Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Danilo Escobar Coronado
Asesorado por el Ing. Carlos Eduardo Guzmán Salazar

Guatemala, abril de 2016

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

DANILO ESCOBAR CORONADO
ASESORADO POR EL ING. CARLOS EDUARDO GUZMÁN SALAZAR

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, ABRIL DE 2016

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Inga. Elvia Miriam Ruballos Samayoa
VOCAL IV	Br. Raúl Eduardo Ticún Córdova
VOCAL V	Br. Henry Fernando Duarte García
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Carlos Eduardo Guzmán Salazar
EXAMINADORA	Inga. María Magdalena Puente Romero
EXAMINADOR	Ing. Hugo Ernesto Mazariegos Santizo
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería de Mecánica Eléctrica, con fecha 20 noviembre de 2014.

Danilo Escobar

Danilo Escobar Coronado

Guatemala, 4 de febrero de 2016

Coordinador Área de Electrónica
Facultad de Ingeniería
Escuela de Mecánica Eléctrica
Presente

A quien corresponda:

Le informo que he revisado el trabajo de graduación "**PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**" desarrollado por el estudiante de Ingeniería Electrónica Danilo Escobar Coronado con carné 2004-12971.

Considero que este trabajo está bien desarrollado y representa un aporte para la Facultad de Ingeniería y habiendo cumplido con los objetivos del referido trabajo doy mi aprobación al mismo.

Atentamente,

ID Y ENSEÑAD A TODOS

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Carlos Guzmán Salazar
Asesor de Tesis

CARLOS GUZMAN Salazar
Ingeniero Electricista
Col. No. 2763

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

UNIDAD DE EPS

Guatemala, 23 de febrero de 2016.
Ref.EPS.DOC.112.02.16.

Ing. Silvio José Rodríguez Serrano
Director Unidad de EPS
Facultad de Ingeniería
Presente

Estimado Ingeniero Rodríguez Serrano.

Por este medio atentamente le informo que como Supervisor de la Práctica del Ejercicio Profesional Supervisado (E.P.S.), del estudiante universitario **Danilo Escobar Coronado** de la Carrera de Ingeniería Electrónica, **con carné No. 200412971 ,** procedí a revisar el informe final, cuyo título es **"PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA".**

En tal virtud, **LO DOY POR APROBADO**, solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,

"Id y Enseñad a Todos"

Ing. Francisco González
Supervisor de EPS
Área de Ingeniería Eléctrica

Universidad de San Carlos de Guatemala
Ing. Francisco González
ASESOR - SUPERVISOR DE EPS
Unidad de Prácticas de Ingeniería y EPS
Facultad de Ingeniería

c.c. Archivo
/ra

NIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

REF. EIME 10. 2016.
Guatemala, 8 de FEBRER0 2016.

Señor Director
Ing. Francisco Javier González López
Director Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado: PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, del estudiante Danilo Escobar Coronado, que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Carlos Eduardo Guzmán Salazar
Coordinador Área Electrónica

sto

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

UNIDAD DE EPS

Guatemala 23 de febrero de 2016.
Ref.EPS.D.99.02.16.

Ing. Francisco Javier González
Director Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería
Presente

Estimado Ingeniero González.

Por este medio atentamente le envío el informe final correspondiente a la práctica del Ejercicio Profesional Supervisado, (E.P.S) titulado **"PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA"** que fue desarrollado por el estudiante universitario, **Danilo Escobar Coronado,** quien fue debidamente asesorado por el Ing. Carlos Guzmán Salazar y supervisado por el Ing. Francisco González.

Por lo que habiendo cumplido con los objetivos y requisitos de ley del referido trabajo y existiendo la aprobación del mismo por parte del Asesor - Supervisor de EPS, en mi calidad de Director apruebo su contenido solicitándole darle el trámite respectivo.

Sin otro particular, me es grato suscribirme.

Atentamente,
"Id y Enseñad a Todos"

Ing. Silvio José Rodríguez Serrano
Director Unidad de EPS

Universidad de San Carlos de Guatemala
DIRECCIÓN
Unidad de Prácticas de Ingeniería y EPS
Facultad de Ingeniería

SJRS/ra

NIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

REF. EIME 10. 2016.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto bueno del Coordinador de Área, al trabajo de Graduación del estudiante; DANILO ESCOBAR CORONADO Titulado: PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA, procede a la autorización del mismo.

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

Ing. Francisco Javier González López

GUATEMALA, 29 DE FEBRERO 2016.

Universidad de San Carlos
De Guatemala

Facultad de Ingeniería
Decanato

Ref. DTG.D.204.2016

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica al trabajo de graduación titulado: **PROPUESTA DE DISEÑO E IMPLEMENTACIÓN DEL LABORATORIO DE TELECOMUNICACIONES Y REDES LOCALES DE LA ESCUELA DE MECÁNICA ELÉCTRICA, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA,** presentado por el estudiante universitario: **Danilo Escobar Coronado,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, se autoriza la impresión del mismo.

IMPRÍMASE.

Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, abril de 2016

/gdech

ACTO QUE DEDICO A:

Dios	A quien agradezco por darme la perseverancia y paciencia suficientes para concluir mis estudios.
Mi papá y mi abuela	Quienes siempre me apoyaron y aconsejaron.
Mis hermanos	Rodrigo y Laura Escobar Coronado, quienes siempre estuvieron conmigo.
Shihan Arturo Armas García	Quien ha sido un ejemplo para mí.
Mis amigos	Ana Morales, Dante Crovella, David Recinos, Elmer Alvarado, Gelion Osorio, Gustavo Lima, Haroldo Rufino, Heber Hernández, Hermes Escobar, Humberto Barrera, Jaime López, Jorge Guzmán, Josman Flores, Juan Manuel Elías, Juan Pablo Guerra, Marco Crocker, Mario Beteta, Mayron Rodríguez, Milton Rodas, Randy Hernández, René Rocael, Rosario González, Sergio Rodriguez, Daniel Tavico, Estefana Lemus, Verónica López, Samuel Hernández , Waldemar de León y todos aquellos con los que compartí experiencias.

Mis estudiantes	Haber contribuido a su formación es el máximo honor que me llevo de la universidad.
Los ingenieros	Hugo Mazariegos, Carlos Guzmán, Helmunt Chicol e Ingrid de Loukota, quienes fueron amables conmigo cuando lo necesité.
Carolina Villatoro	Sin quien mis proyectos no hubieran sido posibles.
Jessica Villagrán	Mi mejor amiga, ya que sin su apoyo y compañía no hubiera podido graduarme.
Mi mamá	Ingrid Coronado, por ser el motor principal del cierre de este ciclo.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

GLOSARIO

Autosumarización Se llama así a la sumarización que se realiza de manera automática, sin ninguna intervención por parte del usuario. Este comportamiento es propio de los protocolos de enrutamiento vector distancia.

Adyacencias Relación establecida entre dispositivos vecinos donde es posible el intercambio de información relativa al enrutamiento.

Ancho de banda La capacidad de un medio de transmitir información.

Bit Unidad mínima de información empleada en informática.

Bucles Circuitos lógicos cerrados en donde la información se queda dando vueltas sin jamás llegar a su destino.

Concentrador Dispositivo destinado a proveer de conectividad de red a varios dispositivos.

Enlace Conexión a través de la cual se transmiten datos.

Enlace troncal	Conexión utilizada por los *switches* para transmitir información de varias VLANs a través de un solo enlace.
Enrutamiento	Acción de encontrar la mejor ruta (o camino) entre todas las posibles, inclusodentro de una disposición con un alto grado de interconectividad o redundancia.
Estado de enlace	Categoría en la que se clasifican aquellos protocolos de enrutamiento que tienen una visión completa de las interconexiones de los demás dispositivos, ejecutando dicho protocolo dentro de la red.
Dirección IP	Identificador utilizado por el protocolo de internet (Internet *Protocol*).
Identificador	Parámetro empleado para distinguir entre varios nodos.
Interfaz	Nombre otorgado a las conexiones que presentan los dispositivos.
Listas de control de acceso	Mecanismo utilizado para identificar tráfico dentro de una red para luego aplicar una acción sobre el mismo.

Máscara de subred	Máscara de bits que en combinación con la dirección IP es capaz de proveer información importante acerca de la conexión de red a un dispositivo.
NAT	Mecanismo a través del cual se pueden cambiar o traducir direcciones IP, usualmente es utilizado para convertir direcciones IP privadas (no enrutables en internet) a públicas.
Nodo	Punto de intersección, unión o terminación de los distintos elementos que componen la red.
Paquete	Nombre dado a los bloques en los cuales es dividida la información antes de su envío en el momento que se les asigna una dirección IP de origen y destino.
Protocolo	Una serie de reglas definidas para llevar a cabo una acción.
Red local	Anteriormente el término designaba a un conjunto de dispositivos que se encontraban en la misma área geográfica. Actualmente, se utiliza, para designar a todos aquellos aparatos que se encuentran bajo una misma administración.
Red a pie	La primera red de transmisión de datos donde los usuarios debían levantarse de sus estaciones e intercambiar información, utilizando algún tipo de almacenamiento portátil.

Router	Dispositivo encargado de encontrar todas las posibles rutas y elegir las mejores para que sean utilizadas en la transmisión de datos.
Ruta	Vía a través de la cual es posible alcanzar una red determinada.
Sumarización	Acción de advertir varias redes pequeñas como si fueran parte de un esquema más grande con el fin de reducir el tamaño de las tablas de enrutamiento de los demás dispositivos.
Switch	Concentrador de área local capaz de crear circuitos únicos entre los dispositivos a nivel de *hardware*.
Tabla de enrutamiento	El lugar donde los *routers* almacenan las mejores rutas.
Topología	Disposición física en la que se conectan los dispositivos a la red.
Vecindades	Relación establecida entre dispositivos que comparten un mismo segmento de la red.
Vector distancia	Categoría en la que se clasifican aquellos protocolos de enrutamiento que no tienen una visión completa de la disposición de la red y que se limitan a analizar la información de la dirección y la lejanía (medida

	acorde a diferentes parámetros según el protocolo que se esté utilizando) de una red determinada.
VLAN	Redes locales virtuales creadas a nivel lógico dentro de los *switches,* destinadas a eliminar limitaciones físicas y mejorar la organización de la red local.

RESUMEN

Desde 1967, la carrera de Ingeniería Electrónica ha sido la rama de la ingeniería que se ocupa de transmitir a los estudiantes de la Universidad de San Carlos de Guatemala los principios que hacen posible el intercambio de información a gran escala.

Dicho campo del conocimiento ha sido partícipe e importante actor de la innegable revolución de la información, de la cual la mayoría de la humanidad ha sido parte durante las últimas dos décadas y que es consecuencia directa del nacimiento del sistema conocido comúnmente como “la red”, el cual permite la interconexión no jerárquica de muchos nodos permitiendo que estos compartan recursos e información.

De esta nueva forma de comunicación surge la necesidad, como academia, de la creación e implementación de un nuevo laboratorio dentro de la carrera mencionada, en donde el estudiante pueda poner en práctica y familiarizarse con los principios detrás de esta nueva tecnología, siendo el objetivo principal de este trabajo responder a esta necesidad.

Por esta razón, en el transcurso de este trabajo se presentan de manera muy breve algunas de las generalidades de la Universidad de San Carlos de Guatemala y de la Escuela de Ingeniería Mecánica Eléctrica, seguido de la distribución sugerida del laboratorio propuesto que incluye un programa, un calendario de actividades, el marco teórico del mismo, bibliografía, así como una guía para el auxiliar con actividades, tareas y referencias a otros recursos tanto de terceros como a aquellos creados por el autor.

OBJETIVOS

General

Presentar una propuesta de diseño y un plan de implementación para el laboratorio de la clase “Telecomunicaciones y Redes Locales (código 969)” de la carrera de Ingeniería Electrónica.

Específicos

1. Presentar las generalidades de la Universidad de San Carlos de Guatemala, así como de la Facultad de Ingeniería y su Escuela de Ingeniería Mecánica Eléctrica.

2. Proponer la distribución del nuevo laboratorio, contenido y calendarización.

3. Presentar los fundamentos teóricos necesarios para comprender el funcionamiento de las redes locales.

4. Redactar una propuesta de guía de prácticas para orientar al auxiliar del curso.

5. Proporcionar referencias al material de apoyo y equipo en existencia.

INTRODUCCIÓN

Desde su fundación en 1988, la carrera de Ingeniería Electrónica ha sido la rama de la Escuela de Ingeniería Mecánica Eléctrica encargada de transmitir a los estudiantes de la Universidad de San Carlos de Guatemala los principios que hacen posible el intercambio de información a través del estudio de las telecomunicaciones, uno de los campos de más rápida evolución y crecimiento en los últimos veinte años.

Debido a la revolución informática, la posibilidad de nuevos servicios y a la reducción de costos, muchas tecnologías tradicionales, regularmente dispares entre ellas, han sido reemplazadas o adaptadas para funcionar en un nuevo sistema en donde es posible el intercambio de información de un nodo o participante a una cantidad arbitraria de los mismos a través de un serie de conexiones conocidas popular y simplemente como “la red”.

Como consecuencia natural de este nuevo salto, en la forma en que el ser humano se comunica, surge la necesidad de implementar un nuevo laboratorio dentro de la carrera mencionada, en donde los estudiantes tengan la oportunidad de familiarizarse con este nuevo campo, siendo el objetivo primordial de este trabajo responder a dicha necesidad.

1. GENERALIDADES DE LA UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

1.1. Historia

La Universidad de San Carlos de Guatemala, también conocida por sus siglas como la USAC, es la más antigua de Guatemala, fundada en 1676 es la única casa de estudios pública del país.

Establecida durante el tiempo de La Colonia por la corona española como la Universidad Real y Pontificia de San Carlos de Borromeo, evoluciona a su forma actual tras la Revolución de 1944.

Su sede principal se encuentra en la capital de Guatemala, en el sector que se conoce como Ciudad Universitaria, zona 12. Cuenta con centros universitarios en la mayoría de regiones de Guatemala y con un Centro Universitario Metropolitano (CUM) donde funciona la Facultad de Medicina y la Escuela de Psicología.

1.2. Misión

“En su carácter de única universidad estatal, le corresponde con exclusividad dirigir, organizar y desarrollar la educación superior del Estado y la educación estatal, así como la difusión de la cultura en todas sus manifestaciones. Promoverá por todos los medios a su alcance la investigación

en todas las esferas del saber humano y cooperará al estudio y solución de los problemas nacionales".[1]

1.3. Visión

"La Universidad de San Carlos de Guatemala es la institución de educación superior estatal, autónoma, con una cultura democrática, con enfoque multi e intercultural, vinculada y comprometida con el desarrollo científico, social, humanista y ambiental, con una gestión actualizada, dinámica, efectiva y con recursos óptimamente utilizados, para alcanzar sus fines y objetivos, formadora de profesionales con principios éticos y excelencia académica".[2]

1.4. Facultad de Ingeniería

La Facultad de Ingeniería es una de las 10 facultades que conforman la Universidad de San Carlos de Guatemala. Fundada en 1880 es la Facultad de Ingeniería más grande de Guatemala. Atiende a una población estudiantil de más de 12 000 estudiantes de pregrado siendo por ende una de las unidades académicas más pobladas de la Universidad.

1.4.1. Historia

El origen de la enseñanza de ciencias exactas en Guatemala se remonta a 1834, con la creación de la "Escuela de Ciencias" donde empezaron a impartirse cursos de Álgebra, Geometría, Trigonometría y Física; los cuales

[1] Universidad San Carlos de Guatemala. *Misión y visión.* http://digi.usac.edu.gt/sitios/transparencia/misioacuten-visioacuten-y-objetivos.html. Consulta: abril de 2015.

[2] Ibíd.

más tarde se volverían la piedra angular para que en 1873 iniciara la carrera de Ingeniería en la Escuela Politécnica.

Más adelante, en 1879 se estableció la Escuela de Ingeniería en la Universidad de San Carlos de Guatemala, elevada a la categoría de Facultad tres años más tarde. Su primer decano fue el Ing. Cayetano Batres del Castillo.

Actualmente, la Facultad de Ingeniería es una de las más grandes de la Universidad de San Carlos con una población estudiantil de más de 12 000 estudiantes. Ofrece 12 programas de pregrado, 8 de posgrado y 14 de maestría.

1.4.2. Misión

"Formar profesionales en las distintas áreas de la ingeniería que, a través de la aplicación de la ciencia y la tecnología, conscientes de la realidad nacional y regional, y comprometidos con nuestras sociedades, sean capaces de generar soluciones que se adapten a los desafíos del desarrollo sostenible y los retos del contexto global".[3]

1.4.3. Visión

"Somos una institución académica con incidencia en la solución de la problemática nacional, formando profesionales en las distintas áreas de la ingeniería, con sólidos conceptos científicos, tecnológicos, éticos y sociales, fundamentados en la investigación y promoción de procesos innovadores orientados hacia la excelencia profesional".[4]

---

[3] Universidad de San Carlos de Guatemala, Facultad de Ingeniería. *Misión y visión*. https:// www.ingenieria.usac.edu.gt/nosotros.php. Consulta: abril de 2015.

[4] Ibíd.

1.4.4. Escuela de Ingeniería Mecánica Eléctrica

La creación de la Escuela de Ingeniería Mecánica Eléctrica fue aprobada por el Consejo Superior Universitario en agosto de 1967, bajo la dirección del ingeniero fundador de la misma Rodolfo Koenigsberger. Inicialmente contaba solamente con las carreras de Ingeniería Eléctrica y Mecánica Electricista, y no fue sino hasta 1988 que se creó la carrera de Ingeniería Electrónica.

1.4.4.1. Misión

"Formar profesionales competentes, con principios éticos y conciencia social, en los campos de las ingenierías Mecánica Eléctrica, Eléctrica y Electrónica, mediante técnicas de enseñanza actualizadas y fundamentados en la investigación, comprometidos con la sociedad, con el fin de contribuir al bien común y al desarrollo sostenible del país y de la región".[5]

1.4.4.2. Visión

"Ser la institución académica líder a nivel nacional y regional, con incidencia en la problemática nacional, en la formación de profesionales de calidad, en los campos de las ingenierías Mecánica Eléctrica, Eléctrica y Electrónica; emprendedores con sólidos conocimientos científicos, tecnológicos, éticos, sociales; fundamentados en la investigación, orientados hacia la excelencia, reconocidos internacionalmente y comprometidos con el desarrollo sostenible de Guatemala y de la región".[6]

[5] Escuela de Mecánica Eléctrica, Usac. *Misión y Visión*. http://eime.ingenieria.usac.edu.gt/index.php?option=com_content&view=article&id=5&Itemid=7. Consulta: abril de 2015.
[6] Ibíd.

2. DISTRIBUCIÓN DEL LABORATORIO PROPUESTO

2.1. Misión

Contribuir con los objetivos de la Escuela de Ingeniería Mecánica Eléctrica, al coadyuvar en la formación de los futuros profesionales en materia de redes de área local. Se pretende establecer un balance entre teoría y práctica, y utilizar los mejores recursos para conseguirlo.

2.2. Visión

Constituir un laboratorio en donde el estudiante sea el recurso más valioso, siempre en una constante evolución que facilite su adaptación y anticipación a las necesidades del mercado nacional.

2.3. Objetivos

- Mejora continua del nivel académico de la institución académica.
- Promover la formación de los estudiantes en áreas complementarias a la ingeniería, así como la práctica de valores y principios éticos y morales.
- Promover la formación en la investigación e impulsar su práctica en docentes y estudiantes.
- Promover la extensión de la ingeniería a través de su práctica con proyección social.
- Lograr la acreditación a nivel regional.[7]

[7] Escuela de Mecánica Eléctrica, Usac. *Objetivos*. http://eime.ingenieria.usac.edu.gt/ index.php/2015-10-06-22-23-45/objetivos-estrategicos. Consulta: abril de 2015.

2.3.1. Objetivo general

Contribuir con el proceso de enseñanza-aprendizaje, al proveer al funcionamiento de las redes locales; además de un programa donde la teoría y la práctica se presenten en proporciones adecuadas, para ayudarlo a desarrollar habilidades que le permitan implementar soluciones en el ámbito laboral. Asimismo, orientarlo al estudio de una especialización o investigación sobre un tema relacionado.

2.3.2. Objetivos específicos

- Proveer los fundamentos teóricos necesarios para la comprensión del funcionamiento de las redes locales.

- Proporcionar al estudiante un programa donde la teoría y la práctica se presenten en proporción adecuada. En el cual los ejercicios y tareas sean relizados con un equipo real y software de código abierto, de modo que desarrolle las habilidades necesarias para implementar soluciones en el ámbito laboral.

- Orientar al estudiante hacia otras fuentes de información, a manera que pueda profundizar en un tema específico o iniciar el estudio de una especialización dentro del mismo contexto.

2.4. Perfil del auxiliar del laboratorio

El auxiliar a cargo deberá tener conocimiento de los fundamentos de las tecnologías cubiertas en el programa del laboratorio e idealmente poseer alguna certificación internacional en la materia. El equipo en existencia es marca

Cisco, por lo que se recomienda el Cisco Certified Network Associate (CCNA) o el Cisco Certified Network Professional (CCNP).

Además, tener facilidad de expresión y paciencia, ya que debe transmitir conceptos complejos utilizando ideas sencillas.

De no existir una persona con este perfil, se recomienda realizar un examen de oposición entre los alumnos que hayan aprobado el curso con las puntuaciones más altas.

2.5. Responsabilidades del auxiliar del laboratorio

Además de presentarse a impartir clases, le corresponde verificar al inicio de cada semestre los siguientes aspectos:

- Acordar el horario, lugar y duración del laboratorio con los estudiantes de la carrera y el ingeniero a cargo.

- Actualizar el material didáctico, software necesario y documentación a medida que sea necesario y asegurarse que los mismos estén disponibles para todos los estudiantes interesados.

- Determinar el estado del equipo de red del laboratorio y darle mantenimiento, si fuera necesario.

- Asegurarse que el estado de las computadoras del laboratorio sea óptimo y que el software necesario haya sido instalado.

- Informar a los nuevos estudiantes de la existencia del material de apoyo (documentación, software, entre otros) y cómo pueden obtenerlo. También de la calendarización del curso, distribución de exámenes y tareas y de cualquier otra información que el auxiliar considere pertinente.

- Preparar, recibir y calificar tareas y exámenes.

- Acordar con el tutor del laboratorio la fecha en la que se presentará el proyecto final del curso.

2.6. Metodología

El contenido del laboratorio está distribuido en clases de 2 o 4 horas semanales y con excepción de las primeras sesiones (hasta la introducción del Cisco IOS para ser exacto), este debe ser presentado de manera que la teoría siempre sea respaldada y reforzada por ejercicios prácticos guiados, así como tareas relacionadas con el tópico de cada clase.

La creación y ponderación de exámenes, tareas y ejercicios, así como sus respectivas fechas de entrega, quedan a criterio del auxiliar del laboratorio, aunque más adelante se presentan sugerencias.

Como nota final de este apartado se sugiere que, tanto las tareas como investigaciones sean presentadas en formato digital para evitar la utilización de papel y así ayudar a proteger al medioambiente.

2.7. Contenido propuesto

El contenido que se propone para el laboratorio se expone de forma abreviada y desglosada a continuación.

2.7.1. Contenido abreviado

- Introducción al estudio de las redes y el modelo OSI.
- Introducción al modelo TCP/IP.
- TCP y UDP.
- *Ethernet.*
- Introducción al Cisco IOS.
- Configuración básica de un dispositivo.
- Subredes y superredes.
- *Dynamic host configuration protocol* (DHCP).
- Enrutamiento estático.
- Enrutamiento dinámico.
- *Open shortest path first* (OSPF).
- *Enhanced interior gateway routing protocol* (EIGRP).
- *Virtual* LANs (VLANs), enlaces troncales y *dynamic trunking protocol* (DTP).
- VLAN *trunking protocol* (VTP) e inter VLAN *routing.*
- *Spanning Tree protocol* (STP).
- *Access control lists* (ACLs).
- *Network address translation* (NAT).
- Introducción a la seguridad informática.
- Introducción a las redes inalámbricas.
- Introducción a IPv6.

2.7.2. Desglose del contenido

Introducción al estudio de las redes y el modelo OSI

- Introducción al estudio de las redes
- Red a pie (*sneakernet*)
- Clasificación de las redes según su extensión geográfica
- Partes que componen una red
- Ejemplos de aplicaciones que usan la red
- Topologías de red
- El modelo OSI

Introducción al modelo TCP/IP

- Relación entre el modelo OSI y el TCP/IP.
- Tipos de transmisión en IP versión 4 (IPv4).
- Tipos de direcciones en IP versión 4 (IPv4).
- Formato de una dirección IPv4.
- Dirección física, dirección lógica y necesidad de las mismas en una transmisión.
- Clases de direcciones IP por defecto.
- Direcciones públicas *versus* direcciones privadas.

TCP y UDP

- Diferencia entre TCP y UDP
- Funcionamiento de TCP
- Números de puerto

- Los números de puerto más conocidos

Ethernet

- Historia y evolución de *ethernet*
- Numeración de interfaces en equipo Cisco
- CSMA/CD
- Cables utilizados en *ethernet*
- Funcionamiento de un *switch*
- Modos de transmisión

Introducción al Cisco IOS

- Indicadores físicos
- ¿Qué es el Cisco IOS?
- Conexión a un dispositivo a través de una línea de comandos (CLI)
- Cómo obtener ayuda dentro del Cisco IOS
- Modos del Cisco IOS.
- Ayuda y edición en el Cisco IOS.

Configuración básica de un dispositivo

Subredes y superredes

- Máscara de subred
- Notaciones de la máscara de subred
- VLSM y CIDR
- *Subnetting*

Dynamic Host Configuration Protocol (DHCP)

- Proceso e implementación de DHCP
- DHCP *Relay*

Enrutamiento estático

- Propósito del enrutamiento
- Ruta por defecto

Enrutamiento dinámico

- Protocolos vector distancia *versus* estado de enlace
- Bucles de enrutamiento (*Routing loops*)
- Comportamiento *Classful* y *Classless*
- *Routing Information Protocol* (RIP)
- Autosumarización en la frontera discontinua
- Funcionamiento de la tabla de enrutamiento

Open shortest path first (OSPF)

- Historia
- Tablas mantenidas por OSPF
- Funcionamiento basado en áreas
- Requerimientos
- Vecindades y adyacencias.
- *Wildcard Mask*
- Tipos de red

- Sumarización de rutas

Enhanced interior gateway routing protocol (EIGRP)

- Historia
- Tablas mantenidas por EIGRP
- Sistemas autónomos

Virtual LANs (VLANs), enlaces troncales y *dynamic trunking protocol* (DTP)

- Modos de un puerto
- Enlaces troncales (*Trunks*)
- *Dynamic trunking protocol* (DTP)

VLAN *trunking protocol* (VTP) e inter VLAN *routing*

- VTP
- Inter VLAN *Routing*

Spanning Tree protocol (STP)

- Bucles de capa 2
- Operación de *Spanning Tree*
- Mejoras a *Spanning Tree*
- *Rapid Spanning Tree*
- Modelo jerárquico de 3 capas de Cisco

Access control lists (ACLs)

- Reglas de las listas de control de acceso
- Listas estándares
- Listas extendidas

Network address translation (NAT)

- NAT estático, dinámico y sobrecargado

Introducción a la seguridad informática

Introducción a las redes inalámbricas

Introducción a IPv6

2.8. Calendarización

Es importante que el laboratorio sea impartido en un período de dos horas a la semana. El número de clases y la duración de las mismas quedan a criterio de cada auxiliar.

A continuación, en la tabla I muestra una calendarización sugerida para el laboratorio que podría adaptarse, según la situación lo amerite.

Una descripción más detallada, así como propuestas de tareas y prácticas para cada clase se presentan en un capítulo posterior.

Tabla I. **Calendarización sugerida para el laboratorio**

	1er. Mes				2do. Mes				3er. Mes				4to. Mes				5to. Mes				6to. Mes			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Temas																								
Presentación de la clase																								
Introducción al estudio de las redes y el modelo OSI																								
Introducción al modelo TCP/IP																								
TCP y UDP																								
Ethernet																								
PRIMER EXAMEN PARCIAL																								
Introducción al Cisco IOS																								
Configuración básica de un dispositivo																								
Subredes y superredes																								
Dynamic host configuration protocol (DHCP)																								
Enrutamiento estático																								
Enrutamiento dinámico																								
Open shortest path first (OSPF)																								
Enhanced interior gateway routing protocol (EIGRP)																								
Virtual LANs (VLANs), enlaces troncales y DTP																								
VLAN *trunking protocol* (VTP) e Inter VLAN routing																								
Spanning Tree protocol (STP)																								
Access control lists (ACLs)																								
Network address translation (NAT)																								
Introducción a la seguridad informática																								
Introducción a las redes inalámbricas																								
Introducción a IPv6																								
EXAMEN FINAL																								
REPASO / REVISIONES / PROYECTO																								

Fuente: elaboración propia.

2.9. Bibliografía recomendada

Algunas de las normas técnicas que se pueden consultar son:

2.9.1. Principal

- Molenaar R. (s.f.) *How to Master CCNA.* Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en http://gns3vault.com/product/how-to-master-ccna-rs/

2.9.2. Secundaria

- Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1 - ICDN 2* [Videos]**.** Estados Unidos. Consultado el 9 de agosto de 2015 en http://www.cbtnuggets.com/

2.9.3. Complementaria

- Anderson A. & Benedetti R. (2009) *Head First Networking***.** Estados Unidos. O'reilly.

- Donahue G. (2011) *Network Warrior.* Estados Unidos. O'reilly.

- Lammle T. (2013) *CCNA Routing and Switching Study Guide*. Estados Unidos. Sybex.

3. MARCO TEÓRICO

3.1. Introducción al estudio de las redes

¿Qué es la red?

Al explicar el concepto de una manera asequible es simplemente un camino (o carretera) destinado a interconectar usuarios, máquinas y aplicaciones.

Por lo tanto, el estudio de las redes o del Networking trata acerca del diseño y construcción de estos caminos, cómo lograr que los mismos sean más amplios para facilitar el tráfico dentro de ellos y en qué puntos deben colocarse bloqueos, puestos de registro o garitas de peaje.

Figura 1. **La red es esencialmente un "camino"**

Fuente: elaboración propia, empleando *Edraw Max.*

Antes de continuar el estudio es importante introducir, por razones que se aclaran más adelante, a la empresa Cisco *Systems.*

Fundada en 1984, en San Francisco, Estados Unidos; ciudad de donde toma su nombre y logotipo (el puente Golden Gate); empresa que de ahora en adelante será referida solamente como Cisco, fue la creadora del primer *router* comercial exitoso y desde entonces se ha convertido en un referente en la industria, creando protocolos propietarios que a menudo sirven de inspiración o fundamento de estándares abiertos que regularmente aparecen unos años después.

Lo equipos de esta marca se caracterizan por presentar una experiencia final consistente e intuitiva y son utilizados para mostrar ejemplos prácticos en muchos textos académicos incluyendo este trabajo.

Otra característica distintiva de dicha empresa es una iniciativa educacional a nivel global conocida como la academia de networking de Cisco, la cual desglosa mucho del conocimiento existente en este campo dividiéndolo en varias especialidades, cada una de ellas estructurada en varios niveles (certificables por esta empresa) con el objetivo de presentar al educando una ruta de estudios a seguir, siendo los estratos inferiores aquellos donde se encuentran los temas básicos.

Es precisamente de uno de los niveles iniciales presentes en la especialidad de *routing & switching*, la de asociado, que se han tomado muchos de los temas más importantes para ser presentados en el marco de tiempo disponible para este curso, siendo la mayoría de ellos parte de la certificación conocida como CCNA R+S (*Cisco Certified Network Associate Routing and Switching*).

Como nota final se recomienda seguir los ejemplos presentados en este trabajo a través de un emulador de redes gratuito llamado GNS3 (*Graphical Network Simulator 3*).

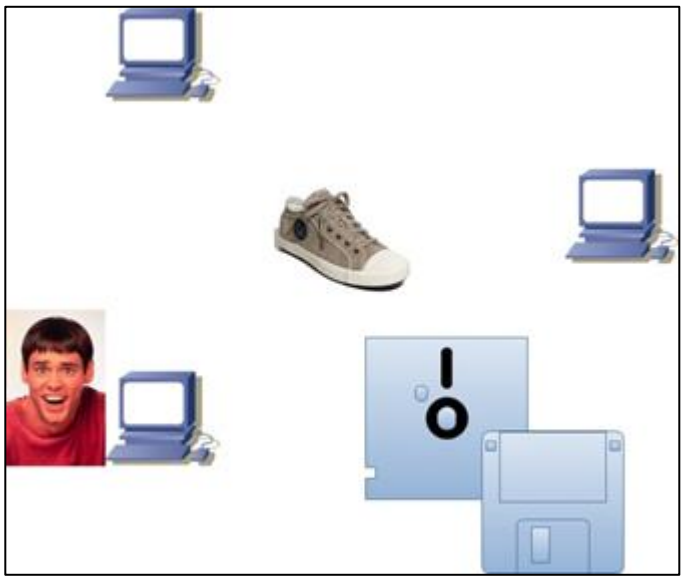
3.2. Partes de una red y modelo OSI

Las dos únicas partes del modelo con las que de hecho, interactúa el usuario son la primera es física, y la ultima de aplicación. La parte física abarca los aspectos físicos de la red (es decir, los cables, *hubs* y el resto de dispositivos que conforman el entorno físico de la red). La parte de aplicación proporciona la interfaz que utiliza el usuario en su computadora para enviar mensajes de correo electrónico o ubicar un archivo en la red.

3.2.1. Red a pie (*Sneakernet*)

En un principio, una red local de comunicaciones estaba compuesta por un grupo de ordenadores dispersos y aislados entre sí donde el intercambio de información debía ser llevado a cabo de forma manual, utilizando dispositivos de almacenamiento portátiles conocidos como *diskettes,* mismos que eran llevados de computadora a computadora por cada usuario, por lo que esta primera disposición recibió el nombre de red a pie o red del tenis (*Sneakernet*).

Figura 2. **Red del tenis o Sneakernet**



Fuente: elaboración propia, empleando *Edraw Max.*

La falta de comunicación directa entre cada dispositivo empezó a provocar fuertes pérdidas económicas dentro de las organizaciones, al impactar el desarrollo de sus actividades debido a la falta de control sobre los cambios realizados, duplicación de funciones, entre otros. Fue necesario crear un sistema capaz de proveer una conexión entre usuarios, lo que resultó en el desarrollo de las primeras propuestas para modelar el intercambio de información entre varios nodos, siendo estas OSI y TCP/IP.

3.2.2. Clasificación de las redes según su extensión geográfica

Acorde al área geográfica que abarcan las redes pueden ser clasificadas en una de las siguientes categorías.

- *Personal Area Network* (PAN): compuesta por aquellos dispositivos de uso personal y área de cobertura limitada proporcionada por computadoras portátiles, teléfonos celulares, entre otros.

- *Local Area Network* (LAN): formada por aquellos dispositivos que se encuentran dentro de una misma organización o bajo el mismo dominio administrativo.

- *Metropolitan Area Network* (MAN): aquella que se extiende a través de un área metropolitana.

- *Wide Area Network* (WAN): la que se extiende a través de países o continentes.

Es importante mencionar que en los últimos años, a causa del desarrollo de nuevas tecnologías, se ha producido un ofuscamiento de los antiguos límites geográficos impuestos sobre las redes de comunicaciones, por lo que las categorías mencionadas son utilizadas actualmente como una aproximación.

3.2.3. Partes que componen una red

La red está compuesta por una serie de dispositivos especializados y distintos medios de transmisión.

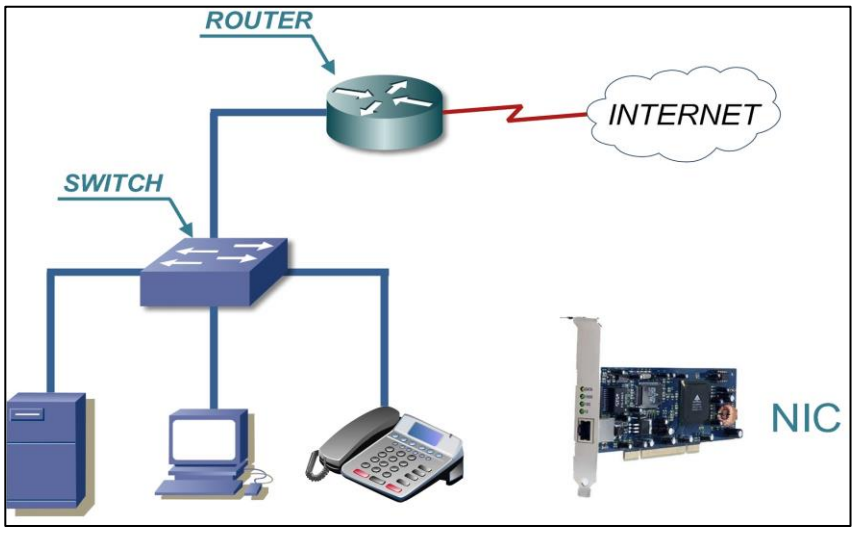
Todos los dispositivos finales deben conectarse a la red utilizando una tarjeta de interfaz de red (*Network Interface Card* (NIC)), desde donde se transmite la información a través de un cable de cobre o fibra óptica, usando electricidad o luz respectivamente, o de manera inalámbrica, conectando a los dispositivos a un concentrador de área local. Dicho concentrador posibilita la

comunicación entre dispositivos pertenecientes a una misma LAN y utiliza con propósitos de identificación, ciertas direcciones embebidas dentro de las NIC (conocidas como direcciones MAC) recibiendo el nombre de *switch.*

Para establecer conectividad con una red externa es necesario la utilización de un dispositivo capaz de encontrar una ruta a través de la cual pueda alcanzarse la misma y que sirva a los demás dispositivos como una puerta de enlace predeterminada (*default gateway*) a donde pueda enviarse toda aquella transmisión destinada a una red diferente.

El aparato que cumple la función de puerta de enlace hacia otras redes es conocido como *router*, el cual se vale de las direcciones proporcionadas por el protocolo de internet o direcciones IP (internet *protocol*), para encontrar la mejor ruta hacia un destino en particular.

Figura 3. **Partes de una red**



Fuente: elaboración propia, empleando *Edraw Max*

3.2.4. Topologías de red

El término “topología” utilizado en este ámbito hace referencia a la manera en que se conectan físicamente los dispositivos dentro de una red.

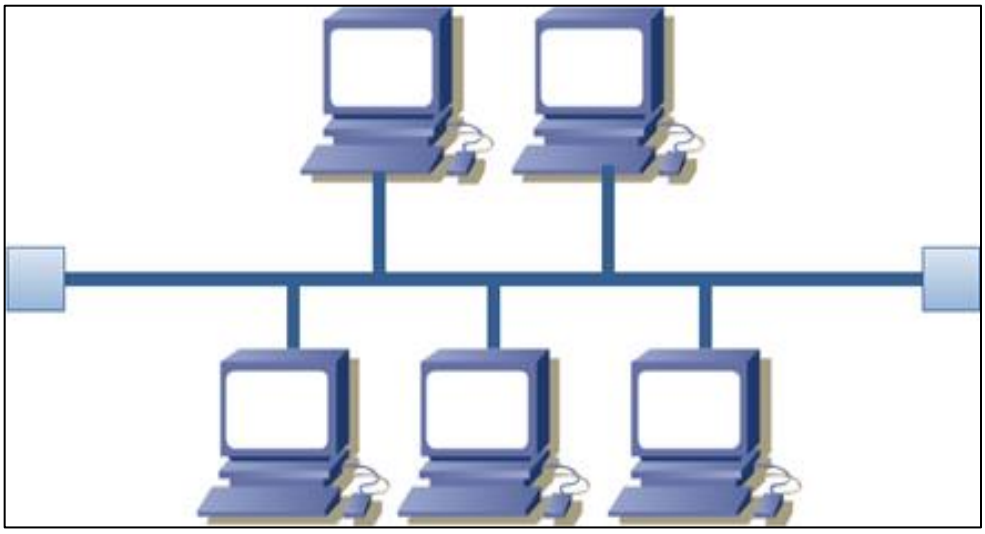
Algunas de las topologías más conocidas se presentan a continuación.

3.2.4.1. Topología de bus

Es una de las disposiciones más antiguas, donde todos los ordenadores estaban conectados a un solo medio compartido, siendo este un tipo de cable coaxial con terminadores de señal conectados a sus extremos.

Esta topología en particular se caracterizaba por su gran inestabilidad.

Figura 4. **Topología de bus**

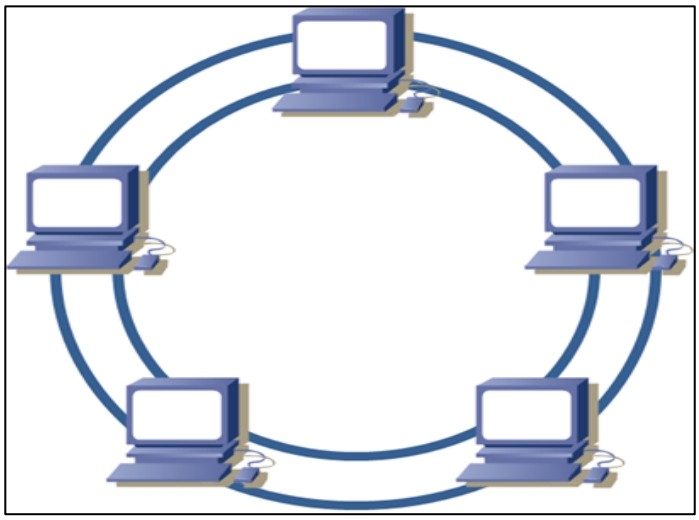


Fuente: elaboración propia, empleando *Edraw Max*.

3.2.4.2. Topología de anillo

Esta disposición es conocida por ser el arreglo utilizado por *token ring*, una arquitectura de red creada por IBM en 1970 y en donde el intercambio de información era posible a través de la transmisión de un testigo o *token* entre cada una de las computadoras, siendo la poseedora del mismo la única capaz de comunicarse en un momento dado.

Figura 5. **Topología de doble anillo**

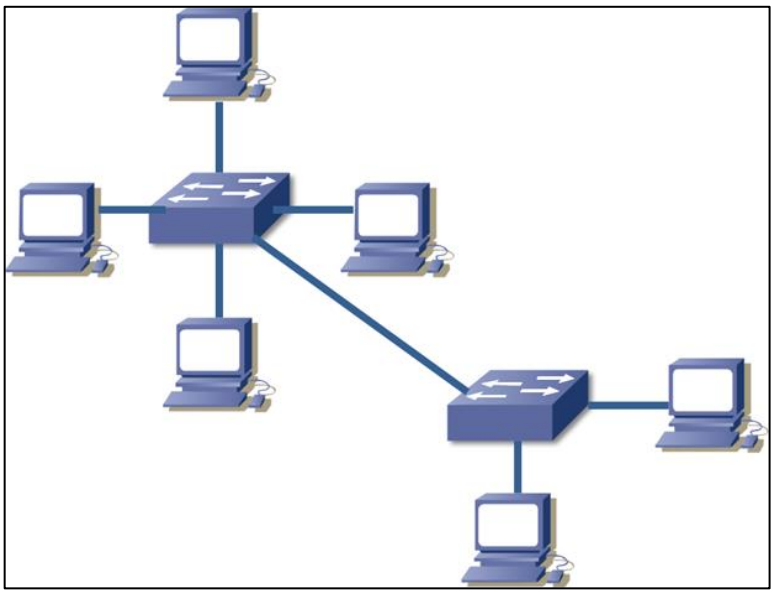


Fuente: elaboración propia, empleando *Edraw Max*.

3.2.4.3. Topología en estrella o estrella extendida

Es la topología utilizada actualmente en las redes de área local (LAN), en donde varios dispositivos están conectados a un solo concentrador, siendo posible extender la red al conectar el mismo a otros concentradores.

Figura 6. **Topología en estrella extendida**



Fuente: elaboración propia, empleando *Edraw Max*.

3.2.5. El modelo OSI

Creado por la ISO (International Organization for Standardization), el modelo OSI (Open Systems Interconnection) fue originalmente una suite (o conjunto) de protocolos destinados a estandarizar la comunicación entre dispositivos de diferentes tipos, ya que no era posible en ese momento, y que se encontraba en competencia con el modelo TCP/IP creado por el Departamento de la Defensa de los Estados Unidos (Department of Defense (DoD)).

Con muchos países y empresas como respaldo, el modelo OSI se enfocó en la seguridad y en la escalabilidad de las redes de comunicaciones, siendo llamado en su momento como el modelo del futuro. Sin embargo, debido a la lentitud de su desarrollo, lo complejo de su implementación, el excesivo número

de direcciones propuestas (para los requerimientos de esa época) y a su costo, OSI finalmente perdió la batalla contra TCP/IP, siendo los protocolos propuestos por este último modelo los utilizados actualmente.

No obstante, la iniciativa que dio origen al modelo OSI trajo consigo muchos beneficios al separar lógicamente las operaciones necesarias para el funcionamiento de una red, facilitando su entendimiento y permitiendo la especialización en ciertas áreas por parte de los fabricantes.

Por dichas razones, el modelo OSI sigue utilizándose tanto de una manera académica para introducir nuevos educandos al campo, así como un marco de referencia para clasificar dispositivos o buscar problemas dentro de una red, siendo referido, irónicamente, de manera más usual que TCP/IP.

El modelo OSI consta de siete capas, cada una con una función definida que aporta su propia información a la transmisión (encapsulamiento).

- Capa 7 - Capa de aplicación: es la más familiar al usuario final, provee acceso a la red a las aplicaciones. (ej.: un navegador web).

- Capa 6 - Capa de presentación: esta capa convierte la información a manera que pueda ser usada por la aplicación a la que está destinada y provee servicios de encriptación y desencriptación.

- Capa 5 - Capa de sesión: maneja las conexiones entre dispositivos y ayuda a mantener separados los distintos flujos de información.

- Capa 4 - Capa de transporte: establece el tipo de comunicación confiable o no confiable, y provee una manera de distinguir entre varios procesos utilizando direcciones de transporte conocidas como puertos.

- Capa 3 - Capa de red: provee direccionamiento lógico a través de las direcciones del internet *protocol* o direcciones IP.

- Capa 2 - Capa de enlace: provee direccionamiento físico. En *ethernet,* a través de las direcciones MAC.

- Capa 1 - Capa física: se encarga de la transmisión de datos en bits.

3.3. Introducción al modelo TCP/IP

El modelo TCP/IP fue desarrollado en 1970 por Vinton Cerf y Robert Kahn bajo el auspicio del Departamento de Defensa de los Estados Unidos para poder establecer comunicación entre varios sistemas, cuya interconexión evolucionaría más tarde a lo que hoy es conocido como internet.

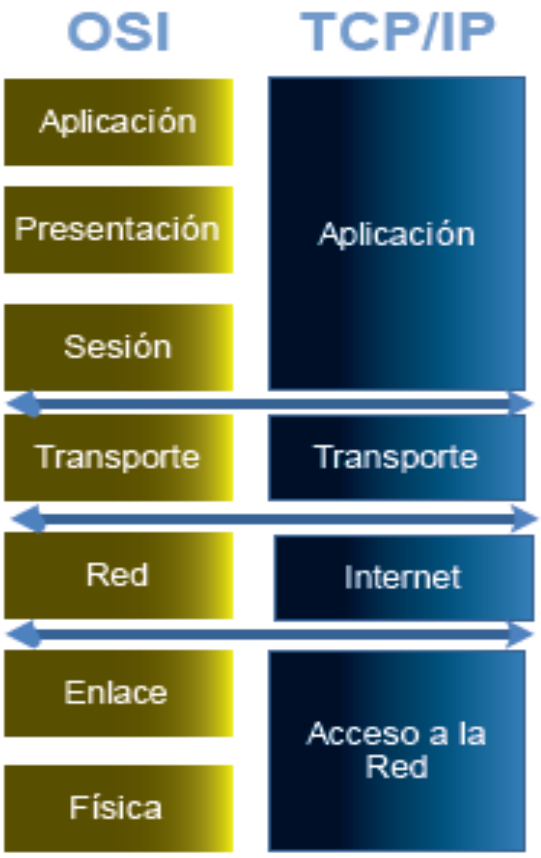
La investigación de Cerf y Kahn empleaba originalmente un solo protocolo medular conocido como Transmission Control Program, siendo su funcionalidad dividida posteriormente en dos protocolos más modulares y que recibieron los nombres de Transmission Control Protocol (TCP) e Internet Protocol (IP) los cuales se convirtieron en la piedra fundamental del modelo y le dieron nombre al mismo al ser publicados en 1980 como TCP/IP versión 4.

A pesar de que su nombre hace referencia a sus dos protocolos más importantes, TCP/IP es referido usualmente como una familia o una *suite* de

protocolos, esto indica que es un conjunto de los mismos, clasificados acorde a su funcionalidad en alguna de las capas de este modelo.

Para finalizar este apartado es necesario mencionar que pese a que TCP/IP es el motor de la comunicación de las redes modernas sin embargo, tanto profesionales como estudiantes utilizan el modelo OSI como referencia habitual en sus respectivos ámbitos. A continuación se presenta la relación entre ambos modelos.

Figura 7. **Relación entre los modelos OSI y TCP/IP**



Fuente: elaboración propia, empleando *Edraw Max*

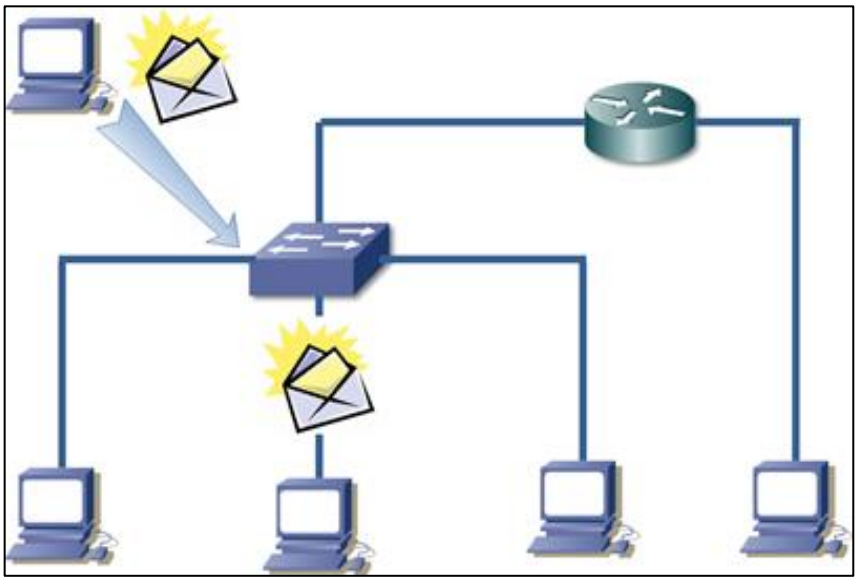
3.3.1. Tipos de transmisión en IP versión 4 (IPv4)

En IPv4 existen 3 tipos diferentes de transmisión: *unicast, broadcast* y *multicast.*

3.3.1.1. *Unicast*

Es una transmisión desde un único emisor hacia un único receptor. En el ejemplo de la figura 8 se muestra una comunicación de un *host* hacia otro *host.* Donde "*host*" será el apelativo utilizado para referirse a cualquier dispositivo final capaz de mantener una dirección IP como un computador o un teléfono inteligente.

Figura 8. ***Unicast*, transmisión de "uno a uno"**

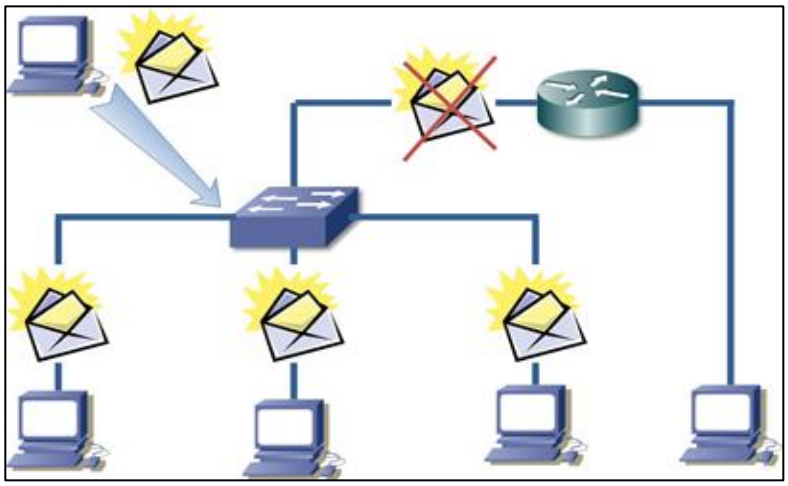


Fuente: elaboración propia, empleando *Edraw Max*.

3.3.1.2. *Broadcast*

Es una transmisión que se realiza desde un único emisor hacia todos los demás dispositivos existentes dentro de la misma red, siendo limitado por defecto por dispositivos de capa 3, lo que significa que los *routers* no transmiten este tipo de tráfico de una red hacia otra.

Figura 9. ***Broadcast,* transmisión de "uno a todos" dentro de una misma red**

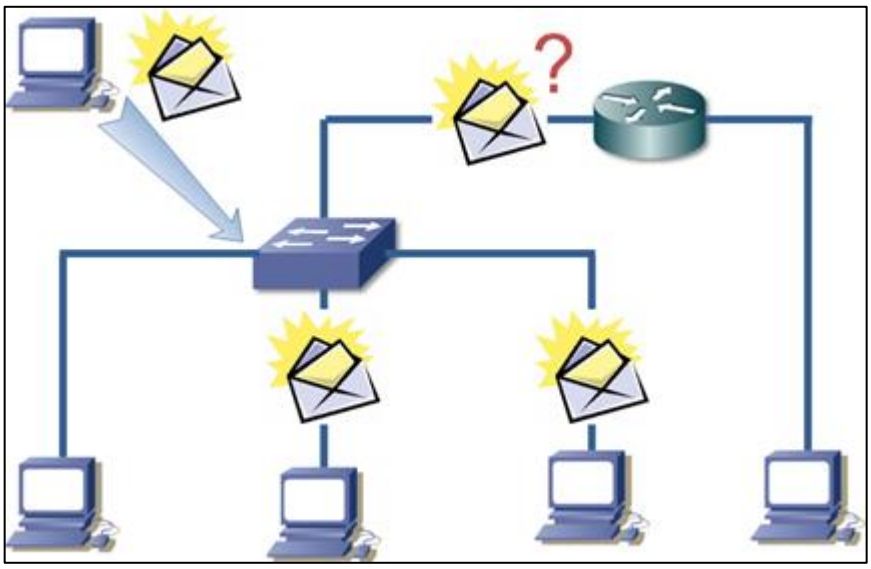


Fuente: elaboración propia, empleando *Edraw Max*.

3.3.1.3. *Multicast*

Es una transmisión de un único emisor hacia un grupo de dispositivos en particular. Su propósito consiste en evitar colocar la misma información varias veces sobre el mismo medio para lograr así una red más eficiente. El alcance de este tipo de tráfico puede ser limitado según el caso.

Figura 10. ***Multicast*, transmisión de "uno a un grupo"**



Fuente: elaboración propia, empleando *Edraw Max*.

