Packet Tracer[1] es la herramienta de aprendizaje y simulación de redes interactiva para los instructores y alumnos de Cisco CCNA. Esta herramienta les permite a los usuarios crear topologías de red, configurar dispositivos, insertar paquetes y simular una red con múltiples representaciones visuales. Packet Tracer se enfoca en apoyar mejor los protocolos de redes que se enseñan en el currículum de CCNA.

Este producto tiene el propósito de ser usado como un producto educativo que brinda exposición a la interfaz comando – línea de los dispositivos de Cisco para practicar y aprender por descubrimiento.

Packet Tracer 5.3.3 es la última versión del simulador de redes de Cisco Systems, herramienta fundamental si el alumno está cursando el CCNA o se dedica al networking.

En este programa se crea la topología física de la red simplemente arrastrando los dispositivos a la pantalla. Luego clickando en ellos se puede ingresar a sus consolas de configuración. Allí están soportados todos los comandos del Cisco OS e incluso funciona el "tab completion". Una vez completada la configuración física y lógica de la red, también se puede hacer simulaciones de conectividad (pings, traceroutes, etc) todo ello desde las mismas consolas incluidas.

Principales funcionalidades:

- Entre las mejoras del Packet Tracer 5 encontramos:
- Soporte para Windows (2000, XP, Vista) y Linux (Ubuntu y Fedora).
- Permite configuraciones multiusuario y colaborativas en tiempo real.
- Soporte para IPv6, OSPF multi-área, redistribución de rutas, RSTP, SSH y Switchs multicapa.

Soporta los siguientes protocolos:

- HTTP, TCP/IP, Telnet, SSH, TFTP, DHCP y DNS.
- TCP/UDP, IPv4, IPv6, ICMPv4 e ICMPv6.
- RIP, EIGRP, OSPF Multiárea, enrutamiento estático y redistribución de rutas.
- Ethernet 802.3 y 802.11, HDLC, Frame Relay y PPP.
- ARP, CDP, STP, RSTP, 802.1q, VTP, DTP y PAgP, Polly Mkt.

Nuevos recursos, actividades y demostraciones:

- OSPF, IPv6, SSH, RSTP, Frame Relay, VLAN's, Spanning Tree, Mike mkt etc.

Cisco Packet Tracer es un programa de simulación de red de gran alcance que permite a los estudiantes a experimentar con el comportamiento de la red y preguntar "¿qué pasaría si" las preguntas. Como parte integral de la experiencia integral de aprendizaje de Networking Academy, Packet Tracer ofrece simulación, visualización, creación, evaluación y capacidades de colaboración y facilita la enseñanza y el aprendizaje de los conceptos de tecnología compleja.

Packet Tracer son suplementos de equipo físico en el aula, que al permitir a los estudiantes a crear una red con un número casi ilimitado de dispositivos, fomentan la práctica, el descubrimiento y solución de problemas. El entorno de simulación basado en el aprendizaje ayuda a los estudiantes a

desarrollar habilidades del siglo 21, tales como la toma de decisiones, pensamiento creativo y crítico, y solución de problemas.

Packet Tracer complementa los planes de estudio de Networking Academy, permite a los instructores para enseñar y demostrar fácilmente complejos conceptos técnicos y diseño de sistemas de redes. El software está disponible de forma gratuita a todos los instructores de Networking Academy, los alumnos y ex alumnos.

QUE ES GNS3[2]

GNS3 es un simulador grafico que permite procesar simulaciones de redes complejas.

Para ofrecer simulaciones completas y atinadas, GNS3 es fuertemente relacionado con:

- **Dynamips**, Cisco IOS emulador.
- **Dynagen**, un interpretador basado en texto para apoyo de Dynamips.
- **Qemu**, emulador y virtualizador genérico y de código abierto.
- **Virtual Box**, libre y poderoso software de virtualización.

GNS3 es una excelente herramienta complementaria para ingenieros que trabajan con laboratorios reales, administradores y personas interesadas en estudiar para obtener las certificaciones CISCO, tales como: Cisco CCNA, CCNP, CCIP y CCIE como también Juniper JNCIA, JNCIS y JNCIE.

También puede ser usado para experimentar con diferentes detalles de Cisco IOS, Juniper JunOS o para revisar configuraciones que necesitan ser instaladas mas tarde en router reales.

Agradecimientos a integración Virtual Box, en nuestros días cada ingeniero en sistemas y administradores, pueden tomar mucha ventaja de GNS3 para hacer sus propios laboratorios Red Hat (RHCE, RHCT), Microsoft (MSCE, MSCA), Novell (CLP) y muchas otras marcas de vendedores que ofrecen certificaciones.

Este proyecto es código abierto, programas gratis que pueden ser usados e múltiples sistemas operativos, incluyendo Windows, Linux, and MacOS X.

CARACTERÍSTICAS

- ✓ Diseño de topologías de redes de alta calidad y complejidad.
- ✓ Emulación de muchas plataformas de IOS de ruteadores Cisco IOS, IPS, PIX y firewalls ASA, JunOS.
- ✓ Simulación de Ethernet simple, ATM y switches Frame Relay.
- ✓ ¡Conexión de redes simuladas al mundo real!
- ✓ Captura de paquetes utilizando Wireshark.

ENRUTAMIENTO.

Con las máscaras de red podemos comunicar **subredes** entre sí pero esto no es suficiente porque también necesitamos comunicarnos con el mundo exterior. Aquí aparece el concepto de **enrutamiento**. Se conoce como **gateway** o **puerta de enlace** el cual define el punto de nuestra red que se conecta con otras redes.

Cuando nosotros tenemos un **router** en una red conectado a **Internet** lo definimos como *puerta de enlace*. Este dispositivo sirve para conectar dos segmentos de red separados físicamente. Es a través del *router* como vamos redirigiendo los paquetes para hacerlos llegar desde su origen hasta su destino.

Este proceso de enrutado se hace mediante **tablas de enrutamiento** para mantener la info. Acerca de otras *redes y hosts*. Se guarda toda la información de cada nodo de cómo comunicarse con otros *hosts*.

Cuando se va a enviar un *paquete IP* desde una maquina está inserta la dirección origen y la de destino. Luego el equipo busca la *dirección IP* de destino en la **tabla de enrutamiento** siguiendo un orden. Es el siguiente:

- ✓ Primero busca una entrada que corresponda exactamente con la *dirección de IP* destino.
- ✓ Si no encuentra coincidencia se busca por el identificador de red de la dirección destino.
- ✓ Y si no se encuentra esta tampoco busca la ruta predeterminada (0.0.0.0).
- ✓ Sino encuentra coincidencia en la tabla el paquete es descartado.

Para obtener la *tabla de enrutamiento* en Windows escribimos en la consola **"route print".**

La tabla se genera automáticamente basándose en la configuración de *TCP/IP* de la máquina. Podemos añadir rutas con **"route add"** se llama **enrutamiento estático.** Hay otra forma de actualizar esta tabla de forma **dinámica** y se hace mediante algoritmos automáticamente y lo hacen basándose en la comunicación **broadcast** entre **routers** para descubrir las mejores rutas y aquí es donde aparece el concepto de **métrica** que es una medida de lo óptimo que es utilizar una ruta y otra.

Descripción de cada campo:

- **Destino de red:** Es la dirección por la que se va a buscar coincidencia con la dirección **IP de destino.** Con valores entre 0.0.0.0 y 255.255.255.255 para la multidifusion limitada.
- **Mascara de red:** cuando no se encuentra coincidencia exacta en la tabla se aplica la **máscara** de red sobre la dirección de destino. Se utilizan los *bits* coincidentes por ejemplo 0.0.0.0 no sería necesario que ningún bit coincida, y 255.255.255.255 será necesario que coincidan todos.
- **Puerta de enlace:** es la dirección a la que se produce la **redirección** cuando se produce una coincidencia con esa entrada.
- **Interfaz:** la interfaz es la dirección IP configurada en el equipo local para el adaptador de red que se utiliza cuando se envía un *datagrama IP* en la red.
- **Métrica:** nos da información de las rutas disponibles y cuál es la mejor. Siempre se selecciona la que tiene mejor **métrica**. Es el número de **saltos** que un **paquete** tiene que hacer hasta su destino.

Hay unos comandos que nos pueden resultar útiles. Uno es el **ipconfig** para Windows o **ifconfig** si es unix. Nos da la información sobre las diferentes **interfaces** de red que tiene instalada nuestra máquina. Nos muestra la **interfaz**, **la dirección IP, la máscara de subred y la puerta de enlace predeterminada.**

Otro comando seria el **tracert** para Windows **y traceroute** para unix. Con este lo que obtenemos es el número de saltos que un paquete tiene que hacer para llegar al destino que le indicamos en el mismo. Por ejemplo tracert www.google.es. En la pantalla veríamos los diferentes routers por los que nuestro paquete va pasando.

REFERENCIAS

2009, Redes y Seguridad, **ENRUTAMIENTO**, consultado en: agosto 27, 2011, en: http://www.redesyseguridad.es/enrutamiento/

PROTOCOLOS DE ENRUTAMIENTO

En este capítulo introduciremos conceptos básicos de enrutamiento, uno de los puntos clave para obtener la Certificación CCNA, al ser base para comprender muchos detalles del funcionamiento del protocolo TCP/IP a nivel de Red (también conocido como nivel Internet). Hablaremos de los distintos Tipos de Enrutamiento, del Direccionamiento IP (tipos y/o clases de direcciones, etc.), de los Algoritmos utilizados por los Protocolos de Enrutamiento, de los Bucles de Enrutamiento y de los Agujeros Negros, de los Protocolos de Enrutamiento Internos y Externos (de Pasarela), de los Sistemas Autónomos (SA), etc.

A continuación se puede acceder al contenido de las distintas partes del capítulo de Protocolos de Enrutamiento del Manual Cisco CCNA:

- Tipos de Enrutamiento
- Tipos de Direccionamiento y otros conceptos
- Algoritmos de enrutamiento por vector de distancia
- Bucles de Enrutamiento en Algoritmos por Vector de Distancia
- Algoritmos de enrutamiento de estado de enlace
- Sistemas Autónomos
- Protocolos Internos de Pasarela (Interior Gateway Protocols o IGP)
- Protocolos Externos de Pasarela (Exterior Gateway Protocols o EGP)
- Criterios de Selección de Protocolos de Enrutamiento
- La regla de enrutamiento de correspondencia más larga
- Bucles de Enrutamiento y Agujeros Negros

- Resumen de Protocolos de Enrutamiento

TIPOS DE ENRUTAMIENTO

Los protocolos de enrutamiento proporcionan mecanismos distintos para elaborar y mantener las tablas de enrutamiento de los diferentes routers de la red, así como determinar la mejor ruta para llegar a cualquier host remoto. En un mismo router pueden ejecutarse protocolos de enrutamiento independientes, construyendo y actualizando tablas de enrutamiento para distintos protocolos encaminados.

- **Enrutamiento Estático**. El principal problema que plantea mantener tablas de enrutamiento estáticas, además de tener que introducir manualmente en los routers toda la información que contienen, es que el router no puede adaptarse por sí solo a los cambios que puedan producirse en la topología de la red. Sin embargo, este método de enrutamiento resulta ventajoso en las siguientes situaciones:
    - Un circuito poco fiable que deja de funcionar constantemente. Un protocolo de enrutamiento dinámico podría producir demasiada inestabilidad, mientras que las rutas estáticas no cambian.
    - Se puede acceder a una red a través de una conexión de acceso telefónico. Dicha red no puede proporcionar las actualizaciones constantes que requiere un protocolo de enrutamiento dinámico.
    - Existe una sóla conexión con un solo ISP. En lugar de conocer todas las rutas globales, se utiliza una única ruta estática.

- Un cliente no desea intercambiar información de enrutamiento dinámico.

- **Enrutamiento Predeterminado**. Es una ruta estática que se refiere a una conexión de salida o Gateway de "último recurso". El tráfico hacia destinos desconocidos por el router se envía a dicha conexión de salida. Es la forma más fácil de enrutamiento para un dominio conectado a un único punto de salida. Esta ruta se indica como la red de destino **0.0.0.0/0.0.0.0**.

- **Enrutamiento Dinámico**. Los protocolos de enrutamiento mantienen tablas de enrutamiento dinámicas por medio de mensajes de actualización del enrutamiento, que contienen información acerca de los cambios sufridos en la red, y que indican al software del router que actualice la tabla de enrutamiento en consecuencia. Intentar utilizar el enrutamiento dinámico sobre situaciones que no lo requieren es una pérdida de ancho de banda, esfuerzo, y en consecuencia de dinero.

TIPOS DE DIRECCIONAMIENTO Y OTROS CONCEPTOS

Para el diseño de arquitectura de cualquier red, es también muy importante conocer y utilizar los siguientes conceptos, con el fin de optimizar y simplificar el direccionamiento y el tamaño de las tablas de enrutamiento. Gracias a la utilización de estas técnicas, los datos reales a principios de 2000 mostraban que el tamaño de la tabla de enrutamiento era aproximadamente de 76000 rutas.

- **Direccionamiento con Clase**. Es también conocido como Direccionamiento IP básico. Siguiendo este modelo de direccionamiento, a una dirección IP únicamente se le puede asignar su máscara predeterminada o máscara natural. Esto supone muy

poca flexibilidad, y no es recomendable salvo para redes locales muy pequeñas.

- **Subnetting**. La técnica de subnetting, permite dividir una red en varias subredes más pequeñas que contienen un menor número de hosts. Esto nos permite adquirir, por ejemplo, un red de clase B, y crear subredes para aprovechar este espacio de direcciones entre las distintas oficinas de nuestra empresa. Esto se consigue alterando la máscara natural, de forma que al añadir unos en lugar de ceros, hemos ampliado el número de subredes y disminuido el número de hosts para cada subred.

- **Máscara de Subred de Longitud Variable (VLSM)**. Utilizar protocolos de enrutamiento y dispositivos que soporten VLSM, nos permite poder utilizar diferentes máscaras en los distintos dispositivos de nuestra red, lo cual no es más que una extensión de la técnica de subnetting. Mediante VLSM, podemos dividir una clase C para albergar dos subredes de 50 máquinas cada una, y otra subred con 100 máquinas. Es importante tener en cuenta que RIP1 e IGRP no suportan VLSM.

- **Supernetting o Agregación**. La técnica de supernetting o agregación, permite agrupar varias redes en una única superred. Para esto se altera la máscara de red, al igual que se hacía en subnetting, pero en este se sustituyen algunos unos por ceros. El principal beneficio es para las tablas de enrutamiento, disminuyendo drásticamente su tamaño. Un dominio al que se le ha asignado un rango de direcciones tiene la autoridad exclusiva de la agregación de sus direcciones, y debería agregar todo lo que sea posible siempre y cuando no introduzca ambigüedades, lo cual es posible en el caso de redes con interconexiones múltiples a distintos proveedores.

- **Notación CIDR**. La notación CIDR, permite identificar una dirección IP mediante dicha dirección, seguida de una barra y un número que identifica el número de unos en su máscara. Así, se presenta una forma de notación sencilla y flexible, que actualmente es utilizada en la configuración de gran cantidad de dispositivos de red. Un ejemplo sería: 194.224.27.00/24.

- **Traducción de Dirección de Red (NAT)**. La tecnología NAT permite a las redes privadas conectarse a Internet sin recurrir a la re numeración de las direcciones IP. El router NAT se coloca en la frontera de un dominio, de forma que cuando un equipo de la red privada se desea comunicar con otro en Internet, el router NAT envía los paquetes a Internet con la dirección pública del router, y cuando le responden reenvía los paquetes al host de origen. Para realizar esto, basta con relacionar los sockets abiertos desde el equipo NAT a los equipos de la red privada, con los sockets abiertos desde el equipo NAT a los equipos de Internet, así como modificar las cabeceras de los paquetes reenviados. Al igual que Cisco provee NAT es su sistema operativo IOS, otros muchos routers también lo ofrecen, como también es el caso de paquetes de software como Windows 2000, Microsoft Proxy, WinGate, etc.

- **Convergencia**. La convergencia se refiere al tiempo que tardan todos los routers de la red en actualizarse en relación con los cambios que se han sufrido en la topología de la red.

Todas las interfaces operativas conectadas al router se sitúan en la tabla de enrutamiento. Por ello, si sólo hay un router en la red, éste tiene información sobre todas las redes o subredes diferentes y no hay necesidad de configurar un enrutamiento estático o dinámico.

ALGORITMOS DE ENRUTAMIENTO POR VECTOR DE DISTANCIA

El término vector de distancia se deriva del hecho de que el protocolo incluye un vector (lista) de distancias (número de saltos u otras métricas) asociado con cada destino, requiriendo que cada nodo calcule por separado la mejor ruta para cada destino. Los envían mensajes actualizados a intervalos establecidos de tiempo, pasando toda su tabla de enrutamiento al router vecino más próximo (routers a los que está directamente conectado), los cuales repetirán este proceso hasta que todos los routers de la red están actualizados. Si un enlace o una ruta se vuelve inaccesible justo después de una actualización, la propagación del fallo en la ruta se iniciará en la próxima propagación, ralentizándose la convergencia. Los protocolos de vector de distancia más nuevos, como EIGRP y RIP-2, introducen el concepto de **actualizaciones desencadenadas**. Éstas propagan los fallos tan pronto ocurran, acelerando la convergencia considerablemente. Los protocolos por vector de distancia tradicionales trabajan sobre la base de actualizaciones periódicas y contadores de espera: si no se recibe una ruta en un cierto periodo de tiempo, la ruta entra en un estado de espera, envejece y desaparece, volviéndose inalcanzable.

BUCLES DE ENRUTAMIENTO EN ALGORITMOS POR VECTOR DE DISTANCIA

Los bucles de enrutamiento producen entradas de enrutamiento incoherentes, debido generalmente a un cambio en la topología. Si un enlace de un router A se vuelve inaccesible, los routers vecinos no se dan cuenta inmediatamente, por lo que se corre el riego de que el router A crea que puede llegar a la red perdida a través de sus vecinos que mantienen entradas antiguas. Así, añade una nueva entrada a su tabla de enrutamiento con un coste superior. A su vez, este proceso se repetiría una y otra vez, incrementándose el coste de las rutas, hasta que de alguna

forma se parase dicho proceso. Los métodos utilizados para evitar este caso son los que siguen:

- **Horizonte Dividido**. La regla del horizonte dividido es que nunca resulta útil volver a enviar información acerca de una ruta a la dirección de dónde ha venido la actualización original.

- **Actualización Inversa**. Cuando una red de un router falla, este envenena su enlace creando una entrada para dicho enlace con coste infinito. Así deja de ser vulnerable a actualizaciones incorrectas proveniente de routers vecinos, donde esté involucrada dicha red. Cuando los routers vecinos ven que la red ha pasado a un coste infinito, envían una actualización inversa indicando que la ruta no está accesible.

- **Definición de Máximo**. Con este sistema, el protocolo de enrutamiento permite la repetición del bucle hasta que la métrica exceda el valor máximo permitido. Una vez que la red alcanza ese máximo, se considera inalcanzable.

- **Actualización desencadenada**. Normalmente, las nuevas tablas de enrutamiento se envían a los routers vecinos a intervalos regulares. Una actualización desencadenada es una nueva tabla de enrutamiento que se envía de forma inmediata, en respuesta a un cambio. El router que detecta el cambio envía inmediatamente un mensaje de actualización a los routers adyacentes que, a su vez, generan actualizaciones desencadenadas para notificar el cambio a todos sus vecinos. Sin embargo surgen dos problemas:

    - Los paquetes que contienen el mensaje de actualización podrían ser descartados o dañados por algún enlace de la red.

- Las actualizaciones desencadenadas no suceden de forma instantánea. Es posible que un router que no haya recibido aún la actualización desencadenada genere una actualización regular que cause que la ruta defectuosa sea insertada en un vecino que hubiese recibido ya la actualización.

Combinando las actualizaciones desencadenadas con los temporizadores se obtiene un esquema que permite evitar estos problemas

ALGORITMOS DE ENRUTAMIENTO DE ESTADO DE ENLACE

Utiliza un modelo de base de datos distribuida y replicada. Los routers intercambian paquetes de estado de enlace que informa a todos los routers de la red sobre el estado de sus distintos interfaces. Esto significa que sólo se envía información acerca de las conexiones directas de un determinado router, y no toda la tabla de enrutamiento como ocurre en el enrutamiento por vector de distancia. Aplicando el algoritmo SPF (primero la ruta más corta), más conocido como algoritmo Dijkstra, cada router calcula un árbol de las ruta más cortas hacia cada destino, situándose a sí mismo en la raíz. Los protocolos de estado de enlace no pueden proporcionar una solución de conectividad global, como la que se requiere en grandes redes como Internet, pero si son utilizados por muchos proveedores como protocolo de enrutamiento en el interior de un SA. Los protocolos más conocidos son OSPF e IS-IS. Algunos de los beneficios de estos protocolos son:

- No hay límite en el número de saltos de una ruta. Los protocolos del estado de enlace trabajan sobre la base de las métricas de enlace en lugar de hacerlo en función del número de saltos.
- El ancho de banda del enlace y los retrasos puede ser factorizados cuando se calcule la ruta más corta hacia un destino determinado.

- Los cambios de enlace y nodo son inmediatamente introducidos en el dominio mediante actualizaciones del estado de enlace.

- Soporte para VLSM y CIDR, ya que intercambian información de máscara en las actualizaciones.

SISTEMAS AUTÓNOMOS

Un Sistema Autónomo (SA) es un conjunto de redes, o de routers, que tienen una única política de enrutamiento y que se ejecuta bajo una administración común, utilizando habitualmente un único IGP. Para el mundo exterior, el SA es visto como una única entidad. Cada SA tiene un número identificador de 16 bits, que se le asigna mediante un Registro de Internet (como RIPE, ARIN, o APNIC), o un proveedor de servicios en el caso de los SA privados. Así, conseguimos dividir el mundo en distintas administraciones, con la capacidad de tener una gran red dividida en redes más pequeñas y manipulables. En un POP dónde se junten varios SA, cada uno de estos utilizará un router de gama alta que llamaremos **router fronterizo**, cuya función principal es intercambiar tráfico e información de rutas con los distintos routers fronterizos del POP. Así, un concepto importante de comprender es el **tráfico de tránsito**, que no es más que todo tráfico que entra en un SA con un origen y destino distinto al SA local.

En Internet, la IANA es la organización que gestiona las direcciones IP y números de AS, teniendo en cuenta que cada Sistema Autónomo se identifica por un número inequívoco que no puede ser superior a 65535, teniendo en cuenta que la colección 65412-65535 son SA privados para ser utilizados entre los proveedores y los clientes. Así, podemos ponernos en contacto con RIPE, ARIN o APNIC para solicitar rangos de direcciones IP o números de AS.

- **SA de conexión única, sin tránsito**. Se considera que un SA es de conexión única cuando alcanza las redes exteriores a través de un único punto de salida. En este caso disponemos de varios métodos por los cuales el ISP puede aprender y publicar las rutas del cliente.
    - Una posibilidad para el proveedor es enumerar las subredes del cliente como entradas estáticas en su router, y publicarlas a Internet a través de BGP.
    - Alternativamente, se puede emplear un IGP entre el cliente y el proveedor, para que el cliente publique sus rutas.
    - El tercer método es utilizar BGP entre el cliente y el proveedor. En este caso, el cliente podrá registrar su propio número SA, o bien utilizar un número de SA privado si el proveedor tiene soporte para ello.
- **SA de múltiples conexiones, sin tránsito**. Un SA puede tener múltiples conexiones hacia un proveedor o hacia varios proveedores, sin permitir el pasó de tráfico de tránsito a través de él. Para ello, el SA sólo publicará sus propias rutas y no propagará las rutas que haya aprendido de otros SA. Los SA sin tránsito y con múltiples conexiones no necesitan realmente ejecutar BGP con sus proveedores, aunque es recomendable y la mayor parte de las veces es requerido por el proveedor.
- **SA de múltiples conexiones, con tránsito**. Esto es un SA con más de una conexión con el exterior, y que puede ser utilizado para el tráfico de tránsito por otros SA. Para ello, un SA de tránsito publicará las rutas que haya aprendido de otros SA, como medio para abrirse al tráfico que no le pertenezca. Es muy aconsejable (y en la mayoría de los casos requerido) que los SA de tránsito de múltiples conexiones

utilicen BGP-4 para sus conexiones a otros SA, mientras que los routers internos pueden ejecutar enrutamiento predeterminado hacia los routers BGP.

PROTOCOLOS INTERNOS DE PASARELA (INTERIOR GATEWAY PROTOCOLS O IGP)

Se encargan del enrutamiento de paquetes dentro de un dominio de enrutamiento o sistema autónomo. Los IGP, como **RIP** o **IGRP**, se configuran en cada uno de los routers incluidos en el dominio.

- **Routing Information Protocol (RIP)**. RIP es un protocolo universal de enrutamiento por vector de distancia que utiliza el número de saltos como único sistema métrico. Un salto es el paso de los paquetes de una red a otra. Si existen dos rutas posibles para alcanzar el mismo destino, RIP elegirá la ruta que presente un menor número de saltos. RIP no tiene en cuenta la velocidad ni la fiabilidad de las líneas a la hora de seleccionar la mejor ruta. RIP envía un mensaje de **actualización del enrutamiento cada 30 segundos** (tiempo predeterminado en routers Cisco), en el que se incluye toda la tabla de enrutamiento del router, utilizando el protocolo UDP para el envío de los avisos. **RIP-1 está limitado a un número máximo de saltos de 15, no soporta VLSM y CIDR, y no soporta actualizaciones desencadenadas**. RIP-1 puede realizar **equilibrado de la carga** en un máximo de seis rutas de igual coste. **RIP-2** es un protocolo sin clase que admite CIDR, VLSM, resumen de rutas y seguridad mediante texto simple y autenticación MD5. **RIP publica sus rutas sólo a los routers vecinos**.

- **Open Short Path First (OSPF)**. OSPF es un protocolo universal basado en el algoritmo de estado de enlace, desarrollado por el IETF

para sustituir a RIP. Básicamente, OSPF utiliza un algoritmo que le permite calcular la distancia más corta entre la fuente y el destino al determinar la ruta para un grupo específico de paquetes. **OSPF soporta VLSM**, ofrece **convergencia rápida**, **autenticación** de origen de ruta, y publicación de ruta mediante multidifusión. **OSPF publica sus rutas a todos los routers del mismo área**. En la RFC 2328 se describe el concepto y operatividad del estado de enlace en OSPF, mientras que la implementación de OSPF versión 2 se muestra en la RFC 1583. OSPF toma las decisiones en función del corte de la ruta, disponiendo de una **métrica máxima de 65535**.

- OSPF funciona dividiendo una intranet o un sistema autónomo en unidades jerárquicas de menor tamaño. Cada una de estas áreas se enlaza con un área backbone mediante un router fronterizo. Así, todos los paquetes direccionados desde un área a otra diferente, atraviesan el área backbone. OSPF envía Publiciones del Estado de Enlace (Link-State Advertisement – LSA) a todos los routers pertenecientes a la misma área jerárquica mediante **multidifusión IP**. Los routers vecinos intercambian mensajes **Hello** para determinar qué otros routers existen en una determinada interfaz y sirven como mensajes de actividad que indican la accesibilidad de dichos routers. Cuando se detecta un router vecino, se intercambia información de topología OSPF. La información de la LSA se transporta en paquetes mediante la capa de transporte OSPF (con acuse de recibo) para garantizar que la información se distribuye adecuadamente. Para la configuración de OSPF se requiere un número de proceso, ya que se pueden ejecutar distintos procesos OSPF en el mismo routers. Los administradores acostumbran usar un número de SA como número de proceso

- **Interior Gateway Protocol (IGRP)**. IGRP fue diseñado por Cisco a mediados de los ochenta, para corregir algunos de los defectos de RIP y para proporcionar un mejor soporte para redes grandes con enlaces de diferentes anchos de banda, siendo un protocolo **propietario de Cisco**. IGRP es un protocolo de enrutamiento por vector de distancia capaz de utilizar hasta **5 métricas distintas** (ancho de banda K1, retraso K3, carga, fiabilidad, MTU), utilizándose por defecto únicamente el ancho de banda y el retraso. Estas métrica pueden referirse al ancho de banda, a la carga (cantidad de tráfico que ya gestiona un determinado router) y al coste de la comunicación (los paquetes se envían por la ruta más barata). Para la configuración de OSPF se **requiere un número de proceso**, ya que se pueden ejecutar distintos procesos OSPF en el mismo routers. Los administradores acostumbran usar un número de SA como número de proceso. IGRP envía mensajes de actualización del enrutamiento a intervalos de tiempo mayores que RIP, utiliza un formato más eficiente, y **soporta actualizaciones desencadenadas**. IGRP posee un número máximo predeterminado de 100 saltos, que puede ser configurado hasta 255 saltos, por lo que puede implementarse en grandes interconexiones donde RIP resultaría del todo ineficiente. IGRP puede mantener hasta un máximo de **seis rutas paralelas de coste diferente**; Por ejemplo, si una ruta es tres veces mejor que otra, se utilizará con una frecuencia tres veces mayor. **IGRP no soporta VLSM**. **IGRP publica sus rutas sólo a los routers vecinos**.

- **Enhaced IGRP - EIGRP**. Basado en IGRP y como mejora de este, es un protocolo híbrido que pretende ofrecer las ventajas de los protocolos por vector de distancia y las ventajas de los protocolos de estado de enlace. **EIGRP soporta VLSM** y soporta una

convergencia muy rápida. **EIGRP publica sus rutas sólo a los routers vecinos**. Para la configuración de OSPF se **requiere un número de proceso**, ya que se pueden ejecutar distintos procesos OSPF en el mismo routers. Los administradores acostumbran usar un número de SA como número de proceso.

PROTOCOLOS EXTERNOS (EXTERIOR GATEWAY PROTOCOLS O EGP)

Los protocolos de enrutamiento exterior fueron creados para controlar la expansión de las tablas de enrutamiento y para proporcionar una vista más estructurada de Internet mediante la división de dominios de enrutamiento en administraciones separadas, llamadas **Sistemas Autónomos (SA)**, los cuales tienen cada uno sus propias políticas de enrutamiento. Durante los primeros días de Internet, se utilizaba el protocolo **EGP** (no confundirlo con los protocolos de enrutamiento exterior en general). NSFNET utilizaba EGP para intercambiar información de accesibilidad entre el backbone y las redes regionales. Actualmente, **BGP-4** es el estándar de hecho para el enrutamiento entre dominios en Internet.

