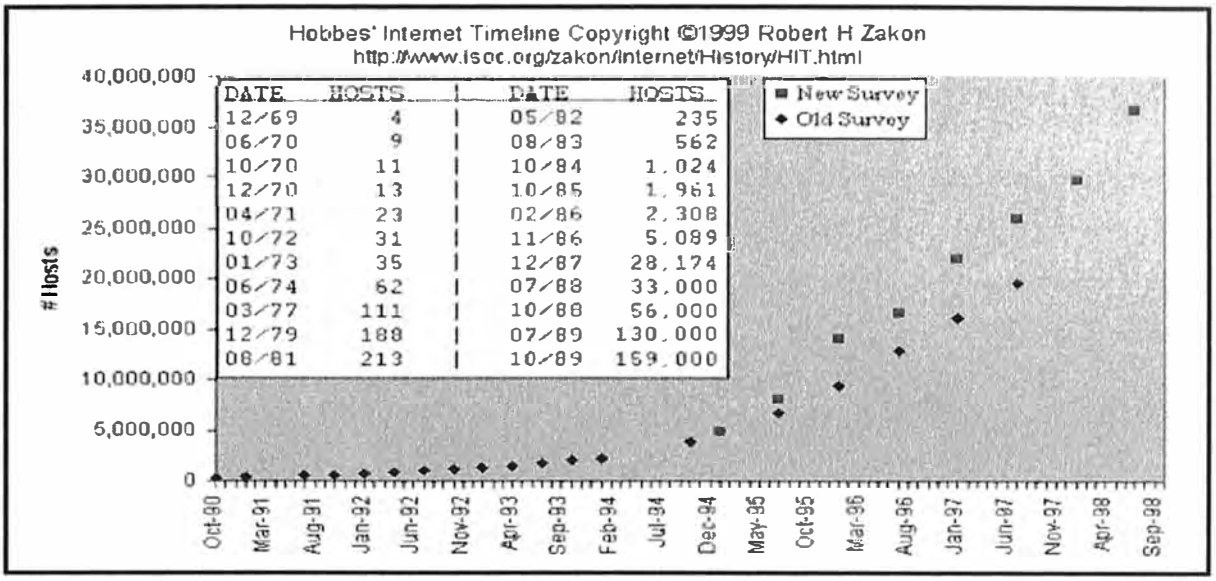


Fig. 1.4 Crecimiento de host desde 1990 (Ref 3)

1.3 Protocolos TCP/IP

Para comunicarse, Internet utiliza los protocolos TCP/IP, los protocolos consisten en un esquema de capas las cuales se comunican únicamente con el protocolo situado inmediatamente superior o inferior, sea el caso de recepción o transmisión.

Cada capa tiene una tarea específica para poder comunicar diferentes tipos de computadoras, independientemente de la red en la que estén, el fabricante de los equipos de cómputo o el sistema operativo que utilicen. El protocolo TCP/IP utiliza cuatro capas, las cuales se pueden ver en la figura 1.5

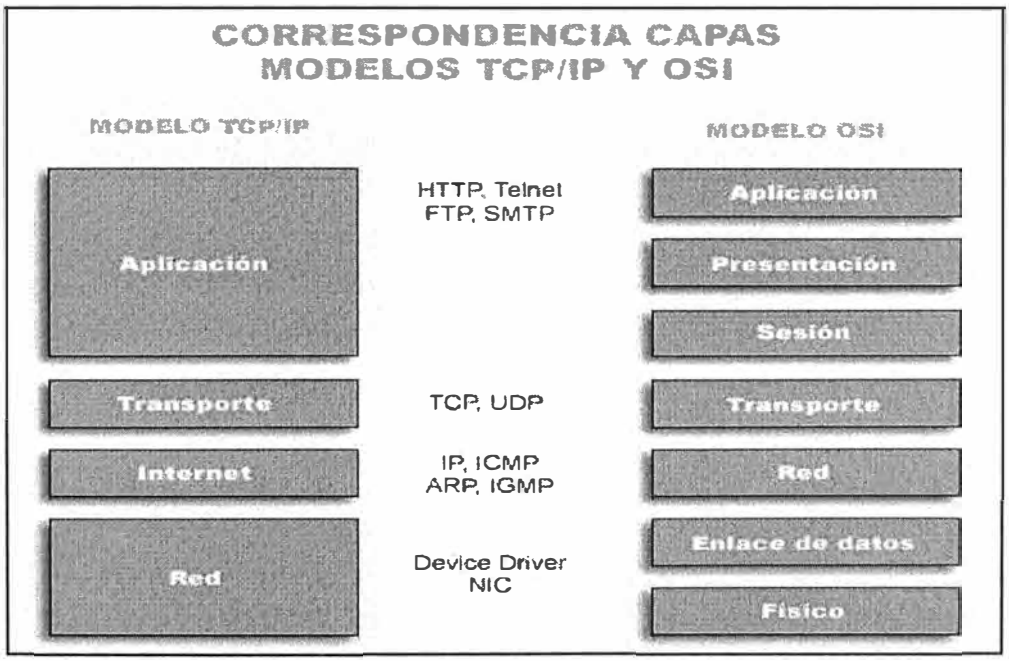


Fig. 1.5: Estructura 4 capas vs Referencia OSI (Ref 4)

- La capa de aplicación hace la interconexión con las aplicaciones (valga la redundancia) que se van a utilizar en la red. (WWW, Telnet, FTP, etc...)
- La capa de transporte tiene como ocupación el dar el flujo de información a las computadoras en la red, ya sea confiable (TCP) o no confiable (UDP).
- La capa de enlace o Internet es la encargada de que los sistemas operativos puedan enviar y recibir información, ésta capa también se denomina capa de datos o capa de acceso a red, aquí trabaja el protocolo IP.
- La capa de red se encarga de mover los paquetes a través del medio físico que puede ser transmisión por medio de cobre , óptico, inalámbrico , etc.

Las capas del modelo hacen que la comunicación entre dos computadoras sea por medio de las capas, haciendo cada capa independiente a la otra y así facilitando cambios o mejoras en los modelos. Cada capa en el modelo añade información en el encabezado de los paquetes enviados, es decir, en la encabezado se almacena la información del tipo de protocolo, el numero de paquete a quien va dirigido, de quien viene, etc.
Cuando se envía información en el encabezado se pone la información de cada capa y se pasa a la capa inferior hasta que se llega a la capa inferior quien envía los datos a través de la red, cuando se recibe información pasa exactamente lo inverso como se puede ver en la figura 1.6.

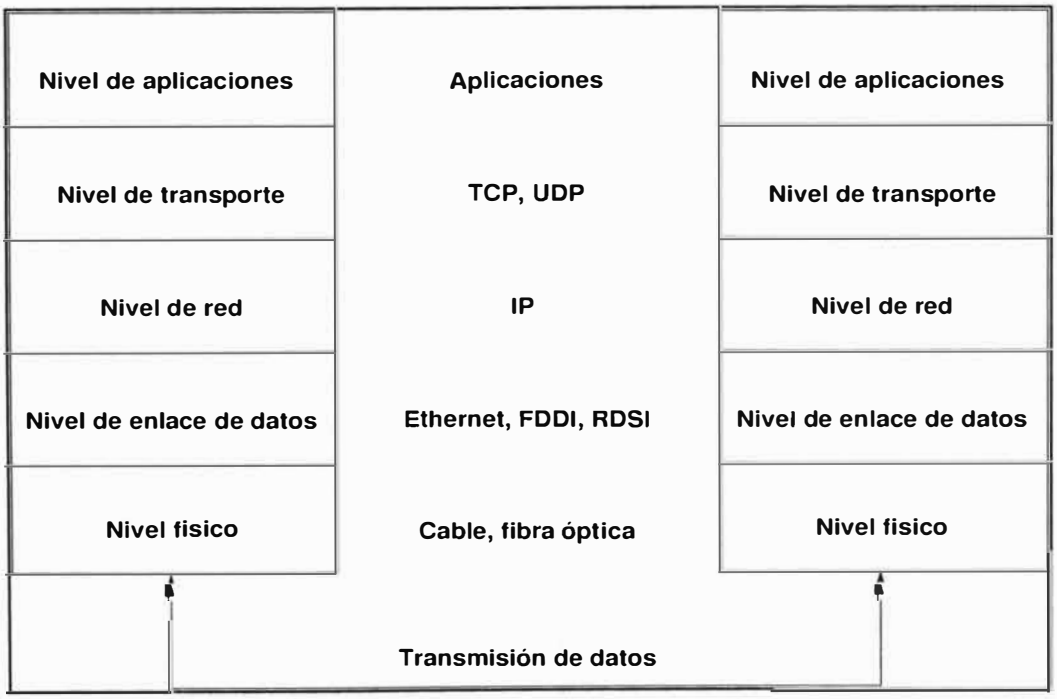


Fig. 1.6: Esquema por capas

Para que cada equipo se comunique con otro en Internet, el modelo genera paquetes los cuales viajan a través de la red por los equipos conectados que son los encargados de hacer la conmutación de paquetes hasta llegar a su destino, existen dos tipos de equipos conectados a Internet, los que envían paquetes de información (hosts) y los que hacen la conmutación para que los paquetes generados lleguen a su destino (router, gateway, bridge, etc.).

1.4 Escenarios de transición.

Existen dos escenarios de transición:

1. Apagar la red y volverla a encender con hardware y software compatible con IPv6.
2. Un cambio lento y gradual.

Considerando el tamaño actual de Internet, la primera opción no es considerable ya que la coordinación requerida para apagar la red es imposible de alcanzar y el esfuerzo administrativo para migrar todo el hardware y software antes del encendido tampoco es factible, para éste efecto se tendría que apagar la red días, más bien semanas o meses. La diversa variedad de plataformas de hardware y software también lo hacen ser una tarea imposible.

El escenario más viable es el segundo, una transición lenta y gradual en la cual IPv4 e IPv6 coexistan lateralmente, es posible que la transición dure varios años antes de completarse.

La traducción de encabezados IP, representa un problema serio, se ha convenido que en la práctica será muy difícil de alcanzar un tiempo real de traducción corto, por lo tanto existen dos posibilidades. La primera, que los hosts y los ruteadores empleen un dual-stack que sea capaz de manejar datagramas tanto IPv4 como IPv6. La segunda es hacer redes de túneles de IPv6 sobre IPv4, hasta que la transición se haya completado que es precisamente el motivo de esta trabajo.

Los mecanismos actuales de transición que están integrados en el diseño de IPv6 incluyen hosts y ruteadores con un dual-stack IPv4/IPv6, tunneling de IPv6 vía IPv4 y un número de servicios de IPv6 incluyendo DNS, DHCP y demás para IPv6 serán tratados mas adelante..

1.5 Justificación para el cambio a Ipv6

Existen varias consideraciones importantes a cerca de la transición. Cuando se adoptó IPv4

los hosts se forzaron a cambiar a IPv4 a principios de 1983, no fue hasta 1984 que el número de hosts conectados al backbone rebasaron los 1000, se debe de recordar que en ese tiempo, la mayoría de los usuarios de la Internet eran expertos en el campo.

Para mediados de 1999 existían 60,000,000 de hosts fijos registrados en el DNS, y un número dinámico no especificado de hosts asignados dinámicamente, desde entonces el número creciente de hosts en Internet ha rebasado la imaginación. De estos usuarios, la mayoría tiene muy poco o nulo conocimiento de las telecomunicaciones, las computadoras, las redes, protocolos y demás ámbitos técnicos. La transición a IPv6 no será como nada visto anteriormente, es un problema de gran escala.

El amplio y flexible número de direcciones en IPv6 habilita la definición de arquitecturas de ruteo globales, flexibles y jerárquicas, con varios niveles. Una arquitectura jerárquica de direcciones IPv6, se puede asignar a áreas geográficas utilizando los prefijos flexibles tipo CIDR (Classless Inter-Domain Routing). El direccionamiento IPv6 puede ser enfocado de una manera que facilite la sumarización del ruteo y controle la expansión de las tablas de ruteo en los ruteadores de backbone.

También implica que los proveedores de Internet tendrán suficientes direcciones para asignar a empresas medianas y a usuarios de dial-up y/o ADSL o en redes Wimax que necesitan direccionamientos globales para que así puedan explotar al máximo la Internet.

En lo concerniente a la telefonía, el direccionamiento de IPv6 permite a la industria conectada en red, ir más allá del sistema telefónico actual con el advenimiento de la 3G y los equipos IPhone que soportan UMTS y EDGE pronto la integración en una sola red se tendra que dar si o si.

El cambio implica varias consideraciones, la reducción de trabajo, la seguridad, el factor humano, el hardware y software y sobre todo, la simplicidad y estandarización del formato de IPv6. Para ver cada uno más a fondo, se decribirá cada uno brevemente.

1.5.1.- Reducción en la administración de direcciones

Las redes IPv6 pueden muy bien emplear direccionamiento dinámico (DHCP) para reducir el esfuerzo asociado con asignar manualmente direcciones a estaciones de trabajo, DHCP es una herramienta que permite una configuración estática de direcciones ya que mantiene las tablas que determinan las direcciones a asignar estáticas, ya sea a estaciones de trabajo ya existentes que se mueven de lugar o nuevas estaciones de trabajo.

IPv6 también provee una nueva dimensión a la autoconfiguración mediante el servicio de autoconfiguración independientemente de su lugar, el cual no requiere servidores configurados manualmente. La autoconfiguración independiente de lugar hace posible que las estaciones de trabajo configuren su propia dirección con ayuda de un ruteador o gateway Ipv6 local. Típicamente la estación de trabajo combina su MAC Address de 48 bits con un prefijo de red que aprende del ruteador del vecindario este tema lo veremos mas adelante en nuestra implementacion del gateway el cual nos muestra una autoconfiguración Plug and Play .

Las capacidades robustas de autoconfiguración en IPv6 serán de gran ayuda para los usuarios cuando una empresa sea forzada a cambiar el direccionamiento debido a un cambio en el proveedor de Internet, la autoconfiguración de IPv6 permitirá a los hosts recibir nuevos prefijos sin la necesidad de reconfigurar manualmente las estaciones de trabajo o el direccionamiento DHCP.

Esta función también es muy útil en menor escala en empresas que tiene problemas en el seguimiento de los movimientos de los usuarios dinámicos. La autoconfiguración es muy importante para habilitar el cómputo móvil ya que permite a las computadoras móviles recibir direcciones IP válidas automáticamente independientemente de donde estén conectadas a la red.

1.5.2.- Un formato simplificado de IPv6

Las direcciones IP en IPv6 es cuatro veces más grande que en IPv4, sin embargo, como resultado de la simplificación y la mejora del mismo, es solamente dos veces más largo, el largo del encabezado se espera que no afecte el desempeño gracias a su simplificación y estandarización. Más allá del formato simplificado y estandarizado, IPv6 fue mejorado por medio de los encabezados de extensión, lo cual cambia la manera de manejar las opciones en los encabezados y mejora su ruteo y manejo.

Los encabezados de extensión opcionales se localizan después del encabezado IPv6 y antes de los datos en cada paquete. La mayoría de los encabezados de extensión no son examinados por los ruteadores en el camino, lo que si sucedía en IPv4. Los encabezados de extensión ahora son de longitud variable y tienen menos restricciones en su tamaño.

IPv6 proporciona a los diseñadores de redes una manera muy sencilla de introducir más encabezados de extensión en un futuro, los campos de opciones se han definido por cargar información explicita de ruteo, creada en el nodo de origen, para facilitar el control de la

autenticación, codificado y fragmentación, a nivel de aplicación, los encabezados de extensión se pueden utilizar para aplicaciones punto a punto que requieran sus propios campos dentro del paquete IP.

1.5.3.- Seguridad

La coexistencia del direccionamiento IPv4 e IPv6 provee problemas de accesibilidad para las aplicaciones que no están diseñadas para IPv6, los hosts que tengan corriendo esas aplicaciones requerirán de sistemas de soporte para la traducción de direcciones, por razones de desempeño, el DNS se supone que debe de retornar ambas direcciones, tanto la IPv4 como la IPv6, lo que puede confundir a los hosts no adaptados para el proceso de direcciones IPv6.

El soporte del dual-stack no es un tema de seguridad, sin embargo la implementación del mismo se vuelve un problema de seguridad. IPv6 maneja un stack más grande y si es transportado sobre IPv4, se pueden introducir datos falsos en el transporte por medio de IPv4, por lo tanto la combinación de los dos protocolos no asegura que las fallas en la seguridad de IPv4 sean eliminadas. Se espera que solamente al migrar a redes solo IPv6, se eliminarán las fallas de seguridad.

1.5.4.- El tema humano

La gente se debe dar cuenta de la necesidad de la transición para que se lleve a cabo. Como no existe una autoridad centralizada que fuerce a todos a hacer el cambio, depende de cada sitio individual el empezar a hacer el cambio.

El cambio necesita convencer a las autoridades dueñas de los sitios que el cambio es necesario, después capacitar el personal para que puedan ser capaces de sobrellevar la transición, y después calendarizar la transición dependiendo de los calendarios de actividades de las empresas.

Debido a la poca experiencia con IPv6, muchos sitios pueden ser susceptibles a errores de configuración en sus redes, las herramientas de administración automatizadas pueden ayudar a los administradores a hacer las configuraciones y las tareas de mantenimiento, sin embargo, no les ayudarán a entender la estructura subyacente del protocolo.

1.5.5.- Hardware y software

La nueva tecnología requiere de nuevo hardware y software para soportarla, durante la

transición varios sitios puede que corran software que esté por lo menos parcialmente en desarrollo o en etapas experimentales. Las fallas en el software y hardware serán muy comunes debido a la premura de lanzar nuevos productos.

Los problemas asociados al software y hardware deben de ser sobrellevados sin embargo por la lentitud de la transición. Si el desarrollo de IPv6 empieza en ambientes cerrados, protegidos por firewalls, entonces, no debe de haber problemas.

Tabla 1.1 Resumen de las consideraciones para la migración

IPv4	**IPv6**
Las direcciones origen y destino tienes 32 bits (4 bytes)	Las direcciones origen y destino tienes 128 bits (16 bytes)
La compatibilidad con IPSec es opcional	La compatibilidad con IPSec es obligatoria
No hay identificación de carga para el control de QoS por parte de los ruteadores en el encabezado IPv4, aunque esta el campo Tipo de Servicio no esta implementado	La identificación de cargar para el control de QoS por parte de los ruteadores se incluye en el encabezado IPv6 mediante campos de etiqueta de flujo y clase de trafico.
La fragmentación es posible en los ruteadores y en el host de envió	La fragmentación solo es posible en sl host de envió mas no en los ruteadores
El encabezado incluye una suma de comprobación	No existe una suma de comprobación por que otros mecanismo de encapsulación ya realizan esa función
El encabezado incluye opciones	Todos los datos opcionales se mueven a extensiones del encabezado IPv6
El protocolo de resolución de direcciones (ARP) utiliza tramas de solicitud ARP de difusión para resolver direcciones IPv4 en una dirección de nivel de enlace	Las tramas de solicitud ARP se reemplazan por mensajes Neighbor Solicitation (Solicitud de vecino) de multidifusión
Se utiliza el protocolo de administración de grupos de Internet IGMP para administrar la pertenencia a grupos de subredes locales	El protocolo IGMP se reemplaza por mensajes Multicast Listener Discovery MLD (descubrimiento de escucha de multidifusión)
Para determinar la dirección IPv4 de la mejor puerta de enlace predeterminada se utiliza el descubrimiento de enrutadores ICMP que es opcional .	Para determinar la direccion IPv6 de la mejor puerta de enlace predeterminada se utiliza mensajes Router Solicitation (Solicitud de enrutador) y Router Advertisement (Anuncio de enrutador) de ICMv6 que son necesarios
Las direcciones de difusión se utilizan para enviar trafico a todos los nodos de la subred	No hay direcciones de difusión en IPv6 en su lugar se emplean direcciones de multidifusión
La asignación de dirección puede ser manual o mediante dhcp	La asignación de dirección puede ser manual, automática o mediante dhcp.
Los DNS emplean registros IN A para asignar nombre a host IPv4	Los DNS emplean registros IN AAAA para asignar nombres a hosts en IPv6.

CAPITULO II

CONCEPTOS DEL PROTOCOLO IPV4 Y SU TRANSICIÓN A IPV6

El presente capitulo mostrara la base teórica del protocolo IP (Capa 3 OSI) que es el que justamente se esta tratando en el presente trabajo y es el motivo del transición expuesta en el capitulo I , primero mostraremos la base teórica de IP versión 4 y de ahí entraremos a fondo en IP versión 6, como muestra el grafico 1.7.

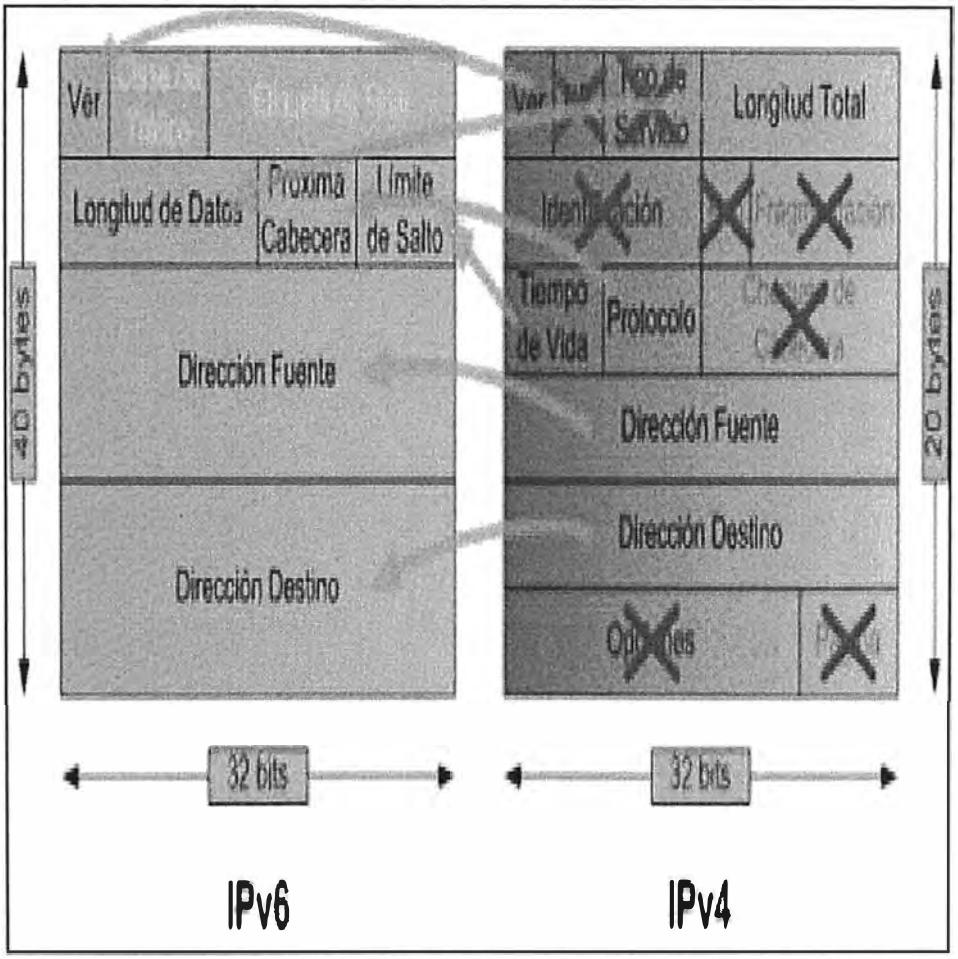


Fig 2.1 IPv4 vs IPv6 (Ref 5)

El datagrama IPv6 es mas eficiente por que posee un menor números de campos (Eficiencia de encaminamiento) , extensibilidad de la cabecera, alienación a 64 bits (por lo cual portabilidad a arquitectura de 64 bits) y etiquetas de flujo para redes que soportan QoS

Ademas de las características obligatorias de seguridad (autentificación y encriptación integradas) , movilidad optimizada, multicast (no broadcast) que conlleva a redes mas eficiente y escalables , todos esos factores mencionados permitirán un despliegue y soporte para los nuevos servicios que urge actualmente .

2.1 El protocolo IP versión 4

El protocolo IP (Internet Protocol) es en el cual se basa la transmisión de datos actual en Internet, su definición se encuentra en el RFC 791, su base es la transmisión de datagramas a través de la Internet, lo cual hace por medio de un sistema conection less y unreliable y da el servicio best effort, TCP provee las características de confiabilidad y de conexión e IP le delega ese trabajo para evitar volverlo a hacer, los protocolos trabajan en conjunto pero cada uno haciendo lo que es necesario para que los datos lleguen con seguridad a su destino sin tener que ser enviados todos los datagramas por el mismo camino.

La capacidad de best effort de IP funciona de manera que si existe una falla en el enlace por el cual se están transmitiendo los datos, se tengan caminos alternos por los cuales se pueda transmitir la información por medio de un sistema muy sencillo de solución de errores.

El mecanismo de control de errores es controlado por el Internet Control Message Protocol (ICMP), por ejemplo, si a un ruteador le falla un enlace por el cual estaba transmitiendo los datos, elimina el datagrama y manda un mensaje de ICMP al equipo que está enviando los datos y se olvida del datagrama, no trata de retransmitirlo, el equipo que estaba transmitiendo, retransmite el datagrama, no teniendo la información de cual enlace está activo o no.

Cuando el datagrama llega al ruteador él verá la manera de hacerlo llegar a su destino por otro enlace, lo que nos refleja éste tipo de servicio es que no implica fiabilidad (unreliable) y no conexión (conection less) por un camino específico.

