UNIVERSIDAD DE LAS AMÉRICAS

ESCUELA DE TECNOLOGIAS

ANALISIS TECNICO Y OPERATIVO DE LOS PROTOCOLOS DE ENRTUTAMIENTO OSPF Y RIP EN BASE AL USO DE SNMP PARA LA EMPRESA HUMANA S.A.

Trabajo de titulación presentado en conformidad a los requisitos para obtener el titulo de Tecnólogo en Redes y Telecomunicaciones

Profesor Guía: Ing Rodrigo Chancusig

MADRIL ACURIO PAUL VINICIO

2009

QUITO

DECLARION DEL PROFESOR GUIA

Declaro haber dirigido este trabajo a través de reuniones periódicas con el estudiante, orientando a sus conocimientos y competencias para un eficiente desarrollo del tema y tomando en cuenta la Guía de Trabajos de Titulación correspondientes.

Ing. Rodrigo Chancusig

DECLARION DEL ESTUDIANTE

Declaro que este trabajo es original, de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes.

Paúl Vinicio Madril Acurio

RESUMEN

Teniendo presente la gran importancia de las redes de datos en la sociedad de la información y el Conocimiento, resulta imprescindible facilitar su estudio especialmente en instituciones que no cuentan con la posibilidad de brindar grandes redes reales para que sus estudiantes trabajen y experimenten con ellas; de allí la importancia de las herramientas de simulación de redes, que permiten su estudio bajo diferentes cargas de tráfico, facilitando la experimentación y el estudio personalizado y a distancia de los alumnos. Los protocolos de enrutamiento para la capa de red son usados para resolver peticiones de servicios de envío de paquetes de datos a través de diferentes redes de datos. El punto más importante de este estudio es mostrar el comportamiento de los diferentes algoritmos de enrutamiento como el protocolo de enrutamiento vector-distancia (RIP) y el protocolo de enrutamiento de estado de enlace (OSPF), y a su vez compararlos en forma cualitativa para conocer cuáles son sus fortalezas y cuáles son sus puntos débiles, como la forma de incorporar automáticamente diferentes configuraciones de redes para el mismo, mediante el simulador didáctico llamado Packet Tracer 5.0 de Cisco.

INDICE

CAPITULO I

RIP

CAPITULO II
OSPF

CAPITULO III

SNMP

CAPITULO VI

ESQUEMAS DE SIMULACION

INTRODUCCION

El objetivo de la Interconexión de Redes (internetworking) es dar un servicio de comunicación de datos que involucre diversas redes con diferentes tecnologías de forma transparente para el usuario. Este concepto hace que las cuestiones técnicas particulares de cada red puedan ser ignoradas al diseñar las aplicaciones que utilizarán los usuarios de los servicios.

Los dispositivos de interconexión de redes sirven para superar las limitaciones físicas de los elementos básicos de una red, extendiendo las topologías de esta.

Teniendo en cuenta las necesidades y los avances producidos en una sociedad sumamente compleja, resulta de gran importancia destacar tanto la transmisión de información, como la necesidad de que ésta llegue al destino en el momento preciso mediante el uso de las redes.

Es a través de la Internet que queda probado, y todos los días se muestra con mejor y mayor detalle, que ésta ha sido y será revolucionaria en las áreas de los servicios financieros, de entretenimiento, salud, educación y gobierno.

El proceso de digitalización de todas las técnicas de comunicación, transmisión (cable, satélite) y recepción, producen nuevas convergencias entre diferentes sectores (cultura, comunicación, lengua, educación, telecomunicaciones, etc.), pero muy especialmente lo que producen es la transformación de los "espacios de comunicación" los límites y las fronteras y, como consecuencia, la transformación de los espacios de intercambios culturales. De hecho, todas las sociedades, por definición, han sido y serán "sociedades de la comunicación".

Los principales cambios estructurales de la sociedad se producen ahora en torno del tratamiento y de la transmisión de la información.

La capa de Red, dentro de una arquitectura de rede de datos, es la que se encarga de llevar los paquetes de datos desde el origen (estación transmisora) hasta el destino (estación receptora). Llegar a destino, en tiempo y forma, puede requerir que el algoritmo de ruteo, que es el encargado de escoger las rutas y las estructuras de datos, cumpla con ciertas propiedades que aseguren la eficiencia de su trabajo.

CAPITULO I

PROTOCOLO DE ENRUTAMIENTO RIP

Routing Informartion Protocol

La Internet es una colección de varios sistemas autónomos (AS). Cada AS posee una tecnología de enrutamiento que puede diferir de otros sistemas autónomos. El protocolo de enrutamiento utilizado dentro de un AS se conoce como Protocolo de enrutamiento interior (IGP). Un protocolo distinto utilizado para transferir información de enrutamiento entre los distintos sistemas autónomos se conoce como Protocolo de enrutamiento exterior (EGP). RIP está diseñado para trabajar como IGP en un AS de tamaño moderado. No ha sido concebido para utilizarse en entornos más complejos.

Rip v1 fue definido en el RFC 1058 en junio de 1988. RIP es un protocolo vector distancia que usa la cuenta de saltos como métrica. No soporta el uso de máscaras de sub-red ni tampoco permite autenticar el traspaso de rutas. Un router RIP envía una copia de su tabla de rutas a sus vecinos cada 30 segundos, Rip usa "Split horizon with Poisson Reverse", como método de prevención de lazos, por lo tanto las actualizaciones de rutas son enviadas fuera de la interfaz con una métrica infinita para las rutas aprendidas ó recibidas desde la misima interfaz. El estandar RIP esta basado en el popular programa "routed" del sistema operativo Unix.

RIP v1 se considera un IGP con clase. RIP v1 es un protocolo de vector-distancia que envía la tabla de enrutamiento completa en broadcast a cada router vecino a determinados intervalos. RIP utiliza el número de saltos como métrica, siendo 15 el número máximo de saltos.

Si el router recibe información sobre una red y la interfaz receptora pertenece a la misma red pero se encuentra en una subred diferente, el router aplica la máscara de subred que está configurada en la interfaz receptora:

- Para las direcciones de Clase A, la máscara con clase por defecto es 255.0.0.0.

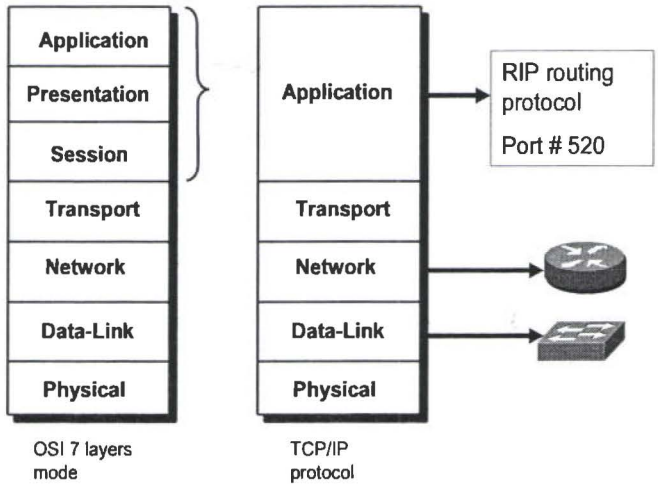
- Para las direcciones de Clase B, la máscara con clase por defecto es 255.255.0.0.
- Para las direcciones de Clase C, la máscara con clase por defecto es 255.255.255.0.

RIP v1 es un protocolo de enrutamiento común dado que prácticamente todos los routers IP lo admiten. La popularidad de RIP v1 se basa en la simplicidad y su demostrada compatibilidad universal. RIP es capaz de equilibrar las cargas hasta en seis rutas de igual costo, siendo cuatro rutas la cantidad por defecto.

RIP v1 posee las siguientes limitaciones:

- No envía información de máscara de subred en sus actualizaciones.
- Envía las actualizaciones en broadcasts a 255.255.255.255.
- No admite la autenticación
- No puede admitir enrutamiento entre dominios de VLSM o sin clase (CIDR).

Si bien la funcionalidad básica de un Router está focalizada en la capa 3 del modelo OSI RIP es un protocolo presente en las tres últimas capas del modelo OSI, el manejo de mensajes, cálculo de métricas, sumarización, etc, son actividades focalizadas en las capas superiores del modelo OSI, sin embargo, producto de este trabajo, se modifican las entradas en la tabla de rutas de los equipos. La siguiente figura ilustra esta situación.



Rip V1 entrega la siguiente información en cada mensaje de actualización

Dirección IP: Dirección IP de host o red de destino

Gateway: El primer Gateway a lo largo del camino de destino

Interface: La interfaz física que se usa para alcanzar el destino

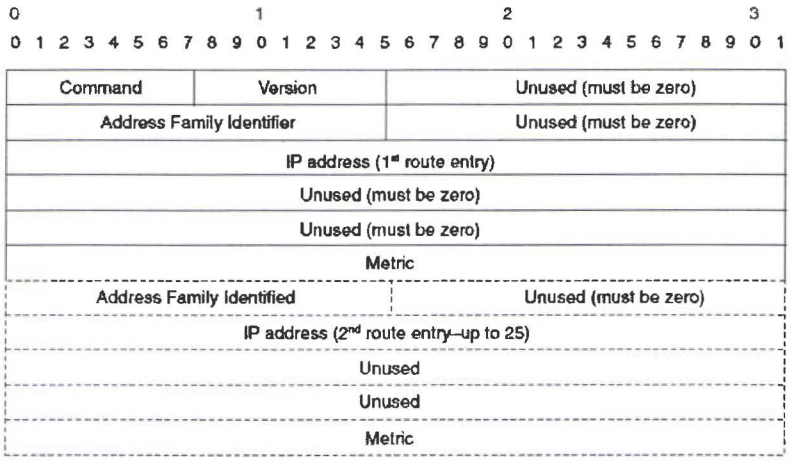
Métrica: Número de saltos hacia el destino

Timer: Tiempo transcurrido desde la última actualización.

La base de datos de rutas de cada equipo es actualizada con la información entregada en los mensajes de actualización.

Formato del Mensaje RIP

De acuerdo a la descripción entregada por el RFC 1058, la versión uno de RIP trabaja en base a paquetes UDP, los cuales tienen el siguiente formato:



La descripción de los campos de este paquete UDP se presenta a continuación:

Command: Describe el propósito del paquete, indicando, si el paquete es una consulta ó una respuesta.

Versión: Versión del Protocolo RIP

Address Family Identifier (AFI): Configura en 2 para IP.

Dirección IP: Ruta de destino

Métrica: Campo de 32 bits de largo, que indica un valor de 1 a 15 inclusive, especificando la métrica actual para la dirección de destino especificada.

Para RIP, una cantidad de saltos infinita se especifica usando la métrica 16. La información de cada ruta se entrega en 20 bytes, los cuales se dividen en 5 palabras (Words) de 32 bits cada una. Las cuales se distribuyen de la siguiente forma: AFI 16 Bits, Unused 16 Bits, IP Address 32 Bits, 2 campos unused adicionales de 32 bits cada uno, Métrica 32 Bits.

Considerando que el paquete tiene una extensión máxima de 512 bytes (sin incluir la cabecera IP), es posible enviar 25 rutas en cada actualización (25x20 + la cabecera RIP 4bytes + 8 bytes de cabecera UDP = 512 bytes)

Procesamiento de Mensajes de Actualización

Una vez que un mensaje de actualización de ruta es entregado por un router, este lo procesa de la siguiente forma:

Si el mensaje no está contenido en la tabla de rutas actual: Entonces, las rutas del mensaje serán ingresadas en la tabla de rutas del equipo, y la cuenta de saltos será recalculada.

Si el mensaje recibido tiene rutas presentes en la tabla de rutas del router y el puerto de próximo salto de estas rutas, no coinciden con el puerto de próximo salto especificado por la ruta recibida, entonces, dos acciones se pueden tomar:

Si la cuenta de saltos de la ruta existente en la tabla de rutas del router es menor o igual a la cuenta de saltos definidos en la ruta recibida incrementada en una unidad, entonces la ruta existente en la tabla de router es mantenida

Si la cuenta de saltos de la ruta existente en la tabla de rutas del router es mayor a la cuenta de saltos de la ruta recién recibida incrementado en uno, entonces la ruta existente en la tabla de rutas será reemplazada por la ruta recibida y se recalculará la cuenta de saltos.

Si la información de rutas recibidas está presente en la tabla de rutas del router, y el puerto de próximo salto de la ruta existente es consistente con el puerto de próximo salto de la ruta recibida, entonces: La ruta existente será reemplazada por la ruta recibida y la cuenta de saltos será recalculada

Estas tres reglas permiten entender la forma en que los lazos son generados y las soluciones planteadas para esta situación.

Contadores RIP

RIP incorpora el uso de los siguientes contadores:

Update*:* Este contador especifica la frecuencia con la cual se enviarán a los vecinos, las tablas de rutas de un equipo, por default está en 30 segundos.

Invalid*:* Contador que especifica el tiempo que una ruta es considerada como válida en la tabla de rutas. Cada vez que llega un mensaje de actualización

para una ruta, este contador es reiniciado en cero, si en el tiempo especificado por este contador no se ha recibido un mensaje de actualización que contenga esta ruta, la ruta es marcada como inválida con métrica 16. Por default este parámetro está configurado en 180.

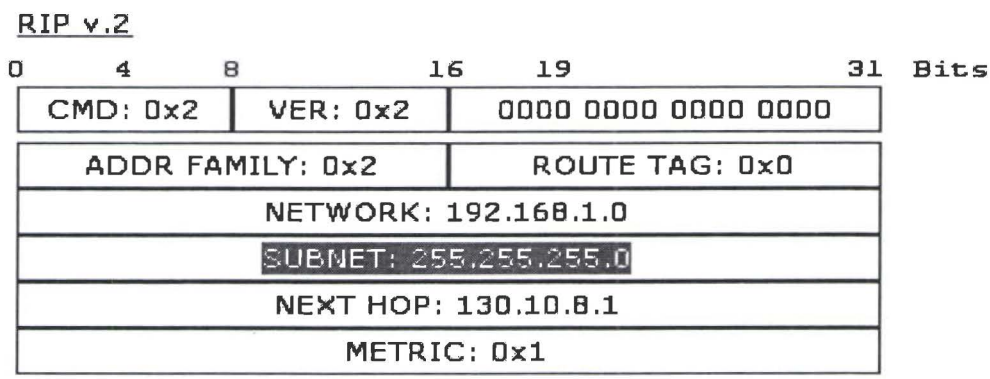
Flush: Una ruta marcada como inválida, es mantenida en la tabla de rutas, hasta que el contador de Flush expire. Este parámetro normalmente viene configurado en 240 segundos.

RIP VERSION 2

La versión 2 de Rip fue especificada inicialmente en los RFC 1388 y 1723, actualmente, las implementaciones de RIP v2 están especificadas en el RFC 2453. Rip v2 incorpora el manejo de sub-redes, autenticación de rutas, y envía mensajes de actualización usando paquetes de multicast. Las actualizaciones siguen siendo enviadas cada 30 segundos y el límite de 15 saltos como máximo es respetado. Las estrategias de prevención de lazos siguen estando basadas en "split horizon with poison Reverse".

Los mensajes de actualización toman ventaja de los campos no usados en Rip Versión 1, en los cuales agregan información sobre la máscara de sub-red entre otras cosas. Al igual que en Rip versión 1, el contenido de estos mensajes está sometido al manejo de reglas usadas para decidir cuando reemplazar las rutas y al manejo de contadores, los cuales no varían en Rip V2.

FORMATO DEL MENSAJE RIP ver2



RIP v2, que es una versión mejorada de RIP v1. Ambas versiones de RIP comparten las siguientes funciones:

- Es un protocolo de vector-distancia que usa el número de saltos como métrica.
- Utiliza temporizadores de espera para evitar los bucles de enrutamiento, la opción por defecto es 180 segundos.
- Utiliza horizonte dividido para evitar los bucles de enrutamiento.
- Utiliza 16 saltos como métrica para representar una distancia infinita

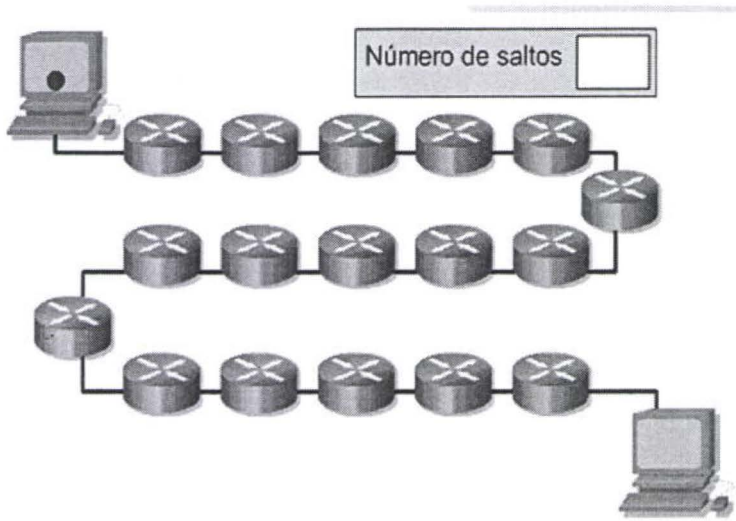
RIP v2 ofrece el enrutamiento por prefijo, que le permite enviar información de máscará de subred con la actualización de la ruta. Por lo tanto, RIP v2 admite el uso de enrutamiento sin clase en el cual diferentes subredes dentro de una misma red pueden utilizar distintas mascaras de subred, como lo hace VLSM.

RIP v2 ofrece autenticación en sus actualizaciones. Se puede utilizar un conjunto de claves en una interfaz como verificación de autenticación. RIP v2 permite elegir el tipo de autenticación que se utilizará en los paquetes RIP v2. Se puede elegir texto no cifrado o cifrado con Message-Digest 5 (MD5). El texto no cifrado es la opción por defecto. MD5 se puede usar para autenticar el origen de una actualización de enrutamiento. MD5 se utiliza generalmente para cifrar las contraseñas enable secret y no existe forma reconocida de descifrarlo.

RIP v2 envía sus actualizaciones de enrutamiento en multicast con la dirección Clase D 224.0.0.9, lo cual ofrece mejor eficiencia.

DIFERENCIAS ENTRE RIP v1 y RIP v2

RIP utiliza algoritmos por vector-distancia para determinar la dirección y la distancia hacia cualquier enlace en la internetwork. Si existen varias rutas hasta un destino, RIP elige la ruta con el menor número de saltos. Sin embargo, debido a que el número de saltos es la única métrica de enrutamiento que RIP utiliza, no siempre elige el camino más rápido hacia el destino.



RIP v1 permite que los routers actualicen sus tablas de enrutamiento a intervalos programables. El intervalo por defecto es de 30 segundos. El envío continuo de actualizaciones de enrutamiento por parte de RIP v1 implica un crecimiento muy rápido del tráfico de red. Para evitar que un paquete entre en un bucle interminable, RIP permite un número máximo de 15 saltos. Si es necesario pasar por más que 15 routers para llegar al destino, la red se considera inalcanzable y el paquete se descarta. Esta situación crea un problema de escalabilidad cuando se efectúa el enrutamiento en redes heterogéneas más grandes. RIP v1 usa el horizonte dividido para evitar los bucles. Esto significa que RIP v1 publica las rutas por una interfaz sólo si las rutas no se conocieron por medio de actualizaciones que entraron por esa interfaz. Utiliza temporizadores de espera para evitar bucles de enrutamiento. Las esperas pasan por alto cualquier nueva información acerca de una subred si esa subred tiene una métrica menos conveniente en un lapso de tiempo igual al del temporizador de espera.

RIP v2 es una versión mejorada de RIP v1. Comparte muchas de las mismas funciones que RIP v1. RIP v2 también es un protocolo de vector-distancia que utiliza el número de saltos, temporizadores de espera y horizonte dividido.

CAPITULO II

PROTOCOLOS DE ENRUTAMIENTO OSPF

El protocolo OSPF (Open Shortest Path First) (Primero la ruta libre más corta) fue creado a finales de los 80. Se diseño para cubrir las necesidades de las grandes redes IP que otros protocolos como RIP no podían soportar, incluyendo VLSM, autenticación de origen de ruta, convergencia rápida, etiquetado de rutas conocidas mediante protocolos de enrutamiento externo y publicaciones de ruta de multidifusión.

OSPF funciona dividiendo una Intranet o un sistema autónomo en unidades jerárquicas de menor tamaño. Cada una de estas áreas se enlaza a un área backbone mediante un router fronterizo. Todos los paquetes enviados desde una dirección de una estación de trabajo de un área a otra de un área diferente atraviesan el área backbone, independientemente de la existencia de una conexión directa entre las dos áreas. Aunque es posible el funcionamiento de una red OSPF únicamente con el área backbone, OSPF escala bien cuando la red se subdivide en un número de áreas más pequeñas.

OSPF es un protocolo de enrutamiento por estado de enlace que a diferencia de RIP e IGRP que publican sus rutas sólo a routers vecinos, los routers OSPF envían Publicaciones del estado de enlace LSA (Link-State Advertisment) a todos los routers pertenecientes a la misma área jerárquica mediante una multidifusión de IP. La LSA contiene información sobre las interfaces conectadas, la métrica utilizada y otros datos adicionales necesarios para calcular las bases de datos de la ruta y la topología de red. Los routers OSPF acumulan información sobre el estado de enlace y ejecutan el algoritmo SPF (que también se conoce con el nombre de su creador, Dijkstra) para calcular la ruta más corta a cada nodo. Para determinar que interfaces reciben las publicaciones de estado de enlace, los routers ejecutan el protocolo OSPF Hello. 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.

Cuando los routers están sincronizados, se dice que han formado una adyacencia.

Las LSA se envían y reciben sólo en adyacencias. La información de la LSA se transporta en paquetes mediante la capa de transporte OSPF que define un proceso fiable de publicación, acuse de recibo y petición para garantizar que la información de la LSA se distribuya adecuadamente a todos los routers de un área. Existen cuatro tipos de LSA. Los tipos más comunes son los que publican información sobre los enlaces de red conectados de un router y los que publican las redes disponibles fuera de las áreas OSPF.
La métrica de enrutamiento de OSPF es el coste que se calcula en base al ancho de banda de la interfaz y es configurable por parte del usuario.

