

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES

ROBER COSTA GOULARTE
VINÍCIUS KAWASAKI

**IMPLEMENTAÇÃO DE QoS EM REDES MPLS ATRAVÉS DO
SIMULADOR GNS3**

TRABALHO DE CONCLUSÃO DE CURSO

CURITIBA
2016

ROBER COSTA GOULARTE
VINÍCIUS KAWASAKI

IMPLEMENTAÇÃO DE QoS EM REDES MPLS ATRAVÉS DO SIMULADOR GNS3

Trabalho de Conclusão de Curso de Graduação, apresentado ao Curso Superior de Tecnologia em Sistemas de Telecomunicações, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof.^a Dr.^a Tânia Lúcia Monteiro.

CURITIBA
2016

TERMO DE APROVAÇÃO

ROBER COSTA GOULARTE
VINÍCIUS KAWASAKI

IMPLEMENTAÇÃO DE QoS EM REDES MPLS ATRAVÉS DO SIMULADOR GNS3

Este trabalho de conclusão de curso foi apresentado no dia 24 de outubro de 2016, como requisito parcial para obtenção do título de Tecnólogo em Sistemas de Telecomunicações, outorgado pela Universidade Tecnológica Federal do Paraná. Os alunos foram arguidos pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado.

Prof. Dr. Danillo Leal Belmonte
Coordenador de Curso
Departamento Acadêmico de Eletrônica

Prof. M.Sc. Sérgio Moribe
Responsável pela Atividade de Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Kleber Kendy Horikawa Nabas
UTFPR

Prof. M. Alexandre Jorge Miziara
UTFPR

Prof.^a Dr.^a Tânia Lúcia Monteiro
Orientador - UTFPR

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

RESUMO

GOULARTE, Rober Costa e KAWASAKI, Vinícius. **Implementação de QoS em redes MPLS através do simulador GNS3**. 2016. 77 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

Este trabalho é voltado ao estudo da solução MPLS, e trata especificamente da implementação de QoS em uma rede local através de um simulador. O foco principal do trabalho é o de apresentar os conceitos da solução MPLS e demonstrar a importância da realização de simulações de projetos, neste caso através do software GNS3, antes da implementação real dos mesmos. O projeto em questão, aplicado em um ambiente de simulação, envolve a comunicação de duas máquinas, onde os caminhos para a troca de dados e informações encontram-se em uma rede configurada com MPLS. Nessa rede, há a priorização de tráfego de vídeo em relação a outros tipos de tráfego. Considerando a gama de opções e de extensões que o MPLS permite explorar, pode-se concluir que os principais resultados esperados foram obtidos com sucesso.

Palavras chave: MPLS. QoS. GNS3. Simulação.

ABSTRACT

GOULARTE, Rober Costa e KAWASAKI, Vinícius. **QoS implementation in MPLS networks by GNS3 simulator**. 2016. 77 f. Trabalho de Conclusão de Curso (Curso Superior de Tecnologia em Sistemas de Telecomunicações), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2016.

This work is oriented to the study of MPLS solution, and deals specifically of QoS implementation in a local network using a simulator. The main focus of this work is to present the concepts of MPLS solution and demonstrate the importance of conducting projects simulation, in this case through the GNS3 software, before real implementation. The project in question, applied in a simulation environment, involves communication between two hosts, where the exchange of data and information are on a network configured in MPLS. In this network, there is priority of video traffic compared with other types of traffic. Considering the range of options and extensions to the MPLS allow to explore, it can be concluded that the main expected results were successfully obtained.

Keywords: MPLS. QoS. GNS3. Simulation.

LISTA DE ILUSTRAÇÕES

Figura 1 - Cabeçalho MPLS	16
Figura 2 - Componentes da arquitetura <i>IntServ</i>	20
Figura 3 - Blocos funcionais do <i>DiffServ</i>	21
Figura 4 - Arquitetura de serviços diferenciados	22
Figura 5 - Processo de QoS	26
Figura 6 - Software GNS3 (Versão)	30
Figura 7 - Rede MPLS simulada no GNS3.....	31
Figura 8 - Software VirtualBox (Versão).....	33
Figura 9 - Exemplo de <i>appliance</i> de um Sistema Operacional.....	33
Figura 10 - Gerenciador do VirtualBox	34
Figura 11 - Informações básicas do Sistema Operacional da máquina real	35
Figura 12 - Informações básicas do Sistema Operacional da máquina virtual	35
Figura 13 - Configuração do adaptador de rede da máquina virtual	36
Figura 14 - Janela de preferências do GNS3	37
Figura 15 - Janela de seleção de máquinas virtuais	37
Figura 16 - Janela com uma máquina virtual importada.....	38
Figura 17 - Janela de seleção de uma imagem de roteador	38
Figura 18 - Nome e plataforma do roteador	39
Figura 19 - Memória alocada ao roteador	39
Figura 20 - Slots <i>PA-GE (GigabitEthernet)</i>	40
Figura 21 - Janela de configuração do <i>Idle-PC</i>	41
Figura 22 - Mensagem do valor de <i>Idle-PC</i> encontrado.....	41
Figura 23 - Janela com uma imagem de roteador importada.....	41
Figura 24 - Exemplo de um esquemático de diferenciação de áreas.....	43
Figura 25 - Host 1.....	44
Figura 26 - Host 2.....	44
Figura 27 - Host 3.....	44
Figura 28 - Janela do Firewall do Windows.....	45
Figura 29 - Rotas ativas do Host 1	46
Figura 30 - Rotas ativas do Host 2	46
Figura 31 - Rotas ativas do Host 3.....	46
Figura 32 - Software OmniPeek (Versão)	47
Figura 33 - Processo de captura do fluxo em tráfego na rede	48
Figura 34 - Aba de opções de monitoração.....	48
Figura 35 - Opção com a conexão local do adaptador.....	49
Figura 36 - Janela de definição da velocidade da rede	49
Figura 37 - Janela de monitoramento de utilização da rede com QoS.....	50
Figura 38 - Janela de monitoramento de utilização da rede sem QoS.....	51

LISTA DE QUADROS

Quadro 1 - Classes utilizadas na simulação do ambiente QoS.....	18
Quadro 2 - Valores de QoS.....	23
Quadro 3 - Legenda dos parâmetros de QoS	23
Quadro 4 - Formação do quadro de ToS	24
Quadro 5 - Descrição do quadro de ToS.....	24
Quadro 6 - Sub-redes utilizadas no projeto.....	32

LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

AF	Assured Forwarding
ASICs	Application Specific Integrated Circuits
ATM	Asynchronous Transfer Mode
CIDR	Classless Inter-Domain Routing
CoS	Class of Service
CPU	Central Processing Unit
DiffServ	Differentiated Services
DP	Drop Probability
DSCP	Differentiated Services Code Point
ECN	Engineering Change Notice
FEC	Forwarding Equivalent Class
GE	Gigabit Ethernet
GNS3	Graphical Network Simulator-3
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IOS	Internetwork Operating System
IP	Internet Protocol
IPP	IP Precedence
MPLS	Multiprotocol Label Switching
OSPF	Open Shortest Path First
PA	Port Adapters
PC	Personal Computer
QoS	Quality of Service
RAM	Random Access Memory
RSVP	Resource Reservation Protocol
SPF	Shortest Path First
TCP	Transmission Control Protocol
TE	Traffic Engineering
ToS	Type of Service
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network

SUMÁRIO

1	INTRODUÇÃO	9
1.1	ASPECTOS HISTÓRICOS	9
1.2	CONTEXTUALIZAÇÃO	10
1.3	PROBLEMA	11
1.4	JUSTIFICATIVA	12
1.5	OBJETIVOS	13
1.5.1	Objetivo Geral	13
1.5.2	Objetivos Específicos	13
1.6	PROCEDIMENTOS METODOLÓGICOS	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	CARACTERÍSTICAS DO MPLS	16
2.1.1	O Cabeçalho MPLS	16
2.1.2	Estrutura do MPLS	17
2.2	QoS COM MPLS	17
2.3	ARQUITETURA DE QoS	19
2.3.1	Serviços Integrados (<i>IntServ</i>)	19
2.3.2	Serviços Diferenciados (<i>DiffServ</i>)	20
2.4	VALORES DE QoS	23
2.4.1	Legenda dos Parâmetros de QoS	24
2.5	MECANISMOS DE QoS	25
2.5.1	Classificação e Marcação	26
2.5.2	Policimento	26
2.5.3	Engenharia de Tráfego com MPLS	27
2.6	PROTOCOLOS	27
3	DESENVOLVIMENTO DO TEMA	29
3.1	IMPLEMENTANDO A REDE MPLS	29
3.1.1	A Ferramenta de Simulação do Ambiente	29
3.1.2	Emulação de Máquinas Virtuais	32
3.1.3	Configuração de Softwares	36
3.1.4	Descrição das Configurações dos Roteadores	42
3.1.5	Configuração das Máquinas Virtuais	44
3.1.6	Análise da Prioridade de Tráfego	47
3.1.7	Resultados das Simulações	50
4	CONSIDERAÇÕES FINAIS	52
	REFERÊNCIAS	54
	GLOSSÁRIO	57
	APÊNDICE	59

1 INTRODUÇÃO

Atualmente, as redes de computadores vêm sofrendo grandes mudanças e avanços tecnológicos no ramo das telecomunicações, o que requer a existência de sistemas de transmissão de dados com alto desempenho. A tecnologia *Multiprotocol Label Switching* (MPLS) surge como uma solução que permite a evolução, otimização e flexibilidade da rede, através de alternativas como Engenharia de Tráfego (TE), Redes Virtuais Privadas (VPNs) e Qualidade de Serviço (QoS), sendo o último item o foco principal deste projeto de pesquisa.

O QoS é um recurso muito utilizado para distinguir várias classes de tráfego com regras predeterminadas, ou seja, é capaz de fornecer vários níveis de tratamento para diferentes tipos de tráfego na rede. Todo esse processo de integração e convergência faz com que seja fundamental a utilização do QoS para atender melhor às aplicações que necessitam de tratamento diferenciado.

Ao aplicar uma técnica que priorize o QoS é possível otimizar o uso da banda passante de uma rede, o que possibilita transportar vários tipos de tráfego como vídeo, voz e dados de maneira eficiente e sem interferência mútua, de forma que possa atender às necessidades do cliente final.

1.1 ASPECTOS HISTÓRICOS

Na segunda metade da década de 1990, a tecnologia *Asynchronous Transfer Mode* (ATM), embora ainda com preço elevado e protocolo complexo em planos e camadas (TANENBAUM, 2011), já era a tecnologia dominante para a construção de backbones. Ao mesmo tempo, já se sabia que a pilha de protocolos *Transmission Control Protocol / Internet Protocol* (TCP/IP) era um padrão de fato no mundo, e que todas as tecnologias que fossem desenvolvidas a partir de então deveriam ser compatíveis com esses protocolos. No entanto, a natureza da tecnologia ATM, com células de tamanho fixo e qualidade de serviço intrínseca, difere totalmente da natureza do protocolo IP.

O mapeamento de pacotes IPs no ATM é uma tarefa complexa, já que os processos de segmentação em pequenas células e a remontagem dos pacotes acarretam desperdício de banda passante, acrescentando informações adicionais e exigindo mais processamento dos roteadores. Desse modo, a união desses dois mundos nunca permitiu uma utilização plena e harmônica das duas tecnologias (OLIVEIRA *et al*, 2012).

Nessa mesma década (1990) surgiram pesquisas que levaram a uma quebra total de paradigma e foram inicialmente chamadas de “comutação IP”. Alguns fabricantes entendiam que pacotes IPs não precisavam ser roteados nos núcleos da rede e que era possível adquirir a qualidade de serviço de redes ATM por meio da comutação de pacotes IPs. Tal comutação seria realizada por rótulos adicionados a cada pacote (OLIVEIRA *et al*, 2012).

Assim, o MPLS é uma tecnologia desenvolvida no âmbito do *Internet Engineering Task Force* (IETF) (LUCEK e MINEI, 2005), inicialmente como uma tentativa de padronizar a comutação de pacotes baseada na troca de rótulos e, com isso, melhorar a eficiência de fluxos de tráfegos através da rede, modificando um paradigma fundamental até então existente nas redes IPs com a inserção de um rótulo ao datagrama, propiciando assim a comutação IP.

O MPLS é uma tecnologia aberta que foi apresentada inicialmente como uma solução que possibilitava melhorar o desempenho das redes IPs na função de encaminhamento de pacotes IPs, combinando o processo de roteamento de nível 3 com a comutação de nível 2, para realizar o encaminhamento de datagramas através de pequenos rótulos de tamanho fixo. Tais rótulos são números inteiros utilizados no protocolo MPLS e, através destes, a decisão de qual interface encaminhar o datagrama é tomada (ROSEN *et al*, 2001).

1.2 CONTEXTUALIZAÇÃO

O MPLS do Cisco IOS funde a inteligência do roteamento ao desempenho do switching e proporciona benefícios significativos para redes com arquitetura puramente IP e também para aquelas com IP e ATM ou um híbrido de outras tecnologias da Camada 2. A tecnologia MPLS é a chave das redes VPNs

dimensionáveis e QoS fim a fim, tornando possível a utilização eficiente das redes existentes para atender ao crescimento futuro e à rápida correção de falhas de enlaces e nós. A tecnologia também ajuda a fornecer serviços IP fim a fim altamente dimensionáveis e diferenciados com configuração, gerenciamento e provisionamento mais simples para provedores de Internet e assinantes (CISCO SYSTEMS, 2016).

O objetivo da QoS é fornecer serviço de rede melhor e mais previsível, fornecendo largura de banda dedicada, jitter controlado e latência, e perda de características melhoradas. QoS atinge esses objetivos, fornecendo ferramentas para gerenciar o congestionamento da rede, formação de rede tráfego, utilizando-se de maneira ampla, área de enlaces de forma mais eficiente, e definindo políticas de tráfego em toda a rede. QoS oferece serviços de rede inteligente que, quando corretamente aplicados, ajudam a fornecer desempenho consistente e previsível (CISCO SYSTEMS, 2006).

O conceito de QoS serve para mensurar a qualidade dos serviços oferecidos por uma rede de comunicações, ou seja, refletir o quanto ela é capaz de atender às expectativas de seus usuários através dos serviços que a mesma os oferecem. Esse conceito, inicialmente focado na rede, evoluiu para uma noção mais ampla, contemplando as múltiplas camadas da interação usuário-sistema.

No panorama atual das redes, de integração e convergência, onde as redes transportam todo tipo de informação, é primordial a utilização do QoS, para melhor atendimento às aplicações que requerem tratamento diferenciado.

1.3 PROBLEMA

O MPLS teve como fator decisivo para sua implantação, funcionalidades próprias (QoS e TE) que são complexas de serem realizadas em redes IP convencionais. A técnica resolveu problemas enfrentados por muitas redes como: escalabilidade (capacidade de um sistema em suportar um aumento de carga total quando os recursos são requeridos) e acompanhamento das políticas de QoS (reserva de banda, percentual de banda e definições de classes para variados tipos de dados).

Aliando a implementação de QoS à tecnologia MPLS, é possível priorizar, por exemplo, o tráfego de vídeo (indispensável a uma videoconferência). Configurando-se adequadamente é possível classificar o tráfego de acordo com o tipo (voz, vídeo, mensagens de texto, dados em geral), sem perda de pacotes.

Atualmente, no ambiente corporativo, ações de testes e simulação são indispensáveis para antecipar e prever possíveis problemas e empecilhos de projetos a serem implementados. Os testes e simulações não podem ser descartados de forma alguma. Para uma empresa, não se deve descartar a hipótese de falhas graves na aplicação definitiva de um projeto.

O problema a ser estudado situa-se nesse contexto, onde é claro e evidente que uma série de projetos causam transtornos pós-implantação para uma equipe, por falta de um maior planejamento, principalmente na hora de realizar testes e simulações. Para uma rede MPLS, que geralmente é aplicada em alto nível, detalhes importantes devem ser considerados e testados, justificando-se a estratégia de simulação de um modelo de rede.

Através de um emulador de sistemas operacionais (Oracle VM VirtualBox) e o simulador de rede *Graphical Network Simulator-3* (GNS3) – que estão presentes no projeto como ferramentas para auxiliar o entendimento do problema – é possível realizar uma simulação e assim, desenvolver questionamentos e soluções para os problemas, sem correr grandes riscos. O GNS3 tem a função de fazer a interligação dos equipamentos de rede com os hosts. Todos os componentes do projeto passam por configurações individuais para obter a comunicação esperada.

1.4 JUSTIFICATIVA

Ao aplicar uma técnica que prioriza a qualidade de serviço é possível focar nas necessidades da empresa, onde seus recursos serão otimizados de maneira que a informação mais utilizada esteja disponível a todos de maneira eficiente: ao considerar um serviço de videoconferência, não seria otimizado deixar metade dos recursos de comunicação destinadas a mensagens de texto ou a envio de grandes arquivos, em detrimento da qualidade de comunicação entre os participantes de uma

reunião. Desta maneira seria inviável ocorrer a reunião, pois não teria uma “banda” suficiente para tal tipo de tráfego, o desempenho da rede estaria comprometido.

Portanto, a justificativa está voltada ao incentivo e motivação na realização de testes e simulações, buscando reforçar a importância de tal prática em um ambiente corporativo e, ao mesmo tempo, apresentar um exemplo básico e típico de rede MPLS. A estratégia é a de possibilitar a atribuição de conhecimento e auxiliar na criação de modelos para implantações reais, que sejam úteis a empreendedores e funcionários que procuram uma solução para os problemas de tráfego e troca de dados na rede de suas empresas.

Tudo isso está acompanhado do objetivo de salientar e atender às necessidades de uma empresa, de forma que as demandas que surjam possam ser atendidas. O projeto atende a requisitos básicos de um bom projeto de rede, como: funcionalidade, escalabilidade, adaptabilidade, eficácia de custos e capacidade de gerenciamento.

1.5 OBJETIVOS

1.5.1 Objetivo Geral

Implementar parâmetros de QoS em uma rede MPLS com a utilização do Simulador GNS3.

1.5.2 Objetivos Específicos

- Simular uma rede *Wide Area Network* (WAN) que utilize MPLS;
- Simular tráfego de vídeo em rede MPLS;
- Aplicar protocolo *Open Shortest Path First* (OSPF);
- Aplicar técnicas de QoS;
- Aplicar políticas de tráfego e QoS;

- Aplicar protocolos MPLS em roteadores;
- Documentar principais comandos utilizados no simulador GNS3;
- Aplicar reserva de banda *Resource Reservation Protocol* (RSVP) para tráfego de vídeo;
- Aplicar prioridade para classe de vídeo.

1.6 PROCEDIMENTOS METODOLÓGICOS

O trabalho proposto será realizado através de pesquisas teóricas que servirão de fundamentação teórica e complemento aos estudos realizados durante o curso. Serão utilizados livros, apostilas e material de fabricantes de roteadores (CISCO), através de um levantamento bibliográfico. O trabalho terá etapas pré-definidas para sua realização (GIL, 1989):

- Pesquisa teórica e bibliográfica;
- Descrição das características do MPLS;
- Descrição das técnicas de QoS;
- O emulador de domínio público GNS3 será utilizado para simulação das aplicações MPLS, permitindo um melhor entendimento dos conceitos apresentados;
- Configuração e simulação de uma rede WAN que utilize MPLS;
- Apresentação e interpretação dos resultados obtidos através da simulação de uma rede MPLS com aplicação de técnicas de QoS.

2 FUNDAMENTAÇÃO TEÓRICA

A tecnologia MPLS é indicada para prover evolução, otimização e flexibilidade ao núcleo da rede que une vários enlaces atuais de alta velocidade, mostrando-se como uma tecnologia emergente a ser empregada nos provedores de acesso à Internet.

O principal protocolo de rede da Internet é o IP. O protocolo IP foi criado com o objetivo simples de tornar possível a comunicação entre máquinas, não dependendo do meio de transmissão utilizado e não possuindo mecanismos de notificação ou correção de erro. O protocolo IP não tem mecanismos que permitam realizar consultas de gerenciamento, é sem controle de fluxo, não orientado a conexão e não prevê qualidade de serviço (OLIVEIRA *et al*, 2012).