2.1.1.- Estructura del datagrama IPv4

La estructura de un datagrama IP, se divide en bloques de 32 bits (4 bytes), comenzando de izquierda a derecha y de arriba hacia abajo, el primer bit es el bit 0, el orden es importante ya que dependiendo del equipo al que se está comunicando es su manera de guardar los bits en memoria, ver figura 2.2. A ésta manera de transmitir los bits se le denomina network byte order.

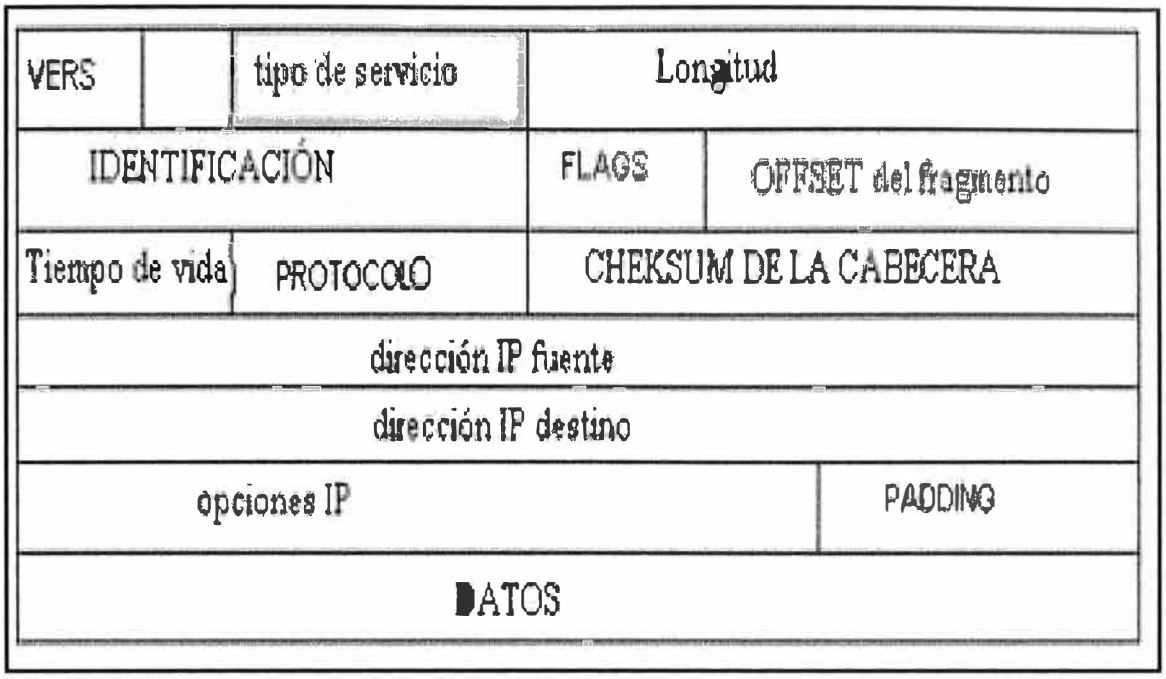


Fig. 2.2: Estructura de un datagrama IPv4 (Ref 6)

Los datos del encabezado son importantes ya que son la manera de dar a conocer al ruteador o al otro host lo que se está enviando. Para tener más claros los campos se detalla su contenido a continuación.

El campo de versión especifica que formato de versión es el datagrama, ésta información solamente lo utilizan los ruteadores y la capa IP de la conexión, permite que coexistan varias versiones de IP en las diferentes redes conectadas a la Internet sin que el usuario sepa de su existencia.

El campo de tamaño del encabezado indica el número de palabras de 32 bits que ocupa el encabezado, estos 4 bits limitan el tamaño de la encabezado a 60 bytes, sin embargo por lo regular se ocupan 20 bytes.

El campo de tipo de servicio son 8 bits, los primeros 3 no se usan, los siguientes 4 definen el tipo de servicio, el cual se detalla en la tabla 2.2 y el último bit no se utiliza pero debe de tener valor de 0 siempre, en los bits de tipo de servicio, solamente uno puede estar activo a la vez.

El tipo de servicio se tiene para darle a entender al ruteador la política de servicio que se debe de tener con el datagrama, minimizar el retrazo, maximizar el rendimiento, maximizar la fiabilidad del transporte y minimizar el costo económico del transporte.

Tabla. 2.1: Valores típicos del tipo de servicio según la aplicación

Tipo de Aplicación	Minimizar retrazo	Maximizar rendimiento	Maximizar fiabilidad	Minimizar costo	Valor Hexadecimal
TELNET	1	0	0	0	0x10
FTP	0	1	0	0	0x08
SMTP	0	1	0	0	0x08
DNS (UDP)	1	0	0	0	0x10
DNS (TCP)	0	0	0	0	0x00
ICMP	0	0	0	0	0x00
BOOTP	0	0	0	0	0x00

El campo de longitud del datagrama mide 16 bits y dice cuanto espacio se debe guardar en la memoria para la recepción de cada datagrama, también dice cuantos bytes se deben leer por datagrama, con esto se puede tener un control muy sencillo de si los datagramas llegan completos o no, también limita el tamaño máximo de los datagramas a 65515 bytes, el Maximum Tranfer Unit (MTU) es 216 bytes 65525 – 20 bytes de encabezado.

En dado caso que un paquete que se quiera enviar por la red excede el máximo disponible para dicha red, se divide en varios pedazos. (fragmentación).

El campo de número de identificación del datagrama indica el número de paquete que se esta recibiendo o enviando cuando se tiene que dividir en pedazos un paquete, así cuando se recibe el paquete se puede ordenar adecuadamente, mide 16 bits, por lo que un datagrama se puede dividir hasta en 65535 fragmentos.

El campo de banderas mide 3 bits y especifica diferentes actividades según el bit que esté encendido, si el primero está encendido quiere decir que el datagrama es parte de un datagrama mayor, si el segundo está encendido quiere decir que el datagrama no debe de ser fragmentado y el tercero no se utiliza, teniendo siempre el valor 0.

El campo de número de byte en el datagrama, indica cual es la posición en bytes que ocupan los datos en el datagrama original, obviamente solo se ocupa si el fragmento es parte de un paquete mayor, mide 13 bits y sirve para reconstruir el paquete original.

El campo de tiempo de vida mide 8 bits y es el que indica cuanto tiempo vivirá el datagrama en transición, es decir, cuanto tiempo tiene el datagrama para llegar a su destino

para que los datagramas no circulen para siempre por la red, éste campo tiene un valor máximo de 255 y cada vez que pasa por un ruteador su valor se decrementa en uno, si el valor llega a cero, el ruteador que le toca proporcionar ese valor, envía un ICMP al origen para que el datagrama sea retransmitido.

El campo de tipo de protocolo indica el protocolo superior que se esta utilizando, ya sea TCP, UDP, ICMP, etc. El campo se ocupa ya que todos los protocolos de Internet utilizan IP como medio de transporte y al llegar al destino hay que entregarlo en los medios adecuados.

El campo de suma de comprobación ("checksum") del encabezado del datagrama se utiliza solo para verificar el encabezado, ya que tanto UTP, TCP y demás protocolos tienen su propio "checksum" y verificarán sus datos de manera autónoma, sirve para verificar que el encabezado llegue completo y no se descarte el datagrama por pérdida de información en el camino.

2.1.2.- Direccionamiento IPv4

La dirección IP origen y la dirección IP destino son dos números de 32 bits, cada una. Cada equipo tiene un número específico, dentro del protocolo IPv4 se denominan 4 octetos de 8 bits separados por un punto para especificar cada equipo en la red. Existen varios tipos de redes los cuales se describen en la tabla 2.2:

Tabla 2.2: Clases de direcciones IPv4 en Internet

CLASE	DESDE	HASTA
A	0.0.0.0	127.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255
D	224.0.0.0	239.255.255.255

E	240.0.0.0	247.255.255.255

Las clases de redes sirven para definir el tamaño de las redes, como se vió en la figura anterior existen 5 clases de redes, en la figura 2.4 se puede ver la cantidad de equipos que se pueden conectar a cada red:

Tabla. 2.3: Subdivisión de los 32 bits para las clases A, B, C, D Y E

Clase A	0	Identificador de red (7 bits)																												Número de equipo (24 bits)	
Clase B	1	0	Identificador de red (14 bits)																											Número de equipo (16 bits)	
Clase C	1	1	0	Identificador de red (21 bits)																										Número de equipo (8 bits)	
Clase D	1	1	1	0	Identificador de red (28 bits)																										
Clase E	1	1	1	1	0	Reservado para uso futuro (27 bits)																									

Cabe mencionar que el número máximo en cada octeto es 255 ya que al ser de 8 bits (28=256) el rango es entre 0 y 255 para cada octeto.

Se definieron los diferentes tipos de redes para hacer más fácil la ubicación de redes chicas, medianas y grandes, es decir:

- La red clase A es para redes grandes, se pueden tener 128 (27) redes de 16,777,216 (224) equipos conectados en cada una.
- La red clase B es para redes medianas, se pueden tener 16,348 (214) redes de 65,535 (216) equipos conectados cada una.
- La red clase C es para redes chicas, se pueden tener 2,097,154 (221) redes de 256 (28) equipos conectados.

- Las redes clase D y E son de multicast y reservada respectivamente, también para futuros usos.

La numeración de las direcciones puede variar desde 0.0.0.0 hasta 255.255.255.255, entonces el aprenderse cada una de las direcciones para acceder a ellas sería muy difícil, pera ayudarnos a recordar las direcciones más fácilmente, existen los DNS (Domain Name Server), ellos hacen la traducción de la dirección en números a una dirección que se pueda recordar más fácilmente, por ejemplo una dirección 164.149.10.1 sería más fácil recordarla como www.nombre.com.pe.

La estructura también tiene una especificación definida, la última parte define por lo regular donde se encuentra la página, ".pe" es Peru, ".uk" es Inglaterra, ".es" es España, etc. El único pais que no tiene definida una abreviación es Estados Unidos, ya que ellos generaron la terminología, para éste caso la última parte y en los demás en la penúltima parte, se define el tipo de red, ".gov" para empresas de gobierno, ".net" para empresas de telecomunicaciones, ".com" para empresas del ámbito general, ".mil" para militares y ".edu" para universidades o empresas educativas, se han agregado algunas como ".tv" para la televisión pero no están bien especificadas.

La segunda parte define el nombre de la empresa o la página a donde se quiere acceder.

La primera parte define por lo regular que se está accesando a una página de Internet (WWW), recientemente se ha desechado ésta parte ya que siempre es lo mismo y algunas empresas mejor la ocupan para diferenciar servidores.

2.2 Antecedentes de IPv6

Debido al crecimiento explosivo en el uso de Internet nos ha llevado a aprender nuevas tecnologías, así como palabras en nuestro vocabulario como es el IP, es el protocolo encargado de transportar paquetes de información de una máquina a otra sobre la capa de Red su versión actual es la cuatro (IPv4).

IPv4 está lleno de muchas carencias en su diseño. Como ejemplo, el American Registry for Internet Number (ARIN)establece que el total de las direcciones de la clase A, el 62% de la clase B y el 37% de las direcciones de la clase C ya han sido asignadas desde 1996. La firma Network Wizards realiza una revisión periódica de ambos, hosts y dominios conectados a Internet y detectó que el crecimiento del número de estos, se ha dado a una taza exponencial, por lo que el IETF NGTrans [6] ha previsto que las direcciones IPv4 serán agotadas aproximadamente entre los años 2005 y 2011 [7], Es por eso que surgen

algunas interrogantes de como solucionar los problemas que trae consigo un protocolo mal diseñado como es el caso de IPv4 y como hacerlo sin tener que cambiar toda la infraestructura actual. Una posible solución que el IETF ha propuesto es el Protocolo de Internet versión 6 (IPv6), que además de solucionar el problema del agotamiento de direcciones IPv4, también cuenta con nuevas características como es la seguridad incorporada por medio de dos nuevos campos en el encabezado del paquete IP, reducción de las tablas de enrutamiento ya que se redujo el número de campos en el encabezado del paquete IP y muchos otros se hicieron opcionales. Calidad de servicio (QoS), por medio de etiquetado de paquetes, además se cuenta con otras características adicionales que trataremos más detalladamente en la sección 1.3. En este capítulo hablaremos primeramente de la historia de IPv6, después trataremos las mejoras que IPv6 tiene sobre IPv4. Describiremos también el encabezado de un paquete IPv6 y cada uno de sus campos. Por último, hablaremos de la estructura de direcciones IPv6.

2.3 Surgimiento de IPv6

A principios de la década de los 90's, cuando en Julio de 1991, el Internet Engineering Task Force (IETF) comenzó a trabajar para desarrollar un nuevo protocolo que resolviera en primer lugar el problema de saturación de direcciones de IPv4 y además adicionar a este nuevo protocolo algunas características que no se contemplaron en el diseño de IPv4. En noviembre de 1992 surgió una nueva área de investigación llamada Internet Protocol Next Generation (IPng) comisionada por el IETF para formalmente estudiar las diferentes propuestas para el desarrollo de este nuevo protocolo.

En diciembre de 1993 fue distribuido el RFC 1550, el cual invitaba a todas las partes interesadas a participar dando sus comentarios acerca de cualquier requerimiento específico que consideraran pertinente incluir durante el proceso de selección de IPng. Veintiún respuestas fueron recibidas, las cuales contenían puntos de vista de diferentes tipos de industrias, tales como la industria celular, televisión por cable y seguridad, solo por mencionar algunas. En el RFC 1726 , el grupo de investigación Ipng definió un conjunto de 17 criterios que serían usados para el proceso de evaluación del IPng y eran los siguientes:

- Escalabilidad: El nuevo protocolo debería ser capaz de identificar y direccionar por lo menos 1012 sistemas finales y 109 redes individuales.

- Flexibilidad topológica: La arquitectura de enrutamiento y protocolos para IPng debían permitir utilizar muchas topologías distintas de red.
- Rendimiento: Para IPng los host deberían ser capaces de transferir datos a tasas comparables a las alcanzadas con IPv4 utilizando niveles similares de recursos máquina.
- Servicio robusto: El servicio de red junto con los protocolos de control y enrutamiento para IPng deberían ser suficientemente robustos.
- Transición: Debían existir mecanismos para realizar la transición de IPv4 hacia IPng de manera transparente para los protocolos y aplicaciones de las capas superiores.
- Independencia del medio: Este nuevo protocolo debe de trabajar a través de una Internet con diferentes medios LAN, WAN y MAN, así como distintas velocidades de conexión, que van desde algunos bits/segundo hasta cientos de giga bits/segundo.
- Servicio de datagramas no confiables: En nuevo protocolo debía soportar un servicio no confiable de entrega de datagramas.
- Configuración, Operación y Administración: Este nuevo protocolo también debía permitir conexiones fáciles, además de operación y configuración ampliamente distribuida. También debía permitir la configuración automática de host y enrutadores.
- Operación segura: IPng también debía proveer una capa de red segura (IPSec).
- Acceso y documentación: Los protocolos que definen a IPng, sus protocolos asociados y protocolos de enrutamiento deberían ser publicados en los RFC's, así como estar disponibles libremente y no requerir licencia para su implementación.
- Nombrado único: IPng debía asignar a todos los objetos de la capa IP de manera global nombres de Internet únicos.
- Multicast: IPng debía soportar transmisión de paquetes Unicast y Multicast.
- Extensibilidad: IPng debía ser capaz de evolucionar para cubrir las necesidades futuras del Internet. Así mismo, conforme este evolucione, debería permitir diferentes versiones de él, que puedan coexistir sobre la misma red.
- Servicio de red: IPng debía permitirle a la red asociar paquetes con clases de servicio en particular y proveerlas con los servicios especificados por esas clases.
- Movilidad: El protocolo debía soportar huéspedes, redes e Inter redes móviles.

- Protocolo de control: El protocolo debía incluir soporte elemental para probar y depurar redes.
- Redes privadas: Por último, IPng debía permitir a los usuarios construir redes privadas sobre la infraestructura básica de red, soportando ambas, redes basadas ó no basadas en IP.

En base a este criterio varias propuestas fueron revisadas y en enero de 1995, el RFC 1752 fue publicado. En él se resumían las evaluaciones hechas a tres propuestas para el IPng:

- Arquitectura Común para el Protocolo de Internet de la Siguiente Generación (CATNIP).
- Protocolo de Internet Simple Plus (SIPP).
- TCP/IP con Direcciones más Grandes (TUBA).

CATNIP proponía una concordancia entre Internet, OSI y los protocolos Novell. Para lograrlo integraba protocolos de red tales como IP, Novell's Internetwork Packet Exchance (IPX) e ISO Connectionless Network Protocol (CLNP). El diseño de CATNIP permitía un gran número de protocolos de transporte, tales como el ISO Transport Protocol, class 4 (TP4), Connectionless Transport Protocol (CLTP), TCP,UDP, y Novell's Sequenced Packet Exchange (SPX), sin embargo los revisores de esta propuesta sintieron que CATNIP solo cumplía con cinco de los criterios establecidos, dos más no eran cumplidos y no tenían una conclusión acerca de los criterios restantes.

SIPP, por su parte proponía una evolución a IPv4, por esto, todas las funciones de IPv4 que les parecieron buenas fueron mantenidas en su nueva propuesta, también fue aumentado el tamaño de las direcciones de 32 a 64 bits de longitud y lo mejor de todo, su instalación sería como una actualización de software. SIPP además sería interoperable con IPv4. En cuanto a esta propuesta, los revisores decidieron que SIPP cumplía con diez de los criterios clave, dos criterios no eran cumplidos y no tenían una conclusión acerca de los criterios restantes.

TUBA proponía remplazar IPv4 con CLNP, lo cual traía consigo dos beneficios inmediatos: incremento en el espacio de direcciones y permitir a protocolos de la capa de transporte operar de manera transparente. Los revisores de TUBA determinaron que esta propuesta cumplía con cinco de los criterios clave, no cumplía un criterio y no tenían una conclusión acerca de los criterios restantes.

Como resultado de las revisiones a estas tres propuestas se decidió elegir a SIPP, incorporarle direcciones de 128 bits de longitud y hacer algunas otras modificaciones. El resultado final a todas estas modificaciones es lo que se conoce actualmente como IPv6 ó IPng.

2.4 Mejoras en IPv6

La primera y más importante mejora que IPv6 presenta frente a IPv4 es el incremento en el tamaño de las direcciones que este maneja, es decir 128 bits de IPv6 vs. 32 de IPv4, esto le permite soportar más niveles de direccionamiento jerárquico y tener muchos más nodos direccionables. La siguiente mejora que se ha tomado en cuenta en el diseño de IPv6 es la simplificación del formato del encabezado, ya que se eliminaron y en otros casos se hicieron opcionales algunos de los campos del encabezado, ganando con esto la reducción de las tablas de enrutamiento y mejorando el rendimiento de los enrutadores al ocupar menos tiempo analizando los campos de los encabezados, por lo tanto, con Ipv6 se mantienen bajos los costos del ancho de banda a pesar de que se cuadriplicaron las direcciones.

La tercera mejora es el soporte para extensiones y opciones en el encabezado, debido a la manera en que los campos del encabezado de IPv6 están organizados y al uso de encabezados opcionales, con esto es posible incrementar la flexibilidad del protocolo IP añadiendo nuevas características en un futuro, sin que sea necesario rediseñar por completo toda la estructura del paquete IP. La cuarta mejora en IPv6 son las capacidades en la Calidad de Servicio, ya que este permite etiquetar paquetes pertenecientes a un flujo particular para el cual el emisor desea un trato especial por parte de los enrutadores IPv6 que intervengan en la comunicación. La quinta mejora es la autenticación y privacidad gracias a dos encabezados opcionales que en conjunto nos brindan autenticación, integridad de datos y confidencialidad.

2.5 Encabezado del paquete IPv6

El encabezado de un paquete IPv6 consta de dos partes: Un encabezado IPv6 base y una extensión de encabezados opcionales, tal y como puede verse en la figura 2.3.

Un paquete IPv6 puede tener cero, uno o más encabezados opcionales. Cada encabezado opcional tiene una longitud múltiplo de 8 bits y deben ser colocadas en el orden mostrado en la tabla 2.4 por cuestiones de rendimiento:

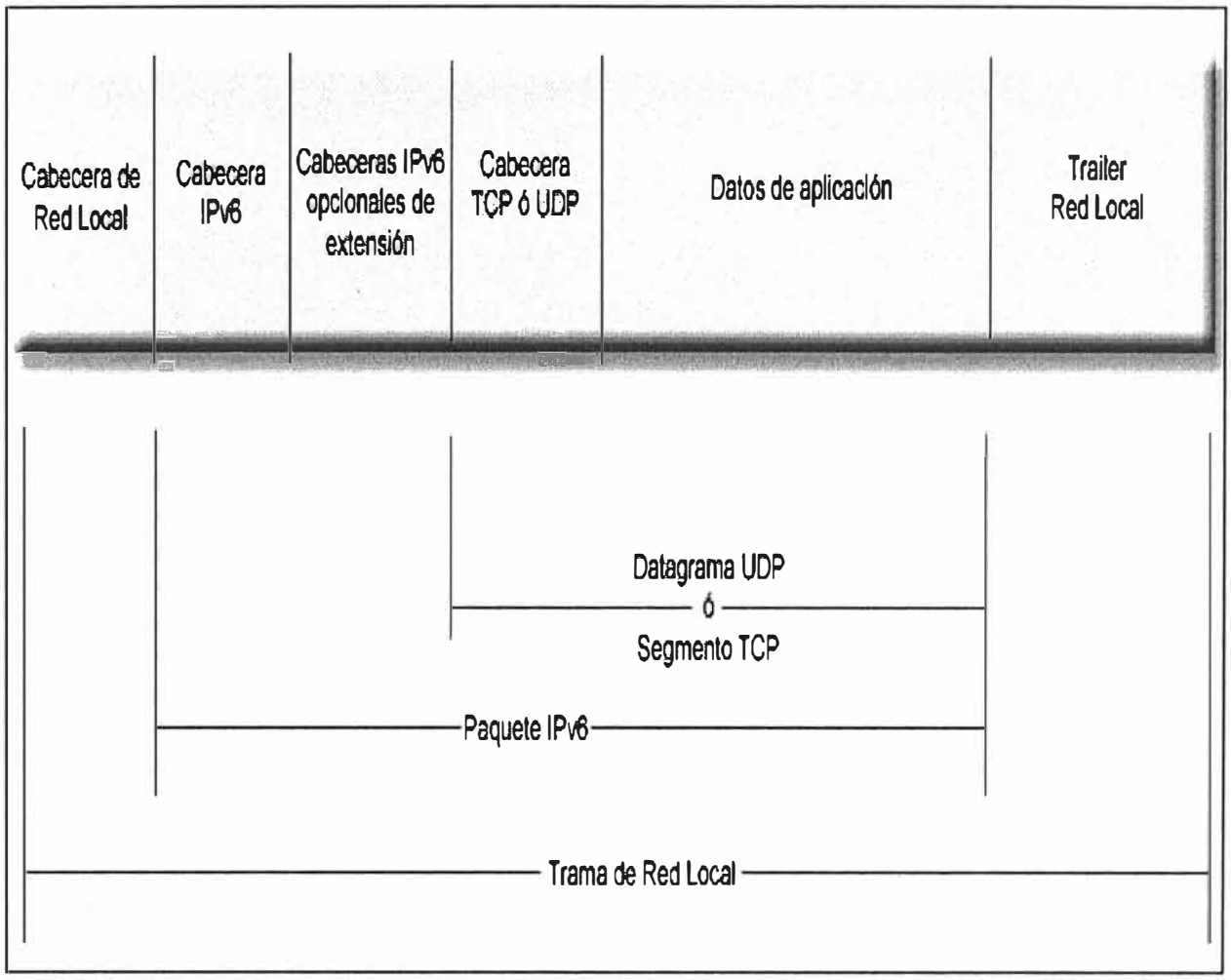


Figura 2.3 Formato de una trama IP (Ref 7)

Tabla 2.4 Orden en el que deben ser colocados los encabezados de extensión Ipv6

Orden	Cabecera	Características
1	Cabecera IPv6	Encabezado del paquete IPv6
2	Cabecera de opciones Hop by Hop	Contiene información opcional que debe ser analizada por cada nodo a lo largo de la ruta del paquete
3	Cabecera de opciones de Destino	Contiene opciones que serán procesadas por el primer destino que aparezca en el campo de dirección destino, más destinos subsecuentes que aparezcan listados en la cabecera de enrutamiento
4	Cabecera de Enrutamiento	Este encabezado opcional lista todos los nodos intermedios que deben ser visitados por el paquete en su trayecto hacia el nodo destino
5	Cabecera de Fragmentación	Es utilizada por un emisor IPv6 para enviar un paquete que es más grande que la Unidad Máxima de Transmisión (MTU) más pequeño de los nodos intermedios hacia el destino

6	Cabecera de Autenticación	Provee integridad de datos y autenticación del origen de los datagramas IP, con esto se logra tener protección contra reenvío de paquetes
7	Cabecera de Encapsulación de seguridad de la carga	Está diseñado para proveer confidencialidad, autenticación del origen de los datos, integridad sin conexión y servicio anti-reenvío
8	Cabecera de opciones de Destino (2)	Contiene opciones que serán procesadas solamente por el destino final
9	Cabecera de protocolos de capas superiores	Encabezados de protocolos de transporte tales como TCP ó UDP deben ir aquí

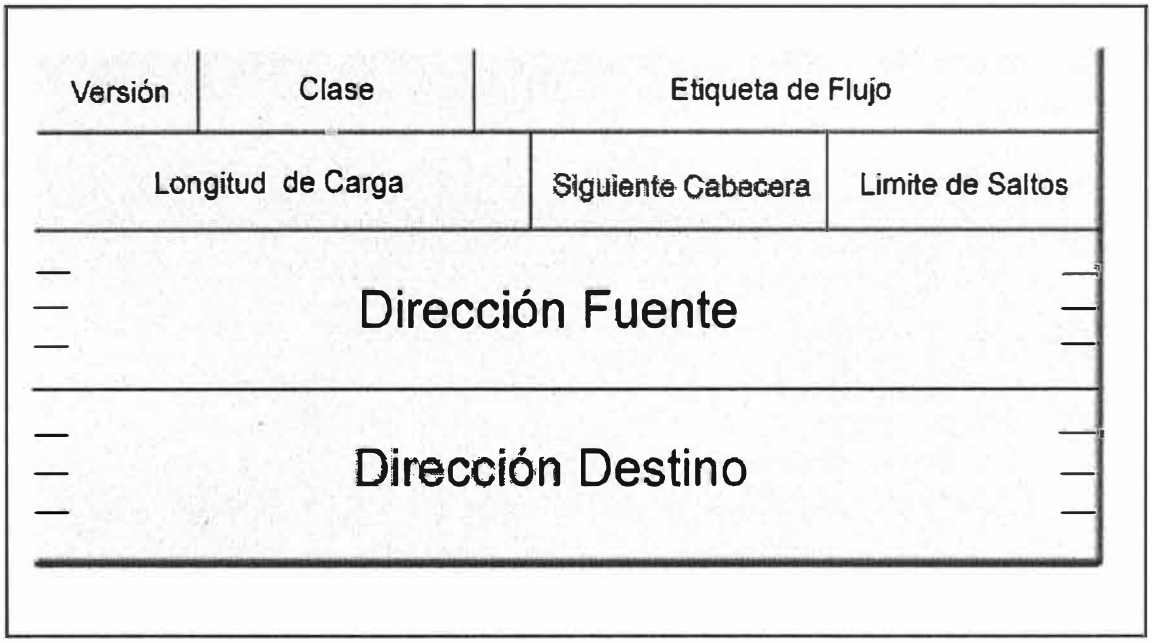


Figura 2.4. Encabezado IPv6 (Ref 8)

El encabezado IPv6 tiene una longitud de 40 octetos y tal como se ve en la figura 2.4 consta de los siguientes campos:

- Versión: Con 4 bits de longitud, este campo sirve para identificar la versión del protocolo IP, es decir IPv4 ó IPv6.
- Clase: Con 8 bits de longitud, este campo está diseñado para que enrutadores de envío y nodos originadotes de paquetes identifiquen y distingan entre diferentes clases ó prioridades de paquetes Ipv6.
- Etiquetado de Flujos: Con 20 bits de longitud, este campo puede ser utilizado por un huésped para solicitar un trato especial a ciertos paquetes, tales como aquellos

que requieran una calidad de servicio no por defecto ó una calidad de servicio de tiempo- real.

