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

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO

COLEGIO DE CIENCIA Y TECNOLOGÍA

Monitoreo en la red Avanzada CANARIE:

Emulación

TESIS

QUE PARA OPTAR POR EL TÍTULO DE

LICENCIADO EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

P R E S E N T A :

ALONSO DELGADO VILLEGAS

DIRECTOR DE TESIS

M. en C. José Ignacio Castillo Velázquez

Ciudad de México, octubre de 2017

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©

F1 REGISTRO DE TRABAJO RECEPCIONAL / TESIS

México, D.F. a 3 de Octubre de 2016

Mtro. José Luis Fernández Silva
Coordinador de Certificación y Registro
P r e s e n t e

Por este medio solicito el registro del trabajo recepcional, titulado:

Monitoreo en la red avanzada CANARIE: Emulación

(Título del trabajo recepcional)

Tema de estudio: Redes Avanzadas

El cual será dirigido por M. en C. José Ignacio Castillo Velázquez
Academia de Ingeniería -ISET-SLT-UACM

(Especificar nombre completo, grado académico, academia o institución de educación superior a la que pertenece)

A t e n t a m e n t e

Firma del estudiante

Firma del Director/a

Nombre:	Delgado Villegas Alonso
Matrícula:	05-003-1891
Licenciatura:	Ingeniería en Sistemas Electrónicos y de Telecomunicaciones
Plantel:	San Lorenzo Tezonco
Teléfono(s):	55 3392 0157
Correo electrónico:	alonso.delgado3@gmail.com

C.c.p. Interesado

Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno
03 OCT 2016
RECIBIDO
Hora 15:10
Area Titulación

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

Universidad Autónoma de la Ciudad de México

Nada humano me es ajeno

Coordinación de Certificación y Registro

F 5 FECHA DE EXAMEN PROFESIONAL

Ciudad de México. a 04 de Octubre de 2017

El(La) que suscribe, **M. en C. José Ignacio Castillo Velázquez**, profesor de la academia de ingeniería en Sistemas Electrónicos y de Telecomunicaciones del Plantel San Lorenzo Tezonco, considero que el trabajo recepcional cumple con los requisitos académicos para programar el EXAMEN PROFESIONAL del egresado:

Nombre:	**Delgado Villegas Alonso**
Matrícula:	**05-003-1891**
Licenciatura:	**Ingeniería en Sistemas Electrónicos y de Telecomunicaciones**
Nombre del trabajo recepcional:	**Monitoreo en la red avanzada CANARIE: Emulación**

Para el día viernes 20 del mes de octubre del presente año, en el plantel San Lorenzo Tezonco a las 12 horas.

Asimismo, se propone que el jurado este integrado por los siguientes profesores:
(Especificar nombre completo, grado académico, academia o institución a la que pertenecen)

1. Dr. Gerardo Abel Laguna Sánchez (UAM-Lerma) Presidente
2. M. en C. Joel Yazbek Buendía Gómez (UACM-SLT) Secretario
3. Ing. José Miguel Vargas Pliego (UACM-Cuautepec) Vocal

Atentamente

Firma del Director(a)

C.c.p. Interesado

Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno
04 OCT 2017
RECIBIDO
Hora 13:23 VSG
Area Titulación

F.4 Voto Aprobatorio de Lector

Ciudad de México, a 7 de Julio de 2017

Mtro. José Luis Fernández Silva
Coordinador de Certificación y Registro
Presente

Por este conducto, me permito comunicarle que he revisado detalladamente la tesis intitulada:

"Monitoreo en la red Avanzada CANARIE: Emulación"

Presentado por:

Estudiante		Matrícula:	**05-003-1891**
Nombre completo:	Alonso Delgado Villegas		
Licenciatura:	Ingeniería en Sistemas Electrónicos y de Telecomunicaciones		

El cual cumple con los requisitos académicos, por lo tanto, en mi carácter de lector emito mi **VOTO APROBATORIO** para presentar el Examen Profesional.

Atentamente

M. en C. Joel Yazbek Buendía Gómez
Academia de Sistemas Electrónicos y Telecomunicaciones –SLT
UACM

c.c.p. Interesado

PS/CCYH

4 Voto Aprobatorio de Lector

Ciudad de México, a 29 de Agosto de 2017

Mtro. José Luis Fernández Silva
Coordinador de Certificación y Registro
Presente

Por este conducto, me permito comunicarle que he revisado detalladamente la tesis intitulada:

"Monitoreo en la red Avanzada CANARIE: Emulación"

Presentado por:

Estudiante		Matrícula:	**05-003-1891**
Nombre completo:	Alonso Delgado Villegas		
Licenciatura:	Ingeniería en Sistemas Electrónicos y de Telecomunicaciones		

El cual cumple con los requisitos académicos, por lo tanto, en mi carácter de lector emito mi **VOTO APROBATORIO** para presentar el Examen Profesional.

Atentamente

Dr. Gerardo Abel Laguna Sánchez
Sistemas de Información y Comunicaciones
UAM

c.c.p. Interesado

PS/CCYH

F.4 Voto Aprobatorio de Lector

Ciudad de México, a 30 de Agosto de 2017

Mtro. José Luis Fernández Silva
Coordinador de Certificación y Registro
Presente

Por este conducto, me permito comunicarle que he revisado detalladamente la tesis intitulada:

"Monitoreo en la red Avanzada CANARIE: Emulación"

Presentado por:

Estudiante	Matrícula:	**05-003-1891**
Nombre completo:	Alonso Delgado Villegas	
Licenciatura:	Ingeniería en Sistemas Electrónicos y de Telecomunicaciones	

El cual cumple con los requisitos académicos, por lo tanto, en mi carácter de lector emito mi **VOTO APROBATORIO** para presentar el Examen Profesional.

Atentamente

Lic. José Miguel Vargas Pliego
Academia de Sistemas Electrónicos y Telecomunicaciones -Cuautepec
UACM

c.c.p. Interesado

AGRADECIMIENTOS

El éxito profesional comienza por la apertura de las puertas de la casa de enseñanza, la cual pone su mejor esfuerzo, otorgando los recursos necesarios que permitan alcanzar las metas propuestas, es por ello que, agradezco a la **Universidad Autónoma de la Ciudad de México** poder ver logradas esas metas, al permitir formarme como profesional dentro de sus aulas, así como también, agradezco el apoyo brindado para realizar la impresión y/o empastado de esta tesis. De igual forma, gracias a cada uno de los maestros que compartiendo su conocimiento, hicieron posible dicha formación académica.

"Zombies atados al consumo de la tecnología, es la mejor forma de evadir el conocimiento". Palabras similares, que al principio no comprendí, son parte de la formación que bajo su liderazgo, tras la culminación de este proyecto, serán el motor que impulse el deseo de seguir adquiriendo conocimiento. **M. en C. José Ignacio Castillo Velázquez**, gracias por tomar parte de su valioso tiempo para dirigir este proyecto, permitiéndome ser parte del ADVNETLAB.

La colaboración grupal y el trabajo en equipo facilitan el desarrollo de nuevo conocimiento, es por ello que agradezco el apoyo brindado por el **M. en C. Joel Yazbek Buendía Gómez**, al permitir el uso del sistema "Xexelo" construido bajo el proyecto "Laboratorio de Sistemas Distribuidos y Redes de Alto Desempeño", No. PI2011-34R, convenio: 060/2013, el cual se encuentra dentro del Laboratorio de Redes de Computadoras (B-404), de la Academia de Ingeniería en Sistemas Electrónicos y de Telecomunicaciones, en el campus UACM-SLT. Agradezco también sus enseñanzas dentro del aula, así como el aceptar ser mi lector y aportar las observaciones realizadas a la presente tesis.

DR. Gerardo Abel Laguna Sánchez, gracias por su colaboración y tiempo dedicado en la lectura de la presente tesis. Gracias por las observaciones y sugerencias realizadas para mejorar la calidad del trabajo.

Ing. José Miguel Vargas Pliego, sus sugerencias y críticas hacen de esta una mejor tesis, es por ello que, agradezco su participación y esfuerzo dedicado en la lectura

del presente trabajo, aportando sugerencias con la finalidad de enriquecer su contenido

DEDICATORIA

Dedico esta tesis a mi familia que siempre me ha apoyado e impulsado para alcanzar este grado académico universitario. A ti Wendy García por ser la más paciente, apoyándome en el último trayecto de mi carrera. Dedico esta tesis también a esas personas con quienes el destino me ha cruzado, con las cuales he formado lazos de afecto y amistad y quienes a su vez me han alentado a terminar la licenciatura: Angélica Mateo, Maribel Torres, Adamari Contreras, Socorro Dolores, Maricruz González.

Índice

Resumen

La red Avanzada de Educación e Investigación canadiense es administrada por la organización CANARIE, establecida en 1993, desde su construcción, se ha preocupado por otorgar un *backbone* capaz de soportar el flujo de datos generados por sus miembros participantes. En junio de 2013, CANARIE presentó un *backbone* capaz de soportar 100 Gbps, utilizando equipos Ciena 6500 que incluyen procesadores ópticos *WaveLogic* de tercera generación, y en 2016 realizó pruebas para actualizar su infraestructura a un *backbone* de ultra alta velocidad, para soportar 300 Gbps, utilizando equipos *Waveserver* de Ciena.

Con la finalidad de conocer y entender el funcionamiento del *backbone* CANARIE, se realizó la simulación y emulación de conectividad y gestión de dicho *backbone,* en su versión actualizada al 2016, utilizando las herramientas de simulación en *Packet Tracer* como primera aproximación, y de emulación en GNS3 como segunda aproximación, para lo cual se utilizaron 25 *routers* tipo *core* que conforman el *backbone.* Se configuró a cada uno de los *routers* con el protocolo de enrutamiento OSPF y con el protocolo de gestión SNMP.

La simulación de conectividad ocupó 32% del procesador Intel Pentium Dual CPU E2140 @ 1.60 GHz y 40% (1.20 GB de 3 GB) de memoria RAM para simular la conectividad, desde el *router* Vancouver al *router* ST. John's, en un tiempo de 2.48 minutos. La simulación de gestión en la mayoría de ocasiones realizó el monitoreo de los elementos de gestión, permitiendo configurar dos de las cinco variables elegidas para realizar la simulación de gestión. Por su parte, para llevar a cabo la emulación de conectividad y gestión, se utilizó una computadora con procesador Intel (R) Xeon (R) CPU E5-2620 v2 @ 2.10 GHz. El consumo de recursos en la emulación fue de 4% del procesador y 37.81% (12.1 GB de 32 GB) de memoria RAM. El tiempo establecido desde la ejecución de GNS3 hasta tener las terminales abiertas de las cinco máquinas virtuales fue de 9.50 minutos. La emulación de gestión por su parte, además de permitir la configuración de 3 de las 5 variables propuestas, también permitió recibir los *traps* (alertas) enviados por los equipos gestionados, los cuales se capturaron a través de *Wireshark*. Se aplicaron los conocimientos de configuración y

monitoreo de redes con base en la conectividad y gestión, vía simulación y emulación del *backbone* CANARIE, el cual cuenta con 25 *routers* de *backbone.*

Capítulo 1: Introducción

1.1 Introducción

Desde la apertura al ámbito comercial de la Internet, en abril de 1995, nuevas ideas han surgido para preservar e impulsar la educación e investigación en el sector académico, con el objetivo de crear herramientas que faciliten el desarrollo de nuevos proyectos que aportarán valor a la vida humana. La idea de una red académica prevalece hoy día con el surgimiento de la INTERNET2 en Estados Unidos, desde donde se ramificó hacia todo el mundo, permitiendo la creación de una red mundial de redes avanzadas sin fines de lucro.

Canadá ha creado su propia red avanzada, para permitir la colaboración científica y académica, otorgando conexión a doce redes regionales que, en conjunto con CANARIE (organismo encargado de regular la conectividad a la red avanzada canadiense), colaboran en la investigación en áreas como la física de partículas, genómica, astronomía y neurología [1]. Además de la conexión regional, CANARIE permite de forma internacional la conexión con más de 100 redes similares alrededor del mundo, colaborando en el desarrollo de proyectos tales como:

- ***ATLAS/TRIUMF:*** El laboratorio Nacional de Física Nuclear y de Partículas de Canadá, opera un centro de informática para el experimento ATLAS en el Gran Colisionador de Hadrones, ubicado en el CERN (*European Organization for Nuclear Research*) en Suiza, el cual, genera datos del orden de los petabytes (10^{15}) anualmente, TRIUMF se encarga de almacenarlos y distribuirlos para su procesamiento.
- ***Square Kilometre Array* (SKA):** Desarrollado en Australia y Sudáfrica, se considera el radiotelescopio más grande, al poseer una superficie de un kilómetro cuadrado, operando en un rango de frecuencias entre 0,10 a 25 GHz con la finalidad de analizar el cosmos en busca de planetas similares a la Tierra. Su construcción inició en 2016 y se planea que entre en funcionamiento a partir de 2020.
- ***OutGrid:*** Es una plataforma global utilizada por neurocientíficos, para analizar imágenes cerebrales en 3D y 4D, en busca de comprender el funcionamiento del cerebro y enfermedades que provocan su degeneración [2].

- **Consorcio Internacional del Genoma del Cáncer:** Es una organización internacional dedicada al estudio de 50 tipos de tumores. El proyecto fue puesto en marcha en 2008, con la finalidad de estudiar más de 25,000 genomas de cáncer.

Los proyectos mencionados, son posibles gracias a las plataformas de investigación, que no son más que aplicaciones de *software,* dedicadas al apoyo de investigación realizada por CANARIE. [3]

CANARIE se caracteriza por ser socio del CENGN (*Center of Excellence in Next-Generation Networks* – Centro de Excelencia en Redes de la Siguiente Generación), para el cual trabaja en conjunto con compañías de ICT (*Information and Communication Technology* – Tecnología de la Información y la Comunicación), para desarrollar una plataforma única de multi-proveedores para el desarrollo, prueba y comercialización de nuevas tecnologías de redes y comunicaciones. [4]

La infraestructura del *backbone* CANARIE ha evolucionado, desde 1996 hasta 2016, tanto en la capacidad de sus *routers* como en sus enlaces, para proveer los recursos que permitan a su vez desarrollar proyectos, tanto de investigación como de aplicación, de ahí surge la importancia de analizar la evolución y complejidad de su funcionamiento, para poder determinar su comportamiento futuro.

La red CANARIE, al ser una red avanzada, se vuelve parte de la serie de proyectos a desarrollar en el ADVNETLAB dentro de la UACM-SLT. CANARIE forma parte del compendio que integran las redes avanzadas en el continente Americano, con lo cual, el ADVNETLAB completará el estudio de dichas redes, no sólo a nivel país, si no que se tendrá un análisis a nivel continente.

1.2 Justificación

El estudio de las redes avanzadas han sido el tema de interés del ADVNETLAB, en él, se han producido tesis como: Emulación del *backbone* de la red avanzada Internet2 en México, Implementación de un modelo IPv4 *multicast* y Emulación de la red avanzada CLARA. Así cómo los siguientes artículos arbitrados: *Routing algorithms applied to an advanced academic network known as CUDI* [5], *Emulation of backbone's connectivity and management for the advanced network in Latin America: 2016's topology* [6], *Emulations for CLARA's operation, the advanced network for Latin America* [7], Ingeniería inversa parcial y simulación de la infraestructura de una red de datos MAN, ROC&C México, 2013 [8] y ROC&C México, 2013 [9]. CANARIE, al ser una red avanzada, forma parte de la plantilla que se ha venido generando en el laboratorio ADVNETLAB, dando continuidad al estudio de dichas redes, es por ello que, se lleva a cabo el análisis de su *backbone,* utilizando herramientas de simulación en *Packet Tracer* como primera aproximación y, emulación en GNS3, como segunda aproximación.

1.3 Objetivo general

Conocer y entender el funcionamiento del *backbone* de la red avanzada CANARIE para poder predecir su comportamiento futuro, para lo cual, se simularán y emularán su conectividad y gestión utilizando las herramientas *Packet Tracer* y GNS3. Al mismo tiempo, se comprobará la eficiencia de las herramientas mencionadas anteriormente, al forzarlas a trabajar con los dispositivos de red que conforman el diseño del *backbone* CANARIE.

1.4 Objetivos particulares

Adquirir habilidad para manipular y configurar las herramientas de diseño de red *Packet Tracer* y GNS3, con la finalidad de realizar las pruebas de conectividad y gestión, así como para utilizarlas en el diseño y análisis de redes de datos antes de llevarlas a la implementación.

Adquirir la habilidad de interpretar los resultados obtenidos durante la realización de las pruebas de conectividad y gestión, plasmándolos de forma coherente, para brindar una comprensión adecuada.

Adquirir el hábito de realizar una planeación de actividades, con la finalidad de obtener la habilidad de desarrollar proyectos, siguiendo una metodología y un calendario de trabajo con tareas programadas, para desarrollarlas en tiempo y forma.

1.5 Estructura de la tesis

El presente trabajo está conformado por cinco capítulos, en los cuales se abordan temas de evolución tecnológica de la infraestructura de las redes avanzadas en el continente americano, protocolos de enrutamiento y gestión, simulación y emulación del *backbone* CANARIE y, finalmente, se presentan resultados.

A continuación se hace una breve descripción del contenido de cada uno de ellos.

Capítulo 2. Abarca contextos de evolución de las redes avanzadas en el continente americano. Se da inicio con una breve introducción a la historia de la Internet, posteriormente, se presenta la evolución de la INTERNET2, como una red avanzada enfocada al desarrollo tecnológico en beneficio de la investigación y educación en Estados Unidos, también, se presenta la evolución de la red avanzada CLARA, establecida en Latinoamérica. Sin embargo, se hace énfasis en Canadá, país que alberga a la red avanzada CANARIE, tema principal para el desarrollo de este trabajo.

Capítulo 3. Se hace una revisión de los protocolos de enrutamiento RIPv1, RIPv2 y OSPF versiones 1 y 2, así como del protocolo de gestión de red, cubriendo los aspectos básicos de SNMP versiones 1 y 2.

Capítulo 4. En este capítulo, se realizan las especificaciones metodológicas necesarias para someter a pruebas de conectividad y gestión al *backbone* de la red CANARIE, mediante simulación y emulación. Se describen los requerimientos y la

configuración necesaria para el correcto funcionamiento de las herramientas de simulación (*Packet Tracer*) y emulación (GNS3).

Capítulo 5. Se presentan los resultados obtenidos, tras haber realizado la implementación del capítulo 4, se hace un contraste entre los objetivos planteados y los resultados obtenidos, para hacer un balance de conocimientos adquiridos durante el desarrollo de la tesis, al presentar las conclusiones.

Capítulo 2: Internet y Redes Avanzadas

2.1 Origen de la Internet

El nacimiento de la informática no cuenta con una fecha en específico, sin embargo, toma importancia desde la publicación del artículo "*On computable numbers, with an application to the Entscheidungsproblem",* escrito por Alan Turing en 1936 [10], en dicho artículo se estableció el diseño abstracto de una computadora electrónica. La aportación teórica de Turing propició la construcción de la nueva tecnología computacional. Por otra parte, Konrad Zuse, en 1936, construyó en Alemania la primera computadora electromecánica programable, conocida como la Z1. La segunda versión apareció con la Z2 y, posteriormente, en 1941, se construyó la Z3, primera computadora electromecánica completamente automática, utilizaba el álgebra booleana para realizar operaciones [11].

Durante la década de los 50, después de que el primer satélite artificial se lanzó al espacio por la Unión Soviética, el presidente de Estados Unidos, Dwight D. Eisenhower, fundó en febrero de 1958 la ARPA (*Advanced Research Proyects Agency* – Agencia de Proyectos de Investigación Avanzados), con el objetivo de mantener el liderazgo tecnológico en el área de las aplicaciones militares a nivel mundial. La ARPA se dedicó a financiar instituciones universitarias y militares para que desarrollaran un programa computacional. Como resultado de dicha financiación, aparecieron trabajos teóricos como "*Information flow in large communications net*" de Leonard Kleinrock, publicado en 1961, [12] en el cual se proponía una nueva forma de comunicación electrónica mediante la conmutación de paquetes (tecnología básica detrás de Internet), la cual consta de dos secciones; los datos e información de control, mediante las cuales se indica la ruta a seguir dentro de una red hasta llegar a su destino. Posteriormente, en 1962, Joseph Carl Robnett (J. C. R.) Licklider en conjunto con Welden E. Clark, publicaron su trabajo "*On-Line Man-Computer Communication*", presentando el concepto de "Red Galáctica" [13], una forma de tener varias computadoras conectadas entre sí para acceder a datos y programas, sin importar su ubicación geográfica. En 1964, Paul Baran, quien trabajó para la corporación RAND, con quien ARPA firmó contrato en busca de desarrollar un nuevo proyecto de comunicaciones, se dedicó hacer un estudio detallado de las posibles vulnerabilidades que tenía la red telefónica de AT&T (principal proveedor de telefonía

en EE.UU en esa época), para volverla más robusta ante un posible ataque de misiles nucleares. Como resultado, Baran dio a conocer la primera red de comunicaciones distribuidas, planteando que un sistema centralizado que formaba una conexión tipo estrella era sumamente vulnerable, al contar con una única central. Sin embargo, un sistema jerárquico que enlaza varios sistemas centralizados, puede ser un poco más robusto, pero un sistema de centrales distribuidas por diferentes regiones, enlazándose unas a otras, lo volvía un sistema sumamente resistente y funcional. Esencialmente formuló dos ideas; la primera de ellas fue el uso de una red descentralizada con varios enlaces entre dos puntos, la segunda fue fragmentar los mensajes para que se enviaran por distintos enlaces hasta alcanzar el nodo destino donde se reensamblarían.

Los diagramas diseñados por Baran se muestran en la figura 2.1.

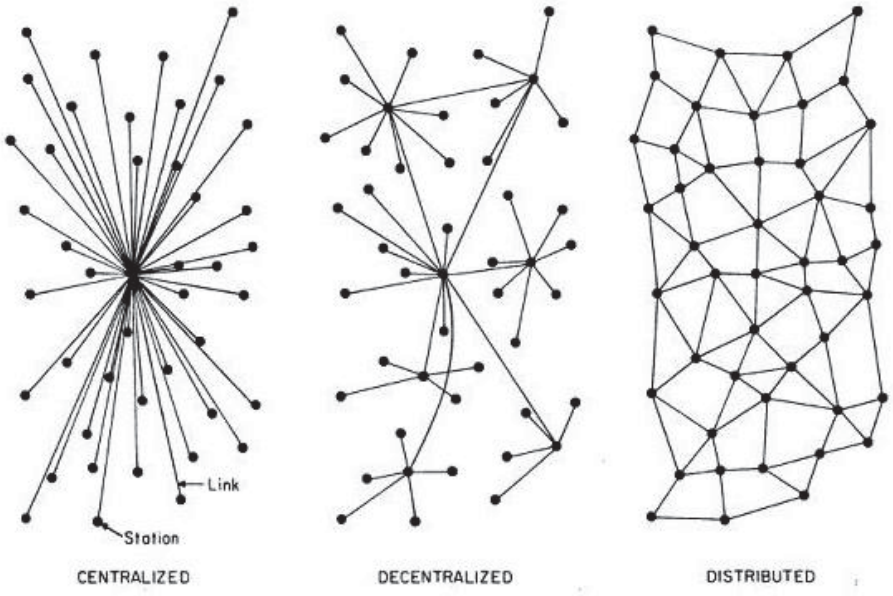

Figura 2.1: Robustez de un sistema distribuido frente a un sistema centralizado, presentado por Paul Baran en 1964. [14]

Casi al mismo tiempo, en Reino Unido, Donald Watts Davies dio a conocer un sistema similar al de Baran, incluyendo los conceptos de “paquete” y “conmutación de paquetes”, definiendo así la forma de transportar la información dentro de la red. Robert W. Taylor se puso en contacto con Lawrence G. Roberts para que liderara el nuevo proyecto que permitiría la creación de la primera red de computadoras, la cual

llevaría el nombre de ARPANET (*Advanced Research Proyects Agency Network* – Red de la Agencia de Proyectos de Investigación Avanzados). El primer mensaje a través de ARPANET se transmitió el 29 de octubre de 1969, al conectar cuatro diferentes computadoras entre sí; la primera fue una SDS sigma 7, ubicada en la UCLA (Universidad de California en Los Ángeles), con un enlace directo a la computadora SDS 940 en el SRI (Instituto de Investigación de Stanford); posteriormente se enlazó la IBM 360/76 en la UCSB (Universidad de California en Santa Bárbara) y, por último, una DEC-PDP-10 en la Universidad de Utha. Con ello, surgió la primera MAN (*Metropolitan Area Network* – Red de Área Metropolitana). La comunicación entre nodos se realizó a través de la línea telefónica, utilizando un MODEM a 56 Kbps que conectó a los IMP (*Interfaz Message Processor* – Procesadores de Interfaz de Mensaje) las cuales son minicomputadoras *Honeywell* DDP-316 dedicada a realizar la función de un enrutador al comunicar las redes. [15]

Al igual que el funcionamiento del correo postal, la información generada en las computadoras necesitaba de “alguien” que las dirigiera a su destino. Esta función estaba a cargo del NCP (*Network Control Protocol* – Protocolo de Control de Red), manteniéndose en funcionamiento desde 1969 hasta 1973, cuando Vinton Cerf y Robert Kahn publicaron el documento “un protocolo para la interconexión de red de paquetes”, donde se describe el TCP (*Transmision Control Protocol* – Protocolo de Control de Transmisión), especificando la forma en cómo dos computadoras mantienen comunicación. TCP incluye conceptos de encapsulamiento, datagrama y funciones de pasarela. Más tarde, fue dividido en 2 secciones; la primera se conoce como Protocolo de Control de Transmisión (TCP), encargado de administrar funciones de segmentación, reagrupamiento y detección de errores; la segunda se conoce como IP (*Internetworking Protocol* – Protocolo de Internet), encargado de manejar el enrutamiento de los datagramas para realizar la comunicación entre *host,* en conjunto, ambos protocolos TCP/IP están formados por cuatro capas de acceso: Interfaz, Internet, transporte y aplicación [16]. Para 1980, el departamento de defensa declaró a TCP/IP como el estándar para las comunicaciones entre redes militares. La migración de ARPANET a TCP/IP, que se había venido gestando desde su aparición, concluyó oficialmente en 1983 [17]. La incompatibilidad de comunicación entre los

diferentes sistemas utilizados por fabricantes de computadoras, obligó a la ISO (*International Organization for Standardization* – Organización Internacional para la estandarización) a construir un modelo de comunicación que permitiera la comunicación entre dichos sistemas, para lo cual, realizó estudios a los modelos de conexión ya existentes, como lo eran DECnet (*Digital Equipment Corporation Network* – Red de la Corporación de Equipamiento Digital. DECnet desarrolló una de las primeras arquitecturas de red *peer-to-peer* entre computadoras de diferente proveedor); la Arquitectura de Sistemas de Red (*Systems Network Architecture*) y TCP/IP, dando como resultado el modelo de referencia OSI (*Open Systems Interconnection* – Interconexión de Sistemas Abiertos), el cual provee un conjunto detallado de estándares utilizados en las redes de comunicaciones. El modelo OSI consta de siete capas relacionadas entre sí, cada una de las capas añade información extra al paquete a transmitir, la cual es retirada por cada capa del sistema que la recibe. El modelo de referencia OSI se describe en la norma ISO 7498-1 (ITU-T X.200). [18]

Una década de operaciones y crecimiento en el número de redes demostraron que ARPANET era la nueva forma de comunicación, al contar con 40 redes en 1981; estos eventos propiciaron que nuevas redes comenzaran a desarrollarse, tal como lo hizo la sección de la ciencia de la NSF (*National Science Fundation* – Fundación Nacional para la Ciencia), que desarrolló una red similar a ARPANET, denominada CSNET (*Computer Science Network* – Red Informática), con la finalidad de conectar a los departamentos de informática de las universidades y centros de gobierno que no podían conectarse directamente con ARPANET, además, logró establecer la comunicación entre Estados Unidos y centros de investigación en Europa y Asía. La CSNET se mantuvo en funcionamiento en el período de 1981 a 1985. Posteriormente, en 1983, la red militar (MILNET) decidió separarse de las redes civiles, dejando a ARPANET con 45 nodos de los 113 que administraba en ese momento, dando paso a la unión entre la CSNET y ARPANET, utilizando el conjunto de protocolos TCP/IP. Para 1985, la NSF decidió replicar el éxito obtenido con la red CSNET, con la finalidad de crear una red de redes, desplegó una nueva red de índole educativa, para apoyar a las universidades de Estados Unidos en el desarrollo

de proyectos de investigación. Para ello, tendió una red a nivel nacional que fungía como *backbone* (red principal), conectando a sus centros de supercomputadoras ubicados en las regiones de San Diego, Boulder, Champaign, Pittsburgh, Ithaca y Princeton. La red se nombró NSFNET (*National Science Foundation Network* – Red de la Fundación Nacional para la Ciencia). [19]

ARPANET dejó de existir en 1990, al considerarse que ya había logrado el propósito por el cual se había construido. En su lugar, continuó operaciones la NSFNET con 16 nodos que proporcionaban conexión a 3,500 redes en 1991. El crecimiento en el número de redes, impulsó una actualización en la capacidad del *backbone,* proporcionándole una velocidad de conexión de 45 Mbps (T3). La cantidad de usuarios continuó en aumento, muchos de ellos con intereses comerciales, por lo que se creó el CIX (*Commercial Internet Exchange* – Intercambio de Internet Comercial), como un consorcio de proveedores de servicios de red, a través de los cuales se les facilita a los usuarios una conexión de acceso a dicha red, esto debido a que las políticas de la NSFNET estipulaba su uso únicamente para la investigación y educación académica, sin embargo, en 1995, la NSFNET se cerró, dando paso al uso privado de Internet (*Interconnected Networks* – Redes Interconectadas) la cual ya contaba con 100,000 redes públicas y privadas. Oficialmente el Internet comercial inicio operaciones el 30 de abril de 1995, quedando a cargo de los ISP (*Internet Service Provider* – Proveedor de servicios de Internet). Durante este proceso, surgieron redes académicas que se conectaban directamente a la NSFNET, sin embargo, tras el cierre de la misma, la NSF estableció una nueva red, denominada vBNS (*very high performance Backbone Network Service* – Servicios de *backbone* de alto rendimiento), para seguir brindando apoyo a las universidades y centros de investigación mediante enlaces de 155 Mbps, alcanzando los 2.5 Gbps en 1999. Dentro de esta red se desarrollaron tecnologías como *IP- multicasting*, calidad de servicio e IPv6. El proyectó propició el surgimiento de una nueva era tecnológica conocida como redes avanzadas, dando inicio a la Internet 2.

2.2 Redes Avanzadas

Las NREN (*National Research and Education Networks* – Redes de Educación e Investigación Nacional), desarrollan la nueva era de redes de computadoras que brindan soporte a la información generada por los proyectos de investigación, los cuales son desarrollados dentro de las universidades y centros de investigación. El término “redes avanzadas”, es asignado a la infraestructura utilizada para proveer conexión a universidades e instituciones, dedicadas a la investigación científica. La infraestructura se caracteriza por poseer enlaces del orden de los gigabits, además, utiliza equipos de la más reciente tecnología, así como la evaluación de nuevos protocolos que permiten ejecutar aplicaciones en tiempo real. Las redes avanzadas se han construido alrededor del mundo, cada una de ellas lideradas bajo un consorcio interno que se encarga de construir la infraestructura necesaria para soportar la transferencia masiva de datos, ejecutar video en tiempo real y sobre todo, apoyar en la investigación y colaboración en el desarrollo de diversos proyectos. Por lo tanto, las redes avanzadas deben cumplir dos objetivos principales; el primero es construir la tecnología que ayude a científicos a manejar los datos generados en el desarrollo de los distintos proyectos que estén ejecutando, el segundo consiste en construir y garantizar el correcto funcionamiento de las aplicaciones que han desarrollado para ponerlas en funcionamiento dentro de la Internet comercial. [20, 21]

A continuación se describe la historia de las redes académicas avanzadas de mayor jerarquía construidas en el continente Americano, indicando principalmente la evolución que ha tenido su *backbone,* utilizando una infraestructura basada en fibra óptica que alcanzan los 100 Gbps en Estados Unidos y Canadá.

2.3 INTERNET2

Las instituciones de enseñanza de nivel superior de EE.UU ya habían probado o, por lo menos, escuchado de los beneficios que se tenían al contar con una red de computadoras. Tras la liberación de la Internet al ámbito comercial, esta se vio saturada con todo tipo de información que muchas veces no proviene de fuentes confiables. En 1996, 34 universidades de Estados Unidos lideraron un nuevo proyecto para construir una red que, en principio, sólo existiese entre universidades con la finalidad de impulsar la educación. Se conformó el consorcio que administra el desarrollo de la red, conocido como UCAID (*University Corporation for Advanced Network Development* – Corporación Universitaria para el Desarrollo de Redes Avanzadas), cambiando posteriormente el nombre a INTERNET2. La sede del consorcio se encuentra en Michigan, Estados Unidos. Un año más tarde, se colocaron por todo el país puntos de interconexión GigaPOPs, conocidos como puntos de presencia (los GigaPoPs son un punto de encuentro entre el *backbone* y 12 redes institucionales que conformaban INTERNET2 a sus inicios, algunos de estos GigaPops permiten la interconexión de redes de investigación similares), para interconectarse a la red de alto rendimiento de la Fundación Nacional para la Ciencia (vBNS) utilizándola como *backbone,* con lo cual se le proporcionó servicios de red a las universidades que formaron parte de INTERNET2. En 1998, la INTERNET2 dio a conocer su primer *backbone* llamado “Abilene”, construido gracias a las aportaciones realizadas por Qwest *Communications*, Cisco *Systems* y Nortel *Networks*. En 1999, la red avanzada CANARIE establecida en Canadá, se convirtió en el primer socio internacional, al interconectarse con la INTERNET2, un poco después, se unió Europa a través de la red avanzada operada por DANTE. La red Abilene se modificó prácticamente desde su creación con la finalidad de brindar el mejor servicio a los investigadores. En 2004, se realizó un cambio de gran magnitud, incluyéndose una red óptica con características OC-192 (*Optical Carrier 192*), en casi toda su infraestructura, que proporciona una capacidad de 10 Gbps. Solamente el enlace entre Indianápolis y Atlanta permaneció a 2,48 Mbps (OC-48) [22].

También se inició la inclusión de IPv6 para conectar a unas 200 universidades y cerca de 20 redes avanzadas a nivel mundial no conectadas de forma directa. En

2006, INTERNET2 inició negociaciones con la empresa *Level 3 Communications* para cambiar el *backbone* a una red de nueva tecnología conocida como SDN (*Software Defined Networking* – Red Definida por Software), sobre fibra óptica de 100 Gbps, con capacidad de transferencia de 8,8 Terabytes de información. La nueva red inició pruebas de operación, enlazando los estados de Washington, D.C y Nueva York. La nueva infraestructura se completó en 2012, retirando en su totalidad a la red Abilene.

Con la primera sección de la nueva red establecida entre Washington, D.C y Nueva York, la red de las Ciencias de la Energía del Departamento de Energía de Estados Unidos (ESnet) e INTERNET2, dieron a conocer el primer enlace transcontinental conectando a Nueva York, Washington, D.C, Cleveland, Chicago, Kansas City, Denver, Salt Lake City y Sunnyvale. [23]

El trabajo en conjunto de las redes de investigación y educación INTERNET2, NORDUnet, ESnet, SURFnet, CANARIE y DANTE/GÉANT, así como los socios industriales Ciena y Tata comunicaciones, pusieron en marcha en 2012 “El Piloto Avanzado del Atlántico Norte 100G”, conocido como ANA-100G, para unir a las redes avanzadas de América y Europa mediante un enlace de 100 Gbps. La primera sección del enlace trasatlántico se completó en 2013, mostrando su funcionalidad en la conferencia de redes TERENA 2013, realizada en Maastricht, Países Bajos. La nueva sección tolerante a fallos se instaló el año siguiente completando el anillo, lo cual dio una capacidad de 200 Gbps, al utilizar cuatro enlaces *Open Lightpath Exchanges*. [24]

INTERNET2 mantiene participación con redes similares a nivel mundial a través de puntos de intercambio de alto rendimiento, por los cuales se ofrecen servicios que facilitan la colaboración internacional.

Los puntos de interconexión se especifican a continuación:

- *Manhattan Landing* (**MAN LAN):** Ubicado en la ciudad de Nueva York, soporta conexiones de Ethernet de capa 2. Este punto de intercambio existe gracias a la aportación de las redes Internet2 y la red de educación e investigación del

estado de Nueva York (NYSERNet, *The New York State Education and Research Network*).

- *Washington International Exchange* (**WIX**): Ubicado en McLean, Virginia. Soporta servicios de capa 2 para permitir la conexión entre las redes internacionales e INTERNET2.
- *Singapore Global Facility* (**SGF**): Es un punto de colaboración global, principalmente utilizada por miembros de INTERNET2 en Asía, el cual proporciona servicios de red neutra y segura para las universidades miembros. [25]
- *Global connectivity:* INTERNET2 establece puntos de conexión a nivel mundial para conectar redes similares, con la finalidad de contribuir con otros investigadores alrededor del mundo.

Otros puntos de intercambio administrados por INTERNET2 son:

- AMPHAT: punto de conexión internacional ubicado en Miami, Florida, para interconectar a las redes avanzadas de América Latina y el Caribe, con Internet2.
- *Atlantic Wave*: Este nodo permite conectar a Estados Unidos, Canadá, Europa y Sudamérica, a través de varios nodos distribuidos en Nueva York, Washington, D.C, Atlanta, Miami y Sao Paulo. También enlaza a los puntos de intercambio en la costa este de Estados Unidos, estos son: MAN LAN, NGIX-East, SoX y AMPHAT.
- *Pacific Wave*: Es un proyecto realizado en conjunto entre la Corporación para las Iniciativas de la Red de Educación en California (CENIC) y el *Pacific Northwets* GigaPoP (PNWGP), ofrece una malla de tres puntos de conexión en la costa del Pacífico de los Estados Unidos: área de la bahía (Sunnyvale y Palo Alto), Los Ángeles y Seattle.
- *StartLight*: es un *router* que permite conexión de alto rendimiento a través de enlaces ópticos. Es administrado por la *Northwest University*, en Chicago, Illinois. [26]

En 2014, la INTERNET2 contaba con miembros afiliados tales como universidades, corporaciones, agencias gubernamentales de investigación y organizaciones sin fines de lucro, que conforman una plantilla de poco más de 93,000 miembros afiliados en Estados Unidos, e instituciones de 100 países alrededor del mundo, es por ello que al tener una red administrada por software, los investigadores pueden generar diferentes servicios, con el objetivo de aprovechar al máximo la capacidad de la red, permitiendo la interacción de cada socio, sin que se muestren problemas de operación [27]. Los servicios diseñados son:

- Servicios avanzados de capa 3: Diseñado para ofrecer una red de vanguardia que solvente las necesidades de la creciente evolución de los programas de investigación que requieren de una red de alta velocidad [28, 29].
- Servicios avanzados de capa 2: Los miembros de Internet2 tienen la facilidad de crear y administrar sus propias redes VLAN a través de la infraestructura ya existente. Este servicio se utiliza como intercambio abierto, permitiendo acceso tanto a redes regionales como a redes de conexión mundial [30, 31].
- Servicios avanzados de capa 1: Permite crear una red más personal, con capacidades de 10 y 100Gbps, contiene la mayor cantidad de puntos de acceso en el *backbone* de INTERNET2 [32, 33].

En la figura 2.2 se presenta la infraestructura de INTERNET2, construida con arquitecturas Multi-Tenant (multi-propietario, es una arquitectura de software como servicio, se aloja en un servidor, permitiendo a los usuarios acceder a los recursos desde una misma plataforma tecnológica) y SDN-*Powered* impulsado por el *software* "*FlowSpace Firewall".* A través de esta tecnología, fue posible dividir la red en múltiples redes discretas y privadas, otorgando un ancho de banda de 100 Gbps. [34]

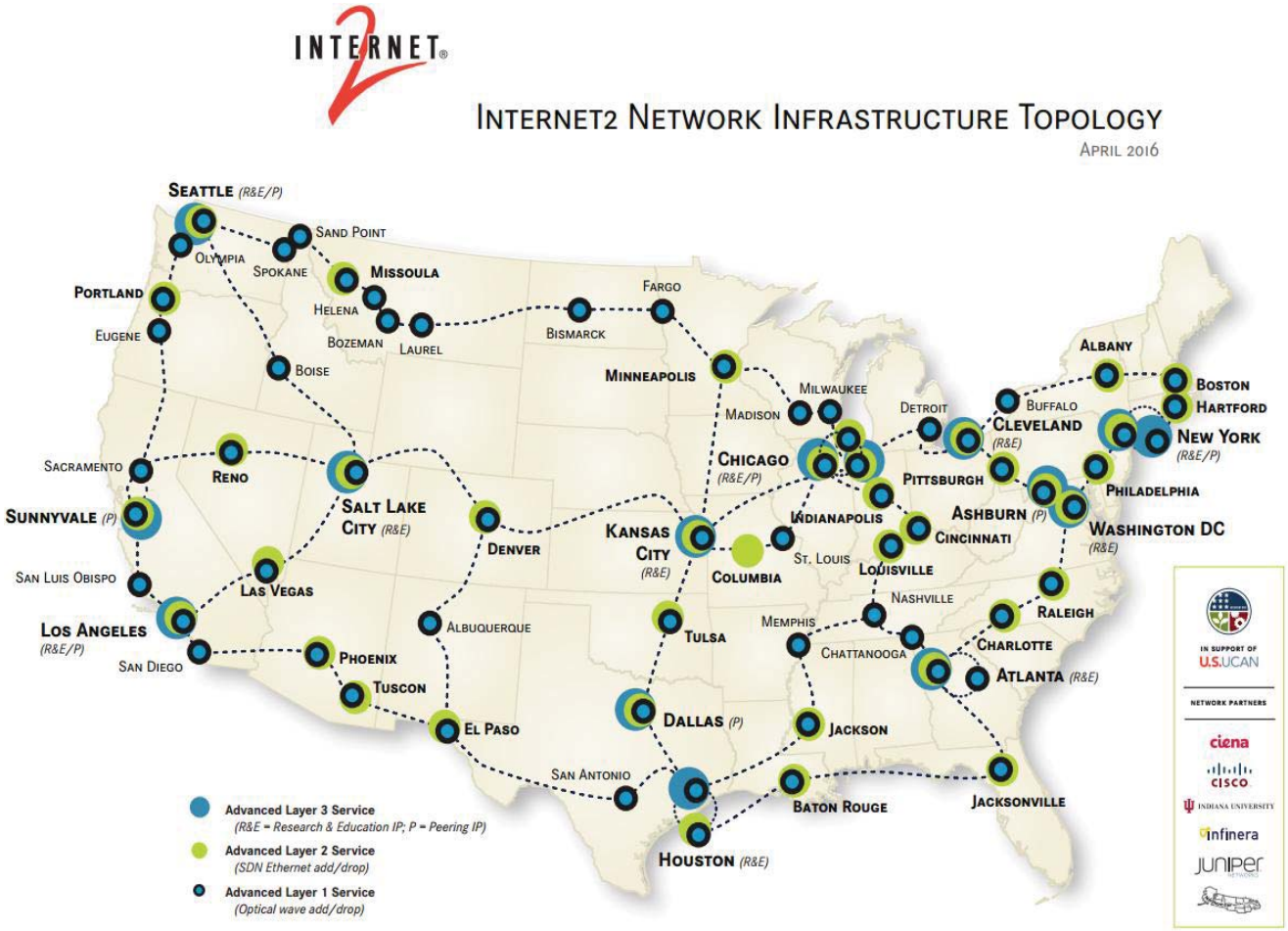

Figura 2.2: Infraestructura de la Internet2 al 2016. Muestra la infraestructura de la red dividida en varias capas para proporcionar un mejor servicio a sus usuarios. [35]

2.4 Red CLARA

A partir de la creación de INTERNET2 en 1996, nuevas redes se crearon bajo el mismo concepto, las cuales son lideradas por universidades y centros de investigación de cada país, el conjunto de todas ellas es denominado “redes avanzadas”. En América Latina, varios países formaron su propia red avanzada con la cual impulsar su desarrollo tecnológico, tal es el caso de México, que administra el Consorcio Universitario para el Desarrollo de INTERNET2 (CUDI) desde 1999. Sin embargo, el conjunto de redes avanzadas en América Latina son administradas por el Consorcio Latinoamericano de Redes Avanzadas (Red CLARA), asociación civil sin fines de lucro, con sede en Uruguay. [36]

El proyecto denominado “Conectando a Todos los Investigadores Europeos y Sudamericanos (CAESAR)”, financiado por la Comisión Europea, se desarrolló entre marzo y octubre de 2002 con la finalidad de estudiar las condiciones necesarias que permitieran establecer conexión directa entre Europa y América Latina. Mientras el proyecto CAESAR se desarrollaba, DANTE, corporación a cargo de la gestión de la red GÉANT, formó alianzas con las redes avanzadas de Europa (red Iris), Francia (RENATER), Italia (GARR) y Portugal (FCCN), para desarrollar el programa @LIS (Alianza por la Sociedad de la Información), liderado por el proyecto ALICE (América Latina Interconectada con Europa), para establecer el enlace entre los dos continentes, con la finalidad de acelerar el desarrollo de la Sociedad de la Información en América Latina, a través de una infraestructura que permitirá la colaboración en proyectos de investigación entre ambos países. El resultado obtenido a través de CAESAR fue que, Latinoamérica debía construir su propia red troncal, por lo que en junio de 2002, en el taller de Toledo realizado en la Universidad de Castilla, La Mancha, en Toledo, España, representantes de las redes latinoamericanas se comprometieron a cooperar en el desarrollo de la infraestructura que permitiría la investigación, educación e innovación, dando origen a la red CLARA. [37]

En Junio de 2003, en Valle de Bravo, México, 16 representantes de países latinoamericanos firmaron los estatutos para la creación de CLARA, con el objetivo

de integrar una red regional de telecomunicaciones de la más alta tecnología, para interconectar a las Redes Académicas Nacionales de la región. [38]

La Red CLARA fue constituida legalmente el 23 de diciembre de 2004, fecha en que la legislación de la República Oriental de Uruguay la reconoció como tal. CLARA fue definida como el sistema Latinoamericano de colaboración mediante redes avanzadas de telecomunicaciones para la investigación, la innovación y educación. Únicamente 5 de los 16 países que firmaron los estatutos dieron inicio al *Backbone* de CLARA, siendo estos: Argentina, Brasil, México, Panamá y Chile, conectados mediante enlaces de 155 Mbps, mientras la conexión internacional hacia Europa se había establecido con velocidad de 622 Mbps. En febrero de 2007, CLARA instaló un nuevo nodo en Miami, con un enlace desde Panamá, con capacidad STM-1 (155 Mbps sobre fibra óptica). Desde el nodo en Panamá se hicieron conexiones a los nodos de El Salvador y Guatemala, con capacidad de 10 Mbps. Posteriormente, el nodo en Miami fue conectado al nodo del proyecto WHREN/LILA (*Western Hemisphere Research and Education Network/Link Interconecting Latin America*), generando un enlace de Circuito Virtual Privado (VPN) ente Miami y Sao Paulo, cerrando un anillo en América del sur: Sao Paulo-Panamá-Santiago de Chile-Buenos Aires-Sao Paulo [39]. El proyecto WHREN/LILA se fundó en 2005, gracias al financiamiento de la Fundación Nacional para la Ciencia (NSF) y el proyecto Fapeps (Fundación de Amparo a la Investigación del Estado de Sao Paulo) de la red académica de Sao Paulo (ANSP), para establecer conexión de fibra obscura (fibra óptica que aún no se encuentra transportando información) entre Tijuana y San Diego, para conectar a la Red CLARA con la CENIC (*Corporation for Education Network Iniciatives of California* – Iniciativa de la Red de Cooperación para la Educación de California), y el enlace de 1,2 Gbps entre Miami y Sao Paulo. La renovación del proyecto ALICE2, inició en diciembre de 2008, con la finalidad de fortalecer la infraestructura de la red CLARA en Sudamérica. [40]

En 2010, el *Backbone* de la Red CLARA aumentó a 7 puntos de presencia, al anexar dos nodos más, uno en Lima y el otro en Miami. Diecisiete países Latinoamericanos formaban la membresía de CLARA: Argentina, Bolivia, Brasil, Colombia, Costa Rica, Cuba, Chile, Ecuador, El Salvador, Guatemala, Honduras, México, Panamá,

Paraguay, Perú, Uruguay y Venezuela, de los cuales sólo trece estaban conectados a CLARA. Los países no conectados eran Bolivia, Paraguay, Cuba y Honduras, de los cuales Cuba y Honduras eran considerados miembros pasivos. [41]

Tras la culminación del proyecto ALICE2, a finales de 2012, el enlace entre la Red CLARA y GÉANT se actualizó a 2.5 Gbps, aumentando cuatro veces su capacidad que había manejado desde su construcción. Por otro lado, en agosto del mismo año, se tendió la primera sección de fibra obscura de 1 Gbps en Centroamérica, que unía a San José, Costa Rica con la ciudad de Panamá. A finales de 2012 otra sección de fibra se tendió de San José a San Salvador [42].

