

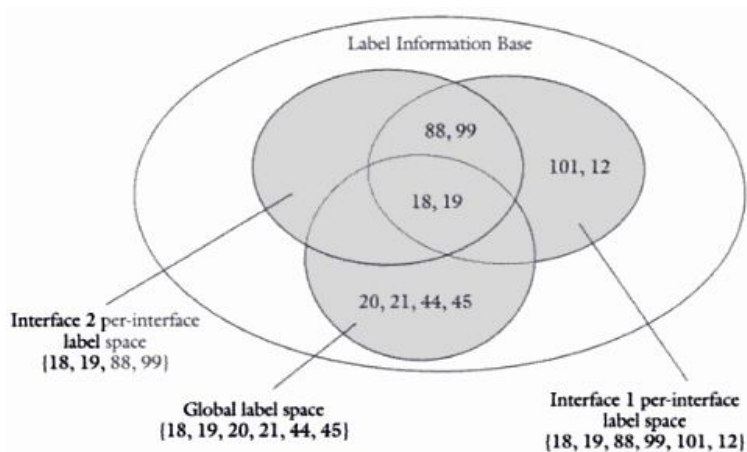
Redes de Computadores II

Objetivos:

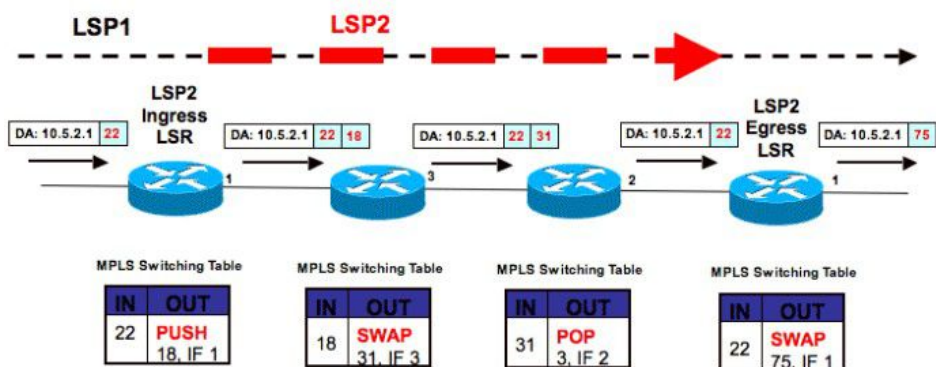
- i) Entender *label spaces* MPLS
- ii) Criar túneis MPLS

Introdução

No MPLS, *labels* podem ser definidos por interface ou por plataforma. Isto significa que em cada LSR há conjuntos de *labels* chamados de *labelspaces*, de forma que *labels* de diferentes *labelspaces* são independentes. Por exemplo, o *label* de valor 1000 dentro do *labelspace* 0 é diferente do *label* 1000 no *labelspace* 1. Cada *labelspace* é associado a uma ou mais interfaces: um *label* recebido por uma interface é classificado no *labelspace* a ela associado, como pode ser visto na figura abaixo:



O mecanismo de empilhamento de *labels* no MPLS possibilita naturalmente a criação de túneis. Um túnel funciona como um enlace ponto-a-ponto entre dois LSR (que podem estar a muitos saltos de distância um do outro), dentro do qual trafegam um ou mais LSPs. O empilhamento de *labels* está exemplificado abaixo:



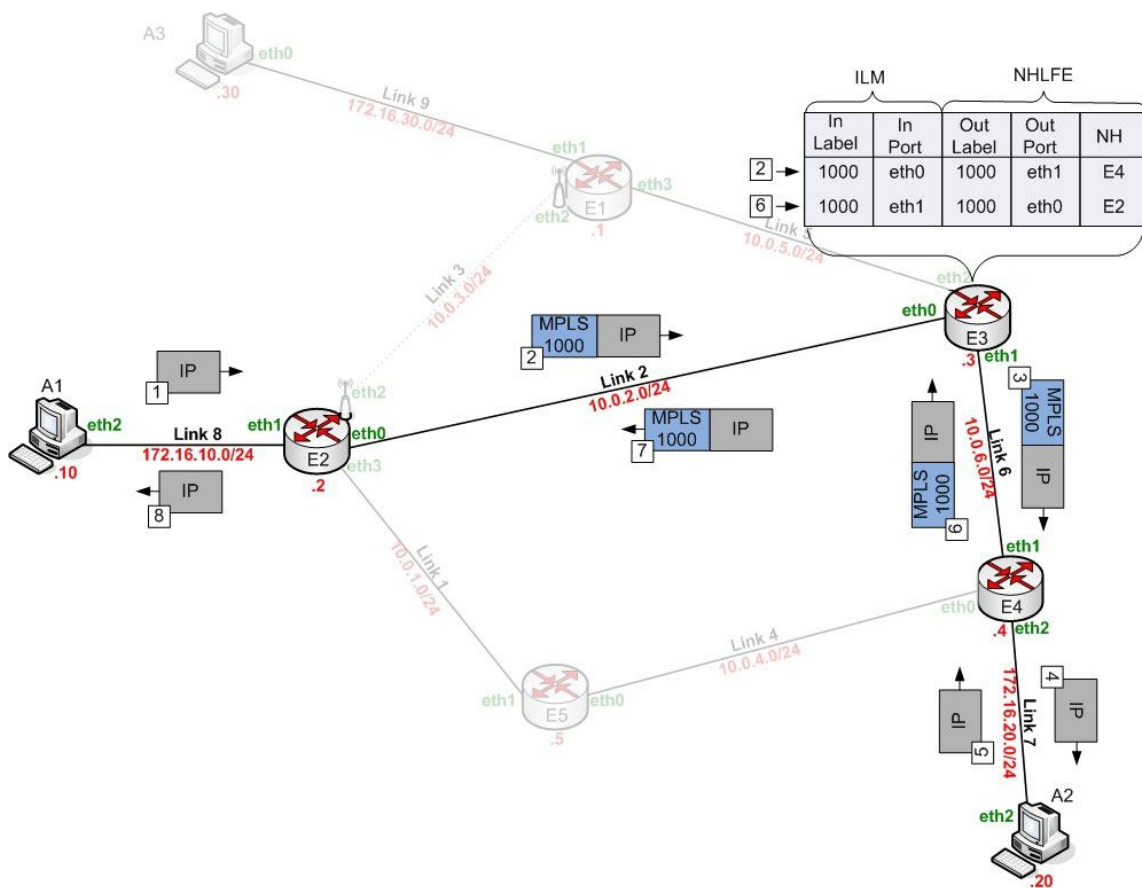
Na figura acima, o LSP1 trafega dentro do LSP2, como se o LSP2 fosse um enlace ponto-a-ponto entre o 1º e o 3º roteador a partir da esquerda. Veja que ao ingressar no LSP2, a PDU que tinha o *label* 22 tem acrescentado a si um outro *label* 18, mantendo o *label* 22 intocado. Ao longo de todo o LSP2 o *label* 22 permanece intocado. Os roteadores usam o *label* mais externo (18, 31 e 3,

nessa sequência) para fazer a comutação dentro do LSP2. Por fim, veja que as operações realizadas nos roteadores são chamadas de PUSH (acrescentar um label), SWAP (trocar o label) e POP (retirar um label).

Túneis são úteis para a criação de VPN (Virtual Private Networking), e também para fazer TE (Traffic Engineering, ou Engenharia de Tráfego). No caso de TE, a utilidade reside em poder agregar em um único LSP vários LSPs aos quais se quer dar o mesmo tratamento quanto a QoS. Já para VPN o atrativo é hierarquizar o roteamento MPLS, o que possibilita criar hierarquias de transmissão dentro de uma WAN.