3.3.2. Tipos de direcciones en IP versión 4 (IPv4)

Dentro de IPv4 existen varios tipos de direcciones, algunas están reservadas específicamente para un uso especial, mientras que otras pueden ser asignadas a conveniencia. Dichos tipos son:

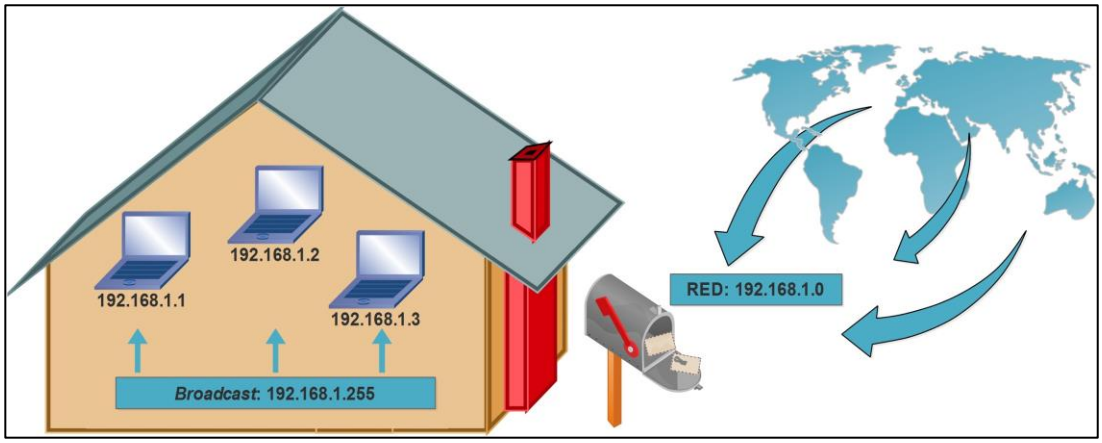
- Dirección de *host*: es aquella adjudicada a cada uno de los dispositivos finales con el fin de identificarlos.

- Dirección de red: destinada a englobar todas las direcciones individuales (o de *host*) presentes en una red a manera que las redes externas puedan ver como un todo (o a través de una sola dirección) a los dispositivos que la componen.

- Dirección de *broadcast*: es la dirección utilizada dentro de cada red para enviar un mensaje destinado a todos los miembros de la misma.

- Dirección de *multicast*: es la dirección utilizada para enviar información a un grupo específico de dispositivos.

Figura 11. **Direcciones de *host,* red y *broadcast***



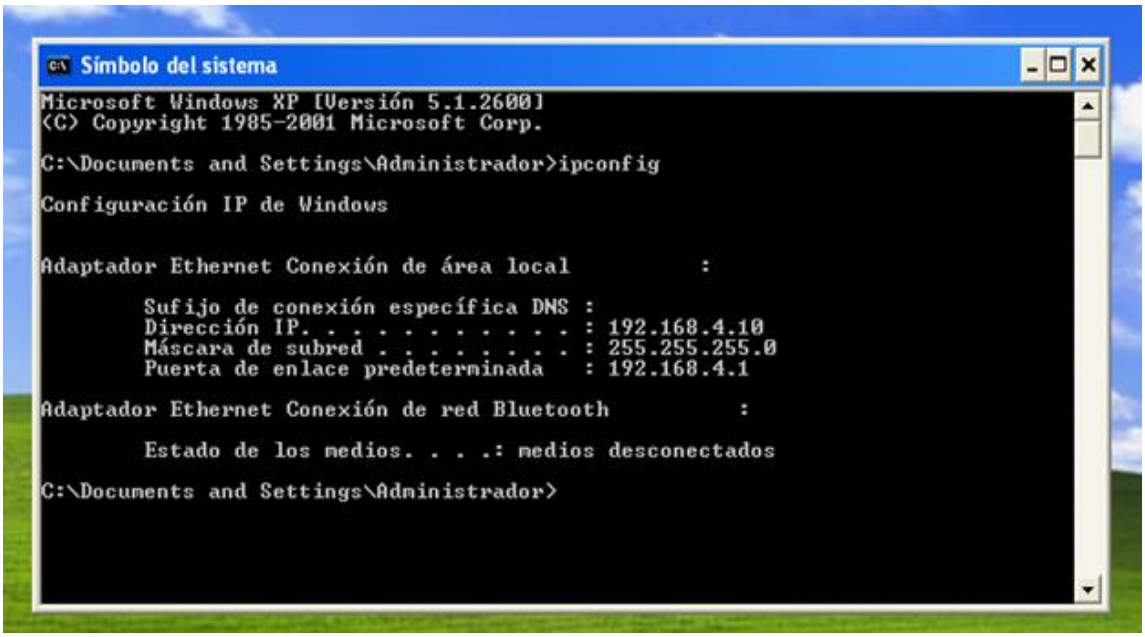
Fuente: elaboración propia, empleando *Edraw Max*.

3.3.3. Formato de una dirección IPv4

Una dirección IPv4 está compuesta por cuatro octetos de dígitos binarios, separados por puntos y escritos en formato decimal por conveniencia.

Para mostrar la configuración relacionada con IP en un dispositivo ejecutando Windows como sistema operativo puede emplearse la herramienta *ipconfig.*

Figura 12. **Captura de pantalla del comando *ipconfig* en Windows XP**



Fuente: elaboración propia.

En la salida de dicha aplicación se puede apreciar la dirección IP de esa máquina en particular, la dirección de su puerta de enlace predeterminada (la dirección IP asignada a la interfaz de un *router* conectada a la misma red) y un nuevo parámetro conocido como la máscara de subred.

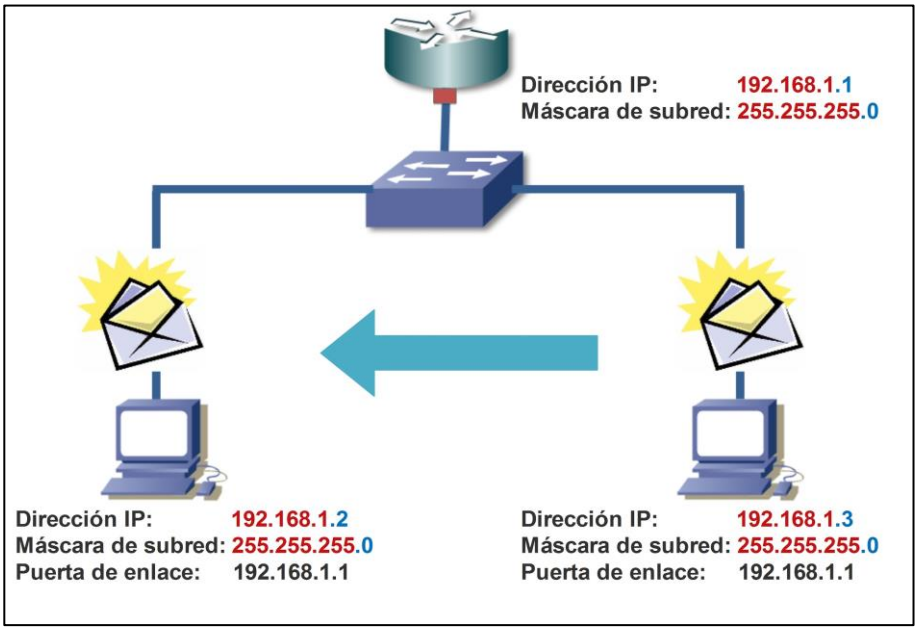
Empleando dicha máscara junto con una operación matemática, un *host* puede conocer la red a la que este pertenece, la dirección de *broadcast* utilizada y reconocer cuando una transmisión está dirigida a una red externa.

La máscara de subred se divide en dos partes, a través de las cuales puede determinar, dada una dirección IP, tanto la dirección de red como el identificador individual de cada dispositivo, siendo los bits con un valor de uno utilizados para reconocer la red y aquellos con un valor de cero empleados para

revelar el identificador de cada *host* (nótese que el número decimal 255 está compuesto por 8 bits con un valor de uno).

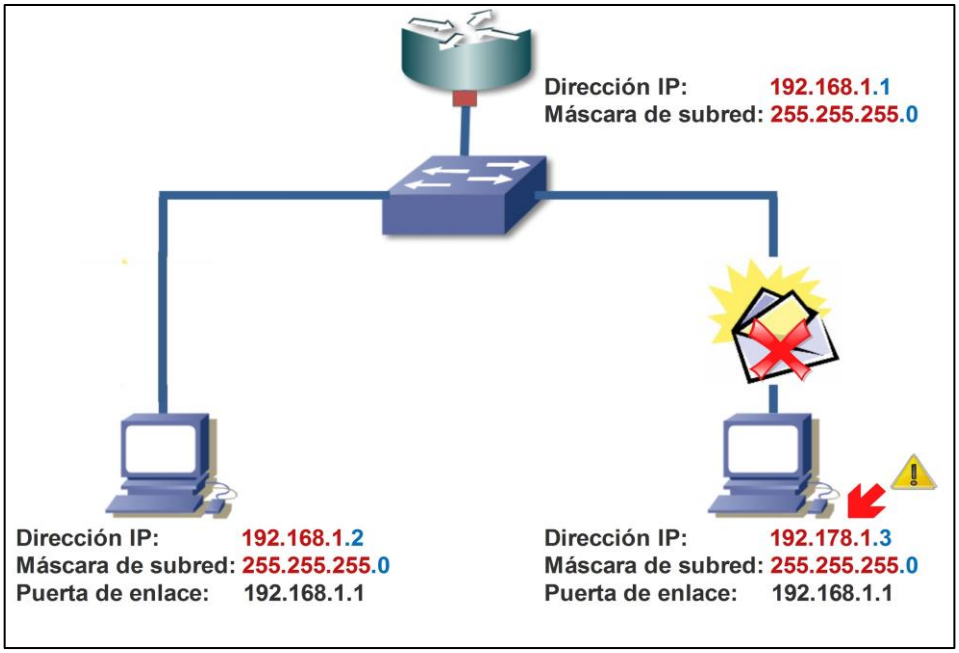
Para que un ordenador pueda alcanzar a otro *host* por si solo, ambos dispositivos deben encontrarse dentro de la misma red, condición que debe cumplirse también entre cualquier dispositivo final y su puerta de enlace predeterminada.

Figura 13. **Transmisión exitosa entre dos computadoras que se encuentran en la misma red**



Fuente: elaboración propia, empleando *Edraw Max*.

Figura 14. **Transmisión fallida entre dos computadoras que se encuentran configuradas en una red distinta**



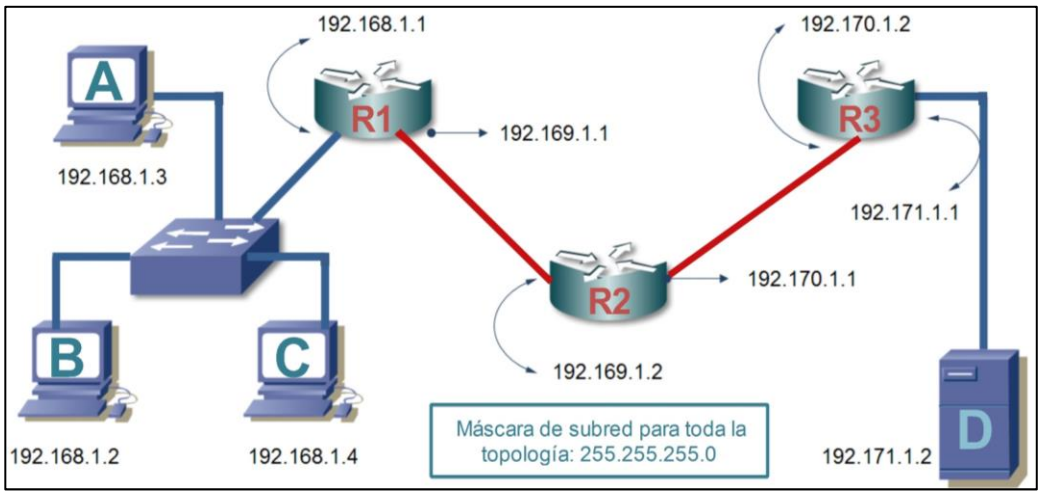
Fuente: elaboración propia, empleando *Edraw Max*.

Como nota final de este apartado obsérvese que para poder comunicarse de una red a otra, un *host* debe tener configurado, por los menos una dirección IP, una máscara de subred y la dirección de su puerta de enlace predeterminada.

3.3.4. Dirección física, dirección lógica y necesidad de las mismas en una transmisión

El orden de presentar los temas de esta sección se introduce la siguiente topología compuesta por varias redes conectadas entre sí.

Figura 15. **Varias redes conectadas entre sí utilizando tres *routers***



Fuente: elaboración propia, empleando *Edraw Max*.

Como se ha explicado anteriormente, el dispositivo empleado para establecer conectividad entre redes dispares es el *router,* por lo que puede ser considerado (simplificando su funcionamiento) como un delimitador (o límite) entre las mismas, siendo esta la razón por la cual todas las interfaces pertenecientes a un solo *router* deben encontrarse necesariamente en una red diferente para que el uso de este aparato tenga sentido.

Partiendo de este último concepto es posible apreciar que la topología anterior está compuesta por 4 redes (2 de ellas proveyendo conexión directa entre *routers,* por lo que son consideradas “de punto a punto”), hecho que puede apreciarse al hacer el análisis de las direcciones asignadas a cada interfaz y su respectiva máscara de subred (255.255.255.0 para toda la topología).

Una vez establecidos los límites de cada red, se procede a examinar los mecanismos necesarios para realizar una transmisión de extremo a extremo entre varias redes, en donde es necesario no solo encontrar la mejor ruta entre ellas, sino también hallar una forma de trasladar la información de un dispositivo hacia otro de forma sucesiva y en orden, para llevarla a su destino final; utilizando para ello esquemas de direccionamiento tanto físicos como lógicos.

Tomando en consideración los modelos organizados en capas presentados anteriormente y la naturaleza modular de los mismos, no es sorpresa encontrar que dentro de cada capa pueden utilizarse distintos protocolos y esquemas de direccionamiento, dependiendo de la tecnología que se esté utilizando, siendo aquella conocida como *ethernet* la que se usará durante la mayor parte de este estudio.

El direccionamiento lógico (capa 3 del modelo OSI) es el encargado de encontrar la mejor ruta para una transmisión de extremo a extremo, siendo las direcciones IP las únicas empleadas actulamente para cumplir este propósito.

Por otro lado, el direccionamiento físico (capa 2 del modelo OSI) es el encargado de llevar la información de un dispositivo hacia otro de manera sucesiva siguiendo la línea establecida por la mejor ruta, utilizándose en el caso de *ethernet* las direcciones de control de acceso al medio, mejor conocidas como direcciones MAC, las cuales son identificadores de 48 bits embebidos en las tarjetas de red.

Si bien es necesario contar tanto con el direccionamiento lógico (direcciones IP origen/destino) como con el direccionamiento físico (direcciones MAC origen/destino), de manera regular los dispositivos solo cuentan con el primero a la hora de iniciar una transmisión, por lo que es necesario introducir

un nuevo protocolo capaz de encontrar una dirección física a partir de una lógica, siendo este llamado *address resolution protocol* (ARP).

Este último protocolo se propaga como un *broadcast,* por lo que su funcionamiento se encuentra limitado a encontrar la dirección MAC asociada a una dirección IP dentro de una misma red.

Si la transmisión está dirigida hacia otro dispositivo perteneciente a la misma LAN, ARP resolverá la dirección física del *host* en cuestión, mientras que si la transmisión está dirigida hacia una red externa ARP retornará la dirección física de la puerta de enlace predeterminada, quien luego establecerá el siguiente dispositivo a dónde debe ser enviada la transmisión para que arribe correctamente a su destino.

Retomando el ejemplo presentado al inicio de este tema, se considera una transmisión desde PC-A hacia el servidor-D.

PC-A inicia la comunicación hacia el servidor-D, utilizando las direcciones IP asignadas a estos dispositivos como las direcciones lógicas de origen y destino, mismas que nunca cambiarán durante el tránsito de la información (192.168.1.3/192.171.1.2).

No obstante, las direcciones presentes en los extremos de la comunicación ya han sido definidas, todavía es necesario determinar aquellas direcciones necesarias para llevar la información desde PC-A hacia el primer nodo presente en la topología (llamado primer salto), siendo este la interfaz de R1 que le sirve como puerta de enlace.

Para cumplir con este objetivo, PC-A realiza una petición ARP a través de la cual puede descubrir la dirección física de la interfaz del *router* requerida, determinando la combinación de direcciones físicas de origen/destino necesaria (MAC PC-A/MAC R1).

Una vez el paquete de información arribe a R1, este utilizará las direccione IP para encontrar la mejor ruta y sobreescribirá las direcciones físicas para llevar la información al siguiente dispositivo (MAC R1/MAC R2), proceso que se repetirá hasta alcanzar el destino de la comunicación.

3.3.5. Clases de direcciones IP por defecto

En un principio, el espacio de direcciones IP se dividió en varias clases de la siguiente manera (la máscara de subred se añadió algún tiempo después):

- Clase A
  1.er octeto (1 - 126)
  Máscara de subred: 255.0.0.0
  Número de direcciones de *host*: más de 16 millones por cada red
  Ejemplo: 10.0.0.1

- Clase B
  1.er octeto (128 -191)
  Máscara de subred: 255.255.0.0
  Número de direcciones de *host*: más de 65 mil por cada red
  Ejemplo: 172.16.0.1

- Clase C
  1.er octeto (192-223)

Máscara de subred: 255.255.255.0

Número de direcciones de *host*: 254 por cada red

Ejemplo: 192.168.0.1

- Clase D

  1.er octeto (224-239)

  Multicast

- Clase E

  1.er octeto (240-255)

  Investigación

3.3.6. Direcciones públicas y direcciones privadas

Con el crecimiento exponencial de dispositivos que estaban siendo conectados a la red en la década de 1990 pronto se hizo evidente que el esquema de direccionamiento propuesto por IPv4 no se daría abasto para proveer de una dirección a cada uno de ellos.

Una de las primeras medidas que se tomaron para ralentizar el agotamiento de dichas direcciones fue reservar un grupo de las mismas para su uso exclusivo en redes internas dando lugar a la división de las direcciones IP como públicas y privadas, esta comparación se muestra en la tabla II.

Tabla II. **Cuadro comparativo de las direcciones públicas y privadas**

Direcciones públicas	Direcciones privadas
• Usables en el internet y redes internas • Otorgadas por la ICANN • Tienen un costo • Direcciones únicas en el mundo	• Usables solamente en redes internas • No son enrutables en internet • Gratuitas • Direcciones únicas solamente dentro de una entidad (pueden repetirse)

Fuente: elaboración propia.

Para su uso privado se reservaron ciertas direcciones dentro de cada una de las primeras tres clases presentadas como se muestra en la siguiente tabla III.

Tabla III. **Direcciones privadas**

Clase A :	10.x.x.x
Clase B :	172.16.x.x hasta 172.31.x.x
Clase C :	192.168.x.x
	x = 0 a 255

Fuente: elaboración propia.

Otras direcciones reservadas para un propósito especial y, que tampoco son enrutables en internet son las direcciones de *loopback* destinadas para realizar ciertas pruebas y las direcciones de autoconfiguración, que son asignadas cuando el dispositivo no puede obtener una dirección IP válida.

Tabla IV. **Otras direcciones reservadas no enrutables en internet**

Direcciones de *loopback* para pruebas 127.x.x.x
Direcciones de auto configuración 169.254.x.x
x = 0 a 255

Fuente: elaboración propia.

3.4. TCP y UDP

TCP (*transmission control protocol*) y UDP (*user datagram protocol*) son los dos protocolos principales de la capa de transporte del modelo OSI.

TCP está orientado a la conexión y a la recuperación de errores (a través de acuses de recibo o *acknowledgements* -ACKs-), por lo que es utilizado por la mayoría de aplicaciones, mientras que UDP fue creado para transmisiones de mejor esfuerzo y es utilizado para comunicaciones en tiempo real.

Las características principales de ambos protocolos se enlistan en la tabla V.

Tabla V. **Cuadro comparativo TCP y UDP**

TCP	UDP
• Crea conexiones (*3-way handshake*) • Usa números de secuencia • Confiable (usa *ACKs*)	• No crea conexiones • Transmisiones de mejor esfuerzo • No confiable

Fuente: elaboración propia.

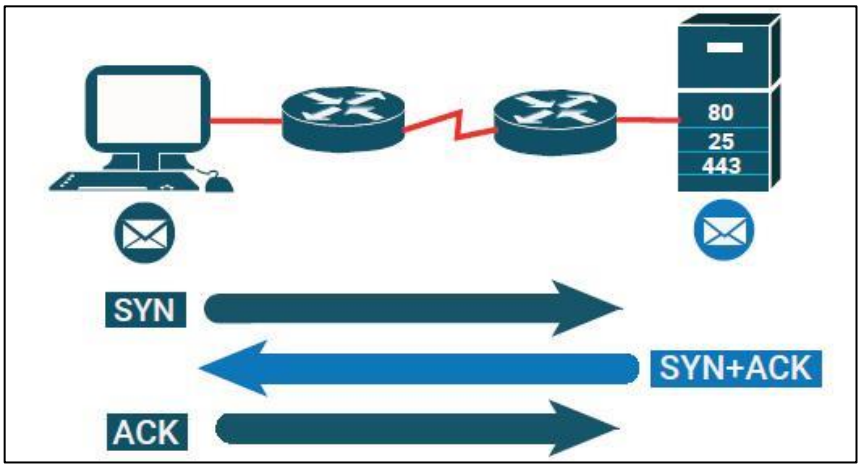
3.4.1. Funcionamiento de TCP

Hace posible enlazar cualquier tipo de computadoras, sin importar el sistema operativo que usen o el fabricante. este protocolo fue desarrollado originalmente por el arpa (advanced research projects agency) del Departamento de Defensa de los Estados Unidos.

3.4.1.1. Intercambio de tres vías (3-Way handshake)

Es un intercambio de tres pasos realizado para iniciar una transmisión confiable (TCP).

Figura 16. **Intercambio de tres vías (3-*Way handshake*)**



Fuente: elaboración propia, empleando *Edraw Max*.

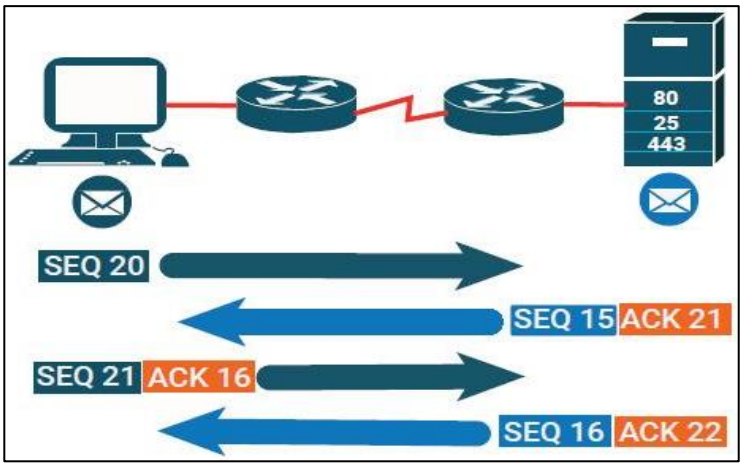
El ejemplo de la figura 16, la computadora envía una solicitud de sincronización (1. SYN) que es respondida por el servidor con un acuse de recibo para indicar que esta ha sido recibida correctamente, al mismo tiempo

que envía su propia información de sincronización (2. SYN + ACK). Si la transmisión es recibida correctamente, por parte de la computadora, esta responderá con un último acuse de recibo (3. ACK) y se dará inicio a la transferencia de datos.

3.4.1.2. Números de secuencia y acuses de recibo

TCP utiliza números de secuencia para poner los paquetes en orden y acuses de recibo o *acknowledgements*, (ACKs) para asegurar una transmisión confiable.

Figura 17. **Números de secuencia y acuses de recibo**



Fuente: elaboración propia, empleando *Edraw Max*.

El ejemplo de la figura 17, la computadora (azul) y el servidor (celeste) intercambian información. Nótese que cada dispositivo usa sus propios números de secuencia (no relacionados) y que ambos notifican a la otra parte

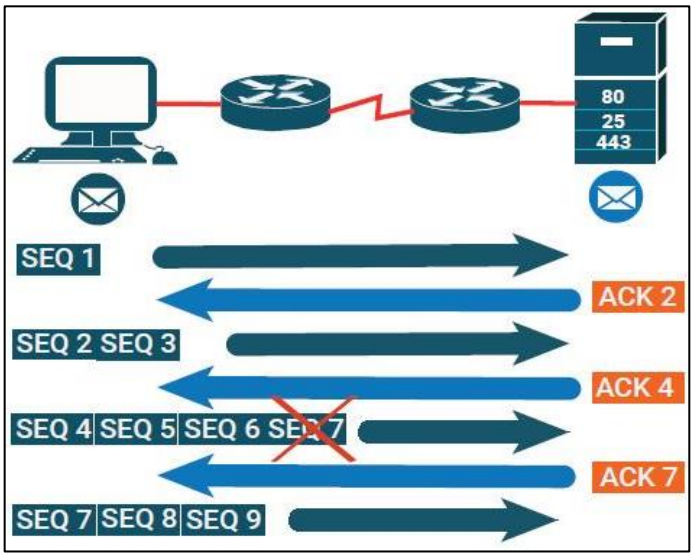
de la correcta recepción de los datos con un ACK (naranja) en donde se pide el siguiente paquete en orden ascendente.

3.4.1.3. Tamaño de ventana y ventana deslizante

Se refiere a la cantidad de información enviada en cada transmisión y a la variabilidad de la misma.

TCP siempre intentará enviar la mayor cantidad posible de información y se valdrá de los acuses de recibo (ACK) para determinar la cantidad óptima de segmentos que pueden transmitirse sin errores en un momento dado.

Figura 18. **Tamaño de ventana y ventana deslizante**



Fuente: elaboración propia, empleando *Edraw Max*.

El ejemplo de la figura 18, se observa que el tamaño de ventana se incrementa hasta poder enviar 4 segmentos, al mismo tiempo cuando el segmento #7 se pierde en la transmisión, lo que provoca la solicitud del segmento perdido y una reducción en el tamaño de la ventana.

3.4.2. Números de puerto

Un puerto es una conexión lógica que puede ser usada entre programas para intercambiar información directamente.

Los números de puerto son asignados de diferentes maneras basados en 3 rangos:

- Puertos “bien conocidos” o puertos de sistema (0-1023)
- Puertos de usuario (1024-49151)
- Puertos privados o dinámicos (49152-65535)

Algunos de los puertos más comunes se presentan en la tabla VI.

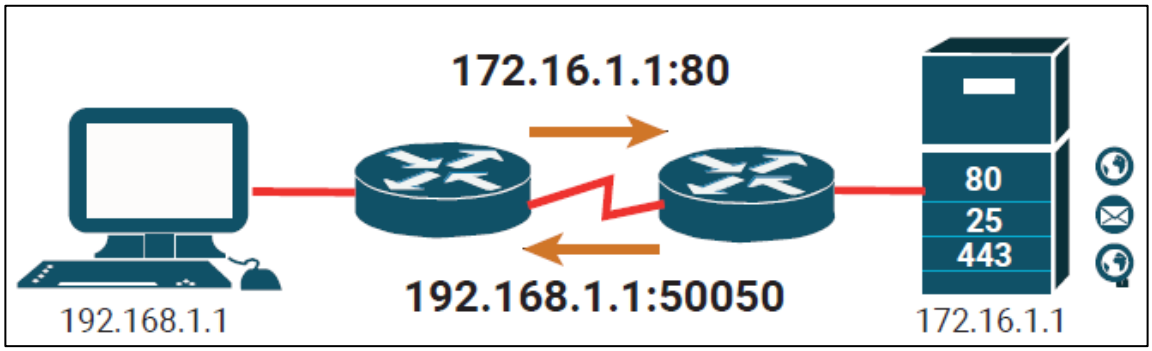
Tabla VI. **Puertos más comunes**

TCP <0 - 65535>	UDP <0 - 65535>
21 -> FTP 22 -> SSH 23 -> Telnet 25 -> SMTP 53 -> DNS 80 -> HTTP 110 -> POP3 443 -> HTTPS	53 -> DNS 69 -> TFTP

Fuente: elaboración propia.

A la combinación de una IP y un número de puerto se le llama *socket,* en el ejemplo que se presenta en la figurta 19, se aprecia la interacción de un servidor web usualmente asociado con el puerto 80 y una aplicación del usuario.

Figura 19. ***Socket***



Fuente: elaboración propia, empleando *Edraw Max*.

3.5. *Ethernet*

Es el estándar de red más popular del mundo, desarrollado por Robert Metcalfe, cuando trabajará para Xerox en 1973, su nombre proviene de la combinación de las palabras "*ether*" y "*network*" haciendo alusión a la capacidad de este protocolo de conectar dispositivos en una red independientemente del fabricante.

Se vuelve un estándar entre vendedores en 1982, a una velocidad de 10 Mbps. Desde entonces la velocidad ha ido aumentando con el transcurso de los años: *FastEthernet* (100 Mbps), *GigabitEthernet* (1000 Mbps), *10-GigabitEthernet* (10000 Mbps) y *100-GigabitEthernet* (100000 Mbps).

3.5.1. Numeración de interfaces en equipo Cisco

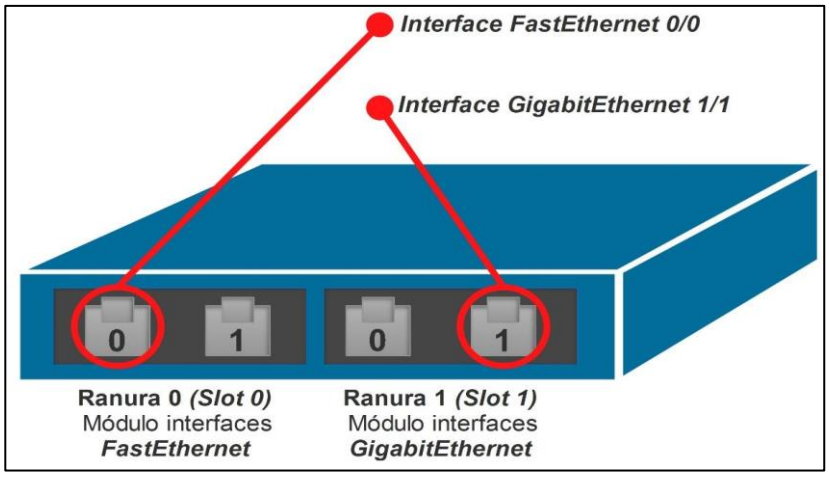
De manera general, la numeración utilizada en los dispositivos Cisco sigue el formato Tipo Ranura/Puerto, como se muestra en la tabla VII.

Tabla VII. **Numeración utilizada en los dispositivos Cisco**

Tipo	Ranura (*Slot*)	Puerto
- *Ethernet* - *FastEthernet* - *Serial*	El número de la ranura donde puede insertarse un módulo, comenzando en cero.	El número de puerto de un módulo específico comenzando en cero.

Fuente: elaboración propia.

Figura 20. **Numeración de interfaces en equipo Cisco**



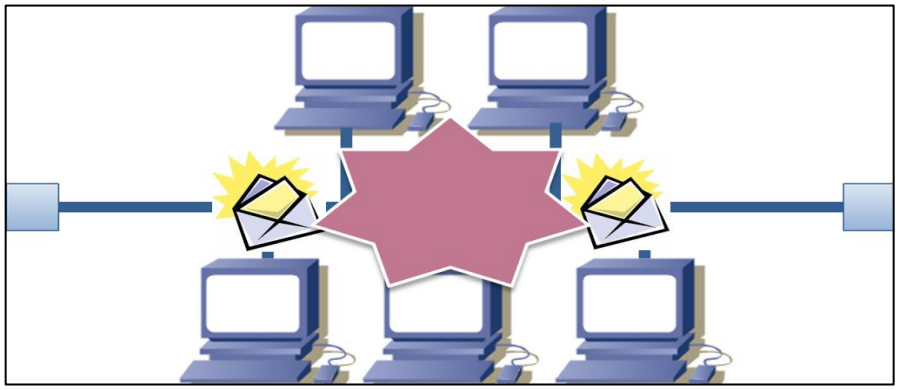
Fuente: elaboración propia, empleando *Edraw Max*.

3.5.2. Colisiones

Al ser una tecnología de múltiple acceso *ethernet* es susceptible al problema de las colisiones.

Una colisión se produce cuando dos paquetes se encuentran al mismo tiempo en un medio compartido causando que estos choquen y se destruyan (ver figura 21).

Figura 21. **Colisiones**



Fuente: elaboración propia, empleando *Edraw Max*.

Aquellos sectores de una red en donde una colisión puede ocurrir en cualquier punto reciben el nombre de dominios de colisión. Por ejemplo, muchos equipos conectados a un *hub* (un simple repetidor) constituyen un solo dominio de colisión, mientras que cada puerto de un *switch* es su propio dominio.

3.5.3. Carrier sense multiple access collision detection (CSMA/CD)

Es el conjunto de reglas que regulan la transmisión en una red *ethernet*, en donde:

- *Carrier* = señal en una red.

- *Sense* = sentir o detectar.

- *Multiple Access* = muchos dispositivos pueden acceder al mismo tiempo.

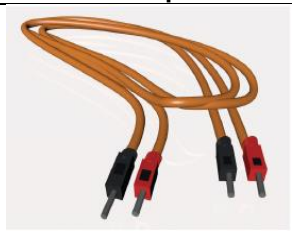
- Collision = cuando paquetes chocan y se destruyen en un medio compartido.

- Detection = cómo manejan las computadoras las colisiones cuando son detectadas.

3.5.4. Cables utilizados en *ethernet*

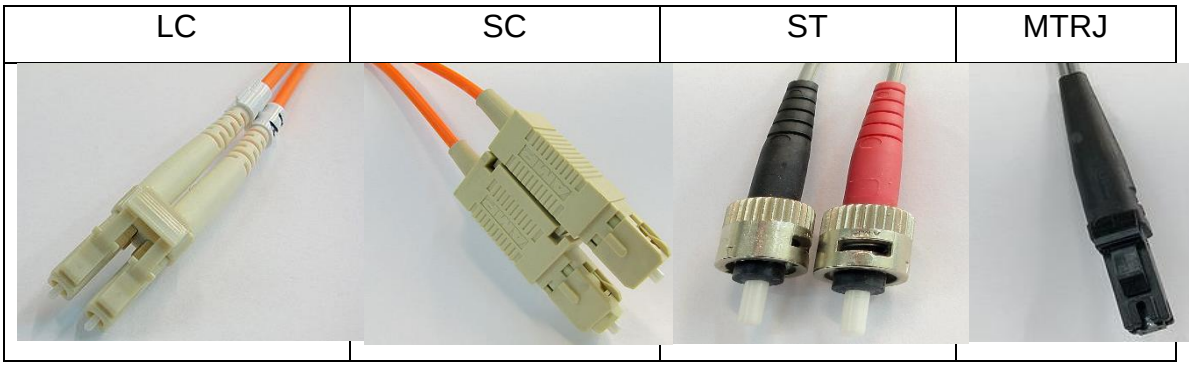
Los cables que se emplean junto a *ethernet* son el cable de par trenzado no blindado (*Unshielded Twisted Pair* (UTP)), hecho de cobre y fibra óptica, compuesta por uno o varios hilos de fibra de vidrio.

Figura 22. **Cables utilizados en *ethernet***

Cable de par trenzado no blindado (UTP) Distancia máx. : **100 m** Conector : **8P8C (RJ-45)**	Fibra multimodo Distancia máx. : **275 m a unos kilómetros** Conector: **variable** Color del forro: **aqua o naranja**	Fibra monomodo Distancia máx. : **2,5 km a muchos kilómetros** Conector: **variable** Color del forro: **amarillo**
		

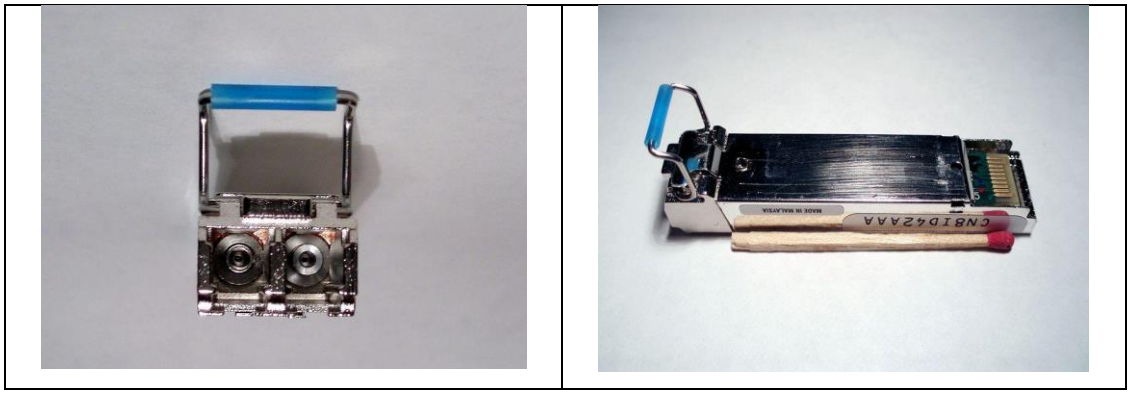
Fuente: elaboración propia.

Figura 23. **Conectores para fibra óptica**



Fuente: *Adamantios (s.f.) [Fotografias] CC-BY-SA 3.0*.
https://commons.wikimedia.org/wiki/User:Adamantios/Objects. Consulta: abril de 2015.

Figura 24. ***Small form-factor pluggable*** **transceiver (módulo para conectar fibra óptica a los dispositivos)**



Fuente: *Adamantios (s.f.) [Fotografias] CC-BY-SA 3.0*.
https://commons.wikimedia.org/wiki/User:Adamantios/Objects. Consulta: abril de 2015.

3.5.5. UTP *versus* fibra óptica

En la tabla VIII se hace la comparación del UTP y la fibra óptica.

Tabla VIII. **Cuadro comparativo UTP y fibra óptica**

UTP	Fibra óptica
● Distancia máxima: 100 m ● Vulnerable a la interferencia electromagnética (EMI) ● Ancho de banda limitado ● Implementación sencilla ● Más barato	● Distancia máxima: muchos kilómetros ● Invulnerable a la interferencia electromagnética (EMI) ● Ancho de banda teóricamente ilimitado ● Implementación especializada ● Más caro

Fuente: elaboración propia.

3.5.6. Inspección y limpieza de conectores de fibra óptica

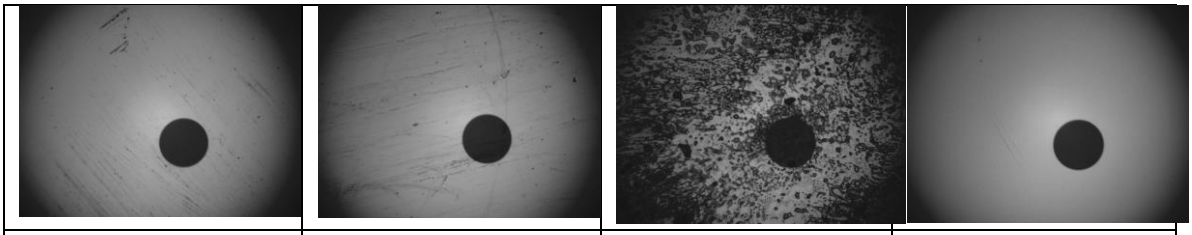
Las conexiones ópticas son extremadamente sensibles a la contaminación.

Incluso las partículas de polvo microscópicas pueden ocasionar una gran variedad de problemas, desde provocar una mala alineación entre el cable y el módulo receptor hasta ocasionar daños en los mismos.

Por esta razón se recomienda el uso de cubiertas anti polvo así como la inspección y limpieza de los conectores antes de cada acople.

Para examinar el estado de las terminaciones puede emplearse un microscopio para inspección de fibra óptica, mientras que para la limpieza existen varias herramientas tales como hisopos y paños especiales.

Figura 25. **Inspección de un conector con un microscopio óptico portátil**



Nuevo	Limpieza con ropa	Contacto con la piel	Limpieza con paño especial

Fuente: elaboración propia.

3.5.7. Consideraciones de seguridad al trabajar con fibra óptica

- Nunca se debe examinar directamente el extremo de un cable de fibra hasta estar seguro de que no existe una fuente de luz del otro lado. Los lentes protectores deben ser de uso obligatorio.

- Los residuos de fibra deben ser manejados y desechados con precaución en recipientes adecuados. Se recomienda trabajar sobre una alfombrilla negra para facilitar la localización de los mismos.

- No deben permitirse alimentos dentro del área de trabajo. La ingestión de astillas de fibra óptica puede causar graves hemorragias internas.

- El espacio de trabajo debe estar bien ventilado y lejos de materiales combustibles debido al calor generado por algunas herramientas. El consumo de cigarrillos debe ser prohibido.

- Se recomienda el uso de batas desechables, o a falta de estas, el uso de cinta adhesiva para limpiar la ropa del operador.

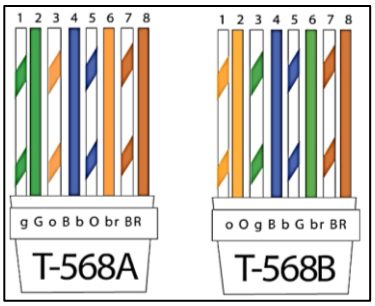
- Al finalizar su tarea, el operador debe limpiar el área de trabajo para luego lavar cuidadosamente sus manos y asegurar que ningún residuo haya quedado en sus ropas.

3.5.8. Estándares para cable de par trenzado

Las mejores prácticas en el diseño e implementación de sistemas de cableado estructurado están definidas en una serie de estándares conocidos como TIA/EIA-568-B. Publicados por primera vez en el 2011. Definen los requisitos generales, pruebas, conectores, cables y distancias recomendados para una instalación.

Este estándar define la asignación de cables y pines en los cables de cobre de 8 hilos y 100 ohmios (cable de par trenzado), siendo posible arreglar los mismos de dos maneras (T-568A y T-568B), las cuales se muestran en la figura 26.

Figura 26. **Estándares para cable de par trenzado**

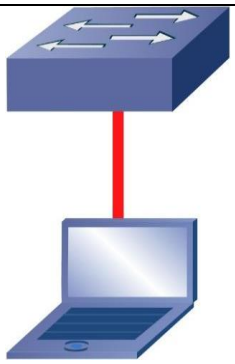
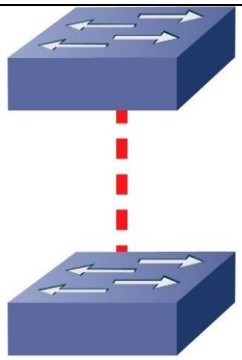


Fuente: elaboración propia, empleando *Edraw Max*.

Con dichas disposiciones es posible crear dos tipos de cables: el cable directos con dos terminaciones idénticas en sus extremos (utilizando solamente una de las Normas, ya sea T-568A o T568B), y el cruzado, con terminaciones desiguales (cada extremo utiliza una norma diferente).

En la mayoría de ocasiones se utilizan cables directos para conectar dispositivos desiguales (ej.: ordenador a *switch*) y cables cruzados para conectar dispositivos similares (ej.: *switch* a *switch*).

Figura 27. **Utilización normal de los cables directos y cruzados**

	
En la mayoría de casos se utiliza un cable directo para conectar dispositivos desiguales.	En la mayoría de casos se utiliza un cable cruzado para conectar dispositivos similares.

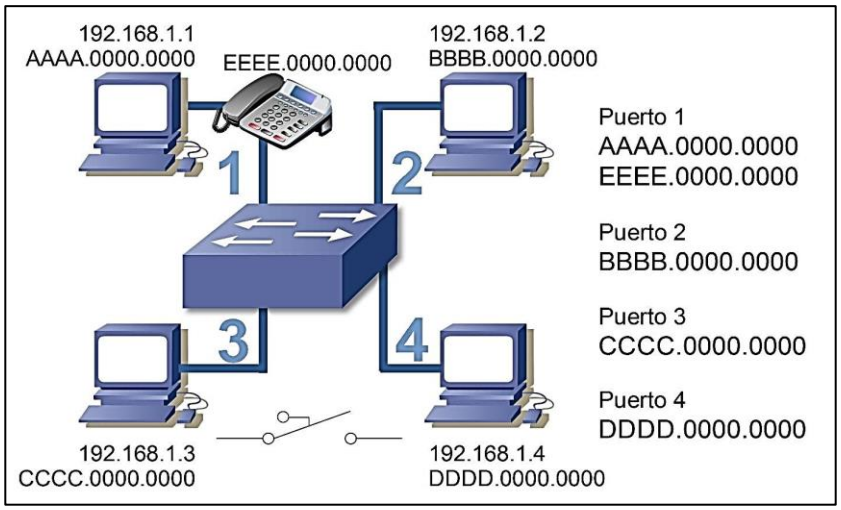
Fuente: elaboración *Edraw Max*.

El código de colores indicado por las normas mencionadas siempre debe ser respetado para que el cable cumpla con las características de funcionamiento definidas por el estándar (especialmente la distancia).

3.5.9. *Switch*

Este dispositivo aprende las direcciones MAC de los demás equipos conectados, lo que le posibilita crear circuitos únicos entre 2 dispositivos, evitar el envío de tráfico innecesario (como pasaba con los *hub*) y mejorar el uso del ancho de banda.

Figura 28. **Funcionamiento de un *switch***



Fuente: elaboración propia, empleando *Edraw Max*.

3.5.10. Modos de transmisión

Una transmisión dada en un canal de comunicaciones entre dos equipos puede ocurrir de diferentes maneras. La transmisión está caracterizada por:

- la dirección de los intercambios
- el modo de transmisión: el número de bits enviados simultáneamente
- la sincronización entre el transmisor y el receptor

3.5.10.1. Half–dúplex

- Comunicación en un solo sentido
- Un dispositivo solo puede recibir/enviar en un tiempo dado
- Típico en una red que usa *hubs*

3.5.10.2. Full–dúplex

- Comunicación en dos sentidos
- Un dispositivo puede enviar y recibir datos al mismo tiempo
- Típico en una red que usa *switches*

3.6. Introducción al Cisco IOS

El Cisco IOS (*Internetwork operating system*) es el sistema operativo utilizado por la mayoría de *routers* y *switches* de este fabricante, presenta una experiencia simple e intuitiva que se mantiene consistente, incluso en dispositivos de la marca que utilizan otros sistemas operativos.

Para interactuar con dicho sistema puede utilizarse la interfaz gráfica conocida como *Cisco Configuration Professional* o la interfaz de línea de comandos (CLI).

3.6.1. Conexión a un dispositivo a través de una línea de comandos (CLI)

Para interactuar con un dispositivo a través de una línea de comandos es necesario utilizar un emulador de terminal.

Antiguamente, una terminal era un dispositivo electromecánico utilizado para interactuar con un computador, siendo las primeras de ellas máquinas de escribir adaptadas para enviar mensajes e imprimir las respuestas sobre papel y que eran referidas como teleimpresoras o teletipos (*Teletypes - TTY*).

Figura 29. **Máquina de teletipo**

Fuente: *NightFlyer (s.f.) [Fotografias] CC-BY-SA 3.0*. https://commons.wikimedia.org/wiki/File:Teletype-Fernschreiber_T100_Siemens.jpg. Consulta: abril de 2015.

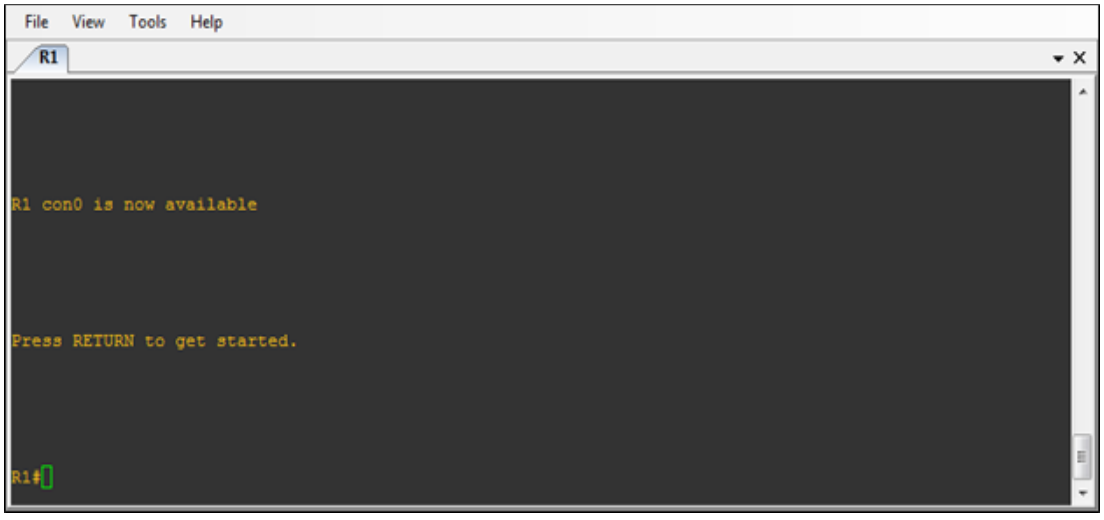
Los teletipos (TTY) fueron reemplazados eventualmente por otros dispositivos, siendo utilizados actualmente, solamente para facilitar la comunicación entre personas con déficit auditivo sobre líneas telefónicas tradicionales cuando no hay otro medio disponible. No obstante, mucha de la terminología utilizada en ese entonces fue heredada y sigue siendo utilizada en ámbitos modernos.

De esta manera, los emuladores de terminal son programas diseñados para imitar el comportamiento de dichos dispositivos, proveyendo al usuario de un medio para enviar instrucciones y recibir respuestas, las cuales ahora son impresas en una pantalla.

Dentro de los emuladores más conocidos se encuentran la hyperterminal (ahora descontinuada) en el S.O. Windows, SecureCRT, Putty y MobaXterm. Siendo los dos últimos gratuitos.

Los emuladores deben conectarse al dispositivo deseado a través de una interfaz (lógica o física) provista por el mismo y que es referida como "línea" o "*line*". Si dicha conexión es exitosa, el emulador mostrará el *prompt*, el cual consiste en una cadena de caracteres (o mensaje) indicando que se está a la espera de una orden.

Figura 30. **Captura de pantalla del programa Putty mostrando el *prompt* del Cisco IOS**

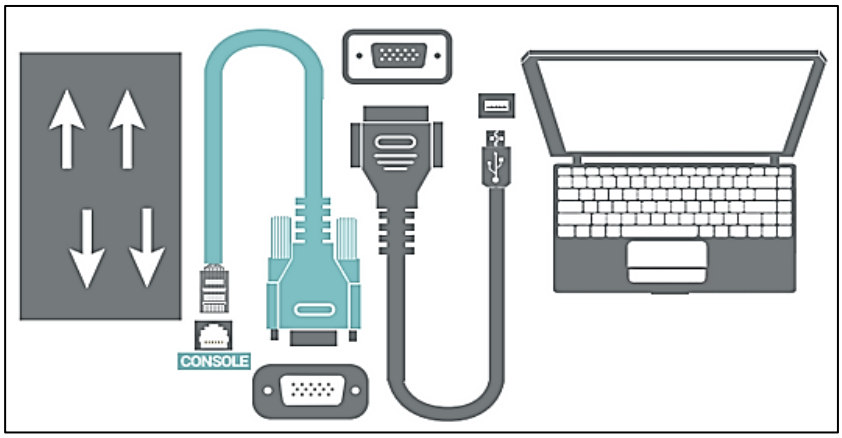


Fuente: elaboración propia.

3.6.2. Conexión local

Se realiza mediante una conexión directa con los dispositivos a través del puerto especial de consola (*console teletype* - CTY).

Figura 31. **Conexión consola con cable especial y conversor USB a serial**



Fuente: elaboración propia, empleando *Edraw Max*.

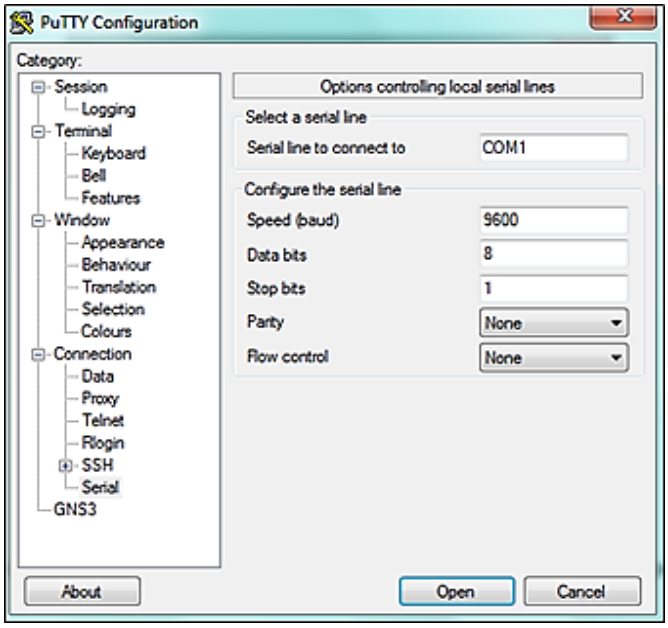
Dicha conexión obedece al estándar RS232 para comunicaciones seriales y debe cumplir con los parámetros que se describen en la tabla IX:

Tabla IX. **Parámetros para una conexión serial como aparecen en la mayoría de emuladores**

Baud rate = 9600	*Parity = none*	*Flow control = none*
Data bits = 8	*Stop bits* = 1	

Fuente: elaboración propia.

Figura 32. **Captura de pantalla del programa Putty mostrando los parámetros necesarios para una conexión a través puerto de consola**



Fuente: elaboracion propia.

La conexión realizada mediante el cable de consola siempre ocupa la primera línea del dispositivo (identificada con un número relativo de cero) y puede realizarse, por defecto, sin ingresar ninguna contraseña.

Figura 33. **Para configurar parámetros relativos a la sesión establecida utilizando el puerto de consola, puede emplearse el comando line console 0**

```
R1(config)#line console 0
R1(config-line)#
```

Fuente: elaboración propia.

3.6.3. Conexión remota

Se realiza utilizando protocolos como *teletype network* (*Telnet*) o *secure shell* (SSH).

- Creado en 1968, *Telnet* es uno de los primeros estándares del internet. Se caracteriza por ser un protocolo sencillo y por no soportar autenticación ni cifrar sus transmisiones. Aunque su uso no es recomendable actualmente, sigue siendo uno de los servicios más comunes encontrados durante las auditorías de red debido a su fácil implementación y por omisión de los administradores, que olvidan deshabilitarlo o desconocen que sigue activo.

- SSH trabaja de manera similar a *telnet* con la ventaja de cifrar la comunicación, aunque su implementación es más compleja y representa una mayor carga al CPU de los dispositivos.

La conexión remota es realizada a través de las líneas VTY (Virtual Teletype), llamadas también líneas virtuales.

Dependiendo del dispositivo, el número de líneas VTY disponible por defecto puede variar, aunque es posible crear nuevas con un número relativo elegido por el usuario.

Las líneas VTY pueden configurarse de manera individual:

Figura 34. **Configuración de la línea VTY con un número relativo de 0**

```
R1(config)#line vty 0
R1(config-line)#
```

Fuente: elaboración propia.

