
# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE INGENIERÍA

# SERVIDORES DE NOMBRES DE DOMINIO CON IPV6

**TESIS PROFESIONAL PARA OBTENER EL TÍTULO DE INGENIERO EN COMPUTACIÓN**

**ÁREA**
**REDES Y SEGURIDAD**

**PRESENTA:**
**GONZÁLEZ ESCAMILLA JOSÉ ALBERTO**

**DIRECTOR DE TESIS**
**ING. ALEJANDRO CRUZ SANTOS**


**CIUDAD UNIVERSITARIA, 2014**

***AGRADECIMIENTOS***

*Agradezco a mis padres con todo mi cariño y mi amor ya que ellos hicieron todo en la vida para que yo pudiera lograr mis sueños, por motivarme y apoyarme, a ustedes por siempre mi corazón y mi agradecimiento.*

*A mi hijo por soportarme durante todo este tiempo y tenerme mucha más consideración de la que merezco.*

*A mi esposa por tenerme paciencia y comprensión, por darme tu tiempo para que yo pudiera cumplir con el mío. Por tu bondad y sacrificio me inspiraste a ser mejor persona.*

*A mis maestros que influyeron con sus lecciones y experiencias en formarme como una persona preparada.*

*A mi director de tesis que me brindó su apoyo y constancia para así poder terminar mi tesis.*

# ÍNDICE

**Página**


**Objetivo**..... 1

**Alcances**..... 1

**Introducción**..... 3

**Capítulo 1. Marco teórico**..... 6

1.1 Historia IPv4..... 7

1.1.1 Formato de IPv4..... 8

1.1.2 Tipos de direcciones IPv4..... 9

1.1.3 Clases de direcciones IPV4..... 10

1.2 Historia IPv6..... 11

1.2.1 Principales técnicas de transición..... 15

1.3 Historia del DNS..... 16

1.4 Transición IPv4 a IPv6 en la UNAM..... 20

**Capítulo 2. La necesidad de IPv6**..... 24

2.1 Problemas existentes en IPv4..... 25

2.1.1 Agotamiento de direcciones IPv4..... 26

2.1.2 Problemas de Arquitectura..... 28

2.2 Motivadores del cambio a IPv6..... 29

2.3 Comparaciones entre IPv4 e IPv6..... 31

2.4 Ventajas de IPv6..... 35

**Capítulo 3. El protocolo IPv6**..... 37

3.1 Características del Protocolo IPv6..... 38

3.2 Estructura de un paquete IPv6..... 40

3.3 Formato de una dirección IPv6..... 42

3.3.1 Identificación de los tipos de direcciones..... 44

3.4 Direccionamiento IPv6..... 49

3.4.1 *Unicast*..... 50

3.4.2 *Multicast*..... 51

3.4.3 *Anycast*……. 52
3.5 Mecanismos de configuración de direcciones IPv6…… 53

**Capítulo 4. Servidor de Nombres de Dominio (DNS/BIND) con IPv6**…….. 54
4.1 BIND…….. 55
4.1.1 Acerca de BIND…… 55
4.1.2 Tipos de servidores de nombres de dominio……. 56
4.1.3 Mejoras al protocolo DNS……. 57
4.1.4 Seguridad……... 58
4.1.5 IPv6…… 59
4.2 *Anycast* en IPv4 e IPv6…….. 59
4.3 Instalación y configuración de BIND y Quagga……... 62
4.3.1 Instalación de OpenBSD 5.1……. 63
4.3.2 Configuración de BIND…… 73
4.3.2.1 Configuración del Servidor……... 74
4.3.2.2 Declaración y Configuración de zonas……. 84
4.3.2.2.1 Zona Directa…… 87
4.3.2.2.2 Zona Inversa…… 90
4.3.3 Paquete Quagga……. 91
4.3.3.1 Instalación de Quagga……... 92
4.3.3.2 Configuración de Quagga…….. 94
4.4 Implementación del DNS con IPv6 en Nodos de RedUNAM…… 104
4.5 Servicios y Aplicaciones en IPv6 en la UNAM……. 113

**Conclusiones**……... 115
**Glosario de términos**…….. 117
**Lista de RFCs**……. 127
**Índice de tablas**……... 129
**Índice de figuras**……. 131
**Bibliografía**……. 134

# Objetivo y alcances

# OBJETIVO

El objetivo de este trabajo es implementar el protocolo IPv6 en un servidor DNS conectado a RedUNAM y lograr que conviva con los actuales servidores DNS de RedUNAM para brindar el servicio de resolución de nombres de dominio en IPv6 a la comunidad universitaria de la UNAM.

# ALCANCES

El alcance de la tesis es delimitado mediante objetivos específicos:

- Identificar las ventajas sobre la integración de IPv6 para entender su impacto sobre RedUNAM.
- Definir un ambiente controlado y llevar a cabo pruebas para la evaluación de servidores DNS en IPv6.
- Poner en producción la resolución de nombres de dominio en IPv6 en RedUNAM.

# Introducción

# Introducción

En las últimas décadas el incremento exponencial de Internet ha dado paso al desarrollo de nuevas tecnologías y protocolos, donde el protocolo IPv4 ha sido el principal protagonista del desarrollo de Internet. Esta expansión tan grande y acelerada en un lapso de tiempo relativamente corto ha puesto en alerta a la comunidad de Internet y por esta razón hace algunos años se comenzaron a crear grupos de trabajo para el desarrollo de nuevos protocolos que permitieran superar algunas de las limitaciones del protocolo de IPv4, como lo es el espacio de direcciones.

Ya que el espacio de direcciones de IPv4 se agotó es necesario adoptar el protocolo IPv6 el cual crea un espacio de direcciones ampliamente mucho mayor que el protocolo IPv4 haciendo con esto una mejor posibilidad para desarrollar la infraestructura actual así como las nuevas tecnologías logrando la base para el desarrollo de Internet durante las próximas décadas.

En la actualidad el soporte de IPv6 que ofrecen los fabricantes de equipos y aplicaciones ha alcanzado un desarrollo que permite la implementación de redes en IPv6 de forma nativa y ya no sería necesario depender de herramientas de traducción y/o túneles para poder desarrollar redes en IPv6 que implementen el mismo tipo de servicios otorgados en redes IPv4.

De manera que para alcanzar el objetivo planteado es que en el primer capítulo se muestran los antecedentes acerca del protocolo IPv4, IPv6, del servicio de DNS y de la transición de IPv4 a IPv6 en RedUNAM. Para que se tenga un panorama de lo importante que fue el desarrollo de estos protocolos y del servicio de resolución de nombres de dominio, además del impacto que causan en nuestra máxima casa de estudios.

En el segundo capítulo se plantean los problemas del protocolo IPv4 así como los motivadores para la transición a IPv6 que entre las comparaciones de estos dos protocolos se puede determinar que es necesaria la transición y en algunos casos el que los protocolos convivan para aprovechar la infraestructura actual de las dependencias conectadas a RedUNAM.

Es necesario conocer el protocolo de IPv6 en el capítulo tres se desarrollan las características de este protocolo así como su estructura, formato, tipo de direccionamiento y los mecanismos de configuración de direcciones.

En el capítulo cuatro explico lo que es BIND, el esquema de anycast en IPv4 e IPv6, el sistema operativo en el cual se desarrollan las pruebas de resolución de nombres de dominio así como el software Quagga herramienta necesaria para el desarrollo de anycast.

# Capítulo 1

# Marco teórico

## 1.1 Historia de IPv4

En un inicio en Internet, la comunicación se hacía mediante las direcciones IP (Protocolo de Internet, Internet Protocol) el cual es un tipo de mecanismo que nos permite consultar información o enviar información desde Internet.

Los protocolos TCP (Protocolo de Control de Transmisión, Transmission Control Protocol) e IP, surgidos hace más de 30 años, son dos de los más importantes impulsores del nacimiento de Internet.

El protocolo IP pertenece a la capa tres del modelo OSI (Sistemas De Interconexión Abiertos, *Open System Interconnection*) que ofrece direccionamiento, enrutado de datagramas, etcétera, por otro lado está el protocolo TCP que es la capa principal de transporte, capa 4 del modelo OSI, y se encarga del establecimiento de conexiones y del transporte de datos.

En los primeros trabajos de la DARPA (Agencia de Proyectos de Investigación Avanzados de Defensa, *Defense Advanced Research Projects Agency*) se incluían una versión de protocolo TCP, y de hecho estas siglas no significaban lo que significan hoy, sino "*Transmission Control Program*". La primera versión de TCP apareció en 1973 la cual fue revisada y documentada en el RFC (Petición De Comentarios, *Request for Comments*) 675, "*Specification of Internet Transmission Control Program*", en diciembre de 1974.

El verdadero surgimiento de TCP/IP se produjo hasta que Jon Postel, uno de los más importantes pioneros de Internet y TCP/IP, postuló que TCP hacía demasiado.

El primer TCP englobaba funciones de las capas 3 y 4 del modelo OSI, y dadas estas observaciones que Postel hizo notar, se culminó con la separación de los protocolos TCP e IP. Uno de los primeros pasos para la separación de TCP e IP se dio en 1978 con la versión 3 y no fue hasta 1980 cuando se publicó la versión 4 (IPv4) que seguimos usando hoy día. (Pérez, 2007)

Cuando surgieron los primeros problemas de TCP/IP, principalmente por el sistema de numeración de direcciones y el evidente agotamiento de direcciones disponibles dado a la gran cantidad de equipos conectados y aunque el protocolo TCP/IP se ha demostrado muy fiable, dados estos problemas se comenzaron a desarrollar nuevas mejoras al protocolo IP llegando a la versión 6 (IPv6) en la que se sigue trabajando para poder integrarla a las redes actuales.

### 1.1.1 Formato de IPv4

Las direcciones IPv4 son direcciones de 32 *bits*, que están representadas en cuatro octetos de la siguiente manera: X.X.X.X

Donde cada X es un número entre 0 y 255, que son todos los números enteros que se pueden representar con 8 *bits*.

Ejemplos: 192.168.1.16, 132.248.115.82, 201.64.26.125.

## 1.1.2 Tipos de direcciones IPv4

En el protocolo IPv4 existen tres tipos de direcciones:

**a)** Públicas- Son aquellas direcciones que son enrutables hacia Internet y con las cuales podemos tener acceso a Internet.

Ejemplos: 201.127.223.2, 145.66.12.122, 23.6.45.55.

**b)** Privadas- Son aquellas direcciones que no se pueden usar para enrutar hacia Internet, son útiles para ser usadas en redes locales, entornos domésticos o corporativos. Descritas en el RFC 1918[1].

Los Siguientes rangos están reservados para uso privado:

- De 10.0.0.0 a 10.255.255.255
- De 172.16.0.0 a 172.16.255.255
- De 192.168.0.0 a 192.168.255.255

**c)** Reservadas- Son aquellas direcciones que no deben usarse salvo para lo que fueron reservadas. Las más importantes son las siguientes:

- 0.0.0.0 (o la dirección .0 de cualquier subred) Esta es dirección se usa para referirse a la red.
- 255.255.255.255 (o la dirección .255 de cualquier subred) Esta es la dirección de *Broadcast*. Equivale a todos los equipos de la red.

[1] Véase lista de RFCs

- 127.X.X.X Este es el rango de direcciones IP de *loopback*. Esta dirección se suele utilizar cuando una transmisión de datos tiene como destino el propio *host*. También llamadas de diagnóstico.
- 127.0.0.1 (o *local host*) Es un caso particular del anterior. Es la más usada para referirnos a nuestra máquina de manera local.

## 1.1.3 Clases de direcciones IPv4

Anteriormente las direcciones eran consideradas usando clases, es decir, que se tomaba la máscara implícita dependiendo de la clase a la que pertenece la dirección. Estas clases son las siguientes (ver tabla 1.1):

- Clase A: De la dirección 1.0.0.0 a la 126.255.255.255

  Máscara de red- 255.0.0.0, *Broadcast*- X.255.255.255

- Clase B: De la dirección 128.0.0.0 a la 191.255.255.255

  Máscara de red- 255.255.0.0, *Broadcast*- X.X.255.255

- Clase C: De la dirección 192.0.0.0 a la 223.255.255.255

  Máscara de red - 255.255.255.0, *Broadcast*- X.X.X.255

- Clase D: De la dirección 224.0.0.0 a la 239.255.255.255 (Direcciones *Multicast)*

- Clase E: De la dirección 240.0.0.0 a la 255.255.255.255

  (Direcciones de Investigación)

*Tabla 1.1 Clases de direcciones IPv4*

| Clase A | Red | *Host* | | |
|---|---|---|---|---|
| Octeto | 1 | 2 | 3 | 4 |
| Clase B | Red | | *Host* | |
| Octeto | 1 | 2 | 3 | 4 |
| Clase C | Red | | | *Host* |
| Octeto | 1 | 2 | 3 | 4 |
| Clase D | *Host* | | | |
| Octeto | 1 | 2 | 3 | 4 |

Cuando surgieron problemas con el tamaño de las redes por clases, que genera una máscara de red fija y cantidad de *host*s iguales a todas las subredes, esto no eran una manera viable para la gran demanda de direcciones IP y en consecuencia se optó por VLSM (Máscaras de Subred de Tamaño Variable, *Variable Length Subnet Mask*) que permiten un mayor aprovechamiento de las direcciones, el proceso de VLSM toma una dirección de red o subred y la divide en subredes más pequeñas adaptando las máscaras según las necesidades de cada subred, generando una máscara diferente para las distintas subredes de una red, de este modo se aprovecha mejor el direccionamiento y la máscara de red no sería fija según la clase.

## 1.2 Historia de IPv6

El crecimiento exponencial de las redes, de Internet y que cada vez más dispositivos requieren de una dirección IP para utilizar servicios de Internet nos ha llevado hacia el agotamiento de las direcciones IPv4 (3 de febrero de 2011). Este tema ha sido una de las principales preocupaciones desde los años 80. Como consecuencia, es un factor

determinante en la creación y adopción de nuevas tecnologías, como IPv6. (The Number Resource Organization, 2011)

La IETF (Grupo de Trabajo de Ingeniería de Internet, *Internet Engineering Task Force*) ha producido un conjunto comprensible de especificaciones (RFC 1752, 1883, 1886, 1971, 1993, etcétera) que definen la siguiente generación del protocolo de internet conocido como "IPng" o "IPv6".

El protocolo de IPv6 es la versión más reciente del Protocolo de Internet que fue diseñada para mejorar y solucionar algunos de los problemas del protocolo IPv4 y representa el fruto de muchas propuestas de la IETF y de grupos de trabajo centrados en desarrollar un IPng (*Internet Protocol for Next Generation*[2]) (ver figura 1.1). (RFC1752, 1995)

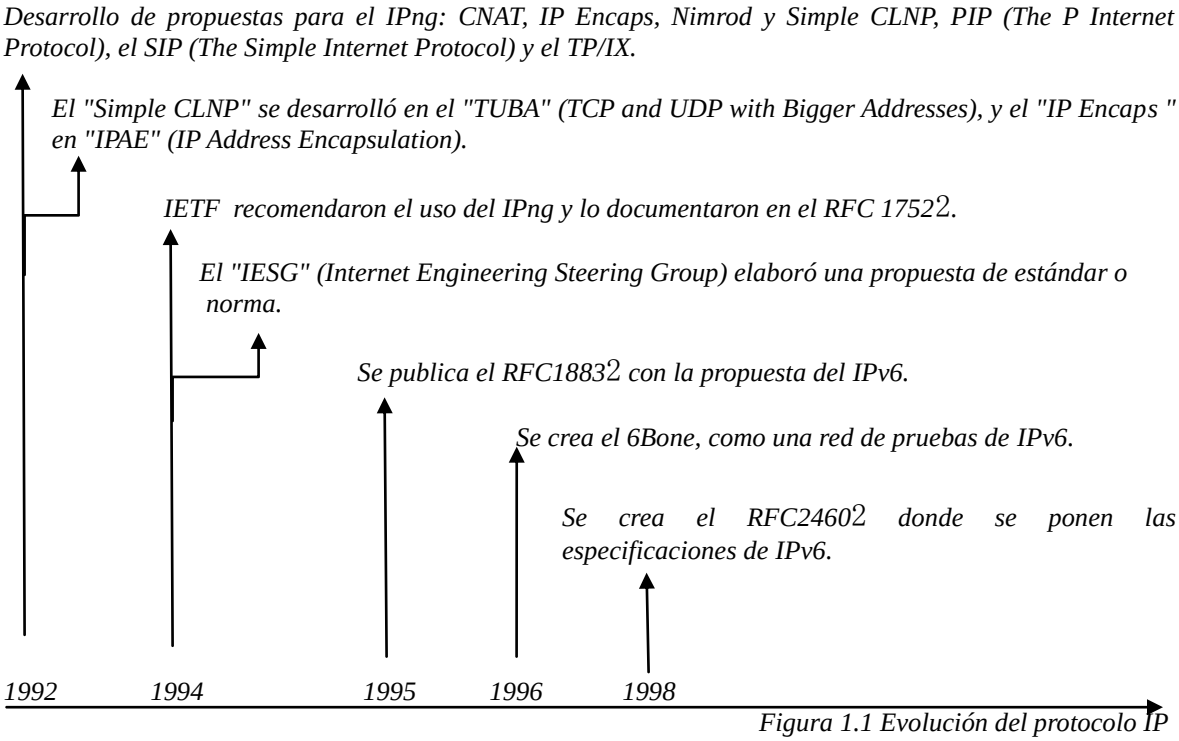


*Figura 1.1 Evolución del protocolo IP*

[2] *Véase lista de RFCs*

En una ceremonia realizada el 3 de febrero de 2011 en Florida (EE.UU), y organizada por la NRO (Asociación de recursos numéricos, *Number Resource Organization*) que es formada como una entidad para representar los intereses, llevar a cabo actividades conjuntas y coordinar globalmente las actividades de los cinco RIR (Registros Regionales de Internet, *Regional Internet Registry*) (ver figura 1.2) que son organizaciones que supervisan la asignación y el registro de recursos de números de Internet dentro de una región particular del mundo.

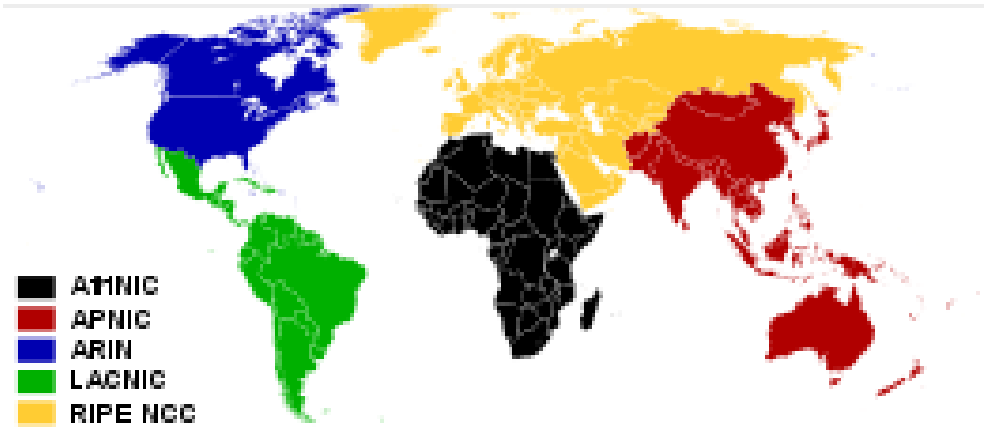


*Figura 1.2 Registros Regionales de Internet*[3]

Actualmente hay 5 RIRs en funcionamiento:

- ARIN (*American Registry for Internet Numbers*) para América Anglosajona.
- RIPE NCC (*RIPE Network Coordination Centre*) para Europa, el Oriente Medio y Asia Central.
- APNIC (*Asia-Pacific Network Information Centre*) para Asia y la Región Pacífica.
- LACNIC (*Latin American and Caribbean Internet Address Registry*) para América Latina y el Caribe.
- AfriNIC (*African Network Information Centre*) para África

[3] *Imagen sacada de http://es.wikipedia.org/wiki/Registro_Regional_de_Internet*

Cada RIR recibió certificados que simbolizaban los últimos cinco bloques /8 de direcciones IPv4, que equivalen a la 256va parte del espacio total de direcciones IPv4 que recibirán de parte de la IANA (Autoridad de Asignación de Números en Internet, *Internet Assigned Numbers Authority*). Actualmente es un departamento operado por ICANN (Corporación de Internet para la Asignación de Nombres y Números, *Internet Corporation for Assigned Names and Numbers*).

El mismo 3 de febrero de 2011 Leo Vegoda, gerente de Recursos Numéricos de la IANA, anunciaba en las listas técnicas de correo electrónico la extinción de los bloques libres *unicast* IPv4.

Ya que se terminaron las direcciones IPv4 del stock central de IANA el futuro de Internet y de las redes de datos está en el protocolo IPv6. Ahora todos los que toman las decisiones correspondientes a la implementación de nuevas tecnologías, deberán realizar las acciones correspondientes para adoptar el protocolo IPv6 en sus organizaciones. (The Number Resource Organization, 2011)

En la versión 6 del protocolo IP se introducen modificaciones fundamentales. No sólo el tamaño de la dirección IP ha sido aumentado a 128 *bits*, sino también ha sido modificado el formato de la cabecera IP y el modo en que se procesa la información que contiene. La transición de IPv4 a IPv6 no es sencilla ya que depende si se quiera que convivan los protocolos o pasar totalmente a IPv6 y los mecanismos que permitan la coexistencia y la transición entre las dos versiones han de estar estandarizadas.

## 1.2.1 Principales técnicas de transición

El grupo de trabajo "NGTrans" creado por la IETF ha definido tres principales técnicas de transición:

**a) Doble Pila**

Esta técnica de transición es de las más sencillas de implementar ya que requiere que los *hosts* y los enrutadores soporten ambas versiones (4 y 6) de IP y, por lo tanto, servicios y aplicaciones tanto en IPv4 como en IPv6.

El enfoque de doble pila es un mecanismo fundamental para introducir el protocolo IPv6 en las arquitecturas con IPv4, pero su punto débil es que obliga a que cada máquina retenga una dirección IPv4, cada vez más escasas ya que el stock central de IANA se agoto, pero esto también depende del manejo de direcciones IPv4 de cada RIR ya que estos pueden tener direcciones IPv4 disponibles para distintos usos. Así, a medida que se difunde IPv6, la técnica de doble pila tendrá que ser aplicada donde ayude al proceso de transición, por ejemplo en *routers* y servidores.

**b) *Tunneling***

Esta técnica de transición permite interconectar las nubes de IPv6 a un servicio IPv4 nativo a través de un túnel. Los paquetes IPv6 son encapsulados por un *router* de extremo antes de ser transportado a través de la red IPv4, siendo desencapsulados en el extremo de la red IPv6 receptora. Los túneles pueden ser configurados estática o dinámicamente como "6to4" (Es un sistema que permite mandar paquetes IPv6 sobre redes IPv4 ignorando la necesidad de configurar túneles manualmente. Fue diseñado para permitir conectividad

IPv6 sin la cooperación de los proveedores de Internet) o “6over4” (Es un mecanismo de transición de IPv6 para transmitir paquetes IPv6 entre nodos con doble pila sobre una red IPv4 con *multicast* habilitado).

**c) Mecanismo de traducción o conversión de protocolos**

Esta técnica de transición es necesaria cuando un *host* IPv6 se comunica con un *host* IPv4. La cabecera IP es convertida y se requiere de un rango de direcciones IPv4 para proporcionar un alias al *host* IPv6 durante la comunicación. La conversión será más compleja si la aplicación procesa las direcciones IP; de hecho tal conversión hereda la mayoría de los problemas de IPv4 *Network Address Translators* (NAT). (NIC México)

## 1.3 Historia del DNS

.

En Internet todos los dispositivos que utilizan el protocolo IP tienen al menos una dirección IP, que debe ser única dentro de la red a la que pertenece, esto hace que la comunicación entre los equipos y los humanos sea más fácil ya que los equipos tienen asignado un nombre o identificador, de esta forma, es más fácil recordar el nombre de una máquina ya que podemos asociar este a la organización o lugar en el que se encuentra, sin tener que memorizar la dirección de IP del equipo.

A este concepto se le conoce como Sistema de Nombres de Dominio, (*DNS, Domain Name System*), el cual nació en la década de los 80's. Creado por Paul Mockapetris en colaboración con Jon Postel y Paul Vixie. Desarrollaron lo que hasta ahora conocemos como BIND (*Berkeley Internet Name Domain*), un sistema tipo cliente/servidor, jerárquico

y distribuido, cuyas características se describen en los RFC (1033, 1034 y 1035)[4] y que son muy parecidas a un sistema de archivos de UNIX, pero distribuido.

El uso del servidor DNS solamente involucró en un principio instituciones académicas, de investigación y la milicia de los Estados Unidos. En aquellos tiempos las universidades empezaban a realizar conexiones con otras redes, entre ellas BitNet. Como el uso de la red empezaba a crecer y era importante poner orden en cuanto a los equipos que ingresaban a la red. Entonces se crearon los nombres de dominio genéricos de primer nivel (*gTLD, generic Top-level Domain*), como el .com, .net y .org, que se habían creado estas tres clasificaciones con el fin de ubicar el tipo de entidades que buscaban tener presencia en Internet. Además de estos gTLD se comenzó por delegar los sufijos nacionales (*nTLD, national Top-level Domain*) a los países que se fueran conectando a la red. De esta forma, a México se le asignó el .mx a finales de 1988 cuando el ITESM (Instituto Tecnológico y de Estudios Superiores de Monterrey) Campus Monterrey se conecta de manera dedicada al Internet. Con el paso del tiempo cada país obtuvo su propio nTLD. Y también se crearon nombres de dominio especiales, sTLD (*sponsored Top-level Domains*) como .mil, .edu, .gob, etcétera, para identificar a algunas de las organizaciones por el ámbito en el que se desarrollan.

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

[4] *Véase lista de RFCs*

En 1992, la NSF (Fundación Nacional de Ciencias, *National Science Foundation*) quien administraba el *backbone* de Internet (en ese entonces NSFNET) decide licitar la operación del InterNIC y le otorgan la función a la NSI (*Network Solutions Inc.*), esta empresa sería adquirida por el grupo SAIC (*Science Application International Corporation*). Cuando la NSI obtuvo el contrato, se estableció un apoyo de cuatro millones de dólares por parte de la NSF a NSI, para realizar la función del registro de los gTLD.

En 1996, el director de la IANA en ese entonces Jon Postel, realizó una propuesta que contemplaba la creación nuevos nombres de dominios genéricos (.com, .net y .org). Esta propuesta tuvo efectos importantes y finalizó en la formación de un grupo que se encargaría de discutir el re-diseño de los gTLD. De esta forma nació el IAHC (*Internet-International Ad Hoc Committee*) impulsado por la ISOC (*Internet Society*), con lo cual se generó el reporte final, donde se manejaban las recomendaciones y requerimientos para nuevos esquemas de gTLD, este documento recibiría el nombre de Memorando de Entendimiento para los Nombres de Dominio genéricos de Nivel Superior.

