HERRAMIENTA SOFTWARE PARA LA SIMULACIÓN DE PROTOCOLOS DE ENRUTAMIENTO EN REDES IP

ARNOL ANDRÉS CHILITO CASTRO
VÍCTOR HUGO PAZ OCAMPO

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Popayán
2003

HERRAMIENTA SOFTWARE PARA LA SIMULACIÓN DE PROTOCOLOS DE ENRUTAMIENTO EN REDES IP

ARNOL ANDRÉS CHILITO CASTRO
VÍCTOR HUGO PAZ OCAMPO

Monografía de Grado

Director: Mag. Ing. Francisco Javier Terán

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Popayán
2003

A mis padres y a toda mi familia,
a quienes agradezco su comprensión y apoyo incondicional
durante esta etapa tan importante en mi vida.
Al bebé que viene en camino a quien le daré lo mejor de mi.

Arnol Andrés Chilito Castro

A mis padres y a mi familia,
que han sido para mi, ejemplo de vida.
Gracias por su dedicación y apoyo constante.

Víctor Hugo Paz Ocampo

AGRADECIMIENTOS

Queremos expresar nuestro agradecimiento al Ing. Francisco Javier Terán por su contribución en el desarrollo de este trabajo, al selecto grupo de docentes del departamento de Telecomunicaciones que contribuyeron en nuestra formación y al laboratorista Ing. Rodrigo Mendez por su especial colaboración.

Agradecemos a nuestros compañeros y amigos que hicieron más agradable esta etapa de nuestras vidas.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

INTRODUCCION

En la actualidad, las redes globales de información reúnen datos sobre temas diversos, como las condiciones atmosféricas, tráfico aéreo, transacciones bancarias, comercio electrónico, investigación, contacto con otras personas y hasta entretenimiento.

La mayor parte de las redes son entidades independientes, establecidas para satisfacer las necesidades de un solo grupo. Los usuarios escogen una tecnología de hardware apropiada a sus problemas de comunicación. De manera más importante, es imposible construir una red universal desde una sola tecnología hardware debido a que ninguna red satisface todas las necesidades de uso. Por esta razón durante los pasados años ha evolucionado TCP/IP, una tecnología que hace posible interconectar muchas redes físicas diferentes y las hace funcionar como una unidad coordinada.

El reto imperativo es que esta gran red funcione bien, lo cual no es una tarea fácil, para tal efecto se deben abordar muchos retos contemplados específicamente en el área de conectividad para garantizar que la comunicación se efectúe con la red que se desea y mas aún, con un equipo específico de esa red; es aquí donde entra en juego un procedimiento muy importante dentro de las redes TCP / IP, el enrutamiento (routing), el cual permite que los datagramas IP enviados por un host origen lleguen al host destino de forma adecuada, después de atravesar distintas redes y enrutadores.

Desde este punto de vista, los protocolos de enrutamiento, que operan a nivel de red, contribuyen y son responsables en la determinación de la ruta óptima y la conmutación del tráfico dentro de los enrutadores (routers) y los conmutadores (switches) de nivel 3.

Con el propósito de lograr una mejor comprensión didáctica de la implementación, operación y funcionamiento de los principales protocolos de enrutamiento más utilizados en las redes IP, se desarrolló este trabajo de simulación, cuyo resultado final beneficiará tanto a investigadores y docentes, como también a estudiantes de pregrado y postgrado, relacionados con esta temática.

1. ROUTING: PROTOCOLOS DE ENRUTAMIENTO EN REDES IP.

El proceso de lograr que cada máquina de una red se encuentre enlazada o unida a Internet se le denomina enrutamiento. Sin éste, la máquina estaría limitada sólo a una red física. El enrutamiento permite al tráfico de una red buscar el camino óptimo a un destino en cualquier lugar del mundo, pasando por supuesto a través de varias redes.

El enrutamiento se encuentra ubicado dentro de la capa de red en el modelo OSI/ISO. Esta capa se encarga de las conexiones entre máquinas a través del protocolo IP y puede ser realizado por los hosts (localmente) y especialmente por los enrutadores en redes externas.

1.1 Principios Básicos

Modelo de referencia OSI

En 1984, la Organización Internacional de Estandarización (ISO) desarrolló un modelo llamado **OSI (Open Systems Interconection**, Interconexión de sistemas abiertos), que consta de siete capas y es usado para describir cómo viaja la información a través de una red entre la conexión física y la aplicación del usuario final. Este modelo es el más conocido y usado para describir los entornos de red.

Figura 1.1 Modelo OSI

Como se muestra en la figura 1.1, en el modelo de referencia OSI, hay siete capas numeradas, cada una de las cuales ilustra una función de red particular. Las funciones más básicas, como el poner los bits de datos en el cable de la red están en la parte de abajo, mientras las funciones que atienden los detalles de las aplicaciones del usuario están arriba. En el modelo OSI el propósito de cada capa es proveer los servicios para la siguiente capa superior, resguardando la capa de los detalles de como los servicios son implementados realmente.

La división de la red en siete capas presenta las siguientes ventajas:

- Divide la comunicación de red en partes más pequeñas y sencillas.
- Normaliza los componentes de red para permitir el desarrollo y el soporte de los productos de diferentes fabricantes.
- Permite a los distintos tipos de hardware y software de red comunicarse entre sí.
- Impide que los cambios en una capa puedan afectar las demás capas, de manera que se puedan desarrollar con más rapidez.
- Divide la comunicación de red en partes más pequeñas para simplificar el aprendizaje.

Como se menciono anteriormente, el proceso de enrutamiento se ubica dentro de la capa de red por lo cual será la única capa que se explicará.

Capa de red

La capa de red es la responsable del desplazamiento de datos a través de un conjunto de redes (internetwork) y utiliza un esquema de direccionamiento para determinar el destino de los datos a medida que se desplazan a través de la red.

Las direcciones de capa de red utilizan un direccionamiento jerárquico que permite la existencia de direcciones exclusivas más allá de los límites de una red, junto con un conjunto de métodos para encontrar una ruta por la cual la información viaje.

Dispositivos de capa de red

Los dispositivos de la internetworking que operan en la capa de red del modelo OSI unen entre si o interconectan segmentos de red o redes completas. Estos dispositivos se denominan enrutadores (routers), los cuales transfieren paquetes de datos entre redes basándose en la información del protocolo de red.

Los enrutadores toman decisiones lógicas con respecto a la mejor ruta para el envío de datos a través de la internetwork y luego dirigen los paquetes hacia el segmento y el puerto de salida adecuado, la figura 1.2, muestra la representación gráfica de un enrutador.

Figura 1.2 Representación gráfica de un enrutador

Direccionamiento de la capa de red

La dirección de red ayuda al enrutador a identificar una red dentro de la internetwork.
El esquema de direccionamiento más usado y difundido es el direccionamiento del protocolo de Internet IP. Las direcciones IP se representan mediante un número binario de 32 bits o se puede representar por un número decimal de notación decimal: se dividen los 32 bits de la dirección en cuatro *octetos* (un octeto es un grupo de 8 bits). El valor decimal máximo de cada octeto es 255 (el número binario de 8 bits más alto es 11111111, y esos bits, de derecha a izquierda, tienen valores decimales de 1, 2, 4, 8, 16, 32, 64 y 128, lo que suma 255 en total).

Campos de la dirección IP

Cada dirección IP tiene dos partes. Una de ellas, identifica a la red y la otra identifica a un dispositivo específico dentro de esa red. Todas las maquinas que pertenecen a la misma red requieren el mismo numero de red el cual debe ser además único en Internet.

Como las direcciones IP están formadas por cuatro octetos separados por puntos, se pueden utilizar uno, dos o tres de estos octetos para identificar el número de red. De modo similar, se pueden utilizar hasta tres de estos octetos para identificar la parte del host de una dirección IP, para mas información sobre el direccionamiento IP ver el anexo A.

1.2 Principios de Enrutamiento

1.2.1 Selección de ruta y conmutación de paquetes

Los enrutadores poseen tablas de enrutamiento en las que almacenan la información de topología de la red que permite evaluar y determinar el mejor camino que pueden seguir los paquetes para llegar a su destino. Cuando le llegan los paquetes a través de una interfaz, el router debe extraer de ellos la dirección de la red a la que se dirigen, para saber a cuál de las redes que une debe enviar los paquetes, pero se puede dar el caso en que el enrutador

compruebe que el host destino no pertenece a ninguna de las redes que conecta. En este caso analizará sus tablas de enrutamiento para realizar las selecciones de ruta y enviará los paquetes a través de una segunda interfaz (función de conmutación) al siguiente enrutador a lo largo de la ruta. Se produce así una serie de saltos en los que los paquetes van siendo enviados a sucesivos enrutadores, hasta llegar a uno de ellos que sí está conectado a la red destino, y quien será el que entregue los paquetes al host destino. En la figura 1.3 se muestra un ejemplo de saltos sucesivos que dan los datagramas IP entre diversos enrutadores, para alcanzar su destino:

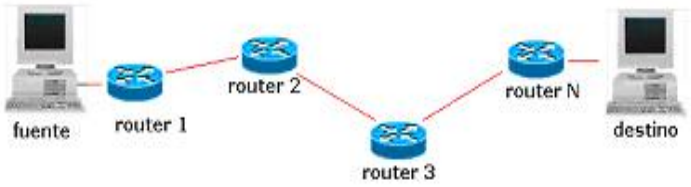


Figura 1.3 Saltos de los datagramas IP en los enrutadores

1.2.2 Diferencia entre protocolos enrutados y protocolos de enrutamiento

Se debe establecer la diferencia entre enrutamiento, protocolos enrutados y protocolos de enrutamiento.

Enrutamiento es el acto de reenviar paquetes basados en la información de las tablas de enrutamiento.

Protocolo enrutado es cualquier protocolo de red que proporcione suficiente información en su dirección de capa de red para permitir que un paquete se envíe desde un host a otro tomando como base el esquema de direccionamiento. Los protocolos enrutados definen los formatos de campo dentro de un paquete. Los paquetes generalmente se transfieren de un sistema final a otro. El Protocolo Internet (IP) es un ejemplo de protocolo enrutado

Los protocolos de enrutamiento soportan un protocolo enrutado proporcionando mecanismos para compartir la información de enrutamiento.

Un protocolo de enrutamiento define el conjunto de reglas utilizadas por un enrutador cuando se comunica con los enrutadores vecinos. Por ejemplo, un protocolo de enrutamiento describe:

- Cómo enviar actualizaciones
- Qué información contienen esas actualizaciones
- Cuándo enviar esa información

- Cómo ubicar a los destinatarios de las actualizaciones

Los siguientes son ejemplos de protocolos de enrutamiento TCP/IP:

- RIP (Routing Information Protocol o Protocolo de información de enrutamiento)
- IGRP (Interior Gateway Routing Protocol o Protocolo de enrutamiento de gateway interior)
- EIGRP (Enhanced Interior Gateway Routing Protocol o Protocolo de enrutamiento de gateway interior mejorado)
- OSPF (Open Shortest Path First o Primero la ruta libre más corta).
- BGP_4 (Border Gateway Protocol).

Para cumplir su objetivo los protocolos de enrutamiento hacen uso de los algoritmos de enrutamiento los cuales serán estudiados en la siguiente sección.

1.2.3 Algoritmos de enrutamiento

En la capa de red es donde residen los algoritmos que implementan los protocolos de enrutamiento.

El algoritmo de enrutamiento es la parte del software de la capa de red encargada de decidir la línea de salida por la que se transmitirá un paquete de entrada. Si la subred usa datagramas entonces esta decisión debe hacerse cada vez que llega un paquete de datos de entrada, debido a que la mejor ruta podría haber cambiado desde la última vez.

Clasificación de los algoritmos de enrutamiento

Algoritmos no adaptables

No basan sus decisiones de enrutamiento en mediciones o estimaciones del tráfico ni en la topología. La decisión de qué ruta tomar se calcula por adelantado, fuera de línea y se cargan en los enrutadores al iniciar la red. Éste procedimiento se llama **enrutamiento estático**. La desventaja de este tipo de algoritmos es que no es posible responder a situaciones cambiantes como por ejemplo saturación, exceso de tráfico o fallo en una línea.

Algoritmos adaptables

En contraste con los algoritmos no adaptables, éstos cambian sus decisiones de enrutamiento para reflejar los cambios de topología y de tráfico. Para poder tomar estas decisiones de encaminamiento dinámicas, los dispositivos involucrados en el enrutamiento deben intercambiar información usando un algoritmo de enrutamiento especial para este propósito. Este tipo de algoritmos no pueden ser demasiado complejos ya que son

implementados en los enrutadores y deben ejecutarse en tiempo real con recursos de CPU y la memoria con que el enrutador dispone.

La mayoría de los algoritmos de enrutamiento adaptables se pueden clasificar como uno de dos algoritmos básicos:

- vector-distancia, o
- estado-enlace.

El enrutamiento por vector-distancia determina la dirección (vector) y la distancia hacia cualquier enlace en la internetwork. El enrutamiento estado-enlace (también denominado *primero la ruta libre más corta*) recrea la topología exacta de toda la internetwork (o por lo menos la porción en la que se ubica el enrutador).

El enrutamiento híbrido balanceado combina aspectos de los algoritmos de estado-enlace y vector-distancia.

Enrutamiento Vector – Distancia

En este tipo de protocolo, cada enrutador conoce las rutas a las diferentes subredes y la métrica asociada a esa ruta. Periódicamente cada enrutador envía su tabla de rutas a todos sus vecinos y ellos calculan sus propias tablas con esta información y con los datos locales. La métrica puede representar el número de saltos o enrutadores intermedios.

El problema del conteo a infinito

Este algoritmo funciona bien en teoría, pero tiene un problema serio en la práctica: aunque converge en la respuesta correcta, puede hacerlo lentamente. En particular reacciona con rapidez a las buenas noticias, pero con lentitud ante las malas. Considere un enrutador cuya mejor ruta al destino *X* es larga. Si en el siguiente intercambio el vecino *A* informa repentinamente un alcance corto a *X*, el enrutador simplemente se conmuta a modo de usar la línea a *A* para enviar tráfico hasta *X*. En el intercambio de mensajes, se procesan las nuevas noticias.

Para ver la rapidez de propagación de las buenas noticias, considere la subred de 5 nodos (lineal) de la Fig. 1.4.

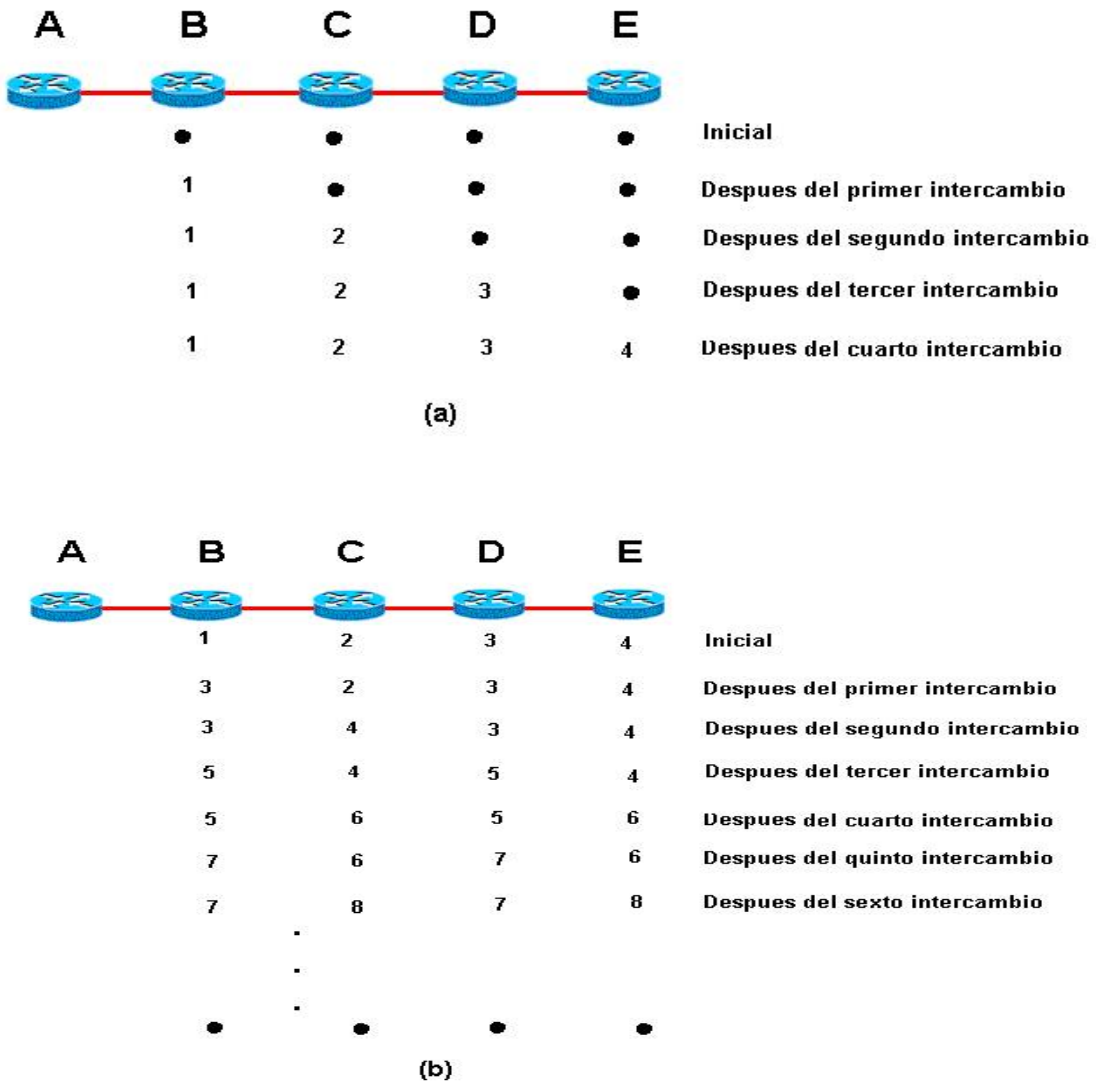


Figura. 1.4 Problema de conteo al infinito

Al activarse *A*, los enrutadores saben de él gracias a los intercambios de mensajes.
En el momento del primer intercambio, *B* se entera de que su vecino de la izquierda tiene una métrica de 1 hacia *A*. *B* crea una entrada en su tabla, indicando que *A* está usando una escala de distancia hacia la izquierda. El resto de los enrutadores aún piensan que *A* está desactivado. Las entradas de la tabla de *A* en este punto se muestran en la segunda fila de la figura 1.4 (a). Durante el siguiente intercambio *C* se entera de que *B* tiene una trayectoria a *A* de longitud 1, por lo que actualiza su tabla de enrutamiento para indicar una trayectoria de longitud 2, pero *D* y *E* no se enteran de las buenas nuevas sino hasta después. Como es evidente, las buenas noticias se difunden a razón de una escala por intercambio. En una subred cuya trayectoria mayor tiene una longitud de *N* escalas, en un lapso de *N* intercambios todo el mundo sabrá las líneas y enrutadores recientemente revividos.

Considerando ahora la situación de la figura 1.4 (b), en que todas las líneas y enrutadores están activos inicialmente. Los enrutadores *B, C, D* y *E* tienen distancias a *A* de 1, 2, 3 y 4, respectivamente. De pronto *A* se desactiva, o bien se corta la línea entre *A* y *B*, que de hecho es la misma cosa desde el punto de vista de *B*. En el primer intercambio de paquetes *B* no escucha nada de *A. C* sabe que tiene una trayectoria a *A* de longitud 2 y avisa a *B*, sin saber este último que la trayectoria de *C* pasa a través de *B* mismo. Como resultado, *B* ahora piensa que puede llegar a *A* por medio de *C. D* y *E* no actualizan sus entradas para *A* en el primer intercambio.

En el segundo intercambio, C nota que cada uno de sus vecinos indica tener una trayectoria a *A* de longitud 3. *C* escoge una de ellas al azar y hace que su nueva distancia sea 4. Los intercambios subsecuentes se muestran en el resto de la figura 1.4(b), a partir de esta figura queda clara la razón porqué las malas noticias viajan con tanta lentitud.

Recorte por horizonte dividido (solución)

Hay muchas soluciones a este problema, pero ninguna lo soluciona completamente. El algoritmo de horizonte dividido funciona de la misma manera que el enrutamiento por vector a distancia, excepto que la distancia a *X* no se informa en la línea por la que se envían paquetes para *X*. Usando el horizonte dividido, las malas noticias se propagan a razón de una escala por intercambio. Esta velocidad es mucho mejor que sin este algoritmo.

Enrutamiento por estado de enlace

Los algoritmos de enrutamiento basados en estado de enlace, también conocidos como algoritmos *SPF (primero la ruta libre más corta)*, mantienen una compleja base de datos de información de topología. Mientras que el algoritmo vector-distancia posee información no específica acerca de las redes distantes y ningún conocimiento acerca de los routers distantes, un algoritmo de enrutamiento estado de enlace conoce perfectamente los routers distantes y cómo se interconectan.

El concepto de este algoritmo es sencillo y se puede describir de la siguiente manera:

A. Descubrir vecinos.

Al ponerse en operación un enrutador, su primera tarea es averiguar quiénes son sus vecinos; esto se logra enviando un paquete especial de HOLA (*HELLO*).

B. Construcción de los paquetes de estado de enlace.

Una vez que se ha obtenido la información necesaria de los vecinos, el siguiente paso es que cada enrutador construya un paquete LSA (publicaciones del estado del enlace), el cual comienza con la identidad del transmisor, seguida de un número de secuencia, una edad y una lista de vecinos. En la figura 1.5 (a) se muestra un ejemplo de una subred, con las métricas en las líneas. Los paquetes de estado de enlace de los seis enrutadores se muestran en la figura 1.5 (b).

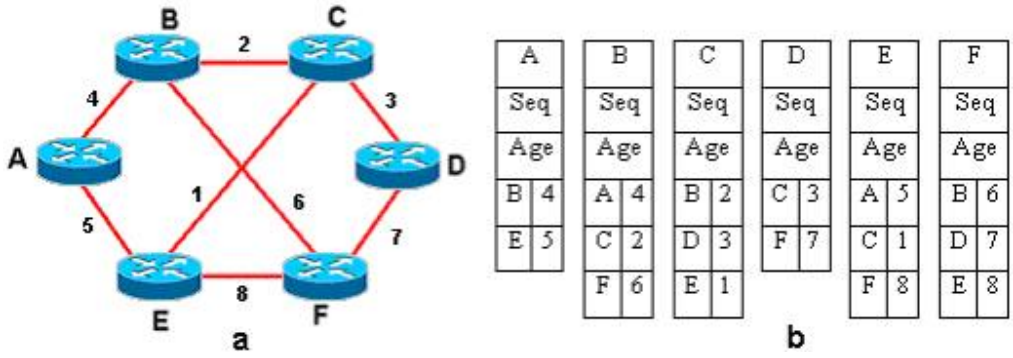


Figura 1.5 Creación de los paquetes del estado del enlace

C. Construcción de la Base de datos de topología

Los enrutadores intercambian LSA entre sí y generan cada uno en paralelo una base de datos topológica que contiene todas las LSA de la internetwork.

D. Cálculo de nuevas rutas.

Con la base de datos de topología el enrutador puede construir el grafo de la subred completa porque todos los enlaces están representados. El enrutador construye esta topología lógica como un árbol, con él mismo como raíz, y con todas las rutas posibles hacia cada red dentro de la internetwork que usa el protocolo estado-enlace.
Ahora puede ejecutarse el algoritmo SPF. Los resultados de este algoritmo se instalan en la tabla de enrutamiento.

E. Cambios en la topología

Con la mayoría de los protocolos de enrutamiento por vector-distancia, las actualizaciones para los cambios de topología consisten en actualizaciones periódicas de las tablas. La información pasa de un enrutador a otro, dando generalmente como resultado una convergencia más lenta. Con los protocolos de enrutamiento estado-enlace, las

actualizaciones son provocadas generalmente por cambios en la topología. Las LSA relativamente pequeñas que se han pasado a todos los demás enrutadores generalmente dan como resultado tiempos más rápidos de convergencia con cualquier cambio de topología de la internetwork.

Enrutamiento híbrido

Un tercer tipo emergente de protocolo de enrutamiento combina los aspectos del enrutamiento por vector-distancia y de estado de enlace. Este tercer tipo se denomina *enrutamiento híbrido balanceado*. Los protocolos de enrutamiento híbrido balanceado utilizan vectores de distancia con métricas más precisas para determinar las mejores rutas hacia las redes destino. Sin embargo, difieren de la mayoría de los protocolos por vector-distancia porque utilizan cambios de topología para provocar actualizaciones en las bases de datos de enrutamiento.

El protocolo de enrutamiento híbrido balanceado converge rápidamente, como los protocolos de estado de enlace. Sin embargo, difiere de los protocolos por vector-distancia y de estado de enlace en el sentido de que utiliza menos recursos de ancho de banda, memoria y ciclos del procesador. Ejemplos de protocolos híbridos son *IS-IS de OSI (Sistema intermedio a Sistema intermedio)* y *el protocolo EIGRP (Protocolo de enrutamiento de gateway interior mejorado) de Cisco*.

1.2.4 Métricas

Cuando un algoritmo de enrutamiento actualiza una tabla de enrutamiento, su objetivo principal es determinar cuál es la mejor información que debe incluir en la tabla. Cada algoritmo de enrutamiento interpreta lo que es mejor a su manera. El algoritmo genera un número, denominado métrica, para cada ruta a través de la red. Normalmente, cuanto menor sea la métrica, mejor será la ruta.

Se pueden calcular las métricas tomando como base una sola característica de la ruta. Se pueden calcular métricas más complejas combinando varias características. Las métricas utilizadas con mayor frecuencia por los enrutadores son las siguientes:

- *Ancho de banda*: capacidad de transmisión de datos de un enlace; (normalmente, se prefiere un enlace Ethernet de 10 Mbps)
- *Retardo*: cantidad de tiempo requerido para transportar un paquete por cada enlace desde el origen hacia el destino
- *Carga*: cantidad de actividad en un recurso de red tal como un enrutador o un enlace
- *Confiabilidad*: generalmente se refiere al índice de error de cada enlace de red

- *Número de saltos*: cantidad de enrutadores que un paquete debe atravesar antes de llegar a su destino
- *Costo*: valor arbitrario, generalmente basado en el ancho de banda, asignado por un administrador de red

1.2.5 Sistemas Autónomos

Un sistema autónomo o AS será la subred que es administrada por una autoridad común, que tiene un protocolo de enrutamiento homogéneo mediante el cual intercambia información en toda la subred y que posee una política común para el intercambio de tráfico con otras redes o sistemas autónomos. En Internet se dan, al menos, dos niveles jerárquicos de enrutamiento, el que realiza dentro de un sistema autónomo y el que se efectúa entre sistemas autónomos.

El primero es denominado **enrutamiento interno o intra áreas***,* al segundo se lo denomina **enrutamiento externo o ínter áreas***.* Dado que los requerimientos en unos y en otros son muy diferentes, se utilizan protocolos de enrutamiento muy distintos.

1.2.6 Protocolos entre gateways exteriores (Exterior Gateway Protocol (EGP))

Son utilizados para intercambiar información de enrutamiento entre diferentes sistemas autónomos en donde cada enrutador es responsable de la información de su propio sistema. Como ejemplo de este protocolo podemos citar al BGP (Border Gateway Protocol) y el EGP.

1.2.7 Protocolos entre gateways interiores (Interior Gateway Protocol (IGP))

Son usados para intercambiar información de enrutamiento entre enrutadores dentro de un sistema autónomo. Entre ellos se encuentran: RIP, IGRP, EIGRP y OSPF entre otros. Todos lo IGP cumplen la misma función, determinar la ruta óptima de destino, para ello utilizan los algoritmos ya estudiados:

- Vector de Distancia y
- Estado de Enlace.

En la tabla 1.1 se muestra una comparación entre los protocolos de enrutamiento interior más utilizados:

Protocolo	RIP	OSPF	IGRP	EIGRP
Tipo	Vector Distancia	Estado Enlace	Vector Distancia	Vector Distancia
Tiempo de convergencia	Lento	Rápido	Lento	Rápido
Consumo BW	Alto	Bajo	Alto	Bajo
Consumo de Recursos	Bajo	Alto	Bajo	Bajo
Soporte múltiples caminos	No	Si	No	Si
Escalabilidad	No	Si	Si	Si
Propietario	No	No	Si	Si
Enrutamiento no IP	No	No	No	Si

Tabla 1.1 Comparación entre los protocolos de enrutamiento interior

1.3 Routing Information Protocol RIP (Protocolo de información de enrutamiento)

Uno de los protocolos de enrutamiento más antiguos es el Routing Information Protocol o más comúnmente llamado RIP. RIP utiliza algoritmos de vector distancia para calcular sus rutas. Este tipo de algoritmos para calcular rutas fueron utilizados durante décadas en sus distintas variantes. De hecho los algoritmos de vector distancia utilizados por RIP están basados en aquellos algoritmos utilizados por ARPANET en el año 1969.

El protocolo RIP, tal cual se conoce actualmente, fue descrito por primera vez en el RFC 1058 (http://www.rfc-editor.org/rfc/rfc1058.txt) por C. Hedrick de la Rutgers University en Junio de 1988, y posteriormente fue mejorado en la RFC 2453 (http://www.rfc-editor.org/rfc/rfc2453.txt) por G.Malkin de la compañía Bay Networks en Noviembre de 1998. Desde el año 1998 el protocolo RIP se ha mantenido estable, aunque posteriormente salió la versión para Ipv6 en RIPng.

RIP es un protocolo de enrutamiento muy extendido en todo el mundo por su simplicidad en comparación a otros protocolos como podrían ser OSPF, IS-IS o BGP. Además de ser un protocolo abierto a diferencia de otros protocolos de enrutamiento como por ejemplo IGRP y EIGRP propietarios de Cisco Systems.

RIP está basado en el algoritmo de Bellman Ford y busca su camino óptimo mediante el conteo de saltos, considerando que cada enrutador atravesado para llegar a su destino es un salto. RIP, al contar únicamente saltos, como cualquier protocolo de vector distancia no tiene en cuenta datos tales como por ejemplo ancho de banda o congestión del enlace. RIP es un protocolo basado en UDP. Cada enrutador que usa RIP tiene un proceso de enrutamiento que envía y recibe datagramas sobre el puerto UDP número 520, los puertos para RIP-1/RIP-2.

1.3.1 Algoritmo de vector distancia implementado por RIP

Por cada destino en el sistema, el enrutador G guardara un estimativo actualizado de la métrica para ese destino (por ejemplo, el costo total de alcanzarlo) y la identidad del enrutador vecino sobre quien el dato de la métrica esta basado. Si el destino esta sobre una red que esta directamente conectada a G, entonces G simplemente usa una entrada que muestra el costo de usar la red y el hecho de que ningún enrutador es necesario para llegar al destino. Es fácil mostrar que una vez el cálculo ha convergido hacia la métrica correcta, el vecino que es recordado por esta técnica es de hecho el primer enrutador en el camino hacia el destino. Esta combinación de destino, métrica, y enrutador es típicamente referida como una ruta hacia el destino con esa métrica usando ese enrutador .

El siguiente procedimiento es llevado a cabo por varias entidades que participan en el protocolo de enrutamiento. Esto debe incluir todos los enrutadores en el sistema.

- Se guarda una tabla con una entrada por cada destino posible en el sistema. La entrada contiene la distancia D hacia el destino y el primer enrutador G sobre la ruta hacia la red. Conceptualmente debería haber una entrada para la entidad hacia si misma con métrica 0, pero esto actualmente no es incluido.
- Periódicamente, se envía una actualización de enrutamiento a cada vecino. Esta actualización es un conjunto de mensajes que contiene toda la información de la tabla de enrutamiento. Esta contiene una entrada por cada destino, con la distancia mostrada hacia ese destino.
- Cuando una actualización de enrutamiento llegue desde un vecino G', se suma el costo asociado con la red que esta compartiendo con G'. (esta debería ser la red sobre la cual la actualización llego.) siendo este resultado la distancia D'. Se compara la distancia resultante con la entrada en la tabla de enrutamiento actual. Si la nueva distancia D' para N es mas pequeña que el valor existente D, se adopta la nueva ruta. Esto es, cambia la entrada de la tabla para N, se pone métrica D' y enrutador G'. Si G' es el enrutador desde el cual había llegado la ruta existente, G' = G, entonces usa la nueva métrica aun si esta es mas grande que la antigua.

Esta discusión asume que la topología de la red es fija. En la práctica los enrutadores y las líneas a menudo fallan y se activan. Para manejar esta posibilidad se necesita modificar el algoritmo ligeramente.

La versión teórica del algoritmo involucra un mínimo sobre todos los vecinos inmediatos. Si la topología cambia, el conjunto de vecinos cambia. Por consiguiente, en el siguiente periodo se realiza el cálculo y el cambio es reflejado. Sin embargo, implementaciones actuales usan una versión incremental de la minimización. Solamente la mejor ruta para cualquier destino dado es recordada. Si el enrutador involucrado en esa ruta esta caído o la conexión de la red

hacia este se rompiera, el calculo puede nunca reflejar el cambio. El algoritmo como se muestra hasta ahora depende de un enrutador notificando a sus vecinos si cambian las métricas. Si el enrutador se cae, entonces este no tiene forma de notificar a sus vecinos del cambio.

