UACM
Universidad Autónoma de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

“Emulación de la gestión de la red europea GEANT bajo protocolos IPv6”

TESIS

PARA OBTENER EL TITULO DE

LICENCIADA (O) EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

P R E S E N T A

ISABEL MUÑOZ MARTÍNEZ Y JORGE ARMANDO DÍAZ RAMÍREZ

D I R E C T O R

M. en C. José Ignacio Castillo Velázquez

Ciudad de México, agosto 2020

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

Isabel Muñoz Martínez

Agradezco a Dios por todo cuánto me has dado, porque tú has estado a mi lado aun en los momentos más difíciles, tú nunca te has olvidado de mí.

A mi más grande amor mi madre, que siempre me impulso en seguir adelante me enseñó a nunca detenerme y no tener miedo de cumplir mis sueños, a ella que me inculcó cada día fuerza, ella que hubiera querido estar en presencia en mi titulación, pero sé que estará en espíritu.

Agradezco a mi padre por todo lo que me ha inculcado, gracias por todo su esfuerzo que hace día a día por sacarnos adelante.

A mis hermanos que amo con todo mi corazón, a ti mi Aní que siempre me has alentado, tu que has sido mi inspiración de superación desde el momento en el que llegaste, gracias por tu apoyo y tus palabras de aliento. A ti Daniel tú has sido mi ejemplo de vida, siempre has estado conmigo apoyándome en todo momento, nunca me has dejado sola, te agradezco todo lo que has hecho por mí y si Dios lo permite siempre juntos seguiremos luchando los tres.

Gracias a mi asesor de tesis M. en C. José Ignacio Castillo Velásquez por su dedicación, tiempo, respaldo e interés; por enseñarme que la investigación y estudio son uno de los pilares de la vida. Le agradezco por todo su apoyo y sus palabras de aliento en los momentos más difíciles que he pasado, gracias porque me permitió conocer otra parte de usted, que no solo es un excelente profesor, sino una gran persona de grandes virtudes.

A mis lectores de tesis, en especial a la M. en C. Magali Cortez Vásquez por tomarse el tiempo y dar sus valiosos consejos para mejorar la presente tesis; también agradezco por sus enseñanzas dentro del aula y su conocimiento compartido.

A Jorge Armando Díaz Ramírez por ser mi persona incondicional, apoyándome en todo momento y en toda decisión, por ser mi pareja y mejor amigo, por tus

consejos y por no dejarme en los momentos difíciles, por estar siempre presente. Gracias, porque tú ves más de lo que hay en mí, que yo misma.

A Fernando Octavo Salinas y Ana Lilia Hernández Abonza por su gran y sincera amistad, por todos los momentos que hemos pasado juntos y por todo su apoyo brindado.

Finalmente quiero agradecer a la Universidad Autónoma de la Ciudad de México por permitirme formarme en ella y por la ayuda económica brindada para empastar está presente tesis.

Muchas mujeres hicieron el bien;
Más tú sobrepasas a todas.
Proverbios 31:29

No importa dónde estés,

no importa que no te pueda ver ni tocar,

siempre estarás conmigo.

Siempre serás mi más grande amor.

En memoria de mi madre Gaudencia Martínez García.

AGRADECIMIENTOS

Jorge Armando Díaz Ramírez

Gracias a Dios por darnos salud a toda mi familia y darnos la fortaleza necesaria para salir adelante en esta dura y difícil vida. Gracias a mis padres por todo el esfuerzo realizado para que yo haya llegado hasta aquí, por todos los consejos y valores que me han inculcado, así como le agradezco a mí hermano por darme su apoyo y guiarme por el camino correcto de la vida.

Agradezco profundamente a nuestro director de tesis al profesor José Ignacio Castillo Velázquez por amar su profesión y hacer que la amemos, así como por toda la paciencia, esfuerzo y tiempo que nos dedicó para realizar la presente tesis.

Les agradezco a los lectores por su tiempo y dedicación, en especial a la profesora Magali Cortez por sus consejos y observaciones, gracias por ser una excelente maestra.

Un agradecimiento especial para mi compañera, amiga y pareja Isabel Muñoz. Mi compañera de una y mil batallas, muchas gracias por esos consejos, ese liderazgo, esa pasión no solo en esta tesis, sino en la vida diaria. Eres una inspiración y motivación.

Gracias a Fernando Octavo y Ana Lilia Hernández, compañeros y amigos de la universidad. Creó que se formó una bonita amistad de toda la vida, gracias chicos por apoyarnos y siempre estar con nosotros.

Un agradecimiento especial a la Sra. Gaudencia Martínez García, madre de Isabel Muñoz, quien desgraciadamente se nos adelantó en el camino y no pudo estar con nosotros en este importante logro, gracias por todo señora, la llevare siempre en el corazón y cuídenos desde el cielo.

Y por último, pero no menos importante, gracias a la Universidad Autónoma de la Ciudad de México por haberme permitido formarme en ella, por permitirme lograr una de mis metas y conocer excelentes personas.

Además le agradezco a la UACM por otorgarnos el apoyo económico para poder realizar el empastado de nuestra presente tesis.

RESUMEN

La red avanzada europea GEANT está dedicada a la investigación y la enseñanza, actualmente conecta 43 países europeos con velocidades de conexión superiores a los 100 Gbps. GEANT es del interés de ADVNETLAB (*Advanced Networking Laboratory*) de la UACM, tal que este documento tiene por objetivo principal describir el funcionamiento de la red GEANT con base en la topología más actualizada (2018). Para esto se buscan 3 objetivos particulares.

1. Emular la conectividad de la red GEANT.
2. Emular la gestión de la red GEANT.
3. Determinar en qué medida el emulador GNS3, permite acercarse al desempeño de la red GEANT.

En ADVNETLAB se emplean emuladores para probar el funcionamiento de redes avanzadas, ya que los equipos como *routers* y *switches*, tienen costos del orden de millones de dólares. Para realizar la emulación se utilizó GNS3, tal que se conectaron 45 *routers* C7200, configurando los protocolos de enrutamiento OSPFV3 y gestión SNMPV3, ambos usan IPV6. Dado que GNS3 tiene como limitante el número de POS (*Packet Over-Sonet*) a 6 interfaces por *router* se usaron interfaces *GigabitEthernet y FastEthernet* para aquellos casos en los que se requirieron interfaces adicionales tal como Alemania_2, Reino Unido, Austria y Hungría. Se realizaron pruebas de conectividad y gestión, para corroborar el correcto funcionamiento de los protocolos, para lo cual se utilizaron varias herramientas: *Wireshark*, *PowerSNMP Free Manager, VMBox.* La emulación se realizó en una computadora con procesador Intel Core i5 y 12 GB en RAM. Toda la emulación se ejecutó satisfactoriamente pero se llevó el equipo al límite usando el 100% del CPU, el 74% de la RAM y tardo 25 minutos. Se encontró que GSN3 presento dos limitantes el ancho de banda a 1Gbps y un único tipo de *router* de backbone. Además *PowerSNMP Free Manager* presento algunas limitaciones al usar SNMPV3.

ABSTRACT

The European advanced network GEANT is dedicated to research and teaching, currently connecting 43 European countries with connection speeds of over 100 Gbps. GEANT is of interest to the ADVNETLAB (Advanced Networking Laboratory) of the UACM, such that this document's main objective is to describe the operation of the GEANT network based on the most updated topology (2018). For this, 3 objectives are sought.

1. Emulate the connectivity of the GEANT network.

2. Emulate the management of the GEANT network.

3. Determine to what extent the GNS3 emulator allows you to get closer to the performance of the GEANT network.

In ADVNETLAB emulators are used to test the operation of advanced networks, since equipment such as routers and switches have costs in the order of millions of dollars. GNS3 was used to perform the emulation, such that 45 C7200 routers were connected, configuring the routing protocols OSPFV3 and SNMPV3 management, both using IPV6. Given that GNS3 is limited by the number of POS (Packet Over-Sonet) to 6 interfaces per router, GigabitEthernet and FastEthernet interfaces were used for those cases where additional interfaces were required such as Germany_2, United Kingdom, Austria, and Hungary.

Connectivity and management tests were carried out to corroborate the correct operation of the protocols, for which various tools were used: Wireshark, PowerSNMP Free Manager, VMBox. The emulation was performed on a computer with an Intel Core i5 processor and 12 GB of RAM. All the emulation was successful, but the team was pushed to the limit using 100% CPU, 74% RAM and took 25 minutes. GSN3 was found to have two bandwidth limitations at 1Gbps and a single type of backbone router. In addition, PowerSNMP Free Manager presented some limitations when using SNMPV3.

Contexto del proyecto ADVNETLAB y este trabajo de tesis

El ADVNETLAB (*Advanced Networking Laboratory*) en la UACM fue fundado en 2013, una vez que hubo interés por parte de los estudiantes de ISET acerca del tema de las redes avanzadas. Desde entonces se ha desarrollado una metodología ADVNETLAB con la que se dirigen las tesis y otros proyectos, también se ha desarrollado 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. A la fecha se han titulado 11 estudiantes de licenciatura mexicanos y peruanos bajo la metodología ADVNETLAB. Además de las 11 tesis, hemos producido junto con 10 egresados en distintos momentos, 14 publicaciones en revistas y proceedings IEEE indexadas en SCOPUS, aportando un total de 23 publicaciones de investigación para la UACM.

En esta ocasión se presentan para 2020 Isabel Muñoz (ANL11) y Jorge Armando Díaz (ANL12) con el trabajo correspondiente al estudio vía emulación del *backbone* de la red avanzada europea GEANT 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 abordó GEANT para IPv4 y para topologías previas. Isabel y Jorge Armando tuvieron por fecha de examen profesional el 24 de abril de 2020, pospuesto por la pandemia de COVID-19, ahora son de los primeros de ISET en presentar examen profesional a distancia. Les expreso mis felicitaciones por su trabajo concluido.

M. en C. José Ignacio Castillo Velázquez

Director de tesis - 7 de Julio de 2020

ÍNDICE

Capítulo 1. Introducción a la Red GEANT

1.1 Historia de las redes europeas y la incursión de ARPANET

Los inicios de las redes europeas comenzaron con la conexión del Colegio Universitario de Londres y el *Royal Radar Establishment* de Noruega con la red ARPANET (*Advanced Research Projects Agency Network*, Red de la Agencia de Proyectos de Investigación Avanzada) de EE.UU. en 1973.

ARPANET, fue la primera red que comenzó con la conexión de equipos informáticos en EE.UU.

A finales de 1970, ISO (*International Organization for Standardization*, Organización Internacional de Normalización) creó OSI (*Open Systems Interconnection*, Modelo de Interconexión de Sistemas Abierto), este modelo ayudó a eliminar la interoperabilidad en equipos informáticos de diferentes marcas. Aun así, OSI no fue de gran relevancia en Europa, sólo el protocoló X.25 fue utilizado en gran medida por redes públicas.

El rápido crecimiento de las redes europeas, hizo que otros centros de investigación y redes académicas se interesaran por lograr alguna conexión transatlántica con centros de investigación de EE.UU [1].

1.1.2 Protocolo TCP/IP

El protocoló TCP (*Transmission Control Protocol,* Protocolo de control de transmisión), fue creado en 1974 por Vint Cerf y E. Kahn. Cabe recalcar, que E. Kanh trabajó para DARPA (*Defense Advanced Research Projects Agency*, Agencia de Investigación de Proyectos Avanzados de Defensa) y Vint Cerf para ARPANET. Esto no fue un impedimento, porque ambos vieron la necesidad de crear una arquitectura abierta de interconexión.

TCP/IP fue liberado en 1980 y en 1983 TCP/IP se propuso como un estándar para las redes de datos [2].

1.1.3 Protocolo X.25

X.25. fue propuesto por la ITU-TS (*Telecommunications Section of the International Telecommunications Union,* Sector de Normalización de

Telecomunicaciones de la UIT), basado en el modelo OSI. Este protocolo trabajó en las tres primeras capas del modelo OSI que son: capa física, capa de enlace de datos y capa de red.

El modelo OSI adoptó el modelo X.25, situación que aprovecharon las redes europeas para tener sus propios protocolos de comunicación y competir con las redes de EE.UU [3,4].

1.1.4 IBM en las primeras redes europeas

Tras el éxito de la red *BITNet*, la cual fue una red internacional de computadoras que ofreció servicios interactivos de correo electrónico y de transferencia de ficheros en EE.UU. IBM propuso la creación de una red similar, basada en *BITNet* con los protocolos de comunicación propios de IBM, a los principales centros académicos de computación europeos. IBM se propuso financiar las conexiones entre centros europeos de 1983 a 1985. Herb Budd, del lado de IBM, fue el iniciador y supervisor de este proyecto.

Una reunión de la fundación se llevó a cabo en Ginebra, fue durante la cual el inglés David Lord, responsable en ese momento de la red CERN, fue elegido Presidente del Comité de Dirección de la nueva Organización: *EARN (European Academic and Research Network,* Red Académica y de Investigación Europea).

Con EARN, se logró conectar a poco más de mil computadoras en su mayoría mainframes con sistemas operativos VM/CMS (*Virtual Machine / Conversacional Monitor System*) ubicadas en distintas universidades e instituciones de investigación en distintos países europeos [4].

1.1.5 Estandarización del protocolo TCP/IP en Europa y el inicio de la red NORDUNET

En los años 90 existieron varias redes como OSI, UUCP, DECNET y BITNET, las cuales tuvieron dificultades para lograr una interconexión entre ellas, por los diferentes protocolos que utilizaron.

Debido a esto y con el auge del protocolo TCP/IP en Europa, varias redes de Europa optaron por utilizar este protocolo y con ello se logró la conexión entre diferentes redes europeas [1].

La infraestructura de NORDUNET soportó protocolos como X.25, DECNET, TCP/IP, utilizó tecnologías Ethernet y *Cisco Multiprotocol Router*.

La arquitectura de red NORDUNET fue diseñada para soportar varios protocolos de red. Esto ayudó a proporcionar conectividad a las redes de investigación más importantes de Europa y Estados Unidos. En la figura 1 se muestra la solución multiprotocolo presentada por Harri Salminen [3].

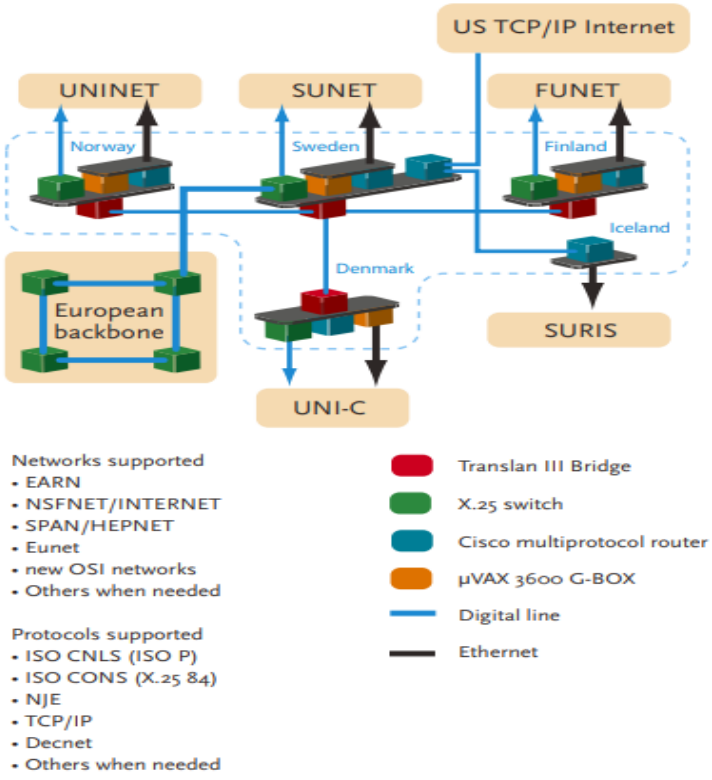


Figura 1. Topologías de las redes nórdicas en 1988 [3].

Pronto más redes europeas fueron incluyendo la tecnología TCP/IP a su infraestructura y empezaron a sustituir a X.25 con Bridges Ethernet para soportar TCP/IP.

1.1.6 Inicio de las NREN europeas

El ámbito de proyectos para la estandarización de redes europeas favoreció para que a principio de la décadas de los noventa se diera inicio formal a las redes de investigación y educación europeas: NREN *(National Research and Education Network,* Red Nacional de Investigación y Educacion*)* [4,5].

RARE (Réseaux Associés pour la Recherche Européenne, Asociación de Redes Académicas y de Investigación), tuvo como objetivo llevar a cabo la coordinación entre las NREN en Europa, así como, la creación *y* coordinación del Proyecto COSINE (*Cooperation for OSI Networking in Europe,* Cooperación para la Creación de Redes OSI en Europa).

El proyecto COSINE creó una red telemática, la cual se basó en las normas OSI y fue usada principalmente por centros de educación e investigación en Europa, logrando conectividad de 64 Kbps entre las redes europeas [1].

1.1.7 Conectividad de las redes europeas

COSINE fue uno de los primeros proyectos en usar *Backbone* (red dorsal o troncal), COSINE creó la red X.25 IXI, la cual usaba los protocolos OSI, a través de esta red interconectó redes académicas de Alemania, Austria, Bélgica, Dinamarca, España, Francia, Gran Bretaña, Grecia, Holanda, Irlanda, Italia, Luxemburgo, Portugal, Suecia, Suiza y ex-Yugoslavia. En la figura 2 se muestra la topología de la red X.25 IXI, la cual funcionó a velocidades de 64 Kbps [1,6].

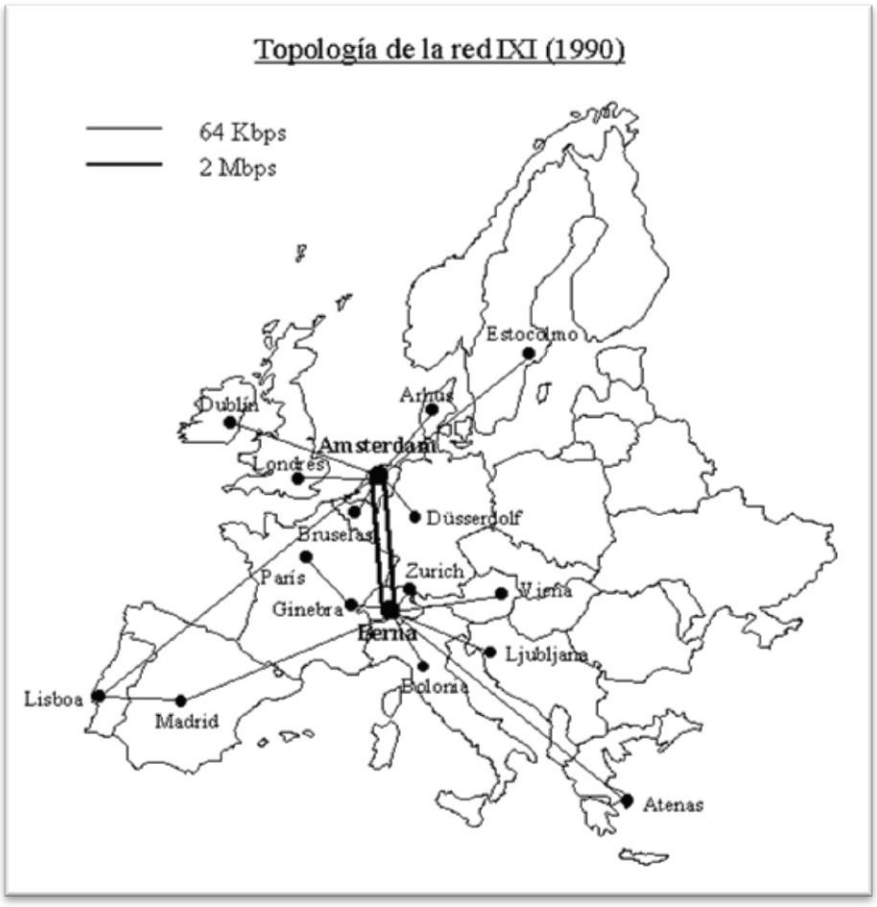


Figura 2. Topología de la red X.25 IXI en 1990 [1].

La red X.25 IXI fue una red académica y con el paso del tiempo surgió la necesidad de crear una red internacional. En 1992 se creó la red EBONE con la topología de cinco nodos o EBS (*Ebone Backbone System,* Sistema Backbone Ebone), con esta red se logró realizar una conexión hacia EE.UU. La topología de la red EBONE se muestra en la figura 3 [6].

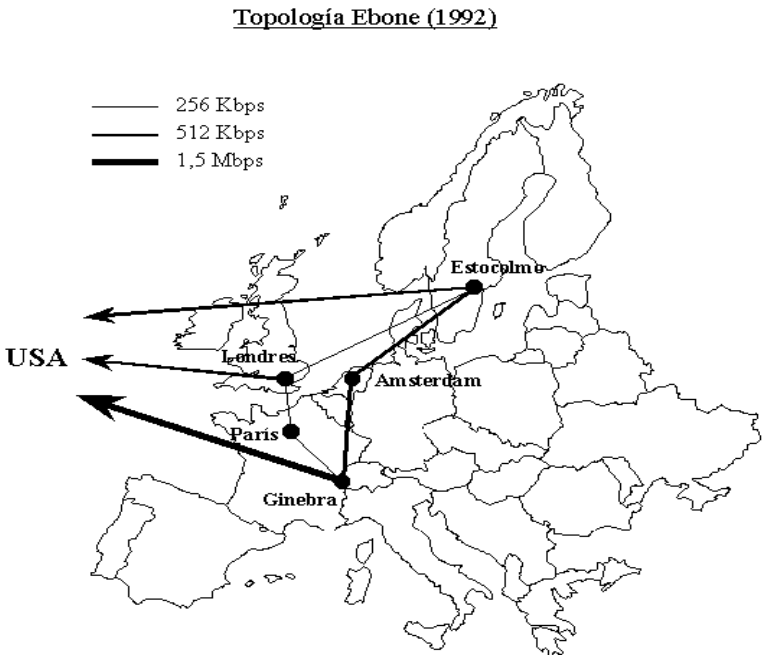


Figura 3. Topología de la red Ebone en 1992 [1].

En 1998 la red EBONE aumentó a trece EBS, como se muestra en la figura 4.

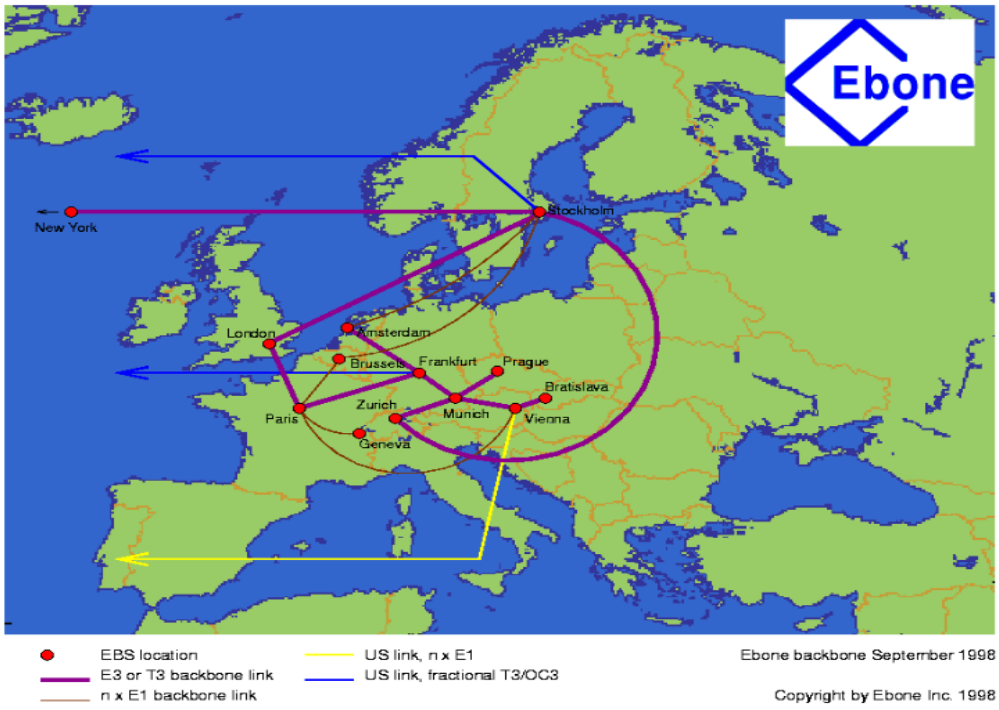


Figura 4. Topología de red Ebone en 1998 [1].

1.1.8. EUROPANET Y DANTE

Debido a la necesidad de aumentar la velocidad de transmisión a 2 Mbps, se decidió reemplazar a la red X.25 IXI, aparte de que la nueva red troncal utilizara múltiples protocolos. En 1992 se creó EuropaNET que ofreció un servicio de 2 Mbps.

La capacidad transatlántica propia de DANTE (*Delivery of Avanced Network Technology to Europe*, Entrega de Tecnología de Red Avanzada a Europa) se agregó a EuropaNET, la cual oficialmente empezó a dar servicio a principios de 1994. También realizó conexiones transatlánticas hacia EE.UU, desde Holanda y el CERN en Suiza como se muestra en la figura 5 [7,8].

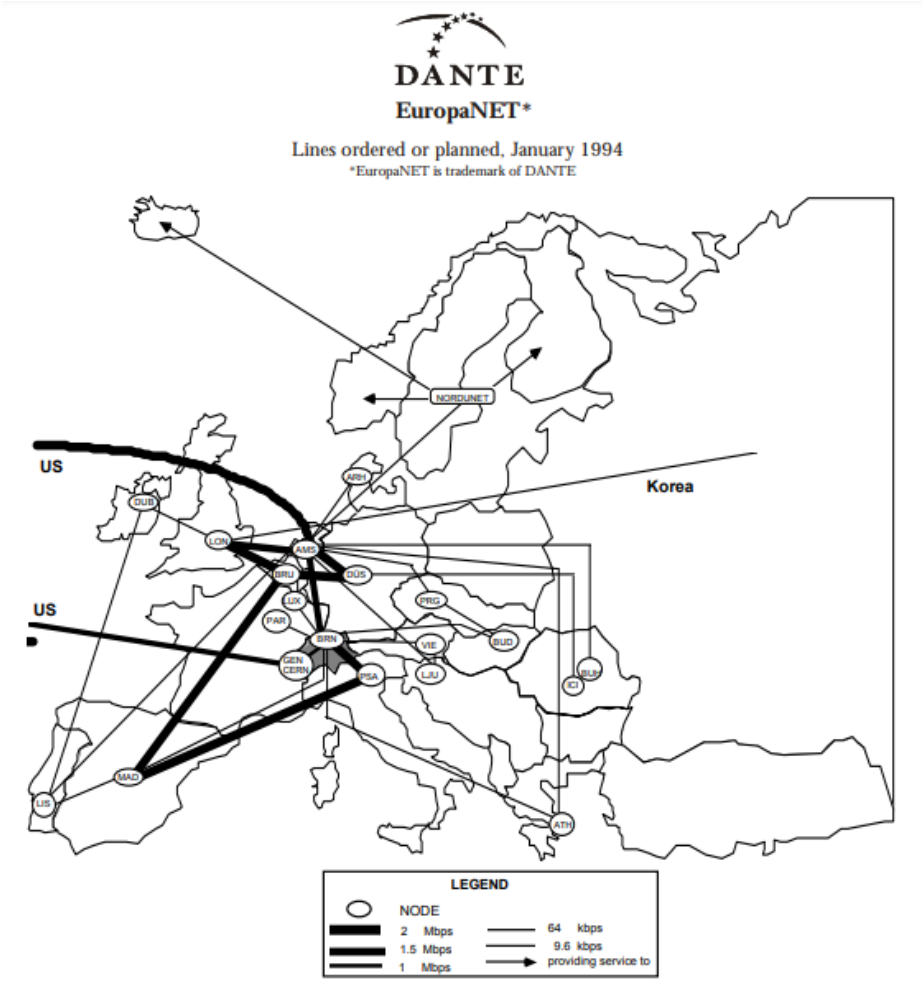


Figura 5. Topología de la red DANTE en 1994 [7].

1.1.9 DANTE

DANTE (*Delivery of Avanced Network Technology to Europe*, Entrega de Tecnología de Red Avanzada a Europa), se creó en 1993 operando en Cambridge, Reino Unido. DANTE asumió la principal red troncal internacional para la comunidad investigadora europea. En 1995 conectó a 10 países y se creó EuropaNet2, la cual se consideró la primera red proporcionada por DANTE [8,9].

1.1.10 EUROCAIRN

En el año de 1993 con la organización de 18 países europeos, se creó un proyecto llamado EUROCAIN (*European Cooperation for Academic and Industrial Research Networking,* Cooperación Europea para la Creación de Redes de Investigación Académica e Industrial*)*, el cual junto con DANTE formaron un plan estratégico para la creación de una red de alta velocidad de 34 Mbps para las NREN de Europa [10,11].

1.1.11 El inicio de la red Internet

En 1990 Tim Berners Lee y el ingeniero Robert Cailliau de Bélgica publicaron una propuesta llamada WWW (*World Wide Web*), cuyo primer prototipo se creó en el CERN (*European Organization for Nuclear Research,* Organización Europea para la Investigación Nuclear). Pero fue hasta 1993, que fue liberada la licencia de WWW para uso de todas las redes del mundo, creando una red global, distribuida, descentralizada y unificada por el protocolo de Internet y la WWW [10, 11].

1.2 REDES AVANZADAS EUROPEAS

1.2.1 El inicio de las redes de alta velocidad en Europa: TEN-34 (*Trans-European Network – 34 Mbps*)

El proyecto TEN-34 tuvo el apoyo de la Comisión Europea. Su objetivo principal fue proporcionar una velocidad de transmisión de 34 Mbps. Ésta red fue creada

para conectar todas las redes universitarias nacionales, con DANTE como socio coordinador del proyecto [12].

DANTE logró conectar a 15 países a velocidades entre 6 y 34 Mbps, además proporcionó un servicio confiable y de alta calidad, ofreció una conexión de 45 Mbps a EE.UU. La figura 6 muestra la topología de la red TEN-34 en 1997 [13].

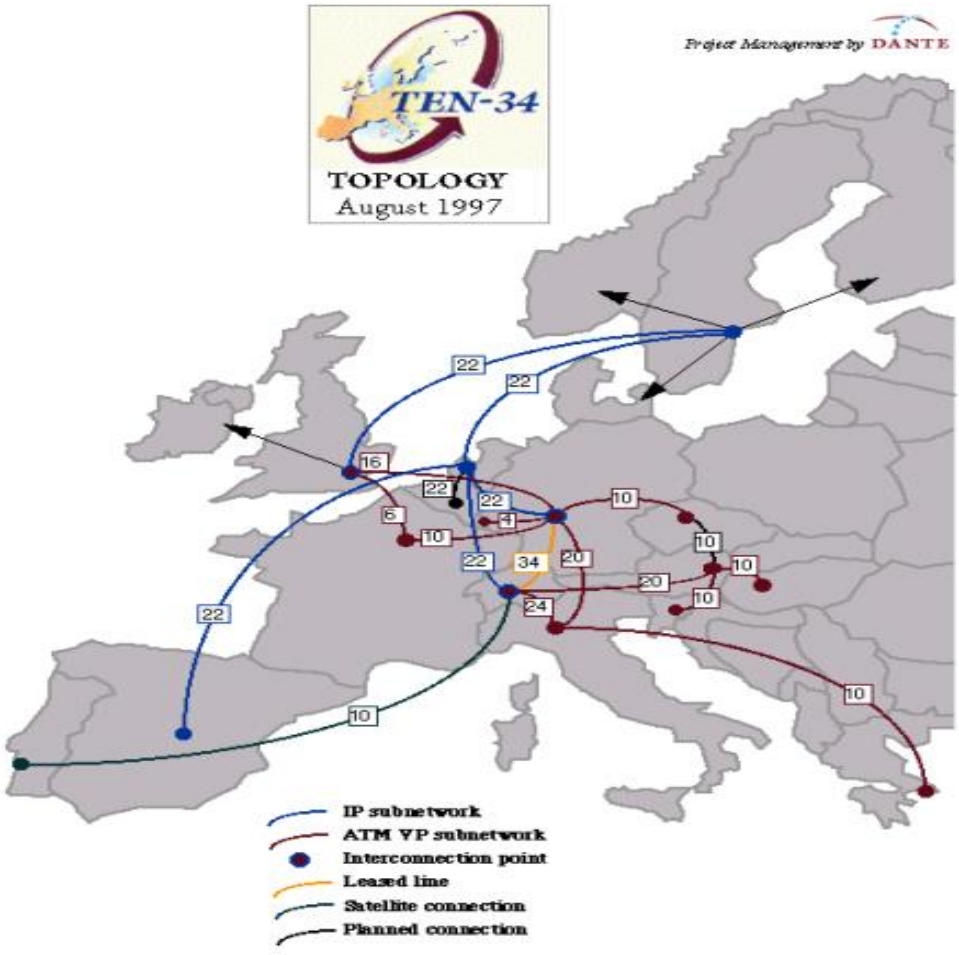


Figura 6. Topología de la red TEN-34 en 1997 [12].

1.2.2 Proyecto *QUANTUM*

DANTE preparó la propuesta de esquema, denominada QUANTUM (*QUAlity Network Technology for User-oriented Multimedia,* Red de Calidad para Multimedia Orientada al Usuario), fue presentada por un grupo formado por RedIRIS (España), DANTE, DFN (Alemania), GR-NET (Grecia), INFN (Italia), Renater (Francia) y SWITCH (Suiza) junto con Telebit (Dinamarca) en 1997.

Los objetivos generales del Proyecto QUANTUM fueron explorar y posteriormente implementar formas de proporcionar una calidad de servicio mejorada, particularmente para aplicaciones multimedia, a través de una red internacional de muy alta velocidad (hasta 155 Mbps) [14, 15].

1.2.3 TEN – 155 (Trans-European Network – 155 Mbps)

En 1998, surgió la red Trans-Europa TEN – 155. La cual proporcionó la red de investigación europea de próxima generación, para facilitar las actividades de desarrollo cooperativo en Europa cada vez más basadas en el uso de servicios multimedia. Ofreciendo un servicio IP, así como un servicio de ancho de banda administrado a través de una superposición de cajeros automáticos para grupos específicos de usuarios y para la configuración temporal de rutas virtuales, con ancho de banda garantizado entre las redes de investigación nacionales.

La adquisición por parte de DANTE del ancho de banda a precios que tenían alguna relación práctica con los costos subyacentes, a través de adquisiciones realizadas en un momento en que la liberalización del mercado europeo de telecomunicaciones había comenzado a tener un impacto [16,17].

En la figura 7 se muestra la topología de la red TEN - 155.

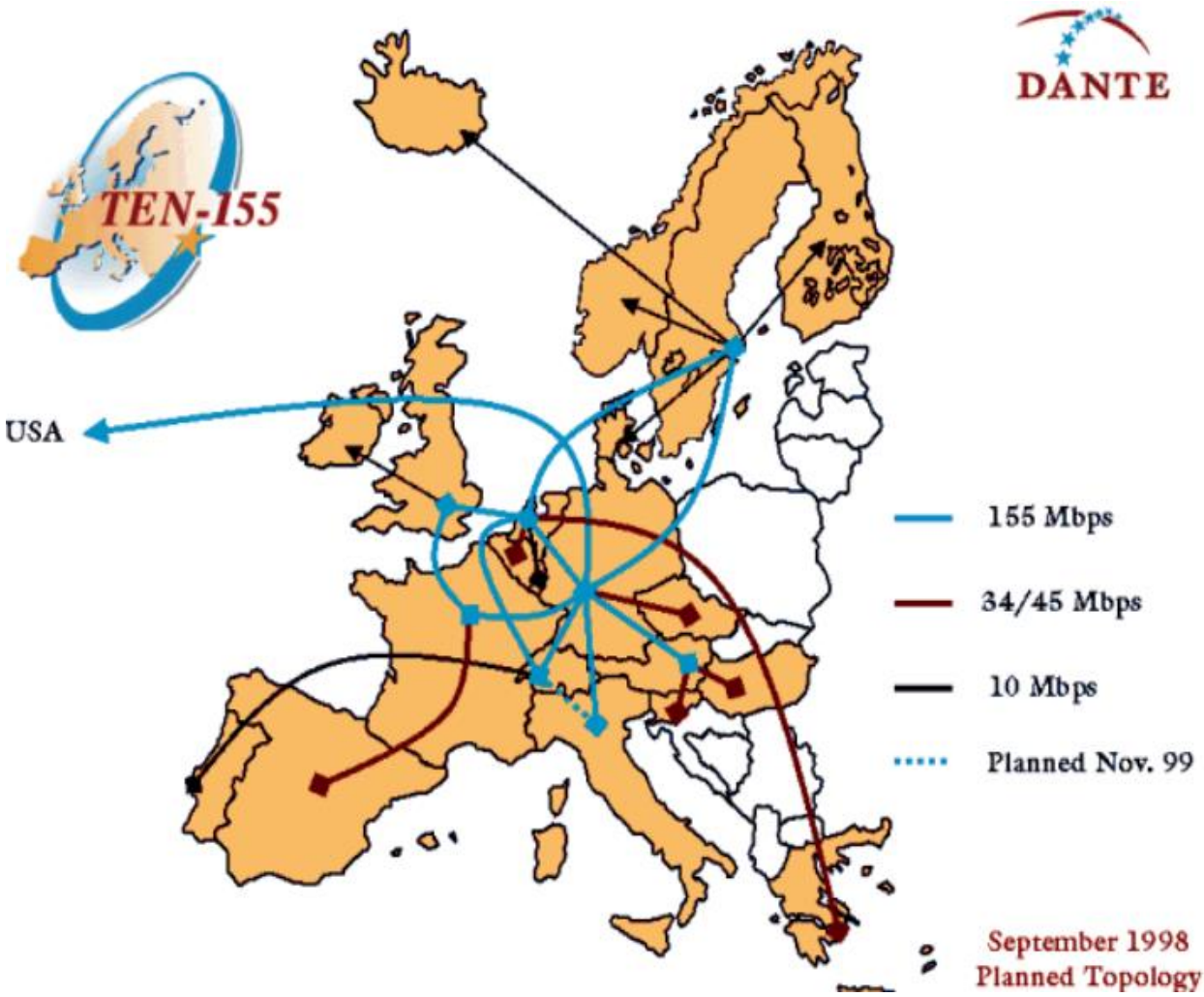


Figura 7. Topología de la red TEN – 155 en 1998 [18].

Para el periodo 1998-2001, TEN-155 alcanzó velocidades de hasta 622 Mbps y en mayo de 2001 se conectó con la red SINET (*Science Information Network,* Redes de Información Científica) de Japón e Israel [19].

1.2.4 Red avanzada Europea GEANT

GEANT (*European Advanced Network*), la red informática europea más grande del mundo, está dedicada a la investigación y la enseñanza. GEANT se puso en marcha el 1 de noviembre del año 2000, esta se usó para mejorar la infraestructura europea de redes de investigación y enseñanza con una inversión de 200 millones de euros financiados por DANTE y la comisión Europea.

Esta red fue sucesora del proyecto TEN-155. En diciembre 2001 GEANT se distribuyó en el continente europeo con 27 países miembros y conexiones en Gbps, como se observa en la figura 8 [20].

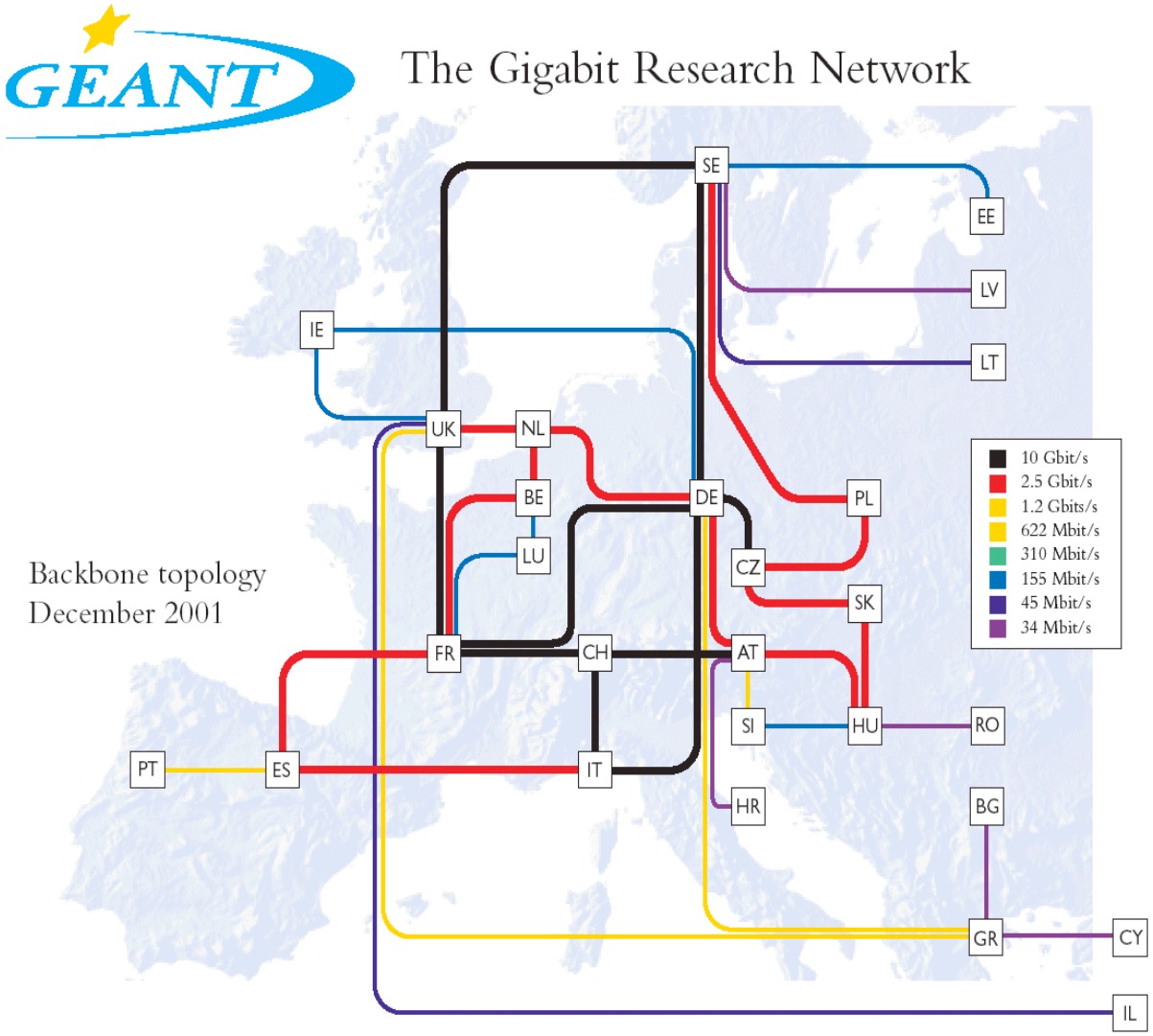


Figura 8. Topología de red GEANT del año 2001 [21].

La comisión Europea contribuyó el 2 de septiembre con 93 millones de euros, los cuales sirvieron para la investigación y la educación de la primera red de comunicaciones de Europa GÉANT. Para el año 2004 conectó 33 países a través de sus redes nacionales como se observa en la figura 9 [22].

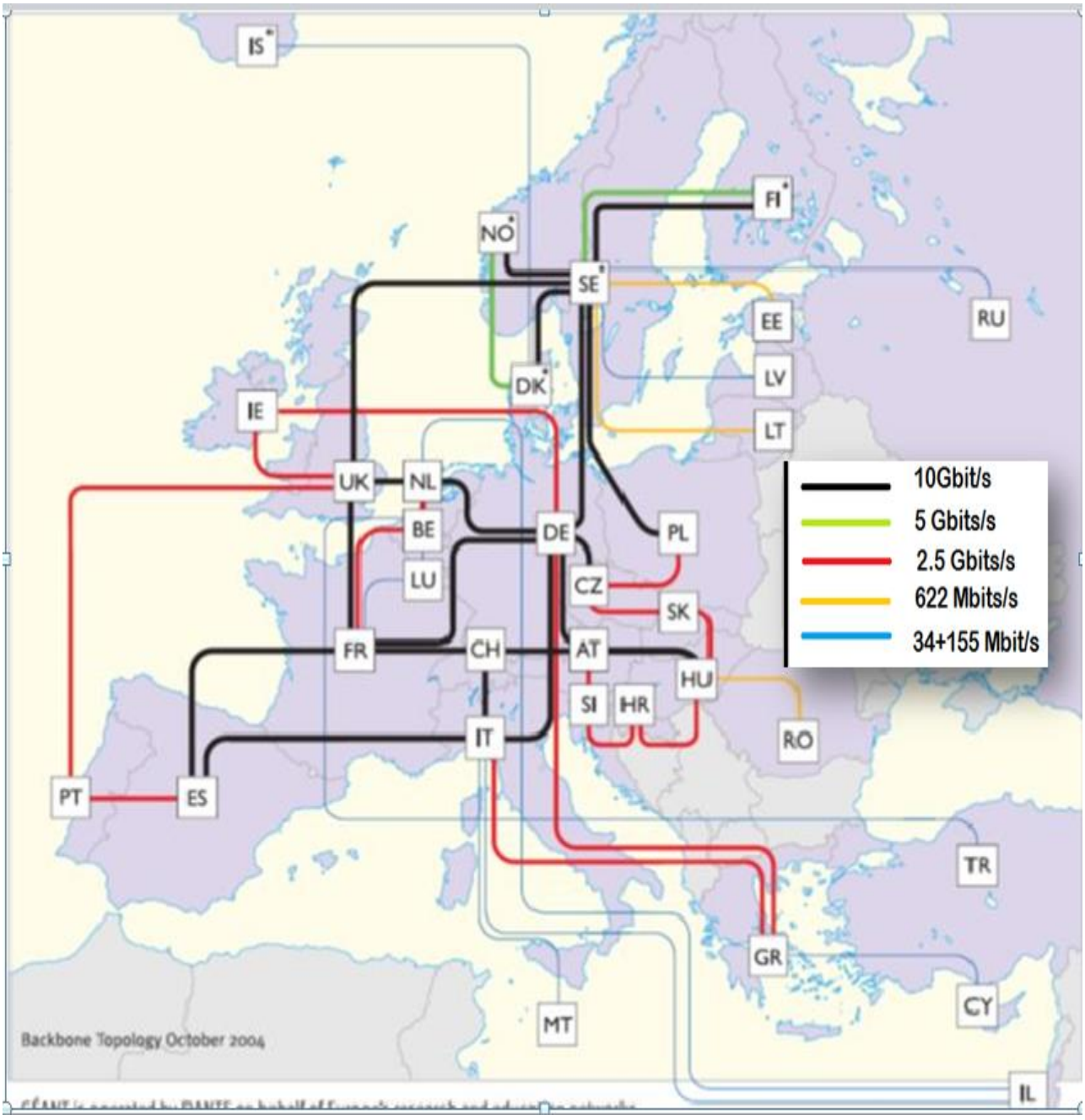


Figura 9. Topología GEANT octubre 2004 [23].