O como un grupo:

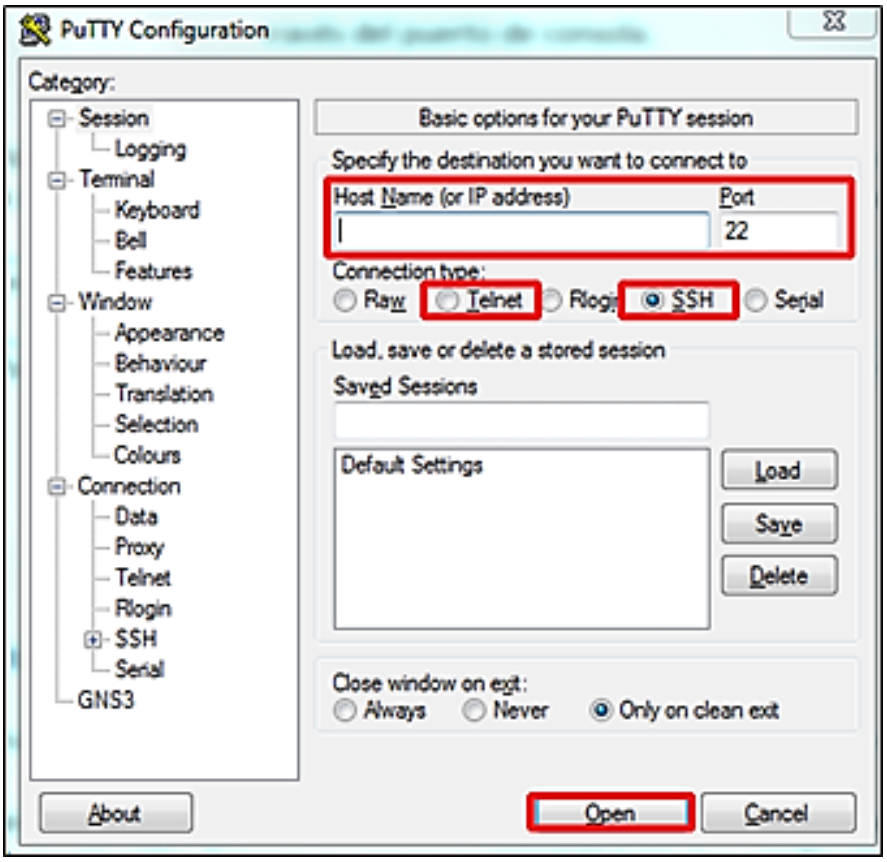
Figura 35. **Configuración de cinco (0-4) líneas VTY**

```
R1(config)#line vty 0 4
R1(config-line)#
```

Fuente: elaboración propia.

Al contrario de la conexión local, establecer una sesión remotamente requiere que el dispositivo cuente con una dirección IP alcanzable y una forma de autenticación configurada en dichas líneas virtuales.

Figura 36. **Captura de la pantalla inicial del programa Putty desde donde puede iniciarse una sesión de telnet SSH**



Fuente: elaboración propia.

3.6.4. Modos del Cisco IOS

El Cisco IOS divide su funcionalidad dentro de varios modos, cada uno de los cuales con su propio subconjunto de comandos, siendo algunos de los más usuales los que se describen a continuación.

- Modo usuario (*User mode*): es el modo por defecto cuando se utiliza la interfaz de línea de comandos (CLI). Con un nivel bajo de privilegios; en este modo solo es posible realizar algunas pruebas básicas y mostrar información general del sistema.

El *prompt* indica este modo utilizando el símbolo mayor que “>”.

Figura 37. **Modo usuario**

```
R1>
```

Fuente: elaboración propia.

- Modo privilegiado (*Privileged mode*): es el modo con el más alto nivel de privilegio, por lo que tiene acceso a todos los comandos. Por defecto no tiene una contraseña asignada, consecuentemente podrá ser utilizado en una conexión local hasta que se haya configurado una.

Para alcanzar el modo privilegiado puede utilizarse el comando *enable* desde el modo de usuario, siendo este modo indicado en el *prompt* con un símbolo de numeral “**#**”.

Figura 38. **Modo privilegiado**

```
R1> enable
R1#
```

Fuente: elaboración propia.

- Modo de configuración global (*Global configuration mode*): es el modo desde el cual pueden configurarse parámetros que afectan a todo el sistema.

Puede accederse desde el modo de usuario privilegiado utilizando el comando *configure terminal* y es indicado por la palabra "*config*" entre paréntesis antes del símbolo de numeral "**#**".

Figura 39. **Modo de configuración global**

```
R1> enable
R1# configure terminal
R1(config)#
```

Fuente: elaboración propia.

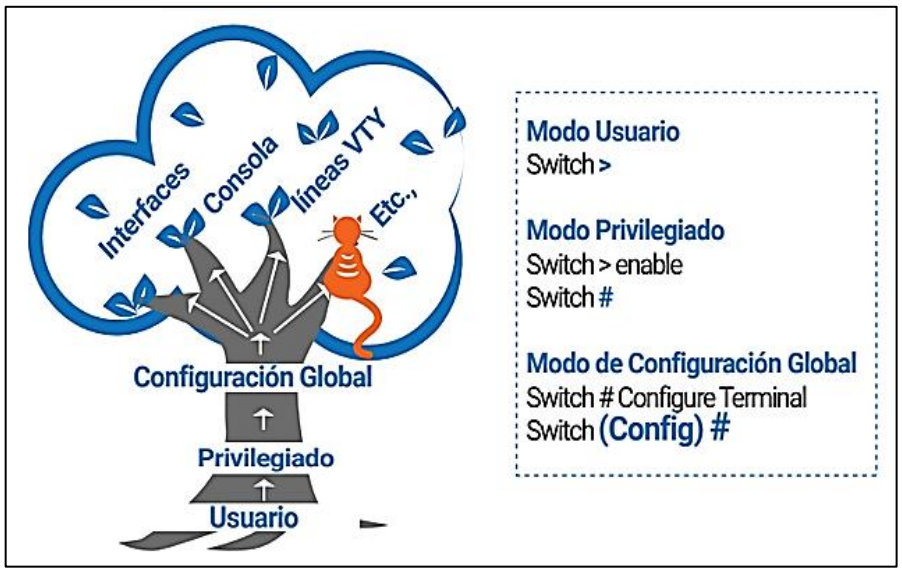
Para salir del modo de configuración global es posible usar el comando *exit* y para regresar al modo de usuario desde el modo privilegiado se utiliza el comando *disable*.

Figura 40. **De regreso al modo de usuario**

```
R1(config)# exit
R1# disable
R1>
```

Fuente: elaboración propia.

Figura 41. **Modos del Cisco IOS**



Fuente: elaboración propia, empleando *Edraw Max*.

3.6.5. Ayuda y edición en el Cisco IOS

La ayuda en el Cisco IOS es simple e interactiva, y es posible acceder a ella utilizando el signo de interrogación “?”, cuyo funcionamiento depende del contexto donde se utilice.

- Para indicar los comandos disponibles, así como una descripción de los mismos basta con presionar “?” en el modo deseado.

Figura 42. **Uso de la ayuda para mostrar los comandos disponibles en un modo**

```
Switch(config)# ?

Configure commands:
  access-list      Add an access list entry
  banner           Define a login banner
  --More--
```

Fuente: elaboración propia.

- Para indicar la siguiente palabra para completar una instrucción puede utilizarse “?”, precedido por el fragmento a completar y un espacio.

Figura 43. **Uso de la ayuda para completar un comando**

```
Switch(config)# hostname ?

WORD  This system's network name
```

Fuente: elaboración propia.

- Para indicar qué comandos empiezan por ciertas letras se emplea “?”, después de las mismas sin espacio alguno.

Figura 44. **Uso de la ayuda para mostrar los comandos que empiezan con ciertas letras**

```
Switch# con?

configure  connect
```

Fuente: elaboración propia.

Este sistema operativo presenta ciertas funciones destinadas a facilitar la configuración y la edición de instrucciones, tales como:

- Completar instrucciones automáticamente utilizando la tecla TAB

Si el fragmento ingresado es reconocido como único (es decir que no es ambiguo), al presionar la tecla mencionada se imprimirá el comando reconocido en la línea que se muestra en la figura 45.

Figura 45. **Completar instrucciones con tecla TAB**

```
R1# tel   ! Al presionar TAB aparece la siguiente línea.
R1# telnet
```

Fuente: elaboración propia.

- Indicar que una instrucción está incompleta.

Figura 46. **Indicación de instrucción incompleta**

```
R1# clock ! Al tratar de enviar la instrucción presionando la tecla ENTER.
% incomplete command.
```

Fuente: elaboración propia.

- Indicar la posición en donde ha ocurrido un error en la sintaxis.

Figura 47. **Indicación de error**

```
R1# clock set ERROR  ! Al tratar de enviar la instrucción presionando la tecla ENTER.
                     ^
% invalid input detected at '^' marker.
```

Fuente: elaboración propia.

- Aceptar fragmentos de instrucciones reconocidos como únicos.

Figura 48. **Aceptación de fragmentos de instrucciones**

```
Switch#con
 % ambiguous command: "con"

Switch#con?
configure  connect

Switch#conf  ! Al enviar la instrucción presionando la tecla ENTER.
Configuring from terminal, memory, or network [terminal]?
Switch(config)#
```

Fuente: elaboración propia.

3.6.6. Comandos *show*

Dentro del Cisco IOS, toda información (sistema, configuración, estado de las interfaces, estadísticas, entre otros) es desplegada utilizando los comandos *show.* De estos comandos los más elementales son:

- Show *running-config*

Muestra la configuración que está siendo ejecutada por el dispositivo y que reside en la memoria volátil del mismo (RAM, por lo que se perdería la configuración en caso de que el dispositivo se quedará sin poder o fuera reiniciado). Este comando presenta usualmente una salida extensa y representa una carga para el CPU, por lo que debe utilizarse con precaución en redes en producción.

Figura 49. ***Show running-config***

```
R1# show running-config
Building configuration...

Current configuration : 932 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
no aaa new-model
no ip domain lookup

interface FastEthernet0/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface Serial0/0
 no ip address
 shutdown
 clock rate 2000000

 --More--  !!!!!  La palabra "more" significa que hay más información, la cual puede
                  !!!!!  mostrarse línea  por línea con la tecla ENTER o pantalla por pantalla
                  !!!!!  usando la barra espaciadora.
```

Fuente: elaboración propia.

- Show *startup-config*

Con una salida similar al comando anterior, este se presenta con la configuración con la que el dispositivo inicia, misma que es almacenada en la memoria no volátil (NVRAM) del equipo.

Figura 50. ***Show startup-config***

```
R1#show startup-config
startup-config is not present
!!!!!    El mensaje anterior indica que no existe configuración inicial.
```

Fuente: elaboración propia.

- Show *ip interface brief*

Es uno de los comandos más útiles. Se presenta con un resumen de las interfaces del dispositivo, la dirección IP asignada a cada una de ellas (*IP-Address*), el método utilizado para conseguir dicha dirección (*method*) y el estado de las mismas, el cual puede ser *administratively down*, *down* o *Up.*

Administratively down indica que la interfaz ha sido deshabilitada por un administrador, *down* muestra que la interfaz está encendida, pero que no se detecta señal en el cable (lo que sugiere que el mismo sufrió alguna falla, desconexión o que el dispositivo conectado en el otro extremo se encuentra apagado), mientras que *up* señala que el puerto se encuentra completamente operacional.

Figura 51. ***Show ip interface brief***

```
R1# show ip interface brief
 Interface              IP-Address      OK?  Method  Status    Protocol
 FastEthernet0/1        unassigned      YES  manual  down        down
 FastEthernet0/2        unassigned      YES  manual  down        down
 FastEthernet0/3        unassigned      YES  manual  down         down
```

Fuente: elaboración propia.

- *Show interface* (nombre de la interfaz)

Muestra información detallada concerniente a una interfaz en particular, dirección IP, estado, velocidad, errores, entre otros.

Figura 52. ***Show interface***

```
R1# show interface fastEthernet 0/0

FastEthernet0/0 is up, line protocol is up
  Hardware is Gt96k FE, address is c001.0950.0000 (bia c001.0950.0000)
  Internet address will be negotiated using DHCP
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive not set
  Half-duplex, 10Mb/s, 100BaseTX/FX
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input never, output 00:00:44, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog
     0 input packets with dribble condition detected
     33 packets output, 17741 bytes, 0 underruns
     0 output errors, 0 collisions, 1 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out
```

Fuente: elaboración propia.

3.7. Subredes y superredes

Las subredes son el método de dividir una red IP en a subestaciones subredes llamadas mientras, superredes es el método de mezcla y redes IP manhy coincidentes con una red general de prefijo. Es importante saber que superredes reducirá el número de entradas en una tabla de encaminamiento y ellos también facilitará el proceso de distribución para que sea más sencillo. Por otra parte, en las subredes, se toman pedacitos de ID de host (para las direcciones IP de una red única ID) para ser utilizado como un identificador de subred.

3.7.1. Máscara de subred

El primer esquema de direccionamiento propuesto por el protocolo de internet dividía todas las direcciones IP en dos partes claramente establecidas, una indicando la dirección de la red a la que esta pertenecía (*Network* ID) y la otra, la parte que podía utilizarse para asignar una dirección individual a un dispositivo específico (*Host* ID), siendo sus respectivos límites establecidos por las antiguas clases de direcciones IP (A, B, C, D, E), las cuales organizaban las mismas dentro de varios rangos.

Esta primera aproximación no se ajustaba a la estructura utilizada dentro de las organizaciones, por lo que requerían usualmente de varias redes, sin importar cuántos dispositivos tuvieran que colocar dentro de cada una de ellas, para poder proveer de conectividad a los grupos dispares existentes.

Agregado al costo económico existía también, la posibilidad del completo agotamiento del esquema de direcciones, por lo que a la espera de una solución

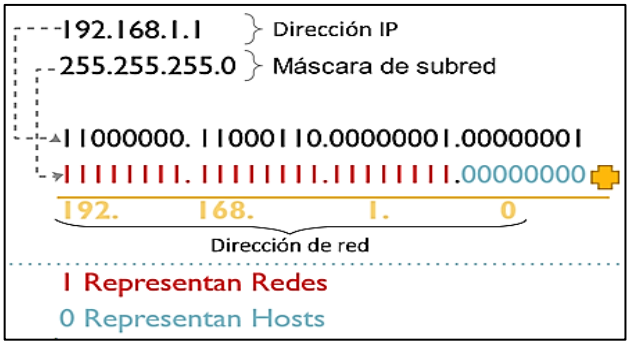
a largo plazo, era necesario crear una manera de hacer un mejor uso de las mismas.

Uno de los ajustes realizados fue la introducción de las subredes, las cuales permitían a las organizaciones conectadas a internet tomar una red asignada y dividirla en redes más pequeñas para acomodar mejor las direcciones adquiridas.

Para poder identificar estas nuevas subredes fue necesario introducir un nuevo trozo de información, en la forma de una máscara de bits llamada de subred, momento en el cual las máscaras por defecto fueron asignadas a las clases de direcciones ya existentes.

La máscara de subred es una combinación de 32 bits (en IPv4) formada por una serie de unos y ceros continuos (en toda implementación moderna) y que indican, respectivamente, qué parte de una dirección IP será utilizada para identificar la red/subred y cuál será la empleada para identificar de manera individual a un dispositivo.

Figura 53. **Operación AND entre una dirección IP y una máscara de subred**



Fuente: elaboración propia, empleando *Edraw Max*.

Es a través de la combinación de la dirección IP y la máscara de subred (empleando varias operaciones matemáticas), que un *host* puede llegar a conocer la red/subred a la que este pertenece, así como la dirección de *broadcast* de la misma, sabiendo por consiguiente, si una transmisión está destinada a un dispositivo dentro de la misma red o si va dirigida a una red externa, utilizando en este último caso la ayuda de su puerta de enlace predeterminada.

3.7.2. VLSM y CIDR

Si bien la introducción de las subredes constituyó un gran avance para las organizaciones, estas se encontraban todavía limitadas debido al hecho de que cada nuevo subconjunto de direcciones era de un tamaño fijo y no podía ser ajustado acorde a un número de dispositivos.

Para proporcionar la flexibilidad deseada se decidió que las máscaras de subred podrían establecerse de una manera arbitraria acuñando el término “máscara de subred de longitud variable”. (*Variable Length Subnet Mask* (VLSM)).

Los beneficios y funcionalidad aportados por la máscara de subred de longitud variable fueron adoptados un poco más adelante, para servir no solamente a las organizaciones sino a todo el internet, por lo que en 1993, la internet *Engineering Task Force* (IETF) introdujo el *Classless Inter Domain Routing* (CIDR) el cual permite romper los límites de las antiguas clases de direcciones, al poder asignar cualquier máscara a cualquier dirección, aunque estas siguieron utilizándose como referencia debido a su familiaridad.

Al modificar la longitud de las máscaras de subred a conveniencia, se hizo posible, no solamente la división de una red en varias subredes sino también agregar varias redes a una sola mucho más grande, referida como superred, lo que ayudó a reducir la cantidad de rutas que los dispositivos debían conocer.

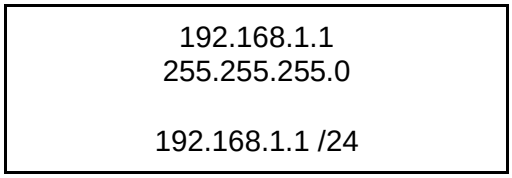
A partir de la introducción de las tecnologías mencionadas, la combinación de una dirección IP junto con su máscara se hizo obligatorio, ya que de manera separada estas no pueden transmitir ninguna información útil.

3.7.3. Notaciones de la máscara de subred

Es posible indicar una máscara de subred utilizando números decimales o a través de una notación alternativa, usualmente referida como de bit, de barra diagonal, CIDR o como longitud del prefijo, misma que está compuesta por una barra oblicua seguida de la cantidad de bits con valor de uno presentes en la máscara.

De esta manera se presentan nuevamente la dirección IP y la máscara de subred utilizadas previamente empleando ambas notaciones.

Figura 54. **Notaciones de la máscara de subred**



Fuente: elaboración propia.

3.7.4. *Subnetting*

Es el nombre que recibe la técnica empleada para dividir una sola red en varias subredes más pequeñas.

3.7.4.1. *Subnetting* tradicional

Para llevar a cabo la división de una red en subredes más pequeñas (de un mismo tamaño) es necesario seguir los siguientes pasos:

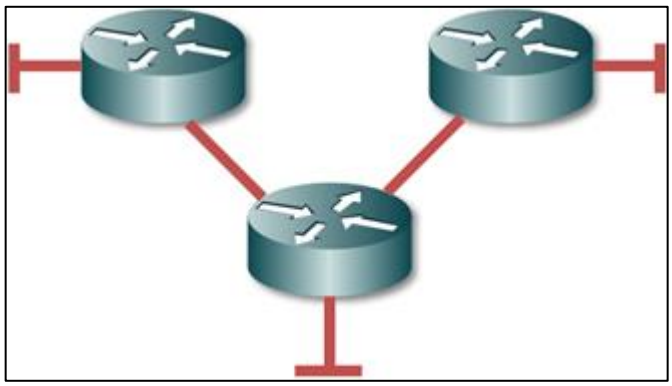
- Definir los requerimientos del diseño, ya sea cantidad de redes o de *hosts* y convertir a binario.

- Reservar los bits necesarios en la máscara de subred y hallar el incremento.

- Usar el incremento y determinar los rangos de las nuevas subredes.

A manera de ejemplo se introduce la siguiente topología donde una organización ha comprado un bloque de direcciones pertenecientes a la antigua clase C: 215.10.5.0 /24, con el objetivo de direccionar las redes que se muestran (ver figura 55).

Figura 55. **Topología a direccionar utilizando *subnetting* tradicional**



Fuente: elaboración propia, empleando *Edraw Max*.

En la topología presentada es posible apreciar la necesidad de dividir el bloque de direcciones en 5 subredes más pequeñas.

A continuación se describen los pasos explicados anteriormente:

- Paso 1: para direccionar esta topología se requiere la creación de 5 subredes. Al convertir dicho requerimiento se identificó que este necesita 3 bits para poder ser expresado en binario.

Tabla X. **Conversión a binario del número cinco**

128	64	32	16	8	4	2	1
0	0	0	0	0	1	0	1

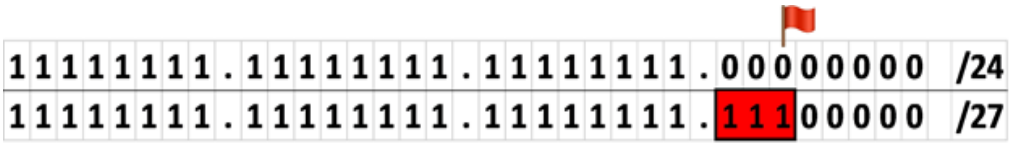
Fuente: elaboración propia, empleando *Edraw Max*.

- Paso 2: se procede a reservar los bits necesarios en la máscara de subred original en orden, para acomodar el requerimiento original.

Dadas las características de la máscara de subred explicadas en las secciones anteriores, se tiene que esta debe ser continua y que los bits con un valor de uno indican la parte de red/subred, y aquellos bits con un valor de cero muestran la parte reservada para las direcciones de *host.*

En este caso, al ser el requerimiento presentado en un número de subredes, se procede a agregar 3 bits con un valor de uno a la máscara original de la manera que se muestra en la figura 56.

Figura 56. **Reserva de los bits necesarios en la máscara original**



Fuente: elaboración propia, empleando *Edraw Max*.

Al agregar dichos bits se encuentra una nueva máscara de subred, esta será /27 o 255.255.255.224.

Adviértase que al aumentar la longitud de la máscara, tomando bits del último octeto, se ha introducido un corrimiento u *offset* (indicado por la banderilla en la figura 56) por lo que el incremento entre subredes será de 2^5 o de 32 en vez de ser de 1 (2^0).

- Paso 3: al haber encontrado el incremento (32) puede encontrarse los rangos de las subredes necesarias al sumar el mismo a la dirección original en el octeto apropiado. Nótese que estos rangos solo son válidos si las direcciones incluidas son utilizadas junto con la máscara de subred determinada en el paso anterior (/27).

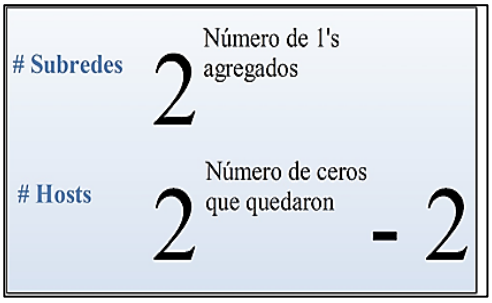
Tabla XI. **Subredes necesarias para direccionar la topología solicitada**

#	Red	Primera Dirección Utilizable	Última Dirección Utilizable	Broadcast
1	215.10.5.0	215.10.5.1	215.10.5.30	215.10.5.31
2	215.10.5.32	215.10.5.33	215.10.5.62	215.10.5.63
3	215.10.5.64	215.10.5.65	215.10.5.94	215.10.5.95
4	215.10.5.96	215.10.5.97	215.10.5.126	215.10.5.127
5	215.10.5.128	215.10.5.129	215.10.5.158	215.10.5.159

Fuente: elaboración propia, empleando *Edraw Max*.

Para corroborar que se han cumplido con todos los requerimientos es posible utilizar las fórmulas mostradas en la figura 57, mismas que toman en cuenta la cantidad de bits con valor de uno agregados y la cantidad de bits con un valor de cero restantes en la nueva máscara de subred.

Figura 57. **Fórmulas para determinar la cantidad de subredes y las direcciones de *host* disponibles dentro de cada una de ellas dada una máscara de subred**



Fuente: elaboración propia, empleando *Edraw Max*.

Así pues, para corroborar el ejemplo anterior, dada la máscara /27 será posible tener $2^3 = 8$ subredes, con $2^5-2 = 30$ *hosts* dentro de cada una de ellas.

3.8. *Dynamic Host Configuration Protocol* (DHCP)

Es un protocolo que permite automatizar la asignación de direcciones IP en dispositivos finales, no está orientado a conexión, por lo que utiliza UDP en el puerto 67 en el caso del servidor y el 68 en el caso del cliente.

El proceso de asignación de direcciones se lleva a cabo de la siguiente manera:

- El cliente envía un DHCP *DISCOVER* como un *broadcast* con el propósito de localizar un servidor DHCP.

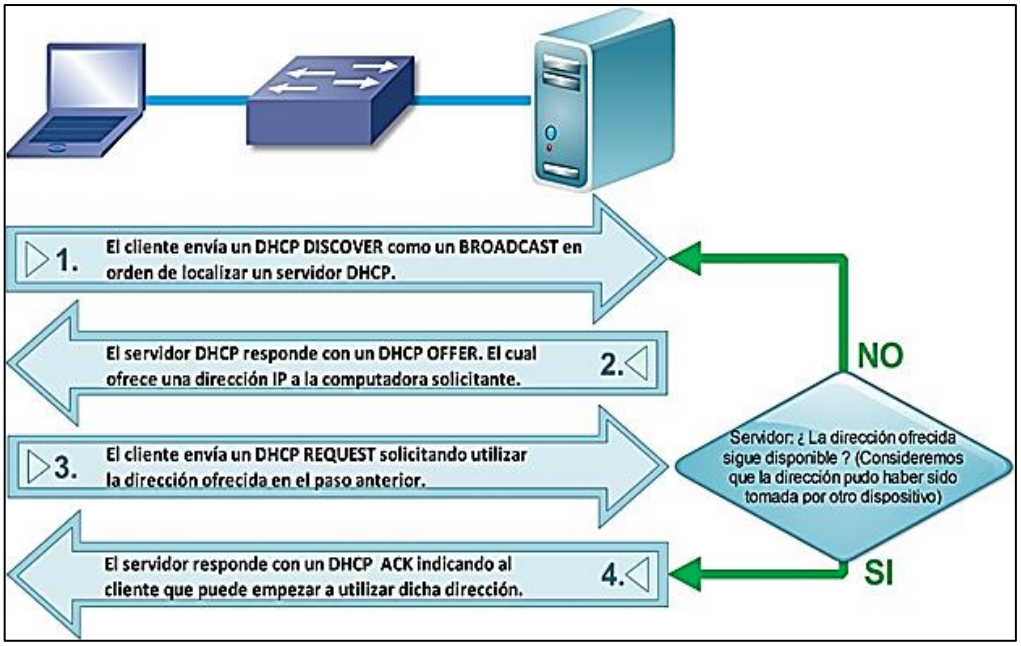
- El servidor DHCP responde con un DHCP *OFFER*, el cual ofrece una dirección IP al solicitante.

- El cliente envía un DHCP *REQUEST* solicitando utilizar la dirección ofrecida en el paso anterior; si esta dirección ya no se encontrara disponible (ha sido tomada por otro dispositivo), entonces el proceso comenzará nuevamente.

- El servidor responde con un DHCP ACK indicando al cliente que puede empezar a utilizar dicha dirección.

Un resumen del proceso se ilustra en la figura 58:

Figura 58. **Resumen del proceso DHCP**



Fuente: elaboración propia, empleando *Edraw Max*.

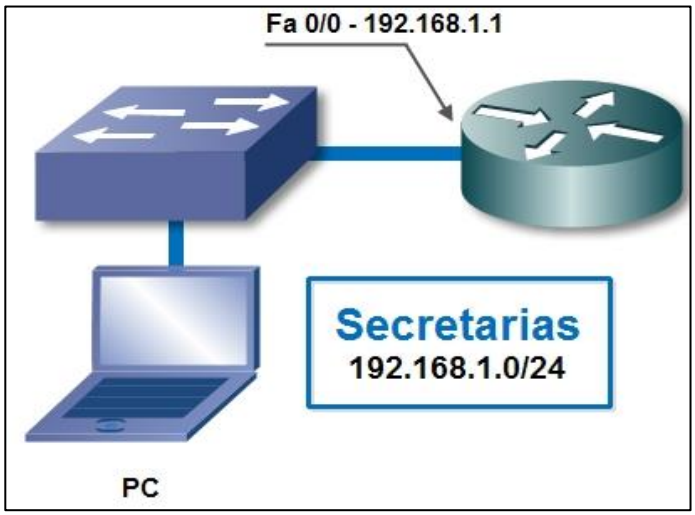
Al configurar DHCP en una red puede encintrarse con uno de los siguientes dos escenarios.

3.8.1. El servidor DHCP se encuentra dentro del mismo dominio de *broadcast*

En este caso, la configuración es directa, y el único problema es el de la seguridad ya que, por diseño el DHCP *DISCOVER* llega indiscriminadamente a todos los miembros de la red, lo que posibilita la introducción de servidores no autorizados.

Muchos de los dispositivos Cisco incluyen un servidor DHCP habilitado por defecto. Para mostrar un ejemplo de su implementación se presenta en la figura 59 la siguiente topología, donde la interfaz del *router* ha sido previamente configurada para poder hacer énfasis en el servicio en cuestión.

Figura 59. ***Router* como servidor DHCP en el mismo dominio de *broadcast***



Fuente: elaboración propia, empleando *Edraw Max*.

En esta oportunidad se pretende configurar el *router* mostrado a manera que asigne una dirección automáticamente a las computadoras pertenecientes a la red “Secretarias”.

Para indicar qué direcciones serán excluidas del proceso DHCP (direcciones que no serán asignadas dinámicamente y que se usan como direcciones estáticas para puertas de enlace predeterminadas [*default gateways*], servidores, impresoras, entre otros), es posible utilizar el siguiente comando a través del cual se reservarán, para este ejercicio, las primeras diez direcciones disponibles para su uso estático.

Figura 60. **Uso del comando *network***

```
Router(config)# ip dhcp excluded-address 192.168.1.1 192.168.1.10
```

Fuente: elaboración propia.

Luego es necesario crear una piscina de direcciones. En este caso se utilizará el nombre “Secretarias”.

Figura 61. **Creación de piscina de direcciones**

```
Router(config)# ip dhcp pool SECRETARIAS
```

Fuente: elaboración propia.

Para establecer la red y la máscara de subred que serán utilizadas por esta piscina se utiliza el comando *network.*

Independientemente del contexto, el comando *network* cumple dos funciones, la primera de ellas es indicar la red que formará parte de un proceso (en este caso en particular DHCP) y la segunda es seleccionar aquella interfaz del dispositivo que posea una dirección IP perteneciente a dicha red, para hacer la conexión del proceso en cuestión con el mundo físico.

Figura 62. **Uso del comando *network***

```
Router(dhcp-config)# network 192.168.1.0 255.255.255.0
```

Fuente: elaboración propia.

En este ejemplo, la interfaz elegida para escuchar y responder las peticiones de los clientes será FastEthernet 0/0, ya que posee una dirección IP que pertenece a la red configurada en el paso anterior.

Acto seguido es necesario indicar a los clientes cuál será la dirección de su puerta de enlace predeterminada.

Figura 63. **Indicación de puerta de enlace**

```
Router(dhcp-config)# default-router 192.168.1.1
```

Fuente: elaboración propia.

Finalmente, es posible enviar otro tipo de información a los clientes, por ejemplo, la dirección del servidor DNS.

Figura 64. **Configuracion del servidor de nombres**

```
Router(dhcp-config)# dns-server 8.8.8.8
Router(dhcp-config)#exit
```

Fuente: elaboración propia.

3.8.2. El servidor DHCP se encuentra en otro dominio de *broadcast*

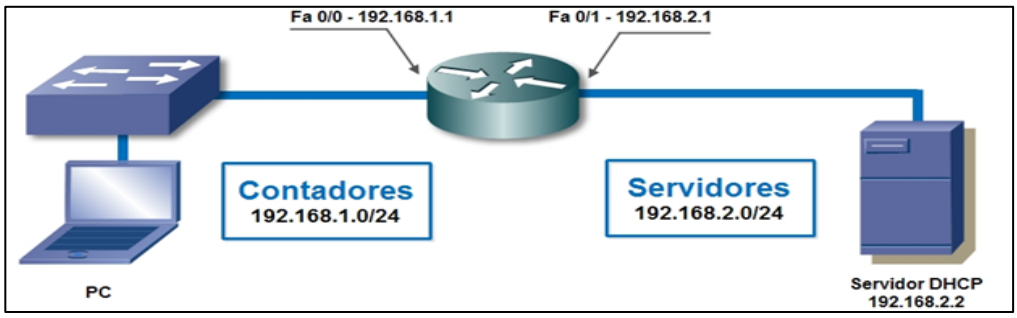
El otro escenario posible consiste en que los dispositivos finales obtengan su configuración desde un servidor localizado dentro de otro dominio de *broadcast* en algún otro punto de la topología.

En este tipo de implementación hay que considerar que cualquier mensaje enviado como un *broadcast* no será retransmitido hacia otras redes, ya que estos son limitados por los dispositivos de capa 3 (ej. *routers*) lo que constituye un problema para la operación de DHCP.

Para este tipo de casos es posible configurar el *router* como un DHCP *Relay,* un dispositivo intermediario entre el servidor DHCP y los clientes, el cual encapsulará las peticiones de estos últimos y las reenviará como un *unicast* al destino deseado.

A manera de ejemplo se presenta la siguiente topología en la figura 65, donde tanto el *router* como el servidor DHCP han sido previamente configurados.

Figura 65. **Servidor DHCP en otro dominio de *broadcast***



Fuente: elaboración propia, empleando *Edraw Max*.

Para que el *router* funcione como un DHCP *Relay* para la red de "Contadores", Cisco proporciona el comando *ip helper-address,* el cual debe ser configurado en la *interface* que se encuentre dentro del mismo dominio de *broadcast* que los clientes, para que este pueda encapsular las peticiones de los mismos y retransmitirlas como un *unicast* al servidor DHCP en la red "Servidores".

Figura 66. **Comando *ip helper-address***

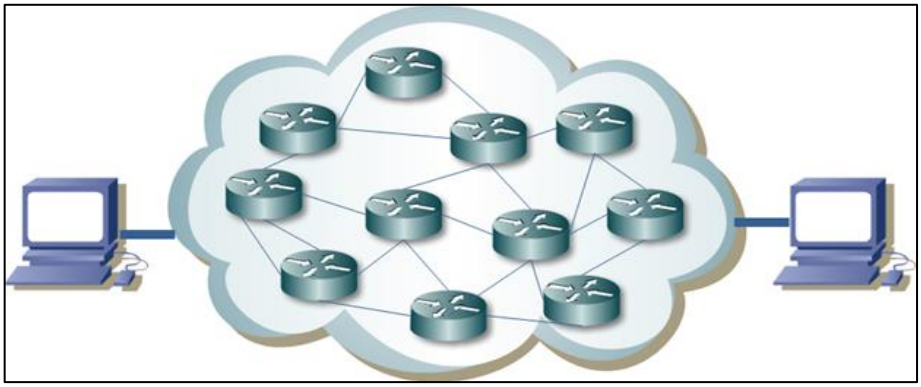
```
Router(config)# interface fastethernet 0/0
Router(config-if)# ip helper-address 192.168.2.2
```

Fuente: elaboración propia.

3.9. Enrutamiento

Es la capacidad de un dispositivo de encontrar la mejor ruta (o camino) entre todas las posibles, incluso dentro de una disposición con un alto grado de interconectividad o redundancia.

Figura 67. **Múltiples rutas para llegar de un punto a otro de la red**

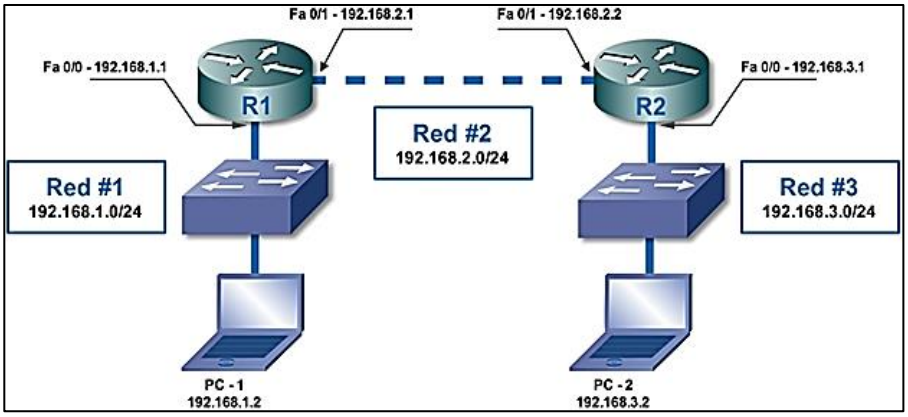


Fuente: elaboración propia, empleando *Edraw Max*.

El dispositivo encargado de encontrar todas las posibles rutas y elegir las mejores para que sean utilizadas en la transmisión de datos es conocido como enrutador o, más comúnmente, como *router,* el cual almacenará las mismas en un espacio de memoria referido como la tabla de enrutamiento.

Para comprender mejor el funcionamiento de un *router* se introduce la topología que se muestra en la figura 68, misma que seguirá siendo utilizada en todas las secciones relacionadas con el enrutamiento y donde se supondrá que todas las interfaces han sido configuradas previamente como se muestran.

Figura 68. **Topología base para los ejemplos de las secciones de enrutamiento**



Fuente: elaboración propia, empleando *Edraw Max*.

Para revisar la tabla de enrutamiento de un dispositivo, se puede utilizar el comando descrito en la figura 69, que nos presenta las rutas aprendidas y una serie de códigos que indican el origen de estas.

Figura 69. **Comando para revisar tabla de enrutamiento**

```
R1# show ip route

Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, FastEthernet0/0
C       192.168.2.0/24 is directly connected, FastEthernet0/1
```

Fuente: elaboración propia.

A manera de ejemplo se analiza la ruta descrita en la figura 70, extraída de la salida anterior, donde se muestra que el dispositivo conoce una manera de alcanzar la red 192.168.1.0/24 a través de su interfaz *FastEthernet* 0/0 y que esta ha sido aprendida gracias a que la misma se encuentra conectada directamente al dispositivo, esto es indicado con el código “C” (*Connected*).

Figura 70. **Ejemplo de análisis de ruta**

```
C    192.168.1.0/24 is directly connected, FastEthernet0/0
```

Fuente: elaboración propia.

De este último ejemplo se desprende un importante concepto: por defecto, un *router* solo conoce aquellas redes a las que está conectado directamente.

Haciendo a un lado aquellas redes que se encuentran directamente conectadas, existen dos maneras en las que un *router* puede aprender nuevas rutas: estáticamente, que requiere configuración manual o dinámicamente, asimismo de la configuración de un protocolo de enrutamiento.

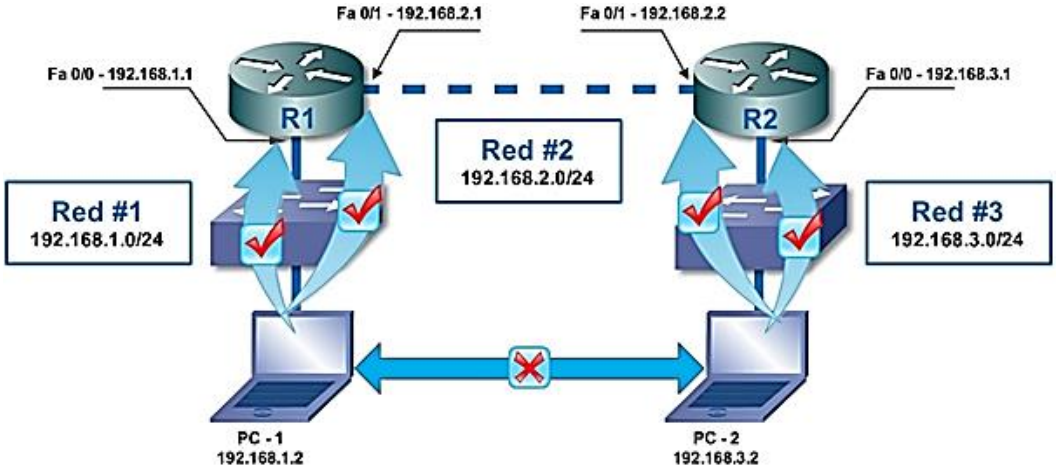
3.9.1. Enrutamiento estático

En este tipo de enrutamiento las rutas deben ser ingresadas manualmente dentro de los dispositivos. Son ideales para redes pequeñas no propensas al cambio, no consume ancho de banda y es, en cierta manera, más seguro que el enrutamiento dinámico, ya que todas las rutas son definidas directamente por el administrador.

Sin embargo, el enrutamiento estático no es muy escalable ni resiliente, en el sentido que no puede ajustarse automáticamente al crecimiento y a los cambios de la red; requiere para este propósito de la intervención humana, esto provoca una sobrecarga administrativa.

Como ejemplo de la implementación de este tipo de enrutamiento se presenta nuevamente la topología base mostrada al inicio de la sección, en donde se observan los resultados de una prueba de conectividad realizada con la herramienta *ping* (*Packet InterNet Groper*).

Figura 71. **Topología base. Prueba de conectividad usando *ping***




```
PC1> ping 192.168.1.1
84 bytes from 192.168.1.1 icmp_seq=4 ttl=255 time=49.003 ms
PC2> ping 192.168.3.1
84 bytes from 192.168.3.1 icmp_seq=4 ttl=255 time=89.005 ms
```

```
PC1> ping 192.168.3.2
*192.168.1.1 icmp_seq=1 ttl=255 time=69.004 ms (Destination host unreachable)
```

Fuente: elaboración propia.

Se puede apreciar en la figura 71 que ambas computadoras son capaces de alcanzar sus puertas de enlace predeterminadas, pero son incapaces de comunicarse entre ellas.

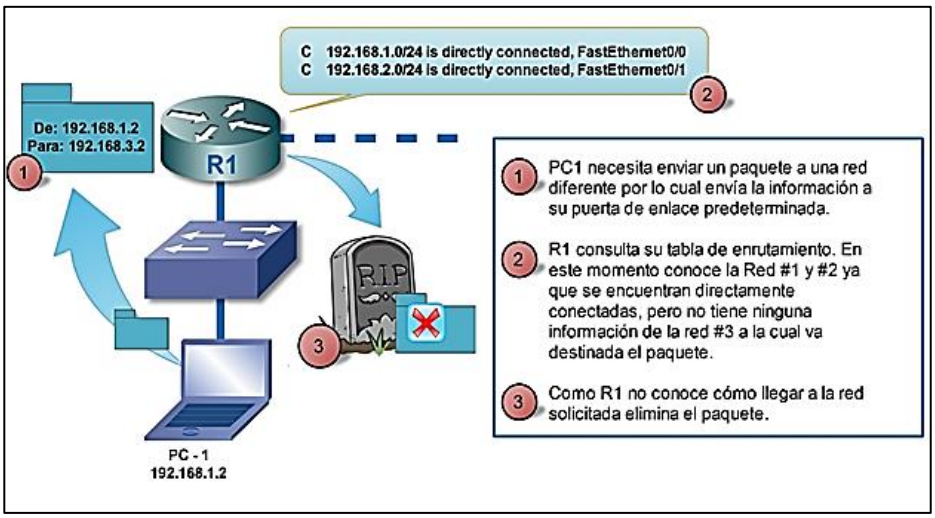
Al revisar paso a paso la prueba de conectividad llevada a cabo entre los dos ordenadores, se tiene a PC-1 tratando de alcanzar a PC-2, la cual posee una dirección IP 192.168.3.2 y se encuentra dentro de la red #3, utilizando *ping.*

Cuando realiza un análisis de su propia dirección y máscara de subred PC-1, descubre que el destino de la transmisión se encuentra en una red diferente, por lo que la envía a su puerta de enlace predeterminada, el *router* R1.

R1 consulta su tabla de enrutamiento en busca de una manera de enviar la información a la red especificada, sin embargo, en este momento la tabla de enrutamiento de R1 solo tiene entradas para la red #1 y la red #2, ya que estas son las que se encuentran conectadas directamente, por lo que los paquetes dirigidos a cualquier otra red serán descartados.

En la figura 72 se aprecia todo el proceso. Puede realizarse un análisis similar para la comunicación entre PC-2 y R2.

Figura 72. **Primer intento de comunicación entre PC-1 y PC-2**



Fuente: elaboración propia, empleando *Edraw Max*.

Para establecer comunicación entre las dos computadoras es posible configurar manualmente rutas estáticas en ambos *routers,* indicando la red que se pretende alcanzar y la forma en que se enviarán los paquetes destinados a la misma, pudiendo utilizarse una dirección IP de otro dispositivo (dirección del "siguiente salto") o una interfaz física del aparato para dar salida a la información.

En el caso de la comunicación iniciada desde PC-1 y dirigida a PC-2, es necesario configurar una ruta estática en R1 para aquellos paquetes destinados a la red #3 que envíe la información hacia otro dispositivo, a través de una dirección IP alcanzable o que seleccione una interfaz física (Fa 0/0 o Fa 0/1 en este caso), para dar salida a la transmisión.

Independientemente del método elegido, en este ejemplo, es necesario reenviar aquellos paquetes destinados a la red #3 a manera que estos alcancen el *router* R2, dispositivo que conoce dicha red al estar directamente conectado.

En esta oportunidad se empleará la dirección 192.168.2.2 (Fa 0/1 - R2) como dirección del siguiente salto en el camino hacia la red #3 (192.168.3.0/24) al utilizar la siguiente instrucción para crear una ruta estática.

Figura 73. **Instrucción para crear una ruta estática**

```
R1(config)# ip route 192.168.3.0 255.255.255.0 192.168.2.2
```

Fuente: elaboración propia.

Al examinar nuevamente la tabla de enrutamiento de R1 se encuentra una nueva ruta de carácter estático, indicado por el código “S” (*Static*).

Figura 74. **Enrutamiento con nueva ruta estática**

```
R1# show ip route
Codes: 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

Gateway of last resort is not set

C       192.168.1.0/24 is directly connected, FastEthernet0/0
C       192.168.2.0/24 is directly connected, FastEthernet0/1
S       192.168.3.0/24 [1/0] via 192.168.2.2
```

Fuente: elaboración propia.

Gracias a la ruta recién creada R1 será capaz de hacer llegar los paquetes destinados a la red #3 al dispositivo adecuado, no obstante, la comunicación entre ambas computadoras no será posible todavía debido a que R2 no posee una ruta hacia la red #1, por lo que los paquetes serán descartados cuando PC-2 intente responder a la comunicación a través de este dispositivo.

Para crear una ruta estática en R2 se utilizará la misma instrucción empleada anteriormente, con la diferencia de que en esta ocasión se configurará una interfaz del dispositivo como salida de la transmisión.

Al examinar nuevamente la topología se hace evidente que, para alcanzar a R1, R2 debe utilizar su interfaz *FastEthernet* 0/1.

Figura 75. **Ruta estatica para alcanzar a R1**

```
R2(config)# ip route 192.168.1.0 255.255.255.0 fastethernet 0/1
```

Fuente: elaboración propia.

Al inspeccionar la tabla de enrutamiento de R2 puede notarse que la red #1 (192.168.1/24) es alcanzable a través de la interfaz referida en la instrucción anterior.

Figura 76. **Tabla de enrutamiento R2**

```
R2# show ip route
Codes: 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
```

Continuación de la figura 76.

```
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

S       192.168.1.0/24 is directly connected, FastEthernet0/1
C       192.168.2.0/24 is directly connected, FastEthernet0/1
C       192.168.3.0/24 is directly connected, FastEthernet0/0
```

Fuente: elaboración propia.

Finalmente es necesario agregar que las rutas estáticas con una dirección IP como siguiente salto son preferibles a aquellas donde se especifica una interfaz de salida, especialmente en topologías de múltiple acceso.

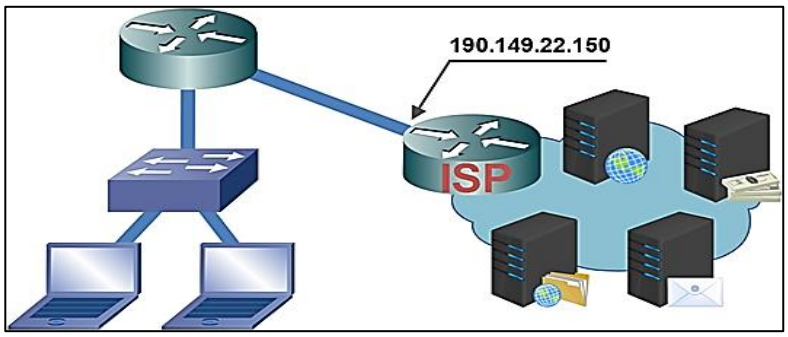
3.9.2. Ruta por defecto

Es una ruta de último recurso para evitar la pérdida de paquetes en el caso de que el *router* no encuentre una entrada más específica en su tabla de enrutamiento.

Son creadas utilizando la misma instrucción empleada para con las rutas estáticas, con la salvedad de que estas utilizan una dirección especial que cumple la función de comodín y que está compuesta por cuatro ceros por lo que recibe el nombre de *quad zero.*

A manera de mostrar un ejemplo de su implementación en la figura 77 se presenta la siguiente topología, en donde la red de una pequeña empresa se conecta a un proveedor de servicios de internet (Internet *Service Provider* -ISP-), siendo este un escenario común de la aplicación de rutas por defecto.

Figura 77. **Una ruta por defecto envía el tráfico al ISP**



Fuente: elaboración propia, empleando *Edraw Max*.

Para crear dicha ruta se utiliza la siguiente instrucción, donde la dirección *quad zero* es utilizada dos veces para indicar que los paquetes destinados a "cualquier red" usando "cualquier máscara", que no encuentren una ruta más específica en la tabla de enrutamiento, serán enviados a la dirección establecida.

Debido a este comportamiento este tipo de rutas, también son llamadas "de último recurso" (*last resort*).

Figura 78. **Instrucción para ruta por defecto**

```
RedInterna(config)# ip route 0.0.0.0 0.0.0.0 190.149.22.150
```

Fuente: elaboración propia.

Al examinar la tabla de enrutamiento del *router* en cuestión, la ruta por defecto aparece como una ruta estática seguida de un asterisco.

Figura 79. **Ruta por defecto en la tabla de enrutamiento**

```
RedInterna# show ip route
Codes: 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

Gateway of last resort is 190.149.22.150 to network 0.0.0.0

     190.149.0.0/16 is variably subnetted, 2 subnets, 2 masks
C       190.149.22.0/24 is directly connected, FastEthernet0/1
C       192.168.1.0/24 is directly connected, FastEthernet0/0
S*   0.0.0.0/0 [1/0] via 190.149.22.150
```

Fuente: elaboración propia.

La dirección del ISP será utilizada como último recurso (*Gateway of last resort*), para no descartar la información.

3.10. Enrutamiento dinámico

El enrutamiento estático es sencillo, pero no es muy escalable ni resiliente, por esta razón en redes más grandes es obligatorio el uso del enrutamiento dinámico a través de algún protocolo de enrutamiento.

Los protocolos de enrutamiento son capaces de descubrir y monitorear todas las rutas existentes en la topología, eligiendo de entre ellas la mejor, para luego incluirla en la tabla de enrutamiento automáticamente.

Para elegir la mejor ruta entre varias posibilidades, le es asignado a cada una de ellas un valor que indica la conveniencia o inconveniencia de ser

utilizada. Este valor es calculado a partir de criterios específicos de cada protocolo de enrutamiento y recibe el nombre de métrica. Las mejores rutas son aquellas con las menores métricas.

3.10.1. Clasificación de los protocolos de enrutamiento

Los protocolos de enrutamiento pueden clasificarse en dos categorías: vector distancia y estado de enlace. Siendo algunas de sus características las enumeradas a continuación:

- Protocolos vector distancia (RIP, IGRP)
    - Envían toda su tabla de enrutamiento a los *routers* vecinos periódicamente.
    - Métrica sencilla calculada a partir del número de saltos entre nodos.
    - Fáciles de configurar.
    - Presentan menos opciones.
    - Autosumarización en la frontera discontinua.
- Protocolos de estado de enlace (OSPF, IS-IS)
    - Envían la información de sus propias interfaces hacia todos los demás *routers* presentes en la topología.
    - Métrica compleja calculada a partir del ancho de banda.
    - Difíciles de configurar.
    - Presentan muchas más opciones.
- Protocolo vector distancia avanzado o mejorado (EIGRP)

- Envían actualizaciones parciales de su tabla de enrutamiento conforme estas son requeridas.
- Métrica compleja calculada por defecto a partir del ancho de banda y el retraso introducido por las interfaces, con otros parámetros opcionales.
- Fácil de configurar.
- Presenta muchas opciones.
- Originalmente un protocolo propietario de Cisco, hoy es un estándar abierto. Sin embargo, no ha sido implementado por ningún otro fabricante.
- Autosumarización en la frontera discontinua.

La elección de un protocolo de enrutamiento dependerá del tamaño de la red, requerimientos y políticas organizacionales, siendo posible la utilización de varios de estos protocolos dentro de una misma institución o empresa.

3.10.2. Bucles de enrutamiento (*routing loops*)

Al aumentar el tamaño y la complejidad de la red crece también la probabilidad de la aparición de ciertos problemas dentro de la misma, uno de los cuales es la ocurrencia de los llamados bucles de enrutamiento, donde los paquetes se encuentran atrapados dentro de cierta ruta que atraviesan una y otra vez incapaces de llegar a su destino.

Para paliar este problema, los protocolos implementan ciertos mecanismos, algunos de ellos son específicos de una categoría a continuación se presentan los siguientes ejemplos.

- *Time to Live* (TTL): es un campo presente en el paquete IP, con un contador o número que se decrementa en cada salto para asegurar que un paquete no se quede estancado en un bucle infinito.

- *Hold down timers*: tiempo de espera que los *routers* utilizan antes de actualizar sus rutas.

- Horizonte dividido (*split horizon*): para protocolos vector distancia es una regla que establece que una ruta no puede ser publicada por la misma interfaz por la que fue aprendida.

- Envenenamiento en reversa (*poison reverse*): para protocolos vector distancia es una excepción a la regla del horizonte dividido, en donde se marcará (o envenenará) una ruta con una métrica inalcanzable.

3.10.3. Comportamiento *Classful* y *Classless*

La tabla de enrutamiento, así como algunos de los protocolos dinámicos presentados anteriormente pueden comportarse en una de las siguientes maneras:

- *Classful:* el comportamiento original de todos los dispositivos, donde se suponía que los límites impuestos entre las clases de direcciones IP (A, B, C, D y E) siempre serían respetados, por lo que las rutas (almacenadas o transmitidas) no incluían la información proporcionada por la máscara de subredes ya que no era necesaria en ese tiempo para poder identificar cómo estaban divididas dichas direcciones.

Este es el comportamiento por activo, por defecto en los sistemas operativos de los *routers* Cisco anteriores a la versión 12, siendo utilizado por algunos de los primeros protocolos de enrutamiento como RIPv1 e IGRP y que presentaba algunos problemas que luego serían heredados a sus sucesores RIPv2 y EIGRP por razones de compatibilidad entre los mismos, por lo que su estudio todavía es necesario.