En 2013 se lanzó el proyecto "Mesoamérica", con la finalidad de impulsar la cooperación de los países Centroamericanos, uniéndolos con México a través de enlaces de fibra óptica. El enlace entre Guatemala y Tapachula, México, se aumentó en 2014 a 2 Gbps, lo que permitió completar la red de fibra obscura tendida en los países de Centroamérica: México, Guatemala, El Salvador, Honduras, Nicaragua, Costa Rica y Panamá. Por otra parte, a finales de mayo de 2014, en la reunión anual del foro técnico de Red CLARA (CLARA-TEC), se presentó la propuesta para aumentar la capacidad del enlace entre la red CLARA en Sao Paulo, Brasil y GÉANT en Londres, Inglaterra, a 5 Gbps. La Comisión Europea presentó el proyecto BELLA (*Building Europe Link to Latin America* – Construyendo el enlace entre Europa y América Latina), con la visión de actualizar la red Latinoamericana a 100 Gbps. [43]

En 2015, CLARA se dedicó a realizar actualizaciones a su *backbone.* Los enlaces entre Santiago de Chile, Panamá y Sao Paulo fueron actualizados a 10 Gbps. Por otra parte, el proyecto BELLA-S (subproyecto que forma parte del proyecto general BELLA) fue designado para realizar la actualización del enlace entre la Red CLARA y GÉANT. [44]

Las actualizaciones al *backbone* realizadas en 2015, mejoraron su capacidad en la parte central, al contar con enlaces de 10 Gbps. En 2016, se logró concretar la conexión de Paraguay a través de un enlace a 100 Mbps, y se establecieron los acuerdos para reconectar a la Red Académica Nicaragüense, también se puso en marcha el proyecto BELLA-T para actualizar la troncal de la Red CLARA en América

del Sur, a una capacidad de 100 Gbps. Esencialmente, el proyecto comprende conectar a las ciudades de Fortaleza, Sao Paulo y Porto Alegre en Brasil, Buenos Aires en Argentina, Santiago en Chile, Guayaquil en Ecuador, Bogotá en Colombia, Cúcuta en la frontera de Colombia con Venezuela y Cartagena en Colombia.

La figura 2.3 muestra la topología de la Red CLARA, actualizada a enero de 2016, en ella se distinguen los enlaces a 10 Gbps actualizados principalmente en Sudamérica.

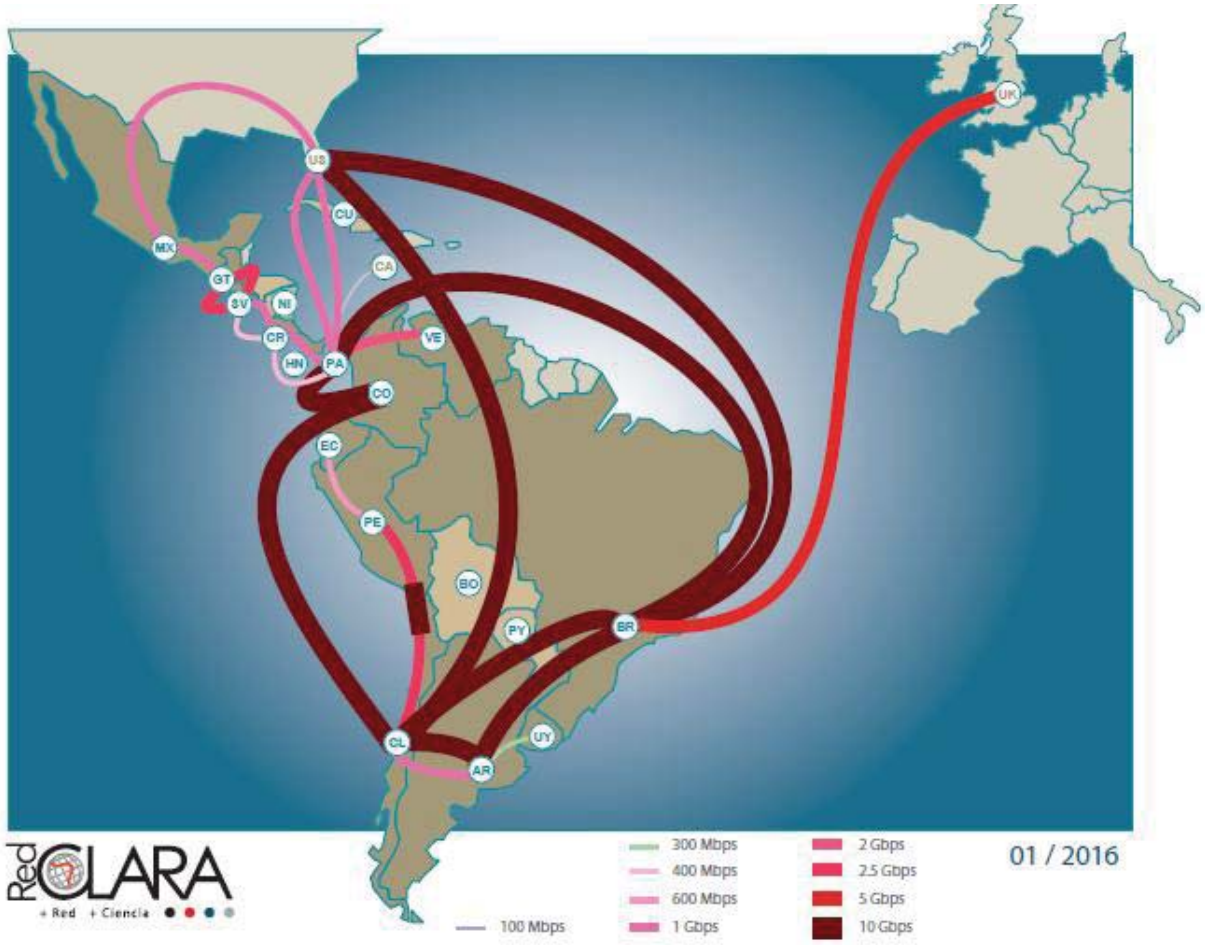

Figura 2.3: Topología de la red CLARA, enero 2016. Tomada de referencia [45]

2.5 Red CANARIE

2.5.1 Origen de la Internet en Canadá

La historia de la evolución de las redes de datos canadiense se narra en el libro "*A Nation goes Online"* [46], donde se especifica que Canadá inició su desarrollo casi al mismo tiempo que Estados Unidos, sin embargo, la investigación por separado en cada región canadiense daba como resultados sistemas de comunicaciones funcionales con protocolos propietarios, haciendo imposible la interacción entre las distintas redes desarrolladas. Una de las redes más avanzadas en 1970, fue la red implementada por el ministerio de educación de Quebec, la cual, utilizando una red con topología tipo estrella, logró interconectar escuelas primarias y secundarias con un *mainframe* (nodo central) ubicado en la sede del ministerio. La conexión se realizó a través de líneas telefónicas dedicadas. Por otra parte, la universidad de Quebec buscaba la forma de enlazar sus campus, mediante un diseño de red innovador, diferente al utilizado por el ministerio de educación. La tarea le fue encomendada a Joseph Reid, ingeniero en redes. Durante la ejecución de dicha tarea, Reid se dio cuenta de que Canadá debía ser conectada a través de una red única, la cual uniría a cada una de sus regiones, tal como lo venía haciendo ARPANET en Estados Unidos. Para lograr su cometido, generó una propuesta al Departamento de Comunicaciones Federal, la cual se firmó como "Red de Universidades Canadienses, CANUnet". En dicha propuesta, se solicitaba la unión con ARPANET, con la finalidad de utilizar su tecnología, sin embargo, la propuesta fue rechazada, debido a que no se deseaba la participación de intereses de Estados Unidos en territorio canadiense. Con el fracaso recibido, Reid continuó con su trabajo, logrando establecer en 1972 una red que conectaba a los campus de la universidad de Quebec con un *mainframe Control Data Cyber 6000.* Posteriormente, se mudó a Nueva Brunswick, donde construyó una nueva red de mayor longitud, la cual se conectó con la red existente en la Isla del príncipe Eduardo. La unión de ambas redes se conoce como NB/PEI ECN (Red Informática Educativa de Nueva Brunswick / Isla del príncipe Eduardo).

Para 1981, la Universidad de Columbia Británica, en Vancouver, dio a conocer su red denominada CDNnet, la cual utilizaba correo electrónico como medio de comunicación entre sus usuarios, administrado a través del *software* EAN (Diseñado

por Nefel, quien trabajó para el Consejo Consultivo Internacional Telegráfico y Telefónico, CCITT, donde desarrolló el estándar X.400. EAN fue incluido en el modelo de referencia OSI).

ARPANET se volvió popular, al incluir la conmutación de paquetes en su tecnología, permitiendo su expansión a gran escala. John Robinson y Keith Hooey, miembros del Centro de Investigaciones de Comunicaciones (CRC) en Ottawa, buscaban la forma de utilizar dicha tecnología, sin embargo, mantenían la firme idea de no replicar ARPANET dentro de Canadá, sino más bien, generar una colaboración Canadá – Estados Unidos, por lo que en 1983, acordaron una reunión con Vinton Cerf y Robert Kahn, para negociar los acuerdos de colaboración. Sin embargo, Canadá no contaba con la infraestructura suficiente para generar el enlace desde Washington hasta Ottawa. Robinson en colaboración con trabajadores de *Software Kinetics* se dedicaron a desarrollar una red IP sobre X.25 (IP/X.25) [47], la cual funcionaría a través del uso de minicomputadoras como la micro-VAX, la cual operaba como *router* (enrutador). La red fue puesta a prueba dentro de los laboratorios del Establecimiento de Investigación de la Defensa, ubicados en provincias del Atlántico en Dartmouth, Nueva Escocia, y los laboratorios del CRC, ubicados a las afueras de Ottawa. Para las pruebas se utilizaron computadoras PIN9101 conectadas a un conmutador de paquetes DATAPAC 3000 de Bell Comunicaciones.

La nueva tecnología permitió en 1983 la creación de la primera red canadiense de gran extensión territorial, la cual se conoce como DREnet (*Defense Research Establishment Network*). La red logró conectar a 20 ciudades de la región, así como la interconexión con ARPANET a través de un enlace que llegaba a los laboratorios Roma, en la ciudad de Roma en Nueva York. DREnet fue la primera red canadiense en utilizar herramientas de gestión, monitoreo, filtrado y control de acceso, a través del centro de operaciones establecido en Ottawa. [48]

Las Universidades de Guelph y Waterloo construyeron la red OUnet (Red de la Universidad de Ontario), utilizaba sistemas distribuidos conectados a un *mainframe* de IBM. La red permitió la conexión de la Universidad de Toronto, la Universidad de Nueva York en Toronto, la Universidad de Western Ontario en Londres, Queen's

Hamber College en Toronto y Ryerson. Posteriormente, la Universidad de Lakehead en Thunder Bay, la Universidad de Manitoba en Winnipeg, la Universidad McGill en Montreal y la Universidad de Nueva Brunswick también se unieron al proyecto. OUnet creció más allá de la región de Ontario, por lo que se le cambió el nombre a NETNorth (nombre dado como referencia al desarrollo tecnológico que se venía desarrollando al norte del país). El 7 de noviembre de 1985, con 21 miembros activos, NetNorth inició operaciones, estableciendo su sede en la Universidad de Toronto. La red logró interconectarse con otras redes, por lo que alcanzó los 65 miembros activos en 1989. Su rápido crecimiento se debió al uso de la tecnología TCP/IP, la cual fue permitida tras un acuerdo firmado por los canadienses como alternativa al rezago tecnológico que Canadá estaba sufriendo. NetNorth se convirtió en la primera red nacional canadiense, logrando conectar de costa a costa a Canadá, a través de dos topologías tipo anillo que se enlazaban en Montreal y Toronto. El control de la red fue asumido por la organización CA*net *Networking Inc.* (CA*net refiere a red canadiense, el asterisco en el nombre representa a una hoja de maple que es distinguible en el logotipo), creada por el Consejo Nacional de Investigación. CA*net asumió el control de la red el 27 de junio de 1990. Cabe destacar que el *backbone* contaba con una conexión de 56 Kbps, velocidad con la cual se enlazaba cada una de las regiones de Canadá. Además de conectar a cada región de Canadá, NetNorth estableció conexión hacia la NSFNET a través de Vancouver, Toronto y Montreal. [49]

2.5.2 CANARIE – Red Avanzada de Educación e Investigación Canadiense

En enero de 1993, el gobierno federal canadiense dio a conocer una nueva organización, creada para estimular la investigación y el desarrollo industrial en instalaciones y aplicaciones de redes de banda ancha. La nueva organización se presentó con el nombre de CANARIE (Red Canadiense para el Avance de la Investigación, la Industria y la Educación. La expansión del acrónimo ya no es utilizado por la organización), Estableció un año más tarde la NTN (*National Test Network* – Red Nacional de Pruebas), a través de la cual se brindó apoyo a la

industria, universidades, hospitales e instituciones gubernamentales, con la finalidad de que estas desarrollaran nuevas tecnologías, así como la construcción de *hardware* y *software* que agilizaran el desarrollo tecnológico. El *backbone* CA*net fue actualizado por primera vez en 1995, pasando de 56 Kbps a 10 Mbps. De forma casi inmediata, tuvo una nueva actualización a 20 Mbps. Sin embargo, en 1996, CA*net fue capaz de otorgar conexiones a 100 Mbps, a través del uso de la tecnología ATM (*Asynchronous Transfer Mode* – Modo de Transferencia Asíncrono), puesta en marcha en la NTN de CANARIE. La nueva tecnología se enfocó en resolver problemas de transferencia de voz, datos y video.

Los intereses comerciales a través de CA*net no se hicieron esperar, por lo que después de negociaciones, la junta directiva realizó la transición de una red de investigación y desarrollo, a una red comercial, otorgándole los derechos a Bell Canadá en 1997. Después de adquirir los derechos comerciales sobre la red, Bell Canadá realizó una nueva actualización al *backbone,* presentando CA*net II con una velocidad de conexión de 155 Mbps, además, estableció enlaces internacionales para conectarse con INTERNET2, a través del *backbone* vBNS de la NSF, y con la red Europea, a través de un enlace trasatlántico desde Halifax. En 1998, CANARIE presentó CA*net 3, primera red en su tipo a nivel mundial en utilizar la fibra óptica como medio de comunicación, la cual alcanzaba una velocidad de conexión de 2.5 Gbps. Sin embargo, la capacidad de la red no era suficiente para soportar el flujo de información generado por los miembros participantes, por lo que en 2002, el gobierno canadiense patrocinó una nueva actualización, surgiendo CA*net 4 con velocidad de conexión a 40 Gbps (OC-192), otorgando a sus usuarios conexiones punto a punto a través de enlaces de fibra óptica (*Lightpath*). [50]

CANARIE es el organismo encargado de proveer conexión interprovincial e internacional a las redes regionales y territoriales, brindando el acceso a datos y herramientas a través de la red principal. La red de Educación e Investigación canadiense está compuesta por doce redes regionales y territoriales que proporcionan conexión a universidades, colegios, institutos de investigación, hospitales y laboratorios de gobierno, para colaborar en programas de investigación y desarrollo a nivel mundial con redes similares. [51]

A continuación se listan las redes miembros a las cuales CANARIE proporciona servicios de red de alta velocidad.

1. ***ACORN-NL (Atlantic Canada Organization of Research Networks, Newfoundlan and Labrador):*** Está dedicada a la implementación y formalización de una red regional avanzada en Terranova y Labrador, para unir a sus principales centros de investigación, así como al sistema K-12 (es una designación utilizada en Norte América para referirse a la educación académica, desde la primaria hasta la secundaria). Proporciona conexión a 3 universidades, 2 colegios y 1 laboratorio de investigación del gobierno federal.
2. ***ACORN- NS (Atlantic Canada Organization of Research Networks, Nova Scotia):*** Se encarga de promover una red avanzada en Nueva Escocia que apoya directamente al sector salud, así como a la investigación y educación para impulsar el desarrollo económico. Conectadas a esta red se encuentran 18 universidades, 16 colegios, 3 laboratorios de investigación del gobierno federal, 11 hospitales de investigación y enseñanza, 1 incubadora de negocios (empresa que brinda asesoría para la creación de nuevos negocios) y 593 escuelas K-12.
3. ***Aurora College:*** Se encarga de suministrar los servicios de la red avanzada en los territorios del noroeste de Canadá a 29 colegios, 1 hospital de investigación y enseñanza y 50 escuelas K-12.
4. ***BCNET (Shared Services for Higher Education and Research in Brithish Columbia):*** Mantiene colaboración con miembros de educación superior para explorar y evaluar los servicios de tecnología, y con ello, acelerar la investigación. A través de esta red, CANARIE conecta a 22 universidades, 12 colegios, 8 laboratorios de investigación del gobierno federal, 9 hospitales de investigación y enseñanza y 1 incubadora de negocios para empresas.
5. ***Cybera (también conocida como ciberinfraestructura):*** Es un sistema avanzado de redes, creado para la investigación universitaria que ocupa la región de Alberta Canadá, permite la ejecución de proyectos piloto. Brinda conexión a 17 universidades, 5 colegios, 1 laboratorio de investigación del

gobierno federal, 2 incubadoras de negocios para empresas y 1,112 escuelas K-12.

6. ***MRnet (Manitoba's Regional Advanced Research and Education Network):*** Proporciona una conexión de alta velocidad hacia otras redes avanzadas mundiales. MRnet se encuentra en Manitoba, además de conectar a redes mundiales, también enlaza 8 universidades, 5 laboratorios de investigación del gobierno federal, 1 hospital de investigación y enseñanza, 2 incubadoras de negocios y 604 escuelas K-12.
7. ***New Brunswick Advanced Network:*** Proporciona una plataforma de red sustentable, económica y fiable de alta velocidad para las instituciones de educación e investigación. Se encarga de conectar a todas las redes de Canadá, así como a 12 universidades, 14 colegios, 1 laboratorio de investigación del gobierno federal y 5 hospitales de investigación y enseñanza.
8. ***ORION (Ontario Research and Innovation Optical Network):*** Es una de las redes más grandes enfocadas a la educación e investigación, proporciona una de las redes con mayor capacidad de banda ancha a nivel mundial. Conecta a 21 universidades, 22 colegios, 1 laboratorio de investigación del gobierno federal, 12 hospitales de investigación y enseñanza, 2 incubadoras de negocios y 2,316 escuelas K-12.
9. ***Prince Edward Island Advanced Network:*** La red CANARIE permite la conexión de la isla a través de Canadá y el resto de las redes avanzadas del mundo. En dicha red participan 1 universidad, 12 colegios y 1 laboratorio de investigación del gobierno federal.
10. ***RISQ (Quebec Scientific Information Network)*:** Es una red privada de telecomunicaciones que opera en Quebec apoyando la investigación y educación, su infraestructura se extiende por 6,000Km a través de Quebec para conectar a 15 universidades, 20 colegios, 30 CEGEPS (colegios de enseñanza general y profesional), 4 laboratorios de investigación del gobierno federal y 6 hospitales de investigación y enseñanza.
11. ***SRnet (Saskatchewan Research Network):*** Permite conectar Saskatchewan mediante enlaces de 1, 10 o 100 Gbps, para realizar investigación a través de

9 universidades, 42 colegios, 5 laboratorios del gobierno federal, 2 incubadoras de negocios y 774 escuelas K-12.

12. ***Yukon College:*** El centro de investigación del Yukón facilita los esfuerzos de sus comunidades para mejorar las oportunidades de educación mediante 15 colegios, 1 hospital de investigación y enseñanza y 28 escuelas K-12. [52]

2.5.3 *Backbone* de la Red CANARIE

CANARIE inició la creación de una red totalmente óptica en 2006, para conectar Canadá de costa a costa. La red contaba con infraestructura DWDM basada en dispositivos ROADM (*Reconfigurable Optical Add-drop Multiplexer* – Multiplexor Óptico de añadir y desechar Reconfigurable). La ROADM es un subsistema totalmente óptico que permite realizar configuración remota a cualquier *router* en la red, se caracteriza por utilizar un sistema que bloquea las longitudes de onda o hace una selección minuciosa para conmutarlas, si así se desea. La nueva infraestructura tenía la capacidad de trabajar con 88 longitudes de onda, con un ancho de banda de 100 Gbps por cada una de las longitudes de onda utilizadas. Las primeras actualizaciones se hicieron en el occidente de Canadá, cubriendo la ruta Seattle-Victoria-Vancouver-Calgary-Edmonton. En el este, se enlazó Chicago hasta Ontario y luego a Montreal, para conectarse con Nueva York, posteriormente, se tendió una sección independiente de Winnipeg a Thunder Bay. [53]

En 2010, la topología de CANARIE se definió en dos secciones para proporcionar conectividad mediante los protocolos de Internet IPv4 e IPv6. La topología principal que administraba ambos protocolos estaba compuesta por cinco *routers* (Calgary, Winnipeg, Toronto, Montreal y Halifax) conectados entre sí con siete enlaces internos y cinco externos, cada uno de estos enlaces utilizaba un *lightpath* con ancho de banda de 10 Gbps. De los segmentos externos, cinco se enlazaban a los puntos de intercambio *Pacific Wave* en Seattle, *StarLight* en Chicago y *MANLAN* en Nueva York, el resto se utilizaron para conectar con puntos de intercambio de Internet (IXP) y con proveedores de servicios (ISP) como TATA *Communications*, SIX y TorIX utilizando el protocolo IPv6.

En la figura 2.4 se muestra la distribución de las conexiones, internas y externas, para que los usuarios de CANARIE pudieran conectarse tanto a proveedores de Internet comercial, como a las redes de investigación y educación canadienses.

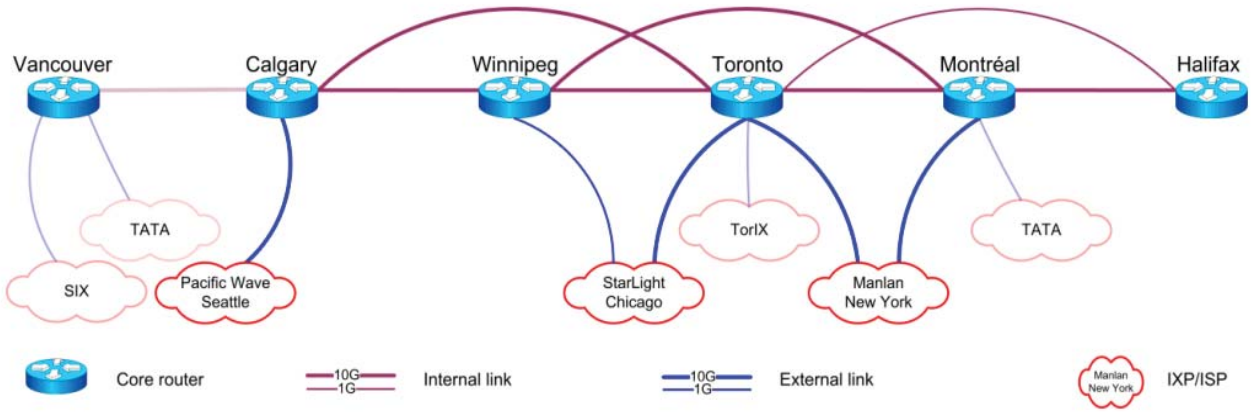

Figura 2.4: Enlaces internos y externos del *backbone* CANARIE. Agosto de 2010. [54]

El nuevo protocolo de Internet IPv6 se encuentra respaldado en las siguientes organizaciones: ARIN (*American Registry for Internet Numbers* – Registro Americano de números de Internet), ARIN es un registro regional de Internet, con sede en el estado libre asociado de Virginia, Estados Unidos, como organización sin fines de lucro, apoya el funcionamiento de la Internet, gestionando los números de Internet en toda su región de servicio [55]; el APNIC (*Asia Pacific Network Information Centre* – Centro de Información de las redes de Asía Pacífico), organización sin fines de lucro, cuya función principal es distribuir y administrar los recursos de números de Internet en las 56 regiones de Asía y el Pacífico [56]; y RIPE NCC (RIPE *Network Coordination Centre* - Centro de coordinación de redes IP europeas), organización sin fines de lucro, actúa como Registro Regional de Internet. [57]

En 2000, ARIN pidió a través de la TLA (*Top-Level Aggregation* – Agregación de Nivel Superior) la asignación del bloque de direcciones 2001:410::/35 a Viagénie (empresa de consultoría e investigación, especializada en tecnologías avanzadas de redes informáticas como IPv4 e IPv6) para que fuera utilizado por CANARIE. Como respuesta, el RIR (*Regional Internet Registry* – Registro de Internet Regional) otorgó los permisos para que CANARIE administre el bloque de direcciones IPv6 2001:410::/32. [58]

CANARIE y BCNET anunciaron el lanzamiento de IPv6 a las universidades e institutos como medida de respaldo ante el agotamiento de las direcciones de red IPv4. A través de un laboratorio de pruebas creado por CANARIE, BCNET y Cisco, se les otorga recursos en tiempo real a los administradores de red que están utilizando IPv6. El laboratorio es una plataforma de *hardware* compuesta por *switches, routers* y servidores virtuales, el cual permite realizar pruebas para establecer rutas, aplicaciones y servicios en la red IPv6. [59]

En junio de 2013, se presentó una nueva plataforma óptica, la cual utiliza equipos Ciena 6500. Los equipos cuentan con procesadores ópticos coherentes *WaveLogic* de tercera generación, para otorgar 100 Gbps de capacidad en la red. El primer segmento que se actualizó con esta nueva infraestructura fue de Montreal a Nueva York, permitiendo una mejor colaboración entre los investigadores nacionales e internacionales, así como el trasporte de grandes flujos de información generados en los centros de investigación canadienses. Las demostraciones de funcionamiento de la nueva infraestructura se realizaron en la conferencia de redes TERENA (TNC2013) donde se presentaron nuevas tecnologías y aplicaciones avanzadas para la ciencia, entre las cuales, resalta un brazo robótico manipulado por los asistentes a dicha conferencia, el robot contaba con retroalimentación háptica (tecnología que permite percibir la sensación de tacto real sobre alguna tecla al pulsar una pantalla plana), permitiendo manipular un mecanismo similar ubicado en un área geográfica distinta, la comprobación de la manipulación remota se demostró en video tridimensional transmitido en vivo a través de la red, sin editar. En 2014 se tendieron nuevas secciones de fibra óptica en la región central de Canadá, cubriendo Calgary-Regina-Winnipeg, con finalización en enero de 2015, mientras que en el Atlántico, se tendió fibra óptica de Montreal a Halifax, concluyendo la instalación en junio de 2015. [60]

En junio de 2016 CANARIE, en asociación con el consorcio *StarLight,* realizaron una demostración en la conferencia de redes TNC16, para mejorar la capacidad de la red utilizando interfaces abiertas y la modulación programable a través de equipos *Waveserver* de Ciena (*Waveserver* fue creado para funcionar como servidor, utilizado en aplicaciones de escala web, tiene la capacidad de entregar hasta 400

Gbps de ancho de banda) [61]. La infraestructura utiliza la modulación óptica coherente del *Chipset Wave Logic 3 Extreme* de Ciena [62]. CANARIE basó sus pruebas para utilizar únicamente 300 Gbps de los 400 Gbps disponibles en los nuevos equipos de Ciena. [63]

EL *backbone* de la red CANARIE, a marzo de 2016, se presenta en la figura 2.5. En ella, se especifican los GigaPoP que proporcionan conexión a cada una de las 12 redes regionales. También se puede observar, en color rojo, la nueva infraestructura de ultra alta velocidad (100 Gbps, proporcionada mediante los equipos Ciena 6500), así como el resto de la topología de alta velocidad.

Figura 2.5: *Backbone* de la Red CANARIE, marzo 2016. [64]

Capítulo 3: Protocolos de enrutamiento y gestión

3.1 Protocolos de enrutamiento

Una red de datos permite el flujo de información de extremo a extremo, donde los datos pueden recorrer diferentes caminos para alcanzar su destino. La red de datos podría verse como un laberinto, con múltiples opciones de salida, 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* (enrutadores). Para saber hacia dónde dirigir los datos, el *router* analiza los paquetes en busca de la dirección IP destino y posteriormente, busca en su tabla de enrutamiento el camino que conecta hacia dicho destino. La tabla de enrutamiento es un archivo utilizado para almacenar la información de las redes conectadas directamente al nodo, así como del nodo al otro extremo de la red, conocido como el siguiente salto. Los protocolos de enrutamiento se pueden ubicar en la capa 3 del modelo ISO/OSI, facilitando el intercambio de información entre *routers.*

Un protocolo de enrutamiento es un conjunto de algoritmos y mensajes utilizados para el intercambio de información de enrutamiento, con la finalidad de permitirle a cada *router* conocer la estructura de la red y, con ello, poder generar nuevos caminos hacia las redes destino en caso de que algún *router* falle o, se agreguen nuevos equipos a la red, modificando la topología.

Su propósito es descubrir nuevas rutas, a través del intercambio de información que realiza cada *router* con sus nodos vecinos.

Los protocolos de enrutamiento están formados por:

- **Base de datos:** Los protocolos de enrutamiento utilizan tablas o bases de datos almacenadas en la memoria RAM del *router*.
- **Algoritmo:** Es una lista limitada de pasos, utilizados para realizar una tarea de forma eficiente. Los algoritmos ayudan al *router* a determinar la mejor ruta para entregar información hacia otras redes de datos.
- **Mensaje del protocolo de enrutamiento:** Estos mensajes se utilizan para conocer a los nodos vecinos, intercambiar información y mantener la información del estado del enlace.

Los protocolos de enrutamiento se clasifican en estáticos, dinámicos e híbridos. A continuación se describe el uso de cada uno de ellos:

- **Enrutamiento estático:** Las rutas estáticas son asignadas en el *router* por el administrador de la red de forma manual, utilizando el mando *ip route,* estas rutas serán las únicas que conocerá el *router,* y será únicamente a través de ellas por donde se manden los paquetes de información. Es recomendable utilizar el enrutamiento estático únicamente cuando la red está constituida por pocos nodos, o sólo se posee un único proveedor de servicios de internet.
- **Enrutamiento dinámico:** El *router* se configura con alguno de los protocolos de enrutamiento dinámico para que las rutas sean aprendidas automáticamente, de esta forma se comparte el estado de la red y las posibilidades de conexión hacia las redes remotas. Cuando existen cambios de topología, los *routers* buscan nuevas rutas para alcanzar su objetivo. Una red remota es aquella que no está conectada directamente al nodo y, sólo se puede llegar a ella a través de otros nodos.

  Algunos de los protocolos existentes dentro de esta clasificación son:
  - **RIP** (*Routing Information Protocol* – Protocolo de Información de Enrutamiento) en sus tres diferentes versiones.
  - **IGRP** (*Interior Gateway Routing Protocol* – Protocolo de Enrutamiento de Puerta Interior)
  - **EIGRP** (*Enhanced Interior Gateway Routing Protocol* – Protocolo de Enrutamiento de Puerta Interior Mejorada).
  - **OSPF** (*Open Shortes Path First* - primero la ruta más corta).
  - **BGP** (*Border Gateway Protocol* – Protocolo de Puerta Exterior).

  Este tipo de protocolos se clasifican a su vez en:
  - **Vector distancia:** Este tipo de protocolos soporta su métrica en el conteo de saltos, es decir, en la cantidad de *routers* por los que debe pasar el paquete antes de llegar a su destino. Se toma en cuenta a la ruta que contenga el menor número de *routers* que conectan a una red destino, la cual es obtenida mediante el algoritmo Bellman-Ford. Los

protocolos que utilizan esta clasificación realizan difusión periódica de la información de enrutamiento. Dentro de esta clasificación se pueden encontrar a los protocolos RIP, IGRP y EIGRP.

- **Estado del enlace:** La métrica de este tipo de protocolos está soportado en el retardo existente en el enlace, ancho de banda, carga y confiabilidad de los enlaces que conectan con la red destino. Los *routers* que utilizan está denominación generan un mapa completo de la red, al recabar información procedente de los equipos que utilizan el mismo protocolo. Este tipo de protocolos solamente difunde la información de enrutamiento cuando existen modificaciones en la topología de la red. En esta clasificación se encuentran los protocolos OSPF e IS-IS (*Intermediate System to Intermediate System* – Sistema Intermedio a Sistema Intermedio).

- **Enrutamiento híbrido:** El *router* tiene la capacidad de manejar tanto la configuración de los protocolos de enrutamiento dinámico, como de las rutas estáticas asignadas por el administrador de la red. [65, 66]

Otro de los puntos importantes que atañe a los protocolos de enrutamiento es la métrica, y la distancia administrativa. La métrica es un valor cuantitativo, utilizado para medir la distancia hacia la ruta deseada, este valor depende de las características con las que se ha creado cada protocolo de enrutamiento. Por otra parte, los *router* Cisco tienen la capacidad de trabajar con diferentes protocolos al mismo tiempo, por lo que se ha diseñado un criterio de selección del mejor protocolo de enrutamiento a utilizar por el *router.* Un *router* que tiene la posibilidad de alcanzar una red a través de dos protocolos diferentes, debe poner en marcha el criterio conocido como distancia administrativa, la cual se encarga de analizar la prioridad de un protocolo sobre otro, esta prioridad está establecida dentro del rango 0 a 255. El protocolo que se utilizará para realizar el enrutamiento de la información será el que presente una menor prioridad. [67]

En la tabla 3.1 se presentan las distancias administrativas por defecto de los protocolos de enrutamiento soportados por Cisco.

Origen de la ruta	**Distancia administrativa**
Interfaz conectada directamente	0
Ruta estática	1
EIGRP sumarizada	5
BGP externa	20
EIGRP interna	90
IGRP	100
OSPF	110
IS-IS	115
RIP	120
EGP	140
ODR	160
EIGRP externa	170
BGP interno	200
Desconocida	255

Tabla 3.1: Valores de la distancia administrativa de los protocolos utilizados en *routers* Cisco.

A continuación se describen los protocolos de enrutamiento como son: RIP versiones 1 y 2, como parte introductoria para comprender el funcionamiento de protocolos más sofisticados como lo es OSPF en sus versiones 1 y 2, del cual se hace una reseña más extensa por ser el protocolo de enrutamiento a utilizar en las pruebas de conectividad del *backbone* CANARIE. Además, también se analiza el protocolo de gestión de redes SNMP, en sus dos primeras versiones. Cabe destacar que el análisis de los protocolos de enrutamiento y gestión está limitado al uso de IPv4.

3.1.1 Protocolo de Información de Enrutamiento - RIP

RIP es uno de los primeros protocolos en ser estandarizados, utiliza el algoritmo vector distancia, escrito por R. E. Bellman, L. R. Ford y D. R. Fulkerson, para conseguir las mejores rutas hacia las distintas redes destino. El primer protocolo vector distancia implementado fue el GIP (*Gateway Information Protocol* – Protocolo de Información de Puerta), protocolo propietario, diseñado por la compañía Xerox e incluido en la arquitectura XNS (*Xerox Network System*) para intercambiar información entre diferentes sistemas autónomos.

La variante del protocolo GIP fue presentada por la Universidad de California en Berkeley, a la cual llamaron *routed,* distribuida mediante BSD (*Berkeley Standard Distribution*), presentó modificaciones como un temporizador de 30 segundos para limitar el tiempo máximo de actualización, y una ventana que permitía modificar las direcciones de red, a demás fue incluido en la plataforma de UNIX, volviéndolo un protocolo abierto.

RIP se dio a conocer en el RFC (*Request For Comment*) 1058, por Charles Hedrick, en 1988 [68]. Actualmente se encuentran dos nuevas versiones; RIPv2 publicada en 1994 y RIPng (RIP de próxima generación para utilizarse con IPv6. Esta versión no forma parte del desarrollo de esta tesis) publicada en 1997, en el RFC 2080 [69].

RIP es un protocolo de enrutamiento interno (IGP) del tipo vector distancia, utiliza como métrica el conteo de saltos (*Hop Count*), soportada en la cantidad de *routers* existentes desde donde se origina la información hasta donde será entregada. La ruta óptima considerada por RIP, es aquella que contiene el menor número de nodos hasta una red destino. RIP fue diseñado para trabajar dentro de redes pequeñas, tolerando un máximo de 15 nodos en la topología. Una red destino con métrica mayor a 15 nodos es considerada por RIP como inalcanzable. La actualización de las tablas de enrutamiento se realiza a través del puerto 520 de UDP (*User Datagram Protocol* – Protocolo de Datagrama de Usuario) utilizado como protocolo de transporte. RIP cuenta con una serie de temporizadores para tomar decisiones referentes al enrutamiento de información, estos temporizadores son:

- ***UPDATE*** (actualización): Establece el período de tiempo en que se debe realizar la difusión de información, sin importar que la topología sufra modificaciones o no. Por defecto, la difusión de información se realiza cada 30 segundos.
- ***INVALID*** (Inválido): Después de recibir una actualización, las rutas se almacenan en la base de datos, si después de transcurridos 180 segundos una de las rutas no ha sido actualizada, entonces se declara como inalcanzable.
- ***FLUSH*** (Limpiar): Define el tiempo que debe transcurrir después de declarar una ruta como inalcanzable, este valor se establece en 240 segundos, el cual, una vez transcurrido, la ruta es eliminada de la tabla de enrutamiento. [70]

3.1.2 RIPv1

La primera versión de RIP no fue diseñada para trabajar con sistemas autónomos, ni con la segmentación de la red para utilizar subredes (máscaras de subred de longitud variable, VLSM), tampoco para realizar enrutamiento entre dominios sin clase (CIDR), carece de métodos de autenticación, por tal motivo, RIPv1 es considerado un protocolo con clase, al asignar las direcciones de red en función de las clases de red IP: A, B y C [71]. La máscara de red no se incluye en los mensajes de actualización, debido a que puede fácilmente obtenerse al leer el primer octeto de la dirección de red. La difusión de la información de enrutamiento se realiza mediante *broadcast* de forma periódica cada 30 segundos.

Los mensajes RIP son encapsulados en datagramas UDP (*User Datagram Protocol* – Protocolo de Datagrama de Usuario), es un protocolo de la capa de transporte, utilizado para el intercambio de datagramas a través de la red, sin que exista previa conexión entre dos equipos que se desean comunicar [72]. UDP consta de un tamaño de 8 *bytes* con cuatro campos: puerto origen y puerto destino, encargados de identificar el proceso de emisión y recepción, ambos utilizan el puerto 520 cuando se utiliza el protocolo RIP. El siguiente campo es la longitud UDP, encargada de mostrar la longitud del datagrama UDP. Por último, un campo que tiene la suma de comprobación (*checksum*) que verifica la integridad de los datos envidos.

0	16	31
Puerto origen	Puerto destino	
Longitud UDP	checksum	
Datos		

Tabla 3.2: Cabecera UDP utilizada por RIP.

El formato de mensaje utilizado por RIPv1 se presenta en la tabla 3.3. De acuerdo con el RFC 1058, el tamaño máximo del datagrama es de 512 bytes, sin incluir las cabeceras de IP y UDP. El mensaje que se envía para actualizar la información de enrutamiento tiene la capacidad de albergar hasta 25 rutas, en las cuales se incluye información de cada una de ellas.

0	8	16	31
Command	Version	Must be zero	
Addres Family Identifier		Must be zero	
IP address			
Must be zero			
Must be zero			
Metric			

Tabla 3.3: Formato de mensaje RIPv1, utilizado para actualizar la información de enrutamiento.

Las funciones de cada campo de la tabla 3.3 se especifica a continuación:

- Mando (***Command***): Indica el tipo de mensaje RIP. En un inicio se definieron cinco tipos de mandos, sin embargo, durante las actualizaciones realizadas al protocolo, sólo dos tipos de mensajes han permanecido: petición y respuesta, el resto se volvió obsoleto. El mando de petición es utilizado por un *router* para solicitar información de enrutamiento a sus nodos vecinos, estableciendo el valor del campo en 1. Para generar una respuesta a la petición, o realizar la distribución de la información de forma automática, el valor del campo se establece en 2.
- Versión (***version***): Este campo se encarga de especificar la versión del protocolo que se está utilizando, el cual debe ser 1 para la primera versión, en caso de que el campo tenga un valor cero, el mensaje es ignorado.

- Identificador de familia de direcciones (***Address family identifier***): Este campo fue establecido para identificar la existencia de rutas procedentes de protocolos diferentes a RIP. Cuando se difunde la información de enrutamiento, este campo se establece en 2, para indicar que contiene direcciones IP, cuando se solicita una actualización de la información de enrutamiento, el campo se establece en cero.
- Direcciones IP (***IP address***): se encarga de describir las rutas que conectan a diferentes redes destino.
- Métrica (***metric***): lleva el conteo de saltos que ha realizado el mensaje antes de ser entregado. La métrica es un número entre 1 y 16, donde 16 significa que la red es inalcanzable y por lo tanto no se puede establecer conexión.
- Campos cero (***must be zero***): Son espacios asignados para una futura actualización del protocolo.

3.1.3 RIPv2

La segunda versión de RIP apareció en 1994, en el RFC 1723, publicado por G. Malking. Más tarde, en 1998, fue actualizado en el RFC 2453 [73]. Las mejoras que aparecieron en esta nueva versión incluyen el uso de máscaras de red de longitud variable, así como un sistema de seguridad, que garantiza la integridad de la difusión de las tablas de enrutamiento, evitando que estas sean clonadas en otros *routers* que no pertenecen a la topología de la red. Las mejoras fueron ubicadas en los campos cero existentes en el formato del mensaje RIPv1. El formato de mensaje para RIPv2 se presenta en la tabla 3.4, en ella se resalta en color gris los campos que se han incluido en esta nueva versión.

0	8	16	31
Command	Version	Unused	
Addres Family Identifier		Route Tag	
IP address			
Subnet Mask			
Next hop			
Metric			

Tabla 3.4: Formato del mensaje RIPv2.

La función de los nuevos campos presentados en la versión dos de RIP se describe a continuación.

- Etiqueta de ruta (***route tag***): es utilizada para diferenciar las rutas de protocolos diferentes a RIP. El valor del campo es puesto en cero cuando únicamente se tienen rutas internas, cuando se obtiene una ruta procedente de un protocolo externo (diferente a RIP), el campo adquiere el valor del sistema autónomo que generó la ruta.
- Máscara de subred (***subnet mask***): incluye direcciones de subred con la finalidad de utilizar máscaras de longitud variable.
- Siguiente salto (***next hop***): Es una notificación que indica la distancia a la que se encuentra una ruta.

Cuando el administrador de la red decide utilizar el sistema de autentificación que se incluye en RIPv2, los mensajes de actualización sólo podrán difundir 24 rutas, una menos que cuando no se utiliza el sistema de seguridad, esto debido a que dicho sistema consta de 20 *bytes*. La tabla 3.5 muestra el mensaje de RIPv2 con el sistema de autentificación activado. Los campos asignados para el sistema de seguridad se resaltan en color gris.

0	8	16	31
Command	Version	Unused	
0xFFFF		Authentication Type	
Authentication			
Address Family Identifier		Route Tag	
IP address			
Subnet Mask			
Next hop			
Metric			

Tabla 3.5: Formato del mensaje RIPv2 utilizando autentificación.