1.2.5 Proyecto ALICE Y GEANT

ALICE (América Latina Interconectada Con Europa) fue creada en junio de 2003, por la Comisión Europea y los representantes de DANTE, el proyecto ALICE comenzó, con una inversión de 12.5 millones de euros (10 millones de euros aportados por la Comisión Europea y 2.5 millones de euros, por los socios latinoamericanos), esto ayudó para la creación de una infraestructura de redes de investigación en América Latina e interconectarla con su par europea, GEANT, mediante el protocolo de Internet (IP), con esta inversión se logró conectar NREN Argentina, Brasil, Chile, Panamá y México con la red GEANT, entre los PoP de España y Brasil, con una conexión de 622 Mbps, que se observa en la figura 10 [24].

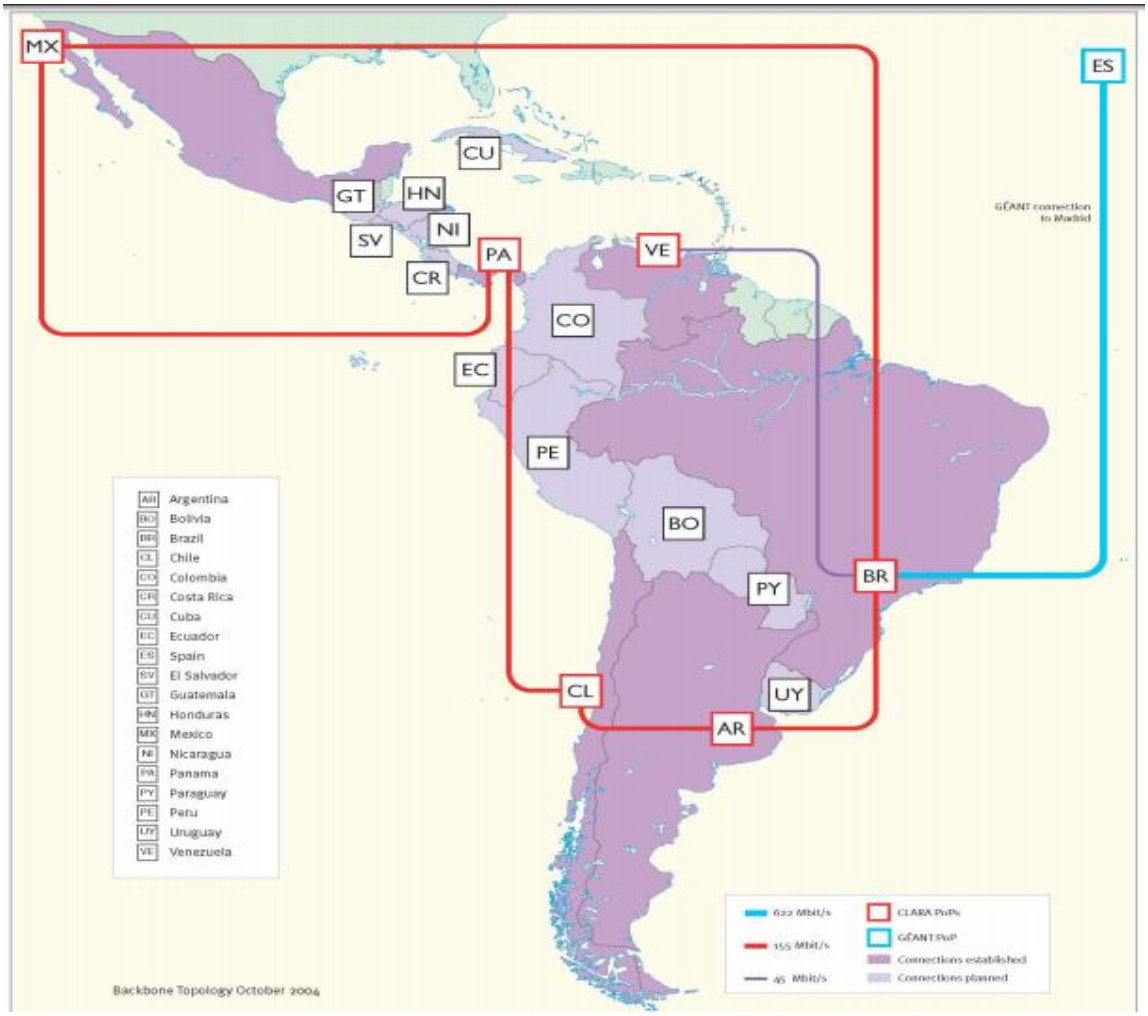


Figura 10. Proyecto ALICE [25].

Este proyecto fue financiado hasta mayo del 2006 con 10 Millones de Euros donados por el Programa @LIS de Cooperación de la Comisión Europea, que persiguió fomentar la Sociedad de la Información en la región, sin embargo, gracias a resultados favorables el proyecto se extendió a marzo 2008 [26]. En la figura 11 se observa la conexión de CLARA con GEANT en el año 2008.

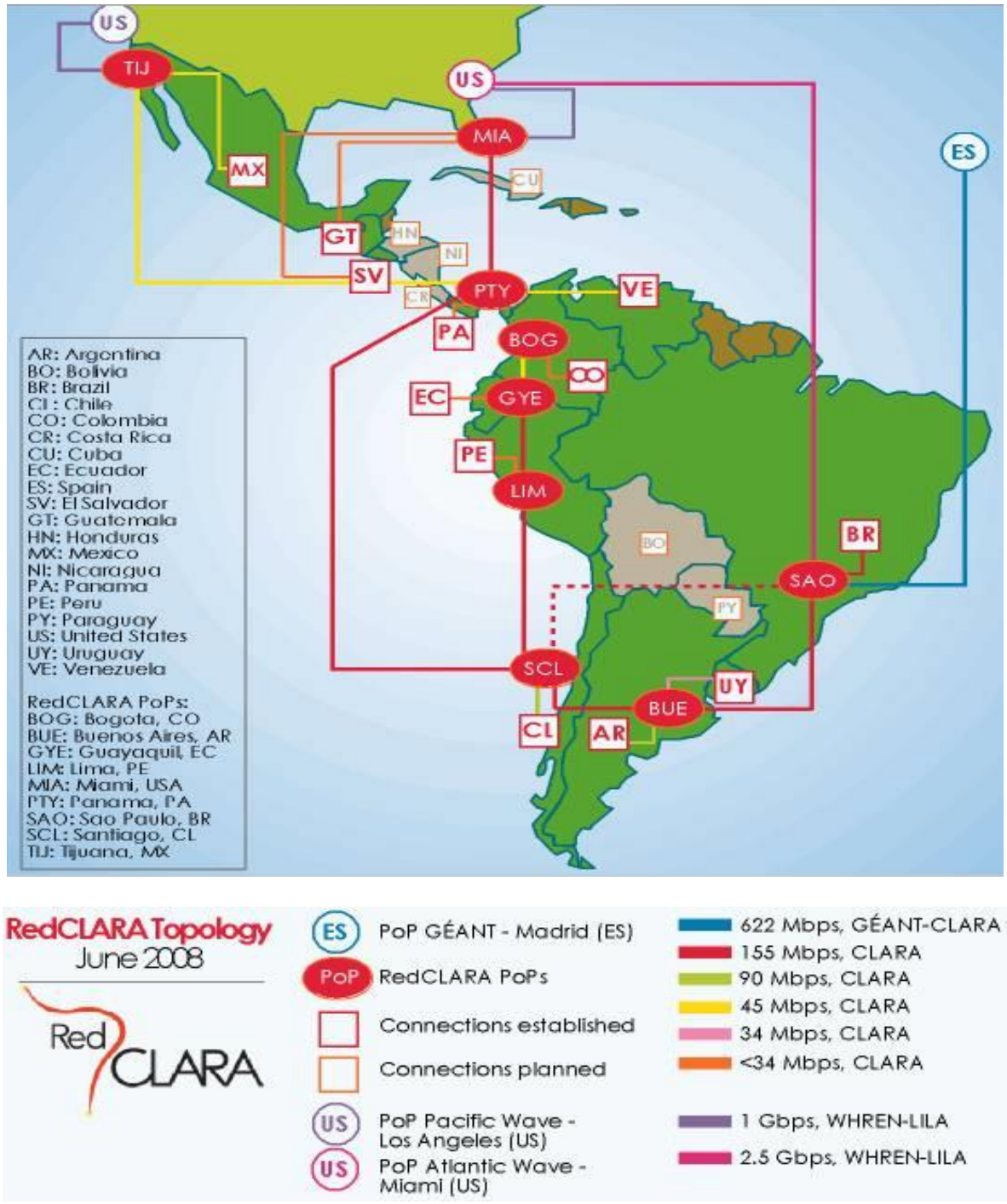


Figura 11. Conexión de CLARA con GEANT año 2008[27].

1.2.6 GEANT 2

GEANT2 fue anunciada en septiembre 2004 con una inversión de 200 millones de euros por parte de la Unión Europea, DANTE, TERENA, y las NREN de Europa. GEANT2 fue la red encargada de conectar las redes académicas de los países europeos entre sí, conectándolos con la red CLARA. Unió a 34 países europeos y cerca de 30 millones de usuarios, con una red hibrida se distribuyó por Europa, ofreció servicios avanzados como IPV6, enlaces de mayor velocidad de transmisión, Premium IP, *Multicast*. La diferencia que tienen las topologías GEANT y GEANT2 es la infraestructura de la fibra óptica oscura. La topología de GEANT2 se observa en la figura 12.

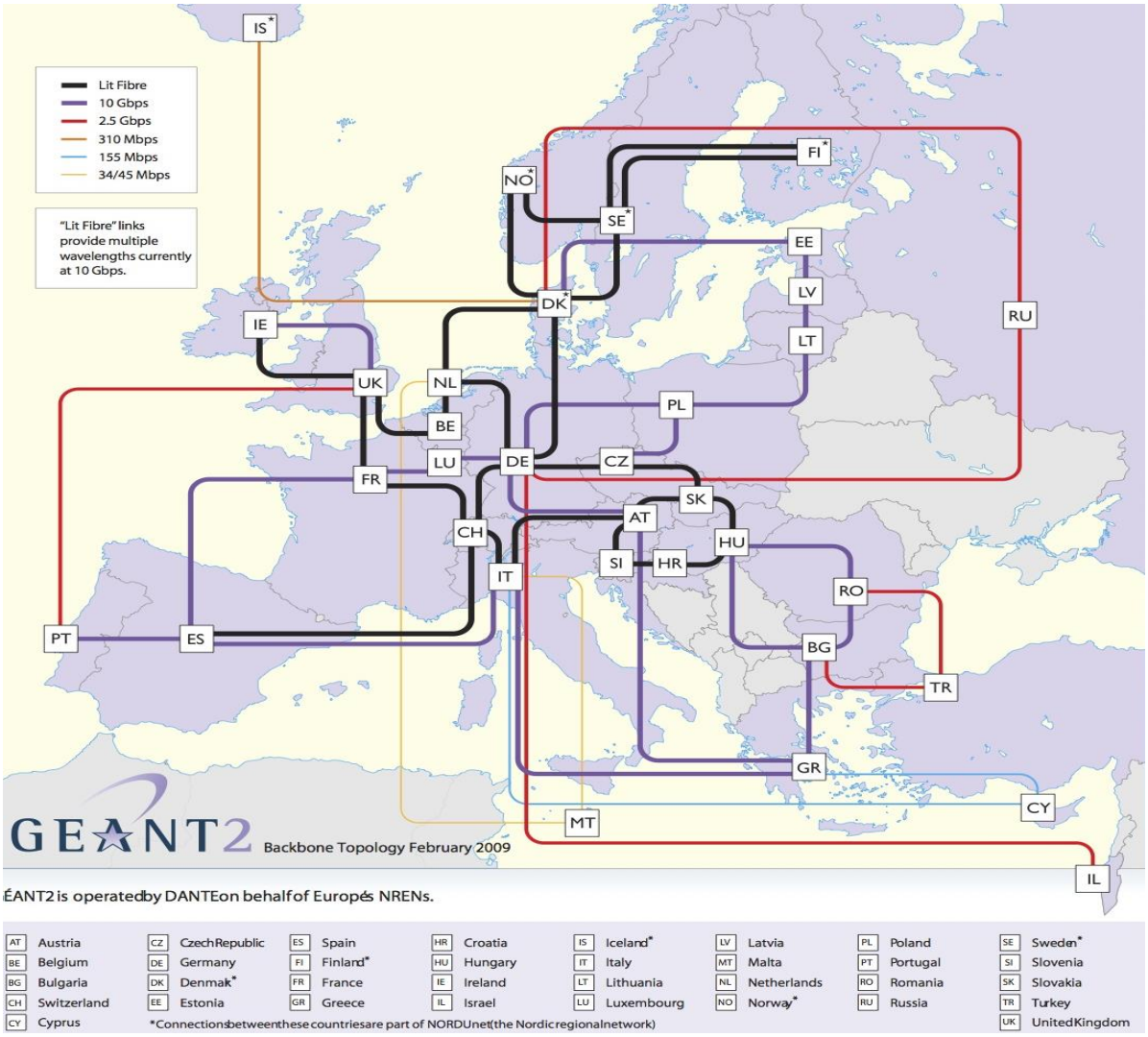


Figura 12. Topología GEANT2 año 2009 [28].

1.2.7 Proyecto ALICE2

El 1 de diciembre del año 2008, inició la segunda etapa del proyecto ALICE con el apoyo de la Comisión Europea a través del programa @LIS2 y la red CLARA, con una inversión de 12 millones de euros por parte de Europa y 6 millones de euros por parte de América Latina, cuyo objetivo fue apoyar la investigación entre América Latina y Europa por medio de una infraestructura de red óptica. El proyecto finalizó en 2013 y en el año 2014 la red CLARA se conectó con la red GEANT con una velocidad de 5 Gbps, como se observa en la figura 13 [29].

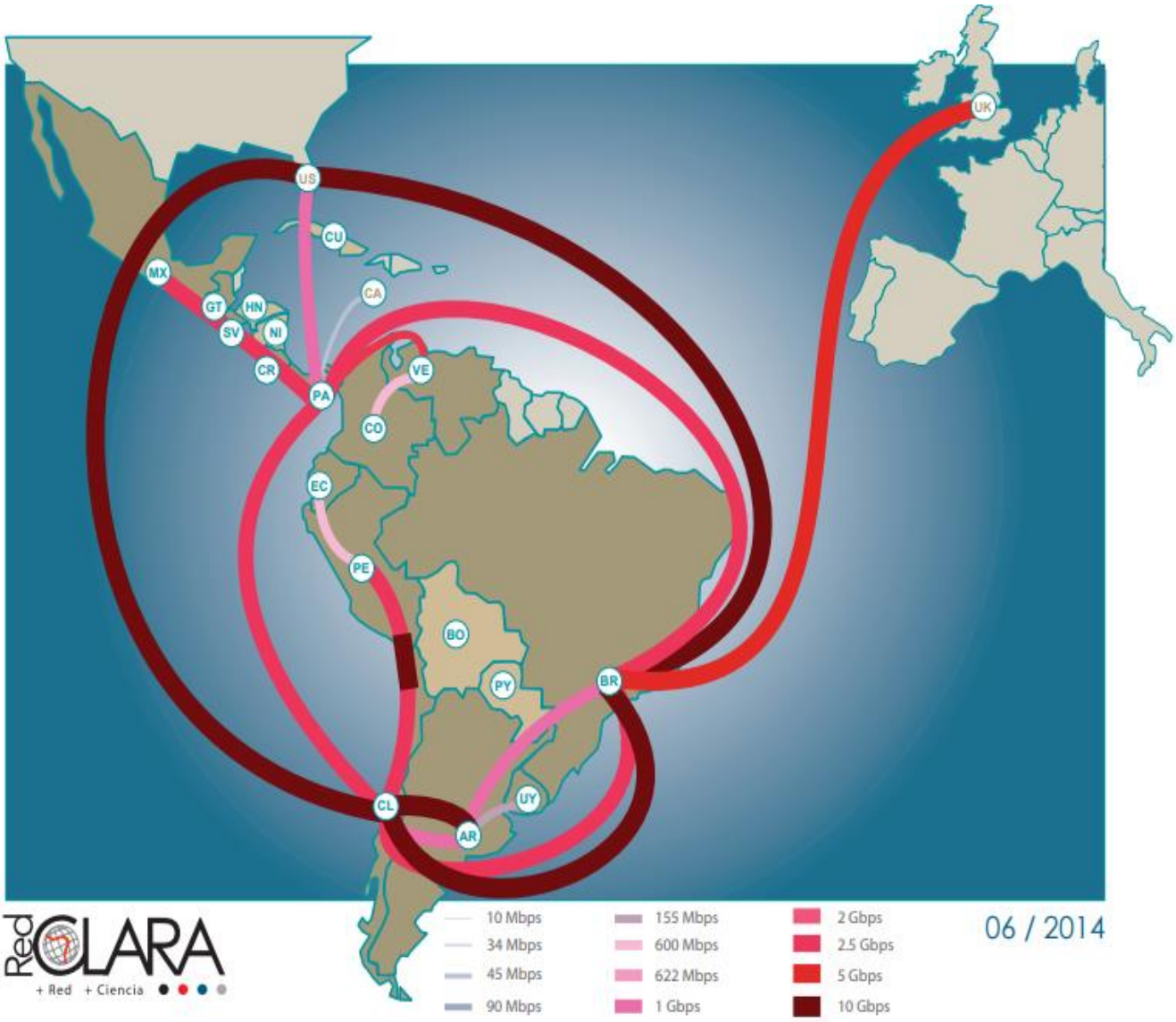


Figura 13. Conexión de la red CLARA con GEANT [30].

1.2.8 GEANT3

GEANT3 o tercera generación, inició el 1 de abril 2009, cuyo objetivo fue ofrecer un beneficio a la sociedad al permitir que las comunidades de investigación de Europa y el mundo transformen la forma en que colaboran en una investigación innovadora, fue financiada por la Comisión Europea con 93 millones de euros durante cuatro años. Contó con 34 socios del proyecto: 32 RNIE europeas, DANTE y TERENA; y cuatro NRENs Asociadas. GEANT3 distribuyó infraestructura de fibra oscura a 19 países europeos, cómo se observa en la figura 14 [31].

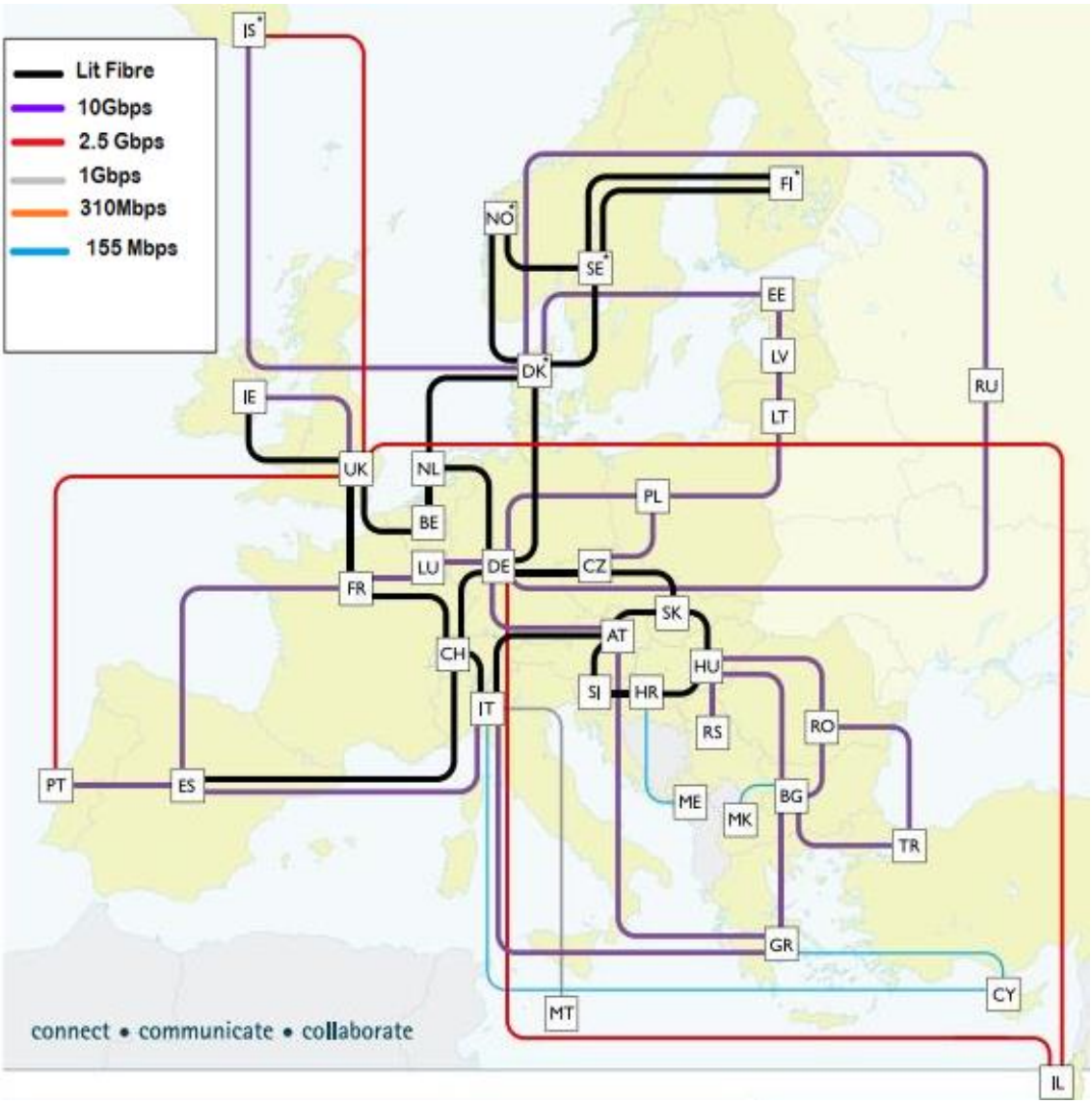


Figura 14. Topología GEANT 3, año 2010 [32].

1.2.9 GEANT 3 PLUS

GEANT 3 PLUS comenzó el 1 de Abril 2013, con un presupuesto de 72 millones de euros financiados por VII Programa Marco de I+D de la Unión Europea, tuvo una duración de dos años, finalizó el 31 de marzo 2015 conectando a 41 países con velocidad de hasta 100 Gbps, como se observa en la figura 15 [33].

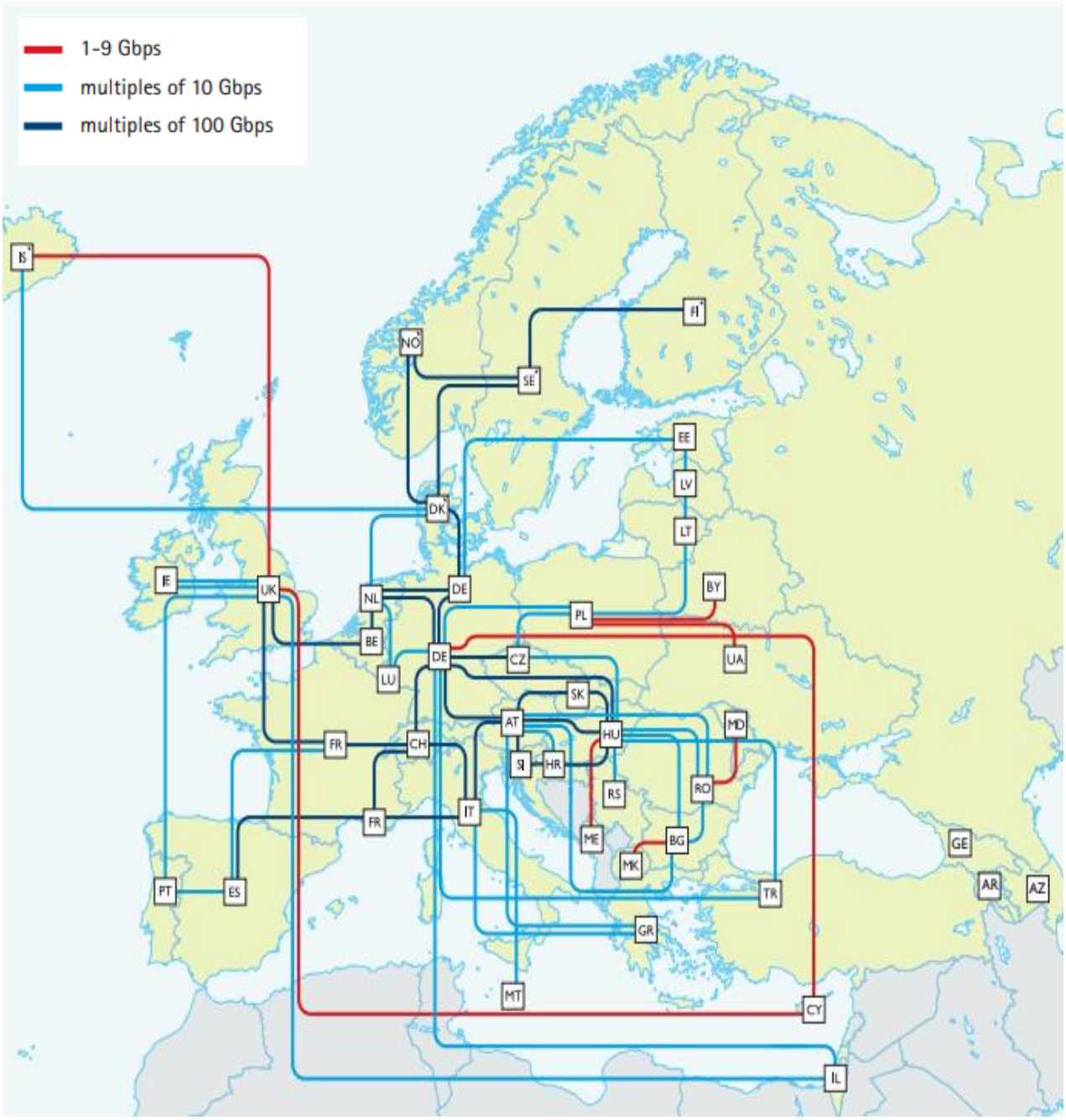


Figura 15. Topología GEANT 3, año 2015 [34].

1.2.10 GEANT 4

GEANT cuarta generación, inició el 1 de mayo del 2016, finalizando el 31 de agosto del 2017, la red GEANT interconecta las NREN de Europa, operó a velocidades de hasta 500 Gbps y llegó a más de la mitad de los países del mundo, conectó a más de 50 millones de usuarios y 10,000 instituciones de investigación en 41 países europeos, como se observa en la figura 16 [35].

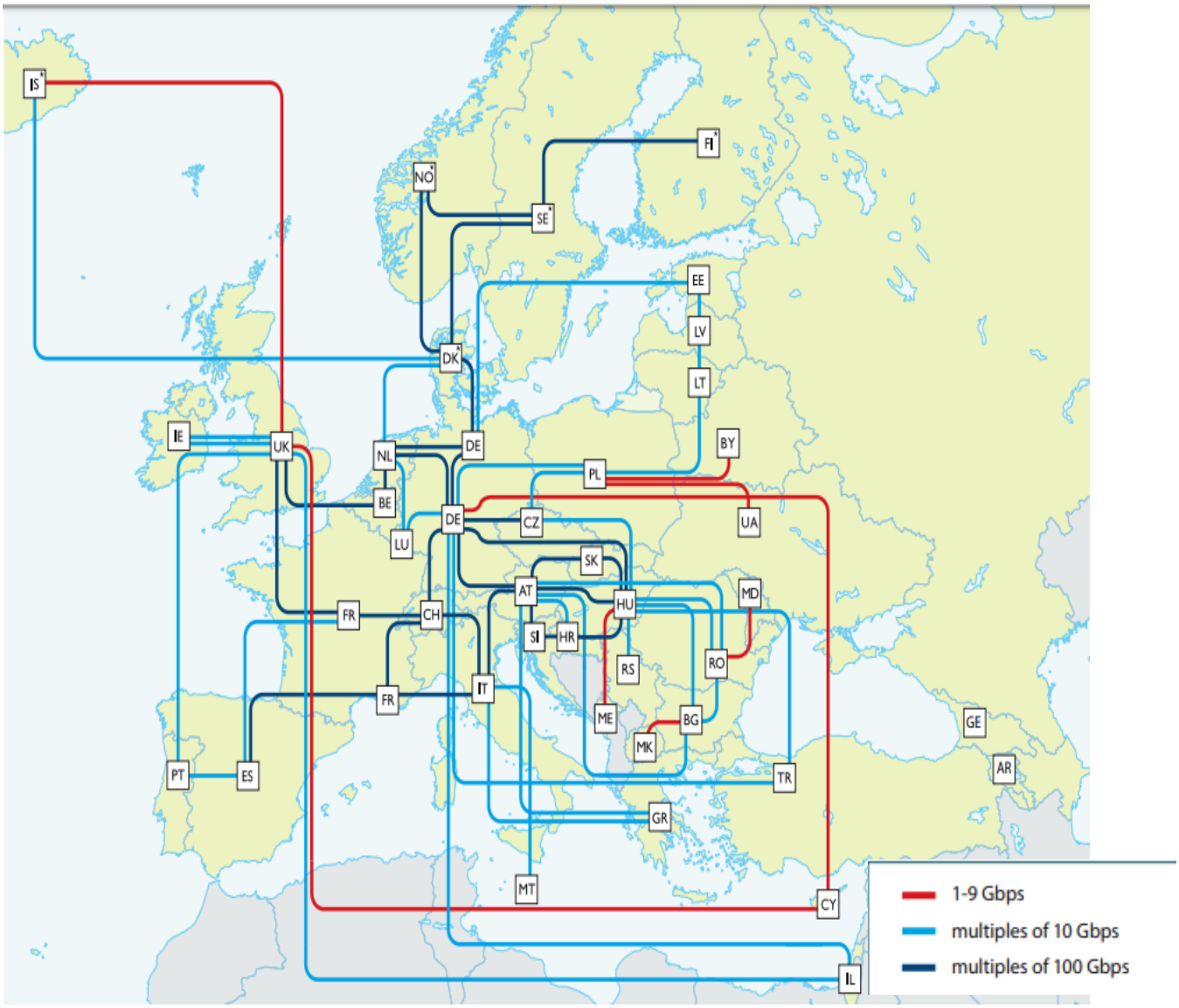


Figura 16. Topología GEANT año 2017 [36].

Este proyecto tuvo un segundo periodo de extensión del 1 de septiembre del 2017 a 31 de diciembre 2018, conectó a 43 países europeos, con infraestructura de red que le permitió ofrecer velocidades de conexión superiores a los 100 Gbps, como se muestra en la figura 17 [35].

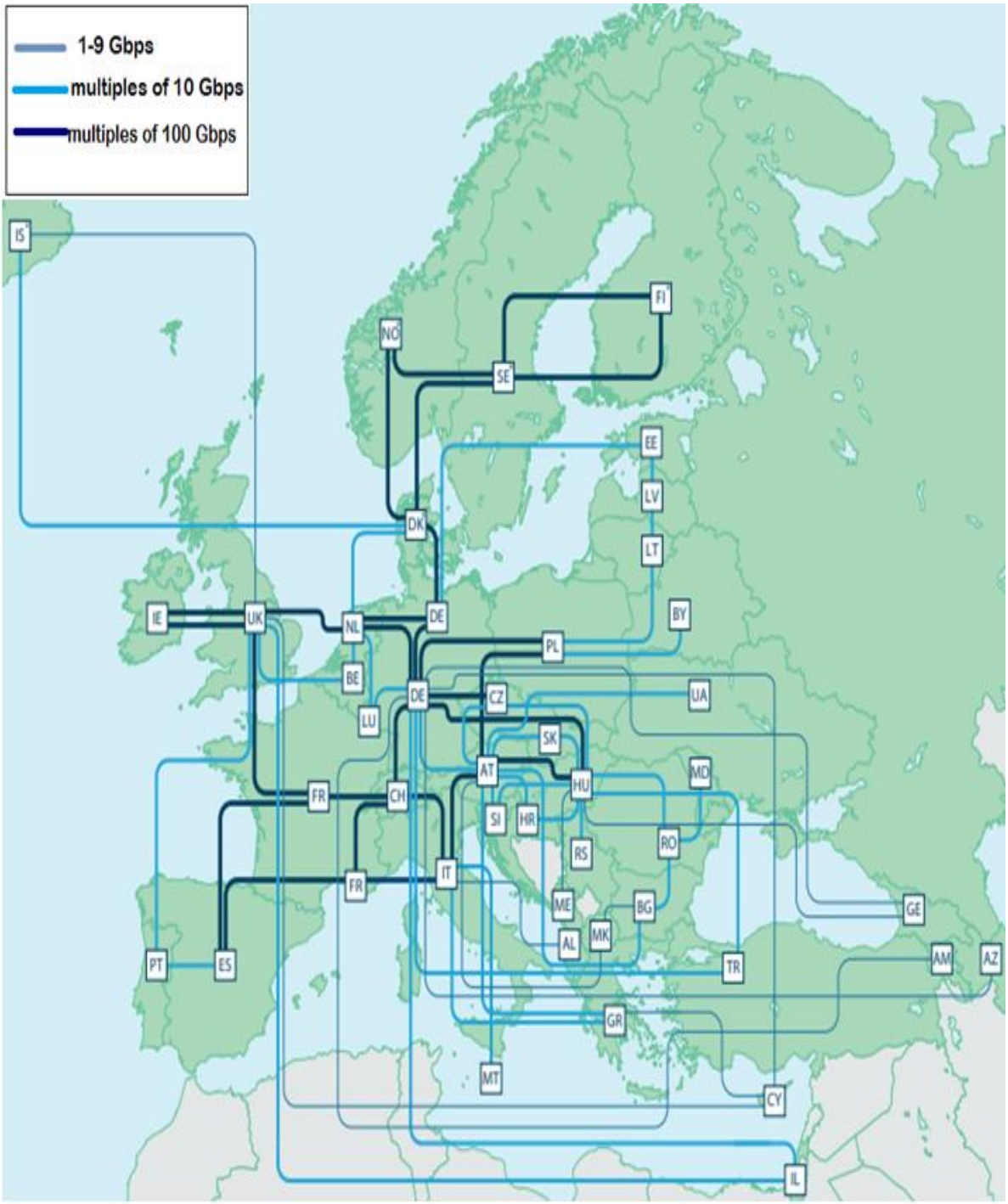


Figura 17. Topología GEANT4, año 2018 [37].

En la tabla 1 se muestran los 43 países miembros de GEANT de diciembre 2018.

AL- Albania	DE - Alemania	IE - Irlanda	MT – Malta	TR - Turquía
AM - Armenia	DK - Dinamarca	IL – Israel	NL - Holanda	UK - Reino Unido
AT - Austria	EE - Estonia	IS - Islandia	NO - Noruega	UA - Ucrania
AZ - Azerbaiyán	ES - España	IT – Italia	PL - Polonia	
BE - Bélgica	FI - Finlandia	LT - Lituania	PT - Portugal	
BG - Bulgaria	FR - Francia	LU -Luxemburgo	RO - Rumania	
BY - Bielorrusia	GE - Georgia	LV - Letonia	RS - Serbia	
CH - Suiza	GR - Grecia	MD - Moldavia	SE - Suecia	
CY - Chipre	HR - Croacia	ME -Montenegro	SI - Eslovenia	
CZ-Rep. Checa	HU - Hungría	MK - Rep. de Macedonia	SK - Eslovaquia	

Tabla 1. Miembros de GEANT, año 2018 [37].

El proyecto GEANT ha permitido el desarrollo de proyectos e investigaciones tecnológicas y científicas, como lo son: la física de partículas, espacio, salud, medicina, observación de la tierra, energía, arte y educación [38].

La red GEANT es el presente y futuro de la Internet Europea. La conectividad global ha seguido creciendo, se han iniciado y desarrollado nuevos proyectos, así como se han aumentado las velocidades de acceso y se han logrado conexiones y apoyo financieros de otras organizaciones tanto de Europa como de otros continentes. En la figura 18, se observa la conexión más actualizada que tiene la red GEANT con el resto de los continentes.

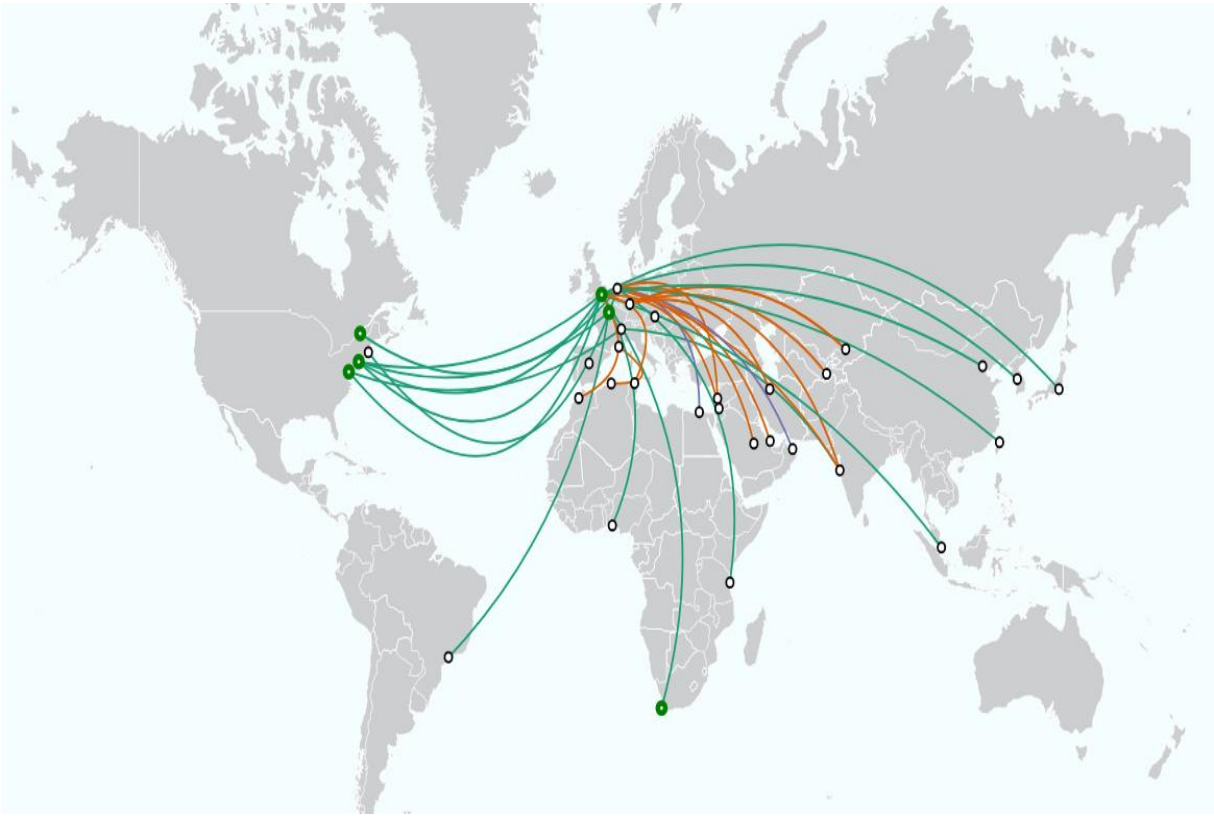


Figura 18. Conexión de la red GEANT en 2019 hacia otros continentes [39].

En ADVNETLAB se han realizado trabajos previos relacionados con la simulación y emulación de la conectividad y gestión de *backbone* de las redes avanzadas CUDI, CLARA, All America, RAAP, INTERNET 2 y REUNA [40 - 45].

Estructura de la tesis

La presente tesis presenta una revisión a los protocolos de enrutamiento OSPF y a los protocolos de gestión SNMP en el capítulo 2, ya que estos protocolos se usarán para lograr nuestros objetivos.

En el capítulo 3, se aborda la metodología empleada para la emulación de la red GEANT, de modo que se generaron los circuitos correspondientes a la topología más actualizada de GEANT. También se configuraron los protocolos OSPF y SNMP con el protocolo IPV6 en el emulador GNS3.

En el capítulo 4, se muestran los resultados de la emulación, así como también se muestran las conclusiones del trabajo.

Capítulo 2.
Protocolos

2.1 INTRODUCCIÓN A IPV6

En el presente documento, la siguiente información se usó para el formato de direcciones IPV6, así como, para entender mejor cómo se escriben las direcciones IPV6 al utilizarlas en la emulación.

Las direcciones IPV4 de 32 bits, nos permiten obtener 4.294.967.296 (2^{32}) direcciones de red diferentes, lo cual, limita el crecimiento de Internet. Esta limitación provocó que la IETF (*Internet Engineering Task Force)* propusiera IPV6.

El protocolo de internet versión 6 se creó para sustituir al estándar IPV4. Las especificaciones de IPV6 están definidas en los RFC 2373 Y 2374 [46].

Los cambios de IPV4 a IPV6 son los siguientes [47]:

a. IPV6 incrementa el tamaño de direcciones de 32 bits a 128 bits.
b. Ciertos campos del encabezado IPV4 se han eliminado o se hicieron opcionales, para limitar el costo del ancho de banda del encabezado IPV6.
c. El etiquetado de flujo permite el etiquetado de ciertos paquetes que pertenecen a flujos de tráfico.
d. En IPV6 se agregó la autenticación y privacidad en e+l envió de paquetes.

2.1.1. Formato de encabezado IPV6

En la figura 19 se muestra el encabezado de IPV6 [47].

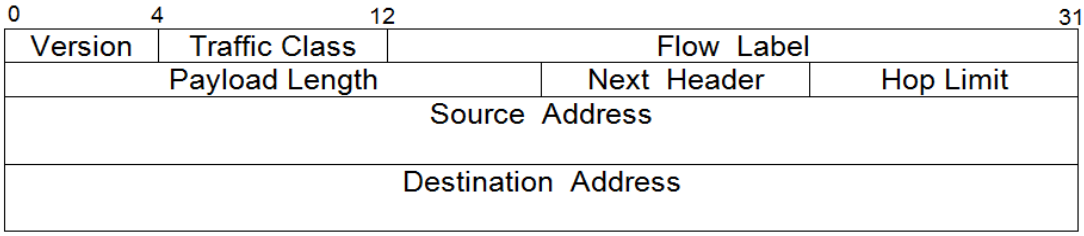


Figura 19. Encabezado de IPV6.

Dónde:

a) Version: Número de versión del protocolo de internet de 4 bits.

b) Traffic Class: Clase de tráfico de 8 bits.

c) Flow Label: Etiqueta de flujo de 20 bits.

d) Payload Length: Longitud de la carga útil IPV6 de 16 bits.

e) Next Header: El encabezado siguiente de 8 bits de IPV6, utiliza los mismos valores que el protocolo IPV4.

f) Hop Limit: Limite de saltos de 8 bits. Disminuye en 1 por cada nodo que reenvía el paquete.

g) Source Address: Dirección origen de 128 bits.

h) Destination Address: Dirección destino de 128 bits.

2.1.1.1. Extensiones de encabezado IPV6

Los encabezados de extensión de IPV6 contienen información utilizada por los dispositivos de red como enrutadores y conmutadores. La longitud de cada encabezado de extensión es un múltiplo entero de 8 octetos, esto permite que los encabezados de extensión siguientes utilicen estructuras de 8 octetos.

a. Encabezado de opción de salto a salto. Lleva información opcional, la cual debe de ser examinada por cada nodo a lo largo de la ruta de entrega de un paquete y tiene el siguiente formato, que se observa en la figura 20.

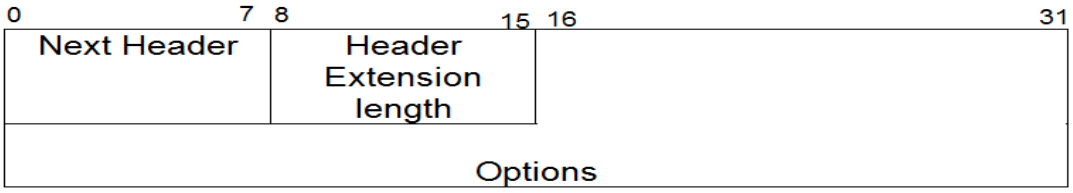


Figura 20. Encabezado de opciones salto a salto para IPV6.

- *Next header*: Encabezado siguiente de 8 bits, identifica el tipo del siguiente encabezado, utiliza los mismos valores que IPV4.
- *Header Extension length*: Longitud del encabezado salto a salto de 8 octetos.

- *Options*: Opciones, campó de longitud variable de 8 octetos de largo.

b. Opciones de destino: Este se usa para especificar los parámetros de entrega de paquetes para los destinos intermedios o el destino final, esté encabezado tiene un valor de 60 en el encabezado del paquete anterior. Su formato se observa en la figura 21.

