Escuela Colombiana de Ingeniería Julio Garavito

Redes IPv6 al Servicio de la Red Empresarial

Proyecto de grado

PGR

Daniel Ospina Bedoya

2019

Bogotá D.C

TABLA DE CONTENIDO

INTRODUCCIÓN

Este proyecto de grado es una continuación del proyecto anteriormente realizado "Implementación de redes modernas: Software Defined Networking e IPv6", con el cual se pretende realizar la implementación del protocolo IPv6 sobre la infraestructura del laboratorio de informática, buscando aprovechar los beneficios de esta.

Este documento tiene la documentación y resultados correspondiente a la investigación e implementación realizada durante la realización del proyecto, y se basa en gran medida en el libro de proyecto [1] realizado previo a este proyecto.

DESCRIPCIÓN DEL PROYECTO

OBJETIVO GENERAL

Conocer y desarrollar los conceptos, implementaciones y mecanismos de transición del protocolo IPv6 en la infraestructura de red de cómputo de organizaciones, así como su convivencia con el protocolo IPv4; suministrando soluciones y alternativas en servicios de red y software. A través de la implementación actual en la red del Laboratorio de Informática de la Decanatura de Ingeniería de Sistemas.

OBJETIVOS ESPECÍFICOS

I. Continuar la investigación y el trabajo hecho en el proyecto de grado "Implementación de redes modernas: Software Defined Networking e IPv6", finalizando así la implementación propuesta.

II. Analizar la propuesta de MinTIC sobre esta temática para determinar su utilización en el proyecto.

III. Definir y poner en operación servicios adicionales de red y seguridad en equipos de cómputo (servidores, computadores e IoT) sobre el ambiente IPv6 para ser implementados en la red del Laboratorio y que contribuyan al beneficio de éste.

IV. Generar un central de conocimiento de IPv6 para el laboratorio de informática, donde se encuentre información general y específica del protocolo, herramientas y procesos involucrados en la implementación de IPv6.

V. Generar prácticas de laboratorio que puedan usar los estudiantes para aprender del tema.

ALCANCE DEL PROYECTO

- Investigación y documentación de la implementación de IPv6 según Guías de Transición y estándares publicados por el Ministerio de Tecnologías de la Información y las Comunicaciones (MINTIC).
- Modificación y/o puesta en marcha (si aplica) de nuevas tecnologías, metodologías o servicios según los resultados del punto anterior.
- Continuación y finalización del trabajo práctico realizado por el proyecto de grado "Implementación de redes modernas: Software Defined Networking e IPv6" en el aspecto IPv6.

ALCANCE DEL PRODUCTO

- Poner en marcha una solución para el filtro de paquetes IP, utilizando "iptables" o una solución de firewall adecuada; como continuación de uno de los criterios de aceptación del proyecto en el cual se basa este nuevo proyecto. Así como documentar los motivos de por que la solución propuesta previamente es insuficiente.
- Adecuar el actual servicio de VPN presente en el laboratorio de informática, para que tenga la capacidad de comunicarse con servidores presentes en el laboratorio que utilicen IPv6.
- Adecuar el actual software de monitoreo de servidores presente en el laboratorio "Pandora FMS" para monitorear servicios del laboratorio funcionando sobre IPv6.
- Poner en marcha nuevos servicios y/o aspectos de seguridad que se consideren necesarios o relevantes sobre IPv6, según el primer criterio de alcance del proyecto, y demás investigación.

PLANTEAMIENTO DEL PROBLEMA

El Protocolo de Internet versión 6 aunque es considerado un estándar de internet hace más de 20 años, su uso aún es limitado. Los RIRs (Registro Regional de Internet) buscan fomentar su uso y entidades gubernamentales, como MINTIC en el caso colombiano, hacen parte del apoyo a la transición y adopción de este protocolo. La pregunta entonces ronda alrededor de cuál sería el beneficio de adoptar IPv6 para una organización el día de hoy, y cómo se vería afectada al tomar un proyecto de cambio como este. Un ejemplo importante son las instituciones académicas que pueden ser las primeras en llevar a cabo esta transición, con objetivos académicos y que pueden servir como modelo de cambio para otro tipo de organizaciones.

JUSTIFICACIÓN

Los primeros protocolos de comunicación entre dispositivos fueron desarrollados a finales de 1960 e inicios de 1970, y el Protocolo de Internet versión 4 (IPv4), hoy en día su mayor problema, es el agotamiento de direcciones, debido a la rápida explosión del número de dispositivos conectados a internet; y aunque diferentes estrategias de mitigación se han desarrollado a través de los años, la adopción del Protocolo de Internet versión 6 (IPv6) permanece como la solución más efectiva a largo plazo.

Es por esto que diferentes entidades regulatorias de internet y entidades gubernamentales hacen esfuerzos por fomentar el uso de IPv6. Aun así, organizaciones no se atreven a llevar a cabo este cambio en algunos casos por desconocimiento y en otros por las implicaciones que este cambio conlleva a nivel de sus equipos, software base (middleware) y servicios a usuario.

Es por esto por lo que es importante entender y contextualizar, el razonamiento detrás de la adopción de IPv6, beneficios, implicaciones, ventajas y desventajas según las necesidades del negocio. Así como entender la metodología necesaria para llevar a cabo la transición, tanto como los servicios y recursos adicionales necesarios.

Según la circular N.º 002 de Julio 06 de 2011 del MINTIC, instituciones de educación superior privadas y públicas están llamadas a promover y divulgar la adopción de IPv6 en Colombia, a través de la apropiación de IPv6 en sus cursos regulares y escenarios de formación en tecnologías de la información y comunicación; con este motivo, el proyecto busca implementar esta tecnología en la infraestructura del Laboratorio de Informática de la Escuela Colombiana de Ingeniería, y que esta sea parte del proceso de formación en el programa.

METODOLOGÍA PROPUESTA

El desarrollo del proyecto tendrá la siguiente estructura para completar los objetivos específicos.

I. Investigación a partir de las guías de transición a IPv6 publicadas por el MINTIC.

   Y demás entidades relacionadas, así como previos proyectos de grado, artículos relacionados y demás documentación bibliográfica que pueda aportar al proyecto.

II. Si aplica, implementar servicios, soluciones o modificaciones según el paso anterior.

   Implementar y documentar sobre la infraestructura actual del laboratorio de informática, los servicios y modificaciones necesarias dados los resultados del punto anterior.

III. Continuación, investigación y finalización de la condición actual de iptables.

   Entre los resultados del proyecto anterior de IPv6, dejaron parcialmente implementada una solución de iptables, el estado actual de esta es insuficiente, por lo tanto, uno de los objetivos de este proyecto es descubrir los motivos de su no funcionamiento, y a su vez brindar una solución sobre el mismo software, o implementar una nueva solución si lo anterior no es posible.

IV. Adecuación del servicio de VLAN.

   El laboratorio actualmente cuenta con un servicio de VLAN, una de las preocupaciones actuales es la disponibilidad de este al momento que se requiera acceder a servidores y/o equipos que funcionen exclusivamente por IPv6. Por lo tanto, el objetivo será proveer este servicio, minimizando limitaciones.

V. Adecuación del servicio de Monitoreo de Servidores.

   Actualmente el laboratorio, cuenta con un sistema basado en SNMP para llevar un registro y monitoreo de servidores y equipos de alta prioridad en la red, el meta es entonces corroborar su compatibilidad con el protocolo IPv6 y agregar equipos que utilicen este protocolo al sistema.

VI. Incorporación de servicios adicionales y/o aspectos de seguridad sobre IPv6.

   Una vez completada la implementación de servicios previos, a partir de la investigación y resultados de los anteriores puntos, se formará e idealmente aplicará un plan para adicionar servicios IPv6 que puedan apoyar el laboratorio de informática y su transición a IPv6.

MARCO TEÓRICO

BACKWARDS COMPATIBILITY & TRANSITION MECHANISMS

El mayor desafío de la transición a IPv6 es, sin duda, hacer que los sistemas existentes funcionen como normalmente lo harían si estuvieran utilizando IPv4, sin hacer compromisos en el rendimiento, la confiabilidad o la funcionalidad.

Desafortunadamente, IPv6 no es compatible con IPv4, ya que fue creado desde cero, para resolver todos los problemas que surgieron con IPv4, por lo que hacerlo funcionar con su predecesor traería más problemas que soluciones a la mesa. Naturalmente, se han creado diferentes soluciones a lo largo de los años, por lo que uno o ambos protocolos se pueden usar en casi todas las situaciones.

Este documento se centra en la solución de "Traducción", que permite el tráfico IPv4 e IPv6 a través de cada dispositivo de red. Pero existen múltiples mecanismos de transición, que pueden ajustarse mejor que otro dependiendo de la infraestructura y las necesidades del administrador de la red.

Bajo la arquitectura de traducción, el objetivo es que los hosts usen solo IPv6 (aunque IPv6 e IPv4 al mismo tiempo, prefiriendo IPv6, también es posible), y la conexión a Internet se realiza bajo dos escenarios: el enrutador tiene conectividad IPv6 nativa, por lo que brinda a los hosts conectividad IPv6 nativa a los servidores IPv6 y "traduce" la comunicación entre un host IPv4 a IPv6 con la ayuda de DNS64 y NAT64, estos dos se explican con mayor profundidad en las siguientes secciones.

AUTOCONFIGURATION & LINK-LOCAL ADDRESSES

Una dirección link-local es asignada a un host mediante SLAAC (Stateless address autoconfiguration), este proceso tiene como objetivo proporcionar una configuración completamente automática de hosts, una configuración mínima (de ser necesario) de enrutadores y ningún servidor adicional [2].

Este proceso comienza con la generación, la prueba de unicidad y la asignación de la dirección Link-Local, el bloque de direcciones reservado es fe80::/10, estas direcciones son suficientes para que los hosts se comuniquen entre pares en el mismo enlace, es decir, su dominio de broadcast; cada interfaz de un dispositivo tiene sus propias direcciones link-local la cual no es enrutable.

Una vez terminado el proceso de dirección link-local, el host contacta al enrutador local y obtiene su dirección global única, la cual usa para contactar servidores fuera de su segmento de red. Este proceso es útil para redes que no se preocupan por las direcciones que se están utilizando, si es necesario, se puede implementar el protocolo DHCPv6 con estado.

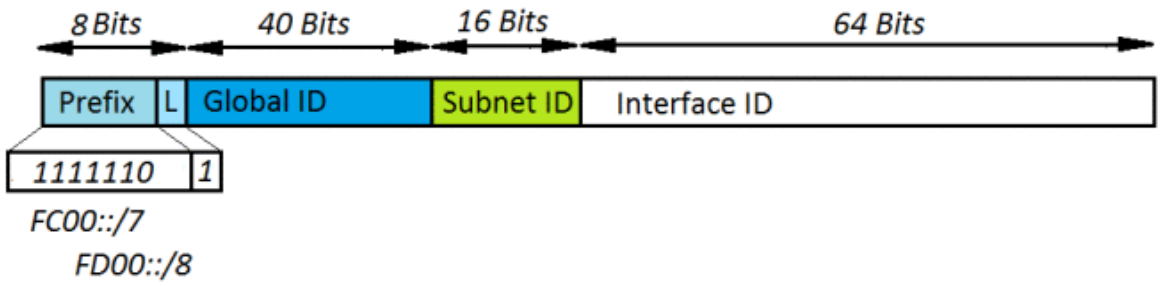
PREFIJO IPV6 RESERVADO PARA DOCUMENTACIÓN

Las direcciones en el rango 2001:db8::/32 están destinadas para fines de documentación, por lo que no deben usarse en entornos de producción, no son enrutables y deben agregarse a los filtros de paquetes. El RFC con respecto a este espacio de direcciones es el RFC 3849 [3].