El campo identificador de la familia de direcciones cuando se encuentra etiquetado como “0xFFFF”, específica que se está utilizando un sistema de seguridad, el cual puede ser una contraseña en texto plano o bien, una contraseña codificada mediante MD5 (*Message-Digest Algorithm 5* – Algoritmo de Resumen del Mensaje 5, el cual es un algoritmo de reducción criptográfico de 128 bits, usado para verificar que algún

archivo no se haya modificado. Fue diseñada por Ronald Rivest, en 1991). El método de autentificación es definido en el campo *authentication Type,* el cual puede tomar los siguientes valores: cero; para indicar que no se utiliza contraseña; 2 para indicar que se utiliza alguna contraseña. El campo *Authentication* contiene la contraseña asignada por el administrador de la red.

A diferencia de RIPv1, la versión 2 utiliza la difusión de la información de enrutamiento mediante *multicast,* utilizando la IP 224.0.0.9, sin embargo, la métrica y el período de tiempo para realizar la distribución de información de enrutamiento siguen siendo los mismos que utiliza la primera versión. [74]

3.2 OSPF

El protocolo OSPF (*Open Shortes Path First* – Primero la Ruta más Corta) es un protocolo de ruteo interior (IGP), dado a conocer en 1989, en el RFC 1131 [75]. Se diseñaron dos implementaciones, la primera para ejecutarse en *routers* y la segunda para ejecutarse en estaciones de trabajo UNIX, esta última se convirtió en un proceso del sistema conocido como GATED. La primera versión de OSPF fue construida como un sistema de pruebas, por lo que no se llevó a la implementación. En su lugar, apareció la versión 2, publicada en 1991, en el RFC 1247, actualizándose posteriormente en 1998, en el RFC 2328 [76]. La versión OSPF para utilizar el protocolo IPv6 (versión no implementada en el desarrollo de esta tesis) se publicó en 1999, en el RFC 2740 y su más reciente actualización se realizó en 2008, en el RFC 5340. OSPF es un protocolo abierto, que se ejecuta sobre IP utilizando el puerto 89, se basa en el algoritmo primero la ruta corta (SPF), mejor conocido como Dijkstra, el cual es utilizado para conseguir la mejor ruta a un destino. La información de enrutamiento se adquiere con base al estado del enlace existente entre los nodos, haciendo una descripción de la interfaz y la relación que mantiene con sus vecinos. Los *routers* configurados con OSPF, crean un mapa completo de la red, al recabar la información almacenada en sus bases de datos. La base de datos de cada nodo se construye a través de las notificaciones de enlace, enviadas en forma de paquete a cada nodo vecino, estas notificaciones se conocen como LSAs (*Link-State Advertisement* – Notificaciones del Estado del Enlace).

OSPF tiene la capacidad de trabajar con redes de gran extensión, generando un mapa completo de la red, lo que en algún momento podría llegar a saturar la base de datos de los *routers,* al construir las tablas de enrutamiento con las mejores rutas*,* desencadenando errores en el enrutamiento de los paquetes. Para evitar que exista una excesiva cantidad de información de enrutamiento en un *router*, OSPF utiliza el concepto de área, es decir, una red puede dividirse en varias áreas dedicadas a administrar una cantidad menor de *routers.* El administrador de la red decide la cantidad de áreas a utilizar, sin embargo, OSPF define la existencia obligatoria de por lo menos un área, a la cual se le conoce como área cero o de *backbone.* Esta área funciona como “puente” de comunicación entre las distintas áreas que el

administrador haya definido, las cuales intercambian información entre sí. Las distintas áreas deben mantener conexión ya sea de forma física o lógica (*virtual link*) con el área cero.

Un área es una colección de redes, *routers* y enlaces que contienen el mismo identificador de área, comparten la misma información de enrutamiento y se utilizan para limitar el alcance de la distribución de la información de enrutamiento. OSPF permite asignar un número de área entre 0 y 4,294,967,296 $(2^{32}-1)$. [77]

Algunos de los conceptos utilizados por OSPF son:

- ***Router* ID:** Es una dirección IP asignada a un *router* que funciona como identificador, con la finalidad de distinguir al *router* del resto de los *routers* que conforman una topología de red OSPF. Si el identificador no es asignado por el administrador de la red, este toma como ID el valor de la dirección de red de interfaz de mayor rango configurada.
- ***Router* vecino:** Dos *routers* se vuelven vecinos cuando están dentro de la misma área y además comparten los mismos intervalos en los mensajes *Hello* y *dead,* así como la misma clave de autentificación.
- ***Router* adyacente:** Es la relación existente entre dos *routers,* formada para intercambiar información de enrutamiento.
- **Paquete *Hello*:** Se encargan de mantener la relación de vecinos activa.
- **Notificaciones del Estado del Enlace (LSA):** son mensajes encargados de indicar el estado en el que se encuentran los enlaces de cada *router*.
- ***Router* Designado (DR):** Es un *router* elegido dentro de un área para administrar el flujo de información de los LSA generados por el resto de *routers* existentes dentro de dicha área.
- ***Router* Designado de Respaldo (BDR):** Este *router* se encarga de sustituir al DR en caso de falla.

El RFC 2328 establece que un *router* que utiliza OSPF como protocolo de enrutamiento, debe seguir una secuencia de ocho pasos hasta construir las tablas de enrutamiento. Este proceso permite conocer a los *routers* vecinos a través de

mensajes de saludo (*Hello*), los cuales son enviados de forma periódica cada 10 segundos mediante difusión *multicast,* a través de la IP 224.0.0.5. Los mensajes de saludo tienen como propósito lograr que dos *routers* formen adyacencia y comiencen el intercambio de información de las tablas de enrutamiento.

Los pasos que sigue un *router* OSPF hasta completar la tabla de enrutamiento mediante el establecimiento de adyacencia, se describen a continuación.

1. **Inactivo (*Down*)**: Los nodos se encuentran en estado pasivo, no reciben ni envían información.
2. **Attempt:** Se le envían mensajes de saludo (*Hello*) a un nodo vecino, con la finalidad de conocer su disponibilidad para formar adyacencia. Este estado existe únicamente en redes NBMA (*No Broadcast Multi Access Network*) en la cual los vecinos se configuran manualmente y los mensajes de saludo (*Hello*) son enviados mediante *unicast*.
3. **Inicio (*Init*):** Un *router* detecta en alguna de sus interfaces activas un paquete de saludo (*Hello*) procedente de algún posible vecino. Dicho saludo contiene información del *router* que generó el mensaje.
4. **Bidireccional (*two-way*):** El *router* que recibió el saludo responde de forma similar al *router* que inició la comunicación. En la respuesta se incluye la información de ambos *routers.* Si ambos *routers* comparten criterios como el intervalo de saludo e intervalo *dead*, se convierten en vecinos y se inicia la comunicación entre ambos, estableciendo el oden de cuál de ellos será el (*Designated Router – Router* Designado) y, cual el BDR (*Backup Designated Router – Router* Designado de respaldo).
5. **Inicio de intercambio (*Exstart*):** En este estado se establece el número de secuencia inicial para el intercambio de información. El nodo con ID mayor se convierte en el primario (maestro), estableciendo la secuencia de sincronización de LSDB (*Link State Database* – Base de datos del estado del enlace), mientras que el otro *router* pasa a ser el secundario (esclavo).
6. **Intercambio (*Exchange*):** Ambos *routers* establecen un número de secuencia que se anexa a los paquetes enviados en el intercambio de

información mediante mensajes de descripción de la base de datos. Con este proceso se garantiza que la información recibida sea la más reciente.

7. **Cargando (*Loading*):** Se actualiza la base de datos con la nueva información. Si algún dato llega incompleto, se vuelve a solicitar la información a través de un LSR (*Link State Request* – Petición del Estado del Enlace).
8. **Completo (*Full*):** Se establece la adyacencia completa tras recibir y sincronizar toda la información de los nodos vecinos. [78]

La figura 3.1 muestra una representación grafica del proceso que realizan los *routers* configurados con OSPF para lograr la adyacencia.

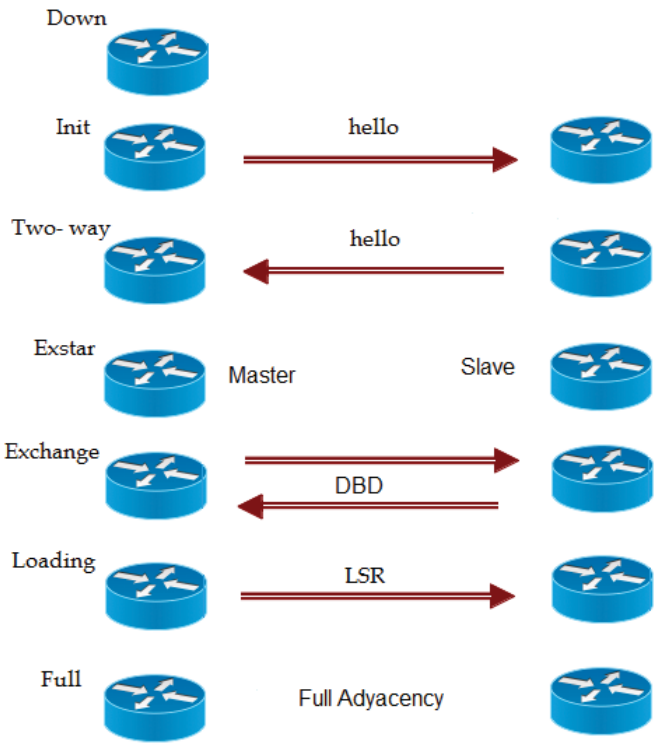

Figura 3.1: Principales estados por los que pasa un *router* OSPF para formar adyacencia con sus vecinos. Diagrama propio con base en referencia [79]

3.2.1 *Routers* OSPF

El protocolo OSPF define una serie de *routers* a través de los cuales administra el flujo de información que pasa por la red. Para evitar la sobrecarga de información de enrutamiento, OSPF la descentraliza en diferentes *routers,* estos son*:*

- **DR (*Designated Router – Router* Designado):** La función del DR es trabajar como elemento central, recolectando las notificaciones del estado del enlace que el resto de *routers* envían a través de *multicast* 224.0.0.6. Los LSA son redistribuidos a cada *router* de la red con la finalidad de que cada uno actualice su base de datos.
- **BDR (*Backup Designated Router – Router* Designado de Respaldo):** Para proporcionar tolerancia a fallo del DR, se asigna el BDR para tomar el lugar del DR en caso de falla.
- ***IR* (*Internal Router – Routers* Internos):** Se encuentran dentro de una misma área, por lo que comparten la misma información de la base de datos.
- ***BR* (*Backbone Routers*):** Son todos aquellos *routers* que mantienen al menos una interfaz conectada al área cero.
- **ABR (*Area Border Router – Router* Fronterizo):** Estos *routers* se conectan a distintas áreas y construyen una base de datos por cada área a la que se encuentra conectado.
- **ASBR (*Autonomous System Boundary Routers – Router* Frontera del Sistema Autónomo):** Estos *routers* tienen conexión con al menos un AS distinto, el cual no necesariamente utiliza OSPF como protocolo de enrutamiento. Estos *routers* distribuyen la información dentro y fuera de una red OSPF.

Los DR y BDR son los *routers* más importantes definidos en OSPF, dedicados a recibir toda la información de notificaciones del estado del enlace, por lo que deben contener una gran cantidad de recursos de memoria y CPU para poder distribuir toda la información que se reciba de los *routers* de la red. Existen dos formas de seleccionar al *router* DR.

1. Un *router* se elige DR si tiene la prioridad de interfaz más alta, mientras que el segundo *router* con prioridad más alta será el BDR. Por defecto, la prioridad de las interfaces de un *routes* está configurada en 0, sin embargo, puede modificarse utilizando el mando "*ip ospf priority"* para establecer un valor entre 0 y 255. Mientras más alta sea la prioridad, existen mayores probabilidades de que el *router* sea elegido como DR.
2. Si todos los *routers* tienen la misma prioridad, se elige como criterio el identificador de *router* (*router* ID) de mayor rango. El identificador de *router* se puede establecer en tres diferentes formas:
   a) El identificador puede configurarse manualmente.
   b) Si no existe un ID configurado, este se determina utilizando la dirección IP de *loopback* más alta.
   c) Si no existe dirección de *loopback,* se asigna como identificador a la dirección de red IPv4 más alta existente en las interfaces del *router.*

3.2.2 Cabecera del mensaje OSPF

Los paquetes OSPF se deben enviar siempre con el campo ToS (*Type of Service* – Tipo de Servicio) de IP puesto en cero, para dar preferencia sobre el tráfico de datos IP. Cada paquete OSPF comparte el mismo formato, el cual inicia con un encabezado de 24 bytes. El encabezado contiene la información necesaria para determinar si el mensaje será aceptado para su procesamiento, o no. El formato del mensaje OSPF se muestra en la tabla 3.4.

0	8	16	31
Version	Type	Packet Length	
Router ID			
Area ID			
Checksum		AuType	
Authentication			
Authentication			
Message Body			

Tabla 3.4: Cabecera OSPF.

Cada uno de los campos existentes en la cabecera OSPF se describe a continuación:

- Versión **(*Version*):** Este campo indica la versión que se está utilizando del protocolo OSPF.
- Tipo **(*Type*):** Específica el tipo de paquete que se está enviando. El paquete puede ser saludo (*Hello*) cuya notación es 1; una descripción de la base de datos, con notación 2; una petición del estado del enlace (LSR), con notación 3; una actualización del estado del enlace, con notación 4, o bien, un acuse de recibo (LSAck), con notación 5. Cada uno de los 5 paquetes mencionados cumplen con una función específica, las cuales se definirán más adelante.
- Longitud del paquete **(*Packet Length*):** Contiene la longitud del paquete, incluyendo la cabecera del protocolo.
- ***Router ID*:** en este campo se incluye el identificador del *router* que generó el mensaje.
- ***Area ID*:** Específica el área donde fue generado el mensaje.
- ***Checksum*:** Es la suma de comprobación de todo el paquete, excluyendo el campo de autenticación. Se utiliza para comprobar la integridad del paquete.
- ***AuType (Authentication Type)*:** Identifica el tipo de autenticación que se utilizará, ya sea que no se utilice, poniendo el valor del campo en 0, o bien, que se cuente con una clave, poniendo el valor del campo en 1, además, se puede utilizar otro sistema de seguridad como lo es el MD5, en este caso el campo toma un valor de 2.
- ***Authentication*:** Este campo contiene el valor de la contraseña que ha definido el administrador de la red.

3.2.3 Tipos de mensajes OSPF

El protocolo de enrutamiento OSPF utiliza cinco tipos de mensajes, mediante los cuales se realiza el intercambio de información del estado del enlace, haciéndolo un protocolo de excelencia para trabajar con redes sofisticadas y complejas.

Los mensajes utilizados por el protocolo para el intercambio de información son:

1. Saludo (*Hello*)
2. Descripción de la base de datos (DBD)
3. Petición del estado del enlace (LSR)
4. Actualización del estado del enlace (LSU)
5. Acuse de recibo del estado del enlace (LSAck)

A continuación se describe el funcionamiento, y el formato del mensaje utilizado por cada uno de los cinco mensajes OSPF.

3.2.3.1 Paquetes *Hello*

Estos son considerados paquetes de tipo 1, se distribuyen mediante *multicast* de forma periódica cada 10 segundos a través de todas las interfaces activas de un *router,* con el propósito de establecer y mantener la relación de vecinos. Un *router* es considerado vecino cuando los siguientes parámetros coinciden: máscara de red, intervalo del mensaje *hello* y el tiempo de vida de la ruta, en caso contrario dos *router* no se establecerán como vecinos. Cuando un *router* envía este tipo de mensajes, establece el valor en 1 dentro del campo tipo (*Type*) de la cabecera OSPF. La tabla 3.5 muestra el formato utilizado por el paquete *Hello*, el cual se incluye después de la cabecera de OSPF dentro del campo designado para el cuerpo del mensaje (*message body*) de la cabecera OSPF.

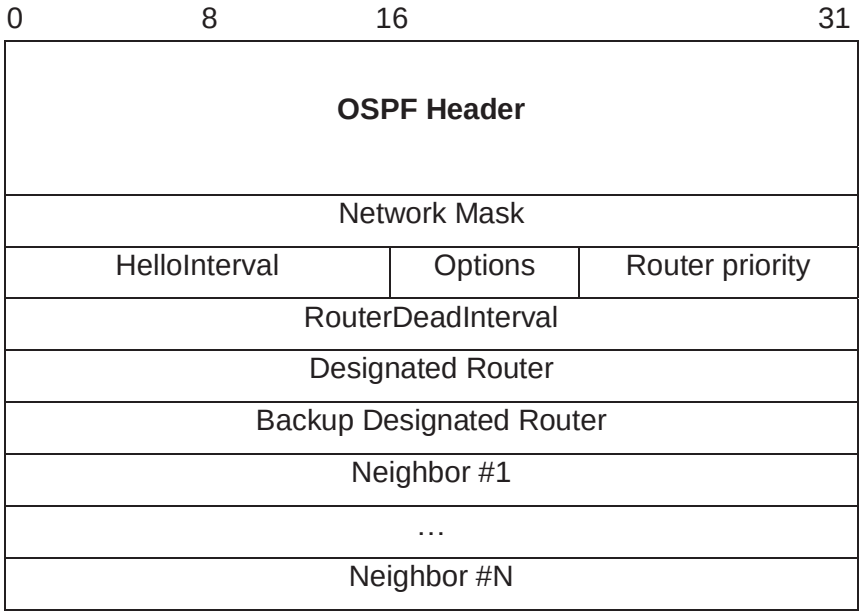

Tabla 3.5: Formato del paquete *Hello.*

La función de cada uno de los campos del formato del mensaje *Hello* se describen a continuación:

- Máscara de red **(*Network mask*):** Contiene la máscara de red que utiliza el *router* que generó el mensaje.
- Intervalo *Hello* **(*HelloInterval*):** Establece el período de tiempo en segundos en los cuales se manda el mensaje *Hello.* Por defecto está establecido en 10 segundos.
- Opciones **(*options*):** En este campo se decide si se rechaza a un *router* candidato a ser vecino, desechando el mensaje debido a un desajuste de capacidades. El campo utiliza los siguientes *bits*:
  - **E:** este *bit* define la forma en cómo se realiza la distribución de las rutas procedentes de AS externos.
  - **MC:** Indica si los datagramas han sido reenviados.
  - **N/P:** Específica la forma en cómo se deben manejar los LSA soportados por los equipos Cisco.
  - **EA:** Indica la disponibilidad del *router* para enviar y recibir atributos externos mediante LSAs.
  - **DC:** Describe el manejo de los circuitos de demanda del *router.*

La tabla 3.6 muestra el campo opciones, seccionado por los *bits* descritos anteriormente.

*	*	DC	EA	N/P	MC	E	*

Tabla 3.6: Campo opciones perteneciente al paquete *Hello.*

- Prioridad del *router* **(*Router priority*):** Esta indica la prioridad del *router* local, la cual es utilizada para elegir al DR y al BDR.
- Intervalo muerto del *router* **(*RouterDeadInterval*):** Es el período de tiempo que debe transcurrir desde el último *Hello* recibido antes de declarar la adyacencia pérdida. Por defecto, se establece en 40 segundos.

- *Router* Designado **(*Designated Router*):** Contiene la dirección IP del *router* designado.
- *Router* Designado de respaldo **(*Backup Designated Router*):** Indica la dirección IP del *router* designado de respaldo.
- Vecinos **(*Neighbors*):** Contiene el ID de todos los *routers* que se han establecido como vecinos del *router* que generó el mensaje.

3.2.3.2 Paquetes de descripción de la base de datos (DBD)

Se clasifican como paquetes tipo 2, utilizados para intercambiar la información de la base de datos cuando se está estableciendo la adyacencia. Para la distribución de la información del estado de la base de datos, OSPF se encarga de distribuirla en forma separada a través de notificaciones del estado del enlace (LSA) pertenecientes a la base de datos. La tabla 3.7 muestra el formato del mensaje DBD.

0	8	16							31
OSPF Header									
Interface MTU	Options	0	0	0	0	0	I	M	MS
DB Sequence Number									
An LS Header									
…									

Tabla 3.7: Formato del mensaje DBD.

Los campos del mensaje de la descripción de la base de datos se refieren a:

- **Interface MTU** (*Maximum Transmission Unit* – Unidad de Transferencia Máxima): Contiene el valor MTU de la interfaz por donde se envía el paquete.
- Opciones **(*options*):** Este campo es similar al utilizado en el paquete del mensaje *hello.*
- **I** (*Initial bit* – bit inicial): Indica el bit inicial en una serie de paquetes de DBD.
- **M** (*more bits* – más bits): Indica que posición ocupan los paquetes DBD.

- **MS** (*master/slave* – maestro/esclavo): Indica mediante el uso de bits cuál *router* opera como maestro (bit en 1), y cuál como esclavo (bit en 0).
- Número de secuencia DD **(*DD sequence number*):** se utiliza para darle secuencia a los paquetes de la base de datos.
- Cabecera LSA **(*LSA Header*):** En este campo se incluyen los LSA que describen la base de datos del *router* que generó los mensajes.

3.2.3.3 Paquete Link State Request

Este es un paquete tipo 3, generado por los *routers* después de que han actualizado su información de enrutamiento, si la información no se ha recibido completamente, el *router* genera este mensaje para solicitar a sus vecinos que le reenvíen la información faltante.

La estructura del mensaje se muestra en la tabla 3.8.

0 8 16 31
OSPF Header
LS Type
Link State ID
Advertising Router
…

Tabla 3.8: Cabecera del mensaje LSR.

De donde se tiene que cada LSA solicitada se específica por el tipo de estado del enlace, así como del ID del estado del enlace y del *router* que genera el mensaje. De forma más detallada, cada campo se denota a continuación.

- Tipo LS **(*LS Type*):** Contiene el tipo de notificación LSA solicitada.
- ID del estado del enlace **(*Link State ID*):** Este valor depende el tipo de LSA
- *Router* de notificación **(*Advertising Router*):** Contiene la ID del *router* que generó la petición del mensaje.

3.2.3.4 Paquete Link State Update

Estos son mensajes OSPF tipo 4, se encargan de realizar la inundación de paquetes LSA que contienen información de enrutamiento, métrica y topología de la red. También se encargan de responder a un mensaje LSR. El formato del mensaje se muestra en la figura 3.9.

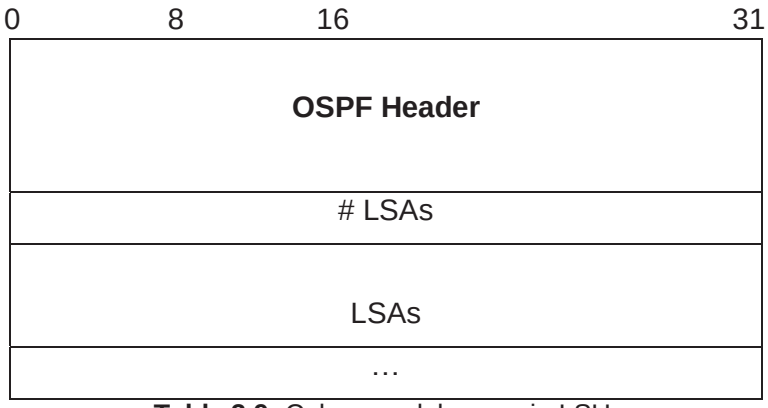

Tabla 3.9: Cabecera del mensaje LSU.

Los campos contenidos son:

- **# LSAs**: Contiene la cantidad de LSAs existentes dentro del mensaje LSU
- **LSAs:** Este campo contiene los LSAs completos.

3.2.3.5 Paquete *Link State Acknowledgment*

Estos son paquetes OSPF tipo 5, se encargan de notificar la recepción correcta de cada LSA. El formato del mensaje se muestra en la tabla 3.10.

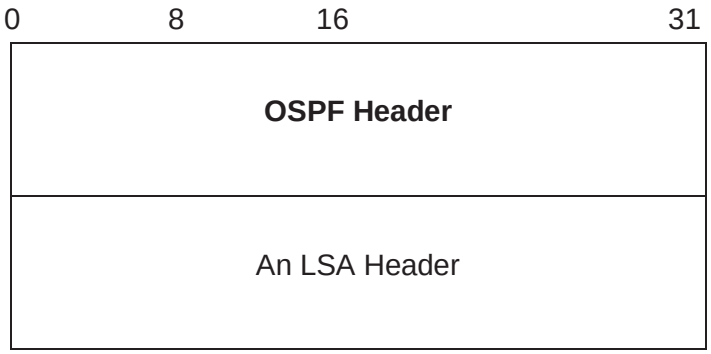

Tabla 3.10: Cabecera del mensaje LSAck.

El mensaje LSAck contiene todos los LSA que ya han sido aceptados por el *router*.

3.2.4 Notificación del estado del enlace (LSA)

En principio, se definen cinco tipos de notificaciones del estado del enlace, acorde con el RFC 2328 sección A.4 (*LSA formats*). La distribución de estas notificaciones se almacenan en una base de datos conocida como LSDB (*Link State Database* – Base de datos del estado del enlace) con la cual, los *routers* OSPF construyen un árbol de rutas hacia los diferentes destinos, viéndose a sí mismos como la raíz de dicho árbol.

El encabezado de los mensajes LSA tiene un tamaño de 20 b*ytes* y contiene la información necesaria para identificar el tipo de notificación que se está recibiendo, debido a que pueden llegar varias notificaciones al mismo tiempo. La cabecera de estas notificaciones se muestra en la tabla 3.11. [80, 81]

0	16	31
LS age	Options	LS Type
Link State ID		
Advertising Router		
LS Sequence Number		
LS Checksum	Length	

Tabla 3.11: Encabezado de las LSA.

La función de cada uno de los campos se especifica a continuación.

- **LS *age:*** Contiene el tiempo en segundos desde que el mensaje fue construido.
- ***Options:*** Este campo es el mismo que se especificó en la sección 3.2.3.1, para el mensaje de saludo (*Hello*).
- **LS *Type*:** Contiene el tipo de LSA que se está difundiendo, estos LSA son:
  1. *Router LSA*
  2. *Network LSA*
  3. *Summary LSA (IP Network)*
  4. *Summary LSA (ASBR)*
  5. *AS external LSA*

- ***Link State ID*:** Indica la sección de la red IP que está siendo descrita mediante los LSA recibida.
- ***Advertising Router*:** Este campo contiene el identificador del *router* que generó la notificación.
- ***LS Sequence Number*:** Este campo está dedicado a detectar las LSA antiguas o que han sido repetidas.
- ***LS Checksum*:** Es la suma de comprobación para garantizar que los datos sean correctos.
- ***Lenght*:** Contiene el tamaño de las LSA en *bytes.*

La función de cada una de las LSA utilizadas por OSPF se describen a continuación.

3.2.4.1 *Router* LSA

Estas notificaciones son consideradas notificaciones de tipo 1, se generan dentro de un área para notificar a los vecinos el estado de las interfaces (costo y estado) que utiliza el *router* que generó el mensaje. El formato del mensaje se muestra en la tabla 3.12.

<table>
<tr><td colspan="6">0 8 16 31</td></tr>
<tr><td colspan="6">LSA Header</td></tr>
<tr><td>0</td><td>V</td><td>E</td><td>B</td><td>0</td><td># Link</td></tr>
<tr><td colspan="6">Link ID</td></tr>
<tr><td colspan="6">Link Data</td></tr>
<tr><td colspan="2">Type</td><td colspan="2"># ToS</td><td colspan="2">Metric</td></tr>
<tr><td colspan="6">…</td></tr>
</table>

Tabla 3.12: Formato del mensaje tipo 1.

Cada campo se define a continuación:

- ***Bit* V:** Cuando se activa, indica que el *router* es un punto final de uno o más enlaces virtuales que mantienen adyacencia.
- ***Bit* E:** Cuando está activo, indica que el *router* funciona como un *router* de frontera del AS.

- ***Bit* B:** Cuando este *bit* está activo, indica que se tiene un ABR.
- **# *Link*:** Contiene el número de enlaces descritos en las notificaciones de tipo 1.
- ***Link ID*:** Identifica el tipo de enlace utilizado, los tipos de enlace pueden ser: ID del *router* vecino, IP del DR, dirección IP/máscara de red, ID del *router* vecino.
- ***Link Data*:** Este valor depende del campo tipo del enlace.
- ***Type*:** Contiene una breve descripción del enlace del *router,* el cual puede ser: conexión punto a punto, conexión a una red de tránsito, conexión a una red troncal o enlace virtual.
- **# *ToS* (*Type of Service* – Tipo de servicio):** Contiene el número de métricas que cada enlace utiliza.
- ***Metric*:** es el costo existente al *router* que generó el mensaje.

3.2.4.2 *Network* LSA

Este tipo de notificaciones está catalogado como del tipo 2, realiza la descripción de todos los *routers* OSPF existentes en la red por parte del DR. El formato de este mensaje se muestra en la tabla 3.13.

0 8 16 31
LSA Header
Network Mask
Attached Router

Tabla 3.13: Formato del mensaje tipo 2.

Los campos del mensaje especifican lo siguiente:

- ***Network Mask*:** Contiene la máscara de subred
- ***Attached Router*:** Este campo contiene los ID de los *routers* que formaron adyacencia con el DR.

3.2.4.3 Summary LSA

Estas notificaciones son de tipo 3, y son generadas por los ABR para distribuir la información de un área hacia las demás áreas, logrando así una conectividad completa. El formato del mensaje se presenta en la tabla 3.14.

3.2.4.4 Summary ASBR LSA

Las notificaciones tipo 4, son utilizadas para dar a conocer la ubicación de los ASBR, estas notificaciones las generan los ABR y las distribuyen dentro de las áreas con las cuales mantienen comunicación.

El formato del mensaje para las notificaciones del tipo 3 y tipo 4 se incluyen en la tabla 3.14, esto debido a que ambos mensajes son similares.

0	8	16	31
LSA Header			
Network Mask			
0		Metric	
ToS		ToS metric	
…			

Tabla 3.14: Formato para los mensajes del tipo 3 y tipo 4.

Los campos de los mensajes tipo 3 y tipo 4 se refieren a:

- ***Network Mask*:** Para los mensajes tipo 3 se incluye la dirección de la máscara de red. Para los mensajes tipo 4 este campo no es relevante, por lo que puede contener un valor cero.
- ***Metric*:** Indica el costo de la ruta
- **Los campos ToS** se incluyen para mantener compatibilidad con versiones anteriores al protocolo OSPF
- **ToS metric:** Incluye información específica de la métrica del TOS.

3.2.4.5 AS *external* LSA

Estas notificaciones son de tipo 5, y son creadas por los ASBR para dar a conocer rutas externas al AS.

El formato del mensaje se muestra en la tabla 3.15

0	8	16	31
LSA Header			
Network Mask			
E	0	Metric	
Forwarding Address			
External Route Tag			
E	TOS	TOS Metric	
Forwarding Address			
External Route Tag			
…			

Tabla 3.15: Formato del mensaje tipo 5.

Los campos se refieren a:

- ***Network Mask:*** Incluye la máscara de red del *router* destino, al cual se le entregará la LSA.
- ***Bit* E:** Indica el tipo de métrica externa, la cual puede ser del tipo 2; esto indica que la métrica puede considerarse mayor que cualquier ruta del estado del enlace, o bien, puede ser de tipo 1, indicando una métrica de igual magnitud que la utilizada en el estado del enlace.
- ***Metric*:** Indica el costo de la ruta, la cual depende del *bit* E.
- ***Forwarding Address*:** Incluye la dirección IP a la cual se reenviará la información. Si este campo se establece con la IP 0.0.0.0 el mensaje se reenvía al *router* que generó el mensaje. [82]
- ***External Route Tag**:* Este campo es utilizado para indicar rutas externas, generalmente se utiliza para intercambiar información entre *routers* fronterizos.

3.2.5 Métrica OSPF

OSPF utiliza el costo como métrica para determinar la mejor ruta. El costo se calcula con la ecuación 1.

$$costo = \frac{referencia\ del\ ancho\ de\ banda}{ancho\ de\ banda\ de\ la\ interfaz} \quad \ldots Ec.1$$

Donde la referencia del ancho de banda está establecida en (10^8) 100 Mbps, por lo que sustituyendo el valor en la formula, esta queda de la siguiente forma.

$$Costo = \frac{10^8}{ancho\ de\ banda} \quad \ldots Ec.2$$

El costo de una interfaz es inversamente proporcional al ancho de banda, es decir, si el ancho de banda es mayor, el costo será menor, y si existe sobre carga y retraso en el enlace el costo será mayor. Los enlaces superiores a 100 Mbps se fijan en un costo de 1 al sustituir los valores en la fórmula 2. [83]

La tabla 3.16 muestra el costo de algunas de las interfaces utilizadas en las redes de datos. Los valores en color serán las utilizadas en la simulación y emulación del *backbone* CANARIE, por ser las interfaces de mayor capacidad que *Packet Tracer* y GNS3 pueden manejar.

Tipo de Interfaz	Ancho de banda de la interfaz	Costo
10 Gigabit Ethernet	10,000,000,000	1
1 Gigabit Ethernet	1,000,000,000	1
100 Mbps, Fast Ethernet	100,000,000	1
10 Mbps, Ethernet	10,000,000	10
1,544 Mbps, Serial	1,544,000	64
128 Kbps, Serial	128,000	781
64 Kbps	64,000	1562

Tabla 3.16: Costos por defecto OSPF.

3.2.6 *Wildcard* OSPF

La *Wildcard* es una secuencia de 32 bits que indican a un *router* que sección de una dirección de red debe coincidir para realizar el enrutamiento de los datos dentro del proceso OSPF. La *wildcard* está considerada como el inverso de la máscara de red que utiliza la lógica binaria para realizar filtraciones del tráfico IP, mediante la aplicación de las siguientes reglas.

1. Un bit en 0 en la *wildcard* indica que coincidirá el valor correspondiente en la dirección IP.
2. La existencia de un 1 lógico en la *wildcard* significa que se debe ignorar el valor correspondiente dentro de la IP.

La dirección *wildcard* se consigue realizando una resta lógica entre la máscara de red a utilizar y la IP 255.255.255.255. Como ejemplo se tiene para la simulación y emulación de conectividad del capítulo 4, las máscaras de red 255.255.0.0 y 255.255.255.0, por lo que las máscaras de *wildcard* a utilizar se presentan en las ecuaciones 3 y 4.

$$\begin{array}{r} 255.255.255.255 \\ -\ 255.255.\ \ 0.\ \ 0 \\ \hline 0.\ \ 0.255.255 \end{array} \quad \ldots \text{EC. 3}$$

$$\begin{array}{r} 255.255.255.255 \\ -\ 255.255.255.\ \ 0 \\ \hline 0.\ \ 0.\ \ 0.255 \end{array} \quad \ldots \text{EC. 4}$$

3.3 Gestión de la red

La gestión de la red consiste en la configuración, monitoreo, análisis y control de los recursos de la red, para cubrir las necesidades de una organización, cuya finalidad es brindar el mejor servicio a sus usuarios. De manera formal, la gestión de la red incluye el despliegue, integración y coordinación del hardware, software y elementos humanos para monitorear, probar, sondear, configurar, analizar, evaluar y controlar los recursos de la red, para conseguir los requerimientos en tiempo real, desempeño operacional y calidad de servicio, a precio razonable. [84]

Para monitorear y controlar las redes de datos, es necesario verlas como una arquitectura única, es por ello que se presenta el protocolo de gestión de red, el cual es una colección de herramientas que facilitan el monitoreo y control de dichas redes.

3.3.1 Protocolo de Gestión de Red – SNMP

SNMP (*Simple Network Management Protocol* – Protocolo de gestión de red), fue desarrollado por la IAB (*Internet Activities Board*), como un protocolo dedicado a llevar a cabo la gestión de diferentes tipos de redes. En conjunto con SNMP, se diseñaban el HEMS (*High-Level Management System* – Sistema de Gestión de Alto Nivel), y el SGMP (*Simple Gateway Monitoring Protocol* – Protocolo de Monitoreo de Gateway Simple), la decisión final fue permitir la evolución de SNMP, complementándolo con características que los otros dos protocolos utilizaban. SNMP es un conjunto de aplicaciones, con las cuales se realiza una gestión de calidad, hace uso de UDP como protocolo de comunicación, el cual permite intercambiar información de solicitud y respuesta entre los dispositivos de la red, utilizando los puertos 161 para realizar peticiones de información y recibir respuestas a dichas peticiones y, del puerto 162, para recibir notificaciones de posibles fallos en la red. Estos puertos son preestablecidos por defecto en el software de gestión. SNMP es un protocolo de la capa de aplicación del modelo ISO/OSI, creado para facilitar el intercambio de información de gestión entre los diversos dispositivos de la red, mediante una estandarización que establece las normas a través las cuales se consulta la información de gestión. [85]

SNMP cuenta con tres versiones; la primera se dio a conocer en 1988 dentro de los siguientes RFCs:

- RFC 1065: define la estructura e identificación de la información de gestión.
- RFC 1066: define la base de información de gestión (MIB-I).
- RFC 1067: define el protocolo de administración de la red. [86, 87, 88]

Cada uno de los RFCs que dieron origen a SNMP, fueron actualizados en 1990, en los RFC 1155 al RFC 1157. La MIB-I se actualizó en 1991, pasando a ser conocida como MIB-II, en el RFC 1213 [89]. SNMP ganó popularidad en la década de 1990, al permitir el monitoreo de dispositivos de forma remota, utilizando un sencillo método de control de acceso a la información, a través una cadena de caracteres denominada “comunidad”, un tipo de contraseña en texto plano. Esta comunidad define dos formas de consultar la información: la primera es una “comunidad pública”, utilizada solamente para consultar la información contenida en la variable de gestión, la segunda, es una “comunidad privada”, mediante la cual se puede leer y modificar algún valor de la variable de gestión. El sistema de seguridad de la primera versión es fácil de obtener mediante el uso de un *sniffer,* debido a su sencillez y su falta de encriptación.

La segunda versión (SNMPv2) surgió en 1996, para mejorar las deficiencias de la primera versión, principalmente en temas de seguridad y formas de obtener la información de gestión, al incluir nuevas operaciones como *Get-Bulk-Request* para recuperar grandes bloques de información como información de una tabla. También se incluyeron las operaciones: *inform* para enviar de forma espontánea información a un NMS (*Network Management System* – Sistema de Gestión de Red) notificando algún error, el NMS se ve obligado a generar una confirmación del recibimiento del mensaje; y *report* para notificar de forma espontánea excepciones y errores del protocolo. SNMPv2 permitió una mejor flexibilidad en los mecanismos de control de acceso, permitiendo asociar un nombre de comunidad con un perfil de dicha comunidad. Posteriormente apareció SNMPv2c, definida en los RFC 1901 al RFC 1908. Esta nueva versión retomó la cadena de caracteres basada en comunidades, utilizada en la primera versión como método de autentificación. Posteriormente

surgió SNMPv2u, incluyendo un método de seguridad neutro entre SNMPv1 y SNMPv2, la cual se define en los RFC 1909 y RFC 1910. Una variante de esta última versión se dio a conocer como SNMPv2*. Los mecanismos de seguridad de SNMPv2u y SNMPv2* dieron origen a SNMPv3.

SNMPv3 trabaja esencialmente en el sistema de seguridad, incluyendo autentificación, privacidad y control de acceso, así como la administración del protocolo para realizar configuraciones remotas. SNMPv3 utiliza el modelo USM (*User-Based Security Model* – Modelo de Seguridad Basado en Usuario), el mecanismo genera un acuse de recibo del mensaje, y garantiza que este no fue modificado durante su trayecto, para ello, utiliza el protocolo de autentificación HMAC-MD5-96 o el HMAC-SHA-96 (*Message Authentication Code* – Código de Autentificación de Mensaje; la función MAC realiza un mapeo aleatorio para garantizar la autenticidad de un mensaje, de igual forma lo hace la función HASH, por lo que se realiza una mezcla de ambas funciones para tener un sistema de autentificación más sólido), la cual utiliza una función hash criptográfica en conjunto con una llave de encriptación como MD5 o SHA-1 (*Secure Hash Algorithm* – Algoritmo de Hash Seguro). Además, SNMPv3 difunde los mensajes dentro de un intervalo de tiempo en el que agente y gestor garantizan la veracidad del mensaje, si después de transcurrido dicho intervalo de tiempo se recibe un mensaje, este se considera erróneo y se desecha, además del intervalo de tiempo, cada mensaje lleva una codificación extra utilizando el algoritmo CBC (*Cipher Block Chaining,* construido por *Data Encryptation Standard,* conocido como DES-56). SNMPv3 se dio a conocer en 2004, en los RFC 3411 al RFC 3418.

A pesar de que la versión 3 del protocolo SNMP cuenta con mejoras en cuestión de seguridad, se continúa utilizando las dos primeras versiones, debido a su simplicidad y fácil manejo.

3.3.2 Arquitectura de SNMP

Dentro de una red de datos gestionada con el protocolo SNMP, es garantía encontrar los siguientes elementos:

- Estación de Gestión (*Manager*)
- Agente de Gestión (*Agent*)
- Base de Información de Gestión (MIB)
- Protocolo de Gestión de Red

A continuación se da una breve explicación del funcionamiento de cada uno de ellos.

3.3.2.1 Estación de Gestión

Es el elemento central de la red, en donde se lleva a cabo la administración de cada uno de los dispositivos gestionados. Se le conoce como NMS (*Network Manager System* – Sistema de Gestión de Red), ejecuta programas denominados NMA (*Network Management Application* – Aplicación de Gestión de Red), como es el caso del gestor, *software* encargado de crear una interfaz gráfica, la cual permite a la persona encargada de administrar la red, tener un mapa completo de los agentes que conforman dicha red. El gestor mantiene comunicación directa con los dispositivos gestionados, recuperando información de cada una de las variables contenidas en los agentes y modificando dicho valor, si los permisos de acceso (comunidad) así lo permiten.

3.3.2.2 Agente de gestión

Dentro de los dispositivos gestionados se ejecuta el *software* conocido como agente, el cual se encarga de responder a cada una de las peticiones realizadas por el gestor. Esencialmente, el agente realiza las siguientes funciones: proporciona la información que se le solicitó desde el NMS, controla el acceso a la información de los objetos gestionados a través de los permisos configurados en cada objeto, crea, modifica y elimina objetos de gestión, así como también, envía notificaciones de forma espontanea al NMS, indicando sucesos ocurridos en los dispositivos administrados. Además del agente, los dispositivos gestionados también cuentan con una MIB (*Management Information Base* – Base de Información de Gestión), donde se almacena toda la información que permite la gestión de los dispositivos. [90]

El agente y el gestor mantienen comunicación a través de diversos mensajes, mediante los cuales se obtiene toda la información necesaria para realizar la gestión. El formato del mensaje SNMP está constituido por dos partes: la primera consta del encabezado del mensaje, el cual contiene la versión SNMP y el nombre de la comunidad utilizada como sistema de seguridad. La segunda parte consta del PDU (*Protocol Data Unit* – Unidad de Datos del Protocolo), que contiene la operación específica para la obtención de los valores de las variables de gestión. La tabla 3.17 muestra el formato PDU de la trama SNMP. [91]

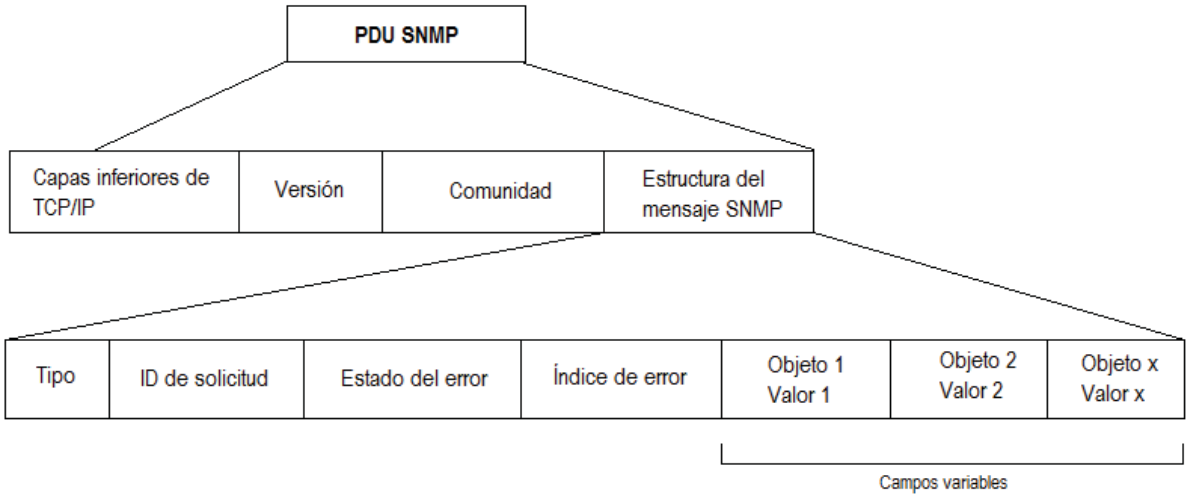

Tabla 3.17: Formato de la trama SNMP.

A continuación se realiza la descripción de cada campo que conforma la estructura del mensaje SNMP.

- **Tipo:** Específica el tipo de operación que se está transmitiendo, los cuales pueden ser:
  - ***Get-Request*:** Este mensaje lo envía el NMS al agente para solicitar un valor específico de alguna variable.
  - ***Get-Next-Request*:** Es un mensaje que el NMS genera para solicitarle un valor sobre alguna tabla al agente, tal es el caso de la recuperación de la información de una tabla de enrutamiento IP.
  - ***Set-Request*:** Mediante este mensaje, el NMS solicita al agente que realice la modificación del valor existente en una variable.

- ***Get-Response:*** Es la respuesta generada por el agente a cada una de las peticiones realizadas por el NMS. En cada respuesta se incluye el valor de la variable solicitada o bien, se indica que se ha realizado modificación a alguna variable.
- ***Get-Bulk-Request:*** Este comando fue introducido en la segunda versión de SNMP, es enviado por el NMS para recuperar toda la información de una tabla con un sólo mensaje.
- ***Trap:*** Este mensaje es generado por el agente de forma espontanea, notifica de algún evento ocurrido en el dispositivo gestionado. El PDU del mensaje *trap* para SNMPv1 se presenta en la tabla 3.18.