O MPLS é uma tecnologia aberta que foi apresentada inicialmente como uma solução que possibilitava melhorar o desempenho das redes IPs, combinando o processo de roteamento de nível 3 com a comutação de nível 2 para realizar o encaminhamento de datagramas através de pequenos rótulos de tamanho fixo. Tais rótulos são números utilizados pelo protocolo MPLS e, através destes, a decisão de qual interface encaminhar o datagrama é tomada (ROSEN *et al*, 2001). Segundo Rosen, a comutação de rótulos multiprotocolos combina a funcionalidade dos protocolos de roteamento da camada de rede e a comutação por rótulos, além de fornecer benefícios significativos às redes com IP e ATM, ou uma combinação de outras tecnologias no nível da camada de rede.

Em uma arquitetura IP sobre MPLS, as informações necessárias para o encaminhamento são obtidas do cabeçalho MPLS (32 *bits*), que é bem menor e menos complexo que o cabeçalho IP (20 *bytes*). Equipamentos de menor poder de processamento e armazenamento apresentam melhor desempenho nesse tipo de arquitetura em relação a outras arquiteturas.

Outra vantagem significativa da arquitetura IP sobre MPLS, que se destaca, diz respeito ao encaminhamento de datagramas ao longo de um caminho. O protocolo MPLS trabalha com encaminhamento dos pacotes baseado em rótulos, pois os roteadores de núcleo não têm acesso ao endereço IP de destino do pacote; assim, não há inteligência de roteamento nesses roteadores de núcleo, e sim o

encaminhamento local, de uma interface para outra, tomando como base os valores dos rótulos dos pacotes, ou seja, fazendo um processo apenas de comutação de rótulos. (OLIVEIRA *et al*, 2012).

2.1 CARACTERÍSTICAS DO MPLS

Nesta seção serão apresentadas as principais características da solução MPLS. Cada item possui embasamento teórico, e todos são relevantes no projeto em questão. Os conceitos descritos, a seguir, são importantes para a compreensão do MPLS e das configurações necessárias para se realizar a simulação e a implementação da rede.

2.1.1 O Cabeçalho MPLS

O item mais importante para o MPLS é o rótulo (DE GHEIN, 2007). O rótulo, conforme a Figura 1, é um identificador curto, de 4 *bytes*, e com significado local no roteador que é usado para identificar uma *Forwarding Equivalent Class* (FEC), isto é, um grupo de pacotes IPs que são enviados da mesma maneira, sobre o mesmo trajeto e com o mesmo tratamento de transmissão.

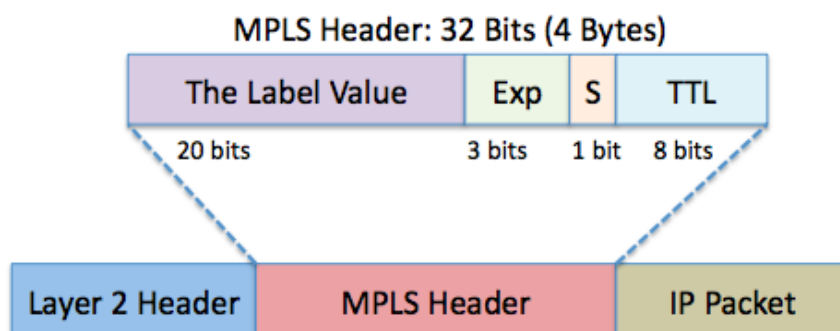


Figura 1 – Cabeçalho MPLS

Fonte: (MPLS, 2010)

2.1.2 Estrutura do MPLS

Por se tratar de uma ligação entre as camadas 2 e 3, já que utiliza a técnica de endereçamento dos protocolos de nível 3 e a técnica de comutação da camada 2, o MPLS pode ser considerado um protocolo de camada 2,5 (HARDENY, 2002). Assim, por ser uma camada de integração, é necessário que esta seja compatível com diversos protocolos da camada 3, assim como tecnologias de camada 2, o que justifica o “MultiProtocol” da sigla MPLS.

O MPLS é uma tecnologia utilizada em backbones. Embora a grande motivação de uso da tecnologia seja melhorar a velocidade de encaminhamento dos pacotes na rede, apenas este fator não seria um motivo suficiente para adoção da tecnologia, visto que a capacidade computacional existente nos equipamentos atuais responsáveis pelo roteamento é suficiente para um rápido atendimento ao tráfego (OLIVEIRA *et al*, 2012).

Atualmente, os algoritmos de encaminhamento de pacotes com alta velocidade são implementados no *hardware*, usando *Application Specific Integrated Circuits* (ASICs); portanto, uma pesquisa de rótulos de 20 *bits* não é significativamente mais rápida do que uma pesquisa IP de 32 *bits* (OSBORNE e SIMHA, 2002).

Um dos usos mais importantes do MPLS é facilitar a engenharia de tráfego nas redes IPs de provedores de serviços de telecomunicações. A principal capacidade que o MPLS traz às redes com engenharia de tráfego é a possibilidade de configurar um circuito virtual comutado para o modelo de roteamento da Internet (OLIVEIRA *et al*, 2012).

2.2 QoS COM MPLS

A Internet fornece um serviço apenas do tipo melhor esforço (*best-effort*), no qual todos os pacotes que trafegam são tratados de maneira uniforme para entregá-los ao destino. A Internet funciona com o protocolo IP, que trabalha com essa filosofia de melhor esforço, onde cada usuário da rede envia seus dados e

compartilha a largura de banda com todos os demais fluxos de dados dos outros usuários. Quando há um congestionamento, pacotes são descartados indiscriminadamente, não havendo garantia de que o serviço será realizado com sucesso, nem que haverá bom desempenho (ODOM e CAVANAUGH, 2004).

Aplicações em tempo real, tais como tráfego de voz, vídeo e multimídia, necessitam de garantia estrita de banda e são aplicações sensíveis a atraso (*delay*), variação do atraso dos pacotes (*jitter*) e perda de pacotes (LINS *et al*, 2011). De uma forma simples, QoS é a habilidade de diferenciar diversas classes de tráfegos com critérios predefinidos e designar prioridades baseadas em vários tipos de tráfegos que afetam o tratamento em cada roteador na rede (LOBO, 2008).

Os tráfegos analisados de forma simulada e as marcações utilizadas na simulação do ambiente QoS são apresentados no quadro 1.

Classe	Perfil do Tráfego	Marcação em DSCP	Marcação em EXP	Limite de Banda
Premium	Mínima garantia de banda, baixo <i>delay</i> , baixo <i>jitter</i> e sem perda de pacotes.	EF	EXP 5 - Dentro do perfil. Descarte - Fora do perfil.	128 Kbps
Business	Mínima garantia de banda e baixa perda de pacotes.	AF 21, AF 22 ou AF23	EXP 2 - Dentro do perfil. EXP 1 - Fora do perfil.	512 Kbps
Best-Effort	Nenhuma garantia.	0		

Quadro 1 – Classes utilizadas na simulação do ambiente QoS

Fonte: (OLIVEIRA *et al*, 2012)

Os parâmetros típicos para a qualidade de serviço da camada de transporte são resumidos em: retardo no estabelecimento da conexão; probabilidade de falha no estabelecimento da conexão; *throughput*, taxa de erros residuais; proteção; prioridade; resiliência.

O parâmetro *throughput* calcula o número de bytes de dados do usuário transmitidos por segundo durante um determinado intervalo de tempo. O *throughput* é medido separadamente para cada direção. O parâmetro de prioridade oferece ao usuário de transporte um modo de indicar que algumas conexões são mais importantes do que outras e, em caso de congestionamento, garantir que as conexões de maior prioridade sejam atendidas primeiro.

Os parâmetros QoS são especificados pelo usuário de transporte quando uma conexão é solicitada. Com isso, uma QoS pode ser descrita como um conjunto de parâmetros que descrevem a qualidade (por exemplo, largura de banda, utilização de *buffers*, prioridades, utilização da *Central Processing Unit* (CPU), etc.) de um fluxo de dados específico.

Somente o aumento na largura de banda não é suficiente para garantir a qualidade do serviço à aplicação, pois em se tratando de redes compartilhadas por múltiplos usuários e muitas vezes, redes de longas distâncias, podem haver congestionamentos, provocando atrasos inadmissíveis em certas aplicações sensíveis, como por exemplo voz e videoconferência (BRUN *et al*, 2002).

2.3 ARQUITETURA DE QoS

O IETF define dois modelos para implementação de QoS numa rede IP (SVERZUT, 2008): A arquitetura de Serviços Integrados (*IntServ*) e a arquitetura de Serviços Diferenciados (*DiffServ*).

2.3.1 Serviços Integrados (*IntServ*)

O objetivo desta arquitetura é obter a largura de banda e a latência necessárias para uma determinada aplicação (DAVIDSON *et al*, 2007). É tipicamente utilizado para garantir que um fluxo em especial receba o nível de QoS apropriado ao longo da rede inteira antes de enviar esse tráfego. A arquitetura de Serviços Integrados baseia-se em quatro componentes, conforme a Figura 2.

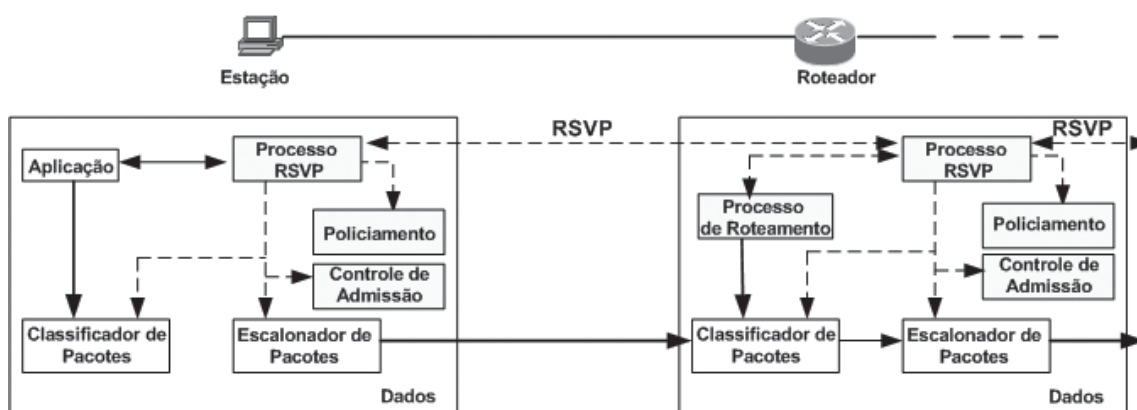


Figura 2 – Componentes da arquitetura *IntServ*

Fonte: (OLIVEIRA *et al*, 2012)

- **Escalonador de pacotes:** gerencia o *buffer* das filas de saída dos roteadores usando alguma política de atendimento.
- **Controle de admissão:** implementa o algoritmo utilizado pelo roteador para determinar se a solicitação de QoS de um novo fluxo pode ser atendida sem interferir nas garantias de outros fluxos já alocados.
- **Classificador de pacotes:** reconhece os fluxos segundo suas identificações, mapeia os pacotes desses fluxos nas diferentes categorias de serviço, notifica a função de policiamento e, caso os pacotes estejam em conformidade com o controle imposto pelo policiamento, os coloca nos *buffers* das filas de saída apropriadas.
- **Policiamento:** verifica se o fluxo está de acordo com as especificações negociadas na fase de estabelecimento da conexão. Fluxos fora do acordo podem ter seus pacotes descartados para evitar congestionamentos. A grande vantagem do Serviço Integrado é que previamente é feita uma alocação de banda, já que cada roteador é consultado ao longo do caminho para fazer essa reserva, garantindo assim a entrega, caso a reserva seja aceita por todos.

2.3.2 Serviços Diferenciados (*DiffServ*)

A arquitetura de *DiffServ* foi introduzida como uma alternativa para a arquitetura de *IntServ*, evitando problemas de escalabilidade e complexidade. A qualidade de serviço na arquitetura *DiffServ* é garantida através de mecanismos de

priorização de pacotes na rede, diferentemente da arquitetura *IntServ*, onde a qualidade de serviço é garantida através de reserva de recursos na rede.

Na Figura 3 são exibidos os principais blocos funcionais em um roteador utilizando a arquitetura *DiffServ*. Todas as funções desse diagrama estão presentes nos roteadores de borda da rede e, eventualmente, adicionados aos roteadores de núcleo da rede.

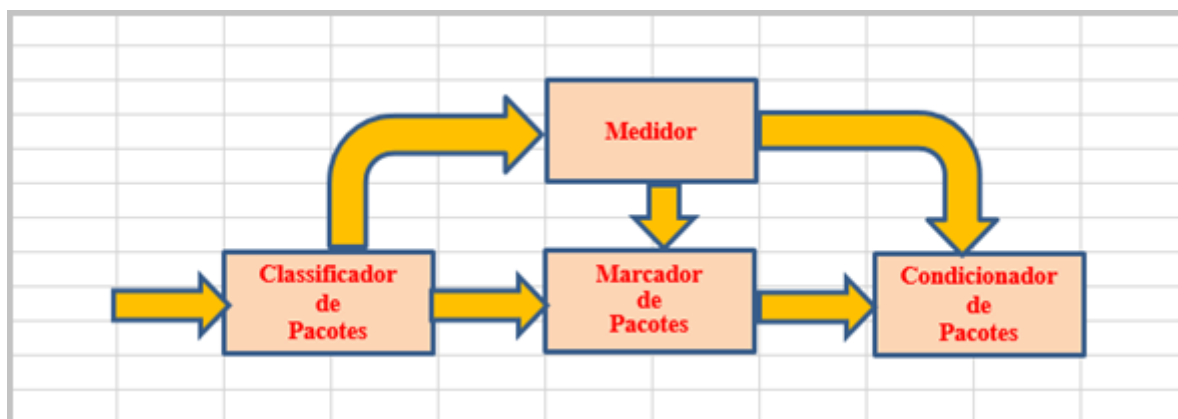


Figura 3 – Blocos funcionais do *DiffServ*

Fonte: (OLIVEIRA *et al*, 2012)

- **Classificador de pacotes:** identifica pacotes que são mapeados para classes.
- **Medidor:** verifica conformidade com parâmetros de tráfego e passa o resultado para o marcador e o condicionador de pacotes, para disparar uma ação específica para pacotes que estão fora ou dentro do perfil definido.
- **Marcador de pacotes:** escreve/sobrescreve o valor do *Differentiated Services Code Point* (DSCP).
- **Condicionador de pacotes:** atrasa alguns pacotes para que eles permaneçam em conformidade com o perfil definido, ou descarta pacotes que excederam o perfil definido.

Os serviços diferenciados são implementados com base na definição de tipos de serviços. Através do campo *Type of Service* (ToS) do cabeçalho IP pode-se representar o tipo de serviço. Os Serviços Diferenciados fazem uso dos 6 *bits* mais significativos do campo ToS, chamado de campo *Differentiated Service* (DS) pelo serviço diferenciado.

Na solução *DiffServ* os pacotes são classificados, marcados e processados segundo a codificação rotulada no cabeçalho do pacote DSCP. Pode-se separar as funções entre os equipamentos de borda e núcleo, conforme Figura 4.

Funções dos equipamentos de borda:

- Examinar os pacotes que chegam e classificá-los de acordo com a política em vigor;
- Marcar os pacotes com um DSCP que reflita o nível de serviço desejado;
- Garantir que o tráfego dos clientes siga as especificações definidas através de policiamento e da conformidade.

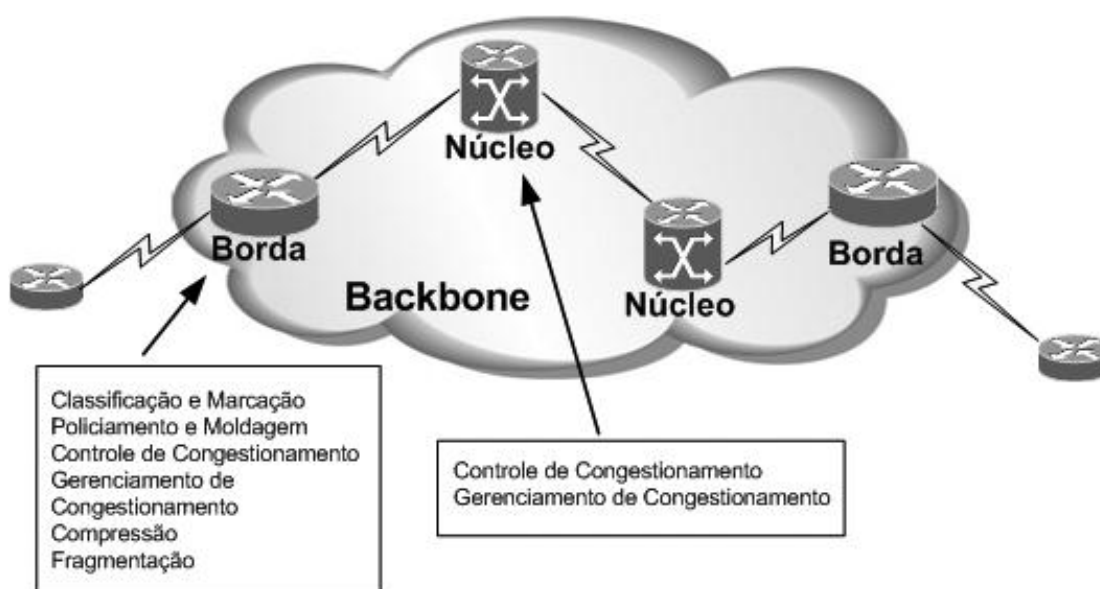


Figura 4 – Arquitetura de serviços diferenciados

Fonte: (OLIVEIRA *et al*, 2012)

Funções dos roteadores de núcleo:

- Examinar os pacotes que chegam e verificar o DSCP marcado em seus cabeçalhos;
- Classificar e encaminhar os pacotes que chegam de acordo com seus DSCPs.

2.4 VALORES DE QoS

Os quadros 2 e 3 trazem as definições teóricas para o cálculo dos valores de QoS. O quadro 2 apresenta os parâmetros, citando as aplicações e dividindo-as de acordo com as classes e os encaminhamentos. Já o quadro 3 consiste em trazer uma legenda desses parâmetros. Também é possível verificar a formação e a descrição dos campos de ToS nos quadros 4 e 5.

Aplicação	Classe = CoS = IPP	AF	DSCP	ToS	ToS (Hexad.)	DP	8° bit	7° bit	6° bit	5° bit	4° bit	3° bit	2° bit	1° bit
Best Effort	0	0	0	0	0		0	0	0	0	0	0	0	0
Scavenger	1	CS1	8	32	20		0	0	1	0	0	0	0	0
Bulk Data	1	AF11	10	40	28	Baixa	0	0	1	0	1	0	0	0
	1	AF12	12	48	30	Média	0	0	1	1	0	0	0	0
	1	AF13	14	56	38	Alta	0	0	1	1	1	0	0	0
Network Mgmt.	2	CS2	16	64	40		0	1	0	0	0	0	0	0
	2	AF21	18	72	48	Baixa	0	1	0	0	1	0	0	0
Transaction Data	2	AF22	20	80	50	Média	0	1	0	1	0	0	0	0
	2	AF23	22	88	58	Alta	0	1	0	1	1	0	0	0
	3	CS3	24	96	60		0	1	1	0	0	0	0	0
Mission-Critical	3	AF31	26	104	68	Baixa	0	1	1	0	1	0	0	0
Video Streaming	3	AF32	28	112	70	Média	0	1	1	1	0	0	0	0
	3	AF33	30	120	78	Alta	0	1	1	1	1	0	0	0
	4	CS4	32	128	80		1	0	0	0	0	0	0	0
Video Interativo	4	AF41	34	136	88	Baixa	1	0	0	0	1	0	0	0
	4	AF42	36	144	90	Média	1	0	0	1	0	0	0	0
	4	AF43	38	152	98	Alta	1	0	0	1	1	0	0	0
Voz	5	CS5	40	160	A0		1	0	1	0	0	0	0	0
	5	EF	46	184	B8		1	0	1	1	1	0	0	0
Routing	6	CS6	48	192	C0		1	1	0	0	0	0	0	0
	7	CS7	56	224	E0		1	1	1	0	0	0	0	0

Quadro 2 – Valores de QoS

Fonte: (NETCONTRACTOR, 2011)

LEGENDA :	
CoS	Classe de Serviço
DSCP	Ponto de Código de Serviços Diferenciados
ToS	Tipo de Serviço
AF	Encaminhamento Assegurado
IPP	Precedência IP
CS	Seletor de Classe
DP	Probabilidade de Droped
ECN	Notificação de Congestionamento Explícito

Quadro 3 – Legenda dos parâmetros de QoS

Fonte: (NETCONTRACTOR, 2011)

ToS (8° ao 1° bit)								
DSCP (8° ao 3° bit)						ECN (2° ao 1° bit)		
AF (CLASSE - 8° ao 6° bit, PROB. DE QUEDA- 5° ao 3° bit)								
IPP = CS (8° ao 6° bit)			DP (5° ao 3° bit)					
			Delay (5°)	Throughput (4°)	Reliability(3°)			
	8° bit	7° bit	6° bit	5° bit	4° bit	3° bit	2° bit	1° bit
ToS	128	64	32	16	8	4	2	1
DSCP	32	16	8	4	2	1		
CoS = IPP	4	2	1					

Quadro 4 – Formação do quadro de ToS

Fonte: (NETCONTRACTOR, 2011)

Campos (Fields)	8	7	6	5	4	3	2	1
Type of Service (ToS)	X	X	X	X	X	X	X	0
Bits	Fields							
8 ao 3	DSCP 2°6 (n° de bits) = 64 (valor máximo)							
2 ao 1	ECN (Explicit Congestion Notification)							
Bits	Fields							
IP Precedence								
8° Bit ao 6° Bit	111 = Network Control							
	110 = InternetWork Control							
	101 = Critic							
	100 = Flash Override							
	011 = Flash							
	010 = Immediate							
	001 = Priority							
000 = Routine								
5° Bit	1 = Low Delay				0 = Normal Delay			
4° Bit	1 = High Throughput				0 = Normal Throughput			
3° Bit	1 = High Reliability				0 = Normal Reliability			
2° Bit	1 = Minimizar Custo (RFC 1349)							
1° Bit	0							

Quadro 5 – Descrição do quadro de ToS

Fonte: Autoria Própria

2.4.1 Legenda dos Parâmetros de QoS

- **Classe de Serviço (CoS):** Classe de serviço é um parâmetro usado em dados de voz e protocolos para diferenciar os tipos de cargas contidas no pacote a ser transmitido. O objetivo de tal diferenciação é geralmente associado com a atribuição de prioridades para a carga de dados (JUNIOR, 2015).

