

NetGUI: MPLS

Planificación y Despliegue de Redes y Servicios

Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación

Noviembre de 2018

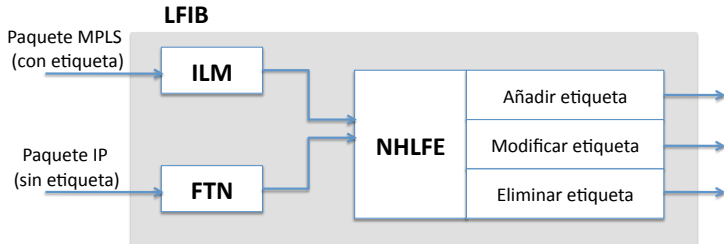


©2018 Grupo de Sistemas y Comunicaciones.
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike
disponible en <http://creativecommons.org/licenses/by-sa/3.0/es>

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)
- 3 MPLS en un router de salida (Egress LER)
- 4 Túnel MPLS

MPLS en Linux

- MPLS (Multi Protocol Label Switching) utiliza fundamentalmente 3 tablas:
 - NHLFE (Next Hop Label Forwarding Entry): gestiona las etiquetas de salida.
 - ILM (Incoming Label Map): relaciona etiqueta de entrada con un valor de la tabla NHLFE.
 - FTN (FEC-To-NHLFE Map): relaciona parámetros que identifican FEC con un valor de la tabla NHLFE.

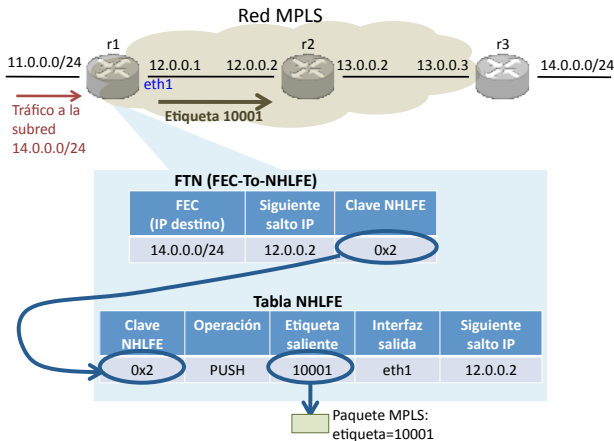


Contenidos

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)
- 3 MPLS en un router de salida (Egress LER)
- 4 Túnel MPLS

MPLS en el router de entrada (Ingress LER) (I)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- En la tabla NHLFE cada fila es un valor de etiqueta de salida diferente.
- En la tabla FTN cada fila es un FEC (IP destino) que está asociado a una fila de la tala NHLFE (clave)

MPLS en el router de entrada (Ingress LER) (II)

Sentido 11.0.0.0/24 a 14.0.0.0/24

- Añadir una etiqueta al tráfico de salida a través de una interfaz, añade una entrada en la tabla NHLFE:

```
mpls nhlfe add key 0 instructions push gen <etiqueta> \
    nexthop <interfaz> ipv4 <siguienteSalto>
```

```
NHLFE entry key NHLFE-Key mtu 0 propagate_ttl
(0 bytes, 0 pkts)
```

donde:

- **<etiqueta>** : Etiqueta MPLS de salida
- **<interfaz>**: Interfaz de salida
- **<siguienteSalto>**: Siguiete salto IP.

valor de retorno:

- **<NHLFE-Key>**: Clave en la tabla NHLFE para la entrada añadida

- Añadir asociación en FTN (FEC-To-NHLFE):

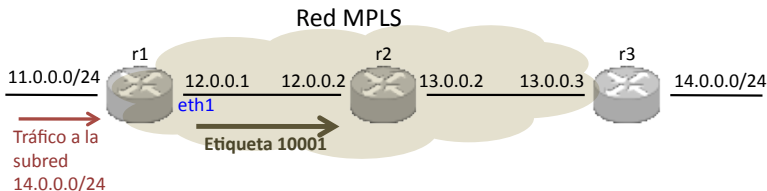
```
ip route add <subredDestino/máscara> via <siguienteSalto> mpls <NHLFE-Key>
```

donde:

- **<subredDestino/máscara>**: Destino IP que se desea alcanzar
- **<siguienteSalto>**: Siguiete salto IP.
- **<NHLFE-Key>**: Clave de la tabla NHLFE asociada a la etiqueta de salida.