Conceptos de Single Area de OSPF

Open Shortest Path First (OSPF), como RIP, está basado en estándares "Abiertos".**RFC 2328**

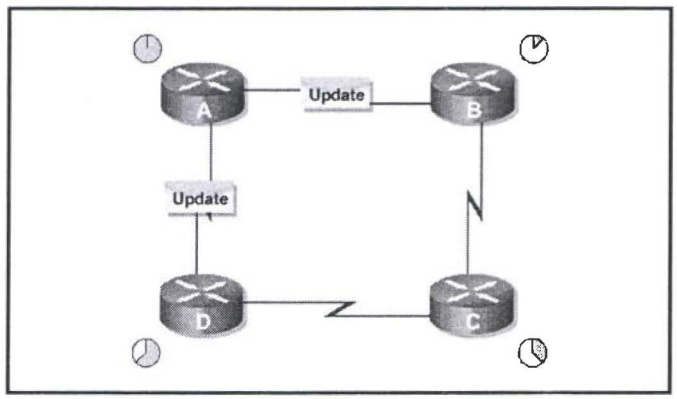
OSPF es frecuentemente preferido sobre RIP por su escalabilidad. Puede ser configurado en redes más pequeñas usando una "Área" O escalado a redes más grandes sin límites virtualmente.

Las Redes y Áreas son fácilmente agregadas o eliminadas.

Actualizaciones

RIP envía en broadcast su tabla de enrutamiento completa cada 30 segundos haya habido cambios o no. cuando un temporizador del router expira, éste envía una actualización a sus vecinos directamente conectados (vea la gráfica). Por esta simplicidad y amplio soporte a través del equipo de los vendedores, RIP es una opción favorita para muchas configuraciones de red. Sin embargo, el comportamiento de RIP llega a ser un problema cuando la red crece más

grande que 30 a 50 routers.



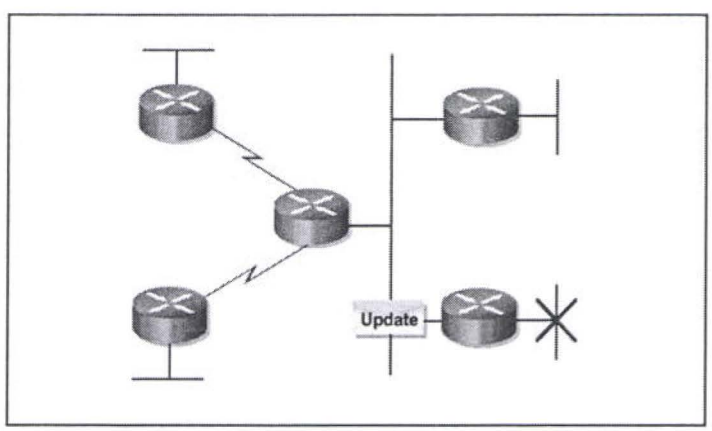
Velocidad de Convergencia

OSPF es manejado-por evento.

Solo cambios son enviados a otros routers.

Envía un LSA (link state advertisement) cuando un cambio ocurre.

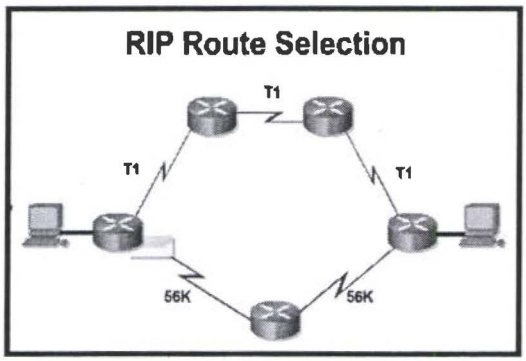
La red casi instantáneamente re-converge con la nueva información contenida en el LSA del router originador.

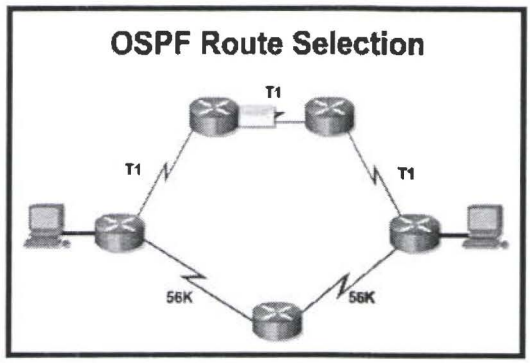


Selección de Ruta

RIP puede tomar rutas subóptimas porque solo considera a los saltos.

OSPF calcula el "Costo" de cada enlace, el cual se basa en el ancho de banda.





Encabezado del Paquete de OSPF (Todos los Tipos)

Un encabezado de 20 bytes es agregado al frente de todos los paquetes de OSPF, conteniendo:

Versión especifica la versión de OSPF; los routers deberán estar corriendo la misma versión o la adyacencia con los vecinos no puede ser establecida.

Tipo especifica el tipo de paquete (Tipo 1, Tipo 2, etc.)

Longitud del Paquete: es la longitud del paquete entero de OSPF en bytes, incluyendo el encabezado estándar del paquete de OSPF.

ID del Router: es la identidad IP del router que está originando el paquete.

ID de Área: es el área de OSPF que el paquete en la que el paquete está siendo enviado.

Autenticación, si es configurada, es especificada.

Encabezado del Paquete Hello (Tipo 1)

Campos adicionales agregados al encabezado del paquete de OSPF para hacer un encabezado de paquete Hello de OSPF incluyen:

Máscara de Red, es el número de bits encendidos en la máscara de Subred usada enviando el ID del Router

Hello Interval, es el número de segundos entre los hellos del router que los envía (10 seg. o 30 seg., dependiendo del tipo de red)

Router Priority, es usada para las elecciones de DR/BDR. Si es puesto en 0, el router que envía no es elegible para llegar a ser Designated Router.

Dead Interval, es el número de segundos antes de que el router que envía considere a un vecino mudo como caído. El predeterminado es 4 veces el Hello

Interval.

Designated Router, es la identidad IP del DR para esta red, desde el punto de vista del router que envía.

Backup Designated Router, es la identidad IP del BDR para esta red, desde el punto de vista del router que envía.

Neighbor Router IDs, son los IDs de cada router de quienes los paquetes Hello han estado recientemente dentro del Dead Interval.

Encabezado de Paquete Hello (Tipo 1)

Network Mask		
Hello Interval	Options	Router Priority
Dead Interval		
Designated Router		
Backup Designated Router		
Neighbor Router ID		
(additional Neighbor Router ID fields, if necessary)		

CAPITULO III

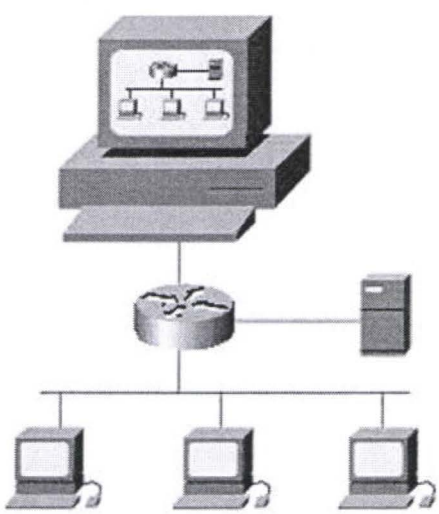
Simple Network Management Protocol

El Simple Network Management Protocol (SNMP) es un protocolo de la capa de aplicación que facilita el intercambio de información de administración entre los dispositivos de red. Es parte de la familia de protocolos TCP/IP. SNMP permite a los administradores supervisar el desempeño de la red, buscar y resolver sus problemas, y planear su crecimiento.

Las versiones de SNMP más utilizadas son dos: SNMP versión 1 (SNMPv1) y SNMP versión 2 (SNMPv2). Ambas versiones tienen un número de características en común, pero SNMPv2 ofrece mejoras, como por ejemplo, operaciones adicionales.

SNMP en su última versión (SNMPv3) posee cambios significativos con relación a sus predecesores, sobre todo en aspectos de seguridad, sin embargo no ha sido mayoritariamente aceptado en la industria.

SNMP facilita el intercambio de información entre dispositivos de red



Los Componentes básicos de SNMP

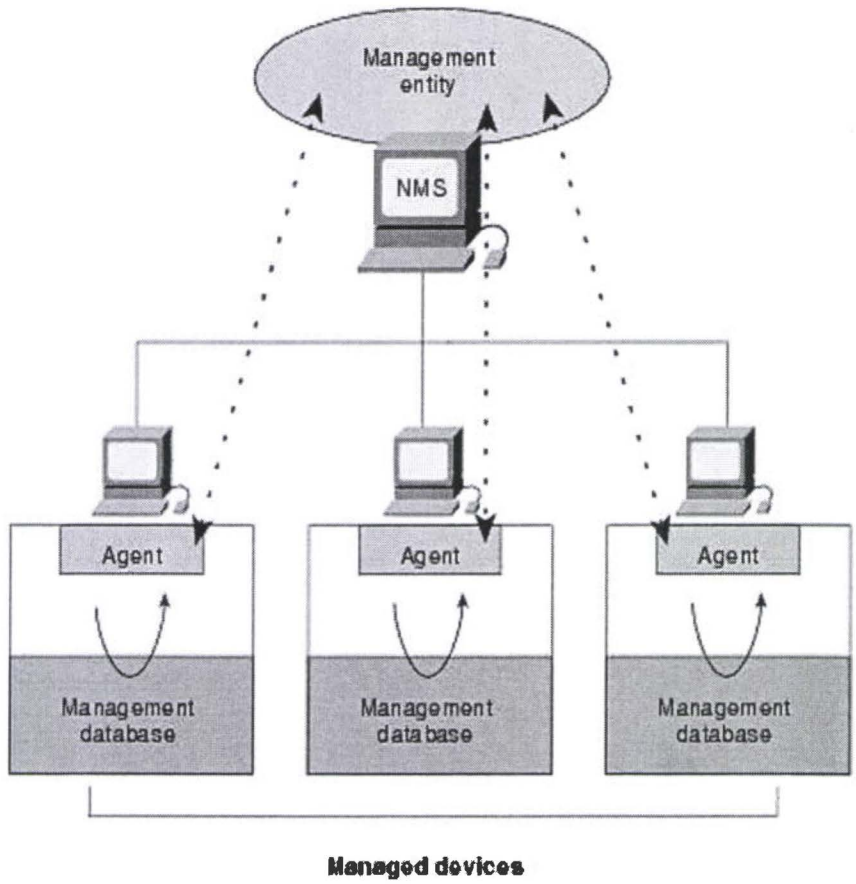
Una red administrada a través de SNMP consiste de tres componentes claves:

- Dispositivos administrados.
- Agentes.
- Sistemas administradores de red (NMS's).

Un dispositivo administrado es un nodo de red que contiene un agente SNMP y reside en una red administrada. Estos recogen y almacenan información de administración, la cual es puesta a disposición de los NMS's usando SNMP. Los dispositivos administrados, a veces llamados elementos de red, pueden ser routers, servidores de acceso, switches, bridges, hubs, computadores o impresoras.

Un agente es un módulo de software de administración de red que reside en un dispositivo administrado. Un agente posee un conocimiento local de información de administración (memoria libre, número de paquetes IP recibidos, rutas, etc), la cual es traducida a un formato compatible con SNMP y organizada en jerarquías.

Un NMS ejecuta aplicaciones que supervisan y controlan a los dispositivos administrados. Los NMS's proporcionan el volumen de recursos de procesamiento y memoria requeridos para la administración de la red. Uno o más NMS's deben existir en cualquier red administrada.



Comandos básicos de SNMP

Los dispositivos administrados son supervisados y controlados usando cuatro comandos SNMP básicos: **lectura**, **escritura**, **notificación** y **operaciones transversales**.

El **comando de lectura** es usado por un NMS para supervisar elementos de red. El NMS examina diferentes variables que son mantenidas por los dispositivos administrados.

El **comando de escritura** es usado por un NMS para controlar elementos de red. El NMS cambia los valores de las variables almacenadas dentro de los dispositivos administrados.

El **comando de notificación** es usado por los dispositivos administrados para reportar eventos en forma asíncrona a un NMS. Cuando cierto tipo de evento

ocurre, un dispositivo administrado envía una notificación al NMS.

Las **operaciones transversales** son usadas por el NMS para determinar qué variables soporta un dispositivo administrado y para recoger secuencialmente información en las tablas de variables, como por ejemplo, una tabla de rutas.

CAPITULO VI

SIMULADOR PACKET TRACER 5.0 DE CISCO

Es un simulador gráfico de redes desarrollado y utilizado por Cisco como herramienta de entrenamiento para obtener la certificación CCNA. Packet Tracer es un simulador de entorno de redes de comunicaciones de fidelidad media, que permite crear topologías de red mediante la selección de los dispositivos y su respectiva ubicación en un área de trabajo, utilizando una interfaz gráfica.

Características generales. Packet Tracer es un simulador que permite realizar el diseño de topologías, la configuración de dispositivos de red, así como la detección y corrección de errores en sistemas de comunicaciones. Ofrece como ventaja adicional el análisis de cada proceso que se ejecuta en el programa de acuerdo a la capa de modelo OSI que interviene en dicho proceso; razón por la cual es una herramienta de gran ayuda en el estudio y aprendizaje del funcionamiento y configuración de redes de comunicaciones y aplicaciones

Requerimientos del sistema. Para una correcta instalación y posterior uso del software de PACKET TRACER.

Requerimientos básicos para la instalación de Packet Tracer

Sistema operativo

Microsoft Windows 98,

ME, 2000, o XP y

Macintosh

Requerimientos Mínimos

Procesador Intel Pentium, 200 MHz o equivalente, 64MB RAM.

Espacio disponible en D.D. 30 MB

Macro media Flash Placer 6.0 o superior

Recomendaciones

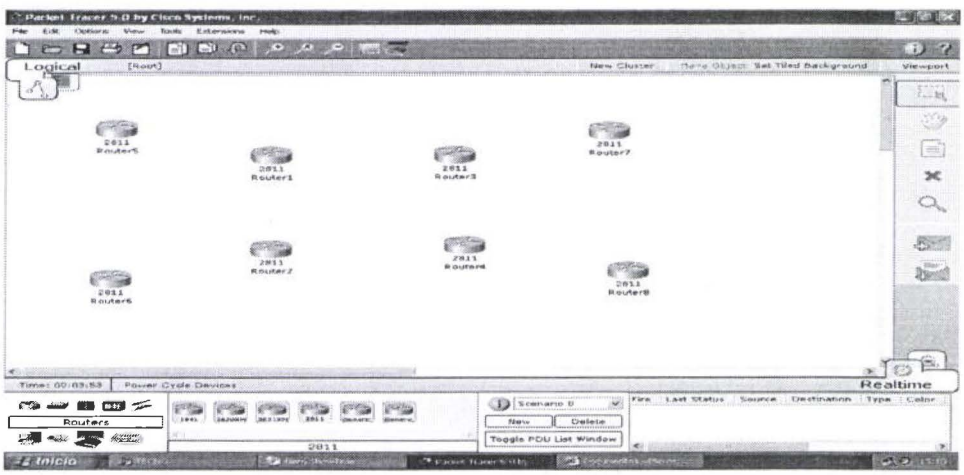
Tarjeta de sonido.

Parlantes.

ESQUEMA DE SIMULACIÓN MEDIANTE EL SOFTWARE PACKET TRACER 5.0 de Cisco

1.1

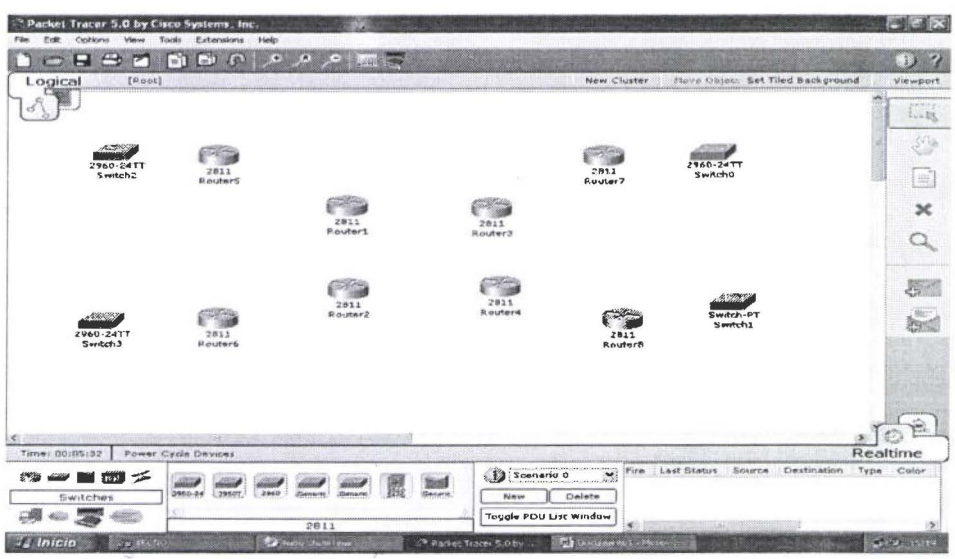
Mediante el Software Packet Tracer 5.0. se realizará un esquema simulado de enlaces Lan y Wan, donde estará levantado el protocolo de enrutamiento RIP ver2. Como se puede ver en el grafico se escogió los routers que van a hacer el enlace Wan, el modelo de los router es el 2811 de cisco. Los cuatro routers del centro representaran el backbone de una empresa con sus distintas ciudades, y los otros cuatro routers representaran las sucursales que se encuentran a grandes distancias.



1.2

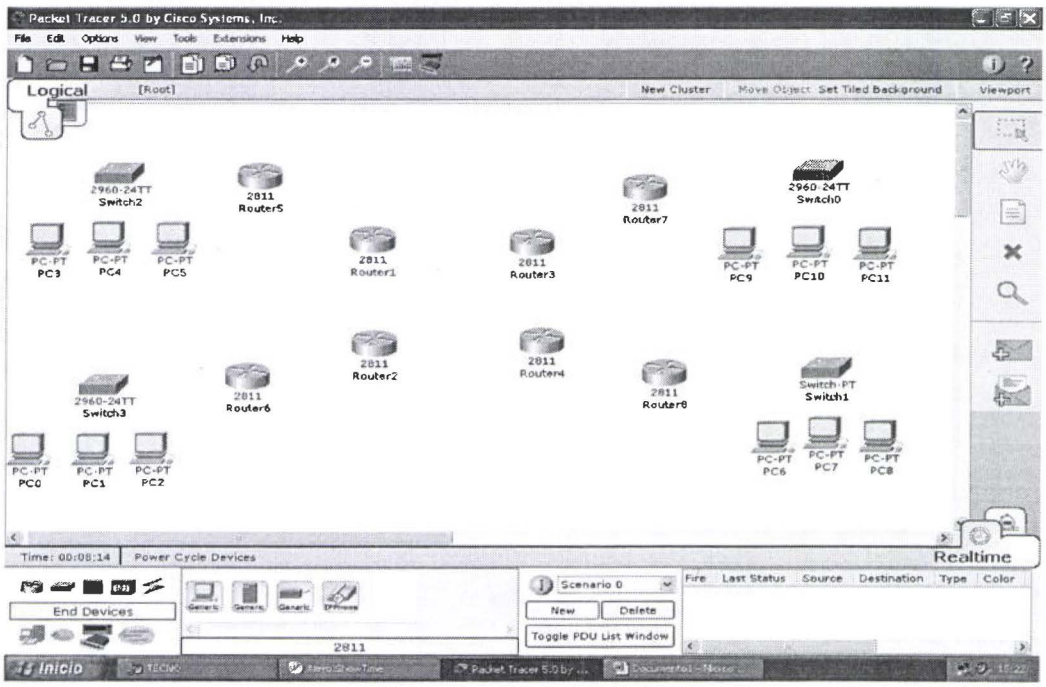
En el grafico 2 se puede observar cuatro switches, los mismos que tendrán la función de hacer la red Lan interna de las diferentes sucursales.

El modelo de los cuatro switches es el 2960-24TT.



1.3

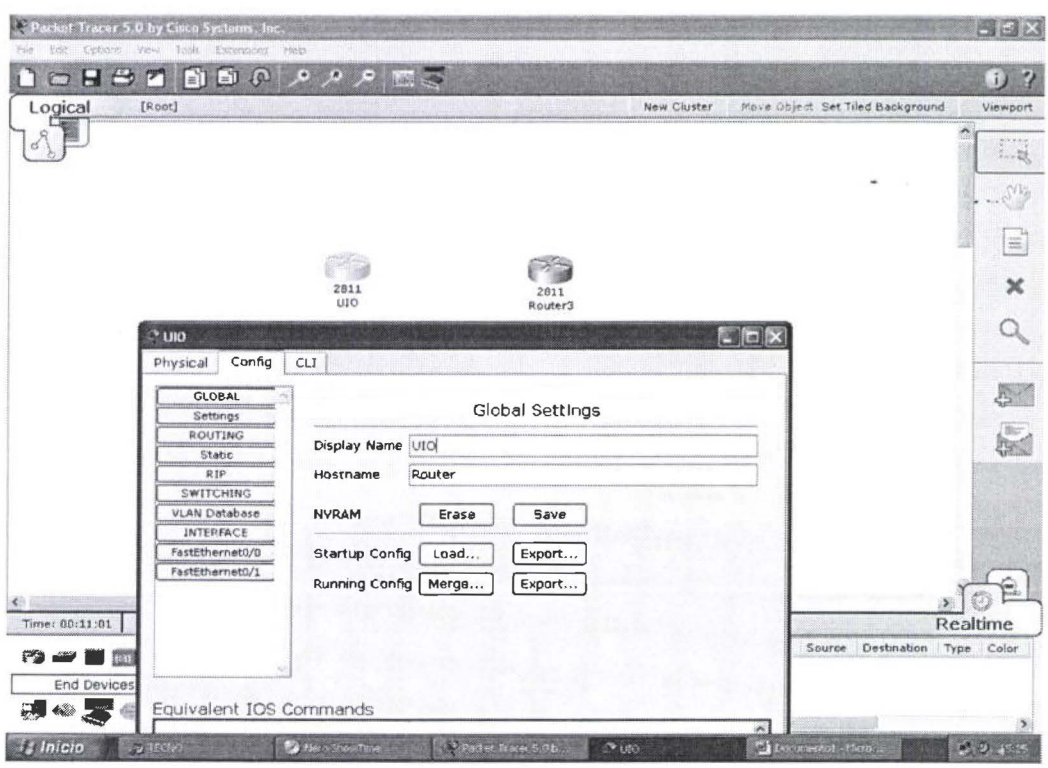
En el grafico 3, se observa los host que representarán a los usuarios finales.