- **Ponto de Código de Serviços Diferenciados (DSCP):** o DSCP pode ser definido com um valor desejado na extremidade da rede para que os dispositivos centrais classifiquem mais facilmente o pacote (CISCO SYSTEMS, 2008).
- **Tipo de Serviço (ToS):** o campo ToS tem como função básica permitir que os pacotes sejam tratados de modo diferente, com base nas necessidades da aplicação. Este campo é utilizado pelos roteadores para determinar como o datagrama deve ser tratado, podendo diferenciar os vários tipos de datagramas (OLIVEIRA *et al*, 2012).
- **Encaminhamento Assegurado (AF):** este tipo de serviço fornece uma expectativa de serviço que será obtida por um determinado tráfego quando existem momentos de congestionamento. O serviço AF assegura que os pacotes marcados como em conformidade serão entregues com alta probabilidade (MAGALHÃES e GUARDIEIRO, 2002).
- **Precedência IP (IPP):** O IP *precedence* é o CS incluindo o BE (DSCP default). O IP *precedence* possui oito nomenclaturas, cada uma correspondente ao seu valor (AMARAL, 2008).
- **Probabilidade de Drop (DP):** é um parâmetro que atribui três probabilidades de queda (baixa, média e alta) para cada classe AF (1 a 4) (FRAHIM *et al*, 2004).
- **Notificação de Congestionamento Explícito (ECN):** o recurso ECN sinaliza que há um certo congestionamento na rede e que é necessária a desaceleração do envio de pacotes (SURYANTOFANG, 2013).

2.5 MECANISMOS DE QoS

Vários mecanismos de QoS são utilizados para provimento da qualidade de serviço em redes de computadores. Estes mecanismos compreendem: classificação e marcação, gerenciamento de congestionamento, policiamento, moldagem e evitar congestionamento. O processo de QoS é apresentado na Figura 5.

2.5.1 Classificação e Marcação

A classificação fornece um serviço preferencial a um determinado tipo de tráfego. Sua principal função é separar/classificar os fluxos em classes de serviços (JUNIOR, 2015). Já a marcação, dependendo do tipo de interface em questão e das características do equipamento, pode ser feita na camada 2 (802.1p/q), na camada 3 (IPP ou DSCP) ou no rótulo MPLS (Campo EXP).

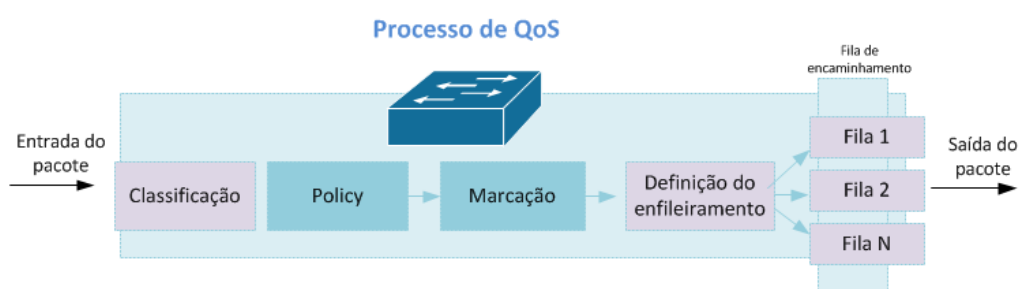


Figura 5 – Processo de QoS

Fonte: (DIAS, 2013)

Todas as políticas de classificação e marcação: gerenciamento de congestionamento, evitar congestionamento, policiamento e moldagem podem ser utilizadas no backbone MPLS.

2.5.2 Policiamento

Para disciplinar o tráfego que entra no backbone, usa-se o mecanismo de policiamento, que limita a banda que uma determinada classe de tráfego pode utilizar. Para os excessos gerados de banda, podem ser definidas as ações de descartar os pacotes, o que efetivamente impossibilita o tráfego acima da banda disponibilizada, ou pode-se remarcar esse tráfego com uma prioridade menor antes de transmiti-lo, o que não interrompe o serviço, porém o penaliza com um maior tempo de resposta às aplicações que não estão com o comportamento dentro do perfil contratado.

2.5.3 Engenharia de Tráfego com MPLS

O protocolo MPLS oferece a capacidade de roteamento baseado na origem, que é conhecido na rede MPLS como roteamento explícito. Apesar de o protocolo IP possuir a característica de roteamento baseado na origem, o mesmo não é apropriado por diversos motivos, incluindo o fato de que apenas um número limitado de saltos pode ser especificado, e que normalmente ele é processado fora do “melhor caminho” na maioria dos roteadores.

Uma das aplicações do roteamento explícito é na TE, onde o objetivo é apresentar, como é possível garantir que os diversos caminhos possam ser utilizados para o envio do tráfego sem sobrecarregar um determinado caminho, deixando o outro subutilizado.

Um dos maiores problemas numa rede é que as rotas preferidas tendem a convergir pelos caminhos de maior banda. Tal decisão causa o desperdício de recursos de forma a haver muito tráfego em poucos enlaces, enquanto outros enlaces permanecem ociosos. Uma premissa importante da TE é distribuir o tráfego por meio dos enlaces disponíveis para garantir que a carga seja dividida segundo critérios. Um fato importante é que a TE está utilizando cada vez mais o MPLS.

O MPLS pode ser utilizado para criar túneis de TE com base na análise do tráfego e com objetivo de fornecer balanceamento de carga entre caminhos de diferentes taxas de transmissão. Na TE é determinado o caminho por meio da rede que diversos fluxos de dados seguirão e também as operadoras de telecomunicações podem oferecer um backbone, para os seus clientes, com mais eficiência (OLIVEIRA *et al*, 2012).

2.6 PROTOCOLOS

É importante citar sobre dois protocolos utilizados no projeto: o OSPF e o RSVP. O OSPF é um protocolo de roteamento do tipo estado de enlace, que envia anúncios sobre o estado da conexão a todos os outros roteadores em uma mesma

área hierárquica e usa o algoritmo *Shortest Path First* (SPF) para calcular o caminho mais curto para cada nó.

Uma das principais características desse protocolo é que não há limite no custo máximo de uma rota e, além disso, o OSPF pode efetuar o balanceamento de carga. O “cálculo” do OSPF seleciona o caminho de menor custo para uma rede, da origem ao destino, usando apenas os enlaces ativos.

No protocolo de roteamento por estado de enlace, cada roteador possui uma visão completa da rede, fornecida pelas informações de todos os roteadores na rede, mas os roteadores precisam construir uma tabela de roteamento “do zero”, usando apenas a informação do caminho mais curto (OLIVEIRA *et al*, 2012).

O outro protocolo utilizado, o RSVP, é o protocolo apropriado para distribuição de rótulos em redes MPLS. Esse protocolo é adequado para extensão ao mundo MPLS porque lida com reservas de recursos fim-a-fim para fluxos de tráfego. É muito semelhante com o MPLS com TE. Por outro lado, ele não atende a todas as exigências necessárias para o MPLS – principalmente quanto a distribuição de rótulo e controle de caminhos por meio de rotas explícitas.

O RSVP foi criado como um protocolo de sinalização para que aplicações fossem capazes de reservar recursos, ou seja, é um protocolo usado por uma aplicação para informar à rede seus requisitos de QoS e efetuar a reserva de recursos ao longo do caminho que o pacote irá percorrer (OLIVEIRA *et al*, 2012).

3 DESENVOLVIMENTO DO TEMA

3.1 IMPLEMENTANDO A REDE MPLS

É importante considerar que toda a estrutura física e lógica do projeto de rede desenvolvido depende da utilização de vários softwares e ferramentas que, em conjunto, permitem o funcionamento esperado da simulação de uma rede MPLS. Alguns pré-requisitos são necessários para que seja possível a execução e também a compreensão da simulação de uma rede MPLS.

Neste capítulo serão descritos os principais softwares, ferramentas e condições necessárias para desenvolver essa rede. Ao considerar as especificidades de cada software, na construção da solução da rede, compreende-se que a maioria dos programas e algumas configurações a serem apresentados também podem servir para outros tipos de simulação que podem ou não envolver MPLS.

3.1.1 A Ferramenta de Simulação do Ambiente

Este trabalho utiliza um ambiente de simulação para os testes através do software emulador GNS3. O software GNS3 é o emulador utilizado neste trabalho, pois atende a todos os requisitos necessários para emulação das funcionalidades que serão exibidas. Trata-se de um emulador gráfico que é fortemente utilizado com o *Dynamips* (um emulador Cisco IOS) e o *Dynagem* (um *front-end* para *Dynamips* baseado em texto) (OLIVEIRA *et al*, 2012).

O emulador GNS3 permite a reprodução fiel das características de diversos modelos de roteadores, sobretudo do fabricante Cisco Systems, possibilitando a criação de diversos cenários, com plataformas de roteadores 1700, 2600, 3600, 3700 e 7200. O GNS3 também permite a adição virtual de alguns módulos disponíveis para cada plataforma de roteador.

Uma das grandes vantagens em adotar um software de emulação como o GNS3 é permitir a interconexão do ambiente virtual com um ambiente real, além da possibilidade de interação com um ambiente idêntico ao proporcionado por elementos de rede reais. Devido ao processo ser bastante intenso, o desempenho de um roteador emulado jamais será igual ao de um roteador real, sendo esta uma das limitações identificadas no uso desse emulador.

GNS3 é um simulador de domínio público que permite simular redes complexas. Ferramenta que auxilia de maneira complementar as práticas de laboratórios, ou seja, é possível ensaiar um cenário real, dando oportunidade ideal para validar produtos. Utilizado para simular aplicações em MPLS facilitando a compreensão de conceitos.

Na figura 6 pode-se visualizar uma janela do GNS3 que traz o número da versão utilizada no projeto.



Figura 6 – Software GNS3 (Versão)

Fonte: Autoria Própria

Na figura 7 é possível analisar a arquitetura física do projeto de rede em questão, com a identificação dos endereços IPs das interfaces.

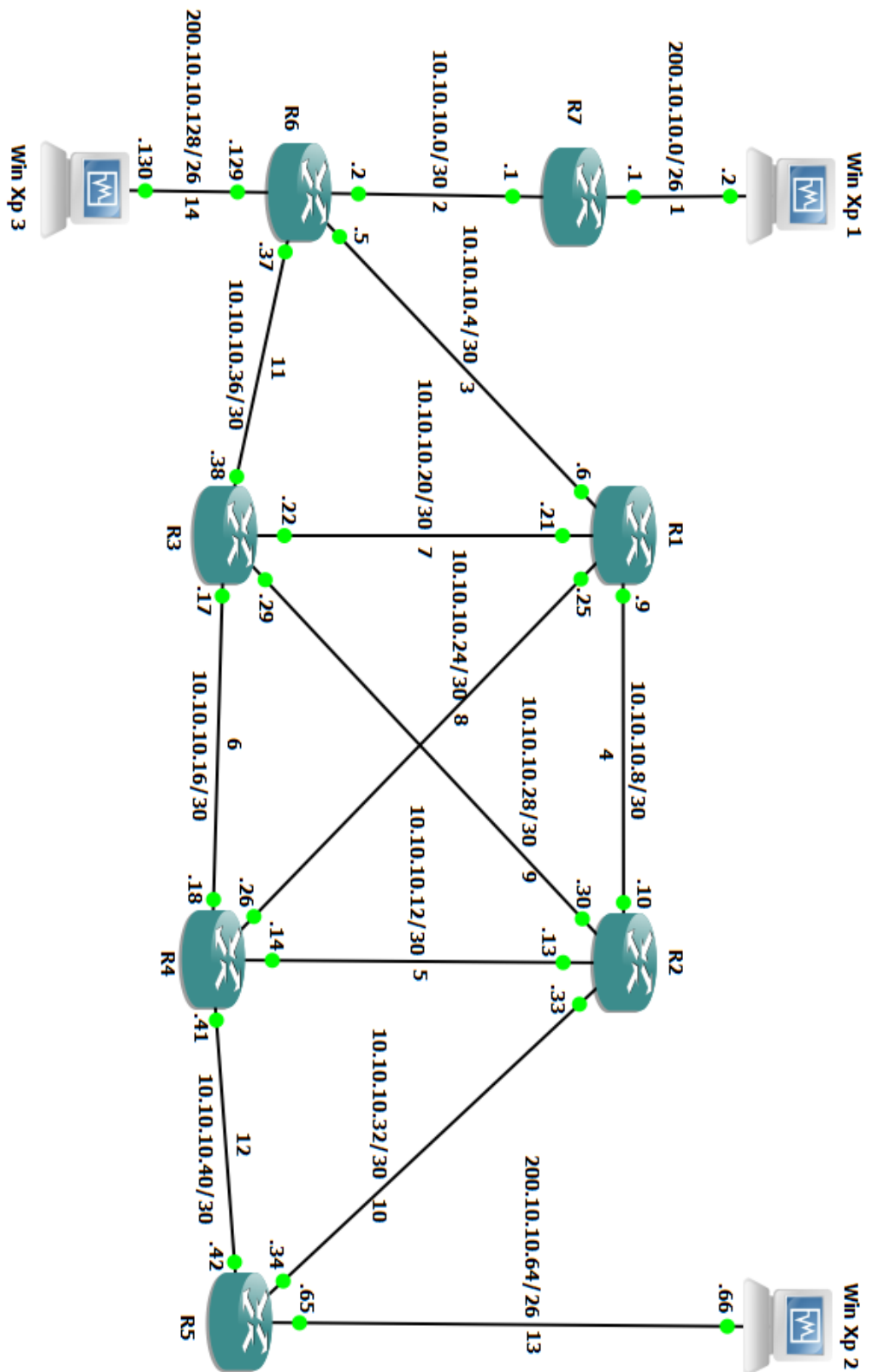


Figura 7 – Rede MPLS simulada no GNS3

Todo a rede necessita de uma determinação de sub-redes na configuração de cada equipamento. No quadro 6 seguem as sub-redes utilizadas no projeto.

Nº Identificador	Rede	Máscara	CIDR
1	200.10.10.0	255.255.255.192	/26
2	10.10.10.0	255.255.255.252	/30
3	10.10.10.4	255.255.255.252	/30
4	10.10.10.8	255.255.255.252	/30
5	10.10.10.12	255.255.255.252	/30
6	10.10.10.16	255.255.255.252	/30
7	10.10.10.20	255.255.255.252	/30
8	10.10.10.24	255.255.255.252	/30
9	10.10.10.28	255.255.255.252	/30
10	10.10.10.32	255.255.255.252	/30
11	10.10.10.36	255.255.255.252	/30
12	10.10.10.40	255.255.255.252	/30
13	200.10.10.64	255.255.255.192	/26
14	200.10.10.128	255.255.255.192	/26

Quadro 6 – Sub-redes utilizadas no projeto

Fonte: Autoria Própria

3.1.2 Emulação de Máquinas Virtuais

A rede simulada no GNS3 depende de imagens de roteadores com suas respectivas plataformas, já citadas anteriormente. Nesse projeto foi utilizado somente o roteador 7200 da Cisco. Além disso, também é essencial a existência de hosts virtualizados para poder verificar o funcionamento do MPLS e das outras configurações de toda a rede.

Neste caso, utiliza-se o software de virtualização Oracle VM VirtualBox, que permite ao computador executar vários sistemas operacionais ao mesmo tempo. O sistema operacional utilizado na emulação das máquinas virtuais foi o Windows XP, o qual teve bom desempenho e exige menos recursos do computador utilizado para a simulação, podendo ser instalado em máquinas mais simples (com processador e memória *Random Access Memory* (RAM) inferiores).

Na figura 8 pode-se visualizar as informações básicas do VirtualBox, que contém a versão do software.

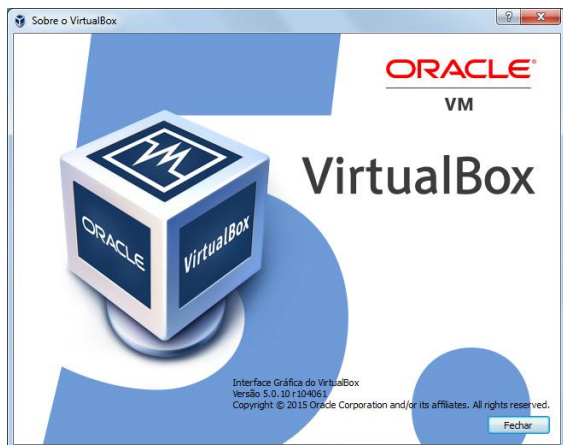


Figura 8 – Software VirtualBox (Versão)

Fonte: Autoria Própria

No VirtualBox, cada sistema operacional emulado deve ser instalado da mesma maneira que a instalação normal de qualquer sistema operacional (inclusive a ativação do Windows). Caso exista uma *appliance* anteriormente instalada e exportada, basta importá-la. Na figura 9 há um exemplo de *appliance* de um sistema operacional para ser importada no VirtualBox. A figura 10 apresenta a interface gráfica do gerenciador do VirtualBox, com as máquinas virtuais importadas, instaladas e em execução.