MPLS en el router de entrada (Ingress LER) (III)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- Realizar la configuración en r1:
 - Añadir una etiqueta al tráfico de salida a través de una interfaz, añade una entrada en la tabla NHLFE:

```
r1:~# mpls nhlfe add key 0 instructions push gen 10001 nexthop eth1 ipv4 12.0.0.2

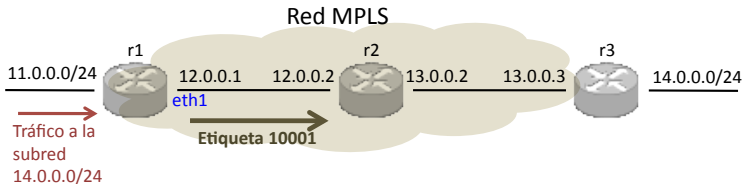
NHLFE entry key 0x00000002 mtu 0 propagate_ttl
(0 bytes, 0 pkts)
```

- Añadir asociación en FTN (FEC-To-NHLFE):

```
r1:~# ip route add 14.0.0.0/24 via 12.0.0.2 mpls 0x2
```


MPLS en el router de entrada (Ingress LER) (IV)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- Observar la configuración en r1:
 - Tabla de encaminamiento de r1:

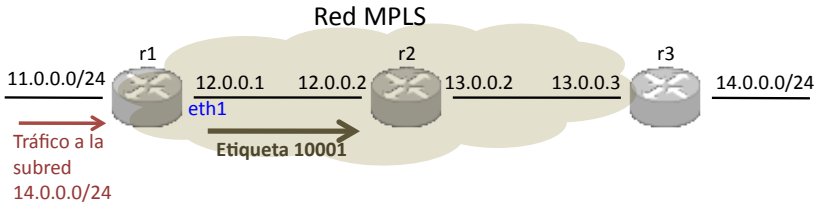
```
r1:~# ip route show
11.0.0.0/24 dev eth1 proto kernel scope link src 11.0.0.1
12.0.0.0/24 dev eth0 proto kernel scope link src 12.0.0.1
14.0.0.0/24 via 12.0.0.2 dev eth1 mpls 0x2
```

- Tabla NHLFE:

```
r1:~# mpls nhlfe show
NHLFE entry key 0x00000002 mtu 1496 propagate_ttl
push gen 10001 set eth1 ipv4 12.0.0.2
```

Script: MPLS en el router de entrada (Ingress LER) (V)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- Script de configuración (para la ejecución del comando `mpls nhlf` es necesario utilizar el acento inverso):

```
#!/bin/sh
echo "Envío a 14.0.0.0/24"
key_1='mpls nhlf add key 0 instructions push gen 10001 \
    nexthop eth1 ipv4 12.0.0.2 | grep key | cut -d " " -f 4'

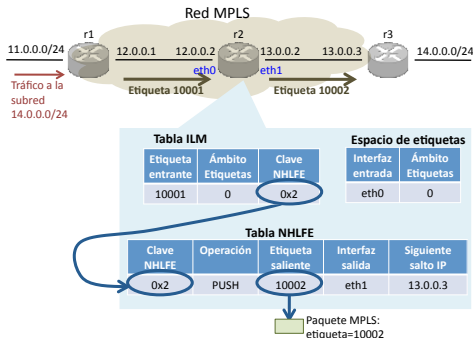
ip route add 14.0.0.0/24 via 12.0.0.2 mpls $key_1
```

Contenidos

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)**
- 3 MPLS en un router de salida (Egress LER)
- 4 Túnel MPLS

MPLS en un router intermedio (LSR) (I)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- En la tabla NHLFE cada fila es un valor de etiqueta de salida diferente.
- En la tabla “Espacio de etiquetas” se define para cada interfaz cuál es el ámbito de etiquetas recibidas a través de dicha interfaz. Si trabajamos con ámbito de router, todas las interfaces pertenecerán al mismo ámbito de etiquetas.
- En la tabla ILM cada fila es una pareja de valores (etiqueta de entrada, ámbito) asociado a un valor de etiqueta de salida en la tabla NHLFE.

