UNIVERSIDADE ESTADUAL DO MARANHÃO - UEMA
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DE COMPUTAÇÃO

JULIANA DOS SANTOS VIANA

PROVIMENTO DE QOS EM REDES IP COM APLICAÇÃO DAS TÉCNICAS DE SERVIÇOS DIFERENCIADOS

São Luís
2017

JULIANA DOS SANTOS VIANA

PROVIMENTO DE QOS EM REDES IP COM APLICAÇÃO DAS TÉCNICAS DE SERVIÇOS DIFERENCIADOS

Monografia apresentada à Banca Examinadora do Curso de Engenharia de Computação da Universidade Estadual do Maranhão para obtenção do título de Engenheiro de Computação.

Orientador: PROF. WESLEY BATISTA DOMINICES DE ARAUJO

São Luís
2017

JULIANA DOS SANTOS VIANA

PROVIMENTO DE QOS EM REDES IP COM APLICAÇÃO DAS TÉCNICAS DE SERVIÇOS DIFERENCIADOS

Monografia apresentada à Banca Examinadora do Curso de Engenharia de Computação da Universidade Estadual do Maranhão para obtenção do título de Engenheiro de Computação.

Aprovada em 05 de Julho de 2017

_____

Prof. Wesley Batista Dominices de Araujo (Orientador)
Departamento de Engenharia de Computação
Universidade Estadual do Maranhão

_____

Prof. Leonardo Henrique Gonsioroski Furtado da Silva (1º membro)
Departamento de Engenharia de Computação
Universidade Estadual do Maranhão

_____

Prof. Antonio Fernando Lavareda Jacob Junior (2º membro)
Departamento de Engenharia de Computação
Universidade Estadual do Maranhão

DEDICATÓRIA

Dedico este trabalho primeiramente a Deus, por ser essencial em minha vida, autor de meu destino, aos meus pais Julião e Mirian, meu irmão João Marcus e meu esposo Alessandro.

AGRADECIMENTOS

Primeiramente agradeço a Deus por ter me sustentado até aqui. Por Sua graça infinita que me alcançou. Aos meus pais Julião e Mirian, que nos momentos de dificuldade e desânimo estiveram ao meu lado dando amor, incentivo e apoio incondicional. Ao meu irmão João Marcus, prestando apoio e sempre preocupado com o avanço do meu trabalho. Ao meu esposo Alessandro, pelo amor, compreensão e apoio moral no decorrer do desenvolvimento deste trabalho. Aos meus irmãos em Cristo, da congregação Assembleia de Deus área 25, por terem compreendido minha ausência nos dias de correria. Agradeço ao meu amigo e irmão Lucas Dias pelas palavras de incentivo. Ao professor Wesley Dominices pela paciência e esforço que desempenhou na orientação deste trabalho, sempre preocupado como melhoramento do mesmo. Agradeço ao corpo docente do curso de Engenharia da Computação da Universidade Estadual do Maranhão, por colaborarem para a minha formação profissional. Agradeço ao Thiago pelo apoio ao compreender minha ausência em alguns turnos do serviço para concluir este trabalho dentro do prazo. Agradeço às minhas primas e amigas Hildelena e Ana Patrícia pelas palavras de incentivo e por torcerem por mim. Quero agradecer aos meus amigos, cito aqui o nome de alguns: Arianne, Andressa, Rayssa, Vanessa e Stephanne que me acompanham desde o início da jornada EngComp.

“Se algum de vocês tem falta de sabedoria, peça-a a Deus, que a todos dá livremente, de boa vontade; e lhe será concedida. “

(Bíblia Sagrada – Tiago 1,5)

RESUMO

O avanço das redes de computadores gerou a necessidade de desenvolvimento de novos serviços que proveem qualidade nos fluxos de dados. Nesse contexto, este trabalho aborda a qualidade de serviço (*Quality of Service* – QoS), em redes TCP/IP. O presente trabalho apresenta uma busca aprofundada sobre a importância da aplicação de novas tecnologias que suportem a grande demanda de serviço no universo das redes de computadores, com uma visão especial na qualidade de serviço (QoS) aplicada através das técnicas dos Serviços Diferenciados (*Differentiated Services – DiffServ*), fazendo uma abordagem teórica, descrevendo e investigando a aplicação de técnicas a fim de verificar a eficácia do serviço. Para que se obtenha uma garantia de que as redes de computadores funcionem corretamente, é necessário aplicar tecnologias que permitam atingir um nível de tráfego satisfatório e confiável para aplicações e dados e que determinados níveis de desempenho sejam garantidos por intermédio de uma política capaz de estabelecer métricas e de caracterizar e descrever o comportamento da rede no que diz respeito a sua utilização e performance. No estudo proposto, foram realizadas duas simulações, uma utilizando a ferramenta GNS3, que é um simulador que permite emular o IOS (*Internetwork Operating System*) e a outra simulação utilizando equipamento reais para produzir conhecimento adquirido na elaboração de projeto de simulação de rede, considerando a importância da modelagem e simulação no planejamento e implementação de redes.

Palavras-chave: Qualidade de Serviço; *DiffServ*; Redes de Computadores; TCP/IP.

ABSTRACT

The advancement of computer networks has generated the need of development of new services that provide quality in data flows. In this context, this paper addresses Quality of Service (QoS) in TCP / IP networks. The present paper presents an in - depth search about the importance of the application of new technologies that support the great demand of service in the universe of computer networks, with a special vision in the quality of service (QoS) applied through the Differentiated Services techniques - DiffServ, making a theoretical approach, describing and investigating the application of techniques in order to verify the effectiveness of the service. In order to ensure that computer networks work properly, it is necessary to apply technologies that allow the achievement of a satisfactory and reliable level of traffic to applications and data and that certain levels of performance are guaranteed through a policy capable of establishing metrics and to characterize and describe the behavior of the network with respect to its use and performance. In this study, two simulations were performed, one using the GNS3 tool, which is a simulator that allows the emulation of the Internetwork Operating System (IOS) and the other simulation using real equipment to produce knowledge acquired in the elaboration of the network simulation project, considering the importance of modeling and simulation in the planning and implementation of networks.

Keywords: *Quality of Services*; *DiffServ*; *Computer Networks*; *TCP/IP*.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE SIGLAS

AF	*Assured Forwarding*
ARPA	*Advanced Research Projects Agency*
ATM	*Asynchronous Transfer Mode*
BA	*Behavior Aggregate*
BE	*Best Effort*
BECN	*Backward Explicit Congestion Notification*
CBWFQ	*Class Based Weighted Fair Queueing*
CoS	*Class of Service*
CS	*Class Selector*
CU	C*urrently Unused*
DoD	*Department of Defense*
DS	*Differentiated Services*
DSCP	*Differentiated Services Code Point*
ECN	*Explicit Congestion Notification*
EF	*Expedited Forwarding*
FECN	*Forward Explicit Congestion Notification*
FIFO	*First In First Out*
FTP	*File Transfer Protocol*
GTS	*Generic Traffic Shaping*
IETF	*Internet Engineering Task Force*
ITU-T	*International Telecommunication Union - Telecommunications section*
LAN	*Local Area Network*
LLQ	*Low Latency Queueing*
MAC	*Media Access Control*
MF	*Multi-Field*
OSI	*Open System Interconnection*
PCM	*Pulse-code modulation*
PHB	*Per-Hop-Behavior*
PQ	*Priority Queueing*
QOS	*Quality of Service*
RED	*Random Early Detection*
RFC	*Request for Comments*
RSVP	*Resource Reservation Protocol*
SLA	*Service Level Agreement*
SMDS	*Switched Multimegabit Data Service*
SMTP	*Simple Mail Transfer Protocol*
SNMP	*Simple Network Management Protocol*
TCP	*Transmission Control Protocol*
TCP/IP	*Transmission Control Protocol / Internet Protocol*
TFTP	*Trivial File Transfer Protocol*
TOS	*Type of Service*
UDP	*User Datagram Protocol*
VoIP	*Voice over IP*
VoPF	*Voice over Frame Relay*
VPCS	*Virtual PC Simulator*
WAN	*Wide Area Network*
WFQ	*Weighted Fair Queueing*
WRED	*Weighted Random Early Detection*

SUMÁRIO

1 INTRODUÇÃO

O crescimento exponencial da Internet ao longo do tempo vem exigindo cada vez mais dos provedores de acesso. Não houve o aumento somente na quantidade de usuários, mas consequentemente a velocidade de transmissão de dados, voz e vídeo, aumentaram em grande escala. Esse aumento de serviço demanda tanto em largura de banda, quanto em espaço, volume de tráfego nos *backbones*, qualidades de serviço (QoS - *Quality of Service*), engenharia de tráfego e aplicações. Algumas aplicações de dados como áudio e vídeo têm exigências rígidas quanto ao atraso fim-a-fim, porém menos sensíveis a perdas mínimas de pacotes; outras, como transferências de arquivos e correio eletrônico, são insensíveis ao retardo, porém, mais sensíveis à perda de pacotes [STALLINGS, 2004].

O protocolo de camada de rede do modelo OSI (*Open System Interconnection*), IP, presta um serviço de melhor esforço (*best effort*). Isso significa que o serviço presta seu melhor esforço para transportar cada datagrama da fonte ao destino o mais rápido possível [KUROSE; ROSS, 2010]. No melhor esforço, cada usuário compartilha largura de banda com outros e, portanto, a transmissão de seus dados concorre com as transmissões dos demais usuários. Os dados empacotados são encaminhados da melhor forma possível, conforme as rotas e banda disponíveis. Quando há congestionamento, os pacotes são descartados sem distinção. Não há garantia de que o serviço será realizado com sucesso. Entretanto, aplicações como voz sobre IP e videoconferência necessitam de tais garantias.

Uma definição para Qualidade de Serviço (QoS) é dada pela recomendação I.350 do ITU-T (*International Telecommunication Union - Telecommunications section*), a partir da recomendação E.800, onde define-se a Qualidade de Serviço como sendo o efeito coletivo provocado pelas características de desempenho de um serviço, determinando o grau de satisfação do usuário, ou seja, a QoS pode ser definida como o conjunto de características de um sistema necessário para atingir uma determinada funcionalidade.

Qualidade de serviço pode ser descrita ainda como um conjunto de parâmetros que descrevem a qualidade de um fluxo de dados específico, por exemplo, largura de banda, prioridades, etc. e tem por objetivo fornecer serviço de rede melhor e mais previsível, fornecendo largura de banda dedicada, *jitter* controlado e latência. QoS atinge esses objetivos, fornecendo ferramentas para gerenciar o congestionamento da rede, formação de rede tráfego, utilizando-se de maneira ampla área de links de forma mais eficiente, e definindo políticas de tráfego em toda a rede.

A QoS oferece serviços de rede inteligente que, quando corretamente aplicados, ajudam a fornecer desempenho consistente e previsível [CISCO SYSTEMS, 2009]. Este conceito serve para comprovar o quanto a Qualidade de Serviço 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 usuários-sistema.

Adentrando ao assunto de Qualidade de Serviço, este trabalho observa com mais afinco os efeitos causados após a aplicação das Técnicas de QoS com ênfase nos serviços diferenciados. A arquitetura Diffserv [RFC 2475] tem como objetivo fornecer um serviço diferenciado – isto é, a habilidade de lidar com as diferentes "classes" de tráfego de diferentes modos na internet – e fazê-lo de modo escalável e flexível [KUROSE; ROSS, 2010]. É uma abordagem mais simples para oferecer qualidade de serviço, uma estratégia que pode ser implementada em grande parte no local em cada roteador, sem configuração antecipada e sem ter de envolver todo o caminho [TANENBAUM; WETHERALL, 2011].

Esse trabalho tem por finalidade verificar a eficiência da utilização dos mecanismos de QoS aqui estudados, mostrando a viabilidade de sua utilização para realizar uma pesquisa aplicada, uma vez que utilizará conhecimento da pesquisa básica para resolver problemas. Para um melhor tratamento dos objetivos e melhor apreciação desta pesquisa, observou-se que ela é classificada como pesquisa explicativa. Detectou-se também a necessidade da pesquisa bibliográfica no momento em que se fez uso de materiais já elaborados: livros, artigos científicos, revistas, documentos eletrônicos e enciclopédias na busca e alocação de conhecimento sobre a tecnologia rede computadores, correlacionando tal conhecimento com abordagens já trabalhadas por outros autores.

A pesquisa assume como pesquisa bibliográfica, o que proporciona maior familiaridade com o problema, tornando-o explícito ou construindo hipóteses sobre ele através de principalmente do levantamento bibliográfico. Por ser um tipo de pesquisa muito específica, quase sempre ela assume a forma de um estudo de caso [GIL, 2008].

O problema foi direcionando a pesquisa para as áreas de Qualidade de Serviço em especial a arquitetura dos Serviços Diferenciados (*Differentiated Services*) no que diz respeito às conexões de redes de computadores. Além disso, este trabalho traz como complemento a simulação de alguns dos processos de QoS citados, tanto em ambiente computacional como em ambiente real.

A simulação é uma técnica utilizada com muita frequência, pela flexibilidade em testar cenários variados, incluindo o comportamento de protocolos e novas tecnologias e efeito de

diferentes topologias [KAMIENSKI,2002], tornando-se uma técnica importante de avaliação de soluções existentes ou emergentes, pois dá suporte a modelagem de um ambiente mais próximo do mundo real incorporando seus métodos e características, e por meio desse, pode-se obter mais detalhes como: a possibilidade de avaliação de mais cenários e análise de desempenho com diferentes níveis de detalhe e escalas de tempo de observação a um custo e tempo razoavelmente pequeno comparado com testes em ambientes físicos. Consequentemente, a chance de sucesso da avaliação estar coerente com a realidade é maior.

Por muito tempo, as redes por comutação de pacotes ofereceram a promessa de dar suporte a aplicações de multimídia, ou seja, aquelas que combinam áudio, vídeo e dados. Afinal, uma vez digitalizadas, as informações de áudio e vídeo tornam-se como qualquer outra forma de dados – um fluxo de bits a serem transmitidos.

Um obstáculo para a realização dessa promessa tem sido a necessidade de enlaces com maior largura de banda. Recentemente, porém, melhorias na codificação reduziram as necessidades de largura de banda das aplicações de áudio e vídeo, enquanto ao mesmo tempo as velocidades de enlace aumentaram [PETERSON; DAVIE, 2013]. Entretanto, não somente o aumento na largura de banda é suficiente para suprir os requisitos de uma rede com bom desempenho.

O usuário exige cada vez mais da rede. Assim, a prontidão da entrega pode ser muito importante. Referimo-nos a aplicações que são sensíveis à prontidão dos dados como aplicações de tempo real, ou seja, aquelas que precisam de garantia da rede de que seus dados provavelmente chegarão prontamente a tempo. Embora uma aplicação que não seja de tempo real possa usar uma estratégia de retransmissão fim a fim para certificar-se de que os dados chegarão corretamente, tal estratégia não é capaz de oferecer prontidão. Se os dados chegarem tarde, a retransmissão só aumentará a latência total. A chegada a tempo precisa ser fornecida pela própria rede (os roteadores), e não apenas nas bordas da rede (os hosts).

> Portanto, concluímos que o modelo de melhor esforço, em que a rede tenta entregar seus dados, mas não faz promessas e deixa a operação de recuperação da perda para as bordas, não é suficiente para aplicações de tempo real. O que precisamos é de um novo modelo de serviço, em que as aplicações que precisam de garantias maiores possam pedi-las à rede. (PETERSON; DAVIE, 2013)

Identificando as necessidades da rede, observa-se que é necessário um modelo de serviço mais robusto, que atenda às necessidades de cada aplicação. Com a implantação de qualidade de serviço (*Quality of Service* – QoS), é possível oferecer maior garantia e segurança para aplicações avançadas, uma vez que o tráfego destas aplicações passa a ter prioridade em relação a aplicações tradicionais. Para isto, existem técnicas desenvolvidas que

oferecem uma grande variedade de qualidade de serviço. Essas técnicas podem ser divididas em duas categorias gerais:

- Técnicas de alta granularidade, que garante QoS a aplicações ou fluxos individuais, nesta categoria encontram-se os Serviços Integrados (*Integrated Services*), onde é associada com o RSVP *(Resource Reservation Protocol* – protocolo de reserva de recursos). Antes de iniciar a transmissão de dados, as aplicações devem usar mecanismos de reserva de recursos na rede.
- Técnicas de baixa granularidade, que garante QoS a grandes classes de dados ou tráfego, nesta categoria podem ser encontrados os Serviços Diferenciados (*Differentiated Services*), onde a QoS é garantida através de mecanismos de priorização de pacotes na rede, onde os pacotes são marcados diferentemente para criar classes de serviços que recebam tratamento diferenciado.

Este trabalho terá a aplicação das técnicas de Qualidade de Serviço com enfoque nos Serviços Diferenciados.

O que impulsionou a realização deste trabalho foi o fato de que a Qualidade de Serviço (QoS) é uma tecnologia que proporciona uma diferenciação de serviços, otimizando processos e garantindo níveis de qualidade. O objetivo do trabalho é apresentar uma busca aprofundada sobre a importância da aplicação o QoS nas redes IP e utilizar a ferramentas para produzir conhecimento adquirido na elaboração de projeto de simulação de redes e demonstrar recursos disponíveis, a fim de caracterizar e descrever o comportamento da rede no que diz respeito a sua utilização e performance.

1.1 Organização do Trabalho

Os 5 capítulos deste trabalho, de forma sucinta, apresentam os seguintes conteúdos:

Capítulo 1: uma breve introdução contextualizando a QoS (Qualidade de Serviço), objeto de estudo principal, com os ambientes onde ela é aplicada e o objetivo do trabalho.

Capítulo 2: Neste capítulo compõe a fundamentação teórica, que aborda um breve histórico dos modelos de referência OSI e TCP/IP, requisitos e características da qualidade de serviço e suas alternativas e mecanismos para sua implantação.

Capítulo 3: Este capítulo apresenta duas simulações de uma rede com provimento de QoS, uma simulação em um ambiente virtual e a outra em um ambiente real. Também são apresentados a topologia da rede e configurações.

Capítulo 4: O capítulo 4 traz os resultados obtidos após os experimentos descritos no capítulo 3.

Capítulo 5: Apresenta as considerações finais e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Modelo de referência OSI

Modelo de referência OSI, tem como principal objetivo ser um modelo padrão para protocolos de comunicação entre diversos tipos de sistema, garantindo a comunicação fim-a-fim. O Modelo OSI (em inglês *Open Systems Interconnection*) foi lançado em 1984 pela Organização Internacional para a Normalização (em inglês *International Organization for Standardization*).

Trata-se de um modelo de redes de computadores organizado em sete camadas para servir de padrão, para protocolos de comunicação entre as mais diversas tecnologias, garantindo a comunicação entre eles. Cada protocolo realiza a inserção de uma funcionalidade em uma camada específica. As camadas do Modelo OSI serão descritas sucintamente na Tabela 1.

Tabela 1 - Camadas do Modelo OSI

7	Aplicação	Esta camada funciona como uma interface de ligação entre os processos de comunicação de rede e as aplicações utilizadas pelo usuário.
6	Apresentação	Aqui os dados são convertidos e garantidos em um formato universal, preocupando-se com a sintaxe e semântica das informações transmitidas.
5	Sessão	A camada de sessão permite que os usuários de diferentes máquinas estabeleçam sessões entre eles.
4	Transporte	Sua função básica é efetuar processos de sequenciamento e, quando for o caso, confirmar recebimento de dados
3	Rede	Roteamento dos dados através da rede é implementado nesta camada
2	Enlace	Principal tarefa é transformar um canal de transmissão bruta em uma linha que pareça livre de erros de transmissão não detectados para a camada de rede.
1	Física	Responsável pela transmissão de bits por um canal de comunicação

Fonte: Tanenbaum, 2010

Cada camada comunica-se com sua equivalente em outro computador. Quando a informação é passada de uma camada para outra inferior, um cabeçalho é adicionado aos

dados para indicar de onde a informação vem e para onde vai. O bloco do cabeçalho somado aos dados de uma camada estarão localizados na área de dados da próxima camada.