El IAHC se disolvió para dar paso al gTLD-MoU (*generic Top level Domain Memorandum of Understanding*), creando un documento que fue respaldado por organizaciones de todo el mundo, entre ellas la WIPO (Organización Mundial de la Propiedad Industrial), ITU (Unión Internacional de Telecomunicaciones), ISOC, MCI y por Latinoamérica sólo NIC-México.

El gTLD-MoU contempló nuevos gTLD (.firm, .shop, .web, .arts, .rec, .info, .nom), una administración múltiple y distribuida de los gTLD, en la cual se tuviera la opción de que más de una organización pudiera registrar nombres de dominio bajo .com, la creación de

un consejo central (*CORE, Council of Registrars*) formada por las organizaciones que tendrían el funcionamiento como nuevos InterNICs, y dos cuerpos más de apoyo al nuevo esquema, PAB (*Policy Advisory Group*) y el POC (*Policy Oversigt Committee*).

El CORE estableció un contrato con *Emergent Corp* para el desarrollo del nuevo esquema distribuido de DNS (*new DNS Shared Registry System*). Todo estaba listo para que empezaran a operar los 89 registros en todo el mundo (ver Figura 1.3), aceptando las solicitudes de dominio bajo los siete nuevos nombres de dominio genéricos. (Robles, 1998)

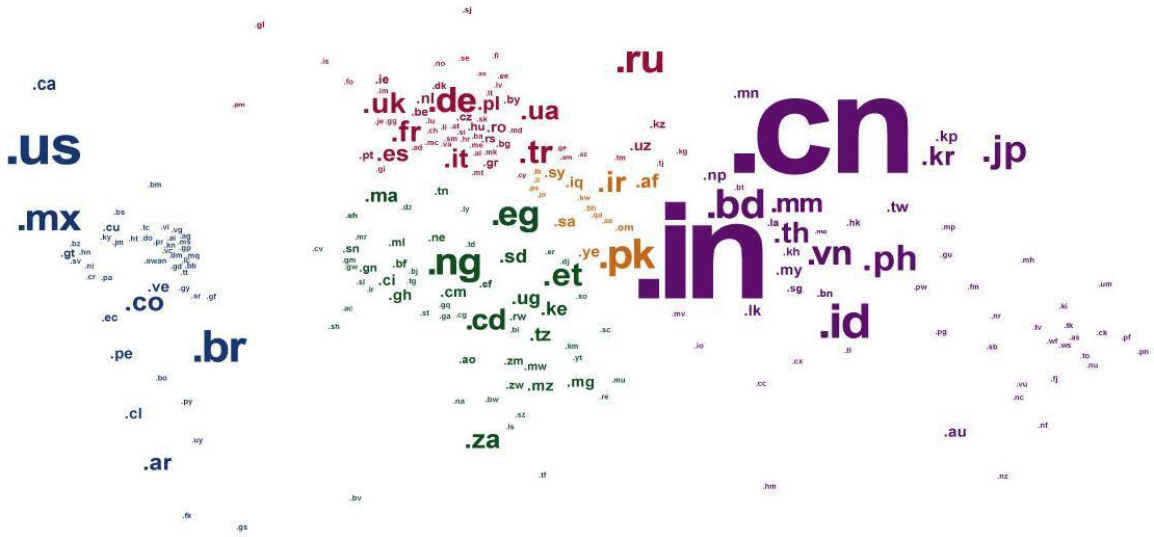


*Figura 1.3 Mapa de ccTLD*[5]

[5] *Imagen sacada de http://www.geographos.com/BLOGRAPHOS/?p=293*

El resultado de estas reuniones de trabajo dio la pauta en los establecimientos de las reglas que se aplicarían a Internet (ver Figura 1.4).

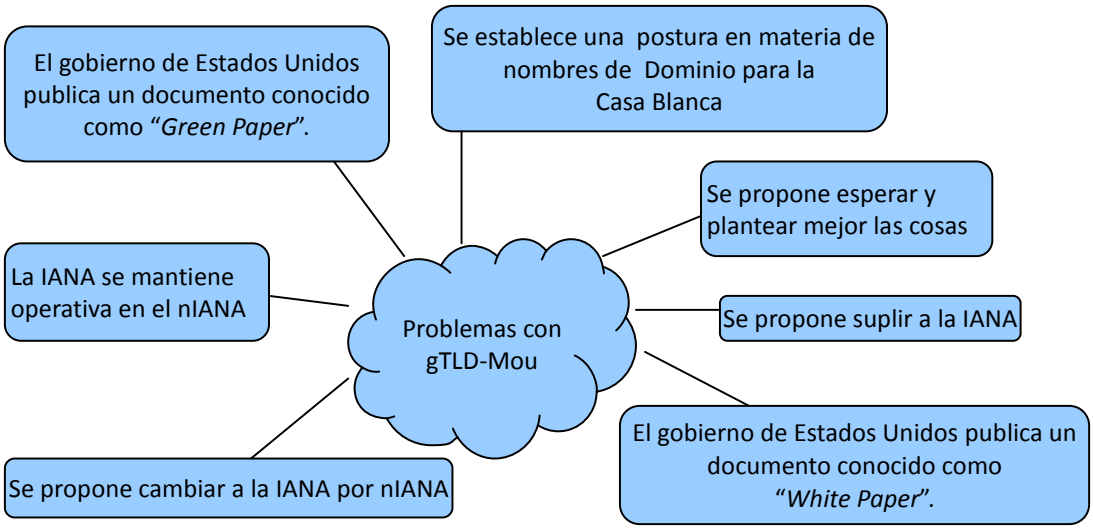


*Figura 1.4 Problemas con gTLD-MoU*

## 1.4 Transición IPv4 a IPv6 en la UNAM

En 1998 la UNAM (Universidad Nacional Autónoma de México) inició investigaciones sobre el protocolo IPv6 y con ello se constituye el proyecto IPv6 en nuestra Máxima Casa de Estudios. En el proyecto IPv6 se establecieron muchas pruebas y trabajos con temas como:

- *Stacks* IPv4/IPv6
- Túneles
- Software de conexión
- Aplicaciones multimedia
- Servidores para Web
- DNS
- Autoconfiguración

- Calidad de servicio
- IPv6 sobre ATM *(Asynchronous Transfer Mode)*
- Conexión con redes internacionales de IPv6 (6Bone, 6REN)
- IPv6 en Internet2

Una de las primeras pruebas realizadas en IPv6 fue la conexión a 6Bone, la cual fue una red mundial experimental y la puesta en operación de IPv6. En la red de 6Bone participaron 47 países, entre ellos México, donde la UNAM fue el primer nodo en el país, registrándose en junio de 1999.

Posteriormente la UNAM fue aceptada como uno de los 68 nodos de *Backbone* que en esa fecha operaban en 6Bone, a la UNAM se le delego un rango de direcciones tipo TLA(*Top-Level Aggregation*, 3ffe:8070::/28). La UNAM ha podido delegar direcciones y configurar túneles a instituciones en México y en el mundo interesadas en realizar pruebas con IPv6.

En el 2000 la UNAM obtuvo un bloque temporal del tipo sTLA (*sub Top-Level Aggregation*, 2001:0448::/35), adjudicado por ARIN (*American Registry for Internet Numbers*), la entidad de registro para Norteamérica y que en aquel entonces daba servicio también a Latinoamérica, y este bloque se ha utilizado en la Red CUDI (Corporación Universitaria para el Desarrollo de Internet A.C.) y la red de Internet2 de México.

Unos años más adelante, en junio de 2005 se obtiene otro bloque de direcciones IPv6 (2001:1218::/32) adjudicado por LACNIC (*Latin America and Caribbean Network Information Centre*), la entidad de registro para Latinoamérica y el Caribe. Con el bloque

adjudicado por LACNIC se pudo desarrollar y poner en marcha una red de pruebas en una primera etapa, y posteriormente con una red de producción, se instaló la Red IPv6 de la UNAM, la primera red IPv6 instalada en México. Esta red contó con varios túneles hacia otros nodos de *Backbone* de 6Bone: SPRINT, FIBERTEL, MERIT, BAY NETWORKS, JANET e ISI-LAP, y hacia los *hosts* que tiene la UNAM corriendo con sistemas operativos como Windows 2003, Windows 2000, Windows XP, Windows Vista, Windows 8, Solaris, Linux y BSD.

En enero del 2010 se puso en producción un Servidor de Túneles para ofrecer conexión automática con IPv6 en RedUNAM y salir a Internet también con IPv6.

A partir del 2011 comenzaron los planes y propuestas la actualización de infraestructura en la red de la UNAM comenzando por el nodo principal que se encuentra en la DGTIC (Dirección General de Cómputo y de Tecnologías de Información y Comunicación) principalmente de equipos de ruteo, *switches*, por mencionar algunos.

En el 2012 se adquieren los equipos nuevos para comenzar con el plan de actualización de la infraestructura de la red de datos comenzando con algunas áreas de la DGTIC como pruebas piloto además de que la UNAM participó en el día mundial de IPv6, pruebas que resultaron satisfactorias pues se tuvieron algunas páginas de la UNAM con soporte en IPv6.

Lo anterior fue posible gracias a que la UNAM cuenta con ISP (Proveedor de Servicio de Internet) que ya brindaran el servicio de IPv6 de manera nativa, es decir, sin la necesidad de usar túneles.

En el 2013 se comenzó con la actualización de todo el nodo de la DGTIC y el nodo de Zona Cultural con lo cual se logró un gran avance con la configuración de enlaces que proveen del servicio de Internet a varias dependencias de la UNAM y hacer uso del direccionamiento proporcionado por NIC-UNAM quien es la única autorizada para la asignación de los segmentos tanto en IPv4 como IPv6 en la UNAM.

Actualmente se sigue trabajando con instituciones mexicanas y de América Latina para realizar su conexión IPv6 hacia la UNAM. Entre las instituciones mexicanas han destacado: Instituto Politécnico Nacional, Universidad Autónoma Metropolitana, Instituto Tecnológico de Estudios Superiores de Monterrey, Universidad Autónoma de Chiapas, Universidad Autónoma de Guerrero, Universidad Autónoma del Estado de Hidalgo, Universidad Autónoma de Nuevo León, Instituto Tecnológico de Oaxaca, Instituto Tecnológico de Mérida, Instituto Tecnológico Autónomo de México, PEMEX, STYX, ASTER, etcétera.

Entre las instituciones latinoamericanas han estado: Instituto de Informática de la Universidad Austral de Chile y las universidades UBio-Bio, UFRO y UDLA; ex-RETINA ahora InnovaRed, y las universidades LINTI-UNLP, UBA, de Argentina; EAFIT y las universidades UdeA, UniCauca y UniPamplona de Colombia; INICTEL, NITCOM, y la UNI de Perú, etcétera. (Alcántara, 2012)

# Capítulo 2

# La necesidad de IPv6

## 2.1 Problemas existentes en IPv4

El protocolo utilizado para gestionar el tráfico de datos en la red es llamado TCP/IP pero en realidad el protocolo está formado por dos protocolos diferentes y que realizan acciones diferentes.

Uno de los protocolos es TCP, el cual se encarga del control de transferencia de datos y el otro protocolo es IP, que se encarga de la identificación del dispositivo en la red.

Los datos que circulan en Internet se les llama datagramas o paquetes, los datagramas son datos encapsulados en los cuales se les agrega un encabezado que contiene información sobre su transporte como la dirección IP origen y la dirección IP destino. Los enrutadores analizan los datos contenidos en un datagrama para que estos puedan llegar a su destino. (RFC791, 1981)

La versión 4 de este protocolo es la primera en ser implementada a gran escala definida en el RFC 791[6] y ha logrado ser un protocolo dominante en Internet.

En la década de los 80 la asignación de direcciones IP de clase A y el uso ineficiente por organizaciones que obtuvieron muchas más direcciones de las que necesitaban, creo un gran desperdicio de direcciones IP que no se utilizan y aunque las organizaciones utilicen direcciones IP públicas para dispositivos que no son accesibles fuera de sus redes locales, estos podrían utilizar la una implementación basada en NAT, así pudiendo dejar un alto rango de direcciones IP que se podrían utilizar.

[6] *Véase lista de RFCs*

Dentro del direccionamiento en IPv4 hay que tener en cuenta que no todas direcciones están disponibles para el protocolo IP público (el que utilizamos en nuestra conexión con Internet y que nos asigna nuestro ISP). Además dentro del rango de direcciones IP hay direcciones reservadas con usos específicos, lo que provoca que el número real de direcciones IP disponibles no sea tan elevado.

### 2.1.1 Agotamiento de direcciones IPv4

Con el gran crecimiento de internet hay varias causas por las cuales el agotamiento de direcciones IPv4 fue inminente, entre ellas están que cada vez más dispositivos requieren de una dirección IP para poder navegar en internet.

Las conexiones *Always-on* que en la década de los 90 predominó el acceso a internet mediante *dial-up*, reducía la presión en las direcciones IP porque los enlaces estaban normalmente desconectados, pero con el acceso de banda ancha que surgió, las conexiones permanecen activas e incluso cuando tienen asignadas dinámicamente una dirección, necesitan de una IP continua. Por lo tanto las direcciones IPv4 que son utilizadas para enrutar públicamente no son suficientes y mucho menos para proporcionar una dirección distinta para todos los dispositivos que lo requieran.

El problema de la falta de direcciones IPv4 se puede minimizar mediante diferentes soluciones:

**a) NAT** (*Network Address Translation*)- Esta solución nos permite que varios dispositivos en una red de área local (LAN) pueda compartir una dirección IP pública para tener acceso a Internet. Los datos enviados por dispositivos a Internet indican tanto su dirección fuente como la IP pública utilizada y el enrutador que proporciona el acceso es capaz de "seguir la pista" de qué dispositivo ha originado el tráfico en la red y así poder responder en consecuencia.

**b) Redes privadas-** En estas se pueden utilizar un rango de direcciones IP especificadas en el RFC 1918[7] y así poder asignarles direcciones cuando se requieran y si es necesario que estas se comuniquen con otras redes o tengan una salida a internet se tendría que hacer mediante una puerta de enlace con una dirección pública que normalmente será con NAT o con un servidor *proxy*.

**c) DHCP** (*Dynamic Host Configuration Protocol*)- Para que la asignación y configuración de direcciones IP sea de forma automática y dinámica, este protocolo se trata como cliente/servidor en donde generalmente el servidor posee una lista de direcciones IP y las va asignando a los clientes conforme éstas van estando libres.

Con estas técnicas se puede reducir la necesidad de más direcciones IP pero con el aumento de usuarios y dispositivos, el crecimiento de Internet y las redes IPv6 se ve como una solución a mediano y largo plazo por el agotamiento de las direcciones IPv4.

[7] *Véase lista de RFCs*

El 3 de febrero de 2011, la IANA asignó los últimos bloques libres a los RIRs, efectivamente agotando el pool de direcciones IPv4 disponibles. (The Number Resource Organization, 2011)

## 2.1.2 Problemas de Arquitectura

El gran crecimiento que ha experimentado Internet en los últimos años provoca que al protocolo IPv4 se le hagan modificaciones y se introduzcan protocolos complementarios con el fin de poder satisfacer la creciente demanda y lo que han causado es que las redes IP estén perdiendo paulatinamente el principio de conectividad punto a punto bajo el cual se diseñó IPv4.

Las direcciones IPv4 son direcciones de 32 *bits*, lo que nos genera $2^{32}$= 4.294.967.296 direcciones únicas, pero el gran crecimiento que ha tenido Internet, combinado con el hecho de que hay desperdicio de direcciones en muchos casos nos ha llevado al agotamiento de direcciones IPv4 y esta limitación nos ha estimulado a que se tenga que migrar hacia IPv6 el cual se espera termine remplazando a IPv4.

Actualmente no quedan direcciones IPv4 disponibles para compra, por ende se está en la forzosa y prioritaria obligación de migrar a IPv6, los sistemas operativos Windows (Vista, 7, 8), Unix/like (Gnu/linux, Unix, Mac OSX), BSD entre otros, tienen soporte nativo para IPv6, mientras que Windows XP y sistemas anteriores no tienen soporte nativo para este. (EcuRed, 2014)

## 2.2 Motivadores del cambio a IPv6

Con el protocolo IPv6 se ha mejorado algunas cosas respecto a IPv4, como la capacidad de autenticación y la privacidad de los datos transmitidos, una cabecera que garantiza que un paquete procede del origen que realmente se indica, mientras que en IPv4 el paquete podría venir de un origen distinto al indicado en la cabecera.

Se puede afirmar que aunque el funcionamiento del protocolo IPv4 ha sido satisfactorio, las razones por las cuales se motiva al cambio de IPv4 a IPv6 son:

- El sorprendente crecimiento del número de direcciones IP en uso.
- La necesidad de transmitir aplicaciones en tiempo real.
- La necesidad de mecanismos de seguridad.

No existe una fecha límite en la que se pueda cambiar totalmente a IPv6 y deshabilitar todas las redes IPv4, este proceso de migración o coexistencia debe de realizarse en forma progresiva para que tanto las personas encargadas de este proceso así como la infraestructura de las organizaciones puedan soportar totalmente IPv6. El tráfico de IPv6 aun no representa mucho del tráfico total de Internet y la mayoría corresponde a Universidades e instituciones que trabajan en el tema.

Para la migración, integración o coexistencia existen una serie de factores motivadores para la implementación a IPv6, como motivadores comerciales, políticos o técnicos.

**a) Motivadores Comerciales**

La implementación del protocolo IPv6 se puede ver como un movimiento estratégico ya que su implementación en las redes de las organizaciones les permite estar preparados para futuras necesidades de los clientes, así pudiendo crear una ventaja con respecto de la competencia.

En un plan de migración o coexistencia a IPv6 realizado con tiempo y planificación desde el punto de vista económico es mejor y más barato. Con el protocolo IPv6 se da la opción de mejorar la infraestructura de la organización e incluir nuevos productos y servicios para ser ofrecidos por empresas TIC (Tecnologías de la información y la comunicación).

**b) Motivadores Políticos**

Algunos gobiernos como el de Estados Unidos, Japón, China y Corea tienen como prioridad la implementación de IPv6, dando gran apoyo para las iniciativas que se manejen en este ámbito. Las olimpiadas de Beijing 2008 fueron un ejemplo de dichas políticas, toda su infraestructura de telecomunicaciones fue implementada mayoritariamente en IPv6.

**c) Motivadores Técnicos**

Con la necesidad de migrar hacia el protocolo IPv6 y que los nuevos equipos de red, sistemas operativos y dispositivos móviles proveen soporte para IPv6, los equipos que se utilizan en redes de datos como *switches*, *routers* y *firewalls* así como algunos ISP (*Internet Service Provider*) ya proveen conectividad IPv6 a usuarios finales, han llegado a un grado en el que permite implementar redes funcionales con el protocolo IPv6. (Cáceres & Ortiz, 2010)

## 2.3 Comparaciones entre IPv4 e IPv6

El esfuerzo que se le ha dado al nuevo estándar ha es causa del rápido crecimiento de Internet, y como consecuencia IPv6 está siendo introducido para superar las restricciones de IPv4 una de ellas es el espacio de direcciones (entre otras cosas).

El protocolo IPv6 crea un mayor espacio de direcciones que IPv4 y aunque no es todo, es uno de los aspectos más importantes. Otro punto importante que dirige el desarrollo de IPv6 es la necesidad de mayor seguridad en la transmisión de datos y un cifrado mejorado.

Con la comunicación privada a través de un medio público como lo es Internet se requieren de servicios con un cifrado que impida que los datos enviados puedan ser interceptados y modificados durante su transporte, por lo tanto para mejorar la seguridad existe un estándar que proporciona seguridad para los paquetes en IPv4 denominado IPsec (*Internet Protocol security*), aunque en IPv4 este estándar es opcional.

Examinando más a fondo el encabezado IPv4 (ver Tabla 2.1) podemos encontrar cómo está formado el paquete a transmitir para entender mejor la información que se transmite y así poder identificar las diferencias entre IPv4 e IPv6.

**Dentro del encabezado de IPv4 podemos encontrar:**

- **Versión**: Este campo describe el formato de la cabecera que en este caso es la versión 4 del protocolo IP.

- **Longitud del encabezado:** Corresponde al largo en número de palabras de 32 *bits* del encabezado (*Internet Header Length*).

- **Tipo de servicio:** Utiliza 8 *bits* para determinar la prioridad del datagrama a transmitir. Proporciona una indicación de los parámetros de la calidad de servicio deseada.

- **Longitud total del datagrama:** Indica el tamaño total del datagrama en *bytes*. El tamaño total del datagrama no puede exceder los 65536 *bytes* y se utiliza junto con el tamaño del encabezado, este campo permite determinar dónde se encuentran los datos.

- **Identificación:** Es un valor de identificación del paquete que se utiliza al reensamblar los fragmentos del datagrama.

- **Banderas:** Son indicadores de control que utiliza 3 *bits*

- Bit 0: reservado, siempre debe ser cero.
- Bit 1: (DF) No Fragmentar (*Don't Fragment*) si es 0 puede fragmentarse y si es 1 no se fragmenta.
- Bit 2: (MF) Más Fragmentos (*More Fragments*) si es 0 es el último fragmento y si es 1 hay más fragmentos.

- **Margen del Fragmento:** El campo indica a que parte del datagrama pertenece el fragmento.

- **Tiempo de vida (TTL)**. Este campo especifica el número máximo de enrutadores por los que puede pasar un datagrama. El campo disminuye cada vez que pasa por un enrutador y cuando alcanza el valor de 0, el enrutador destruye el datagrama. Esto hace que los datagramas que son imposibles de entregar sean descartados.

- **Protocolo:** Este campo permite saber de qué protocolo proviene el datagrama. ICMP (1), IP (4), TCP (6), UDP (17).

- **Suma de comprobación del encabezado:** Este campo contiene un valor codificado en 16 *bits* que permite controlar la integridad del encabezado dado que algunos de los campos del encabezado cambian cada vez que es procesada, y establece si se ha modificado durante la transmisión, si no coincide se descarta el paquete.

-D**irección IP origen y dirección IP destino**: 32 *bits* estructurados para identificar la red y el nodo dentro de la red.

(RFC791, 1981)

*Tabla 2.1 Encabezado de IPv4*

<table>
<tr><td>Versión<br>(4 bits)</td><td>Longitud del Encabezado<br>(4 bits)</td><td colspan="2">Tipo de Servicio<br>(8 bits)</td><td colspan="2">Longitud Total<br>(16 bits)</td></tr>
<tr><td colspan="3">Identificación<br>(16 bits)</td><td>Banderas<br>(3 bits)</td><td>Margen del fragmento<br>(13 bits)</td></tr>
<tr><td colspan="2">Tiempo de Vida<br>(8 bits)</td><td>Protocolo<br>(8 bits)</td><td colspan="2">Suma de comprobación del encabezado<br>(16 bits)</td></tr>
<tr><td colspan="5">Dirección IP origen<br>(32 bits)</td></tr>
<tr><td colspan="5">Dirección IP destino<br>(32 bits)</td></tr>
<tr><td colspan="5">Datos</td></tr>
</table>

Entonces conociendo el paquete IPv4 podemos mencionar algunas de las principales diferencias entre IPv4 e IPv6 (ver Tabla 2.2):

*Tabla 2.2 Comparativo entre los Protocolos de Internet IPv4 e IPv6* **(Rodriguez, 2003)**

| **IPv4** | **vs** | **IPv6** |
|---|---|---|
| Espacio de direcciones de 32 *bits*, $2^{32}$ direcciones IP posibles. | | Espacio de direcciones de 128 *bits*, $2^{128}$ direcciones IP posibles. |
| Configuración manual o Dinámica (DHCP). | | Configuración *Plug & Play*, Manual o Dinámica (DHCPv6). |
| Políticas de Calidad de Servicio se realizan a través del campo Tipo de Servicio (ToS) del paquete IP. | | Políticas de Calidad de Servicio se realizan a través de los campos Etiqueta de Flujo y Clase de Tráfico. |
| La Seguridad es algo opcional, a través de IPsec. | | La Seguridad extremo a extremo implementada en forma nativa. |
| Protocolo no escalable. | | Protocolo escalable. |
| No existe ninguna identificación de flujo de paquetes para que los enrutadores controlen la QoS (*Quality of Service*) en el encabezado IPv4. | | Se incluye la identificación del flujo de paquetes para que los enrutadores controlen la QoS (*Quality of Service*) en el encabezado IPv6, utilizando el campo *Flow Label* (etiqueta de flujo). |
| La fragmentación se lleva a cabo en los enrutadores y el *host* que realiza el envío. | | La fragmentación no la llevan a cabo los enrutadores, sino únicamente el *host* que realiza el envío. |
| En el encabezado incluye una suma de comprobación. | | En el encabezado no incluye una suma de comprobación. |
| El Protocolo de resolución de direcciones (ARP) utiliza los marcos de solicitud ARP de difusión para resolver una dirección IPv4 como una dirección de capa de vínculo. | | Los marcos de solicitud ARP se sustituyen por mensajes de solicitud de vecinos de multidifusión. |
| Se utiliza el Protocolo de administración de grupos de Internet (IGMP). | | IGMP se sustituye con los mensajes de Descubrimiento de escucha de multidifusión (MLD). |
| Se utiliza el Descubrimiento de enrutadores ICMP, y es opcional. | | El Descubrimiento de enrutadores ICMP queda sustituido por la Solicitud de enrutadores ICMPv6 y los mensajes de anuncio de enrutador, y es obligatorio. |
| La dirección de multidifusión se utiliza para enviar tráfico a todos los nodos de una subred. | | No hay direcciones de multidifusión IPv6. De forma alternativa, se utiliza una dirección de multidifusión para todos los nodos de ámbito local del vínculo. |
| Utiliza registros de recurso (A) de dirección de *host* en el Sistema de nombres de dominio (DNS) para correlacionar nombres de *host* con direcciones IPv4. | | Utiliza registros de recurso (AAAA) de dirección de *host* en el Sistema de nombres de dominio (DNS) para correlacionar nombres de *host* con direcciones IPv6. |
| Utiliza registros de recurso (A) de puntero en el dominio DNS IN-ADDR.ARPA para correlacionar direcciones IPv4 con nombres de *host*. | | Utiliza registros de recurso (PTR) de puntero en el dominio DNS IP6.INT para correlacionar direcciones IPv6 con nombres de *host*. |
| Debe admitir un tamaño de 576 *bytes* (posiblemente fragmentado). | | Debe admitir un tamaño de 1280 *bytes* (sin fragmentación). |

## 2.4 Ventajas de IPv6

Desde hace algunos años las organizaciones se están preparando para la transición al protocolo IPv6, una de las grandes razones detrás de la necesidad del IPv6 es que las direcciones IPv4 se han agotado lo que la versión 6 de este protocolo resuelve esta situación.

