UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

Colegio de Ciencia y Tecnología

Análisis de la gestión de la red Pacific Wave de EEUU bajo IPv6

T E S I S

Que para obtener el título de:

Licenciado en Ingeniería en Sistemas Electrónicos y de Telecomunicaciones

P r e s e n t a :

Iván Varela Sánchez

D i r e c t o r :

M. en C. José Ignacio Castillo Velázquez

C o d i r e c t o r:

M. en C. Joel Yazbek Buendía Gómez

Ciudad de México, junio de 2021.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

Agradecimientos

Inicialmente a aquello que siempre me ha cuidado y guiado a lo largo de mi vida…

Agradezco a mi **madre** Lilia, quien me enseñó a enfocar el carácter para siempre salir adelante.

Agradezco a mi **padre** Antonio, quien me enseñó a tener templanza para las situaciones venideras.

Agradezco a mi pequeña **hermana** Paola, quien me enseñó a amar y ser un buen ejemplo.

A mis **tíos,** que cada uno me inculcó una fortaleza diferente para ser un mejor hombre.

A mi **tía** Martha, que hizo ver la vida de manera alegre y energética.

A mi **tía** Marisela, que me brindo fortaleza espiritual.

A mis **primos** Meli, Mario y Edgar, mis hermanos de infancia con quienes siempre he compartido risas e historias.

A los que se adelantaron y que hoy en día ya sólo están en mí corazón.

A la **UACM**, quien abrió sus puertas para recibirme y ser parte de ella.

A todos los profesores de la UACM, por compartir su conocimiento y sabiduría, en especial a **M. en C. José Ignacio Castillo Velázquez** quien me enseño nuevos horizontes en la ingeniería y sus fascinantes lados, a **M. en C. Joel Yazbek Buendía Gómez** quien me enseñó a fortalecerme en la investigación y canalizar mis estudios.

A mis lectores de tesis **Dr. Adolfo Horacio Escalona Buendía**, **Ing. Ricardo Galindo Reyes** y al **Ing. José Miguel Vargas Pliego** por tomarse el tiempo y dedicación para apoyarme en este último salto en la licenciatura.

A mis camaradas de la universidad, que hicieron mi estadía y mis días, divertidos, interesantes y con grandes anécdotas.

He sido afortunado al encontrar a la gente adecuada en el momento y lugar adecuado…

A **Alberto Ledesma** y su hermosa familia, quienes me enseñaron a dar un *plus* en el trabajo y que hoy en día lo llevo siempre presente.

A mi segunda familia en **TP**, quienes siempre me han brindado su apoyo y han estado en mi vida profesional.

"Dream big and dare to fail.
I dare you to do that, because
this is living proof
that it is possible... to make
a DREAM come true"

James Alan Hetfield, Ceremonia de inducción al Salón de la Fama del Rock & Roll 2009

Contenido

Resumen

La Internet nace derivado de la necesidad del uso de información descentralizada en una red de computadoras, enfocada hacia la milicia, gobiernos e instituciones de investigación y académicas. A esta primera red se le conoció como ARPANET, la cual inicio con 4 computadoras, las que conectaron el Instituto de Investigaciones de Stanford (SRI), la Universidad de Utah, la Universidad de California en Los Ángeles (UCLA) y la Universidad de California en Santa Bárbara (UCSB).

En sus inicios Internet no tenía en cuenta la magnitud y el impacto que tendría a nivel global por lo que el protocolo IPv4 encargado de darle un identificador a cada dispositivo para navegar por la red comenzó a agotarse dada la expansión de usuarios y sus demandas de estar el mayor tiempo posible dentro de la red. Es por ello que se vio la necesidad de un nuevo protocolo capaz de satisfacer dichas demandas. El protocolo IPv6 es la respuesta a estas necesidades, este permite una expansión del uso de Internet y mejoramiento de funcionalidades de su antecesor. La evolución de IPv4 a IPv6 trajo nuevas características para los protocolos de enrutamiento como OSPFv3 que es una nueva versión para IPv6 y BGP-4 que únicamente genero una extensión para poder utilizar IPv6.

En 1995 se libera Internet comercial ya como una red operacional para el mundo, antes de ello fue ARPANET y su evolución NSFNET. Con una Internet comercial se requería de un espacio dedicado para realizar investigaciones y continuar innovando, es por ello que surgen las redes avanzadas, redes que están en un área distinta a la internet comercial que permiten a científicos, investigadores, académicos, profesores y estudiantes colaborar, compartiendo información y herramientas mediante una serie de interconexiones de redes, ejemplo de estas redes son, GÉANT (Europa), Internet2 (Estados Unidos), CANARIE (Canadá), TEIN*CC (Asia), WACREN (África del Oeste y Central), UbuntuNet Alliance (África del Este y Sur), ASREN (Estados Árabes), entre otras.

La red avanzada de Pacific Wave es una red distribuida que funge como punto de intercambio de Internet, enfocada en la investigación y la educación. Proporcionando conectividad a Internet de con un *high-performance* entre las instituciones de investigación y desarrollo de ciencia e ingeniería de EE. UU. y sus socios internacionales, y es una infraestructura crítica para el acceso a instrumentos respaldados internacionalmente, fuentes y repositorios de datos a gran escala.

PacWave permite flujos de trabajo científicos a gran escala para acelerar el descubrimiento en todas las áreas de la ciencia y la ingeniería, incluida la física de *high-energy*, ciencias de la tierra, astronomía y astrofísica, biología e ingeniería biomédica, así como visualización escalable, realidad virtual, aprendizaje automático e inteligencia artificial.

Prólogo

En la UACM, los primeros egresados titulados de ISET se lograron en 2012 en el campus IZT y en 2013 en SLT. El ADVNETLAB (*Advanced Networking Laboratory*) en la UACM fue fundado en 2013 en SLT, con recursos personales, una vez que hubo masa crítica de egresados de ISET e interés sobre el tema de redes avanzadas. Con base en mi experiencia en universidades y empresas desarrollé la metodología ADVNETLAB con la que se dirigen las tesis y otros proyectos.

Desde 2015 a la fecha se han titulado 14 estudiantes de licenciatura bajo la metodología ADVNETLAB, hemos producido 13 tesis con 14 estudiantes de telecomunicaciones (ANL-1 al ANL-14), 11 de la UACM México y 3 de la UNAS Perú. Desde ADVNETLAB se han publicado 24 artículos indexados en SCOPUS tanto en redes avanzadas, seguridad informática, software y educación; 15 de ellos publicados con los ahora ingenieros. También se desarrolló UTILCON, un sistema de gestión de congresos o seminarios u otro tipo de eventos académicos, registrado ante el Instituto Nacional de Derechos de Autor, ya que en México los sistemas de software no son patentables como sí lo son en otros países.

En esta ocasión se presenta para junio de 2021 Ivan Varela Sánchez (ANL15) con el trabajo correspondiente al estudio vía emulación del *Backbone* de la red avanzada Pacific Wave bajo IPv6 en su topología más actualizada, para el cual se ponen a prueba su conectividad y gestión; tal y como sucede en los centros de operaciones de red de las compañías proveedoras de internet. En trabajos anteriores en ADVNETLAB se han abordado desde 2013 a 2018 la conectividad y gestión para las redes avanzadas CUDI, CLARA, Internet2, CANARIE, REUNA, GEANT, AFRICACONNECT bajo protocolos IPv4 e IPv6. El presente trabajo inició en noviembre de 2019 y culminó en noviembre de 2020, bajo las condiciones de la pandemia de COVID-19, la cual obliga a presentar examen profesional a distancia. Mis felicitaciones a Iván Varela Sánchez por el trabajo de tesis concluido.

M. en C. José Ignacio Castillo Velázquez

Director de tesis - Junio de 2021

Capítulo 1

Introducción

Hoy en día las telecomunicaciones han sido un pilar de la sociedad, especialmente el uso de la red de redes denominada Internet, dado que, a través de ésta, todo tipo de información es enviada y recibida segundo a segundo, provocando que su uso se vuelva indispensable para el día a día de escuelas, centros de investigación, empresas y gobiernos. Generando una creciente demanda en su uso, demanda que con el actual protocolo de Internet denominado IPv4 (*Internet Protocol version 4*) encargado de la transmisión y recepción de la información que viaja a través de la Internet, mediante el uso de direcciones IP (*Internet Protocol*) para que las computadoras se puedan comunicar a través de la red, no está logrando abastecer de direcciones. Es por ello que el Grupo de Trabajo de Ingeniería de Internet (IETF - *Internet Engineering Task Force*) se vio en la necesidad de una evolución, a la cual denominaron **IPv6** (*IP version 6*) que dentro de sus características está el proveer de direcciones IP a la futura Internet. Sobre esta base se realiza una revisión de la evolución de Internet para un mayor entendimiento del paso de IPv4 a IPv6, a su vez la aparición de Redes Avanzadas, siendo que estas han sido pilar y precursoras para el desarrollo de IPv6.

Para fines prácticos se utilizará el concepto de red como un sistema que proporciona un servicio de transferencia de datos entre computadoras, en relación con lo anterior Internet es una colección de redes interconectadas por *switches* y *routers* [[1], [2]].

1.1 Evolución de Internet

Joseph Carl Robnett Licklider dio una nueva visión de una red mundial de computadoras en su *paper* de marzo de **1960**, *Man-Computer Symbiosis*, es por ello que es considerado uno de los pioneros de la Internet [3].

> "Lo que esperamos es que en no muchos años, los cerebros humanos y las computadoras estén estrechamente acoplados, y que la asociación resultante pensará como nunca un cerebro humano ha pensado y procesará datos de una manera nunca vista por las máquinas de gestión de información que conocemos hoy en día"
>
> *March 1960, J. C. R. Licklider*

En **1961** colaboradores del MIT propusieron una investigación temprana sobre la teoría de conmutación de paquetes para darle un uso adicional a las computadoras para investigaciones científicas y académicas. En **1969** la Red de Agencia de Proyectos de Investigación Avanzada

(ARPANET - *Advanced Research Projects Agency Network*), logró la primera Red de Área Amplia (WAN - *Wide Area Network*), conectando el Instituto de Investigaciones de Stanford (SRI), la Universidad de Utah, la Universidad de California en Los Ángeles (UCLA) y la Universidad de California en Santa Bárbara (UCSB) [[4],[5]].

Ray Tomlinson sobre la red ARPANET presentó en **1972** el *E-mail* (*Electronic-mail*), siendo este uno de los primeros servicios que ofrecía generar una red de computadoras. En **1973** se inician trabajos para el desarrollo del protocolo TCP mediante la colaboración de la Agencia de Proyectos de Investigación Avanzada de Defensa (DARPA - *Defense Advanced Research Projects Agency*) y el SRI. Para **1974** Vinton Cerf y Robert Kahn, redactaron un *paper* llamado *A Protocol for Packet Nertwork Internetworking*, el cual indicó como podría solucionarse el problema de comunicación entre los diferentes tipos de equipos de cómputo, desarrollando el sistema de direccionamiento IP. Esto luego se conoció como IPv4, el cual contaba con aproximadamente 4.3 mil millones de direcciones IPv4. Esta idea es aplicada en **1978** y se le denominó *Transmition Control Protocol/Internet Protocol* (TCP/IP). Siendo para **1979** cuando ISO publicó el modelo de referencia de OSI como guía [[6]-[8]].

En **1980** IPv4 es utilizado por primera vez por John Postel en una red de prueba. En **1981** se publicó el RFC (*Request for Comments*) 791 siendo este el protocolo de internet versión 4. En noviembre de **1983** con el RFC 801 se presentó el plan para realizar el cambio de la transición NCP a TCP/IP. En **1984** ISO liberó el ISO/OSI 7498 como modelo de referencia básico para sistemas abiertos de redes de datos el cual todavía no constituía un estándar. En octubre de **1986** se aprobó una propuesta formal para crear la Red de Ciencias de la Energía (ESnet - *Energy Sciences Network*), la responsabilidad de operar esta red fue asignada al Centro Nacional de Computación de Fusión de Energía Magnética (NMEFCC - *National Magnetic Energy Fusion Computer Center*), con Jim Leighton como director de ESnet. En **1989** Tim Berners-Lee, del CERN (*Conseil Européen pour la Recherche Nucléaire*) en Ginebra (Suiza), desarrolló *World Wide Web* o solamente Web, que es un sistema lógico de distribución de la información basado en hipermedios enlazados o hipertexto y con acceso a través de Internet, a diferencia de otros protocolos, la web accede a través de una interfaz gráfica de usuario (GUI - *Graphical User Interface*) [[9]-[11]].

En **1992** El IETF concluyó que IPv4 no lograría abastecer de direcciones IP a la creciente Internet [12]. Ese mismo año se fundó *Internet Society* (ISOC) constituida como la única organización

dedicada exclusivamente al desarrollo mundial de Internet [13]. CANARIE se creó en **1993,** inicialmente se centró en el desarrollo de una red que proporciona conectividad interprovincial e internacional para la Red Nacional de Investigación y Educación de Canadá [14]. A finales de ese mismo año en diciembre se creó un grupo de trabajo para la llamada Próxima Generación de IP (IPng - *IP next generation*), para seleccionar una nueva versión. *IP: Next Generation (IPng) White Paper Solicitation* fue la solicitud formal de propuestas de este grupo [15]. En julio de **1994**, los directores de IPng votaron a favor de aceptar una versión modificada del Protocolo Simple de Internet (SIP - *Simple Internet Protocol*), como base para IPv6.

En **1995** el Grupo de Trabajo de Redes y Telecomunicaciones (NTTF - *Networking and Telecommunications Task Force*) formado por el Consejo Interuniversitario de Comunicaciones (*EDUCOM: Interuniversity Communications Council*), que proporciona orientación para el uso de la red en la educación superior. A través de una serie de reuniones, talleres y grupos de trabajo, formó el núcleo de lo que se convertiría hoy en día en Internet2. A finales de **1995** en diciembre apareció la primera especificación de IPv6 con el RFC 1883 [16].

Internet2 fue presentado por treinta y cuatro líderes universitarios reunidos en el Chicago O'Hare Hilton el 1 de octubre de **1996** y los cuales se comprometieron a establecer un proyecto para fomentar el desarrollo de capacidades de redes que no sólo promuevan la investigación y la educación, sino que también se abran camino en la Internet comercial global [17].

Durante 1998 IPv6 se presentó como el nuevo esquema de direccionamiento con el RFC 2460 [18]. Aumentando masivamente la cantidad de direcciones IP disponibles con alrededor de $3.4x10^{38}$ de direcciones IPv6. También el 6BONE, lanzó la primera red mundial de prueba IPv6.

En el año **2002** la red panaeuropea GÉANT y sus semejantes nacionales en América Latina comenzaron a analizar la posibilidad de una interconexión directa, a lo que se conoció como CAESAR (*Connecting All European and South (Latin) American Researchers*). En consecuencia, el 3 de junio de **2003** se inició el proyecto ALICE (América Latina Interconectada con Europa), siendo esto demostración de que el desarrollo de CAESAR había rendido frutos. ALICE tenía como principal objetivo crear una infraestructura de redes de investigación en América Latina e interconectarla con GÉANT. El 10 de junio, apenas siete días después de iniciarse oficialmente el proyecto ALICE se creó oficialmente la Red CLARA (Cooperación Latino Americana de Redes Avanzadas) [19].

En enero de **2004**, el CENIC (*Corporation for Education Network Initiatives in California*) y PNWGP (*Pacific Northwest Gigapop*) anunciaron conjuntamente el despliegue de una instalación de interconexión distribuida geográficamente llamada **Pacific Wave** [20]**.**

El conjunto de direcciones IPv4 de ICANN (*Internet Corporation for Assigned Names and Numbers*) se agotó el 3 febrero de **2011**, entregando el último bloque de direcciones disponibles [21].

El 8 de junio del 2011, se celebró el Día Mundial de IPv6 (*World IPv6 Day*), un evento patrocinado y organizado por la *Internet Society* y en el que empresas como *Google*, *Facebook*, *Yahoo* y *Akamai* ofrecieron sus servicios utilizando IPv6, siendo está una primera prueba mundial a gran escala. A lo que al siguiente año el 6 de junio de 2012, se presentó el Lanzamiento Mundial de IPv6 (*World IPv6 Launch*). Desplegando IPv6 de forma permanente en los servicios de Internet [22].

1.2 Redes Avanzadas

Con la invención del telégrafo, el teléfono, la radio y las computadoras; se sentaron las bases para la Internet como una herramienta de uso a nivel mundial, un mecanismo para esparcir información y un medio para la colaboración e interacción entre personas y computadoras, sin tener en cuenta su ubicación geográfica. Desde las primeras investigaciones en conmutación de paquetes, los gobiernos, la industria y la academia se han asociado como autores de la evolución e implementación de esta herramienta. Por otro lado, las Redes Avanzadas (RAs) o Redes Nacionales de Investigación y Educación (NREN - *National Research and Education Network*) fueron originalmente un producto de la investigación académica para encontrar formas eficientes y rentables de compartir recursos informáticos, para comunicarse y colaborar entre investigadores y académicos. A principio de los setenta EE.UU. estuvo a la cabeza de esta investigación y estableció las primeras "Redes a Nivel Nacional" al servicio de la comunidad académica, de investigación y militar. Algunas de las primeras RAs formales incluyen UNINET en Noruega establecida en 1976, *Computer Science Network* (CSNET) en 1981 y más tarde NSFNet en 1985 en los EE. UU., *Joint Academic NETwork* (JANET) en el Reino Unido en 1984, *Swiss Education and Research Red* (SWITCH) en 1987, NORDUNet para los países

nórdicos en 1988 y CA*net en Canadá en 1990 que más tarde se conocería como la RA CANARIE [[23]-[28]].

Se puede decir que estos primeros esfuerzos de redes de investigación han cambiado el mundo; dieron a luz a Internet. En algunos casos, estas primeras redes RAs también desarrollaron la primera red *Backbone* (Red principal o troncal) de Internet nacional de sus países. Por ejemplo, en Canadá, CANARIE fue durante muchos años la única red *Backbone* de Internet en el país, y en Australia, la Red Nacional Académica y de Investigación (AARNet) estableció y administró la infraestructura troncal nacional temprana [[29], [30]].

Una Red Avanzada, es una red de área amplia (WAN - *Wide Area Network*) así como una *segunda internet* es decir forman un área distinta de la Internet comercial, un área que coexiste en un espacio paralelo reservado para comunidades de educación e investigación especializada en servicios que están dedicados a satisfacer las necesidades de las comunidades de investigación y educación dentro de un país. Generalmente se distingue por el soporte para una red *Backbone* de alta velocidad, que a menudo ofrece canales dedicados para proyectos de investigación individuales. Utilizando las redes avanzadas, estudiantes, profesores, académicos, investigadores y científicos pueden colaborar, compartiendo información y herramientas mediante una serie de interconexiones de redes a través de distintos países y continentes, sin importar las distancias ni las fronteras [31].

Las RAs suelen ser los lugares donde se desarrollan nuevos protocolos y arquitecturas de Internet antes de la implementación dentro de la Internet pública. Dos ejemplos de estos protocolos son IPv6 e *IP multicast*, y dos ejemplos de arquitectura son *client/server* y *Cloud computing*. En los últimos años, las RAs también han desarrollado muchos servicios. Las federaciones nacionales de identidad, muchas de las cuales están representadas en los REFED (*Research and Education FEDerations group*), son un ejemplo de tales servicios [32]. Existen varias RAs conectadas entre sí alrededor del mundo, permitiendo un alcance global entre científicos y académicos.

Las RAs tienen la finalidad principal de:

- Proveer una infraestructura de comunicación de datos de gran capacidad para apoyar el trabajo de estudiantes, profesores, académicos, investigadores y científicos, permitiendo la transferencia de grandes cantidades de datos a gran velocidad.

- Proveer una plataforma donde innovadores e investigadores puedan desarrollar y probar nuevos servicios y tecnologías de red, sirviendo como una poderosa herramienta de investigación.

Características de las RAs

- Redes de alto ancho de banda
  - La investigación generalmente necesita redes no congestionadas
- Baja latencia
  - Conexiones de Fibra Óptica
- Son redes que usan *routers* de tipo CORE

Las redes avanzadas son de alta velocidad (contando con anchos de banda de 500 Mbps, 1 Gbps, 10 Gbps, 100 Gbps, etc.) y no están congestionadas, tienen baja latencia y, por lo general, funcionan como redes abiertas sin filtrado.

A continuación, se describe la RA de Pacific Wave que se utilizara en este trabajo.

1.3 La red avanzada: Pacific Wave

Pacific Wave (**PW - PacWave**) es una instalación de red avanzada diseñada para proporcionar intercambio de Internet (IX - *Internet Exchange*) y enfocada en redes para la investigación y educación (R&E - *Research and Education*) en Estados Unidos y conectando con Tokio, Japón; con acceso a intercambio e interconexión de vanguardia, *Science DMZ* (un área reservada en la red de un campus), Todo Definido por Software (SDX - *Software Defined Everything*) y capacidades de Redes Definidas por Software (SDN - *Software Defined Networking*). PacWave permite a científicos trabajar en todas las áreas de la ciencia y la ingeniería, incluida la física de altas energías, ciencias de la tierra, astronomía y astrofísica, biología e ingeniería biomédica, realidad virtual, *machine learning* e inteligencia artificial [33]. PW incluye los siguientes servicios en su red de fibra óptica:

- Una red *peering* y de intercambio; distribuida y totalmente abierta para los usuarios de esta red con puntos de acceso en una red *Backbone* de 100 Gbps que se extiende por Seattle, Sunnyvale y Los Ángeles, al que se conectan casi todas las redes de *Pacific Rim R&E* y que a su vez está interconectado con todas las redes principales de R&E de

EE.UU., Incluyendo Internet2 y ESnet (cada uno con múltiples conexiones de 100 G), así como los principales proveedores de la nube e ISPs internacionales.

- Una plataforma *wide-area Research DMZ* con una red *Backbone* dedicada de 100 G entre Los Ángeles, Sunnyvale y Seattle y también acceso en Tokio (en WIDE/T-REX y Tata pops), Denver, Albuquerque, El Paso y Chicago (en StarLight) a través de un ancho de banda de 100 G.
- Un banco de pruebas SDN/SDX dedicado y paralelo con puntos de acceso en Seattle, Sunnyvale y Los Ángeles, que permite esfuerzos de colaboración con StarLight, WIDE/T-REX y otros para explorar la interoperabilidad regional e internacional de la red de próxima generación y las capacidades de intercambio.

PW es un proyecto conjunto de la Corporación para Iniciativas de Redes Educativas en California (**CENIC** - *Corporation for Education Network Initiatives in California*) y el Gigapop del Noroeste del Pacífico (**PNWGP** - *Pacific Northwest Gigapop*), y se opera en colaboración con la Universidad del Sur de California y la Universidad de Washington. Es la interconexión oficial financiada por la Fundación Nacional de Ciencias (**NSF** - *National Science Foundation*) de EE. UU., de instalación de pares e intercambio SDX para redes de la costa del Pacífico [33].

1.3.1 Historia de la red de Pacific Wave

Basándose en la operación exitosa de los intercambios públicos de Internet creados por la Universidad de Washington en Seattle y la Universidad del Sur de California en Los Ángeles en 1996, CENIC y PNWGP anunciaron conjuntamente el despliegue de una instalación de distribución llamada Pacific Wave en enero de 2004. Con ubicaciones en Los Ángeles, Sunnyvale y Seattle, PW formó el primer Intercambio de Internet (Internet Exchange) distribuido destinado a mejorar el acceso rentable para RAs en todo el Pacífico [34].

Gracias en parte a dos premios independientes de cinco años de la NSF, la International Research Network Connections (IRNC), PW se ha expandido para incluir 30 conexiones de red, apoyando a 29 países a partir de 2015. Trabajando con colegas en StarLight Exchange, servicios se han extendido y expandido a Chicago, y extensiones similares han conectado la Universidad de Hawái con Australia y Nueva Zelanda.

Al cierre del proyecto Translight/Pacific Wave de 10 años, PW admite interconexiones de 100 Gbps y la creciente necesidad de conexiones de gran ancho de banda para apoyar a científicos e investigadores [34].

1.3.2 Sitios de Nodos y Topología

En todos los nodos de PW, los participantes pueden conectar su dispositivo configurado de Capa 3 a un conmutador PW a 10 Gbps o 100 Gbps Ethernet. Otros servicios pueden estar disponibles bajo petición [35]. En la Tabla 1 se muestran los nodos activos de PW a 2020.

Sitios de nodos	
Albuquerque, New Mexico 505 Marquette Ave NW, Albuquerque, NM 87102-3157, NPA-NXX: 505-828	***Los Angeles, California*** 1 Wilshire (624 S. Grand), Los Angeles NPA-NXX: 213 624
Bay Area, California 1380 Kifer Road, Sunnyvale, CA 94086 NPA-NXX: 408-212	***818 W 7th, Los Angeles*** NPA-NXX: 213 228
529 Bryant Street, Palo Alto, CA 94301 NPA-NXX: 650-213	***600 W 7th, Los Angeles*** NPA-NXX: 213 270
Denver, Colorado 1850 Pearl St. Denver, CO 80203 NPA-NXX: 303-318, CLLI: DNVTCO56	***Seattle, Washington*** Westin Building, 2001 Sixth Ave, Seattle, WA NPA-NXX: 206-443
El Paso, Texas 501 W Overland Ave, El Paso, TX 79901 NPA-NXX: 915-532	***Tokyo, Japan*** 3-8-21 Higashi Shinagawa, Shinagawa-ku, Tokyo 140-0002, Japan

Tabla 1. Nodos de Pacific Wave activos a 2020 [35]

En la Figura 1 se muestra la topología de *Backbone* a octubre de 2019 de PW, con sus participantes y afiliados, así como con AP-REX (*Atlantic Pacific Research and Education Exchange*) el cual es un proyecto de PW e Internet2, PW y AP-REX mantienen con la Red Regional Occidental (WRN - *Western Regional Network*) la cual es una asociación de varios estados para proporcionar sólidas RAs de alta velocidad para usos relacionados a la investigación y educación. WRN es una colaboración del PNWGP en Washington, el *Gigapop de Front Range* (FRGP) en Colorado y Wyoming, la Universidad de Nuevo México con el

GigaPoP Albuquerque (ABQG - *Albuquerque GigaPoP*), las Iniciativas de la Red de la Corporación para la Educación en California (CENIC) y la Universidad de Hawái (UH - *University of Hawaii*) [37]. También se muestran los puntos de la Plataforma de Investigación del Pacífico (PRP - *Pacific Research Platform*) la cual está unida a PW. El PRP, dirigido por investigadores de UC San Diego y UC Berkeley, permite transferencias de datos rápidas y seguras entre las instituciones participantes, que incluyen los 10 campus de UC, así como universidades e instalaciones de investigación seleccionadas en toda la región. El proyecto utiliza PW y la Red de Investigación y Educación de California de CENIC (CalREN) para crear una plataforma de investigación amplia y fluida que fomenta la colaboración a nivel estatal, regional e incluso mundial. La Figura 2 muestra una topología reordenada para facilitar el entendimiento y el trabajo de emulación que se realizará en el capítulo correspondiente a la metodología.

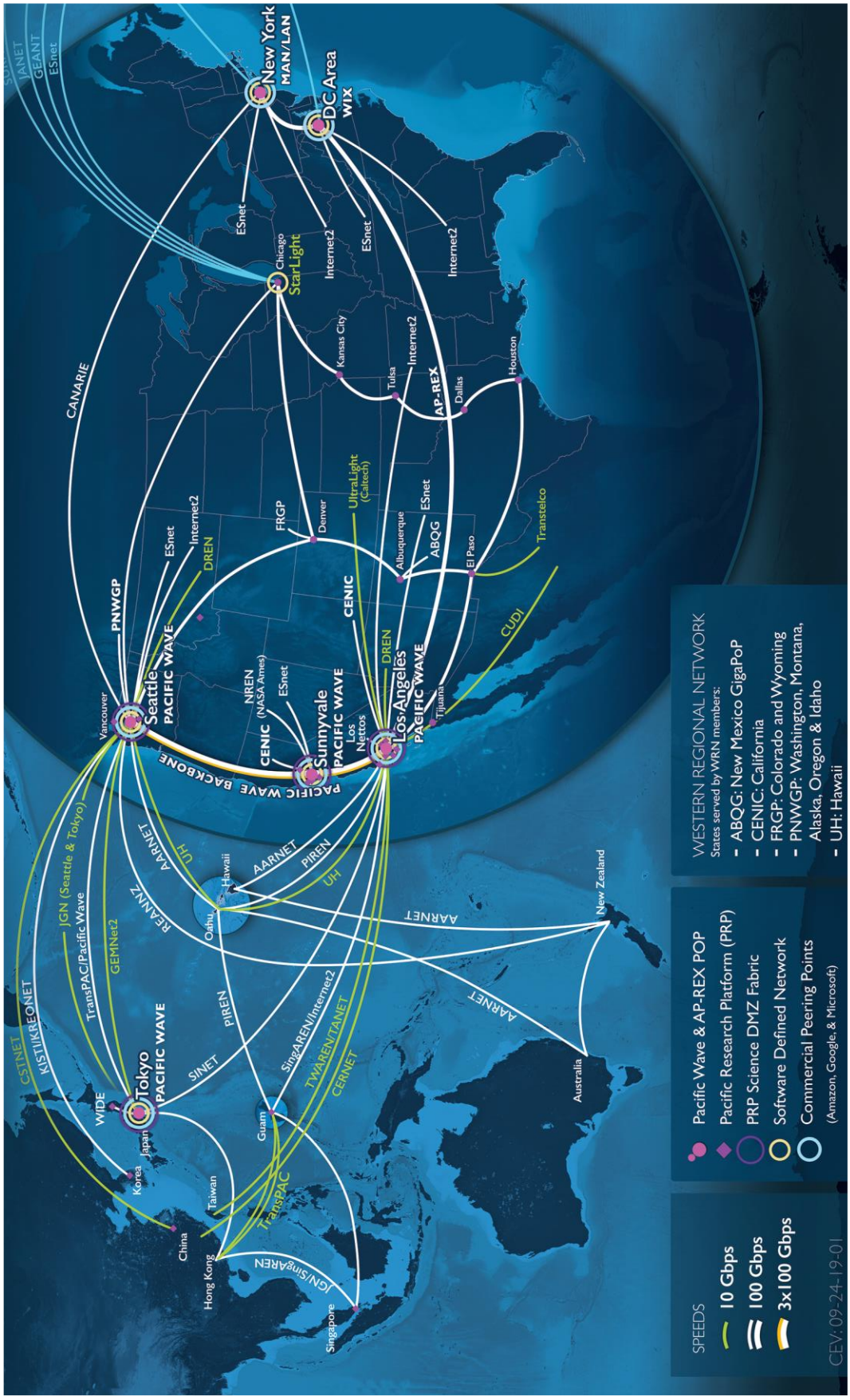


Figura 1 Topología de Backbone de Pacific Wave a agosto 2020. Tomada de ref. [36]

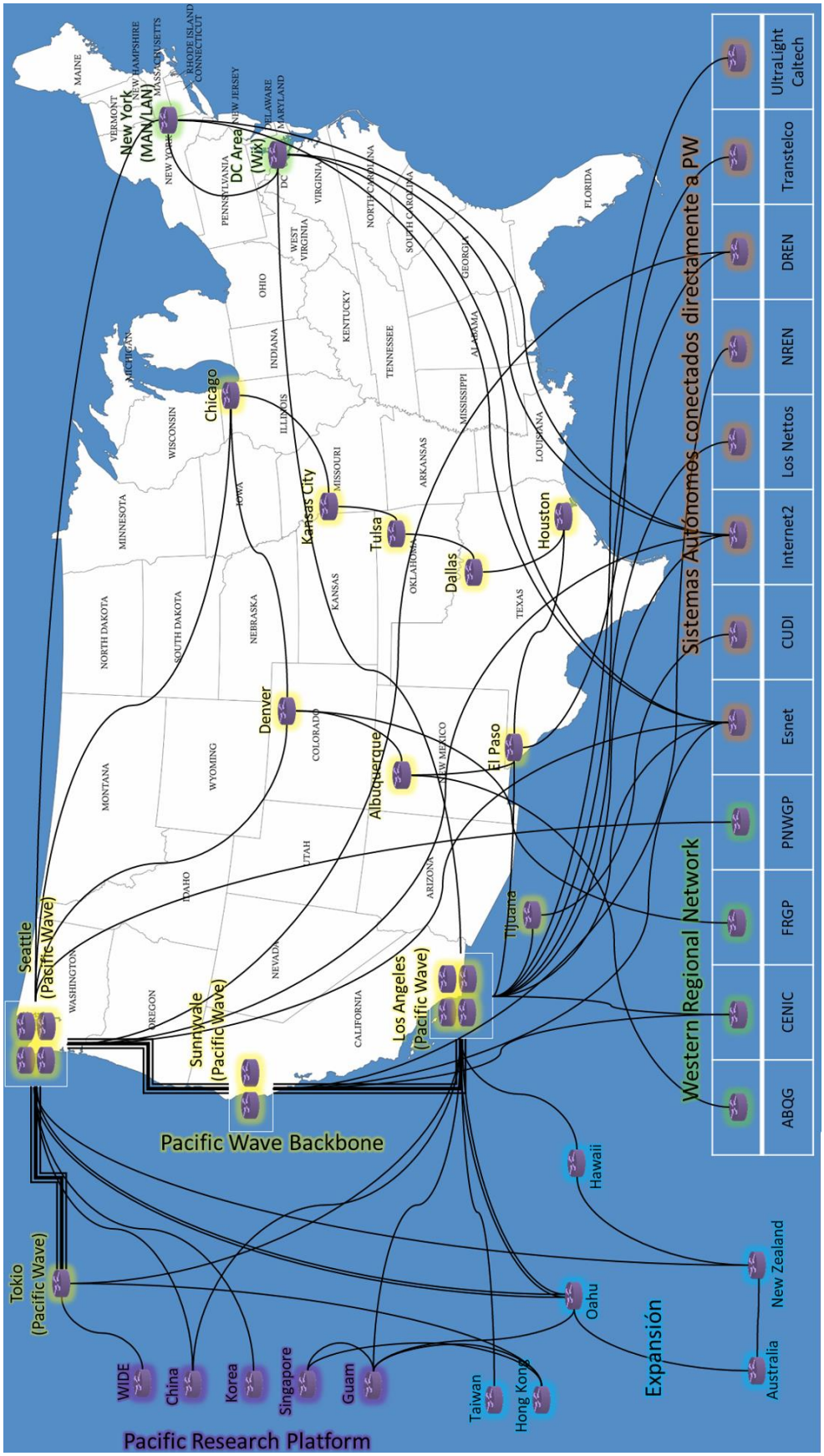


Figura 2 Topología de Backbone de Pacific Wave [Imagen propia con base en ref. [36]]

1.4 Justificación

Las RAs son una *segunda internet* dedicadas a la investigación y educación, siendo redes en las que se introducen nuevos protocolos y arquitecturas de Internet antes de la implementación dentro de la Internet pública. IPv6 es un ejemplo de aquello que fue desarrollado dentro de las RAs. Por tanto, se realiza una exploración de la red: **Pacific Wave**. El estudio de las RAs es de interés para el Laboratorio de Redes Avanzadas (AdvNetLab - *Advanced Networking Laboratory*) en la Universidad Autónoma de la Ciudad de México con sede en San Lorenzo Tezonco, tiene como parte de su núcleo el estudio de las RAs, en él se han generado trabajos previos y relacionados con CUDI, CLARA, CANARIE, GÉANT, AMERONET (integración de Canarie, Internet2 Layer2 y RedCLara), REUNA y AFRICACONNECT como tesis y artículos arbitrados [[38]-[59]].

1.5 Objetivos

Objetivo General

Estudiar la arquitectura Backbone de la red avanzada Pacific Wave mediante su emulación empleando IPv6.

Objetivos Específicos

- Realizar una revisión de la evolución de la topología de la RA: PW.
- Realizar las configuraciones necesarias para la emulación de la conectividad y gestión de la RA: PW, bajo IPv6.
- Emular el funcionamiento de conectividad y gestión de la RA: PW, con IPv6.
- Corroborar los alcances y limitaciones del emulador GNS3.
- Desarrollar las tareas de un administrador de redes en la RA indicada: monitorear, configurar, actualizar y resolver problemas.

1.6 Organización de Tesis

El contenido de la tesis se encuentra dividido en cuatro capítulos:

Capítulo 1, se hace una revisión de la evolución de Internet, una descripción sobre las Redes Avanzadas, realizando una exploración sobre la RA Pacific Wave. Se incluyen también justificación y objetivos del presente trabajo.

Capítulo 2, se presentan los detalles de los protocolos de enrutamiento (OSPFv3 y BGP) y protocolo de gestión (SNMP) para IPv6.

Capítulo 3, se describe la metodología para la emulación, se realizan las configuraciones de los protocolos de enrutamiento y gestión, mencionados en el capítulo 2 sobre la **RA**: **Pacific Wave**.

Capítulo 4, se muestran los resultados y conclusiones del presente trabajo.

Capítulo 2

Protocolos de enrutamiento y gestión

2.1 Características de IPv4 & IPv6

El Protocolo de Internet (IP) se ha consolidado como el pilar de la actual Internet. La versión en uso de IP es la versión 4 (IPv4), el cual ha alcanzado el fin de su vida útil y se ha redefinido un nuevo protocolo conocido como IPv6 destinado a reemplazar a IPv4. La razón principal de este reemplazo es la limitación de direcciones que se pueden asignar a los distintos dispositivos con IPv4.

Para un mayor entendimiento de lo que es IPv4 se indican sus características y se realiza una descripción de su formato [60]:

- IPv4 está diseñado para su uso en sistemas interconectados de redes de comunicación por computadora con conmutación de paquetes.
- IPv4 prevé la transmisión de bloques de datos denominados datagramas de origen a destino, donde origen y destino son hosts identificados por direcciones de longitud fija.
- IPv4 proporciona la fragmentación y el reensamblaje de datagramas, si es necesario, para la transmisión a través de redes de paquetes pequeños.
- IPv4 implementa dos funciones básicas: direccionamiento y fragmentación.
- Esquema de direcciones de 32 bits.

La Figura 3 muestra el encabezado de IPv4, así como sus campos.

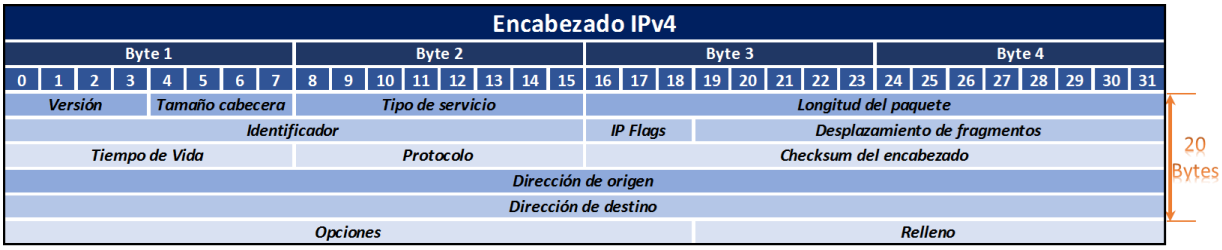


Figura 3 Encabezado de IPv4 [Imagen propia con base en ref. [60]]

Las características de estos campos IPv4 se muestran en la Figura 4:

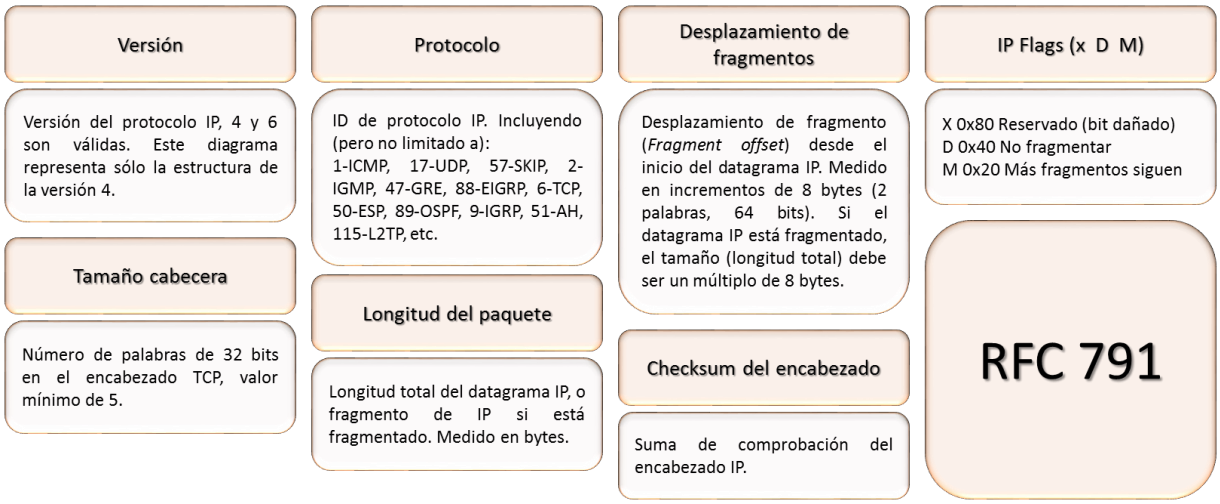


Figura 4 Características de los campos del encabezado de IPv4 [Imagen propia con base en la ref. [60]]

IPv6 incrementó el tamaño en el direccionamiento siendo el de IPv4 de 32 bits a 128 bits para un mayor número de nodos direccionables. El diseño de IPv6 permitió agregar múltiples beneficios en cuanto a seguridad, manejo de calidad de servicio (QoS - *Quality of Service*), a una mayor capacidad de transmisión y mejorar la facilidad en administración, entre otras cosas. Para un mayor entendimiento se realiza una descripción del formato de IPv6**,** así como características [18]:

- El campo de direcciones de 128 bits provee una gran cantidad de direcciones IP, con la posibilidad de asignar direcciones únicas globales a nuevos dispositivos.
- Los múltiples niveles de jerarquía permiten juntar rutas, promoviendo un enrutamiento eficiente y escalable al Internet.
- El proceso de autoconfiguración permite que los nodos de la red IPv6 configuren sus propias direcciones IPv6, facilitando su uso.
- La transición entre proveedores de IPv6 es transparente para los usuarios finales con el mecanismo de renumerado.
- La difusión del protocolo de resolución de direcciones (ARP - *Address Resolution Protocol*) es reemplazada por el uso de *multicast* en el *link local*.
- El encabezado de IPv6 es más eficiente que el de IPv4: tiene menos campos y se elimina *Checksum* del encabezado que tenía IPv4.
- Puede hacerse diferenciación de tráfico utilizando los campos del encabezado.
- Las nuevas extensiones de encabezado reemplazan el campo *Opciones* de IPv4 y proveen mayor flexibilidad.

- IPv6 fue esbozado para manejar mecanismos de movilidad y seguridad de manera más eficiente que el protocolo IPv4.
- Se crearon varios mecanismos junto con el protocolo para tener una transición sin problemas de las redes IPv4 a las IPv6.

La Figura 5 muestra el encabezado de IPv6, así como sus campos.

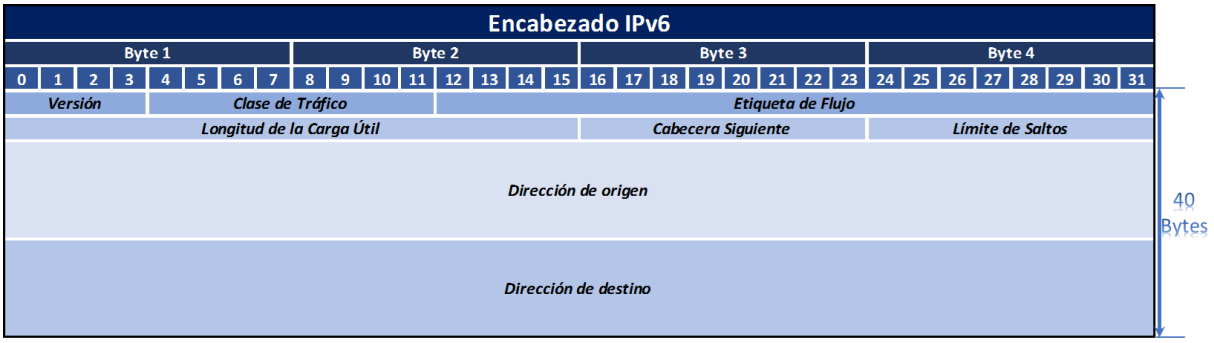


Figura 5 Encabezado IPv6 [Imagen Propia con base en [18]]

Las características de los campos que maneja IPv4 de acuerdo al RFC 791 se muestran en la Figura 6:

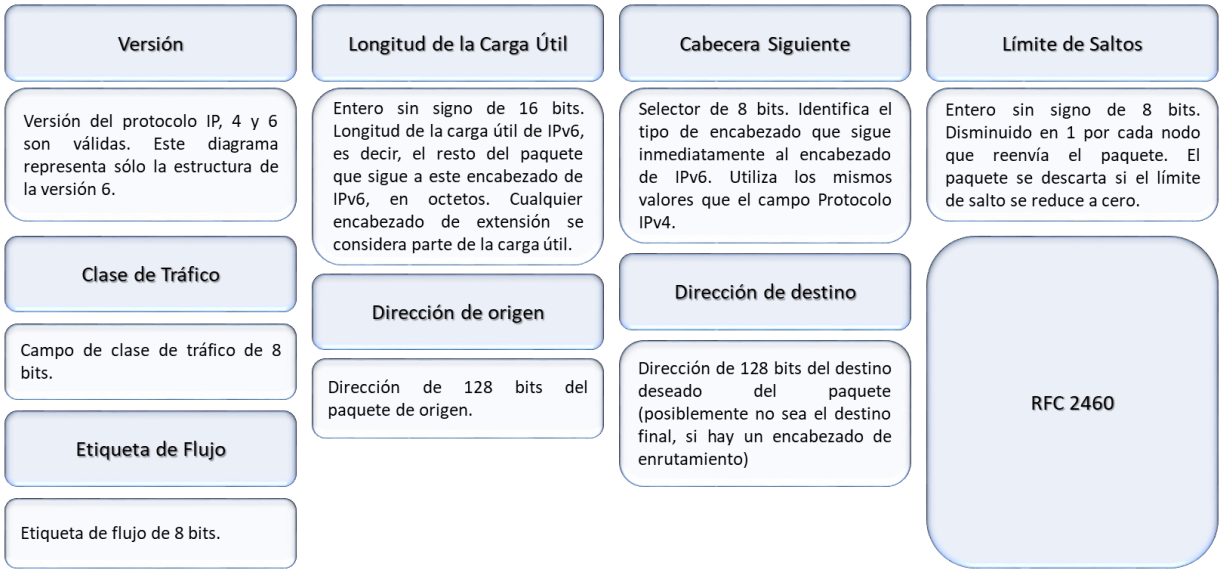


Figura 6 Características de los campos del encabezado de IPv6 [Imagen propia con base en [18]]

Dado que el principal motivo por el que se optó por hacer la transición de IPv4 hacia IPv6 es el direccionamiento a continuación se explica cada una de las características del direccionamiento tanto de IPv4 como IPv6.