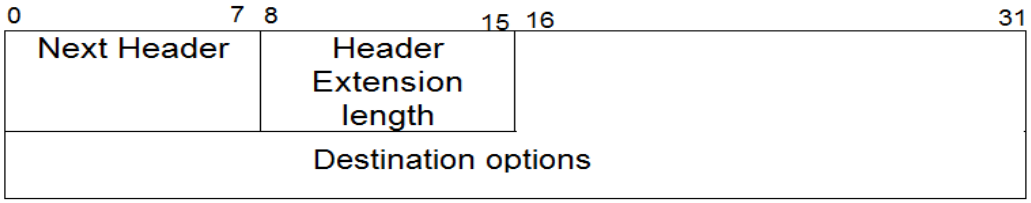


Figura 21. Encabezado de opciones de destino para IPV6.

- *Destination options*: Campo de longitud variable, el encabezado completo de opciones de destino es un entero de 8 octetos de largo.

c. Enrutamiento: El encabezado de enrutamiento es utilizado para enumerar uno o más nodos intermedios para ser “visitados” en el camino hacia el paquete destino. Cuándo un paquete usa este encabezado, el valor del siguiente encabezado del paquete anterior debe ser 43 y tiene el siguiente formato que se observa en la figura 22.

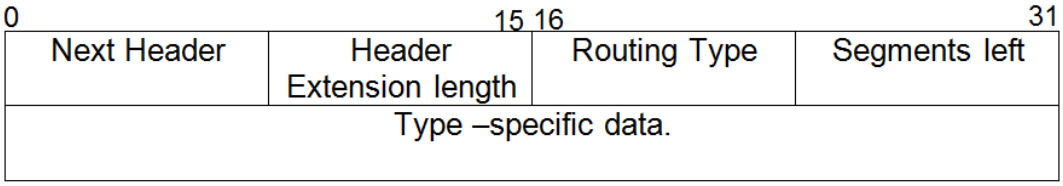


Figura 22. Encabezado de enrutamiento para IPV6.

- *Routing Type*: Tipo de enrutamiento de 8 bits.
- *Segments left*: Segmentos a la izquierda de 8 bits, número de segmentos de rutas restantes.
- *Type-specific data*: Es un campo de longitud variable, de 8 octetos de largo.

d. Fragmento: El encabezado de fragmento es utilizado por IPV6 para enviar un paquete más grande que entra en la ruta MTU (*Maximum Transmission Unit*: Unidad de transmisión máxima de una conexión de red) a su destino, donde su formato de encabezado se observa en la figura 23.

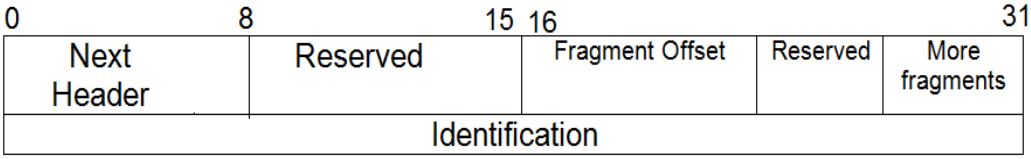


Figura 23. Encabezado de fragmento para IPV6.

e. Autentificación: Proporciona autentificación e integridad de datos, cuándo un paquete usa este encabezado el valor del siguiente encabezado del paquete anterior debe ser 51.
f. Carga de seguridad de encapsulación: Proporciona confidencialidad de datos, autentificación de datos, protección para paquetes de carga de seguridad ESP (*Encapsulating Security Payload*); cuando un paquete usa este encabezado, él valor del siguiente encabezado del paquete anterior debe ser 50.
g. Dirección IP de destino: Identifica el dispositivo *host*, o la interfaz en un nodo al que se enviará el paquete IPV6 [48].

2.1.2 Representación de direcciones de IPV6

Las direcciones IPV6, nos permite representar 2^{128} direcciones diferentes, estas direcciones se representan con 32 dígitos hexadecimales, reunidos en ocho grupos de cuatro dígitos hexadecimales cada uno, como se observa en la dirección 1:

20AB:0D9B:80AF:08DE:A3A0:8A32:037F:703A [Dirección 1]

Si algunos de los grupos tienen el valor de 0000 se puede comprimir tal como se observa, a este formato se le llama forma comprimida. Se muestra un ejemplo a continuación:

20AB:0D9B:80AF:08DE:A3A0:0000:037F:703A

20AB:0D9B:80AF:08DE:A3A0::037F:703A

Si en una dirección IPV6 existe una secuencia de dos o más grupos consecutivos de ceros se comprimen de la siguiente manera:

20AB:0D9B:80AF:0000:0000:0000:037F:703A

20AB:0D9B:80AF::037F:703A

En el siguiente ejemplo se muestra la dirección 2:

0000:0000:0000:0000:0000:0000:0000:0001 [dirección 2]

0:0:0:0:0:0:0:1 [dirección *loopback*]

:: 1

Algunas veces las direcciones IPV6 están organizadas por dos partes lógicas de 64 bits cada una. Los primeros 64 bits son un prefijo de red y los otros 64 bits son los que identifica la interfaz a la que está asignada la dirección [49].

Existen tres tipos de direcciones IPV6 [46]:

a) *Unicast*. Un identificador para una sola interfaz.

b) *Anycast.* Una dirección es asignada a más de una interfaz, (que generalmente pertenece a diferentes nodos). Los nodos son elementos de una red de datos. Cada nodo es un dispositivo electrónico que es capaz de recibir la señal del medio de transmisión empleado.

c) *Multicast:* Identificador para un conjunto de interfaces que normalmente pertenece a diferentes nodos.

2.2 PROTOCOLOS DE ENRUTAMIENTO

Una red de datos puede verse como un laberinto, con múltiples opciones de salidas, con el riesgo de que la información nunca llegue a su destino. Para garantizar la entrega de los datos se utilizan equipos conocidos como *routers*.

Los *routers* utilizan los protocolos de enrutamiento para mantener actualizadas las tablas de enrutamiento y así poder elegir la mejor ruta [50].

Estos protocolos trabajan en la capa tres del modelo ISO/OSI.

La tabla de enrutamiento del *router* contiene la siguiente información:

a) Dirección de red: Son las redes conocidas por el *router*.

b) Interfaz: Es la interfaz usada por el *router* para llegar a una red dada.

c) Métrica: Se refiere a la distancia o coste para llegar a la red destino.

2.2.1 Enrutamiento Estático

Las rutas estáticas son determinadas manualmente por los administradores de red. Estas rutas no necesitan muchos recursos del sistema y son recomendables utilizarlas cuando las redes no cuentan con muchos *routers* conectados

2.2.2 Enrutamiento Dinámico

Las rutas dinámicas son definidas automáticamente por el *router* mediante el intercambio de información entre *routers.* Este tipo de enrutamiento es recomendable utilizarlo cuando la red sea muy grande, es decir, la red cuenta con un gran número de *routers.*

Algunos protocolos que utilizan este enrutamiento son los siguientes:

RIP (*Routing Information Protocol*)

IGRP (*Interior Gateway Routing Protocol*)

EIGRP (*Enhanced Interior Gateway Routing Protocol*)

OSPF (*Open Shortest Path First*)

BGP (*Border Gateway Protocol*)

Estos protocolos de enrutamiento se clasifican en dos tipos:

a) Vector distancia

Se le llama enrutamiento por vector distancia, porque se basa en dos factores: la distancia y dirección. La distancia es la métrica, es decir, el número de saltos que va a realizar el *router* para llegar hasta la interfaz deseada. La dirección (vector), es la dirección que va a tomar el *router* para llegar a dicha interfaz.

b) Enlace estado

Cuando una red realiza alguna modificación a su topología, los protocolos de estado enlace realizan una inundación de LSA mediante *mullticast* a toda la red, para realizar actualizaciones en sus tablas de enrutamiento, un ejemplo que utiliza estado enlace es el protocolo OSPF [51].

En la presente tesis se usó estado enlace, debido a que se utilizó el protocolo OSPFV3 para lograr la conectividad de la emulación de la red GEANT.

2.2.3 RIP

RIP (*Routing Information Protocol,* Protocolo de información de Enrutamiento), está basado en el algoritmo de *Bellman Ford*, utiliza el enrutamiento vector distancia, los cuales utilizan como métrica el costo, es decir, el número de saltos para llegar a la interfaz deseada. RIP solo permite 15 saltos como máximo, a partir del salto 16 lo considera inexistente o inalcanzable.

En RIP la información de enrutamiento se propaga de un router a otro por medio de una difusión de IP (*Internet Protocol*), utilizando el protocolo UDP (*User Datagram Protocol*) en el puerto 520.

Es un protocolo de puerta de enlace interna o interior IGP (*Interior Gateway Protocol*) con esto se hace referencia a los protocolos usados dentro de un sistema autónomo. Existen tres versiones de RIPV1, RIPV2, RIPng.

2.2.3.1 RIPV1

Definido por el RFC (*Request For Comments*) 1058. Es un protocolo con clase, esto quiere decir, que no envían información de la máscara de subred en sus actualizaciones de *routing*. RIPV1 funciona bien en una red pequeña, envía actualizaciones cada 30 segundos que contiene la tabla de *routing*, trabaja en la capa de red y utiliza tramas de 32 bits como se observa en la figura 24 [52].

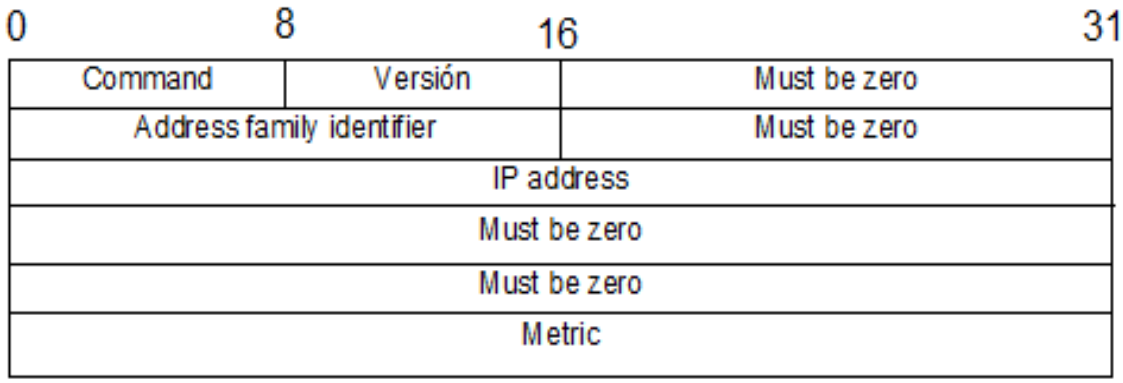


Figura 24. Formato de mensaje RIPV1.

a) *Command* (Comando): 1 para una solicitud o 2 para una respuesta

b) Versión: 1 para RIPV1 *Address family identifier* (Identificador de familias de direcciones): Específica la familia de dirección usada, 2 para IP a menos que realice la solicitud de una tabla de enrutamiento completa, en cuyo caso se establece en 0.

c) *IP Address* (Dirección IP): La dirección de la ruta de destino que puede ser una red, subred o dirección de host.

d) *Metric (*Métrica): Consiste en el conteo de saltos, la métrica total consiste en el total de saltos desde el *router* origen hasta el destino, está tiene una limitación de 15 saltos ya que el salto 16 se considera inaccesible.

2.2.3.2 RIPV2

Después de 10 años que se publicara RIPV1, fue publicado RIPV2 por G. Malkin de la compañía *Bay Networks* en noviembre de 1998, utilizando el estándar RFC 2453. En esta versión de RIPV2 hubo mejoras en comparación con RIPV1 que se mencionan a continuación:

Utilización de máscaras de red, con esto es posible utilizar VSLM (*Variable Length Subnet Mask*)

Autentificación para la transmisión de información de RIP entre vecinos.

El siguiente salto tiene la elección de utilizar máscaras de red, esto permite la utilización de arquitecturas de red discontinuas.

Mediante la dirección de *multicast* 224.0.0.9 se envían las actualizaciones de las tablas RIP.

Aunque RIPV2 haya tenido alguna mejora, sigue teniendo algunas limitaciones como lo es la métrica continúa siendo de 15 saltos como tamaño máximo de la red, lo cual implica que RIPV2 no se puede utilizar en redes de tamaño más grande.

Es un protocolo que al igual que la primera versión sigue generando mucho tráfico al enviar toda la tabla de *routing* en cada actualización, su trama sigue siendo de 32 bits como se observa en la figura 25 [53].

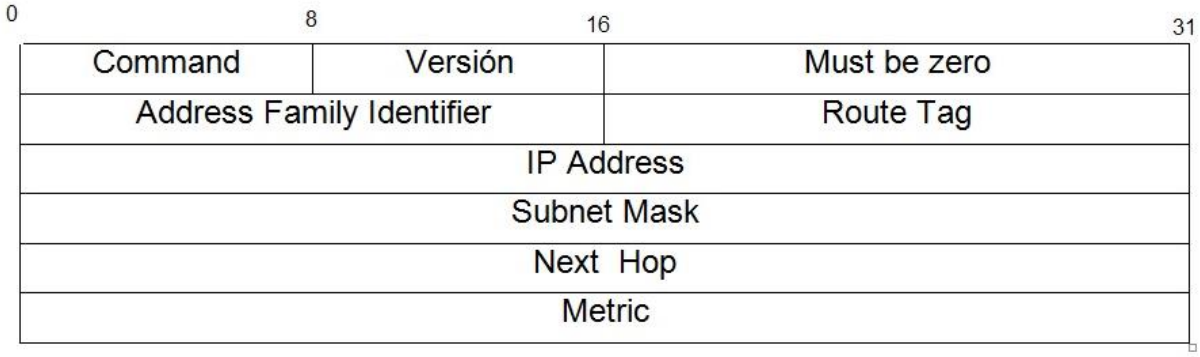


Figura 25. Formato de mensaje RIPV2.

a) *Address Family Identifier* (Identificador de familias de direcciones): Especifica la familia de dirección usada, 2 para IP a menos que realice la solicitud de una tabla de enrutamiento completa, en cuyo caso se establece en 0.

b) *Route Tag* (Etiqueta de ruta): Etiqueta modificable por el usuario según la implementación.

c) *IP Address* (Dirección IP): Dirección IP de la red destino.

d) *Subnet Mask* (Máscara de subred): Máscara de red de la red o subred destino.

e) *Next Hop* (Siguiente salto): Dirección IP del siguiente salto para la red destino.

f) *Metric* (Métrica): Número de saltos (1 al 15)

2.2.3.3 RIPng (Routing Information Protocol next generation)

RIPng se rige por el RFC 2080, básicamente funciona de la misma manera que *RIPv2* con la diferencia de que *RIPng*, sólo es ejecutable en redes IPV6 y su longitud de direcciones IP es de 128 bits.

RIPng es un protocolo de enrutamiento de vector distancia, tal que su métrica al igual que *RIPv1 y RIPv2* es de 15 saltos como máximo, sus actualizaciones ocurren cada 30 segundos y son *multicast* usando la dirección FF02::9 la cual es la dirección del grupo *multicast* de todos los *routers* que estén ejecutando *RIPng.*

RIPng es un protocolo el cual se basa en UDP (*User Datagram Protocol*); todos los enrutadores que utilizan el protocolo *RIPng* reciben y envían datagramas por

medio del puerto UDP 521. El formato del paquete *RIPng* es el que se muestra en la figura 26 [54].

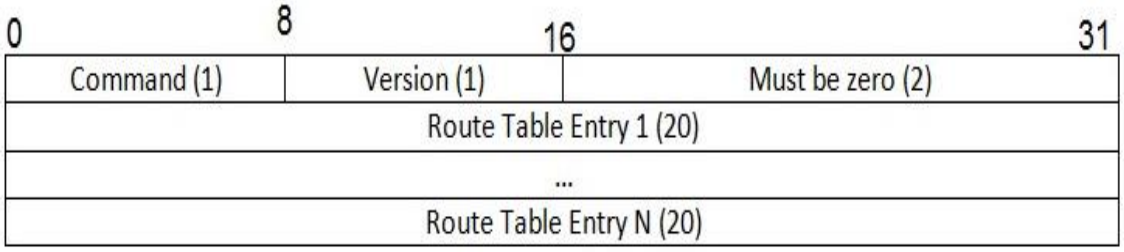


Figura 26. Formato del paquete *RIPng*.

Donde, cada RTE (*RouteTable Entry*), tiene el formato que se observa en la figura 27.

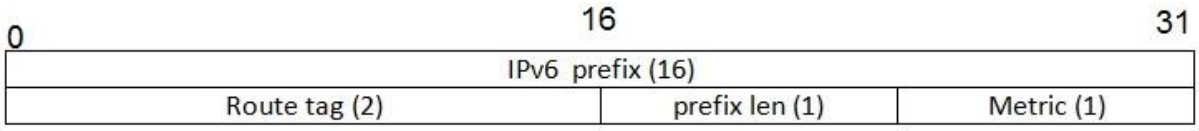


Figura 27. Tablas de ruta de entrada (RTE).

El paquete *RIPng* incluye tres campos, los cuales son:

- *Command*. También conocido como *request* o *response*.
- *Version*. El tipo de versión que usa.
- *Must be zero*. Bits reservados.

RTE (Route Table Entry). Donde cada RTE incluye el prefijo IPV6, *Route Tag* que sirve para separar las rutas internas de las rutas externas, *Prefix length* que es el que determina el número de *bits* y *metric* que es la que define el conteo de saltos.

RIPng soporta dos comandos qué son: *Request* que se utiliza para preguntar por toda o parte de la tabla de ruteo, en muchos casos los *request* son enviados como *multicast*. El segundo comando es response, del cual existen tres tipos: Cuando se

realiza una consulta, una actualización cada 30 segundos a todos los *routers* vecinos y una actualización producto de un cambio de ruta.

El protocolo RIP no se implementó en la emulación, sin embargo, se usó como antecedente para comprender mejor el funcionamiento del protocolo OSPF.

2.2.4 OSPF

OSPF (*Open Shortest Path First*, Primer Camino Más Corto). Con el aumento y crecimiento de las redes y con la limitante de la métrica de RIP, se hizo cada vez más difícil poder utilizar el protocolo RIP.

Es por esto que en 1987, se comenzó con el desarrollo del protocolo OSPF. La primera publicación fue realizada para sistemas operativos UNIX por la IETF (*Internet Engineering Task Force*) y sólo para algunos *routers*. OSPFV1 se publicó en 1989 bajo el RFC 1131, fue un protocolo experimental y nunca se implementó. Para 1991, se desarrolló OSPFV2 por John Moy y se publicó en el RFC 1247, después se actualizo al RFC 2328 en 1998. En 1999, se creó OSPFV3 para IPV6 y se publicó en RFC 2740.

Algunas de sus principales características son las siguientes:

- La utilización de VLSM (*Variable Length Subnet Mask*), para la asignación de una dirección de IP.
- Su métrica es el costo.
- Utiliza el algoritmo *Dijkstra.*
- Utiliza *Multicast* para enviar actualizaciones de estado enlace

2.2.4.1 Encabezado de *OSPFV2*

En la figura 28, se muestra el encabezado de *OSPF***.**

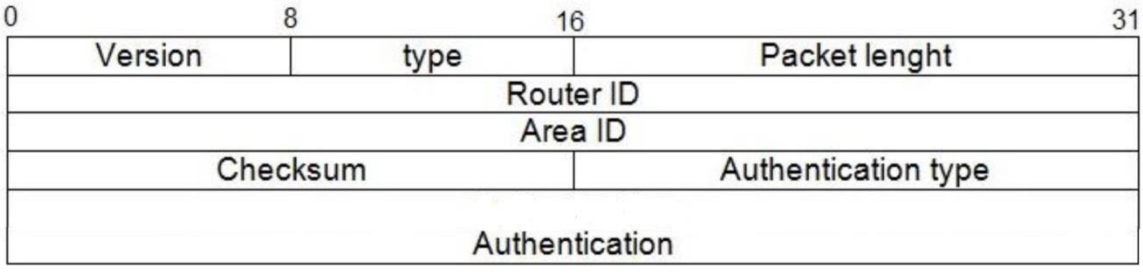


Figura 28. Encabezado de *OSPFV2*.

a) *Router ID*: Identifica al *router* dentro del sistema autónomo.

b) Area ID: El paquete *Hello* tiene que venir de un *router* que esté dentro de la misma área.

c) *Checksum*: Comprueba la integridad del paquete.

d) *Authentication Type*: Asegura la misma autentificación en ambos extremos.

e) *Authentication*: Es utilizado para la seguridad entre sistemas.

2.2.4.2 Tipos de mensajes *OSPFV2*

OSPF utiliza cinco tipos de mensajes diferentes:

1. *Hello.* Identifica a los vecinos y establece comunicación con vecinos conectados directamente. Estos mensajes son enviados con la dirección *multicast* 224.0.0.5 en IPV4 y FF02::5 en IPV6; con una frecuencia en redes *broadcast* cada 10 segundos (por defecto en redes multiacceso y redes *point to point*).
2. *DBD (Database Descriptor, Descripción de la base de datos).* Intercambia información para que un *router* pueda descubrir los datos que le faltan durante la fase de inicialización o sincronización cuando dos nodos han establecido una conectividad.
3. LSR *(Link State Request*, Petición del estado enlace). Pide datos que un *router* se ha dado cuenta que le faltan en su base de datos o que están obsoletos durante la fase de intercambio de información entre dos *router*.
4. *LSU (Link State Update*, Actualización del estado enlace). Se usa como respuesta a los mensajes de petición de estado del enlace y también para informar dinámicamente de los cambios en la topología de la red.
5. *LSAck (Link State Acknowledgements,* Ack del estado del enlace). Confirma la recepción del paquete.

2.2.4.2.1 Paquete *Hello* de *OSPFV2*

El paquete *Hello* de *OSPF* es de tipo 1, el cual se envía periódicamente en todas las interfaces. El paquete *Hello* determina el enrutador designado (DR); el DR es

adyacente a todos los enrutadores en la red cuya función es generar e inundar los LSA.

En la figura 29 se muestra la estructura del paquete *Hello.*

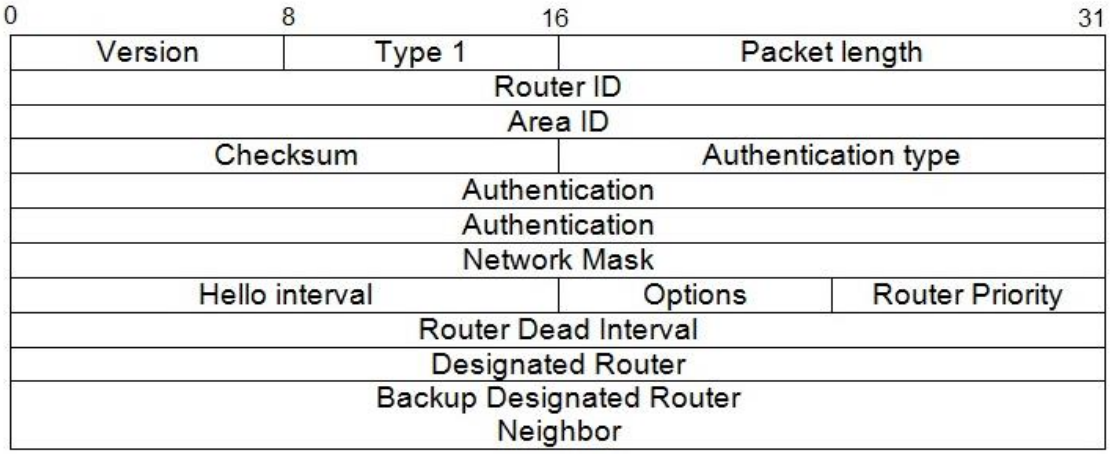


Figura 29. Estructura del Paquete *Hello* de *OSPFV2* [55].

a) *Network Mask* (máscara de red): Es la máscara de red asociada a la interfaz.

b) *Hello interval*: Es el número de segundos entre los paquetes *Hello* enviados por el router.

c) *Router Priority*: Escoge el DR y el BDR de forma manual.

d) *DR (Designate Router)*: *Router* responsable de establecer las adyacencias entre todos los vecinos de una red de multiacceso.

e) *BDR (Backup Designate Router*): En caso de que falle el DR, el BDR tomará sus funciones.

f) *Neighbor*: Este se encuentra en el mismo enlace físico con el que se comparte información de *routing*.

2.2.4.2.2 Paquete de descripción de la base de datos (*Database Description) de OSPFV2*

Estos paquetes se intercambian cuando una adyacencia se está iniciando, estos paquetes son de tipo 2. En la figura 30 se muestra el paquete de la base de datos.

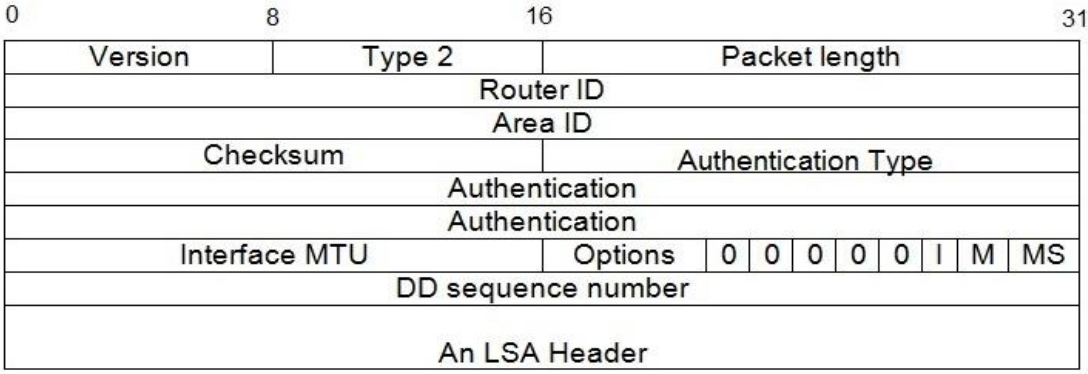


Figura 30. Paquete de descripción de la base de datos de *OSPFV2* [55].

a) *Interface* MTU: Es el tamaño sin fragmentar en octetos del paquete IP más grande que se puede enviar por la interface del originador.

b) *Options*: Describe las capacidades soportadas por el *router*.

c) *Bit I*: Este es el bit de inicialización. Cuando se pone I = 1, esto indica que este paquete es el primero de la secuencia de los paquetes DD, los demás paquetes tienen I = 0.

d) *Bit M*: Este es el bit de más. Cuando se pone M = 1, indica que después de este paquete continúan otros más paquetes. El último paquete tiene M = 0.

e) *Bit M/S:* Si se pone 1 esto indica que el *router* transmisor es el dominante, si es igual a 0 el transmisor es el esclavo.

f) DD *secuence number*: Número de secuencia del paquete DD, este es el número que identifica al paquete DD.

g) *An LSA Header*: Esta es una lista de cabeceras de mensajes de información del estado del enlace (*LSA header*).

2.2.4.2.3 *Link State Request* o Petición del estado enlace de *OSPFV2*

Cuando los paquetes de base de datos sean recibidos, el *router* receptor registrará los datos de las listas de encabezados de los mensajes de información del estado enlace. Esté paquete es de tipo 3 y su formato se muestra en la figura 31.

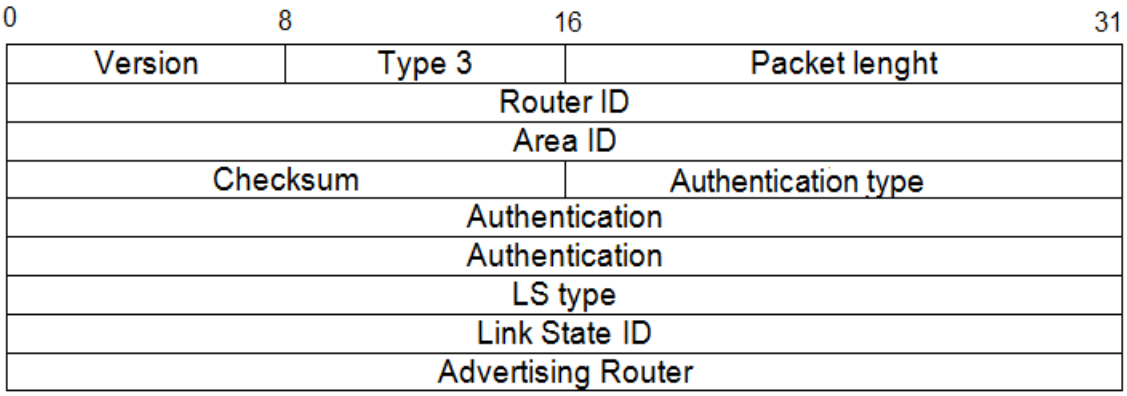


Figura 31. Paquete del estado enlace de *OSPFV2* [55].

a) *LS type* (Tipo del estado enlace): Este número identifica once tipos de LSA

b) *Link State ID* (Identificativo del estado enlace): Campo dependiente del tipo de cabecera del mensaje de información de estado enlace.

c) *Advertising Router* (*Router* Informador): Identifica al *router* originador del mensaje de información del estado de enlace (LSA).

2.2.4.2.4 *Link State Update* o Actualización del estado enlace de OSPFV2

Difunde los mensajes completos de información del estado enlace y envía estos mensajes en respuesta a los paquetes de petición del estado enlace. Este es un paquete de tipo 4. En la figura 32 se muestra la estructura del paquete estado de enlace.

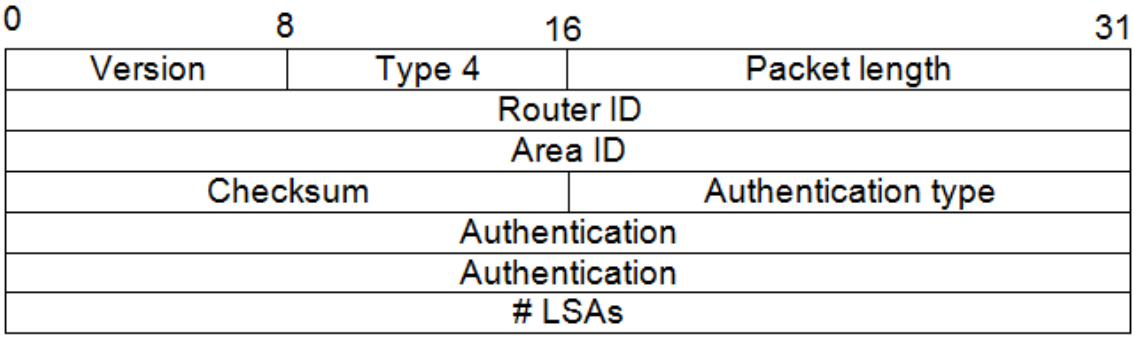


Figura 32. Paquete de actualización del estado enlace de OSPFV2 [55].

a) # *LSAs*: Número *LSA*, este indica la cantidad de *LSA* incluidos en este paquete.

b) *LSA*: Estos son los mensajes de información del estado enlace, cada uno de los paquetes de actualización pueden llevar tantos LSA hasta el tamaño máximo del paquete permitido por el enlace.

2.2.4.2.5 *Link State Acknowledgements* o Ack del estado enlace de *OSPFV2*

Este paquete es de tipo 5. Estos paquetes son empleados para confirmar el paquete de la actualización del estado enlace. En la figura 33 se muestra la estructura del paquete *Ack* del estado enlace.

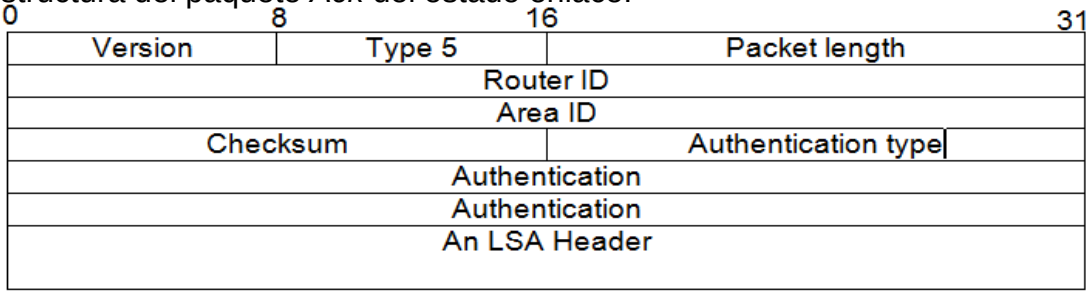


Figura 33. Paquete ACK del estado enlace de *OSPFV2* [55].

a) *LSA Header* (encabezado de mensaje de información de estado enlace): Esto es una lista de encabezados de mensajes de información del estado enlace (LSA) confirmadas.

2.2.4.3 Paquetes LSA

Los LSA (Link State Advertisements), son los paquetes que contienen toda la información relacionada a costos y rutas, así como, transportan la información de que interfaz está en estado de *up* o *down*. Estos paquetes llevan esta información a todos los *routers* de la red.

Existen cinco tipos distintos de mensajes LSA [56]:

a) *LSA Type 1 (Router LSA*)

El LSA tipo 1 es aquel que transporta la ID de cada *router* dentro de un área y los enlaces que lo conectan. Esto lo hace a través de crear un LSA tipo 1 para sí

mismo, es decir, crea un LSA para un *router* y luego lo transmite a los demás *routers* de la misma red.

b) *LSA Type 2 (Network LSA)*

Los LSA tipo 2 transportan las redes que tienen conectadas a alguna interfaz de un *router*, ayudando a configurar la topología de la red. Cabe aclarar que los LSA tipo 1 y tipo 2 solo trabajaban dentro de una misma área.

c) LSA Type 3 (Network summary)

Los routers ABR generan LSA tipo 3 para un área, después las envían a otras áreas, permitiendo que todas las áreas se conozcan.

d) LSA Type 4 (ASBR Summary)

Los LSA tipo 4 son transmitido por un *router* ABR a un *router* ASBR. El LSA tipo 4 sólo contiene métricas.

e) LSA Type 5 (External LSA)

Los LSA Tipo 5 son creados y enviados por los ASBR. Trasportan las rutas que pertenecen a otros protocolos de enrutamiento.

2.2.4.4 *Routers* OSPF

Existen diferentes tipos de *routers* que se utilizan en OSPF [57]:

a) DR. La función del *router DR*, es la de crear una adyacencia con los *routers* vecinos. Para elegir un *router DR*, se hace a través de los paquetes *Hello* a partir de las direcciones IP, es decir, la dirección IP más alta o mediante comandos.
b) BDR. El *router* BDR (*Backup Designated Router*), es aquel que se usa como DR, en caso de que éste falle. OSPF permite que el *router* principal, el DR, sea redundado con el BDR, de esta manera aseguramos que aunque falle el DR, el protocolo siga funcionando correctamente.

c) IR. El *router* IR (*Internal Router*), es aquel que tiene todas sus interfaces en una misma área, los *router* internos utilizan los mismos *routers* como datos ya que ejecutan el algoritmo SPF (*Shortest Path First*).
d) BR. Los *routers* BR (*Backbone Routers),* son los *routers* llamados *backbone* y son aquellos que se encuentran dentro del área 0.
e) ABR. Los *router* ABR (*Area Border Routers*), son los *routers* que cuentan con enlaces en diferentes áreas, cuenta con una base de datos específica para cada área.
f) ASBR. Los *routers* ASBR (*Automous System Boundary Routers*), son los *routers* frontera del Sistema Autónomo, cuentan al menos con una interfaz conectada a un sistema autónomo distinto.

2.2.4.5 Métrica OSPF

Para OSPF la métrica es el coste [58].

La decisión de qué camino tomar se basa en la métrica. En OSPF no está definido el coste, no se define en ningún RFC que describa al protocolo OSPF y es el propio fabricante que se encarga de establecer la métrica o coste. Por ejemplo Cisco utiliza como métrica la siguiente ecuación:

$$Coste = \frac{10^8\ bps}{ancho\ de\ banda} \quad \text{.....Ecuación 1}$$

2.2.4.6 Pasos que sigue un *router OSPF* hasta completar la tabla de enrutamiento

1.- Desactivado (*DOWN*): En el estado *down*, no se ha realizado ningún intercambio de información entre *routers.*

2.- Inicialización (*INIT*): Los *routers* envían paquetes *Hello*, para conocer que *routers* están conectados y a través de que interfaz. Una vez que recibe su primer paquete *Hello* comienza el estado de inicialización.

3.- Bidireccional (*TWO-WAY*): Comienza una comunicación bidireccional entre *routers* vecinos a través de paquetes *Hello*, empiezan a formar una pequeña tabla de enrutamiento entre los *routers* que conforman el área. En este estado todavía no se logra una adyacencia.

4.- Inicio de Intercambio (*EXSTART)*: Los dos *routers* vecinos envían paquetes *Hello* para conocer quién es el *router* maestro y quien es el esclavo. Se envían paquetes del tipo 2 DBD (*Database Description Packet*) para intercambiar bases de datos. Cuando los *routers* establecen quien es el *router* maestro y esclavo inicia el estado de Intercambio y comienzan a enviar información de encaminamiento.

5.- Intercambio (*EXCHANGE*): En el estado de intercambio, los *routers* describen sus bases de datos de estado de enlace entre ellos. Los *routers* comparan la información que van conociendo con la que ya tenían en su base de datos de estado de enlace. Si alguno de los *routers* recibe información acerca de alguna conexión nueva o algún cambio a la topología de la red, éste envía una solicitud de actualización completa a su vecino.

6.- Cargando (*LOADING*): Una vez que se conocen las bases de datos entre vecinos y éste estado requiera información más precisa, esto se logra a través del envío de paquetes del tipo 3 LSR. Cuando un *router* recibe un LSR este contesta enviando un paquete del tipo 4 LSU (*Link State Update*).

7.- Adyacencia completa (*FULL*): Cuando el estado anterior se ha completado los *routers* se vuelven completamente adyacentes, creando finalmente una base de datos [59].

2.2.5 OSPFV3 para IPV6

OSPFV3 se utiliza para IPV6 y es el protocolo que se utilizó en nuestra emulación de GNS3, la sección anterior de OSPFV2 sólo se dio como una breve información.

El protocolo OSPFV3 definido por el RFC 5340, OSPFV3 igual como OSPFV2 es un protocolo de estado de enlace, los nodos comparten toda la información sobre

los enlaces que les interconectan, todos los nodos almacenan la información sobre todos los enlaces de la red.

Los mecanismos fundamentales de OSPF (inundación, elección de enrutador designado (DR), soporte de área, primera ruta más corta, los cálculos de SPF, etc.) permanecen sin cambios. Sin embargo, algunos cambios fueron necesarios para poder emplear mejor el aumento al tamaño de dirección en IPV6. Estos cambios llevaron a incrementar la versión del protocolo OSPF de la versión 2 a la versión 3.

IPV6 utiliza el término "enlace" para indicar, una facilidad de comunicación o medio a través del cual los nodos pueden comunicarse en la capa de enlace. Se pueden asignar múltiples subredes IPV6 a un solo enlace, y dos nodos pueden hablar directamente sobre un solo enlace, incluso sino comparten una subred IPV6 en común.

Por este motivo, OSPF para IPV6 se ejecuta por enlace en lugar de las subredes por IP que utiliza OSPFV2. Del mismo modo, una interfaz OSPF ahora se conecta a un enlace en su lugar de una subred IP [60].

Este cambio afecta a la recepción de paquetes de protocolo OSPF, al contenido de los paquetes de *Hello* y el contenido de los LSA de red.

En detalle, los cambios en el formato del paquete OSPF consisten en lo siguiente [60]:

a) El número de versión de OSPF se ha incrementado de 2 a 3.
b) El campo *Options* en paquetes de *Hello* y *description* de la base de datos. Los paquetes se han ampliado a 24 bits.
c) Los campos *Authentication* y *AuType* se han eliminado del encabezado del paquete OSPF.
d) El paquete *Hello*, ahora no contiene información de dirección, por el contrario, ahora incluye una ID de interfaz que el enrutador de origen ha asignado para identificar de forma única (entre sus propias interfaces) su interfaz al enlace. Esta ID de interfaz será utilizado como el ID de estado de enlace de la red LSA, si el enrutador se convierte en el enrutador designado (DR) en el enlace.

2.2.5.1 OSPFV3 tipos de LSA (*link-state advertisement*)

En OSPFV3 se disponen de los siguientes tipos de LSA. Algunos de ellos son parecidos a los que se utilizan en OSPFV2 para IPV4 [60].

1. *LSA Tipo 1 Router-LSA*
2. *LSA Tipo 2 Network-LSA*
3. *LSA Tipo 3 Inter-Area-Prefix-LSA*
4. *LSA Tipo 4 Inter-Area-Router-LSA*
5. *LSA Tipo 5 AS-External-LSA*
6. *LSA Tipo 6 Deprecated (may be reassigned)*
7. *LSA Tipo 7 NSSA-LSA*
8. *LSA Tipo 8 Link-LSA*
9. *LSA Tipo 9 Intra-Area-Prefix-LSA*

a) *LSA-1 Router-LSA*

Cada uno de los enrutadores genera su propio LSA, este LSA indica los enrutadores vecinos que detecta el enrutador, y el coste para llegar a cada uno de ellos. En OSPFV2 también existe este tipo de LSA, pero se indican las direcciones de los vecinos en cada una de las interfaces, en IPV6 es necesario utilizar los LSA 8 y 9 para determinar las direcciones asignadas en cada una de las interfaces.

b) *LSA-2 Network-LSA*

OSPFV2 igual que OSPFV3 trabajan de forma especial en redes de múltiples accesos, tanto del tipo *broadcast* como *Ethernet*.

c) *LSA-3 Inter-Area-Prefix-LSA*

Este LSA se utiliza en entornos de múltiples áreas, se utiliza para indicar los prefijos de red que existen en cada una de las áreas. Estas entradas las genera el enrutador fronterizo del área ABR (*Area Border Router*).

En OSPFV2 también existía este LSA, pero recibe el nombre de *Summary-LSA.*

d) *LSA-4 Inter-Area-Router-LSA*

Esta LSA también se utiliza en entornos con múltiples áreas, se utiliza cuando se redistribuyen redes fuera del área 0. En OSPFV2 también existen pero reciben el nombre *ASBR-Summary-LSA.*

e) *LSA-5 AS-External-LSA*

Estos LSA se utilizan para redistribuir rutas de otros protocolos en OSPF. Por ejemplo la redistribución de una ruta por defecto.

f) *LSA Tipo 7 NSSA-LSA*

Se utilizan como los LSA tipo 5 en áreas NSSA (*Not-So-Stubby Area*), en que no se permiten los LSA tipo 5.

g) *LSA Tipo 8 Link-LSA*

Este tipo de LSA sólo se publica en el enlace, se genera uno en cada una de las interfaces donde está activo OSPF. Indica el o los prefijos de red declarados en esa interfaz y la dirección link-local que el nodo tiene declarada en la interfaz. En el resto de LSAs sólo se indican los identificadores de las interfaces, o el *router-id* del vecino. Con la información de este LSA se conoce la dirección IPV6 de cada una de las interfaces.

h) *LSA Tipo 9 Intra-Area-Prefix-LSA*

Se propagan dentro de cada una de las áreas, y los genera cada uno de los nodos para indicar los prefijos de red directamente conectados.

2.2.5.2 La estructura de la tabla de enrutamiento

La tabla de enrutamiento utilizada por OSPF para IPV4 se describió anteriormente. Para IPV6, hay entradas de la tabla de enrutamiento análogas. Estas son entradas de la tabla de enrutamiento para los prefijos de direcciones IPV6 y también para AS (Sistemas Autónomos) enrutadores de límite. Las últimas entradas de la tabla de enrutamiento sólo se utilizan para mantener resultados intermedios durante el proceso de compilación de la tabla de enrutamiento [60].

Además, para mantener los resultados intermedios durante el cálculo de la ruta más corta para cada área, hay una tabla de enrutamiento separada para cada área que contiene la siguiente entrada:

a) Una entrada para cada enrutador en el área. Los enrutadores son identificados por su ID de enrutador OSPF. Estas entradas de la tabla de enrutamiento contienen el conjunto de rutas más cortas a través de un área dada a un enrutador dado, que a su vez permite el cálculo de rutas a los prefijos IPV6 anunciados por ese enrutador en el *intra-area-prefix-LSAs*. Si el enrutador es también un enrutador de borde de área, estas entradas también se utilizan para calcular rutas para los prefijos de dirección entre áreas. Si además, el *router* es el otro punto final de un enlace virtual, la entrada de la tabla de enrutamiento describe el costo y la viabilidad del enlace virtual.

b) Una entrada para cada enlace de tránsito en el área. Los enlaces de tránsito tienen redes LSAs asociadas. Tanto el enlace de tránsito como la red LSA se identifican mediante una combinación de la ID de interfaz del enrutador designado en el enlace y la Identificación del enrutador OSPF del enrutador designado. Estas entradas de la tabla de enrutamiento permiten el cálculo posterior de las rutas a los prefijos IP anunciados para el enlace de tránsito *intra-area-prefix-LSA*.

Los campos en la tabla de enrutamiento OSPF de IPV4 (OSPFV2) siguen siendo válidos para IPV6: *optional capabilities, path type, cost, type 2 cost, link state origin.*

2.2.5.3 El encabezado de paquetes OSPFV3

Cada paquete OSPF comienza con un encabezado estándar de 16 bytes. El encabezado OSPF contiene toda la información necesaria para determinar si el paquete debe ser aceptado para su posterior procesamiento. Esta se describe en la figura 34 [60].

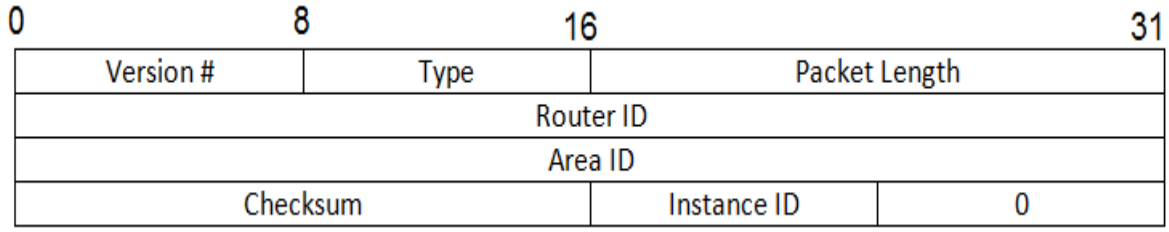


Figura 34. Encabezado del paquete OSPFV3.