En algunos de los dispositivos y sistemas operativos que no son tan nuevos el protocolo IPv6 se puede instalar como una actualización del software, ya que puede soportar el hardware más nuevo, necesitando únicamente de su instalación y configuración.

La ventaja más grande es el espacio de dirección extendido de 32 *bits* a 128 *bits*, esto permite solucionar el problema del agotamiento de las direcciones.

Con los mecanismos diseñados para la transición se puede ir introduciendo el protocolo IPv6 sin la necesidad de afectar la mayoría de las redes existentes en IPv4.

El mecanismo de autoconfiguración sin estado, permite que los dispositivos que requiere una dirección IPv6, utilicen un prefijo global que le permite autoconfigurarse, usando su identificador MAC (*Media Access Control*) o un número aleatorio privado para construir su propia y única dirección IP.

De este modo ya no hay la necesidad de utilizar servidores DHCP ya que las direcciones IP pueden ser asignadas automáticamente y dinámicamente por el dispositivo del cliente. Aunque todavía se pueden asignar las direcciones IP mediante DHCPv6.

La simplificación del formato en el encabezado de IPv6 permite que el procesamiento sea mucho más rápido que con el protocolo IPv4 ya que tiene una longitud fija de 40 *bytes*. Además de que en el protocolo IPv6 ya no hay suma de comprobación porque casi todo el contenido que es enviado mediante redes IPv6 tienen su propio mecanismo de control de errores y ya no hay la necesidad de algún mecanismo a nivel de IP lo que ayudaría a disminuir la carga en la transmisión de datos haciendo que la conexión sea más rápida.

Otra ventaja del protocolo IPv6 es la capacidad de asignar dos o más direcciones al mismo dispositivo, lo que nos beneficia al poder estar conectado a varias redes al mismo tiempo y nos da una mayor flexibilidad ya que las aplicaciones podrán elegir la red que necesitan y así no tendría que alternar entre las redes.

El protocolo IPv6 aun con las mejoras respecto al protocolo IPv4 todavía le falta recorrer un gran camino para que logre ser una presencia importante en el tráfico de Internet como lo es ahora IPv4. (Sánchez, 2006)

# Capítulo 3

# El protocolo IPv6

## 3.1 Características del Protocolo IPv6

La versión del protocolo de Internet versión 6 está definida en el RFC 2460 el cual fue diseñado para reemplazar al protocolo IPv4 definido en el RFC 791. Mientras que IPv4 posibilita $2^{32}$ direcciones de red diferentes, un número que ya no es suficiente para la gran cantidad de dispositivos que actualmente requieren de una dirección IP y debido a que el esquema de direcciones de 128 *bits* provee una gran cantidad de direcciones IP, con la posibilidad de asignar direcciones únicas globales a nuevos dispositivos ya que admite $2^{128}$ (340.282.366.920.938.463.463.374.607.431.768.211.456 o 340 sextillones de direcciones aproximadamente), hace que sea un protocolo adecuado para cubrir la necesidad de falta de direcciones para la gran cantidad de dispositivos que se pueden conectar a internet, además de los dispositivos que en el futuro se requieran conectar.

Dentro de los principales cambios de IPv6 respecto a IPv4 se encuentran:

**a) La capacidad de direccionamiento extendida-** Incrementa el tamaño de la dirección IP de 32 bits a 128 bits.

**b) Se simplifica el formato del encabezado-** El nuevo encabezado de IPv6 es más sencillo que el de IPv4, si se compara con el encabezado de IPv4 se removieron 6 campos: Longitud de encabezado, Identificación, Banderas, Desplazamiento por fragmentación, Suma de verificación de encabezado, Opciones y Relleno. Dado que el encabezado de IPv6 contiene menos campos y es de longitud fija se obtiene una reducción en el tiempo que le toma procesar los datos a los enrutadores al momento de enviar los paquetes de IPv6, lo que conlleva a una mayor eficiencia de la red.

**c) Se mejora el soporte para las extensiones y opciones-** Se cambia el campo opciones que pertenece a IPv4 y en cambio se agregan las extensiones de encabezado, así la manera en que se codifican las opciones del encabezado IP permiten un reenvío más eficiente, límites menos rigurosos en la longitud de opciones y mayor flexibilidad para introducir nuevas opciones.

**d) Se agrega la capacidad de etiquetado de flujo-** Se agrega para permitir el etiquetado de paquetes por el nodo fuente donde el remitente solicita un tratamiento especial, como la calidad de servicio no estándar o el servicio en tiempo real. El campo está dirigido al procesamiento de la estación destino, no para los enrutadores.

**e) La capacidad de autenticación y privacidad-** Existen extensiones para utilizar la autenticación, integridad de los datos, y confidencialidad de los datos. (RFC2460, 1998)

El protocolo IPv6 para aumentar el cifrado y autenticación de los datos hace uso de IPsec que forma parte del protocolo IPv6 a diferencia de IPv4 en donde es opcional.

Las funciones del protocolo IPsec que desarrolla son:

- Limitar el acceso a sólo aquellos autorizados.
- Certifica la autenticación de la persona que envía los datos.
- Cifra los datos transmitidos a través de la red.
- Asegura la integridad de los datos.
- Invalida la repetición de sesiones, para evitar que no sean repetidas por usuarios maliciosos.

Hay dos protocolos que IPsec utiliza para proporcionar servicios de seguridad:

- AH (Autenticación de Encabezado, *Autentication Header*,)
- *ESP* (Carga de Seguridad Encapsulada, *Encapsulated Security Payload*).

Tanto AH y ESP ofrecen control de acceso, por medio de la distribución de claves criptográficas y la gestión del tráfico flujos. Al estar incluidos en la implementación de IPv6 se provee mayor seguridad ya que IPsec está presente en todos los nodos de la red. (RFC4301, 2005)

## 3.2 Estructura de un Paquete IPv6

En el RFC 2460, se especifica el encabezado del protocolo IPv6 que consta de 8 campos, 4 menos que el de IPv4. Entre las mejoras propuestas se encuentra el campo etiqueta de flujo y las extensiones de encabezado. A continuación se presentan todos los campos con su descripción (ver Tabla 3.1):

**-Versión:** Se refiere a la versión de IP que para IPv6 tiene el valor 6.

**-Clase de tráfico:** Este campo está disponible para usarse por los nodos y/o enrutadores para identificar y distinguir entre las diferentes clases o prioridades de paquetes IPv6.

**-Etiqueta de flujo:** Sirve para establecer un flujo o secuencia de paquetes IPv6 para permitir el etiquetado de paquetes por el nodo fuente donde el remitente solicita un tratamiento especial, como la calidad de servicio no estándar o el servicio en tiempo real..

**-Longitud del campo de datos:** Carga útil del datagrama es la parte que sigue al encabezado de IPv6.

**-Siguiente Encabezado:** Define el tipo de información que va a seguir al siguiente encabezado de IPv6 o puede ser alguna de las extensiones de encabezado.

**-Límite de saltos:** Define el número máximo de enrutadores que un paquete IP puede atravesar. Cada salto disminuye el valor por 1, en el caso que el campo llegue a contener el valor 0 el paquete es descartado.

**-Dirección Origen:** Identifica la dirección fuente IPv6 del transmisor.

**-Dirección Destino:** Muestra la dirección destino IPv6 del paquete.

*Tabla 3.1 Estructura de un paquete IPv6*

<table>
<tr><td>Versión<br>(4 bits)</td><td>Clase de<br>Tráfico<br>(8 bits)</td><td colspan="3">Etiqueta de Flujo<br>(20 bits)</td></tr>
<tr><td colspan="2">Longitud del Campo de Datos<br>(16 bits)</td><td colspan="2">Siguiente Encabezado<br>(8 bits)</td><td>Límite de Saltos<br>(8 bits)</td></tr>
<tr><td colspan="5">Dirección Origen (128 bits)</td></tr>
<tr><td colspan="5">Dirección Destino (128 bits)</td></tr>
</table>

En IPv6 la fragmentación se realiza sólo en el nodo origen del paquete, al contrario que en IPv4 en donde los enrutadores pueden fragmentar un paquete. En IPv6, las opciones también desaparecen del encabezado estándar y son especificadas por el campo " Siguiente Encabezado", similar en funcionalidad en IPv4 al campo Protocolo. (RFC2460, 1998)

## 3.3 Formato de una dirección IPv6

Las direcciones IPv6 son de un tamaño de 128 *bits* de longitud y se interpretan como ocho grupos de cuatro dígitos hexadecimales. En la siguiente tabla se muestran algunos ejemplos de de direcciones IPv6:

*Tabla 3.2 Ejemplos de dirección IPv6*

| Dirección IPv6 |
|---|
| 2001:1db8:85a3:1111:1319:8a2e:0310:1334 |
| 3ffe:1200:018d:5611:0000:abcd:1134:0332 |
| 0000:0ab0:5121:ce22:33de:4fb4:0000:2e45 |

Si un grupo de la dirección IPv6 es nulo (0000), se puede comprimir (Ver Tabla 3.3). :

*Tabla 3.3 Ejemplos de formato comprimido 1*

| Formato Preferido | Formato Comprimido Utilizando “ :: ” |
|---|---|
| 2001:0db8:85a3**:0000:**1319:8a2e:0370:7344 | 2001:0db8:85a3**::**1319:8a2e:0370:7344 |
| 2001:0010**:0000:**ffff:fb00:0022:5050:45a3 | 2001:0010**::**ffff:fb00:0022:5050:45a3 |
| 3ffe:1200:018d:5611**:0000:**abcd:1134:0332 | 3ffe:1200:018d:5611**::**abcd:1134:0332 |

Si más de dos grupos consecutivos son nulos se pueden comprimir como "::".

(Ver Tabla 3.4):

*Tabla 3.4 Ejemplos de formato comprimido 2*

| Formato Preferido | Formato Comprimido Utilizando “ :: ” |
|---|---|
| 0000:0000:0000:0000:0000:0000:0000:0000 | **::** |
| 0000:0000:0000:0000:0000:0000:0000:0001 | **::**0001 |
| 3ffe:1200: 0000:0000:0000**:**abcd:1134:0332 | 3ffe:1200**::**abcd:1134:0332 |

Así, en la siguiente tabla se pueden observar las diferentes representaciones posibles de una misma dirección:

*Tabla 3.5 Ejemplos de formato comprimido 3*

| Formato Preferido | Formato Comprimido Utilizando " :: " |
|---|---|
| 2001:0db8:0000:0000:0000:0000:0370:7344 | 2001: 0db8:0000:0000:0000**::**0370:7344 |
| | 2001: 0db8**:0:0:0:0:**0370:7344 |
| | 2001: 0db8**:0::0:**0370:7344 |
| | 2001: 0db8**::**0370:7344 |

Si la dirección tiene más de una serie de grupos nulos consecutivos la compresión sólo se permite en uno de ellos ya que si se agrupan los ceros en dos o más grupos no quedaría claro cuántos grupos de ceros hay de cada lado (ver Tabla 3.6). :

*Tabla 3.6 Ejemplos de formato comprimido 4*

| Formato Preferido | Formato Comprimido Utilizando " :: " |
|---|---|
| 2001:0db8:0000:0000:1319:8a2e:0000:0000 | 2001:0db8**::**1319:8a2e:**0000:0000** |
| | 2001:0db8**:0000:0000:**1319:8a2e**::** |
| 3ffe:0000:0000:0001:0000:0000:ab34:0002 | 3ffe**:0000:0000:**0001**::**ab34:0002 |
| | 3ffe**::**0001**:0000:0000:**ab34:0002 |
| 2001:0410:0000:0000:fb00:0000:0000:45ff | 2001:0410**:0000:0000:**fb00**::**45ff |
| | 2001:0410**::**fb00**:0000:0000:**45ff |

Los ceros iniciales en un grupo también se pueden omitir (ver Tabla 3.7). :

*Tabla 3.7 Ejemplos de formato comprimido con ceros iniciales en cada grupo*

| Formato Preferido | Formato Comprimiendo "0" y Utilizando " :: " |
|---|---|
| 2001**:0db8**:85a3**:0000:**1319:8a2e:**0370:**7344 | 2001:**db8:**85a3**::**1319:8a2e:**370:**7344 |
| 2001**:0410:0000:**1234:fb00:1400:5000:45ff | 2001**:410::**1234:fb00:1400:5000:45ff |
| 3ffe:0b00:0c18:0001:0000:1234:ab34:0002 | 3ffe:**b00:c18:1::**1234:ab34**:2** |

Si la dirección es una dirección IPv6 mapeada a IPv4, los últimos 32 *bits* pueden escribirse en base decimal (ver Tabla 3.8):

*Tabla 3.8 Ejemplo de dirección IPv6 mapeada a IPv4*

| Dirección IPv4 | Dirección IPv4 Mapeada |
|---|---|
| 192.168.1.140 | ::ffff:192.168.1.140 |
| | ::ffff:c0a8:18c |

No se debe confundir con una dirección IPv6 compatible con IPv4 (ver Tabla 3.9):

*Tabla 3.9Ejemplo de dirección IPv6 compatible con IPv4*

| Dirección IPv4 | Dirección IPv4 Compatible |
|---|---|
| 192.168.1.140 | ::192.168.1.140 |
| | :: c0a8:18c |

El formato ::ffff:X.X.X.X se denomina dirección IPv6 mapeada a IPv4, y el formato ::X.X.X.X se denomina dirección IPv6 compatible con IPv4. (RFC5952, 2010) (RFC6052, 2010)

## 3.3.1 Identificación de los tipos de direcciones

Los tipos de direcciones IPv6 pueden identificarse tomando en cuenta los rangos definidos por los primeros *bits* de cada dirección. Cuando lo que se desea es identificar un rango de direcciones diferenciable por medio de los primeros *bits*, se añade este número de *bits* tras el carácter de barra "/" (Ver Tabla 3.10):

*Tabla 3.10Ejemplo de identificador de rango*

| Dirección IPv6 | Identificador |
|---|---|
| 2001:0DB8::1428:57AB/96 | 2001:0DB8:: |
| 2001:0DB8::874B:2B34/96 | |
| 2001:0DB8::de34:1b12/96 | |

**a) ::/128 o 0000:0000:0000:0000:0000:0000:0000:0000 -**La dirección con todo ceros se utiliza para indicar la ausencia de dirección, es una dirección *unicast* que no se asigna a alguna interface y es usada para propósitos especiales.

**b) ::1/127 o 0000:0000:0000:0000:0000:0000:0000:0001-** Al igual que en IPv4, cada dispositivo tiene una dirección *loopback* que es una dirección que puede usar un nodo para enviarse paquetes a sí mismo y no puede asignarse a ninguna interfaz física.

**c) ::X.X.X.X/96 -**La dirección IPv6 compatible con IPv4 se usa en computadoras y enrutadores como un mecanismo de transición para crear automáticamente túneles IPv4 en las redes duales IPv4/IPv6. De esa forma se entregan paquetes IPv6 sobre redes IPv4.

En la siguiente figura se muestra el formato descriptivo de una dirección IPv6 compatible con IPv4. En éste el prefijo se crea con el bit puesto a cero del de más alto nivel de los 96 *bits*, y los restantes 32 *bits* de menor nivel representan la dirección IPv4 en formato decimal.

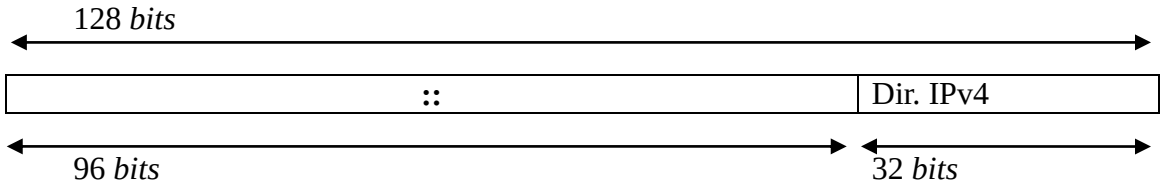


*Figura 3.1 Formato descriptivo de una dirección IPv6 compatible con IPv4*

**d) ::ffff:X.X.X.X/96 -**La dirección IPv6 mapeada a IPv4 se usa como mecanismo de transición en terminales duales y se utiliza sólo en el ámbito local de nodos que tienen las direcciones IPv4 e IPv6 (ver Figura 3.2). Los nodos usan direcciones IPv6 mapeadas a IPv4 de forma interna solamente. Estas direcciones no son conocidas afuera del nodo y no llegan al cable de comunicación como direcciones IPv6.

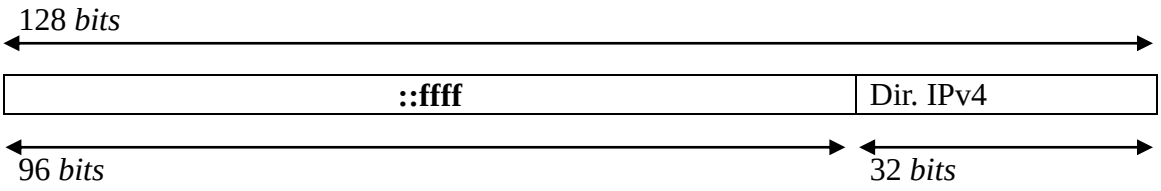


*Figura 3.2Dirección IPv6 mapeada a IPv6*

**e) fe80::/10 -**Enlace Local (*Link-Local*) Se usa para mecanismos de autoconfiguración, descubrimiento de vecinos y en redes sin enrutadores. Es útil para crear redes temporales y puede ser utilizada sin un prefijo global (ver Tabla 3.11).

*Tabla 3.11 Formato de Dirección Link-Local*

| 1111 1110 10 (FE80) | 0 | Identificador de Interface |
|---|---|---|
| 10 *Bits* | 54 *Bits* | 64 *Bits* |

**f) fec0:: -**El prefijo de sitio local (*site-local*) contiene información de subred dentro de la dirección y específica que la dirección sólo es válida dentro de una organización local, son enrutadas dentro de un sitio, pero los enrutadores no deben enviarlas fuera de éste (ver Tabla 3.12) .

En el RFC 3879[8] está declarado obsoleto, estableciendo que los sistemas futuros no deben implementar ningún soporte para este tipo de dirección especial. Se deben sustituir por direcciones *Unicast*.

*Tabla 3.12 Formato de Dirección Site-local*

| 1111 1110 11(FEC0) | 0 | Identificador de Subred | Identificador de Interface |
|---|---|---|---|
| 10 *Bits* | 38 *Bits* | 16 *Bits* | 64 *Bits* |

**g) ff00::/8 -**El prefijo se usa para las direcciones *multicast* (ver Tabla 3.13).

Hay que resaltar que no existen las direcciones de *broadcast* en IPv6, aunque la funcionalidad puede utilizarse con la dirección *multicast* FF01::1/128, la cual denomina a todos los nodos (*all nodes*).

*Tabla 3.13 Prefijos de multicast*

| Dirección *Multicast* | Área de Funcionamiento | Significado | Descripción |
|---|---|---|---|
| FF01::1 | Nodo | Todos los nodos | Todos los nodos en la interface local |
| FF01::2 | Nodo | Todos los enrutadores | Todos los enrutadores en la interface local |
| FF02::1 | Enlace Local | Todos los nodos | Todos los nodos en el enlace local |
| FF02::2 | Enlace Local | Todos los enrutadores | Todos los enrutadores en el enlace local |
| FF05::2 | Sitio | Todos los enrutadores | Todos los enrutadores en un sitio |

[8] *Véase lista de RFCs*

**h) Nodo Solicitado *Multicast*.** Es un tipo de dirección en la que se debe unir cada nodo por cada dirección *unicast* y *anycast* asignada. La dirección se conforma con los 24 *bits* de bajo nivel de una dirección IPv6, a esta dirección se le agrega el prefijo FF02:0:0:0:0:1:FF00::/104, de tal manera que el rango de direcciones *Multicast* de Nodo Solicitado va de FF02:0:0:0:0:1:FF00:0000 a FF02:0:0:0:0:1:FFFF:FFFF.

**i) *Agregable Global*.** Son similares a las direcciones *unicast* usadas para comunicarse a través de Internet en IPv4. Su estructura permite una agregación estricta de prefijos de enrutamiento para limitar el tamaño de la tabla de enrutamiento global de Internet.

Cada Dirección Agregable Global consta de tres partes (ver Figura 3.3):

- Prefijo recibido del proveedor: El prefijo asignado a una organización por un proveedor debe ser al menos de 48 *bits* (recomendado por el RFC 3177[9]).
- Sitio: La organización puede usar los *bits* 49 a 64 (16 *bits*) del prefijo recibido para subredes.
- Computadora: Representa los 64 *bits* de más bajo orden de la dirección.

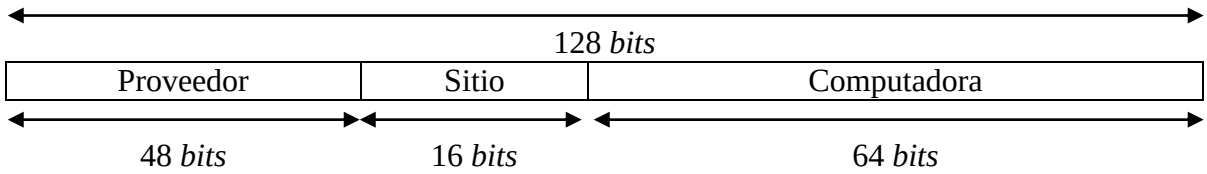


*Figura 3.3 Dirección Agregable Global*

(RFC4291, 2006)

[9] *Véase lista de RFCs*

## 3.4 Direccionamiento IPv6

Dentro de los tipos de direcciones IPv6 creados se encuentran las direcciones *Unicast, Multicast y Anycast*:

- *Unicast*. La dirección IPv6 de tipo *unicast* se utiliza para identificar la interface de un nodo IPv6, en el cual un paquete que es enviado a una dirección *unicast* es entregado a la interface identificada por esa dirección. (Ver Figura 3.4)

- *Multicast*. La dirección IPv6 de tipo *multicast* se utiliza para identificar a un grupo de interfaces IPv6, en el cual un paquete que es enviado a una dirección *multicast* es procesado por todos los miembros del grupo *multicast*. (Ver Figura 3.5)

- *Anycast*. La dirección IPv6 de tipo *anycast* se utiliza para asignar a múltiples interfaces IPv6, en el cual un paquete enviado a una dirección *anycast* es entregado a una de las interfaces que usualmente es la más cercana. (Ver Figura 3.6)

Cada uno de los tres tipos se subdivide en direcciones diseñadas para resolver casos específicos de direccionamiento IP.

### 3.4.1 *Unicast*

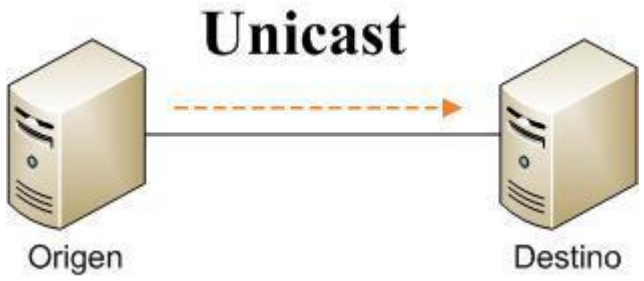


*Figura 3.4. Unicast*

Todas las interfaces están obligadas a tener al menos una dirección *unicast* de enlace local. Sin embargo, una característica fundamental de IPv6 es que una única interfaz puede tener múltiples direcciones IPv6 de cualquier tipo (*unicast*, *anycast* y *multicast*).

Los tipos de direcciones que se pueden clasificar dentro de *unicast* son:

- Enlace Local (*Link-Local*)
- Sitio Local (*Site-local*)
- Agregable Global
- Loopback
- Sin-Especificar (*Unspecified*)
- Compatible con IPv4

## 3.4.2 *Multicast*

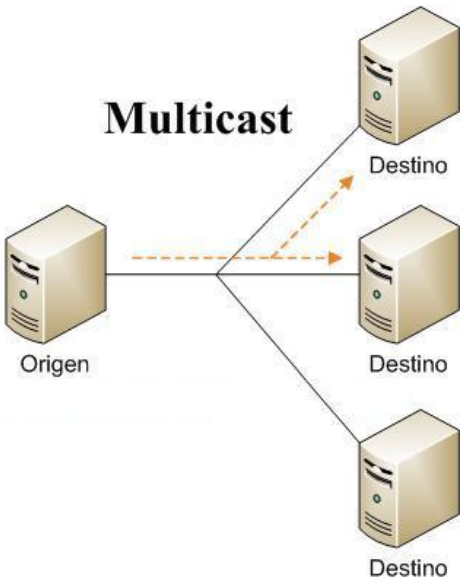


Figura 3.5 *Multicast*

Como el protocolo IPv6 no implementa *broadcast*, se puede lograr el mismo efecto enviando un paquete al grupo de *multicast* de *Link-Local* a todos los nodos. Por lo tanto la dirección más alta de la red es considerada una dirección normal en IPv6.

El *multicast* IPv6 comparte algunas características comunes con IPv4, pero también incorpora cambios y mejoras. Incluso cuando se le asigne a una organización el más pequeño de los prefijos de ruteo global IPv6, ésta también tiene la posibilidad de usar los grupos *multicast* IPv6 enrutables para asignarlos a las aplicaciones *multicast* entre dominios (RFC 3306[10]).

Los tipos de direcciones que se pueden clasificar dentro de *multicast* son:

- Asignada (*Assigned*).
- Nodo Solicitado (*Solicited Node*).

[10] *Véase lista de RFCs*

### 3.4.3 *Anycast*

Con la creación del protocolo IPv6 se define un nuevo tipo de dirección llamada *anycast*. En el cual hay una asociación de una dirección destino a varias máquinas y se selecciona una de estas máquinas para ser la destinataria de la información, lo más común es que el paquete se entregue a la máquina más cercana y esto dependerá de la topología de la red y del protocolo de enrutamiento que se esté utilizando.

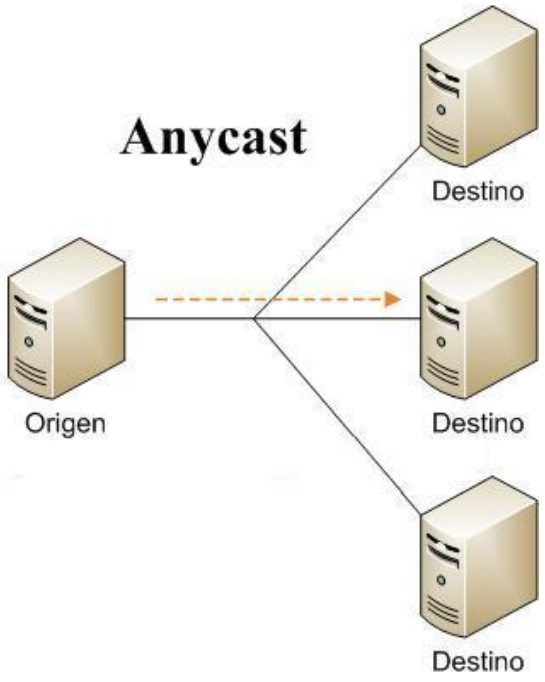


*Figura 3.6 Anycast*

Los tipos de direcciones que se pueden clasificar dentro de *anycast* son:

- Agregable Global.
- Sitio Local (*Site Local*).
- Enlace Local (*Link Local*).

