UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

FHILIPE GUIMARÃES DE LINHARES

Avaliação de desempenho de VPNs sobre redes MPLS-Linux

Trabalho de Graduação.

Prof. Dr. Sérgio Luis Cechin
Orientador

Porto Alegre, dezembro de 2010.

AGRADECIMENTOS

Primeiramente, gostaria de agradecer a minha família e a minha esposa, pois sempre me apoiaram e deram a sustentação necessária para que eu pudesse superar as barreiras impostas até a conclusão deste curso.

Agradeço ao professor Sérgio Luis Cechin pelas suas valiosas orientações e por partilhar seu conhecimento e experiência na produção deste trabalho.

Por último, agradeço aos professores e funcionários do Instituto de Informática da UFRGS por propiciarem diversos momentos de aprendizado e trabalho “duro”.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

ATM	Asyncronous Transfer Mode
CE	Client Edge Device
CRS	Cell Switching Router
DUT	Device Under Test
LSP	Label Switch Path
EOMPLS	Ethernet Over MPLS
EXP	Experimental
FCS	Frame Check Sequence
FEC	Forwarding Equivalence Class
FIB	Forwarding Information Base
FR	Frame Relay
IETF	Internet Engineering Task Force
ILM	Incoming Label Map
IP	Internet Protocol
LAN	Local Area Network
LBS	Label Based Switching
LDP	Label Distribution Protocol
LER	Label Edge Router
LSE	Label Stack Entry
LSR	Label Switch Router
LIB	Label Information Base
MPLS	Multiprotocol Label Switching
MTU	Maximum Transfer Unit
NHLFE	Next Hop Label Forwarding Entry
OSI	Open Systems Interconnection
PC	Personal Computer
PE	Provider Edge Router

RFC	Request For Comments
SDH	Synchronous Digital Hierarchy
TDM	Time-Division Multiplexing
VPN	Virtual Private Network
VRF	VPN Routing and Forwarding

LISTA DE FIGURAS

LISTA DE TABELAS

RESUMO

As redes MPLS (Multiprotocol Label Switching) estão sendo largamente utilizadas e comercializadas pelas companhias de telecomunicação no mercado brasileiro. O MPLS é disponibilizado pelos principais fabricantes de dispositivos de comutação de pacotes e roteadores. Desta forma, o MPLS pode ser considerado uma tecnologia que está consolidada no mercado e com grande potencial de expansão. Neste contexto, este trabalho tem como objetivo fazer uma avaliação de desempenho da implementação do protocolo MPLS no Linux. Esta implementação é de domínio público e destina-se a microcomputadores com sistema operacional do tipo Unix. A avaliação de desempenho terá como foco dos experimentos as MPLS VPNs. Para melhor entendimento dos experimentos, serão apresentados os conceitos fundamentais do MPLS, bem como suas principais características e aplicações. Juntamente com o MPLS serão apresentados os conceitos fundamentais das Redes Privadas Virtuais (VPNs), o objetivo dessa abordagem é mostrar como essas duas tecnologias podem se "encaixar" para formar as MPLS VPNs. Após, serão apresentadas as técnicas para a avaliação de desempenho e as métricas que foram utilizadas nos testes. Reservou-se um capítulo para mostrar as definições que foram utilizadas para a configuração do ambiente de testes utilizado durante os experimentos. Por fim, serão apresentados os resultados obtidos nas medições e o que se pode concluir a respeito deles.

Palavras-Chave: MPLS, VPN, Avaliação de desempenho, MPLS-Linux.

Performance Analysis of VPNs over MPLS-Linux Networks

ABSTRACT

The MPLS (Multiprotocol Label Switching) networks are being widely used and marketed by telecom companies. MPLS is provided by leading manufacturers of routers and switch routers. Thus, the MPLS can be considered a technology that is consolidated in the market and with huge potential for expansion. In this context, this work aims to make a performance analysis of the implementation of the MPLS-Linux. This implementation is public domain and is intended for computers with Unix-type operating system. The performance analysis experiments will focus in MPLS VPNs. For a better understanding of the experiments, will be present the basic concepts of MPLS and its main characteristics and applications. Along with the MPLS, will be present the fundamental concepts of Virtual Private Networks (VPNs), the aim of this approach is to show how these two technologies can "fit" to form the MPLS VPNs. Next, will be shown techniques for performance analysis and the metrics that were used in the tests. Has reserved a section to display the definitions that were used for configuring the test environment used during the experiments. Finally, will be presented the results from the measurements and what can be concluded about them.

Keywords: MPLS, VPN, network benchmark, MPLS-Linux.

1 INTRODUÇÃO

O MPLS (*Multiprotocol Label Switching*) é um protocolo de transporte baseado em pacotes rotulados, onde cada rótulo representa um índice na tabela de encaminhamento para o próximo *hop*. Pacotes com o mesmo rótulo e mesma classe de serviço são indistinguíveis entre si e por isso recebem o mesmo tipo de tratamento.

As redes MPLS não têm o objetivo de se conectar diretamente a sistemas finais. Ao invés disto elas são redes de trânsito, que transportam pacotes entre pontos de entrada e de saída. O MPLS é padronizado pelo IETF através da RFC-3031 e ele é chamado de Multiprotocolo, pois pode ser utilizado com qualquer protocolo da camada 3 do modelo OSI. Na arquitetura do modelo OSI, ele é referido com um protocolo intermediário entre as camadas 2 e 3, muitos chamam de camada 2,5.

O MPLS surgiu como uma resposta de fabricantes de equipamentos e centros de pesquisa a várias necessidades que surgiram com a popularização da internet e diversificação de seus serviços.

Talvez a mais relevante dessas necessidades seja a sobrecarga que vem sendo aplicada sobre os roteadores tendo em vista o crescente número de usuários. Os algoritmos de roteamento utilizados nos Roteadores IP são ineficientes à medida que a rede aumenta, pois para definir qual é o próximo salto (*hop*) do pacote, cada roteador, geralmente, precisa analisar mais informações do que realmente seria necessário para fazer o encaminhamento do pacote. Além disso, cada roteador realiza o mesmo processamento para todos os pacotes, o qual é muito semelhante em cada um deles e sem guardar nenhum tipo de memória sobre cada pacote. Esse tipo de processo é particularmente ineficiente, pois a maioria dos pacotes IP pertence a fluxos com mesma origem e destino. [DEA 2002]

Com base nesses fatos chegou-se a conclusão que redes baseadas em algoritmos de roteamento IP padrão não são escaláveis. Ou seja, não é possível aumentar o tamanho da rede indefinidamente, pois, por mais rápido que sejam os roteadores individualmente, a repetição das tarefas de roteamento tornaria o atraso da rede proibitivo. [DEA 2002]

Desta forma, ficou evidente a necessidade de um novo algoritmo de encaminhamento. Entretanto, não bastaria que fosse eficiente, ele deveria ser compatível com os protocolos e equipamentos já existentes. Além disso, soma-se a necessidade de novas funcionalidades de roteamento como, por exemplo, as classes de serviços para atender fluxos de vídeo ou voz sobre IP. [DEA 2002]

Com base no contexto do crescente uso do protocolo MPLS e escasso número de análises de desempenho sobre ele, o presente trabalho tem como objetivo fazer uma avaliação de desempenho do protocolo MPLS na implementação para o sistema

operacional Linux. O cenário que será o foco da avaliação será de MPLS VPNs. Para isso, primeiramente será mostrado o funcionamento do protocolo e como ele pode ser utilizado para criar uma VPN. A metodologia de medição que será utilizada seguirá as recomendações das seguintes RFCs: *RFC- 2544 Benchmarkings Methodology for Network Interconnect Devices, RFC 5695 - MPLS Forwarding Benchmarking Methodology for IP Flows, RFC 3032 - MPLS Label Stack Encoding.*

Neste trabalho, supõe-se que o leitor tenha conhecimentos mínimos de protocolos de roteamento, protocolo IP, redes Ethernet e Shell scripts em Linux. Se esse não for o caso, sugere-se as seguintes referências: KUROSE, J. F., & ROSS, K. W. **Redes de Computadores e a Internet** ou TANENBAUM, Andrew S. **Redes de Computadores**.

2 CONCEITOS INICIAIS

O MPLS é uma tecnologia consolidada no mercado e tem se mostrado uma tendência. O uso de redes MPLS proporciona diversas vantagens como à possibilidade de migração de tecnologias mais antigas como, por exemplo, Frame-Relay e, além disso, possibilita agregar novos serviços. Entre os principais serviços disponíveis está o de Redes Privadas Virtuais, ou VPNs. [PRE 2008]

2.1 Histórico do MPLS

Quando o ATM foi proposto, esperava-se que ele acabasse por dominar o cenário mundial devido às suas altas velocidades. Porém a tecnologia ATM possuía uma desvantagem significativa, ela não era compatível com o IP, o protocolo de rede mais difundido do mundo. Nesse contexto, ocorreram várias tentativas de compatibilizar o ATM com IP. Uma destas tentativas foi o ATMARP (*ATM Address Resolution Protocol*), este protocolo mapeia os endereços IP em endereços ATM para que pacotes IP possam ser roteados através de uma nuvem ATM. Porém o ATMARP possuía várias desvantagens. Entre elas está a impossibilidade de conexão direta entre endereços IP de diferentes sub-redes. Para isto era necessário que os pacotes passassem por roteadores, mais lentos do que os comutadores ATM, formando assim gargalos na rede. [DEA 2002]

Para contornar o problema com os roteadores foi desenvolvido o NHRP (*Next Hop Routing Protocol*). O NHRP funciona utilizando NHS (*Next Hop Servers*) nos roteadores IP. Quando um NHC (*Next Hop Client*) manda um pacote de *request* para uma sub-rede IP, os NHS roteiam este pacote até o roteador final. O NHS do roteador final transforma o endereço IP final em um endereço ATM e manda de volta para o NHC. Assim, os próximos pacotes IP poderão ser encaminhados inteiramente dentro da rede ATM. Mas o NHRP também apresentava sérias desvantagens. Percebeu-se que a interoperabilidade entre IP e ATM não seria uma tarefa simples. A partir desse momento algumas empresas iniciaram tentativas de desenvolver um protocolo que resolvesse os problemas apresentados pela conexão IP e ATM. Surgiram então, o IP Switching, da Ipslon, ARIS (*Aggregate Route-Based IP Switching*) da IBM, *CSR (Cell Switching Routers*) da CISCO que em seguida foi chamado de *Tag Switching* e a partir daí iniciaram-se esforços independentes pela padronização de suas arquiteturas. Porém, foi criado um grupo de trabalho para lidar com todas as tecnologias que estavam sendo desenvolvidas para garantir a interoperabilidade. Deste grupo de trabalho nasceu o MPLS. [DEA 2002]

2.2 Multiprotocol Label Switching - MPLS

O MPLS é um protocolo de transporte que faz a comutação de pacotes rotulados (labels). A Rede MPLS é utilizada, geralmente, nos backbones das empresas de Telecomunicações, pois por definição é uma rede de transporte que carrega os pacotes de um ponto de entrada até um ponto de saída da rede. Além disso, provê a capacidade

de gerir serviços de redes de computadores como velocidade, escalabilidade, gerenciamento de qualidade de serviço (QoS) e contempla a necessidade de engenharia de tráfego.

A Figura 1 exemplifica o funcionamento básico do MPLS, ou seja, o transporte de pacotes através da comutação de pacotes rotulados. A comutação dos pacotes é feita pelos roteadores através de um mapeamento label-to-label. Os labels são inseridos quando os pacotes entram na Rede MPLS, a partir desse momento, os roteadores vão encaminhando os pacotes apenas verificando o Label e não mais o endereço IP, cada vez que o pacote é repassado o rótulo é substituído até que o pacote chegue ao roteador de saída da rede MPLS.

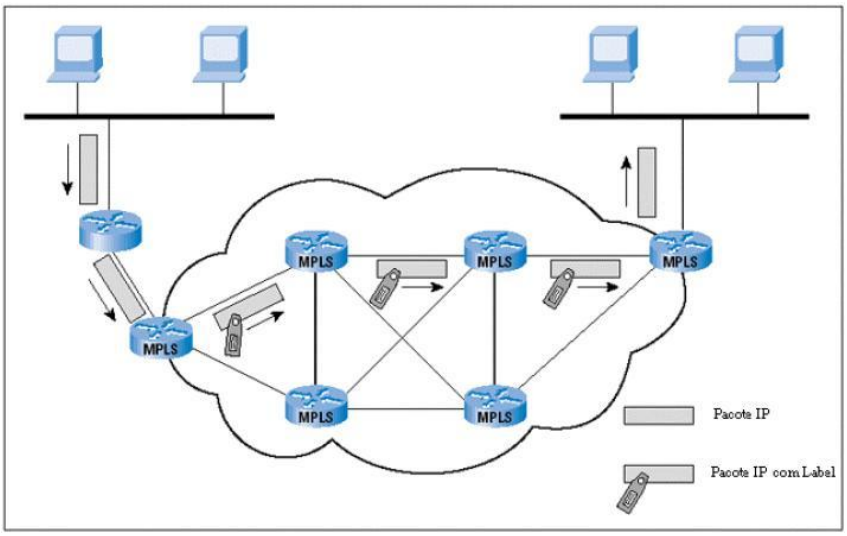


Figura 1 - Funcionamento básico do MPLS

2.2.1 Vantagens do MPLS

2.2.1.1 Velocidade

A comutação de pacotes IP em uma CPU é mais lenta que a comutação por labels a qual só verifica o label no início do pacote. Normalmente um roteador encaminha um pacote analisando o IP destino e localiza a melhor rota para o pacote. Essa pesquisa pela melhor rota depende da implementação do algoritmo de roteamento dentro do roteador. Contudo, como endereços IP podem ser de multicast e unicast essa pesquisa pode ser complexa e a tomada de decisão pode levar um tempo considerável. Entretanto, com a evolução dos roteadores e sua capacidade de processamento essa vantagem do MPLS acaba sendo questionável. [GHE 2007]

2.2.1.2 A utilização de uma infra-estrutura de rede unificada

Com o MPLS é possível comutar o tráfego de diferentes protocolos através de uma infra-estrutura comum. A idéia é rotular o pacote na entrada da rede MPLS de acordo com seu destino ou por um critério pré-definido e chavear todo o tráfego através desta infra-estrutura comum. Essa é a grande vantagem do MPLS.

Usando o MPLS com IP, pode-se estender as possibilidades do que se pode transportar. O que era possível com Frame Relay e ATM Layer 2, agora pode ser feito com o MPLS, como por exemplo: transportar IPv4, IPv6, Ethernet, HDLC, PPP e outras tecnologias da camada 2. [GHE 2007]

2.2.1.3 Peer-to-Peer VPN Model versus Overlay VPN Model

Uma VPN é uma rede que emula uma rede privada sobre uma infra-estrutura de rede pública. Uma rede privada exige que todos os locais possam se interligar, e ao mesmo tempo devem estar completamente isoladas das outras VPNs. Uma VPN normalmente pertence a uma empresa e tem vários locais interconectados através da infra-estrutura comum do prestador de serviços. Os prestadores de serviços podem implantar dois modelos de serviços: VPN Overlay Model e VPN Peer-to-Peer. [GHE 2007]

VPN Overlay Model

O prestador de serviços fornece um serviço de ligações ponto-a-ponto ou de circuitos virtuais através de sua rede entre os roteadores do cliente. Os roteadores dos clientes criam uma conexão direta pelo link ou circuito virtual entre eles, através da infra-estrutura do prestador de serviços. Os roteadores e switches do prestador de serviços carregam os dados dos clientes através de sua rede, mas nenhuma interconexão de roteamento ocorre entre roteador cliente e do provedor de serviços. O resultado disso é que os roteadores do provedor de serviços nunca enxerga as rotas dos clientes. [4]

Estes serviços ponto-a-ponto podem ser na Camada 1, 2 ou até na 3. Exemplo na Camada 1 são por TDM (time-division multiplexing), E1, E2, SONET e links SDH. Exemplo na Camada 2 x.25, ATM ou Frame-Relay. [GHE 2007]

Na Figura 2 é mostrado um exemplo de rede Overlay sobre Frame-Relay, pode-se observar que na rede do provedor de serviços estão os switches Frame-Relay os quais criam os circuitos virtuais entre os roteadores dos clientes (Customer Router) na borda da rede Frame-Relay.

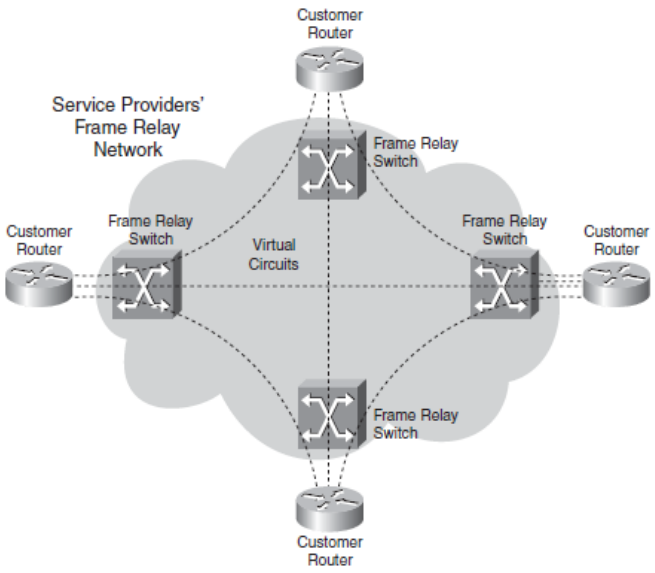


Figura 2 - Rede Overlay sobre Frame-Relay [GHE 2007]

VPN Peer-to-Peer Model

Nesse modelo os roteadores do prestador de serviços carregam os dados do cliente em toda a rede e também participam do roteamento dos pacotes. Em outras palavras, os roteadores do prestador de serviços conectam-se diretamente com os roteadores dos clientes em nível de camada 3. O resultado disso é que existe um protocolo de roteamento entre os roteadores do cliente e do provedor.