Version. Especifica qué tipo de versión se utiliza.

Type. Los tipos de paquetes OSPF son los siguientes.

1. *Hello*
2. *Database Description*
3. *Link State Request*
4. *Link State Update*
5. *Link State Acknowledgment*

a) *Packet Length*. La longitud del paquete de protocolo OSPF en *bytes*.

b) *Router ID*. La ID del enrutador de la fuente del paquete.

c) *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. La mayoría viaja a sólo un salto.

d) *Checksum*. OSPF utiliza el cálculo de suma de comprobación estándar para aplicaciones IPV6.

e) *Instace ID*. Permite que varias instancias de OSPF se ejecuten a través de un sólo enlace. Los paquetes recibidos cuya ID de instancia no es igual a la ID de instancia de la interfaz receptora se descartan.

f) Cero: Estos campos están reservados. Deben establecerse en 0 al enviar paquetes de protocolo y deben ignorarse al recibir paquetes de protocolo.

2.2.5.4 Tipos de paquetes OSPFV3

2.2.5.4.1 Paquete *Hello* para OPSFV3

Estos paquetes se intercambian entre los dispositivos directamente conectados.

Se utilizan para comprobar la conectividad bidireccional entre ellos. Los nodos tienen definido un tiempo de *Hello* y un tiempo *Dead*, el tiempo de *Hello* indica la periodicidad con que se envían los saludos, y el tiempo de *Dead* es el periodo máximo de ausencia de paquetes de *Hello*, normalmente es cuatro veces el tiempo de *Hello*. Si un vecino no envía un saludo pasado el tiempo de *Dead*, se le considera caído. La estructura del paquete *Hello* se muestra en la figura 35 [60].

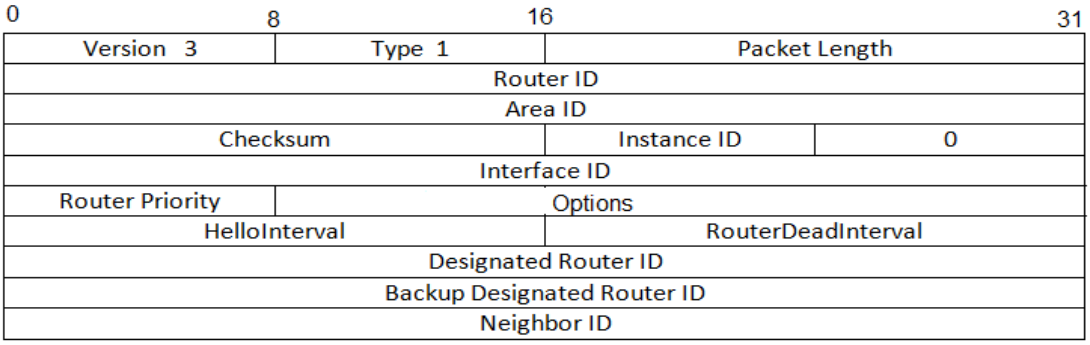


Figura 35. Estructura del paquete *Hello* para OSPFV3.

a) *Interface ID*. Número de 32 *bits* que identifica de forma exclusiva esta interfaz entre la colección de interfaces de este enrutador.

b) *Router Priority*. Utilizado en la elección del enrutador designado (de respaldo). Si se establece en 0, el enrutador no será elegible para convertirse en el enrutador designado.

c) *Options*. Las capacidades opcionales soportadas por el enrutador.

d) *HelloInterval*. El número de segundos entre los paquetes de saludo de este enrutador.

e) *RouterDeadInterval*. El número de segundos antes de declarar un enrutador fuera de línea.

f) *Designated Router ID*. El enrutador designado se identifica por su ID del enrutador. Se establece en 0 sino hay un enrutador designado.

g) *Backup Designated Router ID*. El enrutador designado de respaldo es identificado por su ID de enrutador IP. Se establece en 0 sino hay un enrutador designado de respaldo.

h) *Neighbor ID*. Las ID de cada enrutador en la red con otros vecinos.

2.2.5.4.2 DBD (*Database Description*) para OSPFV3

Este paquete se intercambia entre los enrutadores una vez que se ha establecido adyacencia. Contienen un resumen de la base de datos de topología. La estructura del paquete *database description* se muestra en la figura 36 [60].

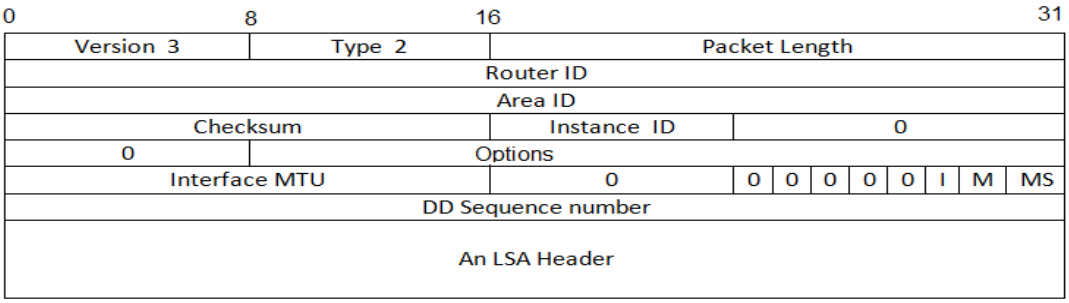


Figura 36. Estructura del paquete DBD (*database description)* para OSPFV3.

a) *Options*. Las capacidades opcionales soportadas por el enrutador.

b) *Interface MTU*. El tamaño en *bytes* del datagrama IPV6 más grande que se puede enviar.

c) *I-bit*. El bit inicial. Cuando se establece en 1, este paquete es el primero en la secuencia de paquetes de descripción de la base de datos.

d) *M-bit*. El bit más. Cuando se establece en 1, indica que hay más paquetes en la descripción de la base de datos

e) *MS-bit*. El bit maestro / esclavo. Cuando se establece en 1, indica que el *router* es el maestro durante el proceso de intercambio de base de datos. De otra manera, el *router* es el esclavo.

f) *DD Sequence number*. Se utiliza para secuenciar la colección de paquetes de descripción de base de datos. El valor inicial (indicado por el bit de inicio que

se establece) debe ser único. El número de secuencia DD luego aumenta hasta que se completa la base de datos para los *routers* maestro y esclavo.

2.2.5.4.3 LSR (*Link-state Request)* para OSPFV3

Los paquetes de LSR se utilizan para solicitar un LSA (*link-state advertisement*) concreto al nodo vecino. Tras intercambiar los DBD los dispositivos piden a sus vecinos con un LSR aquellos registros que no tienen o que están más actualizados en el dispositivo vecino. En la figura 37 se muestra la estructura del mensaje *Link State Request* [60].

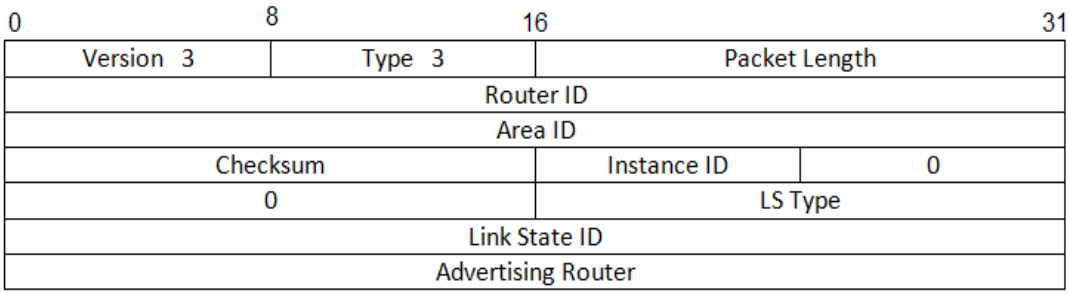


Figura 37. Estructura de paquete LSR (*Link-state Request*) para OSPFV3.

a) *LS Type* (Tipo del estado enlace). Este número identifica el tipo de LSA.

b) *Link State ID*. Campo dependiente del tipo de encabezado del mensaje de información del estado enlace.

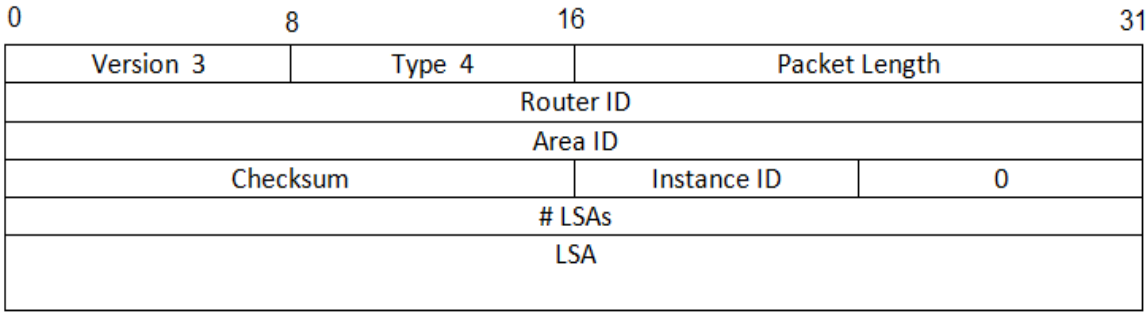
c) Advertising Router. Identifica al *router* originador del mensaje de información del estado de enlace (LSA).

Cada LSA solicitado se especifica por su *LS Type, Link State ID y Advertising Router.* Estos identifican de forma única la LSA sin especificar su instancia. Los paquetes de solicitud de estado de enlace se envían para hacer solicitudes de la instancia más reciente de los LSA especificados.

2.2.5.4.4 LSU (*Link-State Update*) para OSPFV3

Los paquetes de actualización de estado de enlace son paquetes de tipo 4 de OSPF. Estos paquetes implementan la inundación de LSAs. Se pueden incluir varios LSA en un sólo paquete.

Los paquetes de actualización de estado de enlace son multidifusión en redes físicas que soportan *multicast* / *broadcast*. En la figura 38 se muestra la estructura de *Link State Update* [60].



En la figura 38. Estructura de *Link State Update* para OSPFV3.

a) *LSAs*. El número de LSA incluidos en esta actualización.

b) *LSA*: Estos son los mensajes de información del estado enlace, cada uno de los paquetes de actualización pueden llevar tantos LSA hasta el tamaño máximo del paquete permitido por el enlace.

2.2.5.4.5 LSAck (*Link-State Acknowledgment)* para OSPFV3

En OSPF no se reenvía continuamente la información, así que cuando se envía un LSU se utilizan los LSAck para asegurar su correcta recepción. En la figura 39 se muestra la estructura de *Link State Update* [60].

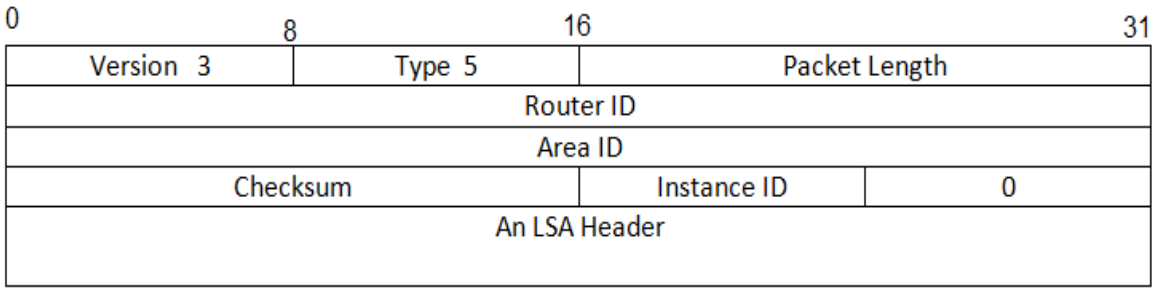


Figura 39. Estructura de *LSAck* para OSPFV3.

a) *LSA Header* (encabezados de mensajes de información de estado enlace). Esto es una lista de encabezados de mensajes de información del estado enlace (LSA) confirmadas.

2.3 PROTOCOLO DE GESTIÓN DE RED SNMP

En la emulación de la red GEANT también se trabajó en la gestión de la red, para esto se utilizó el protocolo SNMPV3 para IPV6 y en los siguientes párrafos se dará una breve explicación.

En 1990 surgió el estándar SNMP (*Simple Network Managenement Protocol*), definido en el RFC 1157. Este protocolo intercambia información de administración entre dispositivos de red, existen tres versiones de este protocolo SNMPV1, SNMPV2 especificado en el RFC 1215 y SNMPV3 especificado por el RFC 3410. El último posee avances significativos en aspectos de seguridad.

SNMP trabaja en la capa de aplicación del modelo OSI, este protocolo recibe solicitudes por el puerto 161 vía UDP. El administrador manda solicitudes de cualquier puerto de origen libre para el puerto 161 en el agente (host, routers, hubs). La contestación del agente será enviada al puerto de origen. Se reciben notificaciones tipo Trap e InformRequests por el puerto 162 [61].

SNMPV2 utiliza cinco mensajes:

1. Get Request

2. Get Next Request

3. Set Request

4. Get Response

5. Trap

En SNMPV3 se agregaron dos mensajes más, los cuales fueron *GetBulkRequest e InformRequest*.

2.3.1 Arquitectura de SNMPV2

La arquitectura SNMP consta de los siguientes componentes:

a) Estaciones de gestión.
b) Agentes de gestión.

c) MIB (Base de Información de administración).
d) SMI.

2.3.1.1 Estación de gestión para SNMPV2

La estación de gestión es un dispositivo autónomo, esta funciona como la interfaz entre el administrador de red y el sistema a gestionar. La estación administradora deberá tener [61]:

a) Grupo de aplicaciones de administración para analizar datos.
b) A través de una interfaz el administrador controla la red.
c) Interpretar las peticiones de administración de red dentro de los elementos de control de la red.
d) Base de información sacada de las MIBs.

2.3.1.2 Agente de gestión para SNMPV2

Un agente administrador es un elemento tal como: *host, routers y hubs*, estos pueden ser abastecidos con agentes SNMP para que puedan ser administrados a través de una estación administradora.

Este agente contesta a solicitudes de información y acciones desde la estación de gestión [61].

2.3.2 Mensajes de SNMPV2

Los mensajes SNMP tienen dos formas de trabajar [61]:

a) *Polling*: Transmite consultas remotas de forma activa o baja demanda realizando una operación síncrona de consulta.
b) *Traps*: Es generado por el agente para reportar ciertas condiciones y cambios de estado a un proceso de administración.

La estación de gestión y el agente pueden intercambian información en forma de mensajes SNMP.

Las tramas de SNMP tienen el siguiente formato, que se observa en la figura 40.

Versión	Comunidad	SNMP PDU

Figura 40. Formato de la trama SNMPV2 [61].

a) Versión: Este indica la versión del protocolo.

b) Comunidad: Autentifica el mensaje SNMP.

c) SNMP PDU (*Protocol Data Unit*): Depende del tipo de operación a realizar.

Los mensajes del protocolo SNMP emplean la estructura en el campo SNMP PDU, Figura 41.

Tipo	Identificador	Estado de error	Índice de error	Enlazado de variables.

Figura 41. Estructura en el campo SNMP PDU para SNMPV2 [61].

a) Tipo: Este nos dice el tipo de PDU (El código de comando).

b) Identificador: Es el número que identifica los mensajes intercambiados entre el agente de gestión y la estación de gestión.

c) Estado de error: Los posibles errores son 0 que significa sin error, 1 significa demasiado grande.

d) Índice de error: Cuando es el estado de error es distinto de cero

e) Enlazado de variables: Éstos son los campos de SNMP, en estos es donde va la información de los parámetros gestionados con sus valores, codificados por medio del estándar SMI (*Structure of Management Information).*

La comunicación entre un agente SNMP y un mensaje NMS (*Network Management System*) se realiza a través de cinco mensajes, los cuales proporcionan la información necesaria para la monitorización de la red [61]:

1. *Get Request*: Es una solicitud del administrador al Agente para que mande los valores contenidos en el MIB (Base de datos). El agente contesta enviando un

mensaje de éxito o fracaso de la petición. Si la petición fue correcta, el resultado del mensaje abarcara el valor del objeto solicitado.

2. *Get Next Request*: El mensaje *Get Next Request* se usa después de utilizar el mensaje *GetRequest* y repite la operación con el siguiente elemento de la tabla.
3. *Set Request*: Este mensaje es utilizado por el NMS (Sistemas Administradores de Red) y solicita a un agente cambiar los valores de objetos.

Para los mensajes *GetRequest, GetNextRequest* o *SetRequest* se tiene, que la trama PDU tiene el siguiente formato, figura 42 [61].

PDU type	Request ID	0	0	Variable Bindings

Figura 42. Formato de la trama PDU de SNMPV2.

Dónde:

a) *PDU type*: Indica el tipo de PDU.

b) *Request ID*: Esta identifica las diferentes peticiones y agrega a cada una de ellas un único identificador.

c) *Variable Bindings*: Es una lista de variables y de sus valores. En varios casos como el del mensaje *GetRequest*, el valor es *NULL*. Con respecto a las *Traps*, éstas dan información adicional relativa a la *Trap*.

4. *Get Response*. Este mensaje se usa por el agente para responder un mensaje *GetRequest, GetNextRequest* o *SetRequest.*

Para *Get Response* se tiene el siguiente formato, figura 43.

PDU type	Request ID	Error-status	Error-index	Variable Bindings

Figura 43. Estructura del mensaje *Get Response* para SNMPV2.

Dónde:

a) *Error-status*: Indica si ha sucedido una excepción durante el proceso de una petición, los valores posibles son*: (0) NoError, (1) tooBig, (2) noSuchName, (3) badValue, (4) readOnly, (5) genErr.*

b) *Error-index:* Si el campo *Error-status* es diferente de 0, este da información adicional señalando la variable que causó la excepción.

5. Trap: Este es generado por el agente reporta las condiciones y cambios de estado a un proceso de administración.

El formato PDU de *trap* es diferente, aunque también consta de encabezados PDU, el encabezado contiene la versión del protocolo utilizada y la clave que se usa para autentificar el mensaje. El PDU *trap* tiene el siguiente formato, ver figura 44.

Tipo	Enterprise	Dirección del agente	Tipo genérico de trap	Tipo específico de trap	Timestamp	Enlazado de variables

Figura 44. Formato PDU de *Trap* para SNMPV2.

Tipo genérico de *trap*:

a) *Cold start* (0): Muestra que el agente ha sido iniciado o reiniciado.

b) *Warm start* (1): El agente ha cambiado su configuración.

c) *Link down* (2): Esto nos indica cuando una interfaz de comunicación está inactiva.

d) *Link up* (3): Muestra cuando una interfaz de comunicación se encuentra activa o en servicio.

e) *Authentication failure* (4): Muestra cuando el agente ha recibido una petición de un NMS no autorizado.

f) *EGP neighbour loss* (5): Este muestra que cuando los *routers* están utilizando el protocolo EGP un equipo vecino se encuentra fuera de servicio.

g) *Enterprise* (6): Todos los nuevos *traps* se localizan aquí.

h) *Enterprise*: Este Identifica el sub-sistema de gestión que se ha emitido por el *trap*.

i) Dirección del agente: Es la dirección IP del agente que ha arrojado el *trap*.

j) Tipo genérico de *trap*: Usado para *traps* privados.

k) Enlazado de variables: Proporciona información adicional sobre la causa del mensaje.

Existen dos mensajes más del protocolo SNMP, los cuales sólo son para la versión dos y versión tres del protocolo mencionado anteriormente.

1. *GetBulkRequest*: Este es utilizado por NMS que usa la versión 2 o 3 del protocolo SNMP, típicamente es requerido cuando existe una larga transmisión de datos, tal como la recuperación de largas tablas.

2. *Inform Request*: Lo utilizan la versión 2 y 3 y es utilizada para notificar información sobre objetos administrados.

2.3.3 Estructura de la información de gestión SMI

SMI (Estructura de Información de Gestión) está definida en el RFC 1155.

La SMI es un grupo de reglas que determina las características de los objetos de la red y cómo es que se obtienen los protocolos de gestión de información de ellos.

SMI es la gramática para escribir MIB de SNMP. Existen dos versiones de SMI, estas son SMIV1 y SMIV2 que corresponden a diferentes implementaciones del protocolo SNMP.

SMI determina la forma en que debe definirse y construirse una MIB. También proporciona técnicas de estandarización para:

a) Definir la estructura de un determinado MIB.

b) Codificación de los valores de los objetos.

c) Definir los objetos individuales, sintaxis y valor.

2.3.4 MIB (Base de datos de información de gestión)

Las MIB (*Management Information Base*) definido en el RFC 1155, es una colección de objetos que representa los dispositivos de la red. Cada agente SNMP debe ser capaz de juntar objetos MIB estándar, dichos objetos incluyen dirección de red, tipo de interfaz, contadores, etc.

MIB especifica 126 objetos conectados con los protocolo TCP/IP, todos los fabricantes que así lo deseen consiguen desarrollar extensiones del estándar MIB. Las MIBs privadas, algunas veces contienen objetos similares a los MIBs que ya están definidos [62].

En el sistema de gestión recae la carga de la gestión de todas las MIBs y de las extensiones privadas. El lenguaje de las MIBs está en el lenguaje OSI ASN 1.

En 1990 surgió una nueva versión de MIB, MIB II, que consta de 185 nuevos objetos de extensiones privadas.

Existen 8 grupos de objetos manejados por MIB que son [62]:

a) Sistema: Esté contiene la identidad del vendedor y el tiempo desde la última vez que se reinició el sistema de gestión.

b) Interfaces: Única o múltiples interfaces, local o remota.

c) ATT (*Address Translation Table*): Esta incluye la dirección de la red y las equivalencias con sus direcciones físicas.

d) IP (*Internet Protocol*): Esta conserva estadísticas sobre los datagramas IP que son recibidos.

e) ICMP (*Internet Communication Management Protocol*): Cuenta los errores y los mensajes recibidos ICMP.

f) TCP (*Transmission Control Protocol*): Simplifica la información acerca de las conexiones TCP.

g) UDP (*User Datagram Protocol*): Da el conteo de datagramas UDP enviados, recibidos y entregados.

h) EGP (*Exterior Gateway Protocol*): Recolecta información sobre el número de mensajes EGP recibidos y generados.

Para que los dispositivos SNMP puedan ser monitoreados, el gestor SNMP debe compilar el archivo MIB para cada uno de los equipos en la red. Un OID (*Object Identifier)* es una secuencia de enteros positivo en la que cada entero pertenece a un nodo particular en el árbol. Este dato permite identificar un objeto de gestión y relaciona su lugar en la jerarquía de los objetos [63].

En la figura 45, se observa el árbol de identificador de objetos para objetos de internet. Cuyo camino sombreado, es aquel que se usó en el emulador GEANT 1.3.6.1.2.1 para llegar a administrar el *router* con SNMPV3.

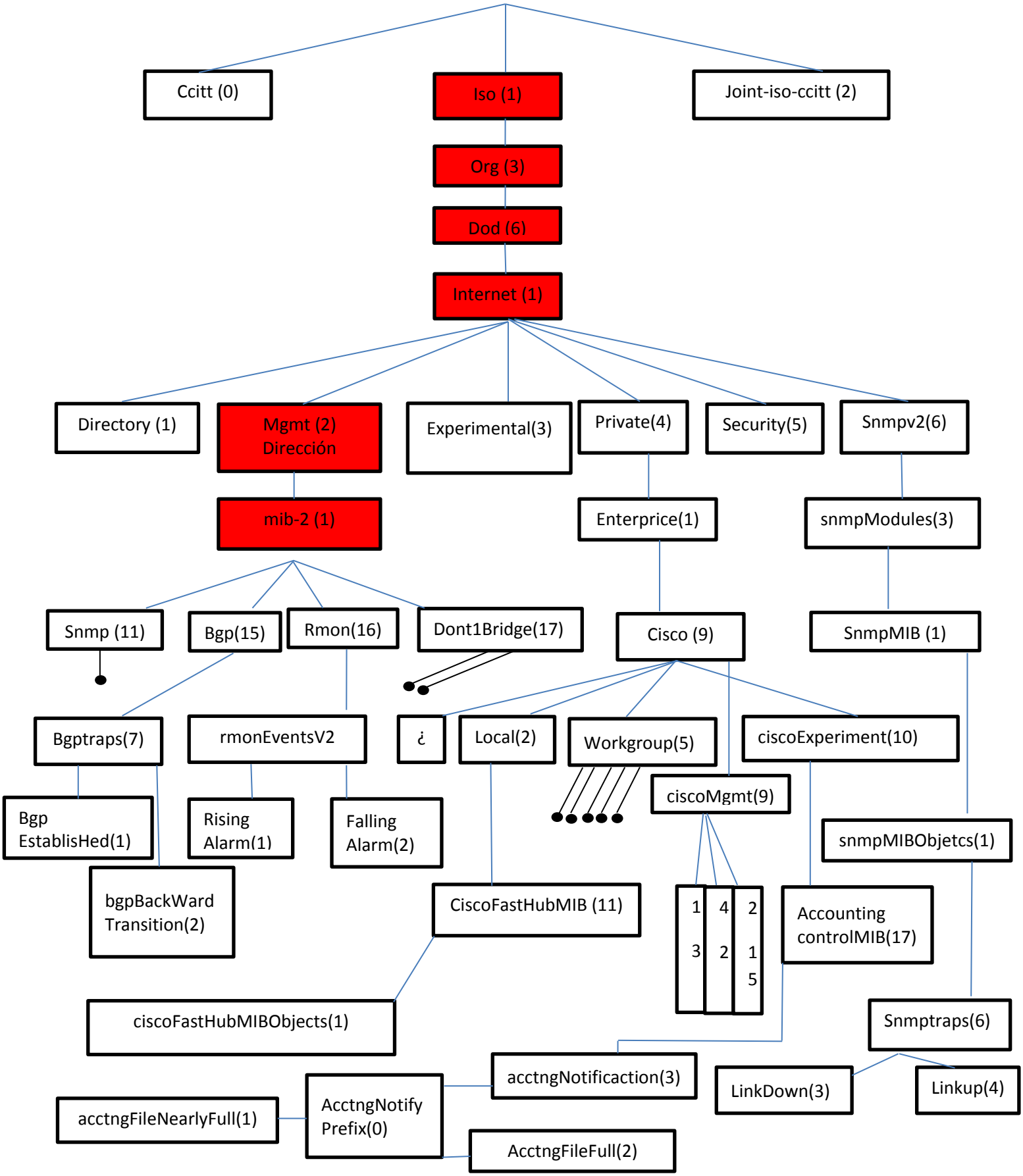


Figura 45. Árbol de internet de acuerdo con su OID [63].

2.3.5 SNMPV3 para IPV6

SNMP se utiliza para la administración y gestión de una red. Utiliza la información que se almacena en las MIB (Base de datos de Información de Gestión) para su funcionamiento. Las MIB definen la estructura de cómo se almacena la información sobre un dispositivo para que se informe a través de SNMP. En otras palabras, los estándares definen las estructuras de las MIB y depende de los proveedores implementar la estructura de la MIB y poder proporcionar la información mediante el acceso a los OID de la MIB.

De otra manera, los proveedores pueden implementar las estructuras o los OID que eligieron para el monitoreo y la administración de la red. Dada la necesidad de la industria de interoperabilidad con las herramientas SNMP, casi todos los proveedores optaron por implementar MIB estandarizadas.

Dado que SNMP es un protocolo de capa de aplicación, no se ve afectado por el protocolo de capa de red. IPV4 e IPV6 podrían utilizarse sin afectar la comunicación SNMP. Por ejemplo, si quisiéramos sondear el mismo OID, usando IPV6 usaríamos una sintaxis similar y la información provista por el dispositivo sería la misma

Como se mencionó anteriormente el protocolo SNMP puede funcionar con IPV4 e IPV6, sin embargo, existe una serie de complicaciones que IPV6 representa para la funcionalidad SNMP, el cual no está tan relacionado con el protocolo sino con los objetos MIB que llevan las direcciones de red. Existían problemas en la falta de soporte para IPV6 en objetos MIB, ya que algunas MIB sólo admiten IPV4.

Otro problema que presenta en referencia de IPV4 a IPV6, es el tamaño de una dirección ahora se ha cuadruplicado (la dirección IPV6 es de 128 bytes donde IPV4 era 32), lo que aumenta el tamaño de los datos que se transportarán en la carga útil de los paquetes UDP de SNMP que contienen información de direcciones IPV6.

Es por esto que algunas MIB que se usan para IPV4 tuvieron que actualizarse, y otras a su vez quedaron obsoletas ante la llegada de IPV6.

SNMPV3 mejora la seguridad, es decir, fortalece la autentificación y privacidad, con una mayor modularidad y la posibilidad de configuración remota. SNMPV3 está definida por el RFC 3414, se basa en un modelo de seguridad conocido como USM (*User-Based Security Model*), el cual, proporciona los servicios de autentificación y privacidad en SNMPV3. USM se encarga de que el mensaje fuera trasmitido por la entidad indicada y que en su trayecto no fuera alterado, retardado o repetido hasta llegar a su destino. Para lograr una autentificación exitosa, el gestor y el agente que desean intercambiar información deben compartir la misma clave de autentificación secreta, esta clave se configura previamente y no es conocida por la MIB.

Los protocolos de autentificación que utiliza SNMPV3 son dos: HMAC-MD5-96 y HMAC-SHA-96. Por otro lado, la privacidad que usa USM posibilita al gestor y al agente encriptar mensajes para evitar que sean analizados por intrusos. El algoritmo de encriptación que utiliza SNMPV3, es el CBC (*Cipher Block Chaining*) de DES (*Data Encryption Standard*) o DES-56. Igualmente como en la autentificación, tanto el agente como gestor deben de compartir la misma contraseña configurada previamente [64-67].

2.3.5.1 Motor SNMPV3

El motor de SNMP se compone de 4 elementos:

a) Despachador (*The Dispatcher*). El despachador, tiene la función de enviar y recibir mensajes determinando la versión de protocolo.

b) Subsistema de procesamiento de mensajes (*Message Processing Subsystem*). El subsistema de procesamiento de mensajes, tiene como función enviar y extraer información de los mensajes recibidos.

c) Subsistema de seguridad (*The Security Subsystem*). El subsistema de seguridad, proporciona los servicios de autentificación y privacidad, para SNMPV3 usa el USM.

d) Subsistema de control de acceso (*The Access Control Subsytem*). El subsistema de control de acceso, tiene como función el controlar el acceso a los objetos del árbol MIB [64, 66, 68].

2.3.5.2 Autentificación SNMPV3

La autentificación entre las entidades SNMP, se realiza mediante algoritmos de tipo MD5 (*Message-Digest 5*) y SHA (*Secure Hash Algorithm*), estos algoritmos se utilizan en SNMPV3 con el uso de claves entre las entidades al momento de configurar el protocolo de gestión. Las claves para el algoritmo MD5 contienen 16 octetos y para el algoritmo SHA contienen 20 octetos [64, 69].

2.3.5.3 Privacidad SNMPV3

El módulo de privacidad SNMPV3 proporciona seguridad contra la difusión de los mensajes SNMP y se realiza mediante el cifrado con criptografía DES (*Data Encryption Standard*) de 56 bits, 3DES DE 168 bits y AES (*Advanced Encryption Standard*) [66, 68, 70].

2.3.5.4 Mensajes de SNMPV3

Como se define en el RFC 2272, el formato de mensaje para SNMPV3, contiene un encabezado y un PDU [67].

En la figura 46 se muestra el formato de mensaje de SNMPV3.

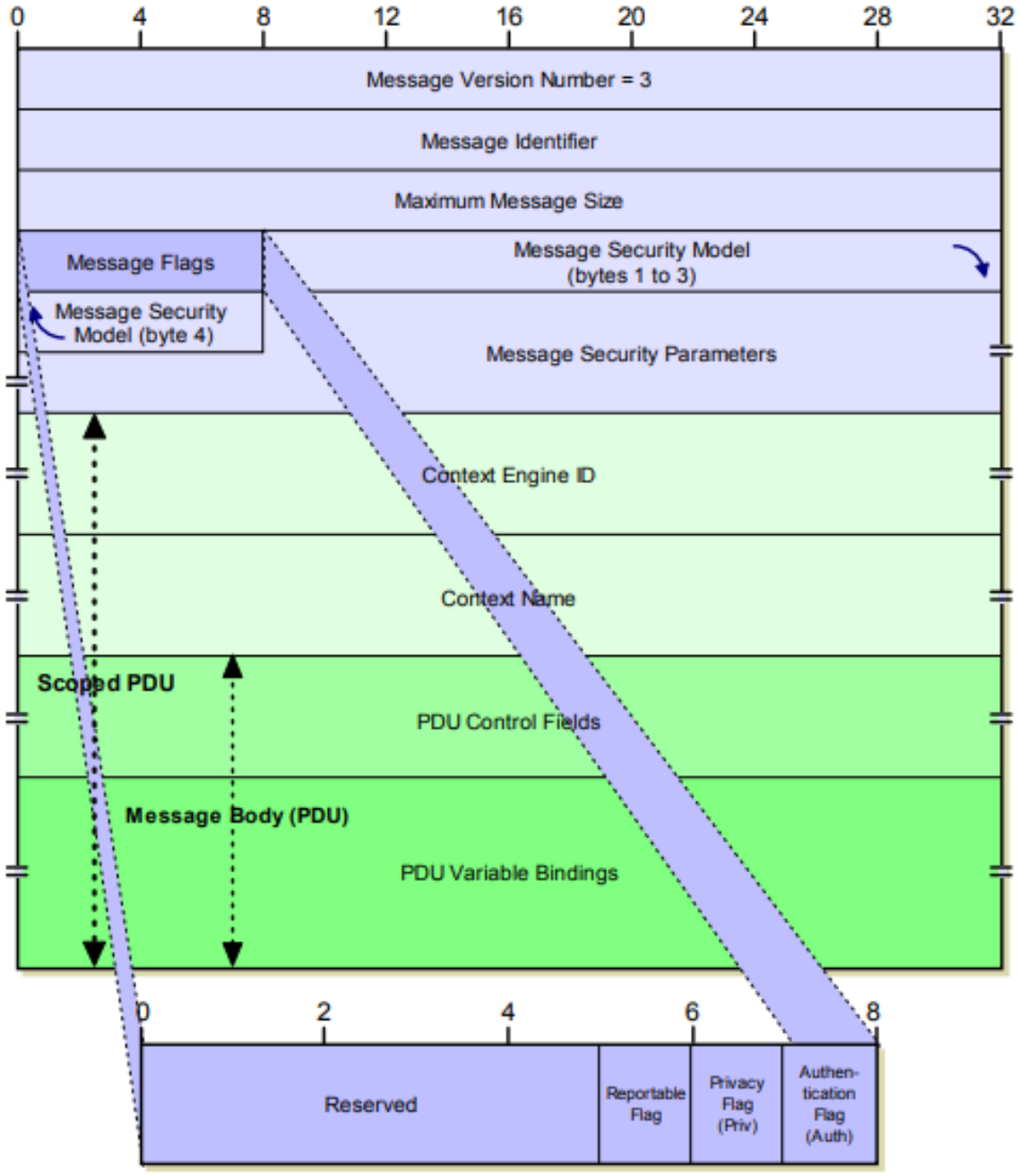


Figura 46. Formato de mensaje de SNMPV3 [66].

a) *Message Version Number*: Este campo identifica el mensaje como una versión 3 del protocolo SNMP.

b) *Message ID*: Es un identificador único usado entre dos entidades para coordinar mensajes de solicitud y respuesta.

c) *Maximum Message Size*: Es el tamaño máximo del mensaje que el remitente puede aceptar.

d) *Message Flags*: Contiene dos campos de bits:

- *reportableFlag*: Determina si un informe PDU debe ser enviado.
- *authFlag, privFlag y msgFlags*: Indican el nivel de seguridad que se aplicó al mensaje antes de ser enviado.

e) *Message Security Model*: Es un identificador el cual indica que modelo de seguridad fue usado por el remitente para preparar el mensaje.

f) *Message Security Parameters*: Es usado para la comunicación entre los módulos del modelo de seguridad.

g) *Context Engine ID*: Identifica de manera única una entidad SNMP, para mensajes provenientes, determina a que aplicación el *scopedPDU* va a ser enviado.

h) *Context Name:* Identifica de manera única un contexto particular dentro de la entidad SNMP.

i) *Scoped PDU*: Contiene información para identificar un contexto único y una PDU.

j) *Reserved*: Espacio reservado para un uso futuro.

k) *Reportable Flag*: Indicador para cuando se establece en 1 una entidad para que envié un informe del tipo de PDU.

l) *Privacy Flag*: Es el indicador de privacidad y cuando se establece un valor en 1 indica que el cifrado se utilizó para proteger la privacidad del mensaje.

m) *Authentication Flag*: Es el indicador de autentificación y se establece en 1, cuando indica que se utilizó la autentificación para proteger la autenticidad del mensaje [60, 64].

Capítulo 3

Metodología para la emulación de la Red GEANT

3.1 GNS3

GNS3 (*Graphic Network Simulation*, Simulación Gráfica de Redes) es un software que permite emular, probar y solucionar problemas de redes virtuales y reales. GNS3 admite muchos dispositivos de múltiples proveedores de red incluidos conmutadores virtuales Cisco, y muchos otros. Así como sistemas operativos *Windows* y *Linux.*

Ya que conocemos el funcionamiento de las redes avanzadas, aplicaremos los protocolos de enrutamiento OSPF y de gestión SNMP, para realizar la emulación de la red GEANT, utilizando el software GNS3, con base en la topología más actual de la red europea GEANT de diciembre 2018, como se muestra en la figura 17.

Para realizar todos estos procesos y pruebas, se utilizó un equipo portátil de la marca DELL, con un procesador Intel Core (TM) i5-3340M CPU @ 2.70 GHz, con memoria RAM de 12 GB y un disco duro de 320 GB.

3.1.2 Emulación de la red avanzada GEANT

Para la instalación de GNS3 sólo se tiene que ingresar a la página https://www.gns3.com/, posteriormente crear una cuenta y descargar el software para el sistema operativo, en este caso se usó sistema de gestión Windows 7.

Una vez instalado el software, se comenzó a realizar la configuración de la red GEANT en el emulador GNS3 aplicando el protocolo de enrutamiento OSPF y el de gestión SNMP. En GNS3 es necesario instalar IOS de los *routers,* ya que éstos no los tienen instalados en el software, por lo tanto, se tienen que descargar y posteriormente instalar en los *routers*. Pará agregar los IOS seleccionamos la pestaña "*edit*" del programa GNS3→ *Preferences Dynamips→IOS routers→New→New Image→Browse→* se elige la imagen IOS del *router,* para esta emulación se usó el IOS del *router* c7200-adventerprisek9-mz124-24.T5.image. Se selecciona y posteriormente *→next → next →* se configuran los *Slots* a utilizar como se muestra en las figuras 47 a la 51.

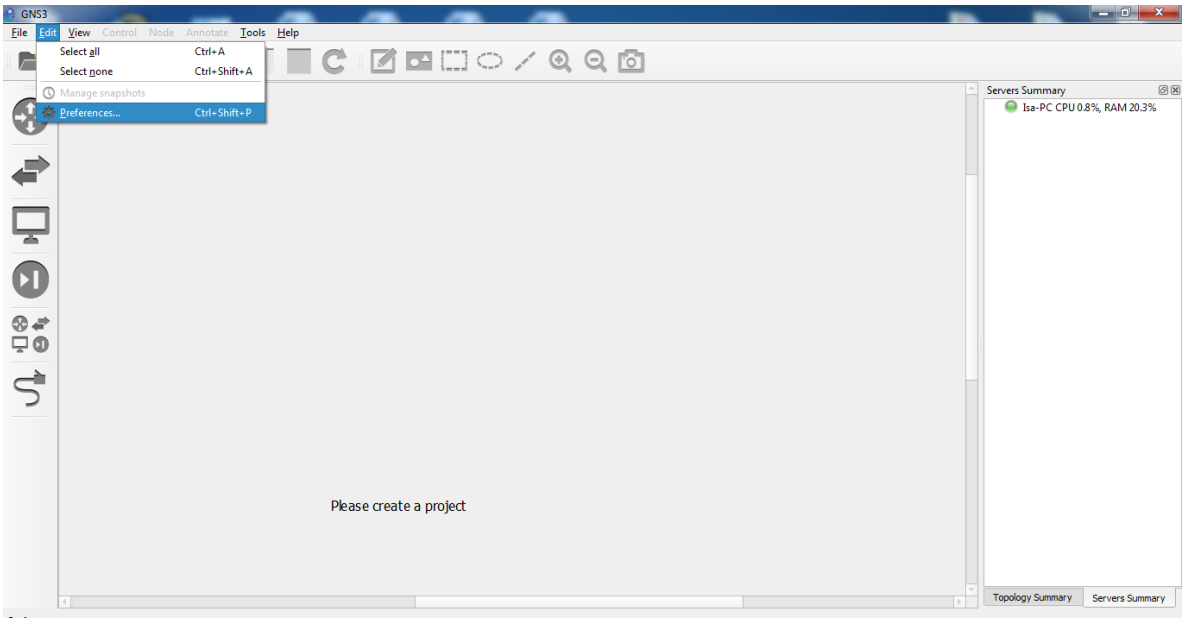


Figura 47. Preferencias de GNS3.

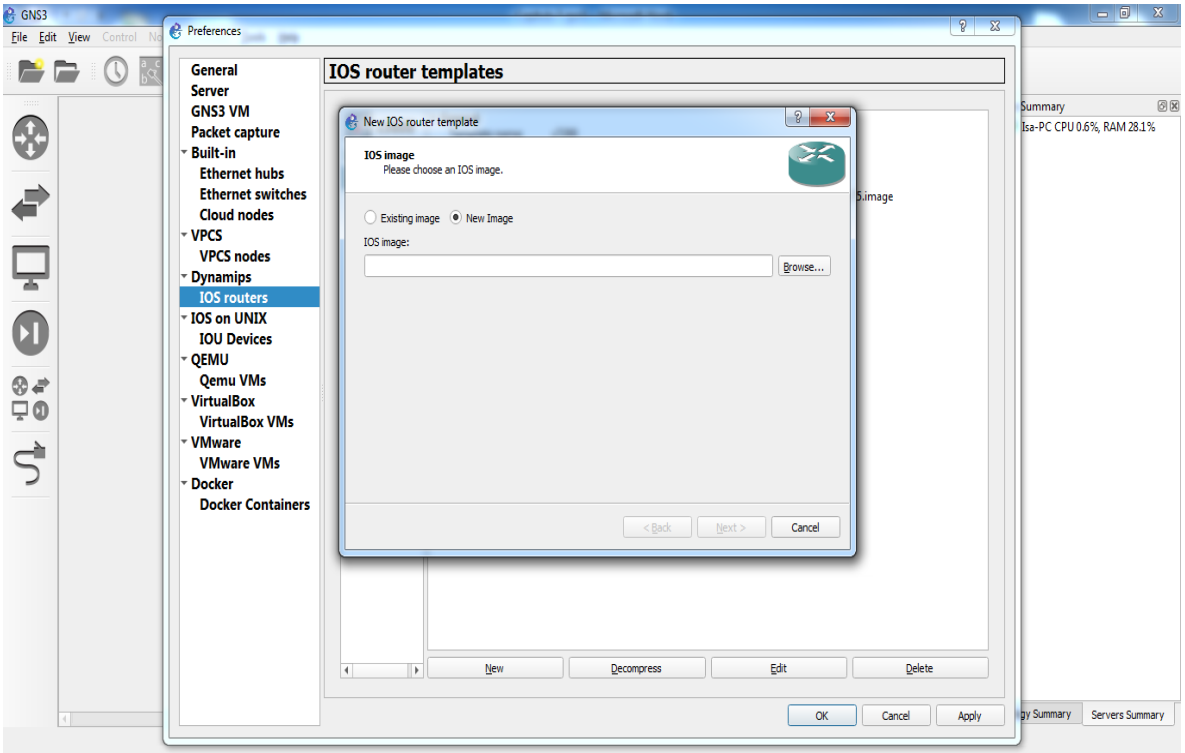


Figura 48. Selección de IOS del *router*.

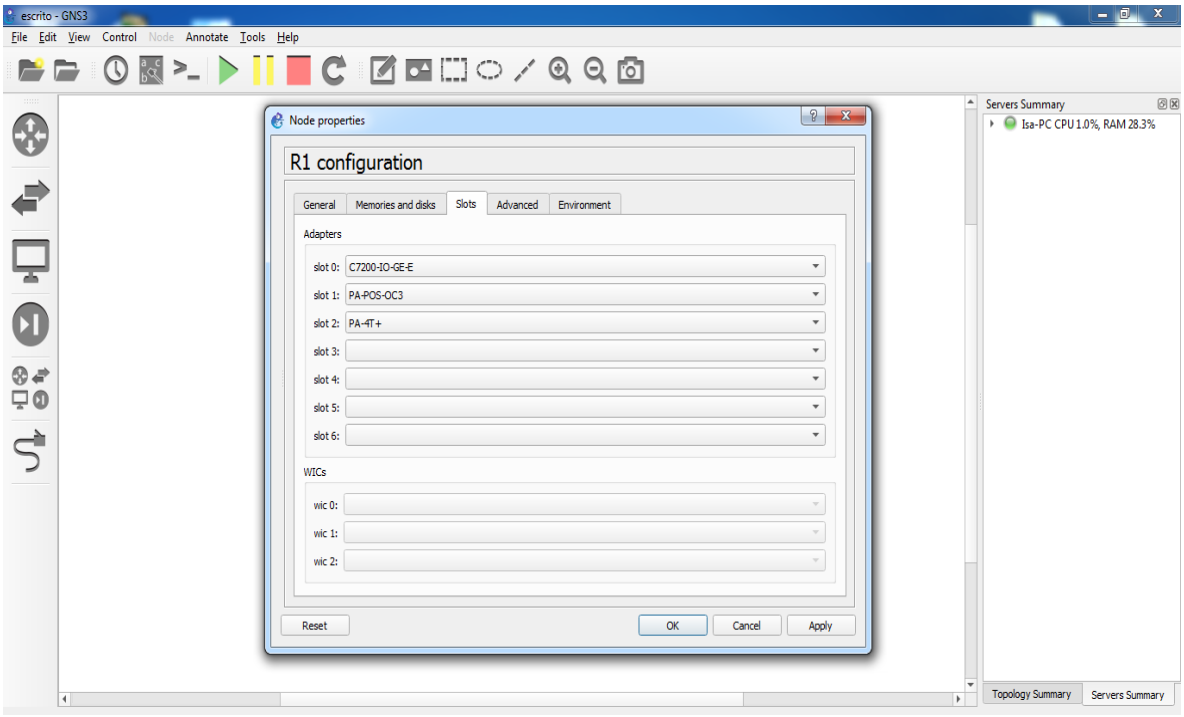


Figura 49. Configuración de interfaces del *router*.