- **Border Gateway Protocol (BGP)**. Es un protocolo de enrutamiento por vector de distancia usado comúnmente para enrutar paquetes entre dominios, estándar en Internet. BGP gestiona el enrutamiento entre dos o más routers que sirven como routers fronterizos para determinados Sistemas Autónomos. BGP versión 4 (BGP-4), es el protocolo de enrutamiento entre dominios elegido en Internet, en parte porque administra eficientemente la agregación y la propagación de rutas entre dominios. Aunque BGP-4 es un protocolo de enrutamiento exterior, también puede utilizarse dentro de un SA como un conducto para intercambiar actualizaciones BGP. Las conexiones BGP dentro de un SA son denominadas **BGP interno (IBGP)**, mientras que las conexiones BGP entre routers fronterizos

(distintos SA) son denominadas **BGP externo (EBGP)**. BGP-1, 2 y 3 están obsoletos. Para la configuración de OSPF **se requiere un número de Sistema Autónomo**, ya que se pueden ejecutar distintos procesos OSPF en el mismo routers. BGP se especifica en las RFC 1163, 1267 y 1771, que definen las veriones 2, 3 y 4 de BGP, respectivamente.

- Los routers BGP se configuran con la información del vecino a fin de que puedan formar una conexión TCP fiable sobre la que transportar información de la ruta de acceso del sistema autónomo y la ruta de la red. Tras establecer una sesión BGP entre vecinos, ésta sigue abierta a menos que se cierre específicamente o que haya un fallo en el enlace. Si dos routers vecinos intercambian información de ruta y sesiones BGP, se dice que son **iguales BGP**. En principio, los iguales BGP intercambian todo el contenido de las tablas de enrutamiento BGP. Posteriormente, sólo se envían actualizaciones incrementales entre los iguales para avisarles de las rutas nuevas o eliminadas.

- Todas las rutas BGP guardan el último número de versión de la tabla que se ha publicado a sus iguales, así como su propia versión interna de la tabla. Cuando se recibe un cambio en un igual, la versión interna se incrementa y se compara con las versiones de los iguales, para asegurar que todos los iguales se mantienen sincronizados. BGP también guarda una tabla de rutas BGP independiente que contiene todas las rutas de acceso posibles a las redes publicadas.

- Los iguales BGP se dividen en dos categorías: Los iguales BGP de distintos sistemas autónomos que intercambian información de enrutamiento son **iguales BGP externos (EBGP)**. Los iguales BGP del mismo sistema autónomo que intercambian información de enrutamiento son **iguales BGP internos (IBGP)**.

- La selección de ruta óptima BGP se basa en la longitud de la ruta de acceso del sistema autónomo para una ruta de red. La longitud se define como el número de sistemas autónomos distintos necesarios para acceder a la red. Cuanto menor sea la distancia, más apetecible será la ruta de acceso. A través del uso de controles administrativos, BGP es uno de los protocolos de enrutamiento más flexibles y totalmente configurables disponibles.

- Un uso típico de BGP, para una red conectada a Internet a través de varios ISP, es el uso de EBGP con los ISP, así como el uso de IBGP en la red interna, para así ofrecer una óptima selección de rutas. Las redes conocidas de otros sistemas autónomos a través de EBGP se intercambiarán entre los iguales IBGP. Si sólo hubiera un ISP, valdría con utilizar una ruta resumen o predeterminada para la salida a internet.

- Tenga en cuenta que los routers BGP publican las rutas conocidas de un igual BGP a todos sus otros iguales BGP. Por ejemplo, las rutas conocidas a través de EBGP con un ISP se volverán a publicar a los iguales IBGP, que a su vez volverán a publicarlos a otros ISP a través de EBGP. Mediante la publicación reiterada de rutas, la red puede pasar a ser una red de tránsito entre los proveedores con los que se conecte. BGP puede parametrizarse tanto para que la red interna actúe como una red de tránsito, como para que no.

CRITERIOS DE SELECCIÓN DE PROTOCOLOS DE ENRUTAMIENTO

- **Topología de Red**. Los protocolos del tipo OSPF e IS-IS requieren un modelo jerárquico formado un backbone y una o varias áreas lógicas, lo que nos puede llegar a exigir que rediseñemos la red.

- **Resumen de Ruta y Dirección**. Mediante VLSM podemos reducir considerablemente el número de entradas en la tabla de enrutamiento, y en consecuencia la carga de los routers, por lo que son recomendados protocolos como OSPF y EIGRP.

- **Velocidad de Convergencia**. Uno de los criterios más importantes es la velocidad con la que un protocolo de enrutamiento identifica una ruta no disponible, selecciona una nueva y propaga la información sobre ésta. Protocolos como RIP-1 e IGRP suelen ser más lentos en converger que protocolos como EIGRP y OSPF.

- **Criterios de Selección de Ruta**. Cuando las diferentes rutas de la Intranet se compongan de varios tipos de medios LAN y WAN, puede ser desaconsejable un protocolo que dependa estrictamente del número de satos, como es el caso de RIP. RIP considera que el salto de un router en un segmento Fast Ethernet tiene el mismo coste que un salto por un enlace WAN a 56 Kbps.

- **Capacidad de ampliación**. Los protocolos de vector de distancia consumen menos ciclos de CPU que los protocolos de estado de enlace con sus complejos algoritmos SPF. Sin embargo, los protocolos de estado de enlace consumen menos ancho de banda que los protocolos de vector de distancia.

- **Sencillez de implementación**. RIP, IGRP, y EIGRP no requieren mucha planificación ni organización en la topología para que se puedan ejecutar de manera eficaz. OSPF e IS-IS requieren que se haya pensado muy cuidadosamente la topología de la red y los modelos de direccionamiento antes de su implementación.

- **Seguridad**. Algunos protocolos como OSPF y EIGRP admiten poderosos métodos de autenticación, como las autenticación de claves MD5.

- **Compatibilidad**. Teniendo en cuenta el carácter propietario de Cisco de protocolos como IGRP y EIGRP, dichos protocolos no los podremos utilizar con protocolos de distintos fabricantes.

Así, si estamos desarrollando una red compuesta exclusivamente de dispositivos Cisco, no tendremos ninguna duda en utilizar **EIGRP** como protocolo de enrutamiento, por ser sencillo de configurar, no requerir una topología específica, admitir VLSM, y ofrecer una convergencia rápida.

LA REGLA DE ENRUTAMIENTO DE CORRESPONDENCIA MÁS LARGA

Un router que tenga que decidir entre dos prefijos de longitudes diferentes de la misma red siempre seguirá la máscara más larga (es decir, la ruta de red más específica). Suponga, por ejemplo, que un router tiene las dos entradas siguientes en su tabla de enrutamiento.

- 192.32.1.0/24 por la ruta 1.

- 192.32.0.0/16 por la ruta 2.

Cuando intenta enviar tráfico al host 192.32.1.1, el router lo intentará pasar por la ruta 1. Si la ruta 1 no estuviese disponible por alguna razón, entonces lo pasaría por la ruta 2.

BUCLES DE ENRUTAMIENTO Y AGUJEROS NEGROS

Un **bucle de enrutamiento** se produce cuando el tráfico circula hacia atrás y hacia delante entre elementos de la red, no alcanzando nunca su destino final. Suponga que la conexión entre el ISP1 y su cliente Foonet (dónde existe la red 192.32.1.0/24) se vuelve inaccesible. Suponga también que el

ISP1 tiene una ruta predeterminada 0.0.0.0/0 que apunta a ISP2 para las direcciones no conocidas. El tráfico hacia 192.32.1.1 no encontrará su destino en ISP1, por lo que seguirá la ruta predeterminada hacia ISP2, volviendo a ISP1 y a ISP2, y así una y otra vez.

Un **agujero negro** ocurre cuando el tráfico llega y se para en un destino que no es el destino propuesto y desde el que no puede ser reenviado.

Estas dos situaciones tienden a ocurrir cuando se dispone de tablas de enrutamiento gestionadas en una parte por protocolos de enrutamiento, y en otra por rutas estáticas, así como por una incorrecta agregación de rutas de otros proveedores.

RESUMEN DE PROTOCOLOS DE ENRUTAMIENTO

	RIP-1	RIP-2	IGRP	EIGRP	OSPF	BGP
¿Soporta VLSM?	NO	SI	NO	SI	SI	SI
Velocidad Convergencia	Lenta	Media	Media	Rápida	Rápida	Rápida
Tecnología	Vector	Vector	Vector	Mixto	Enlace	Vector
Número max. Saltos	15	15	255	255	65535	
Seguridad		MD5		MD5	MD5	
Selección de Ruta	Saltos	Saltos	Varias Métricas	Varias Métricas	Ancho Banda	

Compatibilidad	Universal	Universal	Cisco	Cisco	Universal	Universal
Tipo	IGP	IGP	IGP	IGP	IGP	EGP
¿Proceso / ASN?	NO	NO	PROCESO	PROCESO	PROCESO	ASN
¿Despende de Topología?	NO	NO	NO	NO	SI	NO

REFERENCIA.

http://www.guillesql.es/Articulos/Manual_Cisco_CCNA_Protocolos_Enrutamiento.aspx

ALGORITMOS DE ENRUTAMIENTO Y TEORIA DE GRAFICAS.[3]

Introducción. Existen varios tipos de algoritmos de encaminamiento:

- **Algoritmos de camino más corto:** Cada nodo decide cuál es el camino más corto hacia un destino, en función de la información de control que recibe de otros nodos de la red. Estos algoritmos minimizan el coste o distancia de la ruta que une dos nodos cualesquiera. Por ejemplo, si la métrica utilizada es el número medio de saltos, el algoritmo de camino más corto será el que minimice este número de saltos entre los nodos que pretendemos conectar.

En concreto, veremos los algortimos de Dijkstra, de Floyd - Marshall, de Bellman - Ford y de Bellman - Ford distribuido.

- **algoritmos aislados:** Los nodos no intercambian información de control explícitamente. Podemos distinguir dos clases de algoritmos para un encaminamiento de este tipo, que son algoritmos de aprendizaje y algoritmos de inundación. Ejemplos concretos de este tipo de algoritmos hay muchos, como el llamado algoritmo de Hot Potato,

- **algoritmos de difusión:** Permiten hacer llegar un paquete a todos los nodos de una red. Este procedimiento encuentra una aplicación directa para un encaminamiento basado en estado de enlaces, puesto que la información sobre los estados de los enlaces se *difunde* a toda la red, y en general, lo que se hará será mandar paquetes a todos los nodos y marcarlos para deshabilitarlos si vuelven a pasar (para evitar bucles). Veremos **cuatro métodos** de conseguir la difusión:

    - **1er método:** consiste en enviar paquetes a todos los demás nodos (tráfico unicast). Este método es poco eficiente porque

por un mismo nodo pasan varios paquetes iguales, además de que no elimina el problema de la formación de bucles.

- **2º método:** hace una inundación: se manda un paquete de un nodo origen a todos los demás. Este tampoco es un buen método, porque consume muchos recursos y también puede hacer que aparezcan bucles.

- **3er método:** Deshabilitamos algunos enlaces de forma que todos los nodos estén conectados pero sin bucles. Así, al inundar se consigue tener muy pocos envíos. Es una solución bastante usada, aunque la forma de acordar entre todos los nodos de la red qué enlaces se deshabilitan es complicada.

- **4º método:** se hace un **reenvío por ruta hacia atrás** (*Reverse Path Forwarding)*. No requiere calcular el árbol de expansión. Los nodos tienen calculadas las rutas, y cuando tienen que ayudar a difundir no siempre se hace inundación: se mira la dirección origen del paquete; si llega por el enlace que utiliza el nodo para ir al origen inunda; si llega por otra ruta el nodo considera que el paquete está dando vueltas y lo descarta.

Notación. *Distancia o coste de un canal:* es una medida de la calidad de un enlace en base a la métrica que se haya definido (por ej., para el método de Gerla es la sensibilidad al aumento de tráfico; otras veces es coste económico, probabilidad de error, etc.).

$_{ij}$ es la distancia del enlace entre dos nodos 'i' y 'j' contiguos. Si no existe enlace entre dichos nodos, valdrá infinito. Si 'i' es igual a 'j' (se trata del mismo nodo), esta distancia será nula.

D_{ij} = D[i,j] es la distancia entre dos nodos no contiguos.

P[i,j] = (K_1 = i, ..., K_n = j) es el Path o camino tal que

$$d[P[i,j]] = \sum_{n=1}^{m-1} d_{K_n K_{n+1}} = D[i,j]$$

Si P[i,j] es un camino mínimo y P[r,t] es un subcamino de P[i,j] , éste también será mínimo".

Algoritmo de Dijkstra. Es un algoritmo iterativo que debe ejecutarse para todos y cada uno de los nodos de la red. Este algoritmo, aplicado a un nodo, tiene una complegidad del orden de N^2 operaciones, por lo que aplicado a los N nodos de la red, resultará tener una complejidad del orden de N^3 operaciones (siendo N el número de nodos de la red), por lo que es facil imaginar que no se trata de un algoritmo eficiente para redes de gran tamaño..

- Para un nodo 's', calcula el camino más corto con origen 's' y destino el resto de los nodos de la red *(arbol de divergencia)*.
- Di es la distancia del camino que va de nodo 's' al nodo 'i'.
- P es el conjunto de los nodos que tienen una etiqueta de distancia (Di) que es permanente.
- T es el conjunto de los nodos que no están en P.

A cada nodo se le asigna una **etiqueta**, que es un indicador de la distancia del nodo origen al nodo en cuestión. Así, se hace una **partición de la red**: se crea un conjunto de nodos con *etiqueta permanente* (conjunto **P**) y un conjunto de nodos con *etiqueta tentativa* (conjunto **T**), es decir, un

conjunto de nodos cuya etiqueta puede cambiar en las siguientes iteraciones.

Pasos del algoritmo:

1.- PASO 0 (INICIACIÓN):

$$D_s = 0$$
$$D_j = d_{s,j} \forall j <> s$$
$$P = \{s\}$$
$$T = \{resto\_nodos\}$$

2.- PASO 1:

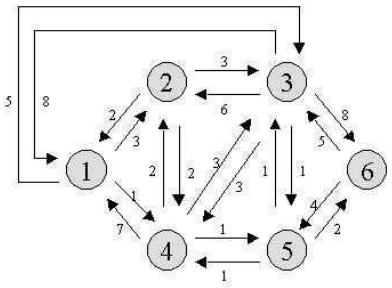
- Encontramos $i \notin P | D_i = \min_{j \notin P} \{D_j\}$. Puede existir empate, es decir, dos caminos igual de cortos, eligiéndose uno de ellos arbitrariamente o de acuerdo con un criterio marcado (no se soporta balanceo de carga).
- $P = P \cup \{i\}$
- Si P contiene a todos los nodos, se para. Si no, continuamos con el paso 3:

3.- PASO 2 (ACTUALIZACIÓN):

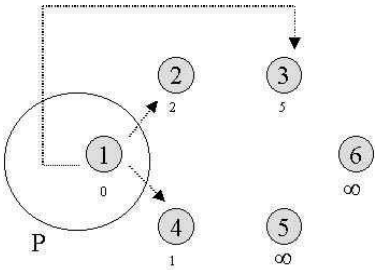
$$\forall j \notin P, D_j = \min\{D_j, d_{i,j} + D_i\}$$

4.- SALTO AL PASO 1.

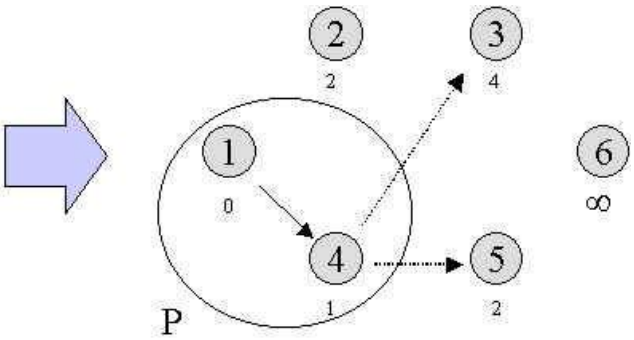
Veamos, para mayor claridad del método, un ejemplo:



Tenemos la siguiente red:



Para s = 1:



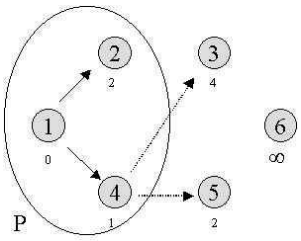
(i=4): añadimos nodo 'i' a P. Como P no contiene a todos los nodos 3, 5 y 6:

Nodo 3: $min\{5, 1+3\} = 4$

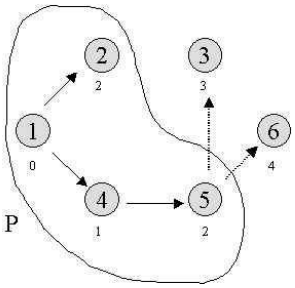
Nodo 6: $min\{\infty, \infty\} = \infty$

Nodo 5: $min\{\infty, 1+1\} = 2$

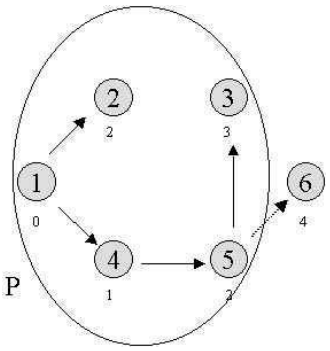
Vemos que tenemos empate entre los nodos 2 y 5, luego elegimos uno al azar: i=2:



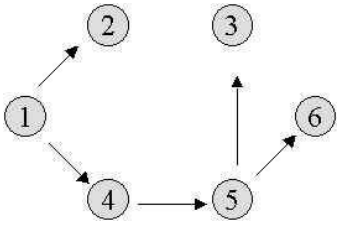
Seguimos con (i=5):



Ahora con (i=3):



El árbol final de encaminamiento del nodo 1 al resto de los nodos de la red queda:



No obtiene directamente las tablas de encaminamiento, sino las distancias entre nodos (árbol de encaminamiento), las cuales se tendrán que traducir a unas tablas de encaminamiento en todos los nodos. Para nuestro ejemplo, en el nodo 1, tendríamos para el caso de encaminamiento fuente y encaminamiento salto a salto las siguientes tablas, respectivamente:

2	1 2
3	1 4 5 3
4	1 4
5	1 4 5
6	1 4 5 6

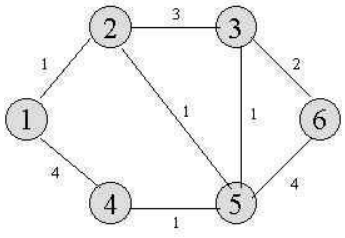
(fuente)

2	2
3	4
4	4
5	4
6	4

(salto a salto)

El algoritmo de Dijkstra puede utilizarse como algoritmo estático y también como algoritmo dinámico centralizado, que consistiría en que hubiera un nodo central al que todos los demás nodos de la red enviaran información de su estado y del de los enlaces que salen de ese nodo. El nodo central recalcularía las distancias de todos los canales, aplicaría el algoritmo de Dijkstra y mandaría las nuevas tablas de encaminamiento a todos los nodos de la red.

Veamos otro ejemplo (i = 1):



El resultado, por pasos, es el siguiente:

1.- P = {1}; T = {2, 3, 4, 5, 6}; D1K = {0, 1, -, 4, -, -}

2.- P = {1,2}; T = {3, 4, 5, 6}; D1K = {0, 1, 4, 4, 2, -}

(NOTA: las dos nuevas distancias las conozco a través del nodo que acabo de incluir en P)

3.- P = {1, 2, 5}; T = {3, 4, 6}; D1K = {0,1,3,3,2,6}

4.- P = {1, 2, 5, 4}; T = {3,6}; D1K = {0, 1, 3, 3, 2, 6}

5.- P = {1, 2, 5, 4, 3}; T = {6}; D1K = {0, 1, 3, 3, 2, 5}

6.- P = {1,2,5,4,3,6}; T = {}; D1K = {0,1,3,3,2,5}

Luego la mínima distancia del nodo 1 al resto es: D1K = {0, 1, 3, 3, 2, 5}

LOS ALGORITMOS DE RUTEO.

Los algoritmos de ruteo son aquellos que construyen la tabla de ruteo, a partir de la cual luego se construye la tabla de forwardeo. La red se puede ver como un grafo donde cada ruteador es un nodo (aunque como nodos también se pueden considerar los hosts, los switches, segmentos de red) y donde cada enlace entre ruteadores es un eje. Viéndolo de este modo, el ruteo se reduce al cálculo de los caminos mínimos de un nodo a los demás.

Algoritmo estático. Dada una red, se pueden calcular los caminos mínimos una única vez y guardar en cada nodo dicha información. Eso tiene sus problemas: no considera la adición de nuevos nodos, ni la caída de los mismos, ni cambios de costos en los ejes.

Algoritmos dinámicos

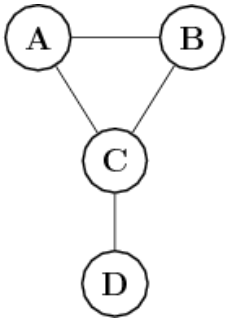
Algoritmo Vector Distancia (RIP). Cada nodo construye un vector con su distancia a cada uno de los demás nodos. Aunque se use el término

distancia, en realidad nos referimos a un costo, que puede estar dado por la distancia pero también por el retardo en milisegundos, por el número total de paquetes encolados esperando ser enviados a ese destino, etc.

Se asume en principio que cada nodo conoce la distancia a sus vecinos. Si la métrica empleada es efectivamente la distancia entonces será de un Salto; si es el número total de paquetes encolados entonces el ruteador solo tiene que examinar la cola para ese destino; si es el retardo en milisegundos entonces el ruteador puede determinarlo mediante un ping.

Al principio, cada nodo construye su tabla poniendo la distancia correspondiente para sus nodos vecinos e infinito para los que no lo son.

Pasemos a verlo con un ejemplo, en este y en los demás, por simplicidad, el costo estará dado por la distancia. Sea la siguiente red:



Para A:

DESTINO	COSTO	SIGUIENTE SALTO
B	1	B
C	1	C

D	INF	…

Para B:

DESTINO	COSTO	SIGUIENTE SALTO
A	1	A
C	1	C
D	INF	…

Para C:

DESTINO	COSTO	SIGUIENTE SALTO
A	1	A
C	1	B
D	1	D

Para D:

DESTINO	COSTO	SIGUIENTE SALTO

A	INF	…
B	INF	…
C	1	C

Luego, cada nodo le manda su vector (el vector es de la forma (Destino, Costo), no incluye el Siguiente Salto) a sus vecinos quienes a su vez le mandan el suyo. Es mediante la comparación entre su propio vector y los enviados por los vecinos que cada nodo va determinando el camino mínimo hacia todos los demás nodos. En cada comparación, en caso de hallar un camino de menor costo, el nodo actualiza su vector.

Volviendo al ejemplo, supongamos que C es el primero en enviarle su vector a A. Como resultado, A decide acceder a D mediante C ya que 2 Saltos (1 salto para ir de A a C más 1 salto para ir de C a D) es menor que la distancia actual de saltos infinitos. Sin embargo, A no va a acceder a B mediante C porque en ese caso le llevaría 2 saltos, que es mayor que la distancia actual a B de 1 salto. La tabla de A entonces quedaría:

Para A:

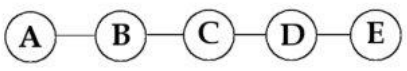
DESTINO	COSTO	SIGUIENTE SALTO
B	1	B
C	1	C
D	2	C

Si más adelante se coloca un nuevo nodo E entre C y D, ahora la distancia entre C y D será de 2 saltos. Cuando C le envíe su vector a A, A cambiara la distancia actual a D de 2 saltos por una distancia de 3 saltos, aun cuando esta última mayor, ya que A accede a D mediante C.

Cada nodo ira actualizando su vector hasta que finalmente todos serán consistentes entre sí, este proceso se denomina CONVERGENCIA. Es importante destacar que cada nodo solamente conoce su distancia a los demás nodos pero desconoce por completo las distancias entre los otros nodos, en otras palabras, no tiene una perspectiva global (más adelante veremos que "Link-State" (OSPF) si la tiene).

Hay dos circunstancias bajo las cuales un nodo puede mandarle su vector al vecino. La primera es por una actualización periódica (periodic update). Aun cuando nada cambio, esto puede servir para hacerle saber a sus vecinos que aún sigue allí. La segunda es por una actualización disparada (triggered update) causada por un evento en la red que hizo que el nodo modifique su vector, el envío a los vecinos se debe a que estos también podrían tener que cambiar los suyos. Uno de estos eventos en la red puede deberse a la caída de un nodo, que será advertida por sus vecinos ya sea porque no envió su actualización periódica o porque estos están continuamente probando el enlace entre ambos. De esta manera, los nodos se mantienen al tanto de la situación de la red.

Sea la siguiente red:



Supongamos que el nodo A acaba de caer y que hay un gong gigante (*) que suena periódicamente haciendo que se produzca un intercambio simultáneo de vectores entre nodos vecinos:

Tabla de distancias a A:

INTERCAMBIOS	NODO B	NODO C	NODO D	NODO E
NINGUNO	1	2	3	4
1	3	2	3	4
2	3	4	3	4
3	5	4	5	4
4	5	6	5	6
5	7	6	7	6
6	7	8	7	8
	…	…	…	…
	INF	INF	INF	INF

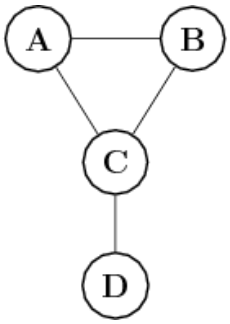
En el primer intercambio B recibe la tabla de C pero no la de A. Asume que el enlace con A esta caído, o sea que la distancia a A es de saltos infinitos. Como la distancia de C a A es de 2 saltos, decide acceder a A mediante C en 3 saltos. El problema es que C accede a A mediante B y esto B no lo sabe (B conoce sus propios saltos siguientes pero desconoce los siguientes saltos de C). En el próximo intercambio, C aumenta su distancia a A a 4 saltos ya que

accede a A mediante B y este está a una distancia de 3 saltos. Siguiendo así sucesivamente, todos los nodos terminan con una distancia de saltos infinitos a A. Este problema se conoce como count-to-infinity.

Una forma de abordarlo puede ser determinar un valor numérico para del infinito lo más bajo posible para terminar cuanto antes con el ciclo. Una buena elección es el diámetro de la red más uno.

Otra forma, que evita parcialmente los ciclos, es el "split horizon", donde los nodos no intercambian con sus vecinos aquellos destinos cuyos saltos siguientes son los mismos vecinos. En el ejemplo, C no hubiera intercambiado (A, 2) con B. Una variante de esto último es el "split horizon with poison reverse", donde se intercambian todos los destinos pero cuando los saltos siguientes son los mismos vecinos, se pone un valor de infinito como costo. En el ejemplo, C hubiera intercambiado (A, INF) con B.

Esto último es efectivo solo cuando el ciclo involucra dos nodos pero falla en el caso general como puede verse en el siguiente ejemplo:



Inicialmente A y B tienen ambos una distancia a D de 2 saltos. Supongamos que D se cae y nuevamente consideremos el gong gigante que determina periódicamente intercambios simultáneos. Usando "Split horizon", ni A ni B intercambian (D,2) con C. Así, C determina que la distancia a D es de infinitos Saltos y reporta esto a A y B. Pero A ve que B tiene una distancia de 2 saltos a D y B ve que A tiene una distancia de 2 saltos a D, ambos

asumen una distancia a D de 3 saltos. En el próximo intercambio su distancia a D será de 4 saltos, esto se repite una y otra vez hasta que su distancia a D es de saltos infinitos, justamente el comportamiento que queríamos evitar.

Algoritmo Estado de Enlace "Link State" (OSPF). OSPF es un protocolo "link-state", que consiste en enviar información del costo de los enlaces a sus vecinos (acá vecinos no se refiere a los vecinos de un nodo, sino del dominio). Si esta información se distribuye sobre todos los nodos, es posible conocer la estructura completa de la red. Cada nodo lleva un mapa completo de la red, y por cada mensaje que llega lo usa para calcular por sí mismo el camino más corto.

Desbordamiento "Reliable flooding". Es el proceso de asegurarse que todos los nodos que participan consiguen una copia del estado del enlace de todos los otros nodos. Consiste en que un nodo envía la información de todos sus enlaces por todos sus vecinos, a su vez los vecinos regresan o "forwardean" está información a sus propios vecinos.

Al igual que RIP, OSPF intercambia información periódicamente y cuando hay algún cambio con sus enlaces vecinos.

Comparación de los algoritmos de ruteo. Los protocolos de ruteo Vector Distancia son simples y eficientes en una red pequeña, y requieren poca si acaso nula administración. Sin embargo, los algoritmos nativos de Vector Distancia no realizan buenas escalas (debido al problema de cuenta a infinito "count-to-infinity problem"), y por tanto, tienen propiedades de convergencia pobre, lo que nos guía a un desarrollo de algoritmos más complejos pero escalables en el uso de redes grandes, como protocolos de ruteo de estado de enlace "link-state routing protocols".

La primera ventaja de ruteo de estado de enlace, "link-state routing" es que estos reaccionan más rápido, y en cantidades medibles de tiempo, para los cambios de conectividad. También los paquetes de Estado de Enlace que se envían sobre la red son más pequeños que los utilizados en el ruteo de Vector Distancia. El ruteo de Vector Distancia requiere una tabla entera de ruteo de nodos que se van a transmitir, mientras que los de Estado de Enlace solo requiere de información del nodo siguiente. Sin embargo, esos paquetes son más pequeños y no utilizan recursos de la red significativos en su transmisión. La primera desventaja del ruteo de Estado de Enlace es que requiere más almacenamiento y mayor capacidad de procesamiento que un simple ruteo de Vector Distancia.

(*) Cita extraída del Tanenbaum

Biografía EDGER DIJKSTRA[4]

Dijkstra estudió física teórica en la Universidad de Leiden. Trabajó como investigador para Burroughs Corporation a principios de los años 1970. En la Universidad de Texas en Austin, Estados Unidos, ocupó el *Schlumberger Centennial Chair in Computer Sciences*. Se retiró en 2000.

