UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

REDES PRIVADAS VIRTUALES (VPN) Y CALIDAD DE SERVICIO (QOS) EN REDES DE CONMUTACIÓN DE PAQUETES (IPV4) BASADOS EN EL PROTOCOLO DE CONMUTACIÓN DE ETIQUETAS (MPLS).

ANEXOS Y APÉNDICES

TRABAJO ESPECIAL DE GRADO

Presentada ante la

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

Como parte de los requisitos para optar al título de

INGENIERO EN TELECOMUNICACIONES

REALIZADO POR Atouguia Dos Santos, Jorge L.

PROFESOR GUÍA Cotúa, José Gregorio

FECHA 26 de septiembre de 2008

UNIVERSIDAD CATÓLICA ANDRÉS BELLO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIONES

REDES PRIVADAS VIRTUALES (VPN) Y CALIDAD DE SERVICIO (QOS) EN REDES DE CONMUTACIÓN DE PAQUETES (IPV4) BASADOS EN EL PROTOCOLO DE CONMUTACIÓN DE ETIQUETAS (MPLS).

ANEXOS Y APÉNDICES

REALIZADO POR Atouguia Dos Santos, Jorge L.

PROFESOR GUÍA Cotúa, José Gregorio

FECHA 26 de septiembre de 2008

CONTENIDO

Anexos y Apéndices

Anexo A

Crear imágenes de Máquinas Virtuales Utilizando Qemu

Se procede a realizar una imagen llamada e1.img, con un tamaño de disco duro igual a 2GB bajo el formato qcow2.

Qemu posee su propio formato para crear imágenes de MV, qcow2 es el más versátil, ocupa poco espacio en disco duro y el tamaño de la imagen se va expandiendo de acuerdo al uso de la misma. A continuación se presenta el comando encargado de efectuar la imagen.

#qemu-img create e1.img -f qcow2 2G

Una vez creada, procedemos a la instalación del SO Debian haciendo uso de la unidad de CD-ROM, nos colocamos en el directorio donde está la imagen y ejecutamos el siguiente comando:

#qemu -hda e1.img -cdrom /dev/hda -boot d -m 256

Recordar que **/dev/hda** es el dispositivo correspondiente a la unidad de lectura. La instalación transcurrirá como si se efectuase sobre un disco de 2 GB, con una arquitectura que posee una memoria RAM de 256MB (en este caso colocamos una mayor cantidad de memoria para efectuar la instalación mas rápidamente). Finalmente, una vez terminada la instalación, para ejecutar la Máquina Virtual con el nuevo SO tendremos que lanzar el comando:

#qemu -m 64 -hda e1.img

Es posible agregar más parámetros a la hora de cargar la MV, para más referencias, consultar el manual de Qemu. En siguientes anexos se explican las opciones mas utilizadas a la hora de ejecutar las MV del proyecto.

Anexo B

Guía para la compilación del *Kernel* MPLS

A fin de cumplir con los objetivos propuestos en el presente Trabajo Especial de Grado, se creó un núcleo MPLS que diese soporte técnico a los nodos involucrados.

Con el objetivo de simplificar el proceso de compilación, se hará énfasis en los comandos ejecutados desde el punto de vista de la consola perteneciente al nodo. Los pasos a seguir son los mostrados a continuación:

1.- Descargar los paquetes necesarios. Esto incluye obtener un núcleo Linux que más adelante será parcheado con el protocolo MPLS. También se descargará el paquete MPLS-Linux, además de paquetes como **iptables**, **iproute** y **edtables** con soporte MPLS. Una vez dentro del directorio /usr/src descargamos los ítems antes mencionados:

#cd /usr/src

 # wget http://switch.dl.sourceforge.net/sourceforge/mpls-linux/mpls-linux-1.950.tar.bz2

 # wget http://switch.dl.sourceforge.net/sourceforge/mpls-linux/iptables-1.3.0-2_mpls_1.950d.i386.rpm

 # wget http://switch.dl.sourceforge.net/sourceforge/mpls-linux/iproute-2.6.11-1_mpls_1.950d.i386.rpm

 # wget http://switch.dl.sourceforge.net/sourceforge/mpls-linux/ebtables-2.0.6-7_mpls_1.950d.i386.rpm

 # wget ftp://ftp.roedunet.lkams.kernel.org/pub/linux/kernel/v2.6/linux-2.6.15.1.tar.bz2

Por sugerencia del grupo de trabajo (MPLS-Linux, 2000) se obtuvo la versión de *kernel* 2.6.15.1 que ya ha sido probada con paquetes MPLS-Linux versión 1.950, demostrado estabilidad.

2.- Descomprimir el *kernel* y el paquete MPLS-Linux. Luego aplicar el parche. Se requieren de los paquetes **tar**, **gzip**, **bzip2** y **patch**.

 # tar jxf linux-2.6.15.1.tar.bz2

 # tar jxf mpls-linux-1.950.tar.bz2

 # cd linux-2.6.15.1

 # patch -p1 < /usr/src/mpls-linux-1.950/patches/linux-kernel.diff

3.- Configurar los módulos que se anexarán al nuevo *kernel*. Requerimos de los siguientes paquetes, compiladores y librerías: **ncurses-dev**, **libn5**, **libqt3-dev**, **libncurses5-dev**, **gcc**, **g++**, **kernel-package**. Dentro del directorio linux-2.6.15.1:

 # make menuconfig

- En "*Networking Options*" marcar las siguientes opciones :
    - <*> *Multiprotocol Label Switching*
    - <*> *MPLS: Virtual tunnel interface*
    - <*> *802.1d Ethernet Bridging*
    - <*> *Bridge: MPLS support*
- En "*Network Packet Filtering (replaces ipchains)*", seleccionamos la opción "*IP: Netfilter configuration*".
- Dentro del menú "*IP: Netfilter configuration*" marcamos las opciones:
    - <*> *Packet Filtering*
    - <*> *Packet Mangling*
    - <*> *MPLS target support*
    - <*> *DSCP target support*
- En "*Network Packet Filtering (replaces ipchains)*", seleccionamos "*Bridge: Netfilter configuration*".
- Dentro del menu "*Bridge: Netfilter configuration*", marcamos:
    - <*> *Ethernet Bridge tables (ebtables) support*
    - <*> *ebt: broute table support*
    - <*> *ebt: filter table support*
    - <*> *ebt: nat table support*
    - <*> *ebt: 802.3 filter support*
    - <*> *ebt: MPLS target support*
- En "*Networking Options*" seleccionar "*QoS and fair queueing*", desde el submenú marcar todas las opciones.

4.- Con las opciones preestablecidas en el paso numero tres, ejecutar el comando encargado de compilar el *kernel*. Este proceso puede tardar entre media hora ó una hora, dependiendo de las características del ordenador.

#make-kpkg clean

#make-kpkg --initrd --append-to-version=-custom kernel_image kernel_headers

Como resultado del comando anterior, se generarán dos paquetes con extensión .deb. Uno contendrá la imagen del *kernel* y el otro las cabeceras del mismo. Dichos paquetes pueden ser instalados en cualquier computadora que contenga el SO debian, con un *kernel* version 2.6.

5.- Instalar los paquetes generados. Primero lo hacemos con la imagen del nuevo *kernel* Linux, luego las cabeceras:

#dpkg -i [Nombre del paquete imagen]

#dpkg -i [Nombre del paquete cabecera]

6.- Por último, trasformamos los paquetes .rmp en paquetes .deb (**iptables**, **iproute** y **edtables**). Volvemos al directorio /usr/src:

 # alien iproute-2.6.11-1_mpls_1.950d.i386.rpm

 # alien --scripts iptables-1.3.0-2_mpls_1.950d.i386.rpm

 # alien --scripts ebtables-2.0.6-7_mpls_1.950d.i386.rpm

 # dpkg -i iproute_2.6.11-2_i386.deb

 # cp /bin/true /sbin/chkconfig

 # dpkg -i iptables_1.3.0-3_i386.deb

 # dpkg -i ebtables_2.0.6-8_i386.deb

7.- Reiniciar el sistema, donde se evidenciará el nuevo *kernel* compilado.

Esta guía se realizó indagando en diferentes páginas Web y foros de consulta, en especial el *site* MPLS-Linux Labs (2006).

Anexo C

Ejemplo de configuración y comandos de una red MPLS de 3 nodos

Para configurar la red mostrada en la figura 1, es necesario cargar de manera adecuada las tablas de reenvío, tal como se explica a continuación:

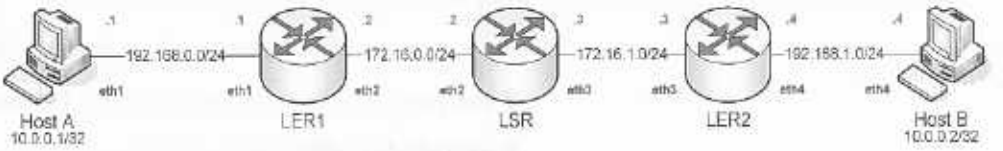

Figura 1 Ejemplo de red MPLS sencilla con 3 nodos.

1.- Tráfico desde el *Host* A, con destino al *Host* B:

Es necesario que todos los nodos y *hosts* estén configurados para trasmitir paquetes IP, esta cualidad puede ser activada haciendo uso del comando:

#echo 1 > /proc/sys/net/ipv4/ip_forward

Configuración del *Host* A:

Cargamos una ruta estática para llegar a la familia 10.0.0.2/32 a través de la interfaz eth1 del nodo LER1

 # ip route add 10.0.0.2/32 via 192.168.0.2 src 10.0.0.1

Configuración del LER1:

Creamos una entrada en la tabla NHLFE para una etiqueta de salida con el valor 1000 a través de la interfaz eth2 del LER1. Se define la dirección IP del siguiente salto.

#key1= `mpls nhlfe add key 0 instructions push gen 1000 nexthop eth2 ipv4 172.16.0.3 |grep key |cut –c 17-26`

En **key1** se almacena el valor de la llave o entrada de la tabla NHFLE, con este valor podemos atribuir el parámetro necesario al siguiente comando. El valor de **key1** debe de ser 0x2 para la primera entrada.

#ip route add 10.0.0.2/32 via 172.16.0.3 mpls Skey1

La ruta agregada en el comando anterior sabe que el enrutamiento se realizará bajo MPLS, puesto que utilizada la entrada NHFLE almacenada en **key1**.

Configuración del LSR:

Primero atribuimos la tabla ILM para la interfaz eth2 y luego cargamos el valor de etiqueta entrante 1000.

#mpls labelspace set dev eth2 labelspace 0

#mpls ilm add label gen 1000 labelspace 0

Creamos una entrada en la tabla NHLFE para una etiqueta de salida con el valor 1001 a través de la interfaz eth3 del LSR. Se define la dirección IP del siguiente salto.

#key1= `mpls nhlfe add key 0 instructions push 1001 nexthop eth3 ipv4 172.16.1.4 |grep key |cut –c 17-26`

Ahora, para conmutar la etiqueta entrante 1000 por la etiqueta 1001. Se ejecuta la siguiente línea de código:

 # mpls xc add ilm_label gen 1000 ilm_labelspace 0 nhlfe_key $key1

Configuración del LER2:

Predefinimos la recepción de etiquetas mediante la interfaz eth3, esperando recibir etiquetas 1001.

#mpls labelspace set dev eth3 labelspace 0

#mpls ilm add label gen 1001 labelspace 0

Creamos una entrada en la tabla NHLFE, esta vez el paquete de salida no tendrá etiqueta, por lo tanto no es necesario atribuir ningún valor en dicho campo. Los paquetes IP saldrán por la interfaz eth4 del LER2. Se define la dirección IP del siguiente salto.

#key1=`mpls nhlfe add key 0 instructions nexthop eth4 ipv4 192.168.1.5 |grep key |cut –c 17-26`

Para remover la etiqueta entrante 1001 y reenviar el paquete IP, se ejecuta la siguiente línea de código:

#mpls xc add ilm_label gen 1001 ilm_labelspace 0 nhlfe_key $key1

2.- Tráfico desde el *Host* B, con destino al *Host* A:

Configuración del *Host* B:

Cargamos una ruta estática para llegar a la familia 10.0.0.1/32 a través de la interfaz eth4 del nodo LER2.

#ip route add 10.0.0.1/32 via 192.168.1.4 src 10.0.0.2

Configuración del LER2:

Creamos una entrada en la tabla NHLFE para una etiqueta de salida con el valor 2000 a través de la interfaz eth3 del LER2. Se define la dirección IP del siguiente salto.

#key2=`mpls nhlfe add key 0 instructions push gen 2000 nexthop eth3 ipv4 172.16.1.3 |grep key |cut –c 17-26`

En **key2** se almacena el valor de la llave o entrada de la tabla NHFLE, con este valor podemos atribuir el parámetro necesario al siguiente comando. Puesto que **key2** es la segunda entrada en la tabla, debe tener el valor 0x3.

#ip route add 10.0.0.1/32 via 172.16.1.3 mpls $key2

La ruta agregada en el comando anterior sabe que el enrutamiento se realizará bajo MPLS, puesto que utilizada la entrada NHFLE almacenada en **key2**.

Configuración del LSR:

Primero atribuimos la tabla ILM para la interfaz eth2 y luego cargamos el valor de etiqueta entrante 2000.

#mpls labelspace set dev eth2 labelspace 0

#mpls ilm add label gen 2000 labelspace 0

Creamos una entrada en la tabla NHLFE para una etiqueta de salida con el valor 2001 a través de la interfaz eth2 del LSR. Se define la dirección IP del siguiente salto.

#key2=`mpls nhlfe add key 0 instructions push gen 2001 nexthop eth2 ipv4 172.16.0.2 |grep key |cut –c 17-26`

Ahora, para conmutar la etiqueta entrante 2000 por la etiqueta 2001, se ejecuta la siguiente línea de código:

#mpls xc add ilm_label gen 2000 ilm_labelspace 0 nhlfe_key $key2

Configuración del LER1:

Predefinimos la recepción de etiquetas mediante la interfaz eth2, esperando recibir etiquetas 2001.

#mpls labelspace set dev eth2 labelspace 0

#mpls ilm add label gen 2001 labelspace 0

Creamos una entrada en la tabla NHLFE (la cual retorna el valor 0x3), esta vez el paquete de salida no tendrá etiqueta, por lo tanto no es necesario atribuir ningún valor en dicho campo. Los paquetes IP saldrán por la interfaz eth1 del LER1. Se define la dirección IP del siguiente salto.

#key2= `mpls nhlfe add key 0 instructions nexthop eth1 ipv4 192.168.0.1 172.16.0.2 |grep key |cut –c 17-26`

Para remover la etiqueta entrante 1001 y reenviar el paquete IP se ejecuta la siguiente línea de código:

 # mpls xc add ilm_label gen 2001 ilm_labelspace 0 nhlfe_key 0x3

Note que en este último comandó no se utiliza el valor **key2**, simplemente se coloca el dato 0x3 en la línea de código. Por motivos de comodidad se utilizan variables para almacenar valores de entradas NHFLE, es posible usar tanto variables como el valor concreto de la entrada.

Anexo D

Red MPLS de doce nodos, "Zodiaco"

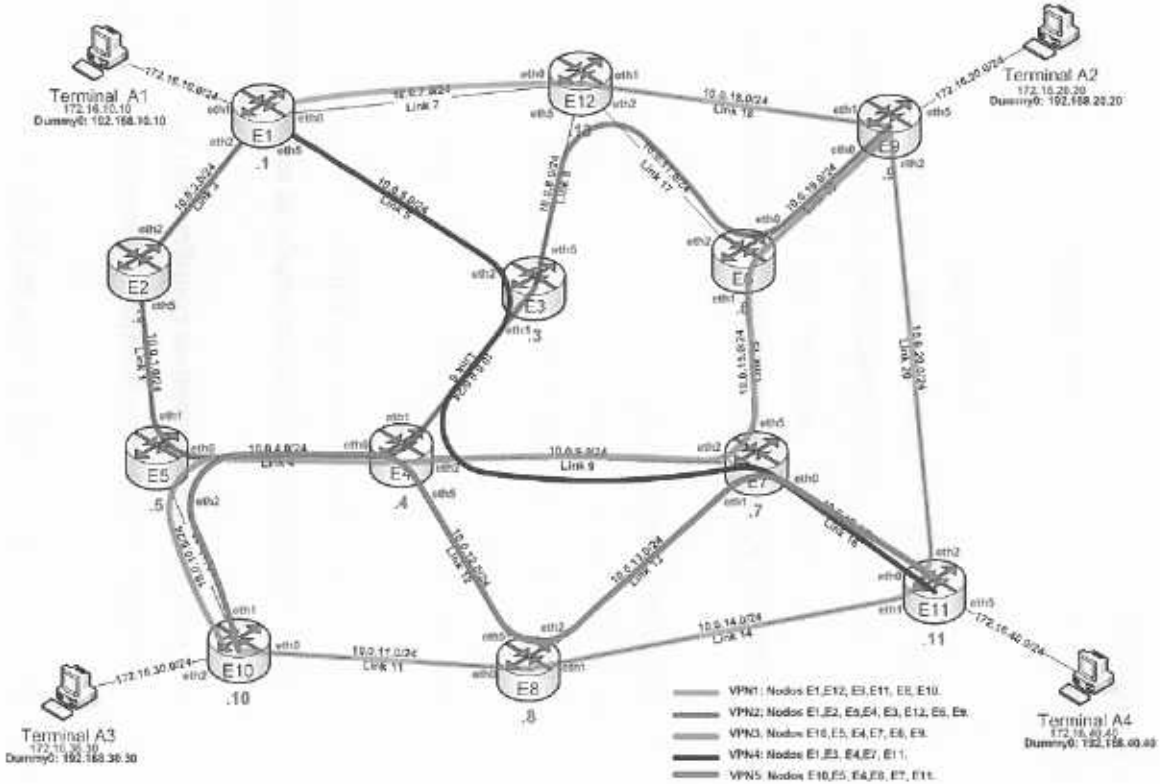

Figura 1 Red MPLS de doce nodos "Zodiaco".

Apéndice A

Configuración de los nodos en base a los experimentos sugeridos por MPLS-Linux Labs (2006)

Estos experimentos y los *scripts* necesarios para su ejecución pueden ser encontrados a través de la página Web del grupo de trabajo.

Antes de realizar cada práctica se ejecuta el *script* **del_mpls.sh** en todos los *routers*, para así borrar cualquier configuración anterior de la red.

En el Capítulo V del presente trabajo se muestran los resultados obtenidos en cada uno de los siguientes experimentos.

Primer experimento: Conmutación de Etiquetas

Este experimento estudió el manejo básico de las etiquetas en distintos puntos de enrutamiento, como por ejemplo, en enrutadores de borde (LER) y en enrutadores intermedios (LSR), ilustrando las 3 formas fundamentales de mapeo:

- Mapeo ILM: correlaciona etiquetas entrantes a un conjunto de NHLFE.
- Mapeo NHLFE: Reenvío de paquetes ya etiquetados, contiene información sobre el siguiente salto y de la pila de etiquetas.
- Mapeo FTN: correlaciona cada FEC con un conjunto de NHLFE, se utilizan para etiquetar paquetes IP.

Para llevar a cabo la configuración de red mostrada en la figura 1, se realizan los siguientes pasos:

1.- En los *host* A1 y A2, y en los nodos E2, E3 y E4 ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.
2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/2ler+more_lsr+ttl/mpls.sh** en E2, E3 y E4.

3.- Por último se realiza un **ping** a 172.16.20.20 en la Máquina Virtual A1.

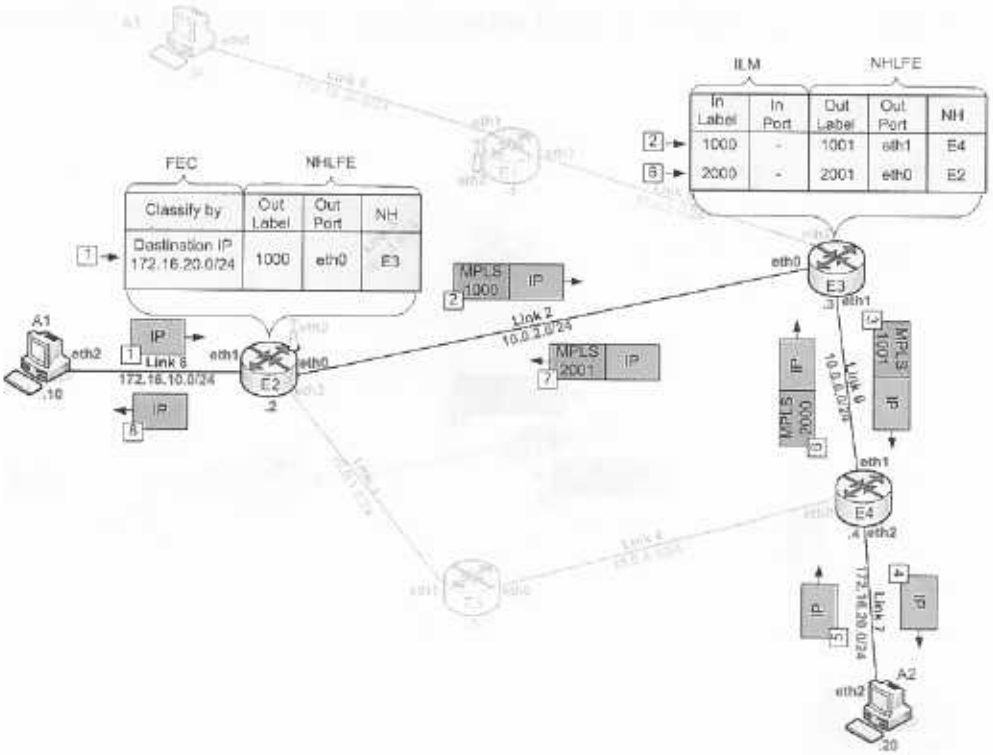

Figura 1. Conmutación de etiquetas (Dumitrascu, 2006).

Los nodos E2 y E4 están configurados como LER, mientras que el *router* E3 funciona como nodo intermedio. En el nodo E2 se crea la primera FEC, que es clasificada por la dirección de destino IP.

El paquete IP, que en este caso es un **ping** *Echo Request*, ingresa en el LER E2; dicho *router* ya tiene una FEC creada y está asociada a la dirección IP destino. Esto quiere decir que cada paquete que tenga como destino la dirección IP 172.16.20.0/24 compartirá el mismo FEC.

Segundo experimento: Espacio de etiquetas por interfaz

Un LSR tiene un repositorio de etiquetas libres con el fin de realizar asociaciones locales. Si el LSR tiene una tabla de encaminamiento por interfaz, entonces tendrá que tener un repositorio de etiquetas por interfaz.

Se planteó este experimento, en donde se demostró la factibilidad de que un LSR reciba una etiqueta, numéricamente igual, a través de múltiples interfaces.

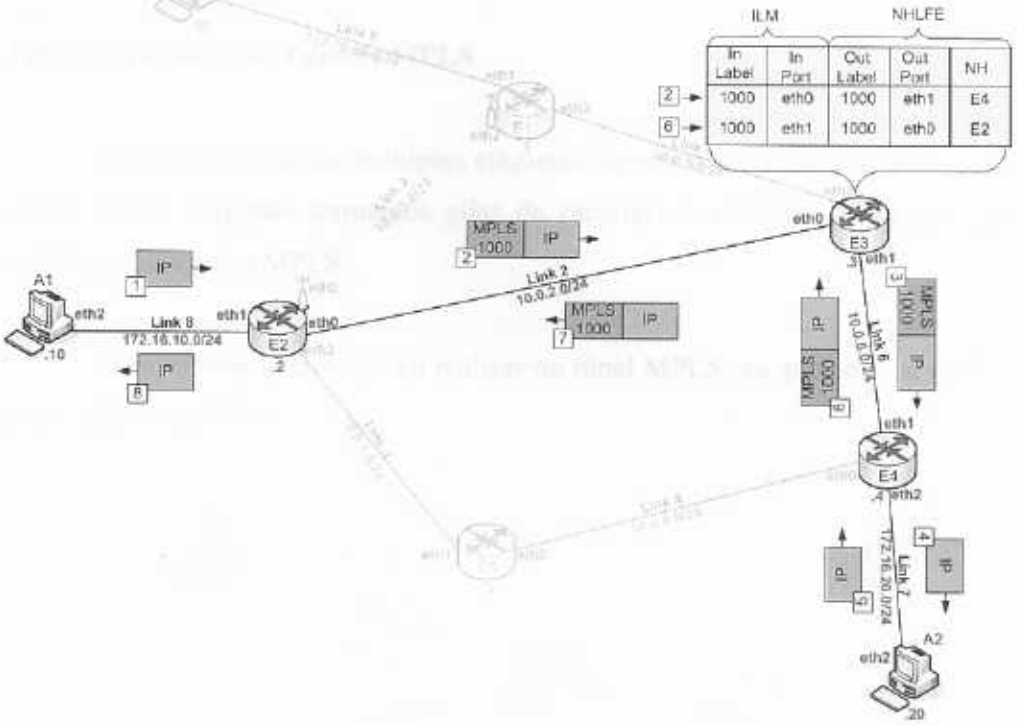

Figura 2. Espacio de etiquetas por interfaz (Dumitrascu, 2006).

Para lograr esta configuración de red se llevan a cabo los siguientes pasos:

1.- En los *host* A1 y A3, y en los nodos E2, E3 y E4 ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/labelspaces /labelspaces.sh** en E2, E3 y E4.

3.- Por último se realiza un **ping** a 172.16.30.30 en la Máquina Virtual A1.

E2 mapea los paquetes a una FEC agregando al etiqueta 1000 y los envía hacia E3, el cual posee la tabla de enrutamiento mostrada en la figura 3. Nótese que tanto las etiquetas de entrada como las de salida poseen el valor numérico 1000.

In Label	In Port	Out Label	Out Port	NH
1000	eth0	1000	eth1	E4
1000	eth1	1000	eth0	E2

Figura 3. Segundo experimento: Tabla de envío del nodo E3

Tercer experimento: Túneles MPLS

Haciendo uso de las múltiples etiquetas logramos crear túneles, y al poder anidar dichas etiquetas formamos pilas de característica LIFO, lo que permite establecer dominios MPLS.

El experimento consiste en realizar un túnel MPLS, ya que en base a ellos es posible crear VPNs.