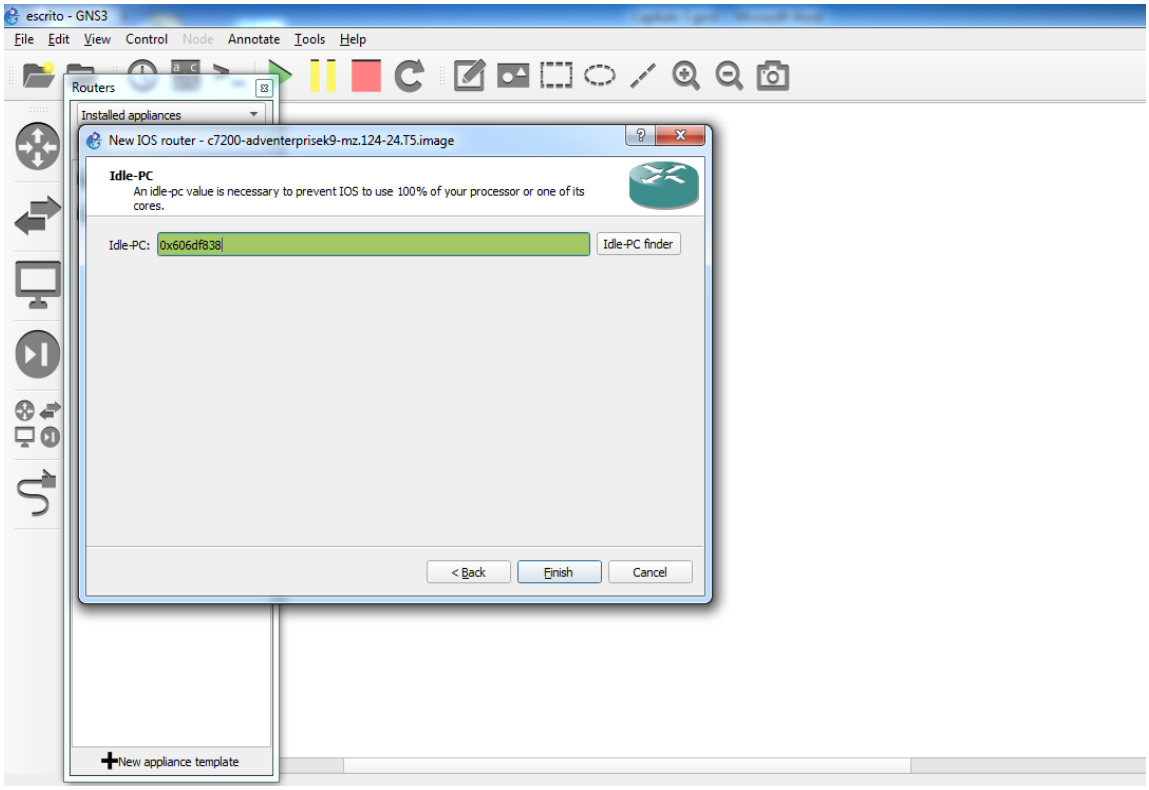


Figura 50. Idle-PC, depuración de la imagen del IOS.

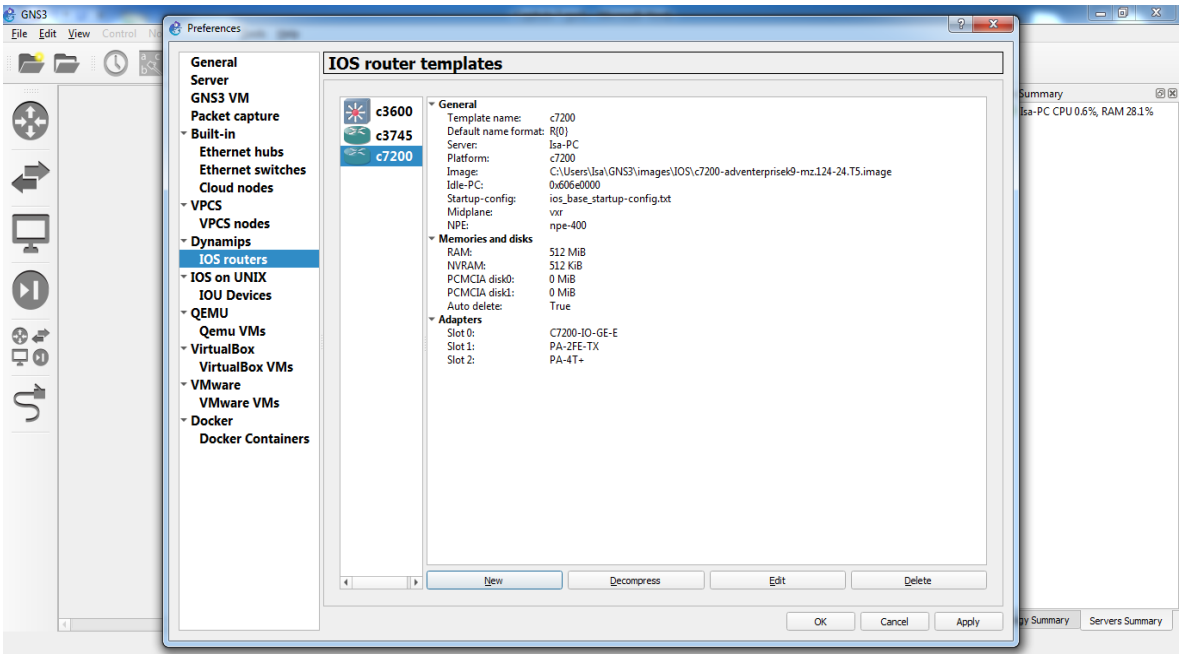


Figura 51. IOS cargando en GNS3.

3.2 Emulación de la Red GEANT

Una vez que se tiene configurado el *IOS* de los *routers*, se comenzó la conexión de la topología GEANT, utilizando interfaces POS (*Packet Over- Sonet*). Dado que GNS3 tiene como limitante el número de POS a 6 interfaces por *router,* esto ocasionó un problema, debido a que la topología de GEANT a emular requiere más de 6 interfaces en los *routers,* que son: Alemania 2 (14 interfaces conectadas), Reino Unido (9 interfaces conectadas), Austria (13 interfaces conectadas), Hungría (11 interfaces conectadas), como se observa en la figura 52.

Debido a este problema se tuvieron que utilizar tanto interfaces POS como interfaces *GigabitEthernet* y *FastEthernet,* para realizar la conexión total de interfaces en los *router* antes mencionados, dadas las limitantes de GNS3.

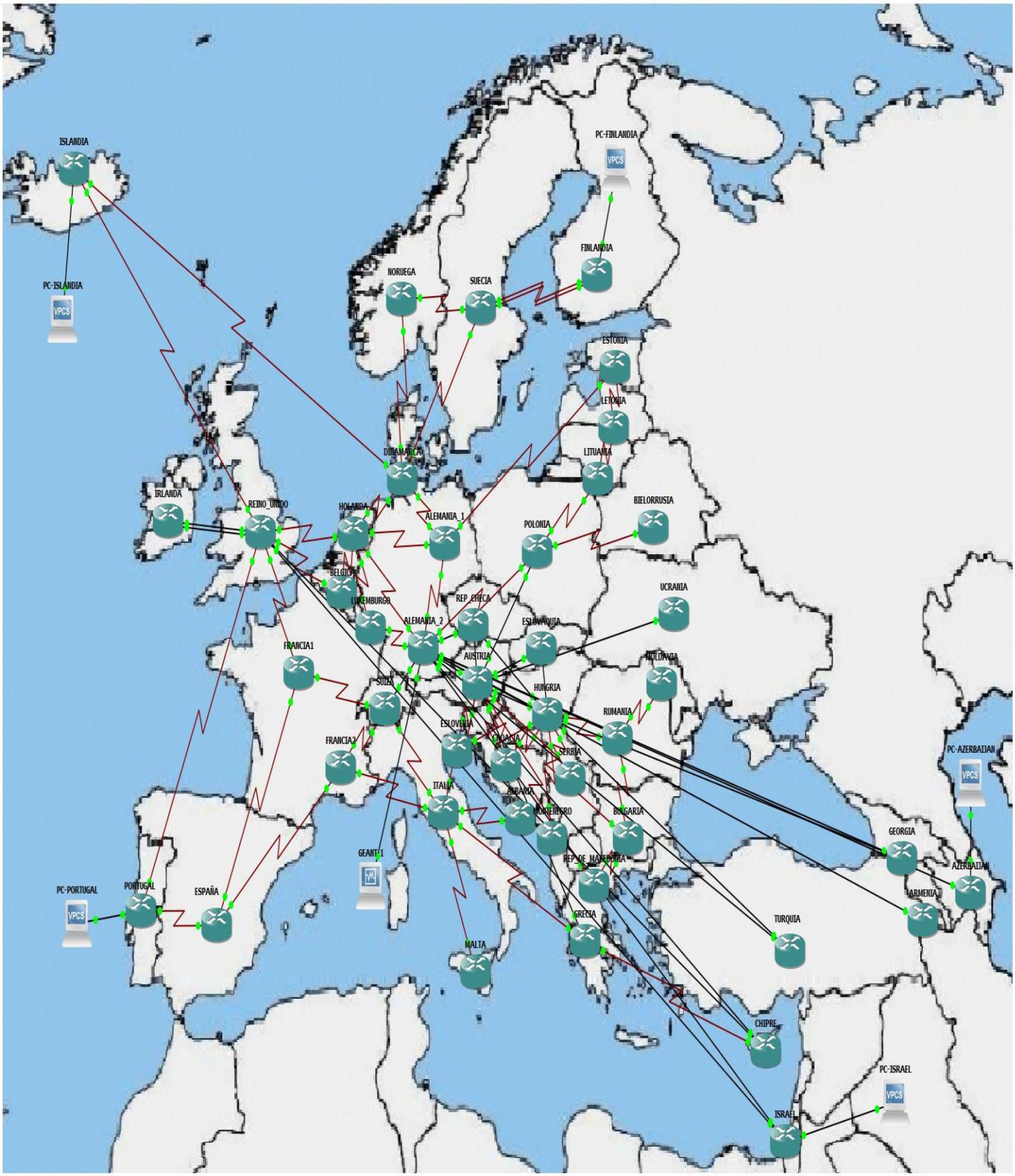


Figura 52. Topología de la red GEANT (2018) emulada en GNS3.

3.2.1 Interfaces de los *routers* de GEANT para la emulación

Como se muestra en la figura 52, se configuró cada uno de los *routers* y se conectó una VPCS (*Virtual Pc Simulation*) en los *routers* Israel, Azerbaiyán, Portugal, Islandia y Finlandia. VPCS es una simulación de una pc con código *Linux*, la cual se configura como host para poder comprobar conexión entre cada *router* de la red. Así como se configuró una máquina virtual mediante el programa *VirtualBox* en el *router* Alemania_2 para realizar la gestión de la red.

El *Router ID* corresponde a un identificador de 32 bits que tiene cada *router*, el cual se encuentra escrito en formato IPV4. Es necesario tener en cuenta que, aunque se escriba como una dirección IPV4, no corresponde a una, es sólo un valor que identifica un *router* con OSPF habilitado.

En la tabla 2 se muestran las conexiones de las interfaces de cada uno de los *routers* de la red GEANT. La tabla completa se puede consultar en el anexo 1.

País	Interface	Dirección IPV6	*Router ID*
Portugal a España	pos2/0 a pos2/0	2001:db8:1::/64	4.4.4.4
Portugal a Reino Unido	pos3/0 a pos2/0	2001:db8:2::/64	4.4.4.4
España a Portugal	pos2/0 a pos2/0	2001:db8:1::/64	3.3.3.3
España a Francia1	pos3/0 a pos2/0	2001:db8:3::/64	3.3.3.3
España a Francia2	pos4/0 a pos2/0	2001:db8:4::/64	3.3.3.3
Francia1 a Reino Unido	pos3/0 a pos3/0	2001:db8:5::/64	2.2.2.2
Francia1 a España	pos2/0 a pos3/0	2001:db8:3::/64	2.2.2.2
Reino Unido a Portugal	pos2/0 a pos3/0	2001:db8:2::/64	1.1.1.1

Tabla 2. Conexiones de la red GEANT (parte 1).

En la tabla 3, se muestran las conexiones de las VPCS en la red GEANT.

País	Interface	Dirección IPV6
VPCS-Islandia a *Router* Islandia	E0 a F1/1	2001:db8:76::/64
VPCS-Portugal a *Router* Portugal	E0 a F1/1	2001:db8:77::/64
VPCS- Finlandia a *Router* Finlandia	E0 a F1/1	2001:db8:78::/64
VPCS- Azerbaiyán a *Router* Azerbaiyán	E0 a F1/1	2001:db8:79::/64
VPCS-Israel a *Router* Israel	E0 a F1/1	2001:db8:80::/64

Tabla 3. Conexiones de las VPCS de la red GEANT.

3.2.1.2 Configuración de Interfaces en los *routers* de la red GEANT

El procedimiento de la configuración para activar las interfaces de los *routers* es el siguiente:

a) Encender el *router* y entrar en modo súper usuario con el comando "*enable*".
b) Entrar al menú configuración, con el siguiente comando "*configure terminal*"
c) Declaramos el comando "*ipv6 unicast-routing"* para habilitar el ruteo para IPV6.
d) Con el comando *"int"* se declaran las interfaces a configurar *(interface pos, interface Giga Ethernet, Fast Ethernet).*
e) Se declara la dirección de red y máscara de red con el comando "*ipv6 address".*
f) Se activa la interface.

g) Para guardar los cambios se usa el comando "*wr*", como se muestra en la figura 53.

```
DINAMARCA#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
DINAMARCA(config)#ipv6 uni
DINAMARCA(config)#ipv6 unicast-routing
DINAMARCA(config)#int pos2/0
DINAMARCA(config-if)#
DINAMARCA(config-if)#ipv6 add
DINAMARCA(config-if)#ipv6 address 2001:db8:49::3/64
DINAMARCA(config-if)#
DINAMARCA(config-if)#nop shu
DINAMARCA(config-if)#no shu
DINAMARCA(config-if)#no shutdown
DINAMARCA(config-if)#
DINAMARCA(config-if)#exit
DINAMARCA(config)#
```

Figura 53.Configuración de interface de red del *router* Alemania1.

Con el comando "*sh ipv6 int br*", nos cercioramos que estén activas y configuradas las interfaces. El comando anterior despliega la información de la interface, dirección IP de la red configurada y el estatus de cada interface, si está habilitada (*up*) o deshabilitada (*down*) como se observa en la figura 54.

```
ALEMANIA_1#sh ipv6 int br
Ethernet0/0                [administratively down/down]
    unassigned
GigabitEthernet0/0         [administratively down/down]
    unassigned
FastEthernet1/0            [administratively down/down]
    unassigned
FastEthernet1/1            [administratively down/down]
    unassigned
POS2/0                     [up/up]
    FE80::C825:22FF:FE00:6
    2001:DB8:48::3
POS3/0                     [up/up]
    FE80::C825:22FF:FE00:6
    2001:DB8:49::2
POS4/0                     [up/up]
    FE80::C825:22FF:FE00:6
    2001:DB8:51::3
POS5/0                     [up/up]
    FE80::C825:22FF:FE00:6
    2001:DB8:62::3
ALEMANIA_1#
```

Figura 54. Interfaces del *router* Alemania1.

3.3 Configuración de host

Para comprobar comunicación entre los *routers* se utilizó las VPCS que nos proporciona el emulador GNS3, como host para los *routers* Israel, Azerbaiyán, Portugal, Islandia y Finlandia. Las VPCS se configuran ingresando al modo consola como se muestra en la figura 55.

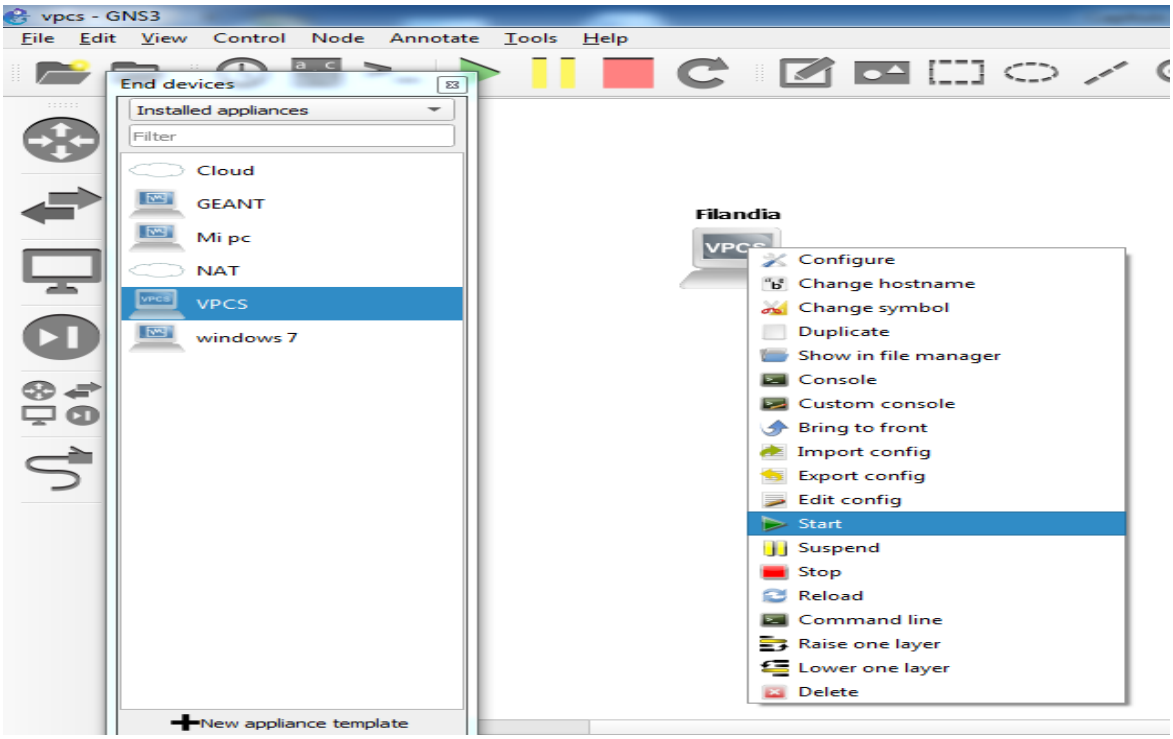


Figura 55. Encendido de VPCS.

Para ingresar la configuración de la red se usa el comando *"ip*" seguido de la dirección IPV6 a configurar, para guardar cambios se usa el comando "*save",* como se muestra en la figura 56.

Figura 56. Configuración de red en VPCS.

3.3.1 Configuración del protocolo OSPF de la red GEANT

Para configurar el protocolo OSPF en el emulador GNS3, se realiza vía CLI en cada uno de los *routers*, siguiendo los siguientes pasos:

a) Encender el *router* Dinamarca y entrar en modo súper usuario con el comando "*enable*".
b) Entrar al menú configuración con el siguiente comando "*configure terminal*"

c) Declarar el protocolo OSPF con el comando "*ipv6 router ospf 1"*

d) Declarar el *router* ID con el comando "*router-id"*

e) Con el comando *"int"* se declaran las interfaces a configurar *"Int pos2/0"*

f) Para crear una dirección link-local de IPV6 se usa el comando "*Ipv6 enable"*

g) Con IPV6, se tienen varias direcciones IPV6 configuradas en una interfaz. La instrucción *network* se eliminó en OSPFV3, usando ahora el comando *"Ipv6 ospf 1 area 0" .*Como se muestra en la figura 57.

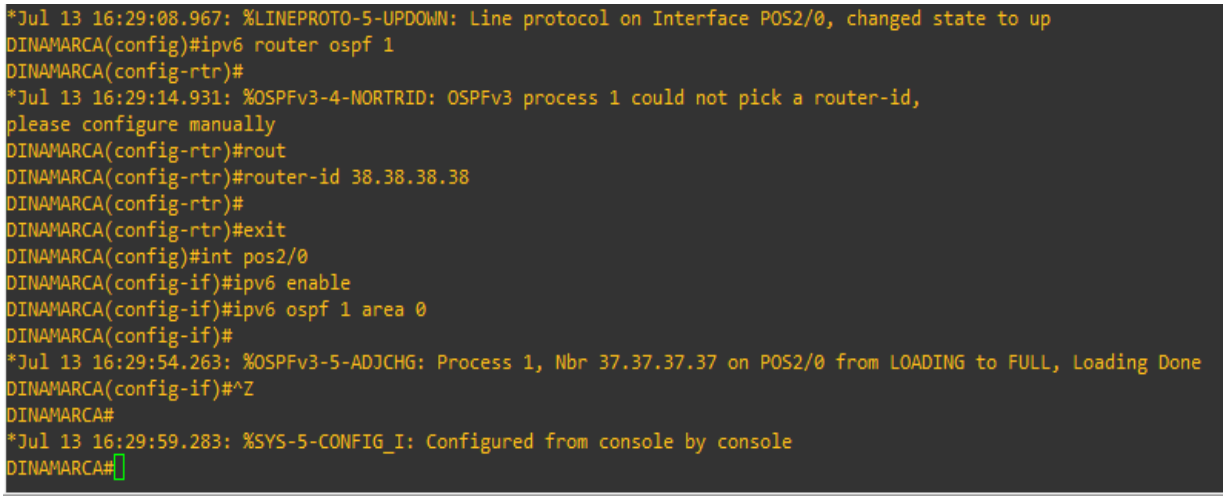


Figura 57. Configuración de OSPF en *router* Dinamarca.

Al configurar los demás *routers*, el protocolo de enrutamiento indica que se ha hecho una adyacencia con un *router*, indicando el número de proceso y el *router-id* con el que se hizo la adyacencia, como se observa en la figura 58.

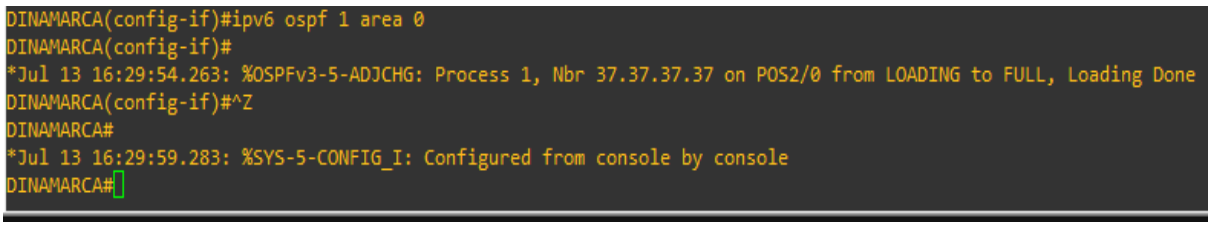


Figura 58. Adyacencia entre *routers* de Dinamarca a Alemania1.

3.3.1.2 Creación de una máquina virtual en GNS3

Se usó una máquina virtual para acercarnos más a la realidad, probar la conectividad y configurarla como una estación de gestión SNMP utilizando aplicaciones como *PowerSNMP free Manager* y *Wireshark*.

Para crear una máquina virtual se usó el software *Oracle VM VirtualBox.* Una vez instalado se siguieron los siguientes pasos:

a) Abrir el programa *VirtualBox.*

b) Abrir la opción nueva→seleccionar el tamaño de memoria → crear disco duro virtual → VDI (*VirtualBox Disk Image*) → Seleccionar el almacenamiento de disco duro → seleccionar *crear*.

c) Ya creada la máquina virtual, seleccionar iniciar → seleccionar la ruta de la imagen (*ISO*) del sistema operativo. El proceso de instalación se realiza tal y como se hace de forma normal en un equipo, como se observa en la figura 59.

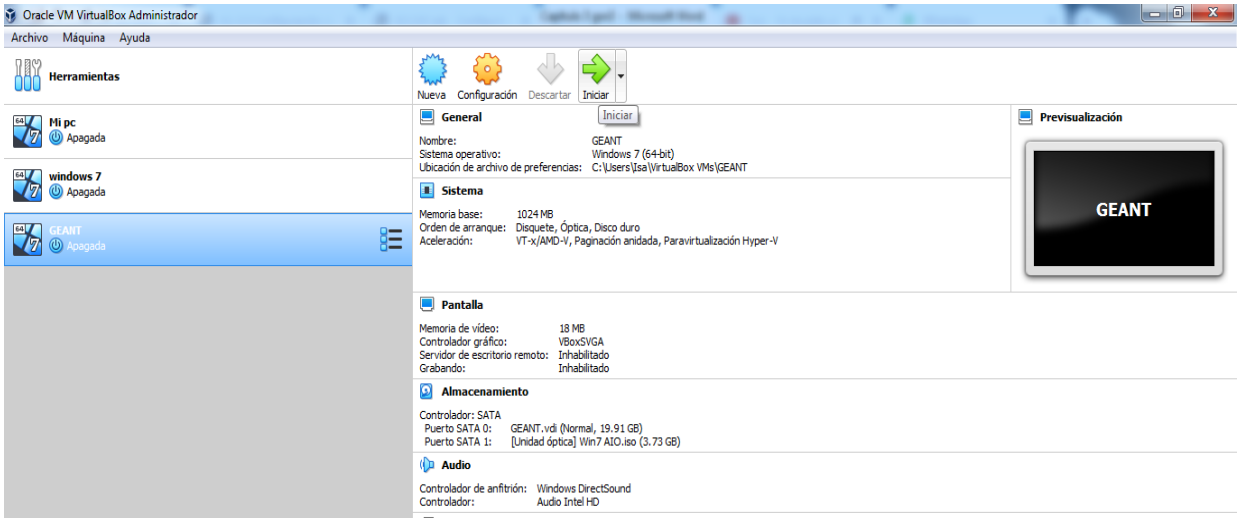


Figura 59. Creación de la máquina virtual llamada GEANT.

Una vez instalado nuestro sistema operativo se configuró la NIC de la máquina virtual, con los siguientes pasos:

a) Encender la máquina virtual.
b) Inicio → panel de control → Redes de Internet.
c) Seleccionar centros de redes y recursos compartidos → cambiar configuración del adaptador → *click* derecho en conexión de red de área local → propiedades → protocolo de internet versión 6 (IPV6).
d) Se usó la dirección IP, mascara de subred y puerta de enlace predeterminada que se muestra en la figura 60.

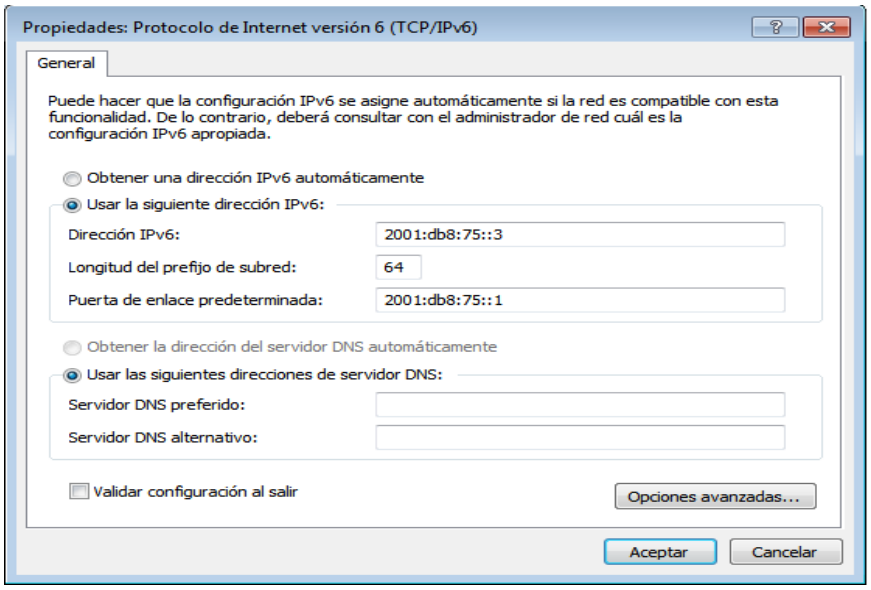


Figura 60. Configuración de la NIC de la máquina virtual GEANT.

El firewall de la máquina virtual se tuvo que desactivar, para evitar un problema de conectividad, como se ve en la figura 61.

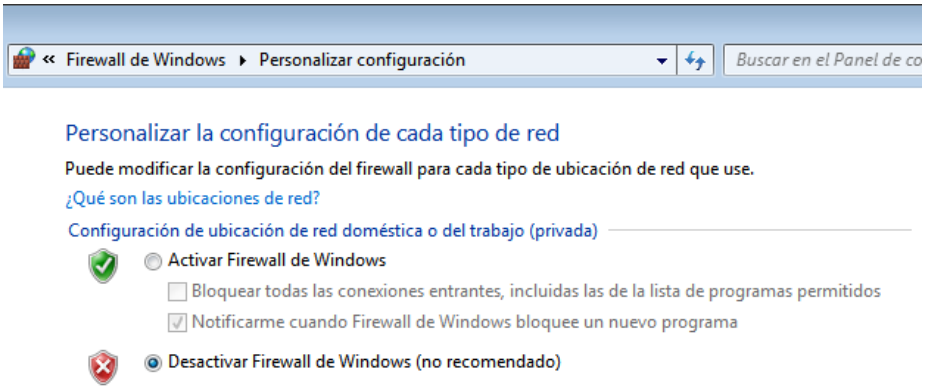


Figura 61. Firewall de máquina virtual deshabilitado.

3.4 Configuración de máquinas virtuales en GNS3

Para poder vincular una máquina virtual con GNS3 los pasos son los siguientes:

a) Abrir GNS3 seleccionar *edit* → *preferences* → *VirtualBox VMs* →*new* → Aparece el listado de máquinas virtuales que se tienen en *VirtualBox* → *finish*, como se muestra en las figuras 62 y 63.

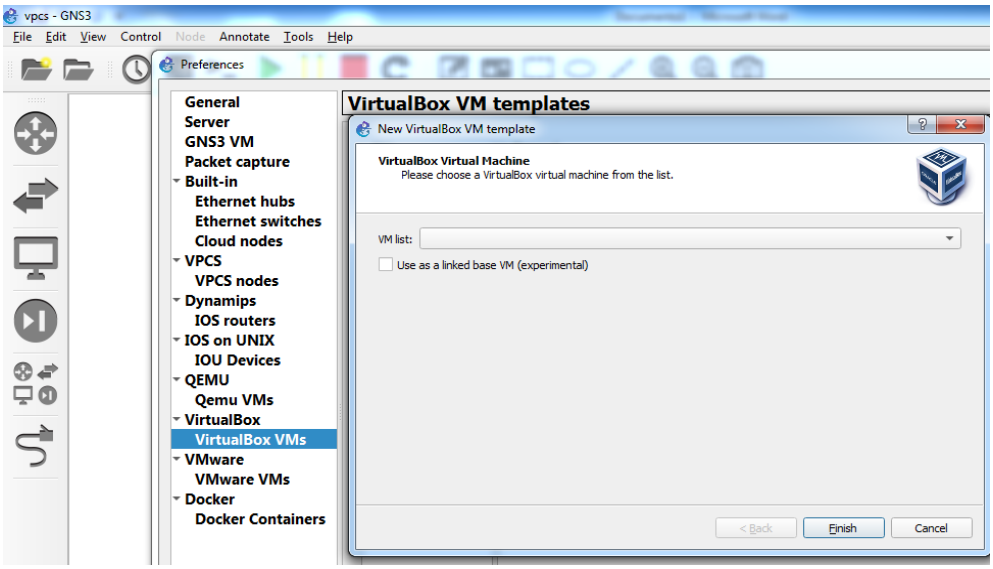


Figura 62. Vinculación de máquina virtual en GNS3.

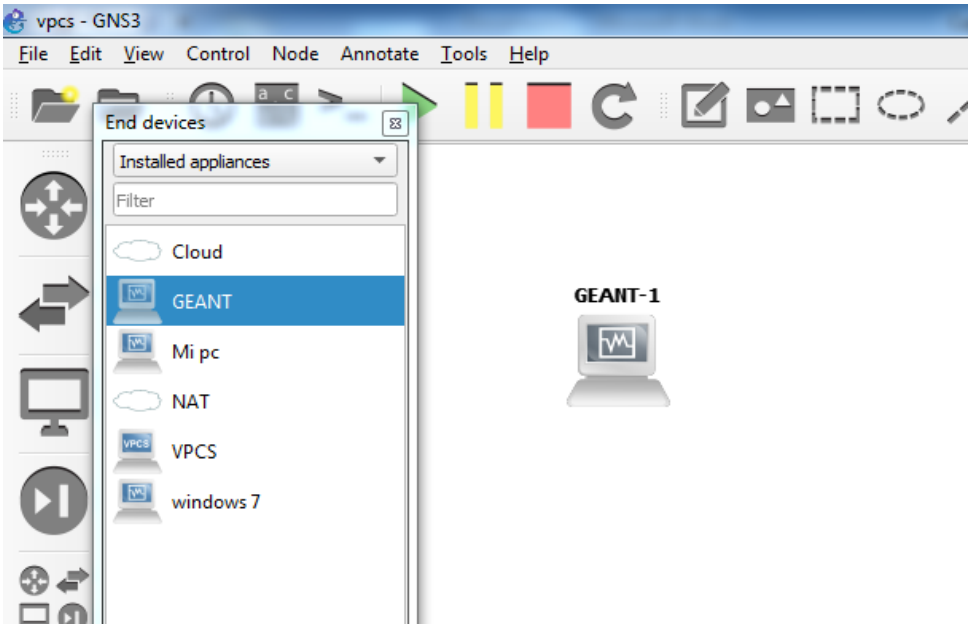


Figura 63. Máquina virtual vinculada en GNS3.

3.5 Configuración de SNMP de la red GEANT en GNS3

Después de haber configurado el enrutamiento en la emulación, el siguiente paso es configurar SNMP y definir el sistema de gestión de red. Se eligió a la estación de gestión llamada GEANT_1 conectada al *router* llamado Alemania_2, como se muestra en la figura 64.

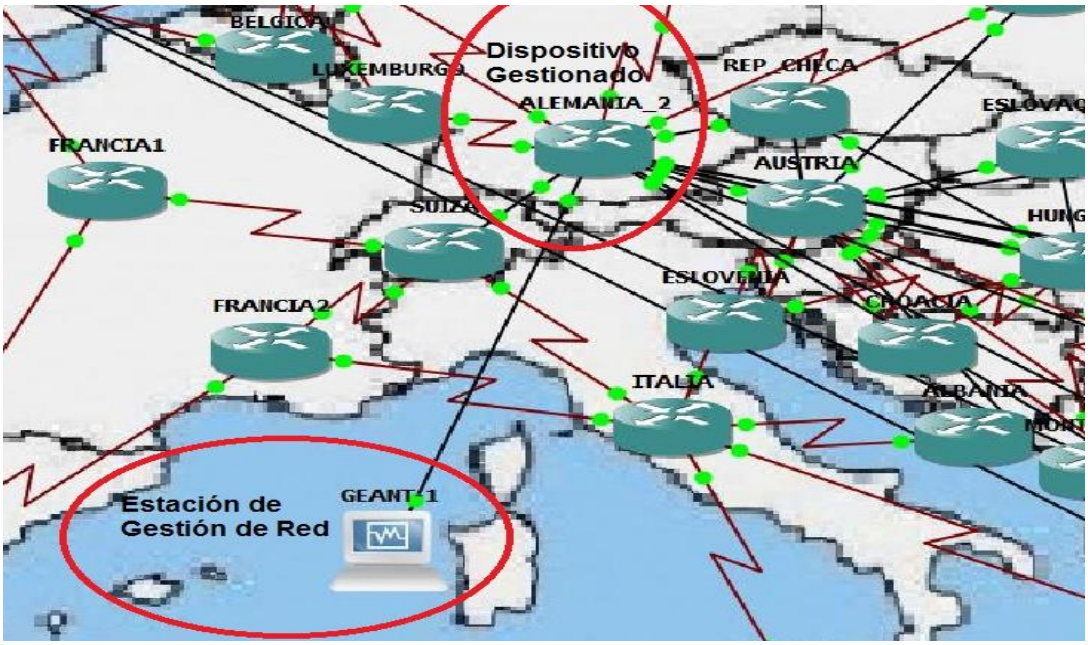


Figura 64. Elementos de Gestión.

3.5.1 Configuración de la estación de gestión de la red GEANT

Para la gestión en la emulación de la red GEANT se necesitó instalar el *software Power SNMP free Manager* en la máquina virtual, cabe aclarar que se instaló la versión gratuita, con este programa se realizó la gestión y monitoreo de los agentes, en la figura 65 se muestra el icono del programa *powerSNMP* instalado.

Figura 65. Instalación de *PowerSNMP free Manager*.

3.6 Configuración de SNMPV3 en la emulación de la red GEANT y en GNS3

Una vez instalada el software de gestión → abrimos la emulación de la red GEANT en GNS3. Se enciende el *router* llamado Alemania_2 y vía consola se configura SNMP accediendo en modo súper usuario y declarando el comando del

protocolo SNMP, con el nombre del grupo llamado “GEANT” y nombre de usuario llamado “UACM”, también se escoge el tipo de autentificación “*sha*” y la privacidad con el cifrado *“DES*” con contraseña “armando86”, como se ve en la figura 66.

```
ALEMANIA_2#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
ALEMANIA_2(config)#snmp-server group GEANT v3 priv
ALEMANIA_2(config)#$ user UACM GEANT v3 auth sha isa10 priv des armando86
ALEMANIA_2(config)#
*Jul 14 17:49:22.359: Configuring snmpv3 USM user, persisting snmpEngineBoots. Please Wait...

ALEMANIA_2(config)#
```

Figura 66. Configuración de SNMPV3.

3.6.1 Configuración de agentes

Ya que estuvo configurado el protocolo SNMP en el agente del *router*, se realizó la configuración para los mensajes *Get, Set* y *Trap de SNMP*. Se enciende la máquina virtual llamada GEANT → se abre *PowerSNMP free Manager* para agregar los *routers* gestionados → seleccionamos SNMP *Agents* → *Add Agent*, como se muestra en la figura 67.

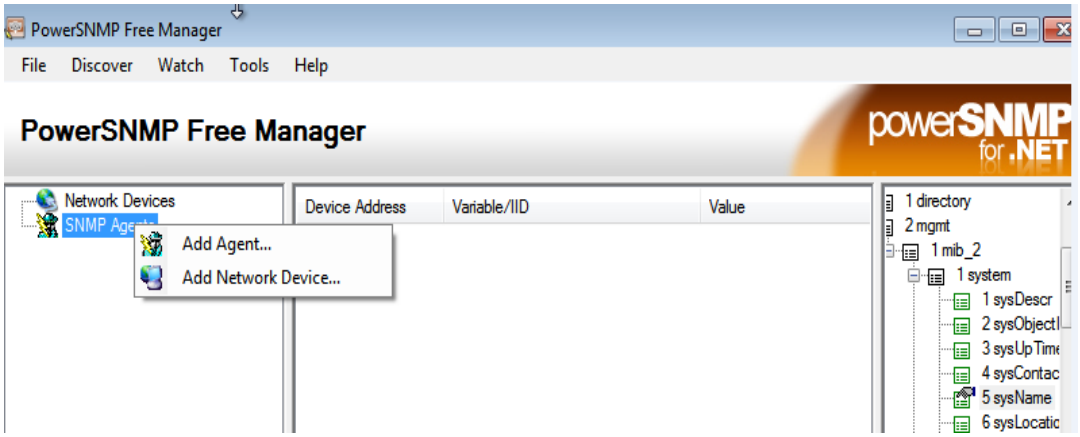


Figura 67. Interfaz de *powerSNMP free Manager* agregando al agente de Alemania_2.

Se configuró al agente agregando la dirección IPV6 2001:db8:75::1, que corresponde al *router* Alemania_2 a gestionar, el puerto 161, la versión 3, el nombre del grupo, nombre de usuario, tipo de autentificación y los datos pedidos que se muestran en la figura 68.

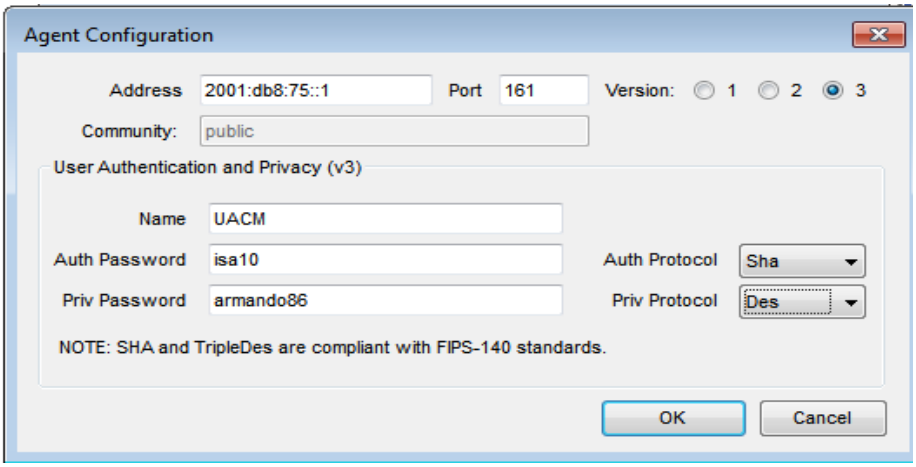


Figura 68. Configuración de parámetros del agente Alemania_2 con *PowerSNMP*.

Realizada la configuración del agente, se verifica que sea detectado, como se observa en la figura 69.

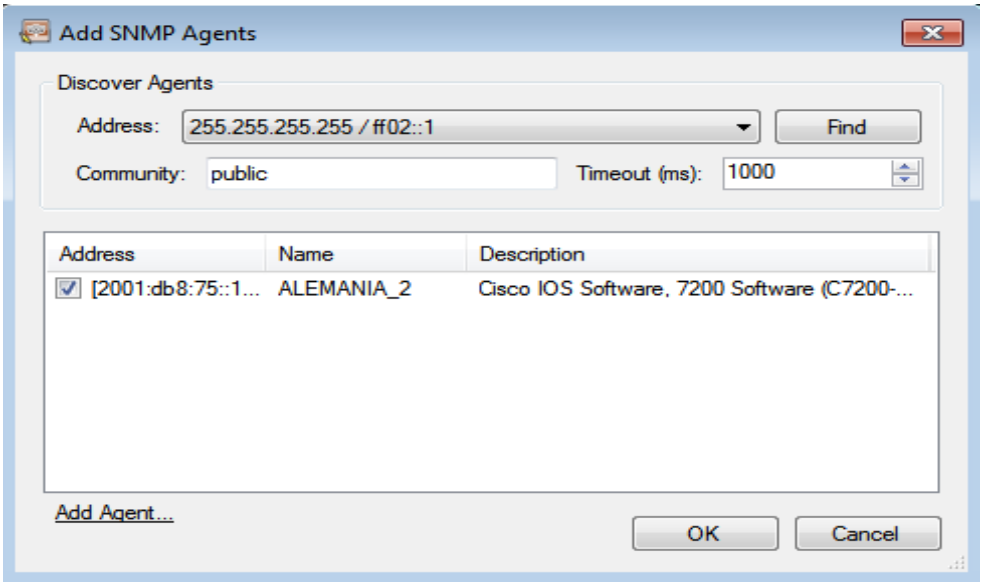


Figura 69. Agente configurado correctamente para Alemania_2.

La figura 70 indica que el agente llamado Alemania_2 fue configurado exitosamente por la entidad SNMPV3 en *PowerSNMP Free Manager,* y que puede ser administrado.

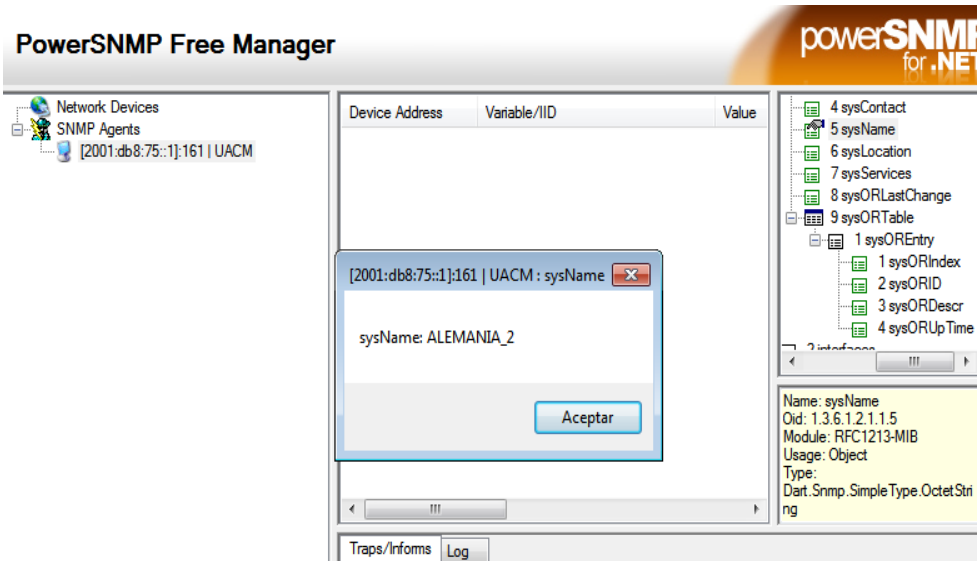


Figura 70. Agente disponible para su gestión en SNMPV3.