Empresa	IP del agente	Tipo de *trap* generico	Código de *trap* específico	Marca de tiempo	Objeto 1 Valor 1	Objeto 2 Valor 2	Objeto x valor x
					Campos variables		

Tabla 2.18: PDU del mensaje *trap* para SNMPv1.

Cada campo que conforma el PDU del mensaje *trap* se describe a continuación:

- **Empresa:** Identifica el objeto que ha disparado el *trap.*
- **IP del agente:** Proporciona la dirección IP del agente que ha generado el *trap.*
- **Tipo de *trap* genérico:** Indica el tipo de *trap* que se ha generado, puede ser cualquiera de la siguiente lista:
  - ***coldStart* (0):** Indica que el agente ha sufrido un reinicio inesperado del sistema.
  - ***warmStart* (1):** Notifica que se ha cambiado la configuración del dispositivo gestionado.
  - ***linkDown* (2):** Notifica que un enlace quedó fuera de servicio (inactiva). El atributo/valor de la variable aparece como *ifIndex* para la interfaz.

- ***linkUp* (3):** Indica que un enlace ha sido puesto en operación (activo). El atributo/valor de la variable aparece como *ifIndex* para la interfaz.
- ***authenticationFailure* (4):** Notifica la acción no permitida sobre algún dispositivo gestionado, al cual se le ha realizado un requerimiento a través de un NMS que no cuenta con los permisos de comunidad.
- **egpN*eighborLoss* (5):** Este mensaje aparece cuando la topología utiliza EGP como protocolo de enrutamiento. Notifica que un *router* vecino ha dejado de recibir mensajes de saludo, perdiendo la relación de vecino.
- ***enterpriseSpecific (6):*** Este campo contiene más información referente al *trap* específico que se ha generado.

El formato del *trap* para SNMPv2c desaparece los campos *enterprise, agent-addr, generic-trap, specific-trap* y *timestamp.* Algunos de los valores que no figuran más como parte del mensaje *trap* para esta versión, suelen aparecer dentro de los campos variables, como parte de la información de la variable a gestionar. Los campos “estado del error” y “error index”, son puestos a cero. El formato del *trap* para SNMPv2 se indica en la tabla 3.19.

Tipo	ID-Solicitud	0	0	Objeto 1 Valor 1	Objeto 2 Valor 2	Objeto x Valor x
				Campos variables		

Tabla 3.19: PDU del mensaje *trap* para SNMPv2c.

Donde:

- **ID-solicitud**: Indica la secuencia asignada a la solicitud de una variable.

- **Código de *trap* específico:** contiene *traps* específicos del fabricante.

- **Marca de tiempo:** Monitorea el tiempo transcurrido desde el último reinicio del sistema hasta la generación de un *trap.*
- **Campos variables:** Indican la causa por la cual se generó el *trap.*

Continuando con la descripción de los campos de la figura 3.2, se tiene:

- **ID de solicitud:** Se utiliza para distinguir los mensajes intercambiados entre la NMS y el agente. Cada solicitud lleva un ID único.
- **Estado del error:** indica los tipos de errores ocurridos ante la petición de información. Estos pueden ser:
  - **0:** No hay errores
  - **1:** Demasiado grande
  - **2:** la variable no existe
  - **3:** el valor es incorrecto
  - **4:** el valor sólo contiene permisos de lectura
  - **5:** Error genérico.
- **Índice de error:** Proporciona información adicional acerca de un error causado por una variable, siempre y cuando el campo estado del error contenga un valor distinto a cero.
- **Campos variables:** contiene una lista de nombres de variables con sus respectivos valores, los cuales han sido solicitados por un *get,* o enviados mediante un *trap.*

3.3.2.3 Estructura de la información de gestión – SMI

La SMI (*Structure of Management Information* – Estructura de la Información de Gestión), se define en el RFC 1155 [92]. El estándar especifica la forma en cómo se debe agrupar y nombrar la información de gestión. Cada objeto debe tener un nombre único, una sintaxis, mediante la cual se escribe la MIB y una codificación, también define los datos que serán permitidos. En otras palabras, la SMI define la arquitectura con la cual se escribe la MIB. La arquitectura SMI contiene los siguientes elementos:

1. **Identificadores de objetos:** cada variable en SNMP tiene un identificador único, ya sea mediante un nombre (iso.org.dod.internet.mgm.mib-2) o en forma numérica (1.3.6.1.2.1). Los identificadores se establecen en orden jerárquico.
2. **Tipos de objetos:** SMI define los tipos de objetos a utilizar como enteros, nulo, cadena de octetos e identificador de objetos. Para ello, utiliza la notación ASN.1 (*Abstract Syntax Notation One* – Notación Sintáctica Abstracta 1. Lenguaje estandarizado por CCITT y la ISO), para definir la sintaxis sobre datos de aplicaciones, los cuales una vez aprobados se convierten en estándar.
3. **Método de codificación de objetos:** Utiliza una cadena de octetos bajo el método BER (*Basic Encoding Rules* – Regla de Codificación Básica), que incluye tipo, etiqueta, longitud y valor, para codificar a cada objeto junto con sus valores, para transmitirlos dentro de los paquetes SNMP.

SMI le da forma a la base de datos que almacena a los objetos de gestión, la cual se conoce como MIB. La información en la MIB es organizada en una estructura de nombres en forma de árbol, brindando un orden jerárquico a cada elemento. El árbol cuenta con una raíz, de la cual se desprende toda la información en líneas que representan ramas, a las cuales se les denomina nodos, contienen al final de cada una de ellas una etiqueta con un valor único. Estas etiquetas pueden dar origen a otros nodos para formar un subárbol. Del nodo raíz se desprenden tres ramas: una de ellas está administrada por la Organización Internacional de Estandarización (OSI), cuya etiqueta es legible como iso (1); la segunda rama la administra el Comité Consultivo Internacional Telegráfico y Telefónico, encontrado en el árbol como ccitt (0); la última rama es la unión de las dos anteriores, cuya etiqueta se lee joint-iso-ccitt (2), tal como se muestra en la figura 3.2. La administración de los elementos de red es accesible siguiendo la ruta que conecta con org (3), asignada a otras organizaciones. El departamento de defensa de los Estados Unidos, aunque no cuenta con elementos gestionables, cuenta con un espacio en el nodo dod (6). El siguiente nodo permite la administración de Internet (1), del cual se desprenden seis ramas, estas son:

- **Directorio (1):** Específica la forma en cómo deben utilizarse las direcciones OSI en la Internet.
- **Mgmt (2):** Identifica los objetos definidos por la IANA (*Internet Assigned Numbers Authority* – Autoridad de Asignación de Números de Internet)
- **Experimental (3):** Son los nuevos elementos bajo análisis de funcionamiento, los cuales una vez aceptados, se estandarizan y se establecen en la etiqueta de mgmt (2).
- ***Private* (4):** Los proveedores de equipos de red cuentan con este espacio para establecer sus sistemas gestionables.
- **Seguridad (5):** Se incluyen nuevos sistemas de seguridad.
- **SNMPv2 (6):** se utiliza para tareas realizadas con la segunda versión de SNMP.

3.3.2.4 Base de Datos de Información de Gestión – MIB

La MIB es una base de datos que contiene la información organizada en una estructura jerárquica, la cual se rige bajo el estándar SMI. Define las variables que utiliza el protocolo de gestión para supervisar y controlar los dispositivos de la red [93]. La MIB identifica a cada objeto mediante un OID (*Object Identifier* – Identificador de Objeto), a través del cual se proporciona acceso a toda la información de gestión. Las variables son almacenadas en grupos y suelen pertenecer a una rama determinada. Los identificadores del árbol de gestión, están constituidos por diferentes organizaciones dedicadas a la construcción de estándares, como lo es la ISO, sin embargo, organizaciones asociadas y algunas empresas han incluido sus propias variables de gestión. La consulta al árbol de gestión se realiza desde la parte superior, conocida como raíz, de la cual se desprenden tres ramas, descritas anteriormente en la sección de la SMI (3.3.2.3). A las variables de interés se ingresa siguiendo la ruta desde la raíz dada por el OID 1.3.6.1, el cual está formado por seis subárboles (directorio (1), mgmt (2), experimental (3), *private* (4), seguridad (5) y snmpv2 (6)), cada uno de estos subárboles contienen toda la información referente a la gestión de los dispositivos de la red. Actualmente se utiliza la segunda versión de

la MIB (MIB-II), definida en el RFC 1213, donde la información de gestión es dividida en diferentes categorías, estas son:

- ***System* (1):** Proporciona información de contacto administrativo, ubicación y servicio del dispositivo gestionado. Su implementación es obligatoria para todos los sistemas, en caso de que algún agente no tenga dicha configuración, se asigna un valor cero.
- ***Interfaces* (2):** Proporciona información referente a las interfaces del dispositivo gestionado.
- ***Address-translation* (3):** Proporciona información sobre la traducción de direcciones físicas a lógicas y viceversa.
- **IP (4):** Proporciona información referente al protocolo IP.
- **ICMP (5):** Proporciona toda la información referente al protocolo ICMP.
- **TCP (6):** Facilita la información referente a conexiones TCP.
- **UDP (7):** contiene la información de configuración y estadísticas del protocolo UDP.
- **EGP (8):** Contiene toda la información referente al protocolo EGP.
- **CMOT (9):** Este nodo está definido en el RFC 1095 [94], describe una arquitectura de red utilizando los protocolos CMIS/CMIP del modelo OSI. Proporciona un canal para que el gestor y el agente intercambien información de forma remota.
- ***Transmission* (10):** Este nodo está reservado para información específica de pruebas para nuevos modelos de gestión.
- **SNMP (11):** Contiene información de la red de datos que soportan el uso de SNMP. [95]

Como caso práctico, en la figura 3.2, se muestra un ejemplo para identificar el OID de un enlace deshabilitado. Siguiendo la ruta desde la raíz hasta el punto deseado se obtiene el OID .1.3.6.1.6.3.1.1.5.3. El identificador de objeto (OID) obtenido, se intentará replicar en el capítulo 5, durante las pruebas de conectividad y gestión del *backbone* CANARIE, para contrastar la teoría con la práctica.

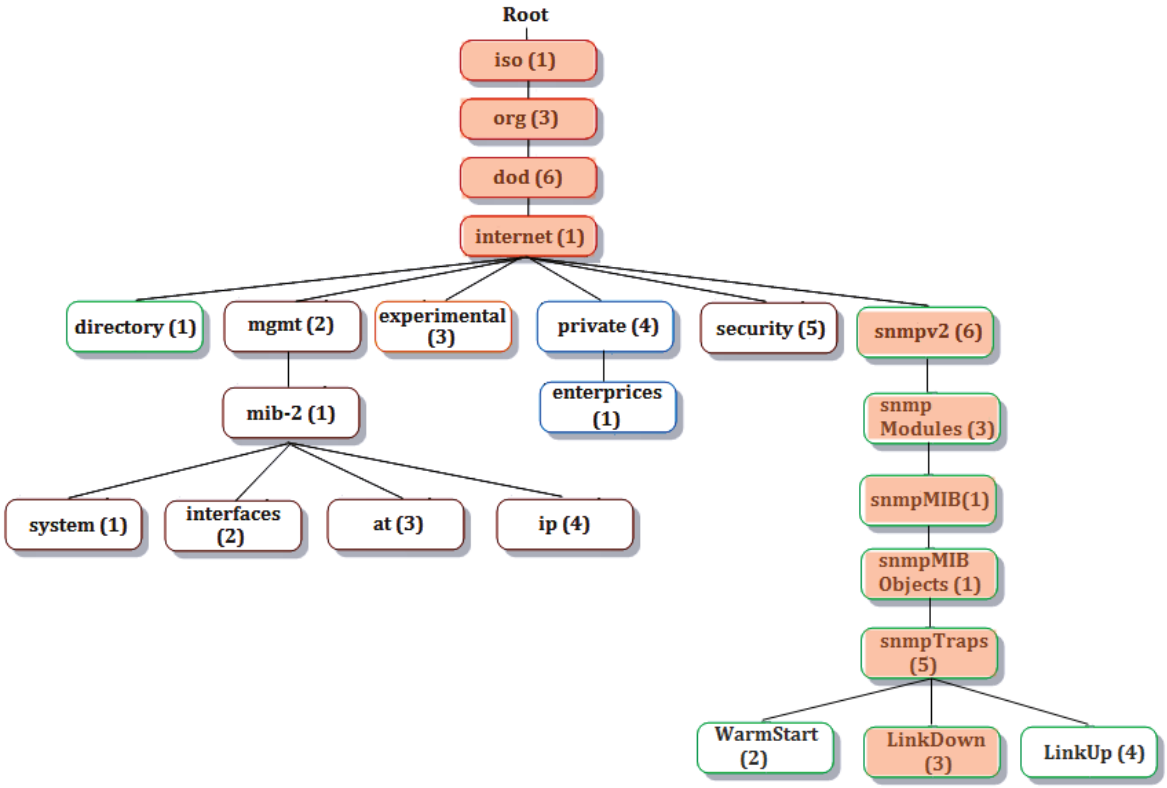

Figura 3.2: Parte del árbol de Internet utilizando OIDs. Figura propia con base en referencia [96]

Capítulo 4: Metodología para la simulación y emulación del *backbone* CANARIE

4.1 Introducción

Una red de datos es funcional únicamente cuando se le ha configurado un protocolo de enrutamiento a cada uno de los *routers* que la conforman. De esta forma se garantiza que exista comunicación entre los distintos equipos de dicha red, con la finalidad de que establezcan relación de vecinos, para construir sus tablas de enrutamiento y, con ello, poder realizar la transmisión de paquetes de un punto a otro. Partiendo de esta idea, se pretende utilizar las herramientas de análisis de redes *Packet Tracer* y GNS3, para realizar la simulación y emulación del diseño de la figura 4.1, la cual pertenece a la topología del *backbone* CANARIE, en su versión actualizada al 2016. Cada uno de los *routers* del diseño del *backbone* CANARIE será configurado con el protocolo de enrutamiento OSPF y, con el protocolo de gestión SNMP.

Packet Tracer es un simulador gráfico de redes, desarrollado por Cisco como herramienta de entrenamiento para sus estudiantes. El simulador provee un entorno gráfico para crear topologías de red y configurar los dispositivos utilizados. Además, ofrece realizar el análisis de los procesos ejecutados en el programa, convirtiendo así a la computadora donde se ha instalado, en un laboratorio de redes de datos [97]. Se entiende por simulador a la representación de un sistema técnico, que imita un proceso real, obedeciendo a un modelo interno, el cual incluye el mayor número de variables encargadas de modular el comportamiento de un sistema real [98].

GNS3 (*Graphical Networks Simulator* – Simulador de Red Grafica) es un software emulador de IOS (*Internetwork Operating System* – Sistema Operativo de Interconexión) de *Routers* Cisco, ASA Firewalls y Juniper. Está basado en Dynamips (emulador de *routers* Cisco), Dynagen (es un *front-end* para Dynamips) y PEMU (emulador de Cisco PIX firewall basado en Qemu), permite diseñar topologías de red para emularlas lo más cercano a la realidad, utilizando IOS de *routers* reales [99]. Un emulador es un programa que crea una plataforma virtual, sobre la cual se ejecutan programas diseñados para otras plataformas distintas a las que se ejecutan en la computadora.

Figura 4.1: Diseño del *backbone* CANARIE para realizar las pruebas de conectividad y gestión, mediante simulación y emulación. Actualizado al 2016. Diagrama propio con base en referencia [100]

Cada uno de los enlaces que conectan a los 25 *routers* que conforman la topología de la figura 4.1 deben utilizar una dirección de red, mediante la cual se puede establecer la comunicación entre dichos *routers.* En la tabla 4.1 se presentan las direcciones de red utilizadas en cada segmento que une a los *routers*. Las direcciones de red utilizadas están dentro del rango 172.1.0.0/16 al 172.28.0.0/16.

Enlaces	Dirección de red
Victoria – Vancouver	172.1.0.0/16
Vancouver – Kamloops	172.2.0.0/16
Vancouver – Edmonton	172.3.0.0/16
Edmonton – Saskatoon	172.8.0.0/16
Edmonton – Fort Nelson	172.4.0.0/16
Fort Nelson – Whitehorse	172.5.0.0/16
Fort Nelson – Fort Simpson	172.6.0.0/16
Fort Simpson – Yellowknife	172.7.0.0/16
Saskatoon – Winnipeg	172.12.0.0/16
Kamloops – Calgary	172.9.0.0/16
Calgary – Regina	172.10.0.0/16
Regina – Winnipeg	172.11.0.0/16
Winnipeg – Toronto	172.14.0.0/16
Winnipeg – Thunder Bay	172.13.0.0/16
Thunder Bay – Toronto	172.15.0.0/16
Toronto – London	172.18.0.0/16
London – Chathaman	172.17.0.0/16
Chathaman – Windsor	172.16.0.0/16
Toronto – Ottawa	172.19.0.0/16
Ottawa – Montreal	172.20.0.0/16
Montreal – Federation	172.22.0.0/16
Montreal – Quebec	172.21.0.0/16
Quebec – Rimouski	172.23.0.0/16
Rimouski – Moncton	172.24.0.0/16
Federation – Halifax	172.27.0.0/16
Moncton – Halifax	172.25.0.0/16 172.26.0.0/16
Halifax – ST. John’s	172.28.0.0/16

Tabla 4.1: Direcciones de red utilizadas en cada enlace que conforma el *backbone* CANARIE.

4.2 Especificaciones técnicas para la simulación de conectividad y de gestión

Para poder llevar a cabo la simulación de conectividad y de gestión, se utilizó el simulador *Packet Tracer* versión 7.0.0.0305, con licencia liberada en 2016. El simulador fue instalado en una computadora con las siguientes características:

- S. O. Windows 7 Professional (32 bits), *Service Pack* 1
- Procesador Intel (R) Pentium (R) Dual CPU E2140 @ 1.60 GHz. Cuenta con 2 núcleos y 2 procesadores lógicos [101].
- Memoria RAM de 3 GB.

En *Packet Tracer* se utilizaron:

- 25 *routers* de la serie 2811, ya que el simulador no cuenta con *routers* tipo *core*.
- 5 *switches* 2950 de 24 puertos, los cuales funcionan en capa 2 del modelo ISO/OSI.
- Así como 5 *host* para realizar las pruebas de conectividad.

Para poder distinguir la procedencia de las rutas, al realizar la consulta de alguna de las tablas de enrutamiento, a cada uno de los 25 *routers* se les asignó un identificador en formato de dirección IPv4. La tabla 4.2 contiene el identificador para cada *router,* además, se especifica el área en la cual se encuentran conectadas sus interfaces y, la prioridad de cada una de ellas, para hacer distinción entre un DR y un BDR. Es decisión personal como administrador de la red, establecer al *router* Winnipeg como el *router* designado, cuya prioridad se establece en 255, mientras que el *router* Ottawa se asigna como el *router* designado de respaldo, con prioridad 253. El resto de los *routers* de la topología se mantendrán por defecto con prioridad 0. Aunque el hecho de asignar un DR y un BDR no tiene impacto en conexiones punto a punto, funciona para reafirmar el concepto teórico y ver que es posible llevar a cabo dicha acción durante la configuración de la red.

Router	Identificador	Área	Prioridad
Victoria	1.1.1.1	1	0
Vancouver	2.2.2.2	0	0
Kamloops	3.3.3.3	0	0
Edmonton	4.4.4.4	0	0
Fort Nelson	5.5.5.5	0	0
Whitehorse	6.6.6.6	2	0
Fort Simpson	7.7.7.7	0	0
Yellowknife	8.8.8.8	3	0
Calgary	9.9.9.9	0	0
Regina	10.10.10.10	0	0
Saskatoon	11.11.11.11	0	0
Winnipeg	12.12.12.12	0	255
Thunder Bay	13.13.13.13	0	0
Toronto	14.14.14.14	0	0
London	15.15.15.15	0	0
Chathaman	16.16.16.16	0	0
Windsor	17.17.17.17	4	0
Ottawa	18.18.18.18	0	253
Montreal	19.19.19.19	0	0
Quebec	20.20.20.20	0	0
Rimouski	21.21.21.21	0	0
Federation	22.22.22.22	0	0
Moncton	23.23.23.23	0	0
Halifax	24.24.24.24	0	0
ST. John's	25.25.25.25	5	0

Tabla 4.2: Asignación de áreas, número de identificación y prioridad para cada *router.*

El enlace entre los *routers* utilizados en *Packet Tracer* se establece mediante conexión de interfaz serial DTE (*Data Terminal Equipment* – Equipo Terminal de Datos), a través de la cual, se reciben servicios tales como sincronización en la trasferencia de datos, generados por un DCE (*Data Communications Equipment* – Equipo de Comunicación de Datos) ubicado al otro extremo de la conexión. Las interfaces seriales utilizan una señal de sincronización para controlar la comunicación

y contienen un ancho de banda predeterminado de 1,544 Mbps. Un cable serial DTE es utilizado para conectar redes WAN (*Wide Area Network* – Red de Área Amplia).

Los *host* utilizados para realizar la simulación de conectividad, son conectados a los *routers* existentes en los extremos del *backbone* (Victoria, Whitehorse, Yellowknife, Windsor y ST. John's) con base en figura 4.1. La interfaz utilizada para conectar a los *host* con los *routers* es de FastEthernet, configurada a 100 Mbps. En la figura 4.2 se muestra la configuración de la interfaz *FastEthernet 0/0* del *router* Victoria, estableciendo de forma fija los 100 Mbps. El mismo proceso se realizó para cada uno de los cuatro *host* restantes. Se estableció el rango de direcciones IP 192.168.1.0/24 – 192.168.5.0/24 con máscara de red natural para utilizarlas en cada uno de los *host.*

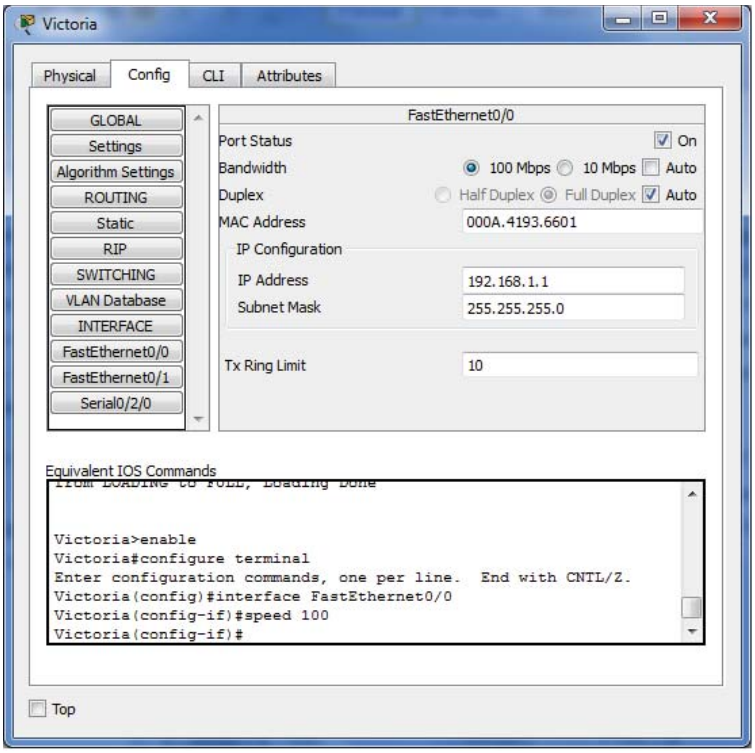

Figura 4.2: Configuración de la interfaz *FastEthernet* para operar a su máxima capacidad.

4.2.1 Simulación de conectividad

El proceso para la simulación de conectividad que será registrado en el desarrollo de esta tesis, se establece solamente para el caso que comprende la prueba de conectividad entre los *routers* Victoria y ST. John's. Se han elegido estos dos puntos de acceso por ser los que mayor cantidad de *routers* contiene, considerando la ruta más larga, sin tomar en cuenta los criterios que utiliza OSPF para conseguir el mejor camino que conecte a dichos puntos. Sin embargo, en los resultados para la simulación de conectividad se dará a conocer si fue o no posible realizar la simulación de conectividad del resto de los *host* propuestos en la figura 4.1.

A cada uno de los 25 *routers* que conforman el diseño de la figura 4.1 se les configuró las interfaces que mantienen conexión con sus vecinos, asignándoles una dirección de red, de acuerdo a las consideraciones de la tabla 4.1, así como también se estableció la velocidad de reloj (*clock rate*) en 64,000 para mantener sincronizada la comunicación entre los *routers* que utilizan conexión serial, esto con la finalidad de que realicen el intercambio de información. Por último, se le indicó a cada interfaz que se mantengan activas a través del mando "*no shutdown"*. La figura 4.3 muestra la configuración de las interfaces para el *router* Vancouver. Cuando se establece la velocidad del reloj, el *router* arroja un mensaje notificando que el mando sólo se aplica a conexiones seriales DCE, esto se debe a que la interfaz configurada se encuentra como un DTE.

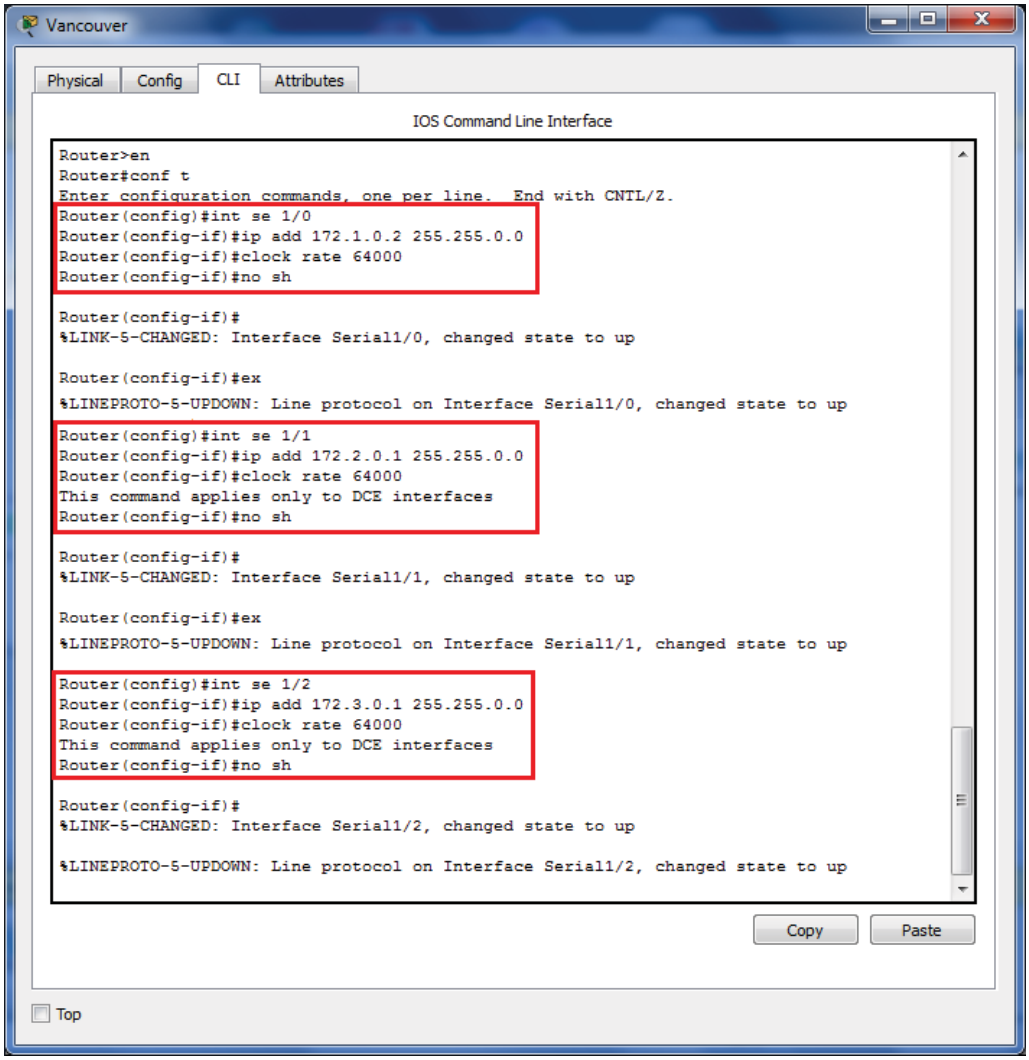

Figura 4.3: Configuración de las interfaces del *router* Vancouver.

Después de configuradas las interfaces con sus respectivas direcciones de red, el siguiente paso es activar el protocolo OSPF en el *router* correspondiente. Primero, se configura el identificador de proceso (*process-ID*) estableciéndolo en 1, después se asigna el identificador del *router* (*router*-id) con base en la tabla 4.2 y por último, se asigna el área correspondiente a cada una de las interfaces configuradas en el *router.*

La figura 4.4 muestra la activación de OSPF en el *router* Vancouver.

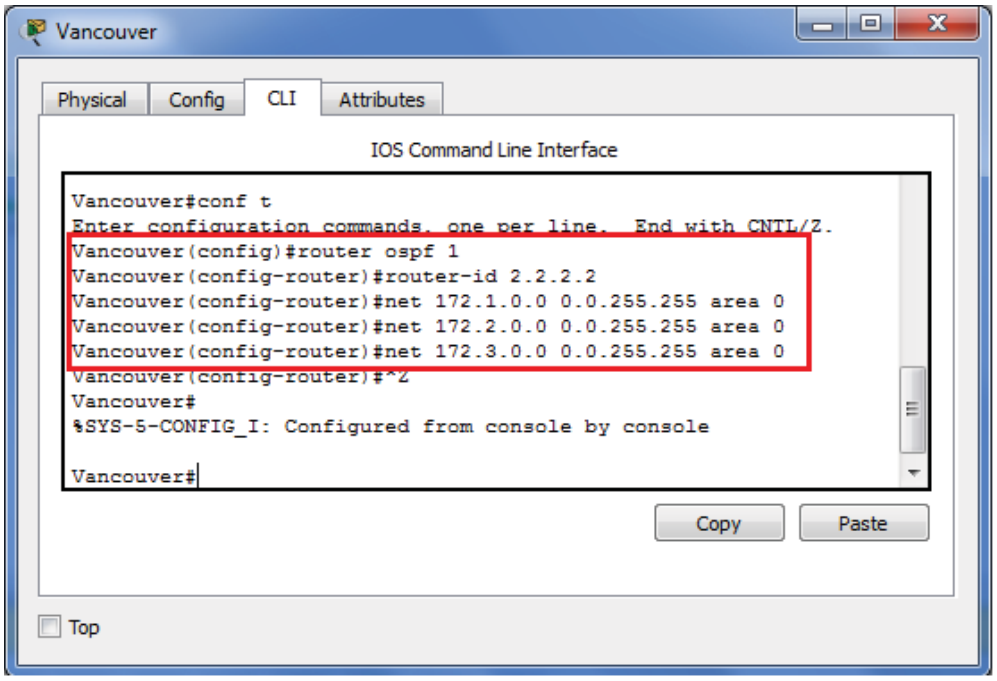

Figura 4.4: Configuración de OSPF en el *router* Vancouver.

4.2.2 Simulación de gestión

Para realizar la simulación de gestión, cada uno de los 25 *routers* se configuró con los permisos de comunidad establecidos como *"canarie",* que permiten el acceso a la información de los elementos para realizar la gestión de cada *router.* Para sustentar las pruebas de simulación de conectividad, se utiliza el *routers* ST. John's gestionado de forma remota desde el *host* ubicado en el *router* Victoria.

Packet Tracer utiliza una sección básica de SNMP, es decir, sólo permite algunas funciones, tal es el caso de permitir la configuración de las comunidades de lectura y escritura, para tener acceso a la información de gestión desde el *host* utilizado como NMS, sin embargo, al configurar la generación de *traps, Packet Tracer* indica a través de un mensaje que esa función no es aceptada. Dicha acción se presenta en la figura 4.5.

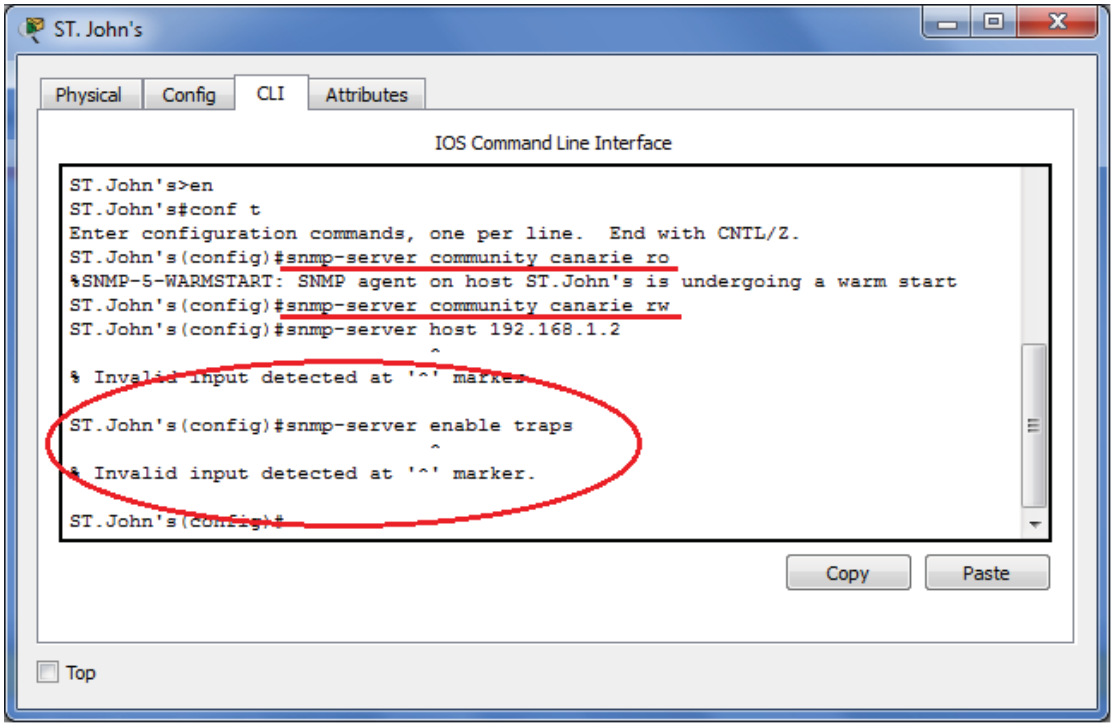

Figura 4.5: Activación de SNMP en el *router* ST. John's.

Por otra parte, el *host* que fungirá como gestor también debe ser configurado con los permisos de comunidad, establecidos anteriormente, esto con la finalidad de permitir el acceso a la información de gestión de cada elemento gestionado. La selección del MIB *Browser* se realiza en la pestaña *Desktop* del *host* utilizado como NMS, como lo muestra la figura 4.6. Una vez abierto el MIB *Browser* se procede a configurarlo, primero, se ingresa la dirección de red de la interfaz del elemento a gestionar, después, en la pestaña *advanced,* se asignan los permisos de lectura y escritura (*canarie*) y se selecciona la versión a utilizar del protocolo, en este caso, la versión 2.

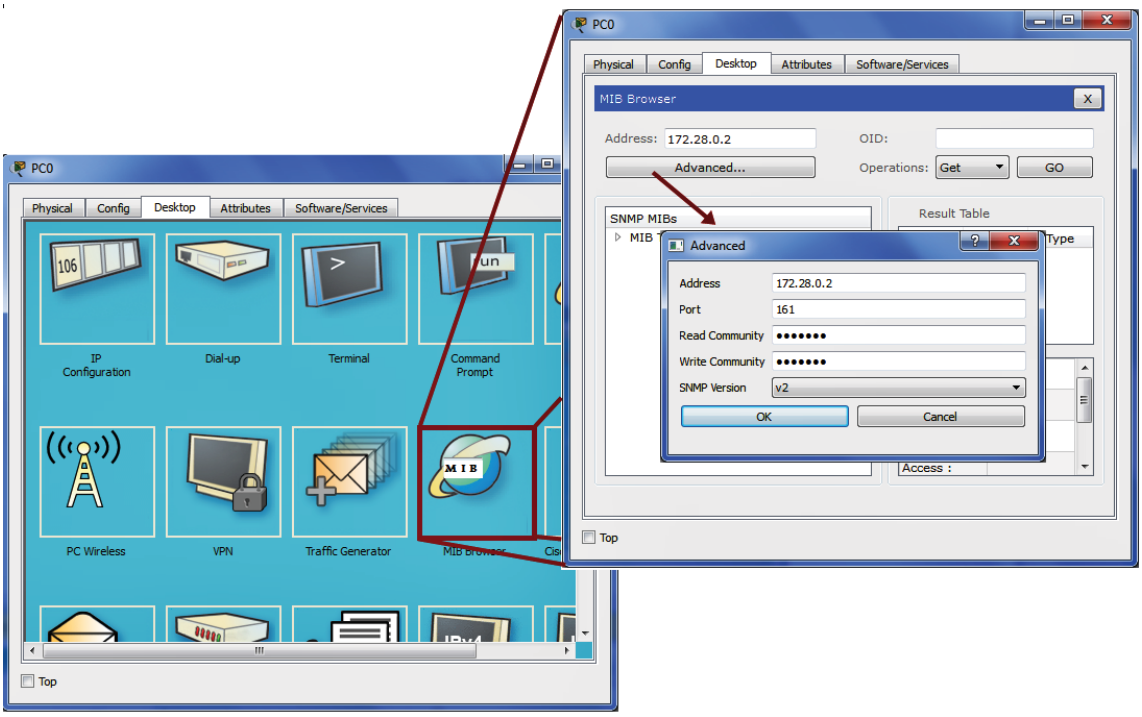

Figura 4.6: Configuración del MIB *Browser* en el *host* conectado al nodo Victoria.

La simulación de gestión se sustenta sobre el *router* ST. John's, sobre el cual se realizan las pruebas de conectividad, monitoreo y configuración de las variables, esto con la finalidad de comprobar que la teoría que describe la MIB-II es aplicable en *Packet Tracer.* Las variables seleccionadas para realizar la simulación de gestión son:

1. **sysName:** Variable con permisos de lectura y escritura, cuyo OID es 1.3.6.1.2.1.1.5. Esta variable contiene el nombre de dispositivo, asignado de forma manual por el administrador de la red.
2. **sysLocation:** Variable con permisos de lectura y escritura, cuyo OID es 1.3.6.1.2.1.1.6. Indica la ubicación física del dispositivo, proporcionada por el administrador de la red.
3. **IfNumber:** Variable con permisos de sólo lectura, cuyo OID es 1.3.6.1.2.1.2.1. Indica el número de interfaces soportadas por el dispositivo gestionado.
4. **ipDefaultTTL:** Variable con permisos de lectura y escritura, cuyo OID es 1.3.6.1.2.1.4.2. Contiene el valor predeterminado 255, insertado en el campo

Time To Live (TTL) de la cabecera IP, siempre y cuando el protocolo de transporte no proporcione un valor de TTL.

5. **ipAddTable:** Variable con permisos de sólo lectura, cuyo OID es 1.3.6.1.2.1.4.20. Contiene las direcciones de la tabla de enrutamiento generadas por los dispositivos gestionados.

4.3 Especificaciones técnicas para la emulación de conectividad y de gestión

Para desarrollar la emulación de conectividad y gestión, se utilizó una computadora de mejores características que la especificada en la sección 4.2, esto debido a que la primera computadora que se planeaba utilizar, únicamente soportó siete *routers* de la serie c7200 de Cisco y una máquina virtual, ejecutándose sin problemas en GNS3 versión 1.3.13, utilizada en computadoras con arquitectura de 32 *bits*. Considerando que el diseño de la figura 4.1 utiliza 25 *routers* y por lo menos dos máquinas virtuales como mínimo para realizar las pruebas de emulación de conectividad, se optó por utilizar un sistema denominado “Xexelo” construido bajo el proyecto “Laboratorio de Sistemas Distribuidos y Redes de Alto Desempeño”, alojado en el Laboratorio de Redes (B-404) de la UACM, plantel San Lorenzo Tezonco, en el cual se ejecuta un sistema operativo Windows 8 con arquitectura de 64 bits. La computadora cuenta con las siguientes características:

- Procesador Intel (R) Xeon (R) CPU E5-2620 v2 @ 2.10 GHz (24 CPUs), ~ 2.1 GHz. Cuenta con 12 núcleos y 24 procesadores lógicos [102].
- Memoria RAM de 32 GB.

En Windows 8 se instaló GNS3 versión 1.5.3, con licencia GLPv3 (*General Public License* – Licencia Pública General), versión más reciente hasta el momento para ejecutarse en plataformas de 64 *bits*.

El diseño sobre GNS3 para las pruebas de emulación de conectividad y gestión se muestra en la figura 4.7, la cual no difiere del diseño planteado originalmente en la figura 4.1. La configuración de la dirección de red IP de cada una de las interfaces se sigue de acuerdo a lo planeado en la tabla 4.1. Las direcciones IP para conectar a

las máquinas virtuales son las mismas que las utilizadas en la simulación de conectividad.

Figura 4.7: Diseño del *backbone* CANARIE sobre GNS3, para realizar la emulación de conectividad y de gestión.

En el emulador se utilizaron los siguientes equipos:

- IOS del *router* c7200, configurado con interfaces Gigabit Ethernet, con capacidad de conexión a 1 Gbps.
- 5 *switches* genéricos que funcionan en capa 2 del modelo ISO/OSI
- 5 máquinas virtuales, de las cuales: 2 con S.O. Ubuntu 16.04, 2 con S. O. Fedora 24 y 1 con Windows 8.

4.3.1 Emulación de conectividad

Para contrastar los resultados obtenidos mediante la emulación de conectividad con los obtenidos en la simulación de conectividad, se utilizaron los mismos puntos de conexión, es decir, se tomaron a los *routers* Victoria y ST. John's, con la finalidad de ver si el comportamiento del protocolo OSPF es igual en una plataforma de IOS real que en el simulador.

La configuración de la dirección de red de interfaz de la máquina virtual Windows 8 se muestra en la figura 4.8. Está máquina virtual se conecta directamente al *router* Victoria, para tener acceso al *backbone* CANARIE.

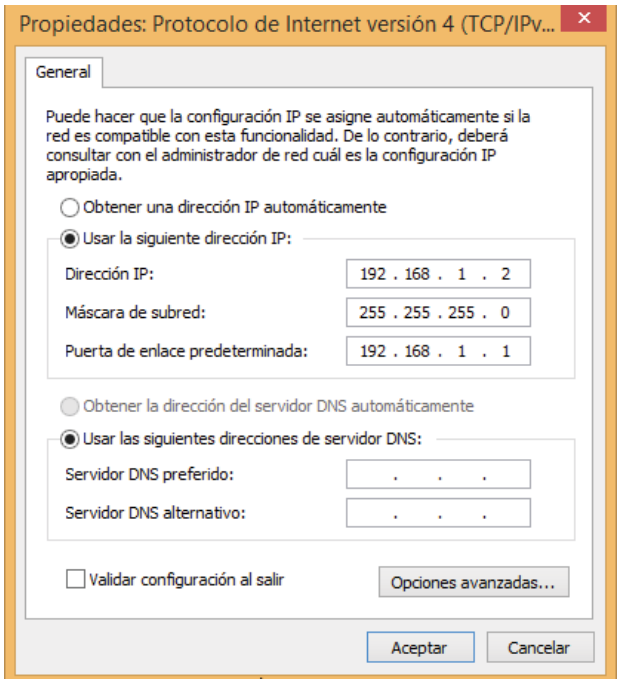

Figura 4.8: Configuración de la interfaz de red de la máquina virtual Windows.

Después de configurada la dirección de red de interfaz de la máquina virtual Windows 8, se puede ver en la figura 4.9, la velocidad de conexión establecida a 1 Gbps entre la máquina virtual y el *router* Victoria.

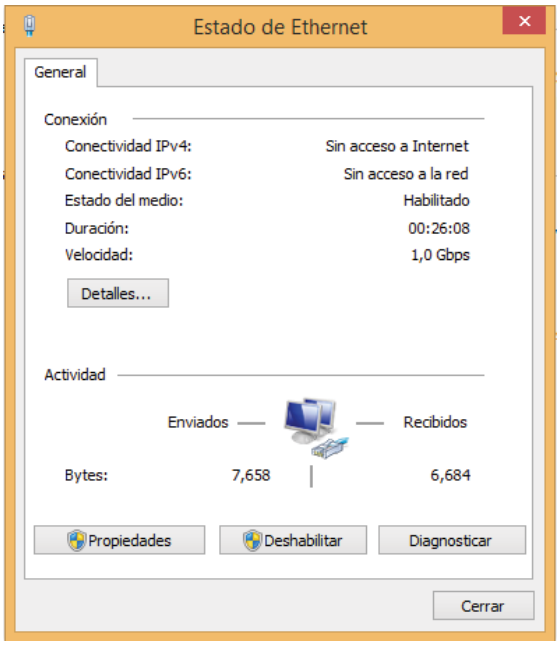

Figura 4.9: Conexión establecida a 1 Gbps entre la máquina virtual y el *router* Victoria en GNS3.

Por otra parte, el sistema operativo Ubuntu, también es configurada con una dirección de red de interfaz, como lo muestra la figura 4.10. Esto para lograr que ambas máquinas virtuales establezcan comunicación a través del *backbone*.

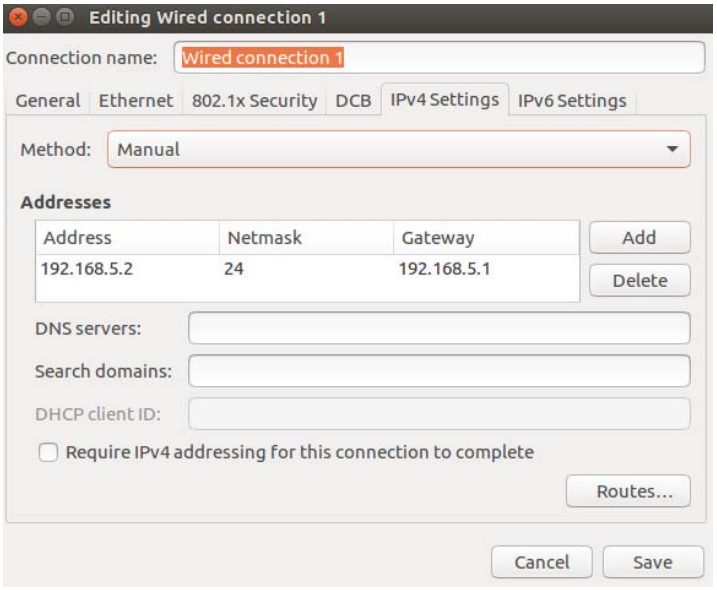

Figura 4.10: Configuración de la interfaz de red de la máquina virtual Ubuntu.