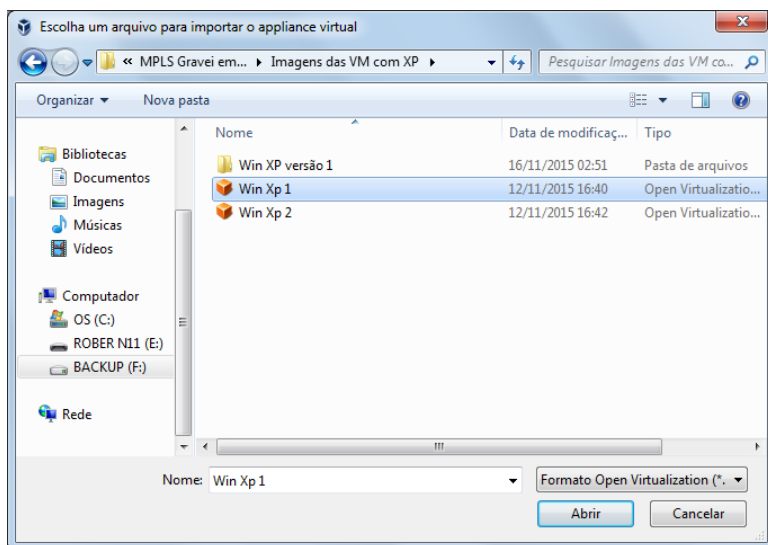


Figura 9 – Exemplo de *appliance* de um Sistema Operacional

Fonte: Autoria Própria

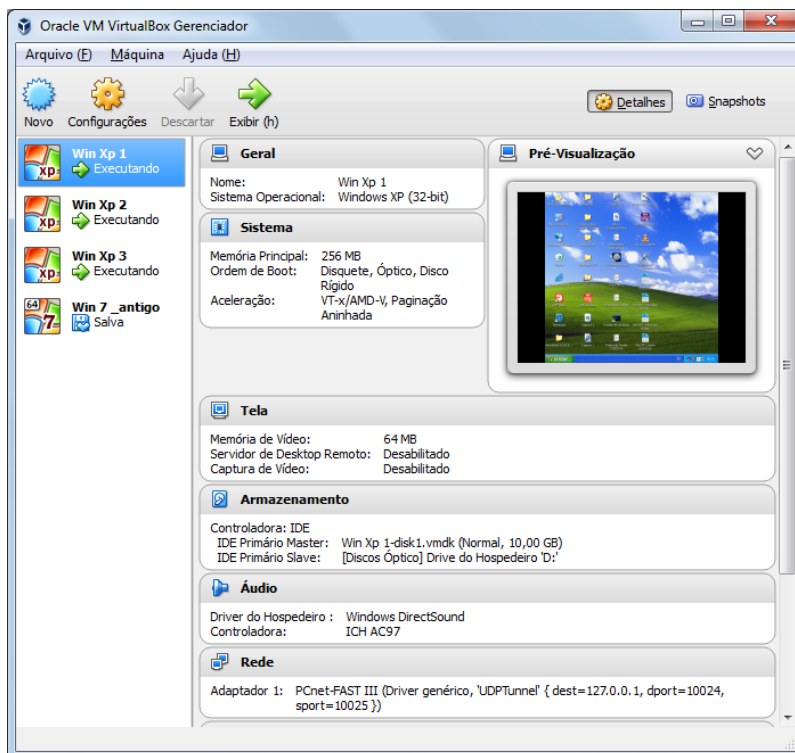


Figura 10 – Gerenciador do VirtualBox

Fonte: Autoria Própria

Outro fator importante, que contribui na decisão da escolha de um sistema operacional para rodar uma rede MPLS, simulada no GNS3, foram os recursos de equipamentos disponíveis para o desenvolvimento do projeto. Nesse caso, trata-se dos parâmetros de sistema da máquina utilizada para a instalação de todos os softwares necessários.

O projeto foi desenvolvido a partir de um notebook da marca Dell, modelo Inspiron n4050, com as seguintes características técnicas:

- Processador: Intel Core i5-2450M;
- CPU: 2,50 GHz
- Memória instalada (RAM): 4 GB;
- Tipo de sistema operacional: 64 bits;
- Sistemas operacional: Windows 7 Home Basic (Service Pack 1).

Na figura 11 estão descritas as informações básicas do computador da máquina real, com Windows 7. O mesmo acontece na figura 12, só que referente às informações básicas da máquina virtual, com Windows XP.

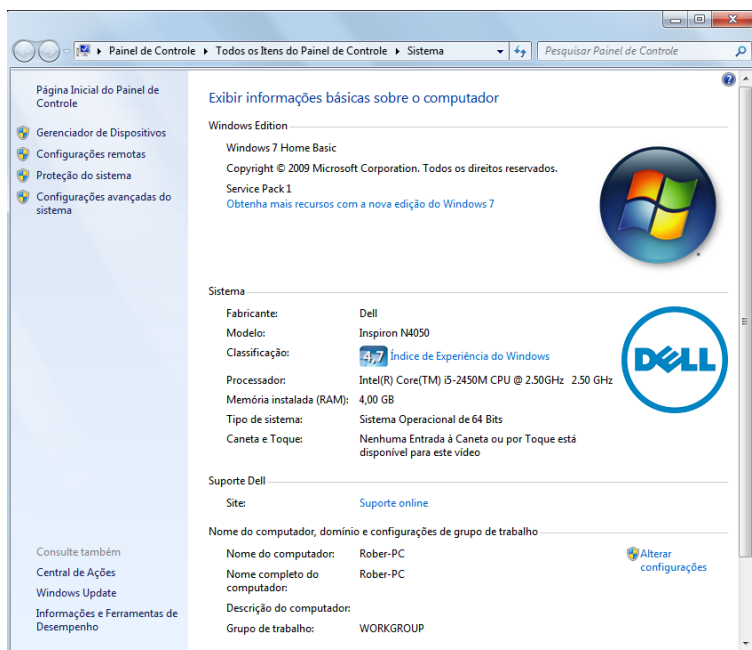


Figura 11 – Informações básicas do Sistema Operacional da máquina real

Fonte: Autoria Própria

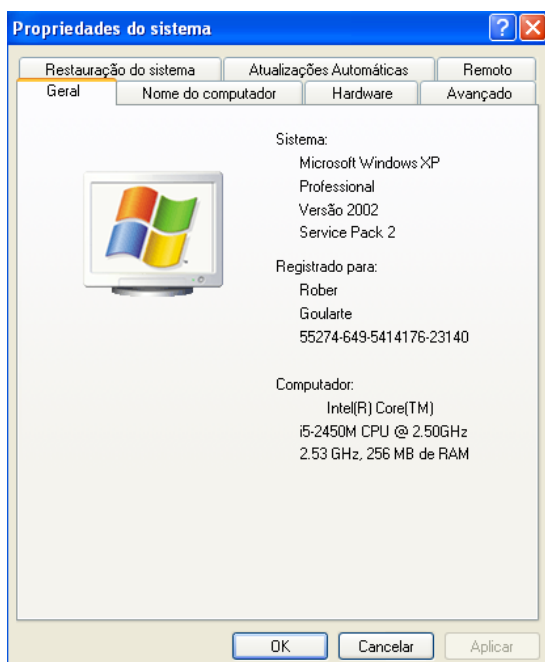


Figura 12 – Informações básicas do Sistema Operacional da máquina virtual

Fonte: Autoria Própria

3.1.3 Configuração de Softwares

Para iniciar este tópico é importante considerar que a base inicial para o desenvolvimento da rede esteja clara e estabelecida. Compreendendo então que os pré-requisitos das máquinas sejam atendidos de forma equivalente ou bem próximos do que foi citado. Levando-se em consideração a disponibilidade de acesso dos softwares GNS3 e VirtualBox, pode-se começar a configuração dos softwares para prosseguir com os próximos passos para a implementação definitiva da rede simulada.

A recomendação é que os procedimentos de configuração se iniciem pelo VirtualBox, pois é importante que as máquinas virtuais estejam corretamente instaladas para que, posteriormente, o GNS3 consiga encontrar um caminho que possibilite a importação. O único procedimento a ser realizado é ajustar o primeiro adaptador de rede de cada máquina virtual. Na figura 13 há um exemplo de como deve ficar a configuração de rede.

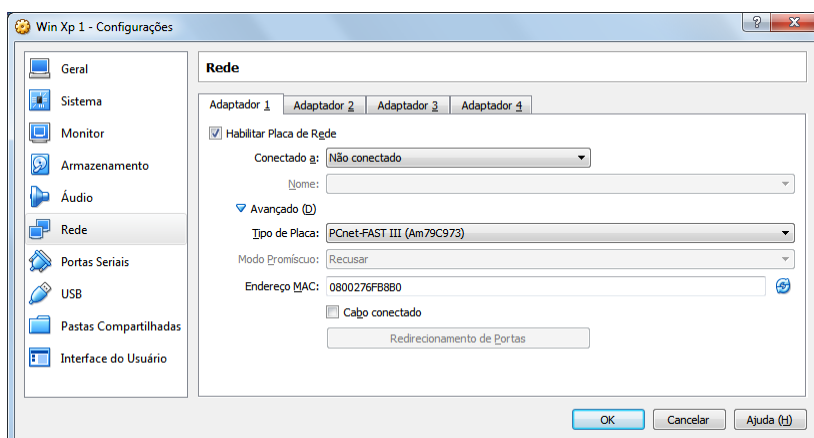


Figura 13 – Configuração do adaptador de rede da máquina virtual

Fonte: Autoria Própria

Com as máquinas virtuais instaladas e com a configuração de rede realizada nas mesmas, pode-se iniciar no software GNS3 a configuração da rede MPLS. Inicialmente monta-se a estrutura física da rede. Para isso deve-se importar as máquinas virtuais do VirtualBox e também as imagens dos roteadores que, neste caso, são da Cisco.

Esses processos de importação devem ser feitos nas preferências do GNS3. A figura 14 mostra a janela de preferências do GNS3 e a funcionalidade “VirtualBox VMs” para adicionar as máquinas virtuais. Na mesma janela há a funcionalidade “IOS routers” para adicionar as imagens dos roteadores.

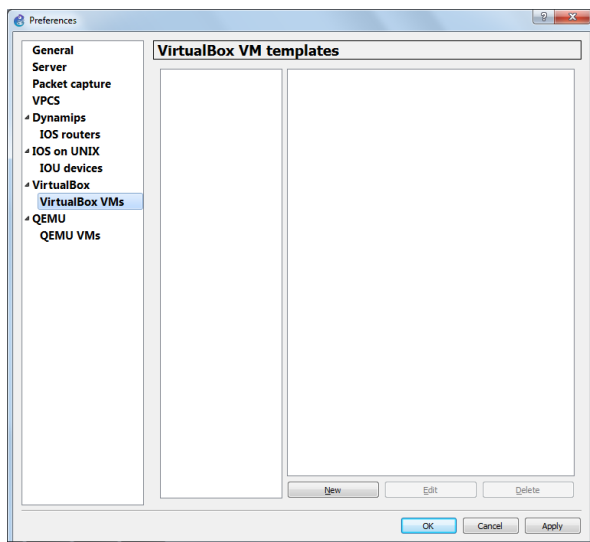


Figura 14 – Janela de preferências do GNS3

Fonte: Autoria Própria

Nas figuras 15 e 16 seguem, respectivamente, as janelas de seleção de uma máquina virtual e da máquina virtual já importada.

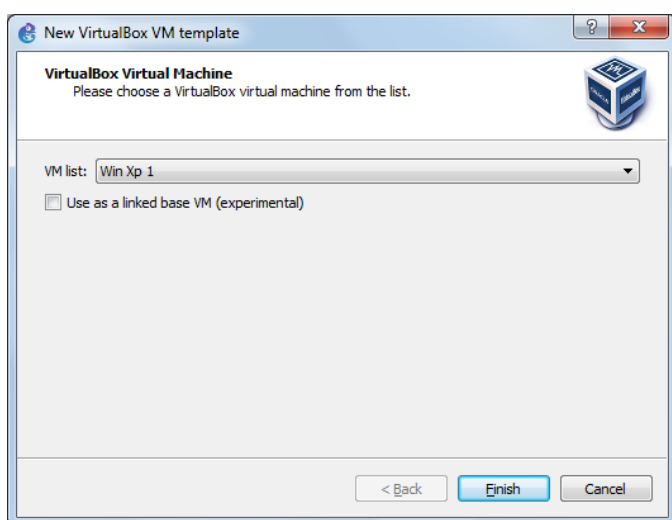


Figura 15 – Janela de seleção de máquinas virtuais

Fonte: Autoria Própria

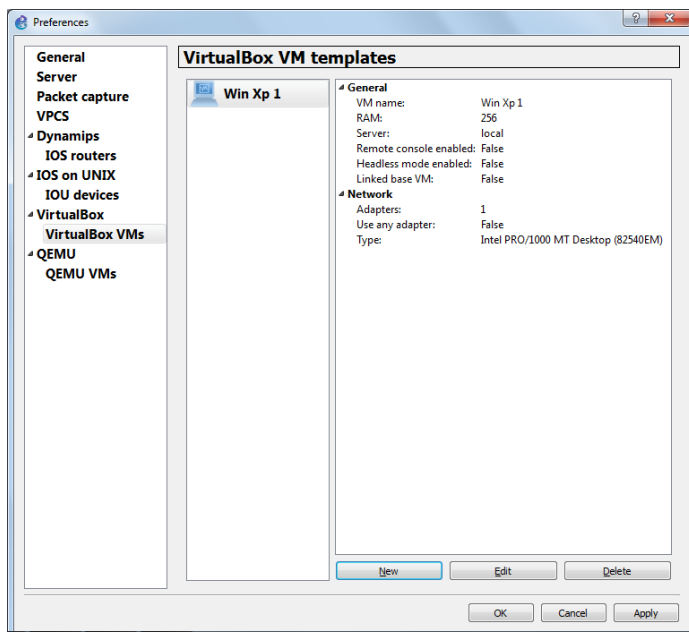


Figura 16 – Janela com uma máquina virtual importada

Fonte: Autoria Própria

Na figura 17 segue a janela de seleção de uma imagem de roteador. No caso desse projeto foi utilizada somente a imagem “*c7200-advipservicesk9-mz150-1M.image*”, que corresponde ao roteador Cisco 7200.

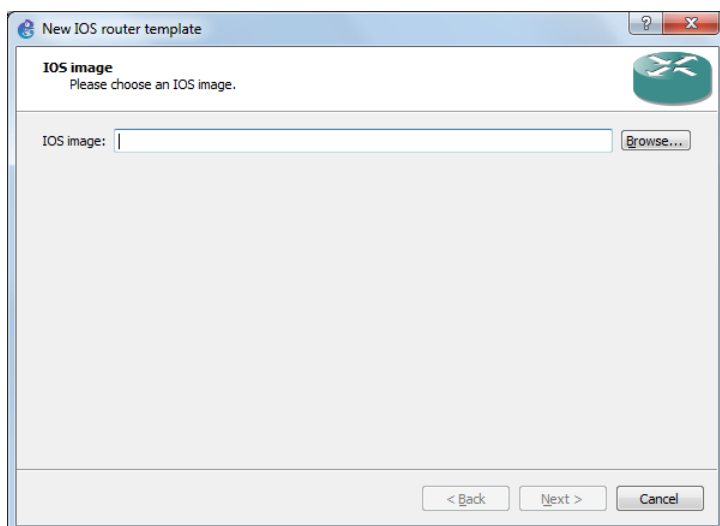


Figura 17 – Janela de seleção de uma imagem de roteador

Fonte: Autoria Própria

Após escolher a imagem do roteador é necessário realizar mais quatro processos, que consistem na escolha do nome e da plataforma, da memória, dos

adaptadores de rede e de um valor chamado *Idle-PC*, o qual é essencial para que o roteador não consuma 100% do uso da CPU e haja indisponibilidade de recursos.

Nas figuras 18 e 19 seguem os exemplos a serem seguidos na escolha do nome e da plataforma, assim como o espaço de memória que deve ser alocado ao roteador.

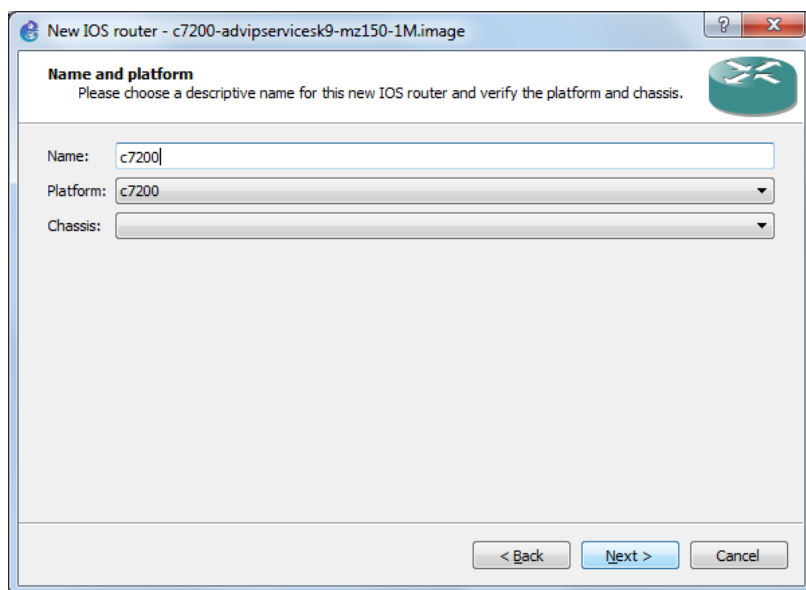


Figura 18 – Nome e plataforma do roteador

Fonte: Autoria Própria

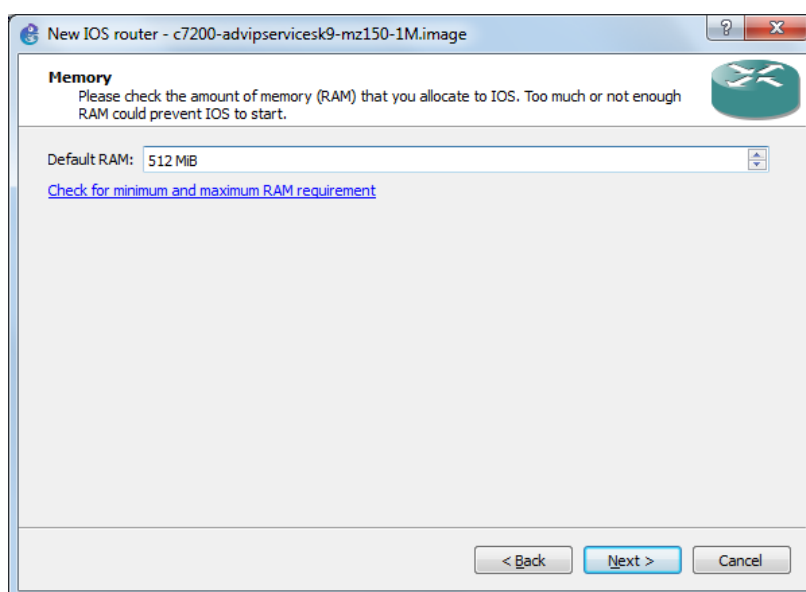


Figura 19 – Memória alocada ao roteador

Fonte: Autoria Própria

É importante destacar que a interligação entre os roteadores do projeto é feita através de interfaces *GigabitEthernet*. A partir daí os slots dos adaptadores são configurados. No GNS3, o adaptador correspondente ao *GigabitEthernet* é o *PA-GE*. Na figura 20 há um exemplo de como essa configuração é representada.

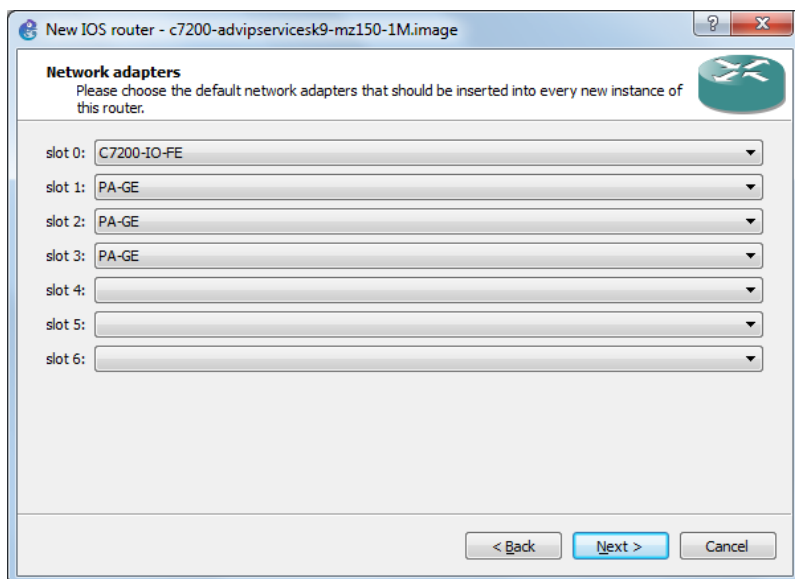


Figura 20 – Slots *PA-GE* (*GigabitEthernet*)

Fonte: Autoria Própria

Novamente, destaca-se a importância do *Idle-PC*. Se esse valor não for configurado, esse processo pode ser decisivo e impedir que haja sucesso na simulação da rede MPLS no GNS3. Para isso, basta clicar em “*Idle-PC finder*” que o próprio sistema do GNS3 encontrará o valor de forma automática, de acordo com a imagem de roteador adicionada. A janela e a mensagem relacionadas ao *Idle-PC* podem ser verificadas nas figuras 21 e 22.

Para finalizar, a figura 23 traz a janela das preferências com a imagem de roteador já importada. A partir daí, mais nenhuma configuração é necessária nos processos de importação do GNS3, tanto para as máquinas virtuais como para os roteadores.