(RFC2373, 1998)

## 3.5 Mecanismos de configuración de direcciones IPv6

Los mecanismos de configuración de direcciones nos permiten configurar las interfaces IPv6 para obtener una dirección IPv6, verificar que no esté duplicada y determinar la información que se ha de ser autoconfigurada.

Con la autoconfiguración definida en el RFC 2462[11] conocida como Configuración Automática de Dirección Sin Estado IPv6 los nodos IPv6 pueden configurarse a sí mismos cuando son conectados a una red ruteada en IPv6 usando los mensajes de descubrimiento de enrutadores de ICMPv6.

Mientras que con la Configuración de Direcciones con Estado IPv6 es posible utilizar DHCPv6 (*Dynamic Host Configuration Protocol* versión 6) o en su defecto los nodos pueden ser configurados en forma estática. (Palet & Cabellos, 2004)

Para que el proceso de renumeración de direcciones IPv6 sea transparente hacia los usuarios finales, se utiliza el mecanismo de autoconfiguración el cual permite una renumeración simple que consiste en enviarles un nuevo prefijo IPv6 *unicast* para la red a los dispositivos que se configuren.

[11] *Véase lista de RFCs*

# Capítulo 4

## Servidor de Nombres de Dominio (DNS/BIND) con IPv6

# 4.1 BIND

Hoy en día la en las redes de datos, incluyendo Internet, los usuarios localizan a otros equipos por medio de un nombre canonico, esto hace más fácil que los usuarios obtengan los recursos que se encuentran en las redes de datos e Internet y ya no se ve en la necesidad de recordar la dirección numérica a la que responden. (Red Hat, Inc., 2005)

## 4.1.1 Acerca de BIND

BIND (*Berkeley Internet Name Domain*, anteriormente *Berkeley Internet Name Daemon*) es el servidor de DNS más usado en Internet, especialmente en sistemas UNIX y Linux, y es patrocinado por la ISC (*Internet Systems Consortium*). Fue creado originalmente por estudiantes de la Universidad de California y liberado por primera vez en el BSD 4.3.

La versión 9 de BIND fue desarrollada desde cero para superar las dificultades arquitectónicas que estaban presentes en las versiones anteriores con el motivo de auditar el código en las primeras versiones de BIND, y también para incorporar DNSSEC (*DNS Security Extensions*).

Algunas de las mejoras más importantes de BIND 9 incluyen la integración del protocolo IPv6, TSIG (*Transaction SIGnature*), notificación DNS, nsupdate, rndc flush, vistas, procesamiento en paralelo, y una arquitectura mejorada en cuanto a portabilidad.

Actualmente todas las versiones de BIND anteriores a la 9 contienen vulnerabilidades que pueden causar conflictos con el proceso *named*, por lo cual se recomienda actualizarse a la última versión estable. (BIND, 2013)

### 4.1.2 Tipos de servidores de nombres de dominio

- **Autoritativo** — Es el representante oficial de una zona.
- **No Autoritativo** — Responde una consulta a partir de su caché y desconoce si los datos son válidos.
- **Maestro o primario** — Almacena los registros de las zonas originales, de autoridad para un cierto espacio de nombres y responde a consultas sobre el espacio de nombres de otros servidores de nombres.
- **Esclavo o secundario** — Responde a las peticiones que provienen de otros servidores de nombres y obtienen la información de sus espacios de nombres desde los servidores maestros.
- **Sólo caché** — Responde a las peticiones pero no tiene ninguna autoridad sobre ninguna zona. Las respuestas en general se introducen en un caché por un período de tiempo fijo, la cual es especificada por el registro de zona consultado.
- **Reenvío** — Reenvía las peticiones a una lista específica de servidores. Si ninguno de los servidores de nombres especificados puede resolver los nombres, la resolución falla.
- **Recursivo**— Hace consultas hasta que devuelve una respuesta o un error.
- **No recursivo**— Si no es capaz de responder la consulta la envía a otro servidor.
- **Distribución**— Es un servidor que solo es visible desde dentro de un dominio, aunque puede ser visible para cualquiera que conozca su dirección IP.

Un servidor de nombres puede ser uno o más de estos tipos. Por ejemplo, un servidor de nombres puede ser un servidor maestro para ciertas zonas, un servidor esclavo para otras zonas y sólo caché para algunas zonas.

Usualmente cuando se implementa BIND solamente se utiliza el demonio *named* para proporcionar el servicio de nombres de dominio, sin embargo, en la version 9 se agregan características avanzadas que permiten un servicio DNS más seguro y avanzado. (Red Hat, Inc., 2005)

### 4.1.3 Mejoras al protocolo DNS

**a) IXFR (Transferencias de Zona Incremental, *Incremental Zone Transfers*)-** Su función es que un servidor DNS tipo esclavo sólo descargará las porciones actualizadas de una zona modificada de un servidor DNS tipo maestro.

IXFR solamente se utiliza cuando hay actualizaciones dinámicas para realizar los cambios en los registros de una zona maestra. En cambio si se modifican manualmente los archivos de zona se tendría que usar AXFR (*Automatic Zone Transfer*).

En dominios con muchas consultas o con archivos de zona muy extensos y con muchos servidores DNS tipo esclavo, IXFR hace que la notificación y los procesos de actualización sean menos exigentes en recursos.

**b) Actualización Dinámica (*Dynamic Update*)-** La función de este mecanismo es añadir, sustituir o eliminar registros de un servidor tipo maestro mediante el envío de mensajes DNS especiales.

**c) DNS Dividido (Split DNS)-** La función de este mecanismo es la creación de diferentes puntos de visibilidad del espacio DNS para ocultar información que no se quiera mostrar a los clientes externos en Internet.

(ISC, 2013)

## 4.1.4 Seguridad

BIND soporta un número de métodos diferentes para proteger la actualización y las zonas de transferencia en los servidores de nombres de dominio maestro y esclavo:

**a) DNSSEC (*DNS SECurity*)-** Es un mecanismo que permite firmar con caracteres criptográficos zonas con una clave de zona. De esta manera se puede verificar que la información de una zona provenga de un servidor de nombres que la ha firmado con caracteres criptográficos con una clave privada, siempre y cuando el recipiente tenga esa clave pública del servidor de nombres.

**b) TSIG (*Transaction SIGnatures*)-** Es un mecanismo que permite una transferencia desde el maestro al esclavo sólo después de verificar que una llave secreta compartida existe en ambos servidores maestro y en el esclavo.

**c) SIG (0)-** Es el método de llave pública/privada de autenticación de mensajes donde el control de acceso se realiza de la misma manera que las claves TSIG y los privilegios se pueden otorgar o denegar basándose en la llave.

**d) TKEY-** Es un mecanismo para la generación automática de claves secretas compartidas. BIND 9 implementa el intercambio de claves Diffie-Hellman. El mecanismo TKEY debe utilizar los mensajes firmados por TSIG o SIG (0).

(ISC, 2013)

### 4.1.5 IPv6

BIND versión 9 puede proporcionar servicios a la resolución de nombres de dominio en ambientes IPv6 a través del uso de registros de zona AAAA o A6.

Si el entorno de red incluye *host*s IPv4 e IPv6, se puede usar el demonio ligero de resolución *lwresd* en todos los clientes de la red. Este demonio es muy eficiente, funciona solamente en caché y además entiende los nuevos registros A6 y DNAME usados bajo IPv6. (ISC, 2013)

## 4.2 *Anycast* en IPv4 e IPv6

*Anycast* es una forma de direccionamiento en la que la información es enrutada al mejor destino desde el punto de vista del tiempo de respuesta que depende de la topología de la red (ver Figura 4.1).

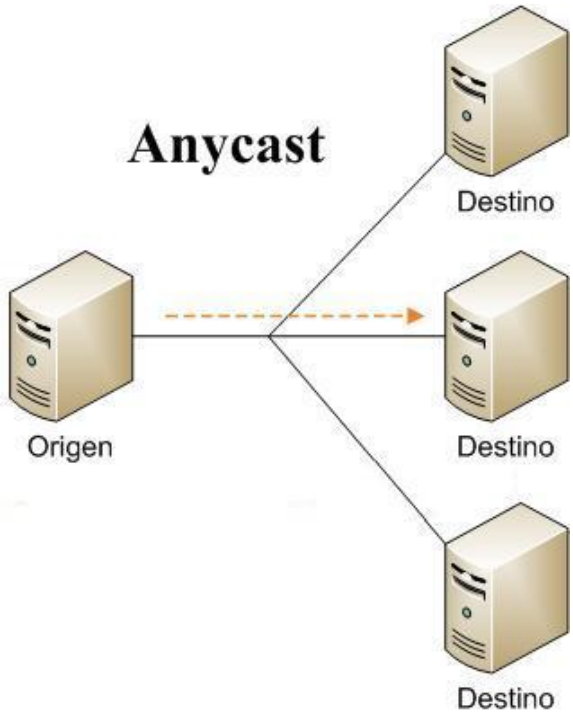


Figura 4.1 *Anycast*

Su similitud con *unicast*, *broadcast* y *multicast*:

- En *unicast*, cada dirección destino se corresponde con un único destino.
- En *broadcast* y *multicast* se asocia una dirección destino a muchos destinos finales.
- En *anycast* también hay una asociación de una dirección destino a varias máquinas. La diferencia está en que se selecciona una de estas máquinas para ser la destinataria de la información.

El direccionamiento tipo *Anycast* es muy común que se use con los protocolos no orientados a la conexión para dar una alta disponibilidad y balanceo de carga, ya que los protocolos orientados a la conexión necesitan mantener información del estado de la comunicación.

Para mejorar la parte de los DNS algunos servidores raíz están distribuidos geográficamente (ver Figura 4.2) para repartir la carga y evitar que la caída de un servidor afecte en gran cantidad a la navegación por Internet. Hay servidores DNS como el C, F, I, J, K y M que se encuentran replicados en diferentes ciudades de continentes diferentes y usan el esquema de *anycast* para proporcionar un servicio descentralizado.

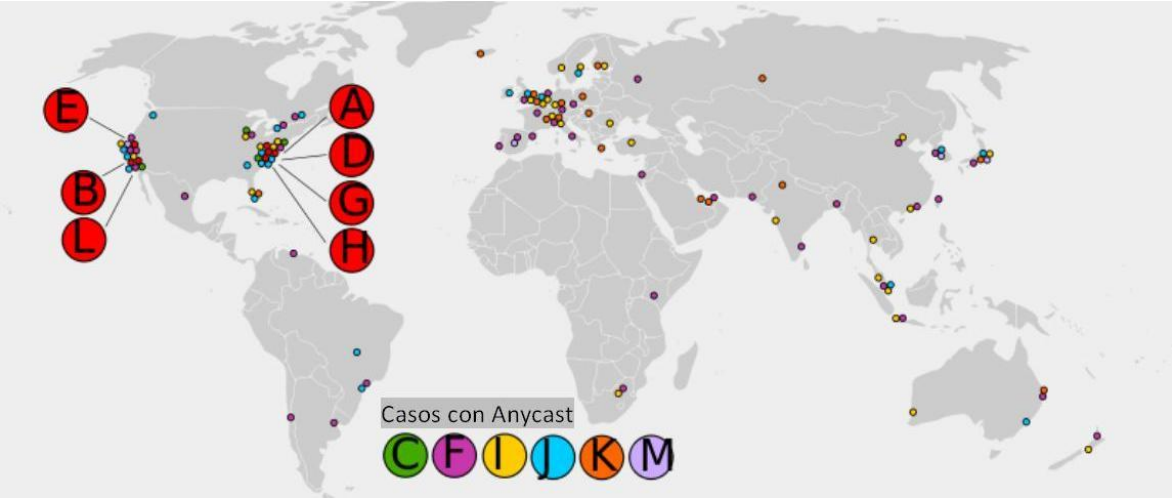


*Figura 4.2 Distribución de servidores DNS raíz*[12]

Así mismo la UNAM utiliza un tipo de distribución parecida (A nivel campus) para garantizar la disponibilidad del servicio de resolución de nombres de dominio.

También hay que reconocer que IPv6 no resuelve ni solucionará todos los problemas de seguridad que ya afectan a IPv4, pero sí ayuda a minimizar muchos de sus efectos y a evitar otros al combinar buenas prácticas y las mejoras en su funcionalidad en los aspectos de seguridad identificados. También es importante destacar que la mayoría de los 13 denominados servidores de raíz (*root servers*) soportan IPv6. (InterNIC, 2013)

[12] *Imagen sacada de http://www.securitynull.net/rootserver-los-13-servidores-raiz-del-mundo/*

Una de las principales razones para el uso de *anycast* en los servidores DNS es que ayuda a contener un ataque distribuido de denegación de servicio, dado que el tráfico es enrutado al nodo más cercano, se distribuye el ataque entre los servidores cercanos, lo cual afectaría sólo a una parte de la red y todavía se tendría el servicio para otros usuarios.

## 4.3 Instalación y configuración de OpenBSD, BIND y Quagga

En la UNAM los servidores de nombres de dominio utilizan el sistema operativo OpenBSD que es una opción muy viable porque es un sistema operativo libre tipo Unix, basado en BSD 4.4. Es un descendiente de NetBSD, con un enfoque especial en la seguridad y la criptografía.

Este sistema operativo se concentra en la portabilidad, cumplimiento de normas y regulaciones, corrección, seguridad y criptografía. OpenBSD está disponible para un gran número de arquitecturas. (OpenBSD, 1996)

En los servidores DNS de RedUNAM se manejan el sistema operativo OpenBSD porque es una opción fiable, segura, estable, tiene una buena integración con BIND y con el paquete de Quagga.

## 4.3.1 Instalación de OpenBSD 5.1

Para la instalación de OpenBSD 5.1 se crean un floppy de instalación y un CD de instalación, las dos opciones se muestran en la siguiente tabla:

*Tabla 4.1 Opciones de instalación*

| Para crear un Floppy de instalación | Para crear un CD de instalación |
|---|---|
| Se descarga el archivo floppy51.fs de: ftp://ftp.openbsd.org.ar/pub/OpenBSD/5.1/i386/<br><br>y en Windows se ejecuta el siguiente comando:<br><br>C:\fdimage –q floppy51.fs a:<br><br>*Nota: Si no reconoce el comando fdimage se tendrá que instalar fdimage.exe desde:<br><br>ftp://ftp.openbsd.org.ar/pub/OpenBSD/5.1/tools/ | Se descarga el archivo install51.iso de:<br><br>ftp://ftp.openbsd.org.ar/pub/OpenBSD/5.1/i386/<br><br>Y se graba en un CD con algún programa que permita hacerlo. |

Cuando comienza la instalación de OpenBSD 5.1 nos muestra una serie de opciones (ver Figura 4.3).


*Figura 4.3 Opciones de OpenBSD*

Para comenzar la instalación se seleccionó la opción (i ó I) (ver Figura 4.4).


*Figura 4.4 Instalación de OpenBSD*

Después se escogió la configuración del teclado, en este caso se seleccionó la opción por defecto (ver Figura 4.5).

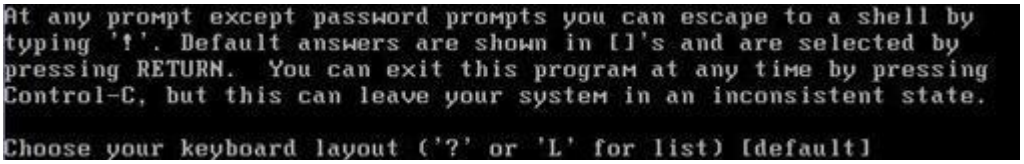


*Figura 4.5 Configuración del teclado*

Luego se seleccionó el nombre del equipo para poder identificarlo, donde el servidor primario se nombro "pruebas1" y el servidor secundario se nombro "pruebas2" (ver Figura 4.6).


*Figura 4.6 Identificación del servidor*

Siguiendo con la instalación, OpenBSD reconoce las interfaces de red que se pueden configurar (ver Figura 4.7).

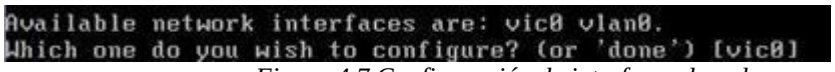


*Figura 4.7 Configuración de interfaces de red*

Luego se pide la dirección IPv4 que se tendrá en la red en la cual trabaje o bien se podrá configurar con DHCP (ver Figura 4.8), en este caso se configuró una dirección IPv4 del segmento de red correspondiente al NIC-UNAM para poder descargar algunos archivos que más adelante se necesitan.

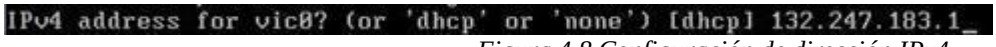


*Figura 4.8 Configuración de dirección IPv4*

Si se desea también se puede configurar la dirección en IPv6 (ver Figura 4.9), no se configuró una dirección IPv6 ya que más adelante dentro de los archivos propios de OpenBSD se configuró.


*Figura 4.9 Configuración de dirección IPv6*

Después de configurar la tarjeta de red podemos seguir configurando las demás que detecte el equipo o bien seguir con la instalación, se seleccionó la opción *done* para seguir con la instalación.

Luego se configuró el servidor DNS para el equipo (ver Figura 4.10), éste se cambia después en la configuración para que el mismo equipo funcione como DNS.


*Figura 4.10 Configuración del servidor DNS*

Después  se puede elegir si se quiere hacer la configuración de red manualmente (ver Figura 4.11).


*Figura 4.11 Configuración de red*

Ahora sólo se escribe *exit* para continuar con la instalación.

Siguiendo con la instalación se pide una contraseña para el usuario *root* (ver Figura 4.12):


*Figura 4.12 Configuración de contraseña*

Y enseguida se escribe de nuevo la contraseña como confirmación.

El demonio de ssh lo ocupamos para administrar remotamente los servidores DNS por lo cual se selecciona la opción por defecto para que se ejecute el demonio de ssh (ver Figura 4.13).


*Figura 4.13 Demonio sshd*

El demonio de ntp no será necesario en este momento (ver Figura 4.14), ya que terminando la instalación se descarga una versión de ntp más actual y se sincronizan los relojes de los servidores DNS.


*Figura 4.14 Demonio ntpd*

La interfaz gráfica no la usaremos, ya que es más común administrar este tipo de servidores de forma remota por medio de ssh, pero por si es necesaria en un futuro dejaremos esto por default (ver Figura 4.15).


*Figura 4.15 Interfaz gráfica*

Como no utilizaremos la interfaz gráfica no será necesario iniciarla (ver Figura 4.16).


*Figura 4.16 Inicio de interfaz gráfica*

Tampoco se cambia el valor predeterminado de la consola (ver Figura 4.17).


*Figura 4.17 Configuración de la consola*

Y no creamos algún usuario adicional, sólo el usuario *root* (ver Figura 4.18), en el caso que se requiera poner en producción un servidor de este tipo es recomendable crear ciertas medidas de seguridad como usuarios con menos privilegios y usar listas de acceso:


*Figura 4.18 Creación de usuario adicional*

En la zona horaria la dejamos por defecto.

El sistema OpenBSD detecta los medios de almacenamiento que se tienen disponibles para la instalación y se muestran las particiones por defecto (ver Figura 4.19).

```
Available disks are: sd0.
Which one is the root disk? (or 'done') [sd0]
Use DUIDs rather than device names in fstab? [yes]
MBR has invalid signature; not showing it.
Use (W)hole disk or (E)dit the MBR? [whole]
Setting OpenBSD MBR partition to whole sd0...done.
The auto-allocated layout for sd0 is:
#                size           offset  fstype [fsize bsize  cpg]
  a:           131.1M               64  4.2BSD   2048 16384    1 # /
  b:           131.1M           268480    swap
  c:          8192.0M                0  unused
  d:           201.7M           536928  4.2BSD   2048 16384    1 # /tmp
  e:           212.8M           949984  4.2BSD   2048 16384    1 # /var
  f:           951.1M          1385728  4.2BSD   2048 16384    1 # /usr
  g:           542.6M          3333504  4.2BSD   2048 16384    1 # /usr/X11R6
  h:          2150.1M          4444800  4.2BSD   2048 16384    1 # /usr/local
  i:          1044.4M          8848256  4.2BSD   2048 16384    1 # /usr/src
  j:          1340.8M         10987232  4.2BSD   2048 16384    1 # /usr/obj
  k:          1483.6M         13733280  4.2BSD   2048 16384    1 # /home
Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout? [a] _
```

*Figura 4.19 Particiones por defecto*

Como se requiere una instalación más personalizada se selecciona la opción (e) para reparticionar el espacio de almacenamiento, si se elige la opción (?) se muestran las opciones que se pueden utilizar para particionar el disco duro y las opciones para poner el tamaño de cada partición (ver Figura 4.20).

```
Use (A)uto layout, (E)dit auto layout, or create (C)ustom layout? [a] e
Label editor (enter '?' for help at any prompt)
> ?
Available commands:
 ? ; h     - show help                 n [part] - set mount point
 A         - auto partition all space  p [unit] - print partitions
 a [part]  - add partition             q        - quit & save changes
 b         - set OpenBSD boundaries    R [part] - resize auto allocated partition
 c [part]  - change partition size     r        - display free space
 D         - reset label to default    s [path] - save label to file
 d [part]  - delete partition          U        - undo all changes
 e         - edit drive parameters     u        - undo last change
 g [d;u]   - [d]isk or [u]ser geometry w        - write label to disk
 i         - modify disklabel UID      X        - toggle expert mode
 l [unit]  - print disk label header   x        - exit & lose changes
 M         - disklabel(8) man page     z        - delete all partitions
 m [part]  - modify partition

Suffixes can be used to indicate units other than sectors:
 'b' (bytes), 'k' (kilobytes), 'm' (megabytes), 'g' (gigabytes) 't' (terabytes)
 'c' (cylinders), '%' (% of total disk), '&' (% of free space).
Values in non-sector units are truncated to the nearest cylinder boundary.
> _
```

*Figura 4.20 Opciones para particionar el almacenamiento*

Como no queremos el particionamiento por defecto, se utilizó la opción (z) para borrar todas las particiones y con la opción (p) se muestra la única partición que no se puede modificar y la cual contiene todo el espacio disponible (ver Figura 4.21).

```
> z
> p
OpenBSD area: 64-16771860; size: 16771796; free: 16771796
#                size           offset  fstype [fsize bsize  cpg]
  c:         16777216                0  unused
```

*Figura 4.21 Espacio total disponible*

Las particiones que se crearon fueron la raíz (/), swap, /usr, /home, /var, /tmp (ver Figura 4.22) con una distribución del espacio parecida a la siguiente tabla:

*Tabla 4.2 Distribución de espacio*

| Partición | % |
|---|---|
| Raíz (/) | 12.5% |
| Swap | 1.25% |
| /usr | 30.5% |
| /home | 12.5% |
| /var | 12.5% |
| /tmp | 30.75% |

```
OpenBSD area: 64-16771860; size: 16771796; free: 37
#                size           offset  fstype [fsize bsize  cpg]
  a:          2104448               64  4.2BSD   2048 16384    1 # /
  b:           192783          2104512    swap
  c:         16777216                0  unused
  d:          4208992          2297312  4.2BSD   2048 16384    1 # /usr
  e:          2088448          6506304  4.2BSD   2048 16384    1 # /home
  f:          2088448          8594752  4.2BSD   2048 16384    1 # /var
  g:          6088640         10683200  4.2BSD   2048 16384    1 # /tmp
> _
```

*Figura 4.22 Distribución de espacio*

Después de crear las particiones se selecciona la opción (q) para salvar los cambios y salir.

Luego se pide una confirmación para empezar a crear las particiones necesarias para el correcto funcionamiento de los servidores (ver Figura 4.23).

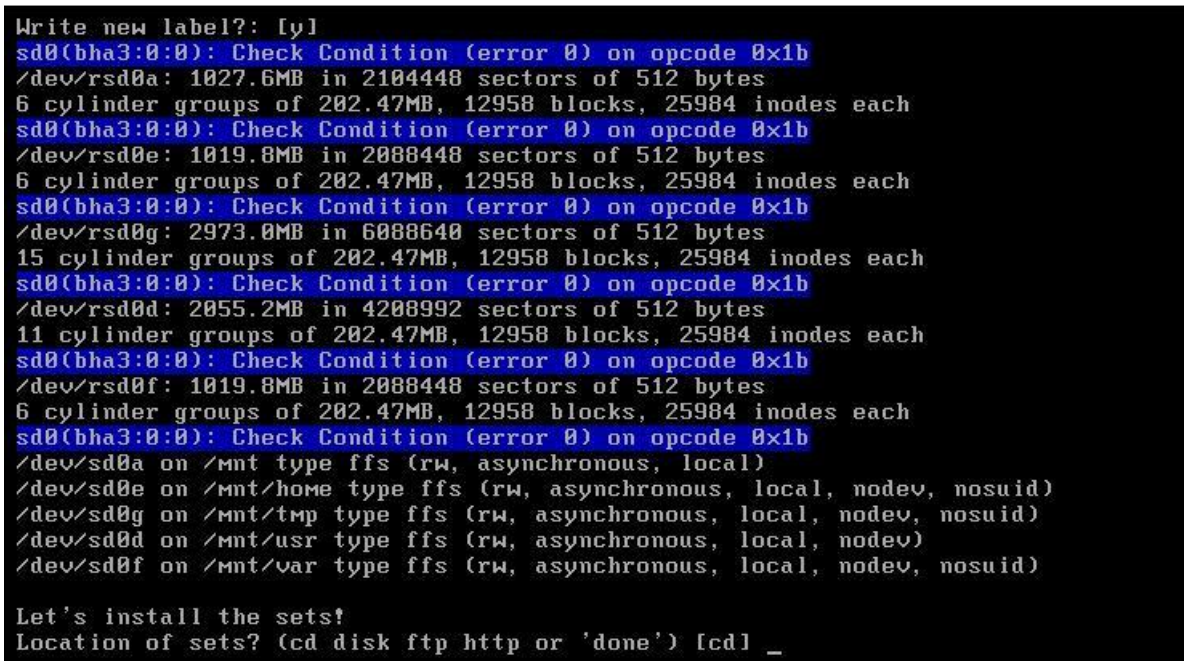


*Figura 4.23 Creación de particiones*

Ya creadas las particiones se instalan los conjuntos necesarios, lo cual se hace mediante ftp, en la parte de *proxy* se deja con la opción por defecto, el servidor de descarga se puede escoger de la lista dada por OpenBSD, el directorio donde se encuentran los archivos se dejó por defecto y accedemos con el usuario por defecto y se muestra una lista de los archivos que se instalarán a lo que dejamos seleccionados todos y se seleccionó la opción por defecto (ver Figura 4.24).