Para lograr que los *routers* c7200 construyan sus tablas de enrutamiento, estos se configuran de forma similar a como se configuraron los *routers* 2811 utilizados en

Packet Tracer. Las interfaces de los *routers* en GNS3 utilizan conexiones de *Gigabit Ethernet* a 1 Gbps, capacidad de conexión máxima permitida en GNS3. A cada una de las interfaces se les asigna una dirección IP, tal como se muestra en la figura 4.11 correspondiente a la configuración del *router* Vancouver.

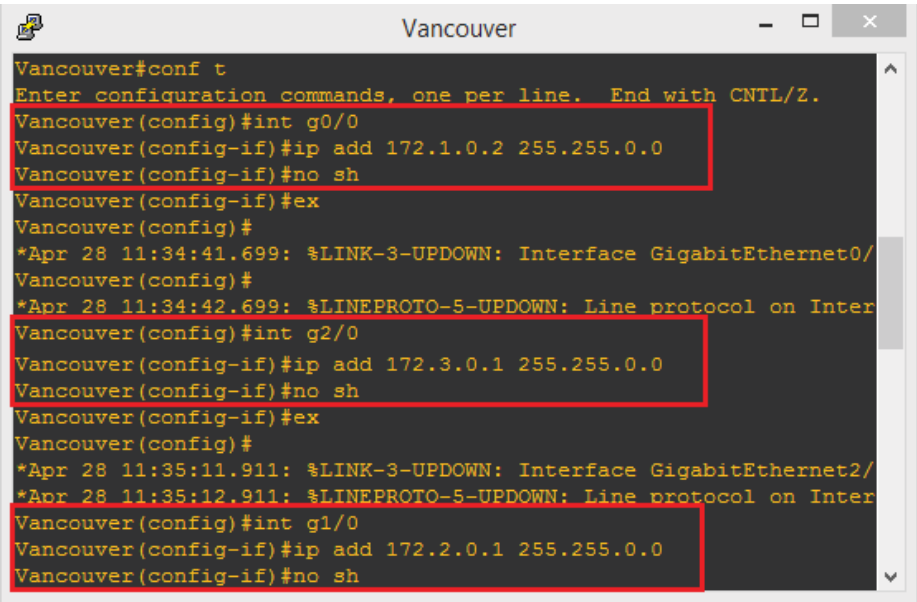

Figura 4.11: Configuración de las interfaces en el nodo Vancouver.

Después de activar las interfaces de cada *router,* el siguiente paso es activar el protocolo de enrutamiento OSPF, como se indica en la figura 4.12. La configuración de OSPF en cada uno de los 25 *routers* no difiere de la configuración OSPF en *Packet Tracer.*

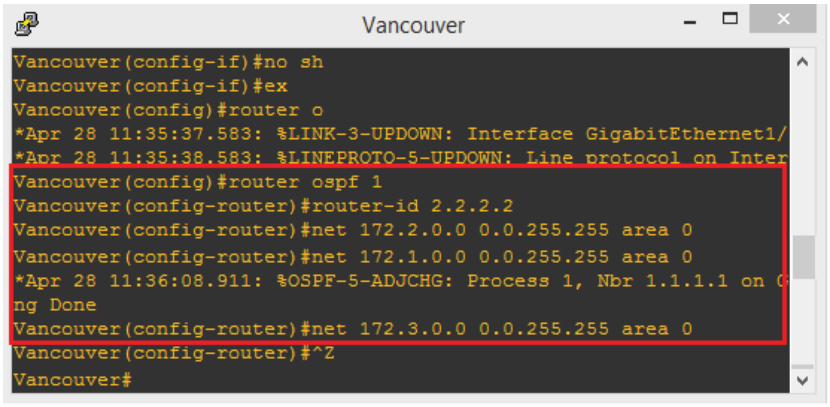

Figura 4.12: Configuración de OSPF en el *router* Vancouver.

4.3.2 Emulación de gestión

Para la emulación de gestión, se configuró cada uno de los 25 *routers* con el protocolo SNMP, tal como se muestra en la figura 4.13. Además de asignar los permisos de comunidad establecidos como *"canarie"* para acceder a la información de gestión, también se asignó la dirección de interfaz del *host* que opera como gestor de la red, y se habilitó el uso de los *traps*. Otros elementos configurados fueron el contacto (nombre del administrador de la red) y la ubicación del equipo gestionado.

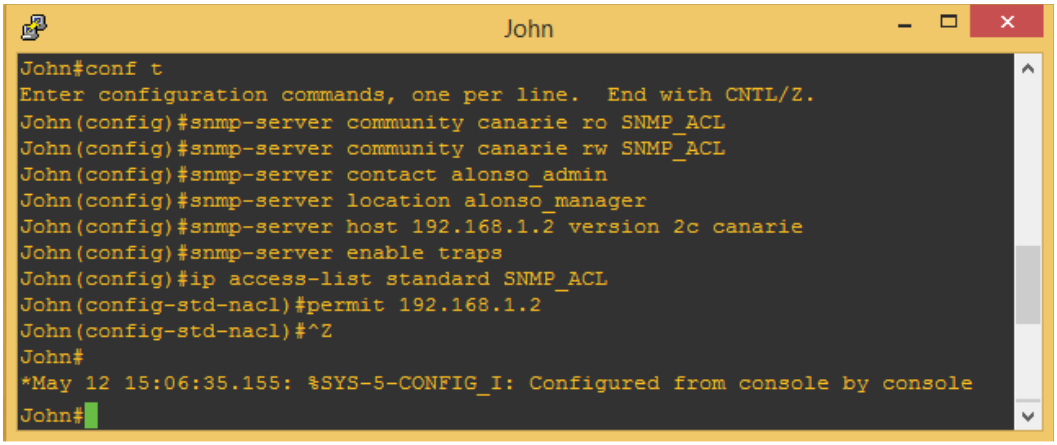

Figura 4.13: Configuración SNMP en el nodo ST. John's.

A diferencia de *Packet Tracer*, en GNS3, el IOS del *router* c7200 habilitado mediante SNMP permite crear una lista de equipos que podrían funcionar como gestores de la red, sin embargo, sólo se utilizará un único NMS para gestionar el *backbone* CANARIE.

La herramienta utilizada como aplicación de gestión de red es la *ireasoning MIB Browser,* instalada mediante interfaz gráfica de usuario (GUI) en la máquina virtual que utiliza Windows 8. Una vez instalada la aplicación, se asignaron los permisos de comunidad que permiten el acceso a la información de gestión como se muestra en la figura 4.14.

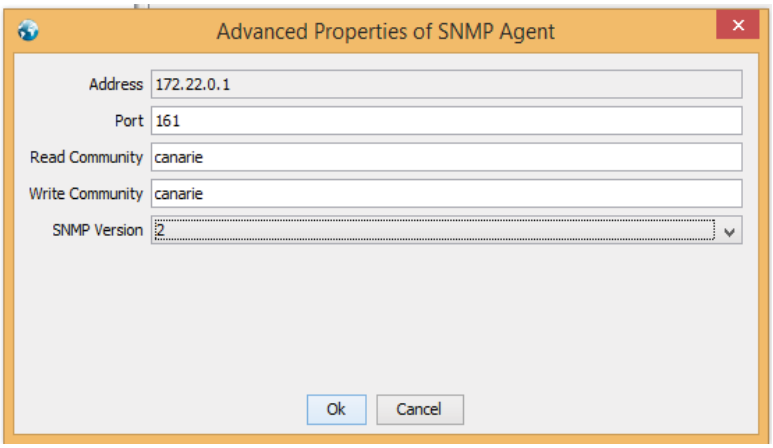

Figura 4.14: Configuración de acceso a la MIB de los elementos gestionados desde el *host* conectado al *router* Victoria.

Para comparar resultados de la emulación de gestión con la simulación de gestión, se toman las mismas variables propuestas en la sección 4.2.2 con la finalidad de realizar la emulación de gestión en el *router* ST. Jonh's. A las variables también se les realizan pruebas de monitoreo, configuración.

Capítulo 5: Resultados y Conclusiones

5.1 Resultados para la simulación de conectividad

Los *routers* configurados con OSPF después de entrar en operación establecen adyacencia con sus vecinos, dicha adyacencia puede ser consultada con el mando *"show ospf neighbor".* La figura 5.1 muestra la adyacencia generada por el *router* Saskatoon a través de las interfaces seriales 0/2/0 y 0/2/1, la figura también muestra las prioridades asignadas a cada interfaz, así como un estado completo (*full*) indicando que la adyacencia entre sus vecinos se construyó satisfactoriamente. Otros valores observados en la figura 5.1 son el tiempo en segundos que dura la adyacencia y las direcciones de interfaz por las cuales se ha establecido la relación de vecinos.

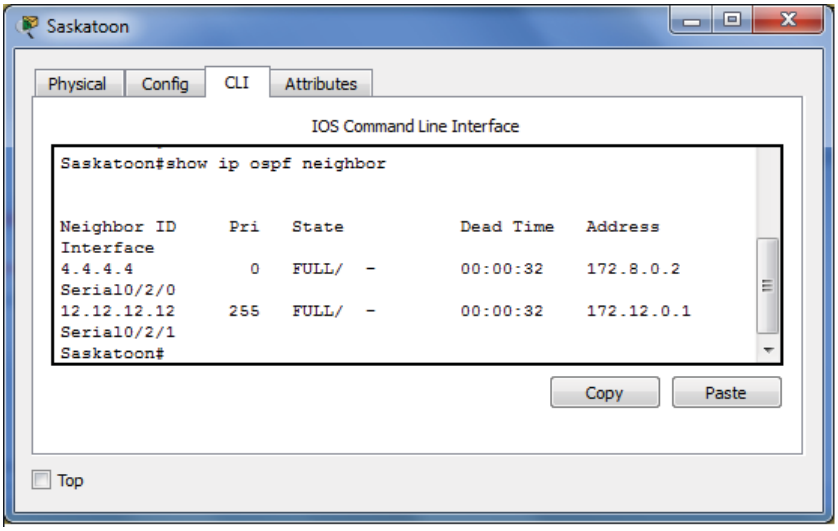

Figura 5.1: Adyacencia del *router* Saskatoon mostrando al *router* Winnipeg como DR.

La figura 5.2 muestra la tabla de enrutamiento generada por el *router* Winnipeg, la cual se consigue mediante el mando *"show ip route"*. En la tabla se distinguen 35 rutas, a través de las cuales se puede enviar un paquete hacia una red destino. La tabla muestra una serie de códigos para indicar la naturaleza de las rutas, sin embargo, las rutas contenidas en la tabla indican los siguientes códigos:

- O – Indica que la ruta es administrada mediante el protocolo OSPF.
- C – Indica una ruta local, es decir, la ruta pertenece al *router* analizado.
- O IA – Indica que la ruta pertenece a un área diferente al área cero o de *backbone.*

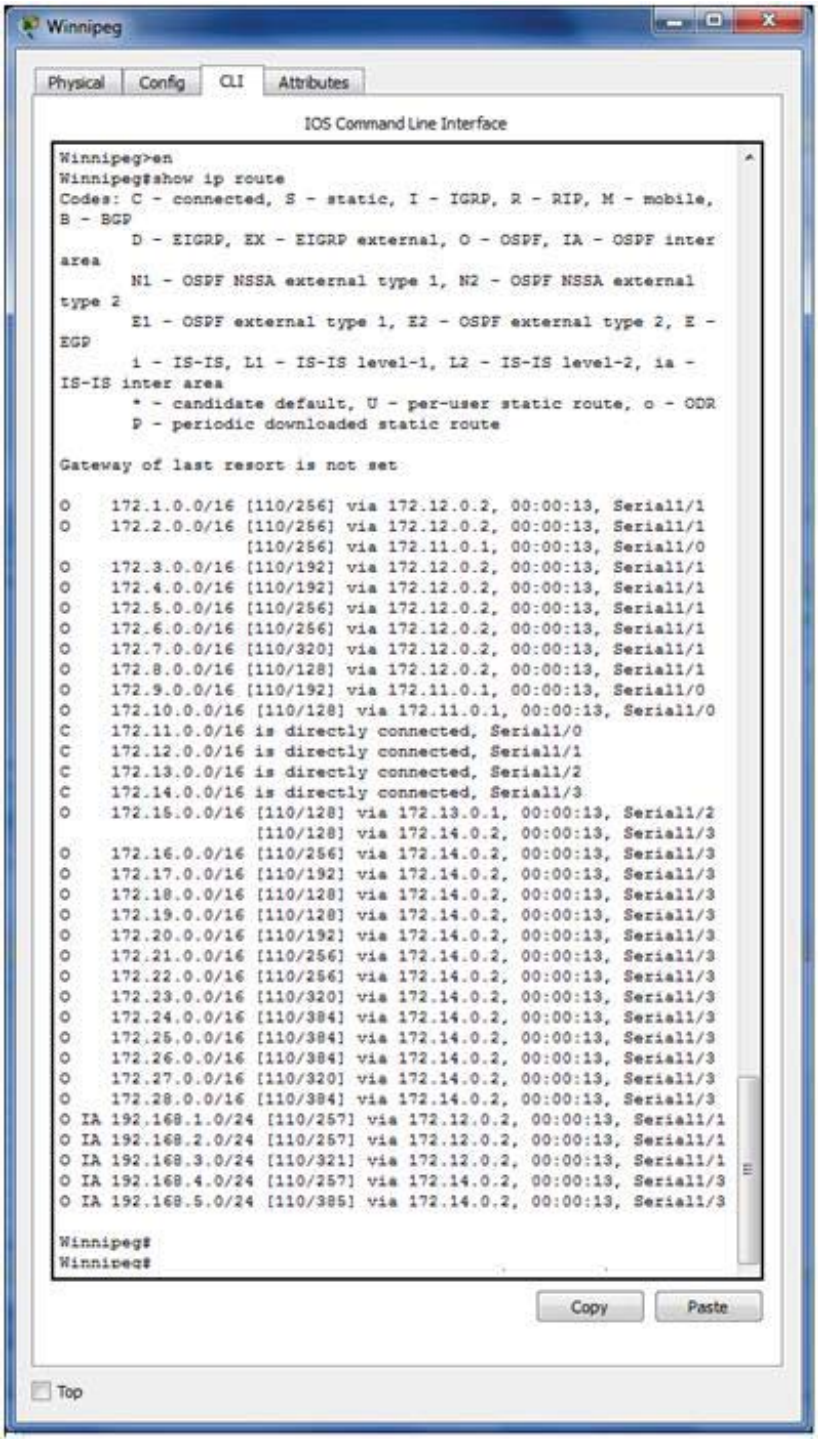

Figura 5.2: Tabla de enrutamiento del *router* Winnipeg.

Las pruebas de simulación de conectividad se realizan utilizando el mando ping (*Packet Internet Groper*), herramienta utilizada primordialmente para diagnosticar la comunicación entre los diferentes dispositivos de una red IP, a través del envío de paquetes ICMP como solicitud (*echo request*) y respuesta (*echo reply*).

Packet Tracer se configuró en modo simulador, con la finalidad de poder ver el recorrido que realizan los paquetes ICMP enviados para verificar la conexión entre los *host*. En la figura 5.3 se muestra la ruta que siguen los paquetes ICMP enviados desde el *host* conectado al *router* Victoria hacia el *host* del *router* ST. John's.

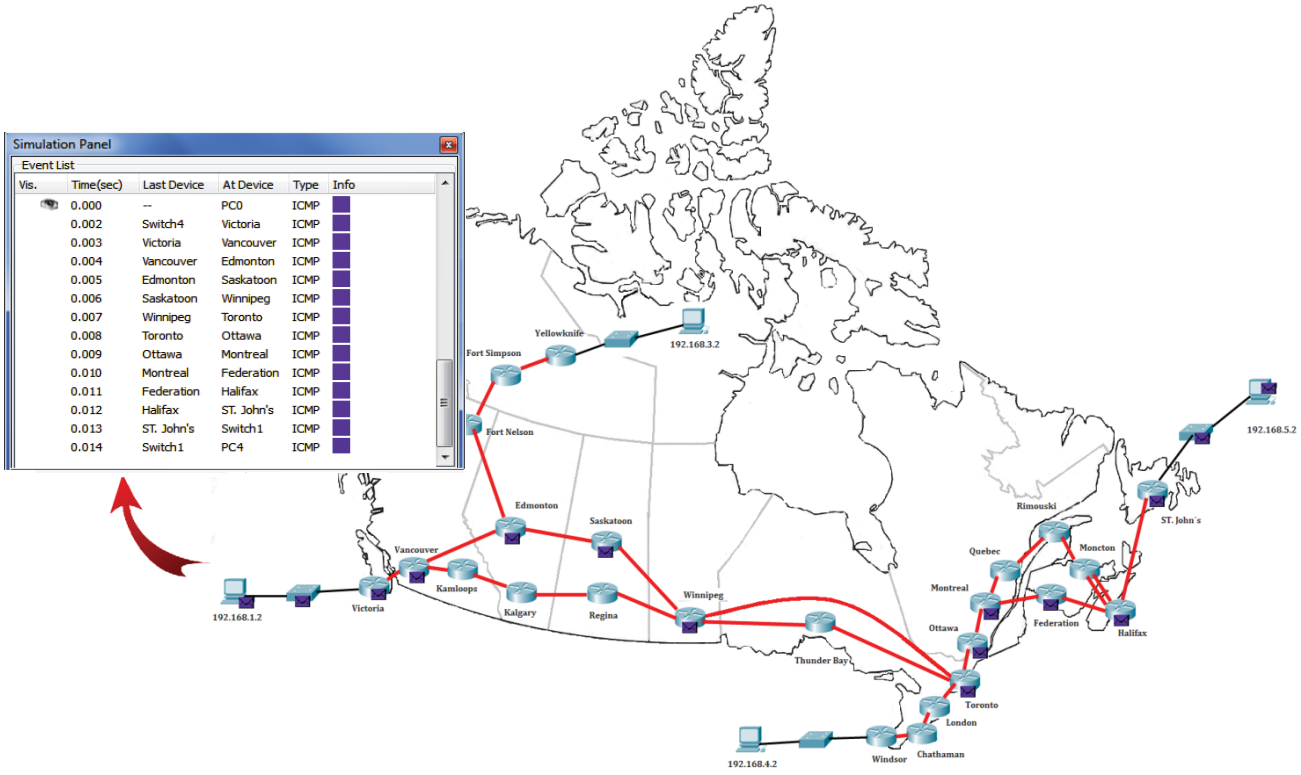

Figura 5.3: Ruta que sigue el mando ping desde el *router* Victoria hasta el *router* ST. John's.

Cuando los ICMP alcanzan su destino, el *host* conectado al *routers* ST. John's genera la respuesta y la envía al *host* que inició la comunicación, indicándole su disponibilidad para generar el intercambio de información. La figura 5.4 muestra el proceso de respuesta al ping, los paquetes siguen la misma ruta por la que se realizó la petición de conexión.

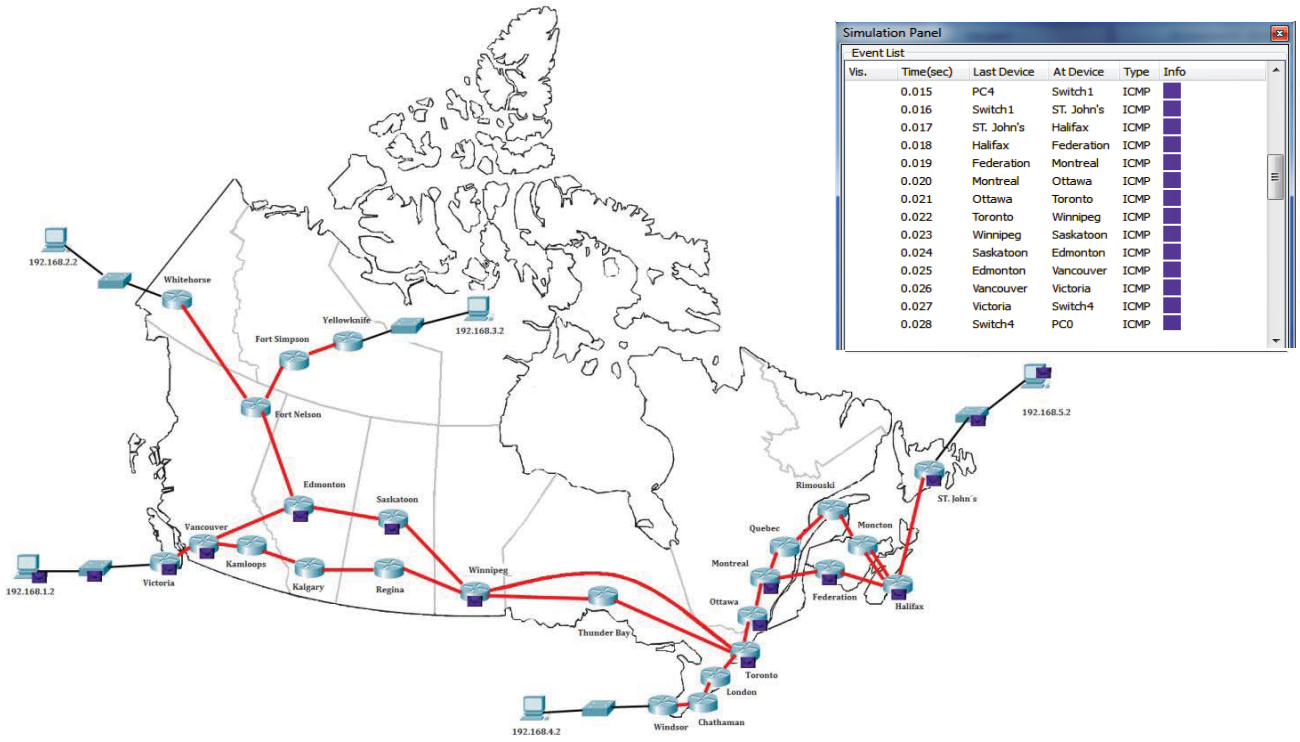

Figura 5.4: Respuesta desde el *router* ST. John's para confirmar la recepción del ping.

Uno de los objetivos planteados fue verificar la estabilidad del simulador, es por ello que se realizó el monitoreo de consumo de recursos de CPU y memoria en la computadora donde se realizaron las pruebas de conectividad. Los resultados se presentan en la tabla 5.1.

Recursos	Computadora sin actividad	Simulador activo sin operación	Ejecutando pruebas de simulación
CPU	1%	29%	32%
Memoria	29% (912 MB de 3 GB)	40% (1.20 GB de 3 GB)	40% (1.20 GB de 3 GB)
Tiempo		51.6 seg.	2.48 min.

Tabla 5.1: Recursos de la computadora consumidos por el simulador.

5.2 Resultados para la simulación de gestión

La simulación de gestión consistió en realizar el monitoreo y configuración de las variables de la MIB-II contenidas en el MIB *Browser,* instalado por defecto en *Packet Tracer*. Al igual que para la simulación de conectividad, en la simulación de gestión, es posible ver el recorrido que realizan los paquetes SNMP, enviados hacia el dispositivo gestionado en busca de la información de gestión. La figura 5.5 muestra el mensaje enviado al *router* ST. John's, solicitando la información almacenada en la MIB.

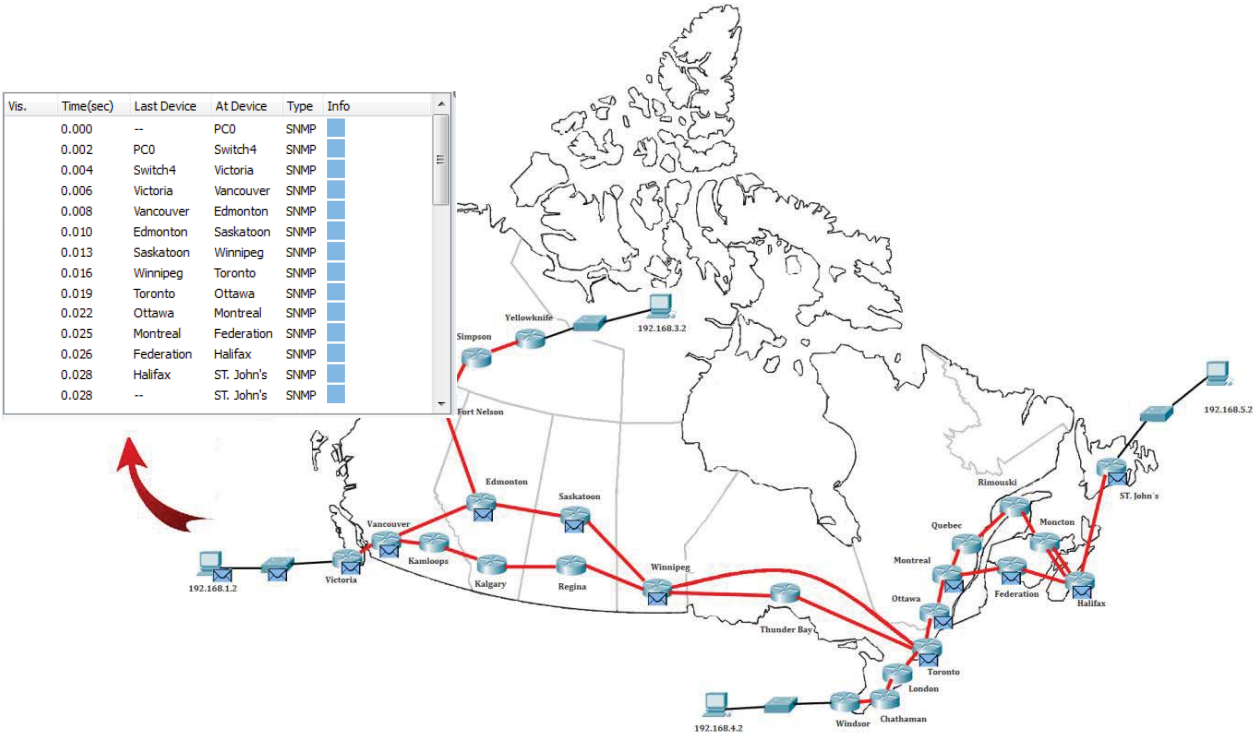

Figura 5.5: Mensajes SNMP solicitando información de la MIB en el *router* ST. John's.

Una vez que el mensaje alcanzó al dispositivo gestionado, el agente genera un mensaje respuesta y lo envía de regreso a la estación de gestión, tal como se muestra en la figura 5.6.

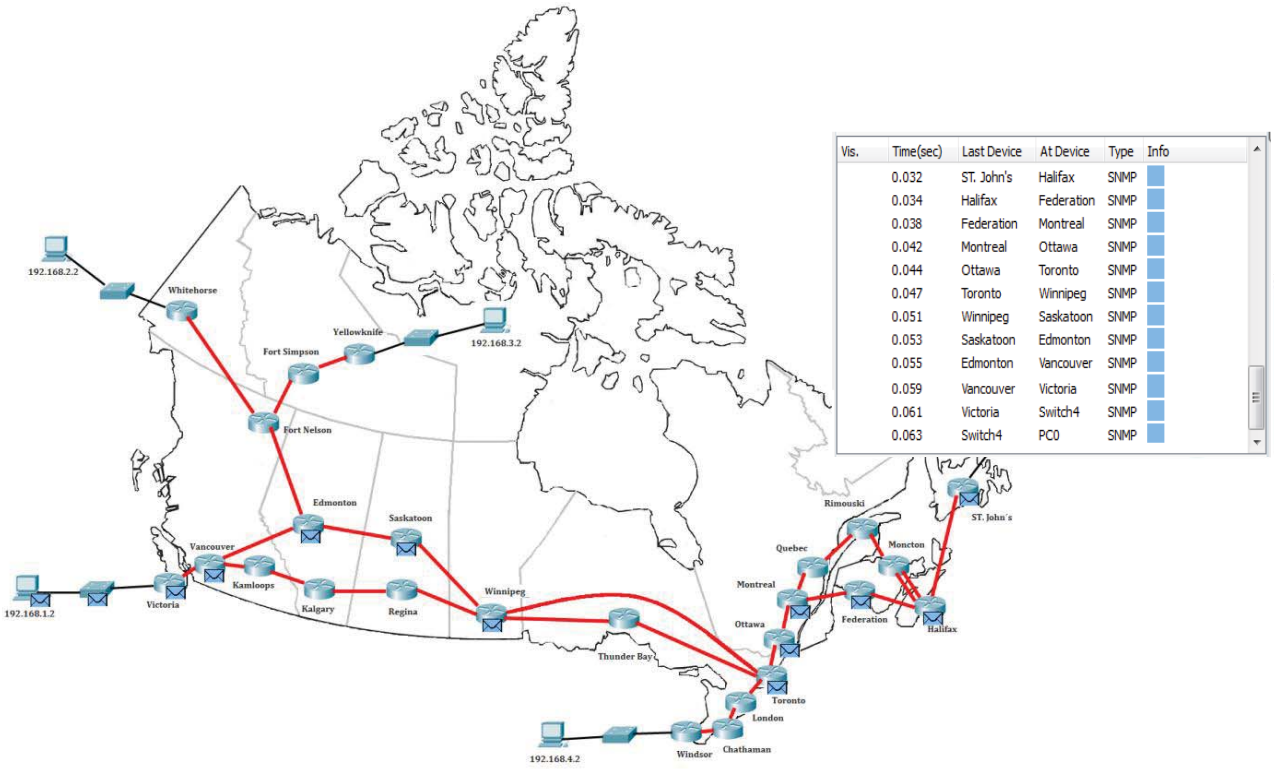

Figura 5.6: Respuesta generada por el agente ante una solicitud de información por parte del gestor.

A continuación, se presentan los resultados obtenidos en cada una de las cinco variables propuestas en el capítulo 4, con lo cual se sustentan las pruebas de la simulación de gestión.

sysName

Esta variable contiene el nombre asignado de forma manual al dispositivo gestionado. La figura 5.7 muestra el valor contenido en dicha variable, el cual se consiguió a través de la operación *get.*

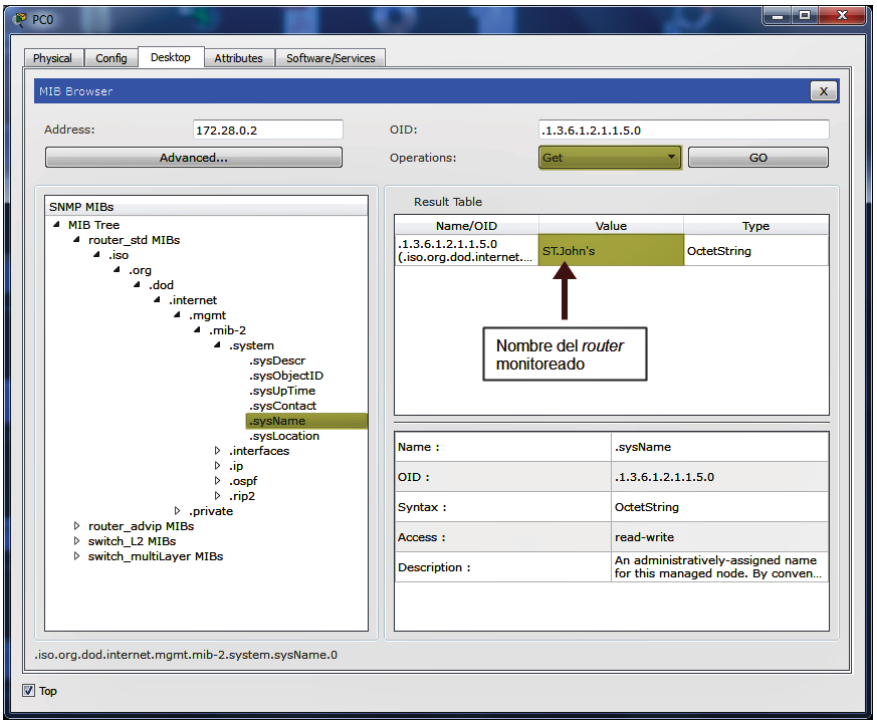

Figura 5.7: Monitoreo del nombre del *router* ST. John's, vía la variable *sysName.*

La variable monitoreada en la figura 5.7 indica que cuenta con permisos de lectura y escritura, por lo cual, es posible modificar su valor. Utilizando la operación *set,* se le asigna el nuevo valor a la variable como *"Router_Jony",* dentro del campo marcado como *value*, en la nueva ventana desplegada al utilizar la operación *set,* tal como se muestra en la figura 5.8.

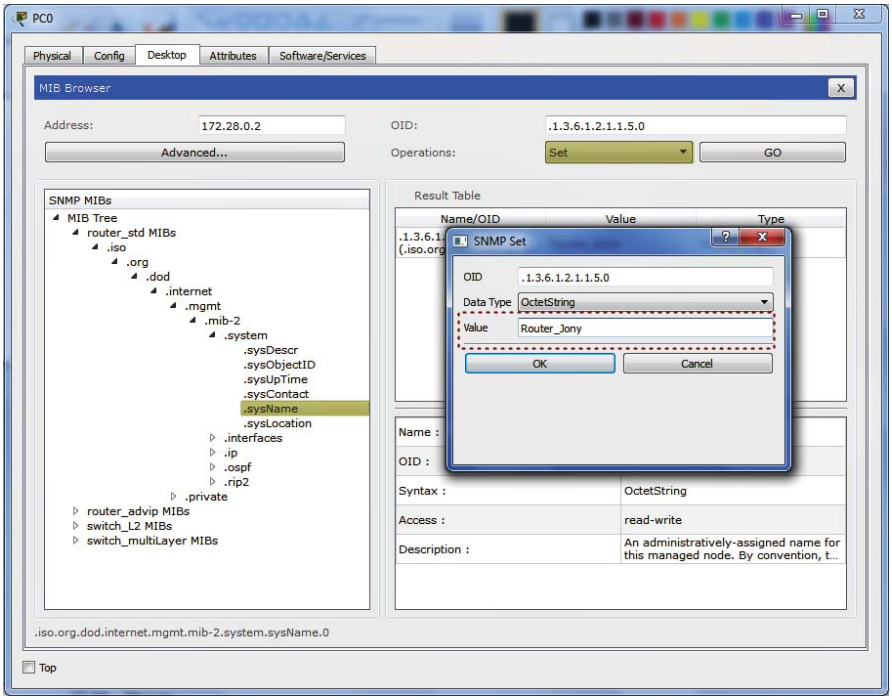

Figura 5.8: Configuración del nombre de la variable *sysName,* a través de la operación *Set.*

Se acepta el cambio y se procede a recuperar el nuevo valor de la variable mediante un *get.* La figura 5.9 muestra la configuración exitosa de la variable *sysName*.

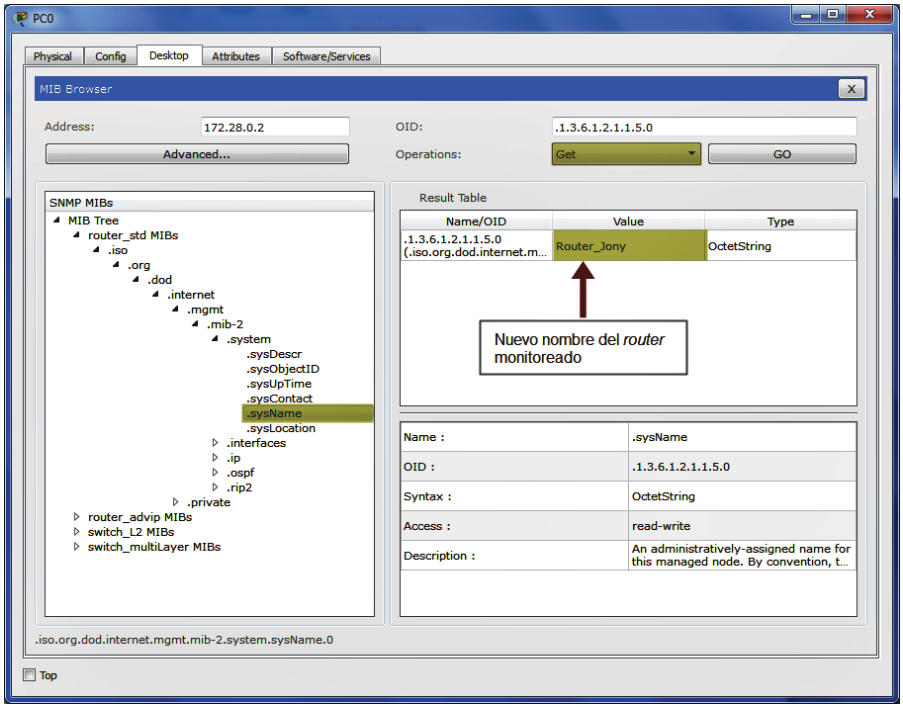

Figura 5.9: Cambio del nombre del *router* ST. John's, vía configuración.

sysLocation

Esta variable se encarga de indicar la ubicación física del dispositivo gestionado, siempre que así se esté configurado. *Packet Tracer* no permitió la configuración manual de esta variable, es por ello que en la figura 5.10 se observa un valor nulo para la variable monitoreada.

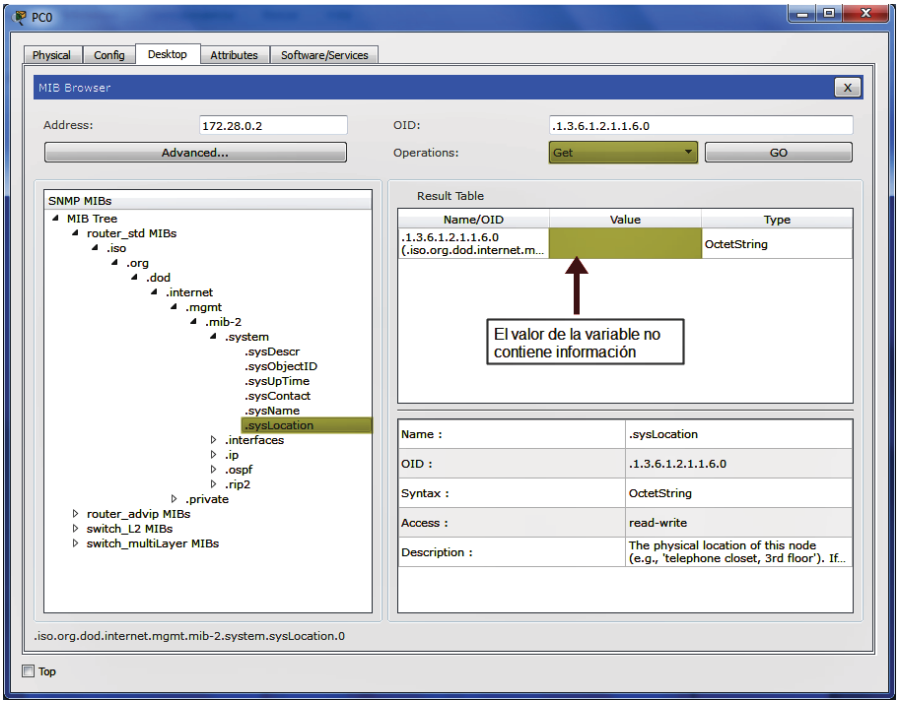

Figura 5.10: Monitoreo de la ubicación del *router* ST. John's, vía la variable *sysLocation.*

La variable *sysLocation* contiene permisos de lectura y escritura, según lo indicado en la figura 5.10, es por ello que mediante la operación *Set,* se intentará configurar dicha variable, cambiando su valor a "Labrador_John". La figura 5.11 muestra el proceso de configuración.

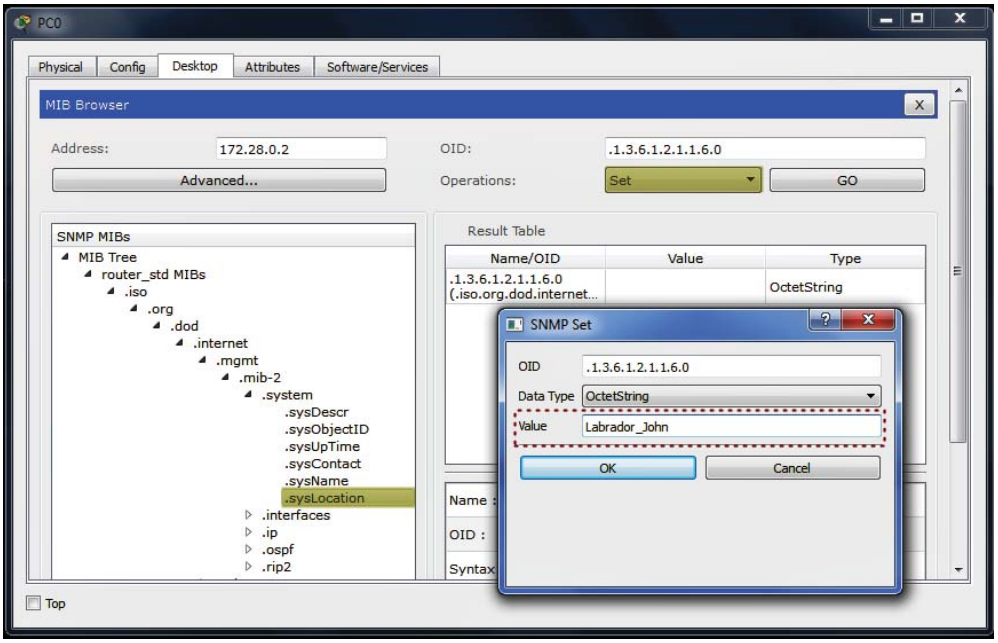

Figura 5.11: Cambio del valor de la variable *sysLocation,* vía configuración.

Después de aceptar los cambios, el resultado de la configuración de la variable se muestra en la figura 5.12.

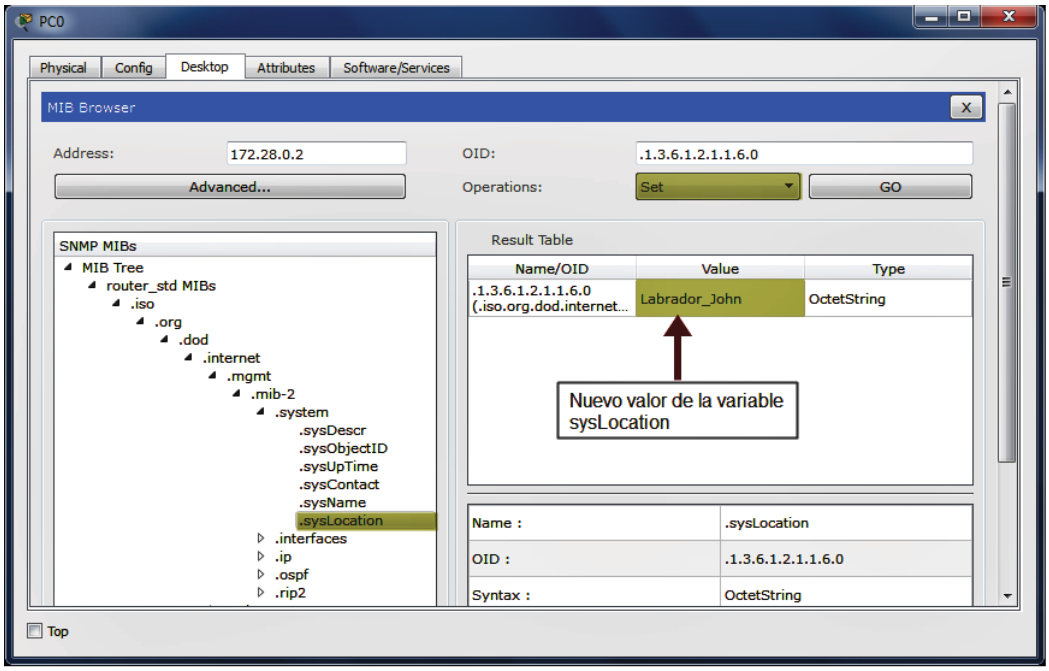

Figura 5.12: Cambio de la ubicación del *router* ST. John's, vía configuración.

ifNumber

Esta variable contiene la cantidad de interfaces existentes en el *router* monitoreado. El valor de la variable se muestra en la figura 5.13, la cual se consigue mediante la operación *get.*

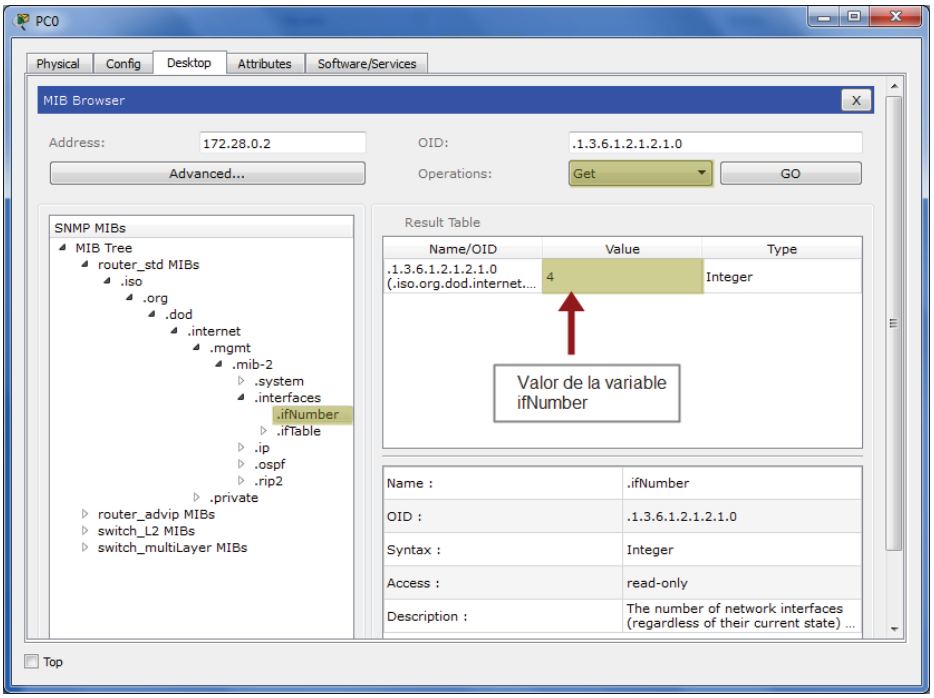

Figura 5.13: Monitoreo del número de interfaces en el *router* ST. John's.