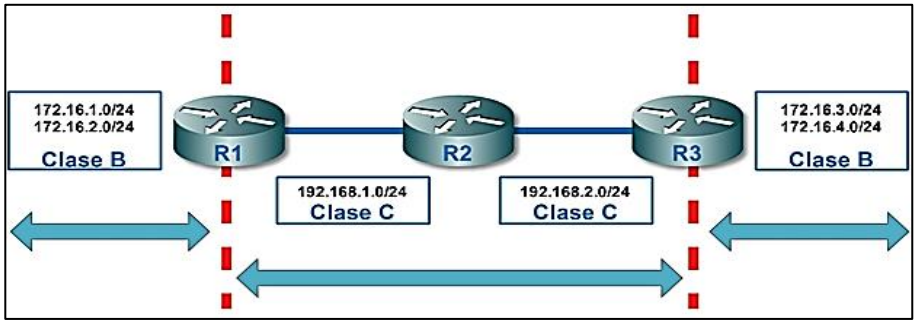
- *Classless*: la manera en cómo se comportan los dispositivos actualmente, en donde toda ruta es almacenada o transmitida junto con su máscara de subred, lo permite el uso de una máscara de longitud variable (VLSM) para un uso más eficiente de las direcciones existentes.

3.10.4. Autosumarización en la frontera discontinua

Es uno de los problemas derivados de un comportamiento *classful* que afecta a los sucesores de los primeros protocolos de enrutamiento: RIP y EIGRP.

Se presenta en topologías que utilizan redes discontinuas, lo que significa que dos o más redes adyacentes se encuentran utilizando una clase de direccionamiento distinta y acorde a los límites que originalmente fueron establecidos para las clases A, B, C, D y E, como se ilustra a continuación:

Figura 80. **Topología con subredes discontinuas**



Fuente: elaboración propia, empleando *Edraw Max*.

La autosumarización pretendía reducir el tamaño de la tabla de enrutamiento de los dispositivos al anunciar varias subredes como una sola ruta (proceso que recibe el nombre de sumarización o agregación de rutas), respetando los límites *classful.*

En el ejemplo presentado, R1 con conocimiento de las redes 172.16.1.0/24 y 172.16.2.0/24, se encuentra en una frontera discontigua (entre esquemas de direccionamiento B y C). por lo que al anunciar estas al dispositivo vecino (R2) realizará una autosumarización incluyendo ambas redes dentro de una sola ruta respetando los límites *classful*: 172.16.0.0/16. Lo que indicará a los demás dispositivos que R1 posee una manera de alcanzar a todas aquellas rutas cuyos primeros dos octetos sean 172 y 16.

Siguiendo el mismo proceso, R3 con conocimiento de las redes 172.16.3.0/24 y 172.16.4.0/24 y que también se encuentra en una frontera discontinua efectuará de manera automática la misma sumarización anunciando la ruta 172.16.0.0/16 a los dispositivos vecinos.

El problema, evidente en este momento, es que R2 recibirá actualizaciones de la red 172.16.0.0/16 desde dos puntos diferentes de la topología, lo que ocasionará problemas de conectividad cuando se quiera acceder a una subred específica.

A pesar de lo explicado en este apartado, la sumarización no es una técnica que deba evitarse, ya que su uso presenta muchos beneficios, simplemente no es conveniente dejar que la misma sea realizada automáticamente, por lo que es una mejor práctica deshabilitar esta característica cuando se utilicen los protocolos mencionados.

3.10.5. *Routing Information Protocol* (RIP)

Creado en 1988 es uno de los protocolos de enrutamiento más antiguos.

Perteneciente a la categoría vector distancia es un protocolo que reside en la capa de aplicación del modelo OSI empleando el puerto UDP 520 y utilizando el algoritmo *Bellman-Ford,* para hallar la ruta más corta entre dos puntos.

Empleado y restringido por una métrica sencilla basada en el número de saltos entre nodos, siendo una ruta con dieciséis saltos considerada inalcanzable o con una métrica infinita (otro mecanismo para prevenir bucles de enrutamiento), por lo que no puede ser utilizado en redes demasiado grandes.

Existen dos versiones de RIP

- Versión 1 (V1)
    - No soporta autenticación

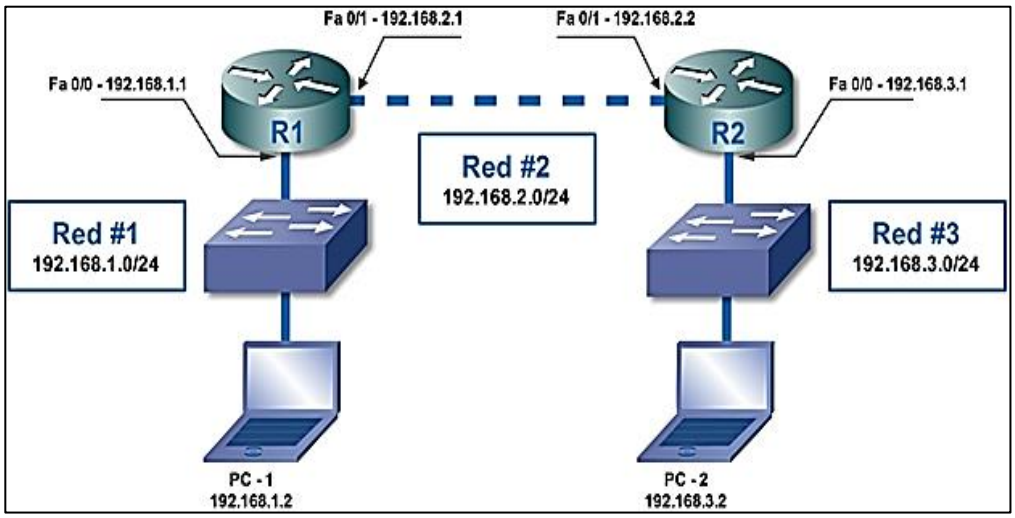
- Comportamiento *classful*
- Transmite sus actualizaciones como un *broadcast*

- Versión 2 (V2)

  - Soporta autenticación
  - Comportamiento *classless*, por lo que puede utilizar VLSM
  - Transmite sus actualizaciones utilizando la dirección de *multicast* 224.0.0.9

Se utilizará nuevamente la topología base, introducida en la sección anterior, para presentar un ejemplo de la implementación de este protocolo,

Figura 81. **Topología base para los ejemplos de las secciones de enrutamiento**



Fuente: elaboración propia, empleando *Edraw Max*.

La funcionalidad de un protocolo de enrutamiento puede ser habilitada en el modo de configuración global, siendo estos precedidos por la palabra clave *router,* por lo que se puede utilizar la ayuda para indicar los protocolos disponibles como se muestra en la figura 82.

Figura 82. **Indicación de protocolos disponibles**

```
R1(config)#router ?

bgp    Border Gateway Protocol (BGP)
eigrp  Enhanced Interior Gateway Routing Protocol (EIGRP)
ospf   Open Shortest Path First (OSPF)
rip    Routing Information Protocol (RIP)
```

Fuente: elaboración propia.

Se habilitará RIP utilizando la versión 2 del protocolo y desactivando la autosumarización con el comando *no autosummary,* de acuerdo a la figura 83.

Figura 83. **Habilitación de RIP**

```
R1(config)# router rip
R1(config-router)# version 2
R1(config-router)# no autosummary
```

Fuente: elaboración propia.

Después de activar un protocolo de enrutamiento se debe configurar las redes que serán anunciadas o publicadas hacia los demás dispositivos, las que en este caso, serán aquellas conocidas por el *router* al estar conectadas directamente.

Para cumplir con este propósito se utilizará nuevamente el comando *network*, y que cumple dos funciones, la primera de ellas es indicar al *router* la red que formará parte de un proceso (en este caso RIP) y la segunda es seleccionar aquella interfaz del dispositivo con una dirección IP que pertenezca a dicha red, para hacer la conexión del proceso en cuestión con el mundo físico.

De modo que para anunciar la red #1 se puede ingresar la siguiente instrucción.

Figura 84. **Instrucción para anunciar la red núm. 1**

```
R1(config-router)# network 192.168.1.0
```

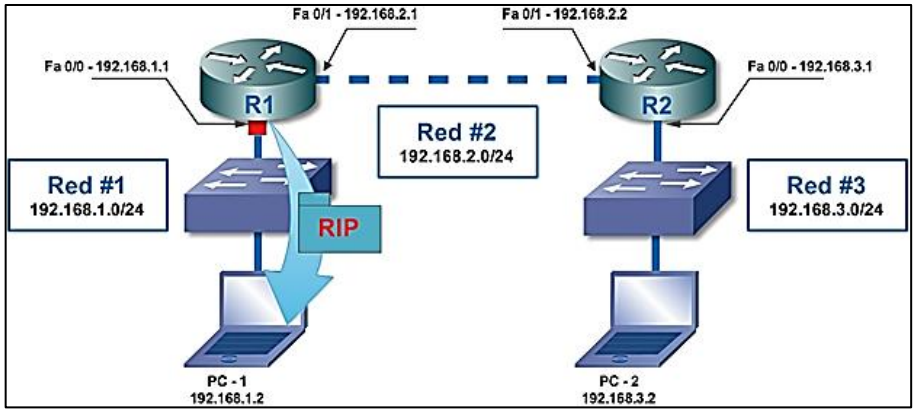
Fuente: elaboración propia.

El comando anterior le indica a RIP que agregue la red 192.168.1.0/24 a sus publicaciones y activa la interfaz *Fastethernet* 0/0 para empezar a enviar y recibir actualizaciones de este protocolo.

En el caso anterior que no es posible añadir una máscara de subred a la nueva publicación, esto es debido a la antigüedad de RIP, por lo que en la versión 2 de este protocolo, esta información será tomada de la interfaz física correspondiente (ej.: *Fastethernet* 0/0).

Al completar la instrucción anterior, R1 enviará y publicará actualizaciones de RIP a través de la interfaz *Fastethernet* 0/0, como se muestra en la figura 85.

Figura 85. **R1 anuncia y recibe publicaciones de RIP a través de la interfaz Fa 0/0**



Fuente: elaboración propia, empleando *Edraw Max*.

Es evidente, que a pesar de estar anunciando la red deseada, las publicaciones no se envían (ni tampoco reciben), entre los dos *routers.*

Para lograr que R1 envíe sus publicaciones a R2 será necesario incluir dentro del proceso RIP (utilizando el comando *network*), la interfaz conectada entre ambos dispositivos, en este caso *Fastethernet* 0/1, como se muestra en la figura 86.

Figura 86. **Interfaz conectada**

```
R1(config-router)# network 192.168.2.0
```

Fuente: elaboración propia.

En esta situación no se pretende utilizar RIP para anunciar la red 192.168.2.0/24 a R2, ya que ambos *routers* conocen esta red al estar directamente conectados, sino habilitar la interfaz que pertenece a esa red, ya que es la que interconecta ambos dispositivos para que reciba y publique las actualizaciones de RIP.

Para lograr conectividad entre ambos ordenadores es necesario configurar R2 para que este pueda intercambiar información (o rutas) con R1, como se exhibe en la figura 87.

Figura 87. **Configuración de RIP en R2**

```
R2(config)# router rip
R2(config-router)# version 2
R2(config-router)# no auto-summary
R2(config-router)# network 192.168.2.0
R2(config-router)# network 192.168.3.0
```

Fuente: elaboración propia.

Cuando todos los *routers* posean en su tabla de enrutamiento la información de todas las redes que conforman la topología se podrá decir que la red ha convergido.

Es necesario resolver el problema de seguridad que proviene del hecho de que los *routers* están enviando y recibiendo actualizaciones por las interfaces conectadas hacia las redes de los usuarios. Como última consideración de esta sección, en donde un posible atacante podría incorporar o emular a través de *software* un nuevo dispositivo, con el fin de conocer la estructura de la topología de red o evadir medidas de seguridad.

Por la imposibilidad de desactivar estas interfaces retirando el comando *network* (ej.: *no network 192.168.1.0*), ya que las redes de los usuarios no serían publicadas, se puede reducir o eliminar esta vulnerabilidad utilizando interfaces pasivas, las cuales son interfaces físicas que no envían ni reciben publicaciones a pesar de pertenecer a una red que está siendo anunciada y que pueden configurarse dentro del protocolo de enrutamiento como se muestra en la figura 88 y 89.

Figura 88. **Configuración de interfaces pasivas 1**

```
R1(config)#router rip
R1(config-router)#passive-interface fastethernet 0/0
```

Fuente: elaboración propia.

Figura 89. **Configuración de interfaces pasivas 2**

```
R2(config)#router rip
R2(config-router)#passive-interface fastethernet 0/0
```

Fuente: elaboración propia.

Es importante mencionar, en el caso específico de RIP, las interfaces pasivas dejan de enviar, pero todavía son capaces de aceptar nuevas publicaciones.

Para visualizar los protocolos de enrutamiento activos dentro de un dispositivo, junto con las redes publicadas, interfaces pasivas y otros parámetros específicos a cada protocolo, se puede ejecutar el comando *show ip protocols* como se muestra a continuación.

Figura 90. **Ejecución del comando *show ip protocols***

```
R1# show ip protocols

Routing Protocol is "rip"

Sending updates every 30 seconds, next due in 23 seconds
Invalid after 180 seconds, hold down 180, flushed after 240
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set

Redistributing: rip
Default version control: send version 2, receive 2
Interface Send Recv Triggered RIP Key-chain
FastEthernet0/1 2 2

Automatic network summarization is not in effect
Maximum path: 4

Routing for Networks:
192.168.1.0
192.168.2.0

Passive Interface(s):
FastEthernet0/0

Routing Information Sources:
Gateway Distance Last Update
192.168.2.2 120 00:00:19
Distance: (default is 120)
```

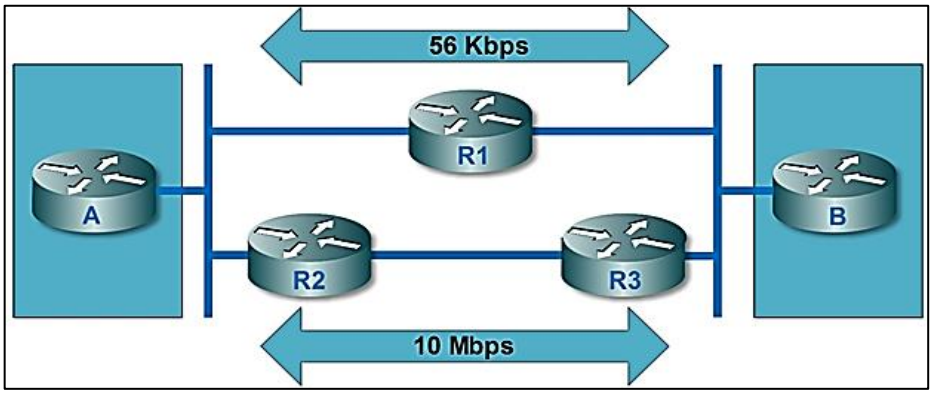
Fuente: elaboración propia.

3.10.6. Funcionamiento de la tabla de enrutamiento

En este punto ya se conoce que la función de la tabla de enrutamiento es almacenar las mejores rutas hacia todos los posibles destinos dentro de una red, sin embargo, todavía no se han considerado todos los criterios utilizados para seleccionar entre varias rutas la mejor, para que esta pueda ser instalada dentro de esta tabla.

Para iniciar la discusión se presenta la siguiente topología donde todos los dispositivos son capaces de ejecutar uno o más protocolos de enrutamiento de manera simultánea y donde existen dos caminos o rutas para la comunicación entre los dispositivos A y B.

Figura 91. **Topología con dos rutas entre A y B**

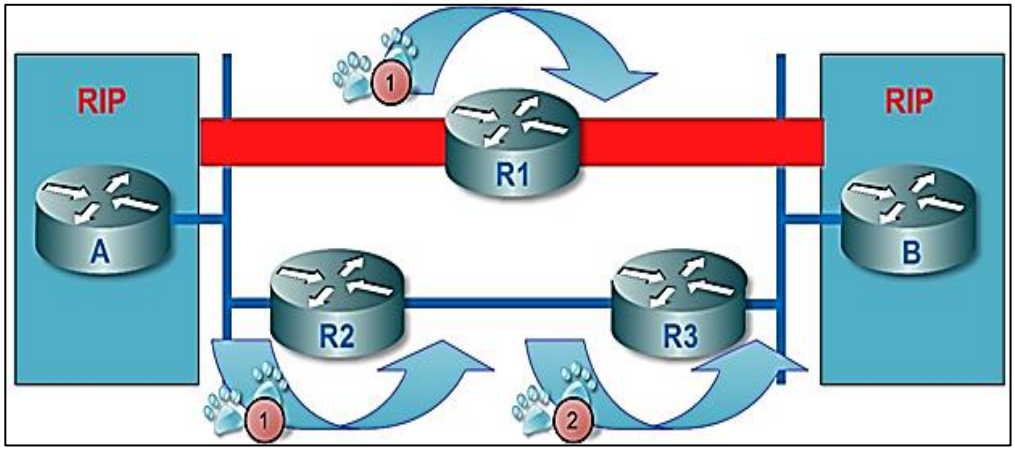


Fuente: elaboración propia, empleando *Edraw Max*.

En el caso de que se esté ejecutando un solo protocolo de enrutamiento en toda la topología, la elección de la mejor ruta dependerá de la métrica (o medida) utilizada por el protocolo correspondiente, en el caso de RIP es el número de saltos mientras que en OSPF es el ancho de banda.

En el ejemplo anterior, de utilizarse RIP se elegirá el camino a través de R1 a pesar de la gigantesca diferencia de ancho de banda con la ruta alternativa, ya que RIP utiliza como única medida de comparación el número de saltos. Mientras que al emplear OSPF, un protocolo más moderno, se elegirá la ruta más rápida que atraviesa R2 y R3.

Figura 92. **RIP prefiere la ruta con menor número de saltos**



Fuente: elaboración propia, empleando *Edraw Max*.

Si dos o más protocolos de enrutamiento se encuentran ejecutándose al mismo tiempo dentro de un dispositivo, cada uno de ellos presentará, acorde a su métrica, las mejores rutas entre un punto y otro, por lo que en primera instancia será necesario decidir el protocolo a utilizar, elección que se realiza a través de un parámetro que indica el grado de confiabilidad de los mismos y que recibe el nombre de distancia administrativa.

La distancia administrativa es un parámetro de carácter local dentro de cada uno de los dispositivos, entre más bajo es este parámetro más confiable se considera al protocolo, un listado con los valores más comunes (utilizados por Cisco) se presenta a en la tabla XII.

Tabla XII. **Distancias administrativas más comunes (Cisco)**

	Distancia administrativa
RIP	120
OSPF	110
EIGRP	90
Estática	1
Directamente conectada	0

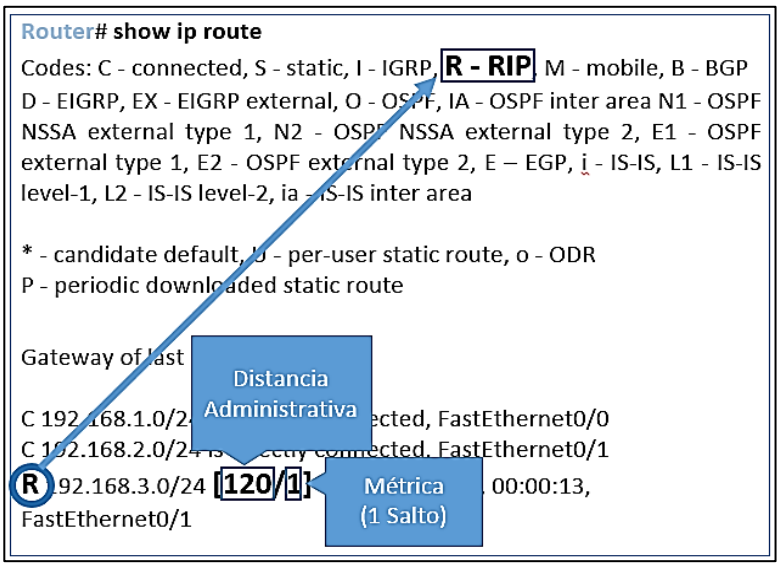
Fuente: elaboración propia.

En un dispositivo ejecutando RIP y OSPF en donde ambos protocolos presenten diferentes alternativas para la conexión entre dos puntos se preferirán aquellas aprendidas por OSPF, ya que tienen una menor distancia administrativa, por lo que son más confiables.

Si en el mismo caso se incluyera una ruta estática, sería esta última la que se instalase en la tabla de enrutamiento al poseer una menor distancia administrativa que OSPF.

Tanto la distancia administrativa como la métrica son visibles en las rutas que se encuentran en la tabla de enrutamiento, como se muestra en la figura 93.

Figura 93. **Enrutamiento de un *router* ejecutando RIP**



Fuente: elaboración propia, empleando *Edraw Max*.

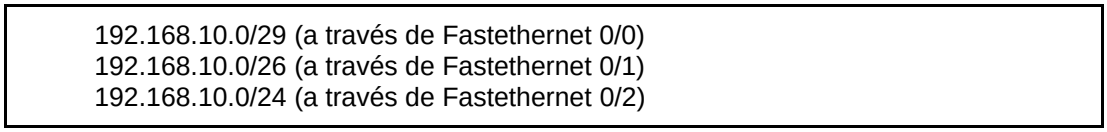
Otro factor, tomado en cuenta la tabla de enrutamiento, en la elección de la mejor ruta es la máscara de subred.

La información de la máscara de subred, también referida como la longitud del prefijo, acompaña a cada una de las rutas (en toda implementación moderna) y juega un rol muy importante en la toma de decisiones.

Si una ruta se presenta varias veces acompañada por distintas máscaras de subred, se considerará a cada una de sus versiones como un destino diferente y todas podrán coexistir al mismo tiempo en la tabla de enrutamiento.

Por ejemplo, es posible que se presente la situación donde se encuentren las siguientes rutas instaladas.

Figura 94. **Tres rutas presentes en una tabla de enrutamiento**



Fuente: elaboración propia.

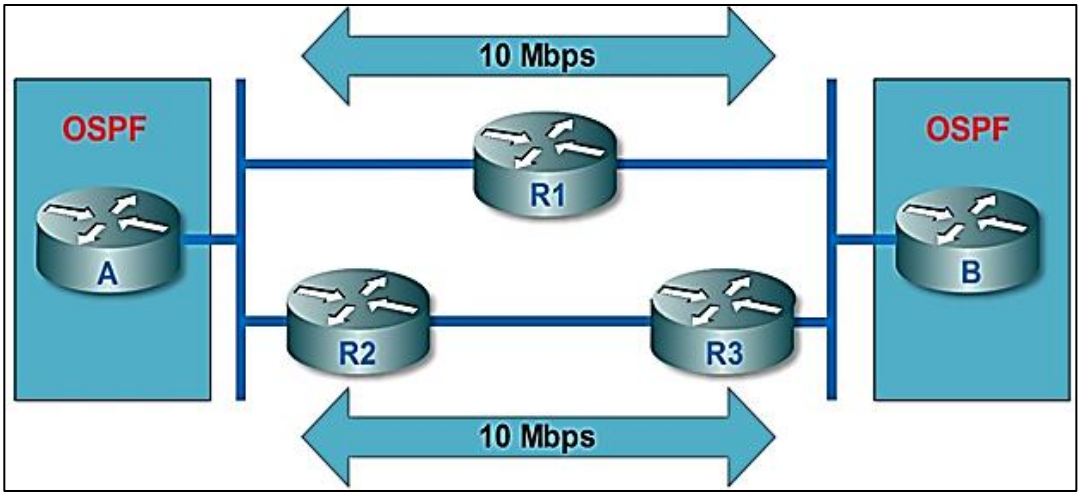
Dentro de las tres rutas que se encuentren presentes, la tabla de enrutamiento siempre elegirá la más específica. Mientras más largo sea el prefijo (o entre más bits con un valor de uno se encuentren presentes en la máscara de subred), más específica será la ruta. Por lo tanto, los paquetes destinados a la dirección 192.168.10.1 serán enviados a través de la interfaz *Fastethernet* 0/0.

Otra función del *router* es verificar en cada ruta, que la dirección IP del siguiente salto (la que se utilizará para llegar a ese destino en particular) sea válida y pueda ser alcanzada; condición que debe cumplirse en cualquier momento, para que una ruta pueda ser instalada y mantenida dentro de la tabla de enrutamiento.

3.10.7. Balanceo de cargas

Otra situación que puede presentarse es que el *router* encuentre dos o más rutas idénticas acorde a los criterios presentados en la sección anterior, como se muestra en la figura 95.

Figura 95. **Dos rutas aprendidas por el mismo protocolo con la misma métrica**

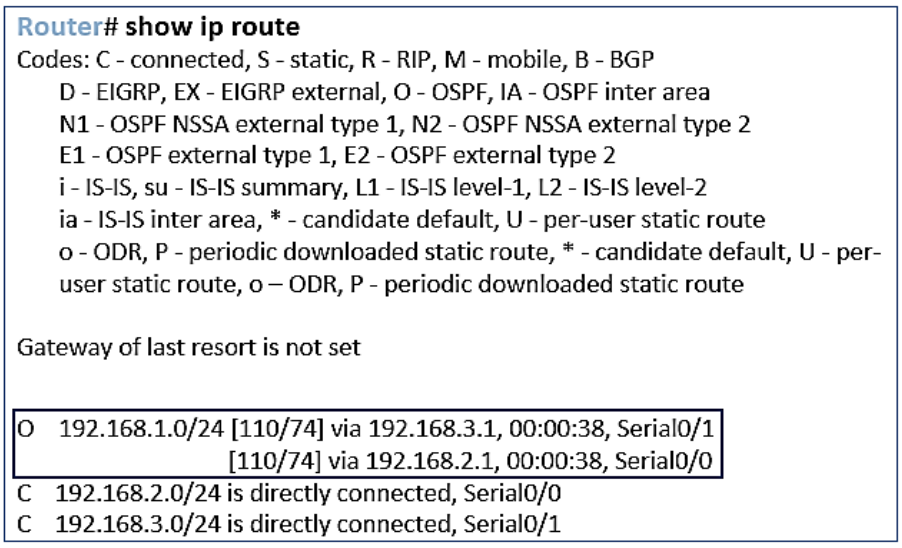


Fuente: elaboración propia, empleando *Edraw Max*.

En ese caso, en lugar de escoger entre rutas idénticas, el *router* las instalará todas dentro de su tabla de enrutamiento provocando un fenómeno conocido como balanceo de cargas, en donde el tráfico será distribuido de manera equivalente entre todas las rutas, esto recibe el nombre de balanceo simétrico.

El balanceo entre rutas desiguales o balanceo asimétrico solo es posible utilizando EIGRP.

Figura 96. **Balanceo de carga entre dos rutas en OSPF**



Fuente: elaboración propia.

3.11. *Open shortest path first* (OSPF)

Es el protocolo de enrutamiento más popular del mundo,creado a principios de 1990, de la categoría estado de enlace, utiliza el algoritmo *Shortest Path First* (SPF) creado por Edsger Dijkstra en 1956.

Se identifica con el número de protocolo 89 y trabaja en la capa de red del modelo OSI (no utiliza TCP/UDP ni un número de puerto), por lo que recurre a otros medios para lograr transmisiones confiables. Presenta un comportamiento *classless,* soporta autenticación y envía sus actualizaciones a la dirección de *multicast* 224.0.0.5.

A diferencia de RIP, que envía su tabla de enrutamiento completa a los *routers* vecinos y que tiene una visión limitada de la red, OSPF envía la

información de sus vecinos a todos los demás *routers* presentes en la topología, por lo que cada uno de los dispositivos ejecutando OSPF conoce la disposición de todos los demás dentro de la red, lo que le permite a cada *router* calcular la ruta más corta y libre de bucles hacia un destino en particular, con la única desventaja que el proceso (la ejecución del algoritmo SPF) representa una carga más pesada para el procesador.

Posee una métrica basada en el ancho de banda llamada costo, calculada como se describe en la figura 97.

Figura 97. **Fórmula para calcular el costo**

$$costo = \frac{ancho\ de\ banda\ de\ referencia}{ancho\ de\ banda\ de\ la\ interface}$$

Fuente: elaboración propia.

El ancho de banda de referencia utilizado por Cisco es de 100 Mbps, por lo que el costo para una interfaz *FastEthernet* (100 Mbps) será de 1, mientras que para una interfaz *ethernet* (10 Mbps) será de 10. La métrica se incrementa conforme las actualizaciones atraviesan diferentes segmentos de la red, siendo la mejor ruta aquella que tenga menor costo.

3.11.1. Tablas mantenidas por OSPF

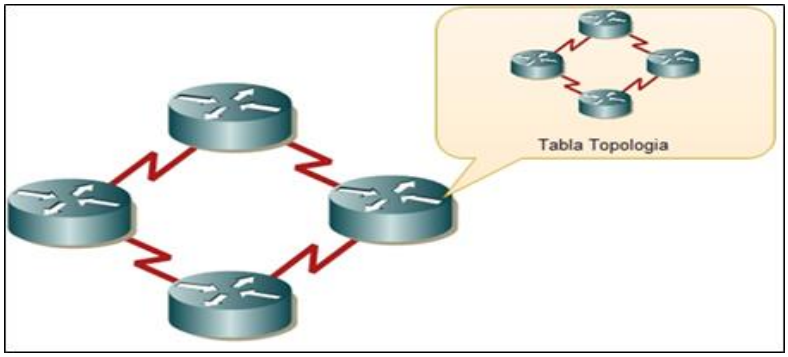
OSPF mantiene tres tablas para almacenar información.

- Tabla de enrutamiento: donde se almacenan las mejores rutas y que sigue el mismo comportamiento y respeta los mismos criterios que se han presentado anteriormente.

- Tabla de vecinos (*neighbor table*): donde se almacena la información de todos los dispositivos que comparten una red común (a través de alguna de sus interfaces), con el dispositivo y que han cumplido con los requisitos para establecer una vecindad.

- Tabla de topología (*link state database* o LSDB): donde se almacena la información de todos los demás dispositivos que se encuentren en la red y que estén ejecutando OSPF.

Figura 98. **La tabala de topología (LSDB) muestra la disposición de los otros *routers* dentro de la red**



Fuente: elaboración propia, empleando *Edraw Max*.

3.11.2. Funcionamiento basado en áreas

En un principio, dos de los más grandes problemas que presentaba la implementación de OSPF en redes, con una gran cantidad de dispositivos, eran el tamaño de la tabla de topología (que debía tener una imagen completa de toda la red) y la cantidad de actualizaciones, ya que un dispositivo debía comunicarse con todos los demás.

Para paliar estos problemas, tomando en cuentas, las limitaciones computacionales y de ancho de banda de la época, se dispuso que el funcionamiento de OSPF se dividiera dentro de distintos tipos de áreas cuya función consistiría en acotar la red para reducir el tamaño de las tablas de enrutamiento y topología y servir como contención para diversos tipos de actualizaciones.

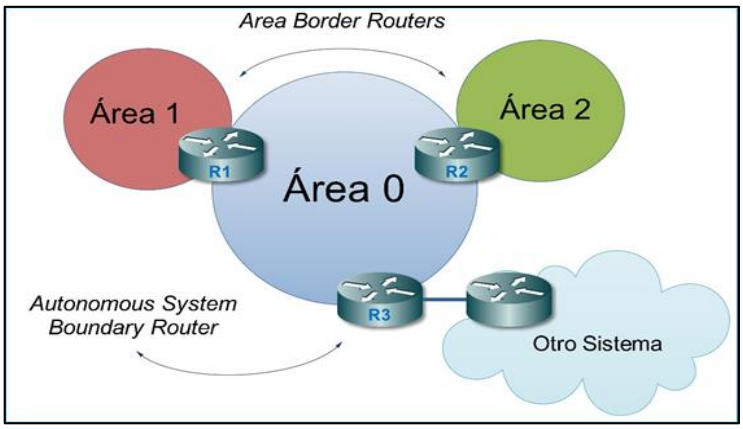
Actualmente, hay varios tipos de áreas, algunas de ellas son definidas en el estándar mientras que otras existen gracias a extensiones propietarias. Los dos tipos esenciales se definen a continuación.

- Área 0, columna vertebral o *backbone*: es fundamental y punto de referencia que debe existir en todas las implementaciones de OSPF con más de un área. Todas las demás áreas deben conectarse al área 0 obligatoriamente para prevenir bucles de enrutamiento entre estas.

- Área estándar: cualquier otra área identificada con otro número, dentro de la cual las tablas de topología (LSDB) de los dispositivos tienen la información de todas las rutas que componen la red. Este es el comportamiento por defecto.

En OSPF, la información presente en las tablas de enrutamiento y topología de un *router* dependerá del tipo de área donde se encuentre, pudiendo inclusive, pertenecer a varias áreas al mismo tiempo, en cuyo caso pasará a tomar un rol especial, como se muestra utilizando la topología que se describe en la figura 99.

Figura 99. **Sistema ejecutando OSPF dividido en tres áreas**



Fuente: elaboración propia, empleando *Edraw Max*.

En el ejemplo, tanto R1 como R2 pertenecen y se encuentran entre dos áreas diferentes sirviendo como frontera entre las mismas, por lo que son considerados área *border routers* (ABR).

La función de los ABR consiste en evitar la propagación indiscriminada de las actualizaciones entre distintas áreas y limitar el tamaño de la tabla de topología al ser puntos naturales de sumarización de rutas.

De acuerdo con el ejemplo, el propósito de R2 es conocer todas las rutas existentes en el área 0 y área 2, para luego limitar el conocimiento de la red de todos los demás dispositivos pertenecientes a las áreas mencionadas a través de una sumarización de rutas manual. De esta manera cualquier *router* perteneciente al área 0, tendrá un conocimiento restringido a los otros dispositivos que se encuentren dentro su misma área y una sola ruta para alcanzar el área 2 a través de R2, De igual forma, los *routers* pertenecientes al área 2 conocerán solamente acerca de los dispositivos dentro de su misma área y contarán con una sola ruta hacia el área 0 a través del mismo *router*.

Como consecuencia, la tabla de topología de los dispositivos será más pequeña, lo que acelera el tiempo de convergencia de la red y representan menos carga para el procesador de los mismos.

Asimismo, R2 limitará el flujo de actualizaciones entre el área 0 y el área 2.

Si una nueva ruta aparece o desaparece, una actualización será enviada a todos los dispositivos forzándolos a tomar la información contenida en su tabla de topología y a ejecutar el algoritmo SPF para ajustarse dinámicamente al cambio.

En su calidad de ABR, R2 evitará que este tipo de actualizaciones se propaguen de un área a otra. En ese sentido, cualquier cambio queda contenido dentro del área en donde haya ocurrido. Si un cambio ocurre dentro del área 0 solo afectará a los dispositivos que se encuentren en esa área, en otras palabras, si un *router* falla dentro del área 0 los dispositivos del área 2 nunca lo sabrán. De este modo se consigue una red más estable, ya que un cambio no provocará un recálculo de las mejores rutas en cada uno de los *routers* presentes en la red, lo que es una condición deseable en el caso de que se presenten problemas, tales como los ocasionados por una *flapping interface,* nombre que se le da a una interfaz que cambia constantemente de estado (encendida/apagada).

Un análisis idéntico puede llevarse a cabo con R1.

Finalmente se considera el funcionamiento de R3, que también es un *router* frontera, no entre áreas, sino con otro sistema, ya sea otra organización o proveedor de servicios u otra parte de la misma infraestructura ejecutando un

protocolo de enrutamiento diferente, por lo que recibe el nombre de *autonomous system boundary router* (ASBR).

3.11.3. Tipos de paquetes

OSPF utiliza 5 tipos diferentes de paquetes:

- *Hello:* se utiliza para el descubrimiento, formación y mantenimiento de vecindades con otros dispositivos.

- *Database description (*DBD*):* verifica que el dispositivo vecino tenga la misma tabla de topología o LSDB.

- *Link state request* (LSR): solicita información específica de la LSDB del vecino.

- *Link state update* (LSU): envía la información solicitada por el LSR. Sirve como contenedor a los diferentes tipos de actualizaciones o *Link State Advertisements* (LSA).

- *Link state acknowledgement* (LSAck): un paquete especial que sirve como acuse de recibo para comprobar que un paquete ha sido recibido con éxito para lograr una transmisión confiable.

3.11.4. Requerimientos

Para configurar OSPF hay que tener en cuenta los siguientes requerimientos.

- Un identificador de proceso (*process ID*): un dispositivo puede ejecutar varias instancias de OSPF al mismo tiempo. El identificador de proceso es un número asignado por el usuario a manera de poder distinguir entre ellos.

- Un número de área: cada interfaz habilitada para enviar y recibir actualizaciones tiene que pertenecer necesariamente a un área específica.

- Un identificador para el *router* (*router ID*): el requerimiento más importante es el nombre utilizado para distinguir un *router* de otros en la topología. No pueden existir dos *router ID* idénticos dentro de una misma red, y sin el mismo, el proceso OSPF no puede iniciar.

  Toma el formato de una dirección IPv4, lo cual puede ser confuso para los principiantes, ya que un dispositivo no es necesariamente alcanzable si se toma el identificador que aparece en la tabla de topología y se utiliza como una dirección IP. Por ejemplo, el hecho de que un *router* aparezca identificado como 10.10.10.10 en la tabla de topología no implica necesariamente que el dispositivo posea una ruta capaz de alcanzar la dirección IP equivalente.

  La elección del *router ID* se realiza utilizando la siguiente secuencia:

- Es posible establecer manualmente el identificador del *router* a través del comando *router-id.*

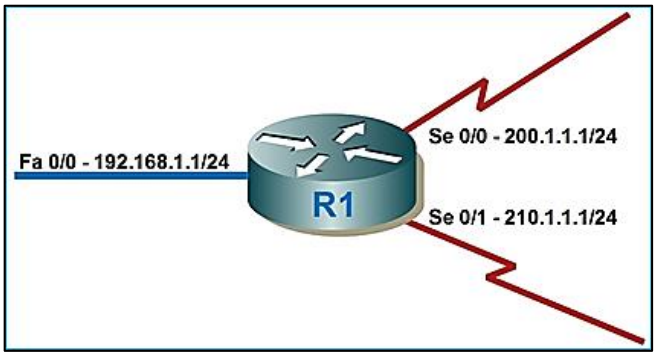
- Si no se establece manualmente. entonces se utiliza la dirección IP de la interfaz virtual o de *loopback* más alta. Las interfaces de *loopback*

ofrecen la gran ventaja de siempre permanecer encendidas al existir solamente a nivel lógico, por lo que son la práctica recomendada en muchas implementaciones.

- Si no existen interfaces de *loopback* se toma la dirección IP más alta de aquellas que estén encendidas dentro del dispositivo.

A manera de ejemplo se presenta el siguiente dispositivo:

Figura 100. **Elección *router* ID–Interfaz física**



Fuente: elaboración propia, empleando *Edraw Max*.

En este caso, al no configurarse manualmente y no existir una interfaz de *loopback* el *router ID* de R1 será la dirección IP más alta dentro de las interfaces encendidas: 210.1.1.1

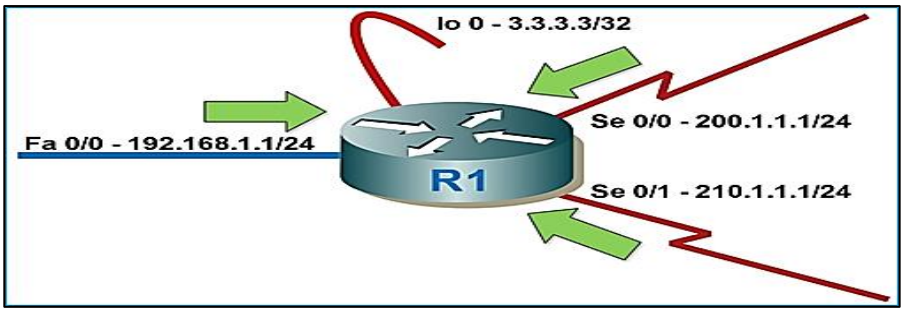
Para utilizar una interfaz virtual en su lugar, es posible agregar una interfaz de *loopback,* según se describe en la figura 101.

Figura 101. **Agregar una interfaz de *loopback***

```
R1(config)# interface loopback ?
<0-2147483647> Loopback interface number

R1(config)# interface loopback 0
%LINK-5-CHANGED: Interface Loopback0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up

R1(config-if)# ip address 3.3.3.3 255.255.255.255
```

Fuente: elaboración propia.

Figura 102. **Elección *router* ID–interfaz virtual**

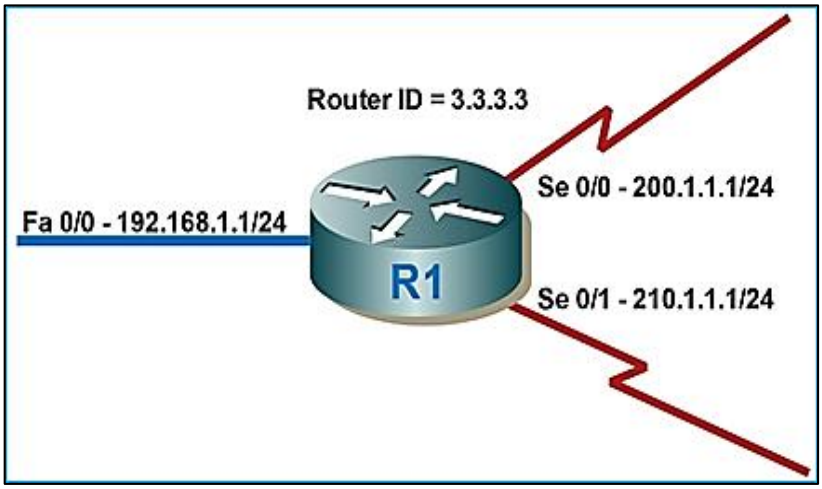


Fuente: elaboración propia, empleando *Edraw Max*.

En esta ocasión, después de iniciar o reiniciar el proceso OSPF, el identificador de R1 será 3.3.3.3. Si se supone que R1 se encuentra dentro de una infraestructura con cierto grado de redundancia y que se tiene conectividad hacia su interfaz de *loopback,* entonces es posible apreciar que la única manera de perder contacto con dicha interfaz es si todos los caminos que llevan a R1 dejan de estar disponibles, lo que ejemplifica los beneficios de utilizar una interfaz virtual o lógica sobre una física para manejar dispositivos dentro de una red.

Finalmente es posible configurar manualmente el identificador usando el comando *router-id.*

Figura 103. **Elección manual del *router* ID**

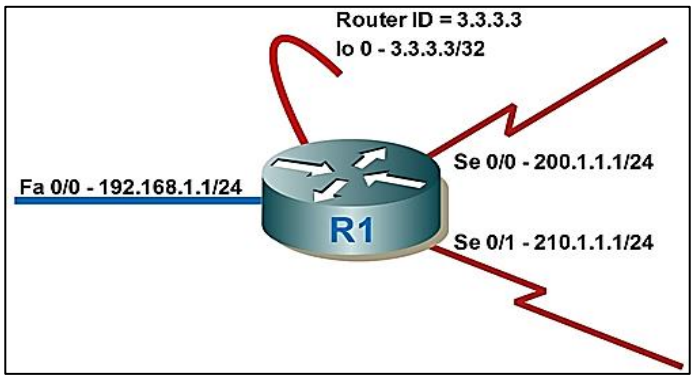


Fuente: elaboración propia, empleando *Edraw Max*.

A pesar de que el *router ID* ha sido configurado, es imposible tener comunicación hacia la dirección IP 3.3.3.3, ya que no existe ninguna interfaz física o virtual que tenga asignada dicha dirección.

A la hora de implementar OSPF y elegir un *router ID* para cada uno de los dispositivos se recomienda una combinación de las últimas dos opciones presentadas. Crear una interfaz de *loopback* con una dirección alcanzable en toda la red y luego asignar manualmente el identificador utilizando la misma dirección IP de la interfaz virtual.

Figura 104. **Elección *router* ID–Configuración recomendada**



Fuente: elaboración propia, empleando *Edraw Max*.

De esta manera se asegura que todos los dispositivos utilicen siempre el mismo *router ID* (ya que en un futuro podrían agregarse más interfaces virtuales), y que se pueda usar la dirección IP equivalente para alcanzar a los mismos desde cualquier punto de la topología.

3.11.5. Vecindades y adyacencias

Las vecindades son establecidas a través de los paquetes *hello,* los cuales se envían a través de todas las interfaces configuradas para formar parte del proceso OSPF en un intervalo regular de tiempo. Dichos paquetes incluyen una gran variedad de información, una lista parcial se muestra a continuación en donde todos los campos resaltados deben coincidir en ambos dispositivos para que se pueda establecer una relación de vecindad. Los campos resaltados deben ser idénticos entre dos dispositivos para que estos puedan iniciar una relación de vecindad

Tabla XIII. **Lista parcial del contenido del paquete *hello***

Máscara de Subred
Router ID
Temporizadores o *Timers*
Área
Vecinos

Fuente: elaboración propia.

Los campos resaltados deben ser idénticos entre dos dispositivos para que estos puedan iniciar una relación de vecindad. Después de establecer una vecindad y dependiendo del tipo de red donde se encuentren (múltiple acceso o punto a punto), los dispositivos tratarán de establecer una adyacencia, la cual es una relación que se lleva a cabo con ciertos vecinos para intercambiar la información contenida en la tabla de topología.

El proceso para formar vecindades y adyacencias atraviesa las siguientes etapas (se recomienda consultar la sección tipos de paquetes presentada anteriormente):

Tabla XIV. **Etapas que atraviesa un dispositivo para formar vecindades y adyacencias**

1.	*Down*	No se han detectado vecinos.
2.	*Init*	Se recibe un paquete *hello.*
3.	*Two-Way*	Se establece la relación de vecindad.
4.	*Exstart*	Comienzo de la adyacencia.

Continuación de la tabla XIV.

5.	*Exchange*	Se envían paquetes DBD.
6.	*Loading*	Se intercambian paquetes LSR y LSU.
7.	*Full*	Los dos dispositivos tienen la misma información en la tabla de topología.

Fuente: elaboración propia.

3.11.6. *Wildcard Mask*

Es una máscara cuya función consiste en marcar bits de una dirección IP para indicar cuáles de ellos serán sujetos a comparación; son más antiguas que las máscaras de subred. Fueron creadas antes de la adopción de CIDR y desarrolladas a manera que pudieran ser implementadas fácilmente utilizando lenguaje ensamblador. Los valores de la misma funcionan como se muestra en la tabla XV.

Tabla XV. **Función de los bits en una *wildcard mask***

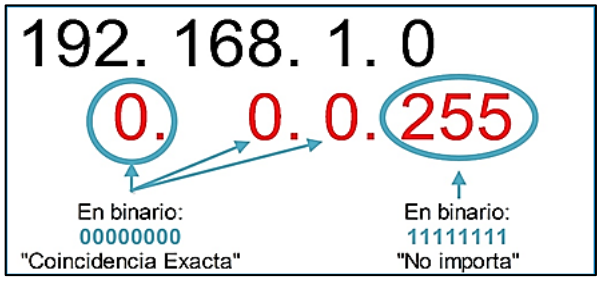
0	Exige una coincidencia exacta del bit equivalente.
1	No importa el valor del bit equivalente.

Fuente: elaboración propia.

Utilizadas actualmente por OSPF, EIGRP y en listas de control de acceso, fueron mantenidas por razones de compatibilidad con implementaciones más antiguas y por ser más versátiles que las máscaras de subred (de ahí el nombre *wildcard*), al no estar limitadas por las restricciones de las últimas y poder realizar una comparación bit a bit con una dirección IP, lo que permite hacer selecciones variadas con un mínimo de sentencias.

Por ejemplo, para seleccionar la red 192.168.1.0/24 es posible utilizar la *wildcard mask* 0.0.0.255 como se puede observar en la figura 105.

Figura 105. ***Wildcard Mask* para seleccionar la red 192.168.1.0/24**



Fuente: elaboración propia, empleando *Edraw Max*.

En el caso anterior, la *wildcard* elegida selecciona todas aquellas direcciones cuyos primeros tres octetos sean exactamente 192.168.1, abarcando efectivamente todas las direcciones requeridas. En la tabla XVI se muestran otros ejemplos.

Tabla XVI. **Ejemplo de la utilización de las *wildcard masks***

Dirección	*Wildcard*	Observaciones
192.168.1.1	0.0.0.0	Una *wildcard* compuesta enteramente por ceros selecciona una y solo una dirección IP.
192.168.1.0	255.255.255.255	Una *wildcard* compuesta enteramente por unos, selecciona todas las redes posibles. En este ejemplo se podría haber utilizado la sentencia: 0.0.0.0 255.255.255.255. Donde los cuatro ceros significan “cualquier red” y los resultados serían los mismos.
192.168.10.0	0.255.0.255	Esta *wildcard* selecciona todas las redes cuyo primer octeto sea 192 y el tercer octeto sea 10. Las *wildcard* discontinuas (en donde se pueden alternar ceros y unos) pueden aplicarse en listas de control de acceso, pero no en OSPF y EIGRP.

Fuente: elaboración propia.

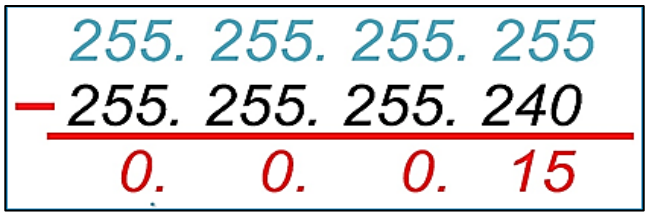
Para realizar implementaciones sencillas en donde quiera marcarse una red en particular, es posible calcular fácilmente la *wildcard* necesaria utilizando la siguiente fórmula:

Figura 106. **Fórmula para calcular la *wildcard mask***

Fuente: elaboración propia, empleando *Edraw Max*.

Por ejemplo, para calcular la *wildcard* necesaria para marcar las direcciones contenidas dentro de la subred 172.18.10.0 255.255.255.240 (/28).

Figura 107. **Cálculo de la *wildcard mask* para la red 172.18.10.0/28**



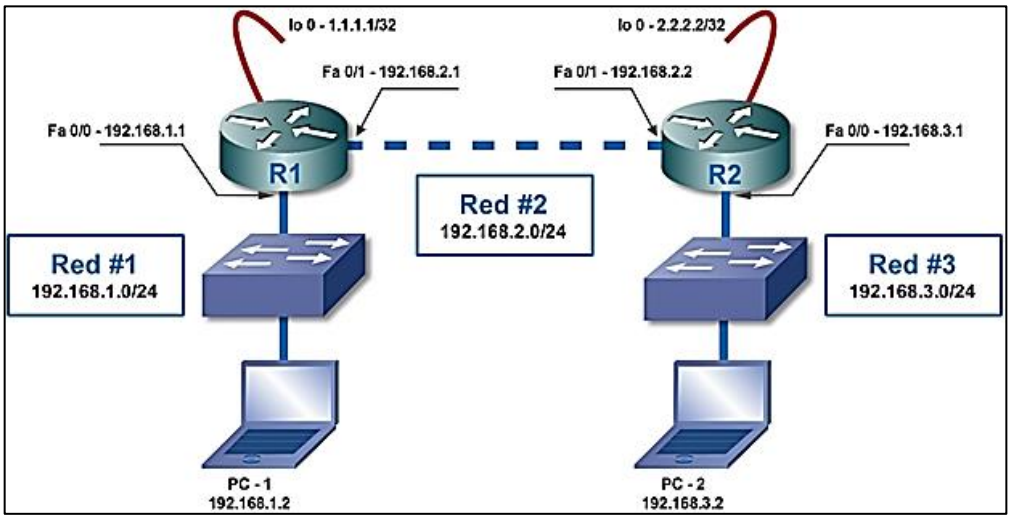
Fuente: elaboración propia, empleando *Edraw Max*.

Por la sencillez de este cálculo usualmente se refiere, aunque erróneamente, a la *wildcard mask* como el inverso de la máscara de subred.

3.11.7. Configuración

Para demostrar la implementación de OSPF se recurre nuevamente a la topología base utilizada en explicaciones anteriores, la única diferencia es que se han configurado interfaces de *loopback* en cada uno de los *routers* como se muestra a continuación en la figura 108.

Figura 108. **Topología base para los ejemplos de las secciones de enrutamiento con interfaces de *loopback***



Fuente: elaboración propia, empleando *Edraw Max*.

En este caso se configurará OSPF utilizando como identificador de proceso el número 1. Una vez iniciado el proceso, el identificador de R1 será 1.1.1.1 debido a la presencia de la interfaz de *loopback,* mismo que será configurado de manera manual con el comando *router-id,* siguiendo la sugerencia presentada anteriormente para evitar problemas a futuro.

Figura 109. **Configuración OSPF**

```
R1(config)# router ospf ?
  <1-65535>  Process ID

R1(config)# router ospf 1
R1(config-router)# router-id 1.1.1.1
```

Fuente: elaboración propia.

Para indicar las interfaces que serán parte del proceso OSPF se utiliza nuevamente el comando *network,* con las adiciones de la *wildcard mask* que introduce flexibilidad en la selección de direcciones y el número de área a las que las mismas serán asignadas.

En este *router* se utilizará una *wildcard mask* compuesta enteramente por **unos** (255.255.255.255), por lo que todas las redes existentes dentro del dispositivo, y por ende todas sus interfaces serán incluidas dentro del proceso. Esta práctica es válida solamente dentro de un laboratorio de pruebas, ya que en implementaciones reales podría resultar en la publicación no deseada de ciertas redes dentro de la topología.

Para complementar la configuración, también se configura la interfaz *Fastethernet* 0/0 para evitar que se envíen o reciban publicaciones a través de la interface donde están conectados los ordenadores de los usuarios. En el caso de OSPF, ya no se enviarán o recibirán paquetes *hello* lo que impide efectivamente la formación de vecinos sobre enlaces no deseados.

Figura 110. **Uso del comando *network* en R1**

```
R1(config-router)# network 0.0.0.0 255.255.255.255 area 0
R1(config-router)# passive-interface fastethernet 0/0
```

Fuente: elaboración propia.

La configuración de R2 se realizará de manera similar a la figura 110. Para indicar las interfaces que formarán parte del proceso se utilizará una *wildcard mask* compuesta enteramente por ceros (0.0.0.0) para activar única y exclusivamente las interfaces necesarias. Además se emplea el comando *passive-interface default* el cual configura todas las interfaces como pasivas, lo que obliga a revertir ese comportamiento sobre las interfaces donde se quiere formar vecindades con otros dispositivos. Estas medidas resultan en una implementación más segura de este protocolo.

Figura 111. **Configuracion de OSPF en R2**

```
R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# passive-interface default
R2(config-router)# network 192.168.2.2 0.0.0.0 area 0
R2(config-router)# network 192.168.3.1 0.0.0.0 area 0
R2(config-router)# network 2.2.2.2 0.0.0.0 area 0
R2(config-router)# no passive-interface fastethernet 0/1
```

Fuente: elaboración propia.

Una vez finalizada la configuración de R2 es posible establecer comunicación entre los ordenadores.

Al ver la tabla de enrutamiento de R1 se puede apreciar que esta posee las rutas necesarias.

Figura 112. **Tabla de enrutamiento de R1**

```
R1# show ip route

Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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

Gateway of last resort is not set

      1.0.0.0/32 is subnetted, 1 subnets
C     1.1.1.1 is directly connected, Loopback0
      2.0.0.0/32 is subnetted, 1 subnets
O     2.2.2.2 [110/2] via 192.168.2.2, 00:00:00, FastEthernet0/1
C   192.168.1.0/24 is directly connected, FastEthernet0/0
C   192.168.2.0/24 is directly connected, FastEthernet0/1
O   192.168.3.0/24 [110/2] via 192.168.2.2, 00:08:30, FastEthernet0/1
```

Fuente: elaboración propia.