- Longitud de Carga: Con 16 bits de longitud, este campo se encarga de medir, dado en octetos, la longitud de la carga del paquete, la cual consta de todo lo que sigue después del encabezado IPv6, incluyendo los encabezados opcionales y protocolos de nivel superior, tales como TCP, FTP, etc. Este campo es similar al llamado Longitud Total en IPv4, pero a diferencia de este, Longitud de carga solo mide los datos después del encabezado, mientras que Longitud Total mide los datos y el encabezado.
- Siguiente Encabezado (next header): Con 8 bits de longitud, sirve para identificar al encabezado que sigue inmediatamente después del encabezado IPv6. La tabla 2.3 muestra los valores posibles que puede tomar este campo. Un paquete IPv6 además puede incluir cero, uno o más encabezados opcionales, por lo que dependiendo del número de encabezados que se contengan será el número de siguientes encabezados más uno extra que también deberá incluir. La figura 2.2 muestra un ejemplo en donde en primer instancia se puede ver un paquete IPv6 sin encabezados opcionales, ya que el siguiente encabezado después del encabezado IPv6 es el de los protocolos de l capa superior (en este caso TCP). En el siguiente caso, se muestra un paquete IPv6 al cual se agregó un encabezado de enrutamiento como encabezado opcional y en el tercer ejemplo podemos apreciar un paquete IPv6 al cual se le agregaron dos encabezados opcionales: Encabezado de Enrutamiento y Encabezado de Fragmentación.
- Limite de Saltos: Con 8 bits de longitud, este campo es análogo al campo time to live (TTL) en IPv4, pero a diferencia de este expresa el número de saltos y no de segundos que un paquete puede permanecer en la red antes de ser destruido. Cada nodo que reenvíe el paquete decrementa este número en uno.
- Dirección fuente: Con 128 bits de longitud, este campo contiene la dirección IPv6 del nodo que originó el paquete.
- Dirección destino: Con 128 bits de longitud, este campo contiene la dirección IPv6 del nodo que se espera sea el destino final del paquete. Las palabras "se espera" son utilizadas ya que la dirección destino puede no ser el último destino del paquete si está presente un encabezado de enrutamiento.

Tabla 2.5. Posibles valores del campo Siguiente Encabezado

	Cabecera
0	Opciones Hop-by-Hop
1	ICMPV4
4	IP en IP (encapsulación)
6	TCP
17	UDP
43	Enrutamiento
44	Fragmento
50	Encapsulado de seguridad de la carga (ESP)
51	Autenticación (AH)
58	ICMPV6
59	Ninguno (no hay siguiente cabecera)
60	Opciones de Destino

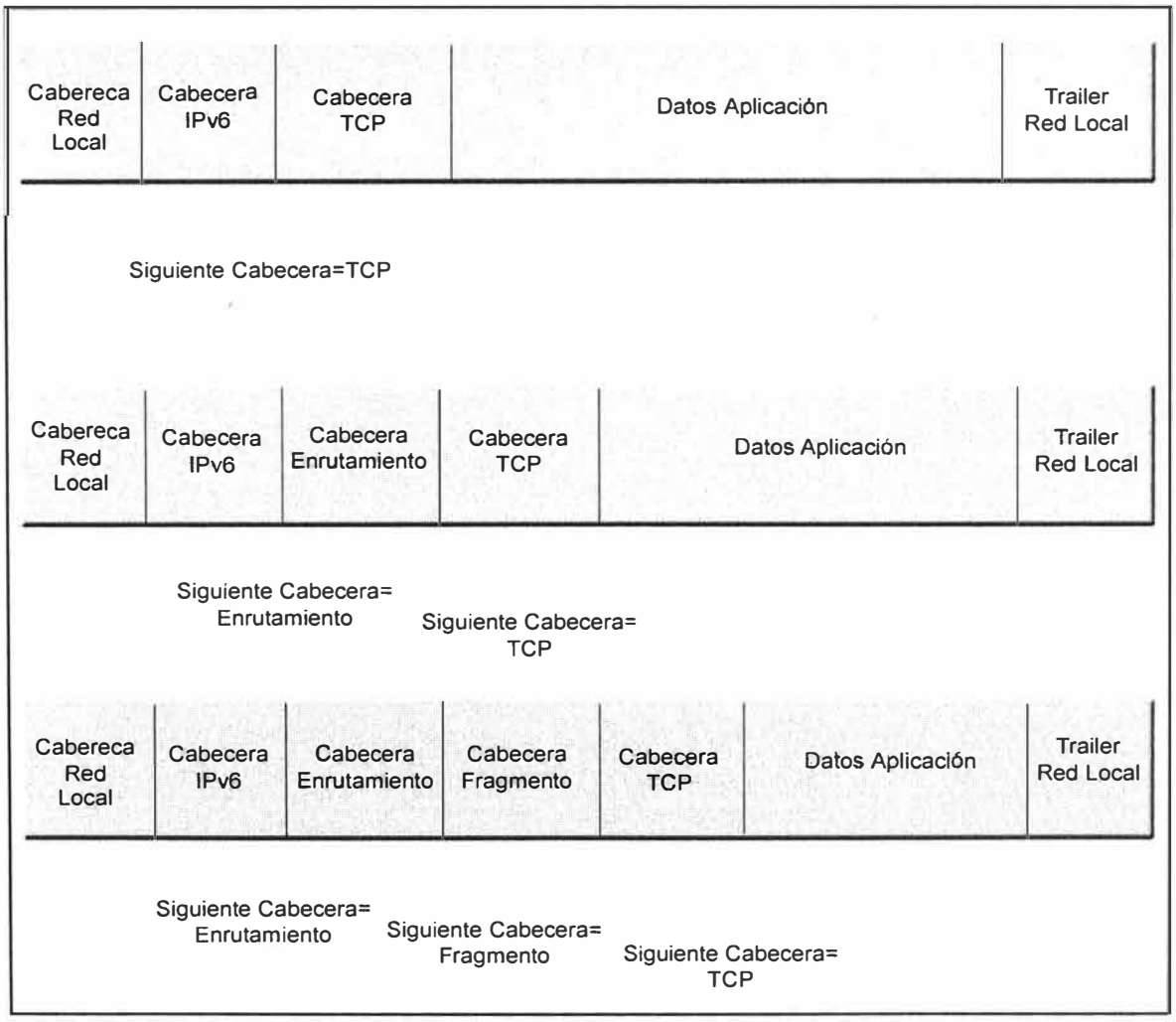


Figura 2.5. Encabezados de Extensión de IPv6 (Ref 9)

2.6 Estructura de direcciones IPv6

De acuerdo como está definido en el RFC 1884 [11] existen tres tipos de direcciones IPv6 y son los siguientes:

- Unicast: Definidas en el RFC 1887 [12] identifican una sola interfaz. Cuando un paquete es enviado a una dirección unicast, este solamente es entregado a la interfaz que tenga dicha dirección. Existen tres tipos de direcciones Unicast y estos son:

  Global: Las direcciones Unicast Globales son direcciones de Internet, es decir, tienen significado y pueden ser enrutadas por Internet, ya sea de manera nativa si así lo permite la infraestructura de red, ó por medio de túneles.

  Sitio: Este tipo de direcciones identifica una interfaz dentro de un dominio IPv6, pero no pueden ser enrutadas fuera de él, ya que pierden significado.

  Local: Este tipo de direcciones sirven para identificar una interfaz únicamente dentro de un mismo segmento de red (LAN), fuera de él pierden totalmente su valor. La figura 2.6 presenta la arquitectura de las direcciones unicast, así como un ejemplo de ellas.

- Multicast: Identifica a un conjunto de interfaces. Este tipo de direcciones son muy parecidas a las direcciones de difusión que maneja IPv4, es decir, un paquete que es enviado a una dirección Multicast es entregado a todas las interfaces identificadas por dicha dirección.

- Anycast: Identifica a un conjunto de interfaces. A diferencia de las direcciones multicast, un paquete que es enviado a una dirección anycast es entregado a una de las interfaces identificadas por dicha dirección (la más cercana de acuerdo al protocolo de enrutamiento). El RFC 1884 da una referencia sobre posibles usos para este tipo de direcciones, entre ellos están

  Identificación de un conjunto de enrutadores pertenecientes a un Proveedor de Servicio de Internet (ISP).

  Identificación de un conjunto de enrutadores agregados a una subred particular

  Identificación de un grupo de enrutadores que sirven como entrada a un dominio en particular.

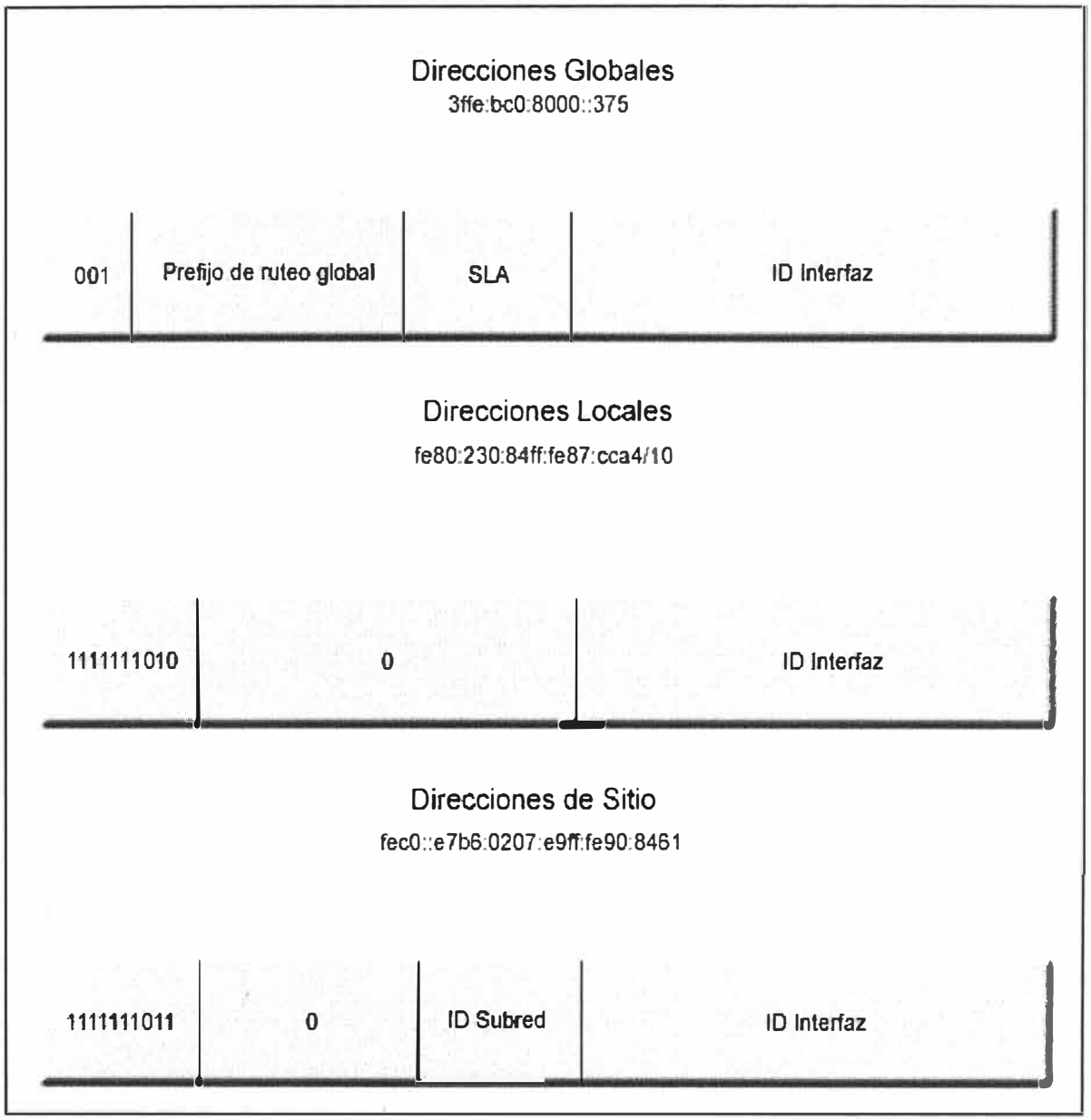


Figura 2.6. Arquitectura de las Direcciones Unicast (Ref 10)

Existen también algunos prefijos ya establecidos para las direcciones Unicast como los que se muestran a continuación:

- 2001::/16 utilizado en redes de producción y en Internet6.
- 2002::/16 utilizado para configurar túneles 6to4.
- 2a01::/16 red comercial europea
- 3FFE::/16 utilizado en el backbone de pruebas 6Bone

En esta investigación se utilizaron los prefijos 2a01 para realizar túneles broker entre enrutadores y nuestro gatweay para poder evaluar el comportamiento de IPSec sobre IPv6.

2.7 Seguridad en IPv6

Los diseñadores de IPv4 en un principio no tuvieron que preocuparse por implementar mecanismos de seguridad en IP ya que el Internet era utilizado por muy pocas personas; sin embargo, en la actualidad es uno de los medios más recurridos para realizar no solo transacciones monetarias, sino también manejo de información delicada. Debido a esto, se han desarrollado protocolos para tratar de asegurar los datos que son transmitidos por Internet, algunos ejemplos son SSL y S/MIME. Dichos protocolos no aseguran del todo la información transmitida ya que su efectividad se limita a las capas superiores a la capa de red.

Esto hace que la información transmitida sea vulnerable a los ataques que se realizan en la capa de red como los presentados en la tabla 3. IPv6 resuelve estos problemas de vulnerabilidad de la información incorporando los servicios de seguridad de IPSec (Internet Protocol Security) definido en el RFC 1825 mediante dos encabezados de extensión:

- Encabezado de Autenticación (AH): El Encabezado de Autenticación como lo define el RFC 1826 provee integridad de datos y autenticación del origen de los datagramas IP, con esto se logra tener protección contra reenvío de paquetes.
- Encapsulado de seguridad de la carga (ESP): ESP, definido por el RFC 1827 [16] está diseñado para proveer confidencialidad, autenticación del origen de los datos, integridad sin conexión y servicio contra reenvío de paquetes.

Tabla 2.6. Ataques más frecuentes en la capa de Red

Protocolo	Ataque
ICMP	Inundación de paquetes ICMP
ICMP	Ping de la muerte
IP	Denegación de servicio por envío de fragmentos de paquetes IP
IP	IP spoofing
IP	IP Timestamp
IP	IP Address Sweep Scan
IP	IP Source Route

2.7.1 Encabezado de Autenticación

El Encabezado de Autenticación provee, como ya se mencionó antes, integridad y autenticación del origen de los datos para datagramas IP, además de proveer protección contra ataques de re-envío de paquetes. La presencia del Encabezado de Autenticación es identificada por un valor de 52 en el campo Siguiente Encabezado y su encabezado tiene el formato mostrado el la figura 2.7

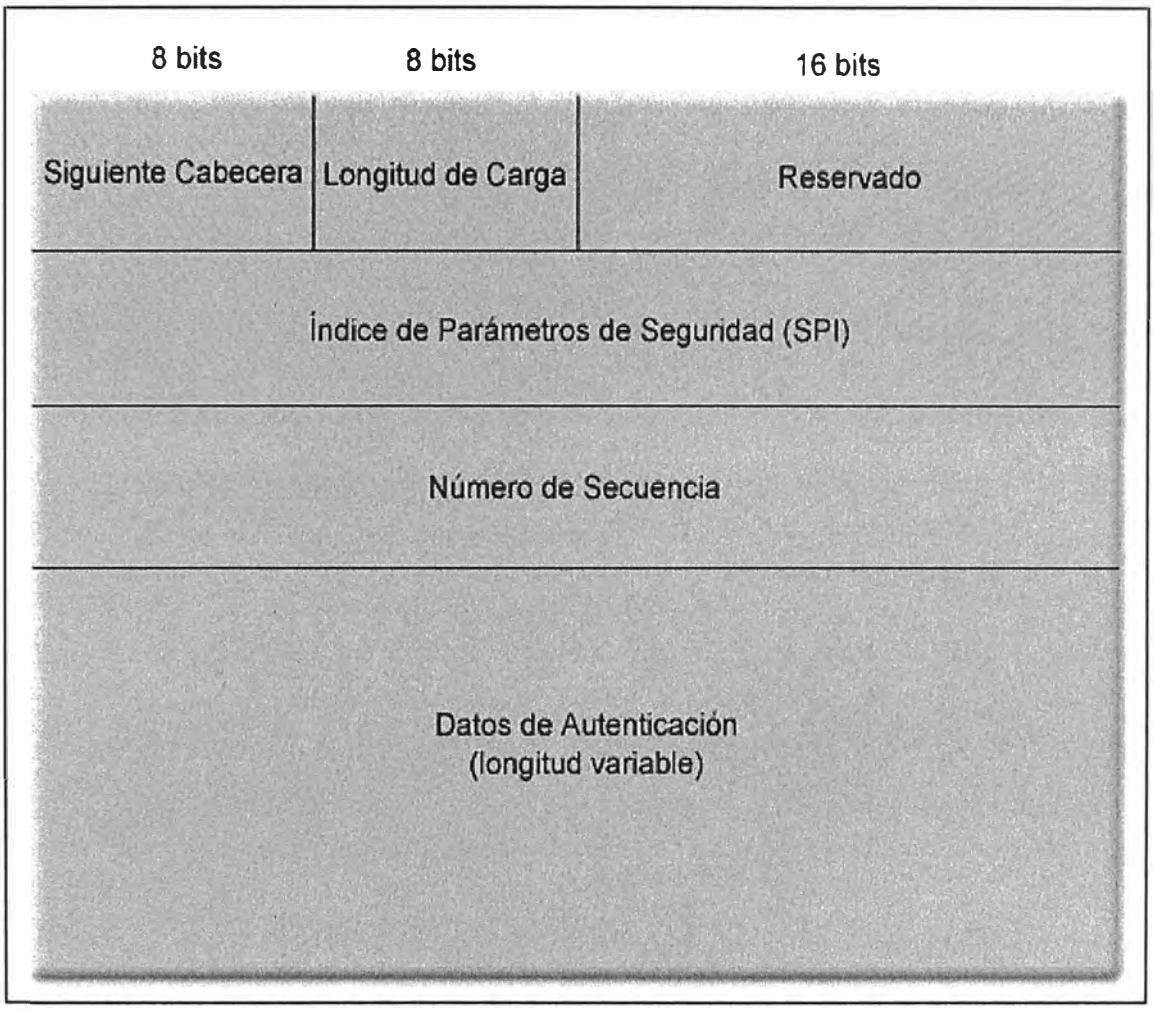


Figura 2.7. Formato del Encabezado de Autenticación

Como vemos en la figura 2.7, el encabezado de Autenticación consta de seis campos, los cuales describimos a continuación:

- Siguiente Encabezado: Tiene 8 bits de longitud e identifica el encabezado que sigue inmediatamente después del Encabezado de Autenticación.

- Longitud de Carga: Tiene 8 bits de longitud y provee la longitud del campo de autenticación en palabras de 32 bits, menos dos (los primeros 64 bits del Encabezado de Autenticación no son contados). El valor mínimo que puede tomar este campo es 1 el cual equivale a 3 palabras de 32 bits y es solamente utilizado para propósitos de depuración.
- Reservado: Tiene una longitud de 16 bits y está reservado para uso futuro. Es inicializado con un valor de cero.
- Índice de Parámetro de Seguridad: Tiene 32 bits de longitud e identifica la Asociación de Seguridad aplicada para este datagrama.
- Número de Secuencia: Contiene un número de 32 bits de longitud, el cual es incrementado monotónicamente. Los contadores tanto del emisor, como del receptor son inicializados a cero cuando una Asociación de Seguridad es establecida.
- Datos de Autenticación: Es un campo de longitud variable que contiene el Valor de Chequeo de Integridad (ICV o checksum) para este paquete. Este campo debe ser un múltiplo de 32 bits en longitud.

Para el cálculo del checksum criptográficamente seguro (también conocido como mensaje digerido o hash), así como para la selección de los campos del encabezado IP y encabezados de extensión se toman en cuenta las siguientes reglas:

- En el encabezado IP son excluidos los campos: Versión, Clase y Etiqueta de Flujo. El valor del campo Cuenta de Saltos se asumirá como cero.
- Todos los encabezados de extensión con el bit Cambio-en-Ruta encendido serán computados como secuencias de ceros.
- Si un Encabezado de Extensión de Enrutamiento está presente (lo cual indica que se deben intercambiar la dirección destino IPv6 y la siguiente dirección listada en el Encabezado de Enrutamiento, así como incrementar el valor del campo Siguiente Dirección) el valor del campo Destino IPv6 es llenado con la dirección del destino final IPv6.

El resultado de tomar todas estas consideraciones es un checksum relativamente corto (128 bits para MD-5 y 160 bits para SHA-1) el cual puede entonces ser truncado ligeramente

para permitir que el Encabezado de Autenticación tenga un tamaño múltiplo de 64 bits y con esto se consiga una optimización en el uso de memoria en los enrutadores.

Algoritmos de checksum En la especificación de IPv6 se cuenta con dos algoritmos para realizar el checksum y estos son: Keyed Message Digest No. 5 (MD-5) y Secure Hash Algorithm No. 1 (SHA-1).

MD-5 tal como lo define el RFC 1321 [17-18], toma como entrada un mensaje de longitud arbitraria y produce como salida una representación condensada de 128 bits de la entrada.

SHA-1 toma como entrada un mensaje de longitud menor a 264 bits y produce como salida una representación condensada de 160 bits de la entrada tal como lo explica el FIPS 180 . Este algoritmo a pesar de ser un poco más lento que MD5, también es más seguro contra ataques de fuerza bruta debido a que produce un mensaje de salida más grande que MD5.

2.7.1 Encapsulación de Seguridad de la Carga

La Encapsulación de Seguridad de la Carga está diseñada para proveer confidencialidad, autenticación del origen de los datos, integridad, un servicio anti re-envío de paquetes y una limitada confidencialidad en tráfico de flujos. La presencia del Encabezado de ESP es identificada por un valor de 50 en el campo Siguiente Encabezado y su encabezado tiene el formato mostrado el la figura 2.8.

Este encabezado consta de 7 campos, mismos que describimos a continuación:

- Indice de parámetros de Seguridad: Con 32 bits de longitud identifica la Asociación de Seguridad aplicada parta este datagrama.
- Número de Secuencia: Contiene un número de 32 bits de longitud, el cual es incrementado monotónicamente. Los contadores tanto del emisor, como del receptor son inicializados a cero cuando una Asociación de Seguridad es establecida.
- Carga de Datos: Tiene una longitud variable y contiene los datos descritos por el campo Siguiente Encabezado.
- Relleno: Puede opcionalmente tener de 0 a 255 octetos de datos de relleno.
- Longitud de relleno: Indica el número de octetos de relleno (0-255) que son agregados en el campo Relleno.
- Siguiente Encabezado: Con 8 bits de longitud este campo identifica el encabezado que sigue inmediatamente después del Encabezado de Encapsulación de Seguridad de la Carga.