UNIQUE LOCAL-ADDRESS

La dirección local única es el equivalente de los rangos de direcciones de redes privadas IPv4 para IPv6, pero con algunas condiciones.

El RFC 4193 reserva el bloque FC00::/7 para identificar direcciones IPv6 unicast locales, el octavo bit establecido como 0 no tiene uso alguno actualmente, pero puede ser definido en el futuro. El octavo bit establecido como 1 identifica direcciones asignadas localmente; los siguientes 40 bits se utilizan para crear un prefijo único global, y los siguientes 16 bits identifican una subred en un sitio, los 64 bits restantes están disponibles para los hosts en una determinada red, permitiendo un total de 18.446.744.073.709.551.616 hosts.



El prefijo único global es una identificación de 40 bits pseudo aleatoriamente autogenerada, de esta manera es poco probable que dos redes tengan una colisión, esto es útil al momento de fusión de redes, prevenir superposición del espacio de direcciones VPN entre otros escenarios.

Este espacio de direcciones no es enrutable fuera de un sitio, lo que significa que los dispositivos de enrutamiento y los protocolos deben configurarse para ignorar este prefijo [4].

DHCP

En IPv4, las direcciones de cada equipo son configuradas manualmente, o se asignan automáticamente con la ayuda de DHCP, donde cada host puede recibir una dirección estática vinculada a su dirección MAC o una dirección asignada entre un rango de direcciones disponibles; un host puede recibir dirección IP, puerta de enlace predeterminada, servidores DNS, incluso servidores de tiempo y registro, por lo tanto, es una serie de servicios de red, que los hosts confían para obtener conectividad. Pero uno de los objetivos de crear IPv6 era minimizar la cantidad de servicios en una red definida, es por eso que tenemos SLAAC,

donde los hosts, por sí mismos, pueden configurar una o más direcciones; pero a lo largo de los años, la industria destacó el DHCP como un servicio útil y necesario [5], hasta julio de 2003, cuando se publicó el primer RFC sobre DHCPv6 [6].

DHCPv6 no es equivalente a su contraparte IPv4, y debe tratarse como un servicio de red complementario, que se utiliza opcionalmente junto con la "Configuración automática de direcciones sin estado IPv6" para obtener parámetros de configuración [7].

A.Router Advertisement

La primera y más obvia diferencia con DHCP IPv4 es la opción de puerta de enlace predeterminada, DHCPv6 ignora por completo esta opción, por lo que los hosts confían en el Protocolo NDP (Neighbor Discovery Protocol) basado en ICMPv6 para obtener la puerta de enlace predeterminada y acceder a otras redes [8].

B.DUID vs MAC

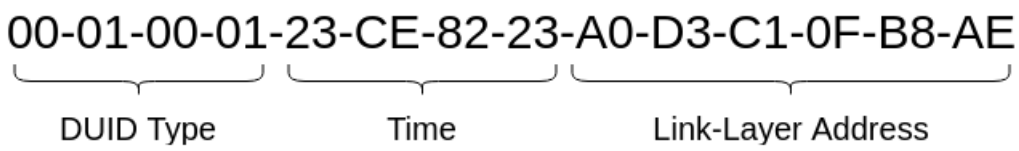
En DHCPv6, la idea es identificar de forma única y confiable un host, por lo que se elimina el uso de la dirección MAC y se utiliza el "DHCP Unique Identifier" (DUID), cada cliente o servidor tiene solo un DUID que no se modifica incluso si la NIC cambia, y cada interfaz se determina mediante el Identity Association Identifier (IAID).

Para que el DUID no cambie, debe generarse y/o almacenarse de manera confiable en un almacenamiento persistente, por lo que hay tres tipos definidos para el DUID, como dice el RFC 3315 [6]: "La motivación para tener más de un tipo de DUID es que el DUID debe ser globalmente único y también debe ser fácil de generar. El tipo de identificador globalmente único que es fácil de generar para cualquier dispositivo puede diferir bastante. Además, algunos dispositivos pueden no contener ningún almacenamiento persistente; no es posible retener un DUID generado en dicho dispositivo, por lo que el esquema DUID debe acomodarse a dichos dispositivos" [6].

- Tipo 1: Link-layer Address plus time (DUID-LLT), este tipo generalmente es utilizado por dispositivos con almacenamiento persistente, y es el mismo en todas las interfaces de red, independientemente de la dirección MAC utilizada para generarlo. Es el tipo utilizado por Windows y Linux para fines DHCPv6.
- Tipo 2 - DUID Assigned by Vendor Based on Enterprise Number (DUID-EN), este tipo es generado por el proveedor y debe almacenarse en un almacenamiento no volátil no borrable.
- Tipo 3: DUID Based on Link-layer Address (DUID-LL), este tipo es utilizado por dispositivos que tienen interfaces de red conectadas permanentemente y su único modo de almacenamiento es memoria volátil.

C. DUID-LLT

Este DUID consta de tres segmentos, los primeros 32 bits indican el tipo de DUID y el código de tipo de hardware (descrito en RFC 826), los siguientes 32 bits indican el valor de tiempo, que se representa en segundos desde la medianoche del 1 de enero de 2000, módulo 2 ^ 32, y los bits restantes (generalmente 64, pero podrían ser variables) representan la dirección de la capa de enlace.



D. Hosts en Windows y Linux

Como se dijo anteriormente, ambos sistemas operativos utilizan DUID-LLT, que funciona perfectamente bien en la mayoría de los casos, pero podría generar problemas en entornos específicos, por ejemplo, una organización que utiliza reservación de direcciones IP para sus equipos, que realiza instalaciones limpias del sistema operativo, o restaura una imagen de disco; Como el DUID se almacena en el disco, se modifica la unicidad y la persistencia del mismo.

Windows almacena el DUID en el Registro, se puede acceder a esta entrada y modificarla en:

```
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters"
```

Por lo tanto, se puede ejecutar un script que edite el registro después de una instalación limpia para modificar el segmento de tiempo a un valor anterior o predefinido.

Linux depende del cliente DHCP utilizado, generalmente este es el ISC DHCP Client, pero puede cambiar por distribución. dhclient utiliza el archivo de configuración /etc/dhcp/dhclient6.conf para modificar el DUID anunciado por el host, por lo que también se puede crear un script para modificar este archivo.

NAT

IPv6 no fue diseñado para ser usado con tecnologías NAT, hay suficientes direcciones IP como para que NAT sea necesario. Un argumento a favor de NAT podría ser que brinda seguridad a la red, pero la seguridad se obtiene gracias al ser un Stateful NAT, este nivel de seguridad es alcanzable a través de un stateful firewall.

Entonces, uno puede preguntarse por qué NAT64 es una solución propuesta para la transición de IPv6, por qué adoptar una tecnología destinada a desaparecer.

La principal fortaleza de la traducción mediante NAT es su flexibilidad y bajo costo, esta tecnología se puede implementar en casi cualquier red sin interrumpir las operaciones normales, utilizando hardware común y es lo suficientemente fácil de entender, configurar y mantener.

A.NAT64

Un servidor NAT64 traduce cada paquete IPv6 que necesita atravesar y/o alcanzar una red/servidor que funciona únicamente en IPv4, esta tecnología es propuesta y respaldada por el IETF a través de RFC 6146 [9] y RFC 6052 [10].

La operación de NAT64 es bastante sencilla, se extrae el contenido de un paquete IPv6 y se inserta en un paquete IPv4 o viceversa, los campos que se pueden asignar de un protocolo a otro como TTL/Hop Limit se asignan acordemente y el resto como flags (IPv4) y flow-label (IPv6) se descartan. Las siguientes tablas muestra la asignación de campos de los paquetes al ser traducidos [11] [12].

Traducción del encabezado IPv4 a IPv6 [13]

Campo IPv4	Campo IPv6 Equivalente
Version 4	Version 6
Internet Header Length	Descartado
Type of Service	Traffic Class
Longitud Total	Longitud Payload = Longitud Total - 20
Identification	Descartado
Flags	Descartado
Offset	Descartado
Time To Live (TTL)	Hop Limit

Protocol	Next Header
Header Checksum	Descartado
Dirección Origen	Dirección IPv6-IPv4 Mapeada
Dirección Destino	Dirección IPv6-IPv4 Mapeada
Options	Descartado

Traducción del Encabezado IPv6 a IPv4

Campo IPv6	Campo IPv4 Equivalente
Version 6	Version 4
Traffic Class	Type of Service
Flow Label	Descartado
Longitud de Payload	Longitud Total = Longitud de Payload + 20
Next Header	Protocol
Hop Limit	Time To Live (TTL)
Dirección Origen	Dirección IPv6-IPv4 Mapeada
Dirección Destino	Dirección IPv6-IPv4 Mapeada
[N/A]	Internet Header Length (IHL) = 5
[N/A]	Header Checksum recalculado

B.Limitaciones

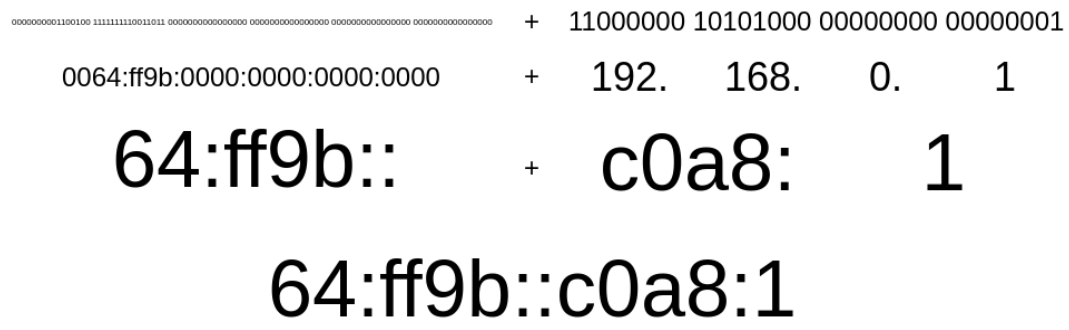
La arquitectura de traducción no es infalible, se espera que cada software con capacidades de red soporte el protocolo IPv6; y aunque software reconocido tiene capacidades IPv6, es muy posible encontrar que generalmente las herramientas especializadas no tienen soporte para IPv6 [14].

DNS

Desde la perspectiva de DNS, IPv6 no es una novedad que cambie en mayor medida su funcionamiento, para que se pueda acceder a un servidor IPv6 por nombre, el servidor DNS debe tener un registro AAAA, de resto, todo se mantiene igual. Aunque, al implementar Traducción como tecnología de transición a IPv6, las cosas se ponen interesantes.