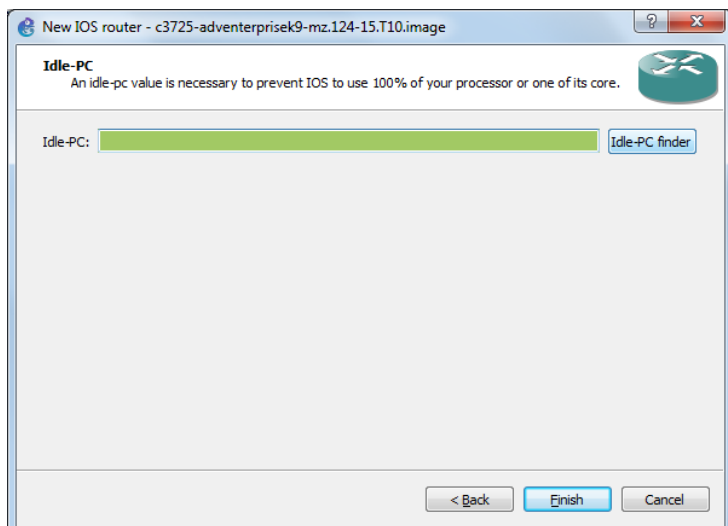


Figura 21 – Janela de configuração do *Idle-PC*

Fonte: Autoria Própria

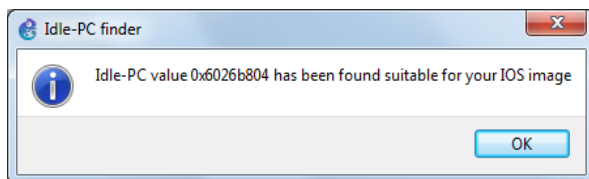


Figura 22 – Mensagem do valor de *Idle-PC* encontrado

Fonte: Autoria Própria

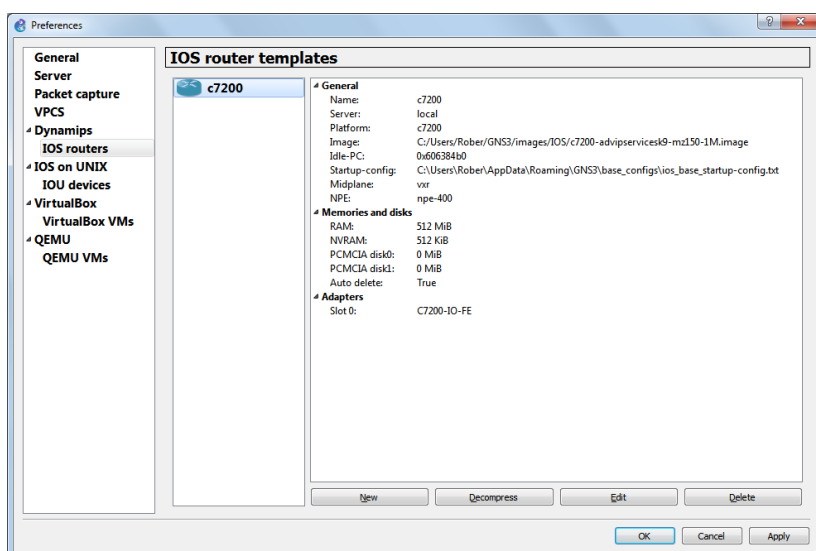


Figura 23 – Janela com uma imagem de roteador importada

Fonte Autoria Própria

3.1.4 Descrição das Configurações dos Roteadores

Aqui são apresentados os conceitos e toda a parte descritiva das configurações dos roteadores, as quais podem ser consultadas no apêndice deste material. Vale citar que nesse projeto foram utilizados sete roteadores, conforme ilustrado na Figura 7, sendo que quatro deles (R1, R2, R3 e R4) são os roteadores do núcleo, dois (R5 e R6) são os roteadores de borda e apenas um (R7) é um simples roteador local interligado ao roteador de borda R6.

É importante destacar que os roteadores R2, R3 e R4 obedecem ao mesmo padrão de configuração do roteador R1. Essa lógica também serve para os roteadores de borda, onde as configurações dos roteadores R5 e R6 também obedecem um mesmo padrão de configuração. Vale lembrar, novamente, que as configurações dos roteadores estão documentadas no apêndice deste trabalho. Considerando essas condições, compreende-se que tais descrições possam auxiliar no entendimento das configurações utilizadas.

Para o roteador R1, além das configurações padrão que um roteador necessita, foram ativados o protocolo OSPF, o MPLS, o tunelamento, o RSVP e aplicações de QoS, como classe, política e prioridade. Primeiramente configura-se os IPs, as máscaras e habilita-se as interfaces. Depois é necessário determinar o protocolo de roteamento - que é o OSPF - e escolher um ID específico para o roteador, além de determinar que o endereçamento e o tráfego sejam “área 0”, isto é, toda a informação é repassada para outra área que deve ser fisicamente conectada a anterior.

A figura 24 facilita a compreensão do que vem a ser a área 0. Na sequência, determina-se em quais interfaces haverá o MPLS e também se aplica um tunelamento no tráfego para que haja uma análise dos requisitos de QoS e, assim, seja possível selecionar qual a melhor rota para o encaminhamento de pacotes. Posterior a isso, o protocolo RSVP é configurado para permitir a reserva de recursos na rede e garantir a qualidade de serviço.

Finalmente, para o QoS é preciso, em primeiro lugar, criar uma classe de vídeo e aplicar um encaminhamento assegurado para essa classe, identificado como AF32 (Streaming de Vídeo), conforme a Tabela 2. Na sequência criou-se uma

política para vídeo, na qual é determinada uma prioridade de largura de banda para este tipo de tráfego na rede, que neste caso foi de 95%.

Vale destacar que a taxa de 600 kbps, aplicada no RSVP, aparece como a mais adequada na simulação de rede em questão, uma vez que a sua saturação permite a visualização das políticas de QoS aplicadas, considerando o tráfego simulado.

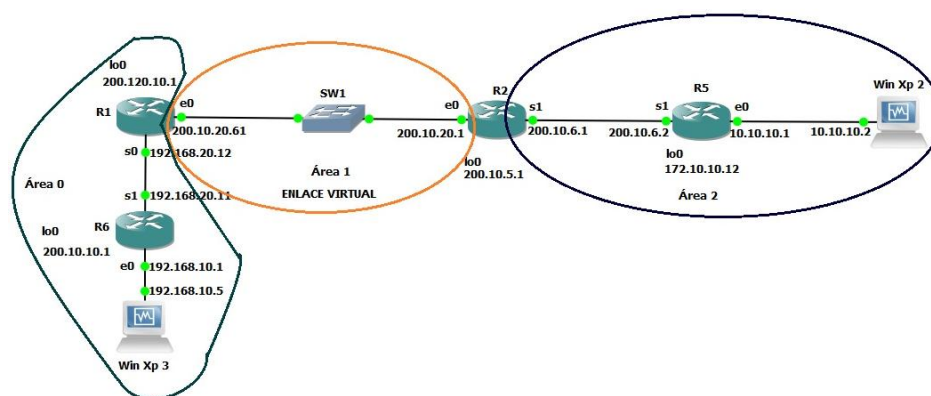


Figura 24 – Exemplo de um esquemático de diferenciação de áreas

Fonte: Autoria Própria

Os roteadores R5 e R6 possuem uma configuração de TE que tem a função de identificar para ambos qual é o destino a ser alcançado, indicando as opções de caminhos possíveis a esse destino, a partir de uma rota dinâmica.

Os roteadores R5 e R6 também possuem a aplicação de uma interface *loopback*, que facilita o processo de roteamento por manter adjacência entre os vizinhos, enviando e recebendo informações sobre o estado da rede. Se a interface *loopback* estiver ativa, terá prioridade sobre as outras na rede. Por ser uma interface virtual que sempre estará ativa, não apresentará problemas físicos (MENDES, 2013).

Outra diferença a ser notada, consiste nas interfaces dos roteadores R5 e R7 que estão ligadas diretamente aos hosts. Tais interfaces são as únicas que possuem uma taxa de transmissão inferior em relação às outras interfaces. Por fim, vale citar também que o roteador R7, por ser um simples roteador local, possui apenas as configurações básicas (IP, máscara e ativação da interface) e a inclusão do protocolo OSPF.

3.1.5 Configuração das Máquinas Virtuais

Todas as configurações citadas nos tópicos anteriores são essenciais para o funcionamento do projeto. No entanto, apenas instalar o sistema operacional das máquinas virtuais não é suficiente.

Antes de qualquer alteração, é necessário configurar manualmente o endereço IP, a máscara de sub-rede e o gateway padrão de cada máquina virtual. Os parâmetros configurados nessa rede podem ser visualizados nas figuras 25, 26 e 27, que correspondem aos hosts 1, 2 e 3 respectivamente.

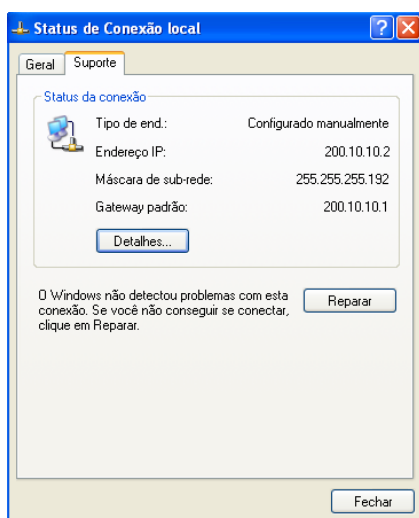


Figura 25 – Host 1

Fonte: Autoria Própria

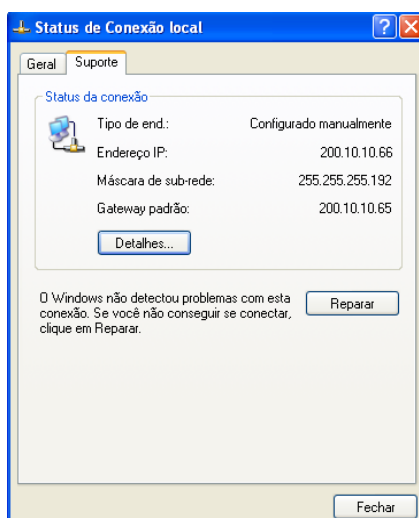


Figura 26 – Host 2

Fonte: Autoria Própria

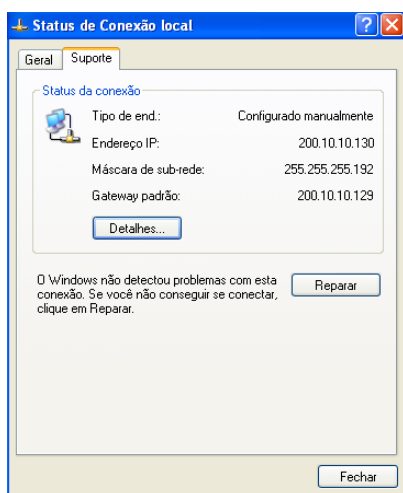


Figura 27 – Host 3

Fonte: Autoria Própria

Outro fator muito importante refere-se à probabilidade de as máquinas virtuais não conseguirem se comunicar, mesmo que todas as configurações básicas estejam corretas, impedindo o *ping* entre elas. Caso isso ocorra, o Firewall do Windows pode ser o causador do problema. Por isso, é importante desabilitá-lo em todos os hosts.

Na figura 28 há um exemplo que mostra a janela do Firewall do Windows, onde é possível alterar a configuração padrão e desabilitá-lo, apesar de ser uma alteração não recomendável por questões de segurança no que diz respeito à instalação e execução de programas. Entretanto, vale lembrar que essa alteração é feita na máquina virtual e por isso não deve afetar a máquina principal.

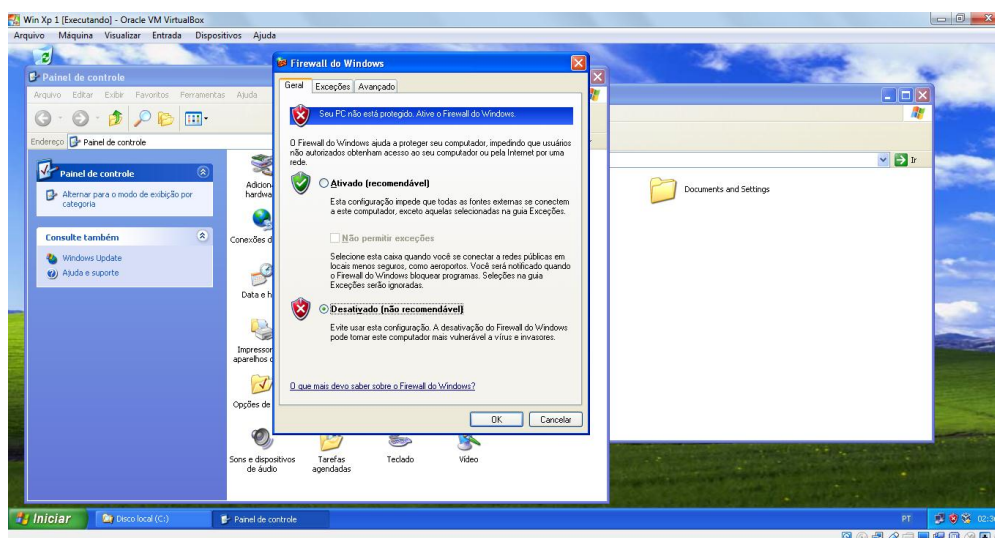


Figura 28 – Janela do Firewall do Windows

Fonte: Autoria Própria

O *ping* é um comando bastante utilizado no projeto, uma vez que é usado para testar a comunicação entre as máquinas na rede. Esse comando pode ser executado através do Prompt de Comando do Windows, o qual também permite outros tipos de testes do funcionamento da rede.

Através do comando *route print* é possível analisar uma tabela de roteamento, onde há várias informações da operação de roteamento que são úteis para a compreensão do encaminhamento de pacotes entre origem e destino. As rotas ativas podem ser visualizadas nas figuras 29, 30 e 31, que correspondem aos hosts 1, 2 e 3 respectivamente.

```

C:\>route print
=====
Lista de interfaces
0x1 ..... MS TCP Loopback interface
0x2 ...00 00 27 6f b0 b0 ..... AMD PCNET Family PCI Ethernet Adapter - Miniport
a do agendador de pacotes
=====
Rotas ativas:
Endereço de rede      Máscara      Ender. gateway      Interface      Custo
0.0.0.0              0.0.0.0       200.10.10.1         200.10.10.2    20
127.0.0.0            255.0.0.0     127.0.0.1          127.0.0.1      1
200.10.10.0          255.255.255.192 200.10.10.2        200.10.10.2    20
200.10.10.2          255.255.255.255 127.0.0.1          127.0.0.1      20
200.10.10.255        255.255.255.255 200.10.10.2        200.10.10.2    20
224.0.0.0            240.0.0.0     200.10.10.2        200.10.10.2    20
255.255.255.255     255.255.255.255 200.10.10.2        200.10.10.2    20
Gateway padrão:      200.10.10.1
=====
Rotas persistentes:
Nenhuma
C:\>

```

Figura 29 – Rotas ativas do Host 1

Fonte: Autoria Própria

```

C:\>route print
=====
Lista de interfaces
0x1 ..... MS TCP Loopback interface
0x2 ...00 00 27 6f b0 b0 ..... AMD PCNET Family PCI Ethernet Adapter - Miniport
a do agendador de pacotes
=====
Rotas ativas:
Endereço de rede      Máscara      Ender. gateway      Interface      Custo
0.0.0.0              0.0.0.0       200.10.10.65        200.10.10.66    20
127.0.0.0            255.0.0.0     127.0.0.1          127.0.0.1      1
200.10.10.64          255.255.255.192 200.10.10.66        200.10.10.66    20
200.10.10.66          255.255.255.255 127.0.0.1          127.0.0.1      20
200.10.10.255        255.255.255.255 200.10.10.66        200.10.10.66    20
224.0.0.0            240.0.0.0     200.10.10.66        200.10.10.66    20
255.255.255.255     255.255.255.255 200.10.10.66        200.10.10.66    1
Gateway padrão:      200.10.10.65
=====
Rotas persistentes:
Nenhuma
C:\>_

```

Figura 30 – Rotas ativas do Host 2

Fonte: Autoria Própria

```

C:\>route print
=====
Lista de interfaces
0x1 ..... MS TCP Loopback interface
0x2 ...00 00 27 6f b0 b0 ..... AMD PCNET Family PCI Ethernet Adapter - Miniport
a do agendador de pacotes
=====
Rotas ativas:
Endereço de rede      Máscara      Ender. gateway      Interface      Custo
0.0.0.0              0.0.0.0       200.10.10.129       200.10.10.130    20
127.0.0.0            255.0.0.0     127.0.0.1          127.0.0.1      1
200.10.10.128         255.255.255.192 200.10.10.130       200.10.10.130    20
200.10.10.130         255.255.255.255 127.0.0.1          127.0.0.1      20
200.10.10.255        255.255.255.255 200.10.10.130       200.10.10.130    20
224.0.0.0            240.0.0.0     200.10.10.130       200.10.10.130    20
255.255.255.255     255.255.255.255 200.10.10.130       200.10.10.130    1
Gateway padrão:      200.10.10.129
=====
Rotas persistentes:
Nenhuma
C:\>

```

Figura 31 – Rotas ativas do Host 3

Fonte: Autoria Própria

3.1.6 Análise da Prioridade de Tráfego

Um dos objetivos desse projeto é salientar o uso da prioridade de tráfego em função do QoS em MPLS. Com isso, mais do que apenas implementar é interessante monitorar e analisar essa funcionalidade da rede.

A partir disso, o software OmniPeek passa ser um item importante no projeto, uma vez que tem a função de auxiliar na monitoração do funcionamento da rede e destacar a prioridade de tráfego em uso, em tempo real. As informações sobre o software OminPeek podem ser visualizadas na figura 32.



Figura 32 – Software OmniPeek (Versão)

Fonte: Autoria Própria

O processo de análise consiste em se realizar a captura do tráfego na rede, fazendo com que o software traga informações relativas a esse tráfego, como a porcentagem do total de bytes do fluxo de dados na rede. É com essa informação que a prioridade de tráfego definida pode ser analisada e, a partir daí, possibilitar o avaliar o funcionamento da rede. Na figura 33 há um exemplo do processo de captura.

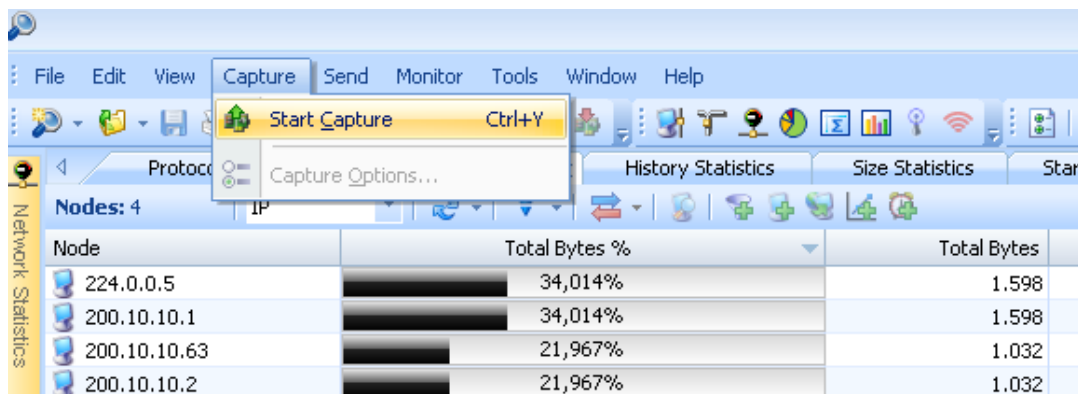


Figura 33 – Processo de captura do fluxo em tráfego na rede

Fonte: Autoria Própria

Além disso, é importante definir a velocidade da rede no software OminPeek para que o acompanhamento seja correto. Essa velocidade pode variar, considerando que ela depende da configuração de largura de banda aplicada nos roteadores. Nas figuras 34, 35 e 36 seguem as janelas de configuração da velocidade da rede.