Na Figura 3, pode-se identificar a conexão direta entre os roteadores dos clientes (Customer Edge Routers) e os roteadores do provedor de serviço (Provider Edge Routers). Este tipo de configuração, aliado as características citadas acima, caracteriza o modelo de VPNs Peer-to-Peer.

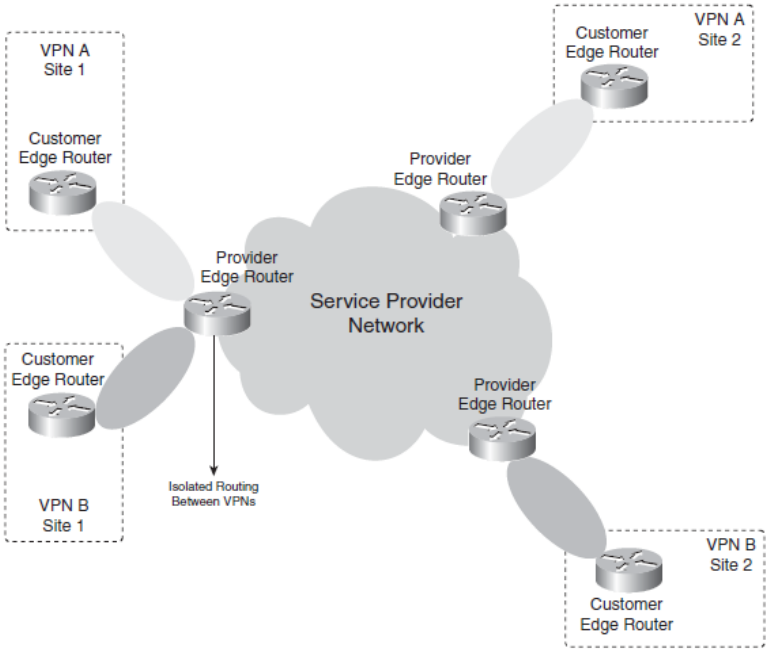


Figura 3 - Peer-to-Peer VPN Model [GHE 2007]

Antes de existir o MPLS, para implementar o modelo de VPN Peer-to-Peer era necessário criar as rotas IP entre os roteadores do cliente e do provedor e para obter o isolamento era necessário a criação de filtros de pacotes (listas de acesso). Assim era muito mais comum o modelo implantado ser o Overlay. Contudo, o advento das MPLS VPNs permitiu que a implantação do modelo VPN Peer-to-Peer fosse muito mais fácil. Adicionar ou remover locais é agora mais fácil de configurar e assim demanda menos tempo e esforço. Isso será melhor explicado na seção que trata das MPLS VPN. [GHE 2007]

2.2.1.4 Engenharia de Tráfego

A idéia básica por trás da engenharia de tráfego é de otimizar a utilização da infra-estrutura da rede, principalmente os links que estão subutilizados. Isto significa que a engenharia de tráfego deve prover a capacidade de direcionar o tráfego por caminhos diferentes do preferencial, que no caso de roteamento IP é o caminho de menor custo. O caminho de menor custo é o caminho calculado pelo protocolo de roteamento dinâmico. Com a engenharia de tráfego implementada em redes MPLS, pode-se ter o tráfego que é destinado para um prefixo específico ou com uma determinada qualidade de serviço

indo a partir de um ponto A até um ponto B ao longo de um caminho que é diferente do caminho de menor custo. O resultado é que o tráfego pode ser distribuído mais uniformemente nos links disponíveis da rede e assim utilizar de melhor forma links subutilizados. A Figura 4 mostra um exemplo disso, o tráfego do ponto A até o ponto B é chaveado para um caminho que é diferente do caminho de menor custo calculado pelo algoritmo de roteamento IP. [GHE 2007]

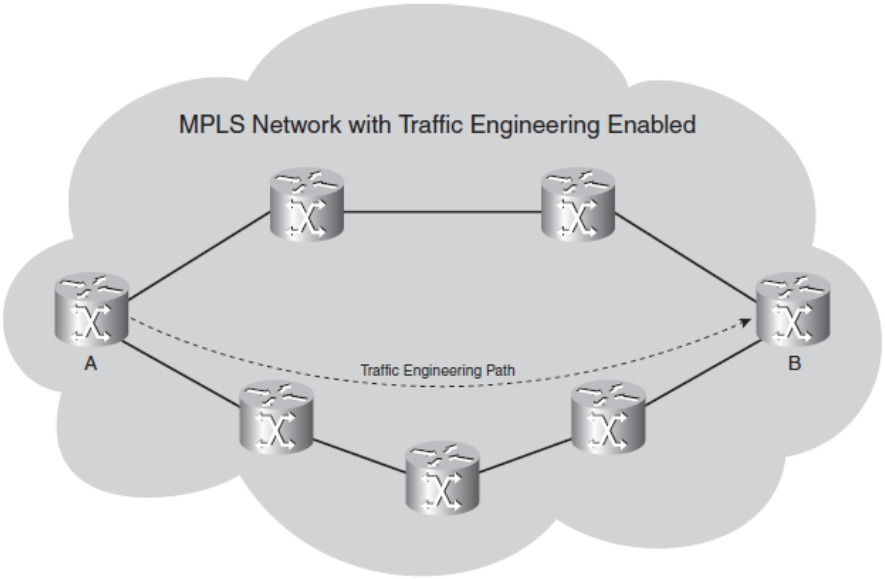


Figura 4 - Exemplo de Engenharia de Tráfego [GHE 2007]

Uma vantagem extra de usar Engenharia de Tráfego com MPLS é a possibilidade do uso de *Fast ReRouting* (FRR). FRR lhe permite mudar o roteamento do tráfego rotulado em torno de um link ou roteador que se tornou indisponível. Este *rerouting* do tráfego ocorre em menos de 50 ms, que é rápido mesmo para os padrões de hoje. [GHE 2007]

2.2.2 O rótulo MPLS

O *header* MPLS é um campo de 32 bits com a estrutura abaixo, mostrado na Figura 5.

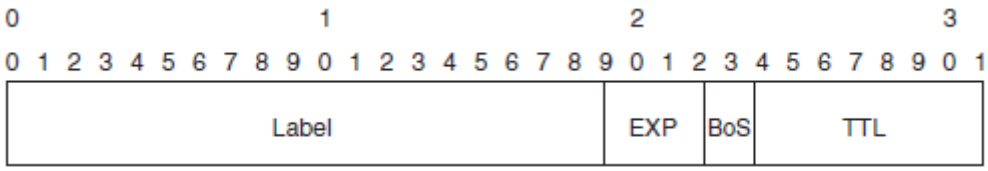


Figura 5 - Sintaxe de um Label MPLS

Os seguintes campos compõem um Label MPLS:

• Label (20 bits): valor atual do *Label*, identificador de LSP.

• EXP (3 bits): experimental bits – pode ser usado para filas de espera, rejeição, QoS etc.

• BoS (1 bit): bit de sinalização de fim de *stack*, este valor é setado para 1 para a última entrada na pilha e 0 para as demais.

• TTL (8 bits): *time to live*, possui a mesma função do TTL do cabeçalho IP.

2.2.3 Empilhamento de Labels

Durante o roteamento de um pacote os roteadores podem precisar de mais de um Label pata efetuar o encaminhamento do pacote. Para resolver esse problema é utilizado um mecanismo de empilhamento de labels que permite operações hierárquicas, ou seja, cada nível de uma pilha de labels representa um nível hierárquico. Assim, cada pilha possui um top-label e um bottom-label e o número de entradas entre eles é livre. A Figura 6 mostra a estrutura da pilha de labels.

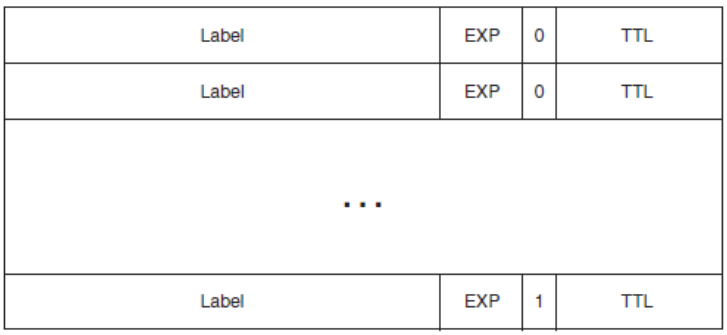


Figura 6 - Pilha de Labels

Na pilha apenas o bottom-label possui o campo BoS setado em um, indica que é o fim da pilha.

2.2.4 Encapsulamento dos labels MLPS

A pilha de labels MPLS é inserida na frente do cabeçalho de Camada 3 do pacote, isto é, antes do cabeçalho do protocolo de transporte e após o cabeçalho do protocolo da Camada 2, a Figura 7 mostra a colocação da pilha de labels em um quadro da Camada 2.

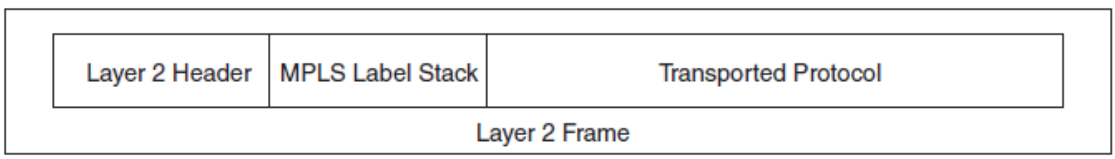


Figura 7 - Encapsulamento do Label MPLS [GHE 2007]

2.2.5 Label Switch Router - LSR

Um Label Switch Router é um roteador que suporta MPLS. Ele é capaz de entender labels MPLS e de receber e transmitir pacotes rotulados em um link de dados. Existem três categorias de LSRs em uma rede MPLS:

LSR de Ingresso: recebe os pacotes que ainda não possuem Label, em seguida insere o Label no início do pacote o envia para o link de dados correto.

LSR de Saída: recebe os pacotes com Label, remove o Label e os envia a um link de dados.

LSR Intermediário: recebe um pacote com Label, efetua uma operação sobre ele, altera o pacote e o envia para o link de dados correto.

2.2.6 Label Switch Path - LSP

Um Label Switch path consiste no caminho que os pacotes irão percorrer dentro de uma rede MPLS, a Figura 8 descreve um LSP. No momento que um pacote ingressa numa rede MPLS ele é classificado e vinculado a uma determinada classe de equivalência (FEC) e com base nessa classificação os roteadores do núcleo da rede MPLS, os LSR Intermediários, irão apenas chavear os pacotes para o caminho definido pela sua classe de equivalência.

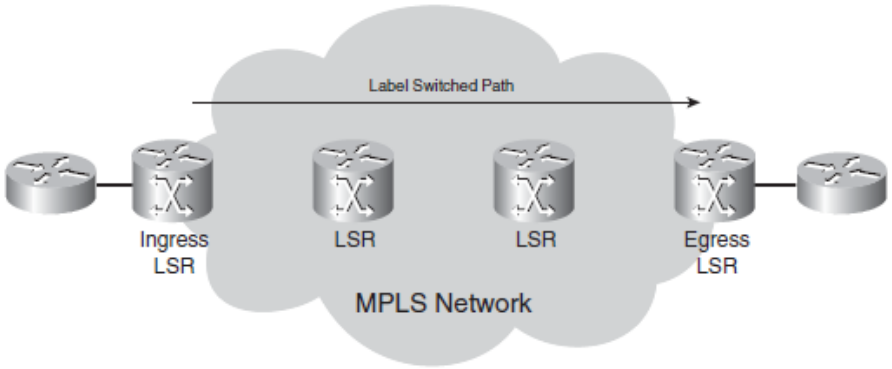


Figura 8 - Um LSP através de uma rede MPLS [GHE 2007]

2.2.7 Forwarding Equivalency Class – FEC

Uma FEC é um grupo ou fluxo de pacotes que é encaminhado através de um mesmo caminho e possui o mesmo tratamento no interior de uma rede MPLS. Todos os pacotes que pertencem a uma FEC possuem os mesmos labels.

Exemplo de FECs que podem ser definidas pelo gerente da rede:

a. Pacotes com endereço IP (camada 3) correspondendo a um determinado prefixo.
b. Pacotes Multicast pertencem a um grupo específico.
c. QoS.
d. ID do protocolo IP.

2.2.8 Label Distribution

O primeiro Label é colocado pelo LSR quando o pacote ingressa na rede MPLS e o Label pertence a um LSP. Os LSR do núcleo por sua vez recebem esses pacotes, modificam o Label de acordo com a sua FEC e o encaminham. A questão é que os LSR de ingresso precisam de um mecanismo para comunicar os LSR do núcleo sobre cada Label criado e seu significado para que essas informações sejam utilizadas na formação das tabelas de roteamento. Portanto, um protocolo de distribuição de labels é necessário.

É possível distribuir os labels de duas maneiras:

a. Utilizando um protocolo de roteamento IP existente. Dessa forma, há a vantagem de que não é preciso um novo protocolo rodando nos LSRs, mas é necessário que protocolo seja capaz de carregar os labels. A principal vantagem de ter um protocolo de roteamento transportando os labels é que a distribuição dos labels e o roteamento estão sempre em sincronia. [GHE 2007]

b. A segunda forma é rodando um protocolo para distribuição de labels. A vantagem é estar rodando um protocolo independente. O protocolo definido pelo IETF para executar esta função foi o *Label Distribution Protocol* (LDP). LDP tem quatro funções principais [GHE 2007]:

- a descoberta dos LSRs que estão executando o LDP;
- o estabelecimento e a manutenção de sessões;
- o anúncio de mapeamento de *labels*;
- a manutenção de sessões LDP por meio de notificação.

2.3 Virtual Private Network – VPN

Rede Privada Virtual ou VPN é uma rede privada, ou seja, uma rede com acesso restrito e construída sobre a infra-estrutura de uma rede pública (recursos públicos, sem controle sobre o acesso aos dados), normalmente a Internet. Em vez de se utilizar links dedicados para conectar redes remotas, utiliza-se a infra-estrutura da Internet ou segmentação de rede, uma vez que para os usuários a forma como as redes estão conectadas é transparente. [SIL 2005]

2.3.1 Características das VPNs

As VPNs apresentam uma série de características que aumentam a segurança da informação que está trafegando.

2.3.1.1 Privacidade

Como a informação que trafega através de uma VPN encontra-se em uma rede pública é importante que ela não seja capturada por usuários que não têm permissão de acesso. Porém, o objetivo é garantir que mesmo que dados privados sejam capturados indevidamente, seu conteúdo não poderá ser decodificado, ou seja, a informação capturada estará cifrada e não terá utilidade. Para isso, as VPNs utilizam mecanismos de criptografia da informação. Podem ser utilizados mecanismos de chaves públicas e privadas e algoritmos que utilizem chaves simétricas e assimétricas como DES e RSA respectivamente. [SIL 2005]

2.3.1.2 Integridade

A informação recebida deve ser a mesma que foi enviada, ou seja, o pacote original não deve ser modificado no trânsito. A integridade pode ser verificada usando funções de *hash*. *Funções de hash* são funções que recebem dados de comprimento arbitrário, comprimem estes dados e devolvem um número fixo de bits, esse conjunto de bits é denominado *hash.* Executa-se a função sobre o pacote enviado na origem e novamente

no destino, se o resultado do *hash* diferir do original então o pacote foi modificado no trânsito. [SIL 2005]

Este trabalho não irá se aprofundar nesse assunto, contudo mais informações sobre *funções de hash* podem ser encontradas no livro: STALLINGS, William; **Cryptography and Network Security Principles and Practices**, 4th Ed.