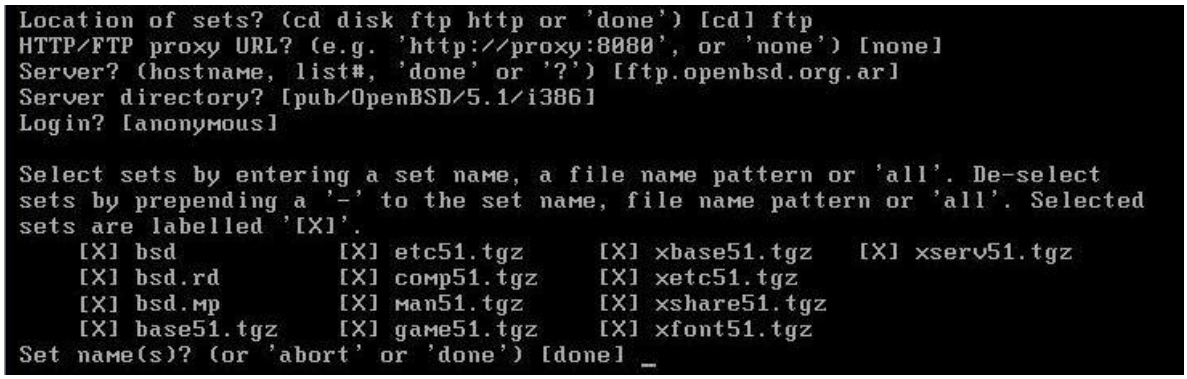


*Figura 4.24 Conjuntos de OpenBSD*

Después de que se descargan e instalan los archivos se pueden agregar otros pero con estos es suficiente ya que para el funcionamiento correcto del servidor DNS solo necesitaremos los archivos básicos, así que se seleccionó la opción por defecto (ver Figura 4.25).


*Figura 4.25 Conjuntos instalados*

Se configuró la hora, día y año que en este caso lo dejamos por defecto, y después se pide reiniciar el equipo para completar la instalación (ver Figura 4.26).

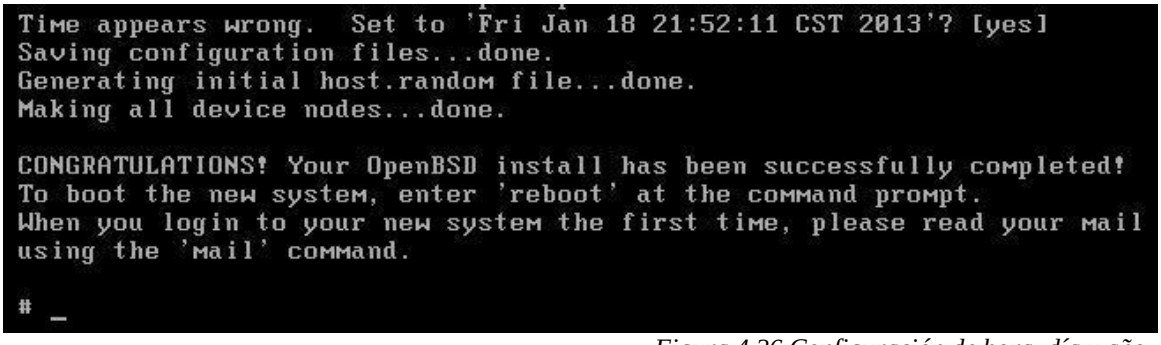


*Figura 4.26 Configuración de hora, día y año*

Ya reiniciado el equipo se puede acceder con el usuario root y la contraseña escogida anteriormente (ver Figura 4.27).

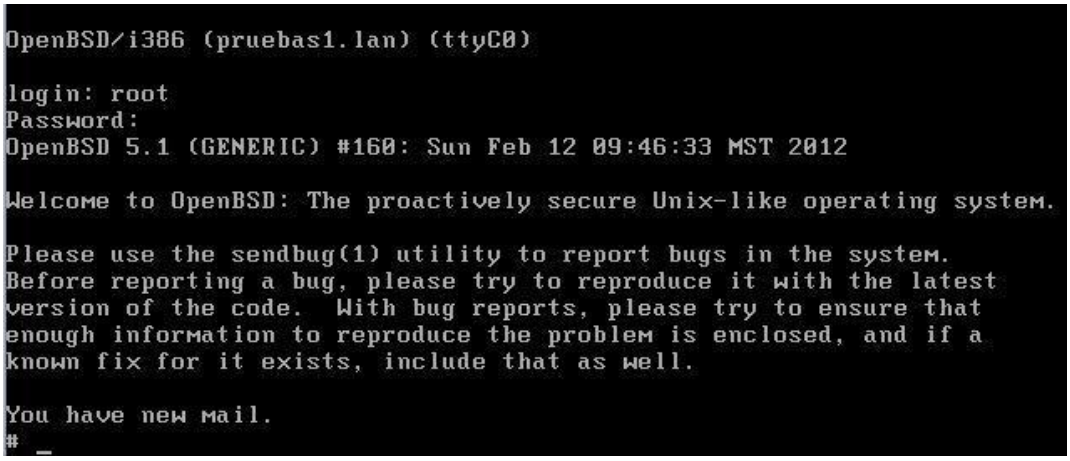


*Figura 4.27 Acceso a OpenBSD*

Luego de terminar la instalación se descargan los archivos "*sys.tar.gz*" que contiene los archivos que se utilizan para crear el *kernel*, y "*src.tar.gz*" que contiene todos los archivos básicos excepto el árbol de puertos y las fuentes X11, porque han sufrido algunos cambios entre el OpenBSD que se instala y en la que está basada.

Se descargan de la página ftp://ftp.openbsd.org/pub/openbsd/5.1/ y se colocan en la ruta "*/usr/src*", con los archivos en este directorio se ocupa el comando "*gunzip*" para descomprimir cada uno de los archivos (ver Figura 4.28).

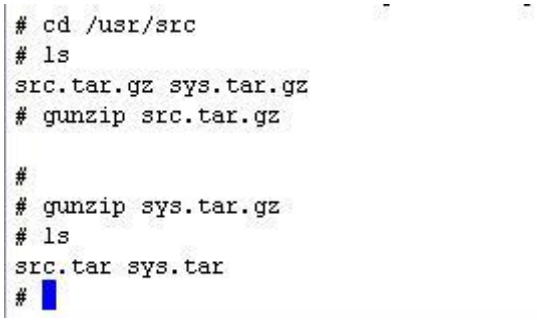


*Figura 4.28 Comando gunzip*

Después se ocupó el comando "*tar*" para extraer cada uno de los archivos (ver Figura 4.29)

```
# tar -xvf src.tar
./usr.sbin/nginx/src/pcre/pcre_exec.c
./usr.sbin/nginx/src/pcre/pcre_fullinfo.c
./usr.sbin/nginx/src/pcre/pcre_globals.c
./usr.sbin/nginx/src/pcre/pcre_internal.h
./usr.sbin/nginx/src/pcre/pcre_newline.c
./usr.sbin/nginx/src/pcre/pcre_ord2utf8.c
./usr.sbin/nginx/src/pcre/pcre_tables.c
./usr.sbin/nginx/src/pcre/pcre_try_flipped.c
./usr.sbin/nginx/src/pcre/pcre_ucd.c
./usr.sbin/nginx/src/pcre/pcre_valid_utf8.c
./usr.sbin/nginx/src/pcre/pcre_xclass.c
./usr.sbin/nginx/src/pcre/ucp.h
./usr.sbin/npppctl
./usr.sbin/npppctl/CVS
./usr.sbin/npppctl/CVS/Repository
./usr.sbin/npppctl/CVS/Entries
./usr.sbin/npppctl/Makefile
./usr.sbin/npppctl/npppctl.8
./usr.sbin/npppctl/npppctl.c
./usr.sbin/npppctl/parser.c
./usr.sbin/npppctl/parser.h
#
# tar -xvf sys.tar
tar: Unable to create ./sys/nnpfs/nnpfs_syscalls.h: No such file or directory
./sys/nnpfs/nnpfs_vfsops-bsd.c
tar: Unable to create ./sys/nnpfs/nnpfs_vfsops-bsd.c: No such file or directory
./sys/nnpfs/nnpfs_vfsops-bsd.h
tar: Unable to create ./sys/nnpfs/nnpfs_vfsops-bsd.h: No such file or directory
./sys/nnpfs/nnpfs_vfsops-common.c
tar: Unable to create ./sys/nnpfs/nnpfs_vfsops-common.c: No such file or directo
ry
./sys/nnpfs/nnpfs_vfsops-openbsd.c
tar: Unable to create ./sys/nnpfs/nnpfs_vfsops-openbsd.c: No such file or direct
ory
./sys/nnpfs/nnpfs_vfsops.h
tar: Unable to create ./sys/nnpfs/nnpfs_vfsops.h: No such file or directory
./sys/nnpfs/nnpfs_vnodeops-bsd.c
tar: Unable to create ./sys/nnpfs/nnpfs_vnodeops-bsd.c: No such file or director
y
./sys/nnpfs/nnpfs_vnodeops-common.c
tar: Unable to create ./sys/nnpfs/nnpfs_vnodeops-common.c: No such file or direc
tory
./sys/nnpfs/nnpfs_vnodeops.h
tar: Unable to create ./sys/nnpfs/nnpfs_vnodeops.h: No such file or directory
./sys/nnpfs/nnpfs_vopdefs.h
tar: Unable to create ./sys/nnpfs/nnpfs_vopdefs.h: No such file or directory
#
```

*Figura 4.29 Comando tar*

Ahora ya con estos archivos descomprimidos, descargamos el archivo de ntp-4.2.6pl2p7.tgz de ftp://ftp.openbsd.org/pub/openbsd/5.1/packages/i386 pudiendo utilizar los siguientes comandos (ver Figura 4.30):

```
pruebas1# setenv PKG_path ftp://ftp.openbsd.org/pub/openbsd/5.1/packages/i386
pruebas1# pkg_add ${PKG_path} ntp-4.2.6pl2p7.tgz
```

*Figura 4.30 Archivo ntp*

Ya que se descargó se mueve el archivo a la ruta *"/etc/rc.d/xntpd"* y la documentación extra se encuentra en la ruta *"/usr/local/share/doc/pkg-readmes"*.

Después se ejecutaron algunos comandos para sincronizar el reloj del equipo con el equipo llamado "tiempo.nic.mx" (ver Figura 4.31).

```
# /usr/local/sbin/ntpdate -s -b tiempo.nic.mx
# /usr/sbin/ntpd
# ps aux|grep ntpd
_ntp      30515  0.0  0.1   728   960 ??  S     12:28AM    0:00.00 ntpd: dns eng
root      27555  0.0  0.1   508   760 ??  Ss    12:28AM    0:00.00 ntpd: [priv]
_ntp      31079  0.0  0.1   592  1024 ??  S     12:28AM    0:00.00 ntpd: ntp eng
# date
Sat Jan 19 00:29:18 CST 2013
#
```

*Figura 4.31 Sincronización de reloj*

## 4.3.2 Configuración de BIND

Para la configuración del servidor de Nombres de Dominio es recomendable preferentemente configurar la máquina para trabajar de manera dedicada al servicio de nombres de dominio para que éste funcione de manera óptima ya que si se tienen más servicios en el mismo servidor éstos pueden llegar a fallar y así interferir en el funcionamiento del servidor.

## 4.3.2.1 Configuración del Servidor

Para que el servidor funcione como servidor de nombres de dominio se modifican varios archivos dentro del mismo. Los archivos que se modificaron fueron para cambiar la dirección IP, el Gateway, así como el nombre del equipo y su dominio.

En la ruta ***/etc*** podemos encontrar los archivos que necesitamos configurar para que nuestro servidor funcione como DNS.

Primero modificamos el archivo *"hosts"* (ver Figura 4.32), para modificar la dirección IP y nombre del equipo, en el caso de que se requiera cambiar el nombre del servidor para que sea más fácil reconocerlo o bien si se requiere cambiar la dirección IPv4 o IPv6 después de la instalación:

```
adduser.conf    ftpusers        login.conf      passwd          sasyncd.conf
afs             gettytab        lynx.cfg        pf.conf         sensorsd.conf
amd             group           magic           pf.os           services
authpf          group.bak       mail            pkg.conf        shells
bgpd.conf       hostapd.conf    mail.rc         ppp             skel
changelist      hostname.em0    mailer.conf     printcap        sliphome
chio.conf       hosts           man.conf        protocols       snmpd.conf
csh.cshrc       hosts.equiv     master.passwd   pwd.db          spwd.db
csh.login       hosts.lpd       mixerctl.conf   quagga          ssh
csh.logout      hotplug         moduli          rbootd.conf     ssl
daily           ifstated.conf   monthly         rc              sudoers
dhclient.conf   iked            motd            rc.conf         sysctl.conf
dhcpd.conf      iked.conf       mrouted.conf    rc.d            syslog.conf
disklabels      inetd.conf      mtree           rc.local        systrace
disktab         ipsec.conf      mygate          rc.securelevel  termcap
dumpdates       isakmpd         myname          rc.shutdown     ttys
dvmrpd.conf     kerberosV       netstart        relayd.conf     usermgmt.conf
exports         ksh.kshrc       networks        remote          weekly
fbtab           ldap            newsyslog.conf  resolv.conf     wsconsctl.conf
firmware        ldapd.conf      nsd.conf        ripd.conf       ypldap.conf
fonts           ldpd.conf       ntpd.conf       rmt
# pwd
/etc
#
```

*Figura 4.32 Archivo hosts*

En el archivo *"hosts"* del servidor "pruebas1" se agregaron las siguientes líneas:

*# Host Database*

# Se especifica la dirección de *loopback* en IPv4

***127.0.0.1 localhost***

#Se especifica la dirección de *loopback* en IPv6

***::1 localhost***

#Se especifica la dirección IPv4 que va a utilizar el servidor, su dominio y el nombre del #servidor

***132.247.183.1 pruebas1.nic.unam.mx pruebas1***

#Se especifica la dirección IPv6 que va a utilizar el servidor, su dominio y el nombre del #servidor

***2001:1218:0101:0183::2 pruebas1.nic.unam.mx pruebas1***

En el archivo *"hosts"* del servidor "pruebas2" se agregaron las siguientes líneas:

*# Host Database*

# Se especifica la dirección de *loopback* en IPv4

***127.0.0.1 localhost***

#Se especifica la dirección de *loopback* en IPv6

***::1 localhost***

#Se especifica la dirección IPv4 que va a utilizar el servidor, su dominio y el nombre del #servidor

***132.247.183.9 pruebas2.nic.unam.mx pruebas2***

#Se especifica la dirección IPv6 que va a utilizar el servidor, su dominio y el nombre del #servidor

***2001:1218:0101:0184::2 pruebas2.nic.unam.mx pruebas2***

Luego de modificar el archivo *"hosts"* se modifican el archivo *"hostname.em0"* (ver Figura 4.33), para cambiar la configuración del adaptador de red. Para el servidor pruebas1 *"hostname.em0"* y *"hostname.xl0"* para el servidor pruebas2:

```
adduser.conf    ftpusers        login.conf      passwd          sasyncd.conf
afs             gettytab        lynx.cfg        pf.conf         sensorsd.conf
amd             group           magic           pf.os           services
authpf          group.bak       mail            pkg.conf        shells
bgpd.conf       hostapd.conf    mail.rc         ppp             skel
changelist      hostname.em0    mailer.conf     printcap        sliphome
chio.conf       hosts           man.conf        protocols       snmpd.conf
csh.cshrc       hosts.equiv     master.passwd   pwd.db          spwd.db
csh.login       hosts.lpd       mixerctl.conf   quagga          ssh
csh.logout      hotplug         moduli          rbootd.conf     ssl
daily           ifstated.conf   monthly         rc              sudoers
dhclient.conf   iked            motd            rc.conf         sysctl.conf
dhcpd.conf      iked.conf       mrouted.conf    rc.d            syslog.conf
disklabels      inetd.conf      mtree           rc.local        systrace
disktab         ipsec.conf      mygate          rc.securelevel  termcap
dumpdates       isakmpd         myname          rc.shutdown     ttys
dvmrpd.conf     kerberosV       netstart        relayd.conf     usermgmt.conf
exports         ksh.kshrc       networks        remote          weekly
fbtab           ldap            newsyslog.conf  resolv.conf     wsconsctl.conf
firmware        ldapd.conf      nsd.conf        ripd.conf       ypldap.conf
fonts           ldpd.conf       ntpd.conf       rmt
# pwd
/etc
#
```

*Figura 4.33 Archivo hostname.em0*

En el archivo *"hostname.em0"* del servidor "pruebas1" se agregaron las siguientes líneas:

#Indicamos la dirección IPv4 y su máscara de red

*inet 132.247.183.1 255.255.255.248*

#Indicamos la dirección IPv6

*inet6 2001:1218:0101:0183::2 64*

En el archivo *"hostname.xl0"* del servidor "pruebas2" se agregaron las siguientes líneas:

#Indicamos la dirección IPv4 y su máscara de red

*inet 132.247.183.9 255.255.255.248*

#Indicamos la dirección IPv6

*inet6* 2001:1218:0101:0184::2 64

Después se modificó el archivo *"mygate"* (ver Figura 4.34) para cambiar el *Gateway* del equipo y así poder tener acceso a la red exterior:

```
adduser.conf    ftpusers        login.conf      passwd          sasyncd.conf
afs             gettytab        lynx.cfg        pf.conf         sensorsd.conf
amd             group           magic           pf.os           services
authpf          group.bak       mail            pkg.conf        shells
bgpd.conf       hostapd.conf    mail.rc         ppp             skel
changelist      hostname.em0    mailer.conf     printcap        sliphome
chio.conf       hosts           man.conf        protocols       snmpd.conf
csh.cshrc       hosts.equiv     master.passwd   pwd.db          spwd.db
csh.login       hosts.lpd       mixerctl.conf   quagga          ssh
csh.logout      hotplug         moduli          rbootd.conf     ssl
daily           ifstated.conf   monthly         rc              sudoers
dhclient.conf   iked            motd            rc.conf         sysctl.conf
dhcpd.conf      iked.conf       mrouted.conf    rc.d            syslog.conf
disklabels      inetd.conf      mtree           rc.local        systrace
disktab         ipsec.conf      mygate          rc.securelevel  termcap
dumpdates       isakmpd         myname          rc.shutdown     ttys
dvmrpd.conf     kerberosV       netstart        relayd.conf     usermgmt.conf
exports         ksh.kshrc       networks        remote          weekly
fbtab           ldap            newsyslog.conf  resolv.conf     wsconsctl.conf
firmware        ldapd.conf      nsd.conf        ripd.conf       ypldap.conf
fonts           ldpd.conf       ntpd.conf       rmt
# pwd
/etc
#
```

*Figura 4.34 Archivo mygate*

En el archivo *"mygate"* del servidor "pruebas1" se agregaron las siguientes líneas:

#Indicamos la dirección del *Gateway* para IPv4

***132.247.183.6***

#Indicamos la dirección del *Gateway* para IPv6

***2001:1218:0101:0183::1***

En el archivo *"mygate"* del servidor "pruebas2" se agregaron las siguientes líneas:

#Indicamos la dirección del *Gateway* para IPv4

***132.247.183.14***

#Indicamos la dirección del *Gateway* para IPv6

***2001:1218:0101:0184::1***

Luego se modificó el archivo *"myname"* (ver Figura 4.35) para cambiar los nombres de los servidores y poderlos identificar con mayor facilidad (Si desde la instalación se escogieron los nombres definitivos de los servidores este paso no será necesario):

```
adduser.conf    ftpusers        login.conf      passwd          sasyncd.conf
afs             gettytab        lynx.cfg        pf.conf         sensorsd.conf
amd             group           magic           pf.os           services
authpf          group.bak       mail            pkg.conf        shells
bgpd.conf       hostapd.conf    mail.rc         ppp             skel
changelist      hostname.em0    mailer.conf     printcap        sliphome
chio.conf       hosts           man.conf        protocols       snmpd.conf
csh.cshrc       hosts.equiv     master.passwd   pwd.db          spwd.db
csh.login       hosts.lpd       mixerctl.conf   quagga          ssh
csh.logout      hotplug         moduli          rbootd.conf     ssl
daily           ifstated.conf   monthly         rc              sudoers
dhclient.conf   iked            motd            rc.conf         sysctl.conf
dhcpd.conf      iked.conf       mrouted.conf    rc.d            syslog.conf
disklabels      inetd.conf      mtree           rc.local        systrace
disktab         ipsec.conf      mygate          rc.securelevel  termcap
dumpdates       isakmpd         myname          rc.shutdown     ttys
dvmrpd.conf     kerberosV       netstart        relayd.conf     usermgmt.conf
exports         ksh.kshrc       networks        remote          weekly
fbtab           ldap            newsyslog.conf  resolv.conf     wsconsctl.conf
firmware        ldapd.conf      nsd.conf        ripd.conf       ypldap.conf
fonts           ldpd.conf       ntpd.conf       rmt
# pwd
/etc
#
```

*Figura 4.35 Archivo myname*

En el archivo *"myname"* del servidor "pruebas1" se agregó la siguiente línea:

***pruebas1.nic.unam.mx***

En el archivo *"myname"* del servidor "pruebas2" se agregó la siguiente línea:

***pruebas2.nic.unam.mx***

Después de configurados estos archivos se modificó el archivo *"resolv.conf"* (ver Figura 4.36) el cual tiene la dirección IP del DNS al cual se le hacen las peticiones de resoluciones de nombres de dominio, que en este caso será la misma dirección IP del servidor.

```
adduser.conf    ftpusers        login.conf      passwd          sasyncd.conf
afs             gettytab        lynx.cfg        pf.conf         sensorsd.conf
amd             group           magic           pf.os           services
authpf          group.bak       mail            pkg.conf        shells
bgpd.conf       hostapd.conf    mail.rc         ppp             skel
changelist      hostname.em0    mailer.conf     printcap        sliphome
chio.conf       hosts           man.conf        protocols       snmpd.conf
csh.cshrc       hosts.equiv     master.passwd   pwd.db          spwd.db
csh.login       hosts.lpd       mixerctl.conf   quagga          ssh
csh.logout      hotplug         moduli          rbootd.conf     ssl
daily           ifstated.conf   monthly         rc              sudoers
dhclient.conf   iked            motd            rc.conf         sysctl.conf
dhcpd.conf      iked.conf       mrouted.conf    rc.d            syslog.conf
disklabels      inetd.conf      mtree           rc.local        systrace
disktab         ipsec.conf      mygate          rc.securelevel  termcap
dumpdates       isakmpd         myname          rc.shutdown     ttys
dvmrpd.conf     kerberosV       netstart        relayd.conf     usermgmt.conf
exports         ksh.kshrc       networks        remote          weekly
fbtab           ldap            newsyslog.conf  resolv.conf     wsconsctl.conf
firmware        ldapd.conf      nsd.conf        ripd.conf       ypldap.conf
fonts           ldpd.conf       ntpd.conf       rmt
# pwd
/etc
#
```

*Figura 4.36 Archivo resolv.conf*

En el archivo *"resolv.conf"* del servidor "pruebas1" se agregaron las siguientes líneas:

*#lookup file bind*

*domain unam.mx*

#Especificamos la dirección IPv6 del servidor

***nameserver 2001:1218:0101:0183::2***

#Especificamos la dirección IPv4 del servidor

***nameserver 132.247.183.1***

En el archivo *"resolv.conf"* del servidor "pruebas2" se agregaron las siguientes líneas:

*#lookup file bind*

*domain unam.mx*

#Especificamos la dirección IPv6 del servidor

***nameserver 2001:1218:0101:0184::2***

#Especificamos la dirección IPv4 del servidor

***nameserver 132.247.183.9***

Ahora ya con los archivos modificados se escogió como DNS primario o maestro a "pruebas1" y como secundario o esclavo (Un servidor esclavo es simplemente un servidor de nombres que replica los ficheros de las zonas de un maestro) a "pruebas2" con lo que nos dirigimos a la ruta */var/named/etc* del servidor secundario y en el archivo *named.conf,* en la parte de "*options"* se pone lo siguiente:

***Allow-transfer { 132.247.183.1 ; } ;***

Con lo cual permite o autoriza la transferencia de zonas del servidor primario al servidor secundario.

Después de configurar el servidor secundario es necesario utilizar una utilidad de BIND que son las llaves de "rndc", que permiten administrar localmente o a distancia, el demonio *named*. El programa rndc utiliza el archivo *rndc.conf* para las opciones de configuración que serán sobrescritas por las opciones de las líneas de comandos.

Para evitar que los usuarios no autorizados puedan controlar BIND en el sistema, se utiliza el método de claves secretas compartidas para dar privilegios a determinados *hosts.* Para que rndc emita comandos hacia cualquier *named*, incluso hacia la máquina local, las claves utilizadas en los ficheros */etc/named.conf* y /*etc/rndc.conf* tienen que coincidir.

Para utilizar las llaves de rndc se utiliza el siguiente comando:

***# rndc-confgen***

Con el cual se genera una serie de instrucciones las cuales se dividen en dos partes, la primera parte se copia en el archivo "*rndc.conf*" en cual se encuentra en */etc* y si el archivo no se encuentra se tendrá que crear y copiar las instrucciones que se generaron con el comando anterior:

```
# Start of rndc.conf
key "rndc-key" {
        algorithm hmac-md5;
        secret "kQS0vn8ELyAhGVAxx6AJIA==";
};
options {
        default-key "rndc-key";
        default-server 127.0.0.1;
        default-port 953;
};
# End of rndc.conf
```

La segunda parte se copia en el archivo *"named.conf"* después de la sección de *options*:

```
# Use with the following in named.conf, adjusting the allow list as needed:
key "rndc-key" {
     algorithm hmac-md5;
     secret "kQS0vn8ELyAhGVAxx6AJIA==";
 };
 controls {
     inet 127.0.0.1 port 953
           allow { 127.0.0.1; } keys { "rndc-key"; };
 };
# End of named.conf
```

(Estas llaves se pusieron tanto en el servidor primario como en el secundario generando distintas llaves para cada servidor)

Cuando se ejecuta el comando rndc en un *host* local, se encuentran disponibles los siguientes comandos (Con el comando man rndc se puede ver el manual sobre el comando rndc donde se muestra el funcionamiento de las siguientes opciones entre otras):

- *halt* — Detiene inmediatamente el servicio *named*.
- *querylog* — Ejecuta la conexión para todas las peticiones efectuadas por los clientes hacia el servidor de nombres.
- *refresh* — Actualiza la base de datos del servidor de nombres.
- *reload* — Dice al servidor de nombres que recargue los ficheros de zona para que conserve todas las respuestas precedentes situadas en caché. Esto le permite realizar cambios en los ficheros de zona y de ponerlos en práctica en los servidores maestros y esclavos sin perder las resoluciones de nombres almacenadas.
- *reconfig* — Reconfigura y/o fuerza a actualizar los ficheros de zona.

Si los cambios no afectan a una zona determinada, puede decir al proceso de *named* que recargue esa zona. Escriba el nombre de la zona después del comando *reload.*

- *stats* —Pasa las estadísticas del comando *named* al fichero */var/named/named.stats*.