MPLS en un router intermedio (LSR) (II)

Sentido 11.0.0.0/24 a 14.0.0.0/24

- Definir el espacio de etiquetas para los paquetes recibidos a través de una interfaz. Si se utiliza el mismo ámbito de etiquetas para todas las interfaces, este número será el mismo en todas ellas.

```
mpls labelspace set dev <interfaz> labelspace <ámbitoEtiqu>
```

donde:

- <interfaz>: Interfaz de entrada de paquetes MPLS
 - <ámbitoEtiqu>: número que identifica el ámbito de etiquetas para esa interfaz de entrada.
- Añadir entrada a la tabla ILM con la etiqueta de entrada y su ámbito de etiquetas.

```
mpls ilm add label gen <etiqueta> labelspace <ámbitoEtiqu>
```

donde:

- <etiqueta>: Etiqueta MPLS de entrada
 - <ámbitoEtiqu>: número que identifica el ámbito de etiquetas.

MPLS en un router intermedio (LSR) (III)

Sentido 11.0.0.0/24 a 14.0.0.0/24

- Añadir una etiqueta al tráfico de salida a través de una interfaz, añade una entrada en la tabla NHLFE:

```
mpls nhlfe add key 0 instructions push gen <etiqueta> nexthop <interfaz> ipv4 <sigSalto>

NHLFE entry key NHLFE-Key mtu 0 propagate_ttl
(0 bytes, 0 pkts)
```

- <etiqueta>: Etiqueta MPLS de salida
- <interfaz>: Interfaz de salida
- <sigSalto>: Siguiente salto IP.

valor de retorno:

- <NHLFE-Key>: Clave en la tabla NHLFE para la entrada añadida

- Añadir asociación en ILM-NHLFE:

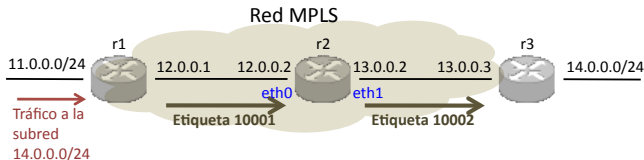
```
mpls xc add ilm_label gen <etiqueta> ilm_labelspace <ámbitoEtiqu> nhlfe_key <NHLFE-Key>
```

donde:

- <etiqueta>: Etiqueta MPLS de entrada
- <ámbitoEtiqu>: Ámbito de la etiqueta de entrada.
- <NHLFE-Key>: Clave de la tabla NHLFE asociada a la etiqueta de salida.

MPLS en un router intermedio (LSR) (IV)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- Espacio de etiquetas para la recepción en la interfaz eth0

```
r2:~# mpls labelspace set dev eth0 labelspace 0
```

- Añadir entrada a la tabla ILM la etiqueta de entrada y su ámbito.

```
r2:~# mpls ilm add label gen 10001 labelspace 0
```

- Añadir la etiqueta 10002 al tráfico de salida a través de eth1 en la tabla NHLFE:

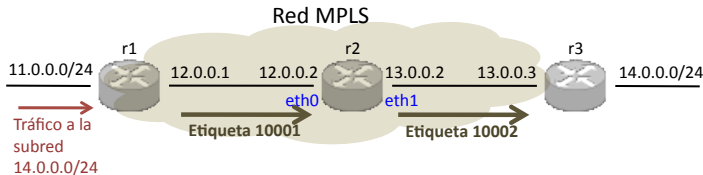
```
r2:~# mpls nhlfe add key 0 instructions push gen 10002 nexthop eth1 ipv4 13.0.0.3
NHLFE entry key 0x00000002 mtu 0 propagate_ttl
(0 bytes, 0 pkts)
```

- Añadir asociación en ILM-NHLFE:

```
r2:~# mpls xc add ilm_label gen 10001 ilm_labelspace 0 nhlfe_key 0x00000002
```