A unidade de informação (PDU - *Protocol Data Unit*) muda de nome ao longo das camadas, de maneira que se pode saber sobre qual camada se está referindo pelos nomes destas unidades. A PDU das camadas de Aplicação, Apresentação e Sessão recebe o nome de dados, a PDU da camada de Transporte é nomeada de segmentos, a PDU da camada de Rede é chamada de pacotes, a PDU da camada de Enlace são os quadros, a PDU da camada Física são os bits. A Figura 1 relaciona os diversos nomes destas unidades de informação ao longo das camadas.

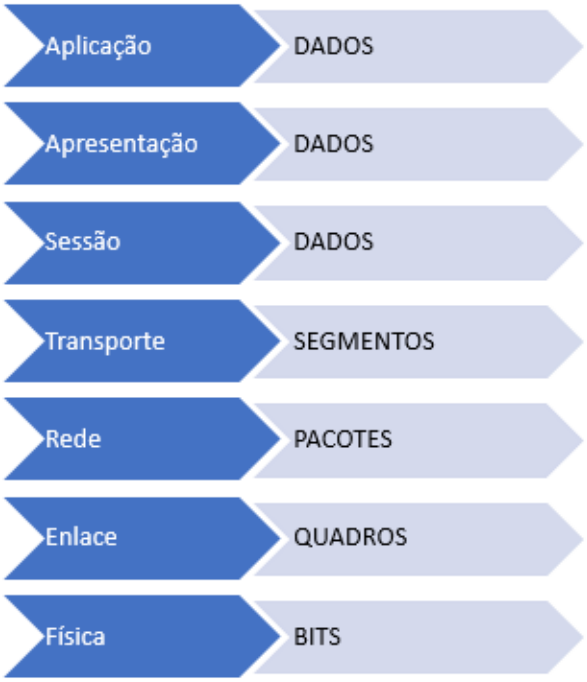

Figura 1 - Unidades de informação

Fonte: KUROSE; ROSS, 2010

2.2 Modelo de referência TCP/IP

Nos anos 60, o Departamento de Defesa dos Estados Unidos (DoD), se interessou em uma arquitetura de redes que estava sendo desenvolvida pelas universidades para interligação dos seus sistemas computacionais e que utilizava a tecnologia de chaveamento de pacotes. O interesse do DoD estava no desejo de manter a comunicação entre os diversos sistemas espalhados pelo mundo, no caso de um desastre nuclear. O problema maior estava na compatibilidade entre os sistemas computacionais de diferentes fabricantes que possuíam

diferentes sistemas operacionais, topologias e protocolos. A integração e compartilhamento dos dados passou a ser um problema de difícil resolução (CBPF-NT-004,2000).

Foi atribuída assim à ARPA (*Advanced Research Projects Agency)* a missão de encontrar uma solução para este problema de tratar com diferentes equipamentos e diferentes características computacionais. Partindo daí, foi proposto um acordo entre universidades e fabricantes para o desenvolvimento de padrões de comunicação. Este acordo descreveu e construiu uma rede de teste de quatro nós, chamada ARPANET, e que acabou sendo a origem da internet, tal qual conhecemos hoje.

Findando os anos 70, esta rede inicial evoluiu, tendo seu modelo principal desenvolvido e transformado para a pilha de protocolos TCP/IP (*Transmission Control Protocol / Internet Protocol*). A aceitação mundial dos conjuntos de protocolos TCP/IP deveu-se principalmente a versão UNIX de Berkeley que além de incluir estes protocolos, colocava-os em uma situação de domínio público, onde qualquer organização, através de sua equipe técnica poderia modificá-los e assim garantir seu desenvolvimento.

Um dos destaques desta evolução foi a IETF (*Internet Engineering Task Force)*, cuja principal função é a manutenção e apoio aos padrões da Internet e TCP/IP, principalmente através da série de documentos RFC (*Request for Comments)*. Estes documentos descrevem as diversas tecnologias envolvidas e servem de base para as novas tecnologias que deverão manter a compatibilidade com as anteriores.

Em resumo, o maior trunfo do TCP/IP é o fato destes protocolos apresentarem a interoperabilidade de comunicação entre todos os tipos de hardware e todos os tipos de sistemas operacionais. Sendo assim, o impacto positivo da comunicação computacional aumenta com o número de tipos computadores que participam da grande rede Internet.

O modelo TCP/IP é mais voltado para a parte prática, do que realmente estava sendo utilizado pelo mercado de trabalho, enquanto o Modelo OSI, ficou apenas como um modelo teórico. O TCP/IP tem apenas quatro camadas em contraposição do Modelo OSI, que tem sete camadas. A Figura 2 faz a comparação dos dois modelos de referência.

Na última coluna da Figura 2, tem-se o modelo TCP/IP. Este é constituído por quatro camadas. A camada superior (camada de aplicação) é responsável por permitir que aplicações possam se comunicar através de hardware e software de diferentes sistemas operacionais e plataformas. Vários protocolos se destacam nesta camada: SMTP, FTP, SNMP, DHCP, HTTP, etc.

A camada a seguir é a de Transporte, tem a função principal de transporte fim-a-fim, começar e terminar uma conexão, controlar o fluxo de dados e de efetuar processos de correção e verificação de erros. Os principais Protocolos desta camada são o TCP e o UDP.

A camada de Internet é a responsável pelo roteamento. Comparativamente ela corresponde no modelo OSI à camada de Rede. Esta camada é usada para atribuir endereço de rede lógico (IP) ao sistema e rotear a informação para a rede correta. Tem ainda a função de ligação entre as camadas superiores e os protocolos de Acesso à Rede.

A primeira camada, Acesso à Rede, refere-se a união das funcionalidades da camada Física e de Enlace do modelo OSI, que contempla a parte física, bits, quadros, de protocolos de Enlace, tais como MAC, PPP, etc....

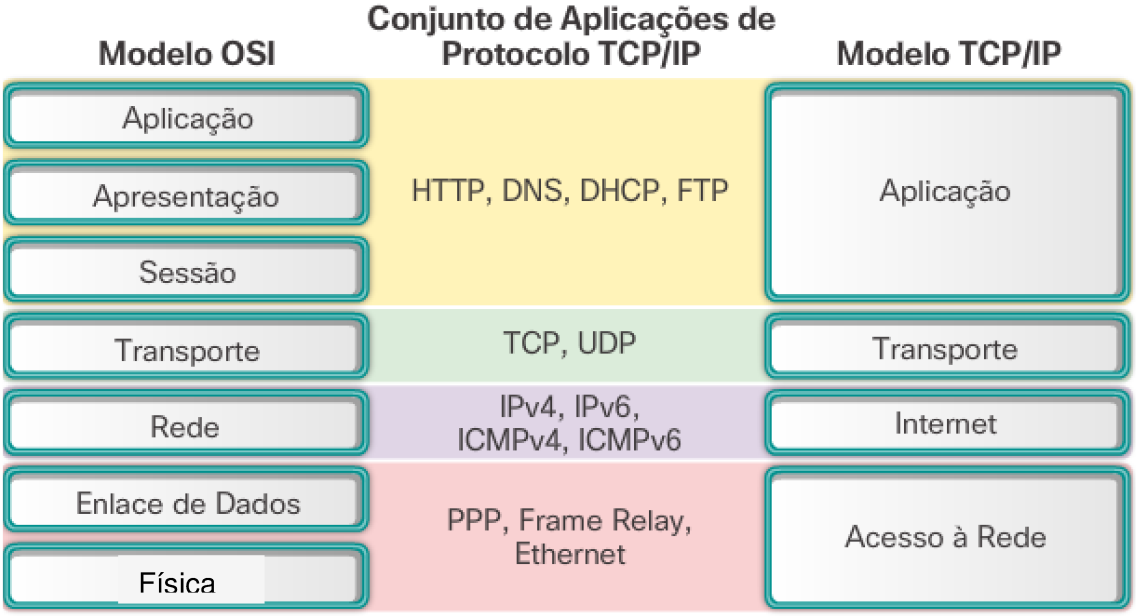

Figura 2 - Comparação das camadas dos dois modelos de referência

Fonte: Cisco CCNA 1 –Introdução às Redes, 2017

2.3 Qualidade de Serviço – QoS

Atualmente, a internet vem sendo utilizada cada vez mais como meio de transporte de informação multimídia, como imagens, voz e vídeo. Por conta disto, a internet tende a receber uma maior quantidade de tráfego. O tráfego gerado pelas aplicações multimídia possui alguns requisitos de Qualidade de Serviço (QoS) não encontrados na maioria das aplicações convencionais [DUTKIEWICZ; BOUSTEAD, 2002].

Uma rede comutada por pacotes pode sofrer vários problemas, relacionados a várias características inerentes da própria tecnologia da rede, como uso da infraestrutura da rede

compartilhada, a forma como os tráfegos de dados passam na rede, entres outros elementos que podem interferir nessa rede.

A transferência de informação com características restringentes em relação a atraso, variação do atraso e taxa de perdas não é o cenário para o qual o protocolo IP foi projetado. Atualmente a arquitetura de rede na Internet foi projetada para o envio da informação usando o modelo de serviço de melhor esforço, ou seja, sem qualquer garantia de QoS [KUROSE; ROSS, 2010]. Logo se faz necessário o suporte de tráfego em tempo real na Internet, com suporte de mecanismos de sinalização para que as aplicações indiquem seus requisitos de Qualidade de Serviço à rede.

O controle de QoS possibilita que a prioridade de alguns tipos de tráfegos seja maior que de outros tráfegos. Depois da classificação, o tráfego com a prioridade mais alta pode ser enviado primeiro, enquanto o tráfego com menor prioridade é enfileirado. A QoS tem como propósito principal determinar qual tráfego deve receber prioridade de acesso ao link, já que o mesmo é implantado para evitar que os dados saturem um *link*, a ponto de outros dados não poderem obter acesso a ele.

De acordo com Maia (2009) a QoS permite que a rede ofereça serviços que atendam às expectativas dos usuários a partir de parâmetros previamente negociados entre o transmissor e a rede de interconexão, como se fosse estabelecido um contrato entre as partes. Para Maia, a utilização de parâmetros para especificar a QoS desejada para uma determinada aplicação pode ser comparada às diversas modalidades para o envio de uma carta pelo correio.

Quando se deseja enviar uma correspondência, o usuário pode, por exemplo, especificar o tempo máximo para que a carta chegue ao seu destino ou exigir do destinatário a confirmação do recebimento (carta registrada). Por outro lado, se o usuário não especificar o nível de serviço desejado, a carta será encaminhada da melhor maneira possível, sem garantia de prazo ou entrega.

Segundo Richter e Meer (1998), um conceito importante para a abordagem da QoS é a definição de fluxo. Um fluxo pode ser definido como uma sequência de dados pertencentes a duas aplicações, uma na origem e outra no destino, que se comunicam. Por exemplo, em uma conversa telefônica entre duas pessoas utilizando uma rede IP, o fluxo seria os pacotes contendo a conversa entre os dois usuários. Os mecanismos de QoS visam garantir a qualidade de transmissão de um determinado fluxo ou um conjunto de fluxos com necessidades semelhantes.

Os parâmetros para a qualidade de serviço são características inerentes ao projeto de rede, aplicação ou serviço. Sendo que por meio dos valores desses parâmetros pode-se

verificar se a QoS está sendo atendida [TANENBAUM; WETHERALL, 2011]. Segundo Tanenbaum (2011), os parâmetros de QoS fim-a-fim devem ser escolhidos de acordo com o ambiente e o tipo de serviço contratado e devem estar dentro de limites bem definidos representados por valores mínimos e máximos aceitáveis para um determinado serviço, que garantam o nível de QoS.

2.3.1 Requisitos para obter Qualidade de Serviço

Partindo dos parâmetros de qualidade de serviço, é possível que o provedor de acesso e o usuário firmem um acordo de nível de serviço expresso e solicitado em termos de uma "Solicitação de Serviço" ou "Contrato de Serviço", chamado SLA (*Service Level Agreement*), o qual permite ao usuário especificar a qualidade do serviço contratado. O SLA permite definir as características do fluxo que será transmitido, criando a ideia de perfil de tráfego [RFC 2475].

O SLA deve definir claramente os níveis mínimos de desempenho que um provedor de serviços deverá manter a disposição do usuário para que as aplicações executem com qualidade e o não cumprimento desse acordo implica em penalidades, estipuladas em contrato. Um exemplo típico de SLA para uma aplicação de voz sobre IP (VoIP - *Voice over IP*) com algumas centenas de canais voz simultâneos numa rede IP WAN poderia ser:

- Vazão $\geq$ 2 Mbps;
- Atraso $\leq$ 250 mseg;
- Disponibilidade $\geq$ 99%.

Uma vez que a rede garanta este SLA, tem-se como resultado que a aplicação VoIP em questão poderá executar garantindo a qualidade de voz prevista para os seus usuários se comunicando simultaneamente através da rede IP.

Segundo Gabos e Carvalho (2009), as aplicações variam em suas exigências sobre os parâmetros da Qualidade de Serviço o qual as redes convergentes precisam acomodar e integrar todas essas exigências. Segundo Farrel (2005), além do encaminhamento de pacotes, os roteadores possuem muito mais utilidades, sendo papel fundamental na implementação de QoS na rede, enfileirando cada pacote de acordo com sua marcação em filas que possuem prioridades distintas, proporcionando assim a classificação do tráfego, pacote a pacote.

Para compreender as necessidades das aplicações que trafegam na rede, pode-se subdividir em dois tipos: elásticas (tempo não real) e não elásticas (tempo real) [PETERSON; DAVIE, 2013]. As aplicações elásticas são as aplicações clássicas da internet, incluindo a maioria das aplicações populares, como correio eletrônico, acesso remoto, transferência de

arquivos, navegação Web, dentre outras. Todas estas aplicações citadas podem trabalhar sem garantias de tempo referentes à entrega de dados, seu recurso principal é a alta confiabilidade e implementam o conceito cliente-servidor, com a transmissão assíncrona.

As aplicações não elásticas surgiram com o aprimoramento das tecnologias de rede, por exemplo: áudio por demanda, vídeos por demanda, telefonia, VoIP e videoconferência. Essas aplicações tem uma alta sensibilidade à flutuação, entretanto, é menos sensível a perda de pacotes. Tem suas informações codificadas como *stream* e tem sua transmissão síncrona, ou seja, exige maior equilíbrio na transmissão dos dados.

A Tabela 3 mostra uma comparação entre os tipos de aplicações de rede e os requisitos de QoS. As quatro primeiras aplicações (Correio Eletrônico, Transferência de Arquivos, Acesso à Web e Acesso Remoto) são clássicas da internet, com requisitos estritos de confiabilidade, onde nenhum bit pode ser entregue de forma incorreta. Essas aplicações não são sensíveis ao retardo, ou seja, se todos os pacotes estiverem uniformemente atrasados, em alguns segundos, não haverá nenhum problema. As quatro últimas aplicações podem tolerar erros, mas são sensíveis ao retardo. Por exemplo, se houver retardo em uma ligação telefônica, a comunicação se torna inviável.

Tabela 2 - Rigidez dos requisitos de qualidade de serviço

Aplicação	**Confiabilidade**	**Retardo**	**Flutuação**	**Largura de banda**
Correio Eletrônico	Alta	Baixa	Baixa	Baixa
Transferência de Arquivos	Alta	Baixa	Baixa	Média
Acesso à Web	Alta	Média	Baixa	Média
Acesso Remoto	Alta	Média	Média	Baixa
Áudio por demanda	Baixa	Baixa	Alta	Média
Vídeo por dmanda	Baixa	Baixa	Alta	Alta
Telefonia	Baixa	Alta	Alta	Baixa
Videoconferência	Baixa	Alta	Alta	Alta

Fonte: Tanenbaum, 2011

Na Figura 2, segue a taxonomia das aplicações, mostrando uma visão geral de como se podem examinar algumas classes de aplicações diferentes. A representação a seguir deve muito aos estudos de Clark, Brade, Shenker e Zhang [CLARK; SHENKER; ZHANG, 1992].

Segundo Peterson e Bruce (2013) podem-se categorizar as aplicações em uma rede de acordo com suas necessidades e exigências. Primeiramente, observa-se quanto à tolerância, dependendo se a aplicação pode ou não tolerar a perda de dados ocasional, por exemplo, uma amostra de áudio perdida pode ser intercalada a partir das amostras adjacentes, podendo o usuário sentir pouca interferência, somente quando uma quantidade de amostras maior é perdida é possível perceber queda na qualidade, tornando a comunicação inviável. Já um

programa de controle de robô provavelmente é um exemplo de uma aplicação de tempo real que não tolera perdas – perder um pacote que contém o comando instruindo o braço do robô a parar é inaceitável.

Um segundo fator a ser analisado é a adaptabilidade. As aplicações que podem ajustar seu ponto de reprodução são chamadas de aplicações adaptáveis ao atraso. Outra classe de aplicações adaptáveis são as aplicações adaptáveis à taxa. Por exemplo, muitos algoritmos de codificação de vídeo podem escolher entre seus requisitos de taxa de bits e qualidade. Assim, se a rede tiver que ser capaz de suportar certa largura de banda, serão definidos novos parâmetros de codificação de acordo com a necessidade. Se mais largura de banda se tornar disponível mais tarde, os parâmetros podem ser alterados para aumentar a qualidade.

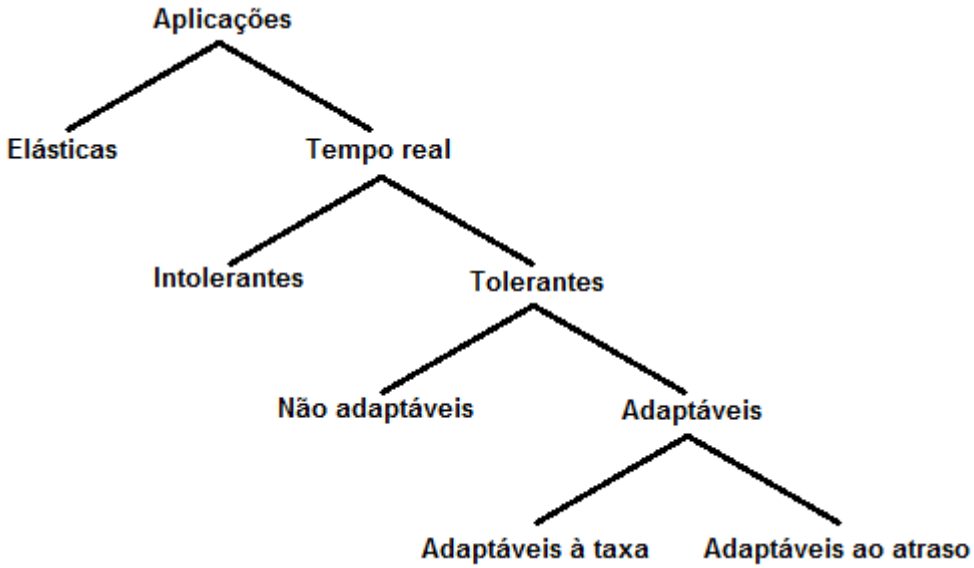

Figura 3 - Taxonomia das aplicações

Fonte: Peterson 2013

2.3.2 Características de Fluxo

Dentro de uma rede comutada por pacotes, existe uma série de recomendações de parâmetros mínimos que devem ser garantidos para que se tenha uma boa qualidade nos serviços oferecidos, ou seja, um conjunto de recomendações desenvolvidos e disponibilizados pela ITU-T (*International Telecommunications Union - Telecommunication Standardization Sector*).

Como mostrado anteriormente, a QoS deve ser determinada por um conjunto de parâmetros de Qualidade de Serviço bem definidos em termos por meio de um SLA. Segundo Forouzan (2006), existem tradicionalmente quatro tipos de características que podem ser

atribuídas aos fluxos de dados: largura de banda, perda de pacotes (confiabilidade), atraso e variação do atraso (*jitter*).

A Figura 3 mostra as principais características de um fluxo.