La variable *ifNumber* indica permisos de sólo lectura, de acuerdo al resultado obtenido en la figura 5.13. Sin embargo, se llevará a cabo el intento de configuración de dicha variable. En la figura 5.14 se muestra la configuración de la cantidad de interfaces existentes en el *router* ST. John's.

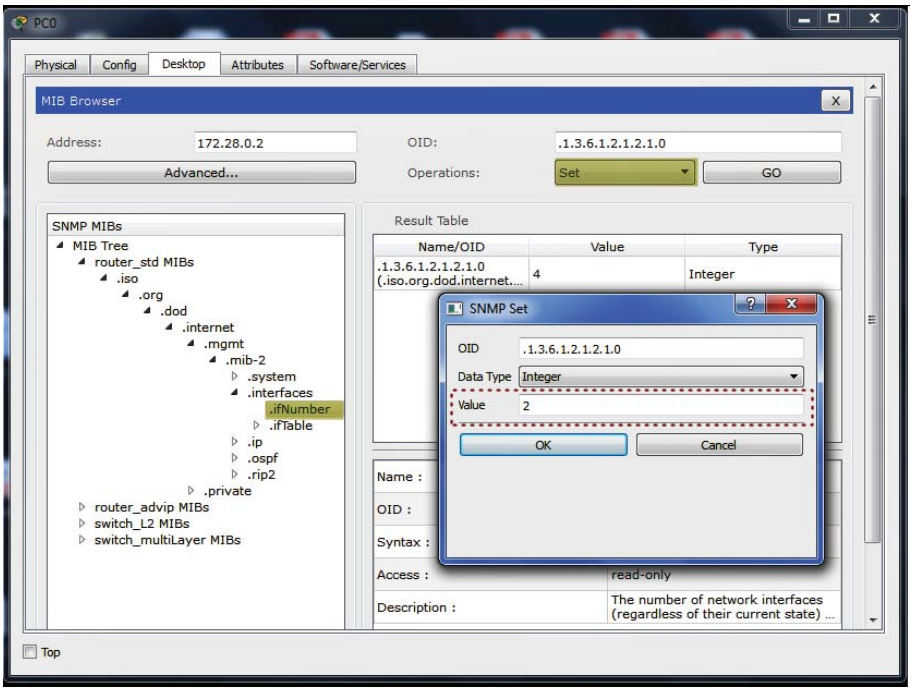

Figura 5.14: Cambio de valor de la variable *ifNumber,* vía configuración.

En la figura 5.15 se muestra un mensaje generado por el agente ejecutado en el *router* gestionado, el cual indica que no es posible completar la acción solicitada, debido a que el valor de la variable contiene únicamente permisos de lectura.

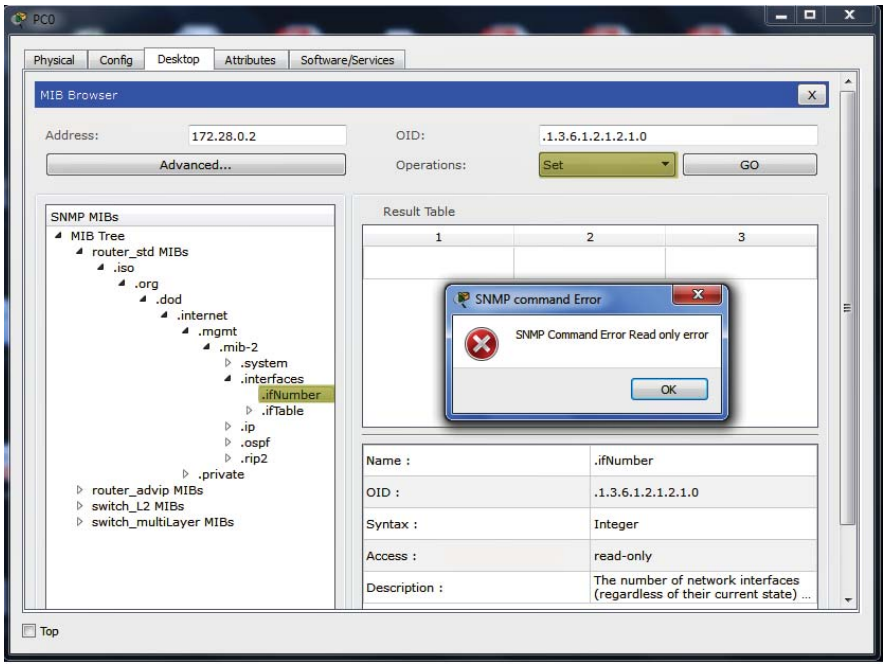

Figura 5.15: Mensaje enviado por el agente, indicando que la variable no puede modificarse, debido a que no cuenta con permisos de escritura.

ipDefaultTTL

Esta variable contiene el tiempo de vida de un paquete enviado en la cabecera IP. La figura 5.16 muestra que la variable propuesta no está soportada en el simulador *Packet Tracer,* versión 7.

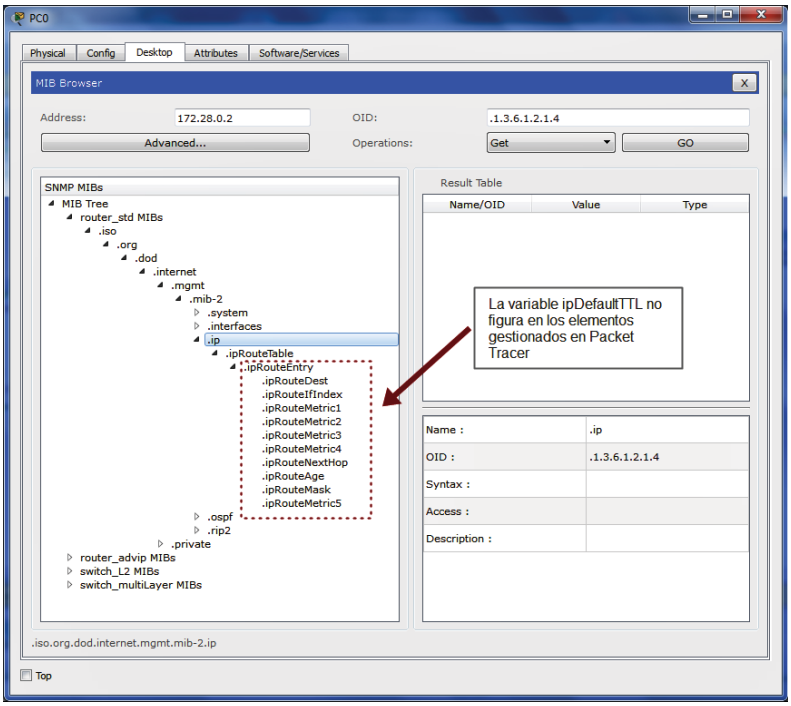

Figura 5.16: La variable *ipDefaultTTL* no figura en la MIB de *Packet Tracer* versión 7.

ipAddrTable

Esta variable contiene la tabla de enrutamiento del dispositivo gestionado.

La figura 5.17 indica que la variable propuesta, no figura en el árbol de gestión del MIB *Browser* en *Packet Tracer* versión 7. Por lo que es imposible realizar monitoreo de dicha variable.

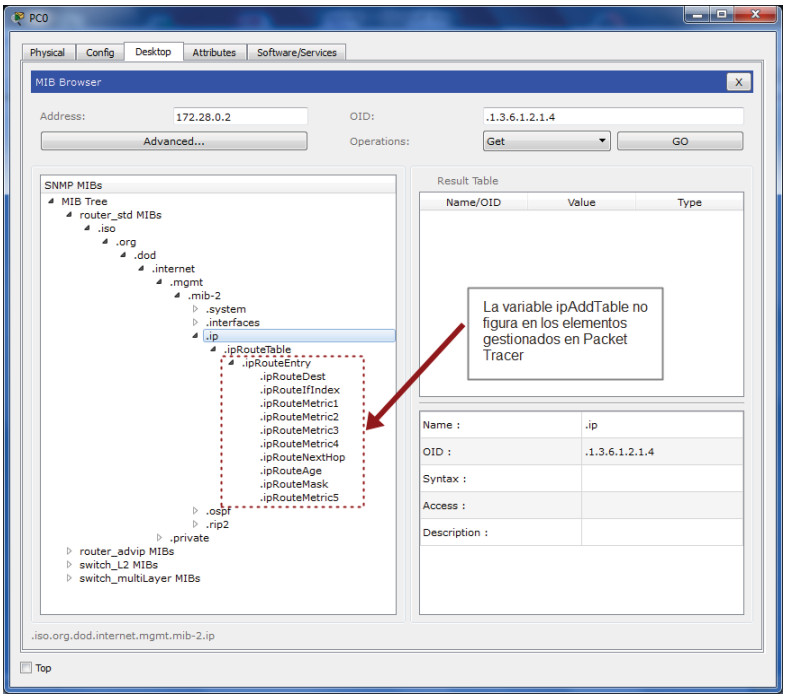

Figura 5.17: La variable *ipDefaultTTL* no figura en la MIB de *Packet Tracer* versión 7.

En la tabla 5.2 se hace un resumen de los resultados obtenidos al monitorear y configurar las variables descritas con anterioridad. Cabe señalar que, tres de las cinco variables cumplen con lo argumentado en la teoría, las otras dos, al no figurar en el MIB *Browser* de *Packet Tracer,* no permiten argumentar a favor o en contra de lo que la teoría dicta. Por lo tanto, las variables a las cuales se les realizó gestión están definidas correctamente y cumplen con los parámetros indicados en el RFC 1213.

Variable	Monitoreo	Configuración	Estatus
sysName	✓	✓	Cumple con la teoría
sysLocation	✓	✓	Cumple con la teoría
ifNumber	✓	X	Cumple con la teoría
ipDefaultTTL	-	-	No hubo elementos para argumentar
ipAddTable	-	-	No hubo elementos para argumentar

Tabla 5.2: Resumen de las variables gestionadas.

5.3 Resultados de emulación de conectividad

Para realizar la emulación de conectividad utilizando GNS3, se sigue el mismo procedimiento utilizado en la simulación de conectividad. Para verificar que la configuración de los *routers* no presenta problemas, se analiza la relación de vecinos de cada *router,* utilizando el mando *"show ip ospf neighbor",* cuyo resultado se presenta en la figura 5.18 para el *router* Saskatoon. El IOS del *router* c7200 establece por defecto la prioridad en 1, para los *routers* a los cuales no se les ha realizado modificaciones en dicha prioridad, además, indica cual es la función de los *routers* vecinos, ya sea DR o BDR. La figura también muestra la interfaz por la cual se ha construido la adyacencia, así como el intervalo de tiempo en la que se mantiene vigente dicha adyacencia.

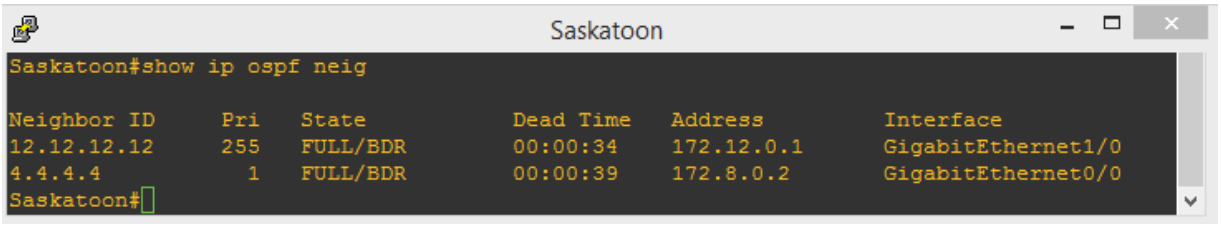

Figura 5.18: Adyacencias del *router* Saskatoon.

Otra forma de verificar que los *routers* han establecido relación de vecinos es a través del análisis de la tabla de enrutamiento, en la cual deben aparecer por lo menos las rutas de los *routers* vecinos. La tabla se consigue con el mando *"show ip route".* La figura 5.19 muestra la tabla de enrutamiento del *router* Winnipeg, la cual contiene 43 rutas almacenadas, a través de las cuales se puede establecer conexión con cualquier *host* conectado en la red.

Winnipeg

```
Winnipeg#
Winnipeg#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is not set

O     172.1.0.0/16 [110/4] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O     172.2.0.0/16 [110/4] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
                   [110/4] via 172.11.0.1, 00:13:07, GigabitEthernet0/0
O     172.3.0.0/16 [110/3] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O     172.4.0.0/16 [110/3] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O     172.5.0.0/16 [110/4] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O     172.6.0.0/16 [110/4] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O     172.7.0.0/16 [110/5] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O     172.8.0.0/16 [110/2] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O     172.9.0.0/16 [110/3] via 172.11.0.1, 00:13:07, GigabitEthernet0/0
O     172.10.0.0/16 [110/2] via 172.11.0.1, 00:13:07, GigabitEthernet0/0
      172.11.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.11.0.0/16 is directly connected, GigabitEthernet0/0
L        172.11.0.2/32 is directly connected, GigabitEthernet0/0
      172.12.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.12.0.0/16 is directly connected, GigabitEthernet1/0
L        172.12.0.1/32 is directly connected, GigabitEthernet1/0
      172.13.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.13.0.0/16 is directly connected, GigabitEthernet2/0
L        172.13.0.2/32 is directly connected, GigabitEthernet2/0
      172.14.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.14.0.0/16 is directly connected, GigabitEthernet3/0
L        172.14.0.1/32 is directly connected, GigabitEthernet3/0
O     172.15.0.0/16 [110/2] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
                    [110/2] via 172.13.0.1, 00:13:07, GigabitEthernet2/0
O     172.16.0.0/16 [110/4] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.17.0.0/16 [110/3] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.18.0.0/16 [110/2] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.19.0.0/16 [110/2] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.20.0.0/16 [110/3] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.21.0.0/16 [110/4] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.22.0.0/16 [110/4] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.23.0.0/16 [110/5] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.24.0.0/16 [110/6] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.25.0.0/16 [110/6] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.26.0.0/16 [110/6] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.27.0.0/16 [110/5] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O     172.28.0.0/16 [110/6] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
O IA  192.168.0.0/16 [110/5] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
                     [110/5] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O IA  192.168.1.0/24 [110/5] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O IA  192.168.3.0/24 [110/6] via 172.12.0.2, 00:03:23, GigabitEthernet1/0
O IA  192.168.5.0/24 [110/7] via 172.14.0.2, 00:13:07, GigabitEthernet3/0
```

Figura 5.19: Tabla de enrutamiento del *router* Winnipeg.

El emulador GNS3 al ser un sistema que ejecuta sistemas operativos reales, no se da el lujo de mostrar el recorrido de los paquetes a través de la red, tal como lo hace *Packet Tracer*. En su defecto, permite el uso de la herramienta *Wireshark,* la cual es utilizada para analizar las redes de datos y darle solución a posibles fallas que pudieran surgir. Se caracteriza por utilizar herramientas que facilitan el análisis de los protocolos de comunicación, además, ofrece un ambiente amigable al usuario

proporcionando una interfaz gráfica sobre la cual es posible ver el tráfico que pasa por la red analizada.

La emulación de conectividad se sustenta entre el *host* que ejecuta Windows 8 conectado al *router* Victoria, y el *host* con Ubuntu 15.04, conectado al *router* ST. John's, de acuerdo al diseño de la figura 4.7. La prueba de emulación de conectividad se realizó mediante el envío de un ping entre ambos *host.* La figura 5.20 muestra una conexión exitosa entre ambos extremos de la red, tomando en cuenta que los 15 paquetes enviados desde Windows 8, se recibieron exitosamente, garantizando que el enlace entre los *host* se encuentra en correcto funcionamiento.

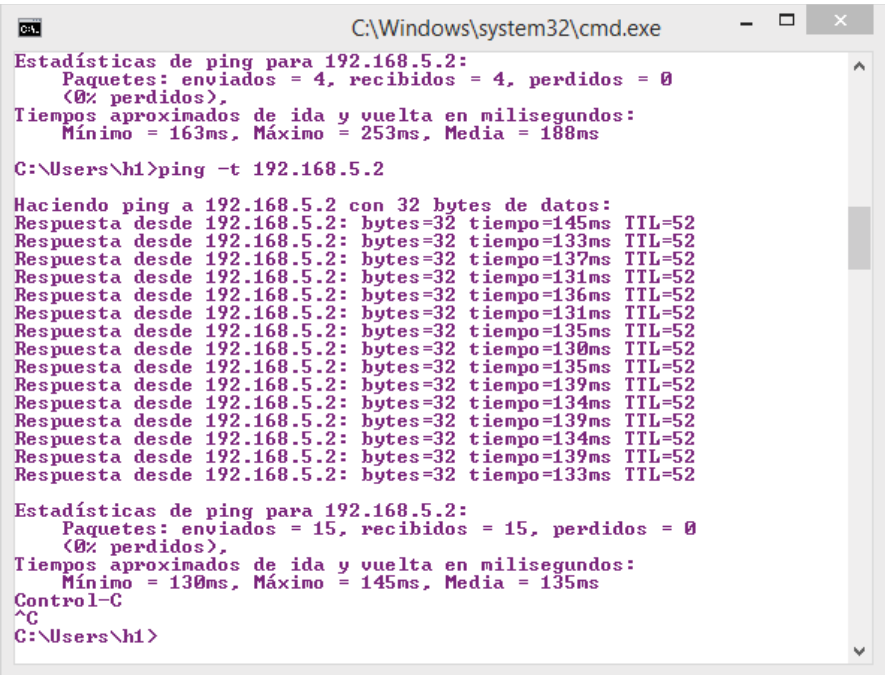

Figura 5.20: Prueba de conectividad desde el nodo Victoria hasta el nodo ST. John's mediante el uso del mando ping.

Una forma útil de conocer la ruta que siguen los paquetes en una red de datos es mediante el uso de la herramienta *"traceroute",* además de conocer la ruta, es posible conocer la IP de cada *router* por la cual los paquetes fueron dirigidos, así como el tiempo de respuesta de cada uno de los *routers* involucrados en el enrutamiento de los paquetes. La figura 5.21 muestra el resultado de ejecutar el mando *"tracert 192.168.5.2",* con la finalidad de conseguir el camino que los paquetes ICMP siguieron desde el *router* Victoria hasta el *router* ST. John's.

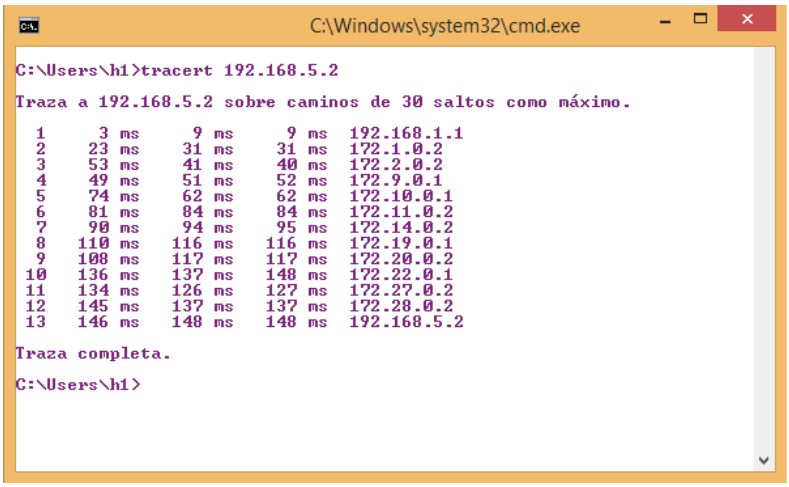

Figura 5.21: Ruta que siguieron los mensajes para alcanzar el nodo destino.

La figura 5.22 muestra de forma gráfica el camino obtenido mediante el mando *traceroute.* Se resalta en color azul dicho camino.

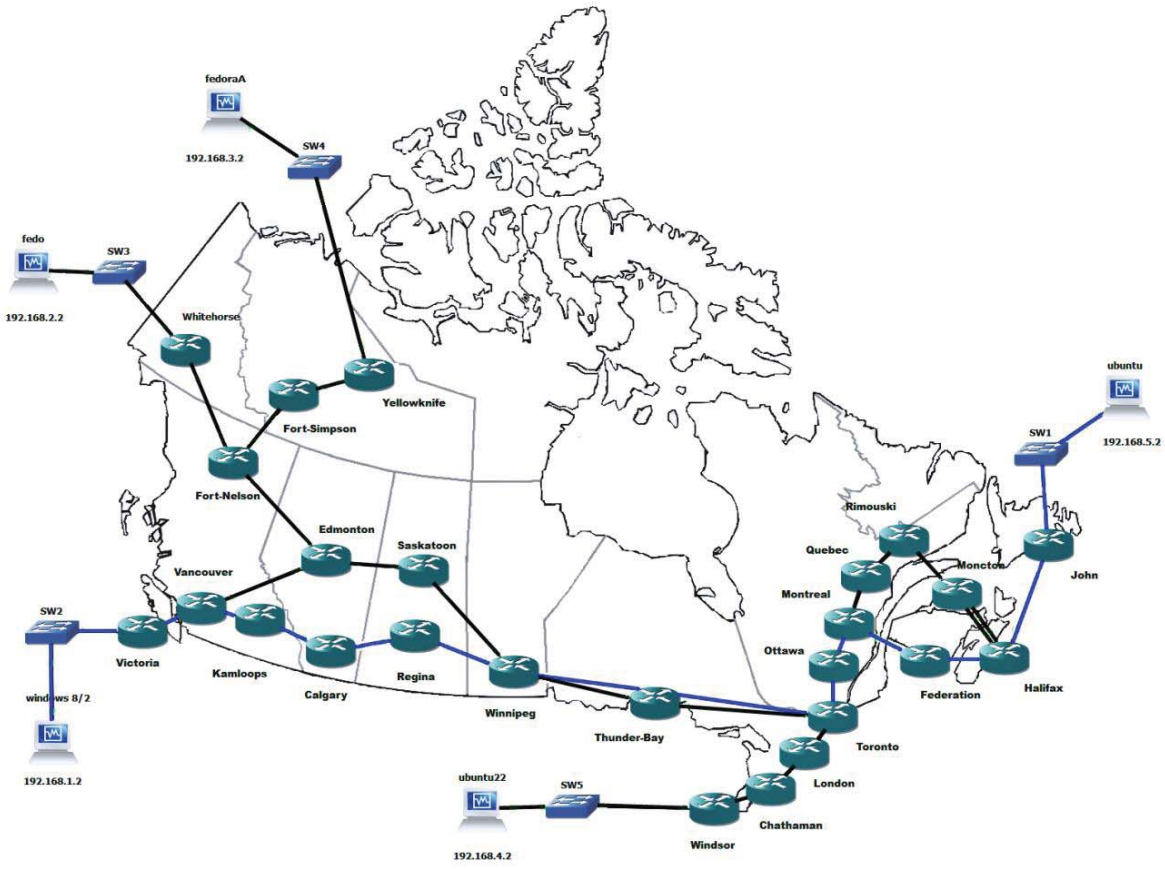

Figura 5.22: Ruta marcada en azul que siguen los paquetes para establecer la conexión entre los *routers* Victoria y ST. John's.

Para comprobar que se tiene comunicación desde ambos extremos del *backbone*, se envía un ping desde el *router* ST. John's hacia el *router* Victoria. En la figura 5.23 se muestra que, efectivamente existe conexión desde Ubuntu hacia el *backbone,* logrando establecer la conexión con Windows 8.

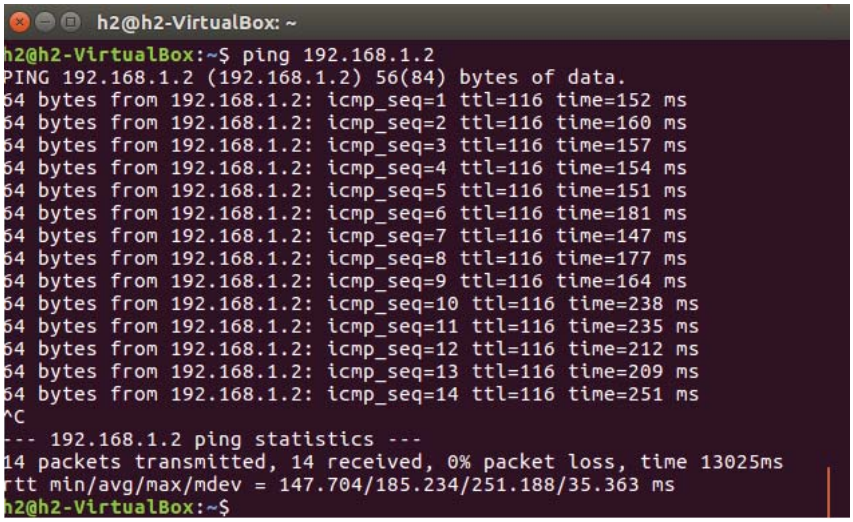

Figura 5.23: Conexión mediante ping desde ST. John´s a Victoria.

La figura 5.24 muestra una captura de los paquetes enviados entre los *routers* Victoria y ST. John's. El analizador de paquetes *Wireshark* se estableció en el enlace entre los *routers* Halifax y ST. John's. Además de poder ver los ping (*request* y *reply*) entre los nodos Victoria y ST. Jonh's, se logró obtener en la misma captura el intercambio de mensajes *Hello* entre las interfaces 172.28.0.1 y 172.28.0.2 pertenecientes a los *routers* Halifax y ST. John's, con la finalidad de mantener activa la adyacencia entre ambos *routers.* La adyacencia se establece a través de la IP 224.0.0.5 de *multicast*.

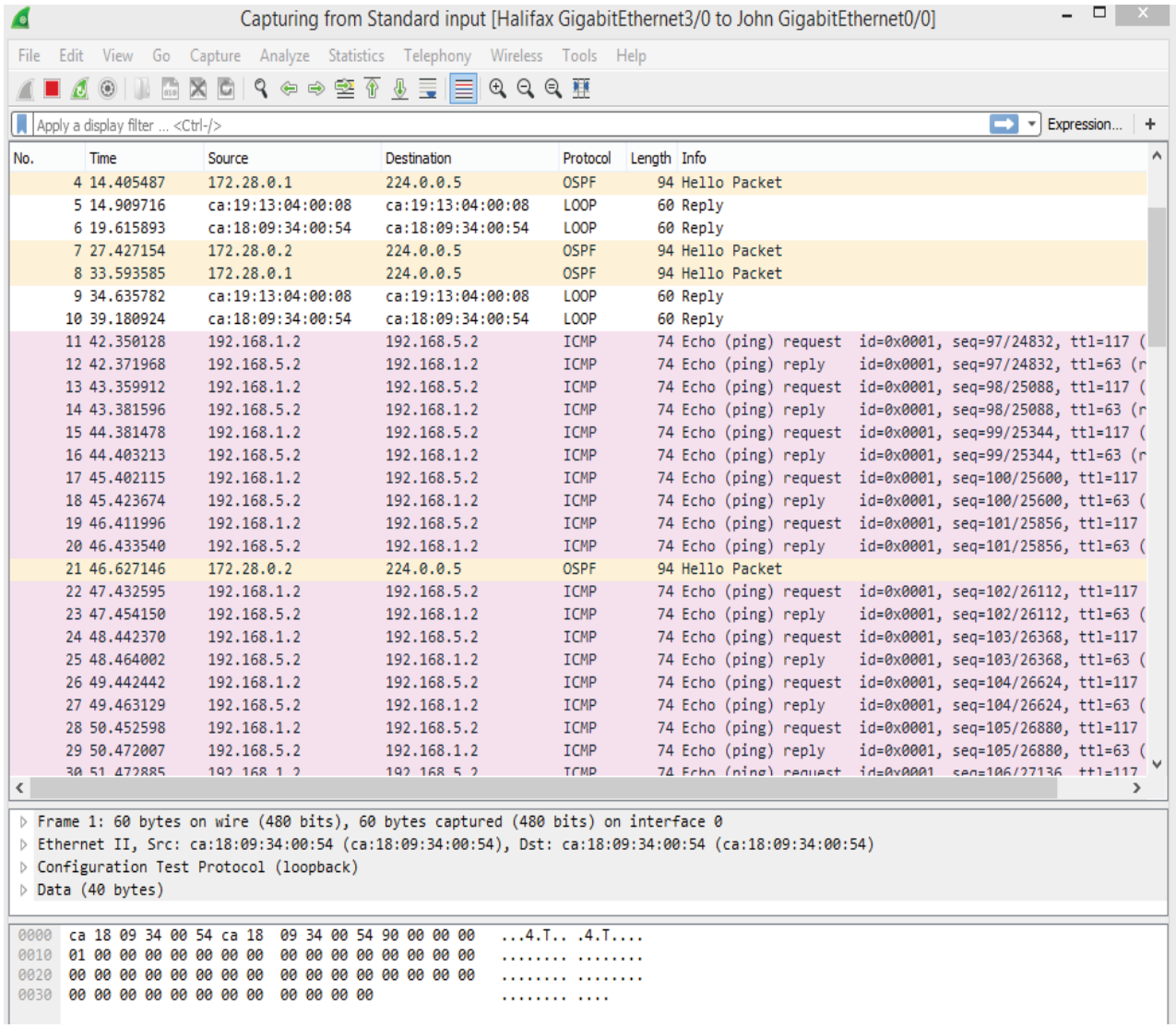

Figura 5.24: Captura de los paquetes ICMP mediante *Wireshark.*

Otra de las capturas logradas mediante *Wireshark* se presenta en la figura 5.25, la cual pertenece al intercambio de información de enrutamiento. La captura muestra el proceso de establecimiento de adyacencia entre los *routers* mediante el intercambio de mensajes de actualización del estado del enlace (LSU), solicitud de información del estado del enlace (LSR), sincronización de la base de datos (DBD) y la generación del acuse de recibo (*Aknowledge*).

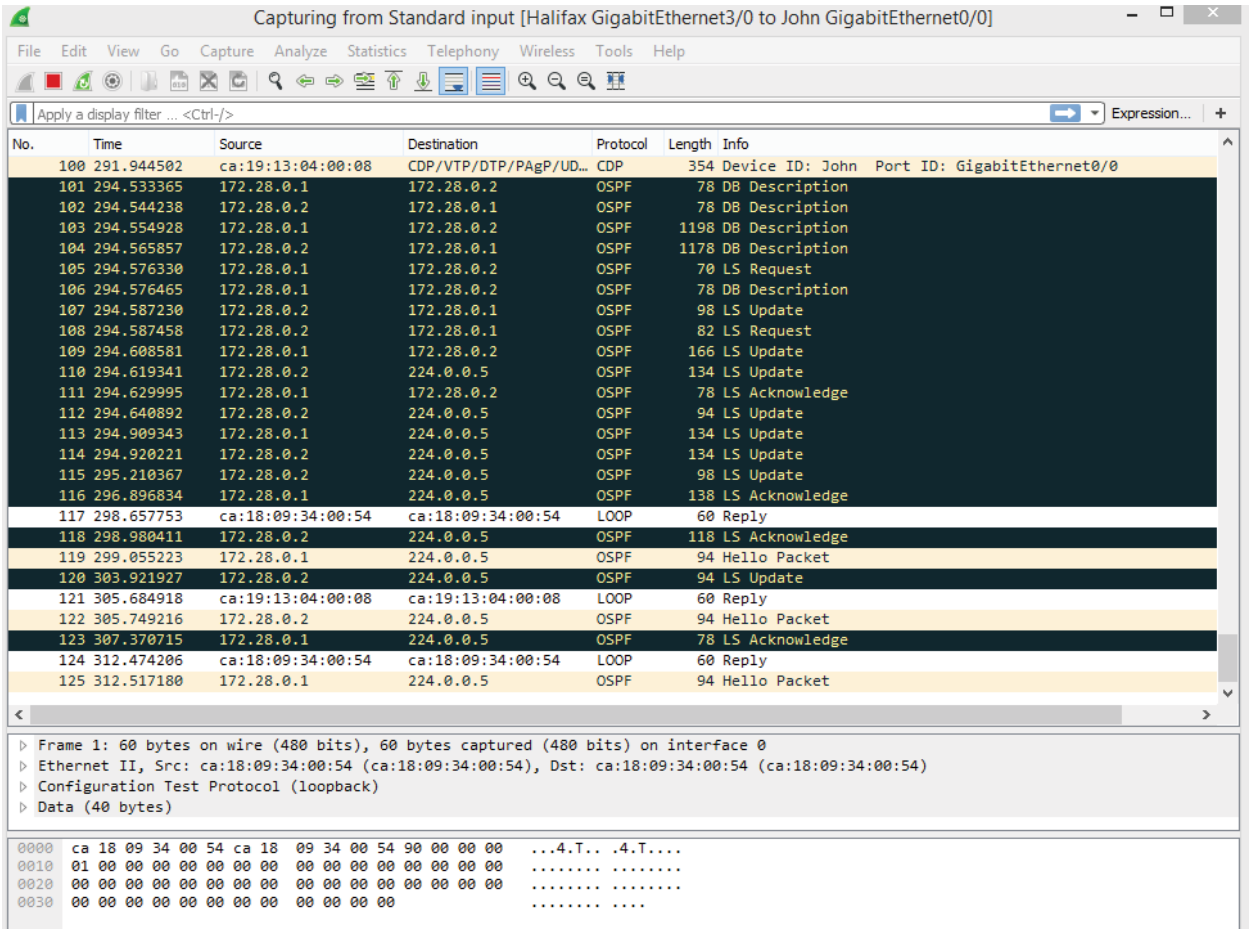

Figura 5.25: Captura de los paquetes generados por los *routers* ST. John's y Halifax para sincronizar sus tablas de enrutamiento.

Durante la ejecución del emulador se realizó el monitoreo del consumo de los recursos de memoria y CPU para analizar la estabilidad de GNS3, al trabajar con los elementos del *backbone* CANARIE propuestos en la figura 4.7. La tabla 5.3 muestra los resultados obtenidos.

Recursos	Computadora sin actividad	Emulador activo sin operación	Ejecutando pruebas de emulación
CPU	1% (1.18 GHz)	4% (1.28 GHz)	4% (1.28 GHZ)
Memoria	5% (1.6 GB de 32 GB)	37.81% (12.1 GB de 32 GB)	37.81% (12.1 GB de 32 GB)
Tiempo			TTL

Tabla 5.3: Recursos consumidos por GNS3.

Debido a que GNS3 utiliza el software de los equipos reales, le tomó un tiempo promedio de 9.50 minutos tener los 30 elementos que conforman el diseño del *backbone* CANARIE listos para realizar las pruebas de emulación de conectividad y gestión. Sin embargo, las pruebas de conectividad se ejecutaron de forma instantánea, es decir, su duración fue igual a lo que dura un TTL.

5.4 Resultados de emulación de gestión

La emulación de gestión se realizó a las mismas variables utilizadas en la simulación de gestión, propuestas en el capítulo 4. La finalidad es comprobar si es posible realizar monitoreo y configuración sobre dichas variables, dentro de un sistema real. Los resultados se muestran a continuación.

sysName

En la figura 5.26 se muestra el valor contenido en la variable *sysName.* El valor de la variable se obtiene mediante la operación *Get.*

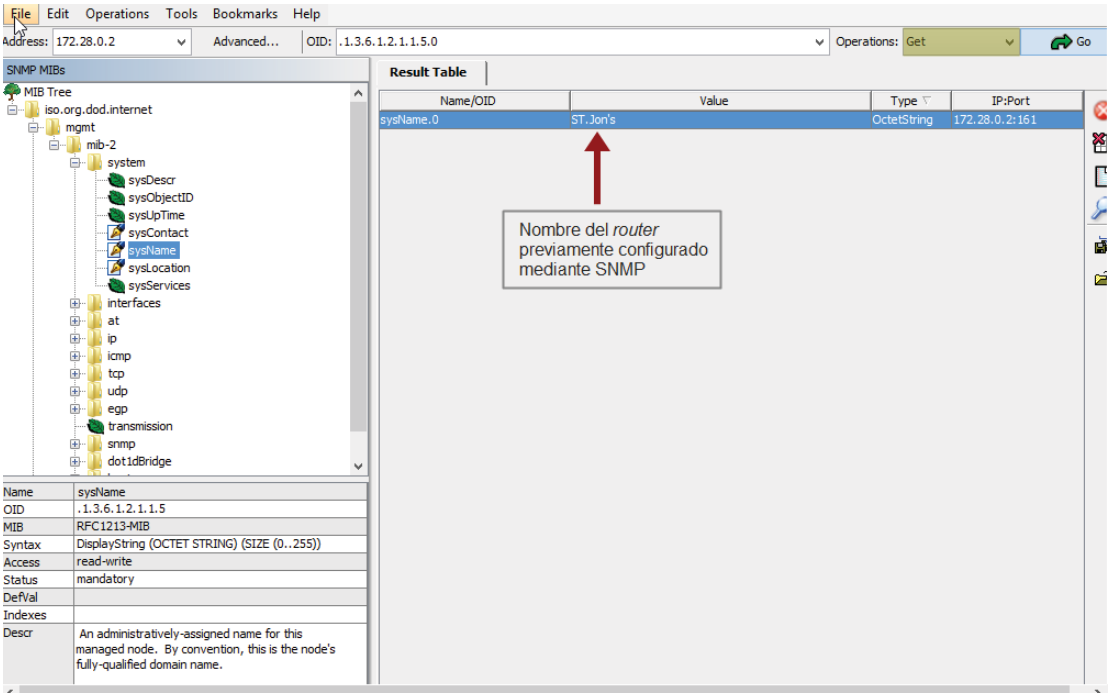

Figura 5.26: Monitoreo de la variable *sysName.*

En la figura 5.27 se muestra la configuración de la variable *sysName.* A través de la operación *Set* se cambia el valor a “Router_Jonny”.

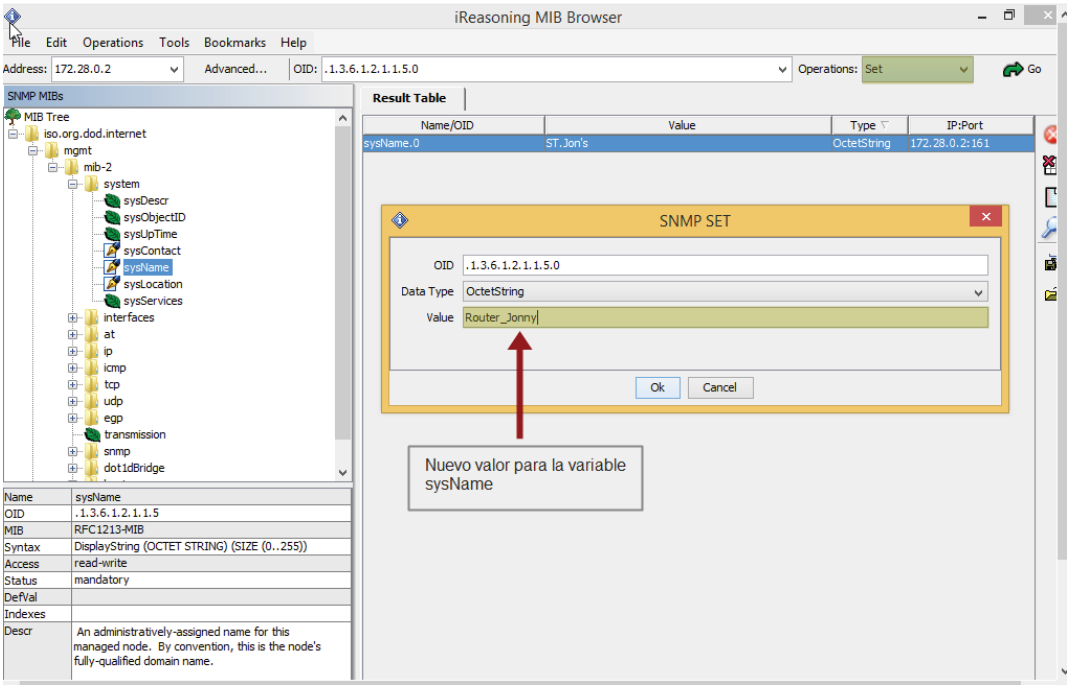

Figura 5.27: configuración de valor de la variable *sysName.*

Después de aceptar la nueva configuración, en la figura 5.28 se muestra el nuevo valor de la variable, obtenido mediante la operación *Get.*

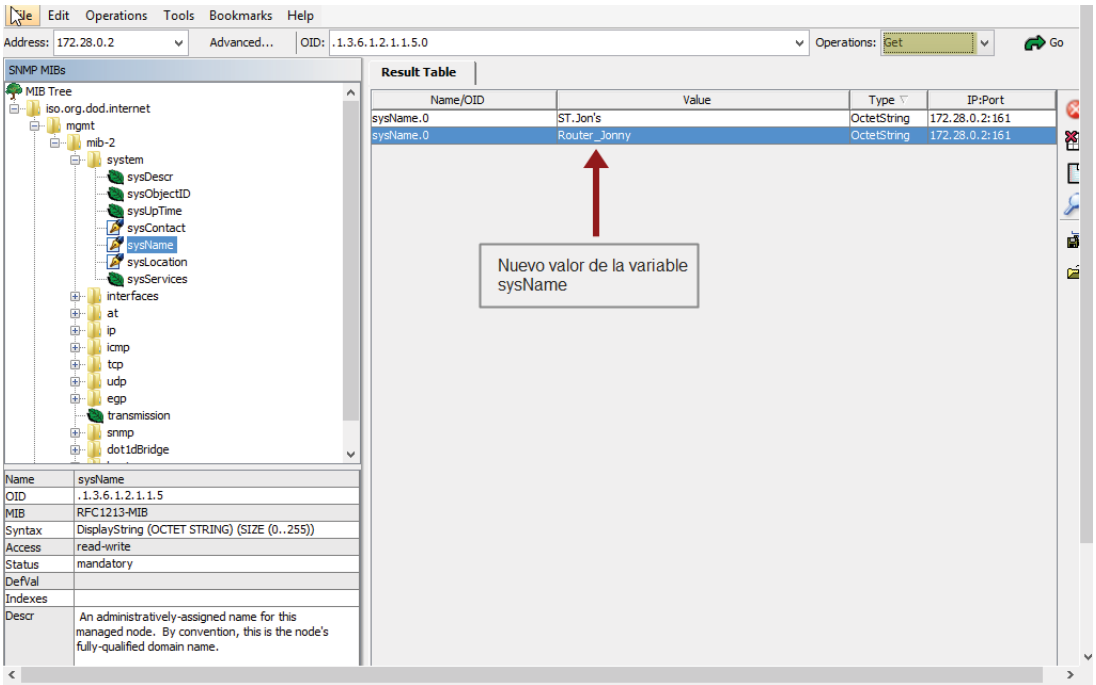

Figura 5.28: Cambio del valor de la variable *sysName.*

sysLocation

La figura 5.29 muestra el valor de la variable *sysName,* obtenida mediante la operación *Get.*

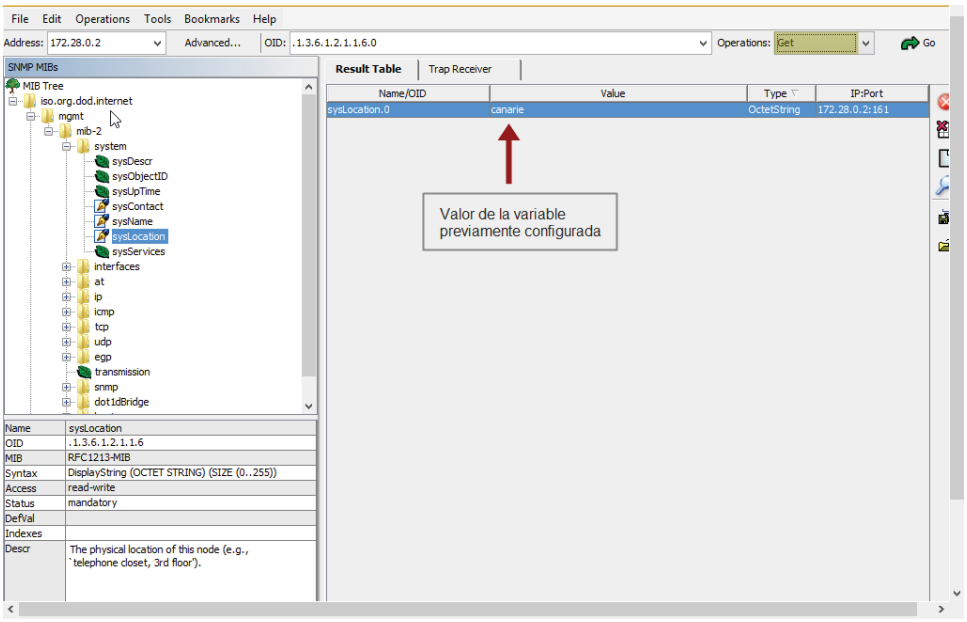

Figura 5.29: Monitoreo del valor de la variable *sysLocation.*

En la figura 5.30 se muestra la configuración de la variable para cambiar su valor a “ubicado_en_ST.John”.

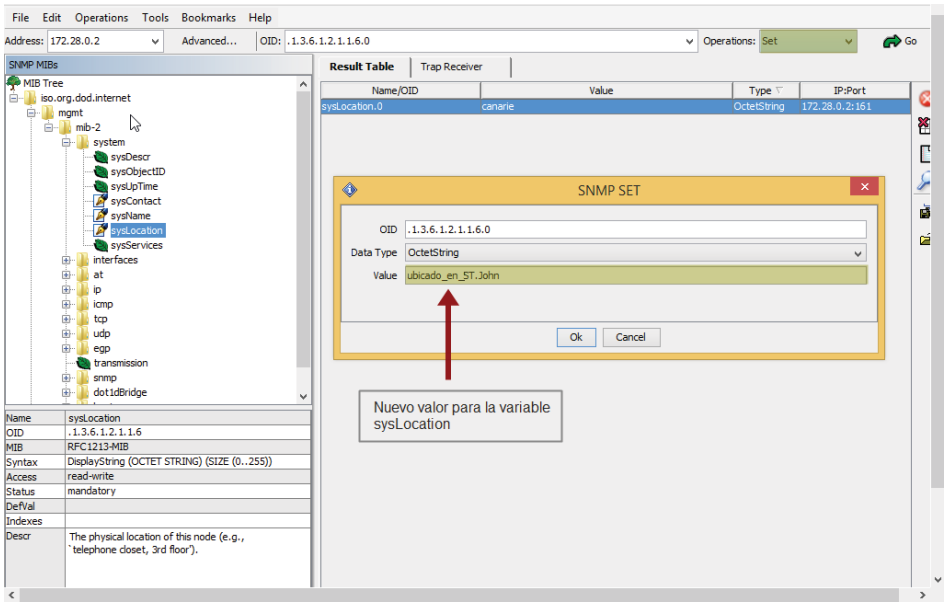

Figura 5.30: Configuración del valor de la variable *sysLocation.*

Después de aceptar los cambios, la figura 5.31 muestra el nuevo valor de la variable.

