[::]

IPv6 para operadores de Red

Alejandro Acosta
Santiago Aggio
Sofía Silva Berenguer
Guillermo Cicileo
Tomas Lynch
Antonio M. Moreiras
Mariela Rocha
Arturo Servin

Internet Society | ISOC-AR Capítulo Argentina

[::]

IPv6 para operadores de Red

Alejandro Acosta

Santiago Aggio

Guillermo Cicileo

Tomas Lynch

Antonio M. Moreiras

Mariela Rocha

Arturo Servin

Sofía Silva Berenguer

Internet Society
ISOC-AR
Capítulo
Argentina

IPv6 para Operadores de Red, 1ª Edición. 2014
Ebook

ISBN 978-987-45725-0-9

Suipacha 128 3° piso F
Ciudad de Buenos Aires, Argentina

Diseño Integral: Transversal Branding

Comité Editor: Christian O'Flaherty y Carlos M. Martínez

Agradecimientos

A Internet Society (www.isoc.org.) por haber donado los fondos que han permitido la realización de este Proyecto y su constante apoyo para estimular la continuidad y relevancia de los Capítulos.

A LACNIC (www.lacnic.net) por sus aportes al contenido de este libro así como también por las tareas de capacitación orientadas a la toma de conciencia, que en torno a IPv6 vienen desarrollando en Latinoamérica y Caribe.

A todos los autores y colaboradores que han posibilitado con su dedicación y trabajo la concreción de este Proyecto, que tiene por objeto contribuir a la Comunidad de Internet en la adopción e implementación del nuevo Protocolo IPv6.

La Comisión Directiva
ISOC-AR Capítulo Argentina de Internet Society

::

Indice de contenidos

Autores

Alejandro Acosta

Alejandro Acosta estudió Licenciatura en Computación en la Universidad de Nueva Esparta, Venezuela (1995-2001) y luego obtuvo un master en Gestión de Tecnologías de la Información de la misma universidad.
Actualmente Alejandro es Ingeniero I+D de Lacnic. Anteriormente fue miembro de la Comisión Electoral de LACNIC y presidente de LAC-TF (IPv6 Task Force). Coordina el encuentro anual del Foro Latinoamericano de IPv6 y modera la lista de correo de la IPv6 Latin America Task Force. Es profesor de TCP/IP en la Universidad de Nueva Esparta para estudiantes del noveno semestre.
También ha participado en varios encuentros durante los últimos años incluyendo LACNIC, LACNOG, IGF, LACIGF y encuentros de la IETF. Ha obtenido varias certificaciones, entre ellas la IPv6 Sage Certified (Hurricane Electric, 10 de noviembre) y la Novell Certified Linux Administrator (Novell CLA, febrero de 2010).
También ha participado en artículos para revistas tecnologicas.
Ha sido miembro de Lacnic, del Grupo de Usuarios Linux de Venezuela, IPv6VE y miembro del Capitulo ISOC de Venezuela.

Santiago Aggio

Ingeniero Electrónico especializado en redes de datos y cómputo en áreas científicas y académicas. Actualmente implementa tecnologías de Computación de Alto Desempeño (HPC) mediante clusters y máquinas virtuales, sobre redes IPv4 e IPv6. Ha participado en múltiples proyectos del ámbito académico, desarrollando soluciones de QoS y recientemente implementando sistemas de monitoreo sobre enlaces de Internet y de Redes Avanzadas. Actualmente se desempeña en gestión de redes en la Universidad Tecnológica Nacional, Facultad Regional Bahía Blanca y como Profesional en el CCTBB (Centro Científico Tecnológico Bahía Blanca) dependiente del Conicet en Argentina.

Guillermo Cicileo

Guillermo Cicileo se desempeña actualmente como Coordinador General de la RIU, red de las universidades nacionales de Argentina. Forma parte del comité de evaluación del FLIP6 - Foro Latinoamericano de IPv6 desde 2007 a la actualidad.
Ha participado activamente de la creación de RedCLARA (Cooperación Latinoamericana en Redes Avanzadas), siendo miembro de la Comisión Técnica inicial del proyecto. Posteriormente tuvo a su cargo la coordinación del Grupo de Trabajo de Multicast de RedCLARA desde 2005 hasta 2008 y miembro de los grupos de trabajo de IPv6 y Ruteo Avanzado.
Ha estado involucrado en LACNIC desde su creación, participando tanto en los grupos de trabajo como en el Foro de Políticas de Lacnic, en las reuniones de operadores LACNOG y en las principales reuniones y foros de Internet de la región.
Ha sido instructor en los workshops de enrutamiento avanzado organizados por distintas organizaciones como CLARA, WALC y LACNIC, dictando capacitaciones sobre multicast, IPv6 y BGP entre otros temas.
Junto a otros autores ha escrito el libro "IPv6 para Todos" (proyecto financiado por Internet Society, Capítulo Argentina).
Su actividad laboral ha estado ligada a las redes científico y académicas a nivel nacional e internacional, desempeñándose en esas tareas durante mas de 15 años.

Tomás Lynch

Recibió su título de ingeniero electrónico de la Facultad de Ingeniería de la Universidad de Buenos Aires en 1997 y su M.Sc. in Engineering Management de la Facultad de Ingeniería de la Florida International University en 2005. Desde 2010, trabaja en Ericsson como Solutions Architect planeando, diseñando y entregando soluciones relacionadas con la convergencia de las redes IP móviles y fijas.
Anteriormente se desempeñó en distintas posiciones en Global Crossing e Impsat Fiber Networks relacionadas con la arquitectura del backbone IP. Sus intereses son diseño y arquitectura de redes, integración de redes IP y el desarrollo de Internet. Es miembro de ISOC desde el 2000. Participa de las reuniones de LACNOG desde el 2010.

Antonio M. Moreiras

Gerente de Proyectos y Desarrollo del CEPTRO (Centro de Estudios y Proyectos en Tecnologías de Redes y Operaciones) en el NIC.br, donde coordina el IPv6.br, una iniciativa para la difusión de IPv6 en Brasil. También es responsable de hacer disponible la Hora Legal Brasilera gratuita en la Internet, via NTP, de entrenamientos dirigidos a proveedores de Internet y Sistemas Autónomos, así como de otros proyectos. Como parte de sus actividades en el NIC.br, es ponente habitual en conferencias y eventos relacionados con IPv6. Participa regularmente de foros técnicos y sobre gobernanza de Internet. Moreiras es tambien miembro fundador de ISOC Brasil. De formación es ingeniero electricista (1999) y Master en ingeniería (2004), por la Escola Politécnica da USP, con un MBA de la UFRJ (2008). Estudió Gobernanza de Internet en la Diplo Foundation (2009) y en la Escuela del Sur de Gobernanza de Interenet (2010). De 1999 a 2007 trabajó en la Agência Estado, donde entre otras actividades, coordinó el equipo de calidad de software. De 2002 a 2007 fue también profesor en cursos de Computación y Redes en la Unicid, Facultades Tancredo Neves y Facultades Radial.

Mariela Rocha

Mariela Rocha es Ingeniera en Sistemas de Información de la Universidad Tecnológica Nacional de Argentina y en la actualidad es la coordinadora técnica en la Red de Interconexión Universitaria, donde dedica su experiencia al despliegue de nuevas tecnologías sobre la red de Universidades Nacionales de Argentina.
Desde sus inicios se ha abocado a las nuevas tecnologías y a la ingeniería de redes, fundamentalmente en el ámbito académico. Comenzó a trabajar con IPv6 en el año 2003, participando en workshops y capacitaciones de la FIU (Florida International University), cuando se desempeñaba en la Red Teleinformática Académica (RETINA), donde contribuyó a consolidar el despliegue de IPv6 en la red nacional.
Ha dictado numerosas capacitaciones sobre IPv6 para Universidades de Argentina, Proveedores de Servicios y otros organismos como NAP CABASE. También se ha desempeñado como expositora sobre el tema en la región.
Entre 2006 y 2011 se desempeñó como coordinadora del Foro Latinoamericano de IPv6 y de la IPv6 Task Force de América Latina y el Caribe.
Es co-autora del libro "IPv6 para Todos", un proyecto impulsado por Internet Society, Capitulo Argentina, cuyo material fue editado en multiples idiomas y distribuido en Latinoamérica y otras regiones del mundo.

Arturo L. Servin

Actualmente trabaja en Google Inc. como Gerente de Peering y Distribución de Contenido para Iberoamérica y el Caribe. Antes de incorporarse a Google, fue Gerente de Tecnología del Registro de Direcciones de Internet para América Latina y el Caribe (LACNIC). También trabajó como ingeniero de investigación, consultor y administrador de red en diferentes organizaciones del Reino Unido y México. Arturo recibió su doctorado del Departamento de Ciencias de la Computación de la Universidad de York, donde sus investigaciones se centraron en la inteligencia artificial, el aprendizaje automático y la seguridad de las redes. Además, cuenta con un Masters en Administración de Telecomunicaciones y un B.S. en Ingeniería en Sistemas Electrónicos, ambos del Campus del ITESM de Monterrey, México. Arturo ha trabajado en numerosos proyectos de innovación, entre ellos el desarrollo de Internet-2 en México, donde se desempeñó como Presidente del Comité de Desarrollo de la Red y Coordinador del Grupo de Trabajo sobre IP-Multicast.

Sofía Silva Berenguer

Egresada de la Universidad de Montevideo como Ingeniera Telemática. Trabaja actualmente como Especialista Senior en Seguridad y Estabilidad en el Registro de Direcciones de Internet para Latinoamérica y el Caribe (LACNIC). Se desempeñó previamente como Ingeniera en Infraestructura, Sistemas y Seguridad, Oficial de Políticas y Analista de Solicitudes también en LACNIC. Antes de trabajar en LACNIC, Sofía trabajó en el área de Networking de IBM Uruguay S.A. y se desempeñó como Jefe de Seguridad Informática en un proyecto de outsourcing de IBM.

::

Introducción

El objetivo de este libro es el de informar de manera sencilla y práctica pasos a seguir para quienes desean implementar IPv6. El libro aborda esta problemática desde varias perspectivas concretas, de la misma manera que se hizo en el exitoso libro "IPv6 para Todos" publicado por el ISOC-AR Capítulo Argentina de Internet Society en el año 2009.

En el mundo habrá en 2014 siete mil millones de teléfonos móviles, esto significa a nivel global una penetración promedio cercana al 100%. Teléfonos fijos y conexiones a Internet muestran índices de penetración mucho más bajos, sin embargo todos los servicios relacionados con Internet siguen en constante crecimiento.

En la actualidad, casi todos los dispositivos conectados a Internet usan IPv4, sin embargo la cantidad de dispositivos a conectar está limitada por la cantidad de direcciones IPv4 existentes, que se están agotando rápidamente en todo el mundo.

IPv6 es el protocolo que permite hoy y hará posible en el futuro que todos los dispositivos, fijos o móviles, puedan conectarse a Internet. Este nuevo protocolo permitirá a Internet seguir creciendo y recibir y enviar información a millones de dispositivitos de todo tipo.

La administración de las direcciones IPv6 ha sido delegada a los RIRs regionales, nuestro RIR Regional es LACNIC, de la misma manera que se hizo con IPv4.

Si bien IPv6 fue acordado en los 90, su implementación ha sido más lenta de lo esperado, ya ha llegado el momento del agotamiento de direcciones IPv4 y crecerá la demanda de direcciones IPv6 así como la necesidad de comprender su uso adecuado.

El futuro de una Internet accesible, neutral y abierta, depende del exitoso desarrollo e implementación de IPv6. Por su relevancia en el presente y futuro de Internet, la transición e implementación de IPv6 es de gran

importancia para todos los actores del ecosistema de Internet. Gobiernos, sector privado, academia y sociedad civil, deben alinear sus esfuerzos para lograr el desarrollo de IPv6 en sus espacios de influencia.

Los gobiernos deben comprender la importancia de incluir en todas las compras de productos y contratos de servicios TIC la compatibilidad con IPv6. Los académicos deben actualizar sus programas educativos con nuevos conocimientos, tratando de incorporar en sus clases las experiencias concretas de implementación por parte de proveedores de servicios, desarrolladores de aplicaciones y de quienes gestionan redes IP.

Los esfuerzos de capacitación deben reforzarse en relación a IPv6, de tal manera de crear conciencia y liderazgo de cambio entre quienes reciben esta capacitación.

El contenido de la presente obra resulta un recurso indispensable para todos quienes quieran implementar IPv6.

¿Cómo IPv6 ayudará al crecimiento exponencial que tienen las redes móviles? En el libro se analizan los componentes de las redes de servicios móviles actuales 2G, 3G y LTE y las distintas opciones de implementación de IPv6. Se muestra también la configuración de los componentes necesarios para soportar IPv6 en la red del operador móvil.

Se han desarrollado diversas tecnologías con el objetivo de permitir la transición y coexistencia entre los protocolos en Internet. En este libro se analizan la mejores técnicas para cada caso, aportando elementos que permitan comprender el principio de funcionamiento y los casos de uso de cada una de ellas.

El ruteo externo necesita del protocolo BGP, el que se ha usado por más de 20 años para llevar la información de rutas de IPv4. Con el tiempo, este protocolo se ha ido extendiendo, permitiendo transportar otro tipo de información más allá de los prefijos IPv4 y es así como el libro explica cómo se ha extendido para poder manejar ruteo externo en IPv6.

El libro aborda el tema de monitoreo en IPv6, utilizando las denominadas herramientas “OpenSource” (Código abierto). El monitoreo de la red y de los servicios que hay implementados sobre ella, cobran mas importancia cuanto mas críticos son los servicios o vínculos de la red y del grado de control que se quiera tener sobre ellos, y no solo habrá que hacerlo para IPv4, sino también para IPv6.

Olga Cavalli
ISOC-AR
Capítulo Argentina
de Internet Society
Secretaria

.1

Plan de direccionamiento

1.1_
Modelo Jerárquico de asignación

1.1.1. Modelo de asignación IANA y RIRs

La asignación de direcciones IP lleva un sistema jerárquico, específicamente es un sistema Top-Down formando un árbol invertido.

La parte más alta de este sistema hace referencia al Internet Assigned Numbers Authority (IANA) quien delega los recursos a los Registros Regionales (RIRs), que a su vez tienen sus políticas para delegar recursos (IPs, ASNs) a sus clientes, estos últimos, entre otros incluyen ISPs (Proveedores de Servicios de Internet) y usuarios finales.

El modelo de asignación de IANA a RIR funciona de la siguiente manera: El IANA posee un pool de direcciones, conocido como el Pool Global de direcciones, ellos realizan la asignación "hacia abajo" siguiente el modelo Top-Down mencionado anteriormente.

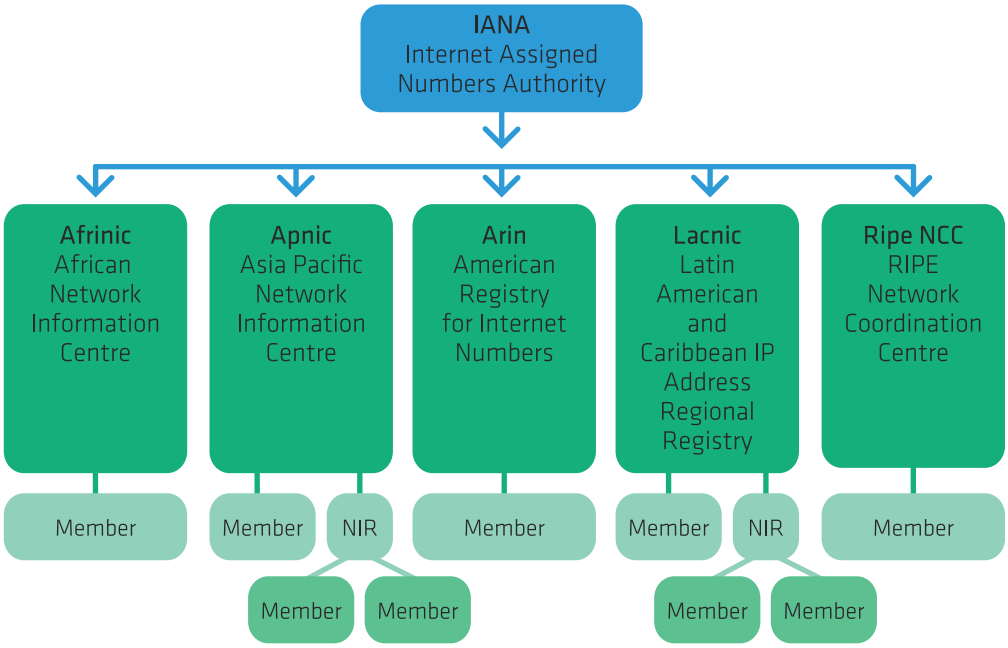

FIGURA 1: MODELO IANA-RIR

En líneas generales la IANA alimenta de recursos de Internet a los RIR, estos a sus clientes y finalmente estos últimos a sus respectivos clientes finales. Los siguientes conceptos son muy importantes:

- **RIR: Regional Internet Registry.** Son quienes reciben recursos directamente del IANA y le entregan recursos a LIRs o NIRs según sea el caso.

- LIR: Local Internet Registry. Son aquellas entidades quienes pueden solicitar recursos al RIR y su país no cuenta con NIR.

- NIR: National Internet Registry. Algunos países cuentan con la modalidad de NIR, en este caso los LIR (ejemplo: ISPs) deben solicitar recursos al NIR directamente y no al RIR. En Latinoamerica contamos con dos NIR: México y Brasil.

Para comprender un poco más como es la asignación de bloques IPv6 a los RIRs demos una vista rápida al siguiente resumen de las políticas y/o reglas de asignación[1]:

- El espacio mínimo que un RIR recibe de IANA es un /12.

- IANA proveerá al RIR con suficiente espacio de direcciones para soportar una operación estimada para al menos 18 meses

- IANA permite al RIR a utilizar sus propias políticas y estrategias de asignación

- Un RIR se vuelve elegible al recibir más espacio IPv6 en caso de que disponga de menos de un 50% de un /12 o el RIR indique que le queda menos de 9 meses de operación con su espacio disponible actual

- Para obtener espacios y recursos adicionales de IANA es necesario que el RIR realice su aplicación respectiva con los justificativos necesarios

Es obligación del RIR actualizar su website y realizar el anuncio respectivo del espacio recibido por IANA.

1.1.2. Modelo general de asignación en un proveedor

El modelo general de un operador se puede apreciar como un pequeño caso de IANA a los RIR.

En líneas generales un proveedor al recibir un bloque IPv6 por parte de su RIR debe poseer un plan de direcciones IPv6 (de la misma manera como se hace en IPv4).

Gracias al enorme espacio de IPv6, se ha tornado muy común asignar bloques específicos para ciertas tareas. Por ejemplo:

a) Bloque de direcciones para redes WAN
b) Bloque de direcciones para LAN
c) Direcciones Loopback para diferentes dispositivos
d) Un espacio para ULAs si es necesario
e) Un espacio para Core de red
f) Bloque de direcciones para clientes

El proveedor puede reservarse el derecho de exigir a sus clientes (solicitantes) una carta solicitando plan actual de direcciones IP, llenar formularios y justificaciones si así lo amerita.

Una práctica importante es no asignar los bloques y las direcciones de manera consecutiva, recordemos que el espacio de IPv6 es enorme y adicionalmente deseamos realizar la implementación de manera segura.

Un pequeño ejemplo (favor revisar el capitulo 4 donde existen prototipos con mayor nivel de detalle):

Escenario:
ISP ACME recibe el siguiente bloque de LACNIC: 2001:db8::/32

Consideraciones:
ACME debe considerar realizar el tradicional subnetting con el objetivo de abastecer a sus clientes y diferentes redes, este subnetting ciertamente puede realizarse de manera libre y como lo desee hacer el operador, sin embargo el objetivo es realizarlo de una manera ordenada y siguiendo las mejores prácticas

Procedimiento:
Las mejores prácticas nos indican que hay que asignar /64 para Loopbacks, /64 para LAN, /64 para WANs, /48 para POPs (entre otros). Lo que vamos a hacer es trabajar con los bits entre /32 y el /48. Es bastante sencillo en realidad. Recordemos que IPv6 está dividido en 8 campos de 16 bits cada uno. Lo que haremos es jugar con una parte de esta nomenclatura. En el ejemplo anterior haremos lo siguiente:

```
[___ NET  ID __     ]   [Subnet]  [Division]   [________  Interface ID ____________]
2001:0db8:0000:0000:0000:0000:0000:0000
   [C1]   [C2]    [C3]    [C4]     [C5]                    [C6]        [C7]      [C8]
```

En este caso lo que haremos es jugar con el tercer campo de ceros (Subnet). Allí tenemos específicamente 16 bits = 65535 subnets que podemos crear para diferentes necesidades. Un plan de direccionamiento puede ser el siguiente:

Plan de direccionamiento (macro):
Para loopbacks:
a) Tomar todo el 2001:db8:00000000::/48
    i. 2001:db8:0:0::1/64 Loopback #1
    ii. 2001:db8:0:1::43/64 Loopback #2
    iii. 2001:db8:0:2::00A7/64 Loopback #3

b) Segmentos LANs:
a. Tomar todo el 2001:db8:000E::/48
    i. 2001:db8:000e:0::/64 Segmento LAN #1
    ii. 2001:db8:000e:23::/64 Segmento LAN #2
    iii. 2001:db8:000e:286::/64 Segmento LAN #3

c) Para WANes
a. Tomar todo el 2001:db8:005a::/48
    i. 2001:db8:005a:0::/64 Segmento WAN #1
    ii. 2001:db8:005a:42::/64 Segmento WAN #2
    iii. 2001:db8:005a:00C2::/64 Segmento WAN #3

d) Para POPs
a. 2001:db8:00d9::/48 Punto de Presencia #1
b. 2001:db8:139::/48 Punto de Presencia #2
c. 2001:db8:02fd::/48 Punto de Presencia #3

1.1.3. ¿Cómo obtener direccionamiento IPv6?

Dependiendo de tu RIR el procedimiento de solicitud de direcciones IPv6 será diferente, siempre es recomendable leer las políticas del mismo antes de llenar cualquier formulario y saber si cumples con los requisitos. Las políticas de asignación de IPv6 de Lacnic se consiguen en su website dentro de la sección 4[2].

Para realizar la solicitud dentro de Lacnic existen dos maneras:

a) Tradicional vía correo electrónico
Si utilizas esta opción debes llenar el formulario[3] de solicitud y enviarlo a hostmaster@lacnic.net

b) Nueva interfaz Web[4]
Esta opción es similar al correo electrónico sin embargo la información debe ser llenada con un navegador dentro del website de Lacnic.

En ambos casos es importante indicar nombre de la organización solicitante, contactos de la entidad, ASN (si posee), dirección, otros. Por defecto Lacnic entrega redes /32 pero es posible solicitar redes mayores si es justificable.

1.2_
Tipos de direcciones

1.2.1. Link local

Las direcciones Link Local (o direcciones de Enlace Local) están definidas en el RFC 4291. Son utilizadas exclusivamente para direccionamiento unicast dentro de un mismo segmento de red, es decir, no son enrutadas, no pasan enrutadores. Su prefijo corresponde a FE8::/10 y su dirección puede ser construida de manera manual, automática derivada de un DHCP, de un algoritmo del SO entre otros. La manera actual es configurar esta dirección IP en base a algún tipo de algoritmo aleatorio para aumentar la seguridad y privacidad del usuario.

Las direcciones Link Local son obligatorias en IPv6 debido a que son utilizadas por otros mecanismos necesarios para su operación como lo es Neighbor Discovery Protocol (NDP) y/o DHCPv6

Tip: Debido a que las direcciones Link Local pertenecen a una interfaz específica del equipo, tengamos presente al momento de realizar un ping6 a un destino FE:80::/64 indicar la interfaz origen (varía según el OS).

1.2.2. Direcciones Globales (Global Unicast)

Las direcciones Globales análogas son las direcciones públicas de IPv4. Es decir, son direcciones que pueden atravesar y encaminarse por los diferentes routers en Internet,

Las direcciones globales actualmente funcionan bajo el siguiente esquema: Los tres primeros bits de una dirección Global Unicast siempre deben comenzar con 001 (RFC 3587). Por ello las direcciones posibles actualmente comienzan con 2000 o 3000:

0010 = 2000 (dirección válida)
0011 = 3000 (dirección válida)
0100 = 4000 (dirección inválida)

1.2.3. ULAs (Unique Local Address)

Las direcciones ULA (RFC 4193) de cierto modo son equivalentes a las direcciones privadas de IPv4 (RFC 1918). La intención de estas direcciones es buscar un direccionamiento que cubra toda una empresa pero a su vez no son direcciones globales. Las direcciones ULAs no deben ser anunciadas a la tabla global de Internet.

Las ULAs vienen a sustituir lo que anteriormente eran las direcciones Site-Local (depreciadas en el RFC 3879). Ocurría que el concepto de Site como tal es un concepto ambiguo y propenso a muchas interpretaciones, por ejemplo, Site es: ¿un piso?, ¿un edificio?, ¿las oficinas en un país?, ¿toda una empresa?. Las ULAs son específicamente creadas para comunicaciones entre dispositivos Internos en un ámbito (generalmente una empresa).

Es importante recordar que un dispositivo puede tener muchas direcciones IPv6 y por ello para comunicarse internamente se utiliza ULA y para comunicarse con el exterior se utilizan las direcciones globales.

Es base a lo anterior, las ULAs pueden ser enrutadas solo dentro de la empresa/entidad, no deben llegar a Internet. En este sentido y gracias a su alto nivel de direcciones únicas es poco probable que existan dos localidades con la misma subred, dicho esto, los problemas de solapamiento que vivimos en IPv4 en redes privadas (ej. Dos localidades con 192.168.1.0/24) no existirían.

El prefijo asignado para ULAs es: fc00::/7

1.2.4. Prefijos especiales

1.2.4.1. Loopback

Se define con ::1/128, utilizado para identificarse uno mismo, se puede utilizar para comunicaciones del mismo OS. Otro uso habitual es para saber si la pila IPv6 esta funcionado. Por ejemplo: ping6 ::1, http://[::1]

1.2.4.2. 6to4

Utiliza el prefijo 2002::/16. Por ello cualquier dirección que veamos con ese prefijo corresponde a este mecanismo de transición

1.2.4.3. Documentación

El prefijo utilizado mundialmente para documentación es 2001:db8::/32. La intención de esta dirección es utilizarla en libros, revistas, documentación, ejemplos, entre otros. Este prefijo no debe ser utilizado en Internet ni como direcciones en nuestra red.

1.2.4.4. Default Gateway (puerta de enlace predeterminada o ruta por defecto)

::/0 este prefijo se utiliza para indicarle al sistema operativo la ruta por defecto. En protocolos de enrutamiento esta ruta también se puede anunciar

1.2.4.5. Teredo

Utiliza el prefijo 2001::/32. Por ello cualquier dirección que veamos con ese prefijo corresponde a este mecanismo de transición

1.2.4.6. Multicast

Las direcciones multicast se encuentran definidas en el prefijo ff00::/12. La lista completa de multicast se encuentra en la página de IANA[5]. Los principales IP multicast son:

ff01::1 - Todos los nodos en el interface local
ff02::1 - Todos los nodos en el enlace local
ff01::2 - Todos los routers en el interface local
ff02::2 - Todos los routers en el enlace local
ff05::2 - Todos los routers en el site-local
ff02::9 - Routers RIP

1.2.4.7. No especificada (unspecified)

Las direcciones no especificadas están definidas con ::/128 (todos ceros). Generalmente se ven durante el arranque de la pila IPv6 y solicitudes DHCPv6.

1.3_
Plan de direccionamiento jerárquico

1.3.1. Consideraciones de Diseño

Como se ha mencionado con anterioridad una de las cualidades de un buen plan de direccionamiento es que siga una esquema jerárquico. Este

esquema de jerarquías permite la agregación de grupos de direcciones lo cual reduce las tablas de enrutamiento. Esto, además de considerarse una de las mejoras prácticas operativas reduce el procesamiento de rutas e incrementa la estabilidad de la red.

En general, un plan de direccionamiento de IPv6 no es muy diferente de su análogo en IPv4. De hecho el lector con experiencia en crear planes de direccionamiento IPv4 encontrará muy familiar el hacer su plan con IPv6.

Para poner un ejemplo imaginemos que en IPv4 tenemos un prefijo /16. Este nos da un total de 256 /24s las cuales pueden estar agrupadas en 16 /20 designados para cada uno de nuestros Puntos de Presencia (POPs). Cada uno de estos POPs contaría con 16 /24s.

De una forma similar en IPv6 podemos tener un /32, el cual podemos dividir en 256 POPs /40, los cuales a su vez cada uno puede tener 256 /48s. Cómo puede verse ambos son muy similares, sin embargo, existen algunas diferencias importantes que describiremos a continuación.

En primer lugar, en la asignación y distribución de direccionamiento IP existen dos principios que entran en conflicto: agregación y conservación. Debido al diminuto espacio de IPv4, la conservación precede a la agregación y en muchas ocasiones terminamos con un direccionamiento fraccionado. A diferencia de IPv4, en IPv6 la agregación precede a la conservación, de tal forma que es posible hacer planes de direccionamiento más eficientes desde el punto de vista de la agregación y más sencillos ya que podemos usar siempre el mismo tamaño de subred.

En segundo lugar, tenemos la diferencia de tamaños de asignaciones. Mientras que en este ejemplo para IPv4 tenemos solo 4 bits, para IPv6 podemos usar hasta 8 bits para agrupar nuestros puntos de presencia, nuestras subredes y las asignaciones a nuestros usuarios finales. En IPv6 cada carácter es representado por una secuencia de 4 bits a la cual llamamos "nibble". Una práctica común y que además recomendamos es hacer los planes de direccionamiento en fronteras de "nibble". Por esa razón nuestro ejemplo segmenta un prefijo /32 en subredes /40 y /48. Más adelante en este capítulo analizaremos esta práctica con más detalle.

Finalmente en IPv4 asignamos direcciones a nuestros usuarios, en IPv6 a los usuarios les asignamos subredes. El tamaño de las subredes en IPv6 puede variar desde un /48 hasta un /64. Más adelante discutiremos las diferentes variables con su pros y sus contras.

1.3.2. Tamaños de Prefijos

Una de los puntos críticos en la creación de un plan de direccionamiento es decidir el tamaño de las subredes a entregar a nuestros usuarios, el tamaño de las direcciones para uso de enlaces punto a punto, loopbacks, etc. Sin embargo, debido a los diferentes paradigmas de diseño en comparación a IPv4, no siempre es sencillo encontrar los valores ideales. A continuación mostraremos algunas consideraciones importantes que esperamos sean útiles para decidir el tamaño adecuado de sus subredes.

1.3.2.1. Subredes de acceso

A diferencia de IPv4 donde las subredes pueden tener un tamaño variable sin mayor problema, en IPv6 en general todas las subredes deben tener una longitud de 64 bits. La razón de usar un prefijo /64 es un requerimiento para el buen funcionamiento de algunos protocolos como lo son el Neighbor Discovery (ND), Secure Neighbor Discovery (SEND), las extensiones de privacidad y Site Multihoming by IPv6 Intermediation (SHIM6) entre otros.

Esto no quiere decir que no puedan usarse subredes con longitudes mayores a 64 bits por ejemplo para enlaces punto a punto, simplemente quiere decir que si llegan a usarse es necesario conocer cuales son las implicaciones de su uso. En las siguientes dos secciones discutiremos el uso de prefijos mayores a 64 bits para enlaces punto y loopbacks.

1.3.2.2. Enlaces Punto a Punto

El uso de subredes con prefijos mayores de 64 bits no es recomendado en IPv6 para el uso general. Sin embargo estos pueden usarse para situaciones especiales donde las direcciones son asignadas manualmente y todos los nodos son enrutadores que no necesitan de las funcionalidades (mencionadas en la sección anterior) que requieren subredes /64 para funcionar.

En el caso de enlaces Punto a Punto no existe una recomendación aceptada universalmente y podemos encontrar una diversidad de alternativas que van desde el uso de un /64, pasando por diferentes tamaños de prefijos como /127, 126, /112 hasta llegar al uso de solo direcciones link-local. A continuación analizamos cada uno de estos tamaños de prefijos con sus respectivas implicaciones.

Prefijos longitud de 64 Bits

El uso de un prefijo de 64 bits para enlaces punto a punto tiene la ventaja de simplificar los planes de direccionamiento y la operación de la red al solo manejar un tamaño de subred tanto para la LAN como para los enlaces. Esto puede reflejarse en una disminución de costos y errores operativos. Otra ventaja que tiene el uso de este prefijo es que no es necesario renumerar en caso de necesitar agregar más nodos, por ejemplo por el cambio de una tecnología punto a punto a multi-acceso.

Una de las principales críticas al uso de este tamaño de prefijo es el "desperdicio" de direcciones, sin embargo el argumento puede verse reducido por la simplicidad que brinda y el enorme espacio de direcciones IPv6. La principal desventaja en el uso de este prefijo es que es vulnerable a ataque de "Neighbor Discovery" (ND)[6]. Para evitar este tipo de ataques es necesario proteger las interfaces del enrutador con listas de acceso y/o que el enrutador tenga los mecanismos de protección definidos en el RFC 6583.

Prefijo de longitud de 126 bits

Este prefijo es el análogo a un prefijo /30 de IPv4. Para muchos administradores de red es el prefijo más natural a usar por su familiaridad con IPv4. A diferencia de un prefijo /64 no presenta problemas de seguridad con ND. Sus desventaja es que es que no es tan simple de usar en planes de direccionamiento como un prefijo /64 y en caso de requerir añadir más nodos a la subred es necesario renumerar.

Prefijo de longitud de 127 bits

El uso de este prefijo es análogo al uso de prefijos /31 en IPv4 y tiene las mismas características de seguridad y usabilidad de un prefijo /126. Sin embargo su uso fue desalentado[7] ya que entraba en conflicto con el Subnet-Router anycast definido en el RFC3627 y podía generar el problema de "ping-pong". Sin embargo la nueva especificación de ICMPv6[8] resuelve este problema y el uso de este tamaño de prefijo ha vuelto a recomendarse siempre y cuando el administrador de la red verifique la compatibilidad del software de los enrutadores con el RFC 6164.

Prefijos intermedios /112, /96

Algunos administradores de red utilizan prefijos intermedios para sin renumerar permitir el crecimiento de nodos dentro de la subred y reducir el impacto de los problemas relacionados con ND. Estos prefijos varían en tamaño y los más comunes en encontrar son /112 y /96.

Uso de direcciones Link-local

Aunque no es una práctica común también es posible no usar direcciones globales para los enlaces punto a punto y solo usar direcciones Link-local. En caso de que estas direcciones sean usadas es necesario utilizar un interfaz loopback con una dirección global como fuente para mensajes de ICMPv6, respuesta a traceroutes, etc.

Entre las ventajas que tiene el uso de Link-local están tener una menor tabla de ruteo, la reducción de posibles ataques (bajo el racional que cada interfaz con una dirección alcanzable localmente es una ataque potencial), una disminución en la complejidad de la configuración y un DNS más simple al tener menos interfaces que agregarse a las zonas reversas.

Entre las desventajas es que las interfaces no responden a pings y traceroutes, y aunque estas puedan responder mediante una interfaz loopback puede existir una pérdida de granularidad en la información al ser difícil saber por cual interfaz está respondiendo el enrutador. Otra desventaja es que la dirección es dependiente del hardware ya que las direcciones Link-local en general usan EUI-64 y cambian cuando la dirección MAC cambia. Otras complejidades existen por el uso de NMS (Network Management Systems) que usan la IP de una interfaz para recolectar información y algunas funciones de MPLS-TE[9].

Uso de /126 y Reserva de /64

Una opción relativamente aceptada como mejor práctica es la reserva de un prefijo /64 para cada interfaz punto a punto y configurar un /126. Esto permite simplificar los planes de direccionamiento al tener subredes homogéneas, poder agregar nodos a la subred sin necesidad de renumerar y al mismo tiempo evitar agregar problemas de seguridad debidos a ND.

1.3.3.2. Loopbacks

Al igual que en IPv4 se recomienda que las loopbacks sean de una longitud 32 bits, en IPv6 analogamente se recomienda que estas sean de longitud de 128 bits.

Existe también la práctica de usar un /64 para cada loopbacks como simplificación de planes de direccionamiento. Sin embargo nosotros consideramos que el uso de loopbacks de longitud de 128 bits agrupadas dentro de un solo prefijo de 64 bits es suficiente para proveer una gran cantidad de direcciones y mantener un plan de direccionamiento simple.

1.3.3.3. Usuarios corporativos

Las redes de usuarios ya sean corporativos (grandes redes) o residenciales se les conoce como "End-sites" o "Sitios Finales". En el pasado se recomendaba que estas redes se les asignara un prefijo de 48 bits, un prefijo de 64 bits o uno de 128 bits[10]. Sin embargo con el tiempo esta recomendación resultó no ser práctica y recientemente ha sido substituida por el RFC 6177 que indica que los "Sitios Finales" deben recibir una asignación correspondiente a su tamaño y necesidad cumpliendo con las siguientes consideraciones:

- Debe ser sencillo para el sitio obtener espacio de direccionamiento para múltiples subredes. Aunque en teoría un /64 puede satisfacer el direccionamiento para un sin fin de dispositivos, los Sitios Finales deben tener la posibilidad de crear múltiples subredes.

- La asignación por defecto debe tener en cuenta que el sitio pueda crecer y evitar los problemas de escasez de IPv4.

- La asignación de prefijos demasiado pequeños muy probablemente incrementará el costo de administrarlos y renumerarlos en el futuro.

- La operación del manejo del DNS reverso y el uso de fronteras de nibble debe considerarse

Las redes corporativas son un subconjunto de los "Sitios Finales". Estas redes pueden llegar a ser bastante complejas, contener una variedad de servicios, centros de datos y múltiples sitios. Por estas razones es necesario que para estas redes se asigne al menos un prefijo de longitud de 48 bits. En caso de que la organización tenga múltiple localidades se recomienda que cada una de ellas tenga al menos un prefijo de 48 bits.

1.3.3.4. Usuarios residenciales

Las redes de usuarios residenciales es otro subconjunto de los "Sitios Finales". A diferencia de las redes corporativas las redes residenciales son mucho más pequeñas y sencillas que las redes corporativas. Aunque el día de hoy un solo prefijo /64 pudiera ser suficiente para estas redes, esto además de no cumplir con las consideraciones de las redes para "Sitios Finales" es ampliamente probable que para el futuro esto no sea suficiente y sea necesario renumerar. Esto lleva a los proveedores de Internet a evaluar el entregar prefijos de una longitud menor para redes residenciales. La pregunta es ¿Cuál es el tamaño de prefijo ideal?

Al día de hoy no existe una recomendación por todos aceptada acerca de cuál es el tamaño ideal de prefijo para una red residencial. Entre las

asignaciones más comunes se encuentran la entrega de prefijos /60, /56 y /48 (nótese que todas son basadas en frontera de "nibble"). La asignación de un prefijo de 60 bits permite el uso de 16 (4 bits) subredes por parte de los usuarios lo cual es un número bastante lógico para las necesidades actuales de redes residenciales. Sin embargo queda la duda si este tamaño será suficiente para las necesidades futuras.

Para evitar renumeraciones futuras, una posibilidad es usar directamente un prefijo de 48 bits. Aunque hoy es difícil imaginar una red hogareña con 65,536 subredes, algunos de los proponentes de un prefijo /48 indican que este tipo de asignaciones no son para hoy sino para ahorrar costos para una posible necesidad en el mañana. A este respecto es importante tomar en cuenta el costo de un tamaño de asignación mayor debido a que para algunos Registros Regionales de Internet (RIR) entre mayor sea el prefijo asignado mayor es el costo.

Una opción intermedia es asignar un prefijo de 56 bits a usuarios residenciales lo cual permite a un usuario llegar a tener hasta 256 subdredes, lo cual es posible que sea suficiente para el corto y mediano plazo. Si el costo de una gran asignación inicial de IPv6 no es un problema, entonces una posibilidad complementaria para la asignación de prefijos /56 es reservar el /48 para un posible uso futuro. Dependiendo de la necesidad futura de los usuarios residenciales esta reserva puede servir para aumentar el tamaño de la asignación sin necesidad de renumerar o para duplicar el número de usuarios con asignaciones /56.

Al día de hoy no existe una recomendación por todos aceptada de cual es el tamaño ideal de prefijo para una red residencial.

1.3.3.5. Infraestructura

Además del direccionamiento para los usuarios es necesario asignar prefijos para uso específicamente de infraestructura como lo son enlaces de interconexión; servicios como DNS, correo, web; dispositivos de monitoreo y seguridad como IDS, colectores de datos y sistemas de monitoreo (NMS) entre otros.

El tamaño de estos prefijos va a depender del tamaño de la organización pero deben cumplir al menos con los siguientes lineamientos básicos:

- Prefijos diferentes para infraestructura interna y pública. Esto permite crear listas de acceso de forma sencilla.

- Prefijos diferentes para enlaces punto a punto y servicios como DNS, correo, NMS, etc.

Como casos específicos es posible que un ISP mediano que recibe un prefijo de 32 bits decida usar 1 /48 para infraestructura pública y un 1 /48 para infraestructura privada. A simple vista parecen muchas direcciones y una estrategia poco eficiente, sin embargo hay que recordar que a diferencia de IPv4, en IPv6 el factor de escasez de direcciones debe influir lo menos posible en nuestras decisiones operativas. En este caso la prioridad de planes de direccionamiento más sencillos es mayor que la de conservación de direcciones.

Para el caso de una organización "Sitio Final" recibiendo un prefijo /48 esta puede optar por definir varios prefijos de 60 bits para el uso de direccionamiento para infraestructura. Más adelante mostraremos unos ejemplos más detallados de estos casos.

1.3.4. Tamaño de Bloque a Solicitar

Dependiendo del tipo de organización dependerá la forma de solicitar un prefijo de IPv6. La mayoría de usuarios finales (residenciales y corporativos) deberán de solicitar sus direcciones de IPv6 a sus respectivos proveedores de Internet. En el caso de los proveedores de Internet y excepcionalmente algunos grandes usuarios finales como universidades, bancos o grandes corporaciones estos pueden solicitar su(s) prefijo(s) de IPv6 a un Registro Regional de Internet.