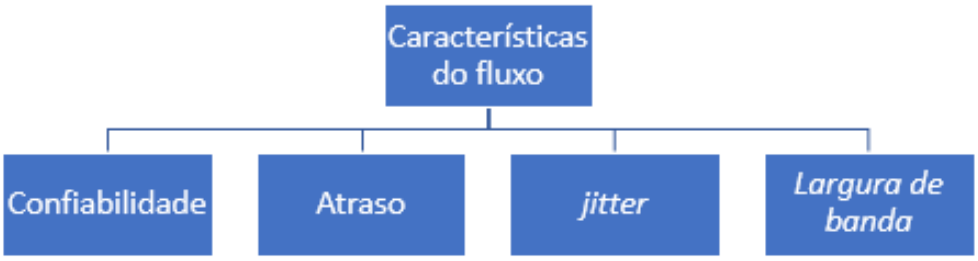

Figura 4 - Características do fluxo

Fonte: Forouzan (2006)

2.3.2.1 Largura de Banda

Segundo Silva (2004), a largura de banda é um termo utilizado para descrever a capacidade de transferência de dados de uma determinada aplicação em uma unidade de tempo. Consumo de banda da aplicação (médio e valor de pico). Pode variar em função dos codificadores de camadas superiores utilizados ou em função da ação do usuário.

Para fins didáticos, pode-se fazer uma analogia ao cano para passagem de água, também pode ser inferido que a largura de banda corresponde à largura do cano para a passagem de uma quantidade pré-determinada de água. Nesse contexto, o bit é a menor unidade de informação transmitida, logo, o cálculo da largura de banda é feito em bits por segundo.

Nem todas as aplicações que funcionam em uma rede de dados necessitam de qualidade de serviços, todavia, todas necessitam de uma largura de banda específica, por isso, este é o parâmetro mais básico na especificação de QoS [MARTINS, 1999]. Partindo desse ponto, pode-se concluir que se deve garantir uma largura de banda mínima para o fluxo de dados da aplicação pré-configurada na rede.

Os diferentes tipos de aplicação precisam de larguras de banda diferentes. Numa conferência de vídeo, por exemplo, é necessário enviar milhões de bits por segundo para restaurar as cores do monitor, enquanto que em um envio de e-mail, a quantidade de bits pode nem mesmo chegar à casa de um milhão.

2.3.2.2 Taxa de perda de pacotes

Esta é uma característica necessária ao fluxo. Segundo Forouzan (2006), A falta de confiabilidade significa que pacotes de dados serão perdidos, ou, que as respostas de

confirmação estão sendo perdidas, o que motiva a retransmissão. Um pacote pode ser enviado e não ser recebido no seu destino, quando isto ocorre diz-se que houve a perda do pacote. A quantidade de pacotes perdidos tende a aumentar conforme o aumento do tráfego na rede, sendo que a maioria das perdas ocorre em nós (*switches* ou roteadores) congestionados. A perda de pacotes pode ser considerada tolerável até o momento em que não comprometa a aplicação. Sua medida pode ser obtida com o percentual de pacotes perdidos em função de pacotes transmitidos. Enquanto a perda de pacotes em uma aplicação de voz deve ficar em torno de 1%, no caso de perda de pacotes acima de 5% do total, uma conversa telefônica utilizando VoIP já fica comprometida [KUROSE; ROSS, 2010].

A perda de pacotes interfere diretamente nas aplicações da rede e deve ser considerada quando se pensa em implantar QoS, pois tais perdas podem inviabilizar a utilização de algumas aplicações sensíveis a perda de pacotes. Os pacotes podem ser perdidos em várias situações, as principais são [FILHO, 2006]:

- perda devido às filas congestionadas nos roteadores: quando a fila do roteador está saturada, os pacotes direcionados a este roteador são descartados.
- falha em algum enlace no caminho do pacote: a queda de um enlace pode interferir na transmissão de um pacote.
- encaminhamento errado por um nó da rede.
- erro na transmissão.

As perdas de pacotes são um grave problema principalmente para aplicações em tempo real, como é o caso da voz sobre IP e da videoconferência, por exemplo, perdas de pacotes de voz digitalizada podem tornar a comunicação inviável. Por isso, existe a necessidade de definição e garantia de perdas mínimas para cada tipo de aplicação.

Em relação à qualidade de serviço da rede (QoS) a preocupação é normalmente no sentido de especificar e garantir limites razoáveis (Taxas de Perdas) que permitam uma operação adequada da aplicação.

2.3.2.3 Atraso (Retardo/*Delay*)

É o acúmulo de atrasos de processamento, de transmissão e de formação de filas nos roteadores; atrasos de propagação nos enlaces e atrasos de processamento em sistemas finais. O atraso nas filas, que é o tempo que os pacotes esperam nas filas de um dado equipamento para serem transmitidos ao enlace; e o atraso de processamento, que corresponde ao tempo consumido pelos equipamentos para examinar e encaminhar um pacote [FILHO, 2006].

A latência e o atraso são parâmetros importantes para a qualidade de serviço das aplicações. De maneira geral, a latência da rede pode ser entendida como o somatório dos atrasos impostos pela rede e equipamentos utilizados na comunicação. Aplicações de tempo real como telefonia e videoconferência, são mais sensíveis ao atraso. Por exemplo, se alguns pacotes forem retardados em uma chamada telefônica, a ligação não terá a qualidade esperada.

Em uma aplicação de telefonia o atraso de até 150 milissegundos não é percebido pelo ouvido humano, atraso entre 150 e 400 milissegundos podem ser aceitáveis, já atraso acima de 400 milissegundos pode tornar a comunicação inviável, fazendo com que o lado receptor de uma aplicação de telefone, por exemplo, desconsidere quaisquer pacotes cujos atrasos ultrapasse determinado limite [KUROSE; ROSS, 2010], em contrapartida, o mesmo atraso em uma aplicação como correio eletrônico não diminui a qualidade do serviço.

Os principais responsáveis pela latência são o atraso de transmissão, de codificação e de empacotamento, que podem ser definidos da seguinte forma [ROESLER,2001]:

- **Atraso de transmissão**: tempo que leva para o pacote sair da placa de rede do computador origem e chegar na placa de rede do computador destino. Este intervalo de tempo engloba diversos fatores, como por exemplo atraso no meio físico, atraso de processamento nos equipamentos intermediários, como roteadores e switches, atraso devido ao tempo de espera nas filas de transmissão dos equipamentos intermediários;
- **Atraso de codificação e decodificação**: o atraso de codificação refere-se ao tempo de processamento na máquina origem e atraso de decodificação na máquina destino para processamento de sinais. Voz e vídeo normalmente são codificados em um padrão, tal como PCM (*Pulse-code modulation*) (G.711 a 64 Kbps) para voz, ou H.261 para vídeo. O atraso varia com o padrão adotado; por exemplo, o G.711 ocupa menos de 1ms de codificação, porém requer 64 Kbps de banda. Um protocolo de voz como o G.729 requer 25 ms de codificação, mas ocupa apenas 8 Kbps de banda;
- **Atraso de empacotamento e desempacotamento**: depois de codificado, o dado deve ser empacotado através dos níveis na pilha de protocolos a fim de ser transmitido na rede. Por exemplo, numa transmissão de voz a 64 Kbps, ou 8000 bytes por segundo, o preenchimento de um pacote de dados com apenas 100 bytes demora cerca de 12,5 ms. Mais 12,5 ms ainda serão necessários no destino a fim de desempacotar os dados.

2.3.2.4 *Jitter* (Variação do atraso)

Considerando uma rede de computadores, *jitter* é a variação no atraso dos pacotes pertencentes a um mesmo fluxo [FOROUZAN, 2006], ou seja, variação na latência da rede. As aplicações de áudio e vídeo em tempo real não toleram atrasos em diferentes graus, causando um impacto negativo sobre a qualidade dessas aplicações. O *jitter* é caracterizado pela quebra da sequência no tráfego dos pacotes, pois os mesmos podem seguir caminhos distintos na rede e, em cada roteador, o pacote deve esperar sua vez para ser processado, fazendo com que alguns pacotes não chegam ao destino na ordem de envio. O intervalo de tempo entre o instante em que um pacote é gerado no transmissor e o momento em que é recebido no receptor pode variar de pacote para pacote.

Como pode ser observado na Figura 4, no primeiro fluxo os pacotes chegam com um intervalo igual de 20 ms, já no segundo tráfego, acontece a variação na latência, pois o intervalo de tempo de chegada entre um pacote e outro é diferente.

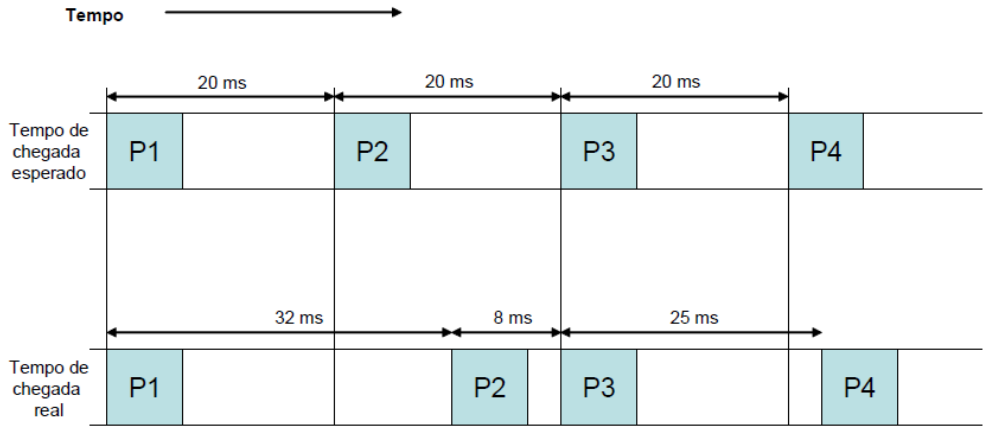

Figura 5 - Exemplo de *jitter*

Ruela, 2006

Aplicações como vídeo e voz, precisam que os pacotes cheguem em ordem e em tempos definidos, exigindo um *jitter* mínimo para não sofrerem a degradação da comunicação. Serviços de voz, vídeo e transações críticas, para serem oferecidos com QoS, precisam que os problemas com a variação no atraso sejam resolvidos ou minimizados. O problema do *jitter* pode ser amenizado com a utilização de um atraso de reprodução [KUROSE; ROSS, 2010].

O *jitter* introduz distorção no processamento da informação na recepção e deve ter mecanismos específicos de compensação e controle que dependem da aplicação em questão. Genericamente, uma das soluções mais comuns para o problema consiste na utilização de

buffers (Técnica de "*buffering*"). Esta técnica consiste em retardar a reprodução das porções de áudio/vídeo, agrupando os pacotes em um buffer antes da reprodução. O atraso deve ser suficientemente longo para que grande parte dos pacotes seja recebida antes do tempo de programado para iniciar reprodução pelo receptor. Assim, os pacotes são reproduzidos sem que a latência degrade a comunicação.

2.3.3 Alternativas técnicas para implantação de QoS em rede

Existem várias possibilidades para implantação de qualidade de serviços em redes IP. Deve-se considerar cada situação em que a QoS é requerida para determinar que mecanismo será implantado, como, por exemplo, nos períodos de pico de tráfego, quando a rede enfrenta situações de congestionamento e de carga muito elevada [MARTINS, 1999]. Esta sessão abordará dois dos principais mecanismos para o provimento de QoS em uma rede de comunicação baseadas em IP: *IntServ* e *DiffServ*.

A utilização dos modelos é enfatizada na camada de rede (IP), embora estes mesmos modelos também possam ser aplicados à camada de enlace de dados. O protocolo de camada de rede, IP, foi originalmente desenvolvido com uma entrega de melhor esforço (*best effort*). Isso significa que cada usuário recebe o mesmo tipo de serviço [FOROUZAN, 2006]. Esse tipo de entrega não garante o mínimo de serviço, como por exemplo largura de banda para aplicações multimídia.

Das duas alternativas apresentadas, dar-se-á uma ênfase para a arquitetura *DiffServ*, visto que esta é o ambiente alvo deste trabalho, mostrando uma comparação entre as duas e justificando a escolha do mecanismo a ser implantado na rede.

2.3.3.1 Arquitetura IntServ

O termo Serviços Integrados (*Integrated Services*, sendo abreviado como IntServ) refere-se a um conjunto de trabalhos que foi produzido pela IETF por volta de 1995 a 1997 [PETERSON; DAVIE, 2013].

Os Serviços Integrados constituem um modelo de QoS baseado em fluxo [FOROUZAN, 2006], ou seja, o usuário necessita criar um fluxo individual, um tipo de circuito virtual da origem ao destino e informar a todos os dispositivos no caminho seus requisitos de recursos, onde cada fluxo possui a sua própria reserva. O IntServ é responsável pela implementação da reserva de recursos em redes IP, primeiramente ele reserva o recurso para depois estabelecer a sessão provendo o fluxo.

Para que essa proposta do IntServ seja empregada em redes IP, que são comutadas por pacotes e consequentemente *best effort*, surge a necessidade de uma sinalização que age como controle responsável pela reserva, quem assume a responsabilidade da sinalização é o protocolo RSVP.

O RSVP (*Resource Reservation Protocol*) é um protocolo de sinalização que tem suas funcionalidades sobre o tráfego de pacotes numa rede. Além da definição de como as aplicações solicitam sua necessidade de QoS à rede, outro aspecto operacional da arquitetura *IntServ* é como os elementos da rede (roteadores, switch, etc) procederão para que seja garantida a qualidade de serviço solicitada, que são detalhadas em várias recomendações RFCs (*Request for Comment* - Requisições de Comentários) produzidas pelo IETF.

O RSVP permite que vários transmissores enviem os dados para vários grupos de receptores, torna possível receptores individuais mudarem livremente de canais e otimiza o uso da largura de banda ao mesmo tempo que elimina o congestionamento [TANENBAUM; WETHERALL, 2011].

Com relação a Classes de Serviços no modelo IntServ ele possui dois níveis de prioridade: Serviços Garantidos e Serviços Controlados. Os Serviços Garantidos que é a qualidade máxima, envolvendo o tráfego em tempo real, como por exemplo as aplicações multimídia, tempo real, que tem tráfegos que exigem alta largura de banda e velocidade, exigindo limites fixos de atraso.

Os Serviços Controlados são utilizados para aplicações que podem aceitar certos atrasos, entretanto com alta sensibilidade a uma rede muito congestionada provocando o descarte de muitos pacotes, como por exemplo as aplicações multimídia sob demanda, transferência de arquivos.

A arquitetura IntServ garante qualidade de serviços a fluxos individuas devido a sua capacidade de requisitar reserva de recursos por fluxo. Entretanto, a reserva de recurso por fluxo induz a algumas dificuldades na implementação desta arquitetura. Segundo Kurose e Ross (2010) existem duas dificuldades inerentes à arquitetura IntServ associadas ao modelo de reserva de recursos por fluxo e a escalabilidade.

A reserva de recursos por fluxo exige que um roteador seja utilizado para processar tal reserva requisitada, mantendo o estado de cada fluxo que passa roteador. Em uma rede de grande porte isso pode ocasionar uma sobrecarga no roteador que executa, influenciando também na escalabilidade da rede. A quantidade de informação de estado aumenta com o número de fluxos, exigindo enorme espaço de armazenamento e gerando sobrecarga de processamento nos roteadores.

As exigências para os roteadores vão se tornando cada vez mais altas. Além disso, todos os roteadores têm que implementar o protocolo RSVP, classificação, controle de admissão e escalonamento de pacotes. Essas considerações levaram a criação da arquitetura Diffserv, que tem como objetivo prove diferenciação de serviços escalável e flexível – isto é, a manipulação de diferentes "classes" de tráfego de diversas formas na internet [KUROSE; ROSS, 2010].

2.3.3.2 Arquitetura DiffServ

Segundo Stallins (2008), à medida em que a demanda de serviços na Internet aumenta, assim como a diversidade nas aplicações, existe uma necessidade imediata de proporcionar níveis diferenciados de QoS a diferentes fluxos de tráfegos.

O mecanismo DiffServ é outra iniciativa do IETF [RFC 2475]. Com base nas limitações encontradas no IntServ, foi proposta a Arquitetura de Serviços Diferenciados (DiffServ) que oferece QoS na Internet com escalabilidade: sem estado para cada fluxo e sinalização a cada nó. Esse mecanismo utiliza a estratégia que pode ser implementada em grande parte no local em cada roteador, sem configuração antecipada e sem ter de envolver todo o caminho. Essa abordagem é conhecida como qualidade de serviço baseada na classe (em vez de ser baseada no fluxo).

A IETF padronizou uma arquitetura para ela, chamada arquitetura de serviços diferenciados, descrita na RFC 2475 [TANENBAUM; WETHERALL, 2011]. A solução DiffServ não utiliza nenhum tipo de mecanismo de reserva de recurso. Nesta arquitetura, os pacotes são classificados, marcados e processados segundo o seu rótulo DSCP [MARTINS, 1999].

A IETF padronizou um conjunto de comportamentos dos roteadores a serem aplicados aos pacotes marcados. Estes são chamados comportamentos por salto (PHBs – *Per-hop behaviors*), um termo que indica que eles definem o comportamento de roteadores individuais, ao invés de serviços fim a fim [PETERSON; DAVIE, 2013]. Como existe mais de um comportamento novo, surgiu a necessidade de se utilizar mais de 1 bit de sinalização no cabeçalho do pacote, a fim de informar aos roteadores qual ação aplicar.

A IETF decidiu usar o antigo byte TOS (*Type of Service* - Tipo de Serviço) do cabeçalho IPv4 ou o campo CoS (*Class of Service* - Classe de Serviço), no caso do IPv6 e redefini-lo. Seis bits desse byte foram alocados para os pontos de código *DiffServ* (DSCP – *DiffServ Code Points*). Como explicado em tópicos anteriores, o DSCP é um valor de 6 bits

que identifica determinado PHB a ser aplicado a um pacote, ou seja, que comportamento eles terão na rede.

O serviço DiffServ é definido através de um Acordo de Nível de Serviço (SLA) entre o cliente e o provedor de serviços. Pode-se destacar também que o SLA contém especificação de condicionamento de tráfego, que determina entre outras coisas alguns parâmetros detalhados do desempenho do serviço, como níveis esperados de vazão, atraso e perda de pacotes. Além disso, são definidos perfis de tráfego, através de parâmetros de balde de fichas, que definem as características do tráfego como também as ações que podem ser aplicadas caso o usuário não cumpra as especificações.

Um Domínio DiffServ (DS) é composto por um conjunto de dispositivos de encaminhamento contíguos [Stallins, 2008]. Isso significa que operam um conjunto comum de ações. É possível chegar a partir que qualquer dispositivo de um domínio a qualquer outro deste mesmo domínio. Dentro de um domínio, a interpretação dos códigos de DS é uniforme, proporcionando um serviço uniforme e consistente.

Domínios DS negociam entre si esses contratos de SLA. Nele são especificados o serviço de encaminhamento e a classificação de tráfego que o cliente deve receber, além de determinar as garantias mínimas de QoS à aplicação cliente. Dessa forma, quando os pacotes fluem de um domínio DiffServ para o outro eles são fiscalizados (policiados) nos roteadores de fronteira a fim de verificar a conformidade nos acordos de SLA.

A Figura 5 mostra as fronteiras onde são observados os contratos de SLA.