- Datos de Autenticación: Tiene longitud variable y contiene el Valor de Chequeo de Integridad (ICV o checksum) para este paquete. La longitud de este campo depende de la función de autenticación que sea seleccionada. Este campo es opcional y es incluido solamente si la asociación de Seguridad ha elegido n servicio de autenticación.

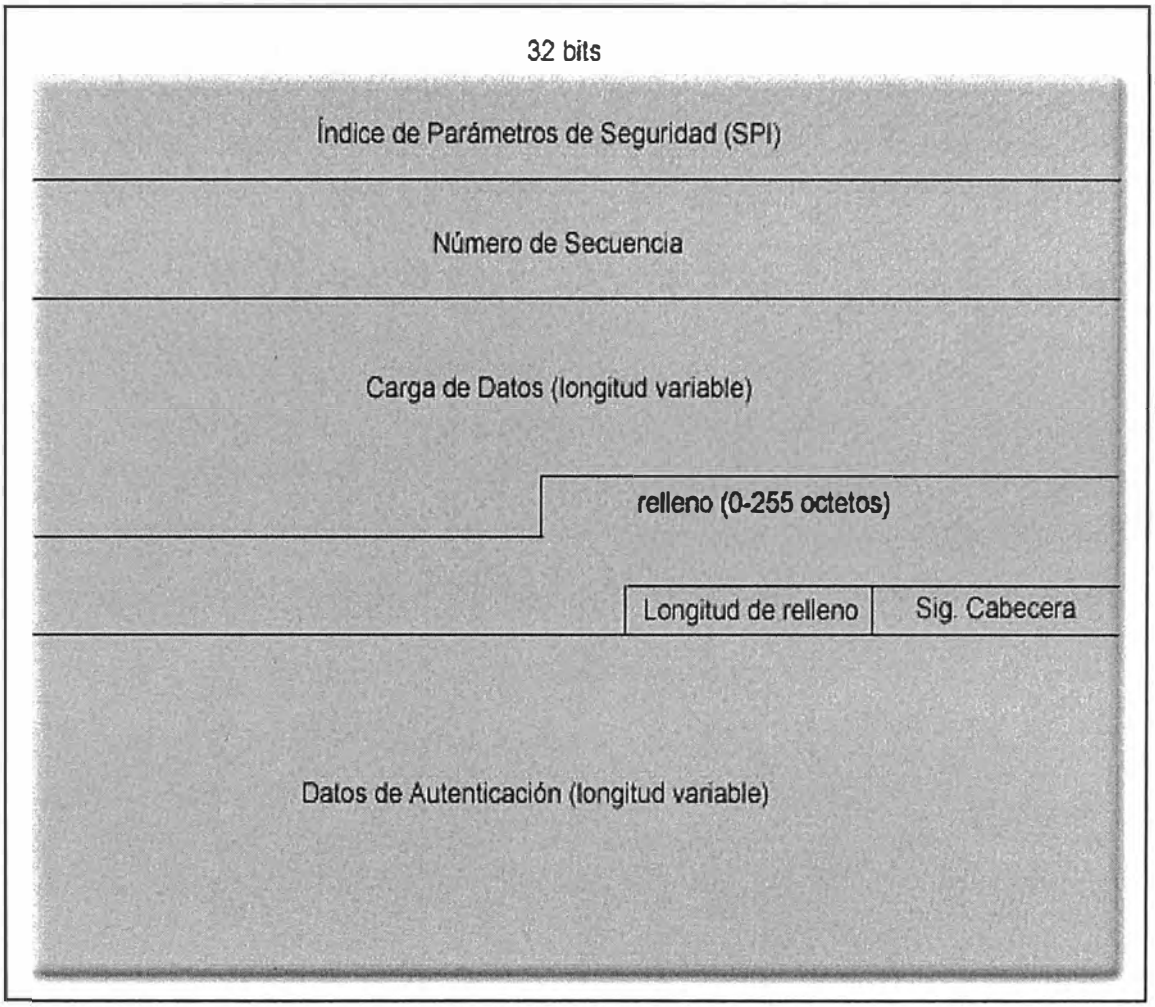


Figura 2.8. Formato del encabezado ESP

2.8 Mecanismos de transición

La conversión de redes IPv4 a IPv6 tardará un largo período de tiempo, por lo que en el diseño de IPv6 se han tomado en cuenta mecanismos que permitan la coexistencia y comunicación de ambos protocolos.

Estos mecanismos de transición se dividen en tres clases principales:

- Dual Stack: Este mecanismo de transición permite a un enrutador, host o servidor utilizar un stack IPv4 y un stack IPv6 simultáneamente, lo que trae consigo dos gran des ventajas: Por un lado un nodo con dual stack puede comunicarse con nodos que solo tienen un stack IPv4 de manera nativa y por el otro también puede comunicarse con nodos que solo tengan habilitado el stack IPv6 de manera nativa. Su principal desventaja es la necesidad de contar con una infraestructura de red que soporte el tráfico IPv6 de manera nativa.
- Túneles: Este mecanismo de transición permite a un enrutador IPv6, host IPv6 o servidor IPv6 comunicarse con otras redes IPv6 a través de la infraestructura IPv4 actual. Esta técnica consiste en encapsular los paquetes IPv6 dentro de paquetes IPv4 y entonces enviarlos sobre una red IPv4 a un nodo IPv4 destino el cual se encargará de extraer los paquetes IPv6 y entregarlos a su destino final. La principal ventaja de éste mecanismo de transición es que solo es necesario tener un Dual Stack en los nodos que servirán como extremos del túnel. Su principal desventaja es el retardo adicional ocasionado por el encapsulado y desencapsulado de paquetes IPv6 en datagramas IPv4, así como el tráfico de un mayor número de paquetes ocasionado por la reducción de espacio para datos en los datagramas IPv4 que contienen dentro paquetes IPv6.
- Traducción de protocolos: Este mecanismo de transición permite a un nodo que solo cuenta con el stack IPv6 habilitado dentro de una red IPv6 comunicarse con otro nodo que solo tiene el stack IPv4 habilitado dentro de una red IPv4. Sin embargo, ésta técnica requiere tener habilitados mecanismos de traducción entre IPv4 e IPv6 en las orillas de ambas redes (enrutadores). La principal desventaja es que todo el peso de este mecanismo de transición recae en los dispositivos encargados de hacer dicha traducción, a los que no siempre se tiene acceso.

A continuación trataremos con más detalle los mecanismos de transición Dual Stack y Túneles ya que éstos son los que se utilizaron para la realización de esta investigación por ser los más ampliamente difundidos.

2.8.1 Dual Stack

Este mecanismo de transición como ya se había mencionado anteriormente permite a un nodo utilizar un stack IPv4 y un stack IPv6 simultáneamente teniendo dos grandes ventajas: por un lado un nodo con Dual Stack puede comunicarse con nodos que solo

tienen Stack IPv4 de manera nativa y por el otro también puede comunicarse con nodos que solo tengan habilitado el Stack IPv6 de manera nativa Esta técnica no es nueva, ya que en el pasado fue muy utilizada para realizar transiciones entre otros protocolos, tales como el desarrollo de IPv4 dentro de redes como: Internet Packet Exchange (IPX) y Digital Equipment Corporation Network (DECnet) entre otros.

Antes de que poder utilizar la capacidad del Dual Stack sobre un nodo es necesario modificar las aplicaciones basadas en IPv4 para que éstas también soporten IPv6, ya que el API de las aplicaciones basadas en IPv4 está codificado para utilizar únicamente direcciones de 32 bits. Como se muestra en la figura 2.9 (a) Las aplicaciones que soportan únicamente el Stack de IPv4 pueden utilizar TCP o UDP como capa de transporte para entregar los datos, después estos datos llegan al Stack IPv4, en donde son puestos dentro de paquetes IPv4. Estos paquetes IPv4 más tarde son llevados a la interfaz de red. El valor del identificador del protocolo de red usado por tramas Ethernet para paquetes IPv4 es 0x0800. Cuando las aplicaciones son modificadas para soportar IPv6 tal como se ve en la figura 2.9 (b), éstas pueden llamar la función del API correcta que pueda manejar direcciones de 128 bits.

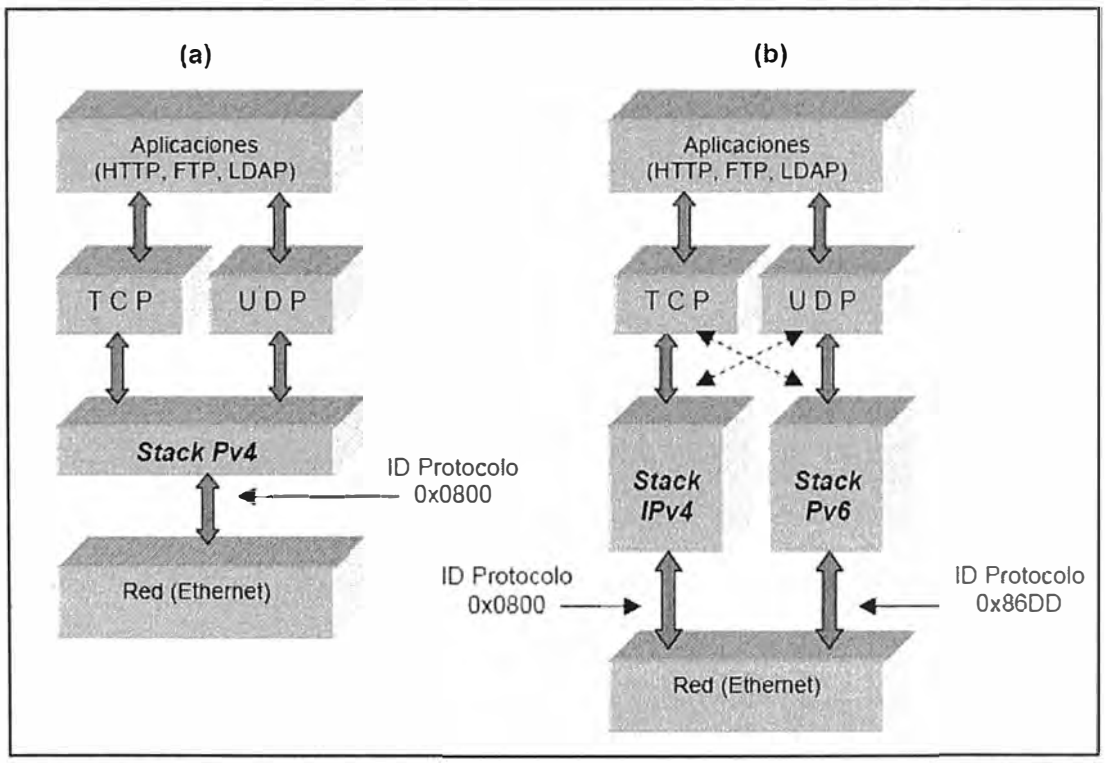


Figura 2.9. Arquitectura de Dual Stack:
(a) Aplicaciones que solo utilizan el Stack IPv4 para enviar paquetes
(b) Aplicaciones que soportan ambos Stack's para enviar paquetes

Así los datos que llegan al dual stack pueden seleccionar cual de ellos utilizar para generar los paquetes. Esta selección se puede realizar de dos maneras:

- Manual: Cuando el usuario conoce la dirección IPv6 del nodo destino. Para aplicaciones Web es necesario utilizar el formato para direcciones en un URL tal como está definido en el RFC 2732. El uso de direcciones manualmente establecidas solo es recomendable para propósitos de depuración, en lo posible debe utilizarse un servicio de nombrado.
- Utilizando un servicio de nombrado: Se puede configurar un Nombre de Dominio Completamente Calificado (FQDN) en un servidor de nombrado DNS con ambas direcciones IPv4 e IPv6 y eventualmente este puede ser consultado para proveer información acerca de la disponibilidad de un nodo sobre IPv4 o IPv6. Una aplicación que soporta ambos stack's IPv4 e IPv6 solicitará al servicio de nombrado le resuelva FQDN en ambos tipos de direcciones, pero generalmente dará preferencia a las direcciones Ipv6.

2.8.2 Túneles

La principal función de los túneles es llevar protocolos incompatibles o datos específicos sobre una red, por ejemplo, los túneles del Protocolo de Enrutamiento Multicast Vector Distancia (DVMRP) llevan datagramas multicast sobre redes unicast. IPSec en modo túnel lleva datos protegido s por un algoritmo de cifrado. Para el desarrollo de IPv6 sobre una infraestructura existente IPv4 los túneles proveen una manera básica de comunicación entre hosts o islas de hosts IPv6 utilizando IPv4 como medio de transporte. En la figura 8 un túnel es creado para comunicar dos islas de hosts IPv6 sobre el Internet. Los enrutadores encargados de administrar el túnel deben tener configurado un dual stack para poder encapsular los paquetes IPv6 en datagramas IPv4 y viceversa.

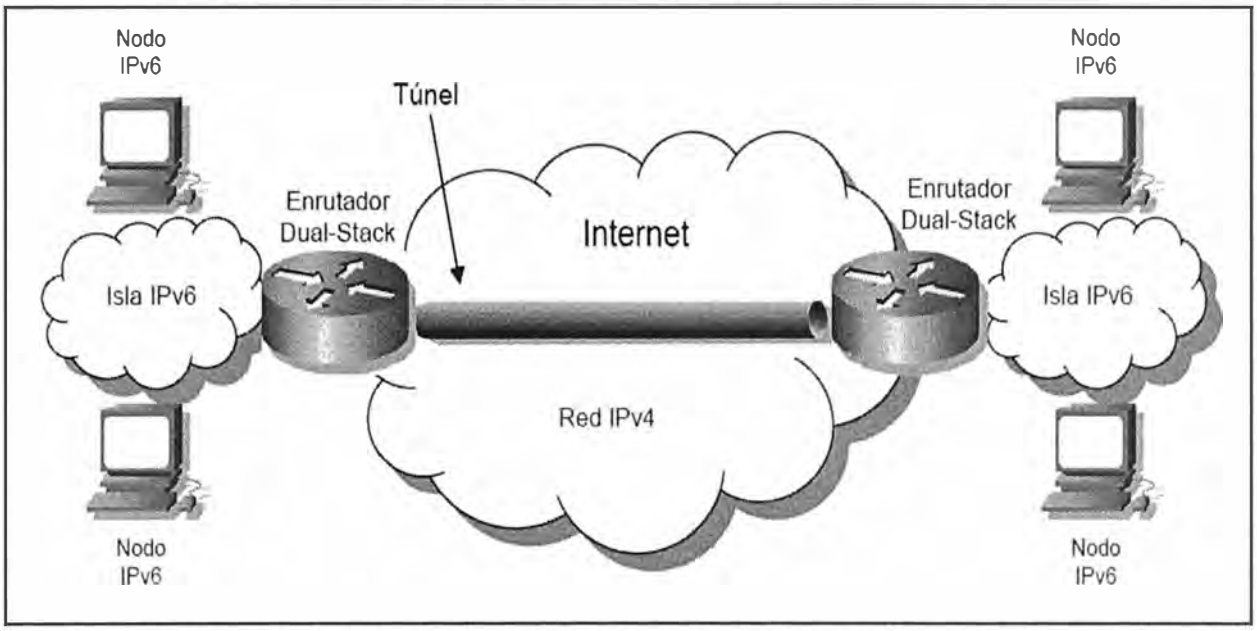


Figura 2.10. Túnel establecido entre dos islas IPv6 a través de la infraestructura IPv4

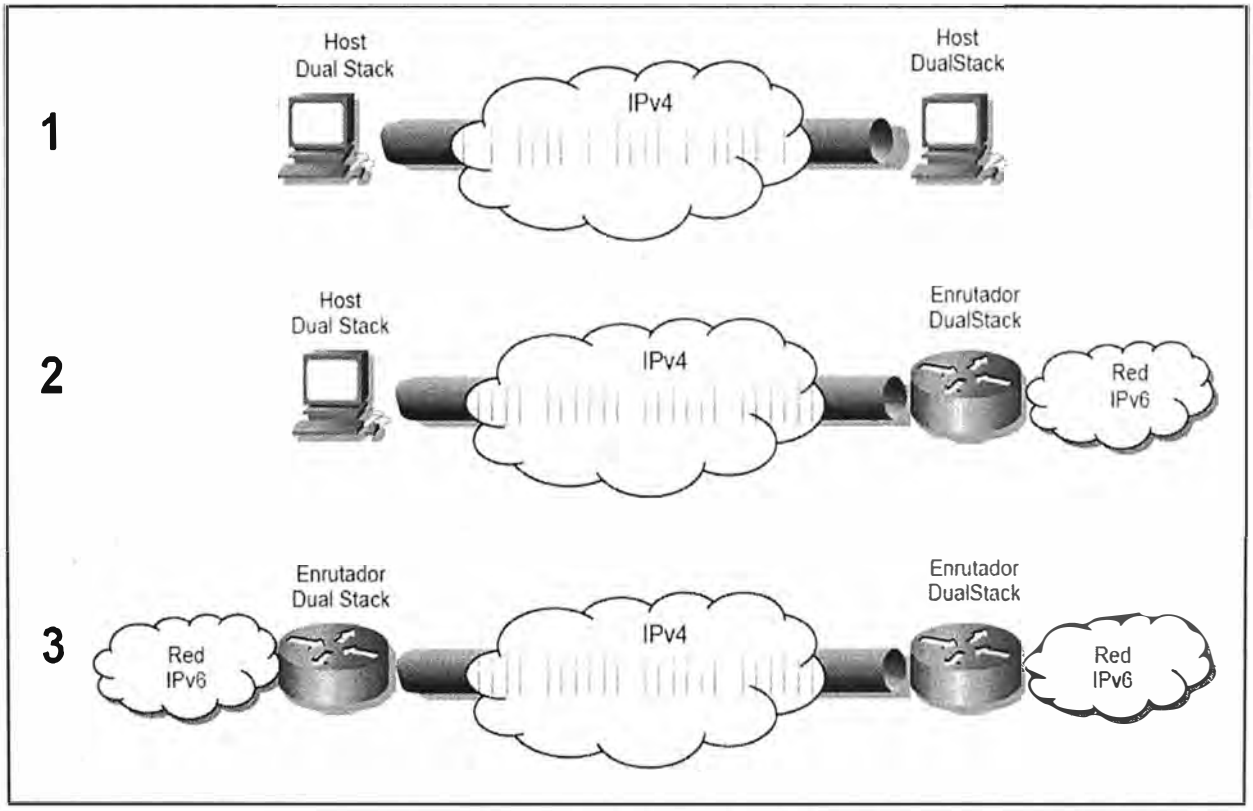
Para poder configurar un túnel primero es necesario tomar en cuenta los siguientes aspectos:

- Habilitar el protocolo 41: Si se tiene configurado un cortafuegos sobre IPv4, es necesarioestablecer una regla que permita el acceso y salida al protocolo 41. Como está descrito en el RFC 2893 "IPv6 Transition Mechanisms" el número de protocolo asignado a la encapsulación de paquetes IPv6 en IPv4 es el 41. Este valor es utilizado en el campo "Número de Protocolo" en el encabezado de IPv4 para especificar la encapsulación de un paquete IPv6 en un paquete IPv4.
- Manejo de mensajes de error (ICMPv4): Algunos viejos enrutadores en caso de error solo regresan ocho octetos de datos, sin embargo, los nodos emisores de los paquetes IPv6 necesitan conocer los campos de direcciones IPv6 en el error y cada uno de ellos ocupa 16 octetos.
- Traducción de Direcciones de Red (NAT): No es posible establecer túneles IPv6 en IPv4 a través de NAT cuando éste está habilitado en modo traducción dinámica de puerto y redirección de puerto. Por otra parte, es posible establecer dichos túneles si NAT es configurado en modo estático como lo muestra el RFC 2766 .

Una vez visto esto es necesario definir un escenario en el cual se usará el túnel. Existen tres posibles escenarios para la creación de un túnel:

- Host a Host: Esta arquitectura requiere que ambos hosts tengan un Dual Stack configurado y solo permite el establecimiento de sesiones IPv6 extremo a extremo entre ellos.
- Host a Enrutador: Hosts con un Dual Stack pueden establecer un túnel con un enrutador que también cuente con un Dual Stack. El enrutador puede tener conectividad IPv6 nativa sobre otra interfaz por lo que esta arquitectura permite el establecimiento de sesiones IPv6 extremo a extremo entre cualquier host de la isla IPv6 y el host aislado a través del enrutador.
- Enrutador a Enrutador: Enrutadores con un Dual Stack sobre una red IPv4 pueden establecer un túnel hacia otro enrutador con Dual Stack. Estos enrutadores pueden ser utilizados para interconectar islas de hosts IPv6, por lo que cualquier host puede establecer sesiones IPv6 extremo a extremo con otro host de la otra isla IPv6.

En la figura 2.11 se muestran los tres escenarios posibles para la creación de túneles, el caso (1) muestra la generación de un túnel host a host. El caso (2) presenta la generación de un túnel host a enrutador y por último, el caso (3) presenta la generación de un túnel enrutador a enrutador



F igura 2.11. Escenarios para la creación de un túnel

2.8.2.1 Técnicas para establecer Túneles

El IETF definió protocolos y técnicas para establecer túneles entre nodos con dual-stack, entre estas técnicas se encuentran las siguientes:

- Túneles 6to4: En esta técnica los extremos del túnel están determinados por las direcciones globales IPv4 embebidas dentro de direcciones IPv6 6to4. Las direcciones IPv6 6to4 están formadas por la combinación de un prefijo de enrutamiento global 2002::/16 y una dirección IPv4 globalmente única. Los túneles 6to4 pueden ser configurados entre dos enrutadores en la orilla de sus respectivas redes, o entre un enrutador y un host. El único inconveniente de esta técnica para establecer túneles es que solo permiten enviar tráfico IPv6 entre hosts con prefijos de enrutamiento 2002. Para poder comunicarse con nodos con otros prefijos de enrutamiento tales como 2001::/16 y 3FFE::/16 es necesario utilizar un enrutador de reenvío (relay router) del 6bone el cual se encargará de proporcionar un servicio de enrutamiento global 6to4.
- Intransite Automatic Tunnel Addressing Protocol (ISATAP): Esta técnica permite crear túneles IPv6-in-IPv4 automáticamente dentro de un sitio IPv4. Cada host solicita a un enrutador dentro del sitio IPv4 una dirección IPv6 e información de enrutamiento, de esta manera, los paquetes enviados al Internet IPv6 son enrutados a través del enrutador ISATAP y los paquetes destinados hacia otros hosts dentro del mismo sitio son entregados directamente mediante túneles ISATAP. Las direcciones IPv6 se configuran automáticamente mediante el protocolo "descubrimiento de enrutador" ISATAP, aunque también pueden ser configuradas de manera manual. Una dirección ISATAP al igual que una dirección 6to4 está formada por la concatenación de un prefijo global de agregación unicast IPv6 y el identificador de interfaz. El prefijo utilizado por ISATAP para habilitar una dirección ISATAP en un host es FE80::/10 (dirección local). El identificador de interfaz es formado agregando los 32 bits de la dirección IPv4, después se concatena el valor 0000:5EFE (reservado por IANA para identificar direcciones ISATAP). Ejemplo: para una dirección IPv4 148.247.54.122 su dirección IPv6 ISATAP sería FE80::5EFE:94F7:367A. Las direcciones ISATAP también pueden utilizar prefijos unicast Globales de 64 bits, los cuales son asignados por los enrutadores. Cuando un nodo ISATAP desea comunicarse con otro nodo ISATAP

sobre IPv6 el paquete IPv6 es encapsulado dentro de un datagrama IPv4 al igual que como sucede con el mecanismo 6to4.

- Tunnel Broker: El IETF definió este mecanismo para facilitar el desarrollo de túneles configurados sobre redes IPv4 ya que mediante esta técnica no se tiene que configurar manualmente cada extremo del túnel. Tal como está establecido en el RFC 3053 "IPv6 Tunnel Broker" [23] el tunnel broker es un sistema externo que actúa como un servidor sobre la red IPv4 y recibe peticiones de nodos con dual stack para configurar túneles automáticamente (modelo cliente-servidor). Estas peticiones son enviadas vía HTTP sobre IPv4 por el nodo que desea configurar dicho túnel. El tunnel broker entonces envía de vuelta al cliente información tal como la dirección IPv4 del servidor del túnel, la dirección IPv6 del servidor del túnel, la nueva dirección IPv6 que será asignada a este host con dual stack y las rutas IPv6 default para la configuración del túnel. Algunos tunnel broker's ya proporcionan scripts de configuración para los hosts clientes. Finamente el tunnel broker aplica comandos de manera remota sobre un enrutador con dual stack y que está conectado a un dominio IPv6 para habilitar el túnel configurado. Para poder hacer uso de esta técnica es necesario utilizar los servicios de alguna entidad que ofrezca el servicio de tunnel broker tales como:
    - Freenet6
    - Dolphins tunnel broker
    - British Telecom tunnel broker
    - Hurricane Electric
    - IPv6TaskForce
    - SixXS

La gran mayoría de los tunnel brokers ofrecen el servicio de manera gratuita, el único requisito es registrarse mediante el llenado de un pequeño formulario. En nuestro proyecto ingresamos datos en IPv6 Task Force que pasamos a detallar en el siguiente capitulo.