2.1.1 Direccionamiento IPv4

El direccionamiento en IPv4 está formado por 4 octetos (grupos de 8 bits en sistema binario o números del 0 al 255 en sistema decimal) un ejemplo se muestra en la Figura 7.

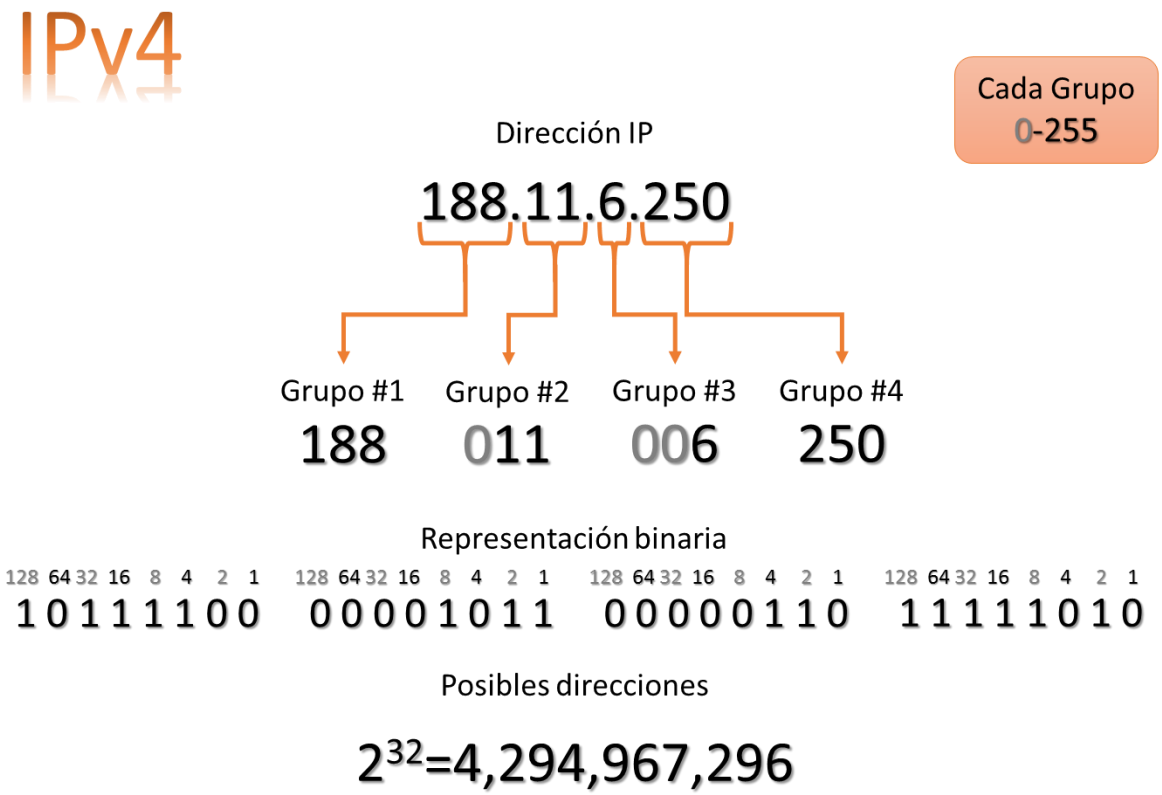


Figura 7 Ejemplo de dirección IPv4 [Imagen propia].

La cantidad de posibles direcciones que se pueden generar es de 2^{32} siendo un total de 4,294,967,296 direcciones en IPv4. Un número considerable para el momento en el que fue creado. Sin embargo, insuficiente para dar una dirección IP a la actual demanda si consideramos que a cada persona del planeta se le proporcionase una IP pensando en la población mundial actual (poco más de 7,500,000,000 habitantes aproximadamente).

Para intentar evitar el agotamiento de direcciones IPv4, se fueron desarrollando algunas medidas, como:

- Redes privadas: Empresas, organizaciones, o incluso hogares con ciertas IPs reservadas como una *Internet local*.

- DHCP (*Dynamic Host Configuration Protocol*): Posibilidad de establecer direcciones IP estáticas o, por otra parte, IPs dinámicas, que permiten ser reutilizadas cuando no están en uso.
- NAT (*Network Address Translation*): Traducción de IPs entre dos redes. Generalmente usada para interconectar redes privadas e Internet.
- CIDR (*Classless Inter-Domain Routing*): Debido a la mala distribución de IPs, se ideó un sistema de división de rangos de IPs más eficiente y flexible.

Aun así, el 3 febrero de **2011** la Corporación para la Asignación de Nombres y Números en Internet (**ICANN**) asignó el último bloque de direcciones de IPv4 disponible para los Registros Regionales de Internet (RIR - *Regional Internet Registries*) [61]. Por lo que el cambio a IPv6 se volvió necesario. Dado que es una nueva versión se realizaron mejoras para este nuevo protocolo dentro de las cuales están:

- Autoconfiguración: Posibilidad de que los propios dispositivos se configuren solos al conectarse a una red.
- Seguridad: El cifrado y autenticación mediante *IPSec* es obligatorio (en IPv4 es opcional), por lo que las comunicaciones serán más seguras.
- Optimización: El diseño de la información enviada ha sido optimizada y simplificada, de forma que tanto los envíos como los procesos que se realizan en los dispositivos de red (como *routers* o similares) es mucho más eficiente.

2.1.2 Direccionamiento IPv6

En la Figura 8 se muestra un ejemplo de una dirección IPv6. Estas direcciones están formadas por 8 grupos de 4 dígitos en hexadecimal cada uno [62]. El espacio de direcciones IPv6 es de 2^{128} siendo un total de 340,282,366,920,938,463,463,374,607,431,768,211,456 direcciones IP.

Este protocolo fue desarrollado por el IETF con apoyo de las RAs, principalmente ESnet quien fue precursor de IPv6.

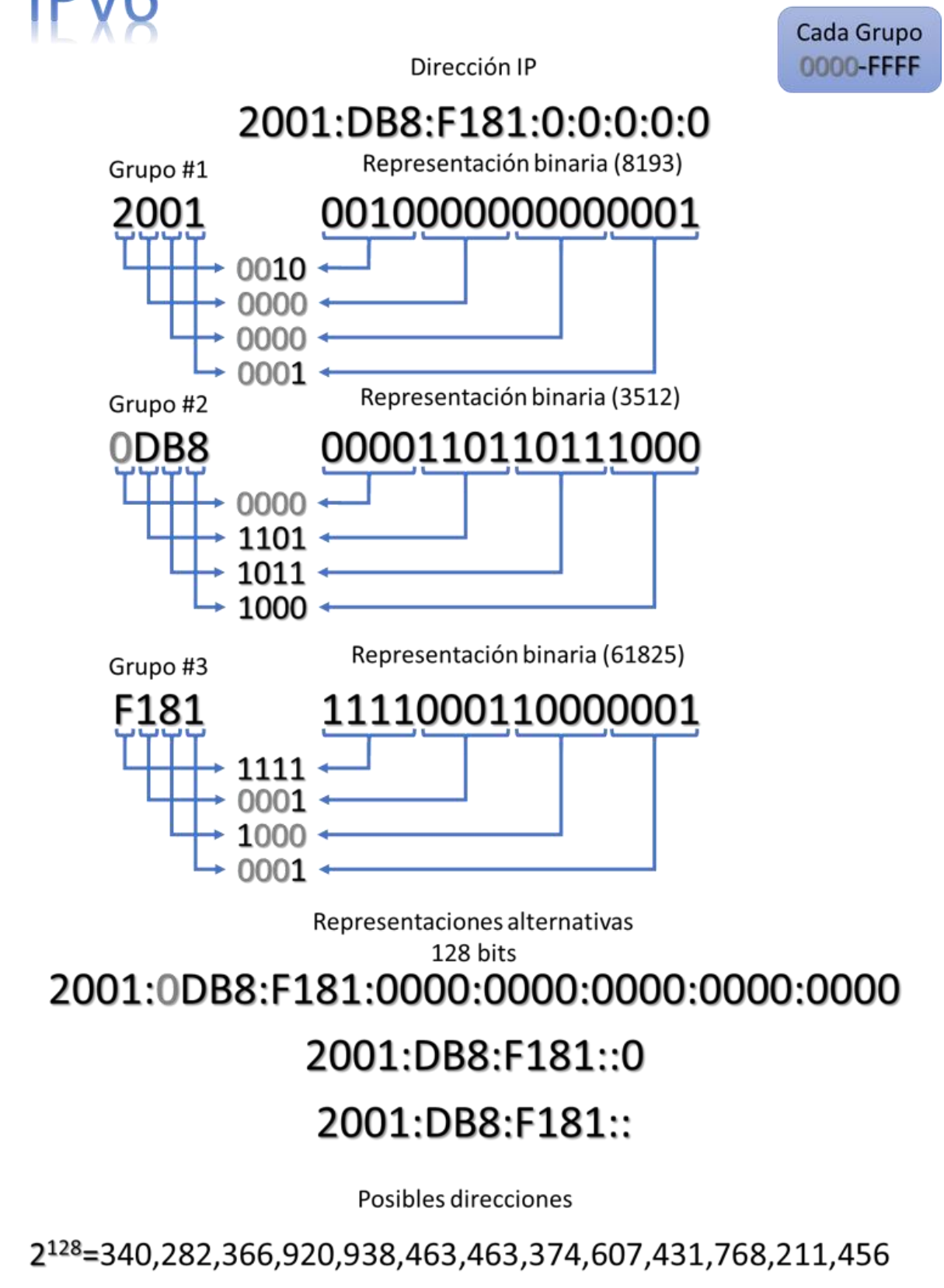


Figura 8 Ejemplo de dirección IPv6 [Imagen propia].

2.2 Protocolos de Enrutamiento

El enrutamiento es el proceso por el cual un *router* decide a dónde enviar un paquete de datos de un origen a un destino, buscando la mejor ruta posible sobre una red, tal que origen y destino se encuentren en redes distintas [60].

Existen dos tipos de enrutamiento: estático y dinámico. En el enrutamiento estático el administrador llena las tablas de enrutamiento manualmente, esto es relativamente fácil de configurar en redes pequeñas, en caso contrario la configuración se vuelve un tanto compleja, de modo que si existe algún cambio en la topología se requerirá del administrador. En el enrutamiento dinámico se usan protocolos para encontrar y actualizar las tablas de enrutamiento de los *routers*, es necesaria en grandes redes, aunque tiene una mayor complejidad para realizar configuraciones, es decir el administrador debe tener un mayor conocimiento para dichas configuraciones, a su vez consume tiempo de procesamiento de los CPU del *router* y el ancho de banda de los enlaces de red [63].

Se le llama tabla de enrutamiento a los registros de direcciones de los nodos en una red.

El *router* es el sistema que realiza las funciones de enrutamiento, interconecta dos o más redes eligiendo el mejor de los caminos o rutas entre las redes, con base en la dirección IP de destino del paquete de datos. Los *routers* deben aprender la dirección de las redes remotas [64].

En una red internacional, como Internet, es muy poco probable que se utilice un único protocolo de enrutamiento para toda la red. Más bien, la red se organizará como una colección de sistemas autónomos.

Un sistema autónomo (AS - *Autonomous System*) es un conjunto de redes IP administradas por uno o más operadores de red, en la que se utiliza una misma política de enrutamiento [65].

Características de un AS

- Los AS se comunican entre sí utilizando **BGP** (*Border Gateway Protocol*) como protocolo de enrutamiento.
- Cada AS posee un número que lo identifica. Regulado por distintas organizaciones internacionales, principalmente por la ICANN (*Internet Corporation for Assigned Names and Numbers*).
- La representación textual de números de sistemas autónomos está definida bajo el RFC 5396 [68].

- Los números de AS de 16 bits fueron definidos mediante el RFC 1930 y para su identificación se utilizan números enteros que van del 0 al 65535. Asimismo, los números de AS de 32 bits fueron definidos mediante el RFC 4893 y para su identificación se utilizan números enteros que van del 0 al 4294967295. Para ambos casos su representación textual del valor decimal *asplain* definida mediante el RFC 5396, dichos números de AS son asignados por la ICANN [[66] - [68]].

Tipos de AS [69]

- De conectividad única (*Stub*). Son aquellos que alcanzan las redes exteriores a través de un único punto de salida.
- De tránsito. Son aquellos que poseen más de una conexión con el exterior y se conectan con distintos sistemas autónomos de manera simultánea, normalmente utilizan su infraestructura para permitir la comunicación entre sus AS vecinos, esto significa que el tráfico entre AS va a pasar a través del AS de tránsito.
- De múltiples conexiones sin tránsito (*Multihomed*). Son aquellos que se conectan con distintos AS vecinos, pero no permiten el tránsito de la información a través de ellos, para lograr esto, el AS pública sólo sus propias rutas y no propagará las rutas aprendidas por otros AS.

Normalmente dentro de un AS se utiliza un protocolo de enrutamiento de tipo **IGP** (*Interior Gateway Protocol*) mientras que para la comunicación entre AS distintos se utiliza un **EGP** (*Exterior Gateway Protocol*), un ejemplo de esto se muestra en la Figura 9 [[68], [69]].

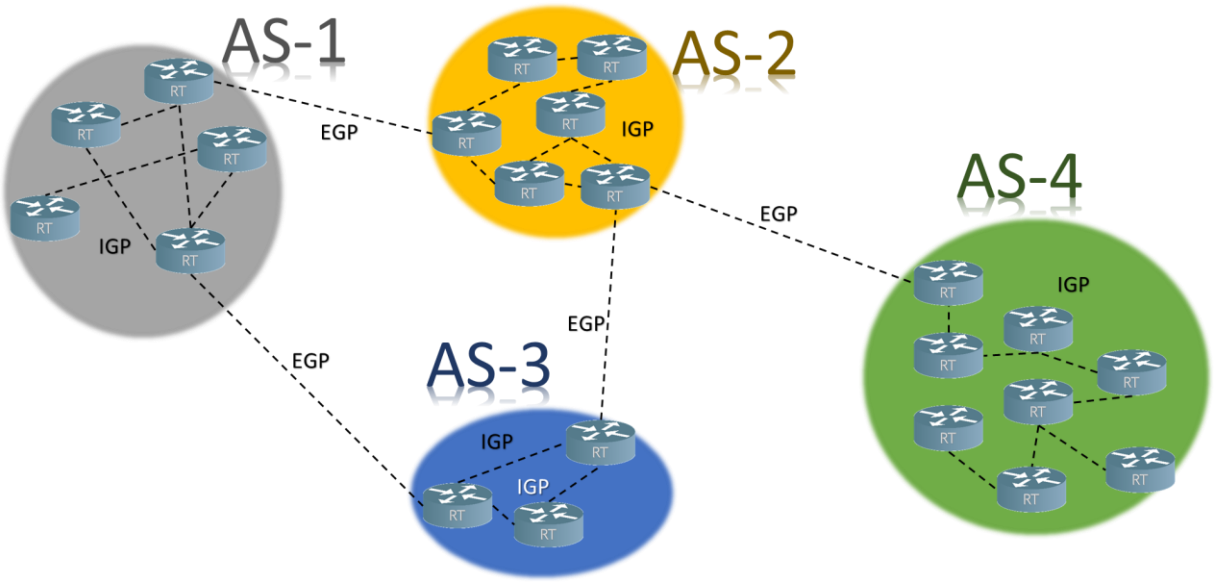


Figura 9 Ejemplo de conexiones entre varios AS [Imagen propia].

Los protocolos IGP se dividen en dos tipos, los protocolos que utilizan algoritmos de tipo Vector-Distancia tales como *RIP* (*Routing Information Protocol*), *IGRP* (*Interior Gateway Routing Protocol*) y *EIGRP*; los protocolos que utilizan algoritmos de tipo Enlace-Estado tales como *OSPF* (*Open Shortest Path First*) e *IS-IS* (*Intermediate System to intermediate System*). Los protocolos EGP utilizan algoritmos de tipo Vector-Ruta como *BGP* (*Border Gateway Protocol*) el cual se usa para la comunicación entre AS. De manera esquemática quedan como se observa en la Figura 10.

El objetivo de un protocolo de enrutamiento es proporcionar la información necesaria para enviar un datagrama desde un origen a un destino [60]. A partir de este punto se realiza la descripción del protocolo OSPF que se emplea en algoritmos de tipo Enlace-Estado y BGP en donde se usan los algoritmos Vector-Ruta [63].

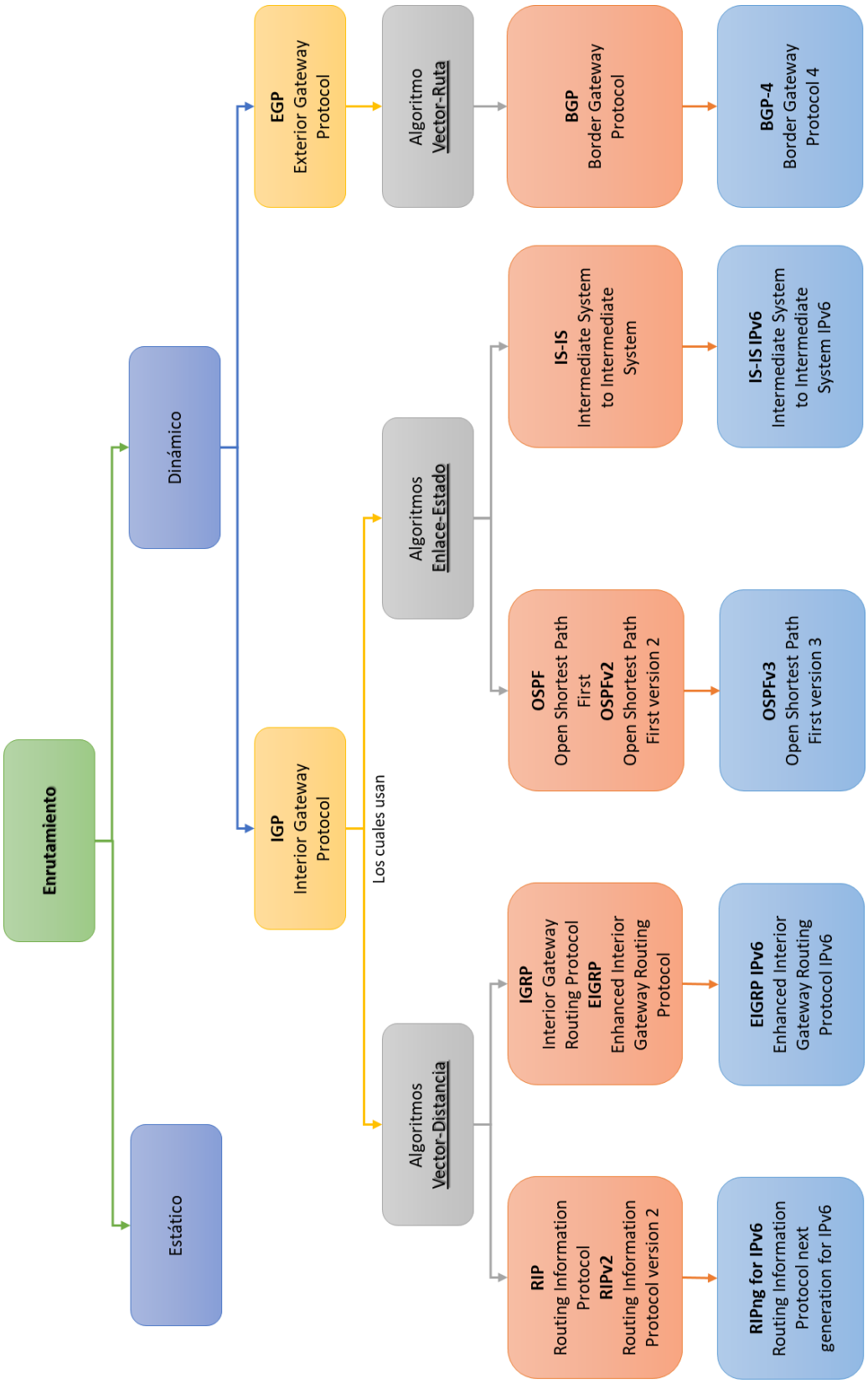


Figura 10 Tipos de enrutamiento [Imagen propia]

2.3 OSPF - Open Shortest Path First

OSPF es un protocolo de enrutamiento de Internet TCP/IP. Clasificado como un IGP. Es decir que distribuye información de enrutamiento entre *routers* que pertenecen a un único AS. El protocolo OSPF se basa en tecnología de Enlace-Estado (LS - *Link-State*) o SPF (*Shortest Path First*). En comparación con un protocolo que usa algoritmos de tipo Vector-Distancia (por ejemplo, RIPng), un LS toma las decisiones de enrutamiento basadas en los estados de los enlaces que conectan el origen y las máquinas de destino. El estado de un enlace es una descripción de esa interfaz y de la relación a sus dispositivos de interconexión de redes vecinos. La información de la interfaz incluye el prefijo del IPv6 de la interfaz. Esta información se propaga mediante anuncios de estado de enlace (LSA - *Link-State Advertisement*).

El protocolo OSPF fue desarrollado por el *Internet Engineering Task Force* (IETF). Ha sido diseñado expresamente para el entorno de Internet TCP/IP, incluido el soporte explícito para CIDR (*Classless Inter-Domain Routing*) y el etiquetado de información de enrutamiento derivado externamente. OSPF también proporciona la autenticación de las actualizaciones de enrutamiento y utiliza la *multicast IP* al enviar y recibir las actualizaciones [[70], [71]].

2.3.1 Breve historia de la tecnología de enrutamiento de Enlace-Estado

El primer protocolo de enrutamiento de LS fue desarrollado para su uso en la red de conmutación de paquetes ARPANET [72]. Ha formado el punto de partida para todos los demás protocolos de LS. El entorno ARPANET homogéneo, es decir, los conmutadores de paquetes de un único proveedor conectados por líneas serie síncronas, simplificó el diseño y la implementación del protocolo original.

El protocolo incluye métodos para la reducción de tráfico de datos y enrutamiento cuando se opera a través de redes de *broadcast* es decir la difusión masiva de información a través de redes. Esto se logra mediante la elección de un *router* designado (DR - *Designated Router*) para cada red de transmisión, que luego origina un LSA para la red. El Grupo de Trabajo OSPF del IETF ha ampliado este trabajo al desarrollar el protocolo OSPF. El concepto de DR se ha mejorado mucho para reducir aún más la cantidad de tráfico de enrutamiento requerido. Las capacidades de *multicast* se utilizan para reducir el ancho de banda de enrutamiento adicional [73].

2.3.2 OSPFv1 Open Shortest Path First version 1

El desarrollo inicial de OSPF comenzó en 1987 por parte del grupo de trabajo IETF. En 1989, se publicó la especificación para OSPFv1 con el RFC 1131 [74]. Se escribieron dos implementaciones. Una implementación se desarrolló para ejecutarse en *routers*, y la otra se desarrolló para ejecutarse en estaciones de trabajo UNIX. Esta última implementación se convirtió en un proceso UNIX generalizado que se conoce como GATED (*Gateway Routing Daemon*). OSPFv1 era un protocolo de enrutamiento experimental y nunca se implementó.

2.3.3 OSPFv2 Open Shortest Path First version 2

En 1991, John Moy introdujo la segunda versión de OSPF en el RFC 1247 *OSPF Version 2*, tres años más tarde en 1994, se actualizó la especificación este protocolo mediante el RFC 1583 *OSPF Version 2*, finalmente en 1998 se hace una última actualización para *OSPF Version 2* con el RFC 2328 que en la actualidad sigue siendo el protocolo de enrutamiento que se maneja para el IPv4 [71].

2.3.3.1 Formato Encabezado OSPFv2

Cada paquete OSPFv2 comienza con un encabezado estándar de 4 bytes. Este encabezado contiene toda la información necesaria para determinar si el paquete puede aceptarse para su posterior procesamiento. En la Figura 11 se muestra el encabezado de OSPFv2 y sus respectivos campos.

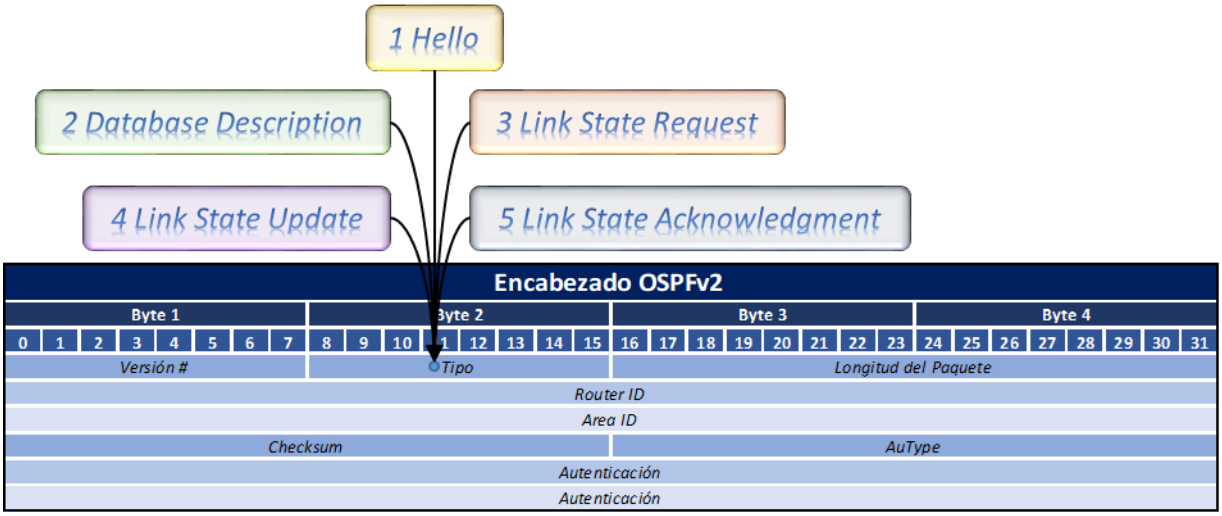


Figura 11 Encabezado OSPFv2 [Imagen propia]

2.3.3.2 Campos del Encabezado de OSPFv2

A continuación, se describen las características de los campos del encabezado de OSPFv2:

Versión #, el número de versión de OSPF.

Tipo, los tipos de paquetes OSPF (*Hello, DBD, LSR, LSU y LSAck*).

Longitud del paquete, esta longitud incluye el encabezado estándar de OSPFv2.

Router ID de la fuente del paquete.

Area ID, un número de 32 bits que identifica el área a la que pertenece este paquete. Todos los paquetes OSPF están asociados con una sola área. Los paquetes que viajan a través de un enlace virtual están etiquetados con el *Area ID* de la red *backbone* de 0.0.0.0.

Checksum es la suma de comprobación de IP estándar de todo el contenido del paquete, comenzando con el encabezado del paquete OSPF, pero excluyendo el campo de autenticación de 64 bits. Esta suma de comprobación se calcula como el complemento de 16 bits de la suma del complemento de todas las palabras de 16 bits en el paquete, excepto el campo de autenticación. Si la longitud del paquete no es un número entero de palabras de 16 bits, el paquete se rellena con un byte de cero antes de la suma de comprobación. La suma de comprobación se considera parte del procedimiento de autenticación de paquetes; para algunos tipos de autenticación se omite el cálculo de suma de comprobación.

AuType, identifica el procedimiento de autenticación que se utilizará para el paquete.

Autenticación, un campo de 64 bits para uso por el esquema de autenticación.

2.3.4 OSPFv3 Open Shortest Path First version 3

Para IPv6 en 1999 aparece el RFC 2740 donde se daba la primera interacción de OSPF con IPv6, años más tarde en julio de 2008 con el RFC 5340 dejando como obsoleto al anterior estándar. Los mecanismos fundamentales de OSPF permanecen sin modificaciones. Sin embargo, algunos cambios han sido necesarios, ya sea debido a cambios en la semántica del protocolo entre IPv4 e IPv6, o simplemente para manejar el aumento del tamaño de la dirección de IPv6. Estas modificaciones necesitaron incrementar la versión del protocolo de la versión 2 a la versión 3. OSPF para IPv6 también se conoce como OSPF versión 3 (OSPFv3) [50].

2.3.4.1 Detalles de implementación de OSPFv3

OSPFv3 usa una ***Link-State Database*** (LSDB) compuesta de ***Link State Advertisements*** (LSAs) y sincronizada entre *routers* adyacentes. La sincronización inicial se realiza a través del proceso de ***Database Exchange***, que incluye el intercambio de ***Database Description*** (DBD), ***Link State Request*** (LSR), y paquetes ***Link State Update*** (LSU). A partir de entonces, la sincronización de la base de datos se mantiene mediante un *flooding* (una inundación de paquetes por la red), utilizando LSU y paquetes de ***Link State Acknowledgment*** (LSAck). OSPFv3 usa los paquetes

Hello para descubrir y mantener relaciones vecinas, así como para elegir ***Designated Routers*** (DR) y ***Backup Designated Routers*** (BDR). La decisión sobre qué relaciones vecinas se convierten en adyacentes, y las ideas básicas detrás del enrutamiento *inter-area*, la importación de información externa en *AS-External-LSAs*, y los diversos cálculos de enrutamiento son las mismas que en OSPFv2 [[70], [71]].

OSPF permite dividir en áreas un AS con la finalidad de una mejor administración cuando la topología es extensa, con base en ello OSPF clasifica los *routers* de acuerdo a su función.

2.3.4.2 Clasificación de *routers*

Cuando un AS se divide en áreas OSPF, los *routers* se dividen según la función en las siguientes cuatro categorías [[70], [71]] :

- *Internal Router* (IR)

Un *router* con todas las redes conectadas directamente que pertenecen a la misma área. Estos *routers* ejecutan una única copia del algoritmo de enrutamiento básico.

- *Area Border Router* (ABR)

Un *router* que se conecta a múltiples áreas. Los *routers* de borde de área ejecutan múltiples copias del algoritmo básico, una copia para cada área adjunta. Los *routers* de borde de área resumen la información topológica de sus áreas adjuntas para su distribución a la red *Backbone*. El *Backbone* a su vez distribuye la información a las otras áreas.

- *Backbone Router* (BR)

Un *router* que tiene una interfaz para el área de la red *Backbone*. Esto incluye todos los *routers* que interactúan con más de un área (es decir, ABR). Sin embargo, los *routers Backbone* no tienen que ser *routers* de borde de área. Se admiten *routers* con todas las interfaces conectadas al área de red *Backbone*.

- *Autonomous System Boundary Router* (ASBR)

Un *router* que intercambia información de enrutamiento con *routers* que pertenecen a otros AS. Tal *router* notifica información de enrutamiento externo AS en todo el AS. Las rutas a cada *router* de límite AS son conocidas por cada *router* en el AS. Esta clasificación es completamente independiente de las clasificaciones anteriores: los *routers* de límites AS pueden ser *routers* internos o de borde de área, y pueden o no participar en la red *Backbone*.

En la Figura 12 se muestra un ejemplo de la clasificación de los *routers* OSPF de acuerdo a su función.

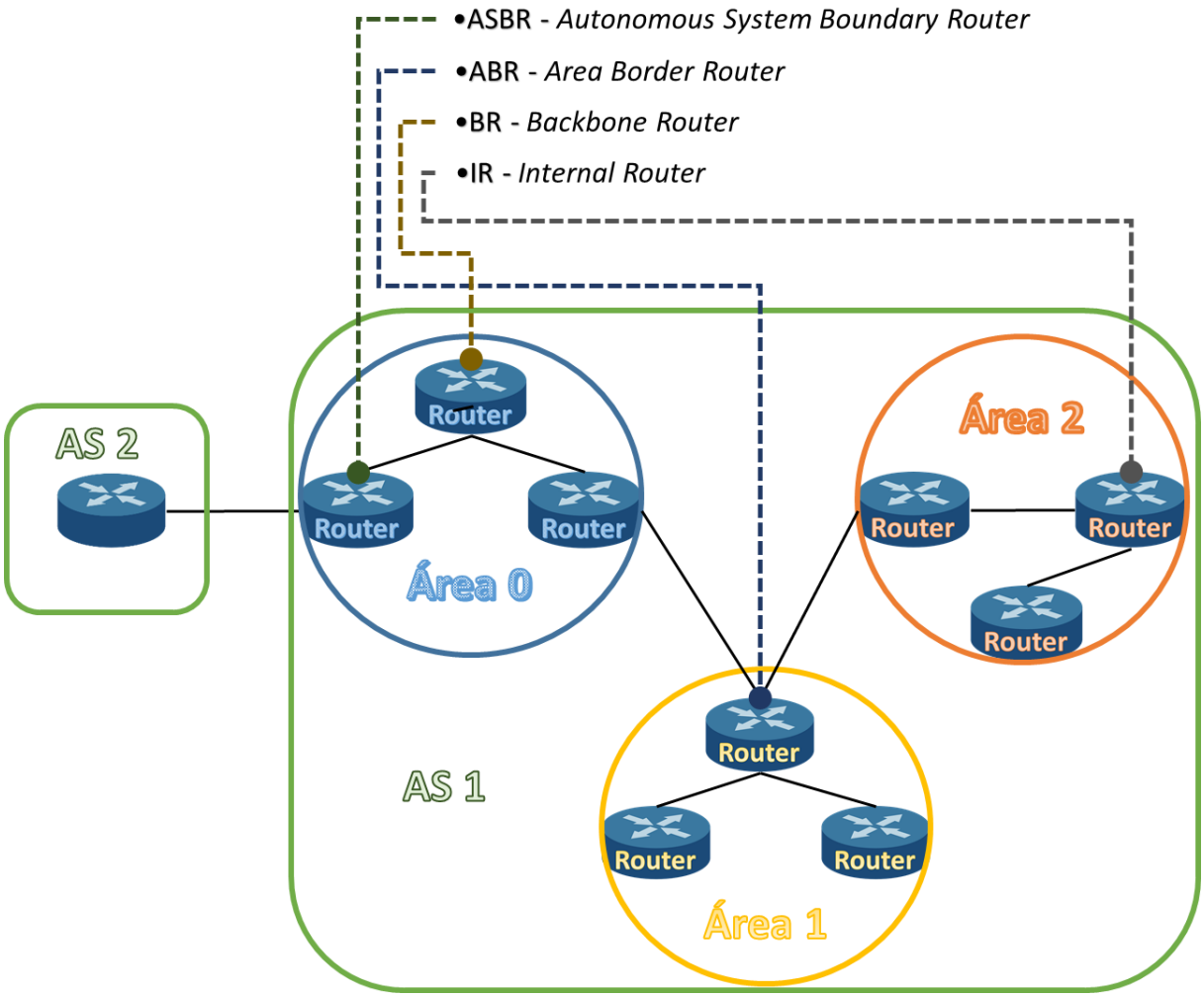


Figura 12 Ejemplo de los tipos de Routers [Imagen propia]

2.3.4.3 Tipos de Áreas

Las áreas en OSPF están diseñadas para minimizar la *LSDB* y los tamaños de la tabla de enrutamiento para los *routers* internos de las áreas. Esto permite que los *routers* con recursos mínimos participen incluso en dominios de enrutamiento OSPF muy grandes. Los tipos de áreas que se manejan son [[70], [71]]:

- *Standard Area*

Es el área por *default* y permite actualización de enlaces, sumarización de rutas y rutas externas.

- *Backbone Area*

Es el área principal de la topología OSPF. Es obligatorio que exista, se etiqueta como área 0 y tiene las mismas características que el área estándar.

- *Stub Area*

Este tipo de área no acepta información acerca de rutas externas al AS (redistribución), tales como rutas desde orígenes no OSPF. Si los *routers* necesitan enrutar hacia redes ubicadas fuera del AS OSPF, utilizan una ruta por *default* que es enviada por el ABR hacia los demás *routers* internos del área *Stub*. En esta área no se permiten ASBR (a menos que el ABR sea al mismo tiempo un ASBR). Las áreas de *stub* solo transportan *Router-LSAs*, *Network-LSAs*, *Inter-Area-Prefix-LSAs*, *Link-LSAs* e *Intra-Area-Prefix-LSAs*. Las áreas NSSA están restringidas a estos tipos y, por supuesto sólo *NSSA-LSAs*.

- *NSSA Area*

El área no exclusiva de rutas internas (*NSSA - Not-So-Stubby Area*), no admiten ningún LSA externo únicamente *NSSA-LSAs.*

2.3.4.4 Tipos de Rutas

Hay cuatro posibles tipos de rutas utilizadas para enrutar el tráfico al destino, enumeradas aquí en orden decreciente de preferencia: *intra-area*, *inter-area*, tipo 1 externo o tipo 2 externo. Las rutas *intra-area* indican los destinos que pertenecen a una de las áreas adjuntas del *router*. Los caminos *inter-area* son rutas a destinos en otras áreas OSPF. Caminos externos tipo 1 y tipo 2 son rutas a destinos externos al AS [70].

2.3.4.5 Base de datos de enlaces (LSDB - Link-State Database)

Un *router* tiene una LSDB por separado para cada área a la que pertenece. Todos los *routers* que pertenecen a la misma área tienen LSDBs idénticas para el área.

Las bases de datos para cada área individual siempre se tratan por separado. El cálculo del camino más corto se realiza por separado para cada área. Los componentes de la LSDB de área realizan un *flooding* solo en la zona [[70], [71]].

La LSDB se divide en tres estructuras de datos separadas.

1. Los LSAs con ámbito de *flooding* AS están contenidos dentro de la estructura de datos OSPF de nivel superior, esto incluye *AS-External-LSAs*.

2. Los LSAs con alcance de *flooding* de área están contenidos dentro de la estructura de área apropiada, esto incluye los *Router-LSAs*, *Network-LSAs*, *Inter-Area-Prefix-LSAs*, *Inter-Area-Router-LSAs*, *NSSA-LSAs* e *Intra-Area-Prefix-LSAs*.
3. Los LSAs con un tipo LS desconocido, esto incluye *Link-LSAs*.

2.3.4.6 Notificaciones de estado de enlace (*LSA - Link State Advertisements*)

Cada *router* en el AS origina una o más LSAs. La colección de LSAs forma la LSDB. Cada tipo de LSA tiene una función separada. Desde la LSDB, cada *router* construye el árbol de ruta más corto con él mismo como raíz. Esto produce una tabla de enrutamiento [[70], [71]].

2.3.4.6.1 Formatos LSA

Cada LSA comienza con un encabezado LSA estándar y describe una parte del dominio de enrutamiento OSPF. Todos los LSAs hacen un *flooding* en todo el dominio de enrutamiento OSPF. El algoritmo de *flooding* es confiable, asegurando que todos los *routers* comunes a su alcance tengan la misma colección de LSAs asociados. Esta colección de LSA se denomina *link-state database*. Desde la *link-state database*, cada *router* construye un árbol de ruta más corta consigo mismo como raíz. Esto produce una tabla de enrutamiento [70].

2.3.4.6.2 El encabezado LSA

Todos los LSAs comienzan con un encabezado común de 20 bytes. Este encabezado contiene suficiente información para identificar de manera única el LSA (*Tipo LS*, *Link State ID* y *Advertising Router*). Varias instancias del LSA pueden existir en el dominio de enrutamiento al mismo tiempo. En consecuencia, es necesario determinar qué instancia es más reciente. Esto se logra al examinar la *Edad LS*, el *Número de Secuencia LS* y los campos de *LS Checksum* que también están contenidos en el encabezado LSA [70]. En la Figura 13 se muestra el formato del encabezado del SLA.

OSPFv3 Encabezado LSA	
Edad LS	***Tipo LS***
Link State ID	
Advertising Router	
Número de Secuencia LS	
LS Checksum	***Longitud***

Figura 13 Formato del encabezado de un SLA

2.3.4.6.3 Tipo de LS

El campo Tipo LS indica la función realizada por el LSA. Los tres bits de orden superior codifican las propiedades genéricas de la LSA, mientras que el resto (llamado Código de función LSA) indican la funcionalidad específica de la LSA [70]. El formato del tipo LS es el mostrado en la Figura 14:

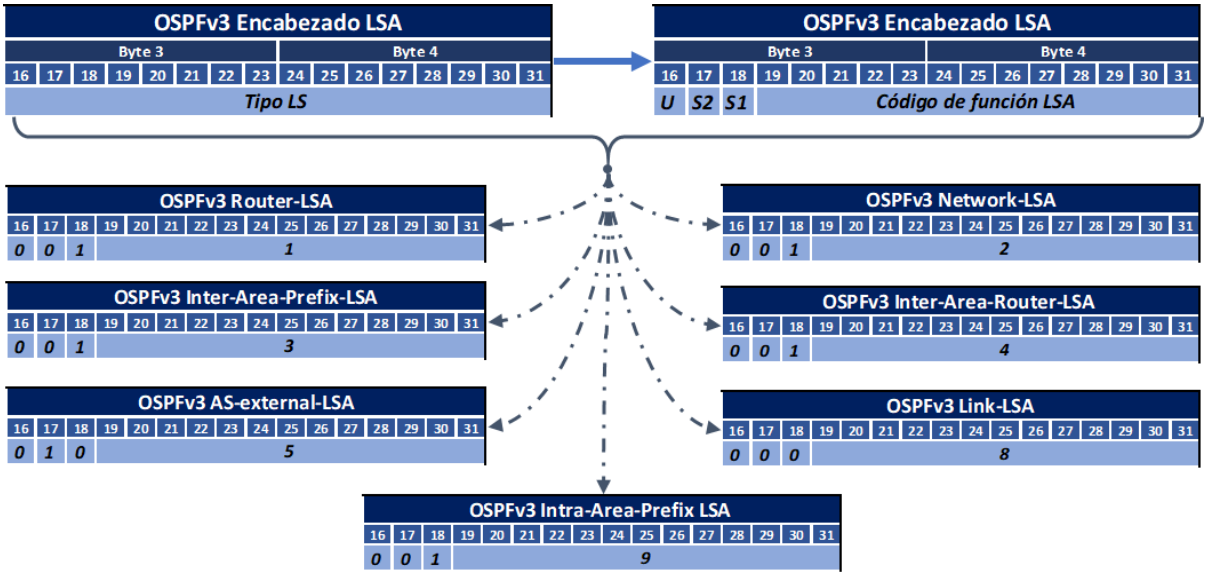


Figura 14 Formatos de tipos de LSAs con base en [70]

El bit U indica cómo el LSA debe ser manejado por un *router* que no reconoce el código de función del LSA. Sus valores son de acuerdo a la Tabla 2:

U-Bit	Manejo de LSA
0	Trate el LSA como si tuviera un alcance de *flooding* de *link-local*
1	Almacenar e inundar el LSA como si se entendiera el tipo.

Tabla 2 Bit U [70]

Los bits S1 y S2 indican el alcance por *flooding* de la LSA. Los valores están codificados como se muestra en la Tabla 3:

S2	S1	Alcance por *flooding*
0	**0**	*Link-Local Scoping* - Inundado solo en el enlace de origen
0	**1**	*Area Scoping* - Inundado solo en la zona de origen.
1	**0**	*AS Scoping* - Inundado a lo largo de AS
1	**1**	Reservado

Tabla 3 Bit S1 y S2 (Flooding Scope) [70]

La Tabla 4 describe los tipos de LSAs, cada uno de los cuales tiene un propósito diferente:

Código de función LSA	Tipo LS	LSA
1	0x2001	*Router-LSA*
Describe el *link-state* y los costos de los enlaces de un *router* del área. Estos LSAs se propagan únicamente dentro de un área. El LSA indica si el *router* es ABR o ASBR. Los *Router-LSA*s también se utilizan para anunciar redes stub. La información de la interfaz del *router* se distribuye entre varios *Router-LSA*s. Los receptores deben concatenar todos los *Router-LSA*s originados por un *router* determinado al ejecutar el cálculo de SPF.		
2	0x2002	*Network-LSA*
Describe el *link-state* y la información de costo de todos los *routers* conectados a la red. Solo un DR rastrea esta información y puede generar un *Network-LSA*.		
3	0x2003	*Inter-Area-Prefix-LSA*
Para ABRs, anuncia redes internas a *routers* en otras áreas (rutas entre áreas). Los LSAs de tipo 3 pueden representar una sola red o un conjunto de redes resumidas en un anuncio. Solo los ABRs generan LSAs de resumen. Las direcciones para estos LSAs se expresan como prefijo, longitud de prefijo en lugar de dirección, máscara. La ruta predeterminada se expresa como un prefijo con longitud 0.		
4	0x2004	*Inter-Area-Router-LSA*
Para ASBRs, anuncia la ubicación de un ASBR. Los *routers* que intentan llegar a una red externa utilizan estos anuncios para determinar la mejor ruta al siguiente salto. Los ABR generan los LSAs de tipo 4 en nombre de los ASBR.		
5	0x4005	*AS-External-LSA*
Redistribuye las rutas de otro AS, generalmente de un protocolo de enrutamiento diferente a OSPFv3. Las direcciones para estos LSAs se expresan como prefijo, longitud de prefijo en lugar de dirección, máscara. La ruta predeterminada se expresa como un prefijo con longitud 0.		
7	0x2007	*NSSA-LSA*
Estas LSAs son originadas por ASBR dentro de una NSSA y describen destinos externos al AS que pueden o no propagarse fuera de la NSSA. Una dirección IPv6 global debe seleccionarse como dirección de reenvío para NSSA-LSA que se propagarán por los ABR NSSA. Las NSSA-LSA tienen alcance de inundación de área.		
8	0x0008	*Link-LSA*
Tienen un alcance de *flooding* de *local-link* y nunca más allá del enlace con el que están asociados. Los *Link-LSA* proporcionan la dirección *local-link* del *router* a todos los demás *router* conectados al enlace, informan a otros *routers* adjuntos al enlace de una lista de prefijos para asociar con el enlace y permiten al *router* afirmar una colección de bits de opciones para asociar con un *Network-LSA* que se originará para el enlace.		
9	0x2009	*Intra-Area-Prefix-LSA*
Un *router* puede originar varios *Intra-Area-Prefix-LSAs* para cada *router*, cada uno con una ID de estado de enlace única. El ID de estado de enlace para cada *Intra-Area-Prefix-LSA* describe su asociación con el *Router-LSA* o con el *Network-LSA* y contiene prefijos para redes de tránsito y stub.		