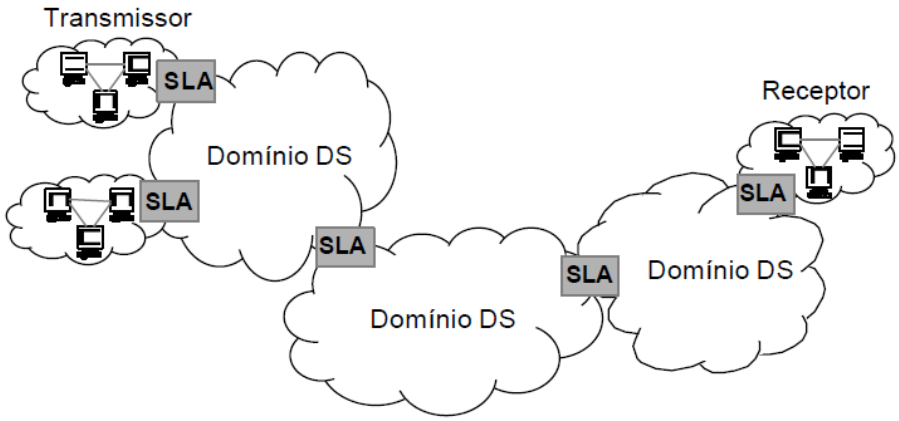

Figura 6 - Arquitetura lógica DiffServ

Fonte: PETERSON; DAVIE, 2013

Em um DS existem roteadores de borda e de núcleo. Segundo Stallins (2008), normalmente os nós de núcleo implementam mecanismos mais simples para processamento

dos pacotes baseando-se nos valores do código DS. Isto inclui o controle de filas para dar um tratamento preferencial baseado no valor do código e nas regras de descarte de pacotes.

Os nós de fronteira incluem os mecanismos PHB, porém necessitam dos demais mecanismos de condicionamento de tráfego mais sofisticado para proporcionar o serviço desejado. Desta forma, os dispositivos de encaminhamento de núcleo têm uma funcionalidade e sobrecarga mínima ao proporcionar o serviço de DS, enquanto a maior complexidade se situa nos nós de fronteira.

As atividades referentes ao policiamento dos pacotes nos roteadores de borda para averiguar sua adequação ao perfil de tráfego contratado são coletivamente chamadas de condicionamento de tráfego.

A Figura 6 mostra a sequência de interação entre os cinco elementos que compõe o condicionamento de tráfego.

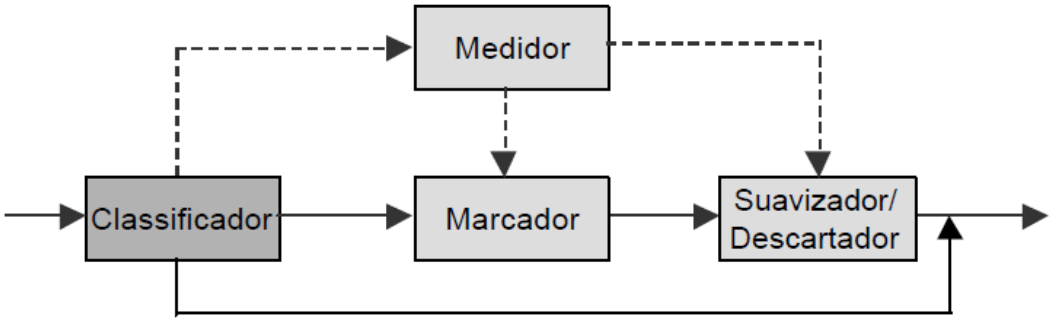

Figura 7 - Condicionamento de tráfego

Fonte: Fonte: PETERSON; DAVIE, 2013

Cada elemento mostrado na Figura 6 tem as seguintes funcionalidades [STALLINS, 2008]:

- **Classificador:** Separa os pacotes enviados em diferentes classes. Esta é a base do provisionamento dos Serviços Diferenciados. Ele seleciona os pacotes que chegam nas interfaces de entrada baseado no conteúdo de algum campo de seu cabeçalho. Dois dos principais marcadores são: Classificador BA (*Behavior Aggregate*) que classifica o pacote baseando-se somente no campo DSCP. Isso ocorre quando o pacote já vem de um DS e os pacotes já chegam em sua interface marcados; Classificador MF (*Multi-Field*), quando o pacote não chega com seu campo DSCP marcado pois em seu domínio proveniente não é habilitado para enviar pacotes com o campo DSCP.

- **Medidor:** Mede o fluxo de pacotes selecionados pelo classificador para comprovar que está conforme o perfil de tráfego especificado. O medidor determina se uma classe

de fluxo de pacotes cumpre ou excede o nível de serviço garantido para esta classe. O mecanismo mais apropriado para a medição é o balde de símbolos. Pacotes fora do perfil são os que chegam quando dentro do balde não existem fichas suficientes. Baseando-se no resultado da medição, os pacotes são encaminhados para um estágio onde uma ação de condicionamento é realizada. Esta ação depende do serviço oferecido, mas geralmente são suavização, descarte, marcação ou contabilização para posterior cobrança. Geralmente os pacotes dentro do perfil não sofrem nenhuma ação, a não ser no caso do Classificador MF, após o qual os pacotes devem ser marcados para um DSCP específico.

- **Marcador:** Remarca os pacotes com um código diferente de acordo com a necessidade. Isso pode ser feito com os pacotes que foram classificados por um Classificador MF, para que os roteadores de núcleo posteriores possam classificar os pacotes com um Classificador BA. Ele é utilizado também quando os dois domínios utilizam valores do DSCP diferentes para o mesmo BA.
- **Suavizador:** Retarda os pacotes para que um fluxo de pacotes de uma classe para que não exceda a taxa de tráfego especificada no perfil, ou seja, molda o tráfego para que fique dentro do perfil contratado.
- **Descartador:** Descarta pacotes quando a taxa de pacotes de uma dada classe excede a taxa especificada no perfil desta classe, ou seja, pacotes fora do perfil. Também chamado de policiamento de tráfego.

Várias alternativas podem ser seguidas, caso um fluxo de dados exceda algum perfil. Os pacotes individuais que excedem o perfil podem ser remarcados para um tratamento com qualidade inferior e os permitir passar para o domínio DS.

O condicionamento de tráfego, em geral, é necessário somente nos roteadores de borda. Isto não influencia no desempenho pois os roteadores de borda geralmente têm um volume de tráfego menos que os roteadores de núcleo. Já os roteadores de núcleo destinam a maior parte dos seus recursos na execução de encaminhamento de pacotes, de acordo com seu PHB.

2.3.3.3 Comparação entre Serviços Integrados e Serviços Diferenciados

A Tabela 4 apresenta uma sucinta comparação entre as duas arquiteturas, abordando suas principais características.

A escolha pela utilização do modelo DiffServ ocorre naturalmente, visto que este mecanismo tem como objetivo prover maior escalabilidade devido a menor granularidade,

pois, não se faz necessário solicitar uma reserva de recursos específica para cada fluxo. Não existe a necessidade de sinalização em cada roteador da rede.

Tabela 3 - Comparação entre IntServ e DiffServ

Serviço	**IntServ**	**DiffServ**
Funcionamento	Reserva de recursos	Classificação de pacotes
Base do Serviço	Fluxo	Classes
Tratamento de Fluxos	Fluxos individuais	Agregação de fluxos
Escalabilidade	Redes pequenas	Grandes redes
Protocolo de sinalização	RSVP	Não requer para esquemas relativos
Gerenciamento da rede	Similar a rede de comutação por circuito	Similar a rede IP
Coordenação para diferenciação do serviço	Fim a Fim	PHB

Fonte: NETO, 2014

2.3.4 Mecanismos de controle de QOS

As redes de computadores evoluíram no sentido de integrar diferentes tipos de tráfegos em uma única infraestrutura, porém o que dificulta essa integração é que aplicações de tempo real possuem necessidades diferentes das aplicações convencionais.

Antes de abordarmos isoladamente sobre os mecanismos de QoS, é importante frisar que as funções de QoS não funcionam de forma isolada, mas conjuntamente, a fim de prover uma rede de tráfego com qualidade e cada função pode consistir em vários recursos diferentes.

Os mecanismos de QoS são a base para as alternativas técnicas que proveem a Qualidade de Serviço. Esses mecanismos compreendem Classificação e Marcação, Precedência IP e DSCP, Gerenciamento de Congestionamento, Moldagem e Policiamento de tráfego e Controle de Admissão.

É necessário que os algoritmos de prioridade, controle de filas, escalonamento e congestionamento sejam implementados de acordo com a política adotada para a solução de algum problema específico, afim de que os mecanismos de controle funcionem corretamente. A seguir será descrito cada mecanismo juntamente com suas características e funções.

2.3.4.1 Classificação e Marcação

A classificação e marcação são termos de grande importância para a QoS. Na classificação o fluxo de pacotes é identificado, podendo ou não ser marcado em seguida. A classificação de pacotes envolve o uso de um descritor de tráfego para categorizar um pacote dentro de um grupo específico e torna-lo acessível para tratamento de QoS na rede. Usando a classificação do pacote, você pode particionar o tráfego de rede em diversos níveis de prioridade ou em uma classe de serviço (CoS) [CISCO SYSTEMS, 2009]. A classificação dos pacotes pode acontecer sem marcação [SZIGETI *et al.,* 2014].

A classificação caracteriza os grupos de pacotes que receberão um determinado tratamento. Esses pacotes podem ser classificados de acordo com diversos parâmetros, onde os roteadores realizam a marcação dos pacotes, e separam os pacotes que entram na rede através de diversas classes de serviço. A Figura 7 mostra alguns pacotes sendo separados e marcado de acordo com deu protocolo ou aplicação, como por exemplo o tráfego de voz, que recebe a marcação com valor DSCP de 46.

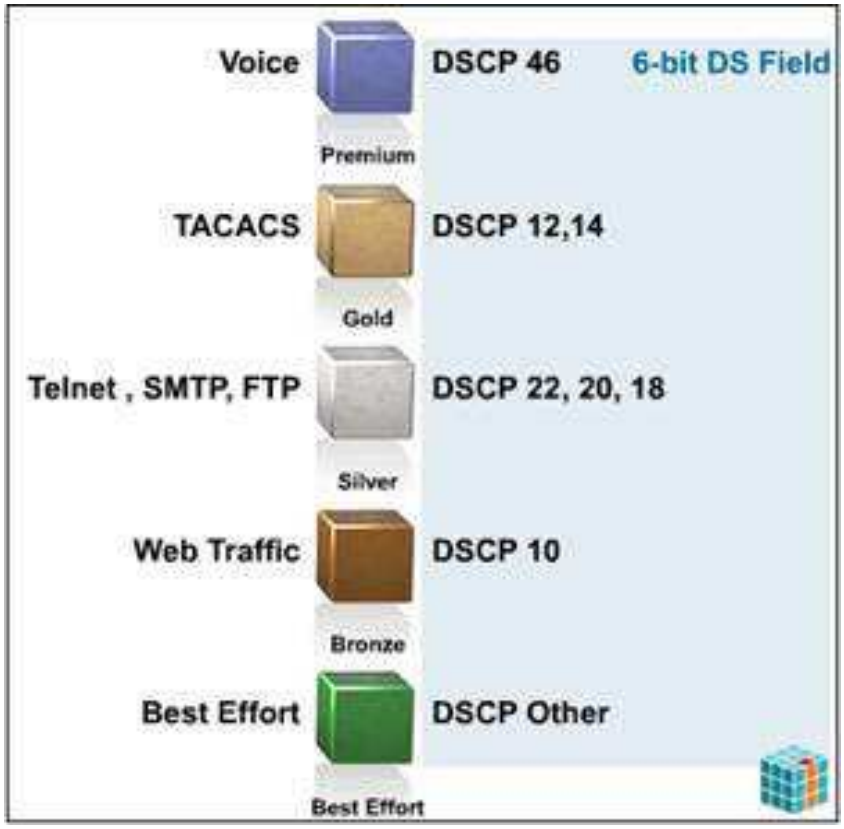

Figura 8 - Exemplos de classes de Serviços

Fonte: FALSARELLA, 2009

Após a classificação, os pacotes que foram marcados podem ser manipulados por outros mecanismos de QoS como alocação de banda, controle de congestionamento, dentre outros. A classificação pode ser realizada baseando-se em portas físicas, IP origem e destino, porta de aplicação, tipo de protocolo e outros critérios que utilizam lista de acessos. A marcação nem sempre é usada exclusivamente para fins de classificação.

A ferramenta de classificação inspeciona um ou mais campos em um pacote para identificar o tipo de tráfego que o pacote está transportando. Uma vez identificado, o tráfego é direcionado para um mecanismo de execução de políticas para este tráfego marcado, recebendo tratamento pré-definido, isto inclui marcação, enfileiramento, moldagem ou qualquer outra combinação de ações de QoS.

A ferramenta de marcação grava um valor no cabeçalho do pacote, quadro, etiqueta ou rótulo para fixar a decisão de classificação alcançada no limite de confiança da rede. A marcação de pacotes de camada 3 com precedência de IP e DSCPs é a marcação mais amplamente utilizada pois, as marcações de pacotes de camada 3 têm significado de rede de ponta a ponta e podem facilmente ser traduzidos para e para as marcações de quadros de camada 2.

2.3.4.2 Precedência IP e DSCP

A classificação dos pacotes se baseia em um campo do cabeçalho IP, denominado DS-*field*. Este pode ser considerado como uma redefinição do campo *Type of Service* (ToS) presente no cabeçalho IPv4, ou *Class of Traffic* do IPv6. Seis bits do DS-*field* são usados para identificar o valor de DSCP (*Differentiated Services Code Point*), usado na seleção do PHB (*Per-Hop-Behavior*). Os dois últimos bits (CU - C*urrently Unused*), atualmente estão sendo utilizados como bits de notificação de congestionamento explícito (ECN - *Explicit Congestion Notification*) [CISCO SYSTENS, 2009].

A Figura 8, a seguir, ilustra os cabeçalhos de pacotes IPv4 e IPv6. Como pode ser observado, os primeiros três bits desses dois campos são os bits IPP (IP Precedência), dando um total de oito possíveis classes de serviço. Estes primeiros 3 bits combinados com os próximos 3 bits são conhecidos coletivamente como os bits DHCP. Estes 6 bits oferecem um máximo de 64 possíveis classes de serviço. Na Figura 8 observam-se os protocolos IPv4 e IPv6, ambos contêm um campo reservado para atribuição de valores DSCP de acordo com sua prioridade.

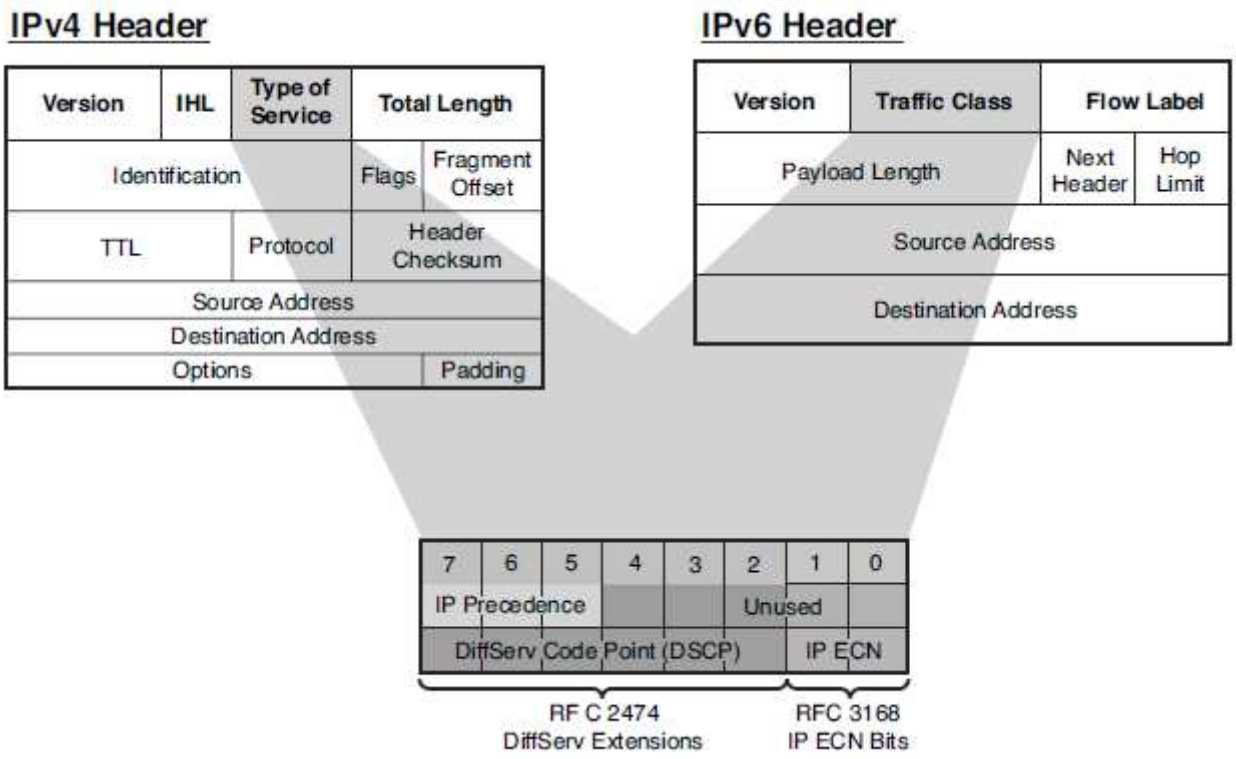

Figura 9 - Campo DSCP nos pacotes IPv4 e IPv6

Fonte: SZIGETI et al., 2014

Ao longo do tempo, o conjunto de RFCs da IETF evolui, se tornando abrangente suficiente para, em sua maior parte, orientar a operação dos recursos em equipamentos de rede. Pode-se considerar com uma contribuição fundamental das RFCs, a caracterização das classes de tráfego, definindo sua identificação e destinando ao devido tratamento [SZIGETI et al., 2014]. A Tabela 5 mostra as principais classes de tráfego consideradas em um fluxo de dados relacionando com seus respectivos valores DSCP.

Tabela 4 - Diretrizes RFC para classes de tráfego

Aplicação	Classificação		IETF
	PHB	DSCP	RFC
Controle de rede	CS6	48	RFC 2474
Telefone VoIP	EF	46	RFC 3246
Sinalização de chamada	CS5	40	RFC 2474
Conferência multimídia	AF41	34	RFC 2597
Interação em tempo real	CS4	32	RFC 2474
Multimídia *Streaming*	AF31	26	RFC 2597
Vídeo em *Broadcast*	CS3	24	RFC 2474
Dados de baixa latência	AF21	18	RFC 2597
Operação/ Administração/ Gestão	CS2	16	RFC 2474
Alta taxa de transferência/ Dados em massa	AF11	10	RFC 2597
Melhor Esforço	DF	0	RFC 2474
Dados de baixa prioridade/*Scavenger Data*	CS1	8	RFC 3662

Fonte: SZIGETI et al., 2014

Para obtermos as classes de serviços (*Class of Service* – CoS), são utilizados 3 bits do campo ToS (*Type of Service*) do cabeçalho IP. A Tabela 6 apresenta os 3 bits para o campo Precedência IP associando às suas configurações e funções.

Tabela 5 - Valores de precedência IP

Valor	Bits	Descrição
0	000	Precedência Padrão (rotina)
1	001	Precedência prioridade
2	010	Precedência Prioridade Imediata
3	011	Precedência Relâmpago (*Flash*)
4	100	Precedência Super Relâmpago
5	101	Precedência crítica;
6	110	Precedência de controle Inter-Redes
7	111	Precedência de controle de rede

Fonte: SZIGETI et al., 2014

Os valores de DSCP podem ser expressos em forma numérica ou por palavras-chave especiais, denominados comportamentos de per-hop (PHBs). Existem três classes definidas de DSCP PHBs: Melhor esforço (BE ou DSCP 0), Encaminhamento garantido (AFxy) e Encaminhamento expedido (EF). Além desses três PHBs definidos, os pontos de código do Seletor de Classe (CSx) foram definidos como compatíveis com a precedência de IP descritos na Tabela 6. (Em outras palavras, CS1 à CS7 são idênticos aos valores de precedência de IP 1 a 7.) Os RFCs que descrevem estes PHBs são 2597 e 3246.

Serão apresentados dois PHBs padronizados, que são:

- Encaminhamento Expedido (EF – *Expedited Forwarding*) (RFC 3246)

  Os pacotes marcados para tratamento EF devem ser encaminhados pelo roteador com o mínimo de atraso ou perda. A única maneira pela qual um roteador pode garantir isso a todos os pacotes EF é se a taxa de chegada dos pacotes EF no roteador for estritamente limitada para ser menor que a taxa em que o roteador pode encaminhar pacotes EF [PETERSON; DAVIE, 2013].