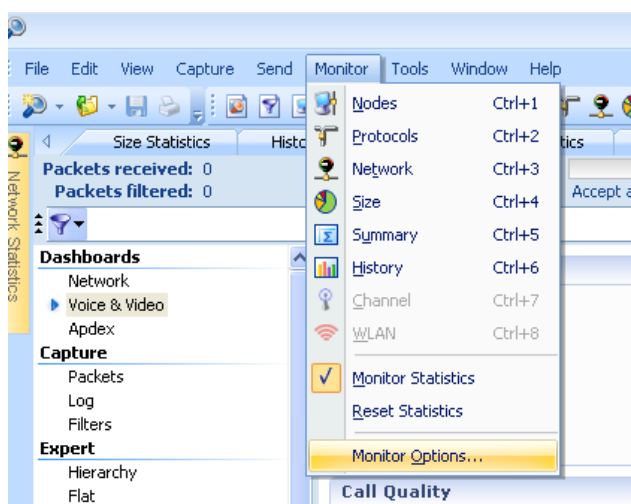


Figura 34 – Aba de opções de monitoração

Fonte: Autoria Própria

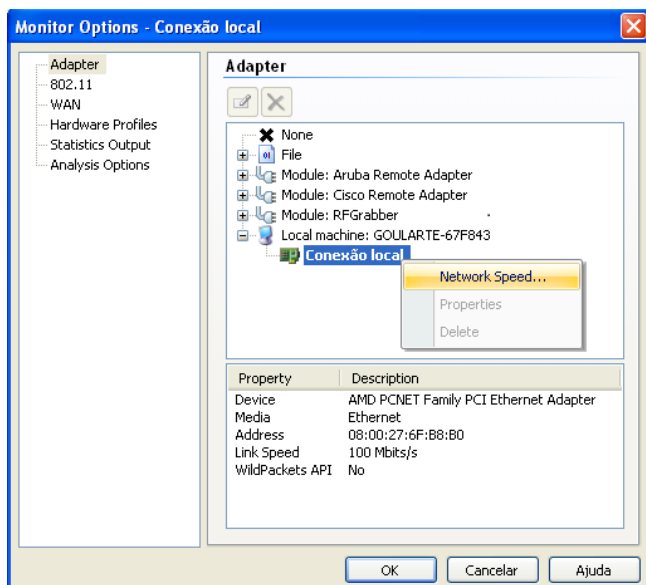


Figura 35 – Opção com a conexão local do adaptador

Fonte: Aatoria Própria

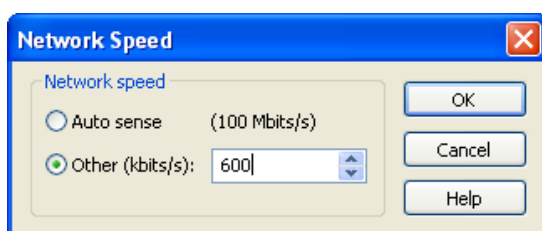


Figura 36 – Janela de definição da velocidade da rede

Fonte: Aatoria Própria

Para este projeto o software OmniPeek, através de interface gráfica, apresentou de forma clara e evidente os dados esperados. Por meio desta ferramenta foi possível comprovar, em tempo real, o uso da prioridade de largura de banda reservada para o tráfego de vídeo na rede, anteriormente configurada nos roteadores a partir de uma política de QoS.

É importante destacar que o software OmniPeek foi decisivo na avaliação dos resultados, mostrando importante influência no desenvolvimento desse trabalho, pelo fato de auxiliar na comprovação do correto funcionamento da rede e, assim, permitir o bom andamento do projeto.

3.1.7 Resultados das Simulações

A partir da implementação de QoS foi possível verificar na simulação do projeto, com o auxílio da ferramenta OmniPeek, o comportamento do tráfego de vídeo, executado em paralelo com o download de um arquivo (um outro tipo de tráfego). Esse procedimento consiste em mostrar que a configuração de RSVP para uma largura de banda de 600 kbps permitiu que 95% dessa banda estivesse destinada ao tráfego de vídeo.

Por esse motivo, a transferência do arquivo fica em segundo plano, o que diminui a velocidade do download e leva mais tempo para concluir (em torno de 5 minutos). A janela de monitoramento da ferramenta OmniPeek mais importante para a comprovação da aplicação de QoS na rede MPLS pode ser vista na Figura 37.

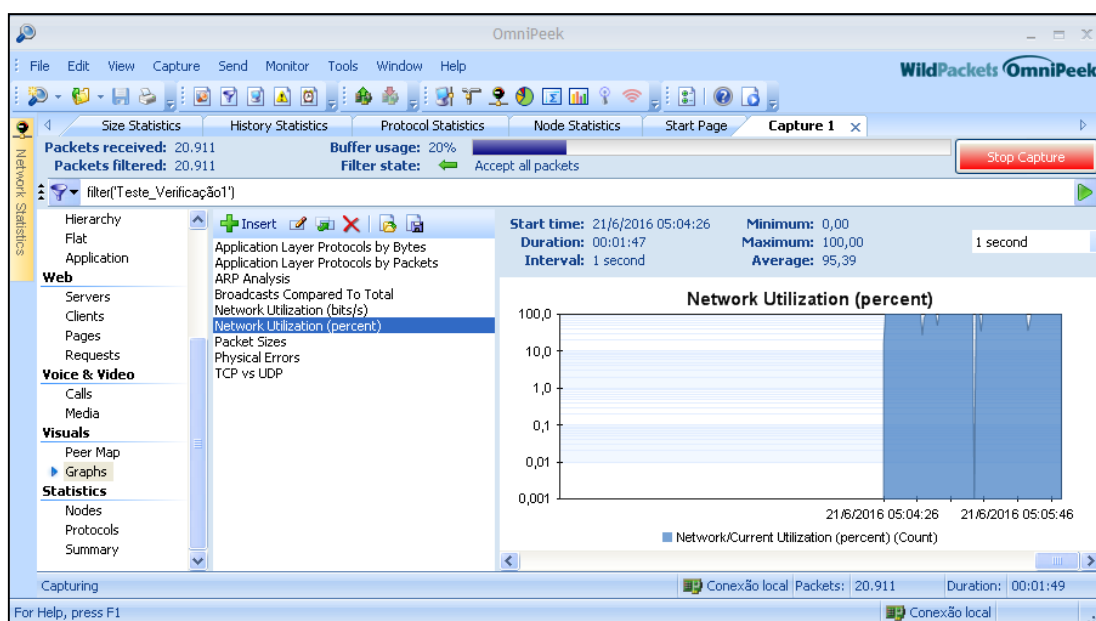


Figura 37 – Janela de monitoramento de utilização da rede com QoS

Fonte: Autoria Própria

Além da simulação anterior foi realizado um outro experimento: analisar o comportamento da rede, no mesmo cenário, sem QoS. Isto é, atuar em condições normais de melhor esforço (*Best Effort*), sem prioridade para o tráfego de vídeo. Com isso, nesta segunda simulação é possível observar a diminuição na média de utilização da rede (aproximadamente 75%) e o aumento na velocidade de

transferência do arquivo que, desta vez, leva menos tempo para concluir (em torno de 1 minuto e 30 segundos).

Na figura 38 é apresentada a janela de monitoramento, da ferramenta OmniPeek, de utilização da rede sem QoS.

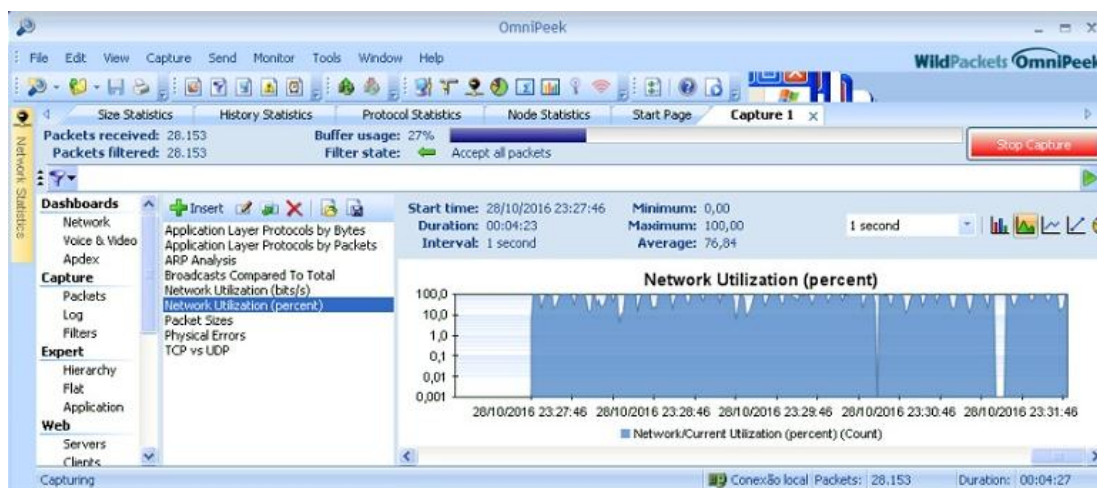


Figura 38 – Janela de monitoramento de utilização da rede sem QoS

Fonte: Autoria Própria

4 CONSIDERAÇÕES FINAIS

A análise do contexto geral deste trabalho permite concluir quais são os principais pontos destacados no decorrer do projeto de pesquisa: o protocolo MPLS, a aplicação de QoS e um ambiente de simulação. A junção desses fatores consiste em desenvolver uma rede que possibilite ao usuário determinar a sua base inicial, conhecer os recursos disponíveis e aprender a melhorar as suas práticas de implementação de projetos.

O trabalho desenvolvido vai além de uma simples apresentação de conceitos do MPLS e do porquê criar políticas de QoS para determinado tipo de tráfego em uma rede WAN. Uma das principais intenções é salientar a importância de uma simulação de um projeto de rede antes de sua implementação real.

Neste trabalho, a simulação do projeto de rede e toda a descrição de configurações no desenvolvimento esteve voltada à aplicação de MPLS com QoS, dando prioridade ao tráfego de vídeo. Percebeu-se que a aplicação de políticas de QoS melhoram a performance de determinado tipo de tráfego selecionado, mas inevitavelmente pioram outro tipo de tráfego ativo na rede.

Neste caso, quando há a aplicação de QoS, o streaming de vídeo funciona em condições favoráveis, com bom desempenho e de forma estável. No entanto, a transferência de um arquivo referente a outro tipo de tráfego acaba sendo prejudicada, justamente por não possuir tal privilégio. Quando não há aplicação de QoS, a rede atua de maneira padrão e usa o método do melhor esforço que, de certa forma, equilibra o uso da rede e não dá prioridade a nenhum dos tráfegos existentes, pois todos passam a ter o mesmo tratamento.

O interesse no assunto surgiu a partir de uma disciplina de Redes que apresentou uma base do MPLS e suas funcionalidades, demonstrando as aplicações práticas e as grandes vantagens desse protocolo. A partir disso, a busca por um maior aprofundamento resultou na elaboração deste trabalho, que permitiu aos integrantes da equipe obter mais conhecimento, principalmente na manipulação de simuladores, emuladores e ferramentas de monitoramento do tráfego da rede.

A partir daí, foi possível concluir que o MPLS é uma das melhores soluções existentes no mercado e já é utilizado em grandes ambientes corporativos em

função da sua escalabilidade, flexibilidade e baixo custo. Viu-se também a importância da utilização do mecanismo QoS para solucionar problemas relacionados ao tráfego de dados na rede, o que se faz fundamental para atender à vasta demanda de serviços e impedir que haja redução na produtividade de grandes, médias e pequenas empresas.

A experiência adquirida é de grande importância para futuros trabalhos na área e também no âmbito profissional, tanto no entendimento da infraestrutura de um ambiente corporativo como na prospecção e análise técnica de novos projetos. Vale destacar que o conhecimento é a maior virtude e também um grande diferencial que ajuda a fundamentar uma visão sistêmica e estratégica, indispensável a uma equipe que deseja implementar projetos com eficácia, otimizando a estrutura de rede de uma organização.

REFERÊNCIAS

(AMARAL, 2008) AMARAL, Italo. **QoS (Qualidade de Serviço) – parte II**. Disponível em: <<http://blog.ccna.com.br/2008/08/25/qos-qualidade-de-servico-parte-ii/>>. Acesso em: 24 jun. 2016.

(BRUN *et al*, 2002) BRUN, Altamir; VOGT, Eide Marta Gonçalves; MENDES, Alessandra da Silveira. **QoS – Qualidade de Serviço em TCP/IP**. Março 2002.

(CISCO SYSTEMS, 2006) CISCO SYSTEMS. **Implementing Cisco Quality of Service volume 1-2 version 2.2**, EUA, 2006.

(CISCO SYSTEMS, 2008) CISCO SYSTEMS. **Implementando políticas de qualidade de serviço com o DSCP**. Disponível em: <http://www.cisco.com/cisco/web/support/BR/8/83/83022_dscpvalues.html>. Acesso em: 24 jun. 2016.

(CISCO SYSTEMS, 2016) CISCO SYSTEMS. Disponível em: <<http://www.cisco.com/>>. Acesso em: 10 ago. 2016.

(DAVIDSON *et al*, 2007) DAVIDSON, Jonathan; PETERS, James; BHATIA, Manoj; KALINDINDI, Satish, MUKHERJEE, Sudipto. **Voice over IP Fundamentals**. 2 ed. Indianapolis: Cisco Press, 2007.

(DE GHEIN, 2007) DE GHEIN, Luc. **MPLS Fundamentals**. Indianapolis: Cisco Press, 2007.

(DIAS, 2013) DIAS, Diego. **Resumo: Suporte de Voz para a LAN**. Disponível em: <<http://www.rotadefault.com.br/tag/dscp/>>. Acesso em: 29 jun. 2016.

(FILIPPETTI, 2002) FILIPPETTI, Marco A. **CCNA 3.0 Guia Completo de Estudo**. Apêndice D – Glossário. 1 ed. 2002.

(FRAHIM *et al*, 2004) FRAHIM, Erum; FROOM, Richard; SIVASUBRAMANIAN, Balaji. **CCNP Self-Study: Understanding and Implementing Quality of Service in Cisco Multilayer Switched Networks**. Indianapolis: Cisco Press, 2004.

(GIL, 1989) GIL, Antonio Carlos. **Como Elaborar Projetos de Pesquisa**. 2 ed. São Paulo: Atlas, 1989.

(HARDENY, 2002) HARDENY, S. **The MPLS Primer**. New Jersey: Prentice Hall, 2002.

(JUNIOR, 2015) JUNIOR, Leomar Viegas. **Redes definidas por software**. Disponível em: <<http://blog.leomarviegas.com/wp-content/uploads/2015/12/Redes-Definidas-por-Software-Leomar-Viegas.pdf>>. Acesso em: 24 jun. 2016.

(LINS *et al*, 2011) LINS, Rafael Dueire; BARBOSA, Douglas; NASCIMENTO, Victor. **VoIP – Conceitos e Aplicações**. 1 ed. Rio de Janeiro: Brasport, 2011.

(LOBO, 2008) LOBO, Lancy. **MPLS Configuration on Cisco IOS Software**. Indianapolis: Cisco Press, 2008.

(LUCEK e MINEI, 2005) LUCEK, Julian; MINEI, Ina. **MPLS – Enabled Applications: emerging developments and new technologies**. Indianapolis: Wiley, 2005.

(MAGALHÃES e GUARDIEIRO, 2002) MAGALHÃES, Johann M. H.; GUARDIEIRO, Paulo R. **Avaliação de um Ambiente de Serviços Diferenciados com Tráfego de Vídeo MPEG-4**. Belo Horizonte: Universidade Federal de Uberlândia. Disponível em: <<http://www.eletrica.ufpr.br/anais/sbrt/SBrT19/001000000009300165.pdf>>. Acesso em: 24 jun. 2016.

(MENDES, 2013) MENDES, Roberto G. **Redes MPLS I: Modelo Conceitual – 1**. Disponível em: <http://www.teleco.com.br/tutoriais/tutorialmplscam/pagina_4.asp>. Acesso em: 24 ago. 2016.

(MPLS, 2010) THE MPLS forwarding plane by INE Instructor. Disponível em: <<http://blog.ine.com/2010/02/21/the-mpls-forwarding-plane/>>. Acesso em: 19 jun. 2016.

(NETCONTRACTOR, 2011) QoS values calculator v3. Disponível em: <<http://www.netcontractor.pl/download/QoS%20Values%20Calculator%20v3.pdf>>. Acesso em: 11 jun. 2016.

(ODOM e CAVANAUGH, 2004) ODOM, Wendell; CAVANAUGH, Michael. **IP Telephony Self-Study Cisco DQOS**. Indianapolis: Cisco Press, 2004.

(OLIVEIRA, José Mário; LINS, Rafael Dueire; MENDONÇA, Roberto. **Redes MPLS: Fundamentos e Aplicações**. 1 ed. Rio de Janeiro: Brasport, 2012.

(OSBORNE e SIMHA, 2002) OSBORNE, Eric; SIMHA, Ajay. **Traffic Engineering with MPLS**. Indianapolis: Cisco Press, 2002.

(ROSEN *et al*, 2001) ROSEN, E.; TAPPAN, D.; FEDORKOW, G.; REKHTER, Y.; LI, T.; CONTA, A. **RFC 3032**. MPLS Label Stack Encoding. The Internet Society, 2001.

(SURYANTOFANG, 2013) SURYANTOFANG. **QoS Lesson Notes: ECN (Explicit Congestion Notification)**. Disponível em: <<https://suryantofang.wordpress.com/2013/08/16/qos-lesson-notes-ecn-explicit-congestion-notification/>>. Acesso em: 24 jun. 2016.

(SVERZUT, 2008) SVERZUT, José Umberto. **Redes Convergentes**. 1 ed. São Paulo: Artliber, 2008.

(TANENBAUM, 2011) TANENBAUM, Andrew. **Redes de Computadores**. 5 ed. São Paulo: Pearson, 2011.

GLOSSÁRIO

Backbone – A interconexão central de uma rede internet. Pode ser entendido como uma espinha dorsal de conexões que interliga pontos distribuídos de uma rede, formando uma grande via por onde trafegam informações (FILIPPETTI, 2002).

Firewall – Um sistema de segurança de rede, cujo principal objetivo é filtrar o acesso a uma rede (FILIPPETTI, 2002).

Gateway – 1. Sistema que possibilita o intercâmbio de serviços entre redes com tecnologias completamente distintas; 2. Sistema e convenções de interconexão entre duas redes de mesmo nível e idêntica tecnologia, mas sob administrações distintas; 3. Roteador (terminologia TCP/IP) (FILIPPETTI, 2002).

Host – Computador ligado a uma rede (FILIPPETTI, 2002).

Internet – Uma coleção de redes locais e/ou de longa distância, interligadas numa rede virtual pelo uso de um protocolo que provê um espaço de endereçamento comum e roteamento (FILIPPETTI, 2002).

IP – O Internet Protocol é o protocolo responsável pelo roteamento de pacotes entre dois sistemas que utilizam a família de protocolos TCP/IP, desenvolvida e usada na internet (FILIPPETTI, 2002).

Pacote – Dado encapsulado para transmissão na rede. Um conjunto de bits compreendendo informação de controle, endereço fonte e destino dos nós envolvidos na transmissão (FILIPPETTI, 2002).

Ping – O ping (Packet Internet Groper) é um programa usado para testar o alcance de uma rede, enviando a nós remotos uma requisição e esperando por uma resposta (FILIPPETTI, 2002).

Protocolo – Uma descrição formal de formatos de mensagem e das regras que dois computadores devem obedecer ao trocar mensagens. Um conjunto de regras padronizado que especifica o formato, a sincronização, o sequenciamento e a verificação de erros em comunicação de dados. O protocolo básico utilizado na Internet é o TCP/IP (FILIPPETTI, 2002).

TCP/IP – Acrônimo de *Transmission Control Protocol / Internet Protocol*, é a família de protocolos para a comunicação de dados inter-redes, originalmente proposta para a *Advanced Research Products Agency Network* (ARPANet). Hoje é um padrão de fato para inter-redes abertas e seu uso é amplamente difundido (FILIPPETTI, 2002).

WAN (Rede de Longa Distância) – Acrônimo de *Wide Area Network*, uma rede que interliga computadores distribuídos em área geograficamente separadas (FILIPPETTI, 2002).