A.DNS64

Según la arquitectura de traducción, si un cliente con solo capacidades IPv6 quiere conectarse a un servidor IPv4 fuera de su red, nuestro enrutador IPv6/IPv4 debe saber, desde el paquete IPv6, cómo llegar al servidor IPv4, esto se hace utilizando el prefijo estándar 64:ff9b::/96, esto deja exactamente 32 bits donde se inserta el IPv4 de destino.



Para que esto se logre, el software de servidor DNS debe tener capacidades DNS64, BIND de ISC posee esta característica desde la versión 9.8.0.

Con esto, la comunicación entre el host y NAT64 es completamente transparente a través de IPv6, NAT64 se encarga de extraer la dirección IPv4 del paquete IPv6 y luego traducir la respuesta del servidor de IPv4 a IPv6.

B.Directorio Activo de Windows y zonas DNS

Un componente de software habitual de muchas redes es el Directorio Activo de Windows Server, la configuración básica de NAT64 y DNS64 no tiene en cuenta este escenario. Cuando se trabaja en esta situación, se pueden necesitar dos cosas: la comunicación IPv6 nativa entre los hosts, junto con la disponibilidad tanto del directorio activo como del Domain Forward Lookup Zone para los hosts.

1. Directorio Activo sobre IPv6

   Un host funcionando solamente por IPv6 puede comunicarse perfectamente con un servidor Windows IPv4 a través de NAT64, pero no solo la comunicación con otros hosts por nombre no es posible, también implica una sobrecarga y traducción innecesaria a través del servidor NAT64, esto es

especialmente malo para el rendimiento de la red cuando trabajando con alto tráfico en la red local. Windows Active Directory tiene capacidades nativas de IPv6, por lo que habilitar IPv6 debería ser suficiente para evitar estos problemas.

2. Domain Forward Lookup Zone

   Para poder comunicarse utilizando nombres de host en el dominio, el servidor DNS64 debe transferir la zona de dominio del DNS Windows, para que pueda entregar estas nuevas solicitudes DNS a los hosts IPv6, para que esto sea posible, necesitamos:

   I. Servidor Linux DNS64

      Configurar como esclavo en /etc/named.conf

```
type                                              slave;
masters   {2001:db8:1::50;};   #Active   Directory   IP
allow-tranfers                        {2001:db8:1::50;};
allow-notify {2001:db8:1::50;};
```

   I. Windows Server

      En el Administrador de DNS:

```
Enable                BIND                Secondaries
Enable                   Round                  Robin
Name               Checking:                MultiByte
Load  Zone  data  on  Startup:  From  AD  or  Registry
Enable              Automatic              Scavenging
Scavenging Period: [Configure as Needed]
```

      En Forward Lookup Zone

```
Allow Zone Transfer: Only servers in “Name Servers” Tab
Add DNS64 Server in the “Name Servers” Tab
```

IP PREFIX POLICY

Un host puede tener múltiples direcciones IP, tanto IPv6 como IPv4, como múltiples direcciones IPv6, las cuales al mismo tiempo tienen diferentes alcances (Address Scope) que deben considerarse al enviar un paquete a través de una red, por lo tanto, se requiere un algoritmo que seleccione la mejor dirección de origen y dirección de destino para una conexión dada. RFC3484 [15] describe un algoritmo para este propósito basado en una tabla llamada Tabla de política de prefijos o IP Prefix Policy Table.

Table 3. Default Policy Table [9]

Prefix	Precedence	Label
::1/128	50	0
::/0	40	1
::ffff:0:0/96	35	4
2002::/16	30	2
2001::/32	5	5
fc00::/7	3	13
::/96	1	3
fec0::10	1	11
3ffe::/16	1	12

La Tabla muestra la tabla actualizada propuesta por el IETF; el valor Label tiene como objetivo que prefijos con el mismo label sean preferidos para la comunicación, y el valor de precedencia le dice al algoritmo que prefiera prefijos de mayor valor. Las direcciones IPv4 son mapeadas (de manera similar a 64:ff9b::/96) al prefijo ::ffff:0:0/96

El RFC 6724 [16] y el RFC 3484 revisión 03 [17] establecen que esta tabla debe ser configurable y que los administradores de red pueden agregar nuevas filas de acuerdo con sus necesidades (incluso automáticamente).

En un escenario NAT64 + DNS64 con IPv4 habilitado al mismo tiempo, esta tabla no dará preferencia al tráfico IPv6 a través de NAT64. Por ejemplo, una solicitud hecha a ietf.com tendrá como resultado:

```
A     4.31.198.44
AAAA  64:ff9b::41f:c02c
Dirección de origen = 10.0.1.85 o fdd2:dedc:9a1f:27f4::1:85
```

Aquí la dirección de origen se encuentra dentro de fc00::/7 y su etiqueta es 13. La dirección de destino está en el rango ::/0 y obtiene una etiqueta de 1. Entonces se prefiere IPv4 ya que las etiquetas (label) no coinciden.

Configurar la tabla para que sólo los labels coincidan resuelve este problema, y se preferirá el tráfico IPv6 cuando el host destino está en internet (64:ff9b::/96), pero esto no contempla los casos donde la dirección destino es una dirección local (fd00::/8), en el siguiente caso:

```
A     10.0.0.65
AAAA  fdd2:dedc:9a1f:27f4::65
Dirección de origen = 10.0.1.85 o fdd2:dedc:9a1f:27f4::1:85
```

Tanto Label 4 con IPv4 como Label 13 con IPv6 coinciden:

```
Origen: 10.0.1.85 (Label 4) - Destino: 10.0.0.65 (Label 4)


Origen:      fdd2:dedc:9a1f:27f4::1:85        (Label 13)
Destino:     fdd2:dedc:9a1f:27f4::65          (Label 13)
```

Por lo que el algoritmo recurre al valor de precedencia más alto, en este caso 35, prefiriendo IPv4. En vez de Precedencia = 3 para direcciones locales ULA.

AVISO LIBRO DE PROYECTO IPV6

CONTENIDO DEL MARCO TEÓRICO

Este libro incluye el conocimiento adicional generado durante el presente proyecto de grado, y funciona como un tomo o volumen adicional al índice de conocimiento del proyecto general de implementación de IPv6 en el laboratorio de informática que incluye: “Implementación de redes modernas: Software Defined Networking e IPv6” y “Redes IPv6 al Servicio de la Red Empresarial” y posteriores proyectos que puedan definirse.

La lectura de este Libro de Proyecto supone el conocimiento presente en el anterior Libro de Proyecto y los conocimientos base del tema.

SECCIONES ACTUALIZADAS

En vista de los cambios realizados desde la finalización del proyecto “Implementación de redes modernas: Software Defined Networking e IPv6” y los resultados presentados durante este proyecto, se identifica y avisa que las siguientes secciones del Libro de Proyecto del primer proyecto de grado sobre IPv6 no representan con precisión la configuración y/o validez sobre la implementación en el laboratorio de informática, pero sigue siendo fuente de conocimiento y herramienta de comparación, así como guías de instalación que deben ser consultadas de ser necesario. El presente Libro de Proyecto, indica en secciones posteriores que diferencias se generan frente al anterior proyecto.

- 6.3.2 Configuración e instalación de NAT64
- 6.3.3 Configuración e instalación de DNS64
- 6.5.3 Iptables

SISTEMAS OPERATIVOS

Durante la realización de proyecto, y la documentación realizada en este libro de proyecto se trabajó y documentó sobre los siguientes sistemas operativos, por lo tanto es necesario tener en cuenta las diferencias generadas por actualizaciones o el uso de otros sistemas operativos:

- Slackware 14.2 para Servidores.
- Linux Ubuntu 18.04 para Estaciones de Trabajo
- Windows 10 Build 1903 para Estaciones de Trabajo.

SOLUCIÓN DE PROBLEMAS

Como primera instancia del proyecto, era necesario encontrar soluciones a problemas que surgieron con los resultados dejados por los integrantes del proyecto anterior, a partir de esto lograr estabilidad en los servicios de IPv6 en el laboratorio.

Esta sección entonces pretende describir los problemas encontrados y el razonamiento detrás de las soluciones propuestas.

DUID - DHCPv6

Al intentar utilizar IPv6 en el laboratorio al iniciar el semestre, se observó que los computadores que debían navegar por IPv6 no tenían IP asignada, lo cual se identificó brevemente como un problema del DHCP, donde los DUID registrados en el archivos de configuración en el servidor DHCP no eran equivalentes al computador correspondiente del laboratorio. No solamente los DUID no eran correspondientes, sino también algunos eran idénticos entre computadores.

Esta situación indica una falta de consistencia en el servicio de DHCPv6 a través del tiempo, lo cual no es adecuado para el laboratorio.

La investigación llevó a concluir que la causa de los problemas era el método de generación y guardado del DUID.

Uno de los procedimientos usuales y recurrentes en el laboratorio es hacer copias idénticas de cada computador todos los semestres, y dado que el DUID se almacena en el disco duro (específicamente en el registro para el caso de Windows), el DUID de varios equipos puede ser el mismo, causando conflictos con el DHCP.

Para solucionar esta situación, se generó un script que extrae la dirección MAC del equipo, y la inserta con un “Tag del Tiempo” (EC-10-EC-10) predefinido por el laboratorio, y modifica el DUID en el registro de Windows o el archivo de configuración en el caso de Linux.

SCRIPT DUID WINDOWS

Este batch script imprime la información de los adaptadores de red actuales del computador, extrae la dirección MAC, modifica y muestra al usuario el DUID que se insertará en el Windows Registry.

```
@echo off
:: THIS IS A COMMENT ::

:: Change next line if at runtime the MAC Address is not correct ::
set ADAPTER=Realtek

echo Printing all MAC Addresses in PC...
getmac /v
echo ============================================================
echo.

:: Get MAC Address based on the ADAPTER variable, and prompts the user to check it ::
for /f "usebackq tokens=3 delims=," %%a in (`getmac /fo csv /v ^| find "%ADAPTER%"`) do set MAC=%%~a
echo Please check the MAC Address is Correct: %MAC%
echo.
echo ============================================================
echo If it is not correct, edit the script in line 5 to match an unique word of the network adapter. Using the command "getmac
/fo csv /v", where each line is a network adapter. Usually words like "Intel" or "Realtek" are good enough.
echo ============================================================
echo.

:: Remove dashes from MAC address and set the prefix ::
set MAC=%MAC:-=%
set PREFIJO=00010001ec10ec10

:: Ask user to cancel execution if something is wrong ::
echo Setting DUID as: %PREFIJO%%MAC%
echo if MAC Address is Wrong please press CTRL+C now to prevent changes in Windows Registry
pause

:: Add to the registry the value (/v) of type (/t) binary with the DUID as data (/d) and without prompting confirmation (/f)::
REG ADD HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip6\Parameters /v Dhcpv6DUID /t REG_BINARY /d %PREFIJO%%MAC% /f

echo Please Restart Windows to Apply Changes
pause
```

El script contiene comentarios (::) que explican cada función de este, este script depende de que el adaptador seleccionado sea único, por lo tanto al momento de ejecución pide al usuario confirmar la información y da indicaciones para modificar el script si es necesario (lo cual usualmente requiere cambiar una línea).