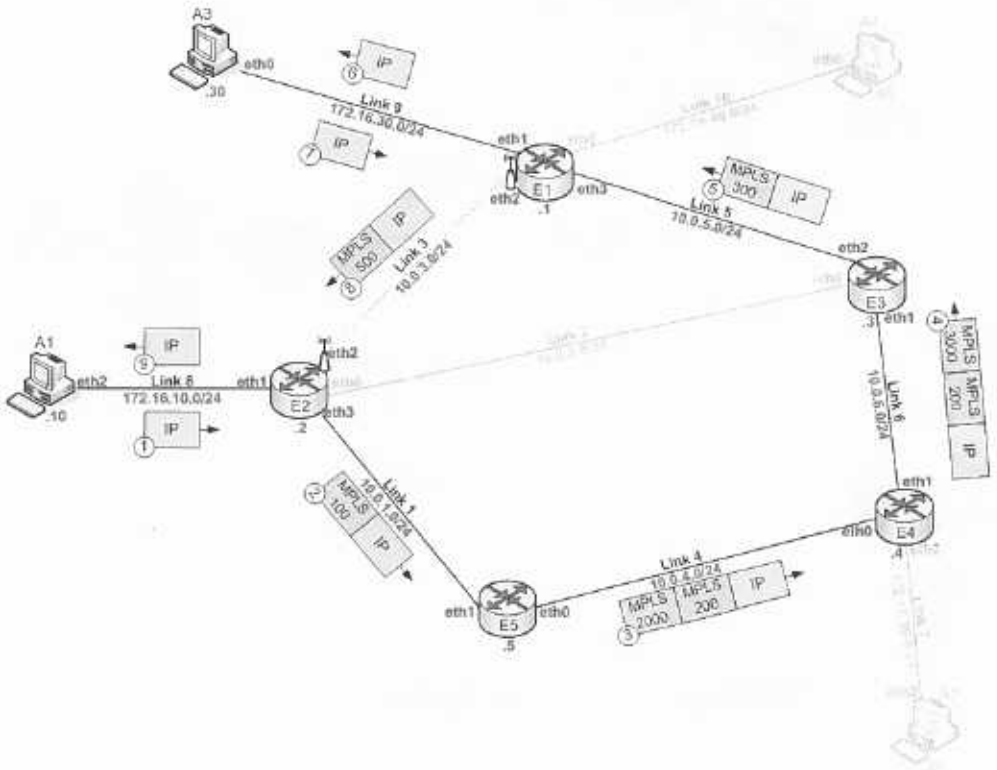

Figura 4. Túneles MPLS (Dumitrascu, 2006).

Para llevar a cabo esta configuración de red se ejecutan los siguientes pasos:

1.- En los nodos E1 hasta E5, y en A1 y A3, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/tunnels/tunnels.sh** en E1 hasta E5.

3.- Por último se realiza un **ping** a172.16.30.30 en la Máquina Virtual A1.

El sistema crea un túnel desde el *host* A1 hasta el *host* A3 y nos demuestra la posibilidad de crear un segundo dominio MPLS en la red, específicamente entre los nodos E5, E4 y E3 (doble etiquetados).

Cuarto experimento: Fusión de etiquetas

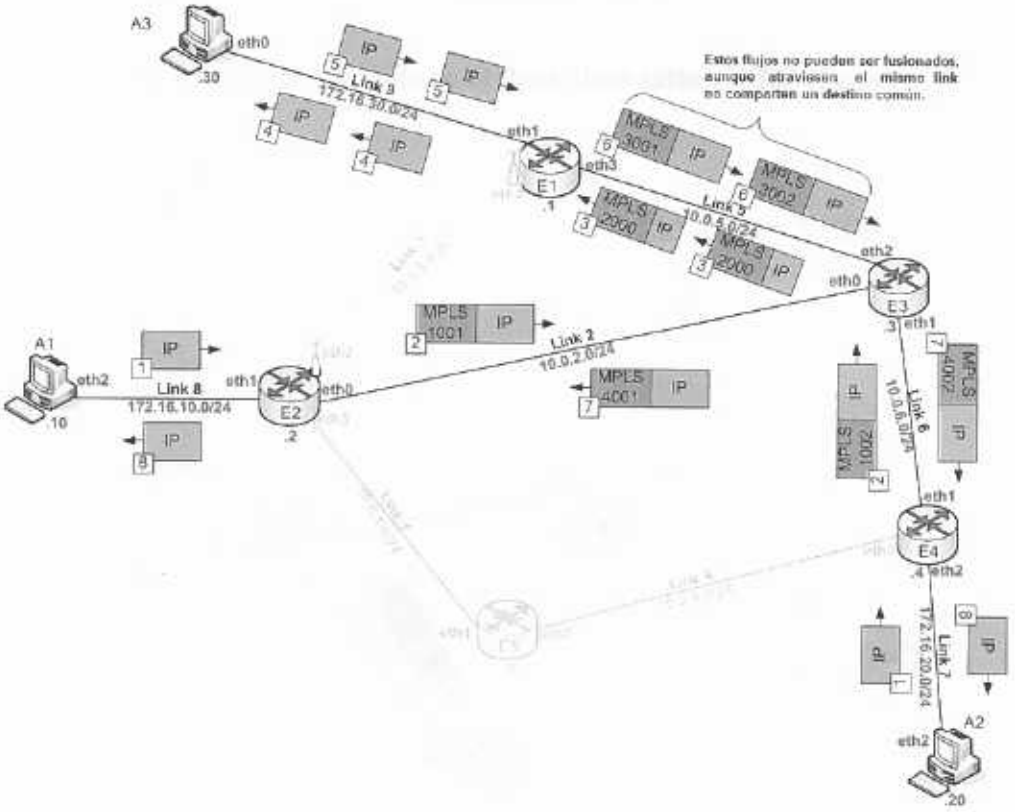

Figura 5. Fusión de etiquetas (Dumitrascu, 2006).

Experimento que demostró la factibilidad de unir dos etiquetas de entrada en una misma etiqueta de salida. Un LSR recibe paquetes por diversas interfaces de entrada y puede reenviarlos con una etiqueta de salida común, siempre y cuando el destino sea igual para todos los paquetes.

Para llevar a cabo esta configuración de red se cumplen los siguientes pasos:

1.- En A1, A2, A3, y desde los nodos E1 a E4, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/label_merging/label_merging.sh** en E1- E4.

3.- Por último se realiza un **ping** a 172.16.30.30 en la Máquina Virtual A1 y A2.

La idea de esta configuración es que entre los nodos E1 y E3 se fusionen las etiquetas con destino al *host* A3. Se realizarán dos **pings**, uno proveniente de A1 y otro proveniente de A2, ambos con destino al *host* A3.

Quinto experimento: Eliminado en el Penúltimo salto o PHP (*Penultimate Hop Popping*)

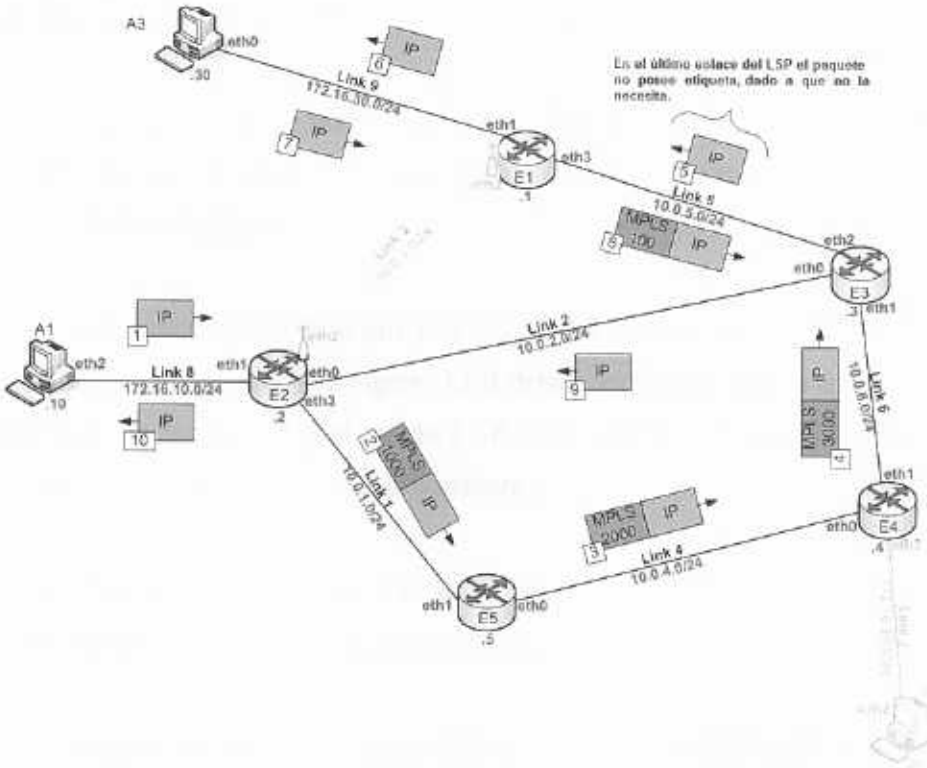

Figura 6. Eliminado de etiquetas en el penúltimo salto (Dumitrascu, 2006).

Para lograr esta configuración de red se llevan a cabo los siguientes pasos:

1.- En A1, A3, y desde el nodo E1 hasta E5, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.
2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/php/php.sh** en E1 hasta E4.
3.- Por último se realiza un **ping** a 172.16.30.30 en la Máquina Virtual A1.

Este experimento demuestra que es posible retirar la etiqueta tope del penúltimo LSR en un LSP. Hay caminos donde la etiqueta no es necesaria en el último nodo, de esta manera se simplifica la carga de procesamiento.

Se realizará un **ping** desde A1 con destino A3, el *Echo Reply* retornará hacia A1 siguiendo un camino diferente. E3 también actúa como un LER que utiliza PHP.

Sexto experimento: E-LSP

Este experimento muestra cómo los bits del campo EXP en la cabecera MPLS pueden ser utilizados para distinguir entre varias clases de servicios diferenciados (*Diffserv*).

El requerimiento para que una red MPLS ofrezca servicios diferenciados es el siguiente: los nodos de ingreso LER deben realizar un mapeo entre el campo DSCP de los paquetes IP y el campo EXP en los paquetes MPLS. El mapeo antes mencionado puede ser hecho de dos maneras:

- Usar los bits del campo EXP en la cabecera MPLS.
- Usar el valor de la etiqueta por si sola.

La primera forma de mapeo DSCP es llamada E-LSP (*EXP Infered Per-hop-behaviour Scheduling LSP*) y es la utilizada es este experimento, el cual pretende mapear una clase de servicio diferenciado valiéndose del valor en el campo EXP de la cabecera MPLS.

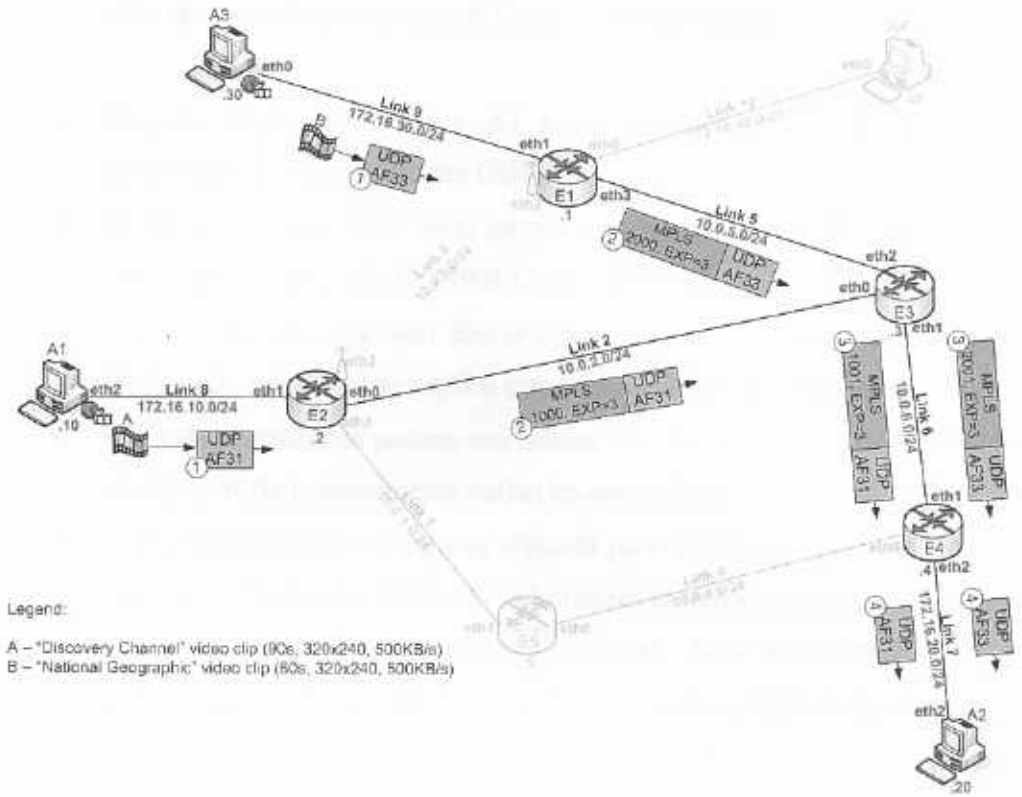

Figura 7. *Diffserv* con mapeo E-LSP, parte 1 (Dumitrascu, 2006).

Para llevar a cabo esta configuración de red, se efectúan los siguientes pasos:

1.- En los *hosts* A1 y A3, y desde el nodo E1 hasta E4, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/qos/e-lsp.sh** en A1, A2 y desde E1 hasta E4.

3.- En A2 activamos el cliente de video usando el *script* **/scripts/network2/qos/video-client.sh**

4.- En A1 y A3 activamos los servidores de video **./scripts/network2/qos/video-server.sh.** Poco después comenzará a fluir el *streaming* de video.

A fin de realizar este experimento fue vital el uso de un *Software* capaz de reproducir y recibir flujos de video, por ello se utilizó el programa **VLC** (el cual se puede descargar fácilmente con la aplicación **apt-get**).

Algunas consideraciones que debemos tener en cuenta:

- En este experimento A1 y A3 actuarán como servidores de video, generando tráfico de paquetes UDP.
- A2 juega el papel de cliente, ya que se encarga de recibir los videos a través de los puertos UDP 13901 (A1) y 13903 (A3).
- Cada puerto esta marcado con una prioridad de *Diffserv* distinta. En la sección de resultados se explica este aspecto de forma concreta.
- Los videos utilizados poseen una misma tasa de datos (500KB/s, 320x240 píxeles y 30fps), únicamente varían en contenido y duración.
- El experimento cuenta con una segunda parte en donde el cliente recibirá dos videos desde el servidor A1, el primero marcado con prioridad baja y el segundo marcado con máxima prioridad. Aquí se podrá apreciar y diferenciar la calidad entre ambos flujos, a consecuencia de las diferentes prioridades. La figura 8 muestra esta segunda parte:

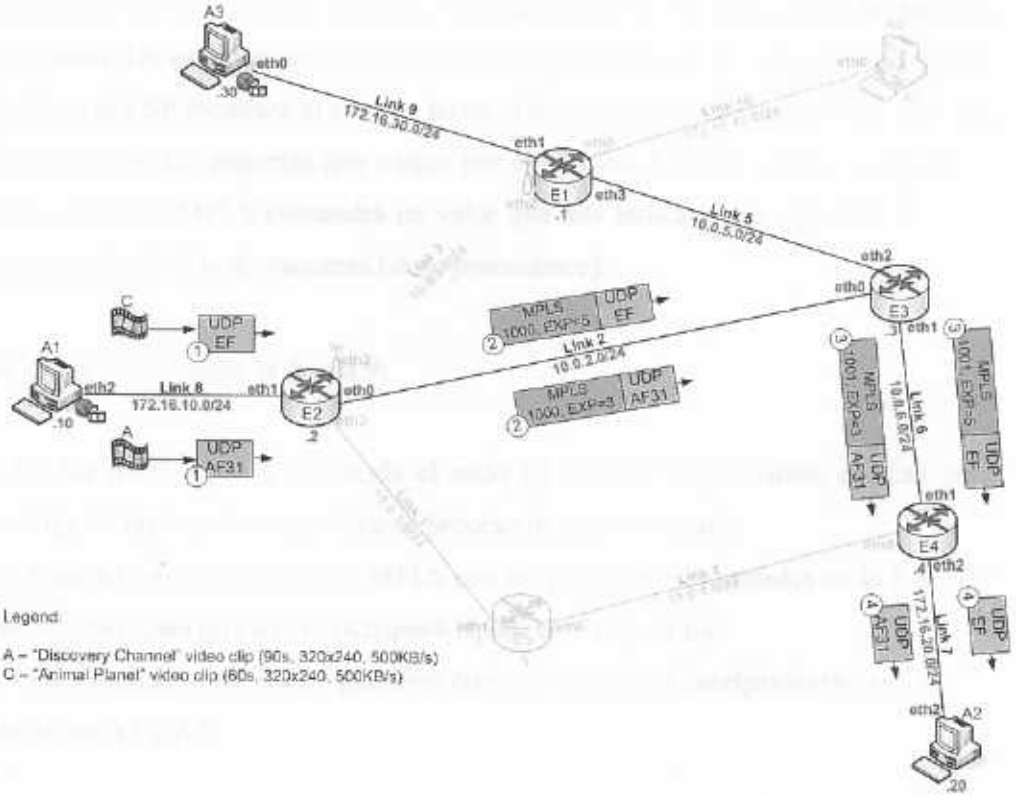

Figura 8. *Diffserv* con mapeo E-LSP, parte 2 (Dumitrascu, 2006).

Los nodos MPLS utilizan un sistema de prioridad basado en colas. De esta manera cada valor del campo EXP posee una tasa de datos de salida específica.

Así, si en un nodo no hay tráfico de alta prioridad el ancho de banda será redistribuido al tráfico existente. En el experimento se desea que el enlace 6 se congestione, por lo que el tráfico necesario sería de 1MB/s y el disponible de 900KB/s. El propósito es notar una decadencia en cuanto a la calidad de ambos videos a medida que circulan por la red.

Séptimo experimento: L-LSP

Dicho experimento muestra como diferenciar el tráfico MPLS usando el valor de las etiquetas. Esta alternativa para realizar el mapeo DSCP se denomina L-LSP (*Label Infered Per-hop-behaviour Scheduling LSP*).

Un proveedor puede ofrecer servicios diferenciados sobre MPLS utilizando el valor de la etiqueta para distinguir entre las clases de servicio existentes. De esta manera un LSP con cierta prioridad será creado para un mismo destino; el LSP mostrará el camino hasta el destino, así como también el trato que deben recibir los paquetes que viajan por el camino. En este caso el campo EXP de la cabecera MPLS contendrá un valor que nos indica la prioridad del flujo en cuanto a la pérdida de paquetes (*drop precedence*).

Se carga la red según la figura 9:

1 En los *hosts* A1 y A3, y desde el nodo E1 hasta E4, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.
2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la Figura 9 ejecutamos: **./scripts/network2/qos/l-lsp.sh** en E1 hasta E4.
3.- Para marcar el tráfico de paquetes corremos el *script* **./scripts/network2/qos/l-lsp.sh** en A1 y A3.

4.- En A2 activamos el cliente de video usando el *script* **./scripts/network2 /qos/video-client.sh**

5.- En A1 y A3 activamos los servidores de video **./scripts/network2/qos/video-server.sh.** Poco después comenzará a fluir el *streaming* de video.

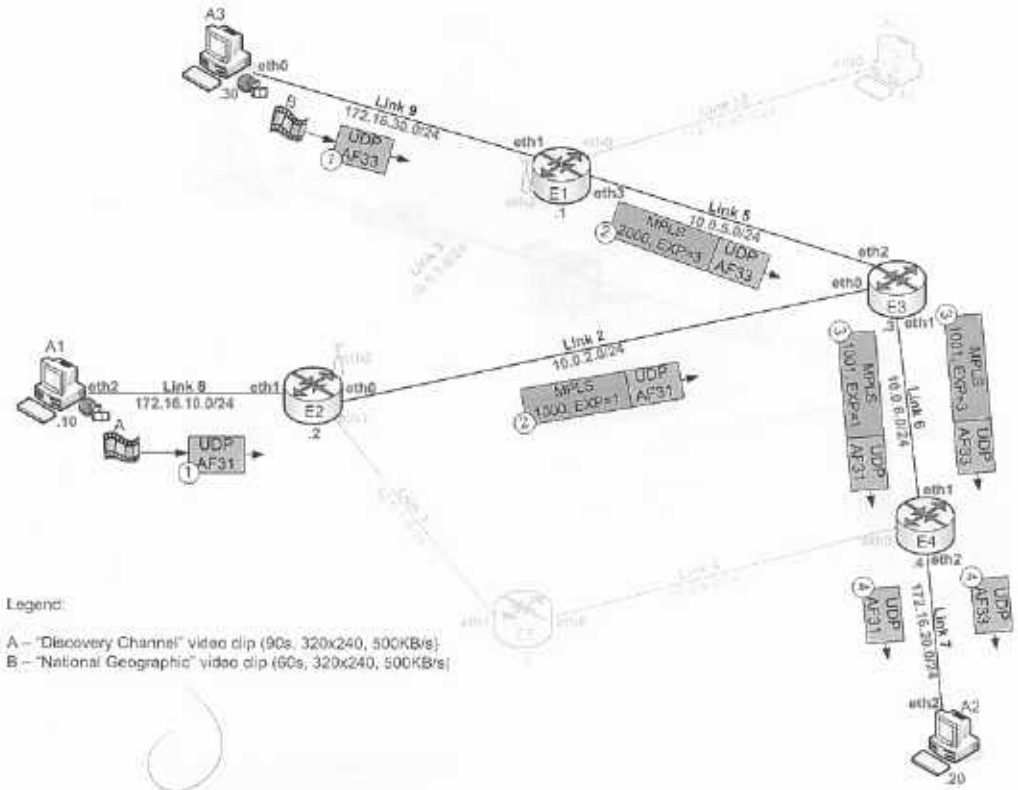

Figura 9. *Diffserv* con mapeo L-LSP, parte 1 (Dumitrascu, 2006).

El RFC2597 (1999) describe a los *Assured Forwarding* (AFx) como un medio para que el proveedor del dominio de *Diffserv* sea capaz de ofrecer niveles de garantías para la transmisión de paquetes IP. Más información en cuanto al mapeo DSCP será encontrada en la redacción teórica (Apéndice B) del presente trabajo.

La clase *Expedited Forwarding* (EF) tiene la característica de baja demora, baja pérdida y de bajo ruido. Son adecuadas para la transmisión de voz, vídeo y otros servicios en tiempo real. Al tráfico EF se suele dar prioridad estricta de gestión de colas por encima de todas las demás clases de tráfico AF.

Dicho experimento cuenta con una segunda parte en donde A1 actuará como el único servidor de video, marcando flujos de paquetes para una Clase de Servicio (CoS) AF31 y EF. La calidad de video debe ser mejor para el flujo EF, dado que posee una mayor prioridad y menos pérdidas de paquetes. La figura 10 muestra la segunda configuración de la red para este apartado.

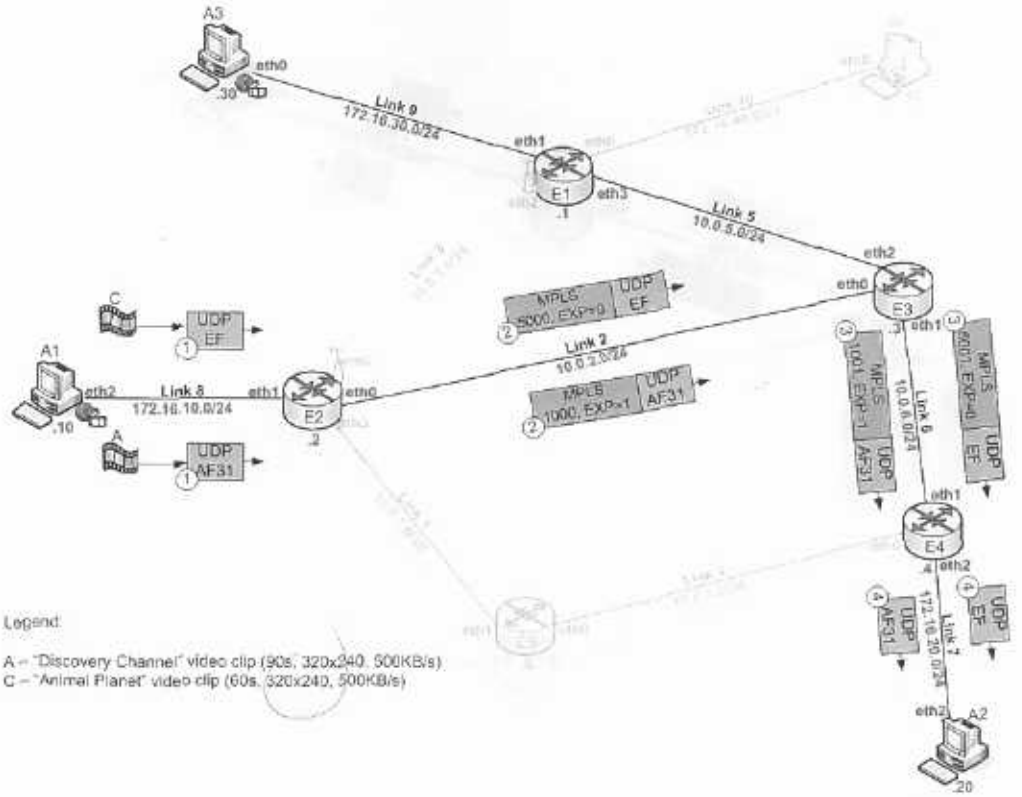

Figura 10. *Diffserv* con mapeo L-LSP, parte 2 (Dumitrascu, 2006).

Octavo experimento: Protección de Enlaces

Aquí se demuestra que en caso de que ocurra la pérdida de un enlace en una red MPLS es posible re-encaminar rápidamente el paquete, de manera que no se pierda. Esto se logra construyendo una ruta de respaldo antes que falle la ruta principal, de forma tal que al ocurrir una falla la ruta de respaldo esté disponible de inmediato.

Las rutas de respaldo pueden ser construidas de manera estática o utilizando protocolos de distribución de etiquetas (LDP o RSVP-TE). Debido a las limitaciones del *Software* MPLS-Linux, el experimento utiliza una solución estática para establecer la ruta de respaldo.