MPLS en un router intermedio (LSR) (V)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- Observar configuración en r2:

- Mostrar la tabla ILM:

```
r2:~# mpls ilm show
ILM entry label ten 10001 labelspace 0 proto ipv4
      pop forward key 0x00000002 (0 bytes, 0 pkts)
```

- Tabla NHLFE:

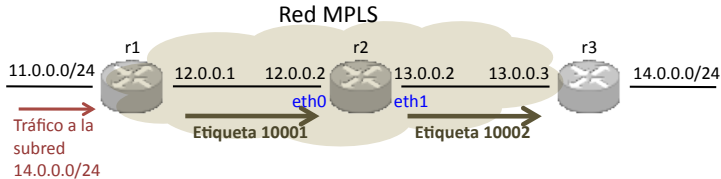
```
r2:~# mpls nhlfe show
NHLFE entry key 0x00000002 mtu 1496 propagate_ttl
      push gen 10002 set eth1 ipv4 13.0.0.3
```

- Tabla XC, asociación entre etiqueta recibida y clave de la tabla NHLFE:

```
r2:~# mpls xc show
XC entry ilm_label gen 10001 ilm_labelspace 0 nhlfe_key 0x00000002
```


Script: MPLS en el router intermedio (LSR) (VI)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- Script de configuración (para la ejecución del comando `mpls nhlfe` es necesario utilizar el acento inverso):

```
#!/bin/sh

echo "Envío a 14.0.0.0/24"

mpls labelspace set dev eth0 labelspace 0

mpls ilm add label gen 10001 labelspace 0

key_1='mpls nhlfe add key 0 instructions push gen 10002 \
nextthop eth1 ipv4 13.0.0.3 | grep key | cut -d " " -f 4'

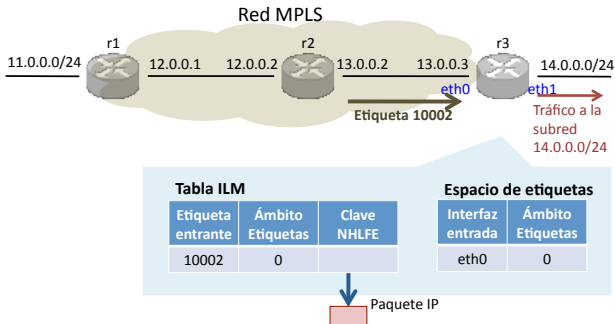
mpls xc add ilm_label gen 10001 ilm_labelspace 0 nhlfe_key $key_1
```

Contenidos

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)
- 3 MPLS en un router de salida (Egress LER)**
- 4 Túnel MPLS

MPLS en un router de salida (Egress LER) (I)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- En la tabla espacio de etiquetas se define para cada interfaz cuál es el ámbito de etiquetas recibidas a través de dicha interfaz. Si trabajamos con ámbito de router, todas las interfaces pertenecerán al mismo ámbito de etiquetas.
- En la tabla ILM cada fila es una pareja de valores (etiqueta de entrada, ámbito) asociado a un valor de etiqueta de salida en la tabla NHLFE. Como es un router de salida MPLS, no tendrá asociada ninguna entrada en la tabla NHLFE.

MPLS en un router de salida (Egress LER) (II)

- Definir el espacio de etiquetas para los paquetes recibidos a través de una interfaz. Si se utiliza el mismo ámbito de etiquetas para todas las interfaces, este número será el mismo en todas ellas.

```
mpls labelspace set dev <interfaz> labelspace <ámbitoEtiqu>
```

donde:

- <interfaz>**: Interfaz de entrada de paquetes MPLS
 - <ámbitoEtiqu>**: número que identifica el ámbito de etiquetas para esa interfaz de entrada.
- Añadir entrada a la tabla ILM con la etiqueta de entrada y su ámbito de Etiquetas.