1.4

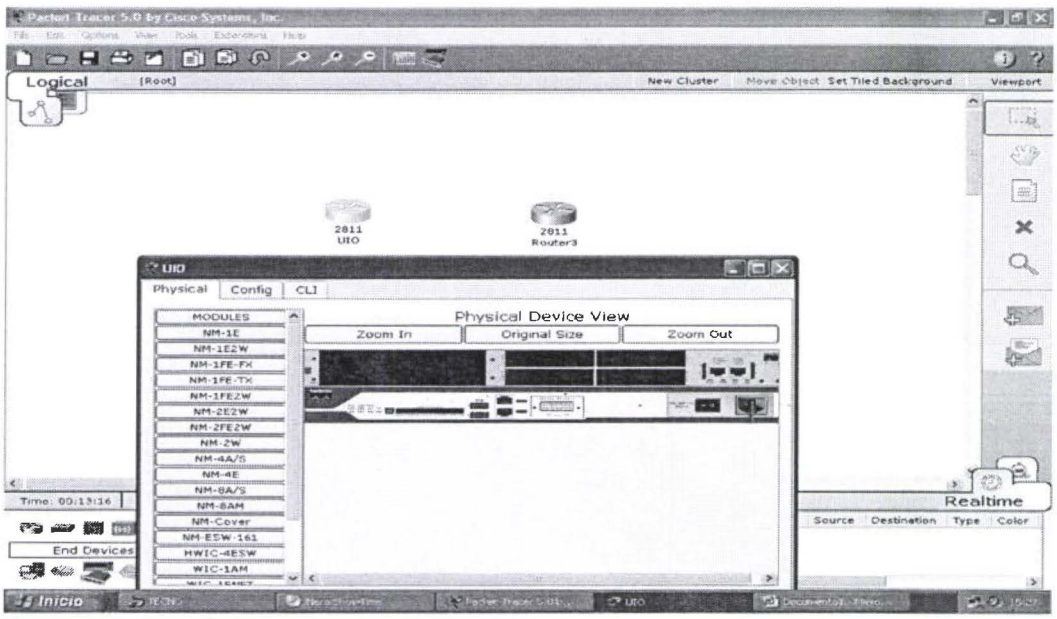
Una vez escogido el tipo de router se procederá a indetificarles a cada uno con los nombres de las diferentes ciudades

Ejemplo: QUITO (UIO)



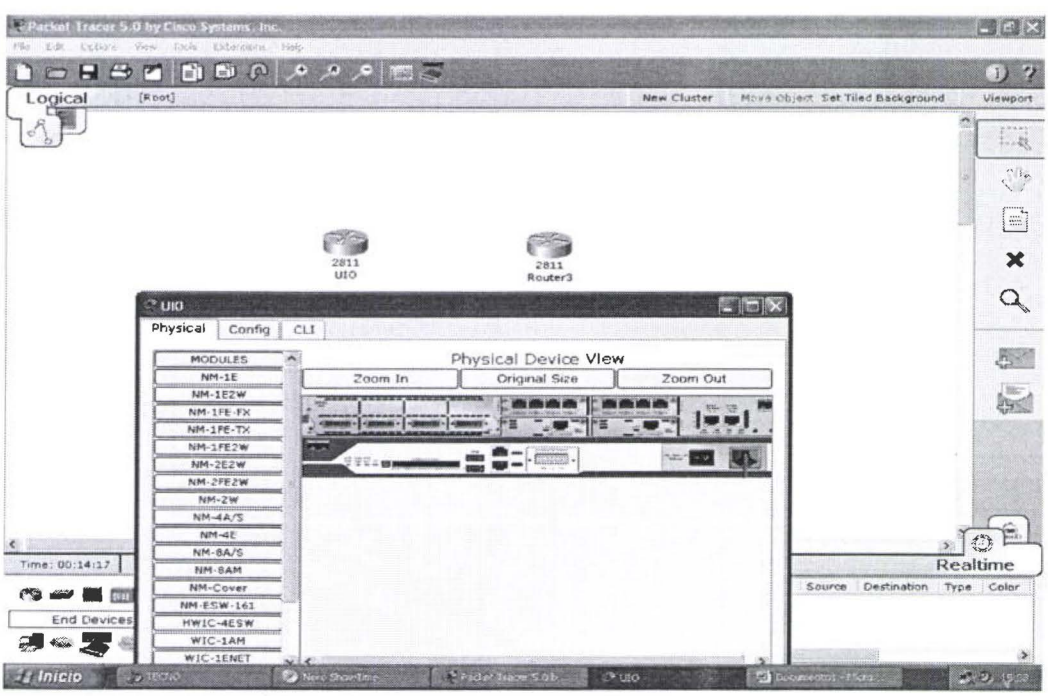
1.5

Una vez asignados los nombres en cada uno de los cuatro routers, se procede a insertar las tarjetas respectivas para la conexión WAN. Estas y otras tarjetas se encuentran en la parte izquierda.



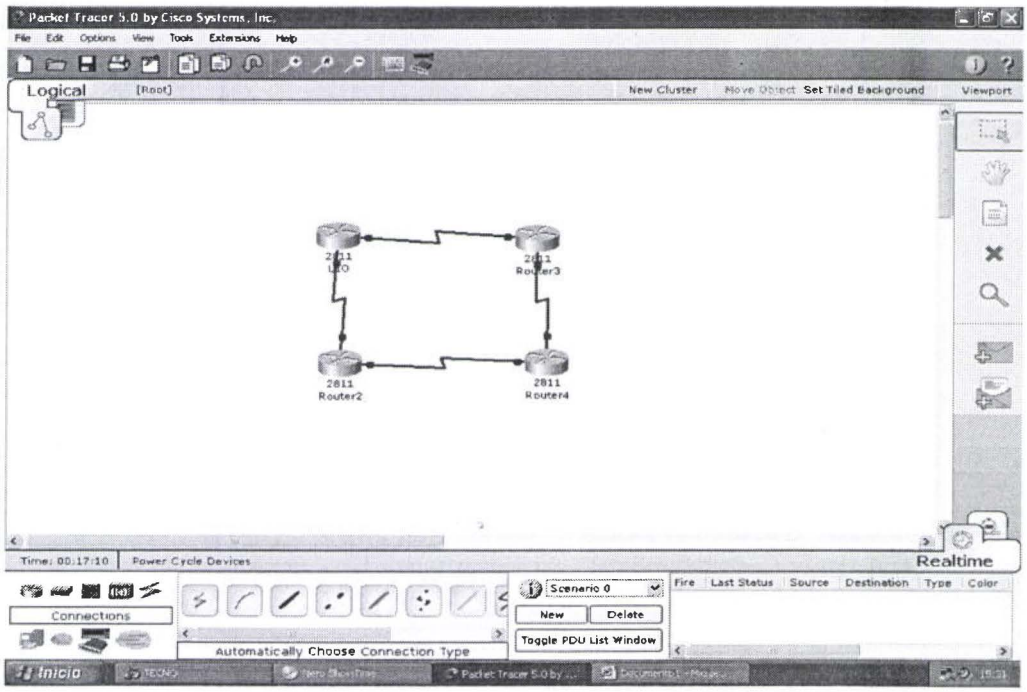
1.6

Se escogerá la tarjeta **(NM-4A/S)** que tiene cuatro interfaces seriales, además dos tarjetas **(WIC-1ENET)** que tienen una interfase Fast ethernet de 10 Mbps, y dos tarjetas **(HWIC-4ESW)** que tiene cada uno cuatro puertos ethernet. Estas tarjetas se utilizarán acorde a las necesidades requeridas.



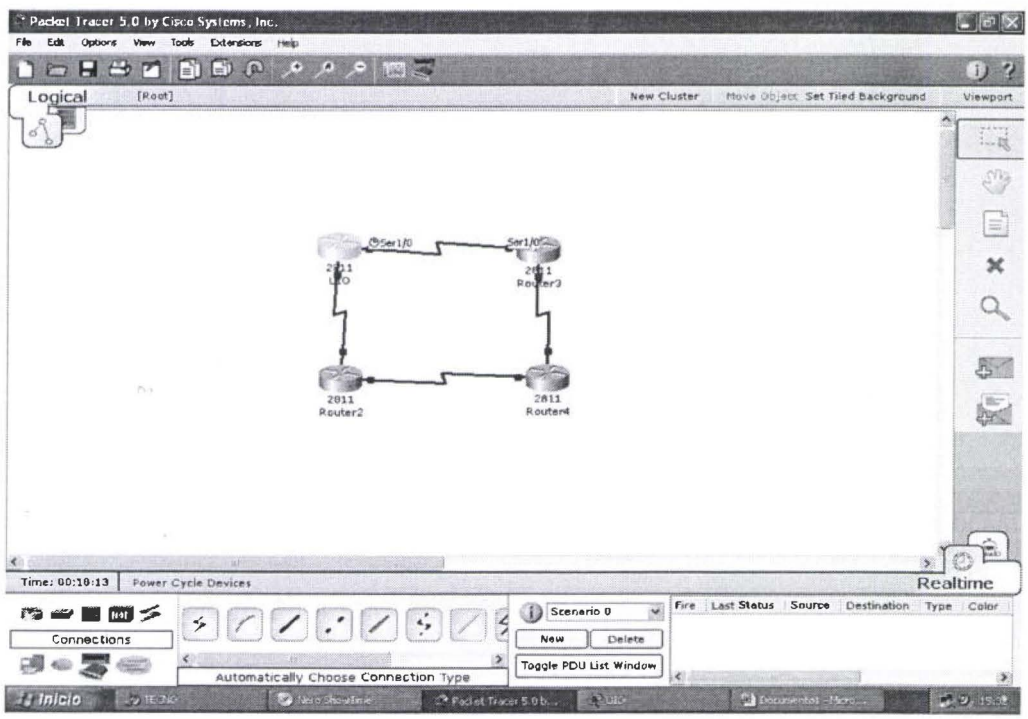
1.7

Una vez terminado de poner las tarjetas, se procederá a realizar las conexiones entre los cuatro routers con sus interfaces seriales de cada uno.



1.8

Como se puede observar en la gráfica se tiene identificado las interfaces seriales que se encuentran conectadas, y además nos indicará cual es el DCE Y DTE de cada router.



CONFIGURACION DE LAS INTERFACES SERIALES 1/0

2811

ROUTER A (UIO)

Router>en

Router#configure ter

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

Router(config)#hostname UIO

UIO(config)#interface serial 1/0

UIO(config-if)#ip add 130.10.62.1 255.255.255.0

UIO(config-if)#no shut

%LINK-5-CHANGED: Interface Serial1/0, changed state to down

UIO(config-if)#

%SYS-5-CONFIG_I: Configured from console by console

UIO#

2811
GUAYAQUIL

Router>en

Router#configure ter

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

Router(config)#hostname GYE

GYE(config)#interface serial 1/0

GYE(config-if)#ip address 130.10.62.2 255.255.255.0

GYE(config-if)#no shut

%LINK-5-CHANGED: Interface Serial1/0, changed state to up

GYE(config-if)#

Como se observa después de haber configurado los dos Router´s (UIO y GYE), sus interfaces seriales 1/0 están levantadas.

2811
CUENCA

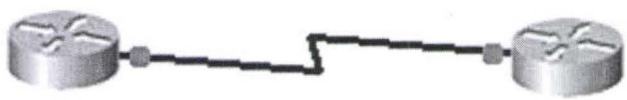
Router>en
Router#configure ter
Enter configuration commands, one per line. End with CNTL/Z.
UE(config)#hostname CUE
CUE(config)#interface serial 1/0
CUE(config-if)#ip address 130.10.65.8 255.255.255.0
CUE(config-if)#no shut
%LINK-5-CHANGED: Interface Serial1/0, changed state to down
CUE(config-if)#

2811
AMBATO

Router>en
Router#configure ter
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname AMB
AMB(config)#interface serial 1/0
AMB(config-if)#ip address 130.10.65.9 255.255.255.0
AMB(config-if)#no shut

1.10

De la misma manera las interfaces seriales 1/0 de los router´s (CUE y AMB) se encuentran levantadas.



CONFIGURACION DE INTERFACES SERIALES 1/1

UIO>en

UIO#configure ter

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

UIO(config)#interface serial 1/1

UIO(config-if)#ip address 130.10.63.1 255.255.255.0

UIO(config-if)#no shut

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

UIO(config-if)#

CUE>en

CUE#configure ter

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

CUE(config)#interface serial 1/1

CUE(config-if)#ip address 130.10.63.3 255.255.255.0

CUE(config-if)#no shut

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

CUE(config-if)#

GYE>en

GYE#configure ter

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

GYE(config)#interface serial 1/1

GYE(config-if)#ip address 130.10.64.2 255.255.255.0

GYE(config-if)#no shut

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

GYE(config-if)#

AMB>en

AMB#configure ter

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

AMB(config)#interface serial 1/1

AMB(config-if)#ip address 130.10.64.3 255.255.255.0

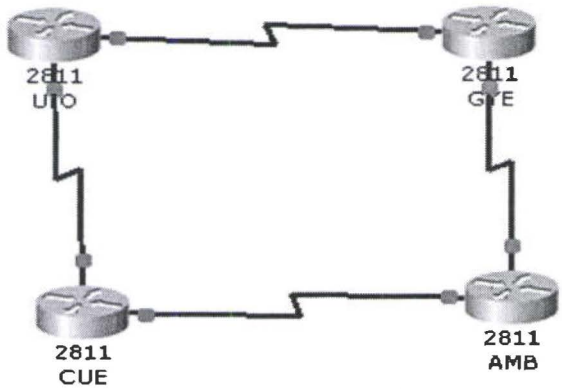
AMB(config-if)#no shut

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

AMB(config-if)#

1.11

Por último se encuentran levantadas todas las interfaces seriales de cada uno de los cuatro router´s.



CONFIGURACION DEL PROTOCOLO DE ENRUTAMINETO RIP ver2

Realizadas las respectivas configuraciones en los cuatro router´s de cada interface serial, estos se encuentran operativos; pero con la aclaración de que solo se están comunicando fronterizamente; aún no se están viendo (comunicándose entre los cuatro).

Para llevar a efecto la comunicación entre los cuatro router´s, se comenzará a configurar el protocolo de enrutamiento RIP ver2, el cual será el backbone de la red WAN.

ROUTER A (UIO)

UIO>en

UIO#configure ter

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

UIO(config)**#router rip**

UIO(config-router)**#ver 2**

UIO(config-router)**#network 130.10.0.0**

UIO(config-router)#^Z

%SYS-5-CONFIG_I: Configured from console by console

Para tener respuesta de que el router GYE se encuentra en la red, y además verificar si el protocolo de enrutamiento RIP ver2 se encuentra levantado, se realizará un ping hacia el router antes mencionado.

UIO#ping 130.10.62.2

Sending 5, 100-byte ICMP Echos to 130.10.62.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms

Una vez configurado el protocolo RIP ver2, y además haciendo que trabajen una sola Red General con la dirección IP 130.10.0.0, entonces se podrán comunicar entre si, e ir actualizando su tabla de ruteo.

Para su comprobación se realizará un ping desde el router UIO hacia los tres router´s restantes, dando la confirmación que se encuentran en total funcionamiento.

ROUTER QUITO INTERFACE SERIAL 1/0

UIO>en

UIO#ping 130.10.62.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.62.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 62/65/79 ms

ROUTER QUITO INTERFACE SERIAL 1/1

UIO#ping 130.10.63.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.63.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 62/62/63 ms

ROUTER GYE INTERFACE SERIAL 1/0

UIO#ping 130.10.62.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.62.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms

ROUTER GYE INTERFACE SERIAL 1/1

UIO#ping 130.10.64.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.64.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms

ROUTER CUENCA INTERFACE SERIAL 1/1

UIO#ping 130.10.63.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.63.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms

ROUTER CUENCA INTERFACE SERIAL 1/0

UIO#ping 130.10.65.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.65.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 16/28/31 ms

ROUTER AMBATO INTERFACE SERIAL 1/1

UIO#ping 130.10.64.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.64.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 47/59/63 ms

ROUTER AMBATO INTERFACE SERIAL 1/0

UIO#ping 130.10.65.9

Type escape sequence to abort.

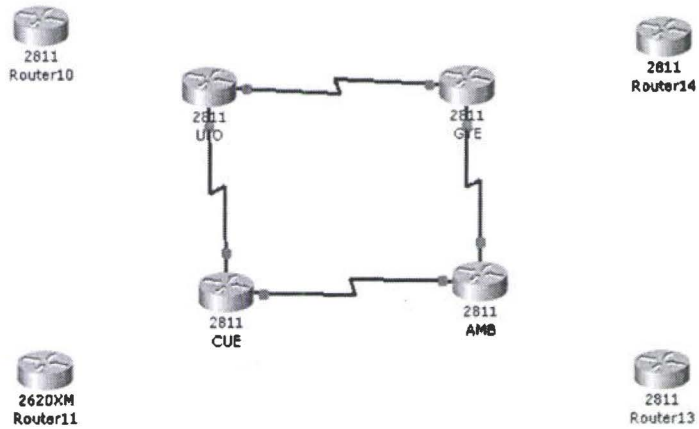
Sending 5, 100-byte ICMP Echos to 130.10.65.9, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 62/62/63 ms.

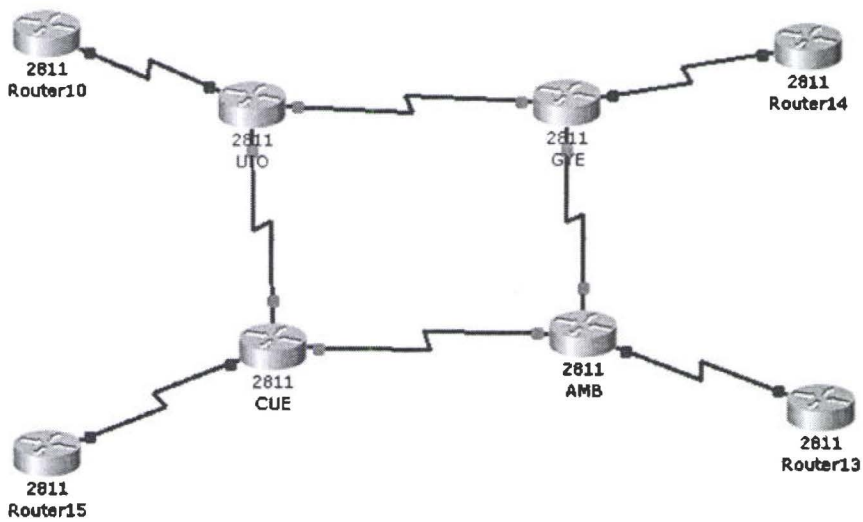
1.12

Ya realizada la parte operativa de la red WAN, ahora se hará la implementación de los cuatro router´s siguientes, los mismo que van realizar la función de sucursales de las ciudades antes mencionadas y la fusión entre la red WAN y la red LAN.



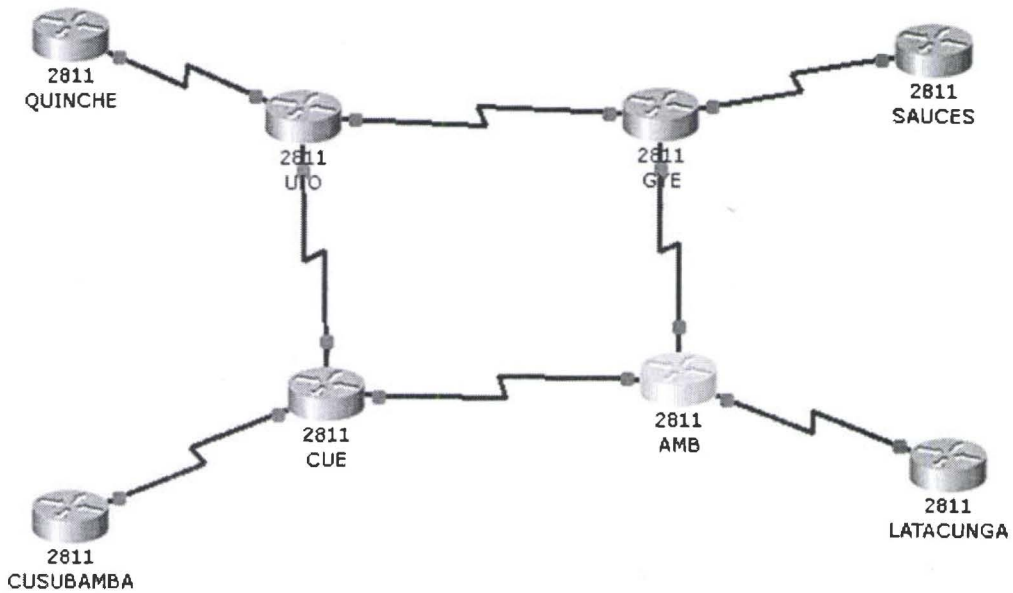
1.13

Estos cuatro router´s representan las sucursales de cada ciudad. De igual manera se procederá a realizar las conexiones y sus configuraciones.



1.14

Configuradas sus interfaces seriales de la misma forma que se realizó antes, y además configurado el protocolo RIP ver2, se tendrá en funcionalidad la red diseñada.