La siguiente figura muestra la configuración predeterminada de la red para el caso ideal:

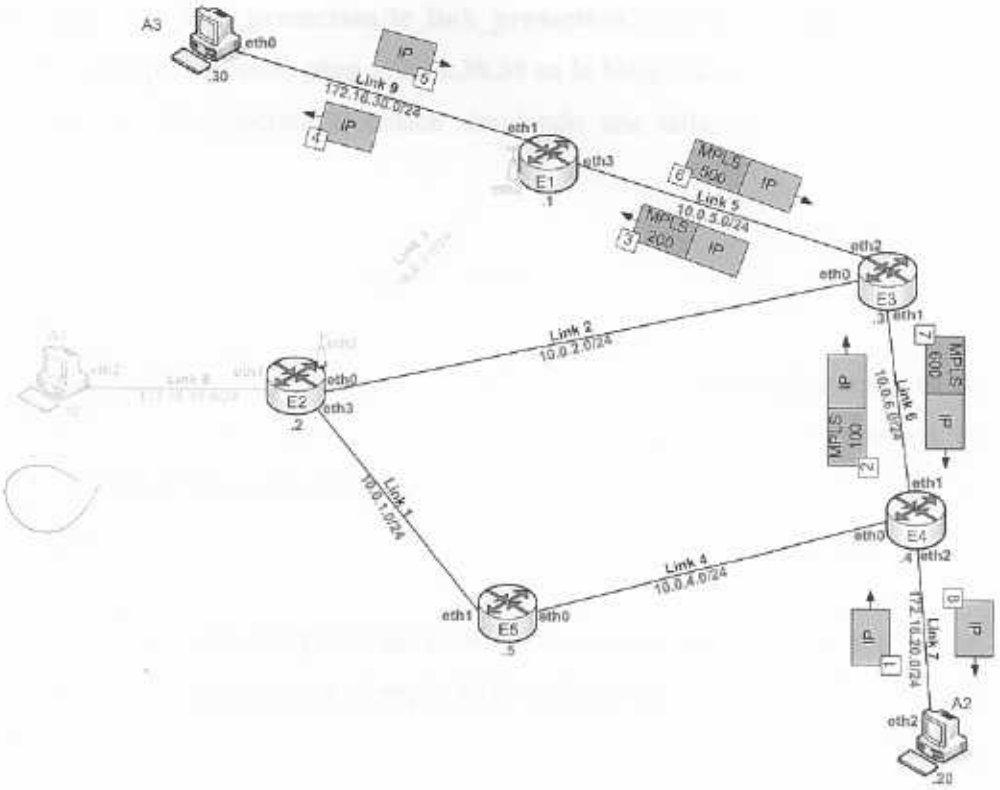

Figura 11. Ruta predefinida (Dumitrascu, 2006).

Los nodos E3 y E4 funcionan con un ciclo infinito que chequea periódicamente el enlace número seis (6). Al inicio la red está configurada para seguir el camino predefinido, el tráfico atraviesa los nodos E4, E3 y E1 para llegar a su destino.

El estado del enlace seis se chequea enviado un **ping** a dicho *link*. Tan pronto se recibe una negativa de conexión entre E3 y E4, los nodos cambiarán el

tráfico que se suponía debía atravesar el enlace seis, hacia una nueva ruta de respaldo (E4, E5, E2 y E3).

La red mostrada en la figura 11, es configurada de la siguiente manera:

1. En los *hosts* A2 y A3, y desde el nodo E1 hasta E5, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./te_link_protection/te_link_protection.sh** en E1 hasta E4.

3.- Utilizamos el comando **ping 172.16.30.30** en la Máquina Virtual A2. Luego de un tiempo se desconectará el enlace simulando una falla, allí se tomarán los resultados.

Noveno experimento: Protección de nodos

Es posible proteger los caminos que circulan por nodos vulnerables de la red cambiando rápidamente las rutas. En este experimento se re-encamina el tráfico alrededor del nodo comprometido y se vigila periódicamente el estado de dicho nodo.

En la vida real esto puede ser hecho de manera dinámica con protocolos de distribución de etiquetas, por ejemplo LDP o RSVP-TE, construyendo caminos explícitos.

Para la red mostrada en la figura 12 los nodos E1, E2 y E4 están constantemente verificando el estado del nodo E3. El nodo E3 se encarga de determinar si el enlace 6 esta arriba o se ha caído. El experimento pretende deshabilitar el nodo E3 y probar las rutas alternas. A3 actuará como un servidor de video y enviará paquetes UDP (junto con *Echo Request*) hacia el *host* A2.

La red mostrada en la figura 12, es configurada de la siguiente manera:

1.- En los *hosts* A2 y A3, y desde el nodo E1 hasta E5, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/te_node_protection/te_node_protection.sh** en E1 hasta E4.

3.- Configuramos el cliente de video en el nodo A2: **./scripts/network2/te_node_protection/video-client.sh.**

4.- Lanzamos el comando **ping 172.16.20.20** en la Máquina Virtual A3 y arrancamos el servidor de video **./scripts/network2/te_node_protection/video-server.sh.**

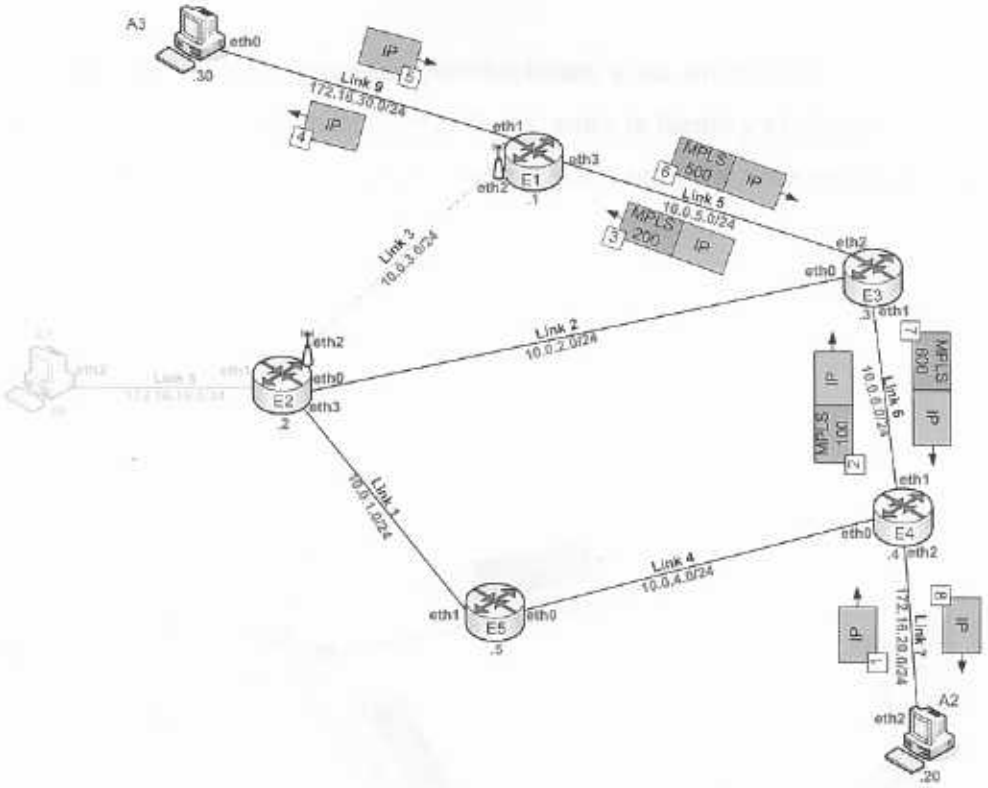

Figura 12. Protección de nodos, camino ideal (Dumitrascu, 2006).

Los nodos E1, E2 y E4 ejecutan un ciclo infinito que envía paquetes *Echo Request* hacia el nodo E3. De esta manera se monitorea al *router* para determinar una falla. El estado de E3 se chequea utilizando el comando **Hping** en lugar de **Ping** (tarda más tiempo en indicar *Time Out*).

En la sección de resultados (Capítulo V del presente trabajo) se comentan los tres diferentes estados de este experimento: un estado ideal, otro con el enlace 6 desconectado y por último una situación en donde el nodo E3 se encuentra deshabilitado.

Décimo experimento: Balanceo de Carga

Es posible separar el tráfico de paquetes IP en varios LSP hacia un destino común utilizando diferentes rutas físicas. Este experimento se enfoca en balanceo de carga para flujos de datos.

En este caso los paquetes pertenecientes a un mismo flujo siguen un mismo camino, sin embargo diferentes flujos entre la fuente y el destino pueden seguir caminos diferentes. Esto asegura un constante retraso y *jitter* entre paquetes del mismo flujo.

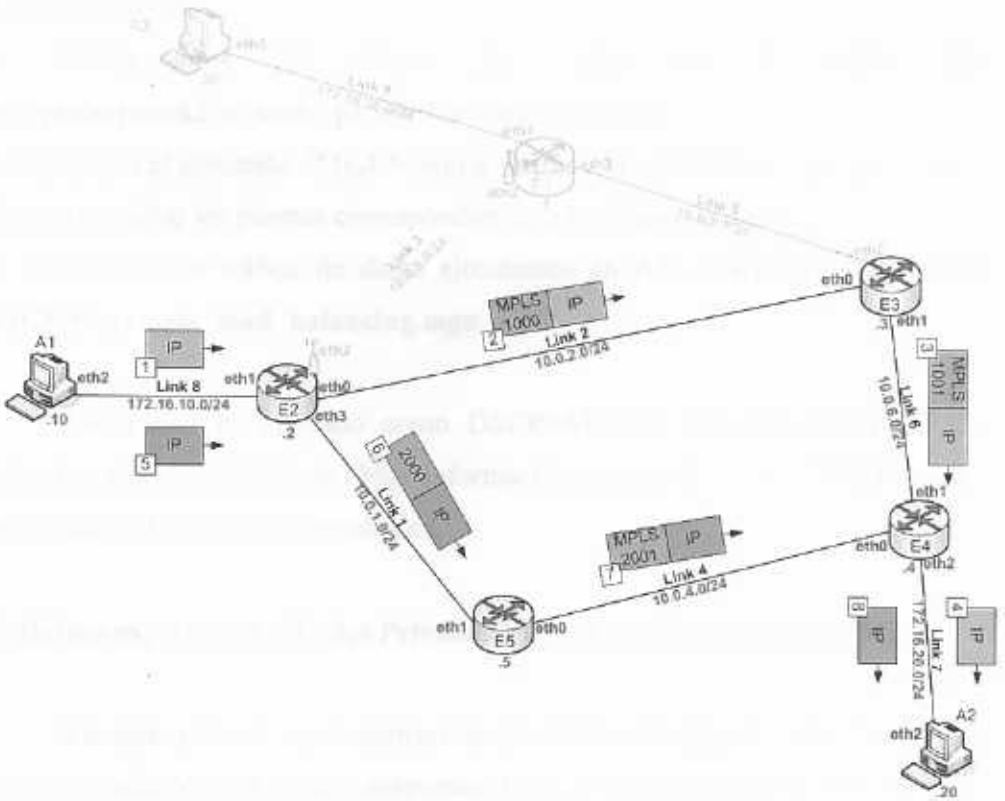

Figura 13. Balanceo de carga (Dumitrascu, 2006).

Según la figura 13 se envían dos flujos de datos desde A1 con destino A2 a través de paquetes UDP. Cada flujo estará marcado con una clase de servicio DSCP diferente, por lo tanto algunos paquetes tendrán prioridad sobre otros.

A fin de generar tráfico UDP se utiliza una herramienta llamada **MGEN.** Este *Software* es capaz de crear flujos de datos estableciendo el ancho de banda deseado, además es capaz de marcar los paquetes con diversas clases de servicio (*Naval Research Laboratory Washington DC*, 2002).

En cuanto a la red, la misma es configurada de la siguiente manera:

1.- En A2 y A3, y desde el nodo E2 hasta E5, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.
2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/load_balancing/load_balancing.sh** en A1, A2 y desde E2 hasta E4.
4.- Configuramos el cliente de video en el nodo A2: **./scripts/network2/te_node_protection/video-client.sh.**
3.- Lanzamos el comando **./MGEN/mgen -input ./MGEN/listen.mgn** en el nodo A2 para escuchar los puertos correspondientes a los flujos de datos.
5.- Para iniciar el tráfico de datos ejecutamos en A1: **./MGEN/mgen -input ./MGEN/example_load_balancing.mgn**

Un tráfico es marcado como DSCP=AF11 y otro con DSCP=AF33, teniendo prioridad el último. Mas información acerca de mapeo DSCP puede encontrase en la sección de resultados.

Undécimo experimento: Redes Privadas Virtuales nivel 2 utilizando MPLS

En esta sección se construyeron dos VPN de capa 2, específicamente Redes Privadas Virtuales que transportan datos capa 2 sobre enlaces MPLS. Esta técnica consigue emular una LAN a través de una red pública existente. El *host*

simula una comunicación con un punto remoto como si ambos estuviesen conectados a un mismo *switch*.

Para establecer la configuración de este experimento, ejecutamos los siguientes comandos y *scripts*:

1.- En los *hosts* desde A1 hasta A4, y en los nodos desde E1 hasta E4, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS con los parámetros mostrados en la figura, ejecutamos: **./scripts/network2/vpn_l2/vpn_l2.sh** desde E1 hasta E4 para configurar los LSP.

3.- Configuramos la dirección IP de A1, A2, A3 y A4: **./scripts/network2 /vpn_l2/vpn_l2.sh** . De esta forma A1 y A3 pertenecen a la misma subred, al igual que A2 pertenece a la subred de A4.

4.- Se hace un **ping** desde A1 con destino A3 (**ping 172.16.30.30**), y otro **ping** desde A2 hasta A4 (**ping 172.16.20.40**).

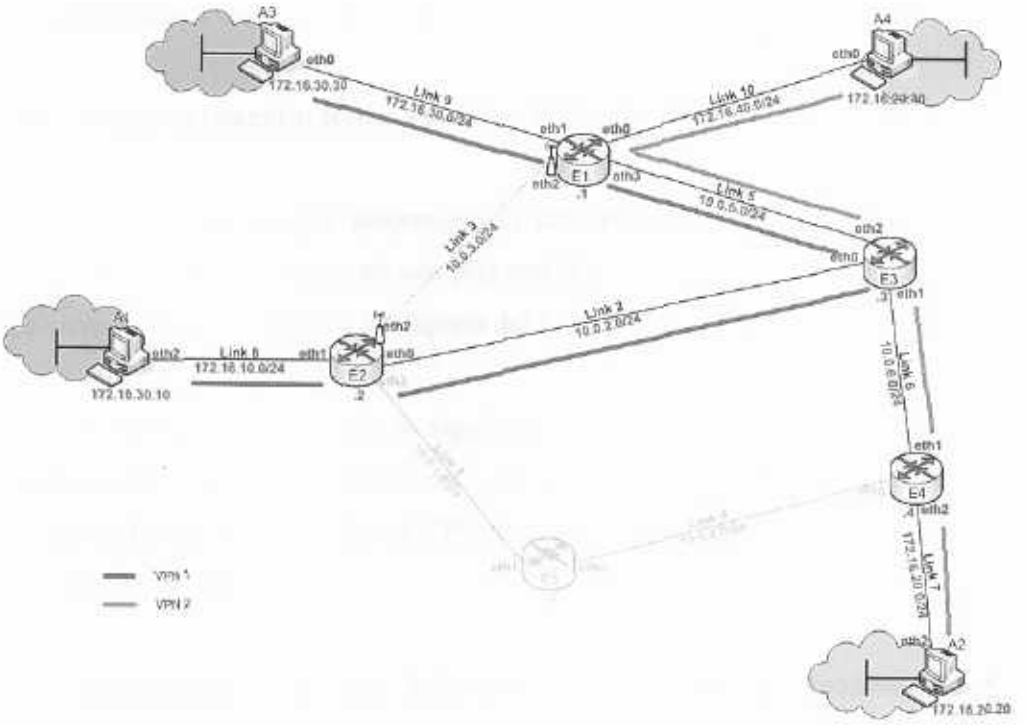

Figura 14. VPN capa 2 (Dumitrascu, 2006).

La figura 14 nos muestra el resultado de la configuración antes mencionada, en donde se presentan dos VPNs de nivel 2. La primera de ellas incluye a A1, E1, E2, E3 y A3. La segunda VPN utiliza A2, E1, E3, E4 y A4. Note que el nodo E1 pertenece a ambas VPN y además actúa como LER en ambos casos.

Es necesario resaltar que cada red VPN se encuentra asociada a una subred respectiva. En el caso de la primera VPN tenemos 172.16.30.0/24 para A1 y A3, en la segunda VPN se tiene la subred 172.16.20.0/24 para A2 y A4.

Las interfaces en dirección al cliente para los nodos E1, E2 y E4 están configuradas en modo puente y sólo realizarán operaciones de capa 2. Estas interfaces no transmitirán los paquetes recibidos a la capa 3.

Dado que estas interfaces trabajan en modo puente, ya no son accesibles a través de la red. Esto quiere decir que todo el tráfico iniciado en A1 con destino E2 será encapsulado en MPLS y reenviado hacia A3 sin ser interpretado por E2, volviendo vecinos capa 2 a los *hosts* A1 y A3.

Duodécimo experimento: Redes Privadas Virtuales nivel 3 utilizando MPLS

El propósito de este laboratorio es construir una red VPN nivel 3 con túneles MPLS. Para ello se creó una pila con dos etiquetas, en donde la etiqueta tope sirve para conmutación y la etiqueta del fondo identifica a la VPN.

El sistema mostrado en la siguiente figura incluye dos redes VPN. La primera de ellas marcada en color rojo, que involucra los nodos E1, E2, E3 y a los *hosts* A1 y A3. La segunda red VPN de color azul, incluye los *routers* E1, E3, E4 y a los *hosts* A2 y A4.

El sistema es configurado de la siguiente manera, según lo mostrado en la figura 15:

1.- En los *hosts* desde A1 hasta A4, y en los nodos desde E1 hasta E4, ejecutamos **conf.sh** para configurar las interfaces e iniciar el proceso de enrutamiento.

2.- Para activar el *Forwarding* MPLS según la figura, ejecutamos: **./scripts/network2/vpn_l3/vpn_l3.sh** desde E1-E4 (configuración de los LSP).

4.- Configuramos la dirección IP de A1, A2, A3 y A4: **./scripts/network2/vpn_l3/vpn_l3.sh.** De esta forma A1 y A3 pertenecen a la misma subred, al igual que A2 pertenece a la subred de A3.

3.- Se hace un **ping** desde A1 con destino A3 (**ping 192.168.30.30**) y otro **ping** desde A2 hasta A4 (**ping 192.168.40.40**)

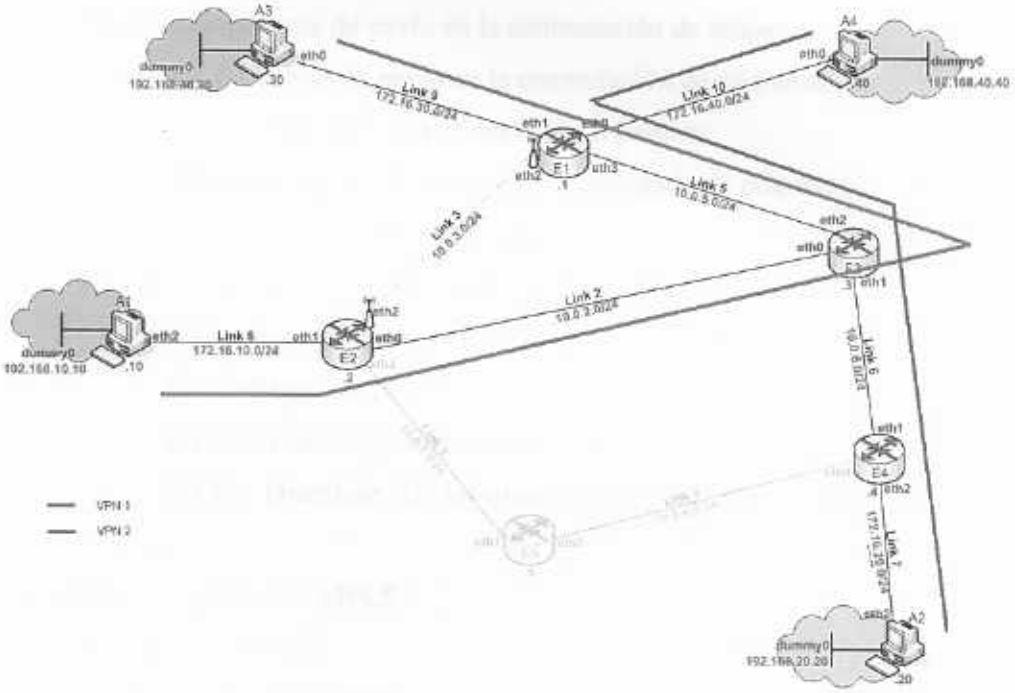

Figura 15. VPN capa 3 (Dumitrascu, 2006).

Evitando utilizar Máquinas Virtuales que emulen redes pertenecientes a clientes, simularemos una interfaz virtual llamada "dummy0" para cada una de las estaciones finales (A1, A2, A3 y A4). Dichas interfaces contarán con un rango de direcciones IP privadas clase C (192.168.0.0/24).

En la sección de resultados podremos encontrar los datos obtenidos de este y cada uno de los experimentos mencionados con anterioridad.

Apéndice B

Redacción Teórica de MPLS

Contenido:

1.- Redes de Paquetes:

En una red de paquetes los dispositivos que se encargan del direccionamiento y del enrutamiento de los mismos son los *routers*. Dichos dispositivos son considerados elementos esenciales para la operación e interconexión de las redes contemporáneas. Básicamente los encaminadores o *routers* deciden cómo y por dónde se envía la información.

Los paquetes viajan a través de la red de distintas maneras y siguiendo diversos parámetros. Es por ello que para efectos de esta investigación tomaremos en cuenta el llamado enrutamiento tradicional (que es el método con mayor implementación hasta la fecha) y un nuevo método a estudiar llamado "conmutación de etiquetas", el cual mejora de forma significativa la manera convencional de enviar paquetes.

2.- Enrutamiento Tradicional IP:

Para entender mejor como se envían los paquetes a través de la red dividamos el proceso de enrutamiento, tal como sugieren Davie y Rekhter (2000), en dos componentes básicos: control y envío.

El componente de control se encarga de crear y mantener las tablas de rutas o caminos, esto lo logra intercambiando información con todos los *routers* de la red. Dicha información es convertida en una tabla implementando uno o dos protocolos de red.

El componente de envío consiste en un conjunto de procesos y algoritmos que el *router* utiliza para tomar decisiones en cuanto a qué ruta deberá seguir el paquete. El algoritmo define la información que el paquete ha de utilizar y qué parámetros en el momento de envío serán tomados en cuenta.

En las redes IP el envío tradicional de paquetes se lleva a cabo analizando la dirección IP destino, dichas direcciones se encuentran almacenadas en las tablas de enrutamiento que poseen los encaminadores o *routers*. Cabe destacar que estas tablas se construyen en base a la información topológica de red enviada por los demás enrutadores mediante el protocolo de enrutamiento.

La dirección IP destino contenida en la cabecera de la capa de red es analizada a medida de que el paquete está viajando de salto a salto, dicho análisis se realiza de forma independiente en cada nodo dentro de la red. Enrutar de esta manera también se conoce como "*hop-by-hop destination-based unicast routing*", básicamente en este método de enrutamiento cada encaminador se concentra solamente en enviar el paquete al siguiente nodo, sin velar por todo el proceso de envío.

Convencionalmente para tomar la decisión de envío de paquetes, como ya se menciono con anterioridad, los dispositivos de enrutamiento actuales se basan en la dirección de destino *Unicast*, implementando protocolos de enrutamiento como BGP (*Border Gateway Protocol*), OSPF (*Open Shortest Path First*) o enrutamiento estático. Estos protocolos analizan la métrica y la distancia administrativa de la ruta. Si nos valemos de la información anterior, tenemos que todos los paquetes que se dirijan a un mismo destino seguirán un mismo camino, a menos que existan rutas equivalentes.

Aunque el enrutamiento IP tradicional ha tenido una implementación exitosa y bien acogida en las redes orientadas a paquetes, presenta una serie de inconvenientes en cuanto a escalabilidad y flexibilidad; es por ello que se está intentando aumentar el nivel de funcionalidad, tanto de la red como de los elementos que la conforman. Entiéndase escalabilidad como la capacidad de adaptarse a un número de usuarios cada vez mayor sin que se de una pérdida en cuanto a calidad de servicio.

Estos problemas han surgido debido al crecimiento y la evolución del Internet, así como también la demanda por un ancho de banda mayor, lo que obliga a optimizar y mejorar el desempeño de las redes IP actuales.

2.1.- Algoritmo convencional de envío:

La manera convencional de enrutar se realiza implementando múltiples algoritmos, tomando en cuenta el tipo de enrutamiento (*Unicast*, *Multicast* o *Unicast* con QoS). Un ejemplo se encuentra enunciado en la siguiente figura:

Función de Enrutamiento	Enrutamiento *Unicast*	Enrutamiento *Unicast* con Tipos de Servicio	Enrutamiento *Multicast*
Algoritmo de *Forwarding* o envío.	Correspondencia más aproximada de la dirección de destino	Correspondencia más aproximada de la dirección de destino + Correspondencia Exacta en Tipo de Servicio	Correspondencia más aproximada de la dirección de origen + Correspondencia exacta en la dirección de fuente, destino e interfaz de entrada

Figura 1 Arquitectura del enrutamiento convencional (Davie, 2000).

Para cada función de enrutamiento se implementan diversos tipos de algoritmo y cada uno de ellos tiene diversos parámetros a ajustar.

Se estudiará mas adelante que en la conmutación de etiquetas sólo se ejecuta un algoritmo a la hora de realizar el enrutamiento, dicho algoritmo se denomina "*Label Swapping*". Este algoritmo soporta un rango alto de funcionalidades a la hora de enrutar y llega a ser más conveniente que los utilizados en los *routers* tradicionales.

3.- Conmutación de etiquetas:

Una etiqueta, tal como la define Davie (2000), es simplemente un número identificador relativamente corto y con una longitud definida que se utiliza para enviar paquetes. Usualmente los dispositivos que manejan etiquetas modifican el valor de dicha etiqueta en el paquete antes de enviarla al siguiente nodo, de acuerdo al destino determinado.

La conmutación de etiquetas busca resolver problemas que sin duda alguna están relacionados. Estos son especificados a continuación:

- La necesidad de evolucionar la arquitectura actual de las redes IP.
- Necesidades en cuanto a un mayor desempeño o una mejor relación costo/desempeño de los *routers* (mientras más rápidos el costo de los *routers* aumenta).
- Escalabilidad.
- La necesidad de agregar nuevas funcionalidades al *router*.

3.1.- Clases equivalentes de envío:

A raíz del uso de etiquetas surgen las FEC (*Forwarding Equivalence Class*) o "Clases equivalentes de envío".

En la conmutación de etiquetas podemos tomar todo el juego de paquetes posibles y dividirlos en subclases. Estas subclases poseen características similares o idénticas entre sí, generalmente comparten una misma dirección de destino. De tal manera que paquetes con diferente contenido a nivel de red pueden ser mapeados como una misma entrada en la tabla de envío, describiendo una FEC en particular.

Una característica particular de una FEC es su "Granularidad", por ejemplo, una FEC puede asociar todos los paquetes cuyo destino sea el mismo,

puede crearse una FEC que incluya sólo paquetes de una aplicación determinada que se está ejecutando, también se puede asociar una FEC a un número de puerto TCP en particular. Esta Granularidad es esencial para lograr un sistema escalable.

3.2.- El componente de envío en la conmutación de etiquetas:

No solamente el sistema de enrutamiento tradicional se puede dividir en componentes de control y de envío, en el enrutamiento por conmutación de etiquetas podemos hacer lo mismo.

El algoritmo utilizado por el componente de envío para la toma de decisiones en relación al "*Forwarding*", requiere dos fuentes de información: la primera es una tabla creada por un LSR (*Label Switching Router*) o "*Router* Conmutador de etiquetas" y la segunda es la etiqueta que posee el paquete.