Entre sus contribuciones a la informática está el problema del camino más corto, también conocido como el algoritmo de Dijkstra, la notación polaca inversa y el relacionado algoritmo shunting yard, THE multiprogramming system, el algoritmo del banquero y la construcción del semáforo para coordinar múltiples procesadores y programas. Otro concepto debido a Dijkstra, en el campo de la computación distribuida, es el de la auto-estabilización, una vía alternativa para garantizar la confiabilidad del sistema. El algoritmo de Dijkstra es usado en *la ruta más corta primero*

(SPF) que es usado en el protocolo de enrutamiento Open Shortest Path First (OSPF). También se le debe la autoría de la expresión "Crisis del software", aparecida en su libro The Humble Programmer y usada ampliamente en la famosa reunión de la OTAN de 1968 sobre desarrollo del software. Recibió el Premio Turing en 1972.

Era conocido por su baja opinión de la sentencia GOTO en programación, que culminó en 1968 con el artículo *Go To Statement Considered Harmful*, visto como un paso importante hacia el rechazo de la expresión GOTO y de su eficaz reemplazo por estructuras de control tales como el bucle while. El famoso título del artículo no era obra de Dijkstra, sino de Niklaus Wirth, entonces redactor de Comunicaciones del ACM. Dijkstra era un aficionado bien conocido de ALGOL, y trabajó en el equipo que desarrolló el primer compilador para este lenguaje. En ese mismo año creó el primer sistema operativo con estructura jerárquica, de niveles o capas. Fue denominado THE (Technische Hogeschool, Eindhoven) que se utilizó con fines didácticos.

Desde los años 1970, el principal interés de Dijkstra fue la verificación formal. La opinión que prevalecía entonces era que uno debe primero escribir un programa y seguidamente proporcionar una prueba matemática de su corrección. Dijkstra objetó que las pruebas que resultan son largas e incómodas, y que la prueba no da ninguna comprensión de cómo se desarrolló el programa. Un método alternativo es la derivación de programas, «desarrollar prueba y programa conjuntamente». Uno comienza con una especificación matemática del programa que se supone va a hacer y aplica transformaciones matemáticas a la especificación hasta que se transforma en un programa que pueda ser ejecutado. El programa que resulta entonces es sabido correcto por la construcción. Muchos de los últimos trabajos de Dijkstra tratan sobre las maneras de hacer fluida la argumentación matemática.

Respecto a su caracter árido y ácido, conocidas son su oposición a la instrucción GO TO y al lenguaje BASIC ("mutila la mente más allá de toda recuperación"). Alan Kay expuso que "en informática, la arrogancia se mide en nanodijkstras".

Dijkstra murió el 6 de agosto de 2002 después de una larga lucha contra el cáncer.

PROTOCOLOS DE ENRUTAMIENTO: ENRUTAMIENTO ESTÁTICO

Como había dicho en la parte 1, el enrutamiento estático, es creado manualmente a diferencia de los protocolos dinámicos, que se intercambian las tablas de enrutamiento mediante actualizaciones periódicas.

Para crear una ruta estática, es en modo configuración global, y el comando IOS es el siguiente:

ip route [ip red destino][mascara de subred][ip siguiente salto]

Ejemplo: router (config)#ip route 200.0.0.0 255.255.255.0 130.0.0.2

O también puede ser:

ip route [ip red destino][mascara de subred][interfaz de salida]

Ejemplo: router(config)#IP route 200.0.0.0 255.255.255.0 s0/2

* Interfaz de salida, se refiere a la interfaz del router local, que está conectado a las redes externas

RUTAS ESTATICAS POR DEFECTO. Las rutas estáticas por defecto, son una ruta estática especial, se crean para enrutar el tráfico de internet, ya que es imposible crear rutas a hacia las redes que están en Internet.

¿Y CÓMO FUNCIONAN? Cualquier IP de una red destino que el router no tenga ninguna coincidencia en su tabla de enrutamiento, este ocupará la ruta por defecto y mandara el paquete hacia donde se le indicó en esta. Las rutas estáticas por defecto también se crean en modo configuración Global.

La estructura es la siguiente:

ip route 0.0.0.0 0.0.0.0 [IP interfaz siguiente salto]

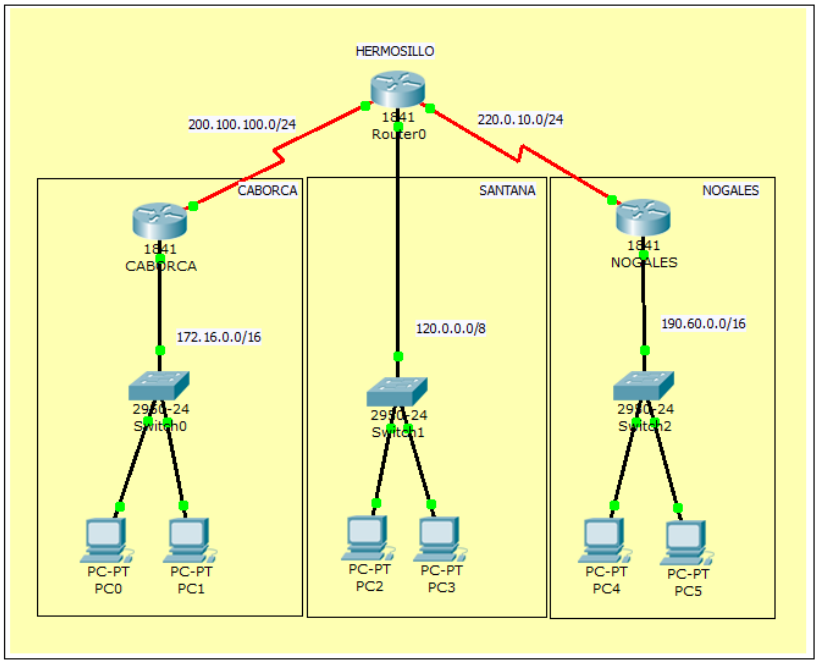
Ejemplo:router(config)#ip route 0.0.0.0 0.0.0.0 120.0.0.2

O también puede ser;

ip route 0.0.0.0 0.0.0.0 [interfaz de salida]

Ejemplo: router(config)#ip route 0.0.0.0 0.0.0.0 s0/1

EJEMPLO PRÁCTICO



TOPOLOGÍA A UTILIZAR PARA ENRUTAMIENTO ESTÁTICO

CONFIGURANDO ROUTER CABORCA. En este router voy a aplicar una ruta estática por defecto, ya que siempre los paquetes van a tener una salida que es el router santana, por lo tanto , cualquiera que sea la IP de la red destino esta ruta por defecto la enviara hacia dicho router, bien vamos con la configuración.

caborca>enable
caborca#configure terminal

caborca(config)#ip route 0.0.0.0 0.0.0.0 200.100.100.2

Aquí ya he creado la ruta por defecto, en mi caso estoy acostumbrado a ocupar la IP del salto siguiente (200.100.100.2), pero también en vez de esa IP podría haber ido ahi la interfaz local del router caborca en este caso la serial serial0/0. Si vemos la tabla de enrutamiento del router caborca, con el comando show ip route veremos lo siguiente:

caborca>enable
caborca#show ip route

C 172.16.0.0/16 is directly connected, FastEthernet0/0

C 200.100.100.0/24 is directly connected, Serial0/0

S* 0.0.0.0/0 [1/0] via 200.100.100.2

Las rutas que tienen una "C", son las interfaces directamente conectadas al router, y la "S*", indica que es una ruta estática por defecto, que es la ruta que creamos, por lo tanto cualquier paquete hacía una IP destino que el router no tenga en esta tabla de enrutamiento, lo enviara hacía la interfaz 200.100.100.2.

CONFIGURANDO ROUTER SANTANA. Este router tendrá que tener una ruta estática para saber llegar tanto a la LAN de router caborca como la de nogales.

Vamos a crear la configuración:

santana>enable
santana#configure terminal

santana(config)#ip route 172.16.0.0 255.255.0.0 200.100.100.1

santana(config)#ip route 190.60.0.0 255.255.0.0 220.0.10.1

He creado las 2 rutas estáticas:

La primera es para llegar a la LAN del router caborca (IP 172.16.0.0/16), que saldrá por la IP de la interfaz del salto siguiente o sea 200.100.100.1 (Teniendo en cuenta que estamos parados en el router santana).

La segunda es para llegar a la LAN del router nogales (IP 190.60.0.0/16), que saldrá por la IP de la interfaz del salto siguiente o sea 220.0.10.1 (Teniendo en cuenta que estamos parados en el router santana).

Si aplicamos el comando show ip route en modo de privilegio. Veremos:

santana>enable
santana#show ip route

C 120.0.0.0/8 is directly connected, FastEthernet0/0

S 172.16.0.0/16 [1/0] via 200.100.100.1

S 190.60.0.0/16 [1/0] via 220.0.10.1

C 200.100.100.0/24 is directly connected, Serial0/0

C 220.0.10.0/24 is directly connected, Serial0/1

Vemos la IP de la 3 interfaces conectadas directamente, y también las rutas estáticas creadas para que el router santana "sepa" llegar a las REDES (LAN) de caborca y nogales.

CONFIGURANDO ROUTER NOGALES. Por último nos queda configurar el router nogales, que es el mismo caso del router caborca, o sea una ruta por defecto, ya que siempre va a tener como salida el router santana, por lo tanto con la ruta por defecto le decimos que cualquier IP que no conozca la envíe hacia el router santana

La configuración sería:

```
nogales>enable
nogales#configure terminal
nogales(config)#ip route 0.0.0.0 0.0.0.0 220.0.10.2
```

Y para ver la tabla de enrutamiento:

```
nogales>enable
nogales#show ip route
```

C 190.60.0.0/16 is directly connected, FastEthernet0/0

C 220.0.10.0/24 is directly connected, Serial0/1

S* 0.0.0.0/0 [1/0] via 220.0.10.2

Ya con esto todos los routers conocen sus LAN. A modo de ejemplo voy a hacer un tracert (comando para hacer el seguimiento de todos los routers por los que pasa un paquete), desde PC5 que está en la LAN de nogales a PC0, que se encuentra en la LAN del router caborca y cuya IP es 172.16.0.2, en donde se puede ver las interfaces por las que pasa el paquete en los routers.

```
PC>tracert 172.16.0.2

Tracing route to 172.16.0.2 over a maximum of 30 hops:

  1    62 ms     47 ms     62 ms     190.60.0.1
  2    110 ms    94 ms     94 ms     220.0.10.2
  3    125 ms    125 ms    96 ms     200.100.100.1
  4    172 ms    172 ms    188 ms    172.16.0.2

Trace complete.

PC>
```

PROTOCOLO RIPv1

RIP son las siglas de Routing Information Protocol (Protocolo de Información de Enrutamiento). Es un protocolo de puerta de enlace interna o IGP (Internal Gateway Protocol) utilizado por los routers (encaminadores), aunque también pueden actuar en equipos, para intercambiar información acerca de redes IP.

En la actualidad existen tres versiones diferentes de RIP, dos de ellas son:

- RIPv1: No soporta subredes ni direccionamiento CIDR. Tampoco incluye ningún mecanismo de autentificación de los mensajes. No se usa actualmente. Su especificación está recogida en el RFC 1058. Es un protocolo de routing con clase.
- RIPv2: Soporta subredes, CIDR y VLSM. Soporta autenticación utilizando uno de los siguientes mecanismos: no autentificación, autentificación mediante contraseña, autentificación mediante contraseña codificada mediante MD5 (desarrollado por Ronald Rivest). Su especificación está recogida en RFC 1723 y en RFC 2453.

También existe un RIP para IPX, que casualmente lleva el mismo acrónimo, pero no está directamente relacionado con el RIP para redes IP, ad-hoc.

El protocolo RIP, es una implementación directa del vector-distancia en los routers. Utiliza UDP para enviar sus mensajes a través del puerto 520.

VENTAJAS E INCONVENIENTES

Ventajas de RIP

RIP es más fácil de configurar (comparativamente a otros protocolos).

Es un protocolo abierto (admite versiones derivadas aunque no necesariamente compatibles).

Es soportado por la mayoría de los fabricantes.

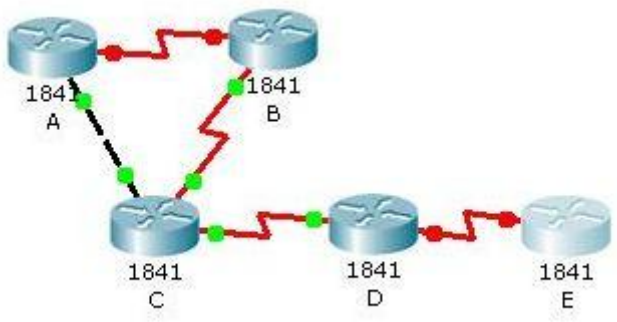
Desventajas de RIP

Su principal desventaja, consiste en que para determinar la mejor métrica, únicamente toma en cuenta el número de saltos, descartando otros criterios (Ancho de Banda, congestión, carga, retardo, fiabilidad, etc.).

RIP tampoco está diseñado para resolver cualquier posible problema de enrutamiento.

PROTOCOLOS DE VECTOR-DISTANCIA

Como ya vimos, un protocolo como **RIP** basado en los saltos anuncia sus vecinos toda su tabla de ruteo, y no conoce de los anchos de banda o estado real de los enlaces en la red, así que, el siguiente escenario nos podría presentar un problema:



Supongamos que hay una red más allá de nuestro router **E** a la que llamaremos **T**, a la cual lógicamente todos nuestros routers llegan por **E**; pero ese enlace falla y **T** es inalcanzable.

Como en **RIP** todos envían sus tablas de ruteo; **A** y **B** anunciarán que conocen la red remota **T** a través de **C**, y **C** anunciará que la conoce a través de **D**, mientras que **D** anunciará que la conoce por **E**, pero **E** anunciará que no está disponible. En ese momento **D** combinará su tabla con el anuncio, pero esperaremos al siguiente anuncio de rutas para que **C** sepa que no hay una ruta hacia **T**; y en el momento en que **C** reciba esa información, **A** hará el anuncio de que conoce una ruta hacia **T** a través de **C**, que le ha anunciado que no es alcanzable, pero que podría utilizar **B** para

alcanzarla, por lo que ahora, todos conocerán a **T** a través de **A** que hizo un anuncio de ruta válida.

Esta situación es un *loop de ruteo*, caímos en una iteración infinita de resolución de una ruta. Para evitar esto, **RIP** tiene un máximo de saltos, en este caso **15 saltos**, después de lo cual declara el destino inalcanzable, es decir, declararemos que **T** está *down* luego de pasar por 15 routers en el proceso de búsqueda. Además, este conteo máximo de saltos nos ayuda a determinar cuánto nos toma declarar una ruta inválida o cuestionable.

Otra posible solución es el algoritmo **Split-Horizon** (horizonte dividido), que reduce la información incorrecta y la carga de tráfico por información de ruteo al aplicar una regla simple: **la información de ruteo no puede regresar por la dirección en que fue recibida**. Es decir, el protocolo diferenciará porque interface aprendió una ruta, y no anunciará la misma ruta por esa interface, así evitaremos que el router A le envíe rutas a B que aprendió de B.

Otra manera de evitar las actualizaciones de rutas inconsistentes y evitar loops de ruteo es envenenar las rutas (**Route Poisoning**). Por ejemplo, cuando T se va down, el router E comienza a anunciar que la ruta hacia T es de 16 saltos (inalcanzable). Así evitamos que D y los demás subsecuentemente, anuncien una ruta inválida, y se asegura mediante el anuncio de una ruta Poison Reverse que D enviará a E, así sabremos que todos los routers del segmento conocen la ruta inalcanzable.

Un **holddown** es un tiempo de espera antes de enviar un anuncio regular de ruteo para una ruta que ha estado cambiando de estado (oscilando o flapping); por ejemplo un enlace serial que pierde conectividad y regresa. Si no hay una manera de estabilizar dicho enlace la red no podrá converger y podría venirse abajo completa. Con el holddown nos aseguramos de que los cambios de estado no sean muy rápidos, dando tiempo para que la ruta afectada regrese o que la red se estabilice antes de volver a usar la ruta afectada. También es una manera de restringir a los routers por un espacio de tiempo los cambios que podrían afectar a rutas que se acaban de retirar. Así evitamos que rutas no operativas se restablezcan en las tablas de otros routers.

REFERENCIAS.

2010, **PROTOCOLOS DE VECTOR DISTANCIA**, investigado en septiembre 2011, de: http://www.ipref.info/2010/04/protocolos-de-vector-distancia.html

ENRUTAMIENTO DINAMICO CON RIP

RIP, es un protocolo de enrutamiento Dinámico de **vector distancia**, esto quiere decir que su métrica para llegar a una red destino se basa en el número de saltos.

Es un protocolo abierto a diferencia de por ejemplo IGRP y EIGRP que son propietarios de Cisco. Es relativamente simple ideal para redes pequeñas, el número de saltos máximo hacia un destino es 15 (cuando hablo de numero de saltos, me refiero a la cantidad de routers, por la que tiene que atravesar el paquete para llegar a destino), ya con 16 la red se declara como inalcanzable.

Existen dos versiones de este protocolo **versión 1 y 2**, la diferencia más importante, es que RIP v1 es lo que se llama un **Protocolo con clase**, lo que significa que cuando publica las tablas de enrutamiento, este no adjunta las máscaras de subred.

En cambio Rip v2 es un **Protocolo sin clase**, que si adjunta la máscara de subred, por lo que permite el uso de VLSM, CIDR, sumarización.

Otra diferencia es que RIP v1 publica sus actualizaciones en forma de **Broadcast**, es decir a todos los equipos de la red, mientras que RIP v2 lo hace en modo de **Multicast**, es decir solo a un grupo de host de una red. Resumiendo las características de las 2 versiones:

RIP Versión 1:
- Protocolo Abierto
- Distancia Administrativa: 120
- Protocolo con clase
- Métrica numero de saltos
- Actualizaciones cada 30 segundos
- Envía las Actualizaciones en forma de Broadcast
- Numero Máximo de Saltos 15
- Red Destino Inalcanzable, se declara como 16 saltos
- No Permite VLSM, CIDR

RIP Versión 2:

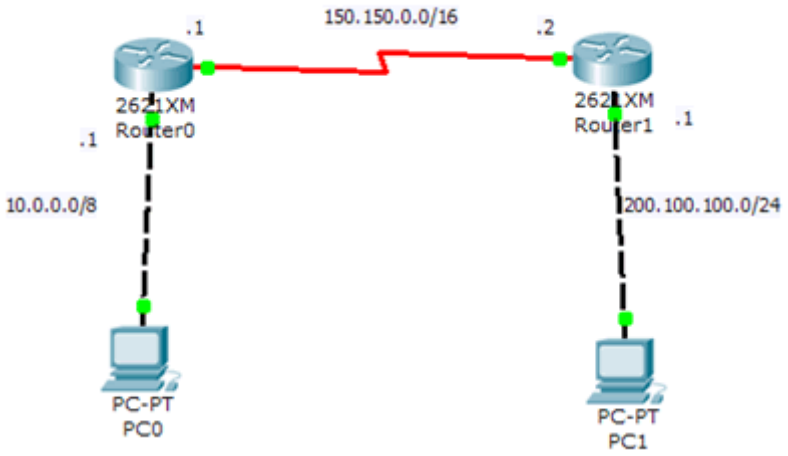
En lo que difiere es lo siguiente, porque lo demás es lo mismo que el versión 1:

- Protocolo sin clase

- Envía las Actualizaciones en forma de Multicast (224.0.0.9)

- Permite VLSM, CIDR.

EJEMPLO PRÁCTICO DE COMO "LEVANTAR" RIP



En Router0

- ✓ router0>enable
- ✓ router0#configure terminal
- ✓ router0(config)#router rip
- ✓ router0(config)#version 2
- ✓ router0(config-router)#network 10.0.0.0
- ✓ router0(config-router)#network 150.150.0.0

En Router1

- ✓ router1>enable
- ✓ router1#configure terminal
- ✓ router1(config)#router rip

- ✓ router1(config)#version 2
- ✓ router1(config-router)#network 200.100.100.0
- ✓ router1(config-router)#network 150.150.0.0

Una explicación de los comandos, se utiliza el comando router en modo configuración global, que es usado para implementar cualquier protocolo de enrutamiento.
Luego se especifica la versión, en este caso yo implemente la versión 2, si no se especifica se configura en la versión 1 que es la por defecto, y por último se utiliza el comando network en sub-modo de protocolo de enrutamiento, en donde se ingresan las redes directamente conectadas al router, y son las que serán publicadas por RIP.

REFERENCIAS.

2008, Blog networking Cisco, Fortaleza Digital, ***PROTOCOLOS DE ENRUTAMIENTO, PARTE 3: ENRUTAMIENTO DINAMICO,*** investigado en agosto 2011, en: http://fortalezadigital08.wordpress.com/2008/09/26/protocolos-de-enrutamiento-parte-2-enrutamiento-estatico/

2001, Bustamante Halime Lucia, ***ENRUTAMIENTO DINAMICO***, investigado en agosto 2011, en: http://www.gfc.edu.co/estudiantes/anuario/2001/sistemas/halime/enruter.html

PROTOCOLO RIPv2

RIP son las siglas de Routing Information Protocol (Protocolo de Información de Enrutamiento). Es un protocolo de puerta de enlace interna o IGP (Internal Gateway Protocol) utilizado por los routers (encaminadores), aunque también pueden actuar en equipos, para intercambiar información acerca de redes IP.

En la actualidad existen tres versiones diferentes de RIP, dos de ellas son:

- RIPv1: No soporta subredes ni direccionamiento CIDR. Tampoco incluye ningún mecanismo de autentificación de los mensajes. No se usa actualmente. Su especificación está recogida en el RFC 1058. Es un protocolo de routing con clase.
- RIPv2: Soporta subredes, CIDR y VLSM. Soporta autenticación utilizando uno de los siguientes mecanismos: no autentificación, autentificación mediante contraseña, autentificación mediante contraseña codificada mediante MD5 (desarrollado por Ronald Rivest). Su especificación está recogida en RFC 1723 y en RFC 2453.

También existe un RIP para IPX, que casualmente lleva el mismo acrónimo, pero no está directamente relacionado con el RIP para redes IP, ad-hoc.

El protocolo RIP, es una implementación directa del vector-distancia en los routers. Utiliza UDP para enviar sus mensajes a través del puerto 520.

Ventajas e Inconvenientes

Ventajas de RIP

RIP es más fácil de configurar (comparativamente a otros protocolos).

Es un protocolo abierto (admite versiones derivadas aunque no necesariamente compatibles).

Es soportado por la mayoría de los fabricantes.

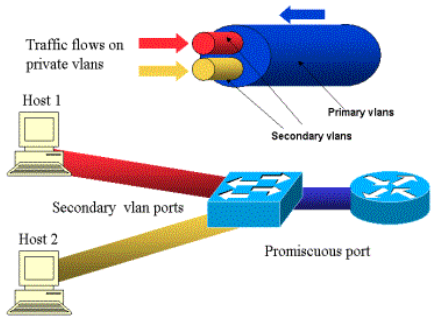
Desventajas de RIP

Su principal desventaja, consiste en que para determinar la mejor métrica, únicamente toma en cuenta el número de saltos, descartando otros criterios (Ancho de Banda, congestión, carga, retardo, fiabilidad, etc.).

RIP tampoco está diseñado para resolver cualquier posible problema de enrutamiento.

VLAN

DEFINICION. Una '''**VLAN**''' (acrónimo de Virtual LAN, Red de Área Local Virtual) es un método de crear redes lógicamente independientes dentro de una misma red física. Varias VLANs pueden coexistir en un único dispositivo de red físico (switch) o en una única red física. Son útiles para reducir el tamaño del dominio de difusión (**BROADCAST**) y ayudan en la administración de la red separando segmentos lógicos de una red de área local (como departamentos de una empresa) que no deberían intercambiar datos usando la red local (aunque podrían hacerlo a través de un ruteador o un switch de capa 3 y 4).

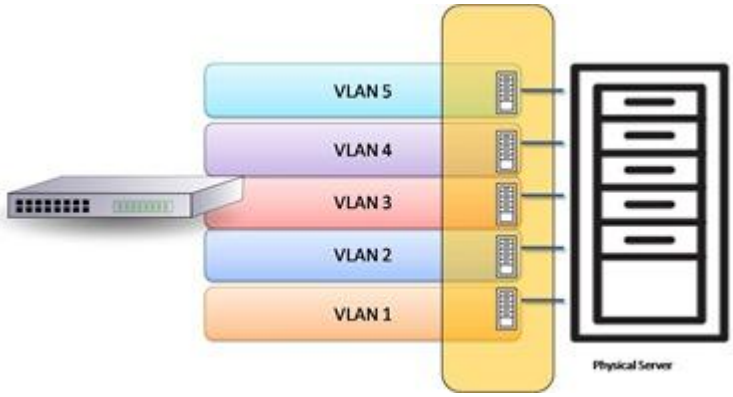


Una VLAN consiste en una red de dispositivos que se comportan como si estuviesen conectados al mismo conmutador, aunque pueden estar en realidad conectados físicamente a diferentes segmentos de una red de área local. Los administradores de red configuran las VLANs mediante software en lugar de hardware, lo que las hace extremadamente flexibles. Una de las mayores ventajas de las VLANs surge cuando se traslada físicamente algún ordenador a otra ubicación: puede permanecer en la misma VLAN sin necesidad de cambiar la configuración IP de la máquina.

PROTOCOLOS Y DISEÑO. El protocolo de etiquetado [**IEEE 802.1Q**] domina el mundo de las VLANs. Antes de su introducción existían varios protocolos propietarios, como el [**ISL**] (''Inter-Switch Link'') de [Cisco], una variante del [**IEEE 802.1Q**], y el **VLT** (''Virtual LAN Trunk'') de [3Com].

Los primeros diseñadores de redes enfrentaron el problema del tamaño de los dominios de colisión (Hubs) esto se logró controlar a través de la introducción de los switch pero a su vez se introdujo el problema del aumento del tamaño de los dominios de difusión (broadcast) y una de las

formas más eficientes para manejarlo fue la introducción de las VLANs. Las VLANs también pueden servir para restringir el acceso a recursos de red con independencia de la topología física de ésta, si bien la robustez de este método es discutible al ser el salto de VLAN ("**VLAN hopping**") un método común de evitar tales medidas de seguridad.



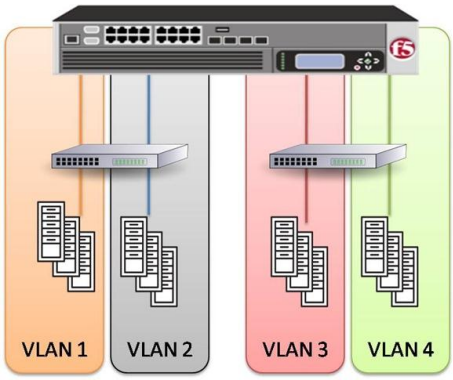
Las VLANs se caracterizan en el nivel 2 (nivel de enlace de datos) del modelo OSI. Sin embargo, los administradores suelen configurar las VLANs como correspondencia directa de una red o subred IP, lo que les da apariencia de funcionar en el nivel 3 (nivel de red).

En el contexto de las VLANs, el término ''**TRUNK**'' (troncal) designa una conexión de red que transporta múltiples VLANs identificadas por etiquetas (o ''**tags**'') insertadas en sus paquetes. Dichos ''trunks'' deben operar entre ''tagged ports'' (puertos etiquetados) de dispositivos con soporte de VLANs, por lo que a menudo son enlaces ''switch'' a ''switch'' o ''switch'' a ''ruteador'' más que enlaces a nodos. (Para mayor confusión, el término ''**TRUNK**'' también se usa para lo que Cisco denomina «**CANALES**»; ó agregado de enlaces). Un ''ruteador'' (''switch'' de nivel 3) funciona como ''columna vertebral'' para el tráfico de red transmitido entre diferentes VLANs.

En los dispositivos Cisco, **VTP** (''**VLAN Trunking Protocol**'') permite definir dominios de VLAN, lo que facilita las tareas administrativas. VTP (Cisco) también permite «podar», lo que significa dirigir tráfico VLAN específico sólo a los ''switches'' que tienen puertos en la VLAN destino.

EJEMPLO DE DEFINICIÓN DE VLAN. Imaginemos que en nuestra empresa tenemos una LAN corporativa con un rango de direcciones IP tipo

172.16.1.XXX/24. Se da el caso de que tenemos asignadas las casi 255 direcciones que como máximo nos permite el mismo y además notamos cierta saturación en la red. Una fácil solución a este problema sería crear unas cuantas VLAN por medio de un "switch de nivel 2 o 3".



Podemos asignar una VLAN a cada departamento de la empresa, así también controlamos que cada uno sea independiente (o no) del resto:

VLAN1: Contabilidad. Direcciones 172.16.2.XXX/24

VLAN2: Compras. Direcciones 172.16.3.XXX/24

VLAN3: Distribución. Direcciones 172.16.4.XXX/24

etc.

De esta forma liberamos direcciones de nuestra red origen 172.16.1.XXX/24 pasándolas a las distintas VLAN que hemos creado. Gracias al switch de nivel 3 podremos gestionar la visibilidad entre las distintas VLAN y notaremos una mejora en el rendimiento de la red ya que las difusiones o broadcast de cada VLAN sólo llegarán a los equipos conectados a la misma.

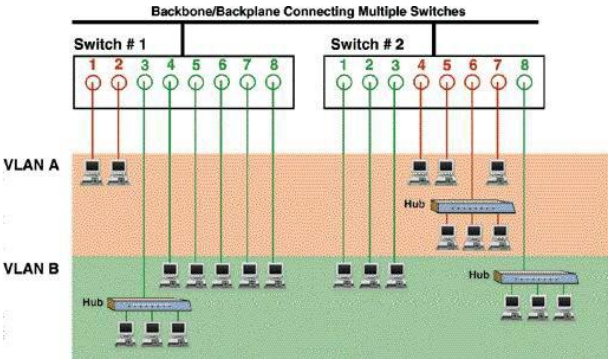
GESTIÓN DE LA PERTENENCIA A UNA VLAN. Las dos aproximaciones más habituales para la asignación de miembros de una VLAN son las siguientes: VLAN estáticas y VLAN dinámicas.

Las VLAN estáticas también se denominan VLAN basadas en el puerto. Las asignaciones en una VLAN estática se crean mediante la asignación de los puertos de un switch o conmutador a dicha VLAN.

Cuando un dispositivo entra en la red, automáticamente asume su pertenencia a la VLAN a la que ha sido asignado el puerto. Si el usuario cambia de puerto de entrada y necesita acceder a la misma VLAN, el

administrador de la red debe cambiar manualmente la asignación a la VLAN del nuevo puerto de conexión en el switch.