Tabla 4 Código de función de LSAs y descripción [70]

2.3.4.7 Mensajes Hello Protocol

El *Hello Protocol* es responsable de establecer y mantener relaciones vecinas. También asegura que la comunicación entre vecinos sea bidireccional. Los *Hello Packets* se envían periódicamente a todas las interfaces del *router*. La comunicación bidireccional se indica cuando el *router* se ve a sí mismo en la lista en *Hello Packet* del vecino [[70], [71]].

2.3.4.8 Paquetes del protocolo de enrutamiento

El protocolo OSPF se ejecuta directamente sobre IP, OSPF no proporciona ningún soporte explícito de fragmentación/reensamblaje. Cuando es necesaria la fragmentación, se utiliza la fragmentación/reensamblaje de IP. Los tipos de paquetes OSPF se enumeran a continuación en la Tabla 5 [[70], [71]].

Tipo	Nombre del Paquete	Función de protocolo
1	*Hello*	Descubrir / mantener vecinos
2	*Database Description [DBD]*	Resumir los contenidos de la base de datos
3	*Link State Request [LSR]*	Descargar base de datos
4	*Link State Update [LSU]*	Actualización de la base
5	*Link State Ack [LSAck]*	Reconocimiento de *floodings*

Tabla 5 Tipos de Paquetes OSPF

El *Hello Protocol* de OSPF usa paquetes *Hello* para descubrir y mantener relaciones vecinas. Los paquetes de DBD y LSR se utilizan en la formación de adyacencias. El mecanismo de LSU confiable de OSPF se implementa mediante los paquetes de LSAck.

Cada paquete de LSU contiene un conjunto de nuevos LSAs con información de un salto más allá de su punto de origen. Un único paquete LSU puede contener los LSAs de varios *routers*.

Los paquetes de enrutamiento OSPF (a excepción de *Hello*) se envían solo por adyacencias. Esto significa que todos los paquetes de protocolo OSPF viajan un solo salto de IP, excepto aquellos que se envían a través de adyacencias virtuales (es decir que el área no está conectada directamente al área 0). La dirección de origen de IP de un paquete de protocolo OSPF es un extremo de una adyacencia de *router*, y la dirección de destino de IP es el otro extremo de la adyacencia o una dirección IP de *multicast* [[70], [71]].

2.3.4.9 Router Designado (*DR - Designated Router*)

El DR seleccionado para la red adjunta. El DR se selecciona en todas las redes *broadcast* y *NBMA* mediante el protocolo *Hello*. Se guardan dos piezas de identificación para el DR: su

Router ID y su dirección de interfaz IP en la red. El DR notifica el estado del enlace para la red; esta *network-LSA* está etiquetada con la dirección IP del DR. El DR se inicializa a 0.0.0.0, lo que indica la falta de un DR [[70], [71]].

2.3.4.10 Router Designado de respaldo (*BDR - Backup Designated Router*)

El BDR también se selecciona en todas las redes de *broadcast* y *NBMA* mediante el protocolo *Hello*. Todos los *routers* de la red conectada se vuelven adyacentes tanto al DR como al BDR. El BDR se convierte en DR cuando falla el DR actual. El BDR se inicializa a 0.0.0.0, lo que indica la falta de un BDR [[70], [71]].

2.3.4.11 Estados de Adyacencias

OSPF crea adyacencias entre *routers* vecinos para intercambiar información de enrutamiento como se muestra en la Figura 15.

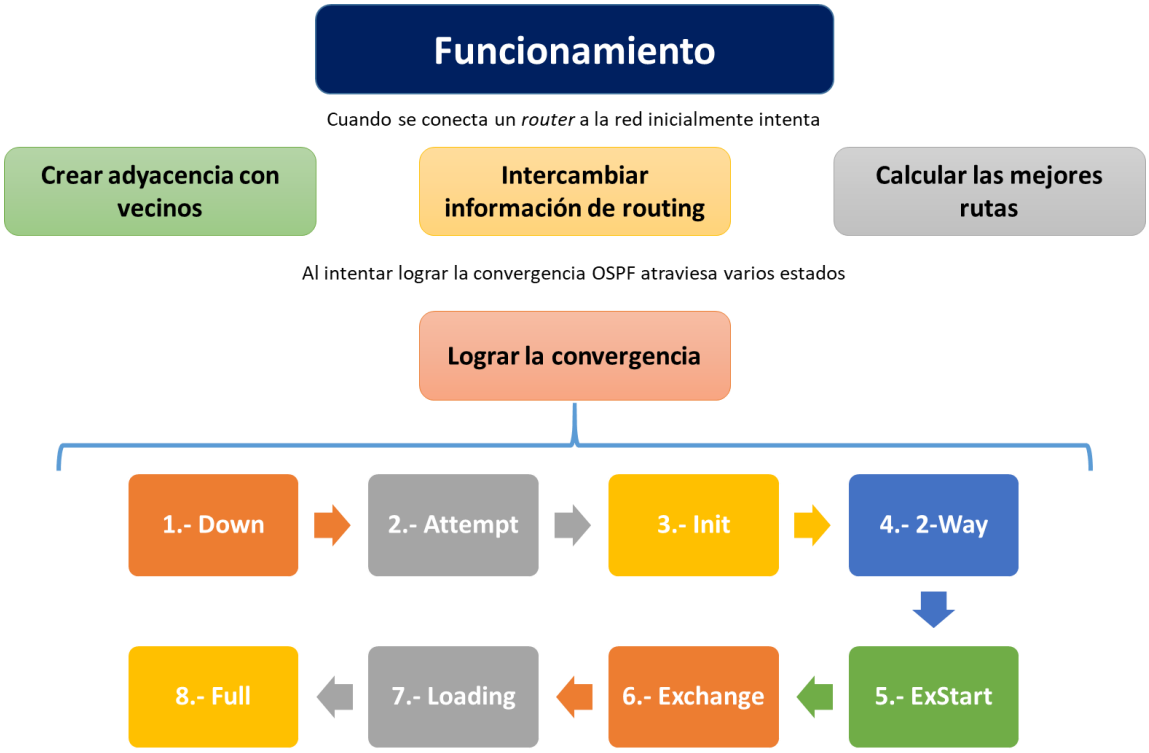


Figura 15 Funcionamiento inicial de un router con OSPF

Cuando se forma la adyacencia OSPF, un *router* pasa por diversos cambios de estado antes de volverse totalmente adyacente con su vecino. Los estados de estas adyacencias se describen en la Tabla 6.

Estado	Descripción
1. Down	Este es el estado inicial de una conversación vecina. Indica que no ha habido información reciente recibida del vecino.
2. Attempt	Este estado sólo es válido para vecinos conectados a redes NBMA. Indica que no se ha recibido información reciente del vecino, pero que se debe hacer un esfuerzo más concertado para contactar al vecino. Esto se hace enviando los paquetes *Hello* vecinos.
3. Init	En este estado, recientemente se ha visto un paquete *Hello* del vecino. Sin embargo, la comunicación bidireccional aún no se ha establecido con el vecino (es decir, el *router* en sí no apareció en el paquete *Hello* del vecino). Todos los vecinos en este estado (o superior) se enumeran en los paquetes *Hello* enviados desde la interfaz asociada.
4. 2-Way	En este estado, la comunicación entre los dos *routers* es bidireccional. Esto ha sido asegurado por la operación del protocolo *Hello*. Este es el estado más avanzado, menos que el estado *init* de adyacencia. El DR se selecciona del conjunto de vecinos en el estado de 2 vías o superior.
5. ExStart	Este es el primer paso para crear una adyacencia entre los dos *routers* vecinos. El objetivo de este paso es decidir qué *router* es el maestro y decidir el número de secuencia de DD (*Database Description*) inicial. Las conversaciones vecinas en este estado o más, se llaman adyacencias.
6. Exchange	En este estado, el *router* describe su LSDB completa enviando paquetes DBD al vecino. Cada paquete de DBD tiene un número de secuencia DD, y está explícitamente reconocido. Sólo se permite un paquete de DBD en cualquier momento. En este estado, los paquetes de LSR también pueden enviarse solicitando los LSAs más recientes del vecino. El procedimiento de *flooding* utiliza todas las adyacencias en el estado de *Exchange* o superior. De hecho, estas adyacencias son totalmente capaces de transmitir y recibir todos los tipos de paquetes de protocolo de enrutamiento OSPF.
7. Loading	En este estado, los paquetes de LSR se envían al vecino que solicita los LSA más recientes que se han descubierto (pero aún no se han recibido) en el estado de Exchange.
8. Full	En este estado, los *routers* vecinos son completamente adyacentes. Estas adyacencias ahora aparecerán en *Router-LSAs* y *Network-LSAs*.

Tabla 6 Estados OSPF [[70], [71]]

2.3.4.12 Fórmula de la métrica de OSPF

El costo (también llamado métrica) de una interfaz en OSPF es una indicación de la sobrecarga requerida para enviar paquetes a través de una interfaz específica. El costo de una interfaz es inversamente proporcional al ancho de banda de dicha interfaz. Un mayor ancho de banda

indica un menor costo. El cruce de una línea serial de 56k implica mayores gastos generales (costo mayor) y más retrasos de tiempo que el cruce de una línea Ethernet de 10M.

OSPF utiliza un ancho de banda de referencia de 100 Mbps para el cálculo de costos. La fórmula para calcular el costo es el ancho de banda de referencia dividido por el ancho de banda de la interfaz [75].

$$costo = \frac{ancho\ de\ banda\ de\ referencia\ [bps]}{ancho\ de\ banda\ de\ la\ interfaz\ [bps]} \rightarrow Ecuación\ 1$$

Por ejemplo, en el caso de Ethernet, es

$$\frac{100Mbps}{10\ Mbps} = 10 \rightarrow Ecuación\ 2$$

El costo es de 10

2.3.4.13 Formatos de OSPFv3

OSPF se ejecuta directamente sobre la capa de red del IPv6. Por lo tanto, los paquetes OSPF están encapsulados únicamente por IPv6 y encabezados locales de enlace de datos. Características importantes del encapsulado IPv6 de OSPF:

- Uso de *multicast* IPv6. Algunos mensajes OSPF son *multicast* cuando se envían a través de redes de broadcast. Se usan dos direcciones distintas de *multicast* IP. Los paquetes enviados a estas direcciones de *multicast* nunca se deben reenviar; están destinados a viajar un solo salto solamente. Como tal, las direcciones de *multicast* se han elegido con alcance de *link-local* y los paquetes enviados a estas direcciones deben tener su Límite de Salto de IPv6 (*IPv6 Hop Limit*) establecido en 1.

AllSPFRouters

A esta dirección de *multicast* se le ha asignado el valor *FF02::5*. Todos los *routers* que ejecutan OSPF deben estar preparados para recibir los paquetes enviados a esta dirección. Los *Hello Packets* siempre se envían a este destino.

AllDRouters

A esta dirección de *multicast* se le ha asignado el valor *FF02::6*. Tanto el DR como el BDR deben estar preparados para recibir los paquetes destinados a esta dirección.

- OSPF es el protocolo IP 89. Este número se inserta en el campo *Next Header* del encabezado de IPv6.

Existen cinco tipos de paquetes OSPF. Todos los tipos de paquetes OSPFv3 comienzan con un encabezado estándar de 16 bytes. Todos los tipos de paquetes OSPF (que no sean los paquetes *Hello* de OSPFv3) se ocupan de listas de LSAs. Por ejemplo, los paquetes de LSU implementan un *flooding* de LSAs en todo el dominio de enrutamiento OSPFv3 [70].

2.3.4.13.1 Paquetes *Hello*

Los paquetes *Hello* se envían periódicamente en todas las interfaces (incluidos los enlaces virtuales) para establecer y mantener relaciones vecinas. Los paquetes *Hello* son *multicast* lo que permite el descubrimiento dinámico de *routers* vecinos.

Todos los *routers* conectados a un enlace común deben aceptar los parámetros *HelloInterval* y *RouterDeadInterval*. *HelloInterval*, número de segundos que el *router* debe esperar para enviar un mensaje Hello. *DeadRouterInterval*, la cantidad de segundos antes de declarar a un *router* OSPF *down* (desconectado).

El paquete *Hello* también contiene los campos utilizados en la elección del DR (*Designated Router ID* y *Backup Designated Router ID*) y los campos utilizados para detectar comunicación bidireccional (los *Router IDs* de todos los vecinos cuyos *Hellos* se han recibido recientemente) [[70], [71]].

2.3.4.13.2 Paquetes de descripción de la base de datos

Los paquetes de OSPF con mensaje del tipo DBD son de tipo 2. Estos paquetes se intercambian cuando se inicia una adyacencia. Describen los contenidos de la base de datos de estado de enlace. Se pueden usar múltiples paquetes para describir la base de datos. Para este propósito, se usa un procedimiento de encuesta. Uno de los *routers* está designado para ser el maestro y el otro es el esclavo. El maestro envía paquetes de DBD (sondeos) reconocidos por paquetes de DBD enviados por el esclavo (respuestas). Las respuestas están vinculadas a las encuestas a través de los números de secuencia DD de los paquetes.

El formato del paquete DBD es muy similar tanto al paquete de LSR como al LSAck. La parte principal de los tres es una lista de elementos, cada elemento que describe una parte de la base de datos de estado de enlace [[70], [71]].

2.3.4.13.3 Paquetes Link State Request

Los paquetes de LSR son paquete OSPF tipo 3. Después de intercambiar paquetes DBD con un *router* vecino, un *router* puede encontrar que las partes de su base de datos de *link-state* están

desactualizadas. El paquete LSR se utiliza para solicitar las piezas de la base de datos del vecino que están más actualizadas. Es posible que se necesiten varios paquetes de LSR.

Un *router* que envía un paquete LSR tiene en cuenta la instancia precisa de las piezas de la base de datos que está solicitando. [[70], [71]].

2.3.4.13.4 Paquetes Link State Update

Los paquetes de LSU son paquetes OSPF tipo 4. Estos paquetes implementan *flooding* de los LSAs. Cada paquete de LSU lleva una colección de LSAs un salto más allá de su origen. Se pueden incluir varios LSAs en un solo paquete.

Los paquetes de LSU son *multicast* en aquellas redes físicas que admiten *multicast/broadcast*. Para que el procedimiento de *flooding* sea confiable, los LSAs inundados son reconocidos en los paquetes de LSAck. Si es necesaria la retransmisión de ciertos LSAs, los LSAs retransmitidos siempre son transportados por los paquetes de LSU de *unicast*. El cuerpo del paquete LSU consiste en una lista de LSAs [[70], [71]].

2.3.4.13.5 Paquetes Link State Acknowledgment

Los paquetes de LSAck son paquete OSPF tipo 5. Para que el *flooding* de los LSAs sea confiable, los LSAs inundados se reconocen explícita o implícitamente. El reconocimiento explícito se logra a través del envío y recepción de paquetes de LSAck. Múltiples LSAs pueden ser reconocidos en un único paquete de LSAck. Dependiendo del estado de la interfaz de envío y el remitente del paquete LSU correspondiente, se envía un paquete de LSAck a la dirección de *multicast AllSPFRouters*, a la dirección de *multicast AllDRouters* o a la dirección de *broadcast* de un vecino. El formato de este paquete es similar al del paquete de *Data Description* (DD). El cuerpo de ambos paquetes es simplemente una lista de encabezados de LSA. Cada LSA reconocida se describe por su encabezado LSA. Contiene toda la información requerida para identificar de manera única tanto la LSA como la instancia actual de la LSA [[70], [71]].

2.3.4.14 Formatos de los paquetes OSPFv3

En la Figura 16 se muestran los formatos de los encabezados OSPFv3 y los diferentes tipos de paquetes de OSPFv3 [70].

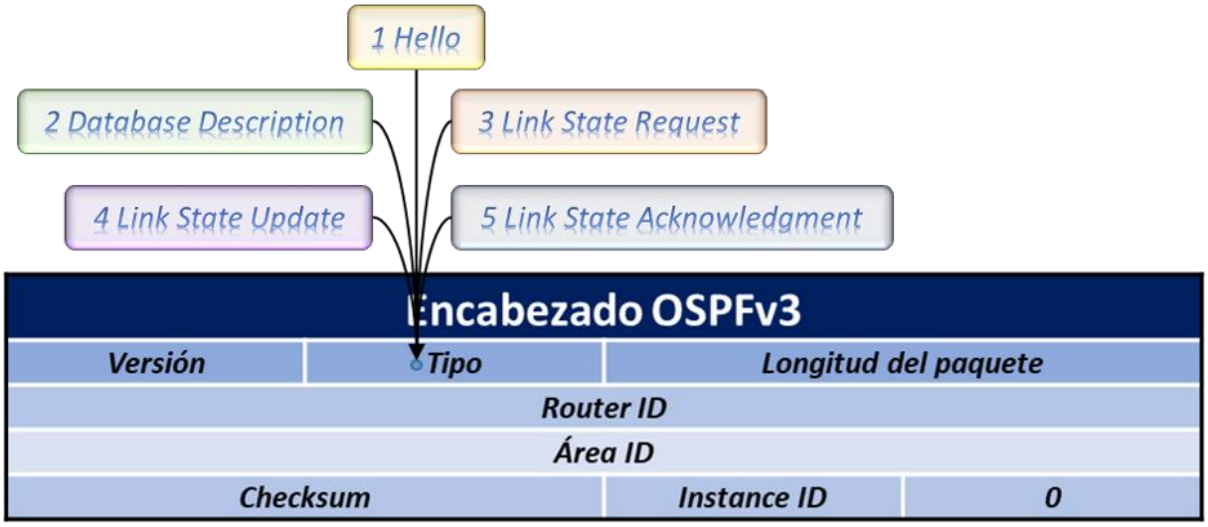


Figura 16 Formato de Encabezados OSPFv3 y tipos de paquetes OSPFv3 con base en [70]

2.4 BGP - Border Gateway Protocol

BGP es un protocolo de enrutamiento del sistema inter-autónomo, diseñado en 1989 (BGP-1) para ser escalable. La función principal de un sistema BGP es intercambiar información de alcance de red con otros sistemas BGP. Esta información es suficiente para construir un gráfico de conectividad entre AS. BGP usa TCP como su protocolo de transporte. Esto elimina la necesidad de implementar la fragmentación explícita de la actualización, la retransmisión, el acuse de recibo y la secuencia. BGP usa el puerto TCP 179. El mecanismo de notificación de errores utilizado en BGP supone que TCP admite un cierre "elegante" (es decir, que todos los datos pendientes se entregarán antes de que se cierre la conexión) [[77], [78]]. Los dos *routers* que forman una conexión TCP para intercambiar información de enrutamiento BGP son *peers* o vecinos. Los *peers* BGP intercambian inicialmente las tablas de enrutamiento BGP completas. Después de este intercambio, los *peers* envían actualizaciones graduales como los cambios de tabla de enrutamiento. BGP guarda un número de versión de la tabla de BGP. El número de versión es el mismo para todos los *peers* BGP. El número de versión cambia cada vez que BGP actualiza la tabla con cambios de información de enrutamiento. El envío de paquetes *keepalive* garantiza que se mantenga activa la conexión entre los *peers* BGP. Los paquetes de notificación se envían en respuesta a errores o condiciones especiales [78].

Antes de intercambiar cualquier tipo de mensaje de BGP, primero se debe establecer una sesión TCP entre los *routers* que quieren formar la adyacencia, una vez establecida la sesión se comienza el intercambio de mensajes [[77]-[79]]:

- *Open*: Para abrir una conexión TCP
- *Update*: Notificar o confirmar una nueva ruta.
- *Keepalive*: Si no hay UPDATES, se usa para mantener abierta la conexión TCP; También se usa como ACK para un mensaje OPEN
- *Notification*: Informar de errores en mensajes; también se utiliza para cerrar conexiones.

2.4.1 Atributos de ruta

Los atributos de ruta BGP es el modo en que se gestiona la red para elegir la ruta que deben tomar los paquetes para satisfacer determinadas necesidades o características de un ISP. Se definen características para el tráfico saliente y para el entrante, siendo este último algo más difícil de controlar. De modo que esta gestión de la red se hace a partir de la selección de las rutas que cualquier *router* va a propagar en una red y de las rutas que va a escoger como preferentes y alternativas [78].

Para ello se cuenta con un conjunto de atributos que dan información para la toma de decisión para filtrar o seleccionar rutas. Se definen a continuación dichos atributos en la Tabla 7 y 8:

- *Well know*
    - Son conocidos por todos los *routers* y pasados a los vecinos BGP.
    - Obligatorio y se incluyen en los mensajes de *UPDATE*.

Nombre	Descripción
Well-known Mandatory	Atributo que todos los *routers* deben soportar y comprender. Siempre debe ser enviado a los vecinos.
1 Origin	Tipo de origen (IGP, EGP, o desconocido)
2 AS Path	Listado de AS's que ha atravesado la notificación.
3 Next Hop	Peer externo en el vecino AS
Well-known Discretionary	Atributo que todos los *routers* deben soportar y comprender. Él envió de este hacia los vecinos es opcional.
5 Local Preference	Métrica para que los vecinos internos lleguen a destinos externos (por *default* 100)
6 Atomic Aggregate	Incluye los AS's que se han eliminado debido a la agregación de rutas.

Tabla 7 Atributos Well Know

- *Optional*
  - Puede que no sea compatible con todas las implementaciones de BGP
  - El bit transitivo determina si un atributo opcional se pasa a los vecinos BGP

Nombre	Descripción
Optional Transitive	Este puede ser comprendido o no por el router local (si no está activa la característica). Siempre debe ser enviado a los vecinos.
7 Aggregator	ID y AS de resumen *router*
8 Community	Etiqueta de ruta
Optional Nontransitive	Atributo opcional. No son reenviados si no se reconoce localmente.
4 Multiple Exit Discriminator (MED)	Métrica para que los vecinos externos alcancen el AS local (valor predeterminado 0)
9 Originator ID	El originador de una ruta reflejada.
10 Cluster List	Lista de IDs de clúster
13 Cluster ID	Clúster originador

Tabla 8 Atributos Optional

2.4.2 Selección de ruta

Todos estos atributos pueden ser utilizados conjuntamente para la selección de rutas, sin embargo, se debe imponer un orden de preferencia de manera que si se tienen varias rutas

que pueden ser preferente solo se elija una. Se recorrerá la siguiente lista y se eliminarán las rutas que no empatan en el mejor valor de cada uno de los criterios. Se ha de tener en cuenta que los criterios de decisión de enrutamiento que incluyen normas de desempate se aplican a cada prefijo IP o conjunto de prefijos IP destino [78]. En la Tabla 9 se describen los diferentes atributos de BGP, así como su nivel de preferencia.

Atributo	Descripción	Preferencia*
1 Weight	Preferencia administrativa	*Highest*
2 Local Preference	Comunicación entre *peers* dentro de un AS	*Highest*
3 Self-originated	Prefiere caminos originados localmente	*True*
4 AS Path	Minimizar los saltos de AS	*Shortest*
5 Origin	Prefiere las rutas aprendidas por IGP sobre EGP, y EGP sobre desconocidas	*IGP*
6 MED	Usado externamente para ingresar un AS	*Lowest*
7 External	Prefiere las rutas eBGP sobre iBGP	*eBGP*
8 IGP Cost	Considere la métrica de IGP	*Lowest*
9 eBGP Peering	Favorecer rutas más estables	*Oldest*
10 Router ID	Desempate	*Lowest*

Tabla 9 Selección de ruta. *Highest= *Más alto,* True= *Cierto,* Shortest= *El más corto,* IGP, Lowest= *Más bajo,* eBGP, Oldest= *Más antiguo [78].*

2.4.3 Estados vecinos

BGP pasa por varios estados en el momento en el que encuentra otro peer estos estados se muestran en la Tabla 10:

Estados Vecinos	
Inactivo (*Idle*)	El vecino no responde
Activo (*Active*)	Intentando conectar
Conectar (*Connect*)	sesión TCP establecida
Abrir Enviado (*Open Sent*)	Abrir mensaje enviado
Abrir Confirmar (*Open Confirm*)	Respuesta recibida
Establecido (*Established*)	Adyacencia establecida

Tabla 10 Estados Peers [[80], [81]].

En la Figura 17 se presenta el diagrama de estados BGP descritos en la Tabla 12. Iniciando con un estado *Idle* posteriormente pasando a *Connect* en donde uno de los extremos intenta una

conexión TCP, el estado *Active* que es cuando uno de los extremos no puede establecer conexión y lo reintenta periódicamente, pasando de nuevo a *Idle*, en *OpenSent* un extremo envía un mensaje de identificación en caso de que uno de los extremos no puede establecer conexión y lo reintenta periódicamente siendo de nuevo el estado *Active*, en caso de no obtener respuesta de confirmación se regresara a *Idle*. En el estado *OpenConfirm* se recibe respuesta al mensaje de identificación generando el sexto estado *Established* en el cual se acepta la identificación, siendo en este punto cuando la sesión es considerada completamente activa.

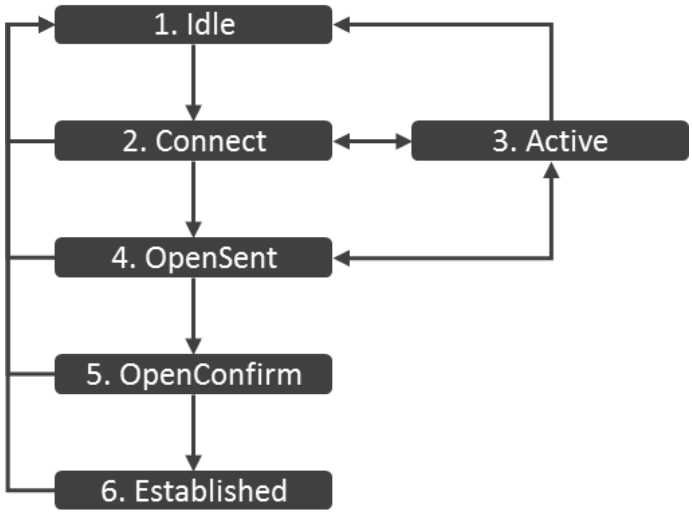


Figura 17 Diagrama de los Estados de BGP [80]

2.4.4 BGP para IPv6

La versión actual de BGP es la versión 4 es decir BGP-4, para IPv4 se describe en RFC 4271, para IPv6 no se generó una nueva versión, sino que se realizó una extensión al protocolo con el RFC 4760, el cual, permite a BGP transportar información de enrutamiento de protocolos distintos de IPv4, por ejemplo, MPLS (*Multiprotocol Label Switching*), IPv6, *Unicast*, *Multicast,* etc [[78], [80], [81]]. Definiendo para ello identificadores de dirección de información familiar (AFI - *Address Family Information*) y subsecuentes (Sub-AFI). Las familias de direcciones de BGP se presentan en la Tabla 11.

AFI/sAFI No.	Nombre
AFI=0	Reservado
AFI=1	IPv4
AFI=2	IPv6
Sub-AFI = 1	Unicast
Sub-AFI = 2	Comprobación de *Multicast* para Reenvío de ruta inversa (RPF - *Reverse Path Forwarding*)
Sub-AFI = 3	Para *Unicast* y *Multicast*
Sub-AFI = 128	Red privada virtual (VPN - *Virtual Private Network*)

Tabla 11 Familias de Direcciones de BGP [[80], [81]]

2.4.5 iBGP y eBGP

Los AS BGP intercambian información de enrutamiento de forma dinámica a través de sesiones de *peering* externas BGP (eBGP). Los *BGP speaker* dentro del mismo AS pueden intercambiar información de enrutamiento a través de sesiones de *peering* internas de BGP (iBGP), en la Tabla 12 se muestran las características de eBGP e iBGP, indicando la distancia administrativa siendo el primer criterio que un *router* usa para determinar qué protocolo de enrutamiento debe utilizar [[80], [81]].

Tipos BGP	
eBGP	**iBGP**
Distancia Administrativa = 20	Distancia Administrativa = 200
Peering entre *routers* en diferentes AS	*Peering* entre *routers* en el mismo AS
TTL = 1 (significa que solo los *peers* conectados directamente pueden comunicarse entre ellos)	TTL = 255 (significa que los *peers* que están a saltos el uno del otro todavía puede comunicarse entre ellos)
NO requiere topología de malla completa	Se requiere topología de malla completa (o cualquier remedio para ello, por ejemplo, reflectores de *routers*, confederaciones, ...)
Next-Hop se cambia a *Local Router* en notificaciones para vecinos	*Next-Hop* NO se cambia a *Local Router* en notificaciones a vecinos
Mecanismos de prevención de bucles: *AS-PATH*	Mecanismos de prevención de bucles: *Split Horizon*
AS se agrega al atributo *AS-PATH* en su notificación.	AS NO se agrega al atributo *AS-PATH* en su notificación.
No envía los atributos *LOCAL-PREF* en notificaciones	Envía los atributos *LOCAL-PREF* en notificaciones.

Tabla 12 Tipos BGP

Un ejemplo de eBGP e iBGP se muestra en la Figura 18.

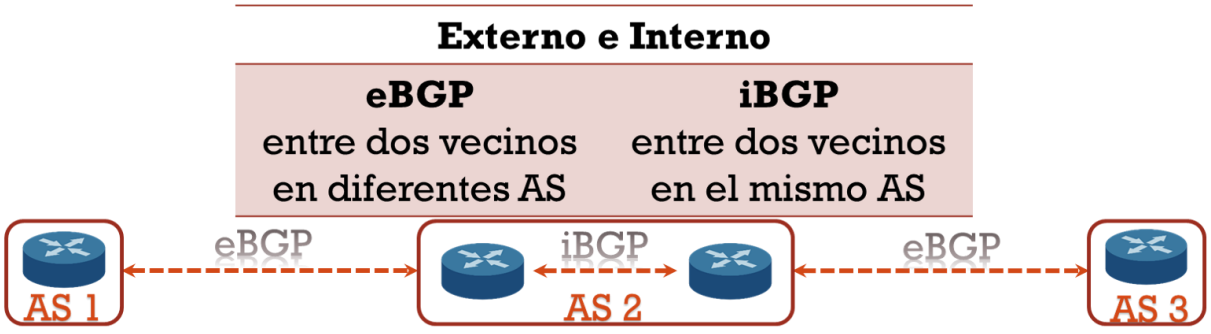


Figura 18 Ejemplo de iBGP y eBGP

Si un AS particular tiene múltiples *routers* BGP y está brindando servicio de tránsito para otros ASes, entonces se debe tener cuidado para asegurar una visión consistente del enrutamiento dentro del AS. El IGP utilizado dentro del AS proporciona una vista consistente de las rutas interiores del AS [[80], [81]]. Los códigos de origen BGP se presentan en la Tabla 13.

Códigos de origen BGP	
Código	**Detalle**
i	IGP: Ruta inyectada en las tablas BGP utilizando la declaración de red
e	EGP: Ya no se usa
?	Incompleto: Redistribuido en BGP (ya sea desde un protocolo de enrutamiento estático o dinámico)

Tabla 13 Códigos de origen BGP

2.4.6 Encabezado BGP

En la Figura 19 se muestra el formato de BGP con sus campos [[80], [81]].

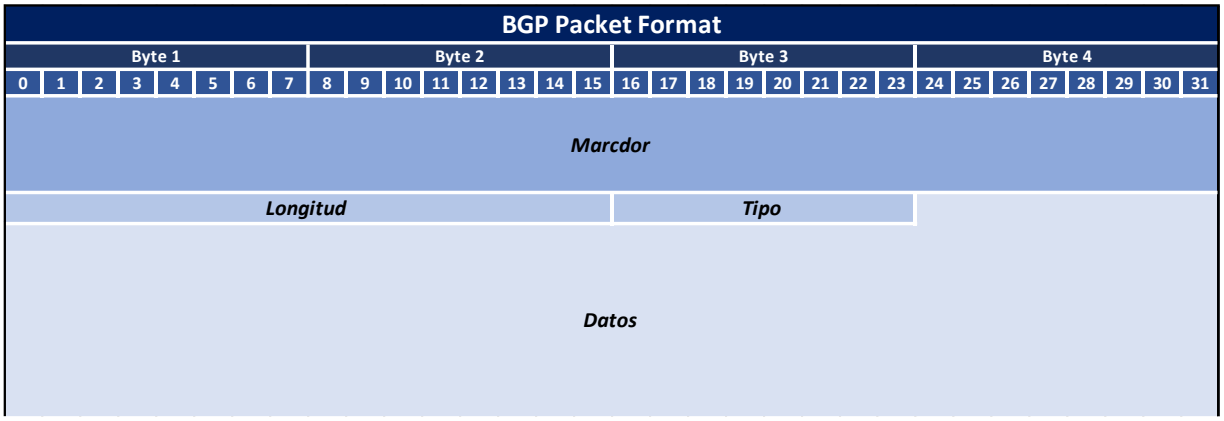


Figura 19 Formato de BGP

Los campos BGP se componen de acuerdo a la Tabla 14.

Campo	Descripción
Marcador	Contiene una secuencia que puede ser precedida por el peer remoto. • (en desuso) Autenticar los mensajes BGP recibidos en caso de usar autenticación. • Secuencia de unos (binario). Detectar perdida de sincronización en caso de no usar autenticación de mensajes.
Longitud	Largo total del mensaje incluido el encabezado.
Tipo	*Open*, *Update*, *Keepalive*, *Notification*.

Tabla 14 Campos del formato de BGP

2.5 Protocolos de Gestión

Gestionar una red implica monitorear, controlar y alterar la configuración de los dispositivos de dicha red. Esta gestión es realizada por un administrador de red el cual hace uso de herramientas, aplicaciones y metodologías para mantener una red optima.

Para esta tesis se utilizó el protocolo simple de administración de red (SNMP - *Simple Network Management Protocol*).

2.6 SNMP - Simple Network Management Protocol

SNMP es un protocolo estandarizado de la capa de aplicación que se utiliza para recopilar y organizar la información de los dispositivos en una red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el desempeño de la red, buscar y resolver sus problemas, y planear su crecimiento. Podemos conocer datos internos de los dispositivos que se monitorean, por ejemplo, el uso de CPU, memoria, Disco, *Uptime* (tiempo activo), etc. SNMP opera a través de protocolos como el *User Datagram Protocol* (UDP), *Internet Protocol* (IP), *Connectionless Oriented Network Service* (CLNS), *Datagram Delivery Protocol* (DDP) y *Novell Internet Packet Exchange* (IPX).

SNMP cuenta con el soporte de varias plataformas comerciales de gestión de red multifabricantes, como *OpenView* de HP, *SunNet* de *Sun Microsystems* o *NetView* de IBM.

2.6.1 Historia y evolución SNMP

Durante los primeros días de *ARPANET* se utilizaban comandos como *ping* para tratar de detectar algún problema en la red, aunque esta información era un tanto insuficiente, además que la escala y el alcance de Internet iban incrementando a gran velocidad, esto provocó la necesidad de una red de gestión en común, para ello, se propusieron algunas iniciativas para el desarrollo de la gestión de una red. A finales de 1987, tres de estas iniciativas se postularon, las cuales fueron: **HEMS** (*High-Level Entity Management System*), **SGMP** (*Simple Gateway Monitoring Protocol*) y el **CMIP** (*Common Management Information Protocol*) [[83], [84]].

Dado que SGMP tenía una implementación avanzada y con una relativa sencillez, fue llevado a una especificación más completa que en 1988 se presentó como **SNMP** (*Simple Network Management Protocol*) que es un protocolo del IETF, definido bajo el RFC 1067 el cual está especialmente diseñado para la administración de redes complejas y de múltiples

proveedores. Con el pasar de los años SNMP ha ido evolucionando mejorando varios aspectos. Un ejemplo de estas evoluciones las podemos ver en la versión 1 de SNMP, donde en 1988 aparecen los primeros RFC's, el RFC 1065, el cual describe las estructuras comunes y el esquema de identificación para la definición de información de gestión utilizada en la gestión de Internet basado en TCP/IP; el RFC 1066, proporciona la versión inicial de la Base de Información para Gestión (MIB - *Management Information Base*) para su uso con protocolos de gestión de red en internet basados en TCP/IP a corto plazo y el RFC 1067, define un protocolo simple por el cual la información de gestión para un elemento de red puede ser inspeccionada o alterada por usuarios lógicamente remotos [[84]-[86]]. Cada RFC ha sido actualizado por nuevos RFC's como se muestra en la Figura 20.

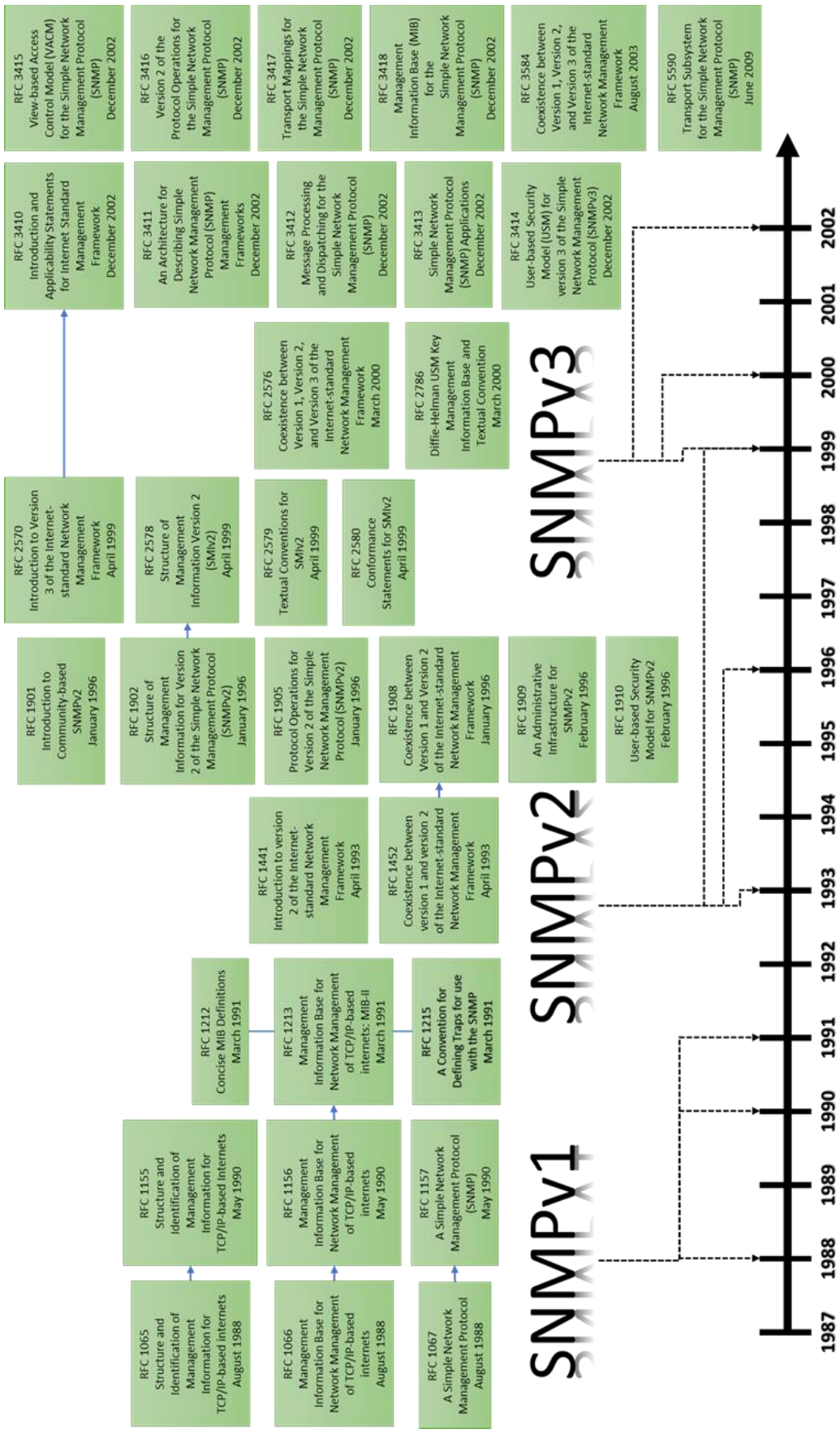


Figura 20 Evolución SNMPv1, SNMPv2 y SNMPv3 [[83]-[117]]

SNMP se desarrolló para permitir que los administradores puedan gestionar los nodos, como los servidores, las estaciones de trabajo, los *routers*, los *switches* y dispositivos de seguridad, en una red. Además, SNMP permite que los administradores:

- Monitoreen el rendimiento de la red
- Detecten y resuelvan problemas de red
- Planifiquen el crecimiento de la red

2.6.2 Elementos SNMP

SNMP cuenta con tres elementos para su funcionamiento:

- Administrador de SNMP
- Agente SNMP
- Dispositivos Administrados SNMP

El administrador SNMP forma parte de un Sistema de Administración de Red (NMS - *Network Management System*), ejecuta el software de SNMP. Los dispositivos gestionados, tienen como huésped un agente, el cual cumple con las tareas de suministrar información de gestión (MI – *Management Information*) respecto del dispositivo y aceptar instrucciones por parte del administrador para configurar un dispositivo [82]. El agente y la MIB residen en los clientes de dispositivo de red. Los dispositivos de red que se deben administrar, como *switches*, *routers*, servidores, *firewalls* y estaciones de trabajo, cuentan con un módulo de software de agente SMNP. La MIB almacena datos sobre el funcionamiento del dispositivo y están diseñadas para estar disponibles para los usuarios remotos autenticados. El agente SNMP es responsable de proporcionar acceso a la MIB local que refleja los recursos y la actividad de los objetos [[84], [89]].

El administrador de SNMP puede recopilar información de un agente SNMP mediante una acción *get* y puede cambiar la configuración en un agente mediante la acción *set*. Además, los agentes SNMP pueden reenviar información directamente a un NMS mediante *traps*, las cuales son notificaciones, que informan sobre los eventos que suceden en el dispositivo administrado es decir se usan para capturar errores e indicar dónde se encuentran, en la Figura 21 se muestra un diagrama de ello. SNMP utiliza el puerto 161 para enviar y recibir mensajes, los mensajes *trap* enviados por los agentes son recibidos en el puerto 162 [89].

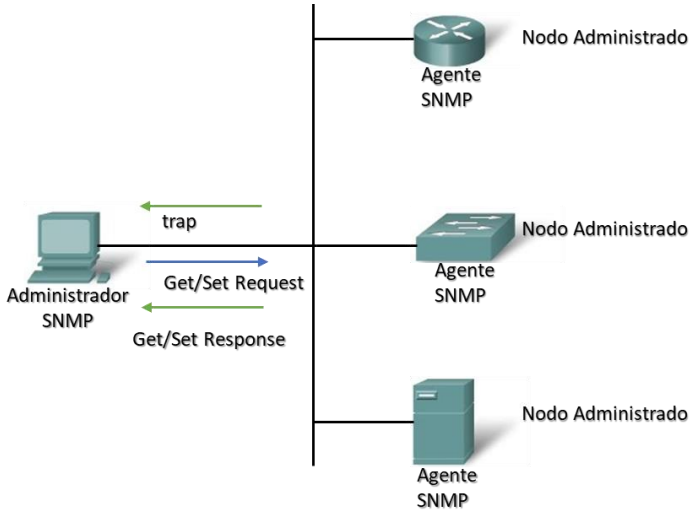


Figura 21 Diagrama de la arquitectura SNMP dentro de una red IP [89]

En la Tabla 15 se explican los comandos de SNMP que son utilizados por las herramientas de monitoreo:

Comandos SNMP	Origen - Destino	Detalles
TRAP	Agente --> Administrador	Notificación de eventos del equipo
GET	Administrador --> Agente	Consulta por un valor
GET-REQUEST	Administrador --> Agente	Recupera un valor de una variable específica
GET-NEXT-REQUEST	Administrador --> Agente	Recupera un valor de una variable dentro de una tabla; el administrador de SNMP no necesita saber el nombre exacto de la variable. Se realiza una búsqueda secuencial para encontrar la variable necesaria dentro de una tabla
GET-BULK-REQUEST	Administrador --> Agente	Recupera grandes bloques de datos, como varias filas en una tabla, que de otra manera requerirían la transmisión de muchos bloques pequeños de datos.
GET-RESPONSE	Agente --> Administrador	Respuesta a GET/SET/NEXT/BULK o error.
GET-BULK	Administrador --> Agente	Solicitud de *GetNext* múltiple
GET-NEXT	Administrador --> Agente	Consulta para el siguiente valor
INFORM	Administrador --> Agente	Confirmación de recibir el mensaje
SET	Administrador --> Agente	Establecer un valor o realizar una acción
SET-REQUEST	Administrador --> Agente	Almacena un valor en una variable especifica

Tabla 15 Comandos SNMP

Los agentes SNMP que se encuentran instalados en los sistemas administrados cumplen con la función de recopilar y almacenar información sobre los sistemas y su funcionamiento.

2.6.3. MIB - Management Information Base

La MIB es una colección de información organizada jerárquicamente. Se accede a esta información mediante SNMP. Hay dos tipos de MIB, escalar y tabular [[84], [85], [87]]:

- Los objetos **escalares** definen una única instancia de objeto
- Los objetos **tabulares** definen varias instancias de objeto relacionadas y que están agrupadas en las tablas de la MIB.

El administrador SNMP usa al agente SNMP para tener acceso a la información dentro de la MIB, en la Figura 22 se presenta el orden del proceso de una MIB.

Figura 22 Proceso de componentes de SNMP de una MIB

De manera general el agente SNMP responde a las solicitudes del administrador de SNMP de la siguiente manera:

- **Obtener una variable de MIB:** el agente SNMP realiza esta función en respuesta a una unidad de datos de protocolo (PDU - *Protocol Data Units*) de solicitud *get* del NMS. El agente recupera el valor de la variable de MIB solicitada y responde al NMS con ese valor.
- **Establecer una variable de MIB:** el agente SNMP realiza esta función en respuesta a una PDU de solicitud *set* de NMS. El agente SNMP cambia el valor de la variable de MIB por el valor que especifica el NMS. La respuesta del agente SNMP a una solicitud *set* incluye la nueva configuración en el dispositivo.