Ejemplo:

CUSU>en

CUSU#configure ter

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

CUSU(config)#interface serial 1/0

CUSU(config-if)#ip add 130.10.24.3 255.255.255.0

CUSU(config-if)#clock rate 800000

CUSU(config-if)#

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up

CUSU(config-if)#exit

CUSU(config)#**router rip**

CUSU(config-router)#**ver 2**

CUSU(config-router)#**redistribute connected**

CUSU(config-router)#**network 130.10.0.0**

CUSU(config-router)#^Z

De igual manera se verifica la funcionalidad entre los router´s de cada sucursal, y los router´s de Backbone, realizando un ping desde **130.10.24.3 (CUENCA-CUSUBAMBA)** hacia los demás router´s

ROUTER SAUCES

CUSU#ping 130.10.16.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.16.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 97/110/125 ms

ROUTER LATACUNGA

CUSU#ping 130.10.17.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.17.3, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 93/96/109 ms

ROUTER QUINCHE

CUSU#ping 130.10.8.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.8.1, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 80/91/94 ms

ROUTER DE BACKBONE (GYE)

CUSU#ping 130.10.62.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 130.10.62.2, timeout is 2 seconds:

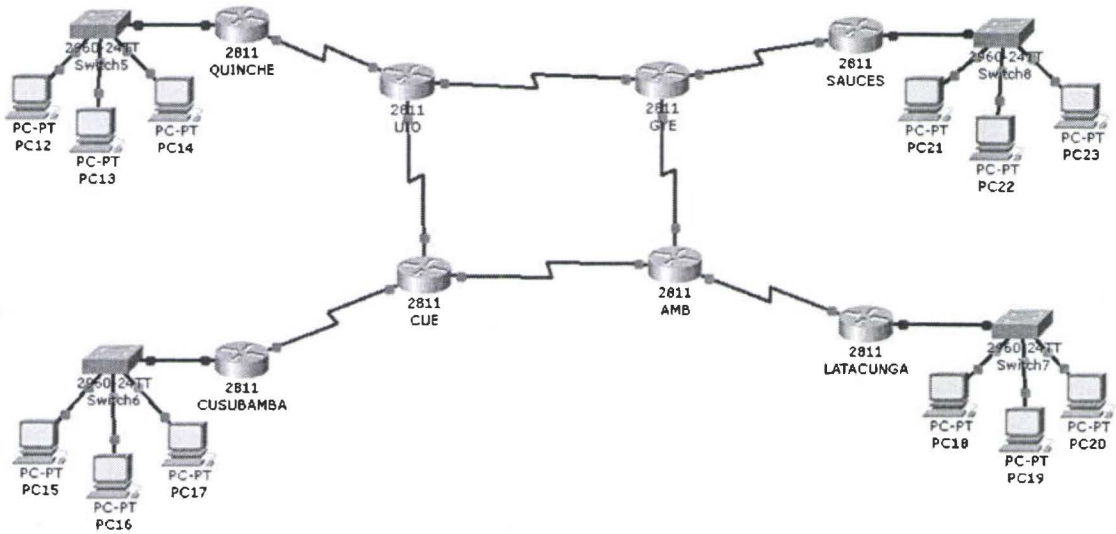
!!!!

CONFIGURANDO SWITCHES CISCO 2960

Después de haber acabado con las configuraciones respectivas en cada uno de los router´s, se continuará con los switche´s a los que estarán conectadas tres Pc´s para la función de la red LAN.

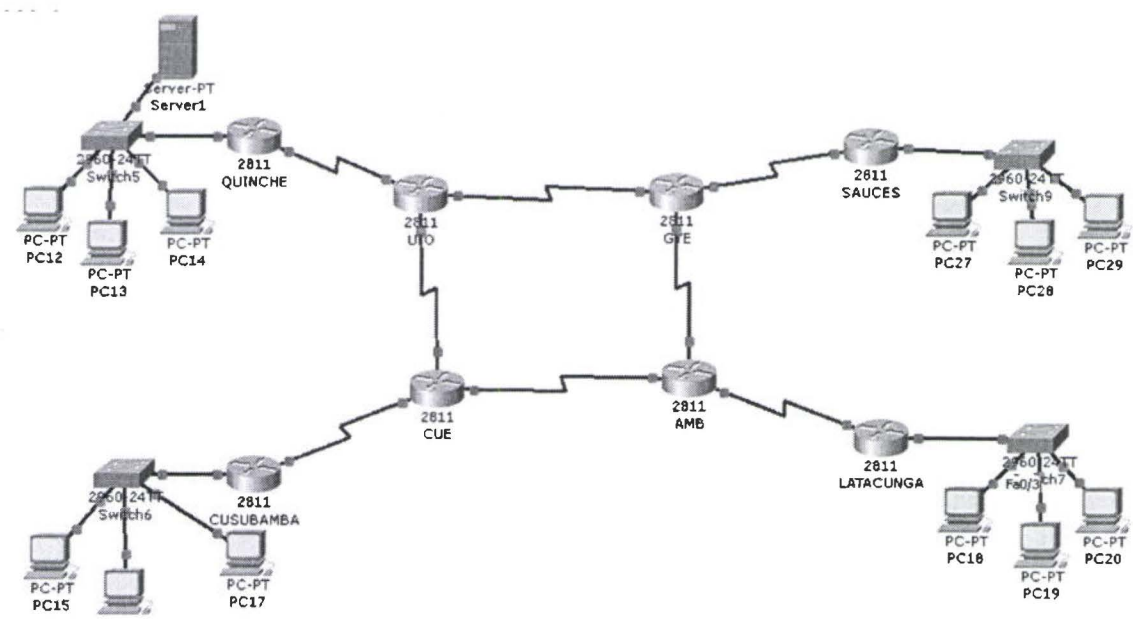
1.15

Por último está diseñado el esquema completo de las redes LAN y WAN, pero aún no tienen comunicación las Pc´s con los Router´s.



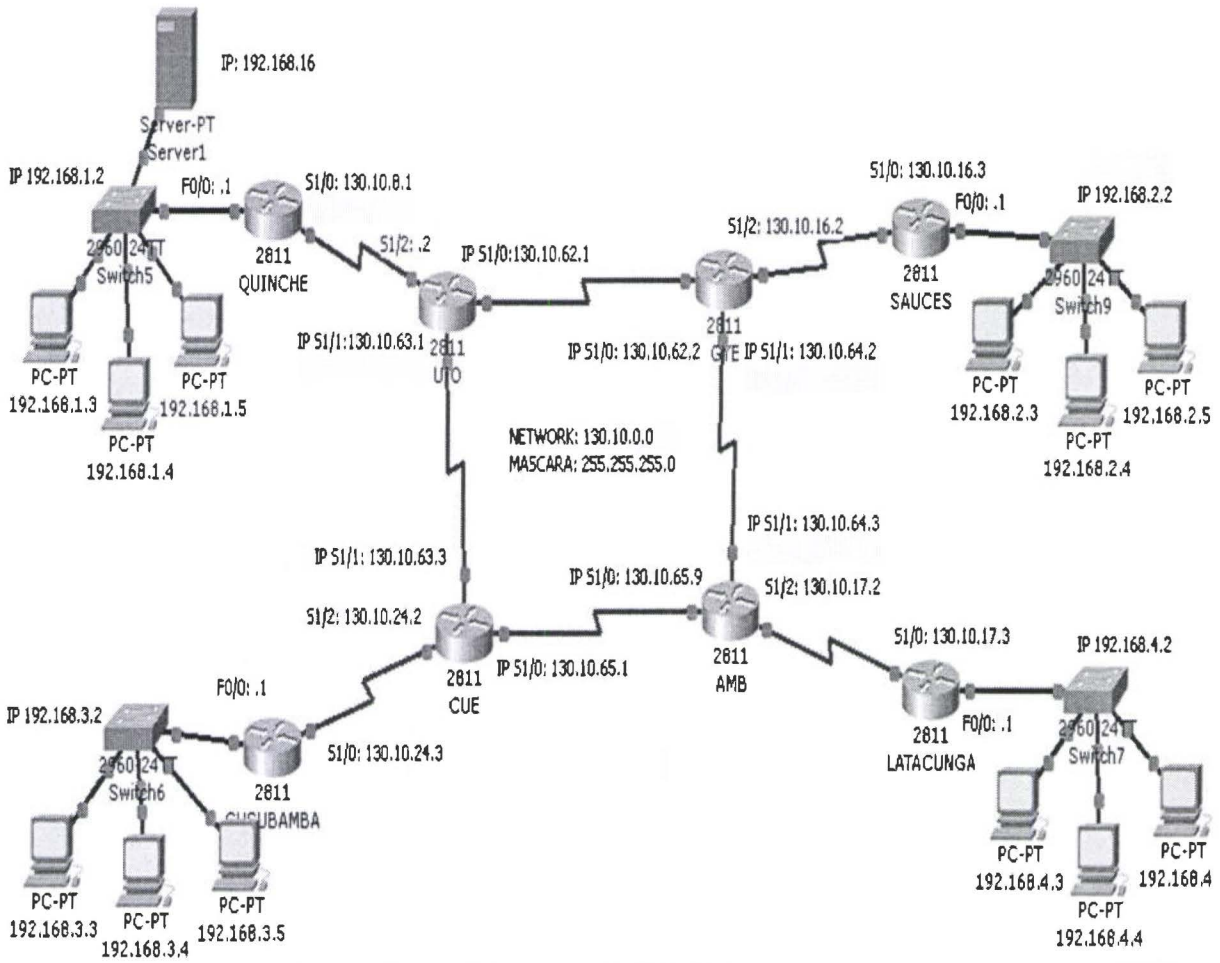
1.16

Terminadas las configuraciones respectivas de cada uno de los equipos, tendremos el esquema funcionando operativamente con todos sus protocolos, mas un servidor WEP.



1.17

Sus direcciones IP de cada elemento activo:



ANALISIS DEL PROTOCOLO DE ENRUTAMIENTO RIP ver2 CON REAL TIME DEL PACKET TRACER 5.0

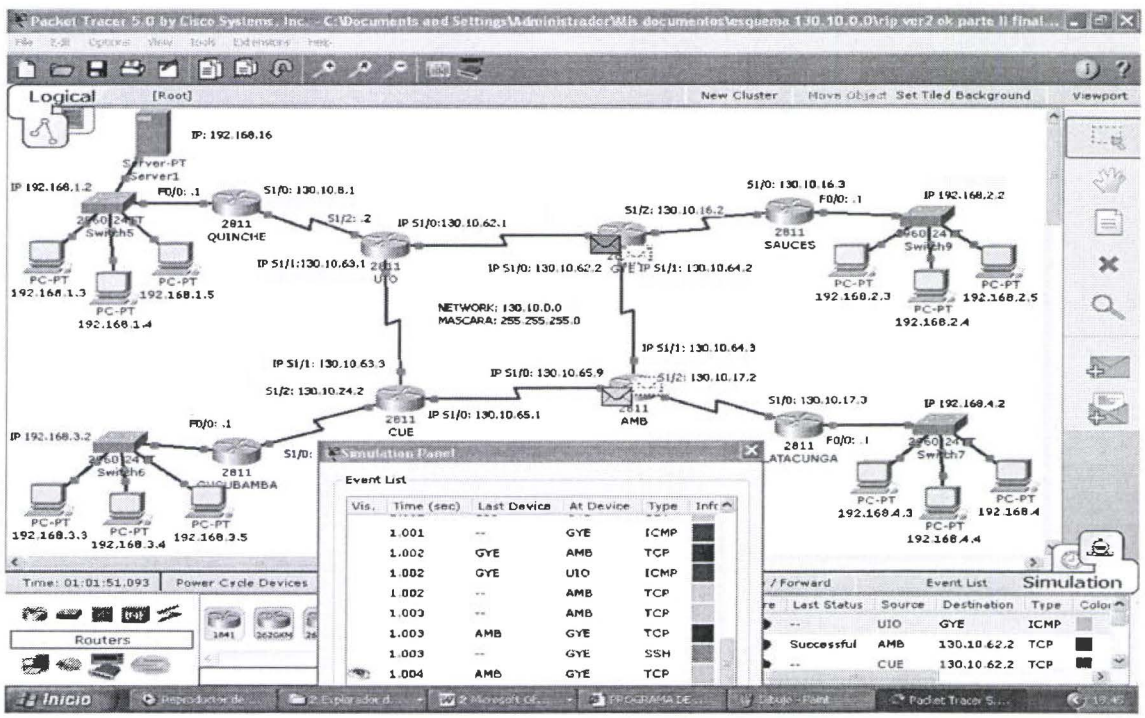
Como primera instancia se observa en el cuadro de Simulation Panel el protocolo ICMP, esto significa que el paquete PDU está recorriendo hacia todos los routers e indicándonos que todas las interfaces estén conectadas, y además este protocolo sirve como si se estuviera realizando un ping entre los dispositivos de la red.

Después que hubiese recorrido por todos los dispositivos de la red, indica que se encuentran operativas las interfaces y además que los usuarios finales se

encuentran comunicándose entre si.

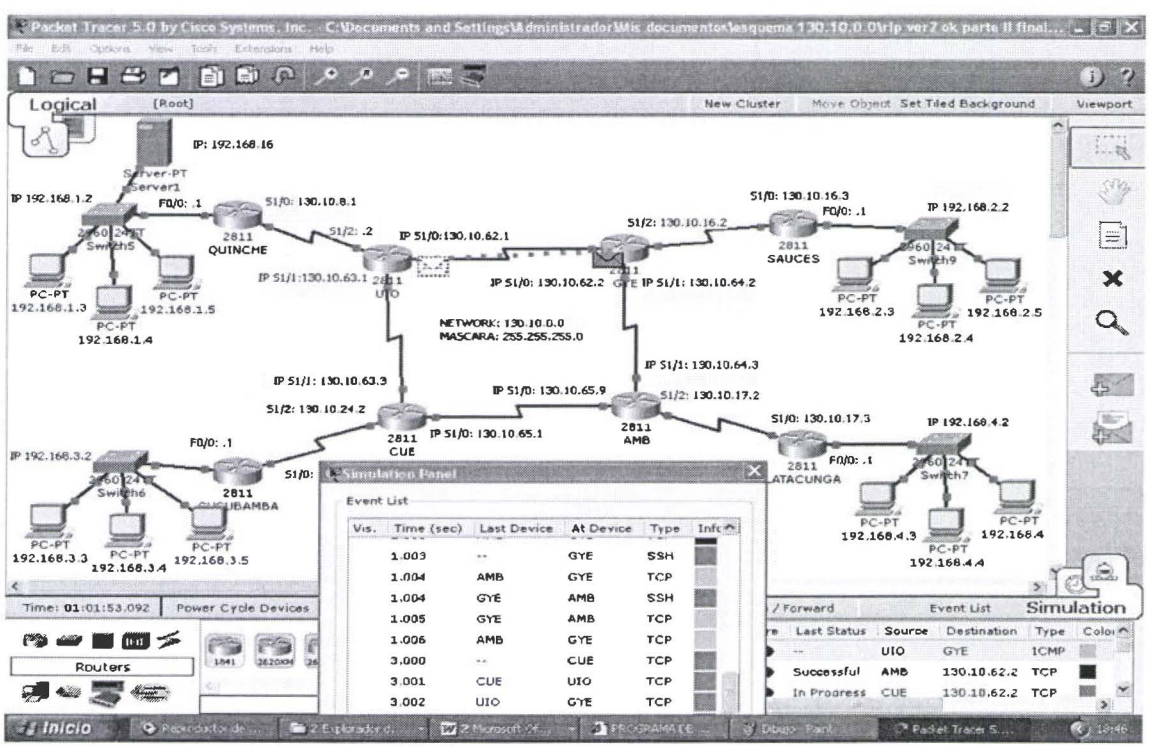
Como se puede observar en el gráfico en la parte izquierda nos da como resultado que el ICMP fue exitoso (Successful)

1.18



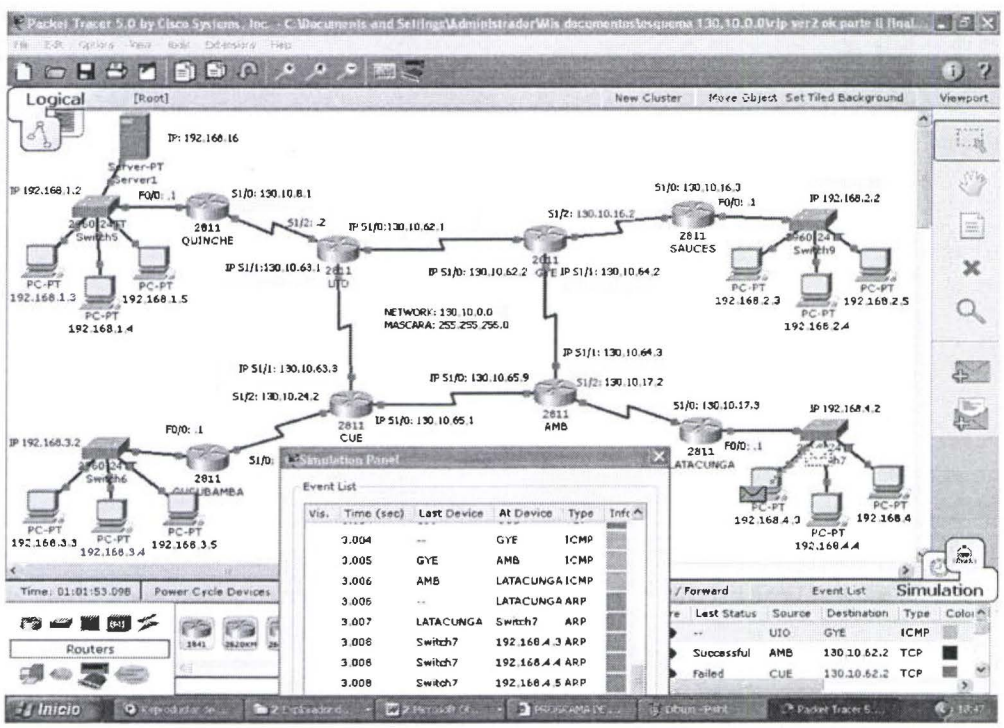
1.19

Se puede observar el protocolo TCP, este protocolo es orientado a conexión.



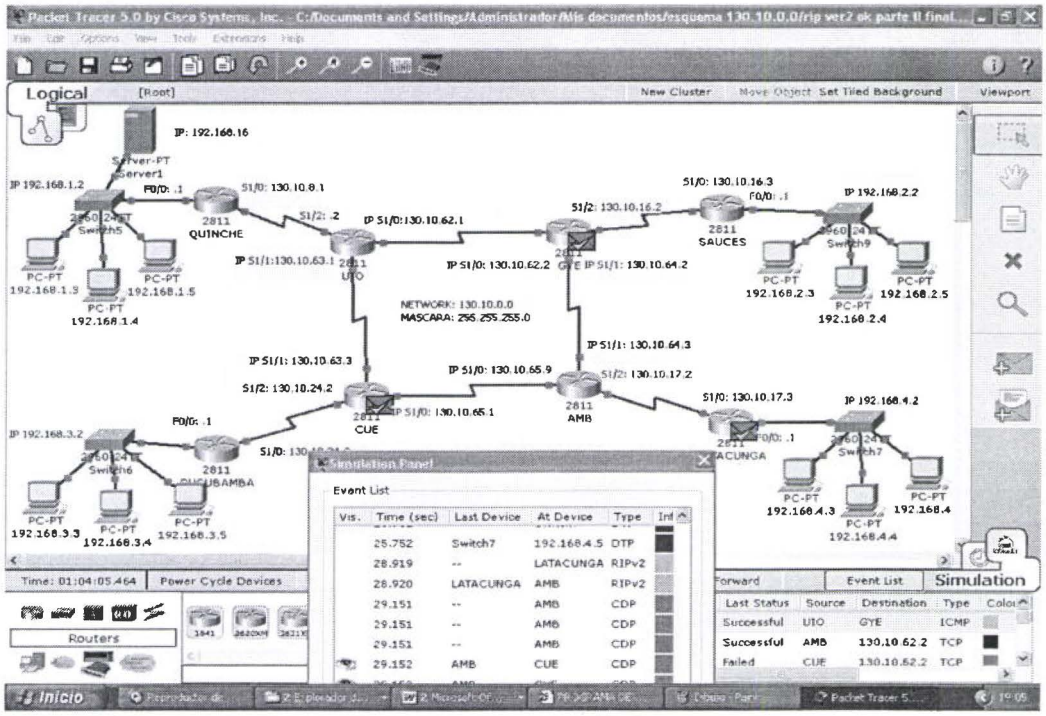
De la misma manera se observará que también esta corriendo el protocolo ARP (Address Resolution Protocol), que es el encargado de convertir las direcciones IP en direcciones de la red física.

Cuando una máquina desea enviar un mensaje a otra máquina que está conectada a través de una red Fastethernet se encuentra con un problema: la dirección IP de la máquina en cuestión es diferente a la dirección física de la misma. La máquina que quiere enviar el mensaje sólo conoce la dirección IP del destino, por lo que tendrá que encontrar un modo de traducir la dirección IP a la dirección física. Esto se hace con el protocolo ARP.