Roteiro

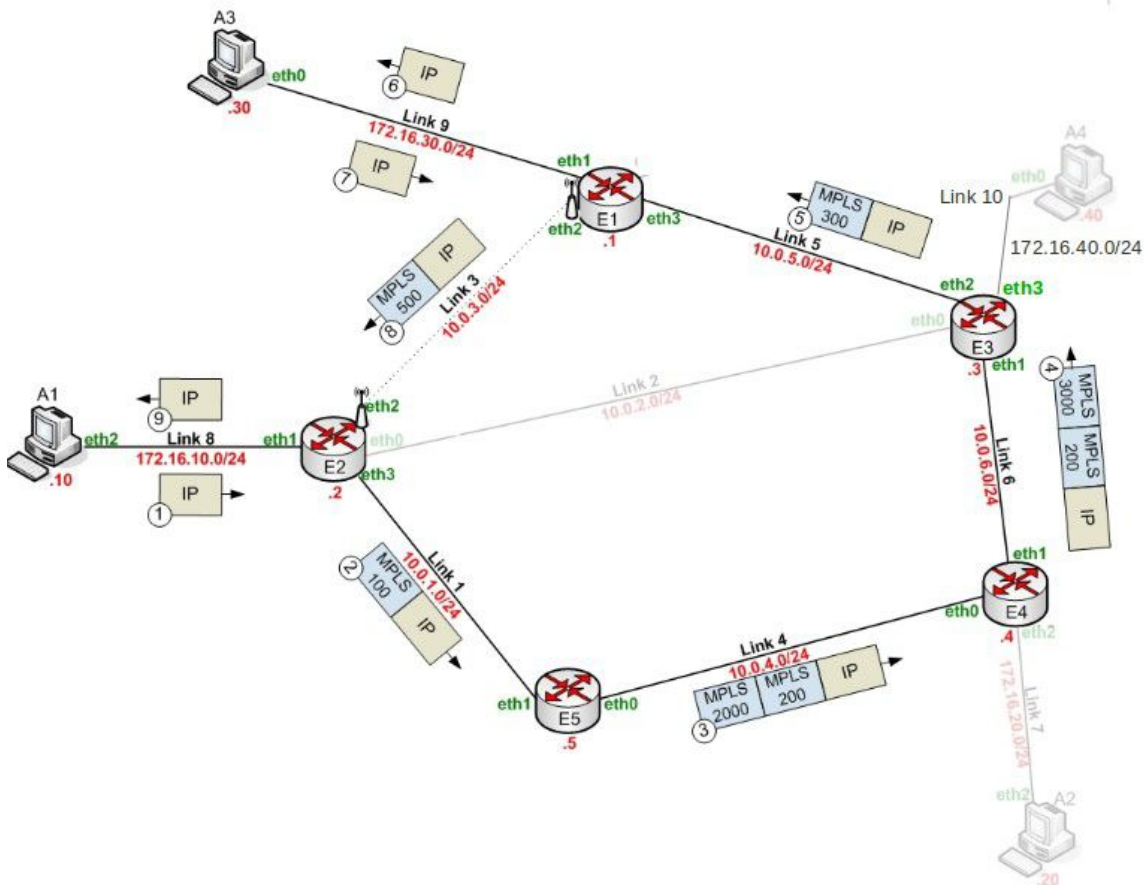
Neste roteiro vamos criar uma rede MPLS, e trabalhar com a definição de labelspaces e túneis. Esse experimento, entre outros, foi obtido do projeto MPLS-Linux, e uma descrição completa e bastante detalhada se encontra em <http://www.sj.ifsc.edu.br/~msobral/RCO2/docs/mpls-linux>.



1. O Netkit já está instalado nos computadores do laboratório. Você precisa apenas obter a configuração do laboratório virtual, que já está na aula de hoje
2. Execute o primeiro experimento sobre labelspaces com o `gnome-netkit`
3. No computador A1 tente pingar o computador A2, e vice-versa.
4. Enquanto o **ping** está em execução, analise o tráfego usando o **tcpdump** em todos

os nós da rede. Por exemplo, para ver o tráfego na interface *eth0* do roteador *E2* execute “**tcpdump -i eth0 -lne**”. Observe as informações sobre os rótulos, e a comutação de rótulos nos roteadores.

5. Você pode inspecionar as informações da FIB contidas nos roteadores. Execute o comando “**mpls nhlfe show**” em cada roteador para ver as entradas NHLFE. Para ver o mapeamento entre rótulos de entrada e de saída, execute “**mpls ilm show**”. Para ver a tabela de comutação, execute “**mpls xc show**”.
6. Veja que o LSR *E3* usa os mesmos *labels* em ambas interfaces, e em ambos LSP (o que leva de *E2* até *E4*, e vice-versa). Isto foi possível porque foram criados diferentes *labelspaces* para as interfaces *eth0* e *eth1* de *E3*. Com isto, usou-se um único *label* para identificar um par de LSPs entre *E2* e *E4*. O que se pode concluir em relação a *labelspaces* e *rótulos* ?
7. Obtenha o segundo arquivo de configuração, agora sobre túneis MPLS:
8. Esse experimento implementa a rede abaixo:



9. Nessa rede há um túnel entre *E3* e *E5*, que transporta internamente o *label* 200. Há assim um LSP entre *E5* e *E3*, que transporta as PDUs do LSP entre *E1* e *E2*. Use o **tcpdump** nos diversos roteadores para ver o empilhamento dos *labels* no túnel.
10. Crie um LSP entre *A1* e *A4*, e adicione-o ao túnel existente.
11. Faça com que o LSP entre *A3* e *A1* passe por dentro do túnel.