Para ver los protocolos que están siendo ejecutados por el dispositivo, así como otros detalles importantes, puede emplearse la instrucción que se muestra en la figura 113.

Figura 113. **Instrucción para ver protocolos y detalles importantes**

```
R1# show ip protocols

 Routing Protocol is "ospf 1"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set
```

Continuación de la figura 113.

Router ID 1.1.1.1

Number of areas in this router is 1. 1 normal 0 stub 0 nssa

Maximum path: 4

Routing for Networks:

0.0.0.0 255.255.255.255 area 0

Passive Interface(s):

Vlan1

FastEthernet0/0

Loopback0

Routing Information Sources:

Gateway Distance Last Update

1.1.1.1 110 00:18:34

2.2.2.2 110 00:10:03

Distance: (default is 110)

Fuente: elaboración propia.

Para ver la tabla de vecinos se emplea el comando *show ip ospf neighbor.*

Figura 114. **Comando *show ip ospf neighbor***

```
R1# show ip ospf neighbor

Neighbor ID Pri State Dead Time Address Interface
2.2.2.2 1 FULL/DR 00:00:32 192.168.2.2 FastEthernet0/1
```

Fuente: elaboración propia.

Para ver la tabla de topología (LSDB) se utiliza el comando *show ip ospf database.*

Figura 115. **Comando *show ip ospf database***

```
R1# show ip ospf database

OSPF Router with ID (1.1.1.1) (Process ID 1)

Router Link States (Area 0)

Link ID ADV Router Age Seq# Checksum Link count
1.1.1.1 1.1.1.1 1265 0x8000000c 0x001eb9 3
2.2.2.2 2.2.2.2 754 0x8000000b 0x006168 3

Net Link States (Area 0)
Link ID ADV Router Age Seq# Checksum
192.168.2.2 2.2.2.2 1269 0x80000002 0x006867
```

Fuente: elaboración propia.

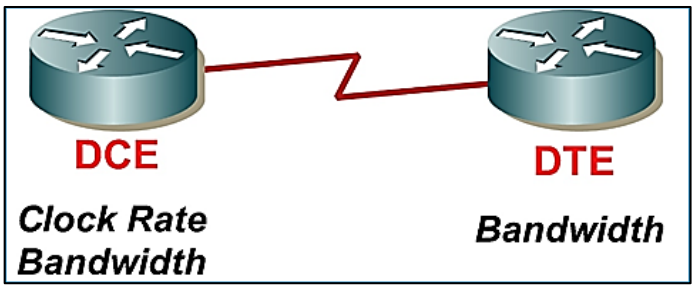
3.11.8. Tipos de red

OSPF reconoce la existencia de distintos tipos de redes y es capaz de ajustar su comportamiento acorde a cada una de ellas. Algunas de estas están definidas en el estándar, mientras que otras son implementaciones propietarias. A continuación se discutirá sobre los casos de las redes punto a punto y múltiple acceso *broadcast.*

3.11.8.1. Red punto a punto

Como su nombre lo indica, esta es una conexión que se realiza exactamente entre dos dispositivos. Es el tipo de red por defecto en las interfaces seriales, mismas que se introducen en este momento y cuya notación se muestra en la figura 116.

Figura 116. **Implementación típica de un enlace serial con referencia a los comandos *clock rate y bandwidth***



Fuente: elaboración propia, empleando *Edraw Max*.

Los enlaces seriales se presentan regularmente en líneas arrendadas a los proveedores de servicio, en donde los últimos toman el rol del *data circuit-terminating equipment*, también referido como *data communication equipment* o simplemente DCE, el cual está encargado de proporcionar la señal de reloj (*clock rate*) de la transmisión, en otras palabras determinar la velocidad de la misma, mientras que los clientes tomarán el rol del *data terminal equipment* o DTE, el cual se limitará a seguir la pauta impuesta por el DCE.

A nivel de la capa de enlace, las interfaces seriales pueden emplear distintos protocolos de encapsulación; ejemplos de ello son el *high-level data link control* (HLDC), protocolo propietario de Cisco y utilizado por defecto en los dispositivos de este fabricante y el estándar abierto *point-to-point protocol* (PPP).

En un laboratorio, los enlaces entre el DCE y el DTE son fácilmente replicados utilizando los cables seriales apropiados, en cuyo caso será necesario establecer la velocidad de la transmisión y la modificación de la

percepción del ancho de banda de los protocolos de enrutamiento utilizando dos comandos:

- *Clock Rate*: este comando impacta físicamente la velocidad de la transmisión de la información, es configurado exclusivamente del lado del DCE y se presenta en el Cisco IOS en bits por segundo (bps).

- *Bandwidth*: este comando no impacta la velocidad de la transmisión, sino que solamente afecta la percepción del ancho de banda de ciertos procesos dentro del *router* (tales como los protocolos de enrutamiento) y es configurado en ciertos enlaces para reflejar la velocidad real de los mismos, ya que por defecto se tomarán los valores presentes (por ejemplo, los enlaces seriales tienen un ancho de banda por defecto de 1544 Kbit). Es importante notar que en el Cisco IOS se presenta en kilobits por segundo (kbps).

El siguiente es un ejemplo de su configuración.

Figura 117. **Ejemplo de configuración**

```
R1(config)#interface serial 0/0
R1(config-if)#ip address 192.168.2.1 255.255.255.0
R1(config-if)#clock rate 64000
R1(config-if)#bandwidth 64

R2(config)#interface serial 0/0
R2(config-if)#ip address 192.168.2.2 255.255.255.0
R2(config-if)#bandwidth 64
```

Fuente: elaboración propia.

Finalmente es posible ver el tipo de red que OSPF utiliza con el comando *show ip ospf interface.*

Figura 118. **Comando *show ip ospf interface 1***

```
R1#show ip ospf interface
Serial0/0 is up, line protocol is up
Internet Address 192.168.2.1/24, Area 0
Process ID 1, Router ID 192.168.2.1, Network Type POINT_TO_POINT, Cost: 1562
Transmit Delay is 1 sec, State UP,
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5 oob-resync timeout 40
```

Fuente: elaboración propia.

3.11.8.2. Red múltiple acceso *broadcast*

Este tipo de red soporta muchos dispositivos conectados a un medio compartido en donde un solo mensaje es capaz de ser replicado a todos los integrantes de la misma (*broadcast),* tal y como ocurre en el caso de Ethernet y presenta ciertas ventajas, tales como el descubrimiento automático de vecinos y ciertos desafíos como el elevado número de adyacencias.

En una red de múltiple acceso, en donde muchos dispositivos comparten una red en común, el número de vecindades entre los mismos, viene por la fórmula donde *n* representa el número de *routers* ejecutando OSPF.

Figura 119. **Fórmula para determinar el número de vecindades en una red de múltiple acceso**

$$\#Vecindades = \frac{n(n-1)}{2}$$

Fuente: elaboración propia.

Al aplicar dicha fórmula es posible darse cuenta que el número de vecindades crece de manera desmesurada a medida que se agregan más dispositivos al mismo segmento. Por ejemplo, en el caso de existir 10 *routers* se tendrían 45 vecindades.

Si todas las posibles vecindades se convierten en adyacencias, sería problemático para el correcto funcionamiento de la red, como se explica a continuación.

Figura 120. **Cálculo del número de vecindades que formarían 10 dispositivos en un mismo segmento de red**

$$\#Vecindades = \frac{10(10-1)}{2} = 45$$

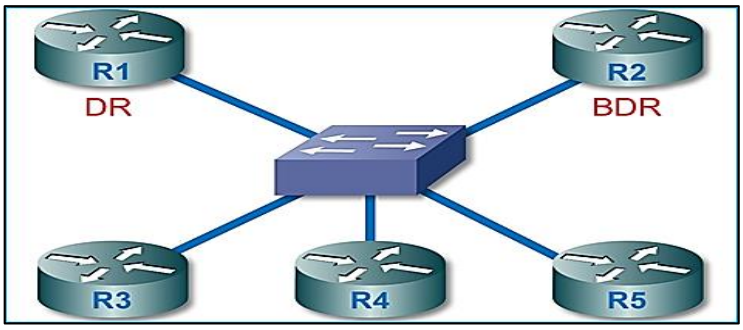
Fuente: elaboración propia.

Una gran cantidad de adyacencias generarían una gran cantidad de actualizaciones que consumirían parte considerable del ancho de banda del medio, además de provocar una gran cantidad de ejecuciones del algoritmo SPF dentro de los dispositivos, razones por las cuales en redes de múltiple acceso se implementan los roles del *router* designado o *designated router* (DR) y del *router* designado de respaldo o *backup designated router* (BDR).

El DR y el BDR son *routers* elegidos dentro de cada dominio de *broadcast* para reducir el número de adyacencias en una red de múltiple acceso. Se comunican a través de la dirección de *multicast* 224.0.0.6 y serán los únicos dispositivos con los que todos los demás formarán una adyacencia reduciendo

efectivamente el número de las mismas. El DR será el encargado de recibir y distribuir actualizaciones hacia todos los demás dispositivos que se encuentren dentro del mismo segmento; mientras que el BDR existe solamente con propósitos de respaldo y pasará a tomar el rol del DR, solo en caso de que este falle.

Figura 121. **Red de múltiple acceso *broadcast* donde R1 ha tomado el rol de DR y R2 el de BDR**



Fuente: elaboración propia, empleando *Edraw Max*.

Si se considera nuevamente la existencia de 10 dispositivos en un solo segmento, se tendrán entonces, que de las 45 vecindades solo 17 llegarán a ser adyacencias completas (la adyacencia entre el DR y BDR y la de todos los demás dispositivos con estos dos).

La elección del DR y BDR se realiza de manera automática dentro de una red, aunque es posible influir la elección de los mismos.

El primer criterio utilizado para dicha elección es un valor existente en las interfaces que conectan a redes de múltiple acceso (ej.: *ethernet*) referido como la prioridad, mientras más alto es este parámetro mayor es la probabilidad del

router de convertirse en el DR. Si todos los dispositivos tienen la misma prioridad (lo que pasará en las implementaciones en donde no se cambien los valores por defecto) se utiliza el identificador de cada *router* (*Router-ID*) como siguiente criterio, siendo preferido el valor más alto.

Para elegir los dispositivos que pasarán a tomar los roles anteriormente mencionados, deberá modificarse manualmente el valor de la prioridad en cada una de las interfaces conectadas al medio compartido.

La prioridad puede tomar valores entre 0 y 255, donde 1 es el valor utilizado por defecto por Cisco y 0 es un valor especial que indica al *router* que este nunca podrá convertirse en el DR o BDR. Se prefieren siempre los valores más altos siendo el mayor de todos el DR y el segundo mayor el BDR, en caso de empate se utilizará aquel dispositivo que posea el *router-ID* más alto, como se ha explicado anteriormente.

La prioridad en una interface puede modificarse con el comando *ip ospf priority,* siendo posible alterarla para que dos dispositivos tomen los roles de DR y BDR, como se muestra en la figura 122.

Figura 122. **Comando *ip ospf priority***

```
R1(config)# interface fastethernet 0/0
R1(config-if)# ip ospf priority ?
  <0-255>  Priority
R1(config-if)# ip ospf priority 255

R2(config)# interface fastethernet 0/0
R2(config-if)# ip ospf priority 254
```

Fuente: elaboración propia.

Un factor importante a tomar en cuenta es que la modificación del valor de la prioridad no provoca una renegociación automática de los roles de los dispositivos; en otras palabras, los *routers* electos como DR y BDR durante el proceso de convergencia se mantendrán hasta que se reinicien los mismos o hasta que se reinicie el proceso OSPF en los *routers* necesarios.

Para apreciar la relación entre los dispositivos, se muestran los vecinos de un *router* sin ningún rol en especial. Nótese las adyacencias (identificadas por el estatus FULL) con el DR y BDR y las vecindades (que se quedan en la etapa de 2WAY) con todos los demás dispositivos sin un rol específico y que se identifican como DROTHER.

Figura 123. **Tabla de vecinos**

```
R5# show ip ospf neighbor

Neighbor ID  Pri   State           Dead Time   Address         Interface
1.1.1.1      255   FULL/DR         00:00:34    192.168.1.1     FastEthernet0/0
2.2.2.2      254   FULL/BDR        00:00:33    192.168.1.2     FastEthernet0/0
3.3.3.3        1   2WAY/DROTHER    00:00:37    192.168.1.3     FastEthernet0/0
4.4.4.4        1   2WAY/DROTHER    00:00:34    192.168.1.4     FastEthernet0/0
```

Fuente: elaboración propia.

Para visualizar la prioridad, el rol, el tipo de red, así como otros datos importantes, puede emplearse nuevamente el comando *show ip ospf interface.*

Figura 124. **Comando *show ip ospf interface 2***

```
R5# show ip ospf interface
FastEthernet0/0 is up, line protocol is up
  Internet Address 192.168.1.5/24, Area 0
  Process ID 1, Router ID 5.5.5.5, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DROTHER, Priority 1
  Designated Router (ID) 1.1.1.1, Interface address 192.168.1.1
  Backup Designated router (ID) 2.2.2.2, Interface address 192.168.1.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
        oob-resync timeout 40
        Hello due in 00:00:03
  Supports Link-local Signaling (LLS)
  Index 1/1, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 4, Adjacent neighbor count is 2
    Adjacent with neighbor 1.1.1.1  (Designated Router)
    Adjacent with neighbor 2.2.2.2  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

Fuente: elaboración propia.

Cabe mencionar que la elección del DR y BDR en una red *ethernet* es regularmente dejada al azar, mientras que en otros tipos de redes (ej.: *frame-relay*), esta elección es de crucial importancia, por lo que debe realizarse manualmente.

3.11.9. Sumarización de rutas

Es una técnica que permite publicar varias subredes como parte de una subred más grande para ahorrar ancho de banda, reducir actualizaciones y acortar la tabla de enrutamiento de otros dispositivos.

En el caso de RIP y EIGRP puede realizarse en cualquier punto de la red; mientras que en OSPF, su uso está reservado a los ABR y ASBR.

La sumarización sigue los siguientes lineamientos generales, cuando se utiliza con los protocolos anteriormente mencionados.

- Una ruta sumarizada será publicada siempre y cuando exista, al menos, una subred dentro de su rango de direcciones.

- Una ruta sumarizada utilizará como métrica alguna de las encontradas en las subredes contenidas dentro de su rango de direcciones. EIGRP utilizará la métrica más pequeña encontrada; en tanto que, en OSPF la elección dependerá de la revisión del estándar donde está definido. Las implementaciones compatibles con el rfc1583 usan la métrica más pequeña, mientras que las compatibles con el rfc2328 (más moderno) usarán la métrica más grande; dentro del Cisco IOS existe un comando que permite elegir entre las mismas (*compatible rfc1583*). RIP es una excepción.

- El dispositivo donde se realice la sumarización generará automáticamente una ruta estática hacia una interfaz especial llamada *null0*, cuya función es desechar los paquetes que lleguen a ella y cuyo propósito en este caso es evitar posibles bucles de enrutamiento. La

excepción a esta regla es RIP, en donde se recomienda que dicha ruta se agregue manualmente.

Se menciona nuevamente que, tanto RIP como EIGRP poseen un mecanismo de sumarización automática (autosumarización), cuando se encuentran en redes discontinuas. Se recomienda desactivar dicha función para realizar una sumarización manual en caso que esta sea necesaria.

Para proceder a la sumarización manual se recomienda utilizar el mismo razonamiento presentado en la sección de *subnetting,* en donde se visualiza el rango de direcciones consumido por cada subred en términos de “incrementos” entre las mismas, donde dicho incremento está dado por la máscara de subred.

A manera de ejemplo se presentan las siguientes ocho subredes para realizar una sumarización; con lo que se pretende anunciar las mismas como parte de una subred más grande y reducir el tamaño de la tabla de enrutamiento de los demás dispositivos, al reducir ocho posibles rutas a solo una.

Tabla XVII. **Rutas a sumarizar**

10.0.0.1/24	10.0.1.1/24	10.0.2.1/24	10.0.3.1/24
10.0.4.1/24	10.0.5.1/24	10.0.6.1/24	10.0.7.1/24

Fuente: elaboración propia.

Para apreciar de una mejor manera el rango de direcciones puede utilizarse nuevamente la siguiente disposición.

Tabla XVIII. **Detalle de las redes a sumarizar**

#	Red	Primera Dirección Utilizable	Última Dirección Utilizable	Broadcast
1	10.0.0.0	10.0.0.1	10.0.0.254	10.0.0.255
2	10.0.1.0	10.0.1.1	10.0.1.254	10.0.1.255
3	10.0.2.0	10.0.2.1	10.0.2.254	10.0.2.255
	...	...	...	...
7	10.0.6.0	10.0.6.1	10.0.6.254	10.0.6.255
8	10.0.7.0	10.0.7.1	10.0.7.254	10.0.7.255

Fuente: elaboración propia.

Es evidente que la subred que se pretende crear debe abarcar desde la dirección 10.0.0.0 hasta la 10.0.7.255 y que la dirección de red de la siguiente subred será 10.0.8.0.

Al hacer un análisis es posible notar que el incremento entre las mismas es de 8 y que tiene lugar en el tercer octeto.

Tabla XIX. **El incremento es de 8 y se encuentra en el tercer octeto**

#	...	...	...	...
	10.0.0.0			10.0.7.255
	10.0.8.0			

Fuente: elaboración propia.

Lo único que resta es encontrar el valor que provocará dicho incremento e incluirlo en la máscara de subred en el octeto necesario, en este caso el tercero (se agregan bits con un valor de uno) de la máscara de subred hasta lograr el incremento necesario en el octeto adecuado.

Tabla XX. **Asignación de bits en la parte de red**

128	64	32	16	8	4	2	1
1	1	1	1	1	0	0	0

Fuente: elaboración propia.

De esta manera se obtiene una nueva máscara que define o que cubre el rango de direcciones necesarias para cumplir con el propósito original siendo esta 255.255.248.0 o /21.

Nótese que para los valores correspondientes de la máscara de subred de los dos primeros octetos en donde no hay cambio (10.0.x.x), se han colocado los valores de 255 y que en el tercer octeto en donde se aprecia el incremento entre ambas subredes, se han agregado bits a la parte de red de la máscara hasta lograr el incremento deseado, dejando los demás bits a cero.

Así que es posible anunciar las ocho subredes deseadas como una sola ruta: 10.0.0.0/21 o 10.0.0.0 255.255.248.0

Para implementar la sumarización en el caso de RIP y EIGRP basta con entrar a la interfaz deseada, en cualquier punto de la red, a través de la cual se publicará la ruta sumarizada utilizando el comando *ip summary-address*, como se muestra la figura 125.

Figura 125. **Comando *ip summary-address* RIP**

```
R1(config)# inter fastethernet 0/0
R1(config-if)# ip summary-address rip 10.0.0.0 255.255.248.0
```

Fuente: elaboración propia.

En EIGRP es necesario agregar el número de sistema autónomo (definido más adelante). En este caso se supondrá que el número de dicho sistema es el "1".

Figura 126. **Comando *ip summary-address eigrp***

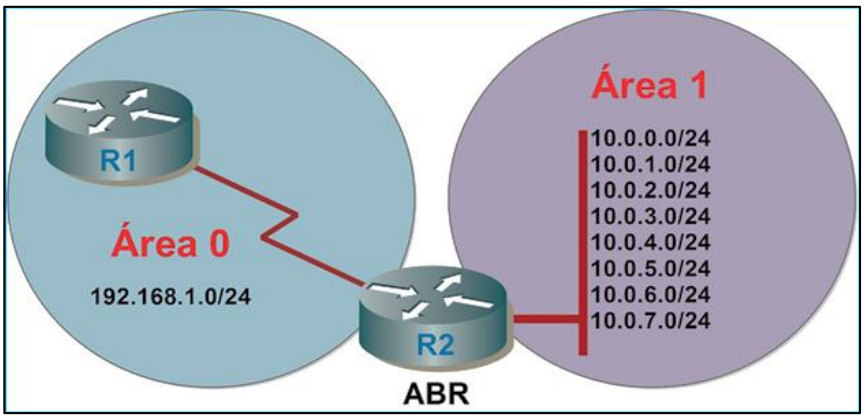
```
R1(config)# inter fastethernet 0/0
R1(config-if)# ip summary-address eigrp 1 10.0.0.0 255.255.248.0
```

Fuente: elaboración propia.

Por otro lado, la sumarización en OSPF está limitada a aquellos *routers* que delimitan o que sirven de frontera entre otras partes de la red, los ABR y los ASBR.

Para ejemplificar su implementación se presenta la siguiente topología, donde R2 está conectado a través de su interfaz serial a R1 en el área 0 utilizando la red 192.168.1.0/24 y tiene conocimiento de las mismas ocho subredes utilizadas anteriormente para demostrar la sumarización y que son utilizadas por conveniencia, siendo asignadas al área 1.

Figura 127. **Ejemplo sumarización de rutas en OSPF**



Fuente: elaboración propia, empleando *Edraw Max*.

El ejercicio puede reproducirse fácilmente en el laboratorio, creando interfaces de *loopback* dentro de un dispositivo y cambiando manualmente el tipo de red utilizada por OSPF a punto a punto, lo que se hace con el propósito de emular un escenario real, ya que de otra manera dichas interfaces se anunciarían con una máscara /32.

A manera de ejemplo se muestra en la figura 128 la creación y publicación de la primera interfaz.

Figura 128. **Creación y publicación de la primera interfaz**

```
R2(config)# interface loopback 0
R2(config-if)# ip address 10.0.0.1 255.255.255.0
R2(config-if)# ip ospf network point-to-point
R2(config)# router ospf 1
R2(config-router)# network 10.0.0.1 0.0.0.0 area 1
```

Fuente: elaboración propia.

Una vez terminadas las preparaciones necesarias, es posible observar en la tabla de enrutamiento de R1, las ocho subredes anunciadas marcadas con el código O IA (OSPF interárea) que es el que identifica a las rutas aprendidas por OSPF, que se encuentran en otras áreas.

Figura 129. **Rutas interárea**

```
R1# show ip route
Codes: 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

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 8 subnets
O IA    10.0.2.0 [110/2] via 192.168.1.2, 00:00:15, Serial0/0
O IA    10.0.3.0 [110/2] via 192.168.1.2, 00:00:15, Serial0/0
O IA    10.0.0.0 [110/2] via 192.168.1.2, 00:00:11, Serial0/0
O IA    10.0.1.0 [110/2] via 192.168.1.2, 00:00:15, Serial0/0
O IA    10.0.6.0 [110/2] via 192.168.1.2, 00:00:15, Serial0/0
O IA    10.0.7.0 [110/2] via 192.168.1.2, 00:00:15, Serial0/0
O IA    10.0.4.0 [110/2] via 192.168.1.2, 00:00:15, Serial0/0
O IA    10.0.5.0 [110/2] via 192.168.1.2, 00:00:17, Serial0/0
     192.168.1.0/30 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, Serial0/0
```

Fuente: elaboración propia.

Para sumarizar las redes indicadas es necesario realizarlo en R2 (ABR) utilizando el comando *area* (número de área), *range* (dirección, máscara) de acuerdo como lo muestra en la figura 131.

Figura 130. **Comando *area-range***

```
R2(config)# router ospf 1
R2(config-router)# area 1 range 10.0.0.0 255.255.248.0
```

Fuente: elaboración propia.

Al examinar nuevamente la tabla de enrutamiento de R1 se tiene como resultado lo descrito en la figura 130.

Figura 131. **Ruta sumarizada**

```
R1# show ip route
Codes: 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

Gateway of last resort is not set

      10.0.0.0/21 is subnetted, 1 subnets
O IA  10.0.0.0 [110/2] via 192.168.1.2, 00:02:02, Serial0/0
      192.168.1.0/30 is subnetted, 1 subnets
C     192.168.1.0 is directly connected, Serial0/0
```

Fuente: elaboración propia.

Puede apreciarse también la creación de una nueva ruta dirigida a *null0* en R2, para evitar posibles bucles de enrutamiento, al ser únicamente utilizada cuando no se encuentra una entrada más específica en la tabla de enrutamiento.

Figura 132. **Nueva ruta dirigida a *null0* en R2**

```
R2# show ip route
Codes: 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
 Gateway of last resort is not set
       10.0.0.0/8 is variably subnetted, 9 subnets, 2 masks
C      10.0.2.0/24 is directly connected, Loopback2
C      10.0.3.0/24 is directly connected, Loopback3
C      10.0.0.0/24 is directly connected, Loopback0
O      10.0.0.0/21 is a summary, 00:00:56, Null0
C      10.0.1.0/24 is directly connected, Loopback1
C      10.0.6.0/24 is directly connected, Loopback6
C      10.0.7.0/24 is directly connected, Loopback7
C      10.0.4.0/24 is directly connected, Loopback4
C      10.0.5.0/24 is directly connected, Loopback5
       192.168.1.0/30 is subnetted, 1 subnets
C      192.168.1.0 is directly connected, Serial0/0
```

Fuente: elaboración propia.

Como se puede apreciar en el ejemplo anterior, para aplicar esta técnica sobre varias subredes es necesario que exista cierto grado de continuidad entre

ellas (ej.: no se puede anunciar las redes 192.168.1.0 y 10.1.1.0 como una sola ruta), por lo que debe seguirse un “diseño jerárquico”, lo que hace referencia a la uniformidad en la asignación de direcciones.

Finalmente, es importante notar que la sumarización será posible siempre y cuando el número de subredes a sumarizar sea una potencia de dos. Por ejemplo, si se quisiera aplicar esta técnica sobre 9 subredes, se tendrían que sumarizar 8 y anunciar la última de manera independiente, por lo que se tendrían 2 rutas en las tablas de enrutamiento de los demás dispositivos.

3.11.10. Consideraciones finales

Para concluir la discusión de OSPF se presentan dos consideraciones a tomar en cuenta a la hora de implementar este protocolo.

3.11.10.1. Reconfiguración del ancho de banda de referencia

Como se ha explicado, la métrica de OSPF utiliza una fórmula sencilla que utiliza un valor de referencia de 100 Mbps para calcular el costo entre varias rutas. Dicho valor, funcional en los orígenes del protocolo, es insuficiente en la actualidad para calcular correctamente el costo de los enlaces con una velocidad superior a fastethernet, debido a las limitaciones del cálculo a valores enteros positivos.

Para resolver este problema es posible ajustar el valor de referencia como se presenta a continuación.

Figura 133. **Ajuste del valor de referencia**

```
R1(config)# router ospf 1
R1(config-router)# auto-cost reference-bandwidth ?
  <1-4294967>  The reference bandwidth in terms of Mbits per second
```

Fuente: elaboración propia.

3.11.10.2. Publicación de una ruta por defecto

En una red con una gran cantidad de dispositivos es conveniente poseer la capacidad de inyectar o redistribuir una ruta por defecto (por ejemplo, la ruta que conduce al ISP), en las publicaciones de un protocolo.

Para este propósito OSPF incluye el comando *default-information originate.*

Figura 134. **Comando *default-information originate***

```
R1(config)#ip route 0.0.0.0 0.0.0.0 [Dirección IP del ISP]
R1(config-router)#router ospf 1
R1(config-router)#default-information originate
```

Fuente: elaboración propia.

3.12. *Enhanced Interior Gateway Routing Protocol* (EIGRP)

Es un protocolo de enrutamiento creado por Cisco en 1993, año en que fue introducido como reemplazo de su predecesor IGRP el cual poseía un comportamiento *classful.* Conocido durante muchos años por ser un protocolo

propietario, se convierte en un estándar abierto en el 2013. A la presente fecha no ha sido implementado por ningún otro fabricante.

EIGRP es un protocolo vector, distancia que ha incorporado algunas características presentes hasta el momento de su creación solamente en los protocolos de estado de enlace, tal como el descubrimiento automático de vecinos, para mejorar su funcionamiento.

Se identifica con el número de protocolo 88 y al igual que en OSPF, implementa sus propios mecanismos para lograr transmisiones confiables. Presenta un comportamiento *classless,* soporta autenticación y envía sus actualizaciones a la dirección de *multicast* 224.0.0.10.

A diferencia de OSPF, no cuenta con una visión completa de toda la topología lo que implica una menor utilización de los recursos de los dispositivos a costa de lidiar nuevamente con el problema de los bucles de enrutamiento por lo que está sujeto a la regla del horizonte dividido (*Split Horizon*) e implementa nuevos mecanismos de prevención por lo que EIGRP no garantiza la utilización de la mejor ruta en el 100 % de los casos. Además, al igual que RIP, presenta el problema de la auto sumarización en la frontera discontinua, aunque en versiones modernas del Cisco IOS se presenta desactivada por defecto.

Utiliza un algoritmo conocido como DUAL (*diffusing update algorithm*), gracias al cual EIGRP es el protocolo de enrutamiento de más rápida convergencia, descubriendo automáticamente dispositivos vecinos con los cuales intercambia actualizaciones parciales que se envían solamente cuando se presenta un cambio en la topología.

Una característica única de este protocolo es la utilización de rutas de respaldo; a diferencia de los demás protocolos de enrutamiento que deben recalcular la mejor ruta cada vez que hay un cambio en la topología EIGRP, mantiene en memoria un listado de las rutas alternativas (libres de bucles) para que, alguna o varias de ellas, puedan ser instaladas en la tabla de enrutamiento en el caso que la mejor ruta deje de estar disponible.

EIGRP posee una métrica muy compleja que puede llegar a considerar el ancho de banda, el retraso introducido por las interfaces, la carga (promedio del tráfico) y la confiabilidad (errores), cada uno de ellos escalado por una serie de parámetros conocidos como los valores K, de los cuales todos están puestos a cero con excepción de aquellos correspondientes al ancho de banda y al retraso, por lo que al utilizar los valores por defecto el cálculo de la métrica se reduce a la fórmula que se detalla en la figura 135.

Figura 135. **Fórmula para calcular la métrica de EIGRP**

$$\text{Métrica} = \frac{10^7 * 256}{\text{Ancho de Banda (Interface más Lenta)}} + \sum \text{Retrasos (decimas microsegundo)}*256$$

Fuente: elaboración propia.

3.12.1. Tablas mantenidas por EIGRP

EIGRP mantiene tres tablas para almacenar información.

- Tabla de enrutamiento: donde se almacenan las mejores rutas se sigue el mismo comportamiento y se respetan los criterios que se han presentado.

- Tabla de vecinos (*neighbor table*): donde se almacena la información de todos los dispositivos que comparten una red común (a través de alguna de sus interfaces), con el dispositivo y que han cumplido con los requisitos para establecer una vecindad.

- Tabla de topología (*topology table*): donde se almacenan todas las rutas aprendidas de los vecinos, las mejores de ellas serán copiadas a la tabla de enrutamiento, mientras que las rutas alternativas serán guardadas en caso de que las primeras dejen de estar disponibles para ser utilizadas en su lugar.

3.12.2. Rutas de respaldo

La tabla de topología distingue dos tipos de rutas: las mejores son llamadas sucesores (*succesors*), mientras que las rutas de respaldo son llamadas sucesores factibles (*feasible succesors*).

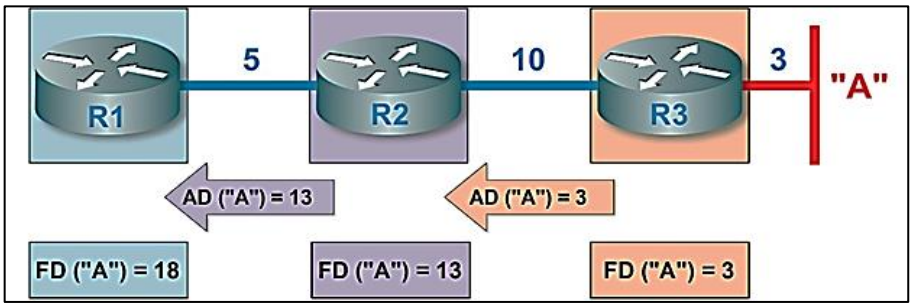
Los sucesores son enviados a la tabla de enrutamiento, mientras que los sucesores factibles se mantienen en la tabla de topología para ser utilizados inmediatamente, en caso que los primeros dejen de ser utilizables. En otras palabras, de estar disponible una ruta de respaldo, esta se instalará inmediatamente en la tabla de enrutamiento si la ruta principal ya no puede ser empleada.

Para realizar la clasificación de las rutas EIGRP se utiliza la misma métrica desde dos puntos de vista diferentes.

La métrica hacia un destino en particular es transmitida por un dispositivo hacia sus vecinos y es referida como distancia notificada (*reported distance* -

RD-) o distancia publicada (*advertised distance* -AD-), la cual es luego utilizada por estos para calcular su propia distancia hacia dicho destino y que recibe el nombre de distancia factible (*feasible distance* -FD-). Para ilustrar dicho concepto se presenta a continuación una topología donde la métrica de cada segmento será 5, 10 y 3 respectivamente.

Figura 136. **La distancia publicada (AD) se transmite a los vecinos para que estos puedan calcular la distancia factible (FD) hacia un destino**



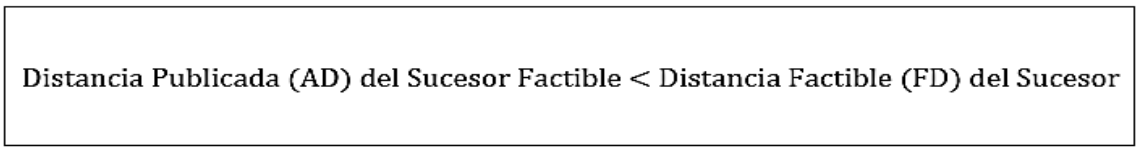
Fuente: elaboración propia, empleando *Edraw Max*.

En el ejemplo, R3 registra una distancia hacia la red “A” con un valor de 3, la misma es anunciada a su vecino R2 quien conoce que la métrica hacia R3 tiene un valor de 10, por lo que “A” se encuentra a una distancia factible con un valor de 13 y así sucesivamente.

3.12.2.1. Condición de factibilidad

Para que una ruta pueda convertirse en un sucesor factible debe cumplir con la condición de factibilidad expresada a continuación:

Figura 137. **Condición de factibilidad**



Fuente: elaboración propia.

En otras palabras, para que un dispositivo vecino pueda ser utilizado como ruta de respaldo debe encontrarse necesariamente más cerca del destino, método que no garantiza la elección de la mejor ruta, pero que evita la producción de bucles de enrutamiento.

3.12.3. Tipos de paquetes

EIGRP utiliza cinco diferentes tipos de paquetes:

- *Hello:* se utiliza para el descubrimiento, formación y mantenimiento de vecindades con otros dispositivos.

- *Update:* contienen información acerca de las rutas. Sincroniza las tablas de topología.

- *Query:* a falta de sucesores factibles, este paquete es utilizado para consultar a los dispositivos vecinos por rutas alternativas.

- *Reply:* es una respuesta a un paquete *Query.*

- ACK: es un acuse de recibo para los paquetes *update, query* y *reply.* Los paquetes *hello* no requieren de acuse de recibo.

3.12.4. Sistemas autónomos

Dentro del internet, un sistema autónomo es un conjunto de equipos o de direcciones IP que se encuentran bajo el control de una organización.

Cada uno de estos se identifica con un número de sistema autónomo (*autonomous system number* -ASN-) son regulados por la internet *corporation for assigned names and numbers* (ICANN), una organización sin fines de lucro. Los ASN eran hasta el año 2007, valores de 16 bits. Ahora son de 32 bits. Esta expansión se debió a la demanda de los mismos.

El protocolo para intercambiar información entre sistemas autónomos es el *border gateway protocol* (BGP), un protocolo de *gateway* exterior, el cual es diferente a los protocolos de enrutamiento explicados anteriormente, considerados de *gateway* interior.

En el caso de EIGRP, un sistema autónomo estará compuesto por un grupo de *routers* entre los que es necesario intercambiar rutas.

3.12.5. Requerimientos y vecindades

El único requerimiento estrictamente necesario para configurar EIGRP es asignar un número de sistema autónomo al proceso.

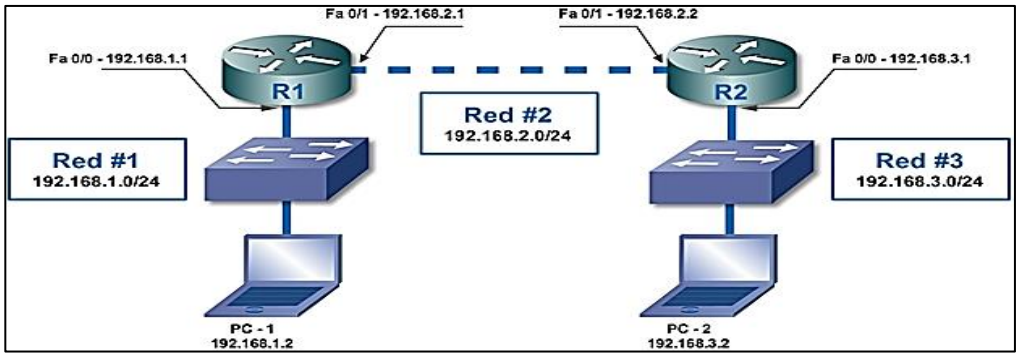
Las vecindades son establecidas a través de los paquetes *hello*, estos se envían a través de todas las interfaces configuradas para formar parte del

proceso EIGRP en un intervalo regular de tiempo. Al contrario de OSPF, EIGRP no exige muchos requisitos para que dos *routers* puedan intercambiar rutas, solamente es necesario que se encuentren dentro del mismo sistema autónomo (que posean el mismo ASN) y que utilicen los mismos valores K.

3.12.6. Configuración

Para demostrar la implementación de EIGRP se recurre a la topología utilizada en las secciones anteriores y que se observa en la figura 138.

Figura 138. **Topología base para los ejemplos de las secciones de enrutamiento**



Fuente: elaboración propia, empleando *Edraw Max*.

Para este ejemplo se utilizará el número de sistema autónomo 1, los conceptos de autosumarización e interfaces pasivas han sido explicados en secciones anteriores.

Figura 139. **Sistema autónomo**

```
R1(config)# router eigrp ?
  <1-65535>  Autonomous system number
R1(config)# router eigrp 1
R1(config-router)# no auto-summary
R1(config-router)# passive-interface default
R1(config-router)# network 192.168.1.0
R1(config-router)# network 192.168.2.0
R1(config-router)# no passive-interface fastethernet 0/1
```

Fuente: elaboración propia.

Una particularidad de EIGRP es que puede ser configurado de varias maneras, ya sea con la sencillez de RIP, como en el caso anterior, o con la flexibilidad de las *Wildcard Masks,* como se muestra en la figura 140.

Figura 140. **Configuración *Wildcard Masks***

```
R2(config)# router eigrp 1
R2(config-router)# no auto-summary
R2(config-router)# passive-interface default
R2(config-router)# network 192.168.2.2 0.0.0.0
R2(config-router)# network 192.168.3.1 0.0.0.0
R2(config-router)# no passive-interface fastethernet 0/1
```

Fuente: elaboración propia.

Adviértase que no debe confundirse el número de sistema autónomo utilizado en EIGRP con el identificador de proceso usado por OSPF.

Una vez terminada la configuración de R2 es posible establecer comunicación entre los ordenadores.

Para mostrar detalles importantes acerca de la configuración se utiliza nuevamente el comando *show ip protocols.*

Figura 141. **Comando *show ip protocols***

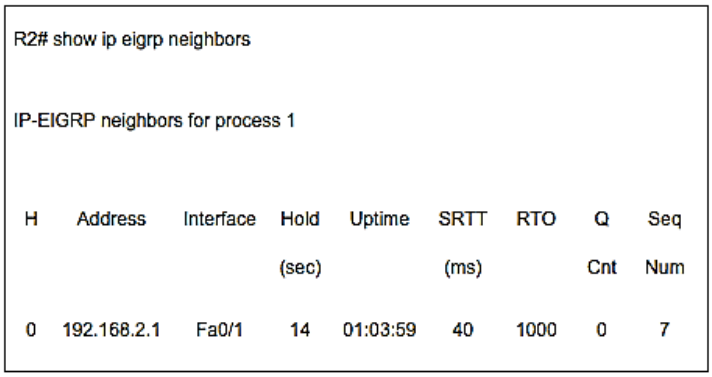
```
R2# show ip protocols

Routing Protocol is "eigrp  1 "
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
Redistributing: eigrp 1
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
        192.168.2.2/32
        192.168.3.1/32
  Routing Information Sources:
        Gateway         Distance      Last Update
        192.168.2.1     90            90401818
  Distance: internal 90 external 170
```

Fuente: elaboración propia.

Para ver la tabla de vecinos se utiliza el comando *show ip eigrp neighbors.*

Finalmente, para ver la topología se emplea la instrucción *show ip eigrp topology.*

Figura 142. **Comando *show ip eigrp neighbors***



R2# show ip eigrp neighbors

IP-EIGRP neighbors for process 1

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
0	192.168.2.1	Fa0/1	14	01:03:59	40	1000	0	7

Fuente: elaboración propia.

Figura 143. **Instrucción *show ip eigrp topology***

```
R2# show ip eigrp topology

IP-EIGRP Topology Table for AS 1/ID(192.168.3.1)

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
r - Reply status

P 192.168.1.0/24, 1 successors, FD is 30720
via 192.168.2.1 (30720/28160), FastEthernet0/1
P 192.168.2.0/24, 1 successors, FD is 28160
via Connected, FastEthernet0/1
P 192.168.3.0/24, 1 successors, FD is 28160
via Connected, FastEthernet0/0
```

Fuente: elaboración propia.

3.13. *Virtual* LANs (VLANs), enlaces troncales y *dynamic trunking protocol* (DTP)

Para comenzar esta sección se explica el funcionamiento de VLAN.

3.13.1. VLAN

Al crecer el número de dispositivos dentro de las primeras redes, algunos problemas se hacen evidentes.

- Más dispositivos implican una mayor cantidad de *switches* y una mayor transmisión de *broadcasts,* esto impacta negativamente el rendimiento y escalabilidad de la red.
- Grupos con diferentes funciones dentro de una organización se encuentran limitados físicamente debido a su conexión con los dispositivos (ej.: todas las secretarías debían estar conectadas al mismo *switch* para acceder a los servicios que necesitan). Lo anterior limita la movilidad y condiciona la ubicación dentro de un inmueble.
- Para conectar grupos asociados a redes diferentes se necesita introducir algún dispositivo capaz de enrutar entre las mismas o añadir interfaces a equipos existentes, lo que incrementa los costos.
- Pobre control de acceso y poca capacidad para aplicar calidad de servicio.

Para solucionar o paliar estos problemas se introduce el concepto de las redes locales virtuales, mejor conocidas como *Virtual* LANs (VLANs).

Las VLANs fueron creadas en 1980 por Walter David "Dave" Sincoskie. Permiten separar los puertos de cada *switch* y asignarlos a grupos lógicos

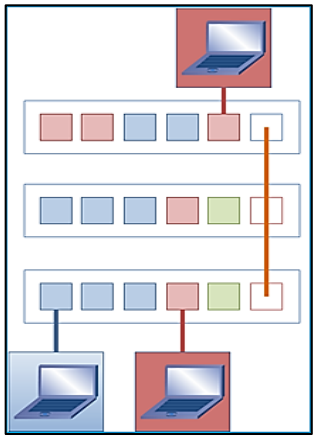
distintos, donde cada uno de ellos constituye su propio dominio de *broadcast.* Con ello se posibilita la utilización de varias redes independientes dentro del mismo concentrador.

Dicha tecnología permite agregar usuarios a una agrupación lógica accesible desde cualquier *switch* de acceso en la infraestructura, lo que elimina las limitaciones físicas, reduce el *broadcast* y mejora el control de acceso.

La asignación a estos grupos se realiza de manera individual dentro de la configuración de cada puerto. Es imposible que dispositivos finales conozcan la VLAN a la que están conectados al estar todos asignados a la VLAN 1 por defecto.

Para mostrar el funcionamiento de esta tecnología se presenta el siguiente ejemplo (figura 144), en donde varias VLANs han sido configuradas y pueden ser consideradas, para propósitos prácticos, redes completamente separadas (ej.: los puertos rojos solo podrán alcanzar los puertos del mismo color).

Figura 144. **Implementación de VLANs**



Fuente: elaboración propia, empleando *Edraw Max*.

Adviértase también, la existencia de un puerto con un funcionamiento especial, encargado de transmitir información de todas las VLANs hacia los demás *switches,* llamado puerto troncal (*trunk)* por Cisco y puerto etiquetado (*tagged port*) por los demás fabricantes. Este facilita, en gran medida, la escalabilidad de la red (de lo contrario se necesitaría un enlace entre *switches* por cada uno de los grupos lógicos).

Para crear una VLAN y asignarle un nombre es posible utilizar la siguiente secuencia de instrucciones desde el modo de configuración global (ver figura 145).

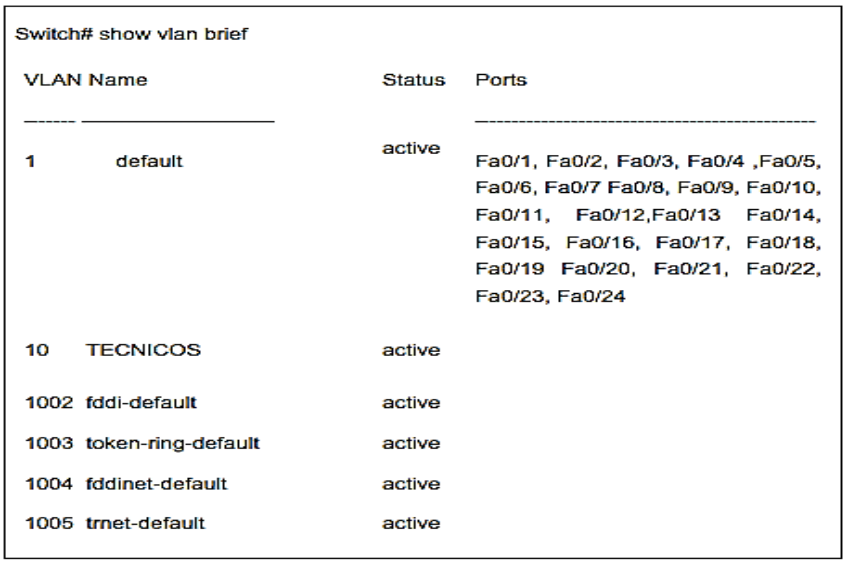
Figura 145. **Secuencia de instrucciones**

```
Switch(config)#vlan 10
Switch(config-vlan)#name TECNICOS
```

Fuente: elaboración propia.

Para visualizar las VLANs existentes, así como los puertos asignados a ellas se utiliza el comando *show vlan brief* (ver figura 146).

Figura 146. **Comando *show vlan brief* 1**



```
Switch# show vlan brief

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/1, Fa0/2, Fa0/3, Fa0/4 ,Fa0/5,
                                                 Fa0/6, Fa0/7 Fa0/8, Fa0/9, Fa0/10,
                                                 Fa0/11,  Fa0/12,Fa0/13  Fa0/14,
                                                 Fa0/15, Fa0/16, Fa0/17, Fa0/18,
                                                 Fa0/19 Fa0/20, Fa0/21, Fa0/22,
                                                 Fa0/23, Fa0/24
10   TECNICOS                          active
1002 fddi-default                      active
1003 token-ring-default                active
1004 fddinet-default                   active
1005 trnet-default                     active
```

Fuente: elaboración propia.

La VLAN 1 es la VLAN por defecto a donde están asignados todos los puertos; mientras que las VLANs 1002-1005 son mantenidas por cuestiones de compatibilidad, por lo que ninguna de ellas puede ser removida.

3.13.2. Modos de un puerto

Los puertos de un *switch* pueden trabajar en uno de los siguientes modos.

- Modo troncal (*mode trunk*): configura el puerto para que etiquete, envíe y posibilite la comunicación entre VLANs en *switches* diferentes.

Figura 147. **Modo troncal**

```
Switch(config)# interface fastEthernet 0/1
Switch(config-if)# switchport mode trunk
```

Fuente: elaboración propia.

Los enlaces troncales son examinados con detalle más adelante.

- Modo de acceso (*mode access*): configura el puerto para no utilizar ninguna marcación y funcionar en una VLAN específica, es la VLAN 1 por defecto.

Figura 148. **Modo de acceso**

```
Switch(config)# interface fastEthernet 0/2
Switch(config-if)# switchport mode access
```

Fuente: elaboración propia.

En el modo de acceso es posible asignar dicho puerto a una VLAN específica, de la forma como se muestra en la figura 149.

Figura 149. **Asignación a VLAN**

```
Switch(config-if)# switchport access vlan 10
```

Fuente: elaboración propia.

Es posible verificar la asignación utilizando nuevamente el comando *show vlan brief* o a través del comando *show vlan name*:

Figura 150. **Comando *show vlan brief 2***

```
Switch# show vlan brief
 VLAN Name                              Status    Ports
 ---- --------------------------------- --------- -------------------------------
 1    default                           active    --
                                                  Fa0/3, Fa0/4 ,Fa0/5, Fa0/6,
                                                  Fa0/7 Fa0/8, Fa0/9, Fa0/10,
                                                  Fa0/11, Fa0/12,Fa0/13 Fa0/14,
                                                  Fa0/15, Fa0/16, Fa0/17,
                                                  Fa0/18, Fa0/19 Fa0/20,
                                                  Fa0/21, Fa0/22, Fa0/23,
                                                  Fa0/24
 10   TECNICOS                          active    Fa0/2

 1002 fddi-default                      active
 1003 token-ring-default                active
 1004 fddinet-default                   active
 1005 trnet-default                     active
```

Fuente: elaboración propia.

Figura 151. **Comando *show vlan name***

```
Switch# show vlan name TÉCNICOS

 VLAN Name                              Status    Ports
 ---- --------------------------------- --------- -------------------------------
 10   TÉCNICOS                          active    Fa0/2

 VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1
 ---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
 10   enet  100010     1500  -      -      -        -        -        0      0
```

Fuente: elaboración propia.

Nótese que los puertos en modo troncal no son mostrados en la salida de ninguno de estos comandos (el puerto Fa 0/1 no se encuentra en la lista de puertos).

3.13.3. Enlaces troncales

Están formados por puertos especiales que posibilitan la transmisión de información de múltiples VLANs a través de un solo enlace y que pueden utilizar uno de los siguientes protocolos para encapsular la información.

- *Inter-Switch Link* (ISL): un protocolo propietario de Cisco ahora deprecado y no soportado en todos los dispositivos de este fabricante, por lo que su funcionamiento y comportamiento no serán discutidos en este trabajo.

- 802.1Q: el estándar abierto creado por la IEEE, cuyo comportamiento y características moldearán el resto de las discusiones presentadas.

La función de un puerto troncal consiste en marcar o etiquetar (de ahí el nombre de puerto etiquetado) las tramas con la información de la VLAN de donde se originaron, antes de enviarlas hacia otro dispositivo donde serán recibidas por otro puerto, con el mismo rol, el cual removerá dicha marcación para luego reenviar dichas tramas a la VLAN correcta.

Al contrario de otros fabricantes en donde deben ser agregadas manualmente, Cisco permite por defecto la transmisión de todas las VLANs a través de los enlaces troncales, siendo posible limitar las mismas con el comando *switchport trunk allowed vlan.*

Figura 152. **Comando *switchport trunk allowed vlan***

```
Switch(config-if)# switchport trunk allowed vlan ?

WORD    VLAN IDs of the allowed VLANs when this port is in trunking mode
add     add VLANs to the current list
all     all VLANs
except  all VLANs except the following
none    no VLANs
remove  remove VLANs from the current list
```

Fuente: elaboración propia.

A manera de ejemplo, si el objetivo fuera permitir solamente las VLAN 10 y 20 a través del enlace, se podría utilizar el comando anterior de la siguiente manera.

Figura 153. **Uso del comando *switchport trunk allowed vlan***

```
Switch(config-if)# switchport trunk allowed vlan 10,20
```

Fuente: elaboración propia.

Para mostrar una lista de las interfaces troncales, las VLANs que pueden ser transmitidas a través de los mismos, así como otros detalles importantes se utiliza la instrucción *show interfaces trunk.*

Figura 154. **Uso del comando *switchport trunk allowed vlan***

```
Switch# show interfaces trunk

Port        Mode         Encapsulation  Status        Native vlan
Fa0/1       on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/1       10,20

Port        Vlans allowed and active in management domain
Fa0/1       10,20

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/1       10,20
```

Fuente: elaboración propia.

3.13.4. *Dynamic trunking protocol* (DTP)

Es un protocolo propietario de Cisco que busca simplificar el uso del *switch* al usuario final, al negociar automáticamente el modo de un puerto para que este funcione como un puerto de acceso o uno troncal.

Para tratar de establecer un enlace troncal automáticamente con otro dispositivo, es posible configurar el puerto de un *switch* como *dynamic desirable, dynamic auto* o directamente en modo *trunk* (cuya configuración se muestra nuevamente para completar el ejemplo).

Figura 155. **Configuración de enlace troncal**

```
Switch(config)#interface fastethernet 0/3
Switch(config-if)#switchport mode dynamic desirable

Switch(config-if)#interface fastethernet 0/4
Switch(config-if)#switchport mode dynamic auto

Switch(config-if)#interface fastethernet 0/5
Switch(config-if)#switchport mode trunk
```

Fuente: elaboración propia.

No todas las combinaciones de los modos mencionados resultan en la formación de un enlace troncal debido a diferencias en su funcionamiento.

Los modos *dynamic desirable* y *trunk* tratarán activamente de formar un enlace troncal, mientras que en el modo *Dynamic Auto,* el puerto será un troncal solamente cuando el otro lado lo solicita. De esta manera si en los dos extremos de un enlace se tienen los modos *dynamic auto* y *dynamic desirable* respectivamente, se volverá un enlace troncal, mientras que si los dos extremos están configurados como *dynamic auto,* el enlace troncal no se formará, quedando los puertos en el modo de acceso.

La ejecución de DTP constituye un gran riesgo de seguridad dentro de una red, ya que un atacante podría negociar, a través de un dispositivo real o software especial, un enlace troncal con el *switch* donde está conectado y tener acceso a todas las VLANs existentes, ataque conocido como *switch spoofing.*

Por esta razón, una mejor práctica es configurar manualmente tanto los puertos troncales como los de acceso, para luego deshabilitar DTP en todas las interfaces, con el comando que se muestra en la figura 156.

Figura 156. **Comando para desabilitar**

Switch(config-if)# *switchport* nonegotiate

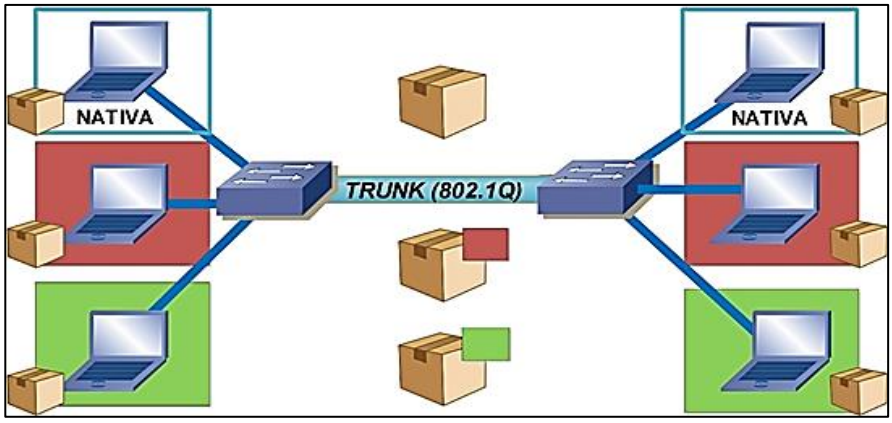
Fuente: elaboración propia.

3.13.5. VLAN nativa

Por cuestiones de retrocompatibilidad e interoperabilidad con otros dispositivos, el estándar 802.1Q ofrece la posibilidad de que las tramas sean enviadas con o sin etiqueta a través de un enlace troncal.

Las tramas sin etiquetar destinadas, originalmente a dispositivos incapaces de trabajar con dicha marcación, pertenecen a una VLAN especial llamada VLAN nativa.

Figura 157. **Tramas pertenecientes a la VLAN Nativa se envían sin etiquetar a través de un enlace troncal**



Fuente: elaboración propia, empleando *Edraw Max*.

Cualquier VLAN, existente o no dentro del *switch*, puede tomar el rol de la VLAN nativa en un troncal, responsabilidad que recae por defecto sobre la VLAN 1 donde están asignados todos los puertos de manera predeterminada, lo que a menudo es fuente de confusión.

La VLAN 1 tiene una importancia especial al ser utilizada como VLAN predeterminada por todos los fabricantes. En Cisco es empleada también, para la transmisión de ciertos protocolos de control, tales como el *Cisco Discovery Protocol* (CDP), *Port Aggregation Protocol* (PAgP) y *Vlan Trunking Protocol* (VTP) independientemente de si la VLAN 1 es la VLAN nativa o no.

Por esta razón la VLAN 1 no puede ser eliminada de un *switch* ni completamente filtrada de un enlace troncal.

Tomando en cuenta que en un puerto troncal de los dispositivos Cisco todas las VLAN son permitidas por defecto, es posible utilizar el siguiente comando para intentar remover la VLAN 1 de dicho enlace (ver figura 158).

Figura 158. **Comando para remover VLAN**

```
Switch(config-if)# switchport trunk allowed vlan remove 1
```

Fuente: elaboración propia.

Al ejecutar el comando anterior se consigue filtrar todo el tráfico enviado por los usuarios asignados a esa VLAN en particular, mas no así el tráfico de los protocolos de control mencionados anteriormente que continuarán funcionando con normalidad.

De manera separada a toda la funcionalidad que acaba de explicarse, la VLAN 1 se utiliza también de manera predeterminada para cumplir la función de la VLAN nativa, aunque esto puede cambiarse de manera individual dentro de cada puerto troncal, como se muestra en la figura 159.

Figura 159. **Cambio de de VLAN Nativa**

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# switchport trunk native vlan ?
  <1-1005>  VLAN ID of the native VLAN when this port is in trunking mode
Switch(config-if)# switchport trunk native vlan 100
```

Fuente: elaboración propia.