A fin de manejar problemas de esta clase, el protocolo de vector distancia debe hacer alguna provisión para cronometrar rutas caídas. El detalle depende en el protocolo específico. Por ejemplo en RIP, cada enrutador que participa en el enrutamiento envía un mensaje de actualización a todos sus vecinos cada 30 segundos. Suponga que la ruta actual para la red N usa el enrutador G. si no oímos a G por 180 seg. Podemos asumir que o el enrutador esta caído, o la conexión a la red es inutilizable. Así marcamos la ruta como invalida. Cuando se escuche desde otro vecino que tiene una ruta valida para N, la ruta valida reemplazara la inválida. Note que se espera 180 segundos antes de sacar una ruta aunque se esperaría oír cada enrutador vecino cada 30 seg. Desafortunadamente, los mensajes son ocasionalmente perdidos por la red. Así es probable que no sea una buena idea invalidar una ruta basados en un simple mensaje perdido.

Es útil tener una forma para notificar vecinos que actualmente no tiene una ruta valida para alguna red. RIP, junto con varios protocolos de esta clase, hace esto a través de un mensaje de actualización normal, marcando la red como inalcanzable. Un valor específico de la métrica es escogido para indicar un destino inalcanzable; el valor de la métrica es mas largo que la métrica valida mas grande que esperamos ver. En la implementación existente de RIP, un valor de 16 es usado. Este valor es normalmente referido como “infinito”. 16 puede parecer un número sorprendentemente pequeño. Es escogido pequeño por las razones que se estudiaron en la sección 1.2.3.2.2 En muchas implementaciones, la misma convención es usada internamente para indicar una ruta como invalida.

Entre otras mejoras que implementa el protocolo RIP, para manejar el problema de conteo al infinito que se presenta en los algoritmos de vector de distancia se encuentran:

Horizonte dividido con poisoned reverse

El esquema de “horizonte dividido simple” estudiado en la sección 1.2.3.2.2 omite rutas aprendidas desde un vecino en actualizaciones enviadas hacia ese vecino. “horizonte dividido con poisoned reverse" incluye tales rutas en las actualizaciones pero pone la métrica en infinito.

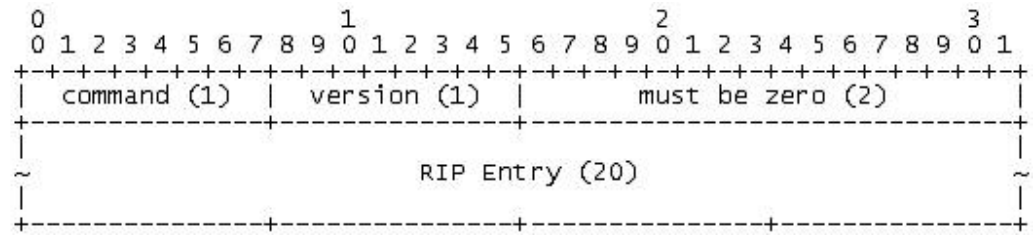
Actualizaciones triggered

El horizonte dividido con poisoned reverse prevendrá cualquier malla de enrutamiento que involucre dos enrutadores solamente. Sin embargo es posible terminar con un patrón en el cual tres enrutadores quedan enganchados en un engaño mutuo.

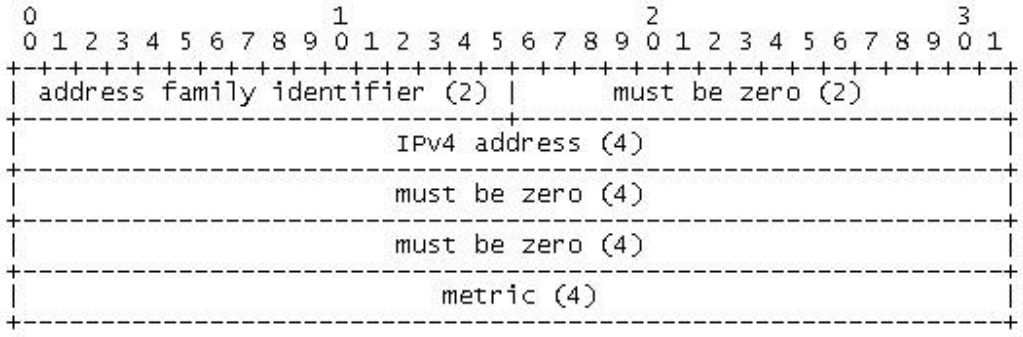
Para obtener actualizaciones triggered, adicionamos la regla de que siempre que un enrutador cambia la métrica para una ruta, se requiere enviar un mensaje de actualización casi instantáneamente, aun si no es tiempo de enviar un mensaje de actualización regular.

1.3.2 Formato de los mensajes

El formato del paquete RIP es:



aquí deben estar entre 1 y 25 (inclusive) entradas RIP. Una entrada RIP-1 tiene el siguiente formato:



El tamaño de los campos están dados en octetos. A menos que se especifique otra cosa, los campos contienen enteros binarios , con el octeto más significativo primero. Cada marca representa un BIT.

Cada mensaje contiene un encabezado RIP el cual consiste de un comando (command) y un número de versión. El campo comando es usado para especificar el propósito del mensaje. Los comandos implementados en la versión 1 y 2 son:

1- *Request* : una solicitud para que el sistema receptor envíe toda o parte de su tabla de enrutamiento
2- *Response*: un mensaje que contiene toda o parte de la tabla de enrutamiento del transmisor. Este mensaje puede ser enviado en respuesta a un request, o este puede ser una actualización de enrutamiento no solicitada generada por el transmisor.

Cada entrada en los mensajes contiene un identificador de la familia de direcciones (AFI), dirección de destino Ipv4, y el costo para alcanzar ese destino , métrica (campo metric).

El campo de la métrica contiene un valor entre 1 y 15 inclusive el cual especifica la métrica actual para el destino; o el valor 16 (infinito) el cual indica que el destino no es alcanzable.

Todos los mensajes de actualización de enrutamiento son enviados desde el puerto RIP. Mensajes de actualización de enrutamiento no solicitados tienen el puerto de fuente y destino igual al puerto RIP. Mensajes de actualización enviados en respuesta a una solicitud son enviados al puerto desde el cual la solicitud vino.

RIP es el protocolo de enrutamiento más ampliamente usado en Internet, por su fácil implementación, configuración y mantenimiento, pero presenta las siguientes desventajas:

- El protocolo esta limitado a redes cuyo camino más largo (diámetro de la red) son 15 saltos. Note que esta afirmación del límite asume que un costo de 1 es usado para cada red. Es así como RIP es normalmente configurado. Si el administrador del sistema escoge usar costos más largos, la frontera más alta de 15 puede fácilmente llegar a ser un problema.
- El protocolo depende de la cuenta al infinito para resolver ciertas situaciones inusuales. Si el sistema de redes tiene varios cientos de redes y una malla de enrutamiento se formó envolviéndolos a todos la resolución de la malla requerirá o mucho tiempo (si la frecuencia de las actualizaciones de enrutamiento fuera limitada) o ancho de banda (si las actualizaciones fueran enviadas cada ves que un cambio fuera detectado).
- Este protocolo usa métricas fijas para comparar rutas alternativas. Esto no es apropiado para situaciones donde las rutas en base a parámetros de tiempo real como una medida del retardo, confiabilidad, o carga.

1.4 Open Shortest Path First OSPF (Primero la ruta libre más corta)

Internet Engineering Task Force (IETF) desarrollo OSPF en 1998. la versión más reciente, que se conoce como versión 2 de OSPF, se describe en la RFC 2328. OSPF es un Interior Gateway Protocol, Protocolo IGP, lo que significa que distribuye información de enrutamiento entre los enrutadores que pertenecen al mismo sistema autónomo. OSPF fue escrito para

dar respuesta a las necesidades de las internetworks grandes y escalables que RIP no podía enfrentar. OSPF ofrece:

- *Velocidad de convergencia*. En grandes redes, la convergencia RIP puede tardar varios minutos mientras el algoritmo de enrutamiento pasa por una espera y un periodo de envejecimiento. Con OSPF, la convergencia es más rápida que con RIP, ya que los cambios en el enrutamiento se inundan inmediatamente y se calculan en paralelo.
- *Soporte para las máscaras de subred de longitud variable (VLSM).* OSPF soporta el enmascaramiento y las VLSM, en contraposición a RIPv1, que sólo soporta el enmascaramiento de subred de longitud fija.
- *Posibilidad de alcanzar la red*. Una red RIP que abarque más de 15 saltos se considera inalcanzable. OSPF no tiene prácticamente limitaciones de ese tipo.
- *Uso del ancho de banda*. Las difusiones RIP completan las tablas de enrutamiento de los vecinos cada 30 segundos. Esta operación es especialmente problemática en enlaces WAN más lentos. OSPF envía actualizaciones de estado de enlace solo cuando hay un cambio en la red.
- *Métodos para la selección de ruta*. OSPF utiliza un valor de costo, que se basa en la velocidad de conexión. Al igual que con RIP, IGRP y EIGRP, OSPF proporciona soporte para múltiples rutas equivalentes.

1.4.1 Terminología OSPF

- *Estado del enlace*. El estado de un enlace entre dos enrutadores, es decir, la interfaz de un enrutador y su relación con los enrutadores vecinos. Los estados de enlace se publican para los demás enrutadores en paquetes especiales, llamados publicaciones del estado de enlace (LSA).
- *Costo*. El valor que se asigna a un enlace. En vez de saltos, los protocolos de estado de enlace asignan un costo a un enlace, el cual esta basado en la velocidad de los medios.
- *Área*. Un conjunto de redes y enrutadores que poseen la misma identificación de área. Cada enrutador de un área posee la misma información de el estado de enlace. Un enrutador que este dentro de un área es un enrutador interno.
- *Vecinos*. Dos enrutadores que tienen interfaces en una red común. Una relación de vecindad la suele establecer y mantener el protocolo Hello.
- *Hello*. Protocolo que utiliza OSPF para establecer y mantener relaciones de vecindad.
- *Base de datos de vecindad*. Un listado de todos los vecinos con los que un enrutador ha establecido una comunicación bidireccional.
- *Base de datos del estado de enlace (base de datos de topología).* Una lista de entradas del estado de enlace de todos los demás enrutadores de la red. Muestra la topología de la red. Todos los enrutadores que hay dentro de un área tienen bases de datos del

estado de enlace idénticas. La base de datos del estado de enlace se fragmenta a partir de las LSA que generan los enrutadores.

- *Tabla de enrutamiento.* Se genera cuando el algoritmo de Dijkstra se ejecuta en la base de datos del estado de enlace. El contenido de cada tabla de enrutamiento es único. OSPF puede ejecutarse en redes retransmitidas o por redes no retransmitidas. La topología de una red repercute en el modo en que se crea la vecindad.
- *Topología de múltiple acceso con difusión.* Redes que soportan más de dos enrutadores conectados, con la capacidad de dirigir un mensaje físico (una difusión) a todos los enrutadores que estén conectados. Un segmento Ethernet constituye un ejemplo de red multiacceso con difusión, ver figura 1.6.
- *Tecnología punto a punto*. Una red que une un par sencillo de enrutadores. Una línea serie dedicada T1 constituye una red punto a punto, ver figura 1.7.
- *Topologías de múltiple acceso con difusión (NBMA).* Redes que soportan varios enrutadores (más de dos), pero que no tienen capacidad de difusión. Frame Relay y X.25 son ejemplos de redes multiacceso sin difusión, ver figura 1.8.

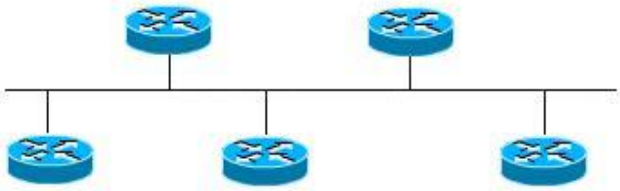


Figura 1.6 Red Multiacceso

Figura 1.7 Tecnología punto a punto

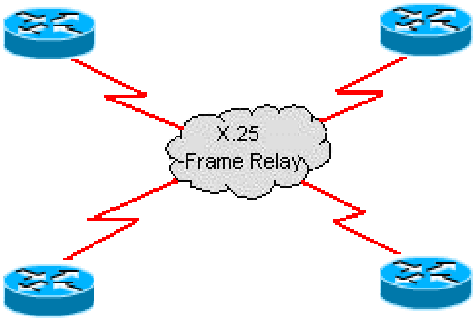


Figura 1.8 Topología NMBA

1.4.2 Funcionamiento de OSPF en una topología multiacceso con difusión

Dado que el enrutamiento OSPF depende del estado del enlace entre dos enrutadores, los enrutadores vecinos deben conocerse entre sí en la red antes de poder intercambiar información. Este proceso se hace por medio del protocolo Hello. El protocolo Hello es el

encargado de establecer y mantener relaciones de vecindad. Garantiza que la comunicación entre vecinos sea bidireccional.

Los paquetes se envían periódicamente desde cada una de las interfaces que participan en OSPF por medio de la dirección IP de multidifusión 224.0.0.5, a la que también se le conoce como dirección ALLSPFEnrutador.

1.4.2.1 Enrutador designado y enrutador de reserva

Los enrutadores de entorno multiacceso, como un segmento de red Ethernet, deberán elegir un DR y un BDR que representen la red. El BDR no lleva a cabo ninguna función DR cuando este último este funcionando. En su lugar, recibe toda la información, pero permite al BDR llevar a cabo tareas de reenvío y sincronización. El BDR solo lleva a cabo tareas DR si el DR falla.

El DR y el BDR añaden valor a la red de las siguientes formas:

- *Reduciendo el tráfico de actualización de enrutamiento*. El DR y el BDR actúan como punto de contacto central para el intercambio de información sobre el estado del enlace en una red multiacceso concreta. Por tanto, cada enrutador debe establecer una adyacencia con el DR y el BDR. El DR representa la red multiacceso en el sentido que envía información sobre el estado de enlace de cada enrutador a todos los demás enrutador de una red multiacceso. Este proceso de inundación reduce significativamente el tráfico relacionado con el enrutador del segmento.
- *Manipulando la sincronización del estado del enlace.* El DR y el BDR aseguran que los demás enrutadores de la red poseen la misma información sobre el estado del enlace acerca de la internetwork. De esta forma se reduce el número de errores de enrutamiento.

La primera vez que aparecen los enrutadores en una red, ejecutan el proceso hello y eligen el DR y el BDR. Los enrutadores luego tratan de formar adyacencias con el DR y el BDR.
Para elegir un DR y un BDR, los enrutadores ven el valor de prioridad de cada uno durante el proceso de intercambio del paquete hello, como se ven en la figura 1.9. Luego utiliza las condiciones siguientes para determinar cual de ellos es elegido.

- El enrutador que tiene un valor de prioridad más alto es el DR (como se puede ver en la figura 1.9).
- El enrutador que tiene el segundo valor de prioridad es el BDR.
- El valor predeterminado de la prioridad OSPF de la interfaz es 1. En caso de empate se usa el ID de enrutador.

- El enrutador que tiene el ID de enrutador (dirección IP de la interfaz) más alto se convierte en el DR, mientras que el enrutador que tiene el segundo ID de enrutador más alto se convierte en BDR.
- Un enrutador con una prioridad de 0 no es elegible para convertirse en un DR o un BDR. Un enrutador que no sea el DR o el BDR se denomina Drother.
- Si un enrutador con valor más alto se añade a la red, el DR y el BDR no cambian. Un DR y un BDR solo cambian si uno de ellos cae. Si el DR es el que cae, el BDR se convierte en el DR, y se elige un nuevo BDR. Si es el BDR el que cae, se elige un nuevo BDR. Para determinar si el DR está caído, el BDR establece un temporizador. Se trata de las características de fiabilidad. Si el BDR no oye que el DR reenvía LSA antes que el temporizador se agote, el BDR presupondrá que el DR está fuera de servicio.

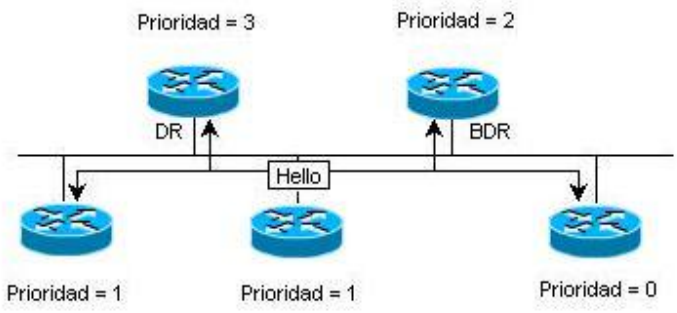


Figura 1.9 DR y BDR

Por consiguiente un enrutador que este conectado con múltiples redes podrá ser un DR en un segmento y un enrutador normal en otro segmento.

1.4.2.2 Inicio de OSPF

En esta sección se aborda los pasos que tienen lugar cuando un enrutador que ejecutan OSPF aparecen en una red.

1.4.2.3 Proceso de intercambio

En el primer paso del inicio de OSPF, el proceso de intercambio se produce con el protocolo Hello, se ve en la figura 1.10. A continuación se explica este proceso de intercambio cuando todos los enrutador están apareciendo en la red a la vez:

Paso 1. El enrutador A está activado en la LAN y está en un estado **caído**, ya que no ha intercambiado información con ningún otro enrutador. Empieza enviando un paquete hello por cada una de las interfaces que participan en OSPF, aunque no conozca la identidad de ningún enrutadores, incluyendo el DR. El paquete hello se envía con la dirección de multidifusión 224.0.0.5.

Paso 2. Todos los enrutadores que ejecutan OSPF reciben el paquete hello del enrutador A y añaden el enrutador A a su lista de vecino. Se trata del estado **INIT**.

Paso 3. Todos los enrutadores que recibieron el paquete envían un paquete hello de respuesta de unidifusión al enrutador A con la información correspondiente, como se ve en el paso 1. El campo del vecino incluye a todos los demás enrutadores vecinos, entre los que se incluye el enrutador A.

Paso 4. Cuando el enrutador A recibe estos paquetes, añade todos los enrutadores que tenían sus ID de enrutador en sus paquetes hello a su propia base de datos de vecindad. Esto se suele denominar estado de **dos vías.** En este punto todos los enrutadores que figuren en su propia lista de vecinos habrán establecido una comunicación bidireccional.

Paso 5. Los enrutadores determinan quién será el DR y el BDR, utilizando el proceso descrito anteriormente. Este proceso debe producirse antes de que los enrutadores puedan empezar a intercambiar información sobre el estado de enlace.

Paso 6. Periódicamente (cada 10 segundos, por defecto), los enrutadores de una red intercambian paquetes hello para asegurar que la comunicación sigue funcionando. Las actualizaciones hello incluyen el DR, el BDR, y la lista de enrutadores cuyos paquetes hello hayan sido recibidos por el enrutador. Recuerde que recibido significa que el enrutador receptor vio su propio ID de enrutador como una de las entradas del paquete hello recibido.

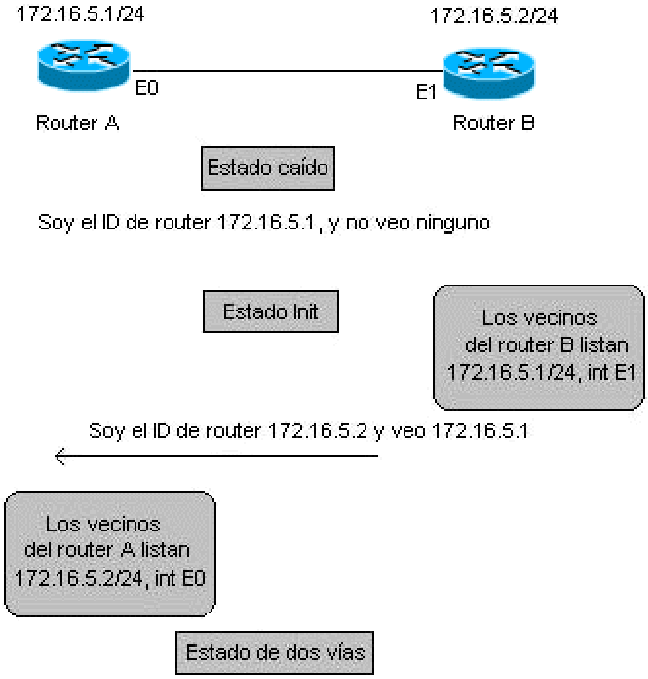


Figura 1.10 Proceso de intercambio

1.4.2.4 Como descubrir rutas

Una vez elegidos el DR y el BDR, se considera que los enrutadores están en estado **exstart** y que están listos para descubrir la información sobre el estado del enlace. El proceso que se utiliza para descubrir las rutas de red se denomina protocolo de intercambio y se lleva a cabo para que los enrutadores entren en un estado de comunicación completo. El primer paso de este protocolo es que el DR y el BDR establezcan adyacencias con cada uno de los demás enrutadores. Cuando los enrutadores adyacentes están en estado completo, no necesitan rehacer el protocolo de intercambio, a menos que cambie el estado completo.
El protocolo de intercambio, que se ilustra en la figura 1.11 funciona así:

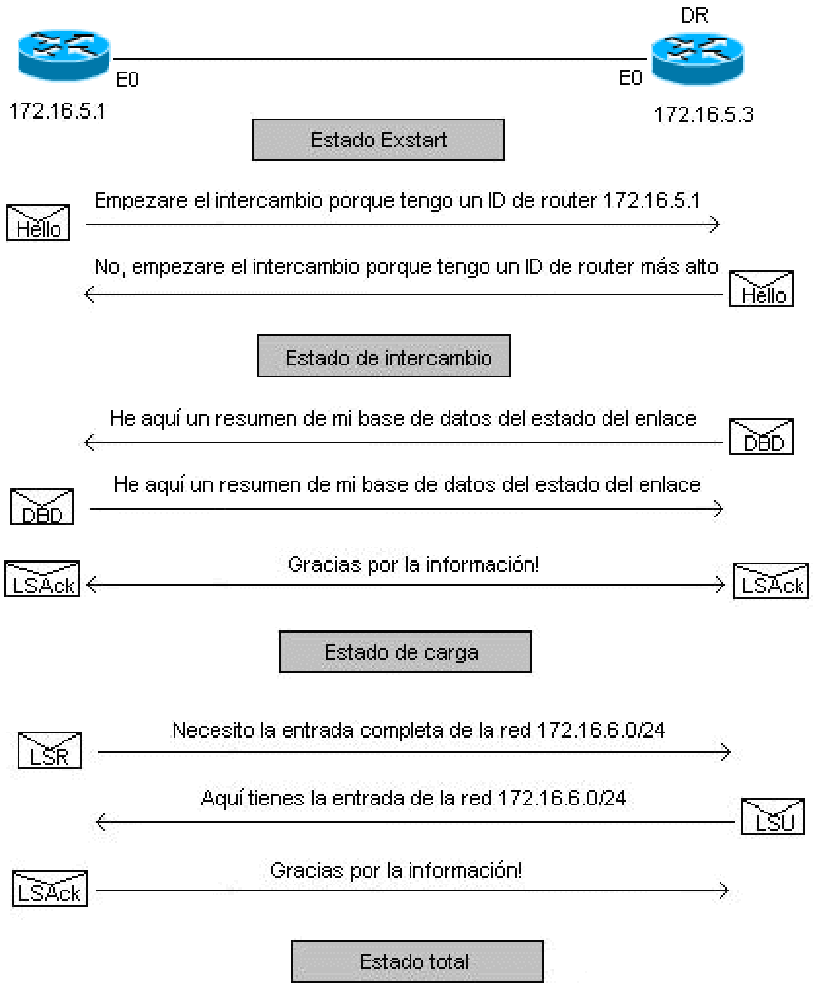


Figura 1.11 Descubrir rutas

Paso 1. En el estado **exstart**, el DR y el BDR establecen adyacencias con cada uno de los enrutadores de la red. Durante este proceso, se crea una relación maestro-esclavo entre cada enrutador y sus DR y BDR adyacentes. El enrutador que tenga el ID de enrutador más alto actúa como maestro. Observe que la información sobre el estado del enlace solo se

intercambia y sincroniza entre el DR y el BDR y los enrutadores con los que hayan establecido adyacencias, ya que el hecho de que el DR represente a la red reduce la cantidad de tráfico de actualización del enrutamiento.

Paso 2. Los enrutadores maestro y esclavo intercambian uno o más paquetes de descripción de base de datos (DBD). Los enrutadores están en el estado de **intercambio.**
Un BDB incluye información de las cabeceras LSA (resumen) de las entradas LSA que aparecen en la base de datos del estado del enlace del enrutador maestro. Las entradas pueden versar sobre un enlace o sobre una red (existen distintos tipos de LSA). Cada cabecera LSA incluye cosas como un tipo de estado de enlace, la dirección del enrutador que publica y el número de secuencia LSA. Este número es la forma en que un enrutador determina la novedad de la información sobre el estado de enlace recibida. El DBD también incluye un número de secuencia DBD que asegura que se reciben todos los DBD en el proceso de sincronización de la base de datos. El maestro define los números de secuencias DBD.

Paso 3. Cuando el enrutador esclavo recibe el DBD, hace lo siguiente:

- Acusa recibo del DBD repitiendo los número de secuencia DBD en un paquete de acuse de recibo de estado del enlace (LSAck)
- Compara la información que hubiera recibido con la información que tenga, comprobando el número de secuencia LSA de la cabecera LSA. Si el DBD tiene una entrada del enlace más actualizada, el enrutador secundario enviará una petición del estado del enlace (LSR) al enrutador principal.
- El enrutador principal responde con la información completa sobre la entrada solicitada en un paquete de actualización del estado de enlace (LSU). Nuevamente, el enrutador secundario envía un LSAck cuando recibe la LSU. El proceso de envío de LSR se suele denominar estado **de carga.**

Paso 4. Todos los enrutadores añaden las nuevas entradas de estado de enlace a su base de datos de estado de enlace.

Paso 5. Cuando se han satisfecho todas las LSR de un determinado enrutador, los enrutadores adyacentes se consideran que están sincronizados y en un estado **completo**. Los enrutadores deben estar en un estado completo antes de poder enrutar el tráfico. En este punto, los enrutadores deben tener bases de datos de estado idénticas.

1.4.2.5 Como elegir rutas

Cuando un enrutador posee una base de datos del estado de enlace completa, esta listo para crear su tabla de enrutamiento, como se ve en la figura 1.12. Recuerde que los protocolos de estado del enlace utilizan una métrica relacionada directamente con el ancho

de banda de los medios. Por ejemplo, una Ethernet de 10-Mbps posee una métrica con un costo inferior que el de una línea de 56Kbps, ya que es más rápida.
Para calcular el costo más bajo a un destino, los protocolos de estado del enlace como OSPF utilizan el algoritmo de Dijkstra, construyendo así su tabla de enrutamiento paso a paso. En términos sencillos, el algoritmo suma los costos totales entre el enrutador local (el raíz) y cada red de destino. Si hay múltiples rutas a un destino, se tomará la de costo inferior. Observe que OSPF conserva hasta seis entradas de ruta de costo equivalente en la tabla de enrutamiento para el equilibrio de la carga.

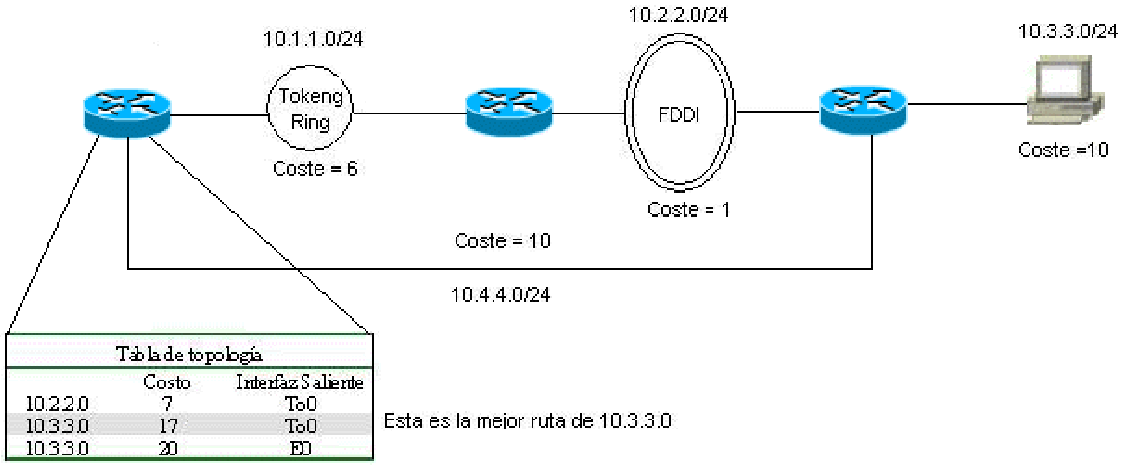


Figura 1.12 Elección de rutas

A veces un enlace, como una línea serie, sube y baja rápidamente (lo que se llama flapping), o un cambio en el estado del enlace puede afectar a otra serie de enlaces. En estas situaciones, se podría generar una serie de LSU, que haría que los enrutadores recalcularan repetidamente una nueva tabla de enrutamiento.

Este proceso puede ser lo suficientemente serio como para que los enrutadores no converjan nunca. Para minimizar el problema, cada vez que se recibe una LSU, el enrutador espera unos instantes para recalcular su tabla de enrutamiento. El valor predeterminado es de 5 segundos.

1.4.2.6 Como mantener información sobre el enrutamiento

En un entorno de enrutamiento por estado de enlace, es muy importante que todas las bases de datos de topología de los enrutadores estén sincronizadas. Cuando hay un cambio en un estado de un enlace, los enrutadores utilizan un proceso de inundación para notificar el cambio a los demás enrutadores de la red, como se ve en la figura 1.13. Los paquetes de actualización del estado del enlace ofrecen el mecanismo para inundar las LSA. El proceso de inundación de un enlace multiacceso es así:

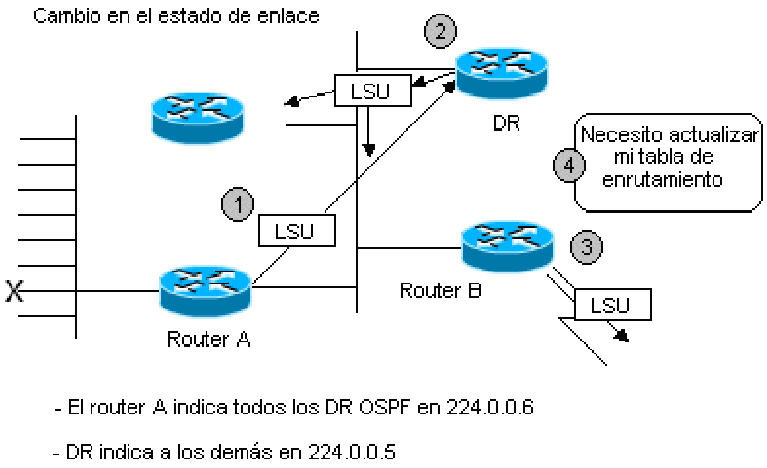


Figura 1.13 Mantener información de enrutamiento

Paso 1. Un enrutador advierte un cambio en un estado del enlace y hace una multidifusión de un paquete LSU que incluye la entrada LSA actualizada, 224.0.0.6 es la dirección de todos los DR y BDR OSPF. Un paquete LSU puede contener varias LSA distintas.

Paso 2. El DR acusa la recepción del cambio e inunda la LSU a otros en la red utilizando la dirección de multidifusión OSPF 224.0.0.5. Para que este procedimiento sea fiable, cada LSA debe ser reconocida por separado. Tras recibir la LSU, cada enrutador responde al DR con un LSAck.

Paso 3. Si se conecta un enrutador con otra red, inundará la LSU en otras redes reenviando la LSU al DR de la red multiacceso, o al enrutador adyacente en una red punto a punto. A su vez, el DR hará una multidifusión de la LSU a los demás enrutadores de la red.

Paso 4. Cuando un enrutador recibe la LSU que incluye la LSA cambiada, el enrutador actualizará su base de datos del estado del enlace. Luego calculará el algoritmo SPF con la nueva base de datos con el fin de agregar una nueva tabla de enrutamiento. Tras unos breves instantes, cambiará a la nueva tabla de enrutamiento. Tenga en cuenta que cada vez que se recibe una LSU, el enrutador espera unos instantes antes de recalcular su tabla de enrutamiento con el fin de reducir los efectos de que las rutas suban y bajen.

1.4.3 Múltiples áreas OSPF

OSPF fue diseñado para permitir separar áreas grandes en áreas más pequeñas y manejables que puedan seguir intercambiando información.
La capacidad de OSPF de separar una gran internetwork en múltiples áreas también se denomina enrutamiento jerárquico. El enrutamiento jerárquico permite separar una

internetwork grande en internetworks más pequeñas que se denominan áreas, como se ve en la figura 1.14.
La topología jerárquica de OSPF posee las ventajas siguientes:

- *Frecuencia reducida de los cálculos SPF.* Dado que la información de ruta detallada se conserva en cada área, no es necesario inundar todos los cambios del estado de enlace en todas las áreas. Por tanto, no todos los enrutadores deben ejecutar el cálculo SPF cuando se produce un cambio. Sólo los que se vean afectados tendrán que recalcular las rutas.
- *Tablas de enrutamiento más pequeñas.* Cuando se usan múltiples áreas, las entradas de ruta detalladas para las redes entre áreas se mantienen en esa área. En vez de publicar estas rutas explícitas fuera del área, estas rutas se pueden resumir en una o más direcciones. La publicación de los resúmenes reduce el número de publicaciones del estado del enlace (LSA) que se propagan entre las áreas, y todas las redes son alcanzables.
- *Estructura reducida de actualización de estado del enlace (LSU).* Las LSU pueden contar con una serie de tipos de LSA, incluyendo información del estado del enlace e información de resumen. En vez de enviar una LSU sobre cada una de las redes de un área, puede publicar una ruta individual o unas cuantas rutas resumidas entre las áreas, reduciendo así la estructura asociada con las actualizaciones del estado de enlace que se pasan a otras área.