Para solicitar un prefijo de IPv6 a un RIR la organización debe cumplir con algunos requisitos. Para el caso de Latinoamérica y el Caribe el prefijo debe solicitarse a LACNIC y entre los requisitos generales se encuentran:

- Plan de direccionamiento
- Pago de asignación
- En el caso de organizaciones "Sitio Final" tener un prefijo IPv4 Independiente de Proveedor o ser multi-proveedor

Para ISPs la mínima asignación es un /32 y no tiene límite máximo. Para organizaciones "Sitio Final" la asignación mínima es un /48 y la máxima un /32. Los detalles de requisitos y costos se pueden encontrar en[11] y [12].

1.4_
Ejemplos

A continuación se describen tres ejemplos de planes de direccionamiento: ISP Pequeño Mediano, ISP Multi-Regional y para una Red Corporativa o Universidad. Una herramienta que simplifica mucho la creación de planes de direccionamiento es Sipcal[13] la cual es gratuita y solo requiere de su compilación usando gcc.

Usando sipcalc pudueden obtenerse todos los prefijos /48 bits de un prefijo /32 con el siguiente comando:

```
sipcalc 2001:db8::/32 -s /48
```

Si se quiere obtener más información por prefijo se puede usar la bandera -u.

1.4.1. Plan de direccionamiento para ISP Mediano - Pequeño

El ISP "Estrella del Sur" tiene presencia en 3 tres ciudades diferentes las cuales están interconectadas con sus enlaces propios lo que le permite

tener tránsito de Internet local y al mismo tiempo tener un respaldo en la ciudad de su sede central. En su sede central cuenta con 500 usuarios corporativos y 25,000 usuarios de banda ancha fija. En las otras dos ciudades tiene en cada una 200 usuarios corporativos y 14,000 usuarios residenciales de banda ancha.

Estrella del Sur ha recibido un prefijo /32 (2001:db8::/32) de su Registro Regional de Internet. Su plan es entregar prefijos /48 para usuarios corporativos y /56 para usuarios residenciales. Además planea dividir el prefijo /32 en 16 prefijos /36 de los cuales asignará 4 a cada ciudad y reservará los primeros 4 bloques para uso futuro y para infraestructura. Se presenta el plan de numeración (por simplicidad se omiten algunos bloques

```
2001:db8::/36         Reservado para infraestructura
2001:db8:1000::/36    Reservado
2001:db8:2000::/36    Reservado
2001:db8:3000::/36    Reservado
2001:db8:4000::/36    Sede Central Bloque 1
2001:db8:5000::/36    Sede Central Bloque 2
2001:db8:6000::/36    Sede Central Bloque 3
2001:db8:7000::/36    Sede Central Bloque 4
2001:db8:8000::/36    Ciudad 1 Bloque 1
2001:db8:9000::/36    Ciudad 1 Bloque 2
…
2001:db8:e000::/36    Ciudad 3 Bloque 3
2001:db8:f000::/36    Ciudad 3 Bloque 4
```

De los 4 /36 de cada ciudad Estrella del Sur también decidió usar uno para usuarios residenciales y otro para usuarios corporativos. Tanto para usuarios corporativos como usuarios residenciales Estrella del Sur decide subdividir el /36 en prefijos /40 para sus puntos de presencia (POPs) en la ciudad (para un total de 16 POPs).

```
Sede Central Bloque 1 POPs
2001:db8:4000::/40
2001:db8:4100::/40
2001:db8:4200::/40
…
2001:db8:4d00::/40
2001:db8:4e00::/40
2001:db8:4f00::/40
```

De una forma similar se divide el bloque para usuarios corporativos (2001:db8:5000::/36) y los bloques análogos de las otras ciudades (2001:db8:8000::/36, 2001:db8:9000::/36, etc.).

Los prefijos /40 para usuarios residenciales se subdividen en prefijos /56 para un total de 65,536 usuarios. Para usuarios corporativos se divide en

/48 para un total de 256 de estos. El plan de direccionamiento se muestra a continuación:

```
Usuarios residenciales
2001:db8:4000::/56
2001:db8:4000:100::/56
2001:db8:4000:200::/56
2001:db8:4000:300::/56
…
2001:db8:40fe:4800::/56
2001:db8:40fe:4900::/56
2001:db8:40fe:4a00::/56
…
2001:db8:40ff:fd00::/56
2001:db8:40ff:fe00::/56
2001:db8:40ff:ff00::/56

Usuarios Corporativos
2001:db8:5000::
2001:db8:5001::
2001:db8:5002::
2001:db8:5003::
…
2001:db8:50fc::
2001:db8:50fd::
2001:db8:50fe::
2001:db8:50ff::
```

Finalmente se aclara que Estrella del Sur decidió usar el primer /64 de cada /56 y cada /48 asignado a sus clientes para interconexión.

1.4.2. ISP Multi-regional

Si eres un ISP multi-regional y te encuentras en varios países probablemente cuentas con un bloque mayor a /32, quizás un /28 u otro valor. Solo como manera de ejemplo vamos a trabajar suponiendo que recibiste una red /28 de tu RIR.

Escenario:

- El ISP está en 8 países
- En cada país cuentas con 6 POPs (Point of Presence)
- El ISP recibe el prefijo 3001:20::/28 de Lacnic (no usaré como ejemplo el prefijo de documentación porque necesitamos escribir mascarás mayores a /32)

Consideraciones:

- Considerar que el ISP se expandirá a 15 países en los próximos dos años
- Duplicará el número de POPs en cada país en los próximos 3 años.

¿Cómo realizar el plan de direccionamiento?:
Lo primero que haremos es evaluar el bloque recibido, evaluar el tamaño de la red actual y ubicarnos en el crecimiento estimado, no queremos tener que cambiar el plan de direccionamiento más adelante y mucho menos en una red ya en funcionamiento.

Pasos:

1) Estudiar la red recibida por el RIR:
   3001:20::/28 = 3001:20:0:0:0:0:0:0

Del prefijo anterior podemos trabajar específicamente con todo lo que está en naranja (a partir del 0 después del 2):

3001:0020:0:0:0:0:0:0
Rango de red: 3001:20::/28 → 3001:2f::/28

2) Estudiar el número de países que deseo cubrir. En nuestro caso 15 países (recordemos que el ISP se va a expandir). Necesito 4 bits para cubrir esta demanda.

3) Asignar una subred a cada país.

De la subred 3001:0020:0:0:0:0:0:0 vamos a utilizar el primer valor naranja (4 bits) y allí asignaré un país (ahora relleno en azul).

```
3001:0021:0:0:0:0:0:0/32     País número 1
3001:0022:0:0:0:0:0:0/32     País número 1
3001:0023:0:0:0:0:0:0/32     País número 3
3001:0024:0:0:0:0:0:0/32     País número 4
3001:0025:0:0:0:0:0:0/32     País número 5
3001:0026:0:0:0:0:0:0/32     País número 6
3001:0027:0:0:0:0:0:0/32     País número 7
3001:0028:0:0:0:0:0:0/32     País número 8
(y así hasta el último país que sería 3001:002f:0:0:0:0:0:0/32,
nótese que el 0 para el país no lo estamos utilizando sin embargo es
perfectamente usable)
}
```

4) Asignar bloques a los POPs en cada país. Vamos a realizar el ejemplo con un solo país:

a. Ya tenemos un /32 en el país 3 el cual es: 3001:0023:0:0:0:0:0:0.
b. Vamos a utilizar el tercer campo de la dirección IP (los bits 32 al 47). Es decir, tenemos 65535 POPs que podemos asignar en este país.
c. Recordar que asignaremos /48 al POP que es una mejor práctica (si es necesario lo ajustas a tu necesidad y/o realidad)
d. Pasos:

i. 3001:0023:0:0:0:0:0:0 tomaremos el primer campo anaranjado (el tercer campo, ahora en blanco). De allí tomaremos 12 bloques de manera aleatoria no consecutivos. Ejemplo:

```
1. 3001:0023:002a:0:0:0:0:0/48  POP #1
2. 3001:0023:009b:0:0:0:0:0/48  POP #2
3. 3001:0023:010d:0:0:0:0:0/48  POP #3
4. 3001:0023:017c:0:0:0:0:0/48  POP #4
5. 3001:0023:026b:0:0:0:0:0/48  POP #5
6. 3001:0023:03ba:0:0:0:0:0/48  POP #6
7. 3001:0023:02df:0:0:0:0:0/48  POP #7
8. 3001:0023:0319:0:0:0:0:0/48  POP #8
9. 3001:0023:07ba:0:0:0:0:0/48  POP #9
10. 3001:0023:03d5:0:0:0:0:0/48 POP #10
11. 3001:0023:03f3:0:0:0:0:0/48 POP #11
12. 3001:0023:0457:0:0:0:0:0/48 POP #12
```

Consejos adicionales:
Dependiendo de tu experiencia, de tu red, topología y otros aspectos, existen ciertos detalles que puedes considerar al momento de construir tu plan de direccionamiento IPv6. La intención es tener orden, y en esta ocasión facilitar la ubicación de problemas de red y acelerar el troubleshooting. Ya hemos visto lo que sería la manera tradicional y siguiendo las mejores prácticas, sin embargo, con IPv6 (y un poco con IPv4) podemos hacer otras cosas muy simpáticas.

Por ejemplo: la empresa tiene presencia en Argentina, Colombia y Venezuela, sus códigos de país son: 54, 57 y 58 respectivamente. En los ejemplos pasados pudimos haber hecho lo siguiente:

Bloque: 3001:0020:0:0:0:0:0:0/28:
Argentina: 3001:0020:54:0:0:0:0:0/48
Colombia: 3001:0020:57:0:0:0:0:0/48
Venezuela: 3001:0020:58:0:0:0:0:0/48

En caso de que no queramos ser tan agresivos se puede asignar el país como lo hicimos de manera inicial pero en su defecto utilizar el tercer campo para asignar la ciudad/estado/provincia según el código interno del país. En el supuesto de la ciudad de Caracas en Venezuela, el cual tiene como código 212 algo como esto sería viable:

Venezuela: 3001:0020:212:0:0:0:0:0/48 ó
Venezuela: 3001:0020:58:212:0:0:0:0/48 (aquí romperíamos los BCPs)

La intención es poder ubicar redes y fallas mucho más fácil, si durante el troubleshooting identificamos el código del país y/o el estado podremos acelerar la resolución de problemas.

Un ejemplo a detalle:

Bloque:
3001:26::/32 (pais)

Escenario:
Presencia en dos estados:

Procedimiento:
Tomar los bits entre /32 y /48 utilizaremos estos bits para definir el estado/provincia dentro del país:

- 8 bits para el estado/provincia
- 8 bits para el POP/Oficina del ISP dentro del estado/provincia

En el ejemplo anterior:
3001:26::/32 (pais)

Tomaremos un /40 (para el estado, es decir 8 bits para el estado/provincia) y /48 para oficinas y Data Centers

3001:0026:0000:0000:0000:0000:0000:0000
El naranja identifica el estado
00= Estado 1
01= Estado 2

Verde identifica el numero de Oficina/Data Center
00= Oficina 1
01= Oficina 2

En este sentido, supongamos dos oficinas dentro del estado 1

Oficina 1 (en el estado 1):
IP Address: 3001:0026:0000:0000:0000:0000:0000:0000
Network range: 3001:0026:0000:0000:0000:0000:0000:0000-
3001:0026:0000:ffff:ffff:ffff:ffff:ffff

Oficina 2 (en el estado 1):
IP Address: 3001:0026:0001:0000:0000:0000:0000:0000
Network range: 3001:0026:0001:0000:0000:0000:0000:0000-
3001:0026:0001:ffff:ffff:ffff:ffff:ffff

En el estado 2:
Oficina 1 (en el estado 2):
IP address: 3001:0026:0100:0000:0000:0000:0000:0000
Network range: 3001:0026:0100:0000:0000:0000:0000:0000-
3001:0026:0100:ffff:ffff:ffff:ffff:ffff

Oficina 1 (en el estado 2):
IP address: 3001:0026:0101:0000:0000:0000:0000:0000
network range: 3001:0026:0101:0000:0000:0000:0000:0000-
3001:0026:0101:ffff:ffff:ffff:ffff:ffff

De cada /48 de estas se pueden tomar las /64 para loopbacks, WANs, LANs, etc. Como lo hemos visto anteriormente.

1.4.3. Plan de direccionamiento para Red Corporativa o Universidad

La "Universidad para Un Futuro Mejor" está implementando IPv6 y a continuación se presenta el plan de direccionamiento con el cual solicitará un prefijo /44.

La universidad tiene presencia en 5 ciudades. En su sede central tiene 3 campus con diversos números de facultades o edificios en cada uno. En las demás ciudades solo tiene 1 campus por ciudad. Su plan es dividir el /44 en 16 /48s. Suponiendo que recibirá un prefijo de la forma 2001:db8:420::/44 su plan de direccionamiento es como sigue:

```
2001:db8:420::/48          Servicios Públicos e Infraestructura
2001:db8:421::/48          Servicios Privados e Infraestructura
2001:db8:422::/48          Reservado
2001:db8:423::/48          Reservado
2001:db8:424::/48          Campus 1 Sede Central
2001:db8:425::/48          Campus 1 Sede Central
2001:db8:426::/48          Campus 2 Sede Central
2001:db8:427::/48          Campus 3 Sede Central
2001:db8:428::/48          Campus Ciudad 2
2001:db8:429::/48          Reservado
2001:db8:42a::/48          Reservado
2001:db8:42b::/48          Reservado
2001:db8:42c::/48          Campus Ciudad 3
2001:db8:42d::/48          Reservado
2001:db8:42e::/48          Reservado
2001:db8:42f::/48          Reservado
```

El racional para asignar al Campus Ciudad 3 el prefijo 2001:db8:42c::/48 y reservar el 2001:db8:429::/48 es para poder facilitar el crecimiento de bloques contiguos del Campus Ciudad 2. Aunque estos prefijos contiguos no están en frontera de nibble el agruparlos simplifica la operación de la red.

1.5_
Lecturas recomendadas

A continuación presentamos una lista (no exhaustiva) de las lecturas de documentos de estándares y recomendaciones operativas relacionadas con el direccionamiento de IPv6 que sugerimos que sean consultadas por el lector:

RFC 5375 IPv6 Unicast Address Assignment Considerations
RFC 4291 IP Version 6 Addressing Architecture
RFC 6177 IPv6 Address Assignment to End Sites
RFC 6164 Using 127-Bit IPv6 Prefixes on Inter-Router Links
RFC 5952 A Recommendation for IPv6 Address Text Representation

RFC 3587 IPv6 Global Unicast Address Format
RFC 419 Unique Local IPv6 Unicast Addresses
RFC6583 Operational Neighbor Discovery Problems
RFC4192 Procedures for Renumbering an IPv6 Network without a Flag Day
draft-ietf-opsec-v6 Operational Security Considerations for IPv6 Networks
draft-ietf-opsec-lla-only-03 Using Only Link-Local Addressing Inside an IPv6 Network
draft-ietf-v6ops-ula-usage-recommendations Recommendations of Using Unique Local Addresses
draft-ietf-v6ops-enterprise-incremental-IPv6 Enterprise IPv6 Deployment Guidelines
draft-ietf-v6ops-design-choices Design Choices for IPv6 Networks

1.6_
Referencias

[1] http://www.icann.org/en/resources/policy/global-addressing/allocation-IPv6-rirs

[2] http://www.lacnic.net/web/lacnic/manual-4

[3] http://lacnic.net/templates/isp-v6-template-sp.txt

[4] https://solicitudes.lacnic.net/sol-user-web/login/language/sp

[5] http://www.iana.org/assignments/IPv6-multicast-addresses/IPv6-multicast-addresses.xhtml

[6] Gashinsky I. et al. RFC6583 Operational Neighbor Discovery Problems, 2012.

[7] Savola P. RFC3627 Use of /127 Prefix Length Between Routers Considered Harmful, 2003.

[8] Kohno M et al. RFC6547 Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification, 2011.

[9] Behringer, M. Using Only Link-Local Addressing Inside an IPv6 Network, 2013.

[10] IAB RFC3177 IAB/IESG Recommendations on IPv6 Address Allocations to Sites, 2001

[11] Solicitando bloques IPv6 para proveedores (ISP), LACNIC http://www.lacnic.net/web/lacnic/IPv6-isp 2013

[12] Solicitando bloques IPv6 Usuario Final, LACNIC http://www.lacnic.net/web/lacnic/IPv6-end-user 2013

[13] Sipcalc, http://www.routemeister.net/ 2013

.2

Monitoreo en IPv6

2.1_
Importancia del monitoreo en la red

El monitoreo de la red y de los servicios que hay implementados sobre ella, cobran mas importancia cuanto mas críticos nos resultan estos servicios o vínculos de la red. Eso dependerá fuertemente del tipo de red de la que hablemos. Por ejemplo: quizás no requiera para cada uno de nosotros el mismo énfasis monitorear la red hogareña, que tener total conocimiento de lo que sucede en la red para la cual trabajamos, y que por ejemplo, presta servicio a terceros. La criticidad o no del monitoreo dependerá del grado de control que queramos llevar sobre los servicios. No obstante, mas allá de esta medida que podría resultar hasta subjetiva, lo cierto es que realizar un buen monitoreo no solo nos permite sentir que tenemos controlada la situación, sino que objetivamente permite, entre otras cosas:

- Detectar y prevenir problemas
- Diagnosticar causas de fallas
- Determinar las acciones que solucionarán el problema.
- Conformar planes de contingencia

Todas estas ventajas cobran sentido cuando el monitoreo de la red está hecho en forma responsable, tratando de cubrir todas las variables posibles. En el contexto de este libro, implica que si queremos realizar correctamente el monitoreo de nuestra red, no solo deberemos hacerlo para IPv4, sino que además deberemos incluir lo propio también para IPv6, cuando de redes DualStack[1] se trata

2.2_
Acerca de este capítulo

El objetivo de este capítulo es abordar el tema de monitoreo en IPv6, intentando cubrir todos las variables posibles que conllevan a realizarlo eficazmente, pero utilizando las denominadas herramientas *"Open-Source"* (Código abierto). Se intentará, en pocas páginas, resumir los aspectos mas relevantes de este tipo de herramientas, concentrándonos en las que son mas comúnmente utilizadas, y siempre manteniendo el foco en su soporte IPv6.

El capítulo intentará además, reunir la información pertinente, basada en diversas fuentes, pero que en general encontramos dispersa, con lo cual se dificulta nuestra tarea a la hora de elegir las herramientas que mas convienen a nuestro propósito. Lograr reunirlas en un mismo lugar nos ahorrará tiempo y facilitará la tarea de selección.

2.3_
Para tener en cuenta

Si el lector esta buscando una guía que le permita evaluar las herramientas de monitoreo en si mismas, este capítulo lamentablemente no podrá ayudarlo. Evitaremos poner en tela de juicio si determinada herramienta presenta ventajas o no respecto a su utilización. Nos concentraremos en verificar si dará resultados positivos su implementación en un escenario con IPv6, y qué se debe considerar para lograrlo.

Por otro lado, si de herramientas OpenSource se trata, la cantidad disponible y al alcance de cualquier buscador, es incontable. Por tal motivo, este capítulo no pretende analizar la totalidad de éstas, sino un escaso subconjunto, basándonos en las mas utilizadas en la actualidad en los ambientes de los operadores de red.

En el marco de lo expuesto anteriormente, este capítulo no pretende marcar una tendencia sobre las herramientas a utilizar para lograr los objetivos que pretende una red bien monitoreada. Análogamente, tampoco pretende disuadirlo en el uso o no de determinadas utilidades que podrían ayudarle en la tarea de control de la red.

Este capítulo no pretende marcar una tendencia sobre las herramientas a utilizar para lograr los objetivos que pretende una red bien monitoreada.

2.4_
Primera clasificación de herramientas de monitoreo

2.4.1. Contadores de tráfico

Llamaremos de esta forma a las herramientas de monitoreo que nos permiten visualizar la carga de tráfico que atraviesa un determinado dispositivo. Estas herramientas solo contabilizan el tráfico en unidades de bits o bytes por segundos. No determinan el origen o destino del flujo que atraviesa la interfaz, ni mucho menos detectan el tipo de tráfico, sino que se trata solo de una medida interpretada en el tiempo.

Ahora bien, hay diferentes formas de medir esta carga de tráfico que atraviesa un dispositivo, pasaremos a describir algunas de los protocolos y/o herramientas que logran esta medición:

2.4.1.1. SNMP:

Esta sigla proviene del ingles: Simple Network Management Protocol[2]. Las versiones de SNMP más utilizadas son SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2).

Podría considerarse que utilizar SNMP para monitorear el tráfico que atraviesa determinado dispositivo, así como también la utilización de la

CPU, consumo de memoria RAM, entre otros parámetros, es una forma simple de llevar a cabo un estudio de la situación de la red.
El protocolo SNMP utiliza un servicio no orientado a la conexión (UDP). Deberá ser el equipo que estemos monitoreando el que deba soportar SNMP en IPv6 y no el SNMP en sí mismo.

Algunas herramientas que utilizan SNMP para obtener datos son por ejemplo: MRTG y CACTI, las cuales serán comentadas mas adelante.

2.4.1.2. NetFlow:

Conocer la cantidad de tráfico que atraviesa una interfaz o que ingresa y egresa de un dispositivo es una fuente de información que nos ayudara a detectar anomalías. Sin embargo, muchas veces esto representa poco a la hora de querer ahondar en la búsqueda de un diagnóstico.

NetFlow es un protocolo que fue originalmente desarrollado por Cisco para obtener información sobre tráfico IP. Sin embargo, mas allá de su origen, Netflow se ha convertido en un estándar para el monitoreo del tráfico en la red[3].

A diferencia de SNMP, NetFlow permite obtener mas información además de la carga de tráfico en la interfaz, como por ejemplo direcciones origen y destino o los protocolos de capas superiores que atraviesan la interfaz.

Ahora bien, a la hora de utilizar NetFlow para la recolección de datos que aporten información acerca del tráfico IPv6, habrá que tener en cuenta que las herramientas a utilizar deberán soportar NetFlow 9, pues las versiones anteriores no permiten la exportación de flujos de IPv6.

2.4.1.3. SFLow e IPFIX:

Como mencionamos, NetFlow fue creado por la empresa Cisco. SFlow e IPFIX son los estandares del IETF, derivados de NetFlow 9.

Básicamente IPFIX esta orientado a recolectar información según el *flujo* de datos, siendo un protocolo flexible y muy extensible, tal como se puede afirmar al ver la cantidad de RFCs que lo han ido siguiendo (RFCs: 3917, 3955, 5101, 5103, entre otros...)

En cuanto a sFlow, es muy similar, pero se diferencia orientándose mas que nada a la recolección de datos según la información que suministran los *paquetes*. El estándar en el cual se basa es el RFC 3176.

2.4.2. Monitores de servicios y equipamiento:

Mas allá del tráfico que atraviesa las interfaces de los dispositivos, nos interesa conocer otras variables que podrían alterar el funcionamiento de la red. Nos referimos a cuestiones tales como:

- Estado de los servicios y aplicaciones
- Actividad de los hosts
- Temperatura del equipamiento
- Etc

En general, este tipo de herramientas utilizan los denominados "plugin" para obtener la información sobre cada uno de los parámetros como los que mencionamos. Serán entonces estos plugins los que habrá que tener en cuenta para que los parámetros monitoreados nos puedan aportar información sobre IPv6 y/o IPv4, y no la herramienta en sí misma.

Un ejemplo de este tipo de utilidades es NAGIOS, y por ello daremos algunos detalles mas adelante.

2.4.3. Analizadores de tráfico

Clasificamos en esta instancia a aquella herramientas que nos permiten ver el tipo de tráfico que atraviesa la red y los dispositivos. Siempre con el objetivo de hacer mas eficiente la administración de nuestra red, conocer las particularidades del flujo de datos nos facilitará la tarea de prevenir y diagnosticar problemas.

Este tipo de herramientas parecieran ser las mas complejas a la hora de recolectar datos, pues nos da la sensación de que nos encontramos frente a la mayor fuente de información cuando detectamos el *tipo* de tráfico y no solo la *cantidad* o su origen y destino, o simplemente su presencia. Sin embargo, uno de los principales motivos de no hallar respuesta cuando investigamos un problema en la red, es no ahondar de la misma manera en IPv4 que en IPv6, aun contando con analizadores de tráfico.

Analizadores de uso común que podemos considerar: FlowTools, Etheral, NTOP, entre otros.

2.5
Algunos ejemplos de herramientas mas comunes

2.5.1. MRTG

MRTG *(Multi Router Traffic Grapher)* es una herramienta escrita en los lenguajes C y Perl. Se trata de una utilidad que se encuentra entre las que clasificamos como "contadores de tráfico", pues como ya hemos mencionado, logra mostrar el comportamiento y cantidad de tráfico a lo largo del tiempo, utilizando el protocolo SNMP para la recolección de datos. Una vez obtenidos los datos, confecciona informes que se pueden visualizar a través de un browser[4].

Es importante aclarar que también es posible que muestre otro tipo de datos, como la carga de CPU a lo largo del tiempo. Sin embargo, el uso mas común, y para el cual fue desarrollado MRTG, es para conocer

la carga en las interfaces, principalmente en equipos routers, pudiendo diferenciar claramente entre la entrada y la salida del tráfico.

De esta forma es como se ve un reporte generado por MRTG, donde el tráfico entrante a la interfaz es el que se señala en color verde, y la linea azul indica la cantidad de tráfico saliente:

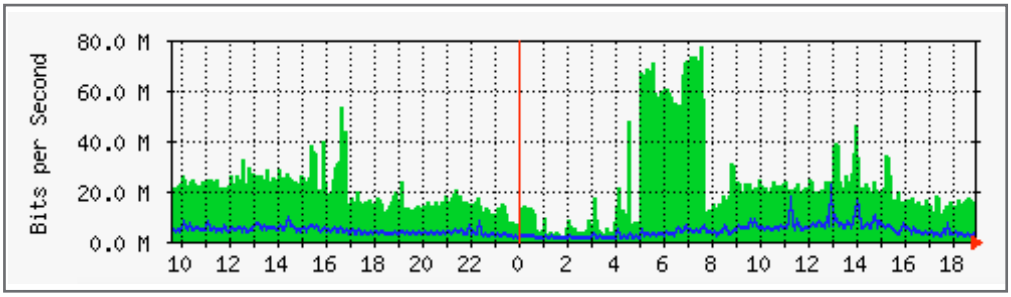

FIGURA 1: GRÁFICO EJEMPLO OBTENIDO DEL SITIO OFICIAL DE MRTG[5]

Además nos permite contar con información acerca de los puntos máximos y mínimos de tráfico alcanzados, promedios, etc.

Mas allá de la herramienta en sí misma, recordemos que nuestro foco es IPv6, por lo que debemos ver las particularidades del caso.

Desde hace varios años MRTG tiene soporte para IPv6, mas precisamente desde la creación de la versión 2.10.0 (en todos sus realeses). Incluso, para aquellos que tenían la versión 2.9.29, se creo un patch para el soporte IPv6.

Asimismo, será necesario contar con dos librerías muy importantes para habilitar el soporte IPv6: Socket6 e INET6, al mismo tiempo que en el archivo de configuración CFGMAKER se deberá declarar la siguiente variable: --enable-ipv6.

Ahora, solo resta que el equipo a monitorear soporte SNMP en IPv6, por lo que, lo que habrá que constatar es la versión de sistema operativo disponible, y sus cualidades respecto a SNMP. Un aspecto importante es que MRTG, fiel a su característica de ser un contador de carga de tráfico, no discrimina entre paquetes de datos IPv6 y paquetes de datos IPv4. Ambas versiones de paquetes IP serán parte de la carga de tráfico sobre la interfaz.

2.5.2. CACTI

Si de “contadores de tráfico” se habla, CACTI se encuentra entre las mas populares. Es muy similar a MRTG, en cuanto a que logra coleccionar (a través de una base de datos MySQL) un conjunto de datos para construir gráficos que muestran la carga de tráfico, tanto entrante como saliente, de una interfaz.

Un gráfico típico de CACTI, puede verse como sigue:

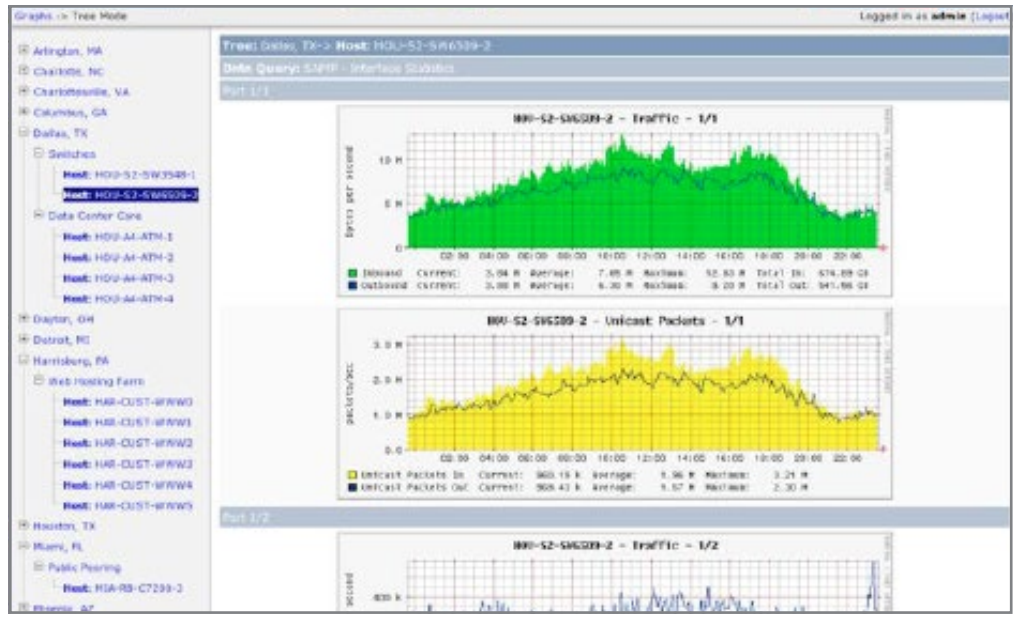

FIGURA 2: GRÁFICO EJEMPLO OBTENIDO DEL SITIO OFICIAL DE CACTI[6]

En cuanto a IPv6, a diferencia de MRTG, no será necesario configurar ninguna variable para que recolecte los datos con la nueva versión del protocolo IP, pero si habrá que tener en cuenta que la versión a utilizar sea 0.8.6 o posterior.

También habrá que tener en cuenta que tampoco podremos diferenciar al tráfico que pasa por la interfaz, si es IPv6 o es IPv4. Solo tendremos esa posibilidad en los casos en que configuremos nuestros equipos de red con interfaces separadas para cada versión del protocolo IP.

2.5.3. NAGIOS

Esta herramienta es una de las que claramente clasificamos dentro de los llamados: "Monitores de Servicios y Equipamiento". Es muy utilizada en la comunidad de los administradores de red, puesto que logra monitorear la actividad de los hosts, las aplicaciones, los servicios y hasta la temperatura de los equipos, todo a través de un conjunto de plugins.

De la misma manera, el soporte para IPv6 viene dado por un grupo de estos plugins, que pueden instalarse a partir de la versión 1.0 de NAGIOS, sin que la instalación básica de éste requiera nada especial para dar el soporte.

La configuración de los recursos a monitorear se hace a través de un grupo de comandos. Estos podrían llegar a ser distintos según queramos realizar el monitoreo sobre recursos IPv4 o IPv6. Asimismo, NAGIOS permite diferenciar los servicios según las diferentes versiones del protocolo IP.

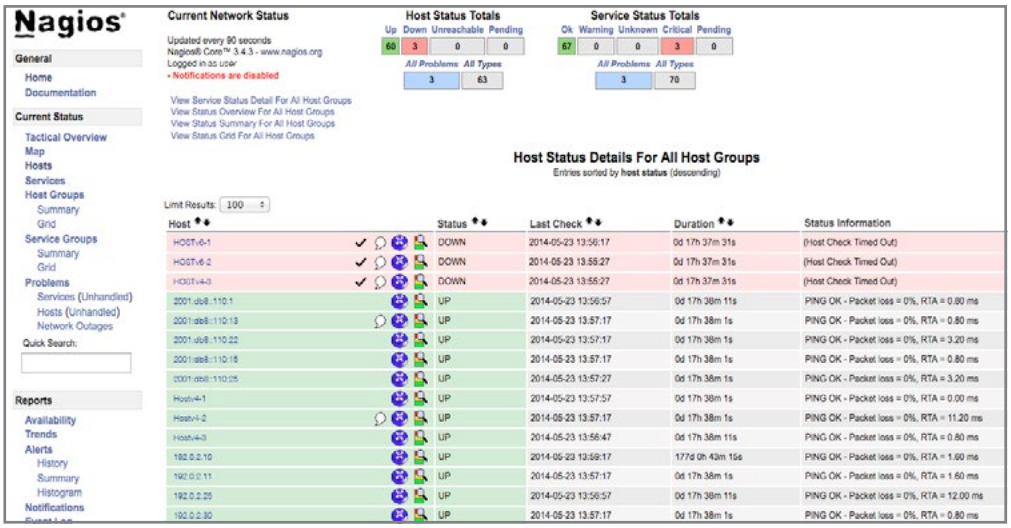

FIGURA 3: GRÁFICO EJEMPLO OBTENIDO DEL SITIO OFICIAL DE NAGIOS

2.5.4. FlowTools

Se trata de un conjunto de herramientas (disponible como paquete Debian) que utilizan NetFlow para recolectar, procesar y generar reportes. Según nuestra clasificación, los consideramos dentro de los "analizadores de tráfico", ya que posee un gran número de opciones configurables que permiten analizar el tráfico por servicios, subredes, sistemas autónomos (ASNs), grupos de direcciones IPs, etc.

Como este conjunto de herramientas utiliza NetFlow para la recolección de información, solo podrá contarse con el soporte de IPv6 si la versión de este último ya es la 9. Sin embargo, hasta la última versión estable de FlowTools esto no era posible.

Bajo este escenario, Flowtools no podrá ofrecernos soporte IPv6 para el análisis del tráfico, por lo que debemos tenerlo en cuenta al momento de monitorear nuestra red.

2.5.5. NTOP

También lo encontramos dentro del conjunto de los "analizadores de tráfico". Se trata de una herramienta que muestra la utilización de la red monitoreada, diferenciando entre direcciones fuente, destino, protocolos utilizados, servicios, etc. Podría verse muy similar a FlowTools, pero con la diferencia que NTOP no trabaja a través de la recolección de flujos y procesamiento de datos en un dispositivo de almacenamiento, sino que muestra en forma instantánea lo que logra monitorear, o sea, no actúa como un colector. Adquiere la información haciendo *sniffing* o utilizando NetFlow, el cual por supuesto, debe considerarse desde la versión 9, tal como ya hemos comentado.

Respecto a la detección de información sobre IPv6, logra hacerlo a partir de la versión de NTOP 3.0. Cualquier implementación a partir de esta versión no requiere ningún tratamiento especial para el soporte en IPv6.

2.5.6. Ethereal/Wireshark

Al mismo conjunto que venimos mencionando pertenece Ethereal, el cual logra capturar el tráfico de las interfaces y analizar los paquetes diferenciando protocolos, también a través de *sniffing*.

Ethereal soporta IPv6 desde versiones anteriores a la 0.9.16, sin embargo, a partir del año 2006 esta herramienta pasa a denominarse Wireshark, por lo que en la actualidad debe ser buscada como tal.

Wireshark definitivamente soporta IPv6, pero es importante tener en cuenta que la versión de *libpcap*[7] que tengamos en nuestro sistema operativo soporte IPv6, si no, no podremos contar con el soporte en Wireshark. Al instalar libpcap entonces, debemos verificar que el proceso de instalación habilite IPv6, pues en muchas versiones ésto no se realiza por defecto, y en ese caso deberemos habilitarlo manualmente.

Otro punto a tener en cuenta es que, originalmente, la resolución reversa para las direcciones IPv6 se encuentra habilitada por defecto, por lo que si esta resolución no esta correctamente configurada podríamos no obtener lo que buscamos.

2.5.7. MTR

Combina las funciones de ping y traceroute en una misma herramienta de diagnóstico. Se puede obtener el resultado con direcciones IPv6 a partir de la versión estable 0.69.

En algunas versiones, al ejecutarlo habrá que forzar el uso de IPv6 a través de la opción: -6. No obstante, si utilizamos nombres en vez de direcciones IP, en la mayoría de las nuevas versiones MTR intentará resolver las direcciones IPv6 antes que IPv4, lo que significa que si no tenemos correctamente configurados los registros AAAA podríamos incurrir en demoras.

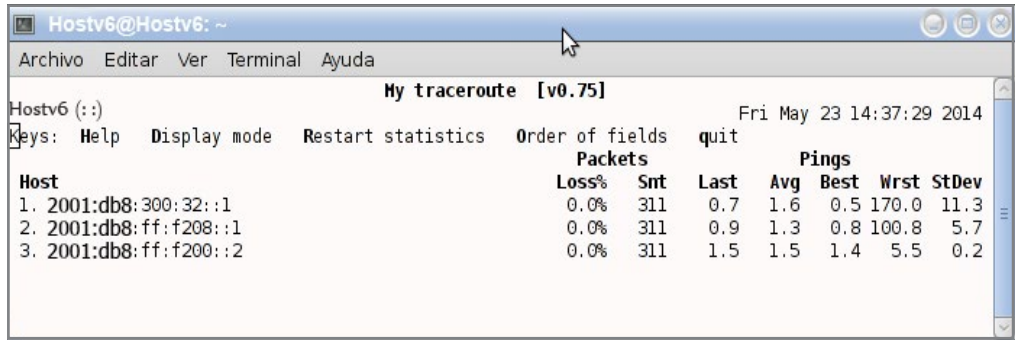

FIGURA 4: GRÁFICO EJEMPLO OBTENIDO DE MTR CON IPV6

2.6_
Conclusiones

En este capítulo hemos podido manifestar la criticidad que representa el monitoreo en una red, sus ventajas y la importancia de que se realice responsablemente, considerando todas las variables en juego e incluyendo no solo lo concerniente a IPv4, sino también lo que atañe a IPv6, cuando tratamos con redes Dual Stack.

Hemos acordado que la elección de las herramientas a utilizar debe ser cuidadosa, y que de ellas hay una gran variedad: desde las que tan solo contabilizan el tráfico en una interfaz, hasta las que nos muestran el contenido de los paquetes y sus encabezados, pudiendo discernir entre protocolos, versiones, servicios, etc.

La complejidad de las herramientas no es lo importante, sino que el resultado de lo que arrojen sea lo que estamos buscando.

El capítulo mostró además un vasto conjunto de utilidades opensource, solo para dejar de manifiesto que a la hora de ponerlas en marcha, cada una de ellas tendrá cierta particularidad que debemos tener en cuenta: desde versiones hasta parámetros de configuración, y todo esto para asegurarnos que no estamos dejando librado al azar ninguna variable no solo de IPv4, sino tampoco de IPv6.

Tomar en cuenta lo dicho nos ayudará a administrar con responsabilidad nuestra red, pues experimentar con la nueva versión del protocolo IP podría resultar atrayente, pero también muy problemático si no consideramos el Monitoreo en IPv6.

2.7_
Referencias

[1 E. Nordmark, R. Gilligan, RFC 4215: Basic Transition Mechanisms for IPv6 Hosts and Routers, IETF. Ver en: http://tools.ietf.org/html/rfc4213

[2] Simple Network Management Protocol, Wikipedia. Ver en: http://es.wikipedia.org/wiki/Simple_Network_Management_Protocol

[3] Netflow, Wikipedia. Ver en: https://es.wikipedia.org/wiki/Netflow
[4] MRTG, Wikipedia. Ver en: http://es.wikipedia.org/wiki/MRTG

[5 The Multi Router Traffic Grapher. Ver en: http://oss.oetiker.ch/mrtg/

[6] Cacti. Ver en: http://cacti.net/image.php?image_id=43

[7] TCPDUMP & Libpcap. Ver en: http://www.tcpdump.org

.3

Centros de Datos y Virtualización en IPv6

3.1_ Introducción

El contenido al que accedemos mediante el uso de Internet se encuentra almacenado en máquinas que se alojan en Centros de Datos distribuidos en el mundo. El acceso a este contenido es posible mediante aplicaciones que se comunican con procesos que corren en servidores con sistemas operativos y capacidades de hardware diferentes. Esta diversidad sumada a que el hardware de estas máquinas en general se encontraba sobre-dimensionado para correr pocas tareas, a la alta demanda de brindar servicios de Internet alojados en servidores propios, y al avance de la tecnología de integración de los microprocesadores actuales que presentan extensiones de virtualización, han producido un aumento considerable en el despliegue y uso de las máquinas virtuales.

Una característica que distingue a las máquinas virtuales es que hacen un mejor aprovechamiento del hardware al permitir tener múltiples máquinas corriendo simultáneamente, sirviendo requerimientos de manera independiente, aumentando la capacidad de acceso a recursos y servicios de red, lo que otorga una mayor flexibilidad al momento de migrar servicios y mejora la seguridad al aislar la máquina física y su administración respecto del sistema operativo que ejecuta cada una de las máquinas virtuales.

Las máquinas virtuales presentan diferentes modelos de interfaces de red que funcionan de manera similar a una interfaz física conocida, como es una placa de red ethernet. En cuanto al protocolo IP, en particular IPv4, la mayoría de los paquetes de software disponibles que se encuentran operando, funcionan de forma similar a la implementación disponible en cualquier sistema operativo.

Si tenemos en cuenta, por un lado, el agotamiento del pool de direcciones IPv4, y por otro el crecimiento del número de mecanismos de transición para hacer uso de IPv6, además de las diferentes propuestas que han surgido recientemente sobre la implementación de centros de datos que operan solo con IPv6 en el troncal de su red, resulta imprescindible preguntarnos que grado de soporte y de implementación del protocolo IPv6 presentan las máquinas virtuales.

En este capítulo vamos a describir las diferentes técnicas de virtualización, analizaremos los diferentes modelos de interfaces virtuales y por último consideraremos el nivel de soporte y de implementación del protocolo IPv6 que proveen las máquinas virtuales que podemos utilizar hoy. Además presentaremos los comandos y pasos necesarios para configurar IPv6 en máquinas virtuales.

3.2
Soporte de Virtualización en Procesadores

Aunque la mayoría de los procesadores actuales presentan las extensiones para virtualización, es recomendable antes de instalar cualquier paquete de software de virtualización en Linux, verificar si el procesador tiene capacidad de virtualización completa. Para esto podemos usar el siguiente comando:

```
# egrep '(vmx|svm)' --color=always /proc/cpuinfo
```

La respuesta puede contener la sigla vmx (Intel), svm (AMD) o no devolver nada en el caso de no ser compatible para virtualización.