SCRIPT DUID LINUX

Este Shell Script confirma que se está corriendo como root, extrae la dirección MAC, e inserta el DUID en el archivo de configuración /etc/dhcp/dhclient6.conf

El script fue probado en Ubuntu 18.04, distribución usada en laboratorio al momento de generar esta guía.

```
#!/bin/sh

# Make sure only root can run our script
if [ "$(id -u)" != "0" ]; then
   echo "This script must be run as root" 1>&2
   exit 1
fi

#Look for MAC Address using ip addr show
#If the script fails change 'link/ether' to match the output of the line that shows the mac address
#In short, grep gets the line that show MAC Address and awk gets the first "word" that line
MAC_ADDRESS=$(ip addr show | grep -Po 'link/ether \K.*$' | awk '{print $1;}')
echo MAC Address used in DUID: $MAC_ADDRESS
rm -rf /etc/dhcp/dhclient6.conf
touch /etc/dhcp/dhclient6.conf
echo interface "eno1" { >> /etc/dhcp/dhclient6.conf
echo "    " send dhcp6.client-id 00:01:00:01:ec:10:ec:10:$MAC_ADDRESS\; >> /etc/dhcp/dhclient6.conf
echo } >> /etc/dhcp/dhclient6.conf
echo DHCP is now using DUID : 00:01:00:01:ec:10:ec:10:$MAC_ADDRESS
echo Restart to apply changes.
```

IPTABLES - FIREWALL

Pre Jool

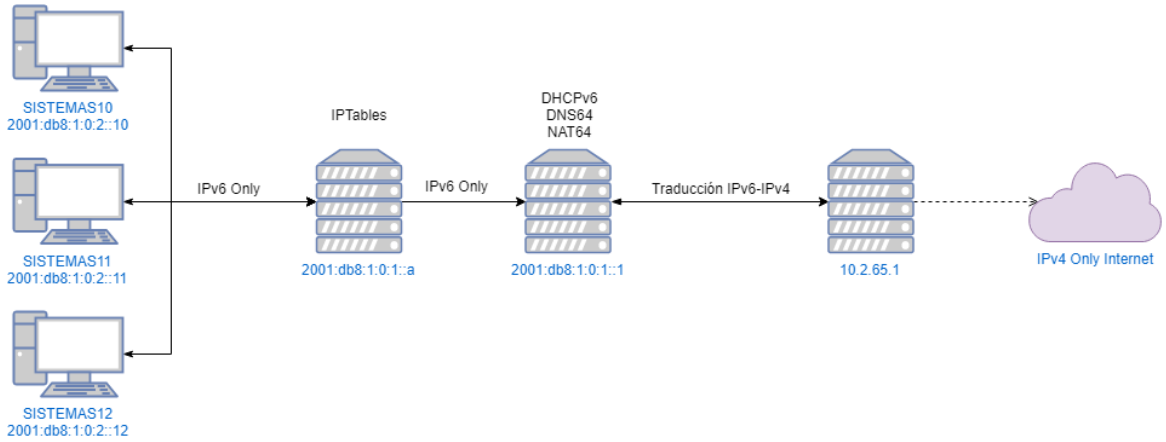
Nota: La sección Pre Jool no tiene uso ahora, dado que se encontró una mejor solución, se mantiene con objetivos de documentación. Pasar a Iptables con Jool.

Uno de los trabajos pendientes del primer proyecto de IPv6 fue la implementación de IPTables, usada para restringir el acceso a internet a estudiantes durante parciales u otros eventos.

El primer paso se siguió fue comprobar la hipótesis dejada por el anterior grupo, donde se argumentaba que por medio de un mensaje anycast el computador que debía tener la conexión limitada, buscada otro computador que le brindara esta conexión. Esta hipótesis se refuta, al tener en cuenta que el único computador capaz de hacer la conversión de IPv6 a IPv4 es nuestro servidor, y este, al tener reglas específicas de salida y revisar el tráfico mediante tcpdump, nunca hay paquetes de salida que cumplan con la situación indicada.

Uno de los puntos faltantes en las pruebas realizadas en el anterior proyecto, fue la habilitación del protocolo ICMPv6 en las reglas del firewall, este protocolo es necesario para la correcta comunicación entre equipos a través de IPv6. Por lo tanto se procedió a generar pruebas teniendo en cuenta estas nuevas variables. Entre las nuevas pruebas, se intentó utilizar otros softwares firewall, sin fruto alguno.

Se generó un entorno de pruebas con un computador intermedio el cual funciona como forwarder de paquetes, y que tendrá las funciones de Firewall y Router Advertisement, en las reglas, se permite todo el tráfico ICMPv6.



Iptables con Jool

Entre los cambios realizados durante PGR2, se instaló un nuevo NAT64, el cual funciona como módulo kernel de iptables en linux, y su configuración se basa en iptables (ver NAT64 y Instalación de Jool NAT64).

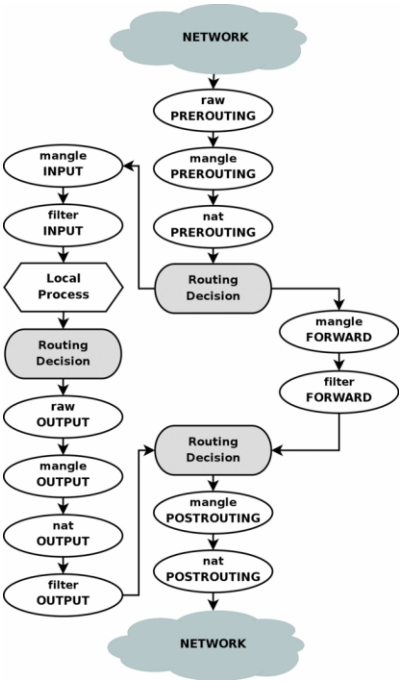
Desafortunadamente la solución no consiste en traducir las reglas de iptables a ip6tables. Por ejemplo:

```
iptables -I FORWARD -s 10.2.67.45 ! -d 149.56.16.159  -j REJECT
```

a

```
ip6tables -I FORWARD -s fd00:3c1:102::67:45 ! -d 64:ff9b::9538:109f
-j REJECT
```

Esta regla simplemente es saltada por ip6tables, esto a causa de el orden que toma iptables al procesar reglas; si una regla se encuentra en la tabla mangle (donde se configura las reglas de Jool), una vez procesada esta se envía directamente al postrouting de la misma tabla, evitando completamente la tabla por defecto filter a la cual pertenecerían las reglas anteriores [18] [19].



Por lo tanto, es necesario agregar las reglas a la tabla mangle en la etapa de prerouting, antes de las reglas para la configuración de Jool. De esta forma se hace un filtrado de paquetes, y el tráfico que no tenga reglas de filtrado saltará a la regla de Jool y será traducido normalmente.

Adicional a esto se debe tener en cuenta lo aprendido en las pruebas realizadas de iptables con WrapSix, y permitir tráfico DNS e ICMP para el funcionamiento correcto de IPv6.

Las reglas completas que deben correrse al iniciar la máquina son las siguientes:

```
#### Script de ip6tables para el correcto funcionamiento de IPv6 al bloquear internet

# ESTE SCRIPT DEBE CORRERSE AL INICIO DEL SISTEMA

# Para que el trafico IPv6 funcione correctamente, se debe permitir el tráfico ICMP y DNS
# En este caso se permite primero el trafico hacia internet con el prefijo 64:ff9b::/96
# Luego el trafico general de IPv6 (tráfico interno en fd00:3c1:102::/64)
# Finalmente el se permite el trafico por el puerto 53 (DNS) hacia Zafiro y Coral

echo "Ejecutando Script Reglas IPv6..."
ip6tables -t mangle -I PREROUTING 1 -p icmpv6 -d 64:ff9b::/96 -j JOOL
ip6tables -t mangle -I PREROUTING 2 -p icmpv6 -j ACCEPT
ip6tables -t mangle -I PREROUTING 3 -p udp -s fd00:3c1:102::/64 -d fd00:3c1:102::65:1 --dport 53 -j ACCEPT
ip6tables -t mangle -I PREROUTING 4 -p tcp -s fd00:3c1:102::/64 -d fd00:3c1:102::65:1 --dport 53 -j ACCEPT
ip6tables -t mangle -I PREROUTING 5 -p udp -s fd00:3c1:102::/64 -d fd00:3c1:102::65:3 --dport 53 -j ACCEPT
ip6tables -t mangle -I PREROUTING 6 -p tcp -s fd00:3c1:102::/64 -d fd00:3c1:102::65:3 --dport 53 -j ACCEPT

# NOTA: Documentación adicional sobre IPv6 (prefijos, ip6tables, fd00:3c1:102, etc) en el
#               proyecto: "Redes IPv6 al Servicio de la Red Empresarial" o
#               "Implementación de redes modernas: Software Defined Networking e IPv6"
```

Estas reglas están presentes en el archivo adjunto: permitir_DNS_ICMP_IPTABLES.rules

Es de notar que la acción tomada con los paquetes cambia de REJECT a DROP, dado que como mangle no es una tabla pensada para el filtro de paquetes, la acción REJECT no está disponible.

Ahora reglas que bloquean tráfico para cada computador pueden ser agregadas en la posición 7 de la tabla mangle (-I PREROUTING 7).

```
ip6tables -t mangle -I PREROUTING 7 -s fd00:3c1:102::67:176 ! -d
64:ff9b::9538:109f -j DROP

ip6tables -t mangle -I PREROUTING 7 -s fd00:3c1:102::67:177 ! -d
64:ff9b::9538:109f -j DROP

ip6tables -t mangle -I PREROUTING 7 -s fd00:3c1:102::67:178 ! -d
64:ff9b::9538:109f -j DROP

[…Continuar según X computadores]
```

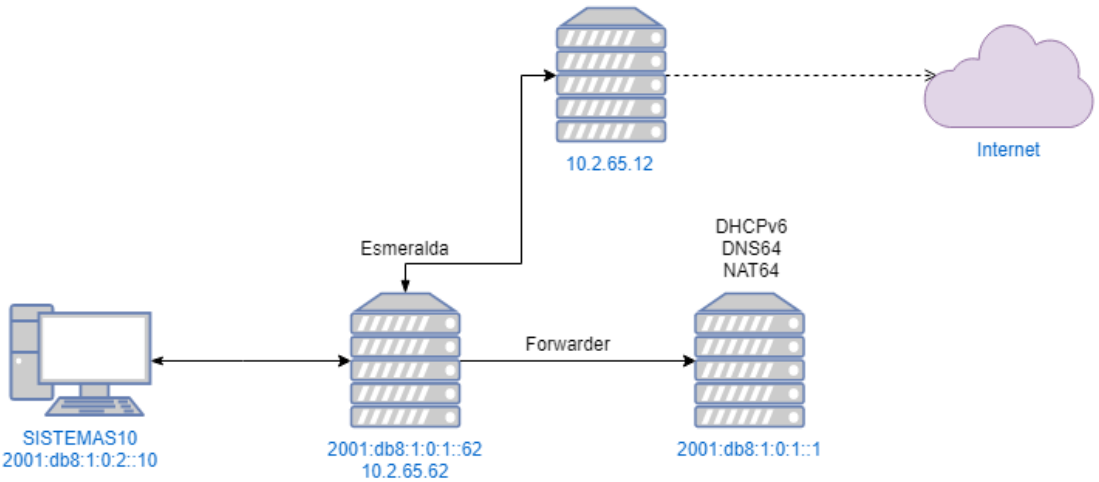
DNS

Nota: Esta sección se divide en dos, la primera se mantiene con objetivos de documentación, la sección mejora final contiene los resultados finales de PGR2

Mejora Inicial

La arquitectura para resolver consultas DNS realizada por el grupo anterior, aunque funcional, es relativamente confusa y posee cierto grado de alto acoplamiento, y tiene margen de mejora importante.