La MIB almacena la información de los identificadores de objetos.

2.6.4. OID - Object Identificator

Un dispositivo administrado SNMP tendrá “objetos” para el nombre del dispositivo, el tiempo de actividad del dispositivo, interfaces del dispositivo y tabla de enrutamiento para el caso de *routers*, por nombrar algunos de estos objetos. A cada objeto se le asigna un identificador de objeto (OID).

Los *OID* (*Object Identificator*) identifican de forma única los objetos gestionados de manera jerárquica en la MIB mediante una secuencia de números para identificar dichos objetos, se pueden representar como un árbol, cuyos niveles son asignados por diferentes organizaciones. Los *OID* de nivel superior pertenecen a diferentes organizaciones estándar. Los proveedores definen las ramas privadas, incluidos los objetos gestionados para sus propios productos [117].

2.6.5 Diagrama de árbol de una MIB

El diagrama de árbol de una MIB para un sistema o dispositivo determinado, incluye algunas ramas con variables comunes a varios dispositivos de red y algunas ramas con variables específicas de ese dispositivo o proveedor [103].

Para entender el detalle de una trama SNMP, lo mejor es verla como un conjunto de campos anidados. El principal fragmento de información es el OID, que identifica exactamente el valor a leer (*get*) o a escribir (*set*). El conjunto de OIDs que tiene disponible un dispositivo se conoce como MIB y es como un índice en forma de árbol donde podemos encontrar la referencia que buscamos. Los tipos de datos de los OID están definidos por ASN.1 y usan una codificación específica denominada BER (*Basic Encoding Rules*) [117]. Un ejemplo de la representación del diagrama de árbol de una MIB y de OID se muestra en la Figura 23 mostrando la ruta a un dispositivo Cisco con un OID 1.3.6.1.4.5.

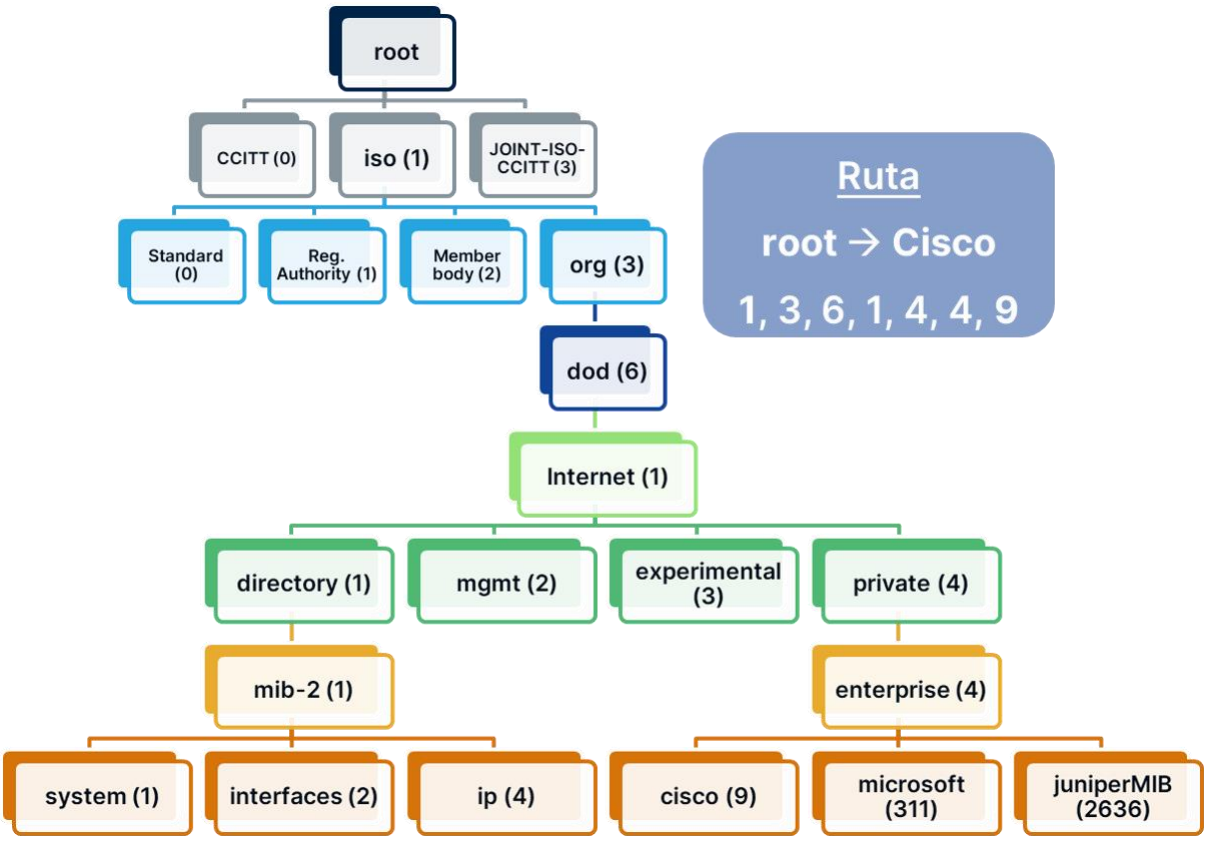


Figura 23 Árbol representativo de MIB y representación de OID

2.6.6 Versiones SNMP

Hasta el momento existen tres versiones del protocolo: SNMPv1 que es la implementación inicial de SNMP; SNMPv2 que incluyó mejoras a la implementación inicial especialmente el uso de cadenas comunitarias (*community strings*) que es agregar autenticación ya sea a nivel de lectura o escritura. SNMPv3, no realiza cambios en el protocolo, aparte de la adición de seguridad criptográfica.

2.6.6.1 SNMPv1

Diseñado a finales de los 80. Su objetivo era lograr una solución temporal hasta la llegada de protocolos de gestión de red con mejores diseños y más completos. Un estándar de Internet completo, definido mediante el RFC 1157. SNMPv1 no estaba pensado para una gran cantidad de redes que cada día iban incrementando, en la Figura 24 se muestra el formato de SNMPv1 [[86], [87], [89]].

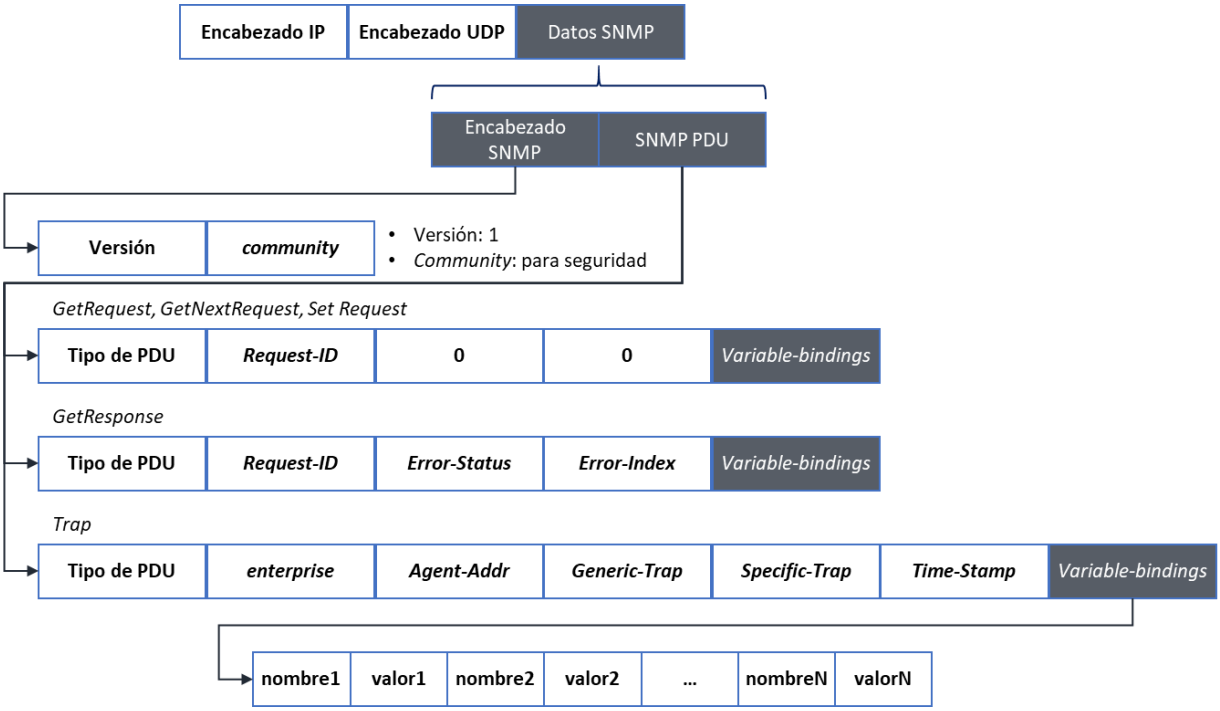


Figura 24 SNMPv1 PDUs con base en [[86], [87], [89]]

- *Request-ID*: Para correlacionar peticiones y respuestas; se usa el mismo valor en ambas.
- *Error-Status*, *Error-Index*: Gestión de errores.

Notificaciones

- *Enterprise*: Tipo de objeto que la genera, según *sysObjectID*.
- *Agent-Addr*: Dirección del objeto que la genera.
- *Generic-Trap*: Tipos y códigos.
- *Time-Stamp*: Desde el último reinicio de la red. Contiene *sysUpTime*.

Operaciones

- Monitoreo con PDUs de lectura (*GetRequest*, *GetNextRequest*, *Trap*).
- Control con PDU de escritura (*SetRequest*). Permite ejecutar comandos modificando el valor de algunos objetos específicos.

GetRequest, *GetNextRequest*, *SetRequest*

- Se confirman con *GetResponse*
- Pueden operar sobre más de una variable
- Son operaciones atómicas: si fallan en alguna variable no actúan sobre ninguna

Trap

- No se confirman con *GetReponse*

GetNextRequest

- Para cada variable se obtiene el siguiente nodo-hoja de la MIB en orden lexicográfico.
- Permite descubrir la estructura de la MIB de forma dinámica.

Concepto de *community*: conjunto de agentes y gestores que actúan sobre los primeros.

Aspectos de Seguridad

- Autenticación basada en el nombre de la comunidad. Es un *password* en texto claro que se comparte, típicamente *public* o *private* por *default*. Esquema muy pobre.
- Acceso basado en vistas (*MIB view*) y modos (ACCESS MODE ej. *Read-only*).
- Se combinan ambos en un *community profile*

2.6.6.2 SNMPv2

Definido en 1993 y revisado en 1995. Se le añade a SNMPv1 mecanismos de seguridad, aunque no se implementaron por completo por lo que quedaron de cierto modo en teoría, mayor detalle en la definición de variables. También se le añaden estructuras de la tabla de datos para facilitar el manejo de los datos. Se define con los RFC 1901 a 1908; utiliza el marco administrativo basado en *Community String* contraseñas para acceso SNMP a dispositivos de red [[95] - [98]].

Cambios respecto a SNMPv1

- Orientación a redes distribuidas, gracias a la comunicación entre administradores.
- Mayor eficiencia en tráfico mediante mensajes de obtención de múltiples valores.

Seguridad mejorada SNMPv2c (c=*community*).

- Aparece SMIv2 y una nueva MIB: 1.3.6.1.6
- Se permite crear y borrar filas en tablas.

No es compatible con SNMPv1, debido a cambio en formato de cabecera, pero se pueden emplear agentes proxy y entornos duales.

Modificaciones respecto a SNMPv1

- Cabeceras de los mensajes como en *Secure SNMP*, en la Figura 25 se muestra la cabecera de *Secure* SNMP.

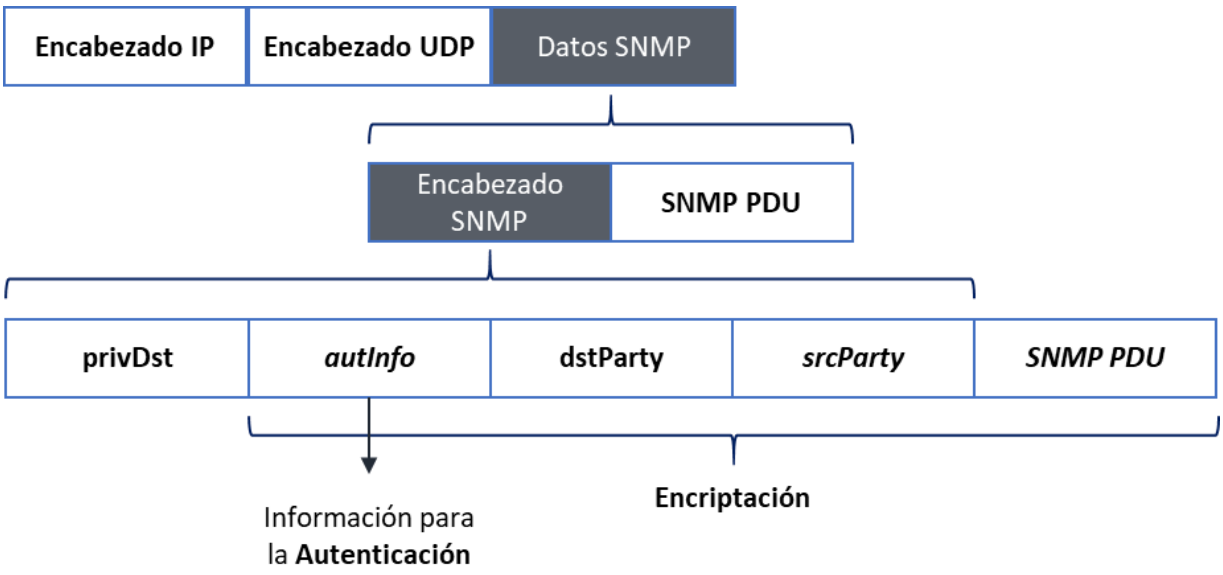


Figura 25 Cabecera de Secure SNMP con base en [[95] - [98]].

- PDUs
  - Nuevas: *GetBulkRequest*, *InformRequest*
  - Modificada *GetResponse* por Response
  - Modificada estructura de *Trap*, *GetRequest*, *GetNextRequest*, *SetRequest*, *InformRequest*, en la Figura 26 se muestra la estructura PDU de SNMPv2.

GetRequest, GetNextRequest, SetRequest, Trap, InformRequest

Tipo de PDU	*Request-ID*	0	0	*Variable-bindings*

Figura 26 PDU de SNMPv2 con base en [[95] - [98]].

  - *GetRequest*, *GetNextRequest* y *SetRequest* no son atómicas, pudiendo actuar sobre algunas (y no todas) las variables del mensaje.

- Hay tráfico de notificaciones entre NMSs

GetBulkRequest

- Se usa para minimizar el número de intercambios necesario para obtener una gran cantidad de datos.
- PDU, en la Figura 27 se muestra el formato del PDU *GetBulkRequest*

		N	*M*	*N+R variables*
Tipo de PDU	*Request-ID*	*Non-Repeaters*	*Max-Repetitions*	*Variable-bindings*

Figura 27 PDU GetBulkRequest con base en [[95] - [98]].

- Para las primeras N (*non-repeaters*) variables la operación es idéntica a *GetNextRequest* PDU, devolviéndose un único sucesor lexicográfico
- Para las otras R variables, se devuelven múltiples (M=*max-repetitions*) sucesores lexicográficos

InformRequest

- Semejante al *Trap*, pero requiere confirmación desde el administrador. Si esta no se recibe en un tiempo, se reenvía el *InformRequest*
- Criterios
  - Eficiencia de tráfico y recursos de memoria: *Trap*
  - Seguridad en recepción: *InformRequest*

2.6.6.3 SNMPv3

SNMP versión 3 proporciona acceso seguro a los dispositivos al autenticar y encriptar los paquetes de datos a través de la red. SNMPv3 es un protocolo interoperable basado en estándares que se define en las RFC 3410 a 3415 [[107] - [111]].

Cambios respecto a SNMPv2

- Seguridad en
  - Integridad de mensajes.
  - Confidencialidad, mediante la encriptación de paquetes.
  - Autenticidad.
  - Enmascaramiento: UDP es vulnerable al IP *spoonfing* (cambio de la dirección de origen) para suplantar dispositivos, y SNMPv3 contiene mecanismos para evitarlo.
- Nuevo *framework* o arquitectura extensible, compatible de forma nativa con SNMPv1 y SNMPv2.
- Nuevas MIBs bajo 1.3.6.1.6

2.6.6.3.1 Funciones de seguridad en SNMPv3

Las características de seguridad proporcionadas en SNMPv3 son las siguientes:

- Integridad del mensaje: garantiza que un paquete no se haya manipulado durante el tránsito.
- Autenticación: determina que el mensaje proviene de una fuente válida.

- Encriptación: codifica el contenido de un paquete para evitar que lo intercepte una fuente no autorizada.

SNMPv3 contiene un modelo de seguridad en el que se configura autenticación para un usuario y el grupo en el que reside el usuario. Una combinación de un modelo de seguridad y un nivel de seguridad determina qué mecanismo de seguridad se usa cuando se maneja un paquete SNMP. Se definen dos mecanismos de seguridad [[111], [112]]

- **USM** (*User-based Security Model*)
  - Proporciona funciones de autenticaciones y confidencialidad (encriptación)
  - Opera a nivel de Mensaje
- **VACM** (*View-based Access Control Model*)
  - Determina si se permite el acceso a los objetos de la MIB para llevar a cabo diferentes acciones
  - Opera a nivel de PDU

En la Tabla 16 se muestran los niveles de seguridad de SNMPv3.

Niveles de Seguridad SNMPv3	
NoAuthPriv	Sin autenticación, sin privacidad
AuthNoPriv	Autenticación, sin privacidad
AuthPriv	Autenticación, con privacidad

Tabla 16 Niveles de seguridad SNMPv3

Los modelos de seguridad disponibles para SNMPv1, SNMPv2 y SNMPv3, se muestran en la Tabla 17.

Modelo	Nivel	Autenticación	Encriptación	Características
V1	*noAuthNoPriv*	*Community String*	No	Utiliza una coincidencia *community string* para la autenticación.
V2c	*noAuthNoPriv*	*Community String*	No	Utiliza una coincidencia *community string* para la autenticación.
V3	*noAuthNoPriv*	*Username*	No	Utiliza una coincidencia de nombre de usuario para la autenticación.
V3	*authNoPriv*	*MD5 o SHA*	No	Proporciona autenticación basada en los algoritmos HMAC-MD5 o HMAC-SHA.
V3	*authPriv*	*MD5 o SHA*	Estándar de cifrado de datos (DES) o estándar de cifrado avanzado (AES)	Proporciona autenticación basada en los algoritmos HMAC-MD5 o HMAC-SHA. Permite especificar el modelo de seguridad basado en usuarios (USM) con estos algoritmos: • Cifrado DES de 56 bits, además de autenticación basada en el estándar CBC-DES (DES-56). • Cifrado 3DES de 168 bits. • Cifrado AES de 128 bits, 192 bits o 256 bits.

Tabla 17 Características de seguridad en SNMP

2.6.6.3.2 SNMPv3: Arquitectura

Concepto *Entidad SNMP* que puede actuar como agente, administrador o ambos a la vez, según los módulos que implemente.

Esquema con base a dos capas [88]:

- Una o varias aplicaciones: capa superior que genera y recibe PDUs
- Un motor: capa inferior
    - Hace de intermediario para los PDUs de aplicaciones y las capas inferiores: versión de SNMP, protocolos, etc.

- Gestiona la seguridad: autenticación, encriptación y acceso.

	Administrador	Administrador/Agente	Agente
Aplicaciones	Generador de comandos	Generador de notificaciones	Comando de respuesta. Notificaciones de respuesta. Promotor de proxy.
Motor		**Administrador** Administrador de PDU Administrador de mensajes Mapeo de transporte (UDP) Subsistema de procesamiento de mensajes (v1, v2, v3) Subsistema de seguridad	Subsistema de acceso

Tabla 18 Arquitectura SNMPv3 con base en [108]

2.6.6.3.3 SNMPv3: Formato de Mensaje

El formato de mensaje de SNMPv3 se muestra en la Figura 28.

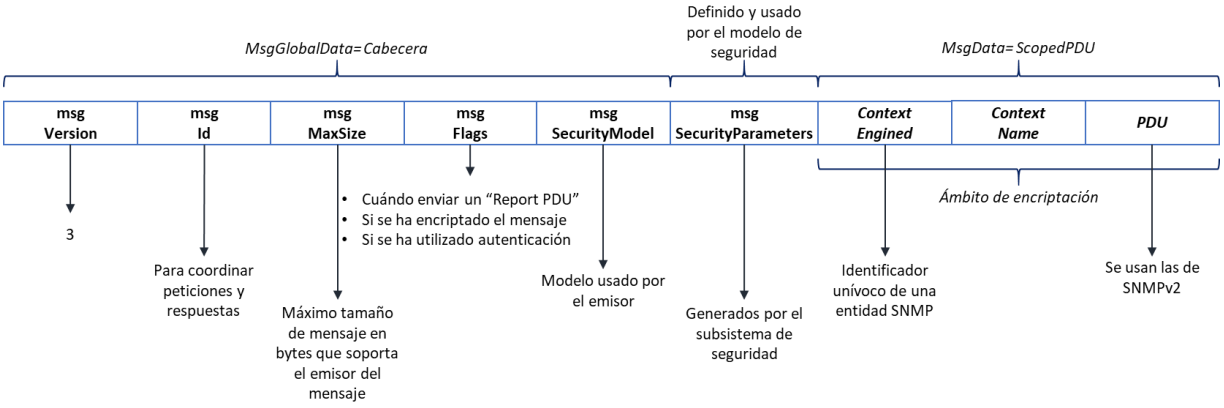


Figura 28 Formato de SNMPv3 con base en [[107] - [111]

Capítulo 3:

Metodología de la Emulación

3.1. Emulación de Pacific Wave

Este trabajo emuló la topología mostrada en la *Figura 2*, la cual comprende: el *Backbone* de PW (Seattle, Sunnyvale y Los Ángeles), así como AP-REX POP (Albuquerque, Chicago, Dallas, Denver, El Paso, Houston, Kansas City, Tijuana y Tulsa) el cual es un proyecto de PW e Internet2, estos a su vez conectan con WRN (ABQG, CENIC, FRGP, PNWGP y UH), también se encuentra la PRP (China, Guam, Corea, Singapur y WIDE) este proyecto utiliza a PW y CaiREN, también se utilizaron como nodos las extensiones de PW (Taiwán, Hong Kong, Oahu, Hawái, Australia y New Zealand) y se colocaron los AS que están directamente conectados con el *Backbone* de PW (CUDI, DREN, ESnet, Internet2, Los Nettos, NREN-NASA, Transtelco, UltraLight Caltech).

La asignación del número de AS de cada red está referido al ASN (*Autonomous System Number*) real de cada red, siendo estos como se presenta en la Tabla 19 [[118] - [120]].

AS	Número de AS	AS	Número de AS
AARNET	7575	**Los Nettos**	226
ABQG	40498	**NREN (NASA Ames)**	21556
CANARIE	6509	**PIREN**	1451058
CENIC	2153	**PNWGP**	101
CERNET	23910	**PRP**	395889
CSTNET	62429	**PW**	62819
CUDI	18592	**REANNZ**	38022
DREN	668	**SINET**	24744
ESnet	293	**SingAREN/Internet2**	136968/11164
FRGP	14041	**TransPAC**	22388
GEMnet	45204	**Transtelco**	32098
Internet2	11164	**TWAREN/TANET**	7539/1659
JGNet	263283	**UH**	10294
KISTI/KREONET	17579	**UltraLight_Caltech**	32361

Tabla 19 Sistema Autónomo y ASN correspondiente a las redes pertenecientes a la red Pacific Wave

Para la emulación se configuraron los *routers* de acuerdo al Apéndice B. Debido a la cantidad de interfaces requeridas para el *Backbone* de PW se utilizaron para Seattle 4 *routers*, Sunnyvale 2 *routers* y Los Ángeles 4 *routers*, contando con un total de 45 *routers*.

3.2. GNS3 y Equipo utilizado para emulación

La emulación se realizó mediante el sistema GNS3 (*Graphical Network Simulator*), el cual permite crear y diseñar una red en un entorno virtual. Para poder trabajar con dicho sistema se debe cumplir una serie de requerimientos los cuales se muestran en la Tabla 20 [121]**.** Al utilizar una herramienta que emula se reproduce el hardware de un equipo por completo.

	Requerimientos Mínimos	Requerimientos Recomendados
Sistema Operativo	Windows 7 (64 bit) y posterior, Mavericks (10.9) y posterior, Cualquier distribución Linux Debian/Ubuntu se proporcionan y son compatibles	Windows 7 (64 bit) y posterior, Mavericks (10.9) y posterior, Cualquier distribución Linux Debian/Ubuntu se proporcionan y son compatibles
Procesador	2 o más núcleos lógicos - Serie AMD-V / RVI o Intel VT-X / EPT - extensiones de virtualización presentes y habilitadas en el BIOS. Más recursos permiten una simulación más grande	4 o más núcleos lógicos - Serie AMD-V / RVI o Intel VT-X / EPT - extensiones de virtualización presentes y habilitadas en el BIOS. Más recursos permiten una simulación más grande
Memoria	4 GB RAM	8 GB RAM
Almacenamiento	1 GB de espacio disponible (la instalación de Windows es <200MB	SSD - 35 GB de espacio disponible
Notas adicionales	Se necesita más almacenamiento para el sistema operativo y las imágenes del dispositivo.	RAM adicional de hasta 16 GB e i7 o equivalente para un uso óptimo. La virtualización de dispositivos requiere mucho procesador y memoria.

Tabla 20 Requerimientos para el uso de GNS3 con base en [121]

Para la emulación se crearon:

- 45 *Routers* Cisco 7200 con imagen IOS 7200v15.2(4)M2 para ser utilizados como *Backbone*; cada uno con interfaces GBE (Gigabit Ethernet) e interfaces POS (*Packet Over SONET*).

- 3 Ethernet Switch.
- 1 máquina Virtual (Windows 10).
- 2 máquinas Virtuales (Centos 8).

El equipo utilizado durante esta tesis fue una laptop Lenovo modelo Y700, como se muestra en la Figura 29.

Figura 29 Equipo usado para la emulación

A continuación, se muestran las características del equipo que se utilizó para este trabajo

Características	Equipo
Sistema Operativo	Windows 10 Pro
Procesador	Intel® Core™ i7-6700HQ CPU @ 2.60GHz 2.60 GHz
Núcleos	4
Procesadores Lógicos	8
Memoria Instalada (RAM)	16.00 GB
Tipo de Sistema	Sistema Operativo de 64 bits, procesador x64

Tabla 21 Características de equipo utilizado

3.3 Metodología para emulación

La configuración en una red es algo fundamental dado que con ella se logra hacer que todas las interconexiones logren una comunicación óptima y escalable. Todas las configuraciones que se realizaron en esta red fueron bajo el protocolo de internet versión 6.

3.3.1 Configuración IPv6

Para habilitar la interfaz y el uso del protocolo IPv6 dentro de cada *router*, se accede a la interfaz de línea de comandos (CLI - *Command-Line Interface*) en modo súper usuario

```
> enable
```

Se ingresa al modo de configuración global (escribiendo en el prompt *#configure terminal*)

```
# configure terminal
```

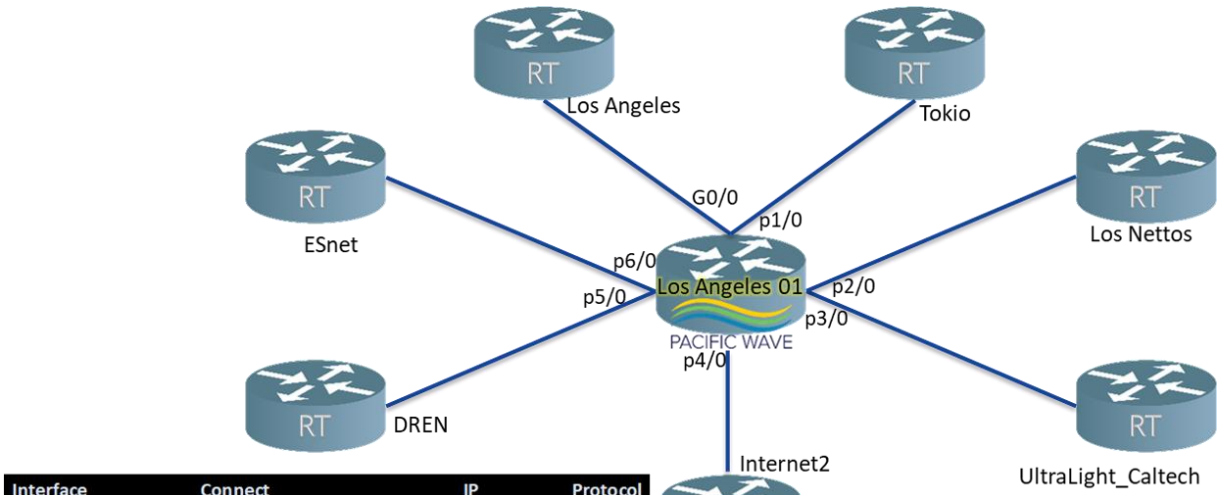
Se habilita el enrutamiento de IPv6 para rutas estáticas y protocolos dinámicos, reenvío de paquetes IPv6 y envío de mensajes ICMPv6. Así como el uso de *Cisco Express Forwarding* (CEF) para IPv6, el cual es una tecnología avanzada de conmutación de IP de capa 3, optimizando el rendimiento y la escalabilidad de la red para redes con patrones de tráfico dinámicos y topológicamente dispersos, como los asociados con aplicaciones basadas en web y sesiones interactivas.

```
(config)# ipv6 unicast-routing
(config)# ipv6 cef
```

Se selecciona la interfaz a configurar, indicando la dirección IPv6 que se utilizara y habilitándola.

```
(config)# interface <Tipo-Número>
(config-if)# no ip address
(config-if)# ipv6 enable
(config-if)# ipv6 address <ipv6-address/prefix-length>
(config-if)# no shutdown
(config-if)# exit
```

Estas configuraciones se realizaron en cada uno de los *routers* de la red de PW. Para ejemplificar se mostrará cómo se configuró el *router* de Los_Angeles_01, el cual tiene las configuraciones de IPv6, OSPFv3 y BGP-4. En la Figura 30 se muestran las conexiones a este *router,* así como sus tipos de interfaces, direcciones IPv6 y protocolo de enrutamiento empleado.

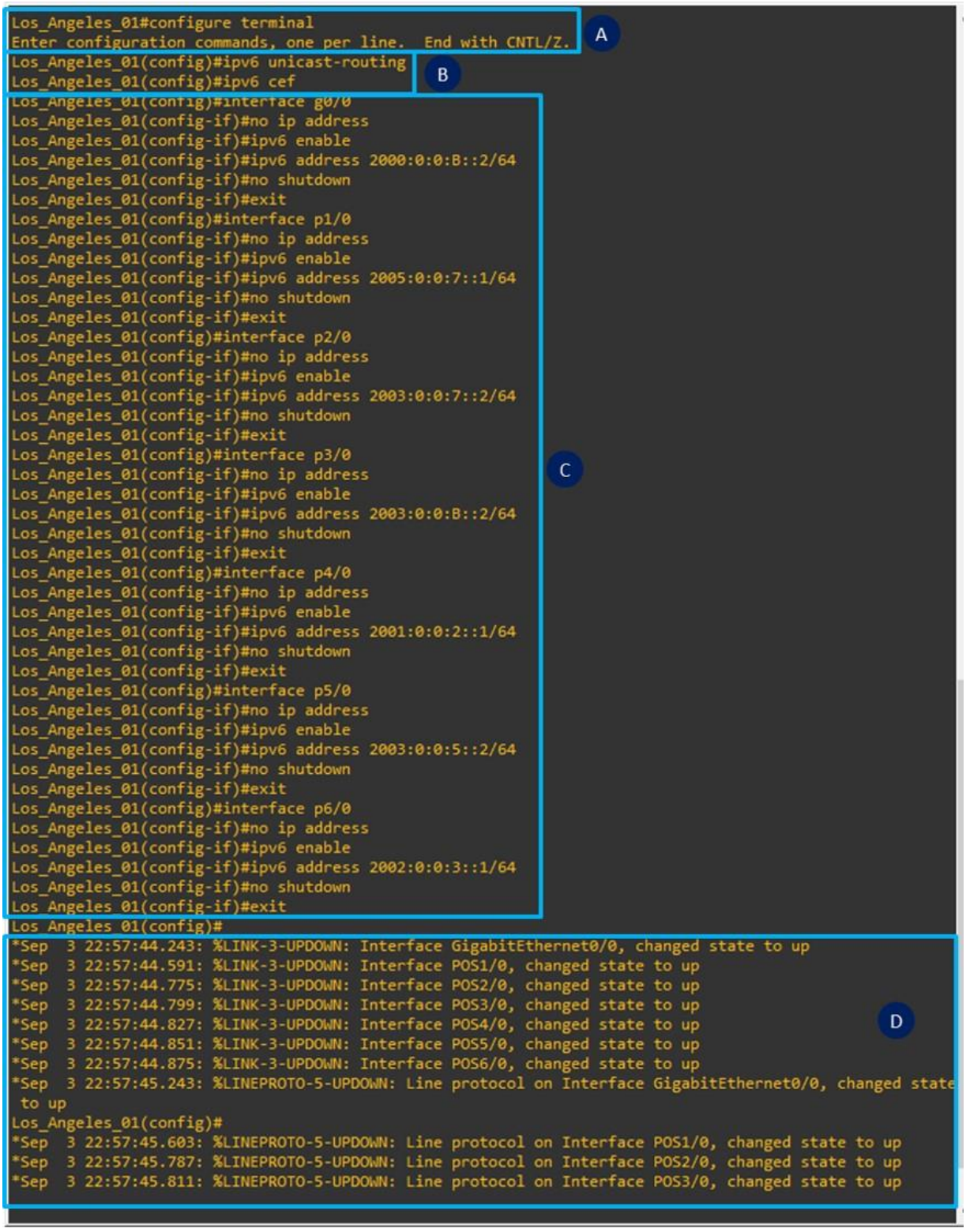


Interface	Connect	IP	Protocol
g0/0	Los_Angeles_(Pacific_Wave)	2000:0:0:B::2/64	OSPFv3
p1/0	Tokio_(Pacific_Wave)	2005:0:0:7::1/64	OSPFv3
p2/0	Los_Nettos	2003:0:0:7::2/64	BGP
p3/0	UltraLight_Caltech	2003:0:0:B::2/64	BGP
p4/0	Internet2	2001:0:0:2::1/64	BGP
p5/0	DREN	2003:0:0:5::2/64	BGP
p6/0	Esnet	2002:0:0:3::1/64	BGP

Figura 30 Topología de Los_Angeles_01

En la Figura 31, se muestran los pasos que se realizaron para esta configuración:

A. Se habilitó el modo de configuración
B. Se habilitó IPv6
C. Se asignó el direccionamiento del IPv6 e interfaz
D. Se muestra el estado de cada interfaz una vez configurado




```
Los_Angeles_01#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Los_Angeles_01(config)#ipv6 unicast-routing
Los_Angeles_01(config)#ipv6 cef
Los_Angeles_01(config)#interface g0/0
Los_Angeles_01(config-if)#no ip address
Los_Angeles_01(config-if)#ipv6 enable
Los_Angeles_01(config-if)#ipv6 address 2000:0:0:B::2/64
Los_Angeles_01(config-if)#no shutdown
Los_Angeles_01(config-if)#exit
Los_Angeles_01(config)#interface p1/0
Los_Angeles_01(config-if)#no ip address
Los_Angeles_01(config-if)#ipv6 enable
Los_Angeles_01(config-if)#ipv6 address 2005:0:0:7::1/64
Los_Angeles_01(config-if)#no shutdown
Los_Angeles_01(config-if)#exit
Los_Angeles_01(config)#interface p2/0
Los_Angeles_01(config-if)#no ip address
Los_Angeles_01(config-if)#ipv6 enable
Los_Angeles_01(config-if)#ipv6 address 2003:0:0:7::2/64
Los_Angeles_01(config-if)#no shutdown
Los_Angeles_01(config-if)#exit
Los_Angeles_01(config)#interface p3/0
Los_Angeles_01(config-if)#no ip address
Los_Angeles_01(config-if)#ipv6 enable
Los_Angeles_01(config-if)#ipv6 address 2003:0:0:B::2/64
Los_Angeles_01(config-if)#no shutdown
Los_Angeles_01(config-if)#exit
Los_Angeles_01(config)#interface p4/0
Los_Angeles_01(config-if)#no ip address
Los_Angeles_01(config-if)#ipv6 enable
Los_Angeles_01(config-if)#ipv6 address 2001:0:0:2::1/64
Los_Angeles_01(config-if)#no shutdown
Los_Angeles_01(config-if)#exit
Los_Angeles_01(config)#interface p5/0
Los_Angeles_01(config-if)#no ip address
Los_Angeles_01(config-if)#ipv6 enable
Los_Angeles_01(config-if)#ipv6 address 2003:0:0:5::2/64
Los_Angeles_01(config-if)#no shutdown
Los_Angeles_01(config-if)#exit
Los_Angeles_01(config)#interface p6/0
Los_Angeles_01(config-if)#no ip address
Los_Angeles_01(config-if)#ipv6 enable
Los_Angeles_01(config-if)#ipv6 address 2002:0:0:3::1/64
Los_Angeles_01(config-if)#no shutdown
Los_Angeles_01(config-if)#exit
Los_Angeles_01(config)#
*Sep  3 22:57:44.243: %LINK-3-UPDOWN: Interface GigabitEthernet0/0, changed state to up
*Sep  3 22:57:44.591: %LINK-3-UPDOWN: Interface POS1/0, changed state to up
*Sep  3 22:57:44.775: %LINK-3-UPDOWN: Interface POS2/0, changed state to up
*Sep  3 22:57:44.799: %LINK-3-UPDOWN: Interface POS3/0, changed state to up
*Sep  3 22:57:44.827: %LINK-3-UPDOWN: Interface POS4/0, changed state to up
*Sep  3 22:57:44.851: %LINK-3-UPDOWN: Interface POS5/0, changed state to up
*Sep  3 22:57:44.875: %LINK-3-UPDOWN: Interface POS6/0, changed state to up
*Sep  3 22:57:45.243: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state
 to up
Los_Angeles_01(config)#
*Sep  3 22:57:45.603: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS1/0, changed state to up
*Sep  3 22:57:45.787: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS2/0, changed state to up
*Sep  3 22:57:45.811: %LINEPROTO-5-UPDOWN: Line protocol on Interface POS3/0, changed state to up
```

Figura 31 Ejemplo de configuración inicial de un router habilitando ipv6 y direccionamiento

3.3.2 Configuración OSPFv3

La configuración del protocolo de enrutamiento OSPFv3, se inicia seleccionando el número de proceso para la activación de un proceso OSPF (es posible ejecutar varios procesos OSPF en el

mismo *router*, pero no se recomienda dado que crea múltiples instancias de base de datos que agregan una sobrecarga adicional al *router*) y asignando un número de identificación.

```
(config)# ipv6 router ospf <Proceso>
(config-rtr)# router-id <router-id>
```

En caso que un *router* utilice los protocolos de OSPFv3 y BGP es necesario habilitar la función de redistribución de OSPFv3.

```
(config-rtr)# log-adjacency-changes
(config-rtr)# redistribute connected
(config-rtr)# redistribute bgp <AS>
(config-rtr)# exit
```

Posteriormente se habilita la interfaz que utiliza OSPFv3.

```
(config)# interface <Tipo-Número>
(config-if)# ipv6 ospf <Proceso> area <# de Área>
(config-if)# exit
```

Continuando con el ejemplo del *router* de Los_Angeles_01 en la Figura 32 se muestra la configuración de OSPFv3.

E. Se seleccionó número de proceso y se asignó un número de identificación para cada *router*.
F. En este caso se habilitó la redistribución BGP dado que este *router* conecta con los AS de Los_Nettos, UltraLight_Caltech, Internet2, DREN y ESnet. Por lo que también se configuró BGP-4 el cual se muestra más adelante.
G. Se habilitó OSPFv3 para las interfaces que usaron este protocolo, en este caso únicamente la interfaz G0/0 la cual conecta con el *router* Los_Angeles.
H. Se muestra la adyacencia establecida con el *router* Los_Angeles con ID 1.1.1.7.

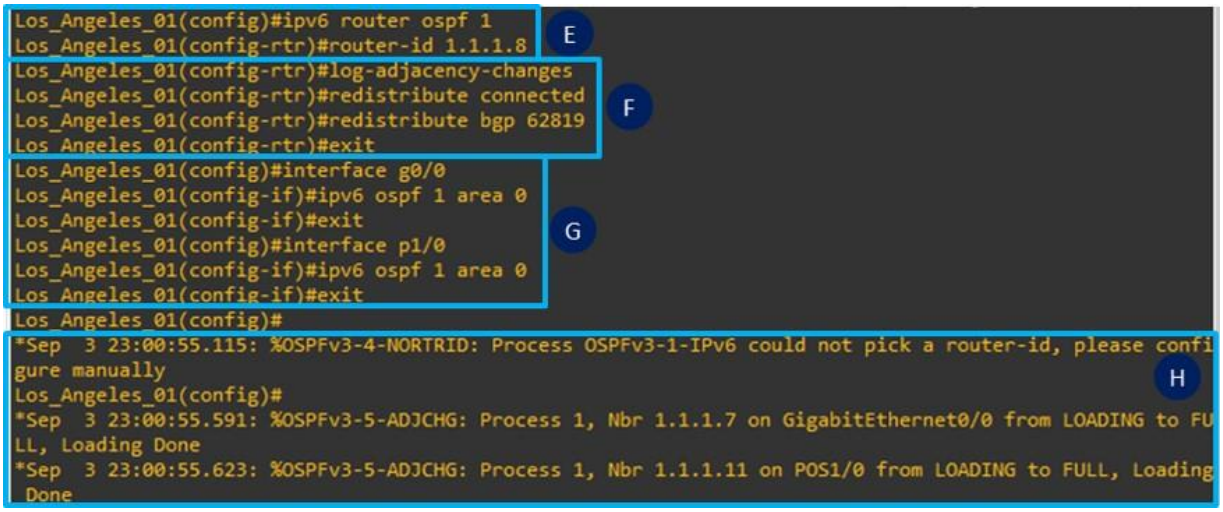


Figura 32 Ejemplo de configuración OSPFv3

Posteriormente se realizó la configuración del protocolo BGP-4 en los *routers* de frontera para conectar con los AS vecinos a PW.

3.3.3 Configuración BGP-4

Configuración del protocolo de enrutamiento BGP-4, lo primero en BGP es indicar el número de AS al que pertenece el *router*, seleccionar el número de proceso y asignar un número de identificación de BGP, así como la especificación de los *routers* y redes vecinas.

```
(config)# router bgp <AS>
(config-router)# bgp router-id <router-id>
(config-router)# no bgp default ipv4-unicast
(config-router)# bgp log-neighbor-changes
(config-router)# neighbor <ipv6-address> remote-as <ASvecino>
(config-router)# neighbor <ipv6-address> update-source <Tipo-Número>
(config-router)# address-family ipv6
(config-router-af)# neighbor <ipv6-address> activate
(config-router-af)# network <network/prefix-length>
```

Para algunos *routers* se habilitó la función de redistribución de BGP, dado que están conectados a un diferente AS.

```
(config-router-af)# redistribute connected
(config-router-af)# redistribute ospf 1 match internal external 1 external 2
(config-router-af)# no synchronization
(config-router-af)# exit-address-family
```

Continuando con el ejemplo del *router* de Los_Angeles_01 se muestra en la Figura 33 la configuración sobre BGP-4. A continuación, se enumeran los pasos de esta configuración:

I. Se estableció el AS del *router*
J. Se generó de modo manual el *Router ID*
K. Se indicaron los AS vecinos a los que se encuentra conectado
L. Se activaron las familias de red sobre el IPv6
M. Se habilitaron las conexiones con OSPFv3 mediante la redistribución.
N. Se muestran las adyacencias establecidas con el *router* después de configurarlo.