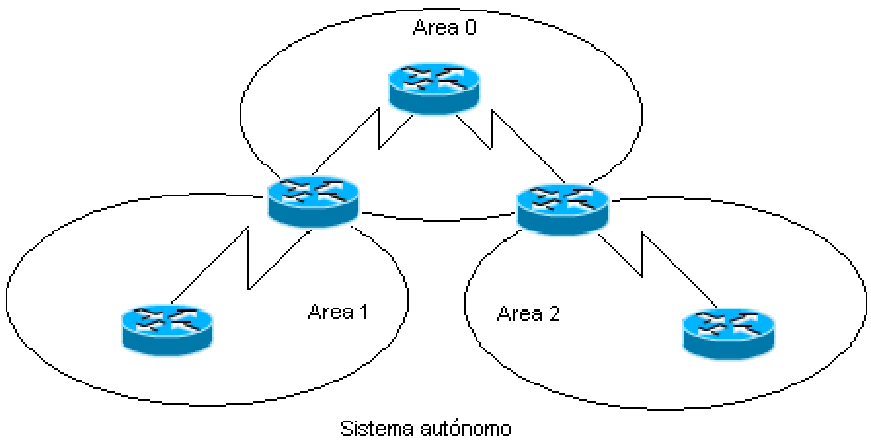


Figura 1.14 Múltiples Areas

Tipos de enrutadores

Distintos tipos de enrutadores OSPF, como se ve en la figura 1.15, controlan de forma diferente el modo en que el tráfico pasa de un área a otra. Los tipos de enrutador son:

- *Enrutador interno*. Los enrutadores que tienen todas las interfaces en la misma área son enrutadores internos y poseen bases de datos del estado de enlace idénticas.

- *Enrutador backbone.* Los enrutadores que permanecen en el área backbone. Tienen al menos una interfaz conectada al área 0. El Área 0 sirve como área de tránsito entre las demás áreas OSPF.
- *Enrutador fronterizo (ABR).* Los enrutadores que tienen interfaces conectadas a múltiples áreas. Estos enrutadores mantienen base de datos del estado del enlace separadas en cada área a la que están conectados, y enrutan el tráfico destinado o procedente de otras áreas.
- *Enrutador límite de sistema autónomo (ASBR).* Los enrutadores que tienen al menos una interfaz con una internetwork externa (otro sistema autónomo).

Un enrutador puede ser de más de un tipo de enrutador. Por ejemplo, si un enrutador se conecta al Área 0 y a Área 1, así como a una red no OSPF, se consideraría un ABR, un ASBR y un enrutador backbone.
Un enrutador posee una base de datos del estado de enlace separada por cada área a la que está conectando. Por tanto, un ABR tendría una base de datos del estado de enlace para el Área 0 y otra la base de datos del estado de enlace para la otra área en la que participa. Dos enrutadores que pertenecen a la misma área tienen, para esa misma área, bases de datos del estado del enlace idénticas.

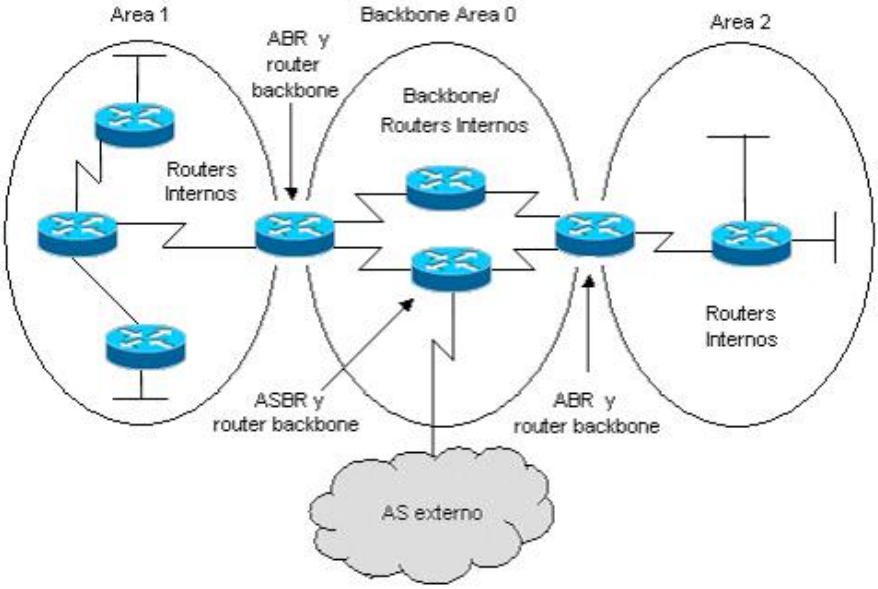


Figura 1.15 OSPF en múltiples áreas

Tipos de área

Las características que asigne a un área controlan el tipo de información de ruta que esta recibe. Los tipos de áreas posibles son los siguientes:

- *Área estándar*. Es un área interna. La cual puede aceptar actualizaciones de enlace (dentro de un área), resúmenes de rutas (entre áreas) y rutas externas.

- *Área backbone (área transito).* Cuando se interconectan múltiples áreas, el área backbone es la entidad central con la que todas las demás áreas se conectan. El área backbone siempre esta marcada como el Área 0. Todas las demás áreas deben conectarse con esta área con el fin de intercambiar y enrutar información. El backbone OSPF posee todas las propiedades de un área OSPF estándar.
- *Área interna.* Hace referencia a un área que no acepta información acerca de las rutas que sean externas al sistema autónomo, como las rutas de orígenes no OSPF. Si los enrutadores tienen que enrutarse con las redes que haya fuera del sistema autónomo, utilizarán una ruta predeterminada. Una ruta predeterminada se indica como 0.0.0.0.

Tipos de publicaciones del estado de enlace

La tabla 1.2 muestra los tipos LSA que se incluyen en una LSU.

Tipo de LSA	**Nombre**	**Descripción**
1	Router LSA	Generadas por cada uno de los enrutadores del área a la que pertenece. Describe los estados del enlace del enrutador en el área. Estos solo se ven inundados en un área determinada. El estado y costo del enlace son dos descriptores que se facilitan.
2	Network LSA	Generadas por los DR en redes multiacceso. Describe el conjunto de enrutadores que están conectados a una determinada red. Estos se ven inundados sólo dentro del área que contiene la red.
3 ó 4	Summary LSA (de resumen)	Originadas por los ABR. Describe los enlaces que hay entre el ABR y los enrutadores internos de un área local. Estas entradas se ven inundadas por el área backbone a los demás ABR. Las LSA de tipo 3 describen los enrutadores de redes que hay en el área local y se envían al área backbone. Las LSA de tipo 4 describen el alcance de los ASBR. Estas entradas de enlace no se inundan por áreas totalmente internas
5	Entrada de enlace externo de sistema autónomo (E1-tipo externo 1 OSPF)(E2-	Originadas por el ASBR. Describe las rutas a los destinos que son externos al sistema autónomo. Se ven inundadas por un sistema

	tipo externo 2 OSPF) (Estados de enlace externos AS)	autónomo OSPF exceptuando en áreas internas, totalmente internas y no tan internas

Tabla 1.2 Tipos de LSA

Funcionamiento de OSPF por múltiples áreas

Antes de repasar cómo los ABR y otros tipos de enrutadores procesan la información sobre las rutas, deberá conocer cómo un paquete entra en múltiples áreas. Por regla general, la ruta que debe seguir un paquete es la siguiente:

- Si el paquete está destinado a una red de un área, será reenviado desde el enrutador interno, a través del área hasta el enrutador interno del destino.
- Si el paquete está destinado a una red que queda fuera del área, deberá pasar por la ruta siguiente:
  - El paquete va desde la red de origen hasta un ABR.
  - El ABR envía el paquete por el área backbone hasta el ABR de la red destino.
  - El ABR de destino reenvía el paquete por el área hasta la red de destino.

Los ABR se encargan de generar información de enrutamiento sobre cada área a la que están conectados y de inundar la información a través del área backbone a las demás áreas con las que estén conectados. El proceso general de inundación es el siguiente.

Paso 1. Se produce el proceso de enrutamiento entre áreas, como se vio anteriormente. Observe que el área debe estar sincronizada antes de que el ABR pueda empezar a enviar LSA de resumen.

Paso 2. El ABR repasa las bases de datos del estado de enlace resultantes y genera LSA de resumen. Por defecto, el ABR envía LSA de resumen a cada red que conoce. Para reducir el número de entradas LSA de resumen, puede configurar el resumen de ruta, de forma que una sola dirección IP puede representar múltiples redes. Para usar el resumen de ruta, las áreas deben usar un direccionamiento jerárquico. Un buen plan de direcciones IP rebaja el número de entradas LSA de resumen que un ABR necesita publicar.

Paso 3. Las LSA de resumen (de los tipos 3 y 4) son colocadas en una LSU y son distribuidas por todas las interfaces ABR que no estén en el área local.

Paso 4. Cuando un ABR o un ABSR recibe LSA de resumen, le añade a su base de datos del estado del enlace y la inunda a su área local. Los enrutadores internos asimilan la información en sus bases de datos.

Cuando todos los tipos de enrutadores reciben las actualizaciones de enrutamiento, deben añadírselas a su base de datos del estado del enlace y recalcular sus tablas de enrutamiento.

1.4.4 Enlaces Virtuales

En algunas situaciones, se añade una nueva área una vez que se ha diseñado y configurado la red OSPF, y no es posible proporcionar a esa nueva área acceso directo al backbone. El enlace virtual proporciona al área desconectada una ruta lógica al backbone.

El enlace virtual tiene dos requisitos que son:

- Debe estar establecido entre dos ABR que compartan un área común.
- Uno de estos dos ABR debe estar conectado al área backbone.

1.5 Interior Gateway Routing Protocol IGRP (Protocolo de enrutamiento de gateway interior)

El éxito del protocolo RIP probó la viabilidad del uso de Vector distancia para calcular las rutas dentro de un sistema autónomo. Desafortunadamente, las limitaciones de RIP se hacían más aparentes cuando existían muchas organizaciones que hacían parte del proceso de enrutamiento. Durante los primeros años de 1980, Cisco System observó una oportunidad en el mercado de mejorar un protocolo vector distancia, que fuera simultáneamente más escalable y con mejores características que RIP.

Cisco respondió a las necesidades del mercado desarrollando el protocolo IGRP.
IGRP fue diseñado para ser tan fácil de usar como RIP, pero sin algunas limitaciones operacionales de RIP.

1.5.1 Visión general de IGRP

IGRP es un protocolo vector distancia diseñado para ser usado dentro de sistemas autónomos. IGRP dio un conjunto de mejores características en comparación con RIP y otros protocolos vector distancia. Una de las más revolucionarias características fue la manera de calcular las distancias. Diferente a protocolos anteriores de vector-distancia que usaban una sola métrica para el cálculo de las rutas, IGRP tiene una serie de métricas, cada una con un amplio rango de valores, permitiendo a los administradores personalizar el cálculo de la ruta

de acuerdo a sus necesidades específicas. Estas métricas, son usadas para calcular una sola ruta compuesta. Esta ruta compuesta es usada para comparar rutas potenciales hacia un destino.
Adicionalmente, IGRP puede soportar múltiples rutas. IGRP puede recordar hasta cuatro rutas hacia un destino dado. Las implicaciones prácticas de está característica se mencionan a continuación:

- Balance de carga a través de dos, tres, o cuatro enlaces.
- Recuperación automática ante fallos en los enlaces.

A pesar del éxito de IGRP alrededor del mundo, IGRP se mantiene como un protocolo propietario. La presencia de enrutadores Cisco pueden hacer que IGRP parezca abierto, pero no lo es. Ningún otro productor a demás de Cisco System lo soporta. Cisco no ha publicado los detalles de los mecanismos internos de IGRP. A pesar de estos obstáculos se examinarán algunas de las principales características de IGRP.

1.5.2 Métricas usadas por IGRP

Una de las áreas en la cual IGRP sobresale es el alto grado de flexibilidad. Esta flexibilidad es ofrecida a través de las métricas de enrutamiento. Diferente a RIP, que solo tiene una sola y estática métrica. IGRP usa seis métricas:

- *Cuenta de saltos*: El número máximo de saltos por defecto es 100, el cual se puede incrementar hasta 255.
- *MTU*: identifica el tamaño del datagrama que un enrutador IGRP aceptará. Este valor no es usado para calcular la única métrica.
- *Ancho de banda del enlace:* Especifica la velocidad de transmisión de un enlace. El ancho de banda usado es el menor ancho de banda que se encuentra en la ruta y cuyo valor se expresa como el inverso del ancho de banda en Kbps multiplicado por 10^7; por ejemplo para un enlace de 64 Kbps el ancho de banda es $10^7/64 = 156250$.
- *Retardo:* mide la cantidad aproximada de tiempo necesitada para atravesar un enlace en la red, asumiendo que el enlace no es usado. El retardo de una ruta en una red IGRP es la suma de los retardos atribuidos a cada interfaz en un enrutador. Esta suma es dividida entre 10 para expresar el resultado en microsegundos. Esta métrica puede tener un valor entre 1 hasta 16.777.215.
- *Carga:* El factor de Carga mide la cantidad de ancho de banda actualmente disponible a través de un enlace dado.
- *Fiabilidad*

No todas las seis métricas son usadas para calcular las rutas. De echo, solo el ancho de banda, el retardo, la carga y la fiabilidad pueden ser usadas para calcular las rutas. Las otras dos, la cuenta de saltos y MTU, facilitan el enrutamiento de otras formas.

La flexibilidad de IGRP va más allá de un incremento del número de métricas. Adicionalmente la flexibilidad es ofrecida en el rango de valores para cada una de las métricas. El administrador puede definir cada una de estas métricas. IGRP también permite establecer valores predeterminados para esas métricas.

Cálculo de la métrica

La métrica de IGRP tiene una longitud de 24 bits y puede tomar un rango de valores de 1 a 16.777.215. Para el cálculo de la ruta, el número más bajo, es la mejor ruta.
En esencia, el valor de la métrica refleja la suma de los retardos y anchos de bandas a través de una ruta dada. En redes construidas de diversos medios (Por ejemplo Ethernet, T1), los beneficios de la comparación matemática de retardos y anchos de banda llegan a ser aparentes.
El cálculo de la métrica esta dado por la siguiente fórmula:

$$\left[K1 * BW + \frac{(K2 * BW)}{(256 - Load)} + K3 * Delay\right] * \left[\frac{K5}{\mathrm{Re}liability + K4}\right]$$

Los valores por defecto de las constantes son: K1 = K3 = 1 y K2 = K4 = K5 = 0. Si K5 es igual a 0, el término [K5/(reliability + K4)] no es usado.

En el ejemplo que se presenta a continuación se realiza el cálculo de la métrica que utiliza el Router1 para llegar a la red 10.0.1.0. La figura 1.16 muestra la topología de red, así como los valores del ancho de banda y retardos.

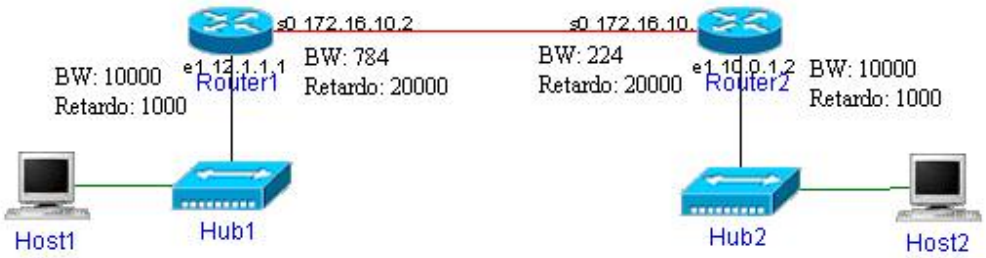


Figura 1.16 Cálculo de la métrica IGRP

La métrica se calcula de la siguiente forma:

- El menor ancho de banda para alcanzar la red es 784.
- Los retardos que se encuentran son: 20000 y 1000.
- Se utilizan los valores por defecto de las constantes, por lo tanto la ecuación se simplifica a: métrica = 10^7/ BW + Retardo/10

- Finalmente el resultado esta dado por: Métrica = BW + Retardo = 10000000/784 + (20000+1000)/10 = **14855**

1.5.3 Mecanismos IGRP

IGRP cuenta con una serie de mecanismos para mantener una internetworking estable. Estos mecanismos están divididos en dos categorías: mecanismos de tiempo y de convergencia, los cuales se estudiarán a continuación.

Mecanismos de tiempo

Como los otros protocolos vector – distancia, IGRP mantiene la integridad de las tablas de enrutamiento mediante el intercambio de información entre los diferentes enrutadores. Cada enrutador envía actualizaciones de su tabla de enrutamiento a sus vecinos durante intervalos fijos. Todas las actualizaciones recibidas automáticamente reemplazan las antiguas rutas de información guardadas en la tabla de enrutamiento.
IGRP cuenta con cuatro temporizadores para mantener la tablas de enrutamiento.

- *Temporizador de actualización (Update Timer):* El tiempo de actualización es usado para iniciar las actualizaciones de la tabla de enrutamiento. El valor por defecto de actualización es de 90 segundos.
- *Temporizador de cuelgue (Hold Timer):* Este temporizador sigue las cantidades de tiempo que los nodos IGRP mantienen las actualizaciones de las tablas de enrutamiento.
- *Temporizador de ruta invalida:* Este tiempo especifica cuanto debe esperar un enrutador en ausencia de mensajes de actualización de una ruta específica, antes de declararla inválida.
- *Temporizador Route-Flush:* Este indica cuanto tiempo deberá pasar antes de que una ruta sea eliminada de la tabla de enrutamiento. El tiempo por defecto es siete veces el periodo de actualización.

Estos mecanismos trabajan de manera muy similar a los mecanismos de RIP.

Mecanismos de convergencia

IGRP incluye varias características diseñadas para reducir los tiempos de convergencia y mejorar la estabilidad de las redes IGRP como:

- Horizonte dividido
- Actualizaciones con poisoned reverse

1.5.4 Operación de IGRP

Los enrutadores convergen a una topología de red común intercambiando lo que ellos conocen acerca de la red. De tal manera los enrutadores periódicamente pasan copias de sus tablas de enrutamiento a sus vecinos inmediatos. Cada receptor adiciona su información a la tabla y envía la tabla modificada a su vecino inmediato. Este proceso ocurre entre enrutadores vecinos.

Cambios en la topología

En cualquier momento en que ocurra un cambio en una red basada en vector-distancia, los enrutadores gradualmente aprenden acerca del impacto. Los enrutadores que se encuentren más cerca del punto del cambio se enteran más rápido de lo sucedido, que otros que se encuentran a mayor distancia. A través de un proceso de intercambio de información, todos los enrutadores de la red gradualmente convergerán unánimente a una nueva topología de red. Esta sección examinará como IGRP mantiene sus tablas de enrutamiento y converge después de cambios de topología.

Supresión de tablas de enrutamiento

IGRP confía en sus cuatro mecanismos de tiempo para mantener la integridad de las rutas. IGRP también puede usar los temporizadores para recobrarse de fallas en la red que impide a los nodos vecinos de pasar actualizaciones a cada uno. Las actualizaciones de las tablas de enrutamiento se inician cada 90 segundos. El tiempo de actualización es usado para seguir esa cantidad de tiempo. Por encima de la expiración de esa cantidad de tiempo, IGRP lanza una serie de paquetes que contienen toda su tabla de enrutamiento. Los paquetes son difundidos a cada vecino. Por consiguiente cada enrutador IGRP deberá recibir una actualización de cada uno de sus vecinos aproximadamente cada 90 segundos. Este proceso es notablemente similar al usado por RIP. La más grande diferencia es la cantidad de tiempo que debe transcurrir entre actualizaciones. RIP envía las actualizaciones cada 30 segundos, un breve periodo de tiempo para que las actualización de tablas puedan arreglar el rendimiento de la red.

Como sucedía con RIP, una falla en la actualización puede ser causada por un número de diferentes eventos. El más simple y fácil de es cuando el paquete que contiene la actualización esta dañado es descartado. Alternativamente, la falla en la actualización podría haber sido como resultado de la capacidad de transmisión o una falla en el enrutador. Estos tipos de cambios cambiarían la topología de la red. Para que la red retorne a la estabilidad, los enrutadores sobrevivientes deben identificar los recursos perdidos y estar de acuerdo con la nueva topología de red.

Debido a la variación entre las posibles causas de una falla en las actualizaciones, es importante tomar el tiempo necesario para entender la causa del problema. Las apropiadas acciones difieren grandemente a lo largo del espectro de fallas. Por ejemplo, si el fallo ocurre porque el paquete fue descartado, actuar muy rápido puede resultar en la invalidación errada de docenas de rutas. De la misma manera, si el fallo se produce debido a que un enrutador llega a ser deshabilitado, no actuar con prontitud resultaría en una notable perdida del servicio. IGRP usa sus cronómetros para invalidar rutas que han fallado.

Identificando Rutas invalidas

Rutas pueden llegar a ser invalidas en una de las dos siguientes formas:

- Una ruta puede expirar
- Una ruta puede ser notificada por otro enrutador como ruta invalida

Independientemente de la causa, los enrutadores necesitan modificar sus tablas para reflejar la indisponibilidad de una ruta dada. Una ruta puede expirar si no recibe una actualización dentro de una cantidad de tiempo especificada, el tiempo por defecto es tres veces el periodo de actualización. El valor por defecto de este periodo es de 90 segundos. Por lo tanto, una ruta llega a ser invalida 270 segundos después no recibir actualizaciones.

Esta información es entonces comunicada a los enrutadores vecinos por medio de las actualizaciones periódicas. Los nodos vecinos que reciben la notificación de ruta inválida usan esta información para actualizar su tabla de enrutamiento.

Una ruta invalida permanece en la tabla de enrutamiento el tiempo suficiente para que los nodos IGRP decidan que hacer. Si la ruta es realmente valida, y los destinos son aun alcanzables, los enrutadores detectarán esto y convergerán. De otra manera, si la ruta es invalida, permanecerá en la tabla de enrutamiento hasta que el tiempo flush expire.

Multirutas

Una de las más importantes características de IGRP es la capacidad de realizar enrutamiento a través de diversos caminos. Diferente a RIP, el cual solo recuerda una sola ruta a un destino dado, IGRP puede recordar hasta cuatro diferentes rutas a un destino. Esto permite balancear el tráfico a través de múltiples rutas, mientras que se protege contra impactos de fallos en los enlaces.

Definición de varianza

La varianza es un atributo modificable, que especifica el porcentaje por el cual el rendimiento de diferentes enlaces puede variar, y aun todavía son considerados caminos viables hacia

un mismo destino. Este atributo se aplica a toda la red IGRP en vez que a enlaces individuales.

El valor por defecto de la varianza es igual a uno (la varianza igual a uno elimina el balance de carga de costo diferente), pero puede ser personalizado de acuerdo a las necesidades. Por ejemplo, si se establece una varianza de dos, enrutadores con métricas hasta dos veces el costo de la mejor métrica pueden ser aceptados, se debe recordar que se puede guardar un máximo de cuatro rutas.

1.6 Enhanced interior gateway routing protocol (EIGRP)

EIGRP es un protocolo original de Cisco que combina las ventajas de los protocolos de enrutamiento por estado de enlace y de vector distancia. Este protocolo híbrido proporciona las características siguientes:

- *Convergencia rápida.* EIGRP utiliza el Diffusing Update Algorithm (DUAL) para conseguir una convergencia rápida. Un enrutador que ejecuta EIGRP almacena rutas de reserva, en la medida de lo posible, para los destinos, de forma que pueda adaptarse rápidamente a las rutas alternativas. Si no hay ruta apropiada o de reserva en la tabla de enrutamiento local, EIGRP consulta a sus vecinos para descubrir una ruta alternativa. Estas rutas se propagan hasta que encuentren una ruta alternativa.
- *Utilización reducida del ancho de banda.* EIGRP no envía actualizaciones periódicas. En su lugar, utiliza actualizaciones parciales cuando cambia la métrica a un destino. Cuando cambia la información de ruta, DUAL envía una actualización sobre ese enlace, en vez de toda la tabla de enrutamiento. Además, la información solo se pasa a los enrutadores que lo requieren, en contraste con el funcionamiento del protocolo de estado del enlace, que envía una actualización de cambio a todos los enrutadores de un área.
- *Soporte de capas de múltiples redes.* EIGRP soporta AppleTalk, IP y Novell Netware, utilizando módulos que dependen del protocolo (PDM).

1.6.1 Terminología EIGRP

A continuación se describen conceptos relacionados con EIGRP:

- *Tabla de vecindad.* cada enrutador EIGRP mantiene una tabla de vecindad que enumera los enrutadores adyacentes. Esta tabla es comparable a la base de datos de vecindad (adyacencia) que utiliza OSPF. Sirve para el mismo fin, para asegurar la comunicación bidireccional entre cada uno de los vecinos directamente conectados. EIGRP mantiene una tabla de vecindad por cada protocolo de red que soporta; es decir como una tabla de vecindad IP, una tabla de vecindad IPX y una tabla de vecindad AppleTalk.

- *Tabla de topología.* Un enrutador EIGRP mantiene una tabla de topología por cada protocolo de red que esta configurado: IP, IPX y AppleTalk. Todas las rutas conocidas a un destino se mantienen en la tabla de topología.
- *Tabla de enrutamiento*. EIGRP elige las mejores rutas a un destino desde la tabla de topología y coloca estas rutas en la tabla de enrutamiento.
- *Sucesor.* Es la ruta principal que se utiliza para llegar a un destino. Los sucesores se mantienen en la tabla de enrutamiento.
- *Sucesor factible.* Las rutas de menor costo hacia un destino forman un conjunto. De ese conjunto, el vecino que tenga publicada una métrica menor que el actual sucesor es considerado como un sucesor factible. Estas rutas son seleccionadas al mismo tiempo que los sucesores, pero son mantenidas en la tabla de topología. Estas tablas pueden mantener múltiples sucesores factibles a un destino.

1.6.2 Funcionamiento de EIGRP

Paquetes EIGRP

EIGRP utiliza cinco tipos de paquetes:

- *Hello.* Los paquetes hello se utilizan para el descubrimiento de redes. Son enviadas como multidifusión y transportan un número de reconocimiento 0.
- *Actualización*. Las actualizaciones se envían para comunicar las rutas que ha utilizado un determinado enrutador para converger. Estas actualizaciones se envían como multidifusión cuando se descubre una nueva ruta y cuando se completa la convergencia. Para sincronizar las tablas de topología, se envían actualizaciones como unidifusión a los vecinos durante la secuencia de inicio de EIGRP. Las actualizaciones se envían de modo fiable.
- *Respuestas.* Este paquete se envía como respuesta a un paquete de consulta. Las respuestas son unidifusión con respecto al origen de la consulta y se envían fiablemente.
- *ACK.* Los ACK se utilizan para confirmar actualizaciones, consultas y respuestas. Los ACK son paquetes Hello enviados como unidifusión y contienen un número de acuse de recibo distinto de cero.

Relación de vecindad EIGRP

El enrutador envía paquetes hello desde las interfaces que están configuradas para EIGRP. La dirección de multidifusión EIGRP que se utiliza es 224.0.0.10 . Cuando un enrutador EIGRP recibe un paquete hello desde un enrutador que pertenece al mismo sistema autónomo, establece una relación de vecindad (adyacencia).

El intervalo de tiempo de los paquetes hello varía en función de los medios. Los paquetes hello se entregan cada 5 segundos en un enlace LAN, como Ethernet, Token Ring y FDI. El

intervalo predeterminado también queda establecido a 5 segundos en los enlaces punto a punto, como el Point to Point Protocol (PPP), el High Level Data Link Control (HDLC), Frame Relay punto a punto, las subinterfaces ATM, y en los circuitos multipunto con ancho de banda mayor de un T1, entre los que se incluyen ISDN, RDSI, SMDS y Frame Relay. Los mensajes Hello se envían con menos frecuencia en enlaces de baja velocidad, como las interfaces seriales multipunto y la ISDN, RSI (BRI). Los hello se generan cada 60 segundos en estos tipos de interfaces.

A través del protocolo Hello, un enrutador EIGRP descubre dinámicamente otros enrutadores directamente conectados con él. La información que se conoce de los vecinos, como la dirección y la interfaz que utilizan los vecinos, se mantiene en la tabla de vecindad. La tabla de vecindad también tiene el tiempo de espera. El tiempo de espera es la cantidad de tiempo que un enrutador considera a un vecino activo, sin recibir un hello o algún otro paquete EIGRP de ese vecino. Los paquetes hello reportan el valor de tiempo de espera.

Si no recibe un paquete antes de la expiración de tiempo de espera, se detecta un cambio en la topología. La adyacencia de vecindad se elimina y se eliminan todas las entradas de tabla de topología conocidas de ese vecino, como si el vecino hubiera enviado una actualización declarando que todas las rutas son inalcanzables. Con esto se activan las rutas para que converjan rápidamente en caso de que una ruta factible alternativa esté disponible. Una ruta se considera pasiva cuando el enrutador no esta llevando a cabo un recálculo de esa ruta. La ruta está activa cuando está experimentando el recálculo.

El tiempo de espera (hold time) se configura por defecto a tres veces el intervalo hello. Por tanto el valor de tiempo de espera determinado es de 15 segundos en interfaces LAN y WAN rápidas, y de 180 segundos en interfaces WAN más lentas.

Es posible que dos enrutadores se conviertan en vecinos EIGRP aunque no coincidan los valores de tiempo hello y hold, eso significa que se puede configurar independientemente el intervalo hello y los valores de tiempo de espera de los distintos enrutadores.

Descubrimiento de la ruta inicial

EIGRP combina en un paso el proceso de descubrir vecinos y conocer las rutas. La figura 1.17 ilustra el proceso:

Paso 1: Un nuevo enrutador (enrutador A) aparece en el enlace y envía un paquete hello a través de todas sus interfaces.

Paso 2: Los enrutadores que reciben el hello en una interfaz (enrutador B) responden con paquetes de actualización que contienen todas las rutas que posee en su tabla de

enrutamiento, excepto los conocido a través de esa interfaz (horizonte dividido). A diferencia de una operación OSPF, el enrutador B no envía un paquete hello de vuelta al enrutador A. En vez de ello el paquete de actualización establece una relación de vecindad entre los dispositivos de comunicaciones. Como tales, estos paquetes de actualización tienen el bit INIT establecido, lo que se trata del proceso de inicialización. Un paquete de actualización contiene información sobre las rutas que conoce el vecino , incluyendo la métrica que un vecino esta publicando para cada destino.

Paso 3: El enrutador A responde a cada vecino con un paquete ACK, indicando que ha recibido la información de actualización.

Paso 4: El enrutador A inserta información sobre paquetes de actualización en su tabla de topología. Esta tabla incluye todos los destinos que publican los enrutadores vecinos (adyacentes). Está organizado de tal forma que se enumera cada destino, junto con todos los vecinos que pueden llegar al destino y su métrica asociada.

Paso 5: El enrutador A intercambia paquetes de actualización con cada uno de sus vecinos.

Paso 6: Al recibir los paquetes de actualización, cada enrutador envía un paquete ACK al enrutador A.

Cuando se reciben todas las actualizaciones, el enrutador esta listo para elegir las rutas principal y duplicada con el fin de mantenerlas en la tabla de topología.

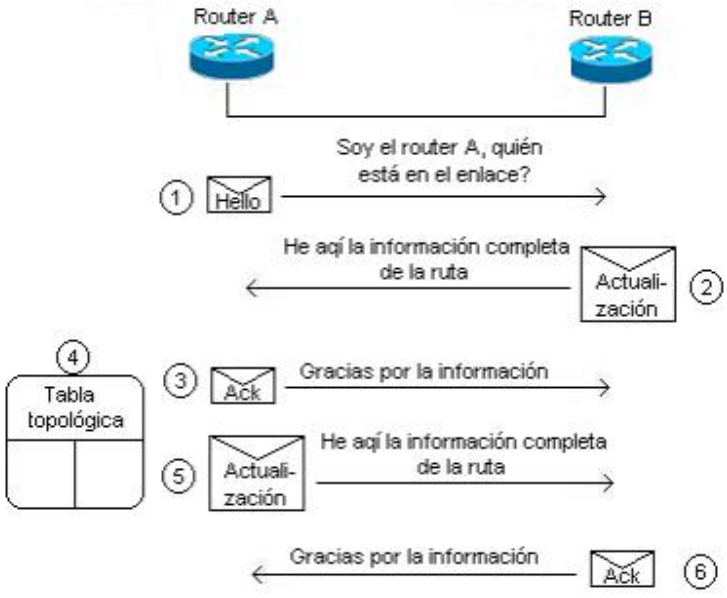


Figura 1.17 Descubrimiento de rutas EIGRP

Selección de ruta

El proceso de selección de ruta EIGRP funciona de forma distinta de los demás protocolos de enrutamiento. Las características clave de la selección de ruta EIGRP son estas:

- EIGRP selecciona las rutas principales y duplicadas y las inyecta en la tabla de topología (hasta seis por destino). Las rutas principales se desplazan a la tabla de enrutamiento.
- La métrica EIGRP es la métrica IGRP multiplicada por 256. El cálculo de la métrica puede usar las 5 variables siguientes:
  - Ancho de banda. El ancho de banda más pequeño entre el origen y el destino.
  - Retraso. El retraso de interfaz acumulativo a lo largo de la ruta.