La arquitectura era la siguiente:



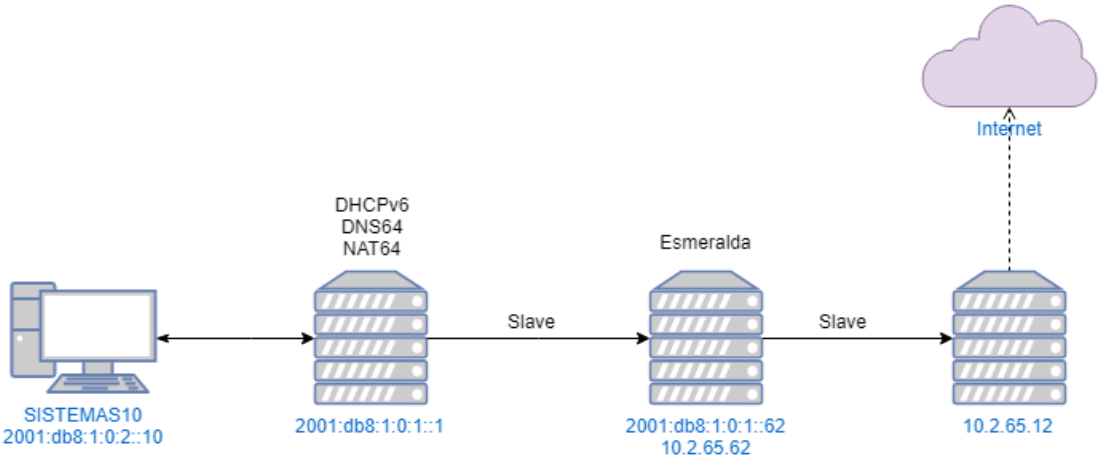
Los computadores tenían como servidor DNS uno de los directorios activos, el cual a su vez recibe respuestas DNS IPv4 del servidor principal DNS del laboratorio. Esmeralda

entonces para retornar direcciones IPv6 recurría a un forwarder que apuntaba al servidor IPv6 DNS64.

Se busca entonces, que nuestro servidor DNS64, sea un servidor autoritario para la zona is.escuelaing.edu.co para los hosts en IPv6, pero un servidor esclavo de labinfo.is.escuelaing.edu.co, de esta forma podemos obtener la información DNS del Directorio Activo sin necesidad de forwarders y delegando el funcionamiento de DNS IPv6 exclusivamente a nuestro servidor.

Ver Marco Teórico para configuración de Linux como DNS esclavo del Directorio Activo.

El resultado final es el siguiente



Mejora Final

Uno de los mayores inconvenientes de tener un servidor DNS adicional al servidor IPv4, es la necesidad de mantener dos zonas .is.escuelaing.edu.co idénticas en dos servidores diferentes. Adicional a esto los clientes Linux no reciben correctamente direcciones DNS64 (ver: Preferencia IPv6 sobre IPv4 en Linux por DNS), por lo que se tomó la decisión de migrar la funcionalidad de DNS64 a Zafiro y Coral, eliminando completamente la funcionalidad DNS64 del servidor IPv6 original.

De esta forma, los cambios hechos en el DNS principal del laboratorio Zafiro no deben ser replicados manualmente. Y se elimina un servicio, al no tener que configurar el servidor IPv6 como esclavo de Zafiro.

Como mejora adicional no correspondiente al proyecto IPv6, se configuró Zafiro como esclavo de la zona labinfo.is.escuelaing.edu.co, de esta forma los computadores pueden resolver nombres del dominio, sin necesidad de tener a esmeralda como primer DNS.

UNIQUE LOCAL ADDRESS - NUEVAS DIRECCIONES

Durante la investigación se observó que el bloque de direcciones usado (2001:db8::/32) está reservado para documentación y no debe ser usado en despliegues reales, este prefijo de direcciones no es enrutable y debe ser agregado a filtradores de paquetes como lo especifica el RFC 3849 [3].

Site-local addresses una solución ya deprecada para abordar este problema, y fue reemplazada en el 2005 por Unique Local IPv6 Unicast Address en el RFC 4193 [4].

El RFC indica que el Global ID debe ser generado de manera pseudo-aleatoria, sin embargo se tomó la decisión para facilitar procedimientos de transición tanto a estudiantes como administrativos, usar un Global ID que sea recordado fácilmente sin ser demasiado sencillo para evitar alterar en gran medida los beneficios de este; se proporciona una guía general posteriormente en el documento para cambiar este Global ID si se toma la decisión de acatar completamente los lineamientos del RFC.

El prefijo utilizado en el laboratorio de informática, al momento de generar esta guía es:

FD00:3C1:102::/64

Las IP en IPv6 son correspondientes en servidores y máquinas a IPv4 de la siguiente forma:

FD00:3C1:102::65:1 == 10.2.65.1

SOFTWARE IDENTIFICADO COMO NO COMPATIBLE CON IPV6

Los siguientes softwares fueron identificados como no compatibles con IPv6, esto quiere decir que cuando las estaciones de trabajo fueron configuradas únicamente con el protocolo IPv6 no se logró su correcto funcionamiento.

Dadas las dificultades presentadas por estos programas se decidió mantener el soporte IPv4 en la infraestructura del laboratorio. Por lo tanto IPv6 e IPv4 se mantienen activados siempre, y se confía en el algoritmo “Default Address Selection” (ver IP Prefix Policy) del sistema operativo para que seleccione el protocolo IPv4 en estos casos.

Bizagi

El servidor de Bizagi (opalo.is.escuelaing.edu.co), es uno de los servicios internos que se identificó como no compatible con IPv6, no fue posible identificar la razón, ya que aunque utiliza el protocolo TCP, el cliente de Bizagi reporta que no puede conectarse.

No se encontró documentación al respecto en internet, y se intentó contactar con soporte al usuario de Bizagi y no se obtuvo respuesta. Se espera que en versiones más recientes de este software se solucione esta problemática.

Maven/Gradle

Las pruebas realizadas con este software son inconsistentes, la documentación online indica que ambos softwares tienen soporte IPv6, y el problema se encuentra en el lenguaje que se encuentra programados, Java y al parecer hay alguna relación con la versión de Java. Se cree que es posible configurarlos para que funcionen correctamente.

Virtulizadores (VirtualBox/VMware)

La funcionalidad de la red en máquina virtuales depende de la configuración de la tarjeta de red virtual para la máquina virtual. Para los casos como Bridge, Host Only e Internal Network, la configuración de Red dentro de las máquinas virtuales se puede mantener sobre IPv4, sin ningún problema (y naturalmente IPv6 también funciona), incluso si IPv4 se encuentra desactivado en la configuración del sistema operativo.

Cuando el adaptador de red de la máquina virtual se configura en modo NAT, configuraciones IPv4 en máquinas virtuales no funcionan.

Software no Académico

Software como Spotify, Steam, Discord, Skype no funcionan correctamente detrás de un NAT64, según documentación online, esto se debe a que tienen "quemadas" direcciones IPv4, y una vez se desactiva IPv4 en l máquina no logran realizar estas conexiones.

En general, software VoIP y videojuegos tienden a fallar al usar NAT64.

PRIORIDAD IPV6 SOBRE IPV4

Como se comentó previamente, el proyecto anterior dejó como prefijo de direcciones 2001:db8:1:0::/64, el cual debió ser cambiado. Uno de los efectos que provocó este cambio fue la preferencia del tráfico IPv4 sobre el tráfico IPv6 en los equipos. Investigación más profunda llevó a concluir que la configuración de la Tabla de Políticas de Prefijos requería cambios.

Para solucionar este problema, es necesario agregar una fila a la lista de prefijos con el prefijo utilizado en el laboratorio, el label igual a 1 (equivalente al prefijo ::/0) y un valor de precedencia mayor a el prefijo de direcciones IPv4 (ffff::0:0/96).

Es necesario subir el valor de precedencia, ya que para comunicaciones de la red local, tanto direcciones de origen y destino IPv6 e IPv4, recibirán un Label idéntico, y el desempate se realizaría por el valor de precedencia, el cual para preferir IPv6 debe ser mayor a 35.

El resultado final sería: Prefijo fd00:3c1:102::/64, precedencia 38, label 1.

Configuración de la Tabla de Política de Prefijos en Windows

Mostrar el estado actual de la tabla:

netsh interface ipv6 show prefixpolicies

Agregar nueva fila a la tabla:

```
netsh interface ipv6 add prefixpolicy [prefijo] [precedencia] [etiqueta] [store]
```

Ejemplo:

```
netsh interface ipv6 add prefixpolicy fd00:3c1:102::/64 38 1 store=persistent
```

store=persistent mantiene la configuración en disco.

En el repositorio git de este proyecto se encuentran scripts que realizan este proceso junto con la configuración del DUID.

Configuración de Tabla de Política de Prefijos en Linux

La política de prefijos se configura en /etc/gai.conf, si este archivo no existe, está en blanco o solo contiene comentarios, se utiliza la tabla de prefijos predeterminada.

Las etiquetas se configuran de la siguiente manera (una línea por prefijo):

label [prefijo] [valor]

La tabla de precedencia se configura de la siguiente manera (una línea por prefijo):

precedence [prefijo] [valor]

Las etiquetas también se pueden configurar con el comando ip:

ip addrlabel list

ip addrlabel add prefix [prefijo] label [etiqueta]

Los cambios realizados con el comando ip no son permanentes, por lo que se necesita un script que se ejecute en el momento del arranque [20].

Para configurar la tabla de prefijos en gai.conf, el script debe agregar todos los prefijos por defecto que indica el RFC 6724 y el RFC 3484 revisión 03, estos ya se encuentran comentados en el archivo gai.conf; y agregar el prefijo del laboratorio fd00:3c1:102::/64. En el repositorio git de este proyecto se encuentran scripts que realizan este proceso junto con la configuración del DUID.

PROBLEMAS MISCELÁNEOS

1. *Símbolo de Conectividad Windows*

Nota: Las soluciones propuestas sobre símbolo de conectividad de Windows han dejado de funcionar consistentemente al momento de finalización de este proyecto, aún así se mantiene como información oficial por parte de Microsoft y puede seguir funcionando en futuras actualizaciones. Dado que las estaciones de trabajo no recibirán únicamente IPv6, este problema deja de existir siempre y cuando las estaciones de trabajo funcionen tanto sobre IPv4 como IPv6.

Windows posee una característica que comprueba la conexión a internet en intervalos de tiempo. Esta característica no es compatible un sistema de traducción a IPv6 como el que usa el laboratorio actualmente. Esta prueba a internet está configurada en el registro de Windows, y puede ser modificada manualmente. Por lo tanto, existen tres posibles soluciones.

Todas las configuraciones de las pruebas de conectividad se encuentran en el registro en la ubicación: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NlaSvc\Parameters\Internet`

a. Desactivar la prueba de conectividad por completo.