2.3.1.3 Autenticidade

Deve-se ter certeza que a mensagem foi enviada por um membro autorizado. Isso pode ser feito calculando o hash da mensagem original e cifrando com a chave privada do autor. O resultado das duas operações é chamado de assinatura digital e essa é adicionada ao final do pacote. Quando o destinatário recebe a mensagem, ele usa a chave pública do par de origem para decifrar a assinatura digital. Dessa forma, é garantida a autenticidade da mensagem. [SIL 2005]

2.3.1.4 Não-repúdio

É a garantia que a mensagem recebida não será negada no futuro, ou seja, há garantia da fonte da mensagem. O não-repúdio é uma etapa posterior à autenticidade (é um atributo opcional da autenticidade). [SIL 2005]

2.3.2 Tipos de VPNs – MPLS x Frame-Relay

As empresas que administram a infra-estrutura da Internet, backbones e provedores de acesso, podem oferecer redes privadas por segmentação de rede, como é o caso do MPLS. A tabela abaixo mostra algumas características de cada modalidade:

Tabela 1 - Tipos de VPNs [SIL 2005]

Características	*Frame-Relay*	*MPLS*
Isolamento de tráfego (VPN)	Sim	Sim
Acesso Discado	Sim	Sim
Banda Assimétrica	Sim	Não
Voz c/QoS IP	Não	Sim
Gerência Pró-ativa	Sim	Sim
Endereçamento Privado	Sim	Sim
Conexão Internacional	Sim	Não
Utilização de VC/Tunel	Sim	Não
Criptografia de dados	Não	Não
Utilização de Labels	Não	Sim

Analisando a tabela é possível observar que criptografia não está prevista para o MPLS e nem para Frame-Relay, porque o tráfego dos dados é controlado pelos roteadores e é garantida a privacidade pelo isolamento da comunicação. [SIL 2005]

2.3.3 MPLS VPN

Cada vez mais empresas e pessoas precisam que seus serviços de rede estejam disponíveis em todos os lugares em que forem necessários. Isso quer dizer que a expansão territorial das empresas, como por exemplo, implantação de filiais, leva a necessidade de conectá-las. Entretanto, o custo de construir uma infra-estrutura particular é extremamente elevado e de difícil gerenciamento. Por isso, as empresas optam por utilizar a infra-estrutura de uma rede pública, como a internet, e a melhor opção nesse caso é utilizar VPNs.

Contudo, com o advento do MPLS surgiram as MPLS VPNs que trazem uma gama de vantagens em relação às outras implementações de VPNs. Entre as principais vantagens está a mais fácil implantação de serviços a seus clientes, o mais fácil gerenciamento das VPNs, e permite um fluxo ideal do tráfego dentro do backbone MPLS.

O MPLS permite a criação de VPN porque garante um isolamento completo do tráfego com a criação de tabelas de Labels usadas para roteamento exclusivas de cada circuito virtual. [GHE 2007] [BOA 2004]

A RFC-2547bis define como os provedores de serviços podem usar seus backbones para prover serviços de VPN para seus usuários. A RFC-2547bis também é chamada de VPN BGP-MPLS porque o BGP é o protocolo utilizado para distribuir as informações de roteamento das VPNs e pelo uso do MPLS para estabelecimento dos circuitos virtuais e encaminhamento do tráfego. [BOA 2004]

Uma VPN MPLS é implementada sobre duas redes: a rede do provedor e a rede do cliente. A rede do provedor é constituída de Provider Edge Router (PE) que provêem serviços de VPN e conectividade para as redes dos clientes. As redes dos clientes são normalmente constituídas, fisicamente, por diferentes pontos de acessos. Os roteadores dos clientes que se conectam aos provedores dos serviços das VPNs são chamados de Customer Edge Router (CE), a Figura 9 mostra os componentes das VPN MPLS. [BOA 2004]

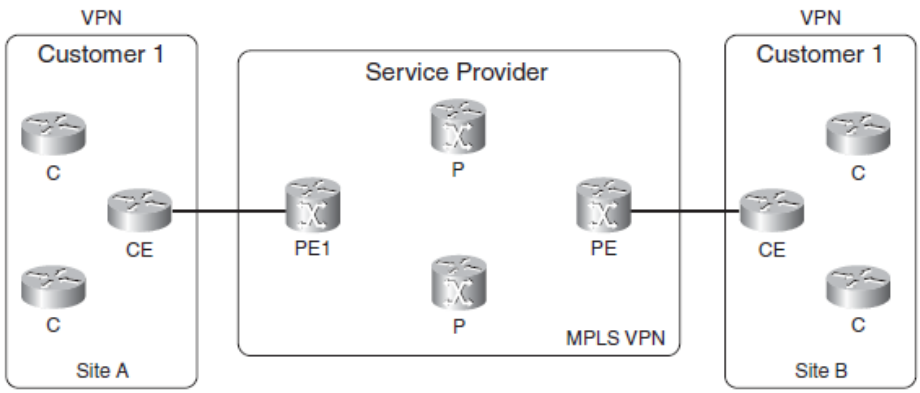


Figura 9 - Componentes da VPN MPLS [4]

A conexão entre os roteadores CE e PE pode ser uma conexão remota através de Frame-Relay ou ATM ou, ainda, um enlace Ethernet. As redes dos clientes trocam rotas com provedores de serviços (CE para PE), usando rotas estáticas ou via RIP, OSPF ou E-BGP. [BOA 2004]

No momento que um roteador PE recebe a rota atualizada é criada uma tabela de roteamento e as informações de alcançabilidade são enviadas para cada nó da VPN conectada ao roteador. [BOA 2004]

Os roteadores PEs estabelecem sessões iBGP (PE a PE) para trocar informações de rotas dos clientes. O tráfego da rede passa através do LSP (Label Switched Path) pré-estabelecido através dos protocolos de sinalização LDP. O roteador PE adiciona dois Labels como prefixos, o mais externo identifica o próximo salto a ser realizado e o mais interno identifica a VPN do cliente. [BOA 2004]

2.3.3.1 Implementação

Na arquitetura tradicional de roteamento IP há uma clara distinção entre rotas externas e internas. Na visão de um provedor de serviço, rotas internas incluem todos os enlaces internos do provedor e interfaces de loopback. Essas rotas internas são trocadas com outras rotas na rede, por meio do protocolo IGP, tais como OSPF ou IS-IS. Todas as rotas aprendidas na Internet por meio de pontos de troca de tráfego (peering) ou de pontos de clientes são classificadas como rotas externas e são distribuídas por meio do protocolo externo (EGP), tais como BGP. Na arquitetura IP tradicional, o número de rotas internas é bem menor que o número de rotas externas [GHE 2007] [BOA 2004].

Customer Edge Device (CE)

Provê acesso do cliente ao provedor de serviços de rede. Normalmente, é um roteador IP que estabelece uma conexão direta com o roteador PE e em seguida anuncia as rotas da VPN local ao PE e aprende as rotas remotas da VPN. [GHE 2007]

Provider Edge Routers (PE)

Os PEs trocam informações de roteamento com os CEs através de roteamento estático RIP, OSPF ou eBGP. Após aprender as rotas das VPNs locais com os roteadores CEs ele troca informações de roteamento com os outros PEs conectados através do BGP. [GHE 2007]

Finalmente, quando se utiliza o MPLS para o encaminhamento do tráfego de dados das VPNs por meio do backbone, os PEs de ingresso e de egresso funcionam com LSRs de ingresso e egresso respectivamente. A Figura 10 apresenta o conjunto de protocolos normalmente disponível nos roteadores PEs para atender as necessidades dos provedores de serviços.

Na Figura 10, múltiplos protocolos de acesso são suportados para permitir ao provedor do serviço flexibilidade em oferecer aos seus clientes vários métodos de tecnologia de acesso.

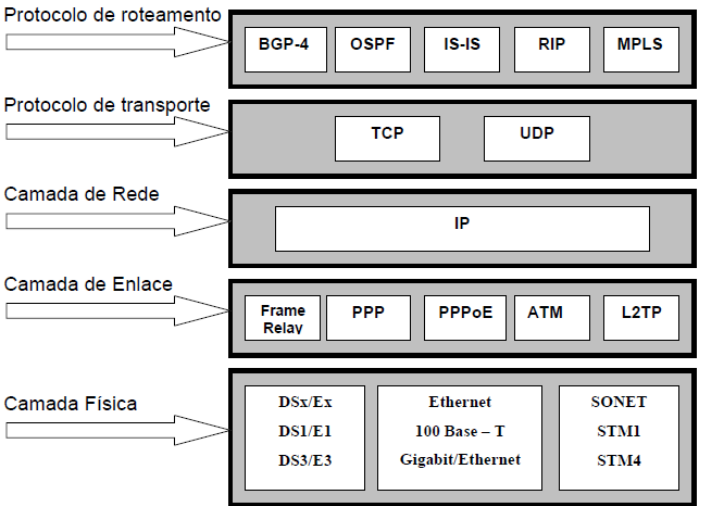


Figura 10 - Protocolos suportados pelos PEs [BOA 2004]

Provider Routers (P)

É um roteador na rede que não troca informações diretamente com os CEs. A função destes roteadores é fazer o encaminhamento do tráfego de dados através do backbone MPLS.

Tabela de Roteamento e encaminhamento da VPN

Tabela de roteamento e encaminhamento da VPN é chamada de VPN Routing and Forwarding (VRF). Esta tabela é um elemento chave na arquitetura de VPNs MPLS. Ela é constituída de uma tabela de encaminhamento e roteamento para cada VPN dentro dos roteadores PEs. Uma VRF privada é acessível somente pelas interfaces da VPN correspondente. Todos os pontos conectados no roteador PE devem fazer parte de uma VRF. Todas as informações das VPNs são refletidas na VRF e os pacotes que viajam através daquele ponto serão roteados e encaminhados com base unicamente na informação encontrada na VRF correspondente.

2.4 Métricas de Desempenho de Redes de Computadores

O objetivo principal para a criação de métricas é prover uma base para a avaliação do desempenho dos diferentes componentes da rede. Estas avaliações podem servir como padrões e são importantes já que permitem suprir algumas necessidades, como:

a. Auxiliar na detecção e resolução de problemas;

b. Permitir o planejamento e previsões de crescimento das redes;

c. Possibilitar aos pesquisadores um melhor entendimento do comportamento das redes e acompanhamento da sua evolução.

Este trabalho enfocará os testes de benchmark nos itens citados abaixo:

2.4.1 Taxa de transferência (Vazão)

Por definição a vazão é a taxa (requisição por unidade de tempo) que um determinado sistema pode prover. Em redes de computadores a vazão é medida em pacotes por segundo (pps) ou bit por segundo (bps). [JAI 1991]

A vazão de um sistema, inicialmente, aumenta à medida que se aumenta a carga do sistema. Após certa carga, a vazão para de aumentar, na maioria dos casos ela começa a diminuir. Tal comportamento pode ser visto na Figura 11. A máxima vazão atingível sobre uma condição de carga ideal é chamada de **capacidade nominal** do sistema, no caso de redes de computadores é chamada de **largura de banda**. Em alguns casos há interesse em descobrir a vazão máxima atingível sem que se exceda um limite de tempo de resposta, isso é chamado de **capacidade utilizável** do sistema. Em muitas aplicações, o Knee da vazão ou curva de **tempo de resposta** é considerado o ponto ótimo de operação do sistema. Como é mostrado na Figura 11, esse é o ponto onde tempo de resposta começa a aumentar rapidamente como função da carga mas o ganho em vazão é pequeno. Antes do ponto de Knee o tempo de resposta não tem aumentos significativos, mas a vazão cresce em função do aumento da carga do sistema. A vazão no ponto de Knee é chamada de **capacidade Knee** do sistema. [JAI 1991]

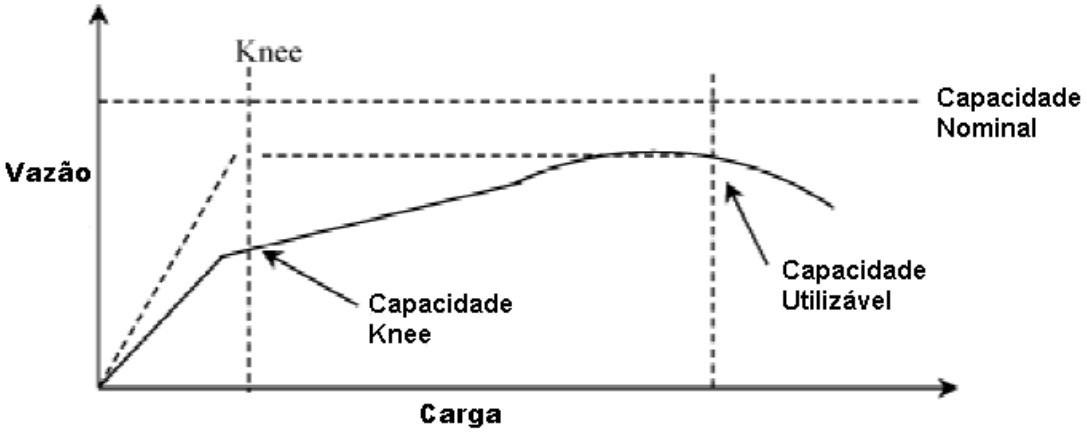


Figura 11 - Comportamento da Vazão X Carga do Sistema [JAI 1991]

No contexto deste trabalho, a vazão é taxa máxima de dados que pode ser transferida desde a origem até o destino na rede. Essa taxa é, geralmente, associada ao conceito de velocidade da rede. A análise da vazão auxilia na localização de possíveis gargalos da rede.

Outro fator relacionado à vazão é a taxa real que pode ser alcançada. Em uma rede Ethernet com velocidade de 100 Mbps, por exemplo, a velocidade de transferência máxima nunca alcançará 100 Mbps devido aos campos que são adicionados aos pacotes (preâmbulos e cabeçalhos) os quais são utilizados no roteamento e transporte dos pacotes.

A taxa de bits por segundo pode ser expressa da seguinte forma (em Mbps):

$$taxa\_bits = (frame\_len / (preamb + frame\_len + igp)) * 100$$

A vazão é um das principais métricas quando se avalia uma rede e ela permite avaliar a qualidade da rede. Na prática, as aplicações geram tráfego na rede e a vazão é uma medida de intensidade desse tráfego.

Reportando os resultados: Os resultados do teste da vazão deverão ser reportados na forma de um gráfico. No gráfico, o eixo de coordenadas x representará o tamanho do quadro, o eixo de coordenadas y será a taxa de quadros. O gráfico deverá conter também uma linha com a taxa máxima teórica para os diversos tamanhos de quadros. A segunda linha deverá mostrar os resultados dos testes. O resultado será mostrado em quadros por segundo. [BRA 1999]

2.4.2 Latência

Latência é o tempo total gasto pelo pacote desde a origem até o destino na rede. Esse tempo absoluto representa todos os atrasos desde tempos de processamento nos nodos da rede até atrasos de propagação através do meio de comunicação.

Para medir a latência, é enviado um pacote de teste com um *timestamp* e essa marca de tempo é analisada no recebimento do pacote. Para isso o pacote precisa retornar ao testador original (atraso de ida e volta).

A Figura 12 mostra o envio um pacote ICMP Request/Reply, que pode ser usado para caracterizar a latência entre o Emissor e o Receptor do pacote.

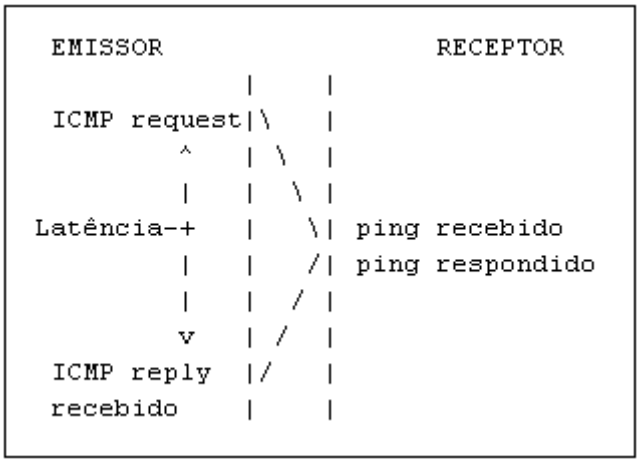


Figura 12 - Representação da Latência

Reportando os resultados: O resultado dos testes deverá ser reportado na forma de uma tabela com uma linha para cada tamanho de quadro. A tabela deverá conter colunas para: tamanho do quadro, taxa de transferência testada e latência para cada tipo de fluxo testado. [BRA 1999]

2.4.3 Variação da Latência (Jitter)

A variação no tempo da chegada dos pacotes do endereço origem é chamada de Jitter. Como os pacotes podem trafegar por redes diferentes, se os pacotes são enviados na origem a cada 10 ns eles podem não chegar ao destino com esse intervalo de 10 ns. Esse comportamento afeta a qualidade do serviço, isso pode ser crítico em tráfego de voz.

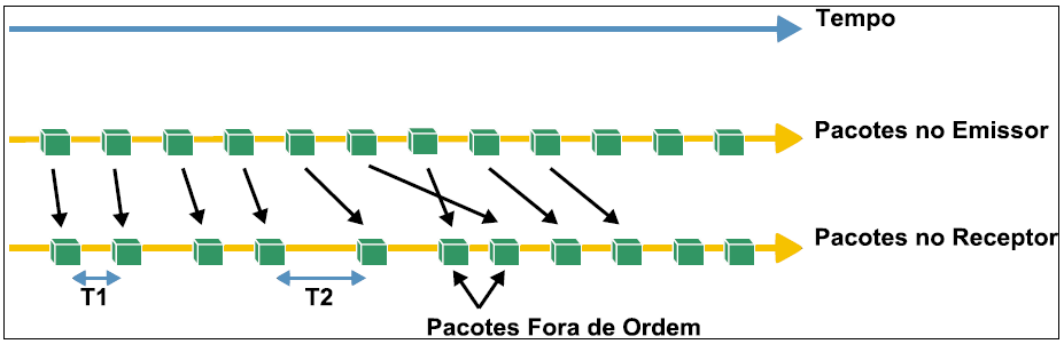


Figura 13 - Efeito do Jitter para aplicações

2.4.4 Perda de Quadros

Para medir a perda de quadros deve-se contar o número de quadros enviados na origem e que não foram recebidos no destino. Basicamente é uma taxa que representa a porcentagem de quadros perdidos em trânsito. Os motivos que levam a perda de quadros podem ser vários, como por exemplo, erros ou atraso excessivo.