En las VLAN dinámicas, la asignación se realiza mediante paquetes de software tales como el CiscoWorks 2000. Con el VMPS (acrónimo en inglés de VLAN Policy Server o Servidor de Directivas de la VLAN), el administrador de la red puede asignar los puertos que pertenecen a una



VLAN de manera automática basándose en información tal como la dirección MAC del dispositivo que se conecta al puerto o el nombre de usuario utilizado para acceder al dispositivo. En este procedimiento, el dispositivo que accede a la red, hace una consulta a la base de datos de miembros de la VLAN. Se puede consultar el software FreeNAC para ver un ejemplo de implementación de un servidor VMPS.

VLAN BASADAS EN EL PUERTO DE CONEXIÓN. Con las VLAN con pertenencia basada en el puerto de conexión del switch, el puerto asignado a la VLAN es independiente del usuario o dispositivo conectado en el puerto. Esto significa que todos los usuarios que se conectan al puerto serán miembros de la misma VLAN. Habitualmente es el administrador de la red el que realiza las asignaciones a la VLAN. Después de que un puerto ha sido asignado a una VLAN, a través de ese puerto no se puede enviar ni recibir datos desde dispositivos incluidos en otra VLAN sin la intervención de algún dispositivo de capa 3.

El dispositivo que se conecta a un puerto, posiblemente no tenga conocimiento de la existencia de la VLAN a la que pertenece dicho puerto. El dispositivo simplemente sabe que es miembro de una sub-red y que puede ser capaz de hablar con otros miembros de la sub-red simplemente enviando información al segmento cableado. El switch es responsable de identificar que la información viene de una VLAN determinada y de asegurarse de que esa información llega a todos los demás miembros de la VLAN. El switch también se asegura de que el resto de puertos que no están en dicha VLAN no reciben dicha información.

Este planteamiento es sencillo, rápido y fácil de administrar, dado que no hay complejas tablas en las que mirar para configurar la segmentación de la VLAN. Si la asociación de puerto a VLAN se hace con un ASIC (acrónimo en inglés de Application-Specific Integrated Circuit o Circuito integrado para una aplicación específica), el rendimiento es muy bueno. Un ASIC permite el mapeo de puerto a VLAN sea hecho a nivel hardware.

REFERENCIAS.

2011, Creative Commons, ***VLAN****, investigado en: http://es.wikipedia.org/wiki/VLAN, en septiembre 2011.*

VLAN

Una VLAN (acrónimo de Virtual LAN) es una subred IP separada de manera lógica, las VLAN permiten que redes IP y subredes múltiples existan en la misma red conmutada, son útiles para reducir el tamaño del broadcast y ayudan en la administración de la red separando segmentos lógicos de una red de área local (como departamentos para una empresa, oficina, universidades, etc.) que no deberían intercambiar datos usando la red local.[5]

Una Red de Área Local Virtual (VLAN) puede definirse como una serie de dispositivos conectados en red que a pesar de estar conectados en diferentes equipos de interconexión (hubs o switches), zonas geográficas distantes, diferentes pisos de un edificio e, incluso, distintos edificios, pertenecen a una misma Red de Área Local.

Con los switchs , el rendimiento de la red mejora en los siguientes aspectos:

- Aísla los "dominios de colisión" por cada uno de los puertos.
- Dedica el ancho de banda a cada uno de los puertos y, por lo tanto, a cada computadora.

- Aísla los "dominios de broadcast", en lugar de uno solo, se puede configurar el switch para que existan más "dominios".
- Proporciona seguridad, ya que si se quiere conectar a otro puerto del switch que no sea el suyo, no va a poder realizarlo, debido a que se configuraron cierta cantidad de puertos para cada VLAN.
- Controla más la administración de las direcciones IP. Por cada VLAN se recomienda asignar un bloque de IPs, independiente uno de otro, así ya no se podrá configurar por parte del usuario cualquier dirección IP en su máquina y se evitará la repetición de direcciones IP en la LAN.

En el estándar 802.1Q se define que para llevar a cabo la comunicación dentro en una VLAN se requerirá de un dispositivo dentro de la LAN, capaz de entender los formatos de los paquetes con que están formadas las VLANs. Este dispositivo es un equipo de capa 3, mejor conocido como enrutador o router, que tendrá que ser capaz de entender los formatos de las VLANs para recibir y dirigir el tráfico hacia la VLAN correspondiente.

Las VLANs se caracterizan en el nivel 2 (nivel de enlace de datos) del modelo OSI. Sin embargo, los administradores suelen configurar las VLANs como correspondencia directa de una red o subred IP, lo que les da apariencia de funcionar en el nivel 3 (nivel de red).

En el contexto de las VLANs, el término ''**TRUNK**'' (troncal) designa una conexión de red que transporta múltiples VLANs identificadas por etiquetas (o ''**tags**'') insertadas en sus paquetes. Dichos ''trunks'' deben operar entre ''tagged ports'' (puertos etiquetados) de dispositivos con soporte de VLANs, por lo que a menudo son enlaces ''switch'' a ''switch'' o ''switch'' a ''ruteador'' más que enlaces a nodos. (Para mayor confusión, el término ''**TRUNK**'' también se usa para lo que Cisco denomina «**CANALES**»; ó agregado de enlaces). Un ''ruteador'' (''switch'' de nivel 3) funciona como ''columna vertebral'' para el tráfico de red transmitido entre diferentes VLANs.

En los dispositivos Cisco, **VTP** (''**VLAN Trunking Protocol**'') permite definir dominios de VLAN, lo que facilita las tareas administrativas. VTP (Cisco)

también permite «podar», lo que significa dirigir tráfico VLAN específico sólo a los ''switches'' que tienen puertos en la VLAN destino.

Ejemplo de definición de VLAN. Imaginemos que en nuestra empresa tenemos una LAN corporativa con un rango de direcciones IP tipo 172.16.1.XXX/24. Se da el caso de que tenemos asignadas las casi 255 direcciones que como máximo nos permite el mismo y además notamos cierta saturación en la red. Una fácil solución a este problema sería crear unas cuantas VLAN por medio de un "switch de nivel 2 o 3".

Podemos asignar una VLAN a cada departamento de la empresa, así también controlamos que cada uno sea independiente (o no) del resto:

VLAN1: Contabilidad. Direcciones 172.16.2.XXX/24

VLAN2: Compras. Direcciones 172.16.3.XXX/24

VLAN3: Distribución. Direcciones 172.16.4.XXX/24

etc.

De esta forma liberamos direcciones de nuestra red origen 172.16.1.XXX/24 pasándolas a las distintas VLAN que hemos creado. Gracias al switch de nivel 3 podremos gestionar la visibilidad entre las distintas VLAN y notaremos una mejora en el rendimiento de la red ya que las difusiones o broadcast de cada VLAN sólo llegarán a los equipos conectados a la misma.

EL BENEFICIO DE IMPLEMENTAR UNA VLAN

¿Por qué están los fabricantes tan interesados en las VLANs? ¿Acaso piensan que son la solución a los problemas que tienen los administradores respecto a los cambios, movimientos, emisión y actuación de la red?

Reducción del Coste de Movimientos y Cambios. La principal excusa para implementar una VLAN es la reducción en el coste de los cambios y movimientos de usuarios. Desde que estos costes son bastante sustanciales, este argumento es suficientemente obligatorio para la implementación de una VLAN.

Muchos fabricantes están prometiendo que la implementación de una VLAN resultará más conveniente a la hora de habilitar la administración de redes dinámicas, y que esto supondrá bastante ahorro. Esta promesa se puede aplicar con buenos resultados a redes IP, ya que, normalmente, cuando un usario se mueve a una diferente subred, las direcciones IP han de ser actualizadas manualmente en la estación de trabajo. Este proceso consume gran cantidad de tiempo que podría ser aprovechado para otras tareas, tales como producir nuevos servicios de red. Una VLAN elimina ese hecho, porque los miembros de una red virtual no están atados a una localización física en la red, permitiendo que las estaciones cambiadas de sitio conserven su dirección IP original.

Sin embargo, cualquier implementación de VLAN no reduce este coste. Una VLAN añade una nueva capa de conexión virtual que ha de ser administrada al mismo tiempo que la conexión física. Esto no quiere decir que no se puedan reducir los costes hablados anteriormente. Sólo que no hay que precipitarse a la hora de implementar una VLAN y es mejor estar bien seguro de que la solución no genera más trabajo de administración de red que el que se pueda ahorrar.

Grupos de Trabajo Virtuales. Uno de los objetivos más ambiciosos de una red virtual es el establecimiento del modelo de grupos de trabajo virtuales. El concepto es que, con una completa implementación de una VLAN a través de todo el entorno de red del campus, miembros del mismo departamento o sección puedan aparentar el compartir la misma red local, sin que la mayoría del tráfico de la red esté en el mismo dominio de *broadcast* de la VLAN. Alguien que se mueva a una nueva localización física pero que permanezca en el mismo departamento se podría mover sin tener que reconfigurar la estación de trabajo.

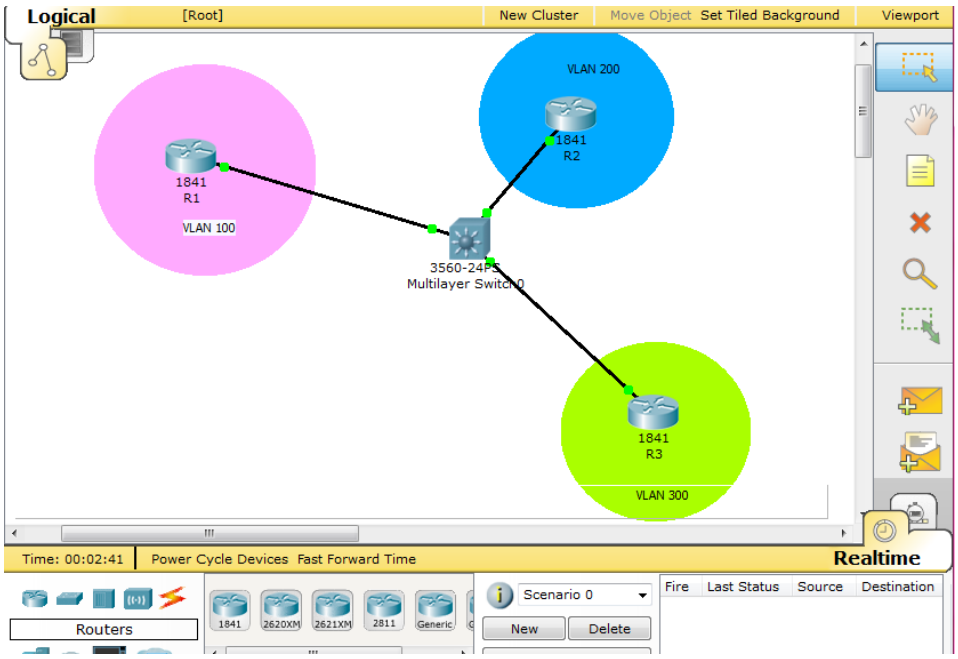
Esto ofrece un entorno más dinámicamente organizado, permitiendo la tendencia hacia equipos con funciones cruzadas. La lógica del modelo virtual por grupos de trabajo va la siguiente forma: los equipos pueden estar conectados virtualmente a la misma LAN sin necesidad de mover físicamente a las personas para minimizar el tráfico a través de una red troncal colapsada. Además, estos grupos serán dinámicos: un equipo destinado a un proyecto puede ser configurado mientras dure ese proyecto, y ser eliminado cuando se complete, permitiendo a los usuarios retornar a sus mismas localizaciones físicas.

Seguridad. El único tráfico de información en un segmento de un sólo usuario será de la VLAN de ese usuario, por lo que sería imposible "escuchar" la información si no nos es permitida, incluso poniendo el adaptador de la red en modo promiscuo, porque ese tráfico de información no pasa físicamente por ese segmento.

http://www.lcc.uma.es/~eat/services/rvirtual/rvirtual.html#link3

- **Práctica**

Configurar una VLAN usando el simulador Packet Tracer



- **Instrucciones**

1.- Utilizar un switch capa 3 para conectar 3 routers, los cuales representarán las 3 VLAN que vamos a configurar.

2.- Configurar cada router por separado, asignándole sus respectivas direcciones IP, máscaras de subred y Gateway.

3.- Configurar en el switch capa 3 VLANs y enlaces troncales

TRUNK

***¿QUÉ ES?** Es una función que permite conectar Switch, Routers o servidores de manera que se conectan por medio de un cable full dúplex para enviar y recibir paquetes simultáneamente y que nos permiten una comunicación con un ancho de banda más grande. Esto nos permite reducir los cuellos de botella en la conexión de varios segmentos de la red.

Esta modalidad de conexión se puede utilizar cuando se realizan redes locales virtuales que nos permiten comunicar una red virtual con otra.

***¿PARA QUÉ SIRVE?** Esta modalidad permite comunicar 2 redes virtuales locales por el mismo medio físico, y al mismo tiempo nos permite evitar el utilizar un cable para cada red virtual.

Por ejemplo:

Una red virtual conectada a un Switch de un lado, y otra red virtual conectada a otro lado igualmente a un Switch. Para conectar estas redes entre sí, ocuparíamos 2 cables, uno para cada red virtual.

En cambio por medio del trunk se utiliza un protocolo (el 802.1q) que permite que haya comunicación entre los equipos y que se entiendan entre ellos.

***¿CÓMO FUNCIONA?** Para entender el funcionamiento del modo trunk, se necesita saber cómo funcionan las redes locales virtuales (VLANS). Estas son redes creadas dentro de una red física, que son independientes de la misma, estas son útiles para reducir para reducir el tamaño del dominio de la red y a su vez ayudan a la administración de la red.

Estas redes virtuales se conectan en un mismo medio físico, es decir, si tomamos por ejemplo dos redes virtuales divididas bajo una red para conectarse cada una mediante un Switch por medio de un Router necesitaríamos poner el Router en modo trunk, que permitiría comunicar

estos segmentos de red bajo el protocolo 802.1q y haría que las redes se entendieran la una con la otra con el mismo lenguaje y reducir el tamaño de la red y aumentar el ancho de banda, de manera que trabajan mas rápido que si estuvieran conectadas con cable.

*** ¿CÓMO SE IMPLEMENTA?** Para implementar el modo trunk ocupamos varias cosas

1. **configurar los equipos adecuadamente; llenar los campos de la IP el Gateway etc.**
2. **Crear las redes virtuales necesarias;**
    a. Switch(config)# **vlan 10**
    b. Switch(config-vlan)# **exit**
    c. Switch(config)# **vlan 20**
    d. Switch(config-vlan)# **exit**
3. **Configurar los puertos por su tipo**
    a. Switch(config)# **interface range fa0/5 - 14**
    b. Switch(config-if-range)# **switchport access vlan 10**
    c. Switch(config-if-range)# **exit**
    d. Switch(config)# **interface range fa0/15 - 26**
    e. Switch(config-if-range)# **switchport access vlan 20**
4. **Configurar el Puerto que se va a utilizar como trunk**
    a. Switch(config)# **interface port-channel 1**
    b. Switch(config-if)# **switchport mode trunk**

TABLA DE DIRECCIONAMIENTO				
DISPOSITIVO	**INTERFAZ**	**DIRECCION**	**MASCARA DE**	**GATEWAY**

		IP	SUBRED	
R1	VLAN 100	172.17.99.11	255.255.255.0	172.17.99.1
R2	VLAN 200	172.17.99.12	255.255.255.0	172.17.99.1
R3	VLAN 300	172.17.99.13	255.255.255.0	172.17.99.1

COMANDOS

En router

interface serial 0/0

ip address "ip address + mascara de red"

interface FastEthernet 1/**0.1**←Interface virtual

encapsulation dot1q + "Nombre VLAN"

ip address "IP address y mascara de subred que manejara la interface"

En el switch capa 3 (Ejemplo)

Switch>enable

Switch#configure terminal

Switch(config)#vlan 100

Switch(config-vlan)#name LAN100

Switch(config-vlan)#exit

Switch(config)#vlan 200

Switch(config-vlan)#name LAN200

Switch(config-vlan)#exit

Switch(config)#interface fastEthernet 0/1

```
Switch(config-if)#switchport mode trunk
Switch(config-if)#switchport trunk allowed vlan 100,200
Switch(config-if)#exit
Switch(config)#interface vlan 100
Switch(config-if)#ip address 192.168.10.1 255.255.255.0
Switch(config-if)#exit
Switch(config)#interface vlan 200
Switch(config-if)#ip address 192.168.20.1 255.255.255.0
Switch(config-if)#exit
```

SUBINTERFASES

¿QUÉ ES? Es una división de una interface física en varias interfaces lógicas comúnmente esto se hace para reducir el tráfico de la red.

Por ejemplo suponiendo que tenemos 3 VLANS en una misma red y estas a su vez en un Switch que está conectado a un Router, si trabajamos con un cable de Ethernet, es una interface Fast Ethernet por default 0/0 al dividirla en sub interfaces la fragmentamos en 0/0.1, esto permitirá que la interfaz 0/0 se pueda dirigir a varias redes virtuales por un mismo conducto físico.

¿PARA QUÉ SIRVE? Nos Permite configurar las redes de un multilayer capa 3 para hacer divisiones de las redes que subdividimos. Por ejemplo si tenemos conectados varios Routers a un capa 3 y de la misma manera tenemos varias redes virtuales conectadas en el, el multilayer nos hace divisiones y nos permite dirigirnos a cada una de ellas de manera que se esté utilizando una misma interface pero dividida en sectores. Y esto nos disminuye el tráfico en la red, además de permitirnos trabajar cuando solo se está trabajando con una sola interface, por ejemplo fast Ethernet 0/0 se puede dividir en fast Ethernet 0/0.1 y fast Ethernet 0/0.2, se usaría una misma interface, pero se dividiría en 2 sub interfaces.

¿CÓMO FUNCIONA? Para hacer las divisiones de las interfaces se utilizan simples divisiones de las interfaces se utiliza la líneas de comandos del CLI, una simple división realiza la división de la interface para poder usarla como interfaces divididas que dan las ventajas de trabajar con redes virtuales divididas.

¿CÓMO SE IMPLEMENTA?

	COMANDO	**PROPOSITO**
Step 1	Router(config)# **interface** *type number*	Specifies the interface and enters interface configuration mode.

Step 2	Router(config-if)# **dialer in-band** [**no-parity** | **odd-parity**]	Enables DDR. Specifies parity, if needed, on synchronous or asynchronous serial interfaces.
Step 3	Router(config-if)# **encapsulation ppp**	Enables PPP encapsulation.
Step 4	Router(config-if)# **ppp authentication**{**chap** | **pap**}	Enables CHAP or PAP authentication.
Step 5	Router(config-if)# **dialer map** *protocol next-hop-address* **name** *hostname dial-string*	Maps the next hop address to the host name and phone number.
Step 6	Router(config-if)# **ppp callback request**	Enables the interface to request PPP callback for this callback map class.
Step 7	Router(config-if)# **dialer hold-queue***packets* **timeout** *seconds*	(Optional) Configures a dialer hold queue to store packets for this callback map class.

	COMANDO	**PROPOSITO**
Step 1	Router(config)# **ip routing**	Enables IPv4 routing. (Required only if IPv4 routing is disabled.)
Step 2	Router(config)# **router** *ip_routing_protocol*	Specifies an IPv4 routing protocol.
Step 3	Router(config-	Configures the IPv4

	router)# *ip_routing_protocol_commands*	routing protocol.
Step 4	Router(config-router)# **exit**	Exists IPv4 routing protocol configuration mode.
Step 5	Router(config)# **interface** {**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}	Selects an interface to configure.
Step 6	Router(config-if)# **ip address** *ip_address subnet_mask*	Configures the IPv4 address and IPv4 subnet.
Step 7	Router(config-if)# **no shutdown**	Enables the interface.
Step 8	Router(config-if)# **end**	Exits configuration mode.
Step 9	Router# **show interfaces** [{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}] Router# **show ip interfaces** [{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel** *port_channel_number*}] Router# **show running-config interfaces** [{**vlan** *vlan_ID*} | {*type*[1] *slot/port*} | {**port-channel***port_channel_number*}]	Verifies the configuration.

PPP

¿QUÉ ES? Es un protocolo de punto a punto que permite establecer enlaces entre dos computadoras por medio de TCP/IP. Este protocolo permite el transporte de datos y además facilita dos funciones

La autentificación para seguridad del transporte de datos, esto se hace por medio de una contraseña o clave de acceso

Asignación dinámica de la IP esto les deja un número limitado de IP y control de estas, esto permite asignarle una dirección particular a cada cliente para que se conecten con el proveedor directamente.

¿PARA QUÉ SIRVE? Permite principalmente comunicación entre computadoras, y conexión a internet desde un proveedor de acceso, a través de modem telefónico, aunque también es utilizado a través de conexiones de banda ancha.

Su principal función es el transporte de datos, pero también asegura la conexión con autentificación y permite la asignación dinámica de la IP que da más control sobre las direcciones IP que se otorgan a clientes y se permiten manipular las entradas de estos a los proveedores.

¿CÓMO FUNCIONA? Consta de 5 fases:

Primero se establece una conexión, se utiliza un protocolo de comunicación (LCP) este permite un método de autenticación que se utiliza en la conexión

Después llega la autenticación, no es una fase obligatoria pero si se hace se utilizan protocolos de protocolos especificados en la autentificación

Luego se establece una configuración de red, en esta fase se negocian los parámetros de protocolo de red que se estén usando, y se manejan los parámetros de los protocolo de cliente y servidor

Después se hace la trasmisión, esta es la fase en la que se manda y recibe la información de red, esto se hace sin cifrado de datos esto es por ppp

Y por último se hace la terminación que es el cierre de conexión

¿CÓMO SE IMPLEMENTA? Configuring a Router as a Callback Client. To configure a router interface as a callback client, use the following commands beginning in global configuration mode:

	COMANDO	PROPOSITO
Step 1	Router(config)# **interface** *type number*	Specifies the interface and enters interface configuration mode.
Step 2	Router(config-if)# **dialer in-band** [**no-parity** | **odd-parity**]	Enables DDR. Specifies parity, if needed, on synchronous or asynchronous serial interfaces.
Step 3	Router(config-if)# **encapsulation ppp**	Enables PPP encapsulation.
Step 4	Router(config-if)# **ppp authentication**{**chap** | **pap**}	Enables CHAP or PAP authentication.
Step 5	Router(config-if)# **dialer map** *protocol next-hop-address* **name** *hostname dial-string*	Maps the next hop address to the host name and phone number.
Step 6	Router(config-if)# **ppp callback request**	Enables the interface to request PPP callback for this callback map class.
Step 7	Router(config-if)# **dialer hold-queue***packets* **timeout** *seconds*	(Optional) Configures a dialer hold queue to store packets for this callback map class.

VTP

¿QUÉ ES? Es un protocolo que nos permite configurar y administrar las VLANS de los equipos cisco. Significa protocolo de trunking de redes virtuales.

Este protocolo permite operar en tres diferentes modos, en modo cliente, en modo servidor, y en modo transparente

1. El modo servidor permite hacer modificaciones directas en el servidor, permite crear eliminar y modificar las VLANS
2. El modo cliente no deja hacer modificaciones entre las VLANS, no se pueden ni crear, ni modificar VLANS.
3. El modo transparente no deja modificar directamente las VLANS pero deja crear y modificar VLANS en el CLI y deja moverlas dentro el mismo equipo pero sin interactuar entre los demás equipos.

¿PARA QUÉ SIRVE? Permite administrar las redes virtuales de manera completa y más segura este protocolo, aunque se puede trabajar sin autentificación, es más recomendable manejarlo con contraseña para darle seguridad a la red

¿CÓMO FUNCIONA? Las configuraciones VTP en una red son controladas por un número de revisión. Si el número de revisión de una actualización recibida por un Switch en modo cliente o servidor es más alto que la revisión anterior, entonces se aplicará la nueva configuración. De lo contrario se ignoran los cambios recibidos. Cuando se añaden nuevos dispositivos a un dominio VTP, se debe resetear los números de revisión de todo el dominio VTP para evitar conflictos. Se recomienda tener mucho cuidado al usar VTP cuando haya cambios de topología ya sean lógicos o físicos.

¿CÓMO SE IMPLEMENTA?

1. Se configurarn los equipos adecuadamente

2. Se crean las vlans necesarias
    a. Switch(config)# **vlan 10**
    b. Switch(config-vlan)# **exit**
    c. Switch(config)# **vlan 20**
    d. Switch(config-vlan)# **exit**

3. Se define el protocolo a usar

*Modo servidor:

a. Switch(config)# **vtp mode server**
b. Setting device to VTP SERVER mode
c. Switch(config)# **vtp domain nombre1**
d. Changing VTP domain name from VitalCom to cisco

1. Modo Cliente:
    a. Switch(config)# **vtp mode client**
    b. Setting device to VTP SERVER mode
    c. Switch(config)# **vtp domain nombre2**
    d. Changing VTP domain name from VitalCom to cisco
2. Modo Transparente:
    a. Switch(config)# **vtp mode transparent**
    b. Setting device to VTP SERVER mode
    c. Switch(config)# **vtp domain nombre3**
    d. Changing VTP domain name from VitalCom to cisco

STP EN SWITCHES CISCO CATALYST[6]

Un protocolo ineludible en la implementación de redes conmutadas es STP (Spanning Tree Protocol). Es un protocolo estándar (IEEE 802.1d), desarrollado inicialmente para administrar enlaces redundantes en redes conmutadas utilizando bridges.

El protocolo inicial presenta 2 limitaciones importantes:

- La administración de redundancia se hace definiendo una topología activa y bloqueando los enlaces redundantes. La primer consecuencia de este proceso es la imposibilidad de aprovechar completamente el ancho de banda instalado realizando balanceo de tráfico, como ocurre con las rutas redundantes en el enrutamiento IP.
- Por otra parte, en el caso de un fallo, un puerto STP demora 50 segundos en pasar del estado de blocking al de forwarding. Estos tiempos de convergencia son muy altos para las redes actuales.

Estas características o limitaciones del protocolo han sido sucesivamente mejoradas en sucesivas revisiones; algunas de ellas estándar, otras, propietarias de Cisco

Port Fast

- ✓ Feature propietario de Cisco.
- ✓ Permite acelerar los tiempos de habilitación de un puerto al momento de conectar una terminal a una boca de un switch.
- ✓ Sólo se habilita en puertos de acceso.
- ✓ Si la interfaz recibe una BPDU de STP, pasa inmediatamente al estado de blocking, y a operar en el modo normal de STP.
- ✓ Switch(config)#interface fastEthernet 0/1
- ✓ Switch(config-if)#spanning-tree portfast
- ✓ Switch(config)#spanning-tree portfast default

PVSTP

- ✓ Per VLAN Spanning Tree Protocol
- ✓ Implementación propietaria de Cisco.
- ✓ Genera una instancia de STP para cada VLAN.
- ✓ Utiliza enlaces troncales ISL.
- ✓ Permite distribuir el tráfico de las diferentes VLANs generando diferentes topologías activas.
- ✓ Switch(config)#spanning-tree mode pvst

PVSTP+

- ✓ Per VLAN Spanning Tree Protocol Plus.
- ✓ Implementación propietaria de Cisco.
- ✓ Semejante a PVSTP, pero para operar sobre enlaces troncales 802.1Q.

RSTP (IEEE 802.1w)

- ✓ Implementación estándar.
- ✓ Mejora notablemente los tiempos de convergencia del protocolo.
- ✓ Incluye una funcionalidad semejante a port fast, denominada port edge.
- ✓ Mantiene compatibilidad con STP.

RPVSTP+

- ✓ Implementación de RSTP propietaria de Cisco.
- ✓ Genera una instancia de RSTP para cada VLAN creada.
- ✓ Utiliza enlaces troncales IEEE 802.1Q.
- ✓ Switch(config)#spanning-tree mode rapid-pvst

MSTP (IEEE 802.1s)

- ✓ Implementación estándar de RSTP.
- ✓ Genera hasta 16 instancias de RSTP.
- ✓ Se deben asociar las VLANs a las instancias creadas.
- ✓ Utiliza enlaces troncales IEEE 802.1Q.
- ✓ Es menos exigente en procesamiento y reduce el número de actualizaciones que se envían.

Los switches Catalyst 2960 implementan por defecto PVSTP+ y permiten configurar port fast, RPVSTP+ y MSTP.

Un punto importante es comprender como STP elige a un Switch como RAIZ del árbol STP, y los diferentes estados por los cuales transitara el proceso de bloqueo y habilitación de los puertos en entornos LAN con STP, y sus variantes, habilitado dado que a la hora de detectar fallos ocurre que al desconocer esos conceptos se presuponen condiciones erróneas y se demora en la resolución de los problemas.

STP no necesariamente necesita un switch potente, pero si necesita un switch ubicado estratégicamente, dado que ese switch tendrá la "visión" completa del proceso STP. (Pregunta recurrente en los foros)

STP es muy útil para conexiones desde puertos del Switch hacia estaciones de trabajo u otros dispositivos, pero esta contra-indicado en entornos con switches en cascada o "back-to-back" dado que puede generar bucles indefinidos entre los puertos up-link vinculantes.

Comandos útiles:

* Show spantree vlan_id — muestra el estado actual del árbol STP según la ID de VLAN (desde la perspectiva del switch en el cual se implemento este comando)

* Show spantree summary - brinda un resumen de los puertos STP conectados, por VLAN.

* Show spantree statistics - brinda información estadística.

* Show spantree backbonefast - permite conocer si la funcion BackboneFast Convergence, esta habilitada.

* Show spantree blockedports— informa sobre los puertos que están en estado de bloqueo.

* Show spantree portvlancost— muestra el costo de ruta de las VLAN's sobre un puerto. esto es útil dado que STP elige su puerto root de STP por el costo del mismo salvo que se haya modificado manualmente la selección de la prioridad.

* Show spantree uplinkfast— muestra los parámetros UplinkFast.

IGRP[7]

IGRP es un protocolo de routing interno utilizado en TCP/IP y OSI. La versión original de IP fue diseñada y desarrollada con éxito en 1986. Se considera un IGP (Interior Gateway Protocol) pero también ha sido utilizado como un protocolo de routing externo para el routing inter-domain. IGRP utiliza el algoritmo del vector distancia. El concepto es que cada router no necesita conocer todas las rutas/enlaces de la red entera. Cada router, informa acerca de los destinos y su distancia correspondiente. Cada router escuchando información, ajusta las distancias y las propaga a los routers vecinos.

La información sobre la distancia en IGRP está representada como una combinación de ancho de banda disponible, retardo, carga y fiabilidad del enlace. Esto permite conseguir rutas óptimas.