  Los siguiente criterios aunque están disponibles, no se suelen usar, ya que suelen terminar en cálculos frecuentes de la tabla de topología.
  - Fiabilidad. La menor fiabilidad entre el origen y el destino en base a los mensajes de actividad.
  - Carga. La peor carga entre el origen y el destino en base a los bits por segundo.
  - Unidad máxima de transmisión (MTU). La MTU más pequeña en la ruta.
- EIGRP utiliza DUAL para calcular la mejor ruta a un destino. DUAL selecciona rutas en base a la métrica compuesta y asegura que las rutas seleccionadas no tienen bucles.

Cálculo de la métrica

EIGRP usa el mínimo ancho de banda del camino hacia una red y el retraso total para calcular la métrica. El ancho de banda y los retrasos son configurados en las interfaces del enrutador.
EIGRP usa la siguiente fórmula para determinar el ancho de banda (BW):
BW = (10000000/BW) * 256

EIGRP usa la siguiente fórmula para determinar el retardo (delay):
delay = dealy * 256

EIGRP usa estos valores para determinar la métrica total hacia una red:

$$\text{metrica} = [K1 * BW + \frac{(K2 * BW)}{(256 - \text{load})} + k3 * \text{delay}] * [\frac{K5}{(\text{reliability} + K4)}]$$

Los valores de las constantes K deben ser usados después de una planeación cuidadosa. Valores errados de K pueden causar fallos en la convergencia de las rutas.

Los valores por defecto de K son:
K1 = 1
K2 = 0

K3 = 1
K4 = 0
K5 = 0

Con estos valores se puede simplificar la fórmula así:
métrica = BW + delay

Combinando la fórmula con los factores escalados se tiene:

$$\left[\left(\frac{10^7}{\min imoBW}\right)+\text{suma de delays}\right]\times 256$$

El retardo está medido en microsegundos.

Los enrutadores solo manejan enteros, por lo tanto se necesita aproximar el resultado al entero más cercano para el correcto cálculo de las métricas.

Tabla de enrutamiento y EIGRP Diffusing Update Algorithm (DUAL)

DUAL es la máquina de estado finito que selecciona que información se va a almacenar en la tabla de topología. Como tal Dual engloba el proceso de decisión para todos los cálculos de ruta.
Traza todas las rutas que publican todos sus vecinos. DUAL utiliza la información sobre la distancia, que se conoce como métrica, para seleccionar la ruta más eficiente y sin bucles a cada destino, e inserta esa opción a la tabla de enrutamiento. La ruta de menor costo se calcula sumando el costo entre el enrutador del próximo salto y el destino (que suele llamarse distancia publicada [AD]) y el costo entre el enrutador local y el enrutador del próximo salto (el total se suele denominar distancia factible [FD]). Un sucesor es un enrutador vecino que se usa para el reenvío de paquetes que tiene una ruta de menor costo a un destino que no forma parte de un bucle de enrutamiento. Pueden haber múltiples sucesores si tienen la misma distancia factible. Todos los sucesores se suman a la tabla de enrutamiento. Esta tabla es en esencia un subconjunto de la tabla de topología.
El enrutador del próximo salto de la ruta duplicada se suele denominar sucesor factible (FS). Cuando el enrutador pierde una ruta, observa en la tabla de topología para buscar un FS. Si hay uno disponible, la ruta no irá a un estado activo; en vez de ello, el mejor sucesor factible será promovido como sucesor y será instalado en la tabla de enrutamiento. Cuando no haya sucesores factibles, una ruta entrará en el estado activo, y se producirá el cálculo de ruta.
Para estar cualificado como sucesor factible, un enrutador de próximo salto debe tener una distancia publicada menor que la distancia factible de la ruta del actual sucesor factible. Se puede mantener más de un sucesor factible a la vez.

Cuando no haya sucesores factibles, sino vecinos, publicando el destino, se debe producir una recompilación. A través de este proceso, se determina un nuevo sucesor. La cantidad de tiempo que lleva el recálculo de la ruta afecta al tiempo de convergencia.

Equilibrio de carga

El equilibrio de la carga es la capacidad que tiene un enrutador de distribuir el tráfico por todos sus puertos de red que están a la misma distancia de la dirección de destino. El equilibrio de las cargas aumenta el uso de segmentos de red, con lo que aumenta el ancho de banda efectivo de la red.

El equilibrio de carga se realiza entre un máximo de cuatro rutas de costo equivalente, y se puede configurar el enrutador para que mantenga hasta un máximo de seis rutas igualmente buenas.

EIGRP puede equilibrar el tráfico por múltiples rutas que tengan métricas distintas. El nivel de equilibrado de la carga que se lleva a cabo puede ser controlado a través de un multiplicador, el cual es una varianza, entre 1 y 128. El valor predeterminado es 1, que significa el equilibrio de carga de costo equivalente. El multiplicador define el intervalo de valores de métrica que aceptará para el equilibrio de carga.

1.7 Border Gateway Protocol BGP

BGP es un protocolo de enrutamiento que se utiliza para pasar información entre sistemas autónomos. El objetivo principal de BGP consiste en proporcionar un sistema de enrutamiento entre dominios que garantice el intercambio sin bucles de información de enrutamiento.

BGP es un protocolo de la clase vector distancia pero con muchas mejoras. No requiere de una topología jerárquica. Los enrutadores que ejecutan BGP intercambian datos de conexión a redes llamados vectores de ruta o atributos.

1.7.1 Características de BGP

BGP difiere mucho de su homologo RIP, ya que utiliza el protocolo para el control de la transmisión TCP como protocolo de transporte, lo que proporciona un envío fiable orientado a la conexión, emplea el puerto 179 de TCP. Dos enrutadores que comprenden BGP establecen una conexión TCP entre si e intercambian mensajes para abrir y confirmar los parámetros de conexión. Estos dos enrutadores se denominan enrutadores iguales o vecinos.

Cuando se realiza la conexión se intercambian las tablas de enrutamiento completas. Sin embargo dado que la conexión es fiable los enrutadores BGP solo tienen que enviar los cambios (actualizaciones incrementales), tampoco se requiere que haya actualizaciones de enrutamiento periódicas en un enlace fiable, por lo que se usan actualizaciones activadas. BGP envía mensajes de actividad, que son parecidos a los mensajes hello que envían OSPF y EIGRP, llamados mensajes KeepAlive que son enviados periódicamente para asegurar que la conexión esté viva. Mensajes de notificación son enviados en respuesta de alguna condición de error o por condiciones especiales. Asimismo, si se ha dado una condición de error, luego de que la notificación es enviada, se cierra la conexión.

Los enrutadores BGP intercambian datos de conexión de red, llamados vectores de ruta, compuestos por atributos de ruta, entre los que se incluye una lista de ruta completa (de números de sistema autónomo BGP) que una ruta debe tomar para llegar a una red de destino. Esta ruta se utiliza para construir un grafo de sistemas autónomos que no tenga bucles. La ruta no tiene bucles ya que un enrutador BGP no aceptara una actualización de enrutamiento que ya incluya su número de sistema autónomo en la lista de rutas.

1.7.2 Vecinos BGP

Cualquiera de dos enrutadores que hayan formado una conexión TCP para intercambiar datos de enrutamiento BGP, se denominan iguales o vecinos. Los iguales BGP pueden ser o bien internos o externos al sistema autónomo.

Cuando BGP se esta ejecutando entre enrutadores de un sistema autónomo, a esto se le denomina BGP interno (IBGP). IBGP se ejecuta en un sistema autónomo con el fin de intercambiar información BGP en el sistema autónomo, de forma que pueda pasarse a otros sistemas autónomos.

Cuando BGP esta ejecutándose entre enrutadores de sistemas autónomos distintos, a esto se le llama BGP externo (EBGP), los enrutadores que ejecutan EBGP suelen estar directamente conectados entre si.

1.7.3 Rutas: Anuncios y Almacenamiento

Para propósitos de este protocolo una ruta es definida como una unidad de información que une un destino con los atributos de una ruta a ese destino.

Las rutas son anunciadas entre un par de iguales BGP mediante mensajes de actualización (UPDATE). El destino es el sistema cuyas direcciones IP son reportadas en el campo NLRI (Network Layer Reachability Information (NLRI) y la ruta es la información reportada en los campos atributos de ruta (path attributes) del mismo mensaje de actualización (UPDATE).

Las rutas son almacenadas en los RIBs (Routing Information Bases); existen el Adj-RIBs-In, el Loc-RIB, y el Adj-RIBs-Out. Las rutas que serán anunciadas a los otros iguales BGP deben estar presentes en el Adj-RIB-Out; las rutas que serán usadas por el igual local BGP debe estar presentes en el Loc-RIB, y las rutas que son recibidas de otros iguales BGP están presentes en el Adj-RIBs-In.

Si el igual BGP escoge anunciar la ruta, el puede adicionar o modificar los atributos de ruta antes de anunciarla a su destino(a su igual BGP).

BGP provee mecanismo por los cuales el igual BGP puede informar a su pareja que una ruta previamente anunciada no es factible o su longitud la hace no disponible para usar. Hay 3 métodos para que un igual BGP pueda indicar esta situación:

- El prefijo IP que expresa los destinos de una ruta previamente anunciada puede ser registrada en el campo WITHDRAWN ROUTES (Rutas retiradas) del mensaje de actualización (UPDATE). De esta manera queda marcada como una ruta deshabilitada para uso.
- Un reemplazo de ruta con la misma información del Network Layer Reachability Information puede ser anunciado, o
- La conexión del igual BGP puede ser cerrado, el cual implícitamente remueve del servicio todas las rutas cuyos pares o destinos (origen-destino) se han anunciado uno al otro.

1.7.4 Formatos de los mensajes

Un mensaje es procesado, solamente después que es enteramente recibido. El tamaño máximo del mensaje es 4090 octetos. Todas las implementaciones son requeridas para soportar el tamaño máximo del mensaje. El mensaje más pequeño que puede ser enviado consiste de un encabezado BGP (header) sin la porción de datos o 19 octetos.

Encabezado de los mensajes BGP

Cada mensaje tiene un encabezado de tamaño fijo. Puede haber o no una porción de datos siguiendo el encabezado, dependiendo del tipo de mensaje. El encabezado contiene los siguientes campos:

- *Marker:* este campo de 16 octetos contiene un valor que el receptor del mensaje puede predecir. Si el tipo de mensaje es OPEN, o si el mensaje OPEN no lleva información de autenticación (como un parámetro opcional), entonces el marker deben ser todos 1. De otra manera el valor de marker debe ser predecido por algún cálculo específico como parte del mecanismo de autenticación. El marker puede ser usado para detectar perdidas

de sincronización entre un par de iguales BGP y para autenticar mensajes BGP entrantes.

- *Length:* estos dos octetos indican la longitud total del mensaje incluyendo el encabezado, en octetos.
- *Type:* este octeto indica el tipo de código del mensaje:
  1- OPEN
  2- UPDATE
  3- NOTIFICATION
  4- KEEPALIVE

Formato del mensaje OPEN

Después de que una conexión del protocolo de transporte se establece, el primer mensaje enviado por cada extremo de los iguales BGP es un mensaje OPEN . Si el mensaje OPEN se acepta, un mensaje KEEPALIVE que confirma el OPEN es enviado de regreso. Una vez que el mensaje OPEN es confirmado, se intercambian mensajes de actualización (UPDATE), KEEPALIVE, y notificación (NOTIFICATION).

Además del encabezado BGP (de tamaño fijo) el mensaje OPEN contiene los siguientes campos:

- *Version:* Este campo de un octeto, entero, sin signo, indica la versión del protocolo. La versión actual de BGP es 4.
- *My Autonomous System:* Este es un entero de dos octetos sin signo que indica el número del Sistema Autónomo de quien envía el mensaje.
- *Hold Timer:* Este campo es también un entero de dos octetos sin signo. Indica el número de segundos que el transmisor propone para el valor del Hold Timer. Por el acuso de recibo de un mensaje OPEN, un igual BGP debe calcular el valor del Hold Timer, usando el más pequeño de su Hold Time configurado y del Hold Time recibido en el mensaje OPEN. El valor calculado indica el número máximo de segundos que puede transcurrir entre la recepción de los sucesivos mensajes KEEPALIVE, y / o UPDATE por el transmisor.
- *Identificador BGP:* Este campo contiene un entero sin signo de 4 octetos e indica el identificador BGP del transmisor.
- *Longitud de parámetros opcionales:* Este campo es de un octeto, que indica la longitud total del campo en octetos de los parámetros opcionales. Si el valor de este campo es cero, los parámetros opcionales no están presentes.
- *Parámetros opcionales:* Este campo puede contener una lista de los parámetros, el único parámetro opcional definido actualmente es la autentificación.

Formato del mensaje UPDATE

Los mensajes UPDATE son usados para transferir información de enrutamiento entre iguales BGP. La información en el paquete UPDATE puede ser usada para construir un grafo que describiría las relaciones entre varios Sistema autónomos.

Un mensaje UPDATE se usa para anunciar una ruta factible a un destino o múltiples rutas que no son factibles en el servicio. Un mensaje UPDATE siempre incluye el encabezado BGP (de tamaño fijo), y puede opcionalmente incluir otros campos como:

- *Longitud de rutas no factible:* Es un campo de 2 octetos, esta información indica la longitud total del campo de rutas no factibles (Withdrawn Routes). Un valor de 0 indica que ninguna ruta está siendo retirada del servicio y que el campo WITHDRAWN ROUTES no está presente en el mensaje UPDATE.
- *Campo Withdrawn Routes:* Esta es un campo de longitud variable que contiene una lista de prefijos de direcciones IP para las rutas que están siendo separadas del servicio. Cada prefijo de las direcciones IP está codificado en 2 duplas:

  - Longitud: Es la longitud en bits del prefijo de la dirección IP. Cero indica que el prefijo corresponde a todas las direcciones IP.
  - Prefijo: Este campo contiene prefijos de direcciones IP.

- *Longitud total de los atributos de ruta (Total Path Attribute Length):* dos octetos, indica la longitud total del campo Path Attributes. Un valor de 0, que el NLRI no está presente en el mensaje UPDATE.
- *Path Attributes:* Es una secuencia de longitud variable presente en cada mensaje UPDATE. Este campo incluye información sobre la métrica BGP, y se llama atributos de ruta.

Atributos de ruta BGP

- Un atributo es bien conocido u opcional, obligatorio o discrecional, y transitivo no transitivo. También puede ser parcial.
- No todas las combinaciones de estas características son validas. De hecho los atributos de red se dividen en cuatro categorías separadas:
  1. bien conocidos, obligatorios
  2. bien conocidos, discrecionales
  3. opcionales, transitivos
  4. opcionales, no transitivos
- Solo los atributos transitivos pueden ser marcados como parciales.

Atributos bien conocidos

Son aquellos que todas las implementaciones BGP deben reconocer. Estos atributos se propagan a los vecinos BGP. Un atributo obligatorio bien conocido debe aparecer en la descripción de una ruta, uno discrecional bien conocido, no.

Atributos opcionales

Un atributo opcional no necesita ser soportado por todas las implementaciones BGP; si se soporta puede ser propagado a los vecinos BGP. Un atributo transitivo opcional que no esta implementado en un enrutador debe ser pasado a otros enrutadores BGP. En este caso, el atributo esta marcado como parcial. Un atributo no transitivo opcional debe se eliminado por un enrutador que no haya implementado el atributo.

Atributos BGP definidos

Los atributos que define BGP incluyen los siguientes:

- Atributos bien conocidos obligatorios:
  - ruta de sistema autónomo
  - próximo salto
  - origen
- atributos bien conocidos discrecionales
  - preferencia local
  - agregado atómico
- atributos opcionales transitivos
  - agregador
  - comunidad
- atributo opcional no transitivo
  - discriminador de salida múltiple (MED)

Atributo de ruta de sistema autónomo

Siempre que una actualización de ruta atraviesa un sistema autónomo, el numero de sistema autónomo se antepone a esa actualización. Este atributo es en realidad la lista de números de sistema autónomo que una ruta ha atravesado para llegar a un destino, con el numero de sistema autónomo que origino la ruta al final de la lista. El atributo de ruta de sistema autónomo lo utilizan los enrutadores BGP para asegurar un entorno sin bucles. Los números de sistema autónomo solo van precedidos por los enrutadores que publican rutas de vecinos EBGP. Los enrutadores que publican rutas de vecinos IBGP no cambian el atributo de ruta de sistema autónomo.

Atributo de próximo salto

Indica la dirección IP de próximo salto que hay que usar para llegar a un destino. En lo que respecta a EBGP, el próximo salto es la dirección IP del vecino que envió la actualización. En lo que respecta a IBGP, el protocolo indica que el próximo salto publicado por EBGP debe ser llevado a IBGP.

Atributo de preferencial local

Proporciona una indicación a los enrutadores del sistema autónomo acerca de que ruta es la preferida para salir del sistema autónomo. Es preferible una ruta que tenga una preferencia local mas alta. Esta solo se intercambia entre enrutadores del mismo sistema autónomo.

Atributo MED

MED es una indicación a vecinos externos sobre la ruta preferida a un sistema autónomo. Es una forma dinámica de que un sistema autónomo trate de influir en otro sistema autónomo cual es el camino que debe elegir para alcanzar una determinada ruta, en caso de que haya múltiples puntos de entrada a un sistema autónomo. Lo preferible es un valor de métrica bajo.

A diferencia de la preferencia local, MED se intercambia entre sistemas autónomos. El atributo MED se lleva a un sistema autónomo y se usa ahí, pero no se pasa al siguiente sistema autónomo.

Atributo de origen

Define el origen de la información de ruta. Este atributo puede ser uno de estos tres valores:

- IGP: la ruta es interna al sistema autónomo que la origina.
- EGP: la ruta se conoce a través del protocolo de gateway exterior.
- Incompleto: el origen de la ruta es desconocido, o se conoce a través de otros medios.

Atributo de comunidad

Las comunidades BGP constituyen una forma de filtrar rutas entrantes o salientes. Las comunidades BGP permiten a los enrutadores etiquetar rutas con un indicador (la comunidad) y permitir que otros enrutadores tomen decisiones en base a esta etiqueta. Cualquier enrutador BGP puede etiquetar las rutas de actualizaciones de enrutamiento entrantes o salientes, o cuando hace la redistribución. Cualquier enrutador BGP puede filtrar rutas de actualizaciones entrantes o salientes , o bien seleccionar rutas preferidas, en base a las comunidades.

Las comunidades BGP se usan para los destinos (rutas) que comparten algunas propiedades comunes, por lo que también comparten normas comunes; los enrutadores actúan así en la comunidad, en vez de sobre rutas individuales. Las comunidades no se limitan a una red o a un sistema autónomo y no tienen fronteras físicas.

Formato del mensaje KEEPALIVE

Los mensajes KEEPALIVE son intercambiados entre los iguales BGP lo suficientemente rápido para que el Hold Timer no expire. Estos mensaje no deben ser enviados más de 1 por segundo.

Si el intervalo Hold Time es cero, entonces, los mensajes periódicos KEEPALIVE no deben ser enviados.

Estos mensajes consisten de un encabezado y tiene una longitud de 19 octetos.

Formato del mensaje NOTIFICATION

Un mensaje de NOTIFICATION es enviado cuando un error es detectado. La conexión BGP se cierra inmediatamente después que se envía este mensaje.

El mensaje NOTIFICATION contiene los siguientes campos:

1. Código de error (Error Code)
2. Subcódigo de error

1.7.5 Sincronización BGP

La regla de sincronización BGP establece que un enrutador BGP no debe usar o publicar a un vecino una ruta conocida por IBGP, a menos que esa ruta sea local o conocida desde el IGP. Si el sistema autónomo esta pasando tráfico de un sistema autónomo a otro, BGP no deberá publicar una ruta antes de que todos los enrutadores del sistema autónomo hayan conocido la ruta a través de IGP.

2. TEORIA DE SIMULACIÓN

2.1 Definición

Cuando alguien tiene la responsabilidad de conducir un sistema dado, como por ejemplo: un banco, una ciudad, un sistema de transporte, etc., debe tomar continuamente decisiones acerca de las acciones que ejecutará sobre el sistema. Estas decisiones deben ser tales que la conducta resultante del sistema satisfaga de la mejor manera posible los objetivos planteados.

Para poder decidir correctamente es necesario saber cómo responderá el sistema ante una determinada acción. Esto podría hacerse por experimentación con el sistema mismo; pero factores de costos, seguridad y otros hacen que esta opción generalmente no sea viable. A fin de superar estos inconvenientes, se reemplaza el sistema real por otro sistema que en la mayoría de los casos es una versión simplificada. Este último sistema es el modelo a utilizar para llevar a cabo las experiencias necesarias sin los inconvenientes planteados anteriormente. Al proceso de experimentar con un modelo se denomina simulación. Al proceso de diseñar el plan de experimentación para adoptar la mejor decisión se denomina optimización. Si el plan de experimentación se lleva a cabo con el solo objeto de aprender a conducir el sistema, entonces se denomina entrenamiento o capacitación.

En este punto, es conveniente plantear las siguientes definiciones:

- **Sistema:** Conjunto de objetos o ideas que están interrelacionados entre sí como una unidad para la consecución de un fin (Shannon, 1988). También se puede definir como la porción del Universo que será objeto de la simulación.

- **Modelo:** Un objeto *X* es un modelo del objeto *Y* para el observador *Z*, si *Z* puede emplear *X* para responder cuestiones que le interesan acerca de *Y* (Minsky).

- **Simulación:** Simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con él, con la finalidad de aprender el comportamiento del sistema o de evaluar diversas estrategias para el funcionamiento del sistema (Shannon, 1988).

2.2 Aplicaciones de la simulación

Actualmente la simulación presta un invalorable servicio en casi todas las áreas posibles, algunos de ellos son:

- **Procesos de manufacturas:** Ayuda a detectar cuellos de botellas, a distribuir personal, determinar la política de producción.
- **Plantas industriales:** Brinda información para establecer las condiciones óptimas de operación, y para la elaboración de procedimientos de operación y de emergencias.
- **Sistemas públicos:** Predice la demanda de energía durante las diferentes épocas del año, anticipa el comportamiento del clima, predice la forma de propagación de enfermedades.
- **Sistemas de transportes:** Detecta zonas de posible congestionamiento, zonas con mayor riesgo de accidentes, predice la demanda para cada hora del día.
- **Construcción:** Predice el efecto de los vientos y temblores sobre la estabilidad de los edificios, provee información sobre las condiciones de iluminación y condiciones ambientales en el interior de los mismos, detecta las partes de las estructuras que deben ser reforzadas.
- **Diseño:** Permite la selección adecuada de materiales y formas. Posibilita estudiar la sensibilidad del diseño con respecto a parámetros no controlables.
- **Educación:** Es una excelente herramienta para ayudar a comprender un sistema real debido a que puede expandir, comprimir o detener el tiempo, y además es capaz de brindar información sobre variables que no pueden ser medidas en el sistema real.
- **Capacitación:** Dado que el riesgo y los costos son casi nulos, una persona puede utilizar el simulador para aprender por sí misma utilizando el método más natural para aprender: el de prueba y error.

2.3 Ventajas de la simulación

La simulación es conveniente cuando:

- No existe una formulación matemática analíticamente resoluble. Muchos sistemas reales no pueden ser modelados matemáticamente con las herramientas actualmente disponibles, por ejemplo la conducta de un cliente de un banco.
- Existe una formulación matemática, pero es difícil obtener una solución analítica. Los modelos matemáticos utilizados para modelar un reactor nuclear o una planta química son imposibles de resolver en forma analítica sin realizar serias simplificaciones.
- No existe el sistema real. Es problema del ingeniero que tiene que diseñar un sistema nuevo. El diseño del sistema mejorará notablemente si se cuenta con un modelo adecuado para realizar experimentos.

- Los experimentos son imposibles debido a impedimentos económicos, de seguridad, de calidad o éticos. En este caso el sistema real esta disponible para realizar experimentos, pero la dificultad de los mismos hace que se descarte esta opción. Un ejemplo de esto es la imposibilidad de provocar fallas en un avión real para evaluar la conducta del piloto, tampoco se puede variar el valor de un impuesto a para evaluar la reacción del mercado.
- El sistema evoluciona muy lentamente o muy rápidamente. Un ejemplo de dinámica lenta es el problema de los científicos que estudian la evolución del clima. Ellos deben predecir la conducta futura del clima dadas las condiciones actuales, no pueden esperar a que un tornado arrase una ciudad para luego dar el mensaje de alerta. Por el contrario, existen fenómenos muy rápidos que deben ser simulados para poder observarlos en detalles, por ejemplo una explosión.

2.4 Desventajas de la simulación

- El desarrollo de un modelo puede ser costoso, laborioso y lento.
- Existe la posibilidad de cometer errores. No se debe olvidar que la experimentación se lleva a cabo con un modelo y no con el sistema real; entonces, si el modelo está mal o se cometen errores en su manejo, los resultados también serán incorrectos.
- No se puede conocer el grado de imprecisión de los resultados. Por lo general el modelo se utiliza para experimentar situaciones nunca planteadas en el sistema real, por lo tanto no existe información previa para estimar el grado de correspondencia entre la respuesta del modelo y la del sistema real.

2.5 La simulación en la enseñanza

Las técnicas de simulación se vienen utilizando desde hace mucho tiempo en diversos campos de la enseñanza, como por ejemplo en el entrenamiento de pilotos de avión, o más recientemente el aprendizaje que hacen médicos anestesistas con cuerpos humanos artificiales.

En el campo de la investigación, las técnicas de simulación son ampliamente conocidas y aplicadas. En los años 40, los físicos atómicos introducen este método para calcular el blindaje de plomo que debía utilizarse para frenar los neutrones producidos por la fisión nuclear.

Actualmente, las técnicas de simulación constituyen una herramienta imprescindible para la predicción en las ciencias naturales y sociales, y para la tecnología.

Con la difusión paulatina de la computadora en los centros de enseñanza, se plantea la utilización de Simulaciones para el aprendizaje de las ciencias, lo cual se propone bajo diversas metodologías.

Poco a poco la aplicación de la simulación se va generalizando, en la medida que los propios docentes comprueban en la práctica las ventajas que representa esta innovación pedagógica.

Un software para enseñanza es un material de estudio, como puede ser un libro, un equipo de laboratorio o una guía de problemas. Como cualquier herramienta para el aprendizaje debe estar correctamente integrada al tema de estudio. Para esto es necesario tener en cuenta su contenido conceptual, su estructura, y las actividades que con el realice el alumno.

Los experimentos controlados por computadora responden a una tecnología hoy ampliamente disponible, cuyo aprovechamiento para la enseñanza es conveniente desde muchos puntos de vista. Sin embargo, no se trata sólo de una cuestión tecnológica, sino que supone un definido enfoque pedagógico, en la medida que esta forma de llevar a cabo los experimentos, podría llegar a reemplazar los métodos tradicionales de laboratorio.

Para analizar las implicaciones de esta nueva modalidad para la enseñanza, es necesario considerar previamente las características fundamentales de los métodos computacionales frente a las técnicas tradicionales de experimentación y medición. Estas se pueden sintetizar en:

1. Mayor exactitud en general.
2. Mayor velocidad y / o frecuencia de adquisición.
3. Posibilidad de procesamiento de datos en línea o en forma inmediata.
4. Obtención casi automática de gráficas y resultados numéricos.
5. Menor manipulación directa de los sistemas físicos.
6. Posibilidad de alcanzar una mayor motivación de los alumnos.

La mayor parte de estos aspectos pueden representar ventajas desde el punto de vista didáctico, según la forma de abordar las aplicaciones. La mayor exactitud puede mejorar los resultados, pero también puede hacer ociosa la repetición de experiencias y el análisis estadístico de los errores. La mayor velocidad puede hacer menos tedioso el trabajo pero también puede enmascarar las operaciones realizadas. La obtención automática de gráficos y resultados puede ahorrar mucho trabajo repetitivo, permitiendo que el alumno centre su atención en los aspectos conceptuales, pero también puede automatizar de tal manera la experiencia que el alumno no sepa como definir una escala, construir una gráfica o realizar

un cálculo de errores. La menor manipulación directa de los sistemas físicos puede hacer que el alumno pierda noción de las magnitudes que se miden.

Por el contrario, medir un intervalo de tiempo con cronómetro de mano puede resultar poco preciso, pero facilita la familiarización del alumno con esa magnitud. Sería como un "anclaje" en la realidad. Lo contrario podría ocurrir a través de la medición computarizada, en la medida que no se comprenda lo que hace. Un ejemplo podría ser el del estudio del movimiento de cuerpos con un sistema que mide distancias a través del rebote de ondas ultrasónicas y tiempos por la misma computadora, experiencia que puede tornarse abstracta para el alumno, si no se le explican los fundamentos del método.

Procurando obtener una mejor formación del alumno se podría considerar lo siguiente:

1. Combinar cierta práctica tradicional con la experiencia automatizada, de manera que el alumno adquiera una noción clara de las magnitudes medidas y una capacitación básica de manipulación de instrumentos tradicionales (*anclaje* en conocimientos previos).
2. Instruir al alumno, en forma sintética, sobre los fundamentos de la técnica de adquisición y transferencia de datos entre el sistema y la computadora, de manera que no mire como algo extraño a los elementos electrónicos de conexión, que pueda diferenciar datos analógicos de digitales, y conocer distintos accesos de la computadora para las conexiones exteriores (*Control* del sistema que estudia).
3. Planificar la actividad para que sea el propio alumno quien manipule el sistema de experimentación y la computadora (*Control* del sistema).
4. Aprovechar la velocidad y exactitud de medición de la computadora para realizar una variedad de experiencias que impliquen una mayor complejidad, en la que se modifiquen distintos parámetros y se puedan llegar a conclusiones que serían prácticamente inalcanzables a través de técnicas manuales, tales como el planteo de situaciones límites (*Curiosidad* por conocer aspectos no triviales).
5. Plantear actividades que trasciendan la simple realización de operaciones absolutamente programadas, buscando por el contrario que el alumno pueda tomar decisiones en el desarrollo de la práctica experimental, para lo cual se podrán proponer problemas, preguntas o incógnitas a develar a través de la experimentación (*Desafío* para resolver situaciones problemáticas).
6. Promover el trabajo colaborativo dentro de cada grupo de alumnos, y la comparación de los resultados encontrados entre distintos grupos (*Desafío* a nivel de cierta confrontación).

Por lo anteriormente mencionado, se puede destacar la importancia del desarrollo del presente proyecto con el fin de mejorar la calidad de la enseñanza ofrecida en esta temática en la Facultad de Ingeniería Electrónica y Telecomunicaciones de la Universidad del Cauca.

2.6 Tipos de modelos de simulación

Los modelos de simulación se pueden clasificar en:

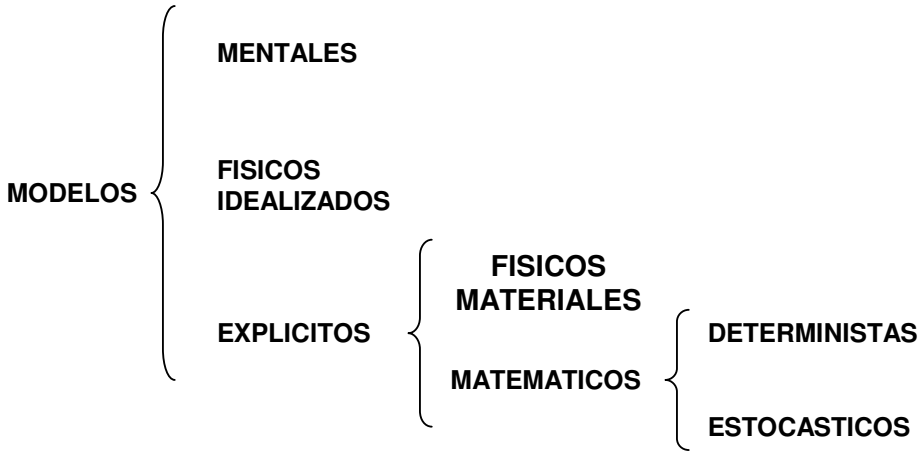


Figura 2.1 Modelos de Simulación

2.6.1 Modelos mentales

Es la imagen o representación mental de un sistema. Se tienen Modelos Mentales de muchos fenómenos físicos: del equilibrio, del movimiento, de la corriente eléctrica, de la energía potencial, etc. También se tienen modelos mentales de los espacios donde habitamos, de los movimientos corporales y de los objetos, de los materiales con que interactuamos, etc.

Cuando se toma la decisión de cruzar o no una calle frente a un vehículo que se aproxima, se esta operando con un modelo mental. No se realizan cálculos matemáticos, sino que se actúa en función de experiencias anteriores. La mente procesa información en la que intervienen distancias, velocidades, características del vehículo, etc.