O procedimento de cálculo da porcentagem de quadros perdidos é igual a:

((nr_quadros_enviados - nr_quadros_recebidos) * 100) / nr_quadros_enviados

O primeiro teste deve ser executado para a taxa de quadros que corresponde a 100% da taxa máxima para o tamanho do quadro na entrada do meio de comunicação. Este procedimento é repetido para a taxa que corresponde a 90% do que foi usado, e depois 80%. Essa sequência deve continuar (em intervalos de 10%) até existir dois testes sucessivos em que não existam perdas de quadros.

Reportando os resultados: O resultado do teste de perda de pacotes deverá ser reportado na forma de um gráfico. No gráfico, o eixo de coordenadas x representará a taxa de quadros na entrada como percentual do valor máximo teórico para o determinado tamanho de quadro, o eixo de coordenadas y representará o percentual de perdas de pacotes para uma determinada taxa de entrada de quadros. [BRA 1999]

2.4.5 Atraso excessivo

O atraso entre nodos da rede pode variar devido a características da infra-estrutura da rede e também pelo número de nodos. Normalmente utiliza-se um valor de tempo limite de espera (*timeout*), 2 segundos, por exemplo. Esse limite de espera é importante, pois, o receptor estará esperando os pacotes e ele precisa saber quando parar de esperar e considerar o pacote como perdido.

2.4.6 Recuperação do Sistema

O objetivo dessa métrica é caracterizar o tempo que o sistema leva para se recuperar de um cenário de sobrecarga.

O procedimento para efetuar essa medida consiste em enviar um fluxo de quadros a uma taxa de 110% da taxa máxima de transferência registrada para o meio de comunicação por pelo menos 60 segundos. Em um *timestamp* A se reduz a taxa de

quadros em 50% e grava-se o tempo do último quadro perdido (*timestamp* B). O tempo de recuperação do sistema é determinado pela subtração do *timestamp* B pelo *timestamp* A. Esse processo deve ser repetido várias vezes e a média dos tempos obtidos deve ser calculada. [BRA 1999]

2.4.7 Reinicialização

O objetivo dessa métrica é caracterizar o tempo que o sistema leva para se recuperar de uma reinicialização do dispositivo ou do software.

O procedimento para efetuar essa medida consiste em enviar um fluxo contínuo de quadros a uma determinada taxa de transferência e com quadros de tamanho mínimo. Em seguida, induzir uma reinicialização no DUT. Então, monitora-se a saída antes de os quadros começarem a ser encaminhados e grava-se o tempo do último quadro (timestamp A) do fluxo inicial enviado e o primeiro quadro do novo fluxo (timestamp B) que foi recebido. Um teste de interrupção no fornecimento de energia poderia ser feito para simular o comportamento acima. [BRA 1999]

O tempo de reinicialização é obtido subtraindo-se o timestamp B pelo timestamp A.

3 DEFINIÇÕES DO EXPERIMENTO

O projeto, a análise e a implementação de redes são tarefas complexas e trabalhosas. De forma geral, existem três soluções que podem auxiliar na execução dessas tarefas: a experimentação com redes reais, a utilização de métodos analíticos e a simulação computacional. Cada uma das soluções possui vantagens e limitações, e embora não sejam mutuamente exclusivas, cada uma delas é aplicável a uma determinada situação particular.

Neste trabalho, escolhemos a utilização de experimentação com redes reais com o auxilio de algumas ferramentas opensource.

O experimento consiste em avaliar o desempenho fim-a-fim de uma rede MPLS com uma topologia básica sobre a qual configura-se as tabelas de encaminhamento para simular VPNs. Para a construção da rede MPLS serão utilizados PCs "rodando" um kernel Linux modificado com a inclusão do modulo MPLS-LINUX.

3.1 Topologia para o experimento

O objetivo do experimento é avaliar o desempenho de VPNs Layer 3 MPLS. Um exemplo de topologia onde duas VPNs comutam pacotes é mostrado na Figura 14, a rede MPLS em questão é constituída por 5 roteadores MPLS e 4 PCs, sendo que dois são bordas de entrada na VPN 1 e dois são bordas de entrada na VPN 2.

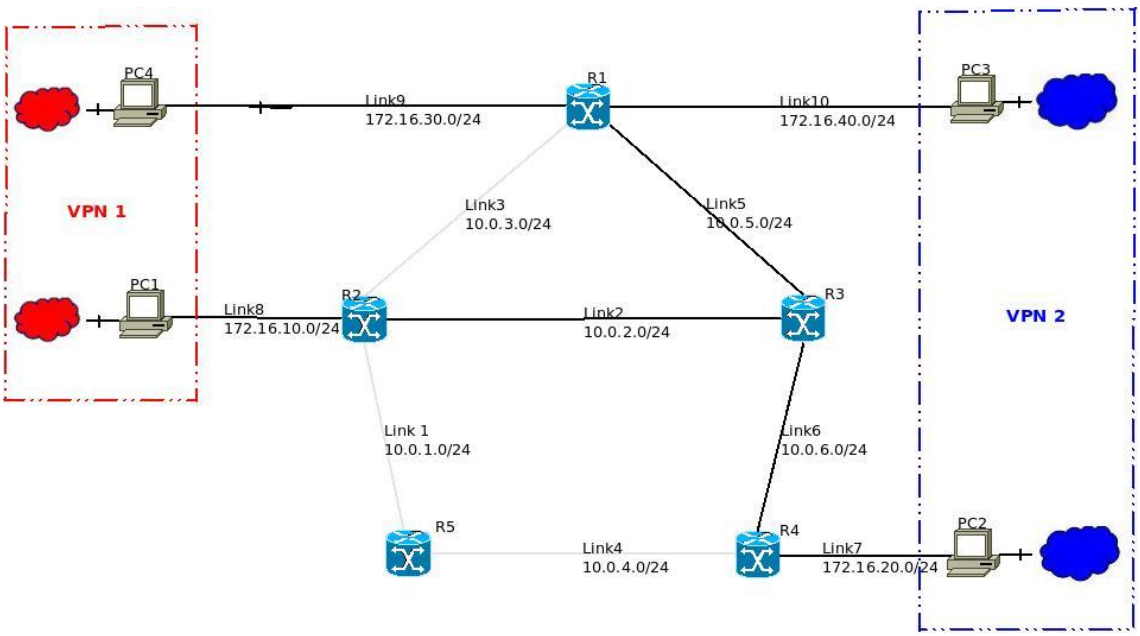


Figura 14 - Topologia de VPNs L3 MPLS

Para efeito de avaliação será utilizada uma topologia mais simples, pois o que nos interessa é avaliar as operações de push, pop e swap dos rótulos. A Figura 15 mostra a topologia que será utilizada para as medições durante os experimentos.

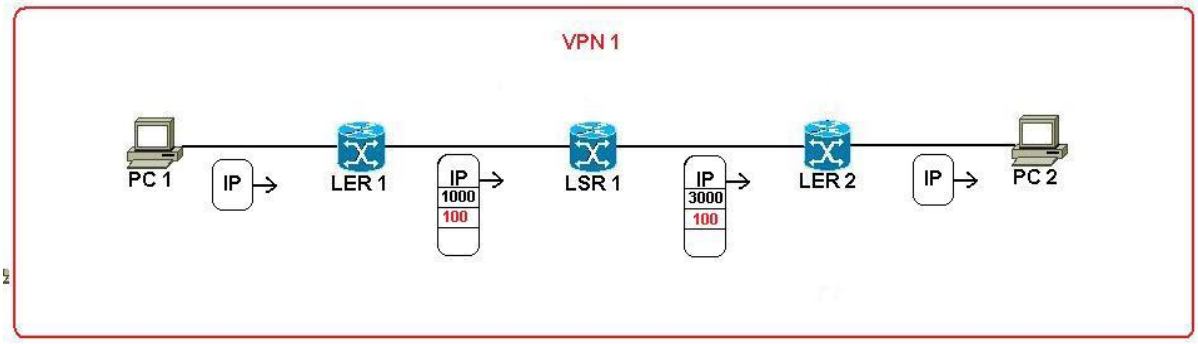


Figura 15 - Topologia para experimento

3.2 Configuração do cenário

A configuração do cenário consiste na configuração das máquinas envolvidas no experimento. O tráfego será simulado pela ferramenta RUDE e capturado pela CRUDE, seguindo as recomendações da RFC 2544 para testes de benchmark sobre dispositivos de interconexão de redes de computadores. Há um detalhamento do funcionamento do RUDE & CRUDE no Anexo B.

As configurações consistem na: instalação do módulo MPLS, instalação do IP ROUTE, instalação do RUDE, definição de endereços IP nas máquinas, configuração das interfaces de rede, criação da tabelas de encaminhamento (uso do módulo MPLS e do IP ROUTE). A instalação do módulo MPLS só é necessária nas máquinas que simularão roteadores MPLS. A instalação do RUDE só é necessária nas máquinas emissoras e receptoras de tráfego, no caso deste experimento, Host 1 e Host 2.

A Figura 16, abaixo, mostra o cenário de configuração da rede sobre a topologia definida:

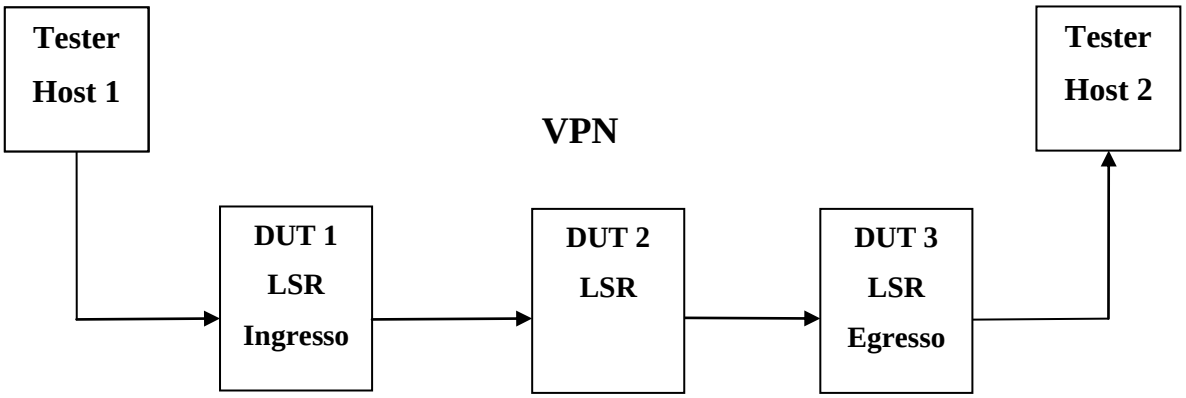


Figura 16 - Topologia para benchmark

* DUT: PCs rodando Linux kernel 2.6.32.11 (kernel MPLS)

* TESTER: PCs rodando Linux (qualquer kernel)

* A rede física de interconexão será Ethernet 100 Mbps

A forma escolhida para a realização dos testes consiste em usar um gerador de tráfego na porta de entrada do DUT (Device Under Test) e um analisador do tráfego na porta de saída do DUT (Figura 16).

3.3 MPLS-Linux

MPLS para Linux é uma iniciativa de código aberto para criar um conjunto de protocolos de sinalização MPLS e um plano de encaminhamento MPLS para o sistema operacional Linux. [MPL 2010]

Neste trabalho o MPLS-Linux foi usando para "transformar" PCs com sistema operacional Linux em roteadores MPLS, isto é, o módulo MPLS foi instalado nesses PCs para que eles fizessem o papel de roteadores na topologia de teste.

3.3.1 Comandos do MPLS-Linux

Este capítulo faz uma breve descrição sobre alguns dos comandos disponíveis no módulo MPLS-Linux. Os comandos abordados aqui serão suficientes para o entendimento do experimento proposto por esse trabalho, bem como os Scripts de configuração da tabelas de encaminhamento, presentes no ANEXO C.

3.3.1.1 Roteador de Ingresso na rede MPLS - LER

O comando abaixo cria uma entrada na tabela NHLFE e adiciona o rótulo '1000', e passa a encaminhar os pacotes para o endereço IP 10.0.0.3 usando a interface de saída eth1.

```
#mpls nhlfe add key 0 instructions push gen 1000 nexthop eth1 ipv4 10.0.0.3

  |    |     |   |        |          |    |   |      |      |    |      |
  |    |     |   |        |          |    |   |      |      |    |      |
  |    |     |   |        |          |    |   |      |      |    |   Endereço IP
  |    |     |   |        |          |    |   |      |      |    |   do próximo hop
  |    |     |   |        |          |    |   |      |      |    |
  |    |     |   |        |          |    |   |      |      |   Protocolo da Camada 3
  |    |     |   |        |          |    |   |      |      |
  |    |     |   |        |          |    |   |      |     Interface de saída
  |    |     |   |        |          |    |   |      |
  |    |     |   |        |          |    |   |    Caminho do próximo hop
  |    |     |   |        |          |    |   |
  |    |     |   |        |          |    |  Valor do label
  |    |     |   |        |          |    |
  |    |     |   |        |          |   Uso de encapsulamento Ethernet
  |    |     |   |        |          |
  |    |     |   |        |       Add um novo label na pilha de labels
  |    |     |   |        |
  |    |     |   |       Intruções da NHLFE
  |    |     |   |
  |    |     |  Uma nova chave(cujo valor será retornado se o comando tiver sucesso)
  |    |     |
  |    |    Adiciona
  |    |
  |    Cria uma nova entrada na tabela NHLFE
  Comando MPLS
```

Figura 17 - Comando MPLS add key

O comando abaixo mapeia uma FEC para a entrada na NHLFE que foi criada acima, para que ela seja realmente usada.

```
# ip route add 192.168.1.0/24 via 10.0.0.3 mpls 0x2

  |    |    |       |          |     |      |   |
  |    |    |       |          |     |      |   a chave que identifica a usada na NHLFE
  |    |    |       |          |     |      |   (deve ser criada anteriormente)
  |    |    |       |          |     |      |
  |    |    |       |          |     |      vincula a FEC (Endereço IP Destino)
  |    |    |       |          |     |      para este NHLFE
  |    |    |       |          |     |
  |    |    |       |          |     Endereço do próximo hop
  |    |    |       |          |
  |    |    |       |          via
  |    |    |       |
  |    |    |       Endereço da rede destino
  |    |    |
  |    |    Adiciona
  |    |
  |    Modifica a tabela de roteamento
  |
  Comando IP
```

Figura 18 - Comando IP Route

3.3.1.2 Roteador de trânsito na rede MPLS - LSR

O comando abaixo define um *labelspace*, assim o rotador irá esperar pacotes MPLS a partir de uma determinada interface, neste caso, seria eth0.

```
# mpls labelspace set dev eth0 labelspace 0

   |       |       |   |   |        |     |
   |       |       |   |   |        |     Valor do labelspace
   |       |       |   |   |        |
   |       |       |   |   |        keyword
   |       |       |   |   |
   |       |       |   |   Interface de entrada dos pacotes MPLS
   |       |       |   |
   |       |       |   keyword
   |       |       |
   |       |       Seta interfaces para um específico labelspace
   |       |
   |       keyword
   |
   Comando MPLS
```

Figura 19 - Comando MPLS add labelspace

O comando seguinte complementa o funcionamento do anterior, adicionando uma entrada na tabela ILM, para especificar qual é o rótulo esperado, neste caso, ‘1000’.

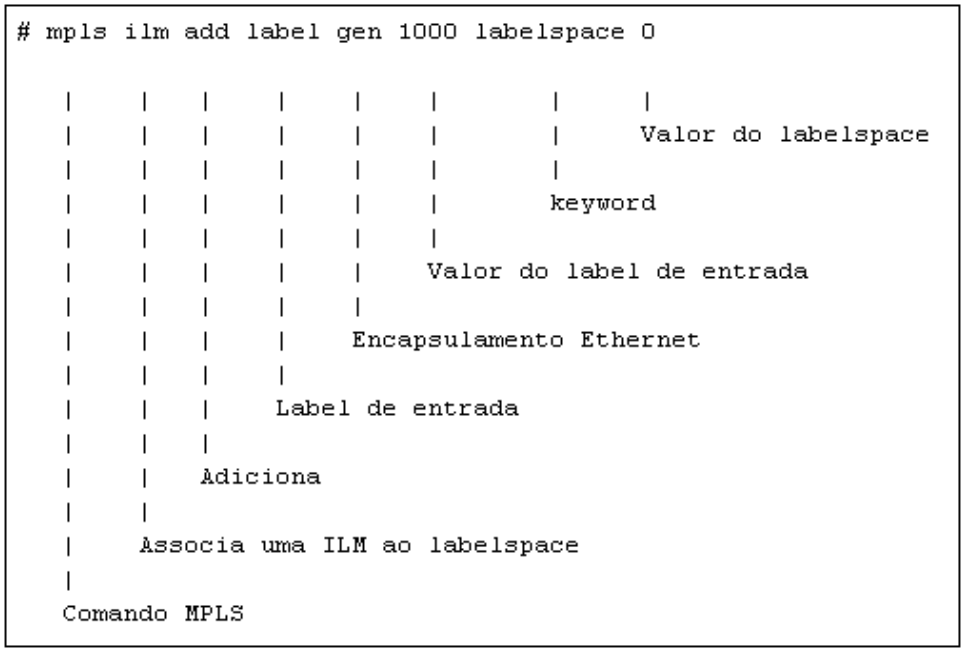


Figura 20 - Comando MPLS add ILM

O comando abaixo faz a troca (switching) do label, para em seguida encaminhar o pacote para o seu próximo hop.