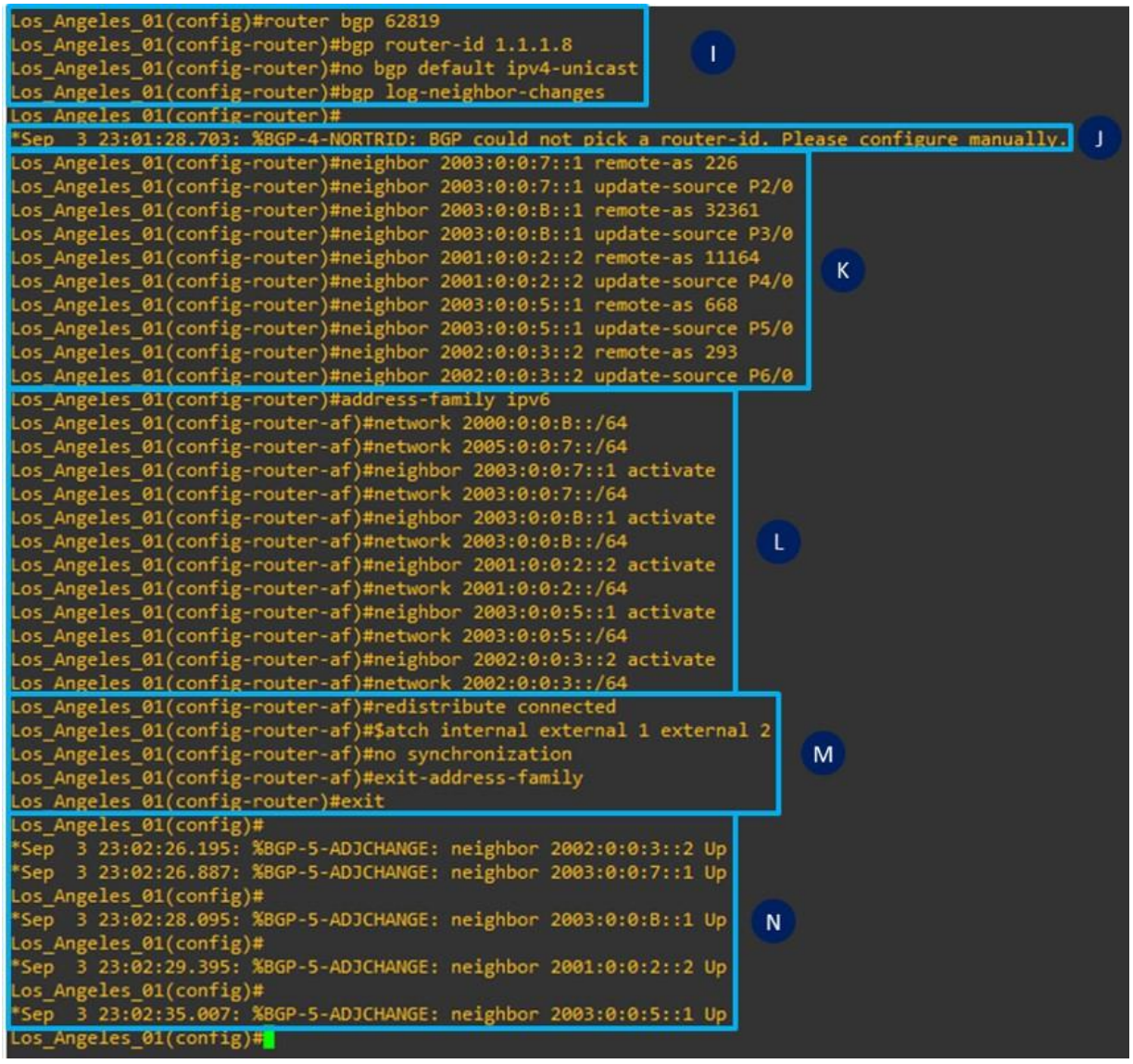


Figura 33 Ejemplo de configuración BGP-4

3.3.4 Configuración SNMPv3

Configuración de SNMP para la gestión de los *routers* en la red.

(config)# snmp-server community ***<contraseña>*** rw

Configuración específica para SNMPv3, en donde se indica el nombre del usuario y el grupo al que se pertenecerá. También se utilizará una contraseña para poder colocar seguridad en cada *router*.

(config)# snmp-server group ***<NombredelGrupo>*** v3 priv

(config)# snmp-server user ***<NombredeUsuario>*** ***<NombredelGrupo>*** v3 auth sha ***<contraseña>*** priv aes ***<contraseña>***

(config)# snmp-server host ***<Dirección IP del destino>*** version 3 priv <NombredeUsuario>

Siguiendo con la configuración en el *router* de Los_Angeles01 esta vez para habilitar el agente para la gestión de cada equipo *router* como se muestra en Figura 34.

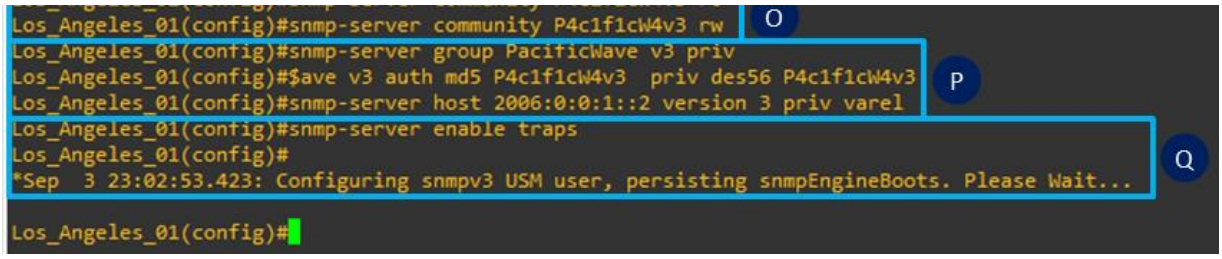


Figura 34 Ejemplo de configuración SNMP para habilitar agentes en routers

O. Se habilitó el acceso SNMP donde se colocó la comunidad (contraseña), en este caso *P4c1f1cW4v3*. Los permisos otorgados de leer y escribir (*rw - read-write*).

P. Se habilitó el grupo, usuario y las credenciales para poder acceder desde el administrador al agente.

Q. Se habilitaron las *traps* y SNMPv3.

3.3.4.1 Administrador SNMP mediante iReasoning

Para las pruebas de gestión se instaló *iReasoning MIB* en la máquina virtual con sistema operativo Windows 10, *iReasoning MIB* es una herramienta para administrar dispositivos y aplicaciones de red habilitados para SNMP, permitiendo a usuarios cargar MIBs. También permite emitir solicitudes SNMP para recuperar los datos del agente o realizar cambios en el agente. Un receptor de capturas incorporado recibe y procesa capturas SNMP de acuerdo con su motor de reglas. En la Figura 35 se muestran los elementos con los que se trabajaron para poder visualizar la gestión de cada *router* en él se habilitó el agente SNMP.

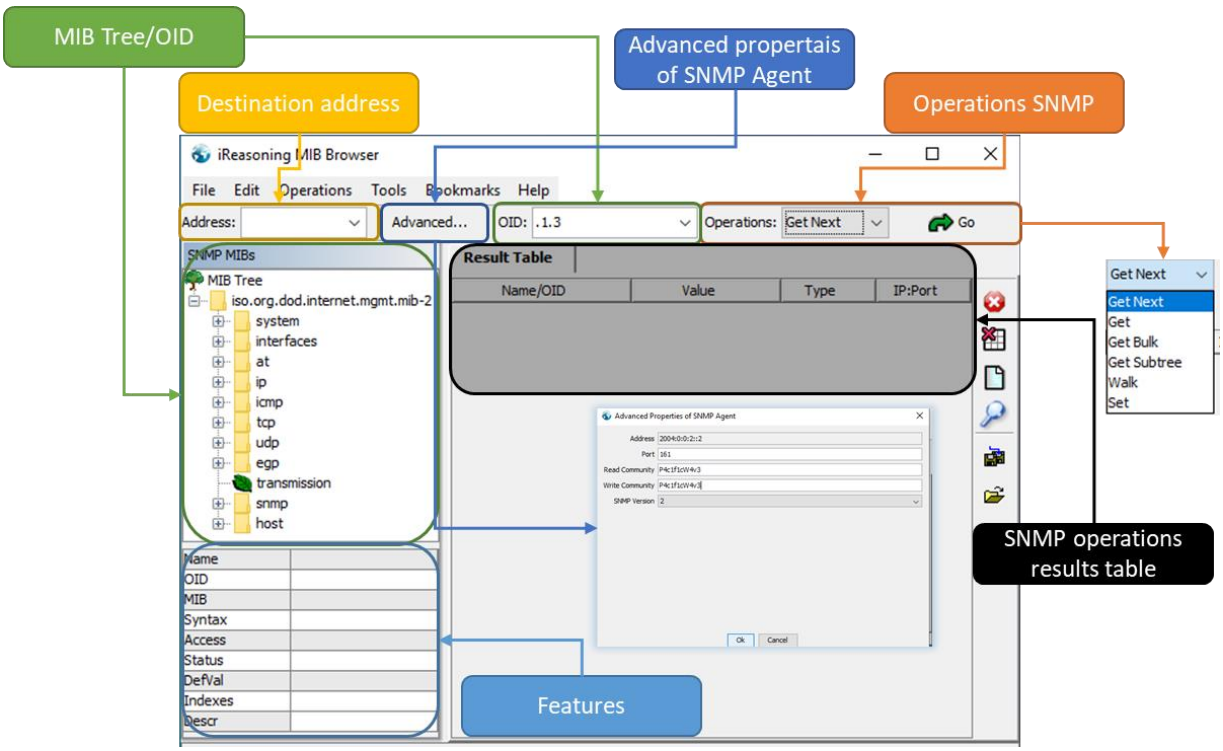


Figura 35 Interfaz de iReasoning MIB Browser con elementos utilizados en esta tesis

En la Figura 36 se muestra la configuración de agentes en iReasoning de los *routers*.

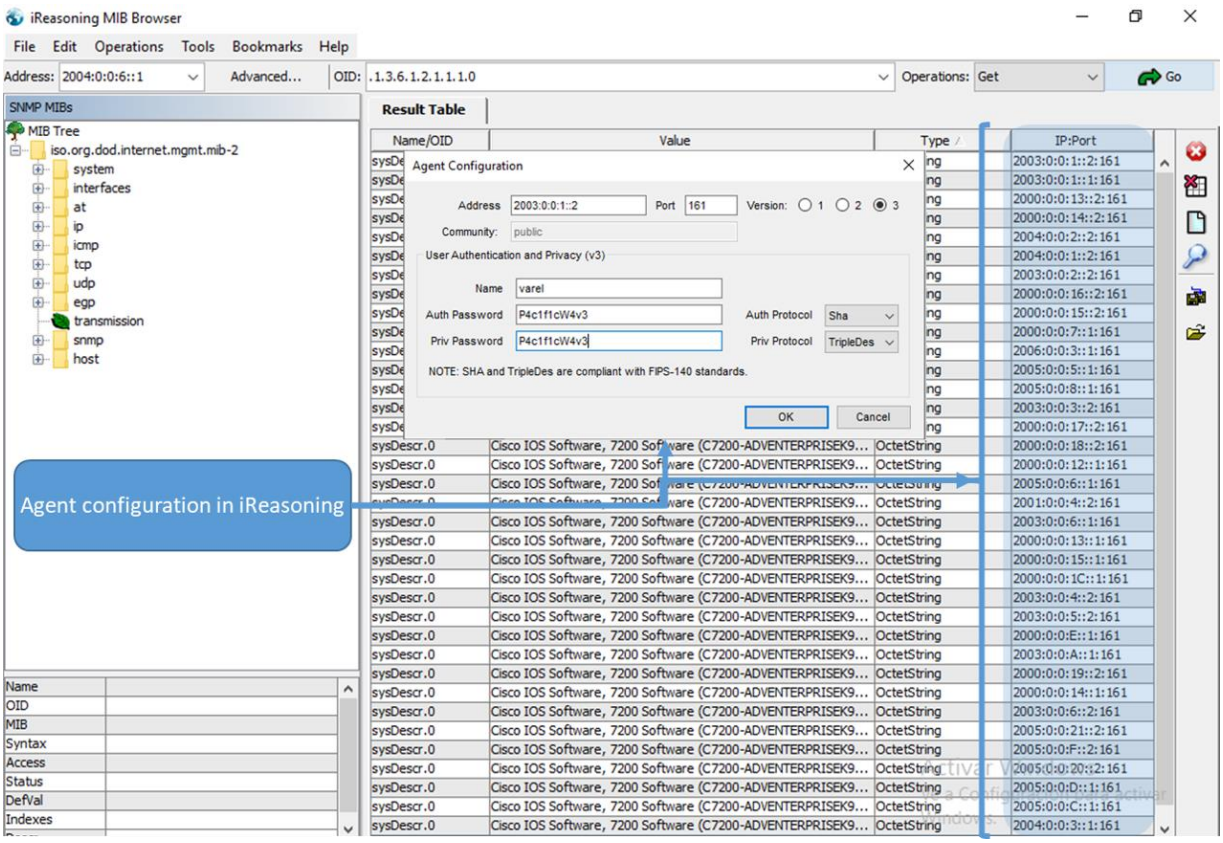


Figura 36 configuración de agentes en iReasoning

Una vez habilitado cada agente *iReasoning* almacena las direcciones para futuras consultas sin necesidad de volver a ingresar los datos de este.

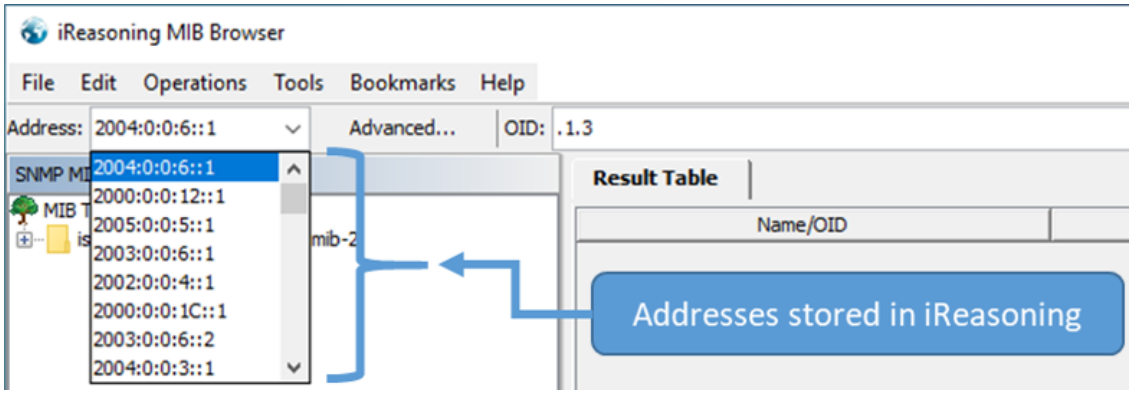


Figura 37 direcciones almacenadas en iReasoning

Posteriormente se indican los OIDS que sirvieron de prueba para este trabajo.

3.3.5 OIDs para pruebas de gestión

En la Tabla 22 se muestran los objetos que se gestionaron en esta tesis, así como su tipo de acceso y la descripción de cada uno de ellos.

Nombre del OID	OID	Acceso	Descripción del objeto
SysName	1.3.6.1.2.1.1.5	*read-write*	Un nombre asignado administrativamente para este nodo gestionado. Por convención, este es el nombre de dominio completo del nodo. Si se desconoce el nombre, el valor es la cadena de longitud cero.
IfNumber	1.3.6.1.2.1.2.1	*read-only*	El número de interfaces de red (independientemente de su estado actual) presentes en este sistema.
IfTable	1.3.6.1.2.1.2.2	*read-only*	Una lista de entradas de la interfaz. El número de entradas viene dado por el valor de *ifNumber*.
IfDescr	1.3.6.1.2.1.2.2.1.2	*read-only*	Una cadena de texto que contiene información sobre la interfaz. Esta cadena debe incluir el nombre del fabricante, el nombre del producto y la versión del hardware/software de la interfaz.

IfOperStatus	1.3.6.1.2.1.2.2.1.8	*read-only*	El estado operativo actual de la interfaz. La interfaz puede estar en uno de los siguientes estados: 1. No disponible 2. La interfaz administrativa está inactiva 3. Disponible
SysUpTime	1.3.6.1.2.1.1.3	*read-only*	El tiempo (en centésimas de segundo) desde que se reinicializó por última vez la parte de administración de red del sistema.

Tabla 22 Objetos de SNMP que se gestionarán en este trabajo

Finalmente, para que al apagar los *routers* no se pierda su configuración realizada se procedió a guardar los cambios.

3.3.6 Guardar cambios de configuraciones en *routers*

Todas las configuraciones en el *router* se almacenan en la memoria RAM (*Random Access Memory*), dada esto al momento de ser apagado el *router* o el sistema se pierde la información de dicha configuración. Para que esto no ocurra se debe copiar la configuración realizada de la RAM del *router* a la NVRAM (*Non-volatile random access memory*) del mismo *router*, mediante el comando:

```
# copy running-config startup-config
```

3.3.7 Softwares instalados en MV Windows

Para poder realizar un análisis de los datos y los protocolos utilizados se hizo uso de la herramienta ***Wireshark*** el cual permite analizar el tráfico red en tiempo real, a menudo usada para solucionar los problemas de Red.

Para transferencia de archivos se instaló ***WinSCP*** el cual es una aplicación libre y de código abierto, el cual sirve como un cliente SFTP (*Secure File Transfer Protocol*) gráfico para Windows que emplea SSH (*Secure* Shell). Su función principal es la transferencia de archivos entre dos sistemas informáticos, el local y uno remoto que funcione como servidor.

Capítulo 4

Resultados y conclusiones

Una vez realizada la configuración con los protocolos de enrutamiento y de gestión, se realizaron las validaciones de conectividad y de gestión, asimismo para dar una revisión del consumo de recursos durante la emulación se tomaron diferentes puntos del rendimiento del equipo llamados “Estados”, a continuación, se indica a que refiere cada uno de estos:

Estado 1 - Sin GNS3, únicamente el sistema de Windows en funcionamiento.

Estado 2 - Línea base, cuando se inicia GNS3

Estado 3 - GNS3 en estado activo, *routers* activos y máquinas virtuales activas

Estado 4 - Prueba de conectividad

Estado 5 - Prueba de gestión

4.1 Estado 1 – GNS3 en estado inactivo

El Estado 1 es el punto inicial, el sistema operativo ejecuta funciones y procesos para estar optimo a cualquier instrucción que se le solicite, tal y como se muestra en la Figura 38 el rendimiento del equipo tiene un uso de recursos del 23%.

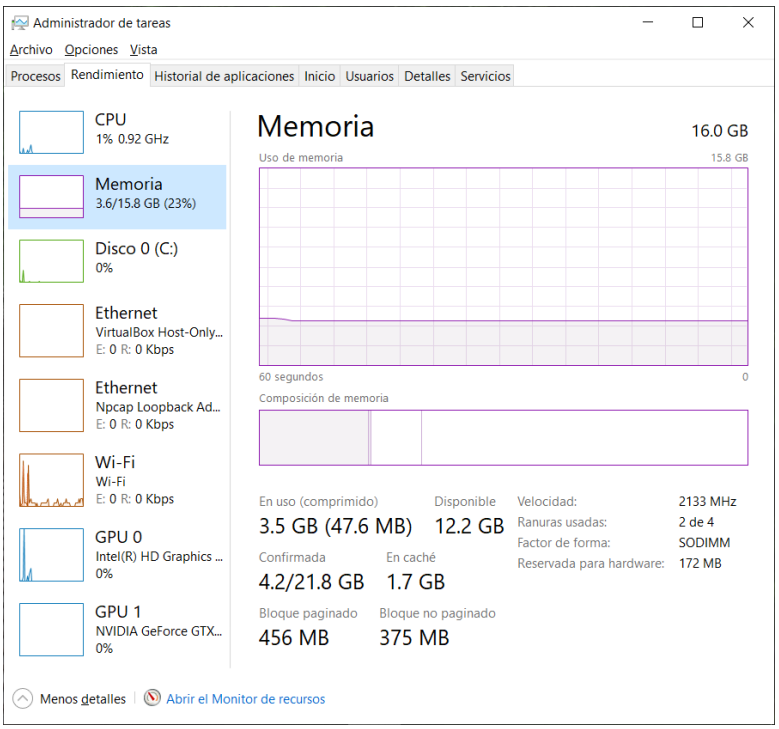


Figura 38 Rendimiento del sistema antes de ejecutar GNS3 (Estado 1)

4.2 Estado 2 – Línea base

Línea base, en este estado se ejecuta GNS3 con la emulación sin los sistemas activos (*Routers* y Máquinas virtuales apagados, marcadas con puntos rojos en las conexiones de GNS3), tal y como se presenta en la Figura 39.

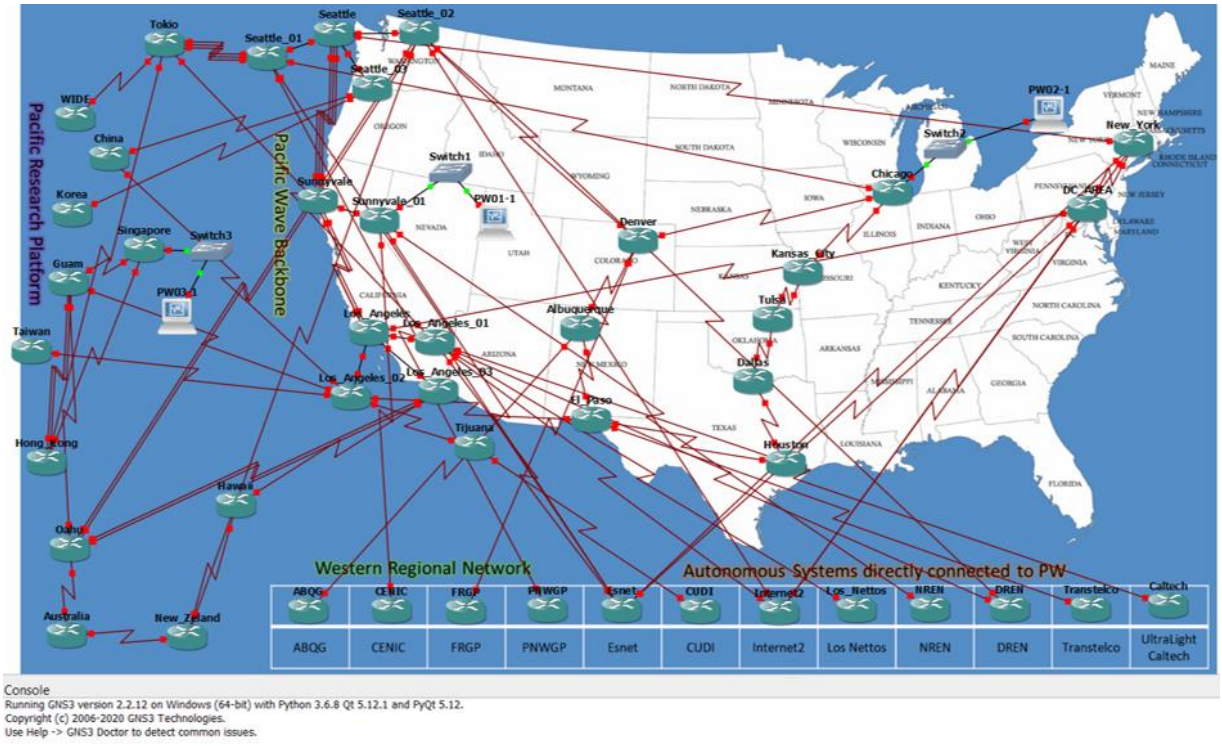


Figura 39 Topología PW sobre GNS3 cuando los routers no están inicializados

En este estado el rendimiento del equipo tiene un incremento del 23% al 27% de uso en la memoria, como se muestra en la Figura 40.

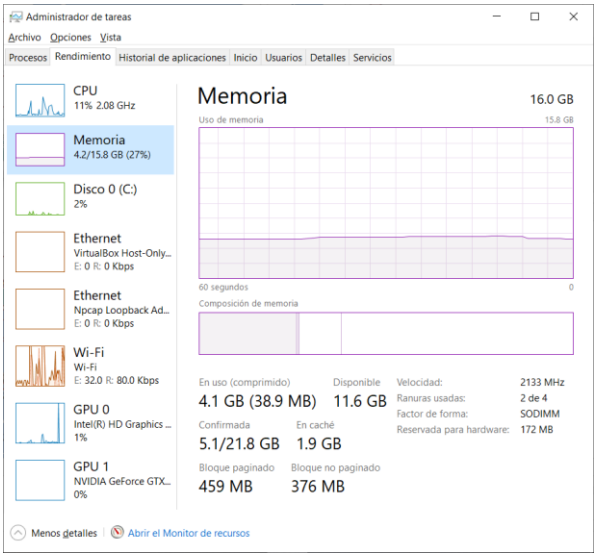


Figura 40 Rendimiento del sistema al ejecutar GNS3

4.3 Estado 3 – GNS3 en estado activo

En el tercer estado GNS3 se encuentra en ejecución junto con todos sus sistemas en funcionamiento es decir *routers* y máquinas virtuales encendidas, en la Figura 41 se muestran activos los *routers* (interfaces marcadas con puntos verdes) con la topología de PW y sus respectivas conexiones.

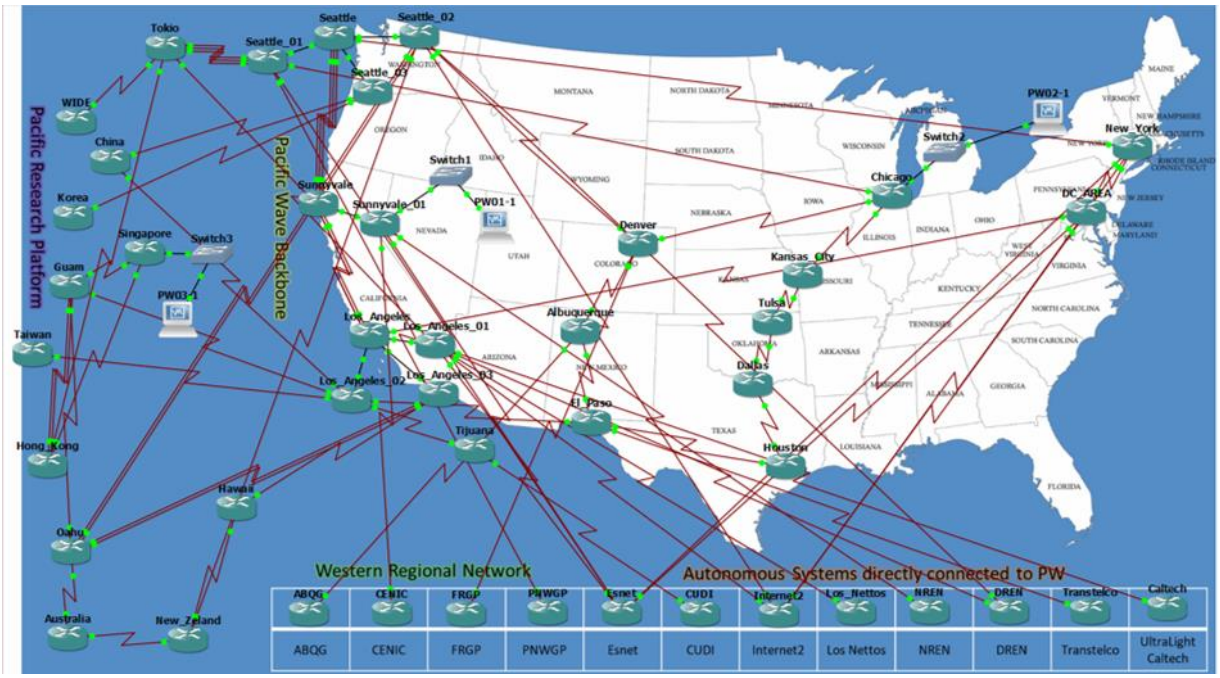


Figura 41 Topología PW sobre GNS3 con routers inicializados y protocolos en funcionamiento

Para el rendimiento del equipo en el estado 3 se tomaron dos puntos: el primero al iniciar de golpe todos los sistemas (Estado 3.1), el segundo, una vez ya en equilibrio con los sistemas encendidos (Estado 3.2). El tiempo que se llevó el sistema en encender todos los *routers* y todas las máquinas virtuales a un estado de estabilidad fue de 5 minutos. En la Figura 42 se muestran los rendimientos de los estados antes mencionados respectivamente.

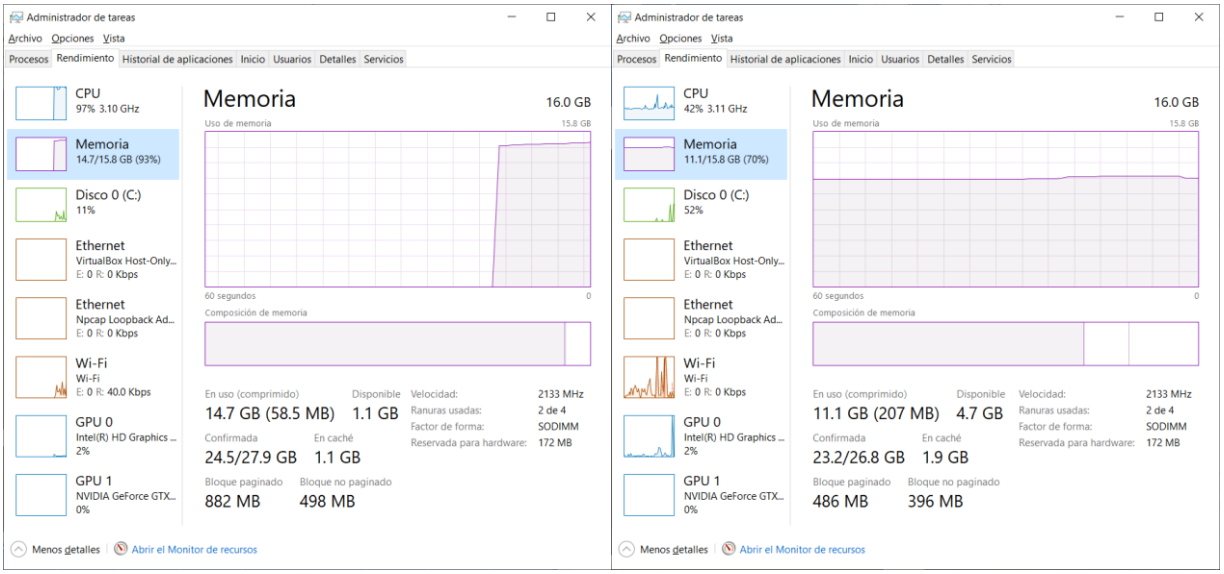


Figura 42 Rendimientos del sistema en ejecución (Inicial - Estado 3.1 izquierda y en Equilibrio - Estado 3.2 derecha)

4.3.1 Validación de configuración de enrutamiento

Como se mencionó anteriormente cada *router* fue configurado de acuerdo a su uso. Es decir, un *router* que representaba una conexión a un AS diferente posiblemente tendría solo la configuración de BGP-4 y SNMP por ejemplo Internet2 el cual se conecta por diferentes *routers* a la red de PW. Un *router* dentro del *Backbone* de PW puede haber sido configurado únicamente por OSPFv3 y SNMP, si este no está conectado a un AS diferente. Por último, están los *routers* que tienen las tres configuraciones BGP-4, OSPFv3 y SNMPv3; ya que estos *routers* mantienen conexión con algún AS diferente y son parte del Backbone de PW.

Para poder validar que los *routers* tengan las configuraciones habilitadas se usó en cada uno el comando *show ipv6 protocols*, el cual tiene como función mostrar los parámetros y el estado de los procesos activos del protocolo de enrutamiento IPv6. En la Figura 43 se muestra a manera de ejemplo el uso de dicho comando sobre el *router* de Los_Angeles_01 el cual tiene las tres configuraciones antes mencionadas.

```
Los_Angeles_01#sh ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 1"
  Router ID 1.1.1.8
  Autonomous system boundary router
  Number of areas: 1 normal, 0 stub, 0 nssa
  Interfaces (Area 0):
    GigabitEthernet1/0
    GigabitEthernet0/0
  Redistribution:
    Redistributing protocol connected
    Redistributing protocol bgp 62819
IPv6 Routing Protocol is "bgp 62819"
  IGP synchronization is disabled
  Redistribution:
    Redistributing protocol connected
    Redistributing protocol ospf 1 (internal, external 1 & 2, )
  Neighbor(s):
    Address                 FiltIn FiltOut Weight RoutemapIn RoutemapOut
    2001:0:0:2::2
    2002:0:0:3::2
    2003:0:0:5::1
    2003:0:0:7::1
    2003:0:0:B::1
IPv6 Routing Protocol is "bgp multicast"
Los_Angeles_01#
```

Figura 43 validación de los protocolos IPV6, OSPFv3 y BGP

La tabla de enrutamiento del *router* de Los_Angeles_01 se muestra mediante el comando *#show ip route* en la Figura 44 podemos observar las conexiones que se establecieron con este *router*.

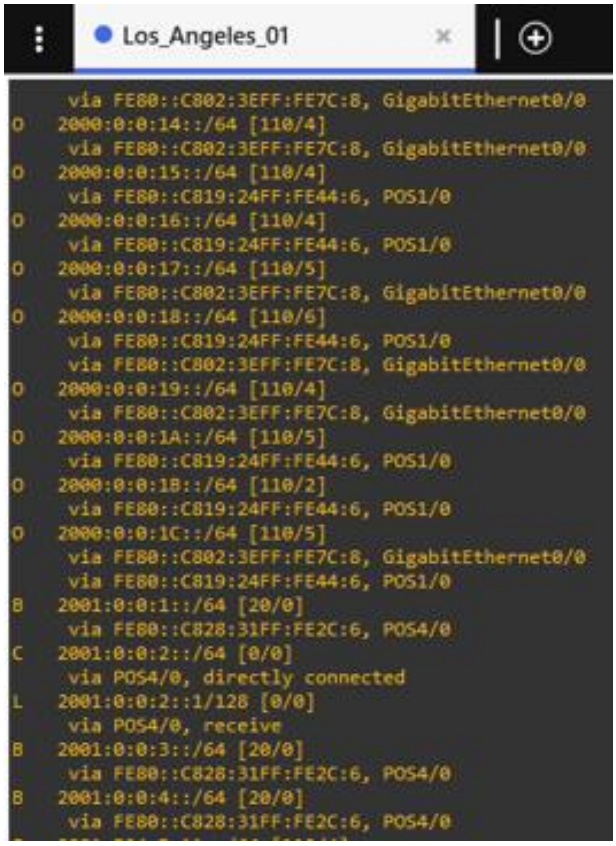


Figura 44 Validación de configuración de protocolos de enrutamiento mediante su tabla de enrutamiento en el Router Los_Angeles_01

En la Tabla 23 se muestran los indicadores del tipo de protocolo de enrutamiento empleado y descripción de cada uno de los símbolos que aparecieron mediante el comando *#show ip route*.

Símbolo	Tipo de conexión	Descripción
C	*Connected*	La red con la cual se encuentra conectada de manera directa el *router*.
L	*Local*	La dirección IPv6 con la que está asociada la interfaz del *router*.
O	*OSPF*	Las conexiones con los *router* que están configurados mediante OSPFv3.
B	*BGP*	Los vecinos BGP que se encuentran conectados al *router*.

Tabla 23 Descripción de los códigos que se trabajaron sobre esta tesis en la tabla de enrutamiento.

A continuación, se explica la información que la tabla de enrutamiento arrojo. Para las rutas conectadas directamente (Figura 45) y las redes remotas (Figura 46).

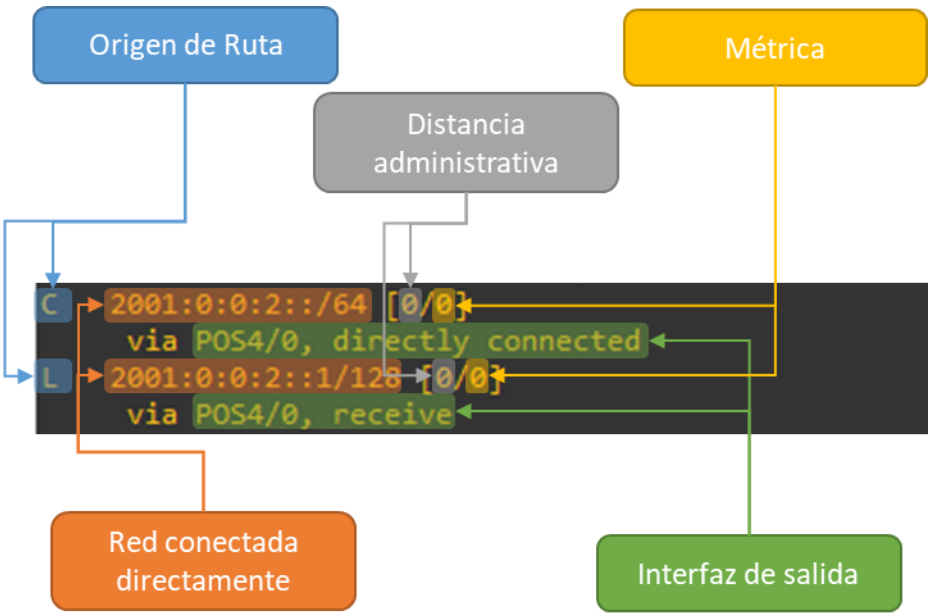


Figura 45 Rutas conectadas directamente en el router Los_Angeles_01

Origen de la ruta: muestra el modo que el *router* descubrió la ruta. Las interfaces que están conectadas directamente contienen dos indicadores de origen de ruta:

- "C" el cual, identifica una red conectada directamente
- "L" el cual, identifica que esta es una ruta local

Red conectada directamente: la red o la dirección IPv6 que se encuentran conectadas de manera directa a la interfaz del *router*.

Distancia administrativa: muestra el valor de acuerdo al indicador o protocolo que se utilizó.

Métrica: muestra el valor asignado para llegar a la red destino.

Interfaz de salida: indica la interfaz de salida que se utilizó para el envío de paquetes a la red de destino.

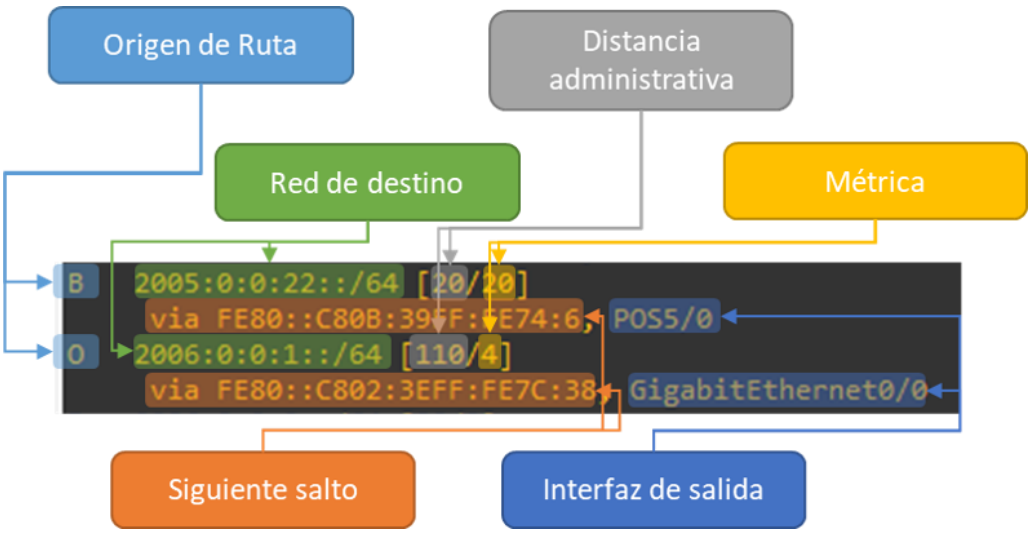


Figura 46 Redes remotas en el router Los_Angeles_01

Origen de la ruta: muestra el modo que el *router* descubrió la ruta. Los códigos comunes incluyen O (OSPF), B (BGP), R (RIP) y S (ruta estática).

Red de destino: muestra la dirección de la red IPv6 destino.

Distancia administrativa: muestra el valor de acuerdo al indicador o protocolo que se utilizó.

Métrica: muestra el valor asignado para llegar a la red destino.

Siguiente salto: identifica la dirección IPv6 del *router* siguiente al que se debe reenviar el paquete.

Interfaz de salida: indica la interfaz de salida que se utilizó para el envío de paquetes a la red de destino.

4.3.2 Captura de paquetes de OSPFv3

En esta sección podremos analizar los paquetes enviados por los *routers* de OSPFv3, en este caso se muestra en la Figura 47 el *Hello Packet*, Wireshark nos permite validar el funcionamiento de OSPF, así como su versión, tipo, longitud, *Router ID*, etc.

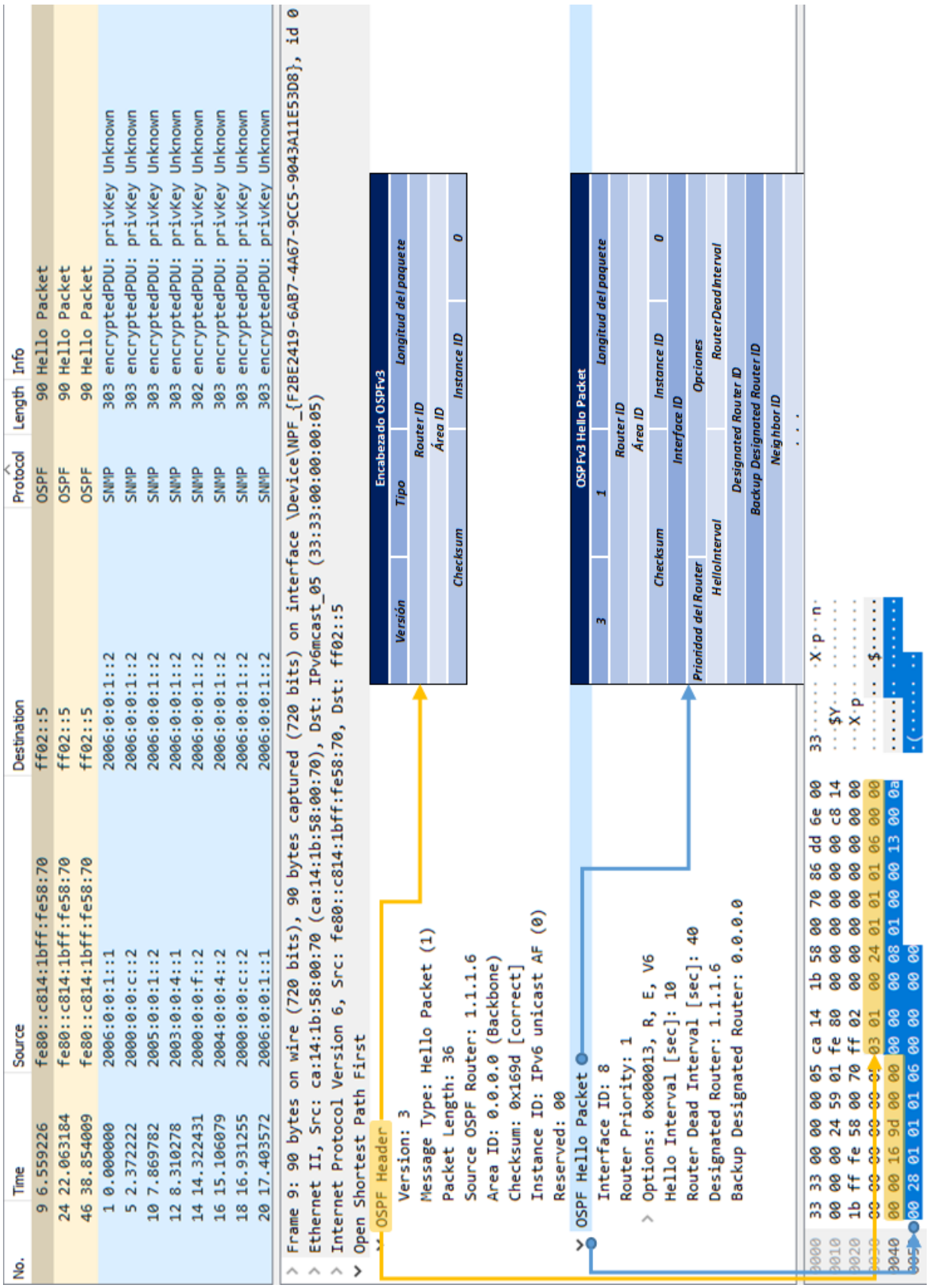


Figura 47 Captura de paquetes de OSPFv3 mediante Wireshark

4.4 Resultados de conectividad y Estado 4

Una vez realizada la configuración de los protocolos de enrutamiento en cada *router* se procedió a realizar pruebas de conectividad mediante el comando Ping, el comando *Traceroute* y mediante la transferencia de archivos.

4.4.1 Comando Ping

El comando ping nos indica latencia y conectividad de equipos locales y remotos. En la Figura 48 se realizó ping a *routers* que estaban dentro del Backbone de PW y *routers* de AS conectados con PW. Se consideró realizar la conexión con mayor cantidad de saltos en la topología. Por lo que se tomó como *router* de referencia Sunnyvale_01 y se envió ping a Tulsa, Australia y Hong Kong.

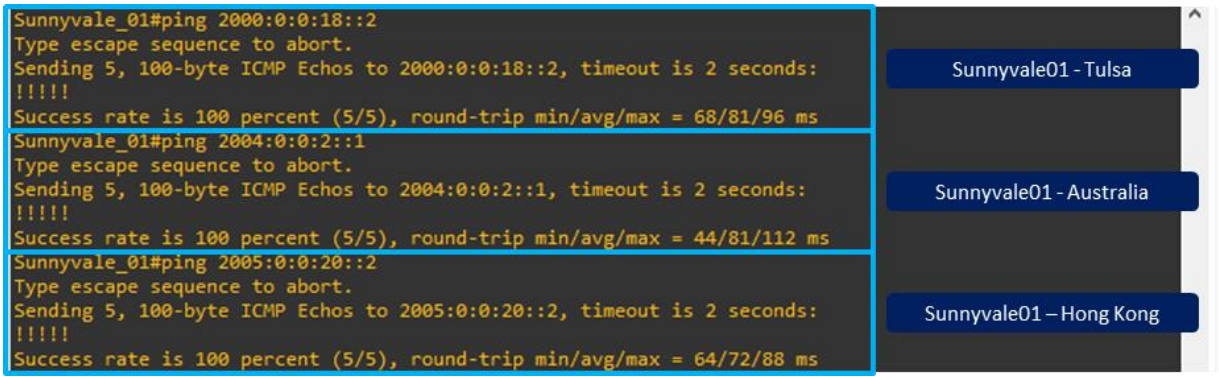


Figura 48 Ping desde el router Sunnyvale_01

Posteriormente se realizó el uso del comando ping desde las máquinas virtuales con CentOS 8 *PW02* y *PW03* (recuadros amarillos) hacia la máquina virtual con Windows 10 *PW01* (recuadro morado) como se muestra en la Figura 49.