CAPITULO III
IMPLEMENTACION DE GATEWAY SOBRE IPV6

Ya luego de mucha teoría nos adentramos en el tema de nuestro trabajo que es la de implementar un túnel para habilitar nuestro gateway y sobre ella todos nuestros servicios. Como en lo mencionamos en el aspecto teórico el método elegido para la conexión al a nube IPv6 es la de Tunnel Broker , para lo cual debemos de adecuar nuestro nodo como dual stack afín de oír peticiones IPv4 e IPv6 .

3.1 Solicitud del pedido.

En el portal de IPv6 Task Force nos brinda un acceso y nos otorga una asignación de números con prefijo /48 , por lo que nos ofrece una buena alternativa de conectividad.
Para lo cual completamos los campos del siguiente link :
http://www.ipv6tf.org/index.php?page=using/connectivity/test&register=1

Adicionalmente existe otros proyectos que nos proporcionan conectividad para solicitarlo gratuitamente por un periodo de tiempo y nos permitan acceder a una red IPv6 comercial.
Otros Tunnel Brokers:

http://tunnelbroker.ipv6.net.au (Australia)
http://tunnel.be.wanadoo.com (Belgium)
http://www.hexago.com/ (Canada)
http://tb.6test.edu.cn/ (China)
http://tunnelbroker.ipv6.estpak.ee/ (Estonia)
http://tb.ngnet.it (Italy)
http://www.iij.ad.jp/en/IPv6/zikken-e.html (Japan)
http://tbroker.mybsd.org.my/ (Malaysia)
http://tbroker.manis.net.my/ (Malaysia)
http://www.sixxs.net/ (Netherlands)
http://www.uninett.no/ (Norway)
http://tb.ptin.euro6ix.org/ (Portugal)
http://tunnel-broker.singnet.com.sg/ (Singapore)
http://www.xs26.net (Slovakia)
http://tunnelbroker.as8758.net/ (Switzerland)
http://tb.ipv6.chttl.com.tw/ (Taiwan)
http://tb.ipv6.btexact.com (United Kingdom)
http://tunnelbroker.net (USA, California)

Nos pedirá para ello que ingresemos los campos siguientes :

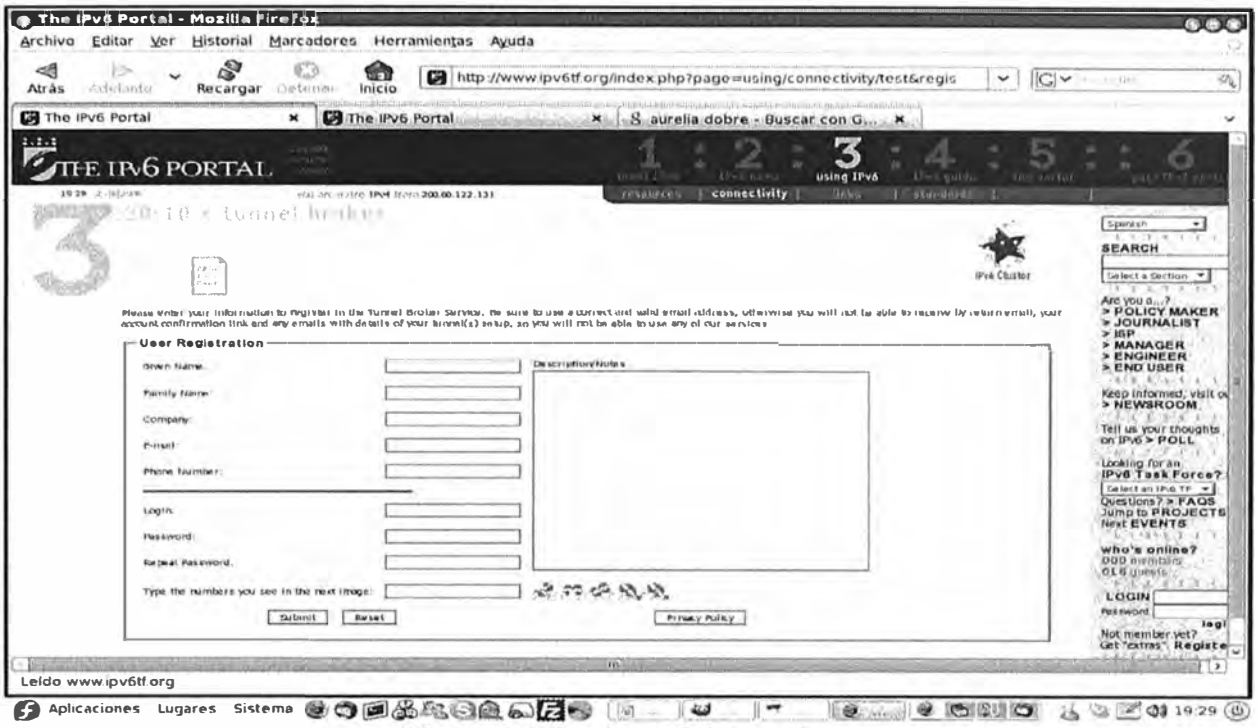


Figura 3.1 Registro del usuario

Confirmado nuestra suscripción el tema es ahora registrar un túnel, validamos nuestro usuario y contraseña creado y pasamos a completar los datos que nos solicitan :

Figura 3.2 Validación del usuario

Ahora solo completamos los campos que no solicitan , para nuestro caso se trata de un router PC soportado por Linux , el cual cumple fielmente con el trabajo asignado hasta ahora , otro dato necesario para establecer el Tunnel es la necesidad de un IP publico estático , el cual lo hago trabajar con el ADSL de mi proveedor.

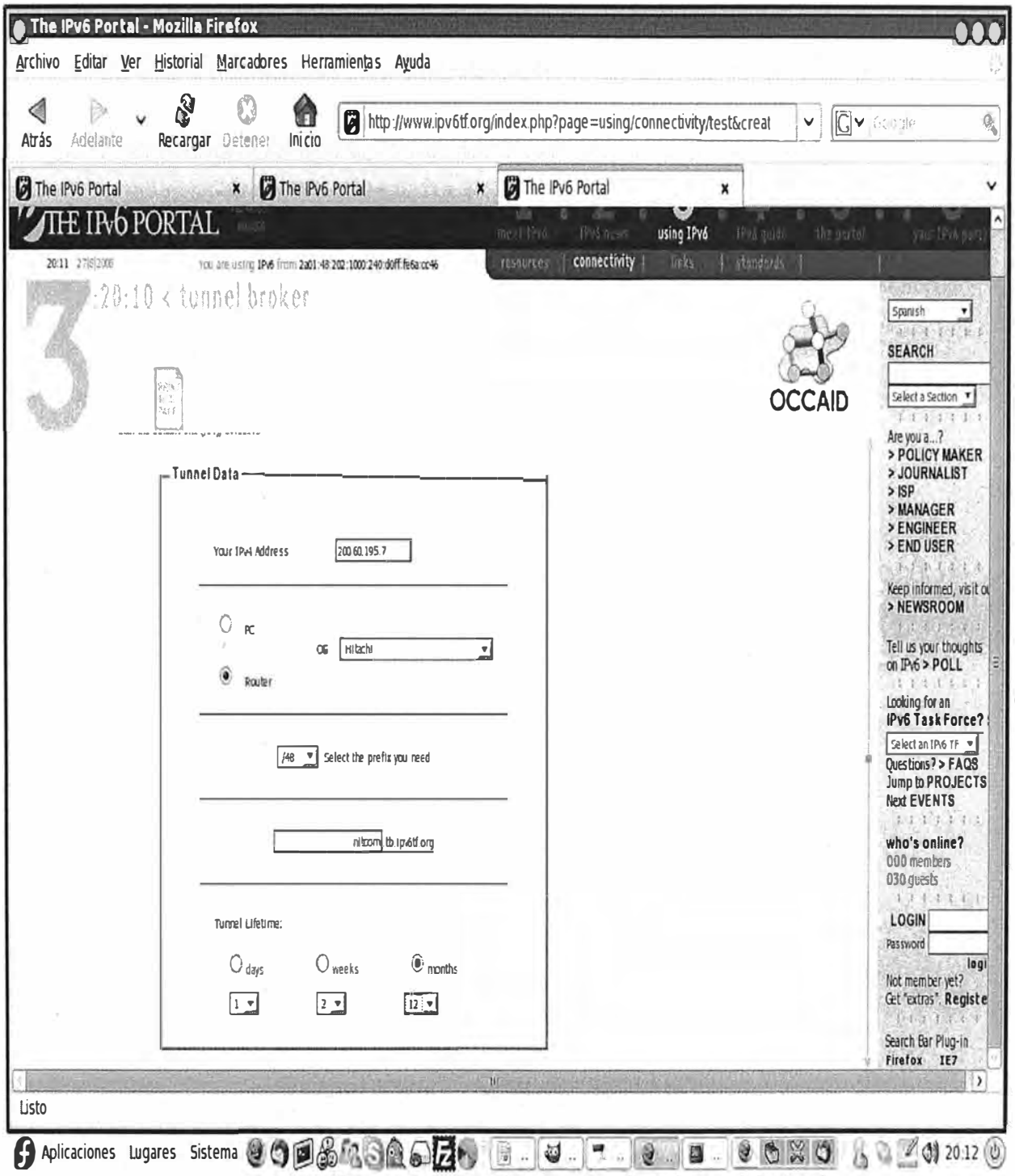


Figura 3.3 Ingreso de los datos para el registro del túnel

Completados esos datos obtendré un correo de respuesta en el cual me establece los parámetros a completar en un script que lo puedo asumir o no , en mi caso me sirve para establecer mi conexión de una manera muy particular .

El correo es el presente :

Dear Gino Alania,

This is a confirmation e-mail about the creation of the tunnel you requested.

Tunnel data is shown below:

Name: 1004

Your IPv4 Address: 200.60.195.7

Server IPv4 Address: 213.172.34.125

Your IPv6 Address: 2A01:0048:0100:0001:0001::032

Server IPv6 Address: 2A01:0048:0100:0001:0001::031

Network Prefix: 2A01:0048:0202::/48

nitcom.tb.ipv6tf.org has been registered in the DNS as your tunnel's side domain name.

Attached to this e-mail you will find a configuration script. You may need to edit it to change your IPv4 address if you are behind a NAT. Then save it with .sh (typically for *BSD, linux-like and unix-like systems) or .bat (for windows systems) extensions rather than .txt. Then you must be able to run it in your machine.

To remove this tunnel in the remote side, you need to login in the Tunnel Broker, then click on "View tunnel information". Select then the remove option that will appear next to the tunnel that you want to remove, otherwise it will be removed automatically after the expiration period that you have chosen.

To login, go to

http://www.ipv6tf.org/index.php?page=using/connectivity/test&autentica=1

Finalmente listamos el túnel creado en el mismo portal el color naranja nos muestra el grafico.

Name	IPv4 Address	IPv6 Client Address	IPv6 Server Address	Domain / Removal Date
1004	200.60.195.7	2A01:0048:0100:0001:0001::032	2A01:0048:0100:0001:0001::031	nitcom.tb.ipv6tf.org
1019	200.60.195.7	2A01:0048:0100:0001:0001::122	2A01:0048:0100:0001:0001::121	2008-07-25 22:01:21
tb-00102	200.60.195.7	2001:800:40:2AA0:0001::122	2001:07F9:0400:0001:0001::B1	2005-10-20 04:46:13
tb-00155	200.60.195.7	2001:07F9:0400:0001:0001::8B2	2001:07F9:0400:0001:0001::8B1	2007-07-31 21:18:18
tb-0017	200.60.195.7	2001:07F9:0400:0001:0001::1E2	2001:07F9:0400:0001:0001::1E1	2005-10-24 21:11:41
tb-0017	200.60.195.7	2001:07F9:0400:0001:0001::1E2	2001:07F9:0400:0001:0001::1E1	2005-11-20 21:14:37
tb-0017	200.60.195.7	2001:07F9:0400:0001:0001::1E2	2001:07F9:0400:0001:0001::1E1	2006-11-15 22:02:16
tb-00181	200.60.195.7	2001:07F9:0400:0001:0001::A22	2001:07F9:0400:0001:0001::A21	2007-06-01 22:03:44
tb-00258	200.60.195.7	2001:07F9:0400:0001:0001::ED2	2001:07F9:0400:0001:0001::ED1	2007-06-01 04:49:43
tb-0038	200.60.195.7	2001:800:40:2AA0:0001::122	2001:07F9:0400:0001:0001::B1	2005-10-22 23:30:05
tb-0094	200.62.170.126	2001:07F9:0400:0001:0001::632	2001:07F9:0400:0001:0001::631	2005-10-24 20:54:58

Figura 3.4 Listado de túneles creados

Como se puede apreciar desde el año 2005 vengo trabajando con este método de una forma muy cómoda , segura y escalable.

Hasta aquí ya tenemos asegurado la asignación de una IPv6 comercial y el túnel respectivo para salir a la nube comercial soportado en IPv6, en adelante nos abocaremos a la configuración del router / gateway.

3.2 Configuración del túnel

Ahora viene la etapa interesante , configurar nuestro router , como les mencione antes estoy empleando una PC que actuara como router.

Para definir antes que nada , GNU/Linux o mas comúnmente conocido como Linux es un sistema operativo libre de empleo , configuración y mejoramiento , nacido en 1991 como una tesis de maestría de un finlandes cuyo nombre es Linus Tolvards proviene de MINIX que es una versión educativa de UNIX , adaptado para procesadores de 32 bits 1386 , que fue justamente la base de todos los procesadores de la década del 90.

Al tratarse de software libre instalamos con gran facilidad software GNU y lo podremos adaptar y trabajando a nuestro propio gusto y exigencia.

Bueno , actualmente Linux viene con kernel en versiones 2.6.x los cuales ya viene soportado en el núcleo IPv6 por lo que ya no es necesario compilarlo para adaptarlo a lo que queremos, liberalmente el tema es mas sencillo y no tan complejo cuando trabaje en el 2002 en INICTEL al realizar la primera maqueta en Perú.

Para comprobar si nuestro Linux tiene el modulo IPv6 cargado lo comprobamos de esta forma :

```
[root@sharp ~]# cat /proc/net/if_inet6
00000000000000000000000000000001 01 80 10 80       lo
fe8000000000000240d0fffe6acc46 02 40 20 80          eth0
```

donde la primer linea nos indica la dirección local y la segunda la dirección asignada a la interfaz que la mac adress.

Si no nos sale nada , habría que cargar el modulo :

modprobe ipv6

Y lo verificamos la carga de esta forma :

```
[root@sharp ~]# lsmod |grep -w 'ipv6' && echo "Módulo IPv6 cargado"
ipv6                  221660  14
"Módulo IPv6 cargado"
```

probando conectividad :

```
[root@sharp ~]# ping6 ::1
PING ::1(::1) 56 data bytes
64 bytes from ::1: icmp_seq=1 ttl=64 time=0.056 ms
64 bytes from ::1: icmp_seq=2 ttl=64 time=0.072 ms
64 bytes from ::1: icmp_seq=3 ttl=64 time=0.070 ms
```

Ahora en Linux elegimos una distribucion en este caso es Centos 5.2 que es el clon de Redhat Enterprice 5.2

La instalacion no es el motivo del trabajo , solo de la configuración de los archivos para establecer el túnel.

Antes observamos la configuración de nuestra interfaz de red de nuestro gateway que a partir de ahora la denominaremos LAB , puesto que el host en mención es conocido en la red como : **lab.nitcom.com**

para lo cual ejecutamos un ifconfig eth0 , el resultado es el siguiente :

```
[root@lab ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:10:5A:00:C5:E8
          inet addr:192.168.1.2  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::210:5aff:fe00:c5e8/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2988474 errors:6013 dropped:0 overruns:0 frame:9051
          TX packets:2806638 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:883340576 (842.4 MiB)  TX bytes:1513894280 (1.4 GiB)
```

Bueno tenemos estos datos :

IPv4 Local : 200.60.195.7

IPv4 Remoto : 213.172.34.125

IPv6 Local : 2A01:0048:0100:0001:0001::032

IPv6 Remoto : 2A01:0048:0100:0001:0001::031

Network Prefijo : 2A01:0048:0202::/48

Con el segundo datos y cuarto editamos los siguientes archivos , cabe notar hasta este momento que yo me encuentro detrás de un módem ADSL , por el cual la IP publica esta establecida en el módem y LAB cuenta con una privada la cual no es impedimento para establecer el túnel como verán ahora.

Editamos los archivos :

[root@lab ~]# nano /etc/sysconfig/network

```
NETWORKING=yes
HOSTNAME=lab.nitcom.com
GATEWAY=192.168.1.1
NETWORKING_IPV6=yes
IPV6_DEFAULTDEV=sit1
IPV6FORWARDING=yes
```

[root@lab ~]# nano /etc/sysconfig/network-scripts/ifcfg-sit1

```
#Consulintel
DEVICE="sit1"
BOOTPROTO="static"
ONBOOT="yes"
IPV6INIT="yes"
IPV6TUNNELIPV4="213.172.34.125"
IPV6ADDR="2A01:0048:0100:0001:0001::32"
IPV6FORWARDING=yes
IPV6_AUTOCONF=no
```

Con esas dos configuraciones ya deberíamos estar en capacidad de conectarnos a la red comercial de ipv6 , para empezar hacemos prueba de conectividad hacia el otro extremo del túnel de ahí hacia un nodo de la red ipv6 y luego la traza obtenida hasta llegar hasta allá, previo reinicio de nuestro OS por única vez para que cargue todos los controladores y la configuración obtenemos.

Verificamos de la interfaz habilitada :

[root@lab ~]# ifconfig sit1

```
sit1      Link encap:IPv6-in-IPv4
          inet6 addr: 2a01:48:100:1:1::32/64 Scope:Global
          inet6 addr: fe80::c0a8:102/64 Scope:Link
          UP POINTOPOINT RUNNING NOARP  MTU:1480  Metric:1
          RX packets:473 errors:0 dropped:0 overruns:0 frame:0
          TX packets:580 errors:227 dropped:0 overruns:0 carrier:227
          collisions:0 txqueuelen:0
          RX bytes:153889 (150.2 KiB)  TX bytes:62097 (60.6 KiB)
```

Prueba local :

```
[root@lab ~]# ping6 2a01:48:100:1:1::31
PING 2a01:48:100:1:1::31(2a01:48:100:1:1::31) 56 data bytes
64 bytes from 2a01:48:100:1:1::31: icmp_seq=0 ttl=64 time=203 ms
64 bytes from 2a01:48:100:1:1::31: icmp_seq=1 ttl=64 time=203 ms
64 bytes from 2a01:48:100:1:1::31: icmp_seq=2 ttl=64 time=205 ms
```

Prueba al extremo remoto :

```
[root@lab ~]# ping6 2a01:48:100:1:1::32
PING 2a01:48:100:1:1::32(2a01:48:100:1:1::32) 56 data bytes
64 bytes from 2a01:48:100:1:1::32: icmp_seq=0 ttl=64 time=0.065 ms
64 bytes from 2a01:48:100:1:1::32: icmp_seq=1 ttl=64 time=0.078 ms
64 bytes from 2a01:48:100:1:1::32: icmp_seq=2 ttl=64 time=0.074 ms
```

Prueba a un nodo remoto :

```
[root@lab ~]# ping6 ipv6.google.com
PING ipv6.google.com(2001:4860:0:2001::68) 56 data bytes
64 bytes from 2001:4860:0:2001::68: icmp_seq=0 ttl=58 time=313 ms
64 bytes from 2001:4860:0:2001::68: icmp_seq=1 ttl=58 time=313 ms
64 bytes from 2001:4860:0:2001::68: icmp_seq=2 ttl=58 time=313 ms
```

Veremos los saltos hasta llegar a ese nodo remoto :

```
[root@lab ~]# traceroute6 ipv6.google.com
traceroute to ipv6.google.com (2001:4860:0:2001::68), 30 hops max, 40 byte packets
 1  2a01:48:100:1:1::31 (2a01:48:100:1:1::31)  203.757 ms  207.032 ms  209.446 ms
 2  bbr01-p6-0.lndn01.occaid.net (2001:4830:d1:10::1)  244.401 ms  246.327 ms  248.247 ms
 3  bbr01-p1-0.nwrk01.occaid.net (2001:4830:fe:1010::2)  321.650 ms  324.072 ms  326.256 ms
 4  bbr01-g1-0.asbn01.occaid.net (2001:4830:ff:f150::2)  338.543 ms  339.491 ms  340.702 ms
 5  pr61.iad07.net.google.com (2001:504:0:2:0:1:5169:1)  343.848 ms  346.032 ms  348.460 ms
```

Como podermos ver hasta aqui todo trabaja conforme lo esperado , con 2 configuraciones sencillas ya podermos establecer acceso a una red comercial IPv6 , ahora veremos la tabla de rutas creada en nuestra interfaz :

```
[root@lab ~]# route -A inet6 | grep sit1
2001:608:6:6::10/128       2001:608:6:6::10        UC    0      5        0 sit1
z.nic.de/128               z.nic.de                UC    0      5        0 sit1
2a01:48:100:1::/64         *                       U     256    5        0 sit1
fe80::/64                  *                       U     256    0        0 sit1
*/0                        *                       U     1      0        0 sit1
```

Que es lo esperado , nuestro router ya esta aprendiendo rutas y lo va agregando a su tabla conforme empecemos a trabajar, el objetivo del capitulo fue cubierto conforme lo esperado ahora vayamos un poco mas adelante.

3.3 Configurando el gateway para la red LAN

Probemos ahora un tema que lo estudiamos en la parte teoría que es que con IPv6 los hosts de una red local que lo definamos no requieren configurar una dirección IPv6 , solo requieren tener el modulo habilitado para el sistema operativo en mención o que se conecte a la red local y una conexión física al gateway LAB.

Ahora en el router es necesario instalar un modulo que establezca la autoconfiguración una forma podría ser el dhcp6 y otra es emplear el **radvd** que es mi preferido por su velocidad de asignación de IP y su robustez

Seguimos los siguientes pasos para establecer la autoconfiguración :

Instalamos el radvd

```
[root@lab~]# yum install radvd
Setting up Install Process
Setting up repositories
Reading repository metadata in from local files
Parsing package install arguments
Resolving Dependencies
================================================================================
 Package         Arch      Version        Repository      Size
================================================================================
Installing:
 radvd           i386      0.9.1-4        base            63 k
Transaction Summary
================================================================================
Install      1 Package(s)
Downloading Packages:
(1/1): radvd-0.9.1-4.i386 100% |=========================|  63 kB    00:05
Running Transaction Test
Finished Transaction Test
Transaction Test Succeeded
Running Transaction
Installing: radvd                ########################### [1/1]
```

Configuramos el radvd con el segmento que nos asigno el Task Force .

[root@lab ~]# nano /etc/radvd.conf

```
interface eth0
{
    AdvSendAdvert on;
    AdvLinkMTU 1280;
    MaxRtrAdvInterval 300;
    prefix 2a01:48:202:1000::/64
    {
        AdvOnLink on;
        AdvAutonomous on;
    };
```

Levantamos el demonio en el gateway y tenemos toda la red local ya listo para ser asignada con una dirección IPv6

```
[root@lab ~]# /etc/init.d/radvd start
```

Para probar esta maravilla , conectamos nuestra portátil en Fedora 9 que tiene el modulo habilitado por defecto y al hacer un ifconfig obtenemos esto :

```
[root@sharp ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:40:D0:6A:CC:46
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2a01:48:202:1000:240:d0ff:fe6a:cc46/64 Scope:Global
          inet6 addr: fe80::240:d0ff:fe6a:cc46/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1821149 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1328376 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2151328113 (2.0 GiB)  TX bytes:157259985 (149.9 MiB)
```

He sombreado la dirección asignada y la comparo con al prefijo de red que asigne en el archivo radvd

Cabe aclarar aqui que el Task Force nos ha asignado :

2a01:0048:0202::/48

Para mi red local he tomado una parte del segmento y le asigne de esta forma :

2a01:0048:0202:1000::/64

Ahora notamos la dirección asignada en forma automática a mi host de la red local :

2a01:48:202:1000:240:d0ff:fe6a:cc46

Que es justamente una dirección 6over4 que lo confirma el prefijo de red y la mac address

Ahora desde nuestro host probamos conectividad a un hosts remoto.

Ahora lo único que nos falta es que nuestro gateway LAB pueda oir a mi host denominado SHARP , hasta este momento la interfaz eth0 de LAB no tiene asignada ninguna dirección IPv6 por el cual no oirá o conversara a nivel IP con el SHARP y por lo cual no establecer el ruteo esperado.

Para lo cual le asignamos una direccion IPv6 respetando el prefijo que asignamos a nuestra red local :

[root@lab ~]# nano /etc/sysconfig/network-scripts/ifcfg-eth0

```
# 3Com Corporation 3c905B 100BaseTX [Cyclone]
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
HWADDR=00:10:5a:00:c5:e8
NETMASK=255.255.255.0
NETWORK=192.168.1.0
IPADDR=192.168.1.2
BROADCAST=192.168.1.255
TYPE=Ethernet
GATEWAY=192.168.1.1
IPV6INIT="yes"
IPV6ADDR="2A01:48:202:1000::1"
```

Sombreo la dirección que he asignado y noto que ya pertenece al segmento de red de mi red local es decir : /64 por lo cual ahora desde mi host SHARP puede ejecutar un ping y notare la respuesta .