Nótese ahora, la VLAN que enviará sus paquetes sin ningún tipo de marcación será la VLAN 100; mientras que la funcionalidad de la VLAN 1 permanecerá inalterada, con la única diferencia de que el tráfico generado por los protocolos que hacen uso de la misma serán etiquetados, lo que tampoco impedirá su correcto funcionamiento.

Una consideración importante es que al ser configurada sobre cada puerto de manera individual, existe la posibilidad de configurar un enlace cuyos dos extremos utilicen una VLAN nativa diferente, condición que se conoce como discrepancia de VLANs nativas (*native VLAN mismatch*) y que en el caso de Cisco puede ser detectada gracias al Cisco Discovery Protocol (CDP), el cual bloqueará las VLANs en conflicto para evitar problemas más serios dentro de la red.

Figura 160. **Native VLAN *Mismatch***

*Switch*1(config)# interface fastethernet 0/1
*Switch*1(config-if)# *switchport* mode trunk

*Switch*2(config)# interface fastethernet 0/1
*Switch*2(config-if)# *switchport* mode trunk
*Switch*2(config-if)# *switchpor*t trunk native vlan 10
*Switch*2(config-if)#
%CDP-4-NATIVE_VLAN_MISMATCH: Native VLAN mismatch discovered on FastEthernet0/1 (**10**), with *Switch* FastEthernet0/1 (**1**).

Fuente: elaboración propia.

Los protocolos que hacen uso de la VLAN nativa para intercambiar información son DTP y *Spanning-Tree Protocol* (STP). Este último es un protocolo cuya función consiste en evitar bucles de capa 2.

La elección de una VLAN nativa no es sencilla, deben considerarse varios factores y aun así, no será posible llegar a una solución satisfactoria para todos los casos.

El primer factor será la interoperabilidad con dispositivos de otros fabricantes. Algunos de estos solo son capaces de utilizar la VLAN 1 como VLAN nativa, forzando a los demás *switches* a utilizar la misma configuración.

El segundo factor es el de la seguridad. El estándar 802.1Q no incluye ninguna limitación acerca del número de etiquetas presentes en una misma trama, por lo que es posible que un atacante marque o doble marque su propia información con la intención de alcanzar una VLAN diferente, ataque que se conoce como salto entre VLANs (*VLAN Hopping*). Para mitigar este ataque se presentan las siguientes opciones:

- No asignar ningún puerto en modo de acceso del *switch* a la VLAN nativa.

- Filtrar la VLAN nativa de los puertos troncales. Lo que no es recomendado, ya que puede interrumpir el correcto funcionamiento de ciertos protocolos.

- Forzar la marcación de los paquetes provenientes de la VLAN nativa, lo que puede realizarse desde el modo de configuración global (ver figura 161).

Figura 161. **Marcacion de la VLAN Nativa en el modo de configuración global**

```
Switch(config)# vlan dot1q tag native
```

Fuente: elaboración propia.

O en cada puerto troncal; como se observa en la figura 162.

Figura 162. **Marcación de la VLAN Nativa en puerto troncal**

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# switchport trunk native vlan tag
```

Fuente: elaboración propia.

Esta opción no está disponible en toda la gama de *switches* Cisco y podría no ser interoperable con dispositivos de otros fabricantes.

Dadas las consideraciones anteriores se recomienda, por razones de diseño y seguridad, no utilizar la VLAN 1 ni la VLAN nativa en ningún puerto de acceso, o en otras palabras, no utilizar estas dos VLANs especiales en redes destinadas a los usuarios.

3.14. VLAN *trunking protocol* (VTP) *e inter VLAN routing*

Para comenzar esta sección se explica el funcionamiento de VTP.

3.14.1. VLAN trunking protocol

Es un protocolo propietario de Cisco que tiene por objetivo facilitar la administración y configuración de las VLAN dentro de una infraestructura, permitiendo que los cambios realizados en un dispositivo se propaguen automáticamente a todos los demás *switches* dentro de un mismo dominio.

Perteneciente a la capa 2 del modelo OSI, VTP previene inconsistencias al utilizar los enlaces troncales (requisito indispensable) para sincronizar información entre varios *switches*.

El seguimiento de los cambios se realiza gracias a un número de revisión, el cual se incrementa cada vez que se crea, elimina o se cambia de nombre una VLAN; entre más alto es este parámetro más reciente es considerada la información, por lo que todos los *switches* buscarán sincronizarse con la información que posea el número de revisión más alto.

Actualmente existen tres versiones de este protocolo, las primeras dos presentan ligeras diferencias en su funcionamiento, mientras que la versión tres es una completa reestructuración del mismo.

Para configurar la versión a utilizar puede emplearse la instrucción que se detalla en la figura 163.

Figura 163. **Instrucción para configuración de versión a utilizar**

```
Switch(config)# vtp version ?
  <1-3>  Set the administrative domain VTP version number

Switch(config)# vtp version 2
```

Fuente: elaboración propia.

Además, VTP puede ser configurado para trabajar en una de las siguientes modalidades

- Servidor (*server*): este es el modo por defecto. Posibilita crear, eliminar y modificar VLANs y establecer parámetros (como la versión del protocolo), que serán utilizados en todos los *switches* dentro de un dominio, dentro del cual se recomienda la existencia de un solo dispositivo que cumpla con esta función.

- Cliente (*client*): en este modo no es posible realizar ningún cambio en la configuración de las VLAN. Los dispositivos que tomen este rol heredarán aquella información proporcionada por el servidor. Sin embargo, si un *switch* en modo cliente es incorporado a la topología con un número de revisión mayor al de cualquier otro dispositivo, este enviará la información más reciente al servidor, para luego ser propagada al resto del dominio.

- Transparente (*transparent*): este modo no participa en el proceso de sincronización de VTP, los *switches* configurados para desempeñar este

rol tendrán control sobre sus propias VLAN y se limitarán a reenviar las actualizaciones VTP recibidas a través de sus enlaces troncales para que estas puedan alcanzar otros dispositivos.

- Apagado (*Off*): deshabilita VTP, los *switches* tendrán control sobre sus propias VLAN y la información relacionada con VTP no será reenviada. Esta modalidad no está disponible en todas las plataformas.

Para configurar el modo VTP se puede emplear la instrucción descrita en la figura 164.

Figura 164. **Configuración modo VTP**

```
Switch(config)# vtp mode ?
  client          Set the device to client mode.
  server          Set the device to server mode.
  transparent     Set the device to transparent mode.
```

Fuente: elaboración propia.

Otra característica de VTP es que, para funcionar correctamente requiere de la capacidad de almacenar información relacionada con el estado de las VLAN y los cambios hechos en las mismas, así como otros parámetros necesarios para su funcionamiento, de manera automática en un espacio de memoria no volátil (memoria que no necesita de energía para perdurar); funcionalidad presente en sistemas operativos antiguos, pero no en el Cisco IOS.

Para superar este problema se creó un archivo especial donde pudiera almacenarse dicha información de manera dinámica y se le dio el nombre de vlan.dat, usualmente referido como VLAN *Database.*

De esta manera, en aquellos modos donde VTP es completamente funcional (servidor y cliente), toda información respecto a su configuración y a las VLAN es almacenada en dicha base de datos y no será mostrada en el archivo de configuración. En las otras modalidades es el caso contrario.

Figura 165. **Base de datos**

```
Switch(config)# vtp mode server
Device mode already VTP SERVER.
Switch# show running-config | include vlan
Switch#



Switch(config)# vtp mode transparent
Setting device to VTP TRANSPARENT mode.
Switch# show running-config | include vlan
vlan 10
vlan 20
vlan 30
Switch(config)#
```

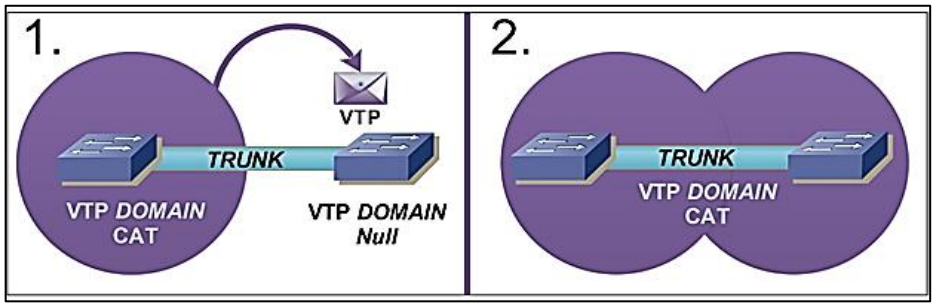
Fuente: elaboración propia.

A pesar de haber establecido la versión y el modo a utilizar, VTP no comenzará a enviar publicaciones hasta que se haya especificado un dominio administrativo dentro del *switch,* ya que de manera predeterminada estos no pertenecen a ninguno, estando este campo indefinido, por lo que se dice que tiene un valor indeterminado (*null*).

Un dominio indeterminado (*domain null*) requiere especial atención, ya que además del comportamiento descrito anteriormente ocasionará que el dispositivo, al recibir una publicación VTP en uno de sus enlaces troncales, pase a formar parte del dominio incluido en la misma para luego heredar toda su información.

Este proceder, exclusivo de las versiones 1 y 2 de VTP, se incluyó en el diseño de este protocolo con el objetivo de facilitar la incorporación de nuevos *switches* a una infraestructura, aunque en la actualidad, el mismo no es conveniente desde el punto de vista de la seguridad en la red.

Figura 166. ***Switch* con un dominio indeterminado**



Fuente: elaboración propia, empleando *Edraw Max*.

Para establecer un dominio se utiliza el comando, que se muestra en la figura 167, nótese que un *switch* puede pertenecer solamente a un dominio administrativo.

Figura 167. **Comando para establecer dominio**

```
Switch(config)# vtp domain CAT
Changing VTP domain name from NULL to CAT
```

Fuente: elaboración propia.

Otra medida que puede tomarse es la configuración de una contraseña para uso dentro del dominio. Dicha contraseña no impedirá que un *switch* pase a formar parte de un dominio de manera automática, pero sí evitará la sincronización de la información entre dispositivos.

Figura 168. **Contraseña para uso dentro del dominio**

```
Switch(config)# vtp password FOX
```

Fuente: elaboración propia.

Un protocolo que depende de la correcta configuración de VTP es el *dynamic trunking protocol* (DTP).

DTP negociará un enlace troncal entre dos *switches,* si ambos se encuentran en el mismo dominio o en el caso de que uno o los dos dispositivos tengan un dominio indeterminado. De otra forma, no se negociará dicho enlace debido a un error llamado *domain mismatch.*

Figura 169. ***Domain mismatch***

```
Sw1(config)# vtp domain CAT
Changing VTP domain name from NULL to CAT

Sw2(config)# vtp domain HAT
Changing VTP domain name from CAT to HAT
00:10:22 %DTP-5-DOMAINMISMATCH: Unable to perform trunk negotiation on port Fa0/1 because of VTP domain mismatch.
```

Fuente: elaboración propia.

La única manera de visualizar la configuración de VTP (almacenada en el vlan.dat) es utilizando el comando *show vtp status,* el cual se muestra en la figura 170. Los demás comandos proporcionan el contexto y complementan el ejemplo.

Figura 170. **Comando *show vtp status***

```
Switch(config)#vtp mode server
Device mode already VTP SERVER.
Switch(config)# vtp password FOX
Setting device VLAN database password to FOX
Switch(config)# vtp domain CAT
Changing VTP domain name from NULL to CAT
Switch(config)# vtp version 2

Switch#
%SYS-5-CONFIG_I: Configured from console by console

Switch# show vtp status
VTP Versión                          : 2
Configuration Revisión               : 0
Maximum VLANs supported locally      : 255
Number of existing VLANs             : 5
VTP Operating Mode                   : Server
VTP Domain Name                      : CAT
VTP Pruning Mode                     : Disabled
VTP V2 Mode                 : Disabled
VTP Traps Generation                 : Disabled
MD5 digest                  : 0xC5 0x1F 0xC8 0x2C 0x6F 0xF9 0x91 0x53

Configuration last modified by 0.0.0.0 at 0-0-00 00:00:00
Local updater ID is 0.0.0.0 (no valid interface found)
```

Fuente: elaboración propia.

Entre la información más importante puede apreciarse el número de revisión (*configuration revision*), el nombre del dominio y el modo que se está utilizando. También puede observarse la dirección IP del último dispositivo en actualizar la base de datos de las VLAN (*configuration last modified by*) y la dirección IP del dispositivo mismo (*local updater ID*), suponiendo en ambos que los dispositivos cuentan con al menos una dirección asignada a través de alguna interfaz virtual. Para configurar manualmente la dirección a ser utilizada por VTP, se selecciona la interfaz que posea la dirección deseada de la siguiente manera.

Figura 171. **Selección de interfaz**

```
Switch(config)# vtp interface ?
  WORD  The name of the interface providing the VTP updater ID for this device.
```

Fuente: elaboración propia.

Un parámetro importante, que no se muestra utilizando los métodos descritos anteriormente, es la contraseña utilizada por VTP, misma que solo se puede visualizar utilizando el siguiente comando.

Figura 172. **Comando para visualizar contraseña**

```
Switch# show vtp password
VTP Password: FOX
```

Fuente: elaboración propia.

A pesar de todos los beneficios explicados, VTP introduce también, el riesgo de interrumpir todas las operaciones de la red, de manera intencional o accidental, ya sea por atacantes o usuarios inexpertos, por lo que en la mayoría de las organizaciones no es utilizado.

Dichos riesgos han sido mitigados o completamente eliminados en la versión 3 de este protocolo, no obstante, dicha versión no está disponible para todas las plataformas.

Finalmente es necesario agregar, que aunque no se planee implementar este protocolo, es necesario conocer su funcionamiento y comportamiento, para poder desactivarlo apropiadamente.

3.14.2. Inter VLAN *routing*

La creación de las VLAN permite la segmentación de la red al posibilitar la existencia de varios dominios de *broadcast* dentro de un mismo *switch.*

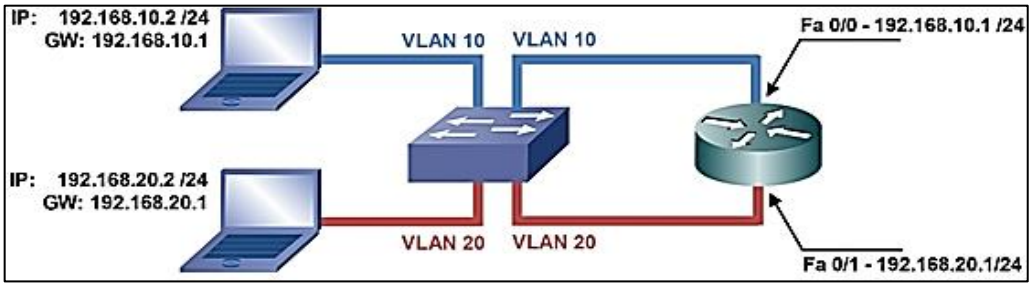
Estos dominios funcionan de manera independiente, sin comunicación alguna entre ellos, a pesar de residir físicamente en el mismo dispositivo, por lo que cada uno de los mismos puede ser utilizado para albergar una red diferente.

Por esta razón, para volver a establecer comunicación entre varias VLANs (*inter vlan routing*), se necesita de un dispositivo capaz de enrutar entre varias redes, pudiendo elegirse entre las siguientes opciones.

3.14.2.1. Un *router* con una interfaz para cada VLAN

En este caso la solución es directa, por cada VLAN existe una interfaz separada en el *router,* que luego se encarga de establecer comunicación entre ellas. No obstante, esta solución no es económica ni escalable, ya que se tiene que incorporar una nueva interfaz por cada nueva VLAN.

Figura 173. ***Router*** **con una interfaz por cada VLAN**



Fuente: elaboración propia, empleando *Edraw Max*.

3.14.2.2. Router en un palo (*router on a stick*)

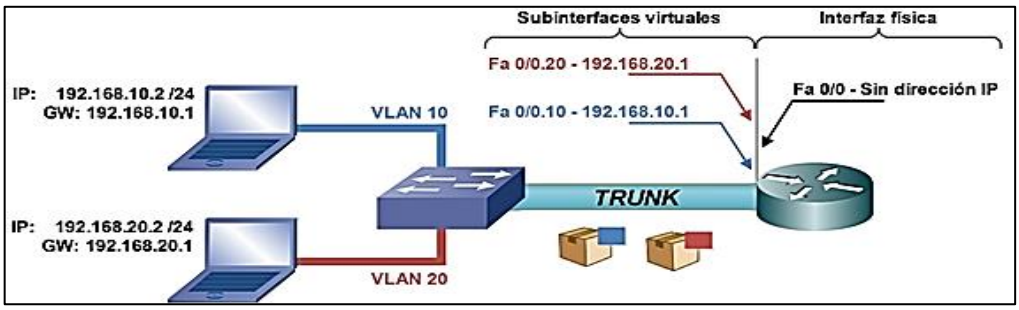
En esta solución se utiliza un solo enlace troncal para llevar el tráfico de todas las VLAN a una interfaz física del *router* (de ahí su nombre); en donde se crearán varias subinterfaces virtuales, siendo cada una de ellas destinada a manejar los datos y servir como puerta de enlace predeterminada de una VLAN específica.

Para que dichas subinterfaces puedan separar y procesar correctamente el tráfico proveniente de cada VLAN es necesario configurar dentro de cada una el protocolo utilizado en el enlace troncal, así como la etiqueta específica asociada con el tráfico que se pretende manejar.

Este arreglo es más económico y escalable que la solución presentada anteriormente, sin embargo, introduce un único punto de falla, el enlace troncal, así como un cuello de botella dentro de la red, ya que el tráfico de todas las VLANs debe pasar necesariamente por dicho enlace.

Para mostrar la implementación de esta solución se usará la siguiente topología. En ella se ha configurado previamente el *switch* con las VLAN indicadas y también se han configurado los puertos en los modos necesarios.

Figura 174. ***Router* en un palo (*router on a stick)***



Fuente: elaboración propia, empleando *Edraw Max*.

Antes de comenzar es importante comprender, que dentro de cada red existen dos tipos diferentes de topologías: la física y la lógica. La topología física es la que nos muestra la disposición de los dispositivos, las interconexiones entre ellos y los cables utilizados para las mismas. Mientras que la topología lógica está compuesta por aquellas construcciones invisibles, formadas por dispositivos e interfaces virtuales, las cuales afectan las rutas que atraviesa la información de un punto a otro de la red y que deben configurarse de la misma manera que sus contrapartes reales.

De esta manera, al analizar la topología anterior, se tiene que físicamente el tráfico de todas las VLANs llega a la interfaz *FastEthernet* 0/0 del *router* mostrado a través de un enlace troncal. No obstante, desde un punto de vista lógico el enlace troncal no existe y cada VLAN está conectada a este dispositivo a través de su propia interfaz.

Así pues, físicamente la interfaz *FastEthernet* 0/0 debe estar encendida para recibir la información proveniente del enlace troncal, sin embargo, a nivel lógico dicha interfaz no recibe ningún tipo de tráfico, por lo que no necesita de ninguna otra configuración, ni siquiera de una dirección IP.

Figura 175. **Interface FastEthernet0/0**

```
Router(config)#interface fastEthernet 0/0
Router(config-if)#no shutdown
Router(config-if)#exit

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
```

Fuente: elaboración propia.

Esta dualidad es un poco difícil de comprender en un inicio. Si bien es cierto que el flujo de la información está determinado por la composición lógica de la red, no puede olvidarse que esta depende del correcto funcionamiento de la parte física en todo momento.

Siguiendo esta línea de razonamiento se crearán, sobre la interfaz física *FastEthernet* 0/0, subinterfaces virtuales para manejar el tráfico de las VLANs presentadas, como se muestra a continuación.

Figura 176. **Subinterfaces virtuales**

```
Router(config)#interface fastethernet 0/0.?
  <0-4294967295>  FastEthernet interface number

Router(config)#interface fastethernet 0/0.10
```

Fuente: elaboración propia.

Al agregar un punto (“.”) después del nombre de la interfaz, puede accederse a la configuración de las subinterfaces virtuales. El rango de posibles valores (0-4294967295) tiene como propósito proporcionar flexibilidad a la hora de elegir un valor y no indica la cantidad de subinterfaces que este *router* puede manejar.

Cual sea el valor elegido para designar una subinterfaz, no tendrá ninguna injerencia sobre su funcionamiento. Sin embargo, se recomienda elegir un nombre relacionado con el propósito de la misma, lo que más adelante facilitará su manejo y la resolución de problemas dentro de la red.

Para utilizar la subinterfaz recién creada para procesar el tráfico proveniente de una VLAN, debe especificarse el protocolo utilizado por el enlace troncal para encapsular la información (*encapsulation*), así como el valor de la etiqueta respectiva.

En este caso, dicho protocolo será el estándar abierto 802.1Q o “punto 1Q” (*dot1q*) y el tráfico que se quiere manejar es el de la VLAN 10.

Figura 177. **Encapsulación**

```
Router(config-subif)# encapsulation dot1q 10
```

Fuente: elaboración propia.

Una vez configurada tanto la encapsulación como la etiqueta del tráfico a procesar, se vuelve posible especificar una dirección IP para la subinterfaz de la misma manera como se haría con una interfaz real.

Figura 178. **Dirección IP para la subinterfaz**

```
Router(config-subif)#ip address 192.168.10.1 255.255.255.0
Router(config-subif)# exit
```

Fuente: elaboración propia

Ahora esta subinterfaz puede ser utilizada como puerta de enlace predeterminada de todos los dispositivos dentro de la VLAN 10.

Es posible seguir el mismo proceso para crear la subinterfaz encargada de manejar el tráfico de la VLAN 20.

Figura 179. **Subinterfaz para tráfico**

```
Router(config)# interface fastethernet 0/0.20
Router(config-subif)# encapsulation dot1q 20
Router(config-subif)# ip address 192.168.20.1 255.255.255.0
```

Fuente: elaboración propia.

Para mostrar el estado de todas las interfaces, físicas y virtuales se utiliza nuevamente el comando *show ip interface brief.*

Figura 180. **Comando *show ip interface brief***

```
Router# show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	up	up
FastEthernet0/0.10	192.168.10.1	YES	manual	up	up
FastEthernet0/0.20	192.168.20.1	YES	manual	up	up
FastEthernet0/1	unassigned	YES	unset	administratively down	down
Vlan1	unassigned		unset	down	

Fuente: elaboración propia.

Finalmente, al revisar la tabla de enrutamiento se puede apreciar que el *router* tiene conocimiento de las redes utilizadas para cada una de las VLAN, ya que estas se encuentran conectadas directamente a través de las subinterfaces creadas anteriormente.

Todo tráfico entre VLANs debe pasar necesariamente a través del *router,* el cual redirigirá el tráfico y modificará las etiquetas para establecer conectividad.

Figura 181. **Uso del comando show *ip route***

```
Router# show ip route

Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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

Gateway of last resort is not set


C       192.168.10.0/24 is directly connected, FastEthernet0/0.10
C       192.168.20.0/24 is directly connected, FastEthernet0/0.20
```

Fuente: elaboración propia.

3.14.2.3. *Switch* multicapa (*multilayer switch*)

Es un dispositivo que, además de proveer las funciones típicas de un *switch* presta otros servicios que operan en otras capas del modelo OSI (de ahí su nombre).

Una de las capacidades del *switch* multicapa es la de enrutar paquetes a gran velocidad utilizando *hardware* dedicado, misma que se encuentra deshabilitada por defecto, pero que puede activarse utilizando el comando que se describe en la figura 182.

Figura 182. **Comando de activación**

```
ML Switch(config)# ip routing
```

Fuente: elaboración propia.

De esta manera, los puertos de este dispositivo pueden funcionar tanto a nivel de la capa 2, como los de un *switch* tradicional o a nivel de la capa 3, en cuyo caso podrá asignársele una dirección IP y utilizarla junto con algún protocolo de enrutamiento.

La funcionalidad por defecto depende de la plataforma, pero es posible alterarlo utilizando el comando *switchport* (Capa 2) y la negación del mismo *no switchport* (Capa 3), como se muestra en la figura 183.

Figura 183. **Comando *switchport***

```
MLSwitch(config)#interface fastEthernet 0/1
MLSwitch(config-if)#switchport
MLSwitch(config-if)#switchport mode access
MLSwitch(config-if)#switchport access vlan 10

MLSwitch(config-if)#interface fastEthernet 0/2
MLSwitch(config-if)#no switchport
MLSwitch(config-if)#ip address 172.16.1.1 255.255.255.0
```

Fuente: elaboración propia.

Otra posibilidad que ofrece este *switch* es la creación de múltiples *switched virtual interfaces* (SVI), interfaces virtuales íntimamente ligadas a las VLAN y que pueden ser usadas por estas como puertas de enlace predeterminadas. Para crear una SVI puede utilizarse el comando que se muestra en la figura 184.

Figura 184. **Comando para crear una SVI**

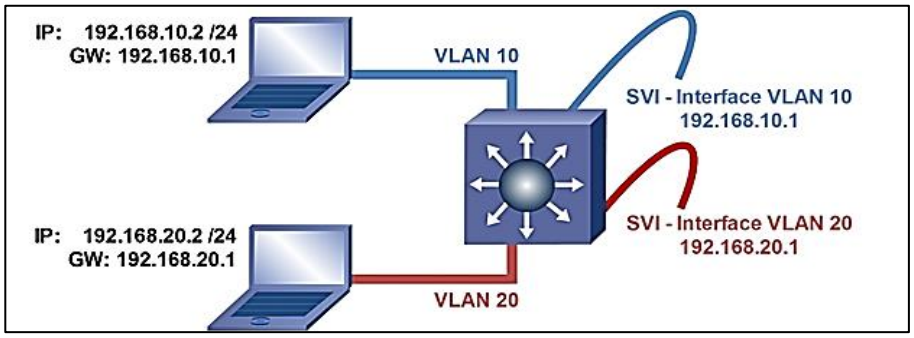
```
MLSwitch(config)# interface vlan 100
MLSwitch(config-if)# no shutdown
```

Fuente: elaboración propia.

Para que una SVI sea operacional, la VLAN relacionada debe existir y estar activa dentro del *switch*, es decir, que debe haber, por lo menos, un puerto activo perteneciente a dicha VLAN y un enlace troncal en donde dicha VLAN sea permitida y que no haya sido bloqueada, ya sea manualmente o por algún protocolo (VTP, *Spanning Tree*, entre otros).

Para mostrar la implementación de las SVI, se presenta la siguiente topología, donde las VLAN y los puertos han sido previamente configurados. Se puede observar nuevamente cómo se activan las capacidades de enrutamiento para que el ejemplo esté completo.

Figura 185. **Creación y configuración de SVI**



Fuente: elaboración propia.

Figura 186. **Enrutamiento SVI**

```
MLSwitch(config)# ip routing
MLSwitch(config)# interface vlan 10
MLSwitch(config-if)# no shutdown
MLSwitch(config-if)# ip address 192.168.10.1 255.255.255.0

MLSwitch(config)# interface vlan 20
MLSwitch(config-if)# no shutdown
MLSwitch(config-if)# ip address 192.168.20.1 255.255.255.0
```

Fuente: elaboración propia.

Al examinar el modelo de enrutamiento se puede obsevar la descripción de la figura 187.

Figura 187. ***Show ip route***

```
MLSwitch# show ip route
Codes: C - connected, S - static, I - IGRP, 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, E - EGP
       i - IS-IS, 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

Gateway of last resort is not set

C       192.168.10.0/24 is directly connected, Vlan10
C       192.168.20.0/24 is directly connected, Vlan20
```

Fuente: elaboración propia.

Al utilizar un *switch* multicapa para comunicar varias VLAN, se eliminan cuellos de botella y retrasos introducidos por otros dispositivos. Desde el punto de vista del mejor diseño de red, es la solución predilecta. Su utilización en el pasado estuvo limitada por el alto costo. En la actualidad, su uso se hace cada vez más frecuente; es implementado en segmentos de la red antes reservados para los *switches* tradicionales, tendencia que irá en aumento conforme su precio se vuelva más asequible.

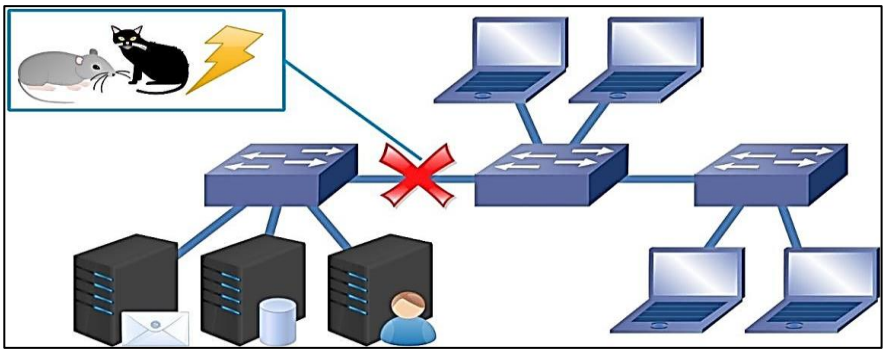
Finalmente es necesario añadir, que a pesar de ejecutar muchas funciones que en el pasado eran exclusivas a los *routers,* no es posible reemplazar estos últimos por *switches* multicapa en todos los casos. Antes de tomar una decisión respecto a cuál de los dos dispositivos debe utilizarse, es necesario tener completo entendimiento de los requerimientos que el diseño debe satisfacer, poniendo especial atención a características especiales como conexiones WAN, calidad de servicio y seguridad, entre otras.

3.15. *Spanning tree protocol* (STP)

De manera planeada o no, las redes de comunicaciones siempre tienden a crecer. Muchas veces ante la urgente necesidad de satisfacer nuevos requerimientos dicho crecimiento se realiza de forma desordenada, conectando los nuevos dispositivos a los ya existentes en cascada conforme estos se hacen necesarios.

Este tipo de disposición hace a la red más propensa a fallar al hacerla menos resiliente contra desperfectos mecánicos, eléctricos y factores externos.

Figura 188. **Una falla en una red pobremente diseñada puede limitar o eliminar completamente la conectividad**



Fuente: elaboración propia, empleando *Edraw Max*.

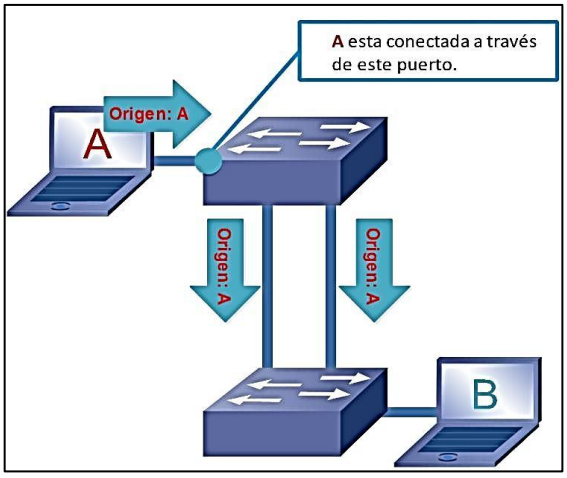
Esta es la razón por la que dentro de un buen diseño de red, siempre debe contemplarse el crecimiento de la misma e incluir también cierto grado de redundancia que permita reducir el tiempo inoperativo y así mitigar las pérdidas económicas derivadas.

No obstante, los evidentes beneficios, la implementación de dicha redundancia tiene su costo tanto económico como operacional. Un caso especialmente problemático es cuando se utiliza con *switches* tradicionales capaces de funcionar solamente a nivel de capa 2.

Considérese el siguiente escenario:

Dos *switches* han sido conectados entre sí utilizando enlaces redundantes y la computadora A trata de enviar un mensaje a la computadora B. Para averiguar la dirección física de B, la computadora A enviará un ARP *request* como un *broadcast* hacia el resto de la red. La trama llegará al *switch,* que examinará la dirección MAC de origen, para luego agregar una entrada en su memoria y situar al dispositivo A en el puerto por donde esta ingresó, antes de reenviar dicha trama por todas las demás interfaces.

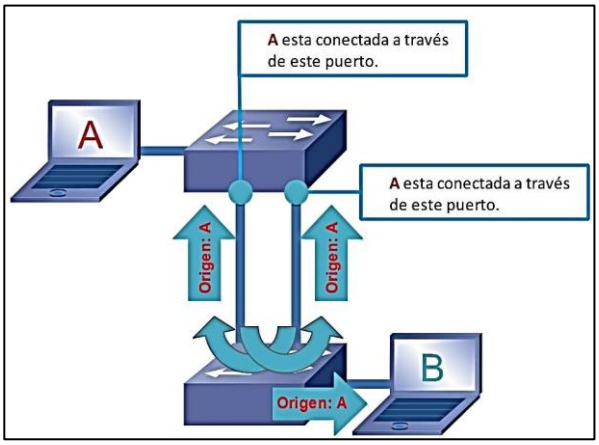
Figura 189. **El *switch* aprende la dirección MAC de la computadora A y reenvía el *broadcast* por todos sus otros puertos**



Fuente: elaboración propia, empleando *Edraw Max*.

La transmisión llega al *switch* inferior, el cual sigue el mismo proceso y reenvía las tramas a través de todos los puertos, exceptuando aquel que la recibió originalmente. De haber una sola conexión entre ambos *switches* dicha transmisión solo alcanzaría a la computadora B, sin embargo, al existir un enlace redundante, dicha información es retransmitida nuevamente al *switch* superior que ahora creerá que la computadora A esta conectada en algún punto del dispositivo inferior.

Figura 190. **El enlace redundante hace posible que el *broadcast* original retorne al dispositivo en donde se originó**

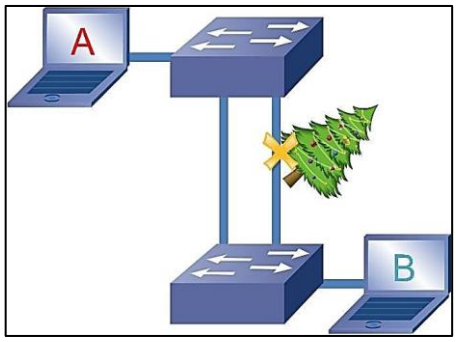


Fuente: elaboración propia, empleando *Edraw Max*.

La inestabilidad en la base de datos de direcciones MAC provocará cada vez más retransmisiones y generará más *broadcast,* el cual, al ser procesado en *software* (por lo que carga al CPU) y ser dirigido hacia todos los dispositivos, llevará a la red a un paro general cuando el límite sea sobrepasado, condición que se conoce como una tormenta de *broadcast.*

Para mitigar dicho problema se utiliza un protocolo capaz de detectar y bloquear enlaces redundantes, para evitar la formación de bucles a nivel lógico, al mismo tiempo que se mantiene la redundancia física en la red. Este protocolo es conocido como *spanning tree protocol* (STP).

Figura 191. ***Spanning tree* desactiva a nivel lógico los enlaces redundantes para prevenir bucles**



Fuente: elaboración propia, empleando *Edraw Max*.

STP fue creado en 1985, por Radia Perlman con el propósito de que los dispositivos de capa 2 pudieran detectar y bloquear enlaces redundantes y luego reactivarlos en caso de una falla. El estándar abierto fue publicado en 1990, con el nombre de 802.1D y presentaba algunas diferencias con respecto al original.

Con el paso de los años se han realizados enmiendas y creado nuevas implementaciones a partir de la primera versión (ej.: 802.1s, 802.1w), todas están contenidas en el estándar 802.1Q.

Es importante remarcar que STP es un protocolo antiguo que ha mantenido mucha de su terminología original, por esta razón algunas definiciones y configuraciones hacen referencia al dispositivo antecesor al

switch: el *bridge.* De esta manera, un bucle entre *switches* es referido como un *bridging loop* y no como un *switching loop* (término más intuitivo). No obstante, durante el resto de la discusión de este protocolo se favorecerá el término *switch* (cuando sea posible) por motivos pedagógicos.

3.15.1. Operación

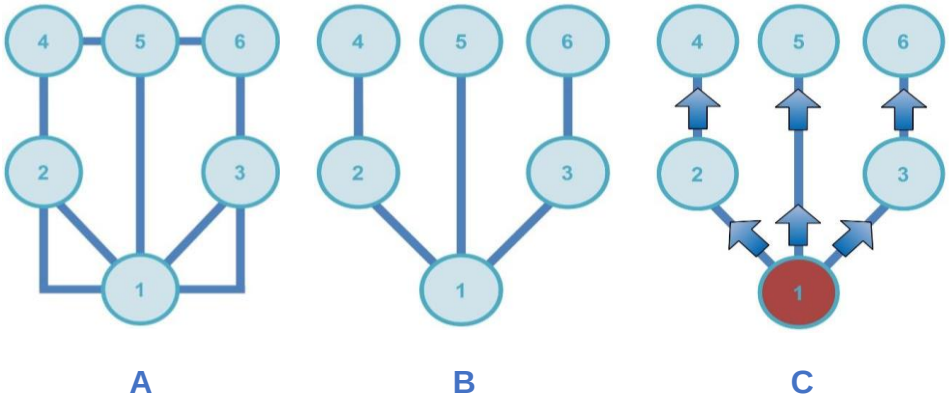
Antes de analizar la operación de STP, hay que considerarse las siguientes definiciones:

- Un árbol (*tree*) en teoría de grafos (una rama de las matemáticas discretas), es una gráfica sin ninguna orientación en especial en donde dos vértices están conectados exactamente por una sola ruta.

- Un árbol enraizado (*rooted tree*), es un tipo de árbol en donde se ha elegido un vértice como punto de referencia o como “raíz” del grafo para agregar estructura y orientación.

- Un árbol de expansión (*spanning tree*), es un compuesto por todos los vértices en un grafo.

Figura 192. **Grafos**



Fuente: elaboración propia, empleando *Edraw Max*.

De esta manera STP incluye a todos los *switches* presentes en un mismo dominio de *broadcast,* dentro de los cuales elegirá a uno en especial para servir como punto de referencia dentro de la topología recibiendo este el nombre de *switch* raíz (*root bridge*), dispositivo a partir del cual se elegirá una y solamente una ruta (compuesta por los mejores enlaces acorde a varios criterios) hacia los demás *switches,* bloqueando todos los demás enlaces al ser considerados redundantes.

Durante el resto de este trabajo se utilizarán los términos *spanning tree* y STP de manera intercambiable.

3.15.1.1. *Bridge protocol data units* (BPDUs)

Al contrario de los protocolos de estado de enlace (*link state*), en donde cada uno de los dispositivos conoce a todos los demás presentes en la topología, los *switches* que ejecutan *spanning tree* trabajan de manera independiente, ajenos a los demás dispositivos y su colocación dentro de la red.

Así pues, para realizar las elecciones necesarias y detectar cambios en la topología, los *switches* envían y reciben a través de sus interfaces tramas especiales llamadas *Bridge protocol data units* (BPDUs), las cuales son enviadas a un grupo de *multicast* al que solo pertenecen los dispositivos que ejecutan STP.

Las BPDUs incluyen mucha información y pueden ser clasificadas acorde a su propósito como:

- BPDU de configuración (*Configuration* BPDU). Incluyen toda la información necesaria para realizar los cálculos requeridos por *Spanning Tree* (Información del *switch, timers,* entre otros)*,* y que son generadas únicamente por el *switch* raíz para luego ser propagadas al resto de dispositivos.

- BPDU de notificación de cambio de topología (*topology change notification – TCN BPDU*). Es utilizada para manejar los cambios que ocurren dentro de la topología. Se generan cuando una interfaz que pertenece al proceso de STP cambia de estado, para luego propagarse de dispositivo a dispositivo hasta llegar al *switch* raíz que indicará a los demás que deben renovar su base de datos de direcciones MAC en un tiempo más corto de lo usual, para que estos puedan ajustarse al cambio.

3.15.1.2. Estados de spanning tree

Los puertos que participan de *spanning tree* pueden estar en uno de los siguientes estados.

- Bloqueando (*blocking*). En este estado, el puerto no es capaz de enviar o recibir información ni de aprender direcciones MAC. Es el estado inicial de todos los puertos (con el propósito de evitar la formación de bucles cuando el dispositivo inicia) y aquel al que regresan aquellos que deben estar bloqueados para quebrar bucles en los enlaces redundantes. En esta fase los puertos no pueden enviar BPDUs, por lo que se limitan a procesar aquellas que reciben.

  Si el puerto se está inicializando y es considerado como candidato para ser utilizado para el envío de información o si se necesita utilizar dicho puerto a causa de una falla en la red, pasará al siguiente estado. Puede llegar a demorarse hasta 20 segundos según sea el caso.

- Escuchando (*listening*). Los puertos que son considerados por el *switch* como candidatos para empezar a enviar información, pasan a este estado que no permite enviar o recibir información ni aprender direcciones MAC. La diferencia entre esta fase y la anterior radica en que el puerto, además de recibir y procesar, también puede enviar sus propias BPDUs, con lo que pasa a participar activamente en el proceso y de las decisiones tomadas por STP.

  Es también en esta etapa donde se decide si un puerto estará bloqueado o si será utilizado para transmitir datos. En este último caso pasará al siguiente estado después de 15 segundos.

- Aprendiendo (*learning*). Este es el último estado antes de empezar a transmitir. Se siguen recibiendo y enviando BPDUs, además el puerto comienza a aprender direcciones MAC. El tiempo que ha sido otorgado para este proceso es de 15 segundos.

- Transmitiendo (*forwarding*): en este estado el puerto es completamente operacional. Es capaz de recibir, enviar y procesar tanto información como BPDUs, asimismo de agregar entradas en la base de datos de direcciones MAC.

- Deshabilitado (*disabled*). estado en el cual el puerto ha sido apagado por un administrador o deshabilitado por algún protocolo. Este no forma parte directamente del proceso de *spanning tree,* pero que sí debe considerarse.

3.15.1.3. Roles en spanning tree

Después que la red haya convergido y que los puertos hayan pasado por uno o varios de los estados expuestos anteriormente (y que los mismos no se encuentren deshabilitados), estos pasarán a operar en uno de los siguientes roles.

- Puerto raíz (*root port*): es aquel que posee la mejor ruta hacia el *switch* raíz. Forma parte de los enlaces activos en la topología por lo que siempre estará transmitiendo (*forwarding*).

Este rol no existe en el *switch* raíz (por lo que a menudo es fuente de confusión). Está reservado para los demás dispositivos que ejecutan STP y que pueden tener un solo puerto cumpliendo esta función.

También es utilizado para comunicarle al *switch* raíz que ha ocurrido un cambio en la topología.

- Puerto designado (*designated port*): los puertos en este rol siempre se encuentran transmitiendo (*forwarding*) y son los únicos capaces de enviar BPDUs de configuración, por esta razón se encuentran presentes en todos los segmentos de la topología STP. En estos debe existir, necesariamente, un único puerto que cumpla esta función. Este diseño permite que *spanning tree* pueda detectar bucles, inclusive en enlaces conectados a un segmento compartido.

  Todos los puertos del *switch* raíz son puertos designados.

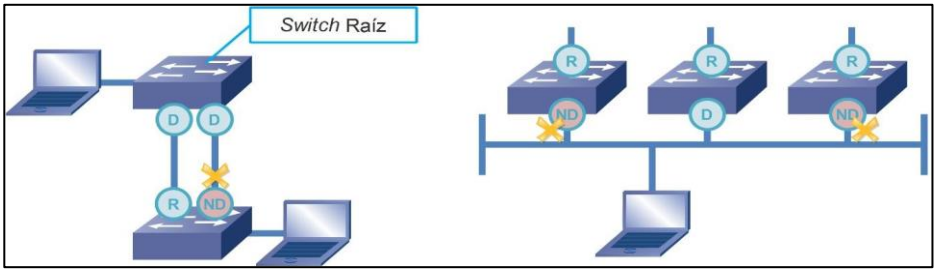
- Puerto no designado (*non-designated* port): este forma parte de un enlace redundante, por lo que no transmite información y se limita a escuchar las BPDUs provenientes de algún puerto designado (*Blocking*).

Así los enlaces activos están compuestos por un puerto raíz y uno designado; en tanto que los enlaces inactivos están formados por un puerto designado y uno que no lo está, a fin de romper el bucle.

Dentro de cada segmento siempre debe existir un único puerto designado. Este diseño permite a *spanning tree* tratar con enlaces redundantes conectados al mismo segmento (lo que no es común en estos días).

Estos roles no son mostrados en las salidas del Cisco IOS.

Figura 193. **Roles de los puertos**



Fuente: elaboración propia, empleando *Edraw Max*.

3.15.1.4. Elección del *switch* raíz y el rol de cada puerto

De acuerdo con lo expuesto, para llevar a la red desde su inicialización hasta lograr una topología lógica libre de bucles,

Spanning tree necesita elegir un *switch* raíz: determinar el puerto raíz en cada dispositivo y luego elegir los puertos designados dentro de cada segmento.

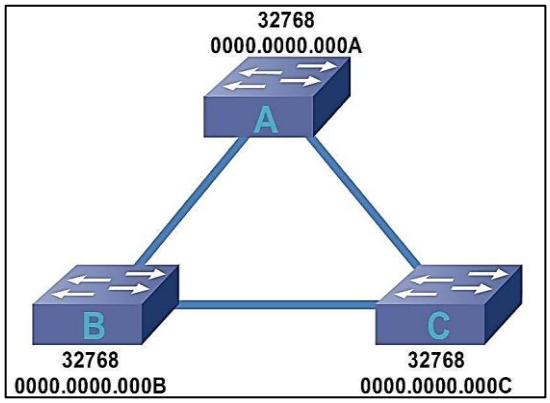
Lo criterios utilizados por STP para tomar estas decisiones tienen en común que siempre prefieren los valores más bajos de sus respectivos parámetros, como se explica posteriormente.

Para elegir el *switch* raíz se utiliza como parámetro el identificador del *switch* (*bridge ID*), un valor compuesto por la combinación de un campo conocido como la prioridad (valor numérico con un tamaño original de 2 bytes) y la dirección MAC del dispositivo. Es seleccionado como raíz aquel con el *bridge ID* más bajo.

Por recomendación del estándar (802.1D), todo *switch* comienza con una prioridad por defecto de 32768, aunque este valor puede ser modificado para influir en la elección. Entre más baja la prioridad, más probabilidades tiene un dispositivo de ser escogido como raíz. Se utiliza la dirección MAC solamente como medio de desempate (en caso que todos los *switches* tengan la misma prioridad).

Para ilustrar el proceso de elección se introduce la siguiente topología, donde todos los dispositivos siguen configurados con la prioridad por defecto.

Figura 194. **Elección del *switch* raíz**



Fuente: elaboración propia, empleando *Edraw Max*.

Los *switches* intercambian y comparan información utilizando BPDUs, tramas dentro de las cuales cada dispositivo coloca el identificador del *switch* que estos reconocen como raíz de la topología, así como su propio *bridge ID,* para que los demás sepan de dónde viene dicha comunicación.

Tabla XXI. **Estructura de un BPDU**

BPDU		
Root Bridge ID	32768 / 0000.0000.000A	← Identificador del *switch* raíz.
Sender Bridge ID	32768 / 0000.0000.000B	← Identificador del *switch* que envía el BPDU.
……	…….	

Fuente: elaboración propia.

Cuando los *switches* se inicializan, cada uno de ellos se considera asimismo el *switch* raíz de la topología y lo anuncia a los dispositivos vecinos al utilizar su propia dirección MAC, tanto en el campo que indica la dirección de origen de la comunicación como en el campo reservado al raíz.

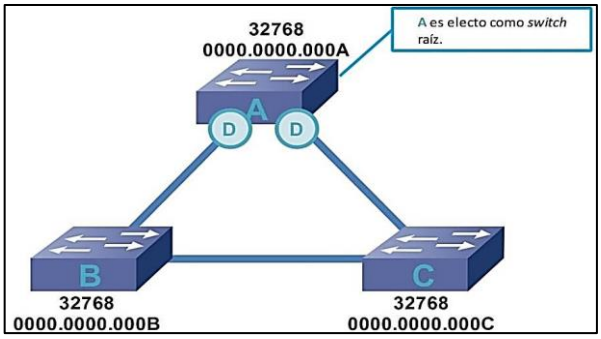
Si durante el intercambio de información el *switch* recibe un BPDU de un dispositivo con un b*ridge ID* menor al suyo, este lo reconoce como el nuevo *switch* raíz de la topología y actualiza la información enviada a los vecinos.

Este proceso se repite hasta que todos los participantes reconocen a un único *switch* raíz.

En el caso presentado, al tener la misma prioridad el *bridge ID* más bajo será determinado por la dirección MAC más baja, razón por la cual el *switch* A será el *switch* raíz de la topología.

Al concluir la elección, cada dispositivo debe determinar el rol que será asignado a cada uno de sus puertos, con excepción del *switch* raíz, ya que todos sus puertos serán puertos designados.

Figura 195. ***Switch* "A" es electo como *switch* raíz y como resultado todos sus puertos son puertos designados (D)**



Fuente: elaboración propia, empleando *Edraw Max*.

Para elegir los roles de los puertos se deben utilizar varios parámetros comparados en secuencia, deteniéndose en el primero de ellos para que no resulte un empate y siempre debe haber una preferencia por los valores más bajos.

El orden en que se comparan dichos parámetros es el siguiente:

- Costo de la ruta hacia el *switch* raíz (*Root Path Cost*).
- Identificador del *switch* que envía el BPDU (*Sender Bridge ID*).
- Identificador del puerto que envía el BPDU (*Sender Port ID*): compuesto por la prioridad del puerto (un valor numérico de 4 bits) y el número de la interface. Al modificar la prioridad de un puerto podemos influir en la elección del puerto raíz, toda vez que llegue el proceso a esta instancia. Este campo también está incluido en el BPDU.

- Identificador del puerto que recibe el BPDU (*Receiver Port ID*): compuesto de la misma forma que el campo anterior, con la diferencia que este es local al dispositivo, por lo que no existe un campo correspondiente en las BPDU.

El *root path cost* es un campo incluido dentro de las BPDU cuyo valor (llamado costo) se ve incrementado cada vez que entra por una interface. Tiene como objetivo determinar las mejores rutas hacia el *switch* raíz.

El costo relacionado con cada una de las interfaces es llamado costo de la ruta (*path cost*) y es determinado por la velocidad de cada una de estas, aunque su valor puede modificarse dentro de cada interfaz para influir en el rol que le será dado.

Al no existir valores recomendados para los costos que debían asignarse a las interfaces, Cisco utilizó la siguiente fórmula en sus primeras implementaciones de STP.

Figura 196. **Fórmula original empleada por Cisco para determinar el costo de cada interfaz con base en su velocidad**

$$Costo = \frac{1\ Gigabit/s}{ancho\ de\ banda}$$

Fuente: elaboración propia.

Sin embargo, debido al gran aumento en la velocidad de las interfaces y al hecho que la fórmula presentada no funcionaba adecuadamente para enlaces

más rápidos de 1 gigabit/s, se hizo una revisión del estándar (802.1D-1998) utilizando una escala no lineal, con otra fórmula no especificada en el mismo y que recomienda la utilización de los siguientes valores.

Tabla XXII. **Costos recomendados por la revisión del estándar original (802.1D-1998)**

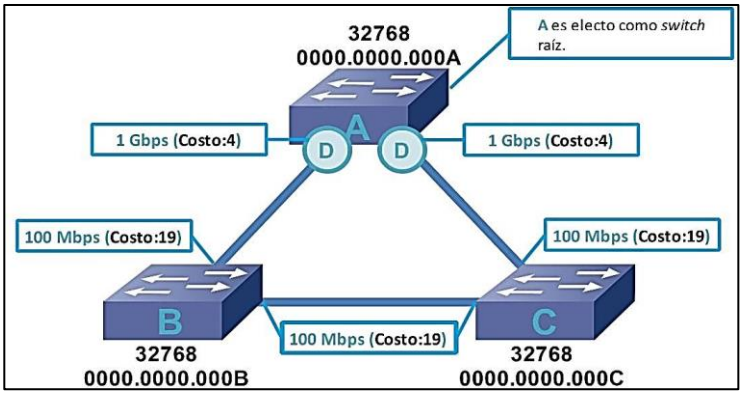
802.1D-1998		
Velocidad del puerto		**Costo**
Ethernet	10 Mbps	100
FastEthernet	100 Mbps	19
GigabitEthernet	1 Gbps	4
10-GigabitEthernet	10 Gbps	2

Fuente: elaboración propia.