Realizando los mismos pasos se agregan los 45 agentes restantes que conforman la red avanzada GEANT. En las figuras 71 y 72 se muestra los agentes agregados a la consola *powerSNMP free Manager.*

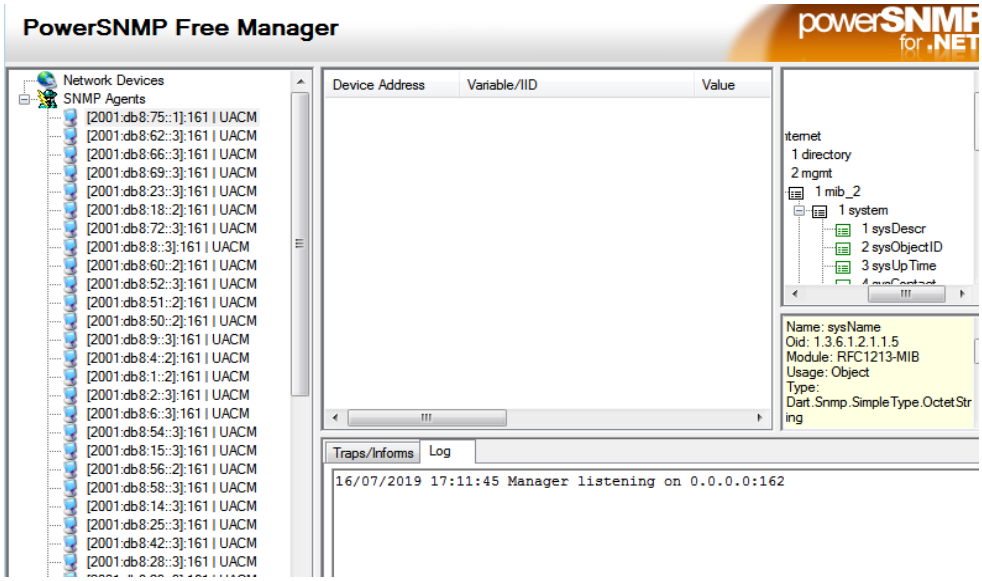


Figura 71. Agentes agregados correctamente como Austria, Suiza, Luxemburgo de la red avanzada GEANT.

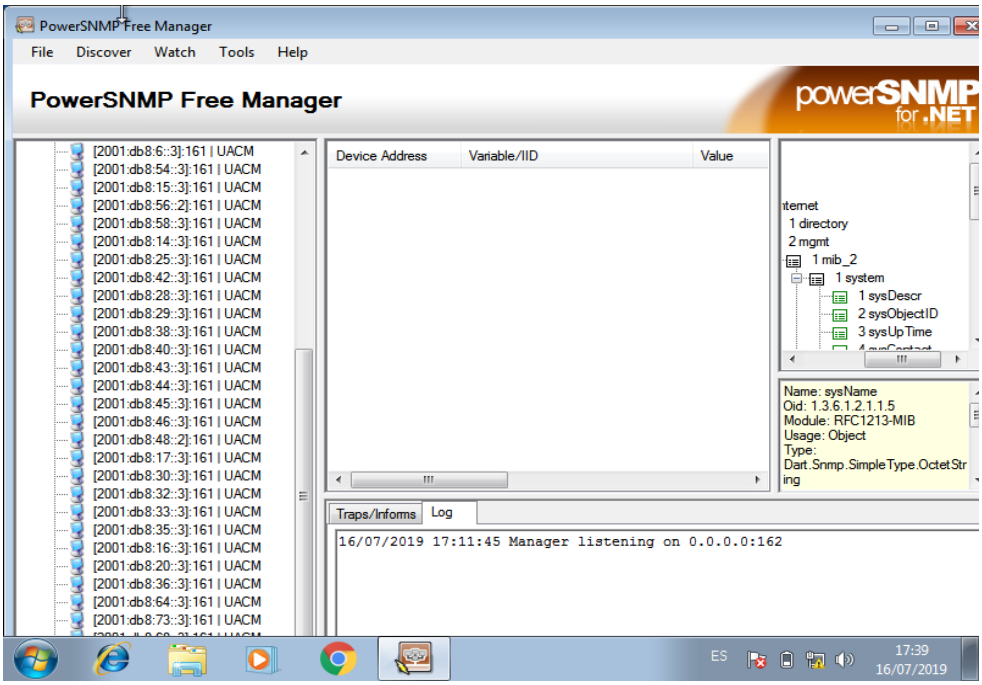


Figura 72. Dinamarca, Suecia, Polonia, Austria son algunos de los agentes agregados exitosamente en *PowerSNMP Free Manager*.

Capítulo 4.

Resultados y conclusiones

4.1 Prueba de conectividad entre *routers* de la Red Avanzada GEANT

Para comprobar que las configuraciones de enrutamiento fueron correctas, se utilizó la herramienta llamada “PING”. Esta herramienta ayudó a comprobar la conectividad en la red, a través, del envío de mensajes ICMPV6 (*Internet Control Message Protocol IPV6)*. Este protocolo realiza el informe de errores y funciones de diagnóstico.

Para realizar la prueba PING accedemos en modo consola del *router*. En la figura 73 se muestra la prueba de conectividad del *router* Bulgaria al *router* Finlandia con dirección IPV6 2001:db8:58::3.

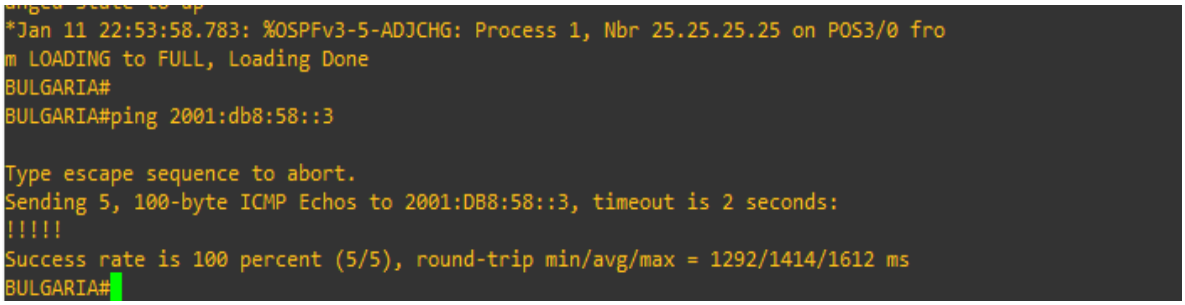


Figura 73. Prueba de conectividad de Bulgaria a Finlandia.

En la figura 73, se muestra la consola del *router* Bulgaria realizando la prueba *ping,* la cual muestra como resultado el número de paquetes enviados, paquetes perdidos y el tiempo de vida de cada paquete. Se observa el envío de 5 paquetes y se reciben los mismos 5 paquetes, por lo que se demuestra que la conectividad es exitosa.

En la figura 74, se observa la conectividad a otros *routers*. De igual modo se realizó el envío y recepción de paquetes con éxito.

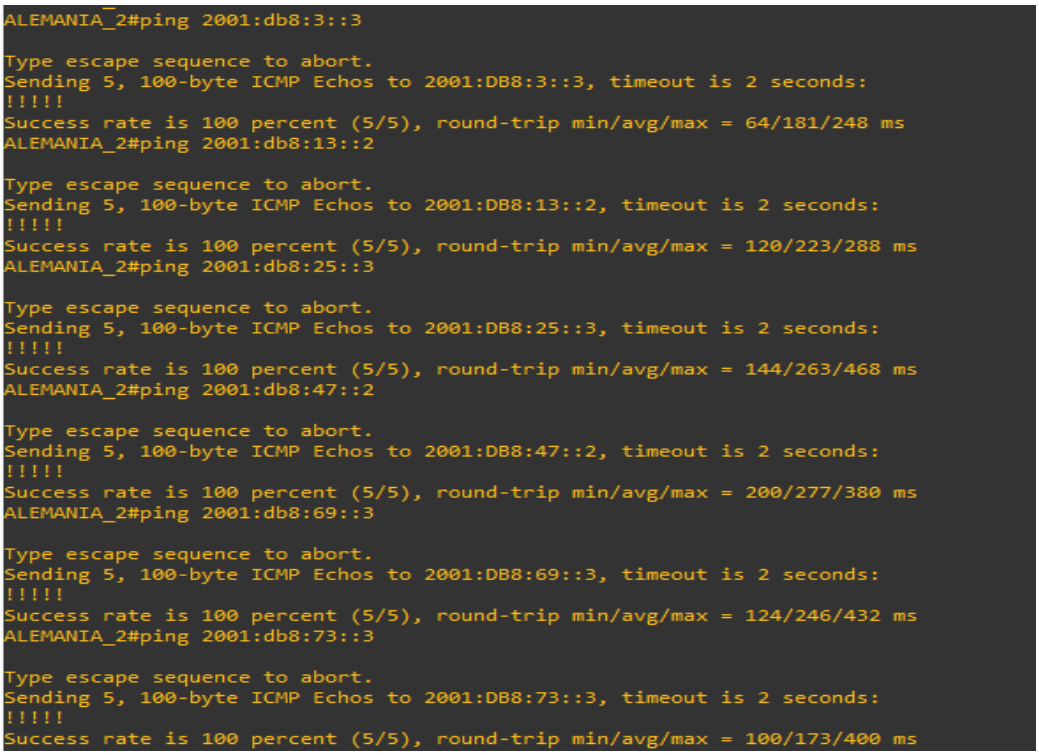


Figura 74. Conectividad de Alemania_2 a España, Reino Unido, Croacia, Letonia, Austria y Armenia.

4.1.2 Prueba de conectividad de VPCS, en la emulación de GEANT

Para realizar la prueba de conectividad entre las VPCS del emulador, se eligió la VPCS-Azerbaiyán y se realizó “ping” a las VPCS-Islandia, Finlandia e Israel, como se observa en la figura 75. En ésta se muestra la latencia que tarda enviar un paquete dentro de la red, el cual es medido en ms (milisegundos) y es el tiempo que tardan en comunicarse las VPCS entre sí.

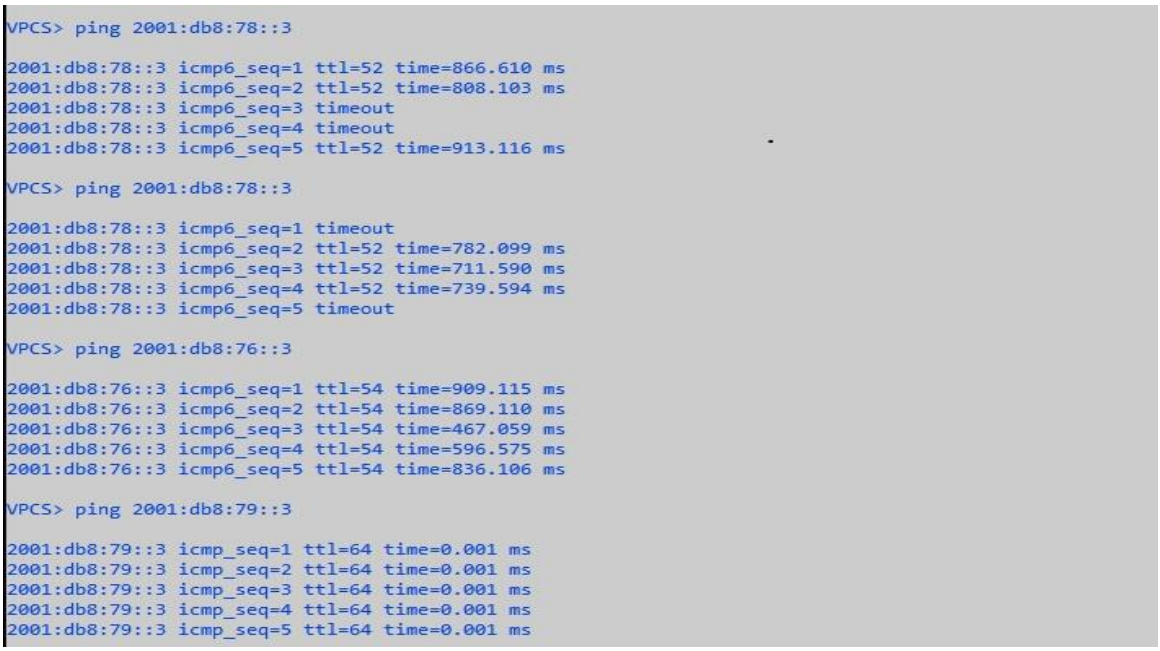


Figura 75. Prueba de conectividad entre VPCS en la emulación GEANT.

4.1.3 Tabla de enrutamiento con el protocolo OSPF

Las tablas de enrutamiento nos indican redes y todos los *router* que formen parte de la red deben tener las mismas rutas. Estas redes están identificadas con el prefijo “O” el cual indica que están configuradas con el protocolo OSPF. Para solicitar la tabla de enrutamiento se usa el comando “*sh ipv6 route*”. En la figura 76 se muestra la ejecución del comando, así como, un análisis de la primera red, en la cual se obtiene lo siguiente:

- Prefijo O. Correspondiente al protocolo OSPF.
- Prefijo C. Corresponde a una red conectada directamente al *router*.
- Prefijo L. Corresponde a una red local.
- 2001:DB8:1::/64. Dirección IPV6, correspondiente a la red conformada entre Portugal y España.
- [110/4]. El número 110 significa la ruta administrativa, este número es utilizado por OSPF, existen diferentes rutas administrivas dependiendo del protocolo utilizado. En la tabla 4 se muestran las rutas administrativas que los *routers* Cisco admiten.

Fuente de la Ruta	Valores Predeterminados de la Distancia
Interfaz conectada directamente	0
Ruta estática	1
EIGRP *(Enhanced Interior Gateway Routing Protocol)*	5
BGP *(External Border Gateway Protocol)*	20
EIGRP Interno	90
IGRP	100
OSPF	**110**
IS-IS *(Intermediate System-to-Intermediate System)*	115
RIP *(Routing Information Protocol)*	120
EGP *(Exterior Gateway Protocol)*	140
ODR (Ruteo a pedido)	160
EIGRP externo	170

Tabla 4. Rutas administrativas utilizadas por Cisco.

- [110/4]. El número 4 significa la métrica, el número de saltos para llegar al *router* deseada.
- via FE80:C810:3FF:FE30:1D. Siguiente dirección IPV6 a la cual se le debe reenviar el paquete.
- FastEthernet0/1. A través de que interfaz va salir el mensaje.

En la figura 76 se muestran algunas de las interfaces del *router* Alemania_2, mientras que la lista completa se puede consultar en el anexo 2.

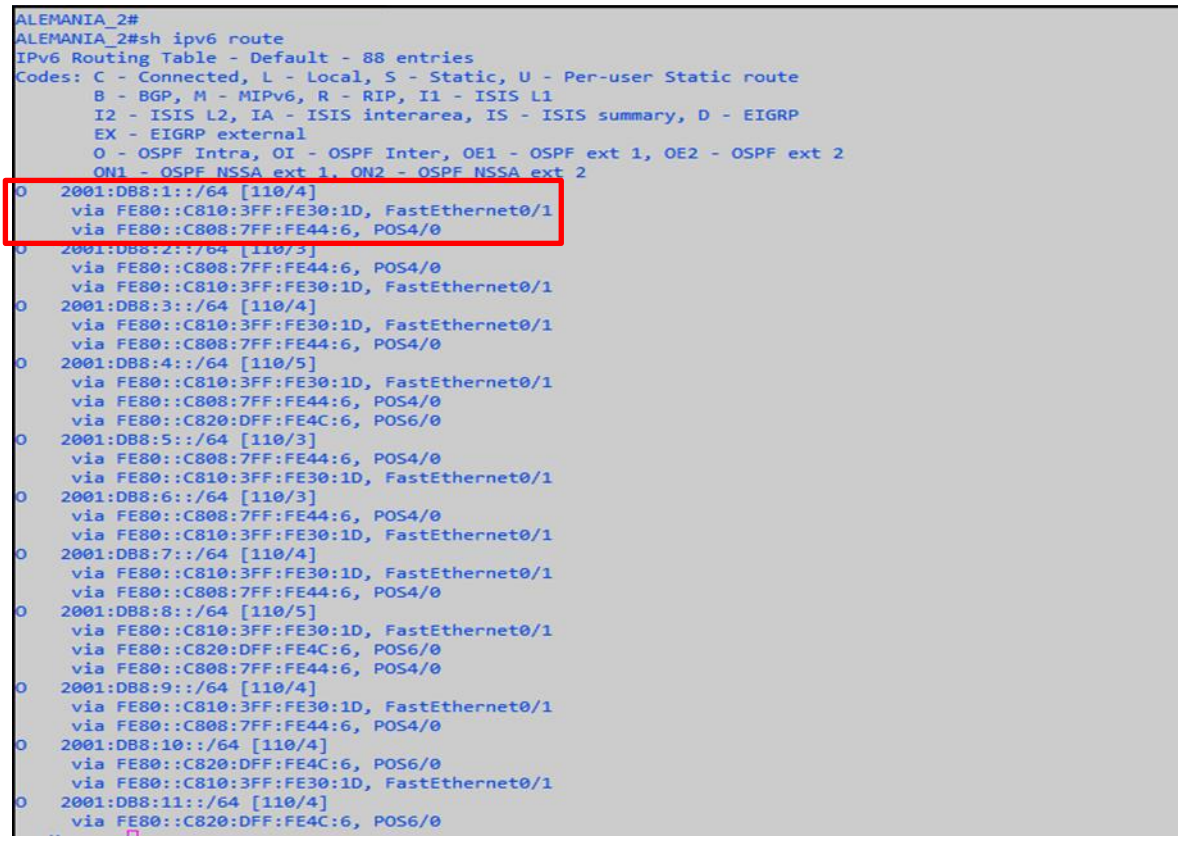


Figura 76. Tabla de enrutamiento de OSPFV3 del *router* ALEMANIA_2 (parte 1).

4.1.4 Análisis de los paquetes OSPF de la emulación GEANT

En la emulación se lograron obtener tres de los cinco tipos de mensajes de OSPF, que son del tipo 1, 4 y 5.

Para analizar los mensajes se utilizó el software *Wireshark*, el cual es un analizador de protocolos, este ofrece una plataforma gráfica que muestra el proceso de convergencia de OSPF.

En la figura 77, se muestra la interfaz del programa Wireshark, este programa nos proporciona información de los mensajes que son enviados en cada conexión, dando el tiempo en el que son generados, la fuente de donde se originaron, el tipo de protocolo, longitud del paquete y el tipo de mensaje que se está enviando.

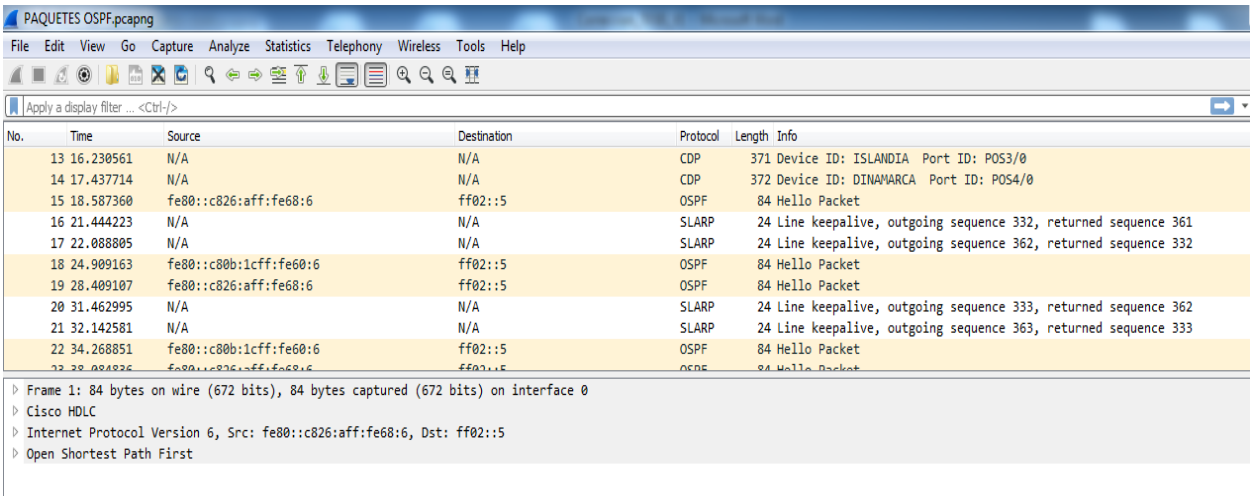


Figura 77. Interfaz del programa *Wireshark*.

4.1.5 Encabezado del paquete OSPF

El programa *Wireshark* nos permitió analizar el formato de cada paquete, así como el encabezado del mensaje. En la figura 78 se muestra el contenido del encabezado de los mensajes OSPF, en la cual nos indica:

- *Src*. Dirección fuente, es decir, de donde proviene el mensaje.
- *Version*. Tipo de versión del protocolo OSPF, en este caso es la versión 3.
- *Message Type*. Tipo de mensaje, en este caso es el mensaje tipo 4 LSU.
- *Packet Length*. Longitud del paquete.
- *Source OSPF Router*. *Router ID* del cual proviene el mensaje.
- AREA ID. Área a la que pertenece el *router*.

Un encabezado OSPF contiene toda la información necesaria para determinar si el paquete debe ser aceptado para su procesamiento.

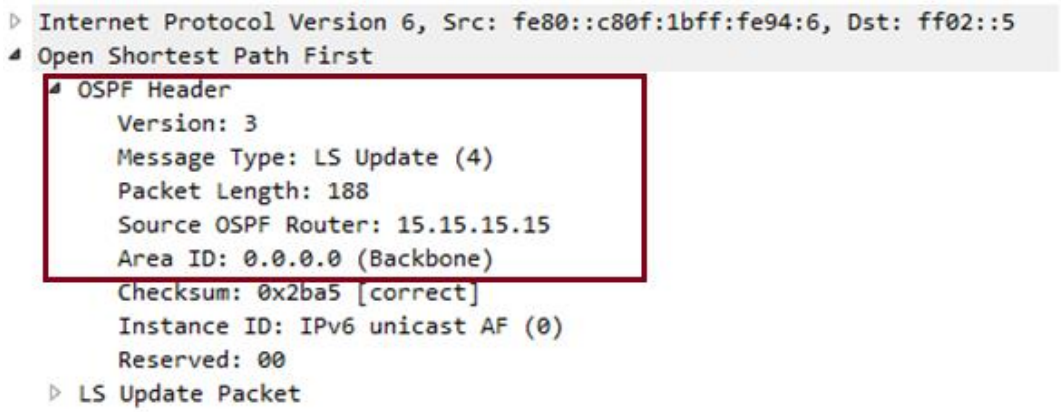


Figura 78. Encabezado de mensaje OSPF.

4.1.6 Paquete *Hello* OSPF en la emulación GEANT

Los paquetes *Hello*, son los primeros mensajes enviados por cada *router* que trabaja con OSPF, el cual es utilizado para formar adyacencias.

Cuando un *router* recibe un mensaje *Hello* de su vecino, y este mensaje contiene el nombre del *router* local en la sección *Neighbor*, se posee una comunicación bidireccional entre ambos equipos (comunicación o estado *two-way*).

En la emulación, el paquete *Hello* proporciona información, la cual nos permitió analizar los criterios de adyacencia de OSPFV3. En la figura 79, se muestra el paquete *Hello*, el cual proporciona información tal como:

- *OSPF Header*. Encabezado del mensaje OSPF.
- *Interface ID*. Índice de interfaz de la tabla de prefijos de dirección IPV6, en este caso el prefijo es 9.
- *Router Priority*. Prioridad del *router*, en este caso es de 1.
- *Hello interval*. Intervalo de Tiempo en el que es enviado el mensaje *Hello*, en este caso es de 10 segundos.
- *Active Neighbor*. Router vecino, en este caso lo registró a través del *router ID*.

Esta información es de gran importancia porque podemos corroborar efectivamente, que es la misma información que nos indica la literatura respecto al mensaje *Hello*, previamente explicado en el capítulo 2.

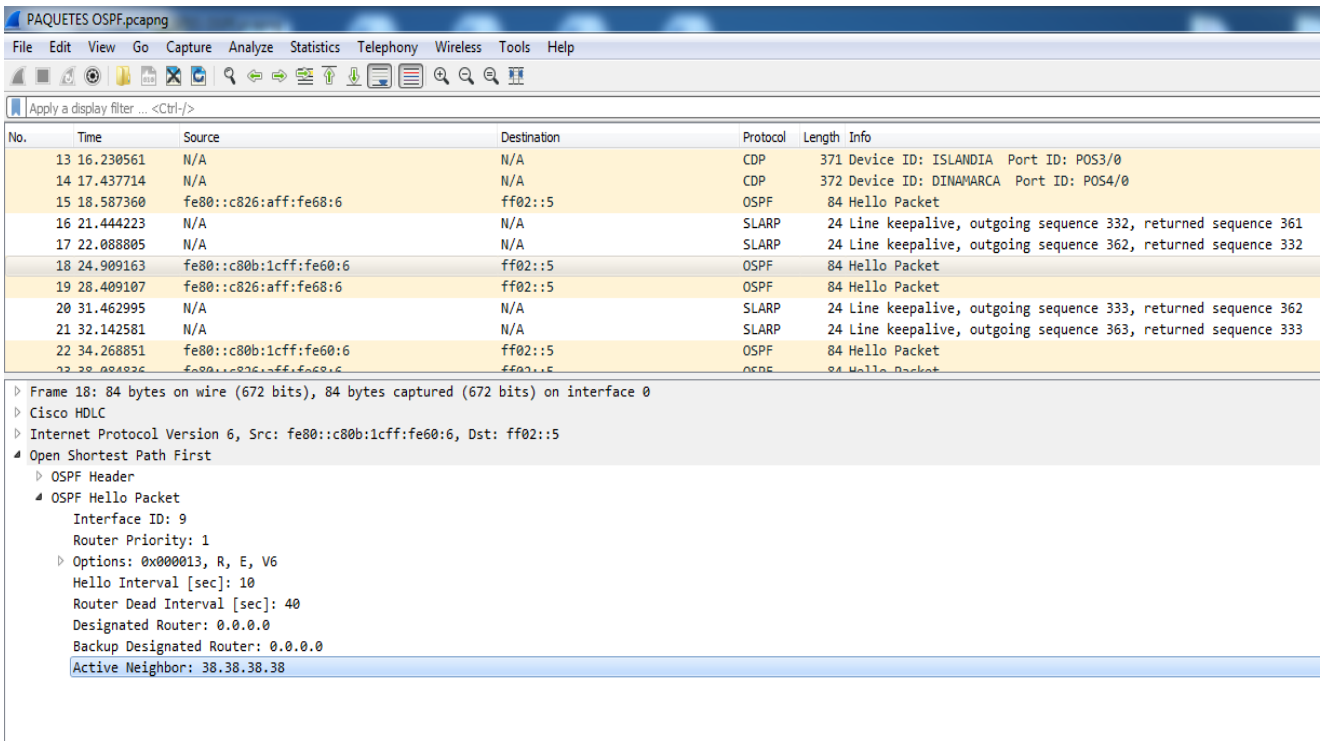


Figura 79. Paquete *Hello* con la herramienta Wireshark.

4.1.7 Paquete LSU (*Link State Update*) OSPF en la emulación GEANT

Estos paquetes inundan con los LSAs consultados por el vecino OSPF, donde un sólo LSU puede contener múltiples LSAs.

Las redes que soportan *multicast*, utilizan dicha dirección para enviar los LSUs.

Todos los LSUs deben ser respondidos por un LSAck, los cuales corresponden a un acuse de recibo de éstos.

En la figura 80 se observa el tipo de mensaje, el cual es *LS Update* y nos muestra:

- *OSPF Header*. Encabezado del paquete LSU.
- *Version*. Tipo de versión de OSPF, en este caso es la versión 3.

- *Message Type*. Tipo de mensaje número 4.
- *Packet Length.* Longitud del paquete.
- *Source OSPF router*. Router fuente del mensaje, en este caso lo indica con el *router ID*.
- *Area ID*. Área del *router*.
- *Checksum*. Comprobación del estado del mensaje, en este caso lo marca correcto.
- *Instance ID*. Tipo de dirección IPV6 que se usó.

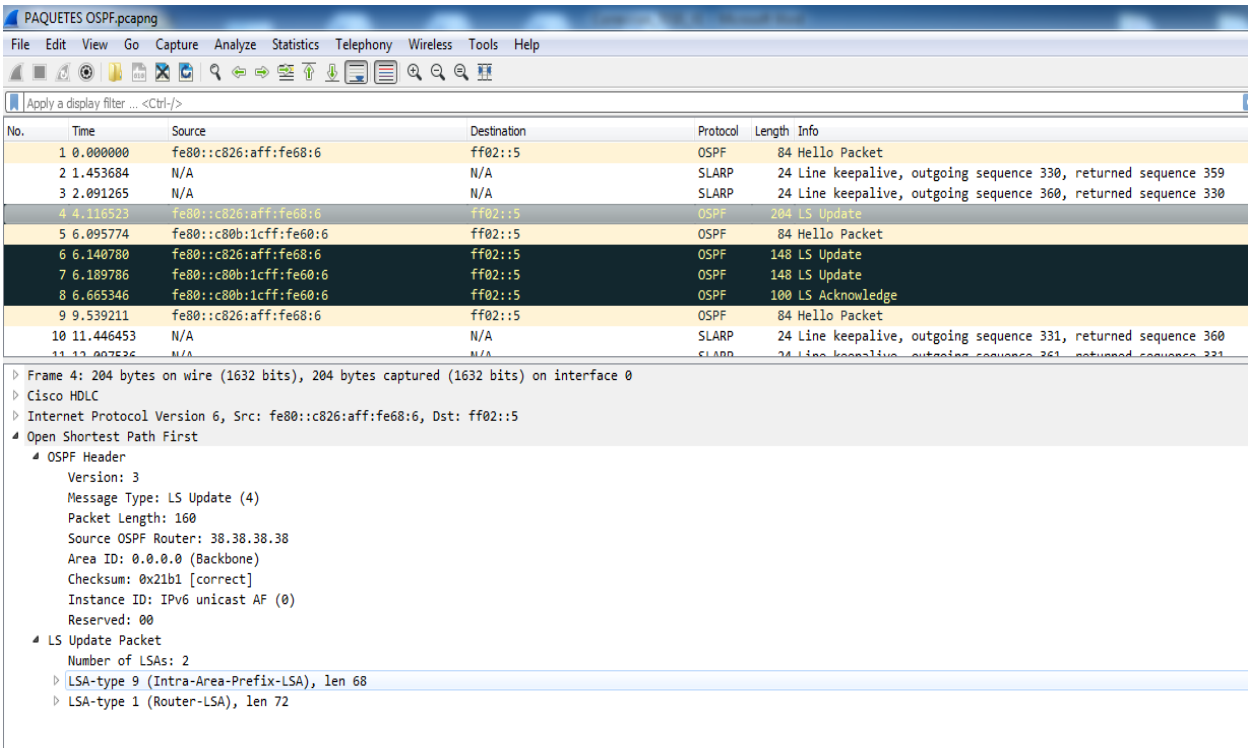


Figura 80. Paquete LSU OSPF en emulación.

La figura 80 también nos indica que este paquete LSU cuenta con dos LSA, los cuales son:

- *LSA type 9.*
- *Lsa type 1*.

Los LSA tipo 9, son aquellos que se propagan dentro de todas las áreas y los genera cada uno de los nodos para indicar los prefijos de red directamente

conectados. En la figura 81 se muestra el LSA que se generó en el paquete LSU y contiene:

- *LSA Type*. Tipo de LSA.
- *LINK STATE ID*. ID del estado enlace.
- *Advertising Router*. Identifica el *router* que originó el LSA.
- *Checksum*. Comprobación del estado del mensaje.
- *Length*. Longitud del LSA.
- *Metric*. Métrica o costo para llegar al destino.

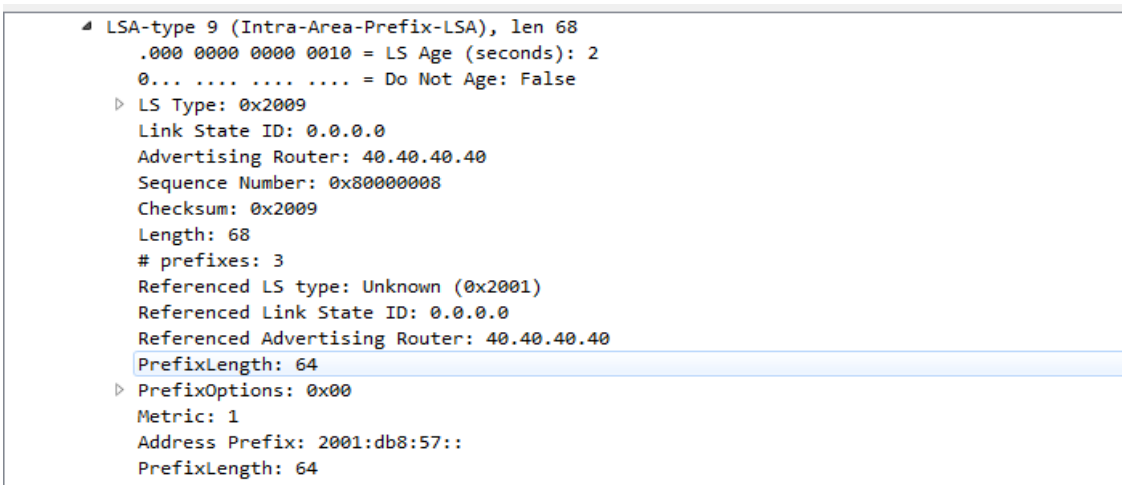


Figura 81. LSA del paquete LSU.

Los LSA tipo 1 indican los enrutadores vecinos, así como el costo para llegar a ellos. En la figura 82, nos muestra el LSA tipo 1 del paquete LSU.

- *LSA Type*. Tipo de LSA.
- *LINK STATE ID*. ID del estado enlace.
- *Advertising Router*. identifica el *router* que originó el LSA.
- *Checksum*. Comprobación del estado del mensaje.
- *Length*. Longitud del LSA.
- *Metric*. Métrica o costo para llegar al destino.
- *Router Interfaces*. Nos indica las interfaces del *router*.
- *Entry*. Interfaz conectada, en este caso tiene 1 interfaz conectada.
- *Type*. Tipo de conexión.

- *Metric*. Métrica o costo para llegar a la interfaz conectada.
- *Neighbor Router ID*. *Router* vecino.

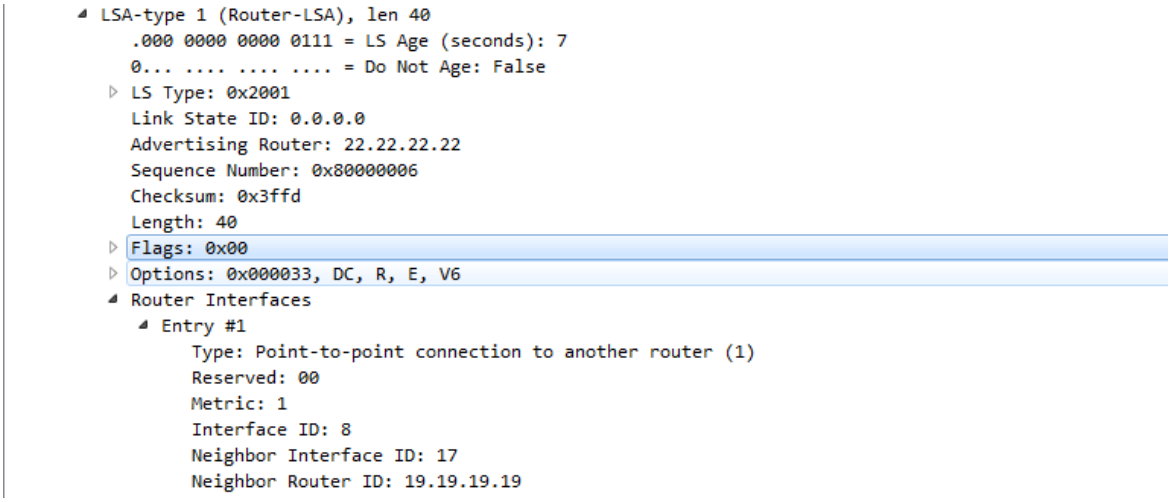


Figura 82. LSA tipo 1 generado en el paquete LSU.

4.1.8 Paquete LS *Acknowledge* en la emulación GEANT

Este mensaje confirma la recepción del paquete enviado. En la figura 83, nos muestra el encabezado del paquete LS Acknowledge, dando información acerca de la longitud del paquete, la versión y el área del *route,* así como, nos indica que cuenta con dos LSAs.

- *OSPF Header*. Encabezado del paquete LS Acknowledge.
- *Version*. Tipo de versión de OSPF, en este caso es la versión 3.
- *Message Type*. Tipo de mensaje número 5.
- *Packet Length*. Longitud del paquete.
- *Source OSPF router*. *Router* fuente del mensaje, en este caso lo indica como el *router ID*.
- *Area ID*. Área del *router*.
- *Checksum*. Comprobación del estado del mensaje, en este caso lo marca correcto.

- *Instance ID*. Tipo de dirección IPV6 que se usó.

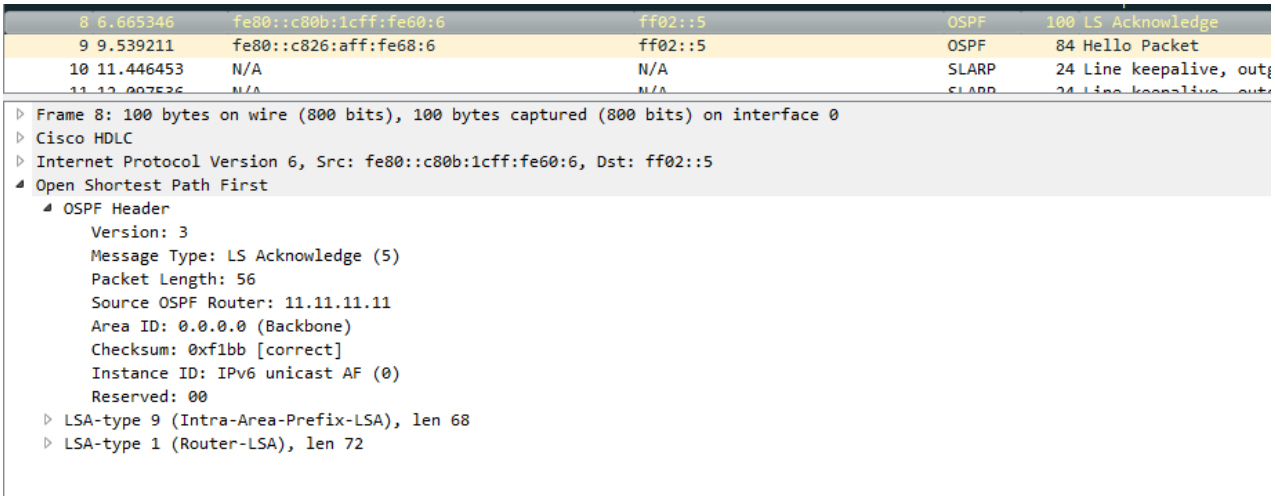


Figura 83. Paquete ACK OSPF en emulación.

En la figura 84 se hace un desglose del LSA tipo 9, el cual contiene:

- *LSA type*. Tipo de LSA, en este caso es del tipo 9.
- *LINK STATE ID*. ID del estado enlace.
- *Advertising Router.* Identifica el *router* que originó el LSA.
- *Checksum*. Comprobación del estado del mensaje.
- *Length*. Longitud del LSA.

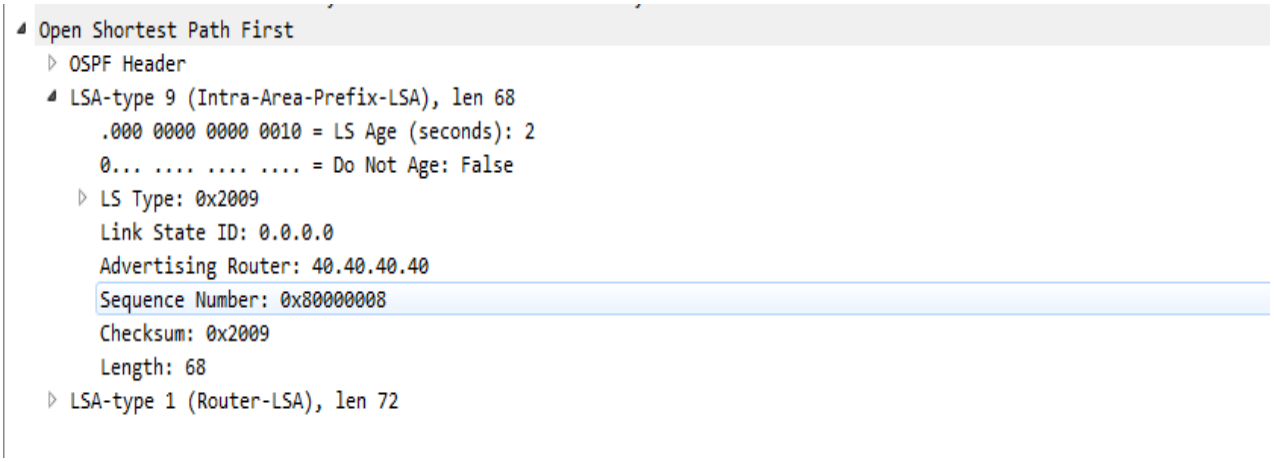


Figura 84. LSA tipo 9 del paquete *LS Acknowledge*.

En la figura 85, se desglosa la información que contiene el paquete LSA tipo 1 del paquete *LS Acknowledge*, el cual contiene:

- *LS Type*. Tipo de LSA
- *LINK STATE ID*. ID del estado enlace.
- *Advertising Router*. Identifica el *router* que originó el LSA.
- *Checksum*. Comprobación del estado del mensaje.
- *Length*. Longitud del LSA.

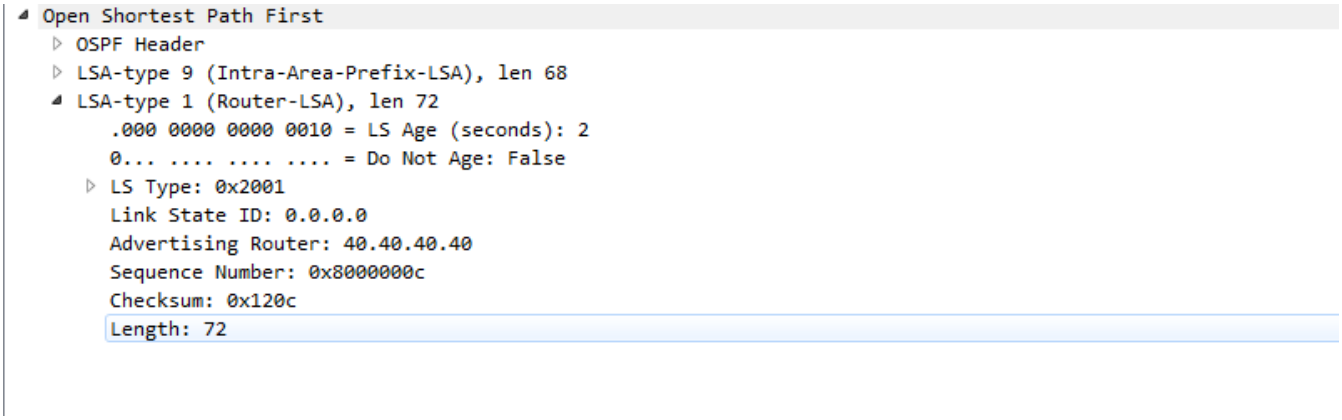


Figura 85. LSA tipo 1 correspondiente al paquete *LC Acknowledge*.

4.1.9 Análisis de los paquetes *Hello, LSU y LS Acknowledge* con la herramienta *Flow Graph*

Wireshark es una herramienta que permite analizar el tráfico de red, dentro de esta herramienta se puede hacer uso de la herramienta *Flow Graph*, el cual permite verificar de forma rápida y sencilla las conexiones entre dos *routers* de manera gráfica.

Para acceder a *Flow Graph* ingresamos a Wireshark posteriormente seleccionamos el menú statistics, se elige *flow graph* y en este se puede elegir observar todos los paquetes capturados o solo los paquetes filtrados.

En la figura 86 se observa él envió de mensajes *Hello*, *LS Update*, *LS Acknowledge* usando la herramienta *Flow Graph*.

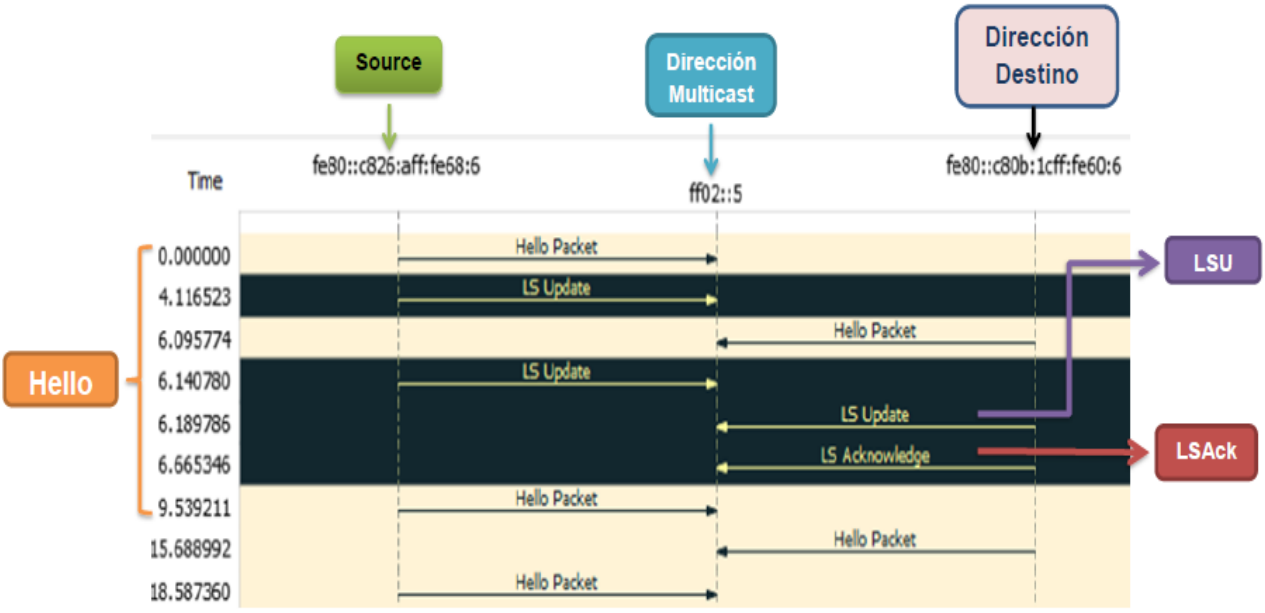


Figura 86. Análisis de los mensajes de OSPF a través de *Flow Graph*.