3.3_
Tipos de virtualización

La tecnología de virtualización[1] permite disponer de múltiples máquinas corriendo en paralelo a partir de un único hardware. Estas múltiples máquinas son virtuales y su nivel de virtualización puede establecerse a partir del hardware mismo, mediante un supervisor que trabaja a modo de capa entre el hardware y el sistema operativo, o a nivel del sistema operativo sobre el que corren múltiples servidores virtuales independientes. En este aspecto podemos hacer una analogía entre virtualización y un sistema multitarea, en donde tenemos corriendo varios procesos simultáneamente sobre un único sistema operativo.

Las arquitecturas de virtualización cuentan en general con un componente que media entre el hardware y el sistema operativo huésped que se denomina supervisor o Monitor de Máquina Virtual (MMV) que se encarga de traducir el código binario, controlar la ejecución y administrar el acceso a los dispositivos y a diferentes recursos del hardware.

A modo de resumen, los 4 tipos de arquitecturas de virtualización que se describen a continuación y se muestran en las Figuras 1, 2, 3 y 4, son las disponibles hoy en diferentes implementaciones y bajo diferentes sistemas operativos (SO).

3.3.1. Emulación

- Emulador de hardware.
- Simula el hardware requerido mediante una Máquina Virtual (MV).
- Ejecuta cualquier SO nativo, sin modificación.

- El SO no advierte que usa un hardware ficticio.
- Ej: QEMU, Parallels, Microsoft Virtual Server.

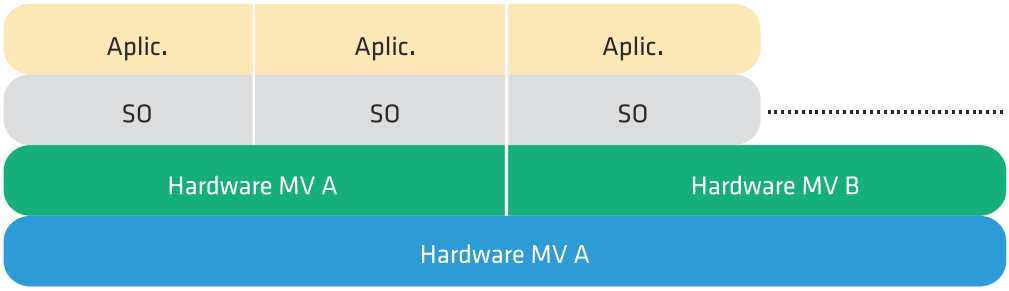

FIGURA 1: EMULACIÓN

3.3.2. Virtualización nativa o completa (Full)

- Máquina Virtual que media entre el SO huésped y el hardware nativo.
- El hardware y los recursos son compartidos y controlados por el supervisor o Monitor de Máquina Virtual (MMV)
- Es mas rápido que emulación, pero su desempeño se ve afectado debido a la intermediación del Monitor
- El SO no requiere ser modificado, pero debe soportar la arquitectura sobre la que corre.
- Ejemplo: VMware, z/VM (IBM), Linux KVM (Kernel Virtual Machine).

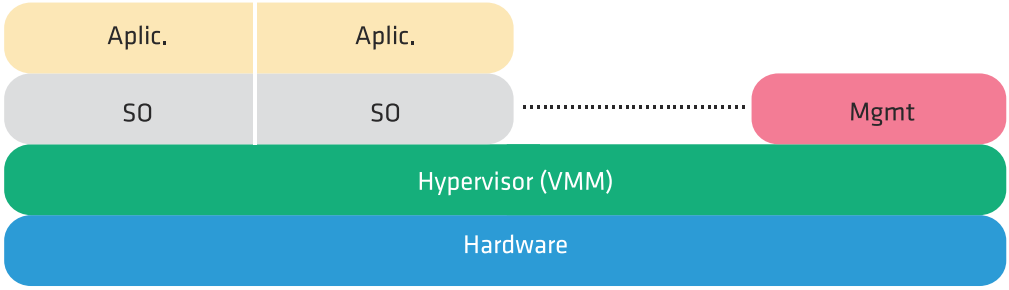

FIGURA 2: VIRTUALIZACIÓN NATIVA O COMPLETA

3.3.3. Para virtualización

- Similar a virtualización completa
- Comparte procesos con el supervisor
- Requiere re-compilar o portar el SO huésped para interactuar con el MMV.
- Soporta múltiples SO simultáneamente.
- Desempeño similar a un sistema no virtualizado.
- Ejemplo: Xen, UML (User Mode Linux)

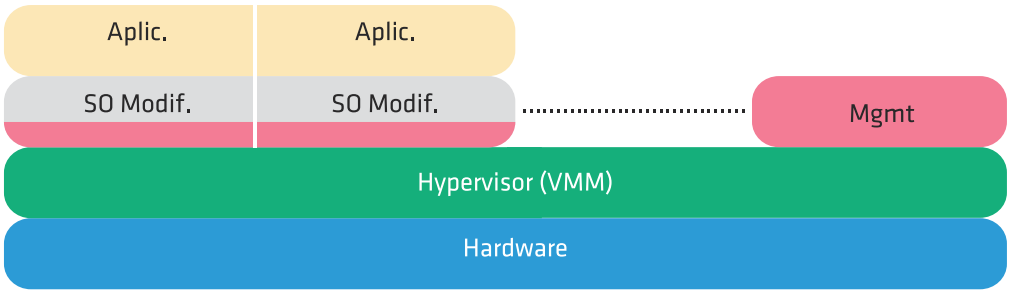

FIGURA 3: PARAVIRTUALIZACIÓN

3.3.4. Virtualización a nivel del sistema operativo

- Virtualiza servidores sobre el kernel de un SO.
- Divide un servidor físico (SF) en múltiples servidores virtuales (SV).
- Cada SV se ve y se comporta como un SF.
- Se pueden ejecutar múltiples copias de un OS (con distintas versiones) sobre un mismo SF.
- Ejemplo: OpenVZ, Virtuozzo, Linux-VServer, Solaris Zones, FreeBSD Jails.

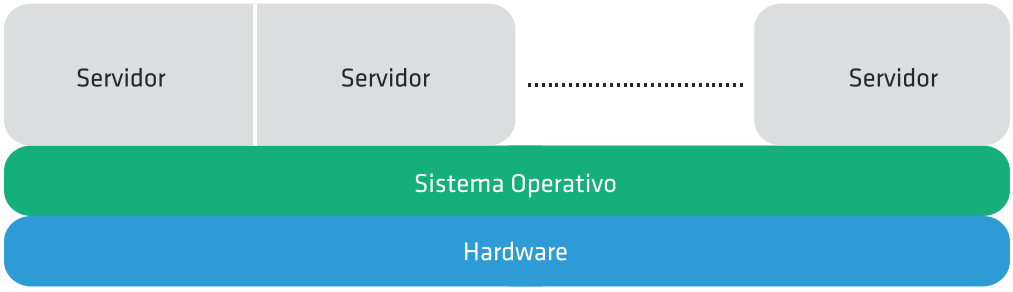

FIGURA 4: VIRTUALIZACIÓN A NIVEL DEL SO

En la Tabla 1. se muestra el soporte que presentan las distribuciones de Linux para con los diferentes paquetes de virtualización de código abierto.

Paquete	Distribución de Linux con soporte
Xen	RedHat 5.x, CentOS 5.x, Arch, Alpine, Debian, Fedora, Finnix, Gentoo, OracleLinux, OpenSuSE, Ubuntu
KVM	RedHat REHL 5.4 o superior, Ubuntu LTS 10.0.4 o superior, openSuSE SLES 11 SP1 o superior
OpenVZ	REHL 6 kernel 2.6.32., REHL 5 kernel 2.6.18. Plantillas oficiales para: CentOS 5 y 6, Fedora 17 y 18, Debian 6.0, Scientific 6, Suse 12.1, 12.2 y 12.3, Ubuntu 8.04, 10.04, 11.10, 12.04 y 12.10

TABLA 1: VIRTUALIZACIÓN SOPORTADA POR DISTRIBUCIÓN DE LINUX

El crecimiento en la adopción de KVM que se observa en la Tabla 1, se debe en parte a que recientemente las empresas HP, Intel, IBM y Red Hat, fundaron la Open Virtualization Alliance (OVA) para establecer un consorcio que actualmente cuenta con cientos de miembros. El objetivo de esta alianza es posicionar a KVM como una opción de virtualización de código abierto que sea rentable y de importancia estratégica para las empresas y proveedores de servicios de computación en la nube. Esta estrategia ha llevado a que, por ejemplo, RedHat se haya volcado a KVM en su distribución versión 6 y haya dejado de incluir a Xen, que fuera adquirido por la empresa Citrix, quién mantiene el proyecto de código abierto junto a versiones comerciales.

3.4_
Modos de virtualización de red

En esta sección describiremos los diferentes modos de virtualización a nivel de red que presentan las implementaciones de Xen, OpenVZ y KVM.

3.4.1. Xen

El principal componente de Xen[2] es el supervisor, quién se aloja entre el hardware y los sistemas operativos huéspedes. El supervisor es responsable de aislar y proteger el sistema, controlando el acceso y la asignación de recursos, además de diagramar la porción de máquina física asignada a cada huésped.

Xen permite albergar múltiples sistemas operativos que reciben el nombre de dominios (Dom). Estos dominios son planificados por el supervisor para hacer uso real de las CPUs físicas disponibles. A su vez cada SO maneja sus propias aplicaciones.

Al iniciar el sistema con el kernel de Xen, este crea el dominio Dom0, que tiene privilegios para el manejo del resto de los dominios y el acceso a los dispositivos virtuales. El resto de los dominios se denominan domU, siendo U un índice de valor 1 a N. Dentro Dom0 el proceso xend es quién se comunica con el supervisor para manejar las máquinas virtuales y permitir el acceso a sus consolas.

Xen puede compartir la interfaz de red física entre múltiples dominios, permitiendo que cada domU pueda tener una o más interfaces de red virtuales[3]. Xen combina los paquetes salientes de cada interfaz de red virtual del respectivo domU sobre la interfaz de red física. Del mismo modo, Xen separa cada paquete que entra por la placa de red física destinado a la interfaz virtual de cada domU activo.

Xen presenta tres modos de configurar las interfaces virtuales permitiendo crear diferentes arquitecturas de red de acuerdo a la cantidad de placas de red disponibles en el servidor, a las direcciones IP y subredes a asignar de forma manual o automática, a los servicios corriendo sobre

la interfaz virtual y que no tienen acceso directo por la interfaz física, etc. De los tres modos que se describen a continuación y son válidos para usar IPv4, solo se tendrán en cuenta los dos primeros en los ejemplos de configuración del protocolo IPv6.

3.4.1.1. Modo bridge

En este modo el tráfico entre interfaces es a nivel de capa 2 teniendo en cuenta solo las direcciones Físicas (MAC) e independizándose de las capas superiores. Este modo es el recomendado para usar en Xen por ser más simple en cuanto a su funcionamiento y configuración. Las direcciones MAC son visibles sobre la interfaz física y en el segmento de red ethernet a la que está conectada.

3.4.1.2. Modo router

En modo router, los paquetes son enviados entre las diferentes IP asignadas a las interfaces físicas y virtuales. Estas direcciones IP asignadas son visibles desde la red ethernet local, no así sus direcciones MAC. Estas direcciones IP son resueltas por ARP a la dirección MAC de la interfaz física, sumando la funcionalidad de proxy-arp para interfaces virtuales.

3.4.1.3. Modo NAT

En modo NAT, Xen funciona de manera similar al modo router, con la diferencia de que sus direcciones IP no son visibles desde el exterior. La diferencia está en la asignación de direcciones IP entre la interfaz del respectivo domU y la dirección IP asignada a la interfaz virtual del Dom0. Dada una clase C, El domU utiliza en su interfaz el rango de IPs desde la .2 a .127, entonces cada interfaz virtual del Dom0 utiliza el rango de .128 a 254 (+127) en concordancia con el domU. Por ejemplo, si configuramos para el domU la subred 10.0.0.2/24 con puerta de enlace 10.0.0.1, la interfaz virtual del Dom0 se auto-asigna la dirección IP 10.0.0.129.

3.4.2. OpenVZ

OpenVZ[3] es un sistema de virtualización a nivel del sistema operativo y presenta un kernel de Linux modificado. Además de la virtualización, OpenVZ presenta funcionalidades que lo destacan, como son la aislación, el manejo de recursos (mediante el uso de cuotas de disco, porciones de tiempo de ejecución en la CPU y contadores de recursos del kernel), y el establecimiento de puntos de control de chequeo, lo que facilita la migración en caliente de un contenedor a otra máquina, guardando previamente el estado completo de la MV.

Cada MV es creada mediante la instalación de plantillas que pueden bajarse de su sitio y están disponibles para diferentes versiones y distribuciones de Linux.

En cuanto a los modos de virtualización de red, OpenVZ presenta dos tipos de interfaces de red bien diferenciadas, que se distinguen por el nivel de seguridad y de acceso a su configuración (por parte del usuario y del administrador). Estas interfaces son:

- Virtual Ethernet device (veth)

- Virtual Network device (venet)

Virtual Ethernet (veth) es un dispositivo que provee funcionalidad en capa 2 y puede utilizarse dentro del contenedor mediante la asignación de una dirección física MAC. Se comporta como un dispositivo ethernet real Virtual Network (venet) es el dispositivo de red que presenta OpenVZ por omisión cuando instalamos un nuevo contenedor. Es un dispositivo que funciona en capa 3 y se comporta como una conexión punto a punto entre el contenedor y el servidor físico. Este dispositivo es el más seguro en cuanto al nivel de aislación, pero presenta algunas limitaciones en cuanto a su funcionalidad y administración.

En resumen, las diferencias entre las interfaces virtuales que provee OpenVZ se muestran en la siguiente tabla:

Características	Veth	Venet
Dirección MAC	Si	No
Broadcast dentro del CT	Si	No
Captura de tráfico	Si	No
Seguridad de red	Baja: Independiente del anfitrión y controlada en cada CT	Alta: controlada por el anfitrión
Utilizada en Bridge	Si	No
Soporta IPv6	Si	Si (no completa)
Desempeño	Rápido	Más rápido y más eficiente

TABLA 2: INTERFACES VIRTUALES EN OPENVZ

En la sección de configuración de IPv6 en OpenVZ utilizaremos la interfaz virtual veth que soporta IPv6 de forma completa.

3.4.3. KVM

KVM[4] es una tecnología que agrega capacidad de virtualización al kernel de Linux. En Linux un proceso tiene 2 modos de ejecución: modo kernel y modo usuario. Lo novedoso de KVM es que agrega un modo más de ejecución que se denomina huésped. La arquitectura que presenta KVM se muestra en la Figura 5.

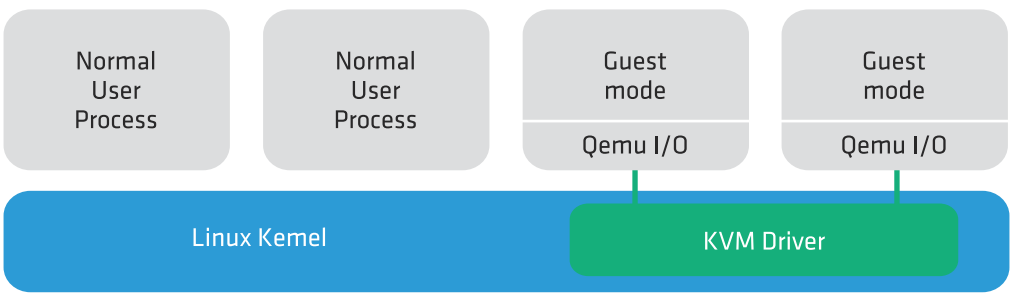

FIGURA 5: ARQUITECTURA Y MODOS DE EJECUCIÓN EN KVM

Las distribuciones de Linux RedHat y CentOS 6 proveen de manera nativa y de base el soporte y las herramientas para usar KVM como medio de virtualización. Esto puede verificarse ejecutando:

```
# yum grouplist | grep -i virt
   Virtualization
   Virtualization Client
   Virtualization Tools
   Virtualization Platform
```

Los paquetes contenidos en estos grupos pueden listarse ejecutando, por ejemplo, para el grupo Virtualization:

```
# yum -q groupinfo "Virtualization"
Group: Virtualization
 Description: Provides an environment for hosting virtualized guests.
 Mandatory Packages:
   qemu-kvm
 Default Packages:
   hypervkvpd
 Optional Packages:
   qemu-guest-agent
   qemu-kvm-toolsyum
```

KVM utiliza dos modos para configurar la red: modo bridge y modo usuario. El modo usuario es el modo por omisión y se basa en un modelo de virtualización de red que utiliza NAT. En la sección 5.3 de configuración de IPv6 en KVM veremos el caso para modo bridge.

3.4.4. Proxmox

Proxmox es un gestor para virtualización de servidores virtuales basado en código abierto. Se basa en OpenVZ y KVM, por lo que permite soportar Windows y Linux. Proxmox es miembro de la Open Virtualization Alliance.

Para su administración presenta un interfaz web muy amigable desde donde se crean, controlan y administran las MV. Estas MV se crean teniendo en cuenta que tecnología de virtualización se selecciona: si funcionan bajo el supervisor en el caso de KVM, se denominan máquinas virtuales (VM) o pueden crearse como huéspedes dentro de OpenVZ y se denominan contenedores (CT). Los módulos del kernel de OpenVZ y KVM se encuentran activos y se verifican ejecutando:

```
# lsmod | grep ^vz
vzethdev               8189  0
vznetdev              19230  5
vzrst                188071  0
vzcpt                142549  1 vzrst
vzdquota              56321  4
vzmon                 25335  7 vzcpt,vzrst,vznetdev
vzdev                  2765  6 vzmon,vzdquota,vznetdev,vzethdev
vzevent                2178  1
lsmod | grep ^kvm
kvm_intel             51799  0
kvm                  321061  1 kvm_intel
```

Las versiones más recientes de Proxmox se basan en Debian GNU/Linux 7.0 sobre un kernel 2.6.32 modificado.

3.4.5. VMware

VMware[5] es el producto comercial de virtualización más difundido y adoptado por los proveedores de contenido y de servicios de computación en la nube. Esta adopción se debe en parte a su interfaz de administración gráfica que resulta muy amigable para quién comienza a experimentar con la tecnología de virtualización y a la robustez que presenta el supervisor a las MV que se ejecutan.

VMware ofrece versiones de algunos de sus productos sin cargo, que pueden bajarse de su sitio y utilizarse luego de ser registrado. Estos productos presentan limitaciones en cuanto a funcionalidad y al número de máquinas virtuales soportadas, pero resultan interesantes para evaluar el producto y para experimentar en ambientes de uso personal o de bajo requerimiento.

3.5_
Implementación de IPv6 en máquinas virtuales

A pesar que el protocolo IPv6 está ampliamente difundido y ya ha sido desplegado en diferentes ámbitos y tipos de redes, los libros de referencia de máquinas virtuales y los manuales de los diferentes paquetes disponibles de código abierto o comerciales, no contienen suficiente información sobre la configuración de IPv6 en máquinas virtuales, comparado con la cobertura que estos presentan para IPv4.

Se podría suponer que el soporte para IPv6 que brindan estos productos aún no ha alcanzado la suficiente madurez o que el requerimiento, de parte de los clientes para con los proveedores de contenido y servicios, sigue siendo mayoritariamente sobre IPv4. El objetivo de las próximas secciones es revertir esta impresión e incentivar el uso y la implementación de IPv6 en ambientes virtualizados sobre Linux.

Antes de comenzar con la configuración de IPv6 en la interfaz de red virtual de una MV, describiremos en esta sección algunas de las limitaciones que pueden presentarse en la conectividad y en el kernel en escenarios simples de conexión, donde tenemos máquinas virtuales, un switch y un router IPv6.

3.5.1. Port Security en Cisco

Los Centros de Datos han incorporado diferentes medidas de seguridad que se aplican a partir del mismo puerto físico al que se conecta un cliente. Una posibilidad es aplicar la facilidad port security que proveen los switches Cisco en sus interfaces, lo que restringe el número de direcciones MAC permitidas sobre el puerto. En este escenario que se muestra en la Figura 6 no es posible el uso de máquinas virtuales en modo bridge, debido a que no son visibles las direcciones MAC de las interfaces virtuales sobre el segmento de red ethernet que conecta al Dom0 con el switch físico. La facilidad port security se configura por puerto y permite, si se especifica, establecer un número máximo de direcciones MAC seguras o especificar cada dirección MAC a conectar a dicho puerto, aunque esta última opción no es la más deseable.

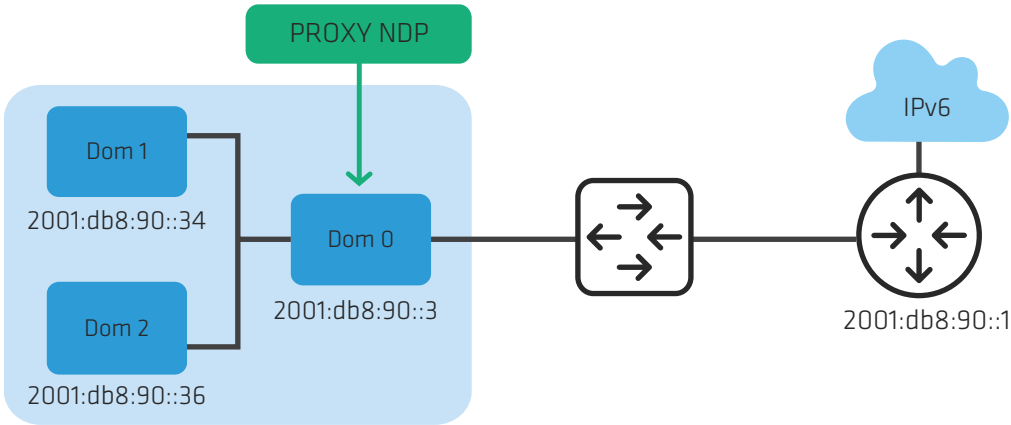

FIGURA 6: ESCENARIO PORT SECURITY

Para implementar IPv6 en ambientes virtualizados con soporte para auto-configuración es recomendable que las MV puedan operar en modo bridge, para lo cual es necesario desactivar la limitación que impone port security. Si esto no es posible, es necesario utilizar el modo router, para lo cual el kernel debe soportar la funcionalidad NDP que se describe a continuación y que facilita la configuración y el uso en IPv6.

3.5.2. Neighbor Discovery Proxy

En la Figura 7 se muestra la arquitectura de conectividad al utilizar Xen en modo router. Al crear las máquinas virtuales bajo este modo, se requiere que el Dom0 actúe para el protocolo IPv6 como un Neighboor Discovery Proxy, cumpliendo con las recomendaciones del RFC4389.

La funcionalidad de Neighbor Discovery Proxies (ND Proxy) descripta en el RFC4389[6] recomienda en primer lugar y de ser posible utilizar la tecnología de bridge a nivel de enlace. Sin embargo, esta tecnología basada en la solución IEEE 802.1D no siempre es aplicable. El RFC 4389 describe dos posibles escenarios de uso, estos son wireless upstream y PPP upstream, pero también es posible aplicarlo en otros escenarios, como es nuestro caso.

Cuando un equipo requiere conectarse a una máquina virtual, este genera paquetes ND tipo multicast (ff02::1) que llegan desde el exterior en primera instancia al dom0. Este debe actuar como un intermediario para el protocolo ND y reenviar estos paquetes dentro de la red virtual que compone el Dom0 y las máquinas virtuales. Las respuestas generadas por la máquina virtual deben a su vez ser transmitidas por el Dom0 sobre la red exterior para que las mismas alcancen al respectivo solicitante. De los 5 tipos de paquete ICMPv6 que se definen en el RFC 2461[7], existen 2 tipos que nos interesa analizar y se relacionan con la identificación de vecinos:

Solicitud de Vecino (Neighbor Solicitation) – generado por los nodos para determinar la dirección en la capa de enlace de sus vecinos, para verificar que el nodo vecino sigue activo (es alcanzable), y para detectar las direcciones duplicadas. Tipo en paquete ICMPv6 = 135.

Anunciación de Vecino (Neighbor Advertisement) – generado por los nodos como respuesta a la "solicitud de vecino", o para indicar cambios de direcciones en la capa de enlace. Tipo en paquete ICMPv6 = 136.

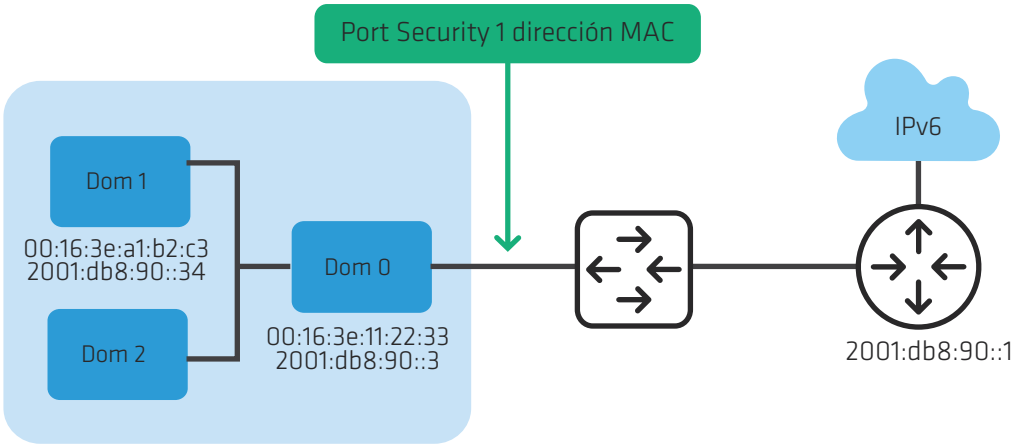

FIGURA 7: ESCENARIO PROXY NDP

A diferencia de la funcionalidad proxy_arp bajo IPv4, que se encuentra disponible en los kernels desde hace tiempo, la funcionalidad proxy_ndp esta disponible a partir de la versión del kernel 2.6.19[8].

Estas funcionalidades que se agregan en el kernel 2.6.19 introducen una Neighbor Cache para cada interfaz donde se almacenan la dirección

IPv6 y su MAC asociada de vecinos dentro del segmento (router, Dom0, domU, etc), y permite además mantener el estado de asignaciones de direcciones IPv6 al utilizar auto-configuración cuando llegan los anuncios de prefijos IPv6 por parte del router. Sin la funcionalidad de proxy_ndp, es necesario agregar en el router la dirección IPv6 y su MAC en la interface, agregar la dirección IPv6 de cada MV y la ruta destino de cada MV en el supervisor. En resumen, este es un ejemplo de los pasos necesarios para la configuración manual antes descripta:

- En el router (2001:db8:90::1)
  ipv6 neighbor 2001:db8:90::34 GigabitEthernet5/1.88 0014.4f8d.e352
- En el Dom0 (2001:db8:90::3)

```
# ip -6 addr add 2001:db8:90::34/64 dev eth0
# ip -6 route add default gw 2001:db8:90::1 dev eth0
```

Para obtener la interfaz virtual de la MV dom1 ejecutamos

```
# xm network-list dom1
```

Luego agregamos la ruta a la MV dom1, su interfaz virtual y la dirección fuente

```
# ip -6 route add 2001:db8:90::34 dev vif11.0 src 2001:db8:90::3
```

- En el dom1 (2001:db8:90::34)

```
# ip -6 neigh add 2001:db8:90::1 lladdr fe:ff:ff:ff:ff:ff dev eth0
# ip -6 neigh add 2001:db8:90::3 lladdr fe:ff:ff:ff:ff:ff dev eth0
# ip -6 neigh show
```

3.5.3. Parámetros del kernel de Linux para IPv6

Cuando utilizamos una tecnología de virtualización en Linux, el kernel resulta la pieza fundamental en el funcionamiento del supervisor o nodo principal y las MV. En la sección 4.2 vimos algunas limitaciones que pueden presentarse por faltantes de soporte de IPv6 en la versión del kernel.

En[9] se enumeran las variables y parámetros del kernel que se configuran para IPv6. Estas variables definen el comportamiento de IPv6 en los diferentes niveles de la capa de red y permiten modificar el estado y el control de diferentes parámetros que afectan su comportamiento, como son la activación del protocolo, la fragmentación y ensamblado de los paquetes IPv6, el reenvío (forwarding) de paquetes entre interfaces, el valor del MTU, la auto-configuración aceptando el prefijo anunciado, la detección de duplicados, entre otros.

Como veremos en los puntos de la Sección 3.3.5. algunas de estas variables influyen en el funcionamiento de las MV en IPv6 y es necesario cambiar su valor para habilitar funcionalidades requeridas y para desactivar otras que lo afectan. Es importante tener en cuenta que algunos de estos cambios en estas variables se realizan automáticamente cuando activamos una MV porque están contemplados en los archivos de configuración cuando ejecutamos un script o activamos una interfaz o servicio, pero otros deberán ser realizados por el administrador del supervisor o nodo principal.

3.5.4. Radvd

Router Advertisement Daemon (radvd) es un demonio que anuncia direcciones y rutas IPv6 sobre la red local y permite asignar direcciones IPv6. Este demonio envía periódicamente mensajes de anuncio definidos en el RFC 2461 ya descriptos en el punto 3.4.2 y recibe los mensajes de solicitud de vecinos para finalmente asignar una dirección IPv6 a un nodo en un modo de configuración automática sin estado.

Radvd resulta interesante de usar en escenarios virtualizados en los cuales no es posible definir un bridge, la funcionalidad de proxy_arp no esta disponible o los mensajes del protocolo ND no llegan a los nodos. En estas situaciones es posible utilizar el nodo principal o el supervisor como un router para anunciar los prefijos y direcciones IPv6 a las MV.

La configuración del demonio radvd se define en Linux en el archivo /etc/radvd.conf. Los parámetros más importantes a definir son la interfaz donde escucha el demonio y sobre el que emite los mensajes de anuncio, el prefijo IPv6, el tiempo de vida del prefijo, y su frecuencia de envío.

Un ejemplo de configuración es el siguiente:

```
interface eth0 {
       AdvSendAdvert on;
        MinRtrAdvInterval 3;
        MaxRtrAdvInterval 10;
        prefix 2001:db8:90::/64 {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };
};
```

Es importante notar que radvd no anuncia parámetros de configuración como podría ser la dirección IPv6 de servidores de nombre de dominio (DNS), de un servidor WINS o de un servidor TFTP para equipos que requieren transferir archivos de configuración al iniciarse. Para estos casos es necesario implementar un servidor DHCPv6.

3.6_
Configuración de IPv6 en máquinas virtuales

La configuración de IPv6 en una interfaz de cualquier máquina virtual no debería ser diferente a la que aplicamos en una máquina real que corre un sistema operativo que está preparado para usar IPv6.

Por esta razón nos centraremos solo en los modelos de virtualización de red que nos brindan el soporte adecuado para configurar IPv6, y evitaremos las implementaciones que no son nativas o completas.

Para los ejemplos utilizaremos el prefijo de dirección IPv6 2001:db8::/32 reservado para documentación en el RFC 3849 y utilizaremos como referencia el sistema operativo Linux incluido en la distribución CentOS.

La configuración de red en la distribución de CentOS es la misma que presentan las distribuciones Fedora y RedHat, basada en un conjunto de archivos donde se definen variables y parámetros que controlan la asignación de IPv6. El archivo de órdenes (script) utilizado para iniciar el servicio de IPv6, configurar las interfaces y rutas, y reasignar parámetros del kernel es ifup-ipv6[10]. El detalle de archivos y variables se describen en la siguiente tabla:

Variables definidas en el archivo /etc/sysconfig/network		
Variable	Descripción	Kernel
IPV6INIT=yes|no	Habilita la configuración de IPv6 para la interfaz.	El módulo ipv6 debe estar cargado en el kernel.
IPV6FORWARDING=yes|no	Controla en reenvio de paquetes	net.ipv6.conf.DEVICE.forwarding=1|0
Variables definidas en el archivo /etc/sysconfig/network-scripts/ifcfg-ethX, con X=0,1,...N		
Variable	Descripción	Kernel
IPV6_DEFAULTGW=<dirección IPv6>	Controla la ruta por defecto o puerta de enlace IPv6 (valor opcional)	
IPV6ADDR= <dirección IPv6>[/<longitud del prefijo>]	Especifica la dirección primaria IPv6 de forma estática o manual	
IPV6ADDR_SECONDARIES= "<dirección IPv6>[/<longitud del prefijo>]..."	(valor opcional)	
IPV6_ROUTER=yes|no	Controla la auto-configuración IPv6 (no: interfaz proveedor múltiple sin encaminado)	net.ipv6. conf.DEVICE.forwarding=1|0
IPV6_AUTOCONF=yes|no	Controla la auto-configuración IPv6	net.ipv6.conf.DEVICE.accept_ra=1|0 redirects=1|0
IPV6_MTU=<MTU for IPv6>	Controla el MTU en IPv6 aplicado a esta interfaz. (valor opcional)	

TABLA 3: CONFIGURACIÓN DE INTERFACES DE RED: VARIABLES Y PARÁMETROS DEL KERNEL

La configuración de IPv6 presenta dos modos de operación por omisión: router y host. Estos modos se definen mediante la combinación de variables que activan diferentes parámetros como vimos en el punto 3.4.3.

Para el modo router, las variables deberían tener los valores:
`IPV6FORWARDING=yes, IPV6_AUTOCONF=no, IPV6_ROUTER=yes`

Para el modo host, se anula el reenvío de paquetes y se habilita la auto-configuración:
`IPV6FORWARDING=no, IPV6_AUTOCONF=yes`

3.6.1. Xen

Como vimos en la sección 3.3.1, Xen presenta 3 modos de virtualización de red: Bridge, Router y NAT. De estos 3 modos, solo describiremos la configuración de IPv6 en los dos primeros, descartando el modo NAT porque es una tecnología orientada al uso de IPv4.

Por cada interfaz virtual de red, Xen crea un par de dispositivos de red. El que se denomina ethN, reside en el dominio huésped y se denomina de forma similar a una interfaz física. Esto significa que para el dominio huésped domU, su configuración es similar a la que usamos sobre una interfaz física ethernet. El segundo dispositivo de red es el que reside en el dominio principal dom0 y se identifica con el nombre vifDOMID.DEVID, donde DOMID es el identificador del dominio huésped y DEVID es el identificador del dispositivo ethernet creado para el domU. Por ejemplo, para el dominio dom5, el identificador es el 5, y los dispositivos creados para Dom0 y el dom5 toman el nombre de vif5.0 y eth0. Si se asignara una segunda interfaz virtual al dom5, esta seria la eth1 y en el Dom0 estaría asociada a la interfaz virtual vif5.1

Entre ambos dispositivos de red se establece un canal de comunicación virtual por el que pasa el tráfico entre el dominio huésped y el Dom0, que finalmente alcanza la interfaz física usando un bridge o router, dependiendo del modo de virtualización seleccionado en la instalación de Xen.

Cada interfaz virtual de red requiere de la asignación de una dirección MAC Ethernet para su funcionamiento. Esta asignación puede definirse en el archivo de configuración del dominio huésped, junto a otros parámetros de red que también pueden asignarse de forma manual.

Xen utiliza 3 formas de hacerlo, con el siguiente orden de preferencia:

- Asignar una dirección MAC de un rango asociado a un Identificador Unico Organizacional (OUI) que sea válido y que quién lo asigna debe controlar dicho rango y ser responsable por esta configuración

- Mediante la generación de una secuencia aleatoria de 6 bytes, con el primer byte siguiendo el patrón de bits xxxxxx10, con cada bit x generado de manera aleatoria, y los restantes 5 bytes también generados al azar.

- Asignando una dirección al azar dentro del espacio 00:16:3e:xx:xx:xx. El proyecto Xen tiene asignado el OUI 00:16:3e y está disponible a los usuarios de Xen para las asignaciones locales.

Estas diferentes formas de asignación de la dirección MAC responden a que esta debe ser única entre todos los dispositivos que se encuentran en el mismo segmento de la red local, sean estos físicos o virtuales. Si no se tiene un OUI propio, es preferible generar la dirección MAC usando la segunda opción ya que aleatoriamente tiene 46 bits frente a los 12 bits de la tercera opción, evitando la duplicidad y mejorando la seguridad cuando somos víctimas de un ataque de barrido de direcciones IPv6.

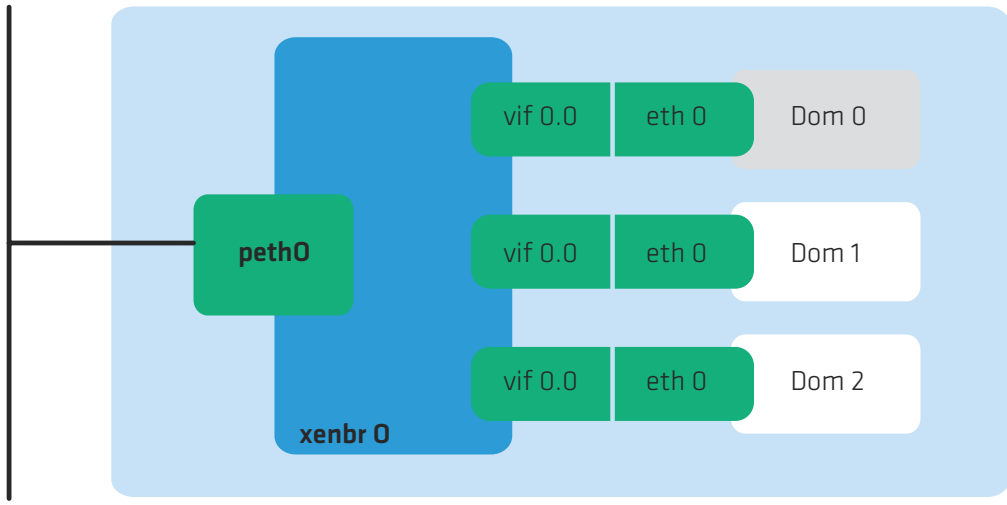

FIGURA 8: XEN EN MODO BRIDGE

El modo bridge en Xen es la configuración de red por defecto y la más común, creando el bridge por software en el Dom0 y permitiendo que todas las domU sean visibles en la red.

En el modo bridge se reasignan los nombres de las interfaces, siendo el dispositivo eth0 renombrado a peth0, el dispositivo físico y se crea un bridge con el nombre xenbr0 como se muestra en la Figura 8.

Cada MV que se crea tiene asociado un archivo de configuración en donde se definen los parámetros como son el número de CPUs, el tamaño de la memoria, los sistemas de archivos asociados al disco, y la interfaz virtual que se denomina vif.

Cuando se ejecuta el demonio xend al iniciar el Dom0 se activan las MV cuyos archivos de configuración se encuentran en el directorio /etc/xen/auto.

Si tenemos 2 o más interfaces físicas es posible definir un bridge para cada interfaz física y luego asociar cada MV a uno de estos bridges dentro del archivo de configuración. En el escenario que se muestra en la Figura 9 tenemos una máquina física que tiene una tarjeta ethernet de 4 puertos en la que se habilitan 2 puertos ethernet y se definen 2 bridges debido a que cada puerto esta conectado a una VLAN diferente.

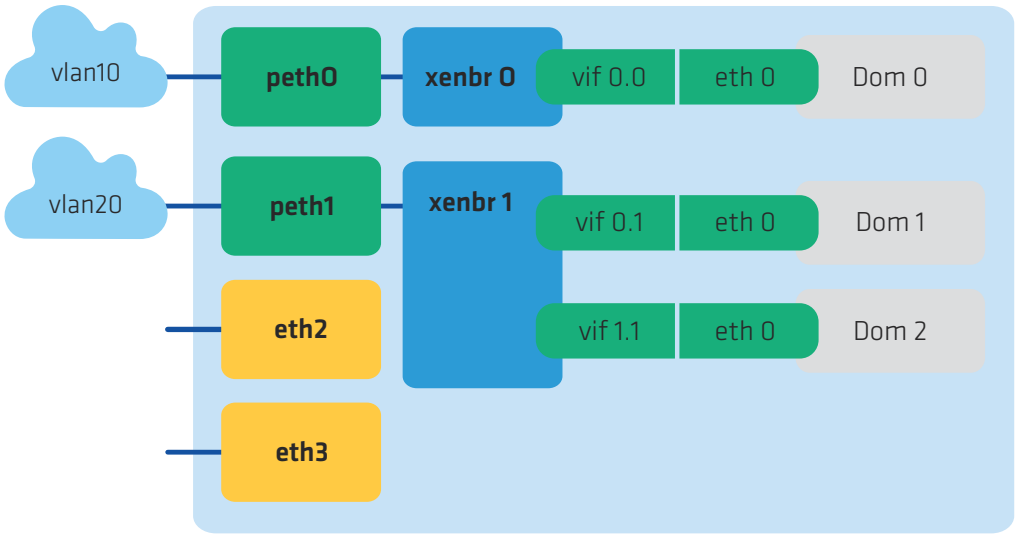

FIGURA 9: EJEMPLO DE ARQUITECTURA BRIDGE EN XEN

Para ver las MV que se están activas y corriendo, usamos el comando xm:

```
# xm list
Name                ID Mem(MiB) VCPUs State   Time(s)
Domain-0             0     5450     8 r----- 103347.5
iris                 1     1024     1 -b---- 1177253.0
ns1                  2     1024     1 -b----  27655.0
vpn                 10      512     1 -b----   7742.6
```

Las 3 MV asociadas a diferentes bridges tienen la siguiente configuración:

```
# cat /etc/xen/auto/* | grep ^vif
iris: vif = [ “mac=00:16:3e:5c:90:25,bridge=xenbr0,script=vif-bridge” ]
ns1:  vif = [ “mac=00:16:3e:20:42:7a,bridge=xenbr1,script=vif-bridge” ]
vpn:  vif = [ “mac=00:16:3e:78:6e:6a,bridge=xenbr0,script=vif-bridge” ]
```

En este caso tenemos 3 MV que utilizan 2 bridges diferentes con los nombres xenbr0 y xenbr1. También podemos observar que las direcciones MAC tienen el OUI 00:16:36 para todos los casos.

```
xenbr0    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:4480186 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:827157828 (788.8 MiB)  TX bytes:0 (0.0 b)
```

```
xenbr1    Link encap:Ethernet  HWaddr FE:FF:FF:FF:FF:FF
          UP BROADCAST RUNNING NOARP  MTU:1500  Metric:1
          RX packets:4480134 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:827893361 (789.5 MiB)  TX bytes:0 (0.0 b)
```

Si observamos la distribución de interfaces virtuales en cada bridge y la interfaz física asociada, tenemos:

```
# brctl show
bridge name    bridge id            STP enabled    interfaces
virbr0         8000.000000000000    yes
xenbr0         8000.fefffffffff     no             vif10.0
                                                   vif1.0
                                                   vif0.0
                                                   peth0
xenbr1         8000.fefffffffff     no             vif2.0
                                                   vif0.1
                                                   peth1
```

Es necesario ahora definir en la MV la configuración IPv6 en su interfaz. A modo de prueba y en forma manual podemos ejecutar los siguientes comandos dentro de vm01:

```
# ip -6 addr add 2001:db8:90::30/64 dev eth0
# ip -6 route add default via 2001:db8:90::1
```

Podemos verificar la configuración IPv6 ejecutando los comandos:

```
# ip -6 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436
    inet6 ::1/128 scope host
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qlen 1000
    inet6 2001:db8:90::30/64 scope global dynamic
    inet6 fe80::5054:ff:fe7d:78ed/64 scope link

# ip -6 route show
2001:db8:90::/64 dev eth0  proto kernel  metric 256  expires 0sec mtu
1500 advmss 1440 hoplimit 4294967295
fe80::/64 dev eth0  proto kernel  metric 256  mtu 1500 advmss 1440
hoplimit 4294967295
default via  2001:db8:90::1 dev eth0  proto kernel  metric 1024
expires 0sec mtu 1500 advmss 1440 hoplimit 64
```

En este momento tendríamos conectividad IPv6. Para verificarlo podemos ejecutar un ping6 destinado al router previamente configurado:

```
# ping6 2001:db8:90::1
PING 2001:db8:90::1(2001:db8:90::1) 56 data bytes
64 bytes from 2001:db8:90::1: icmp_seq=1 ttl=64 time=1.17 ms
64 bytes from 2001:db8:90::1: icmp_seq=2 ttl=64 time=0.457 ms
64 bytes from 2001:db8:90::1: icmp_seq=3 ttl=64 time=0.504 ms
.........
```

3.6.1.1. Modo router en Xen

En modo router Xen requiere actuar como intermediario del protocolo NDP para permitir la auto-configuración IPv6 de las MV. En el Dom0 definimos los parámetros del kernel que habilitan el uso del protocolo NDP agregando las siguientes lineas en el archivo /etc/sysctl.conf

```
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.default.proxy_ndp=1
net.ipv6.conf.all.proxy_ndp = 1
```

El próximo paso es indicar en el archivo de configuración de la MV el script a utilizar en modo router al momento de iniciar la MV:

```
vif=[“mac=00:16:3E:20:42:7A,script=vif-route,ip=192.168.1.30”]
```

Para tener conectividad IPv6 es necesario configurar la MV y podemos hacerlo de forma manual como vimos para Xen en modo bridge en el ejemplo de vm01, agregando estos comandos en el archivo /etc/rc.local para que se ejecuten al iniciar al MV o podemos configurarlo de manera permanente en los archivos que se describieron en la tabla 3.