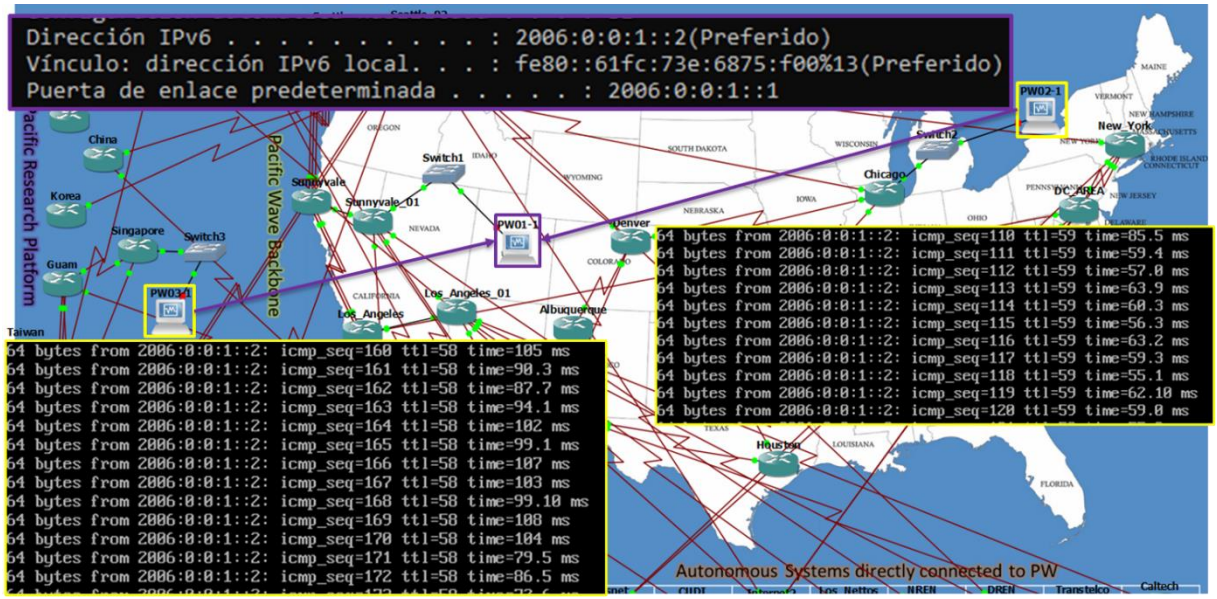


Figura 49 Comando ping de máquinas con CentOS hacia maquina con Windows

Mediante Wireshark se observaron los paquetes enviados entre máquinas virtuales, en la Figura 50 podemos observar algunos paquetes capturados.

- Los paquetes ping de petición (*request*), en donde las fuentes (*Sources*) son *PW02* con dirección IPv6 2006::3:823a:630a:a0a1:641c y *PW03* con dirección IPv6 2006::2:c28d:fa78:12da:ca54, y el destino (*Destination*) *PW01* con dirección IPv6 2006:0:0:1::2.
- Los paquetes ping de respuesta (*reply*), en donde *Source* es la máquina virtual *PW01* y como *Destinations* fueron las máquinas virtuales *PW02* y *PW03*.

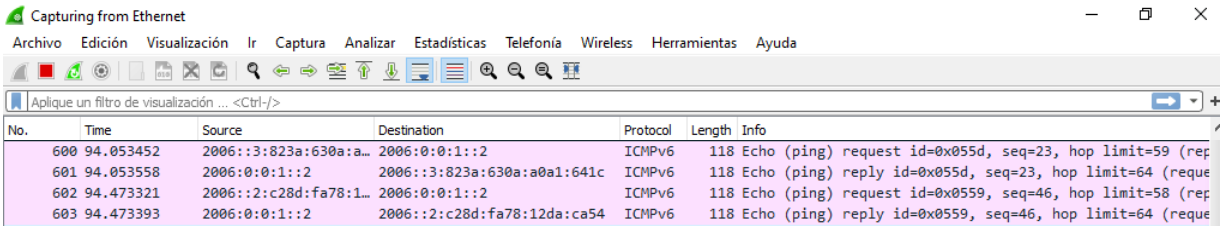


Figura 50 Captura de paquetes ping mediante Wireshark

El rendimiento del equipo se muestra en la Figura 51, en el Estado 4 que es cuando se encuentra la emulación enviando un flujo constante de paquetes ping.

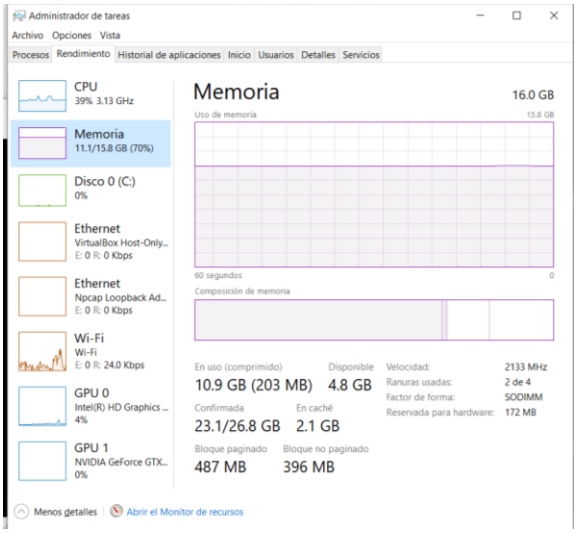


Figura 51 Rendimiento del equipo durante el proceso Estado 4

En la Figura 52 se muestra la captura de paquete mediante Wireshark logrando visualizar origen y destino de paquetes, así como el protocolo que se está ejecutando.

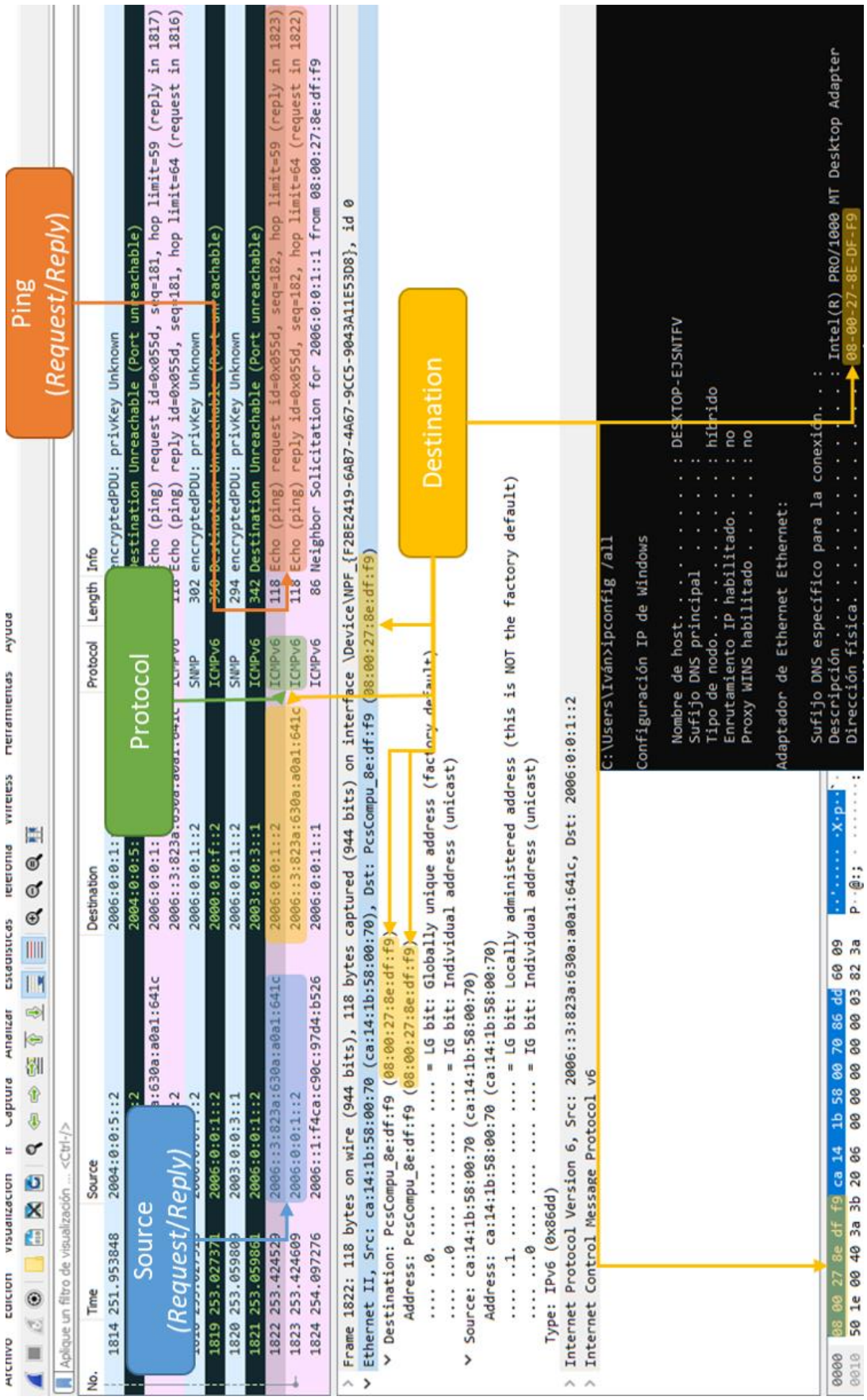


Figura 52 Wireshark captura de paquetes ICMPv6

Mediante Wireshark se validó el envío de paquetes con el protocolo IPv6, mostrando los campos antes revisados, tal y como se muestra en la Figura 53 así mismo la información de los paquetes de ICMPv6.

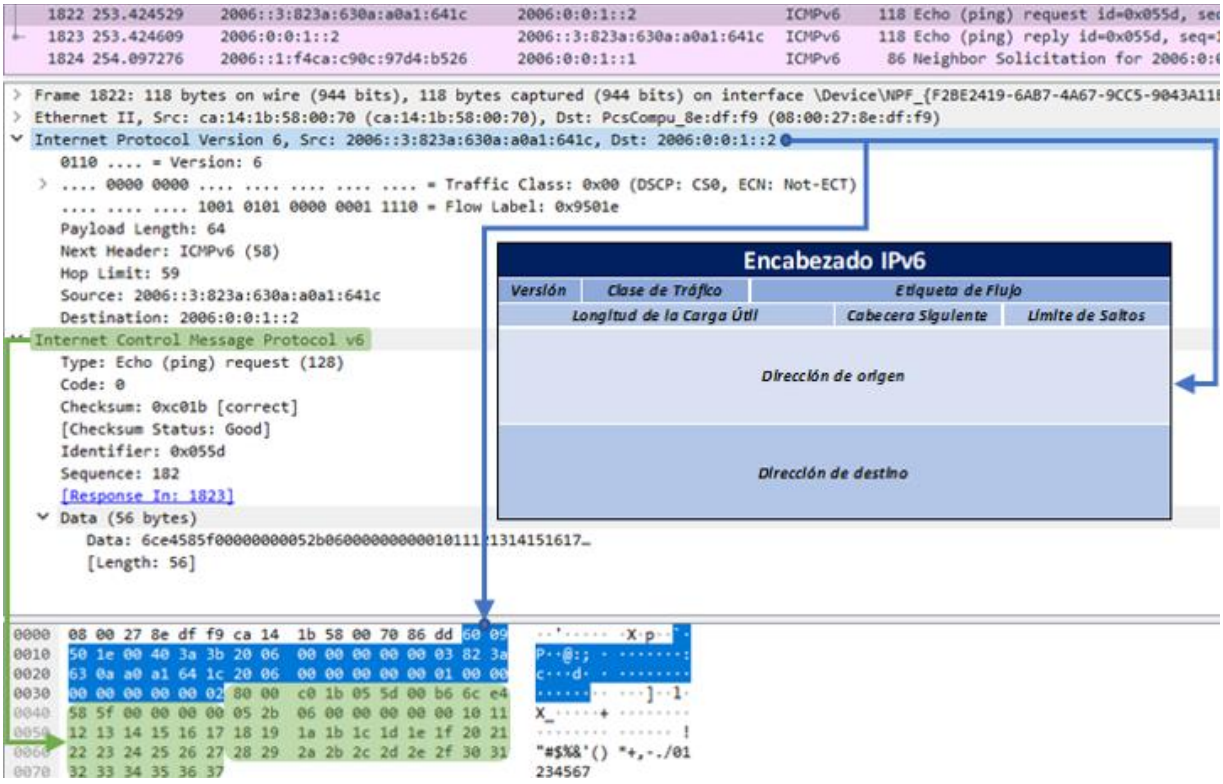


Figura 53 Captura de paquetes mediante Wireshark con el encabezado de IPv6

4.4.2 Comando *Traceroute*

El comando *traceroute* nos indica los saltos y la latencia de los paquetes enviados de un sistema a otro, en la Figura 54 se muestra dicho comando usado hacia los mismos *routers* que en la prueba de conectividad.

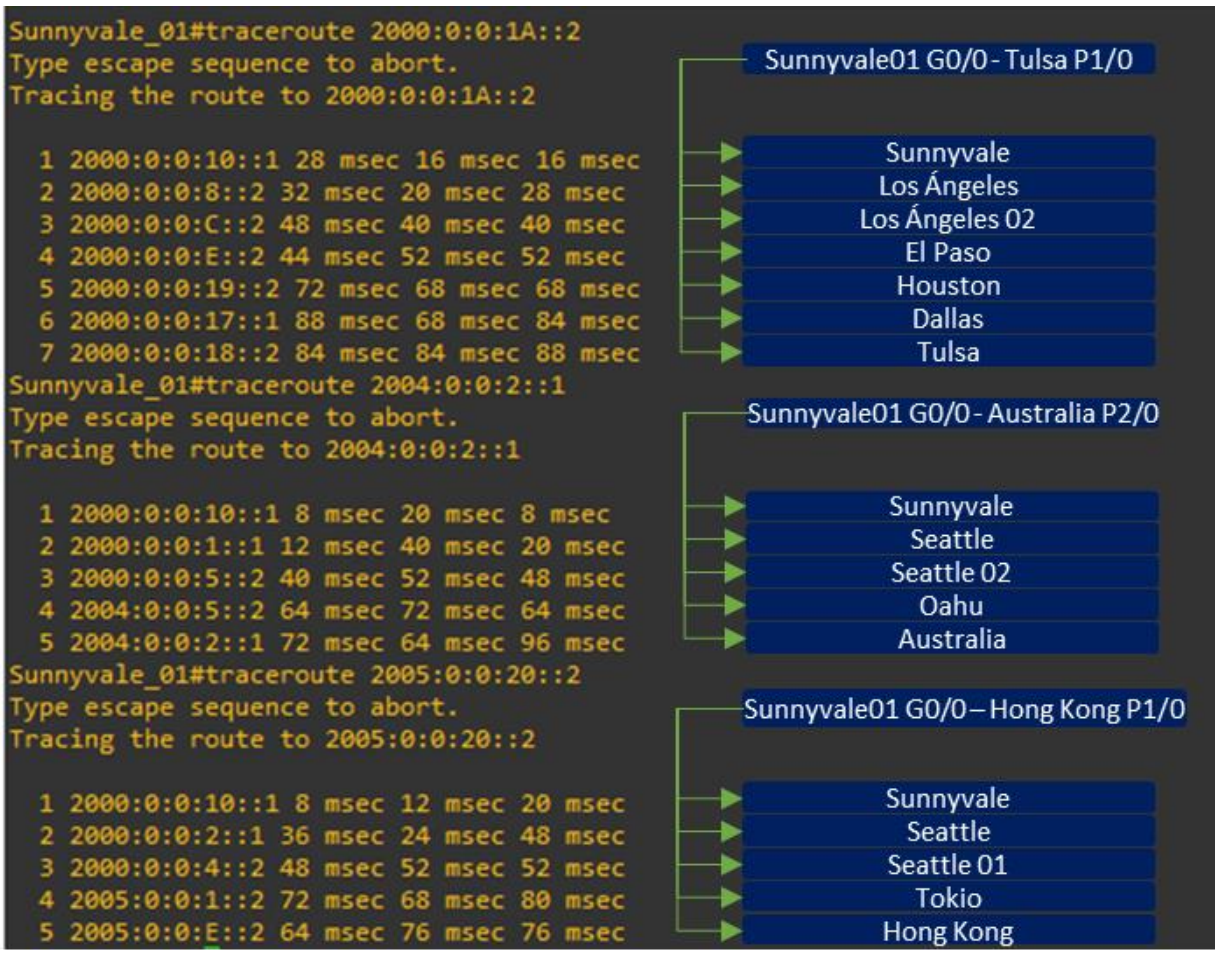


Figura 54 Uso de Traceroute desde el router Sunnyvale_01

Del mismo modo se hizo uso del comando ping y *tracert* para las máquinas virtuales, en este caso se realizó desde el equipo Windows hacia los equipos Linux en la Figura 55, se muestra el uso de los comandos de la máquina virtual que se encuentra conectada en Sunnyvale01 (PW01) hacia la máquina virtual que está conectada en Singapore (PW03), mostrando tanto la latencia como los caminos que tomaron los paquetes al enviarse.

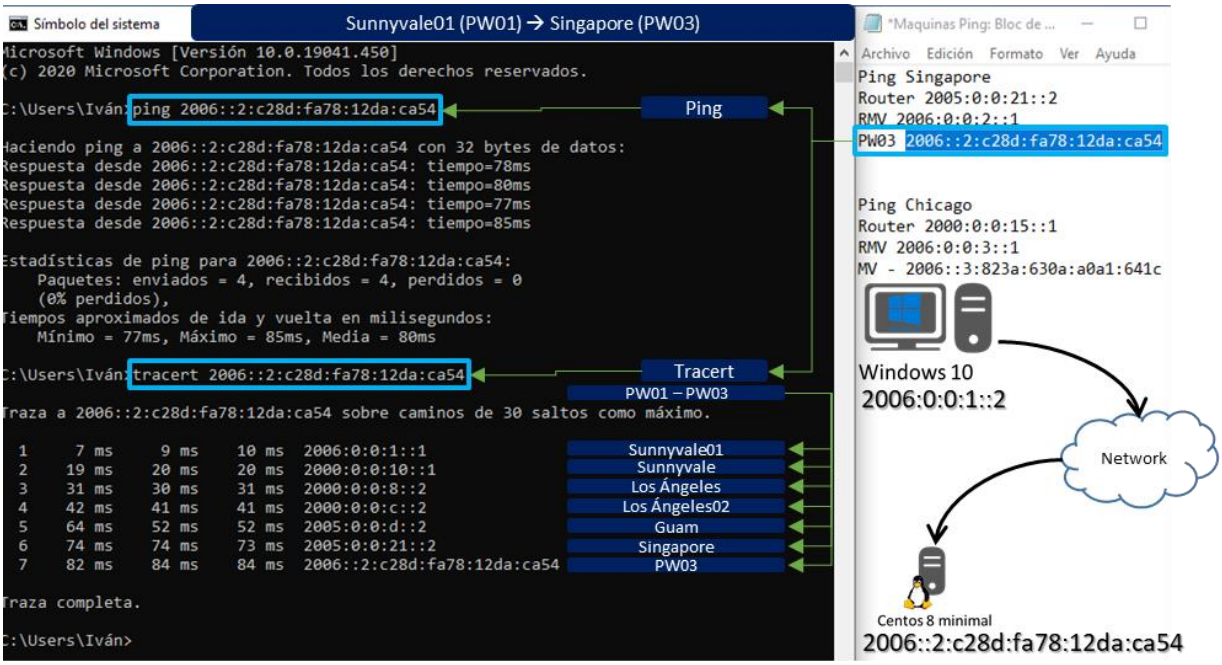


Figura 55 Uso de comando Ping y Tracert de Sunnyvale01 PW01 hacia Singapore PW03

También se realizó el uso de ambos comandos para la máquina virtual (PW02) que se encuentra conectada a Chicago, mostrando latencia y el camino que tomaron los paquetes.

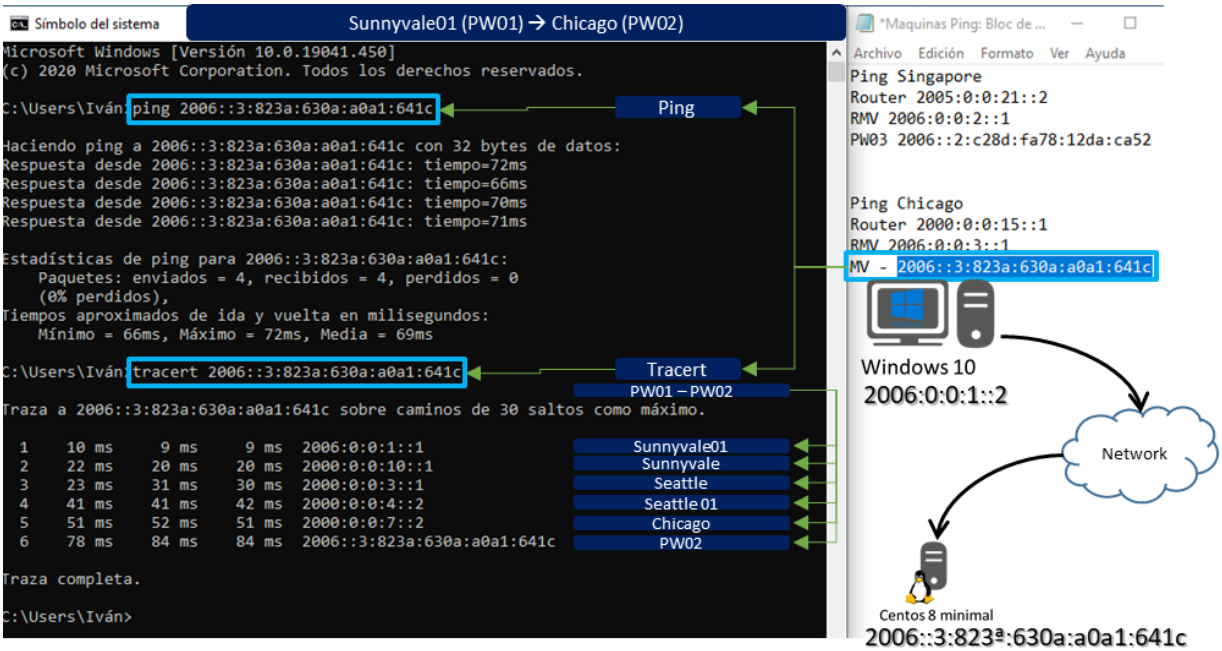


Figura 56 Uso de comando Ping y Tracert de Sunnyvale01 PW01 hacia Chicago PW02

Una vez validada la conectividad en la red se procedió a realizar una prueba de transferencia de diferentes tipos de archivos.

4.4.3 Transferencia de archivos

Para este ejercicio lo primero fue cargar cuatro archivos con diferentes formatos y tamaños en *PW01* posteriormente se ejecutó *WinSCP* instalado en *PW01*, para poder acceder a *WinSCP* se colocó la dirección IPv6 del servidor (2006::2:c28d:fa78:12da:ca54) el cual pertenece a *PW03*, el usuario y contraseña de *PW03* como se muestra en la Figura 57.

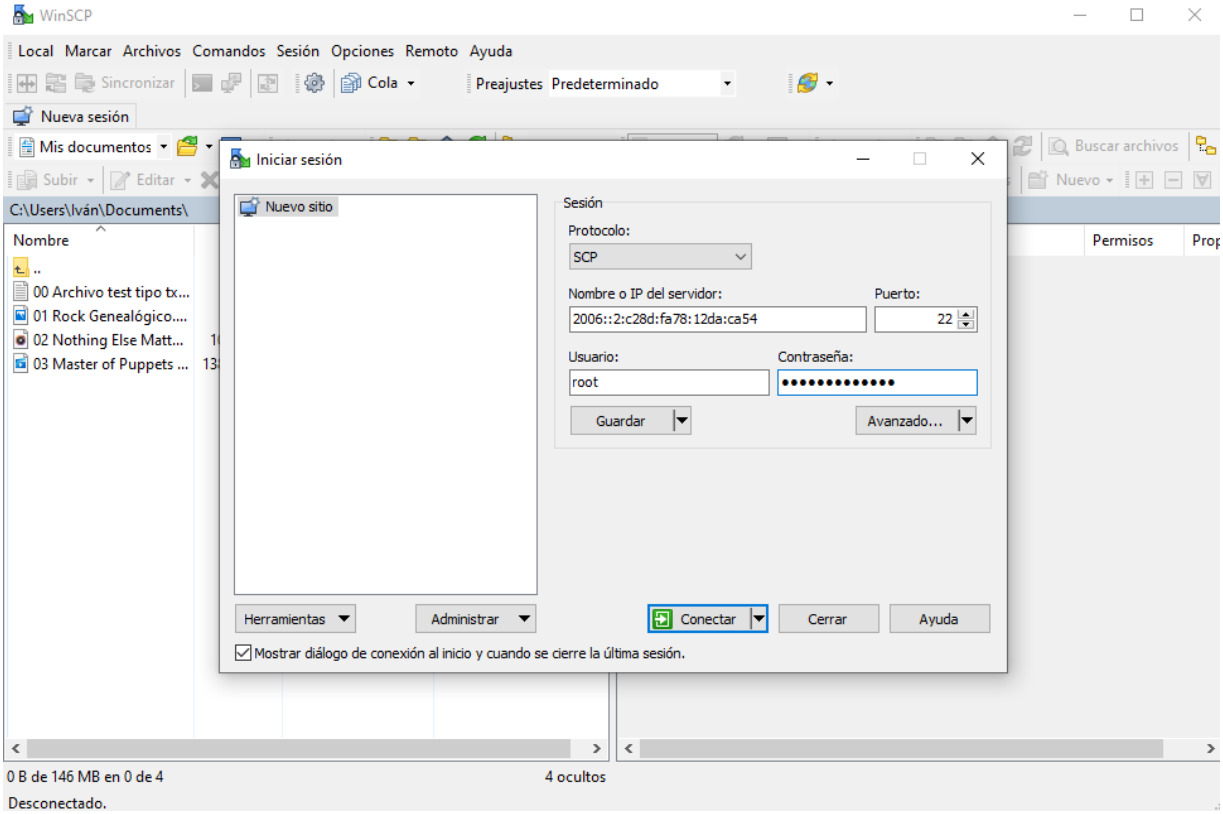


Figura 57 Inicio de sesión desde PW01 hacia PW03 mediante WinSCP

Una vez establecida la conexión con *PW03*, *WinSCP* mostró un aviso indicando el cifrado que se estableció durante la sesión como se observa en Figura 58.

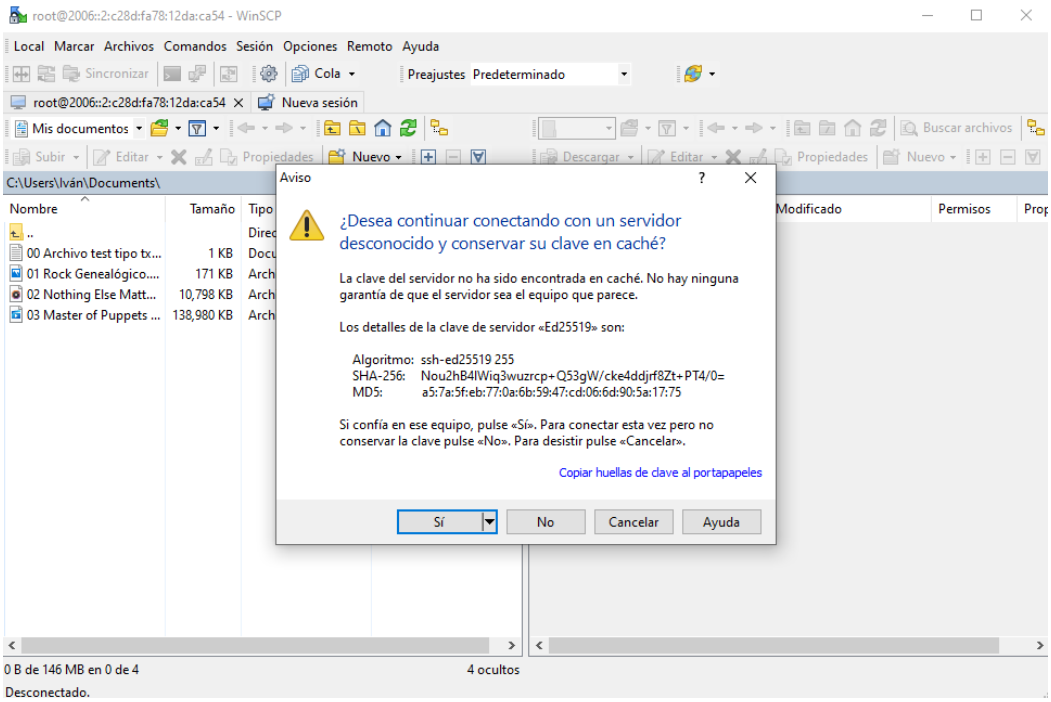


Figura 58 Cifrado al establecer conexión entre PW01 y PW03

Una vez iniciada la sesión se realizaron las transferencias, en primera instancia la transferencia de archivos de manera simultánea, es decir, de *PW01* hacia los equipos de *PW02* y *PW03* se enviaron archivos de texto (1 KB), imagen (171 KB), audio (10,798 KB) y video (138,980 KB). En la Figura 59 se muestra el esquema de las transferencias de archivos.

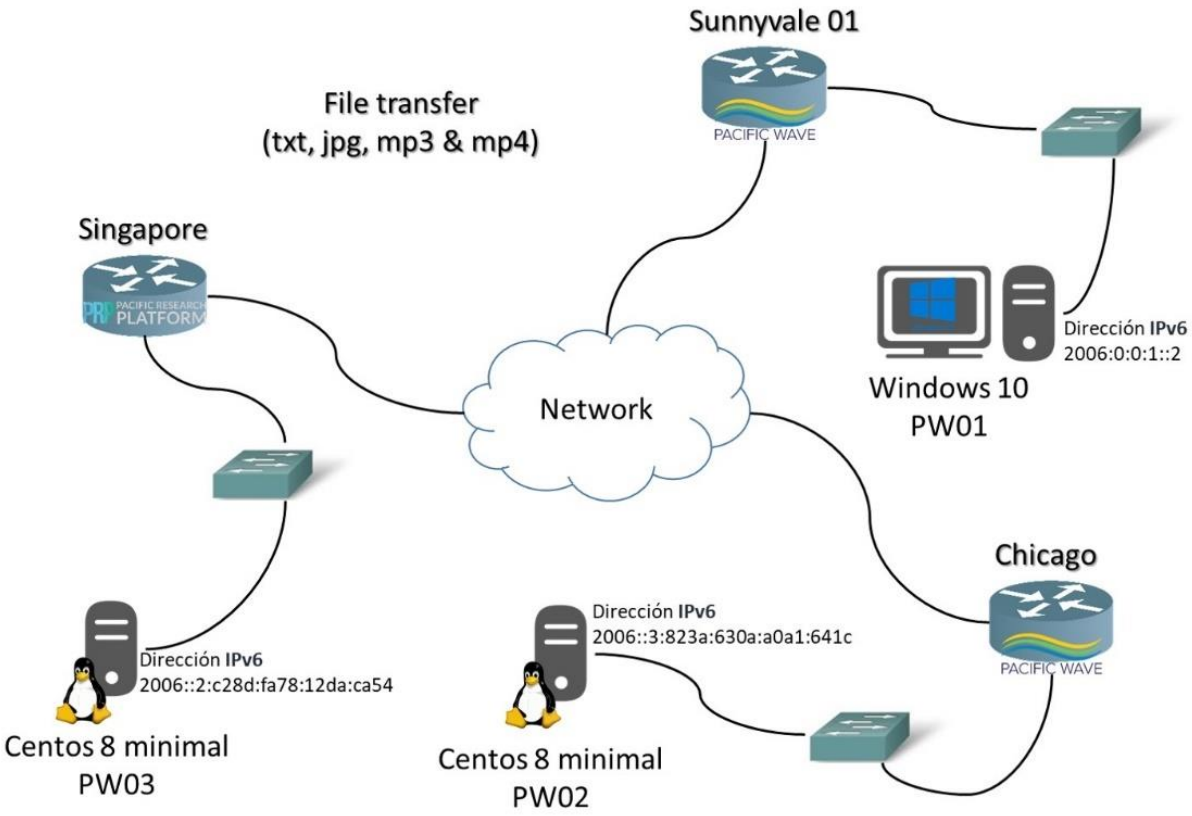


Figura 59 Esquema de máquinas virtuales para transferencia de archivos

En la Figura 60 se muestra la transferencia de los archivos. El tiempo que se llevó el traspaso de los cuatro archivos fue de 5:12 horas, dado este tiempo se realizó un único ejercicio para posteriormente realizar la transferencia de uno a uno de los mismos archivos.

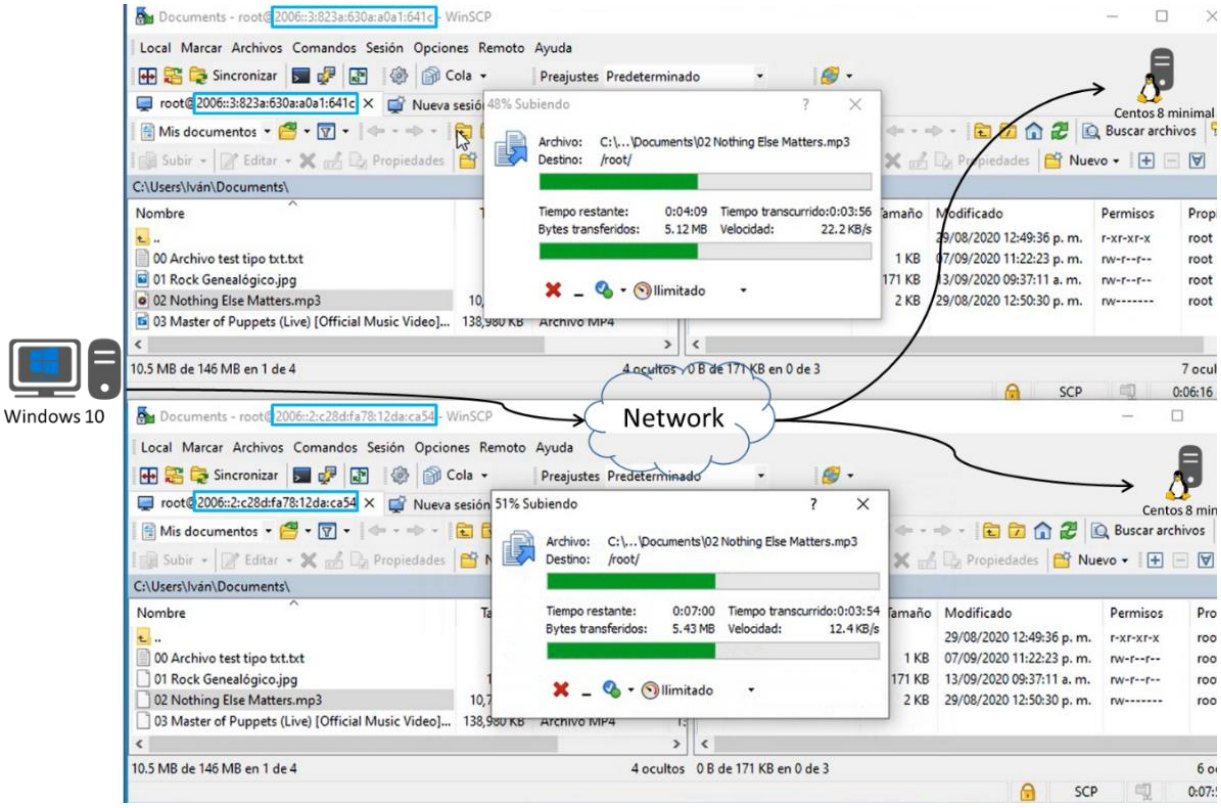


Figura 60 Transferencias simultaneas de PW01 hacia las MVs PW02 y PW03

Posteriormente se realizó la transferencia uno a uno es decir de *PW01* con dirección IPv6 2006:0:0:1::2 el cual se encuentra dentro del AS de PW hacia una de las máquinas virtuales, en este caso se envió a *PW03* con dirección 2006::2:c28d:fa78:12da:ca54 el cual se encuentra en el AS de PRP. Este ejercicio se replicó 5 veces para cada uno de los archivos.

4.4.3.1 Archivo de texto

El archivo de texto (1 KB) se transfirió de manera inmediata por lo que únicamente se logró capturar los datos en Wireshark mientras que en *Winscp* no se logró verificar la velocidad de la transferencia, como se muestra en Figura 61.

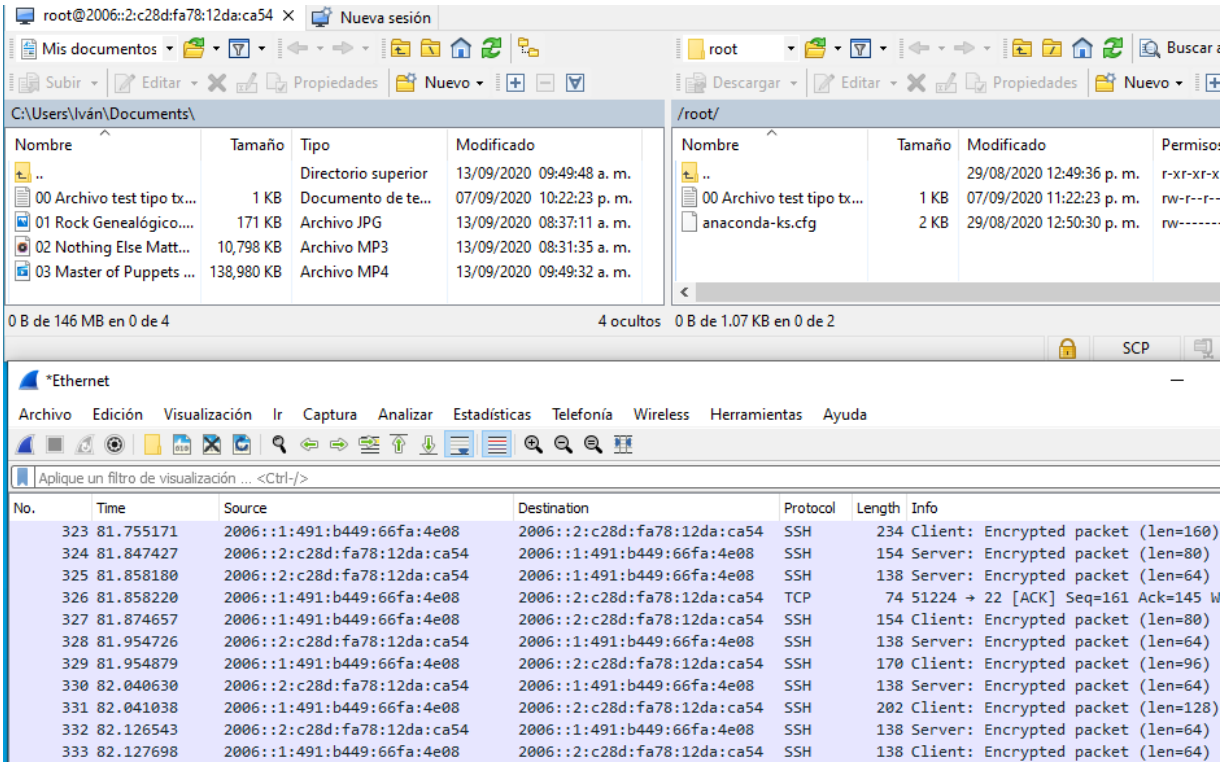


Figura 61 Captura de paquetes mediante Wireshark al enviar archivo de texto de PW01 hacia PW03

4.4.3.2 Archivo de imagen

La transferencia del archivo de imagen (171 KB) si mostró un recuadro donde se logró visualizar velocidad de transferencia (227 KB/s). Tal y como se muestra en la Figura 62.

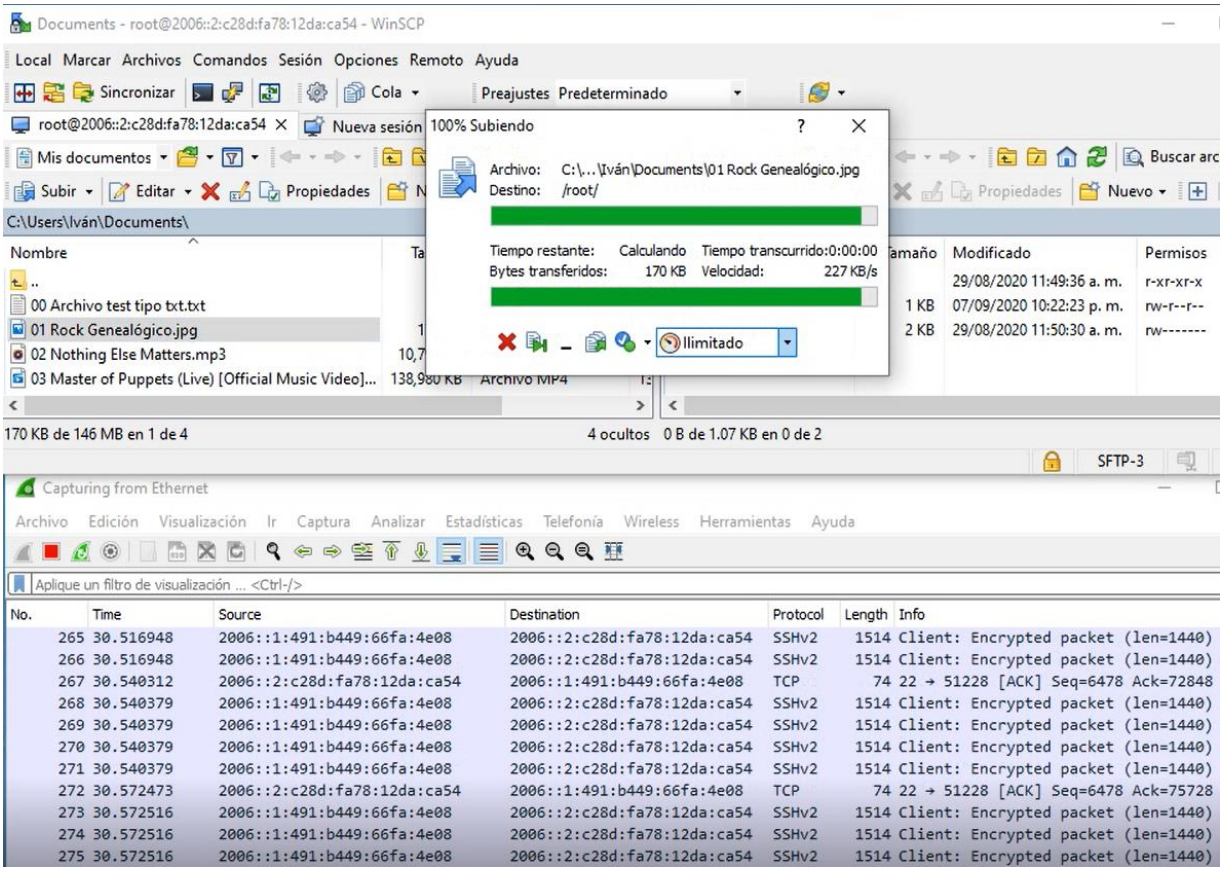


Figura 62 Captura de paquetes mediante Wireshark al enviar archivo de imagen de PW01 hacia PW03

4.4.3.3 Archivo de audio

Para el archivo de audio (10,798 KB) se registraron desconexiones momentáneas y su tiempo de transferencia fue de aproximadamente 5 minutos, esto derivado a que la máquina virtual tiene una conexión de tipo *ethernet*, siendo específicos las conexiones entre *routers* cuentan con una velocidad de transferencia de 1,000 Mbps, mientras que la conexión con las máquinas virtuales cuentan con una conexión *ethernet* de 10 Mbps, en la Figura 63 se muestra la configuración del adaptador que tiene la máquina virtual *PW01*.

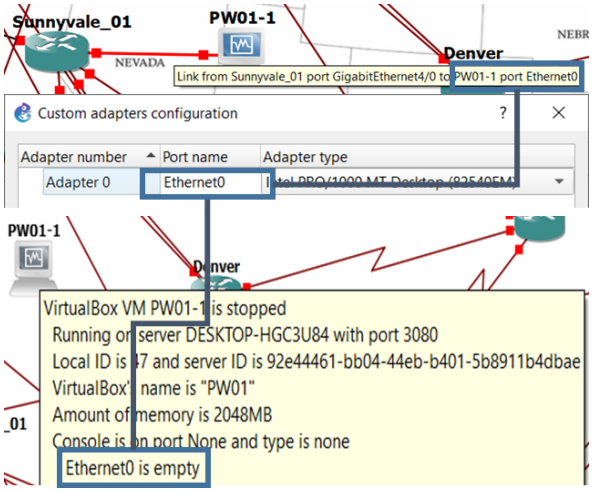


Figura 63 adaptador ethernet de máquina virtual

La Figura 64 muestra la transferencia del archivo de audio de *PW01* hacia *PW03*. En este caso a diferencia de los anteriores se puede apreciar la información de tiempo restante, el tiempo transcurrido, los bytes transferidos y la velocidad de transferencia.

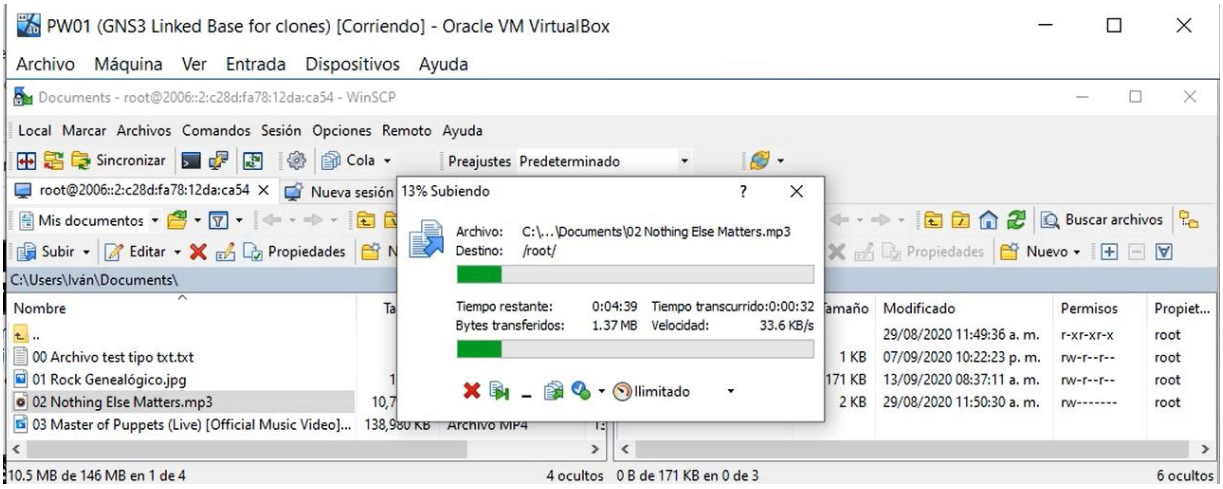


Figura 64 Transferencia de archivo de audio

Durante este experimento se estuvieron registrando el envío de paquetes mediante Wireshark capturando el momento en el que se presentaron las desconexiones tal y como se muestra en

Figura 65. Estas desconexiones se dan después de que *PW01* pierde conexión con *PW03* por más de 15 segundos; posterior a esto se genera una reconexión automática durante un periodo de 60 segundos en los cuales si no logra la reconexión la sesión SSH se pierde y es necesario volver a iniciar sesión.