APÊNDICE

CONFIGURAÇÕES DO PROJETO

Roteador R1 (R2, R3 e R4 – Mesmo padrão de configuração)

```
conf term
enable password rober
enable secret mpls
interface gigabitEthernet4/0
ip address 10.10.10.9 255.255.255.252
no shutdown
interface gigabitEthernet1/0
ip address 10.10.10.21 255.255.255.252
no shutdown
interface gigabitEthernet2/0
ip address 10.10.10.25 255.255.255.252
no shutdown
interface gigabitEthernet3/0
ip address 10.10.10.6 255.255.255.252
no shutdown
```

OSPF

```
conf term
router ospf 1
router-id 2.2.2.2
network 10.10.10.4 0.0.0.3 area 0
network 10.10.10.20 0.0.0.3 area 0
network 10.10.10.24 0.0.0.3 area 0
network 10.10.10.8 0.0.0.3 area 0
router ospf 1
```

```
network 10.10.10.4 0.0.0.3 area 0
network 10.10.10.20 0.0.0.3 area 0
network 10.10.10.24 0.0.0.3 area 0
network 10.10.10.8 0.0.0.3 area 0
mpls traffic-eng area 0
mpls traffic-eng router-id gigabitEthernet 4/0
mpls traffic-eng router-id gigabitEthernet 1/0
mpls traffic-eng router-id gigabitEthernet 2/0
mpls traffic-eng router-id gigabitEthernet 3/0
```

MPLS

```
conf term
mpls label protocol ldp
interface g4/0
mpls ip
interface g1/0
mpls ip
interface g2/0
mpls ip
interface g3/0
mpls ip
```

Tunnel

```
mpls traffic-eng tunnels
interface gigabitEthernet4/0
ip address 10.10.10.9 255.255.255.252
mpls traffic-eng tunnels
interface gigabitEthernet1/0
ip address 10.10.10.21 255.255.255.252
mpls traffic-eng tunnels
interface gigabitEthernet2/0
ip address 10.10.10.25 255.255.255.252
mpls traffic-eng tunnels
interface gigabitEthernet3/0
```

```
ip address 10.10.10.6 255.255.255.252
mpls traffic-eng tunnels
```

RSVP

```
interface gigabitEthernet4/0
ip rsvp bandwidth 600
interface gigabitEthernet1/0
ip rsvp bandwidth 600
interface gigabitEthernet2/0
ip rsvp bandwidth 600
interface gigabitEthernet3/0
ip rsvp bandwidth 600
```

QoS (Classe)

```
class-map match-all Video
match ip dscp af32
```

QoS (Política)

```
policy-map Qos_Video
class Video
set ip dscp af32
policy-map Qos_Prioridade
class Video
```

QoS (Prioridade)

```
priority percent 95
interface g1/0
service-policy output Qos_Video
interface g2/0
service-policy output Qos_Video
interface g3/0
service-policy output Qos_Video
interface g4/0
service-policy output Qos_Video
```

Roteador R5

```
conf term
enable password rober
enable secret mpls
interface gigabitEthernet2/0
ip address 10.10.10.34 255.255.255.252
no shutdown
interface gigabitEthernet3/0
ip address 10.10.10.42 255.255.255.252
no shutdown
interface ethernet1/0
ip address 200.10.10.65 255.255.255.192
no shutdown
interface loopback 0
ip add 200.10.10.65 255.255.255.192
```

OSPF

```
conf term
router ospf 1
router-id 6.6.6.6
network 10.10.10.32 0.0.0.3 area 0
network 10.10.10.40 0.0.0.3 area 0
network 200.10.10.64 0.0.0.63 area 0
router ospf 1
network 10.10.10.32 0.0.0.3 area 0
network 10.10.10.40 0.0.0.3 area 0
mpls traffic-eng area 0
mpls traffic-eng router-id gigabitEthernet 2/0
mpls traffic-eng router-id gigabitEthernet 3/0
```

MPLS

```
conf term
```

```
mpls label protocol ldp
interface g2/0
mpls ip
interface g3/0
mpls ip
```

Tunnel

```
mpls traffic-eng tunnels
interface gigabitEthernet2/0
ip address 10.10.10.34 255.255.255.252
mpls traffic-eng tunnels
interface gigabitEthernet3/0
ip address 10.10.10.42 255.255.255.252
mpls traffic-eng tunnels
```

RSVP

```
interface gigabitEthernet2/0
ip rsvp bandwidth 600
interface gigabitEthernet3/0
ip rsvp bandwidth 600
```

Engenharia de Tráfego

```
interface tunnel 2
ip unnumbered loopback 0
tunnel mode mpls traffic-eng
tunnel destination 200.10.10.2
tunnel mpls traffic-eng path-option 1 dynamic
tunnel mpls traffic-eng bandwidth 600
tunnel mpls traffic-eng priority 1
interface tunnel 2
tunnel mpls traffic-eng autoroute announce
```

QoS (Classe)

```
class-map match-all Video
```



```
match ip dscp af32
```

QoS (Política)

```
policy-map Qos_Video  
class Video  
set ip dscp af32  
policy-map Qos_Prioridade  
class Video
```

QoS (Prioridade)

```
priority percent 95  
interface g2/0  
service-policy output Qos_Video  
interface g3/0  
service-policy output Qos_Video
```

Roteador R6

```
conf term  
enable password rober  
enable secret mpls  
interface gigabitEthernet3/0  
ip address 10.10.10.5 255.255.255.252  
no shutdown  
interface gigabitEthernet0/0  
ip address 200.10.10.129 255.255.255.192  
no shutdown  
interface gigabitEthernet1/0  
ip address 10.10.10.37 255.255.255.252  
no shutdown  
interface gigabitEthernet 2/0  
ip address 10.10.10.2 255.255.255.252  
no shutdown
```

```
interface loopback 0
ip add 200.10.10.129 255.255.255.192
```

OSPF

```
conf term
router ospf 1
router-id 1.1.1.1
network 10.10.10.4 0.0.0.3 area 0
network 10.10.10.0 0.0.0.3 area 0
network 10.10.10.36 0.0.0.3 area 0
network 200.10.10.128 0.0.0.3 area 0
conf term
router ospf 1
network 10.10.10.4 0.0.0.3 area 0
network 10.10.10.0 0.0.0.3 area 0
network 10.10.10.36 0.0.0.3 area 0
network 200.10.10.128 0.0.0.3 area 0
mpls traffic-eng area 0
mpls traffic-eng router-id gigabitEthernet 3/0
mpls traffic-eng router-id gigabitEthernet 1/0
mpls traffic-eng router-id gigabitEthernet 0/0
```

MPLS

```
conf term
mpls label protocol ldp
interface g3/0
mpls ip
interface g1/0
mpls ip
interface g0/0
mpls ip
```

Tunnel

```
mpls traffic-eng tunnels
```

```
interface gigabitEthernet3/0
ip address 10.10.10.5 255.255.255.252
mpls traffic-eng tunnels
interface gigabitEthernet1/0
ip address 10.10.10.37 255.255.255.252
mpls traffic-eng tunnels
```

RSVP

```
interface gigabitEthernet3/0
ip rsvp bandwidth 600
interface gigabitEthernet1/0
ip rsvp bandwidth 600
interface gigabitEthernet0/0
ip rsvp bandwidth 600
```

Engenharia de Tráfego

```
interface tunnel 1
ip unnumbered loopback 0
tunnel mode mpls traffic-eng
tunnel destination 200.10.10.66
tunnel mpls traffic-eng path-option 1 dynamic
tunnel mpls traffic-eng bandwidth 600
tunnel mpls traffic-eng priority 0
interface tunnel 1
tunnel mpls traffic-eng autoroute announce
```

QoS (Classe)

```
class-map match-all Video
match ip dscp af32
```

QoS (Política)

```
policy-map Qos_Video
class Video
set ip dscp af32
```

```
policy-map Qos_Prioridade  
class Video
```

QoS (Prioridade)

```
priority percent 95  
interface g3/0  
service-policy input Qos_Video  
interface g1/0  
service-policy input Qos_Video  
interface g0/0  
service-policy input Qos_Video
```

Roteador R7

```
conf term  
enable password rober  
enable secret mpls  
service password-encryption  
interface ethernet2/0  
ip address 200.10.10.1 255.255.255.192  
no shutdown  
interface gigabitEthernet1/0  
ip address 10.10.10.1 255.255.255.252  
no shutdown  
interface loopback 0  
ip add 200.10.10.1 255.255.255.192
```

OSPF

```
conf term  
router ospf 1  
router-id 7.7.7.7  
network 10.10.10.0 0.0.0.3 area 0  
network 200.10.10.0 0.0.0.63 area 0
```

COMANDOS UTILIZADOS

IP

- sh ip route
- sh ip ospf interface
- sh ip interface brief
- sh ip rsvp interface
- sh ip rsvp interface detail
- sh ip rsvp interface detail gigabitEthernet 3/0
- sh ip rsvp
- sh ip ospf database
- sh ip ospf neighbor detail
- sh ip ospf neighbor
- sh ip protocols
- sh ip route ospf
- sh ip ospf events
- sh ip rsvp policy

MPLS

- sh mpls interface detail
- sh mpls interfaces
- sh mpls interfaces all
- sh tag-switching tdp bindings
- sh tag-switching forwarding-table 200.10.10.2
- sh tag-switching forwarding-table 200.10.10.2 detail
- sh mpls ip binding
- sh mpls ldp bindings
- sh mpls ldp discovery
- sh mpls ldp neighbor

- sh mpls label range
- sh mpls ldp parameters
- sh mpls forwarding-table
- sh mpls ldp backoff
- sh mpls traffic-eng tunnels brief

Policy Map

- sh policy-map
- sh policy-map interface
- sh policy-manager policy
- sh policy-map Qos_Prioridade
- sh class-map
- sh run interface g1/0

DETALHAMENTO DOS PRINCIPAIS COMANDOS

- **sh ip route**

Traz como atingir as rotas (redes) a partir de um roteador, onde: O antes do endereço de rede indica que utiliza Protocolo de Roteamento OSPF; C antes do endereço de rede indica que tal rede está diretamente conectada ao roteador onde executa o comando sh ip route.

```

R1
R1#
R1#sh ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 15 subnets, 2 masks
O    10.10.10.0/30 [110/2] via 10.10.10.5, 00:38:38, GigabitEthernet3/0
C    10.10.10.4/30 is directly connected, GigabitEthernet3/0
L    10.10.10.6/32 is directly connected, GigabitEthernet3/0
C    10.10.10.8/30 is directly connected, GigabitEthernet4/0
L    10.10.10.9/32 is directly connected, GigabitEthernet4/0
O    10.10.10.12/30 [110/2] via 10.10.10.26, 00:38:38, GigabitEthernet2/0
      [110/2] via 10.10.10.10, 00:38:28, GigabitEthernet4/0
O    10.10.10.16/30 [110/2] via 10.10.10.26, 00:38:38, GigabitEthernet2/0
      [110/2] via 10.10.10.22, 00:38:28, GigabitEthernet1/0
C    10.10.10.20/30 is directly connected, GigabitEthernet1/0
L    10.10.10.21/32 is directly connected, GigabitEthernet1/0
C    10.10.10.24/30 is directly connected, GigabitEthernet2/0
L    10.10.10.25/32 is directly connected, GigabitEthernet2/0
O    10.10.10.28/30 [110/2] via 10.10.10.22, 00:38:38, GigabitEthernet1/0
      [110/2] via 10.10.10.10, 00:38:38, GigabitEthernet4/0
O    10.10.10.32/30 [110/2] via 10.10.10.10, 00:38:39, GigabitEthernet4/0
O    10.10.10.36/30 [110/2] via 10.10.10.22, 00:38:42, GigabitEthernet1/0
      [110/2] via 10.10.10.5, 00:38:52, GigabitEthernet3/0
O    10.10.10.40/30 [110/2] via 10.10.10.26, 00:38:53, GigabitEthernet2/0
    200.10.10.0/26 is subnetted, 3 subnets
O    200.10.10.0 [110/12] via 10.10.10.5, 00:38:55, GigabitEthernet3/0
O    200.10.10.64 [110/12] via 10.10.10.26, 00:38:57, GigabitEthernet2/0
      [110/12] via 10.10.10.10, 00:38:47, GigabitEthernet4/0
O    200.10.10.128 [110/2] via 10.10.10.5, 00:39:04, GigabitEthernet3/0
R1#
R1#

```

• sh ip ospf interface

Traz para cada uma das interfaces conectadas ao roteador sua identificação, seu estado (no caso “up”), além do IP das interfaces com seu respectivo valor de máscara e a área. Traz o router ID deste roteador, criado no protocolo de roteamento OSPF, e quando ocorreu o “Hello”.

```

R1
R1#sh ip ospf interface
GigabitEthernet2/0 is up, line protocol is up
  Internet Address 10.10.10.25/30, Area 0
  Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
  0                  1         no           no           Base
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 4.4.4.4, Interface address 10.10.10.26
  Backup Designated router (ID) 2.2.2.2, Interface address 10.10.10.25
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:08
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 4/4, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 8, maximum is 8
  Last flood scan time is 0 msec, maximum is 4 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 4.4.4.4 (Designated Router)
  Suppress hello for 0 neighbor(s)
GigabitEthernet1/0 is up, line protocol is up
  Internet Address 10.10.10.21/30, Area 0
  Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
  0                  1         no           no           Base
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 3.3.3.3, Interface address 10.10.10.22
  Backup Designated router (ID) 2.2.2.2, Interface address 10.10.10.21
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:00
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 3/3, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 8, maximum is 8
--More--

```

```

R1
Last flood scan length is 8, maximum is 8
Last flood scan time is 0 msec, maximum is 4 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 3.3.3.3 (Designated Router)
Suppress hello for 0 neighbor(s)
GigabitEthernet4/0 is up, line protocol is up
Internet Address 10.10.10.9/30, Area 0
Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
Topology-MTID   Cost   Disabled   Shutdown   Topology Name
  0             1       no         no         Base
Transmit Delay is 1 sec, State BDR, Priority 1
Designated Router (ID) 5.5.5.5, Interface address 10.10.10.10
Backup Designated router (ID) 2.2.2.2, Interface address 10.10.10.9
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:00
Supports Link-local Signaling (LLS)
Cisco NSF helper support enabled
IETF NSF helper support enabled
Index 2/2, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 6, maximum is 7
Last flood scan time is 4 msec, maximum is 4 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 5.5.5.5 (Designated Router)
Suppress hello for 0 neighbor(s)
GigabitEthernet3/0 is up, line protocol is up
Internet Address 10.10.10.6/30, Area 0
Process ID 1, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 1
Topology-MTID   Cost   Disabled   Shutdown   Topology Name
  0             1       no         no         Base
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 2.2.2.2, Interface address 10.10.10.6
Backup Designated router (ID) 1.1.1.1, Interface address 10.10.10.5
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:06
Supports Link-local Signaling (LLS)
Cisco NSF helper support enabled
--More--

```

```

R1
Hello due in 00:00:06
Supports Link-local Signaling (LLS)
Cisco NSF helper support enabled
IETF NSF helper support enabled
Index 1/1, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 8, maximum is 23
Last Flood scan time is 0 msec, maximum is 4 msec
Neighbor Count is 1, Adjacent neighbor count is 1
  Adjacent with neighbor 1.1.1.1 (Backup Designated Router)
Suppress hello for 0 neighbor(s)
R1#

```

- **sh ip interface brief**

Traz as interfaces, IPs e o status.

```

R1
R1#
R1#sh ip interface brief
Interface                IP-Address      OK? Method Status          Protocol
FastEthernet0/0          unassigned      YES NVRAM    administrativ  down
GigabitEthernet1/0       10.10.10.21     YES NVRAM    up              up
GigabitEthernet2/0       10.10.10.25     YES NVRAM    up              up
GigabitEthernet3/0       10.10.10.6      YES NVRAM    up              up
GigabitEthernet4/0       10.10.10.9      YES NVRAM    up              up
R1#

```

- **sh ip rsvp interface**

Traz as interfaces 3 com o status do RSVP e a largura de banda configurada.

```

R1
R1#
R1#sh ip rsvp interface
interface  rsvp  allocated  i/f max  flow max  sub max  VRF
Gi1/0     ena    0           600K     600K     0
Gi2/0     ena    0           600K     600K     0
Gi3/0     ena    0           600K     600K     0
Gi4/0     ena    0           600K     600K     0
R1#

```


- **sh ip route ospf**

Traz todas as informações das rotas OSPF configuradas na rede.

```

R1#sh ip route ospf
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, + - replicated route

Gateway of last resort is not set

 10.0.0.0/8 is variably subnetted, 15 subnets, 2 masks
O    10.10.10.0/30 [110/2] via 10.10.10.5, 01:50:43, GigabitEthernet3/0
O    10.10.10.12/30 [110/2] via 10.10.10.26, 01:50:43, GigabitEthernet2/0
     [110/2] via 10.10.10.10, 01:50:33, GigabitEthernet4/0
O    10.10.10.16/30 [110/2] via 10.10.10.26, 01:50:43, GigabitEthernet2/0
     [110/2] via 10.10.10.22, 01:50:33, GigabitEthernet1/0
O    10.10.10.28/30 [110/2] via 10.10.10.22, 01:50:33, GigabitEthernet1/0
     [110/2] via 10.10.10.10, 01:50:33, GigabitEthernet4/0
O    10.10.10.32/30 [110/2] via 10.10.10.10, 01:50:33, GigabitEthernet4/0
O    10.10.10.36/30 [110/2] via 10.10.10.22, 01:50:33, GigabitEthernet1/0
     [110/2] via 10.10.10.5, 01:50:43, GigabitEthernet3/0
O    10.10.10.40/30 [110/2] via 10.10.10.26, 01:50:43, GigabitEthernet2/0
200.10.10.0/26 is subnetted, 3 subnets
O    200.10.10.0 [110/12] via 10.10.10.5, 01:50:48, GigabitEthernet3/0
O    200.10.10.64 [110/12] via 10.10.10.26, 01:50:48, GigabitEthernet2/0
     [110/12] via 10.10.10.10, 01:50:38, GigabitEthernet4/0
O    200.10.10.128 [110/2] via 10.10.10.5, 01:50:49, GigabitEthernet3/0
R1#
  
```

- **sh ip rsvp policy**

Traz as informações das políticas RSVP configuradas na rede.

```

R1#
R1#sh ip rsvp policy
Local policy:
COPS:

Generic policy settings:
  Default policy: Accept all
  Preemption:      Disabled
R1#
  
```

- **sh mpls interfaces**

Traz a identificação das interfaces que possuem mpls ativado e sua operacionalidade.

```

R1#
R1#sh mpls interfaces
Interface          IP          Tunnel  BGP  Static  Operational
GigabitEthernet1/0 Yes (ldp)   Yes     No   No       Yes
GigabitEthernet2/0 Yes (ldp)   Yes     No   No       Yes
GigabitEthernet3/0 Yes (ldp)   Yes     No   No       Yes
GigabitEthernet4/0 Yes (ldp)   Yes     No   No       Yes
R1#
  
```

- **sh tag-switching tdp bindings**

Traz o conteúdo da base de informações do rótulo (LIB).

```

R1
R1#sh tag-switching tdp bindings
% Command accepted but obsolete, unreleased or unsupported; see documentation.

tib entry: 10.10.10.0/30, rev 26
  local binding: tag: 24
  remote binding: tsr: 200.10.10.129:0, tag: imp-null
  remote binding: tsr: 10.10.10.38:0, tag: 22
  remote binding: tsr: 10.10.10.41:0, tag: 22
  remote binding: tsr: 10.10.10.33:0, tag: 25
tib entry: 10.10.10.4/30, rev 6
  local binding: tag: imp-null
  remote binding: tsr: 200.10.10.129:0, tag: imp-null
  remote binding: tsr: 10.10.10.38:0, tag: 21
  remote binding: tsr: 10.10.10.41:0, tag: 21
  remote binding: tsr: 10.10.10.33:0, tag: 24
tib entry: 10.10.10.8/30, rev 8
  local binding: tag: imp-null
  remote binding: tsr: 200.10.10.129:0, tag: 28
  remote binding: tsr: 10.10.10.38:0, tag: 26
  remote binding: tsr: 10.10.10.41:0, tag: 20
  remote binding: tsr: 10.10.10.33:0, tag: imp-null
tib entry: 10.10.10.12/30, rev 22
  local binding: tag: 22
  remote binding: tsr: 200.10.10.129:0, tag: 26
  remote binding: tsr: 10.10.10.38:0, tag: 25
  remote binding: tsr: 10.10.10.41:0, tag: imp-null
  remote binding: tsr: 10.10.10.33:0, tag: imp-null
tib entry: 10.10.10.16/30, rev 24
  local binding: tag: 23
  remote binding: tsr: 200.10.10.129:0, tag: 23
  remote binding: tsr: 10.10.10.38:0, tag: imp-null
  remote binding: tsr: 10.10.10.41:0, tag: imp-null
  remote binding: tsr: 10.10.10.33:0, tag: 23
tib entry: 10.10.10.20/30, rev 2
  local binding: tag: imp-null
  remote binding: tsr: 200.10.10.129:0, tag: 22
  remote binding: tsr: 10.10.10.38:0, tag: imp-null
  remote binding: tsr: 10.10.10.41:0, tag: 19
  remote binding: tsr: 10.10.10.33:0, tag: 22
tib entry: 10.10.10.24/30, rev 4
  local binding: tag: imp-null
  remote binding: tsr: 200.10.10.129:0, tag: 29
--More--