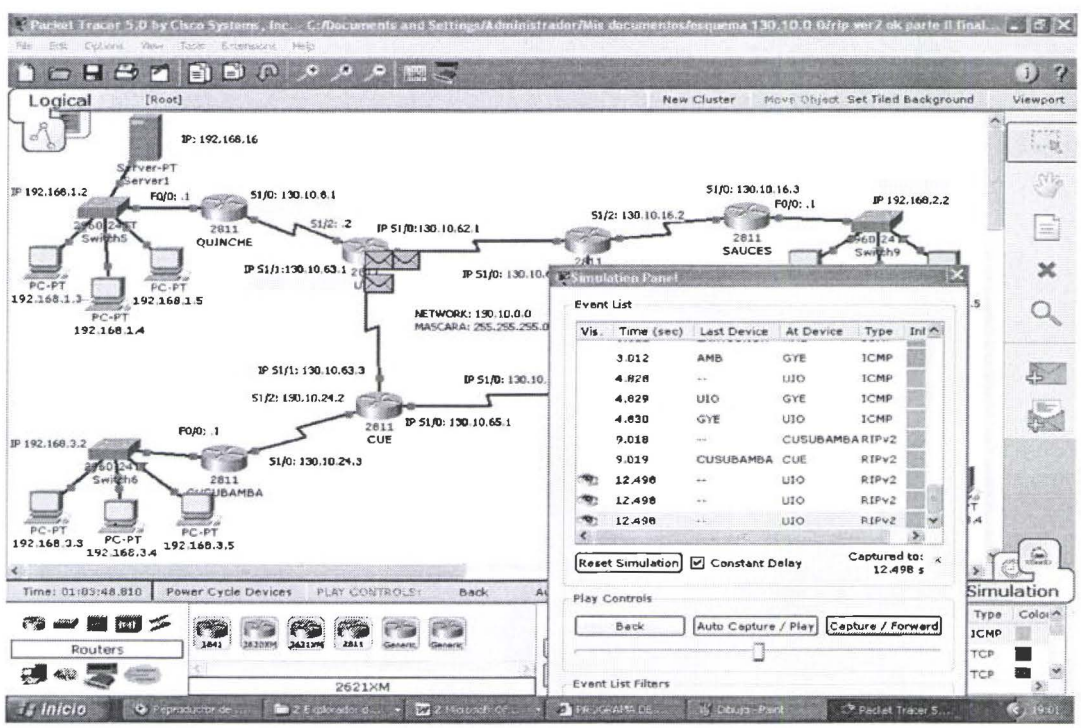
1.20

El protocolo CDP (**C**isco **D**iscovery **P**rotocol), es un protocolo de red propietario de nivel 2, desarrollado por Cisco Systems y usado en la mayoría de sus equipos. Es utilizado para compartir información sobre otros equipos Cisco directamente conectados, tal como la versión del sistema operativo y la dirección IP. CDP también puede ser usado para realizar encaminamiento bajo demanda (ODR, *On-Demand Routing*), que es un método para incluir información de encaminamiento en anuncios CDP, de forma que los protocolos de encaminamiento dinámico no necesiten ser usados en redes simples



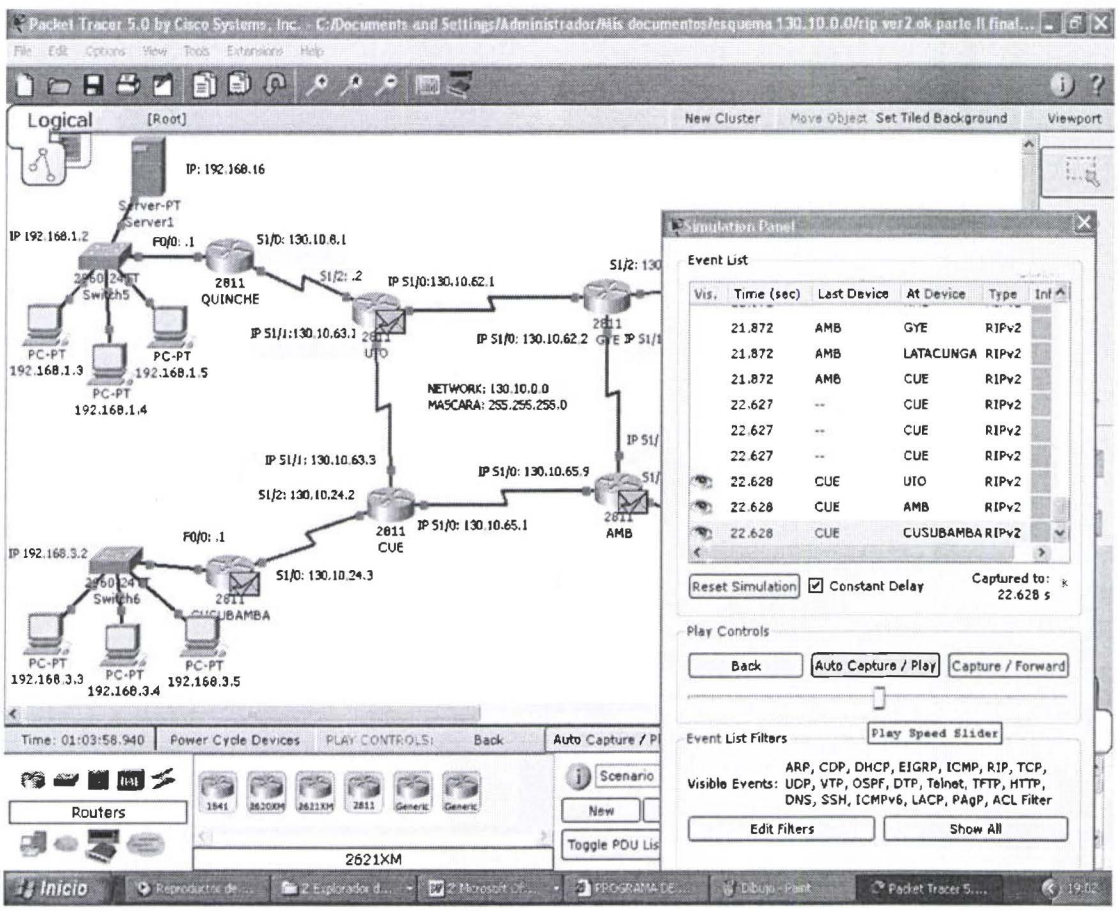
1.21

Una vez que todos los protocolos antes mencionados hayan pasado, se tiene el protocolo de enrutamiento RIP ver2. Como se puede observar, se va a enviar el paquete PDU con el protocolo RIP ver2 a sus tres routers vecinos: El Quinche, Guayaquil y Cuenca, y se verificará que se están comunicando y actualizando sus tablas de enrutamiento.



1.22

Se obtiene las respuestas de los routers, los cuales se están comunicando con el protocolo RIP ver2, demostrando que los mensajes fueron recibidos por los anteriores routers sin perdida de datos.



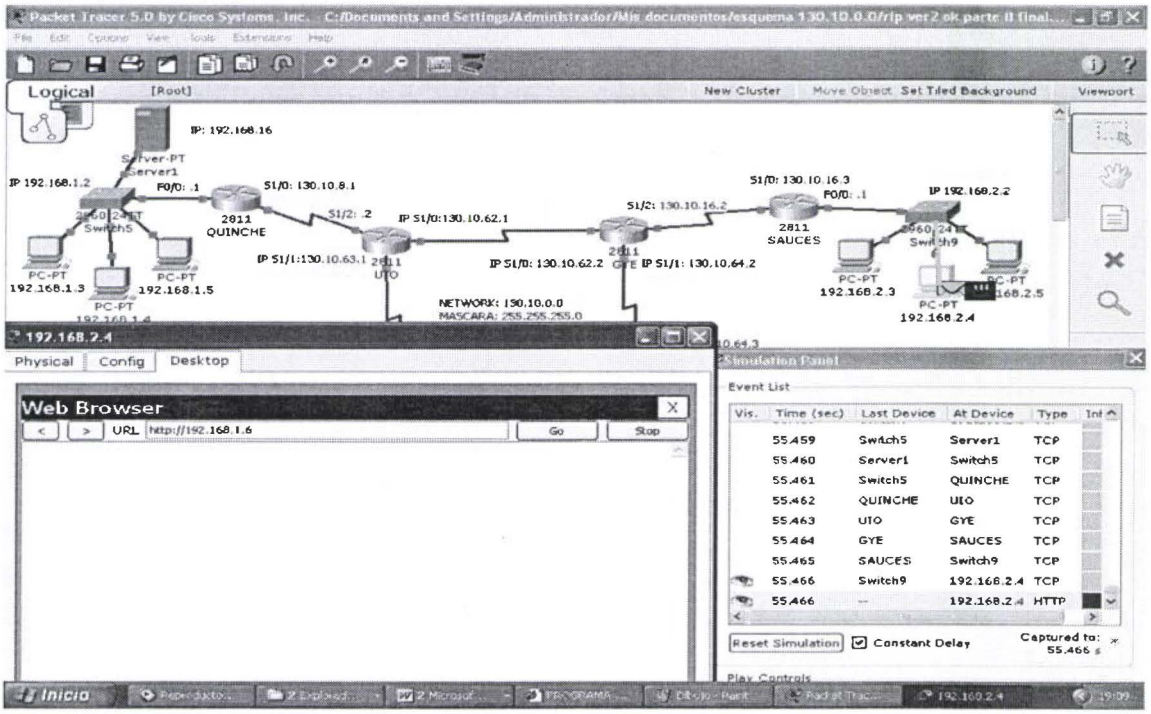
DEMOSTRACION de la CAPA 7 DE APLICACIÓN DEL MODELO OSI

1.23

En esta gráfica se realizará una petición al servidor WEP, el cual se encuentra en la sucursal El Quinche de la empresa en Quito, esta petición es realizada desde la sucursal los Sauces de la empresa Guayaquil por el usuario (192.168.2.4).

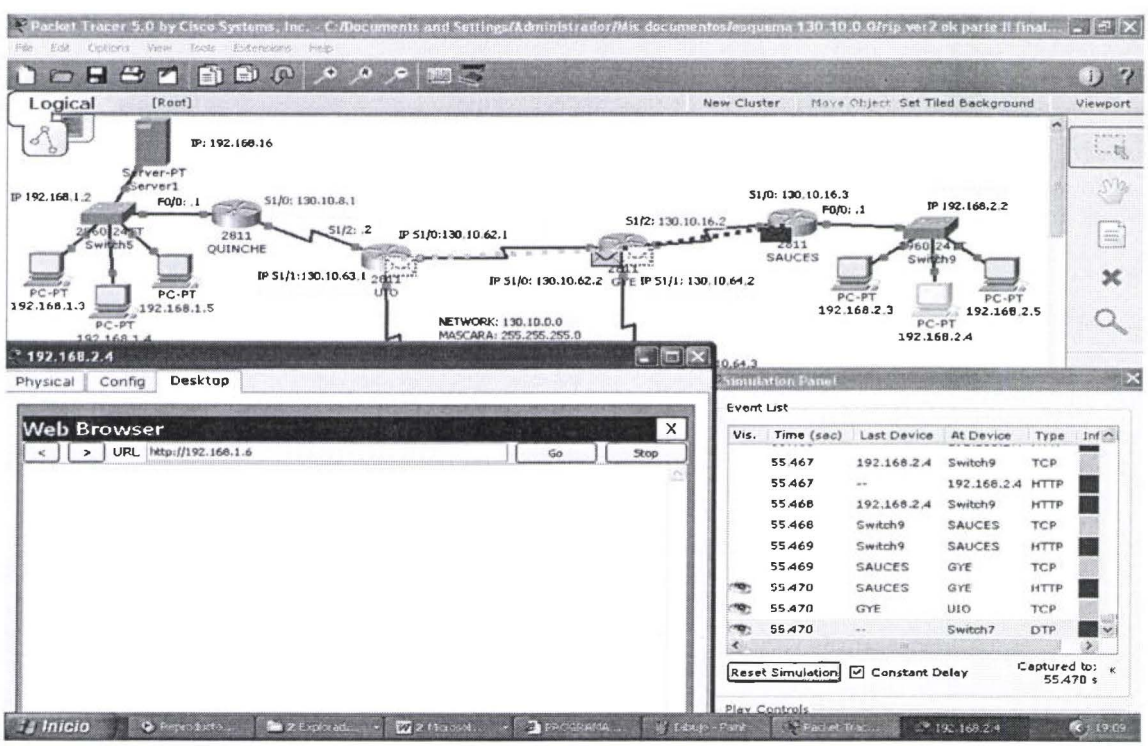
Se observará el protocolo http siendo enviado hacia el servidor WEP, el cual tiene la dirección 192.168.1.6. Como se podrá observar, para esta petición

corren dos protocolos (http y tp), los cuales se encargara de encaminar la solicitud requerida.



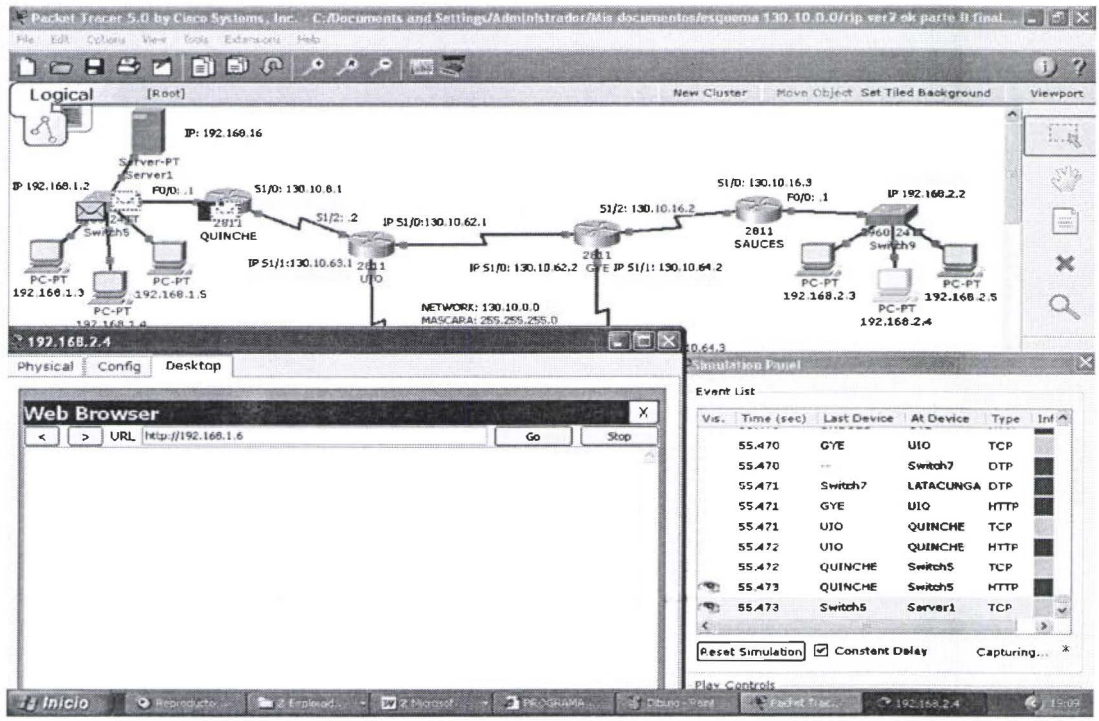
1.24

Una vez realizada la petición hacia el servidor WEP, el mensaje comienza a recorrer hacia su destino final. El protocolo que se encarga de encaminar el mensaje es http mediante tcp



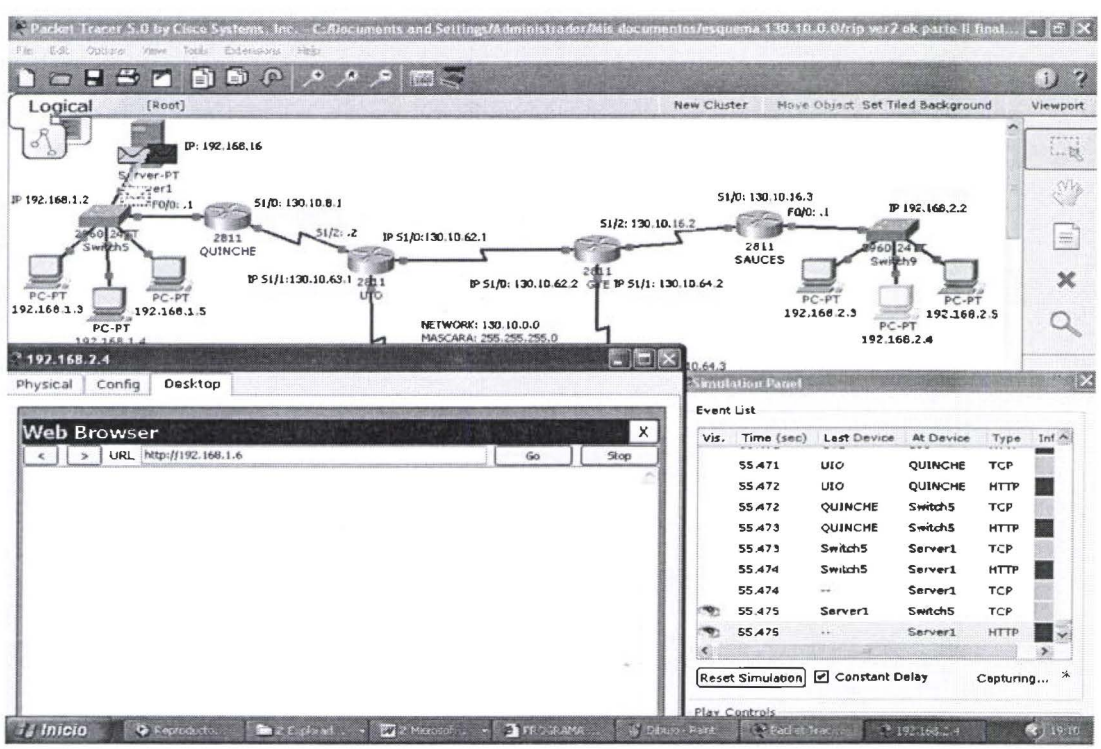
1.25

El paquete llega al servidor WEP y escucha la petición requerida del usuario.



1.26

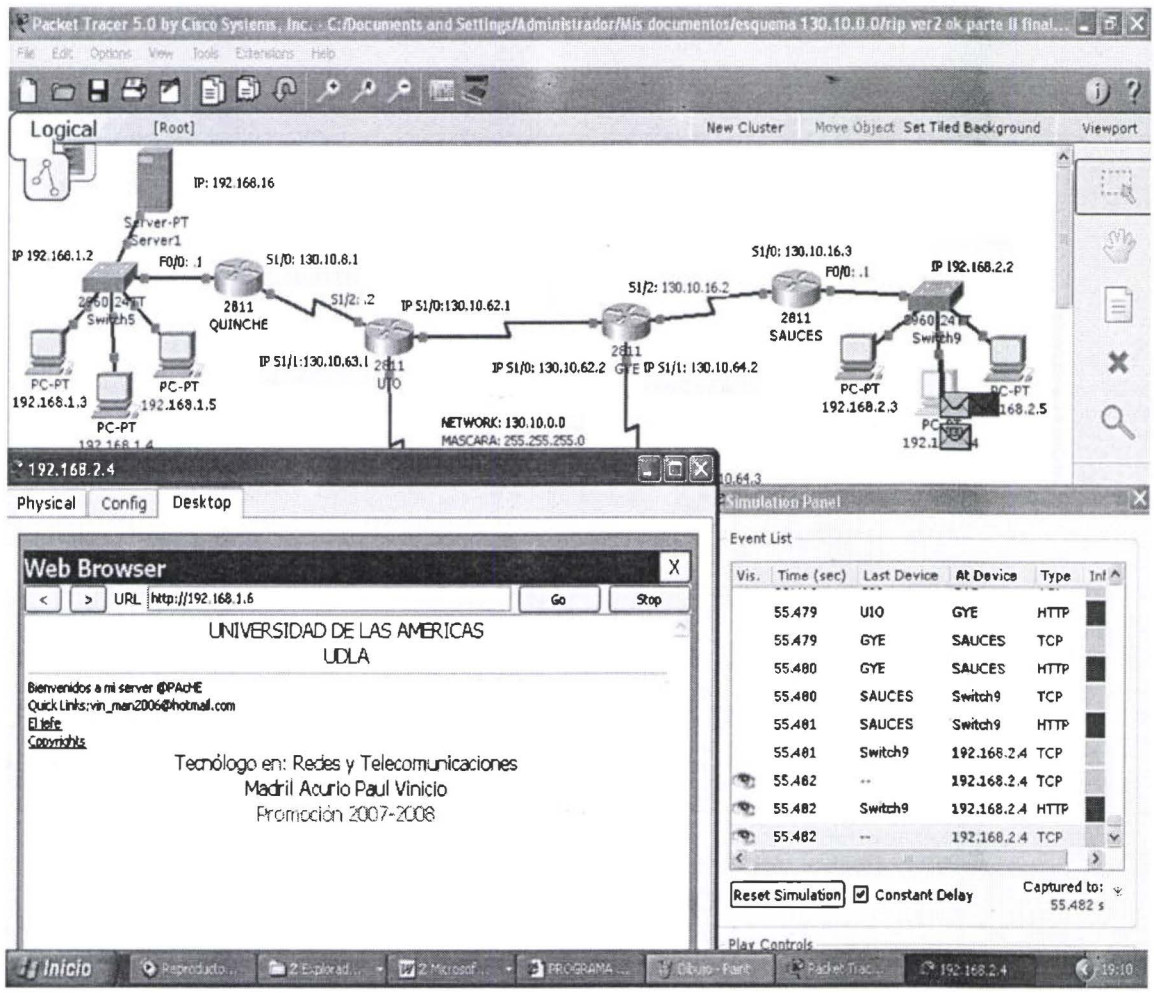
Una vez que llegó la petición http al servidor, este se encarga en reenviar la información al destinatario final, con la respuesta de que su petición ha sido escuchada.



1.27

Una vez que llegó al destinatario final el usuario ya puede hacer uso de la petición realizada.

Por último se tiene como resultado la pagina Web en la pantalla del usuario final.



CONFIGURACION DEL PROTOCOLO DE ENRUTAMIENTO OSPF CON EL SOFTWARE PACKET TRACER 5.0

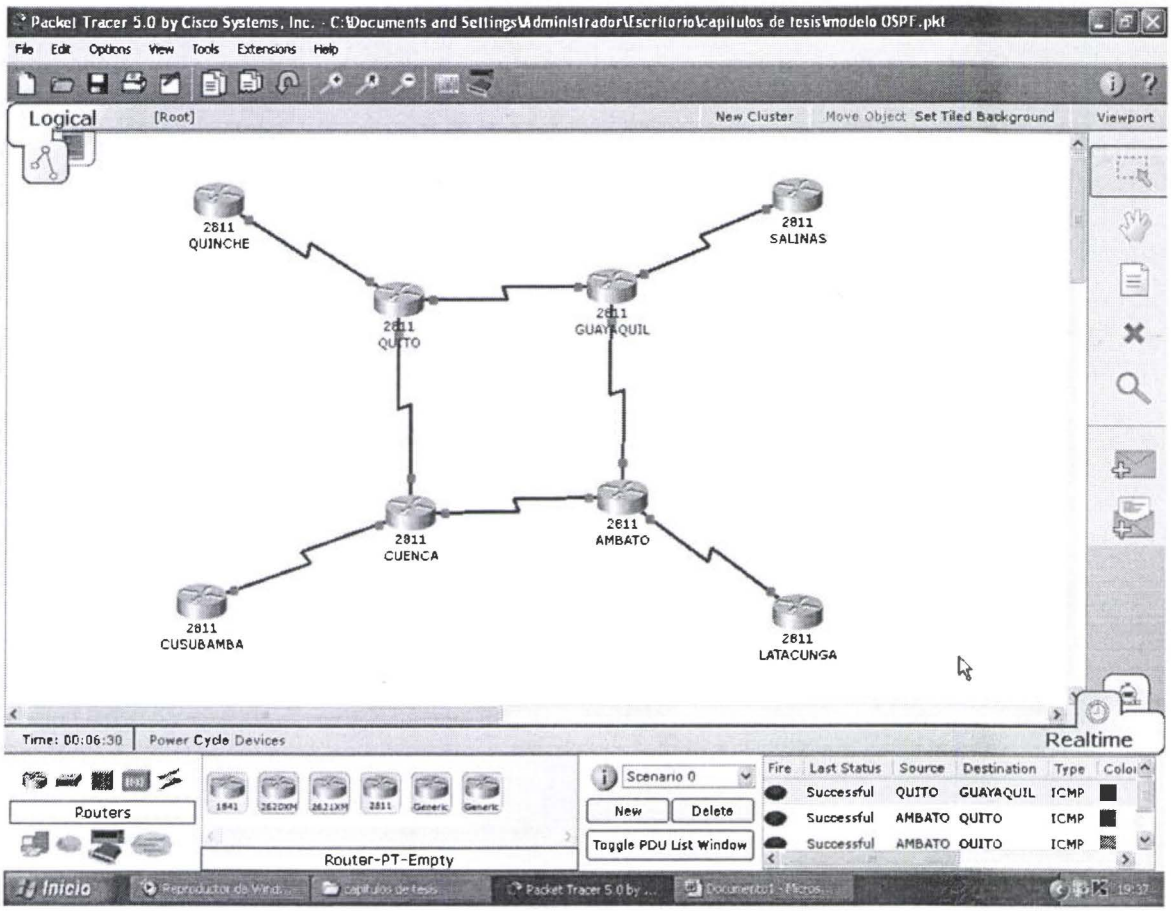
2.1

Realizadas las configuraciones de las interfaces seriales se obtendrá como resultado la conexión entre los cuatro routers del centro, los mismos que son (Quito, Guayaquil, Cuenca y Ambato), los cuales representan el backbone de la empresa, y de los cuatro routers que se encuentran a sus lados que representan las sucursales de las ciudades mencionadas. Los primeros comandos que se utilizaron para levantar las interfaces seriales de cada router fueron las misma que su utilizo en el esquema del protocolo de enrutamiento RIP ver2.