En donde:

- Dirección *multicast*: Es la dirección en IPV6 de todos los *routers* ISO usando el protocolo OSPFV3.
- *Hello*: El paquete *Hello* es enviado por la dirección *multicast* ff02::5 en IPV6 cada 10 seg.
- *LSU*: *LS Update* informa de los cambios realizados en la topología de la red, este mensaje será retransmitido hasta que se confirme con un mensaje *LS Acknowledge (LSAck).*
- *Source*: fe80::c826:aff:fe68:6 es la dirección fuente.
- Dirección Destino: fe80::c80b:1cff:fe60:6 es la dirección siguiente a la que se le debe renviar el mensaje.

4.2 Prueba de Gestión de la Red Avanzada GEANT en emulación

Para poder monitorear de manera segura y de forma constante la red avanzada GEANT, se utilizó el software *powerSNMP*, para tener comunicación constante entre la estación de gestión de red y los agentes configurados.

4.2.1 Pruebas de funcionamiento de los agentes en SNMPV3

A) Prueba de solicitud de nombre a los *router* Islandia e Italia

Para realizar esta prueba se escogió al agente del *router* Islandia (2001:db8:14::3), posteriormente se seleccionó la variable *sysName*, esto nos dio como resultado el nombre del *router*, como se observa en la figura 87. Se realizó una segunda prueba y al seleccionar el agente del *router* Italia (2001:db8:18::2), muestra el nombre asignado como se observa en la figura 88.

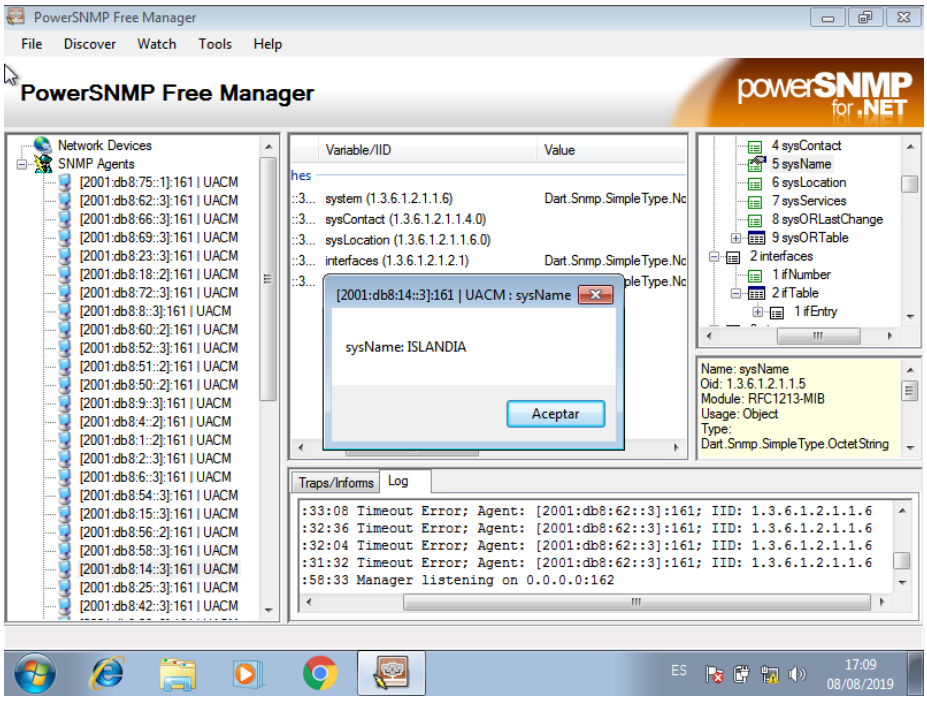


Figura 87. Agente de Islandia indicando su nombre correctamente

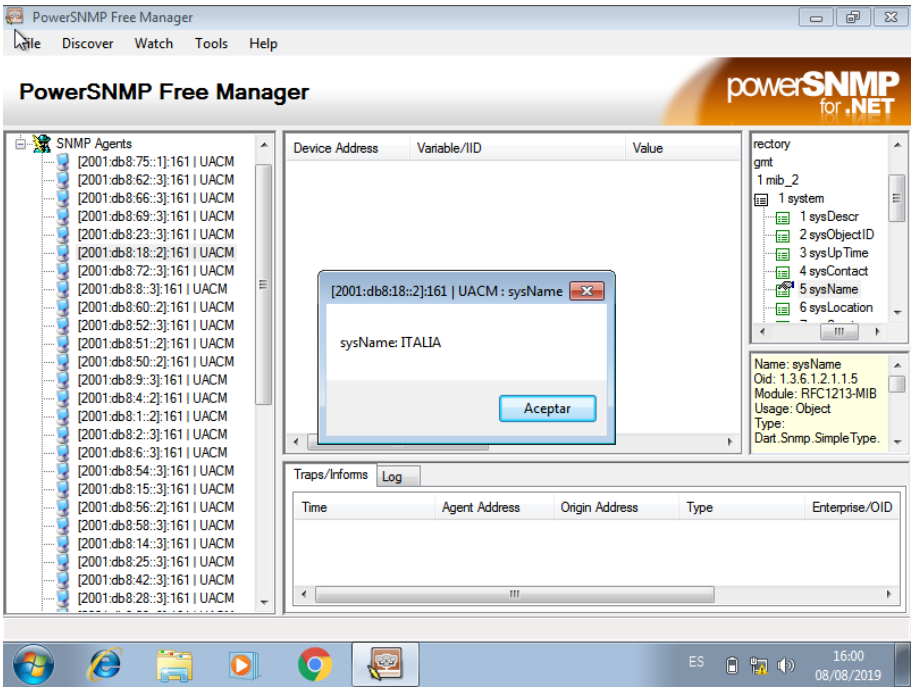


Figura 88. Agente de Italia indicando su nombre correctamente.

B) Prueba de mensajes de entradas *GetRequest*

Después de haber solicitado el nombre del *router* de Alemania_2 y de otros *routers*, podemos analizar otras opciones que nos ofrece el programa *powerSNMP*. Se analizó el mensaje *GetRequest,* cabe aclarar que este mensaje es usado por el agente como una respuesta, indicando el éxito o fracaso a una petición. La figura 89, indica el número de paquetes *GetRequests* que recibió el *router* Islandia, que en este caso fueron 16 mensajes.

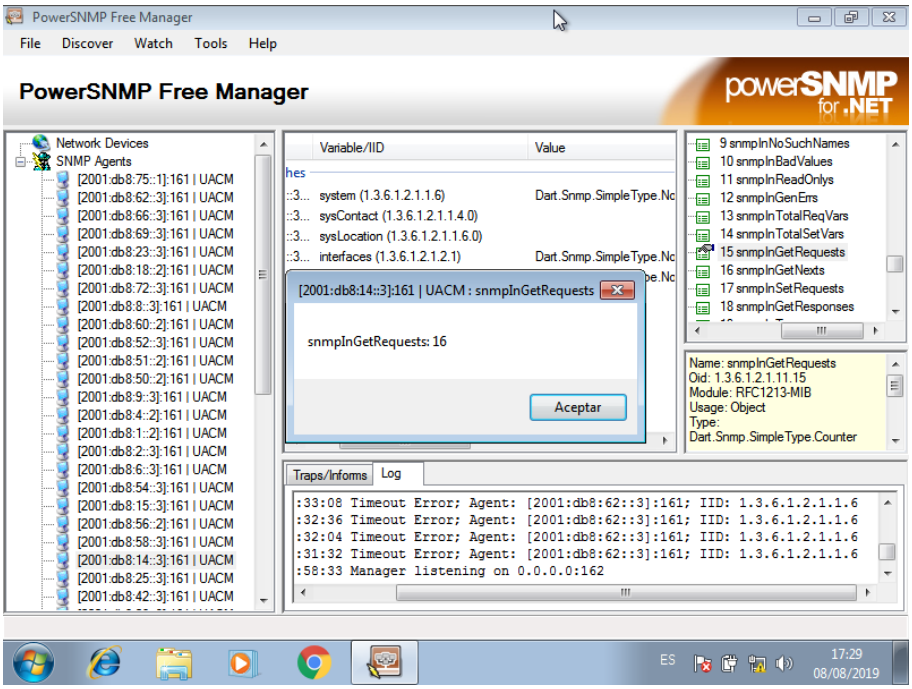


Figura 89. Paquete *GetRequest* recibidos en el agente del *router* Islandia.

C) Prueba de solicitud de descripción del *router*

Algunas veces es difícil recordar que tipo de *router* se usó al conectar una red o el administrador de red no nos proporciona ese dato. Así que si necesitamos conocer la descripción del *router* que vamos a gestionar, esto lo podemos obtener mediante SNMP con la variable *sysDescr*.

En la figura 90 se obtiene la información del *router* Alemania_2, donde se observó que es un *router* Cisco 7200, así como la página donde se obtiene el soporte de este *router*, y por último podemos ver la fecha de compilación del *router*.

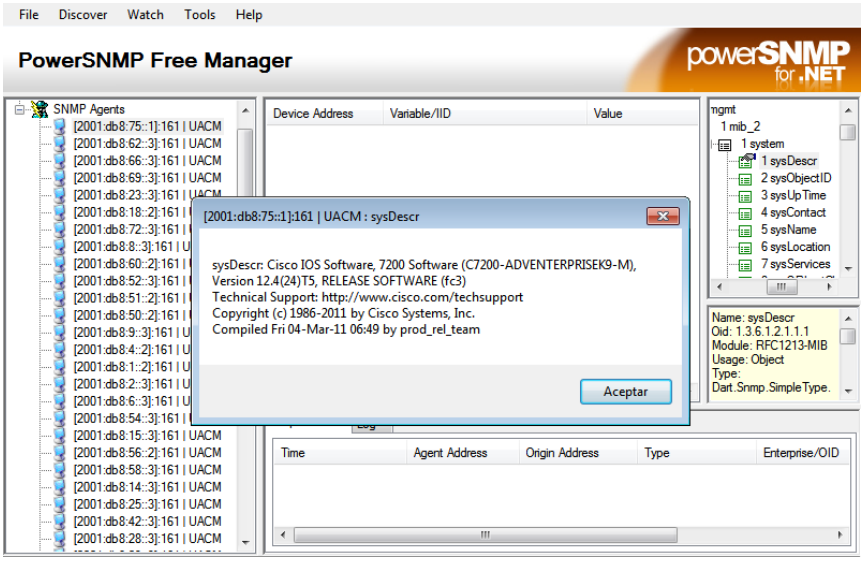


Figura 90. Información de descripcion del *router* Alemania_2.

En la figura 91 se observa la descripción del *router* Italia, debido a que se usó el mismo IOS del *router* Cisco 7200, la información desplegada en la descripción del *router* es la misma. Cabe recalcar que la direccion IPV6 del *router* Italia es diferente.

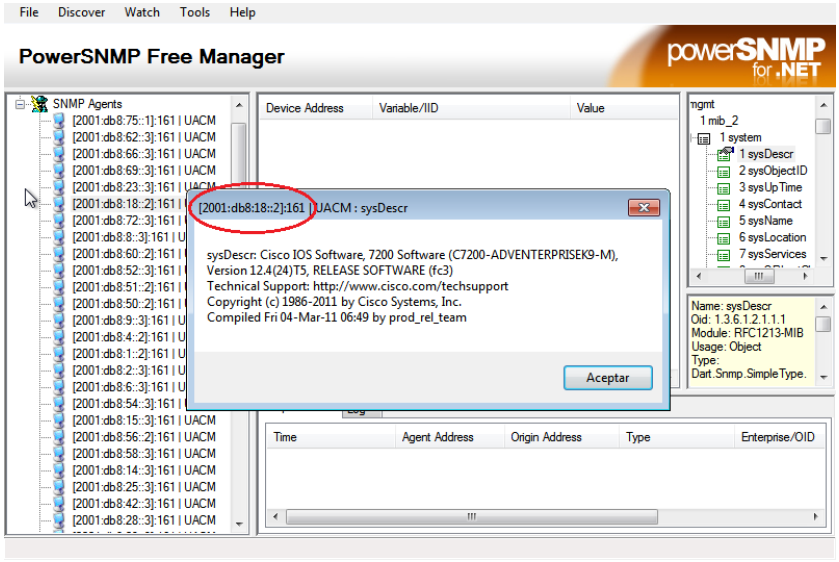


Figura 91. Información de descripcion del *router* Italia.

D) Prueba de solicitud del número de interfaces

Algo que nos gustaría conocer cuando administramos un *router,* es conocer el número de interfaces con las que cuenta, para saber cuales están en *up* o en *down*.

Este resultado se muestra en la figura 92, el cual es el número de interfaces con las que cuenta el *router* Alemania_1.

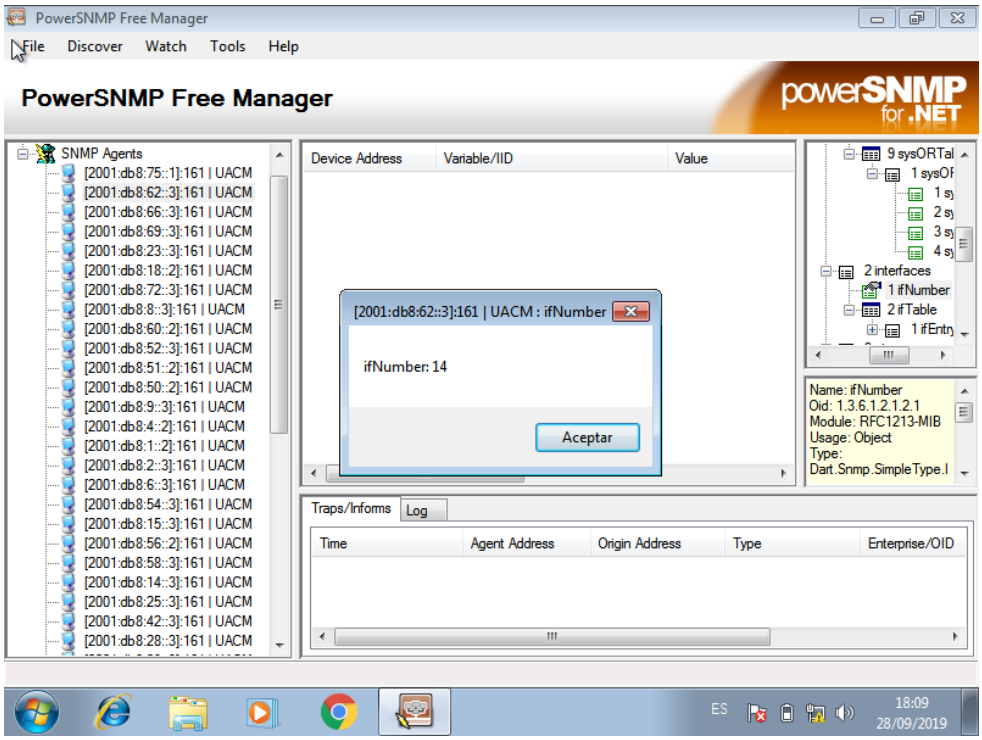


Figura 92. Número de interfaces con las que cuenta el *router* Alemania_1.

E) Solicitud de tabla de enrutamiento

Después de haber solicitado el nombre del *router*, el número de interfaces, etc., se procedió a solicitar información de la tabla de enrutamiento del *router* Alemania_1 por medio de *powerSNMP* obteniendo la siguiente información, figura 93.

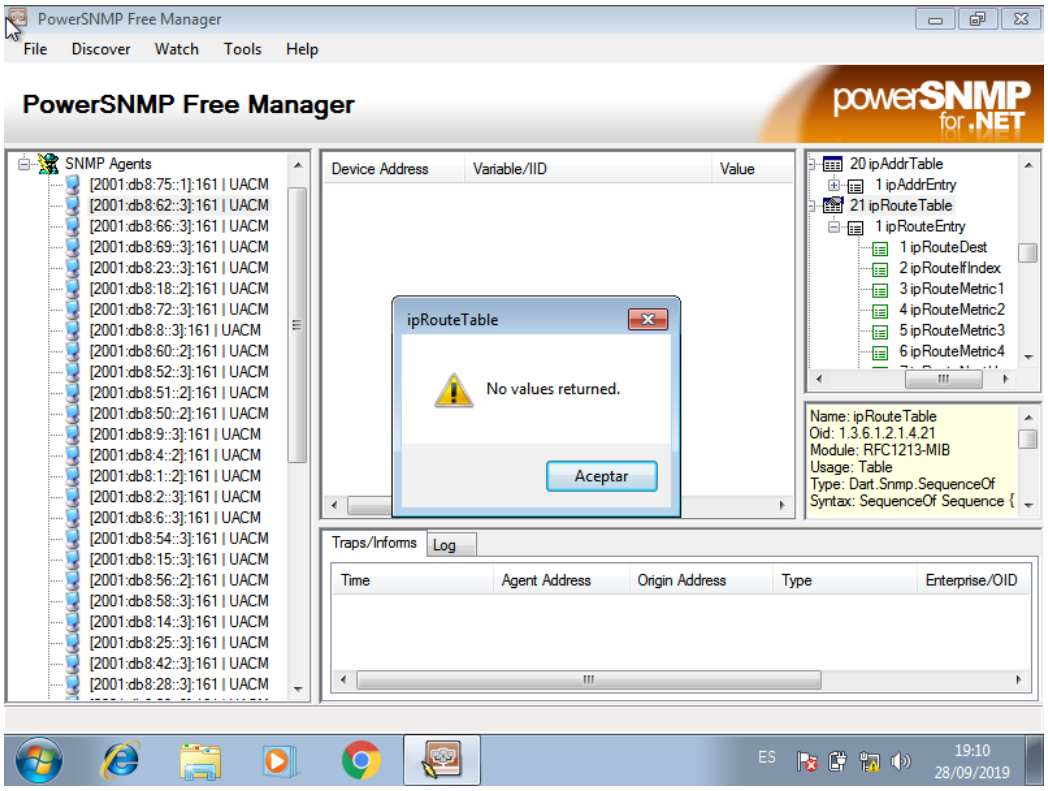


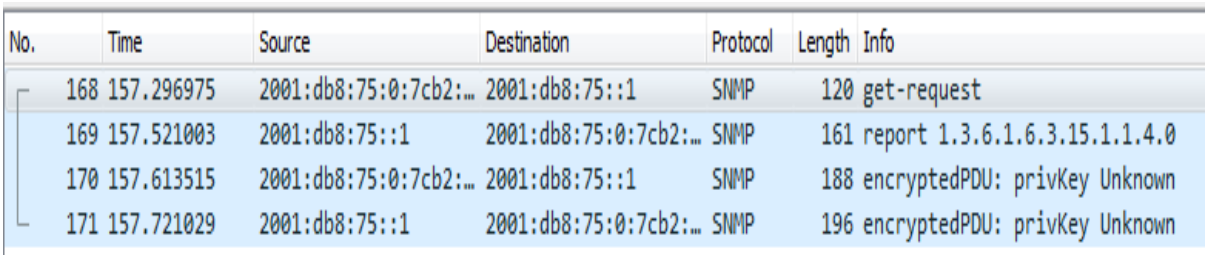
Figura 93. Mensaje que se obtuvó al solicitar la tabla de enrutamiento.

Sin embargo, cuando enviamos la solicitud de la tabla de enrutamiento, se género un mensaje de error de no valores de regreso. Esto sucedió porque sólo enviamos un mensaje de “get”, este mensaje es para pedir información que no sea muy grande, es decir, un solo valor. En realidad debimos enviar un mensaje “get bulk”, ya que, este mensaje es utilizado cuando se requiere traer una larga transmisión de datos. Desgraciadamente el software *powerSNMP* no nos permitió enviar mensajes *get bulk*.

4.2.2 Análisis de los mensajes de Gestión de la Red Avanzada GEANT en la emulación

En esta sección expondremos el monitoreo que realizamos a la interface de red de Alemania_2. *Wireshark* detecta cada mensaje que pasa por la interfaz, observamos todos los mensajes que se generaron como se muestra en la figura 94. En esta se muestra el mensaje *get request*, el cual es el primer mensaje que

envía SNMP y como se explicó en capítulos anteriores este mensaje nos indica un éxito o fracaso a una petición.



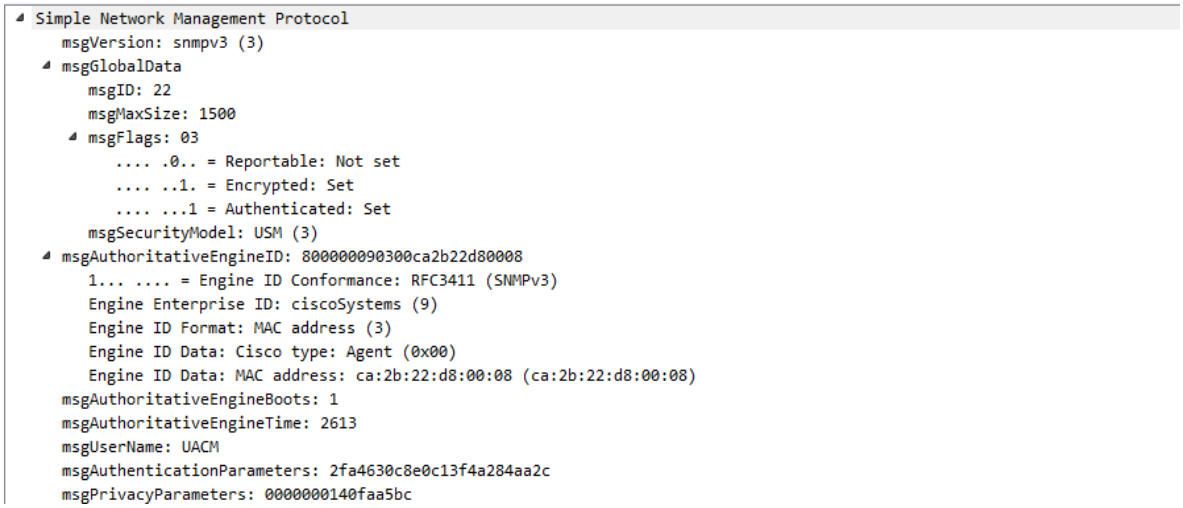
No.	Time	Source	Destination	Protocol	Length	Info
168	157.296975	2001:db8:75:0:7cb2:…	2001:db8:75::1	SNMP	120	get-request
169	157.521003	2001:db8:75::1	2001:db8:75:0:7cb2:…	SNMP	161	report 1.3.6.1.6.3.15.1.1.4.0
170	157.613515	2001:db8:75:0:7cb2:…	2001:db8:75::1	SNMP	188	encryptedPDU: privKey Unknown
171	157.721029	2001:db8:75::1	2001:db8:75:0:7cb2:…	SNMP	196	encryptedPDU: privKey Unknown

Figura 94. Mensajes obtenidos al monitorear al *router* Alemania_2.

Analizando el paquete de SNMP más a detalle con el software *Wireshark*, nos desglosa información como son:

- *msgVersion*. El tipo de versión de SNMP, que en este caso es la versión 3.
- *msgMaxSize*. El tamaño máximo del mensaje 1500 bits.
- *msgID*. El identificador del mensaje 22.
- *MsgsecurityModel*. En este caso está usando USM (*User-Based Security Model*), el cual proporciona los servicios de autentificación y privacidad en SNMPV3.
- *MsgUserName*. Este nos indica el nombre que se le asignó como usuario al configurar el comando SNMPV3 en los *router*, que en este caso es UACM.

Esta información se muestra en la figura 95.



```
▲ Simple Network Management Protocol
    msgVersion: snmpv3 (3)
  ▲ msgGlobalData
      msgID: 22
      msgMaxSize: 1500
    ▲ msgFlags: 03
        .... .0.. = Reportable: Not set
        .... ..1. = Encrypted: Set
        .... ...1 = Authenticated: Set
      msgSecurityModel: USM (3)
  ▲ msgAuthoritativeEngineID: 800000090300ca2b22d80008
      1... .... = Engine ID Conformance: RFC3411 (SNMPv3)
      Engine Enterprise ID: ciscoSystems (9)
      Engine ID Format: MAC address (3)
      Engine ID Data: Cisco type: Agent (0x00)
      Engine ID Data: MAC address: ca:2b:22:d8:00:08 (ca:2b:22:d8:00:08)
    msgAuthoritativeEngineBoots: 1
    msgAuthoritativeEngineTime: 2613
    msgUserName: UACM
    msgAuthenticationParameters: 2fa4630c8e0c13f4a284aa2c
    msgPrivacyParameters: 0000000140faa5bc
```

Figura 95. Desglose de información del mensaje *GETRequests* en SNMPV3.

Toda la información desglosada del mensaje *get-request* a través del software *Wireshark*, es de utilidad para comprobar que concuerde con la misma información con la que se configuro vía CLI a los *routers*. Así como analizar los parámetros de la literatura explicada en el capítulo 2.

4.2.3 Análisis del paquete *Get-Request* con la herramienta *Flow Graph*

Como sabemos del capítulo 2, el protocolo SNMP permite administrar los dispositivos de red y diagnosticar sus problemas, en la figura 96 se observa el análisis de este protocolo a través de *Flow Graph*.

En la figura 96 se puede observar que a los 58 segundos es enviado un mensaje *get request* por el puerto 161, que es el puerto por donde el agente recibe solicitudes, desgraciadamente no se obtuvo alguna respuesta del agente. A los 291.37 segundos, se realizó una petición al *router* con dirección IPV6 2001:db8:56::2, este a su vez envía una respuesta la cual indica el éxito o el fracaso de la petición a los 291.45 segundos. Y lo hace a través de una OID 1.3.6.1.6.3.15.1.1.4.0, la cual muestra que la petición fue correcta, indicando la respuesta a la petición solicitada.

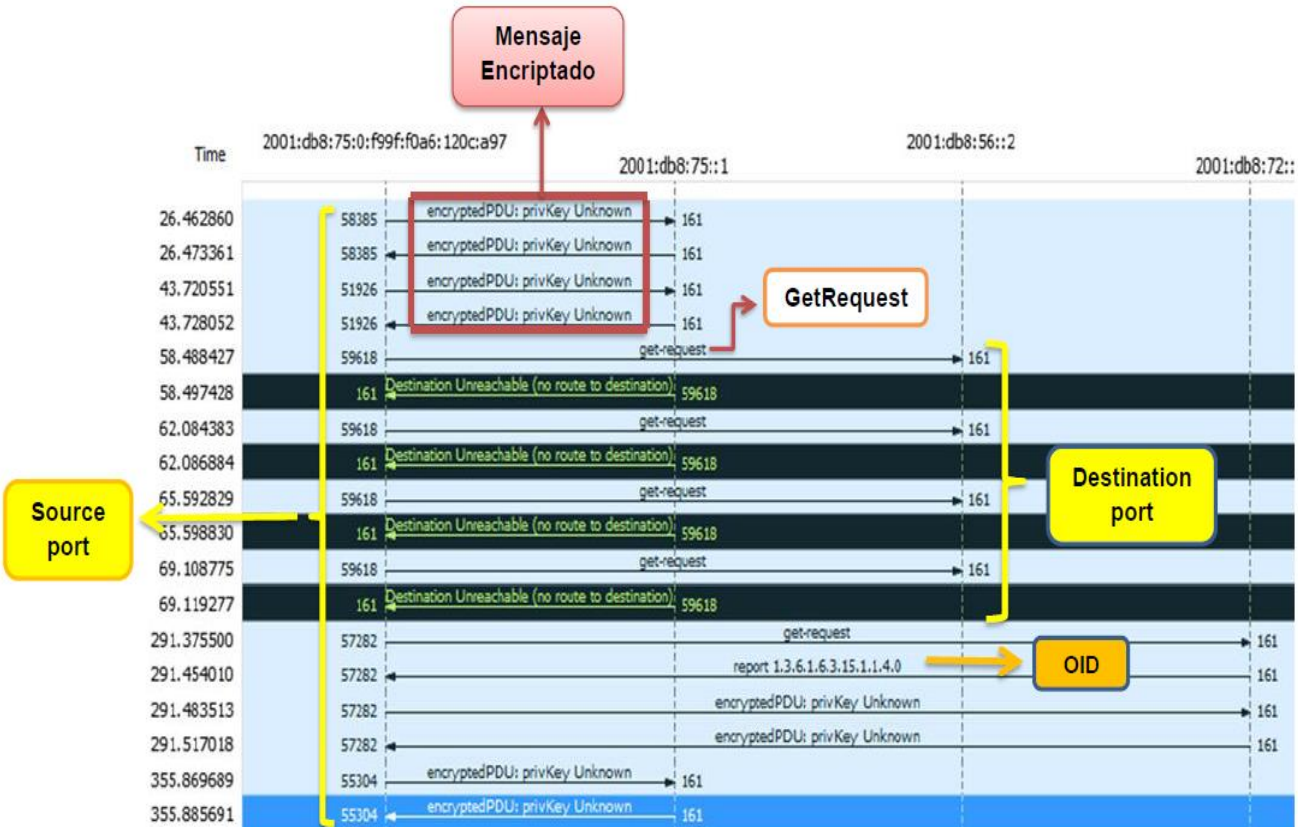


Figura 96. Análisis del paquete *Get Request* con la herramienta *Flow Graph*.

4.3 Análisis del Rendimiento de la máquina real

Para la emulación, se trabajó con 51 elementos virtuales, tal que 45 fueron *routers*, 5 fueron *hots* virtuales y una máquina virtual para el NMS.

4.3.1 Análisis de rendimiento de la máquina real en pruebas de conectividad

En relación a la conectividad y funcionando toda nuestra red. Se usó el 100 % del CPU y el 74 % de memoria RAM. Cabe recalcar que para levantar todas las máquinas virtuales requeridas la emulación tomó alrededor de 25 minutos.

En la figura 97 se observa el rendimiento, tanto del CPU como de la memoria RAM al realizar pruebas de conectividad con la emulación de red avanzada GEANT.

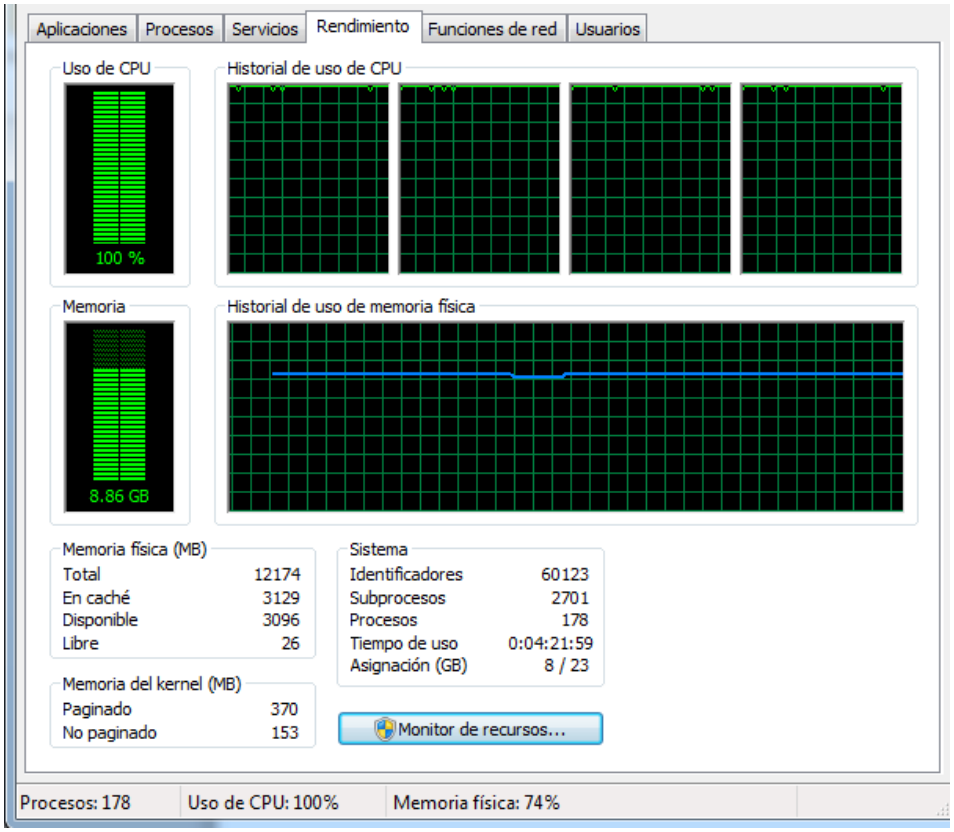


Figura 97. Rendimiento del CPU y memoria RAM en pruebas de conectividad.

4.3.2 Análisis de rendimiento en la máquina real en pruebas de gestión

En términos de gestión y funcionando toda la red, se ocupó menos recursos respecto a la prueba de conectividad. Se usó el 99 % del CPU y el 51 % de memoria RAM.

En la figura 98 se observa el rendimiento, tanto del CPU como de la memoria RAM al realizar pruebas de gestión con la emulación de red avanzada GEANT.

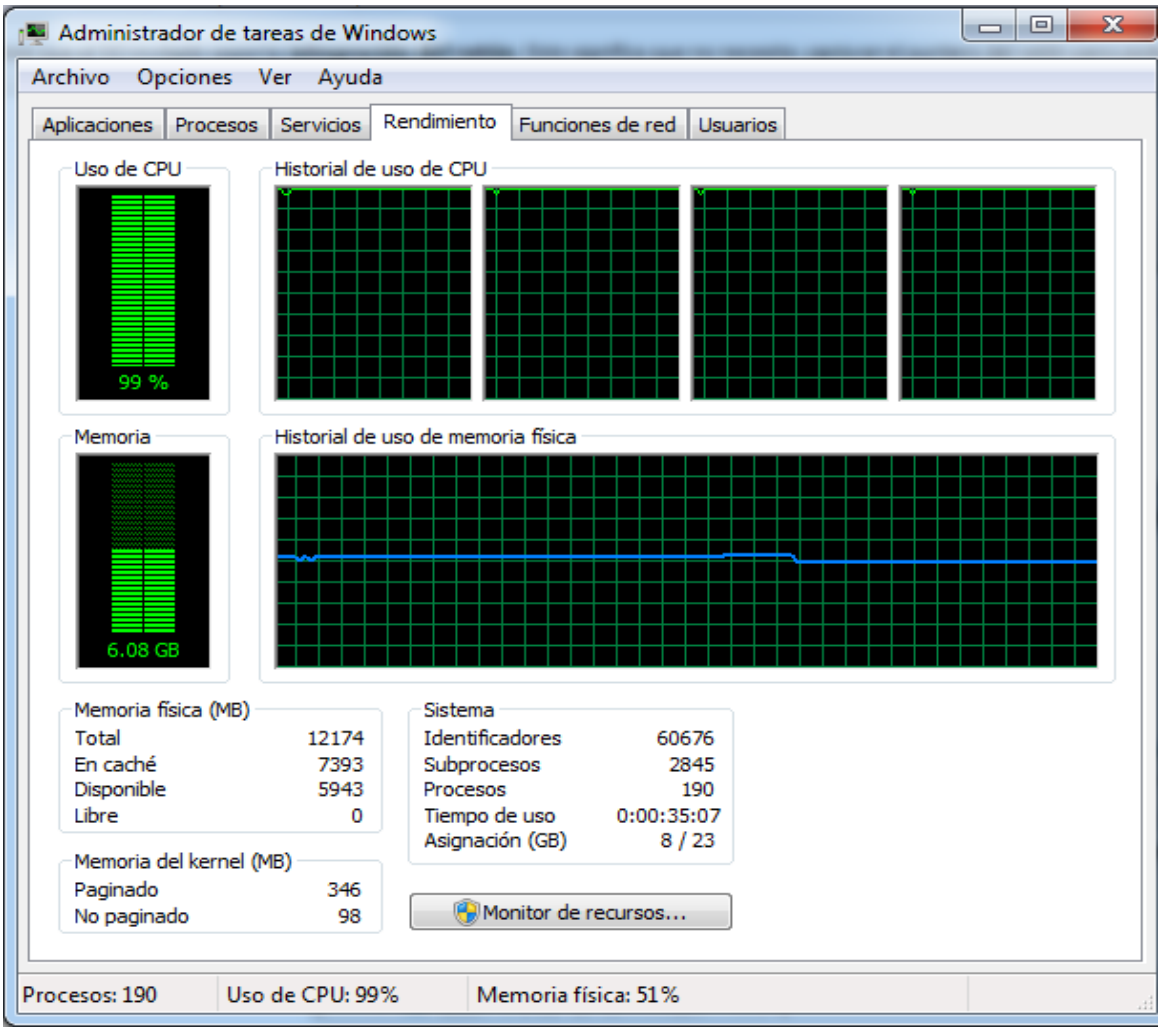


Figura 97. Rendimiento del CPU y memoria RAM en pruebas de gestión.

4.4 CONCLUSIONES

Como conclusión general, trabajar en la presente tesis con IPV6 fue un reto, debido a que no tuvimos en nuestra educación universitaria alguna clase respecto a este tema, a pesar de ello se implementó bastante bien y funcionó a la perfección con la red avanzada GEANT.

Al comenzar esta tesis una de las hipótesis que nos planteamos fue la de conocer qué tanto se acerca el emulador GNS3 a la red real GEANT, así como, medir el desempeño del emulador en la conectividad y gestión. Respecto a esto, y en términos de desempeño, se trabajó al 100%, con una computadora de alto rendimiento, es decir, con suficiente memoria RAM, mayor o igual a los 12 GB, y un procesador mayor o igual a un Core i5 3340M 2.70 GHz. El emulador trabajó con un rendimiento óptimo con los protocolos OSPFV3 y SNMPV3, así como con, IOS reales en los *routers*. Esto fue de gran ayuda porque se trató de igualar el funcionamiento de un *router* real, con todas sus características y funcionalidades. Se logró configurar SNMPV3 con autentificación y cifrado.

4.4.1 Pruebas de OSPFV3 en la emulación

En la emulación, la configuración del protocolo OSPFV3 se realizó de manera satisfactoria con el protocolo IPV6. En esta emulación se utilizaron *routers* Cisco 7200, los cuales cuentan con 6 interfaces POS, también tiene la ventaja que se les pueden agregar más slot para utilizar otras interfaces como son *GigaEthernet, FastEthernet* y *Ethernet.*

En términos de conectividad con OPSFV3 en la emulación, se hicieron pruebas de *pings* entre VPCS, *routers* y pruebas a la máquina virtual, logrando una conectividad y una eficiencia del 100 % en cada una de ellas. También se utilizó el programa *Wireshark* para el monitoreo y seguimiento de los paquetes.

Otro punto importante fue el obtener la tabla de enrutamiento de un r*outer*, comprobando efectivamente que los datos obtenidos fueron exactamente iguales a los que nos indica la teoría respecto al tema de OSPFV3, con esto se pudo

comprobar la métrica, las redes IPV6 y las interfaces a través de las cuales estaban conectadas los *routers* para cerciorarnos que efectivamente fueran las que se habían configurado.

La herramienta *Flow Graph* de *Wireshark*, nos fue de gran utilidad para comprobar conexiones entre los *routers* en nuestra red, tiempo de espera y a través de qué puertos trabajan.

Se utilizó el programa *VirtualBox* para la máquina virtual, la máquina virtual cuenta con sistema operativo Windows 7, memoria RAM de 1 GB y fue conectada al *router* de Alemania_2, trabajando satisfactoriamente con el protocolo OSPFV3.

4.4.2 Pruebas de gestión con SNMPV3 en la emulación

En la emulación, la configuración del protocolo SNMPV3 se realizó de manera satisfactoria con el protocolo IPV6. Se utilizó el programa *VirtualBox* para crear una máquina virtual para gestionar nuestra red GEANT, con ayuda del software *powerSNMP Free Manager*.

El programa *powerSNMP*, nos permitió monitorear y gestionar nuestra emulación de GEANT, así como se logró implementar autentificación y cifrado en cada *router*.

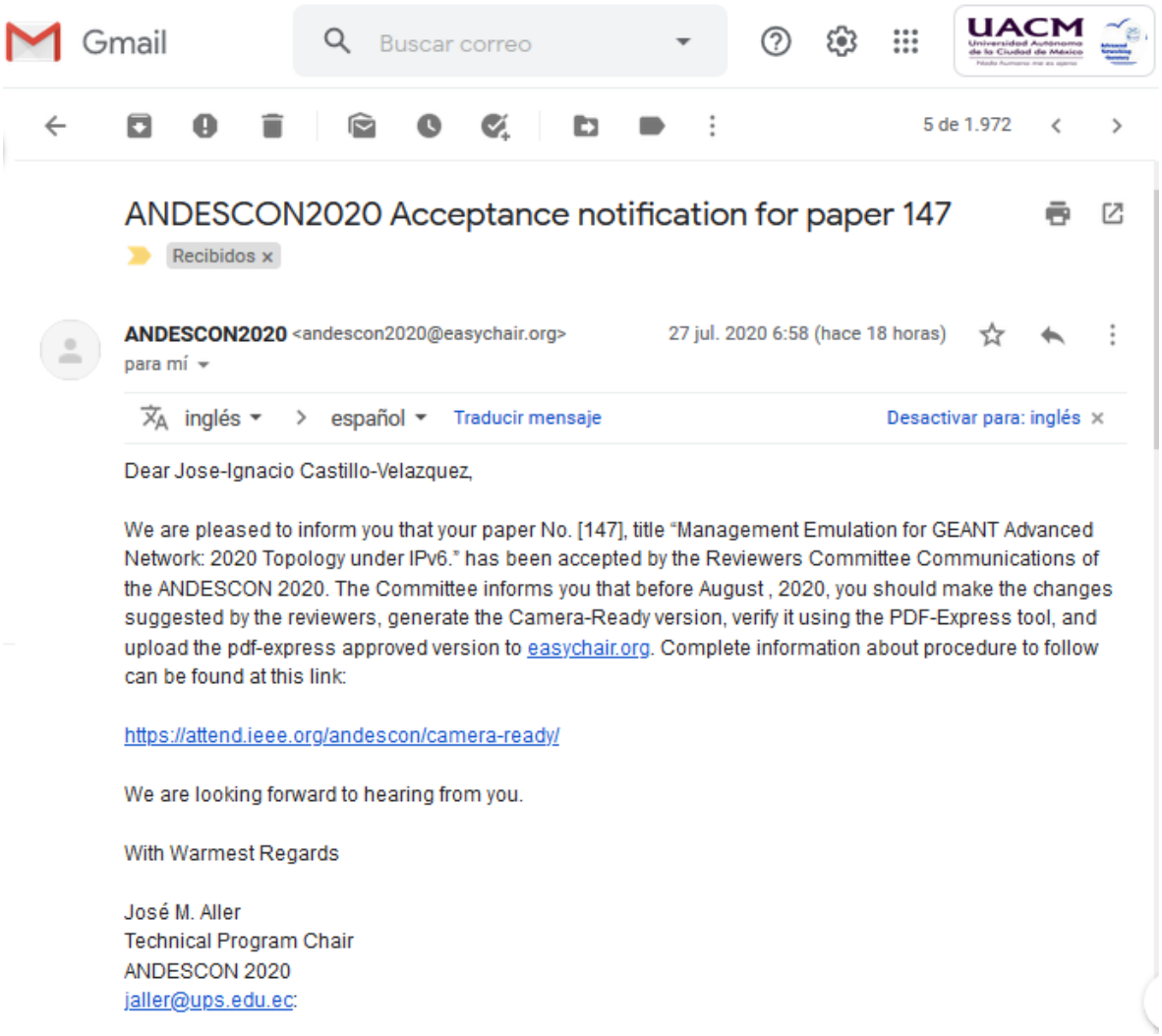
Concluimos que SNMP es un protocolo bastante funcional y fácil de implementar en los agentes y es debido a que su configuración es muy sencilla. Las limitantes las tuvimos con el software *PowerSNMP*, debido a que se instaló su versión gratuita. Desgraciadamente el programa *PowerSNMP*, sólo nos permitió utilizar mensajes del tipo “get” y no pudimos utilizar los mensajes del tipo “set” y “get bulk”, los cuales nos permiten modificar valores de objetos y para poder traer gran cantidad de información de un agente.

Aun así, consideramos que es un software muy útil y recomendable para monitorear en cualquier momento y en cualquier lugar a un *router,* basta con contar con una computadora y conectarse a un *router* que se encuentre dentro de la misma red previamente configurada con el protocolo SNMP, para comenzar con la gestión de la misma.

Igualmente se trabajó con la herramienta *Flow Graph* de *Wireshark*, logrando observar el proceso que realizan la estación de gestión y un agente, al realizar una petición y la obtención de la respuesta del agente. Así como, pudimos demostrar que efectivamente el protocolo SNMPV3 trabaja a través del puerto 161 como lo indica la teoría.

Como conclusión final y después de haber estudiado e implementado los protocolos de enrutamiento y de gestión, adquirimos habilidades y un mayor conocimiento en redes avanzadas, a fin de que en un futuro seamos administradores de red en un ISP (*Internet Service Provider*).

4.4.3 Logros adicionales a la tesis: Publicación indexada a SCOPUS



Gracias a el trabajo en conjunto con el M. en C. José Ignacio Castillo Velázquez se trabajó un *paper* donde se obtuvo, el premio al mejor artículo de investigación del congreso IEEE ANDESCON 2020, entre los cuales participaron 175 artículos, con 38 países participantes.