```

```

R1
  remote binding: tsr: 10.10.10.38:0, tag: 29
  remote binding: tsr: 10.10.10.41:0, tag: imp-null
  remote binding: tsr: 10.10.10.33:0, tag: 28
tib entry: 10.10.10.28/30, rev 18
  local binding: tag: 20
  remote binding: tsr: 200.10.10.129:0, tag: 21
  remote binding: tsr: 10.10.10.38:0, tag: imp-null
  remote binding: tsr: 10.10.10.41:0, tag: 24
  remote binding: tsr: 10.10.10.33:0, tag: imp-null
tib entry: 10.10.10.32/30, rev 16
  local binding: tag: 19
  remote binding: tsr: 200.10.10.129:0, tag: 25
  remote binding: tsr: 10.10.10.38:0, tag: 24
  remote binding: tsr: 10.10.10.41:0, tag: 23
  remote binding: tsr: 10.10.10.33:0, tag: imp-null
tib entry: 10.10.10.36/30, rev 28
  local binding: tag: 25
  remote binding: tsr: 200.10.10.129:0, tag: imp-null
  remote binding: tsr: 10.10.10.38:0, tag: imp-null
  remote binding: tsr: 10.10.10.41:0, tag: 25
  remote binding: tsr: 10.10.10.33:0, tag: 26
tib entry: 10.10.10.40/30, rev 20
  local binding: tag: 21
  remote binding: tsr: 200.10.10.129:0, tag: 27
  remote binding: tsr: 10.10.10.38:0, tag: 28
  remote binding: tsr: 10.10.10.41:0, tag: imp-null
  remote binding: tsr: 10.10.10.33:0, tag: 29
tib entry: 200.10.10.0/26, rev 14
  local binding: tag: 18
  remote binding: tsr: 200.10.10.129:0, tag: 20
  remote binding: tsr: 10.10.10.38:0, tag: 23
  remote binding: tsr: 10.10.10.41:0, tag: 18
  remote binding: tsr: 10.10.10.33:0, tag: 21
tib entry: 200.10.10.64/26, rev 12
  local binding: tag: 17
  remote binding: tsr: 200.10.10.129:0, tag: 24
  remote binding: tsr: 10.10.10.38:0, tag: 27
  remote binding: tsr: 10.10.10.41:0, tag: 17
  remote binding: tsr: 10.10.10.33:0, tag: 27
tib entry: 200.10.10.128/26, rev 10
  local binding: tag: 16
  remote binding: tsr: 200.10.10.129:0, tag: imp-null
--More--

```

```

R1
  remote binding: tsr: 10.10.10.38:0, tag: 20
  remote binding: tsr: 10.10.10.41:0, tag: 16
  remote binding: tsr: 10.10.10.33:0, tag: 20
R1#

```

- **sh mpls ip binding**

Traz as informações especificadas sobre ligações de etiqueta aprendidas pelo protocolo de distribuição de rótulo (LDP).

```

R1
R1#sh mpls ip binding
10.10.10.0/30
  in label:      24
  out label:    imp-null lsr: 200.10.10.129:0 inuse
  out label:    22      lsr: 10.10.10.38:0
  out label:    22      lsr: 10.10.10.41:0
  out label:    25      lsr: 10.10.10.33:0
10.10.10.4/30
  in label:      imp-null
  out label:    imp-null lsr: 200.10.10.129:0
  out label:    21      lsr: 10.10.10.38:0
  out label:    21      lsr: 10.10.10.41:0
  out label:    24      lsr: 10.10.10.33:0
10.10.10.8/30
  in label:      imp-null
  out label:    28      lsr: 200.10.10.129:0
  out label:    26      lsr: 10.10.10.38:0
  out label:    20      lsr: 10.10.10.41:0
  out label:    imp-null lsr: 10.10.10.33:0
10.10.10.12/30
  in label:      22
  out label:    26      lsr: 200.10.10.129:0
  out label:    25      lsr: 10.10.10.38:0
  out label:    imp-null lsr: 10.10.10.41:0 inuse
  out label:    imp-null lsr: 10.10.10.33:0 inuse
10.10.10.16/30
  in label:      23
  out label:    23      lsr: 200.10.10.129:0
  out label:    imp-null lsr: 10.10.10.38:0 inuse
  out label:    imp-null lsr: 10.10.10.41:0 inuse
  out label:    23      lsr: 10.10.10.33:0
10.10.10.20/30
  in label:      imp-null
  out label:    22      lsr: 200.10.10.129:0
  out label:    imp-null lsr: 10.10.10.38:0
  out label:    19      lsr: 10.10.10.41:0
  out label:    22      lsr: 10.10.10.33:0
10.10.10.24/30
  in label:      imp-null
  out label:    29      lsr: 200.10.10.129:0
  out label:    29      lsr: 10.10.10.38:0
  out label:    imp-null lsr: 10.10.10.41:0
--More--

```

```

R1
  out label:    28      lsr: 10.10.10.33:0
10.10.10.28/30
  in label:      20
  out label:    21      lsr: 200.10.10.129:0
  out label:    imp-null lsr: 10.10.10.38:0 inuse
  out label:    24      lsr: 10.10.10.41:0
  out label:    imp-null lsr: 10.10.10.33:0 inuse
10.10.10.32/30
  in label:      19
  out label:    25      lsr: 200.10.10.129:0
  out label:    24      lsr: 10.10.10.38:0
  out label:    23      lsr: 10.10.10.41:0
  out label:    imp-null lsr: 10.10.10.33:0 inuse
10.10.10.36/30
  in label:      25
  out label:    imp-null lsr: 200.10.10.129:0 inuse
  out label:    imp-null lsr: 10.10.10.38:0 inuse
  out label:    25      lsr: 10.10.10.41:0
  out label:    26      lsr: 10.10.10.33:0
10.10.10.40/30
  in label:      21
  out label:    27      lsr: 200.10.10.129:0
  out label:    28      lsr: 10.10.10.38:0
  out label:    imp-null lsr: 10.10.10.41:0 inuse
  out label:    29      lsr: 10.10.10.33:0
200.10.10.0/26
  in label:      18
  out label:    20      lsr: 200.10.10.129:0 inuse
  out label:    23      lsr: 10.10.10.38:0
  out label:    18      lsr: 10.10.10.41:0
  out label:    21      lsr: 10.10.10.33:0
200.10.10.64/26
  in label:      17
  out label:    24      lsr: 200.10.10.129:0
  out label:    27      lsr: 10.10.10.38:0
  out label:    17      lsr: 10.10.10.41:0 inuse
  out label:    27      lsr: 10.10.10.33:0 inuse
200.10.10.128/26
  in label:      16
  out label:    imp-null lsr: 200.10.10.129:0 inuse
  out label:    20      lsr: 10.10.10.38:0
  out label:    16      lsr: 10.10.10.41:0
  out label:    20      lsr: 10.10.10.33:0

```

- **sh mpls ldp bindings**

Traz o conteúdo da base de informações label (LIB).

```

R1
R1#sh mpls ldp bindings
lib entry: 10.10.10.0/30, rev 26
  local binding: label: 24
  remote binding: lsr: 200.10.10.129:0, label: imp-null
  remote binding: lsr: 10.10.10.38:0, label: 22
  remote binding: lsr: 10.10.10.41:0, label: 22
  remote binding: lsr: 10.10.10.33:0, label: 25
lib entry: 10.10.10.4/30, rev 6
  local binding: label: imp-null
  remote binding: lsr: 200.10.10.129:0, label: imp-null
  remote binding: lsr: 10.10.10.38:0, label: 21
  remote binding: lsr: 10.10.10.41:0, label: 21
  remote binding: lsr: 10.10.10.33:0, label: 24
lib entry: 10.10.10.8/30, rev 8
  local binding: label: imp-null
  remote binding: lsr: 200.10.10.129:0, label: 28
  remote binding: lsr: 10.10.10.38:0, label: 26
  remote binding: lsr: 10.10.10.41:0, label: 20
  remote binding: lsr: 10.10.10.33:0, label: imp-null
lib entry: 10.10.10.12/30, rev 22
  local binding: label: 22
  remote binding: lsr: 200.10.10.129:0, label: 26
  remote binding: lsr: 10.10.10.38:0, label: 25
  remote binding: lsr: 10.10.10.41:0, label: imp-null
  remote binding: lsr: 10.10.10.33:0, label: imp-null
lib entry: 10.10.10.16/30, rev 24
  local binding: label: 23
  remote binding: lsr: 200.10.10.129:0, label: 23
  remote binding: lsr: 10.10.10.38:0, label: imp-null
  remote binding: lsr: 10.10.10.41:0, label: imp-null
  remote binding: lsr: 10.10.10.33:0, label: 23
lib entry: 10.10.10.20/30, rev 2
  local binding: label: imp-null
  remote binding: lsr: 200.10.10.129:0, label: 22
  remote binding: lsr: 10.10.10.38:0, label: imp-null
  remote binding: lsr: 10.10.10.41:0, label: 19
  remote binding: lsr: 10.10.10.33:0, label: 22
lib entry: 10.10.10.24/30, rev 4
  local binding: label: imp-null
  remote binding: lsr: 200.10.10.129:0, label: 29
  remote binding: lsr: 10.10.10.38:0, label: 29
--More--

```

```

R1
  remote binding: lsr: 10.10.10.41:0, label: imp-null
  remote binding: lsr: 10.10.10.33:0, label: 28
lib entry: 10.10.10.28/30, rev 18
  local binding: label: 20
  remote binding: lsr: 200.10.10.129:0, label: 21
  remote binding: lsr: 10.10.10.38:0, label: imp-null
  remote binding: lsr: 10.10.10.41:0, label: 24
  remote binding: lsr: 10.10.10.33:0, label: imp-null
lib entry: 10.10.10.32/30, rev 16
  local binding: label: 19
  remote binding: lsr: 200.10.10.129:0, label: 25
  remote binding: lsr: 10.10.10.38:0, label: 24
  remote binding: lsr: 10.10.10.41:0, label: 23
  remote binding: lsr: 10.10.10.33:0, label: imp-null
lib entry: 10.10.10.36/30, rev 28
  local binding: label: 25
  remote binding: lsr: 200.10.10.129:0, label: imp-null
  remote binding: lsr: 10.10.10.38:0, label: imp-null
  remote binding: lsr: 10.10.10.41:0, label: 25
  remote binding: lsr: 10.10.10.33:0, label: 26
lib entry: 10.10.10.40/30, rev 20
  local binding: label: 21
  remote binding: lsr: 200.10.10.129:0, label: 27
  remote binding: lsr: 10.10.10.38:0, label: 28
  remote binding: lsr: 10.10.10.41:0, label: imp-null
  remote binding: lsr: 10.10.10.33:0, label: 29
lib entry: 200.10.10.0/26, rev 14
  local binding: label: 18
  remote binding: lsr: 200.10.10.129:0, label: 20
  remote binding: lsr: 10.10.10.38:0, label: 23
  remote binding: lsr: 10.10.10.41:0, label: 18
  remote binding: lsr: 10.10.10.33:0, label: 21
lib entry: 200.10.10.64/26, rev 12
  local binding: label: 17
  remote binding: lsr: 200.10.10.129:0, label: 24
  remote binding: lsr: 10.10.10.38:0, label: 27
  remote binding: lsr: 10.10.10.41:0, label: 17
  remote binding: lsr: 10.10.10.33:0, label: 27
lib entry: 200.10.10.128/26, rev 10
  local binding: label: 16
  remote binding: lsr: 200.10.10.129:0, label: imp-null
--More--

```

```

R1
  remote binding: lsr: 10.10.10.38:0, label: 20
  remote binding: lsr: 10.10.10.41:0, label: 16
  remote binding: lsr: 10.10.10.33:0, label: 20
R1#
R1#
R1#

```

- **sh mpls ldp neighbor**

Traz informações das interfaces que possuem o MPLS configurados nelas. Abaixo, por exemplo, na interface GigabitEthernet 1/0 deste Roteador R1 é possível conhecer os vizinhos: 10.10.10.17, 10.10.10.22, 10.10.10.29 e 10.10.10.38, que no caso pertencem ao Roteador R3.

```

R1#sh mpls ldp neighbor
Peer LDP Ident: 200.10.10.129:0; Local LDP Ident 10.10.10.25:0
TCP connection: 200.10.10.129.46972 - 10.10.10.25.646
State: Oper; Msgs sent/rcvd: 77/78; Downstream
Up time: 00:53:16
LDP discovery sources:
  GigabitEthernet3/0, Src IP addr: 10.10.10.5
Addresses bound to peer LDP Ident:
  200.10.10.129 10.10.10.37 10.10.10.2 10.10.10.5
Peer LDP Ident: 10.10.10.38:0; Local LDP Ident 10.10.10.25:0
TCP connection: 10.10.10.38.34239 - 10.10.10.25.646
State: Oper; Msgs sent/rcvd: 77/77; Downstream
Up time: 00:53:15
LDP discovery sources:
  GigabitEthernet1/0, Src IP addr: 10.10.10.22
Addresses bound to peer LDP Ident:
  10.10.10.22 10.10.10.29 10.10.10.38 10.10.10.17
Peer LDP Ident: 10.10.10.41:0; Local LDP Ident 10.10.10.25:0
TCP connection: 10.10.10.41.25651 - 10.10.10.25.646
State: Oper; Msgs sent/rcvd: 78/79; Downstream
Up time: 00:53:15
LDP discovery sources:
  GigabitEthernet2/0, Src IP addr: 10.10.10.26
Addresses bound to peer LDP Ident:
  10.10.10.14 10.10.10.26 10.10.10.41 10.10.10.18
Peer LDP Ident: 10.10.10.33:0; Local LDP Ident 10.10.10.25:0
TCP connection: 10.10.10.33.26150 - 10.10.10.25.646
State: Oper; Msgs sent/rcvd: 78/77; Downstream
Up time: 00:53:25
LDP discovery sources:
  GigabitEthernet4/0, Src IP addr: 10.10.10.10
Addresses bound to peer LDP Ident:
  10.10.10.13 10.10.10.30 10.10.10.33 10.10.10.10

```

- **sh mpls forwarding-table**

Traz os números, tags ou etiquetas MPLS dados aos pacotes com seus respectivos destinos às redes precedidas de suas máscaras /26 ou /30 e o próximo salto.

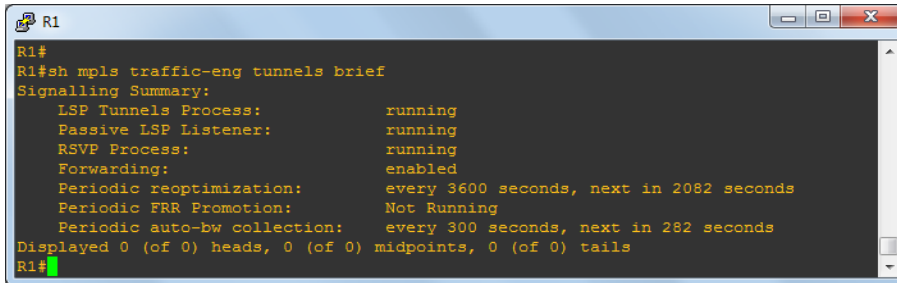
```

R1#sh mpls forwarding-table
Local   Outgoing  Prefix          Bytes Label  Outgoing  Next Hop
Label   Label    or Tunnel Id   Switched     interface
16      Pop Label 200.10.10.128/26 0           Gi3/0      10.10.10.5
17      Pop Label 200.10.10.64/26 0           Gi4/0      10.10.10.10
        Pop Label 200.10.10.64/26 0           Gi2/0      10.10.10.26
18      Pop Label 200.10.10.0/26 0           Gi3/0      10.10.10.5
19      Pop Label 10.10.10.32/30 0           Gi4/0      10.10.10.10
20      Pop Label 10.10.10.28/30 0           Gi4/0      10.10.10.10
        Pop Label 10.10.10.28/30 0           Gi1/0      10.10.10.22
21      Pop Label 10.10.10.40/30 0           Gi2/0      10.10.10.26
22      Pop Label 10.10.10.12/30 0           Gi4/0      10.10.10.10
        Pop Label 10.10.10.12/30 0           Gi2/0      10.10.10.26
23      Pop Label 10.10.10.16/30 0           Gi1/0      10.10.10.22
        Pop Label 10.10.10.16/30 0           Gi2/0      10.10.10.26
24      Pop Label 10.10.10.0/30 0           Gi3/0      10.10.10.5
25      Pop Label 10.10.10.36/30 0           Gi3/0      10.10.10.5
        Pop Label 10.10.10.36/30 0           Gi1/0      10.10.10.22
R1#

```

- **sh mpls traffic-eng tunnels brief**

Traz informações sobre os túneis MPLS.

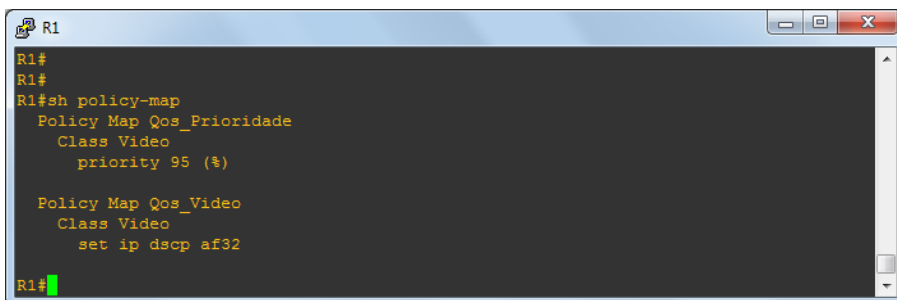


```

R1#
R1#sh mpls traffic-eng tunnels brief
Signalling Summary:
  LSP Tunnels Process:      running
  Passive LSP Listener:    running
  RSVP Process:            running
  Forwarding:              enabled
  Periodic reoptimization: every 3600 seconds, next in 2082 seconds
  Periodic FRR Promotion:  Not Running
  Periodic auto-bw collection: every 300 seconds, next in 282 seconds
Displayed 0 (of 0) heads, 0 (of 0) midpoints, 0 (of 0) tails
R1#
  
```

- **sh policy-map**

Traz a política aplicada para QoS – TE. O “Qos_Prioridade” representa a reserva de banda de 95% de toda a banda para esta prioridade, usando DSCP (CS3), que corresponde ao tráfego de vídeo.



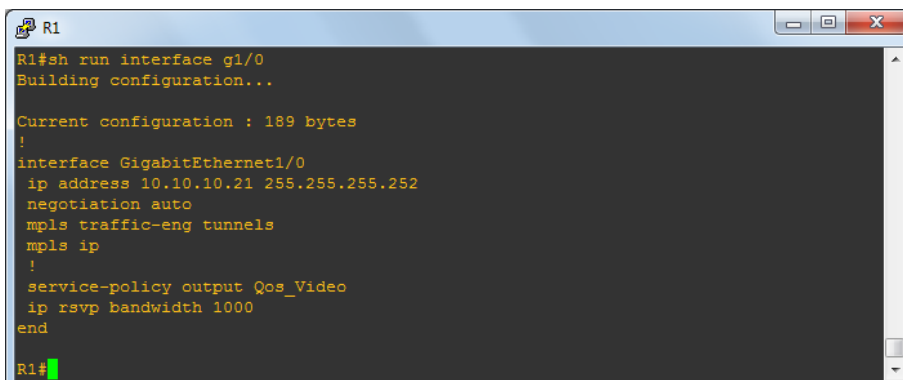
```

R1#
R1#
R1#sh policy-map
  Policy Map Qos_Prioridade
    Class Video
      priority 95 (%)

  Policy Map Qos_Video
    Class Video
      set ip dscp af32
R1#
  
```

- **sh run interface g1/0**

Traz informações relacionadas ao que está rodando em determinada interface.



```

R1#
R1#sh run interface g1/0
Building configuration...

Current configuration : 189 bytes
!
interface GigabitEthernet1/0
 ip address 10.10.10.21 255.255.255.252
 negotiation auto
 mpls traffic-eng tunnels
 mpls ip
!
 service-policy output Qos_Video
 ip rsvp bandwidth 1000
end
R1#
  
```