Ejemplo:

ROUTER A (UIO)

```
Router>en
Router#configure ter
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#hostname UIO
UIO(config)#interface serial 1/0
UIO(config-if)#ip add 192.168.0.1 255.255.255.0
UIO(config-if)#no shut
%LINK-5-CHANGED: Interface Serial1/0, changed state to down
UIO(config-if)#
%SYS-5-CONFIG_I: Configured from console by console
```



2.2

Una vez terminado la configuración de todas las interfaces seriales de los routers, se procederá a verificar su conexión utilizando el comando ping para tener respuesta, no obstante se tienen que tener en cuenta que solo se están viendo fronterizamente, porque aún no se ha configurado el protocolo de enrutamiento con el cual se van a comunicar los routers.

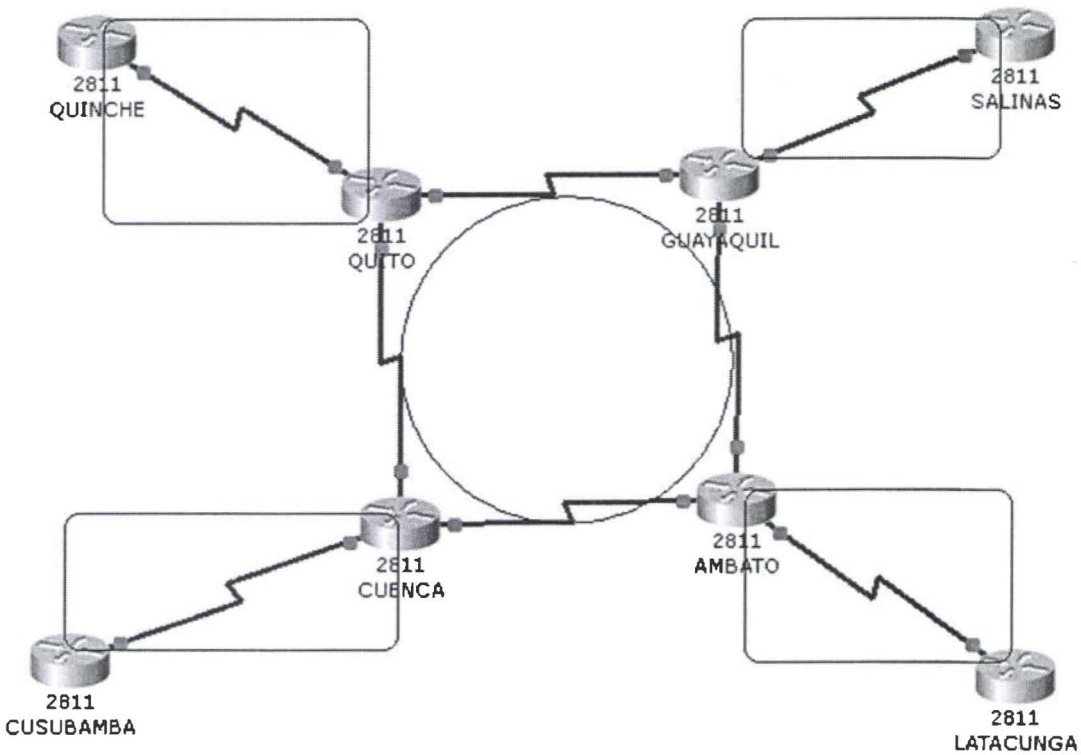
UIO#ping 192.168.0.2

Sending 5, 100-byte ICMP Echos to 192.168.0.2, timeout is 2 seconds:

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms

CONFIGURANDO OSPF

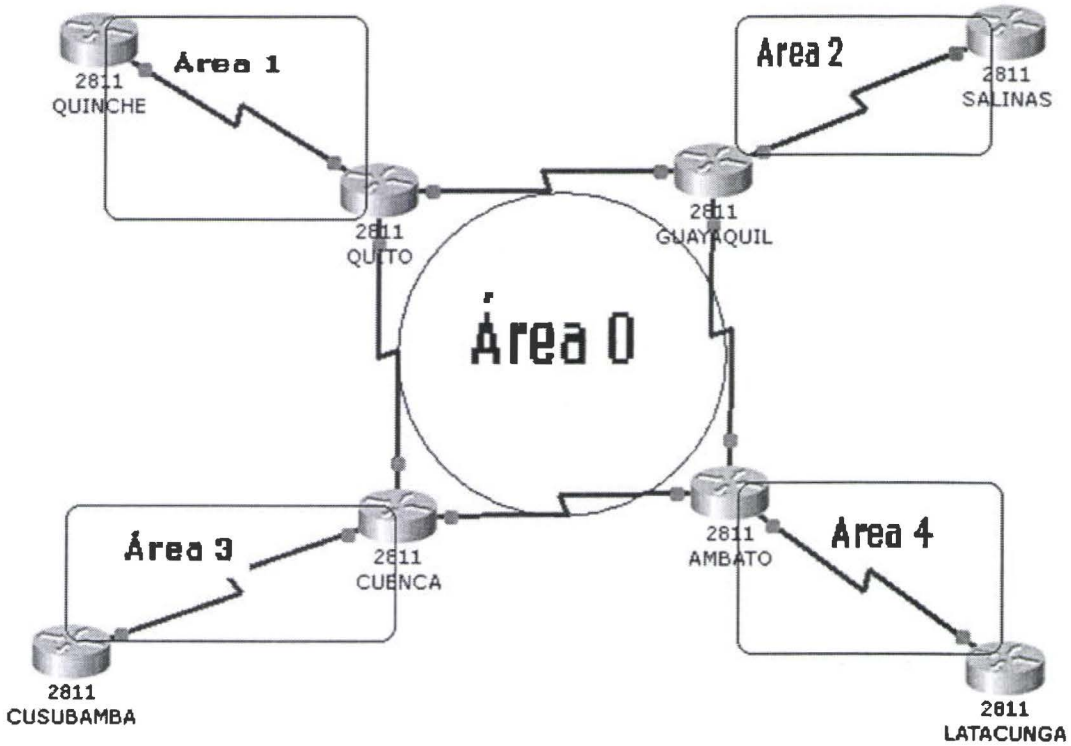


2.3

Después de haber comprobado la funcionalidad entre los routers se procede a levantar el protocolo de enrutamiento de estado de enlace OSPF. Para poder levantar el protocolo de enrutamiento OSPF se debe configurar por áreas. El área de backbone debe ser el Área 0.

OSPF funciona dividiendo una Intranet o un sistema autónomo en unidades jerárquicas de menor tamaño. Cada una de estas áreas se enlaza a un área backbone mediante un router fronterizo.

Ejemplo:



QUITO>en

QUITO#configure ter

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

QUITO(config)#**no router eigrp 100**

QUITO(config)#**router ospf 109**

QUITO(config-router)#**network 192.168.0.0 0.0.0.255 area 0**

QUITO(config-router)#**network 192.168.0.1 0.0.0.255 area 0**

%SYS-5-CONFIG_I: Configured from console by console

Realizado las configuraciones en los routers del Área 0 de backbone se tiene como repuesta.

00:00:10: %OSPF-5-ADJCHG: Process 109, Nbr 192.168.200.1 on Serial1/2 from LOADING to FULL, Loading Done

00:00:10: %OSPF-5-ADJCHG: Process 109, Nbr 192.168.22.3 on Serial1/1 from LOADING to FULL, Loading Done

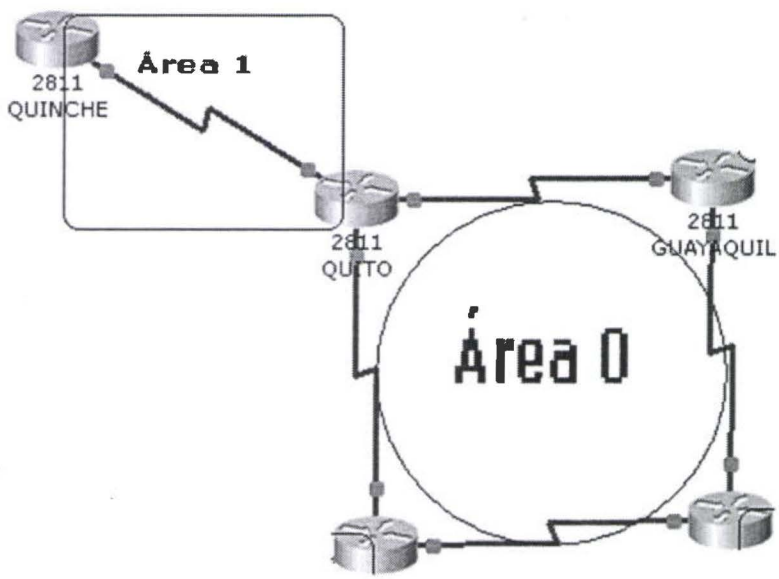
00:00:10: %OSPF-5-ADJCHG: Process 109, Nbr 192.168.21.3 on Serial1/0 from LOADING to FULL, Loading Done

2.4

Después de que se haya levantado el **Área 0** de backbone, se procederá a levantar las siguientes áreas: **Área 1, Área 2, Área 3 y el Área 4.**

Para llevar a efecto el levantamiento de cada área se debe configurar de igual manera que se realizó para el Área 0, con la diferencia de que en el router fronterizo únicamente se debe configurar el área en la cual se encuentra

Ejemplo:



ROUTER QUITO AREA 0

QUITO>en

QUITO#configure terminal

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

QUITO(config)#no router eigrp 100

QUITO(config)#router ospf 109

QUITO(config-router)#network 192.168.0.0 0.0.0.255 area 0

QUITO(config-router)#network 192.168.0.1 0.0.0.255 area 0

QUITO(config-router)#**network 192.168.20.1 0.0.0.255 area 1**

%SYS-5-CONFIG_I: Configured from console by console

ROUTER QUINCHE AREA 1

QUINCHE>en

QUINCHE#configure terminal

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

QUINCHE(config)#no router eigrp 100

QUINCHE(config)#router ospf 109

QUINCHE(config-router)#**network 192.168.20.1 0.0.0.255 area 1**

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to up

00:00:10: %OSPF-5-ADJCHG: Process 109, Nbr 192.168.20.2 on Serial1/0 from LOADING to FULL, Loading Done

Este es el proceso que deben seguir cada una de las siguientes tres áreas de los routers fronterizos y los routers de backbone que se encuentran el Área 0, para su comunicación entre cada área. Además se puede observar en el IOS de los routers los Database y Neighbor de cada uno de ellos con el comando:

Show ip ospf database:

QUITO#sho ip ospf database

OSPF Router with ID (192.168.20.2) (Process ID 109)

Router Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
192.168.21.3	192.168.21.3	47	0x80000004	0x0041f0	4
192.168.20.2	192.168.20.2	47	0x80000004	0x0063d3	4
192.168.22.3	192.168.22.3	47	0x80000004	0x005ad0	4
192.168.23.3	192.168.23.3	43	0x80000004	0x0082a2	4

Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.20.0	192.168.20.2	43	0x80000001	0x00f35e
192.168.23.0	192.168.23.3	43	0x80000001	0x00b793
192.168.21.0	192.168.21.3	43	0x80000001	0x00db73
192.168.22.0	192.168.22.3	43	0x80000001	0x00c983

Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum Link count
192.168.20.2	192.168.20.2	47	0x80000002	0x0007ee 2
192.168.200.1	192.168.200.1	47	0x80000002	0x000143 2

Summary Net Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum
192.168.1.0	192.168.20.2	43	0x80000001	0x00c59f
192.168.0.0	192.168.20.2	43	0x80000002	0x00ce96
192.168.3.0	192.168.20.2	38	0x80000003	0x004908
192.168.2.0	192.168.20.2	38	0x80000004	0x0052fe
192.168.23.0	192.168.20.2	38	0x80000005	0x000625
192.168.21.0	192.168.20.2	38	0x80000006	0x007cbf
192.168.22.0	192.168.20.2	38	0x80000007	0x006fca

Con el comando **show ip ospf neighbor** se puede observar los vecinos que tiene el router

QUITO#sho ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.21.3	1	FULL/-	00:00:37	192.168.0.2	Serial1/0
192.168.22.3	1	FULL/-	00:00:37	192.168.1.3	Serial1/1
192.168.200.1	1	FULL/-	00:00:37	192.168.20.1	Serial1/2

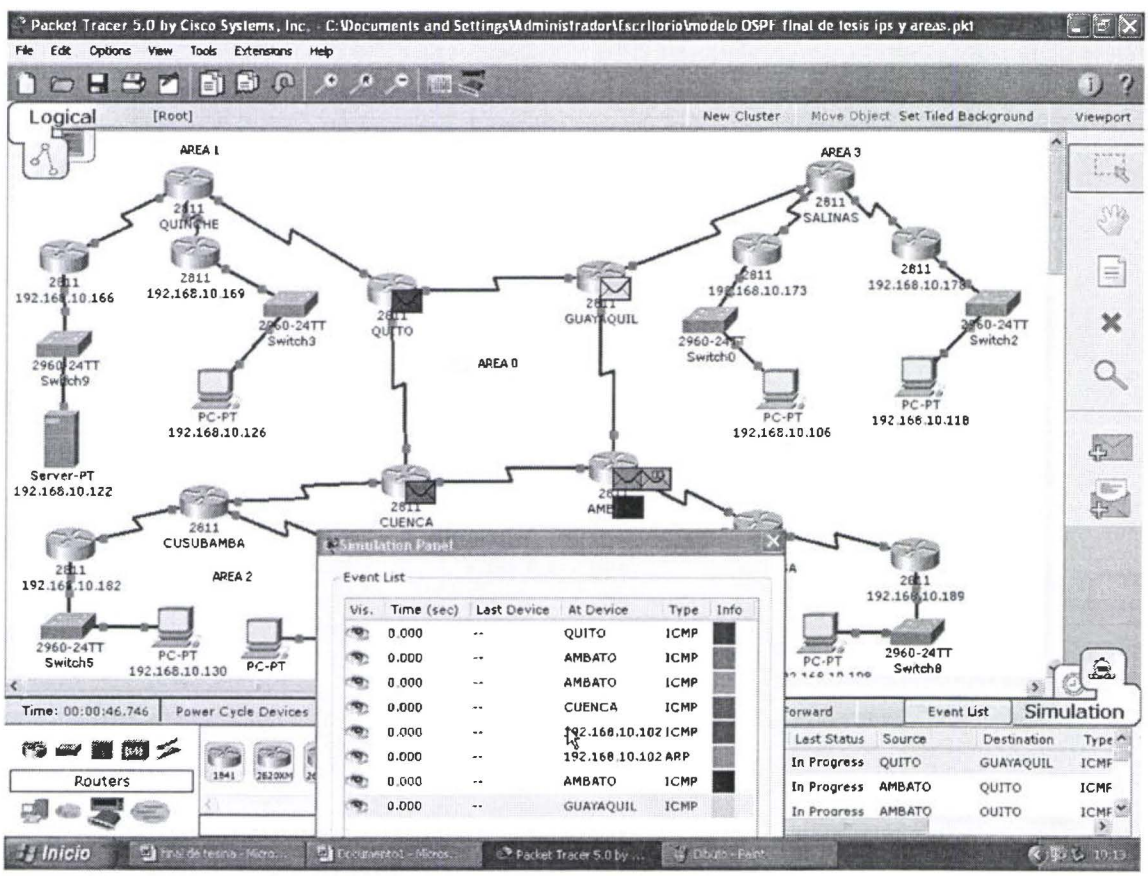
ANALISIS DEL PROTOCOLO DE ENRUTAMIENTO OSFP CON REAL TIME DEL PACKET TRACER 5.0

2.5

El paquete PDU está analizando las conexiones de los routers con el protocolo ICMP, es un ping que se está realizando a cada uno de los routers que conforman la red; y que permitirá saber si se tiene conexión con todas las interfaces de la red WAN.

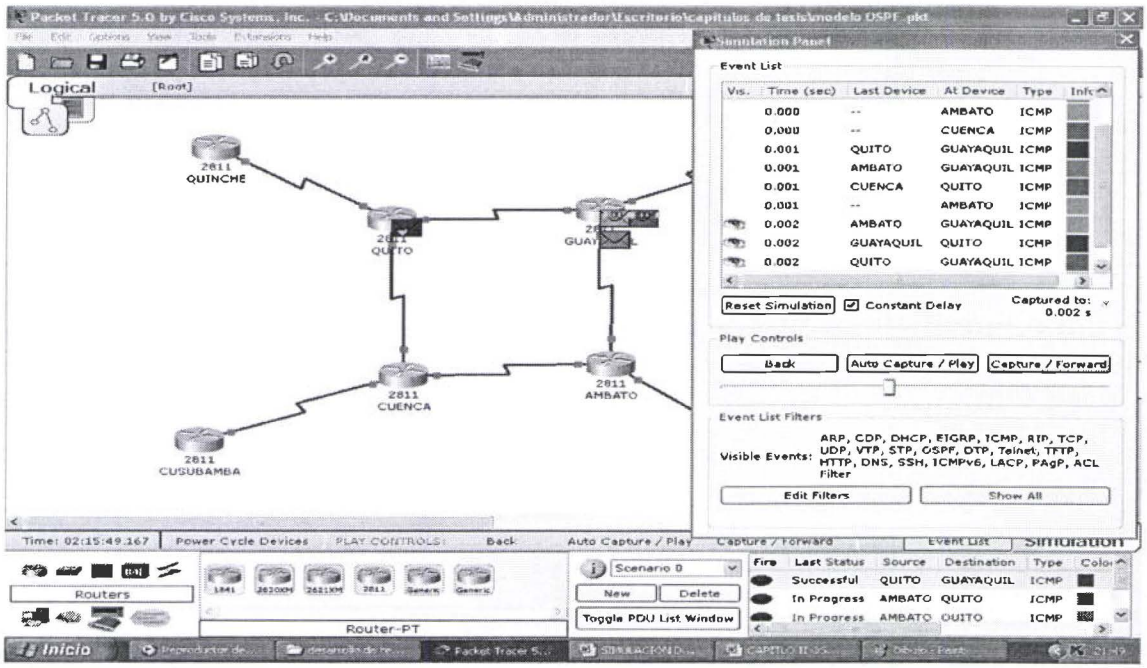
En la gráfica se observará tres paquetes PDU los cuales van a ser enviados desde el router Quito y el router Ambato

Desde el router Ambato se envía el ICMP hacia el router Guayaquil que tendrá como respuesta que no se perdió el paquete.



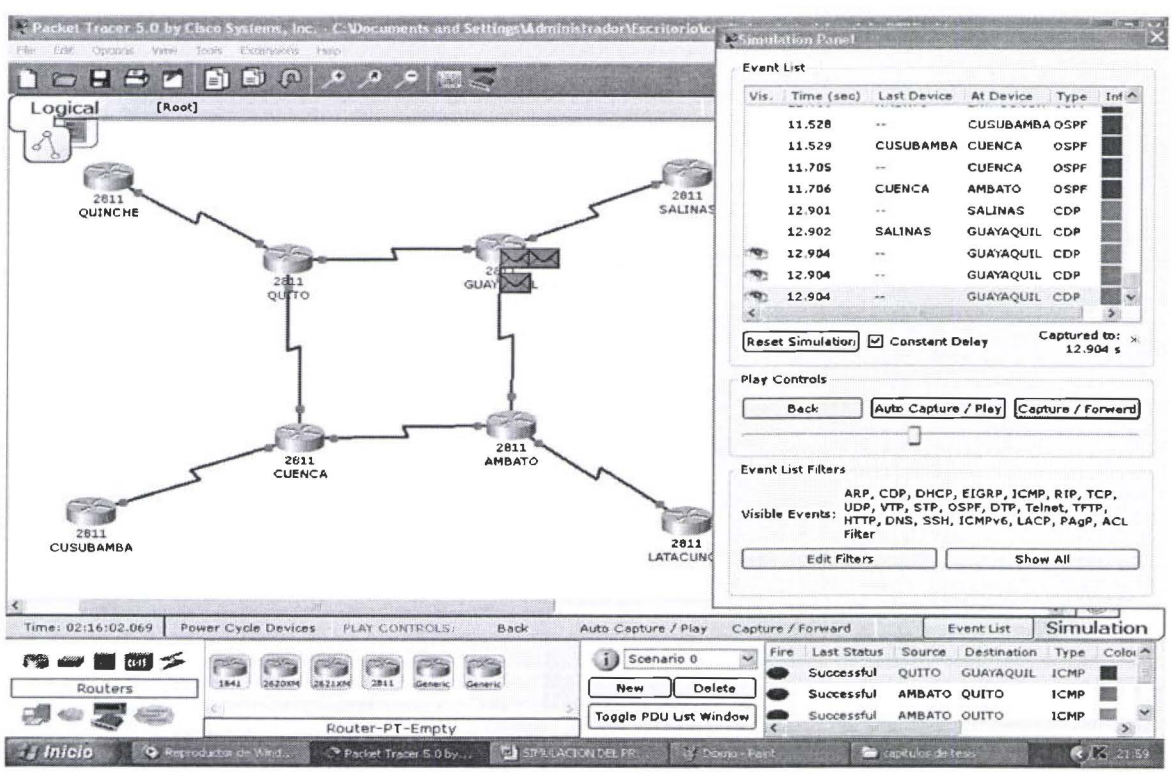
2.6

De igual forma el siguiente ICMP se realizará desde el router Guayaquil hacia el router Quito.