Si estamos dentro de un centro de datos tipo solo IPV6, necesitamos tener conectividad IPv6 en las MV desde el momento de la creación para tener acceso a las mismas por fuera del supervisor y de la consola que este provee. Xen no ofrece de base esta capacidad en sus scripts, por lo que es necesario proveer una solución que puede provenir de aplicar un parche al script vif-route para la versión de Xen en uso, o bajar un script disponible en Internet que nos provea la solución sin necesidad de modificar nosotros el script vif-route. Algunas de estas soluciones pueden consultarse en sitios similares a[11].

Con el script vif-route ya con soporte para IPv6, tenemos que ajustar la configuración de red de Xen para que funcione en modo router. Esto lo hacemos editando el archivo /etc/xen/scripts/xend-config.sxp y quitando el comentario a las lineas:

```
(network-script network-route)
(vif-script     vif-route)
```

Ahora podemos agregar en la definición de la interfaz virtual la dirección IPv6 a asignar a la MV:

```
vif=[“mac=00:16:3E:20:42:7A,ip=192.168.1.30 2001:db8:90::30/64”]
```

Al iniciar la MV tendremos conectividad IPv6 dentro del segmento de red y sobre el prefijo IPv6 donde se encuentra la MV.

3.6.2. OpenVZ

Para que los contenedores puedan usar IPv6 es necesario previamente configurar IPv6 en la interfaz del nodo principal y activar algunos parámetros del kernel.

Este es un ejemplo para configurar IPv6 para la distribución de Linux CentOS.

Agregar los siguientes parámetros de configuración de red en el archivo /etc/sysconfig/network

```
IPV6INIT=yes
```

Agregar las siguientes lineas en el script de configuración de red de la respectiva interfaz. Para el caso de la interfaz eth0, el archivo a modificar es /etc/sysconfig/network-scripts/ifcfg-eth0

```
NETWORKING_IPV6=yes
IPV6FORWARDING=yes
IPV6_DEFAULTDEV=eth0
IPV6_AUTOCONF=no
IPV6_DEFAULTGW=2001:db8:90:192::1
IPV6ADDR=2001:db8:ab34:90:192::2/64
```

En este ejemplo utilizamos el prefijo 2001:db8:90:192::/64, con la dirección ::1 para identificar a la puerta de enlace y la ::2 para principal.

En el archivo /etc/sysctl.conf es necesario agregar los siguiente parámetros del kernel para permitir el reenvío de paquetes IPv6 y activar la funcionalidad de intermediario del protocolo ND para la auto-configuración de la interfaz virtual del contenedor

```
net.ipv6.conf.default.forwarding = 1
net.ipv6.conf.all.forwarding = 1
net.ipv6.conf.default.proxy_ndp=1
net.ipv6.conf.all.proxy_ndp = 1
```

Finalmente, para que estos valores queden activos en el kernel y se active cuando reiniciamos el contenedor, ejecutamos:

```
# sysctl -p
```

Para una última comprobación de que estos parámetros del kernel están activos con el valor 1, ejecutamos:

```
# sysctl -a | grep net.ipv6.conf
```

Después de configurar el nodo principal, procedemos a configurar la máquina huésped. Es importante primero verificar que el módulo IPv6 del kernel este activo. Podemos verificarlo ejecutando:

```
# lsmod | grep ipv6
```

Y luego agregamos en el archivo /etc/sysconfig/network la variable

```
IPV6INIT=yes
```

Para asegurarnos que la pila IPv6 se activa al reiniciar la MV.

A modo de prueba podemos hacer una configuración manual de IPv6 para la MV. Para ello ejecutamos:

```
# ip -6 addr add 2001:db8:90::28/64 dev eth0
# ip -6 route add ::/0 via 2001:db8:90::1
```

Comprobamos que tenemos conexión IPv6 con el nodo principal y con el router

```
# ping6 -q -c 5 2801:db8:90::2
PING 2001:db8:90::2(2001:db8:90::2) 56 data bytes

--- 2001:db8:90::2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.016/0.024/0.046/0.012 ms

# ping6 -q -c 5 2001:db8:90::1
PING 2001:db8:90::1(2001:db8:90::1) 56 data bytes

       --- 2001:db8:90::1 ping statistics ---
       5 packets transmitted, 5 received, 0% packet loss, time 3999ms
       rtt min/avg/max/mdev = 0.332/0.342/0.352/0.014 ms
```

Si queremos que esta configuración sea permanente definimos los parámetros configurados manualmente en las variables del archivo / etc/sysconfig/network-script/ifcfg-eth0

```
NETWORKING_IPV6=yes
IPV6FORWARDING=no
IPV6_DEFAULTDEV=eth0
IPV6_AUTOCONF=no
IPV6_DEFAULTGW=2001:db8:90::1
IPV6ADDR=2001:db8:90::28/64
```

Si utilizamos direcciones IPv6 públicas, podemos verificar que otros sitios IPv6 son alcanzables. Por ejemplo, google ofrece los servidores de dominio DNS públicos para IPv6 2001:4860:4860::8888 y 2001:4860:4860::8844. Podemos verificar que son alcanzables ejecutando:

```
# ping6 -c 5 2001:4860:4860::8888
PING 2001:4860:4860::8888(2001:4860:4860::8888) 56 data bytes
64 bytes from 2001:4860:4860::8888: icmp_seq=1 ttl=54 time=175 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=2 ttl=54 time=175 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=3 ttl=54 time=175 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=4 ttl=54 time=176 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=5 ttl=54 time=175 ms

--- 2001:4860:4860::8888 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 175.747/176.003/176.745/0.531 ms

# ping6 -c 5 2001:4860:4860::8844
PING 2001:4860:4860::8844(2001:4860:4860::8844) 56 data bytes
64 bytes from 2001:4860:4860::8844: icmp_seq=1 ttl=54 time=175 ms
64 bytes from 2001:4860:4860::8844: icmp_seq=2 ttl=54 time=175 ms
64 bytes from 2001:4860:4860::8844: icmp_seq=3 ttl=54 time=175 ms
64 bytes from 2001:4860:4860::8844: icmp_seq=4 ttl=54 time=176 ms
64 bytes from 2001:4860:4860::8844: icmp_seq=5 ttl=54 time=178 ms

--- 2001:4860:4860::8844 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 175.670/176.351/178.209/1.047 ms
```

Luego podemos agregar estos servidores DNS para nuestra máquina virtual en el archivo /etc/resolv.conf de la siguiente forma:

```
nameserver 2001:4860:4860::8888
nameserver 2001:4860:4860::8844
```

3.6.3. KVM

En los ejemplos de configuración de IPv6 en KVM que siguen a continuación nos basaremos en el modo bridge de virtualización de red y utilizando la distribución CentOS de Linux

El primer paso es verificar que están instalados los paquetes que requiere KVM para su funcionamiento y administración. Estos paquetes son, en la mayoría de las distribuciones de Linux:

```
bridge-utils, qemu-kvm-tools, qemu-kvm, libvirt, virt-manager
```

Para verificar que están instalados podemos correr el comando rpm con un filtro específico para cada paquete. Por ejemplo, para verificar que están instalados los paquetes qemu-kvm ejecutamos

```
# rpm -qa | grep qemu-kvm
qemu-kvm-tools-0.12.1.2-2.355.0.1.el6.centos.3.x86_64
qemu-kvm-0.12.1.2-2.355.0.1.el6.centos.3.x86_64
```

Luego verificamos que los módulos de KVM están corriendo.

```
# lsmod | grep kvm
kvm_intel              53484  4
kvm                   316602  1 kvm_intel
```

Ahora iniciamos el demonio libvirtd que nos permite gestionar el sistema de virtualización. Las dos formas de hacerlo son:

```
# /etc/init.d/libvirtd start
# service libvirtd start
```

Para que libvirtd se ejecute cuando reiniciamos la máquina física, configuramos:

```
# chkconfig  level 35 libvirtd on
```

KVM presenta un nueva interfaz denominada virbr0, que es un bridge virtual propio que se crea por defecto para brindar aislación y comunicación entre el nodo principal y las futuras máquinas virtuales. Esta interfaz tiene asignada la dirección IP 192.168.122.1/24 y dentro de esta subred pueden asignarse direcciones IPv4 al resto de las MV a crear.

```
# ifconfig virbr0
virbr0    Link encap:Ethernet  HWaddr 52:54:00:B8:20:57
          inet addr:192.168.122.1  Bcast:192.168.122.255 Mask:255.255.255.0
             UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
             RX packets:0 errors:0 dropped:0 overruns:0 frame:0
             TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
             collisions:0 txqueuelen:0
             RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

Se observa que la dirección MAC tiene el OUI 52:54:00 que es el correspondiente al proyecto QEMU. Si observamos el estado del cortafuegos para IPv4 y la cadena NAT vemos que el bridge virbr0 es local y que utiliza NAT sobre la IP de la interfaz eth0.

```
# iptables -t nat -L -n
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  tcp  --  192.168.122.0/24    !192.168.122.0/24    masq
ports: 1024-65535
MASQUERADE  udp  --  192.168.122.0/24    !192.168.122.0/24    masq
ports: 1024-65535
MASQUERADE  all  --  192.168.122.0/24    !192.168.122.0/24

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Para usar IPv6 en KVM es necesario configurar un bridge que no tenga las limitaciones de virbr0 antes de comenzar a instalar las nuevas MV. Para esto, en el directorio /etc/sysconfig/network-scripts creamos un nuevo archivo ifcfg-br0 y modificamos el archivo ifcfg-eth0 para incorporarlo al bridge. Los archivos son los siguientes:

Se resaltan las principales diferencias entre los archivos de configuración entre el bridge br0 y la interfaz eth0. La variable DEVICE indica el nombre del dispositivo, TYPE define el tipo de dispositivo (Bridge|Ethernet) y en el caso de la interfaz eth0 es necesario asociarla al bridge mediante el parámetro BRIDGE=br0.

/etc/sysconfig/network-scripts/ifcfg-br0	/etc/sysconfig/network-scripts/ifcfg-eth0
DEVICE=br0	**DEVICE=eth0**
BOOTPROTO=static	BOOTPROTO=static
BROADCAST=192.168.1.255	HWADDR=4C:72:B9:B0:E3:D0
DNS1=192.168.1.2	NM_CONTROLLED=no
GATEWAY=192.168.1.1	ONBOOT=yes
IPADDR=192.168.1.32	**TYPE=Ethernet**
IPV6ADDR=2001:db8:90::32/64	**BRIDGE=br0**
IPV6INIT=yes	
IPV6_AUTOCONF=no	
NETMASK=255.255.255.0	
NM_CONTROLLED=no	
ONBOOT=yes	
TYPE=Bridge	

TABLA 4: ARCHIVOS DE CONFIGURACIÓN DE INTERFACES

Otras variables propias de los archivos “ifcfg-X” son BOOTPROTO, que indica si se asigna una dirección IP estática o dinámica (static|dhcp), la variable ONBOOT para configurar la red al iniciar la máquina física y la

variable NM_CONTROLLED que indica si el programa Network Manager controla dicho dispositivo.

Ahora necesitamos reiniciar la red para activar el bridge.

```
# /etc/init.d/network restart
```

Si verificamos las interfaces tenemos las siguientes:

```
# ifconfig
br0       Link encap:Ethernet  HWaddr 4C:72:B9:B0:E3:D0
          inet addr:192.168.1.32  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2001:db8:90::32/64 Scope:Global
          inet6 addr: fe80::4e72:b9ff:feb0:e3d0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:383792 errors:0 dropped:0 overruns:0 frame:0
          TX packets:250606 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:225550607 (215.1 MiB)  TX bytes:39931149 (38.0 MiB)

eth0      Link encap:Ethernet  HWaddr 4C:72:B9:B0:E3:D0
          inet6 addr: fe80::4e72:b9ff:feb0:e3d0/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:602636 errors:0 dropped:0 overruns:0 frame:0
          TX packets:378914 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:558122420 (532.2 MiB)  TX bytes:48845029 (46.5 MiB)
          Interrupt:20 Memory:f7c00000-f7c20000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:22942 errors:0 dropped:0 overruns:0 frame:0
          TX packets:22942 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16110994 (15.3 MiB)  TX bytes:16110994 (15.3 MiB)

virbr0    Link encap:Ethernet  HWaddr 52:54:00:B8:20:57
          inet addr:192.168.122.1  Bcast:192.168.122.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

En este ejemplo el bridge br0 está configurado con direcciones IPv4 e IPv6, y utiliza un modelo de doble pila que permite usar ambos protocolos.

En modo bridge, los parámetros del kernel con valor 0 a verificar son:

```
net.ipv6.conf.all.forwarding = 0
net.bridge.bridge-nf-call-arptables = 0
net.bridge.bridge-nf-call-iptables = 0
net.bridge.bridge-nf-call-ip6tables = 0
```

Aunque no es la finalidad de este capítulo entrar en el detalle de los pasos necesarios para crear una nueva máquina virtual, si nos interesa ver como configurar IPv6 al crear una nueva máquina virtual. En las siguientes imágenes se observa la diferencia en las opciones avanzadas cuando esta previamente configurado un bridge en el supervisor. En la Figura 10 se muestra el caso por defecto, en donde solo se puede configurar la red virtual en modo NAT, lo cual no es compatible para IPv6.

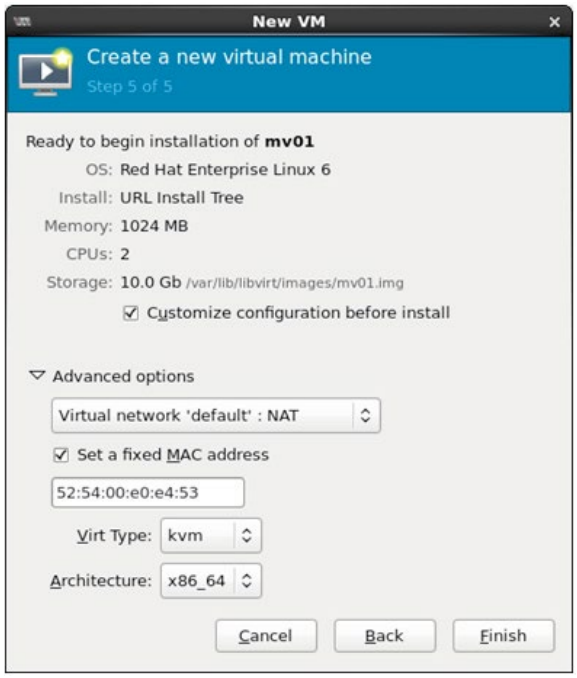

FIGURA 10: CONFIGURACIÓN DE RED VIRTUAL EN MODO NAT

Por otro lado en la Figura 11 se observa que esta disponible el dispositivo eth0 asociado al bridge br0 para la nueva MV. En ambos casos las direcciones MAC pueden fijarse y presentan el OUI 52:54:00 antes mencionado.

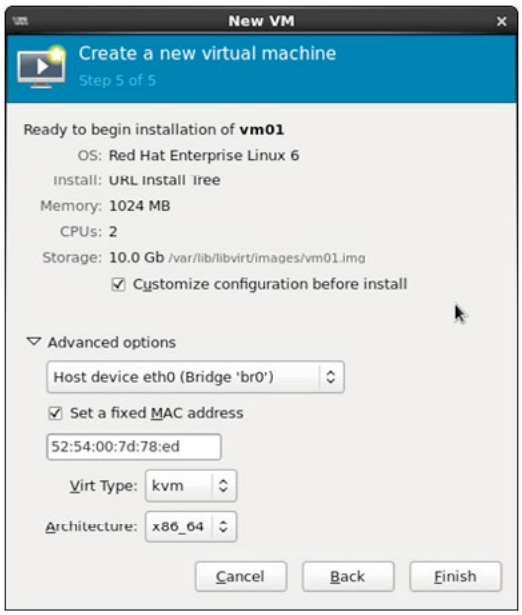

FIGURA 11: CONFIGURACIÓN DE RED VIRTUAL EN MODO BRIDGE

La instalación continúa y se muestra en una consola virtual generada mediante la aplicación VNC. La configuración de la red dentro de la MV no varía de cualquier instalación habitual en la distribución utilizada y es en ese paso en el que debe seleccionarse si la asignación es de forma manual o automática para las direcciones IPv4 e IPv6.

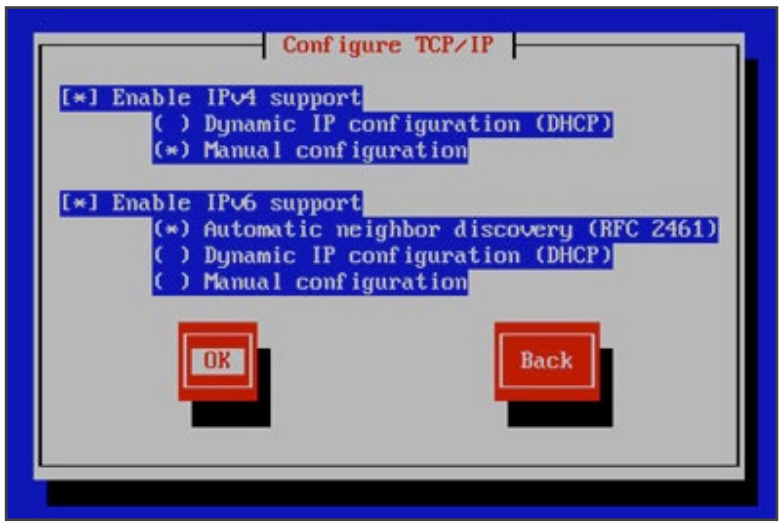

FIGURA 12: CONFIGURACIÓN DE TCP/IP

Si en el paso de configuración que muestra la Figura 12 seleccionamos la opción de configuración manual para ambos protocolos, el proceso de instalación nos ofrece completar los campos de la ventana que se muestra en la Figura 13. Para IPv6 debemos definir el campo de la dirección IPv6 y el del prefijo. Por ejemplo: 2001:db8:90::30/64

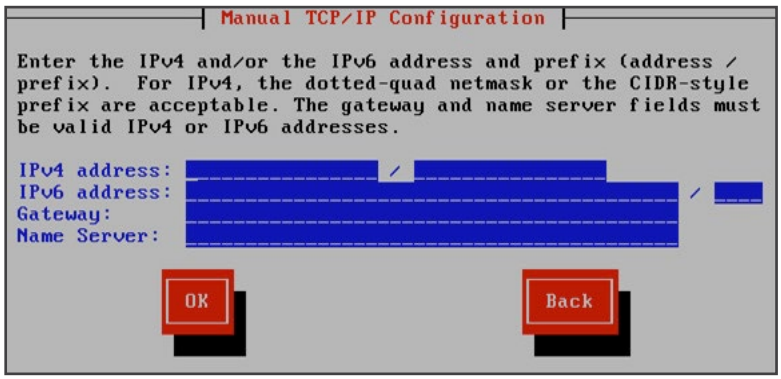

FIGURA 13: CONFIGURACIÓN MANUAL DE TCP/IP

Una vez finalizada la instalación de la MV, podemos conectarnos vía IPv6 y verificar la configuración de la red. En este ejemplo la dirección IPv6 es asignada de forma automática por NDP para un prefijo 2001:db8:90::/64.

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 52:54:00:7D:78:ED
          inet addr:192.168.1.30  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: 2801:0:90::5054:ff:fe7d:78ed/64 Scope:Global
          inet6 addr: fe80::5054:ff:fe7d:78ed/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:125527 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4683 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:20534972 (19.5 MiB)  TX bytes:366917 (358.3 KiB)
```

En el nodo principal tenemos también la interfaz vnet0 que es una interfaz tipo tap asociada al proceso KVM y hace la función de capa de enlace con la interfaz de la respectiva MV. Esta asociatividad con la interfaz de la MV se puede observar en su dirección MAC, la cual solo varía en el primer byte FE.

```
vnet0     Link encap:Ethernet  HWaddr FE:54:00:7D:78:ED
          inet6 addr: fe80::fc54:ff:fe7d:78ed/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:4624 errors:0 dropped:0 overruns:0 frame:0
          TX packets:108631 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:500
          RX bytes:360021 (351.5 KiB)  TX bytes:18974201 (18.0 MiB)
```

En el bridge podemos ver las interfaces definidas y asociadas al br0:

```
# brctl show
bridge name     bridge id               STP enabled     interfaces
br0             8000.4c72b9b0e3d0       no              eth0
                                                        vnet0
virbr0          8000.525400b82057       yes             virbr0-nic
```

Al igual que con Xen y OpenVZ, podemos verificar la configuración y asignaciones de direcciones IPv6 en el supervisor y en las MV utilizando los comandos ip -6 addr|route o ip -6 neigh. Para verificar la conectividad podemos usar ping6, traceroute6 y mtr -6, apuntando a direcciones IP destino dentro del segmento o prefijo de red y fuera de la LAN.

3.7_ Switches virtuales

El número de máquinas virtuales que corren sobre una máquina física ha crecido en función de la mayor capacidad de procesamiento y de virtualización que ofrecen la evolución permanente de los microprocesadores . En los primeros trabajos de investigación este número alcanzaba las 10 MV alojadas en cada máquina física con un desempeño aceptable. En 2010 este número aumento a 40 y llegó hasta 60 MV alojadas, y en la actualidad el número supera las 120 MV corriendo sobre una máquina real.

Este alto número de MVs ha creado el concepto de Centro de Datos Virtualizados, en los cuales el último salto de un paquete que atraviesa un switch sucede dentro del servidor (edge switch) y en la medida que el número de MV aumenta puede transformarse en un cuello de botella.

Por otro lado el uso de tecnologías de red de 10 Gbps en las troncales y en las capas de acceso y de agregación requieren de un alto desempeño por parte de las máquinas virtuales para abastecer el número de conexiones simultaneas, el ancho de banda de cada conexión, los retardos y latencia en la transferencia de los datos requeridos por la aplicación y el uso del procesador, memoria y disco que requieren los diferentes procesos que ejecutan en el requerimiento de un servicio.

El modelo tradicional de virtualización no es escalable cuando pensamos en implementaciones dentro de un Centro de Datos. Este modelo tradicional presenta un supervisor de capa de hardware que provee un modelo de virtualización de red simple, mediante un switch en capa 2 (L2) o a través de un router IP en capa 3 (L3), pero sin independizarse del uso del kernel.

Los switches virtuales presentan una evolución respecto al modelo tradicional, incorporando tecnologías que mejoran el plano de control y de visibilidad en la capa de red, al soportar protocolos de gestión de interfaces como SNMP y con acceso a linea de comandos (CLI) remoto sin necesidad de entrar previamente al supervisor. A esto se agregan otras capacidades como es disponer de más de una interfaz virtual (VIF) por MV, configurar VPN para diferenciar segmentos entre MV, migración de MV entre subredes, y tener acceso a la tabla de reenvío de paquetes (forwarding table) y manejar la salida a uno o más puertos, además de soportar IPV6.

3.7.1. Open vSwitch

Open vSwitch[12] es un switch desarrollado e implementado para ambientes virtualizados y se diferencia del modelo tradicional incluido en los kernels de los sistemas operativos, en que presenta una interfaz de control de reenvío de paquetes (forwarding) de grano fino, lo que permite implementar tecnologías que ofrecen los switches físicos como son la calidad de servicio (QoS), definición de túneles, reglas de filtrado, etc.

Open vSwitch ha sido incorporado en diferentes tecnologías de virtualización basadas en Linux como son Xen/XenServer, KVM, y VirtualBox y su arquitectura se muestra en la Figura 14. La versión 4.3 de Xen incorpora de base el paquete Open vSwitch para el manejo de la red virtual.

Una de las ventajas de Open vSwitch es que la mayor parte del código esta escrito en lenguaje C, lo que hace que sea independiente de la plataforma y es fácilmente trasladable a otros entornos. En cuanto a su funcionalidad como switch, presenta las siguientes características:

- Soporte del estándar de VLAN 802.1Q, con capacidad para definir puertos en modo acceso (access) y modo troncal (trunk).
- Protocolos de monitoreo por flujos: NetFlow y Sflow, y capacidad para espejado de puertos (port mirroring)
- Políticas de QoS (Calidad de Servicio)
- Configuración de Túneles: GRE, GRE sobre IPSEC, VXLAN y LISP
- Manejo de fallas de conectividad mediante el estándar 802.1ag
- Uso de extensiones de OpenFlow 1.0
- Alto desempeño en el reenvío de paquetes al usar un módulo del kernel en Linux.

El módulo es soportado en Linux kernel versión 2.6.18 o superior, siendo desde la versión 2.6.32 la más probadas para Xen con algunos parches sobre CentOS. También presenta soporte para Citrix XenServer y RedHat Enterprise.

Open vSwitch también puede ejecutarse en el espacio del usuario, sin la intervención del módulo del kernel, pero a un mayor costo en el rendimiento.

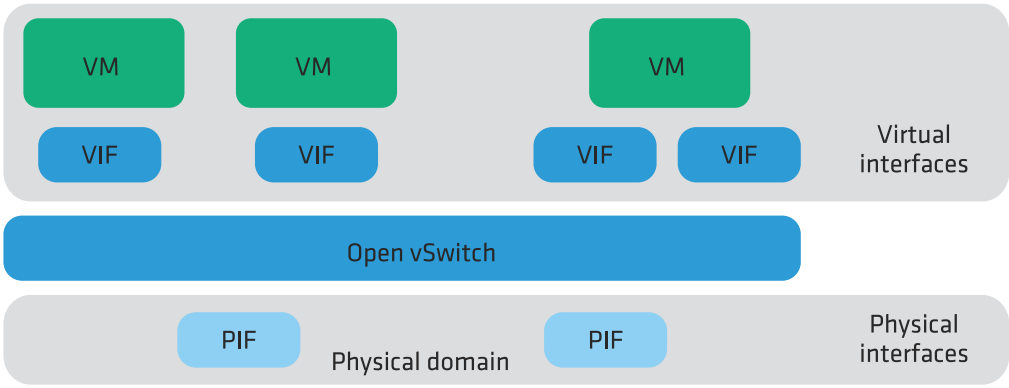

FIGURA 14: OPEN VSWITCH ARQUITECTURA

La especificación de OpenFlow 1.2 incluye un nuevo soporte para identificar flujos IPv6 además de permitir la re-escritura del encabezamiento del paquete IPv6 mediante el uso de estructuras más flexibles en el proceso de identificación de flujos. Esto permite identificar dentro de un flujo IPv6 la dirección de origen, la dirección destino, el número de protocolo, la clase de tráfico, el tipo de paquete ICMPv6, el código de ICMPv6, los campos del encabezamiento del NDP y la etiqueta de flujo (flow label) IPv6.

Open vSwitch presenta una serie de comandos que permiten monitorear y administrar el switch y operar sobre la tabla de flujos mediante el agregado, modificación y borrado de flujos. En el agregado del flujo es necesario especificar las características del flujo a identificar y luego definir una acción sobre ese flujo, como por ejemplo reasignar los puertos de salida, cambiar el identificador de una VLAN, disminuir el límite de saltos del paquete IPv6, etc.

La administración de los flujos IPv6 se realiza mediante el comando ovs-ofctl. El siguiente es un ejemplo de como agregar un flujo IPv6:

```
# ovs-ofctl add-flow br0 in_port=2,dl_type=0x86dd, \
ipv6_src=2001:db8::/32,ipv6_dst=2001:db8::/32,actions=out
put:5
```

Si analizamos la sintaxis del comando ovs-ofctl tenemos 4 partes a analizar:

- add-flow → agrega un flujo.
- br0 → identificador del bridge activo donde agregar el flujo a identificar.

- in_port=2,dl_type=0x86dd,ipv6_src=2001:db8::/32,ipv6_dst=2001:db8::/32 → Campos a comparar para identificar el flujo: puerto de entrada al OpenFlow bridge, flujo IPv6, dirección IPv6 origen, dirección IPv6 destino

- actions=output:5 → Acción a tomar sobre el fujo identificado: el paquete se envía al puerto 5 del OpenFlow bridge.

Para ver el flujo antes agregado, ejecutamos:

```
# ovs-ofctl dump-flows br0
       NXST_FLOW reply (xid=0x4):
       cookie=0x0, duration=3.214s, table=0, n_packets=0, n_bytes=0,
       ipv6,in_port=2,ipv6_src=2001:db8::/32,ipv6_dst=2001:db8::/32
       actions=output:5
```

Si queremos borrar el flujo antes agregado, ejecutamos:

```
# ovs-ofctl del-flows br0 in_port=2,dl_type=0x86dd, \
ipv6_src=2001:db8::/32, ipv6_dst=2001:db8::/32
```

Para más información ver en detalle el manual del comando ovs-ofctl[13].

3.7.2. Switches virtuales comerciales

Hay un número importante de fabricantes que desarrollaron sus propios switches virtuales para dar soporte y nuevas funcionalidades a las tecnologías de virtualización. Estas características los hacen similares al resto de los switches físicos, incorporando nuevos estándares y protocolos que facilitan la instalación y la administración de las máquinas virtuales en los centros de datos.

La mayoría ha sido desarrollado para dar soporte a VMware como tecnología de virtualización. Estos productos proveen diferentes capacidades y características que aumentan la prestación de los switches virtuales tradicionales que se ejecutan en el kernel.

VMware provee su propio producto de software que se llama VMware vNetwork Distributed Switch (VDS)[14]. Con este producto VMware supera a su predecesor vNetwork Standard Switch incorporando nuevas capacidades de administración, monitoreo y provisión a través de una interface centralizada, VLANs privadas y en el manejo de tráfico al agregar limitación en la velocidad de recepción de los puertos.

En colaboración con VMware, Cisco desarrolló el producto Cisco Nexus 1000V[15], un paquete de software que se instala en un hardware convencional para servidor y que agrega nuevas capacidades a las que presenta el propio VDS de VMware, en cuanto a la conectividad (LACP, Virtual Port Channels), al manejo de tráfico, a la Calidad de Servicio (DSCP, ToS), Seguridad (Listas de Acceso, RADIUS, DHCP snooping, ARP inspection), Monitoreo (SNMPv3, NetFlow v9), entre otras.

Otras empresas también han desarrollado productos similares diseñados para soportar el software de VMware. Uno de estos es el IBM Distributed Virtual Switch (DVS) 5000V, el cual también incorpora nuevas tecnologías y protocolos como son el Edge Virtual Bridging (EVB) basado en el estándar IEEE 802.1Qbg que permite la gestión escalable y flexible de la configuración de red y aplicar diferentes políticas por cada máquina virtual, eliminando muchos de los problemas de red introducidos con la virtualización tradicional de servidores.

Algo similar ocurre con HP y su producto FlexFabric Virtual Switch 5900v, el cual se basa en una solución integral de software y un switch físico en el ToR (Top-of-Rack) que mediante la tecnología Virtual Ethernet Port Aggregator (VEPA) permite asignar un puerto virtual de conexión a cada MV. Al igual que otros productos antes mencionados, el FlexFabric Virtual Switch 5900v es una solución de software diseñada para integrarse con el supervisor VMware ESX y como una alternativa al VMware vSwitch.

3.8_
IPv6 en centros de datos

Existen varias formas de introducir y operar IPv6 en Centros de Datos. Una forma es continuar con una operación IPv4 dentro del centro de datos y hacer algún tipo de translación en el borde, una segunda forma es usar la pila doble y una tercera es usar únicamente IPv6. En resumen tenemos:

1) Translación de IPv4 en el borde: En este escenario el centro de datos mantiene su infraestructura interna en IPv4 y hace algún tipo de translación a IPv6 en el borde. Usualmente este mecanismo se aplica sólo a servicios HTTP/HTTPS y se hace uso de proxies reversos.

2) Pila Doble: Aquí encontramos pila doble a través todos los servicios del centro de datos o al menos en los que presentan servicios a usuarios. También puede encontrarse pila doble solo en el borde mientras que las conexiones internas son IPv4 o IPv6 únicamente.

3) Solo IPv6: Esta es generalmente la etapa final de la transición de un centro de datos a IPv6. Aquí encontramos IPv6 en todos los elementos del centro de datos. Para ofrecer servicios a los usuarios legados de IPv4 se utiliza algún tipo de translación en el borde.

El uso de estos escenarios no es necesariamente en la forma secuencial descrita y tampoco ninguno es el mejor, el más correcto o el recomendado. Cada uno ofrece diferentes beneficios y desventajas que deben ser analizados para seleccionar la mejor opción.

3.8.1. Recomendaciones operativas para un centro de datos IPv6

A pesar de que IPv6 tiene más de diez años, aún es muy poca la experiencia operativa como para formular un grupo de mejores prácticas aceptadas universalmente. Sin embargo a continuación presentamos algunas consideraciones operativas a tomarse en cuenta para un centro de datos.

3.8.1.1. Direccionamiento

Existen varias consideraciones importantes en relación al direccionamiento en un centro de datos. La primera es que tipo de direccionamiento se debe usar; esto es Agregado por Proveedor (PA), Proveedor Independiente (PI) o direcciones Unique Local IPv6 (ULAs). En relación de PA vs PI, PI proveé una independencia del ISP y reduce los problemas con renumeración, sin embargo trae consigo el pago por una asignación del RIR y muy posiblemente otros costos extras de administración y operación.

En caso de usar ULAs, éstas deben solo usarse en infraestructura que no requiere acceso al Internet público como servidores de bases de datos, servidores de aplicación e interfaces de administración entre otros. En caso de además usar direccionamiento PA, el uso de ULAs puede disminuir el problema de renumeración.

Otro punto de debate es la longitud de los prefijos en el centro de datos. En general recomendamos el uso de subredes de 64 bits por cada VLAN o segmento de red. El uso de subredes de longitud mayor a 64 bits es aceptable siempre y cuando se conozcan los posibles inconvenientes como el romper SLAAC y tener que usar configuración manual. Finalmente los planes de direccionamiento deben seguir los principios de ser jerárquicos y poder agregar espacio. Se recomienda al menos el uso de un /48 por cada centro de datos.

3.8.1.2. Seguridad

La mayoría de los aspectos de seguridad de IPv6 se aplican a los centros de datos los cuales pueden encontrarse en[16]. Sin embargo un aspecto importante son los ataques a Neighbor Discovery Protocol (NDP). Este ataque es similar a los ataques de ARP de IPv4 y el atacante puede llenar el caché de vecinos y acabarse la memoria del enrutador resultando en la inhabilidad de éste para reenviar paquetes.

A pesar que el espacio de las subredes de 64 bits es muy grande para emplear un escaneo tradicional como en IPv4, existen algunos métodos que permiten reducir el espacio de escaneo. Si el escaneo es una preocupación para el administrador del centro de datos se recomienda no hacer uso de SLAAC y evitar asignar direcciones manualmente usando "low-byte" (i.e. de 0 a 256), direcciones basadas en IPv4 y direcciones que asemejen una palabra (i.e. bebe:cafe).

Aunque los centros de datos son ambientes controlados donde el uso de DHCP no es común y la imperzonalización de RA no es común, se recomienda el uso de herramientas que eviten el secuestro de RA (RFC 6104, RFC 6105) y DHCP[17].

Y sin mayor diferencia que en IPv4, también es necesario tomar todas las precauciones para evitar los ataques de amplificación y aplicar BCP38[18] en filtrado en el ingreso. Al mismo tiempo se debe enfatizar el uso de listas de control de acceso en los puntos de translación.

3.8.1.3. Monitoreo

El monitoreo es una operación crítica para las operaciones de cualquier red y debe ser hecho con el mismo cuidado en IPv6 y en IPv4. En el caso de centros de datos no son diferentes al hecho en cualquier otra red con IPv6. Es sin embargo importante considerar que los equipos de red y el software de monitoreo debe soportar IPv6 en la colección de datos (por ejemplo MIBs) a pesar de que el transporte de estos sea solo IPv4 (por ejemplo es posible recolectar información de IPv6 usando IPFIX a pesar de que los paquetes sean enviados usando transporte IPv4).

3.7.1.4. Sistemas de administración de red y aplicaciones

Los centros de datos pueden usar software para administrar sus operaciones como por ejemplo sistemas de administración de direcciones (IPAM), sistemas de provisionamiento y otra variedad de software de documentación y operación. Es importante que este software este preparado para soportar IPv6 en sus modelos de datos. En general, si IPv6 no ha sido soportado aún por estas aplicaciones los cambios pueden ser más complejos que agregar más espacio en los campos de entrada.

3.8.2. Motivaciones para un centro de datos solo IPv6

Existen varias motivaciones para considerar un centro de datos solamente IPv6. Primeramente tenemos la escasez de direcciones IPv4 puede obligar a tratar de rescatar direcciones donde no son totalmente necesarias. De la misma forma, esta limitación en obtener más direcciones IPv4 limitará el crecimiento de centros de datos en pila doble o en un ambiente IPv4 con translación en el borde.

Otra motivación es el ahorro de costos de administración, operación y mantenimiento que un ambiente de solo IPv6 puede traer en comparación con el manejo de dos pilas de direccionamiento. En principio los administradores de red deben de aprender dos pilas de protocolo, deben aplicar reglas de seguridad en duplicado y deben manejar operaciones en dos protocolos. Todo esto además de agregar un trabajo extra en la administración del centro de datos lo deja propenso a errores de configuración y huecos de seguridad.

Algunos otros factores que incrementan el costo de operación de centros de datos en pila doble son: El desarrollo, prueba y QA (Quality Assurance) de aplicaciones en dos pilas de protocolos; operación y administración de fallas; y administración y monitoreo de la red entre otros.

3.9_ Referencias

[1] M. Tim Jones, Virtual Linux. An overview of virtualization methods, architectures, and implementations. IBM DeveloperWorks article, December 2006

[2] The Xen Project, the powerful open source industry standard for virtualization, http://www.xenproject.org

[3] OpenVZ, http://www.openvz.org

[4] Timo Hirt, KVM - The kernel-based virtual machine, 2010

[5] VMware, http://www.vmware.com

[6] RFC4389, Neighbor Discovery Proxies (ND Proxy), http://tools.ietf.org/rfc/rfc4389.txt

[7] RFC2461, Neighbor Discovery for IP Version 6 (IPv6), http://tools.ietf.org/rfc/rfc2461.txt

[8] IPv6: Updates for net-2.6.19, http://lwn.net/Articles/200018/

[9] sysctl, https://www.kernel.org/doc/Documentation/networking/ip-sysctl.txt

[10] initscripts-ipv6, http://www.deepspace.net/initscripts-ipv6.html

[11] BenV's notes, Xen and routed IPv6, http://notes.benv.junerules.com/tag/xen/

[12] Open vSwitch, http://www.openvswitch.org

[13] ovs-ofctl - administer OpenFlow switches, http://openvswitch.org/cgi-bin/ovsman.cgi?page=utilities%2Fovs-ofctl.8

[14] Vmware vNetwork Distribution Switch, http://www.vmware.com/products/datacenter-virtualization/vsphere/distributed-switch.html

[15] Cisco Nexus 1000V Series Switches for VMware vSphere, http://www.cisco.com/en/US/products/ps9902/index.html

[16] Operational Security Considerations for IPv6 Networks",draft-ietf-opsec-v6. Chittimaneni, K., Kaeo, M., and E. Vyncke. 2013

[17] Network Reconnaissance in IPv6 Networks", draft-ietf-opsec-ipv6-host-scanning Gont, F. and T. Chown. 2013

[18] BCP38 Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. P. Ferguson, D. Senie. 2000

::

.4

Ruteo externo en IPv6

4.1_ Introducción

En distintos capítulos de este libro se tratan temas relacionados a la infraestructura interna de una organización, desde los planes de direccionamiento, el ruteo interno, configuración de equipamiento tal como servidores, firewalls, etc. En este capítulo veremos los aspectos concernientes al ruteo externo, es decir, los pasos que son necesarios cuando queremos vincular nuestra organización con otras en Internet, la forma de interactuar con ellas para intercambiar información de rutas y la manera en que hacemos abstracción de la complejidad interna de nuestra organización cuando publicamos información en Internet.