- *stop* — Para el servidor salvando todas las actualizaciones dinámicas y los datos IXFR antes de parar el servidor completamente.

Se pueden sobrescribir los parámetros predeterminados del fichero */etc/rndc.conf*. Existen varias posibilidades:

- -c <*configuration-file*> — Dice al comando rndc que use otro fichero de configuración diferente del fichero predeterminado */etc/rndc.conf*.
- -p <*port-number*> — Especifica un número de puerto diferente del predeterminado (953) para la conexión del comando rndc.
- -s <*server*> — Dice a rndc que envíe comandos a otro servidor distinto del servidor que designa la opción *default-server* en el fichero */etc/rndc.conf*.

Para que se lleve a cabo esta tarea, se tiene que haber configurado el servicio *named* para que acepte los comandos del *host* y que tenga la clave para este servicio de nombres.

- -y <*key-name*> — Le permite especificar una clave distinta de la opción *default-key* en el fichero */etc/rndc.conf*.

Con esto ya tenemos lista la configuración de los equipos que funcionarán como servidores de nombres de dominio.

## 4.3.2.2 Declaración y Configuración de zonas

Los Archivos de Zona (*Zone Files*) sirven para organizan los Registros de Zona (*Zone Records*) para dominios y subdominios en los servidores DNS. Cada dominio y subdominio tiene un archivo de zona y cada archivo de zona contiene registros de zona, los cuales contienen información de los dominios y subdominios los cuales tienen un enlace hacia una dirección IP, un nombre canónico, un puntero entre otros .

Los archivos de zona contienen directivas y registros de recursos. Las directivas contienen información que le indican al servidor de nombres que realice una determinada acción o que aplique una configuración a la zona. Los registros de recursos definen parámetros de una zona en particular que asigna una identidad a los dominios y subdominios.

Dentro de las directivas que contienen los archivos de zona podemos encontrar las siguientes:

**a) $ORIGIN-** Anexa el nombre del dominio a registros no cualificados, tales como aquellos con el nombre de *host*. El símbolo **@** coloca la directiva $ORIGIN (o el nombre de la zona, si la directiva $ORIGIN no está configurada) como el espacio de nombres que está siendo definido.

**b) $TTL *(Time to Live)* -** Ajusta el valor predeterminado para la zona. Es el tiempo, en segundos, que un registro de recurso de zona es válido. Cada recurso puede contener su propio valor TTL. Cuando se decide aumentar este valor, permite a los servidores de nombres remotos hacer caché a la información de zona para un período más largo de

tiempo, reduciendo el número de consultas para la zona y alargando la cantidad de tiempo requerido para proliferar cambios de registros de recursos.

**c) <*primary-name-server*> -** Es el nombre del *host* del servidor de nombres que tiene autoridad para el dominio.

**d) <*hostmaster-email*>-** Es el correo electrónico de la persona a contactar sobre el espacio de nombres.

**e) <*serial-number*> -** Es un valor numérico que es incrementado cada vez que se cambia el archivo de zona para así indicar al proceso de *named* la recargar de la zona.

**f) <*time-to-refresh*>-** Es el valor numérico que los servidores esclavos utilizan para determinar cuánto tiempo debe esperar antes de preguntar al servidor de nombres maestro si se han realizado cambios a la zona.

**g) <*time-to-retry*>-** Es un valor numérico usado por los servidores esclavo para determinar el intervalo de tiempo que tiene que esperar antes de emitir una petición de actualización de datos en caso de que el servidor de nombres maestro no responda.

**h) <*time-to-expire*>-** Es un valor numérico usado por los servidores. Si el servidor maestro no ha respondido a una petición de actualización de datos antes que se acabe el intervalo de tiempo, los servidores esclavo paran de responder como una autoridad para peticiones relacionadas a ese espacio de nombres.

**i) <*minimum-TTL*>-** Es la cantidad de tiempo que otros servidores de nombres guardan en caché la información de la zona.

Dentro de los registros de recursos que se usan con más frecuencia que podemos encontrar en un archivo de zona son:

**a) SOA (*Start of Authority*)-** El registro contiene información como la dirección email del administrador de la zona, el nombre de servidor maestro para la zona y un número que es incrementado cada vez que el archivo de zona es actualizado.

**b) NS (*Name Server*)-** El registro contiene la información del servidor de nombres para la zona.

**c) MX (*Mail eXchanger*)-** El registro contiene información del servidor de correo electrónico para esa zona. Esto permitirá que el correo electrónico sea enviado al lugar correcto.

**d) A (*Address*)-** El registro define una dirección IPv4 asignada a un nombre de *host*. Generalmente existen varios en un dominio. Éste es el tipo más común de registro en Internet.

**e) CNAME (*Canonical Name*)-** El registro contiene un alias para un *host*. Los CNAME permiten tener más de un nombre DNS para un *host*. Los registros CNAME apuntan de regreso hacia un registro A. Si se cambia la dirección IP en el registro A, todo registro CNAME seguirá automáticamente a la nueva IP del registro A.

**f) TXT (*Text*)-** El registro contiene información adicional sobre un *host*. O se puede usar para proveer información técnica a servidores.

**g) SRV (*Service Records*)-** El registro contiene información para identificar servicios específicos.

**h) AAAA**- El registro define una dirección IPv6 asignada a un *host*. Generalmente existen varios en un dominio.

**j) PTR (*PoinTeR*)-** Se utiliza para apuntar a otra parte del espacio de nombres. Los registros PTR son usados principalmente para la resolución inversa de nombres, pues ellos apuntan direcciones IP de vuelta a un nombre particular.

Aunque éstos no son los únicos registros existentes, son los que más se usan.
(RFC1034, 1987) (RFC1035, 1987) (RFC2181, 1997) (RFC3596, 2003)

## 4.3.2.2.1 Zona Directa

Para la declaración de las zonas directas modificamos el archivo "*named*.conf" en donde se pueden separar en tres tipos de zonas, las *Standard Zones*, las *Master zones* y las *Slave zones*.

Las *Standard zones* por lo general no se modifican ya que contienen información de la zona raíz, la zona de *localhost* y la zona de *loopback* para IPv4 e IPv6.

Dentro de las zonas directas se pueden agregar ya sea zonas Maestras y zonas Esclavas, esto depende del tipo de zonas con que el servidor sea configurado, ya que para algunas zonas estas pueden ser Maestras o Esclavas.

Para declarar una zona Maestra se siguió el siguiente formato:

*zone "prueba.nic.mx" {* **//Esta declaración nos indica a qué nombre de dominio va a responder el servidor**

*type master;* **//Se declara como zona tipo Maestra**

*file "master/prueba.nic.mx";* **//Se indica dónde se encuentra el archivo que tiene la configuración de prueba.nic.mx**

*};*

Los nombres de prueba.nic.mx tanto para zona a la cual va a responder como el archivo de configuración de la zona dependen de la organización o del formato que tengan para que estén organizadas eficientemente y se puedan reconocer fácilmente para su modificación.

Ahora bien para declarar una zona Esclava se siguió el siguiente formato:

*zone "prueba.nic.mx" {*//**Esta declaración nos indica a qué nombre de dominio va a responder el servidor**

*type slave;* //**Se declara como zona tipo Esclava**

*file "slave/prueba.nic.mx";*//**Se indica dónde se encuentra el archivo que tiene la configuración de prueba.nic.mx**

*masters { 132.247.183.1; };* //**Se declara cuál es el Maestro para esta zona tipo Esclava**

*};*

Ahora para configurar el archivo de zona ya sea para zonas tipo maestras (/var/*named*/master) o para zonas esclavas (/var/*named*/slave) se siguió el siguiente formato:

***$ORIGIN .***
***$TTL 7200 ; 2 hours***
***@ IN SOA <primary-name-server> <hostmaster-email> (***
***<serial-number>***
***<time-to-refresh>***
***<time-to-retry>***
***<time-to-expire>***
***<minimum-TTL> )***

***IN NS pruebas1.nic.unam.mx.***
***IN NS pruebas2.nic.unam.mx.***

## 4.3.2.2.2 Zona Inversa

Dentro de las zonas inversas se pueden agregar ya sea zonas Maestras y zonas Esclavas, esto depende del tipo de zonas con que el servidor sea configurado, ya que para algunas zonas estas pueden ser Maestras o Esclavas y se declaran en el archivo de "*named*.conf", que se encuentra en la ruta "/var/*named*/etc".

Para declarar una zona Maestra se sigue un formato parecido al de la zona directa pero en lugar de poner el nombre al cual responde el servidor se pone la dirección IP de la red en sentido contrario y seguido de in-addr.arpa:

*zone "143.248.132.in-addr.arpa" {*//**Esta declaración nos indica a qué dirección IP va a responder el servidor**

*type master;* //**Se declara como zona tipo Maestra**

*file "master/inverso/132.248.143";* //**Se indica dónde se encuentra el archivo que tiene la configuración para el archivo de resolución inversa**

*};*

Las direcciones IP tanto para zona a la cual va a responder como el archivo de configuración de la zona dependen de la organización o del formato que tengan para que estén organizadas eficientemente y se puedan reconocer fácilmente para su modificación.

Ahora bien para declarar una zona Esclava se siguió el siguiente formato:

*zone "143.248.132.in-addr.arpa" {*//**Esta declaración nos indica a qué dirección IP va a responder el servidor**

*type slave;* //**Se declara como zona tipo Esclava**

*file "slave/132.248.143";* //**Se indica dónde se encuentra el archivo que tiene la configuración para el archivo de resolución inversa**

*masters {132.248.204.37;};* //**Se declara cuál es el Maestro para esta zona tipo Esclava**

*};*

Ahora para configurar el archivo de zona inversa ya sea para zonas tipo maestras (/var/*named*/master/inverso) o para zonas esclavas (/var/*named*/slave) se sigue el mismo formato que se utiliza en las zonas directas ya que se usa un archivo de zona de resolución inversa de nombres para traducir una dirección IP en un espacio de nombres particular, con la excepción de que se usan registros de tipo PTR para enlazar las direcciones IP a un nombre de dominio completamente cualificado.

## 4.3.3 Paquete Quagga

Quagga es un software libre que proporciona un conjunto de protocolos de enrutamiento basadas TCP/IP para que una computadora se pueda utilizar como enrutador. Está diseñado mayormente para NetBSD, FreeBSD, Solaris y Linux.

Proporciona los protocolos de encaminamiento basados en TCP/IP:

- RIP v1/v2/ng (*Routing Information Protocol*)
- OSPF v2/v3 (*Open Shortest Path First*)
- BGP -4 y BGP -4+ (*Border Gateway Protocol*)
- IS/IS (*Intermediate system-to-intermediate system*)

Un equipo con el *software* Quagga actúa como un enrutador dedicado, esto es que intercambia información utilizando los protocolos de ruteo configurados. Quagga utiliza la información obtenida para actualizar las tablas de ruteo.

Dentro del software de Quagga hay dos modos de usuario. El primero es el modo normal que sólo puede ver el estado del sistema y el otro es el modo de *enable* que puede ver el estado del sistema y cambiar la configuración del sistema. (Quagga Routing Suite, 2011)

## 4.3.3.1 Instalación de Quagga

Para descargar e instalar el paquete Quagga para la versión 5.1 en una máquina i386 desde el sitio de ftp de OpenBSD (incluyendo las dependencias), se ejecutan los comandos de la siguiente figura:

```
pruebas1# setenv PKG_PATH ftp://ftp.openbsd.org.ar/pub/OpenBSD/5.1/packages/i386
/
pruebas1# pkg_add ${PKG_PATH}quagga-0.99.19p2.tgz
quagga-0.99.19p2: ok
--- +quagga-0.99.19p2 -------------------
Sample Quagga configuration files have been installed in
/usr/local/share/examples/quagga. You will need to install your
configuration files in the /etc/quagga directory.
pruebas1#
```

Figura 4.37 Instalación de Quagga

Los demonios de Zebra, ospfd y ospf6d tienen su propia interfaz terminal o VTY (Virtual Terminal, *Virtual TeletYpe*). Después de la instalación, se habrán agregado las entradas necesarias en '***/etc/services***' que corresponden al número del puerto de cada demonio para poder conectarse a ellos (ver Figura 4.38).

```
zebrasrv        2600/tcp                # servicio zebra
zebra           2601/tcp                # zebra vty
ripd            2602/tcp                # RIPd vty
ripngd          2603/tcp                # RIPngd vty
ospfd           2604/tcp                # OSPFd vty
bgpd            2605/tcp                # BGPd vty
ospf6d          2606/tcp                # OSPF6d vty
```

*Figura 4.38 Demonios de Zebra, ospfd y ospf6d*

Si se especifica un número del puerto al empezar el demonio, estas entradas no pueden necesitarse. Además debemos renombrar los archivos *"/usr/sbin/ospfd"* (para IPv4) y *"/usr/sbin/ospf6d"* (para IPv6) (una sugerencia es renombrarlo por el nombre */usr/sbin/ospfd.bk* y */usr/sbin/ospf6d.bk*, ver Figura 4.39) para que no causen ningún tipo de conflicto al momento de iniciar los servicios de ospf en Quagga.

```
arp             iscsictl        nsdc            rmt             usbdevs
authpf          iscsid          nslookup        rmuser          user
authpf-noip     kadmin          nsupdate        rndc            useradd
bgpctl          kgmon           ntpd            rndc-confgen    userdel
bgpd            kstash          openssl         rotatelogs      userinfo
bos             ktutil          ospf6ctl        route6d         usermod
chat            kvm_mkdb        ospf6d.bk       rpc.bootparamd  vipw
chgrp           ldapctl         ospfctl         rpc.lockd       visudo
chown           ldapd           ospfd.bk        rpc.statd       vos
chroot          ldpctl          pac             rpc.yppasswdd   watchdogd
config          ldpd            pcidump         rtadvd          wsconscfg
cron            logresolve      pkg             rtsold          wsfontload
crunchgen       lpc             pkg_add         rwhod           wsmoused
cryptoadm       lpd             pkg_check       sa              ypbind
cryptoinit      lptest          pkg_create      sasyncd         ypinit
dev_mkdb        mailstats       pkg_delete      sendmail        ypldap
dhcpd           mailwrapper     pkg_info        sensorsd        yppoll
dhcrelay        makedbm         pkg_mklocatedb  sliplogin       yppush
dig             makemap         popa3d          slstats         ypserv
dnssec-keygen   map-mbone       portmap         smtpctl         ypset
dnssec-signzone memconfig       ppp             smtpd           yptest
dvmrpctl        mkalias         pppctl          snkadm          ypxfr
dvmrpd          mkhybrid        pppd            snkinit         ypxfr_lperday
editmap         mknetid         pppoe           snmpctl         ypxfr_lperhour
```

*Figura 4.39 Archivos ospfd y ospf6d*

## 4.3.3.2 Configuración de Quagga

Para poder configurar las interfaces dentro del gestor de enrutamiento Zebra es necesario dar de alta las interfaces que vayamos a utilizar dentro de OpenBSD.

Para iniciar los demonios Zebra, ospfd y ospf6d utilizamos los siguientes comandos:

***# zebra -d***

***# ospfd -d***

***# ospf6d -d***

Como se puede observar en la siguiente Figura, los demonios de zebra, ospfd y ospf6d corresponden al proceso de Quagga.

```
pruebas2# ps aux | grep quagga
_quagga  15515  0.0  0.4  1360  2180 ??  Ss    12:52PM    0:00.34 zebra -d
_quagga   2110  0.0  0.7  2416  3496 ??  Ss    12:52PM    0:01.61 ospfd -d
_quagga   8769  0.0  0.4  1208  1956 ??  Ss    12:52PM    0:00.17 ospf6d -d
root     28657  0.0  0.0   420   196 p0  R+/1   1:49PM    0:00.00 grep quagga (csh)
```

Figura 4.40 Procesos de Quagga

Una vez iniciados los demonios y configuradas las interfaces del servidor utilizamos un VTY para la configuración de zebra, ospfd y ospf6d. Esto significa que es posible conectarse al demonio vía protocolo telnet. Una vez dentro del VTY la configuración es muy similar a la de un enrutador normal.

El siguiente comando nos permite conectarnos al VTY de Zebra:

***# telnet localhost 2601***

***Trying 127.0.0.1...***

***Connected to localhost.***

***Escape character is '^]'.***

***Hello, this is Quagga (version 0.99.19).***

***Copyright 1996-2005 Kunihiro Ishiguro, et al.***

***User Access Verification***

***Password: *****

***Router>***

//Dentro del VTY Zebra solo hay que configurar las interfaces de red de cada servidor.

//Para el servidor pruebas1.

//Con el comando *enable* nos permite acceder al modo privilegiado.

***Router> enable***

***Password: *****

***Router#***

//El comando *configure terminal* nos permite acceder al modo de configuración global //desde el modo privilegiado.

***Router# configure terminal***

//El comando *interface em0* nos permite acceder al submodo de configuración de la //interface de red em0 desde el modo de configuración global.

***Router(config)# interface em0***

//Con el comando *ip address* asignamos una dirección IPv4 a la interfaz de red em0.

***Router(config-if)# ip address 132.247.183.1/28***

//Con el comando *IPv6 address* asignamos una dirección IPv6 a la interfaz de red em0.

***Router(config-if)# IPv6 address 2001:1218:101:183::2/64***

//Con el comando *no shutdown* habilitamos la interfaz em0.

***Router(config-if)# no shutdown***

***Router(config-if)# exit***

//El comando *interface lo0* nos permite acceder al submodo de configuración de la //interface de red lo0 desde el modo de configuración global.

***Router(config)# interface lo0***

//Con el comando *ip address* asignamos una dirección IPv4 a la interfaz de red lo0 que en //este caso estas dos direcciones serán nuestras direcciones de loopback.

***Router(config-if)# ip address 132.247.146.233/32***

***Router(config-if)# ip address 132.247.146.234/32***

//Con el comando *IPv6 address* asignamos una dirección IPv6 a la interfaz de red lo0 que //en este caso estas dos direcciones serán nuestras direcciones de loopback.

***Router(config-if)# IPv6 address 2001:1218:101:186::1/128***

***Router(config-if)# IPv6 address 2001:1218:101:187::1/128***

//Con el comando *no shutdown* habilitamos la interfaz lo0.

***Router(config-if)# no shutdown***

***Router(config-if)# exit***

***Router(config)# exit***

//El comando *copy running-config startup-config* guarda los cambios realizados en la //configuración activa a un archivo de configuración de respaldo que es el que será //utilizado en caso de que por cualquier motivo el dispositivo sea reiniciado.

***Router# copy running-config startup-config***

***Configuration saved to /etc/Quagga/zebra.conf***

Con esto se tendrá configurado el demonio de Zebra para el servidor pruebas1, en el servidor pruebas2 la configuración es parecida y se muestra a continuación:

//Con el comando *enable* nos permite acceder al modo privilegiado.

***Router> enable***

***Password: ********

***Router#***

**//**El comando *configure terminal* nos permite acceder al modo de configuración global //desde el modo privilegiado.

***Router# configure terminal***

//El comando *interface xl0* nos permite acceder al submodo de configuración de la //interface de red xl0 desde el modo de configuración global.

***Router(config)# interface xl0***

//Con el comando *ip address* asignamos una dirección IPv4 a la interfaz de red xl0.

***Router(config-if)# ip address 132.247.183.9/28***

//Con el comando *IPv6 address* asignamos una dirección IPv6 a la interfaz de red xl0.

***Router(config-if)# IPv6 address 2001:1218:101:184::2/64***

//Con el comando *no shutdown* habilitamos la interfaz xl0.

***Router(config-if)# no shutdown***

***Router(config-if)# exit***

//El comando *interface lo0* nos permite acceder al submodo de configuración de la //interface de red lo0 desde el modo de configuración global.

***Router(config)# interface lo0***

//Con el comando *ip address* asignamos una dirección IPv4 a la interfaz de red lo0 que en //este caso estas dos direcciones serán nuestras direcciones de loopback.

***Router(config-if)# ip address 132.247.146.233/32***

***Router(config-if)# ip address 132.247.146.234/32***

//Con el comando *IPv6 address* asignamos una dirección IPv6 a la interfaz de red lo0 que //en este caso estas dos direcciones serán nuestras direcciones de loopback.

***Router(config-if)# IPv6 address 2001:1218:101:186::1/128***

***Router(config-if)# IPv6 address 2001:1218:101:187::1/128***

//Con el comando *no shutdown* habilitamos la interfaz lo0.

***Router(config-if)# no shutdown***

***Router(config-if)# exit***

***Router(config)# exit***

//El comando *copy running-config startup-config* guarda los cambios realizados en la //configuración activa a un archivo de configuración de respaldo que es el que será //utilizado en caso de que por cualquier motivo el dispositivo sea reiniciado.

***Router# copy running-config startup-config***

***Configuration saved to /etc/Quagga/zebra.conf***

Al terminar la configuración de Zebra en los dos servidores DNS se configura ospfd, para conectarnos al VTY de ospfd utilizamos el siguiente comando:

***# telnet localhost 2604***

***Trying 127.0.0.1...***

***Connected to localhost.***

***Escape character is '^]'.***

***Hello, this is Quagga (version 0.99.19).***

***Copyright 1996-2005 Kunihiro Ishiguro, et al.***

***User Access Verification***

***Password: *****

//Para el servidor pruebas1

//Con el comando *enable* nos permite acceder al modo privilegiado.

***OSPF > enable***

//El comando *configure terminal* nos permite acceder al modo de configuración global //desde el modo privilegiado.

***OSPF # configure terminal***

//Con el comando *router ospf* se habilita el proceso ospf. Ospfd todavía no soporta //múltiples procesos ospf. Así que no es posible especificar el número de proceso.

***OSPF(config)#router ospf***

//Con el comando *network* se asigna una dirección de red, lo que hará que se envíen y //reciban publicaciones de enrutamiento a través de esas interfaces, además de que sean //publicadas a los enrutadores vecinos.

***OSPF(config-router)# network 132.247.146.233 /32 area 0.0.0.4***

***OSPF(config-router)# network 132.247.146.234/32 area 0.0.0.4***

***OSPF(config-router)# network 132.247.183.0/29 area 0.0.0.4***

//Con el comando *ospf router-id* se habilita el proceso de ospf y se le asigna la dirección IP
//del servidor pruebas1

***OSPF(config-router)# ospf router-id 132.247.183.1***

***OSPF(config-router)# exit***

***OSPF(conf)#exit***

//El comando *copy running-config startup-config* guarda los cambios realizados en la
//configuración activa a un archivo de configuración de respaldo que es el que será
//utilizado en caso de que por cualquier motivo el dispositivo sea reiniciado.

***OSPF # copy running-config startup-config***

***Configuration saved to /etc/Quagga/ospfd.conf***

//Para el servidor pruebas2 es la misma configuración que el servidor pruebas1 con la única
//diferencia que al habilitar el proceso de ospf se le asigna la dirección IP del servidor
//pruebas2.

***OSPF(config-router)# ospf router-id 132.247.183.9***

Con lo anterior se tendrá configurado el protocolo de enrutamiento ospf para direcciones IPv4 y para la configuración del protocolo ospf para IPv6 nos conectarnos al VTY de ospf6 con el siguiente comando:

***# telnet localhost 2606***

***Trying 127.0.0.1...***

***Connected to localhost.***

***Escape character is '^]'.***

***Hello, this is Quagga (version 0.99.19).***

***Copyright 1996-2005 Kunihiro Ishiguro, et al.***

***User Access Verification***

***Password: *****

//Para el servidor Pruebas1

***ospf6d@plant#***

**//**El comando *configure terminal* nos permite acceder al modo de configuración global.

***ospf6d@plant# configure terminal***

**//**Con el comando *router ospf* se habilita el proceso ospf.

***ospf6d@plant(config)# router ospf6***

**//**Con el comando *router-id* se le asigna la dirección IP del servidor pruebas1.

***ospf6d@plant(config-ospf6)# router-id 132.247.183.1***

***ospf6d@plant(config-ospf6)# exit***

***ospf6d@plant(config)# exit***

//El comando *copy running-config startup-config* guarda los cambios realizados en la //configuración activa a un archivo de configuración de respaldo que es el que será //utilizado en caso de que por cualquier motivo el dispositivo sea reiniciado.

***ospf6d@plant# copy running-config startup-config***

***Configuration saved to /etc/Quagga/ospf6d.conf***

//Para el servidor Pruebas2 es la misma configuración que el servidor pruebas1 con la //única diferencia que se le asigna la dirección IP del servidor pruebas2.

***ospf6d@plant(config-ospf6)# router-id 132.247.183.9***

En las tablas de ruteo de nuestros servidores (ver Figuras 4.41, 4.42 y 4.43) se muestra la lista de todas las redes que el dispositivo puede alcanzar, su métrica, y la forma en que accede a ellas. Si todo funciona adecuadamente, cada dispositivo debiera tener al menos una ruta a cada red que potencialmente sea destino de tráfico.

```
Router# sh ip route
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       I - ISIS, B - BGP, > - selected route, * - FIB route

O   0.0.0.0/0 [110/41] via 132.247.183.22, xl0, 01:10:50
K>* 0.0.0.0/0 via 132.247.183.22, xl0
O>* 10.0.0.0/8 [110/31] via 132.247.183.22, xl0, 01:10:50
O>* 10.128.20.176/29 [110/29] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.72.152/29 [110/140] via 132.247.183.22, xl0, 01:10:50
O>* 10.128.72.248/30 [110/29] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.74.0/29 [110/26] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.74.8/29 [110/26] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.90.16/28 [110/31] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.90.32/27 [110/31] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.90.64/28 [110/31] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.255.0/27 [110/21] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.255.32/27 [110/21] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.255.64/27 [110/21] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.255.96/27 [110/21] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.255.128/27 [110/21] via 132.247.183.22, xl0, 01:10:51
O>* 10.128.255.224/27 [110/25] via 132.247.183.22, xl0, 01:10:51
O>* 10.129.20.0/24 [110/21] via 132.247.183.22, xl0, 01:10:51
O>* 10.129.50.0/24 [110/31] via 132.247.183.22, xl0, 01:10:51
O>* 10.129.55.224/29 [110/30] via 132.247.183.22, xl0, 01:10:51
O>* 10.129.55.232/29 [110/30] via 132.247.183.22, xl0, 01:10:51
O>* 10.129.56.0/24 [110/21] via 132.247.183.22, xl0, 01:10:51
O>* 10.129.57.0/29 [110/21] via 132.247.183.22, xl0, 01:10:51
```

*Figura 4.41 Rutas de ospf en IPv4*

```
Router# sh ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng, O - OSPFv3,
       I - ISIS, B - BGP, * - FIB route.

O   ::/0 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
K>* ::/0 via 2001:1218:403:200::1, xl0
K>* ::/96 via ::1, lo0, rej
K>* ::/104 via ::1, lo0, rej
C>* ::1/128 is directly connected, lo0
K>* ::127.0.0.0/104 via ::1, lo0, rej
K>* ::224.0.0.0/100 via ::1, lo0, rej
K>* ::255.0.0.0/104 via ::1, lo0, rej
K>* ::ffff:0.0.0.0/96 via ::1, lo0, rej
O>* 2001:448::/32 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218::5010:0/126 [110/3] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:0:5::/64 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:1::/48 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101::/48 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:1a::/64 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:100::/64 [110/12] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:101::/64 [110/12] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:103::/64 [110/12] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:104::/64 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:105::/64 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:106::/64 [110/12] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:107::/64 [110/12] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:108::/64 [110/12] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:109::/64 [110/12] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
O>* 2001:1218:101:10a::/64 [110/22] via fe80::20c:dbff:fee3:b400, xl0, 01:11:52
```

*4.42 Rutas de ospf en IPv6*

```
ospf6d@plant# sh ipv6 ospf6 route
*N E1 ::/0                          fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N E1 2001:448::/32                 fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218::5010:0/126         fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N E1 2001:1218:0:5::/64            fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N E1 2001:1218:1::/48              fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N E1 2001:1218:101::/48            fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:1a::/64         fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:100::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:101::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:103::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N E1 2001:1218:101:104::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N E1 2001:1218:101:105::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:106::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:107::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:108::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:109::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:10a::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:10b::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:10e::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:10f::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:110::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:111::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:112::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:113::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:114::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:115::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:116::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
*N IE 2001:1218:101:117::/64        fe80::20c:dbff:fee3:b400     xl0 01:14:31
```

*4.43 Rutas de ospf6 en IPv6*

Con esto tendremos configurado Quagga para que nuestros servidores DNS den el servicio de resolución de direcciones correctamente tanto en IPv4 como en IPv6 y para que en caso de que el servidor primario no funcione, el servidor secundario desarrolle la función del servidor primario y así cumpla el objetivo de *anycast* en IPv4 e IPv6.