CERTIFICATE

IEEE Advancing Technology for Humanity

IEEE Andescon TECHNOLOGY AND INNOVATION FOR ANDEAN INDUSTRY

Presented to :

Jose-Ignacio Castillo-Velazquez, Isabel Muñoz-Martínez, Jorge-Armando Díaz-Ramírez & Esteban F. Ordoñez-Morales.

Authors of the research paper :

"Management Emulation for GEANT Advanced Network: 2020 Topology under IPv6"

Awarded in recognition as :

BEST PAPER : Communications Track

at the Technical and Scientific Conference of the Andean Council of the IEEE (ANDESCON 2020)

Awarded in Quito - Ecuador, October 13-16 , 2020

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC. FOUNDED NEW YORK 1884

Co-organized with

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

M.Sc. Mara Falconí
ANDESCON 2020 General Chair
IEEE Ecuador Section Chair

Dr. Carlos Lozano
ANDESCON 2020 General Chair
Andean Council President

ANEXO 1

El presente anexo contiene la lista completa de las interfaces conectadas, las cuales se describieron en el capítulo 3 correspondiente a la tabla 2.

País	Interface	Dirección IPV6	*Router* ID
Reino Unido a Francia1	pos3/0 a pos3/0	2001:db8:5::/64	1.1.1.1
Irlanda1 a ReinoUnido2	G0/0 a G0/0	2001:db8:6::/64	5.5.5.5
Reino Unido1 a Irlanda2	G0/0 a G0/0	2001:db8:6::/64	5.5.5.5
Irlanda1 a Reino Unido1	F1/1 a E1/0	2001:db8:7::/64	5.5.5.5
Reino Unido1 a Irlanda1	E1/0 a F1/1	2001:db8:7::/64	5.5.5.5
Suiza a Francia2	pos2/0 a pos3/0	2001:db8:8::/64	7.7.7.7
Suiza a Francia1	pos3/0 a pos4/0	2001:db8:9::/64	7.7.7.7
Francia2 a Suiza	pos3/0 a pos2/0	2001:db8:8::/64	6.6.6.6
Francia2 a España	pos2/0 a pos4/0	2001:db8:4::/64	6.6.6.6
Francia2 a Italia	pos4/0 a pos1/0	2001:db8:10::/64	6.6.6.6
Suiza a Italia	pos4/0 a pos2/0	2001:db8:11::/64	7.7.7.7
Italia a Suiza	pos2/0 a pos4/0	2001:db8:11::/64	8.8.8.8
Italia a Francia2	pos1/0 a pos4/0	2001:db8:10::/64	8.8.8.8
Reino Unido a Bélgica	pos4/0 a pos2/0	2001:db8:12::/64	1.1.1.1

Tabla 2. Conexiones de la red GEANT (parte 2).

País	Interface	Dirección IPV6	*Router ID*
Reino Unido a Islandia	pos6/0 a pos2/0	2001:db8:14::/64	1.1.1.1
Islandia a Reino Unido	pos2/0 a pos6/0	2001:db8:14::/64	11.11.11.11
Francia1 a Suiza	pos4/0 a pos3/0	2001:db8:9::/64	2.2.2.2
Italia a Malta	pos3/0 a pos2/0	2001:db8:15::/64	8.8.8.8
Malta a Italia	pos2/0 a pos3/0	2001:db8:15::/64	12.12.12.12
Italia a Grecia	pos4/0 a pos1/0	2001:db8:16::/64	8.8.8.8
Grecia a Italia	pos1/0 a pos4/0	2001:db8:16::/64	13.13.13.13
Italia a Albania	pos5/0 a pos2/0	2001:db8:17::/64	8.8.8.8
Albania a Italia	pos2/0 a pos5/0	2001:db8:17::/64	14.14.14.14
Italia a Austria	pos6/0 a pos1/0	2001:db8:18::/64	8.8.8.8
Austria a Italia	pos1/0 a pos6/0	2001:db8:18::/64	15.15.15.15
Austria a Grecia	E2/0 a G0/0	2001:db8:19::/64	15.15.15.15
Grecia a Austria	G0/0 a E2/0	2001:db8:19::/64	13.13.13.13
Grecia a Chipre	pos2/0 a pos2/0	2001:db8:20::/64	13.13.13.13
Chipre a Grecia	pos2/0 a pos2/0	2001:db8:20::/64	16.16.16.16
Reino Unido a Chipre	E1/1 a F1/0	2001:db8:21::/64	5.5.5.5
Chipre a Reino Unido	F1/0 a E1/1	2001:db8:21::/64	16.16.16.16
Reino Unido a Israel	E1/2 a F1/0	2001:db8:22::/64	1.1.1.1

Tabla 2. Conexiones de la red GEANT (parte 3).

País	Interface	Dirección IPV6	*Router ID*
Israel a Reino Unido	F1/0 a E1/2	2001:db8:22::/64	17.17.17.17
Austria a Eslovenia	pos3/0 a pos2/0	2001:db8:23::/64	15.15.15.15
Eslovenia a Austria	pos2/0 a pos3/0	2001:db8:23::/64	18.18.18.18
Eslovenia a Hungría	pos3/0 a pos2/0	2001:db8:24::/64	18.18.18.18
Hungría a Eslovenia	pos2/0 a pos3/0	2001:db8:24::/64	19.19.19.19
Austria a Croacia	pos4/0 a pos2/0	2001:db8:25::/64	15.15.15.15
Croacia a Austria	pos2/0 a pos4/0	2001:db8:25::/64	20.20.20.20
Croacia a Hungría	pos3/0 a pos3/0	2001:db8:26::/64	20.20.20.20
Hungría a Croacia	pos3/0 a pos3/0	2001:db8:26::/64	19.19.19.19
Austria a Hungría	E2/1 a E1/0	2001:db8:27::/64	15.15.15.15
Hungría a Austria	E1/0 a E2/1	2001:db8:27::/64	19.19.19.19
Hungría a Servía	pos4/0 a pos2/0	2001:db8:28::/64	19.19.19.19
Servía a Hungría	pos2/0 a pos4/0	2001:db8:28::/64	21.21.21.21
Hungría a Eslovaquia	E1/3 a F1/0	2001:db8:39::/64	19.19.19.19
Austria a Ucrania	G0/0 a G0/0	2001:db8:40::/64	15.15.15.15
Ucrania a Austria	G0/0 a G0/0	2001:db8:40::/64	30.30.30.30
Austria a Eslovaquia	E2/2 a G0/0	2001:db8:38::/64	15.15.15.15

Tabla 2. Conexiones de la red GEANT (parte 4).

País	Interface	Dirección IPV6	*Router ID*
Hungría a Montenegro	pos5/0 a pos2/0	2001:db8:29::/64	19.19.19.19
Montenegro a Hungría	pos2/0 a pos5/0	2001:db8:29::/64	22.22.22.22
Austria a Rep. de Macedonia	pos5/0 a pos1/0	2001:db8:30::/64	15.15.15.15
Rep. de Macedonia a Austria	pos1/0 a pos5/0	2001:db8:30::/64	23.23.23.23
Rep. de Macedonia a Bulgaria	pos2/0 a pos2/0	2001:db8:31::/64	23.23.23.23
Bulgaria a Rep. de Macedonia	pos2/0 a pos2/0	2001:db8:31::/64	24.24.24.24
Austria a Bulgaria	pos6/0 a pos1/0	2001:db8:32::/64	15.15.15.15
Bulgaria a Austria	pos1/0 a pos6/0	2001:db8:32::/64	24.24.24.24
Bulgaria a Rumanía	pos3/0 a pos2/0	2001:db8:33::/64	24.24.24.24
Rumanía a Bulgaria	pos2/0 a pos3/0	2001:db8:33::/64	25.25.25.25
Hungría a Rumanía	pos6/0 a pos3/0	2001:db8:34::/64	19.19.19.19
Rumanía a Hungría	pos3/0 a pos6/0	2001:db8:34::/64	25.25.25.25
Rumanía a Moldavia	pos4/0 a pos2/0	2001:db8:35::/64	25.25.25.25
Moldavia a Rumanía	pos2/0 a pos4/0	2001:db8:35::/64	26.26.26.26
Hungría a Turquía	E1/1 a F1/0	2001:db8:36::/64	19.19.19.19
Turquía a Hungría	F1/0 a E1/1	2001:db8:36::/64	27.27.27.27
Hungría a Georgia	E1/2 a F1/0	2001:db8:37::/64	19.19.19.19

Tabla 2. Conexiones de la red GEANT (parte 5).

País	Interface	Dirección IPV6	*Router ID*
Austria a Rep. Checa	E2/3 a G0/0	2001:db8:41::/64	15.15.15.15
Rep. Checa a Austria	G0/0 a E2/3	2001:db8:41::/64	31.31.31.31
Rep. Checa a Hungría	G2/0 a G0/0	2001:db8:42::/64	31.31.31.31
Hungría a Rep. Checa	G0/0 a G2/0	2001:db8:42::/64	19.19.19.19
Austria a Polonia	E2/4 a F1/0	2001:db8:43::/64	15.15.15.15
Polonia a Austria	F1/0 a E2/4	2001:db8:43::/64	32.32.32.32
Polonia a Bielorrusia	pos2/0 a pos2/0	2001:db8:44::/64	32.32.32.32
Bielorrusia a Polonia	pos2/0 a pos2/0	2001:db8:44::/64	33.33.33.33
Polonia a Lituania	pos3/0 a pos2/0	2001:db8:45::/64	32.32.32.32
Lituania a Polonia	pos2/0 a pos3/0	2001:db8:45::/64	34.34.34.34
Lituania a Letonia	pos3/0 a pos2/0	2001:db8:46::/64	34.34.34.34
Letonia a Lituania	pos2/0 a pos3/0	2001:db8:46::/64	35.35.35.35
Letonia a Estonia	pos3/0 a pos2/0	2001:db8:47::/64	35.35.35.35
Estonia a Letonia	pos2/0 a pos3/0	2001:db8:47::/64	36.36.36.36
Estonia a Alemania1	pos3/0 a pos2/0	2001:db8:48::/64	36.36.36.36
Alemania1 a Estonia	pos2/0 a pos3/0	2001:db8:48::/64	37.37.37.37
Alemania1 a Dinamarca	pos3/0 a pos2/0	2001:db8:49::/64	37.37.37.37

Tabla 2. Conexiones de la red GEANT (parte 6).

País	Interface	Dirección IPV6	*Router ID*
Dinamarca a Alemania1	pos2/0 a pos3/0	2001:db8:49::/64	38.38.38.38
Dinamarca a Holanda	pos3/0 a pos2/0	2001:db8:50::/64	38.38.38.38
Holanda a Dinamarca	pos2/0 a pos3/0	2001:db8:50::/64	10.10.10.10
Holanda a Alemania1	pos3/0 a pos4/0	2001:db8:51::/64	10.10.10.10
Alemania1 a Holanda	pos4/0 a pos3/0	2001:db8:51::/64	37.37.37.37
Holanda a Bélgica	pos4/0 a pos3/0	2001:db8:52::/64	10.10.10.10
Bélgica a Holanda	pos3/0 a pos4/0	2001:db8:52::/64	9.9.9.9
Dinamarca a Islandia	pos4/0 a pos3/0	2001:db8:53::/64	38.38.38.38
Islandia a Dinamarca	pos3/0 a pos4/0	2001:db8:53::/64	11.11.11.11
Dinamarca a Noruega	pos5/0 a pos2/0	2001:db8:54::/64	38.38.38.38
Alema.2 a Rep. Checa	E1/0 a E3/0	2001:db8:66::/64	43.43.43.43
Noruega a Dinamarca	pos2/0 a pos5/0	2001:db8:54::/64	39.39.39.39
Dinamarca a Suecia	pos6/0 a pos2/0	2001:db8:55::/64	38.38.38.38
Suecia a Dinamarca	pos2/0 a pos6/0	2001:db8:55::/64	40.40.40.40
Suecia a Noruega	pos3/0 a pos3/0	2001:db8:56::/64	40.40.40.40
Noruega a Suecia	pos3/0 a pos3/0	2001:db8:56::/64	39.39.39.39
Suecia a Filandia1	pos4/0 a pos2/0	2001:db8:57::/64	40.40.40.40

Tabla 2. Conexiones de la red GEANT (parte 7).

País	Interface	Dirección IPV6	*Router ID*
Finlandia a Suecia1	pos2/0 a pos4/0	2001:db8:57::/64	41.41.41.41
Suecia a Filandia2	pos5/0 a pos3/0	2001:db8:58::/64	40.40.40.40
Finlandia a Suecia2	pos3/0 a pos5/0	2001:db8:58::/64	41.41.41.41
Holanda a Luxemburgo	pos5/0 a pos 2/0	2001:db8:59::/64	10.10.10.10
Luxemburgo a Holanda	pos2/0 a pos5/0	2001:db8:59::/64	42.42.42.42
Luxemburgo a Alem.2	pos3/0 a pos3/0	2001:db8:60::/64	42.42.42.42
Alem.2 a Luxemburgo	pos3/0 a pos3/0	2001:db8:60::/64	43.43.43.43
Alemania2 a Holanda	pos4/0 a pos6/0	2001:db8:61::/64	43.43.43.43
Holanda a Alemania2	pos6/0 a pos4/0	2001:db8:61::/64	10.10.10.10
Alemania2 a Alemania1	pos5/0 a pos5/0	2001:db8:62::/64	43.43.43.43
Alemania1 a Alemania2	pos5/0 a pos5/0	2001:db8:62::/64	37.37.37.37
Alemania2 a Polonia	pos6/0 a pos4/0	2001:db8:63::/64	43.43.43.43
Polonia a Alemania2	pos4/0 a pos6/0	2001:db8:63::/64	32.32.32.32
Alemania2 a Georgia	F0/0 a F1/1	2001:db8:64::/64	43.43.43.43
Georgia a Alemania2	F1/1 a F0/0	2001:db8:64::/64	28.28.28.28
Alemania2 a Chipre	F0/1 a F1/1	2001:db8:65::/64	43.43.43.43
Chipre a Alemania2	F1/1 a F0/1	2001:db8:65::/64	16.16.16.16

Tabla 2. Conexiones de la red GEANT (parte 8).

País	Interface	Dirección IPV6	*Router ID*
Alemania2 a Hungría	E1/1 a E1/4	2001:db8:67::/64	43.43.43.43
Hungría a Alemania2	E1/4 a E1/1	2001:db8:67::/64	19.19.19.19
Alem.2 a Azerbaiyán	E1/2 a E2/0	2001:db8:68::/64	43.43.43.43
Azerbaiyán a Alem.2	E2/0 a E1/2	2001:db8:68::/64	44.44.44.44
Alemania2 a Austria	E1/3 a E2/5	2001:db8:69::/64	43.43.43.43
Austria a Alemania2	E2/5 a E1/3	2001:db8:69::/64	15.15.15.15
Alemania2 a Turquía	E1/4 a F1/1	2001:db8:70::/64	43.43.43.43
Turquía a Alemania2	F1/1 a E1/4	2001:db8:70::/64	27.27.27.27
Alemania2 a Israel	E1/5 a F1/1	2001:db8:71::/64	43.43.43.43
Israel a Alemania2	F1/1 a E1/5	2001:db8:71::/64	17.17.17.17
Alemania2 a Suiza	E1/6 a F1/0	2001:db8:72::/64	43.43.43.43
Suiza a Alemania2	F1/0 a E1/6	2001:db8:72::/64	7.7.7.7
Alemania2 a Armenia	E1/7 a E2/0	2001:db8:73::/64	43.43.43.43
Armenia a Alemania2	E2/0 a E1/7	2001:db8:73::/64	45.45.45.45
Rep. Checa a Alem.2	E3/0 a E1/0	2001:db8:66::/64	31.31.31.31
Georgia a Hungría	F1/0 a E1/2	2001:db8:37::/64	28.28.28.28

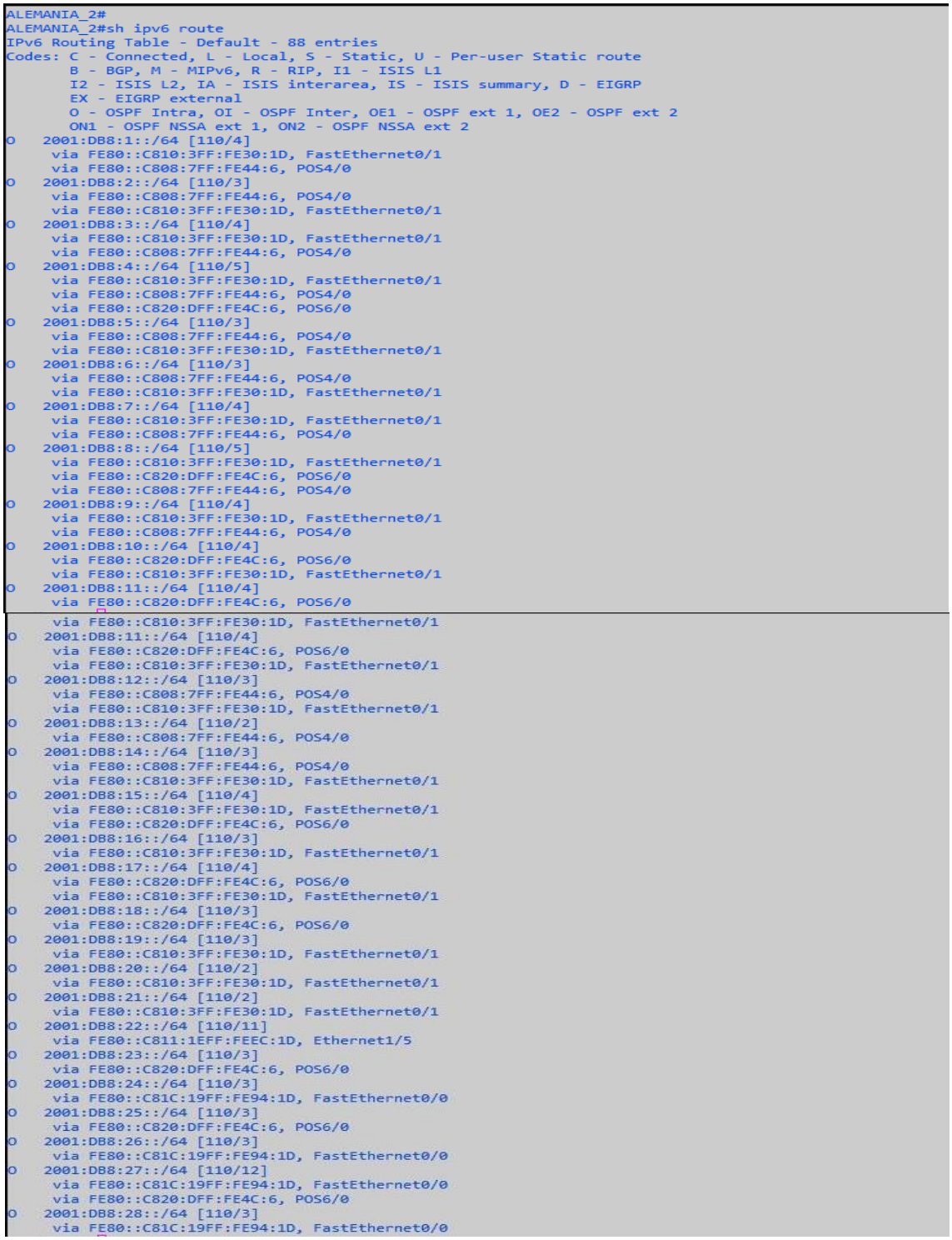
Tabla 2. Conexiones de la red GEANT (parte 9).

País	interface	Dirección IPV6	*Router ID*
Eslovaquia a Austria	G0/0 a E2/2	2001:db8:38::/64	29.29.29.29
Eslovaquia a Hungría	F1/0 a E1/3	2001:db8:39::/64	29.29.29.29
Bélgica a Reino Unido	pos2/0 a pos4/0	2001:db8:12::/64	9.9.9.9
Reino Unido a Holanda	pos5/0 a pos1/0	2001:db8:13::/64	1.1.1.1
Holanda a Reino Unido	pos1/0 a pos 5/0	2001:db8:13::/64	10.10.10.10

Tabla 2. Conexiones de la red GEANT (parte 10).

ANEXO 2.

En el anexo 2 se muestra la lista completa de la tabla de enrutamiento del *router* Alemania_2, la cual se describió en el capítulo 4 correspondiente a la figura 76.



```
ALEMANIA_2#
ALEMANIA_2#sh ipv6 route
IPv6 Routing Table - Default - 88 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O   2001:DB8:1::/64 [110/4]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
     via FE80::C808:7FF:FE44:6, POS4/0
O   2001:DB8:2::/64 [110/3]
     via FE80::C808:7FF:FE44:6, POS4/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:3::/64 [110/4]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
     via FE80::C808:7FF:FE44:6, POS4/0
O   2001:DB8:4::/64 [110/5]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
     via FE80::C808:7FF:FE44:6, POS4/0
     via FE80::C820:DFF:FE4C:6, POS6/0
O   2001:DB8:5::/64 [110/3]
     via FE80::C808:7FF:FE44:6, POS4/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:6::/64 [110/3]
     via FE80::C808:7FF:FE44:6, POS4/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:7::/64 [110/4]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
     via FE80::C808:7FF:FE44:6, POS4/0
O   2001:DB8:8::/64 [110/5]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
     via FE80::C820:DFF:FE4C:6, POS6/0
     via FE80::C808:7FF:FE44:6, POS4/0
O   2001:DB8:9::/64 [110/4]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
     via FE80::C808:7FF:FE44:6, POS4/0
O   2001:DB8:10::/64 [110/4]
     via FE80::C820:DFF:FE4C:6, POS6/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:11::/64 [110/4]
     via FE80::C820:DFF:FE4C:6, POS6/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:11::/64 [110/4]
     via FE80::C820:DFF:FE4C:6, POS6/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:12::/64 [110/3]
     via FE80::C808:7FF:FE44:6, POS4/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:13::/64 [110/2]
     via FE80::C808:7FF:FE44:6, POS4/0
O   2001:DB8:14::/64 [110/3]
     via FE80::C808:7FF:FE44:6, POS4/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:15::/64 [110/4]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
     via FE80::C820:DFF:FE4C:6, POS6/0
O   2001:DB8:16::/64 [110/3]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:17::/64 [110/4]
     via FE80::C820:DFF:FE4C:6, POS6/0
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:18::/64 [110/3]
     via FE80::C820:DFF:FE4C:6, POS6/0
O   2001:DB8:19::/64 [110/3]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:20::/64 [110/2]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:21::/64 [110/2]
     via FE80::C810:3FF:FE30:1D, FastEthernet0/1
O   2001:DB8:22::/64 [110/11]
     via FE80::C811:1EFF:FEEC:1D, Ethernet1/5
O   2001:DB8:23::/64 [110/3]
     via FE80::C820:DFF:FE4C:6, POS6/0
O   2001:DB8:24::/64 [110/3]
     via FE80::C81C:19FF:FE94:1D, FastEthernet0/0
O   2001:DB8:25::/64 [110/3]
     via FE80::C820:DFF:FE4C:6, POS6/0
O   2001:DB8:26::/64 [110/3]
     via FE80::C81C:19FF:FE94:1D, FastEthernet0/0
O   2001:DB8:27::/64 [110/12]
     via FE80::C81C:19FF:FE94:1D, FastEthernet0/0
     via FE80::C820:DFF:FE4C:6, POS6/0
O   2001:DB8:28::/64 [110/3]
     via FE80::C81C:19FF:FE94:1D, FastEthernet0/0
```

Figura 76. Tabla de enrutamiento de OSPFV3 del *router* ALEMANIA_2 (parte 2).

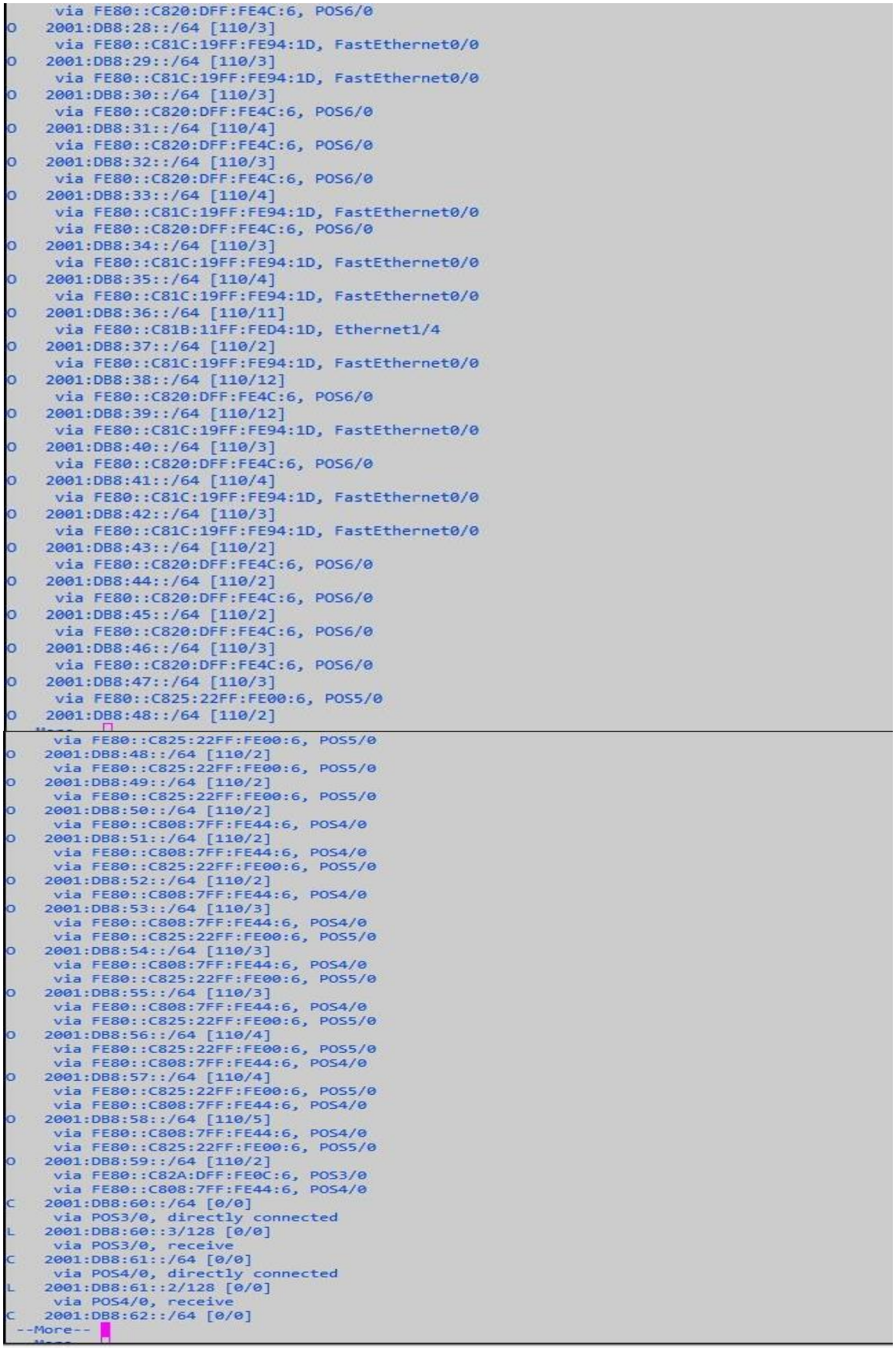


Figura 76. Tabla de enrutamiento de OSPFV3 del *router* ALEMANIA_2 (parte 3).

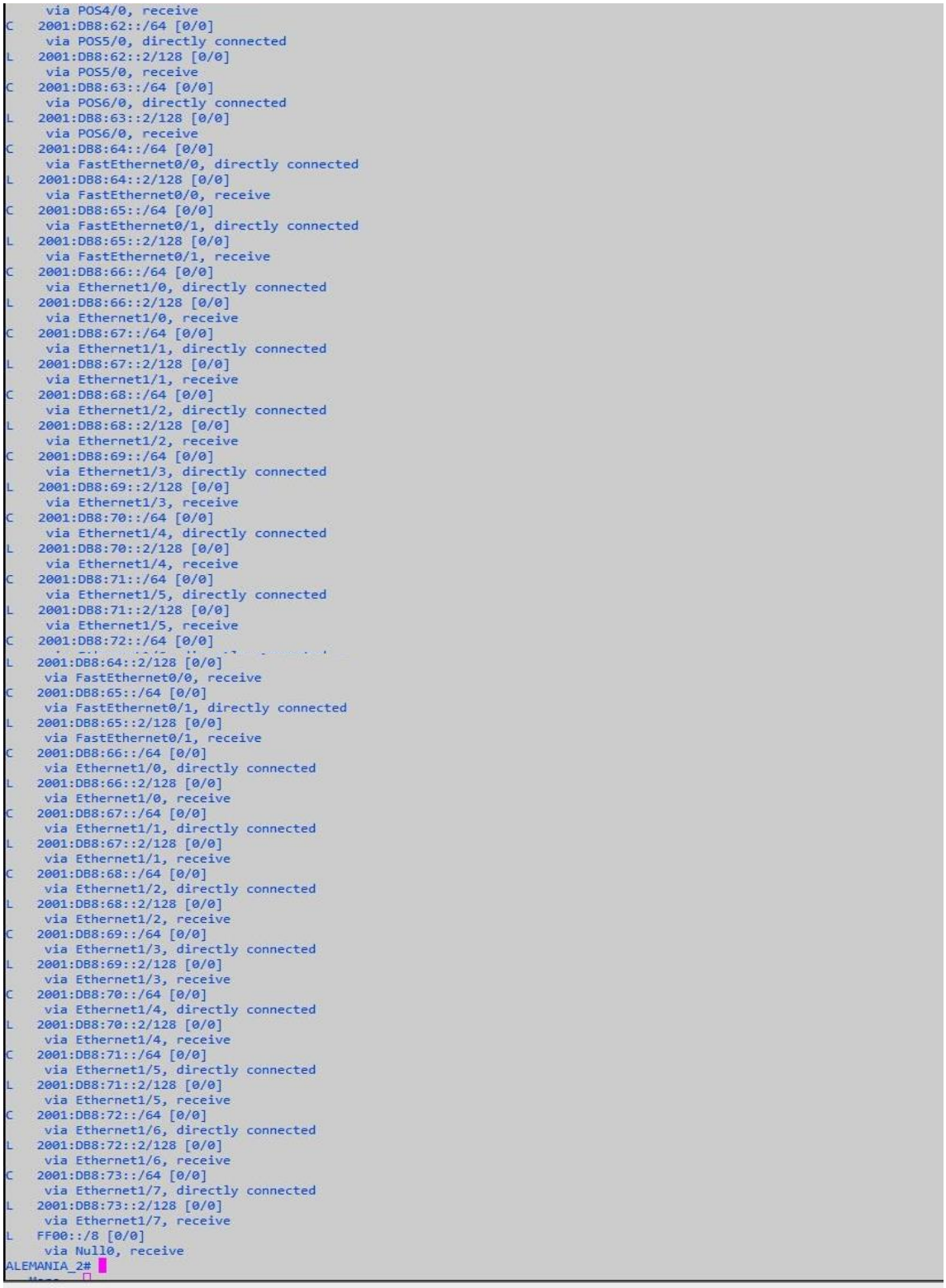


Figura 76. Tabla de enrutamiento del *router* OSPFV3 de ALEMANIA_2 (parte 4).

REFERENCIAS.

[1] Miguel A. Sanz, “RedIRIS-Fundamentos históricos de la Internet en Europa y en España”. Rediris.es, 2007. [En línea], disponible: http://www.rediris.es/difusion/publicaciones/boletin/45/enfoque2.html [Fecha de acceso: 13- diciembre- 2019].

[2] Oliver Martin, The “Hidden” Prehistory of European Research Networking, EEUU, 2012, ISBN: 978-1-4669-3872-4. [Fecha de acceso: 13- diciembre- 2019].

[3] Kaarina Lehtisalo. The History of Nordunet, [en línea], disponible: https://www.nordu.net/history/TheHistoryOfNordunet_simple.pdf [Fecha de acceso: 13- diciembre- 2019].

[4] Francois Fluckiger. The European researchers Network, CERN, Geneva 2000, [en línea], disponible: https://fluckiger.web.cern.ch/Fluckiger/Articles/F.Fluckiger-The_European_Researchers_Network.pdf [Fecha de acceso: 13- diciembre- 2019].

[5] The History of the EARN Network, [en línea], disponible: https://earn-history.net/ [Fecha de acceso: 13- diciembre- 2019].

[6] DANTE archive, Phare COSINE, [en línea], disponible: https://new-archive.dante.net/Backbones/Pages/Backbones.aspx [Fecha de acceso: 13- diciembre- 2019].

[7] Archive conneXions. Dante and Europanet # 4. The Interoperability Report, Vol 8 No 6, 1994. [en línea], disponible: https://new-archive.dante.net/Backbones/Documents/DiP04.pdf [Fecha de acceso: 13- diciembre- 2019].

[8] DANTE archive, The Europanet [en línea], disponible: https://new-archive.dante.net/Backbones/EuropaNET/Pages/EuropaNET.aspx [Fecha de acceso: 13- diciembre- 2019].

[9] DANTE archive, 20 years of DANTE, [en línea], disponible: https://dante.archive.geant.org/About_Us/20_years_of_DANTE/Pages/20_Years_of_Networking_Excellence.aspx [Fecha de acceso: 13- diciembre- 2019].

[10] DANTE archive, European Cooperation for Academic and Industrial Research Networking (Eureka Project 1061), [en línea], disponible: https://new-archive.dante.net/Backbones/EuroCAIRN/Pages/EuroCAIRN.aspx [Fecha de acceso: 13- diciembre- 2019].

[11] DANTE archive, The EuroCAIRN project, [en línea], disponible:
https://new-archive.dante.net/Backbones/Documents/DiP07.pdf
[Fecha de acceso: 13- diciembre- 2019].

[12] DANTE Archive, TEN-34 Network Maps, Archivo Digital, disponible:
https://new-archive.dante.net/Backbones/Documents/TEN-34maps.pdf
[Fecha de acceso: 13- diciembre- 2019].

[13] Dante Archive, European researchers Network Moves into the Fast Lane, [en línea], disponible:
http://archive.dante.net/Backbones/Documents/IC93-97-TWOD.pdf
[Fecha de acceso: 13- diciembre- 2019].

[14] DANTE Archive, QUANTUM, [en línea], disponible:
https://new-archive.dante.net/Backbones/QUANTUM/Pages/QUANTUM.aspx
[Fecha de acceso: 13- diciembre- 2019].

[15] Network Research Laboratory, University of Ciprus. Q-MED: Quality Network Tecnology for User-Oriented Multimedia in the Eastern Mediterranean Region. [en línea], disponible:
http://www.netrl.cs.ucy.ac.cy/index.php?option=com_content&task=view&id=56&Itemid=11
[Fecha de acceso: 13- diciembre- 2019].

[16] DANTE Archive, TEN – 155 [en línea], disponible:
https://new-archive.dante.net/Backbones/TEN-155/Pages/TEN-155.aspx
[Fecha de acceso: 13- diciembre- 2019].

[17] DANTE Archive, European researchers Network Moves into the Fast Lane. [en línea], disponible:
https://new-archive.dante.net/Backbones/Documents/TEN-155_FAQ9812.pdf
[Fecha de acceso: 13- diciembre- 2019].

[18] DANTE Archive, TEN – 155 sep 1998. [en línea], disponible:
https://new-archive.dante.net/Backbones/Documents/TEN-155-1998-09.pdf
[Fecha de acceso: 13- diciembre- 2019].

[19] DANTE Archive, TEN – 155 may 2001. [en línea], disponible:
http://www.gateway.nameflow.net/ten-155/ten155net.gif
[Fecha de acceso: 13- diciembre- 2019].

[20] La red informática europea GÉANT [en línea], disponible:
http://europa.eu/rapid/press-release_IP-08-354_es.pdf
[Fecha de acceso: 13- diciembre- 2019].

[21] GEANT, topology 2001, [en línea], disponible: https://geant3plus.archive.geant.net/Resources/Media_Library/Documents/topology_map_2001.pdf
[Fecha de acceso: 13- diciembre- 2019].

[22] Comisión Europea. La UE mejora la red paneuropea de investigación, [En línea], disponible: https://cordis.europa.eu/news/rcn/22563/es
[Fecha de acceso: 13- diciembre- 2019].

[23] GEANT Archive .Topology GEANT April 2004, [En línea], disponible: https://www.rediris.es/jt/jt2004/archivo/ficheros/JTRI2004-ylamilla-Cisco.pdf
[Fecha de acceso: 13- diciembre- 2019].

[24] ALICE Archive. [En línea], disponible: https://www.redclara.net/index.php/es/proyectos/red-e-infraestructura/alice
[Fecha de acceso: 13- diciembre- 2019].

[25] ALICE Archive. [En línea], disponible: http://www.interlab.ait.ac.th/tein2/Presentations/TEIN2-overview-kick-off.pdf
[Fecha de acceso: 13- diciembre- 2019].

[26] Redes académicas de alta velocidad y tecnología avanzada [en línea], disponible: https://dialnet.unirioja.es/descarga/articulo/4564570.pdf
[Fecha de acceso: 13- diciembre- 2019].

[27] RedCLARA, Topology 2008, [en línea], disponible: http://blogs.laprensagrafica.com/litoibarra/wpcontent/uploads/2008/12/topology_redclara_june20081.gif
[Fecha de acceso: 13- diciembre- 2019].

[28] GEANT2.Topology GEANT2, [en línea], disponible: https://www.redirisnova.es/mm/presentacion-RedIRIS-NOVA.pdf
[Fecha de acceso: 13- diciembre- 2019].

[29] ALICE2 Archive, [En línea], disponible: https://www.redclara.net/index.php/es/proyectos/red-e-infraestructura/alice2
[Fecha de acceso: 13- diciembre- 2019].

[30] Red CLARA, Topología de la red CLARA año 2014, [en línea], disponible: http://www.reduniv.edu.cu/index.php/redclara/
[Fecha de acceso: 13- diciembre- 2019].

[31] GÉANT3. [En línea], disponible: https://new-archive.dante.net/Backbones/GEANT3/Pages/GEANT3.aspx
[Fecha de acceso: 13- diciembre- 2019].

[32] TERENA, Bert van Pinxteren. TERENA NREN Compendium, 2010, [en línea], disponible:
https://geant3.archive.geant.org/Network/NetworkTopology/Pages/home.aspx
[Fecha de acceso: 13- diciembre- 2019].

[33] GEANT 3 PLUS, [en línea], disponible:
https://www.rediris.es/proyectos/gn3plus/descripcion.html
[Fecha de acceso: 13- diciembre- 2019].

[34] GEANT 3 Topología año 2015, [en línea], disponible:
https://www.geant.org/Resources/Documents/topology_map16OCT15.PDF#search=G%C3%89ANT%20Topology%202015
[Fecha de acceso: 13- diciembre- 2019].

[35] El proyecto GÉANT, [en línea], disponible:
https://www.heanet.ie/projects/gn4-2
[Fecha de acceso: 13- diciembre- 2019].

[36] GEANT, Topology GEANT 2017, [en línea], disponible:
https://www.geant.org/Resources/PublishingImages/GEANT_topology_map_jan2017.jpg
[Fecha de acceso: 13- diciembre- 2019].

[37] GEANT, Topology Backbone 2018, [en línea], disponible:
https://www.geant.org/Resources/Documents/GEANT_Topology_Map_December_2018.pdf
[Fecha de acceso: 13- diciembre- 2019].

[38] GEANT Research Communities, [en línea], disponible:
https://www.geant.org/People/research_communities/Pages/Home.aspx
[Fecha de acceso: 13- diciembre- 2019].

[39] Mapa de conectividad de GEANT, [en línea], disponible:
https://map.geant.org/
[Fecha de acceso: 13- diciembre- 2019].

[40] J. I. Castillo and N. Galicia, Routing Algorithms applied to an advanced network known as CUDI, IEEE Latin America Transactions, Vol. 14. No. 6. pp. 2974-2979, June, 2016.

[41] Jose-Ignacio Castillo-Velazquez Jose-Joaquin Sanchez-Trejo, Emulation for CLARA´s operation, the advanced network for Latin America. 2016 IEEE ANDESCON. pp. 1-4, June, 2016.

[42] Management Emulation for Advanced Networks Interconection in all America: 2019 topology- IEEE Conference Publication, [en línea], disponible:
https://ieeexplore.ieee.org/document/897646
Fecha de acceso: 09 – febrero- 2020

[43] Use of GNS3 Cloud Environment for Network Management Emulation when Comparing SNMP vs Syslog Applied Over an Advanced Network, [en línea], disponible:
https://ieeexplore.ieee.org/document/8976995
Fecha de acceso: 09-febrero-2020

[44] Emulation of the connectivity of backbone and management for the layer 3 service of INTERNET2: 2016 topology, [en línea], disponible:
https://ieeexplore.ieee.org/document/8278476
Fecha de acceso: 09-febrero-2020

[45] IPV6 Connectivity and Management Emulation for REUNA, the Chilean Advanced Network, [en línea], disponible:
https://ieeexplore.ieee.org/document/8526390
Fecha de acceso: 09-febrero-2020

[46] S Deering, Arquitectura de direccionamiento IP versión 6, RFC 2373, pp 2, Julio 1998. [Fecha de acceso: 13- diciembre- 2019].

[47] Deering, Especificaciones del IPv6, RFC 2460, pp 6, Diciembre 1998. [Fecha de acceso: 13- diciembre- 2019].

[48] S.Kent, IP Encapsulating Security Payload, RFC 2406, pp 3, Noviembre 1998. [Fecha de acceso: 13- diciembre- 2019].

[49] Julio Barbancho Consejero, Redes Locales. España, 2014, pp. 52-53,107-109. ISBN: 978-84-283-3530-0.
[Fecha de acceso: 13- diciembre- 2019].

[50] Ernesto Ariganello, Redes Cisco, Guía de estudio para la certificación CCNA Routing y Switching 4ª edición, Madrid España: RA-MA, 2016, pp capítulo 5.3 Protocolos de enrutamiento, ISBN: 978-84-9964-664-0 [Fecha de acceso: 13- diciembre- 2019].

[51] Eduardo Collado Cabeza, Fundamentos de Routing, capítulo 6 y 7 pp 90,116-118,126, España, 2009, ISBN: 9781409284635.
[Fecha de acceso: 13- diciembre- 2019].

[52] C.Hedrick, Rutgers University, Routing Information Protocol, RFC 1058, pp 18, Junio1988. [Fecha de acceso: 13- diciembre- 2019].

[53] G. Malkin, RIP Version 2, RFC 2453, pp 31, Noviembre 1998.
[Fecha de acceso: 13- diciembre- 2019].

[54] G Malkin, R Minnear, RIPng for IPV6, RFC 2080, pp 5, Enero 1997.
[Fecha de acceso: 13- diciembre- 2019].

[55] J. Moy, OSPF Version 1, RFC 1131, Octubre 1989.
[Fecha de acceso: 13- diciembre- 2019].

[56] J. Moy, OSPF Version 2, RFC 2328, pp 118, Abril 1998.
[Fecha de acceso: 13- diciembre- 2019].

[57] José Ignacio Castillo Velázquez, Switching & Routing Introducción, pp 82-83, editorial SAMSARA, México 2016, ISBN 978-970-94-2977-0
[Fecha de acceso: 13- diciembre- 2019].

[58] Ernesto Ariganello, Enrique Barrientos Sevilla, Redes Cisco, Guía de estudio para la certificación CCNA Routing y Switching 3ª edición, España 2015 Grupo editorial RA-MA, pp capítulo 4.1.2 Metrica OSPF, ISBN: 978-84-9964-569-8
[Fecha de acceso: 13- diciembre- 2019].

[59] J. Moy, OSPF Version 2, RFC 2328, pp 85-86, Abril 1998.
[Fecha de acceso: 13- diciembre- 2019].

[60] D. Ferguson, OSPF for IPV6, RFC 5340, pp 5 - 84, Julio 2008.
[Fecha de acceso: 13- diciembre- 2019].

[61] Case J. Fedor M, Schoffstall M. Davin J., A Simple Network Management Protocol, RFC 1157, pp 21-33, Marzo 1991. [Fecha de acceso: 13- diciembre- 2019].

[62] M. Rose, K McCloghrie, Structure and Identification of Management Information for TCP/IP-based Internets, Marzo 1991.
[Fecha de acceso: 13- diciembre- 2019].

[63] J.I. Castillo-Velázquez, “El árbol de internet y la estructura de la información de gestión de una red”, IEEE, NoticIEEEro Number 62, Abril, 2009
[Fecha de acceso: 13- diciembre- 2019].

[64] U. Blumenthal, B. Wijnen, User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPV3), RFC 3414, Diciembre 2002.
[Fecha de acceso: 13- diciembre- 2019].

[65] D. Harrington R. Presuhn B. Wijnen, Procesamiento y despacho de mensajes para el Protocolo simple de gestión de red (SNMP), RFC 2272, pp 18-23, Enero 1998. [Fecha de acceso: 13- diciembre- 2019].

[66] Charles M. Kozierok, Book, TCP/IP GUIDE, pp 1205-1280, EEUU, 2005, [en línea], disponible:
https://lira.epac.to/DOCS-TECH/Networking/The%20TCP-IP%20Guide.pdf
[Fecha de acceso: 13- diciembre- 2019].

[67] EARLANG, Ericsson AB. Simple Network Management Protocol (SNMP), 2017 [en línea], disponible:
https://marianoguerra.github.io/otp/lib/snmp-5.2.5/doc/pdf/snmp-5.2.5.pdf
[Fecha de acceso: 13- diciembre- 2019].

[68] IBM, SNMPV3, [en línea], disponible:
https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/network/snmpv3_intro.html. [Fecha de acceso: 13- diciembre- 2019].

[69] S. Waldbusser, Protocol Operations for version 2 of the Simple Network Management Protocol (SNMPV2), RFC 1905, Enero 1996.
[Fecha de acceso: 13- diciembre- 2019].

[70] B. Wijnen, Procesamiento y despacho de mensajes para el protocolo simple de gestión de red (SNMP), RFC 3412, Diciembre 2002.
[Fecha de acceso: 13- diciembre- 2019].