Ese procesamiento es puramente analógico y se realiza en función de un modelo mental.

2.6.2 Modelos físicos idealizados

Se definen como una representación idealizada del sistema, con la enunciación de los atributos que se tomarán en cuenta y la explicitación de las simplificaciones realizadas.

Cuando se estudia la caída de un vaso, se prescinde de su forma, del material con que está construido, de su color, etc. Solo se representa por una partícula. Luego se desprecia la fuerza resistente del aire, la variación del campo gravitatorio con la altura y las fuerzas inerciales producidas por la rotación terrestre. Llegando así al Modelo Físico Idealizado: una

partícula material con una cierta masa, que se mueve bajo la acción de una sola fuerza constante (el peso). Ahora se pueden aplicar las ecuaciones correspondientes, es decir, formular el modelo explícito, y resolver el problema de acuerdo a sus datos.

2.6.3 Modelos explícitos

Se consideran como una representación operativa del mundo físico. Como tal, es comunicable, estable y bien definido.
Los modelos explícitos más importantes son los modelos físicos materiales y los modelos matemáticos, aunque se pueden dar combinaciones.

Modelos físicos materiales

Es la representación del sistema físico por medio de otro sistema físico, entre los cuales puede haber coincidencia de atributos o simple analogía.

Modelos matemáticos

Se llaman Modelos Matemáticos a la representación de un sistema por medio de ecuaciones matemáticas o distribuciones estadísticas de valores aleatorios.

Dentro de este tipo de modelos se destacan los modelos determinísticos y los estocásticos. Los primeros están formados por ecuaciones que ante un determinado juego de valores de los parámetros iniciales, reproducen siempre la misma solución.

Son los modelos más comunes, tales como el del campo eléctrico o el de la caída libre que ya se mencionó, o el del movimiento planetario. Las ecuaciones diferenciales representan modelos de este tipo. Los segundos son aquellos que se componen por el procesamiento de gran cantidad de datos aleatorios que responden a algún tipo de distribución, no reproduciendo los mismos resultados a partir de idénticos parámetros iniciales.

Para la simulación de los protocolos de enrutamiento se requiere un modelo de este tipo; matemático determinístico, ya que estos protocolos implementan algoritmos matemáticos que deberán ser simulados.

2.7 Etapas de la simulación

En el desarrollo de una simulación se pueden distinguir las siguientes etapas:

2.7.1 Formulación del problema

En este paso debe quedar perfectamente establecido el objeto de la simulación. El cliente y el desarrollador deben acordar lo más detalladamente posible los siguientes factores: los resultados que se esperan del simulador, el plan de experimentación, el tiempo disponible, las variables de interés, el tipo de perturbaciones a estudiar, el tratamiento estadístico de los resultados, la complejidad de la interfaz del simulador, etc. Se debe establecer si el simulador será operado por el usuario o si el usuario sólo recibirá los resultados. Finalmente, se debe establecer si el usuario solicita un trabajo de simulación o un trabajo de optimización.

2.7.2 Formulación del modelo

Esta etapa es un arte y será discutida más adelante. La misma comienza con el desarrollo de un modelo simple que captura los aspectos relevantes del sistema real. Los aspectos relevantes del sistema real dependen de la formulación del problema; para un ingeniero de seguridad los aspectos relevantes de un automóvil son diferentes de los aspectos considerados por un ingeniero mecánico para el mismo sistema. Este modelo simple se irá enriqueciendo como resultado de varias iteraciones.

2.7.3 Colección de datos

La naturaleza y cantidad de datos necesarios están determinadas por la formulación del problema y del modelo. Los datos pueden ser provistos por registros históricos, experimentos de laboratorios o mediciones realizadas en el sistema real. Los mismos deberán ser procesados adecuadamente para darles el formato exigido por el modelo.

2.7.4 Implementación del modelo en la computadora

El modelo es implementado utilizando algún lenguaje de computación, para esto se hará uso de la metodología RUP.

2.7.5 Validación

En esta etapa se comprueba la exactitud del modelo desarrollado. Esto se lleva a cabo comparando las predicciones del modelo con: mediciones realizadas en el sistema real, datos históricos o datos de sistemas similares. Como resultado de esta etapa puede surgir la necesidad de modificar el modelo o recolectar datos adicionales.

2.7.6 Documentación

Incluye la elaboración de la documentación técnica y manuales de uso. La documentación técnica debe contar con una descripción detallada del modelo y de los datos; también, se debe incluir la evolución histórica de las distintas etapas del desarrollo. Esta documentación será de utilidad para el posterior perfeccionamiento del simulador.

2.8 Redes de Petri

Las redes de Petri representan una de muchas alternativas para modelar sistemas, sus características hacen que se acoplen a las necesidades del presente proyecto, motivo por el cual, fue escogida como la herramienta de modelación, dichas características y su operación serán estudiadas en la siguiente sección.

2.8.1 Introducción a las redes

Las PN como ahora conoceremos a las redes de Petri (Petri Net) fueron inventadas por el alemán Karl Adam Petri en 1962. En su tesis doctoral "kommunikation mit automaten" (Comunicación con autómatas), establece los fundamentos para el desarrollo teórico de los conceptos básicos de las PN.

Las PN son consideradas una herramienta para el estudio de los sistemas. Con su ayuda podemos modelar el comportamiento y la estructura de un sistema, y llevar el modelo a condiciones límite, que en un sistema real son difíciles de lograr o muy costosas.
La teoría de PN ha llegado a ser reconocida como una metodología establecida en la literatura de la robótica para modelar los sistemas de manufactura flexibles.
Comparada con otros modelos de comportamiento dinámico gráficos, como los diagramas de las máquinas de estados finitos, las PN ofrecen una forma de expresar procesos que requieren sincronía. Y quizás lo más importante es que las PN pueden ser analizadas de manera formal y obtener información del comportamiento dinámico del sistema modelado.

Una red de Petri es un grafo dirigido bipartito, con un estado inicial, llamado marcación inicial. Los dos componentes principales de la red de Petri son los sitios (pleces) (también conocidos como estados) y las transiciones. Gráficamente, los sitios son dibujados como círculos y las transiciones como barras o rectángulos. Las aristas del grafo son conocidas como *arcos*. Estos tienen un peso específico, el cual es indicado por un número entero positivo, y van de sitio a transición y viceversa. Por simplicidad, el peso de los arcos no se indica cuando éste es igual a 1. Un arco que esté etiquetado con *k* puede ser interpretado como *k* arcos paralelos.

El estado del sistema que la red esté modelando es representado con la asignación de enteros no-negativos a los sitios. Esta asignación es conocida como una marcación, la cual es representada gráficamente mediante unos pequeños círculos negros dentro de un sitio *p*,

llamados tokens . Si el número de tokens es demasiado grande, los *k* tokens son representados con un número no-negativo dentro del correspondiente sitio.

Típicamente, los estados representan algún tipo de condición en el sistema, y una transición representa un evento. Un sitio de entrada (salida) a una transición representan las pre- (post-) condiciones. Los tokens pueden tener muchas interpretaciones. Por ejemplo, cuando un sitio está marcado con un token, este puede representar que la correspondiente condición es verdadera. En otros casos, *k* tokens pueden representar *k* recursos, por ejemplo, el número de *c*licks del mouse realizados. Debido a que las redes de Petri pueden modelar muchos tipos de sistemas, lo que los sitios, transiciones y tokens representen varía enormemente.

2.8.2 Disparo y habilitación de las transiciones

Los cambios en los estados de un sistema son modelados mediante las reglas de activación y habilitación. Las reglas se describen de la siguiente manera:
1. Una transición *t* está habilitada con una marcación *M* si cada sitio de entrada *p* está marcado con al menos *W(p,t)* tokens.
2. Una transición puede o no ser disparada al habilitársele. Cuando más de una transición es habilitada, alguna de esas transiciones es seleccionada de manera no-determínistica dependiendo del modelo empleado.

3. Un disparo de una transición *t* resulta en *W(p,t)* tokens eliminados de cada sitio de entrada *p* de *t* y la adición de *W(t,p')* tokens a cada sitio de salida *p'*.

Si *t* no tiene estados de entrada, se trata de una transición fuente y está habilitada. Si *t* no tiene sitios de salida, se dice que esta es una transición sumergida . Una transición sumergida ``consume'' tokens, pero no produce ningún token.
Si una transición *t* es habilitada bajo la marcación *M*, y *M'* es la marcación resultante del disparo de *t*, se representa como $M \xrightarrow{t} M'$.
Debe hacerse notar el hecho de que conforme las transiciones son disparadas, el número total de tokens distribuidos a lo largo de la red puede variar, esto es, la conservación de los tokens no siempre sucede.

En la actualidad existen muchas extensiones del modelo de las redes de Petri como fueron concebidas originalmente, esto con el fin de adaptarlas para modelar gran cantidad de sistemas, entre las principales variaciones de las redes de Petri tenemos:

- Redes de Petri Temporizadas
- Redes de Petri orientadas a objetos
- Redes de Petri Coloreadas

Por las características que presentan las redes de Petri coloreadas, las cuales serán estudiadas en la próxima sección, se adoptaron para modelar los protocolos de enrutamiento a simular.

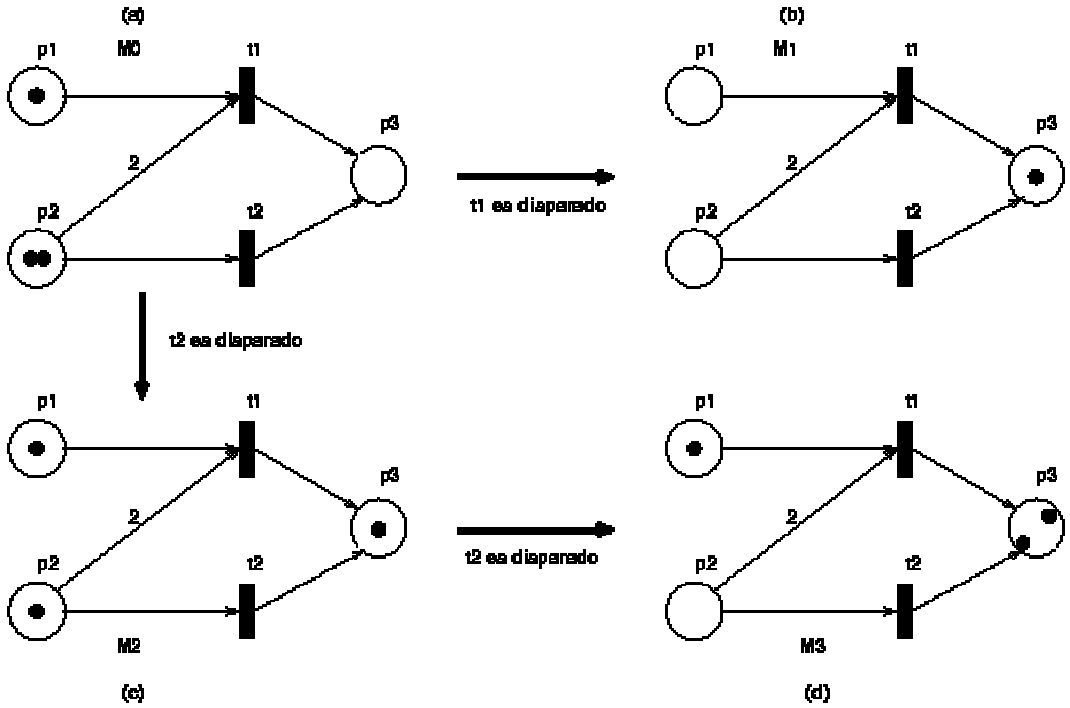


Figura 2.2 Disparo y habilitación de las transiciones en las redes de Petri

2.8.3 Redes de Petri Coloreadas (CPN)

Un modelo CPN de un sistema describe los estados en los cuales el sistema puede estar y las transiciones entre estos estados. Las CPN combinan la fuerza de las redes de Petri con la fuerza de los lenguajes de programación. Las redes de Petri proveen las primitivas para describir sincronización y procesos concurrentes mientras que el lenguaje de programación provee las primitivas para definir los tipos de datos y los valores de los datos manipulados.

Los modelos CPN pueden estar estructurados dentro de un numero de módulos, esto es importante para cuando se modelan sistemas grandes. El concepto de módulo de la CPN esta basado sobre un mecanismo de estructuramiento jerárquico, el cual soporta un estilo de trabajo top-down. Nuevos módulos pueden ser creados desde módulos existentes. Las CPN incluyen el concepto de tiempo, lo cual hace posible capturar el tiempo llevado por las diferentes actividades del sistema.

2.8.3.1 Modelamiento de los estados en las CPN

Tipo (Types): los sitios (places) tienen un type asociado determinando la clase de datos que el sitio puede contener, son similares a los tipos de datos que se definen en un lenguaje de programación.

Marcación (Marking): Un estado en una CPN es llamado un marking, este consiste de un número de tokens posicionados (distribuidos) sobre los places individuales. Cada token lleva un valor el cual pertenece al type del place sobre el cual el token reside. Los tokens que estan presentes sobre un place en particular son llamados el marking de ese place. El marking de un place es en general un multi-set de valores de tokens. Esto significa que un lugar (place) puede tener varios tokens con el mismo valor de token.

Marcación Inicial (Initial marking): es usado para describir el estado inicial de un sistema. Este por convención esta escrito arriba (derecha o izquierda) del place.

2.8.3.2 Modelamiento de las transiciones (acciones)

Transiciones: las acciones en una CPN como en todos los modelos por redes de Petri están representadas por las transiciones. El nombre de las transiciones se escribe dentro de los rectángulos que las representan.

Arcos y expresiones de arcos (arc expresion): las transiciones y los places están unidos por arcos. Una ocurrencia de una transición remueve tokens desde los lugares (places) conectados a los arcos entrantes y suma tokens a los lugares conectados a los arcos de salida. El número exacto de tokens sumados y removidos por la ocurrencia de una transición y los valores de datos son determinados por las arc expresión.

2.8.3.3 CPN jerárquicas

La idea básica de una CPN jerárquica es permitir el modelado para la construcción de un modelo grande usando un número de pequeñas redes CPN, estas pequeñas redes son llamadas páginas. Estas páginas deben estar relacionadas unas con otras de una forma bien definida.

En una CPN jerárquica es posible relacionar una transición (y sus arcos y lugares) a una CPN separada.

En el nivel 1 (nivel mas abstracto) de una CPN jerárquica se debe dar una descripción simple de la actividad modelada, sin tener que dar detalles internos sobre como esto es llevado a cabo, en un nivel 2 se debe especificar mas detalladamente el comportamiento y así sucesivamente se van teniendo niveles de acuerdo a la complejidad del sistema a modelar.

Transiciones sustitutas y subpáginas (sustitution transition y subpages): cada transición que esta marcada con una etiqueta HS, en la esquina superior izquierda, indica que es una sustitution transition . La inscripción siguiente a la HS es llamada “hierarchy inscriptions” y define la subpágina que tiene los detalles de la sustitution transition.

Port y socket places: cada subpágina tiene un número de places los cuales están marcados con una etiqueta In, Out o I/O, estos places son llamados port places y ellos constituyen la interfaz a través de la cual la subpágina se comunica con sus alrededores. Para especificar la relación existente entre una sustitution transition y su subpágina se debe describir como los port places de la subpágina están relacionados con los socket places de la sustitution transition. Los port y socket places siempre tienen idéntico marking.

Fusion places: esto permite el modelamiento para especificar que un conjunto de places son considerados idénticos. Cuando un token es sumado o removido de uno de los places un token idéntico será sumado o removido de todos los places en el fusion set.

2.8.3.4 CPN temporizadas

En esta sección se introduce el concepto de tiempo en las CPN’s. Esta basado en la inserción de un reloj global, los valores del reloj representan el tiempo modelado y ellos pueden ser o enteros o reales (continuo). Además del valor del token, se permite a cada token llevar un valor de tiempo, el cual es llamado time stamp. El time stamp describe el tiempo mas temprano en el cual el token puede ser usado.

Para modelar una actividad que toma n unidades de tiempo, la transición correspondiente crea time stamp para sus tokens de salida que son n unidades de tiempo mas grandes que el valor del reloj en el cual la transición ocurre. Esto implica que los tokens producidos estan inhabilitados n unidades de tiempo.

2.9 Proceso unificado para el desarrollo de programas (RUP)

Para la construcción de la herramienta software que se pretende desarrollar con el presente proyecto se adoptará el proceso unificado para el desarrollo de programas que es un proceso de ingeniería de software. Provee un enfoque ordenado para asignar tareas y responsabilidades dentro de una organización de desarrollo. Su virtud principal es asegurar la producción de software de alta calidad, apropiado a las necesidades del usuario final, dentro de un cronograma y un presupuesto predecibles.

2.9.1 Características principales

- Guiado / Manejado por casos de uso: La razón de ser de un sistema software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.

- Centrado en arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.

- Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un miniproyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.

- Desarrollo basado en componentes: La creación de sistemas intensivos en software requiere dividir el sistema en componentes con interfaces bien definidas, que posteriormente serán ensamblados para generar el sistema. Esta característica en un proceso de desarrollo permite que el sistema se vaya creando a medida que se obtienen o que se desarrollan y maduran sus componentes.

- Utilización de un único lenguaje de modelamiento: UML es adoptado como único lenguaje de modelamiento para el desarrollo de todos los modelos.

- Proceso Integrado: Se establece una estructura que abarque los ciclos, fases, flujos de trabajo, mitigación de riesgos, control de calidad, gestión del proyecto y control de configuración; el proceso unificado establece una estructura que integra todas estas facetas. Además esta estructura cubre a los vendedores y desarrolladores de herramientas para soportar la automatización del proceso, soportar flujos individuales de trabajo, para construir los diferentes modelos e integrar el trabajo a través del ciclo de vida y a través de todos los modelos.

2.9.2 Fases de la metodología

Como se mencionó anteriormente el proceso unificado para el desarrollo de programas (RUP) divide el proceso de desarrollo en ciclos, teniendo un producto al final de cada ciclo. Cada ciclo se divide en cuatro Fases, las cuales son:

Fase 1 : Preparación Inicial

Su objetivo principal es establecer los objetivos para el ciclo de vida del producto. En esta fase se establece el caso del negocio con el fin de delimitar el alcance del sistema, saber qué se cubrirá y delimitar el alcance del proyecto.

Fase 2 : Preparación Detallada

Su objetivo principal es plantear la arquitectura para el ciclo de vida del producto. En esta fase se realiza la captura de la mayor parte de los requerimientos funcionales, manejando los riesgos que interfieran con los objetivos del sistema, acumulando la información necesaria para el plan de construcción y obteniendo suficiente información para hacer realizable el caso del negocio.

Fase 3: Construcción

Su objetivo principal es alcanzar la capacidad operacional del producto. En esta fase a través de sucesivas iteraciones e incrementos se desarrolla un producto software, listo para operar, éste es frecuentemente llamado versión beta.

Fase 4 :Transición

Su objetivo principal es realizar la entrega del producto operando, una vez realizadas las pruebas de aceptación por un grupo especial de usuarios y habiendo efectuado los ajustes y correcciones que sean requeridos.

3. DISEÑO E IMPLEMENTACION DE LA HERRAMIENTA SOFTWARE

Se va a mostrar en este capítulo mostrar los resultados obtenidos en cada una de las etapas del proceso de simulación que se especificaron en el capítulo anterior.

3.1 Formulación de problema

Para el desarrollo de la herramienta de simulación se tuvieron presentes los aspectos que se mencionan a continuación:

- **Resultados esperados**: De la herramienta se espera que simule de manera detallada el comportamiento de los protocolos de enrutamiento, es decir, que se muestre el intercambio de mensajes de los protocolos, la tablas de enrutamiento una vez la red se estabilice, los eventos que se producen al caerse un enlace; para esto el usuario deberá tener la facilidad de implementar las redes en las que desea ver en funcionamiento los protocolos y de configurar los dispositivos que conforman dichas redes. Como información adicional la herramienta debe presentar al usuario una ayuda de las funcionalidades y modo de uso de la misma, así como información teórica sobre los protocolos de enrutamiento que implementa y algunos laboratorios que faciliten el entendimiento de los protocolos y muestren características especiales del sistema.
- **Tiempo disponible**: para el análisis, diseño e implementación de la herramienta se cuenta con un periodo de 12 meses el cual finaliza en el mes de noviembre del año 2003.
- **Variables de interés**: De la herramienta se desea obtener las tablas de enrutamiento para una red (implementada por el usuario) ejecutando un determinado protocolo, no esta dentro de los requerimientos del presente sistema tener en cuenta los tiempos de transmisión de mensajes entre diferentes dispositivos de la red, las perturbaciones que el medio de transmisión introduce a las tramas, métodos para corrección de errores, tiempo de procesamiento de mensajes dentro de los dispositivos, ni simulación simultanea de diferentes protocolos.
- **Interfaz de usuario**: La interfaz de la herramienta debe ser sencilla y fácil de utilizar, se debe exigir al usuario una mínima intervención, es decir, el usuario podrá manipular la herramienta a través del ratón (mouse) sin necesidad de introducir complejos comandos por teclado.

3.2 Formulación del modelo

3.2.1 Restricciones y aproximaciones del modelo

Después de estudiar y analizar los protocolos de enrutamiento y buscando una menor complejidad en el desarrollo de los modelos se hicieron algunas aproximaciones y restricciones que de ninguna manera afectan los requerimientos generales del sistema.

Las restricciones y aproximaciones se presentan a continuación:

Protocolo OSPF

- El comportamiento del protocolo en redes STUB se hará como si se tratará de una red Broadcast.
- En una red Brodcast la interfaz que sea nombrada como Enrutador Designado (DR) no se podrá detener.
- En el modelo no se tendrán en cuenta los enlaces virtuales que maneja OSPF.
- Se generará un modelo que permita manejar un solo enrutador de frontera por área.

Protocolo BGP4

- Para la construcción del modelo se tendrán en cuenta los atributos bien conocidos obligatorios: ruta de sistema autónomo, próximo salto, origen y el atributo opcional no transitivo discriminador de salida múltiple MET.
- La comparación del atributo MET se hará para todas las rutas hacia una red simulando el comando *always-compare-met* de los enrutadores físicos.
- El proceso de decisión para múltiples rutas hacia una red se hará de la siguiente forma: prevalece la ruta que ha sido originada por el enrutador localmente, si el enrutador no ha originado ninguna ruta se tomará aquella que tenga la ruta de sistema autónomo más corta, si persiste la igualdad se tomará la ruta que tenga menor métrica (MET) y si aún persiste se dejará la ruta que haya sido aprendida primero. Esto es una aproximación al proceso de decisión de los enrutadores físicos que consta de más pasos.

3.2.2 Generación del modelo.

El modelado de los protocolos de enrutamiento se realizó con las redes de Petri Coloreadas (CPN) por las características y ventajas que fueron explicadas en el capítulo anterior. Estos modelos fueron la base para la programación del comportamiento de los protocolos.

A continuación se presenta el modelo con redes de Petri coloreadas para el protocolo RIP. El modelado de los otros protocolos se encuentran en el anexo B.

Modelado del protocolo RIP usando las CPN

En la figura 3.1 se presenta el diagrama básico del protocolo RIP, el cual presenta dos estados, Transmisor y Receptor. El primero de ellos modela el protocolo para la transmisión de datos hacia otros enrutadores que estén corriendo RIP, el segundo estado modela la recepción y el tratamiento de dichos datos. Ambos estados acceden a un buffer que contiene la información necesaria para la realización de sus funciones.

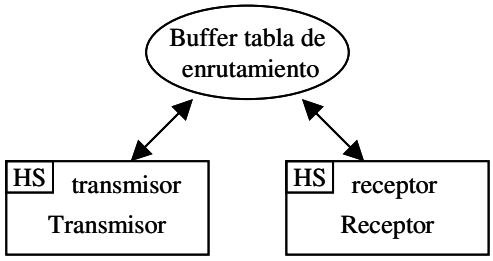


Figura 3.1 Modelo básico

El estado Transmisor el cual de acuerdo a la terminología de CPN es una transición sustituta (*sustitution transition)*, esta detallado en la subpágina transmisor que se presenta en la figura 3.2.

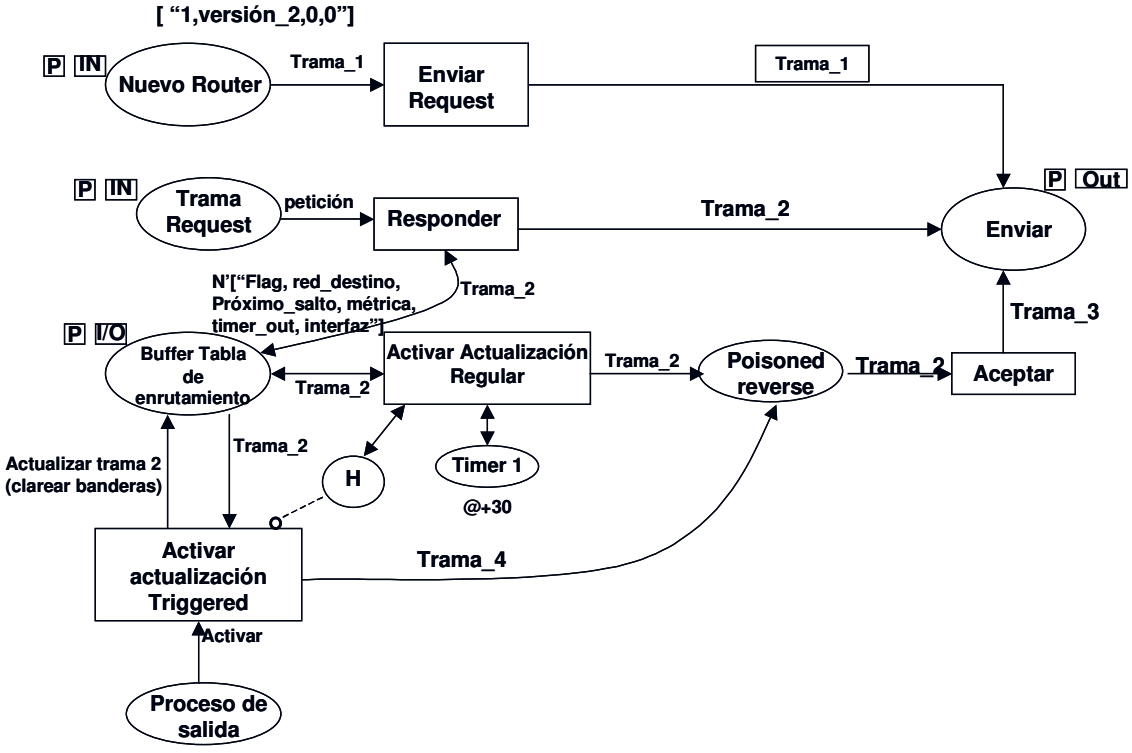


Figura 3.2 Subpágina transmisor

El transmisor esta modelado por 7 estados y 5 transiciones. Los cuales se describirán a continuación:

Estado	Nuevo Enrutador
Transición	Enviar Request
Nuevo estado	Enviar
Descripción	Una vez se ponga en marcha el protocolo por parte del usuario un token del tipo tramaRequest aparecerá en el estado *Nuevo Enrutador* habilitando la transición *Enviar Request*. Esta transición estará encargada de enviar tramas request con el fin de solicitar la tabla de enrutamiento completa de los vecinos y poder actualizar su propia tabla de enrutamiento. Este proceso se realiza solamente una vez durante la inicialización del protocolo.

Estado	Trama Request
Transición	Responder
Nuevo estado	Enviar
Descripción	Cuando llega una trama Request al enrutador, este debe responder con su tabla de enrutamiento completa en una trama Response, este proceso es modelado por la transición *Responder* la cual accede al buffer *tabla de enrutamiento* para obtener los datos requeridos, arma la trama y la pone en el estado *Enviar*.

Estado	Timer 1
Transición	Actualización Regular
Nuevo estado	Poisoned Reverse
Descripción	El protocolo Rip cada 30 segundos debe enviar una trama de actualización que contiene toda la tabla de enrutamiento. Este procedimiento esta modelado por el estado *Timer 1* el cual activa la transición *Actualización Regular* que accede al buffer de la tabla de enrutamiento para obtener los datos requeridos y pasarlos al estado *Poisoned Reversed*.

Estado	Poisoned Reverse
Transición	Aceptar
Nuevo estado	Enviar
Descripción	Para evitar las mallas de enrutamiento, el protocolo Rip implementa el poisoned Reverse el cual consiste en no enviar la información aprendida de un vecino al mismo vecino. Este proceso es llevado a cabo en la transición *Aceptar* y los resultados son llevados al estado *Enviar*.

Estado	Proceso de salida
Transición	Actualización Triggered
Nuevo estado	Poisoned Reverse
Descripción	Otro mecanismo utilizado por Rip para evitar las mallas de enrutamiento son las actualizaciones activadas o triggered, así una vez llega una actualización, un token aparecerá en el estado *Proceso de Salida* activando la transición *Activar* A*ctualización Triggered* enviando los datos hacia el estado *Poisoned Reverse*.

El Receptor esta modelado en la subpágina receptor que se presenta en la figura 3.3. Este consta de 8 estados y 5 transiciones, dos de las cuales son transiciones sustitutas: Actualizar Buffer y Temporizar, y están detalladas en las subpáginas actualizar buffer y temporizar respectivamente.

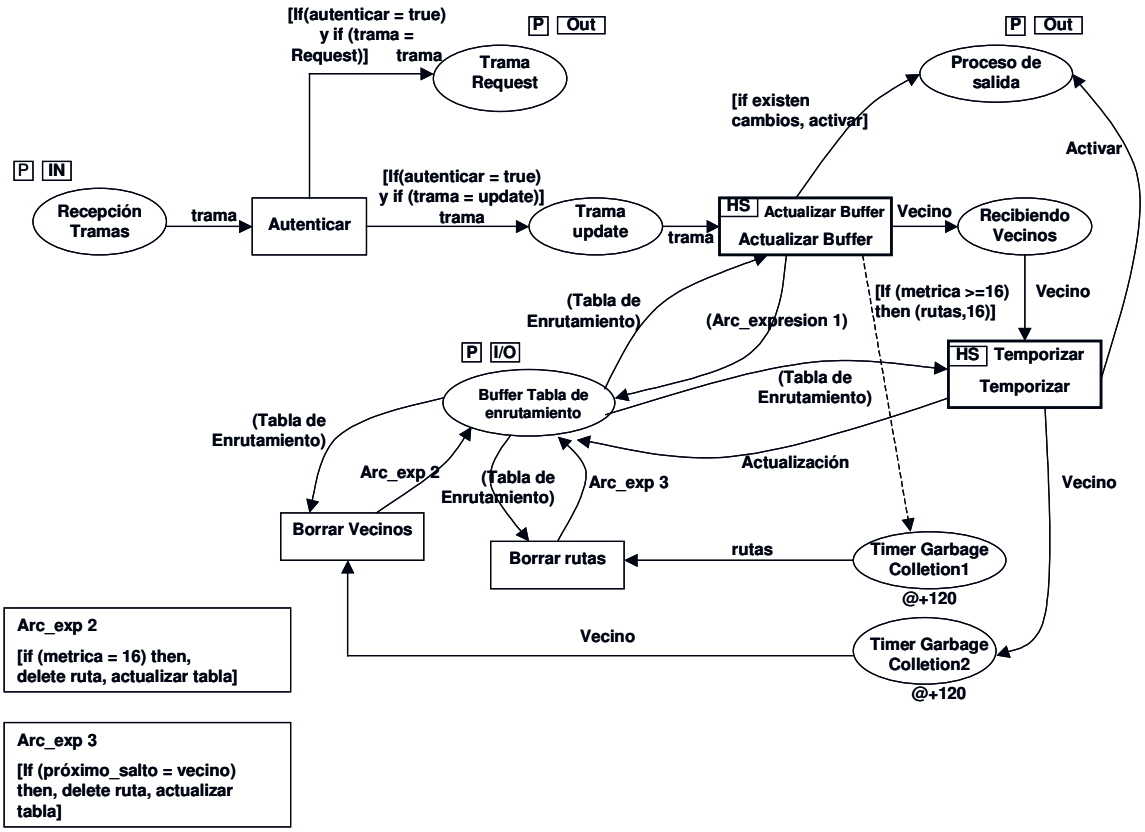


Figura 3.3 Subpágina Receptor

Estado	Recepción tramas
Transición	Autenticar
Nuevo estado	Trama request o trama update
Descripción	Cuando llega una trama, un token aparece en el estado *Recepción Tramas* habilitando la transición *Autenticar*, la cual se encarga de

	autenticar y detectar el tipo de trama que llega, para así habilitar cualquiera de los dos estados *Trama Request* o *Trama Update*, el primer estado es un puerto de salida y fue modelado en la subpágina transmisor.

Estado	Trama update
Transición	Actualizar buffer
Nuevo estado	Proceso de salida, recibiendo vecinos, buffer tabla de enrutamiento, timer garbage colletion 1.
Descripción	La recepción de una trama update activa la transición *Actualizar Buffer* que será detallada en la subpágina del mismo nombre mas adelante.