   Cambiar a 0 la llave "EnableActiveProbing", esto desactiva por completo el triángulo que representa conectividad limitada (el círculo rojo aún estará presente), en caso de fallas de conexión a internet tanto en IPv6 como en IPv4, no es posible identificarlo de esta forma.

b. Ajustar las direcciones donde se realiza la prueba.

   En este caso, se busca que las direcciones para hacer la prueba de conectividad sean equivalentes tanto en IPv6 como en IPv4. El defecto de esta solución es que el triángulo de conectividad estará presente durante el login de Windows, pero desaparecerá una vez se inicie sesión. Un script para realizar estos cambios está presente en los documentos de este proyecto.

   - ActiveWebProbeHostV6 debe cambiarse para ser igual a ActiveWebProbeHost
   - ActiveDnsProbeContentV6 debe cambiarse a la misma IP presente en ActiveDnsProbeContent pero utilizando el prefijo de NAT64, ejemplo: 64:ff9b::131.107.255.255 == :64:ff9b::836b:ffff

c. Crear un servidor local para la prueba de conectividad.

Los contenidos de ActiveDnsProbeHostV6, ActiveDNSProbeContentV6, ActiveWebProbeHostV6, ActiveWebProbeContentV6, ActiveWebProbePathV6, deben ser modificados para apuntar a un servidor local, al utilizar IPv6 nativa y localmente la prueba de conectividad se realiza sin problema alguno.

2. *IPv6 sobre Red Inalámbrica*

Al implementar IPv6 sobre todo el laboratorio, se identificó que los computadores portátiles del edificio H, no reciben dirección IPv6. Las pruebas realizadas con routers inalámbricos propios del laboratorio, diferentes a los controlados por OSIRIS en la red "EXTREME_WPA2" dieron los siguientes resultados:

- Tanto routers de OSIRIS como el router del Laboratorio se funcionan como Access Point para la red 10.2.0.0 del Laboratorio.
- Computadores conectados a la red inalámbrica del router del laboratorio reciben correctamente dirección IPv6 por DHCPv6
- Computadores conectados a la red inalámbrica de (el/los) router(s) de OSIRIS, no reciben dirección IPv6 por DHCPv6
- Las configuraciones en los equipos son idénticas y solo es necesario cambiarse de red para ver la diferencia.
- Una captura de tráfico al conectarse a la red EXTREME_WPA2 indica que los portátiles intentan enviar el paquete multicast para DHCPv6, pero no reciben ninguna respuesta.

Se establece como hipótesis que la red EXTREME_WPA2 sea su hardware o configuración de software no soporta IPv6. Se propone como trabajo futuro solucionar esta situación.

3. *Actualización de Jool*

La primera instalación de Jool en producción se realizó sobre la versión 4.0.3, a la fecha de finalización del proyecto se actualizó Jool a la versión 4.0.6. Esta actualización trae una novedad importante y es que en versiones anteriores todo paquete que no podía ser traducido (no entra en una regla de iptables o Jool no lo reconoce) era descartado; en esta nueva versión Jool retorna el paquete que no pudo ser traducido a iptables, y este puede ser procesado por otra regla posterior.

Entre una de las posibilidades presentadas durante el proyecto, se buscó la implementación de NAT (trabajo realizado por Zafiro y Coral) y NAT64, en un mismo servidor, de esta forma se buscaba establecer a tanzanita como un reemplazo de Coral, y una vez estable esta dualidad configurar Zafiro como NAT64.

Previo a la versión 4.0.6, dado que los paquetes eran descartados, cualquier paquete que tenia como objetivo NAT (IPv4) el cual era una regla de iptables masquerade

posterior a la regla de traducción de Jool, era descartado dado que Jool no lo reconoce como paquete traducible; por lo tanto, era imposible la funcionalidad dual de NAT y NAT64.

Ahora se propone (no se pudo realizar pruebas) que como los paquetes no traducibles son retornados a iptables, estos pueden ser procesados por la regla posterior iptables para NAT (IPv4) y la funcionalidad dual NAT y NAT64 es posible. Se propone como trabajo futuro.

Para más información sobre esto ver wiki de Jool sobre la versión 4.0.6 y 4.0.3, los cambios en la wiki para cada versión están disponibles en el repositorio oficial de Jool, así como información acerca de iptables como NAT.

4. *Pruebas de Rendimiento*

Pruebas de velocidad de conexión utilizando WrapSix como NAT64

En este tipo de pruebas, uno de los problemas recurrentes es la imposibilidad de conocer la velocidad de subida, utilizando varios de estos servicios, ninguno puede terminar la prueba y falla al intentar hacer el test de subida. Se establece como hipótesis a esta situación que WrapSix no controla correctamente los paquetes SYN de subida (dada la alta incidencia de errores TCP SYN al usar WrapSix como NAT64).

Se hace la suposición que, dada esta alta incidencia de paquetes con error, la falta de velocidad y sensación de velocidad comparado con IPv4 la cual era menor según retroalimentación de estudiantes, se debía a esto. La situación se soluciona gracias a la implementación de Jool como NAT64.

Pruebas de velocidad de conexión utilizando Jool como NAT64

Al instalar NAT64 se pudo observar que los tests de velocidad eran exitosos a comparación de los resultados con WrapSix, a su vez se observó que la cantidad de incidencias de errores TCP se redujo considerablemente al usar Jool. Gracias de su constante desarrollo, Jool probó ser superior en pruebas de estabilidad y rendimiento, y navegar con IPv6 se volvió una experiencia equivalente e indiferenciable a IPv4.

5. *Preferencia IPv6 sobre IPv4 en Linux por DNS*

Al hacer pruebas con IPv6 e IPv4 funcionando simultáneamente, se observó que Ubuntu 18.04 elige el primer servidor DNS que encuentra en /etc/resolv.conf . Un ejemplo de esta lista era la siguiente:

```
10.2.65.60      # DNS Directorio Activo
10.2.65.62      # DNS Directorio Activo
```

10.2.65.1 # Zafiro
10.2.65.3 # Coral

Y aunque un servidor DNS IPv4 puede retornar respuestas IPv6 sin problema, esto es un problema si el primer servidor de la lista no retorna respuestas DNS64; por lo tanto Linux al no tener una dirección IPv6 de la respuesta del DNS, recurre a utilizar IPv4. No fue posible encontrar un solución por software para cada equipo.

Para solucionar esta situación en el laboratorio, se configuraron los servidores principales de DNS (Zafiro y Coral) como DNS64; de esta forma, un computador configurado con IPv6 hace la petición DNS a un servidor IPv4, pero aun así recibe una dirección IPv6 y puede navegar utilizando IPv6.

6. *Direcciones desconocidas en directorio activo*

Entre los procesos del laboratorio, la entrada y salida de máquinas al dominio es recurrente, esto junto a pruebas de DHCP y DNS, agregó varias direcciones IPv6 al directorio activo no reconocidas e inservibles, las cuales representaron algunos problemas menores al realizar pruebas de IPv6. Uno de las formas de ejemplarizar la situación se manifiesta al realizar un “nslookup labinfo.is.escuelaing.edu.co”, donde se espera que direcciones IPv6 de servidores como esmeralda y murano, siendo este el caso junto con otras direcciones desconocidas. Este es un punto a tener en cuenta al momento de la solución de problemas relacionados con DNS y DHCP; estas direcciones pueden ser eliminadas manualmente en alguno de los servidores del directorio activo en el DNS Manager.

7. *Manual Solución de Problemas IPv6*

Entre los entregables y documentación generada en este proyecto, se generó un Manual de Troubleshooting para la solución de posibles problemas relacionados con la infraestructura IPv6 en el laboratorio de informática, este manual está dirigido a monitores del laboratorio que requieran una guía rápida para solucionar inconvenientes IPv6, o que necesiten conocer el funcionamiento del protocolo en el laboratorio.

TRADUCCIÓN VS OTROS MÉTODOS DE TRANSICIÓN A IPV6

Existen distinto métodos de transición a IPv6, entre estos se encuentran:

- Dual Stack

  Este método opera IPv4 e IPv6 simultáneamente, bajo la misma infraestructura y usando (opcionalmente) el mismo router. Esto implica que los dispositivos no compatibles con IPv6 funcionan de manera normal con los servicios previos y es necesario mantener ambas arquitecturas al mismo tiempo.

- Tunneling

  Tunneling es la encapsulación completa de un paquete IP de un protocolo en otro, el cual se transporta de manera transparente hacia el destino, en donde el paquete original se extrae y se retransmite en una nueva red. Existen distintos métodos de tunneling, entre estos: Manual IPv6 Tunnels, Generic Routing Encapsulation IPv6 tunnels, 6to4 Tunnels, IPv6 Rapid Deployment, IPv4 Compatible Tunnels, ISATAP Tunnels.

  6to4 y Manual IPv6 Tunnels son los más usados y tienen por concepto la encapsulación del paquete IPv6 dentro de un paquete IPv4, que viaja a través de la red IPv4 de manera transparente, y es desencapsulado en otra red que funciona en IPv6.

- Translation

  Traducción es un método que convierte los paquetes de un protocolo a otro, por ejemplo, una red local que trabaja mediante IPv6 traduce los paquetes antes enviar el paquete por su conexión a Internet IPv4.

TRADUCCIÓN COMO MÉTODO ELEGIDO EN EL LABORATORIO

Dual-Stack no se consideró como opción, dado que no es posible obtener salida IPv6 nativa a internet, de ser posible cada computador requeriría una IP pública detrás de un firewall que controla el tráfico hacia y desde el laboratorio.

Traducción mediante NAT64 es el esquema más implementado, maduro y relevante para la transición a IPv6, este método no requiere alterar dispositivos de usuarios finales, ni requiere hardware especializado, y a su vez no afecta la topología de la red. Es una opción que tiene pocas consecuencias y puede ser excluido fácilmente, lo que lo hace una opción ideal para el laboratorio.

GUÍAS

Adicional a las guías presentadas a continuación se agrega como anexo el Manual de Troubleshooting para IPv6, donde están presentes guías más relevantes para e la operación diaria del laboratorio como reinicio de servicios y configuración de equipos.

TRANSICIÓN DE PREFIJOS IP

Esta guía es una serie de pasos recomendados en caso de que sea necesario alterar el bloque de direcciones IP que se está usando actualmente, con el objetivo de reducir al mínimo los tiempos de inactividad de los servicios IPv6.

El software utilizado para esta guía es: WrapSix o Jool (NAT64), RADVD (Router Advertisement), ISC DHCP dhcpd (DHCPv6), Bind 9 (DNS64). Es importante realizar un listado de servidores y máquinas que requieren revisión y cambios, así como la realización de pruebas cada que sea posible.