3.2.1.- Tablas de envío en la conmutación de etiquetas:

Conceptualmente la tabla que mantiene un LSR consiste en una secuencia de entradas, cada una de ellas compuestas con:

- Una etiqueta de llegada o entrada, asociada a un valor "N".
- Una o más subentradas, en donde se especifica: el valor de la etiqueta de salida, la interfaz de salida y la dirección del siguiente salto.

Dicha tabla está indexada por el valor contendido en la etiqueta de llegada, es decir, el valor enésimo "N" de la etiqueta entrante. La información adicional que controla el envío de los paquetes, como por ejemplo, datos de los recursos a utilizar e interfaces de salida y direcciones, serán almacenados en subdivisiones de la tabla.

Etiqueta entrante	Primera subentrada	Segunda subentrada
Etiqueta de entrada "N"	• Etiqueta de salida • Interfaz de salida • Dirección del próximo salto	• Etiqueta de salida • Interfaz de salida • Dirección del próximo salto

Figura 2. Entradas de la tabla de envío (Davie, 2000).

Cabe destacar que un LSR es capaz de manejar tantas tablas como interfaces posea, de esta manera el control de los paquetes no sólo depende de la etiqueta perteneciente al mismo, sino que además se toma en cuenta la interfaz de arribo.

3.2.2.- Insertando una etiqueta en un paquete:

Hay diversas maneras de insertar una etiqueta en un paquete. Varias tecnologías que funcionan a nivel de enlace de datos como ATM o *Frame Relay* cargan una etiqueta en la cabecera de su trama. ATM posee una etiqueta asociada en los campos pertenecientes al VCI y VPI (circuitos virtuales), de igual forma en *Frame Relay* la etiqueta se carga en el campo DLCI de la cabecera.

Utilizar etiquetas a nivel de enlace de datos es algo poco conveniente porque limita de forma significativa su conmutación y utilidad. Aplicando el mismo concepto pero a un nivel de red, y buscando que la cabecera del paquete no se vea afectada por la etiqueta, insertamos un campo entre la capa de Enlace y la cabecera de la capa de Red que contenga el valor de la etiqueta.

Cabecera de la capa de Enlace	Etiqueta	Cabecera de la capa de Red	Datos de la capa de red

Figura 3 Carga de la etiqueta entre las capas de red y enlace (Davie, 2000).

Tal como se mostró en la figura anterior, el uso de una pequeña cabecera para la etiqueta, permite implementar esta tecnología sobre otras que funcionen a nivel de enlace (tal como *Ethernet*, FDDI, *Token Ring*, enlaces punto a punto, etc.).

3.2.3.- Algoritmo de envío en el conmutado de etiquetas:

El principal algoritmo utilizado para conmutar etiquetas se basa en el "*Label Swapping*" o cambio de etiquetas. El algoritmo trabaja de la siguiente manera:

Cuando un LSR recibe un paquete, el *router* extrae la etiqueta del mismo y la utiliza como un indicador para acceder a la tabla de *Forwarding*. Una vez que el índice de entrada "N" es encontrado, es decir, el índice de la tabla y el de la etiqueta son iguales, se procede a sustituir la etiqueta del paquete por la que se encuentra especificada en la tabla. Recordemos que en cada sub-entrada de la tabla hay 3 campos (etiqueta de salida, interfaz de salida y dirección del próximo salto), la etiqueta de entrada en el paquete se sustituye por la etiqueta de salida que está especificada en la tabla. A continuación se envía el paquete por la interfaz asignada en la tabla, considerando el campo de "interfaz de salida" e implementando la dirección del próximo salto.

La descripción anterior asume que el LSR mantiene una sola tabla de envío. Para casos en donde el LSR genere una tabla por interfaz de salida, el proceso anterior tiene una pequeña variación. Luego de que el LSR reciba el paquete y de acuerdo a la interfaz por donde llegó el mismo, se selecciona la tabla de envío. Esta tabla es particular a la interfaz de entrada.

Un punto importante del conmutado de etiquetas, en contraste con otras tecnologías como ATM o *Frame Relay*, es que la información de la etiqueta no contiene especificaciones concretas del envío de los paquetes (celdas en caso de ATM y tramas en el caso de *Frame Relay*). Estos paquetes solamente poseen un

identificador "N" en particular, por lo que en la etiqueta no hay información de direccionamiento ni enrutamiento de la misma, al igual que información relacionada a reservación. Es el LSR el que se encarga de mirar la etiqueta y buscar en la tabla la información importante para el envío. El punto fuerte del conmutado de etiquetas es la simplicidad del envío. Dicha simplicidad facilita la implementación del algoritmo en el *Hardware*, aumentando la velocidad y desempeño de *Forwarding*.

Desarrollando de forma más amplia el punto anterior, tenemos que el LSR puede obtener toda la información necesaria para enviar paquetes, así como también decidir los recursos a utilizar por el mismo a través de un solo acceso a memoria. Esto se debe a que:

- Primero: una entrada en la tabla de envío posee toda la información necesaria para el transporte del paquete.
- Segundo: la etiqueta que carga el paquete provee un índice sencillo para realizar la búsqueda de los datos entre las diversas entradas de la tabla.

Entiéndase que la conmutación de etiquetas y su habilidad de trabajar bajo una amplia gama de protocolos al nivel de enlace de datos (o capa dos del modelo OSI), admite el uso de dispositivos variados que pueden funcionar como LSR. Las etiquetas pueden ser manejadas por la mayoría de los *routers* que se encuentran en el mercado, siempre y cuando se utilice el *Software* adecuado. Un *router* convencional puede evolucionar en un LSR sin realizar cambios en el *hardware*, solamente hay que programarlo mediante *Software*.

Una propiedad importante del conmutado de etiquetas es que el componente de envío solo tiene asociado un único algoritmo de *Forwarding*.

Función de Enrutamiento	Enrutamiento *Unicast*	Enrutamiento *Unicast* con Tipos de Servicio	Enrutamiento *Multicast*
Algoritmo de *Forwarding* o Envío	*Forwarding* común (*Label swapping* o Cambio de etiquetas)		

Figura 4. Arquitectura de Conmutación de etiquetas (Davie, 2000).

Si comparamos la figura anterior con la figura 3, se puede visualizar una distinción importante entre la manera convencional de enrutar y el enrutamiento realizado a través de la conmutación de etiquetas. Sólo tenemos un algoritmo común de envío en el conmutado de etiquetas, el *Label Swapping*. La eficiencia y funcionalidad de dicho algoritmo evita la necesidad de implementar otro, debido a que se ajusta a las necesidades de la conmutación de etiquetas.

3.2.4.- Capacidad multiprotocolo:

Ya con la información descrita en los apartados anteriores tenemos un par de observaciones importantes:

- El componente de envío de etiquetas no está definido en ninguna capa de Red particular, por lo que se puede utilizar de forma indiferente en conjunto con cualquier protocolo de capa tres OSI.
- Más allá de la habilidad multiprotocolo a nivel de Red, el conmutado de etiquetas opera sobre cualquier protocolo capa dos (a nivel de Enlace de Datos).

Estas dos características hacen que el conmutado de etiquetas funcione entre diversos protocolos y es por ello que el grupo de trabajo de la IETF estandarizó la tecnología bajo el nombre de: "*Multiprotocol Label Switching*" (MPLS) o Multiprotocolo de conmutación de etiquetas.

Para describir mejor este apartado se anexa la siguiente imagen que muestra de forma sencilla la localización lógica del conmutado de etiquetas entre los diversos protocolos de Red y de Enlace.

<table>
<tr><td colspan="2">IPv6</td><td colspan="2">IPv4</td><td colspan="2">IPX</td><td colspan="2">AppleTalk</td><td rowspan="2">Protocolos de Red</td></tr>
<tr><td colspan="8">"Label switching" o Conmutado de etiquetas</td></tr>
<tr><td colspan="2">Ethernet</td><td colspan="2">FDDI</td><td colspan="2">ATM</td><td colspan="2">Frame Relay</td><td rowspan="2">Protocolos de Enlace</td></tr>
</table>

Figura 5. Multiprotocolo, tanto en capa de Enlace como en capa de Red (Davie, 2000).

3.3.- El componente de control en la conmutación de etiquetas:

Una vez estudiado el componente de envío, es de suma necesidad entender cómo se distribuye la información de enrutamiento entre los diversos LSR de la red. Igualmente se debe asimilar los procedimientos que dichos equipos se ven en la necesidad de emplear para construir las tablas de *Forwarding* en donde se almacenará información de enrutamiento. El componente de control es el encargado de realizar estas acciones.

El componente de control ha de brindar una distribución consistente de la información de *routing* entre todos los LSR, asimismo ha de brindar procedimientos consistentes a la hora de construir las tablas de envío.

El componente de control en el conmutado de etiquetas también emplea los mismos protocolos que el componente de control tradicional (OSPF, BGP, PIM, etc.). De tal manera que el componente de control tradicional forma parte de la arquitectura del componente de control de etiquetas. Sin embargo, la información ofrecida tradicionalmente, no es suficiente para construir las tablas en base a etiquetas y para llenar estas deficiencias los LSR deben ser capaces de:

- Crear lazos entre etiquetas y FECs.
- Informar a los otros LSR de los lazos creados.
- Utilizar conjuntamente los dos puntos anteriores para construir y mantener la tabla de *Forwarding* utilizada por el componente de la red.

Protocolos de enrutamiento de la capa de red. (OSPF,BGP,PIM,etc.)	Procedimientos para crear lazos entre etiquetas y FECs.	Procedimientos para distribuir la información de los lazos creados.
Mantenimiento de la tabla de envío		

Figura 6. Componente de control en el conmutado de etiquetas (Davie, 2000).

Los protocolos de enrutamiento a nivel de red proveen al LSR el mapeo entre las FECs y las direcciones del siguiente salto. Los procesos para crear lazos entre FECs y etiquetas, y los procesos para distribuir esta información a través de los switches proporcionan al LSR el mapeo entre FECs y etiquetas. Nótese que es necesario realizar 2 mapeos:

- FECs al próximo salto.
- FECs a etiquetas.

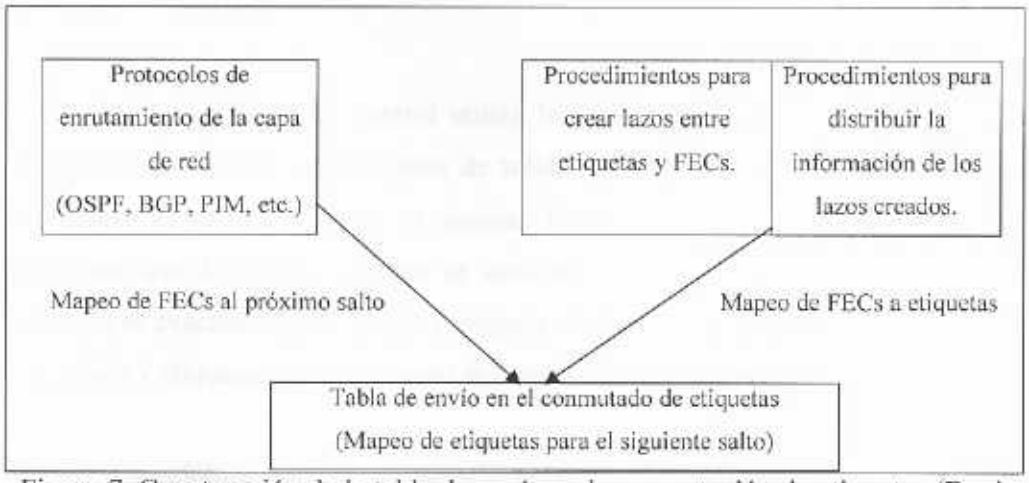

Figura 7. Construcción de la tabla de envío en la conmutación de etiquetas (Davie, 2000).

En la figura 7 se demuestra de forma esquemática la construcción de la tabla de envío o *Forwarding* en una red que conmuta etiquetas. Aquí los dos procesos de mapeo proporcionan la información necesaria para construir dicha tabla.

3.3.1.- Asociaciones locales y asociaciones remotas:

Es esencial conocer la noción de lazos o enlazado del etiquetado, básicamente consiste en asociar una etiqueta a una ruta de red o FEC, es decir, enlazar representa una asociación. En un LSR cada entrada en la tabla de *Forwarding* contiene una etiqueta de entrada y una o más etiquetas de salida. De acuerdo a esos dos tipos de etiquetas en la tabla el componente de control en la conmutación de etiquetas provee dos tipos de asociaciones.

Los dos tipos de lazos o asociaciones entre etiquetas consisten en lo siguiente: el primer tipo de enlazado ocurre cuando un LSR crea una asociación de manera local, es decir, el encaminador local establece la asociación entre la etiqueta y la FEC. Por lo tanto, la etiqueta pertenecerá al encaminador. Nos referimos a esta asociación como "local". En el segundo tipo de enlazado, el LSR recibe la información concerniente a la asociación desde otro LSR, es decir, el enlace es creado desde un *router* ajeno al de referencia. Este tipo de asociación es llamada "remota".

El componente de control utiliza lazos locales y remotos para colmar la tabla de *Forwarding* con etiquetas de salida y de llegada. Esto se realiza de dos maneras: El primer método es asociar lazos locales a etiquetas de llegada, mientras que los lazos remotos se asocian a etiquetas de salida. El segundo método es exactamente lo opuesto, etiquetas de lazo local se relacionan a etiquetas de salida y etiquetas de lazo remoto se corresponden a etiquetas de llegada.

La primera opción es llamada enlazado de etiquetas "*Downstream*", o asociación de etiquetas río abajo.

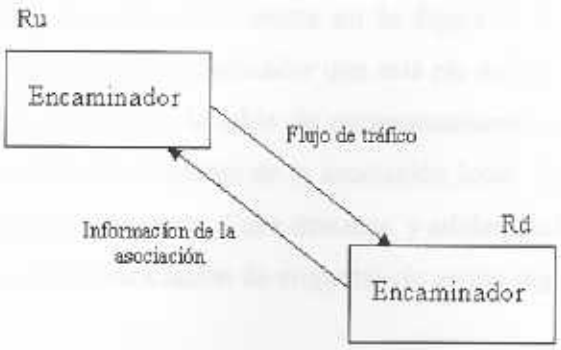

Figura 8. Asociación de etiquetas río abajo (Davie, 2000).

La asociación de la etiqueta a la FEC es realizada por el encaminador que está río abajo con respecto al flujo de paquetes. En la figura 8 el encaminado río abajo es llamado "Rd". Por lo tanto, en la tabla de encaminamiento de "Ru" tendremos como etiquetas de salida las etiquetas de la asociación remota (puesto que el LSR de referencia ha sido el encaminador que está río abajo) y como etiquetas de entrada las de la asociación local.

Por ejemplo, si "Ru" envía un paquete a Rd, este paquete habrá sido identificado con anterioridad como perteneciente a una FEC y tendrá una etiqueta "E" asociada a esa FEC. Análogamente "Ru" le habrá puesto al paquete la etiqueta de salida "E".

Igualmente existe otra técnica para asociar etiquetas y se denomina: "Asociación de etiquetas río arriba" o "*Upstream*"

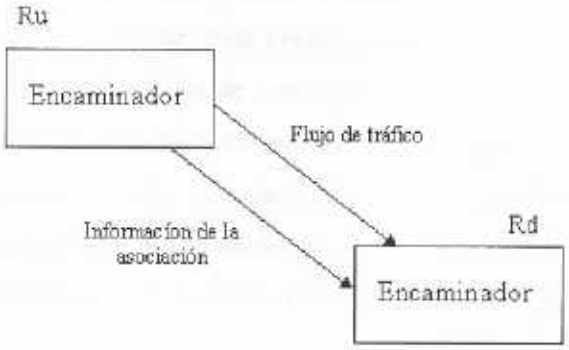

Figura 9. Asociación de etiquetas río arriba (Davie, 2000).

En este caso, tal como se muestra en la figura 9, la asociación de la etiqueta a la FEC la realiza el encaminador que está río arriba "Ru" con respecto al flujo de paquetes. Luego, en la tabla de encaminamiento de "Ru" tendremos como etiquetas de salida las etiquetas de la asociación local y como etiquetas de entrada las de la asociación remota. Cabe destacar, y adelantándonos un poco, que en MPLS sólo se utiliza la asociación de etiquetas río arriba según el RFC3031.

3.3.2.- Etiquetas libres:

Un LSR mantiene una serie o "Piscina" de etiquetas libres (etiquetas sin lazos o asociación). Al inicializar el LSR, el mismo contiene una amplia gama de etiquetas sin utilizar y que están disponibles para realizar lazos locales. La cantidad de etiquetas libres está limitada por el dispositivo LSR. Cuando el *router* crea un lazo toma una etiqueta libre y realiza la asociación; cuando un *router* destruye un lazo, retorna la etiqueta libre.

Recordemos que un LSR puede mantener una o múltiples tablas de *Forwarding* (una por interfaz). Cuando el *router* conserva una sola tabla solo mantiene un arreglo de etiquetas. Si posee más tablas, entonces tendrá una tabla por interfaz.

3.3.3.- Creación y destrucción de lazos:

Un LSR crea o destruye lazos entre una etiqueta y una FEC como resultado de un evento particular. Este evento puede ser activado por paquetes de datos enviados o por información de control procesada a través del LSR. Si la creación o destrucción de los lazos es originada por paquetes nos referimos a un enlazado de paquetes manejado por datos. Cuando la creación o destrucción del enlazado ocurre debido a la información de control, nos referimos a un enlazado manejado por información de control. Para dejar claro el punto, tomemos en cuenta que:

- La asociación de etiquetas a FECs dirigida por control se establece de antemano.
- La asociación de etiquetas a FECs dirigida por los datos ocurre dinámicamente, a medida que fluyen los paquetes.

Por ejemplo, en el enlazado de paquetes manejado por datos un lazo puede ser creado para un flujo concerniente a una aplicación apenas se detecte el primer paquete o una serie de paquetes de la misma.

Normalmente ambos tipos de asociaciones se usan conjuntamente, pero escoger uno de estos dos métodos para establecer los lazos tendrá un notorio impacto en el desempeño y escalabilidad a medida que aumenta la red. También se espera ver un efecto sobre la robustez, en el sentido de determinar que tan bien funciona el método ante diversas situaciones y condiciones.

3.3.4.- Distribuir la información de las asociaciones o lazos creados:

La arquitectura MPLS, como establece el RFC3031 (2001), define el protocolo de distribución de etiquetas como el conjunto de los procedimientos gracias a los cuales un LSR le informa a otro del significado de las etiquetas usadas para reenviar el tráfico a través de ellos.

Cada vez que un LSR crea o destruye lazos entre etiquetas locales y FECs, éste informa a los demás LSR; así los datos se propagan en toda la red como lazos remotos. Las opciones que tenemos para transportar y distribuir la información del enlazado son tan solo dos: empleando del protocolo de enrutamiento (*Piggybacking*) o utilizando el protocolo de distribución de etiquetas.

Una manera de distribuir la información de enlazado de etiquetas es llevar dicha información a cuestas sobre el tope de los protocolos de enrutamiento. Esta aproximación sólo es posible cuando los lazos se crean a partir de la información

de control, ya que se adhiere la data de enrutamiento a la data de distribución de lazos. Este procedimiento simplifica la operación del sistema en general, eliminando la necesidad de un protocolo por separado para distribuir la información de lazos.

La principal ventaja del *piggybacking* es que nunca tendremos condiciones de carrera, es decir, bajo ninguna circunstancia se dará el caso de que tengamos la asociación de la etiqueta y no tengamos la información de encaminamiento asociada (asociación entre FECs y siguientes saltos), tal como comenta Davie (2000). Otra ventaja radica en el hecho de que al estar utilizando un protocolo existente no sobrecargamos el sistema con nuevos mensajes.

El inconveniente es que no siempre resulta sencillo extender protocolos existentes, puesto que podría ser necesario modificar el formato de los mensajes y por tanto tener problemas de compatibilidad con versiones anteriores. Asimismo, existe la posibilidad de que el mensaje sea recibido por un dispositivo que no entienda el etiquetado.

El LDP (*Label Distribution Protocol*) existe en contraparte como una alternativa para distribuir la información de los lazos implementando un protocolo separado. Mientras que LDP es independiente de otros protocolos, en el primer método (*Piggybacking*) integramos ambos en uno solo.

La habilidad de soportar conmutación de etiquetas bajo una amplia gama de protocolos que no son tolerados por el método anterior es la máxima ventaja del LDP. Asimismo es posible plantear una situación en donde un LSR que contenga información de enlazado (asociación de la etiqueta a un FEC), no posee datos del enlazado para enrutamiento (asociación de un FEC con el siguiente salto) y no se completa el mapeo general.

En el apartado número 9 se ampliará el contenido del LDP, y se estudiará más o fondo su ejecución y funcionalidad.

4.- Preámbulo al protocolo MPLS:

El multiprotocolo de conmutación de etiquetas (MPLS), tal como su nombre lo indica, es un protocolo que implementa etiquetas y reduce significativamente el procesamiento de paquetes en la red, mejorando el desempeño de dichos dispositivos y el desempeño de la red en general.

En los apartados anteriores estudiamos el funcionamiento básico de las redes que conmutan etiquetas, ahora aplicaremos estos conceptos bajo el protocolo MPLS. En los siguientes puntos se mostrará de forma detallada el manejo de paquetes, etiquetas y encabezados, igualmente cómo se ejecuta el recorrido de los paquetes a lo largo de una red privada implementando MPLS.

MPLS está en desarrollo constantemente y en los últimos años la demanda de esta tecnología ha ido creciendo. Esta creciente demanda se debe principalmente a las ventajas que ofrece dicha tecnología, tales como lo son: Soporte de Calidad sobre Servicio (QoS), Ingeniería de Tráfico, soporte para Redes Privadas Virtuales (VPNs) y soporte multiprotocolo.

Antes de continuar desarrollando el contenido concerniente a MPLS necesitamos estudiar conceptos fundamentales como: QoS, Ingeniería de Tráfico, VPNs y la importancia del soporte multiprotocolo, ya que son pilares para entender las ventajas de MPLS.

4.1.- Soporte QoS:

La QoS o Calidad de Servicio, garantiza que se transmitirá cierta cantidad de datos en un tiempo dado, llamado "*throughput*". Esto permite a los proveedores de servicios garantizar a sus clientes que el retardo de extremo a extremo no excederá un nivel específico de tiempo. Además, ofrecen prioridades a las aplicaciones de extremo a extremo con una serie de reglas.

La QoS permite a los administradores de redes el uso eficiente de los recursos de dichas redes, con la ventaja de garantizar que se asignaran más recursos a aplicaciones que así lo necesiten sin arriesgar el desempeño de las demás aplicaciones. En otras palabras el uso de QoS le da al administrador un mayor control sobre su red, lo que significa menores costos y mayor satisfacción del cliente o usuario final.

Podemos decir que en los últimos años el tráfico en las redes ha aumentado considerablemente, al igual que la necesidad de transmitir cada vez más información en menos tiempo, como video y audio en tiempo real. Hay que considerar que la solución no es sólo aumentar el ancho de banda cada vez más, lo que nos lleva a aspirar administrar de forma efectiva los recursos de la red.

Como ya mencionamos, la QoS trabaja a lo largo de la red y se encarga de asignar recursos a las aplicaciones que lo requieran. Cuando hablamos de recursos, nos referimos generalmente al ancho de banda. Dichos recursos se asignan en base a prioridades, algunas aplicaciones podrán tener mayores prioridades que otras, sin embargo se garantiza que todas las aplicaciones tendrán los recursos necesarios para completar sus transacciones en un periodo de tiempo aceptable.

En resumidas cuentas la QoS otorga mayor control a los administradores sobre sus redes, mejora la interacción del usuario con el sistema y reduce costos al asignar recursos con mayor eficiencia. De igual manera, mejora el control sobre la latencia y la confiabilidad.

Si nos orientamos hacia MPLS, tenemos que éste impone un marco de trabajo orientado a conexión en un ambiente de Internet basado en IP (*Internet Protocol*) y facilita el uso y gestiones de tráfico con QoS exigentes.

4.2.- Ingeniería de Tráfico:

Según Morales (2006), la Ingeniería de Tráfico es "la habilidad de definir rutas dinámicamente y planear la asignación de recursos con base en la demanda, así como optimizar el uso de la red".

Todo tráfico entre dos puntos finales sigue la misma ruta, si utilizamos Ingeniería de Tráfico podemos equilibrar el flujo de paquetes para evitar la congestión. Morales (2006) continúa expresando que: "En MPLS la Ingeniería de Tráfico facilita la asignación de recursos en las redes para balancear la carga dependiendo de la demanda, además, proporciona diferentes niveles de soporte dependiendo de las solicitudes de tráfico de los usuarios. El protocolo IP provee una forma primitiva de Ingeniería de Tráfico al igual que el protocolo del Camino Más Corto Primero (OSPF) que permite a los enrutadores cambiar la ruta de los paquetes cuando sea necesario para balancear la carga. Sin embargo esto no es suficiente ya que este tipo de enrutamiento dinámico puede llevar a congestionar la red y no soporta QoS".

En MPLS se transmiten flujos de paquetes con su respectivo QoS y con una demanda de tráfico predecible. En caso de amenaza ante la congestión las rutas MPLS pueden ser reenviadas inteligentemente, cambiando las rutas de flujo de paquetes de forma dinámica conforme a las demandas de tráfico de cada flujo.

La Ingeniería de Tráfico es esencial para los ejes troncales o "*backbone*" de los proveedores de servicios. Dichos ejes deben soportar un uso elevado de su capacidad de transmisión.

Morales (2006) comenta que: Al utilizar MPLS las capacidades de Ingeniería de Tráfico son integradas a la capa tres (capa de Red en el modelo OSI), lo que optimiza el enrutamiento de tráfico IP gracias a las pautas establecidas por la topología y las capacidades de la troncal. También permite al "*backbone*" de la red expandirse sobre las capacidades de la Ingeniería de Tráfico.

Otro beneficio de la Ingeniería de Tráfico en MPLS es que el flujo de tráfico se encamina a lo largo de la red basándose en los recursos requeridos y en los recursos disponibles en toda la red. De igual manera MPLS emplea la ruta más corta que cumpla con los requisitos del flujo de tráfico, es decir, ancho de banda, medios de transmisión y de prioridades sobre otros flujos.

4.2.1- Funcionamiento de la Ingeniería de Tráfico en MPLS:

El objetivo básico de la Ingeniería de Tráfico, como ya se mencionó anteriormente, es adaptar los flujos de tráfico a los recursos físicos de la red. La idea es equilibrar de forma óptima la utilización de esos recursos. A comienzos de los 90, según plantea Canalis (2002), los esquemas para adaptar de forma efectiva los flujos de tráfico a la topología física de las redes IP eran bastante rudimentarios. Los flujos de tráfico siguen el camino más corto calculado por el protocolo IGP correspondiente. Tomemos en cuenta que los protocolos IGP son protocolos de pasarela a nivel interno o "*Interior Gateway Protocol*" y hacen referencia a los protocolos usados en un sistema autónomo, mientras que los "*Exterior Gateway protocol*" (EGP) hacen referencia a los protocolos usados entre dos o mas sistemas autónomos para comunicarse entre sí.