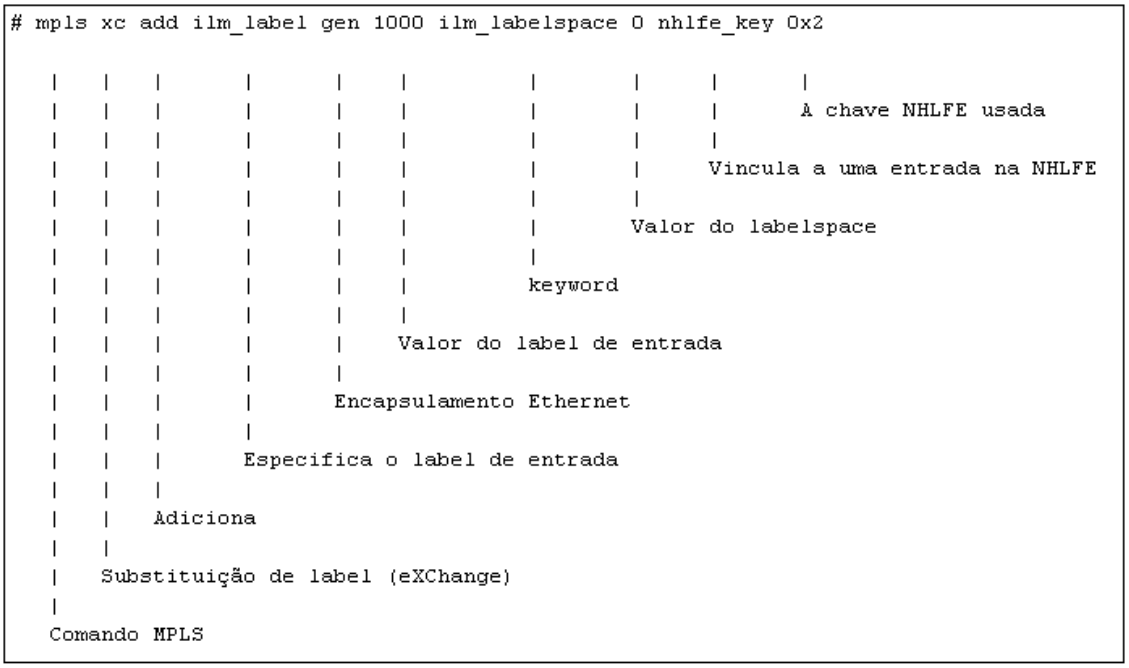


Figura 21 - Comando MPLS xc (eXChange)

3.3.1.3 Roteador de Egresso da rede MPLS

Neste caso, basta apenas definir a interface que estará recebendo os pacotes MPLS e em seguida remover o label especificado pela tabela ILM.

```
# mpls labelspace set dev eth2 labelspace 0

# mpls ilm add label gen 2000 labelspace 0
```

Figura 22 - Removendo Labels MPLS

3.3.1.4 Distribuição dos Labels

Como pode ser visto nos comandos acima, a atribuição de labels neste experimento se dará de forma estática, ou seja, não será utilizado nenhum outro protocolo para essa tarefa como o LDP e o BGP. Essa escolha foi tomada pela impossibilidade da utilização de ambos, visto que não estão implementados no MPLS-Linux.

As tabelas ILM e NHLFE para o experimento estarão configuradas como é mostrado nas tabelas abaixo:

Tabela 2 - Encaminhamento do LER 1 - Experimento

FEC	***ILM***		***NHLFE***		
Classify by	**In Label**	**In Port**	**Out Label**	**Out Port**	**Next Hop**
172.16.30.0/24	n/a	n/a	1000 + 100	eth1	LSR (10.0.2.3)
n/a	4001+100	eth1	0	eth0	Host1 (172.16.10.10)

Tabela 3 - Encaminhamento para LSR – Experimento

FEC	***ILM***		***NHLFE***		
Classify by	**In Label**	**In Port**	**Out Label**	**Out Port**	**Next Hop**
Label	1000+100	eth0	3000+100	eth2	LER2 (10.0.5.1)
Label	4000+100	eth2	4001+100	eth0	LER1(10.0.2.2)

Tabela 4 - Encaminhamento do LER 2 – Experimento

FEC	***ILM***		***NHLFE***		
Classify by	**In Label**	**In Port**	**Out Label**	**Out Port**	**Next Hop**
171.16.10.0/24	n/a	n/a	4000+100	eth3	LSR (10.0.2.3)
n/a	3000+100	eth3	n/a	eth0	Host2 (172.16.30.30)

3.4 Formato do Quadro para o experimento

A topologia de teste será construída sobre uma rede Ethernet e serão transportados pacotes IP como mostrados na Figura 23. No momento da entrada e saída dos pacotes IP na rede MPLS, os roteadores LSR irão efetuar a inclusão de dois rótulos, um para indicar a qual VPN o pacote pertence e outro para o encaminhamento do pacote no LSP.

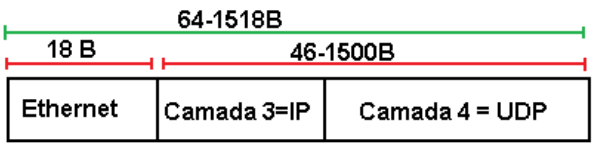


Figura 23 - Quadro Ethernet IP

No escopo desse experimento os pacotes relevantes terão o formato mostrado na Figura 24.

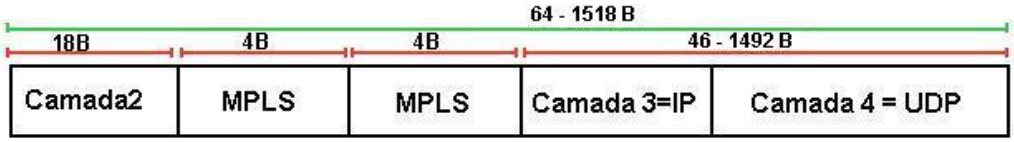


Figura 24 - Quadro Ethernet IP Rotulado

3.5 Tamanho dos Quadros para o experimento

Os testes deverão ser executados com um número de tamanhos de quadros. Devem ser incluídos, especialmente, os tamanhos máximos e mínimos do protocolo sobre teste e de acordo com a rede de interconexão. No caso deste experimento será uma rede Ethernet com pacotes IP. Os tamanhos, em bytes, de quadros recomendados pela RFC 2544 são:

64, 128, 256, 512, 1024, 1518

Os tamanhos incluem o mínimo e máximo do protocolo Ethernet padrão, e a seleção entre esses extremos irá refletir nas maiores e menores taxas de transferências.

3.6 Carga de trabalho

O dispositivo será submetido a diversos tráfegos de pacotes em testes isolados. Cada teste irá utilizar um dos tamanhos de quadro definido, e terá uma duração de pelo menos 60 segundos. Com objetivo de alcançar as taxas máximas de utilização do dispositivo poderá ser utilizado mais de fluxo de dados, durante casa teste. Será utilizado como referência para as taxas máximas de pacotes por segundo os valores mostrados na Tabela 5, esses valores são fornecidos pela RFC 2544.

Tabela 5 - Taxa de quadros por segundo (teórico)

Fluxo	***Tamanho (bytes)***	***Ethernet 100 Mbps Teórico (pps)***
1	64	148809
2	128	84459
3	256	45286
4	512	23496
5	768	15862
6	1024	11973
7	1280	9615
8	1518	8127

Como os valores mostrados na Tabela 5 se referem a um máximo teórico, eles foram utilizados como referência para medição do valor máximo real da taxa de pacotes na

rede utilizada no experimento. Com auxílio do RUDE e do CRUDE (Ver scripts no Anexo III) foram definidos os fluxos [1-8] do Host 1 para o Host2, Figuras 12 e 13. Desta forma foi obtido o valor real para cada taxa de pacotes por segundo que podem trafegar para cada tamanho de quadro na rede que está sendo avaliada.

Verificação dos quadros recebidos

Para efeito de avaliação, todos os quadros que não pertençam aos fluxos do experimento deverão ser desconsiderados. O equipamento sob teste deverá descartar os quadros que não sejam de encaminhamento MPLS.

A coleta dos resultados será feita através da análise sistemática dos arquivos de log capturados pelo CRUDE. Estes arquivos contêm um identificador de número de sequência para cada pacote, um timestamp, e as estatísticas de cada fluxo capturado. Nas estatísticas são mostrados os seguintes campos: o número de pacotes recebidos, o número de pacotes recebidos fora de ordem, o número de pacotes perdidos, o total de bytes recebidos (subtraindo preâmbulos e cabeçalhos) e o atraso médio. Assim, será possível identificar se o número de pacotes recebidos corresponde ao número de pacotes enviados, ou se houve perdas e qual seu montante.

Após a coleta dos dados e sua contagem, os dados devem ser consolidados e gráficos devem ser produzidos para auxiliar a compreensão dos resultados. Na seção 2.4 há recomendações de como esses gráficos dever ser construídos bem como as tabelas de resultados.

4 AVALIAÇÃO DE DESEMPENHO

A avaliação de desempenho consistiu em três fases. Sendo a primeira fase a implementação da infra-estrutura de rede baseando-se na topologia definida na seção 3.1, em seguida, na segunda fase executou-se a configuração do ambiente de teste, isto é, configuração de endereços IP, interfaces de rede, tabelas de encaminhamento dos pacotes (uso do *ip route*), e configuração das tabelas de encaminhamento MPLS. A terceira fase consistiu nas medições. As fases 2 e 3 foram feitas com auxílio de Shell Scripts, eles podem ser vistos nos ANEXOS B e C.

Após a execução da configuração de cada uma das máquinas da rede foi necessário a execução de um teste de validação do ambiente. Esse teste consistiu no exame do caminho feito por um determinado pacote, ou seja, foi disparado um comando PING no Host1 endereçado ao Host2. A ideia é verificar se o pacote percorre a rota MPLS e se os rótulos são incluídos e removidos corretamente e se não há erros durante esse percurso, isso foi feito com auxilio do aplicativo Wireshark.

Algumas constatações precisam ser citadas. Nos testes iniciais de vazão percebeu-se um desempenho extremamente baixo do MPLS em relação ao uso da rede sem o módulo MPLS e com valores de referência para redes Ethernet. Após análises dos resultados iniciais, percebeu-se:

1. Nos primeiros testes o CRUDE foi utilizado com a opção de gravação de log (opção -l) para cada pacote recebido. Ficou constatado que estes logs estavam impactando no desempenho do Host2 (receptor dos fluxos). Por isso, foi removida a geração de logs e substituída pela geração das estatísticas ao final de cada fluxo (opção -s). Após essa remoção, houve um aumento significativo no desempenho de cada um dos fluxos.
2. Tendo em vista o que foi constatado no item acima, ficou evidente que a plataforma estava impactando significativamente nos resultados do experimento. Desta forma, outra providência tomada foi a configuração de todas as máquinas do experimento para executar o Linux em modo texto, evitando, assim, o overhead gerado pelo modo gráfico do Linux.
3. Outra medida adotada foi a utilização do parâmetro de prioridade do processo, tanto para o RUDE quanto para o CRUDE (opção –P). Com esse parâmetro é possível aumentar a prioridade do processo no sistema operacional Linux. Com o objetivo de ganho de desempenho nos experimentos, ou que o impacto das limitações da plataforma sejam as menores possíveis.

4.1 Descrição do hardware e software utilizados

Na tabela abaixo é mostrado o hardware utilizado em cada máquina presente no experimento:

Tabela 6 - Descrição do Hardware utilizado

Máquina	*CPU*	*MEM*	*HD*	*Rede*
Host 1	Intel(R) Pentium(R) 4 CPU 1.80GHz	1024 MB	40 GB 7200 rpm	100 MB
Host 2	Intel(R) Pentium(R) 4 CPU 1.70GHz	512 MB	80 GB 7200 rpm	100 MB
LER 1	Intel(R) Pentium(R) 4 CPU 2.40GHz	512 MB	80 GB 7200 rpm	100 MB
LSP	Intel(R) Pentium(R) 4 CPU 1.70GHz	256 MB	40 GB 7200 rpm	100 MB
LER 2	Intel(R) Pentium(R) 4 CPU 2.40GHz	512 MB	80 GB 7200 rpm	100 MB

Na tabela abaixo é mostrado o software utilizado em cada máquina presente no experimento:

Tabela 7 - Descrição do Software utilizado

Máquina	*Kernel*	*MPLS*	*Ip Route*
Host 1	2.6.31.14-generic	-	iproute2-2.6.33
Host 2	2.6.31.14-generic	-	iproute2-2.6.33
LER 1	2.6.32.11-mpls	mpls 1.963	iproute2-2.6.33
LSP	2.6.32.11-mpls	mpls 1.963	iproute2-2.6.33
LER 2	2.6.32.11-mpls	mpls 1.963	iproute2-2.6.33

4.2 Teste de vazão através da Rede MPLS

Para medir a vazão da rede submeteu-se a rede a uma taxa de pacotes equivalente ao máximo teórico da taxa de quadros para o meio de comunicação, neste caso uma rede Ethernet 100 Mbps. Após cada teste verificou-se a ocorrência de perda de pacotes.

Na Tabela 8 e 9, cada linha da primeira coluna mostra o tamanho do quadro em bytes. As demais colunas representam a taxa em Mbps alcançada nas medições. Para cada tamanho de quadro aplicou-se um determinado percentual do valor máximo teórico do meio de comunicação. Os valores iniciaram em 100% e eram reduzidos em 10% a cada teste.

Tabela 8 - Vazão Medida Sem MPLS

Tamanho (Bytes)	*100%*	*90%*	*80%*	*70%*	*60%*	*50%*	*40%*	*30%*	*20%*	*10%*	*Teórico*
64	29,47	29,96	29,32	29,88	29,26	29,37	28,89	23,23	15,52	7,64	76,19
128	54,23	54,31	54,68	55,69	52,89	44,39	35,30	26,26	17,36	8,68	86,49
256	87,52	81,35	72,47	63,27	54,60	44,97	36,14	27,38	18,37	9,31	92,75
512	91,49	83,48	74,15	65,61	56,36	46,55	37,47	28,34	19,08	9,65	96,24
768	92,90	84,11	75,58	65,61	56,36	47,11	37,94	28,56	19,28	9,75	97,46
1024	93,58	85,32	75,57	66,15	56,76	47,39	38,19	28,73	19,40	9,81	98,08
1280	94,00	85,32	75,59	66,51	56,97	47,74	38,17	28,87	19,47	9,86	98,46
1516	94,38	85,44	76,03	66,57	56,95	47,59	38,29	28,87	19,52	9,86	98,56

Tabela 9 - Vazão Medida Com MPLS

Tamanho (Bytes)	*100%*	*90%*	*80%*	*70%*	*60%*	*50%*	*40%*	*30%*	*20%*	*10%*	*Teórico*
64	26,21	26,40	26,38	26,49	26,21	26,18	26,37	23,17	15,52	7,64	76,19
128	50,61	51,62	51,48	51,63	52,16	44,40	35,31	26,26	17,36	8,68	86,49
256	86,65	81,99	73,05	63,78	55,06	45,21	36,31	27,51	18,45	9,31	92,75
512	91,88	83,78	74,38	65,65	56,42	46,64	37,57	28,41	19,11	9,64	96,24
768	93,18	84,43	75,84	65,82	56,54	47,24	38,03	28,62	19,28	9,75	97,46
1024	93,98	85,62	75,83	66,37	56,94	47,52	38,26	28,79	19,42	9,81	98,08
1280	94,37	85,62	75,85	66,72	57,15	47,88	38,26	28,93	19,49	9,86	98,46
1516	95,80	86,61	77,11	67,58	57,86	48,35	38,89	29,32	19,80	9,99	98,56

A Figura 25 apresenta de forma gráfica os resultados apresentados na Tabela 9.