1. Cambiar IP en el SO de la máquina NAT64
2. Archivos de configuración DHCP
    - subnet6 [prefijo]
    - option dhcp6.name-servers [Dirección IPv6 Servidor DNS64];
    - Direcciones IP de DHCP
3. Archivos de configuración DNS
    - Entradas AAAA
    - Configuración Esclavo-Master
4. Archivos de configuración de Wrapsix/Jool y RADVD no requieren cambio alguno, revisar su correcto funcionamiento.
5. Cambiar IPs de servidores críticos (esmeralda, amatista, murano, opalo, zafiro, coral, granate)
6. En el servidor DNS Master de la zona labinfo.is.escuelaing.edu.co, cambiar configuraciones de transferencia de zonas para la nueva IP del servidor DNS64.
7. Cambiar IPs de otros servidores y computadores.
8. Realización de Pruebas
    - Conexión y funcionamiento de servicios del laboratorio (Arenas, Oracle)
    - Conexión a Internet
    - Zona DNS

AGREGAR EQUIPOS A DHCPV6

Para agregar un nuevo equipo a el DHCPv6 el proceso es muy similar a agregar un equipo en un servidor DHCP IPv4. Se necesita:

- Acceso al servidor IPv6: tanzanita.is.escuelaing.edu.co
- Dirección MAC del equipo a agregar.

Pasos

1. En tanzanita se debe acceder a /etc/dhcp/dhcpd6.conf
2. Este archivo se encuentra ordenado por salón, por lo tanto se debe buscar la sección apropiada.
3. Esta es una base para agregar el equipo:

```
host computador-X {
        host-identifier option dhcp6.client-id 00:01:00:01:EC:10:EC:10:[Dirección-MAC];
        fixed-address6 [Dirección-IPv6];
}
```

Ejemplo Completo:

Se debe notar que la sección 00:01:00:01:EC:10:EC:10 es constante en todos los equipos, y solo cambia los últimos 6 pares de caracteres hexadecimales, que se ajustan al tamaño de una dirección MAC.

```
host sistemas87 {
        host-identifier option dhcp6.client-id 00:01:00:01:EC:10:EC:10:18:60:24:DE:F2:27;
        fixed-address6 fd00:3c1:102::67:49;
}
```

4. Una vez agregado al DHCP, se reinicia el servidor, o el servicio DHCP como se indica detalladamente en “Guía de Troubleshooting - IPv6 Labinfo”, captura de pantalla de esta sección a continuación:

DHCP

Para reiniciar el servicio de DHCP se debe identificar primero la ID del demonio que está corriendo el programa "dhcpd" encargado de DHCPv6, para esto se utiliza el comando:

```
pgrep dhcp
```

Esto retornará uno o más IDs del proceso, el cual se procede a parar utilizando el comando:

```
kill -9 [id-del-proceso]
```

Ahora se puede volver a iniciar el servicio utilizando el comando:

```
dhcpd -6 -cf [archivo-de-configuracion-dhcp] [interfaz-de-red]
```

Si nada la implementación del servicio no ha cambiado desde la creación de este manual, el comando exacto sería:

```
dhcpd -6 -cf /etc/dhcp/dhcpd6.conf eth0
```

Confirmar que al iniciar el servicio no se presenten errores, el servicio entonces estaría funcionando.

5. Ahora se puede correr el script para configurar el DUID en el equipo que se quiere agregar al DCHPv6. El siguiente manual muestra los pasos para este proceso.

MODIFICAR Y CREAR SCRIPTS IPTABLES – BLOQUEO INTERNET

NOTA: Esta es una guía rápida sobre iptables, para explicación a profundidad y razonamiento, ver sección IPTables – Firewall.

El caso usual para crear scripts para bloquear acceso a recursos o internet es el bloqueo de internet exceptuando campusvirtual.escuelaing.edu.co en situaciones de exámenes. O es necesario modificar un script porque cambió una dirección IP.

Las diferencias para IPv6 son:

- ip6tables en vez de iptables.
- Las direcciones IPv6 en vez de IPv4.
- La tabla es mangle en vez de la tabla por defecto (-t mangle).
- La acción que se toma debe ser: -j DROP
- La regla se inserta en la sección PREROUTING posición 7 (-I PREROUTING 7)

Para obtener la dirección IP que se bloquea o que se permite, hay dos opciones, en un computador configurado con IPv6:

- Se hace ping al nombre del recurso, ejemplo:
    - ping campusvirtual.escuelaing.edu.co

  Esto debe devolver una dirección de la forma: 64:ff9b::

  Ejemplo campusvirtual.escuelaing.edu.co: 64:ff9b::9538:109f

- Si este recurso no tiene nombre, se puede hacer un ping a la dirección IPv4 a través de IPv6, utilizando el comando:
    - ping 64:ff9b::[dirección IPv4]

      Ejemplo:
    - ping 64:ff9b::192.168.0.1

  Esto debe retornar la dirección completamente convertida a IPv6.

Para bloquear entonces internet, exceptuando campusvirtual se hace un script de la siguiente forma:

```
ip6tables -t mangle -I PREROUTING 7 -s fd00:3c1:102::67:176 ! -d
64:ff9b::9538:109f -j DROP
ip6tables -t mangle -I PREROUTING 7 -s fd00:3c1:102::67:177 ! -d
64:ff9b::9538:109f -j DROP
ip6tables -t mangle -I PREROUTING 7 -s fd00:3c1:102::67:178 ! -d
64:ff9b::9538:109f -j DROP

[…Continuar según X computadores]
```

CORRER SCRIPT CONFIGURACIÓN DE IPV6 EN LINUX Y WINDOWS

Dos versiones de este script están disponibles en la plataforma GitLab del Laboratorio en: http://10.2.65.88/daniel.ospina-b/labinfo-eci-ipv6/tree/labinfo/DHCPv6-DUID y públicamente en https://github.com/danielospina-b/Labinfo-ECI-IPv6/tree/master/DHCPv6-DUID

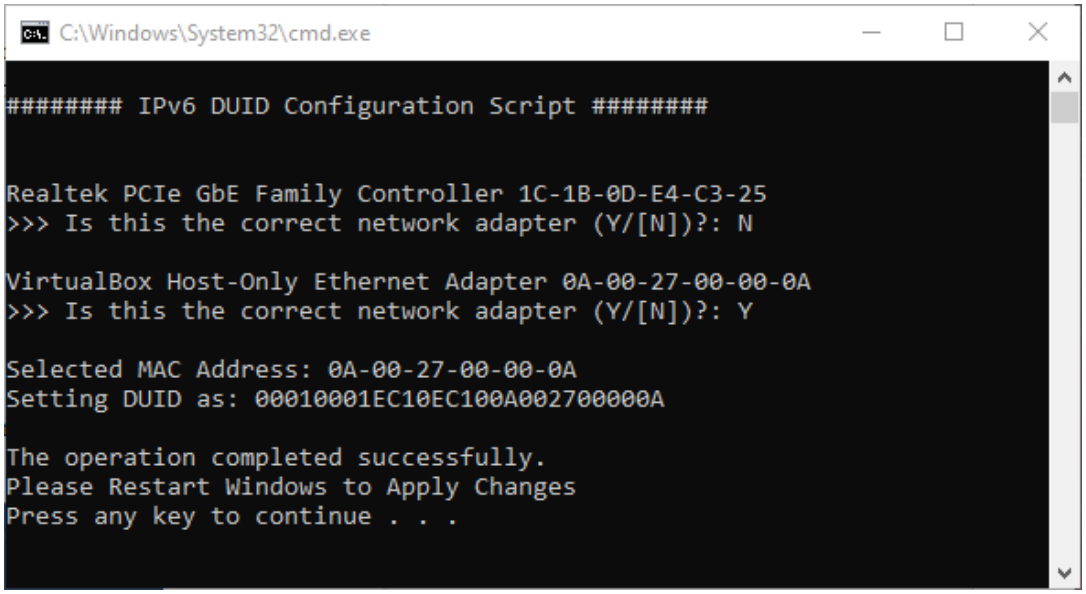
Estos scripts modifican el DUID y la política de Prefijos del sistema operativo ([1]). Estos repositorios se encuentran documentados en cuanto al funcionamiento interno de cada script.

El script "IPv6-int.bat" para Windows y "IPv6-int.sh" para Linux muestran al usuario detalladamente el proceso al usuario de configuración IPv6, y están diseñados para ser modificados y ejecutados en cualquier situación.

Los scripts de powershell "IPv6-auto-X.ps1" donde X corresponde a la marca de la tarjeta de red, configuran directamente el equipo y están diseñados con el objetivo de correrlos rápidamente en equipos conectados a la red del laboratorio donde ya se conoce la marca de la tarjeta de red[2].

Ejecutar Script interactivo completo de IPv6 en Windows

Este script se corre con Click Derecho > Correr como Administrador.



[1] Ver secciones correspondientes en el Libro de Proyecto.
[2] Ejemplo: Procesos poscopias que realizan monitores del laboratorio cada semestre.

El script preguntará por cada interface de red, si la dirección MAC es la que se quiere agregar a el DHCPv6, al responder afirmativamente mostrará los resultados de los cambios.

Ejecutar Script de IPv6 en Linux

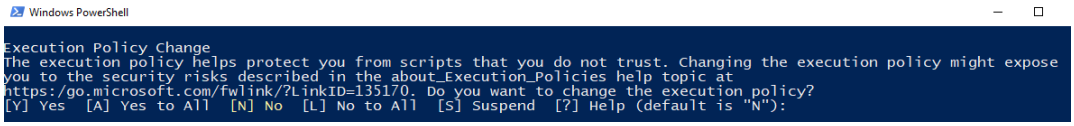
sudo sh duid-script.sh

El script mostrará todas las direcciones MAC presentes en la máquina donde el usuario deberá confirmar que se seleccionó la dirección correcta. Sí esta no es correcta, se debe modificar la línea 5 del script para ajustarse a uno de los nombres de adaptadores de red mostrados.

Ejecutar Script automático de IPv6

Este Script se corre con Click Derecho > Correr con PowerShell

El script puede mostrar el siguiente mensaje, al cual aceptamos con “Y”:



Mostrará el nombre de la máquina, el DUID que se establecerá y el resultado de cambiar el Prefix Policy.

Al establecer el Prefix Policy, hay dos posibles mensajes esperados:

- “Ok” - Se agregó correctamente

- “The object already exists” - El prefijo ya se estableció previamente y no hay cambios

Si el mensaje de error es diferente a estos dos mensajes, es necesario revisar la tabla de prefijos. (Ver IP Prefix Policy).

CAMBIAR PREFIX POLICY EN WINDOWS - PRIORIDAD IPV6 SOBRE IPV4

Esta configuración se realiza cuando se detecta que el computador prefiere tráfico IPv4 sobre IPv6, esto se puede detectar a través de Wireshark.