El protocolo que se utiliza actualmente para el ruteo externo en Internet es BGP. Dicho protocolo ha estado en uso por mas de 20 años para llevar la información de rutas de IPv4, llegando a un alto grado de maduración. Este hecho queda puesto de manifiesto en las tablas de ruteo globales, que hoy en día tienen alrededor de medio millón de rutas y son manejadas en forma eficiente por los routers y equipos que utilizan BGP. Con el tiempo, este protocolo se ha ido extendiendo, permitiendo transportar otro tipo de información mas allá de los prefijos IPv4 y es así como veremos que se ha extendido para poder manejar ruteo externo en IPv6.

En este capítulo no veremos una introducción a BGP, ni todas las posibilidades que brinda para la administración del ruteo externo, ya que se asume conocimiento previo acerca del protocolo. Sin embargo, se mencionarán algunos conceptos fundamentales con el fin de sentar las bases para una comprensión adecuada de los temas que se traten.

4.2_ Sistemas autónomos y routers de borde

Uno de los conceptos fundamentales de BGP es el de sistema autónomo. Por definición, un sistema autónomo es un conjunto de redes que comparten una política de ruteo en común. Esto es, una infraestructura que está gestionada de manera unificada, pudiendo ser vista desde afuera como una entidad unitaria.

Por otro lado, un sistema autónomo es totalmente "autónomo" de otros en Internet, pudiendo manejar internamente su infraestructura de red de una manera independiente y con la complejidad que requiera, mediante protocolos de ruteo interno, rutas estáticas, etc. Esta información interna puede ser muy extensa, involucrando más de una organización, incluyendo por ejemplo clientes de un ISP o instituciones miembro de una red académica, por citar algunos casos. Sin embargo, toda esa complejidad quedará para la parte "interna" del sistema autónomo y hacia

afuera sólo se verá un conjunto de rutas o prefijos que indicarán a otros sistemas autónomos las redes que componen el sistema autónomo.

Esta capacidad de "abstracción" de la información interna es otro concepto fundamental de BGP: evita superpoblar de información las tablas de ruteo globales y poder concentrarnos sólo en un esquema simplificado de la infraestructura interna.

En la siguiente figura se puede ver un esquema de una red con una infraestructura interna que involucra distintas subredes y routers: un sistema autónomo. Sin embargo, sólo algunos de esos routers estarán conectados con otros sistemas autónomos y "hablarán" BGP externo con routers de esos otros sitemas autónomos. Dichos routers son llamados routers de borde.

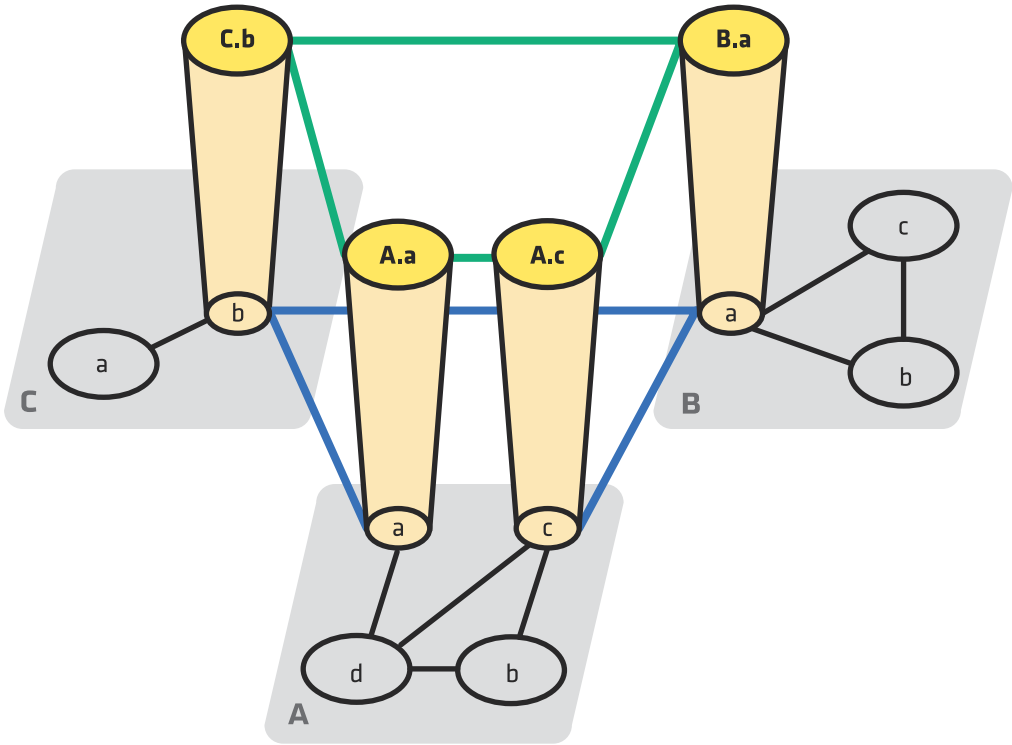

Figura 1: routers internos de cada sistema autónomo y routers de borde

Una característica importante de BGP, es la ausencia de jerarquía en el intercambio de información de ruteo entre sistemas autónomos. No hay desde el punto de vista técnico limitaciones en cuanto a las posibilidades de intercambio de rutas entre un sistema autónomo y otro. Cualquier organización puede interconectarse con otras en Internet, mas allá del tamaño o complejidad de su red interna. No hay, a priori, razones para que un sistema autónomo deba depender de otros, si bien veremos mas adelante que puede haber razones de tipo comercial que pueden influir en estas cuestiones.

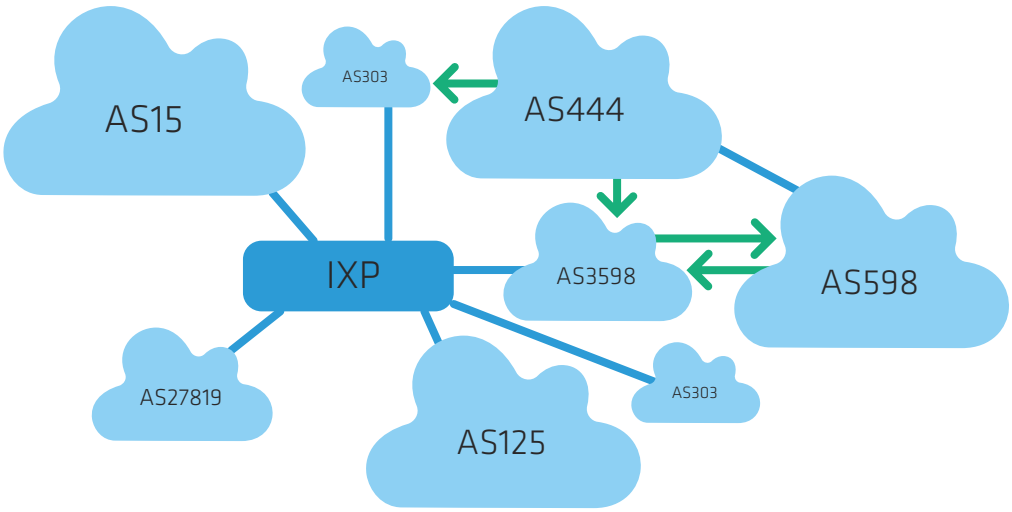

FIGURA 2: SISTEMAS AUTÓNOMOS DE DISTINTO TAMAÑO SE PUEDEN INTERCONECTAR ENTRE SÍ

4.3_
Aprendizaje y enseñanza de prefijos

Tal como dijimos, en BGP sólo informaremos acerca de las redes que un sistema autónomo contiene (o a las cuales se puede llegar a través de él). A su vez, necesitaremos obtener de otros sistemas autónomos el mismo tipo de información. De esta forma, el protocolo se basa en "anunciar" y "aprender" conjuntos de prefijos o redes. Es decir, nuestro sistema autónomo publicará hacia afuera todas las redes a las cuales queremos anunciar que es posible llegar a través de nuestro sistema autónomo. A su vez, aprenderemos de los otros sitemas autónomos con los cuales nos interconectemos, todas las redes a las que se puede llegar a través de ellos. Esta es la base del intercambio de información de ruteo externo en Internet y es similar en IPv4 e IPv6, como veremos.

4.4_
BGP multiprotocolo

Tal como mencionamos, el protocolo BGP existe desde hace mucho tiempo, permitiendo el intercambio de ruteo externo en IPv4. Con los años, se han agregado extensiones al mismo, para poder contemplar necesidades que no fueron previstas en un principio. Asi surge la extensión BGP multiprotocolo, definida en la RFC 4760 (que reemplaza a la RFC 2858), para poder llevar información de otros protocolos de red, además de IPv4. Estos agregados, no interfieren con routers que no soportan la capacidad, ya que el protocolo es compatible hacia atrás, permitiendo que las extensiones sean simplemente ignoradas por

aquellos routers que no las contemplan.

Para diferenciar entre la información propia de IPv4 y la de IPv6, se hace uso del concepto de "Address Family" (AFI), tal como se define en "Address Family Numbers", http://www.iana.org/numbers.html y se introduce el concepto de "Subsequent Address Family" (SAFI).

A su vez, se agregan dos atributos: Multiprotocol Reachable NLRI (MP_REACH_NLRI) y Multiprotocol Unreachable NLRI (MP_UNREACH_NLRI). El primero es utilizado para llevar el conjunto de rutas alcanzables junto con el "next hop" para esos destinos. El segundo para informar un conjunto de destinos inalcanzables. Ambos atributos son opcionales y no transitivos; de esa forma, un router BGP que no soporta estas extensiones, simplemente ignorará los atributos y no los pasará a otros vecinos BGP.

De esta manera, con las combinaciones entre AFI y SAFI, tenemos distintos grupos de información a transportar:

AFI = 1 → IPv4
SAFI = 1 → Unicast
SAFI = 2 → Multicast

AFI = 2 → IPv6
SAFI = 1 → Unicast
SAFI = 2 → Multicast

Al tener separada la información de IPv6 e IPv4 por address families, se obtienen topologías independientes. De esta forma, existirá una tabla BGP para IPv6 completamente separada de la tabla de BGP de IPv4, lo que permite diferenciar las consideraciones administrativas para una versión y otra del protocolo, aplicando en cada caso las políticas que correspondan. Habrá casos en que tendremos con el mismo neighbor sesiones tanto en IPv4 como IPv6, mientras que en otros casos, es posible que sólo tengamos configurada una sesión en IPv4 o en IPv6, de acuerdo a la topología de la red.

4.5_
Peerings y tránsito

Si bien anteriormente mencionamos que el ruteo entre sistemas autónomos no conlleva una jerarquía en Internet desde el punto de vista de diseño, en la práctica existen distintos tipos de Sistemas Autónomos según la función que cumplen. Es así que podemos diferenciar entre peering entre dos sistemas autónomos cuando ellos dos se ponen de acuerdo para intercambiar las rutas de cada uno entre sí. Por otro lado, podemos hablar de sistemas autónomos de tránsito, cuando dicho sistema autónomo permite a otros utilizar sus redes para llegar al resto de Internet (o a otros sistemas autónomos).

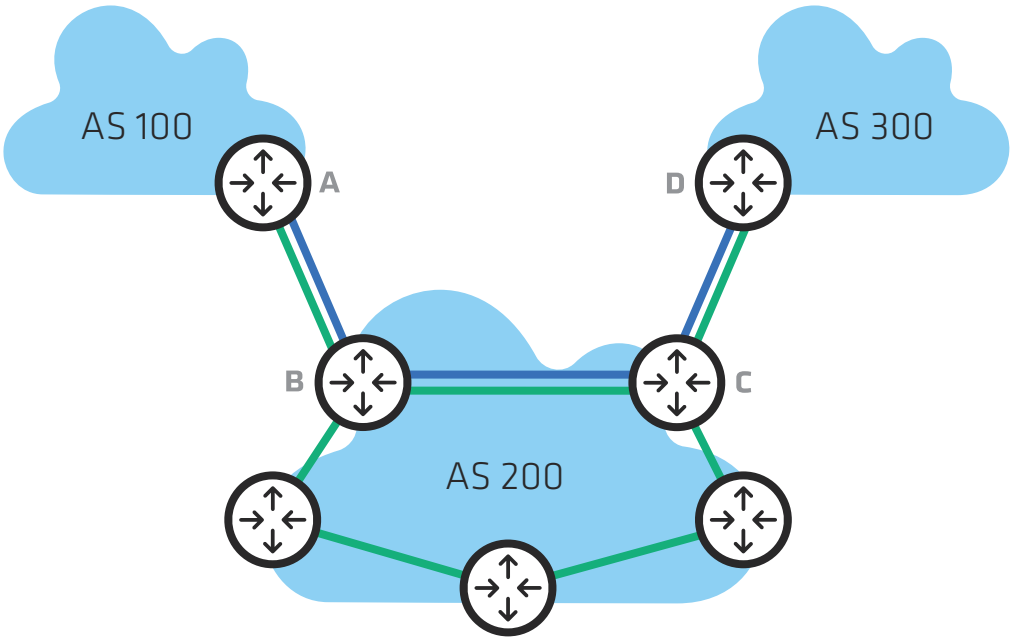

FIGURA 3: INTERCONEXIÓN DE SISTEMAS AUTÓNOMOS

Este concepto de "horizontalidad" en los sistemas autónomos se ve reforzado con la existencia de puntos de interconexión y de intercambio de tráfico. En ellos, es posible que muchas organizaciones se interconecten, facilitando y mejorando la calidad de la red global.

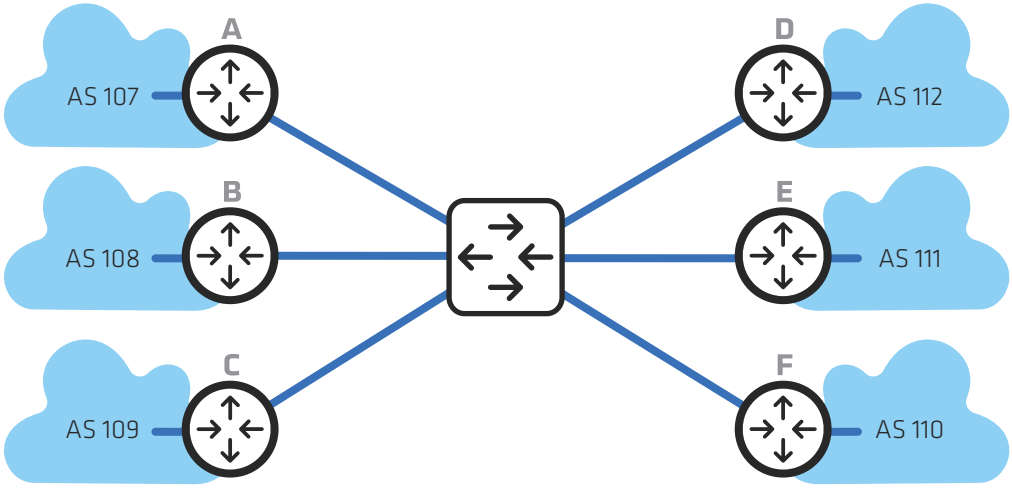

FIGURA 4: PUNTO DE INTERCAMBIO DE TRÁNSITO

En estos casos, habrá posibilidad de configurar sesiones BGP en IPv6 con aquellos routers que lo soporten. O, en el caso que existan route-servers centrales, éstos deberán soportar IPv6 para poder configurar las sesiones BGP.

En la arquitectura actual de Internet, no todas las organizaciones están dispuestas a intercambiar tráfico con el resto. Esto sucede en el caso de los grandes proveedores, que sólo tienen sesiones de peering con otros proveedores similares, mientras que a los proveedores de menor rango les cobrarán por el servicio de tránsito. De esta forma, en la práctica tendremos distintos niveles de proveedores, como muestra la figura:

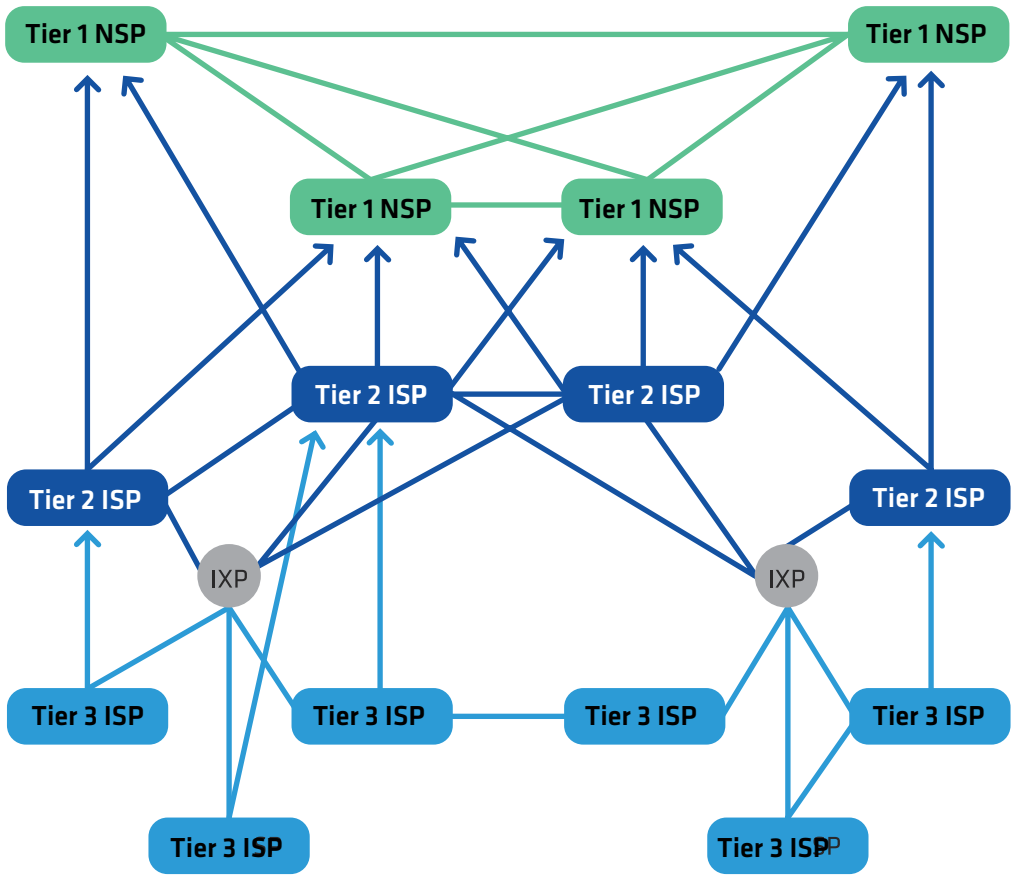

FIGURA 5: CLASES DE ISPS SEGÚN SU JERARQUÍA

Actualmente la totalidad de los proveedores tier1 soportan IPv6 pero no todos los nacionales o regionales. Esto dificulta establecer sesiones BGP en IPv6 con todos los proveedores. En algunos casos esto se puede resolver a través de un túnel con algún otro proveedor que sí soporte IPv6, si bien no estaremos utilizando una conexión nativa y puede no ser lo adecuado para nuestra organización.

4.5.1. Configuración de BGP

La configuración de BGP en IPv6 es muy similar a la de IPv4, mas allá de que, como fue mencionado, se utiliza una address-family distinta.

Las diferencias van a encontrarse fundamentalmente en todo lo que involucre direcciones IP, por ejemplo prefix-list, pero también en algunos detalles que se deben tener en cuenta al momento de configurar las sesiones BGP.

Veremos a continuación ejemplos de configuraciones en routers Cisco y Juniper, que son las plataformas mas difundidas. No obstante, estos ejemplos sirven de guía para otras plataformas que utilizan sintaxis similar.

4.5.2. Configuración básica

En los routers Cisco deberemos activar los vecinos IPv6 explícitamente dentro del "address-family IPv6". En general conviene utilizar el comando "no bgp default ipv4-unicast" para que no se intercambie información de IPv4 con los neighbours a menos que se configure explícitamente. También de esa forma daremos más uniformidad a la configuración, ya que los vecinos se definen en la configuración general de BGP y luego se activan en la correspondiente address-family.
El comando network, al igual que en IPv4, nos permitirá inyectar rutas en el BGP para poder anunciarlas a los vecinos. En este caso, la diferencia estará dada por la forma de especificar una red como prefijo en vez de utilizando máscara.

```
router bgp 64500
no bgp default ipv4-unicast
neighbor 192.0.2.1 remote-as 64501
neighbor 2001:db8:ffff::1 remote-as 64502
!
address-family ipv4 unicast
network 192.0.2.0 mask 255.255.255.0
neighbor 192.0.2.1 activate
no auto-summary
no synchronization
exit-address-family
!
address-family ipv6 unicast
network 2001:db8:abcf::/48
neighbor 2001:db8:ffff::1 activate
no synchronization
exit-address-family
```

Al igual que lo hacemos en IPv4, podríamos configurar peer-groups de vecinos IPv6 para agrupar aquellos que tengan políticas similares. Para esto deberemos definir el peer-group en la configuración general de BGP junto con los neighbours que pertenecerán a ese peer-group. Dentro de address-family IPv6 se deben activar los neighbours.

4.5.3. Configuración básica en Juniper

```
bgp {
    group external-peers {
        type external;
        export to-v6-peers ;
        peer-as 64502;
        neighbor 2001:db8:ffff::1 ;
        family inet6 {
            any ;
        }
    }
}
policy-options {
    policy-statement to-v6-peers {
       term allow {
           from {
               route-filter 2001:db8:abcf::/48 exact;
           }
           then {
               next-hop self;
               accept;
           }
       }
       term deny {
           then reject;
       }
    }
}
```

4.6_
Sumarización de rutas

Un punto importante a tener en cuenta en IPv6 es la sumarización de rutas. Si tenemos en cuenta que en IPv4 actualmente tenemos alrededor de medio millón de rutas, debido fundamentalmente a la desagregación de prefijos, en IPv6 tendremos potencialmente mucha más cantidad de rutas para anunciar, debido a la cantidad de direcciones disponibles. En muchos casos, la desagregación no es necesaria y sólo es producto de errores de configuración o de no tomarnos el trabajo necesario para realizar una configuración adecuada. Por esta razón, siempre trataremos de anunciar a nuestros vecinos las rutas sumarizadas a su máxima expresión. Por ejemplo, si tenemos un prefijo /32 y asignamos a nuestros clientes prefijos /48, no deberíamos publicar en Internet esos /48, sino solamente el /32. Para ello deberemos crear una ruta ficticia al bloque completo, que tenga una distancia administrativa mayor que todas las otras rutas y que descarte los paquetes.

4.6.1. En Cisco

```
ip route 2001:db8::/32 null 0 254
ip route 2001:db8:aaaa::/48 2001:db8:ffff:a::2
ip route 2001:db8:bbbb::/48 2001:db8:ffff:b::2
ip route 2001:db8:cccc::/48 2001:db8:ffff:c::2
!
router bgp 64500
  no bgp default ipv4-unicast
  ...
  address-family ipv6 unicast
  network 2001:db8::/48
  neighbor 2001:db8:ffff::1 activate

En Juniper:

routing-options {
    rib inet6.0 {
        static {
            route 2001:db8::0/32 {
                discard; install; readvertise;
            } ;
            route 2001:db8:aaaa::/48 next-hop 2001:db8:ffff:a::2 ;
            route 2001:db8:bbbb::/48 next-hop 2001:db8:ffff:b::2 ;
            route 2001:db8:cccc::/48 next-hop 2001:db8:ffff:c::2 ;
        }
    }
}
policy-options {
    policy-statement to-v6-peers {
       term allow {
           from {
               route-filter 2001:db8::/32 exact;
           }
           then {
               next-hop self;
               accept;
           }
       }
       term deny {
           then reject;
       }
    }
}
```

4.7_
Filtros por prefijos

Al igual que en IPv4, podemos definir filtros en BGP por direcciones o por números de AS. Los filtros por ASN son idénticos, ya que no tenemos en cuenta información que incluya las direcciones IP. Sin embargo, cuando filtramos anuncios de redes por prefijos específicos, debemos utilizar una variante del prefix-list: IPv6 prefix-list

4.7.1. En Cisco

```
ipv6 prefix-list ipv6-filtrar deny 3ffe::/16 le 128
ipv6 prefix-list ipv6-filtrar deny 2001:db8::/32 le 128
ipv6 prefix-list ipv6-filtrar deny fc00::/7 le 128
ipv6 prefix-list ipv6-filtrar deny fe80::/10 le 128
ipv6 prefix-list ipv6-filtrar deny ff00::/8 le 128
...
```

4.7.2. En Juniper

```
policy-options {
    policy-statement ipv6-filtrar {
        term deny-IPv6 {
            from {
                route-filter 3ffe::/16 orlonger
                route-filter 2001:DB8::/32 orlonger
                route-filter fc00::/7 orlonger
                route-filter fe80::/10 orlonger
                route-filter ff00::/8 orlonger
                ...
            }
            then {
                reject;
            }
        }
    }
}
```

4.8_ Consideraciones particulares para IPv6

Como mencionamos, la configuración de BGP en IPv6 no difiere en gran medida de la de IPv4, mas allá de las cuestiones lógicas referentes a las direcciones IP. Sin embargo, hay algunos puntos sutiles a tener en cuenta que se describen a continuación.

4.8.1. Router-id

Uno de los parámetros que el BGP necesita es el "BGP Identifier" (RFC6286). Este es un número de 32 bits que identifica al router y es intercambiado en los mensajes "OPEN" al establecer una sesión BGP. Dicho identificador debe ser único dentro de un sistema autónomo y normalmente se define en forma automática por una de las direcciones IPv4 del router. Sin embargo, en el caso que estemos configurando BGP en una red sólo IPv6, será necesario definir manualmente este identificador, para que puedan establecerse las sesiones BGP.

4.8.1.1. Cisco

```
router bgp 64500
  bgp router-id X.X.X.X
```

4.8.1.2. Juniper

```
routing-options {
  router-id X.X.X.X ;
}
```

4.8.2. Next-Hops

Normalmente cuando se establece una sesión BGP externa entre dos peers, existe una subred que comparten y por lo tanto existirán direcciones de tipo link-local que los interconectan y con las cuales es posible establecer la sesión BGP. Sin embargo, el next hop que un vecino BGP debe anunciar a otro debe ser una dirección IPv6 global, ya que de lo contrario, en caso de utilizar una dirección link-local, no se podría acceder al next-hop desde otras partes de la red.

El único caso en que es posible que una dirección link-local sea el next hop es cuando el peer BGP está en una misma subred compartida junto con el router que anuncia la ruta y el que la recibe (RFC 2545). Esto se ve esquemáticamente en la siguiente figura:

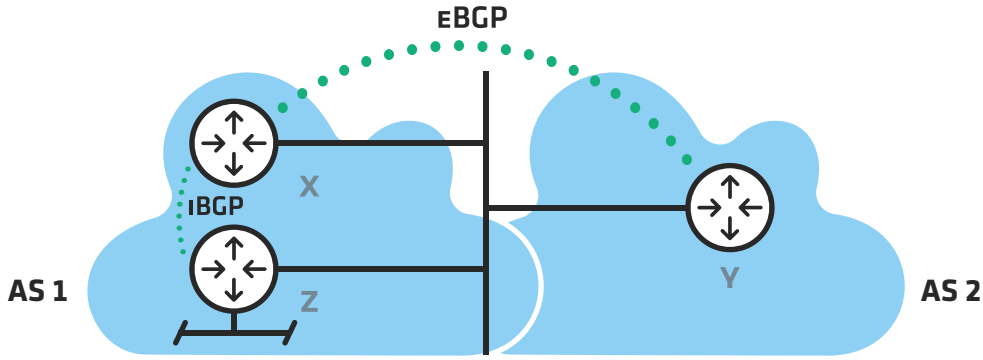

Figura 6: caso especial de next-hop en una red compartida

En este caso, los routers X, Y y Z comparten una subred, por lo que el next hop que el router X pase por iBGP al router Z podrá ser la dirección link-local del router Y.

En los demás casos, la dirección del next-hop deberá ser global, por lo que la recomendación es configurar direcciones globales unicast para los enlaces punto a punto sobre los que vamos a establecer sesiones BGP. Habitualmente se utiliza un rango reservado para esto, a partir del cual se van asignando las direcciones IPv6 que se utilizarán para los enlaces con otros vecinos externos.

4.9_ Conclusiones

Vimos en este capítulo que BGP continúa siendo el protocolo utilizado para intercambiar información de ruteo externo en Internet. Esto permite utilizar todas las herramientas desarrolladas durante años y aprovechar la estabilidad que el mismo posee.

Para configurar IPv6 es necesario utilizar las extensiones multiprotocolo para BGP, definiendo una nueva familia de direcciones o address-family, permitiendo contar con topologías independientes en IPv4 y en IPv6.

La configuración de BGP es similar a la de IPv4, salvando algunas particularidades que se mencionaron. Los filtros por direcciones IP mediante listas de prefijos son intuitivos y permiten continuar con las mismas prácticas que se utilizan en IPv4.

Otro tipo de filtros como los filtros por AS-PATH o cuestiones relacionadas a las políticas administrativas a aplicar, no difieren entre una versión del protocolo y la otra, por lo que no son parte de este capítulo.

Por último, conviene poner de relieve una vez mas la necesidad de configurar de manera adecuada los anuncios de BGP externos, de manera de no superpoblar las tablas de BGP de Internet. Por dicha razón se hace necesario comprender la necesidad de sumarizar la información que se expone hacia afuera de nuestra organización y en ese sentido, al comienzo de este capítulo se menciona el concepto de abstracción que un sistema autónomo lleva asociado. En el caso de IPv6, esto es aún más crítico dada la cantidad de prefijos posibles que podrían anunciarse a Internet si no se tiene el debido cuidado.

4.10_ Referencias

P. Marques, F. Dupont, RFC 2545: Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing, IETF. Ver en https://www.ietf.org/rfc/rfc2545.txt

T. Bates, Y. Rekhter, R. Chandra, D. Katz, RFC 2858: Multiprotocol Extensions for BGP-4, IETF. Ver en https://www.ietf.org/rfc/rfc2858.txt

Y. Rekhter, Ed, T. Li, Ed., S. Hares, Ed., RFC 4271: A Border Gateway Protocol 4 (BGP-4), IETF. Ver en https://www.ietf.org/rfc/rfc4271.txt

T. Bates, R. Chandra, D. Katz, Y. Rekhter, RFC 4760: Multiprotocol Extensions for BGP-4, IETF. Ver en https://www.ietf.org/rfc/rfc4760.txt

E. Chen, J. Yuan, RFC 6286: Autonomous-System-Wide Unique BGP Identifier for BGP-4, IETF. Ver en https://www.ietf.org/rfc/rfc6286.txt

“Address Family Numbers”, IANA. Ver en http://www.iana.org/assignments/address-family-numbers/address-family-numbers.txt

“Subsequent Address Family Identifiers (SAFI) Parameters”, IANA. Ver en http://www.iana.org/assignments/safi-namespace/safi-namespace.txt

BGP Enhancements for IPv6, Cisco Systems. Se puede encontrar en http://www.pacnog.org/pacnog5/track2/presos/IPv6-5.pdf

.5

IPv6 en Redes Móviles

5.1_ Introducción

Este capítulo describe las redes de acceso móviles y cómo IPv6 es incorporado en ellas. Comienza describiendo a IPv6 como protocolo necesario para soportar el crecimiento exponencial que tienen las redes móviles. Se muestran luego las arquitecturas de redes móviles actuales de 2G, 3G y LTE incluyendo sus componentes, descripciones y diagramas de los componentes básicos. Se estudia a continuación las distintas opciones de implementación. Para finalizar se muestra la configuración de los componentes necesarios para soportar IPv6 en la red del operador utilizando como ejemplo dos proveedores de equipos para redes móviles.

5.2_ La necesidad de IPv6 en redes móviles

La red de acceso móvil es uno de los componentes de Internet con mayor crecimiento. Cada día más y más personas acceden a Internet desde dispositivos como celulares y tabletas utilizando infraestructura de redes móviles. La cantidad de subscripciones móviles en Mayo de 2013 es de 6.8 mil millones y se espera que en 2014 supere la población mundial calculada en 7.1 mil millones de personas[1]. Esta cantidad de subscripciones supera ampliamente a otras estadísticas de necesidades básicas como se indica en[2]. Según mediciones de Akamai, un año después del lanzamiento Mundial IPv6, la mayoría de los requerimientos IPv6 provienen de equipos móviles[3].

Este crecimiento crea la necesidad de una mayor cantidad de direcciones IP para los dispositivos móviles pero la numeración actual de IPv4 no alcanza para asignarle a cada usuario una dirección IP. Entonces, el crecimiento escalable solamente puede estar acompañado por la asignación de direcciones IPv6 utilizando esquemas de transición de IPv4 a IPv6.

Pero no solamente existe la necesidad para personas físicas usando sus equipos sino que también existen terminales inteligentes que, conectadas a Internet, realizan transacciones con otras terminales inteligentes sin que haya interacción humana de por medio como se ven en redes machine-to-machine (M2M)[4]. Estas redes M2M generalmente utilizan la red móvil para su interconexión y se suman también al total de móviles, tabletas y otros dispositivos que necesitan una dirección IP en este tipo de redes. Algunos proveedores de equipos de red[5] [6] calculan que para el año 2020 habrá 50 mil millones de dispositivos conectados a Internet de los cuales su mayoría estarán conectados a redes móviles.

{

50 MIL MILLONES

de dispositivos conectados a internet para el año 2020 de los cuales su mayoría estarán conectados a redes móviles

5.3_
Arquitecturas de redes móviles

Se estudia a continuación los componentes básicos de tecnologías 2G, 3G y LTE involucrados en la transmisión de datos en redes de acceso móviles.

5.3.1. General Packet Radio Service

El General Packet Radio Service (GPRS) es un servicio de datos móviles orientado a paquetes para los sistemas de comunicación celular de 2G y 3G. El protocolo dominante en estas redes es IP y por consiguiente es un servicio best effort con distintas capacidades de ancho de banda. En el caso de 3G la velocidad mínima de transmisión es de 200Kbps pero los proveedores hoy ofrecen mayores velocidades de transmisión, en el orden de los Mbps, sobre este tipo de red utilizando protocolos como EDGE, HSDPA y HSDPA+.

Los componentes básicos de esta red se muestran en el siguiente diagrama en bloques:

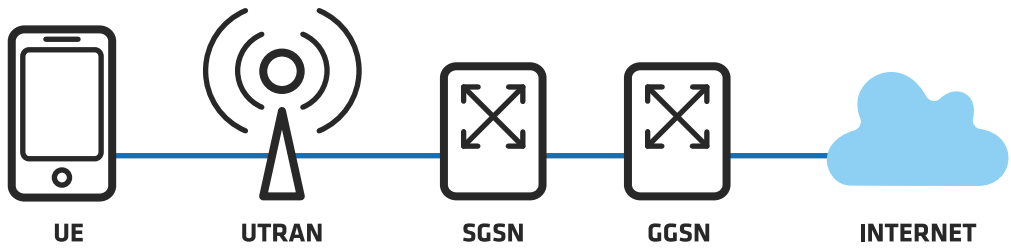

FIGURA 1: ARQUITECTURA GPRS BÁSICA

La descripción y funcionalidad de cada componente se describe a continuación:

- UE es el User Equipment es decir el dispositivo móvil.

- UTRAN es la red de radios donde están ubicados los Nodos B y los controladores de la red de Radio (RNC). Es decir, es la red que transporta tanto la voz como los datos del usuario hacia la red central de GPRS.

- SGSN es el nodo responsable por el ruteo de paquetes entre los usuarios móviles y el GGSN. Otra de sus funciones principales es manejar la movilidad de los usuarios.

- GGSN es el nodo principal de GPRS, actúa como gateway entre la red móvil y otras redes como por ejemplo Internet o redes internas del proveedor que estén basadas en IP. Puede utilizarse para soporte de autenticación de usuarios con elementos externos de la red GPRS.

5.4_
System Architecture Evolution

La SAE es la evolución de GPRS para el estándar de LTE también llamada 4G. Comparado con su antecesor, SAE simplifica la arquitectura utilizando solamente IP, es decir que es una red completamente basada en paquetes. A su vez soporta mayores velocidades, tiene menos latencia y es capaz de soportar movilidad entre distintas redes de acceso de radio incluyendo 3G. La señalización para conectarse a una red de datos también está simplificada como se verá más adelante. Los componentes básicos de la red son los que siguen:

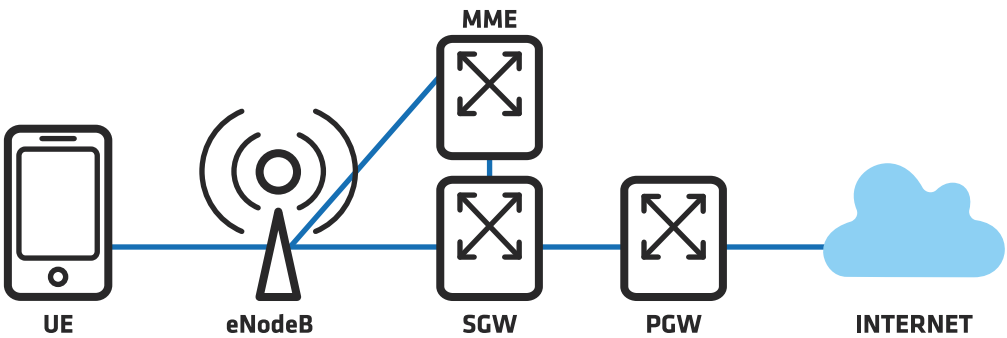

FIGURA 2: ARQUITECTURA SAE BÁSICA

Los componentes tienen aproximadamente las mismas funciones que en GPRS. El SGSN en esta arquitectura se llama MME y el GGSN se divide en dos: SGW y PGW. Se describen a continuación los distintos componentes de la arquitectura SAE.

- UE es el User Equipment, es decir el dispositivo móvil,
- eNodeB es la evolución del Nodo B de GPRS donde estos nodos incorporan capacidades de control del RNC, este último deja de existir simplificando la arquitectura de red.
- MME es responsable por el control de acceso a la red LTE incluyendo la autenticación del usuario. El MME es quien determina hacia qué SGW y cuál PGW se levantará el PDN.
- SGW es el nodo responsable por el ruteo de paquetes de los usuarios móviles. Otra de sus funciones principales es manejar la movilidad entre las redes LTE y otras del 3GPP como 3G.
- PGW tal como el GGSN, actúa como gateway entre la red móvil. Asimismo tiene capacidades para políticas de acceso, filtrado y análisis de paquetes.

La descripción completa de sus componentes se puede ver en[7].

5.5_
Planos de la red de acceso móvil

Ambas arquitecturas poseen tres planos donde tiene incidencia IP:
1) plano de gestión
2) plano de control
3) plano de datos

El plano de gestión sirve para monitorear y acceder remotamente a los equipos. Es interno a las operadoras y utiliza direcciones IP privadas. Dado que generalmente no tiene acceso público a través de Internet no es necesario cambiarlo a IPv6 para soportar a los usuarios.

El plano de control sirve para señalización entre nodos y, nuevamente, es interno a la operadora de red móvil. La capa de control usando GPRS Tunneling Protocol (GTP) posibilita la movilidad del usuario entre distintos nodos B ya que tuneliza los paquetes entre el UE y el GGSN. Esta capa generalmente utiliza direcciones IP privadas y en GPRS es interna al operador. Sin embargo en LTE, y para facilitar el roaming de los usuarios, las capas de control de los distintos operadores pueden necesitar comunicarse entre sí. En este caso IPv6 es fundamental ya que si se configuran direcciones IPv4 privadas puede haber solapamiento entre ellas y ser necesario un NAT para soportar la comunicación. Configurando la capa de control de LTE directamente en IPv6 evita los inconvenientes del NAT y permite un intercambio de información más fluida entre las distintas operadoras.

El plano de datos o plano de usuario es donde se encuentra el impacto inmediato de la falta de direcciones IP y es donde es necesario configurar direcciones IPv6 de inmediato. Es en este plano donde se asignan direcciones IP a los usuarios móviles que luego se usarán para conectarse a redes de datos desde la red móvil.

Se muestra a continuación las pilas de protocolos e interfaces en los elementos de red GPRS. Las capas en SAE son las mismas con algunas diferencias funcionales en los protocolos de señalización, los nombres de las interfaces también cambian pero no hacen efecto para la configuración básica de IPv6.

Se han identificado en la figura siguiente las capas que utilizan IP. La capa azul es la capa IP para los protocolos de control donde básicamente se configura un túnel entre el UE y el GGSN que permite movilidad, y en verde la capa IP para las aplicaciones de los usuarios como ser HTTP, SMTP, etc.

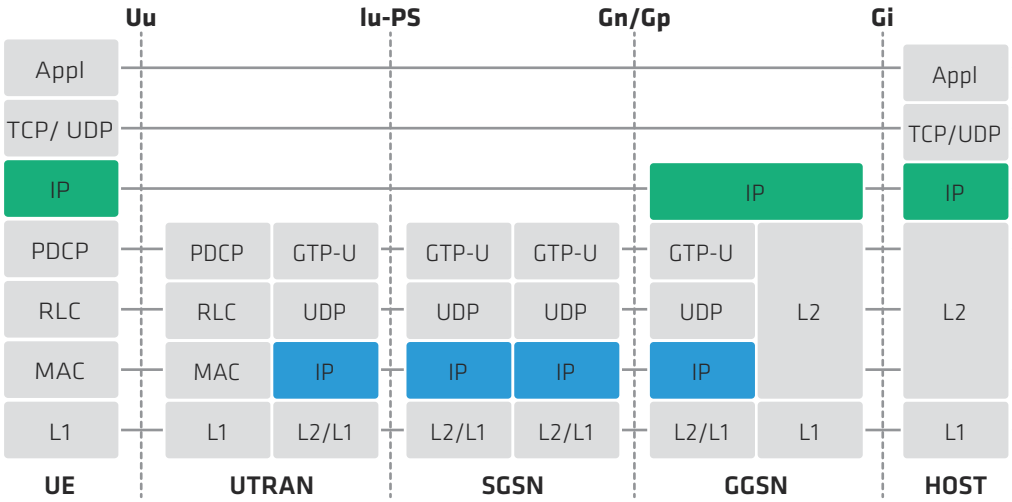

FIGURA 1: PILA DE PROTOCOLOS EN GPRS

5.6_
Access Point Name

El Access Point Name (APN) es simplemente un nombre que indica el gateway o punto de acceso a otras redes. Generalmente para las redes móviles se define para acceso a Internet, es decir que define el servicio de Internet, pero puede utilizarse para otras redes como por ejemplo redes internas de la empresa proveedora del servicio móvil.