Hay algunas áreas en las que la descripción de este protocolo puede esperarse que sea diferente de la implementación de Cisco. Estas áreas son:

- ✓ Cisco no implementa todavía múltiples tipos de servicio, no testea lo que depende de la cuenta de los saltos. De cualquier modo hacen mantenimiento y propagan la información necesaria.
- ✓ Cisco tiene un número de controles administrativos, permitiendo filtros y modificaciones de varias clases en la información de routing.
- ✓ Cisco proporciona varios caminos para definir las rutas por defecto

OBJETIVOS DE IGRP. IGRP es un protocolo que asigna un número de routers para coordinar su routing. Sus metas son:

- ✓ routing estable incluso en redes muy grandes y complejas. No deben producirse bucles, incluso si son transitorios.
- ✓ rápida respuesta a cambios en la topología de la red
- ✓ pequeño overhead, IGRP no usa más ancho de banda que lo que necesita para su tarea.
- ✓ reparte el tráfico entre rutas paralelas diferentes cuando éstas son en términos generales igual de buenas.
- ✓ toma en cuenta la tasa de errores y el nivel de tráfico en diferentes caminos
- ✓ la capacidad de manejar múltiples "tipos de servicio" con un conjunto simple de información.

La actual implementación de IGRP maneja routing para TCP/IP. De todos modos, el diseño básico esta propuesto para ser capaz de manejar una variedad de protocolos.

Durante los últimos años, el routing se ha convertido de repente en un problema más difícil al que solíamos. Hace pocos años, protocolos como RIP eran suficientes, pero el crecimiento de la Internet, y la descentralización del control de su estructura, ha resultado en un sistema de redes que está muy lejos de nuestra capacidad de manejarlo. IGRP es una herramienta propuesta para atacar este problema.

Ninguna herramienta va a resolver todos los problemas de routing. Generalmente el problema del routing se rompe en varias piezas. Protocolos como IGRP son llamados "protocolos de routing interno" (IGPs). Estan propuestos para su uso en un conjunto simple de redes, bajo una dirección simple o una estrecha coordinación de los directores. Estos conjuntos de redes son conectados por "protocolos de routing externo" (EGPs). Un IGP está diseñado para mantener gran cantidad de detalles sobre la topología de la red. Su prioridad es fija en producir rutas óptimas y respondiendo rápidamente a los cambios. Un EGP está destinado a proteger un sistema

de redes contra errores o una intencionada tergiversación por otros sistemas. Su prioridad está en los controles de estabilidad y administrativos.

IGRP tiene algunas similitudes con viejos protocolos con Xerox's Routing Information Protocol, Berkeley's RIP, and Dave Mill's Hello. Difiere con estos protocolos en que está diseñado para redes más grandes y complejas.

Como estos viejos protocolos, IGRP es un protocolo basado en el algoritmo del vector distancia. Los routers intercambian información de routing solo con sus routers vecinos. Esta información de routing contiene un resumen de información sobre el resto de la red. Cada router solo necesita resolver parte del problema, y solo tiene que recibir una porción de los datos totales.

La principal alternativa es una clase de algoritmos referidos a SPF (shortest-path first). Que están basados en la técnica de "flooding"(inundación), donde todo router debe mantener información del estado de toda interface en todos los otros routers. Cada router independientemente resuelve el problema desde su pinto de vista usando información de toda la red. En algunas circunstancias SPF puede ser capaz de responder a cambios más rápidamente. Para prevenir los bucles, IGRP tiene que ignorar nuevos datos durante unos pocos minutos después de fijar los cambios. Porque SPF tiene información directamente de cada uno de los routers, es posible evitar estos bucles en el routing. Puede actuar con la nueva información inmediatamente. De todos modos, SPF tiene más información que IGRP, tanto en las estructuras de datos internas y como en los mensajes que intercambian los routers. Las implementaciones de SPF tienen más overhead que las implementaciones de IGRP, en otras cosas son iguales.

EL PROBLEMA DEL ROUTING. IGRP esta diseñado para usarse en routers que conectan distintas redes. Asumimos que loas redes usan la tecnología basada en paquetes. De hecho los routers actúan como conmutadores de paquetes. Cuando un equipo conectado a una red quiere enviar un paquete a otro equipo en una red diferente, dirige el paquete al router. Si el destino se encuentra en una de las redes conectadas al router, el router mandará el paquete al destino. Sino lo enviará a otro router que se

encuentre cerca del destino. Los routers utilizan las tablas de rutas para ayudarse a decidir qué hacer con el paquete.

La principal propuesta de IGRP es permitir a los routers construir y mantener las tablas de rutas.

RESUMEN DE IGRP. IGRP es un protocolo que permite a los routers construir las tablas de routing a partir del intercambio de información con otros routers. Un router comienza con entradas en sus tablas para todas las redes que están directamente conectadas a él. En el caso más simple, el router encontrará una ruta que representa la mejor para llegar a cada red. Un camino se caracteriza por el próximo router al que deben ser enviados los paquetes, la interface de red que debe utilizarse e información de la métrica. La métrica es un conjunto de números que determinan cuánto de buena es una ruta. Esto permite al router comparar rutas y elegir la mejor. Hay a menudo casos donde hace sentir que se reparte el tráfico entre 2 o más rutas. IGRP hará esto cuando 2 o más rutas sean igualmente buenas. El usuario puede configurarlo para repartir el tráfico cuando las rutas sean igualmente buenas.

La métrica utilizada por IGRP incluye:

- ✓ el retardo de la topología (topological delay time)
- ✓ el ancho de banda (bandwidth of the narrowest bandwidth segment of the path)
- ✓ la ocupación de la línea (channel occupancy of the path)
- ✓ la fiabilidad (reliability of the path)

El retardo de la topología es la cantidad de tiempo que pasa hasta llegar al destino a través de la ruta, asumiendo una red no cargada. Desde luego hay un retardo adicional cuando la red está cargada.

- ✓ De todos modos, la carga se mide por la ocupación del canal, no intentando medir el retraso actual.
- ✓ El ancho de banda de la ruta es simplemente el ancho de banda en bits por segundo del enlace más lento de la ruta.
- ✓ La ocupación del canal indica cuánto de este ancho de banda está actualmente en uso. Éste es medido y cambiará con la carga.
- ✓ La fiabilidad indica la actual tasa de error. Es una fracción de los paquetes que llegan al destino sin error. Se mide.

Aunque no son usadas como parte de la métrica, dos piezas de información adicionales son pasadas con ella: la cuenta de saltos y la MTU (Maximun Transfer Unit). El contador de saltos es simplemente el número de routers que el paquete debe atravesar para llegar al destino deseado. Y la MTU es el máximo tamaño de paquete que puede ser enviado a lo largo de todo el trayecto sin fragmentación. Es la mínima de las MTUs de todas las redes incluidas en la ruta al destino.

Basado en la información de la métrica, una simple "métrica compuesta" es calculada para la ruta. Esta métrica compuesta combina el efecto de varios componentes métricos en un número simple que representa lo buena que es la ruta. Esta métrica se usa para decidir la mejor ruta.

Cuando un router es por primera vez encendido, su tabla de routing es inicializada. Esto, debe ser hecho por un operador desde un terminal, o bien leyendo la información desde los archivos de configuración. Se proporciona una descripción de cada red conectada al router, incluyendo el retraso a través del enlace (cuánto le cuesta a un bit atravesar el enlace) y el ancho de banda del enlace

Periódicamente cada router emite broadcast su tabla entera de routing a los routers vecinos. Cuando un router recibe esta información de otro router, compara la tabla con la suya. Cualquier nuevo destino o ruta es añadida a la tabla de routing del router. Las rutas en el broadcast son comparadas con las rutas existentes. Si una nueva ruta es mejor, remplazará la que tenía por la nueva. La información en el broadcast es también utilizada para actualizar la ocupación del canal y otra información sobre las rutas existentes.

El proceso básico de construcción de las tablas de routing por intercambio de información con los vecinos es descrito por el algoritmo de Bellman - Ford.

En IGRP, el algoritmo general de Bellman-Ford es modificado en tres aspectos críticos:

1.- en lugar de una métrica simple, un vector de métricas es utilizado para caracterizar la ruta. Una simple métrica compuesta puede ser computada a partir de este vector de acuerdo con la ecuación 1. El uso de un vector permite al router acomodar diferentes tipos de servicio utilizando coeficientes distintos en la ecu.1.

2.- en lugar de escoger la ruta con la métrica más pequeña, el tráfico es repartido entre diferentes rutas, cuyas métricas caen dentro de un determinado rango. Esto permite distintas rutas para ser utilizadas en paralelo, proporcionando un ancho de banda efectivo mayor que con una solo ruta. Una varianza V es especificada por el administrador de red. Todas las rutas con métrica mínima se mantienen. También, todas las rutas cuya métrica es menor que VxM se mantienen. El tráfico es distribuido a través de múltiples rutas en una proporción inversa a las métricas compuestas.

3.-diferentes características son introducidas para proporcionar estabilidad en situaciones donde la topología está cambiando. Estas características han sido propuestas para prevenir bucles en la topología y el problema de la cuenta a infinito. Las principales características de estabilidad son: "holddowns", "triggered updates", "split horizon", and "poisoning".

El reparto de tráfico (punto 2.) entraña un peligro. La varianza V está designada para permitir al router usar rutas paralelas de diferente velocidad. Si la varianza es 1, solo la mejor ruta será usada. Subiendo la varianza podemos permitir al tráfico ser repartido entre la mejor ruta y otras rutas que están cerca de ser tan buena como la mejor. Pero existe el peligro de que con una varianza suficiente grande, rutas que no solo son más lentas sino que actualmente van en la dirección equivocada, se vuelvan válidas. No se envía tráfico a través de caminos cuya métrica remota (la métrica calculada en el siguiente salto) sea mayor que la métrica calculada en el router. En general, los administradores de sistema han llegado al

acuerdo de utilizar una varianza de valor 1, excepto en situaciones específicas donde se necesita usar rutas paralelas.

La mejor ruta es elegida según una métrica compuesta (composite metric) descrita a continuación:

[(K1 /Be) + (K2 * Dc)] r **ecuación 1**

Donde:

K1, K2: constantes → indican el peso asignado al ancho de banda y al delay. Dependerán del "tipo de servicio"

Be: ancho de banda efectivo. Ancho de banda cuando la red no está cargada x (1 - ocupación del canal)

Dc: delay

r: (reliability) fiabilidad → % de transmisiones que son recibidas con éxito en el siguinte salto

En principio, Dc (composite delay), puede ser definido como:

Dc = Ds + Dcir + Dt

Donde:

Ds =switching delay

Dcir =delay del circuito (retardo de propagación de 1 bit)

Dt =retardo de transmisión

La ruta que minimice esta métrica será la mejor.

Cuando existe más de una ruta para un mismo destino, el router puede enrutar los paquetes por más de una ruta.

Se dan 2 ventajas por utilizar un vector de información métrica:

1.-proporciona capacidad de soportar múltiples "tipos de servicio" desde el mismo conjunto de datos.

2.-precisión

Cuando se utiliza una métrica simple, normalmente se trata como si fuera un delay. Cada enlace en el camino es añadido a la métrica total. Si hay un enlace con un bajo ancho de banda, normalmente se representa por un gran delay.

IGRP proporciona un sistema para la interconexión de redes de ordenadores que pueden de forma estable manejar un grafo de la topología incluyendo bucles. El sistema mantiene mucha información métrica de rutas, o sea, conoce los parámetros de ruta de todas las otras redes a las cuales algún router está conectado. El tráfico puede ser distribuido sobre caminos paralelos y múltiples parámetros del camino pueden ser simultáneamente computados sobre la red entera.

IGRP está definido para manejar múltiples tipos de servicio y múltiples protocolos. El "tipo de servicio" es una especificación en un paquete de datos que modifica las rutas a ser evaluadas. Por ejemplo, en TCP/IP el protocolo permite al paquete especificar la importancia relativa de un gran ancho de banda, bajo retardo, o alta fiabilidad. Generalmente, las aplicaciones interactivas especificarán un bajo retardo y las aplicaciones de transferencia especificarán un gran ancho de banda. Estos requerimientos determinan los valores de K1 y K2 que son utilizados en la ecuación 1. Cada combinación de especificaciones en el paquete que va a ser soportada se refiere a un "tipo de servicio". Para cada tipo de servicio, un conjunto de parámetros K1 y K2 puede ser elegido. Una tabla de routing es mantenida para cada tipo de servicio. Esto se hace porque las rutas son elegidas y ordenadas de acuerdo con la métrica compuesta definida por la ecuación 1. Esto es diferente para cada tipo de servicio. La información procedente de todas estas tablas de routing es combinada para producir mensajes de actualización de la información de routing que son intercambiados por los routers.

PROTOCOLO EIGRP.

El protocolo de enrutamiento de gateway interior mejorado (Enchaced Interior Gateway Routing Protocol, **EIGRP**) es una versión mejorada del protocolo **IGRP** original desarrollado por Cisco Systems. EIGRP combina las ventajas de los protocolos de estado de enlace con las de los protocolos de vector de distancia.[8]

ATRIBUTOS:

- ✓ Protocolo vector distancia avanzado.
- ✓ Soporta VLSM.
- ✓ Soporta sumarización manual en las interfaces necesarias.
- ✓ Manda updates parciales.
- ✓ Utiliza unicast y multicast en vez de broadcast.
- ✓ Soporta varios protocolos de capa 2.
- ✓ Utiliza mensajes de HELLO para mantener sus vecindades.
- ✓ Utiliza el algoritmo DUAL para determinar sus rutas.
- ✓ Utiliza el RTP para garantizar la transferencia de información.
- ✓ Tiene módulos independientes para cada protocolo ruteado.

EIGRP mantiene las siguientes tres tablas:

- ✓ Tabla de vecinos
- ✓ Tabla de topología
- ✓ Tabla de enrutamiento

Los routers vecinos se descubren por medio de un protocolo **Hello** sencillo intercambiado por los routers que pertenecen a la misma red física estableciendo adyacencias. Hello utiliza para intercambiar paquetes de saludo una dirección multicast **224.0.0.10**. Una vez descubiertos los routers vecinos, EIGRP utiliza un protocolo de transporte fiable(RTP) para

garantizar la entrega correcta y ordenada de la información y las actualizaciones de la tabla de enrutamiento.

Un router hace el seguimiento de sus propias rutas conectadas y, además, de todas las rutas publicas de los routers vecinos. Basándose en esta información, EIGRP puede seleccionar eficaz y rápidamente la ruta de menor coste hasta un destino y garantizar que la ruta no forma parte de un bucle de enrutamiento, esta ruta escogida como principal será la llamada Sucesor.

Al almacenar la información de enrutamiento de los routers vecinos, el algoritmo puede determinar con mayor rapidez una ruta de sustitución o un Sucesor factible en caso de que haya un fallo de enlace o cualquier otro evento de modificación de la topología. El saludo y la información de enrutamiento EIGRP son transportados mediante el protocolo de transporte EIGRP. El transporte EIGRP define un protocolo fiable de publicación, acuse de recibo y petición para garantizar que el saludo y la información de enrutamiento de distribuyen adecuadamente a todos los routers vecinos. Cuando existen cambios de topologías EIGRP recurre a **DUAL** (algoritmo de actualización difusa) para conseguir una rápida convergencia entre los routers, estos almacenan sus propias tablas de enrutamiento con rutas alternativas (**Sucesor factible**), si no existiera alguna ruta alternativa EIGRP recurres a sus routers vecinos para conseguir información acerca de ese camino alternativo.

CONCEPTOS

Distancia factible (FD). Es la métrica calculada más baja para llegar a la red de destino. FD es la métrica enumerada en la entrada de la tabla de enrutamiento como el segundo número dentro de paréntesis. De la misma manera que con otros protocolos de enrutamiento también se conoce como la métrica de la ruta.

EJ: D 192.168.1.0/24 [90/3014400] vía 192.168.10.10, 00:00:31, Serial0/0/1

Sucesor

Un sucesor es un router vecino que se utiliza para el renvío de paquetes y es la ruta menos costosa hacia la red de destino. La dirección IP del sucesor

se muestra en una entrada de tabla de enrutamiento justo después de la palabra vía.

EJ: D 192.168.1.0/24 [90/3014400] vía 192.168.10.10, 00:00:31, Serial0/0/1

Sucesor Factible (FS)

Un sucesor factible (FS) s un vecino que tiene una ruta de respaldo sin bucles hacia la misma red que el sucesor por cumplir con la condición de factibilidad. Para que un router sea un sucesor factible, debe satisfacer la condición de factibilidad (FC).

Condición de Factibilidad (FC)

La condición de factibilidad (FC) se cumple cuando la distancia notificada (RD) de un vecino hacia una red es menor que la distancia factible del router local hacia la misma red de destino. La distancia notificada o la distancia publicada es simplemente una distancia factible EIGRP de vecinos a la misma red de destino. La distancia notificada es la métrica que un router informa a un vecino acerca de su propio costo hacia esa red.

Sintaxis de la configuración de EIGRP

router(config)#router eigrp 240
router(config-router)#network network-number
router(config-if)#bandwidth kilobits

router eigrp 240 especifica como protocolo de enrutamiento a EIGRP para el sistema autónomo 240, este valor varia de 1 a 65535

network específica las redes directamente conectadas al router que serán anunciadas por EIGRP

bandwidth el proceso de enrutamiento utiliza el comando bandwidth para calcular la métrica y es conveniente configurar el comando para que coincida con la velocidad de línea de la interfaz.

En versiones actuales de **IOS EIGRP** agrega al comando network la correspondiente wilcard esto permite al protocolo la identificación de subredes,
router(config)#router eigrp 240

router(config-router)#network 192.168.16.0 0.0.0.255(En versiones de IOS 12.x se puede colocar la mascara y solo calculará la Wilcard)

Algunos comandos para la verificación y control EIGRP son:

show ip route

Muestra la tabla de enrutamiento

show ip protocols

Muestra los parámetros del protocolo

show ip eigrp neighbors

Muestra la información de los vecinos EIGRP

show ip eigrp topology

Muestra la tabla de topología EIGRP

debug ip eigrp

Muestra la información de los paquetes

Métrica compuesta EIGRP y valores K

EIGRP utiliza los siguientes valores que componen su métrica:

- ✓ Bandwidth
- ✓ Delay
- ✓ Reliability
- ✓ Load

Cisco recomienda que no se utilice la confiabilidad ni la carga a menos que el administrador tenga una necesidad explicita de hacerlo.

Formula por defecto

Métrica= [K1*ancho de banda+K3*retraso]

Formula completa

Métrica=[K1*Bandwidth+(K2*Bandwidth)/(256-carga)+K3*Delay] *[K5/(Reliability+K4)]

K1 y K3 se establecen en 1 y K2, K4 y K5 se establecen en 0

Los valores K predeterminados pueden cambiarse con el comando R1(config-router)#metric weights tos k1 k2 k3 k4 k5

tos (Type of service)

Con el comando show ip protocols podremos ver los valores K

El ancho de banda lo podremos ver con el comando show interface MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, reliability 255/255, txload 1/255, rxload 1/255

Ancho de Banda. La modificación del ancho de banda no cambia el ancho de banda real del enlace, pero este valor si se usa para los cálculos de la métrica. El ancho de banda se muestra en Kbit(kilobits) por defecto se muestra 1544 Kbit ó 1544000 bps(1544 Mbps) T1.

Para modificar el ancho de banda utilizamos el comando bandwidth **R1(config-if)#bandwidth kilobits**

Y para deshabilitarla:

R1(config-if)#no bandwidth

Delay. MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, reliability 255/255, txload 1/255, rxload 1/255

Es el tiempo que necesita un paquete en atravesar una ruta, es un valor estático dependiendo del tipo de enlace. El valor de Delay y el de Bandwidth son valores predeterminados que pueden ser modificados por el administrador.

Confiabilidad. MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, reliability 255/255, txload 1/255, rxload 1/255

Es la probabilidad o la frecuencia con la que un enlace puede presentar errores, se mide dinámicamente con un valor desde 0 hasta 255, siendo la

mínima confiabilidad 1 y la máxima 255. La confiabilidad se calcula en un promedio ponderado de 5 min.

Carga. MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec, reliability 255/255, txload 1/255, rxload 1/255

Refleja la cantidad de trafico que utiliza el enlace, se mide dinámicamente con un valor entre 0 y 255, mientras menor es el valor es mejor para nuestro enlace. El txload es un valor de carga de transmisión o saliente, el rxload es un valor de carga entrante o receptor. Se calcula en un promedio ponderado de 5 min.

EIGRP de forma predeterminada no utiliza la carga en sus cálculos para la métrica.

Calculo de la métrica EIGRP. En la Ruta se elige el Bandwidth más lento.

Null0 Summary Route. EIGRP incluye automáticamente una ruta resumida hacia Null0 como ruta secundaria, esto sucede cuando:

- ✓ Por lo menos existe una subred que se aprendió a través de EIGRP.
- ✓ El resumen automático se encuentra habilitado.

Null0 no es una interfaz real. si un paquete no coincide con una de las rutas secundarias de nivel 2, se lo envía a la interfaz Null0, es decir si el paquete coincide con la dirección de red con clase pero no lo hace con ninguna de las subredes, se desecha el paquete.

EIGRP resume automáticamente en los bordes de las redes principales mediante el comando auto-summary.

Para deshabilitar el resumen automático se utiliza no auto-summary

Ejemplo:

```
D       3.0.0.0/8 is a summary, 00:00:04, Null0
D       3.3.0.0/16 [90/3651840] vía 192.168.10.10, 00:00:04, Serial0/0/1
D       3.6.0.0/16 [90/3651840] vía 192.168.10.10, 00:00:04, Serial0/0/1
D       3.9.0.0/16 [90/3651840] vía 192.168.10.10, 00:00:04, Serial0/0/1
```

Fíjense que el router conoce 3 subredes que coinciden con la red con clase 3.0.0.0 /8 y además el router esta sumarizando por tanto, si el

paquete coincide con la dirección de red con clase pero no lo hace con ninguna de las subredes, se desecha el paquete enviándolo a la interfaz Null0.Si un paquete va dirigido a la Red 3.4.0.0, este paquete coincide con la red de clase mayor 3.0.0.0 pero no coincide con ninguna subred.

Resume Manual. Para establecer el resumen manual en todas las interfaces que envían paquetes EIGRP se utiliza el siguiente comando: Router (config-if) #ip summary-address EIGRP as-number network-address subnet-mask

EIGRP Default Route. El uso de la ruta hacia 0.0.0.0/0 como ruta por defecto no depende de ningún protocolo de enrutamiento. La ruta por defecto generalmente se configura en el router que tiene una conexión con una red fuera del dominio de enrutamiento EIGRP. EIGRP requiere del comando redistribute static para que incluya esta ruta estática en sus actualizaciones EIGRP de otros routers.

R2(config)#ip route 0.0.0.0 0.0.0.0 Serial 0/0
R2(config)#router eigrp 1
R2(config-router)#redistribute static

Utilización del Ancho de Banda EIGRP. Por defecto EIGRP sólo utilizará el 50% del ancho de banda de una interfaz para información EIGRP, el comando ip bandwidth-percent EIGRP se puede utilizar para configurar el porcentaje de ancho de banda.

R1(config)#interface serial 0/0/0
R1(config-if)#bandwidth 64
R1(config-if)#ip bandwidth-percent eigrp 1[numero de sistema autonomo] 75 [porcentaje de utilizacion de los 64 kb, se ocupara solo hasta 48 kb para informacion de EIGRP]

Configurar Hello y Hold Times. Los intervalos de saludos y tiempos de espera se configuran por interfaz y no tienen que coincidir con otros routers EIGRP para establecer adyacencias.

Router (config-if) #ip hello-interval eigrp as-number seconds. Si cambia el intervalo de Hello no olvide de cambiar también e tiempo de espera a un valor igual o superior al intervalo de saludo.

Router (config-if) #ip hold-time eigrp as-number seconds. El valor en segundos varía desde 1 a 65535, más de 18 horas. Si le anteponemos no a los comandos regresaremos a los valores predeterminados.

R2(config)#int s0/0/0
R2(config-if)#ip hello-interface eigrp 1 60
R2(config-if)#ip hold-time eigrp 1 180
R2(config-if)#end

OSPF[9]

¿Qué es OSPF?. OSPF es un protocolo de routing dinámico que usa el algoritmo de estado del enlace de Dijkstra LSA-Link State Algorithm, a diferencia de otros protocolos (como RIP) un sistema autónomo AS de vector distancia.

Es un protocolo de pasarela interno (IGP) y opera en un único sistema autónomo (AS).

Es OSPF cada router advierte el estado de sus propios enlaces o conexiones en anuncios de estado del enlace (LSA) que se envían como paquetes multicast a otros routers de la red. Además, cada router usa los LSA´s que recibe de otros routers para construir un grafo que representa la topología de la red. Para construir su propia tabla de rutas el router OSPF aplica el algoritmo SPF (Shortest Path First) de Dijkstra para encontrar el mejor camino (ruta mas corta) en el grafo a cada red de la topología representada. Ese "árbol de rutas cortas" se convierte en la base de la tabla de rutas de cada router OSPF.

OSPF es jerárquico, es decir, en OSPF la red se divide en áreas, dentro de cada área los routers envían solo información local de rutas. La información de routing entre áreas se calcula mediante sumarización de rutas o supernetting intercambiadas entre áreas, lo que reduce la cantidad de información sobre la topología de la red que los routers han de intercambiar, generar y mantener haciendo a OSFP una buena alternativa para grandes redes corporativas.

Open Short Path First versión 2, es un protocolo de routing interno basado en el estado del enlace o algoritmo Short Path First, estándar de Internet, que ha sido desarrollado por un grupo de trabajo del Internet Engineering task Force, cuya especificación viene recogida en el RFC 2328.

OSPF, ha sido pensado para el entorno de Internet y su pila de protocolos TCP/IP, como un protocolo de routing interno, es decir, que

distribuye información entre routers que pertenecen al mismo Sistema Autónomo.

¿Por qué OSPF?. OSPF es la respuesta de IAB a través del IETF, ante la necesidad de crear un protocolo de routing interno que cubriera las necesidades en Internet de routing interno que el protocolo RIP versión 1 ponía de manifiesto:

- Lenta respuesta a los cambios que se producían en la topología de la red.
- Poco bagaje en las métricas utilizadas para medir la distancia entre nodos.
- Imposibilidad de repartir el trafico entre dos nodos por varios caminos si estos existían por la creación de bucles que saturaban la red.
- Imposibilidad de discernir diferentes tipos de servicios.
- Imposibilidad de discernir entre host, routers, diferentes tipos de redes dentro de un mismo Sistema Autónomo.

Algunos de estos puntos han sido resueltos por RIP versión 2 que cuenta con un mayor número de métricas así como soporta CIRD, routing por subnet y transmisión multicast.

Pero el desarrollo de OSPF por parte del IETF se basa fundamentalmente en la introducción de una algoritmia diferente de la utilizada hasta el momento en los protocolos estándar de routing interno en TCP/IP para el cálculo del camino mínimo entre dos nodos de una red:

Algoritmo de Dijkstra. El algoritmo puede ser descrito como:

N= conjunto de nodos en la red.

S = nodo origen.

M = conjunto de nodos incorporados en un instante t por el algoritmo.

D ij = el coste del enlace del nodo i al nodo j. Teniendo en cuenta que:

Dii = 0;

Dij = infinito si los dos nodos no están conectados directamente.

Dn = coste del camino de coste mínimo desde un nodo s hacia un nodo n que es conocido por el algoritmo.

El algoritmo tiene tres pasos; los pasos 2 y 3 son repetidos hasta que M = N, es decir, se han calculado todos los caminos posibles con todos los nodos de la red.

1.- Inicializar:

M = {s}

Dn = dsn para n<>s

2.- Encontrar el nodo vecino que no está en M tal que

Dw = min DjDw = min Dj

Y j no pertenece a M.

Añadir w a M.

3.- Actualizar el camino de coste mínimo:

Dn = min [Dn, Dw + dwn] para todo n no perteneciente a M.

Si el último término es el mínimo, el camino desde s hasta n es ahora el camino desde s hasta w concatenado con el enlace desde w hasta n.

Mensajes de OSPF. Existen cinco tipos de mensajes del protocolo OSPF:

- **HELLO o Saludo** se usa para:

Identificar a los vecinos, para crear una base de datos en mapa local.

Enviar señales de <estoy vivo>, al resto de routers para mantener el mapa local.

Elegir un router designado para una red multi envío

Encontrar al router designado existente.

Enviar señales de <estoy vivo>

- **Database Description Packets o Descripción de la base de datos** se usa para:

Intercambiar información para que un router pueda descubrir los datos que le faltan durante la fase de inicialización o sincronización cuando dos nodos han establecido una conectividad.

- **Link State Request o Petición del estado del enlace** se usa para pedir datos que un router se ha dado cuenta que le faltan en su base de datos o que están obsoletos durante la fase de intercambio de información entre dos routers.
- **Link State Request o Actualización del estado del enlace** se usa como respuesta a los mensajes de Petición de estado del enlace y también para informar dinámicamente de los cambios en la topología de la red. El emisor retransmitirá hasta que se confirme con un mensaje de ACK.
- **Link State ACK o ACK del estado del enlace** se usa para confirmar la recepción de una Actualización del estado del enlace.

Funcionamiento básico de OSPF. El fundamento principal en el cual se basa un protocolo de estado de enlace es en la existencia de un mapa de la red el cual es poseído por todos los nodos y que regularmente es actualizado.

Para llevar a cabo este propósito la red debe de ser capaz de entre otros objetivos de:

- Almacenar en cada nodo el mapa de la red.
- Ante cualquier cambio en la estructura de la red actuar rápidamente, con seguridad si crear bucles y teniendo en cuenta posibles particiones o uniones de la red.

Mapa de Red Local. La creación del mapa de red local en cada router de la red se realiza a través de una tabla donde:

- Fila: representa a un router de la red; y cualquier cambio que le ocurra a ese router será reflejado en este registro de la tabla a través de los registros de descripción.
- Columna: representa los atributos de un router que son almacenados para cada nodo. Entre los principales atributos por nodo tenemos: un identificador de interfase, el número de enlace e información acerca del estado del enlace, o sea, el destino y la distancia o métrica.

Con esta información en todos los router de la red el objetivo es que cada router sea capaz de crear su propio mapa de la red, que sean todos idénticos lo cual implicará que no se produzcan bucles y que la creación de este mapa de red local se realiza en los router lo más rápido posible.