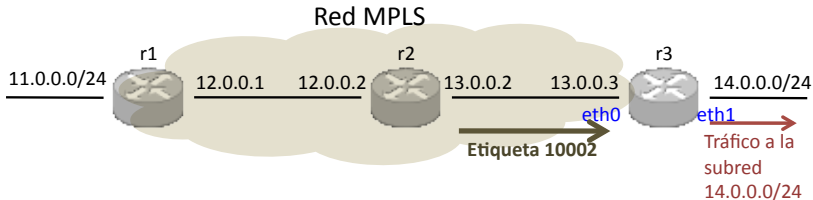
```
mpls ilm add label gen <etiqueta> labelspace <ámbitoEtiqu>
```

donde:

- <etiqueta>**: Etiqueta MPLS de entrada
 - <ámbitoEtiqu>**: número que identifica el ámbito de la etiqueta de entrada

MPLS en un router de salida (Egress LER) (III)

Sentido 11.0.0.0/24 a 14.0.0.0/24



- Espacio de etiquetas para la recepción en la interfaz eth0

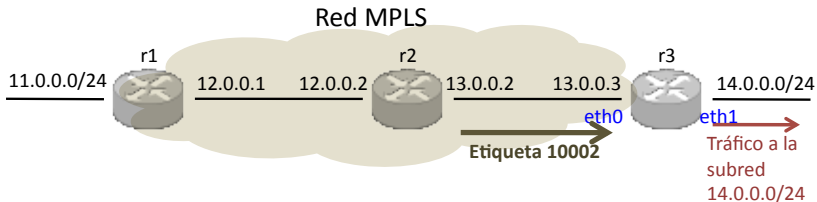
```
r3:~# mpls labelspace set dev eth0 labelspace 0
```

- Añadir entrada a la tabla ILM la etiqueta de entrada y su ámbito.

```
r3:~# mpls ilm add label gen 10002 labelspace 0
```

MPLS en un router de salida (Egress LER) (IV)

Sentido 11.0.0.0/24 a 14.0.0.0/24

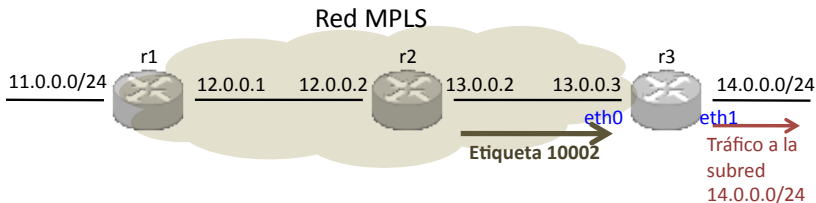


- Observar configuración en r3:
 - Mostrar la tabla ILM:

```
r3:~# mpls ilm show
ILM entry label gen 10002 labelspace 0 proto ipv4
      pop peek (0 bytes, 0 pkts)
```

Script: MPLS en el router de salida (Egress LER) (V)

Sentido 11.0.0.0/24 a 14.0.0.0/24



• Script de configuración:

```
#!/bin/sh

echo "Envío a 14.0.0.0/24"

mpls labelspace set dev eth0 labelspace 0

mpls ilm add label gen 10002 labelspace 0
```

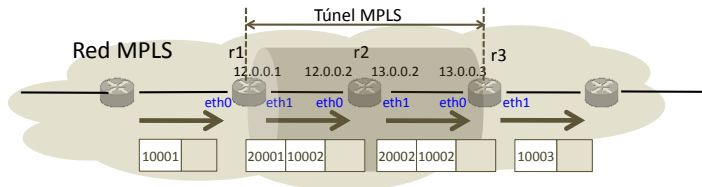
Contenidos

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)
- 3 MPLS en un router de salida (Egress LER)
- 4 Túnel MPLS**

Contenidos

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)
- 3 MPLS en un router de salida (Egress LER)
- 4 **Túnel MPLS**
 - **Entrada al túnel**
 - Reenvío en el túnel
 - Salida del túnel

Túnel MPLS: entrada al túnel (I)



- En r1 se crea un túnel MPLS:
 - El tráfico se recibe con etiqueta 10001 a través de eth0.
 - En la salida, se modifica la etiqueta 10001 por 10002.
 - Se añade una nueva etiqueta 20001, siendo esta etiqueta ahora la más externa.