El APN tiene una estructura definida en[8] y consiste en dos partes:

1) Network Identifier; y
2) Operator Identifier; pudiéndose prescindir de este último.

Algunos ejemplos son:

- Internet
- Internet.Operador1.com
- RedInterna.Operador2

Este nombre es resuelto por un DNS interno de la red del operador y conecta el UE al GGSN correspondiente que le dará acceso a la red final.
Packet Data Protocol (PDP) Context

Una vez que el UE conoce el GGSN donde tendrá acceso a la red deseada como por ejemplo Internet, se debe activar el PDP Context. Éste es una estructura de datos que contiene la dirección IP asignada al usuario, su IMSI y parámetros para el túnel de datos. EL PDP context puede ser de distintos tipos para IP y los siguientes son soportados:

- IPv4
- IPv6
- IPv4v6 (dual stack)

Es decir que un APN puede definirse según el tipo de versión IP que soporte el PDP context. En la siguiente figura podemos ver los distintos tipos: APN1, soporta solamente contextos PDP IPv6; APN2 soporta contextos IPv6, IPv4 o Dual Stack; y APN3 solamente IPv4. En el ejemplo de la figura, cada APN está relacionado a una VRF en particular.

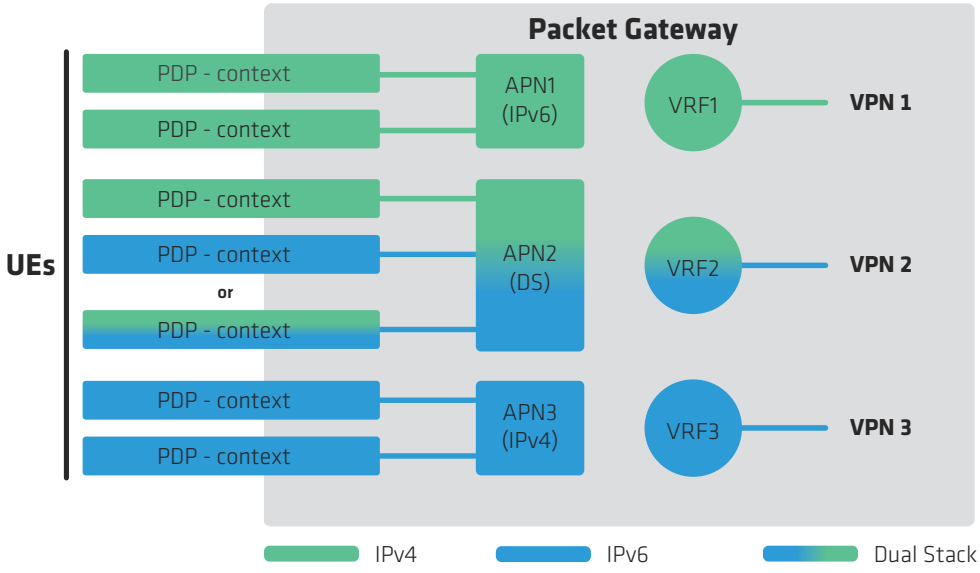

FIGURA 4: CONTEXTOS PDP PARA IP

El PDP type a elegir para soportar IPv6 debe coincidir con las políticas de la operadora que lo aplique. Muchas eligen utilizar un PDP type dual stack ya que se configura un solo PDP context y se ahorra en licencias de usuarios en el sistema en general. También esto ayuda ya que por lo general los operadores de red ya tienen un backbone dual stack y no dos redes separadas para cada protocolo IP. También tiene como ventaja el seguimiento de problemas ya que si utilizáramos dos APN que estén en dos gateways distintos podríamos tener problemas en uno de los protocolos y en otro no, haciendo difícil este seguimiento aparte de duplicar la infraestructura de acceso a las redes IP.

5.7_ Evolved Packet System (EPS) Bearer Setup

Si bien en las redes GPRS es necesario activar PDP contexts, en las redes LTE la activación del equivalente al PDP context es automática, el EPS bearer. Esto sigue la lógica de que si el UE quiere conectarse a un APN es porque necesita transmitir datos. Es decir que la conexión a redes de datos en LTE requiere menos señalización que en la de GPRS[9]. El PDN type del EPS bearer también puede ser IPv4, IPv6 o IPv4v6.

5.8_
IPv6 en otros componentes de la red móvil

Si bien la asignación de IPv6 llega de la mano de la falta de direcciones IPv4 y permite el crecimiento de la cantidad de usuarios soportados por nuestra red, no debemos olvidar que al ser un nuevo protocolo y tener otro formato la configuración del mismo impacta en otros sistemas de nuestra red que pueden estar vinculados directa o indirectamente en el tráfico de paquetes entre el equipo del usuario y la red final de destino.

Los clientes generalmente tienen un ancho de banda máximo asignado tanto de bajada como de subida según los distintos planes que hayan contratado. Ese ancho de banda generalmente es independiente del protocolo IP y la mayoría de los equipos hoy soportan aplicar un límite de velocidad indistintamente a paquetes IPv4 e IPv6. Pero se puede tener políticas que apliquen distintos anchos de banda dependiendo de la dirección origen/destino del paquete o políticas con permisos distintos basadas en estas direcciones. En estos casos se deben revisar esas políticas de acceso en el PCRF para configurar las distintas reglas ya aplicadas a IPv4 contemplando IPv6.

Los sistemas de monitoreo y control de la red, provisión de servicio, es decir el OSS del operador, deben también ser tenidos en cuenta y modificados para el soporte de IPv6.

5.9_
Opciones de implementación

Las opciones de implementación de IPv6 son muy parecidas a las de redes fijas, y las consideraciones sobre el despliegue de IPv6 en las redes móviles se pueden encontrar resumidas en[10]. Las opciones más recomendables de implementación, según el autor, son dos:
1) Dual Stack con NAT44 y,
2) IPv6 solamente con NAT64.

Muchos operadores de redes móviles han adoptado alguno de estos dos esquemas dejando de lado las soluciones con túneles.

5.9.1. Implementación Dual Stack

Esta opción es indicada para una red ya existente en producción. El NAT44 si bien es optativo, muchas empresas ya lo están implementando inclusive desde antes de que hubiera escasez de direcciones IPv4. El esquema está diseñado con un PDP context dual stack por usuario y un NAT44 que se recomienda que esté separado del PGW como se muestra a continuación.

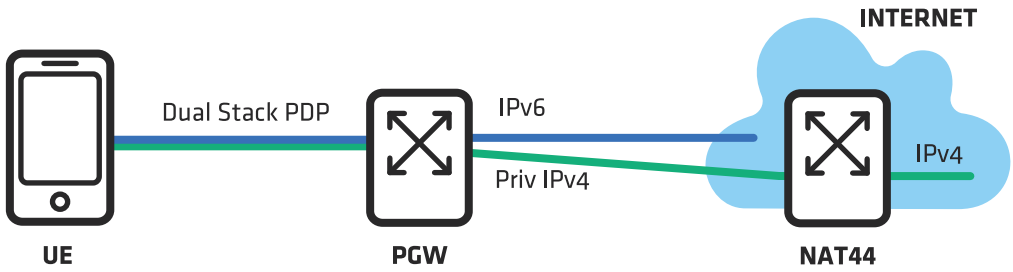

FIGURA 5: IMPLEMENTACIÓN DE DUAL STACK

Los paquetes IPv6 luego en el backbone serán ruteados al destino correspondiente mientras que los de IPv4 lo serán luego de aplicarles NAT a direcciones públicas.

5.9.2. Implementación IPv6 solamente

La opción de IPv6 está recomendada a nuevas tecnologías de red o cuando la mayoría de los sitios actuales de Internet se manejen con IPv6. Como ejemplo de nuevas tecnologías, un operador puede tener en funcionamiento su red 3G e implementar Dual Stack en la misma e implementar solamente IPv6 en el despliegue de su red LTE. Para la comunicación con los hosts IPv4 se utiliza un NAT64 como se muestra en la siguiente figura.

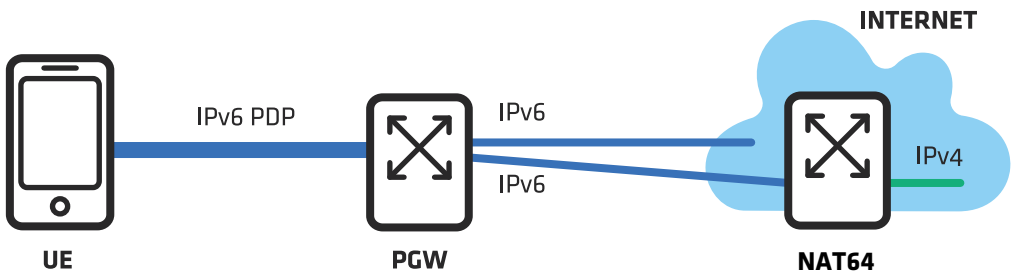

FIGURA 6: IMPLEMENTACIÓN DE IPV6 SOLAMENTE

En este caso los paquetes de IPv6 con destino IPv6 serán enviados directamente al backbone mientras que los que por DNS64 requieran contactarse con destinos IPv4 atravesarán por un NAT64.

Una de las ventajas que tiene implementar solamente IPv6 es que se puede hacer un mejor manejo de la red, el seguimiento de los problemas y simplifica el diseño de la red. La desventaja que tiene hoy una implementación como esta es que el tráfico de IPv6 y sitios que soporten IPv6 todavía es muy bajo a pesar de que los tres sitios de Internet más visitados según Alexa, Google, Facebook y YouTube[11], ya soportan completamente IPv6.

En ambos casos de implementación, el NAT será cada vez menos utilizado cuanto más los usuarios y servicios utilicen únicamente IPv6.

5.10_ Configuraciones

5.10.1. Configuración de IPv6 en el plano de datos

Para configurar los parámetros en el APN relacionados con la asignación de direcciones IP y que son obligatorios, hay que seguir los siguientes pasos:

- Configurar el nombre del APN y habilitar los tipos de PDP y PDN
- Configurar el rango de direcciones IP para los UE

El siguiente paso relacionado a la asignación de direcciones IP es opcional:

- Configurar servidores de DNS

Este capítulo se concentrará en la configuración de estos parámetros en EPG en equipos Juniper M120 y Ericsson Smart Services Router (SSR). Los comandos para GGSN son los mismos que los de EPG para ambas plataformas excepto cuando son indicados.

5.10.2. Configuración del nombre del APN y habilitación de tipos de PDP y PDN

Un nombre debe ser especificado para cada APN. El nombre debe corresponder al formato de identificador de red descripto en[8].

Los APNs pueden ser configurados para permitir los siguientes tipos de PDP y PDN:

- IPv4 solamente
- IPv4 e IPv6
- IPv6 solamente
- IPv4 e IPv6 simultáneamente (PDP dual stack)

La configuración en el M120 es la siguiente:

```
[edit services epg pgw]
set apn $apn-name pdp-context pdp-type (ipv4 | ipv4-ipv6 | ipv6 |
ipv4v6);
```

La configuración en el SSR es la siguiente:

```
(config-ManagedElement=1,Epg=1,Pgw=1,Apn=$apn-name,PdpContext=1)
  pdpType=(ipv4 | ipv4-ipv6 | ipv6 | ipv4v6)
```

5.10.3. Configuración del rango de direcciones IP

Para cada APN un rango de direcciones IP debe ser asignado. Durante la sesión PDP o el establecimiento de la conexión PDN, el EPG verifica si las direcciones IP asignada al UE están dentro del rango especificado por el APN. Si lo está, entonces el EPG acepta la sesión PDP o el establecimiento de la conexión PDN. Si no está en el rango rechaza la sesión o el establecimiento de la conexión.

Los rangos de direcciones IPv4 pueden ser asignado entre /8 y /30 y los de IPv6 entre /44 y /62. Por default IPv4 entrega prefijos /32 e IPv6 entrega /64 para ambas plataformas.

La configuración en el M120 es la siguiente:

```
[edit services epg pgw apn $apn-name pdp-context]
address $address-range;
ipv6-address $ipv6-address-range;
```

La configuración en el SSR es la siguiente:

```
(config-ManagedElement=1,Epg=1,Pgw=1,Apn=$apn-name,PdpContext=1)
  Address=$address-range
  Ipv6Address=$ipv6-address-range
```

Si se requiere que un APN tenga un solo tipo de rango, entonces es suficiente con configurar uno de ellos.

5.10.4. Configuración de DNS

Cuando un PDP context o un EPS bearer es establecido para un APN, la dirección de los DNS es enviada al UE. Esta información puede estar provista por el EPG a través de un servidor DHCP, RADIUS o localmente en el APN. La prioridad de asignación la tiene el DHCP, luego el RADIUS y por último los DNS configurados localmente. Se puede asignar uno o dos DNS por APN.

La configuración en el M120 es la siguiente:

```
[edit services epg pgw apn $apn-name]
name-server {
  $dns1-address;
  [$dns2-address;]
}
ipv6-name-server {
  $dns1-ipv6-address;
  [$dns2-ipv6-address;]
}
```

La configuración en el SSR es la siguiente:

```
(config-ManagedElement=1,Epg=1,Pgw=1,Apn=$apn-name)
   NameServer=$dns1-address
      priority=$dns1addressPriority
   up
   [NameServer=$dns1-address]
      priority=$dns2addressPriority
   up
   Ipv6NameServer=$dns1-ipv6-address
      priority=$dns1-ipv6-addressPriority
   up
   [Ipv6NameServer=$dns2-ipv6-address]
      priority=$dns2-ipv6-addressPriority
```

5.10.5. Ejemplos de APN con PDP IPv6

Configuración de APN con PDP IPv6 en M120:

```
[edit services epg pgw]
apn ipv6only {
      pdp-context {
                  pdp-type ipv6;
                  ipv6-address 2001:db8:db8:/48;
     }
     ipv6-name-server {
                      2001:db8::3434;
                      2001:db8::3435;
     }
}
```

Configuración de APN con PDP IPv6 en SSR:

```
(config-ManagedElement=1,Epg=1,Pgw=1)
 apn = ipv6only
  pdpContext = 1
   pdpType = ipv6
   Ipv6Address = 2001:db8:db8:/48
   up
  Ipv6NameServer = 2001:db8::3434
   Priority = 1
   up
  Ipv6NameServer = 2001:db8::3435
   Priority = 2
```

5.10.6. Ejemplos de APNs con PDP dual stack

Configuración de APN con PDP IPv4v6 en M120:

```
[edit services epg pgw]
apn dualstack {
      pdp-context {
                  pdp-type ipv4v6;
                  address 192.0.2.128/25;
                  ipv6-address 2001:db8:db8:/48;
     }
     name-server {
                 192.0.2.3;
                 192.0.2.5;
     }
     ipv6-name-server {
                      2001:db8::3434;
                      2001:db8::3435;
     }
}
```

Configuración de APN con PDP IPv4v6 en SSR:

```
(config-ManagedElement=1,Epg=1,Pgw=1)
 Apn = dualstack
  pdpContext = 1
   pdpType = ipv4v6
   Address = 192.0.2.128/25
   Ipv6Address = 2001:db8:db8:/48
   up
  NameServer = 192.0.2.3
   Priority = 1
   up
  NameServer = 192.0.2.5
   Priority = 2
   up
  Ipv6NameServer = 2001:db8::3434
   Priority = 1
   up
  Ipv6NameServer = 2001:db8::3435
   Priority = 2
```

5.10.7. Configuración de IPv6 en los planos de control y gestión

Estos planos se pueden ver como una red IP donde los nodos se comunican entre ellos a través de este protocolo. En estos planos, la configuración es simplemente asignar direcciones IPv6 a las interfaces que comunican los distintos nodos. El ruteo de los paquetes es realizado por un IGP o simples rutas estáticas, es decir que el ruteo también debe soportar IPv6 y será configurado según los parámetros presentados en otros capítulos de este libro.

5.14_ Soporte de IPv6 en dispositivos móviles

El soporte de IPv6 en las diversas marcas de dispositivos móviles, que incluyen teléfonos y dongles, depende del sistema operativo que tengan instalado y el modelo en particular. En general los sistemas operativos iOS, Android y Windows Phone soportan IPv6[13] pero cada modelo de teléfono depende del fabricante del mismo. Dada la velocidad con que las empresas colocan en el mercado nuevos y mejores dispositivos móviles de la mano de la exigencia del mercado, es recomendable verificar los dispositivos directamente con los fabricantes.

5.15_ Conclusiones

Como se indica al comienzo de este capítulo, IPv6 es el protocolo que posibilita el crecimiento actual de los usuarios móviles y dispositivos M2M que usan esta red para su comunicación. Sin IPv6 el riesgo de no poder continuar dando servicios a los usuarios es muy alto. Las distintas arquitecturas de redes móviles como ser la de 3G y LTE presentadas aquí, están diseñadas para soportar IPv6, los proveedores de equipos para esta red ya soportan este protocolo y los operadores de red comienzan a desplegar IPv6 en sus redes ya instaladas o próximas a instalarse.

Si bien los esquemas de transición de IPv4 a IPv6 son variados y tratados en otros capítulos, para las redes móviles se presentan dos esquemas recomendados para el plano de datos: Dual Stack e IPv6 solamente. El primero de los esquemas es útil para redes ya desarrolladas como las 3G mientras que el segundo es recomendado para nuevos desarrollos como LTE. Asimismo se recomienda configurar IPv6 en los distintos planos siguiendo la urgencia y necesidad del mismo: comenzando por el plano del usuario donde la necesidad es más obvia, siguiendo por el plano de control que en el caso de LTE es útil para conectar redes móviles de distintos operadores, y finalizando por los planos de gerenciamiento de equipos.

5.16_
Acrónimos

3GPP	3rd. Generation Partnership Project
EDGE	Enhanced Data rates for GSM Evolution
GGSN	Gateway GPRS support node
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communications
HSDPA	High-Speed Downlink Packet Access
IMSI	International Mobile Subscriber Identity
LTE	Long Term Evolution
MME	Mobility Management Entity
NAT	Network Address Translation
OSS	Operations Support System
PCRF	Policy Charging and Rules Function
PDN	Packet Data Network
PGW	PDN Gateway
RNC	Radio Network Controller
SAE	System Architecture Evolution
SGSN	Serving GPRS support node
SGW	Serving Gateway
UE	User Equipment
UTRAN	Universal Terrestrial Radio Access Network

5.17_
Referencias

[1] "Mobiles 'to outnumber people next year', says UN agency," BBC News; disponible en http://bbc.in/174o1uW; Internet; accedido el 6 de Julio de 2013.

[2] Wang, Yue; "More People Have Cell Phones Than Toilets," Time Newsfeed; disponible en http://ti.me/YBdpyh; Internet; accedido el 6 de Julio de 2013.

[3] Nygren, Erik, "World IPv6 Launch Anniversary: Measuring Adoption One Year Later", Akamai blog; disponible en http://bit.ly/12ZVoie; Internet; accedido el 3 de Septiembre de 2013.

[4] "Machine to machine," Wikipedia; disponible en http://bit.ly/QOftLQ; Internet; accedido el 6 de Julio de 2013.

[5] "More than 50 billion connected devices – taking connected devices to mass market and profitability," Ericsson; disponible en http://bit.ly/gvWIvi; Internet; accedido el 6 de Julio de 2013.

[6] "M2M: the Internet of 50 billion devices," Huawei; disponible en http://bit.ly/14UmROe; Internet; accedido el 6 de Julio de 2013.

[7] General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access. 3GPP Technical Specification 23.401 Release 12.

[8] Numbering, addressing and identification. 3GPP Technical Specification 23.003 Release 11.

[9] "PDP Context vs. EPS Bearer – A Battle of the Data Session Setups," LTE University; disponible en http://bit.ly/16qqhHQ; Internet; accedido el 2 de Septiembre de 2013.

[10] Koodli, R., Mobile Networks Considerations for IPv6 Deployment, RFC 6342, IETF.

[11] "The top 500 sites on the web," Alexa; disponible en http://www.alexa.com/topsites; Internet; accedido el 3 de Septiembre de 2013.

[12] "Comparison of IPv6 support in operating systems," Wikipedia; disponible en http://bit.ly/KPYOZH; Internet; accedido el 3 de Septiembre de 2013

.6

Mecanismos de transición

6.1_
Introducción

Este capítulo trata sobre la transición de IPv4 a IPv6. Se han desarrollado diversas **tecnologías con el objetivo de permitir –o facilitar– la transición entre los protocolos** en Internet y en las redes. El objetivo principal de este capítulo es analizarlas, aportando elementos que permitan comprender el principio de funcionamiento y los casos de uso de cada una, permitiendo escoger la mejor técnica para cada caso.

El texto comienza con un breve análisis de lo que se podría denominar el **plano de transición**, partiendo de la idea que se tenía del mismo en las décadas de 1990 y 2000, para llegar a la visión contemporánea de lo que debe ser la migración a IPv6. Luego se analizan los mecanismos de transición en sí.

Con fines didácticos, en este trabajo clasificaremos a estos mecanismos en dos grandes grupos. El primero se refiere al contexto del plano de transición original. A las técnicas que pertenecen a este primer grupo les llamaremos **tradicionales**. El segundo grupo se refiere a un contexto más actual de transición. A las técnicas que pertenecen a este segundo grupo les llamaremos **nuevas**. No obstante, este trabajo no pretende abordar todas las técnicas que existen, ya que su cantidad es demasiado elevada. Con el paso del tiempo, algunas evolucionaron y dieron origen a técnicas nuevas, otras fueron abandonadas o consideradas obsoletas. Las técnicas seleccionadas son las que consideramos serían de mayor utilidad para los operadores de red. Se incluyeron excepciones cuando se juzgó que era importante conocerlas debido a que tienen implicancias de seguridad para las redes o para Internet, o porque su comprensión ayuda a entender mejor alguna otra técnica.

6.2_
La transición de IPv4 a IPv6

El primer punto a tener en cuenta es que la dificultad de la transición de IPv4 a IPv6 radica en el hecho de que ambos protocolos son incompatibles entre sí. Esta incompatibilidad fue una decisión de proyecto. Durante la década de 1990, en algún momento del proceso de creación de la nueva generación del protocolo de Internet, se pensó que la creación de un protocolo no compatible permitiría incorporar características importantes. Las ventajas asociadas con estas características compensarían una potencial mayor dificultad en la transición. Con el advenimiento de técnicas que ayudaban a conservar los recursos IPv4, tales como CIDR (Classless Inter-Domain Routing), NAT (Network Address Translation) y DHCP (Dynamic Host Configuration Protocol) y el uso de direccionamiento privado, se creyó que esta transición podría darse en el correr de muchos años y se vislumbró una solución muy sencilla desde el punto de vista técnico.

La solución planeada para la transición de IPv4 a IPv6 fue el uso de la doble pila. El protocolo IPv4 se seguiría usando normalmente y, paulatinamente, en cada componente de Internet se iría desplegando IPv6. Esto tendría lugar a lo largo de una década, quizás dos. Ciertamente, en algún momento antes del agotamiento de IPv4, el protocolo IPv6 estaría presente en todos los elementos de la red. Así, IPv4 se volvería innecesario y poco después este protocolo sería abandonado.

Aquí podemos adelantar que este plan no funcionó. Incluso considerando este contexto original en el cual se pensaba que IPv6 se implementaría en toda Internet mientras todavía hubiera suficientes direcciones IPv4 disponibles, de todos modos sería necesario utilizar técnicas de transición auxiliares. Estas técnicas son las que aquí se clasifican como tradicionales. Básicamente, estas técnicas intentaban resolver un problema: **interconectar redes IPv6 utilizando túneles sobre una red que es predominantemente IPv4.**

Es importante notar que la idea de que la transición de IPv4 a IPv6 sería gradual y de que ambos protocolos convivirían en las redes por muchos años se concibió en este contexto, que al día de hoy ha perdido validez. Se podría argumentar que, en realidad, IPv6 se viene implementado desde hace años y que esta convivencia entre ambos protocolos ya lleva varias décadas. Sin embargo, este argumento es cuestionable, ya que en muchos aspectos el despliegue de IPv6 todavía puede considerarse incipiente. Lo importante es comprender que, hoy en día, la expectativa es que la transición de Internet a IPv6 se haga rápidamente.

Aunque el plan original era técnicamente sencillo y elegante, este plan fracasó porque no se previeron las consecuencias administrativas y financieras de la implementación de IPv6. Para la mayor parte de las corporaciones involucradas, IPv6 se podía considerar una tecnología que solo traería beneficios a largo plazo, por lo que su implementación

no requería nuevas inversiones y recursos de forma inmediata. Por otra parte, tampoco había problema en dejarla de lado por un tiempo, hasta que el agotamiento de IPv4 fuera inminente. Así, la mayoría de las empresas pospuso la implementación del nuevo protocolo todo lo que pudo. Se llegó entonces al límite en que ya no es posible implementar IPv6 en toda internet antes del agotamiento de las direcciones IPv4.

La realidad actual es de que las direcciones libres de IPv4 o bien ya se agotaron o bien están extremadamente próximas a agotarse. Sin embargo, en términos generales, todavía no se ha implementado IPv6. **En esta nueva realidad, en este nuevo contexto, los proveedores de acceso necesitan conectar a los usuarios usando IPv6, pero también deben proporcionarles una dirección IPv4, utilizando alguna técnica de uso compartido.** Proveer conectividad IPv4 paliativa es importante para permitir la comunicación con aquellas partes de Internet que todavía no han migrado a IPv6. Para resolver este desafío se desarrollaron nuevas tecnologías. En este trabajo, a estas técnicas de transición se les denomina nuevas.

> Lo importante es comprender que, hoy en día, la expectativa es que la transición de Internet a IPv6 se haga rápidamente.

6.3_
Técnicas de transición tradicionales

Las técnicas aquí descritas son apropiadas para sortear las partes de la red, o de la Internet, que solo soportan IPv4, interconectando las regiones que soportan IPv6. A pesar de que fueron creadas para un contexto de transición bastante diferente del actual, estas técnicas siguen siendo muy útiles en ciertas situaciones. Todos los mecanismos que se abordan en este trabajo se basan en el uso de túneles y cada uno de ellos tiene su propia utilidad y sus propios casos de uso.

6.3.1. Túneles estáticos 6in4

Es posible encapsular paquetes IPv6 directamente dentro de paquetes IPv4, en forma de payload. En este caso, en el campo de protocolo del encabezado IPv4 se especifica el valor 41 (29 en hexadecimal). Este tipo de encapsulamiento se describe en la RFC 4213 y se conoce como 6in4, o IPv6-in-IPv4. Popularmente también se le suele llamar “Protocolo 41”.

En sí, el encapsulamiento es muy simple pero, al encapsular un paquete IPv6 dentro de un paquete IPv4, es necesario considerar algunos temas de mayor complejidad. Por ejemplo, podría no haber espacio suficiente para el paquete y quizás haya que fragmentarlo, o devolver un mensaje de “packet too big” a quien lo originó. También hay que convertir los errores ICMPv4 que ocurren a lo largo del camino en errores ICMPv6.

Los túneles 6in4 se pueden configurar manualmente. Esta configuración básicamente consiste en definir las direcciones IPv4 de origen y destino utilizadas en cada extremo del túnel.

Los túneles IPv6 estáticos se configuran manualmente y son útiles en diferentes situaciones. Por ejemplo, se pueden utilizar para evitar un equipo o enlace que no soporta IPv6 en una determinada red. También se pueden usar para interconectar dos redes IPv6 por medio de la Internet IPv4. La Figura 1 ilustra el uso de un túnel 6in4.

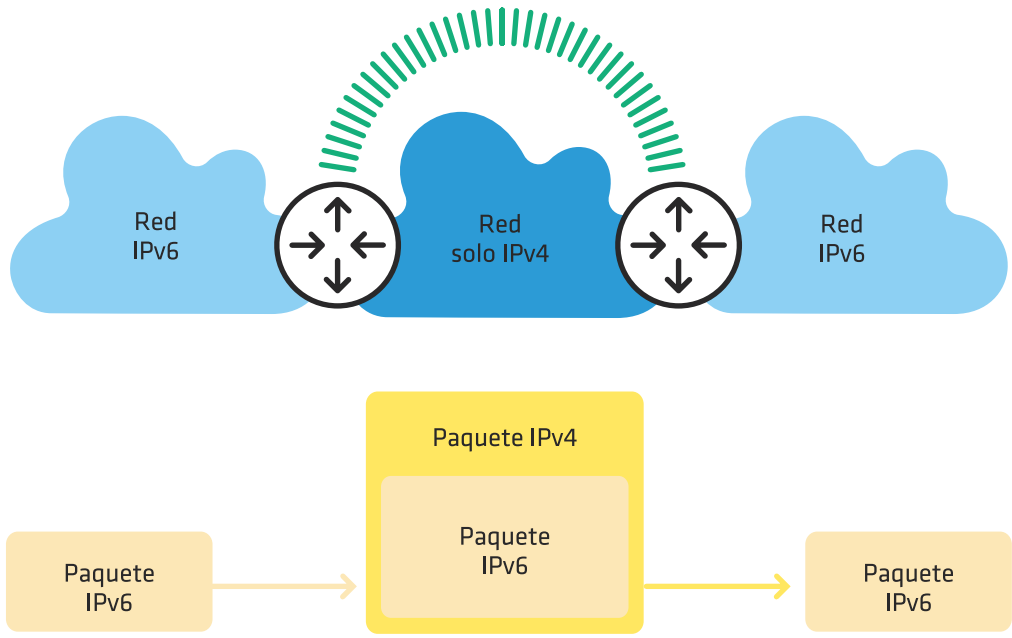

FIGURA 1: TÚNEL 6IN4

Es importante observar que el encapsulamiento 6in4 también se puede utilizar en otras técnicas que involucran túneles automáticos. Vale la pena mencionar que también existe el encapsulamiento equivalente pero inverso –4in6– y que este tipo de encapsulamiento se utiliza en varias de las técnicas nuevas, las cuales se abordarán más adelante.

Prácticamente todos los tipos de routers y sistemas operativos soportan túneles estáticos 6in4.

6.3.2. Túneles estáticos GRE

GRE (Generic Routing Encapsulation) es un tipo de encapsulamiento genérico definido en la RFC 2784, luego actualizada por la RFC 2890. GRE tiene un encabezado propio y puede transportar diferentes tipos de protocolos. También puede ser transportado en varios tipos de protocolos.

Para encapsular paquetes IPv6 en IPv4, primero se agrega el encabezado GRE. Luego se agrega el encabezado IPv4 y en el campo de protocolo se especifica el valor 47 (2F en hexadecimal), indicando que se está transportando IPv4 como payload de GRE.

El túnel GRE también se configura estáticamente, de manera muy semejante a un túnel 6in4. Prácticamente todos los routers y sistemas operativos soportan esta tecnología.

Una ventaja del túnel GRE en relación con el 6in4 es que el primero puede transportar diferentes protocolos simultáneamente, mientras que el segundo solo transporta IPv6. Por ejemplo, con GRE se puede crear un túnel para transportar simultáneamente IPv6 y CLNS, el protocolo usado por el ISIS. La desventaja –bastante obvia– es que el overhead es mayor. La Figura 2 ilustra el uso de un túnel GRE.

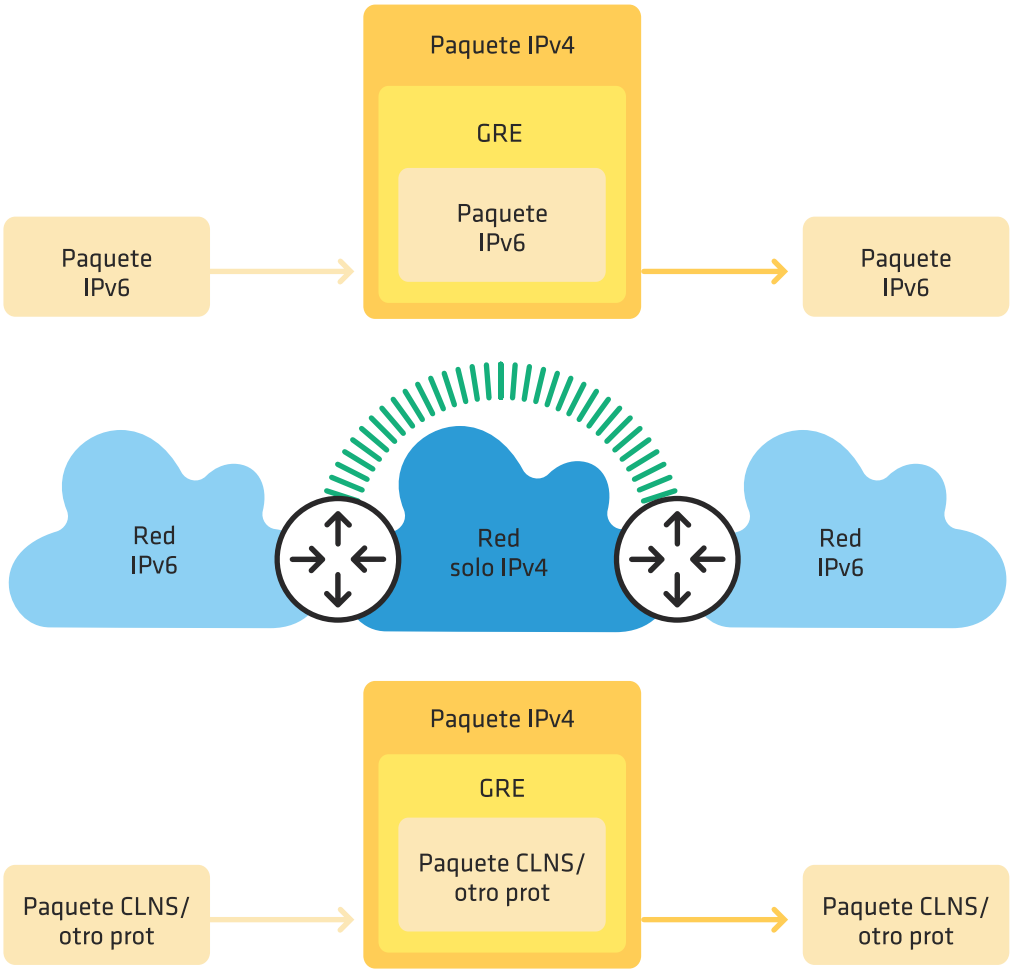

FIGURA 2: TÚNEL GRE

Los casos de uso para túneles estáticos 6in4 y GRE son, por lo tanto, muy semejantes. Escoger entre ambos es simple. Si IPv6 es el único protocolo que se debe transportar en el túnel, el encapsulamiento 6in4 es la mejor opción. Si es necesario transportar otros protocolos, como por ejemplo CLNS, entonces se debe utilizar GRE.

6.3.3. Tunnel Brokers

Los Tunnel Brokers pueden ser considerados como proveedores de acceso IPv6 virtuales. Este mecanismo está documentado en la RFC 3053. Fue creado a comienzos de la década del 2000, época en la que la oferta de conectividad IPv6 nativa era todavía muy pequeña, y su objetivo era ofrecer a todos los interesados una alternativa de conexión estable y persistente, por medio de túneles.

Los Tunnel Brokers funcionan de la siguiente manera. Generalmente hay un sitio web donde el interesado crea una cuenta y solicita el servicio. En algunos casos, el proceso de aprobación es manual; en otros, automático. Una vez aprobada la prestación del servicio de túnel, el proveedor lo configura en un servidor de túneles. Luego se envían las instrucciones para que el usuario pueda configurar su extremo del túnel. El túnel queda establecido cuando el usuario lo configura correctamente en su red y, a partir de ese momento, puede proveer conectividad IPv6 sobre la Internet IPv4. Existen diferentes tecnologías que se pueden utilizar para establecer estos túneles. Por ejemplo, encapsulamiento 6in4, encapsulamiento UDP, el protocolo AYIYA (Anything on Anything), el TSP (Tunnel Setup Protocol, descrito en la RFC 5572). Esta tecnología se ilustra en la figura 3.

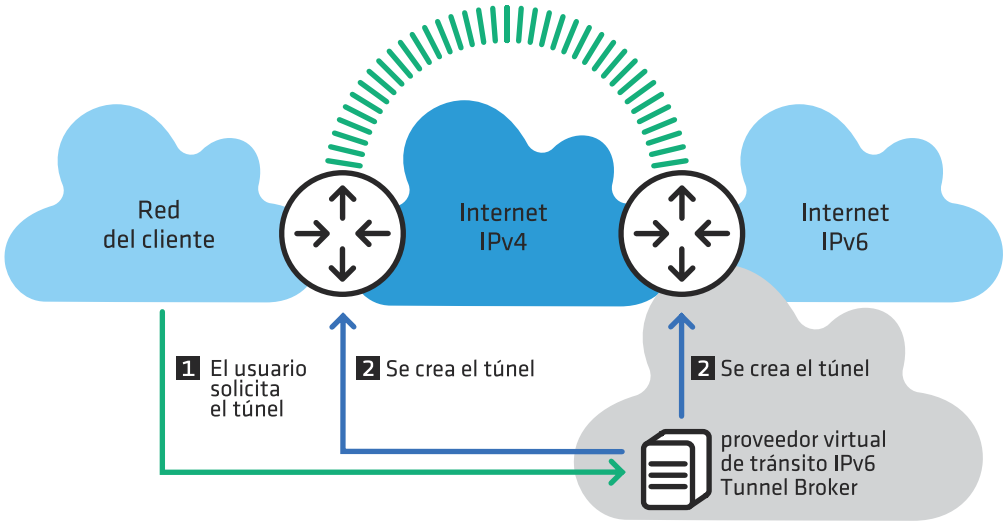

FIGURA 3: TÚNEL BROKER

En el momento de escribir este texto a mediados de 2013, aún existían diferentes opciones de Tunnel Brokers, algunos con el objetivo de ofrecer conectividad a usuarios finales, locales y corporativos. En estos casos, generalmente se puede obtener un prefijo /64 o /48 para ser utilizado en la red. Otros ofrecen túneles sobre los cuales un sistema autónomo puede establecer sesiones BGP y anunciar sus propios prefijos. La mayoría ofrece su servicio de forma gratuita.

Utilizar un Tunnel Broker puede ser una opción para los operadores de red que están comenzando a implementar el protocolo pero que todavía no cuentan con tránsito IPv6 nativo en la localidad donde operan. Se recomienda utilizar este mecanismo solo a modo de prueba, como parte del proceso de transición.

6.3.4. 6PE y 6VPE

Actualmente, el uso de MPLS sobre IPv4 está bastante difundido en las redes. 6PE y 6VPE son dos tecnologías que permiten implementar IPv6 sin alterar el núcleo MPLS de la red. Además de la nueva configuración, solo es necesario actualizar el software de los routers de borde, conocidos como PE (Provider Edge), si es que no soportan esta funcionalidad.

Las técnicas 6PE y 6VPE se describen, respectivamente, en las RFC 4798 y 4659. La comunicación se logra utilizando LSP (Label Switch Paths) a través del núcleo MPLS. Ambas técnicas utilizan MP-BGP (Multiprotocol BGP) sobre IPv4 para intercambiar rutas IPv6. Los routers PE deben ser doble pila. Los routers del núcleo MPLS no se dan cuenta que están transportando paquetes IPv6, dado que solo consideran los encabezados MPLS.

En 6PE solamente se mantiene una tabla de enrutamiento, por lo que la técnica es más adecuada para brindar el servicio de Internet. En 6VPE se pueden mantener diferentes tablas de enrutamiento independientes, separadas lógicamente. Por lo tanto, esta técnica es más apropiada para brindar servicios de VPN (Virtual Private Network). La figura 4 ilustra ambas técnicas.

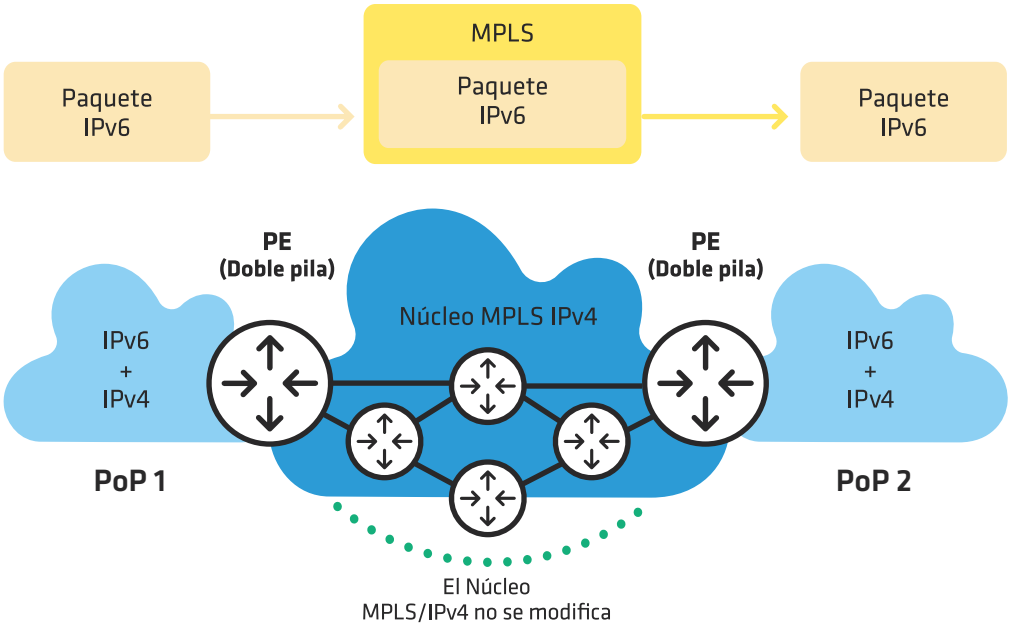

FIGURA 4: 6PE Y 6VPE

Tanto 6PE como 6VPE se recomiendan para todos los operadores de red que utilizan MPLS en su núcleo. De hecho, actualmente no es posible utilizar MPLS sobre una red solo IPv6. En caso de que se opte por utilizar esta tecnología, ésta se debe utilizar sobre IPv4. 6PE y 6VPE son las mejores opciones para implementar IPv6 en estas redes. Son tecnologías maduras, ampliamente utilizadas y soportadas por los principales fabricantes de equipos de red.