  Este PHB pode ser usado para construir serviço fim-a-fim com as seguintes características: baixa latência, baixa perda, baixo *jitter* e banda

assegurada. Configurando os roteadores de borda de um domínio administrativo, pode-se limitar a taxa dos pacotes EF que chegam na rede.

Uma abordagem simples, embora conservadora, seria garantir que a soma das taxas de todos os pacotes EF que entram no domínio seja menor que a largura de banda do enlace mais lento no domínio. Isso garantiria que, até mesmo no pior caso, onde todos os pacotes EF convirjam para o enlace mais lento, ele não seja sobrecarregado e possa oferecer o comportamento correto. Os pacotes expedidos devem ser capazes de transitar pela sub-rede como se nenhum outro pacote estivesse presente [TANEMBAUM, 2003]. Recomenda-se o ponto de código 101110 (46) para o PHB de EF [CISCO, 2001].

- Encaminhamento Garantido (AF – *Assured Forwarding*) (RFC 2597)

  O grupo AF PHB especifica quatro classes de prioridade, onde, para cada classe é alocada uma determinada quantidade de recursos para o encaminhamento do tráfego (buffer e banda). Dentro de cada classe, um pacote IP pode ser associado a um dentre três níveis de precedência de descarte. O grupo AF pode ser usado para implementar serviços que garantam um valor de banda, mesmo em tempos de congestionamento.

A prioridade no tratamento e alocação de recursos da rede é diretamente proporcional ao nível de classificação do pacote, ou seja, quanto maior o nível de classificação do pacote, maior será a prioridade no tratamento e alocação de recursos da rede. Todos os pacotes são normalmente marcados com nível zero (melhor esforço). Em roteadores Cisco, o campo precedência IP pode ser modificado através de listas de acesso (*access lists*) ou mapas de rotas (*route maps*).

Associando um pacote com um *Ip Precedence* ou IP DSCP, é concedido aos usuários classificar o tráfego baseados nos valores de *Ip Precedence* e IP DSCP. Esta ação serve para a decidir como os pacotes serão tratados por outros mecanismos de QoS, a fim de controlar tráfego na rede. Os valores de 3 bits do *IP Precedence* e os 6 bits do DSCP podem coexistir na mesma rede. Lembrando que ambos os valores residem no octeto TOS, o valor do *IP Precedence* representa o seletor de classes dentro do DSCP [HERSENT, 2002].

O RFC 2597 define quatro classes de encaminhamento garantido, denotadas pelas letras AF seguidas por dois dígitos. O primeiro dígito denota a classe AF e pode variar de 1 a 4. O segundo dígito refere-se ao nível de preferência de descarte dentro de cada classe AF e pode variar de 1 (menor preferência de descarte) para 3 (preferência de descarte mais alta).

Por exemplo, durante períodos de congestionamento, AF33 estatisticamente descartaria com mais frequência do que AF32, que, por sua vez, seria descartado mais frequentemente do que a AF31.

2.3.4.3 Gerenciamento de Congestionamento / Política de Filas

Congestionamento numa rede pode ocorrer se a carga na rede, a quantia de pacotes enviados para a rede, supera a capacidade da rede, isto é, a quantidade de pacotes que uma rede consegue controlar. O controle de congestionamento trata as técnicas e dos mecanismos de controle para manter a carga abaixo da capacidade da rede [FOROUZAN, 2006]. Este controle é realizado à medida que o congestionamento se estabelece em uma comunicação de rede.

Conceitualmente, congestionamento é definido pelo guia de configuração do Cisco IOS Software como: "Durante períodos de congestionamento da transmissão na interface de saída, os pacotes chegam mais rápido do que a interface é capaz de enviá-los" [CISCO, 2008], ou seja, o congestionamento geralmente acontece quando uma interface de saída lenta é alimentada por uma interface de entrada mais rápida. Funcionalmente, o congestionamento é definido como o preenchimento do anel de transmissão na interface. Um anel é uma estrutura de controle de buffer especial. Toda interface tem um par de anéis: um de recepção para receber pacotes e outro de transmissão para transmitir pacotes. O tamanho dos anéis varia de acordo com a controladora da interface e com a largura de banda da interface ou do circuito virtual (VC) [CISCO, 2008].

O Cisco IOS, também denominado processador da Camada 3 (L3), e o driver de interface utilizam o anel de transmissão ao mover pacotes para a mídia física. Os dois processadores interagem da seguinte maneira:

• A interface transmite os pacotes de acordo com a taxa de interface ou com uma taxa de modelagem;

• A interface mantém uma fila de hardware ou um anel de transmissão, onde armazena os pacotes que aguardam transmissão no cabo físico;

• Quando a fila do hardware ou o anel de transmissão é preenchido, a interface faz uma pressão contrária explícita no sistema do processador L3. A interface notifica o processador L3 para interromper o “desenfileiramento” de pacotes para o anel de transmissão da interface porque esse anel está cheio. O processador L3 agora armazena os pacotes excedentes nas filas L3;

• Quando a interface envia os pacotes no anel de transmissão e esvazia o anel, ela dispõe novamente de buffers suficientes para armazenar os pacotes. Ela alivia a pressão contrária, e o processador L3 retira da fila os novos pacotes para a interface.

O aspecto mais importante desse sistema de comunicação é que a interface reconhece que o seu anel de transmissão está cheio e acelera o recebimento de novos pacotes vindos do sistema do processador L3. Assim, quando a interface está congestionada, a decisão de descarte muda de uma decisão aleatória do tipo último a entrar/primeiro a ser descartado na fila FIFO (first in, first out) do anel de transmissão para uma decisão diferenciada baseada em políticas de serviço de nível IP implementadas pelo processador L3.

Existem vários mecanismos de enfileiramento para controle e prevenção de congestionamento em interfaces de roteadores (Ethernet, seriais, *Frame Relay*, etc.) e switches camada 3, que podem ser aplicados tanto em redes WAN quanto LAN. Dentre os principais mecanismos, tem-se:

- ***First In First Out* (FIFO):** Na Fila FIFO (*Firt-in-First-out*), os pacotes alocados num buffer (fila) esperam até que o nó (roteador ou switch) possa realmente processá-los. Se a taxa de chegada dos pacotes for maior que a taxa de processamento dos pacotes, a fila será preenchida e os novos pacotes serão descartados [FOROUZAN, 2006]. Geralmente este tipo de enfileiramento é implementado para controlar o tráfego nas conexões seriais e é um mecanismo de armazenamento e repasse (*store and forward*), não implementa nenhum tipo de classificação.

  Como mostra a Figura 9, à medida que os pacotes chegam, é determinada a alocação de banda, e o que chega primeiro é logo atendido. Esse comportamento é encontrado por padrão nos roteadores, já que não tem a necessidade de configuração. Em contrapartida, quando ocorre o tráfego em rajadas pode causar longos atrasos em aplicações sensíveis ao tempo, logo, as filas FIFO não se enquadram em um comportamento esperado para provimento de QoS.

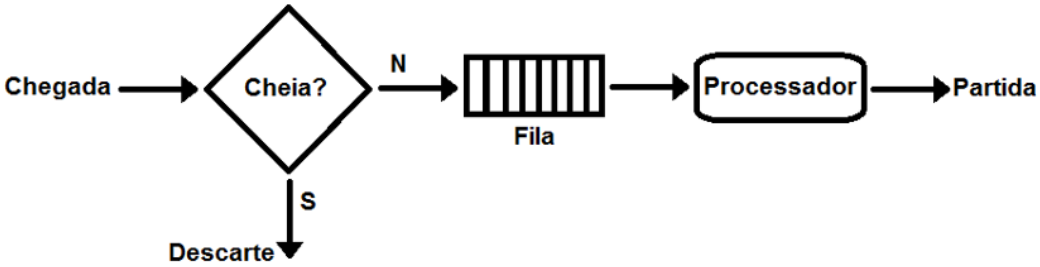

Figura 10 - Fila FIFO

Fonte: Forouzan 2006

- ***Class Based Weighted Fair Queueing* (CBWFQ):** Este tipo de enfileiramento se propõe a oferecer uma alocação de banda mais justa entre os fluxos de dados. Para que aconteça esta ordenação, as mensagens são ordenadas em sessões, e, para cada sessão, um canal é alocado. Este procedimento provê uma alocação de banda mais justa entre os fluxos.

  A classificação dos fluxos de dados pode ser realizada de várias formas, como por endereço origem ou destino, por protocolo, pelo campo precedência IP, pelo par porta/*socket*, dentre outros. Pode-se configurar a quantidade de filas, e a ponderação pode ser definida por precedência IP ou em conjunto com outros protocolos de QoS como RSVP, VoPF (*Voice over Frame Relay*), parâmetros FECN (*Forward Explicit Congestion Notification*), BECN (*Backward Explicit Congestion Notification*) e DE (*Discard Eligible*).

  O CBWFQ (*Class Based Weighted Fair Queueing*) apresenta uma solução para as limitações das versões anteriores a este algoritmo de enfileiramento, como por exemplo a não escalabilidade quando a quantidade de fluxo na rede aumenta, estendendo a funcionalidade padrão de WFQ (*Weighted Fair Queueing*) para prover sustentação para classes de tráfego.

  A Figura 10 mostra a operação do CBWFQ, onde ele permite definir as classes de tráfego e aplicar os parâmetros, como largura de banda e limites de fila a estas classes, permite definir listas do controle de acesso (ACLs) e interfaces de entrada.

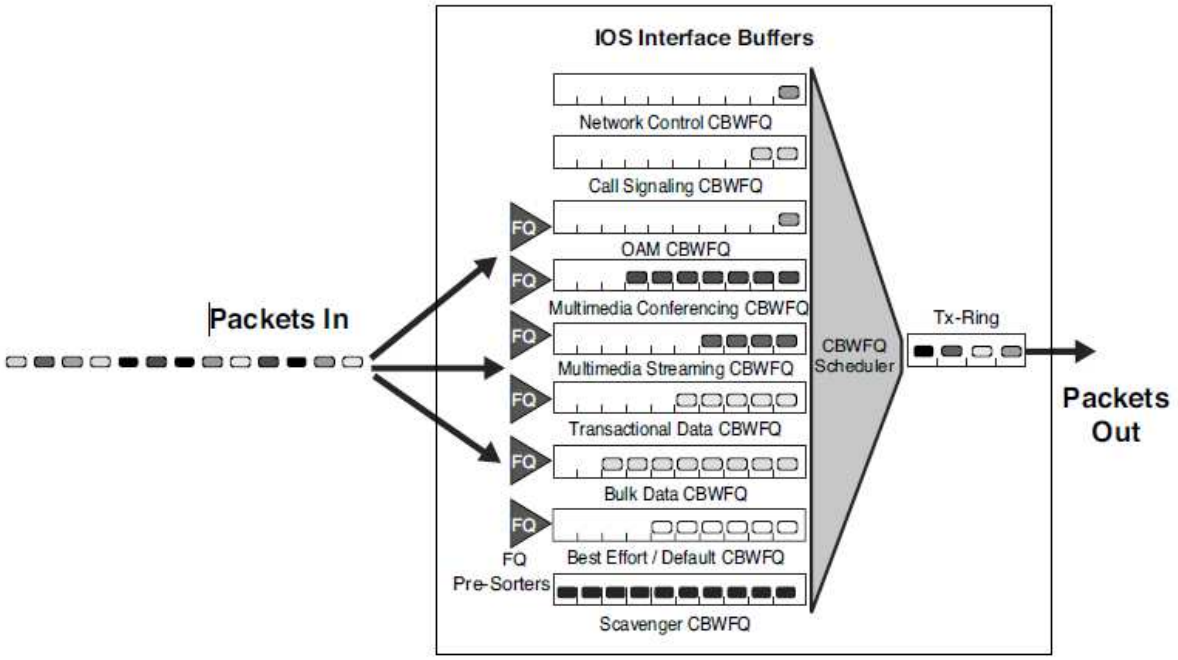

Figura 11 - Operação CBWFQ

Fonte: SZIGETI et al., 2014

O CBWFQ permite a criação de até 256 filas, servindo até 256 classes de tráfego. Cada fila é atendida com base na largura de banda atribuída a essa classe.

A largura de banda atribuída a uma classe é utilizada para calcular o "peso" desta classe. Esse peso é proporcional à largura de banda configurada para cada classe. O peso é, mais precisamente, uma função da largura de banda da interface dividida pela largura de banda da classe. Consequentemente, quanto maior o parâmetro de largura de banda, menor o peso. O WFQ é aplicado preferencialmente às classes (que incluem vários fluxos) do que aos próprios fluxos [SZIGETI et al., 2014].

O CBWFQ é configurado usando a palavra-chave "*bandwidth*" em um mapa de políticas. Com CBWFQ, uma largura de banda mínima é explicitamente definida e aplicada. A largura de banda pode ser especificada em termos absolutos ou percentuais (palavra-chave "*percent*").

- **Enfileiramento *Priority Queueing* (PQ):** O Enfileiramento prioritário - *Priority Queueing* (PQ) foi preparado para dedicar maior prioridade de enfileiramento aos tráfegos de dados que demandam uma certa urgência em seu processamento (sensíveis ao retardo). Este enfileiramento classifica o tráfego de entrada em quatro níveis

(Figura 11): alta, média, normal e baixa. Os pacotes não classificados recebem sua marcação, por default, como normal [SILVA, 2000].

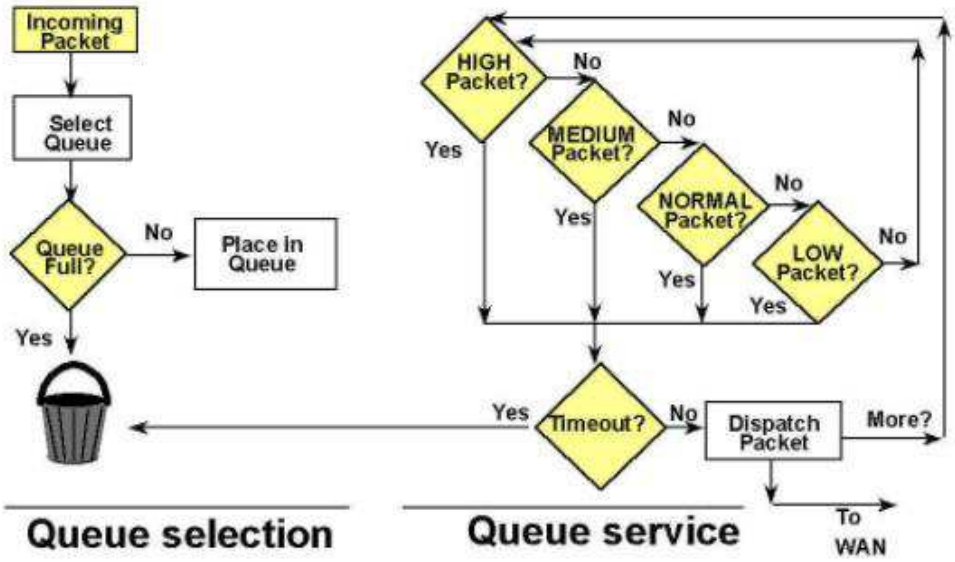

Figura 12 - Enfileiramento *Priority Queueing*

Fonte: Silva, 2000

O tráfego classificado e marcado como prioritário tem preferência absoluta durante a transmissão, logo, este mecanismo requer muito cuidado, para preservar-se de longos atrasos e aumento de *jitter* nas aplicações que requerem menor prioridade. Num pior caso, um tráfego de menor prioridade pode até nunca ser transmitido se o de maior prioridade tomar toda a banda. Este problema tem maiores chances de ocorrer em conexões de baixa velocidade. A fila *default* deve sempre estar habilitada para receber o fluxo não classificado, caso contrário, todo este fluxo que não se encontra em uma lista de prioridades (não classificado) também poderá não ser enviado.

A classificação do tráfego em uma fila PQ pode acontecer por várias opções, por exemplo por protocolo (IP, IPX, DecNet, SNA, etc.), por Lista de Acesso ou por interface de entrada.

- ***Low Latency Queueing* (LLQ):** O LLQ é essencialmente o CBWFQ combinado com um PQ único e rigoroso. Esse recurso permite que você configure a largura de banda como uma porcentagem em enfileiramento de baixa latência [SZIGETI et al., 2014].

O tráfego atribuído à fila de prioridade restrita, usando a palavra-chave "*priority*", é servido até sua largura de banda atribuída, antes que outras filas do CBWFQ sejam atendidas. Todo o tráfego em tempo real deve ser configurado para ser

atendido pela fila de prioridade. Podem ser definidas várias classes de tráfego em tempo real e atribuir garantias de largura de banda separadamente para cada uma, mas uma única fila de prioridade escala todo esse tráfego combinado.

Tal como acontece com CBWFQ, o LLQ pode ser configurado com largura de banda especificada em termos absolutos ou baseada em porcentagem alocações (palavra-chave "*priority*"); outras classes de tráfego recebem garantias de largura de banda por meio do CBWFQ.

A Figura 12 mostra a operação do LLQ. Neste exemplo, três classes de tráfego em tempo real estão em funcionamento na fila prioritária LLQ, enquanto outras classes de tráfego são servidas pela fila do algoritmo CBWFQ.

Neste exemplo foi fornecido uma configuração que corresponde ao fluxo de tráfego agregado de 10 Mbps de tráfego em tempo real na fila de prioridade única, entretanto cada classe é policiada individualmente com 1 Mbps para VoIP, 4 Mbps para vídeo de transmissão e 5 Mbps de vídeo interativo.

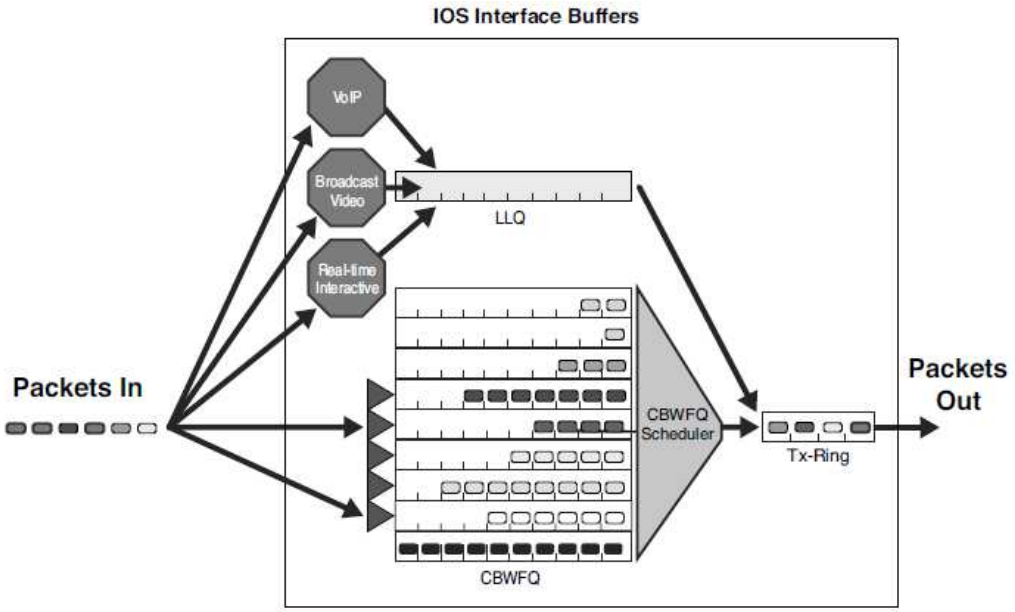

Figura 13 - Operação LLQ
Fonte: SZIGETI et al., 2014

- ***Weighted Random Early Detection* (WRED):** Lidar com o congestionamento após sua detecção inicial é mais eficaz do que permitir que o congestionamento se consolide e depois tentar lidar com ele. Essa observação nos leva à ideia de descartar pacotes antes que todo o espaço dos buffers realmente se esgote [TABNENBAUM, 2003].