Estado	Recibiendo vecinos
Transición	Temporizar
Nuevo estado	Timer garbage colletion 2
Descripción	La transición *Actualizar Buffer* envía un token del tipo vecino a la transición *Temporizar* con el cual la activa. Dicha transición se encarga de contar el tiempo transcurrido entre actualizaciones sucesivas de un vecino para percatarse cuando esta fuera de servicio y enviar un token al estado *Timer Garbage Colletion 2* con la información del vecino que salió del servicio.

Estado	Timer garbage colletion 2
Transición	Borrar vecinos
Nuevo estado	Buffer tabla de enrutamiento
Descripción	Una vez llaga un token con la información de vecino que ha salido del servicio al estado *Timer Garbage Colletion 2*, este temporiza 120 segundos, pasados los cuales activa la transición *Borrar Vecinos* que borra del buffer tabla de enrutamiento todas las rutas aprendidas del vecino que salió del servicio.

Estado	Timer garbage colletion 1
Transición	Borrar ruta
Nuevo estado	Buffer tabla de enrutamiento
Descripción	Una vez llega un token con la información de las rutas que tienen una métrica igual a 16 o superior al estado *Timer Garbage Colletion 1*, este temporiza 120 segundos, pasados los cuales activa la transición *Borrar Rutas*, que borra del buffer *tabla de enrutamiento* las rutas con

	dicha métrica.

La figura 3.4 muestra la subpágina actualizar buffer que modela la transición sustituta Actualizar Buffer. Esta red CPN esta formada por 6 estados y 2 transiciones que se explican a continuación.

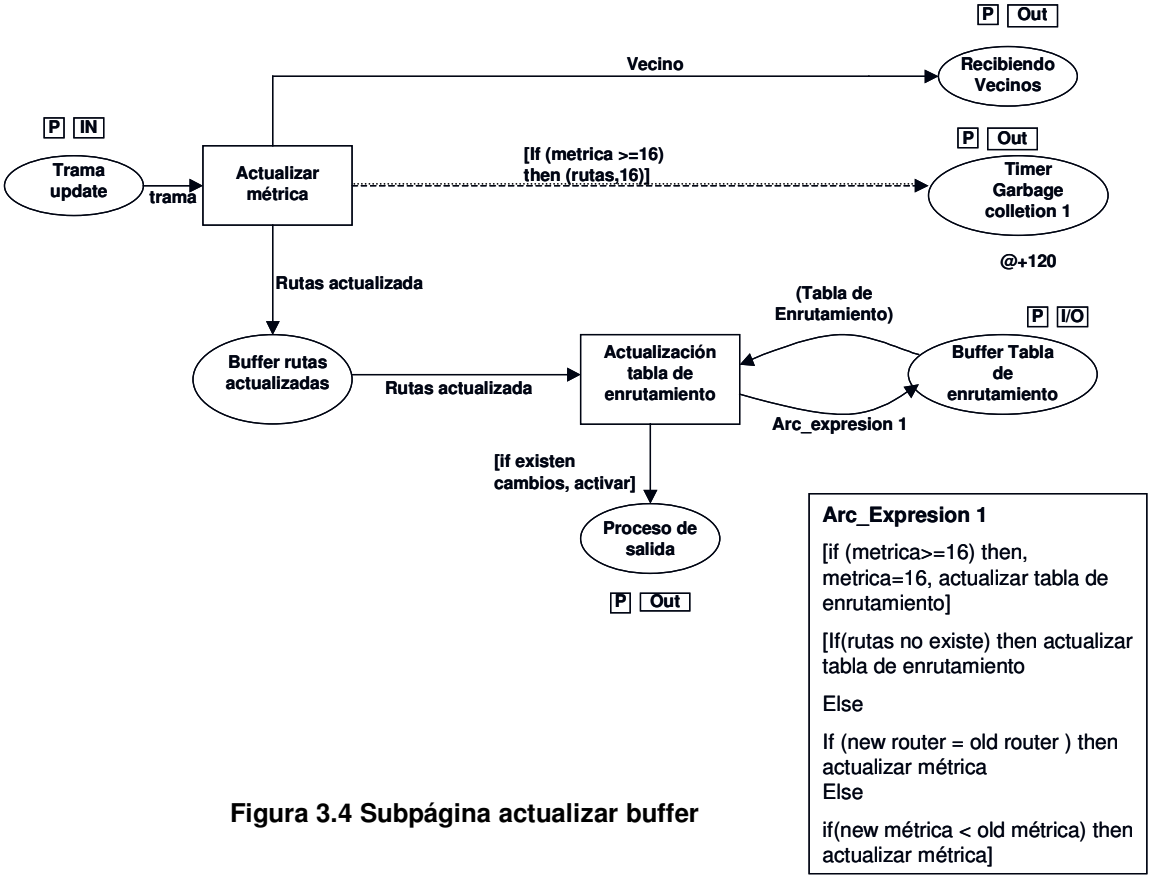


Figura 3.4 Subpágina actualizar buffer

Estado	Trama update
Transición	Actualizar métrica
Nuevo estado	Buffer rutas actualizadas y recibiendo vecinos
Descripción	La presencia de una trama update en el estado *Trama Update* activa la transición *Actualizar Métrica*, la cual se encarga de incrementar en 1 la métrica de las rutas. Una vez actualizada la métrica esta transición pasa un token a dos estados: *Recibiendo Vecinos*, que fue explicado anteriormente, y *Buffer Rutas Actualizadas*, el cual guardará las rutas aprendidas con la nueva métrica.

Estado	Trama update
Transición	Actualizar métrica
Nuevo estado	Timer garbage colletion 1.

Descripción	Si una vez actualizada la métrica el resultado es igual a 16 o mayor, la transición enviará un token al estado *Garbage Colletion 1* para iniciar el proceso de borrado de la ruta que se explicó anteriormente.

Estado	Buffer rutas actualizadas
Transición	Actualización tabla de enrutamiento
Nuevo estado	Buffer tabla de enrutamiento y proceso de salida
Descripción	El estado *Buffer Rutas Actualizadas* activará la transición *Actualización Tabla de Enrutamiento*, la cual se encarga de guardar en la tabla de enrutamiento las nuevas redes aprendidas, los cambios en redes ya existentes y de pasar un token al estado *Proceso de Salida* para activar las transiciones estudiadas en la subpágina transmisor.

La figura 3.5 muestra la subpágina temporizar que modela la transición sustituta Temporizar. Esta red CPN esta formada por 5 estados y 2 transiciones que se explica a continuación.

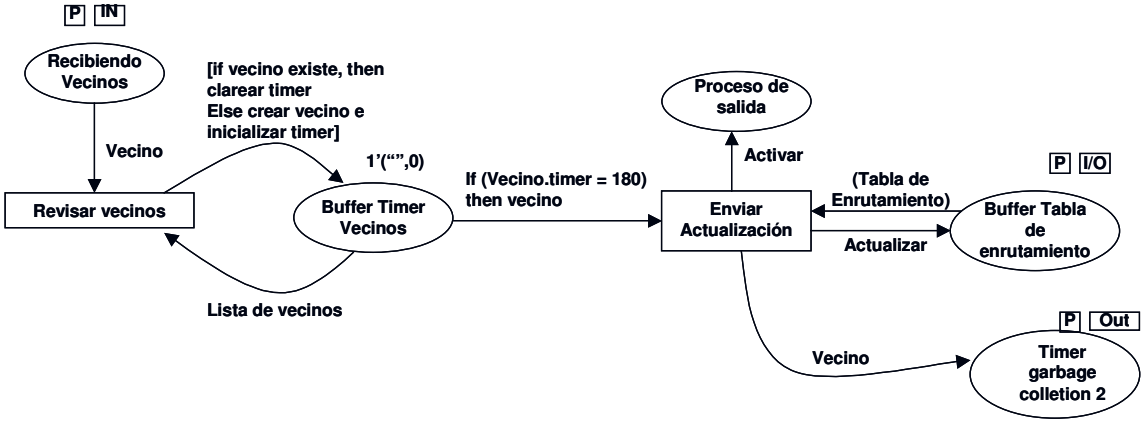


Figura 3.5 Subpágina temporizar

Estado	Recibiendo vecinos
Transición	Revisar vecinos
Nuevo estado	Buffer timer vecinos
Descripción	El estado *Recibiendo Vecinos* habilita la transición *Revisar Vecinos*, la cual se encarga de mirar en el *Buffer Timer Vecinos* si ya se han recibido mensajes, en cuyo caso la transición clarea el timer; en caso contrario, la transición crea el timer para el nuevo vecino.

Estado	Buffer timer vecinos
Transición	Enviar actualización
Nuevo estado	Proceso de salida, buffer tabla de enrutamiento, timer garbage colletion 2

Descripción	El estado *Buffer Timer Vecinos* temporiza la llegada de paquetes de los vecinos, y cuando no recibe mensajes por un periodo de 180 segundos de un determinado vecino habilita la transición *Enviar Actualización*, la cual se encarga de enviar tokens a los estados *Proceso de Salida* y *Timer Garbage Colletion*, además de actualizar el buffer de la tabla de enrutamiento indicando las rutas que van a ser eliminadas.

3.3 Colección de datos

Para la colección de los datos se realizaron pruebas de laboratorio donde se implementaron diferentes topologías de red. Para tal fin, se utilizaron tres enrutadores Cisco, dos de la serie 2500 y uno de la serie 4000, además se usó un software freeware de enrutamiento denominado Zebra que corre bajo la plataforma Linux y que soporta los protocolos de enrutamiento RIP, OSPF y BGP-4. Zebra ofrece la capacidad de convertir un computador con dos o más tarjetas de red en un enrutador. A nivel de laboratorio esto presenta una gran ventaja cuando no se dispone de la cantidad necesaria de enrutadores físicos para realizar la práctica. Los resultados son los mismos y la forma de configurar el software de enrutamiento es muy similar a la utilizada en cualquier enrutador físico.

Para el protocolo IGRP las pruebas se realizaron utilizando enrutadores Cisco de las series 2500 y 4000 ya que el software Zebra no soporta este protocolo; para el protocolo BGP-4 las pruebas se realizaron utilizando los enrutadores físicos de referencia 2514 y 4000 y el software Zebra.

Los resultados obtenidos de estos laboratorios con la comparación de los resultados arrojados por el simulador se presentan en el capítulo 4.

3.4 Implementación del modelo en la computadora

Para el modelado de la herramienta software se hizo uso de la metodología RUP, la cual fue explicada en el capítulo anterior. En esta sección se presentan los requerimientos, las funciones y los casos de uso más representativos del sistema.

El diseño completo de la herramienta se presenta en el anexo B

3.4.1 Descripción de Requerimientos

Los requerimientos de la herramienta software para la simulación de protocolos de enrutamiento en redes IP se han dividido en tres grupos.

El primero esta relacionado con los mecanismos que deberá proveer la herramienta para crear una topología de red. Para ello, se debe contar con un área de dibujo donde el usuario tendrá la libertad de colocar dispositivos de red tales como host, hubs y enrutadores.

El sistema deberá implementar funciones que brinden flexibilidad durante el proceso de creación de la topología de red, que permitan mover los componentes de un lugar a otro dentro del área de dibujo, de borrarlos si se requiere, y que permitan realizar conexiones entre los dispositivos de una manera intuitiva y rápida.

El segundo grupo de requerimientos hace referencia a la simulación, donde el sistema deberá mostrar las opciones necesarias para que se escoja el protocolo de enrutamiento a simular, presentar los diálogos de configuración de los dispositivos de red haciendo especial énfasis en los enrutadores, y permitir empezar y detener la simulación. También, cuando se este simulando, se deberán mostrar los tipos de mensajes que pasan a través de la red.

El último grupo esta relacionado con la presentación de información de los procesos de enrutamiento que se llevan a cabo dentro de la herramienta, la cual deberá ser organizada y entendible, y se procurará que el usuario pueda escoger el tipo de información que desea observar.

También se debe contar con ayudas, manual de usuario y laboratorios que faciliten el uso de la herramienta y la comprensión de la temática de enrutamiento.

3.4.2 Propósito del sistema

El sistema debe permitir la simulación de los protocolos de enrutamiento RIP, OSPF, BGP-4, IGRP para que proporcione una mejor comprensión del funcionamiento real de ellos, contribuyendo así a elevar la calidad de la enseñanza de esta temática en la FIET.

3.4.3 Construcción del árbol de funciones

A continuación se identifican las funciones del sistema y se establece una jerarquía donde las funciones más generales agrupan las mas específicas, la siguiente tabla muestra el árbol de funciones para el sistema:

1. Dibujar Figura (hace referencia a los dispositivos de red que se podrán manipular con la herramienta, estos dispositivos son: enrutadores, hubs y hosts).
    1.1 Seleccionar figura.
    1.2 Copiar figura.
    1.3 Pegar figura.

1.4 Eliminar figura
1.5 Conectar figura.
1.6 Mover figura.

2. Manipular Archivos (hacen referencia a los archivos generados por la herramienta a partir de las redes construidas por el usuario).
   2.1 Guardar archivo.
   2.2 Abrir archivo.
   2.3 Modificar archivo.
   2.4 Imprimir archivo.
   2.5 Crear archivo.

3. Hacer consultas
   3.1 Consultar manual de usuario.
   3.2 Consultar teoría de los protocolos de enrutamiento.
   3.3 Consultar laboratorios.
   3.4 Consultar datos de diseñadores.

4. Ejecutar simulación
   4.1 Seleccionar protocolo a simular.
   4.2 Configurar interfaces del enrutador.
   4.3 Empezar simulación.
   4.4 Mostrar tramas.
   4.5 Detener simulación.
   4.6 Desactivar interfaz del enrutador
   4.7 Activar interfaz del enrutador.
   4.8 Mirar tabla de enrutamiento.
   4.9 Monitorear figura enrutador.

Tabla 3.1 Funciones del Sistema

3.4.4 Diagrama de casos de uso

El sistema cuenta con un solo autor que se relaciona de manera directa con el sistema, y de ahora en adelante lo llamaremos usuario.

Los casos de usos se desprenden del árbol de funciones, y se presentan en la figura 3.6.

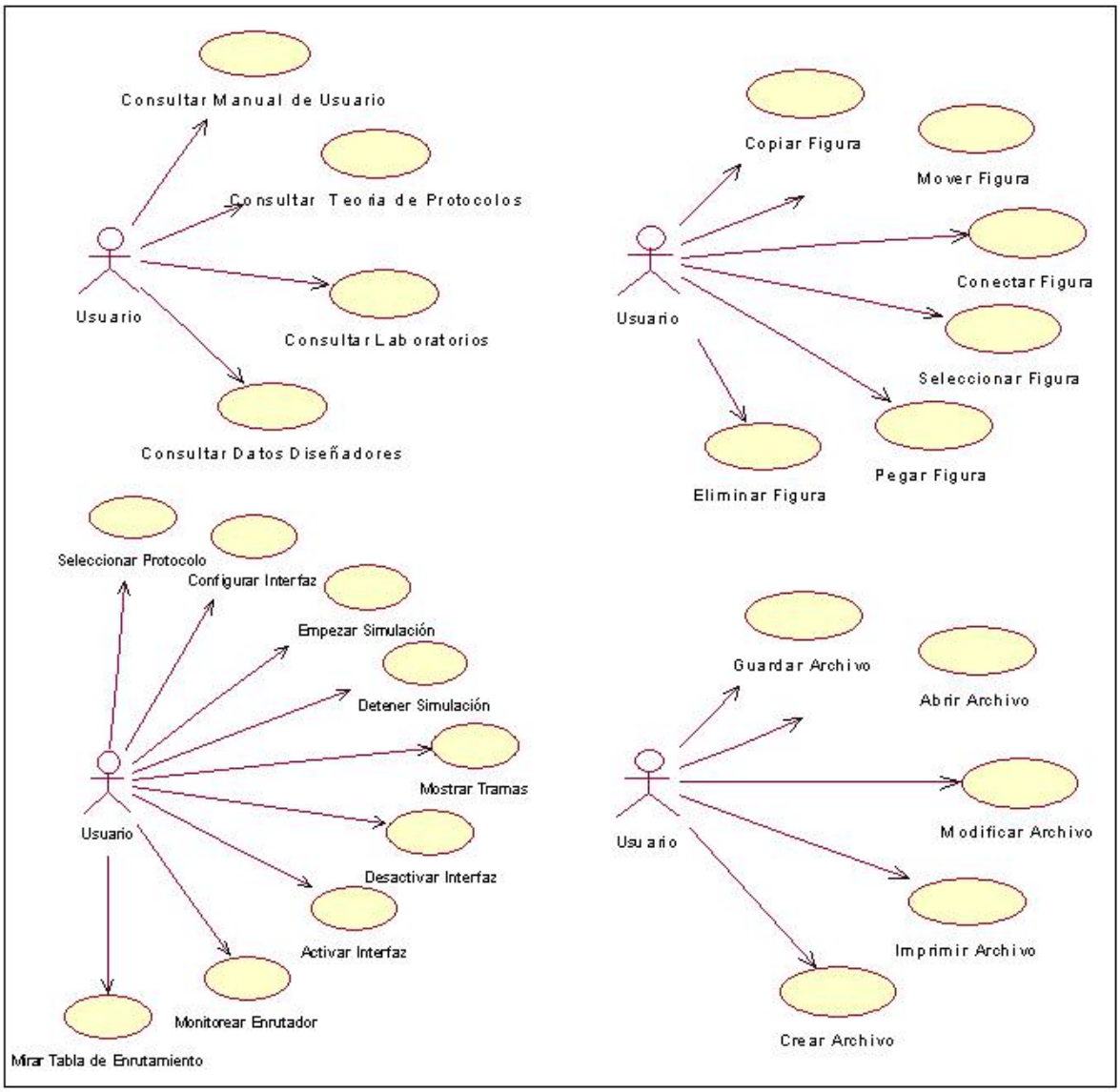


Figura 3.6 Diagrama de casos de uso

3.4.5 Descripción de los casos de uso

Caso de uso	**Dibujar figuras**
Actor	Usuario
Tipo	Primario
Descripción	El usuario selecciona la figura (enrutador, hub o host) que desea colocar en el área de dibujo. El sistema guarda la información de su elección. El usuario da clic sobre el área de dibujo. El sistema dibuja la figura.

Caso de uso	**Copiar**
Actor	Usuario
Tipo	Secundario
Descripción	El usuario selecciona dentro del área de dibujo la figura que desea copiar. El usuario da la orden al sistema de copiar el dispositivo. El sistema guarda la figura que el usuario copio.

Caso de uso	**Pegar**
Actor	Usuario
Tipo	Secundario
Descripción	El usuario solicita al sistema pegar una figura. El sistema dibuja la figura que se había copiado anteriormente en el área de dibujo.

Caso de uso	**Eliminar**
Actor	Usuario
Tipo	Secundario
Descripción	El usuario selecciona una figura en el área de dibujo y solicita al sistema borrarlo. El sistema borrará el dispositivo del área de dibujo y si tiene conexiones con otros dispositivos también se quitarán dichas conexiones.

Caso de uso	**Conectar figuras**
Actor	Usuario
Tipo	Primario
Descripción	El usuario da clic con el botón derecho del mouse sobre una de las figuras que coloco en el área de dibujo. El sistema mostrará las interfaces del dispositivo. El usuario selecciona cualquier interfaz. Esta operación se repite en la figura con la cual se quiere realizar la conexión. El sistema dibujará una línea indicando que se ha realizado la conexión.

Caso de uso	**Seleccionar**
Actor	Usuario

Tipo	Primario
Descripción	El usuario da clic sobre una figura que se encuentra en el área de dibujo. El sistema dibuja unos pequeños cuadros negros alrededor del dispositivo que selecciono.

Caso de uso	**Mover**
Actor	Usuario
Tipo	Primario
Descripción	El usuario da clic sobre una figura que se encuentra en el área de dibujo dejando el botón del mouse presionado. A medida que mueva el mouse, la figura se moverá con el. El sistema captura las posiciones del cursor y dibuja la figura en la nueva posición.

Caso de uso	**Guardar**
Actor	Usuario
Tipo	Primario
Descripción	El usuario solicita al sistema que guarde la configuración de red realizada. El sistema despliega un formulario para que se especifique el directorio y el nombre del archivo. El usuario debe escoger el directorio y el nombre del archivo. El sistema guardará la configuración de red en un archivo.

Caso de uso	**Abrir**
Actor	Usuario
Tipo	Primario
Descripción	El usuario da clic sobre botón abrir. El sistema despliega un formulario para que el usuario escoja el archivo que desea abrir. El sistema presentará la información guardada anteriormente en el archivo.

Caso de uso	**Imprimir**
Actor	Usuario
Tipo	Secundario
Descripción	El usuario solicita al sistema imprimir. El sistema imprimirá la topología de red que se encuentra en el área

	de dibujo.

Caso de uso	**Manual de usuario**
Actor	Usuario
Tipo	Secundario
Descripción	El usuario solicita ayuda. El sistema despliega una página web que contiene la ayuda para el manejo de la herramienta.

Caso de uso	**Consultar Protocolos**
Actor	Usuario
Tipo	Secundario
Descripción	El usuario solicita información de los protocolos. El sistema desplegará una página web que contiene información teórica sobre los protocolos de enrutamiento.

Caso de uso	**Consultar Laboratorios**
Actor	Usuario
Tipo	Secundario
Descripción	El usuario solicita consultar laboratorios. El sistema desplegará una página web que contiene los laboratorios que se diseñaron para el manejo de la herramienta.

Caso de uso:	**Consultar datos de los diseñadores**
Actores:	Usuario
Tipo:	Secundario
Descripción:	El usuario solicita al sistema ver la información de los diseñadores e implementadores de la herramienta. El sistema despliega en pantalla la información solicitada. El usuario recibe la información.

Caso de uso:	**Seleccionar protocolo**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario escoge el protocolo con el que desea hacer la simulación. El sistema deshabilita la selección de los otros protocolos y muestra al usuario de manera permanente y hasta que dure la simulación el tipo de tramas que implementa el protocolo seleccionado y los campos de la tabla de enrutamiento con la que trabaja dicho

	protocolo.

Caso de uso:	**Configurar interfaces**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario selecciona un enrutador y solicita al sistema configurar las interfaces de dicho componente. El sistema verifica que la operación puede efectuarse y despliega las interfaces del componente y los campos necesarios para la correcta configuración de cada una de estas. El usuario ingresa la información (dirección IP, mascara, etc) para la configuración. El sistema modifica el estado y los parámetros de la interfaz.

Caso de uso:	**Empezar simulación**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario solicita al sistema iniciar la simulación de uno de los protocolos de enrutamiento que este soporta. El sistema presenta al usuario de manera gráfica el intercambio de paquetes entre los enrutadores y los cambios en la tablas de enrutamiento de estos.

Caso de uso:	**Mostrar tramas**
Actores:	Usuario
Tipo:	Secundario
Descripción:	El usuario solicita al sistema información del contenido de las tramas que se presentan en la simulación. El sistema muestra al usuario la información solicitada.

Caso de uso:	**Detener simulación**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario solicita al sistema detener un proceso de simulación que ya había sido iniciado. El sistema libera todos los recursos asignados a la simulación.

Caso de uso:	**Desactivar interfaz de los componentes**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario selecciona un componente, escoge una interfaz y solicita al sistema desactivarla. El sistema desactiva la interfaz y cambia la apariencia del formulario.

Caso de uso:	**Activar interfaz de los componentes**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario selecciona un componente, escoge una interfaz y solicita al sistema activarla. El sistema activa la interfaz y cambia la apariencia física del formulario.

Caso de uso:	**Mirar tabla de enrutamiento**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario selecciona un enrutador y el sistema le responde con la tabla de enrutamiento del componente seleccionado.

Caso de uso:	**Monitorear enrutador**
Actores:	Usuario
Tipo:	Primario
Descripción:	El usuario le solicita al sistema monitorear el trafico enviado y recibido por un enrutador. El sistema presenta al usuario el tipo de paquetes que puede monitorear. El usuario selecciona el tipo de paquete a monitorear. El sistema presenta al usuario todos los paquetes entrantes y salientes del tipo seleccionado.

A continuación se describirá detalladamente el caso de uso configurar interfaces el cual es de gran importancia en la aplicación. Los restantes casos de uso se detallan en el Anexo B.

3.4.6 Caso de uso Extendido Configurar Interfaces

Caso de uso:	**Configurar Interfaces**
Actores:	Usuario (iniciador)
Tipo:	Primario

Resumen:	El usuario selecciona un enrutador y solicita al sistema configurar las interfaces de dicho componente. El sistema verifica que la operación puede efectuarse y despliega a través de una GUI las interfaces del componente y los campos necesarios para la correcta configuración de cada una de estas. El usuario ingresa la información (dirección IP, mascara, etc) para la configuración. El sistema modifica el estado y los parámetros de la interfaz.
Referencias cruzadas	Funciones: 4.2, 4.3 Casos de uso: Seleccionar protocolo, Conectar figuras.

Precondiciones

- El usuario debe haber ejecutado el caso de uso seleccionar protocolo.
- El usuario debe haber ejecutado el caso de uso conectar figuras.
- La interfaz que desea configurar debe estar conectada a otra interfaz del mismo tipo (Ethernet o serial).

Flujo principal

- Este caso de uso empieza cuando el usuario pulsa doble clic sobre uno de los enrutadores.
- El sistema presenta al usuario el formulario de configuración del enrutador de la figura 3.7.
- El usuario escoge la opción configurar interfaces.
- El usuario selecciona la interfaz que desea configurar e introduce su dirección IP, la mascara de red, la métrica y opcionalmente el numero de área o sistema autónomo (dependiendo del protocolo que desea simular).

Figura 3.7 Formulario configuración enrutador

- Una vez introducidos los datos el usuario escoge la opción aceptar (E1, E2, E3, E4). Los valores de loas campos del anterior formulario pueden ser modificados siempre y cuando el usuario no halla ejecutado el caso de uso empezar simulación, para realizar algún cambio simplemente se modifica el campo y se escoge la opción aceptar nuevamente.
- El sistema muestra un mensaje al usuario anunciándole que ha configurado correctamente la interfaz.

Flujos de Excepción

E1: El sistema despliega un mensaje informando que se deben llenar todos los campos habilitados.

- El sistema regresa al formulario de configuración del enrutador.

E2: El sistema despliega un mensaje informando que la dirección IP o la mascara no tienen un formato adecuado.

- El sistema regresa al formulario de configuración del enrutador.

E3: El sistema despliega un mensaje informando que la dirección IP ya ha sido asignada.

- El sistema regresa al formulario de configuración del enrutador.

E4: El sistema despliega un mensaje informando que no se ha podido configurar la interfaz que revise las conexiones.

- El sistema regresa al formulario de configuración del enrutador.

3.4.7 Diagrama de Secuencia Configurar Interfaces

El diagrama de secuencia se presenta en la figura 3.8 y se explica a continuación: el usuario pulsa doble clic en el área de dibujo, evento que es recogido por la operación mouseClicked() de *AreaDibujo*, la cual analiza si este se dio sobre un objeto de la clase *Enrutador*, de ser así y de cumplirse con los requerimientos del caso de uso, se activa un objeto de la clase *DialogoConfiguración,* donde el usuario selecciona la interfaz que desea configurar, introduce los datos y presiona clic sobre el botón Aceptar, evento que invoca la operación configurarInterfaz(), que se encarga de analizar los parámetros introducidos y de guardar los datos. El sistema retorna una respuesta exitosa a *DialogoConfiguracion* quien finalmente activa un objeto de la clase *IU_OperacionExitosa.*

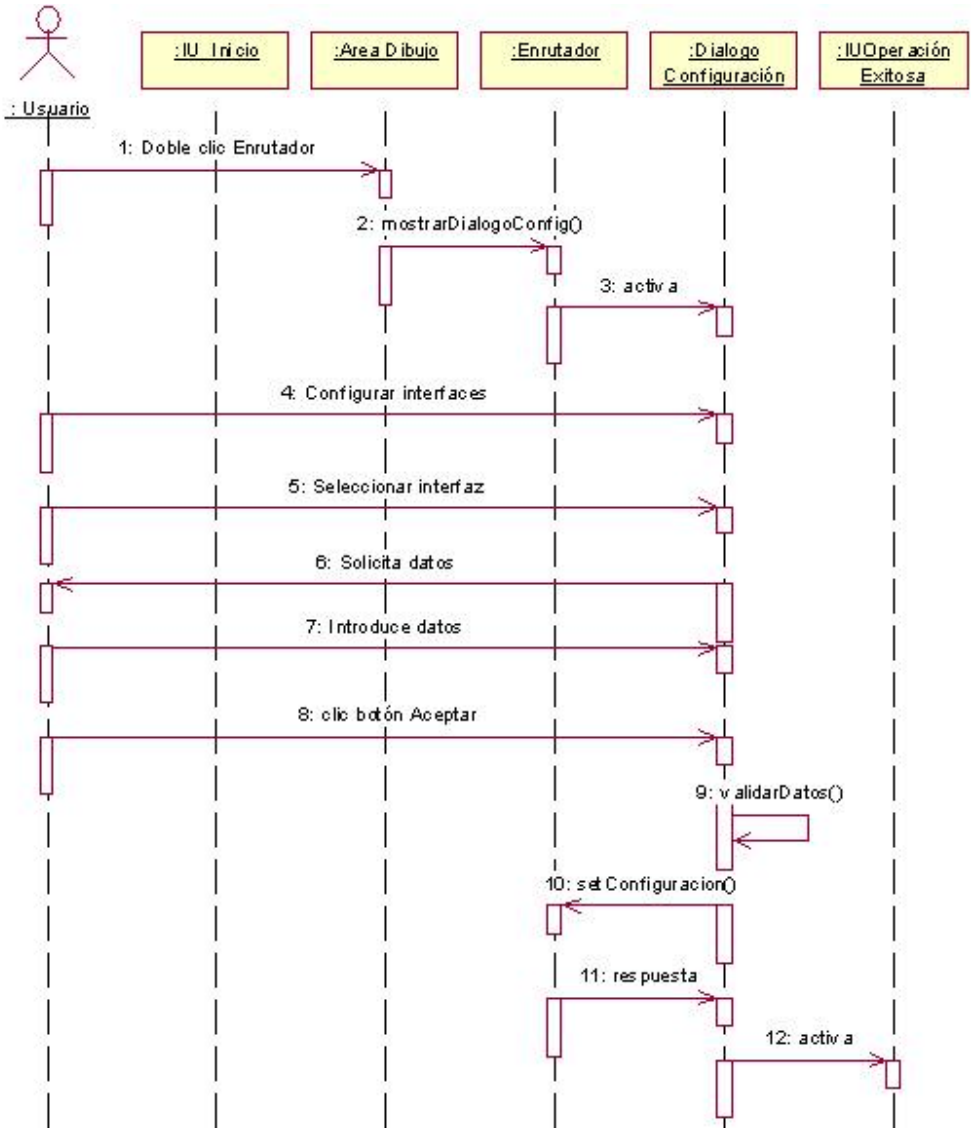


Figura 3.8 Diagrama de secuencia Configurar interfaces

3.4.8 Diagrama de clases

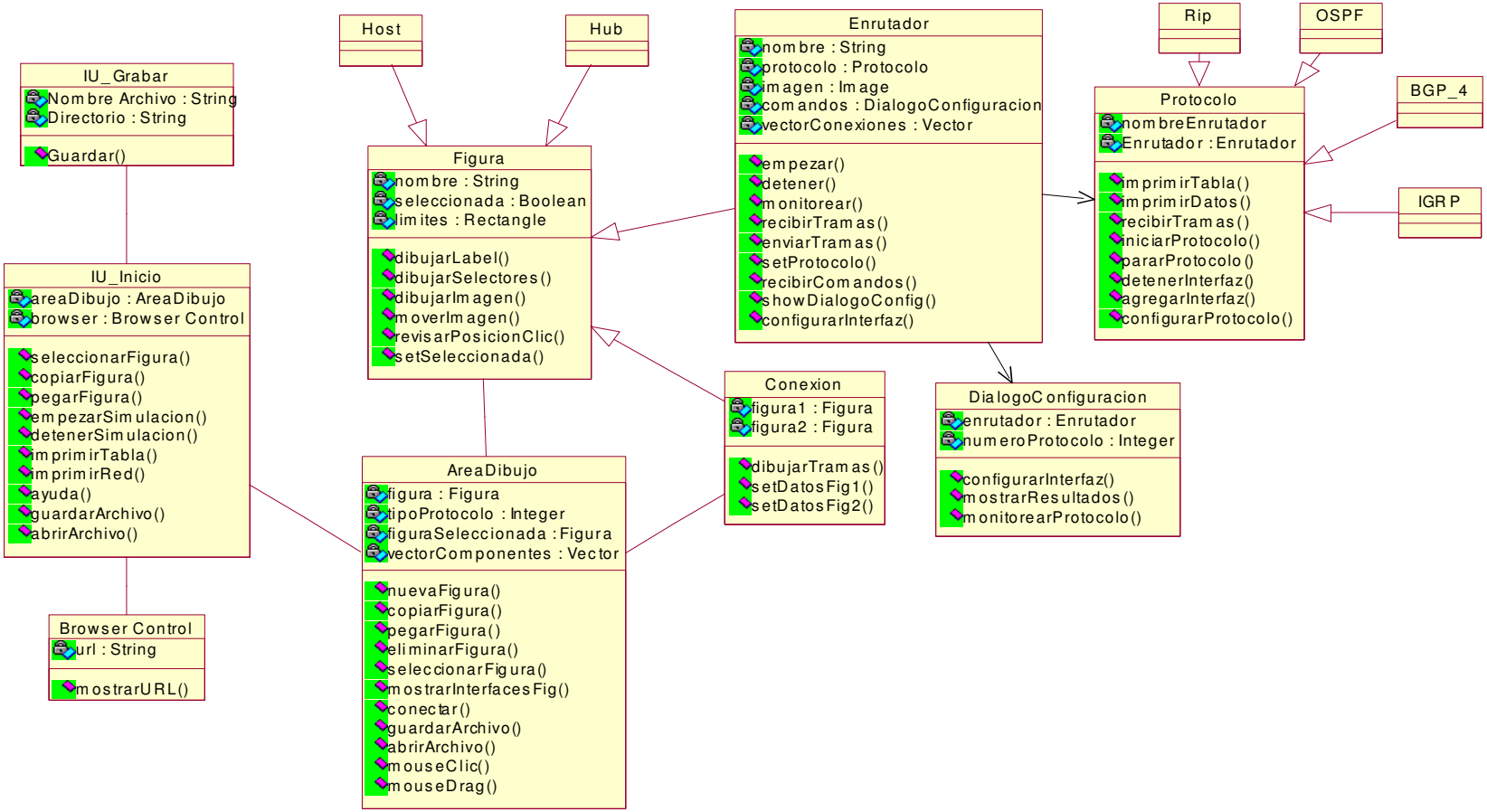


Figura 3.9 Diagrama de Clases

Para la implementación de la herramienta se hizo uso de la plataforma J2SE de JAVA con el JDK1.3.1.