6.3.5. Túneles automáticos 6to4

El mecanismo 6to4 se describe en la RFC 3056 y se incluye aquí para que luego sea más fácil comprender la técnica 6rd. También se incluye para que los administradores de red estén al tanto de los problemas de seguridad relacionados con el mismo, aunque actualmente es de poca utilidad práctica.

Esta técnica tiene tres elementos principales: los clientes 6to4, los routers 6to4 y los relays 6to4. Los clientes son las computadoras conectadas a una red que utiliza este tipo de túnel para obtener conectividad IPv6. Se trata de clientes IPv6 convencionales. Un router 6to4 es aquel que en la red del cliente oficia como extremo del túnel y, por lo tanto, debe tener una dirección IPv4 válida. A partir de allí, utilizando el prefijo 2002::/16 más los 32 bits de la dirección IPv4, se forma un prefijo IPv6 /48 para ser utilizado en la red. El otro extremo del túnel lo proveen los relays 6to4, que son routers con conectividad nativa IPv4 e IPv6. Muchas redes ofrecen el servicio de relay 6to4 colaborativamente en Internet, utilizando para la conectividad IPv4 la dirección anycast 192.88.99.1. En la Internet IPv6, estos relays se anuncian como routers para el prefijo 2002::/16.

Los paquetes IPv6 se encapsulan utilizando 6in4. El router encuentra el relay más cercano enviando el paquete a la dirección IPv4 anycast. El relay desencapsula el paquete y lo envía a su destino en la Internet IPv6. El destino enruta la respuesta al relay más próximo, que es el router para 2002::/16. Este encapsula nuevamente el paquete y lo envía al router cuya dirección IPv4 forma parte de la dirección IPv6 de destino. Nótese que los túneles no son necesariamente simétricos. También se puede utilizar direcciones unicast y configurar los routers manualmente para especificar relays 6to4. De todas maneras, desde el punto de vista del usuario es imposible controlar el camino inverso. Para un proveedor de servicios o de contenido que opera en doble pila podría ser ventajoso contar con un relay 6to4, no anunciado públicamente, exclusivamente para responder a consultas provenientes de clientes 6to4 y así garantizar el encapsulamiento del paquete en su origen. Esta técnica se ilustra en la figura 5.

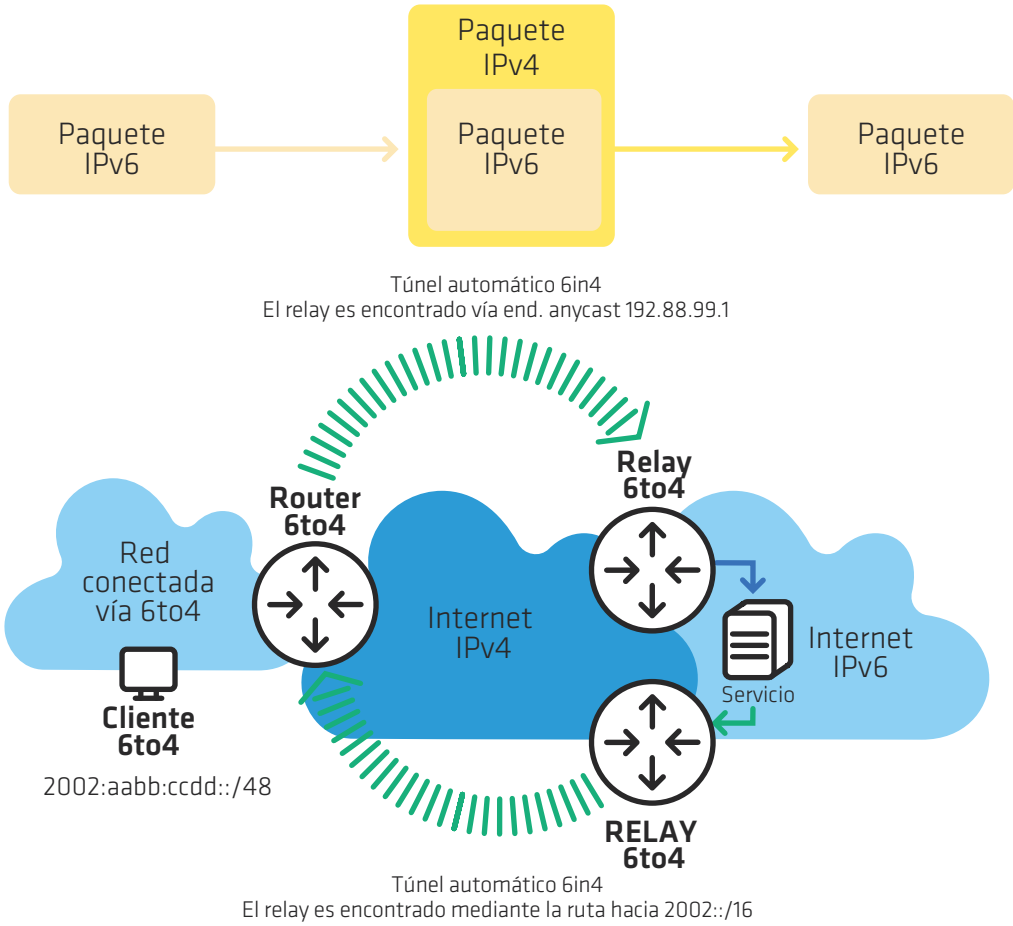

FIGURA 5: 6IN4

6to4 es afectado por diferentes problemas. Los relays son públicos y no hay garantía de que ofrezcan un servicio de calidad. Tampoco de que no sufrirán problemas de seguridad. Varios sistemas operativos soportan túneles automáticos 6to4, en particular, Windows XP, Windows Vista y Windows 7. En estos sistemas, una vez que se obtiene una dirección IPv4 válida, la computadora pasa a actuar como cliente y router 6to4, sin que sea necesaria ninguna intervención por parte del usuario. En las redes empresariales generalmente esto no es deseable porque, inadvertidamente, los túneles podrían saltear ciertas medidas de seguridad tales como firewalls. Además, con estos túneles la conectividad con ciertos sitios y servicios de Internet que ya operan en doble pila puede tener una calidad inferior a la que puede lograrse con IPv4 o IPv6 nativos. Por lo tanto, en estas redes es aconsejable bloquear el

uso del "protocolo 41", evitando así que los usuarios utilicen túneles automáticos. También es aconsejable deshabilitar esta funcionalidad en los sistemas operativos de las computadoras.

6to4 es una técnica de transición que tuvo un papel histórico importante, pero que actualmente tiene poca utilidad. Se recomienda desactivarla y bloquear su utilización en las redes corporativas. En caso que se desee obtener conectividad IPv6 pero que no haya oferta disponible, se debe optar por el uso de Tunnel Brokers en lugar de 6to4. Para las redes de doble pila donde se alojan servicios públicos en Internet, en especial servicios web, puede ser deseable instalar un relay 6to4 a fin de brindar una mejor conectividad a los usuarios de esta tecnología. Esta técnica luego dio origen a 6rd, la cual se describe a continuación.

6.3.7. TÚNELES 6RD

La técnica conocida como 6rd (IPv6 rapid deployment) es una extensión de 6to4 que resuelve los problemas de asimetría y de falta de control sobre los relays utilizados. Así como 6PE y 6VPE permiten que un operador de red implemente IPv6 sin modificar su núcleo basado en MPLS IPv4, 6rd permite utilizar la infraestructura de red de acceso IPv4 sin modificaciones para realizar una implementación rápida de IPv6 hasta el usuario final.

Esta técnica se describe en la RFC 5569. Tiene dos elementos principales: el CPE 6rd y el relay 6rd. El CPE 6rd funciona de manera similar a un router 6to4, pero el prefijo utilizado es el del bloque de direcciones del proveedor de acceso. Normalmente se utiliza un prefijo de 32 bits, aunque también se puede escoger otro largo de prefijo. El relay 6rd se aloja en la red del proveedor y tiene conectividad nativa IPv6 e IPv4. El encapsulamiento es 6in4. Esta técnica se ilustra en la figura 6.

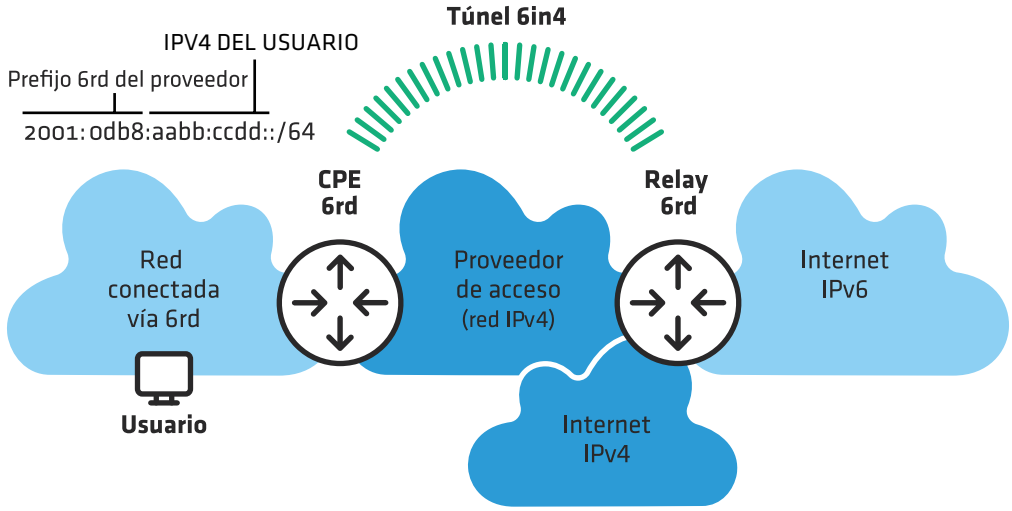

FIGURA 6: TÚNEL AUTOMÁTICO 6RD

Nótese que 6rd no resuelve el problema de la escasez de direcciones IPv4, por lo que esta técnica solo se puede utilizar cuando hay direcciones disponibles. Teóricamente, sería posible utilizarla junto con algún tipo de mecanismo para compartir direcciones IP en el proveedor, aunque esto no es para nada aconsejable ya que complicaría excesivamente la red. Además, 6rd requiere actualizar el software de los CPE o bien cambiarlos.

Su utilización se recomienda para los proveedores que no se van a ver afectados por el agotamiento de direcciones en el corto o mediano plazo, por ejemplo, en aquellos casos donde la base de usuarios crece muy poco. Además, el proveedor debe poder gestionar los CP y realizar una actualización para agregar esta funcionalidad. Por último, no debe ser posible implementar IPv6 nativo en su red en el corto plazo, por ejemplo, debido a la existencia de algún equipo importante que no soporta IPv6 y que no se puede sustituir de inmediato. Si alguna de estas condiciones no se cumple, es probable que exista otra técnica más adecuada.

Por último, es importante destacar que también es posible utilizar 6rd como una técnica provisoria durante la transición. En otras palabras, mientras todavía haya direcciones IPv4 disponibles, se utiliza 6rd. Cuando el problema del agotamiento empieza a afectar al proveedor, se comienza a utilizar otra técnica, probablemente alguna de las que se analizarán más adelante en la sección sobre las nuevas técnicas de transición. En algún momento futuro el proveedor pasa a ofrecer solamente IPv6. En este caso es importante considerar cuidadosamente los costos y las dificultades operativas que implica dividir la transición en diferentes etapas.

6.3.8. Teredo

Teredo es un mecanismo bastante similar a 6to4. Este mecanismo se describe en la RFC 4380 y se incluyó aquí para que los administradores de red sean conscientes de los problemas relacionados con su uso. Actualmente es de poca utilidad práctica.

El prefijo utilizado para los clientes es 2001:0000::/32. Con Teredo se conecta un único cliente, no toda una red. Para el encapsulamiento se utiliza UDP para que los túneles funcionen también en redes con NAT. Además de relays similares a los que se utilizan con 6to4, también existen servidores que ayudan a descubrir el tipo de NAT utilizado por la red del usuario y a iniciar la comunicación.

Los problemas para las redes corporativas son similares a los que se describieron para 6to4. Por ello es conveniente bloquear activamente el uso de esta técnica, lo que se puede lograr deshabilitándola en los sistemas operativos o bien bloqueando en la red el tráfico de salida al puerto UDP 3544.

Los mecanismos más apropiados para su utilización por parte de los proveedores de acceso a Internet en un contexto donde el agotamiento de IPv4 es ya una realidad.

6.4_ Nuevas técnicas de transición

Los mecanismos que aquí se describen son los más **apropiados para su utilización por parte de los proveedores de acceso a Internet en un contexto donde el agotamiento de IPv4 es ya una realidad.** Permiten conectar a los usuarios con IPv6 nativo, ofreciéndoles también una conexión parcial a IPv4 por medio de algún mecanismo que permita compartir estas direcciones en la red del proveedor. Aquí describiremos NAT64 y DNS64, 464XLAT, MAP, DS-Lite y NAT444.

Tal vez la inclusión de NAT444 genere alguna polémica. La técnica consiste en la doble aplicación de NAT44 en el usuario y en el proveedor de acceso. NAT44 no es algo particularmente nuevo. Tampoco se lo puede clasificar estrictamente como una técnica de transición, sino que es apenas una técnica de conservación de direcciones IPv4. Por lo tanto, si el proveedor la aplica de manera aislada, sin ofrecer simultáneamente conectividad IPv6 nativa, puede ser extremadamente perjudicial para Internet. Pero si se aplica de forma simultánea con la implementación de IPv6 nativo hasta el usuario final, puede ser considerada una de las opciones disponibles, con ventajas y desventajas como cualquiera de las otras técnicas.

DNS64 y NAT64 son dos técnicas que se aplican en conjunto y son las únicas que implican entregar solo IPv6 al usuario final: con ellas el usuario final no recibe ninguna dirección IPv4, ni válida ni privada. El acceso a la Internet IPv4 se da por medio de la traducción entre los protocolos.

En las demás técnicas –464XLAT, MAP y DS-Lite– el usuario tiene conectividad IPv6 nativa pero también recibe una dirección IPv4 privada por medio de la cual tiene una conexión parcial a la Internet IPv4 a través de una dirección compartida. En estas técnicas también se utiliza una traducción stateful, similar a lo que ocurre en NAT44. En algunas esto lo realiza el proveedor de acceso, mientras que en otras lo hace el usuario final, pero nunca se hace en ambos lugares, es decir que nunca hay una doble traducción stateful. El IPv4 llega al usuario por medio de un túnel sobre IPv6 o por medio de una doble traducción, de IPv4 a IPv6 y viceversa, dependiendo de la técnica particular utilizada. Cuando hay una doble traducción de IPv4 a IPv6, una de las dos traducciones es siempre stateless.

A continuación, se describen detalladamente estos mecanismos.

6.4.1. NAT444

Esta técnica también se conoce como CGNAT (Carrier Grade NAT). Cualquier técnica que utilice traducción de direcciones en la red del proveedor de acceso puede ser considerada una CGNAT. Consideramos que el nombre NAT444 es más preciso y específico y es por ello que se lo utiliza en este trabajo.

Se trata de la primera de las tecnologías descritas en esta sección que se aplica al problema de conectar un usuario a Internet utilizando IPv6 e IPv4 cuando ya no hay más direcciones IPv4 libres disponibles. Es importante tener en cuenta que no por ello se debe considerar la más importante ni la más recomendada. En muchos casos es justamente lo contrario.

Sin embargo, con fines didácticos, es importante presentarla en primer lugar porque es la que a muchos les parece más familiar. Su componente básico es la traducción NAT44, ampliamente utilizada hoy en día en las redes de los usuarios finales. Luego se podrán discutir algunos problemas comunes a todas las demás técnicas que involucran el uso compartido de direcciones, lo que facilitará la comprensión de las demás. Además, es un hecho que muchos proveedores han escogido esta técnica y que muchos fabricantes la han recomendado. Esto se da, en algunos casos, por desconocimiento de las alternativas y por desconocimiento de los problemas que son exclusivos del uso de la doble traducción stateful.

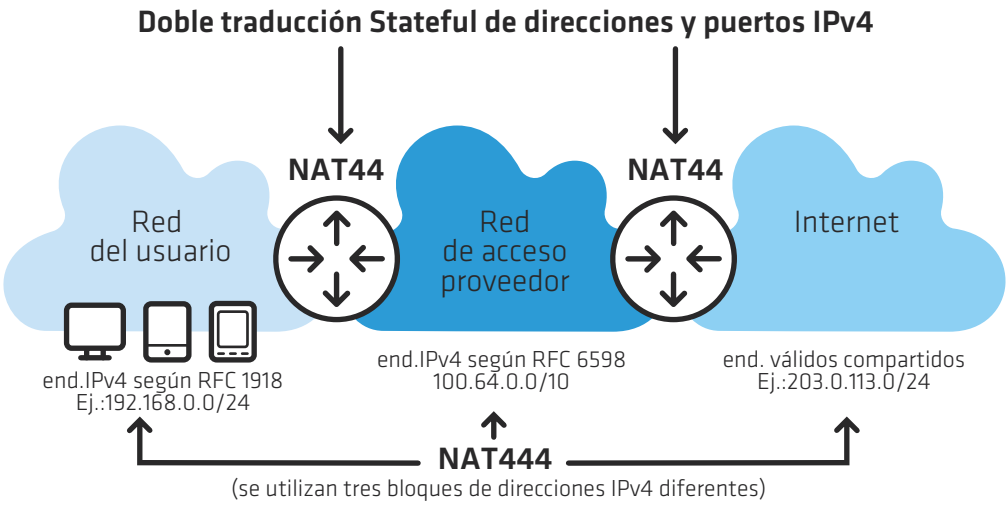

FIGURA 7: NAT444

La figura 7 ilustra cómo funciona el NAT444. Se puede ver que este mecanismo consiste en aplicar dos traducciones de direcciones y puertos de IPv4 a IPv4. Una de ellas se da en la red del usuario, algo muy común desde la década de 1990, mientras que la novedad de esta técnica es que la segunda traducción se da en el proveedor de acceso. Hay tres bloques de direcciones IPv4 involucrados: las direcciones válidas en Internet, las direcciones privadas utilizadas en la red del usuario, y las direcciones privadas utilizadas en la red del proveedor. Nótese que las direcciones utilizadas en la red del usuario son las que se definen en la RFC 1918 y son muy conocidas. La IANA reservó un nuevo bloque para su utilización exclusiva en la red del proveedor con el objetivo de compartir direcciones, el bloque 100.64.0.0/10. Su uso se describe en la RFC 6598.

{

1 IP=65535 TCP

Cada dirección IP puede ser el origen de 65535 conexiones TCP simultáneas, utilizando un puerto de origen diferente para cada una de ellas.

El uso de NAT44 en la red del usuario es una necesidad que surgió en los últimos años para preservar las direcciones IPv4 libres, pero que

rompe la conectividad extremo a extremo en Internet. Esta propiedad de la red, que permite que cualquier dispositivo inicie la comunicación con cualquier otro, es uno de los factores clave de su éxito. Es de gran importancia, principalmente para garantizar la libre creación de nuevas aplicaciones y la innovación. Cuando se usa NAT, los dispositivos que reciben direcciones privadas no pueden ser alcanzados directamente por otros dispositivos en la Internet.

Aunque la traducción rompe la conectividad de extremo a extremo, se han desarrollado algunas soluciones paliativas para sortear esta dificultad en las redes de los usuarios finales. El usuario puede configurar manualmente un mapeo de puertos en el router. El software de la red privada puede utilizar uPNP, un protocolo que permite el mapeo automático. También puede utilizar STUN para identificar la dirección IP externa compartida, o bien puede optar por emplear otras técnicas.

De esta manera, las aplicaciones desarrolladas para correr en redes de usuarios con NAT muchas veces son más complejas y caras, aunque existen técnicas que permiten su funcionamiento. Estas técnicas no son compatibles con una traducción en la red del proveedor de acceso, es decir, el uso de NAT en la red del proveedor rompe de una manera mucho más grave la conectividad extremo a extremo de Internet. Las aplicaciones que dependen de la conectividad extremo a extremo, como por ejemplo el intercambio de archivos P2P, las conferencias de voz y video y, en particular, diferentes tipos de juegos en línea, podrían no funcionar con NAT444. Por las mismas razones ya expuestas, algunas de estas aplicaciones podrían no funcionar tampoco con otros mecanismos que también utilizan NAT en el proveedor de acceso. Esto significa que es importante comprender que una doble traducción stateful representa un desafío mucho mayor que una única traducción.

Otro problema que existe siempre que se comparten direcciones pero que es potencialmente agravado por el doble uso de la técnica es la escasez de puertos. Cada dirección IP puede ser el origen de 65535 conexiones TCP simultáneas, utilizando un puerto de origen diferente para cada una de ellas. Esto es mucho más que suficiente para la mayor parte de las aplicaciones. Si un usuario comparte una dirección IP en su red con 100 computadoras, cada una de ellas potencialmente podrá tener 655 conexiones simultáneas abiertas. Si la misma dirección IP ya fue compartida por el proveedor entre 100 usuarios, este número se reduce a cerca de seis conexiones por computadora. Algunas aplicaciones web llegan a usar, por sí solas, unas treinta conexiones simultáneas. En este escenario, tales aplicaciones dejarían de funcionar.

Es importante tener en cuenta que este problema se puede evitar con algunos cuidados, manteniendo una baja tasa de uso compartido de las direcciones IP en el proveedor, evitando el uso de NAT444 para los usuarios corporativos (que tienden a compartir las direcciones IP entre una mayor cantidad de dispositivos) y ofreciendo simultáneamente IPv6.

El uso compartido de direcciones IPv4 también aumenta la dificultad para identificar un usuario determinado por medio de los registros

de acceso a un servicio o a un contenido en Internet. Cuando existe la necesidad de realizar este tipo de identificación, lo que ocurre habitualmente es que el proveedor del servicio o del contenido proporciona un registro que contiene la dirección IP de origen y el instante en que ocurrió el acceso. Las autoridades competentes solicitan al proveedor de acceso los datos sobre el usuario. Sin embargo, si la dirección IP es compartida por varios usuarios, ya no es posible realizar la identificación de esta manera.

Para que en un escenario de uso compartido de direcciones IP la identificación sea posible, es necesario que tanto el proveedor del servicio y del contenido como el proveedor de acceso conserven un conjunto de datos adicionales: los puertos de origen de las conexiones. Este problema es común tanto para NAT444 como para las demás técnicas de uso compartido de direcciones IPv4. Pero es más grave en el caso del NAT444, ya que este mecanismo no requiere que el usuario tenga conectividad IPv6 y, por lo tanto, no garantiza que el tráfico vaya migrando a IPv6 a medida que los proveedores de servicios y contenidos pasan a utilizar el nuevo protocolo.

El principal problema de la técnica NAT444 es justamente el no exigir la implementación de IPv6. Las demás técnicas necesitan de IPv6 para funcionar. Esto favorece la migración del tráfico en la red del proveedor hacia el nuevo protocolo, a medida que los contenidos, servicios y demás usuarios de Internet pasan a utilizarlo. El uso aislado de NAT444 es, por lo tanto, una pésima idea. Primero, porque cambia drásticamente la manera en la que funciona Internet, rompiendo principios que hasta hoy fueron importantes para su éxito, como ser la simplicidad del núcleo de la red y la conectividad extremo a extremo. Segundo, porque NAT444 es una solución stateful, que tiene elevados costos computacionales y financieros. Si el tráfico no migra gradualmente a IPv6, para el proveedor, el costo de la inversión continuada en la solución de NAT444 tiende a aumentar con el tiempo y, de hecho, tiende a superar el costo de la migración a IPv6. El uso de NAT444 junto con la implementación de IPv6 soluciona este problema, al menos parcialmente.

Se puede considerar un punto positivo el hecho de que el mecanismo utiliza una tecnología bien conocida, aunque a una escala diferente a la mayor parte de las aplicaciones actuales. Prácticamente todos los fabricantes de equipos de red ofrecen soluciones para NAT444. También hay soluciones basadas en software libre. Es la única técnica donde no hay necesidad inmediata de cambiar los CPE, ya sea los de propiedad del proveedor de acceso o los de los propios usuarios. Sin embargo, una consecuencia de no cambiar los equipos de red es que quizás IPv6 no funcione, aunque el proveedor lo ofrezca, lo que daría origen al mismo problema de no migración del tráfico que genera la no implementación de IPv6.

Solamente se recomienda que los proveedores utilicen NAT444 cuando las demás técnicas aquí descritas resulten inviables. En caso de que se opte por su uso, siempre se debe hacer simultáneamente con la implementación nativa de IPv6, nunca de manera aislada.

6.4.2. NAT64 y DNS64

El mecanismo NAT64 se define en la RFC 6146. Es una técnica stateful para traducción de paquetes y puertos IPv6 a IPv4. Permite, simultáneamente, el uso compartido de direcciones IPv4. El DNS64 se define en la RFC 6147 y es una técnica auxiliar de mapeo para nombres de dominio, que se utiliza en conjunto con NAT64.

Con NAT64 y DNS64 es posible que los usuarios reciban únicamente direcciones IPv6 de parte del proveedor y que incluso así puedan acceder a dispositivos IPv4 en Internet. Para el software de la computadora del usuario, parece que todos los sitios y servicios en Internet fueran IPv6. Para el sitio o servicio en Internet, es como que todas las conexiones se originasen en un usuario IPv4 con una IP compartida. El mecanismo se ilustra en la figura 8.

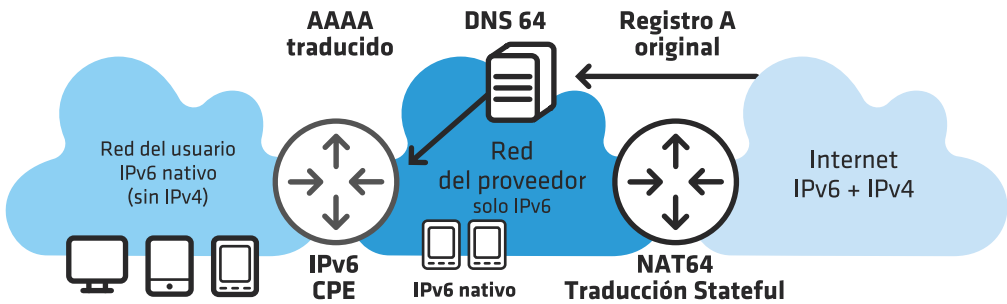

FIGURA 8: NAT64 Y DNS64

Las direcciones IPv4 de Internet se mapean a un prefijo IPv6 predefinido de tamaño /96 en la red del proveedor de acceso. Se puede utilizar cualquier prefijo del propio proveedor, aunque hay un bloque de direcciones reservado exclusivamente para este fin: el 64:ff9b::/96. Este bloque se definió en la RFC 6052. Por ejemplo, la dirección IPv4 203.0.113.1 en Internet se traduciría y mapearía a la dirección IPv6 64:ff9b::203.0.113.1 en la red del proveedor.

Es importante notar que el usuario está conectado de forma nativa a Internet vía IPv6, por lo que puede acceder a los sitios y otros servicios IPv6 directamente, sin necesidad de utilizar ninguna técnica de traducción.

Cuando es necesario acceder a algún recurso IPv4 en Internet, el primer paso que se realiza es una consulta al DNS. DNS64 funciona como un recursivo común, pero en caso de que el nombre consultado no tenga originalmente un registro AAAA, este registro se agrega a la respuesta, utilizando la misma regla de mapeo de direcciones definida para la traducción NAT64. Si la respuesta original llegase solamente con el registro A, no habría más nada que hacer, ya que en la red del usuario solo hay conectividad IPv6. Como la respuesta llega con el registro AAAA "falso" agregado, para la computadora del usuario es como si el servicio o el sitio en realidad ya trabajasen con IPv6 y la conexión se inicia utilizando este protocolo.

Como la dirección del sitio de destino utiliza el prefijo de mapeo NAT64, los paquetes se enrutan hacia el dispositivo responsable por realizar la traducción stateful hacia IPv4. Una vez realizada la traducción, el paquete IPv4 sigue hacia Internet. La dirección de origen es parte de un pool de uso compartido. En la respuesta se realiza la traducción inversa.

Esta técnica también se puede clasificar como CGNAT. Utiliza una traducción stateful en la red del proveedor. Por lo tanto, también existe la necesidad de conservar un registro de los puertos de origen para identificar los accesos realizados a recursos IPv4 en Internet. Esta técnica también aumenta la complejidad del núcleo de la red y tiene un costo computacional alto por ser stateful. Al igual que cualquier NAT, rompe la conectividad extremo a extremo, aunque es importante observar que lo hace con una severidad menor, ya que solo utiliza una traducción y no dos.

La principal desventaja del uso de NAT64 y DNS64 es el hecho de que todavía existen aplicaciones que simplemente no soportan IPv6. En el momento de escribir este texto a mediados de 2013, estas aplicaciones eran pocas pero significativas. Si la aplicación en sí, en la computadora o en el dispositivo del usuario, no puede trabajar con IPv6, de nada sirve la traducción. Este problema puede hacer inviable la implementación inmediata de esta técnica, aunque es probable que deje de existir en un plazo relativamente corto.

Este mecanismo tiene una particularidad interesante que tal vez lo convierta en la mejor opción en términos generales si consideramos la transición a IPv6 como un todo. Es el único donde los usuarios utilizan solamente IPv6. Esto significa que, una vez que los sitios u otros servicios de Internet pasen a usar IPv6 y tengan un registro AAAA en sus nombres de dominio, el tráfico migrará automáticamente a IPv6. No es este el caso cuando se utiliza NAT444, donde IPv6 no es un requerimiento, y tampoco es el caso para los demás mecanismos de transición.

En los demás mecanismos el usuario también recibe direcciones IPv4 privadas. Cuando los sitios y servicios pasan a funcionar con IPv6, las aplicaciones pasan a tener dos opciones: pueden usar IPv6 o IPv4. Si la implementación está bien hecha, IPv6 debería tener preferencia. Esto vale para las técnicas de programación más tradicionales, en las que se intentaba la conexión con una dirección por vez. También vale para técnicas más modernas, como ser la de happy eyeballs, donde las conexiones IPv4 e IPv6 se inician de manera casi simultánea. No obstante, se conocen casos de implementaciones mal realizadas que no garantizan preferencia alguna de IPv6 sobre IPv4.

El uso de NAT64 y DNS64 es de particular interés para los proveedores móviles que trabajan en redes 3G. En este escenario, en algunos casos por limitaciones de licencias, limitaciones de los equipos de red o de los smartphones de los usuarios, solo se consigue conectividad IPv4 o IPv6, no ambas simultáneamente. En este caso, la tecnología es la mejor solución disponible para ofrecer conectividad IPv6 a los usuarios, permitiendo a la vez que accedan a recursos de Internet IPv4.

Además de soporte para IPv6 nativo básico, el uso de NAT64 y DNS64 no requiere ninguna funcionalidad especial en el CPE ni en la terminal del usuario. Se trata de una técnica bastante madura que ofrecen diversos fabricantes de equipos de red. También hay soluciones basadas en software libre. Esta técnica se recomienda principalmente para los proveedores de Internet móvil, pero también se puede utilizar en otros casos. Si la falta de soporte de algunas aplicaciones hace que su implementación sea inviable, se recomienda considerar la técnica 464XLAT, la cual se presenta a continuación.

6.4.3. 464XLAT

La técnica 464XLAT está documentada en la RFC 6877 y, a los fines prácticos, se puede considerar como un complemento del mecanismo NAT64 descrito en el punto anterior. Más específicamente, esta técnica consiste en una combinación del uso de traducción stateful entre IPv6 e IPv4 en la red del proveedor, descrita en la RFC 6146 y en la sección anterior, con otro tipo de traducción entre IPv6 e IPv4, stateless, denominada SIIT (Stateless IP/ICMP Translation Algorithm), descrita en la RFC 6145.

Lo que motivó su desarrollo fue justamente la limitación que se presenta con NAT64 y DNS64 en relación con las aplicaciones que no soportan IPv6. Agregando una segunda traducción es posible brindar a los usuarios una dirección IPv4 privada, de forma que funcionen incluso las aplicaciones que no soportan IPv6.

Nótese que para implementar esta tecnología en una red que ya usa NAT64 basta agregar la traducción en la red o en el dispositivo del usuario, sin necesidad de realizar modificaciones en la red del proveedor.

En esta técnica, ilustrada en la figura 9, la traducción stateful en el proveedor recibe el nombre de PLAT (Provider-side Translator). En la red del usuario, la traducción stateless se conoce como CLAT (Customer-side Translator).

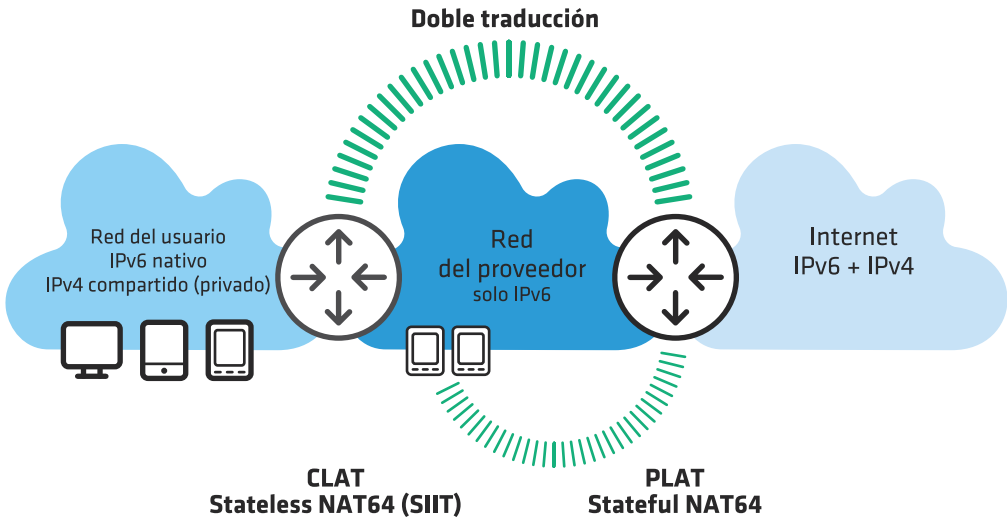

FIGURA 9: 464XLAT

En 464XLAT no se utiliza DNS64. El CLAT aprende el prefijo utilizado por el PLAT por medio de técnicas heurísticas o mediante otro tipo de configuración. Preferentemente, se debería utilizar un prefijo IPv6 /64 específico para el mapeo 1:1 que se realiza para las direcciones IPv4 privadas en la red del usuario.

Es conveniente recordar que la rede del usuario tiene conectividad IPv6 nativa. De esta forma, el acceso a la Internet IPv6 se realiza de forma directa, sin pasar por ninguna traducción, ya sea en el CLAT o en el PLAT.

Nótese que la doble traducción funciona de manera similar a un túnel para el IPv4. No se trata de dos traducciones stateful, como en el caso de NAT444, sino que una de las traducciones es stateless y la otra stateful. Una traducción stateless 1:1 no quiebra, por sí sola, la conectividad extremo a extremo, de manera tal que 464XLAT se puede considerar menos perjudicial para la Internet que NAT444. No obstante, al ser también un tipo de CGNAT, todas las consideraciones realizadas para NAT64 son también válidas para 464XLAT.

Este mecanismo es relativamente reciente, aunque en realidad no involucra nuevas tecnologías. Se trata de una aplicación diferente, un reacomodo de tecnologías preexistentes. Como se vio anteriormente, existe una gran cantidad de implementaciones de PLAT (NAT64). En el momento de escribir este texto a mediados de 2013, el CLAT ya se había implementado y testeado satisfactoriamente en Android y Linux. Es posible que en breve esté disponible en los smartphones como una funcionalidad estándar, especialmente si los operadores y proveedores de Internet móvil así lo desean.

El uso de 464XLAT se recomienda para los proveedores de Internet móvil, en especial para aquellos que pretenden implementar IPv6 a corto plazo, siempre que NAT64 y DNS64 no sean viables debido a problemas con las aplicaciones. O incluso como alternativa para que la utilicen solamente los usuarios para los cuales el no funcionamiento de ciertas aplicaciones realmente significa un problema. La técnica también se puede utilizar en otros casos.

6.4.4. DS-Lite

La técnica DS-Lite (Dual Stack Lite) se describe en la RFC 6233. Resuelve el problema de forma similar a 464XLAT, aunque utiliza un túnel que encapsula IPv4 en IPv6 y no una doble traducción entre protocolos. Por lo tanto, el usuario se conecta vía IPv6 en forma nativa y también recibe una dirección IPv4 privada.

DS-Lite también es una clase de CGNAT, es decir, depende de NAT44 stateful en el proveedor de acceso. En esta técnica, el equipo responsable por el CGNAT recibe el nombre de AFTR (Address Family Transition Router). En la red del usuario, el CPE recibe el nombre de B4 (Basic Bridge BroadBand) y actúa como un bridge para el IPv4, en la terminación del túnel. La figura 10 ilustra esta técnica.

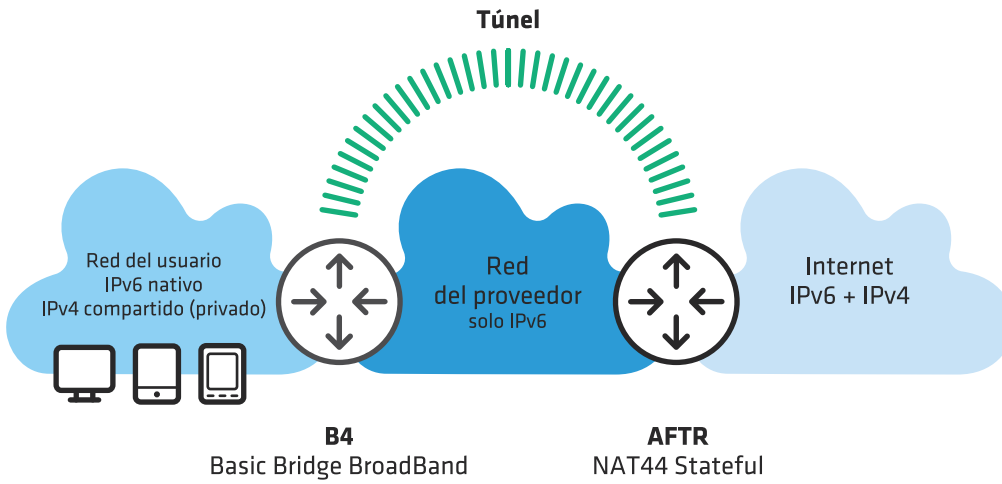

FIGURA 10: DS-LITE

Vale la pena observar que con el túnel y el CPE en la función de B4 (bridge), el AFTR tiene un puerto conectado directamente en la red del usuario. El mismo ofrece la función de NAT directamente para los dispositivos. En otras palabras, no hay una doble traducción como ocurre con NAT444, sino solamente una única traducción stateful en la red del proveedor.

Por ser un tipo de CGNAT, se aplican las mismas consideraciones que hicimos para NAT64 y 464XLAT. Para posibilitar la identificación de los accesos vía IPv4 es necesario llevar un registro de los puertos de origen de los usuarios. La técnica también rompe la conectividad extremo a extremo y, por ser stateful, tiene un costo computacional alto para el proveedor.

Para implementar DS-Lite se requiere un CPE que soporte esta técnica. Existen diversos modelos disponibles en el mercado con esta funcionalidad. Diferentes fabricantes ofrecen soluciones de equipos para la función del AFTR; también existen soluciones basadas en software libre. Por todo lo anterior, DS-Lite es una técnica madura que se puede implementar en una red en el corto plazo.

Su uso se recomienda para los proveedores de acceso a Internet en general, para quienes ya están sufriendo el efecto del agotamiento de las direcciones IPv4 y necesitan realizar una implementación de IPv6 en el corto plazo. La técnica MAP, que se presenta a continuación, es una solución similar, pero con algunas ventajas técnicas. Antes de optar directamente por DS-Lite, se recomienda analizar la viabilidad de su utilización.

6.4.5. MAP

Desde el punto de vista del usuario, la técnica MAP (Mapping of Addressing and Port) es muy similar a DS-Lite o 464XLAT. El usuario está conectado nativamente vía IPv6 y vía IPv4 utilizando una dirección IPv4

privada. Existen dos versiones de esta técnica: MAP-T (Traducción), que utiliza una doble traducción stateless entre IPv4 e IPv6 de forma similar a lo que hace 464XLAT, y MAP-E (Encapsulamiento), que utiliza un túnel para encapsular IPv4 en IPv6 de forma similar a lo que se hace en DS-Lite. En MAP, el router responsable por compartir las direcciones IP en el proveedor recibe el nombre de MAP Border Relay. En la red del usuario, el CPE recibe el nombre de MAP CE. En las figuras 11 y 12 se ilustran ambas versiones, MAP-T y MAP-E.

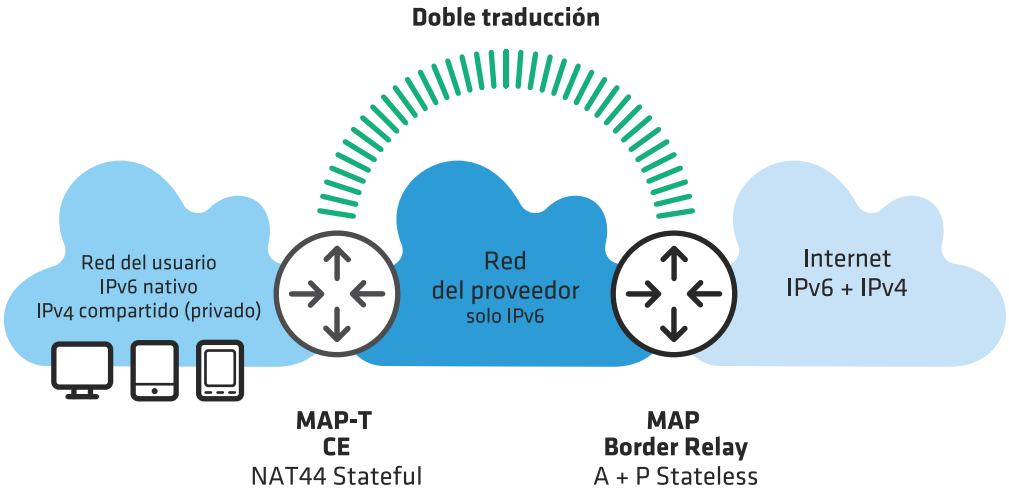

FIGURA 11: MAP-T

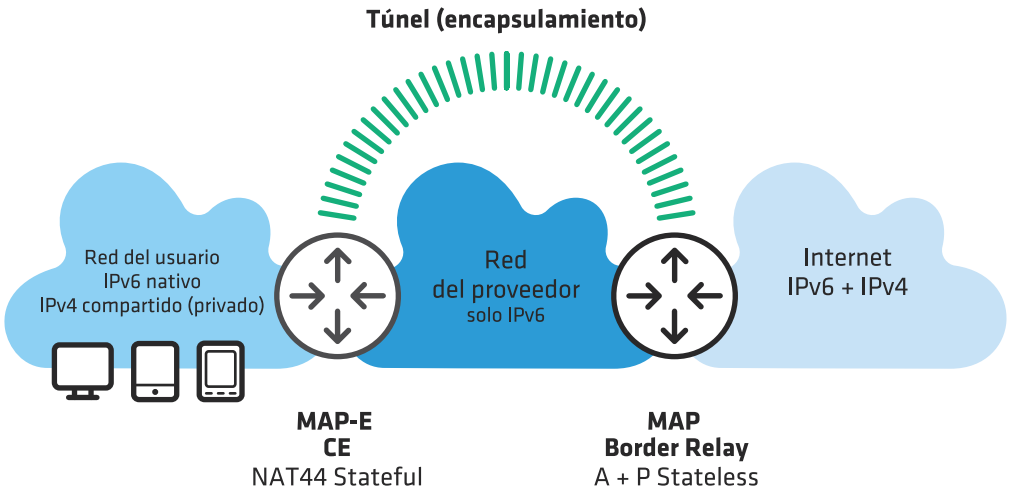

FIGURA 12: MAP-E

¡La gran diferencia entre MAP y las demás técnicas es que MAP no es un CGNAT! MAP no utiliza NAT en la red del proveedor de acceso. El uso compartido de las direcciones IPv4 se realiza por medio de la técnica A+P (Address plus Port), descrita en la RFC 6346.