O mecanismo de controle de congestionamento RED (*Random Early Detection*) consiste em não esperar que o buffer de um Gateway ou Roteador fique cheio para constatar a sobrecarga, mas a detecção é feita antes mesmo desta sobrecarga acontecer, ou seja, através do monitoramento e controle do tamanho da fila acontece a prevenção da ocorrência do congestionamento.

As funções de controle de congestionamento TCP são utilizadas neste algoritmo, junto ao monitoramento do tráfego antecipadamente, indicando para a origem reduzir a taxa de transmissão, amenizando assim, as situações de congestionamento antes mesmo dos picos de tráfego. Ao ser aplicado a uma interface, o RED descarta pacotes a uma taxa que pode ser configurada.

Uma variação do RED foi desenvolvida e implementada pela Cisco, o *Weighted Random Early Detection* (WRED), esta implementação é uma extensão ponderada do mecanismo RED e combina as funcionalidades do RED com a classificação de pacotes por precedência IP, ou seja, a aleatoriedade da seleção dos pacotes para descarte é inibida pelos pesos dos valores de precedência sinalizados nos pacotes [SZIGETI et al., 2014]. Os pesos definidos a cada fluxo são baseados na política de QoS que se deseja implementar, descartando pacotes de forma seletiva baseado na classificação por precedência IP, onde os pacotes de menor prioridade são descartados inicialmente.

É possível desabilitar a classificação precedência IP e habilitar o descarte baseando-se apenas no tamanho do buffer da fila. WRED é normalmente usado em roteadores centrais (*core routers*), com precedência IP habilitada pelos roteadores de borda (*edge routers*), mas esse mecanismo pode ser útil em qualquer interface onde exista a possibilidade de congestionamento.

Os pacotes com valores de precedência inferiores são descartados de forma mais agressiva do que os pacotes sinalizados com valores de precedência mais elevados, por exemplo, o IPP 1 seria descartado de forma mais agressiva que o IPP 6.

Os pesos WRED baseados em DSCP são indicados pelos valores de preferência de descarte garantidos (AF) e os pacotes de preferência de descarte de AF maiores é superior, por exemplo, AF23 estatisticamente foi descartado mais rapidamente que AF22, que por sua vez foi descartado mais agressivamente do que AF21.

2.3.4.4 Moldagem e Policiamento do Tráfego

A moldagem de tráfego está relacionada à regulagem da taxa média (e do volume) da transmissão de dados [TANENBAUM; WETHERALL, 2011], adequando o tráfego da rede a fim de evitar ou amenizar problemas de congestionamentos futuros. O mecanismo *Generic Traffic Shaping* (GTS) provê mecanismos para controle de tráfego utilizando filtros conhecidos como *token bucket* (balde de fichas), esse mecanismo limita o tráfego de saída de uma interface a uma determinada taxa. O fluxo classificado vai para um buffer limitador, sendo liberado por regras pré-definidas de acordo com uma política de controle de tráfego, podendo ser configurada pelo administrador ou derivada da interface.

A moldagem de tráfego pode ser utilizada em várias situações, servindo para limitar o tráfego de rajada sem prejudicar o tráfego prioritário, reduzir a latência; ou, em casos de congestionamento, limitar um determinado tipo de tráfego não sensível ao retardo, como por exemplo, transferências de arquivos, eliminando possíveis gargalos.

A moldagem GTS é aplicada somente em interfaces de saída, com o uso de Listas de Controle de Acesso para classificar e selecionar o tráfego. Trabalha com qualquer tecnologia de enlace, como Frame Relay, ATM (*Asynchronous Transfer Mode*), SMDS (*Switched Multimegabit Data Service*) e Ethernet.

O policiamento corresponde ao conjunto de regras aplicadas nos nós de borda ao tráfego que entra ou sai do domínio para verificar a conformidade com parâmetros de serviço especificado [MELO, 2005]. O tráfego pode ser atrasado descartando ou remarcando pacotes de menor prioridade dependendo do tipo de policiamento em uso e da sua parametrização.

Dois algoritmos bastante utilizados para o policiamento de tráfego em redes de pacotes são o método do balde furado (*leaky bucket*) e o método do balde de tokens/fichas (*token bucket*).

- **Método do Balde Furado (*Leaky Bucket*):** Este método é comparado a um balde com um pequeno furo no fundo. A velocidade com que a água entra no balde não interfere no fluxo de saída através do furo, a saída sempre ocorrerá a uma taxa constante. Aplicando ainda a analogia, a água que entrar enquanto o balde estiver cheio se perderá.

  A mesma ideia pode ser aplicada a pacotes. Conceitualmente, cada host está conectado à rede por uma interface que contém um balde furado, ou seja, uma fila interna finita. Se um pacote chegar à fila quando ela estiver cheia, este pacote será descartado. Ou seja, se um ou mais processos dentro do host tentar enviar um pacote

quando o número máximo já estiver enfileirado, o novo pacote será descartado. [TANENBAUM; WETHERALL, 2011].

Este processo busca policiar a taxa de entrada de pacotes na rede por meio do controle da frequência com que os pacotes são encaminhados, garantindo que a vazão dos pacotes seja uniforme, atenuando as flutuações e reduzindo a possibilidade de congestionamento na rede. Quando todos os pacotes têm o mesmo tamanho, o algoritmo tem um bom desempenho, caso os pacotes sejam de tamanhos distintos, a melhor opção é permitir a passagem de um número fixo de bytes de unidade de tempo.

A Figura 13 mostra o processo do método do balde furado, onde os pacotes têm um padrão de saída rígido, onde a velocidade de entrada é passível de variação, entretanto a velocidade de saída será constante, transformando assim um fluxo de pacotes irregular em um fluxo regular.

- **Método do Balde de fichas (*Token Bucket*):** O método do Balde Furado impões padrões rígidos à taxa média sem se preocupar com a irregularidade do tráfego. Existem situações em que é necessário que se aumente a vazão de pacotes. O método do Balde de Símbolos serve para contornar esse problema. Nesse algoritmo, o balde retém fichas (tokens), gerados a cada pulso de clock numa variação de T segundos [TANENBAUM; WETHERALL, 2011].

  Quando um pacote chega na fila de saída do enlace, é verificado se existe alguma ficha dentro do balde, se houver, o pacote é encaminhado e a ficha descartada. Imaginando a situação em que o tamanho do balde é n e chega uma rajada repentina de tamanho n, o algoritmo do Balde de Fichas dá vazão a esses pacotes conjuntamente, mas por um tempo determinado.

  Quando o balde enche, o algoritmo descarta as fichas, mas nunca os pacotes. Uma opção para tornar o tráfego mais constante, seria a associação do balde furado após o balde de fichas. A união desses dois algoritmos evita o problema da ociosidade solucionado pelo balde de símbolos e mantém o fluxo constante, realizado pelo balde furado.

2.3.4.5 Controle de Admissão

Neste ponto, o tráfego de entrada de algum fluxo já está bem modelado e pode potencialmente seguir uma única rota na qual a capacidade pode ser reservada com antecedência nos roteadores ao longo do caminho. Quando este fluxo é proposto ao roteador,

ele deve decidir, baseado na quantidade de compromisso que já assumiu e em sua capacidade se deve, ou não, aceitar ou rejeitar o fluxo [TANENBAUM; WETHERALL, 2011].

Se o fluxo for aceito, o roteador solicita ao Classificador de Pacotes e ao Programador de Pacotes para reservarem a QoS para esse fluxo. Cada roteador, ao longo do caminho, solicita o Controle de Admissão para analisar se tem ou não condições de aceitar o novo fluxo [MACEDO, 2010].

O Controle de Admissão é um mecanismo essencial para o *IntServ,* visto que, como dito em tópicos anteriores, para reservar recursos num roteador, o RSVP estabelece uma comunicação com os módulos locais de controle de admissão, determinando se o nó tem ou não capacidade para aplicar a QoS solicitada.

3 EXPERIMENTOS

Neste capítulo, serão realizados os testes, a fim de comprovar a finalidade da aplicação da Qualidade de Serviços, em especial, DiffServ. Os Serviços diferenciados (DiffServ) fornecem um padrão de operação e construção, e possibilita a implantação de diferenciação de serviços escaláveis na rede.

Como foi visto nos capítulos anteriores que serviram de base para a fundamentação teórica neste trabalho, a arquitetura DiffServ especifica um mecanismo simples e escalável para classificação e gerenciamento do tráfego, provendo qualidade de serviço em rede IP.

O cenário para deste experimento será observações em dois ambientes, ambiente computacional utilizando ferramentas para simulação e ambiente real, utilizando equipamento para observar o comportamento do tráfego na rede após a aplicação das técnicas de QoS.

Apesar de o DiffServ ser uma arquitetura recomendada para grandes redes, isto dar-se-á principalmente por efeito de sua escalabilidade, foi utilizada uma topologia pequena, devido a algumas limitações no aumento da carga de tráfego gerada no ambiente computacional simulado. Como este trabalho tem o objetivo de realizar duas simulações, uma real e outra computacional, foi utilizada uma carga de tráfego para gerar congestionamento equivalente em ambas simulações, a fim de comparar a funcionalidade da aplicação dos Serviços Diferenciados nos dois cenários.

3.1 Topologia da rede

Será apresentada o cenário da topologia utilizada para os testes. Trata-se de uma rede de uma pequena empresa, com um link WAN de 2 Mbps. A Figura 13, a seguir, mostra a topologia da rede. Dois roteadores interligando duas redes LAN.

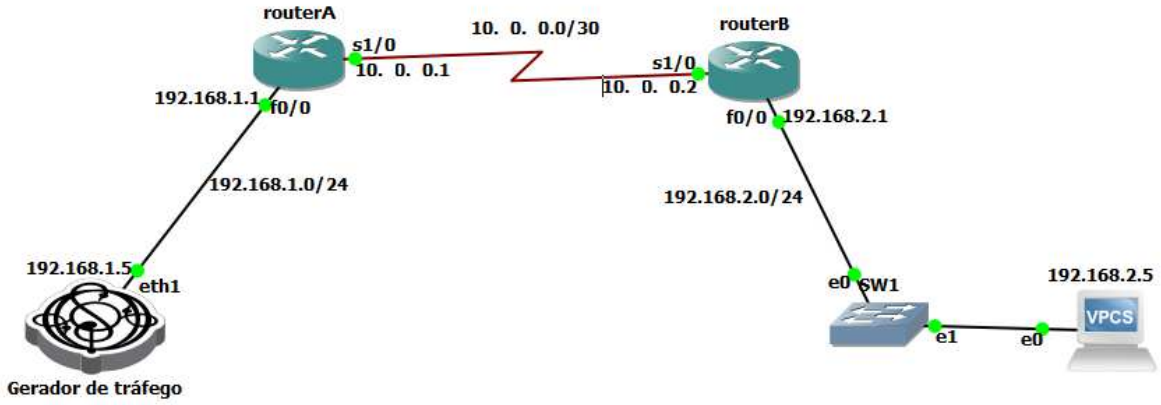

Figura 14 - Topologia da rede

Fonte: Autor

Endereçamento IP

Para o provimento de comunicação e operabilidade entre as empresas, seguem os dados de endereçamento das interfaces:

- Sub-rede WAN:
  Rede: 10.0.0.0/30
  Mácara: 255.255.255.252
  IP Wan RouterA: 10.0.0.1
  IP Wan RouterB: 10.0.0.2

- Sub-rede A
  Rede: 192.168.1.0/24
  Máscara: 255.255.255.0
  Gateway: 192.168.1.1
  PC Gerador de tráfego: 192.168.1.5

- Sub-rede B
  Rede: 192.168.2.0/24
  Máscara: 255.255.255.0
  Gateway: 192.168.2.1
  PC Receptor de tráfego: 192.168.2.5

3.2 Simulação Computacional

Para a simulação computacional foram utilizados fundamentalmente duas ferramentas, uma para emulação da imagem dos roteadores e outra para simulação do tráfego na rede.

3.2.1 Ferramentas utilizadas

- **GNS3**

  Para emular as imagens dos roteadores foi utilizado o software GNS3. O GNS3 é uma ferramenta complementar aos laboratórios reais. O simulador GNS3 é um software gratuito, *open source*. As imagens IOS de alguns roteadores são emuladas através de uma ferramenta chamada Dynamips. O GNS3 age como interface gráfica do Dynamips. A figura 14 mostra a tela inicial de trabalho do GNS3, onde são gerados os projetos da rede.

Na figura 14, o campo 1 destacado mostra a lista de dispositivos que já foram instalados e podem ser utilizados nas simulações, como roteadores, switchs, PC, etc. O campo 2 mostra a topologia criada. O campo 3 sumariza todos os dispositivos que estão incluídos no projeto criado e mostra seu status. O campo 4 mostra os servidores envolvidos no projeto.

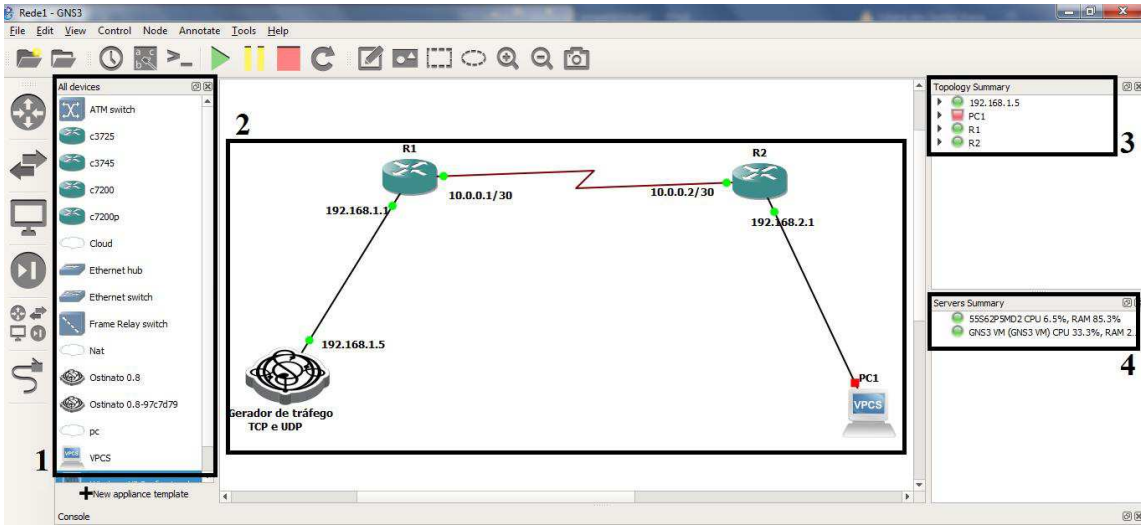

Figura 15 – Tela inicial do GNS3

Fonte: Autor

- **OSTINATO**

Para injetar o tráfego na rede, utilizada a ferramenta Ostinato, que é um organizador de pacotes e gerador de tráfego. É uma poderosa API (Application Programming Interface) Python para automação de testes de rede. Envia pacotes de vários fluxos com diferentes protocolos a diferentes taxas. Ostinato é um software de código aberto licenciado sob GNU GPLv3.

O Ostinato pode ser importado para o GNS3. Para efetuar essa integração, foi instalada a aplicação Ostinato dentro da ferramenta GNS3 com o auxílio de uma máquina virtual GNS3 VM. O GNS3 VM é uma VM Ubuntu, neste caso foi executada no VMWorkstation (versão trial) na máquina local, permitindo executar imagens IOS dentro dele, já que o Windows não pode executar nativamente o IOU.

O ambiente teve dois roteadores Cisco modelo 7200, processador com 491520K/32768K bytes de memória. Casa roteador contém duas interfaces FastEthernet, quatro interfaces Seriais, 509K bytes de NVRAM e IOS na versão 12.4.

Para integrar/simular computadores no ambiente, foi utilizado o VPCS (Virtual PC Simulator).

Em cada roteador foram implantadas as técnicas dos Serviços Diferenciados. Como trata-se de duas empresas se comunicando, as configurações de QoS foram implantadas em ambos roteadores, visto que existe a comunicação entre as empresas acontece nas duas direções. Esta implementação ocorrerá aplicando o serviço *Assured Forwarding*, condicionando o tráfego de acordo com suas políticas, marcando, classificando e dando o devido tratamento a cada pacote de acordo com seu valor DSCP.

Foram criados alguns perfis de tráfego no Ostinato, como fluxo TCP e UDP, nomeados como tráfego de voz, corporativo, dentre outros fluxos de protocolos de sinalização e gerenciamento para que fossem injetados na rede em um determinado intervalo de tempo.

As interfaces de configuração do gerador de tráfego Ostinato, assim como suas principais funcionalidades na elaboração de um perfil para gerar tráfego serão descritas abaixo.

A Figura 15 mostra a interface inicial onde pode ser vista na parte superior da imagem a lista dos perfis de tráfegos criados, foram criados seis perfis (voz, telnet, icmp, corp, ssh e outros dados) cada um com configurações distintas. Mais abaixo, nesta mesma figura, aparecem as portas disponíveis para origem do tráfego.

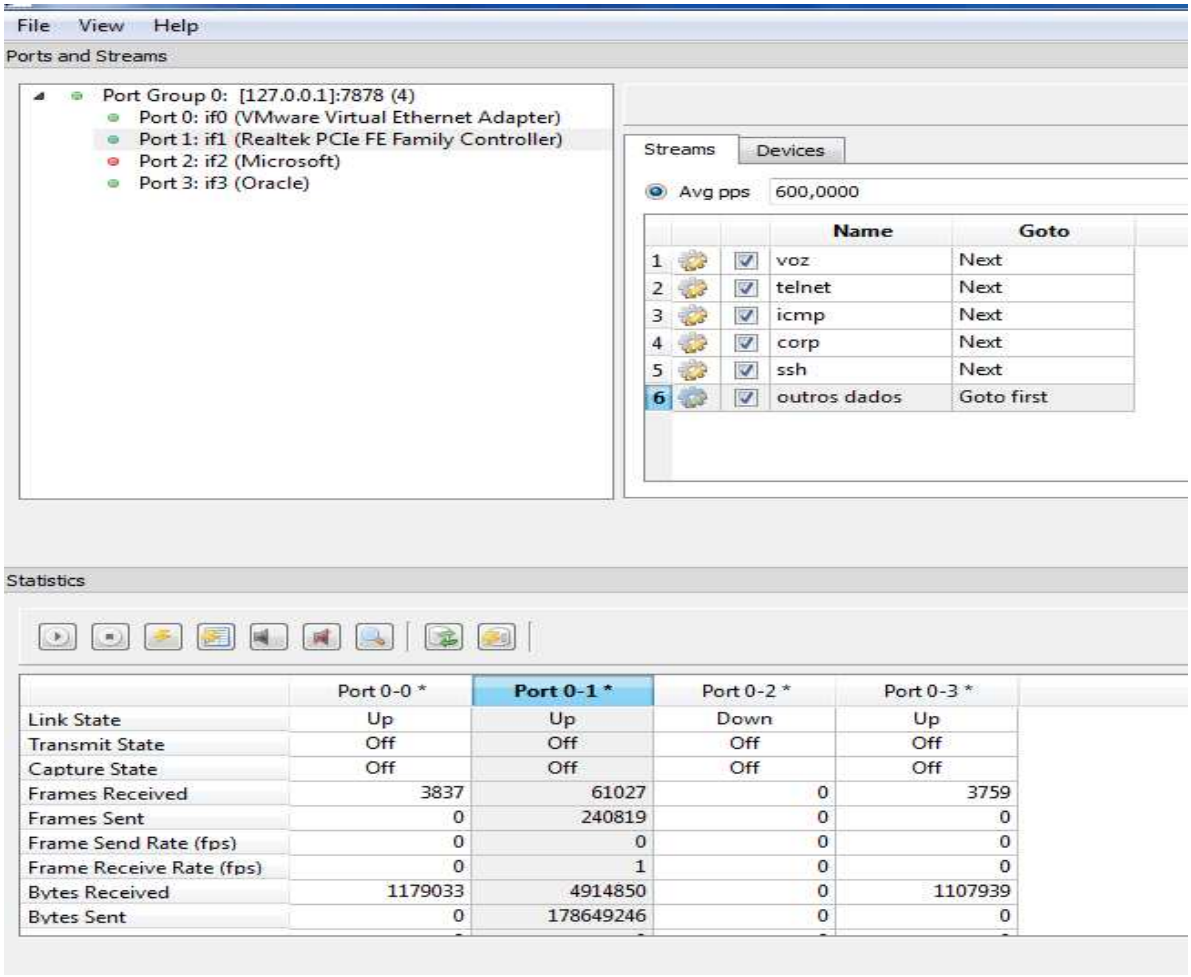

Figura 16 - Interface inicial do Ostinato

Fonte: Autor

Na interface mostrada na Figura 16, a seguir, na parte superior foi configurado o comprimento dos quadros que serão gerados nesse perfil. O caso mostrado na Figura 15 gera um tráfego com *frames* de comprimentos randômicos, variando entre os tamanhos de 64 e 1200. Mais abaixo nesta mesma imagem foi configurado o perfil geral do pacote de acordo com cada camada, no nosso exemplo na camada de rede (L3) foi selecionado o protocolo IPv4, e na camada de transporte (L4) o protocolo UDP.