Desde LAB : *ifconfig down y luego ifconfig up*

Desde SHARP :

```
[root@sharp ~]# ping6 2A01:48:202:1000::1
PING 2A01:48:202:1000::1(2a01:48:202:1000::1) 56 data bytes
64 bytes from 2a01:48:202:1000::1: icmp_seq=1 ttl=64 time=1.42 ms
64 bytes from 2a01:48:202:1000::1: icmp_seq=2 ttl=64 time=0.230 ms
64 bytes from 2a01:48:202:1000::1: icmp_seq=3 ttl=64 time=0.265 ms
64 bytes from 2a01:48:202:1000::1: icmp_seq=4 ttl=64 time=0.209 ms
```

Ahora solo nos queda probar conectividad hacia un host externo :

```
[root@sharp ~]# ping6 ipv6.google.com
PING ipv6.google.com(2001:4860:0:2001::68) 56 data bytes
64 bytes from 2001:4860:0:2001::68: icmp_seq=1 ttl=57 time=313 ms
64 bytes from 2001:4860:0:2001::68: icmp_seq=2 ttl=57 time=314 ms
64 bytes from 2001:4860:0:2001::68: icmp_seq=3 ttl=57 time=331 ms
```

Observamos ahora la traza desde mi host para llevar hasta ese host :

```
[root@sharp ~]# traceroute6 ipv6.google.com
traceroute to ipv6.google.com (2001:4860:0:2001::68), 30 hops max, 40 byte packets
 1  2a01:48:202:1000::1 (2a01:48:202:1000::1)  0.268 ms  0.181 ms  0.148 ms
 2  2a01:48:100:1:1::31 (2a01:48:100:1:1::31)  204.327 ms  206.920 ms  209.239 ms
 3  bbr01-p6-0.lndn01.occaid.net (2001:4830:d1:10::1)  243.185 ms  245.867 ms  248.287 ms
 4  bbr01-p1-0.nwrk01.occaid.net (2001:4830:fe:1010::2)  321.678 ms  324.330 ms  334.296 ms
 5  bbr01-g1-0.asbn01.occaid.net (2001:4830:ff:f150::2)  336.124 ms  339.185 ms  341.069 ms
 6  pr61.iad07.net.google.com (2001:504:0:2:0:1:5169:1)  343.531 ms  346.025 ms  347.682 ms

 7  2001:4860:0:2001::68 (2001:4860:0:2001::68)  320.381 ms  313.626 ms  314.224 ms
```

Sombreamos el salto #1 que es justamente nuestro gateway

Por lo cual ya garantizamos que estamos en el mundo IPv6 pasando por nuestro Router / Gateway LAB , ahora hacermos pruebas sencillas de acceso como la de apreciar la web de google en IPv6 que es : http://ipv6.google.com

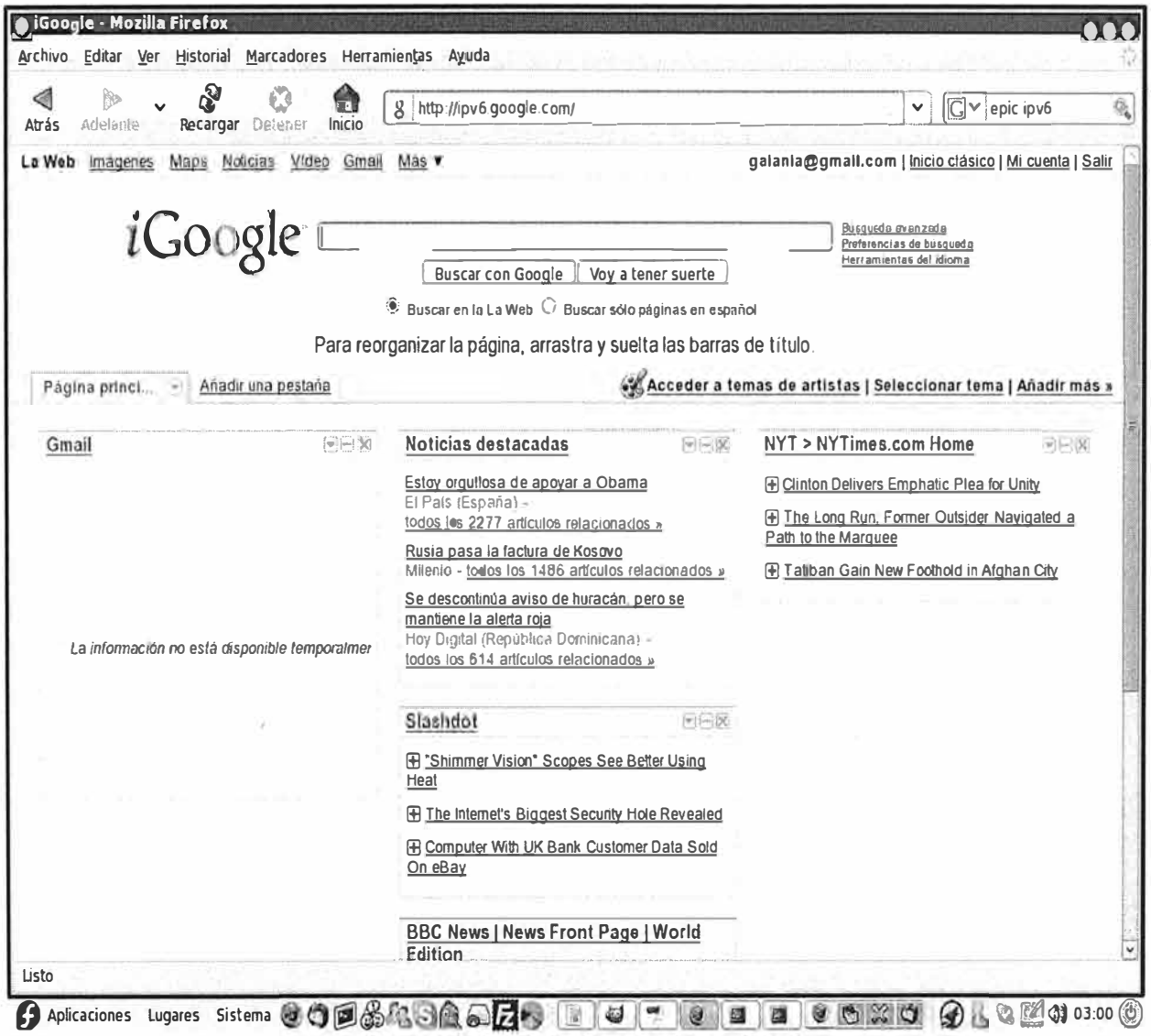


Figura 3.5 Web de google en IPv6

Y la prueba fundamental accedes a http://www.kame.net y ver si la tortuguita navega si lo hace quiere decir : "welcome IPV6"

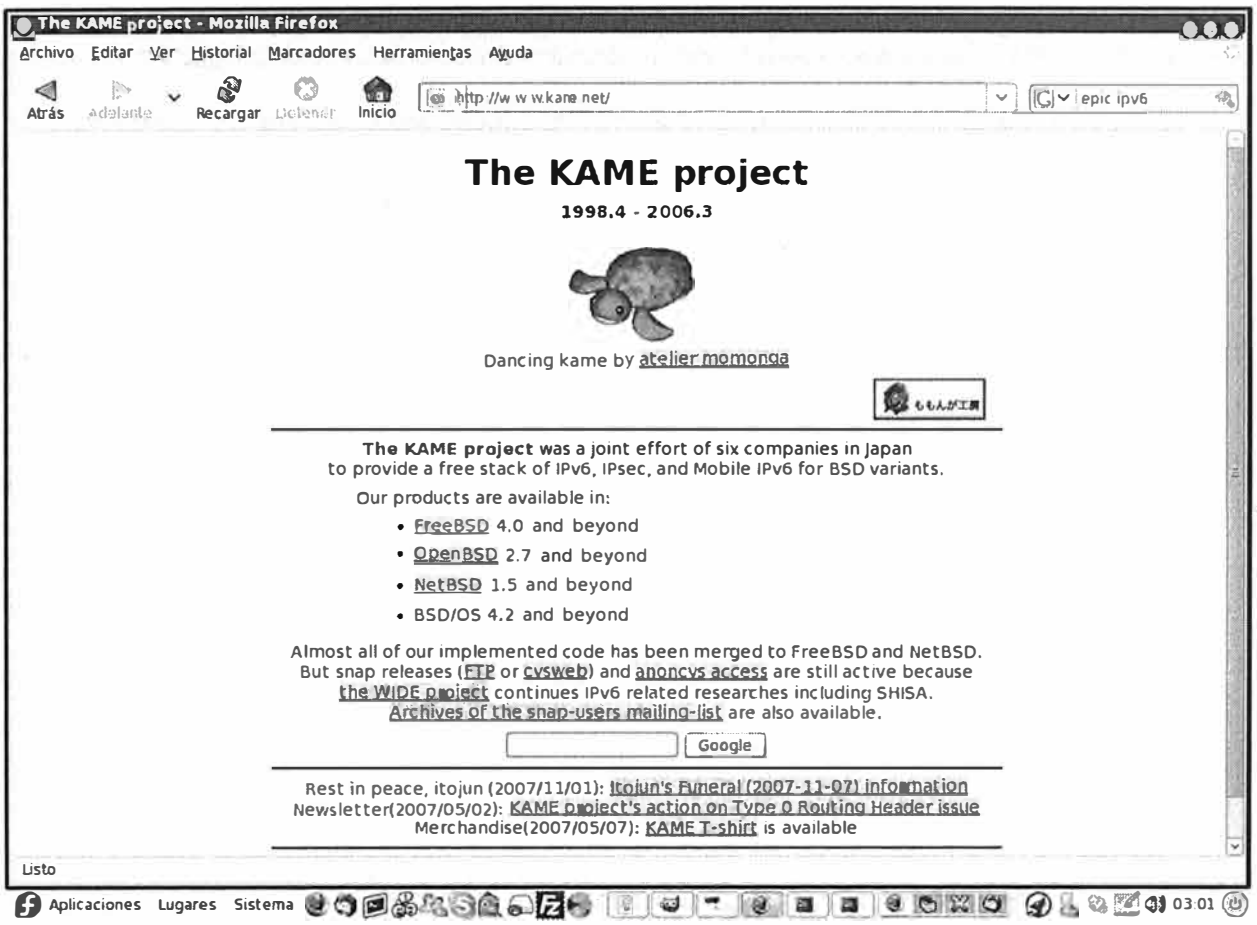


Figura 3.6 Web de kame.net , prueba fundamental de estar en el mundo IPv6

Con esas pruebas hemos probado conectividad y que podemos navegar es decir emplear el puerto 80 , pero seamos mas ambiciosos y probemos IRC por IPv6 , para lo cual en nuestro host SHARP instalamos el XCHAT que es un cliente IRC que soporta IPv6 y nos conectamos a una red que soporta IPv6 como puede ser freenode , consultando la web de freenode observamos que tiene 4 servidores que estan soportando trafico IPv6

http://freenode.net/irc_servers.shtml

goethe.freenode.net

calkins.freenode.net

crichton.freenode.net

weber.freenode.net

Para nuestro ejemplo accedemos a calkins

/server calkins.freenode.net y accedemos desde la consola de comandos ıngresamos , ejecutamos un whois obtenemos esto :

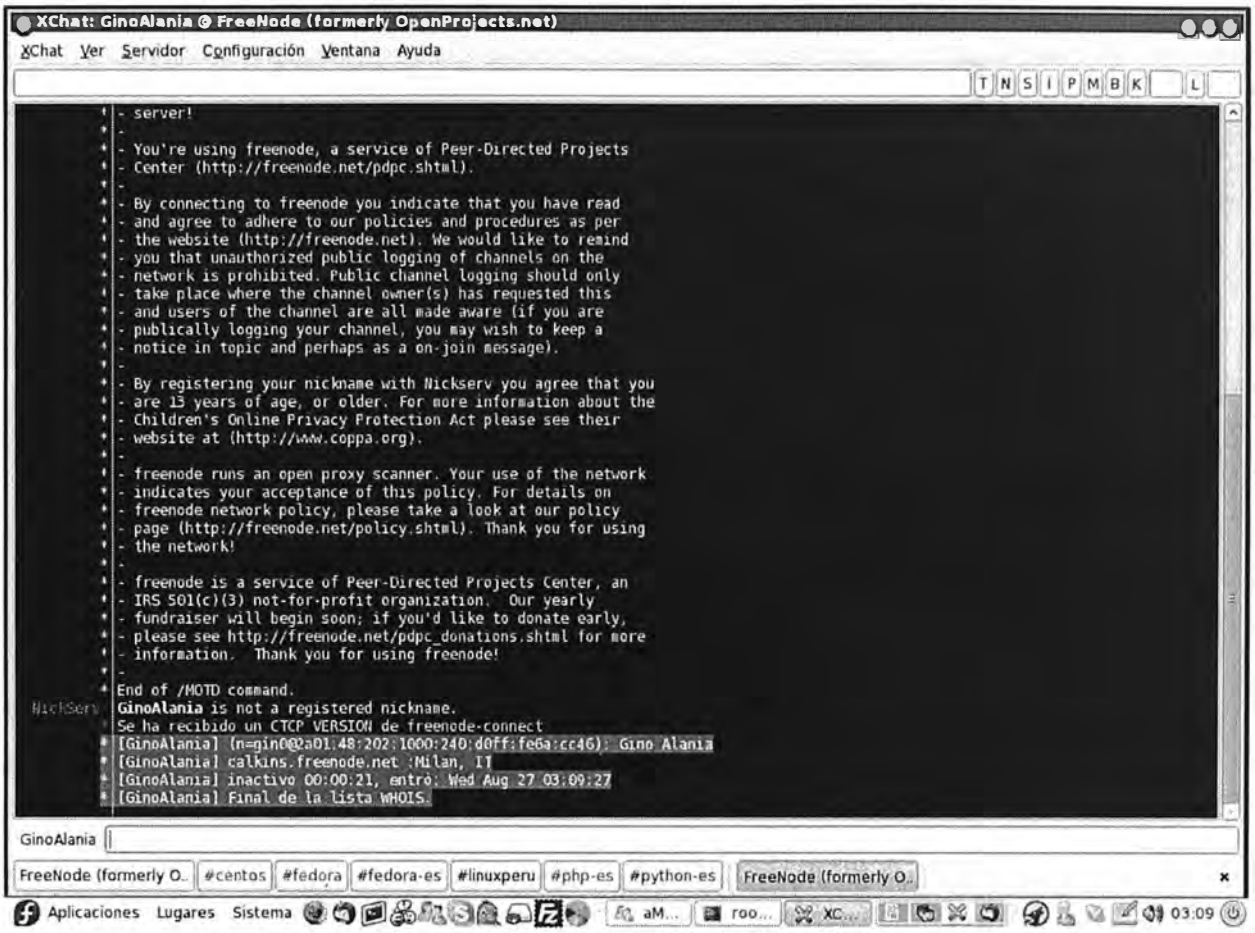


Figura 3.7 Acceso a IRC sobre IPv6

3.4 Establecimiento de túneles IPv6 / IPv4

Hasta aquí ya contamos con una red local nativa en IPv6 que tiene conexión a la nube IPv6 comercial , ahora veamos el siguiente paso que es la de implementar túneles con otros host en la nube Internet y que seamos nosotros el by pass para los mismos , para lo cual nuestro Router establecerá un ruteo de este mismo trafico y ira incrementando sus tablas de rutas, la idea de esta capitulo es establecer encaminamiento con otros hosts y administrarlo es un tema especial aplicado a un ISP .

Para esta etapa elegimos un host también con IPv4 publico y estático , el cual como veremos ensayaremos el mismo mecanismos que efectua Task Force .

Host2 remoto : IPv4 : 200.60.122.131

En Lab creamos una interfaz llamada sit2000 , notamos que elegimos un numero que acompaña a sit en referencia al segmento que emplearemos para nuestros tuneles que sera el 2000 (1000 para la red local)

[root@lab ~]# nano /etc/sysconfig/network-scripts/ifcfg-sit2000

```
DEVICE="sit2000"
BOOTPROTO="none"
ONBOOT="yes"
IPV6INIT="yes"
IPV6TUNNELIPV4="200.60.122.131"
```

Observemos lo que definimos aquí solo dos cosas ; el IPv4 remoto que sera del Host2 y una dirección IPv6 a la interfaz sit2000 , por facilidad asignaríamos :2 para el host 2 remoto.

Ahora vayamos al host2 remoto pero aquí la configuración podríamos hacer de la misma forma con la que hicimos con LAB , pero aquí hemos preferido variar y emplear un script similar al enviado por Task Force para probar que existen alternativas de configuración

En nuestro host remoto creamos un archivo de esta forma :

[root@telefonica ~]# nano /etc/rc.d/nitcom

```
ip tunnel add nitcom mode sit remote 200.60.195.7 local 200.60.122.131 ttl 64
ip link set nitcom up mtu 1420
ip address add 2A01:48:202:2000::2/48 dev nitcom
ip route add 2000::/3 dev nitcom
ip route add 2001::/3 dev nitcom
ip route add 2A01::/3 dev nitcom
```

Expliquemos brevemente lo que significa cada linea

La primera linea es el comando ip que es parte de iproute2 que ya viene implementado en toda distribucion de Linux , para lo cual esa linea establece un túnel para lo cual define el host local y el remoto a través de los IPv4.

La segunda linea es el alta de la interfaz del túnel que llamamos nitcom en referencia al host que es donde le provee el acceso a la red comercial IPv6 pero con un MTU de 1420 se entiende que limitamos el MTU por que estos viajaran sobre paquetes IPv4 en nuestro caso por ser ADSL y estar configurado el módem por PPPoE el MTU es 1500 por lo tanto para evitar fragmentación en los datagramas IPv4 lo hacemos limitar en IPv6 a fin que la fragmentación se haga a este nivel y no generemos una carga innecesaria en IPv4.

La tercera linea es la asignación de la dirección IPv6 a la interfaz y finalmente declaramos las rutas que puede ubicar la interfaz como cuarta y demás lineas.

Ahora hacemos que nuestro script sea ejecutable y listo

[root@telefonica ~]# chmod 700 /etc/rc.d/nitcom

Lo ejecutamos : [root@telefonica ~]# ./etc/rc.d/nitcom

Y listo ...

Ahora probamos conectividad con el host ipv6.google.com y observamos

```
[root@telefonica ~]# ping6 ipv6.google.com
PING ipv6.google.com(2001:4860:0:2001::68) 56 data bytes
64 bytes from 2001:4860:0:2001::68: icmp_seq=0 ttl=57 time=329 ms
64 bytes from 2001:4860:0:2001::68: icmp_seq=1 ttl=57 time=329 ms
64 bytes from 2001:4860:0:2001::68: icmp_seq=2 ttl=57 time=330 ms
```

Como se notara hay un pequeño delay justamente producido por que el trafico del host2 llega a LAB y este a su vez hacia la red de Task Force , recuerdan que desde LAB era 313 por una aritmética simple obtenemos que el retardo entre LAB y el HOST2 sera de 16 ms.

Aquí un pantallazo comparando en la parte superior el delay desde el HOST2 y en el otro cuadro desde nuestro host hacia la IPv4 del HOST2

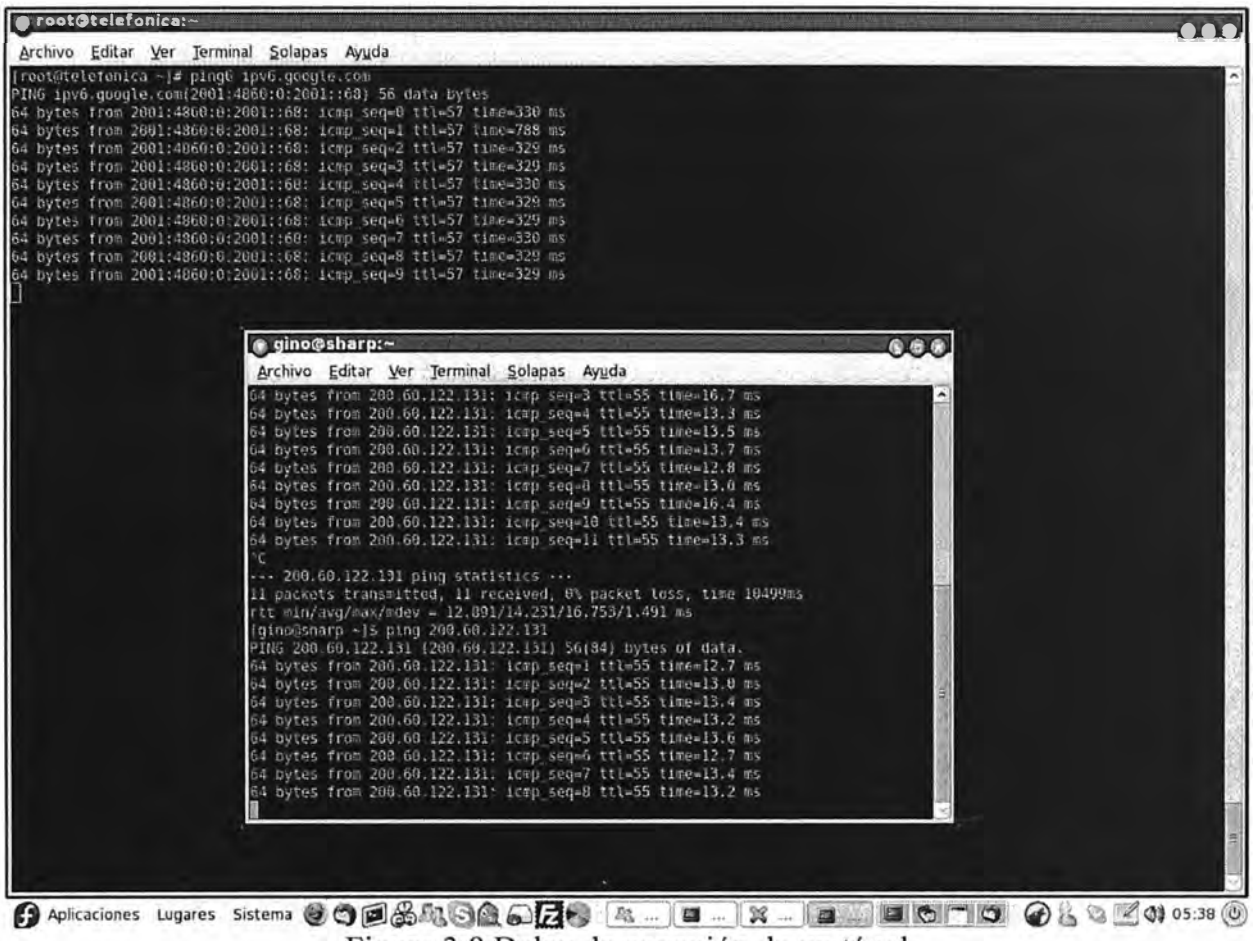


Figura 3.8 Delay de conexión de un túnel

Por lo que mostramos hasta el momento lo aplicamos con un determinado host denominado HOST2 de la misma forma puede existir un HOSTn para lo cual demostramos que es posible establecer un mecanismo de transición de IPv4 a IPv6 sobre la red de Internet actual .