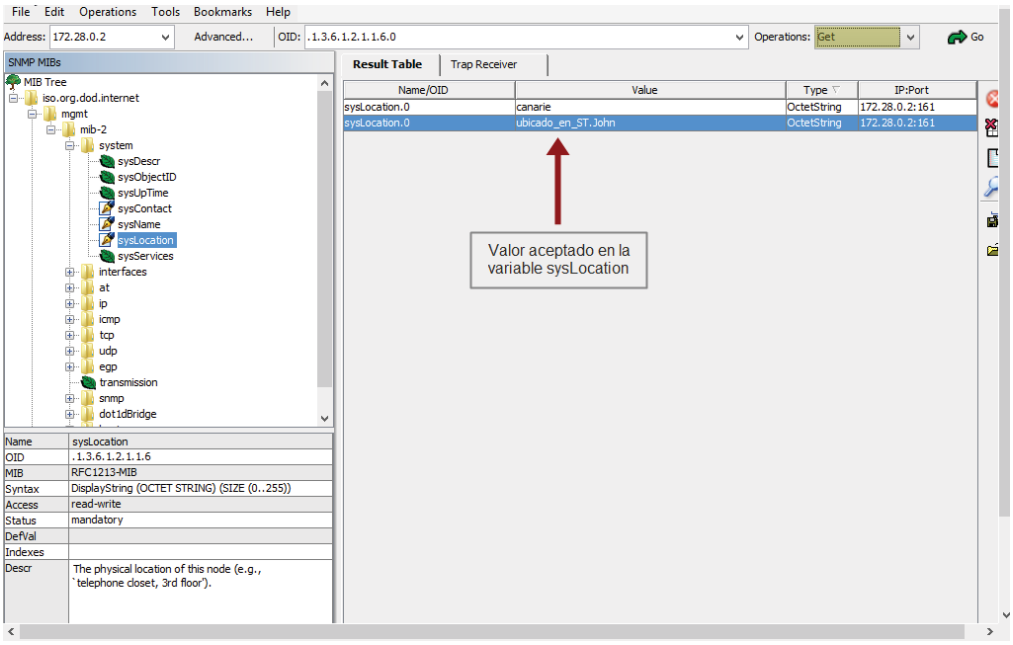

Figura 5.31: Cambio del valor de la variable *sysLocation.*

ifNumber

La figura 5.32 muestra los valores de la variable *ifNumber,* obtenidos mediante la operación *Get-Bulk.*

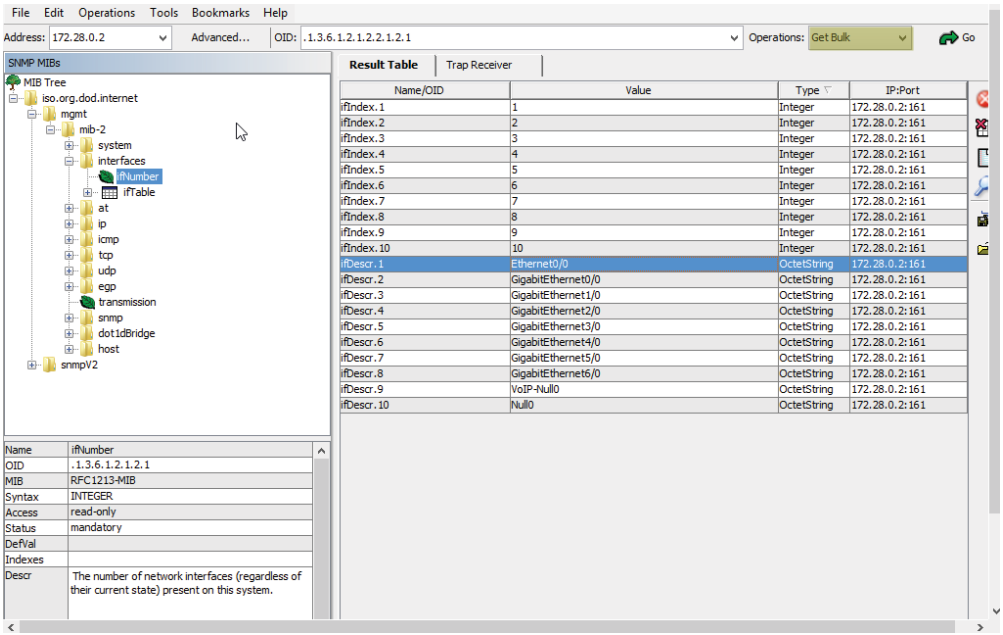

Figura 5.32: Monitoreo de la variable *ifNumber.*

Dentro de la información desplegada mediante la operación *Get-Bulk*, se elige el valor del campo *ifDescr.2* para configurar su valor. La figura 5.33 muestra esta operación.

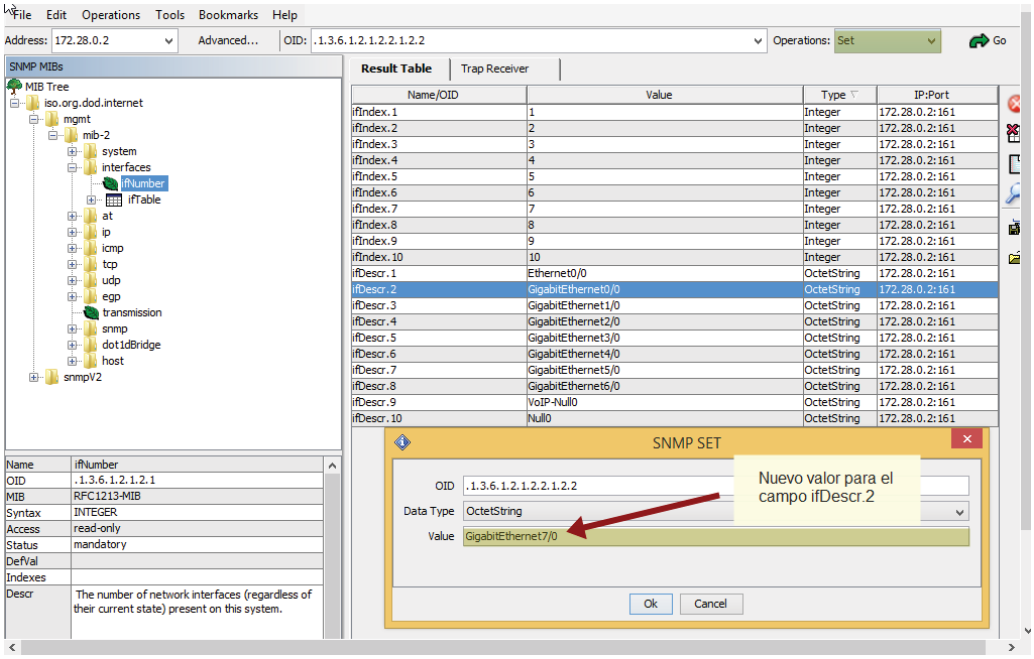

Figura 5.33: Configuración del campo *ifDescr.2* de la variable *ifNumber*

En la figura 5.34 se muestra el mensaje generado por el agente, el cual indica que no es posible configurar el valor de la variable.

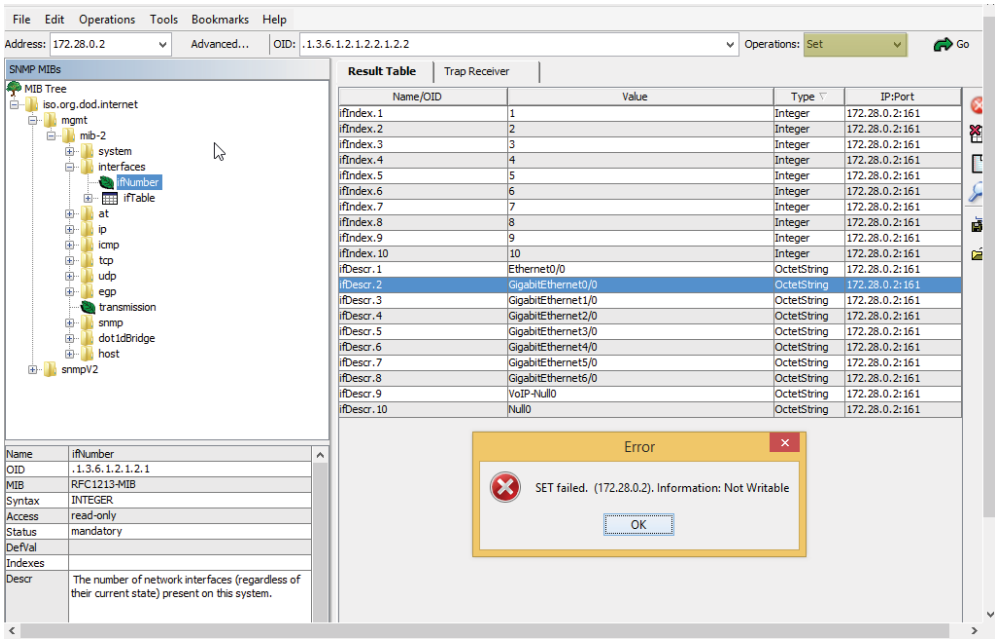

Figura 5.34: No fue posible realizar la configuración, la variable *ifNumber,* sólo se permite el monitoreo de sus elementos.

ipDefaultTTL

La figura 5.35 muestra parte de la información que almacena la variable ipDefaultTTL, obtenida mediante la operación *Get-Bulk.*

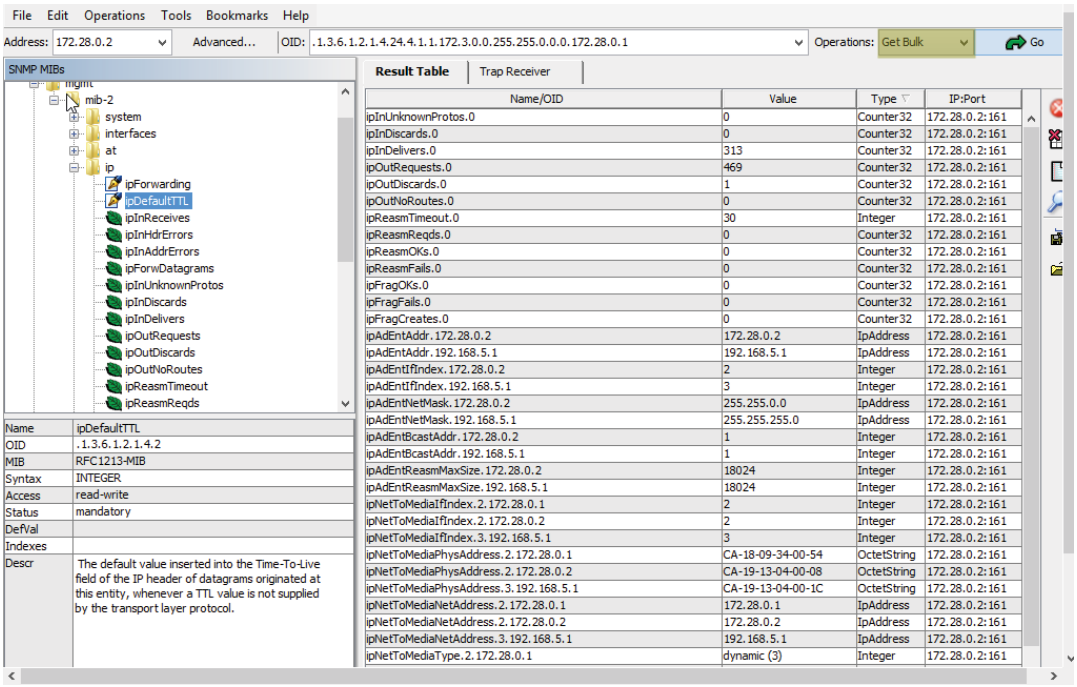

Figura 5.35: Monitoreo de la variable *ipDefaultTTL.*

La figura 5.36 muestra la configuración de uno de los campos de la variable *ipDefaultTTL*.

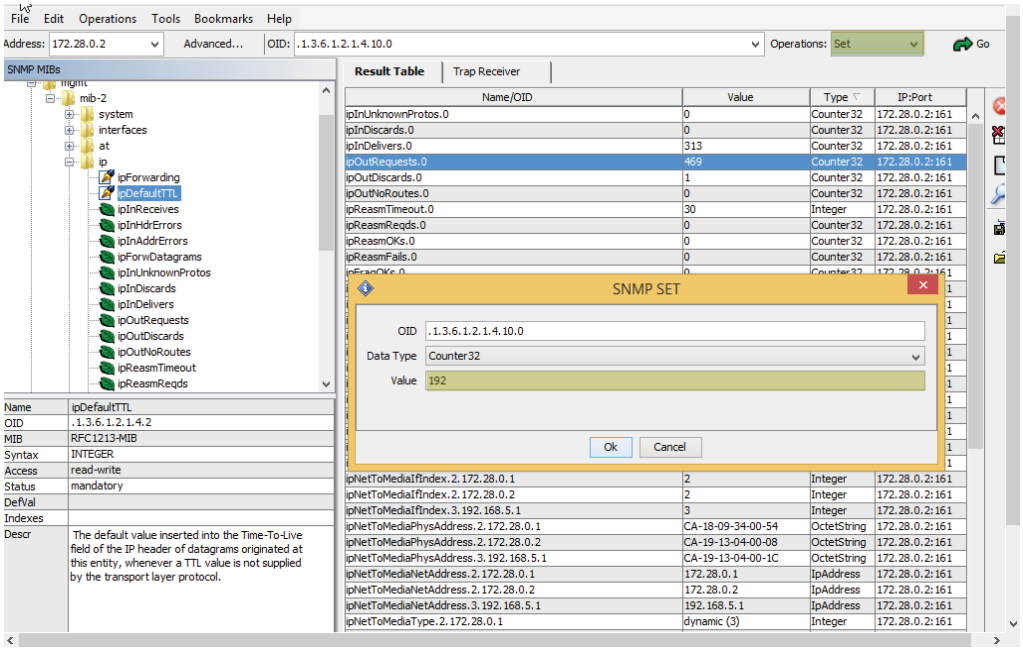

Figura 5.36: Configuración de un campo de la variable.

La figura 5.37 muestra un mensaje enviado por el gestor, indicando que no fue posible realizar la configuración del elemento seleccionado.

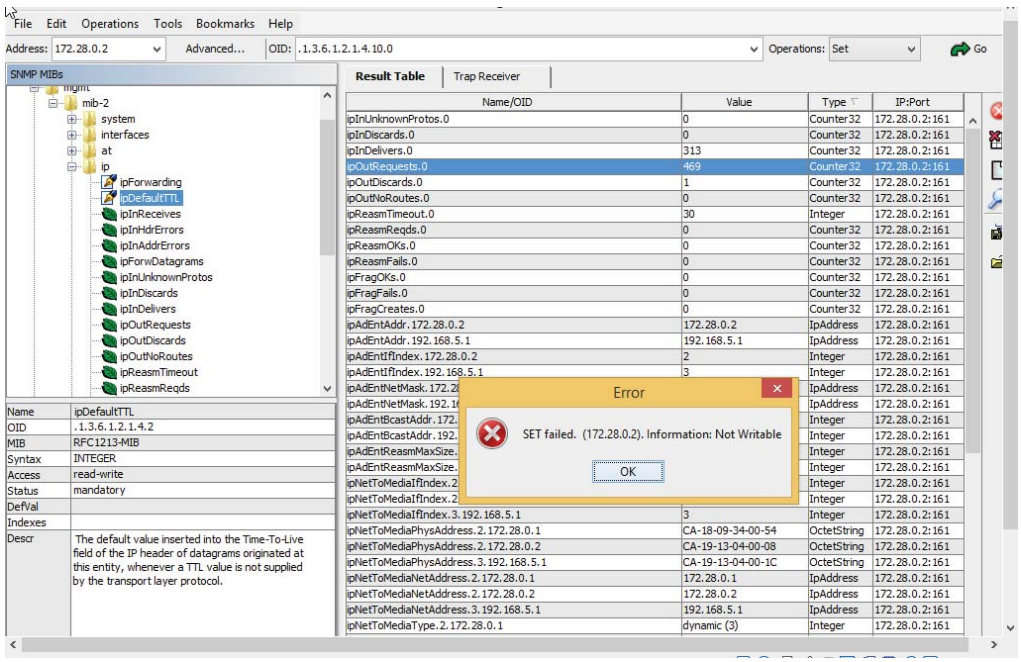

Figura 5.37: El cambio del valor del campo de la variable no se realizó.

Sin embargo, al aplicar la operación *Get* sobre la variable *ipDefaultTTL,* se obtiene el monitoreo de dicha variable tal como se muestra en la figura 5.38.

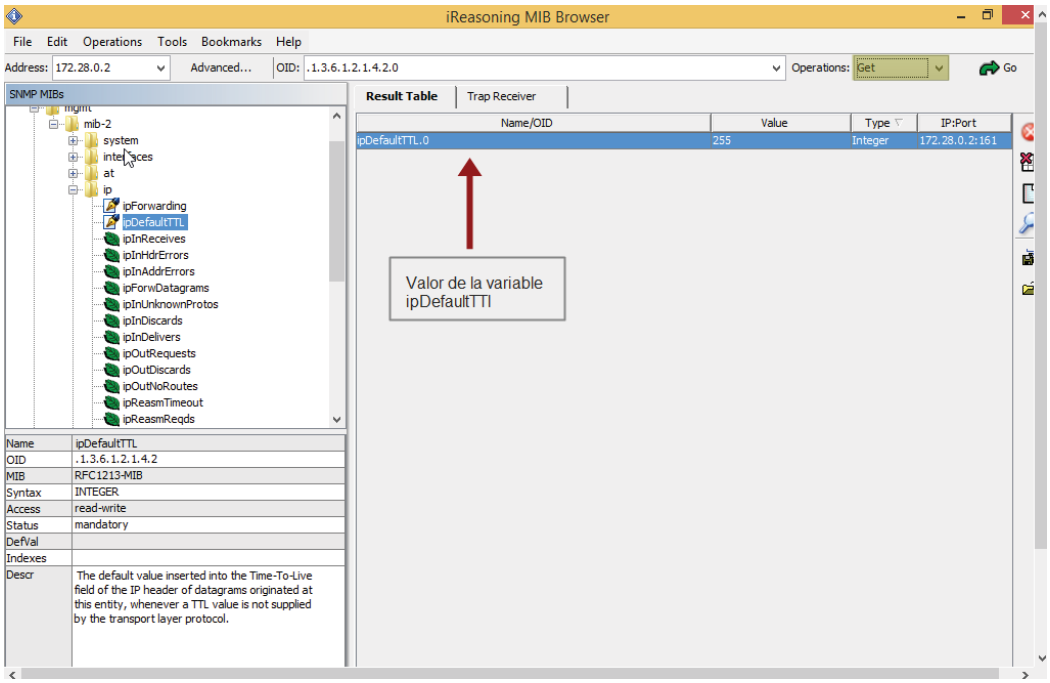

Figura 5.38: Monitoreo de la variable *ipDefaultTTL* utilizando la operación *get.*

La configuración de la nueva variable obtenida, se muestra en la figura 5.39

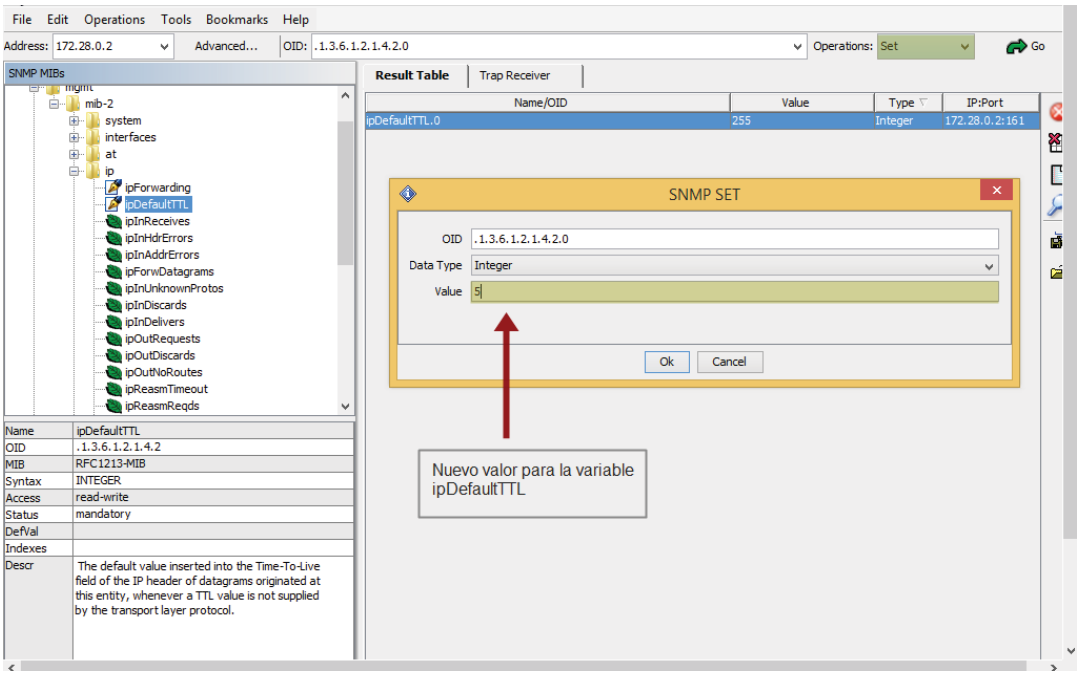

Figura 5.39: Configuración de la variable *ipDefaultTTL.*

Después de aceptar los cambios, el resultado de la configuración se indica en la figura 5.40.

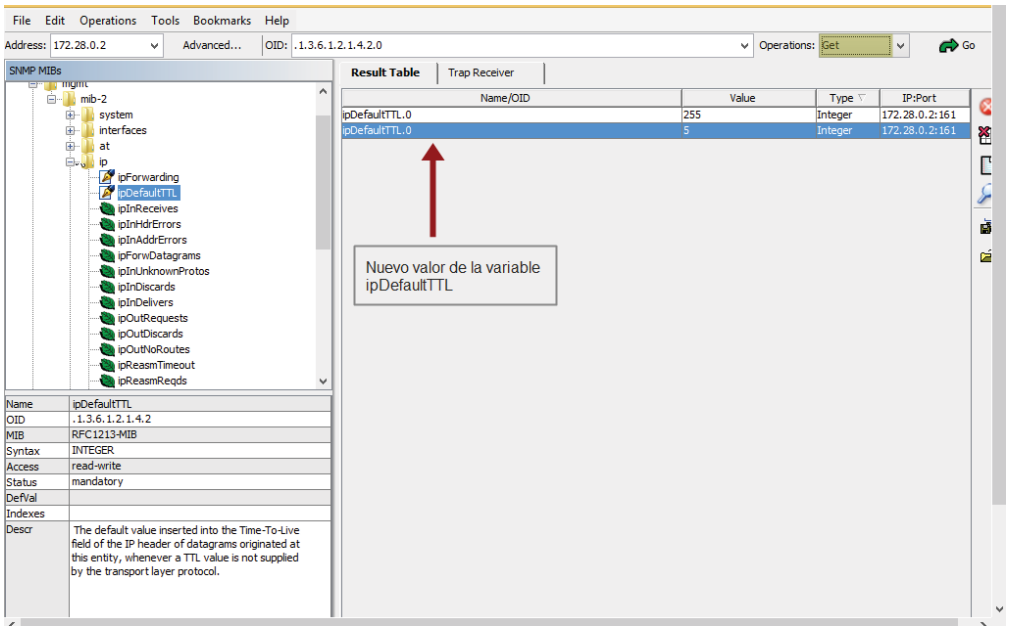

Figura 5.40: Cambio del valor de la variable *ipDefaultTTL*.

ipAddrTable

La figura 5.41 muestra parte de la información de la variable obtenida mediante la operación *Get-Bulk.*

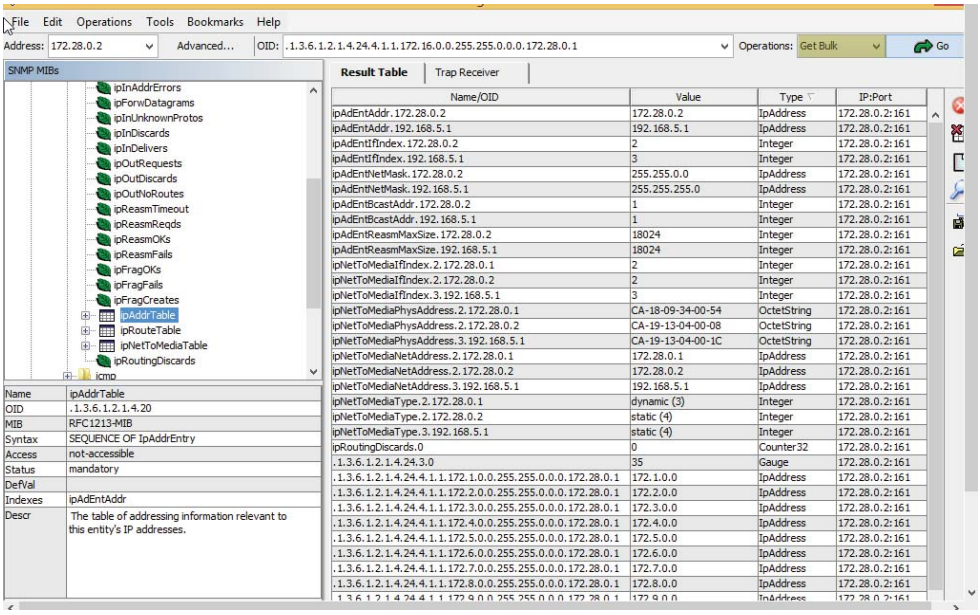

Figura 5.41: Monitoreo de la variable *ipAddTable.*

La configuración de una de las entradas de la variable *ipAddTable* se muestra en la figura 5.42

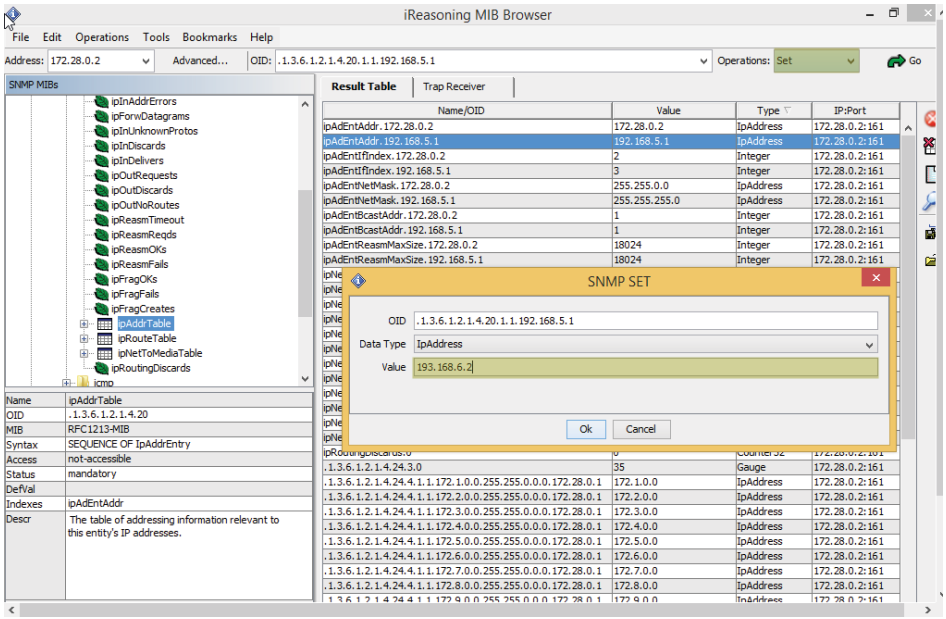

Figura 5.42: Configuración de una entrada de la variable *ipAddTable.*

Los resultados de la configuración se presentan en la figura 5.43.

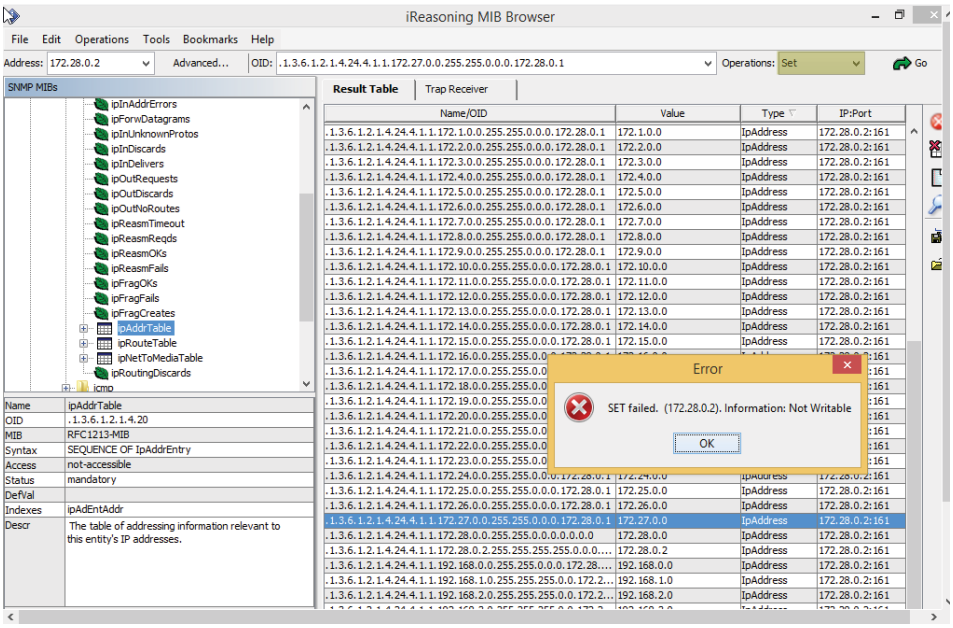

Figura 5.43: La variable *ipAddTable* solamente permite el monitoreo de la información.

La tabla 5.4 muestra el resumen de los resultados obtenidos al gestionar las variables mediante la emulación.

Variable	Monitoreo	Configuración	Estatus
sysName	✓	✓	Cumple con la teoría
sysLocation	✓	✓	Cumple con la teoría
ifNumber	✓	X	Cumple con la teoría
ipDefaultTTL	✓	✓	Cumple con la teoría
ipAddrTable	✓	X	Cumple con la teoría

Tabla 5.4: Resumen de la gestión realizada a las variables propuestas.

Con la finalidad de comprobar el ejemplo propuesto en la teoría de SNMP en el capítulo 3, se deshabilitó mediante línea de mandos la interfaz del *router* ST. John's para conseguir los *traps* en el gestor. La figura 5.44 muestra una serie de *traps* recibidos en el MIB *Browser* de donde se distingue el *trap* deseado.

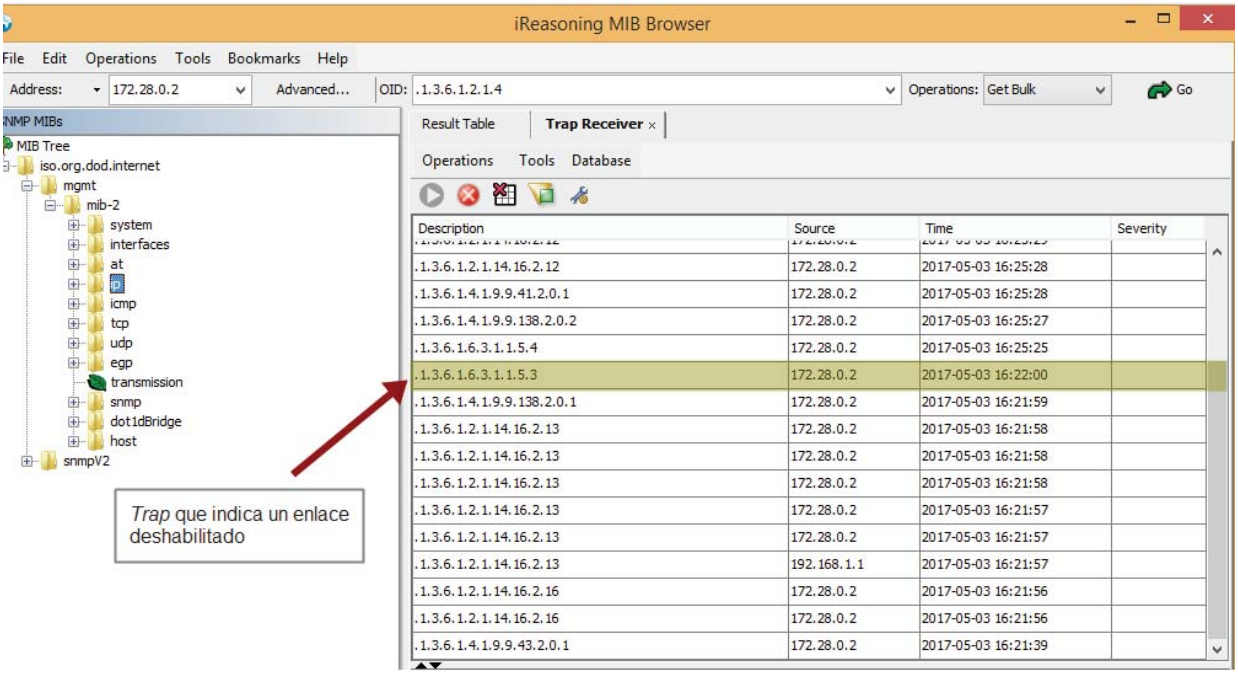

Figura 5.44: *Traps* recibidos en la estación de gestión generados por el nodo ST. John´s**.**

Para tener una buena interpretación de los *traps* recibidos, Cisco ofrece la herramienta en línea; *Cisco SNMP Object Navigator,* la cual después de autentificarse como usuario, permite traducir los OID en nombres de objetos o nombres de objetos en OID [103].

La deshabilitación de una interfaz genera el OID .1.3.6.1.6.3.1.1.5.3, como se especificó en el ejemplo del capítulo 3, y considerando que dicho OID apareció en el *trap* recibido en la figura 5.44, se tiene la traducción en la figura 5.45.

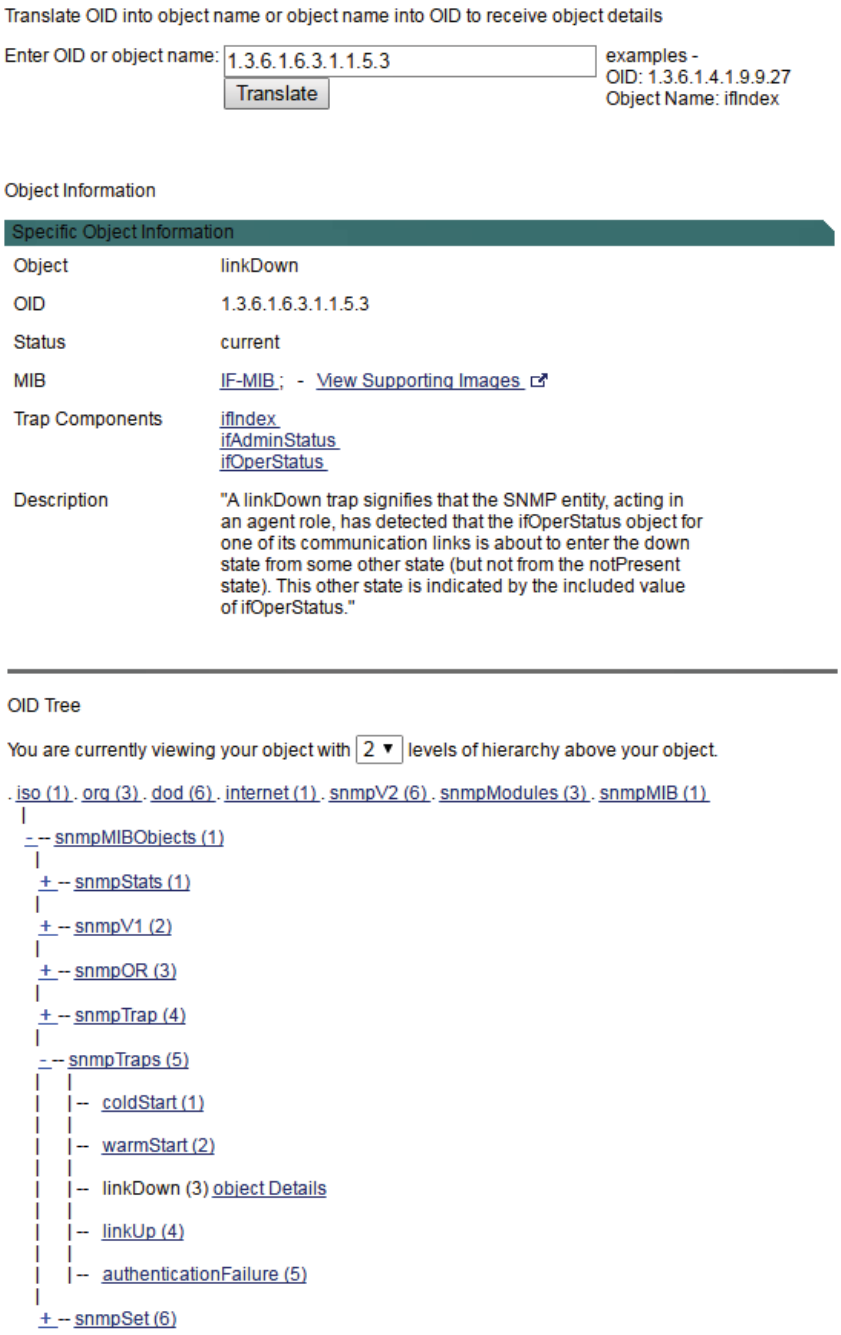

Figura 5.45: Interpretación de un OID recibido en un *trap,* utilizando la herramienta Cisco SNMP *Object Navigator*.

5.5 Conclusiones para la conectividad y gestión

5.5.1 Para la conectividad

Simulación en *Packet Tracer* V7.0.0.0305

- **Estabilidad:** El simulador *Packet Tracer* versión 7.0 se mantiene estable al trabajar con los 25 *routers* configurados mediante el protocolo de enrutamiento OSPF. Dadas las características de la computadora en la cual se desarrollaron las pruebas de simulación de conectividad, el simulador no presentó problema alguno (cierre inesperado, congelamiento, consumo excesivo de recursos de CPU y memoria) al ejecutar 35 elementos de red en total.
- **Herramientas de visualización:** La visualización gráfica del recorrido que realizan los mensajes ICMP enviados mediante ping, permiten generar una comprensión sólida del comportamiento de las redes de datos reales.
- **Enlaces:** Un enlace serial en *Packet Tracer* puede tener una capacidad máxima de 1,544 Mbps, el cual está muy por debajo a la utilizada por CANARIE en 2016, pero estas permiten construir una primera aproximación del funcionamiento real del *backbone.*

Emulación con GNS3

- **Mejor aproximación aunque mayor requerimiento**: La computadora con la que se simuló, no soportó la ejecución de la emulación de los elementos que conforman el diseño del *backbone* CANARIE, por lo cual, se cambió el equipo, tal como se indicó en el apartado de metodología. Las características del segundo equipo son: procesador Intel Xeon E5-2620 v2 @ 2.10 GHz y 32 GB de memoria RAM. Se logró tener una emulación de conectividad entre los routers casi de forma similar a la especificada por la teoría de CANARIE, al poder utilizar enlaces del orden de los Gigabits, así como el IOS de un *router* de *backbone*.
- **Estabilidad:** La ejecución del emulador requiere de un tiempo más prolongado para poner en marcha las máquinas virtuales. El tiempo requerido para esta acción varía, dependiendo de la cantidad de dispositivos utilizados,

sin embargo, la ejecución de GNS3 durante las pruebas de conectividad, se mantuvo estable, no se presentaron cierres del programa o congelamiento del sistema por el consumo de recursos excesivos por parte del emulador.

- **Enlaces:** Las interfaces utilizadas en los *routers* 7200 de Cisco son interfaces Gigabit Ethernet, las cuales proporcionan 1 Gbps de velocidad de conexión, si bien esta capacidad no es la utilizada por CANARIE, permite satisfactoriamente tener una segunda aproximación de la comprensión del funcionamiento del *backbone.*

5.5.2 Para la gestión

Gestión en *Packet Tracer,* V7

- **Capacidad de monitoreo:** El monitoreo de las variables propuestas se realizó para la mayoría de ellas. Por otra parte, el simulador no permite que los agentes generen traps, así como tampoco es posible tener un control de los host que fungen como gestores, ya que cualquier host en la red puede hacer el papel de gestor con los permisos de acceso correctos.
- **Límite del árbol de gestión:** El árbol de gestión utilizado por el MIB *Browser* en Packet Tracer versión 7, no cuenta con una amplia variedad de variables gestionables, tal como se mostró en la tabla 5.2. Esto se debe a que el simulador es una herramienta de estudio, por lo cual, únicamente se agregan las variables de mayor relevancia para el estudio de las certificaciones Cisco.

Gestión en GNS3, V1.5

Por su parte, GNS3 permite un monitoreo más completo que en el simulador, así com la posibilidad de realizar la configuración remota de las variables desde las unidades gestoras. El gestor, MIB y agentes presentan características muy superiores al simulador, debido a que se pueden instalar las librerías de variables específicas, en caso de que estas no vengan en la versión del MIB *Browser* utilizado.

Se logró monitorear las 5 variables propuestas, y realizar configuración sobre tres de ellas.

En la tabla 5.5 se muestran los resultados para la simulación y emulación de gestión.

	SIMULACIÓN Packet Tracer		EMULACIÓN GNS3	
Variable	**Monitoreo**	**configuración**	**Monitoreo**	**Configuración**
sysName	✓	✓	✓	✓
sysLocation	✓	✓	✓	✓
ifNumber	✓	X	✓	X
ipDefultTTL	-	-	✓	✓
ipAddTable	-	-	✓	X

Tabla 5.5: Comparativa de las variables gestionadas en la simulación vs la emulación

Además de poder realizar el monitoreo y configuración de algunas de las variables descritas en la metodología para la gestión, en GNS3 fue posible manejar el uso de *traps,* al realizar acciones como la deshabilitación y habilitación de interfaces dentro de la consola de comandos del agente gestionado, caso contrario para el simulador *Packet Tracer* V7.0 el cual no permitió el uso de los *traps.*

En general, el simulador permitió una primera aproximación didáctica del funcionamiento de la red CANARIE, mientras que el emulador GNS3 permitió un acercamiento a la realidad del funcionamiento de la red del *backbone* CANARIE, haciendo notorio el hecho de que a pesar de tener una plataforma que permite utilizar el *software* de los equipos reales, estos utilizan interfaces de red con capacidades de conexión muy por debajo a los utilizados por la red avanzada CANARIE.

Siguiendo los objetivos planteados al inicio de la tesis, se concluye que estos fueron logrados, al poner a prueba las herramientas de simulación y emulación con las cuales se realizó el análisis de funcionamiento del *backbone* CANARIE, comprobando de esta forma, la eficiencia de cada una de ellas y, exponiendo sus

limitantes a través de los elementos utilizados para llevar a cabo la simulación y emulación de conectividad y gestión.

5.6 Conclusiones no técnicas del trabajo de tesis

Este trabajo forma parte del conjunto de proyectos de ADVNETLAB, aportando una pieza en un rompecabezas mayor, en este caso se provee simulación y emulación de la conectividad y gestión de la red avanzada de Canadá, CANARIE, para la infraestructura de *backbone* más reciente.

Algunas de las habilidades que desarrollé durante la construcción de este proyecto son:

- La habilidad de gestionar proyectos apegados a una planeación de actividades, las cuales se programan previamente y son proyectadas a terminarse en la fecha establecida.
- El uso de fuentes de información confiables, es decir, fuentes de información propias de la organización de la cual se está realizando la investigación, como lo es el caso de CANARIE cuya fuente de información más confiable radica en su sitio web. Así como también, la consulta a libros de texto que contengan los temas requeridos.
- La habilidad adquirida para utilizar el emulador GNS3 con el cual poder emular redes de datos antes de que sean llevadas a la implementación.
- Adquirir la habilidad de poder interpretar los datos obtenidos durante el desarrollo de proyectos para generar conclusiones coherentes que permitan validar la aceptación del mismo.

Trabajo futuro

Durante el desarrollo de la presente tesis, se generó un *paper* relacionado al análisis del funcionamiento del *backbone* CANARIE, el cual se encuentra siendo arbitrado por expertos internacionales al momento de que esta tesis se está imprimiendo. Se espera la resolución de los árbitros para que den su aprobación para que dicho artículo sea publicado en una revista internacional de IEEE.

Como parte del estudio de las redes avanzadas, el laboratorio ADVNETLAB continuará con el análisis de dichas redes a nivel mundial, hasta lograr realizar un análisis que conecte a los cinco continentes, a través de sus redes de mayor jerarquía. Muy probablemente, el estudio de redes académicas continúe hacia la nueva tendencia tecnológica, la cual consistirá en interconectar planetas.

Apéndice A

Índice de acrónimos

ACK – *Aknowledge*

Carácter o secuencia de caracteres enviados por el receptor de un mensaje para notificarle al emisor, qué el último mensaje fue recibido con éxito.

Adyacencia

Relación que se forma entre *routers* vecinos, con la finalidad de realizar intercambio de información de enrutamiento.

Algoritmo

Es la parte del software en la capa tres del modelo ISO/OSI que se encarga de encontrar el mejor camino por el cual se transmitirá el mensaje hasta una red destino.