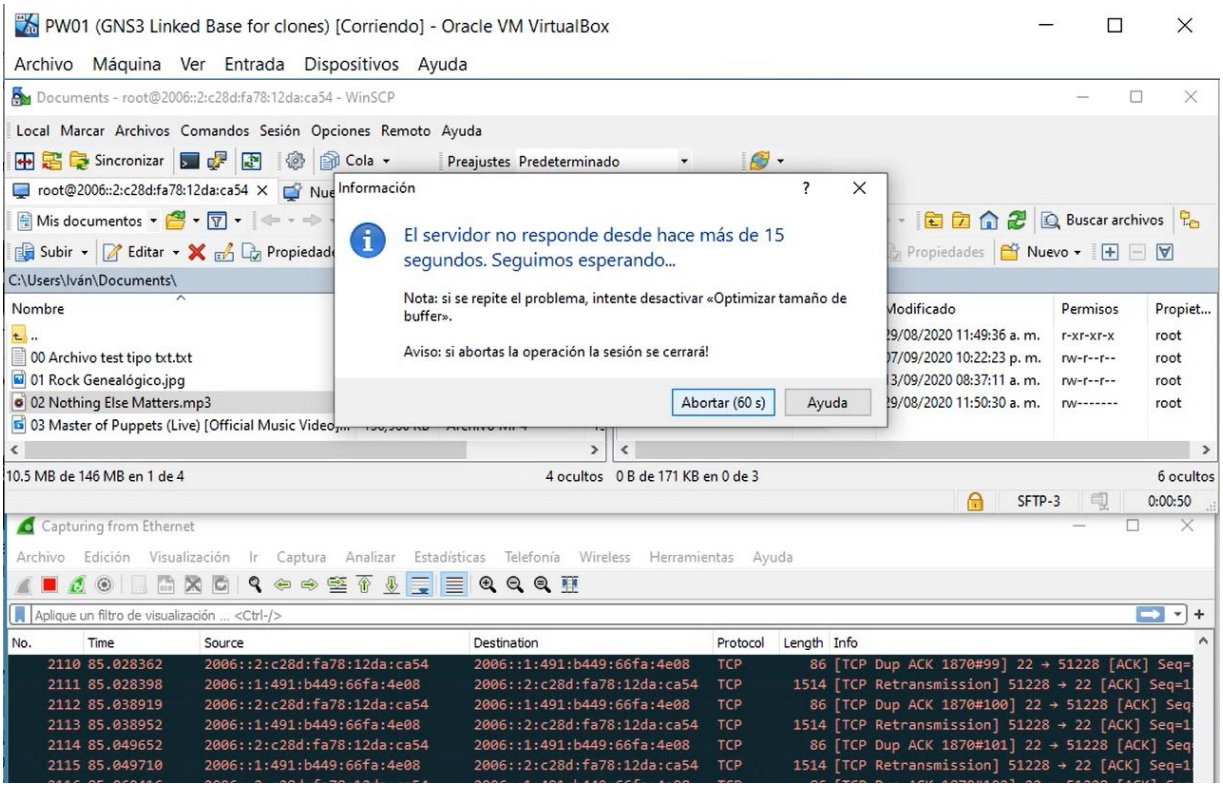


Figura 65 Captura de paquetes mediante Wireshark al enviar archivo de audio de PW01 hacia PW03

4.4.3.4 Archivo de video

El archivo de video (138,980 KB), tuvo un tiempo promedio de transferencia de 160 minutos.

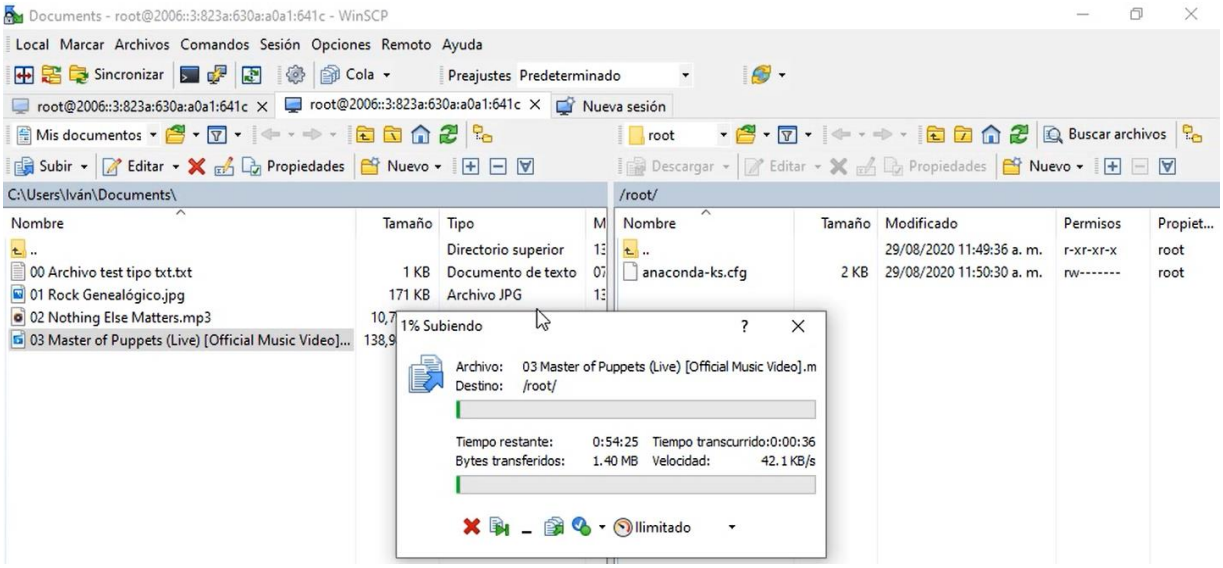


Figura 66 Transferencia de archivo de video de PW01 hacia PW03

Al igual que en la transferencia del archivo de audio, el archivo de video presentó desconexiones como se muestra en la Figura 67.

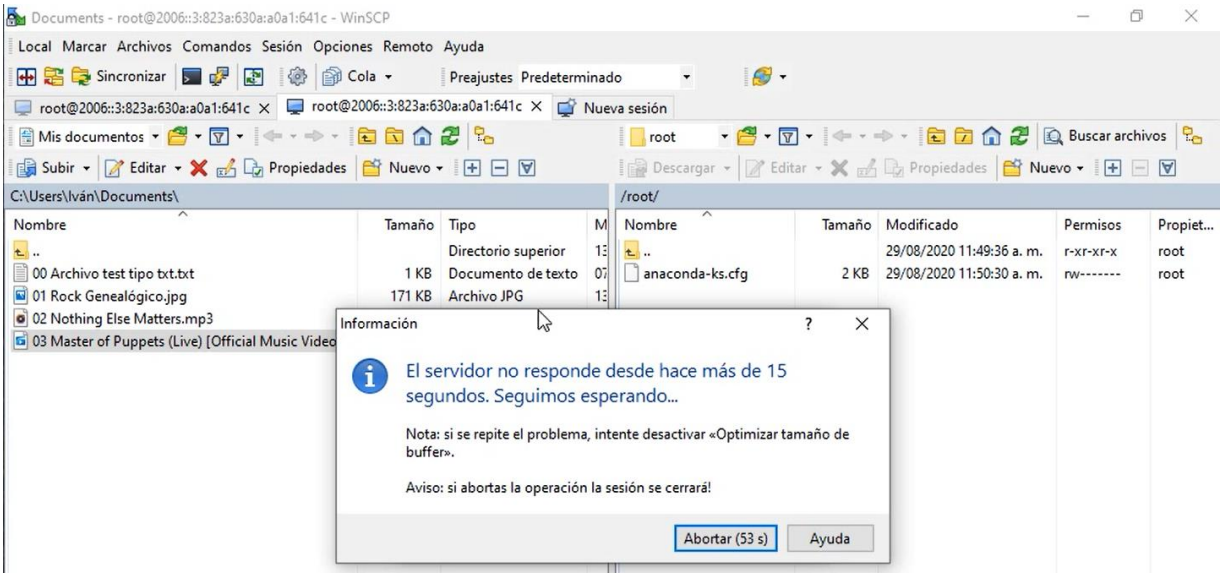


Figura 67 Transferencia de archivo de video de PW01 hacia PW03 desconexión

Por lo tanto, para estos cuatro casos se construyó la Tabla 24 con los datos recabados al realizar la transferencia de archivos. Con cada archivo se realizó el ejercicio 5 veces y se plasmó el resultado promedio de dichos ejercicios.

Tipo	Tamaño [KB]	Envió [Si/No]	Tiempo promedio [min]
Texto (*txt*)	1	Si	0.01
Imagen (*jpg*)	171	Si	0.05
Audio (mp3)	10,798	Si	5
Video (mp4)	138,980	si	160

Tabla 24 Transferencia de archivos mediante WinSCP

4.5 Resultados de la emulación de la gestión de la red PW

Las pruebas de gestión se realizaron en *PW01* con *iReasoning MIB Browser*, del mismo modo, se utilizó Wireshark para corroborar los paquetes enviados mediante SNMP. En la Figura 68 se muestran la versión SNMP y el usuario habilitado durante la configuración.

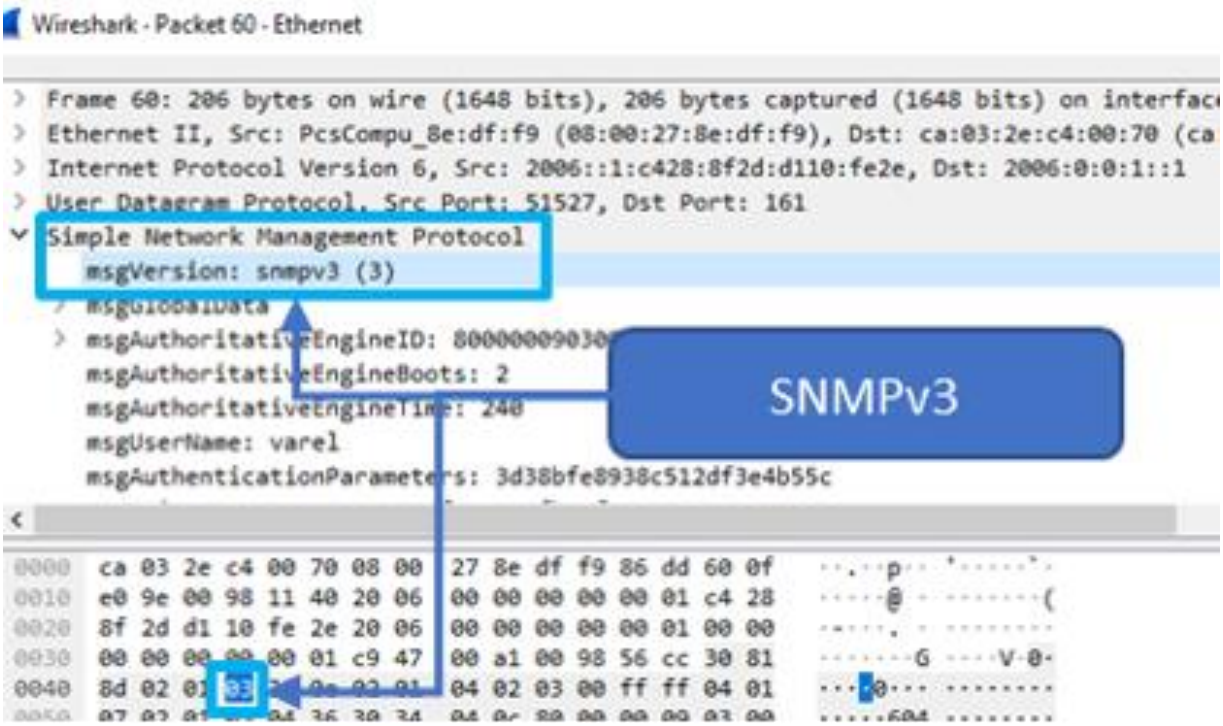


Figura 68 Captura de paquetes mediante Wireshark de SNMPv3

También mediante Wireshark se visualizó el nombre de usuario que se configuro en el *router* para SNMPv3 como se muestra en Figura 69.

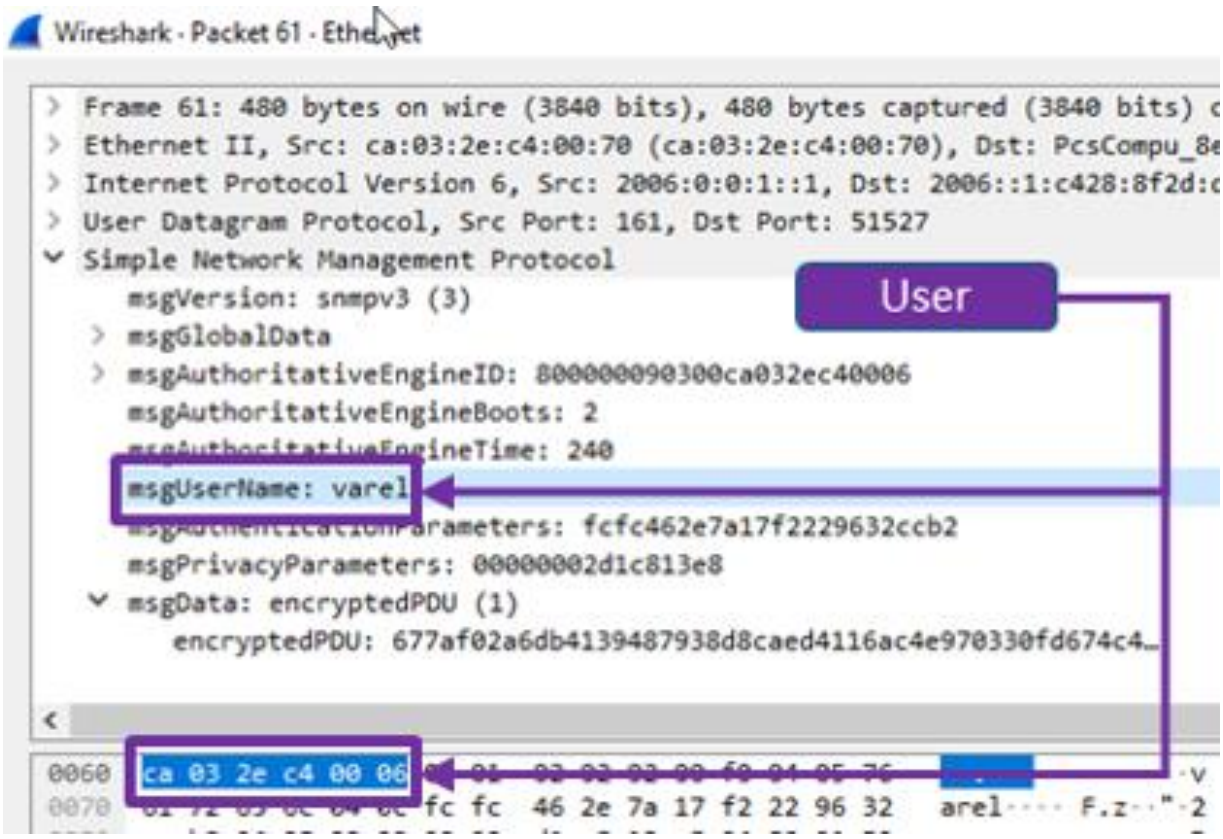


Figura 69

Posteriormente se procedió a realizar las pruebas de los OIDs mencionados en la tabla de la sección 3.1.5.

4.5.1 Prueba de gestión: *SysName* y Estado 5

Para la prueba de *SysName,* se solicitó el nombre asignado al *router* con dirección IPv6 2004:0:0:2::2 el cual mediante un *get* mostro el valor *New_Zealand*. Con la herramienta Wireshark se estuvieron monitoreando los paquetes capturados, detectando la solicitud del valor del *router* correspondiente a la dirección antes mencionada, tal y como en la Figura 70.

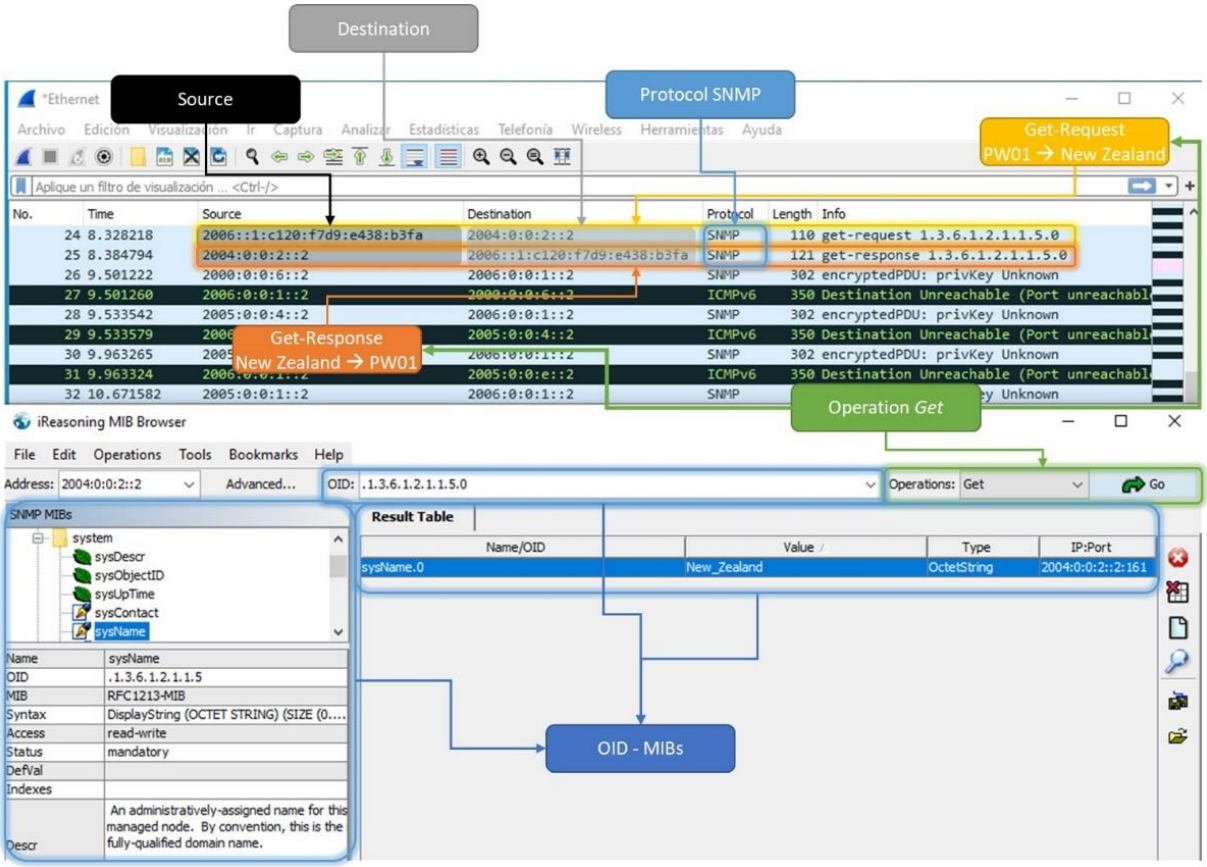


Figura 70 Captura de paquetes mediante Wireshark de una solicitud SysName y uso de iReasoning MIB con SNMP

Dentro del OID *SysName* se puede realizar un cambio de valor en el nombre del nodo por lo que se realizó la modificación de *New_Zealand* a un nuevo valor con el nombre de *NewZealand_snmp* tal y como se muestra en Figura 71.

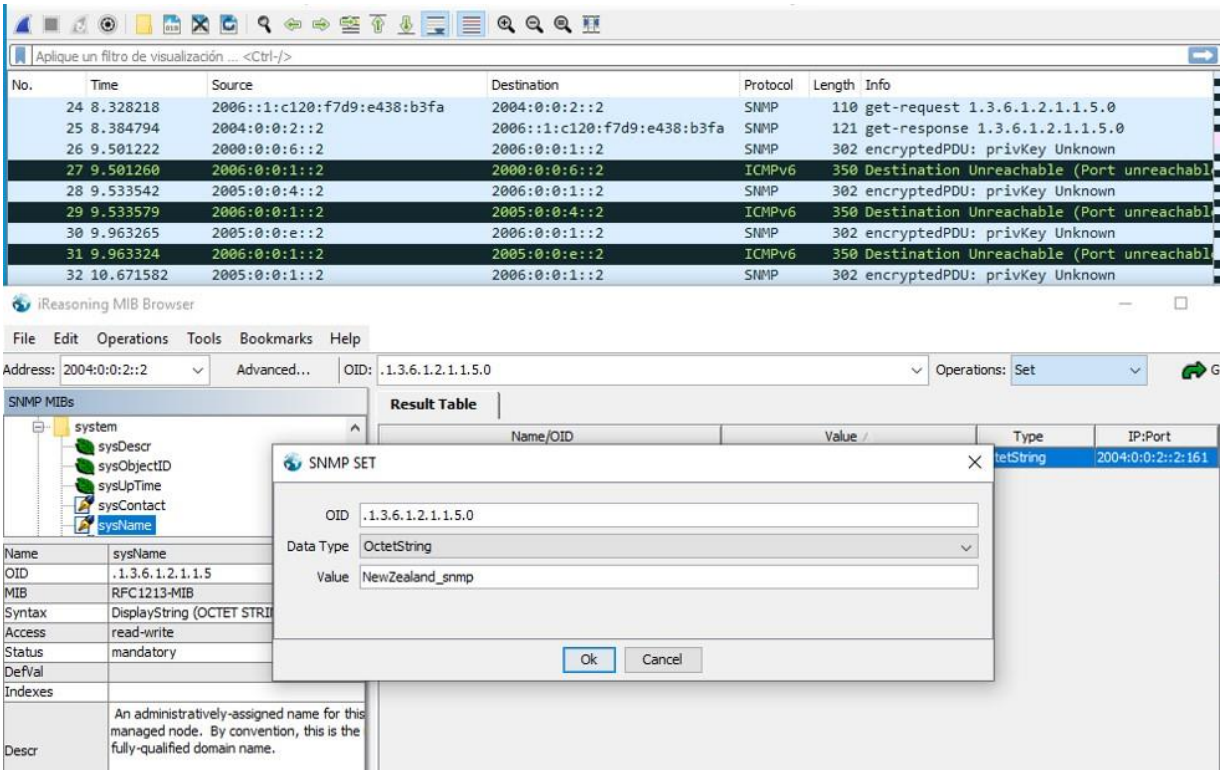


Figura 71 Modificación de valor en el nombre del router de New_Zealand a NewZealand_snmp

Una vez realizada la modificación *WinSCP* muestra un recuadro notificando que se realizó la operación con éxito como se visualiza en Figura 72. Así mismo durante esta modificación se realizó la captura de paquetes mediante Wireshark logrando observar la solicitud y la respuesta de este OID.

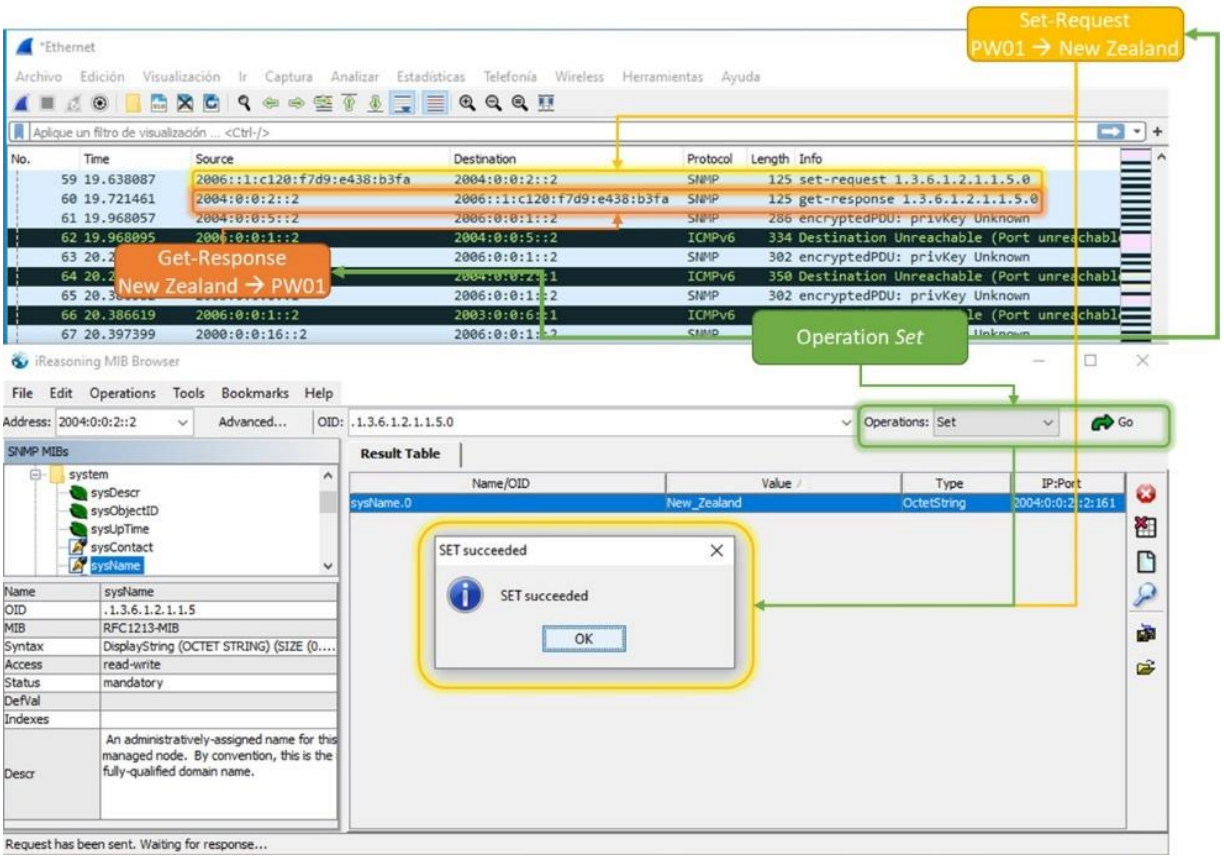


Figura 72 Se establece un nuevo valor para modificar el nombre del router

Para validar que se realizó la modificación en el valor se vuelve a ejecutar la solicitud de información del valor del nodo, en la Figura 73 se muestra que se realizó el cambio con éxito.

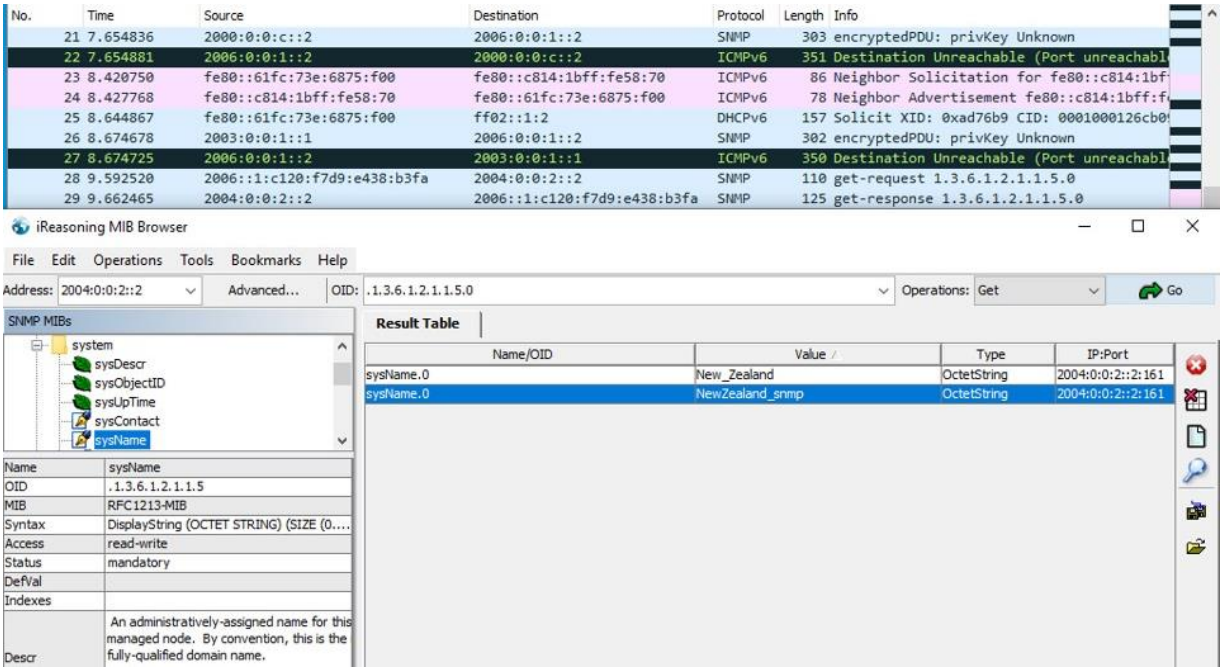


Figura 73 Validación de modificación realizada

Finalmente, para el Estado 5 observamos un incremento en el rendimiento del equipo de 3% como se muestra en Figura 74.

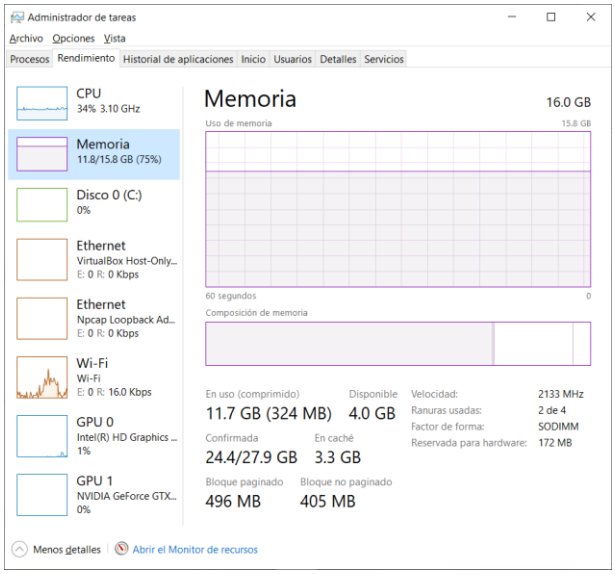


Figura 74 Estado 5, Rendimiento del uso de la memoria mientras se realizó el uso de un OID de SNMP

4.5.2 Prueba de gestión: *IfNumber*

IfNumber en el *router* de *New_Zealand* muestra un valor de 16 interfaces de red presentes en el sistema, como se muestra en la Figura 75

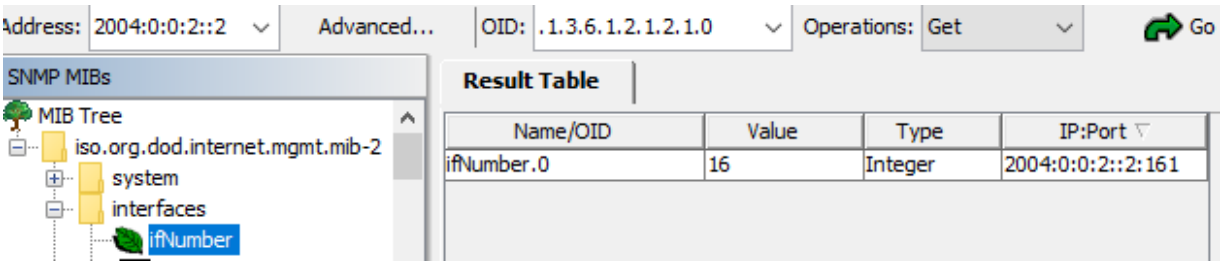


Figura 75 Uso de iReasoning MIB con SNMP OID IfNumber

4.5.3 Prueba de gestión: *IfTable*

IfTable de la Figura 76 de *New_Zealand* muestra el listado de las interfaces presentes en el sistema que con *IfNumber* se obtuvo.

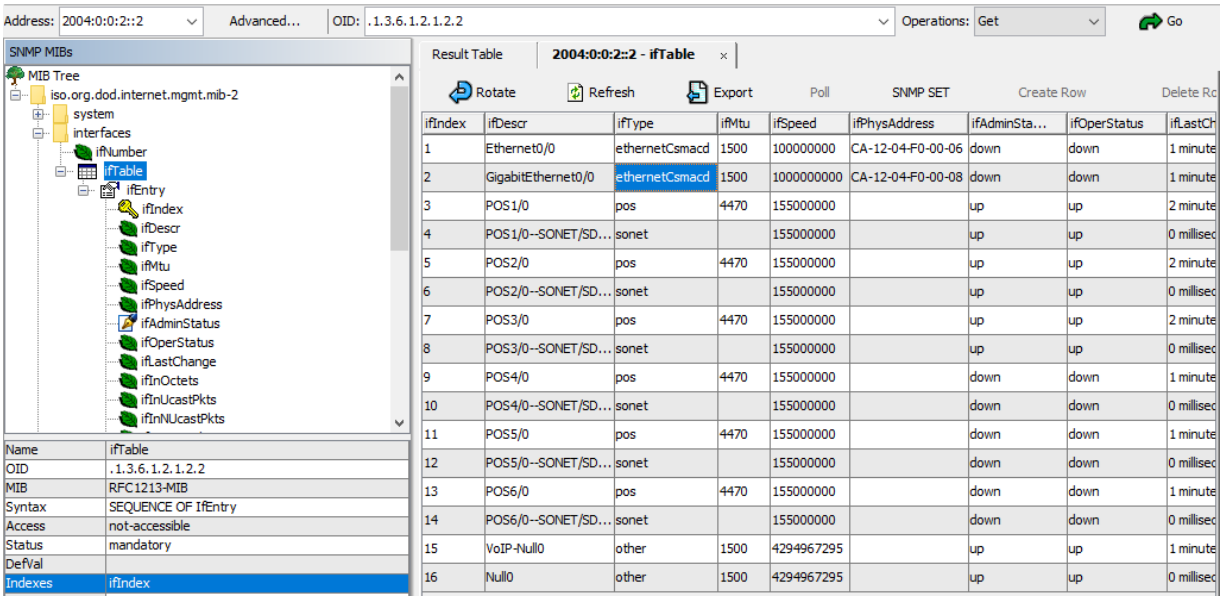


Figura 76 Uso de iReasoning MIB con SNMP OID IfTable

4.5.4 Prueba de gestión: *IfDescr*

IfDescr en el *router* de *New_Zealand* muestra la información de cada interfaz presente, como se muestra en la Figura 77.

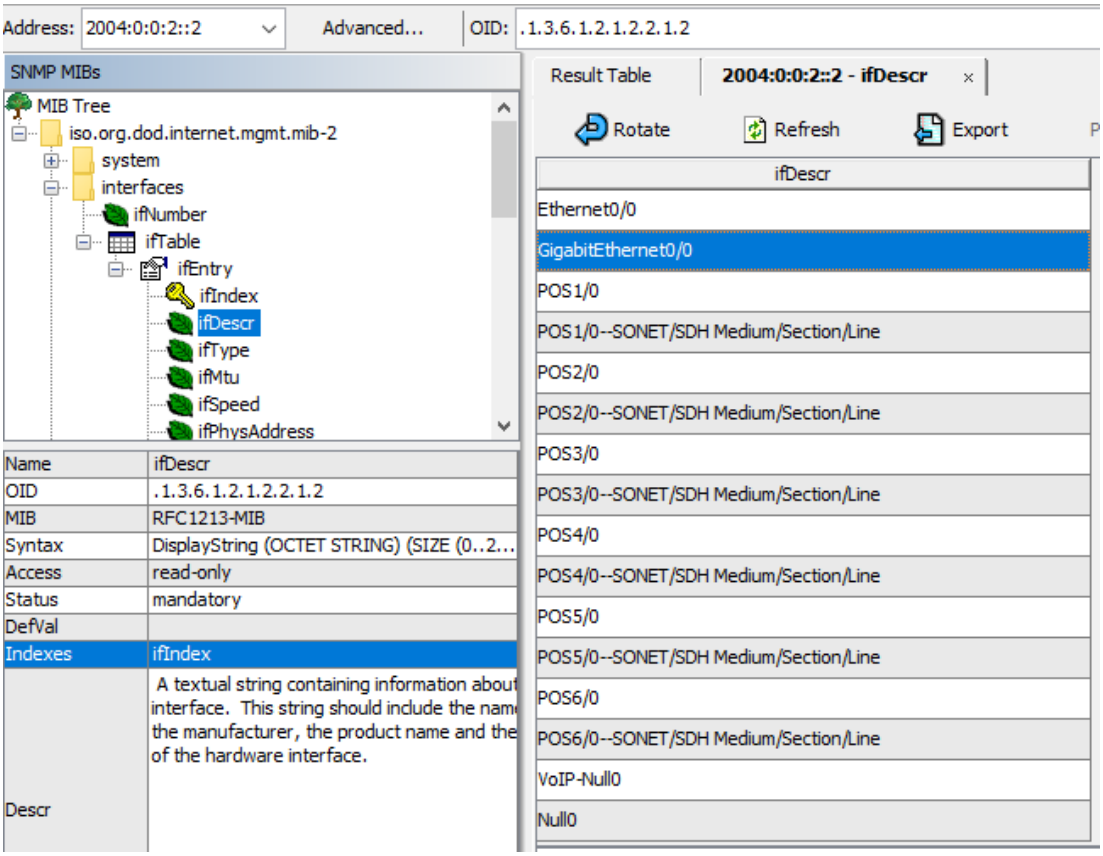


Figura 77 Uso de iReasoning MIB con SNMP OID IfDescr

4.5.5 Prueba de gestión: *IfOperStatus*

IfOperStatus de la Figura 78 muestra el estado de cada interfaz

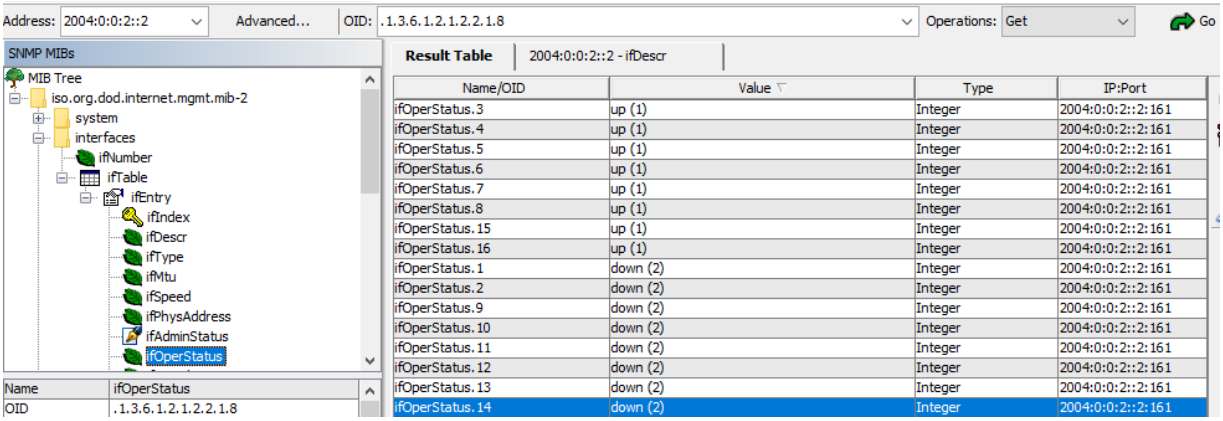


Figura 78 Uso de iReasoning MIB con SNMP OID IfOperStatus

4.5.6 Prueba de gestión: *SysUpTime*

SysUpTime de la Figura 79 nos indica el tiempo que ha estado activo el sistema de gestión en el *router*.

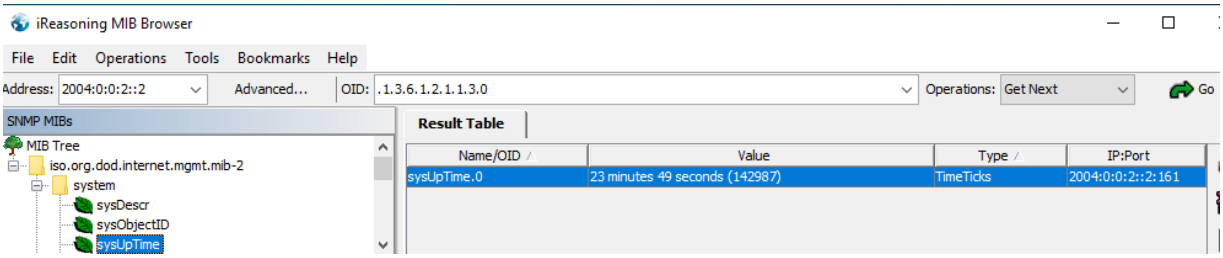
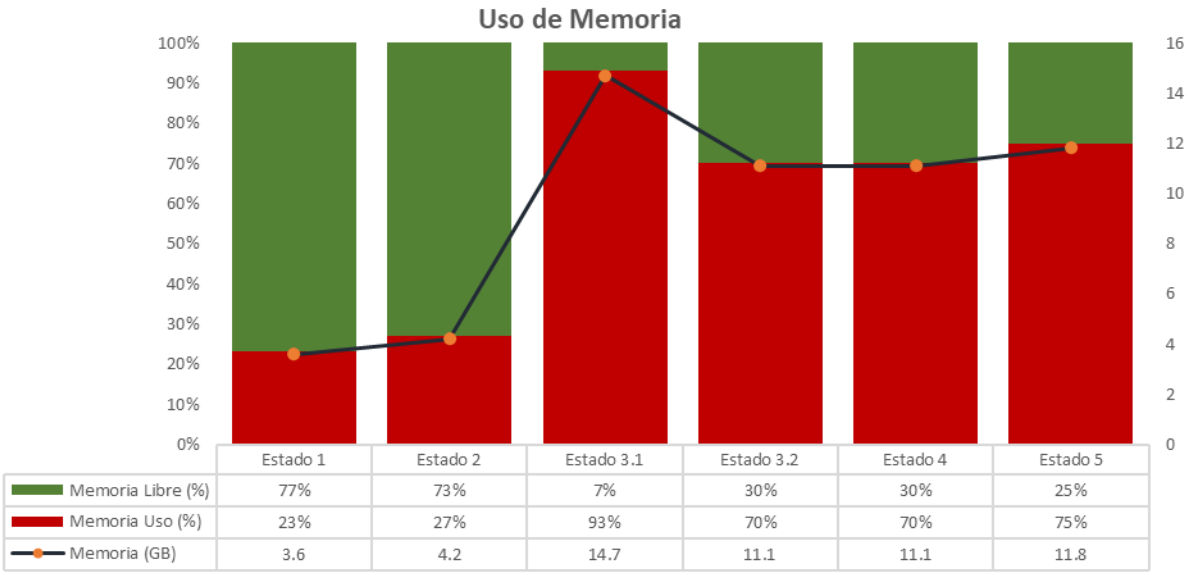


Figura 79 Uso de iReasoning MIB con SNMP OID sysUpTime

4.6 Uso de recursos del equipo utilizado a través de los 5 estados

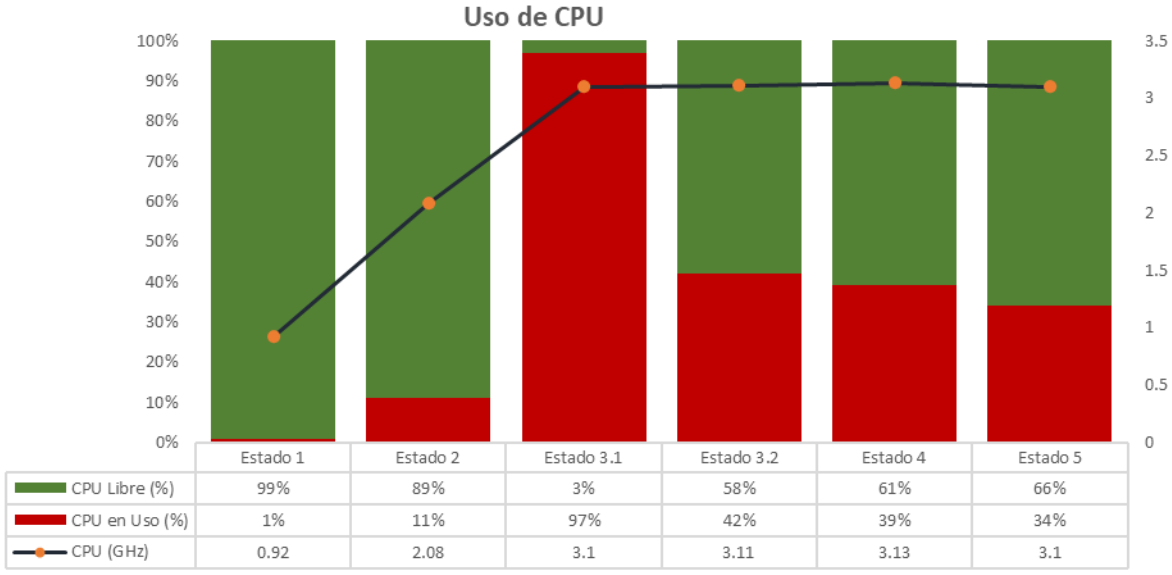
Finalmente, respecto del uso de recursos computacionales se generaron dos gráficos el primero indicando el uso de memoria RAM (Figura 80) y el segundo el uso del CPU del equipo (Figura 81), tomados durante el paso de los estados que anteriormente se mencionaron. Para el uso de memoria RAM de un 100% que son los 16 GB contenidos en el equipo utilizado, se llegó al máximo de uso del 93% en el Estado 3.1 que es cuando se iniciaron de golpe todos los *routers* y todas las máquinas virtuales, el uso mínimo fue de 23% durante el Estado 1 con GNS3 en estado inactivo.



	Estado 1	Estado 2	Estado 3.1	Estado 3.2	Estado 4	Estado 5
Memoria Libre (%)	77%	73%	7%	30%	30%	25%
Memoria Uso (%)	23%	27%	93%	70%	70%	75%
Memoria (GB)	3.6	4.2	14.7	11.1	11.1	11.8

Figura 80 Uso de Memoria RAM del equipo utilizado

Mientras que, para el uso de actividad de la CPU, se llegó al máximo de uso del 97% en el Estado 3.1, el uso mínimo fue de 1% durante el Estado 1.