Este comando se debe correr en una línea de comandos o powershell con permisos de administrador. El repositorio git contiene scripts que hacen este procedimiento junto con la configuración del DUID.

```
netsh interface ipv6 add prefixpolicy [prefijo] [precedencia] [etiqueta] [store]
```

Ejemplo:

```
netsh interface ipv6 add prefixpolicy fd00:3c1:102::/64 38 1
store=persistent
```

INSTALACIÓN JOOL NAT64 EN SLACKWARE

Esta guía es específica para Slackware 14.2 en una instalación mínima usual de Labinfo, esta es una guía para la versión al momento de desarrollo de este proyecto, pero dado que Jool es un software en constante desarrollo, se recomienda mirar la documentación oficial en el sitio web oficial: https://www.jool.mx/en/install.html

Los paquetes necesarios previos a la instalación de Jool son:

- gcc-5.3.0-x86_64-3.txz
- make-4.1-x86_64-2.txz
- kernel-headers-4.4.14-x86-1.txz (Ojo, revisar que la version del kernel de linux coincida (4.4.14))
- libnl3-3.2.27-x86_64-1.txz
- iptables-1.6.0-x86_64-2.txz
- dkms-2.6.1-x86_64-1_slonly.txz
- gawk-4.1.3-x86_64-1.txz
- kernel-source-4.4.14-noarch-1.txz
- guile-2.0.11-x86_64-2.txz
- gc-7.4.2-x86_64-3.txz
- libffi-3.2.1-x86_64-1.txz
- libunistring-0.9.3-x86_64-1.txz
- libmpc-1.0.3-x86_64-1.txz
- binutils-2.26-x86_64-3.txz
- pkg-config-0.29.1-x86_64-2.txz
- flex-2.6.0-x86_64-1.txz

Los paquetes no explícitamente necesarios para el funcionamiento de Jool, pero recomendados para hacer pruebas con ping, iptables, etc.

gnutls-3.4.13-x86_64-1.txz, p11-kit-0.23.2-x86_64-1.txz, libpcap-1.7.4-x86_64-1.txz, dbus-1.10.8-x86_64-1.txz

Todas las releases oficiales de este software están presentes en: https://github.com/NICMx/Jool/releases, descargar la última versión.

Se descomprime el archivo: tar -xzf jool_4.0.6.tar.gz

Procedemos a compilar e instalar el software:

```
dkms install jool-4.0.6/
```

Si el comando dkms lanza algún error es necesario remover el módulo con:

```
dkms remove jool/[version-de-jool] --all
dkms remove jool/4.0.1 --all
```

Se solucionan los errores que salta en la instalación con dkms, repetimos el comando de instalación con dkms, y continuamos la instalación con:

```
cd jool-4.0.6/
./configure
make
make install
```

Es posible (dependiendo de la instalación del sistema operativo), que alguna dependencia o paquete no sea encontrado, será entonces necesario buscar el archivo faltante a qué paquete pertenece e instalar su correspondiente dependencia.

CONFIGURAR JOOL PARA TRABAJAR COMO NAT64

Dado que este es un software en constante desarrollo, es recomendable seguir los pasos indicados por la documentación oficial, disponible en: https://www.jool.mx/en/run-nat64.html

Se debe configurar el servidor con su dirección IPv6, y habilitar forwarding; la dirección debe obtenerla el servidor al momento de iniciar.

```
ifconfig eth0 add fd00:3c1:102::b/64
sysctl -w net.ipv4.conf.all.forwarding=1
sysctl -w net.ipv6.conf.all.forwarding=1
```

Ahora se habilita el módulo de kernel de linux con:

```
/sbin/modprobe jool
```

Se configura iptables de Jool

```
jool instance add "example" --iptables --pool6 64:ff9b::/96
ip6tables -t mangle -A PREROUTING -j JOOL --instance "example"
iptables  -t mangle -A PREROUTING -j JOOL --instance "example"
```

Ahora se puede realizar pruebas usando en equipos mediante ping:

```
ping 64:ff9b::1.1.1.1
```

Esta configuración no sobrevive reinicios, por lo tanto se debe:

- Agregar las líneas de forwarding y dirección ip a rc.local
- Agregar la configuración de iptables de jool a rc.local

- Agregar la línea de habilitación del módulo de kernel jool a rc.modules.local

EVALUAR ESTADÍSTICAS DE USO DE IPV6 VS IPV4

Cada regla de iptables tiene un contador de bytes y paquetes que han pasado por esta. Se puede aprovechar esta característica para comparar el rendimiento de IPv6 vs IPv4.

Para ver los contadores se utiliza el comando:

- NAT (IPv4): iptables -t nat -nvL
- NAT64: iptables -t mangle -nvL (Traducción IPv4 a IPv6)
  ip6tables -t mangle -nvL (Traducción IPv6 a IPv4)

Para esto se utiliza el siguiente comando en los servidores NAT (IPv4) para poner en ceros todos los contadores:

```
iptables -Z -t NAT
```

Y en el servidor IPv6:

```
iptables -Z -t mangle
```

Se establecen entonces los contadores en ceros y se vuelven a revisar en un periodo determinado de tiempo, y se calculan las diferencias.

MODIFICACIÓN DEL REPOSITORIO GIT DE IPV6

Este repositorio se anexa como resultado del proyecto y tiene dos copias remotas: una en la plataforma GitLab de Labinfo y otra en GitHub. Este repositorio se encuentra dividido en dos ramas, siendo la rama labinfo una rama adicional solo funcional dentro del laboratorio, que contiene scripts específicamente diseñados para el laboratorio.

La rama master cumple la función de ser el repositorio público que cualquier persona que busca scripts IPv6 puede usar, disponible en: https://github.com/danielospina-b/Labinfo-ECI-IPv6

Si es necesario modificar algún script del repositorio, se recomienda trabajar de la siguiente forma:

1. Si la modificación pertenece únicamente a la rama labinfo (es decir que contiene características solo aplicables al laboratorio), el cambio se realiza directamente a esta rama.
2. Si la modificación es general para configuraciones IPv6 (cambios necesarios por actualizaciones de Windows o Linux) y puede pertenecer a la rama master, se realiza el cambio a la rama master y posteriormente el mismo cambio a la rama labinfo.

   Para esto se puede seguir el siguiente procedimiento:

   a. Se hacen los cambios en la rama master y se hace commit (ejemplo: se modifica DHCPv6-DUID/duid-script.sh)
   b. Se hace checkout a la rama labinfo
   c. Se traen los archivos modificados desde la rama master utilizando el comando:

      git checkout master DHCPv6-DUID/duid-script.sh

   d. Se realiza commit. Ahora ambas ramas tienen el mismo cambio.

Para acceder al repositorio interno de labinfo, contactar a la administración del laboratorio de informática. El administrador debe agregar su usuario de LabinfoGitlab[3] como colaborador del repositorio de IPv6.

El repositorio público https://github.com/danielospina-b/Labinfo-ECI-IPv6 es una copia del repositorio interno de labinfo, por lo tanto cambios realizados al interno sería ideal moverlos también a este repositorio; para esto se puede realizar un Pull Request, o escribir a: daniel.ospina-b@mail.escuelaing.edu.co

[3] GitLab es la plataforma instalada en el laboratorio.

TRABAJO FUTURO

Se propone como trabajo futuro para este proyecto los siguientes aspectos, se trata a profundidad cada uno de estos a través del libro de proyecto:

- Objetivos del proyecto no alcanzados como VPN y Monitoreo de Servidores.
- Revisión de Software no compatible con IPv6 como Maven y Bizagi.
- IPv6 sobre red inalámbrica en los laboratorios del edificio H.
- Servidor IPv6 como servicio dual para NAT (IPv4) y NAT64 (IPv6).

REFERENCIAS

[1] J. Salinas, C. Sánchez, J. Herrera y C. Santiago, «Propuesta de implementación de IPv6 en una infraestructura IPv4 en operación,» 24 July 2019.

[2] S. Thomson, Cisco, T. Narten, IBM, T. Jinmei y Toshiba, «[RFC4862] IPv6 Stateless Address Autoconfiguration,» September 2007.

[3] G. Huston, Telstra, A. Lord, APNIC, P. Smith y Cisco, «[RFC3849] IPv6 Address Prefix Reserved for Documentation,» July 2004.

[4] R. Hinden, Nokia, B. Haberman y JHU-APL, «[RFC4193] Unique Local IPv6 Unicast Addresses,» October 2005.

[5] O. E. Johansson, «DHCPv6 – an introduction to the new host configuration protocol,» 16 December 2011. [En línea]. Available: https://ipv6friday.org/blog/2011/12/dhcpv6/. [Último acceso: 2 August 2019].

[6] J. Bound, B. Volz, T. Lemon, C. E. Perkins y M. Carney, «[RFC3315] Dynamic Host Configuration Protocol for IPv6 (DHCPv6),» July 2003.

[7] E. Rey, «Reliable & Secure DHCPv6,» [En línea]. Available: https://www.ernw.de/download/ERNW_Troopers15_IPv6SecSummit_DHCPv6_clean.pdf. [Último acceso: 2 August 2019].

[8] T. Narten, IBM, E. Nordmark, Sun MicroSystems, W. Simpson, Daydreamer, H. Soliman y Elevate Technologies, «[RFC4861] Neighbor Discovery for IP version 6 (IPv6),» September 2007.

[9] M. Bagnulo, UC3M, P. Matthews, Alcatel-Lucent, I. van Beijnum y IMDEA Networks, «[RFC6146] Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers,» April 2011.

[10] C. Bao, CERNET Center/Tsinghua University, C. Huitema, Microsoft Corporation, M. Bagnulo, UC3M, M. Boucadair, France Telecom y X. Li, «IPv6 Addressing of IPv4/IPv6 Translators,» October 2010.

[11] A. K. Tsetse, A. L. Wijesinha, R. K. Karne y A. Loukili, «Measuring the IPv4-IPV6 IVI Translation Overhead,» October 2012.

[12] A. Shiranzaei y R. Zaman Khan, «A Comparative Study on IPv4 and IPv6,» *International Journal of Advanced Information Science and Technology*, vol. 33, nº 33, pp. 13-15, 2015.

[13] X. Ling, C. Bao, M. Chen, H. Zhang y J. Wu, «The CERNET IVI Translation Design and Deployment for the IPv4/IPv6 Coexistence and Transition. RFC6219,» p. 10, May 2011.

[14] M. Mawatari, Japan Internet Exchange, M. Kawashima, NEC AccessTechnica, Ltd., C. Byrne y T-Mobile USA, «[RFC6877] 464XLAT: Combination of Stateful and Stateless Translation,» April 2013.

[15] Microsoft Research, «[RFC3484] Default Address Selection for Internet Protocol version 6 (IPv6),» February 2003.

[16] D. Thaler, Microsoft, R. Draves, Microsoft Research, A. Matsumoto, T. Chown, NTT y University of Southhampton, «[RFC6724] Default Address Selection for Internet Protocol Version 6 (IPv6),» September 2012.

[17] A. Matsumoto, J.-y. Kato y T. Fujisaki, «Update to RFC 3484 Default Address Selection for IPv6,» 2011.

[18] Tecnológico de Monterrey, «Introduction to Jool,» [En línea]. Available: https://www.jool.mx/en/intro-jool.html. [Último acceso: 2019 December 01].

[19] toreanderson, «Need to enable firewall-like features on the NAT64,» 9 March 2015. [En línea]. Available: https://github.com/NICMx/Jool/issues/41. [Último acceso: 1 December 2019].

[20] K. Auer, «Controlling IPv6 source address selection,» 22 August 2012. [En línea]. Available: http://biplane.com.au/blog/?p=30. [Último acceso: 29 September 2019].

[21] IETF Network Working Group, «IPv4 Declared Historic,» 15 September 2016.

[22] IEEE-USA, «Next Generation Internet: IPv4 Address Exhaustion, Mitigation Strategies and Implications for the U.S.,» 2009.

[23] A. Jayasekara, N. Wickamasinghe, W. Wijetunge y W. Kumara, «IPv6: TRANSITION METHODS AND ADVANTAGES,» 2012.

[24] J. Zorz, «Introducing the NAT64 checker,» 23 August 2017. [En línea]. Available: https://blog.apnic.net/2017/08/23/introducing-nat64-checker/. [Último acceso: 09 August 2019].