Cuando un BPDU es generado por el *switch* raíz inicia con un *root path cost* de cero. Cuando dicho BPDU sea recibido por la interfaz de otro *switch,* este incrementará ese campo utilizando el costo asociado a dicha interfaz (*path cost*) antes de reenviarlo hacia otros dispositivos que luego utilizarán el valor total del mismo para determinar las rutas con menor costo y así elegir sus respectivos puertos raíces.

Como ejemplo se presenta la topología anterior, con las velocidades asociadas a sus respectivos puertos.

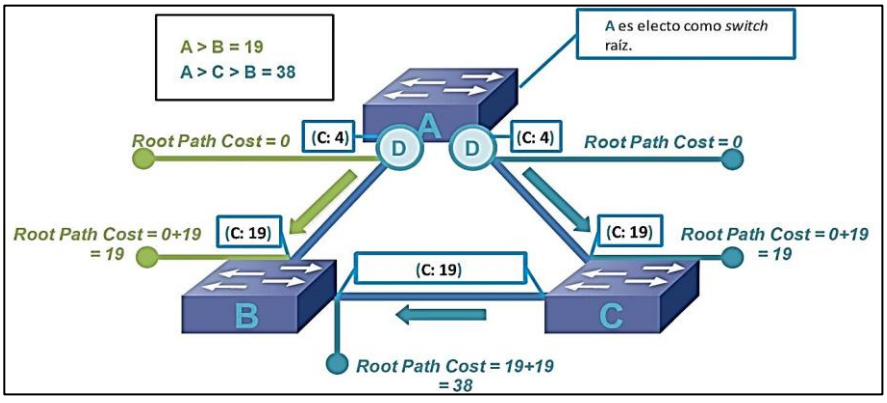
Figura 197. **Elección de los puertos raíces. Interfaces y costos asociados**



Fuente: elaboración propia, empleando *Edraw Max*.

En este caso, el *switch* raíz generará un BPDU con un *root path cost* inicial de cero, mismo que se incrementará con el costo asociado a cada interfaz por donde acceda (ver figura 198).

Figura 198. **Cálculo del *root path cost***



Fuente: elaboración propia, empleando *Edraw Max*.

El *switch* B recibirá dos BPDUs provenientes del *switch* raíz, una a través de su interfaz directamente conectada con un costo de 19 y otra que viene desde el *switch* C con un costo de 38 y seleccionará como puerto raíz aquella interface por donde ingresó el BPDU con la mejor ruta.

El costo está asociado con la velocidad de cada interface y no con cualquier velocidad negociada en el segmento.

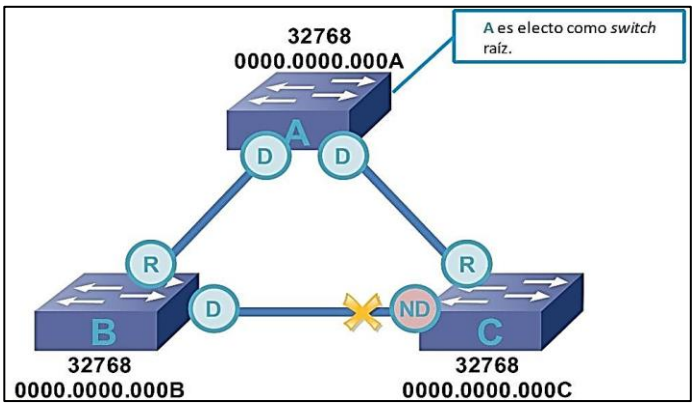
Después de hacer un análisis similar con el *switch* C, se establecen los puertos raíces y se determina que el enlace redundante es el que conecta el referido dispositivo con el *switch* B.

Una vez seleccionados los puertos raíces, se procede a seleccionar los puertos designados, uno por cada segmento.

En el caso del enlace redundante tanto *switch* B como *switch* C reciben BPDUs con el mismo costo hacia el *switch* raíz, por lo que se utiliza el siguiente parámetro en la lista: el identificador del *switch* que envía el BPDU (*Sender Bridge ID*).

Como B posee un *bridge ID* más bajo que el *switch* C sus puertos serán preferidos sobre los de este último si se llega a esta instancia, por lo que el puerto asociado con el primero ocupará la función de puerto designado, mientras que el extremo asociado con C será un puerto no designado.

Figura 199. **Convergencia de Spanning Tree**

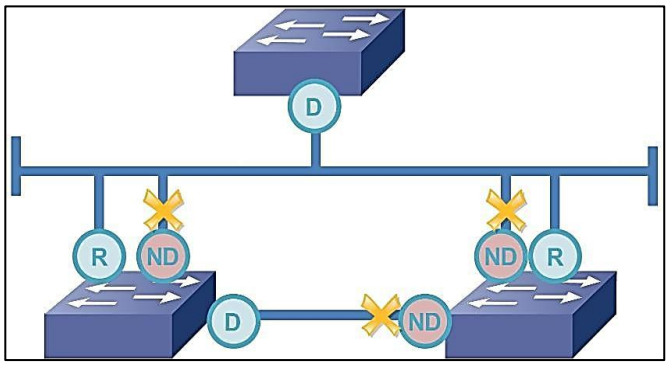


Fuente: elaboración propia, empleando *Edraw Max*.

Los dos últimos parámetros de la lista (*sender/receiver port ID*) no son utilizados en el ejemplo anterior, esto no quiere decir que no existan situaciones en donde deban utilizarse.

El siguiente caso es un ejemplo (donde deben utilizarse todos los parámetros presentados) que evidencia la versatilidad de STP.

Figura 200. **Versatilidad de *spanning tree***



Fuente: elaboración propia, empleando *Edraw Max*.

3.15.1.5. Portfast

Una de las debilidades de *spanning tree* consiste en que este protocolo no ofrece la opción para que un puerto pueda hacer distinción directa entre dispositivos, en otras palabras, STP funciona de la misma manera independientemente del dispositivo al que sea conectado.

Este comportamiento, inofensivo en un principio, empieza a generar problemas cuando las redes se hacen más grandes y los elementos que las componen más modernos. Ha llegado a ser un caso especialmente difícil, el de los dispositivos finales conectados a la red a través de puertos sujetos a STP.

Los dispositivos (computadoras, impresoras, entre otros) están destinados a satisfacer las necesidades de los usuarios, por lo que entran y salen de la red a medida que son necesarios. Si los puertos asociados a los mismos forman parte del proceso STP, cada vez que un *host* cambie de estado (ej.: una computadora es apagada o encendida), se generará una notificación de cambio de topología (TCN BPDU) que provocará que todos los *switches* renueven su base de datos de direcciones MAC en un tiempo más corto de lo usual (a pesar de no existir un cambio significativo) provocando inestabilidad.

Además, a pesar que la conexión con un solo *host* no puede introducir un bucle en la topología, este debe esperar de 30 a 50 segundos para incorporarse a la red, retraso que no es aceptable actualmente.

En vista de estos problemas, desactivar *spanning tree* en los puertos destinados para estos dispositivos puede parecer tentador, sin embargo, debido a la gran probabilidad de que los usuarios introduzcan bucles de manera accidental en la topología STP jamás debe desactivarse.

Ante esta situación, Cisco creó una mejora propietaria llamada *portfast* para mitigar estos problemas.

Un puerto configurado con *portfast* es un puerto que sigue ejecutando STP (sigue enviando BPDUs) con la diferencia que un cambio en su estado no genera un TCN (por lo que es más estable) y que su estado inicial es transmitiendo (*forwarding*) en vez de bloqueando (*blocking*).

Esto permite una red más estable y que los dispositivos finales se incorporen a la red inmediatamente. Como desventaja se puede mencionar la posibilidad de la formación de bucles temporales en caso que otros equipos como *hubs, switches, access points,* entre otros, sean conectados a puertos configurados con esta característica, debido a que cuando se inicializan pasan a transmitir información inmediatamente.

Los bucles aludidos son de carácter temporal debido a que al recibir un BPDU los puertos configurados con *portfast* deshabilitan esta característica.

3.15.2. *Rapid spanning tree* (RSTP)

Seguido de una serie de mejoras propietarias al estándar original, finalmente emergió en el 2001, un nuevo estándar abierto llamado 802.1W. Una implementación más moderna de STP con un tiempo de convergencia mucho menor al del original, por lo que recibió el nombre de *rapid spanning tree Protocol* (RSTP).

Esta nueva implementación es mucho más proactiva que la tradicional, ya que permite a los *switches* negociar activamente con sus vecinos en lugar de esperar pasivamente.

RSTP es retrocompatible con el protocolo original, por consiguiente pueden trabajar juntos en una red (aunque se perderían las ventajas de RSTP), llegando en ciertos tipos de enlace a comportarse de la misma manera.

3.15.2.1. *Bridge protocol data units* (BPDUs)

En STP se utilizaban dos tipos de BPDUs: el de configuración y el de notificación de cambio de topología. Los BPDU de configuración eran generados exclusivamente por el *switch* raíz para ser reenviados por todos los demás dispositivos. Asimismo, el *switch* raíz era el único capaz de indicar a los demás *switches* que debían renovar su base de datos de direcciones MAC, para adaptarse a un cambio en la red.

En RSTP, por otra parte, se utiliza un solo BPDU para cumplir las dos funciones mencionadas anteriormente; permite que se generen en todos los *switches* y así los cambios son asimilados más rápidamente. Además sirve para la detección de fallos (si se pierde 3 BPDUs del vecino esto significa que este ha fallado). Con un número de versión de 2 para distinguirlo de otras implementaciones (STP tradicional utiliza el número 0), posee la misma estructura que el original, para hacerlo compatible con versiones anteriores. Emplea campos antes no utilizados para transmitir nueva información y permite negociaciones entre dispositivos vecinos.

3.15.2.2. Estados y roles de los puertos en *Rapid Spanning Tree*

Para lograr una convergencia más rápida, RSTP separa completamente el estado de los puertos de los roles que ocuparán los mismos. Es asignado

primero el rol (ahora incluido dentro del BPDU), para luego determinar su estado.

RSTP reconoce los siguientes roles:

- Puerto raíz (*root port*): con un funcionamiento idéntico al estándar original.

- Puerto designado (*designated port*): con un funcionamiento idéntico al estándar original.

- Puerto elternativo (*alternate port*): puerto de respaldo que provee una ruta alternativa al *switch* raíz, que será utilizada en caso que la ruta a través del puerto raíz deje de estar disponible.

- Puerto de respaldo (*backup port*): puerto de respaldo que provee un enlace alternativo dentro de un mismo segmento de red, para ser utilizado en caso que el puerto designado falle.

Utiliza los siguientes estados:

- Descartando (*discarding*): en este estado el puerto no es capaz de enviar o recibir información ni de aprender direcciones MAC. Recibe y procesa BPDU y dependiendo del rol asignado, también podría enviarlas. Esta fase combina los estados *disabled, blocking* y *listening* del estándar original.

- Aprendiendo (*learning*): en esta fase no se puede enviar o recibir información, pero sí se puede aprender direcciones MAC.

- Transmitiendo (*forwarding*): el puerto es completamente operacional.

La combinación de rol y estado para un puerto que acaba de encenderse es *designated discarding.*

Estos roles son utilizados dentro del Cisco IOS para mostrar la función de los puertos, inclusive en las implementaciones de *Spanning Tree* tradicional. Otro punto que es fuente de confusión.

3.15.2.3. Tipos de enlace

RSTP introduce el concepto de los tipos de enlace, de los cuales dependerá su operación.

- Enlace punto a punto (*point-to-point link*): enlace que conecta directamente a dos dispositivos ejecutando RSTP y que pueden aprovechar todas sus mejoras.

- Enlace compartido (*shared link*): enlace hacia un segmento compartido, donde RSTP revertirá su funcionamiento al del estándar original.

3.15.2.4. Elección del *switch* raíz y el rol de cada puerto en *Rapid Spanning Tree*

La elección del *switch* y el puerto raíz siguen exactamente los mismos criterios y son considerados en la misma secuencia que en el estándar original.

La única diferencia radica en el costo de la ruta (*path cost*): costo asignado a cada interfaz con base en su velocidad, ya que al ser un estándar más

moderno presenta nuevas recomendaciones acerca del valor que debe dársele a los mismos. Utiliza la siguiente fórmula.

Figura 201. **Fórmula utilizada por el estándar 802.1 W para determinar el costo asociado a una interfaz a partir de su velocidad**

$$Costo = \frac{20\ Terabit/_s}{ancho\ de\ banda}$$

Fuente: elaboración propia.

Comparativa entre los valores utilizados en el estándar original y el nuevo.

Tabla XXIII. **Comparación de los costos utilizados por STP y RSTP**

		802.1D-1998	802.1W
Velocidad del puerto		**Costo**	**Costo**
Ethernet	10 Mbps	100	2 000 000
FastEthernet	100 Mbps	19	200 000
GigabitEthernet	1 Gbps	4	20 000
10-GigabitEthernet	10 Gbps	2	2 000

Fuente: elaboración propia.

Para determinar los roles de cada uno de los puertos, se realiza una negociación entre dispositivos vecinos. La red se converge de una manera progresiva.

La explicación de este proceso se encuentra fuera de los límites propuestos para este trabajo.

3.15.2.5. *Edge*

Es la solución estandarizada para el problema de los puertos que deben ser conectados a dispositivos finales. Presenta la misma funcionalidad que *portfast.*

3.15.3. Relación entre VLANs y *spanning tree*

Existe una profunda relación entre las VLAN existentes en una topología y el funcionamiento de *spanning tree.* Si bien este último fue creado primero, con el posterior advenimiento de las VLAN y su masiva adopción, estas dos tecnologías se volvieron inseparables, al punto que hoy la funcionalidad de ambas está definida en un solo estándar (802.1Q).

De esta manera, las implementaciones de *spanning tree,* tanto de STP tradicional como de *Rapid* STP, son clasificadas en función de su interacción con las VLAN. Existen estándares abiertos como *Mono Spanning Tree* (MST) y *Multiple Spanning Tree Protocol* (MSTP) y soluciones propietarias como *Per VLAN Spanning Tree Protocol Plus* (PVSTP+) y *Rapid* PVST+ (*RPVST+*) en el caso de Cisco, las que serán examinadas más adelante.

3.15.3.1. Ajustes

Para acomodar el creciente número de puertos y las VLAN disponibles dentro de cada dispositivo, fue necesario realizar ciertos ajustes en las disposiciones originales; uno de ellos es llamado reducción de direcciones MAC (*MAC address reduction*).

Como se mencionara, cada *switch* posee un identificador único dentro la topología llamado *bridge ID,* un campo de 64 bits que estaba compuesto por la prioridad asignada al dispositivo (16 bits) y su dirección MAC (48 bits). Con la introducción de las VLAN se debía asignar un *bridge* ID único a cada una de ellas, razón por la cual los *switches* eran fabricados junto con un grupo de direcciones MAC únicas a cada uno, para poder disponer de una por cada VLAN que pudieran llegar a configurarse. Al aumentar el número de VLAN potenciales, se hizo evidente que ese modelo no era sostenible, ya que requería que cada dispositivo tuviera a su disposición miles de direcciones MAC irrepetibles.

Se hizo entonces, indispensable encontrar la forma de reducir la cantidad de direcciones MAC necesarias (de ahí el nombre), al mismo tiempo ofrecer un *bridge* ID único a cada VLAN y mantener su tamaño original.

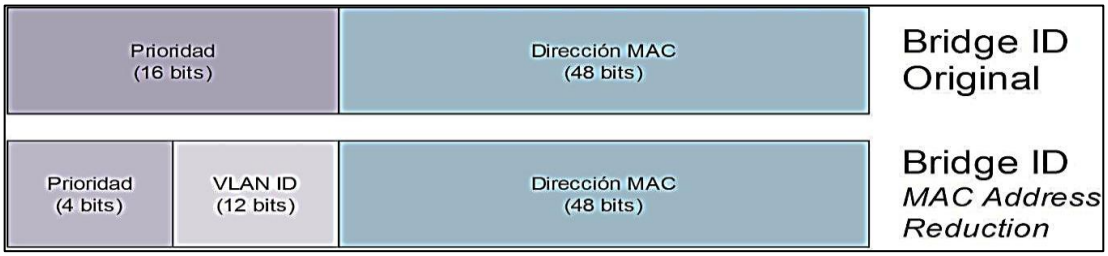
La solución fue utilizar la misma dirección MAC para todas las VLAN. Se introdujo el identificador de cada VLAN a manera que el *bridge* ID sea único para cada una de ellas.

A manera de agregar esta nueva información y mantener el tamaño del campo original dentro del BPDU, se tomaron 12 bits que pertenecieron originalmente a la prioridad, razón por la que esta solo puede configurarse en incrementos de 4096 (por el corrimiento de los 12 bits) en un rango de 0 a 61440.

Debido a esta modificación, la prioridad de un *switch* se presenta ante los demás dispositivos como la sumatoria de la parte configurable de la misma y el identificador de la VLAN. A manera de ejemplo, si se tiene el valor asignado por

defecto (32768) y la VLAN 10, la adición de estos valores resultará en una prioridad de 32778.

Tabla XXIV. **Comparación entre el *bridge* ID original y el que implementa MAC *address reduction***



Fuente: elaboración propia.

Un proceso similar fue llevado a cabo para aumentar el número disponible para identificar interfaces, al modificar el campo perteneciente al identificador del puerto (*port ID*) de donde se tomaron 4 bits (de los 8 bits originales), por lo que solo admite incrementos de 16.

Otro ajuste del que hay que hablar está a discreción del usuario y relacionado con el costo de la ruta (*path cost*) asignado a cada puerto. Como fue referido previamente, dicho costo es asignado con base en dos fórmulas diferentes, la primera de ellas con un tamaño de 16 bits utilizada con el estándar 802.1D y conocida como el modo corto (*short mode*) y la segunda con un tamaño de 32 bits definida en el estándar 802.1w conocida como el modo largo (*long mode*). Una comparación entre las dos escalas es reproducida nuevamente.

Tabla XXV. **Comparación entre las escalas de 16 y 32 bits de largo**

		802.1D-1998	802.1W
Velocidad del puerto		**Costo**	**Costo**
FastEthernet	100 Mbps	19	200 000
10-GigabitEthernet	10 Gbps	2	2 000

Fuente: elaboración propia.

Las implementaciones citadas tanto PVST+ como RPVST+ utilizan por defecto el *short mode,* mientras que MSTP siempre utiliza el *long mode.* Este comportamiento puede cambiarse, de ser necesario para PVST+ y RPVST+ con el comando que se muestra en la figura 202.

Figura 202. **Comando para PVST+ y RPVST**

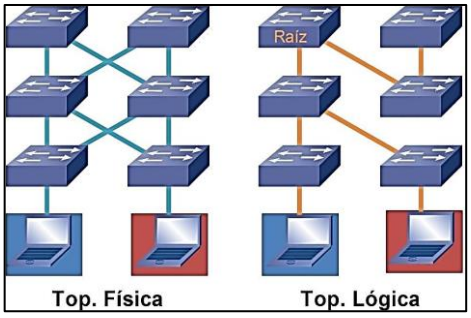
```
switch(config)# spanning-tree pathcost method long
```

Fuente: elaboración propia.

3.15.3.2. *Mono Spanning tree* (MST)

Definida en el estándar 802.1Q, establece una sola instancia de *Spanning Tree* para ser compartida por todas las VLAN. Esto implica que es posible solo una topología lógica, por lo que ciertos enlaces y equipos redundantes nunca serán utilizados, esta es su principal desventaja.

Figura 203. **Con MST todas las VLAN comparten una sola instancia de *spanning tree***

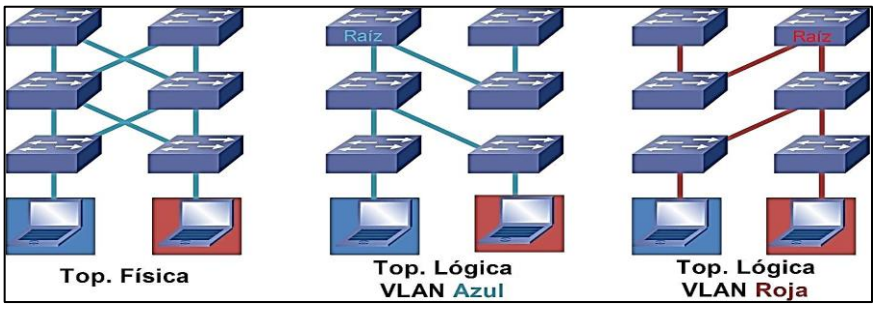


Fuente: elaboración propia, empleando *Edraw Max*.

3.15.3.3. Per VLAN spanning tree plus (PVST+)

Es la implementación del algoritmo original de *spanning tree* que se ejecuta por defecto en la mayoría de las plataformas Cisco. Admite ejecutar una instancia de STP por cada VLAN, lo que permite una distribución artificial del tráfico de la red (provista la configuración correcta), para aprovechar de mejor manera los enlaces y dispositivos disponibles.

Figura 204. **PVST+ ejecuta una instancia de STP por cada VLAN**



Fuente: elaboración propia, empleando *Edraw Max*.

Al tener varias instancias de STP, la mayoría de instrucciones relacionadas requiere ingresar las VLAN que serán afectadas, como se muestra a continuación:

- Para configurar la prioridad de una instancia de STP y afectar la elección del *switch* raíz es posible emplear la instrucciónd escrita en la figura 205.

Figura 205. **Instrucción para configuración de la prioridad**

```
Switch(config)# spanning-tree vlan ?
  WORD  vlan range, example: 1,3-5,7,9-11

Switch(config)# spanning-tree vlan 1 priority 4096
```

Fuente: elaboración propia.

Para modificar el *path cost* asignado a un puerto, puede utilizarse el comando descrito en la figura 206.

Figura 206. **Comando para modificar el *path cost* asignado a un puerto**

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# spanning-tree vlan 1 cost ?

  <1-65535>  Change an interface's per VLAN spanning tree path cost
```

Fuente: elaboración propia.

Para cambiar la prioridad de un puerto se usa la configuración que se muestra en la figura 207.

Figura 207. **Configuración para cambiar prioridad de un puerto**

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# spanning-tree vlan 1 port-priority ?
  <0-240>  port priority in increments of 16

Switch(config-if)# spanning-tree vlan 1 port-priority 128
```

Fuente: elaboración propia.

Para habilitar *portfast* puede hacerse de manera global en todos los puertos configurados en modo de acceso, con la instrucción que se muestra en la figura 208

Figura 208. **Instrucción para habilitar *portfast***

```
Switch(config)# spanning-tree portfast default
```

Fuente: elaboración propia.

- Individualmente en cada interfaz, en cuyo caso desplegará una advertencia pidiendo precaución al usuario (ver figura 209).

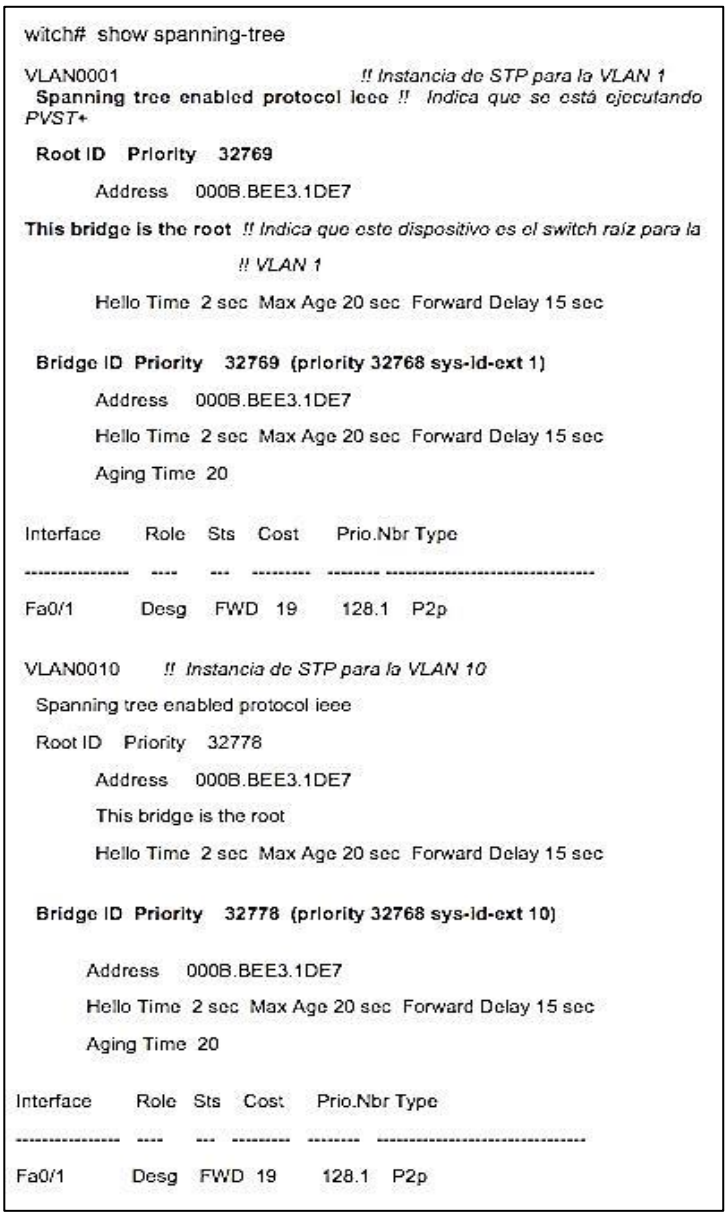
Figura 209. **Advertencia de precaución**

```
Switch(config)# interface fastethernet 0/1
Switch(config-if)# spanning-tree portfast

%Warning: portfast should only be enabled on ports connected to a single
host. Connecting hubs, concentrators, switches, bridges, etc... to this
interface  when portfast is enabled, can cause temporary bridging loops.
Use with CAUTION

%Portfast has been configured on FastEthernet0/1 but will only
have effect when the interface is in a non-trunking mode.
```

Fuente: elaboración propia.

Para mostrar el identificador tanto del *switch* raíz como del dispositivo, prioridades, roles de los puertos, estado de las interfaces y el modo que se está utilizando como base (STP tradicional o RSTP), ver figura 210.

Figura 210. **Uso del comando *show spanning-tree***



Fuente: elaboración propia.

3.15.3.4. *Rapid Per* VLAN *spanning tree plus* (RPVST+)

Es la implementación de Cisco del algoritmo de *rapid spanning tree,* presenta una instancia de STP por cada VLAN y es el protocolo activo por defecto en las plataformas más modernas. Con un tiempo de convergencia menor que PVST+ puede habilitarse, en los dispositivos que no lo estén ejecutando, con el comando que se muestra en la figura 211.

Figura 211. **Habilitación de RPUST+**

```
Switch(config)# spanning-tree mode rapid-pvst
```

Fuente: elaboración propia.

Al ejecutar nuevamente la instrucción *show spanning tree* es posible corroborar que este es el protocolo que se está utilizando.

Figura 212. **Instrucción *show spanning tree***

```
Switch# show spanning-tree

VLAN0001
  Spanning tree enabled protocol rstp
  Root ID    Priority    32769
             Address     000B.BEE3.1DE7
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     000B.BEE3.1DE7
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20

Interface        Role      Sts      Cost      Prio.Nbr  Type
---------------- ----      ---      ---------  --------  --------------------------------
Fa0/1            Desg      FWD      19         128.1     P2p
```

Fuente: elaboración propia.

Las instrucciones para configurar RPVST+ son las mismas utilizadas por su antecesor y los puertos *edge* son configurados usando la misma instrucción que los puertos *portfast.*

3.15.3.5. Multiple Spanning Tree Protocol (MSTP)

Los dispositivos que ejecutan una instancia de *Spanning Tree* por cada VLAN pueden experimentar problemas con el consumo de recursos, a medida que el número de estas últimas aumentan.

Para limitar el consumo de recursos y mantener las ventajas introducidas por protocolos como PVST+, se creó *Multiple Spanning Tree Protocol* (MSTP). Definida en un inicio en el estándar 802.1s, es en la actualidad, también parte del estándar 802.1Q.

MST permite agrupar un número arbitrario de VLAN dentro de una sola instancia de este protocolo, el cual es una extensión de *Rapid Spanning Tree.* Una discusión detallada de MST se encuentra fuera de los límites fijados para el desarrollo de este trabajo.

3.15.4. Modelo jerárquico de tres capas de Cisco

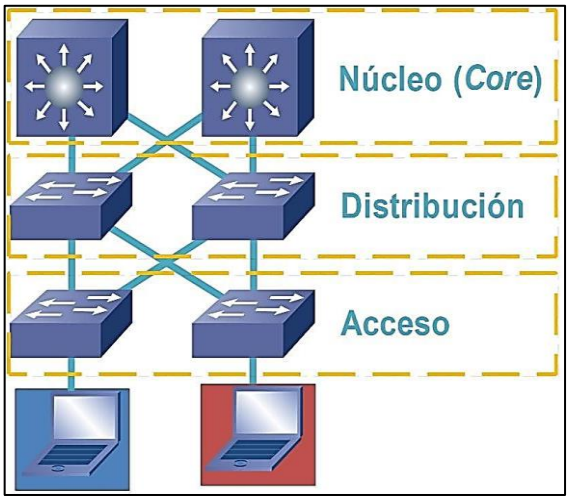
El diseño de la red es crucial para su correcto funcionamiento. Es un proceso complejo es necesario y documentar requerimientos, suposiciones y otros aspectos; además de planear el crecimiento de la misma.

Para ayudar con esta tarea existen varios marcos de referencia. Uno de los más elementales es el modelo jerárquico de tres capas de Cisco.

Como su nombre lo indica, este modelo tiene 3 capas, cada una con las siguientes características:

- Capa de acceso (*access layer*): es donde se conectan los dispositivos finales a la red. Deben ser baratos y tener disponible una gran cantidad de puertos.
- Capa de distribución (*distribution layer*): en esta se agrupan todos los *switches* de la capa de acceso, por lo que tiene que ser capaz de manejar el tráfico combinado de todos.
- Capa de núcleo (*core layer*): el núcleo de la topología encargada de transmitir la información de la manera más rápida posible entre distintas partes de la infraestructura local y hacia otras redes.

Figura 213. **Modelo jerárquico de tres capas**



Fuente: elaboración propia, empleando *Edraw Max*.

3.15.5. Recomendaciones al incluir *Spanning Tree* dentro del diseño de una red

- Si se trata de una nueva instalación, considerará otras alternativas a STP para lograr la redundancia deseada, ya que es un protocolo muy antiguo. De no ser esto factible debe tratar de reducir al máximo el dominio de cada instancia de *Spanning Tree,* inclusive tratar de limitarlo a la capa de acceso del diseño.

- Se desaconseja el uso de VLAN que abarquen toda la infraestructura. Es importante utilizar en su lugar VLAN más pequeñas geográficamente, limitadas por ejemplo, a un edificio en particular. Esta medida segmenta los dominios de *broadcast* y hace menos posible y más fácil de localizar problemas dentro de la topología.

- La elección de *switch* raíz nunca debe dejarse al azar. Debe configurarse un *switch* raíz, así como un *switch* raíz de respaldo y deben ser colocados lo más cerca posible, o directamente en el núcleo (*Core*) de la topología, para evitar que el tráfico atraviese por rutas subóptimas.

- La documentación de la red es esencial, es necesario separar la topología física de la lógica. En caso de una tormenta de *broadcast* se puede recurrir a la misma para saber dónde están los enlaces redundantes.

- *Spanning Tree* nunca debe ser desactivado para evitar la introducción accidental (o no) de bucles dentro de la red.

3.15.6. Macroinstrucciones

Las macroinstrucciones, mejor conocidas como macros, consisten en una serie de instrucciones que son almacenadas para ser ejecutadas en una sola llamada.

Relativas a *Spanning Tree* y al Cisco IOS se pueden encontrar predefinidas las siguientes.

- *Switchport host:* configura el puerto en modo de acceso y habilita *Portfast.*

Figura 214. ***Switchport host***

```
Switch (config)# interface fastethernet 0/1
Switch (config-if-range)# switchport host
```

Fuente: elaboración propia.

- *Spanning Tree* VLAN (rango de VLAN que serán afectadas) root (*primary / secondary*). Están destinados para configurar automáticamente el *switch* raíz y el *switch* raíz de respaldo, aunque no hay garantía que el resultado de la operación sea exitoso. Este macro solo puede ejecutarse una vez, por lo que no impide la reelección del *switch* raíz en caso se baje la prioridad de otro dispositivo.

Figura 215. **Spanning Tree VLAN *root primary/secondary***

```
Switch(config)# spanning-tree vlan 10 root primary
Switch(config)# spanning-tree vlan 20 root secondary
```

Fuente: elaboración propia.

Cuando se tome la decisión de elegir el *switch* raíz y el de respaldo dentro de una topología, se recomienda que se configure la prioridad manualmente en lugar de usar los macros disponibles.

3.15.7. Alternativas a *Spanning tree*

Actualmente, debido a los altos niveles de competitividad, la existencia de enlaces no utilizados es inaceptable. Por esta razón, recientemente se han desarrollado protocolos que son capaces de proporcionar redundancia y al mismo tiempo utilizan todas las conexiones disponibles. Algunos de ellos son:

- Transparent Interconnection of Lots of Links (TRILL): creado por Radia Perlman, es un estándar de la Internet Engineering Task Force (IETF).

- Shortest Path Bridging (SPB): definido en el estándar 802.1aq de la IEEE.

Al momento de realizar el presente estudio, estos protocolos de reciente creación no son soportados por todos los fabricantes.

3.16. *Access control lists* (ACLs)

Las listas de control de acceso, mejor conocidas como *access control lists* (ACLs), son herramientas que permiten identificar o marcar un flujo de datos acorde a ciertos criterios para realizar una operación especial sobre él, por este motivo son utilizadas para control de acceso, calidad de servicio, enrutamiento basado en políticas, traducción de direcciones y otros.

Consisten en una lista de sentencias que pueden permitir (*permit*) o denegar (*deny*) marcando o ignorando el tráfico que se les indique, aunque su efecto siempre dependerá de cómo y dónde estas sean aplicadas.

Las listas son examinadas sentencia por sentencia, en orden secuencial, deteniéndose la operación al hallar la primera coincidencia. Por tal razón las ACLs deben diseñarse con cuidado, colocando las sentencias más específicas al principio para que estas puedan llegar a ser evaluadas.

La configuración de estas listas está separada de su implementación, aunque no hay ningún mecanismo que impida aplicar ACLs inexistentes, en cuyo caso se permitirá (o marcará) todo el tráfico que sea comparado con ella, hasta que esta sea creada y las sentencias sean agregadas.

No obstante, se recomienda diseñar y configurar las listas de control de acceso antes de su implementación, debido al comportamiento que estas presentan.

Una lista vacía (o inexistente) permitirá todo el tráfico de la manera descrita anteriormente, sin embargo, al ingresar la primera sentencia dentro de la misma se creará automáticamente otra al final de la lista, implícita e invisible, encargada de desestimar toda aquella información que no haya encontrado una coincidencia en las sentencias previas.

Dicha sentencia es comúnmente conocida como “denegar todo” y siempre se encuentra al final de toda lista de control, razón por la cual siempre debe incluirse por lo menos una sentencia “permitir” cuando se pretende utilizar ACLs para regular tráfico.

Existen varios tipos de listas de control de acceso, entre ellas se pueden citar:

- Estándares
- Extendidas
- Reflexivas
- Basadas en el tiempo

Cuando las ACLs hicieron su aparición se clasificaron dentro de diferentes rangos acordes a su propósito. Por ejemplo, las listas estándares utilizaban los intervalos <1-99> y <1300-1999> y las extendidas <100-199> y <2000-2699>. Más adelante se les asignó un nombre, situación que contribuyó a que su uso fuera más conveniente.

Al crear una lista de control se recomienda utilizar y apegarse a una convención, agregar las observaciones pertinentes y recordar al implementarlas que los nombres son sensibles a minúsculas y mayúsculas (*Case sensitive*).

En este trabajo solo se tratarán las listas estándares y extendidas y su aplicación para filtrar tráfico en una interfaz. Además se favorecerá el procedimiento más moderno para crearlas.

3.16.1. Listas de control de acceso estándares

Son las listas de control más simples, utilizan como único parámetro de comparación el origen del tráfico empleando *wildcard masks* (introducidas en la sección de OSPF), para seleccionar rangos específicos de direcciones e impactan muy poco al procesador.

Para proveer un ejemplo. Se creará una lista de control de acceso estándar, para proponer un ejemplo, destinada a identificar el tráfico proveniente de la red 192.168.10.0/24 y que ignore todo lo demás, misma que será nombrada como “Técnicos”.

Figura 216. **Lista de control de acceso estándar**

```
Router(config)# ip access-list standard Tecnicos
Router(config-std-nacl)# remark [> Esta lista identifica a los tecnicos. <]
Router(config-std-nacl)# permit 192.168.10.0 ?
A.B.C.D  Wildcard bits
<cr>

Router(config-std-nacl)# permit 192.168.10.0 0.0.0.255
Router(config-std-nacl)# deny any
```

Fuente: elaboración propia.

Nótese el uso del comando *remark* y la palabra clave *any. Remark* permite agregar una observación a la lista de control, mientras que *any* es un atajo para seleccionar todas las direcciones posibles y es el equivalente de utilizar la instrucción *deny 0.0.0.0 255.255.255.255.*

Además, se incluyó la sentencia *deny any,* aunque no era necesario, ya que está implícita al final de toda lista, debido a que facilita la resolución de problemas y permite ver la cantidad de paquetes que han llegado a esta instancia al ser ahora visible, por lo que es una buena práctica.

3.16.2. Listas de control de acceso extendidas

Son mucho más granulares que las listas estándares, permiten seleccionar tanto el origen como el destino del tráfico, protocolos específicos y números de puerto.

A continuación se muestran los protocolos que pueden ser evaluados con una lista extendida. Durante el resto de esta discusión se tratará exclusivamente con TCP, UDP e IP. Donde la palabra clave “IP” abarca todos los protocolos disponibles en este tipo de lista.

Figura 217. **Protocolos que pueden ser evaluados con una lista extendida**

```
Router(config)# ip access-list extended EJEMPLO

Router(config-ext-nacl)# permit ?
 <0-255>  An IP protocol number
 ahp      Authentication Header Protocol
 eigrp    Cisco's EIGRP routing protocol
 esp      Encapsulation Security Payload
 gre      Cisco's GRE tunneling
 icmp     Internet Control Message Protocol
 igmp     Internet Gateway Message Protocol
 ip       Any Internet Protocol
 ipinip   IP in IP tunneling
 nos      KA9Q NOS compatible IP over IP tunneling
 ospf     OSPF routing protocol
 pcp      Payload Compression Protocol
 pim      Protocol Independent Multicast
 tcp      Transmission Control Protocol
 udp      User Datagram Protocol
```

Fuente: elaboración propia.

Para ejemplificar: supóngase que se pretende identificar el tráfico que se origina en el *host* con la dirección IP 192.168.1.1/24 con destino al servidor web

192.168.2.1/24 (escuchando en el puerto 80) e ignorar el resto de la transmisión, podría utilizarse una lista extendida como se indica en la figrua 217.

Figura 218. **Lista extendida**

```
Router(config)# ip access-list extended EJEMPLO2
Router(config-ext-nacl)# permit tcp host 192.168.1.1 host 192.168.1.2 eq 80
Router(config-ext-nacl)# deny ip any any
```

Fuente: elaboración propia.

En esta ocasión se ha utilizado la palabra clave "*host*", que identifica una dirección específica y es el equivalente a usar una *wildcard* 0.0.0.0. La última parte de la primera sentencia "eq 80" significa que se seleccionará el tráfico destinado al *host* 192.168.1.2, cuyo puerto de destino sea equivalente al puerto 80 y "*deny ip any any*" ignorará cualquier otro protocolo dirigido desde cualquier origen hacia cualquier destino.

Cuando se emplean listas de control de acceso extendidas, hay que tener cuidado de no seleccionar el puerto utilizado por la parte que origina el tráfico, en lugar de la lista a la que este puerto está destinado.

Al retomar el ejemplo anterior, si se hubiera ingresado por error en la siguiente sentencia, difícilmente se hubiera encontrado una coincidencia debido a que las transmisiones regresan al origen en un puerto generado aleatoriamente.

Figura 219. **Ejemplo de sentencia**

```
Router(config-ext-nacl)# permit tcp host 192.168.1.1 eq 80 host 192.168.1.2
```

Fuente: elaboración propia.

3.16.3. Listas de control de acceso aplicadas para regular tráfico en una interfaz

Las listas de control de acceso son capaces de regular el tráfico tanto en interfaces físicas o virtuales (líneas VTY). En estos casos la aplicación de las mismas es directa, las sentencias permitir (*permit*) dejarán pasar el tráfico mientras que las sentencias denegar (*deny*) lo descartarán.

Los comandos para aplicar dichas listas dependen del tipo de interfaz en donde se pretendan configurar. Debe indicarse además, si la lista será evaluada cuando los paquetes entran o salen de la interfaz.

Para aplicar una lista sobre una interfaz se utiliza el comando *ip access-group* como se puede apreciar en la muestra.

Figura 220. **Comando *ip access-group***

```
Router(config-if)# interface fastethernet 0/0
Router(config-if)# ip access-group ?
  <1-199>          IP access list (standard or extended)
  <1300-2699>   IP expanded access list (standard or extended)
  WORD             Access-list name

Router(config-if)# ip access-group NOMBRE_LISTA ?
  in     inbound packets
  out  outbound packets

Router(config-if)# ip access-group NOMBRE_LISTA in
```

Fuente: elaboración propia.

Para aplicar una lista sobre una línea VTY se utiliza el comando *access-class*. Es recomendable que esta siempre se evalúe cuando los paquetes están entrando (*in*) para evitar comportamiento errático. Las listas aplicadas sobre las líneas mencionadas son muy útiles para limitar el acceso remoto a los dispositivos, ya sea a través de *telnet* o SSH.

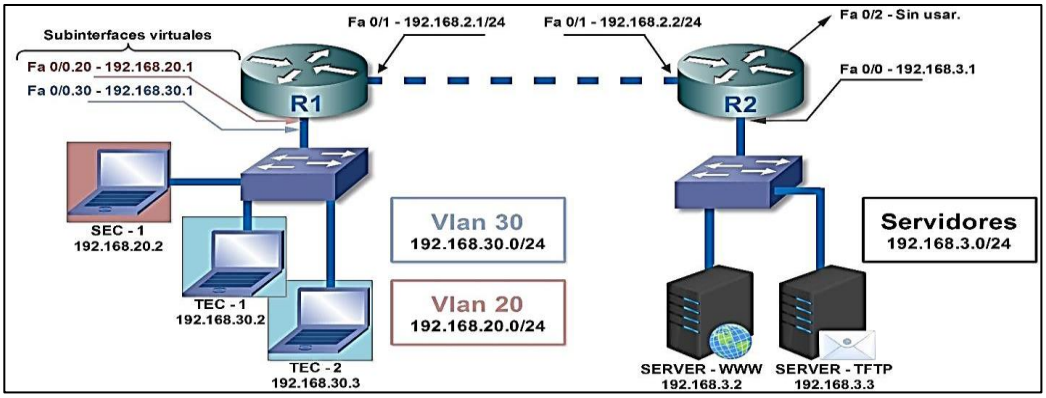
Figura 221. **Comando *access-class***

```
Router(config-if)# line vty 0 4
Router(config-line)# access-class ?
  <1-199>          IP access list
  <1300-2699>   IP expanded access list
  WORD             Access-list name

Router(config-line)# access-class NOMBRE_LISTA ?
  in   Filter incoming connections
  out  Filter outgoing connections

Router(config-line)# access-class NOMBRE_LISTA in
```

Fuente: elaboración propia.

La topología descrita en la figura 222 complementa el ejemplo; la configuración necesaria para establecer conexión de extremo a extremo ha sido ingresada previamente.

Figura 222. **Topología para mostrar la configuración y aplicación de ACL**



Fuente: elaboración propia, empleando *Edraw Max*.

Este ejercicio presenta tres redes: técnicos, secretarias y servidores. Los objetivos, que se enumeran a continuación, han sido determinados por su valor pedagógico y no reflejan necesariamente escenarios reales.

- Limitar el acceso a los *routers* (*telnet* o SSH) exclusivamente a la red de técnicos.

- Impedir que la red perteneciente a las secretarias pueda comunicarse con la red de servidores.

- La computadora del técnico 1 (TEC - 1) no podrá ingresar en el servidor web mientras que la computadora del técnico 2 (TEC - 2) no podrá ingresar al servidor TFTP. Todos los demás servicios serán permitidos.

Para cumplir con el primer objetivo se pueden seguir dos aproximaciones diferentes. Es posible denegar específicamente a la red de las “secretarias”, sin

embargo, es factible que en un futuro aparezca otra red que tampoco deba tener acceso a la configuración de los dispositivos (ej.: ventas), por esta razón, una mejor solución es permitir solamente a la red de “técnicos” y denegar a todas las demás.

Para este propósito se creará una lista llamada “permitirtecnicos” y será aplicada en las líneas VTY de ambos *routers.*

Figura 223. **Lista llamada “PermitirTecnicos”**

```
R1(config)# ip access-list standard PermitirTecnicos
R1(config-std-nacl)# remark [> Permite solo a los tecnicos a traves de SSH o telnet <]
R1(config-std-nacl)# permit 192.168.30.0 0.0.0.255
R1(config-std-nacl)# deny any
```

Fuente: elaboración propia.

Figura 224. **Aplicación en líneas VTY**

```
R1(config)#line vty 0 4
R1(config-line)# access-class PermitirTecnicos in
```

Fuente: elaboración propia.

Para ver la composición de las listas creadas, así como para verificar su funcionamiento se puede emplear el comando *show ip access-list,* que se muestra en la figura 224, después de que usuarios pertenecientes a ambas VLANs hayan tratado de establecer una conexión a través de *telnet.*

Figura 225. **Comando *show ip access-list***

```
R1# show ip access-lists
Standard IP access list PermitirTecnicos
10 permit 192.168.30.0 0.0.0.255 (2 match(es))
20 deny any (10 match(es))
```

Fuente: elaboración propia.

Cada sentencia está numerada para indicar el orden en que se ejecuta. Se utiliza un incremento de diez para que nuevas sentencias puedan ser agregadas fácilmente. Nótese también, que junto a cada una de ellas aparece el número de paquetes en los que se ha encontrado una coincidencia.

Para limitar el acceso en R2, basta con crear la lista nuevamente en este dispositivo y aplicarla a las líneas VTY de la misma manera. Las observaciones hechas a cada una (*remark*) aparecerán al examinarse la configuración del dispositivo.

Para llevar a cabo el segundo objetivo de este ejercicio, podrá utilizarse la siguiente lista de control.

Figura 226. **Lista de control**

```
Router(config)# ip access-list standard DenegarSecretarias
Router(config-std-nacl)# remark [> Deniega a las secretarias <]
Router(config-std-nacl)# deny 192.168.20.0 0.0.0.255
Router(config-std-nacl)# permit any
```

Fuente: elaboración propia.

Es necesario advertir la presencia de la sentencia "*permit any*" al final de la lista, ya que de otro modo, no solo las secretarias, sino que todo el tráfico

sería denegado en la interfaz donde llegará a aplicarse, debido a las razones explicadas anteriormente.

Una vez creada la lista, al menos de manera conceptual, es necesario decidir en qué *router,* en qué interface y en qué dirección, esta va a aplicarse.

Una posibilidad sería aplicar la lista en R2, en la interfaz *FastEthernet* 0/1 (Fa 0/1), cuando los paquetes estén entrando (*in*), lo que ciertamente cumpliría el propósito original. No obstante, si se llegara a implementar una nueva red en la interfaz *FastEthernet* 0/2, ahora sin utilizarse, esta también estaría negada a las secretarias a pesar de no estar incluida en el alcance original.

Lo explicado anteriormente constituye la principal desventaja de las listas estándares, ya que al utilizar solamente la dirección de origen del tráfico se corre el riesgo de impedir el acceso a partes de la red que no debían ser restringidas si se aplican en la interfaz incorrecta. Por tal motivo es una buena práctica configurar las listas de este tipo lo más cerca posible a su destino (para no restringir de más).

En este caso se aplicará la lista recién creada en R2 en la interfaz *FastEthernet* 0/0 cuando el tráfico está saliendo de dicha interface.

Figura 227. **Creación de la lista de control de aceso en R2**

```
R2(config)# ip access-list standard DenegarSecretarias
R2(config-std-nacl)# remark [> Deniega a las secretarias <]
R2(config-std-nacl)# deny 192.168.20.0 0.0.0.255
R2(config-std-nacl)# permit any
```

Fuente: elaboración propia.

Figura 228. **Aplicación de la lista creada en R2 en la interfaz FastEthernet 0/0**

```
R2(config)#interface fastEthernet 0/0
R2(config-if)#ip access-group DenegarSecretarias out
```

Fuente: elaboración propia.

Para concluir este ejercicio hay que impedir que la computadora del técnico 1 (192.168.30.2/24) tenga acceso al servidor web (192.168.3.2:80 (TCP)) y que la computadora del técnico 2 (192.168.30.3/24) tenga acceso al servidor TFTP (192.168.3.3:69 (UDP)). Todos los otros servicios deben de ser permitidos.

Para cumplir dichos requerimientos puede elaborarse una lista de control de acceso extendida, como se muestra en la figura 229.

Figura 229. **Lista de control de acceso extendida**

```
Router(config)# ip access-list extended servicios
Router(config-ext-nacl)# deny tcp host 192.168.30.2 host 192.168.3.2 eq 80
Router(config-ext-nacl)# deny udp host 192.168.30.3 host 192.168.3.3 eq 69
Router(config-ext-nacl)# permit ip any any
```

Fuente: elaboración propia.

Las listas extendidas tienen la ventaja, que al ser más granulares, pueden aplicarse en muchos puntos de la topología y aun así cumplir su propósito. Sin embargo, para evitar tráfico y procesamiento innecesario es una buena práctica colocarlas lo más cerca posible al origen de la transmisión.

La última consideración antes de aplicar la lista extendida, involucra nuevamente la diferencia existente entre la topología física y la lógica. La red asignada a los técnicos utiliza la VLAN 20 y como puerta de enlace predeterminada la subinterfaz virtual *FastEthernet* 0/0.20, de aplicarse la lista en la interfaz *FastEthernet* 0/0 esta no tendrá ningún efecto, ya que en el nivel lógico, esta interfaz no recibe tráfico.

Figura 230. **Lista de control de acceso extendido de "servicios"**

```
R1(config)# ip access-list extended servicios
R1(config-std-nacl)# remark [> Deniega http y ftp a ciertos hosts <]
R1(config-ext-nacl)# deny tcp host 192.168.30.2 host 192.168.3.2 eq 80
R1(config-ext-nacl)# deny udp host 192.168.30.3 host 192.168.3.3 eq 69
R1(config-ext-nacl)# permit ip any any
```

Fuente: elaboración propia.

Figura 231. ***Inter FastEthernet* 0/0.20**

```
R1(config)#inter fastEthernet 0/0.20
R1(config-subif)#ip access-group servicios in
```

Fuente: elaboración propia.

3.16.4. Otras herramientas

Uno de los riesgos de trabajar con listas de control de acceso consiste en que un mal diseño o aplicación puede cortar la comunicación en una red o terminar una sesión remota de manera inesperada.

Asimismo, es una tarea común modificar estas listas, ya sea para agregar o quitar sentencias u optimizarlas en algún sentido. Se ofrecen algunas herramientas para minimizar el riesgo y ayudar al mantenimiento de las ACLs.

3.16.4.1. Números de secuencia

Como ya se había mencionado, las sentencias de una lista de control poseen un número de secuencia que indica el orden en que estas serán evaluadas y que facilitan la introducción y remoción de las mismas.

Según la lista extendida del último ejemplo se tiene la configuración mostrara en la figura 232.

Figura 232. ***Show ip access-lists***

```
R1# show ip access-lists

Extended IP access list servicios
    10 deny tcp host 192.168.30.2 host 192.168.3.2 eq www
    20 deny udp host 192.168.30.3 host 192.168.3.3 eq tftp
    30 permit ip any any
```

Fuente: elaboración propia.

Si se pretende modificar esta lista, para permitir TFTP y denegar al *host* 192.168.30.2 acceso al servidor web, pueden utilizarse los números de secuencia de estas para remover e incluir las sentencias necesarias.

Figura 233. **Uso de números de secuencia**

```
R1(config)# ip access-list extended servicios
R1(config-ext-nacl)# no 20
R1(config-ext-nacl)# 15 deny tcp host 192.168.30.3 host 192.168.3.2 eq 80
R1(config-ext-nacl)# do show ip access-lists

Extended IP access list servicios
    10 deny tcp host 192.168.30.2 host 192.168.3.2 eq www
    15 deny tcp host 192.168.30.3 host 192.168.3.2 eq www
    30 permit ip any any
```

Fuente: elaboración propia.

Si el incremento entre las sentencias no es suficiente para incluir nuevas de ellas, puede utilizarse el comando *resequence* y especificar el número de secuencia inicial y el incremento a utilizar.

Figura 234. **Comando *resequence***

```
R1(config)#ip access-list resequence servicios 10 10

R1#show ip access-list
Extended IP access list servicios
    10 deny tcp host 192.168.30.2 host 192.168.3.2 eq www
    20 deny tcp host 192.168.30.3 host 192.168.3.2 eq www
    30 permit ip any any
```

Fuente: elaboración propia.

3.16.4.2. Reinicio programado

Una manera burda de prevenir los problemas ocasionados por una lista de control mal aplicada es el reinicio programado. Existen dos maneras de programarlo con las siguientes palabras clave:

- *At:* Reinicia el dispositivo en una fecha específica
- *In:* Reinicia el dispositivo en una cantidad determinada de minutos

De este modo puede programarse el reinicio de un dispositivo, si hubiere mala aplicación de una ACL. Este arrancará de nuevo utilizando la última configuración guardada.

Figura 235. **Reinicio programado**

```
Router# reload in 5
Reload scheduled in 5 minutes by console
Reload reason: Reload Command
Proceed with reload? [confirm]
Router#

***
*** --- SHUTDOWN in 0:05:00 ---
***

Router#
*Mar  1 00:01:02.571: %SYS-5-SCHEDULED_RELOAD: Reload requested for 00:06:00
UTC Fri Mar 1 2002 at 00:01:00 UTC Fri Mar 1 2002 by console. Reload Reason: Reload
Command.

Router# show reload
Reload scheduled in 4 minutes and 52 seconds by console
Reload reason: Reload Command
```

Fuente: elaboración propia.

Adviértase el uso del comando *show reload*, para visualizar cuándo será el próximo reinicio programado.

Para cancelar el reinicio del dispositivo puede utilizarse la instrucción *reload cancel*, según se muestra en la figura 236.

Figura 236. **Instrucción para cancelar reinicio**

```
Router# reload cancel
Router#


***
*** --- SHUTDOWN ABORTED ---
***

Router#
*Mar  1 00:03:38.599: %SYS-5-SCHEDULED_RELOAD_CANCELLED:  Scheduled reload
cancelled at 00:03:38 UTC Fri Mar 1 2002
```

Fuente: elaboración propia.

3.16.4.3. *Configuration rollback*

Una manera más moderna de retornar a una configuración funcional después de haber cometido un error es realizar un *configuration rollback*, donde la palabra inglesa *rollback* hace referencia a desplegar o traer algo de regreso, en este caso una configuración anterior.