Los instaladores fueron generados con la herramienta freeware para aplicaciones no comerciales **Install4j**.

4. VALIDACIÓN DEL SISTEMA

En este capítulo se confrontarán los resultados obtenidos en las prácticas de laboratorio con los resultados obtenidos con la herramienta de simulación.

A continuación se presentan los resultados para cada uno de los protocolos de enrutamiento.

4.1 RIP

4.1.1 Laboratorio RIP 1

Topología de red

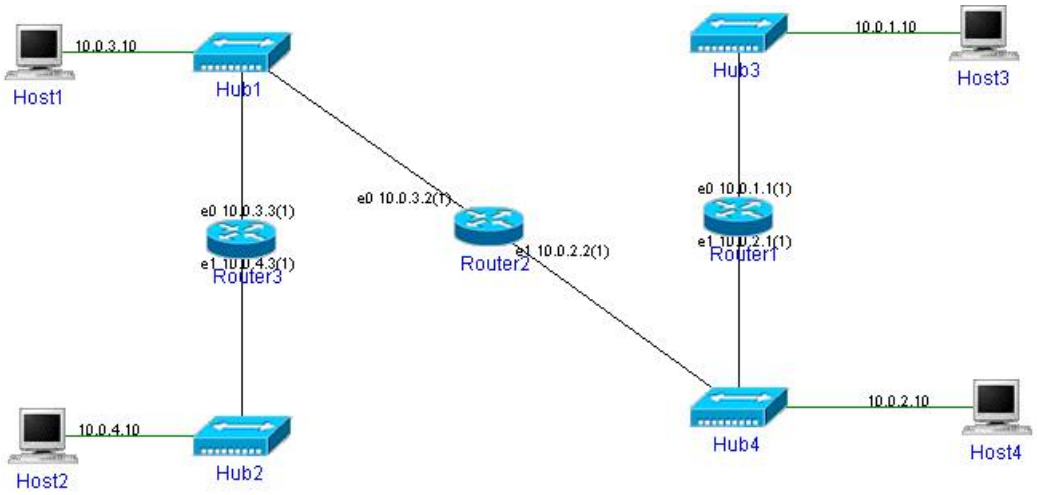


Figura 4.1 Topología RIP 1

Objetivo

Observar y analizar el comportamiento del protocolo RIP y los resultados de las tablas de enrutamiento.

Resultados de Laboratorio

Router 1

```
Router1#show ip rip
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

      Network          Next Hop          Metric From          Time
C(i) 10.0.1.0/24       0.0.0.0                1  self
C(i) 10.0.2.0/24       0.0.0.0                1  self
R(n) 10.0.3.0/24       10.0.2.2               2  10.0.2.2     02:37
R(n) 10.0.4.0/24       10.0.2.2               3  10.0.2.2     02:37
```

Se describirá la tabla de enrutamiento para este enrutador solamente. La interpretación es similar para los demás enrutadores.

- C indica que la red que se encuentra en la columna Network esta conectada directamente a una interfaz del enrutador.
- R indica que la red se ha aprendido a través del protocolo RIP.
- Network muestra la dirección IP de la red a la que puede alcanzar y la máscara
- Next Hop indica el próximo salto para llegar a una red específica.
- Metric indica el número de enrutadores por los cuales se debe pasar para llegar a la red destino.
- From especifica el enrutador de donde se aprendió la red.

Análisis de la tabla de enrutamiento

- Como se puede observar, el enrutador 1 tiene dos redes directamente conectadas a sus interfaces, la red 10.0.1.0 y la red 10.0.2.0. En la tabla se refleja este resultado.
- Las demás redes las ha aprendido usando los paquetes RIP que son enviados por el enrutador 2 a través de la interfaz e1 cuya dirección es 10.0.2.2.
- Para este enrutador el próximo salto hacia otras redes es 10.0.2.2.
- En cuanto a la métrica, la redes que están directamente conectadas cuentan con métrica de 1, de allí en adelante solo resta contar los routers durante el trayecto hasta el destino para calcular la métrica hacia otras redes.

Router 2

```
Router2#show ip rip
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

      Network          Next Hop          Metric From          Time
C(i) 10.0.2.0/24       0.0.0.0                1  self
C(i) 10.0.3.0/24       0.0.0.0                1  self
R(n) 10.0.1.0/24       10.0.2.1               2  10.0.2.1     02:37
R(n) 10.0.4.0/24       10.0.3.3               2  10.0.3.3     02:37
```

Router 3

```
Router3#show ip rip
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

      Network          Next Hop          Metric From          Time
C(i) 10.0.3.0/24       0.0.0.0                1  self
C(i) 10.0.4.0/24       0.0.0.0                1  self
R(n) 10.0.1.0/24       10.0.3.2               2  10.0.3.2     02:18
R(n) 10.0.2.0/24       10.0.3.2               2  10.0.3.2     02:18
```

Resultados de la herramienta

Router 1

flag	Red destino	Máscara	Next Hop	Interfaz	Metrica
0	10.0.1.0	255.255.255.0	Directamente conectada	E0	1
0	10.0.2.0	255.255.255.0	Directamente conectada	E1	1
0	10.0.3.0	255.255.255.0	10.0.2.2	E1	2
0	10.0.4.0	255.255.255.0	10.0.2.2	E1	3

Router 2

flag	Red destino	Máscara	Next Hop	Interfaz	Metrica
0	10.0.3.0	255.255.255.0	Directamente conectada	E0	1
0	10.0.2.0	255.255.255.0	Directamente conectada	E1	1
0	10.0.4.0	255.255.255.0	10.0.3.3	E0	2
0	10.0.1.0	255.255.255.0	10.0.2.1	E1	2

Router 3

flag	Red destino	Máscara	Next Hop	Interfaz	Metrica
0	10.0.1.0	255.255.255.0	Directamente conectada	E0	1
0	10.0.2.0	255.255.255.0	Directamente conectada	E1	1
0	10.0.3.0	255.255.255.0	10.0.2.2	E1	2
0	10.0.4.0	255.255.255.0	10.0.2.2	E1	3

Como se puede comparar, los resultados de las tablas de enrutamiento obtenidos en la herramienta de simulación son los mismos que los obtenidos con las pruebas de laboratorio. Varían algunos campos que no son relevantes para el entendimiento del protocolo como la columna time.

4.1.2 Laboratorio RIP 2

Topología de red

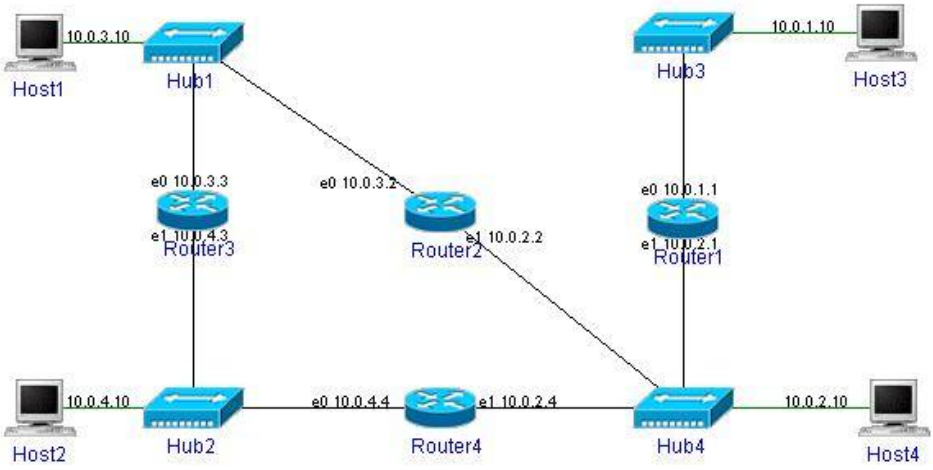


Figura 4.2 Topología RIP 2

Objetivo

En esta práctica se adiciona un nuevo enrutador y se pretende observar los cambios en las tablas de enrutamiento.

Resultados de laboratorio

Una vez se conectó el enrutador 4, la tabla de enrutamiento que se modificó fue la del enrutador 1.

Router 1

```
Router1#show ip rip
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

      Network          Next Hop          Metric From          Time
C(i)  10.0.1.0/24      0.0.0.0                1  self
C(i)  10.0.2.0/24      0.0.0.0                1  self
R(n)  10.0.3.0/24      10.0.2.2               2  10.0.2.2     03:00
R(n)  10.0.4.0/24      10.0.2.4               2  10.0.2.4     02:29
```

Router 4

```
Router4#show ip rip
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

      Network          Next Hop          Metric From          Time
C(i)  10.0.2.0/24      0.0.0.0                1  self
C(i)  10.0.4.0/24      0.0.0.0                1  self
R(n)  10.0.1.0/24      10.0.2.1               2  10.0.2.1     02:59
R(n)  10.0.3.0/24      10.0.2.2               2  10.0.2.2     02:50
```

De estos resultados se puede observar que el enrutador 1 encontró un camino más corto para llegar a la red 10.0.4.0 a través del enrutador 4, modificando su métrica de 3 a 2.

Resultados de la herramienta

Router 1

flag	Red destino	Máscara	Next Hop	Interfaz	Metrica
0	10.0.1.0	255.255.255.0	Directamente conectada	E0	1
0	10.0.2.0	255.255.255.0	Directamente conectada	E1	1
0	10.0.3.0	255.255.255.0	10.0.2.2	E1	2
0	10.0.4.0	255.255.255.0	10.0.2.4	E1	2

Router 4

flag	Red destino	Máscara	Next Hop	Interfaz	Metrica
0	10.0.1.0	255.255.255.0	Directamente conectada	E0	1
0	10.0.2.0	255.255.255.0	Directamente conectada	E1	1
0	10.0.3.0	255.255.255.0	10.0.2.2	E1	2
0	10.0.4.0	255.255.255.0	10.0.2.4	E1	2

Como se puede apreciar, los cambios en las tablas de enrutamiento de los enrutadores 1 y 4 se reflejan de igual forma en la herramienta.

4.2 OSPF

4.2.1 Laboratorio OSPF 1

Topología de red

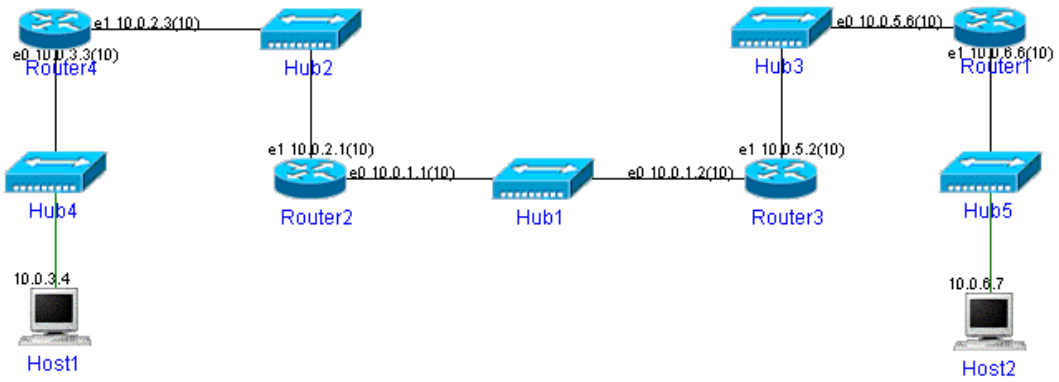


Figura 4.3 Topología OSPF 1

Objetivo

Observar y analizar el comportamiento del protocolo OSPF, los resultados de las tablas de enrutamiento y de la base de datos del estado del enlace.

Resultados de laboratorio

Router 1

```
Router1> show ip ospf route

============ OSPF network routing table ============
N    10.0.1.0/24           [20] area: 0.0.0.1
                           via 10.0.5.2, eth0
N    10.0.2.0/24           [30] area: 0.0.0.1
                           via 10.0.5.2, eth0
N    10.0.3.0/24           [40] area: 0.0.0.1
                           via 10.0.5.2, eth0
N    10.0.5.0/24           [10] area: 0.0.0.1
                           directly attached to eth0
N    10.0.6.0/24           [10] area: 0.0.0.1
                           directly attached to eth1
```

Router 2

```
Router2> show ip ospf route

============ OSPF network routing table ============
N    10.0.1.0/24           [10] area: 0.0.0.1
                           directly attached to eth0
N    10.0.2.0/24           [10] area: 0.0.0.1
                           directly attached to eth1
N    10.0.3.0/24           [20] area: 0.0.0.1
                           via 10.0.2.3, eth1
N    10.0.5.0/24           [20] area: 0.0.0.1
                           via 10.0.1.2, eth0
N    10.0.6.0/24           [30] area: 0.0.0.1
                           via 10.0.1.2, eth0
```

En las tablas de enrutamiento, se pueden observar las redes conectadas al enrutador directamente y la ruta hacia otras redes indicando cual es el siguiente enrutador por el cual se debe pasar para alcanzar cada red. El valor que se encuentra entre corchetes corresponde al costo total de la ruta. Por defecto se asigna un costo de 10 a un enlace.

Base de datos del Estado del Enlace

Router 1

```
Router1> show  ip ospf database

       OSPF Router with  ID (10.0.5.6)

                Router Link States  (Area 0.0.0.1)

Link ID         ADV  Router      Age  Seq #       CkSum  Link count
10.0.1.1        10.0.1.1         209 0x80000012 0x1cb6 2
10.0.1.2        10.0.1.2        1521 0x8000000f 0xae1e 2
10.0.3.3        10.0.3.3         215 0x80000010 0x9342 2
10.0.5.6        10.0.5.6        1482 0x8000000e 0x249e 2

                Net Link States  (Area 0.0.0.1)

Link ID         ADV  Router      Age  Seq #       CkSum
10.0.1.2        10.0.1.2        1557 0x8000000c 0x4dce
10.0.2.3        10.0.3.3         215 0x80000001 0x62bc
10.0.5.2        10.0.1.2        1521 0x8000000c 0x9777
```

Esta información es idéntica en cada uno de los enrutadores era de esperarse, pues cada enrutador debe mantener una copia de esta base de datos. La única diferencia se encuentra

en el campo age, que indica el tiempo en segundos que ha transcurrido desde que aprendió esa información.

El campo Link ID identifica el enlace del enrutador, que corresponde al router-id, el campo ADV Router identifica el enrutador que anunció el enlace. El campo Link count se refiere al número de enlaces que tiene conectado el enrutador correspondiente. El campo #Seq corresponden al número de secuencia que se utiliza para identificar una nueva LSA.

Resultados de la herramienta

Router 1

Tipo	Opcion	Red	Mascara	Metrica	Area	NextHop
N		10.0.5.0	255.255.255.0	10	1	Directamente conectada
N		10.0.6.0	255.255.255.0	10	1	Directamente conectada
N		10.0.1.0	255.255.255.0	20	1	10.0.5.2
N		10.0.2.0	255.255.255.0	30	1	10.0.5.2
N		10.0.3.0	255.255.255.0	40	1	10.0.5.2

Router 2

Tipo	Opcion	Red	Mascara	Metrica	Area	NextHop
N		10.0.1.0	255.255.255.0	10	1	Directamente conectada
N		10.0.2.0	255.255.255.0	10	1	Directamente conectada
N		10.0.5.0	255.255.255.0	20	1	10.0.1.2
N		10.0.3.0	255.255.255.0	20	1	10.0.2.3
N		10.0.6.0	255.255.255.0	30	1	10.0.1.2

Base de datos del Estado del Enlace

```
                Router Liks States (Area 1)
LinkID      ADVRouter   Age    Seq#   Cksum    Link Count
10.0.1.1    Router2     1128   2      0x77bc   Numero links2
10.0.5.2    Router3     1151   2      0x25bc   Numero links2
10.0.2.3    Router4     1139   2      0x46bc   Numero links2
10.0.5.6    Router1     1122   2      0x56bc   Numero links2

            Network Liks States (Area 1)
LinkID      ADV Router  Age    Seq#   Cksum
10.0.1.2    Router3     1193   0      0x6bc
10.0.3.3    Router4     1105   0      0x30bc
10.0.2.3    Router4     1174   0      0x7bc
10.0.5.6    Router1     1173   0      0x6bc
10.0.6.6    Router1     1161   0      0x21bc
```

Los resultados de las tablas de enrutamiento generados por la herramienta son semejantes a los obtenidos en la práctica.
Las diferencias en la Base de datos del estado del Enlace se explican a continuación:

- Los campos Age, Seq# y Cksum difieren ya que no presentan un comportamiento homogéneo. Por ejemplo, el campo Age representa el tiempo en segundos que han transcurrido desde que se aprendió esa información, el número de secuencia empieza con un valor aleatorio y se encuentra expresado en notación hexadecimal mientras que en la herramienta esta en notación decimal y el Cksum depende de la longitud de las tramas.
- Para que sea más entendible la lectura de los datos, en el campo ADVRouter se especifica el nombre del enrutador y no su dirección IP más alta como especifica el protocolo.
- La herramienta no tiene en cuenta redes STUB, estas son simuladas como redes Broadcast, por lo tanto se genera una Network Link States de más en los routers 1 y 4. Por este motivo la Base de datos del Estado del Enlace de la herramienta posee dos entradas más en los campos de Network Link States.

4.2.2 Laboratorio OSPF 2

Topología de la red

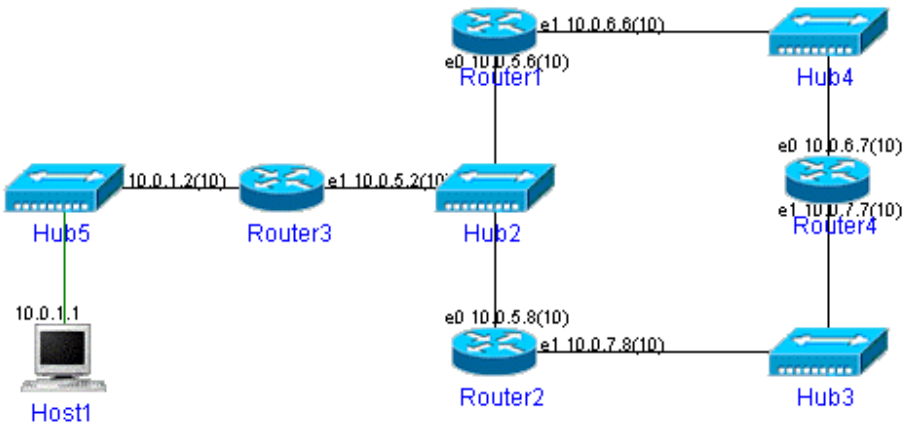


Figura 4.4 Topología OSPF 2

Objetivo

Observar la capacidad que tiene OSPF de hacer balance de carga.

Resultados de laboratorio

Router 2

```
Router2> show ip ospf route

============ OSPF network routing table ============
N    10.0.1.0/24           [20] area: 0.0.0.1
                           via 10.0.5.2, eth0
N    10.0.5.0/24           [10] area: 0.0.0.1
                           directly attached to eth0
N    10.0.6.0/24           [20] area: 0.0.0.1
                           via 10.0.5.6, eth0
                           via 10.0.7.7, eth1
N    10.0.7.0/24           [10] area: 0.0.0.1
                           directly attached to eth1
```

Router 4

```
Router4> show ip ospf route
============ OSPF network routing table ============
N    10.0.1.0/24           [30] area: 0.0.0.1
                           via 10.0.6.6, eth0
                           via 10.0.7.8, eth1
N    10.0.5.0/24           [20] area: 0.0.0.1
                           via 10.0.6.6, eth0
                           via 10.0.7.8, eth1
N    10.0.6.0/24           [10] area: 0.0.0.1
                           directly attached to eth0
N    10.0.7.0/24           [10] area: 0.0.0.1
                           directly attached to eth1
```

Observe en las tablas la presencia de rutas alternas para los routers 1 y 4, como consecuencia de tener el mismo costo para llegar a una red específica. En la topología de red se aprecia que el enrutador 3 no tiene rutas alternas, ya que en este caso solo existe una ruta óptima calculada con el algoritmo de Dijkstra para llegar a cada una de las redes. Con este resultado se demuestra que OSPF tiene la capacidad de tener rutas alternas de igual costo para llegar a un destino.

Base de datos del Estado del Enlace

Router 1

```
Router1> show ip ospf database

       OSPF Router with ID (10.0.5.6)

                Router Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.0.1.2        10.0.1.2         531 0x80000004 0xac33 2
10.0.5.6        10.0.5.6         528 0x80000003 0x109b 2
10.0.5.8        10.0.5.8         534 0x80000006 0x6836 2
10.0.6.7        10.0.6.7         531 0x80000004 0x4e53 2

                Net Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.5.8        10.0.5.8         535 0x80000002 0xb33a
10.0.6.7        10.0.6.7         531 0x80000001 0x9669
10.0.7.7        10.0.6.7         552 0x80000001 0x9f5d
```

Como se dijo anteriormente, esta información se encuentra presente en cada uno de los enrutadores que componen la red y es una copia exacta para cada uno.

Resultados de la herramienta

Router 1

Tipo	Opcion	Red	Mascara	Metrica	Area	NextHop
N		10.0.6.0	255.255.255.0	10	1	Directamente conectada
N		10.0.5.0	255.255.255.0	10	1	Directamente conectada
N		10.0.7.0	255.255.255.0	20	1	10.0.6.7
N		10.0.7.0	255.255.255.0	20	1	10.0.5.8
N		10.0.1.0	255.255.255.0	20	1	10.0.5.2

Router 4

Tipo	Opcion	Red	Mascara	Metrica	Area	NextHop
N		10.0.6.0	255.255.255.0	10	1	Directamente conectada
N		10.0.7.0	255.255.255.0	10	1	Directamente conectada
N		10.0.5.0	255.255.255.0	20	1	10.0.6.6
N		10.0.5.0	255.255.255.0	20	1	10.0.7.8
N		10.0.1.0	255.255.255.0	30	1	10.0.6.6
N		10.0.1.0	255.255.255.0	30	1	10.0.7.8

Base de datos del Estado del Enlace

```
Router Liks States (Area 1)
LinkID       ADVRouter   Age    Seq#   Cksum     Link Count
10.0.1.2     Router3     1173   2      0x1bc     Numero links2
10.0.5.8     Router2     1199   2      0x80bc    Numero links2
10.0.6.6     Router1     1120   2      0x60bc    Numero links2
10.0.6.7     Router4     1162   2      0x14bc    Numero links2

Network Liks States (Area 1)
LinkID       ADV Router  Age    Seq#   Cksum
10.0.1.2     Router3     1144   0      0x14bc
10.0.5.8     Router2     1108   0      0x38bc
10.0.7.8     Router2     1103   0      0x41bc
10.0.6.7     Router4     1156   0      0x42bc
```

Los resultados son iguales y refleja que la herramienta trabaja con múltiples rutas hacia una red.

La base de datos del estado del enlace tiene una Network Link State de más por lo explicado anteriormente.

4.2.3 Laboratorio OSPF 3

Topología de red

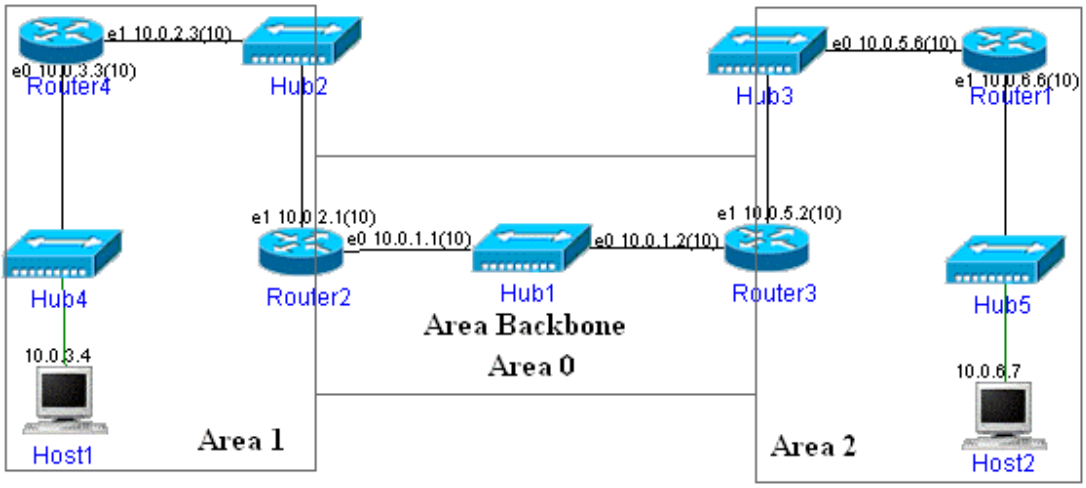


Figura 4.5 Topología OSPF 3

Objetivo

Analizar el comportamiento de OSPF cuando trabaja en múltiples áreas y observar los resultados de la tabla de enrutamiento y de la base de datos del estado del enlace.

Resultados de laboratorio

Router 2

```
Router2# show ip ospf route
============ OSPF network routing table ============
N    10.0.1.0/24           [10] area: 0.0.0.0
                           directly attached to eth0
N    10.0.2.0/24           [10] area: 0.0.0.1
                           directly attached to eth1
N    10.0.3.0/24           [20] area: 0.0.0.1
                           via 10.0.2.3, eth1
N IA 10.0.5.0/24           [20] area: 0.0.0.0
                           via 10.0.1.2, eth0
N IA 10.0.6.0/24           [30] area: 0.0.0.0
                           via 10.0.1.2, eth0

============ OSPF router routing table =============
R    10.0.1.2              [10] area: 0.0.0.0, ABR
                           via 10.0.1.2, eth0
```

Router 4

```
Router4> show ip ospf route
============ OSPF network routing table ============
N IA 10.0.1.0/24           [20] area: 0.0.0.1
                           via 10.0.2.1, eth1
N    10.0.2.0/24           [10] area: 0.0.0.1
                           directly attached to eth1
N    10.0.3.0/24           [10] area: 0.0.0.1
                           directly attached to eth0
N IA 10.0.5.0/24           [30] area: 0.0.0.1
                           via 10.0.2.1, eth1
N IA 10.0.6.0/24           [40] area: 0.0.0.1
                           via 10.0.2.1, eth1

============ OSPF router routing table =============
R    10.0.1.1              [10] area: 0.0.0.1, ABR
                           via 10.0.2.1, eth1
```

En las tablas mostradas, la sigla IA (Inter-Area) significa que estas redes son alcanzadas pasando por otra área. El ABR (Area boundary router) es el enrutador de frontera que se

encuentra ubicado en el backbone. El ABR mantiene una copia de la base de datos del estado del enlace diferente por cada área a la cual está conectado.

Base de datos del Estado del Enlace

Router 4

```
Router4> show ip ospf database

       OSPF Router with ID (10.0.3.3)

                Router Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.0.1.1        10.0.1.1        1378 0x80000004 0x7b96 1
10.0.3.3        10.0.3.3        1378 0x80000003 0xad35 2

                Net Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.2.3        10.0.3.3        1383 0x80000001 0x62bc

                Summary Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.0.1.0        10.0.1.1        1424 0x80000001 0xda61 10.0.1.0/24
10.0.5.0        10.0.1.1        1372 0x80000001 0x131b 10.0.5.0/24
10.0.6.0        10.0.1.1         128 0x80000002 0x6ab7 10.0.6.0/24
```

Como se puede apreciar, el Router 4 solo conoce la base de datos topológica de su área (área 1). En la zona de Summary Link States, se observa que el enrutador conoce las redes que están fuera de su área pero no la base de datos topológica de las mismas.

Router 2

```
Router2# show ip ospf database

                OSPF Router with ID (10.0.1.1)

            Router Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.0.1.1        10.0.1.1        1076 0x80000003 0x5db8 1
10.0.1.2        10.0.1.2        1081 0x80000003 0x5bb7 1

            Net Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.1.2        10.0.1.2        1081 0x80000001 0x63c3

            Summary Link States (Area 0.0.0.0)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.0.2.0        10.0.1.1        1116 0x80000001 0xcf6b 10.0.2.0/24
10.0.3.0        10.0.1.1        1065 0x80000001 0x2907 10.0.3.0/24
10.0.5.0        10.0.1.2         135 0x80000002 0xa68f 10.0.5.0/24
10.0.6.0        10.0.1.2        1079 0x80000001 0x022a 10.0.6.0/24

            Router Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Link count
10.0.1.1        10.0.1.1        1071 0x80000004 0x7b96 1
10.0.3.3        10.0.3.3        1072 0x80000003 0xad35 2

            Net Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum
10.0.2.3        10.0.3.3        1077 0x80000001 0x62bc

            Summary Link States (Area 0.0.0.1)

Link ID         ADV Router      Age  Seq#       CkSum  Route
10.0.1.0        10.0.1.1        1116 0x80000001 0xda61 10.0.1.0/24
10.0.5.0        10.0.1.1        1065 0x80000001 0x131b 10.0.5.0/24
10.0.6.0        10.0.1.1        1065 0x80000001 0x6cb6 10.0.6.0/24
```

Por tratarse de un enrutador que pertenecen al backbone, el Router 2 mantienen una copia de la base de datos topológica de las áreas a las que están conectados (áreas 0 y 1). Al igual que en el caso anterior, este enrutador conoce las redes que se encuentran en otra área, pero no su base de datos topológica completa.

Resultados de la herramienta

Router 2

Tipo	Opcion	Red	Mascara	Metrica	Area	NextHop
N		10.0.2.0	255.255.255.0	10	1	Directamente conectada
N		10.0.3.0	255.255.255.0	20	1	10.0.2.3
N		10.0.1.0	255.255.255.0	10	0	Directamente conectada
R	ABR	Router3		10	0	10.0.1.2
N	IA	10.0.5.0	255.255.255.0	20	0	10.0.1.2
N	IA	10.0.6.0	255.255.255.0	30	0	10.0.1.2

Router 4

Tipo	Opcion	Red	Mascara	Metrica	Area	NextHop
N		10.0.3.0	255.255.255.0	10	1	Directamente conectada
N		10.0.2.0	255.255.255.0	10	1	Directamente conectada
R	ABR	Router2		10	1	10.0.2.1
N	IA	10.0.1.0	255.255.255.0	20	1	10.0.2.1
N	IA	10.0.5.0	255.255.255.0	30	1	10.0.2.1
N	IA	10.0.6.0	255.255.255.0	40	1	10.0.2.1

Base de datos del Estado del Enlace

Router 4

```
      Router Liks States (Area 1)
LinkID     ADVRouter    Age    Seq#  Cksum     Link Count
10.0.3.3   Router4      1115   2     0x31bc    Numero links2
10.0.2.1   Router2      1128   2     0x68bc    Numero links1

               Network Liks States (Area 1)
LinkID      ADV Router   Age    Seq#  Cksum
10.0.3.3    Router4      1119   0     0x54bc
10.0.2.3    Router4      1180   0     0x96bc

              Summary Liks States (Area 1)
LinkID      ADV Router   Age    Seq#  Cksum    Route
10.0.1.0    Router2      1108   0     0x39bc   10.0.1.0
10.0.5.0    Router2      1186   0     0x56bc   10.0.5.0
10.0.6.0    Router2      1129   0     0x15bc   10.0.6.0
```

Router 2

```
Router Liks States (Area 1)
LinkID      ADVRouter   Age    Seq#  Cksum   Link Count
10.0.2.1    Router2     1189   2     0x60bc  Numero links1
10.0.3.3    Router4     1167   2     0x17bc  Numero links2

               Network Liks States (Area 1)
LinkID      ADV Router  Age    Seq#  Cksum
10.0.3.3    Router4     1161   0     0x0bc
10.0.2.3    Router4     1180   0     0x29bc

              Summary Liks States (Area 1)
LinkID      ADV Router  Age    Seq#  Cksum    Route
10.0.1.0    Router2     1130   0     0x16bc  10.0.1.0
10.0.5.0    Router2     1164   0     0x81bc  10.0.5.0
10.0.6.0    Router2     1105   0     0x38bc  10.0.6.0

               Router Liks States (Area 0)
LinkID      ADVRouter   Age    Seq#  Cksum   Link Count
10.0.1.1    Router2     1138   2     0x68bc  Numero links1
10.0.1.2    Router3     1179   2     0x6bc   Numero links1

               Network Liks States (Area 0)
LinkID      ADV Router  Age    Seq#  Cksum
10.0.1.2    Router3     1146   0     0x44bc

              Summary Liks States (Area 0)
LinkID      ADV Router   Age   Seq#   Cksum   Route
10.0.2.0    Router2      1141  0      0x69bc  10.0.2.0
10.0.5.0    Router3      1115  0      0x3bc   10.0.5.0
10.0.3.0    Router2      1166  0      0x52bc  10.0.3.0
10.0.6.0    Router3      1191  0      0x43bc  10.0.6.0
```

Los resultados demuestran la capacidad que tiene la herramienta de trabajar en múltiples áreas para el protocolo OSPF.