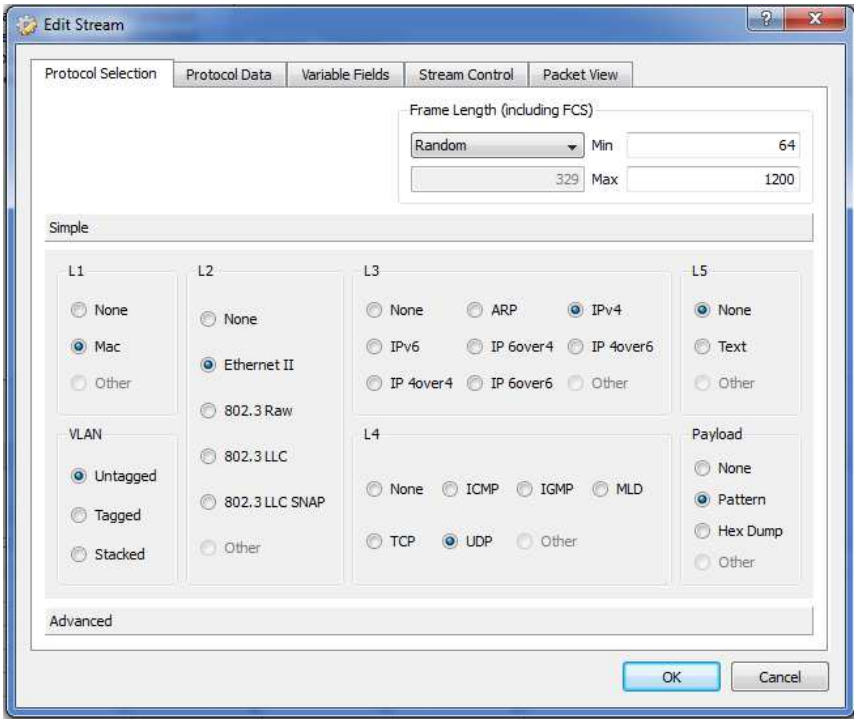

Figura 17 - Aba de modelagem das definições iniciais do pacote

Fonte: Autor

Na interface apresentada pela Figura 17, são mostrados os endereços MAC de origem e destino do tráfego. O MAC de origem é o endereço físico da onde o tráfego foi gerado, o MAC de destino é o endereço físico da porta do roteador pela qual o tráfego irá entrar.

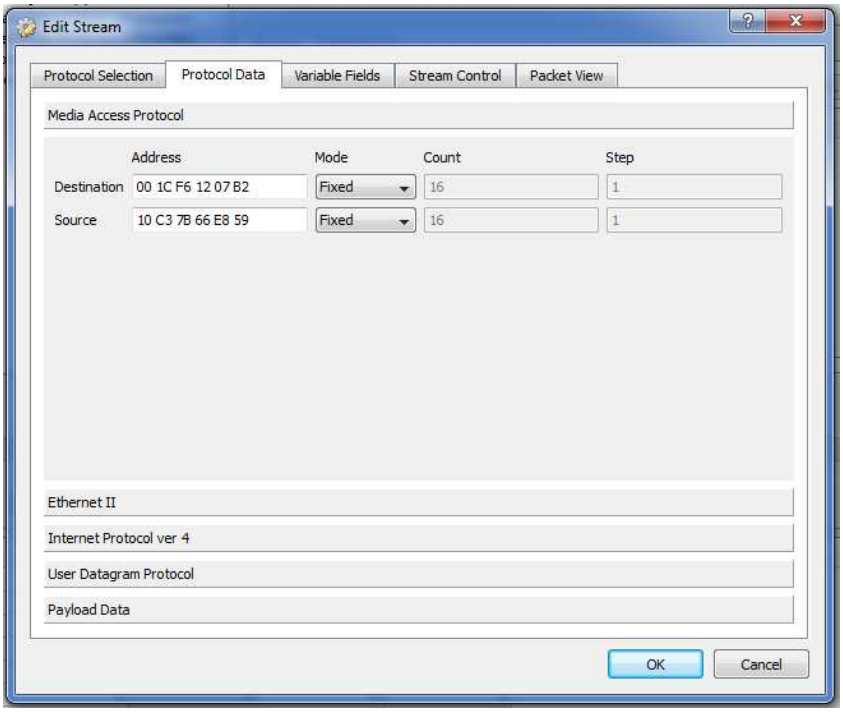

Figura 18 - Aba de definição de origem e destino (camada 2)

Fonte: Autor

Na Figura 18 tem-se a interface onde são mostrados os endereços IP de origem e destino do tráfego, que pode ser configurado de forma fixa ou dinâmica.

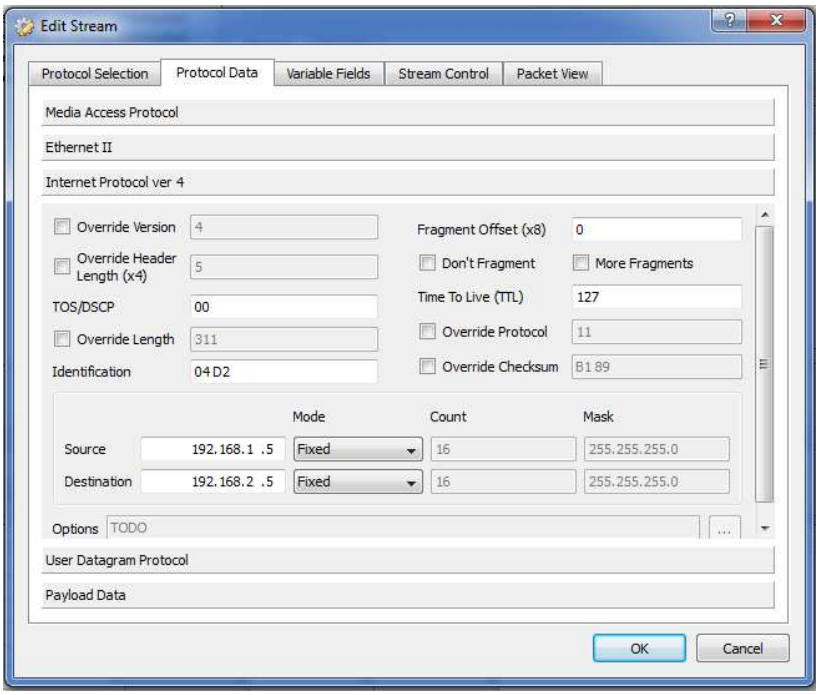

Figura 19 - Aba de definição de origem e destino (camada 3)

Fonte: Autor

Na interface mostrada na Figura 19, tem-se as opções de controle de *stream*. No caso mostrado na figura, o tráfego terá 900 pacotes gerados na taxa de 600 pacotes/segundo.

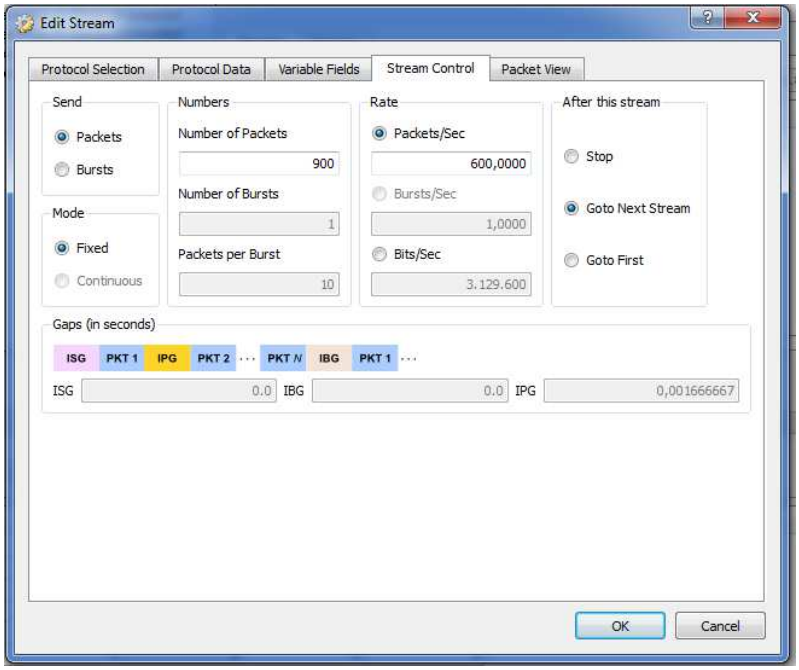

Figura 20 - Configuração de controle do *stream*

Fonte: Autor

3.2.2 Classificação, Marcação e Priorização de Tráfego

A classificação do tráfego de entrada do roteador é feita por meio da aplicação de listas de controle de acesso e do mapeamento de classes (class-map) e logo após passa pela marcação do campo DSCP do pacote e, na interface de saída, é feita uma análise do campo DSCP e definida a prioridade da banda disponível na saída.

As Listas de acesso ou lista de controle de acesso têm a finalidade de permitir, ou negar, um serviço, por exemplo, um determinado IP, servidores, impressora, aplicativo, ou seja, qualquer dispositivo na rede. Estas listas realizam a classificação por endereços IP e/ou portas da aplicação ou outro requisito que caracterize o tráfego a ser priorizado.

A figura 20 mostra o exemplo de uma a lista de acesso que foi utilizada para classificar o tráfego. A classificação pode ser realizada baseando-se em portas físicas, IP origem e destino, porta de aplicação, tipo de protocolo e outros critérios que utilizam lista de acessos. A lista de acesso mostrada na figura 20, marca o tráfego de entrada, baseando-se na porta 80 (www) ou pacotes correspondentes que já tem valor dscp atribuído, neste caso, dscp AF31.

Exemplo de uma Listas de Controle de Acesso (ACL)

```
ip access-list extended Trafego_AF31
 permit ip any any dscp af31
 permit tcp any any eq www
 permit tcp any eq www any
```

Figura 21 – Exemplo de listas de Controle de Acesso (ACL)

Fonte: Autor

O tráfego de rede pode ser classificado com base em uma série de critérios de correspondências de fluxo especificadas por um class-map. Cada class-map define uma classificação de tráfego. A Figura 21 mostra a criação de dois class-maps, uma utilizada na interface de entrada de tráfego do roteador, servindo para classificar o tráfego de acordo com o tráfego identificado pela ACL “Trafego_AF31”, e a outra aplicada na interface de saída WAN, correspondendo o pacote a um valor DSCP, neste caso o valos DSCP AF31.

Exemplo de mapeamento de classes

```
class-map match-any Marcacao_AF31
 match access-group name Trafego_AF31
class-map match-any Classe_AF31
 match ip dscp af31
```

Figura 22 - Exemplo de mapeamento de classes

Fonte: Autor

O comando *policy-map* é utilizado para configurar as características de QoS que devem estar associadas ao tráfego previamente classificado. Um *policy-map* define uma série de ações (funções) que se deseja aplicar a um fluxo de tráfego previamente classificado. Foram elaboradas duas *policy-maps* (Figura 22), uma para ser aplicadas na interface de entrada (EntradaLAN (entrada_de_tráfego)) e outra para ser aplicada na interface de saída do roteador (SaidaWAN (fa0/1)).

Políticas de marcação de pacotes

```
policy-map EntradaLAN(entrada_de_tráfego)
class Marcacao_AF31
  set ip dscp af31

policy-map SaidaWAN(fa0/1)
class Classe_AF31
  priority percent 7
```

Figura 23 - Políticas de marcação de pacotes

Fonte: Autor

3.3 Simulação Real

Na simulação com equipamentos reais, foram utilizados dois roteadores modelo Cisco 1841 (figura 23), com duas interfaces FastEthernet, 191 bytes de NVRAM, IOS na versão 12.4. Para compor as LANs das duas sub-redes utilizou-se dois notebooks ligados aos roteadores.

Figura 24 - Roteador Cisco modelo 1841

Fonte: Internet[1]

[1] Disponível em: https://tipidpc.com/viewitem.php?iid=40814342

Foram encontradas limitações na montagem do laboratório real pois um dos roteadores não tinha sua interface serial para simular um link WAN. Diante disso, foram utilizadas as portas FastEthernet para aplicação das políticas de tráfego para o provisionamento da qualidade de serviço.

As configurações como ACL, *class-map* e *policy-map* aplicadas nos roteadores reais foram as mesmas aplicadas na simulação computacional, afim de ao final dos testes, fazermos as comparações entre as simulações nos dois ambientes. Para injetar o tráfego na rede, também foi utilizada a ferramenta Ostinato, instalada no notebook que simula a LAN de onde o tráfego é gerado.

3.4 Comparação de cenários

Apesar da simulação em ambos cenários ser composta das mesmas configurações e mesma topologia. Foram identificadas algumas limitações com relação à simulação computacional.

Inicialmente foi feito um experimento com perfis de tráfego semelhantes em ambos ambientes, computacional e real. Foi injetada uma carga de tráfego igual nos dois ambientes e ambos mostraram comportamento semelhantes. Entretanto, ao aumentar a carga, a simulação com os equipamentos reais pôde suportar uma carga consideravelmente maior, e ainda assim atingir o objetivo de provimento de qualidade de serviço garantindo a classificação, moldagem e policiamento do tráfego.

Na simulação computacional foi necessário reduzir de forma equivalente o tráfego, para que se pudesse observar por um determinado tempo a ação das arquiteturas de QoS sendo aplicadas.

4 RESULTADOS

Os resultados foram analisados por meio dos contadores de pacotes na interface do roteador, verificando os dados entregues como resposta ao solicitar o status da interface serial onde estão aplicadas as políticas de QoS.

O comando show “policy-map interface” traz o resultado dos contadores de pacotes da interface serial do roteador, por onde os dados irão seguir pela rede. Podem ser vistas nas figuras uma parte dos resultados trazidos pelos contadores após um período de tráfego na rede.

Nas Figuras 24 e 25 os resultados da “class-map Voz” podem ser observados. em ambos os casos, simulação computacional e real, onde foram aplicados os mesmos perfis de tráfegos em período de tempo semelhante. Observou-se que as respostas trazidas por ambos os roteadores são semelhantes, visto que tanto o tamanho dos pacotes quanto a configuração das taxas de *streams* gerados são as mesmas.

A informação “**5590 packets, 308851 bytes**”, extraída da figura 24 traz o número de pacotes que atenderam aos critérios da classe “class-map Voz” aplicada na interface por meio de uma policy-map. Este contador é incrementado independentemente de a interface estar ou não congestionada. A linha 2 mostra que nenhum pacote desta classe foi descartado durante o teste, isto ocorre devido a aplicação das políticas de prioridade nesta classe.

Também pode-se observar nas figuras 24 e 25, por meio da informação “**drop rate 0 bps**” que não houve perda de pacotes em nenhum dos casos atendidos pela “class-map Voz”. O campo de resultado “(**pkts matched/bytes matched)**”, mostrados em ambas as imagens apresentam a quantidade de pacotes que se encaixam nos critérios definidos pela classe durante o congestionamento da interface, ou seja, estes pacotes excedentes foram enfileirados pelo driver em conjunto com o processador L3, às quais as políticas de serviço se aplicam. Os pacotes que são comutados por processo passam sempre pelo sistema de enfileiramento L3 e, portanto, incrementam o contador de "pacotes correspondentes" (pkts matched).

Resultado: show policy-map Serial 0/1 – Simulação Computacional

```
Serial 1/0

 Service-policy output: SaidaWAN(fa0/1)

  Class-map: Voz (match-any)
    5590 packets, 308851 bytes
    5 minute offered rate 14000 bps, drop rate 0 bps
    Match: ip dscp ef (46)
      5590 packets, 308851 bytes
      5 minute rate 14000 bps
    Queueing
      Strict Priority
      Output Queue: Conversation 264
      Bandwidth 30 (%)
      Bandwidth 2560 (kbps) Burst 64000 (Bytes)
      (pkts matched/bytes matched) 5433/300234
      (total drops/bytes drops) 0/0
```

Figura 25 - Resultado da class-map voz na simulação computacional

Fonte: Autor

Resultado: show policy-map Fa1/0 – Simulação Real

```
FastEthernet 0/1

 Service-policy output: SaidaWAN(fa0/1)

  Class-map: Voz (match-any)
    6711 packets, 445702 bytes
    5 minute offered rate 18000 bps, drop rate 0 bps
    Match: ip dscp ef (46)
      6711 packets, 445702 bytes
      5 minute rate 18000 bps
    Queueing
      Strict Priority
      Output Queue: Conversation 264
      Bandwidth 30 (%)
      Bandwidth 600 (kbps) Burst 15000 (Bytes)
      (pkts matched/bytes matched) 6709/445586
      (total drops/bytes drops) 0/0
```

Figura 26 - Resultado da class-map voz na simulação real

Fonte: Autor

A Figura 26 mostra a “class-map BestEffort”, esta classe tem seu valor DSCP padrão igual a zero (0), ou seja, não existe priorização de tráfego para esta classe. Devido a isto, houve perda de pacotes nesta classe. Essa perda pode ser observada através da informação “**drop rate 77000 bps**”.

```
Class-map: BestEffort (match-any)
  8340 packets, 2877022 bytes
  5 minute offered rate 80000 bps, drop rate 77000 bps
  Match: ip dscp default (0)
    8340 packets, 2877022 bytes
    5 minute rate 80000 bps
  Queueing
    Output Queue: Conversation 268
    Bandwidth 6 (%)
    Bandwidth 614 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 996/231100
    (depth/total drops/no-buffer drops) 0/367/0
  police:
      cir 32000 bps, bc 1500 bytes
    conformed 1004 packets, 231956 bytes; actions:
      set-dscp-transmit default
    exceeded 7336 packets, 2645066 bytes; actions:
      drop
    conformed 12000 bps, exceed 75000 bps
```

Figura 27 - Resultado da classe BestEffort - Simulação computacional

Fonte: Autor

Foi realizado na rede simulada pelos equipamentos reais um teste adicional aplicando os mesmos perfis de tráfego na rede, entretanto com uma carga maior e por um período de tempo mais prolongado. O resultado obtido pelos contadores de pacotes na interface do roteador é mostrado no Apêndice A.3.

A simulação real trouxe um resultado mais satisfatório em relação ao provimento da qualidade de serviço baseado nos valores DSCP marcados nos pacotes. Houve perda de pacotes somente na classe *Best Effort*.

A carga de tráfego que são atendidos pela “class-map Voz" foi aumentada propositalmente. Com isso, a quantidade de pacotes que atendem aos requisitos da classe de voz é consideravelmente maior que a quantidade de pacotes que atendem aos requisitos da classe de *BestEffort*, entretanto, não houve nenhuma perda de pacote na classe de voz.

5 CONCLUSÃO

Os mecanismos estudados e avaliados neste trabalho mostram a sua importância no fornecimento de qualidade de serviço em uma rede IP. O conjunto da aplicação dos diversos mecanismos oferecidos pelo DiffServ, policiando os fluxos de acordo com suas respectivas prioridades mostrou que uma rede pode ter um aprimoramento no provisionamento de serviço de qualidade.