Ejemplo

A --- 1 --- B --- 2 --- C --- 4 --- D --- 3 --- A

		DE	A	ENLACE	DISTANCIA
		A	B	1	1
		B	C	2	1
		C	D	4	1
		D	A	3	1
B	A	1		1	
		C	B	2	1
		D	C	4	1
		A	D	3	1

Los routers envían periódicamente mensajes HELLO para que el resto de routers, tanto si pertenecen al mapa local como a un circuito virtual para sepan que están activos.

Para que un router sepa que sus mensajes se están escuchando los mensajes HELLO incluyen una lista de todos los identificadores de los vecinos cuyos saludos ha oído el emisor.

Respuesta ante un cambio en la topología de la red. Un cambio en la topología de la red es detectado en primer lugar o por el nodo que causo el cambio o por los nodos afectados por el enlace que provoco el cambio. El protocolo o mecanismo de actualización la información por la red debe ser rápido y seguro, y estos son los objetivos del protocolo de inundación y de intercambio o sincronización empleado en OSPF.

Protocolo de Inundación: The flooding Protocol. Este protocolo consiste en el paso de mensajes entre nodos, partiendo el mensaje del nodo o nodos que han advertido el cambio, tal que cada nodo envía el mensaje recibido por todas sus interfaces menos por la que le llega siempre y cuando no haya recibido ese mensaje, para ello cada mensaje cuenta con un identificador de mensaje o contador de tiempo para constatar su validez.

Ejemplo

Supongamos que en la red anterior el enlace que va del nodo A hacia B, queda fuera de servicio tal que la distancia pasa a ser infinito.

El mensaje que A enviara a D será:

Desde A hacia B, enlace 1, distancia infinito, numero 2.

El mensaje que B enviara a C será:

Desde B hacia A, enlace 1, distancia infinito, numero 2.

La base de datos después del protocolo de flooding quedaría:

DE	*A*	*ENLACE*	*DISTANCIA*	*NUMERO*
A	B	1	infinito	2
	B	C 2	1	1
	C	D 4	1	1
	D	A 3	1	1
B	A	1	infinito	2
	C	B 2	1	1
	D	C 4	1	1
	A	D 3	1	1

Hay que tener que un cambio en un enlace de la red puede dejar aislados a unos nodos de la red, es decir, puede partir la red. Este cambio tal como está planteado el mapa local no es problema ya que aunque todos los nodos de la red inicial no tendrán el mismo mapa local este si que será idéntico para cada uno de los nodos en cada una de sus particiones.

Del mismo modo debemos considerar el caso contrario que ocurre cuando un cambio en la topología de la red provoca una unión de redes de nodos, ya que pueden surgir problemas como la existencia de enlaces modificados en un mapa local de un nodo de una subred que no esta modificado en el mapa local de la otra subred. El proceso mediante el cual se produce el chequeo del mapa local de las diferentes subredes para formar uno idéntico para todos los nodos de la nueva red se denomina:

Protocolo de Chequeo de Mapas: Bringing Up Adjacencies. Se basa en la existencia de que existen identificadores de enlace y número de versiones, a partir de estos OSPF forma unos paquetes de descripción del mapa local e inicializa un proceso de sincronización entre un par de routers de la red que tiene dos fases:

Intercambio de paquetes de descripción del mapa local entre los nodos y en cada nodo creación de una lista de nodos especiales a tener en cuenta o bien porque su número de versión es mayor que la copia local o bien porque no existía en ese mapa local el identificador del enlace.

Creación en cada nodo de paquetes con información acerca de esos nodos especiales que se envían a sus vecinos para que corroboren la información.

Tras terminar este intercambio de información, ambos routers conocen:

- Nodos que son obsoletos en su mapa local.
- Nodos que no existían en su mapa local.

Los mensajes que se usan para solicitar todas las entradas que necesiten actualización son los Link State Request o mensajes de petición de estado de enlace.

Los mensajes de respuesta son los Link State Update.

Características de OSPF. Las principales características son:

Respuesta rápida y sin bucles ante cambios. La algoritmia SPF sobre la que se basa OSPF permite con la tecnología actual que existe en los nodos un tiempo de respuesta en cuanto tiempo de computación para el calculo del mapa local de la red mucho más rápido que dicho calculo en el protocolo RIP. Además como todos los nodos de la red calculan el mapa de manera idéntica y poseen el mismo mapa se genera sin bucles ni nodos que se encuentren contando en infinito; principal problema sufrido por los protocolos basados en la algoritmia de vector distancia como RIP.

Seguridad ante los cambios. Para que el algoritmo de routing funcione adecuadamente debe existir una copia idéntica de la topología de la red en cada nodo de esta.

Existen diversos fallos que pueden ocurrir en la red como fallos de los protocolos de sincronización o inundación, errores de memoria, introducción de información errónea.

El protocolo OSPF especifica que todos los intercambios entre routers deben ser autentificados. El OSPF permite una variedad de esquemas de autentificación y también permite seleccionar un esquema para un área diferente al esquema de otra área. La idea detrás de la autentificación es garantizar que sólo los routers confiables difundan información de routing.

Soporte de múltiples métricas. La tecnología actual hace que sea posible soportar varias métricas en paralelo. Evaluando el camino entre dos nodos en base a diferentes métricas es tener distintos mejores caminos según la métrica utilizada en cada caso, pero surge la duda de cual es el mejor. Esta elección se realizara en base a los requisitos que existan en la comunicación.

Diferentes métricas utilizadas pueden ser:

- Mayor rendimiento
- Menor retardo
- Menor coste
- Mayor fiabilidad

La posibilidad de utilizar varias métricas para el calculo de una ruta, implica que OSPF provea de un mecanismo para que una vez elegida una métrica en un paquete para realizar su routing esta sea la misma siempre para ese paquete, esta característica dota a OSPF de un routing de servicio de tipo en base a la métrica.

Balanceado de carga en múltiples caminos. OSPF permite el balanceado de carga entre los nodos que exista más de un camino. Para realizar este balanceo aplica:

- Una versión de SPF con una modificación que impide la creación de bucles parciales.
- Un algoritmo que permite calcular la cantidad de tráfico que debe ser enviado por cada camino.

Escalabilidad en el crecimiento de rutas externas. El continuo crecimiento de Internet es debido a que cada vez son más los sistemas autónomos que se conectan entre si a través de routers externos. Además de tener en cuenta la posibilidad de acceder al exterior del sistema autónomo a través de un determinado router externo u otro se debe tener en cuenta que se tiene varios proveedores de servicios y es más versátil elegir en cada momento el router exterior y servicio requerido que establecer una ruta y servicio por defecto cuando se trata de routing externo como se tenia hasta ahora.

OSPF soluciona este problema permitiendo tener en la base de datos del mapa local los denominados "Gateway link state records". Estos registros nos permiten almacenar el valor de las métricas calculadas y hacen más fácil el cálculo de la ruta óptima para el exterior. Por cada entrada externa existirá una nueva entrada de tipo "Gateway link state records" en la base de datos, es decir, la base de datos crecerá linealmente con el número de entradas externas tal como ocurre con los protocolos de vector distancia, pero el coste del calculo de las rutas crecerá en función de N*log*N para OSPF y no en función de N^2 como ocurre en los protocolos de vector distancia.

Integrando OSPF a la tecnología actual. Una de las grandes ventajas de OSPF es que este ha sido diseñado para adaptarse al máximo a los protocolos TCP/IP.

Redes Locales. La existencia de redes locales formadas por host que se conectaban a un router para acceder al exterior era un hecho patente cuando se creo OSPF y siguiendo el procedimiento explicado anteriormente cada nodo hubiese tenido que especificar su enlace con el router.

OSPF introduce un nuevo enlace el "link to a stub network" que es una variante del "router link" que basándose en el concepto de subred del modelo IP permite asignar a la red local un número de subred y especificar solamente un enlace entre el router y la subred.

El enlace hacia un vecino es identificado por la dirección IP de su vecino y el enlace hacia la red local es identificado por su red o número de subred.

Redes Broadcast. OSPF da soporte a los servicios broadcast para ello implementa un mecanismo que simula el funcionamiento broadcast que se basa en la elección de un router como maestro a través del cual se pasaran todas las comunicaciones entre dos routers, es decir se establece el "designated router" y se crea un "virtual node".

Para realizar el mapa local cada router tendrá dos enlaces:

- Un enlace de él hacia su propia red broadcast cuyo enlace conocerá el propio router.
- Un enlace de él hacia el "virtual node", que será identificado por el router designado o"designated router"

La presencia del "designated router" es la de simplificar el procedimiento broadcast, ya que cuando un router quiere enviar un mensaje envía un mensaje al "designated router" usando la dirección multicast "all-designated router" (224.0.0.6). Si es un nuevo mensaje el "designated router" lo renvía a la red usando la dirección multicast "all-OSPF-routers" (224.0.0.5).

Si el "designated router" tiene problemas de funcionamiento todo este procedimiento fallará, por ello cuando se elige al "designated router" OSPF también elige al mismo tiempo al"backup designated router" con el cual también mantienen enlaces virtuales todos los routers, que en caso de fallo asumirá el rol de router designado y otro router será elegido como backup.

El router de backup permanece siempre en escucha de todos los mensajes cuya dirección multicast es "all-designated-router" a la espera del fallo del "designated router", que es detectado por el protocolo HELLO del OSPF.

Redes No Broadcast. En la documentación de OSPF este tipo de redes son aquellas que ofrecen conectividad entre todos sus miembros pero no permiten un servicio broadcast o multicast como pueden ser redes "frame-relay o"ATM".

OSPF trata este tipo de redes con un mecanismo parecido al explicado en redes broadcast, eligiendo al "designated router" y al "backup router", pero estableciendo los circuitos virtuales entre routers solo bajo demanda.

En estas redes los mensajes son enviados punto a punto, del "designated router" a cada uno de los routers. De igual modo cuando un router envía un mensaje al "designated router" lo envía también al "backup designated".

Routing Jerárquico. El routing jerárquico surge de la necesidad de resolver el problema debido al aumento del tamaño de las redes que implica un mayor coste en calculo de rutas, tiempo de transmisión de datos, memoria.

OSPF establece una jerarquía en la red y la parte en "áreas", existiendo una área especial denominada "backbone área".

En un "área" se aplica el protocolo OSPF de manera independiente como si de una red aislada se tratase, es decir, los routers del área solo contiene en su mapa local la topología del área, así que el coste en calculo es proporcional al tamaño del área y no de la totalidad de la red.

Cada área incluye un conjunto de subredes IP. La comunicación entre routers de un área se resuelve directamente a través del mapa local de área que cada router posee.

Estas áreas se conectan entre si a través del "backbone área", mediante routers que pertenecen normalmente a una "área" y al "backbone área". Estos routers se denominan "area-border routers" y como mínimo existe uno entre un área y el backbone.

Los "area-border routers" mantienen varios mapas locales de estado de enlaces, uno por cada área a las cuales pertenecen. Estos emiten unos registros de estados de enlaces para anunciar que conjunto de subredes IP son accesibles a través de ellos. Cuando un router de un área quiere intercambiar tráfico con un router de otra área, estos deben realizarlo a través de los "area-border routers". Estas se denominan "inward routes".

Existe otro tipo de router el que realiza el intercambio de tráfico con routers de otros sistemas autónomos. La información almacenada en cada router externo es idéntica para cada una de ellos

La sumarización de registros representa los enlaces entre un "area-border router" y una red en el "backbone área" o en otra área. La métrica utilizada es la longitud del camino entre el "area-border router" y la red. Este mecanismo va a permitir que diferentes "area-border router" establezcan para un destino diferentes caminos, según el resultado de su métrica pero con la salvedad de que no producirán bucles, debido a que la estricta jerarquía de OSPF solo permite que se conecten áreas a través del backbone.

OSPF provee en su jerarquía de routing la posibilidad de que un área se divida en dos a causa de algún fallo en los enlaces o en los routers pero siempre se quedan los fragmentos conectados directamente al "backbone área" a través de dos condiciones:

Los "area-border router" solo se guardan los enlaces de las redes y subredes que son alcanzables por ese router en un momento determinado.

El "backbone área" se guarde información de las redes que componen cada área aunque no de su topología.

El mecanismo OSPF para solucionar el caso de una partición del "área backbone" está un poco sujeto a por donde se realiza está partición ya que este podrá ser cubierto siempre y cuando existan "area-border router" que sean capaces de establecer caminos virtuales por dentro de sus áreas para establecer nuevos caminos de intercambio de información.

Estos describirán enlaces virtuales que deben ser almacenados en la base de registros del "área backbone".

La métrica del enlace virtual será calculada teniendo en cuenta el coste de los enlaces reales por los que pasa el enlace virtual en el área local donde se realiza el enlace virtual.

A partir de este enlace virtual deben ser sincronizados y actualizados todos los routers del "área backbone".

Stub Áreas. El problema del incremento de rutas externas que debían ser sumarizadas en multitud de áreas pequeñas ha quedado resuelto con la introducción del concepto de "stub área" un área donde todas las rutas externas son sumarizadas por una ruta por defecto.

Una stub área funciona exactamente igual que una arrea normal de OSPF con unas cuantas restricciones, acerca de prohibir la entrada de rutas externas en las bases de datos de los routers.

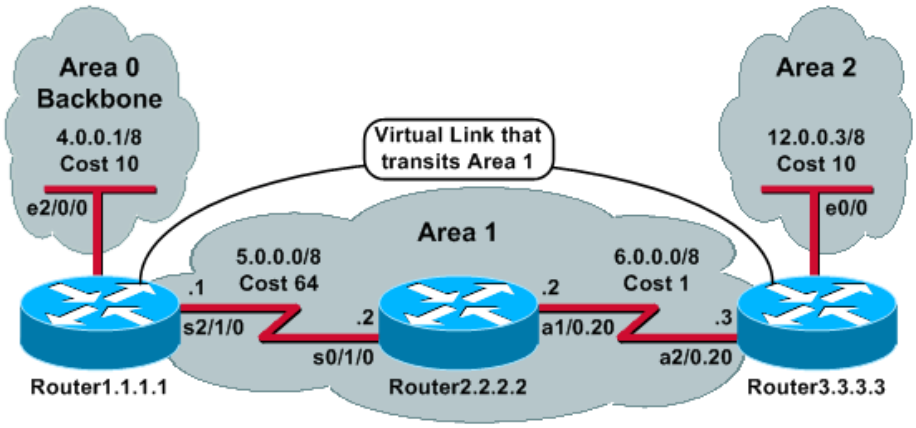
Una stub área puede estar conectada por mas de un "area-border router" al backbone, pero no se podrá elegir para salir del área el router, ni configurar un enlace virtual sobre una stub área.

También no se podrá conectar un "border router" con una "stub área". Esto es lógico si nosotros consideramos que los "border routers" conectan los sistemas autónomos con Internet y normalmente deberían estar sujetos a la "backbone área".

VIRTUAL LINK

El diseño de dos capas de OSPF requiere que todas las áreas estén conectadas directamente al área 0

Un "Virtual Link" es un enlace que permite la ínter conectividad de áreas discontinuas o la conexión de un área que no está conectada al área 0



Todas la áreas en un sistema OSPF deben estar conectadas físicamente al área 0. En algunos casos en donde esta conexión física no es posible, puede utilizar virtual links para conectarse al área 0 por medio de un área diferente. También puede utilizar los virtual link para conectar dos partes de área 0 divididas a través de un área diferente. El área por la cual quiere configurar el virtual link conocida como área de transito, debe tener la información de enrutamiento completa. El área de transito no puede ser un área de rutas internas.

NAT

Vamos a explicar los conceptos básicos de lo que conocemos como NAT. No se pretende dar una explicación profunda de todos los tipos de NAT existentes, sino dar una explicación que, sirva para comprender el porqué de la necesidad de este mecanismo, qué tiene que ver con lo que se conoce habitualmente como "abrir puertos" (como popularmente se conoce), y entender que pasa en nuestro router cuando "abrimos" esos puertos.

Unas palabras sobre direcciones IP. Como seguramente muchos habrán oído hablar, hay dos tipos (en realidad más, pero sólo nos interesan estos dos) de direcciones IP: direcciones públicas y direcciones privadas.

Las direcciones privadas son rangos especiales de direcciones IP que se reservan para ser utilizadas en redes locales, y se llaman privadas (o no-enrutables) porque **no** pueden ser utilizadas en Internet. Los routers intermedios que componen todo Internet, no "entienden" este tipo de direcciones y no las encaminan.

Esto da una gran flexibilidad para configurar redes locales, ya que por ejemplo, yo puedo tener en mi red local direcciones del tipo 192.168.0.0, y mi vecino también, pero como esas direcciones no salen de la red local no hay ningún conflicto. Esto no pasa con las direcciones públicas, que son las que se usan en Internet, y han de pertenecer a un único equipo (host); no puede haber varios con la misma IP pública.

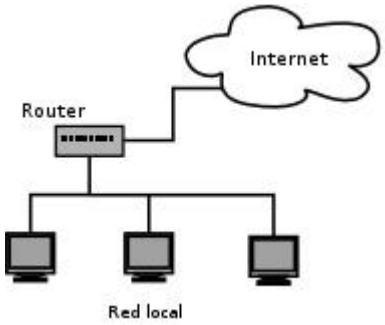
Pero esto tiene también un problema. ¿No hemos dicho antes que estas direcciones no pueden ser usadas en Internet? Entonces, cuando nosotros enviamos a Internet (a google, por ejemplo) algo desde nuestra red local, en el paquete que enviamos figura como IP de origen nuestra IP privada... ¿Cómo va a poder devolvernos el paquete (en ese caso google) si ha de devolverlo a una IP que no puede ser usada en Internet?

La solución a este problema es NAT: Network Address Translation (o Traducción de Direcciones de RED)

NAT: Traducción de Direcciones de Red. La idea básica que hay detrás de NAT es traducir las IPs privadas de la red en una IP publica para que la

red pueda enviar paquetes al exterior; y traducir luego esa IP publica, de nuevo a la IP privada del pc que envió el paquete, para que pueda recibirlo una vez llega la respuesta. Con un ejemplo lo veremos mejor.

Imaginemos que tenemos nuestra siguiente red:



Podría ser la típica red casera en la que tenemos un par de PCs que salen a Internet a través del router. Cada PC tiene asignada una IP privada, y el router tiene su IP privada (puerta de enlace) y su IP pública (que es nuestra IP de Internet).

Cuando uno de los PCs de la red local quiere enviar un paquete a Internet, se lo envía al router (o a la puerta de enlace o gateway), y este hace lo que se conoce como SNAT (Source-NAT) y cambia la dirección de origen por si IP pública. Así, el host remoto sabrá a qué IP pública ha de enviar sus paquetes. Cuando una respuesta o un paquete pertenecientes a esa conexión lleguen al router, éste traducirá la dirección IP de destino del paquete (que ahora es la IP del router) y la cambiará por la dirección privada del host que corresponde, para hacer la entrega del paquete a la red local.

DNAT: Destination-NAT. Hasta ahora hemos visto como actúa el software de NAT para permitir que un PC de una red privada pueda acceder a Internet y recibir respuestas. El mecanismo que utiliza NAT para las asociaciones entre IP pública e IP privada es una tabla (tabla de NAT) en la que guarda una entrada por cada conexión. Cuando un host de la red local inicia una conexión hacia el exterior, el software de NAT asigna una entrada en la tabla, para que a partir de ahora, todo lo que llegue perteneciente a esa conexión sepa traducirlo hacia la IP privada que inició la conexión.

Pero ¿qué pasa si la conexión se inicia desde el exterior? Por ejemplo, si montamos en nuestra red local un servidor web, lo que queremos es que se puedan iniciar conexiones hacia él. Para poder hacer esto se utiliza DNAT (Destination-NAT).

Cuando iniciábamos una conexión desde la red local, se creaba automáticamente una entrada en la tabla de NAT para que todo lo que perteneciera a esa conexión fuera dirigido hacia el PC correspondiente. Pero si la conexión se inicia desde fuera ¿cómo y cuándo se crea esa entrada en la tabla de NAT? La respuesta es que si queremos permitir conexiones desde el exterior a un PC de nuestra red local, hemos de añadir una **entrada fija** en la tabla de NAT, indicando que todo el tráfico que llegue que vaya a determinado puerto, sea dirigido al PC en cuestión. El puerto es el único elemento que tenemos para "distinguir" conexiones, ya que todo llegará a la IP del router, pero tendrán un puerto de destino según sea una conexión u otra. Así que, en nuestro ejemplo, deberíamos crear una entrada fija en la tabla de NAT en la que indicáramos que lo que llegue al puerto 80 (web) sea dirigido al PC en el que corre el servidor web.

Esto es lo que se conoce habitualmente como "abrir puertos" en el router. Al abrir puertos, simplemente estamos añadiendo una entrada a la tabla de NAT del router para que sepa hacer la traducción y sepa a qué PC enviar los paquetes. Ya que desde el exterior, aunque nuestra red tenga varios PCs, se verá como si sólo fuera uno (solo se conoce la IP del router, éste lo traduce todo) y necesitamos que éste router al que le llega todo el tráfico sepa a quién ha de entregárselo.

TIPOS DE NAT Y CONFIGURACIÓN EN CISCO. NAT estático. Consiste básicamente en un tipo de NAT en el cuál se mapea una dirección IP privada con una dirección IP pública de forma estática. De esta manera, cada equipo en la red privada debe tener su correspondiente IP pública asignada para poder acceder a Internet. La principal desventaja de este esquema es que por cada equipo que se desee tenga acceso a Internet se debe contratar una IP pública. Además, es posible que haya direcciones IP públicas sin usar (porque los equipos que las tienen asignadas están apagados, por ejemplo), mientras que hay equipos que no puedan tener acceso a Internet (porque no tienen ninguna IP pública mapeada). Para configurar este tipo de NAT en Cisco nos valemos de los siguientes comandos, donde se ve que el equipo con IP 192.168.1.6 conectado por

medio de la interfaz fastEthernet 0/0 será nateado con la IP pública 200.41.58.112 por medio de la interfaz de salida serial 0/0.

```
Router(config)# ip nat inside source static 192.168.1.6 200.41.58.112
Router(config)# interface fastEthernet 0/0
Router(config-if)# ip nat inside
Router(config)# interface serial 0/0
Router(config-if)# ip nat outside
```

NAT dinámico. Este tipo de NAT pretende mejorar varios aspectos del NAT estático dado que utiliza un pool de IPs públicas para un pool de IPs privadas que serán mapeadas de forma dinámica y a demanda. La ventaja de este esquema es que si se tienen por ejemplo 5 IPs públicas y 10 máquinas en la red privada, las primeras 5 máquinas en conectarse tendrán acceso a Internet. Si suponemos que no más de 5 máquinas estarán encendidas de forma simultánea nos garantiza que todas las máquinas de nuestra red privada tendrán salida a Internet eventualmente. Para configurar este tipo de NAT definimos el pool de IPs públicas disponibles y el rango de direcciones privadas que deseamos que sean nateadas.

En el siguiente ejemplo se cuenta con las direcciones IPs públicas desde la 163.10.90.2 a la 163.10.90.6 y la subred privada 192.168.1.0/24.

```
Router(config)# ip nat pool name DIR_NAT_GLOB 163.10.90.2 163.10.90.6 netmask 255.255.255.240
Router(config)# access-list 10 permit 192.168.1.0 0.0.0.255
Router(config)# ip nat inside source list 10 pool DIR_NAT_GLOB
Router(config)# interface fastEthernet 0/0
Router(config-if)# ip nat inside
Router(config)# interface serial 0/0
```

Router(config-if)# ip nat outside

NAT con sobrecarga. El caso de NAT con sobrecarga o PAT (Port Address Translation) es el más común de todos y el más usado en los hogares. Consiste en utilizar una única dirección IP pública para mapear múltiples direcciones IPs privadas. Las ventajas que brinda tienen dos enfoques: por un lado, el cliente necesita contratar una sola dirección IP pública para que las máquinas de su red tengan acceso a Internet, lo que supone un importante ahorro económico; por otro lado se ahorra un número importante de IPs públicas, lo que demora el agotamiento de las mismas.

La pregunta casi obvia es cómo puede ser que con una única dirección IP pública se mapeen múltiples IPs privadas. Bien, como su nombre lo indica, PAT hace uso de múltiples puertos para manejar las conexiones de cada host interno. Veamos esto con el siguiente ejemplo:

La PCA quiere acceder a www.netstorming.com.ar. El socket está formado por:

- IP origen: PCA.
- Puerto origen: X.
- IP destino: www.netstorming.com.ar.
- Puerto destino: 80.

Al llegar el requerimiento anterior al router que hace PAT, el mismo modifica dicha información por la siguiente:

- IP origen: router.
- Puerto origen: Y.
- IP destino: www.netstorming.com.ar.
- Puerto destino: 80.

Además, el router arma una tabla que le permite saber a qué máquina de la red interna debe dirigir la respuesta. De esta manera, cuando recibe un segmente desde el puerto 80 de www.netstorming.com.ar dirigido al puerto

Y del router, este sabe que debe redirigir dicha información al puerto X de la PCA.

La forma de configurar NAT con sobrecarga es la siguiente.

Router(config)# access-list 10 permit ip 192.168.1.0 0.0.0.255

Router(config)# ip nat inside source list 10 interface serial 0/0 overload

Router(config)# interface fastEthernet 0/0

Router(config-if)# ip nat inside

Router(config)# interface serial 0/0

Router(config-if)# ip nat outside

CONCLUSIÓN. Como se puede apreciar NAT es un mecanismo muy potente que nos permite crear redes locales con gran flexibilidad, pero también tiene sus inconvenientes. Muchas aplicaciones podrían no funcionar detrás de NAT. Pero eso es quizá un tema un poco más avanzado que queda fuera del objetivo de este tutorial. De todos modos las funcionalidades comentadas de NAT (que son el SNAT y el DNAT) son las más usuales e importantes, y comprenderlas es suficiente para tener una idea bastante buena de qué es NAT y qué es lo que pasa en los routers/gateways que dan salida a nuestra red local.

REFERENCIAS.

2010, NETSORMING, ***PERDIDOS EN INTERNET***, consultado en: http://www.netstorming.com.ar/2010/06/06/tipos-de-nat-y-configuracion-en-cisco/, en: octubre 2011

ACL (LISTAS DE ACCESO)

INTRODUCCIÓN. Antes que nada, y en caso de que algún lector no sepa (o no recuerde) qué significa ACL, éste es una sigla que traduce **lista de control de acceso -Access Control Lists en inglés-** y es un método popular en redes para controlar qué nodos de la red tienen qué permisos sobre el sistema que implementa las ACLs. En Cisco, las ACLs son un mecanismo genérico para clasificar conjuntos de direcciones o flujos de datos, en éso **yo siempre hago mucho énfasis,** porque las ACLs en CCNA se ven como un mecanismo de seguridad, pero se dan visos de lo que realmente son: **un mecanismo para clasificar direcciones y flujos de datos.**

Un sistema de red, como Squid por ejemplo, es un sistema que hace algo con el tráfico que entra y sale de él. Las ACLs interceptan el tráfico y, para cada paquete, se comparan sus valores particulares con valores predefinidos por el administrador en la Lista y, con base en ese condicionamiento, se le aplica a los paquetes alguna acción según lo que quiera el administrador que suceda.

La dinámica compleja de las ACLs es el hecho de imaginar un sólo paquete y llevarlo a una secuencia de paquetes mezclada. El hecho es que cuando en una ACL especificamos los valores que queremos comparar, realmente estamos aplicando eso a cada paquete dentro de un flujo particular de paquetes, así para diseñarla nos imaginemos sólo un paquete.

¿PARA QUÉ SIRVEN LAS ACLS EN CISCO? En el currículo de CCNA, las ACLs se usan para aplicar una política de seguridad que **permite o niega el acceso de cierta parte de la red a otra.** La granularidad de las ACLs permite que estas partes sean o bien PC específicos o partes de una subred arbitrariamente, es decir, permite que se conceda o niegue el acceso desde un único PC hasta otro, de un segmento de red a otro o cualquier combinación que se quiera.

En Cisco en general, las ACLs sirven para **clasificar conjuntos de direcciones,** por ejemplo una subred o una parte de una subred. Pero más allá de eso la palabra importante es **arbitrariamente,** porque las reglas de ACLs permiten cosas tan particulares como seleccionar los PCs que tengan

direcciones IP con el último octeto en número impar (sin importar a qué subredes pertenecen). Ésta característica hace que **Cisco** utilice ACLs **en cualquier parte en la que se deba especificar un conjunto de direcciones o un flujo de datos,** por ejemplo, en **NAT** se especifican las direcciones privadas o internas creando una ACL que permite las direcciones a traducir. Si se quiere filtrar o alterar la forma en que un protocolo de enrutamiento arma sus actualizaciones se usan listas de acceso (**route-map**) , si se quiere alterar la forma en que trabaja la tabla de enrutamiento se usan listas de acceso (**policy-based routing**), si se quiere especificar qué direcciones pasan por una VPN se usan ACLs, etc. (**IPSec**). Como se ve, **las ACLs son mucho más que un mecanismo de seguridad** y por eso es un tema **muy importante** si se quiere hacer carrera **en las certificaciones de Cisco** o tener un buen desempeño en enrutamiento y conmutación Cisco.

¿Cómo es una ACL?

Las ACLs, como ya comenté, son la especificación de **una acción** a realizar sobre **paquetes que cumplan ciertas condiciones.** Una ACL es un **conjunto de reglas** identificadas con un número o un nombre y cada regla especifica una acción y una condición, las acciones a aplicar son **permitir o denegar** todos los paquetes que cumplan la condición asociada a la regla. Una ACL se identifica con un número o un nombre y todas las reglas que tengan el mismo número/nombre hacen parte de la ACL, éstos identificadores suelen indicar también qué tanta expresividad tendrá la ACL, es decir, qué tan específicas pueden ser las reglas.

Un ejemplo de cómo es conceptualmente una ACL es así

- Lista-de-acceso X ACCION1 CONDICION1
- Lista-de-acceso X ACCION2 CONDICION2
- Lista-de-acceso X ACCION3 CONDICION3

La X es el nombre o número que identifica la ACL, por lo tanto todas las reglas anteriores componen la ACL X, una sola ACL. Si cierto paquete cumple la condición1 se le aplica la Acción1, si un paquete cumple la condición 2 se le aplica la acción 2 y así sucesivamente. Las acciones son sólo permitir o denegar y las condiciones dependen del tipo de ACL, las más simples, las estándar especifican valores para comparar con la dirección IP

origen de cada paquete, en las más expresivas, llamadas extendidas, las condiciones permiten especificar valores para comparar tanto con la dirección IP origen como con la IP destino e incluso protocolos de capa 4 y parámetros de capa 4 como puertos y banderas de la conexión TCP.