4.3 IGRP

4.3.1 Laboratorio IGRP 1

Topología de red

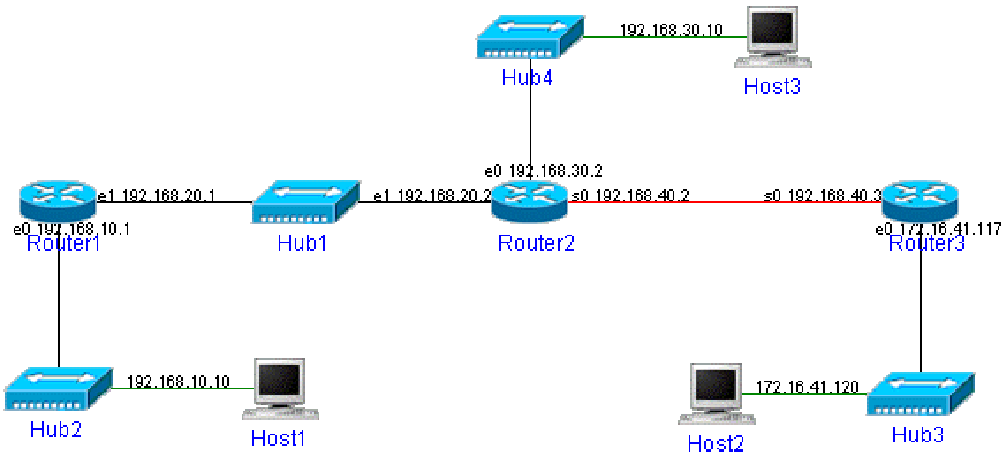


Figura 4.6 Topología IGRP 1

Enrutador	Ancho de banda(Kbps)			Retardo(useg)		
	E0	E1	S0	E0	E1	S0
Router1	10000	10000	-	1000	1000	-
Router2	100000	10000	56	100	1000	20000
Router3	10000	-	1544	1000	-	20000

Tabla 4.1 Configuración IGRP 1

Objetivo

Observar el comportamiento del protocolo IGRP y comprobar las métricas.

Resultados de laboratorio

A continuación se presenta la tabla de enrutamiento de los enrutadores:

Router 1

```
Router1#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

Gateway of last resort is not set

I    192.168.40.0 [100/180671] via 192.168.20.2, 00:00:24, Ether  net1
C    192.168.10.0 is directly connected, Ethernet0
I    192.168.30.0 [100/1110] via 192.168.20.2, 00:00:24, Etherne  t1
C    192.168.20.0 is directly connected, Ethernet1
I    172.16.0.0 [100/180771] via 192.168.20.2, 00:00:25, Etherne  t1
```

Se describirá la tabla de enrutamiento para este enrutador solamente. La interpretación es similar para los demás enrutadores.

- C: indica que la red se encuentra conectada directamente a una interfaz del enrutador.
- I: indica que la red se ha aprendido a través del protocolo IGRP.
- Dirección IP (A.B.C.D): muestra la dirección IP de la red a la que puede alcanzar
- [100/180671]: El primer valor indica la distancia administrativa y el segundo valor indica la métrica usada por IGRP.
- Vía: indica el próximo salto para llegar a una red específica.
- Interfaz: especifica la interfaz por donde aprendió la red.

Análisis de la tabla de enrutamiento

- Como se puede observar, el enrutador 1 tiene dos redes directamente conectadas a sus interfaces, la red 192.168.10.0 y la red 192.168.20.0 En la tabla se refleja este resultado.
- Las demás redes las ha aprendido usando el protocolo IGRP a través de la interfaz e1.
- Para este enrutador el próximo salto hacia otras redes es 192.168.20.2.
- La métrica es calculada por el protocolo a través de la fórmula estudiada en el capítulo 1.

Router 2

```
Router2#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

Gateway of last resort is not set

C    192.168.30.0/24 is directly connected, Ethernet0
I    192.168.10.0/24 [100/1200] via 192.168.20.1, 00:00:16, Ethernet1
C    192.168.40.0/24 is directly connected, Serial0
I    172.16.0.0/16 [100/180671] via 192.168.40.3, 00:00:36, Serial0
C    192.168.20.0/24 is directly connected, Ethernet1
```

Router 3

```
Router2#sh ip route

Codes: I - IGRP derived, R - RIP derived, H - HELLO derived, O - OSPF derived
       C - connected, S - static, E - EGP derived, B - BGP derived

Gateway of last resort is not set

C    192.168.40.0 is directly connected, Serial0
I    192.168.10.0 [100/8676] via 192.168.40.2, 0:01:05, Serial0
I    192.168.30.0 [100/8486] via 192.168.40.2, 0:01:05, Serial0
I    192.168.20.0 [100/8576] via 192.168.40.2, 0:01:05, Serial0
C    172.16.0.0 is directly connected, Ethernet0
```

Resultados de la herramienta

Router 1

Red	Mascara	NextHop	Interfaz	Número de saltos	Métrica
192.168.10.0	255.255.255.0	Directamente conectada	E0	1	1100
192.168.20.0	255.255.255.0	Directamente conectada	E1	1	1100
192.168.30.0	255.255.255.0	192.168.20.2	E1	2	1110
192.168.40.0	255.255.255.0	192.168.20.2	E1	2	180671
172.16.0.0	255.255.0.0	192.168.20.2	E1	3	180771

Router 2

Red	Mascara	NextHop	Interfaz	Número de saltos	Métrica
192.168.30.0	255.255.255.0	Directamente conectada	E0	1	110
192.168.20.0	255.255.255.0	Directamente conectada	E1	1	1100
192.168.40.0	255.255.255.0	Directamente conectada	S0	1	180571
172.16.0.0	255.255.0.0	192.168.40.3	S0	2	180671
192.168.10.0	255.255.255.0	192.168.20.1	E1	2	1200

Router 3

Red	Mascara	NextHop	Interfaz	Número de saltos	Métrica
192.168.40.0	255.255.255.0	Directamente conectada	S0	1	8476
172.16.0.0	255.255.0.0	Directamente conectada	E0	1	1100
192.168.30.0	255.255.255.0	192.168.40.2	S0	2	8486
192.168.20.0	255.255.255.0	192.168.40.2	S0	2	8576
192.168.10.0	255.255.255.0	192.168.40.2	S0	3	8676

Si se comparan los campos importantes como Red, NextHop y Métrica se aprecia el correcto funcionamiento de la herramienta ya que tanto en el laboratorio como en el simulador los resultados son iguales.

4.3.2 Laboratorio IGRP 2

Topología de red

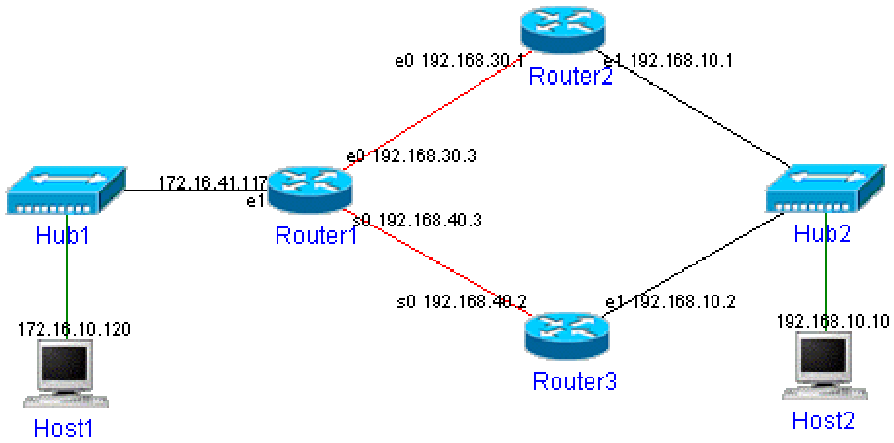


Figura 4.7 Topología IGRP 2

Enrutador	Ancho de banda(Kbps)			Retardo(useg)		
	E0	E1	S0	E0	E1	S0
Router1	10000	100000	10000	1000	100	1000
Router2	10000	10000	-	1000	1000	-
Router3	10000	-	10000	1000	-	1000

Tabla 4.2 Configuración IGRP 2

Objetivo

Observar la capacidad de IGRP de tener más de una ruta hacia una red destino.

Resultados de laboratorio

Router 3

```
Router3# show ip route

Codes: I - IGRP derived, R - RIP derived, H - HELLO derived, O - OSPF derived
       C - connected, S - static, E - EGP derived, B - BGP derived

Gateway of last resort is not set

C    192.168.40.0 is directly connected, Serial0
C    192.168.10.0 is directly connected, Ethernet0
I    192.168.30.0 [100/1200] via 192.168.40.3, 0:00:12, Serial0
                  [100/1200] via 192.168.10.1, 0:00:49, Ethernet0
I    172.16.0.0 [100/1110] via 192.168.40.3, 0:00:12, Serial0
```

Como se observa en la tabla de enrutamiento, este enrutador para alcanzar la red 192.168.30.0 tiene dos caminos, uno a través de la interfaz Ethernet 0 y el otro a través de la interfaz Serial 0 ya que la métricas son iguales y se maneja una varianza de 1.

Router 2

```
Router2#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

Gateway of last resort is not set

I    192.168.40.0 [100/1200] via 192.168.30.3, 00:01:04, Ethernet0
                  [100/1200] via 192.168.10.2, 00:00:42, Ethernet1
C    192.168.10.0 is directly connected, Ethernet1
C    192.168.30.0 is directly connected, Ethernet0
I    172.16.0.0 [100/1110] via 192.168.30.3, 00:01:05, Ethernet0
```

Router 1

```
Router1#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

Gateway of last resort is not set

C    192.168.30.0/24 is directly connected, Ethernet0
I    192.168.10.0/24 [100/1200] via 192.168.30.1, 00:00:24, Ethernet0
                     [100/1200] via 192.168.40.2, 00:00:50, Serial0
C    192.168.40.0/24 is directly connected, Serial0
C    172.16.0.0/16 is directly connected, Ethernet1
```

Resultados de la herramienta

Router 1

Red	Mascara	NextHop	Interfaz	Número de saltos	Métrica
172.16.0.0	255.255.0.0	Directamente conectada	E1	1	110
192.168.30.0	255.255.255.0	Directamente conectada	E0	1	1100
192.168.40.0	255.255.255.0	Directamente conectada	S0	1	1100
192.168.10.0	255.255.255.0	192.168.40.2	S0	2	1200
192.168.10.0	255.255.255.0	192.168.30.1	E0	2	1200

Router 2

Red	Mascara	NextHop	Interfaz	Número de saltos	Métrica
192.168.30.0	255.255.255.0	Directamente conectada	E0	1	1100
192.168.10.0	255.255.255.0	Directamente conectada	E1	1	1100
172.16.0.0	255.255.0.0	192.168.30.3	E0	2	1110
192.168.40.0	255.255.255.0	192.168.30.3	E0	2	1200
192.168.40.0	255.255.255.0	192.168.10.2	E1	2	1200

Router 3

Red	Mascara	NextHop	Interfaz	Número de saltos	Métrica
192.168.30.0	255.255.255.0	Directamente conectada	E0	1	1100
192.168.10.0	255.255.255.0	Directamente conectada	E1	1	1100
172.16.0.0	255.255.0.0	192.168.30.3	E0	2	1110
192.168.40.0	255.255.255.0	192.168.30.3	E0	2	1200
192.168.40.0	255.255.255.0	192.168.10.2	E1	2	1200

Se aprecia que la herramienta soporta múltiples rutas para IGRP.

4.4 BGP_4

4.4.1 Laboratorio BGP 1

Topología de red

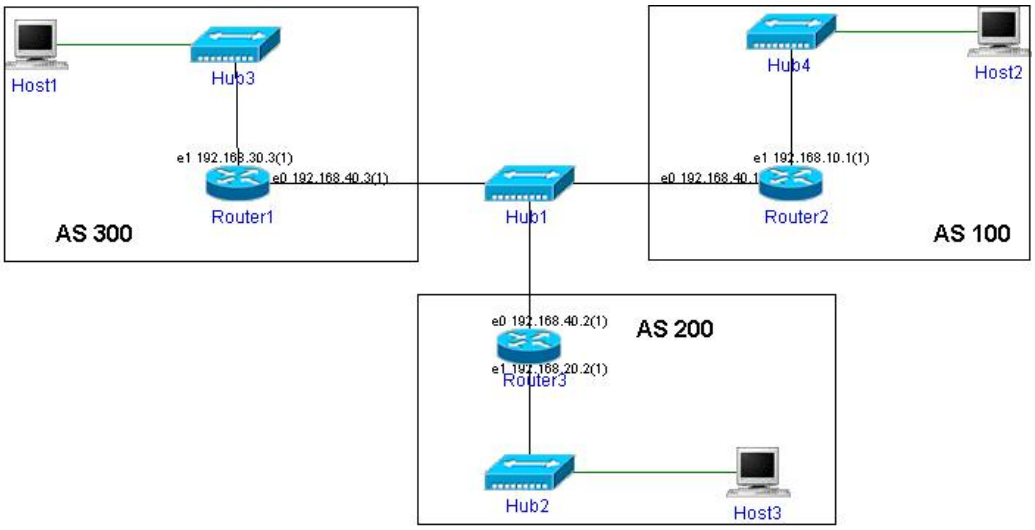


Figura 4.8 Topología BGP-4 1

Objetivo

Analizar el comportamiento del protocolo BGP_4 operando es tres diferentes sistemas autónomos y analizar las tablas de enrutamiento.

Resultados de laboratorio

Router 1

```
Router1#sh ip bgp
BGP table version is 5, local router ID is 192.168.40.3
Status codes: s suppressed, d damped, h history, * valid, > best , i
- internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.10.0     192.168.40.1             0             0 100 i
*> 192.168.20.0     192.168.40.2             0             0 200 i
*> 192.168.30.0     0.0.0.0                  0         32768 i
*> 192.168.40.0     0.0.0.0                  0         32768 i
```

Se describirá la tabla de enrutamiento para este enrutador solamente. La interpretación es similar para los demás enrutadores.

- * : especifica una ruta valida.
- >: la ruta señalada con este indicador especifica la mejor ruta hacia la red destino y es la que utiliza BGP.
- i : indica que es una ruta aprendida a través de IBGP.
- Network: muestra la dirección IP de la red a la que puede alcanzar.
- Next Hop: especifica el próximo salto para alcanzar la red destino.
- Metric (Atributo discriminador de salida múltiple MED): es una indicación a vecinos externos sobre la ruta preferida a un sistema autónomo
- LocPrf (Preferencia local): proporciona una indicación a los routers de sistema autónomo acerca de que ruta es la preferida para salir del sistema autónomo.
- Weight: atributo de peso utilizado por los enrutadores Cisco para definir normas de enrutamientos locales (no se propaga a ningún vecino).
- Path: Sistemas autónomos que se deben atravesar para alcanzar la red destino.

Análisis de la tabla de enrutamiento

- Como se puede observar, el enrutador 1 tiene dos redes directamente conectadas a sus interfaces, la red 192.168.30.0 y la red 192.168.40.0 En la tabla se refleja este resultado con el valor indicado en el campo Next Hop con el valor 0.0.0.0.
- Las demás redes las ha aprendido usando el protocolo BGP a través de los vecinos externos con direcciones 192.168.40.2 y 192.168.40.1.

Router 2

```
Router2# show ip bgp
BGP table version is 6, local router ID is 192.168.40.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.10.0     192.168.40.1             0             0 100 i
*> 192.168.20.0     0.0.0.0                  0         32768 i
*> 192.168.30.0     192.168.40.3             0             0 300 i
*> 192.168.40.0     0.0.0.0                  0         32768 i
```

Router3

```
Router3# sh ip bgp
BGP table version is 0, local router ID is 192.168.40.1
Status codes: s suppressed, d damped, h history, * valid, > best  , i - internal
Origin codes: i  - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Pat  h
*> 192.168.10.0     0.0.0.0                  0         32768 i
*> 192.168.20.0     192.168.40.2                           0 300   200 i
*> 192.168.30.0     192.168.40.3             0             0 300   i
*> 192.168.40.0     0.0.0.0                  0         32768 i
```

Resultados de la herramienta

Router 1

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.40.0	directamente conectada	0	300*	E0
EGP	192.168.10.0	192.168.40.1	0	300*100*	E0
EGP	192.168.20.0	192.168.40.2	0	300*200*	E0
static	192.168.30.0	directamente conectada	0	300*	E1

Router 2

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.40.0	directamente conectada	0	100*	E0
EGP	192.168.30.0	192.168.40.3	0	100*300*	E0
EGP	192.168.20.0	192.168.40.2	0	100*200*	E0
static	192.168.10.0	directamente conectada	0	100*	E1

Router 3

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.40.0	directamente conectada	0	200*	E0
EGP	192.168.30.0	192.168.40.3	0	200*300*	E0
EGP	192.168.10.0	192.168.40.1	0	200*100*	E0
static	192.168.20.0	directamente conectada	0	200*	E1

Comparando los resultados del laboratorio con los arrojados por la herramienta, se puede observar que estos últimos son correctos, presentándose algunas diferencias que se explican a continuación:

- Campo Origen: este campo de la herramienta especifica la vía por donde fue aprendida la ruta, la cual puede ser static (para redes directamente conectadas), IGP (para redes aprendidas vía IBGP) y EGP (para redes aprendidas vía EBGP). En el laboratorio este campo se presenta por el símbolo “i” y solo para redes aprendidas vía IBGP.
- NextHop: las redes que se encuentran directamente conectadas se representan por 0.0.0.0 en el laboratorio, mientras que en la herramienta se utiliza el mensaje “directamente conectada” .
- Path: los resultados del laboratorio omiten en este atributo el número de sistema autónomo del último enrutador que recibe la ruta, la herramienta no hace esta omisión con el fin de presentar unos resultados más claros.
- Interfaz: es un campo que esta presente solo en la herramienta, para mostrar al usuario el nombre de la interfaz que el enrutador debe usar para alcanzar la red que se especifica en el campo red.
- LocPrf : este campo no se muestra en la herramienta, ya que esta trabaja con un valor por defecto de 100 (como lo especifica el RFC 1771).
- Weight: la herramienta no implementa este atributo, ya que no se encuentra definido en el RFC 1771 y solo es implementado por enrutadores Cisco.

4.4.2 Laboratorio BGP 2

Topología de red

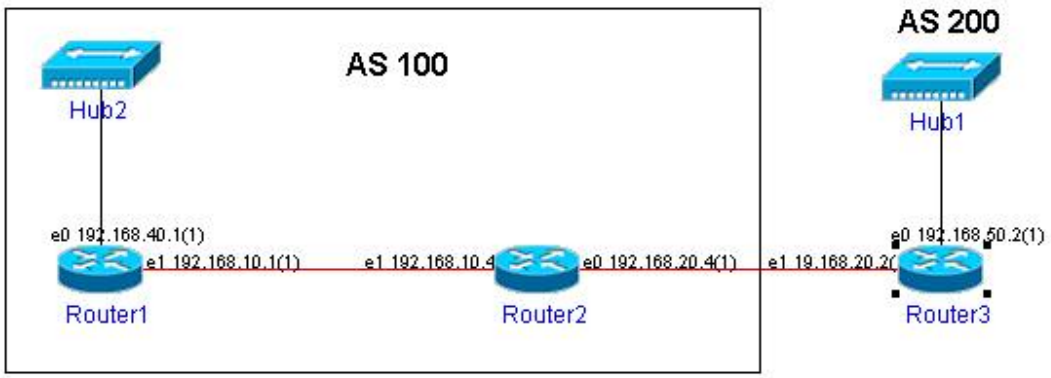


Figura 4.9 Topología BGP-4 2

Objetivo

Estudiar el funcionamiento del protocolo BGP entre sistemas autónomos (EBGP) y dentro de un sistema autónomo (IBGP), así como la sincronización en BGP.

Resultados con sincronización no activada (no synchronization) de laboratorio

En este caso el enrutador acepta las rutas aprendidas a través de IBGP y la anuncia a sus vecinos.

Router 1

```
Router1# sh ip bgp
BGP table version is 0, local router ID is 192.168.40.1
Status codes: s suppressed, d damped, h history, * valid, > best , i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Pat h

*> 192.168.10.0     0.0.0.0                  0         32768 i
*>i192.168.20.0     192.168.10.4             0    100      0 i
*> 192.168.40.0     0.0.0.0                  0         32768 i
* i192.168.50.0     192.168.20.2             0    100      0 200  i
```

De acuerdo a los resultados que se presentan en la tabla, se puede concluir que el software Zebra tiene un defecto en la sincronización BGP, ya que al no estar activada el router debe aceptar entradas aprendidas por IBGP y debe anunciarlas a sus vecinos, pero la red 192.168.50.0 no será anunciada por este enrutador. Este defecto no se presenta en los

enrutadores Cisco, lo que se puede comprobar con los resultados arrojados por los Routers 2 y 3.

Router2

```
Router2#sh ip bgp
BGP table version is 7, local router ID is 192.168.20.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.10.0     0.0.0.0                  0         32768 i
*> 192.168.20.0     0.0.0.0                  0         32768 i
*>i192.168.40.0     192.168.10.1             0    100      0 i
*> 192.168.50.0     192.168.20.2             0             0 200 i
```

Router3

```
Router3#sh ip bgp
BGP table version is 11, local router ID is 192.168.50.2
Status codes: s suppressed, d damped, h history, * valid, > best,i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 192.168.10.0     192.168.20.4           0             0 100 i
*> 192.168.20.0     0.0.0.0                0         32768 i
*> 192.168.40.0     192.168.20.4                         0 100 i
*> 192.168.50.0     0.0.0.0                0         32768 i
```

Vemos como el router 3 aprendió la red 192.168.40.0 debido a que la actualización no estaba activada.

Resultados de la herramienta

Router1

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.40.0	directamente conectada	0	100*	E0
static	192.168.10.0	directamente conectada	0	100*	E1
IGP	192.168.20.0	192.168.10.4	0	100*	E1
IGP	192.168.50.0	192.168.20.2	0	100*200*	E1

Router 2

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.20.0	directamente conectada	0	100*	E0
EGP	192.168.50.0	192.168.20.2	0	100*200*	E0
static	192.168.10.0	directamente conectada	0	100*	E1
IGP	192.168.40.0	192.168.10.1	0	100*	E1

Router 3

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.50.0	directamente conectada	0	200*	E0
static	192.168.20.0	directamente conectada	0	200*	E1
EGP	192.168.10.0	192.168.20.4	0	200*100*	E1
EGP	192.168.40.0	192.168.20.4	0	200*100*	E1

Los resultados que muestra la herramienta son correctos y se puede apreciar que no se presenta el problema de sincronización que tiene el software Zebra.

Resultados con sincronización activada (synchronization)

En este caso el enrutador no debe usar o publicar a un vecino una ruta conocida por IBGP, a menos que esa ruta sea local o conocida desde el IGP.

Router1

```
Router1# sh ip bgp
BGP table version is 0, local router ID is 192.168.40.1
Status codes: s suppressed, d damped, h history, * valid, > best,i- internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Pat
*> 192.168.10.0     0.0.0.0                  0         32768 i
*>i192.168.20.0     192.168.10.4             0    100      0 i
*> 192.168.40.0     0.0.0.0                  0         32768 i
```

En esta tabla de enrutamiento se vuelve a apreciar el defecto del software Zebra ya que la red 192.168.20.0 esta siendo usada por el enrutador a pesar de haber sido aprendida a través de IBGP y no existir un IGP corriendo.

Router2

```
Router2#sh ip bgp
BGP table version is 6, local router ID is 192.168.20.4
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.10.0     0.0.0.0                  0         32768 i
*> 192.168.20.0     0.0.0.0                  0         32768 i
*> 192.168.50.0     192.168.20.2             0             0 200 i
```

En esta tabla de enrutamiento se aprecia como el enrutador a pesar de tener una ruta hacia la red 192.168.40.0 no la usa ni la publica a vecinos porque no se esta ejecutando ningún IGP.

Router 3

```
4000#sh ip bgp
BGP table version is 11, local router ID is 192.168.50.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.10.0     192.168.20.4             0             0 100 i
*> 192.168.20.0     0.0.0.0                  0         32768 i
*> 192.168.50.0     0.0.0.0                  0         32768 i
```

Resultados de la herramienta

Router 1

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.40.0	directamente conectada	0	100*	E0
static	192.168.10.0	directamente conectada	0	100*	E1

Router 2

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.20.0	directamente conectada	0	100*	E0
EGP	192.168.50.0	192.168.20.2	0	100*200*	E0
static	192.168.10.0	directamente conectada	0	100*	E1

Router 3

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.50.0	directamente conectada	0	200*	E0
static	192.168.20.0	directamente conectada	0	200*	E1
EGP	192.168.10.0	192.168.20.4	0	200*100*	E1

La herramienta presenta unos resultados correctos demostrando un buen funcionamiento.

4.4.3 Laboratorio BGP 3

Topología de red

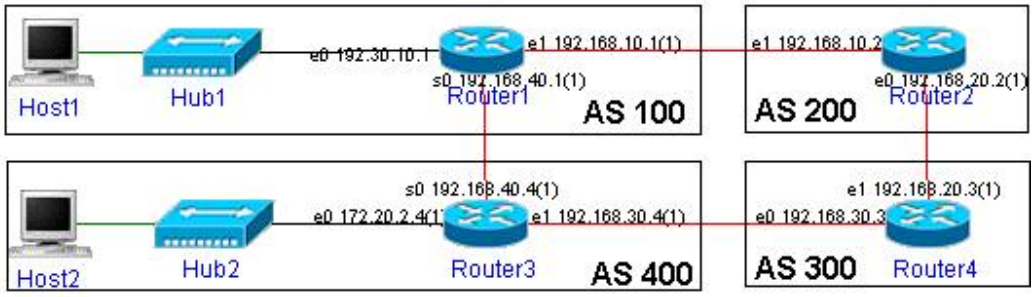


Figura 4.10 Topología BGP-4 3

Objetivo

Observar el trabajo del atributo Met en el proceso de selección de ruta. Met = 0 para todos los enrutadores

Resultados de laboratorio

Router 2

```
Router2#sh ip bgp
BGP table version is 32, local router ID is 192.168.20.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i   - IGP, e  - EGP, ?  - incomplete

   Network          Next Hop          Metric LocPrf Weight Path

*> 172.20.0.0       192.168.20.3                           0 300 400 i
*> 172.30.0.0       192.168.10.1           0               0 100 i
*> 192.168.10.0     0.0.0.0                0           32768 i
*> 192.168.20.0     0.0.0.0                0           32768 i
*> 192.168.30.0     192.168.20.3           0               0 300 i
*> 192.168.40.0     192.168.10.1           0               0 100 i
```

Se puede apreciar en la topología de red usada, que el router tiene 2 rutas a 172.20.0.0 y ambas tienen la misma longitud de ruta de sistema autónomo (hay dos sistemas autónomos en cada ruta). En este caso, siendo todos los demás atributos iguales, el router seleccionará la rutas más antigua. Para el presente caso el router 2 selecciona la ruta con el atributo Next Hop 192.168.20.3 por haber sido aprendida primero.

Resultados de la herramienta

Router 2

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.20.0	directamente conectada	0	200*	E0
EGP	192.168.30.0	192.168.20.3	0	200*300*	E0
EGP	172.20.0.0	192.168.20.3	0	200*300*400*	E0
static	192.168.10.0	directamente conectada	0	200*	E1
EGP	172.30.0.0	192.168.10.1	0	200*100*	E1
EGP	192.168.40.0	192.168.10.1	0	200*100*	E1

Los resultados son similares a los obtenidos en el laboratorio.

Se cambia el atributo Met a 10 en el router3 para las rutas anunciadas al router2 y se ejecuta el comando always-compare-met en el router2.

Resultados de laboratorio

Router 2

```
4000#sh ip bgp
BGP table version is 16, local router ID is 192.168.20.2
Status codes: s suppressed, d damped, h history, * valid, > best , i-internal
Origin codes: i- IGP, e - EGP, ? - incomplete

   Network          Next Hop          Metric LocPrf Weight Path
*> 172.20.0.0       192.168.10.1                           0 100 400 i
*> 172.30.0.0       192.168.10.1             0             0 100 i
*> 192.168.10.0     0.0.0.0                  0         32768 i
*> 192.168.20.0     0.0.0.0                  0         32768 i
*> 192.168.30.0     192.168.20.3            10             0 300 i
*> 192.168.40.0     192.168.10.1             0             0 100 i
```

Como todos los atributos son iguales (LocPrf, Path), el router selecciona la ruta con valor de métrica más bajo, en este caso el router selecciono la ruta hacia la red 172.16.20.0 a través del Next Hop 192.168.10.1 por tener un valor de métrica más bajo.

Resultados de la herramienta

Router 2

origen	red	NextHop	Métrica	Path	Interfaz
static	192.168.20.0	directamente conectada	0	200*	E0
EBGP	192.168.30.0	192.168.20.3	10	200*300*	E0
static	192.168.10.0	directamente conectada	0	200*	E1
EBGP	172.30.0.0	192.168.10.1	0	200*100*	E1
EBGP	192.168.40.0	192.168.10.1	0	200*100*	E1
EBGP	172.20.0.0	192.168.10.1	0	200*100*400*	E1

Con los resultados de la tabla anterior se aprecia el buen funcionamiento que presenta la herramienta en cuanto al uso del atributo metric (métrica) en el proceso de decisión de ruta.

CONCLUSIONES Y RECOMENDACIONES

- El estudio teórico, técnico y práctico en todas las etapas del presente proyecto ayudaron a obtener una herramienta software agradable, confiable y precisa, desarrollada en JAVA para entornos Windows y Linux, que permitirá a los estudiantes, profesores e investigadores analizar el funcionamiento y comportamiento de los protocolos de enrutamiento más importantes en las redes IP.

- El uso de una metodología para el desarrollo de una herramienta software es de vital importancia para la obtención de un producto de calidad que satisfaga las necesidades del cliente.

- Las redes de Petri coloreadas representan una excelente alternativa para modelar sistemas ya que conjugan las primitivas para describir sincronización y procesos concurrentes con las primitivas para definir los tipos de datos y los valores de los datos manipulados.

- Los protocolos de enrutamiento son un elemento fundamental dentro de las redes TCP/IP, ya que garantizan la conectividad y por lo tanto la transmisión confiable de los datos.

- La comprensión del funcionamiento de los protocolos de enrutamiento y las métricas que utilizan son información valiosa para los administradores de la redes, ya que con estas podrán seleccionar y configurar el protocolo que más se adapte a las necesidades de su red. Se recomienda el uso del protocolo RIP o IGRP, en pequeñas redes, debido a las ventajas que presentan en términos de uso de ancho de banda, tiempo de configuración y mantenimiento; además son muy fáciles de implementar. Para redes medianas y grandes se recomienda el uso del protocolo OSPF, ya que permite dividir la red en áreas, incrementando la velocidad de convergencia y evitando el envió constante de paquetes que congestionan la red.

- La herramienta software esta diseñada de tal manera que se puedan introducir a futuro nuevos protocolos de enrutamiento, por lo tanto sería muy fácil actualizarla y aumentar sus capacidades hacia los protocolos de enrutamiento Ipv6 como: RIPng, OSPFng, BGP_4 plus.

BIBLIOBRAFIA

- RCF 2453. “Routing Information Protocol”
- RFC 2328. “Open Shorest Path First”.
- RFC 1771. “Border Gateway Protocol 4”.
- RENDON, Alvaro. “El lenguaje de modelado unificado UML”. Universidad del Cauca. Abril 2000.
- RENDON, Alvaro. “Apuntes sobre el proceso unificado para el desarrollo de programas”. Universidad del Cauca. Abril 2000.
- LARS Kristensen, SOREN Christensen, KURT Jensen. “The practitioner's guide to coloured Petri nets”. University of Aarhus, Denmark.
- TARIFA, Enrique Eduardo. “Teoría de Modelos y Simulación”. Universidad Nacional de Jujuy.
- PAQUET Caterini, TEARE Diane. “Creación de redes cisco escalables”. 2001
- www.java.sun.com
- www.cisco.com

ANEXOS

- Anexo A. Direccionamiento IP.
- Anexo B. Diseño de la herramienta.
- Anexo C. Manual de usuario.
- Anexo D. Código fuente.

CONTENIDO CD-ROM

- Instalador de la herramienta Software
- Monografía del proyecto en .PDF
- Anexos del proyecto en .PDF
- Código fuente de la herramienta.
- Herramientas utilizadas por el proyecto