Finalmente lograsmos nuestro objetivo de establecer tuneles con diferentes hosts y ser un ISP de acceso a la nube IPv6 , el presente grafico nos mostrata un proyecto voluntario plasmado en :

http://ipv6.nitcom.com

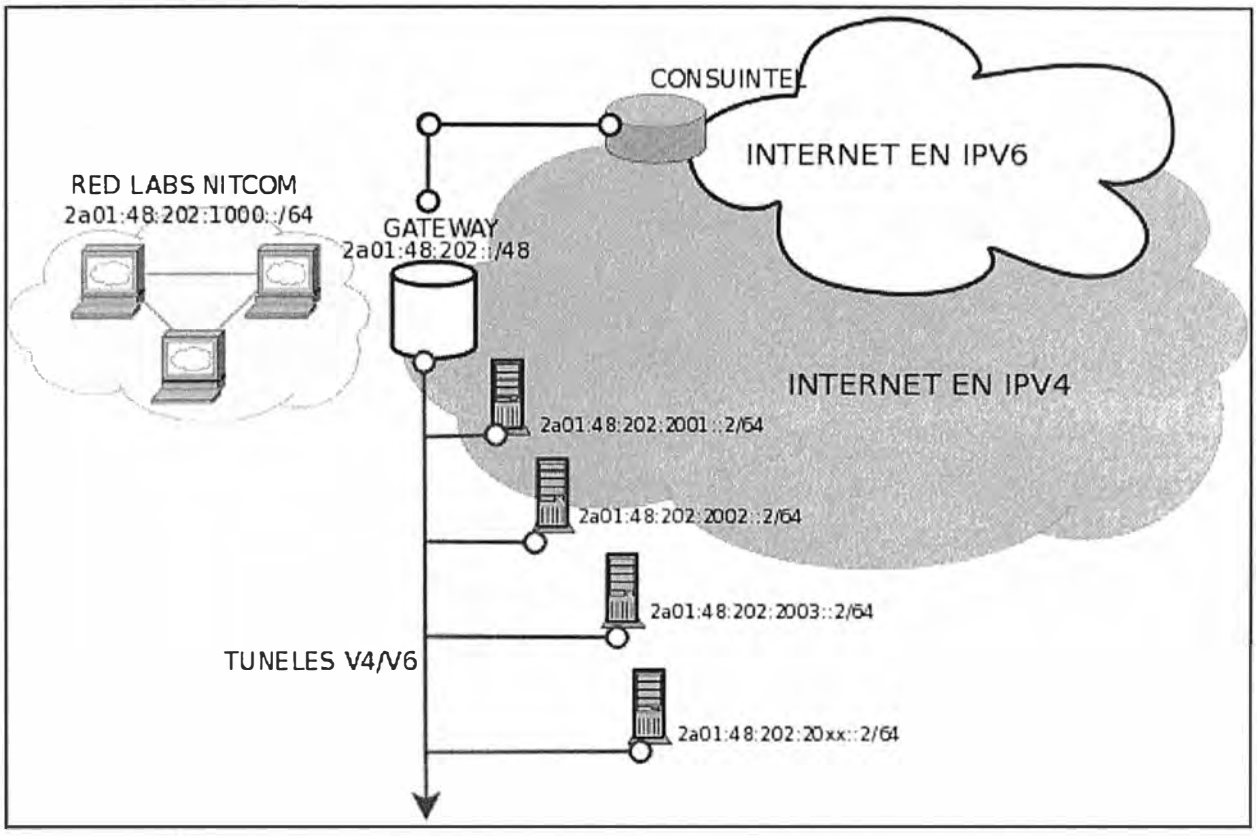


Figura 3.9 Topología de la maqueta implementada

La web del proyecto que es el mencionado para el presente trabajo de investigación

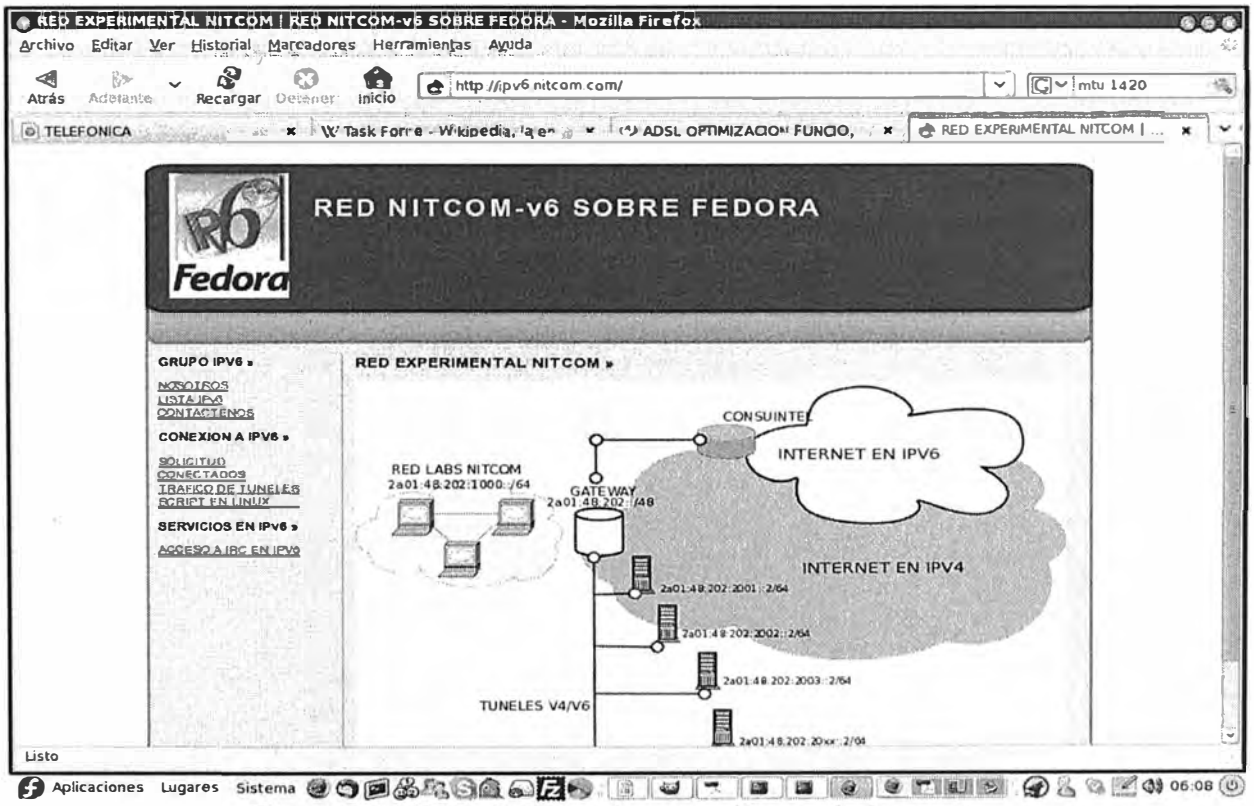


Figura 3.10 Web de la implementación en la Internet

CAPITULO IV
GESTION DE TRAFICO Y MIGRACION DE SERVICIOS

Aquí viene finalmente la parte concluyente de nuestro trabajo que es la gestión de los túneles que hemos efectuado , para lo cual nos valdremos de herramientas empleadas en software libre y el protocolo SNMP a fin de cumplir con el objetivo de ser un ISP .

4.1 Teoría de SNMP

El Protocolo Simple de Administración de Red o SNMP es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el desempeño de la red, buscar y resolver sus problemas, y planear su crecimiento.

Las versiones de SNMP más utilizadas son dos: SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2). Ambas versiones tienen un número de características en común, pero SNMPv2 ofrece mejoras, como por ejemplo, operaciones adicionales.

SNMP en su última versión (SNMPv3) posee cambios significativos con relación a sus predecesores, sobre todo en aspectos de seguridad, sin embargo no ha sido mayoritariamente aceptado en la industria.

4.1.1Componentes básicos

Una red administrada a través de SNMP consiste de tres componentes claves:

- Dispositivos administrados;
- Agentes;
- Sistemas administradores de red (NMS's).

Un dispositivo administrado es un nodo de red que contiene un agente SNMP y reside en una red administrada. Estos recogen y almacenan información de administración, la cual

es puesta a disposición de los NMS's usando SNMP. Los dispositivos administrados, a veces llamados elementos de red, pueden ser routers, servidores de acceso, switches, bridges, hubs, computadores o impresoras.

Un agente es un módulo de software de administración de red que reside en un dispositivo administrado. Un agente posee un conocimiento local de información de administración (memoria libre, número de paquetes IP recibidos, rutas, etcétera), la cual es traducida a un formato compatible con SNMP y organizada en jerarquías.

Un NMS ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados. Los NMS's proporcionan el volumen de recursos de procesamiento y memoria requeridos para la administración de la red. Uno o más NMS's deben existir en cualquier red administrada.

4.1.2 Comandos Básicos

Los dispositivos administrados son supervisados y controlados usando cuatro comandos SNMP básicos: lectura, escritura, notificación y operaciones transversales.

El comando de lectura es usado por un NMS para supervisar elementos de red. El NMS examina diferentes variables que son mantenidas por los dispositivos administrados.

El comando de escritura es usado por un NMS para controlar elementos de red. El NMS cambia los valores de las variables almacenadas dentro de los dispositivos administrados.

El comando de notificación es usado por los dispositivos administrados para reportar eventos en forma asíncrona a un NMS. Cuando cierto tipo de evento ocurre, un dispositivo administrado envía una notificación al NMS.

Las operaciones transversales son usadas por el NMS para determinar qué variables soporta un dispositivo administrado y para recoger secuencialmente información en tablas de variables, como por ejemplo, una tabla de rutas.

4.1.3 MIB

Una Base de Información de Administración (MIB) es una colección de información que está organizada jerárquicamente. Las MIB's son accedidas usando un protocolo de administración de red, como por ejemplo, SNMP.

Un objeto administrado (algunas veces llamado objeto MIB, objeto, o MIB) es uno de cualquier número de características específicas de un dispositivo administrado. Los

objetos administrados están compuestos de una o más instancias de objeto, que son esencialmentevariables.

Existen dos tipos de objetos administrados: Escalares y tabulares. Los objetos escalares definen una simple instancia de objeto. Los objetos tabulares definen múltiples instancias de objeto relacionadas que están agrupadas conjuntamente en tablas MIB.

Un ejemplo de un objeto administrado es atInput, que es un objeto escalar que contiene una simple instancia de objeto, el valor entero que indica el número total de paquetes AppleTalk de entrada sobre una interfaz de un router.

Un identificador de objeto (object ID) únicamente identifica un objeto administrado en la jerarquía MIB. La jerarquía MIB puede ser

representada como un árbol con una raíz anónima y los niveles, que son asignados por diferentes organizaciones.

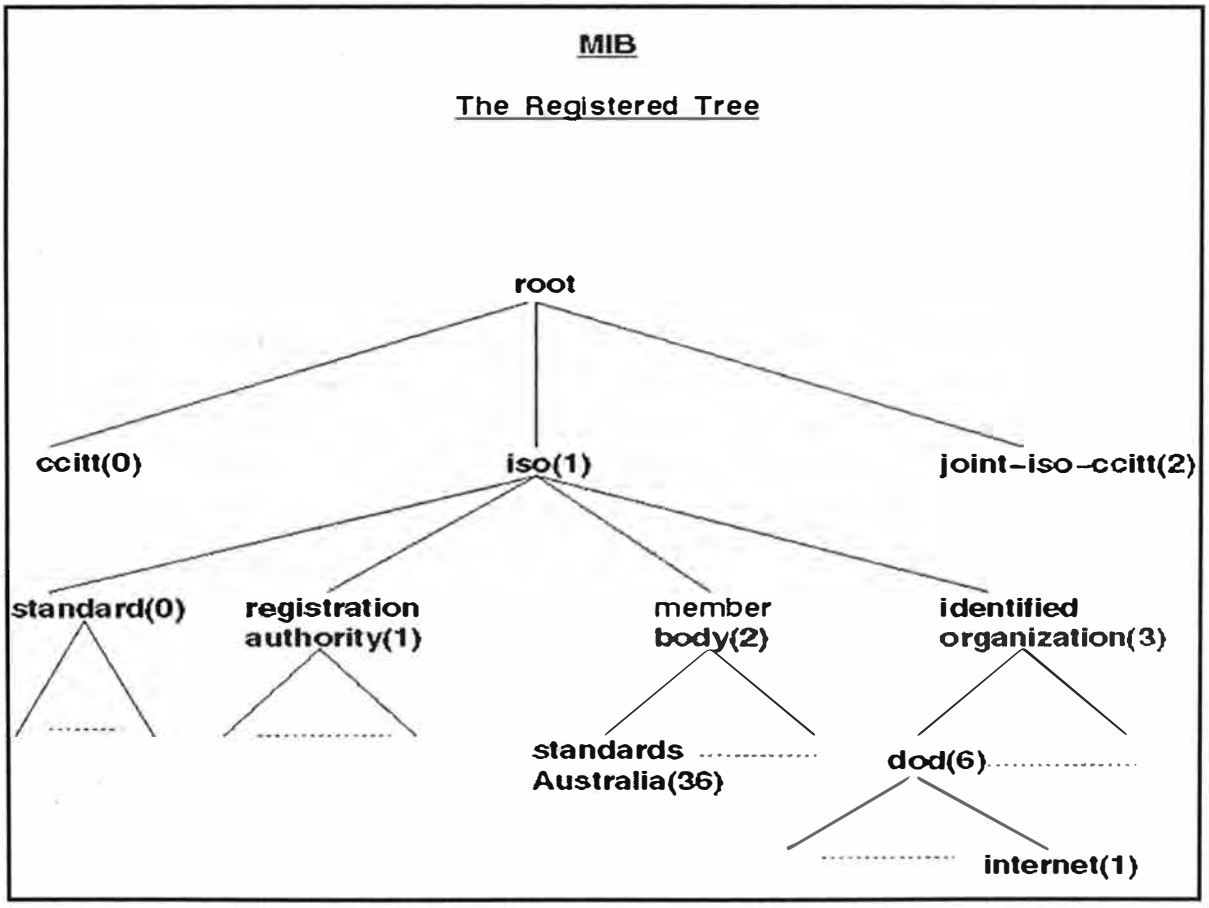


Figura 4.1 Árbol de la estructura de los Mibs

El árbol MIB ilustra las variadas jerarquías asignadas por las diferentes organizaciones

Los identificadores de los objetos ubicados en la parte superior del árbol pertenecen a

diferentes organizaciones estándares, mientras los identificadores de los objetos ubicados en la parte inferior del árbol son colocados por las organizaciones asociadas.

Los vendedores pueden definir ramas privadas que incluyen los objetos administrados para sus propios productos. Las MIB's que no han sido estandarizadas típicamente están localizadas en la rama experimental.

El objeto administrado atInput podría ser identificado por el nombre de objeto iso.identified-organization.dod.internet.private.enterprise.cisco.temporary.AppleTalk.atInput o por el descriptor de objeto equivalente 1.3.6.1.4.1.9.3.3.1.

El corazón del árbol MIB se encuentra compuesto de varios grupos de objetos, los cuales en su conjunto son llamados mib-2. Los grupos son los siguientes:

- System (1).
- Interfaces (2).
- AT (3).
- IP (4).
- ICMP (5).
- TCP (6).
- UDP (7).
- EGP (8).
- Transmission (10).
- SNMP (11).

Es importante destacar que la estructura de una MIB se describe mediante el estándar Notación Sintáctica Abstracta 1 (Abstract Syntax Notation One).

4.1.4 Mensajes SNMP

Para realizar las operaciones básicas de administración anteriormente nombradas, el protocolo SNMP utiliza un servicio no orientado a la conexión (UDP) para enviar un pequeño grupo de mensajes (PDUs) entre los administradores y agentes. La utilización de un mecanismo de este tipo asegura que las tareas de administración de red no afectarán al rendimiento global de la misma, ya que se evita la utilización de mecanismos de control y recuperación como los de un servicio orientado a la conexión, por ejemplo TCP.

Los puertos comúnmente utilizados para SNMP son los siguientes:

Numero	Descripcion
161	SNMP
162	SNMP-trap

Los paquetes utilizados para enviar consultas y respuestas SNMP poseen el siguiente formato:

- Versión: Número de versión de protocolo que se está utilizando (por ejemplo 1 para SNMPv1).
- Comunidad: Nombre o palabra clave que se usa para la autenticación. Generalmente existe una comunidad de lectura llamada "public" y una comunidad de escritura llamada "private".
- SNMP PDU: Contenido de la unidad de datos del protocolo, el que depende de la operación que se ejecute.

Los mensajes GetRequest, GetNextRequest, SetRequest y GetResponse utilizan la siguiente estructura en el campo SNMP PDU:

- Identificador: Es un número utilizado por el NMS y el agente para enviar solicitudes y respuesta diferentes en forma simultánea.
- Estado e índice de error: Sólo se usan en los mensajes GetResponse´(en las consultas siempre se utiliza cero). El campo "índice de error" sólo se usa cuando "estado de error" es distinto de 0 y posee el objetivo de proporcionar información adicional sobre la causa del problema. El campo "estado de error" puede tener los siguientes valores:
  - o 0: No hay error;
  - o 1: Demasiado grande;
  - o 2: No existe esa variable;
  - o 3: Valor incorrecto;
  - o 4: El valor es de solo lectura;
  - o 5: Error genérico.

- Enlazado de variables: Es una serie de nombres de variables con sus valores correspondientes (codificados en ASN.1).

4.1.5 GetRequest

A través de este mensaje el NMS solicita al agente retornar el valor de un objeto de interés mediante su nombre. En respuesta el agente envía una respuesta indicando el éxito o fracaso de la petición. Si la petición fue correcta, el mensaje resultante también contendrá el valor del objeto solicitado. Este mensaje puede ser usado para recoger un valor de un objeto, o varios valores de varios objetos, a través del uso de listas.

4.1.6 GetNextRequest

Este mensaje es usado para recorrer una tabla de objetos. Una vez que se ha usado un mensaje GetRequest para recoger el valor de un objeto, puede ser utilizado el mensaje GetNextRequest para repetir la operación con el siguiente objeto de la tabla. Siempre el resultado de la operación anterior será utilizado para la nueva consulta. De esta forma un NMS puede recorrer una tabla de longitud variable hasta que haya extraído toda la información para cada fila existente.

4.1.7 SetRequest

Este tipo de mensaje es utilizado por el NMS para solicitar a un agente modificar valores de objetos. Para realizar esta operación el este tipo de mensaje es utilizado por el NMS para solicitar a un agente modificar valores de objetos. Para realizar esta operación el NMS envía al agente una lista de nombres de objetos con sus correspondientes valores.

4.1.8 GetResponse

Este mensaje es usado por el agente para responder un mensaje GetRequest, GetNextRequest, o SetRequest. En el campo "Identificador de Request" lleva el mismo identificador que el "request" al que está respondiendo.

4.1.9 Trap

Una trap es generado por el agente para reportar ciertas condiciones y cambios de estado a un proceso de administración. Una trap es un mensaje espontáneo enviado por el Agente al Administrador, al detectar una condición predeterminada, como es la conexión/desconexión de una estación o una alarma. El formato de la PDU es diferente:

- Tipo Enterprise Dirección del agente Tipo genérico de trap Tipo específico de trap Timestamp Enlazado de variables.
- Enterprise: Identificación del subsistema de gestión que ha emitido el trap.
- Dirección del agente: Dirección IP del agente que ha emitido el trap.
- Tipo genérico de trap.
  - o Cold start (0): Indica que el agente ha sido inicializado o reinicializado.
  - o Warm start (1): Indica que la configuración del agente ha cambiado.
  - o Link down (2): Indica que una interfaz de comunicación se encuentra fuera de servicio (inactiva).
  - o Link up (3): Indica que una interfaz de comunicación se encuentra en servicio (activa).
  - o Authentication failure (4): Indica que el agente ha recibido un requerimiento de un NMS no autorizado (normalmentecontrolado por una comunidad).
  - o EGP neighbor loss (5): Indica que en sistemas en que los routers están utilizando el protocolo EGP, un equipo colindante se encuentra fuera de servicio;
  - o Enterprise (6): En esta categoría se encuentran todos los nuevos traps incluidos por los vendedores.
- Tipo específico de trap: Es usado para traps privados (de fabricantes), así como para precisar la información de un determinado trap genérico.
- Timestamp: Indica el tiempo que ha transcurrido entre la reinicialización del agente y la generación del trap.
- Enlazado de variables: Se utiliza para proporcionar información adicional sobre la causa del mensaje.

4.1.10 GetBulkRequest

Este mensaje es usado por un NMS que utiliza la versión 2 del protocolo SNMP típicamente cuando es requerida una larga transmisión de datos, tal como la recuperación de largas tablas. En este sentido es similar al mensaje GetNextRequest usado en la versión 1 del protocolo, sin embargo, GetBulkRequest es un mensaje que implica un método mucho más rápido y eficiente, ya que a través de un solo mensaje es posible solicitar la totalidad de la tabla.

4.1.11 InformRequest

Un NMS que utiliza la versión 2 del protocolo SNMP transmite un mensaje de este tipo a otro NMS con las mismas características, para notificar información sobre objetos administrados.

4.2 Instalación y Configuracion de SNMP en LAB

Para nuestro caso emplearemos la versión 2c en Linux el software que maneja snmp es el net-snmpd . Para lo cual en nuestro Router ejecutamos :

[root@lab data]# yum -y install net-snmpd mrtg gd

El cual el primero es el servidor que se encarga de gestionar el trafico , el segundo la herramienta de visualización y finalmente el ultimo las librerías que hacen posible mostrar gráficamente el trafico.

Una vez ejecutado la instalacion procedemos a configurar el net-snmpd , para lo cual ingresamos al archivo de configuración y establecemos la configuración como se muestra.

```
com2sec   local      localhost        nuke
com2sec6  mynetwork  2a01:48:100:1:1::32 nuke
group MyRWGroup v1       local
group MyRWGroup v2c      local
group MyRWGroup usm      local
group MyROGroup v1       mynetwork
group MyROGroup v2c      mynetwork
group MyROGroup usm      mynetwork
# Ramas MIB que se permiten ver
view all   included .1                    80
# Establece permisos de lectura y escritura
access MyROGroup ""     any     noauth   exact all   none  none
access MyRWGroup ""     any     noauth   exact all   all   none
# Informacion de Contacto del Sistema
syslocation Gino_Alania
syscontact Gino_Alania
```

Adicionalmente habilitamos el puerto para diferenciar cuando es IPv4 y cuando es IPv6 , lo habilitamos en el siguiente archivo :

[root@lab data]# nano /etc/snmp/snmpd.options

```
#Set cli options here
OPTIONS="udp:161,udp6:6161"
```

Finalmente :

[root@lab data]# /etc/init.d/snmpd restart

Parando snmpd: [OK]

Iniciando snmpd: [OK]

Y ya tenemos el servicio corriendo y listo para ser recogido por un gestor gráfico

Comprobamos el trafico de los tuneles creado en nuestro gateway , ejecutamos :

[root@lab data]# cat /proc/net/dev

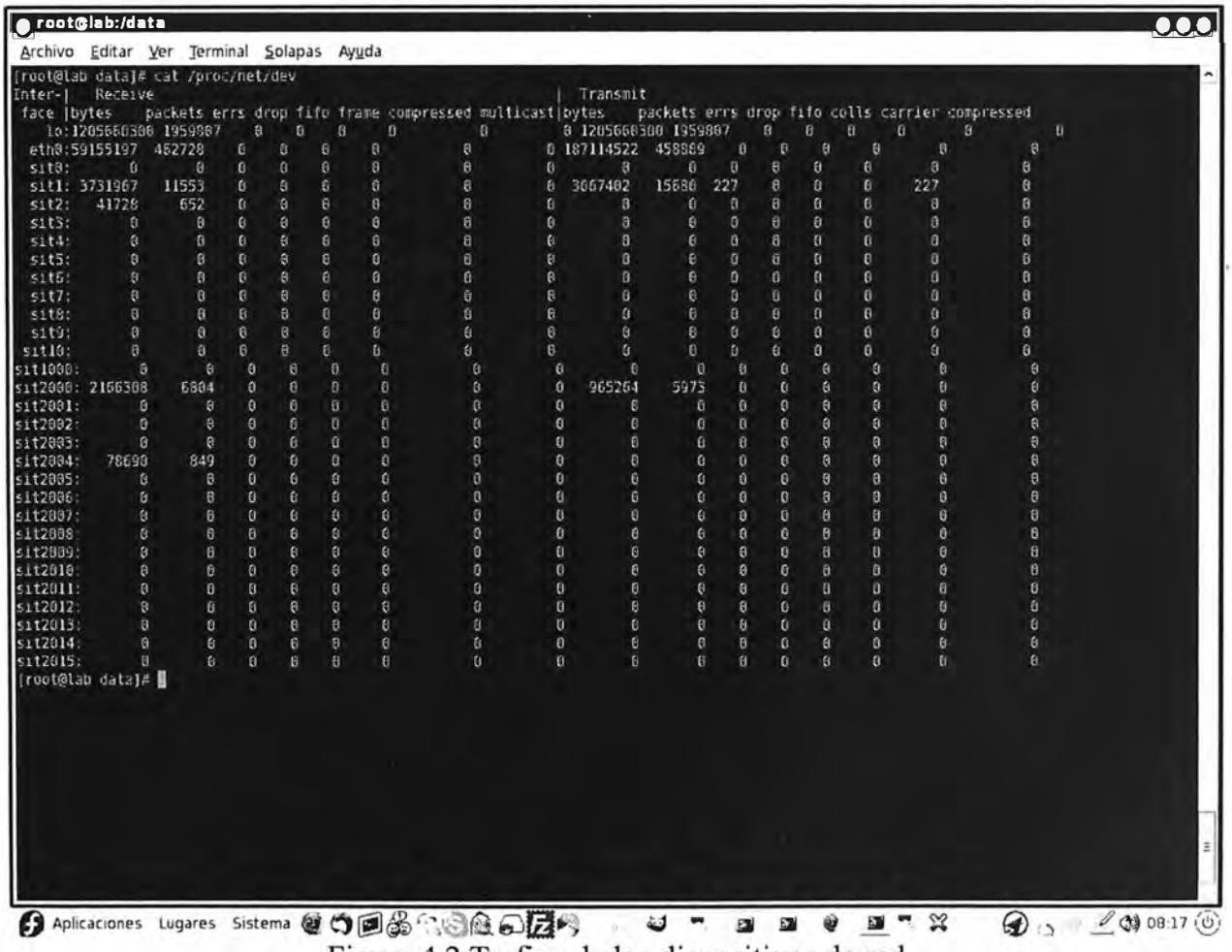


Figura 4.2 Trafico de los dispositivos de red

El resultado lo podemos apreciar en :

http://ipv6.nitcom.com/mrtg

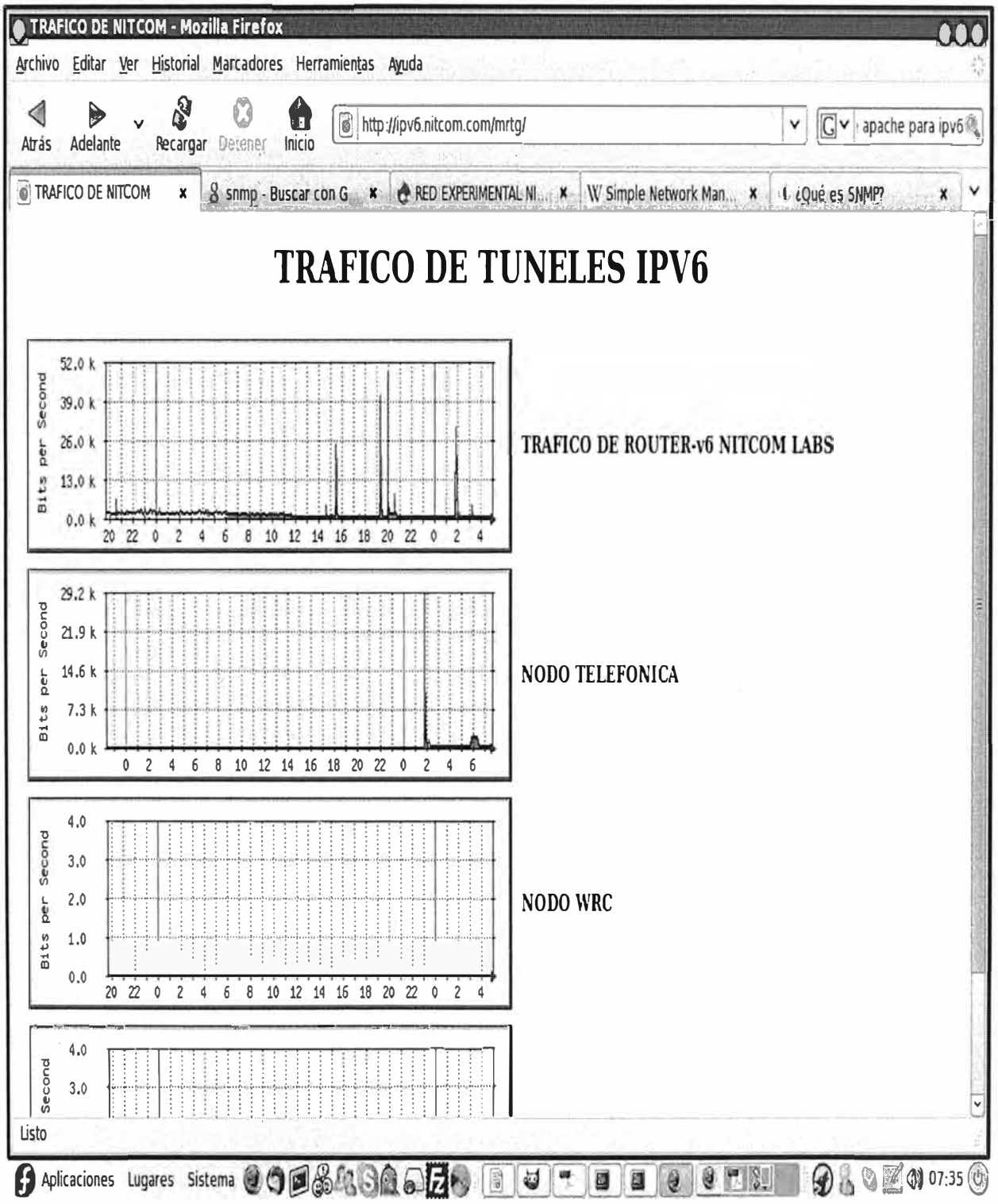


Figura 4.3 Gráfica de gestión de todas las interfaces en IPv6

Para el gestor gráfico emplearemos MRTG , para lo cual configuramos el archivo : lanzamos el archivo de configuración de esta forma :

```
/usr/bin/cfgmaker --global 'WorkDir: /home/nitcom/linux/mrtg/nitcom' --global
'Language: spanish'
 --global 'Options[_]:bits,growright' --ifdesc=descr nuke@192.168.1.1 --output
/etc/mrtg/nitcom.cfg --EnablaIPv6=yes
```