La lógica de funcionamiento de las ACLs es que **una vez que se cumpla una condición, se aplica su acción correspondiente y no se examinan más reglas de la ACL.** Ésto para disminuír la cantidad de procesamiento del enrutador, pero también tiene una consecuencia, si una regla abarca un conjunto de direcciones y otra un subconjunto del primero, la regla de subconjunto debe estar antes de la regla del conjunto completo. Por ejemplo, si yo especifico en una regla *denegar el acceso a un host de cierta subred* y en otra *permitir toda la subred,* la ACL diría *permita el acceso a todos los hosts de la subred X menos al host Y.* Si la ACL se escribe con la regla de la subred antes que la regla del host, la ACL permitiría incluso al host, porque la regla de host cumpliría también la regla de la subred y la regla del host nunca se examinaría. En otras palabras, las **reglas más específicas deben estar al principio de la ACL** para evitar que las reglas genererales se apliquen siempre y nunca se examinen las específicas. Finalmente todas las ACLs **terminan, implícitamente, con una regla *No permitir nada más*.**

Condición = ValorDeReferencia BitsAComparar, donde ValorDeReferencia tiene el formato de dirección IP y BitsAComparar es una máscara wildcard.

La condición entonces es un valor que el administrador va a escribir arbitrariamente con el fin de aplicar la acción a los paquetes que la cumplan. La condición en ACLs estándar consiste en una **dirección de referencia** y una máscara **wildcard** que indica **qué bits de la dirección origen de los paquetes comparar con la dirección de referencia** que indicó el administrador. Por ejemplo: si yo en mi red tengo una subred de dirección 192.168.1.0/26, para indicar el tráfico que provenga de todos los hosts de esa subred se escribiría la condición 192.168.1.0 0.0.0.63, la dirección es una dirección de referencia y no se puede entender sin la wildcard porque ésta dice qué bits se van a comparar. Cada **bit en cero en la WC hace comparar el bit correspondiente en la dir.** IP origen de los paquetes interceptados con la dirección de referencia escrita por el administrador.

Si yo quisiera aplicar una acción sólo a los hosts de dirección impar de esta misma subred escribiría la condición 192.168.1.1 0.0.0.62, note que traduciendo el último octeto de la WC a binario 62 = 01111110, **el cero al final le indica al enrutador que compare el último bit de la dirección de referencia con el último bit de cada paquete interceptado,** por lo tanto, como sabemos que todo número impar en binario tiene que tener el último bit en 1, la condición se cumple para cada paquete que tenga los primeros 3 octetos y el último bit iguales a la dirección de referencia, es decir, toda dirección IP de la forma 192.168.1.[impar], con el último octeto en binario así 0 X X X X X X 1, donde X es un bit cualquiera, porque un **1 en la WC significa no comparar** el bit con la dirección de referencia. Si no lo comprende, traduzca los números impares menores que 63 a binario y verá el patrón. Por ejemplo 9 = 00001001, una dirección 192.168.1.9 cumple la condición pero 192.168.1.8 no la cumple, porque 8 = 00001000 y el último bit no es 1, no todos los bits de la dirección IP origen de éste paquete particular coinciden con la dirección de referencia, el último no coincide. Note también que si yo pusiera una condición 0.0.0.1 255.255.255.**254**, eso significaría que sin importar la red de la que provenga el paquete (la WC indica no comparar los primeros 31 bits, o en otras palabras, no importa qué tenga ni la dirección de referencia ni la dirección origen de los paquetes en los primeros 31 bits), la acción se aplicaría a los paquetes cuyo origen sea una dirección impar (las que tienen el último bit en 1). **¿Será que con eso puede usted deducir qué condición se aplicaría a los paquetes que provengan de direcciones IP con el último octeto en valor par?** (Por favor no lo deje en un comentario).

¿CÓMO APLICAR LAS ACL? Finalmente, dado que entendemos la lógica fundamental de las ACLs, debemos mirar un último aspecto conceptual: ¿cómo se aplican?. La idea es que **el tráfico de red circula en dos sentidos** y en ambos sentidos los **patrones de dirección IP origen y destino se intercambian,** por lo tanto y como las ACLs se aplican a una interfaz en particular, es necesario tener en cuenta en qué sentido se aplica, porque en un sentido las reglas aplican y en otro sentido no aplicarán porque las direcciones origen no serán las mimas. Es decir, si dos PCs están transfiriendo un archivo, hay dos flujos de datos, uno del PC1 al PC2 en el que la dirección IP origen de todos los paquetes en ese sentido tienen la dirección Ip del PC1 pero el tráfico de retorno tendrá como dirección IP origen la del PC2. Lo anterior nos indica que si diseñamos una ACL que en

una de sus reglas aplica una acción a la dirección del PC1, hay que aplicarla en una interfaz en el sentido en el que ese flujo de datos provenga del PC1. El sentido del flujo se entiende como de entrada o salida del enrutador por la interfaz, es decir, si el tráfico sale del enrutador por la interfaz específica o el tráfico entra al enrutador por esa interfaz.

Supongamos que el PC1 tiene la dirección 172.17.20.20/24, el PC2 tiene la dirección 192.168.200.200/24 y nuestro enrutador es el Gateway del PC1 por la interfaz Fastethernet 0/0. Si el flujo de datos hacia el PC2, sale por una interfaz serial digamos la serial 0/0, ¿en qué interfaz y en qué sentido los paquetes de este flujo tienen como dirección IP origen la dirección IP del PC1? Si la ACL va a ser aplicada en la Fa 0/0, el flujo de datos de PC1 a PC2 entrando a Fa 0/0 tiene como dirección origen PC1, en la dirección de salida el origen es PC2 y la regla no aplicaría. En la interfaz serial, el flujo de datos entrante tendría como origen PC2 y de salida tendría como origen PC1. Dado lo anterior, si yo diseño una ACL con una regla que diga *permitir 172.17.20.20 0.0.0.0*, ésta regla sólo encontraría paquetes coincidentes en la interfaz fa 0/0 si la aplico de entrada y en la interfaz serial 0/0 si la aplico de salida.

¿QUÉ ES UNA ACL ESTÁNDAR (STANDARD ACLs)? Dentro de las ACL más comunes están las ACL estándar y las ACL extendidas, diferenciadas entre sí por su granularidad: las extendidas permiten más detalles de filtrado, ambos tipos de listas se pueden numerar o nombrar. Dentro de las menos comunes están las que CCNA Exploration llama complejas: ligadas a rangos de tiempo, reflexivas y dinámicas. Las más simples en todo sentido son las ACLs estándar, que permiten defnir tráfico con base en las direcciones IP de origen de los paquetes que correspondan con las reglas de la ACL.

Las ACL estándar entonces especifican un sólo par dirección de referencia/wildcard contra el que se comparan todos los paquetes que entren o salgan de la interfaz en la que se instale la ACL, en otras palabras, una ACL estándar filtra tráfico con base en la dirección IP origen de los paquetes. Estas ACL se crean en modo de configuración global con el comando *access-list* seguido de un número de *1 a 99 o de 1300 a 1999,* éstos rangos identifican que el tipo de ACL es estándar, otros rangos identifican ACLs extendidas (100 a 199 y 2000 a 2699). Cada regla debe tener el mismo número para pertenecer a la misma ACL, si el número

cambia, la regla en particular pertenecerá a otra ACL. Luego de *Access-list <número>* sigue la acción a ejecutar (*permit* o *deny*) y finalmente la condición que deben cumplir los paquetes para aplicarles la acción o continuar examinando más reglas. Las ACL estándar usan un sólo par dirección/wildcard para especificar la condición que deben cumplir los paquetes para que se les aplique la acción *permit* o *deny*. La condición examina la dirección IP origen de cada paquete y la compara con el par dirección/wildcard pero sólo en los bits en los que la wildcard tenga ceros.

¿CÓMO SE CONFIGURAN ACL ESTÁNDAR? Los pasos generales para configurar ACLs son 3:

1. Crear la ACL en modo de configuración global
2. Aplicar la ACL en una interfaz indicando la dirección del tráfico al que se le va a aplicar
3. Verificar su funcionamiento

La creación de la ACL consiste en crear una secuencia de reglas con un mismo identificador, cuyo orden filtre el tráfico según los objetivos. Cada regla tiene la forma *access-list <n> [permit | deny] <referencia1> <wildcard1>*, donde n es el número que identifica la ACL (0 a 99 ó 1300 a 1999 para ACLs estándar) y referenciaN/wildcardN son los pares con los que se compararán los paquetes para aplicarles la acción . Entonces una ACL tiene la forma:

- access-list <n> permit <referencia1> <wildcard1>
- access-list <n> deny <referencia2> <wildcard2>

Como todas las reglas coinciden en el número (n), la ACL está compuesta por todas las reglas listadas. Para simplificar, puse permit y deny pero en las reglas se puede elegir cualquiera de las dos según los objetivos perseguidos. Todas las ACLs terminan implícitamente en una regla deny any, es decir, al final de la lista, cualquier paquete que no haya correspondido con ninguna regla se va a descartar por defecto.

Para aplicar una ACL, ésta ya debe estar creada. Las listas de acceso se aplican en una interfaz, por lo tanto hay que ingresar en modo de interfaz y el comando tiene la forma *ip access-group <n> [in | out]* donde *n* es el

número común a todas las reglas de la ACL y las palabras *in/out* indican en qué sentido se aplicarán las reglas y ésto tiene importantes implicaciones: el tráfico en una dirección tiene ciertas direcciones IP origen pero en la otra dirección éstas mismas direcciones serán IP destino.

- interface serial 0/0
- ip access-group <n> [in|out]

Finalmente verificar la ACL se hace con varios comandos, uno es *show access-list*, que muestra todas las listas de acceso activas y cuántos paquetes han correspondido (match) con cada regla. El comando que muestra si una interfaz tiene una ACL aplicada y en qué dirección es *show ip interface*, este comando muestra mucha información, por la mitad de toda esa información dice **inbound ACL Outbound ACL.**

- show access-list
- show ip interface serial 0/0

¿QUÉ CONSIDERACIONES HAY QUE TENER PARA INSTALAR ACLS?

Denegación por defecto y Log. La primera consideración importante es tener en cuenta siempre que las listas de acceso terminan en denegación por defecto, por lo tanto, si una ACL sólo tiene reglas de denegación lo único que logra es denegar TODO el tráfico. Una ACL debe tener siempre por lo menos una regla de permitir. Algunos administradores prefieren poner una regla final, sea *deny any* o *permit any* de manera explícita para poder ver con *show access-list* cuántos paquetes se han filtrado por la última regla o mejor, cuántos paquetes no han correspondido con ninguna otra regla. Otros administradores usan la lista de acceso para recolectar información sobre el tráfico de la red, combinando reglas que terminan con la palabra *log* que hace que la ACL genere entradas de registro como si fueran mensajes del sistema. Combinar reglas *permit* con *log* hace que la acl evidencie algún tráfico que se necesita saber cómo se está comportando.

ORDEN DE VERIFICACIÓN: REGLAS ESPECÍFICAS Y GENERALES. Como cada regla se verifica en secuencia comenzando por la primera, si una regla es general, es decir, abarca más direcciones o flujos de datos que otra, ésta regla debería ir después de las más específicas. Para ilustrar ésto,

observe el siguiente ejemplo: yo quiero bloquear un host de la red 192.168.1.0/24 pero permitir el resto de esta red, necesito dos reglas: permitir la red y denegar el host, como la regla para la red es más general e incluye el host mismo, ponerla de primera va a tener como efecto que nunca se mire la regla que dice denegar el host, porque siempre aplicará la primera y no se verificarán más reglas, permitiendo al host transmitir información cuando el objetivo era denegar precisamente ese host.

La regla se debería escribir de la siguiente manera:

- access-list 1 deny 192.168.1.1 0.0.0.0
- access-list 1 permit 192.168.1.0 0.0.0.255

La anterior ACL tiene como resultado, cuando se aplica a una interfaz, que sólo el tráfico perteneciente a la red 192.168.1.0, excepto el host 192.168.1.1, puede salir por la interfaz en la que se aplique. Lo anterior siempre y cuando, el tráfico tenga como orígenes éstas direcciones.

TRÁFICO CON ORIGEN EN EL ENRUTADOR. Finalmente, cierto tráfico proveniente del enrutador no pasa por las listas de acceso, por ejemplo, el acceso a VTY (telnet/ssh) al enrutador no es examinado por las ACL, por lo tanto hay que poner una regla especial para este tráfico. La regla se llama *access-group <n>* y debe ser ACL estándar. Una regla de este tipo limita el acceso por telnet al enrutador sólo a los hosts que correspondan a la lista especificada.

CARACTERÍSTICAS DE LAS ACL ESTÁNDAR: POCA GRANULARIDAD. Antes de comentar las cualidades de las ACL extendidas (extended ACL) debemos recordar las ACL estándar y ver qué diferencia funcionan tienen las extendidas respecto a las primeras, es decir, para valorar los beneficios de las extendidas. La idea de las **ACLs estándar** es filtrar tráfico con base en las **direcciones origen** de los paquetes que entran o salen de una interfaz, aquella en la que se instala la ACL. Lo anterior implica un nivel básico de filtrado: direcciones IP origen de todos los paquetes interceptados, para ilustrarlo con un ejemplo, digamos que deseamos filtrar el tráfico proveniente de la red 192.168.1.0/26, pero que de esa red queremos permitir un host en particular y las demás redes diferentes deberían pasar.

A éstas alturas tenemos muy claro que las **ACLs son conjuntos de reglas con un identificador común** y que las **reglas aplican una acción** a los paquetes que **cumplan una condición** que, en el caso de las ACL estándar, es que tengan la dirección origen coincidente con la dirección de referencia. La ACL que filtra el tráfico como se solicita (bloquear 192.168.1.0/26, permitir un host 192.168.1.1 y permitir paquetes de cualquier otra subred) creamos la siguiente ACL:

- access-list 1 permit 192.168.1.1 0.0.0.0
- access-list 1 deny 192.168.1.0 0.0.0.63
- access-list 1 permit any

En efecto, cada vez que llegue un paquete se compararán las direcciones IP origen de cada uno con cada una de las reglas de la lista de acceso, si el paquete corresponde con alguna, se aplica la acción (*permit* o *deny*) y no se compara con ninguna otra regla. En este caso, la regla permite primero el host, luego niega la red y finalmente permite cualquier otra cosa.

La acl descrita significa que todo el tráfico del host particular se va a permitir, no se puede bloquear un tráfico específico que provenga del host, se deniega o se permite todo el tráfico y sería deseable bloquear sólo una porció de su tráfico, algo de lo que hace éste host en caso de ser necesario. Para la red también sucede lo mismo: si se pudiera bloquear sólo el tráfico que sale de esa red a un destino específico sin bloquear todo el tráfico con origen en esta red sería mucho mejor. Ese es el problema que resuelve la ACL extendida.

ACLS EXTENDIDAS. A diferencia de lo que sucede con la ACL estándar, las extendidas **permiten especificar hacia dónde se dirige el tráfico** y con ésta característica, yo puedo bloquear o permitir un tráfico **mucho más específico:** sólo tráfico que proviene del host pero se dirige a una red en particular o a otro host en particular o sólo el tráfico de una red que se dirige a otra red en particular. El truco se logra con el hecho de permitir **comparar las direcciones destino** de los paquetes contra la acl, no sólo las direcciones origen. Dentro de lo que hemos venido manejando, hablamos que una acl está compuesta por un conjunto de reglas todas con el mismo identificador, que cada regla era una línea compuesta por una acción y una condición que el paquete debe cumplir para aplicarle la acción

(permitir o denegar). Las condiciones en las acl estandar están compuestas por una dirección de referencia y una wildcard que dice qué bits de la dirección origen de los paquetes se deben comparar con la dirección de referencia, en las acls extendidas se especifica dos pares de direcciones de referencia/wildcard, un par para la dirección origen de los paquetes y otro par para la dirección destino de los mismos. Vamos a extender el ejemplo que venimos usando y usar ésta idea de filtrado más granular.

El requisito dado es permitir un host de una red, el resto de la red la vamos a bloquear y cualquier otra red la vamos a permitir. Para extender el ejemplo digamos que queremos permitir el tráfico del host, excepto lo que vaya a un host particular, digamos el 172.16.1.1, y que de la red completa queremos permitir lo que vaya a un servidor en especial de la empresa, digamos el 192.168.2.1. Las reglas de la acl estandar nos sirven de inicio, como de costumbre lo más específico lo vamos a poner de primero en la regla para evitar que las reglas más generales incluyan a las particulares.

- access-list 100 deny ip 192.168.1.1 0.0.0.0 172.16.1.1 0.0.0.0
- access-list 100 permit ip 192.168.1.1 0.0.0.0 0.0.0.0 255.255.255.255
- access-list 100 permit ip 192.168.1.0 0.0.0.63 192.168.2.1 0.0.0.0
- access-list 100 deny ip 192.168.1.0 0.0.0.63 0.0.0.0 255.255.255.255
- access-list 100 permit ip any any

En ésta lista observamos varias cosas nuevas: *ip*, las acl extendidas no sólo permiten especificar las direcciones origen y destino sino **discriminar por protocolos** e incluso por **parámetros particulares de cada protocolo** pero eso lo veremos luego, por lo pronto lo importante es que *ip* indica que todos los protocolos que se encapsulan dentro de ip serán afectados por ésta lista de acceso. En este caso, la palabra ***ip* para los protocolos es similar a *any* en las direcciones,** casi todo se encapsula en ip por lo tanto especificar ip es como especificar cualquier protocolo (de capa 4 en adelante). En vez de *ip* se puede poner un protocolo equivalente o de capa 4, por ejemplo se puede filtrar *icmp, tcp o udp,* cambiando la palabra *ip* por éstas últimas.

Otra cosa importante y nueva es un segundo par de dirección de referencia/máscara wildcard, éste segundo par compara la dirección destino de los paquetes con la dirección de la regla. Para las acls extendidas, el paquete debe coincidir tanto en la dirección origen como en la destino.

Finalmente, la dirección de referencia *0.0.0.0* con máscara wildcard *255.255.255.255.* Como esta máscara es todo unos, eso significa que ningún bit del paquete se compara con la dirección de referencia, es decir, no importa qué escriba en la dirección de referencia cualquier destino coincide. Esta máscara es lo mismo que *any,* debido a que la máscara es equivalente a cualquier dirección y puede usarse tanto para el origen como para el destino.

EXPLICACIÓN DE LA ACL. La primera regla aplica *deny* sólo si el paquete tiene como origen la dirección 192.168.1.1 y dirección destino 172.16.1.1, por lo tanto sólo el tráfico específico de entre esos host se deniega, la segunda regla permite el resto del tráfico del host hacia cualquier destino. La tercera regla permite el tráfico de la red 192.168.1.0/26 hacia el host 192.168.2.1. La 4a regla complementa a la anterior y niega todo el tráfico de la red, como ésta regla general esta después de la específica, el tráfico comparado con ésta regla ya no coincidió con el tráfico dirigido al servidor, que es una condición más específica dentro de la misma red. Finalmente cualquier tráfico que no coincida con las reglas anteriores se permite sin importar de dónde provenga y hacia dónde vaya.

¿Qué más? Finalmente y para no dejar incompleto el ejemplo, hay que instalarla en una interfaz por la que pase el tráfico que se quiere interceptar y recordar que el sentido en el que se instala la acl, indica cuáles serán las direcciones origen y destino (que se invierten si se invierte el sentido del tráfico).

- interface serial 0/0
- ip access-group 100 in

Las listas de acceso extendidas no difieren de las estándar más que en las características mencionadas, por lo tanto los comandos usados para verificar las estándar siguen siendo válidos.

- show ip interface serial 0/0

- show ip access-list

CONCLUSIÓN. Las ACLs extendidas son mucho más eficientes en el filtrado que las ACLs estándar, pero como ya he mencionado en otras entradas, las ACLs son mecanismos de clasificación de tráfico y direcciones y hay algunas aplicaciones que se corresponden mejor con las ACLS estándar que con las extendidas, por lo tanto se siguen usando tanto como las extendidas.

¿CÓMO FUNCIONAN LAS ACLS? Un tema nuevo en el currículo de CCNA Exploration son las ACLs complejas, que no son necesariamente complejas en el sentido de la dificultad sino en lo sofisticado de su funcionamiento, también menciono las ACL nombradas. Vamos a explorar algunas de ellas superficialmente y dejar pendiente la última entrada sobre ACLs: Ejemplos.

¿Qué son ACL complejas?

En CCNA Exploration varios tipos de ACLs no vistos en las versiones anteriores de la currícula, estas son denominadas ACLs complejas. La idea de las ACLs complejas es complementar lo que ya sabemos sobre ACLs estándar y extendidas con comportamientos que las hacen útiles en contextos más interesantes. Para comprender correctamente el tema de las ACL complejas debe entender bien todo lo relacionado con ACLs.

Dentro de las ACLs complejas tenemos 3 tipos: **dinámicas, reflexivas y basadas en tiempo** pero en el currículo oficial **no se ven muy a fondo** ni se dan mayores ejemplos. A continuación les describo cada una de ellas y al final de la entrada describo las acl nombradas son un tipo de acl que facilita la configuración y administración de ACLs.

ACLs DINÁMICAS. Éstas usan un **mecanismo básico de autenticación**, generalmente **Telnet**, para activar la ACL, lo que permite usar una ACL como mecanismo de autenticación o vincular una ACL con la autenticación de los usuarios con medios reconocidos. La idea consiste en crear **una regla en la ACL que sólo se activará si es disparada por algún evento**, en éste caso un acceso por **telnet** al enrutador. La regla en cuestión agrega antes de la acción (permit/deny) las palabras reservadas ***dynamic testlist timeout <n>,*** donde *n* es la cantidad de minutos que la regla será activa una vez que es disparada, luego de estos parámetros va la regla ordinaria que se hará activa, por ejemplo *permit ip host 10.1.1.1*

any. Como esta última regla está asociada con un acceso por telnet como disparador, en las líneas de vty se debe poner un comando especial *autocommand access-enable host timeout 5,* que establece el acceso permitido al telnet como disparador de la acl dinámica.

- access-list 101 permit ip any host 10.1.1.1 eq telnet
- access-list 101 *dynamic testlist timeout 10* permit ip 172.16.0.0 0.0.255.255 172.17.0.0 0.0.255.255
- interface fa 0/0
- ip access-group 101 in
- username cesarcabrera password cecab123
- line vty 0 4
- login local
- autocommand access-enable host timeout 5

El anterior listado de comandos instala una lista de acceso dinámica de entrada en la interfaz *fa 0/0* que sólo después que un usuario *cesar* abre exitosamente una sesión por telnet con la clave *cecab123* con el enrutador se activa, permitiendo acceso de la red 172.16.0.0/16 a la 172.17.0.0/16. Valga la aclaración que el comando autocommand cierra automáticamente la sesión de telnet pero dispara la acl, es decir, la sesión de telnet es sólo un disparador de la acl y no tiene que quedar activa para que la acl esté en funcionamiento.

ACLs REFLEXIVAS. Las reflexivas son un tipo de firewall primitivo que **permite el tráfico sólo si es iniciado en una dirección**, pero sin usar las banderas de conexión de TCP. Ya en las ACLs extendidas habíamos visto que en vez de IP se pueden poner otros protocolos y al final poner criterios adicionales particulares al protocolo en cuestión. Específicamente, TCP permite agregar al final del identificador de origen o destino un identificador de puerto en incluso banderas de conexión como *established*, que indica que la conexión ya se abrió. Éste caso particular de TCP es muy útil cuando se tienen dos redes de las cuales una es confiable y la otra no, entonces es preferible permitir sólo conexiones cuya solicitud provenga de la red confiable, es decir, que se abran desde la red interna y no se puedan abrir

conexiones desde la externa. Con el truco de la bandera *established* (ACK activo) se puede permitir de entrada sólo los paquetes con ésta condición, de tal manera que si llegan paquetes solicitando una conexión desde fuera (todavía no tienen el bit ACK activo) se rechazan, mientras que si las conexiones se abren desde adentro, todos los paquetes entrantes deberán tener el ACK activo y por lo tanto se van a permitir. Pero ¿qué pasa con UDP y otros protocolos no orientados a la conexión? Pues ahí entran en juego las ACL reflexivas. La idea es hacer lo mismo que el truco de established, pero basándose sólo en los parámetros básicos de capa 3 y 4.

Las ACLs reflexivas son un poco complejas en su configuración, ya que **se aplican varios comandos para establecer las entradas temporales**, adicionalmente las ACLs reflexivas **son un caso particular de ACL nombrada extendida**, por lo tanto **no se pueden configurar en ACL numeradas ni en ACLs nombradas estándar**. Primero, en una de las direcciones del tráfico se debe marcar la regla cuyo tráfico de vuelta se va a permitir con la palabra clave *reflect* <*nombre*>, donde ***nombre* es un identificador arbitrario** que le ponemos a esta instancia, luego en la dirección de vuelta del tráfico (la acl que se va a instalar en la dirección contraria) se agrega la sentencia *evaluate* <*nombre*> donde ***nombre* es el identificador arbitrario que pusimos en la otra dirección**. En otras palabras, se le pone un identificador al tráfico que inicia la acl reflexiva, luego en la otra dirección se le ordena que evalúe si el tráfico corresponde con la regla marcada para permitirlo si coincide. Finalmente se instalan las listas, una de entrada y otra de salida en la misma interfaz (el tráfico entra y sale por la misma interface).

- ip access-list extended OUTB
- permit udp 172.16.0.0 0.0.255.255 any reflect UDPTRAFFIC
- permit icmp 172.16.0.0.0.0.255.255 any reflect ICMPTRAFF
- ip access-list extended INB
- evaluate UDPTRAFFIC
- evaluate ICMPTRAFF
- interface ser 0/0

- ip access-group OUTB out
- ip access-group INB in

El listado anterior instala una lista de acceso reflexiva que permite el tráfico de UDP e ICMP sólo si se originó en la red 172.16.0.0/16.

ACLs BASADAS EN FECHAS/HORARIOS. Finalmente, las más simples de comprender son las basadas en fechas/horarios. La idea de estas ACLs son que **se activan en las fechas y horarios que se hayan establecido previamente**, la precondición evidente es que **el enrutador debe tener configuradas su hora y fecha correctamente**, para esto se puede configurar manualmente, confiando que el equipo no se vaya a reiniciar por ningún motivo y que el administrador va a mantener actualizado el reloj en caso contrario. Otra alternativa (más confiable) es configurar un servidor ntp para que el enrutador mantenga su tiempo actualizado.

La configuración de las ACLs basadas en tiempo consiste en crear un rango de tiempo (time-range) el cual es después usado en las reglas de la ACL.

- time-range NOCHES
- periodic Monday Tuesday Wednesday Thursday Friday 17:00 to 00:00
- access-list 101 permit tcp 172.16.0.0 0.0.255.255 any eq www time-range NOCHES
- int fa 0/0
- ip access-group 101 out

El anterior listado crea una lista de acceso que se permite el acceso a Internet para la red 172.16.0.0 sólo después de las 17hrs en días de trabajo (Lunes a Viernes).

ACLs NOMBRADAS. Finalmente, hay una forma más fácil de editar las listas de acceso llamadas listas de acceso nombradas. La idea básica de éstas ACLs es permitir una administración mnemónica de las ACL, ya que en vez de números se usan nombres arbitrarios. Éstas listas pueden ser extendidas o nombradas con las mismas características que las ACLs numeradas y abren un modo especial de configuración (*nacl*) en el que se introducen las reglas una por una empezando por la acción

(*permit/deny*). Después de la versión 12.3 del IOS, éstas listas de acceso permiten eliminar y crear reglas particulares entre las reglas existentes, contrario a la edición ordinaria de ACLs en la que tocaba eliminar completamente una ACL para poder modificarla. En su configuración las palabras clave son *ip access-list*, lo que hemos visto hasta este momento, todas las listas de acceso comienzan con la palabra reservada *access-list*, éstas comienzan con *ip access-list*, seguidas del tipo de lista *extended/standard* y el *nombre* (arbitrario). Luego se entra en el modo especial de configuración.

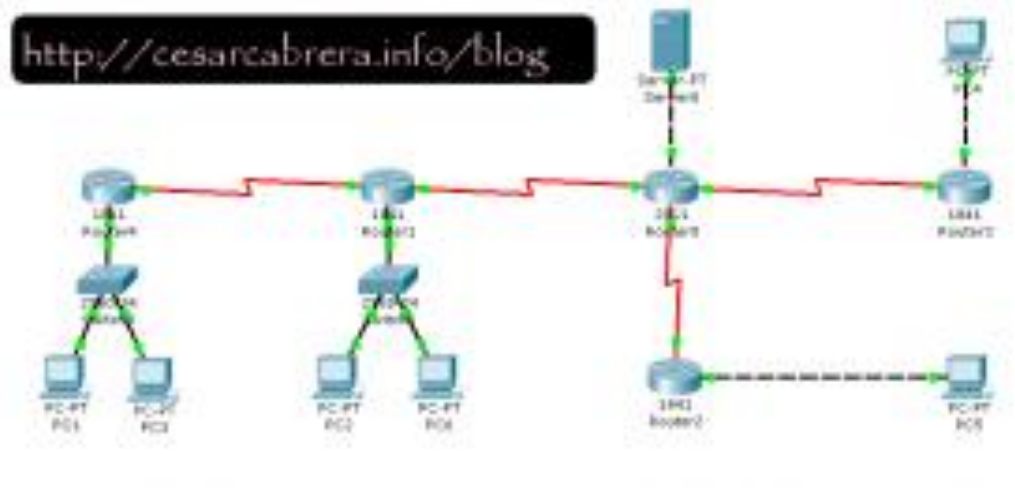
- *ip* access-list extended *INB*
- (config-ext-nacl)#permit 172.16.0.0 0.0.255.255 172.17.0.0 0.0.255.255
- (config-ext-nacl)#deny any any

Éstas listas se aplican como se aplican todas las ACLs y se verifican con los mismos comandos. *show ip access-list* y *show ip interface*.

Conclusión. Como podemos ver, las listas de control de acceso en Cisco son mucho más versátiles de lo que creíamos y eso que no hemos visto sino la aplicación en filtrado de tráfico. Como les dije en la primera entrada de la serie: las ACL son un mecanismo para clasificar tráfico y por lo tanto se usan en muchas tecnologías en los enrutadores, por ejemplo para re direccionar tráfico usando PBR (policy based routing). Para la próxima entrega haremos muchos ejemplos en una topología compleja en la que podamos ver en la práctica todo lo que hemos explorado tan teóricamente.