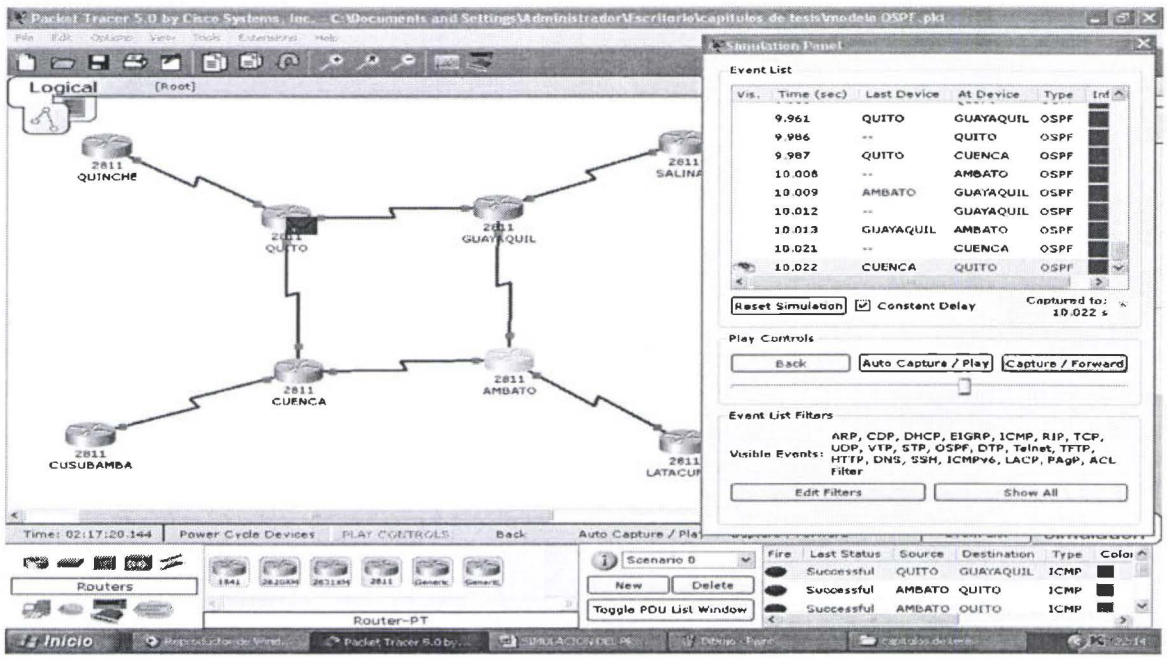
2.7

Una vez que haya pasado a todos los routers el protocolo ICMP, el siguiente paquete que se enviara es el CDP (**C**isco **D**iscovery **P**rotocol), que es un protocolo de red propietario de nivel 2, desarrollado por Cisco Systems y usado en la mayoría de sus equipos. Es utilizado para compartir información sobre otros equipos Cisco directamente conectados, tal como la versión del sistema operativo y la dirección IP.



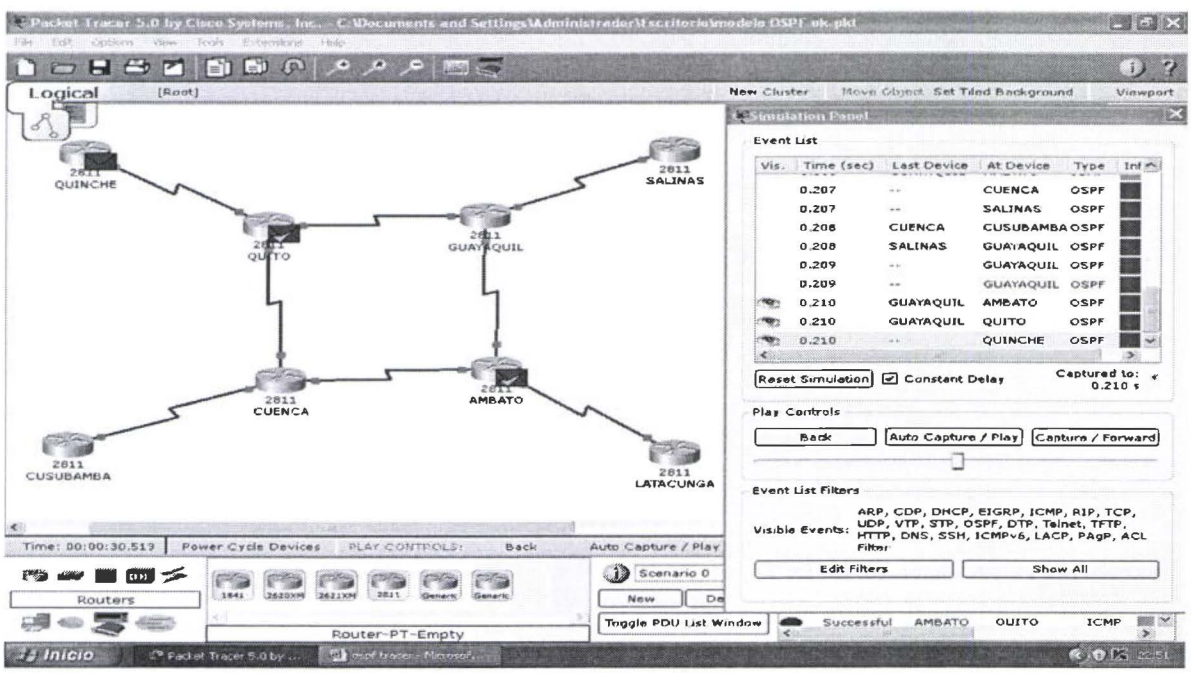
2.8

Una vez finalizado el protocolo CDP se tendrá el siguiente protocolo, el de enrutamiento OSPF que tiene como función enrutar el paquete PDU a cada uno de los routers, tanto en el área 0 de backbone, como a las demás áreas.



2.9

En segunda instancia se va a enviar los paquetes desde el router Guayaquil hacia Quito y Ambato comunicando los cambios de topología que hubiera habido para que actualicen sus tablas de ruteo.



ANALISIS DE RESULTADOS DE LOS PROTOCOLOS DEENRUTAMIENTO RIP y OSPF

RIP ver1

- Es fácil de configurar
- Solo admite protocolos de enrutamiento con clase
- No incluye información de subred en la actualización de enrutamiento
- No admite autenticación en actualizaciones
- Utiliza el número de saltos como métrica para selección de una ruta
- Si el numero se saltos es superior de 15, el paquete se descarta
- Por defecto, se envía un broadcast de actualización de enrutamiento cada 30 segundos.
- El protocolo RIP ver1 depende de los routers vecinos para obtener información de la red
-

RIP ver2:

- Protocolo de enrutamiento de Gateway interior (IGP)
- Copia la tabla de enrutamiento de los vecinos
- Se actualiza frecuentemente
- RIP ver2 usan el numero de saltos como métrica
- Visualiza la red desde la perspectiva de la red de los vecinos
- Converge lentamente
- Es susceptible a los bucles de enrutamiento
- Fácil de configurar y administrar
- Consume una gran cantidad de ancho de banda

OSPF:

- Usa la ruta mas corta
- Las actualizaciones son desencadenadas por los eventos
- Envía paquetes de estado de enlace a todos los routers de la red
- Tiene una vista común de la red
- Converge rápidamente
- No es susceptible a los bucles de enrutamiento
- Es fácil de configurar
- Requiere más memoria y potencia de procesamiento que el de vector-distancia
- Consume menos ancho de banda que el vector-distancia

ANALISIS DE RESULTADOS ENTRE RIP Y OSPF

PROTOCOLO DE ENRUTAMIENTO RIP	PROTOCOLO DE ENRUTAMIENTO OSPF
En la primera tabla de datos se realizó un ping desde el host 192.168.1.4 hacia el host 192.168.2.4 teniendo como respuesta positiva: En la primera respuesta se tiene que fueren enviados 32 bytes con el tiempo 176ms y el TTL que significa el tiempo de vida de un paquete hasta su destino fue 124. En cada una de las respuestas significa el tiempo que	En la primera tabla de datos se realizó un ping desde el host 192.168.10.106 hacia el host 192.168.10.126 teniendo como respuesta positiva: En la primera respuesta se tiene que fueren enviados 32 bytes con el tiempo 143ms y el TTL que significa el tiempo de vida de un paquete hasta su destino fue 125. En cada

recorre el paquete hacia su destino, y de la contestación de la solicitud requerida por el mismo.	una de las respuestas significa que es el tiempo que recorre el paquete hacia su destino. Se tiene que en cada una de las respuesta, el tiempo es menor al compararlo con rip
PC>ping 192.168.2.4 Pinging 192.168.2.4 with 32 bytes of data: Request timed out. Reply from 192.168.2.4: bytes=32 time=**176ms** TTL=124 Reply from 192.168.2.4: bytes=32 time=**154ms** TTL=124 Reply from 192.168.2.4: bytes=32 time=**140ms** TTL=124 Ping statistics for 192.168.2.4: Packets: Sent = 4, Received = 3, Lost = 1 (25% loss), Approximate round trip times in milli-seconds: **Minimum = 140ms, Maximum = 176ms, Average = 156ms** **PC>ping 192.168.4.4** Pinging 192.168.4.4 with 32 bytes of data:	**PC>ping 192.168.10.106** Pinging 192.168.10.106 with 32 bytes of data: Reply from 192.168.10.106: bytes=32 time=**143**ms TTL=125 Reply from 192.168.10.106: bytes=32 time=**140ms** TTL=125 Reply from 192.168.10.106: bytes=32 time=**125ms** TTL=125 Reply from 192.168.10.106: bytes=32 time=**112ms** TTL=125 Ping statistics for 192.168.10.106: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: **Minimum = 112ms, Maximum = 143ms, Average = 130ms** **PC>ping 192.168.10.126** Pinging 192.168.10.126 with 32 bytes of data:

Reply from 192.168.4.4: bytes=32 time=219ms TTL=123 Reply from 192.168.4.4: bytes=32 time=216ms TTL=123 Reply from 192.168.4.4: bytes=32 time=203ms TTL=123 Reply from 192.168.4.4: bytes=32 time=203ms TTL=123 Ping statistics for 192.168.4.4: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 203ms, Maximum = 219ms, Average = 210ms	Reply from 192.168.10.126: bytes=32 time=250ms TTL=122 Reply from 192.168.10.126: bytes=32 time=220ms TTL=122 Reply from 192.168.10.126: bytes=32 time=203ms TTL=122 Reply from 192.168.10.126: bytes=32 time=265ms TTL=122 Ping statistics for 192.168.10.126: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 203ms, Maximum = 265ms, Average = 234ms
PC>ping 192.168.3.3	**PC>ping 192.168.10.122**
Pinging 192.168.3.3 with 32 bytes of data:	Pinging 192.168.10.122 with 32 bytes of data:
Reply from 192.168.3.3: bytes=32 time=188ms TTL=124 Reply from 192.168.3.3: bytes=32 time=172ms TTL=124 Reply from 192.168.3.3: bytes=32 time=167ms TTL=124 Reply from 192.168.3.3: bytes=32 time=204ms TTL=124 Ping statistics for 192.168.3.3: Packets: Sent = 4, Received = 4,	Reply from 192.168.10.122: bytes=32 time=171ms TTL=122 Reply from 192.168.10.122: bytes=32 time=219ms TTL=122 Reply from 192.168.10.122: bytes=32 time=203ms TTL=122 Reply from 192.168.10.122: bytes=32 time=184ms TTL=122 Ping statistics for 192.168.10.122: Packets: Sent = 4, Received = 4,

Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 167ms, Maximum = 204ms, Average = 182ms **PC>ping 192.168.1.5** Pinging 192.168.1.5 with 32 bytes of data: Reply from 192.168.1.5: bytes=32 time=109ms TTL=128 Reply from 192.168.1.5: bytes=32 time=46ms TTL=128 Reply from 192.168.1.5: bytes=32 time=47ms TTL=128 Reply from 192.168.1.5: bytes=32 time=47ms TTL=128 Ping statistics for 192.168.1.5: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 46ms, Maximum = 109ms, Average = 62ms	Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 171ms, Maximum = 219ms, Average = 194ms **PC>ping 192.168.10.130** Pinging 192.168.10.130 with 32 bytes of data: Reply from 192.168.10.130: bytes=32 time=172ms TTL=121 Reply from 192.168.10.130: bytes=32 time=153ms TTL=121 Reply from 192.168.10.130: bytes=32 time=231ms TTL=121 Reply from 192.168.10.130: bytes=32 time=266ms TTL=121 Ping statistics for 192.168.10.130: Packets: Sent = 4, Received = 4, Lost = 0 (0% loss), Approximate round trip times in milli-seconds: Minimum = 153ms, Maximum = 266ms, Average = 205ms
Protocolo RIP ver2 desde QUITO	**Protocolo OSPF desde QUITO**

Para verificar la tabla de ruteo de los routers, se debe tener levantado el servicio (debug ip rip), este comando es el que nos informa sobre las actualizaciones que tienen cada uno de los routers, y además nos indica los números de saltos o métrica que dan para llegar hacia los siguientes routers. En la vida real las métricas o saltos que son factibles son 15, esto es una debilidad que tiene RIP, ya que no se podría tener una red estable cuando la red supere los 50 routers, es por este motivo que RIP es susceptible a los bucles de datos.	Desde el router UIO se realizo el levantamiento del (debug ip ospf events), el cual nos mostrará la tabla de ruteo, la cual se encuentra en constante actualización, claramente se observa que la tabla de ruteo se actualiza de una manera mucho mayor que RIP, ya que lo hace por milésimas de segundo, por el motivo de que no tiene encuenta el número de saltos para su actualización, si no que lo hace de una manera desencadenada, por este motivo la convergencia de la red es sumamente rápida que RIP. OSPF puede superar más de 100 routers conectados ya que no depende de los números de saltos para actualizar su tabla de ruteo.

QUITO (Debug ip rip)	**QUITO (Debug ip ospf events)**
RIP: sending v2 update to 224.0.0.9 via Serial1/1 (130.10.63.1) RIP: build update entries 130.10.8.0/24 via 0.0.0.0, metric 1, tag 0 130.10.16.0/24 via 0.0.0.0, metric 2, tag 0 130.10.62.0/24 via 0.0.0.0, metric 1, tag 0	**OSPF events debugging is on** QUITO# 00:06:30: OSPF: Rcv hello from 192.168.22.3 area 0 from Serial1/1 192.168.1.3 00:06:30: OSPF: End of hello processing 00:06:30: OSPF: Rcv hello from 192.168.200.1 area 1 from Serial1/2

130.10.64.0/24 via 0.0.0.0, metric 2, tag 0 192.168.1.0/24 via 0.0.0.0, metric 2, tag 0 192.168.2.0/24 via 0.0.0.0, metric 3, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/0 (130.10.62.1) RIP: build update entries 130.10.8.0/24 via 0.0.0.0, metric 1, tag 0 130.10.24.0/24 via 0.0.0.0, metric 2, tag 0 130.10.63.0/24 via 0.0.0.0, metric 1, tag 0 130.10.65.0/24 via 0.0.0.0, metric 2, tag 0 192.168.1.0/24 via 0.0.0.0, metric 2, tag 0 192.168.3.0/24 via 0.0.0.0, metric 3, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/2 (130.10.8.2) RIP: build update entries 130.10.16.0/24 via 0.0.0.0, metric 2, tag 0 130.10.17.0/24 via 0.0.0.0, metric 3, tag 0 130.10.24.0/24 via 0.0.0.0, metric 2, tag 0 130.10.62.0/24 via 0.0.0.0, metric 1, tag 0	192.168.20.1 00:06:30: OSPF: End of hello processing 00:06:30: OSPF: Rcv hello from 192.168.21.3 area 0 from Serial1/0 192.168.0.2 00:06:30: OSPF: End of hello processing 00:06:40: OSPF: Rcv hello from 192.168.22.3 area 0 from Serial1/1 192.168.1.3 00:06:40: OSPF: End of hello processing 00:06:40: OSPF: Rcv hello from 192.168.200.1 area 1 from Serial1/2 192.168.20.1 00:06:40: OSPF: End of hello processing 00:06:40: OSPF: Rcv hello from 192.168.21.3 area 0 from Serial1/0 192.168.0.2 00:06:40: OSPF: End of hello processing 00:06:50: OSPF: Rcv hello from 192.168.22.3 area 0 from Serial1/1 192.168.1.3 00:06:50: OSPF: End of hello processing 00:06:50: OSPF: Rcv hello from 192.168.200.1 area 1 from Serial1/2 192.168.20.1 00:06:50: OSPF: End of hello

130.10.63.0/24 via 0.0.0.0, metric 1, tag 0 130.10.64.0/24 via 0.0.0.0, metric 2, tag 0 130.10.65.0/24 via 0.0.0.0, metric 2, tag 0 192.168.2.0/24 via 0.0.0.0, metric 3, tag 0 192.168.3.0/24 via 0.0.0.0, metric 3, tag 0 192.168.4.0/24 via 0.0.0.0, metric 4, tag 0	processing 00:06:50: OSPF: Rcv hello from 192.168.21.3 area 0 from Serial1/0 192.168.0.2 00:06:50: OSPF: End of hello processing 00:07:00: OSPF: Rcv hello from 192.168.22.3 area 0 from Serial1/1 192.168.1.3 00:07:00: OSPF: End of hello processing 00:07:00: OSPF: Rcv hello from 192.168.200.1 area 1 from Serial1/2 192.168.20.1 00:07:00: OSPF: End of hello processing 00:07:00: OSPF: Rcv hello from 192.168.21.3 area 0 from Serial1/0 192.168.0.2 00:07:00: OSPF: End of hello processing 00:07:10: OSPF: Rcv hello from 192.168.22.3 area 0 from Serial1/1 192.168.1.3 00:07:10: OSPF: End of hello processing 00:07:10: OSPF: Rcv hello from 192.168.200.1 area 1 from Serial1/2 192.168.20.1 00:07:10: OSPF: End of hello processing 00:07:10: OSPF: Rcv hello from
GUAYAQUIL RIP: sending v2 update to 224.0.0.9 via Serial1/1 (130.10.64.2) RIP: build update entries 130.10.8.0/24 via 0.0.0.0, metric 2, tag 0 130.10.16.0/24 via 0.0.0.0, metric 1, tag 0 130.10.62.0/24 via 0.0.0.0, metric 1, tag 0 130.10.63.0/24 via 0.0.0.0, metric 2, tag 0 192.168.1.0/24 via 0.0.0.0, metric 3, tag 0 192.168.2.0/24 via 0.0.0.0, metric 2, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/2 (130.10.16.2) RIP: build update entries	

130.10.8.0/24 via 0.0.0.0, metric 2, tag 0 130.10.17.0/24 via 0.0.0.0, metric 2, tag 0 130.10.24.0/24 via 0.0.0.0, metric 3, tag 0 130.10.62.0/24 via 0.0.0.0, metric 1, tag 0 130.10.63.0/24 via 0.0.0.0, metric 2, tag 0 130.10.64.0/24 via 0.0.0.0, metric 1, tag 0 130.10.65.0/24 via 0.0.0.0, metric 2, tag 0 192.168.1.0/24 via 0.0.0.0, metric 3, tag 0 192.168.3.0/24 via 0.0.0.0, metric 4, tag 0 192.168.4.0/24 via 0.0.0.0, metric 3, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/0 (130.10.62.2) RIP: build update entries 130.10.16.0/24 via 0.0.0.0, metric 1, tag 0 130.10.17.0/24 via 0.0.0.0, metric 2, tag 0 130.10.64.0/24 via 0.0.0.0, metric 1, tag 0 130.10.65.0/24 via 0.0.0.0, metric 2, tag 0	192.168.21.3 area 0 from Serial1/0 192.168.0.2 00:07:10: OSPF: End of hello processing **GUAYAQUIL** GUAYAQUIL#debug ip ospf events OSPF events debugging is on GUAYAQUIL# 00:04:20: OSPF: Rcv hello from 192.168.21.2 area 3 from Serial1/2 192.168.21.2 00:04:20: OSPF: End of hello processing 00:04:20: OSPF: Rcv hello from 192.168.23.3 area 0 from Serial1/1 192.168.2.3 00:04:20: OSPF: End of hello processing 00:04:20: OSPF: Rcv hello from 192.168.20.2 area 0 from Serial1/0 192.168.0.1 00:04:20: OSPF: End of hello processing 00:04:30: OSPF: Rcv hello from 192.168.23.3 area 0 from Serial1/1 192.168.2.3 00:04:30: OSPF: End of hello processing 00:04:30: OSPF: Rcv hello from 192.168.21.2 area 3 from Serial1/2

192.168.2.0/24 via 0.0.0.0, metric 2, tag 0 192.168.4.0/24 via 0.0.0.0, metric 3, tag 0 **CUENCA** RIP: sending v2 update to 224.0.0.9 via Serial1/1 (130.10.63.3) RIP: build update entries 130.10.17.0/24 via 0.0.0.0, metric 2, tag 0 130.10.24.0/24 via 0.0.0.0, metric 1, tag 0 130.10.64.0/24 via 0.0.0.0, metric 2, tag 0 130.10.65.0/24 via 0.0.0.0, metric 1, tag 0 192.168.3.0/24 via 0.0.0.0, metric 2, tag 0 192.168.4.0/24 via 0.0.0.0, metric 3, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/2 (130.10.24.2) RIP: build update entries 130.10.8.0/24 via 0.0.0.0, metric 2, tag 0 130.10.16.0/24 via 0.0.0.0, metric 3, tag 0 130.10.17.0/24 via 0.0.0.0, metric 2, tag 0 130.10.62.0/24 via 0.0.0.0, metric 2, tag 0	192.168.21.2 00:04:30: OSPF: End of hello processing 00:04:30: OSPF: Rcv hello from 192.168.20.2 area 0 from Serial1/0 192.168.0.1 00:04:30: OSPF: End of hello processing 00:04:40: OSPF: Rcv hello from 192.168.21.2 area 3 from Serial1/2 192.168.21.2 00:04:40: OSPF: End of hello processing 00:04:40: OSPF: Rcv hello from 192.168.23.3 area 0 from Serial1/1 192.168.2.3 00:04:40: OSPF: End of hello processing 00:04:40: OSPF: Rcv hello from 192.168.20.2 area 0 from Serial1/0 192.168.0.1 00:04:40: OSPF: End of hello processing 00:04:50: OSPF: Rcv hello from 192.168.21.2 area 3 from Serial1/2 192.168.21.2 00:04:50: OSPF: End of hello processing 00:04:50: OSPF: Rcv hello from 192.168.23.3 area 0 from Serial1/1 192.168.2.3 00:04:50: OSPF: End of hello