En casos de congestión de algunos enlaces el problema se resolvía añadiendo más capacidad a los mismos. La Ingeniería de Tráfico traslada ciertos flujos seleccionados por el algoritmo IGP, que se encuentran sobre enlaces más congestionados a otros enlaces más libres, aunque estén fuera de la ruta más corta o con menos saltos.

Esta acción se puede evidenciar con el siguiente ejemplo: en el esquema de la figura 10 se comparan estos dos tipos de rutas para el mismo par de nodos origen-destino.

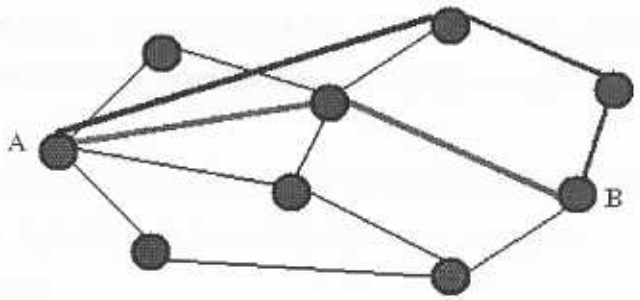

Figura 10. Comparación entre camino más corto IGP con Ingeniería de Tráfico (Elaboración propia).

El camino más corto entre A y B según la métrica normal de un protocolo IGP es el que tiene sólo dos saltos (verde), pero puede que el exceso de tráfico sobre esos enlaces conlleve a utilizar caminos alternativos así implique un salto más (azul).

El funcionamiento de la Ingeniería de Tráfico en MPLS básicamente consiste en integrar la capa dos (enlace de datos) con la capa tres (capa de Red) del modelo OSI. De esta manera el flujo de paquetes viaja a través de un túnel de datos en el *backbone* o eje troncal de la red.

El túnel de datos es creado por el Protocolo de Reserva de Recursos (RSVP); dicho protocolo crea túneles, reservando el ancho de banda para flujo de datos y así garantizar QoS en la ruta. Los parámetros que utiliza el protocolo dependen de los requisitos de recursos del túnel y de la red. Una vez realizado el túnel, el protocolo de enrutamiento interno o IGP encamina el tráfico a dichos túneles.

4.3.- Soporte Multiprotocolo:

Tal como sugiere Davie (2000), se dice que MPLS es un multiprotocolo porque engloba diversas tecnologías, y además no es necesario actualizar los *routers* IP existentes. Los *routers* MPLS, también conocidos como LSR, pueden

trabajar con *routers* IP a la par, lo que facilita la introducción de dicha tecnología a redes existentes como por ejemplo ATM y *Frame Relay*. Al igual que los *routers*, los switches MPLS pueden trabajar con switches normales.

MPLS puede operar con protocolos como son IP-Internet, ATM y Frame Relay. Brindando la ventaja de tener redes mixtas y con QoS para optimizar y expandir los recursos.

Estas características hacen de MPLS una tecnología innovadora que se puede aplicar a redes nuevas así como a redes ya existentes (debido a su alto nivel de escalabilidad).

4.4.- Soporte a Clases de Servicio (CoS):

MPLS soporta diferentes clases de servicio para cada LSP. Como caso particular puede soportar servicios diferenciados en el mismo LSP.

Tal como enuncia Davie (2000), históricamente la red Internet ha ofrecido un solo nivel de servicio: "*Best effort*". Con la aparición de aplicaciones multimedia y aplicaciones en tiempo real, surgió la necesidad de la diferenciación de servicios en Internet. De esta forma se podrán diferenciar servicios que dependen mucho más del retardo, como el correo electrónico, de otros que dependen de la variación del mismo, como el video y la voz interactiva.

El modelo de los servicios diferenciados define los mecanismos para poder clasificar el tráfico en clases de servicio con diferentes prioridades. Para clasificar el tráfico se emplea el campo ToS (*Type of Service*: Tipo de Servicio). A este campo se le llama DS en DiffServ. Una vez clasificados los paquetes en la frontera de la red, los paquetes se reenvían basándose en el campo DS. El reenvío se realiza por salto, es decir, el nodo decide por sí solo como se deberá realizar el reenvío. A este concepto se le denomina comportamiento por salto PHB (*Per-Hop Behavior*).

MPLS se adapta bien a este modelo, ya que las etiquetas MPLS tienen el campo "Exp" para poder propagar la clase de servicio CoS en el correspondiente LSP. Por tanto, una red MPLS puede transportar distintas clases de tráfico. Entre cada par de LSRs exteriores se pueden tener distintos LSPs con distintas prestaciones y distintos anchos de banda.

4.5.- Soporte para Redes Virtuales Privadas (VPN):

MPLS maneja y opera de forma eficiente las redes privadas virtuales de forma transparente al usuario, eliminando cualquier tráfico externo y protegiendo la información.

Las VPNs MPLS se basan en el uso de túneles LSP para el reenvío de los datos entre los encaminadores frontera de un proveedor de servicios. Al etiquetar los datos que entran en la VPN un LSR podrá separar los flujos VPN del resto de los datos que fluyen por la espina dorsal el *backbone* del proveedor de servicios.

Las ventajas que ofrece MPLS para VPNs IP son las siguientes, según Canalis (2002):

- Modelo de enrutamiento acoplable al existente.
- La provisión del servicio es sencilla.
- Es más fácilmente escalable.
- Se puede garantizar la QoS de los datos que entran en la VPN reservando los recursos necesarios para el túnel LSP.
- Permite aprovechar las posibilidades de la Ingeniería de Tráfico de tal forma que se pueda garantizar la respuesta global de la red (ancho de banda, retardo, etc.)
- Actualización transparente para el usuario.
- Diferenciación entre servicios.
- Reducción de costos mediante consolidación de servicios.
- Seguridad y rapidez de transmisión de información.

5.- Tecnologías VPN:

CISCO en su publicación "*MPLS Virtual Private Networks*", realizada el año 1999, plantea las principales características que hacen atractivas a las redes privadas virtuales. Dichas características son las siguientes:

- Reducen los costos de conexión entre oficinas, teleconmutadores y usuarios móviles en una intranet corporativa que opere sobre la infraestructura pública de Internet.
- Hay una mejor relación entre costo/efectividad que en las WANs privadas construidas bajo líneas dedicadas.

Sin embargo, las VPN convencionales no son muy escalables. Se basan en crear y mantener una red llena de túneles o circuitos virtuales permanentes, haciendo uso de los siguientes esquemas:

- *Frame Relay*
- Circuitos y protocolos ATM.
- Túneles tradicionales IP-IP y GRE.
- IPSec.
- L2F.
- PPTP.
- L2TP.

Las exigencias para abastecer y manejar estos esquemas basados en conexión no pueden ser soportadas en una red con cientos o miles de VPNs y *routers* asociados a la misma. Es por ello que las MPLS VPNs creadas en capa 3 carecen de conexiones, siendo más escalables y fáciles de construir que las VPN convencionales. Es mas, se pueden integrar servicios con valor agregado tal como almacenamiento de aplicaciones o datos, comercio en red y servicios telefónicos para una MPLS VPN; dado que el *backbone* del proveedor reconoce cada MPLS VPN por separado.

Las tecnologías enumeradas con anterioridad (*Frame Relay*, Circuitos y protocolos ATM, Túneles tradicionales IP-IP y GRE, IPSec, L2F, PPTP y L2TP) serán estudiadas a continuación, según la investigación realizada por Morales (2006).

5.1.- *Frame Relay*:

Para desarrollar de forma breve la información referente a *Frame Relay* contamos con el estudio de Santone (1998), en donde se comenta que: FR o *Frame Relay* es un protocolo de transporte orientado a conmutación de paquetes que comprende velocidades desde 2.4 hasta 45 Mbps. Esta tecnología nació con el fin de transmitir datos y fue evolucionando con el tiempo para soportar transporte de voz y de video.

Desde su inicio fue concebido como un protocolo para manejar interfaces de tipo Red Digital de Servicios Integrados (ISDN).

La característica que hace de *Frame Relay* una opción atractiva es su bajo costo en comparación con ATM, tomando en cuenta que depende de una buena línea de transmisión si se desea un funcionamiento eficiente. Como desventaja notable los nodos conectados a la red *Frame Relay* manejen sus propios protocolos para control de flujo, envío de acuses de recepción y recuperación de errores. Su uso es principalmente en el eje troncal de redes de datos en donde se transportan protocolos heredados como lo son:

5.1.1.- X.25:

X.25 se conoce como una interfaz estándar para conexión de terminales de datos a redes públicas. Este estándar se basa en HDLC, también conocido como el Protocolo de Enlace de Alto Nivel (HDLC) propiedad de IBM.

X.25 fue el conjunto de protocolos mejor conocido y más ampliamente utilizado de todos los protocolos basados en el modelo OSI.

5.1.2.- Arquitectura de Sistema de Red (SNA):

Protocolo desarrollado por IBM para el manejo de redes con "*Mainframes*", actualmente se considera obsoleto, sin embargo se estima que gran parte de las redes institucionales utilizan SNA como único protocolo de transporte.

5.1.3.- Comunicación Binaria Síncrona (Bisync):

Es uno de los primeros protocolos de transferencia de *frames* o tramas, en el que se transmiten tramas de bytes y no de bits como en los protocolos de la última década.

5.1.4.- HDLC:

Protocolo de comunicaciones de datos punto a punto entre dos elementos. Proporciona recuperación de errores en caso de pérdida de paquetes de datos, fallos de secuencia y otros. Mediante una red de conmutadores de paquetes conectados por líneas punto a punto y con los usuarios se constituye en la base de las redes de comunicaciones X.25.

5.2.- Circuitos ATM:

ATM o Modo de Transferencia Asíncrona es un protocolo de transporte de celdas a altas velocidades, hoy día se encuentra implementado principalmente en redes locales de compañías que requieren altas velocidades para transferencia de datos. ATM se implementa principalmente como un *backbone* en Redes de Área Amplia (WAN) y tiene facilidad de conexión a redes de alta velocidad (Como

carriers y proveedores de servicios). Los anchos de banda soportados por ATM permiten el transporte de vídeo, voz y datos.

Esta tecnología define dos velocidades de transmisión: STM-1 (155Mbps) y STM-4 (620Mbps). Actualmente esta tecnología es utilizada ampliamente, sin embargo esta siendo sustituida por medios de transmisión síncronos y ópticos.

Las celdas son enviadas con una longitud fija de 53 bytes, de los cuales 48 son la información (*payload*) y los 5 restantes son el encabezado (*header*) que es donde se lleva acabo el direccionamiento. Esta característica permite diferentes tipos de tráfico en la misma red ya que la información es transportada de una manera segura y predecible gracias a la longitud física de sus paquetes.

ATM esta basado en conmutadores, lo cual tiene sus ventajas sobre el bus de datos como son: Reservar ancho de banda, mayor ancho de banda, velocidades flexibles y procedimientos de conexión bien definidos.

5.2.1.- Arquitectura ATM:

La arquitectura ATM esta dividida en tres capas:

1. Capa de adaptación: Divide los diferentes tipos de datos en el *payload*.
2. Capa intermedia: Añade los datos de la Capa de Adaptación (OSI) con los 5 bytes del encabezado y garantiza que el paquete será enviado por la conexión adecuada.
3. Capa física: Define las características físicas del enlace entre las interfaces de red. ATM no esta ligado a un tipo de transporte físico en particular, este puede ser par trenzado, coaxial u óptico.

En el campo de las VPN's el Modo de Transferencia Asíncrona reserva Circuitos Virtuales Permanentes (PVC) con un ancho de banda determinado para

cada unos de los puntos a conectar. Los PVC son líneas virtuales punto apunto que se interconectan a través de un circuito establecido.

5.2.2.- Desventajas que presenta ATM:

Hoy en día el tráfico de Internet es IP en un gran porcentaje, el manejo de redes ATM es diferente al de IP por lo que se tienen que duplicar dichos sistemas, uno para cada uno. Esto significa mayores problemas de operación y mantenimiento (añadidos al alto costo de implementar dicha tecnología). Además, se necesita de muchos enlaces para conectar cada nodo de la red.

Desde el punto de vista económico existen tecnologías de alta velocidad que proveen alto rendimiento a precios que los productos ATM no pueden competir.

5.3.- Túneles IP:

Aparte del modelo ATM existe otro método para la transmisión y transporte de datos. Es posible crear caminos en donde los datos viajan a través de la red como si hubiera un túnel directo entre cada nodo de origen y destino. Los túneles IP aportan pocas ventajas sobre los PVC de ATM, salvo que los PVC solo funcionan con ATM y los túneles, al estar por encima del nivel físico y de enlace del modelo OSI, son independientes del medio de transmisión.

Los túneles más comunes son los de Enrutamiento de Encapsulación Genérica (GRE) y los IP sobre IP (IP-IP). Si se desea trasportar otro tipo de protocolos es necesario utilizar túneles GRE, sin embargo los dos tipos de túneles son inseguros en cuanto a protección de información.

La desventaja en este tipo de arquitectura es que se utilizan concentradores de túneles para manejarlos, dichos equipos son muy costosos y complejos. Otra desventaja es que el tráfico implementando túneles no es monitoreado por los

routers, con lo que se pierde la información del *header* IP en donde se manejan las políticas QoS.

5.4.- IP Security (IPSec):

La característica mas importante de IpSec es la posibilidad de encriptar los datos transmitidos, esta cualidad hace de este protocolo una opción muy atractiva y de rápida difusión en el mundo empresarial.

Algunas de las desventajas principales del protocolo Ipsec radican principalmente en que es un protocolo complejo, con muchas características y opciones. Esto hace que la configuración sea complicada, además se requiere configuración en el equipo del cliente.

5.5.- Reenvío de capa dos (L2F):

Este protocolo fue desarrollado para establecer túneles de tráfico desde usuarios remotos hasta sus sedes. L2F no depende de IP, por lo que es capaz de trabajar directamente con otros medios como *Frame Relay* y ATM.

L2F utiliza protocolos de autenticación, principalmente el Protocolo Punto a Punto (PPP) para la autenticación del usuario remoto. Hay dos niveles de autenticación del usuario, primero por parte del Proveedor de Servicio de Internet (ISP) anterior al establecimiento del túnel, y por otra parte cuando se ha establecido la conexión con la puerta de enlace (*gateway*) corporativa. Como L2F es un protocolo de Nivel de Enlace de Datos, o capa dos del modelo OSI, ofrece a los usuarios flexibilidad para manejar protocolos distintos a IP, como IPX o NetBEUI.

5.6.- Protocolo de Túneles punto a punto (PPTP):

En la actualidad no se utiliza y fue sustituido por L2TP e IPSec. Este protocolo era vulnerable en cuanto a seguridad y no se podía utilizar donde la privacidad de los datos fuese importante.

Estos fallos en el PPTP fueron causados principalmente por errores de diseño en la criptografía del protocolo de Autenticación y por las limitaciones de longitud de clave.

5.7.- Protocolo de Túneles de Capa dos (L2TP):

Fue creado para corregir las deficiencias de L2F y PPTP. Esta tecnología utiliza PPP para proporcionar autenticación, definiendo su propio protocolo para el establecimiento de túneles basado en L2F. El transporte de L2TP está definido para una gran variedad de tipos de paquete, incluyendo X.25, *Frame Relay* y ATM.

Los túneles L2TP se crean encapsulando tramas L2TP en paquetes de tipo Protocolo de Datagramas de Usuario (UDP). Dicho protocolo encapsula la información en un paquete IP, por lo que las direcciones de origen y destino definen los extremos del túnel.

6.- Tecnologías predecesoras al MPLS:

Esta breve historia describe la evolución de las tecnologías predecesoras al conmutado de etiquetas, y cómo el desenvolvimiento de los estudios llevo a la unificación del protocolo MPLS.

Se requiere destacar que el presente resumen fue obtenido a través de las investigaciones expuestas por Davie y Rekhter (2000) en su libro "*MPLS: Technology and Aplications*".

6.1.- IP sobre ATM:

La arquitectura ATM difiere significativamente de la arquitectura IP. La primera es una arquitectura orientada a conexión, mientras que la segunda no lo es. Además el esquema de direcciones es totalmente diferente, al igual que lo es el modelo de comunicación *Multicast*.

Dada la necesidad existente de transportar paquetes IP sobre redes ATM han surgido grupos de trabajo del IETF. Entre los mas importantes esta el grupo de trabajo IP sobre ATM (RFC1483) y el IP clásico sobre ATM y ATMARP (RFC1577).

El primero de ellos trata sobre cómo encapsular datagramas IP en enlaces ATM. En el modelo clásico o tradicional los nodos IP pueden comunicarse entre sí, siempre y cuando pertenecen a la misma subred IP lógica (LIS, Logical IP Subnet). Una LIS es simplemente un conjunto de *hosts* y encaminadores conectados a través de una red ATM. Dichos *hosts* y encaminadores están en una red IP, por lo que comparten la misma dirección de subred.

Si un dispositivo en una LIS se desea comunicar con otro que esté en otra LIS distinta debe hacerlo a través de un encaminador que conectado entres ambas LISs (no podrán establecer un único circuito virtual para comunicarse).

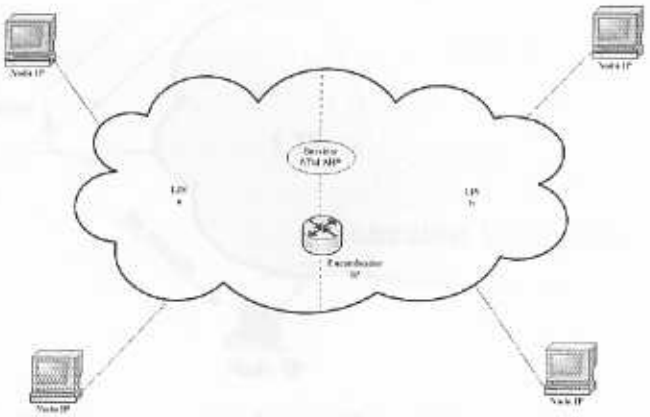

Figura 11. Comunicación entre dos LIS (Elaboración propia).

Para que dos dispositivos que están en la misma LIS se puedan comunicar es necesario que conozcan sus direcciones ATM, haciendo uso de un servidor ARP (*Address Resolution Protocol*). En redes convencionales, como las redes *Ethernet*, los dispositivos IP aprenden sus respectivas direcciones gracias al protocolo ARP que se apoya en la difusión del nivel de enlace. Como las redes

ATM carecen de esto se necesita el mencionado servidor para hacer la conversión de direcciones IP a direcciones ATM. Los dispositivos registran en dicho servidor su dirección ATM y su dirección IP, de tal forma que cuando un dispositivo se quiere comunicar con otro que está en la misma LIS le solicita al servidor que haga la traducción de la dirección IP a la dirección ATM. Con la dirección ATM ya podrá comunicarse estableciendo un circuito virtual.

La RFC1577 no aborda el tema de que dos dispositivos que se encuentren en distinta LIS puedan establecer un circuito virtual para comunicarse. Para esto surgió el grupo del IETF ROLC (*Routing Over Large Clouds*) y se definió el protocolo NHRP (*Next Hop Resolution Protocol*, Protocolo de Resolución del Siguiente Salto) para resolver el problema del salto a través del encaminador que conecta ambas LISs. Esto se consigue teniendo servidores NHS (*Next Hop Servers*), de tal forma que un dispositivo en una LIS puede aprender la dirección ATM de otro dispositivo de otra LIS a través del mencionado servidor. Con la dirección ATM ya pueden establecer un circuito virtual.

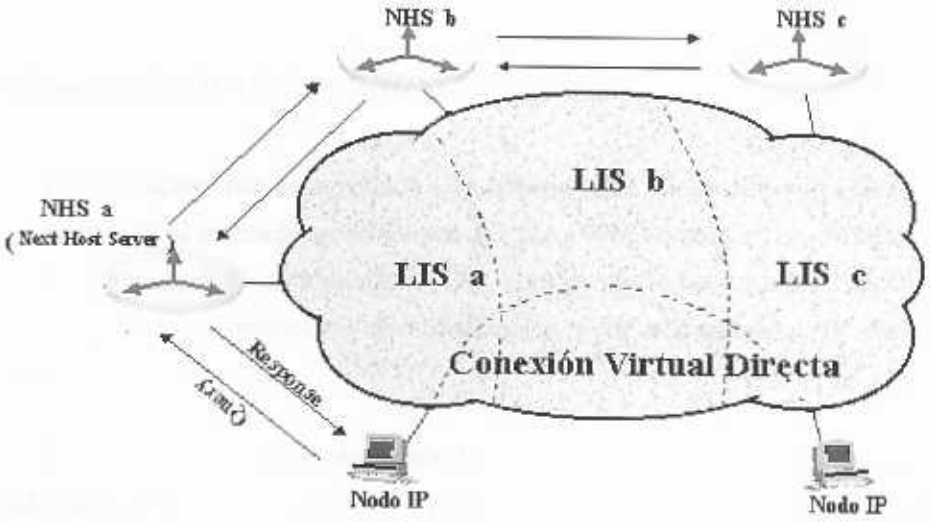

Figura 12. ATM implementando servidores NHS (Elaboración propia según Davie, 2000).

La arquitectura MPOA, *Multiprotocol Over ATM* (Multiprotocolo a través de ATM), del ATM Forum, contempla la integración de IP con ATM mediante emulación de LAN versión 2 y NHRP.

6.2.- Cell Switching *Router* (CSR) de Toshiba:

La idea del CSR fue concebida y desarrollada por Toshiba y presentada al IETF en 1994. Su utilización comercial se centró en redes académicas de Japón.

Esta solución fue una de las primeras propuestas que trataba de utilizar los protocolos de encaminamiento del mundo IP para controlar conmutadores ATM, y básicamente fue diseñada para conectar subredes IP utilizando una aproximación clásica de "IP sobre ATM", es decir, RFC1483.

En este caso los distintos conmutadores de etiquetas se comunican utilizando circuitos virtuales típicos de ATM, y las etiquetas son asignadas basándose en las características de los flujos de datos que se deben conmutar.

Un nuevo protocolo, denominado *Flow Attribute Notification Protocol* (FANP), es el responsable de identificar los VCs (circuitos virtuales) entre los nodos CSR. Asimismo se utiliza este protocolo para establecer la asociación entre los flujos de datos individuales y los VCs dedicados.

6.3.- Conmutación IP (*IP SWITCHING*) de Ipsilon:

Esta solución fue desarrollada por Ipsilon (que luego fue adquirida por Nokia) y lanzada al mercado a comienzos del año 1996. Se basa en un dispositivo que realiza funciones de conmutador ATM, eliminando todas aquellas funciones relacionadas con los protocolos de señalización y de encaminador IP de una manera sencilla y eficiente.

Los dispositivos de conmutación IP utilizan los distintos flujos de tráfico para el establecimiento de etiquetas (que en este caso son cabeceras ATM). El funcionamiento de estos dispositivos puede describirse resumidamente de la siguiente forma: Un dispositivo de conmutación IP funciona como un encaminador normal hasta que detecta que existe una cierta cantidad de tráfico dirigida hacia un destino concreto. Una vez detectada esta situación, establece un VC ATM para este flujo de datos concreto.

Para realizar correctamente estas funciones se definieron dos nuevos protocolos, uno destinado a establecer la relación entre los flujos de datos y las etiquetas, denominado *Ipsilon Flow Management Protocol* (IFMP); y otro para gestionar las funciones del conmutador ATM y controlar el establecimiento de los CV a través de él, conocido como *General Switch Management Protocol* (GSMP).

6.4.- Conmutación de etiquetas (*Tag Switching*) de CISCO:

La solución desarrollada por Cisco para la conmutación de etiquetas fue bautizada como "*Tag Switching*". Esta solución, a diferencia de las comentadas anteriormente, se basa en el establecimiento de "caminos virtuales" entre los extremos de la red sin que existan flujos de datos que estimulen o dirijan el establecimiento de estos caminos virtuales, es decir, estos caminos son establecidos por necesidades de control de la red antes de que existan los flujos de datos que los utilicen.

Básicamente una red de conmutación de etiquetas consiste en un conjunto de encaminadores frontera (*Tag Edge Routers*) encargados de añadir a la entrada y eliminar a la salida la información (*tag*) de encaminamiento interno, y un conjunto de encaminadores internos, denominados *Tag Switching Routers*, encargados de conmutar y encaminar los flujo de datos basándose en la etiqueta o "*tag*" añadida a la entrada.

El esfuerzo de normalización que empezó Cisco con la conmutación de etiquetas culminó en el grupo de trabajo MPLS (*Multiprotocol Label Switching*) del IETF y hoy día MPLS se utiliza como un término genérico para referirse a la conmutación de etiquetas.

6.5.- *Agregate route-based IP switching* (ARIS) de IBM:

Otro de los gigantes de la industria, IBM, desarrolló su propia solución en el entorno de la conmutación de etiquetas. Esta solución, conocida como ARIS, es conceptualmente similar a la solución de Cisco anteriormente descrita. En este caso los caminos, y por tanto las etiquetas asociadas, son establecidos como

respuesta a las acciones de control del tráfico. Los encaminadores que soportan esta tecnología son conocidos como "*Integrated Switch Routers*" (ISR) en la terminología IBM.

La idea que subyacía a la hora de diseñar ARIS fue la utilización de ATM como nivel de enlace; por lo que los protocolos propios de ARIS son protocolos "*peer-to-peer*" o entre iguales, que se establecen entre los ISR implicados directamente a nivel IP y permiten establecer conexiones con los vecinos e intercambiar las correspondientes etiquetas asociadas a los distintos flujos de datos. Este mecanismo de distribución de etiquetas comienza en el extremo donde finaliza el flujo de datos en la red ARIS, también conocido como "*Egress Router*", y es propagado de forma ordenada hasta el ISR que comenzó el flujo.

7.- Arquitectura de MPLS:

Una vez vistos los conceptos fundamentales de la conmutación de etiquetas veamos los beneficios de MPLS. La información desarrollada a continuación se basan en los estudios realizados por Davie y Rekhter (2000), en su publicación: *MPLS Technology and Aplications*.

Davie (2000) comenta que en MPLS la asignación de un paquete a una FEC se realiza cuando el paquete entra en la red asignándole a dicho paquete una etiqueta. En los siguientes saltos sólo se usará la etiqueta para determinar la interfaz por donde reenviar el paquete, por lo que no será necesario analizar la cabecera del nivel de red. La etiqueta se usa como índice en la tabla de encaminamiento donde se obtiene el siguiente salto y la nueva etiqueta con la que sustituir la anterior. Hay que recordar que las etiquetas son locales a los encaminadores. En MPLS los conmutadores pueden realizar el reenvío, pero estos no tienen necesidad de analizar las cabeceras del nivel de red. Al basar el reenvío en las etiquetas en vez de en la cabecera del nivel de red obtenemos las siguientes ventajas:

- Dado que un paquete se asigna a una FEC cuando entra en la red, el encaminador frontera que encapsula el paquete podrá usar toda la información que tenga sobre el paquete, incluso información que no esté

en la cabecera del nivel de red. Por ejemplo podrá usar información del nivel de transporte, como los números de puerto, para asignar paquetes a FECs. Por tanto, gran parte del trabajo se realiza antes de que el tráfico entre en la red. Con el encaminamiento convencional sólo se puede examinar la cabecera del nivel de red.