Esta instrucción, en particular, tiene ciertos requerimientos, entre ellos que la memoria disponible del dispositivo sea más grande que el tamaño de los dos archivos de configuración (actual/anterior) combinados y que la capacidad de archivar (*archive*) configuraciones se encuentre activa.

Al utilizar este comando es posible revertir la configuración automáticamente a un estado anterior, si las instrucciones ingresadas no son confirmadas en cierto límite de tiempo.

Para activar la capacidad de archivar configuraciones se utilizará la siguiente secuencia de comandos (ver figura 237). Esta indica la ruta donde

será almacenada la copia de seguridad y la acción que desencadenará la creación del mismo. En este caso se creará un respaldo cada vez que se guarde una nueva configuración.

Figura 237. **Secuencia de comandos para archivar la configuración**

```
Router(config)# archive
Router(config-archive)# path flash:/backup/backup.cfg
Router(config-archive)#write-memory

Router#dir flash:backup/
Directory of flash:/backup/

   5  -rw-        1056   Mar 1 2002 00:38:06 +00:00  backup.cfg-1

876544 bytes total (851968 bytes free)
```

Fuente: elaboración propia.

Para retornar a una configuración anterior después de 10 minutos, se puede ejecutar la instrucción descrita en la figura 238.

Figura 238. **Retorno a una configuración anterior**

```
Router# configure replace flash:/backup/backup.cfg-1 time 10
Timed Rollback: Backing up to flash:/backup/backup.cfg-2

This will apply all necessary additions and deletions
to replace the current running configuration with the
contents of the specified configuration file, which is
assumed to be a complete configuration, not a partial
configuration. Enter Y if you are sure you want to proceed. ? [no]: y
Total number of passes: 0
Rollback Done
```

Fuente: elaboración propia.

Para guardar los nuevos cambios, utilizar el comando *configure confirm,* antes que se cumpla el tiempo asignado para ejecutar el *rollback.*

Figura 239. ***Comando configure confirm***

```
Router# configure confirm
```

Fuente: elaboración propia.

3.17. *Network address translation* (NAT)

A principios de 1990 el crecimiento explosivo del internet empezó a causar preocupación entre los expertos debido al rápido crecimiento de las tablas de enrutamiento y el agotamiento de direcciones disponibles. A la espera de soluciones que pudieran funcionar a largo plazo, se crearon una serie de pequeños arreglos destinados originalmente a ser soluciones temporales de estos problemas, sin contar con su enorme y rápida adopción, lo que ha ocasionado que estos sigan vigentes, por lo menos hasta el momento en que se presenta este trabajo.

Uno de estos ajustes fue la reserva de ciertas direcciones para que pudieran ser reutilizables dentro de cada organización, para ralentizar de esta manera el agotamiento de direcciones disponibles y que en la actualidad reciben el nombre de direcciones privadas.

Al dejar de ser únicas, las direcciones reservadas para un uso privado dejaron de ser enrutables a través de internet, por lo que se hizo necesaria la creación de un mecanismo que permitiera cambiar o traducir estas direcciones a otras que pudieran comunicarse utilizando la red pública.

Para realizar dicha función se creó la traducción de direcciones de red comúnmente referida como *network address translation* (NAT).

Al estar en contraposición con la visión original del internet, en donde se favorecía la conexión de extremo a extremo y al ser considerado solamente como un paliativo temporal, NAT jamás fue estandarizado; lo que ocasionó que cada fabricante realizara su propia implementación y que muchos protocolos presenten problemas al ser utilizados en combinación con esta tecnología.

No obstante, los inconvenientes, NAT presenta también grandes ventajas al permitir que muchos dispositivos se conecten a la red mediante unas pocas direcciones públicas. De este modo se reducen costos y se facilita la migración de un proveedor de servicios hacia otro.

3.17.1. Tipos de NAT

Cisco define tres tipos de traducción: estática, dinámica y sobrecargada.

En las traducciones se distinguen las direcciones locales y globales. Las primeras son utilizadas dentro de las organizaciones y las últimas, empleadas fuera de las mismas.

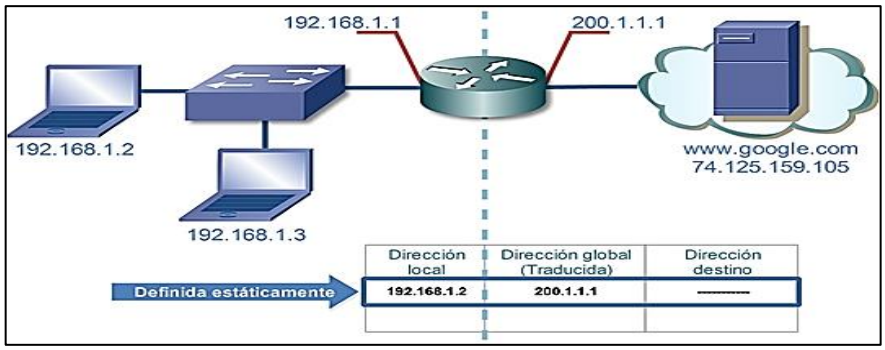
3.17.1.1. NAT estático

Es una traducción configurada manualmente y la única que permite el inicio de una conexión desde una red externa.

Puede realizarse de un modo sencillo, traduce de una dirección a otra o de una forma más granular, utiliza también distintos protocolos y números de puerto.

Es empleada regularmente cuando se necesita que un servicio presente en la red interna, sea accesible desde la red pública.

Figura 240. **NAT estático**

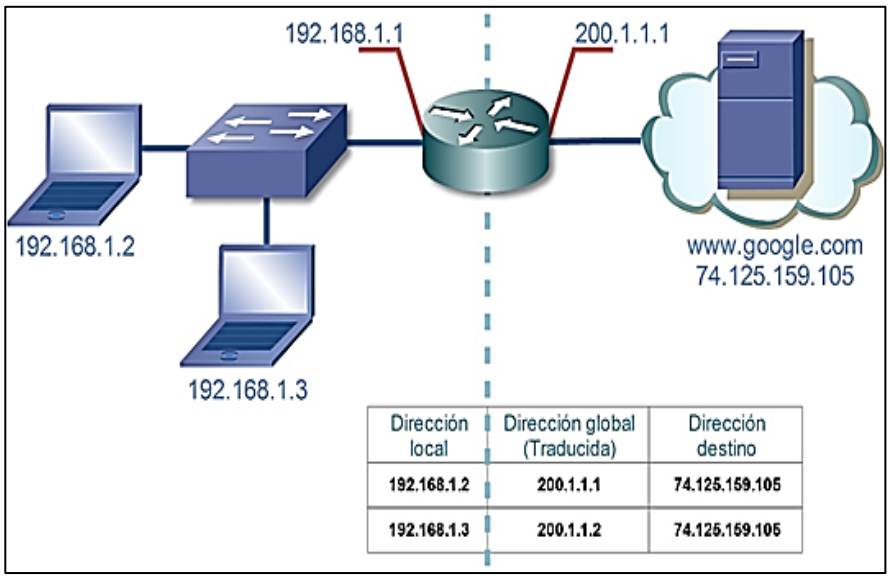


Fuente: elaboración propia, empleando *Edraw Max*.

3.17.1.2. NAT dinámico

Es una traducción realizada de manera automática, con carácter temporal. Puede realizarse de una dirección a otra; perteneciente a una interfaz o a una piscina de direcciones públicas. Este tipo de traducción es el que más consume de estas últimas, ya que se necesita de una dirección enrutable en internet, por cada dispositivo que requiera comunicarse a través de la misma.

Figura 241. **NAT dinámico**

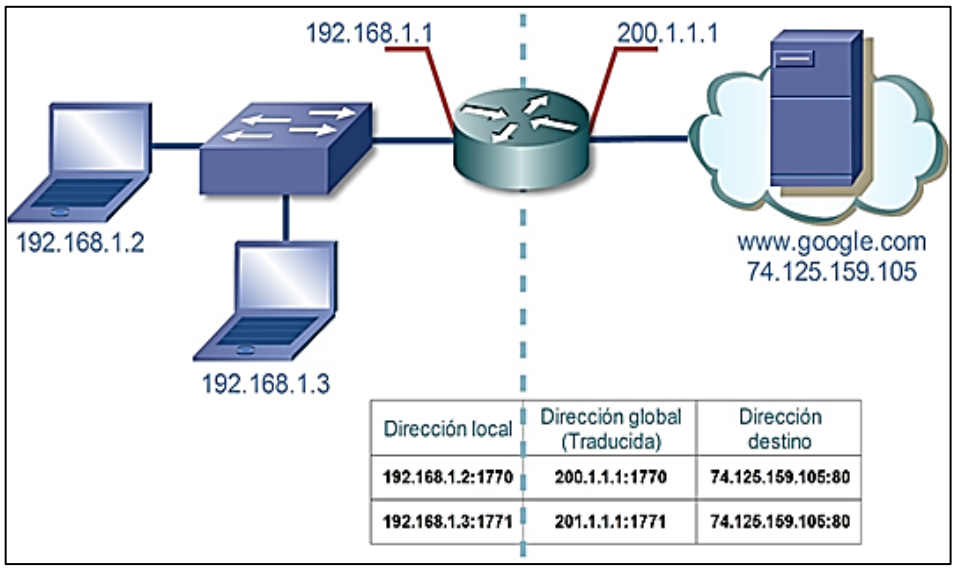


Fuente: elaboración propia, empleando *Edraw Max*.

3.17.1.3. NAT sobrecargado

También conocido como *port address translation* (PAT). Es una traducción que se realiza de manera automática utilizando la dirección presente en una interfaz o en una piscina de direcciones. Se distingue de NAT dinámico por su capacidad de utilizar números de puerto durante la traducción, por lo que varios dispositivos privados pueden compartir una sola dirección pública característica, esto lo convierte en el tipo de traducción más común.

Figura 242. **NAT sobrecargado o PAT**

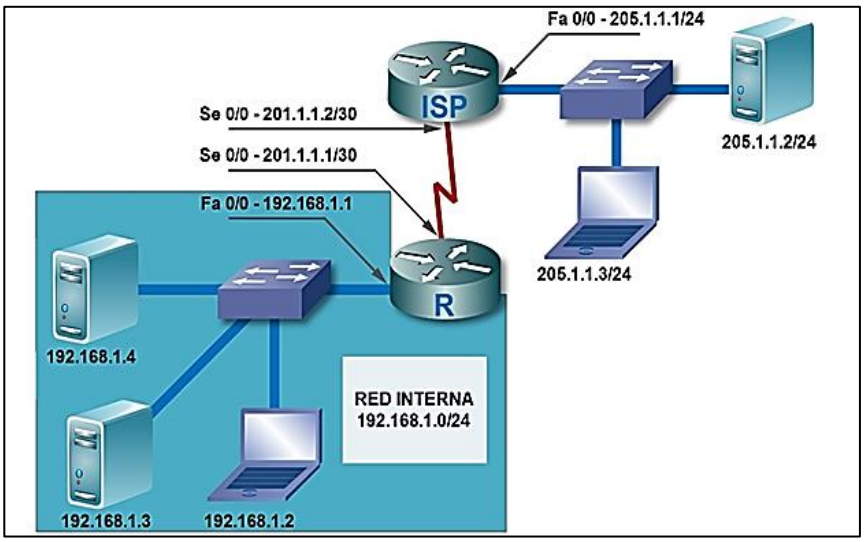


Fuente: elaboración propia, empleando *Edraw Max*.

3.17.2. Configuración tradicional

Se muestra la implementación de NAT en la siguiente topología. Todas las interfaces han sido previamente configuradas y existe una lista de control de acceso, en el *router* del proveedor de servicios de internet (ISP), encargada de descartar las transmisiones provenientes de redes que utilizan direccionamiento privado.

Figura 243. **Topología para mostrar la implementación de NAT**



Fuente: elaboración propia, empleando *Edraw Max*.

En este ejercicio se trabajará exclusivamente con el *router* R, perteneciente a la empresa en cuestión y donde deben cumplirse los objetivos que se describen en la figura 244.

Figura 244. **Objetivos del ejercicio**

1.	**Posibilitar la conectividad entre la red interna y el internet.**
2.	**Hacer accesibles, desde el internet, aquellos servidores presentes en la red interna, al utilizar:**
a.	**Direcciones públicas distintas para cada servidor.**
b.	**La misma dirección pública para ambos servidores.**

Fuente: elaboración propia.

Para cumplir el primer objetivo debe configurarse NAT sobrecargado dentro del *router* R, para que los dispositivos de la red interna con direcciones

privadas puedan compartir una sola dirección pública, siendo en este caso la dirección perteneciente a la interfaz *Serial 0/0* (201.1.1.1)

De manera general, los pasos a seguir para posibilitar la traducción de direcciones, consisten en identificar el tráfico que será traducido mediante una ACL, reconocer el rol de las interfaces ubicadas dentro (*inside*) o afuera (*outside*) de la red y habilitar NAT desde el modo de configuración global.

Para identificar el tráfico de la red interna a ser traducido se crea la lista de control estándar llamada “traducir” como se muestra en la figura 245.

Figura 245. **Lista de control estándar llamada “traducir”**

```
R(config)# ip access-list standard traducir
R(config-std-nacl)# remark [> Esta lista identifica el tráfico a traducir <]
R(config-std-nacl)# permit 192.168.0.0 0.0.255.255
R(config-std-nacl)# deny any
```

Fuente: elaboración propia.

Acto seguido, identificar las interfaces correspondientes a la parte interna y externa de la red. En esta oportunidad la interfaz *FastEthernet 0/0* pertenece adentro, mientras que la interfaz *Serial 0/0,* afuera de la misma.

Figura 246. **Identificación de interfaces**

```
R(config)# interface fastEthernet 0/0
R(config-if)#ip nat inside

R(config)#interface serial 0/0
R(config-if)#ip nat outside
```

Fuente: elaboración propia.

Finalmente es posible habilitar NAT con la siguiente instrucción.

Figura 247. **Habilitación de NAT**

```
R(config)# ip nat inside source list traducir interface serial 0/0 overload
```

Fuente: elaboración propia.

Dicha instrucción indica al *router* que habilite la traducción de las direcciones pertenecientes al interior de la red, utiliza aquellas definidas en la lista con el nombre “traducir”, alterándolas para utilizar en su lugar la dirección asignada a la interfaz *Serial 0/0* (201.1.1.1).

La palabra clave *overload* (sobrecarga) habilita NAT sobrecargado, su omisión da como resultado la activación de NAT dinámico.

Una vez lograda la conectividad con el internet, se procede a hacer los servidores internos accesibles desde la red pública, donde se parte del hecho que los roles (*inside/outside*) necesarios en NAT, han sido configurados en el paso anterior, por lo que se procede a realizar una traducción estática.

Para cumplir con el primer inciso del segundo objetivo, se emplea una dirección IP pública distinta para cada uno de ellos.

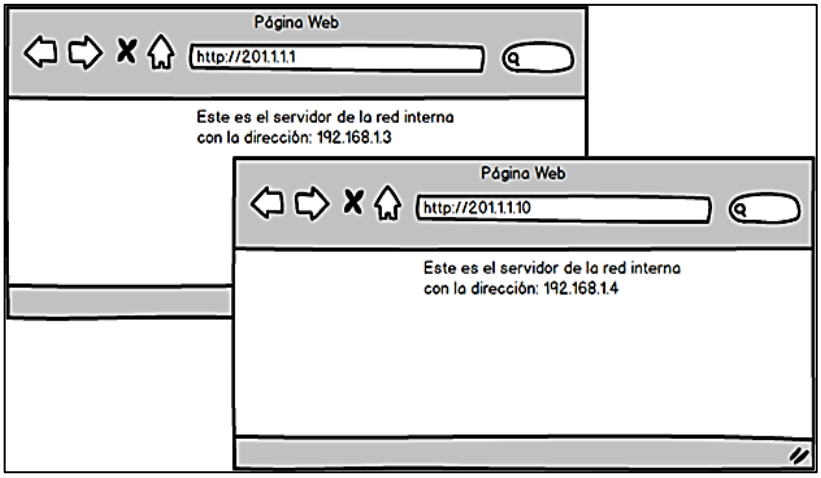
Figura 248. **Traducción estática**

```
R(config)# ip nat inside source static ?
  A.B.C.D   Inside local IP address
  esp         IPSec-ESP (Tunnel mode) support
  network  Subnet translation
  tcp          Transmission Control Protocol
  udp        User Datagram Protocol

R2(config)# ip nat inside source static 192.168.1.3 ?
  A.B.C.D    Inside global IP address
  interface  Specify interface for global address

R(config)# ip nat inside source static 192.168.1.3 201.1.1.1
R(config)# ip nat inside source static 192.168.1.4 201.1.1.10
```

Fuente: elaboración propia.

En esta ocasión se le indica a NAT que implemente una entrada estática (la cual siempre estará activa), para traducir entre una dirección local y una global, lo que significa que los servidores con las direcciones privadas 192.168.1.3 y 192.168.1.4 serán accesibles desde el mundo exterior, a través de las direcciones públicas 201.1.1.1 y 201.1.1.10, respectivamente.

Figura 249. **Servidores internos vistos desde la red pública**



Fuente: elaboración propia, empleando *Edraw Max*.

Si bien es necesario que el proveedor de servicios envíe todo el tráfico destinado a la dirección 201.1.1.10 al *router* de la empresa, adviértase que esta dirección no ha sido asignada en ningún momento a interfaz alguna de dicho dispositivo. Esto es debido a que la traducción de direcciones es realizada antes que el *router* consulte su tabla de enrutamiento, en otras palabras, NAT tiene precedencia.

La solución anterior es aceptable en algunos casos. Se vuelve problemática cuando se desea volver accesibles desde la red pública más servicios, por este motivo y para finalizar el ejercicio se eliminarán las entradas estáticas creadas anteriormente y se procederá a realizar una traducción más granular para que ambos servidores utilicen la misma dirección pública, pero un número de puerto diferente.

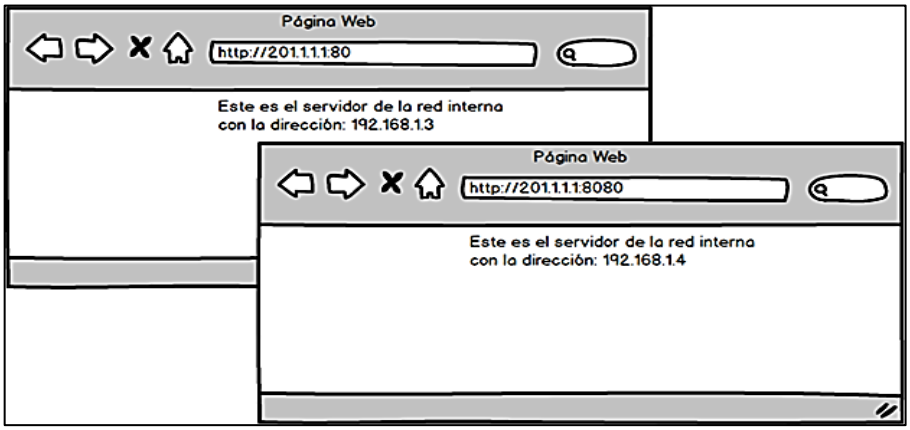
Figura 250. **Traducción más granular**

```
R(config)# no ip nat inside source static 192.168.1.3 201.1.1.1
R(config)# no ip nat inside source static 192.168.1.4 201.1.1.10

R(config)# ip nat inside source static ?
  A.B.C.D  Inside local IP address
  esp      IPSec-ESP (Tunnel mode) support
  network  Subnet translation
  tcp      Transmission Control Protocol
  udp      User Datagram Protocol

R(config)#ip nat inside source static tcp 192.168.1.3 ?
  <1-65535>  Local UDP/TCP port

 R(config)#ip nat inside source static tcp 192.168.1.3 80 201.1.1.1 ?
  <1-65535>  Global UDP/TCP port

R(config)# ip nat inside source static tcp 192.168.1.3 80 201.1.1.1 80
R(config)# ip nat inside source static tcp 192.168.1.4 80 201.1.1.1 8080
```

Fuente: elaboración propia.

En este caso se está realizando una traducción de los *sockets* compuestos por las direcciones privadas y el puerto 80 (puerto por defecto de HTTP) y la dirección pública. Nótese que junto a esta última deben utilizarse dos números de puerto diferentes (el 80 y el 8080), para poder realizar las dos traducciones requeridas.

Figura 251. **Servidores internos vistos desde la red pública**



Fuente: elaboración propia, empleando *Edraw Max*.

Para mostrar las traducciones (estáticas y dinámicas) puede utilizarse la instrucción *show ip nat translations,* como se muestra en la figura 251.

Figura 252. **Instrucción *show ip nat translations***

```
R# show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 201.1.1.1:80       192.168.1.3:80     ---                ---
tcp 201.1.1.1:8080     192.168.1.4:80     ---                ---
icmp 201.1.1.1:2644    192.168.1.4:2644   205.1.1.2:2644     205.1.1.2:2644
icmp 201.1.1.1:2900    192.168.1.4:2900   205.1.1.2:2900     205.1.1.2:2900
icmp 201.1.1.1:3156    192.168.1.4:3156   205.1.1.2:3156     205.1.1.2:3156
icmp 201.1.1.1:3412    192.168.1.4:3412   205.1.1.2:3412     205.1.1.2:3412
```

Fuente: elaboración propia.

3.17.3. Configuración con NAT *virtual interface* (NVI)

La NAT virtual interface (NVI) es una característica introducida por Cisco como una alternativa a asignar roles (inside/outside), permite activar NAT en las interfaces deseadas al usar el comando *ip nat enable*.

Una discusión detallada de la configuración de NAT con una NVI está fuera de los alcances de este estudio.

4. GUÍA PROPUESTA PARA EL AUXILIAR DEL LABORATORIO

Se propone una guía para el auxiliar con los temas, referencias y puntos más importantes del laboratorio. Nuevamente ubicados en el supuesto que la clase sea impartida una vez por semana y en un periodo de dos horas.

Las referencias: libros, materiales, recursos electrónicos, entre otros, se presentan según sea conveniente.

El contenido detallado de cada uno de los temas que se discuten en esta propuesta puede encontrarse en el capítulo 2.

4.1. Primera clase

- Inicio
  - Presentación de la clase

- Tema
  - Introducción al estudio de las redes y el modelo OSI

- Lineamientos

Durante esta clase se realiza la presentación del auxiliar del laboratorio, se explican generalidades de las redes de computadoras y la necesidad del estudio de las mismas. Se presentan también el horario en el que será impartido, el lugar, tareas y exámenes, así como sus respectivas

ponderaciones, el contenido del mismo (que viene del programa de estudios de la conocida certificación *Cisco Certified Network Associate* o CCNA), los simuladores, emuladores y el equipo disponible. Durante la segunda hora se presentan los temas pertinentes a la sección del "Modelo OSI". También es recomendable antes de terminar, elaborar una lista con los datos personales e información para contactar a los alumnos.

- Nota al auxiliar

  La parte más importante de esta clase es empezar a familiarizar al alumno con los distintos dispositivos que hacen posible la comunicación en una red, especialmente la diferencia entre el funcionamiento de un *switch* (que conecta dispositivos de manera local) y el *router* (que busca la mejor ruta hacia un destino), de igual importancia, es que el alumno comprenda las distintas capas del Modelo OSI y por qué son importantes, inclusive si todavía no conoce acerca de los distintos protocolos que regularmente se clasifican dentro de las mismas.

- Referencias

  - Capítulos 2 y 3 – *How to Master CCNA*
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "Cisco Foundations Network Components, Diagrams, Cables, and Speed"

- “Cisco Foundations Speaking the Language of the OSI Model” [Videos]
  Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
  Estados Unidos. Consultado el 9 de agosto de 2015 en http://www.cbtnuggets.com/

- Estándar ISO 7498-1:1994
  Página web. Consultado el 9 de agosto de 2015 en http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip

4.2. Segunda clase

- Temas
  - Introducción al estudio de las redes y el modelo OSI
  - Introducción al modelo TCP/IP

- Lineamientos

Como preámbulo realizar un repaso de los puntos más importantes de la clase anterior, componentes básicos de una red y el modelo OSI, sobre el cual es necesario extenderse un poco más. Se explicará cómo este modelo puede ser utilizado para buscar problemas dentro de una red y cómo se ubican los distintos dispositivos (cables, *switches y routers*) en cada una de sus capas y la razón. Luego proseguir con la temática relativa a “Introducción al modelo TCP/IP”.

- Nota al auxiliar

  Los temas de esta clase son importantes, ya que es un aprendizaje básico sobre el que se construirán las demás lecciones. El alumno debe comprender los diferentes tipos de tráfico en IPv4 (*unicast, broadcast, multicast*), así como los diferentes tipos de direcciones (*host, broadcast, multicast* y de red), saber cuál es la configuración mínima para que un dispositivo pueda comunicarse de una red a otra, así como verificar dicha configuración utilizando la herramienta "ipconfig" en Windows. Finalmente es necesario que comprenda sobre la función de un *router:* "conectar entre redes" y por qué no es posible configurar dos interfaces de este dispositivo dentro de una misma red. Al finalizar la lección, explicar la necesidad de las direcciones físicas (capa 2 direcciones MAC) y lógicas (capa 3 direcciones IP) y el *Address resolution protocol* (ARP).

- Tarea

  "Planeación es lo que eres y documentar es lo que haces" Un agregado de esta clase en particular es que se puede empezar a recalcar la importancia de la planeación y documentación, ya sea para implementar una nueva red o resolver un problema. Por esta razón se recomienda dejar como tarea la investigación de los siguientes temas: ITIL, COBIT y eTOM. Los cuales son a menudo utilizados en la industria como marcos de referencia.

- Referencias

  - Capítulo 4 – How to Master CCNA
    Molenaar R. (s.f.)

Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en http://gns3vault.com/product/how-to-master-ccna-rs/

- "Cisco Foundations Basic IP Addressing"

- "Cisco Foundations Basic IP Addressing - Filling in the Gaps" [Videos]
  Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
  Estados Unidos. Consultado el 9 de agosto de 2015 en http://www.cbtnuggets.com/

- "*The DoD* Internet *Architecture Model*"
  Página web. Consultado el 9 de agosto de 2015 en http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.88.7505&rep=rep1&type=pdf

4.3. Tercera clase

- Temas
  - Introducción al Modelo TCP/IP
  - TCP y UDP

- Lineamientos

De la misma forma que en las lecciones previas, es ideal comenzar con un repaso de los puntos más importantes de la clase anterior. Acto seguido se discutirá acerca de los dos protocolos que se ubican en la capa 4 del modelo OSI que son el *Transmission Control Protocol* (TCP) y

el *User Datagram Protocol* (UDP). Hacer énfasis en el primero, porque es el que utiliza la mayoría de aplicaciones. Proseguir con los temas de la sección “TCP y UDP”.

- Nota al auxiliar

Al finalizar esta clase es importante que el alumno comprenda la diferencia que existe entre los dos protocolos mencionados (básicamente la diferencia entre transmisión confiable y de mejor esfuerzo), así como el *3-Way Handshake* y el funcionamiento de los acuses de recibo y el tamaño de ventana. También deberá comprender la forma en que los números de puerto se relacionan con las aplicaciones, las cuales por conveniencia suelen utilizar los mismos números de puerto, ej.: HTTP utiliza el puerto 80, pero que no está necesariamente obligada a hacerlo. Antes de terminar la clase es recomendable un ejemplo de la herramienta *netstat* en Windows para que el alumno pueda apreciar cómo se utilizan los números de puerto de origen y destino en las conexiones hechas entre las computadoras.

- Tarea

Investigación de NMAP. El *Network Mapper* (NMAP) es una herramienta de auditoría de redes y análisis de seguridad que utiliza mucho de los conceptos utilizados durante esta clase para descubrir las aplicaciones que están funcionando dentro de una red. El programa y su documentación puede encontrarse de manera gratuita en nmap.org. Queda a criterio del auxiliar si realiza o no una demostración de dicha herramienta, aunque es muy recomendable.

- Referencia

  - Capítulo 5 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "Cisco Foundations How Applications Speak - TCP and UDP"
    "Cisco Foundations How Applications Speak - TCP and UDP, Part 2"
    [Videos]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

  - Página de NMAP"
    Página web. Consultado el 9 de agosto de 2015 en
    http://nmap.org/

4.4. Cuarta clase

- Temas
  - TCP y UDP
  - *Ethernet*

- Lineamientos

  Nuevamente se recomienda recalcar los puntos más importantes de la clase anterior. Este será un buen momento para que los estudiantes utilicen el analizador de protocolos llamado *Wireshark* y muestren una captura con todos los conceptos cubiertos en las clases anteriores (ARP, DNS, TCP, números de puerto, HTTP), para después continuar con los temas en la sección perteneciente a *ethernet.*

- Nota al auxiliar

  Esta clase es muy importante debido a que es la última oportunidad de repasar los conceptos básicos y resolver las dudas que los estudiantes tengan acerca de la teoría fundamental. Asimismo, temas importantes como el cableado estructurado y la fibra óptica son abordados muy superficialmente debido a las limitaciones de tiempo y equipo de laboratorio. Al finalizar la lección es importante que el estudiante comprenda la diferencia entre un *hub* y un *switch*, tipos de transmisión: *half* y *full-duplex,* y una noción básica de las categorías y estándares de cables: UTP, fibra, plenum, entre otros, mejores prácticas para la instalación de cableado y especialmente medidas de seguridad.

- Tareas
  - Lectura y resumen del capítulo 2 del libro *Head First Networking*.
  - Investigación de la diferencia entre SAN, NAS y DAS.

- Referencias

  - Capítulo 6 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - Capitulo 2 - Head First networking
    Al Anderson & Ryan Benedetti (2009)
    Estados Unidos. O'reilly.

  - "Switching Welcome to the World of Switching!"
    [Video]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

  - The Fiber Optic Association
    Página web. Consultado el 9 de agosto de 2015 en
    http://www.thefoa.org/

  - Estándar IEEE802.3 (*ethernet*)
    Página web. Consultado el 9 de agosto de 2015 en
    http://standards.ieee.org/findstds/standard/802.3-2012.html

4.5. Quinta clase

- Primer examen parcial

- Lineamientos

Durante la quinta sesión se sugiere llevar a cabo el primer examen parcial. Aunque a primera vista la evaluación pareciera ser precipitada, a través de ella se pueden afianzar, en el estudiante, los conceptos fundamentales antes de avanzar a temas más complejos. El auxiliar debe asegurarse que se evalúen los conceptos más importantes del contenido impartido.

4.6. Sexta clase

- Tema
  - Introducción al cisco IOS

- Lineamientos

El estudiante aprenderá cómo establecer una conexión con un dispositivo de red y a interactuar con el sistema operativo utilizado por el equipo disponible en el laboratorio: el Cisco IOS. Antes de empezar la clase es importante decidir la manera en que los estudiantes trabajarán con el equipo. Lo ideal es preparar previamente los dispositivos que serán utilizados, junto con su cable de poder y el cable de consola. Que haya por lo menos un aparato por cada tres estudiantes. Después se podrá continuar con los temas indicados en la sección “Introducción al Cisco IOS”.

- Nota al auxiliar

  En esta clase el estudiante deberá aprender a conectarse a un dispositivo de manera local, utilizando un cable y puerto de consola, así como a leer los indicadores físicos del equipo y buscar ayuda dentro del Cisco IOS, los diferentes modos del mismo y los atajos de teclado que puede utilizar. Además se discutirá con ellos el software utilizado para la simulación/emulación de redes, en este caso, el *Packet tracer* de Cisco y el GNS3.

- Práctica

  “Configuración del reloj de un dispositivo utilizando la ayuda del Cisco IOS”. Dentro de los puntos importantes se encuentran los atajos del teclado, el uso de la tecla “Tab” para completar comandos y el hecho de que el IOS puede aceptar comandos incompletos siempre y cuando estos sean reconocidos como únicos.

- Referencias

  - Capítulo 7 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - “Switching: Working with the Cisco IOS”
    [Video]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*

Estados Unidos. Consultado el 9 de agosto de 2015 en http://www.cbtnuggets.com/

4.7. Séptima clase

- Tema
  - Configuración básica de un dispositivo.

- Lineamientos

En esta oportunidad se introducirán algunos parámetros necesarios para conectarse a un dispositivo de manera remota y así asegurar los dispositivos. Para este propósito, el dispositivo elegido será un *switch* porque su configuración es más sencilla. Nótese que este dispositivo en particular, no necesita de ninguna configuración, en especial para su labor principal: conmutar paquetes dentro de una red local, lo que deberá ser indicado al estudiante para evitar confusión.

- Nota al auxiliar

El estudiante deberá comprender los parámetros que necesita configurar para poder establecer una conexión remota con un dispositivo mediante un protocolo sencillo y no seguro: Telnet. Asímismo, establecer la diferencia entre encriptación (en esta práctica se podrá apreciar un ejemplo de cifrado Vigenere) y un *hash* (en esta práctica se apreciará el MD5). Al final se tendrá que visualizar la configuración en ejecución (*running-config*) y la que se ejecuta cuando el dispositivo inicia (*startup-config*) y cómo salvar dicha configuración.

- Práctica
  - **“**Configuración inicial de un *switch*”

- Tarea
  - Repetir la práctica en casa para que el estudiante se vaya familiarizando con los comandos.

- Referencias

  - Capítulo 7 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - “Switching Base Configuration”

  - “Switching Base Configuration Part 2”
    [Videos]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.8. Octava clase

- Tema
  - Subredes y superredes

- Lineamientos

  En este punto es necesario realizar un repaso de algunos de los temas de la segunda clase, tales como los tipos de direcciones (red*, host, broadcast, multicast*), clases de direcciones IP (A, B, C y D) y la diferencia que existe entre direcciones públicas y privadas. Luego presentar la función de la máscara se subred.

- Nota al auxiliar

  En esta clase se introducirá la herramienta del *subnetting* que permite dividir una red grande en varias subredes para optimizar la utilización de las direcciones IPv4. Antes de la clase se necesita que todos los alumnos posean una copia de las hojas de trabajo o que el auxiliar tenga copias para repartir. Se examinan los casos en donde los requerimientos son cantidad de redes y número de *hosts.* A lo largo de los ejercicios, enfatizar la importancia del orden al realizar estos cálculos. Además, no olvidar mencionar que las direcciones de red y *broadcast* no pueden ser asignadas a ningún dispositivo. También, que las fórmulas para calcular cuántas redes y cuántos *hosts* se podrán obtener al aplicar una determinada mascara de subred.

- Práctica
  - Hojas de trabajo con ejercicios

- Tarea
  - Por lo menos un ejercicio de cada hoja de trabajo

- Referencias

- Capítulo 12 – How to Master CCNA
  Molenaar R. (s.f.)
  Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
  http://gns3vault.com/product/how-to-master-ccna-rs/

- "Routing Speaking Binary"

- "Routing Creating Subnets Based on Network Requirements"

- "Routing Creating Subnets Based on Host Requirements"

- "Routing Reverse Engineering Subnet Problems"
  [Videos]
  Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
  Estados Unidos. Consultado el 9 de agosto de 2015 en
  http://www.cbtnuggets.com/

4.9. Novena clase

- Tema
  - Subredes y superredes

- Lineamientos

En esta clase se cubrirá lo que haya quedado pendiente de la clase anterior y el último punto de la sección que cubre la técnica de *subnetting* llamada *Variable Length Subnet Mask* (VLSM).

- Nota al auxiliar

  Nuevamente se resalta la importancia que tiene documentar y llevar un orden adecuado al realizar estos cálculos. Además, recordar al alumno la importancia de la escalabilidad (la capacidad de crecer sin la necesidad de introducir cambios importantes), después de todo, las redes siempre tienden a crecer, así que este es un factor importante a tomar en cuenta en todos los planes y cálculos que se realicen.

- Práctica
  - Hoja de trabajo con ejercicios

- Tarea
  - Por lo menos un ejercicio de la hoja de trabajo

- Referencias

  - Capítulo 12 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - “Routing Variable Length Subnet Masking (VLSM)”
    [Video]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.10. Décima clase

- Tema
  - *Dynamic Host Configuration Protocol* (DHCP)
  - Enrutamiento estático

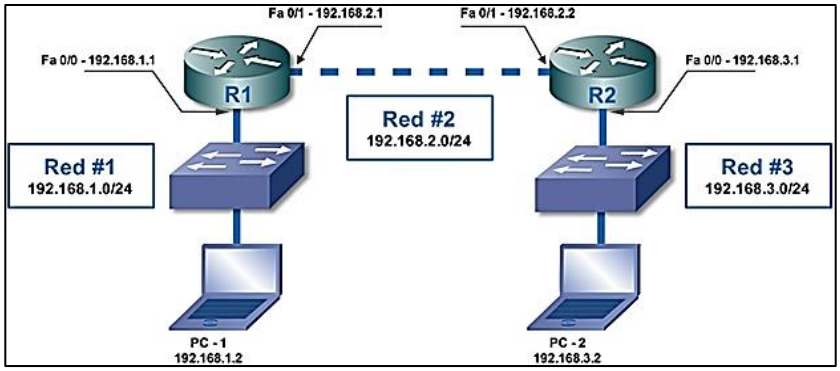
- Lineamientos

  En esta lección se cubrirán los puntos correspondientes a las secciones de "*Dynamic host configuration protocol* (DHCP)" y "enrutamiento estático"

- Nota al auxiliar

  En esta clase en particular, la teoría es bastante sencilla, por consiguiente se podrá dedicar más tiempo a la práctica. Al inicio deberá proporcionar una breve descripción de DHCP. Debe cubrirse el proceso a través del cual una computadora puede obtener su direccionamiento de manera automática, sin olvidar puntos importantes como reservar direcciones para su uso estático en servidores, impresoras, puertas de enlace por defecto, entre otros, y el hecho de que DHCP utiliza *broadcast,* razón por la que el uso de un DHCP *Relay* se hace necesario si se desea alcanzar un servidor de direcciones que se encuentre en un punto diferente de la red. Esta será la primera vez que los estudiantes trabajen con un *router,* por lo que es importante realizar la práctica a un ritmo adecuado y explicar a los alumnos cómo encender interfaces (*no shutdown*) y asignarles direcciones. Al finalizar los ejercicios de DHCP, se pedirá a los estudiantes que armen la topología básica que se introdujo en el capítulo anterior y que se reproduce nuevamente.

Figura 253. **Topología base para los ejemplos de las secciones de enrutamiento**



Fuente: elaboración propia, empleando *Edraw Max*.

Dicha topología se utilizará durante toda la parte de enrutamiento, por lo que debe pedírsele al estudiante que guarde la configuración. Se recomienda que antes de configurar el enrutamiento estático los estudiantes realicen un *ping* para verificar conectividad (las computadoras deberían tener conexión con sus respectivas puertas de enlace) y se introduzca la herramienta *traceroute* junto con el importante concepto de la tabla de enrutamiento, hacer referencia al comando que se utiliza en el Cisco IOS para mostrarla, ya que un *router* solo aprende por defecto aquellas redes que están directamente conectadas y finalmente explicar qué es una ruta por defecto.

- Prácticas
  - Configuración de DHCP en un *router.*
  - Configuración de un DHCP *relay.*
  - Configuración topología inicial (encender interfaces y asignar las direcciones IP correspondientes).
  - Configuración enrutamiento estático

- Tarea
  - Hoja de trabajo: “DHCP y enrutamiento estático”

- Referencias
  - Capítulo 13 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - “Routing Implementing Static Routing”
    [Video]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.11. Undécima clase

- Tema
  - Enrutamiento dinámico

- Lineamientos

  En esta lección se cubrirán los puntos correspondientes a la sección “Enrutamiento dinámico”.

- Nota al auxiliar

  En esta clase se explicará cómo los *routers* aprenden nuevas rutas a través de protocolos de enrutamiento, se discutirá al principio la clasificación de los mismos y luego se abordará el *Routing Information Protocol* (RIP). Este es un protocolo antiguo que ya no se toma en cuenta en ningún examen de certificación; sin embargo, su métrica sencilla y la facilidad de configuración lo hacen ser el protocolo ideal para introducir al estudiante al tema. En el transcurso de la práctica se plantearán conceptos importantes como los mecanismos para prevenir bucles de enrutamiento, el funcionamiento de la tabla de enrutamiento y los criterios que utiliza para elegir la mejor ruta (distancia administrativa, métrica y ruta más específica).

- Prácticas
  - Configuración de RIP en la topología básica

- Referencias
  - Capítulos 13 y 15 – *How to Master CCNA*
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "Routing Protocols Concepts"
    [Video]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.12. Duodécima clase

- Tema
  - *Open shortest path first* (OSPF)

- Lineamientos

  Al inicio de esta clase es importante hacer un repaso de los temas relevantes de la lección anterior, especialmente aquellos relacionados con el funcionamiento de la tabla de enrutamiento. Después se cubrirán los primeros puntos de la sección: "*Open shortest path first* (OSPF), hasta la parte de la configuración básica del protocolo.

- Nota al auxiliar

  OSPF es el tema más complejo a tratar durante este curso, por este motivo su enseñanza debe tomarse con calma. Entre los conceptos más importantes se encuentran los requerimientos para configurar OSPF, su funcionamiento basado en áreas, las tablas que mantiene y el uso de la *Wildcard Mask*.

- Prácticas
  - Configuración de OSPF en la topología básica

- Referencias
  - Capítulo 16 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en

http://gns3vault.com/product/how-to-master-ccna-rs/

- “Routing Understanding and Configuring OSPF” [Video] Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1* Estados Unidos. Consultado el 9 de agosto de 2015 en http://www.cbtnuggets.com/

4.13. Decimotercera clase

- Tema
  - *Open shortest path first* (OSPF)

- Lineamientos

Es la segunda clase dedicada a OSPF, se debe empezar como es usual, con un repaso de los conceptos más importantes de la clase anterior. Recalcar la parte del funcionamiento en áreas y los requerimientos de OSPF, con énfasis en el identificador del router o *router ID* y el rol que juegan las interfaces de *loopback*. Continuar con los puntos restantes de la sección.

Nota al auxiliar

En esta clase asegura que el estudiante tenga un fundamento sólido del funcionamiento y características de este protocolo, así como de la técnica de sumarización de rutas.

- Prácticas
  - Sumarización de rutas

- Tarea
  - Ejercicio de sumarización de rutas

- Referencias
  - Capítulos 12 y 16 – *How to Master CCNA*
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "OSPF Multi-Area Configuration and Verification"
    [Video]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 2*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

  - RFC OSPF v. 2
    Página web. Consultado el 9 de agosto de 2015 en
    http://www.ietf.org/rfc/rfc2328.txt

4.14. Decimocuarta clase

- Tema
  - *Enhanced interior gateway routing protocol* (EIGRP)

- Lineamientos

  En esta clase se cubrirán los puntos correspondientes a la sección **"***Enhanced interior gateway routing protocol* (EIGRP)**"**

- Nota al auxiliar

  EIGRP es el protocolo de enrutamiento que al momento de la realización de este escrito solo ha sido implementado en equipo de la marca Cisco.

- Prácticas
  - Configuración de EIGRP en la topología básica

- Referencias
  - Capítulos 17 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "EIGRP The Benefits Terms and Metrics of EIGRP"

  - "EIGRP Configuring EIGRP"
    [Videos]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 2*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.15. Decimoquinta clase

- Tema
  - *Virtua*l LANs (VLANs), enlaces troncales y DTP

- Lineamientos

  En esta lección se vuelve a trabajar con los *switches,* mientras se cubren los temas correspondientes a la sección "*Virtua*l LANs (VLANs), enlaces troncales y DTP"

- Nota al auxiliar

  Las VLANs constituyen uno de los temas más importantes del laboratorio. El estudiante debe comprender que cada VLAN es su propio dominio de *broadcast,* lo que significa que cada una de ellas es una red diferente. Otro concepto importante será el de la VLAN nativa, especialmente para el protocolo 802.1q y cómo utilizar esta tecnología para mejorar el control de acceso y aplicar calidad de servicio.

- Práctica
  - Configuración de VLANs

- Referencias
  - Capítulos 9 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

- “Switching Understanding VLANs and Trunks”

- “Switching Understanding VTP and 802.1q”
  [Videos]
  Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
  Estados Unidos. Consultado el 9 de agosto de 2015 en
  http://www.cbtnuggets.com/

4.16. Decimosexta clase

- Tema
  - VLAN *trunking protocol* (VTP) e *Inter* VLAN *routing*

- Lineamientos

  En esta ocasión se retoma el concepto de VLANs y sus puntos más importantes. Se amplía la base teórica al definir la administración automática de las mismas a través del protocolo propietario VTP. Además se explica cómo volver a establecer conectividad entre ellas. Esta lección cubre los puntos de la sección “VLAN *trunking protocol* (VTP) e *Inter* VLAN *routing*”.

- Nota al auxiliar

  En esta lección el estudiante debe comprender que en toda red existe una topología física y una topología lógica. Dentro de la práctica perteneciente a esta clase, la topología física presenta un *switch* conectado a un *router* a través de un solo enlace, mientras que la lógica posee múltiples interfaces virtuales conectadas. Es necesario recalcar

que las interfaces virtuales poseen un funcionamiento muchas veces idéntico a una interfaz real (“que no las veamos no significa que no existan”); asimismo deben ser consideradas al momento de configurar otros protocolos. El método descrito en la parte de Inter-VLAN *routing* es el del “Router en un palo” o *router-on-a-stick*. Este procedimiento es una solución antigua, pero actualmente implementada. Si el tiempo de la clase lo permite, debería abordarse también el método más moderno que consiste en utilizar un *switch* multicapa.

- Práctica
  - Configuración de VTP e *Inter*-VLAN *Routing*

- Referencias

  - Capítulos 9 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - “Switching Configuring Trunking, VTP, and VLANs”

  - “Routing Practical Routing - Enhancing VLANs”
    [Videos]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.17. Decimoséptima clase

- Tema
  - *Spanning Tree* Protocol (STP)

- Lineamientos

  En esta oportunidad se cubrirá uno de los protocolos más importantes a nivel de la capa de enlace, desarrollado por la doctora Radia Perlman. Este protocolo permite cierto grado de redundancia a nivel de capa 2 y previene el problema conocido como "tormenta de *broadcast*". La clase cubre los puntos de la sección "*Spanning Tree Protocol* (STP)"

- Nota al auxiliar

  En esta clase, al finalizar la descripción y configuración de Spanning-tree, es importante hacer la observación que debido a la lentitud de estos protocolos (si se toma en cuenta las exigencias actuales), en los diseños se restrinjan las partes que trabajan puramente con *switches* de capa 2 tanto como sea posible. Un buen marco de referencia que se puede utilizar en el diseño de redes es el modelo jerárquico de 3 capas de Cisco.

- Práctica
  - Configuración de *Spanning Tree* y *Rapid Spanning Tree*

- Referencias
  - Capítulos 11 – How to Master CCNA
    Molenaar R. (s.f.)

Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en http://gns3vault.com/product/how-to-master-ccna-rs/

- "Spanning Tree Protocol Understanding STP"

- "Spanning Tree Protocol Enhancements to STP"
  [Videos]
  Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 2*
  Estados Unidos. Consultado el 9 de agosto de 2015 en http://www.cbtnuggets.com/

- Estándar IEEE 802.1D
  Página web. Consultado el 9 de agosto de 2015 en http://standards.ieee.org/getieee802/download/802.1D-2004.pdf

4.18. Decimoctava clase

- Tema
  - *Access control lists* (ACLs)

- Lineamientos

En esta clase se cubrirán los puntos de la sección "*Access control lists* (ACLs)".

- Nota al auxiliar

  En esta sesión es importante presentar los conceptos básicos del funcionamiento de las listas de control de acceso. Además, recordar que, por su nombre y por el hecho de que cada una de las listas está compuesta por sentencias “permitir” y “denegar”, puede generarse confusión en el alumno; ya que no están diseñadas solamente para limitar conectividad (aunque esta es la aplicación cubierta acorde al alcance de este curso), sino que son utilizadas para identificar o marcar tráfico. Otro punto importante es que en la creación y aplicación de listas de control de acceso pueden causar muchos problemas (pérdida de conexión, denegación de servicios e incluso impedir al usuario acceder remotamente), por lo que se debe proceder con cuidado.

- Práctica
  - Configuración de listas de control de acceso estándares y extendidas

- Referencias

  - Capítulos 18 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en http://gns3vault.com/product/how-to-master-ccna-rs/

  - “Routing Using Access Control Lists”

  - “Routing Configuring and Applying Standard Access Control Lists”

- “Routing Configuring and Applying Extended Access Control Lists” [Videos]
  Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
  Estados Unidos. Consultado el 9 de agosto de 2015 en
  http://www.cbtnuggets.com/

- Estándar IEEE 802.1D
  Página web. Consultado el 9 de agosto de 2015 en
  http://standards.ieee.org/getieee802/download/802.1D-2004.pdf

4.19. Decimonovena clase

- Tema
  - *Network address translation* (NAT)

- Lineamientos

  En esta clase se cubrirán los puntos de la sección “*Network address translation* (NAT)”. NAT depende de las listas de control de acceso para poder identificar el tráfico que será “traducido” a una nueva dirección, por lo que es recomendable realizar un repaso de los puntos más importantes de la clase pasada.

- Nota al auxiliar

  NAT es la pieza final que permitirá establecer conectividad desde LAN a la red pública. Es importante que el estudiante comprenda por qué es necesario traducir direcciones de una red interna a una externa en las redes IPv4 y cómo las traducciones estáticas permiten ofrecer servicios

que se encuentran en la red interna al mundo exterior. NAT es el último tema de la parte fundamental del curso.

- Práctica
  - Configuración NAT sobrecargado y estático

- Referencias

  - Capítulos 19 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "*Routing NAT Concepts*"

  - "*Routing NAT Configuration*"
    [Videos]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.20. Vigésima clase

- Tema
  - Introducción a la seguridad informática

- Lineamientos

  En la penúltima lección del curso se renunciará, de manera muy superficial, algunos temas que complementan el contenido del mismo.

- Nota al auxiliar

  En esta ocasión se darán directrices para crear las mejores contraseñas y salvaguardar la seguridad física de los dispositivos. También se establecerá la diferencia entre Telnet y SSH, con una muy breve introducción a la criptografía. Para finalizar, se indicará cómo aumentar la seguridad de un *switch* al utilizar *port-security*.

- Referencias
  - Capítulos 8 – *How to Master* CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "*Switching Configuring SSH, User Accounts, and Password Encryption*"

  - "*Switching Managing Port Security*"
    [Videos]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

4.21. Vigesimoprimera clase

- Temas
  - Introducción a las redes inalámbricas
  - Introducción a IPv6

- Lineamientos

  La última lección del curso cubre los puntos de las secciones “Introducción a las redes inalámbricas” e “introducción a IPv6”

- Nota al auxiliar

  En esta última clase se impartirán algunas nociones fundamentales de las tecnologías inalámbricas y la última versión del Internet Protocol. Si bien los dos temas son importantes, se dedicará más tiempo a IPv6, ya que será el nuevo protocolo de direccionamiento en los años venideros pese a que la fecha para su completa implementación todavía sigue siendo objeto de debate. La falta de educación en la versión 6, así como la fácil convivencia de la nueva y antigua versión, hará que IPv4 continúe siendo utilizada en el país durante muchos años. Si un cambio en la tecnología obligara a una migración completa hacia la última versión del protocolo, será función del auxiliar actualizar el contenido del programa para reflejar esta evolución. Sin embargo, como se podrá apreciar en esta sesión, mucho del conocimiento adquirido en la versión 4 puede ser utilizado directamente en la versión 6.

- Referencias

  - Capítulos 21 – How to Master CCNA
    Molenaar R. (s.f.)
    Estados Unidos. Autopublicado. Consultado el 9 de agosto de 2015 en
    http://gns3vault.com/product/how-to-master-ccna-rs/

  - "Routing IPv6 Concepts"

  - "Routing IPv6 Configuration"
    [Videos]
    Ciora J. (s.f.) *CBT Nuggets – CCNA ICDN 1*
    Estados Unidos. Consultado el 9 de agosto de 2015 en
    http://www.cbtnuggets.com/

  - Estándares 802.11 (Redes Inalámbricas)
    Página web. Consultado el 9 de agosto de 2015 en
    http://standards.ieee.org/getieee802/802.11.html

  - RFC 2460 (IPv6)
    Página web. Consultado el 9 de agosto de 2015 en
    http://tools.ietf.org/html/rfc2460

CONCLUSIONES

1. Puede apreciarse en el presente estudio, la distribución, contenido y calendarización del laboratorio propuesto que ha tenido como base el formato de otros laboratorios de la Escuela Mecánica Eléctrica. El contenido ha sido dispuesto a manera de cubrir los puntos más importantes del programa de la conocida certificación CCNA de la empresa Cisco, la cual es fabricante del equipo de red disponible en el laboratorio, y cuyo sistema operativo, el Cisco IOS, es utilizado como referencia en muchos textos en la materia.

2. Se presentan un resumen de los fundamentos teóricos para comprender el funcionamiento de las redes locales, con ejemplos y material complementario.

3. Acompaña este trabajo, una guía para el auxiliar dividido por clase con los temas a tratar, sugerencias para la enseñanza y recomendaciones de tareas y prácticas, así como referencias a textos académicos y estándares para que el docente pueda reforzar su conocimiento de cara a una nueva lección o para que el alumno ahonde más en un tópico específico.

4. El laboratorio propuesto se ha creado a manera que, al haber cubierto los conceptos fundamentales, pueda ser fácilmente extendido con la creación de nuevos laboratorios que aborden otras tecnologías como aquellas utilizadas por los proveedores de servicio o la telefonía IP.

RECOMENDACIONES

1. Extender el laboratorio propuesto (una vez este ya esté implementado), ya sea con la creación de nuevo material, incorporación de nuevos temas en cursos existentes o directamente a través de la creación de nuevos laboratorios.

2. Analizar periódicamente el contenido propuesto por este trabajo y determinar si el mismo sigue siendo relevante o necesita ser actualizado para mantener el paso de las nuevas tecnologías y cumplir la demanda del mercado laboral.

3. Debido al avance de las tecnologías de la información se percibirá, dentro de algunos años, el valor de la electrónica en general en función al grado de conectivad e interacción que esta pueda tener hacia otros dispositivos. Por esta razón se recomienda que se revise el pénsum de la carrera de Ingeniería Electrónica a manera de incluir materias relacionadas a la programación, los distintos sistemas operativos (en especial Linux) y la administración de proyectos, anticipándose de esta manera a la evolución de la electrónica a nivel mundial.

BIBLIOGRAFÍA

1. ANDERSON A.; BENEDETTI, R. *Head First Networking***.** Estados Unidos: O'reilly, 2009. 312 p.

2. CIORA J. (s.f.) *CBT Nuggets – CCNA ICDN 1 - ICDN 2* [Videos]**.** Estados Unidos. [en línea]. <http://www.cbtnuggets.com/>. [Consulta: 9 de agosto de 2015].

3. DONAHUE G. *Network Warrior.* Estados Unidos: O'reilly, 2011. 248 p.

4. Internet *Engineer Task Force Page*. [en línea.]. <www.ietf.org/>. [Consulta: 20 de agosto de 2015].

5. LAMMLE T. *CCNA Routing and Switching Study Guide*. Estados Unidos: Sybex, 2013. 164 p.

6. MOLENAAR R. (s.f.) *How to Master CCNA.* Estados Unidos. [en línea.]. <http://gns3vault.com/product/how-to-master-ccna-rs/>. [Consulta: 9 de agosto de 2015].