Ancho de banda

Es el rango (frecuencia comprendida entre dos límites) de las frecuencias que se pueden pasar por un canal de comunicación. Se expresa en términos de la diferencia entre el límite de la frecuencia alta y el límite de la frecuencia baja.

APNIC – *Asia Pacific Network Information Centre*

Centro de Información de las Redes de Asía Pacífico, es una Organización sin fines de lucro, cuya función es administrar los recursos de números de Internet en las regiones de Asía y el Pacífico.

ARIN – *American Registry of Internet Numbers*

Registro Americano de Números de Internet, es una Organización en Estados Unidos que gestiona las direcciones IP del país.

ARPA – *Advanced Research Proyects Agency*

Agencia de Investigación de Proyectos Avanzados. Agencia creada por el Departamento de Defensa de los Estados Unidos en 1958. Conocida como DARPA desde 1996, se caracteriza por el desarrollo de ARPANET, precursora del Internet actual.

ARPANET

Red desarrollada por ARPA, la cual utiliza técnicas de conmutación de mensajes. Fue la red precursora del Internet actual.

ATM – *Asynchronous Transfer Mode*

Modo de transferencia asíncrona, es la tecnología de conmutación y multiplexado de alta velocidad, utilizada para transmitir diferentes tipos de tráfico como voz, video y datos.

Backbone

Es el enlace principal que provee conectividad a diversas redes a través de un país, continente u océano.

CIX – *Commercial Internet Exchange*

Intercambio de Internet Comercial, es un nodo de intercambio de tráfico de organizaciones con fines lucrativos.

GHz

Giga Hertz; Es un múltiplo de unidad de medida de frecuencia Hercio (Hz) y equivale a 10ˆ9 (1, 000, 000, 000).

GigaPoP

Es un punto de acceso a la red, definido por la Internet2, el cual admite conexiones del orden de los Gigabits.

GNS3

Es un programa que provee una plataforma virtual para ejecutar sistemas operativos que han sido construidos para ejecutarse en equipos de red.

IMP – *Interface Message Processor*

Máquina encargada del intercambio de paquetes en ARPANET. Tiene como tarea conectar los *routers* entre sí, dirigir los mensajes a su destino y confirmar su llegada a dicho destino.

Interface

Provee los medios para la interconexión de equipos de red.

K-12

Es una designación que refiere a la educación primaria y secundaria. Es utilizada en países como Estados Unidos, Canadá, Turquía, Las Filipinas y Australia.

MAN – *Metropolitan Area Networks*

Red de Área Metropolitana, es una red de comunicaciones que cubre una extensión grande de una ciudad o campo, mediante la cual dos o más LANs se conectan.

Mbps

Es una unidad utilizada para cuantificar una transferencia de datos equivalente a 1,000 Kbps.

MODEM – Modulador/Demodulador

Es un dispositivo que convierte las señales de datos digitales y binarios a una señal compatible con el medio utilizado.

NCP – *Network Control Protocol*

Protocolo de Control de Red. Fue el primer protocolo *host-to-host* utilizado en ARPANET de 1970 a 1983.

NSFNET – *National Science Fundation Network*

Red de la Fundación Nacional para la Ciencia, creada por el gobierno de los Estados Unidos con enfoque a la comunicación, investigación y educación.

OC-192:

Es un enlace de red en fibra óptica con velocidad de transmisión de hasta 9953,28 Mbps. La estandarización establece una velocidad de 10 Gbps.

OSI – *Open System Interconnection*

Conocido como el Modelo de Referencia OSI o el Modelo OSI. Es un estándar de ISO que implementa los protocolos de red dentro de 7 diferentes capas: física, enlace, red, transporte, sesión, presentación y aplicación.

OSPF – *Open Shortest Path First*

Primero la ruta más corta, es el protocolo de enrutamiento interior basado en el estado del enlace. Es un protocolo abierto, por lo que puede ser utilizado en diversos *routers* de diferentes fabricantes.

Packet Tracer

Es un programa de simulación de redes que permite a quien lo usa experimentar y analizar el comportamiento de las redes.

RFC – *Request for Comments*

Medio en el que se publican propuestas sobre Internet. Cada RFC está caracterizado por un número previamente asignado.

RIP – *Routing Information Protocol*

Protocolo de Información de Enrutamiento, es uno de los primeros protocolos estandarizados. Se clasifica como protocolo vector distancia y utiliza el conteo de saltos como métrica.

ROADM – *Reconfigurable Optical Add-drop Multiplexer*

Multiplexor Óptico de añadir y desechar Reconfigurable; es un sistema óptico que permite realizar configuraciones remotas a un *router* en una red IP.

SDN – *Software Defined Networking*

Red Definida por Software; son un conjunto de técnicas, cuyo objetivo es facilitar la implementación de servicios de red de forma determinística, dinámica y escalable.

SONET – *Synchronous Optical Network*

Red Óptica Sincrónica; es un estándar del Instituto Americano de Estándares Nacionales (ANSI) de alta velocidad para transmisión de alta velocidad en fibra óptica.

SNMP – *Simple Network Management Protocol*

Protocolo de Administración de Red; es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre los distintos equipos de la red.

TCP/IP *Transmission Control Protocol/Internet Protocol*

Protocolo de Control de Transmisión/Protocolo de Internet. Es la familia de protocolos que se emplean en las comunicaciones de Internet.

UDP – *User Datagram Protocol*

Protocolo de Datagrama de Usuario, protocolo de servidor a servidor que permite a un programa de aplicación de una computadora enviar datagramas a otra computadora vía red de comunicaciones.

X.25

Recomendación de la UIT publicada en 1976, describe la interfaz entre la terminal de datos de usuarios y el equipo de comunicación.

Apéndice B

Configuración básica del iReasoning MIB *Browser*

Los dispositivos de red como *switches* y *routers* son fabricados con la aplicación de gestión conocida como agente, sin embargo, para poder interactuar con dicho agente, es necesario utilizar la aplicación que funciona como gestor. Para ello se instala en la máquina virtual Windows 8 el programa iReasoning MIB *Browser*, herramienta utilizada por ingenieros para administrar redes de datos mediante el protocolo SNMP. La aplicación se encuentra disponible para diversos sistemas operativos, en la página de iReasoning Networks (www.ireasoning.com/download.shtml).

Una vez instalada la aplicación, en la figura B.1 se muestra la ventana principal de la aplicación.

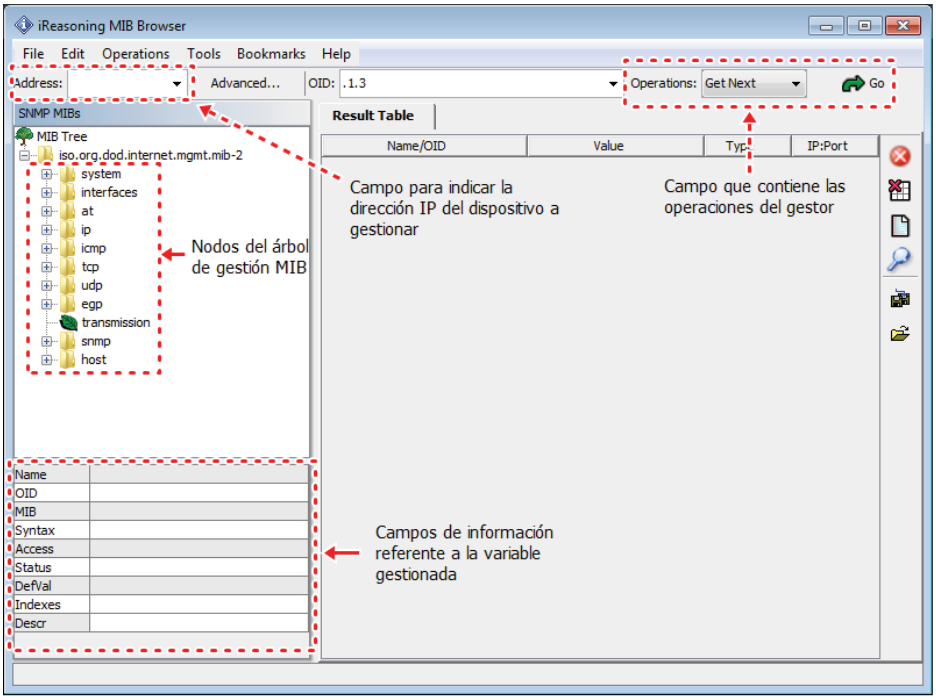

Figura B.1: Ventana principal de la aplicación de gestión.

Para poder acceder a la información de gestión de cualquiera de los dispositivos que se desea administrar se siguen los siguientes pasos.

1. En la casilla "A*ddress"* se coloca la dirección de IP de la interfaz por la cual se tendrá acceso al dispositivo gestionado, después, en el campo marcado como "*Advanced",* se asignan los permisos de acceso (comunidad) y se selecciona la versión del protocolo a utilizar, tal como se indica en la figura B.2.

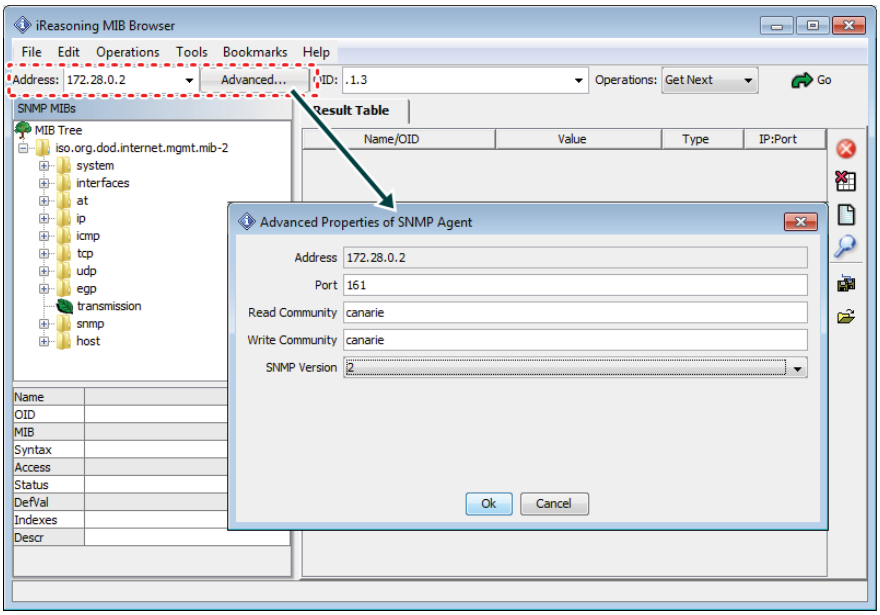

Figura B.2: Indicación de la IP del dispositivo a gestionar y permisos de acceso a la MIB.

2. Una vez configurados los permisos de comunidad, se despliega el árbol de gestión para conocer la información almacenada en cada variable. La figura B.3 muestra la selección de la variable *sysName*. En la parte superior se muestra el OID correspondiente a cada variable seleccionada.

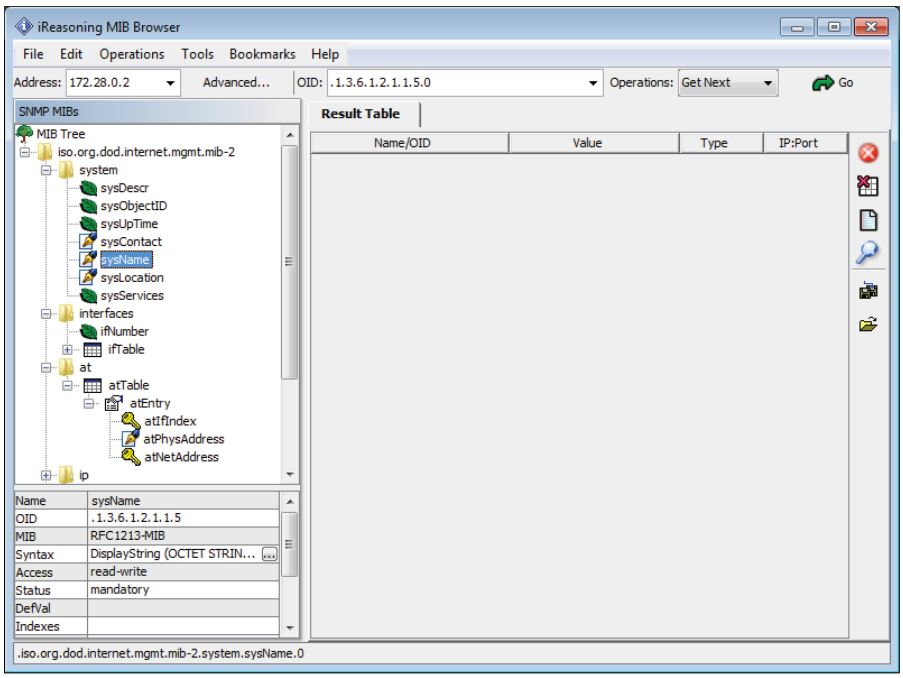

Figura B.3: Selección de la variable sysName.

3. Para consultar o modificar los valores almacenados en las variables, se selecciona una de las operaciones que el gestor utiliza para dicho propósito. La figura B.4 muestra dichas operaciones.

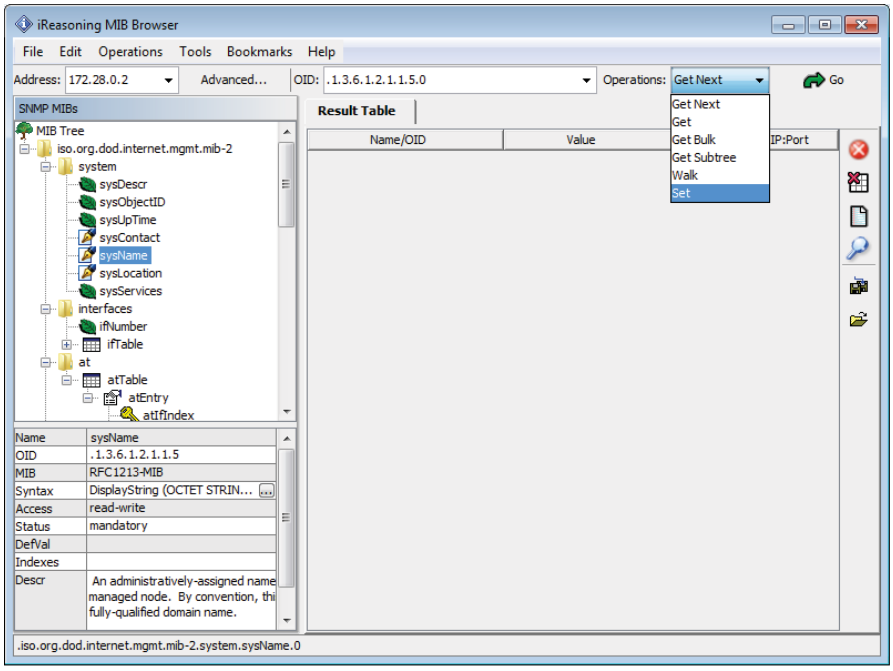

Figura B.4: Operaciones soportadas por el MIB *Browser.*

El valor de la variable se muestra en pantalla después de seleccionar la casilla "*Go".* De esta forma se pueden realizar acciones sobre el equipo gestionado, siempre y cuando lo permitan los permisos de acceso a la información (comunidad).

4. La figura B.5 muestra la opción para obtener los *traps* generados por los dispositivos gestionados cuando estos notifican diversos sucesos.

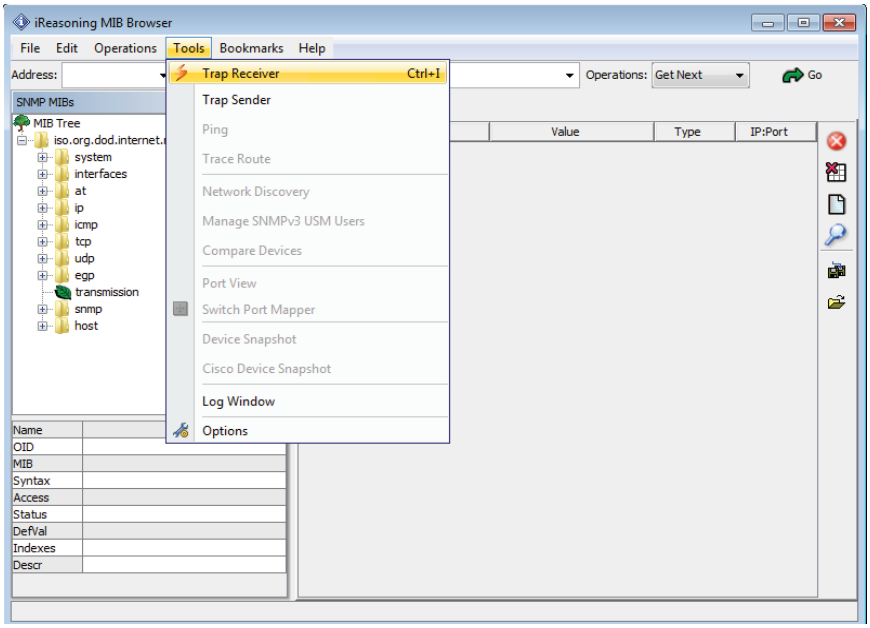

Figura B.5: Selección de la captura de *traps*

Apéndice C

Configuración del *router* Vancouver en *Packet Tracer* y GNS3 utilizando los protocolos OSPF y SNMP

Configuración del *router* con el protocolo OSPF

El *router* Vancouver cuenta con tres interfaces conectadas a los *routers* Victoria, Edmonton y Kamloops. El proceso de configuración consiste en asignar una dirección de red IP a cada una de las interfaces y después, configurar el protocolo OSPF para que se realicen las pruebas de conectividad. De los 25 *routers* que conforman el diseño del *backbone* CANARIE utilizado para las pruebas de simulación y emulación, se sustenta la configuración realizada al *router* Vancouver, la cual se describe a continuación. El proceso de configuración es similar para el resto de *routers,* la diferencia de configuración radica en las direcciones de interface de red asignadas a cada *router,* tal como se especificó en la tabla 4.1.

Configuración de la interfaz que conecta con el *router* Victoria

Vancouver> enable
Vancouver# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Vancouver (config)# interface serial 1/0
Vancouver (config-if)# ip address 172.1.0.2 255.255.0.0
Vancouver (config-if)# clock rate 64000
Vancouver (config-if)# no shutdown
%LINK-5-CHANGED: Interface Serial1/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up
Vancouver #
%SYS-5-CONFIG_I: Configured from console by console

Configuración de la interfaz que conecta con el *router* Edmonton

Vancouver > enable
Vancouver # configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Vancouver (config)# interface serial 1/1
Vancouver (config-if)# ip address 172.2.0.1 255.255.0.0
Vancouver (config-if)# clock rate 64000
This command applies only to DCE interfaces
Vancouver (config-if)# no shutdown
Vancouver (config-if)#

%LINK-5-CHANGED: Interface Serial1/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/1, changed state to up

Configuración de la interfaz que conecta con el *router* Kamloops.

Vancouver > enable
Vancouver Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Vancouver (config)# interface serial 1/2
Vancouver (config-if)# ip address 172.3.0.1 255.255.0.0
Vancouver (config-if)# clock rate 64000
This command applies only to DCE interfaces
Vancouver (config-if)# no shutdown
Vancouver (config-if)#
%LINK-5-CHANGED: Interface Serial1/2, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/2, changed state to up

Después de configuradas cada una de las interfaces con su respectiva dirección de red IP, se procede a configurar el protocolo OSPF.

Vancouver #configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Vancouver (config)# router ospf 1
Vancouver (config-router)# router-id 2.2.2.2
Vancouver (config-router)# network 172.1.0.0 0.0.255.255 area 0
Vancouver (config-router)# network 172.2.0.0 0.0.255.255 area 0
Vancouver (config-router)# network 172.3.0.0 0.0.255.255 area 0
Vancouver (config-router)#^Z
Vancouver #
%SYS-5-CONFIG_I: Configured from console by console

Con la configuración anterior, el *router* está listo para comenzar a construir las tablas de enrutamiento con sus vecinos.

Configuración del *router* ST. John's con SNMP

La configuración básica de SNMP permitida en *Packet Tracer* es la siguiente.

ST.John's > enable
ST.John's # configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ST.John's (config)# snmp-server community canarie ro

%SNMP-5-WARMSTART: SNMP agent on host Router is undergoing a warm start

ST.John's (config)# snmp-server community canarie rw

Configuración del *router* Vancouver con los protocolos OSPF y SNMP utilizando GNS3

Configuración de la interfaz que conecta con el *router* Victoria

Vancouver> enable

Vancouver# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Vancouver (config)# interface gigaethernet 0/0

Vancouver (config-if)# ip address 172.1.0.2 255.255.0.0

Vancouver (config-if)# no shutdown

Vancouver (config-if)# exit

**Apr 28 11:34:41.699: %LINK-3-UPDOWN: Interface GigabitEthernet 0/0*

Vancouver (config)#

Configuración de la interfaz que conecta con el *router* Edmonton

Vancouver> enable

Vancouver# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Vancouver (config)# interface gigaethernet 1/0

Vancouver (config-if)# ip address 172.2.0.1 255.255.0.0

Vancouver (config-if)# no shutdown

Vancouver (config-if)# exit

**Apr 28 11:35:11.911: %LINK-3-UPDOWN: Interface GigabitEthernet 1/0*

Vancouver (config)#

Configuración de la interfaz que conecta con el *router* Kamloops.

Vancouver> enable

Vancouver# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Vancouver (config)# interface gigaethernet 2/0

Vancouver (config-if)# ip address 172.3.0.1 255.255.0.0

Vancouver (config-if)# no shutdown

Vancouver (config-if)# exit

**Apr 28 11:35:12.911: %LINK-3-UPDOWN: Interface GigabitEthernet 2/0*

Vancouver (config)#

El protocolo OSPF utilizando GNS3 se configura de la forma siguiente.

Vancouver (config)# router ospf 1
Vancouver (config-router)# router-id 2.2.2.2
Vancouver (config-router)# network 172.1.0.0 0.0.255.255 area 0
Vancouver (config-router)# network 172.2.0.0 0.0.255.255 area 0
Vancouver (config-router)# network 172.3.0.0 0.0.255.255 area 0
Vancouver (config-router)#^Z

Una vez que se ha configurado el *router* con el protocolo de enrutamiento OSPF, este se encuentra listo para configurar el protocolo de gestión SNMP.

Configuración del *router* ST. John's con SNMP

La configuración básica de SNMP permitida en *Packet Tracer* es la siguiente.

John > enable
John # configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
John (config)# snmp-server community canarie ro SNMP_ACL
John (config)# snmp-server community canarie rw SNMP_ACL
John (config)# snmp-server contact Alonso_admin
John (config)# snmp-server location Alonso_manager
John (config)# snmp-server host 192.168.1.2 version 2c canarie
John (config)# snmp-server enable traps
John (config)# ip access-list standard SNMP_ACL
John (config-std-nacl)# permit 192.168.1.2
John (config-std-nacl)#^z

Referencias

Las referencias que sustentan el desarrollo de esta tesis están descritas siguiendo la norma **ISO 690-2.**

[1] CANARIE, *Scientific Research,* [en línea], 2017, [consulta: 18 de Marzo de 2017], Disponible: https://www.canarie.ca/scientific-research/

[2] GÉANT, *GÉANT and outGRID – Underpinning a global neuroscience grid infrastructure,* [en línea], 2017, [consulta: 18 de Marzo de 2017], Disponible: https://www.geant.net/Users/Health-and-Medicine/Pages/OutGRID.aspx

[3] CANARIE, *Research Platforms,* [en línea], 2017, [consulta: 18 de Marzo de 2017], Disponible: https://www.canarie.ca/software/platforms/

[4] CANARIE, *Private Sector,* [en línea], 2017, [consulta: 18 de Marzo de 2017], Disponible: https://www.canarie.ca/private-sector/

[5] J. I. Castillo & N. Galicia, *Routing Algorithms Applied to an Advanced Academic Network know as CUDI,* IEEE Latin America transactions, vol. 14, No. 6, pp. 2974-2979, Junio 2016, doi:10.1109/TLA.2016.7555284.

[6] Jose-Ignacio Castillo-Velázquez, Daniel-Javier Serrano-Martinez, Augusto Morales, *Emulation of backbone's connectivity and management for the advanced network in Latin America: 2016 topology, International conference on Sensor Networks Smart and Emerging Technologies (SENSET 2017); Beirut, Leabanon, 2017, pp. 1-4 (accepted paper).*

[7] J. I. Castillo-Velázquez *and* J. J. Sánchez-Trejo, *Emulation for CLARA's operations, the advanced networks for Latin America,* 2016, IEEE ANDESCON, Arequipa, 2016, pp. 1-4. Doi: 10.1109/ANDESCON.2016.7836205.

[8] Jose-Ignacio Castillo-Velázquez, Noe Galicia-Gutierrez, Juan-Arnulfo López-Ruiz, *Ingeniería inversa parcial y simulación de la infraestructura de una red de datos MAN, ROC&C Mexico, 2013*.

[9] Jose-Ignacio Castillo-Velázquez, Juan-Arnulfo López-Ruiz, ROC&C, México, 2013.

[10] A. M. Turing, *On Computable Numbers, with an application to the Entscheidungsproblem,* 12 de Noviembre de 1936.

[11] Raul Rojas, Konrad Zuse's legacy, IEEE Annals of the history of computing, vol. 19, No.2, pp. 5-16, 1997.

[12] Leonard Kleinrock, *Information flow in large communication nets*", [en línea], 31 de mayo de 1961, [consulta: 5 de Diciembre de 2016], Disponible:

https://www.lk.cs.ucla.edu/data/files/Kleinrock/Information%20Flow%20in%20Large%20Communication%20Nets.pdf

[13] J. C. R. Licklider, Welden E. Clark, "On-Line Man-Computer Communication", Cambridge, Massachusetts, 1962.

[14] Paul Baran, *On Distributed Communications: I. Introduction to Distributed Communications Networks,* Memorandum, RM-3420-PR, Agosto de 1964.

[15] José Ignacio Castillo Velázquez, *Redes de datos: Contexto y evolución,* segunda edición, México, Ed. Samsara, Enero 2016, pp. 52-56, ISBN: 978-970-94-2968-8.

[16] T. Socolofsky, C. Kale, *"A TCP/IP Tutorial",* RFC 1180, enero de 1991.

[17] Andrew G. Blank, *TCP/IP JumpStart: Internet protocol Basic,* segunda edición, Estados Unidos, Ed. Subex, 2002, pp. 6-7, ISBN: 0-7821-4101-3.

[18] UIT-T, *Redes de Datos y Comunicaciones entre Sistemas Abiertos: Interconexión de sistemas abiertos – Modelo y Notación, Recomendación UIT-T X.200,* 1995.

[19] Raymond McLeod, *Sistemas de Información Gerencial,* séptima edición, México, Ed. Prentice Hall, 2000, pp. 74-76, ISBN: 970-17-0255-7.

[20] Dago Hernando Bedoya Ortiz y Luis Alejandro Flétscher Bocanegra, *Y se crearon las redes académicas de alta velocidad… Y vieron que era bueno,* Revista académica e institucional de la UCPR, No. 76, Septiembre 2006, pp. 133-152, ISSN 0121-1633.

[21] Royer C. Hermand, *Advanced Networking Technology,* Estados Unidos, Ed. DIANE Publishing, Junio 1993, pp. 5-13, ISBN: 0-16-041805-4.

[22] Douglas Howell, *Abilene Network Upgrade to 10 Gbps Complete,* [en línea], 4 de febrero de 2004, [consulta: 13 de Febrero de 2017], Disponible: http://www.internet2.edu/news/detail/1978/

[23] Internet2, *Internet2 Community Timeline,* [en línea], 2016, [Consulta: 13 de Febrero de 2017], Disponible: http://www.internet2.edu/about-us/internet2-community-timeline/

[24] NORDUnet, *ANA-100G,* [en línea], 2016, [consulta: 26 de Febrero de 2017], Disponible: https://www.nordu.net/content/ana-100g

[25] Internet2, *Global Services,* [en línea], 2016, [consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/products-services/advanced-networking/global-services/

[26] Internet2, *Global Services Details,* [en línea], 2016, [consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/products-services/advanced-networking/global-services/global-services-details/

[27] Internet2, *Internet2,* [en línea], Abril de 2014, [consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/media/medialibrary/2014/03/17/AboutInternet2.pdf

[28] Internet2, *Layer 3 Services,* [en línea], 2016, [consulta: 14 de Febrero de 2016], Disponible: http://www.internet2.edu/products-services/advanced-networking/layer-3-services/

[29], Internet2, *Advanced Layer 3 Service: Dependable IP solutions-Engineered for research & education,* [en línea], 2016, [consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/media/medialibrary/2016/04/29/IS-advanced-layer-3-service-20160429.pdf

[30] Internet2, *Layer 2 Services,* [en línea], 2016, [consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/products-services/advanced-networking/layer-2-services/

[31] Internet2, *Advanced Layer 2 Service*, [en línea], 2017, [Consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/media/medialibrary/2016/04/29/IS-advanced-layer-2-service-20160429.pdf

[32] Internet2, *Layer 1 Services*, [en línea], 2016, [consulta: 14 de Febrero de 2017], disponible: http://www.internet2.edu/products-services/advanced-networking/layer-1-services/

[33] Internet2, *Advanced Layer 1 Service: Control your own network-without building it,* [en línea], 2016, [consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/media/medialibrary/2016/04/29/IS-advanced-layer-1-service-20160429.pdf

[34] Glenn Lipscomb, *Internet2 Announces First Full-Production Virtual Internet Network Architecture,* [en línea], 28 de Octubre de 2014, [consulta: 14 de Febrero de 2017], Disponible: http://www.internet2.edu/news/detail/7257/

[35] Internet2, *Internet2 Network Infraestructure Topology,* [en línea], Abril de 2016, [consulta: 14 de Febrero de 2016], Disponible: http://www.internet2.edu/media/medialibrary/2016/10/06/I2-Network-Infrastructure-Topology-All-201610_Yjej6o7.pdf

[36] Red CLARA, *Red CLARA y su Historia,* [en línea], 2016, [consulta: 27 de Febrero de 2017], Disponible: http://www.redclara.net/index.php/somos/sobre-redclara/historia

[37] Alice2, *Antecedente: Proyecto ALICE,* [en línea], 2016, [consulta: 27 de Febrero de 2017], Disponible: http://alice2.redclara.net/index.php?option=com_content&view=article&id=12&Itemid=8&lang=es

[38] CUDI*, Antecedentes; Cooperación Latino Americana de redes Avanzadas-CLARA,* [en línea], 9 de Junio de 2003, [consulta: 27 de Febrero de 2017], Disponible: http://www.cudi.edu.mx/acerca-de-cudi/antecedentes/convenio_RedCLARA

[39] Red CLARA, *Memoria 2007,* [en línea], Abril de 2008, [consulta: 28 de Febrero de 2017], Disponible: https://www.redclara.net/index.php/somos/sobre-redclara/memorias-anuales

[40] Red CLARA, *WHREN/LILA,* [en línea], 2007, [consulta: 28 de Febrero de 2017], Disponible: https://www.redclara.net/index.php/productos-y-servicios/concta-redclara/33-contenido/3124-whren

[41] Alberto Cabezas B. y M. Soledad Bravo M., *Redes Avanzadas en América Latina: Infraestructura para el desarrollo regional en ciencia, tecnología e innovación,* [en línea], 2010, [consulta: 27 de Febrero 2017], Disponible: http://alice2.redclara.net/images/ALICE2/documents/libro_blanco_espanol.pdf

[42] Red CLARA, *Memoria anual 2012,* [en línea], Octubre de 2013, [consulta: 1 de Marzo de 2017], Disponible: http://dspace.redclara.net/bitstream/10786/606/1/Memoria_RedCLARA_2012.pdf

[43] Red CLARA, *Memoria anual 2014,* [en línea], Junio de 2015, [consulta: 1 de Marzo de 2017], Disponible: http://dspace.redclara.net/bitstream/10786/954/1/RedCLARA_memoria_2014.pdf

[44] Red CLARA, *Memoria anual 2015,* [en línea], Abril de 2016, [consulta: 1 de Marzo de 2016], Disponible: https://www.redclara.net/index.php/somos/sobre-redclara/memorias-anuales

[45] Red CLARA, *Topología de Red CLARA,* [en línea], 2017, [consulta: 8 de Agosto de 2017], Disponible: www.redclara.net/index.php/red-y-conectividad/topologia

[46] CANARIE, *A Nation Goes On Line,* [en línea], 22 Enero de 2010, [consulta: 13 de Marzo de 2017], Disponible: https://www.canarie.ca/about-us/documents/

[47] IBM, Protocolo de control de Transmisiones/Protocolo Internet (Transmission Control Protocol/Internet Protocol), [en línea], 2017, [consulta: 13 de Marzo de 2017], Disponible: https://www.ibm.com/support/knowledgecenter/es/ssw_aix_72/com.ibm.aix.networkcomm/tcpip_intro.htm

[48] P Gauvin, DREnet, *The DRENET \ (Defense Reseach Establishment Network)\: Planning and Development Perspective*, [en línea], Marzo 1994, [consulta: 13 de Marzo de 2017], Disponible: http://cradpdf.drdc-rddc.gc.ca/PDFS/zba0/p146579.pdf

[49] Robert H Zakon, *Hobbes' Internet Timeline 24,* [en línea], 1993, [consulta: 13 de Marzo de 2017], Disponible: https://www.zakon.org/robert/internet/timeline/

[50] Claude Cantin, *A Network of Networks,* [en línea], Octubre de 2010, [consulta: 14 de Marzo de 2017], Disponible: http://rcsg-gsir.imsb-dsgi.nrc-cnrc.gc.ca/documents/internet/node6.html

[51] CANARIE, *About US,* [en línea], 2017, [consulta: 15 de Marzo de 2017], Disponible: https://www.canarie.ca/about-us/

[52] CANARIE, *National Research and Education Network,* [en línea], 2017, [consulta: 4 de Septiembre de 2016], Disponible: https://www.canarie.ca/network/nren/

[53] HEPnet/Canada, *Canadian Networks for Particle Physics Research: 2014 Report to the Standing Committee on Interregional Connectivity, ICFA Panel,* [en línea], Enero 2015, [consulta: 15 de Marzo de 2017], Disponible: http://hepnetcanada.ca/assets/themes/hepnet/documents/Canadian-ICFA-SCIC-Network-2014.pdf

[54] CANARIE, *CANARIE Network Routing Policy,* [en línea], 25 de Agosto de 2010, [Consulta: 16 de Marzo de 2017], Disponible: https://www.canarie.ca/wpdm-package/canarie-policy-network-routing/?wpdmdl=6880

[55] ARIN, *ARIN at Glance,* [en línea], 1997, [consulta: 16 de Marzo de 2017], Disponible: https://www.arin.net/about_us/overview.html

[56] APNIC, *What we do,* [en línea], 2017, [consulta: 16 de Marzo de 2017], Disponible: https://www.apnic.net/about-apnic/organization/

[57] RIPE NCC, *What we do,* [en línea], 23 de Diciembre de 2010, [consulta: 16 de Marzo de 2017], Disponible: https://www.ripe.net/about-us/what-we-do

[58] CANARIE, *IPv6 Address Allocation and Assignament Policy*, [en línea], 27 de Agosto de 2010, [consulta: 16 de Marzo de 2017], Disponible: https://www.canarie.ca/?wpdmdl=6877

[59] Red CLARA, *CANARIE y BCNET ayudan a los canadienses a prepararse para la transición a IPv6,* [en línea], 11 de Junio de 2012, [consulta: 16 de Marzo de 2017], Disponible: http://www.redclara.net/index.php/en/component/content/article/15-noticias/1175-canarie-y-bcnet-ayudan-a-los-canadienses-a-prepararse-para-la-transicion-a-ipv6

[60] Ciena, *CANARIE Expands 100G Research & Education Network with Ciena,* [en línea], 18 de Junio de 2013, [consulta: 15 de Marzo de 2017], Disponible: http://www.ciena.com/about/newsroom/press-releases/CANARIE-Expands-100G-Research--Education-Network-with-Ciena_prx.html

[61] Ciena, *Waveserver,* [en línea], 2017, [consulta: 17 de Marzo de 2017], Disponible: http://www.ciena.com/products/waveserver/?src=PR

[62] Ciena, *WaveLogic Photonics: Maximize capacity with industry-leading WaveLogic coherent optics,* [en línea], 2017, [consulta: 17 de Marzo de 2017], Disponible: http://www.ciena.com/products/wavelogic/

[63] Ciena, *CANARIE, StarLight and Ciena Complete 300G Trial,* [en línea], 15 de Junio de 2016, [consulta: 17 de Marzo de 2017], Disponible: http://www.ciena.com/about/newsroom/press-releases/CANARIE-StarLight-and-Ciena-Complete-300G-Trial_prx.html

[64] SRnet, *Canada's National Research and Education Network,* [en línea], 2016, [consulta: 15 de Marzo de 2017] Disponible: http://srnet.ca/network/canadas-nren/

[65] María del Carmen Romero Ternero y Julio Barbancho Consejero, *Redes Locales,* segunda edición, España, 2014, pp. 209-211, ISBN: 978-84-283-3530-0.

[66] IBM, *Direccionamiento estático y Dinámico,* [en línea], 2017, [Consulta: 17 de Marzo de 2017], Disponible: https://www.ibm.com/support/knowledgecenter/es/ssw_aix_61/com.ibm.aix.networkcomm/tcpip_routing_types.htm

[67] CISCO, *CCNA 2 exploration: Conceptos y protocolos de enrutamiento*, v 4.0, 2008, pp. 123-128.

[68] C. Hedrick, Rutgers University, "*Routing Information Protocol*", RFC 1058, Junio de 1988.

[69] G. Malkin, R. Minnear, "*RIPng for IPv6",* RFC 2080, Enero de 1997.

[70] Eduardo Collado Cabeza, *Fundamentos de Routing,* España, Ed. Autor-Editor, 2009, pp. 71-81, ISBN: 978-1409284635.

[71] Julio Barbancho Consejero, Jaime Benjumea Mondéjar, *Redes Locales,* segunda edición, España, Ed. Paraninfo, 2014, pp. 119-110, ISBN: 978-84-283-3530-0.

[72] J. Postel, *User Datagra Protocol, RFC 768,* 28 de Agosto de 1980.

[73] G. Malking, Bay Networks, "*RIP verison 2*", RFC 2453, Noviembre de 1998.

[74] Deepankar Medhi, Karthikeyan Ramasamy, *Network Routing: Algorithms, Protocols, and Architectures,* Estados Unidos, Ed. Morgan Kaufmann, 2007, pp. 148-152, ISBN: 978-0-12-088588-6.

[75] DARPA, *RFC 1131, The OSPF Specification,* Octubre de 1989.

[76] J. Moy, Proteon Inc., "*OSPF Version 2*", RFC 1247, Julio de 1991.

[77] J. Moy, Proteon Inc., "OSPF Version 2", RFC 2328, Abril de 1998.

[78] Pablo Gil, Jorge Pomares y Francisco Candelas, *Redes y Transmisión de Datos,* España, Publicaciones de la universidad de Alicante, 2010, pp. 180-181 ISBN: 978-84-9717-125-0.

[79] Eduardo Collado Cabeza, *Fundamentos de Routing,* España, Ed. Autor-Editor 2009, pp. 99-124, ISBN: 978-1409284635.

[80] John T. Moy, *OSPF Anatomy of an Internet Routing Protocol,* Estados Unidos, Ed. Pearson, 1998, pp. 74-84, ISBN: 0-201-63472-4.

[81] Phani Raj Tadimety, *OSPF: A Network Routing Protocol,* Nueva York, Ed. Apress, 2015, pp. 76-89, ISBN: 978-1-4842-1410-7.

[82] Diane Teare, Bob Vachon y Rick Graziani, *Implementing Cisco IP Routing (ROUTE): Foundation Learning Guide,* CCNP ROUTE 300-101, Estados Unidos, Cisco Press, 2015, pp.160-161, ISBN: 978-1-58720-456-2.

[83] Diane Teare, Bob Vachon y Rick Graziani, *Implementing Cisco IP Routing (ROUTE): Foundation Learning Guide,* CCNP ROUTE 300-101, Estados Unidos, Cisco Press, 2015, pp. 211, ISBN: 978-1-58720-456-2.

[84] T. Saydam y T. Magedanz, *From Networks and Network Management into Service and Service Management, Journal of Networks and Systems Management,* Vol. 4, No. 4, Diciembre de 1996.

[85] Julian Veron Piquero, *Prácticas de Redes,* segunda edición, 2010, pp. 201-210, ISBN: 978-84-692-5173-7.

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

[87] K. McCloghrie, M. Rose, *Management Information Base for network Management of TCP/IP-based internets, RFC 1066.* Agosto de 1988.

[88] M. Fedor, M. Schoffstall, J. Davin, *A Simple Network Management Protocol, RFC 1067.* Agosto de 1988.

[89] Thomas Akin, *Hardening Cisco Router: Help for Network Administrators,* Estados Unidos, Ed. O'Reilly, 2002, pp. 69-71, ISBN: 0-596-00166-5.

[90] Behrouz A. Forouzan, *Transmisión de datos y redes de comunicaciones,* segunda edición, España, Ed. McGrawHill, 2001, ISBN: 84-481-3390-0.

[91] IBM, *Protocol Data Unit (PDUs),* [en línea], 2017, [consulta: 20 de Marzo de 2017], Disponible: https://www.ibm.com/support/knowledgecenter/en/SSB23S_1.1.0.12/gtpc1/pdus.html

[92] M. Rose, *"Structure and Identification of Management Information for TCP/IP-based Internets",* RFC 1155, Mayo de 1990

[93] José Ignacio Castillo, *El árbol de Internet y la estructura de información de gestión de una red,* IEEE Latin America and the Caribbean Newsletter, Año 20, No. 62, pp. 15-17, Abril de 2009, ISSN: 2157-8354.

[94] U. Warrier, L. Besaw, *The Common Management Information Services and Protocolo over TCP/IP (CMOT), RFC 1095.* Abril de 1989.

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

[96] Douglas R. Mauro y Keving J. Schmidt, *Essential SNMP,* segunda edición, Estados Unidos, septiembre 2005, Ed. O'Reilly, pp. 19-35, ISBN: 0-596-00840-6.

[97] Cisco, *Cisco Packet Tracer,* [en línea], 2017, [consulta: 4 de Julio de 2017], Disponible: http://www.cisco.com/c/dam/en_us/training-events/netacad/course_catalog/docs/Cisco_PacketTracer_AAG.pdf

[98] María Cecilia Guillermo G., Carlos G. Alonzo Blanqueto, *Medios de enseñanza: Material para el autoaprendisaje,* Yucatán, México, Ediciones de la Universidad Autónoma de Yucatán, 1997, pág. 63, ISBN: 968-7556-53-6.

[99] GNS3, *Getting Started with GNS3: Introduction,* [en línea], 2017, [consulta: 4 de Julio de 2017], Disponible: https://docs.gns3.com/1PvtRW5eAb8RJZ11maEYD9_aLY8kkdhgaMB0wPCz8a38/index.html

[100] BCNET, *National Research & Education Network,* [en línea], 2017, [consulta: 25 de Abril de 2017], Disponible: https://www.bc.net/advanced-network/national-research-education-network

[101] Intel, *Intel Pentium Dual-Core Desktop Processor E2000 Series,* [en línea], Marzo 2008, [consulta: 17 Mayo de 2017], Disponible: http://www.intel.com/content/dam/support/us/en/documents/processors/pentiumdualcore/sb/316981.pdf

[102] Intel, *Intel Xeon Processor E5-2620 v2,* [en línea], 2017, [consulta: 17 de Mayo de 2017], Disponible: https://ark.intel.com/es/products/75789/Intel-Xeon-Processor-E5-2620-v2-15M-Cache-2_10-GHz

[103] Cisco, *SNMP Object Navigator,* [en línea], 2017, [consulta: 8 de Mayo de 2017], Disponible: http://snmp.cloudapps.cisco.com/Support/SNMP/do/BrowseOID.do?local=en&translate=Translate&objectInput=clogMessageGenerated#oidContent

Index

Abstract

CANARIE's network was established in 1993, since its construction has been concerned to build a backbone capable to support the information generated by its members. In June 2013, CANARIE introduced a new infrastructure supporting 100 Gbps; using technology by Ciena 6500 whitch includes third generation optical Wavelogic processor. Then, in 2016 CANARIE started a new plan to get a new backbone with the capacity of 300 Gbps using the waveserver by Ciena.

With the purpose of knowing and understanding the operation of the CANARIE'S backbone, it was simulated using Packet Tracer as first approximation and emulated using GNS3 as second approximation. The backbone has 25 routers, each one of them was configured using OSPF routing protocol and management SNMP protocol.

Connectivity simulation used 32% of the Intel (R) Pentium (R) Dual CPU E2140@ 1.60 GHZ processor and 40% (1.20 GB of 3 GB) of RAM simulating connectivity, from router Vancouver to router ST. Jonh's, in 2.48 minutes. Management simulation in most of the cases performed the monitoring of the management elements, allowing configuring two of the five variables. On the other hand, the emulation of connectivity and management was made in a computer with better capacities of CPU and RAM. The emulation of connectivity used 4% of Intel (R) Xeon (R) E5-2620 v2 @ 2.10 GHz processor and 37.81% (12.1 GB of 32 GB) of RAM. Spent time since the execution of GNS3 until having the open terminals of the five virtual computers was of 9.50 minutes. The management let configure 3 of the 5 proposed variables, also allowed to receive traps sent by the management devices. Network configuration and monitoring knowledge was applied based on connectivity and management via simulation and emulation of the CANARIE's backbone, which has 25 backbone routers.

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

SCIENCE AND TECHNOLOGY SCHOOL

BACHELOR OF ENGINEERING IN ELECTRONICS AND TELECOMMUNICATIONS SYSTEMS

Monitoring in the Advanced Network CANARIE:

Emulation

THESIS

TO OBTAIN THE TITLE OF BACHELOR IN ENGINEERING IN ELECTRONIC AND TELECOMMUNICATIONS SYSTEMS

P R E S E N T S:

ALONSO DELGADO VILLEGAS

THESIS DIRECTOR

M. en C. José Ignacio Castillo Velázquez

Mexico City, October 2017