- Un paquete que entra en la red por un determinado encaminador puede etiquetarse de distinta forma que si hubiera entrado por otro. Por ende, se pueden tomar decisiones dependientes del encaminador frontera que encapsula el paquete. Esto no se puede hacer en el encaminamiento convencional porque la identidad del encaminador frontera que introdujo el paquete en la red no viaja con el paquete. Más adelante se toca con mayor detalle el tema del encaminador frontera.
- Se podría forzar a un paquete a seguir una ruta elegida explícitamente antes o en el momento que el paquete entre en la red, en lugar de elegirse por el algoritmo dinámico de encaminamiento a medida que el paquete fluye por la red. Esto podría hacerse para permitir la Ingeniería de Tráfico.

En el encaminamiento convencional el paquete tendría que llevar la información de la ruta (encaminamiento fuente). En MPLS se puede usar una etiqueta para representar la ruta, de tal forma que el paquete no tiene por qué llevar la información de la ruta.

Algunos encaminadores analizan la cabecera del nivel de red para determinar la clase de servicio a la que pertenece el paquete, así como para determinar el siguiente salto. Con la información de la clase de servicio el encaminador podrá o no aplicar alguna disciplina planificada a los paquetes. MPLS permite, pero no impone, que la clase de servicio se infiera total o parcialmente de la etiqueta. Así podremos decir que una etiqueta representa la combinación de una FEC y una clase de servicio.

Como vimos en apartados anteriores, MPLS recibe su nombre porque soporta cualquier protocolo de nivel de red, así como cualquiera de nivel de enlace. Por tanto, MPLS puede o no usar tecnologías subyacentes de *backbone* como ATM, Frame Relay, SDH y DWDM.

Para comprender la arquitectura y funcionamiento de MPLS se presenta la siguiente terminología. Estos términos son extraídos del RFC3031:

- Fusión de etiquetas (*Label Merging*): Reemplazo de múltiples etiquetas de entrada para una FEC particular por una sola etiqueta de salida.
- Salto de conmutación de etiquetas (*Label Switched Hop*): salto entre dos nodos MPLS en los que el reenvío se hace usando etiquetas.
- Pila de etiquetas (*Label Stack*): conjunto ordenado de etiquetas.
- Punto de fusión (*Merge Point*): nodo en el que se realiza la fusión de etiquetas.
- Fusión de circuitos virtuales (*VC Merge*): fusión de etiquetas en donde la etiqueta MPLS se transporta en el campo ATM VPI/VCI. De esta forma se permite que múltiples circuitos virtuales se fusionen en un único circuito virtual.
- Fusión de caminos virtuales (*VP merge*): fusión de etiquetas en donde la etiqueta MPLS se transporta en el campo ATM VPI. De esta forma se permite que múltiples caminos virtuales se fusionen en uno sólo. Dos células con el mismo valor VCI se han originado en el mismo nodo.

7.1.- Nodos MPLS:

Los LSRs frontera, también conocidos como LER, son los encargados de etiquetar los paquetes que entran en la red. Para poder realizar este trabajo estos LSRs deben implementar el componente de control y el componente de reenvío tanto del encaminamiento convencional como de la conmutación de etiquetas.

Si un paquete entra en la red MPLS el encaminador frontera utilizará el componente de reenvío de la conmutación de etiquetas para determinar la etiqueta a colocar en el paquete. Si el siguiente salto no es un LSR y el paquete no tiene etiqueta, entonces el LSR deberá reenviar el paquete usando el componente de reenvío del encaminamiento convencional.

Cuando el paquete va a salir de la red MPLS, el LER que recibe el paquete le quitará la etiqueta y lo reenviará al siguiente salto usando el componente de reenvío del encaminamiento convencional. Dicho LSR sabrá que

el paquete quiere abandonar la red simplemente porque el siguiente salto no es un LSR.

Basándonos en el RFC3031, podemos definir los siguientes tipos de nodos MPLS:

- LER (*Label Edge Router*): dispositivo de ingreso o egreso a una red MPLS.
- LSR de entrada (*Ingress LSR*): dispositivo LER, es un LSR que recibe tráfico de usuario (por ejemplo datagramas IP) y lo clasifica en su correspondiente FEC. Genera una cabecera MPLS asignándole una etiqueta y encapsula el paquete junto a la cabecera MPLS, obteniendo una PDU MPLS (PDU = Protocol Data Unit o unidad de datos del protocolo).
- LSR de salida (*Egress LSR*): dispositivo LER, es un LSR que realiza la operación inversa al de entrada, es decir, desencapsula el paquete removiendo la cabecera MPLS.
- LSR intermedio o interior: LSR que realiza el intercambio de etiquetas examinando exclusivamente la cabecera MPLS (obteniendo la etiqueta para poder realizar la búsqueda en la tabla de encaminamiento).

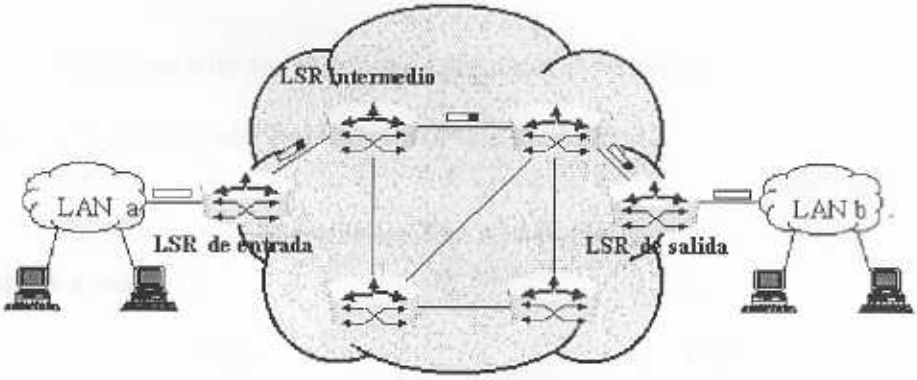

Figura 13. Distintos tipos de nodos en una red MPLS (Davie, 2000).

7.2.- Protocolos de distribución de etiquetas:

Un protocolo de distribución de etiquetas, como ya vimos en apartados anteriores, es un conjunto de procedimientos por los que un LSR le informa a otro de las asociaciones de etiquetas a FECs que ha hecho.

A dos LSRs, que utilizan un protocolo de distribución de etiquetas para intercambiar información de asociaciones de etiquetas a FECs, se les conoce como un par de distribución de etiquetas (*Label Distribution Peers*) respecto a la información de las asociaciones que intercambian.

MPLS no asume que haya sólo un protocolo de distribución de etiquetas. De hecho, se están normalizando distintos protocolos de distribución de etiquetas. Tal como el LDP (*Label Distribution Protocol*) que fue mencionado con anterioridad.

En MPLS la decisión correspondiente a la asignación de una etiqueta a una FEC la realiza el LSR que está río abajo (*downstream*) con respecto a la asociación.

El LSR que está río abajo informa al LSR que está río arriba de la asociación. Por tanto, las etiquetas se asignan o asocian río abajo y se distribuyen desde el LSR que está río abajo al LSR que está río arriba.

MPLS permite variaciones en la asociación río abajo:

7.2.1.- Río Abajo Solicitado (*Downstream-on-Demand*):

Un LSR le solicita explícitamente a su siguiente salto la asociación de una etiqueta a una FEC.

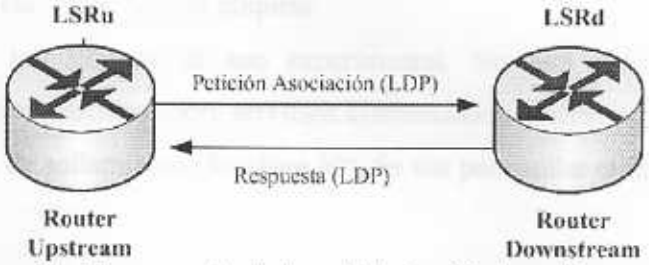

Figura 14. .Esquema río abajo solicitado (Elaboración propia).

7.2.2.- Río Abajo no Solicitado (*Unsolicited-Downstream*):

Un LSR distribuye asociaciones de etiquetas a LSRs que no lo han solicitado explícitamente.

Figura 15. Esquema río abajo no solicitado (Elaboración propia).

Estas aproximaciones se pueden usar por separado o conjuntamente. En caso de usarlas conjuntamente en una adyacencia de distribución de etiquetas (es decir, cuando tenemos dos LSRs que son pares de distribución de etiquetas) ambos LSRs se tendrán que poner de acuerdo en la técnica a usar.

7.3.- Formato de las Etiquetas:

Una etiqueta MPLS tiene 32 bits y se sitúa entre la cabecera de nivel 2 y la de nivel 3.

<u>Etiqueta Genérica (Para redes sin campo de etiquetas: PPP o LAN)</u>

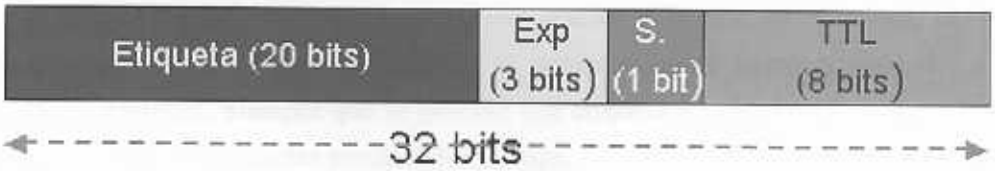

Figura 16. Formato y campos de una etiqueta (Davie, 2000).

- Etiqueta: valor N de la etiqueta
- Exp: inicialmente de uso experimental. No está definido totalmente. Algunos artículos sobre servicios diferenciados (DiffServ) discuten su uso.
- S: bit de apilamiento (*Stacking bit*). Se usa para apilar etiquetas.

- TTL: tiempo de vida (*Time To Live*). Número de nodos (saltos) que puede atravesar el paquete MPLS. Se necesita porque los LSRs intermedios no analizan el campo IP TTL.

7.4.- Pila de etiquetas:

En MPLS un paquete puede tener más de una etiqueta, organizadas a modo de pila. A esto se le conoce como pila de etiquetas.

El procesado de etiquetas en MPLS siempre se basa en la etiqueta superior, por lo que en cualquier LSR se puede añadir (*push*) o remover (*pop*) una etiqueta. Esto nos da la ventaja de añadir rutas parciales dentro de la red a un LSP existente, creando así túneles.

Al principio de cada túnel los LSR asignan la misma etiqueta a los paquetes que van entrando, esto lo hacen mediante la operación *push* que mencionamos anteriormente. Al final de cada túnel pasa lo inverso, el LSR de salida remueve la etiqueta superior (añadida a la entrada del túnel) para mostrar la etiqueta original con el fin de que siga su trayectoria original. Esta operación se puede realizar indefinidamente formando así una red de túneles dentro de cada LSP original

Aunque MPLS soporte una jerarquía gracias a la pila de etiquetas, el procesamiento de un paquete etiquetado es completamente independiente del nivel de la jerarquía. Siempre que se procese una etiqueta ésta será la de la cima, sin importar cuántas etiquetas pueda haber debajo.

Se puede considerar a un paquete no etiquetado como un paquete con una pila de etiquetas vacía.

Si la profundidad de la pila de etiquetas de un paquete es "m", a la etiqueta que está al fondo de la pila se le llama etiqueta de nivel uno (1), a la que está encima etiqueta de nivel dos (2), y así sucesivamente según el RFC3031.

En la siguiente figura tenemos tres dominios. Supongamos que el dominio 2 es un dominio de tránsito, es decir que en dicho dominio no se originan paquetes y tampoco hay paquetes destinados a él. Para anunciar las direcciones

del dominio 3 el LSR F le distribuye la información al LSR E, seguidamente el LSR E le distribuye la información al LSR B, el cual se la distribuye al LSR A. No se distribuye la información a los LSRs C y D porque son LSRs interiores.

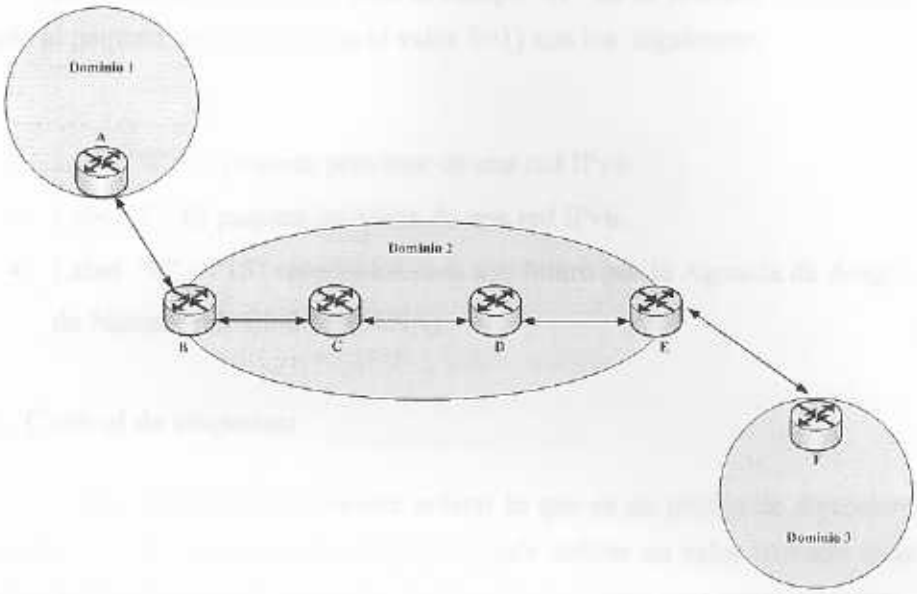

Figura 17. Niveles de etiquetado (Elaboración propia).

Se usan dos niveles de etiquetas. Cuando el tráfico entra en el segundo dominio se apila una nueva etiqueta en la cima de la pila, por lo que las etiquetas que hubiera en la pila descienden un nivel.

Recordemos el campo S en el formato de las etiquetas, en donde está contenida la información del orden en la pila. Cuando S= 1 indica que es la última etiqueta y que al salir quedará vacía la pila, esto generalmente ocurre en el *router* de salida. Cuando es S= 0 indica que por lo menos hay otra etiqueta en la pila.

Es muy importante considerar que cuando un LER saca el último encabezado MPLS del paquete éste debe mandar la información (*payload*) fuera de la nube MPLS al destino contenido en el encabezado IP, previamente obtenido por el *router* de entrada. La importancia de esto radica en que los *routers* MPLS no cuentan con tablas de búsqueda de etiquetas.

Cuando a un LSR llega un paquete cuyo campo S posee un valor S=1 en el encabezado MPLS, se sabe que el siguiente campo es el encabezado de red y

que debe usarlo para reenviar el paquete conforme al mecanismo de ese tipo de red.

Los valores reservados para el campo "N" de la primera etiqueta que se añade al paquete (la etiqueta con el valor S=1) son los siguientes:

- Label "0": El paquete proviene de una red IPv4.
- Label"2": El paquete proviene de una red IPv6.
- Label "4" – "15" reservados para uso futuro por la Agencia de Asignación de Números de Internet (IANA).

7.5.- Control de etiquetas:

Antes de continuar conviene aclarar lo que es un prefijo de dirección. En lugar de utilizar la máscara de subred se puede utilizar un valor llamado valor de prefijo. El valor de prefijo describe cuántos bits se deben usar como máscara.

En MPLS existen dos formas para asignar etiquetas a FECs: independiente y ordenado.

7.5.1.- Control independiente de etiquetas:

Cuando un LSR reconoce una FEC realizará una asociación de forma independiente de una etiqueta a esa FEC. Una vez hecho esto informará de dicha asociación a los LSRs vecinos.

Esta es la forma de trabajar en el encaminamiento IP convencional: cada nodo encamina los paquetes de forma independiente, apoyándose en el hecho de que el algoritmo de encaminamiento converge rápidamente, garantizando de esta forma que los datagramas son entregados de forma correcta. Esto lo podemos ver en el ejemplo mostrado a continuación:

En la Figura 18 el LSR A utiliza OSPF para informarle al LSR C el prefijo de dirección 192.165/16.

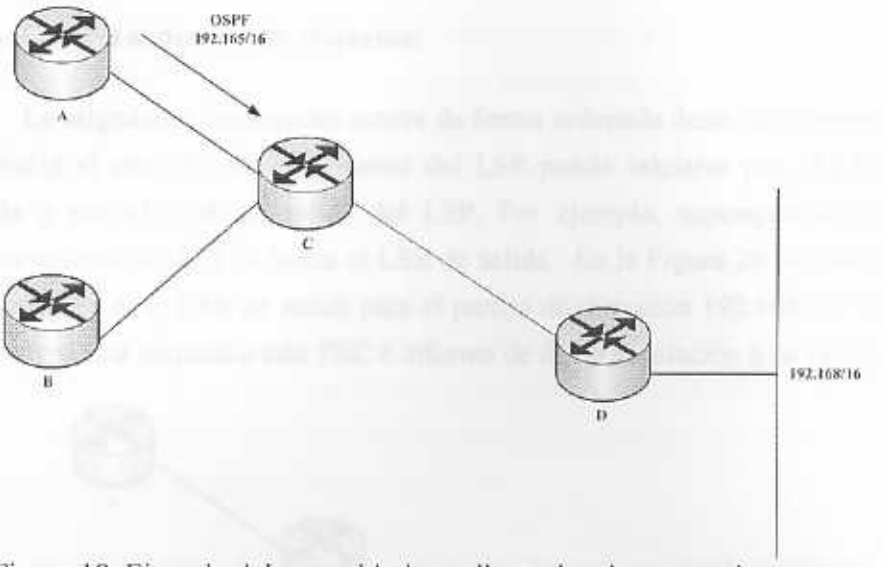

Figura 18. Ejemplo del control independiente de etiquetas, primera parte (Elaboración propia)

Luego, en la figura 19, cuando C recibe el prefijo asigna de forma independiente una etiqueta a esta FEC e informa de dicha asociación a los LSRs vecinos.

Un inconveniente del control independiente ocurre cuando dos vecinos no están de acuerdo en las FECs que van a usar. Cuando esto ocurre algunas FECs no tendrán LSPs asociadas a ellas.

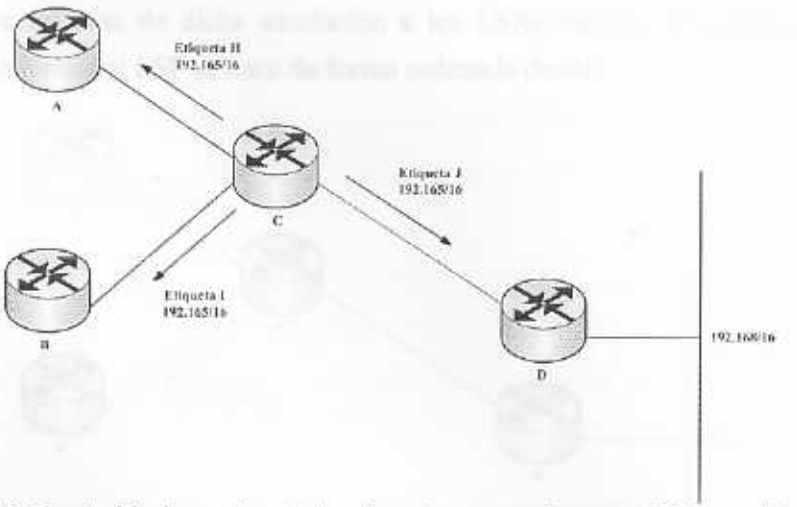

Figura 19. Control independiente de etiquetas, segunda parte (Elaboración propia).

7.5.2.- Control ordenado de etiquetas:

La asignación de etiquetas ocurre de forma ordenada desde un extremo del LSP hacia el otro. El establecimiento del LSP puede iniciarse por el LSR de entrada o por el LSR de salida del LSP. Por ejemplo, supongamos que el establecimiento del LSP lo inicia el LSR de salida. En la Figura 20 el nodo D se da cuenta que es el LSR de salida para el prefijo de dirección 192.168/16. Dicho nodo asigna una etiqueta a esta FEC e informa de dicha asociación a su vecino.

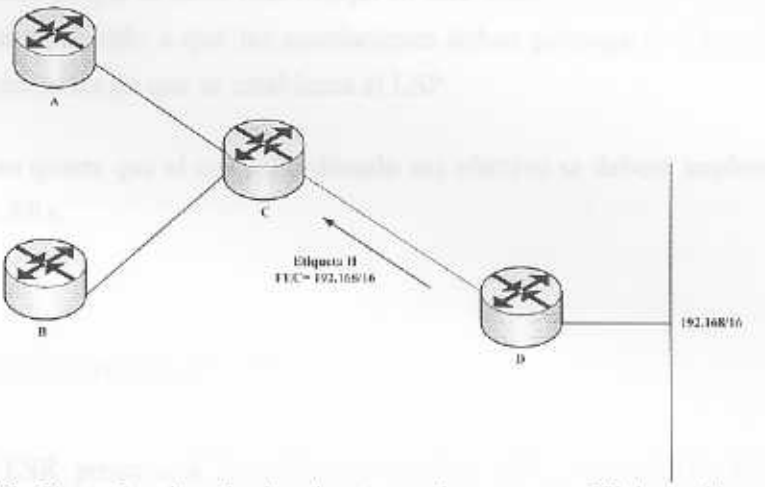

Figura 20. Control ordenado de etiquetas, primera parte (Elaboración propia).

En la Figura 21, cuando el LSR C recibe dicha información, asigna una etiqueta e informa de dicha asociación a los LSRs vecinos. De esta forma el establecimiento del LSP se hace de forma ordenada desde

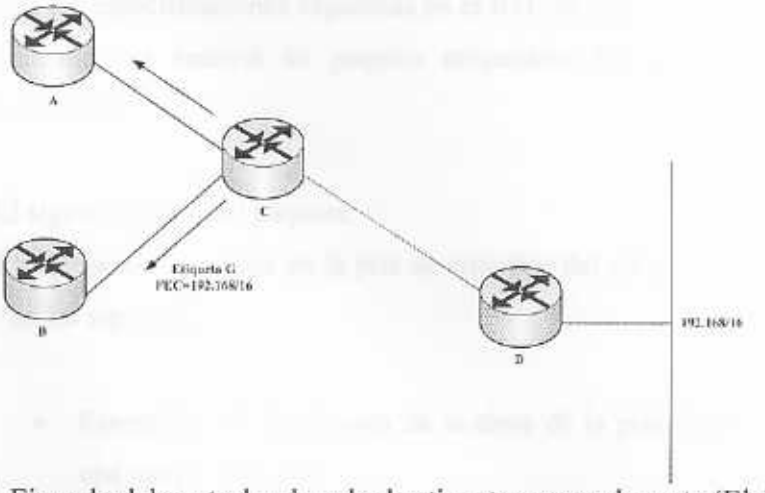

Figura 21. Ejemplo del control ordenado de etiquetas, segunda parte (Elaboración propia).

Si se pretende garantizar que el tráfico de una FEC particular sigue un camino que tiene una serie de propiedades, se debe usar el control ordenado.

MPLS permite tanto el control independiente como el control ordenado. Un LSR sólo necesita implementar uno u otro.

El control ordenado facilita la prevención de bucles y también permite a los administradores de la red controlar cómo se establecen los LSPs. Un inconveniente es que se tarda más tiempo en establecer un LSP que con el control independiente, debido a que las asociaciones deben propagarse a través de una región entera antes de que se establezca el LSP.

Si se quiere que el control ordenado sea efectivo se deberá implementar en todos los LSRs.

7.6.- Funcionamiento del LSR:

El LSR posee una base de información del reenvío (FIB: *Forwarding Information Base*), la cual esta compuesta por:

1. Entrada para el reenvío con la etiqueta del siguiente salto (NHLFE: *Next Hop Label Forwarding Entry*):
   Según las especificaciones expuestas en el RFC3031 (2001), la NHLFE se usa cuando se reenvía un paquete etiquetado. Contiene la siguiente información:

   a. El siguiente salto del paquete.
   b. La operación a realizar en la pila de etiquetas del paquete, que será una de las siguientes:

      - Reemplazo de la etiqueta de la cima de la pila de etiquetas con una nueva etiqueta.
      - Extracción de la pila de etiquetas.

- Reemplazo de la etiqueta de la cima de la pila de etiquetas con una nueva etiqueta y posterior apilamiento de una o más nuevas etiquetas en la pila de etiquetas.

También puede tener:

- Encapsulación del nivel de enlace a usar cuando se transmita un paquete.
- Forma de codificar la pila de etiquetas cuando se transmita el paquete.

2. Correlación de la etiqueta entrante (ILM, "*Incoming Label Map*"):
   El ILM correlaciona cada etiqueta entrante con un conjunto de NHLFEs. Se utiliza cuando se reenvían paquetes que llegan etiquetados. Si la ILM correlaciona una etiqueta particular con un conjunto de NHLFEs que contienen más de un elemento, se deberá elegir exactamente un elemento del conjunto antes de reenviar el paquete. La etiqueta de la cima de la pila se utiliza como índice de la ILM. Puede ser útil hacer que la ILM correlacione una etiqueta con un conjunto que contenga más de una NHLFE porque podríamos hacer un balance de la carga de tráfico a través de múltiples enlaces.

3. Correlación de la FEC con la NHLFE (FTN, "FEC-to-NHLFE"):
   La FTN correlaciona cada FEC con un conjunto de NHLFEs. Se utiliza cuando se quieren reenviar paquetes que no llegan etiquetados, pero que se quieren reenviar etiquetados. Si la FTN correlaciona una etiqueta particular con un conjunto de NHLFEs que contienen más de un elemento, se deberá elegir exactamente un elemento del conjunto antes de reenviar el paquete.

7.6.1.- Módulo de determinación de la ruta:

También conocido como componente de control; este módulo construye las entradas de la FIB en una operación MPLS. Con la información de los protocolos de encaminamiento se determinan las FECs para las que se quieren crear NHLFEs. Igualmente obtiene la información del siguiente salto que se necesita para crear la NHLFE.

Dado que en MPLS sólo se permite la asociación de etiquetas río abajo, una NHLFE no tendrá información de la etiqueta de salida hasta que el LSR del mismo nivel que está río abajo le asigne una etiqueta.

Un LSR puede construir las NHLFEs de dos formas, tal como especifica el RFC3031 (2001):

1. Asignando una o más etiquetas para usar como la etiqueta de entrada, creando ILMs para cada una, asociando cada ILM a un conjunto de NHLFEs y distribuyendo las etiquetas asociadas a los LSRs que están río arriba.

2. Creando FTNs para las FECs asociadas con entradas específicas de encaminamiento y asociando cada una a un conjunto de NHLFEs con la información del siguiente salto.

La función de determinación de la ruta se usa también para borrar o actualizar las entradas de la FIB cuando las rutas asociadas a una FEC son removidas o cuando cambia la información del siguiente salto.