	Estado 1	Estado 2	Estado 3.1	Estado 3.2	Estado 4	Estado 5
CPU Libre (%)	99%	89%	3%	58%	61%	66%
CPU en Uso (%)	1%	11%	97%	42%	39%	34%
CPU (GHz)	0.92	2.08	3.1	3.11	3.13	3.1

Figura 81 Uso de CPU del equipo utilizado

4.7 Conclusiones

Durante esta tesis se emuló una infraestructura Backbone real para PW (2020), iniciando con la configuración de los protocolos de OSPFv3 para sistemas intrautonomos y BGP-4 para sistemas interautonomos. La conectividad entre los *routers* se validó mediante el uso de los comandos *ping* y *traceroute*, *ping* para verificar la conexión entre dispositivos y *traceroute* que muestra las rutas en las que viajan los paquetes de un dispositivo a otro. También se realizó transferencia de archivos de texto, imagen, audio y video mediante la herramienta *WinSCP*, en este caso se presentaron transferencias con baja velocidad, esto derivado del ancho de banda que se tiene de la máquina virtual hacia los *routers*; en el caso de la transferencia del archivo de tipo texto fue prácticamente inmediata de 0.01 minutos, mientras que para el caso de la transferencia del archivo de tipo imagen se llevó a cabo en 0.05 minutos, en el caso de la transferencia de archivos de audio y video se presentaron desconexiones momentáneas de host a host con una duración de 5 y 140 minutos respectivamente. Durante estos ejercicios se pudo capturar los paquetes mediante *Wireshark* y llevar un monitoreo de la red.

Para la gestión de la red mediante SNMP se realizaron 6 pruebas con los OIDs *SysName*, *IfNumber*, *IfTable*, *IfDescr*, *IfOperStatus* y *SysUpTime*, para solicitar consultas y realizar modificaciones en los dispositivos así se logró administrar la red.

Así mismo se tuvieron limitaciones del emulador GNS3, como el ancho de banda de los enlaces de *router* a *router* sustituyendo el ancho de banda de 100 Gbps de los enlaces reales mediante fibra óptica por enlaces que solo poseen hasta 1 Gbps. Además, el ancho de banda que se tiene por parte de las máquinas virtuales es de 10 Mbps lo cual reduce el tiempo de transferencias entre equipos. Otra limitante de GNS3 es la cantidad de interfaces que se pueden manejar por *router*, teniendo un máximo de 7 interfaces por *router*.

Los mecanismos de mejora para realizar la emulación que se deben considerar, son el equipo que realiza el procesamiento, en este caso el que se utilizó quedo al límite durante el funcionamiento por lo que, para este mismo sistema se recomienda un equipo con una RAM por lo menos de 32 GB, o varios equipos que realicen la emulación repartiendo la demanda de los recursos de modo que los sistemas no estén trabajando al límite.

Uno de los objetivos que se buscaban fue recolectar evidencias para el grupo de desarrollo de GNS3 para mejorar sus características.

Durante esta tesis se utilizaron los sistemas operativos Windows 10, Centos 8 y Solaris. Además de las herramientas GNS3, VirtualBox, Wireshark, *WinSCP* y iReasoning MIB.

Finalmente puedo decir que desarrollé las habilidades para desempeñarme como un administrador de redes, cumpliendo con los objetivos planteados.

El tiempo invertido de esta tesis fue de aproximadamente 2,000 horas.

Referencias

[1] W. Stallings, Comunicaciones y redes de computadores, 7th ed. Madrid: Prentice Hall, 2004, p. 597.

[2] Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, *Digital systems: principles and applications*, 10th ed. New Jersey: Pearson Prentice Hall., 2007, pp. 3-7. ISBN 0131725793

[3] J. C. R. Licklider, *Man-Computer Symbiosis*, in IRE Transactions on Human Factors in Electronics, vol. HFE-1, no. 1, pp. 4-11, March 1960. doi: 10.1109/THFE2.1960.4503259

[4] UCLA, *Birthplace of the Internet, 1969*, IEEE Milestones, IEEE History Center, IEEE, 2009.

[5] SRI, *Inception of the ARPANET, 1969*, IEEE Milestones, IEEE History Center, IEEE, 2009.

[6] Internethalloffame.org. (2019). Raymond Tomlinson | Internet Hall of Fame. [En línea] Disponible en: https://www.internethalloffame.org/inductees/raymond-tomlinson [Acceso 20 Dec. 2019].

[7] Castillo Velázquez José Ignacio, *Redes de Datos Contexto y Evolución*, Tercera Edición México, Samsara Editorial, 2019, p. 54. ISBN: 978-970-94-2968-8.

[8] V. Cerf and R. Kahn, *A Protocol for Packet Network Intercommunication*, in IEEE Transactions on Communications, vol. 22, no. 5, pp. 637-648, May 1974. doi: 10.1109/TCOM.1974.1092259

[9] Coronado, A. (2019). Evolución del Internet e IPv6. [En línea] Ipv6.mx. Disponible en: http://www.ipv6.mx/index.php/informacion/noticias/1-latest-news/538-evolucion-del-internet-e-ipv6 [Acceso 20 Dec. 2019].

[10] Es.net, 2018. [En línea]. Disponible en: https://es.net/assets/Timeline-30/77-89/1986ESnetdirective.pdf. [Acceso: 02- Oct- 2018].

[11] Berners-Lee, Tim (March 1989). Information Management: A Proposal. World Wide Web Consortium. Retrieved 24 August 2010.

[12] IETF Journal - About Us. (2019). [En línea]. Disponible en: https://www.ietfjournal.org/about-us/ [Acceso: 02- Oct- 2018].

[13] History | Internet Society. (2019). [En línea]. Disponible en: https://www.internetsociety.org/es/ [Acceso: 02- Oct- 2018].

[14] About Us | CANARIE, Canarie.ca, 2018. [En línea]. Disponible en: https://www.canarie.ca/about-us/. [Acceso: 02- Oct- 2018].

[15] S. Bradner, Harvard University, A. Mankin. *IP: Next Generation (IPng) White Paper Solicitation*, RFC 1550, December 1993

[16] S. Deering, Xerox PARC, R. Hinden, Ipsilon Networks. Internet Protocol, Version 6 (IPv6) Specification, RFC 1883, December 1995

[17] Internet2 Community Timeline | Internet2, Internet2.edu, 2018. [En línea]. Disponible en: https://www.internet2.edu/about-us/internet2-community-timeline/. [Acceso: 02- Oct- 2018].

[18] S. Deering, Cisco, R. Hinden, Nokia. Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, December 1998.

[19] M. Pourailly, História de RedCLARA, Redclara.net, 2018. [En línea]. Disponible en: http://redclara.net/index.php/es/somos/redclara-la-organizacion/historia-de-redclara. [Acceso: 02- Oct- 2018].

[20] Pacificwave.net. (2019). History & Vision | Pacific Wave. [En línea] Disponible en: http://pacificwave.net/history-vision [Acceso 25 nov. 2019].

[21] Icann.org. (2019). Análisis del uso de las funciones de la IANA. [En línea]. Disponible en: https://www.icann.org/news/blog/analisis-del-uso-de-las-funciones-de-la-iana [Acceso 30 nov. 2018].

[22] World IPv6 Launch, Worldipv6launch.org, 2018 [En línea]. Disponible en: http://www.worldipv6launch.org/. [Acceso: 02- Oct- 2018].

[23] P. Baran, Acerca de las comunicaciones de redes distribuidas, IEEE Trans. Comm. Systems, marzo de 1964.

[24] Internet Society. (2019). Brief History of the Internet | Internet Society. [En línea] Disponible en: https://www.internetsociety.org/internet/history-internet/brief-history-internet/ [Acceso 20 jun. 2019].

[25] Lehtisalo, K. (2019). The History of nordunet. [En línea] Nordu.net. Disponible en: http://www.nordu.net/history/TheHistoryOfNordunet_simple.pdf [Acceso 20 jun. 2019].

[26] Nsf.gov. (2019). A Brief History of NSF and the Internet. [En línea] Disponible en: https://www.nsf.gov/news/news_summ.jsp?cntn_id=103050 [Acceso 20 jun. 2019].

[27] Jisc. (2019). About us. [En línea] Disponible en: https://www.ja.net/about [Acceso 20 jun. 2019].

[28] Switch.ch. (2019). Foundation - About us - SWITCH. [En línea] Disponible en: https://www.switch.ch/about/foundation/ [Acceso 20 jun. 2019].

[29] Canarie.ca. (2019). About Us | CANARIE. [En línea] Disponible en: https://www.canarie.ca/about-us/ [Acceso 20 jun. 2019].

[30] Aarnet.edu.au. (2019). [En línea] Disponible en: https://www.aarnet.edu.au/about-us/history/ [Acceso 20 jun. 2019].

[31] Pourailly, M. (2018). ¿Qué son y para qué sirven las Redes de Investigación y Educación? [En línea]. Disponible en: http://redclara.net/index.php/es/red/redes-de-investigacion-y-educacion/que-son-y-para-que-sirven [Acceso: 31- Jul- 2018].

[32] REFEDS – The Voice of Research and Education Identity Federations, Refeds.org, 2018. [En línea]. Disponible en: https://refeds.org [Acceso: 31- Jul- 2018].

[33] Pacificwave.net. (2020). Home | Pacific Wave. [En línea] Disponible en: http://pacificwave.net/ [Acceso 11 Jan. 2020].

[34] Pacificwave.net. (2020). History & Vision | Pacific Wave. [En línea] Disponible en: http://pacificwave.net/history-vision [Acceso 11 Jan. 2020].

[35] Pacificwave.net. (2020). Node Sites | Pacific Wave. [En línea] Disponible en: http://pacificwave.net/nodesites [Acceso 11 Jan. 2020].

[36] Pacificwave.net. (2020). Maps | Pacific Wave. [En línea] Disponible en: http://pacificwave.net/about-maps [Acceso 11 Jan. 2020].

[37] Westernregional.net. (2020). Western Regional Network home. [En línea] Disponible en: https://www.westernregional.net/ [Acceso 9 Feb. 2020].

[38] Noé Galicia Gutiérrez (2015), *Emulación de BB de la red avanzada de Internet2 en México*, Tesis de Licenciatura, Universidad Autónoma de la Ciudad de México, Ciudad de México.

[39] Juan Arnulfo López Ruiz (2015), *Implementación de un modelo IPv4 Multicast*, Tesis de Licenciatura, Universidad Autónoma de la Ciudad de México, Ciudad de México.

[40] José Joaquín Sanchez Trejo (2015), *Emulación de la Red Avanzada CLARA*, Tesis de Licenciatura, Universidad Autónoma de la Ciudad de México, Ciudad de México.

[41] J. I. Castillo and N. Galicia, *Routing algorithms applied to an advanced academic network known as CUDI*, IEEE Latin America Transactions, vol 14, no; pp 2974-2679, June 2016 doi: 10.1109/TLA.2016.7555284.

[42] J. I. Castillo and J.J Sánchez Trejo, *Emulation for CLARA's operation, the advanced network for Latin America*, 2016 IEEE ANDESCON, Arequipa, 2016, pp. 1-4. doi 10.1109/ANDESCON 2016.7836205

[43] José-Ignacio Castillo-Velázquez, Daniel-Javier Serrano-Martinez, Augusto Morales, Emulation of Backbone´s connectivity and management for the advanced network in Latin America: 2016's topology. International Conference on Sensors Networks Smart and Emerging Technologies (SENSET 2017) Beirut, Lebanon, 2017. pp 1-4 TBP.

[44] Daniel Javier Serrano Martínez (2017), *Integración de las redes avanzadas en América: CANARIE, I2 y CLARA*, Tesis de Licenciatura, Universidad Autónoma de la Ciudad de México, Ciudad de México.

[45] Fernando De La Cruz Alejandre (2018), *Análisis de gestión en la red avanzada europea GEANT*, Tesis de Licenciatura, Universidad Autónoma de la Ciudad de México, Ciudad de México.

[46] Victor Raúl Cobos Panduro, Simulación y Emulación de la Red Universitaria Nacional (REUNA) de Chile y el nivel de comprensión del funcionamiento de redes avanzadas, Fac. de Ingeniería en Informática y Sistemas, Universidad Nacional Agraria de la Selva, UNAS, Perú, July, 2018.

[47] Adrian Ramírez, Monitoreo del backbone de CLARA: emulación, Ingeniería en Electrónica y Telecomunicaciones, CCyT-UACM, Mexico, August. 2018.

[48] Manuel Trigueros, Desarrollo e Implementación del Sistema de Gestión de Conferencias: UTILCON, Ingeniería en Electrónica y Telecomunicaciones, CCyT-UACM, Mexico, March, 2019.

[49] Yovaka, Simulación y Emulación SNMP y Syslog, Facultad de Ingeniería en Informática y Sistemas, Universidad Nacional Agraria de la Selva, UNAS, Perú, September, 2019.

[50] Juan Revilla Melo, Gestión de la red avanzada AFRICACONNECT con IPv6, Ingeniería en Electrónica y Telecomunicaciones, CCyT-UACM, Mexico, agosto 2020.

[51] Isabel /Armando, Gestión de la red avanzada GEANT con IPV6, Ingeniería en Electrónica y Telecomunicaciones, CCyT-UACM, Mexico, agosto 2020.

[52] José-Ignacio Castillo-Velazquez, Elf-Yovanka Ramírez Díaz, William Rogelio Marchand Niño, Use of GNS3 Cloud Environment for Network Mangement Emulation when Comparing SNMP vs Syslog Applied Over an Advanced Network, 2019 IEEE CONCAPAN XXXIX Guatemala, 2019, pp1-4.

[53] José-Ignacio Castillo-Velazquez, Daniel Javier Serrano Martinez and Monica Huerta, Management Emulation for Advanced Networks Interconnection in all America: 2019 topology, 2019 IEEE CONCAPAN XXXIX Guatemala, 2019, pp1-4.

[54] José-Ignacio Castillo-Velazquez and Fernando DeLaCruz-Alejandre and Monica Huerta, An Approach to Management Assessment for GEANT Backbone Using GNS3 for SNMPv3, 2018 IEEE CONCAPAN XXXVIII El Salvador, 2018, pp1-4. 10.1109/CONCAPAN.2018.8596667.

[55] José-Ignacio Castillo-Velazquez, Victor-Raul Cobos-Pandoro, William Rogelio Marchand Niño, IPV6 connectivity and management emulation for REUNA, The Chilean Advanced Network, 2018 IEEE XXV International Conference on Electronics Electrical Engineering and Computing (INTERCON) Lima, 2018, pp1-4. 10.1109/INTERCON.2018.8526390

[56] José-Ignacio Castillo-Velazquez, Daniel-Javier Serrano-Martinez and Augusto Morales, Emulation of the connectivity of backbone and management for the layer 3 service of INTERNET2: 2016´s topology, 2017 IEEE 37th Central America and panama Convention (CONCAPAN XXXVII) Managua, 2017, pp1-4. DOI: 10.1109/CONCAPAN.2017.8278476
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8278476&isnumber=8278456

[57] José-Ignacio Castillo-Velazquez, Daniel-Javier Serrano-Martinez and Augusto Morales, Emulation of Backbone´s Connectivity and Management for the Advanced Network in Latin America: 2016´s topology, 2017 Sensors Networks Smart and Emerging Technologies (SENSET) Beirut, 2017, pp. 1-4. DOI:10.1109/SENSET.2017.8125029
http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8125029&isnumber=8124995

[58] Jose Ignacio Castillo and Noe Galicia, Routing algorithms applied to an advanced academic network known as CUDI, IEEE Latin America Transactions, Vol 14, No. 6, pp 2974-2979, June 2016. ISSN 1548-0992. DOI:10.1109/TLA.2016.7555284 [IF 0.59]

[59] José-Ignacio Castillo-Velazquez and José-Joaquin Sanchez-Trejo, Emulation for CLARA´s operation, the advanced network for Latin America, 2016 IEEE ANDESCON Proceedings, Electronic ISBN 978-1-5090-2532-9; Print ISBN: 978-1-5090-2533-6, Arequipa, Peru, Oct. 19-21, 2016. DOI:10.1109/ANDESCON.2016.7836205 http://ieeexplore.ieee.org/document/7836205/

[60] Postel, J., Internet Protocol, STD 5, RFC 791, September 1981.

[61] Icann.org. (2019). Análisis del uso de las funciones de la IANA. [En línea] Disponible en: https://www.icann.org/news/blog/analisis-del-uso-de-las-funciones-de-la-iana [Acceso 7 Sep. 2019].

[62] Hinden, R. and S. Deering, IP Version 6 Addressing Architecture, RFC 4291, February 2006.

[63] Castillo Velázquez José Ignacio, *Switching and Routing introducción*, México, Samsara Editorial, 2016, ISBN: 978-970-94-2977-0.

[64] Community.cisco.com. (2019). Enrutamiento: Conceptos Fundamentales. [En línea] Disponible en: https://community.cisco.com/t5/documentos-routing-y-switching/enrutamiento-conceptos-fundamentales/ta-p/3166553 [Acceso 9 Feb. 2019].

[65] Autonomous System (AS) Numbers, Iana.org, 2018. [En línea]. Disponible en: https://www.iana.org/assignments/as-numbers/as-numbers.xhtml. [Acceso: 31- Jul- 2018].

[66] Guidelines for creation, selection, and registration of an Autonomous System (AS). RFC 1930. [En línea] Disponible en: <https://tools.ietf.org/html/rfc1930> [Acceso 30 September 2020].

[67] Tools.ietf.org. 2020. RFC 4893 - BGP Support For Four-Octet AS Number Space. [En línea]. Disponible en: <https://tools.ietf.org/html/rfc4893> [Acceso 30 September 2020].

[68] DARPA, RFC 5396, Textual Representation of Autonomous System (AS) Numbers, Dec 2008.

[69] Ibm.com. (2019). IBM Knowledge Center. [En línea]. Disponible en: https://www.ibm.com/support/knowledgecenter/es/ssw_aix_71/com.ibm.aix.networkcomm/protocols_autonomous.htm [Acceso 9 Feb. 2019].

[70] J. Moy, R. Coltun and D. Ferguson, OSPF for IPv6, RFC 5340, July 2008

[71] J. Moy, OSPF Version 2, RFC 2328, April 1998.

[72] McQuillan, J., et.al., The New Routing Algorithm for the ARPANET, IEEE Transactions on Communications, May 1980.

[73] Digital Equipment Corporation, Information processing systems -- Data communications -- Intermediate System to Intermediate System Intra-Domain Routing Protocol, October 1987.

[74] Proposed Standard, The OSPF Specification, RFC 1131, October 1989.

[75] Tecnologías, S., Routing, I. and Tecnología, W. (2020). Guía de diseño de OSPF. [En línea] Cisco. Disponible en: https://www.cisco.com/c/es_mx/support/docs/ip/open-shortest-path-first-ospf/7039-1.html#t6 [Acceso 20 enero. 2019].

[76] Murphy, P., The OSPF Not-So-Stubby Area (NSSA) Option, RFC 3101, January 2003.

[77] K. Lougheed, Y. Rekhter, *A Border Gateway Protocol 3 (BGP-3)*, RFC **1267**, October 1991

[78] P. Marques, F. Dupont, *Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing*, RFC **2545**, March 1999

[79] T. Bates, Y. Rekhter, R. Chandra, D. Katz, *Multiprotocol Extensions for BGP-4*, RFC **2858**, June 2000

[80] Y. Rekhter, T. Li, S. Hares, A Border Gateway Protocol 4 (BGP-4), RFC 4271, January 2006

[81] T. Bates, R. Chandra, D. Katz, Y. Rekhter, Multiprotocol Extensions for BGP-4, RFC 4760, January 2007

[82] Castillo Velázquez, J. I. El árbol de internet y la estructura de la información de gestión de una red. IEEE Latin America and the Caribbean Newsletter No. 62, 2009. (Consultado febrero 2018). http://ewh.ieee.org/sb/mexico/uacm/jicv/2009_arbolinternet_n62_pp1517.pdf

[83] Davin, J., J. Case, M. Fedor, and M. Schoffstall, *A Simple Gateway Monitoring Protocol*, RFC 1028, Proteon, University of Tennessee at Knoxville, Cornell University, and Rensselaer Polytechnic Institute, November 1987.

[84] M. Rose, K. McCloghrie, *Structure and Identification of Management Information for TCP/IP-based internets*, RFC 1065, August 1988

[85] M. Rose, K. McCloghrie, *Management Information Base for Network Management of TCP/IP-based internets*, RFC 1066, August 1988

[86] J. Case, M. Fedor, M. Schoffstall, J. Davin, *A Simple Network Management Protocol*, RFC 1067, August 1988

[87] M. Rose, K. McCloghrie, *Structure and Identification of Management Information for TCP/IP-based Internets*, RFC 1155, May 1990

[88] M. Rose, K. McCloghrie, Management Information Base for Network Management of TCP/IP-based internets, RFC 1156, May 1990

[89] J. Case, M. Fedor, M. Schoffstall, J. Davin, *A Simple Network Management Protocol (SNMP)*, RFC 1157, May 1990

[90] M. Rose, K. McCloghrie, *Concise MIB Definitions*, RFC 1212, March 1991

[91] M. Rose, K. McCloghrie, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, RFC 1213, March 1991

[92] M. Rose, *A Convention for Defining Traps for use with the SNMP*, RFC 1215, March 1991

[93] M. Rose, K. McCloghrie, J. Case, S. Waldbusser, *Introduction to version 2 of the Internet-standard Network Management Framework*, RFC 1441, April 1993

[94] M. Rose, K. McCloghrie, J. Case, S. Waldbusser, *Coexistence between version 1 and version 2 of the Internet-standard Network Management Framework*, RFC 1452, April 1993

[95] M. Rose, K. McCloghrie, J. Case, S. Waldbusser, *Introduction to Community-based SNMPv2*, RFC 1901, January 1996

[96] M. Rose, K. McCloghrie, J. Case, S. Waldbusser, *Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2)*, RFC 1902, January 1996

[97] M. Rose, K. McCloghrie, J. Case, S. Waldbusser, *Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)*, RFC 1905, January 1996

[98] M. Rose, K. McCloghrie, J. Case, S. Waldbusser, *Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework*, RFC 1908, January 1996

[99] K. McCloghrie, *An Administrative Infrastructure for SNMPv2*, RFC 1909, February 1996

[100] G. Waters, *User-based Security Model for SNMPv2*, RFC 1910, February 1996

[101] J. Case, R. Mundy, D. Partain, B. Stewart, *Introduction to Version 3 of the Internet-standard Network Management Framework*, RFC 2570, April 1999

[102] R. Frye, D. Levi, S. Routhier, B. Wijnen, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*, RFC 2576, March 2000

[103] K. McCloghrie, D. Perkins, J. Schoenwaelder, *Structure of Management Information Version 2 (SMIv2)*, RFC 2578, April 1999

[104] K. McCloghrie, D. Perkins, J. Schoenwaelder, *Textual Conventions for SMIv2*, RFC 2579, April 1999

[105] K. McCloghrie, D. Perkins, J. Schoenwaelder, *Conformance Statements for SMIv2*, RFC 2580, April 1999

[106] M. St. Johns, *Diffie-Helman USM Key Management Information Base and Textual Convention*, RFC 2786, March 2000

[107] J. Case, R. Mundy, D. Partain, B. Stewart, *Introduction and Applicability Statements for Internet Standard Management Framework*, RFC 3410, December 2002

[108] D. Harrington, R. Presuhn, B. Wijnen, *An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks*, RFC 3411, December 2002

[109] J. Case, D. Harrington, R. Presuhn, B. Wijnen, *Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)*, RFC 3412, December 2002

[110] D. Levi, P. Meyer, B. Stewart, *Simple Network Management Protocol (SNMP) Applications*, RFC 3413, December 2002

[111] U. Blumenthal, B. Wijnen, *User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)*, RFC 3414, December 2002

[112] B. Wijnen, R. Presuhn, K. McCloghrie, *View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)*, RFC 3415, December 2002

[113] R. Presuhn, *Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)*, RFC 3416, December 2002

[114] R. Presuhn, *Transport Mappings for the Simple Network Management Protocol (SNMP)*, RFC 3417, December 2002

[115] R. Presuhn, *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)*, RFC 3418, December 2002

[116] R. Frye, D. Levi, S. Routhier, B. Wijnen, *Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework*, RFC 3584, August 2003

[117] D. Harrington, J. Schoenwaelder, *Transport Subsystem for the Simple Network Management Protocol (SNMP)*, RFC 5590, June 2009

[118] He.net. (2020). Hurricane Electric Internet Services - Internet Backbone and Colocation Provider. [En línea]. Disponible en: https://www.he.net/ [Acceso 9 Feb. 2020].

[119] Bgpview.io. (2020). Home Page - BGPView. [En línea]. Disponible en: https://bgpview.io/ [Acceso 9 Feb. 2020].

[120] Peeringdb.com. (2020). PeeringDB. [En línea]. Disponible en: https://www.peeringdb.com/ [Acceso 9 Feb. 2020].

[121] Gns3.com. (2020). GNS3 | [En línea]. Disponible en: https://www.gns3.com/software [Acceso 9 Feb. 2020].

[122] Support, P., 2020. Routers - Routers Cisco De La Serie 7200. [En línea] Cisco. Disponible en: <https://www.cisco.com/c/es_mx/support/routers/7200-series-routers/series.html> [Acceso 24 August 2020].

Apéndice A

Características del Router 7200

Para este trabajo se utilizó *Dynamips* un emulador de IOS que permite a los usuarios ejecutar binarios de imágenes IOS de *Cisco Systems*. La imagen IOS que se utilizo fue de un router 7200 con imagen IOS 7200v15.2(4)M2.

Con interfaces de:

- Ethernet 10BASE-T and 10BASE-FL
- Fast Ethernet 100BASE-T (RJ-45 and Mil)
- Gigabit Ethernet
- Token Ring (half and full duplex)
- Synchronous serial ISDN BRI, PRI, HSSI, T3, E3
- Multichannel T1, ISDN PRI
- Multichannel E1, ISDN PRI
- Multichannel T3, E3
- Multichannel STM-1
- Packet Over SONET (POS)
- Dynamic Packet Transport (DPT)
- ATM (single-mode and multimode)
- ATM-CES
- Digital Voice Port Adapter, Enhanced
- Mix-enabled T1/E1
- Integrated Service Adapter (ISA)
- VPN Acceleration Module (VAM)

Apariencia física de un router Cisco 7200

Las series de Cisco 7200 son una familia de *routers* de acceso multiservicio que proporcionan:

- Integración multiservicio de voz y datos
- Acceso a Internet/intranet con firewall
- Acceso a redes privadas virtuales (VPN) con opciones de firewall
- Servicios de acceso telefónico analógico y digital
- Enrutamiento con gestión de ancho de banda
- Enrutamiento entre VLAN

Con un potente procesador de alto rendimiento y procesadores auxiliares en varias interfaces, la serie Cisco 7200 admite calidad de servicio (QoS) avanzada, seguridad y las características de integración en la red.

Apéndice B

Características de configuración de cada *router*

Las conexiones que se establecieron para este trabajo fueron con direccionamiento de tipo Unicast, con prefijo de 64. Contando con 152 direcciones IPv6 asignadas y 76 redes IPv6. A continuación se muestran las interfaces de cada *router* de esta tesis, los protocolos que se habilitaron y su número de AS asignado.

Seattle_(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Seattle_01(Pacific_Wave)	2000:0:0:4::1/64	OSPFv3			
g1/0	Seattle_02(Pacific_Wave)	2000:0:0:5::1/64	OSPFv3	✓	✓	
g2/0	Seattle_03(Pacific_Wave)	2000:0:0:6::1/64	OSPFv3	SNMP	PCs	AS
p3/0	Sunnyvale(Pacific_Wave)	2000:0:0:1::1/64	OSPFv3	✓		62819
p4/0	Sunnyvale(Pacific_Wave)	2000:0:0:2::1/64	OSPFv3			
p5/0	Sunnyvale(Pacific_Wave)	2000:0:0:3::1/64	OSPFv3			
p6/0	New_York_(MAN/LAN)	2001:504:b:11::1/64	OSPFv3			

Seattle_02(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Seattle_(Pacific_Wave)	2000:0:0:5::2/64	OSPFv3			
p1/0	Internet2	2001:0:0:1::1/64	BGP	✓	✓	✓
p2/0	DREN	2003:0:0:4::2/64	BGP	SNMP	PCs	AS
p3/0	Denver	2000:0:0:1C::1/64	OSPFv3	✓		62819
p4/0	Oahu	2005:0:0:3::1/64	BGP			
p5/0	Oahu	2004:0:0:4::1/64	BGP			
p6/0	New_Zealand	2004:0:0:5::1/64	BGP			

Sunnyvale(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Sunnyvale_01(Pacific_Wave)	2000:0:0:10::1/64	OSPFv3			
p1/0	Seattle_(Pacific_Wave)	2000:0:0:1::2/64	OSPFv3	✓	✓	
p2/0	Seattle_(Pacific_Wave)	2000:0:0:2::2/64	OSPFv3	SNMP	PCs	AS
p3/0	Seattle_(Pacific_Wave)	2000:0:0:3::2/64	OSPFv3	✓		62819
p4/0	Los_Angeles_(Pacific_Wave)	2000:0:0:8::1/64	OSPFv3			
p5/0	Los_Angeles_(Pacific_Wave)	2000:0:0:9::1/64	OSPFv3			
p6/0	Los_Angeles_(Pacific_Wave)	2000:0:0:A::1/64	OSPFv3			

Los_Angeles_(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Los_Angeles_01(Pacific_Wave)	2000:0:0:B::1/64	OSPFv3			
g1/0	Los_Angeles_02(Pacific_Wave)	2000:0:0:C::1/64	OSPFv3	✓	✓	
g2/0	Los_Angeles_03(Pacific_Wave)	2000:0:0:D::1/64	OSPFv3	SNMP	PCs	AS
p3/0	DC_Area_(Wix)	2005:0:0:6::1/64	OSPFv3	✓		62819
p4/0	Sunnyvale(Pacific_Wave)	2000:0:0:8::2/64	OSPFv3			
p5/0	Sunnyvale(Pacific_Wave)	2000:0:0:9::2/64	OSPFv3			
p6/0	Sunnyvale(Pacific_Wave)	2000:0:0:A::2/64	OSPFv3			

Los_Angeles_02(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Los_Angeles_(Pacific_Wave)	2000:0:0:C::2/64	OSPFv3			
p1/0	El_Paso	2000:0:0:E::1/64	OSPFv3	✓	✓	✓
p2/0	Tijuana	2000:0:0:F::1/64	OSPFv3	SNMP	PCs	AS
p3/0	China	2005:0:0:8::1/64	BGP	✓		62819
p4/0	Taiwan	2005:0:0:9::1/64	BGP			
p5/0	Guam	2005:0:0:D::1/64	BGP			
-	Down	Down	Down			

Tokio_(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
-	Down	Down	Down			
p1/0	Seattle_01(Pacific_Wave)	2005:0:0:1::2/64	OSPFv3	✓	✓	✓
p2/0	Seattle_01(Pacific_Wave)	2000:0:0:1B::2/64	OSPFv3	SNMP	PCs	AS
p3/0	Seattle_01(Pacific_Wave)	2005:0:0:2::2/64	OSPFv3	✓		62819
p4/0	Los_Angeles_01(Pacific_Wave)	2005:0:0:7::2/64	OSPFv3			
p5/0	WIDE	2000:0:0:11::1/64	BGP			
p6/0	Hong_Kong	2005:0:0:E::1/64	BGP			

DC_Area_(Wix)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
-	Down	Down	Down			
p1/0	Los_Angeles_(Pacific_Wave)	2005:0:0:6::2/64	OSPFv3	✓	✓	✓
p2/0	New_York_(MAN/LAN)	2000:0:0:12::2/64	OSPFv3	SNMP	PCs	AS
p3/0	Esnet	2002:0:0:5::1/64	BGP	✓		62819
p4/0	Internet2	2001:0:0:4::1/64	BGP			
-	Down	Down	Down			
-	Down	Down	Down			

Chicago

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Chicago_V	2006:0:0:3::1/64	OSPFv3			
p1/0	Seattle_01(Pacific_Wave)	2000:0:0:7::2/64	OSPFv3	✓	✓	
p2/0	Denver	2000:0:0:15::1/64	OSPFv3	SNMP	PCs	AS
p3/0	Kansas_City	2000:0:0:16::1/64	OSPFv3	✓	✓	62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Seattle_01(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Seattle_(Pacific_Wave)	2000:0:0:4::2/64	OSPFv3			
p1/0	Tokio_(Pacific_Wave)	2005:0:0:1::1/64	OSPFv3	✓	✓	✓
p2/0	Tokio_(Pacific_Wave)	2000:0:0:1B::1/64	OSPFv3	SNMP	PCs	AS
p3/0	Tokio_(Pacific_Wave)	2005:0:0:2::1/64	OSPFv3	✓		62819
p4/0	PNWGP	2003:0:0:9::2/64	BGP			
p5/0	Chicago	2000:0:0:7::1/64	OSPFv3			
p6/0	Esnet	2002:0:0:1::1/64	BGP			

Seattle_03(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Seattle_(Pacific_Wave)	2000:0:0:6::2/64	OSPFv3			
p1/0	Korea	2005:0:0:4::1/64	BGP	✓	✓	✓
p2/0	China	2005:0:0:5::1/64	BGP	SNMP	PCs	AS
-	Down	Down	Down	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Sunnyvale_01(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Sunnyvale(Pacific_Wave)	2000:0:0:10::2/64	OSPFv3			
p1/0	CENIC	2003:0:0:2::2/64	BGP	✓	✓	✓
p2/0	NREN_(NASA_Ames)	2003:0:0:8::2/64	BGP	SNMP	PCs	AS
p3/0	Esnet	2002:0:0:2::1/64	BGP	✓	✓	62819
g4/0	Sunnyvale_V	2006:0:0:1::1/64	OSPFv3			
-	Down	Down	Down			
-	Down	Down	Down			

Los_Angeles_01(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Los_Angeles_(Pacific_Wave)	2000:0:0:B::2/64	OSPFv3			
p1/0	Tokio_(Pacific_Wave)	2005:0:0:7::1/64	OSPFv3	✓	✓	✓
p2/0	Los_Nettos	2003:0:0:7::2/64	BGP	SNMP	PCs	AS
p3/0	UltraLight_Caltech	2003:0:0:B::2/64	BGP	✓		62819
p4/0	Internet2	2001:0:0:2::1/64	BGP			
p5/0	DREN	2003:0:0:5::2/64	BGP			
p6/0	Esnet	2002:0:0:3::1/64	BGP			

Los_Angeles_03(Pacific_Wave)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
g0/0	Los_Angeles_(Pacific_Wave)	2000:0:0:D::2/64	OSPFv3			
p1/0	Oahu	2005:0:0:A::1/64	BGP	✓	✓	✓
p2/0	Oahu	2005:0:0:B::1/64	BGP	SNMP	PCs	AS
p3/0	Hawaii	2004:0:0:6::1/64	BGP	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

New_York_(MAN/LAN)

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
-	Down	Down	Down			
p1/0	Seattle_(Pacific_Wave)	2001:504:b:11::2/64	OSPFv3	✓	✓	✓
p2/0	DC_Area_(Wix)	2000:0:0:12::1/64	OSPFv3	SNMP	PCs	AS
p3/0	Esnet	2002:0:0:4::1/64	BGP	✓		62819
p4/0	Internet2	2001:0:0:3::1/64	BGP			
-	Down	Down	Down			
-	Down	Down	Down			

Albuquerque

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
-	Down	Down	Down			
p1/0	ABQG	2003:0:0:1::2/64	BGP	✓	✓	✓
p2/0	Denver	2000:0:0:13::1/64	OSPFv3	SNMP	PCs	AS
p3/0	El_Paso	2000:0:0:14::1/64	OSPFv3	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Dallas

Interface	Connect	IP	Protocol	Características		
				IPv6	OSPFv3	BGP
-	Down	Down	Down			
p1/0	Houston	2000:0:0:17::1/64	OSPFv3	✓	✓	
p2/0	Tulsa	2000:0:0:18::1/64	OSPFv3	SNMP	PCs	AS
-	Down	Down	Down	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Denver

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_02(Pacific_Wave)	2000:0:0:1C::2/64	OSPFv3	✓	✓	✓
p2/0	Albuquerque	2000:0:0:13::2/64	OSPFv3	SNMP	PCs	AS
p3/0	Chicago	2000:0:0:15::2/64	OSPFv3	✓		62819
p4/0	FRGP	2003:0:0:6::2/64	BGP			
-	Down	Down	Down			
-	Down	Down	Down			

El_Paso

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Los_Angeles_02(Pacific_Wave)	2000:0:0:E::2/64	OSPFv3	✓	✓	✓
p2/0	Albuquerque	2000:0:0:14::2/64	OSPFv3	SNMP	PCs	AS
p3/0	Houston	2000:0:0:19::1/64	OSPFv3	✓		62819
p4/0	Transtelco	2003:0:0:A::2/64	BGP			
-	Down	Down	Down			
-	Down	Down	Down			

Tijuana

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Los_Angeles_02(Pacific_Wave)	2000:0:0:F::2/64	OSPFv3	✓	✓	✓
p2/0	CUDI	2003:0:0:3::2/64	BGP	SNMP	PCs	AS
-	Down	Down	Down	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Tulsa

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Dallas	2000:0:0:18::2/64	OSPFv3	✓	✓	
p2/0	Kansas_City	2000:0:0:1A::2/64	OSPFv3	SNMP	PCs	AS
-	Down	Down	Down	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Oahu

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_02(Pacific_Wave)	2004:0:0:4::2/64	BGP	✓	✓	✓
p2/0	Seattle_02(Pacific_Wave)	2005:0:0:3::2/64	BGP	SNMP	PCs	AS
p3/0	Los_Angeles_03(Pacific_Wave)	2005:0:0:A::2/64	BGP	✓		7575
p4/0	Los_Angeles_03(Pacific_Wave)	2005:0:0:B::2/64	BGP			
p5/0	Australia	2004:0:0:1::2/64	OSPFv3			
p6/0	Guam	2005:0:0:C::1/64	BGP			

Guam

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Los_Angeles_02(Pacific_Wave)	2005:0:0:D::2/64	BGP	✓	✓	✓
p2/0	Oahu	2005:0:0:C::2/64	BGP	SNMP	PCs	AS
p3/0	Hong_Kong	2005:0:0:F::1/64	BGP	✓		395889
p4/0	Hong_Kong	2005:0:0:20::1/64	BGP			
p5/0	Singapore	2005:0:0:21::1/64	OSPFv3			
-	Down	Down	Down			

Hawaii

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Los_Angeles_03(Pacific_Wave)	2004:0:0:6::2/64	BGP	✓	✓	✓
p2/0	New_Zealand	2004:0:0:3::2/64	OSPFv3	SNMP	PCs	AS
-	Down	Down	Down	✓		7575
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

China

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_03(Pacific_Wave)	2005:0:0:5::2/64	BGP	✓		✓
p2/0	Los_Angeles_02(Pacific_Wave)	2005:0:0:8::2/64	BGP	SNMP	PCs	AS
-	Down	Down	Down	✓		395889
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Korea

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_03(Pacific_Wave)	2005:0:0:4::2/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		395889
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Singapore

Interface	Connect	IP	Protocol	Características		
g0/0	Singapore_V	2006:0:0:2::1/64	OSPFv3	IPv6	OSPFv3	BGP
p1/0	Guam	2005:0:0:21::2/64	OSPFv3	✓	✓	✓
p2/0	Hong_Kong	2005:0:0:22::1/64	BGP	SNMP	PCs	AS
-	Down	Down	Down	✓	✓	395889
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Taiwan

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Los_Angeles_02(Pacific_Wave)	2005:0:0:9::2/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		7539
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Hong_Kong

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Tokio_(Pacific_Wave)	2005:0:0:E::2/64	BGP	✓		✓
p2/0	Guam	2005:0:0:F::2/64	BGP	SNMP	PCs	AS
p3/0	Guam	2005:0:0:20::2/64	BGP	✓		22388
p4/0	Singapore	2005:0:0:22::2/64	BGP			
-	Down	Down	Down			
-	Down	Down	Down			

WIDE

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Tokio_(Pacific_Wave)	2000:0:0:11::2/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		395889
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Australia

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Oahu	2004:0:0:1::1/64	OSPFv3	✓	✓	
p2/0	New_Zealand	2004:0:0:2::1/64	OSPFv3	SNMP	PCs	AS
-	Down	Down	Down	✓		7575
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

New_Zealand

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_02(Pacific_Wave)	2004:0:0:5::2/64	BGP	✓	✓	✓
p2/0	Hawaii	2004:0:0:3::1/64	OSPFv3	SNMP	PCs	AS
p3/0	Australia	2004:0:0:2::2/64	OSPFv3	✓		7575
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

ABQG

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Albuquerque	2003:0:0:1::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		40498
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

CENIC

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Sunnyvale_01(Pacific_Wave)	2003:0:0:2::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		2153
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

CUDI

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Tijuana	2003:0:0:3::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		18592
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Houston

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Dallas	2000:0:0:17::2/64	OSPFv3	✓	✓	
p2/0	El_Paso	2000:0:0:19::2/64	OSPFv3	SNMP	PCs	AS
-	Down	Down	Down	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Kansas_City

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Chicago	2000:0:0:16::2/64	OSPFv3	✓	✓	
p2/0	Tulsa	2000:0:0:1A::1/64	OSPFv3	SNMP	PCs	AS
-	Down	Down	Down	✓		62819
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

DREN

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_02(Pacific_Wave)	2003:0:0:4::1/64	BGP	✓		✓
p2/0	Los_Angeles_01(Pacific_Wave)	2003:0:0:5::1/64	BGP	SNMP	PCs	AS
-	Down	Down	Down	✓		668
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Esnet

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_01(Pacific_Wave)	2002:0:0:1::2/64	BGP	✓		✓
p2/0	Sunnyvale_01(Pacific_Wave)	2002:0:0:2::2/64	BGP	SNMP	PCs	AS
p3/0	Los_Angeles_01(Pacific_Wave)	2002:0:0:3::2/64	BGP	✓		293
p4/0	New_York_(MAN/LAN)	2002:0:0:4::2/64	BGP			
p5/0	DC_Area_(Wix)	2002:0:0:5::2/64	BGP			
p6/0	Down	Down	Down			

FRGP

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Denver	2003:0:0:6::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		14041
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Internet2

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_02(Pacific_Wave)	2001:0:0:1::2/64	BGP	✓		✓
p2/0	Los_Angeles_01(Pacific_Wave)	2001:0:0:2::2/64	BGP	SNMP	PCs	AS
p3/0	New_York_(MAN/LAN)	2001:0:0:3::2/64	BGP	✓		11164
p4/0	DC_Area_(Wix)	2001:0:0:4::2/64	BGP			
-	Down	Down	Down			
-	Down	Down	Down			

Los_Nettos

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Los_Angeles_01(Pacific_Wave)	2003:0:0:7::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		226
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

NREN_(NASA_Ames)

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Sunnyvale_01(Pacific_Wave)	2003:0:0:8::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		21556
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

PNWGP

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Seattle_01(Pacific_Wave)	2003:0:0:9::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		101
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Transtelco

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	El_Paso	2003:0:0:A::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		32098
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

UltraLight_Caltech

Interface	Connect	IP	Protocol	Características		
-	Down	Down	Down	IPv6	OSPFv3	BGP
p1/0	Los_Angeles_01(Pacific_Wave)	2003:0:0:B::1/64	BGP	✓		✓
-	Down	Down	Down	SNMP	PCs	AS
-	Down	Down	Down	✓		32361
-	Down	Down	Down			
-	Down	Down	Down			
-	Down	Down	Down			

Abstrac

The Internet was born derived from the need to use decentralized information in a computer network, focused on the military, governments and research and academic institutions. This first network was known as ARPANET, which started with 4 computers, which connected the Stanford Research Institute (SRI), the University of Utah, the University of California at Los Angeles (UCLA) and the University of California in Santa Barbara (UCSB).

In its beginnings, the Internet did not take into account the magnitude and impact it would have on a global level, so the IPv4 protocol in charge of giving an identifier to each device to navigate the network began to run out given the expansion of users and their demands to be within the network for as long as possible. That is why the need for a new protocol capable of satisfying these demands was seen. The IPv6 protocol is the answer to these needs, it allows an expansion of the use of the Internet and improvement of the functionalities of its predecessor. The evolution from IPv4 to IPv6 brought new features for routing protocols such as OSPFv3 which is a new version for IPv6 and BGP-4 that only generated an extension to be able to use IPv6.

In 1995 the commercial Internet is released as an operational network for the world, before that it was ARPANET and its evolution NSFNET. With a commercial Internet, a dedicated space was required to carry out research and continue innovating, which is why advanced networks emerge, networks that are in an area different from the commercial Internet that allow scientists, researchers, academics, teachers and students to collaborate , sharing information and tools through a series of network interconnections, examples of these networks are, GÉANT (Europe), Internet2 (United States), CANARIE (Canada), TEIN*CC (Asia), WACREN (West and Central Africa) , UbuntuNet Alliance (East and South Africa), ASREN (Arab States), among others.

The Pacific Wave Advanced Network is a distributed network that serves as an Internet exchange point, focused on research and education. Providing high-performance Internet connectivity between US science and engineering research and development institutions and their international partners, and is a critical infrastructure for accessing internationally supported instruments, sources, and repositories of data to big scale.

PacWave enables large-scale scientific workflows to accelerate discovery in all areas of science and engineering, including high-energy physics, earth science, astronomy and astrophysics, biology and biomedical engineering, as well as scalable visualization, virtual reality, machine learning and artificial intelligence.

UACM
Universidad Autónoma
de la Ciudad de México
Nada humano me es ajeno

Autonomous University Of Mexico City

Science and Technology College

"Emulation for the Operation of the Pacific Wave Advanced Network"

Thesis

To obtain the grade of

Bachelor in Engineering in Electronic Systems and Telecommunications

Presents:

Iván Varela-Sánchez

Thesis Director:

M. en C. José Ignacio Castillo Velázquez

Thesis Co-Director:

M. en C. Joel Yazbek Buendía Gómez

Mexico City, June 2021.