A+P es una forma de compartir direcciones IPv4 de manera stateless. Una misma dirección válida se atribuye a varios usuarios diferentes, pero

El mapeo de A+P es realizado por un algoritmo. Es mucho más liviano que NAT44. Para el proveedor, es potencialmente más barato y más escalable que cualquier otra de las técnicas aquí presentadas.

cada usuario solo puede utilizar un rango restringido de puertos. Con MAP, el CPE recibe una dirección IPv4 válida junto con información sobre cuál rango de puertos de origen puede utilizar. El EPC es responsable por implementar un NAT44 stateful, entregando direcciones privadas a los dispositivos. Al hacer la traducción, este NAT44 debe respetar la restricción de puertos establecida por el A+P. Esto es totalmente transparente para las aplicaciones en los dispositivos de la red del usuario, que no tienen por qué conocer la restricción del rango de puertos.

Es importante destacar que el mapeo de A+P es realizado por un algoritmo. Esto significa que, desde el punto de vista computacional, es mucho más liviano que NAT44. Para el proveedor, es potencialmente más barato y más escalable que cualquier otra de las técnicas aquí presentadas.

Otro punto clave a destacar es que el CPE recibe una dirección IPv4 válida. Una dirección IPv4 con restricción de rango de puertos utilizables, pero válida. Esto significa que las técnicas que actualmente se utilizan para evitar la rotura de la conectividad extremo a extremo que ocasiona el uso de NAT44 en la red del usuario también funcionarían con MAP, por lo que sería posible utilizar mapeo manual de puertos, o mapeo automático vía uPnp o incluso STUN.

Por todo lo anterior, MAP es el mecanismo de transición que potencialmente implica menos problemas operativos, tanto para el proveedor de acceso como para los usuarios. En el momento de escribir este texto a mediados de 2013, MAP todavía no se había estandarizado en una RFC. A pesar de ello, el proceso estaba bastante adelantado en el Grupo de Trabajo Softwires del IETF y ya existían algunas implementaciones que interoperaban correctamente, tanto de fabricantes de equipos de enrutamiento como disponibles en forma de software libre. Es bastante probable que, antes de que se produzca el agotamiento de las direcciones IPv4 en las regiones de ARIN y LACNIC, el MAP ya haya sido estandarizado y esté maduro y disponible para que los proveedores de acceso lo puedan usar con seguridad.

Se recomienda la utilización de MAP para los proveedores de acceso en general, a menos de que ocurra algo inesperado y que su implementación no parezca viable.

6.4.6. Consideraciones sobre las nuevas técnicas de transición

Es importante considerar que, para los proveedores que todavía no están sufriendo los efectos del agotamiento de IPv4, la doble pila con IPv4 e IPv6 nativos continúa siendo una buena opción para la transición. Incluso en un escenario de agotamiento de direcciones IPv4 a nivel mundial, algunos proveedores de acceso podrían no verse afectados por tener muy poco crecimiento de su base de usuarios. En otros casos, esto puede ser válido solo para algunos tipos de servicios. Por ejemplo, un proveedor podría tener una base de usuarios corporativos que crece muy lentamente y otra base de usuarios residenciales con un crecimiento muy rápido. En este caso, se podría implementar doble pila, con IPv6 e IPv4 nativos, para la base de usuarios corporativos y escoger una de las técnicas aquí presentadas para los usuarios residenciales.

Sin embargo, conviene también considerar que el uso de MAP, DS-Lite, 464XLAT o NAT64 tiene la ventaja de que toda la red de acceso del proveedor pasa a ser solamente IPv6. Esto puede ayudar a reducir la carga en los routers, la cantidad de problemas y los costos operativos. Es también un paso hacia la desactivación de IPv4 en las redes. De esta forma, se puede estudiar la posibilidad de implementar estas técnicas incluso en una situación en la que el uso compartido de IPv4 no es todavía absolutamente necesario.

La elección del mecanismo específico a utilizar es tarea de cada operador de red. Esperamos que las breves descripciones presentadas en este capítulo sean de ayuda en esta tarea. En términos generales, conviene recordar que Internet está migrando hacia IPv6, por lo que se prefieren las tecnologías que implican brindar conectividad nativa IPv6 a los usuarios. Se deben evitar las técnicas que solo prolongan la vida útil de IPv4, sin forzar la migración hacia IPv6. Se deben preferir los mecanismos que menos interfieran con los principios de funcionamiento de Internet. Se deben preferir los mecanismos más escalables. Además, es necesario tener en cuenta la madurez y la adecuación a las condiciones específicas de la red del proveedor.

6.5_ Referencias

Nordmark, E.; Gilligan, R. RFC 4213: Basic Transition Mechanisms for IPv6 Hosts and Routers. 2005. IETF.
Consultado en: http://tools.ietf.org/html/rfc4213

Farinacci, D.; Hanks, S.; Meyer, D.; Traina, P. RFC 2890: Generic Routing Encapsulation (GRE). 2000. IETF.
Consultado en: http://tools.ietf.org/html/rfc2784

Dommety, G.. RFC 2890: Key and Sequence Number Extensions to GRE. 2000. IETF.
Consultado en: http://tools.ietf.org/html/rfc2890

Durand, A.; Guardini, I.; Lento, D. RFC 3053: IPv6 Tunnel Broker. 2001. IETF.
Consultado en: http://tools.ietf.org/html/rfc3053

Massar, J. AYIYA: Anything In Anything - draft-massar-v6ops-ayiya-02. 2004. IETF.
Consultado en: http://tools.ietf.org/html/draft-massar-v6ops-ayiya-02

Blanchet, M.; Parent, F. RFC 5572: IPv6 Tunnel Broker with the Tunnel Setup Protocol (TSP). 2010. IETF.
Consultado en: http://tools.ietf.org/html/rfc5572

De Clercq, J.; Ooms, D.; Prevost, S.; Le Faucheur, F. RFC 4798: Connecting IPv6 Islands over IPv4 MPLS Using IPv6 Provider Edge Routers (6PE). 2007. IETF.
Consultado en: http://tools.ietf.org/html/rfc4798

De Clercq, J.; Ooms, D.; Carugi, M.; Le Faucheur, F. RFC 4659: BGP-MPLS IP Virtual Private Network (VPN) Extension for IPv6 VPN. 2006. IETF.
Consultado en: http://tools.ietf.org/html/rfc4659

Carpenter, B.; Moore, K. RFC 3056: Connection of IPv6 Domains via IPv4 Clouds. 2001. IETF. Consultado en: http://tools.ietf.org/html/rfc3056

Despres, R. RFC 5569: IPv6 Rapid Deployment on IPv4 Infrastructures (6rd). 2010. IETF.
Consultado en: http://tools.ietf.org/html/rfc5569

Huitema, C. RFC 4380: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). 2006. IETF. Consultado en: http://tools.ietf.org/html/rfc4380

Rekhter, Y.; Moskowitz, B.; Karrenberg, D.; Groot, G. J. de; Lear, E.. RFC 1918: Address Allocation for private Internet. 1996. IETF.
Consultado en: http://tools.ietf.org/html/rfc1918

Weil, J.; Kuarsingh, V.; Donley, C.; Liljenstolpe, C.; Azinger, M. RFC 6598: IANA-Reserved IPv4 Prefix for Shared Address Space. 2012. IETF. Consultado en: http://tools.ietf.org/html/rfc6598

Bagnulo, M.; Matthews, P.; van Beijnum, I. RFC 6146: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers. 2011. IETF.
Consultado en: http://tools.ietf.org/html/rfc6146

Bagnulo, M.; Sullivan, A.; Matthews, P.; van Beijnum, I. RFC 6147: DNS64: DNS Extensions for Network Address Translation from IPv6 Clients to IPv4 Servers. 2011. IETF.
Consultado en: http://tools.ietf.org/html/rfc6147

Bao, C.; Huitema, C.; Bagnulo, M.; Boucadair, M.; Li, X. RFC 6052: IPv6 Addressing of IPv4/IPv6 Translators. 2010. IETF.
Consultado en: http://tools.ietf.org/html/rfc6052

Mawatari, M.; Kawashima, M.; Byrne, C. RFC 6877: 464XLAT: Combination of Stateful and Stateless Translation. 2013. IETF. Consultado en: http://tools.ietf.org/html/rfc6877

Li, X.; Bao, C.; Baker, F. RFC 6145: IP/ICMP Translation Algorithm. 2011. IETF.
Consultado en: http://tools.ietf.org/html/rfc6145

Durand, A.; Droms, R.; Woodyatt, J.; Lee, Y. RFC 6333: Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion. 2011. IETF.
Consultado en: http://tools.ietf.org/html/rfc6333

Troan, O.; Dec, W.; Li, X.; Bao, C. Matsushima, S.; Murakami, T.; Taylor, T. Mapping of Address an Port with Encapsulation (MAP) - draft-ietf-softwire-map-07. 2013. IETF.
Consultado en: http://tools.ietf.org/html/draft-ietf-softwire-map-07

Li, X.; Bao, C.; Dec, W.; Troan, O.; Matsushima, S.; Murakami, T. Mapping of Address and Port using Translation (MAP-T) - draft-ietf-softwire-map-t-03. 2013. IETF.
Consultado en: http://tools.ietf.org/html/draft-ietf-softwire-map-t-03

Sun, Q.; Chen, M.; Chen, G.; Tsou, T.; Perreault, S. Mapping of Address and Port (MAP) - Deployment Considerations - draft-ietf-softwire-map-deployment-02. 2013. IETF.
Consultado en: http://tools.ietf.org/html/draft-ietf-softwire-map-deployment-02

Bush, R. RFC 6346: The Address plus Port (A+P) Approach to the IPv4 Address Shortage. 2011. IETF.
Consultado en: http://tools.ietf.org/html/rfc6346

.7

Servicios y Firewalls

7.1_ Firewalls

7.1.1. Aspectos generales de las redes dual-stack

En una red dual-stack es necesario que la Política de Seguridad sea consistente y que un atacante no pueda lograr una ventaja al elegir un protocolo o el otro. Hay algunos aspectos de las redes dual-stack que pueden llevar a que lograr una Política de Seguridad consistente sea un desafío:

- Algunas aplicaciones y protocolos de capas superiores que son nuevos y específicos de IPv6 o que cambiaron significativamente en una forma que afecta a la seguridad.

- Diferencias en el protocolo y en el formato del encabezado IP.

- Tráfico asociado a mecanismos de transición.

7.1.1.1. Aplicaciones y protocolos de capas superiores

El procesamiento de protocolos de capas superiores es prácticamente igual en IPv6 y en IPv4. La mayoría de los números de protocolo usados en IPv4 se mantienen en IPv6. Por ejemplo, TCP y UDP. Sin embargo, ICMPv6 tiene un nuevo número de protocolo, distinto al número de protocolo correspondiente a ICMP.

En IPv6 los firewalls deberían poder filtrar cada uno de los mensajes ICMP basándose en los valores de Tipo (Type) y Código (Code) para permitir la máxima granularidad con respecto a estos mensajes. Se debería evitar la práctica de descartar automáticamente todos los mensajes ICMP ya que debido a algunas de las nuevas características de IPv6, algunos mensajes ICMP son necesarios y no pueden ser descartados (Por ejemplo, Path MTU Discovery)[1].

7.1.1.2. Diferencias en el protocolo

Las optimizaciones de performance logradas por la nueva versión del protocolo IP que aplican principalmente a los routers intermedios, no se extienden a los firewalls intermedios ya que estos siempre necesitan analizar el encabezado completo y la información de capas superiores para poder aplicar una política de seguridad robusta. Esto lleva a que la tarea del firewall sea más compleja, pero no afecta la tarea del operador de la red.

7.1.1.3. Túneles

El uso de túneles lleva a que en la red haya presencia de paquetes IPv4 encapsulados dentro de paquetes IPv6 o a la inversa, paquetes IPv6 encapsulados dentro de paquetes IPv4. Si se van a utilizar túneles, es necesario permitir en los firewalls el tráfico asociado a los protocolos correspondientes (Por ejemplo GRE, 6in4, etc.).

Sin embargo, es importante tener en cuenta que el hecho de permitir túneles en la red puede llevar a que se pierda el control sobre el tráfico

que viaja encapsulado. El uso de túneles puede ser explotado por un atacante, cuando no se realizan chequeos del paquete interno.

7.1.2. Implementaciones de firewall

A continuación se describen algunas implementaciones de firewall para distintas plataformas. Se mencionan estrictamente características relacionadas con el soporte de IPv6 en estas implementaciones pero no se comentan características generales de las mismas, sintaxis de los comandos ni otros detalles que no sean específicos del filtrado de paquetes IPv6.

7.1.2.1. Ip6fw

El firewall ipfw, el cual fue desarrollado originalmente para BSDI, y reescrito completamente para FreeBSD, fue exportado al proyecto de IPv6 KAME bajo el nombre ip6fw. Ip6fw fue integrado completamente a FreeBSD, pero a pesar de esto no soportaba inspección de paquetes con estado. Más tarde, se agregó soporte de IPv6 a ipfw quedando ip6fw en estado obsoleto. Para utilizar ipfw con soporte de IPv6 es necesario que la variable del kernel IPV6FIREWALL tenga el valor "enabled".

Además de las acciones disponibles en ipfw, ip6fw tiene disponible la acción "unreach6 <code>" la cual implica que los paquetes que cumplan las condiciones establecidas por la regla serán descartados y se enviará un paquete "unreachable" de ICMPv6 con el código indicado.

Por otro lado, en cuanto a las opciones, ip6fw tiene algunas opciones específicas de IPv6:

- "protocol ipv6" (matchean todos los paquetes IPv6)
- "protocol ipv6-icmp" (matchean solo los paquetes ICMPv6)
- "addr me6" (matchea cualquier dirección IPv6 configurada en una interfaz del sistema)
- "ip6-addr <ip> <mask>" (utilizado para especificar un host o una subred IPv6)
- "ext6hdr <header>" (matchean paquetes IPv6 que contengan el encabezado de extensión especificado).
- "flow-id <labels>" (matchean paquetes IPv6 que contengan cualquiera de las etiquetas de flujo especificadas).
- "icmp6types <types>" (matchean los paquetes ICMPv6 cuyo tipo esté en la lista <types>).
- "src-ip6 <ipv6 address>" o "dst-ip6 <ipv6 address>" (matchean paquetes IPv6 cuya dirección de origen o de destino (según corresponda) sea una de las direcciones IPv6 especificadas).

7.1.2.2. IP6Tables

La funcionalidad de firewall en equipos Linux está dada por el comando iptables, el cual permite configurar, mantener e inspeccionar las tablas de reglas de filtrado de paquetes en el kernel de Linux. El comando análogo para IPv6 es ip6tables. La mayoría de las directivas para ip6tables son idénticas a aquellas usadas por iptables.

Ip6tables soporta filtrado con estado utilizando las opciones "match". Se deberá incluir la opción "-m state" y la opción "--state <estados>" para especificar los estados que se desea filtrar. Los estados posibles son: INVALID (el paquete no pudo ser identificado), ESTABLISHED (el paquete está asociado con una conexión que ya ha visto paquetes en ambas direcciones), NEW (el paquete ha iniciado una nueva conexión o está asociado con una conexión que no ha visto paquetes en las dos direcciones), RELATED (el paquete está iniciando una nueva conexión, pero está asociado con una conexión existente) y UNTRACKED (el paquete no está siendo rastreado, lo cual ocurre si se utiliza el objetivo NOTRACK en la tabla).

Nota: Es importante tener en cuenta a la hora de elaborar reglas de filtrado de ICMP con ip6tables que la opción -p de ip6tables acepta el protocolo "icmp", sin embargo, la opción "-p icmp" no tiene ningún efecto en IPv6. Lo correcto es utilizar la opción "-p ipv6-icmp" o "-p icmpv6".

Es importante tener en cuenta a la hora de elaborar reglas de filtrado de ICMP con ip6tables que la opción -p de ip6tables acepta el protocolo "icmp", sin embargo, la opción "-p icmp" no tiene ningún efecto en IPv6. Lo correcto es utilizar la opción "-p ipv6-icmp" o "-p icmpv6".

7.1.2.3. Listas de Control de Acceso (ACLs) de Cisco para filtrado de tráfico IPv6

Versiones de IOS superiores a 12.2[2] soportan filtrado de paquetes IPv6. Inicialmente solo existía soporte IPv6 para las ACLs estándar, las cuales solo permiten filtrado basado en direcciones IPv6 de origen/destino. Estas ACLs se pueden aplicar al tráfico entrante o saliente de una determinada interfaz. Más tarde se extendió el soporte de IPv6 en las ACLs para incluir filtrado basado en encabezados de extensión, en campos de los encabezados de extensión y filtrado de capas superiores. Ésta es una funcionalidad similar a las ACLs Extendidas de IPv4.

Es importante tener en cuenta que al final de cada ACL IPv6 existen ciertas reglas implícitas para permitir Neighbor Discovery de ICMPv6.

Si NO hay ninguna regla "deny", implícitamente se habilitan los paquetes de Neighbor Discovery:

```
permit icmp any any nd-na
permit icmp any any nd-ns
deny ipv6 any any
```

Si existe alguna regla "deny", es necesario permitir explícitamente los paquetes de Neighbor Discovery.

Para simular el comportamiento de un firewall con estado se pueden utilizar ACLs reflexivas. Las ACLs reflexivas se crean dinámicamente cuando algún paquete matchea una entrada "permit" que incluye la

palabra clave "reflect". La regla implícita "deny any any" no aplica al final de una ACL reflexiva.

7.1.2.4. Firewall de Cisco IOS para IPv6

El firewall de CIsco IOS para redes IPv6 coexiste con el firewall de Cisco IOS para redes IPv4 y es soportado en todos los routers dual-stack.

A continuación se detallan las características del firewall de Cisco IOS para IPv6:

- Inspección de paquetes fragmentados -El procesamiento de fragmentos es disparado por el encabezado de fragmentación. El reensamblador virtual de fragmentos (VFR por su nombre en inglés "Virtual Fragment Reassembler") examina fragmentos fuera de secuencia y ordena los paquetes, examina la cantidad de fragmentos de una única IP dado un identificador único y realiza un reensamblaje virtual para pasar los paquetes a los protocolos de capas superiores.

- Mitigación de ataques de denegación de servicio (DoS) en IPv6 - Se han implementado mecanismos de mitigación similares a la implementación para IPv4, incluyendo conexiones incompletas SYN (SYN half-open connections).

- Inspección de paquetes encapsulados -Los paquetes IPv6 encapsulados, correspondientes a túneles terminados en un firewall Cisco IOS, pueden ser inspeccionados por el firewall de Cisco IOS para IPv6.

- Inspección de paquetes con estado - Es posible inspeccionar paquetes correspondientes a sesiones TCP, UDP, ICMPv6 y FTP.

- Inspección con estado de paquetes originados en la red IPv4 y terminados en un entorno IPv6 - Esta característica utiliza servicios de traducción IPv4 a IPv6.

- Interpretación o reconocimiento de la mayoría de la información de encabezados de extensión IPv6 - Esta característica provee información de encabezados de extensión IPv6 incluyendo el encabezado de ruteo, el encabezado de opciones hop-by-hop y el encabezado de fragmentación.

- Mapeo de puerto a aplicación (PAM, Port-to-application mapping)

Una restricción (en 2014) del firewall de Cisco IOS para IPv6 es que no soporta el Sistema de Detección de Intrusos (IDS).

7.1.2.5. Soporte de IPv6 en equipos Juniper

IPv6 es soportado desde la versión 6.2 de Screen OS, en la cual por defecto, IPv6 no está habilitado. Para habilitarlo es necesario asignar el valor "yes" a la variable "ipv6" como se muestra a continuación:

```
set envar ipv6=yes
save
reset save-config yes
```

En JunOS, el soporte de IPv6 se incluye desde la versión 10.4.

Firewall Filter de Juniper

La funcionalidad de Firewall Filter es la implementación de firewall sin estado de Juniper. Para configurar filtros para IPv6 se debe utilizar la familia inet6 ("family inet6").

Las condiciones de matcheo que se pueden configurar para la familia inet6 son prácticamente las mismas a las condiciones disponibles para la familia inet (address, destination-address, destination-port, icmp-code, icmp-type, interface, etc.).

Una diferencia que cabe mencionar es que dentro de la familia inet6 no existe la condición "protocol" sino que debe utilizarse la condición "next-header", cuyo valor puede ser un valor numérico o bien alguno de los siguientes textos sinónimos: ah, dstops, egp, esp, fragment, gre, hop-by-hop, icmp, icmp6, icmpv6, igmp, ipip, ipv6, no-next-header, ospf, pim, routing, rsvp, sctp, tcp, udp o vrrp.

Security Policies de Juniper

Las políticas de seguridad de Juniper son su implementación de firewall con estado basado en zonas. Juniper permite la creación de Address Books y Address Sets que no son más que variables que pueden ser utilizadas para referirse a direcciones IP o bloques de direcciones (Address Books) o a grupos de direcciones IP y/o bloques de direcciones IP (Address Sets). Estas variables pueden incluir indistintamente direcciones IPv4 y direcciones IPv6, por lo que permiten abstraerse de la versión del protocolo IP al momento de crear reglas de seguridad.

Es importante tener en cuenta que los equipos de la serie SRX y de la serie J por defecto descartan los paquetes IPv6. Para habilitar las funcionalidades de firewall con estado para el tráfico IPv6 es necesario habilitar el reenvío basado en flujos (flow-based forwarding). Para esto es necesario ejecutar el siguiente comando:

```
set security forwarding-options family inet6 mode flow-based
```

Además de esto, para que el firewall procese tráfico IPv6 es necesario configurar direcciones IPv6 en las interfaces de tránsito que reciben y reenvían el tráfico[3].

7.1.2.6. Política de seguridad

Filtrado de ICMPv6

Para el correcto funcionamiento de IPv6, es necesario permitir en el firewall una variedad de mensajes ICMPv6. En primer lugar, es necesario permitir Echo Requests y Echo Replies a través del firewall y con origen y/o destino en las interfaces del mismo con el fin de poder detectar problemas en la red y sus causas. Además, es necesario permitir otros mensajes como por ejemplo Destination Unreachable y Parameter Problem (Unrecognizaed Next Header, Unrecognized Extension Header).

Los nodos IPv6 en un mismo enlace utilizan el protocolo Neighbor Discovery para detectar la presencia de nodos vecinos, determinar las direcciones de capa de enlace de estos nodos, encontrar routers y mantener información de alcanzabilidad a los vecinos activos. Por este motivo también es necesario permitir los mensajes multicast de ICMPv6 que son parte del protocolo Neighbor Discovery (Router Advertisement, Router Solicitation, Neighbor Advertisement, Neighbor Solicitation).

Por otro lado, en IPv6 la fragmentación en los routers intermedios no es posible. Únicamente el nodo de origen puede fragmentar paquetes. Si un router recibe un paquete que es demasiado grande para el enlace, el paquete es descartado y un mensaje ICMPv6 (Packet Too Big) es enviado al originador del paquete para informarle la situación. Por este motivo, este tipo de mensajes deben ser permitidos en el firewall. Caso contrario, los paquetes demasiado grandes serían descartados silenciosamente y el nodo de origen seguiría intentando enviar paquetes del mismo tamaño.

7.1.2.7. Consideraciones de autoconfiguración

Al utilizar mecanismos de autoconfiguración en una red es importante tener en cuenta que el identificador de red de un dispositivo depende de la dirección MAC de la tarjeta de red del mismo. Si se incluyen en un firewall reglas que involucren direcciones IP autoconfiguradas, es necesario tener en cuenta que si por algún motivo un dispositivo requiere un cambio de tarjeta de red, se deberán actualizar las reglas que corresponda.

También es importante tener en cuenta que en las interfaces que lo requieran se deberá permitir el tráfico DHCP así también como el tráfico DHCPv6.

7.1.2.8. Familias de direcciones

Al desplegar IPv6 probablemente se opte por tener una red dual-stack, es decir que los dispositivos de la red tendrán implementado tanto el stack IPv4 como el stack IPv6, por lo que en sus interfaces tendrán configurada una dirección IPv4 y una o más direcciones IPv6. Por lo tanto, será necesario duplicar las reglas para considerar tanto la familia de direcciones inet (IPv4) como la familia de direcciones inet6 (IPv6).

En algunas implementaciones de firewall será posible utilizar variables que engloben direcciones IPv4 y direcciones IPv6 y de esta forma abstraerse de la versión del protocolo IP.

Ejemplo de reglas duplicadas:

```
allow ip from publicServerv4 to any via eth0
allow ipv6 from publicServerv6 to any via eth0
```

7.1.2.8. Filtrado de entrada/salida

Al igual que en IPv4, en IPv6 es importante realizar un filtrado de entrada y de salida para bloquear paquetes con direcciones de origen y de destino "falsificadas" (spoofed). Se deberían especificar reglas anti-spoofing de entrada para bloquear paquetes con direcciones de origen con un prefijo correspondiente a la red interna llegando por la interfaz externa. También deberían especificarse reglas de filtrado de salida para bloquear paquetes que salen de la red con direcciones de destino que contienen el prefijo global de la red local. Por último, deberían especificarse reglas que eviten que el tráfico multicast local al sitio salga hacia Internet.

Por ejemplo:

```
publicServer=publicServerv4, publicServerv6
allow all from publicServer to any via eth0
```

7.1.2.9. Consideraciones varias al crear reglas

A la hora de insertar nuevas reglas en el firewall se deben tener en cuenta ciertos aspectos de la versión 6 del protocolo IP. En primer lugar, en IPv6 los dispositivos pueden tener varias direcciones IP configuradas en una misma interfaz. Las reglas deberán ser especificadas para cada dirección que esté configurada en una interfaz.
Si se utiliza OSPFv2 como protocolo de ruteo interno para las rutas de IPv4 y OSPFv3 para las rutas de IPv6, habrá que tener en cuenta que en las interfaces del firewall que participen en este protocolo se deberá permitir el tráfico OSPF en sus dos versiones, 2 y 3.

7.1.2.10. Recomendaciones sobre paquetes que deberían rechazarse

Es recomendable rechazar paquetes salientes que contengan alguno de los siguientes prefijos de uso especial en el campo de dirección de origen:

- Prefijos que contengan direcciones loopback (::1/128), direcciones no especificadas (::/128), direcciones reservadas por el IETF (direcciones IPv6 que solían ser compatibles con IPv4) (::/96) y direcciones IPv6 mapeadas a IPv4 (::ffff:0:0/96).
- Direcciones reservadas por el IETF que solían ser direcciones locales al sitio (site-local) (fec0::/10)
- Direcciones Unique-local (fc00::/7)
- Direcciones multicast (ff00::/8)
- Direcciones de documentación (2001:db8::/32)

Los paquetes de ICMPv6 con dirección no especificada (::/128) son necesarios en los mecanismos de detección de dirección duplicada (DAD - Duplicate Address Detection), por lo que no deberían rechazarse estos paquetes.

Los paquetes de ICMPv6 con dirección no especificada (::/128) son necesarios en los mecanismos de detección de dirección duplicada (DAD - Duplicate Address Detection), por lo que no deberían rechazarse estos paquetes.

Por otro lado, en las reglas que apliquen a tráfico entrante deberían rechazarse paquetes que tengan el prefijo propio de la red en el campo de dirección de origen.

7.2_ Servicios

7.2.1. Configuracion básica de un servicio DHCPDv6 en Linux

7.2.1.1. Ejemplo en Ubuntu 13.04

Instalacion:

Como es tradicional sobre Linux existen muchas maneras de realizar la instalacion, generalmente las maneras principales son compilando los fuentes o utilizar las herramientas de gestion de paquetes que ofrecen las distintas distribuciones. En este ejemplo utilizaremos el gestor apt-* existente en Ubuntu.

Procedimiento:

1) Agregar la siguiente linea al final de /etc/apt/sources.list:
   deb http://ftp.de.debian.org/debian experimental main
2) Eliminar cualquier dhcp de ISC que tuviesemos antes:
   #apt-get purge isc-dhcp-server (notese que podemos usar purge o remove, lo dejo a tu criterio)
3) Actualizar la DB de repositorios:
   #apt-get update
4) Instalar el isc-dhcp-server indicando que use el repositorio experimental:
   #apt-get -t experimental install isc-dhcp-server
5) Archivos ejemplos de configuracion:
   Ejemplo de: /etc/dhcp/dhcpd.conf

```
subnet6 2001:db8:0:1::/64 {
#        # Range for clients
        range6 2001:db8:0:1::129 2001:db8:0:1::254;
#        # Additional options
        option dhcp6.name-servers 22001:db8:4::2;
        option dhcp6.domain-search “domain.example”;
}
```

Ejemplo de /etc/dhcp/dhcpd6.conf:

```
default-lease-time 600;
max-lease-time 7200;
log-facility local7;
option dhcp6.name-servers 2001:db8:4::2;
option dhcp6.domain-search “domain.example”;
```

6) Levantar isc-dhcp-server:
/usr/sbin/dhcpd -6 -f -cf /etc/dhcp/dhcpd.conf eth5

Importante: La configuración del DHCP(d) debe estar funcionando, sino, el DHCPD no levantará.

Es posible que al momemento de correr el servicios dhcpd tengas algun inconveniente, aqui te mencionamos los 3 principales con los que nos hemos encontrado y sus posibles soluciones:

7.2.1.2. Error 1

```
Can’t open lease database /var/lib/dhcp/dhcpd6.leases: No such file or
directory --  check for failed database rewrite attempt! Ejemplo:root@
IPv6-RTR:/etc#  /usr/sbin/dhcpd -6 -f -cf /etc/dhcp/dhcpd.conf eth5
Internet Systems Consortium DHCP Server 4.3.0a1 Copyright 2004-2013
Internet Systems Consortium. All rights reserved. For info, please visit
https://www.isc.org/software/dhcp/ Can’t open lease database /var/lib/
dhcp/dhcpd6.leases: No such file or directory --  check for failed
database rewrite attempt! Please read the dhcpd.leases manual page if you
don’t know what to do about this. root@IPv6-RTR:/etc# touch /var/lib/
dhcp/dhcpd6.leases
```

Solución a Error 1:
#touch /var/lib/dhcp/dhcpd6.leases
Adicionalmente verificar si el usuario con el que se esta ejecutando dhcpd posee escritura en /var/lib/dhcp Probablemente haya que tambien realizar: #cd /var/lib/ #chown -R root.root dhcp

7.2.1.3. Error 2

Solución a Error 2:

```
No subnet6 declaration for eth5 (fe80::a00:27ff:fee7:b7c)
Ejemplo del error:
root@IPv6-RTR:/etc/dhcp# /usr/sbin/dhcpd -6 -f -cf /etc/dhcp/dhcpd.
conf eth5 Internet Systems Consortium DHCP Server 4.3.0a1
Copyright 2004-2013 Internet Systems Consortium. All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/ Wrote 0
NA, 0 TA, 0 PD leases to lease file. No subnet6 declaration for eth5
(fe80::a00:27ff:fee7:b7c). ** Ignoring requests on eth5.  If this is
not what   you want, please write a subnet6 declaration   in your
dhcpd.conf file for the network segment   to which interface eth5 is
attached. ** Not configured to listen on any interfaces!
```

Quitar la declaracion de subnet6 en dhcpd6.conf y copiarla a dhcpd.conf

7.2.1.4. Error 3

```
Can't set SO_REUSEPORT option on dhcp socket: Protocol not available
Ejemplo del error: root@IPv6-RTR:/etc/dhcp# /usr/sbin/dhcpd -6 -f -cf
/etc/dhcp/dhcpd.conf eth5 Internet Systems Consortium DHCP Server
4.3.0a1 Copyright 2004-2013 Internet Systems Consortium.
All rights reserved. For info, please visit https://www.isc.org/
software/dhcp/ Wrote 0 NA, 0 TA, 0 PD leases to lease file.
Can't set SO_REUSEPORT option on dhcp socket: Protocol not available
```

Solución a Error 3:
Actualizar el kernel a una version >= 3.9 SO_REUSEPORT es requerido por ISC DHCP 4.3 por ello es necesario tener un kernel > 3.9

7.2.2. Configuracion básica de un servidor Apache en Linux (IPv6)

Apache tiene soporte IPv6 desde la version 2.0.14 y habilitarlo es muy sencillo. Aqui te dejamos un sencillo ejemplo sobre Ubuntu 13.04:

7.2.2.1. Procedimiento:

1) Identificar que direccion IPv6 tiene el servidor.

```
#ifconfig venet0

venet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-
00-00-00-00-00
          inet addr:127.0.0.2  P-t-P:127.0.0.2  Bcast:0.0.0.0
Mask:255.255.255.255
          inet6 addr: 2001:db8:12:3452::89/128 Scope:Global
          UP BROADCAST POINTOPOINT RUNNING NOARP  MTU:1500  Metric:1
          RX packets:75946 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45439 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:106881827 (106.8 MB)  TX bytes:3620283 (3.6 MB)
```

2) Indicarle a Apache que escuche en dicha direccion IPv6:
En el archivo /etc/apache2/ports2.conf agregar la directiva:

```
Listen [2001:db8:12:3452::89]:80
NameVirtualHost [2001:db8:12:3452::89]:80
```

Notese que corresponde a la direccion IPv6 que conseguimos en el paso uno y se encuentra entre llaves ([ ]). Es obligatorio que se encuentre entre llaves para poder separar los ":" de la

direccion IP y del puerto. Esto es una pequena diferencia con la directiva Listen de IPv4.

3) Configurar un VirtualHost
En este paso la configuracion la realizarás tradicionalmente en los archivos y sitios que desees. Este es un ejemplo en el sitio por defecto de apache.
En el archivo: /etc/apache2/sites-available/default agregar la siguiente directiva:

```
<VirtualHost [2001:db8:12:3452::89]:80>
        ServerAdmin tu@example.com
        ServerName my.example.com
        ErrorLog ${APACHE_LOG_DIR}/my-example-com-error.log
        CustomLog ${APACHE_LOG_DIR}/my-example-com-access.log combined
        DocumentRoot /var/www
</VirtualHost>
```

4) Revisar que Apache este escuchando correctamente sobre IPv6:

```
#netstat -pan | grep apache
tcp        0      0 192.168.190.89:80       0.0.0.0:*
LISTEN      533/apache2
tcp6       0      0 2001:db8:12:3452::89:80 :::*
LISTEN      533/apache2
```

La salida anterior indica que Apache se encuentra escuchando correctamente en el puerto 80 tanto de IPv4 (tcp) y de IPv6 (tcp6). El puerto se encuentra en estado LISTEN y el numero de proceso de apache2 es 533.

7.2.3. Configuración básica de un servidor DNS en Linux (IPv6)

7.2.3.1. Introduccion:

Habilitar IPv6 en el servidor BIND es muy sencillo y en muchas ocasiones este servicio es considerado el más simple para implementar y un comienzo para el despliegue de IPv6 en la red.

Recordemos algunas pocas cosas sobre DNS en el mundo de IPv6:

- Los registros que apuntan a direcciones IPv6 son AAAA
- Los registros A6 son obsoletos y no deben ser utilizados mas
- Los registros para configurar los reversos se mantienen y son PTR
- El RFC principal que habla sobre IPv6 y DNS es: RFC 3596
- Otro buen documento que habla sobre DNS e IPv6 es el RFC 4472

llamado “Operational Considerations and Issues with IPv6 DNS”

7.2.3.2. Procedimiento:
En esta ocasión utilizaremos el servidor BIND de la ISC como prestador de servicios:

1) Indicarle a Bind que escuche en IPv6:
En el archivo /etc/bind/named.conf (quizas named.conf.options). Colocar:

```
options {
     listen-on-v6 { any; };
}
```

En el ejemplo anterior BIND escucharia en todas las direcciones IPv6 que tenga el servidor.

2) Restringir la transferencia de zona por IPv6 solo a una lista de servidores permitidos:

```
options {
       allow-transfer {
        192.168.12.1;
        2001:db8:3::3;
       };
```

Notese que la misma directiva allow-transfer de v4 es utilizada para v6.

3) Permitir consultas únicamente desde los clientes IPv6 legítimos:

```
options {
     allow-query {
        192.168.125.0/24;
        2001:db8::/32;
       };
```

Ejemplos de manipulación de registros:
a) Queremos que www apunte a 2001:db8:2006:1::1:1

```
www                  IN  AAAA  2001:db8:3006:1::1:1
```

b) Recordemos que host puede tener tanto IPv6 como IPv6.
Un ejemplo en esta situación sería:

```
www             IN  A     192.168.0.49
www             IN  AAAA  2001:db8:3006:1::1:1
```

En aquellas situaciones donde el destino y el origen son Dual Stack, el cliente decide que protocolo utilizar. Esto viene definido entre la aplicacion y el sistema operativo. Para mas información leer el RFC 6555 (Happy Eyeballs)

c) Un ejemplo de tener un servidor de correo en IPv6 sería:

```
example.com             MX      10  mail.example.com.
mail.example.com    A            192.168.125.5
mail.example.com    AAAA         2001:db8:1::5
```

d) En el ejemplo anterior podemos incluso hacer mezclas de sabores entre las prioridades de IPv6 e IPv4. Por ejemplo

```
example.com               MX      10  mail6.example.com.
example.com               MX      20  mail4.example.com.
mail4.example.com         A           192.168.125.5
mail6.example.com         AAAA       2001:db8:1::5
```

Los números 10 y 20 indican las prioridades del mail server, mientras más bajo sea este numero mayor es la prioridad. En el ejemplo anterior un MTA intentaría entregar el correo al server con prioridad 10, si es infructuoso se conectaria al mail4.example.com.

e) En el mundo de IPv6, round robin DNS tambien es soportado. Ejemplo

```
www.example.com           AAAA        2001:db8:1::4
www.example.com         AAAA        2001:db8:1::5
```

En el ejemplo anterior la mitad de las conexiones irian al server ::4 y la otra mitad al ::5

f) Ejemplo de un CNAME e IPv6

```
webnew.example.com    AAAA        2001:db8:1::4
www.example.com         CNAME     webnew.example.com
```

```
root@vm1:~# netstat -pan | grep named | grep udp6
udp6       0      0 :::53                   :::*
696/named

root@vm1:~# netstat -pan | grep named | grep tcp6
tcp6       0      0 :::53                   :::*
```

7.3_ Lectura complementaria

http://www.nsa.gov/ia/_files/ipv6/I733-041R-2007.pdf

http://ipv6.niif.hu/m/IPv6firewallsandSecurity_eng

http://www.freebsd.org/cgi/man.cgi?query=ipfw&apropos=0&sektion=0&manpath=FreeBSD+9.1-RELEASE&arch=default&format=html

http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-sec_trfltr_fw.html#wp1072407

http://www.sixxs.net/wiki/IPv6_Firewalling

http://www.hamilton.ie/publications/orla_mcgann_thesis.pdf

http://www.team-cymru.org/ReadingRoom/Templates/IPv6Routers/xsp-recommendations.html

7.4_ Referencias

[1] http://www.nsa.gov/ia/_files/ipv6/I733-041R-2007.pdf

[2] Es posible ver en detalle las releases de IOS que soportan filtrado de paquetes IPv6 en el siguiente enlace: http://www.cisco.com/en/US/docs/ios/ipv6/configuration/guide/ip6-sec_trfltr_fw.html#wp1072407

[3] http://www.juniper.net/techpubs/software/junos-security/junos-security10.2/junos-security-swconfig-security/topic-45426.html

"IPv6 para operadores" representa el siguiente paso a ***"IPv6 para todos"***, publicación que el capítulo argentino de Internet Society lanzó en el año 2010. Si bien en términos de disponibilidad de direcciones IPv4 la situación cuatro años atrás era muy diferente a la actual, existía ya una gran preocupación entre las organizaciones técnicas de Internet por la falta de concientización e interés percibidos en la comunidad con respecto al despliegue de IPv6.

Esa primer edición del libro sirvió para explicar, con ejemplos simples y concretos, como utilizar IPv6 en diferentes entornos (en el hogar, en la oficina, en redes académicas, etc.).

Actualmente la situación es muy diferente. No hay direcciones IPv4 disponibles para grandes asignaciones en los registros regionales y los principales proveedores de contenido ya tienen sus páginas disponibles en IPv6. Sin embargo, a pesar de la mayor difusión y conocimiento, el tráfico IPv6 sigue siendo demasiado bajo debido en gran parte a la poca adopción entre operadores.

Por esta razón y, aprovechando el éxito del primer libro, es que el capítulo argentino de Internet Society invitó a los principales expertos de la región a escribir un capítulo dedicado a la implementación de IPv6 en operadores para cada una de las áreas de operación.

Este compendio de experiencias servirá para que todos los proveedores de servicio de Internet puedan conocer en detalle y con ejemplos qué consideraciones deben ser tenidas en cuenta y cuál es el impacto del despliegue de IPv6 en cada componente de la red de un proveedor de servicio de Internet.

Internet Society
ISOC-AR
Capítulo
Argentina