## 4.4 Implementación del DNS con IPv6 en Nodos de RedUNAM

Con lo importante que es la red de datos en la UNAM uno de los servicios como el de resolución de nombres de dominio debe tener una alta fiabilidad así como disponibilidad para que los usuarios finales puedan desarrollar sus trabajos, tareas o investigaciones.

Con el esquema de *Anycast* y el de servidores primarios y secundarios NIC-UNAM (Centro de Información de RedUNAM) puede dar este servicio garantizando un buen funcionamiento de los servicios que brinda.

En el esquema de RedUNAM podemos encontrar cuatro nodos "DGTIC, IIMAS, Arquitectura y Zona Cultural" en los cuales se utiliza el esquema de *anycast* en los servidores DNS para proporcionar el servicio de resolución de nombres de dominio en IPv4.

Como parte de la implementación del protocolo IPv6 en los servidores DNS se empezó de manera local configurando en un cliente los servidores DNS configurados para que puedan responder peticiones tanto en IPv4 como en IPv6.

Primero se configuró el cliente con el direccionamiento en IPv4 e IPv6 en las propiedades de conexión de área local (ver Figura 4.44) en el caso de Windows:

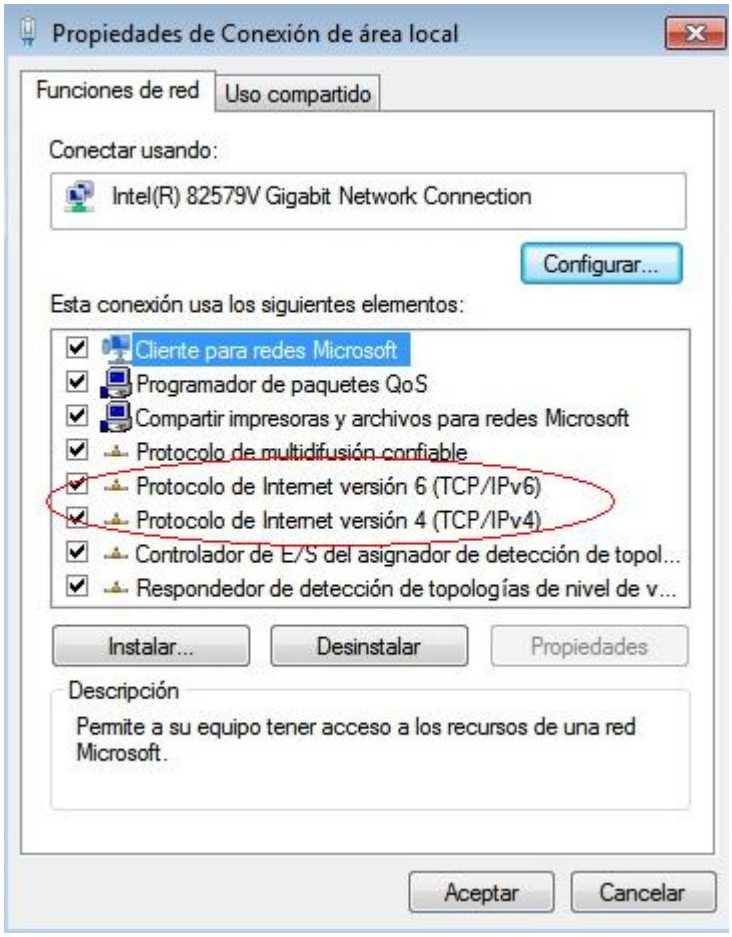


Figura 4.44 Propiedades de Conexión de área local

Una vez configuradas las direcciones tanto IPv4 e IPv6 como las direcciones de los servidores DNS en el cliente en la siguiente figura se puede ver que la conectividad en IPv4 e IPv6 está funcionando correctamente (ver Figura 4.45).

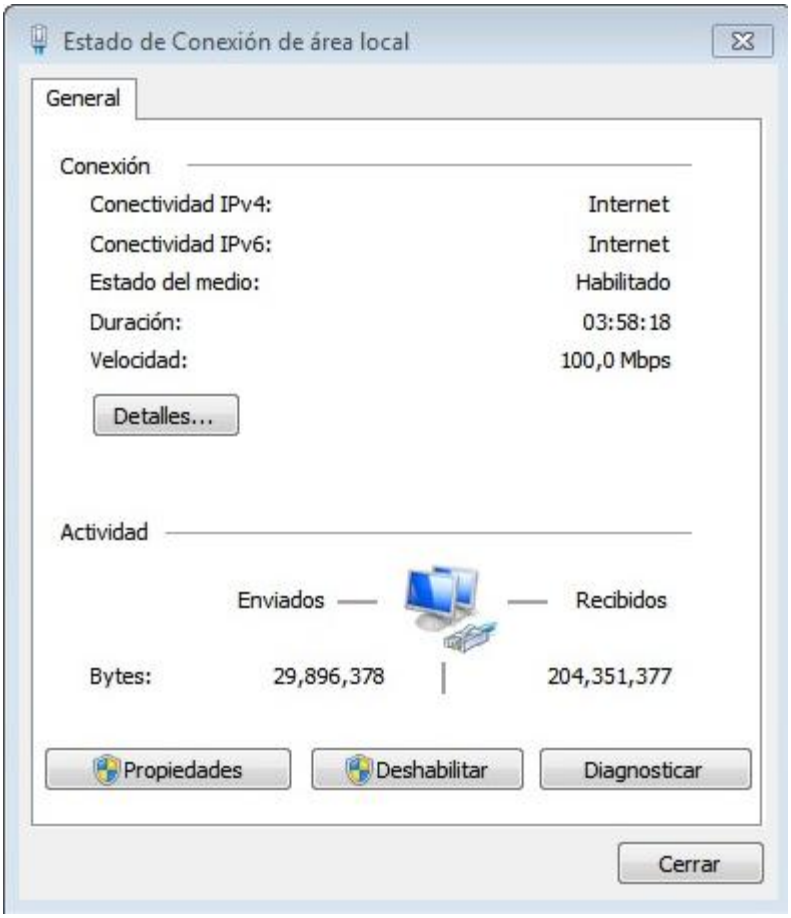


*Figura 4.45 Estado de conexión de área local*

En una terminal utilizando el comando *ipconfig* se observa la dirección IPv4 e IPv6, así como la dirección IPv4 e IPv6 del servidor DNS (ver Figura 4.46)

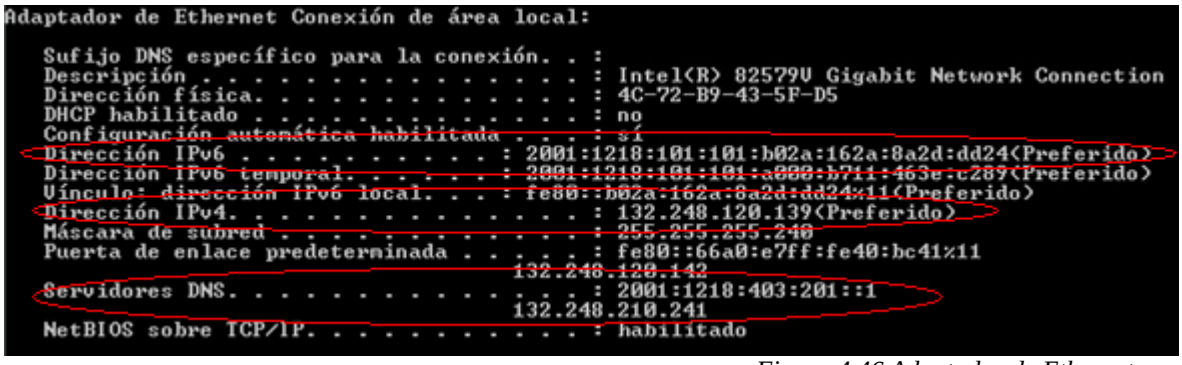


*Figura 4.46 Adaptador de Ethernet*

Teniendo el cliente configurado y los servidores en funcionamiento con el esquema de la figura siguiente:

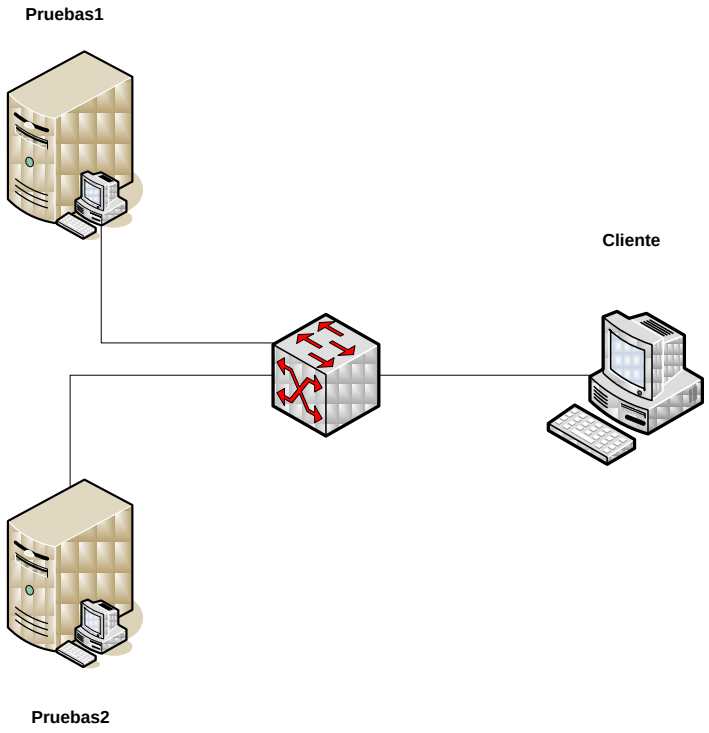


*Figura 4.47 Configuración de prueba 1*

Con el comando "nslookup" podemos comprobar si nuestros DNS están resolviendo los nombres de dominio correctamente (ver Figura 4.48).


*Figura 4.48 Ejemplos de respuestas con nslookup*

Dado el esquema de anycast se verificó que los servidores de nombres de dominio sean estables y comprobando el archivo *"queries"* creado en *"var/named/slave/category"*, con el comando *tail –f* podemos monitorear las peticiones que se le hacen a los servidores (ver Figura 4.49) y en caso de que el primario llegara a dejar de dar el servicio, el secundario empezará a resolver las peticiones.

```
pruebas2# tail -f queries
02-Dec-2013 14:10:09.293 queries: info: client 2001:1218:403:200::5#47266: query: localhost IN A +
02-Dec-2013 14:10:09.298 queries: info: client 2001:1218:403:200::5#32313: query: 1.0.0.127.in-addr.arpa IN PTR +
02-Dec-2013 14:10:09.302 queries: info: client 2001:1218:403:200::5#24410: query: pruebas2.nic.unam.mx IN AAAA +
02-Dec-2013 14:10:09.321 queries: info: client 2001:1218:403:200::5#35591: query: localhost.unam.mx IN A +
02-Dec-2013 14:10:09.321 queries: info: client 2001:1218:403:200::5#13869: query: localhost IN A +
02-Dec-2013 14:10:19.280 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#57421: query: www.google-analytics.com IN A +
02-Dec-2013 14:10:19.330 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#53135: query: www.google-analytics.com IN AAAA +
02-Dec-2013 14:10:32.556 queries: info: client 2001:1218:101:10e:dccf:5127:108f:883e#63425: query: client.akamai.com IN A +
02-Dec-2013 14:10:35.538 queries: info: client 2001:1218:101:10e:dccf:5127:108f:883e#60592: query: liveupdate.symantecliveupdate.com IN A +
02-Dec-2013 14:10:35.541 queries: info: client 2001:1218:101:10e:dccf:5127:108f:883e#60347: query: liveupdate.symantecliveupdate.com IN AAAA +
02-Dec-2013 14:11:30.003 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#55158: query: www.youtube.com IN A +
02-Dec-2013 14:11:30.003 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#60279: query: ad.doubleclick.net IN A +
02-Dec-2013 14:11:30.007 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#50058: query: csi.gstatic.com IN A +
02-Dec-2013 14:11:30.008 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#56745: query: i1.ytimg.com IN A +
02-Dec-2013 14:11:30.008 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#64620: query: s.ytimg.com IN A +
02-Dec-2013 14:11:30.052 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#57091: query: www.youtube.com IN AAAA +
02-Dec-2013 14:11:30.056 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#58149: query: ad.doubleclick.net IN AAAA +
02-Dec-2013 14:11:30.057 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#61569: query: apis.google.com IN A +
02-Dec-2013 14:11:30.060 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#57990: query: csi.gstatic.com IN AAAA +
02-Dec-2013 14:11:30.060 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#53429: query: i1.ytimg.com IN AAAA +
02-Dec-2013 14:11:30.061 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#60089: query: s.ytimg.com IN AAAA +
02-Dec-2013 14:11:30.106 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#52132: query: lh3.googleusercontent.com IN A +
02-Dec-2013 14:11:30.106 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#52039: query: apis.google.com IN AAAA +
02-Dec-2013 14:11:30.113 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#57387: query: lh5.googleusercontent.com IN A +
02-Dec-2013 14:11:30.156 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#53616: query: lh6.googleusercontent.com IN A +
02-Dec-2013 14:11:30.158 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#59323: query: lh3.googleusercontent.com IN AAAA +
02-Dec-2013 14:11:30.158 queries: info: client 2001:1218:101:101:a000:b711:463e:c289#60245: query: lh5.googleusercontent.com IN AAAA +
```

*Figura 4.49 Registro de consultas*

Después de hacer consultas, verificar el archivo de *queries* y comprobar que cuando uno de los servidores deja de resolver las peticiones el otro entra para suplirlo y responde las consultas, se configuraron más equipos dentro del segmento correspondiente a NIC-UNAM para que todos los clientes del segmento pudieran tener el servicio de DNS tanto en IPv4 como en IPv6.

En la siguiente fase para la implementación de los servidores DNS en RedUNAM y verificando que los servidores fueran fiables y tuvieran una buena disponibilidad al hacer las resoluciones de nombres de dominio se procedió a ponerlos en producción en el nodo de la DGTIC (ver Figura 4.50).

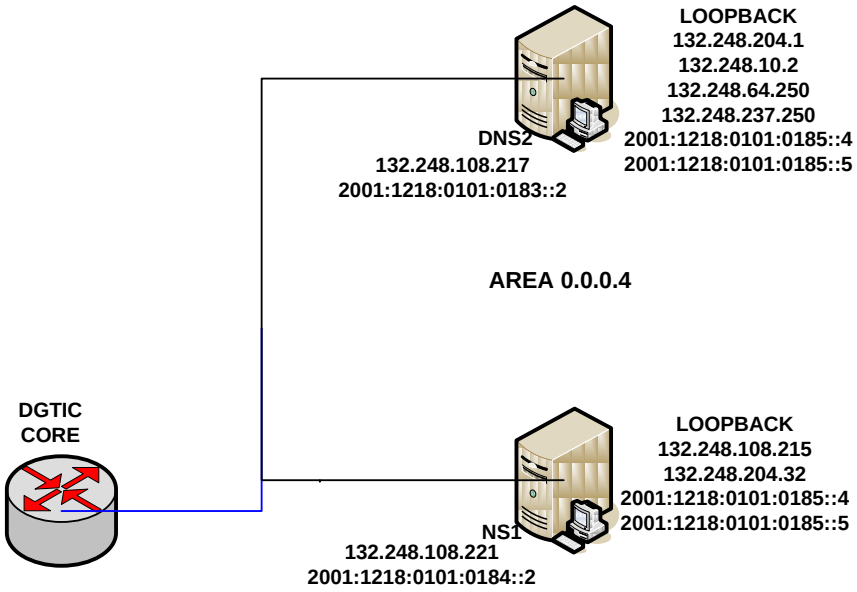


*Figura 4.50 Configuración de prueba 2*

Demostrando la fiabilidad, la disponibilidad y el buen funcionamiento de la implementación de los servidores DNS tanto en IPv4 como en IPv6 en la misma área se decidió proceder a dejar el servidor primario en el nodo de DGTIC y el servidor secundario en el nodo de Zona Cultural (ver Figura 4.51) y así comprobar su funcionamiento en diferentes áreas.

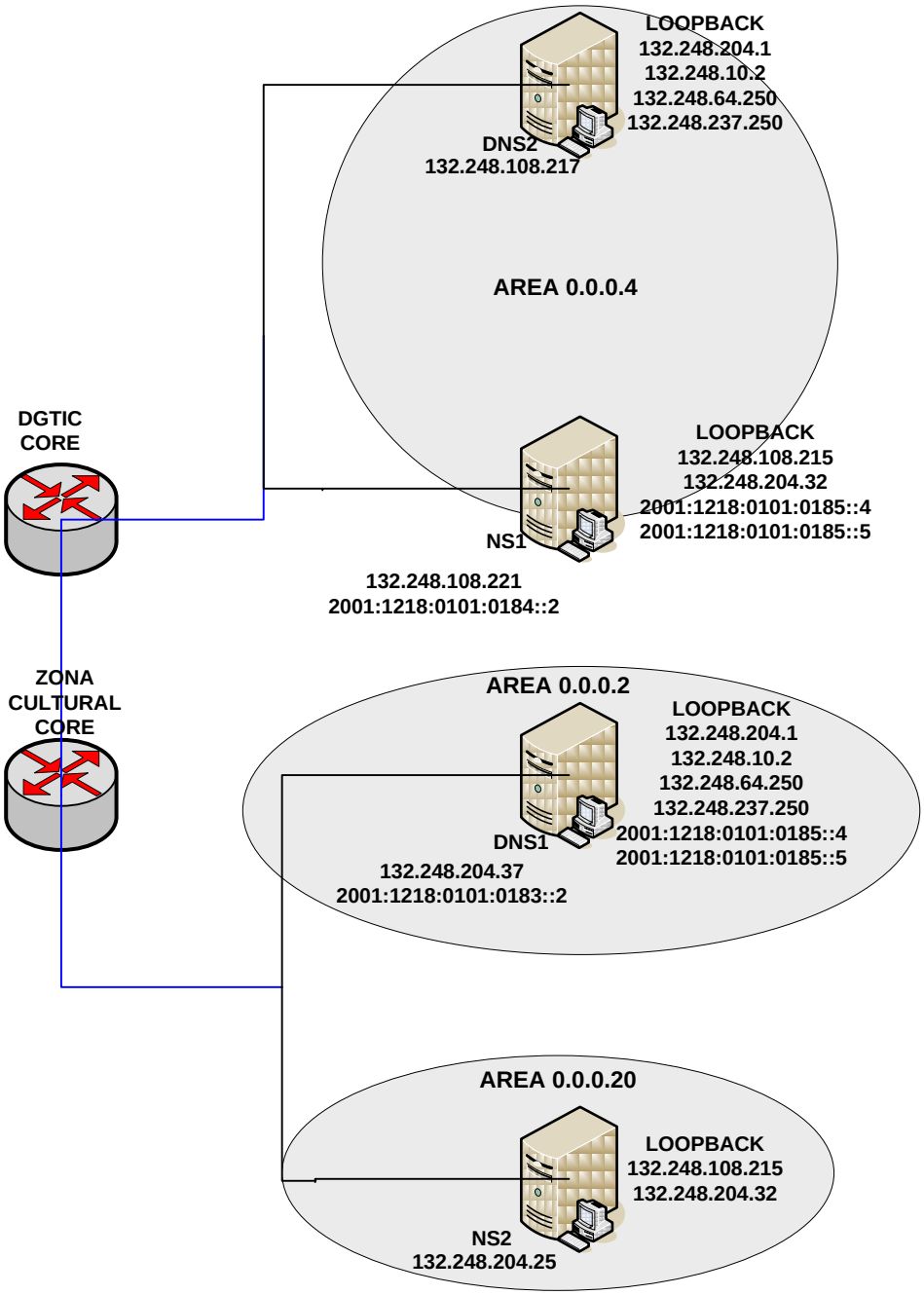


Figura 4.51 Configuración de prueba 3

Demostrando el correcto funcionamiento de los servidores en diferentes áreas se propuso implementarlo en todos los servidores DNS de RedUNAM.

Ya que este esquema es muy factible y comprobando que funcionan correctamente en producción, en el futuro se tendrán todos los servidores DNS resolviendo las peticiones de nombres de dominio tanto en IPv4 como en IPv6 (ver Figura 4.52).

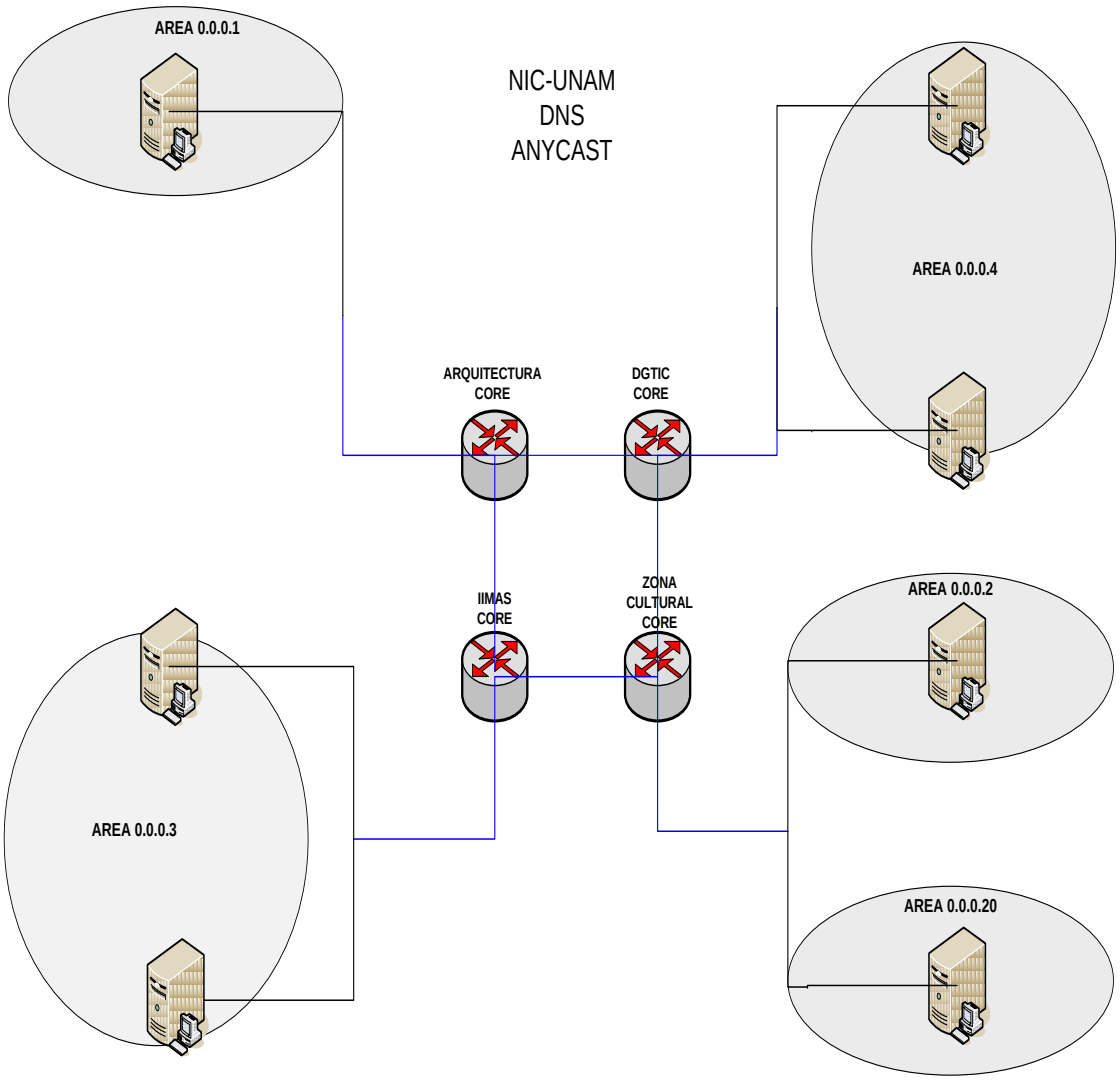


*Figura 4.52 Esquema de Anycast en RedUNAM*

# 4.5 Servicios y Aplicaciones en IPv6 en la UNAM

Con la necesidad de servicios tanto en IPv4 como en IPv6 en RedUNAM se tiene el objetivo de mejorar los servicios que se ofrecen e ir evolucionando junto con las tecnologías que se desarrollan día a día, para ello es necesario que la DGTIC actualice ciertos equipos, así como cada una de las dependencias también necesitan actualizar ciertos equipos para que puedan ser compatibles con las nuevas tecnologías y así ir haciendo la transición y/o coexistencia entre los protocolos de IPv4 e IPv6.

Como este cambio será gradual no es necesario que las dependencias se deshagan de sus equipos actuales ya que la convivencia entre los protocolos de IPv4 e IPv6 se va a ir desarrollando y así poder utilizar el equipo que sólo cuente con funcionalidad para IPv4, lo cual significaría un ahorro en infraestructura en las dependencias.

Pero esto está sujeto a las necesidades de cada dependencia, en caso de que se requieran más servicios y/o espacio de direccionamiento para aumentar su red y usuarios como dependencia ya que el stock de direcciones IPv4 en RedUNAM es casi nulo.