130.10.63.0/24 via 0.0.0.0, metric 1, tag 0 130.10.64.0/24 via 0.0.0.0, metric 2, tag 0 130.10.65.0/24 via 0.0.0.0, metric 1, tag 0 192.168.1.0/24 via 0.0.0.0, metric 3, tag 0 192.168.2.0/24 via 0.0.0.0, metric 4, tag 0 192.168.4.0/24 via 0.0.0.0, metric 3, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/0 (130.10.65.1) RIP: build update entries 130.10.8.0/24 via 0.0.0.0, metric 2, tag 0 130.10.24.0/24 via 0.0.0.0, metric 1, tag 0 130.10.62.0/24 via 0.0.0.0, metric 2, tag 0 130.10.63.0/24 via 0.0.0.0, metric 1, tag 0 192.168.1.0/24 via 0.0.0.0, metric 3, tag 0 192.168.3.0/24 via 0.0.0.0, metric 2, tag 0 **AMBATO** RIP: sending v2 update to 224.0.0.9 via Serial1/0 (130.10.65.9) RIP: build update entries	processing 00:04:50: OSPF: Rcv hello from 192.168.20.2 area 0 from Serial1/0 192.168.0.1 00:04:50: OSPF: End of hello processing 00:05:00: OSPF: Rcv hello from 192.168.21.2 area 3 from Serial1/2 192.168.21.2 00:05:00: OSPF: End of hello processing 00:05:00: OSPF: Rcv hello from 192.168.23.3 area 0 from Serial1/1 192.168.2.3 00:05:00: OSPF: End of hello processing 00:05:00: OSPF: Rcv hello from 192.168.20.2 area 0 from Serial1/0 192.168.0.1 00:05:00: OSPF: End of hello processing 05:00: OSPF: End of hello processing 00:05:00: OSPF: Rcv hello from 192.168.23.3 area 0 from Serial1/1 192.168.2.3 00:05:00: OSPF: End of hello processing 00:05:00: OSPF: Rcv hello from 192.168.20.2 area 0 from Serial1/0 192.168.0.1 00:05:00: OSPF: End of hello

130.10.16.0/24 via 0.0.0.0, metric 2, tag 0 130.10.17.0/24 via 0.0.0.0, metric 1, tag 0 130.10.62.0/24 via 0.0.0.0, metric 2, tag 0 130.10.64.0/24 via 0.0.0.0, metric 1, tag 0 192.168.2.0/24 via 0.0.0.0, metric 3, tag 0 192.168.4.0/24 via 0.0.0.0, metric 2, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/2 (130.10.17.2) RIP: build update entries 130.10.8.0/24 via 0.0.0.0, metric 3, tag 0 130.10.16.0/24 via 0.0.0.0, metric 2, tag 0 130.10.24.0/24 via 0.0.0.0, metric 2, tag 0 130.10.62.0/24 via 0.0.0.0, metric 2, tag 0 130.10.63.0/24 via 0.0.0.0, metric 2, tag 0 130.10.64.0/24 via 0.0.0.0, metric 1, tag 0 130.10.65.0/24 via 0.0.0.0, metric 1, tag 0 192.168.1.0/24 via 0.0.0.0, metric 4, tag 0 192.168.2.0/24 via 0.0.0.0, metric	processing 00:05:10: OSPF: Rcv hello from 192.168.21.2 area 3 from Serial1/2 192.168.21.2 00:05:10: OSPF: End of hello processing 00:05:10: OSPF: Rcv hello from 192.168.20.2 area 0 from Serial1/0 192.168.0.1 00:05:10: OSPF: End of hello processing 00:05:10: OSPF: Rcv hello from 192.168.23.3 area 0 from Serial1/1 192.168.2.3 00:05:10: OSPF: End of hello processing **QUINCHE#** 06:28:39: OSPF: Rcv hello from 192.168.20.2 area 1 from Serial1/0 192.168.20.2 06:28:39: OSPF: End of hello processing 06:28:40: OSPF: Rcv hello from 192.168.10.169 area 1 from Serial1/2 192.168.10.169 06:28:40: OSPF: End of hello processing 06:28:41: OSPF: Rcv hello from 192.168.10.166 area 1 from Serial1/1 192.168.10.166 06:28:41: OSPF: End of hello

3, tag 0 192.168.3.0/24 via 0.0.0.0, metric 3, tag 0 RIP: sending v2 update to 224.0.0.9 via Serial1/1 (130.10.64.3) RIP: build update entries 130.10.17.0/24 via 0.0.0.0, metric 1, tag 0 130.10.24.0/24 via 0.0.0.0, metric 2, tag 0 130.10.63.0/24 via 0.0.0.0, metric 2, tag 0 130.10.65.0/24 via 0.0.0.0, metric 1, tag 0 192.168.3.0/24 via 0.0.0.0, metric 3, tag 0 192.168.4.0/24 via 0.0.0.0, metric 2, tag 0 **QUINCHE** RIP: sending v2 update to 224.0.0.9 via Serial1/0 (130.10.8.1) RIP: build update entries 192.168.1.0/24 via 0.0.0.0, metric 1, tag 0 RIP: received v2 update from 130.10.8.2 on Serial1/0 130.10.16.0/24 via 0.0.0.0 in 2 hops 130.10.17.0/24 via 0.0.0.0 in 3	processing 06:28:49: OSPF: Rcv hello from 192.168.20.2 area 1 from Serial1/0 192.168.20.2 06:28:49: OSPF: End of hello processing 06:28:50: OSPF: Rcv hello from 192.168.10.169 area 1 from Serial1/2 192.168.10.169 06:28:50: OSPF: End of hello processing 06:28:51: OSPF: Rcv hello from 192.168.10.166 area 1 from Serial1/1 192.168.10.166 06:28:51: OSPF: End of hello processing

hops 130.10.24.0/24 via 0.0.0.0 in 2 hops 130.10.62.0/24 via 0.0.0.0 in 1 hops 130.10.63.0/24 via 0.0.0.0 in 1 hops 130.10.64.0/24 via 0.0.0.0 in 2 hops 130.10.65.0/24 via 0.0.0.0 in 2 hops 192.168.2.0/24 via 0.0.0.0 in 3 hops 192.168.3.0/24 via 0.0.0.0 in 3 hops 192.168.4.0/24 via 0.0.0.0 in 4 hops	
Una vez terminado de tener los valores y los saltos con los cuales se esta actualizando los routers, se procedió a verificar la información, el cual nos indica que todas las interfases se encuentran en funcionamiento. **GYE#show ip route** Gateway of last resort is not set	En esta segunda instancia se desconectara la interface serial 1/0 correspondiente al router Cuenca y Ambato, desde el Ambato se analizará el tiempo de convergencia con el cual las tablas de ruteo de sus vecinos actualizan, el protocolo de enrutamiento OSPF tiene la opción de ver sus cambios por si mismo, y no depende de otros routers que lo avisen los cambios efectuados para que actualicen sus tablas de ruteo.

130.10.0.0/24 is subnetted, 8 subnets R 130.10.8.0 [120/1] via 130.10.62.1, 00:00:16, Serial1/0 C 130.10.16.0 is directly connected, Serial1/2 R 130.10.17.0 [120/1] via 130.10.64.3, 00:00:18, Serial1/1 R 130.10.24.0 [120/2] via 130.10.64.3, 00:00:18, Serial1/1 [120/2] via 130.10.62.1, 00:00:16, Serial1/0 C 130.10.62.0 is directly connected, Serial1/0 R 130.10.63.0 [120/1] via 130.10.62.1, 00:00:16, Serial1/0 C 130.10.64.0 is directly connected, Serial1/1 R 130.10.65.0 [120/1] via 130.10.64.3, 00:00:18, Serial1/1 R 192.168.1.0/24 [120/2] via 130.10.62.1, 00:00:16, Serial1/0 **R 192.168.2.0/24 [120/1] via 130.10.16.3, 00:00:19, Serial1/2** R 192.168.3.0/24 [120/3] via 130.10.64.3, 00:00:18, Serial1/1 [120/3] via 130.10.62.1, 00:00:16, Serial1/0 R 192.168.4.0/24 [120/2] via 130.10.64.3, 00:00:18, Serial1/1	**AMBATO#sho ip route** Gateway of last resort is not set O 192.168.0.0/24 [110/1562] via 192.168.2.1, 00:04:57, Serial1/1 O 192.168.1.0/24 [110/1562] via 192.168.3.1, 06:10:53, Serial1/0 C 192.168.2.0/24 is directly connected, Serial1/1 C 192.168.3.0/24 is directly connected, Serial1/0 192.168.10.0/30 is subnetted, 16 subnets O 192.168.10.100 [110/1563] via 192.168.23.2, 01:13:14, Serial1/2 O IA 192.168.10.104 [110/2344] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.116 [110/2344] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.120 [110/3125] via 192.168.3.1, 00:07:31, Serial1/0 [110/3125] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.124 [110/3125] via 192.168.3.1, 01:36:54, Serial1/0 [110/3125] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.128 [110/2344] via 192.168.3.1, 00:51:00, Serial1/0 O IA 192.168.10.132 [110/2344] via 192.168.3.1, 00:51:00, Serial1/0

En esta ocasión se desconectara la interfase serial ½, para comprobar el tiempo de convergencia que tiene una ves que se vuela a conectar dicha interfase.	O IA 192.168.10.164 [110/3124] via 192.168.3.1, 05:29:41, Serial1/0 [110/3124] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.168 [110/3124] via 192.168.3.1, 05:29:31, Serial1/0 [110/3124] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.172 [110/2343] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.176 [110/2343] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.180 [110/2343] via 192.168.3.1, 00:51:00, Serial1/0 O IA 192.168.10.184 [110/2343] via 192.168.3.1, 00:51:00, Serial1/0 O 192.168.10.188 [110/1562] via 192.168.23.2, 05:00:22, Serial1/2 Se desconectara la interface serial 1/0 correspondiente a Cuenca y Ambato, desde Ambato se analizará el tiempo de convergencia con el cual las tablas de ruteo de sus vecinos se actualizan.

Gateway of last resort is not set **130.10.0.0/24 is subnetted, 7 subnets** R 130.10.8.0 [120/1] via 130.10.62.1, 00:00:09, Serial1/0	Gateway of last resort is not set O 192.168.0.0/24 [110/1562] via 192.168.2.1, 00:04:57, Serial1/1 O 192.168.1.0/24 [110/1562] via 192.168.3.1, 06:10:53, Serial1/0

R 130.10.17.0 [120/1] via 130.10.64.3, 00:00:19, Serial1/1 R 130.10.24.0 [120/2] via 130.10.64.3, 00:00:19, Serial1/1 [120/2] via 130.10.62.1, 00:00:09, Serial1/0 C 130.10.62.0 is directly connected, Serial1/0 R 130.10.63.0 [120/1] via 130.10.62.1, 00:00:09, Serial1/0 C 130.10.64.0 is directly connected, Serial1/1 R 130.10.65.0 [120/1] via 130.10.64.3, 00:00:19, Serial1/1 R 192.168.1.0/24 [120/2] via 130.10.62.1, 00:00:09, Serial1/0 R 192.168.3.0/24 [120/3] via 130.10.64.3, 00:00:19, Serial1/1 [120/3] via 130.10.62.1, 00:00:09, Serial1/0 R 192.168.4.0/24 [120/2] via 130.10.64.3, 00:00:19, Serial1/1 Una vez desconectada la interface, en la tabla de ruteo no se encontrara dicha	C 192.168.2.0/24 is directly connected, Serial1/1 **C 192.168.3.0/24 is directly connected, Serial1/0** 192.168.10.0/30 is subnetted, 16 subnets O 192.168.10.100 [110/1563] via 192.168.23.2, 01:13:14, Serial1/2 O IA 192.168.10.104 [110/2344] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.116 [110/2344] via 192.168.2.1, 00:04:57, Serial1/1 **O IA 192.168.10.120 [110/3125] via 192.168.3.1, 00:07:31, Serial1/0** [110/3125] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.124 [110/3125] via 192.168.3.1, 01:36:54, Serial1/0 [110/3125] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.128 [110/2344] via 192.168.3.1, 00:51:00, Serial1/0 O IA 192.168.10.132 [110/2344] via 192.168.3.1, 00:51:00, Serial1/0 O IA 192.168.10.164 [110/3124] via 192.168.3.1, 05:29:41, Serial1/0 [110/3124] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.168 [110/3124] via 192.168.3.1, 05:29:31, Serial1/0 [110/3124] via 192.168.2.1, 00:04:57, Serial1/1

interface, pues bien nuevamente volvemos a conectar la interface, y el tiempo para que todos los router actualicen su tabla sabiendo que está conectada, es el siguiente: **192.168.2.0/24 [120/1] via 130.10.16.3, 00:00:02, Serial1/2** QUITO **R 192.168.1.0/24 [120/1] via 130.10.8.1, 00:00:08, Serial1/02** Teniendo como respuesta en el router GYE el tiempo fue de 02ms, y en el router QUITO fue de 08ms.	O IA 192.168.10.172 [110/2343] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.176 [110/2343] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.10.180 [110/2343] via 192.168.3.1, 00:51:00, Serial1/0 O IA 192.168.10.184 [110/2343] via 192.168.3.1, 00:51:00, Serial1/0 O 192.168.10.188 [110/1562] via 192.168.23.2, 05:00:22, Serial1/2 O 192.168.10.192 [110/1562] via 192.168.23.2, 05:00:32, Serial1/2 O 192.168.10.196 [110/1563] via 192.168.23.2, 01:10:10, Serial1/2 O IA 192.168.20.0/24 [110/2343] via 192.168.3.1, 06:10:43, Serial1/0 [110/2343] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.21.0/24 [110/1562] via 192.168.2.1, 00:04:57, Serial1/1 O IA 192.168.22.0/24 [110/1562] via 192.168.3.1, 00:51:10, Serial1/0 C 192.168.23.0/24 is directly connected, Serial1/2 AMBATO# **%LINK-5-CHANGED: Interface Serial1/0, changed state to down** **%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial1/0, changed state to down** **06:15:57: %OSPF-5-ADJCHG: Process 109, Nbr 192.168.22.3 on**

	Serial1/0 from FULL to Down: Interface down
En el siguiente paso se desconectará la interface serial 1/0 que corresponde a QUITO Y GUAYAQUIL, del mismo modo se va analizará el tiempo con el cual actualizarán su tabla de ruteo, una ves que se haya conectado.	Una vez que se desconecto la interface se tiene como resultado la siguiente tabla, indicando la desconexión de la interface
El protocolo de enrutamiento RIP depende de sus vecinos para saber los cambios que hubo en la red, este es un claro ejemplo, el tiempo que se registro el router CUENCA fue menor comparado al de GUAYAQUIL, porque estos dos routers fueron informados por el router QUITO, el router de AMBATO se informo a través de sus vecinos, esto hace que el tiempo que tuvo AMBATO para actualizar su tabla fue tardío, esto hace la posibilidad a la subseptibilidad de bucles de datos que esta expuesta la red.	Por último el tiempo para que la red vuelva a converger fue mucho menor que el protocolo RIP, esto es por que el protocolo OSPF realiza sus actualizaciones en forma desencadenada, los routers avisan si hubo un cambio en la red al DR, y este a su vez se encarga en comunicar a todas las áreas el cambio topológico de la red.

CONCLUSIONES

- RIP ver1 está en la capacidad de proveer soluciones optimas en enrutamiento, en la medida en que las condiciones de la red como el tamaño y la presencia de bucles entre los enrutadores (o host en caso de ejecutar RIP) sean adecuados para su implementación. Esto hace que protocolos más complejos y elaborados como OSPF o RIP ver2 no sean los únicos que se tengan en cuenta como posibles soluciones de enrutamiento.

- RIP v2 es una versión mejorada de RIP v1. Comparte muchas de las mismas funciones que RIP v1. RIP v2, también es un protocolo de vector-distancia que utiliza el número de saltos, temporizadores de espera y horizonte dividido. Pero con la diferencia que RIP ver1 no soporta VLSM.

- OSPF es un protocolo de enrutamiento por estado de enlace que a diferencia de RIP e IGRP publican sus rutas sólo a routers vecinos, los routers OSPF envían publicaciones del estado de enlace LSA a todos los routers pertenecientes a la misma área jerárquica mediante una multidifusión de IP.

- Herramientas como el simulador Packet Tracer 5.0 de Cisco, representa una oportunidad de crecimiento bilateral, tanto la plataforma de simulación, como para la Universidad de las Americas, en este caso la universidad se esta proveyendo a si mismo una herramienta de gran aporte, en el proceso de aprendizaje y enseñanza de conceptos básicos de redes de datos, sin dejar de lado el prestigio que representa que sus estudiantes sean quienes implementen estas nuevas funcionalidades a futuro.

- Desde el punto de vista técnico, es recomendable y factible la migración del protocolo de enrutamiento RIP ver2 al protocolo de estado de enlace

OSPF en las redes del Ecuador, debido a que la gran mayoría de empresas están migrando a este protocolo de enrutamiento, ya que el mismo funciona sobre áreas, el cual le permite al administrador de Red facilitar su administración.

- En el presente proyecto se estudió la interoperabilidad de estos dos protocolos de enrutamiento, donde la comunicación de cada uno de los protocolos son aceptables en cualquier equipo sean estos de marcas diferentes y además pueden trabajar fusionados entre ellos.

- La implementación de una red basada en el protocolo de enrutamiento OSPF requiere más memoria y potencia de procesamiento, el cual brinda una mejor calidad de transmisión de datos y a su vez mejores servicios como: video conferencia, Voz IP, etc.

- Dado los avances tecnológicos actuales, y al auge de tecnologías como RIP, OSPF sería un complemento ideal para un mejor desarrollo de las redes de área metropolitana y extendida. La idea en un principio abarcaría implementar OSPF en áreas de mayor congestión de tráfico para lograr una mejor distribución y administración del mismo.

BIBLIOGRAFIA

http://www.cisco.com
http:// www.google.com
Libro cisco system CCNA 1
Libro cisco system CCNA 2
Libro cisco system CCNA 3

VERSION LIBRE Y SIN JURAMENTO

En la ciudad de san francisco de quito a los 14 días del mes de abril del 2009 siendo las 16h30, en las oficinas de la policía judicial de Pichincha, ante la presencia del señor Agente Investigador, rindo mi versión en forma libre y voluntaria, con relación a la indagación previa N° 09-03-27064-JRC, al efecto manifiesto llamarme **SEGUNDO DAVID GIRÓN CONCHA,** con CC. 0600706907 de estado civil casado, de nacionalidad ecuatoriano de 59 años de edad de instrucción secundaria con teléfono número 2521725, nacido en la provincia de Chimborazo cantón Riobamba, y domiciliado en la Urb. Jardín del Valle Pasaje 2E calle Luis Pérez , concretándome al caso que se investiga expongo textualmente lo siguiente: Soy propietario del centro de computo **GIRÓN TÉCNICA INFORMÁTICA "RIO@NET"** ubicado en la calle Ulloa 21 -10 y San Gregorio del sector Santa Clara, mismo que lo tengo por el tiempo de tres años, es así que el 25 de marzo del 2009, a eso de las 03 H00 mi hijo José Aníbal Girón Villa había recibido una llamada telefónica por parte de un funcionario de la compañía de seguridad LAARCOM, quien le había comunicado que mi local en mención había sido objeto de robo de los bienes que se detallan en la denuncia presentada por mi hijo José Girón, por lo que de inmediato mi hijo conjuntamente con mis tres hijos y yerno de nombres: Myrian, Carolina, Edison y Cristian Escudero se trasladaron a verificar mi local, donde habían tomado contacto con un funcionario de la compañía LAARCOM con quien conjuntamente había verificado que los bienes que se detallan en la denuncia no se encontraban en el lugar que permanecieron hasta el día 24 de marzo del 2009 a las 21H00 que dejé cerrando el local con las respectivas seguridades y activado el sistema de alarmas. Debo indicar que la compañía de seguridad LAARCOM mediante un contrato acordó al cuidado permanente de mi local, por lo que son los únicos responsables del custodio del local y de los bienes que existen en su interior por lo mismo que solicito que la compañía indemnice en su totalidad el costo de los bienes sustraídos, así mismo en la madrugada del 25 de marzo no pude concurrir conjuntamente con mis hijos y ,mi yerno por mi estado de salud, por lo que a eso de las 7H30 que llegué observe que mi local estuvo con la puerta lanfor violentada y en su interior vi que no había una sola computadora y todo se encontraba en un total desorden, por lo que envíe que mi hijo presentara la denuncia respectiva y proceda la investigación, adjunto fotografías tomadas a mi local, el día lunes 16 de febrero del 2009, en las que se puede apreciar que las computadoras robas se encontraban el mi local en el momento del robo, así mismo adjunto las fotografías tomadas horas después del robo. Es todo cuanto puedo manifestar en honor a la verdad, en esta parte intervienen el Señor agente investigador formulando las siguientes preguntas: 1P.- Diga el compareciente el 24 de marzo del 2009 a eso de las 21 horas que se retiró observó alguna persona en actitud sospechosa. R.- No. 2P.- Diga el compareciente si los locales aledaños a su centro de computo tienen guardias de seguridad encargados del custodio durante toda la noche. R.- No. 3P.- Diga el compareciente a que monto asciende la cantidad total del robo R.- Una vez realizado el detalle de los objetos sustraídos que constan en la denuncia constaté que sufrí un perjuicio de 9113.04 dólares americanos. 4P.- Diga el Compareciente si esta su versión la rin de en forma libre y voluntaria sin presión de ninguna naturaleza. R.- Si y para constancia de la misma firmo al pie de la presente con juntamente con el Sr. Agente Investigador. XXXXXXXXXXXX

SEGUNDO DAVID GIRÓN CONCHA
EL COMPARECIENTE

LUIS MONTES
AGENTE INVESTIGADOR PJ-P