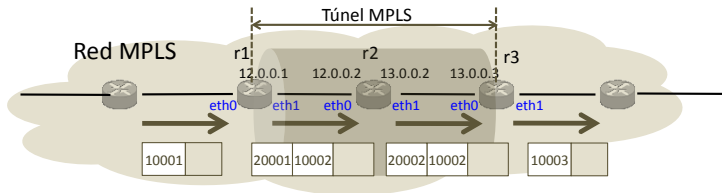
```

mpls labelspace set dev eth0 labelspace 0
mpls ilm add label gen 10001 labelspace 0
key_1='mpls nhlf add key 0 instructions push gen 20001 \\  
      nexthop eth1 ipv4 12.0.0.2 | grep key | cut -d " " -f 4'
key_2='mpls nhlf add key 0 instructions push gen 10002 \\  
      forward $key_1 | grep key | cut -d " " -f 4'
mpls xc add ilm_label gen 10001 ilm_labelspace 0 nhlf_key $key_2

```

- Se definen 2 entradas en la tabla NHLFE encadenadas:
 - Una entrada para modificar la etiqueta de 10001 a 10002 (véase la línea \$key_2). Y desde esa entrada se redirige el tráfico a otra entrada de NHLFE \$key_1.
 - Una entrada para añadir una etiqueta, 20001 (véase la línea \$key_1).

Túnel MPLS: entrada al túnel (II)



- Tabla ILM:

```
r1:~# mpls ilm show
ILM entry label gen 10001 labelspace 0 proto ipv4
      pop forward key 0x00000003 (0 bytes, 0 pkts)
```

- Tabla NHLFE:

```
r1:~# mpls nhlfe show
NHLFE entry key 0x00000003 mtu 1496 propagate_ttl
      push gen 10002 forward key 0x00000002 (0 bytes, 0 pkts)
NHLFE entry key 0x00000002 mtu 1496 propagate_ttl
      push gen 20001 set eth1 ipv4 12.0.0.2 (0 bytes, 0 pkts)
```

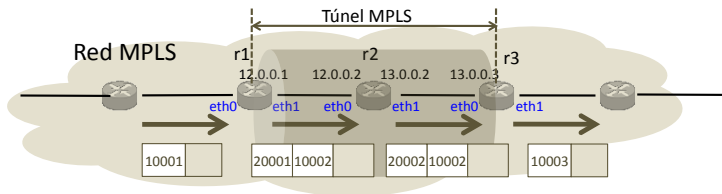
- Tabla XC, asociación entre etiqueta recibida y clave de la tabla NHLFE:

```
r1:~# mpls xc show
XC entry ilm_label gen 10001 ilm_labelspace 0 nhlfe_key 0x00000003
```

Contenidos

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)
- 3 MPLS en un router de salida (Egress LER)
- 4 **Túnel MPLS**
 - Entrada al túnel
 - **Reenvío en el túnel**
 - Salida del túnel

Túnel MPLS: reenvío en el túnel (I)

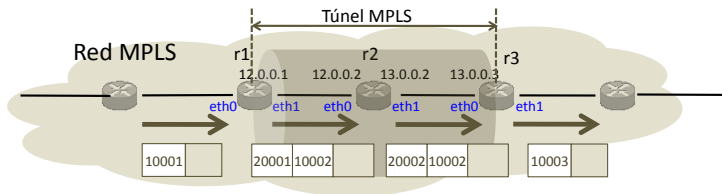


- En r2 se reenvía en el túnel MPLS:
 - El tráfico se recibe con etiqueta 20001 a través de eth0.
 - En la salida, se modifica la etiqueta 20001 por 20002.

```
mpls labelspace set dev eth0 labelspace 0
mpls ilm add label gen 20001 labelspace 0
key_1='mpls nhlf add key 0 instructions push gen 20002 \\  
        nexthop eth1 ipv4 13.0.0.3 | grep key | cut -d " " -f 4'
mpls xc add ilm_label gen 20001 ilm_labelspace 0 nhlf_key $key_1
```

- La configuración es similar a la de reenvío en un router MPLS sin túnel, pero ahora sólo se actúa sobre las etiquetas más externas: se recibe tráfico con etiqueta 20001 y se reenvía con etiqueta 20002.