7.6.2.- Módulo de reenvío:

La función de reenvío se basa en la comparación exacta entre una etiqueta y la ILM, que a su vez se correlaciona con una NHLFE. El LSR seguirá las instrucciones de manipulación de la etiqueta que indique la NHLFE y enviará el paquete por la interfaz especificada en la información del siguiente salto. Puede ser que el LSR necesite usar la información de encapsulación de nivel de enlace

especificada en la NHLFE para encapsular el paquete antes de enviarlo al siguiente salto. Ejemplo de esto último sería un enlace *Ethernet*, en donde la dirección MAC (*Media Access Control*: Control de Acceso al Medio) de la interfaz de salida y el siguiente salto se tienen que incluir en la encapsulación *Ethernet* para poder entregar el paquete al siguiente salto.

El RFC3031 (2001) considera los siguientes casos para el intercambio de etiquetas:

1. Si se quiere reenviar un paquete etiquetado, el LSR examinará la etiqueta de la cima de la pila de etiquetas. Con dicha etiqueta escogerá una ILM y obtendrá la NHLFE. Con la NHLFE obtendrá la nueva etiqueta y la información necesaria para reenviar el paquete.

2. Si se quiere reenviar un paquete no etiquetado, el LSR examinará la dirección del nivel de red para determinar la FEC a la que pertenece el paquete. Una vez hecho esto usará la FTN para correlacionar la FEC con una NHLFE. Con la NHLFE sabrá dónde reenviar el paquete y realizará una operación en la pila de etiquetas del paquete.

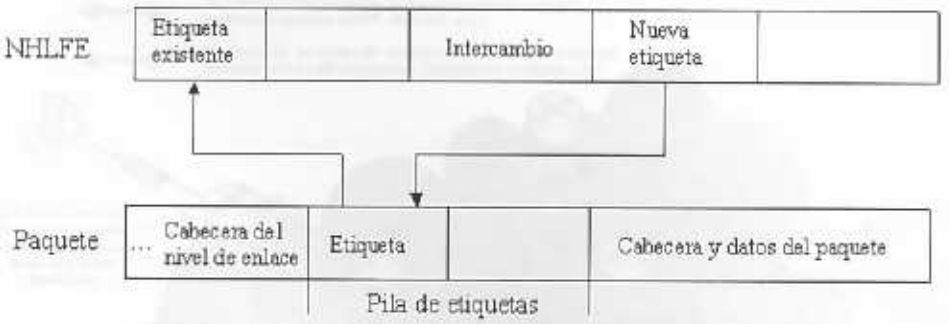

Figura 22. Modulo de reenvío en el LSR

En ambos casos el LSR codificará la nueva pila de etiquetas en el paquete y lo reenviará.

7.7.- Funcionamiento del MPLS:

Una red MPLS esta conformada por un conjunto de Enrutadores de Conmutación de Etiquetas (LSR), estos enrutadores tienen la capacidad de

conmutar y encaminar paquetes en base a la etiqueta que se ha añadido a los mismos. Cada etiqueta define un flujo de paquetes entre dos puntos finales. A este flujo se le conoce como: Clase de Equivalencia de envío (FEC). Cada FEC, además de la ruta de los paquetes, contiene una serie de caracteres que define los requerimientos de QoS del flujo. Los *routers* de la red MPLS no necesitan examinar ni procesar el encabezado IP, sólo es necesario reenviar cada paquete dependiendo el valor de su etiqueta. Esta es una de las ventajas que tienen los *routers* MPLS sobre los *routers* IP, en donde el proceso de reenvío es más complejo.

En un *router* IP, cada vez que se recibe un paquete se analiza su encabezado IP para compararlo con la tabla de enrutamiento y ver cuál es el siguiente salto o *next hop*. El hecho de examinar estos paquetes en cada uno de los puntos de tránsito que deberán recorrer para llegar a su destino final significa un mayor tiempo de procesamiento en cada nodo, y por lo tanto una mayor duración en el recorrido.

Para conocer mejor el funcionamiento del protocolo MPLS se presenta a continuación un diagrama que explica la operación del mismo, dicho diagrama es expuesto por Canalis (2002).

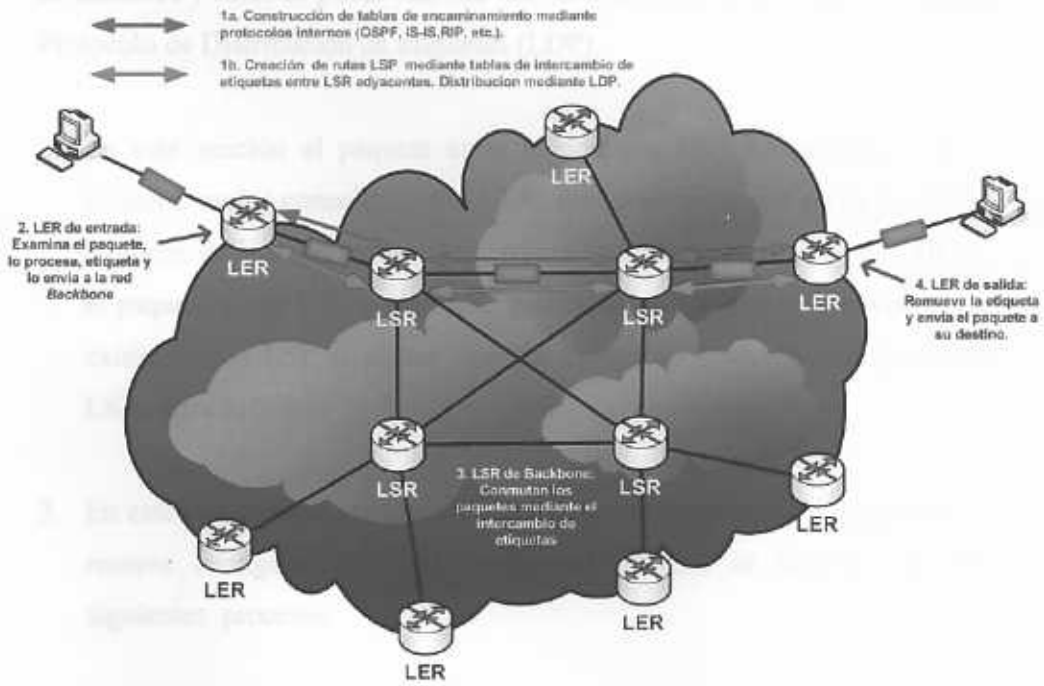

Figura 23. Funcionamiento de una red MPLS (Canalis, 2002).

Lo primero a tomar en cuenta es que las tablas de encaminamiento ya están construidas. Como ya se mencionó en apartados anteriores esta tabla se puede levantar utilizando protocolos de enrutamiento internos, tal como OSPF, IS-IS, RIP, etc.

En la figura 23 se exponen y explican los pasos que sigue el flujo de paquetes MPLS:

1. Antes de mandar la información por el flujo es necesario establecer un Camino de Conmutación de Etiquetas (LSP) entre los *routers* que van a transmitir la FEC. Dichos LSP sirven como túneles de transporte a lo largo de la red MPLS e incluyen los parámetros QoS específicos del flujo. Estos sirven para determinar dos cosas:
   a. La cantidad de recursos a reservar al LSP.
   b. Las políticas de desechado y la cola de procesos en cada LSR.

   Se le asignan etiquetas a cada flujo FEC particular para evitar el uso de etiquetas globales que dificultan el manejo y la cantidad de las mismas. Por esta razón las etiquetas solo hacen referencia al flujo específico. La asignación de nombres y rutas se puede realizar manualmente o bien se puede utilizar el Protocolo de Distribución de Etiquetas (LDP).

2. En esta sección el paquete entra al dominio MPLS mediante un LSR frontera, mejor conocido como LER, que determina qué servicios de red requiere, definiendo así su QoS. Al concluir dicha asignación el LSR fija el paquete a una FEC y a un LSP particular, lo etiqueta y lo envía. Si no existe ningún LSP el *router* frontera trabaja en conjunto con los demás LSRs para definirlo.

3. En este momento el paquete ya está dentro del dominio MPLS, cuando los *routers* contiguos del LSR reciben el paquete se llevan acabo los siguientes procesos:

a. Se desecha la etiqueta de entrada y se le añade la nueva etiqueta de salida al paquete.

b. Se envía el paquete al siguiente LSR dentro del LSP.

4. El LSR de salida "abre" la etiqueta y lee el encabezado IP para enviarlo al destino final.

7.8.- Ventajas específicas de MPLS:

En este momento ya es posible identificar algunas de las ventajas internas más importantes que MPLS presenta:

1. Un dominio MPLS consiste de una serie de *routers* continuos y contiguos habilitados con MPLS. El tráfico puede entrar por un punto final físicamente conectado a la red o por otro *router* que no sea MPLS y que esté conectado a una red de computadoras sin conexión directa a la nube MPLS.

2. Se puede definir un Comportamiento por Salto (PHB) diferente en cada *router* de la FEC. El PHB define la prioridad en la cola y las políticas de desechado de los paquetes.

3. Para determinar el FEC se pueden utilizar varios parámetros que define el administrador de la red.

a. Dirección IP fuente o destino y/o las direcciones IP de la red

b. Utilizar el ID del protocolo IP.

c. Etiqueta de flujo IPv6.

d. Numero de puerto de la fuente o del destino.

e. El punto de código (codepoint) de los servicios diferenciados (DSCP).

4. El reenvío de la información se lleva acabo mediante una búsqueda simple o "*lookup*" en una tabla predefinida que enlaza los valores de las etiquetas con las direcciones del siguiente salto (*next hop*).

5. Los paquetes enviados desde los mismos *endpoints* pueden tener diferente FEC, por lo que las etiquetas serán diferentes y tendrán un PHB distinto en cada LSR. Esto puede genera diferentes flujos en la misma red.

8.- MPLS y DSCP (Servicios diferenciados):

El requerimiento para que una red MPLS funcione como una red de servicios diferenciados es el siguiente: los nodos de ingreso MPLS deben realizar un mapeo entre el campo DSCP de los paquetes IP y el campo EXP en los paquetes MPLS.

Realizar un mapeo de tráfico de servicios diferenciados a MPLS, puede ser hecho de dos maneras:

- Usar los bits de EXP en la cabecera MPLS.
- Usar la etiqueta por si sola.

Si el proveedor quiere marcar los paquetes usando el campo EXP, entonces un LSP puede sustentar un máximo de 8 servicios por clase (debido a que el campo EXP posee una longitud de 3 bits). El DSCP soporta 64 clases de servicio, así que debe haber un mapeo entre estas dos. IETF aun no ha propuesto un estándar de mapeo, aunque Cisco y otros productores de equipos copian los 3 bits más importantes del campo DSCP para el campo EXP como estándar. De tal modo, podemos construir la siguiente tabla:

DSCP	EXP	DSCP	EXP
Mejor esfuerzo - 000000	0	AF32 – 011100	3
AF11 – 001010	1	AF33 – 011110	3
AF12 – 001100	1	AF41 – 100010	4
AF13 – 001110	1	AF42 – 100100	4
AF21 – 010010	2	AF43 – 100110	4
AF22 – 010100	2	EF – 101110	5
AF23 – 010110	2	Control de tráfico - 110000	6
AF31 – 011010	3	Control de tráfico - 111000	7

Tabla 1. Mapeo DSCP al campo EXP de la cabecera MPLS (Dumitrascu, 2006).

De acuerdo a esta tabla, se aprecia que por definición, los flujos marcados como servicios diferenciados se fusionan en un único flujo desde el punto de vista de las clases de servicio. También, los valores EXP=6,7 se usan solamente para control de tráfico.

La tabla previa, es tan solo un ejemplo. Si el proveedor no posee todas las clases de servicio, existe la factibilidad de hacer otro mapeo dentro del dominio (en este caso, el proveedor solo utiliza AF11-AF23 y EF).

DSCP	EXP
Mejor esfuerzo – 000000	0
AF11 – 001010	1
AF12 – 001100	2
AF13 – 001110	3
AF21 – 010010	4
AF22 – 010100	5
AF23 – 010110	6
EF – 101110	7

Tabla 2. Mapeo E-LSP (Dumitrascu, 2006).

Esta forma de mapeo DSCP es llamada E-LSP (*EXP Infered Per-hop-behaviour SCheduling LSP*). En redes MPLS reales, este mapeo es negociado a través de protocolos dinámicos de distribución de etiquetas.

La segunda forma en la que un proveedor puede ofrecer servicios diferenciados sobre MPLS es utilizando la etiqueta para distinguir entre las clases de servicio existentes. De esta manera un LSP con cierta prioridad será creado para un mismo destino; El LSP mostrará el camino hasta el destino, así como también el trato que deben recibir los paquetes que viajan por el camino. En este caso el campo EXP de la cabecera MPLS contendrá un valor que nos indica la prioridad del flujo en cuanto a la pérdida de paquetes (drop precedence).

Con este método no nos limitamos a sólo 8 clases de servicio, como sucedía en el caso anterior. Sin embargo, el procesamiento en los nodos es mayor, teniendo que administrar mas etiquetas a un mismo destino. Esta alternativa para realizar el mapeo DSCP se denomina L-LSP (*Label Infered Per-hop-behaviour Scheduling LSP*).

El RFC2597 (1999) describe a los *Assured Forwarding* (AFx) como un medio para que el proveedor del dominio de *Diffserv* sea capaz de ofrecer niveles de garantías para la transmisión de paquetes IP.

La clase *Expedited Forwarding* (EF) tiene la característica de baja demora, baja pérdida y de bajo ruido. Son adecuadas para la transmisión de voz, vídeo y otros servicios en tiempo real. Al tráfico EF se suele dar prioridad estricta de gestión de colas por encima de todas las demás clases de tráfico AF.

9.- Protocolo de distribución de etiquetas LDP:

El protocolo de distribución de etiquetas LDP (*Label Distribution Protocol*) se ejecuta sobre TCP y, por tanto, es un protocolo de estado duro. Dado que se ejecuta sobre TCP, éste le proveerá de fiabilidad en el envío de mensajes.

El uso más sencillo de LDP consiste en establecer enlaces unitarios de LSPs. Para hacer esto se puede usar la distribución de etiquetas río abajo no solicitado o río abajo por demanda y es compatible con el control ordenado y con el control independiente. Se podrá usar el modo de retención de etiquetas

conservador o el liberal. Pero habrá combinaciones no factibles. Veámoslo con un par de ejemplos:

- Si los LSRs vecinos utilizan la distribución de etiquetas río abajo no solicitado y el LSR local utiliza el modo conservador de retención de etiquetas, habrá mucho tráfico de liberación de etiquetas.
- Si los LSRs vecinos utilizan la distribución de etiquetas río abajo por demanda y el LSR local utiliza el modo liberal de retención de etiquetas habrá mucho tráfico de petición de etiquetas.

LDP es un protocolo muy útil para los casos en los que se desea establecer un LSP a través de LSRs que no soporten *piggybacking* (básicamente esta es la única ventaja de LDP). LDP es bidireccional y podrá operar entre LSRs adyacentes o no adyacentes.

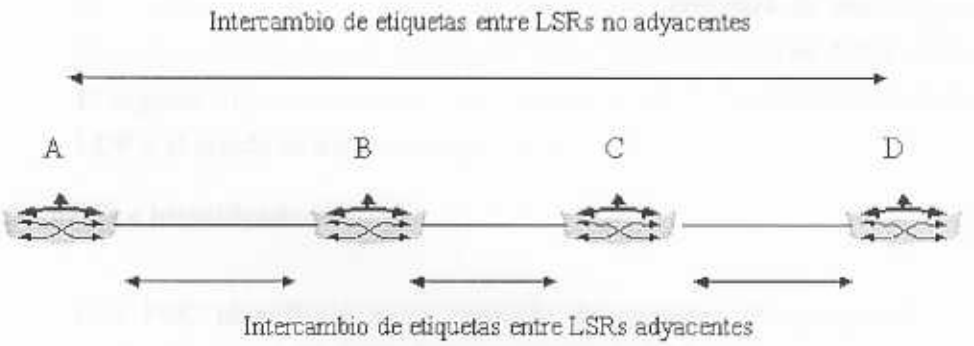

Figura 24. Intercambio de etiquetas mediante LDP (Davie, 2000)

El protocolo de distribución de etiquetas asocia una FEC con cada LSP que crea. Dos LDPs serán pares LDP (LDP peers) cuando ambos LSRs intercambien información de asociaciones de etiquetas y FECs. Para intercambiar dicha información establecerán una sesión LDP.

9.1.- Mensajes LDP:

Tal como indica el RFC3036 (2001), los pares LDP se podrán intercambiar cuatro clases de mensajes:

1. Mensajes de descubrimiento (*discovery messages*): se usan para anunciar y mantener la presencia de un LSR en la red. Un LSR mandará periódicamente por la red mensajes HELLO a través de un puerto UDP con la dirección *multicast* "todos los encaminadores de esta subred".
2. Mensajes de sesión: se utilizan para establecer, mantener y terminar sesiones entre pares LDP. Cuando un LSR descubre a otro por medio de mensajes HELLO utilizará un procedimiento de iniciación LDP por medio de TCP.
3. Mensajes de anuncio (*advertisement messages*): se usan para crear, modificar y eliminar asociaciones de etiquetas a FECs. Se transportan vía TCP. Cuando se haya establecido la asociación los pares LDP podrán intercambiarse este tipo de mensajes.
4. Mensajes de notificación: Los mensajes de notificación también se transportan vía TCP. Hay dos tipos de mensajes de notificación: notificaciones de error y notificaciones de aviso. El primer tipo se utiliza para notificar errores fatales, en cuyo caso terminará la sesión y se descartarán todas las asociaciones de etiquetas aprendidas en dicha sesión. El segundo tipo se utiliza para pasarle a un LSR información de la sesión LDP o el estado de algún mensaje anterior.

9.2.- FECs e identificadores:

Una FEC identificará a un conjunto de paquetes IP que podrán ser enviados a través de un LSP. LDP define dos tipos de FECs:

- Prefijo de dirección
- Dirección de host

Habrá una correspondencia entre una dirección particular y un prefijo de dirección si la dirección comienza con el prefijo. Habrá una correspondencia entre un paquete y un LSP si existe una correspondencia entre el prefijo de dirección del LSP y la dirección de destino del paquete.

El procedimiento para correlacionar un paquete a un LSP está formado por una serie de reglas. Estas reglas se aplicarán hasta que el paquete pueda ser correlacionado a un LSP. Las reglas son:

- Si hay exactamente un LSP con un elemento FEC de dirección host con la misma dirección destino que el paquete, entonces el paquete se correlacionará con ese LSP.
- Si hay varios LSPs, cada uno con un elemento FEC de dirección host idéntica a la dirección destino del paquete, entonces el paquete se correlacionará con uno de esos LSPs.
- Si hay una única equivalencia entre un paquete y un LSP, entonces el paquete se correlacionará con ese LSP.
- Si hay múltiples equivalencias entre un paquete y varios LSPs, entonces el paquete se correlacionará con el LSP que tenga mayor porcentaje de igualdad en el prefijo (es decir, el más largo).
- Si un paquete debe atravesar un encaminador frontera, y existe un LSP con un elemento FEC de prefijo de dirección que es una dirección de ese encaminador, entonces el paquete se correlacionará con ese LSP.

9.3.- Identificadores LDP:

Un identificador LDP se utiliza para identificar el espacio de etiquetas de un LSR. Se compone de seis octetos, de los cuales los cuatro primeros identifican al LSR y los dos últimos identifican el espacio de etiquetas de dicho LSR. Como se comentó en capítulos anteriores, el espacio de etiquetas puede ser por interfaz o por plataforma. Si los dos últimos octetos tienen un valor de cero el espacio de etiquetas será por plataforma.

La especificación de LDP en el RFC3036 utiliza la siguiente nomenclatura para representar un identificador LDP:

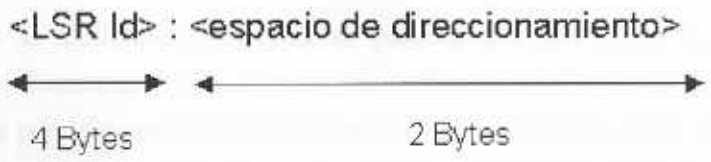

Figura 25. Identificador LDP (Elaboración propia).

9.4.- Sesión LDP:

Cuando un LSR utiliza LDP para anunciar más de un espacio de etiquetas a otro LSR, utilizará diferentes sesiones LDP para cada espacio de etiquetas. Como se comentó anteriormente, LDP utiliza TCP. Cuando dos LSRs requieren múltiples sesiones LDP, se establecerán sesiones TCP distintas para cada sesión LDP. En la especificación del protocolo se definen dos fases para el establecimiento de la sesión LDP:

- Descubrimiento.
- Establecimiento y mantenimiento de sesiones LDP.

9.5.- Descubrimiento:

El protocolo de descubrimiento de LDP utiliza UDP como protocolo de transporte. Existen dos modalidades de descubrimiento: básica y extendida.

En la modalidad básica el LSR envía periódicamente mensajes HELLO a un puerto bien conocido con la dirección multicast "todos los encaminadores de esta red". Los encaminadores están escuchando continuamente en este puerto a la espera de recibir mensajes HELLO. Por tanto, llegará un momento en el que el LSR conocerá todos los LSRs con los que tiene una conexión directa. Por tanto este mecanismo se utiliza si los LSRs están conectados directamente por medio de un enlace.

Los mensajes HELLO transportarán el identificador LDP con el espacio de etiquetas que LSR pretende usar en esa interfaz, además de otro tipo de información.

Con la modalidad extendida se permite que dos LSRs que no están conectados directamente establezcan una sesión LDP. Con esta modalidad, un LSR emite periódicamente mensajes HELLO a un puerto (UDP) bien conocido y con una dirección específica, que habrá aprendido de algún modo (por ejemplo, por configuración). Los mensajes HELLO transportarán el identificador LDP con el espacio de etiquetas que LSR pretende usar, además

de otro tipo de información. El LSR al que se le están enviando los mensajes HELLO podrá responder o ignorar dicho mensaje. Si decide responder a dicho mensaje deberá mandar periódicamente mensajes HELLO al LSR que inició el proceso.

La modalidad extendida es útil cuando se ha configurado un LSP entre dos LSRs por ingeniería de tráfico, deseando mandar paquetes ya etiquetados a través de ese LSP. El LSR situado al principio del LSP necesitará saber como etiquetar los paquetes que le enviará la LSR situado al final del LSP.

9.6.- Establecimiento y mantenimiento de sesiones LDP:

Una vez conocidos los vecinos se podrá establecer la sesión. Cada uno de los LSRs implicados puede jugar un papel activo o pasivo. El establecimiento de una sesión consta de dos fases:

1. Establecimiento de la conexión de transporte

   Esta fase consiste en el establecimiento de una conexión TCP entre los LSRs implicados, para una nueva sesión LDP.

2. Inicio de la sesión

   Una vez establecida la conexión TCP los LSRs deben negociar los parámetros de la sesión. Esto se hace intercambiando mensajes de iniciación. Estos parámetros incluyen la versión del protocolo LDP, el método de distribución de etiquetas, valor de los temporizadores, etc.

   Si el LSRa juega el papel activo, éste iniciará la negociación de los parámetros de la sesión enviando un mensaje de iniciación al LSRb. Este mensaje contendrá tanto el identificador LDP del LSRa como el identificador del LSRb.

   Cuando un LSR recibe un mensaje de iniciación, mirará dicho mensaje para determinar si los parámetros son aceptables. Si lo son, responderá con su propio mensaje de iniciación proponiendo los parámetros que desea usar y un mensaje de mantenimiento (KeepAlive) para notificar al otro LSR que acepta los parámetros. Si los parámetros no son aceptables,

responderá con un mensaje de notificación de error de parámetros rechazados.

3. Máquina de estado de la negociación de la sesión LDP

A continuación se muestra la Tabla de transición, y el diagrama de transición de estado de inicialización de la sesión.

Estado	Evento	Nuevo estado
NO EXISTENTE	Conexión TCP de sesión establecida	INICIALIZADO
INICIALIZADO	Transmitir mensaje de inicio(papel activo)	OPENSENT
INICIALIZADO	Recibir mensaje de inicio aceptable (papel pasivo). Acción: transmitir mensajes de inicio y mantenimiento.	OPENREC
INICIALIZADO	Recibir cualquier otro mensaje LDP. Acción: transmitir mensaje de notificación de error (NAK) y cerrar la conexión de transporte	NO EXISTENTE
OPENREC	Recibir mensaje de mantenimiento	OPERACIONAL
OPENREC	Recibir cualquier otro mensaje LDP. Acción: transmitir mensaje de notificación de error (NAK) y cerrar la conexión de transporte.	NO EXISTENTE
OPENSENT	Recibir un mensaje de iniciación aceptable. Acción: transmitir un mensaje de mantenimiento.	OPENREC
OPENSENT	Recibir cualquier otro mensaje LDP. Acción: transmitir mensaje de notificación de error (NAK) y cerrar la conexión de transporte.	NO EXISTENTE
OPERACIONAL	Recibir mensaje de finalización. Acción: transmitir mensaje de finalización y cerrar la conexión de transporte.	NO EXISTENTE
OPERACIONAL	Recibir cualquier otro mensaje LDP	OPERACIONAL
OPERACIONAL	Intervalo de tiempo sobrepasado. Acción: transmitir mensaje de finalización y cerrar la conexión de transporte.	NO EXISTENTE

Tabla 3. Comandos LDP (Elaboración propia según el RFC 3036, 2001).

10.- VPN MPLS:

Según el *Whitepaper* presentado por Sonicwall Inc. (2005) se elaboró el siguiente resumen, en donde se explica de forma concreta los beneficios e inconvenientes de las VPNs MPLS.

10.1.- Tecnologías VPN:

El nuevo modelo de VPNs implementando MPLS, plantea un problema: la seguridad. Si bien es importante compartir los recursos de la red, esto ha de hacerse de forma segura. Una VPN constituye una agrupación lógica de usuarios dentro de una red de cierto tamaño. La VPN envía el tráfico mediante un túnel privado más seguro a través de una red pública compartida, que puede ser Internet o, en el caso de MPLS, la red de un proveedor de servicios.

Las redes ATM y Frame Relay han permitido aumentar la capacidad y la velocidad de las WAN, ofreciendo una mayor facilidad de gestión mediante el uso de circuitos virtuales Frame Relay o ATM. No obstante, estas dos tecnologías de conexión en red tienen una capacidad limitada para transportar diferentes tipos de tráfico y están dando paso a nuevas alternativas VPN basadas en Protocolo Internet (IP). Estas VPN permiten una mayor flexibilidad, pues soportan distintos tipos de información en la red. Las VPN basadas en IPSec y MPLS representan el siguiente nivel de la tecnología WAN, permitiendo la creación de redes multi-servicio capaces de transportar cualquier tipo de tráfico.

Aunque el concepto de VPN continúa siendo el mismo, las nuevas VPN ofrecen hoy una mayor funcionalidad y un nivel de seguridad superior, así como la capacidad para transmitir diferentes tipos de tráfico. Además, como ocurre con cualquier tecnologia, su coste se ha reducido, ya que las innovaciones más recientes ofrecen mejoras a un precio inferior.