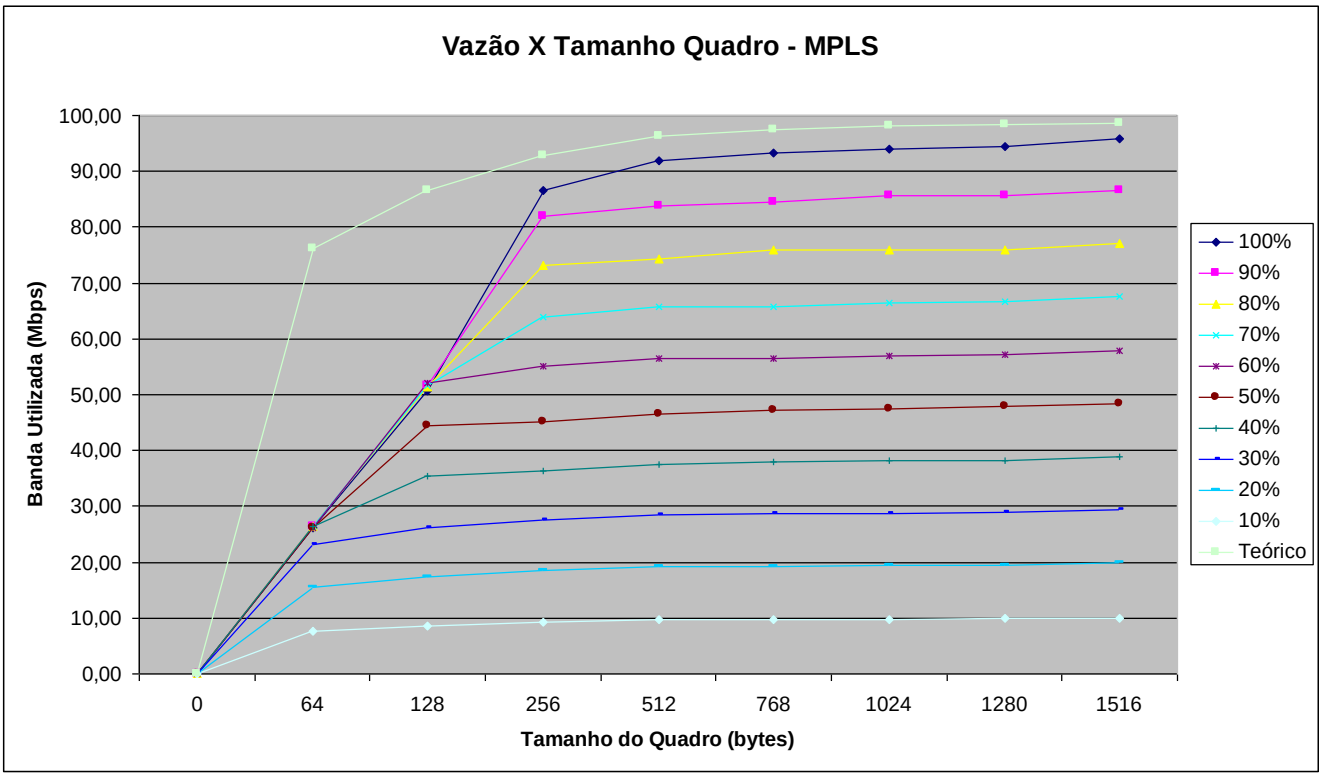


Figura 25 - Gráfico Vazão da rede MPLS

Analisando o gráfico da vazão, fica evidente que os fluxos que utilizaram pacotes com tamanhos menores (64 e 128 bytes) apresentaram um desempenho inferior em relação aos demais. É possível observar que a carga ideal de operação da rede fica em torno de 30% do valor teórico, à medida que este percentual cresce o desempenho afasta-se do teórico.

Observando-se o fluxo com a carga máxima (100%) é possível observar que com pacotes maiores que 256 bytes a vazão se aproxima da curva que representa o valor máximo teórico. Por outro lado, os fluxos com tamanhos de pacotes de 64 e 128 bytes tiveram um desempenho, respectivamente, de 34% e 55% do valor teórico referência.

4.3 Teste da Latência na Rede MPLS

A metodologia para a avaliação da latência da rede foi definida na seção 2.4.2 e será executada da seguinte forma: para obtenção dos valores utilizou-se do comando PING presente no sistema operacional. Cada fluxo durou pelo menos 120 segundos, ao final de cada execução os valores de rtt min, rtt máx e rtt desv padrão eram registrados. Cada teste foi repetido 100 vezes e ao final calculou-se a média para cada um dos fluxos.

Na tabela abaixo é mostrado o resultado obtido para cada um dos fluxos:

Tabela 10 - Latência para os Fluxo sem MPLS

Fluxos (Sem MPLS)	*Tamanho (bytes)*	*rtt (min)*	*rtt (média)*	*rtt (máx)*	*rtt (desv padrão)*
1	**64**	0.626	1.300	1.782	0.527
2	**128**	0.611	1.207	1.842	0.500
3	**256**	0.650	1.509	2.166	0.576
4	**512**	1.088	1.890	2.435	0.530
5	**768**	1.597	2.182	2.688	0.491
6	**1024**	1.938	2.772	3.185	0.488
7	**1280**	2.351	2.924	3.446	0.501
8	**1516**	2.611	3.150	3.862	0.507

Tabela 11 - Latência para os Fluxos com MPLS

Fluxos (Com MPLS)	*Tamanho (bytes)*	*rtt (min)*	*rtt (média)*	*rtt (máx)*	*rtt (desv padrão)*
1	**64**	0.626	1.233	1.959	0.505
2	**128**	0.661	1.315	1.993	0.507
3	**256**	0.695	1.484	2.140	0.543
4	**512**	1.231	2.073	2.488	0.492
5	**768**	1.677	2.277	2.795	0.494
6	**1024**	1.936	2.554	3.199	0.503
7	**1280**	2.391	3.005	3.725	0.503
8	**1516**	2.664	3.255	3.977	0.512

Com base nos resultados apresentados nas tabelas 10 e 11, foi produzido o gráfico mostrado na Figura 26, que apresenta uma comparação da latência da rede utilizando o módulo MPLS e não utilizando o módulo MPLS.

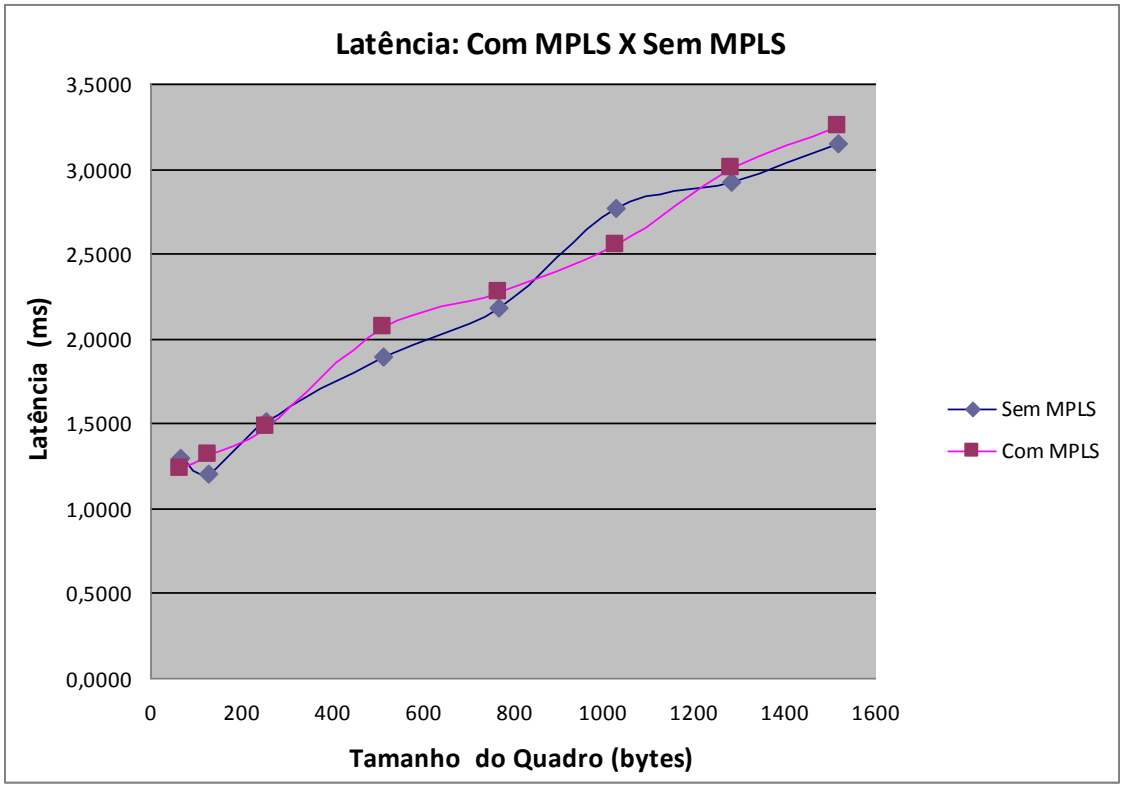


Figura 26 – Gráfico da Latência da rede Com MPLS X Sem MPLS

Observando os valores das médias da latência para os fluxos que utilizam o MPLS e para os que não utilizam, percebe-se que os valores seguem um padrão semelhante, algumas vezes alternando-se. Desta forma, chegou-se a conclusão que o impacto do uso do MPLS na latência não é significativo.

A Figura 26 representa graficamente os valores da latência da rede com o uso do MPLS e sem o uso do MPLS.

4.4 Teste de número de perda de pacotes na Rede MPLS

Para medir a perda de quadros usou-se os 8 fluxos já definidos, porém eles foram subdivididos em 10 sub-fluxos para cada fluxo original, somando um total de 80 fluxos. Esta subdivisão consiste na redução da taxa de pacotes em intervalos de 10%, ou seja, cada fluxo inicia enviando a uma taxa igual 100% do valor teórico e vai sendo reduzido até chegar a 10% do valor teórico. Os valores coletados nesse experimento podem ser vistos na tabela abaixo:

Tabela 12 - Medições dos Fluxos de Perda de Pacotes

Fluxo ID	*% Valor Teórico*	*Tamanho (bytes)*	*Pacotes Recebidos*	*Pacotes Perdidos*	*pps*
10	100%	64	511956	**81225**	51196
11	90%	64	515711	**83256**	51571
12	80%	64	515207	**86711**	51521
13	70%	64	517399	**75851**	51740

14	60%	64	511879	**103875**	51188
15	50%	64	511377	**105319**	51138
16	40%	64	515095	**78754**	51510
17	30%	64	452634	**1912**	45263
18	20%	64	303031	**0**	30303
19	10%	64	149254	**0**	14925
20	100%	128	494263	**80660**	49426
21	90%	128	504109	**69233**	50411
22	80%	128	502754	**68282**	50275
23	70%	128	504240	**69881**	50424
24	60%	128	509341	**16975**	50934
25	50%	128	433606	**1177**	43361
26	40%	128	344828	**0**	34483
27	30%	128	256411	**0**	25641
28	20%	128	169492	**0**	16949
29	10%	128	84746	**0**	8475
30	100%	256	423082	**31464**	42308
31	90%	256	400357	**16310**	40036
32	80%	256	356697	**13674**	35670
33	70%	256	311421	**11160**	31142
34	60%	256	268864	**8914**	26886
35	50%	256	220748	**6525**	22075
36	40%	256	177296	**4523**	17730
37	30%	256	134346	**2641**	13435
38	20%	256	90079	**831**	9008
39	10%	256	45455	**0**	4546
40	100%	512	224305	**13791**	22431
41	90%	512	204551	**8215**	20455
42	80%	512	181586	**7024**	18159
43	70%	512	160276	**6391**	16028
44	60%	512	137737	**5121**	13774
45	50%	512	113863	**3785**	11386
46	40%	512	91715	**2625**	9172
47	30%	512	69356	**1566**	6936
48	20%	512	46665	**505**	4667
49	10%	512	23530	**0**	2353
50	100%	768	151659	**7072**	15166
51	90%	768	137423	**5434**	13742
52	80%	768	123439	**4767**	12344
53	70%	768	107135	**3977**	10714
54	60%	768	92020	**3219**	9202
55	50%	768	76894	**2472**	7689
56	40%	768	61901	**1794**	6190
57	30%	768	46577	**1043**	4658
58	20%	768	31384	**363**	3138

59	10%	768	15874	**0**	1587
60	100%	1024	114717	**5765**	11472
61	90%	1024	104514	**4182**	10451
62	80%	1024	92563	**3591**	9256
63	70%	1024	81019	**3015**	8102
64	60%	1024	69504	**2439**	6950
65	50%	1024	58011	**1870**	5801
66	40%	1024	46706	**1371**	4671
67	30%	1024	35145	**827**	3515
68	20%	1024	23709	**272**	2371
69	10%	1024	11977	**0**	1198
70	100%	1280	92158	**3971**	9216
71	90%	1280	83618	**3339**	8362
72	80%	1280	74068	**2856**	7407
73	70%	1280	65153	**2415**	6515
74	60%	1280	55809	**1995**	5581
75	50%	1280	46756	**1554**	4676
76	40%	1280	37366	**1096**	3737
77	30%	1280	28253	**649**	2825
78	20%	1280	19031	**200**	1903
79	10%	1280	9625	**0**	963
80	100%	1516	77955	**3200**	7796
81	90%	1516	70483	**3047**	7048
82	80%	1516	62754	**2606**	6275
83	70%	1516	54995	**2148**	5500
84	60%	1516	47086	**1695**	4709
85	50%	1516	39352	**1299**	3935
86	40%	1516	31654	**920**	3165
87	30%	1516	23855	**536**	2386
88	20%	1516	16106	**155**	1611
89	10%	1516	8131	**0**	813

Com base nos valores coletados, mostrados na Tabela 12, é possível calcular o percentual de pacotes perdidos para cada tamanho de quadro. Para esse cálculo utiliza-se a seguinte fórmula:

Percentual de Perda = (Total Perdidos*100) / (Total de enviados)

Fluxo/ Taxa	***64***	***128***	***256***	***512***	***768***	***1024***	***1280***	***1526***
0%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
10%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
20%	0,00%	0,00%	0,18%	0,21%	0,23%	0,23%	0,21%	0,19%
30%	0,13%	0,00%	0,58%	0,66%	0,66%	0,69%	0,67%	0,66%
40%	5,30%	0,00%	1,00%	1,11%	1,13%	1,14%	1,14%	1,13%

50%	8,54%	0,14%	1,44%	1,61%	1,56%	1,56%	1,61%	1,60%
60%	10,12%	1,94%	1,93%	2,15%	2,03%	2,03%	2,07%	2,08%
70%	8,95%	8,52%	2,42%	2,68%	2,51%	2,51%	2,50%	2,63%
80%	11,52%	9,57%	2,95%	2,98%	2,97%	2,99%	2,97%	3,19%
90%	12,51%	10,87%	3,52%	3,47%	3,42%	3,46%	3,46%	3,73%
100%	13,69%	14,03%	6,92%	5,79%	4,46%	4,78%	4,13%	3,94%

Tabela 13 - Percentual de Perda de pacotes por fluxo

Com base nos resultados acima foi produzido o gráfico mostrado na Figura 27, que representa o percentual de pacotes perdidos em relação ao uso da banda disponível.

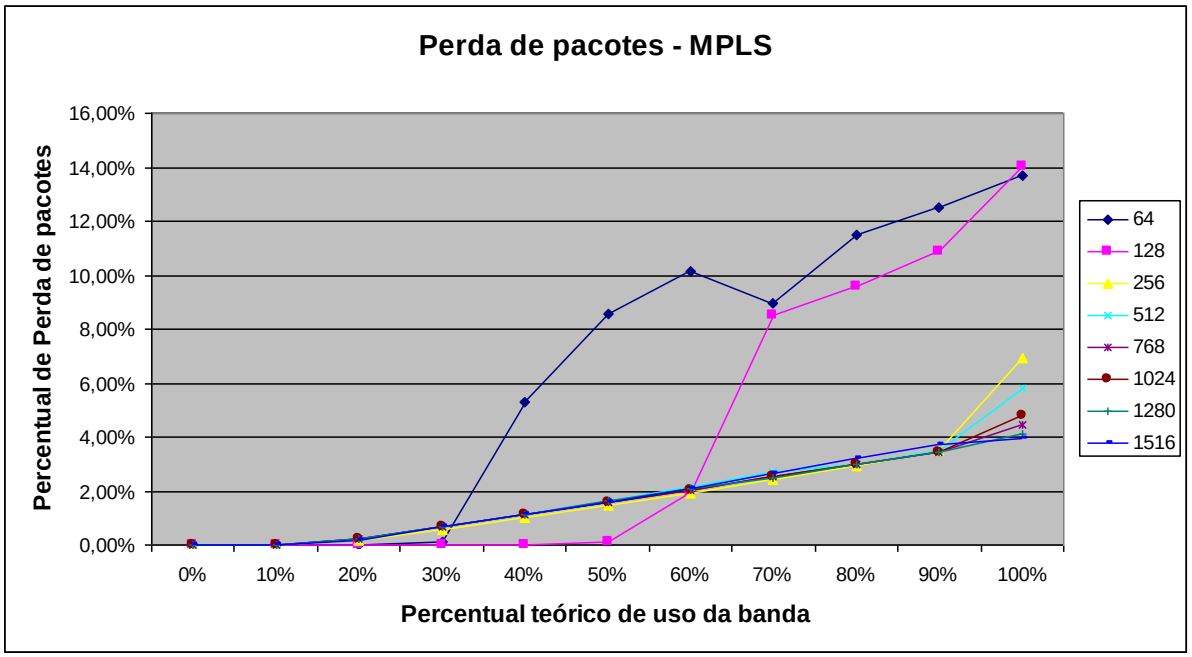


Figura 27 - Gráfico de Perda de pacotes - MPLS

O gráfico da Figura 27 mostra que as perdas de pacotes foram mais acentuadas com quadros de 64 e 128 bytes, estes fluxos apresentam taxas de pacotes mais elevadas. Fica evidente que o módulo MPLS não conseguiu processar as taxas mais altas de pacotes.

As perdas de pacotes apresentaram um percentual baixo (máximo de 7%) com tamanhos de Quadros maiores, de 256 até 1516 bytes. Com tamanhos de Quadros de 64 e 128 bytes as perdas a 14%.