Dentro de los servicios y aplicaciones ofrecidos por RedUNAM se encuentran

- DNS
- Voz sobre IP
- Educación a distancia
- Bibliotecas digitales
- VPN´s
- Red Inalámbrica (RIU)
- Acceso Remoto
- ISP con instituciones u organismos con los que se tiene convenios
- Servidores de correo
- Servidores web

La UNAM a través de la integración de IPv6 puede ofrecer dichos servicios con lo que tendría la oportunidad de atraer más universidades o instituciones que requieran tener acceso a la red IPv6.

# Conclusiones

# CONCLUSIONES

Con la implementación de mi trabajo de tesis se pudo llegar al objetivo y los alcances estimados para que en RedUNAM se pudieran poner en producción servidores que puedan dar el servicio de resolución de nombres de dominio tanto directamente como inversamente.

Esto se puede comprobar realizando consultas a los DNS que se encuentran en producción en los nodos de DGTIC y Zona Cultural o configurando la dirección IPv6 en nuestros equipos (PCs), con esto verificamos la resolución tanto directa como inversa.

Con esto se logró que convivan los protocolos de IPV4 e IPv6 sin afectar los servicios que se ofrecen permitiendo que dos nodos (DGTIC y Zona Cultural) de RedUNAM puedan conectarse tanto en IPv4 como en IPv6.

El sistema operativo OpenBSD ha demostrado poseer un buen soporte en IPv6, lo cual nos permite su uso en ambientes de este tipo ya que la configuración de los servidores DNS es parte vital para la resolución de nombres de dominio, siendo uno de los principales factores para el acceso a internet.

Este trabajo permite que RedUNAM siendo una entidad de educación superior pueda estar preparada para las futuras necesidades de los usuarios y de las demás dependencias que puedan solicitar este servicio.

Y sobre todo que RedUNAM sea un ejemplo y marque una tendencia para que el protocolo IPv6 sea adoptado por otras instituciones de educación superior ya que en un futuro IPv6 será dominante en las nuevas Tecnologías de la Información.

# Glosario de términos

# Glosario de términos

**6Bone:** La red 6bone era una red IPv6 de carácter experimental creada para ayudar a los vendedores y usuarios a participar en la evolución y transición a IPv6. Su enfoque original fue la prueba de normas e implementaciones. Su objetivo principal era la realización de pruebas de procedimientos interoperacionales y transicionales. El día 6 de junio de 2006, concluyó el proyecto 6Bone,

**6REN:** Red IPv6 para Investigación y Educación. En octubre de 1998 la "Energy Science Network" (Esnet) estableció el proyecto de 6REN, el cual es un proyecto de redes de investigación y educación para proveer servicios de tránsito de IPv6, con el fin de facilitar una alta calidad, alto desempeño y operación robusta en redes de IPv6.

**AH:** El encabezado de autenticación proporciona autenticación de datos, una integridad sólida y protección de repetición para los datagramas IP, protege la mayor parte del datagrama IP ya que se inserta entre el encabezado IP y el encabezado de transporte.

**AXFR:** Transferencia por zonas de un DNS primario a un DNS secundario o de un DNS primario a un servidor maestro y de un servidor maestro a un DNS secundario.

**BGP:** Es un protocolo mediante el cual se intercambia información de encaminamiento o ruteo entre sistemas autónomos.

**BIND:** Es el servidor de DNS más comúnmente usado en Internet, especialmente en sistemas Unix.

**Bitnet:** Era una antigua red internacional de computadoras de centros docentes y de investigación que ofrecía servicios interactivos de correo electrónico y de transferencia de ficheros utilizando un protocolo de almacenaje y envío basado en los protocolos Network Job Entry de IBM.

**Broadcast:** Difusión en español, es una forma de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

**CNAT:** Fue parte de las propuestas para el protocolo de la siguiente generación (IPng).

**CUDI:** Fue fundada en abril de 1999 para promover y coordinar el desarrollo de una red de telecomunicaciones de alta tecnología y capacidad, enfocada al desarrollo científico y educativo en México.

**DARPA:** Es la Agencia de Proyectos de Investigación Avanzados de Defensa, es una agencia del Departamento de Defensa de Estados Unidos responsable del desarrollo de nuevas tecnologías para uso militar.

**DGTIC:** Identificada anteriormente como DGSCA (Dirección General de Servicios de Cómputo Académico) y antes de ella como PUC (Programa Universitario de Cómputo). Es la entidad líder en la UNAM en lo relacionado con las tecnologías de información y comunicación (TIC). Es una dependencia de la administración central de la UNAM que gestiona los asuntos relacionados con cómputo de alto rendimiento, redes avanzadas, visualización científica asistida por computadora y realidad virtual.

**DHCP:** Se trata de un protocolo de tipo cliente/servidor en el que generalmente un servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van estando libres, teniendo registro en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después.

**DNS:** Es un sistema de nomenclatura jerárquica para computadoras, servicios o cualquier recurso conectado a Internet o a una red privada. Su función más importante, es traducir (resolver) nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

**DNSSEC:** Se trata de un conjunto de extensiones al DNS que proporcionan a los clientes DNS la autenticación del origen de datos DNS, la negación autenticada de la existencia e integridad de datos, pero no disponibilidad o confidencialidad.

**DSCP:** Hace referencia al segundo byte en la cabecera de los paquetes IP que se utiliza para diferenciar la calidad en la comunicación que quieren los datos que se transportan.

**DSTM:** En el Mecanismo de Transición de Doble Pila, el *host*, el servidor y el enrutador en la red pueden manejar una pila de IPv4 y una IPv6 de forma simultánea. Cuando las dos pilas son utilizadas en los nodos conectados a las redes en los cuales ambos protocolos están habilitados simultáneamente, provee a los nodos la flexibilidad para establecer sesiones extremo a extremo sobre IPv4 o IPv6.

**ESP:** Ofrece confidencialidad para los elementos que encapsula, proporciona los servicios que proporciona AH. Sin embargo, sólo proporciona sus protecciones de la parte del datagrama que encapsula. Proporciona servicios de autenticación opcional para asegurar la integridad de los paquetes protegidos.

**FTP:** Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor.

**IANA:** La Autoridad de Asignación de Números en Internet es la entidad que supervisa la asignación global de direcciones IP, sistemas autónomos, servidores raíz de nombres de dominio DNS y otros recursos relativos a los protocolos de Internet. Actualmente es un departamento operado por ICANN.

**ICANN:** La Corporación de Internet para la Asignación de Nombres y Números es una organización sin fines de lucro con el objetivo de encargarse de cierto número de tareas realizadas con anterioridad por la IANA.

**ICMP:** Es el sub protocolo de control y notificación de errores del Protocolo de Internet. Como tal, se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un enrutador o *host* no puede ser localizado.

**IETF:** Es una organización internacional abierta de normalización, que tiene como objetivos el contribuir a la ingeniería de Internet, actuando en diversas áreas, como transporte, encaminamiento y seguridad.

**IGMP:** Se utiliza para intercambiar información acerca del estado de pertenencia entre enrutadores IP que admiten la multidifusión y miembros de grupos de multidifusión.

**IGP:** Un Protocolo de Pasarela Externo determina si la red es accesible desde el sistema autónomo, y se usa para resolver el encaminamiento dentro del propio sistema.

**InterNIC:** Fue el principal organismo gubernamental de internet responsable de los nombres de dominio y las Direcciones IP, las asignaciones fueron hasta el 18 de septiembre de 1998, cuando este papel fue asumido por la ICANN.

**IP:** El Protocolo de Internet es un protocolo de comunicación de datos digitales clasificado funcionalmente en la Capa de Red según el modelo internacional OSI.

**IPv4:** Es el Protocolo de Internet versión 4 y el primero en ser implementado a gran escala.

**IS/IS:** Es un protocolo de estado de enlace, o SPF (*shortest path first*), por lo cual, básicamente maneja una especie de mapa con el que se fabrica a medida que converge la red. Es también un IGP.

**ISOC:** Es una organización no gubernamental y sin ánimo de lucro, constituida como la única organización dedicada exclusivamente al desarrollo mundial de Internet y con la tarea específica de concentrar sus esfuerzos y acciones en asuntos particulares sobre Internet.

**ISP:** Es una empresa que brinda conexión a Internet a sus clientes. Un ISP conecta a sus usuarios a Internet a través de diferentes tecnologías como DSL, Cablemódem, GSM, Dial-up.

**IXFR:** Un servidor de nombres esclavo sólo descargará las porciones actualizadas de una zona modificada en un servidor de nombres maestro. El proceso de transferencia estándar requiere que la zona completa sea transferida a cada servidor de nombres esclavo hasta por el cambio más pequeño. Para dominios muy populares con archivos de zona muy largos y muchos servidores de nombres esclavos, IXFR hace que la notificación y los procesos de actualización sean menos exigentes en recursos.

**Loopback:** Es una dirección especial que los *host*s utilizan para dirigir el tráfico hacia ellos mismos.

**MLD:** Es el equivalente en IPv6 de la versión 2 del Protocolo de administración de grupos de Internet (IGMPv2) para IPv4. Es un conjunto de mensajes que se intercambian enrutadores y nodos, que permite a los enrutadores descubrir el conjunto de direcciones de multidifusión para las que hay nodos a la escucha en cada interfaz conectada.

**MTU:** Expresa el tamaño en *bytes* de la unidad de datos más grande que puede enviarse usando un protocolo de comunicaciones.

**Multicast:** Multidifusión en español, es el envío de la información en múltiples redes a múltiples destinos simultáneamente.

**NAT:** Es un mecanismo utilizado por enrutadores IP para intercambiar paquetes entre dos redes que asignan mutuamente direcciones incompatibles. Consiste en convertir, en tiempo real, las direcciones utilizadas en los paquetes transportados. También es necesario editar los paquetes para permitir la operación de protocolos que incluyen información de direcciones dentro de la conversación del protocolo.

**NIC-UNAM:** Se encarga de proporcionar servicios como: la asignación de direcciones IP; la asignación de dominios bajo *unam.mx,* administración de dominios externos*(.mx, .edu.mx, .com, etc.)* adquiridos por dependencias de la UNAM, la asignación de dominios inversos, el servicio de servidor secundario, así como el tratamiento de incidentes y quejas de seguridad.

**NGTrans:** Es un grupo de trabajo creado por la IETF para garantizar la coexistencia de IPv4 e IPv6 mientras se ocurre la transición entre ambos protocolos.

**NRO:** La Asociación de recursos numéricos es una organización que une a los 5 registros de Internet regionales, AfriNIC, APNIC, ARIN, LACNIC y RIPE NCC. Entre los principales objetivos de la NRO están, proteger el espacio de direcciones IP no asignadas, promover y proteger el proceso de desarrollo de políticas de Internet y actuar como un punto focalizador y receptor de los comentarios y opiniones de la comunidad de Internet acerca del sistema de RIRs.

**NSI:** Es un proveedor líder de servicios de registro de nombres de dominio.

**NTP:** Es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos a través del enrutamiento de paquetes en redes con latencia variable.

**OSI:** Es el modelo de Interconexión de Sistemas Abiertos de red descriptivo, que fue creado por la Organización Internacional para la Estandarización (ISO). Es un marco de referencia para la definición de arquitecturas en la interconexión de los sistemas de comunicaciones.

**OSPF:** Es un protocolo de enrutamiento jerárquico de pasarela interior, de envestidura dinámica IGP (*Interior Gateway Protocol*), que usa el algoritmo SmoothWall Dijkstra enlace-estado (*LSE - Link State Algorithm*) para calcular la ruta más corta posible.

**PIP:** The P Internet Protocol fue uno de los candidatos que se consideró por la IETF para la próxima versión del Protocolo de Internet y fue diseñado por un lado para ser capaz de manejar muchos enrutamiento, direccionamientos y paradigmas de flujo, pero por otra parte para permitir el reenvío de información relativamente rápido.

**PMTUD:** Es un mecanismo el cual nos permite enviar paquetes grandes, pero especificando en la cabecera IP que no se fragmente, si alguna máquina en el camino del paquete no puede manejar un paquete tan grande, responderá con un ICMP *Destiantion Unreachable Code* 4 y a partir de ahí, se irá repitiendo el proceso con diferentes tamaños de paquete, hasta que se encuentre con un tamaño que sí sea aceptado por todas las máquinas.

**Pool:** Es una cantidad de direcciones IP reservadas para un fin.

**RIP:** Es un protocolo de puerta de enlace interna o IGP utilizado por los *routers*, aunque también pueden actuar en equipos, para intercambiar información acerca de redes IP. Es un protocolo de Vector de distancias ya que mide el número de "saltos" como métrica hasta alcanzar la red de destino. El límite máximo de saltos en RIP es de 15, 16 se considera una ruta inalcanzable o no deseable.

**RIR:** Registro Regional de Internet son organizaciones que supervisan la asignación y el registro de recursos de números de Internet dentro de una región particular del mundo. Los recursos incluyen direcciones IP (tanto IPv4 como IPv6) y números de sistemas autónomos (para su uso en encaminamiento BGP).

**RP:** Se refiere a la asignación de la dirección en la que la dirección del punto de encuentro se codifica en una dirección de grupo de multidifusión IPv6.

**Simple CLNP:** Fue uno de los candidatos que se consideró por la IETF para la próxima versión del Protocolo de Internet que proporciona una solución a largo plazo para el direccionamiento y enrutamiento en Internet.

**SIP:** *The Simple Internet Protocol* fue uno de los candidatos que se consideró por la IETF para la próxima versión del Protocolo de Internet (la versión actual es generalmente conocido como IPv4).

**TCP:** El Protocolo de Control de Transmisión garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron.

**TCP/IP:** En ocasiones se le denomina conjunto de protocolos, en referencia a los dos protocolos más importantes que la componen TCP e IP, que fueron dos de los primeros en definirse, y que son los más utilizados.

**TIC:** Las TIC conforman el conjunto de recursos necesarios para manipular la información: computadoras, los programas informáticos y las redes necesarias para convertirla, almacenarla, administrarla, y transmitirla.

**TLA:** Top Level Aggregation. Identifica a la autoridad de mayor nivel dentro de la jerarquía de encaminamiento

**TP/IX:** Cambió su nombre por el de CATNIP (*Common Architecture for the Internet*) y fue parte de las propuestas para el protocolo de la siguiente generación (IPng).

**TSIG:** Es un método para firmar las transacciones y mensajes de DNS mediante el uso de claves simétricas (secretas) compartidas. Esto incluye los mensajes de consulta recursiva, notificación o consultas dig. Esto nos permitirá restringir quién puede transferir las zonas DNS entre servidores.

**UDP:** Es un protocolo basado en el intercambio de datagramas. Permite el envío de datagramas a través de la red sin que se haya establecido previamente una conexión, ya que el propio datagrama incorpora suficiente información de direccionamiento en su cabecera. Tampoco tiene confirmación ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros; y tampoco se sabe si ha llegado correctamente, ya que no hay confirmación de entrega o recepción.

**UNAM:** Es una universidad pública mexicana, la más grande del país y de América Latina, así como una de las 30 más conocidas del planeta.

**UNIX:** Es un sistema operativo portable, multitarea y multiusuario.

**VLSM:** Las máscaras de subred de tamaño variable representan otra de las tantas soluciones que se implementaron para el agotamiento de direcciones IP y otras como la división en subredes, el enrutamiento de interdominio CIDR, NAT y las direcciones IP privadas.

**VTY:** Permiten el acceso a un ruteador mediante Telnet.

**Web:** La World Wide Web (WWW) o Red informática mundial comúnmente conocida como la web, es un sistema de distribución de documentos de hipertexto o hipermedios interconectados y accesibles vía Internet.

# Lista de RFCs

# Lista de RFCs


[RFC 791] *INTERNET PROTOCOL DARPA INTERNET PROGRAM PROTOCOL SPECIFICATION*
"http://www.ietf.org/rfc/rfc791.txt"

[RFC 1033] *DOMAIN ADMINISTRATORS OPERATIONS GUIDE*
"http://www.ietf.org/rfc/rfc1033.txt"

[RFC 1034] *DOMAIN NAMES - CONCEPTS AND FACILITIES*
"https://www.ietf.org/rfc/rfc1034.txt"

[RFC 1035] *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION*
"http://www.ietf.org/rfc/rfc1035.txt"

[RFC 1550] IP*: Next Generation (IPng) White Paper Solicitation*
"https://tools.ietf.org/html/rfc1550"

[RFC 1752] *The Recommendation for the IP Next Generation Protocol*
"http://tools.ietf.org/html/rfc1752"

[RFC 1883] *Internet Protocol, Version 6 (IPv6) Specification*
"http://tools.ietf.org/search/rfc1883"

[RFC 1918] *Address Allocation for Private Internets*
"https://tools.ietf.org/html/rfc1918"

[RFC 2460] *Internet Protocol, Version 6 (IPv6) Specification*
"http://tools.ietf.org/search/rfc2460"

[RFC 2462] IPv6 Stateless Address Autoconfiguration
https://www.ietf.org/rfc/rfc2462.txt

[RFC 2908] *The Internet Multicast Address Allocation Architecture*
http://tools.ietf.org/html/rfc2908

[RFC 3177] IAB*/IESG Recommendations on IPv6 Address Allocations to Sites*
http://tools.ietf.org/rfc/rfc3177.txt

[RFC 3306] *Unicast-Prefix-based IPv6 Multicast Addresses*
http://tools.ietf.org/html/rfc3306

[RFC 3879] *Deprecating Site Local Addresses*
https://tools.ietf.org/html/rfc3879

[RFC 3956] *Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address*
http://tools.ietf.org/html/rfc3956

# Índice de tablas

# Índice de tablas Página

Tabla 1.1 Clases de direcciones IPv4…….. 11
Tabla 2.1 Encabezado de IPv4……. 33
Tabla 2.2 Comparativo entre los Protocolos de Internet IPv4 e IPv6…….. 34
Tabla 3.1 Estructura de un paquete IPv6……. 41
Tabla 3.2 Ejemplos de dirección IPv6……. 42
Tabla 3.3 Ejemplos de formato comprimido 1……... 42
Tabla 3.4 Ejemplos de formato comprimido 2……... 42
Tabla 3.5 Ejemplos de formato comprimido 3……... 43
Tabla 3.6 Ejemplos de formato comprimido 4……... 43
Tabla 3.7 Ejemplos de formato comprimido con ceros iniciales en cada grupo. 43
Tabla 3.8 Ejemplo de dirección IPv6 mapeada a IPv4……... 44
Tabla 3.9 Ejemplo de dirección IPv6 compatible con IPv4……. 44
Tabla 3.10 Ejemplo de identificador de rango…….. 44
Tabla 3.11 Formato de Dirección Link-Local…….. 46
Tabla 3.12 Formato de Dirección Site-local…… 47
Tabla 3.13 Prefijos de multicast……... 47
Tabla 4.1 Opciones de instalación……... 63
Tabla 4.2 Distribución de espacio…… 68

# Índice de figuras

# Índice de figuras Página


Figura 1.1 Evolución del protocolo IP……. 12
Figura 1.2 Registros Regionales de Internet…… 13
Figura 1.3 Mapa de ccTLD…….. 19
Figura 1.4 Problemas con gTLD-MoU……... 20
Figura 3.1 Formato descriptivo de una dirección IPv6 compatible con IPv4…. 45
Figura 3.2 Dirección IPv6 mapeada a IPv4……. 46
Figura 3.3 Dirección Agregable Global……... 48
Figura 3.4. Unicast……... 50
Figura 3.5 Multicast……. 51
Figura 3.6 Anycast……... 52
Figura 4.1 Anycast……... 60
Figura 4.2 Distribución de servidores DNS raíz…….. 61
Figura 4.3 Opciones de OpenBSD……... 63
Figura 4.4 Instalación de OpenBSD…… 64
Figura 4.5 Configuración del teclado…….. 64
Figura 4.6 Identificación del servidor…….. 64
Figura 4.7 Configuración de interfaces de red……. 64
Figura 4.8 Configuración de dirección IPv4 …….. 64
Figura 4.9 Configuración de dirección IPv6…… 65
Figura 4.10 Configuración del servidor DNS…….. 65
Figura 4.11 Configuración de red…… 65
Figura 4.12 Configuración de contraseña…… 65
Figura 4.13 Demonio sshd……... 66
Figura 4.14 Demonio ntpd……... 66
Figura 4.15 Interfaz gráfica…… 66
Figura 4.16 Inicio de interfaz gráfica…….. 66
Figura 4.17 Configuración de la consola……. 66
Figura 4.18 Creación de usuario adicional…….. 66
Figura 4.19 Particiones por defecto …….. 67
Figura 4.20 Opciones para particionar el almacenamiento……. 67

Figura 4.21 Espacio total disponible …….. 68
Figura 4.22 Distribución de espacio …….. 68
Figura 4.23 Creación de particiones …….. 69
Figura 4.24 Conjuntos de OpenBSD …….. 69
Figura 4.25 Conjuntos instalados…… 70
Figura 4.26 Configuración de hora, día y año……. 70
Figura 4.27 Acceso a OpenBSD…….. 71
Figura 4.28 Comando gunzip…….. 71
Figura 4.29 Comando tar……. 72
Figura 4.30 Archivo ntp……... 73
Figura 4.31 Sincronización de reloj …….. 73
Figura 4.32 Archivo *host*s…… 74
Figura 4.33 Archivo *host*name.em0 …….. 76
Figura 4.34 Archivo mygate …….. 77
Figura 4.35 Archivo myname…….. 78
Figura 4.36 Archivo resolv.conf…….. 79
Figura 4.37 Instalación de Quagga…….. 93
Figura 4.38 Demonios de Zebra, ospfd y ospf6d……. 93
Figura 4.39 Archivos ospfd y ospf6d…….. 94
Figura 4.40 Procesos de Quagga……. 95
Figura 4.41 Rutas en IPv4……... 102
Figura 4.42 Rutas de ospf en IPv6 …….. 103
Figura 4.43 Rutas de ospf6 en IPv6 …….. 103
Figura 4.44 Propiedades de Conexión de área local ……... 105
Figura 4.45 Estado de conexión de área local……. 106
Figura 4.46 Adaptador de Ethernet…….. 106
Figura 4.47 Configuración de prueba 1……... 107
Figura 4.48 Ejemplos de respuestas con nslookup…….. 108
Figura 4.49 Registro de consultas…… 109
Figura 4.50 Configuración de prueba 2……... 110
Figura 4.51 Configuración de prueba 3……... 111
Figura 4.52 Esquema de Anycast en RedUNAM…… 112

# Bibliografía

# Bibliografía


Alcántara, A. F. (1 de Junio de 2012). *IPv6 en la UNAM.* Obtenido de IPv6 México: http://www.ipv6.unam.mx/documentos/IPv6_Mexico-UNAM-RDU_junio2012.pdf

*BIND*. (Diciembre de 2013). Obtenido de https://www.isc.org/downloads/bind/

Cáceres, D., & Ortiz, O. (2010). *Estudio comparativo de VoIP y telefonía IP en IPv6 e IPv4.* Obtenido de http://dspace.espoch.edu.ec/bitstream/123456789/1313/1/98T00001.pdf

EcuRed. (25 de Febrero de 2014). *IPv4*. Obtenido de http://www.ecured.cu/index.php/Ipv4

InterNIC. (3 de Enero de 2013). Obtenido de ftp://ftp.internic.net/domain/named.root

ISC. (2013). *BIND 9 Administrator Reference Manual*. Obtenido de http://ftp.isc.org/isc/bind9/cur/9.6/doc/arm/Bv9ARM.pdf

NIC México. (s.f.). *Mecanismos de transición.* Obtenido de http://www.ipv6.mx/index.php/informacion/rfcs

*OpenBSD*. (1996). Obtenido de http://www.openbsd.org/index.html

Palet, J., & Cabellos, A. (1 de Mayo de 2004). *El Protocolo IPv6*. Obtenido de http://www.6sos.org/documentos/6SOS_El_Protocolo_IPv6_v4_0.pdf

Pérez, J. M. (Junio de 2007). *Protocolos TCP/IP los pilares de la tierra.* Obtenido de http://www.ecobachillerato.com/monograficos/baquiapdf-200706.pdf

*Quagga Routing Suite*. (1 de Abril de 2011). Obtenido de http://www.nongnu.org/quagga/docs/docs-info.html

Red Hat, Inc. (2005). *Berkeley Internet Name Domain (BIND)*. Obtenido de http://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-bind.html

RFC1034. (Noviembre de 1987). *DOMAIN NAMES - CONCEPTS AND FACILITIES*. Obtenido de http://tools.ietf.org/html/rfc1034

RFC1035. (Noviembre de 1987). *DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION*. Obtenido de http://tools.ietf.org/html/rfc1035

RFC1752. (Enero de 1995). *The Recommendation for the IP Next Generation Protocol*. Obtenido de http://www.ietf.org/rfc/rfc1752.txt

RFC2181. (Julio de 1997). *Clarifications to the DNS Specification*. Obtenido de http://tools.ietf.org/html/rfc2181

RFC2373. (Julio de 1998). *IP Version 6 Addressing Architecture*. Obtenido de http://www.ietf.org/rfc/rfc2373.txt

RFC2460. (Diciembre de 1998). *Internet Protocol, Version 6 (IPv6)*. Obtenido de https://www.ietf.org/rfc/rfc2460.txt

RFC3596. (Octubre de 2003). *DNS Extensions to Support IP Version 6*. Obtenido de https://tools.ietf.org/html/rfc3596

RFC4291. (Febrero de 2006). *IP Version 6 Addressing Architecture*. Obtenido de http://tools.ietf.org/html/rfc4291

RFC4301. (Diciembre de 2005). *Security Architecture for the Internet Protocol*. Obtenido de http://www.ietf.org/rfc/rfc4301.txt

RFC5952. (Agosto de 2010). *A Recommendation for IPv6 Address Text Representation*. Obtenido de http://tools.ietf.org/html/rfc5952

RFC6052. (Octubre de 2010). *IPv6 Addressing of IPv4/IPv6 Translators*. Obtenido de http://tools.ietf.org/html/rfc6052

RFC791. (Septiembre de 1981). *INTERNET PROTOCOL*. Obtenido de http://www.ietf.org/rfc/rfc791.txt

Robles, O. (1 de Septiembre de 1998). *Cronología del DNS.* Obtenido de http://www.dominiuris.com/boletines/doctrinal/cronologia.htm

Rodriguez, M. C. (2 de Septiembre de 2003). *Cuadro comparativo IPv4 - IPv6*. Obtenido de http://www.evidalia.es/trucos/index_v2-261-11.html

Sánchez, D. M. (29 de Noviembre de 2006). *Estudio del proceso de transición del protocolo IPv4 hacia el IPv6*. Obtenido de http://dspace.ups.edu.ec/bitstream/123456789/205/5/Capitulo%204.pdf

The Number Resource Organization. (3 de Febrero de 2011). *Free Pool of IPv4 Address Space Depleted.* Obtenido de http://www.nro.net/news/ipv4-free-pool-depleted