La tecnología VPN IPSec permite establecer conexiones remotas seguras usando un estándar abierto y aprovechando una infraestructura compartida ya existente.

Es evidente que la transmisión de datos corporativos por Internet puede generar numerosos problemas de seguridad. No obstante, cualquier VPN con prestaciones sencillas ofrece un nivel básico de seguridad, ya que permite separar los flujos de datos y formar grupos lógicos de usuarios, restringiendo de esta forma el acceso de cada usuario al contenido de su propia VPN. Sin embargo, el aislamiento de los recursos de la red como la única medida de seguridad resulta restrictivo e inadecuado para la mayoría de las redes corporativas. Por ello, las VPN basadas en tecnología IPSec van más allá: utilizan tecnologías de cifrado y autenticación para crear un túnel privado seguro a través de una red IP que, de otra forma, no sería segura.

Según un estudio llevado a cabo por Infonetics Research, la constante evolución de las VPN y el deseo de adquirir la tecnología más reciente son las razones principales por las que las empresas sustituyen sus productos VPN. Según este estudio, la seguridad constituye una de las grandes preocupaciones para las empresas que utilizan tecnologías VPN: de hecho, más de un tercio de los encuestados afirmaron que la seguridad representa una barrera de cara a la implementación de una VPN.

10.2.- MPLS para VPNs:

A diferencia del estándar IPSec, MPLS utiliza la red propietaria del proveedor de servicios y se comercializa normalmente como servicio gestionado. Por ello, MPLS resulta mucho más rentable para los proveedores de servicios.

MPLS permite crear un enlace entre dos o más puntos terminales fijos dentro de la red del proveedor de servicios. Sin embargo, al contrario que en redes IPSec, al crear una red MPLS no se instalan equipos en las oficinas del cliente,

debiéndose encontrar todos los puntos terminales en el ámbito de la red del proveedor de servicios. Por ello, el uso de MPLS no es frecuente cuando se requiere una conectividad entre emplazamientos geográficamente distantes y tampoco resulta práctico para aplicaciones de conectividad remota y móvil. En cambio, sí suele utilizarse para conectar múltiples divisiones corporativas o sucursales grandes ubicadas en una región geográfica definida.

MPLS sólo funciona en la red del proveedor de servicios, no es posible conectar todas las ubicaciones a menos que la red del proveedor de servicios se extienda hasta dichos puntos. MPLS no fue diseñado para establecer conexiones entre trabajadores móviles remotos y las oficinas centrales. Para establecer este tipo de entorno de teletrabajo corporativo, es preferible una VPN IPSec o SSL. Además, mientras MPLS resulta útil para la creación de un canal multiprotocolo, por sí mismo, no ofrece muchas ventajas en cuanto a transporte seguro de los datos. Una VPN MPLS aísla el tráfico de la misma forma que ATM o Frame Relay, pero no incluye ninguna función para el cifrado, a nos ser que se utilice IPSec.

En realidad, MPLS no fue diseñado para ser un protocolo seguro. El objetivo de sus creadores era desarrollar una forma de etiquetar paquetes para conseguir una transferencia más eficaz (y esta función la cumple con creces). Sin embargo, MPLS no encripta dichos paquetes, por lo que pueden resultar muy vulnerables a la intrusión, las intervenciones y otros tipos de ataques nefastos. También es posible falsificar el espacio de dirección del cliente o la propia etiqueta MPLS. Esto no significa que MPLS sea una tecnología deficiente, sino simplemente que no se debe implementar MPLS sin una función de seguridad añadida, como puede ser un punto de seguridad en el emplazamiento del cliente (p.ej., cortafuegos con VPN IPSec).

10.3.- IPsec:

IPSec se concibió desde el principio como un estándar para autentificar y cifrar datos en una red IP, así como para proteger la privacidad de los recursos de la red.

Los costes de propiedad son un factor decisivo a la hora de implementar IPSec, ya que este protocolo permite adquirir una conectividad de Internet a bajo precio, sin perder por ello las ventajas de una red de malla extensa y segura basada en las tecnologías más recientes.

Mientras que las VPN basadas en IP (como p.ej., redes MPLS) presentan amplios agujeros seguridad, IPSec fue concebido como un protocolo seguro. La VPN IPSec ofrece confidencialidad e integridad de datos, siendo la mejor solución de acceso remoto para emplazamientos múltiples.

IPSec se ha convertido en un estándar para la implementación de redes VPN, no sólo por su nivel de seguridad inherente, sino también porque no requiere ningún cambio en las estaciones de trabajo clientes.

Por regla general, para operar una VPN IPSec han de instalarse equipos en el emplazamiento del cliente. No obstante, muchos proveedores ofrecen servicios VPN IPSec en los que ellos mismos se encargan de gestionar la pasarela IPSec. Normalmente, este tipo de servicio gestionado incluye cierta garantía de rendimiento.

10.4.- Una comparación MPLS e IPsec:

MPLS e IPSec no son, como algunos fabricantes afirman, tecnologías que compiten la una con la otra. Más bien, cada una resulta útil para funciones distintas. MPLS (VPN IP) permite crear enlaces fijos entre emplazamientos.

Además, la VPN IP no es tan robusta en términos de seguridad. IPSec, por el contrario, se concibió desde un principio para garantizar el máximo nivel de seguridad. Por ello, cuando se crea una VPN mediante el protocolo IPSec, los datos quedan protegidos tanto por cifrado como por autentificación.

Prestación	VPN MPLS (VPN IP)	VPN IPSec
Conectividad entre emplazamientos	Sí	Sí
Conectividad de malla completa	Sí	Sí
Gestión QoS y de ancho de banda	Sí (CoS)	Sí (QoS granular)
Conectividad entre cliente y emplazamiento	No	Sí
Cifrado AES de 256 bits	No	Sí
Autentificación basada en PKI	No	Sí
Disponible en todo el mundo	No	Sí
Interoperabilidad entre proveedores de servicios	No	Sí
Requisito de contrato con el proveedor de servicios	Sí	No
Costes de implementación	$$$$	$$
Coste mensual del ancho de	$$	$

Figura 26. VPN-MPLS vs. VPN IPsec

La tarea de identificar la mejor ruta, ya sea Internet o la red propietaria de un proveedor de servicios, constituye un problema esencial. En términos de re-enrutamiento, MPLS tiene la capacidad de almacenar una ruta secundaria: al fallar la ruta principal, simplemente reenruta el tráfico por la ruta secundaria. No obstante, si falla también la ruta secundaria, ha de crearse una nueva ruta de forma manual. IPSec, por el contrario, utiliza protocolos de enrutamiento dinámico, es decir, detecta los fallos de ruta e identifica rutas alternativas de forma automática.

Al ser un estándar ampliamente extendido y probado, IPSec ofrece funciones de autentificación y cifrado completas, proporcionando el máximo nivel de seguridad. MPLS es una tecnología más reciente que se utiliza sobre todo en sustitución de ATM o Frame Relay para la conexión de emplazamientos grandes. Y mientras que IPSec, como protocolo basado en Internet, se puede implementar

prácticamente en cualquier parte, MPLS está limitado al ámbito de conectividad de la red del proveedor de servicios.

No obstante, gracias a la tecnología de etiquetado de MPLS, el tráfico se mueve de forma eficaz a través de la red del proveedor de servicios. Así, MPLS ofrece un rendimiento sólido de ancho de banda y garantías de nivel de servicio. Sin embargo, las VPN IPSec también pueden incluir prestaciones de gestión de ancho de banda. Además con la rápida disminución del coste del ancho de banda, es fácil añadir el ancho de banda requerido en la IPSec VPN a un precio asequible. De esta forma, se consigue, como mínimo, el mismo rendimiento que en una VPN MPLS, pero a un coste inferior.

El inconveniente de MPLS es que carece del elevado nivel de cifrado y autentificación que ofrece IPSec. Esto no significa que una red MPLS no proporcione seguridad alguna. MPLS ofrece de forma inherente la separación de flujos VPN, garantizando cierto nivel de privacidad basado en una estrategia de aislamiento, que dificulta el acceso de los intrusos a través de una puerta VPN individual. No obstante, aunque las VPN MPLS clasifican el tráfico, no cifran los paquetes ni ofrecen autentificación. Por ello, para conseguir un nivel de seguridad alto, ha de cifrarse el tráfico mediante IPSec antes de encapsularlo en MPLS.

De acuerdo con estas consideraciones, queda patente que MPLS no sirve realmente para sustituir a una VPN IPSec. MPLS es un sustituto eficaz de ATM o Frame Relay cuando se trata de crear una conectividad entre emplazamientos grandes, pero debería utilizarse en combinación con las VPN IPSec para lograr el máximo nivel de seguridad y alcance.

10.5.- MPLS e IPsec, Tecnologías complementarias:

Aunque en el pasado ha habido cierta polémica acerca del protocolo que se convertirá de facto en el estándar para conexiones VPN, la discusión no se ha enfocado de manera adecuada. En realidad, las empresas y los proveedores de servicios pueden utilizar tanto MPLS como IPSec para crear un entorno óptimo y

seguro que aproveche los puntos fuertes de ambos. Para los proveedores de servicios, es posible crear una oferta más amplia y lograr así un hueco competitivo en el mercado; para las empresas, es posible un entorno IPSec/MPLS integrado, capaz de garantizar una intranet/extranet plenamente segura con funciones de VPN remota igualmente seguras.

El objetivo de una solución VPN es conseguir una conectividad de alta calidad entre la sede principal y los empleados, clientes, oficinas remotas y cualquier usuario fuera del cortafuegos que necesite tener acceso a los recursos internos. Mientras que la red MPLS ofrece algunas ventajas (especialmente en el caso de conexiones entre emplazamientos grandes), su alcance es limitado (por ser una tecnología orientada hacia la conexión) y las sucursales que se encuentran fuera de la nube MPLS no tienen acceso. IPSec, por otra parte, es una tecnología sin conexiones y puede extenderse hasta casi cualquier parte del mundo, ya que se basa en la Internet. Por otro lado, las extranet corporativas son cada vez mayores, ya que cada vez son más lo partners que requieren acceso. Bajo estas circunstancias, MPLS podría no resultar práctico para una extranet muy extensa. Con IPSec, sin embargo, resulta fácil ampliar el alcance de las redes MPLS y ofrecer la conectividad que una empresa moderna requiere.

10.6.- Integración de IPSec en una VPN MPLS:

IPSec puede contribuir considerablemente a reducir los gastos, independientemente de si se utiliza sobre Internet, la red interna IP privada o una red MPLS. Un proveedor de servicios con una red MPLS, puede agregar las VPN IPSec y vincularlas al núcleo de la red VPN para obtener una mejor escalabilidad, seguridad, funcionalidad QoS y otras ventajas.

Normalmente, MPLS se ofrece como servicio gestionado y la VPN comienza y termina en la misma red IP del proveedor de servicios. Por el contrario, una VPN IPSec se origina y termina en el CPE y, con frecuencia, es la empresa cliente quién se encarga de su gestión (aunque algunos proveedores

también ofrecen VPN basadas en IPSec como servicio gestionado), siendo esta oferta combinada la elección más natural y lógica.

Además de ampliar el alcance de la red MPLS, se pueden integrar ambas tecnologías para lograr una infraestructura redundante. Para ello se utiliza la VPN IPSec como enlace redundante, en lugar de una red MPLS redundante, que resultaría mucho más cara. La integración de las dos tecnologías es una opción ideal y el Grupo de trabajo MPLS dispone ya de una especificación para el encapsulado *MPLS-in-IP*. La ventaja es evidente: al tiempo que se utiliza IPSec para enviar paquetes MPLS de forma segura por una red no MPLS, pueden añadirse funciones de autentificación y cifrado para proteger los datos.

Algunos proveedores de servicios ya ofrecen un enfoque combinado, lo que proporciona a los clientes un entorno VPN flexible apto para cualquier necesidad.

Apéndice C

Lista de acrónimos

Acrónimos	Inglés	Español
ARP	Address Resolution Protocol.	Protocolo de resolución de direcciones.
ATM	Asynchronous Transfer Mode.	Modo de transferencia asíncrono.
BGP	Border Gateway Protocol.	Protocolo de pasarela externa.
DiffServ	Differentiated Services.	Servicios diferenciados.
FEC	Functional Equivalence Class.	Clase de equivalencia funcional.
FIB	Forwarding Information Base.	Base de información del reenvío.
FIFO	First In First Out.	Cola
FTN	FEC-to-NHLFE.	Correlación de la FEC con la NHLFE.
IETF	Internet Engineering Task Force.	Grupo de trabajo de ingenieros de Internet.
IGP	Interior Gateway Protocol.	Protocolo interior de pasarela.
ILM	Incoming Label Map.	Correlación de la etiqueta entrante.
IP	Internet Protocol.	Protocolo de Internet.
IPv4	IP version 4.	IP versión 4.
IPv6	IP version 6.	IP versión 6.
LDP	Label Distribution Protocol.	Protocolo de distribución de etiquetas.
LER	Label Edge Router.	Encaminador de etiquetas frontera.

LIS	Logical IP Subnet.	Subred lógica IP.
LSP	Label Switched Path.	Camino de conmutación de etiquetas.
LSR	Label Switching Router.	Encaminador de conmutación de etiquetas.
MAC	Media Access Control.	Control de acceso al medio.
MPLS	Multiprotocol Label Switching.	Conmutación de etiquetas multiprotocolo.
NHLFE	Next Hop Label Forwarding Entry.	Protocolo de resolución del siguiente salto.
OSPF	Open Shortest Path First.	Protocolo abierto del primer camino más corto.
PHB	Per-Hop-Behaviour.	Comportamiento por salto.
QOS	Quality Of Service.	Calidad de servicio
RFC	Request For Comments	Documento de especificaciones del IETF.
RIP	Routing Information Protocol.	Protocolo de información de encaminamiento.
RSVP	Resource reSerVation Protocol.	Protocolo de reserva de recursos.
RSVP-TE	RSVP tunnel Extensions	Extensiones de RSVP para túneles LSP.
TCP	Transmisión Control Protocol.	Protocolo de control de la transmisión.
ToS	Type Of Service.	Tipo de servicio.
TTL	Time To Live.	Tiempo de vida.
UDP	User Datagram Protocol.	Protocolo de datagramas de usuario.
VPN	Virtual Private Network.	Red privada virtual.

Apéndice D

Glosario de términos

Algoritmo de enrutamiento o *Forwarding*: Parte del *Software* de un *router* responsable de tomar las decisiones relacionadas con el enrutamineto de los paquetes. Cuando se reciba un paquete deberá decidir por qué línea de salida deberá transmitir el paquete.

Ancho de banda: Capacidad de transmisión medida en bits por segundo. Indica la máxima capacidad teórica de conexión, aunque puede verse deteriorada por factores negativos como el retardo de transmisión.

ARP: *Address Resolution Protocol*. Ver protocolo de resolución de direcciones.

ATM: Tecnología utilizada tanto para redes locales como redes de área amplia. Utiliza conmutadores que establecen circuitos lógicos entre sistemas finales por lo que hay una garantía de QoS. Esta tecnología se utiliza como espina dorsal en redes de proveedores y en grandes compañías. Tiene una alta escalabilidad.

***Backbone*:** Nivel más alto de una red jerárquica. Se garantiza que las redes aisladas y de tránsito conectadas al mismo eje troncal están interconectadas.

Base de información del reenvío: Tabla que forma parte de un LSR y que contiene la NHLFE, la ILM y la FTN. Se utiliza para reenviar paquetes.

***Best-effort*:** "mejor esfuerzo". Los paquetes se entregan de la mejor forma posible.

Cabecera de un paquete: Información de control de un sistema definido que precede a los datos del usuario.

Calidad de servicio (QoS): Nivel de prestaciones de una red, basada en parámetros tales como la velocidad de transmisión, la variación del retardo, el rendimiento y la pérdida de paquetes.

Camino de conmutación de etiquetas: Camino a través de uno o más LSRs en un nivel de la jerarquía que siguen los paquetes de una FEC particular.

Clase de equivalencia de envío: Grupo de paquetes IP que se reenvían de la misma forma. La FEC permite agrupar paquetes en clases.

Clase de servicio: Categoría basada en el tipo de usuario, aplicación o criterio que los sistemas de QoS usan para proporcionar diferentes servicios.

Circuito virtual: Conexión establecida entre dos estaciones al comienzo de la transmisión. La ruta se establecerá antes de la transferencia de los datos. Todos los paquetes seguirán el mismo camino.

Cola o pila: Conjunto de paquetes en espera de ser procesados.

Condiciones de carrera: Condición que se da cuando se tiene la asociación de la etiqueta y no se tiene la información de encaminamiento asociada (asociación entre FECs y siguientes saltos).

Congestión: Circunstancia producida cuando el tráfico existente sobrepasa la capacidad de una ruta de comunicación de datos.

Conmutación de etiquetas: Término genérico usado para referirse al reenvío de paquetes IP usando el algoritmo de intercambio de etiquetas.

Conmutar: Operación que realizan *routers* y conmutadores. Éstos reciben un paquete por la línea de entrada y redirigen el paquete a la línea de salida adecuada en base a la información en la cabecera del paquete.

Correlación de la etiqueta entrante: Entrada de la FIB que sirve para correlacionar cada etiqueta entrante con un conjunto de NHLFEs. Se utiliza cuando se reenvían paquetes que llegan como paquetes etiquetados.

Correlación de la FEC con la NHLFE: Esta entrada de la FIB se utiliza cuando se quieren reeviar paquetes que no llegan etiquetados, pero que se quieren reenviar etiquetados.

Dominio de conmutación de etiquetas: Conjunto contiguo de nodos que operan con conmutación de etiquetas y que pertenecen a un mismo dominio de encaminamiento IP (o dominio administrativo).

***Router* conmutador de etiquetas (LSR):** Dispositivo que implementa la conmutación de etiquetas.

***Router* frontera conmutador de etiquetas (LER):** Nodo que conecta un dominio de conmutación de etiquetas con un nodo externo al dominio, bien porque no soporta la conmutación de etiquetas o porque pertenece a otro dominio de conmutación de etiquetas.

Enrutamiento salto a salto: Encaminamiento usual en redes IP. Cada LSR elegirá el siguiente salto hacia donde reenviar los paquetes de una FEC de forma independiente.

Encapsular: Información de control que le añade una entidad del protocolo a los datos obtenidos de un usuario de protocolo.

Entrada para el reenvío con la etiqueta del siguiente salto: Entrada de la FIB utilizada para reenviar paquetes etiquetados.

Espacio de etiquetas *(Lablespace)*: Alcance de una etiqueta en un LSR y cómo este alcance se relaciona con su par adyacente. Se hablará de alcance por interfaz y alcance por plataforma.

Espacio de etiquetas por interfaz: Una etiqueta se podrá interpretar de distinta forma dependiendo de la interfaz de entrada de dicha etiqueta.

Espacio de etiquetas por plataforma: Una etiqueta se interpretará de la misma forma independientemente de la interfaz de entrada de dicha etiqueta, siempre y cuando estas interfaces sean comunes con su par LSR.

***Ethernet*:** IEEE 802.3 (CSMA/CD). Red de difusión basada en bus con control descentralizado que opera a 10, 100, 1000 Mbps. En una red *ethernet*, los

computadores pueden transmitir cuando quieran. Si dos o más paquetes colisionan, los computadores esperarán un tiempo aleatorio y probarán a retransmitir más tarde.

Etiqueta: *Identificador de tamaño fijo que tiene significado local. Se usa para* reenviar paquetes. Un dispositivo de conmutación de etiquetas reemplazará la etiqueta de un paquete antes de reenviarlo.

Extracción en el penúltimo salto (PHP): Extracción de la etiqueta en el penúltimo LSR del LSP.

FEC: *Functional Equivalence Class*. Ver clase de equivalencia funcional.

Fiabilidad: Tasa media de error en la red.

FIFO: First In First Out. Término que se utiliza para referirse a una pila.

FTN: FEC-to-NHLFE. Ver correlación de la FEC con la NHLFE.

Fusión de etiquetas: Reemplazo de múltiples etiquetas de entrada para una FEC particular por una sola etiqueta de salida.

Identificador LDP: Usado para identificar el espacio de etiquetas de un LSR.

IETF: *Internet Engineering Task Force*. Grupo de ingenieros, que a través de su grupo de dirección (*Internet Engineering Steering Group*) se responsabiliza de toda la problemática técnica a corto plazo.

Ingeniería de Tráfico: Persigue adaptar flujos de tráfico a recursos físicos de la red, de tal forma que exista un equilibrio entre dichos recursos. De esta forma se conseguirá que no haya recursos excesivamente utilizados, con cuellos de botella, mientras existan recursos poco utilizados.

Intercambio de etiquetas: Algoritmo empleado por el componente de reenvío de un LSR. Cuando un LSR recibe un paquete extrae el valor de la etiqueta y accede con él a la tabla de encaminamiento. En dicha tabla de encaminamiento

encontrará el nuevo valor de la etiqueta que ha de ponerle al paquete antes de reenviarlo, así como la interfaz de salida por donde ha de mandarlo. También podrá encontrar información sobre si debe o no encolar el mensaje.

Interfaz: Zona de contacto o conexión entre dos aplicaciones o entre un usuario y una aplicación.

ILM: *Incoming Label Map*. Ver correlación de la etiqueta entrante.

***Label merging*:** Ver fusión de etiquetas.

***Label stack*:** Ver pila de etiquetas.

***Label swapping*:** Ver intercambio de etiquetas.

LDP: *Label Distribution Protocol*. Ver protocolo de distribución de etiquetas.

LSR de entrada: LSR que recibe tráfico de usuario (por ejemplo datagramas IP) y lo clasifica en su correspondiente FEC. Genera una cabecera MPLS asignándole una etiqueta y encapsula el paquete junto a la cabecera MPLS obteniendo una PDU MPLS.

LSR de salida: LSR que desencapsula un paquete removiendo la cabecera MPLS.

LSR frontera: LSR encargado de etiquetar los paquetes que entran en la red. Para poder realizar este trabajo, dicho LSR deberá implementar el componente de control y el componente de reenvío tanto del encaminamiento convencional como de la conmutación de etiquetas.

LSR interior: LSR que realiza el intercambio de etiquetas examinando exclusivamente la cabecera MPLS (obteniendo la etiqueta para poder realizar la búsqueda en la tabla de encaminamiento).

Mensaje de solicitud de reserva: Mensaje RSVP que establece la reserva desde el receptor al emisor creando en cada nodo RSVP el estado de la reserva.

***Merge point*:** Ver punto de fusión.

MPLS: *Multiprotocol Label Switching.* Ver Muliprotocolo conmutador de etiquetas.

Multidifusión *(Multicast):* Modo de difusión de información que permite que ésta pueda ser recibida por múltiples nodos de la red y por tanto, por múltiples usuarios.

Multiprotocolo conmutador de etiquetas: Estándar del IETF para la conmutación de etiquetas. Se basa en el uso de etiquetas las cuales identifican la ruta para encaminar los paquetes.

NHLFE: *Next Hop Label Forwarding Entry.* Ver entrada para el reenvío con la etiqueta del siguiente salto.

Nivel de enlace: Nivel 2 del modelo de referencia OSI. La tarea principal de este nivel es transformar unos recursos de transmisión y presentárselo al nivel de red como una línea libre de errores de transmisión sin detectar. Este nivel debe resolver los problemas causados por daño, pérdida y duplicado de tramas.

Nivel de red: Nivel 3 de la arquitectura OSI. Controla la operativa relacionada con la utilización de redes de comunicaciones. El aspecto clave está en la determinación de cómo encaminar los paquetes desde la fuente al destino.

Nivel de transporte: Nivel 4 de la arquitectura OSI. La función básica de este nivel es el de aceptar datos del nivel de sesión, descomponerlos en unidades más pequeñas, en caso de ser necesario, pasárselos al nivel de red y asegurarse de que llegan correctamente al otro extremo. En condiciones normales, el nivel de transporte crea conexiones de red distintas para cada conexión de transporte requerida por el nivel de sesión.

Nodo: Dispositivo direccionable conectado a una red de ordenadores.

Paquete: Unida de datos del protocolo de red. Un paquete incluirá datos y señales de control.

Paquete etiquetado: Paquete que tiene al menos una etiqueta en la pila de etiquetas.

Piggybacking: En MPLS, protocolos que incorporan la etiqueta encima de protocolos existentes de encaminamiento.

Pila de etiquetas: Conjunto ordenado de etiquetas.

Protocolo: Conjunto de reglas que gobiernan el formato y significado de las tramas, paquetes o mensajes que se intercambian entidades pares dentro de un nivel.

Protocolo de distribución de etiquetas (LDP): Conjunto de los procedimientos gracias a los cuales un LSR le informa a otro del significado de las etiquetas usadas para reenviar el tráfico a través de ellos.

Protocolo de encaminamiento: Los protocolos de encaminamiento sirven para determinas los caminos y mantener las tablas de encaminamiento.

Protocolo de Internet (IP): Protocolo orientado a conexión. Pertenece al nivel de red.

Protocolo de reserva de recursos (RSVP): Protocolo de estado blando utilizado para reservar recursos en una sesión en un entorno IP. Este protocolo permite la asignación de diferentes niveles de servicio a diferentes usuarios. Se utiliza para ofrecer discriminación de servicio a las aplicaciones sensibles al retardo mediante la asignación de recursos.

Protocolo de resolución de direcciones: Protocolo TCP/IP que convierte direcciones IP en direcciones físicas, como por ejemplo una dirección ethernet. Un host que desee obtener una dirección física enviará una petición ARP a la red.

Proveedor de servicios Internet: Organización que da acceso a Internet ofreciendo una serie de servicios.

Punto de fusión: Nodo en el que se realiza la fusión de etiquetas.

Red privada virtual (VPN): Red en la que la conectividad entre múltiples lugares se realiza a través de una infraestructura compartida con las mismas políticas de acceso y seguridad que en una red privada.

Reenvío: Operación que realizan tanto conmutadores como encaminadores. Consiste básicamente en encaminar un paquete recibido por la línea de entrada en base a unos campos que contiene el paquete.

Retransmisión de tramas: Forma de conmutación de paquetes basada en el uso de tramas del nivel de enlace. No existe capa de red.

RSVP: *Resource reSerVation Protocol.* Ver protocolo de reserva de recursos.

TCP/IP: Pila de protocolos de Internet.

Tipo de servicio (ToS): Campo de la cabecera IP utilizado por los elementos de la red para realizar una solicitud con un determinado nivel de QoS.

Unidad de datos del protocolo: Conjunto de datos especificado en un protocolo en un nivel dado. Está compuesto por datos de control del protocoo y datos de usuario.

Unidifusión *(Unicast)*: Dirección que es reconocida por un sólo sistema anfitrión.

X.25: Primera red internacional normalizada de conmutación de paquetes. X.25 se diseñó para ser una red datos pública a nivel mundial. Tecnología orientada a conexión para la transmisión en medios no fiables.