Podemos controlar com eficiência os requisitos de rede definidos no início deste trabalho, como largura de banda, latência e perda de pacotes. Garantindo os resultados, os mecanismos de QoS conduzem a serviços eficientes e previsíveis para aplicações de fundamental importância na instituição que o aplica. A utilização de QoS permitirá a oferta de gerenciamento, banda por demanda entre outros serviços disponíveis.

A simulação do projeto de uma rede de comunicações de uma pequena empresa é efetivamente possível com o auxílio de ferramentas de virtualização de desktops e de emulação de roteadores, desde que sejam observadas suas limitações nos testes de congestionamento de rede.

Apesar deste trabalho ter avaliado os mecanismos de QoS em uma pequena rede montada para este fim, fica a perspectiva de sua utilização em uma rede maior como a Internet, que apesar de exigir maior complexidade de implementação, pode viabilizar a disseminação de aplicações multimídias em tempo real.

Este trabalho pode servir como suporte para a elaboração da linha de base de uma rede. A linha de base é uma medida do comportamento "normal". Muitas redes experimentam " picos de tráfego " em vários momentos relacionados a operações de negócios fundamentais - acesso de correio eletrônico e outros recursos. Uma linha de base é útil para distinguir um dia " ruim ", ou uma anomalia aleatória, dos dias "normais". As linhas de base auxiliam um administrador a apontar uma mudança repentina, que possa indicar um problema na rede. Com o tempo, as linhas de base indicam tendências nas atividades para finalidades de planejamento.

O alvo deste trabalho foi atingido, pois foi mostrado o ganho com a implementação de QoS, demonstrando a eficácia da arquitetura dos Serviços Diferenciados, quando permite a tráfego de fluxos de diversos tipos, mesmo em condições de congestionamento.

5.1 Sugestões para trabalhos futuros

Como trabalhos futuros, são sugeridos os seguintes tópicos:

- A aplicação dos mecanismos aqui apresentados para provimento de qualidade de serviço em tráfego de vídeo e voz em tempo real, aplicando em uma rede mais robusta, como por exemplo, a rede da Universidade Estadual do Maranhão.
- A elaboração da linha de base da rede, a fim de identificar as anomalias na rede em um momento de congestionamento.

REFERÊNCIAS

CISCO SYSTENS. **Cisco IOS Quality of Service Solutions Configuration Guid**, EUA, 2009. Disponível em < http://www.cisco.com/c/en/us/td/docs/ios/qos/configuration/guide/12_2sr/qos_12_2sr_book.pdf >

CISCO, Networking Academy. **CCNA Exploration – Instrodução à Redes**. Cisco Systems, Inc., 2017.

CLARK, D.; SHENKER, S.; ZHANG, L. **Supporting real-time applications in an integrated services packet network: architecture and mechanism**. Proceedings of the SIGCOMM '92 Symposium, p. 14-26, Agosto de 1992.

DUTKIEWICZ, E; BOUSTEAD, P. **Analisis of Per-Flow and Aggregate QoS in Scalable QoS Network**. Procedings, 2002.

FALSARELLA, Douglas. **Qualidade de Serviço - componentes do QoS**, Brasil, 2009. Disponível em: < https://imasters.com.br/artigo/13340/redes-e-servidores/qualidade-de-servico-componentes-do-qos?trace=1519021197&source=admin >

FARREL, Adrian. **A internet e seus protocolos:** uma análise comparativa. Rio de Janeiro: Campus, 2005.

FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores.** 3. ed. Porto Alegre: Bookman, 2006. 840 p.

GABOS, D.; CARVALHO, T.C.M.B. **Tecnologias Convergentes.** São Paulo: Epusp, 2009.

GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social.** 6. ed. São Paulo: Atlas, 2008. 2008 p.

HERSENT, Oliver.; GURLE, David.; PETIT, Jean-Pierre. **Telefonia IP:** Comunicação multimídia baseada em pacotes. São Paulo: Prendice Hall, 2002.

KAMIENSKI, C.A.; SADOK, D.; CAVALCANTI, D. A. T.; SOUZA, D. M. T.; DIAS, K.
KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a internet:** uma abordagem top-down. 5. ed. São Paulo: Addison Wesley, 2010.

L. **Simulando a Internet:** Aplicações na Pesquisa e no Ensino, 21ª JAI (Jornada de Atualização em Informática), Congresso da SBC, Florianópolis/SC, julho de 2002.

MAIA, Luiz Paulo. **Arquitetura de redes de computadores.** 2. ed. Rio de Janeiro: Ltc, 2013. 288 p.

MARTINS, J. **Qualidade de Serviço (QoS) em Redes IP Princípios Básicos, Parâmetros e Mecanismos**, JSMNet Networking Reviews – Vol 1 – Nº 1, 1999.

MELO, J.C de. **Estudo da Utilização de Mecanismos de QoS em Redes com Enlace de Banda Estreita**, 2005. 136p. Dissertação de (Mestrado) - Universidade Federal do Maranhão, São Luís, 2005.

PETERSON, Larry L.; DAVIE, Bruce S. **Redes de computadores:** uma abordagem de sistemas. 5. ed. Rio de Janeiro: Elsevier Editora Ltda, 2013.

RICHTER, J.P.; MEER, H. **Towards Formal Semantics for QoS Support**. Draft, IEEE, 1998.

SILVA, Dinalton José da. **Tecnologia Análise de Qualidade de Serviço em Redes Corporativas.** 2004. 112 f. Dissertação (Mestrado) - Curso de Instituto de Computação, Universidade Estadual de Campinas, Campinas - Sp, Campinas, 2004.

STALLINGS, William. **Redes e Sistemas de Comunicação de Dados:** Teoria e aplicações corporativas. 5. ed. Rio de Janeiro: Campus, 2005.

SZIGETI, Tim et al. **End-to-end qos network design:** Quality of service for rich-media and cloud networks. 2. ed. Indianapolis: Cisco Press, 2014.

SZIGETI, Tim; HATTINGH, Christina. **End-to-end qos network design:** Quality of service in LANs, WANs and VPNs. Indianapolis: Cisco Press, 2004.

TANENBAUM, Andrew S.; WETHERALL, David. **Redes de computadores.** 5. ed. São Paulo: Pearson Prentice Hall, 2011.

APÊNDICES

APÊNDICE A – RESULTADOS DAS SIMULAÇÕES

A.1 Simulação Real

Resultado: show policy-map Fa1/0 – Simulação Real

```
FastEthernet 0/1

 Service-policy output: SaidaWAN(fa0/1)

  Class-map: Voz (match-any)
    6711 packets, 445702 bytes
    5 minute offered rate 18000 bps, drop rate 0 bps
    Match: ip dscp ef (46)
      6711 packets, 445702 bytes
      5 minute rate 18000 bps
    Queueing
      Strict Priority
      Output Queue: Conversation 264
      Bandwidth 30 (%)
      Bandwidth 600 (kbps) Burst 15000 (Bytes)
      (pkts matched/bytes matched) 6709/445586
      (total drops/bytes drops) 0/0

  Class-map: Classe_AF31 (match-any)
    6700 packets, 931970 bytes
    5 minute offered rate 30000 bps, drop rate 0 bps
    Match: ip dscp af31 (26)
      6700 packets, 931970 bytes
      5 minute rate 30000 bps
    Queueing
      Strict Priority
      Output Queue: Conversation 264
      Bandwidth 23 (%)
      Bandwidth 460 (kbps) Burst 11500 (Bytes)
      (pkts matched/bytes matched) 6700/931970
      (total drops/bytes drops) 0/0

  Class-map: Classe_AF32 (match-any)
    1413 packets, 66190 bytes
    5 minute offered rate 4000 bps, drop rate 4000 bps
    Match: ip dscp af32 (28)
      1413 packets, 66190 bytes
      5 minute rate 4000 bps
    Queueing
      Output Queue: Conversation 265
      Bandwidth 7 (%)
      Bandwidth 140 (kbps)Max Threshold 64 (packets)
      (pkts matched/bytes matched) 1413/66190
      (depth/total drops/no-buffer drops) 0/1052/0
```

```
Class-map: Classe_AF33 (match-any)
  670 packets, 30820 bytes
  5 minute offered rate 2000 bps, drop rate 0 bps
  Match: ip dscp af33 (30)
    670 packets, 30820 bytes
    5 minute rate 2000 bps
  Queueing
    Output Queue: Conversation 266
    Bandwidth 7 (%)
    Bandwidth 140 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 670/30820
    (depth/total drops/no-buffer drops) 0/347/0

Class-map: NetworkControl (match-any)
  9 packets, 504 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: ip dscp cs2 (16) cs6 (48) cs7 (56)
    9 packets, 504 bytes
    5 minute rate 0 bps
  Queueing
    Output Queue: Conversation 267
    Bandwidth 1 (%)
    Bandwidth 20 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

Class-map: BestEffort (match-any)
  14740 packets, 953946 bytes
  5 minute offered rate 31000 bps, drop rate 31000 bps
  Match: ip dscp default (0)
    14740 packets, 953946 bytes
    5 minute rate 31000 bps
  Queueing
    Output Queue: Conversation 268
    Bandwidth 6 (%)
    Bandwidth 120 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 4863/288624
    (depth/total drops/no-buffer drops) 0/4574/0
  police:
      cir 32000 bps, bc 1500 bytes
    conformed 4863 packets, 288624 bytes; actions:
      set-dscp-transmit default
    exceeded 9877 packets, 665322 bytes; actions:
      drop
    conformed 16000 bps, exceed 26000 bps

Class-map: class-default (match-any)
  29 packets, 1856 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

A.2 Simulação Computacional

Resultado: show policy-map Serial 0/1 – Simulação Computacional

```
Serial 1/0

 Service-policy output: SaidaWAN(fa0/1)

  Class-map: Voz (match-any)
    5590 packets, 308851 bytes
    5 minute offered rate 14000 bps, drop rate 0 bps
    Match: ip dscp ef (46)
      5590 packets, 308851 bytes
      5 minute rate 14000 bps
    Queueing
      Strict Priority
      Output Queue: Conversation 264
      Bandwidth 30 (%)
      Bandwidth 2560 (kbps) Burst 64000 (Bytes)
      (pkts matched/bytes matched) 5433/300234
      (total drops/bytes drops) 0/0

  Class-map: Classe_AF31 (match-any)
    5590 packets, 2940340 bytes
    5 minute offered rate 79000 bps, drop rate 0 bps
    Match: ip dscp af31 (26)
      5590 packets, 2940340 bytes
      5 minute rate 79000 bps
    Queueing
      Strict Priority
      Output Queue: Conversation 264
      Bandwidth 23 (%)
      Bandwidth 716 (kbps) Burst 17900 (Bytes)
      (pkts matched/bytes matched) 5581/2935606
      (total drops/bytes drops) 0/0

  Class-map: Classe_AF32 (match-any)
    2861 packets, 142537 bytes
    5 minute offered rate 10000 bps, drop rate 1000 bps
    Match: ip dscp af32 (28)
      2861 packets, 142537 bytes
      5 minute rate 10000 bps
    Queueing
      Output Queue: Conversation 265
      Bandwidth 7 (%)
      Bandwidth 1843 (kbps)Max Threshold 64 (packets)
      (pkts matched/bytes matched) 2817/140305
      (depth/total drops/no-buffer drops) 0/174/0
```

```
  Class-map: Classe_AF33 (match-any)
    2800 packets, 151200 bytes
    5 minute offered rate 10000 bps, drop rate 2000 bps
    Match: ip dscp af33 (30)
      2800 packets, 151200 bytes
      5 minute rate 10000 bps
    Queueing
      Output Queue: Conversation 266
      Bandwidth 7 (%)
      Bandwidth 1228 (kbps)Max Threshold 64 (packets)
      (pkts matched/bytes matched) 2751/148554
      (depth/total drops/no-buffer drops) 0/565/0

  Class-map: NetworkControl (match-any)
    7 packets, 462 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: ip dscp cs2 (16) cs6 (48) cs7 (56)
      7 packets, 462 bytes
      5 minute rate 0 bps
    Queueing
      Output Queue: Conversation 267
      Bandwidth 1 (%)
      Bandwidth 102 (kbps)Max Threshold 64 (packets)
      (pkts matched/bytes matched) 1/66
      (depth/total drops/no-buffer drops) 0/0/0

  Class-map: BestEffort (match-any)
    8340 packets, 2877022 bytes
    5 minute offered rate 80000 bps, drop rate 77000 bps
    Match: ip dscp default (0)
      8340 packets, 2877022 bytes
      5 minute rate 80000 bps
    Queueing
      Output Queue: Conversation 268
      Bandwidth 6 (%)
      Bandwidth 614 (kbps)Max Threshold 64 (packets)
      (pkts matched/bytes matched) 996/231100
      (depth/total drops/no-buffer drops) 0/367/0
    police:
        cir 32000 bps, bc 1500 bytes
      conformed 1004 packets, 231956 bytes; actions:
        set-dscp-transmit default
      exceeded 7336 packets, 2645066 bytes; actions:
        drop
      conformed 12000 bps, exceed 75000 bps

  Class-map: class-default (match-any)
    26 packets, 2992 bytes
    5 minute offered rate 0 bps, drop rate 0 bps
    Match: any
```

A.3 Simulação Real – Carga maior

Resultado: show policy-map interface Fa0/1 – Simulação Real – Carga maior

```
FastEthernet0/1

  Service-policy output: SaidaWAN(fa0/1)

    Class-map: Voz (match-any)
      122122 packets, 76435710 bytes
      5 minute offered rate 1370000 bps, drop rate 0 bps
      Match: ip dscp ef (46)
        122122 packets, 76435710 bytes
        5 minute rate 1370000 bps
      Queueing
        Strict Priority
        Output Queue: Conversation 264
        Bandwidth 25 (%)
        Bandwidth 25000 (kbps) Burst 625000 (Bytes)
        (pkts matched/bytes matched) 1/535
        (total drops/bytes drops) 0/0

    Class-map: Classe_AF31 (match-any)
      27000 packets, 39852000 bytes
      5 minute offered rate 933000 bps, drop rate 0 bps
      Match: ip dscp af31 (26)
        27000 packets, 39852000 bytes
        5 minute rate 933000 bps
      Queueing
        Strict Priority
        Output Queue: Conversation 264
        Bandwidth 7 (%)
        Bandwidth 7000 (kbps) Burst 175000 (Bytes)
        (pkts matched/bytes matched) 1/1476
        (total drops/bytes drops) 0/0

    Class-map: Classe_AF32 (match-any)
      4340 packets, 347550 bytes
      5 minute offered rate 10000 bps, drop rate 0 bps
      Match: ip dscp af32 (28)
        4340 packets, 347550 bytes
        5 minute rate 10000 bps
      Queueing
        Output Queue: Conversation 265
        Bandwidth 18 (%)
        Bandwidth 18000 (kbps)Max Threshold 64 (packets)
        (pkts matched/bytes matched) 1/65
        (depth/total drops/no-buffer drops) 0/0/0

    Class-map: Classe_AF33 (match-any)
      2700 packets, 529200 bytes
```

```
  5 minute offered rate 12000 bps, drop rate 0 bps
  Match: ip dscp af33 (30)
    2700 packets, 529200 bytes
    5 minute rate 12000 bps
  Queueing
    Output Queue: Conversation 266
    Bandwidth 12 (%)
    Bandwidth 12000 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

Class-map: NetworkControl (match-any)
  12 packets, 792 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: ip dscp cs2 (16) cs6 (48) cs7 (56)
    12 packets, 792 bytes
    5 minute rate 0 bps
  Queueing
    Output Queue: Conversation 267
    Bandwidth 1 (%)
    Bandwidth 1000 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0

Class-map: BestEffort (match-any)
  27000 packets, 17794890 bytes
  5 minute offered rate 435000 bps, drop rate 430000 bps
  Match: ip dscp default (0)
    27000 packets, 17794890 bytes
    5 minute rate 435000 bps
  Queueing
    Output Queue: Conversation 268
    Bandwidth 6 (%)
    Bandwidth 6000 (kbps)Max Threshold 64 (packets)
    (pkts matched/bytes matched) 0/0
    (depth/total drops/no-buffer drops) 0/0/0
  police:
      cir 32000 bps, bc 1500 bytes
    conformed 1214 packets, 371819 bytes; actions:
      set-dscp-transmit default
    exceeded 25786 packets, 17423071 bytes; actions:
      drop
    conformed 8000 bps, exceed 336000 bps

Class-map: class-default (match-any)
  38 packets, 4043 bytes
  5 minute offered rate 0 bps, drop rate 0 bps
  Match: any
```

APÊNDICE B – CONFIGURAÇÕES DO ROTEADOR "RouterA"

Resultado: Show runn config (RouterA)

```
Building configuration...

Current configuration : 3368 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname routerA
!
boot-start-marker
boot-end-marker
!
no aaa new-model
no ip icmp rate-limit unreachable
!
ip cef
no ip domain lookup
!
ip tcp synwait-time 5
!
class-map match-any BestEffort
 match ip dscp default
class-map match-any NetworkControl
 match ip dscp cs2  cs6  cs7
class-map match-any Voz
 match ip dscp ef
class-map match-any Marcacao_AF32
 match access-group name Trafego_AF32
class-map match-any Marcacao_AF33
 match access-group name Trafego_AF33
class-map match-any Marcacao_AF31
 match access-group name Trafego_AF31
class-map match-any Classe_AF31
 match ip dscp af31
class-map match-any MarcacaoVoz
 match access-group name TrafegoVoz
class-map match-any Classe_AF32
 match ip dscp af32
class-map match-any Classe_AF33
 match ip dscp af33
class-map match-any MarcacaoSinalizacao
 match access-group name TrafegoSinalizacao
!
policy-map EntradaLAN(entrada_de_trafego)
 class MarcacaoVoz
```

```
  set ip dscp ef
 class MarcacaoSinalizacao
  set ip dscp cs3
 class Marcacao_AF31
  set ip dscp af31
 class Marcacao_AF32
  set ip dscp af32
 class Marcacao_AF33
  set ip dscp af33
 class class-default
  set ip dscp default
policy-map SaidaWAN(fa0/1)
 class Voz
  priority percent 30
 class Classe_AF31
  priority percent 7
 class Classe_AF32
  bandwidth percent 18
 class Classe_AF33
  bandwidth percent 12
 class NetworkControl
  bandwidth percent 1
 class BestEffort
  bandwidth percent 6
  police cir 32000
    conform-action set-dscp-transmit default
    exceed-action drop
!
interface FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
 service-policy input EntradaLAN(entrada_de_trafego)
!
interface FastEthernet0/1
 bandwidth 10240
 no ip address
 shutdown
 duplex auto
 speed auto
 service-policy output SaidaWAN(fa0/1)
!
interface Serial1/0
 bandwidth 2000
 ip address 10.0.0.1 255.255.255.252
 serial restart-delay 0
 service-policy output SaidaWAN(fa0/1)
!
interface Serial1/1
 no ip address
```

```
 shutdown
 serial restart-delay 0
!
interface Serial1/2
 no ip address
 shutdown
 serial restart-delay 0
!
interface Serial1/3
 no ip address
 shutdown
 serial restart-delay 0
!
interface FastEthernet2/0
 no ip address
 shutdown
 duplex half
!
router rip
 network 10.0.0.0
 network 192.168.1.0
!
no ip http server
no ip http secure-server
!
ip access-list extended TrafegoSinalizacao
 permit ip any any dscp cs3
ip access-list extended TrafegoVoz
 deny   udp any any fragments
 deny   ip any any fragments
 permit tcp any any eq 2000
 permit tcp any eq 2000 any
 permit udp any any range 16384 32767
 permit ip any any dscp ef
ip access-list extended Trafego_AF31
 permit ip any any dscp af31
 permit tcp any any eq www
 permit tcp any eq www any
ip access-list extended Trafego_AF32
 permit ip any any dscp af32
 permit icmp any any
ip access-list extended Trafego_AF33
 permit tcp any any eq ftp-data
 permit tcp any any eq ftp
 permit tcp any any eq telnet
no cdp log mismatch duplex
!
control-plane
!
gatekeeper
```

```
 shutdown
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end
```