Lo cual se obtiene en

[root@lab data]# nano /etc/mrtg/ipv6.cfg

Para una interface :

```
EnableIPv6: yes
WorkDir: /home/nitcom/linux/mrtg/ipv6
Options[_]: bits,growright
####Para SIT1
Target[sit1]: 4:nuke@[2a01:48:100:1:1::32]:6161:
SetEnv[sit1]: MRTG_INT_IP="" MRTG_INT_DESCR="sit1"
MaxBytes[sit1]: 125000
Title[sit1]: Analisis del trafico IPv6
PageTop[sit1]: <H1>GATEWAY IPV6</H1>
### Para Sit2000
Target[sit2000]: 15:nuke@[2a01:48:100:1:1::32]:6161:
SetEnv[sit2000]: MRTG_INT_IP="" MRTG_INT_DESCR="sit2000"
MaxBytes[sit2000]: 125000
Title[sit2000]: TELEFONICA
PageTop[sit2000]: <H1>TELEFONICA</H1>
```

Finalmente hacemos que se ejecute un crond cada 5 minutos :

[root@lab data]# crontab -l

*/5 * * * * env LANG=C mrtg /etc/mrtg/ipv6.cfg

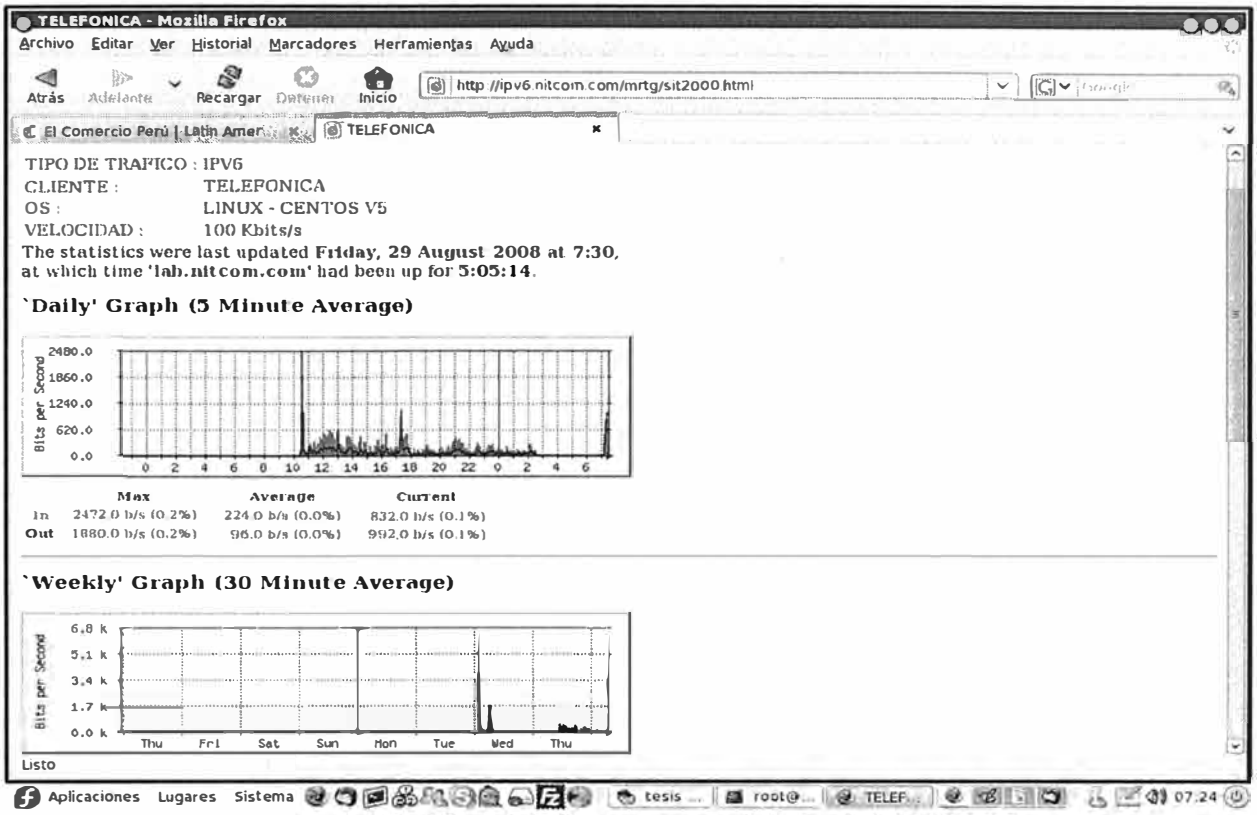


Figura 4.4 Gráfica del día y semana del túnel del HOST2

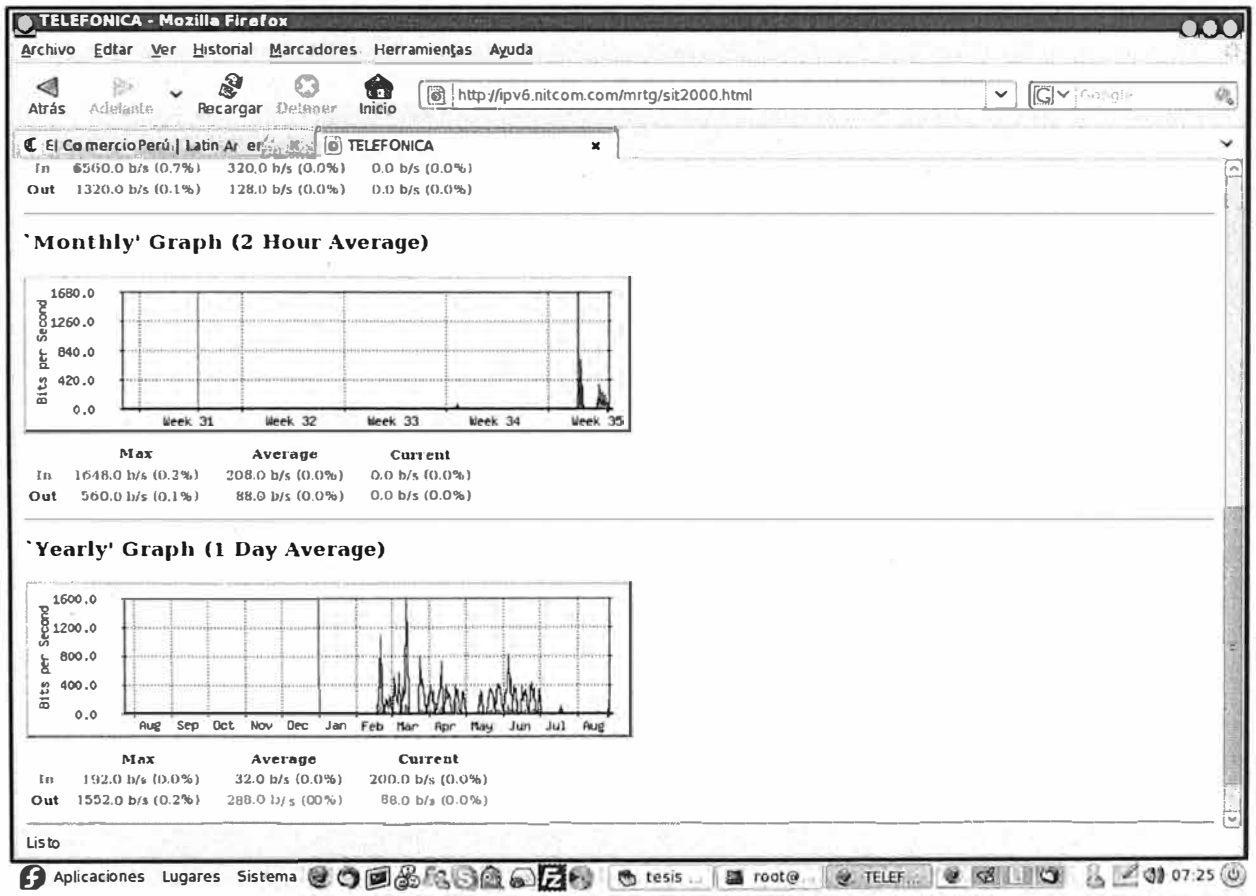


Figura 4.5 Gráfica del mes y año del túnel del HOST2

4.3 Servicios a migrar

En esta sección se establecerá los servicios a migrar de IPv4 y que estén soportados en IPv6, para lo cual solo efectuaremos las configuraciones mínimas y necesarias para demostrar su funcionamiento

4.3.1 SSH :

Para este caso solo es necesario instalar el openssh -server version 4.0 o superior en nuestro caso ya tenemos esas versiones tanto en el lado del Router LAB como en los clientes :

```
[root@lab data]# rpm -qa openssh*
openssh-clients-4.3p2-26.el5
openssh-askpass-4.3p2-26.el5
openssh-4.3p2-26.el5
openssh-server-4.3p2-26.el5
```

Efectuamos una prueba de conectividad desde nuestro gateway hacia el HOST2 usando IPv6.

```
[root@lab ~]# ssh 2a01:48:202:2000::2
root@2a01:48:202:2000::2's password:
Last login: Wed Aug 27 05:22:23 2008 from 2a01:48:202:2000::1
```

Con esta simple prueba probamos que el SSH trabaja y el stack dual también.

4.3.2 HTTP

Ahora migraremos nuestro server apache que es el contamos por excelencia en nuestras distribuciones Linux , para lo cual solo descargamos la ultima versión de la seria 2.x que es la que soporta y ya esta preparado para Linux.

Ahora solo nos queda dos pasos simples para que nuestra pagina se vea en la nube IPv6

Ahora finalmente la prueba de rigor a fin de probar el trabajo de nuestro servidor web en la nube IPv6

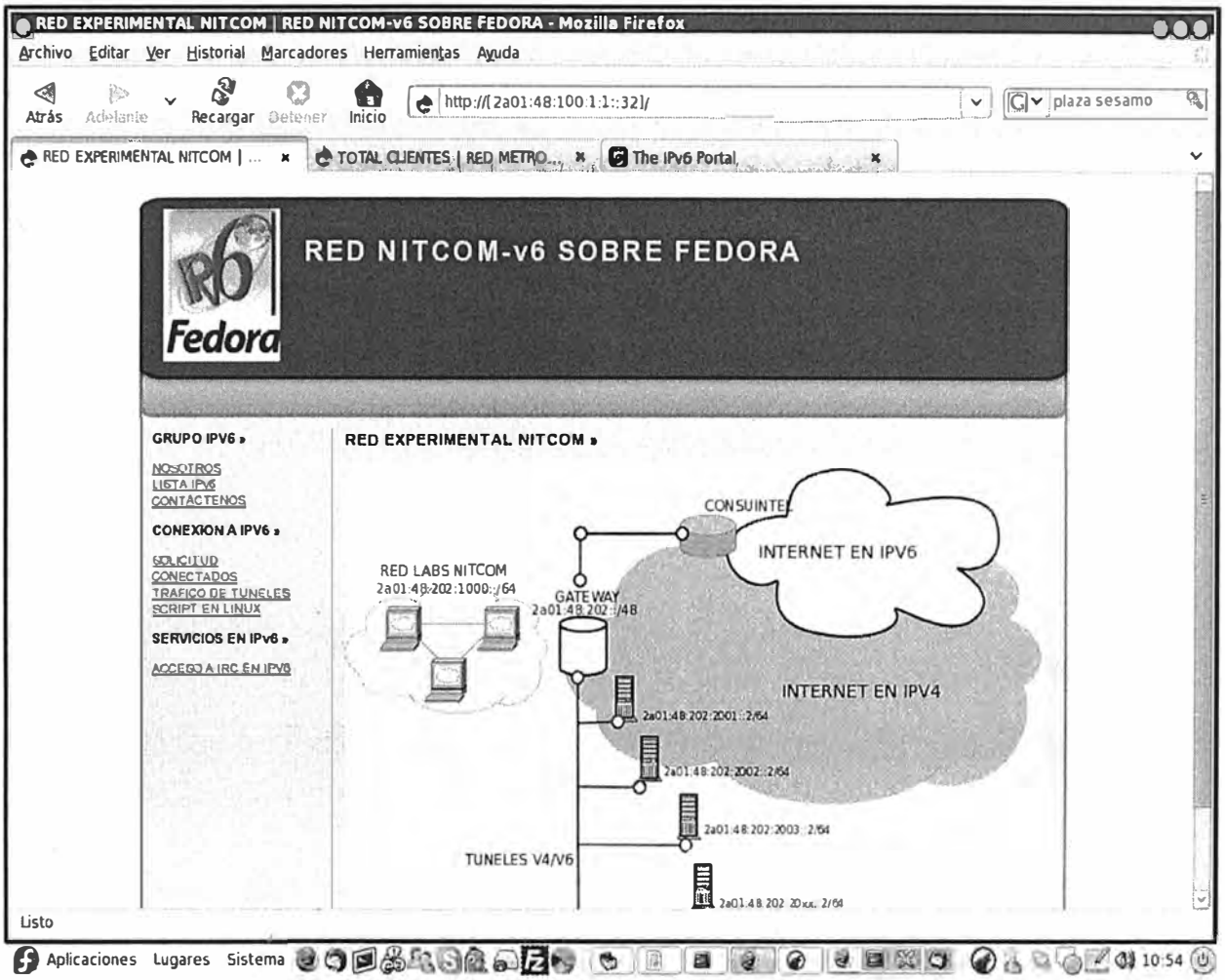


Figura 4.6 Servidor web de la maqueta migrado a IPv6

4.3.3 FTP

Para esta prueba emplearemos el vsftpd , lo descargamos desde la red :

[root@lab ~]# yum -y install vsftpd

Una vez instalado ahora solo procedemos a configurlarlo , para ello modificamos dos lineas , comentamos una linea y otra aperturamos :

[root@lab ~]# nano /etc/vsftpd/vsftpd.conf

```
#listen=YES
listen_ipv6=YES
```

Editamos el fichero :

[root@lab ~]# nano /etc/httpd/conf/httpd.conf

Estas lineas deberian estar presentes :

```
#
Listen 192.168.1.2:80
Listen [2a01:48:100:1:1::32]:80
#
NameVirtualHost [2a01:48:100:1:1::32]:80
<VirtualHost [2a01:48:100:1:1::32]:80>
DocumentRoot /home/nitcom/ipv6
ServerName lab.nitcom.com
ServerAlias ipv6.nitcom.com
</VirtualHost>
<VirtualHost [2a01:48:100:1:1::32]:80>
DocumentRoot /home/nitcom/ipv6
ServerName lab.nitcom.com
ServerAlias  2a01:48:100:1:1::32
</VirtualHost>
```

Finalmente reiniciamos el servicio para actualizar los cambios :

```
[root@lab ~]# /etc/init.d/httpd restart

Parando httpd:                                [  OK  ]
```

Reiniciamos el servicio :

```
[root@lab ~]# /etc/init.d/vsftpd restart

Apagando vsftpd:                                    [FALLÓ]
```

Desde la maquina SHARP hacemos la prueba de conectividad en IPv4 y en Ipv6

```
[gino@sharp ~]$ ftp lab.nitcom.com
Connected to lab.nitcom.com (192.168.1.2).
220 (vsFTPd 2.0.5)
Name (lab.nitcom.com:gino): ^C[gino@sharp ~]$
[gino@sharp ~]$ ftp lab.nitcom.com
Connected to lab.nitcom.com (192.168.1.2).
220 (vsFTPd 2.0.5)
Name (lab.nitcom.com:gino): galania
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bye
221 Goodbye.
[gino@sharp ~]$ ftp 2a01:48:100:1:1::32
Connected to 2a01:48:100:1:1::32 (2a01:48:100:1:1::32).
220 (vsFTPd 2.0.5)
Name (2a01:48:100:1:1::32:gino): galania
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

4.3.4 DNS

Para el caso del DNS tendría que explicar todo el concepto del protocolo el cual no es el sentido de este trabajo, solo mencionare los pasos previos para tener un sencillo dns en v6 sobre mi ya existente servidor dns que trabaja en IPv4.

Para lo cual emplearemos el bind que es el programa que se emplea por excelencia en los mas grandes servidores DNS del mundo, inicialmente sobre sistemas UNIX se implemento para Linux , heredo todas sus cualidades y finalmente se mejoro en una sistema "enjaulado" como lo es el bind-chroot

Para lo cual procedemos a instalarlo :

[root@lab ~]# yum -y install bind-chroot

En bind existe algunos archivos de configuracion como el named.conf en donde defines las zonas segun el dominio a emplear, en este caso definimos la zona nitcom.com que es justamente el dominio.

Nosotros agregamos solo para el tema IPv6 sobre nuestra archivo de zona :

[root@lab ~]# nano /var/named/chroot/var/named/nitcom.com

```
.....
    IN      NS      ns,nitcom.com.
    IN      NS      lab.nitcom.com.
    IN      NS      wrc.nitcom.com.
    IN      AAAA    2a01:48:100:1:1::32
            MX      10 mail1.nitcom.com.
            MX      20 mail.nitcom.com.
            MX      30 mail2.nitcom.com.
            TXT     "Servidor NITCOM Labs"

nitcom.com.     A       200.60.195.7
ipv6            AAAA    2a01:48:100:1:1::32
lab             A       200.60.195.7
ns              A       200.60.195.7
```

Podemos notar el incremento y lo resaltamos para que se note lo que ha agregado a fin de que el dominio ya responda.
Ahora viene dos pruebas de ley , desde el mismo gateway y luego desde el host remoto.

Desde el gateway

```
[root@lab ~]# ping6 ipv6.nitcom.com
PING ipv6.nitcom.com(2a01:48:100:1:1::32) 56 data bytes
64 bytes from 2a01:48:100:1:1::32: icmp_seq=0 ttl=64 time=0.086 ms
64 bytes from 2a01:48:100:1:1::32: icmp_seq=1 ttl=64 time=0.089 ms
64 bytes from 2a01:48:100:1:1::32: icmp_seq=2 ttl=64 time=0.082 ms
64 bytes from 2a01:48:100:1:1::32: icmp_seq=3 ttl=64 time=0.080 ms
```

Desde el host remoto

```
[root@telefonica ~]# ping6 ipv6.nitcom.com
PING ipv6.nitcom.com(2a01:48:100:1:1::32) 56 data bytes
64 bytes from 2a01:48:100:1:1::32: icmp_seq=0 ttl=64 time=25.4 ms
64 bytes from 2a01:48:100:1:1::32: icmp_seq=1 ttl=64 time=18.9 ms
64 bytes from 2a01:48:100:1:1::32: icmp_seq=2 ttl=64 time=56.6 m
```

Finalmente hacemos la consulta al DNS de nuestro proveedor comprobando que la asignación IPv6 ya trabaja.

```
[root@lab ~]# host -t AAAA ipv6.google.com
ipv6.google.com is an alias for ipv6.l.google.com.
ipv6.l.google.com has IPv6 address 2001:4860:0:1001::68
[root@lab ~]# host -t AAAA ipv6.nitcom.com
ipv6.nitcom.com has IPv6 address 2a01:48:100:1:1::32
```

CONCLUSIONES

1. La primera conclusión que se obtiene del presente trabajo es que IPv6 es un protocolo necesario para el crecimiento de la red en el futuro.
2. IPv6 no constituye una revolución mas bien una evolución del protocolo IPv4 , es decir a partir del protocolo IPv4 y de sus falencias es donde se construye el protocolo IPv6 .
3. IPv6 aparte de su escalabilidad como protocolo nos permite romper muchos paradigmas que lo teníamos enmarcado en IPv4, como por ejemplo el concepto de la puerta de enlace, usualmente se pensaba que siempre era fija y determinada por la red Local , con IPv6 simplemente la puerta de enlace se traduce a un punto en la nube que no necesariamente es la red Local , este concepto es la Movilidad en el cual IPv6 lo soporta de muy buena forma.
4. El decremento de cabeceras y la inclusión de otras hace que nuestro protocolo fácilmente trabaje con QoS sin necesidad de construir arquitecturas de red complejas .
5. La seguridad es un objetivo claro del protocolo el cual IPv4 lo descuida de sobremanera , no por que sea deficiente sino que cuando se concibió el protocolo jamás se pensó en la vulnerabilidad del mismo y el ataque a servicios que jamás se pensaron y que están soportados por el protocolo , como por ejemplo el envenenamiento de los DNS que es la base de la Internet actual y que aun no se tenga solución frente a una vulnerabilidad muy peligrosa.
6. En la implementación rescatamos una buena experiencia sobre el software libre y notamos que este fácilmente soporta la migración a IPv6 a través de los servicios conocidos de la capa de aplicación .
7. El dual stack es el método mas practico para empezar con una migración del protocolo , el cual se demostró con la implementación haciendo convivir en un mismo hardware los dos protocolos soportando ambos servicios.
8. El manejo de saltos y colas lo maneja muy eficiente el protocolo y eso lo demuestra las pruebas realizadas en la implementación.

9. Actualmente casi todos los servicios es posible migrar con suma facilidad asi como lo demuestra la implementación en producción que la tenemos , lo cual simboliza que ya estamos preparados para el paso #1 que es la etapa de implementación haciendo convivir los dos stack hasta el día cero.
10. Finalmente IPv6 nos permite la apertura en mundo nuevo manteniendo una jerarquía en la asignación de IPv6 y en los saltos respectivos para ubicar un host y nos los saltos que denotamos actualmente , en nuestra implementación los saltos lo notamos mas limpio claro y ordenado.
11. IPv6 nos permite proyectos nuevos y retos sumamente desafiantes debido a la movilidad a la escalabilidad y lo que es mejor a que nos nos limita el protocolo a crecer con las aplicaciones

BIBLIOGRAFÍA

[1] Redes de computadoras
Andrew Tanembaum
Publisher: Prentice Hall (December 2003)

[2] Ettikan Kandasamy Karuppiah, Gopi Kurup, Takefumi Yamazaki,"Application Performance Analysis In Transition Mechanism From Ipv4 to IPv6", Proceedings APAN Conf. 2000, Tsukuba, Japan, Feb. 14-18, 2000.
http://www.my.apan.net/ipv6/Papers/ettikan.PDF

[3] Ettikan Kandasamy, Tong Hui Tee, Seow Chen Yong, "Transition Mechanism Between IPv4 & IPv6 and deciding your choice", APAN Conf. 2002, Jan. 22-25, Paper,
http://www.my.apan.net/ipv6/Papers/Transition-Mechanism-IPv4_IPv6.pdf

[4] Ettikan Kandasamy ,"IPv6 Dual Stack Transition Technique Performance Analysis : KAME on FreeBSD as the Case", Proceedings MMU International Symposium on Information and Communication Technologies, Malaysia, 5th - 6th Oct.,

2000. http://www.my.apan.net/ipv6/Papers/M2USIC_Perf.PDF

[5] Reporte de espacio de direcciones IPv4,

http://bgp.potaroo.net/ipv4/

[6] F. Solensky, "IPv4 Address Lifetime Expectations," in IPng:

Internet Protocol Next Generation (S. Bradner, A. Mankin,

ed), Adddison Wesley, 1996

[7] "Global IPv6 allocations made by the Regional Internet

Registries", http://www.ripe.net/cgi-bin/ipv6allocs

[8] Wikipedia http://es.wikipedia.com