TOPOLOGÍA DE EJEMPLO. La siguiente va a ser nuestra topología de ejemplo. Infortunadamente el Packet Tracer no soporta las ACLs reflexivas ni dinámicas, así que si usted desea hacer el ejercicio completo debe hacerlo o bien con enrutadores reales o con GNS3. La topología consiste en 5 enrutadores interconectados por enlaces seriales, uno de ellos es el enrutador central que conecta con el servidor y el resto tienen sólo una subred conectada. Las direcciones de las LAN pertenecen al prefijo 172.16.0.0/12, asignándolas en orden de izquierda a derecha 172.16.0.0/24, 172.17.0.0/24, 172.18.0.0/24 (PC4) y 172.19.0.0/24 (PC5). El servidor tiene la dirección 10.1.1.1 y los enlaces entre enrutadores son 10.0.0.0/30, 10.0.0.4/30, 10.0.0.8/30, 10.0.0.12/30. El enrutamiento se lleva a cabo con EIGRP de AS 1 sin auto resumen por el hecho de haber redes discontinuas (10.0.0.0). La política (requerimientos) de seguridad de la organización son los siguientes:

- Estándar: Filtrar el acceso de la red del PC1 al servidor
- Extendidas: Filtrar el acceso del PC2 al servidor
- Dinámicas (No soportadas): Acceder al servidor desde PC5 sólo si se autentica previamente por telnet.
- Reflexivas (No soportadas): Permitir el tráfico de ICMP que se origine en PC4, pero no al revés



Un aspecto importante, **antes de hacer cualquier cosa con ACLs es verificar que exista conectividad completa en la red objetivo**. Si no

comprobamos eso previamente, podríamos ignorar problemas actuales de conectividad en la red y podríamos creer que eso es efecto de la instalación de las ACLs. En esos casos el diagnóstico de un problema de esa naturaleza puede resultar muy difícil de diagnosticar y más difícil aun de solucionar. Otra consideración que hay que hacer es verificar que la instalación de una ACL afecta la red estrictamente como se espera y no genera efectos no previstos e indeseados. Lo anterior hay que hacerlo por cada ACL y verificando la conectividad total -o la más importante si la red es muy grande-.

LISTAS DE CONTROL DE ACCESO ESTÁNDAR. En el ejemplo, el requerimiento de filtrar la red del PC1 con ACL estándar nos enfrenta a la primera decisión: ¿dónde instalar la ACL? La respuesta tiene dos sentidos, **en qué enrutador y en qué interfaz de ese enrutador**. Como las ACL estándar sólo filtran el tráfico con base en las direcciones IP origen, si la ACL se instala en Router1, eso filtraría todo el tráfico de la red hacia todos los destinos, por lo tanto no es viable esa decisión. Una alternativa, si no es posible instalarla en otro enrutador, sería filtrar el tráfico proveniente del servidor en ese mismo enrutador lo que impediría que las respuestas a tráfico que salió de la red de PC1 y PC3 regresen, lo cual sería un cumplimiento indirecto de la política de impedir conectividad entre esa red y el servidor.

Según lo anterior, **la única alternativa es instalar la ACL estándar en Router0**, y con eso se cumple la regla de oro de las ACLs estándar: **instale lo más cerca posible del destino**. En éste caso, en el que podemos configurar el enrutador más cercano al servidor, la instalamos en la interfaz por la que se conecta el servidor en la dirección de salida, filtrando efectivamente el tráfico cuyo origen es la red del PC1.

El resto es carpintería como decía un antiguo profesor que tuve:

- access-list 1 deny 172.16.0.0 0.0.0.255
- access-list 1 permit any
- interface FastEthernet0/0
- ip address 10.0.0.1 255.255.0.0
- ip access-group 1 out

Una vez que se instala esta ACL, los paquetes originados en cualquier PC de la red del PC1 no llegarán al servidor pero sí a cualquier otro pc de la red. La conectividad con el resto de las redes de la topología sigue intacta.

ACL extendidas. El segundo requerimiento es filtrar el tráfico desde el PC2 hasta el Servidor. Evidentemente **no se puede hacer un filtrado así con una sola ACL estándar**, por ejemplo, si ponemos una ACL estándar en router0 que filtre el tráfico cuya IP es la de PC2 en la fa 0/0 de salida, éste quedará sin conectividad con cualquier PC de la red del servidor, no sólo el servidor. Si, por otro lado, la ponemos de entrada la situación es peor: el servidor no se podrá comunicar. Otra alternativa sería instalarla en la interfaz LAN que pertenece a la red de PC2 (en Router1) y éste no se podrá comunicar con ninguna otra red.

La solución es una ACL extendida, que **por norma se instala lo más cercano al origen posible**. El razonamiento es que haciéndolo de ésta manera evitamos que tráfico innecesario corra por la red ocupando ancho de banda y procesamiento en los dispositivos.

- access-list 101 deny ip host 172.17.0.3 host 10.0.0.5
- access-list 101 permit ip any any
- interface FastEthernet0/0
- ip address 172.17.0.1 255.255.0.0
- ip access-group 101 in

Lo anterior en el enrutador Router1, donde se conecta el host que se quiere filtrar. De nuevo, hay que verificar que otros PCs no quedan afectados por la ACL, eso lo verificamos enviando y recibiendo paquetes exitosamente desde el PC6 al servidor. La conectividad del resto de las redes de la topología sigue inalterada, eso incluye la conectividad de otras redes al pc2.

ACLs DINÁMICAS Y REFLEXIVAS. Recordemos que las dinámicas son, entre otras cosas, un mecanismo de autenticación simple. Lo primero que haremos será crear un nombre de usuario y contraseña para autenticar al PC 5, luego crearemos la ACL que incluya la palabra reservada dynamic, cuidando que la misma ACL permita el tráfico de telnet desde el pc en

cuestión, instalamos la ACL y luego en la configuración de las VTY agregamos el comando que vincula estas dos instrucciones.

- username cesarcabrera password cecab123
- access-list 101 permit ip any host 172.19.0.1 eq telnet
- access-list 101 *dynamic testlist timeout 15* permit ip host 172.19.0.3 host 10.0.0.5
- interface fa 0/0
- ip access-group 101 in

Lo anterior, una vez instalado en el enrutador Router2, sólo permitirá el acceso del PC5 al servidor si primero se intenta hacer un telnet al enrutador y se autentica exitosamente al mismo.

El requerimiento de ACL reflexiva se debe instalar en el último enrutador, Router3, usando una ACL nombrada extendida -no numerada- y con dos palabras clave adicionales: reflect/evaluate. En la dirección de salida se permite el tráfico pero se establece que se creen las ACLs necesarias para el tráfico de retorno con reflect y de entrada se le indica a la ACL que evalúe las entradas dinámicas creadas por la ACL de salida.

- ip access-list extended SALIDA
- permit icmp 172.18.0.0.0.0.0.255 any reflect TICMP
- ip access-list extended ENTRADA
- evaluate TICMP
- interface ser 0/0/0
- ip access-group SALIDA out
- ip access-group ENTRADA in

Note que una vez que se instalan estos comandos en el último enrutador, lo único que se puede hacer desde la red 172.18.0.0n es enviar exitosamente pings, pero no serán exitosos si se originan en otras redes hacia ésta última.

Otros usos de las ACLs

Finalmente, como les he venido mencionando en otras entradas, las ACLs son un mecanismo de clasificación de tráfico y por eso **son útiles en otros contextos**. Voy a citar dos, uno de ccna y otro de ccnp, particularmente de BSCI. En el último semestre de CCNA se estudia el tema de NAT, ***NAT se usa para tener una red con direccionamiento privado arbitrariamente grande conectada a una red pública usando sólo un pequeño conjunto de direcciones públicas***. El mecanismo consiste en examinar los paquetes provenientes de la red privada y cambiar las direcciones IP y puertos TCP/UDP del encabezado por las direcciones públicas disponibles. De ese proceso se guarda en memoria una registro de qué puertos origen han sido asignados a qué dirección privada, de tal manera que cuando se recibe la respuesta de la red pública con IP destino pública (o global como dice el currículo), el puerto TCP/UDP destino determina la dirección IP de host local al que hay que cambiar la dirección IP para enviar el paquete al interior de nuestra red (en otra ocasión escribo más en detalle el proceso).

NAT debe especificar dos conjuntos de direcciones: las direcciones privadas a traducir a direcciones públicas y el conjunto de direcciones públicas. El conjunto de direcciones públicas es un rango de direcciones arbitrarias que difícilmente corresponderán con una regla tipo ACL, pero las direcciones privadas sí deben tener un patrón que se corresponda con una ACL estándar, en la que las direcciones a las que se aplique la acción permit serán las direcciones que hay que traducir a direcciones públicas (o globales). En otras palabras, ***para crear una regla de traducción de direcciones, se especifica por medio de una ACL qué direcciones privadas (o locales) deben ser traducidas***.

En BSCI (uno de los exámenes de ccnp) se habla de un mecanismo de manipulación de tráfico llamado **mapas de ruta (route-map)**. Los mapas de ruta **permiten manipular la forma en que se realiza el enrutamiento** por ejemplo yo podría arbitrariamente y sin contar con la tabla de enrutamiento, decir que el tráfico de cierta red debe usar siempre un enlace en particular de salida. Ése es el ejemplo más simple de un mapa de ruta, pero los mapas de ruta permiten muchas cosas más, por ejemplo cambiar parámetros de enrutamiento como métricas o filtrar actualizaciones de enrutamiento que provengan de otro enrutador. **El mecanismo básico**

por el que se especifica qué tráfico será afectado por las reglas del mapa de ruta son las ACL extendidas.

CONCLUSION. Espero que después de toda esta secuencia sobre listas de control de acceso, hayan quedado claros muchos conceptos y formas de usar las ACLs y sobre todo, ver que éstas son un mecanismo muy potente y muy importante en el mundo de la configuración de equipos de red, en especial de Cisco. **Les dejo también la topología** de ejemplo con los requerimientos para que ensayen en sus casas -si tienen el Packet Tracer-, la conectividad básica y el enrutamiento ya están configurados, **sólo faltan las ACLs**. Espero que hayan disfrutado la lectura y que les haya resultado de alguna utilidad.

- Topologia de ejemplo en PT 5.1

Autor

César A. Cabrera E.

CCNP-CCAI-CCNA. Instructor de CISCO certificado (CCAI) para cursos de CCNA desde 2005 y **docente universitario** en la UTP desde 2007. Desarrollo mis actividades en Pereira (Risaralda) y el eje cafetero, región centro-occidente de **Colombia**.

- Ingeniero de Sistemas y Computación de la UTP desde 2006.
- Estudiante de la **especialización en redes de datos** del programa de Ingeniería de sistemas y computación de la UTP desde **principios de 2009**.
- CCNP (Cisco Certified Networking Professional) hasta 2012. Los exámenes aprobados fueron BSCI, BCMSN, ISCW y ONT.
- CCNA (Cisco Certified Networking Associate) desde 2004 (válido hasta 2012 en virtud de la política de recertificación por exámenes de nivel superior).
- Artículos de tema tecnológico y social en el blog Informática++ por César A. Cabrera E. se encuentra bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0

2011, Cabrera Cesar, Informática++, **TUTORIAL SOBRE LISTAS DE ACCESO**, http://cesarcabrera.info/blog/contexto-e-instrucciones/

PROTOCOLO AAA

AAA es una arquitectura de seguridad, la cual está dividida en tres módulos (Authentication, Authorization and Accounting, por sus siglas en ingles), los cuales trabajan en conjunto, creando una forma eficiente y segura de conectarse a una red.

Sus funcionalidades son:

- **Autenticación:** Proporciona el método de identificación de usuarios, incluyendo nombre de usuario y contraseña, desafío y respuesta, soporte de mensajería, y, según el protocolo de seguridad que seleccione, puede ofrecer cifrado. La autenticación es la forma en que un usuario se identifica antes de poder acceder a la red y los servicios que esta ofrece. Se configura la autenticación AAA mediante la definición de una lista llamada métodos de autenticación, y luego aplicando esa lista a varias interfaces. En la lista de métodos se definen los tipos de autenticación a realizar y la secuencia en la que se llevará a cabo, esto debe ser aplicado a una interfaz específica antes de que cualquiera de los métodos de autenticación definidos se utilicen. La única excepción es la lista método por defecto (que se denomina "default"). lista de métodos por defecto se aplica automáticamente a todas las interfaces si ninguna lista de otro método está definida. Una lista de método definida remplaza automáticamente la lista de métodos por defecto. Todos los métodos de autenticación, excepto local, línea de contraseña y habilitación de la autenticación, deben ser definidas a través de AAA.

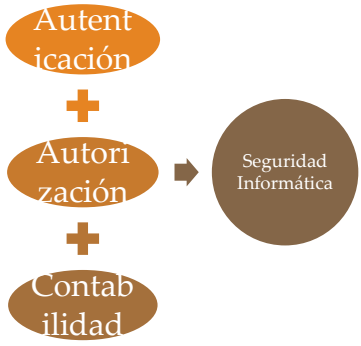
- **Autorización:** Provee el método de control de acceso remoto, incluyendo autorización total o autorización para cada servicio, lista de cuentas y perfil por usuario, soporte para grupos de usuarios, y soporte para IP, IPX, ARA y Telnet. El módulo de autorización trabaja agrupando un grupo de atributos que describen lo que el usuario está habilitado a usar o acceder. Estos atributos son comparados con la información contenida en una base de datos de un usuario determinado y el resultado se devuelve a AAA para determinar las capacidades reales de los usuarios y las restricciones. La base de datos se puede localizar de forma local en el servidor de acceso o

Router o puede ser alojado de forma remota en un servidor de seguridad RADIUS o TACACS+. Los servidores remotos de seguridad, tales como RADIUS y TACACS+, autorizan a los usuarios de los derechos específicos mediante la asociación de atributos de valor (AV) pares, que definen los derechos con el usuario apropiado. Todos los métodos de autorización deben ser definidos a través de AAA. Así como en la autenticación, configurar AAA Autorización es definida por una lista llamada métodos de autorización, y luego aplicando esa lista a varias interfaces.

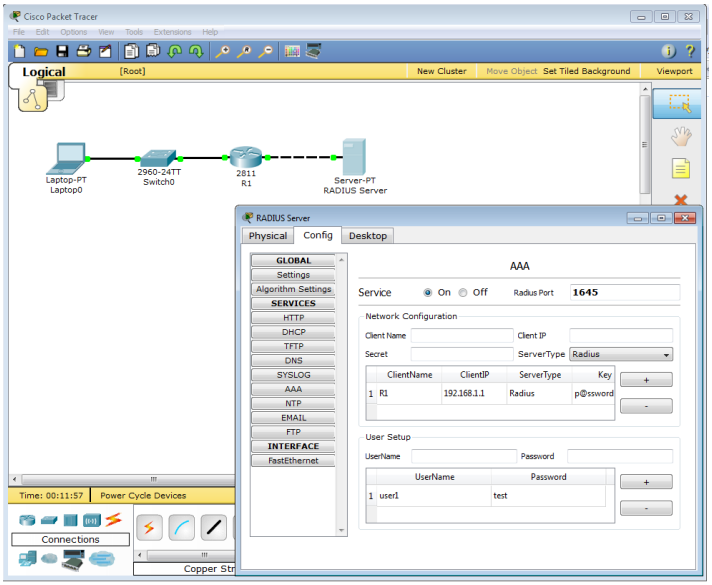
- **Contabilización:** Posee un método de recolección y envió de información al servidor de seguridad, el cual es usado para facturar, auditar y reportar: nombres de usuario, tiempo de inicio y final, comandos ejecutados (como PPP), cantidad de paquetes enviados, y número de bytes. Contabilización permite realizar el seguimiento de los usuarios que tienen acceso a los servicios, así como la cantidad de recursos de red que están consumiendo. Cuando ésta se activa, el acceso a la red del servidor informa la actividad del usuario al servidor de seguridad de RADIUS o TACACS+ (según el método de seguridad que se haya implementado) en la forma de los registros contables. Cada registro contable se compone de la contabilidad de pares AV y se almacena en el servidor de control de acceso. Estos datos pueden ser analizados para la gestión de la red, la facturación del cliente, y / o auditoría. Todos los métodos de contabilización deben ser definidos a través de AAA. Al igual que con la autenticación y autorización, este método se configura mediante la definición de una lista llamada métodos de contabilización, y luego la aplicación de esa lista a varias interfaces

AAA está diseñado para que el administrador de la red pueda configurar dinámicamente el tipo de autenticación y autorización que se quiera, puede ser por línea (por usuario) o por servicio (por ejemplo, IP, IPX, o VPDN) base. Para definir el tipo de autenticación y autorización que se desee, se hace mediante la creación de listas de método, a continuación, la aplicación de esas listas de método para determinados servicios o interfaces. Una lista de método es una lista secuencial que define los métodos de autenticación usados para autenticar usuarios. Las listas de método le permiten designar uno o varios protocolos de seguridad que se utilizarán para la autenticación, lo que garantiza un sistema de copia de seguridad para la autenticación en

caso de que el método inicial falla. El software utiliza el primer método la lista para autenticar a los usuarios, y si ese método no responde, el software selecciona el método de autenticación siguiente en la lista de métodos. Este proceso continúa hasta que haya una comunicación exitosa con un método de autenticación de la lista o la lista de método de autenticación se ha agotado, en los que la autenticación caso de falla.



¿Cómo configurar los servicios AAA usando el simulador Packet Tracer?



Router R1:

- FastEthernet 0/0 : 192.168.1.1/24
- FastEthernet 0/1 : 192.168.2.1/24

Servidor RADIUS: 192.168.1.2/24

Cliente (Laptop0): 192.168.2.1/24

Configuracion del router R1 en el CLI	
aaa new-model	El primer comando aaa new-model indica que el router que se esta manejando utiliza el protocolo TACACS+ o RADIUS para autenticación.
radius-server host 192.168.1.2 key p@ssword	Muestra al router la dirección IP del servidor RADIUS y la contraseña
aaa authentication login	El comando de autenticación AAA es usado para iniciar la autenticación de RADIUS en el router como método de login por default.
line vty 0 15 login authentication default	Configura las interfaces vty para login

2011, Packet Tracer, **PACKET TRACER 5.3 - RADIUS CONFIGURATION**, consultado en: mayo 2012, en: http://www.packettracernetwork.com/radiusconfiguration.html

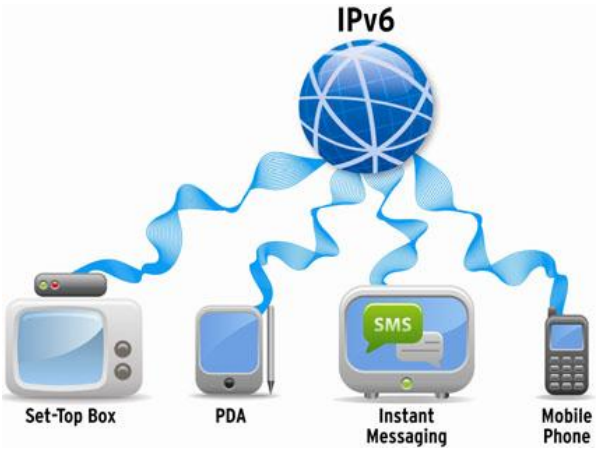
PROTOCOLO IPv6

Cuando utilizamos Internet para cualquier actividad, ya sea correo electrónico, navegación web, descarga de ficheros, o cualquier otro servicio o aplicación, la comunicación entre los diferentes elementos de la red y nuestro propio ordenador o teléfono, utiliza un protocolo que denominamos Protocolo de Internet (IP, Internet Protocol).

En los últimos años, prácticamente desde que Internet tiene un uso comercial, la versión de este protocolo es el número 4 (IPv4).

Para que los dispositivos se conecten a la red, necesitan una dirección IP. Cuando se diseñó IPv4, casi como un experimento, no se pensó que pudiera tener tanto éxito comercial, y dado que sólo dispone de 2^32 direcciones (direcciones con una longitud de 32 bits, es decir, 4.294.967.296 direcciones), junto con el imparable crecimiento de usuarios y dispositivos, implica que en pocos meses estas direcciones se agotarán.

Por este motivo, y previendo la situación, el organismo que se encarga de la estandarización de los protocolos de Internet (IETF, Internet Engineering Task Force), ha trabajado en los últimos años en una nueva versión del Protocolo de Internet, concretamente la versión 6 (IPv6), que posee direcciones con una longitud de 128 bits, es decir 2^128 posibles direcciones (340.282.366.920.938.463.463.374.607.431.768.211.456), o dicho de otro modo, 340 sextillones.



Es claro que muchas de las direcciones IPv4 que figuran como asignadas, no están siendo utilizadas por diversas razones. Durante algún tiempo se pensó que mediante la optimización del uso de las direcciones IPv4, la recuperación de direcciones no utilizadas y el incremento de uso de tecnologías tipo NAT (del inglés "Network Address Translation" o Traducción de Direcciones de Red), se podía resolver la demanda de direcciones IP, sin la necesidad de adoptar una nueva versión del Protocolo de Internet.

Gradualmente esta idea se ha ido desvaneciendo, en la medida que se ha ido viendo la enorme cantidad de dispositivos que necesitarán, en el mediano plazo, sus propias direcciones IP para conectarse a Internet, mucho de los cuales necesitarán, incluso, varias direcciones. Aún en el caso de una utilización más óptima de las direcciones IP, las más de 4 mil millones de direcciones que el protocolo IPv4 permite, no serán suficientes.

IPv6 en el entorno académico y de investigación

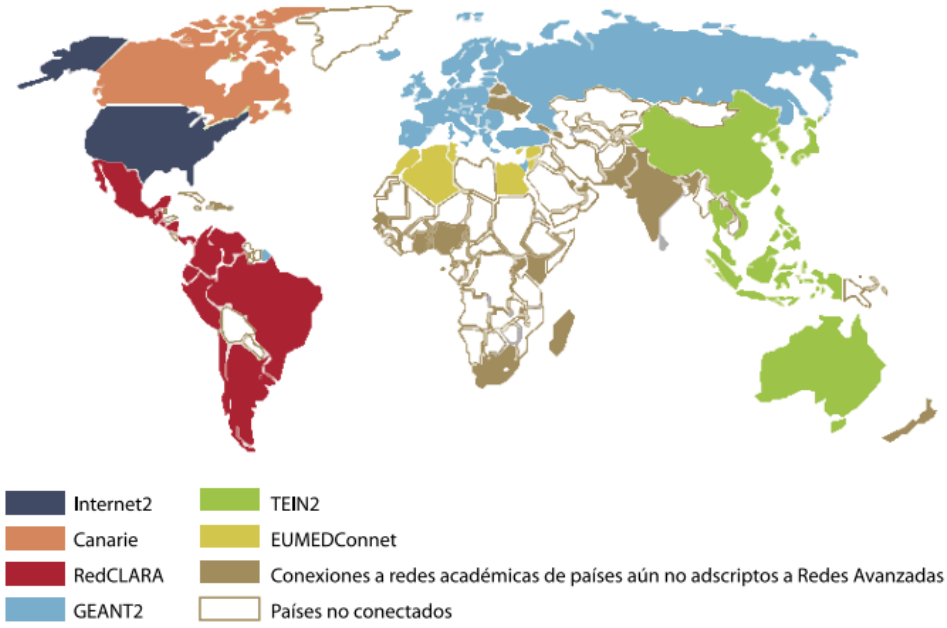
Los primeros despliegues de IPv6 en gran escala se dan en el marco de redes académicas o de investigación, como Abilene (Internet2) en EE.UU., Geant en Europa, CERNET2 y CSTNET2 en China o WIDE y JGN2 en Japón.

Una particularidad de las redes académico/científicas actuales es que cuentan con servicios que no son habituales en otro tipo de redes. Actualmente existe en este tipo de redes lo que se denomina "grids", algunas veces traducido como "mallas", que son sistemas que se encuentran en una capa entre las aplicaciones y los servicios de red y en los cuales la idea es compartir recursos que pueden estar distribuidos globalmente, accediéndolos desde sitios remotos. La posibilidad de tener direcciones IP globales, públicamente alcanzables, permite el despliegue en gran escala de los servicios grids, tanto desde el punto de vista de los recursos como de los dispositivos en condiciones de utilizarlos.

Otro tipo de tecnología habitual en este tipo de redes es multicast. Esto permite una utilización óptima del ancho de banda cuando se hacen transmisiones de datos a muchos destinatarios, ya que no es necesario replicar esa transmisión para cada receptor. Es posible entonces emitir contenidos con una señal con mejor calidad, ya que el ancho de banda a utilizar no se multiplica en función de los receptores. Multicast se utiliza para hacer streaming de video y audio, para contenidos bajo demanda, videoconferencia multipunto, etc. Si bien en IPv4 esta tecnología esta

disponible, en IPv6 es parte del protocolo desde su diseño y su utilización es mucho más sencilla.

Redes académicas en el mundo



La mayoría de estas redes tienen soporte de IPv6 desde hace años, por lo que dependiendo de la región del mundo donde nos encontremos, vamos a poder aprovechar esa disponibilidad para lograr la conectividad nativa de la institución a la que pertenecemos.

REFERENCIAS.

2012, Cicileo Guillermo, Gagliano Roque, O´Flaherty Christian, Olvera Cesar, Martínez Jordi, Rocha Mariela, Vives Alvaro, **IPV6 PARA TODOS, GUIA DE USO Y PLICACION PARA DIVERSOS ENTORNOS**, consultado en: mayo 2012, en: http://blog.utp.edu.co/libroteca/files/2012/04/IPv6-Para-Todos-Gu%C3%ADa-de-Uso-y-Aplicaci%C3%B3n-Para-Diversos-Entornos.pdf, http://blog.utp.edu.co/libroteca/category/computacion-e-informatica/

2012, Gobierno de España, **¿QUÉ ES IPV6?**, consultado en mayo 2012, en: http://www.ipv6.es/es-ES/introduccion/Paginas/QueesIPv6.aspx

REFERENCIAS

[1] 2012, Wikipedia, **PACKET TRACER**, consultado en: mayo 2012, en: http://es.wikipedia.org/wiki/Packet_Tracer

[2] 2012, GNS3, GNS3 Graphical Network Simulator, consultado en: mayo 2012, en: http://www.gns3.net/

[3] 2010, Universidad Carlos III de Madrid, **ENCAMINAMIENTO**, consultado en agosto 2011, en: http://www.it.uc3m.es/~prometeo/rsc/apuntes/encamina/encamina.html

[4] 2011, Wikipedia, Biografiar, ***EDGER DIJKSTRA***, consultado en: http://es.wikipedia.org/wiki/Edsger_Dijkstra, en marzo 2011.

Tanenbaum, Andrew S., *Computer Networks*, Prentice-Hall, 1996.

Bertsekas, D. y Gallager, R., *Data Networks*, 2ª edición, Prentice-Hall, 1992.

Comer, Douglas E., *Internetworking with TCP/IP*, 3ª edición, Volumen 1: *Principles, Potocols and Architectures*, Prentice-Hall, 1995.

[5] 2011, Morales Cristóbal Jonathan, **CISCO REDES**, consultado en: abril 2012, en: http://www.ciscoredes.com/tutoriales/65-vlan.html

[6] 2009, Gerometa Oscar, **STP EN SWITCHES CISCO**, consultado en: mayo 2012, en: http://librosnetworking.blogspot.mx/2008/09/stp-en-switches-cisco-catalyst.html

[7]2012, Wikipedia, INTERIOR GATEWAY ROUTING PROTOCOL, consultado en: mayo 2012, en: http://es.wikipedia.org/wiki/Interior_Gateway_Routing_Protocol

http://docente.ucol.mx/al964186/public_html/IGRPeigrp.htm
http://www.google.com.mx/url?sa=t&rct=j&q=&esrc=s&source=web&cd=28&ved=0CGQQFjAHOBQ&url=http%3A%2F%2Fcedesdb.net%2Fcisco%2Fmaterial%2FModulo2%2FCCNA2_Cap7.ppt&ei=e6G0T-

WyOqGXsQKHs5ChAQ&usg=AFQjCNFajwsHDOuhFxTudDVs4SdtmdanrQ&sig2=7J6k6S8Xx31NqK2pIe8Urg
http://andersonramirez.tripod.com/protocolo.htm#IGRP
2010, sistemasuma, PROTOCOLOS IGRP EIGRP, consultado en: mayo 2012, en: http://sistemasumma.wordpress.com/2010/11/08/protocolos-igrp-eigrp/

[8] 2010, Zystrax, **PROTOCOLO DE ENRUTAMIENTO EIGRP**, consultado en: abril 2012, en: http://zystrax.wordpress.com/2010/03/31/protocolo-de-enrutamiento-eigrp/
2009, García Gastón, CONFIGURAR EIGRP, PROTOCOLO DE ENRUTAMIENTO DE GATEWAY INTERIOR MEJORADO, consultado en: mayo 2012, en: http://www.garciagaston.com.ar/verpost.php?id_noticia=204
2009, **PROTOCOLOS DE ENRUTAMIENTO: EIGRP**, consultado en: mayo 2012, en: http://www.dsi.uclm.es/asignaturas/42650/PDFs/practica4.pdf

ENLACES RELACIONADOS.
http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipr_c/ipcprt2/1cfeigrp.htm#wp1000880
http://www.cisco.com/warp/public/103/eigrp-toc.html#

REFERENCIAS.

[9] "Routing in the Internet",Christian Huitema, Pretince Hall.
"TCP/IP", Dir Sidnei Feit, Mc Graw Hill.
"TCP/IP Illustrated the protocols",Volume 1,W.Richard Sterems,Addison Wesley.
"Internetworking with TCP/IP: Principles, Protocols and Architecture", Fourth Edition, Douglas E. Comer, Prentice Hall.
"Local & Metropolitan Area Network",William Stallings,Prentice Hall.
"Redes Globales de Información con Internet y TCP/IP", Comer, Douglas, Prentice may.
"RFC 1403. The Internet Society". BGP OSPF Interaction
"RFC 1584. The Internet Society". Multicast Extensions to OSPF.
"RFC 1586. The Internet Society". Guidelines for Running OSPF Over Frame relay Networks
"RFC 2328. The Internet Society". OSPF version 2

LINKS DE INTERÉS

http://www.cisco.com/warp/public/104/1.html
http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ospf.html

http://www.freesoft.org/CIE/Topics/89.html
http://www.faqs.org/rfcs/rfc1583.html