5 CONCLUSÃO

O mercado de telecomunicações apresenta uma tendência e uma necessidade crescente de maiores bandas, maior velocidade e qualidade de serviços. No capítulo 2, são apresentadas as características do MPLS que mostram que ele é um protocolo que atente essas necessidades e agrega algumas vantagens em relação às outras tecnologias existentes, como Fram-relay e ATM. Além disso, o MPLS tornou mais fácil o gerenciamento e a criação de VPNs no modelo Peer-to-Peer.

Este trabalho teve como foco principal a avaliação de desempenho da implementação do MPLS no sistema operacional Linux. A metodologia de avaliação de desempenho foi construída com base nas RFCs 2544 e 5695. A metodologia proposta pelas RFCs se mostrou bastante objetiva e de fácil entendimento. Entretanto, em alguns pontos a metodologia não atendeu as necessidades, por exemplo: a metodologia não previa o uso de mais de um label por pacote o que resultou em modificações nos Quadros utilizados nos experimentos. Isto se deve ao fato de que para a criação de VPNs MPLS L3 são necessários no mínimo 2 labels por pacote. Além disso, para a avaliação da vazão da rede MPLS, a metodologia proposta pela RFC gerou um impasse. A RFC afirma que o valor máximo da vazão é aquele em que durante o fluxo de dados não houver nenhuma perda de pacotes. Porém, as medições mostraram perdas de pacotes em quase todos os fluxos, inclusive naqueles com baixo percentual de uso da banda.

Os resultados das medições mostraram que o desempenho geral do MPLS se aproxima dos valores de referência, com exceção dos fluxos que utilizaram tamanhos de quadros de 64 e 128 bytes. Coincidentemente, estes fluxos são que demandam maiores taxas de pacotes por segundo. Os seguintes fatos reforçam essa constatação:

a. **Vazão**: A vazão medida na rede MPLS foi semelhante aos valores de referência medidos sem o uso do MPLS, porém os fluxos com tamanhos de pacotes de 64 e 128 bytes tiveram um desempenho, respectivamente, de 34% e 55% do valor teórico referência.

b. **Latência**: A latência da rede observada nos experimentos mostrou que o uso do MPLS não impactou no tempo, ou seja, a latência da rede com o MPLS não foi maior ou menor. Os valores medidos seguiram um padrão de crescimento de acordo com o aumento do tamanho do Quadro tanto com MPLS quanto sem.

c. **Perda de Pacotes**: As perdas de pacotes apresentaram um percentual baixo (máximo de 7%) com tamanhos de Quadros maiores, de 256 até 1516 bytes. Com tamanhos de Quadros de 64 e 128 bytes as perdas chegaram a 14%.

Em linhas gerais a metodologia definida neste trabalho se mostrou adequada para a avaliação de desempenho, pois, os resultados obtidos nas medições apresentaram valores coerentes em relação aos valores de referência. Além disso, esta metodologia poderá ser facilmente reutilizada em outros cenários de teste. Neste trabalho os roteadores utilizados eram PCs executando o módulo MPLS-Linux. As mesmas medições podem ser feitas utilizando roteadores reais. As únicas modificações necessárias são o mapeamento dos comandos MPLS-Linux para os comandos MPLS do roteador a ser avaliado. Os scripts de geração e coleta de fluxos não seriam modificados.

REFERÊNCIAS

[BAT 2007] BATTISTI, Gerson. **Modelo de Gerenciamento para Infra-Estruturas de Medições de Desempenho em Redes de Computadores**. Porto Alegre: [s.n.], 2007.

[COS 2008] COSTA, Giovani Hoff Da. **Métricas para Avaliação de Desempenho em Redes QoS sobre IP**. Porto Alegre: [s.n.], 2008.

[GHE 2007] GHEIN, Luc De. **MPLS Fundamentals**. Indianapolis: Cisco Press. 2007.

[HUS 2005] HUSSAIN, Iftekhar. **Fault-tolerant IP and MPLS networks**. Indianapolis: Cisco Press. 2005.

[NOG 2008] NOGUEIRA, Roger Fonseca. **Aplicação de QoS sobre MPLS em equipamentos Cisco**. Porto Alegre, Brasil: UFRGS. 2008.

[PRE 2008] PRETO, Gerson. **Rede MPLS, Tecnologias e Tendências de Evoluções Tecnológicas**. Porto Alegre. 2008.

[RAM 2008] RAMOS, Talles. T. **Encapsulando Frames Ethernet em Conexões MPLS**. Porto Alegre. 2008.

[SCH 2002] SCHABBACH, Tatiana Rotava. **Análise Comparativa de Desempenho de Redes IP e ATM com tráfego Multimídia Interativo**. Porto Alegre, Brasil: UFRGS. 2002.

[SIL 2005] SILVA, Lino. D. **Virtual Private Network.** 2ªed.. São Paulo: Novatec Editora. 2005.

[BOA 2004] BOAVA, Adão. **Estratégia de Projeto de VPN MPLS com Qualidade de Serviço (QoS)**. Campinas, Brasil: Unicamp. 2004.

[KUR 2007] KUROSE, J. F., & ROSS, K. W. **Redes de Computadores e a Internet.** Pearson. 2007.

[JAI 1991] JAIN, Raj. **The art of computer systems performance analysis**: techniques for experimental desingn, measurement, simulation, and modeling. New York: John Wiley, 1991.

[AKH 2009]AKHTER, A; ASATI R; PIGNATARO, C. **MPLS Forwarding Benchmarking Methodology for IP Flows**: RFC 5695. [S.l.]: Internet Engineering Task Force, Network Working Group, 2009.

[BRA 1999] BRADNER, S; MCQUAID, J. **Benchmarking Methodology for Network Interconnect Devices**: RFC 2544. [S.l.]: Internet Engineering Task Force, Network Working Group, 1999.

[ROS 2001] ROSEN, E; TAPPAN, D; FEDORKOW, G; REKHTER, Y. **MPLS Label Stack Encoding**: RFC 3032. [S.l.]: Internet Engineering Task Force, Network Working Group, 2001.

[DEA 2002]DE ASSIS, Martin Seefelder. **Introdução ao MPLS**, Rio de Janeiro, Fevereiro de 2002. Disponível em: <http://www.gta.ufrj.br/grad/01_2/mpls/>. Acesso em: set. 2010.

[MPL 2010] MPLS_LINUX GROUP. **MPLS Linux project**, 2010. Disponível em: <http://http://sourceforge.net/apps/mediawiki/mpls-linux/index.php?title=Main_Page>. Acesso em: Nov. 2010.

ANEXO A - SIMULAÇÃO DE REDE MPLS COM O GNS 3

O GNS 3 é uma ferramenta gráfica de simulação de redes de computadores desenvolvida pela Cisco. O GNS é muito utilizado para o estudo dos dispositivos da CISCO e simulação dos mesmos, também é útil para estudos de protocolos e observação de seu funcionamento. Neste trabalho, o GNS foi utilizado para simular uma rede operando com o protocolo MPLS. O objetivo disso é consolidar os conceitos estudados sobre o MPLS e observar o funcionamento do protocolo.

Com auxilio do GNS, é possível observar o encapsulamento do MPLS em um frame Ethernet e também o encaminhamento dos pacotes através das trocas do Labels. Na Figura 28 é mostrada uma topologia básica de rede composta por dois PCs (são dois roteadores configurados para se comportar como PCs) e três roteadores, sendo o R1 e o R3 de ingresso/egresso da rede MPLS, e o R2 de núcleo que somente encaminha os pacotes baseando-se no Label presente.

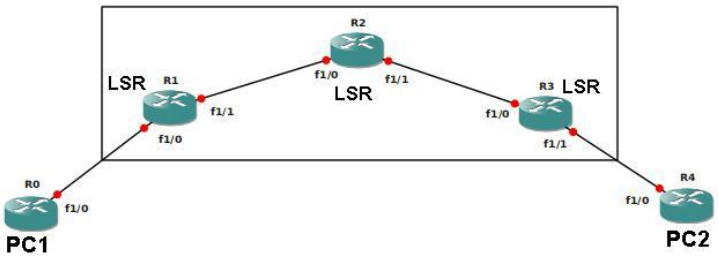
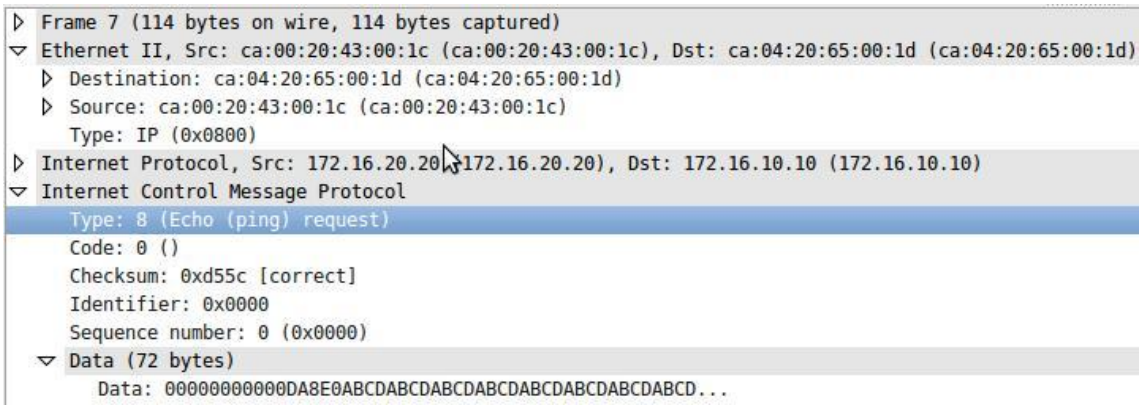


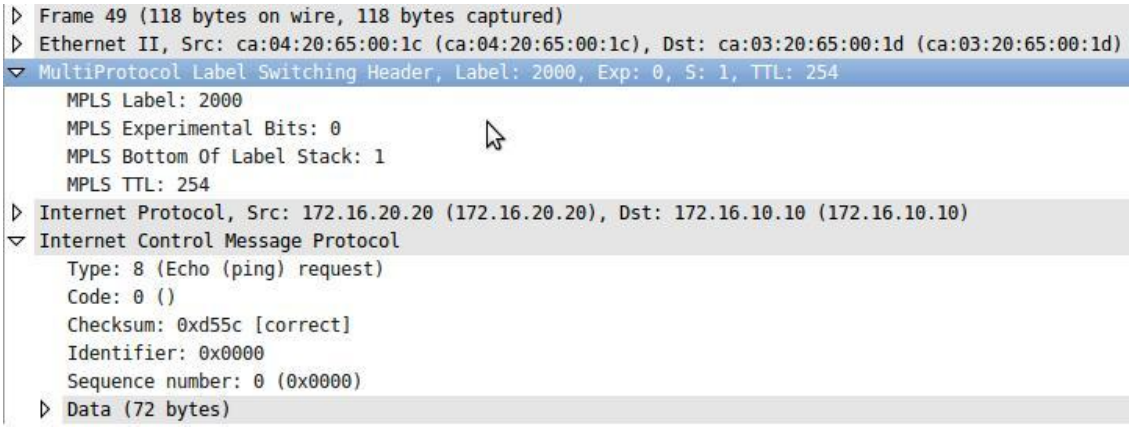
Figura 28 -Topologia básica no GNS 3

Abaixo é mostrado um trace de um pacote de ping request partindo do PC2 até o PC1, os pacotes foram capturados com uso do aplicativo Wireshark:

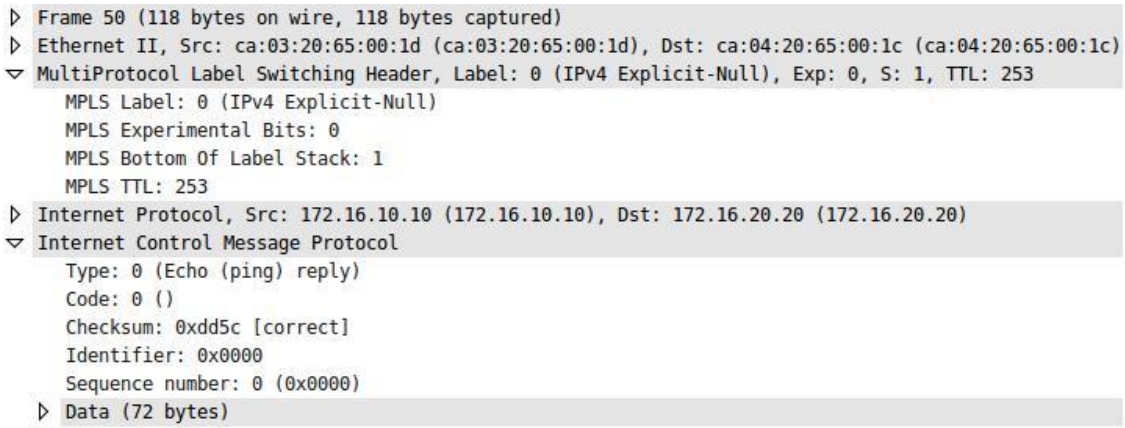
Pacote partindo do PC 2 para o roteador R3:



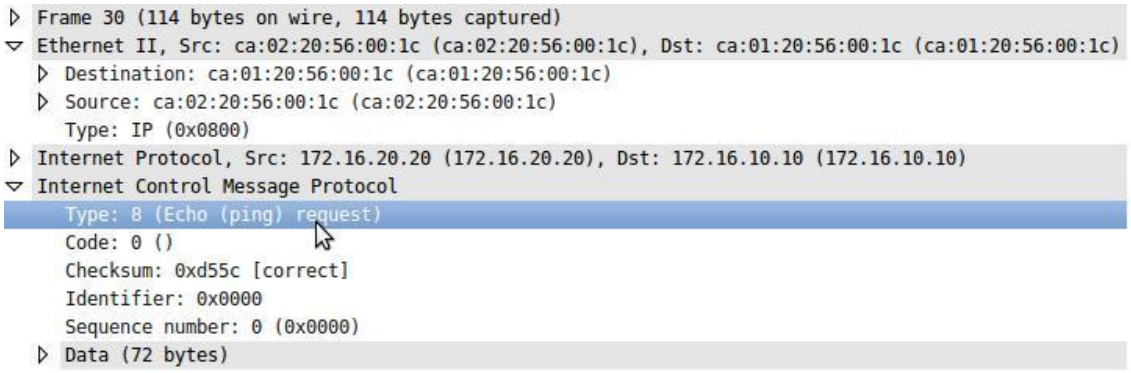
Pacote ingressando na rede MPLS, o R3 insere o Label MPLS **2000** e redireciona para o R2:



Pacote R3 lê o Label, altera e encaminha o R1:



Pacote deixando a rede MPLS, sendo removido o Label MPLS, R1 envia para o PC1:


Frame 30 (114 bytes on wire, 114 bytes captured)
Ethernet II, Src: ca:02:20:56:00:1c (ca:02:20:56:00:1c), Dst: ca:01:20:56:00:1c (ca:01:20:56:00:1c)
Destination: ca:01:20:56:00:1c (ca:01:20:56:00:1c)
Source: ca:02:20:56:00:1c (ca:02:20:56:00:1c)
Type: IP (0x0800)
Internet Protocol, Src: 172.16.20.20 (172.16.20.20), Dst: 172.16.10.10 (172.16.10.10)
Internet Control Message Protocol
Type: 8 (Echo (ping) request)
Code: 0 ()
Checksum: 0xd55c [correct]
Identifier: 0x0000
Sequence number: 0 (0x0000)
Data (72 bytes)

ANEXO B - SCRIPTS DE CONFIGURAÇÃO

Os scripts abaixo foram utilizados para a configuração do ambiente de testes. As configurações consistem nos seguintes itens:

a. Definição do papel de cada PCs na rede, ou seja, se será um roteador MPLS ou não.

b. Configuração da interfaces de rede (eth0, eth1... etc) e os endereços IP referente a cada uma delas.

c. Adição das rotas de encaminhamento de pacotes, através do ***ip route***

d. Configuração das tabelas de encaminhamento MPLS, e dos rótulos que serão utilizados.

Para que as máquinas estejam habilitadas a executar os scripts abaixo, é necessário que o kernel Linux com o módulo MPLS já esteja instalado, que o Ip Route seja reinstalado com patch MPLS.