Túnel MPLS: reenvío en el túnel (II)



- Tabla ILM:

```
r2:~# mpls ilm show
ILM entry label gen 20001 labelspace 0 proto ipv4
      pop forward key 0x00000002 (0 bytes, 0 pkts)
```

- Tabla NHLFE:

```
r2:~# mpls nhlfe show
NHLFE entry key 0x00000002 mtu 1496 propagate_ttl
      push gen 20002 set eth1 ipv4 13.0.0.3 (0 bytes, 0 pkts)
```

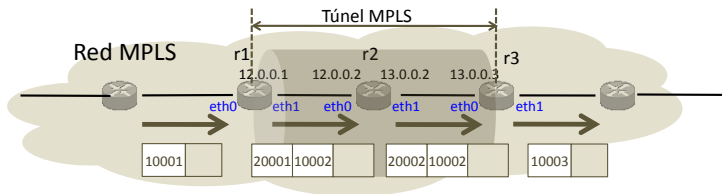
- Tabla XC, asociación entre etiqueta recibida y clave de la tabla NHLFE:

```
r2:~# mpls xc show
XC entry ilm_label gen 20001 ilm_labelspace 0 nhlfe_key 0x00000002
```

Contenidos

- 1 MPLS en el router de entrada (Ingress LER)
- 2 MPLS en un router intermedio (LSR)
- 3 MPLS en un router de salida (Egress LER)
- 4 Túnel MPLS**
 - Entrada al túnel
 - Reenvío en el túnel
 - **Salida del túnel**

Túnel MPLS: salida del túnel (I)

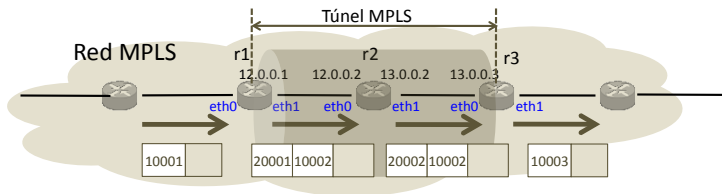


- En r3 se elimina el túnel MPLS y se reenvía según la siguiente etiqueta:
 - El tráfico se recibe con etiqueta 20002 a través de eth0. Se elimina.
 - El tráfico tiene la etiqueta 10002, se modifica la etiqueta 10002 por 10003.

```
mpls labelspace set dev eth0 labelspace 0
mpls ilm add label gen 20002 labelspace 0
mpls ilm add label gen 10002 labelspace 0
key_1='mpls nhlf add key 0 instructions push gen 10003 \\  
    nexthop eth1 ipv4 14.0.0.4 | grep key | cut -d " " -f 4'
mpls xc add ilm_label gen 10002 ilm_labelspace 0 nhlf_key $key_1
```

- Como la etiqueta 20002 no tiene asociada ninguna entrada NHLFE, esta etiqueta se elimina. La siguiente etiqueta que tiene la cabecera MPLS es 10002 que tiene asociada una entrada en NHLFE para que se reenvíe con la nueva etiqueta 10003.

Túnel MPLS: salida del túnel (II)



- Tabla ILM:

```
r3:~# mpls ilm show
ILM entry label gen 10002 labelspace 0 proto ipv4
    pop forward key 0x00000002 (0 bytes, 0 pkts)
ILM entry label gen 20001 labelspace 0 proto ipv4
    pop peek (0 bytes, 0 pkts)
```

- Tabla NHLFE:

```
r3:~# mpls nhlfe show
NHLFE entry key 0x00000002 mtu 1496 propagate_ttl
    push gen 10003 set eth1 ipv4 14.0.0.4 (0 bytes, 0 pkts)
```

- Tabla XC, asociación entre etiqueta recibida y clave de la tabla NHLFE:

```
r3:~# mpls xc show
XC entry ilm_label gen 10002 ilm_labelspace 0 nhlfe_key 0x00000002
```

Referencias

- MPLS for Linux
(<http://sourceforge.net/apps/mediawiki/mpls-linux/>)
- MPLS-Linux labs
(<http://ontwerpen1.khlim.be/lrutten/cursussen/comm2/mpls-linux-docs/>)