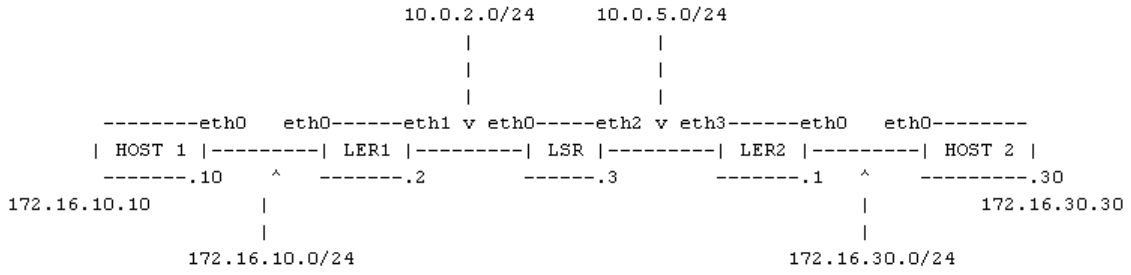


Figura 29 -Topologia após configuração

Configuração para o Host 1:

```
>echo Configurando interfaces no Host1
>ifconfig eth0 down
>ifconfig eth0 172.16.10.10 netmask 255.255.255.0 up
>echo "eth0 is 172.16.10.10"
>ip route add 172.16.30.0/24 via 172.16.10.2 dev eth0
>echo Iniciando roteamento
```

```
>echo 1 > /proc/sys/net/ipv4/ip_forward
>echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

Configuração para o Host 2:

```
>echo Configurando interfaces no Host2
>ifconfig eth0 down
>ifconfig eth0 172.16.30.30 netmask 255.255.255.0 up
>echo "eth0 is 172.16.30.30"
>ip route add 172.16.10.0/24 via 172.16.30.1 dev eth0
>echo Iniciando roteamento
>echo 1 > /proc/sys/net/ipv4/ip_forward
>echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
```

Configuração para o LER 1:

```
>echo Configurando as interfaces no LER1
>ifconfig eth0 down
>ifconfig eth0:1 down
>ifconfig eth0 172.16.10.2 netmask 255.255.255.0 up
>ifconfig eth0:1 10.0.2.2 netmask 255.255.255.0 up

>echo eth0 is 172.16.10.2
>echo eth0:1 is 10.0.2.2
>echo Iniciando roteamento
>echo 1 >/proc/sys/net/ipv4/ip_forward
>echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
>echo No MPLS debug
>echo 0 >/sys/mpls/debug

>echo 'HOST1->HOST2 VPN1'
#adiciona labels 1000 e 100
>key1=`mpls nhlfe add key 0 instructions push gen 1000 nexthop
eth0:1 ipv4 10.0.2.3|grep key |cut -c 17-26`
>key2=`mpls nhlfe add key 0 instructions push gen 100 forward
$key1|grep key|cut -c 17-26`
>ip route add 172.16.30.0/24 via 10.0.2.3 table 1 mpls $key2
```

```
>echo 'HOST2->HOST1 VPN1'
#No caminho de retorno do pacote, remove os labels
>mpls labelspace set dev eth0:1 labelspace 0
>mpls ilm add label gen 4001 labelspace 0
>mpls ilm add label gen 100 labelspace 0

>ip rule add from 172.16.30.0/24 table 1
>ip rule add from 172.16.10.0/24 table 1

>ip route add 172.16.10.0/24 dev eth0 table 1
>ip route add 10.0.2.0/24 dev eth0:1 table 1
```

Configuração para o LSP:

```
>echo Configurando as interfaces no LSP
>ifconfig eth0 down
>ifconfig eth0:2 down
>ifconfig eth0 10.0.2.3 netmask 255.255.255.0 up
>ifconfig eth0:2 10.0.5.3 netmask 255.255.255.0 up
>echo eth0 is 10.0.2.3
>echo eth0:2 is 10.0.5.3
>echo Iniciando Roteamento
>echo "1" >/proc/sys/net/ipv4/ip_forward
>echo 0 > /proc/sys/net/ipv4/conf/all/rp_filter
>echo No MPLS debug
>echo 0 >/sys/mpls/debug

>echo 'HOST1 -> HOST2 in VPN1'
#switch label 1000
>mpls labelspace set dev eth0 labelspace 0
>mpls ilm add label gen 1000 labelspace 0

>var=`mpls nhlfe add key 0 instructions push gen 3000 nexthop
eth0:2 ipv4 10.0.5.1 |grep key | cut -c 17-26`
#switch labels 1000 para 3000
>mpls xc add ilm_label gen 1000 ilm_labelspace 0 nhlfe_key $var
```

```
>echo 'HOST2 -> HOST1 in VPN1'
#switch 4000 para 4001
>mpls labelspace set dev eth2 labelspace 0
>mpls ilm add label gen 4000 labelspace 0
>var1=`mpls nhlfe add key 0 instructions push gen 4001 nexthop
eth0 ipv4 10.0.2.2 | grep key |cut -c 17-26`
>mpls xc add ilm_label gen 4000 ilm_labelspace 0 nhlfe_key $var1
```

Configuração para o LER 2:

```
>echo Configurando as interfaces no LER2
>ifconfig eth0 down
>ifconfig eth0:3 down
>ifconfig eth0 172.16.30.1 netmask 255.255.255.0 up
>ifconfig eth0:3 10.0.5.1 netmask 255.255.255.0 up
>echo eth0 is 172.16.30.1
>echo eth0:3 is 10.0.5.1
>echo Iniciando roteamento
>echo "1" >/proc/sys/net/ipv4/ip_forward
>echo 0 >/proc/sys/net/ipv4/conf/all/rp_filter
>echo No MPLS debug
>echo "0" >/sys/mpls/debug

echo 'HOST1->HOST2 VPN1'
#No caminho de retorno do pacote, remove os labels
>mpls labelspace set dev "eth0:3" labelspace 0
>mpls ilm add label gen 3000 labelspace 0
>mpls ilm add label gen 100 labelspace 0

>echo 'HOST2->HOST1 VPN1'
#adiciona labels 4000 e 100
>key1=`mpls nhlfe add key 0 instructions push gen 4000 nexthop
"eth0:3" ipv4 10.0.5.3|grep key |cut -c 17-26`
>key2=`mpls nhlfe add key 0 instructions push gen 100 forward
$key1|grep key|cut -c 17-26`
>ip route add 172.16.10.0/24 via 10.0.5.3 table 1 mpls $key2
>ip rule add from 172.16.30.0/24 table 1
>ip rule add from 172.16.10.0/24 table 1
```

```
>ip route add 172.16.30.0/24 dev eth0 table 1
>ip route add 10.0.5.0/24 dev "eth0:3" table 1
```

ANEXO C - RUDE & CRUDE E SCRIPTS DE TESTE

Para gerar os fluxos de tráfego para o experimento foi utilizada a ferramenta RUDE (*Real-time UDP Data Emitter*). Com essa ferramenta é possível simular tráfego de áudio e vídeo tráfego de dados genérico. A ferramenta RUDE utiliza um *script* com os parâmetros para o experimento. Nesse script pode ser definida a duração, o início de cada fluxo de dados, porta origem, porta destino, endereço origem, endereço destino, tamanho dos pacotes e taxa de geração dos pacotes. Já a ferramenta CRUDE tem o objetivo de efetuar a captura dos dados no receptor (destino do tráfego de dados), o CRUDE recebe o tráfego de dados criado pelo RUDE e armazena as informações relevantes de cada pacote em arquivos de *log*. Com base nesses arquivos é possível extrair as informações para a avaliação de desempenho. Essa ferramenta foi desenvolvida para plataforma Linux

Para diparar o fluxo no emissor utiliza-se o comando:

- rude –P 90 –s <nome_do_arquivo_de_fluxo.cfg>

Para capturar o fluxo no receptor utiliza-se o comando:

- crude –P 90 –s <nr_fluxo1>, <nr_fluxo2>, <nr_fluxoN>

O parâmetro –P é utilizado para aumentar a prioridade do processo no sistema operacional Linux.

Um arquivo de fluxo é composto basicamente por uma tag de inicio (START), e um comando de início de fluxo (ON) e um comando de fim de fluxo (OFF) da seguinte forma:

Tempo(ms)	**Id**	**PortaO**	**IP**	**PortaD**	**Taxa(pps)**	**Size**	**Pct**
00000	0001	ON	3002	127.0.0.1:10001	CONSTANT	148809	20
60000	0001	OFF					

Em seguida, serão mostrados os scripts utilizados para as medições de Vazão, Perda de Pacotes e Latência da rede especificada para o experimento.

Os comandos abaixo descrevem os fluxos utilizados para os testes de vazão e perda de pacotes, cada fluxo pode ser colocado em um arquivo separado e chamado automaticamente por um Shell Script.

START NOW

##Envia um fluxo de pacotes a uma taxa de 148809 pps

##com tamanho de:
<data> + <UDP> + <IP> + <ETH> = <TOTAL>
20 + 8 + 20 + 16 = 64 bytes
00000 0001 ON 3002 172.16.30.30:10001 CONSTANT 148809 20
60000 0001 OFF
Duracao 60 segundos

START NOW
##Envia um fluxo de pacotes a uma taxa de 84459 pps
##com tamanho de :
<data> + <UDP> + <IP> + <ETH> = <TOTAL>
84 + 8 + 20 + 16 = 128 bytes
00000 0002 ON 3002 172.16.30.30:10001 CONSTANT 84459 84
60000 0002 OFF
Duracao 60 segundos

START NOW
##Envia um fluxo de pacotes a uma taxa de 45289 pps
##com tamanho de :
<data> + <UDP> + <IP> + <ETH> = <TOTAL>
212 + 8 + 20 + 16 = 256 bytes
00000 0003 ON 3002 172.16.30.30:10001 CONSTANT 45289 212
60000 0003 OFF
Duracao 60 segundos

START NOW
##Envia um fluxo de pacotes a uma taxa de 23496 pps
##com tamanho de :
<data> + <UDP> + <IP> + <ETH> = <TOTAL>
468 + 8 + 20 + 16 = 512 bytes
00000 0004 ON 3002 172.16.30.30:10001 CONSTANT 23496 468
60000 0004 OFF
Duracao 60 segundos

START NOW

```
##Envia um fluxo de pacotes a uma  taxa de 15862 pps
##com tamanho de:
##                    <data> + <UDP> + <IP> + <ETH> = <TOTAL>
##                     724  +   8   + 20  +   16  = 768 bytes
00000 0005 ON 3002 172.16.30.30:10001 CONSTANT 15862 724
60000 0005 OFF
## Duracao 60 segundos
START NOW
##Envia um fluxo de pacotes a uma  taxa de 11973 pps
##com tamanho de :
##                    <data> + <UDP> + <IP> + <ETH> = <TOTAL>
##                     980  +   8   + 20  +   16  = 1024 bytes
00000 0006 ON 3002 172.16.30.30:10001 CONSTANT 11973 980
60000 0006 OFF
## Duracao 60 segundos

START NOW
##Envia um fluxo de pacotes a uma  taxa de 9615 pps
##com tamanho de:
##                    <data> + <UDP> + <IP> + <ETH> = <TOTAL>
##                     1236  +   8   + 20  +   16  = 1280 bytes
00000 0007 ON 3002 172.16.30.30:10001 CONSTANT 9615 1236
60000 0007 OFF
## Duracao 60 segundos

START NOW
##Envia um fluxo de pacotes a uma  taxa de 8127 pps
##com tamanho de :
##                    <data> + <UDP> + <IP> + <ETH> = <TOTAL>
##  MTU =1500     1464  +   8   + 20  +   16  = 1508 bytes
## Dentro da rede MPLS serão incluídos mais 8 bytes de labels
00000 0008 ON 3002 172.16.30.30:10001 CONSTANT 8127 1464
60000 0008 OFF
## Duracao 60 segundos
```

Para medir a perda de quadros usou-se os 8 fluxos já definidos, porém eles foram subdivididos em 10 sub-fluxos para da fluxo original. A subdivisão consiste na redução da taxa de pacotes em intervalos de 10%, iniciando a uma taxa de 100% do valor teórico até chegar a 10%. Desta forma, termos um total de 80 fluxos.

Abaixo são mostrados os fluxos para medição da perda de quadros:

##Envia um fluxos de pacotes a taxas de:

148809, 133928, 119047, 104166, 89285,

##74405, 59524, 44643, 29762, 14881 pps

##com tamanho de:

<data> + <UDP> + <IP> + <ETH> = <TOTAL>

20 + 8 + 20 + 16 = 64 bytes

START NOW

00000 0010 ON 3002 172.16.30.30:10001 CONSTANT 148809 20

10000 0010 OFF ## Duracao 10 segundos - 100%

00000 0011 ON 3002 172.16.30.30:10001 CONSTANT 133928 20

10000 0011 OFF ## Duracao 10 segundos - 90%

00000 0012 ON 3002 172.16.30.30:10001 CONSTANT 119047 20

10000 0012 OFF ## Duracao 10 segundos - 80%

00000 0013 ON 3002 172.16.30.30:10001 CONSTANT 104166 20

10000 0013 OFF ## Duracao 10 segundos - 70%

00000 0014 ON 3002 172.16.30.30:10001 CONSTANT 89285 20

10000 0014 OFF ## Duracao 10 segundos - 60%

00000 0015 ON 3002 172.16.30.30:10001 CONSTANT 74405 20

10000 0015 OFF ## Duracao 10 segundos - 50%

00000 0016 ON 3002 172.16.30.30:10001 CONSTANT 59524 20

10000 0016 OFF ## Duracao 10 segundos - 40%

00000 0017 ON 3002 172.16.30.30:10001 CONSTANT 44643 20

10000 0017 OFF ## Duracao 10 segundos - 30%

00000 0018 ON 3002 172.16.30.30:10001 CONSTANT 29762 20

10000 0018 OFF ## Duracao 10 segundos - 20%

00000 0019 ON 3002 172.16.30.30:10001 CONSTANT 14881 20

10000 0019 OFF ## Duracao 10 segundos - 10%

Os fluxos para os demais tamanhos de quadros é apenas uma repetição da subdivisão acima, substituindo apenas as taxas de pacotes com base na tabela xx, e trocando o número identificador do fluxo. Desta forma, eles serão omitidos aqui.

Abaixo, é mostrado o Shell Script que dispara os testes de perda de quadros:

```
#!/bin/sh

echo "\n*** Script de Teste 02 - MPLS - Frame Loss***"
echo "Executa os seguintes fluxos:\n"
echo "Fluxo 01 - Tamanho de Quadro:   64 bytes "
echo "Fluxo 02 - Tamanho de Quadro:  128 bytes "
echo "Fluxo 03 - Tamanho de Quadro:  256 bytes "
echo "Fluxo 04 - Tamanho de Quadro:  512 bytes "
echo "Fluxo 05 - Tamanho de Quadro:  768 bytes "
echo "Fluxo 06 - Tamanho de Quadro: 1024 bytes "
echo "Fluxo 07 - Tamanho de Quadro: 1280 bytes "
echo "Fluxo 08 - Tamanho de Quadro: 1516 bytes "

echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
     then
     echo "Abortando ..."
     exit
fi

echo "Fluxo 01:\t>rude -P 89 -s size-64.cfg [100% até 10%]"
./64.sh; # Inicia o processo rude para o trafego 1

echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
     then
     echo "Abortando ..."
     exit
fi

echo "Fluxo 02:\t>rude -P 89 -s size-128.cfg"
./128.sh; # Inicia o processo rude para o trafego 2
```

```
echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
      then
      echo "Abortando ..."
      exit
fi

echo "Fluxo 03:\t>rude -P 89 -s size-256.cfg"
./256.sh; # Inicia o processo rude para o trafego 3

echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
      then
      echo "Abortando ..."
      exit
fi

echo "Fluxo 04:\t>rude -P 89 -s size-512.cfg"
./512.sh; # Inicia o processo rude para o trafego 4

echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
      then
      echo "Abortando ..."
      exit
fi

echo "Fluxo 05:\t>rude -P 89 -s size-768.cfg"
./768.sh;  # Inicia o processo rude para o trafego 5

echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
      then
```

```
      echo "Abortando ..."
      exit
fi

echo "Fluxo 06:\t>rude -P 89 -s size-1024.cfg"
./1024.sh; # Inicia o processo rude para o trafego 6

echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
      then
      echo "Abortando ..."
      exit
fi

echo "Fluxo 07:\t>rude -P 89 -s size-1280.cfg"
./1280.sh # Inicia o processo rude para o trafego 7

echo -n "Continuar[ sim | nao ]: "
read answer
if [ "$answer" != "sim" ]
      then
      echo "Abortando ..."
      exit
fi

echo "Fluxo 08:\t>rude -P 89 -s size-1516.cfg"
./1516.sh; # Inicia o processo rude para o trafego 8

echo "\nFIM do Script de Teste 02 MPLS - Fram Loss"
```

Abaixo é mostrado Script para avaliação da Latência:

```
#!/bin/sh

echo "\n*** Script de Teste 04 - Latencia - Ethernet ***"
echo "Executa os seguintes fluxos:\n"
```

```
echo "Fluxo 01 - 100 PINGs Tamanho de Quadro:   64 bytes |"
echo "Fluxo 02 - 100 PINGs Tamanho de Quadro:  128 bytes |"
echo "Fluxo 03 - 100 PINGs Tamanho de Quadro:  256 bytes |"
echo "Fluxo 04 - 100 PINGs Tamanho de Quadro:  512 bytes |"
echo "Fluxo 05 - 100 PINGs Tamanho de Quadro:  768 bytes |"
echo "Fluxo 06 - 100 PINGs Tamanho de Quadro: 1024 bytes |"
echo "Fluxo 07 - 100 PINGs Tamanho de Quadro: 1280 bytes |"
echo "Fluxo 08 - 100 PINGs Tamanho de Quadro: 1516 bytes |"

ping 172.16.30.30 -c 100 -s 22
sleep 10;
ping 172.16.30.30 -c 100 -s 84
sleep 10;
ping 172.16.30.30 -c 100 -s 212
sleep 10;
ping 172.16.30.30 -c 100 -s 468
sleep 10;
ping 172.16.30.30 -c 100 -s 724
sleep 10;
ping 172.16.30.30 -c 100 -s 980
sleep 10;
ping 172.16.30.30 -c 100 -s 1236
sleep 10;
ping 172.16.30.30 -c 100 -s 1464
echo "\nFIM do Script de Teste 04 "
```