

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

AVALIAÇÃO DO SERVIÇO DE QOS NO FREEBSD

AUTOR: ALEXANDRE BUNN

FLORIANÓPOLIS, 2004

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

AVALIAÇÃO DO SERVIÇO DE QOS NO FREEBSD

Relatório do projeto apresentado como
requisito para conclusão do bacharelado
em Sistemas de informação da
Universidade Federal de Santa Catarina

AUTOR: ALEXANDRE BUNN

ORIENTADOR: DR. ROBERTO WILLRICH

BANCA EXAMINADORA: DR. MÁRIO ANTÔNIO RIBEIRO DANTAS

DR. VITÓRIO BRUNO MAZZOLA

FLORIANÓPOLIS, 2004

Agradecimentos

Aos meus pais e irmãos que apostaram em meu potencial durante toda a faculdade, sempre me dando apoio em todas as situações.

À todos os professores que participaram deste curso e que contribuíram para a nossa formação acadêmica e profissional.

Ao professor e orientador Roberto Willrich, pela atenção dispensada durante a realização deste trabalho e o empenho para que eu tivesse todos os recursos necessários para a realização da experimentação prática.

Aos nossos colegas de curso, por termos compartilhado angústias, aprendizados e vitórias durante todo o percurso.

RESUMO

A Internet, desde seu início, utiliza-se de um mecanismo de escalonamento de pacotes do tipo FIFO (*First In First Out*) também chamado de modelo Melhor Esforço. As soluções de telefonia e videoconferência usando as redes IP, como também as aplicações científicas de banda larga têm uma demanda cada vez maior. Como os fluxos de áudio e vídeo correspondem a serviços em tempo real, eles exigem prazos de transporte muito curtos e uma transferência que apresente poucas variações, isto é, regular. As aplicações tradicionais (web, correio eletrônico, transferência de arquivos) admitem intervalos de tempo maiores, mas por outro lado necessitam de uma baixa taxa de perda de pacotes.

O objetivo deste trabalho é avaliar, com base em algumas métricas de desempenho (*delay*, *jitter*, vazão e perda de pacotes), o serviço de QoS oferecido por Roteadores baseados em FreeBSD. Com base nos resultados obtidos pode-se concluir que o FreeBSD juntamente com suas ferramentas para suporte a QoS em redes atendem os requisitos das empresas em definir políticas para a priorização de pacotes a serem transmitidos pelo seu enlace de saída o qual geralmente tem uma largura de banda limitada se comparada a largura de banda da rede local.

Palavras-chave: Voz sobre IP, FreeBSD, QoS, AltQ, DUMMYNET.

SUMÁRIO

<i>CAPÍTULO 1 - INTRODUÇÃO</i>	1
1.1 Objetivos do Projeto	3
1.2 Motivação	3
1.3 Estrutura do Documento	4
<i>CAPÍTULO 2 - REDES TCP/IP</i>	5
2.1 Camada de Aplicação	6
2.2 Camada de Transporte	6
2.2.1 Serviços do TCP	6
2.2.2 Serviços do UDP	7
2.3 Camada de Rede	8
2.4 Camada de Enlace	8
2.5 Camada Física	9
<i>CAPÍTULO 3 - QUALIDADE DE SERVIÇO (QoS)</i>	10
3.1 Parâmetros de Desempenho da Rede	11
3.2 Arquiteturas de QoS	12
3.2.1 IntServ	12
3.2.2 DiffServ	13
3.2.3 MPLS	15
3.3 Mecanismos de controle e inibição de congestionamento	16
3.3.1 FIFO - <i>First in First Out</i>	16
3.3.2 Enfileiramento Fair Queuing (FQ)	17
3.3.3 Enfileiramento <i>Weighted Fair Queueing</i> (WFQ)	17
3.3.4 Enfileiramento <i>Priority Queueing</i> (PQ)	19
3.3.5 Enfileiramento <i>Class Based Queueing</i> (CBQ)	20
3.3.6 A detecção RED	23
3.3.7 A detecção WRED	23
<i>CAPÍTULO 4 - VOZ SOBRE IP (VOIP)</i>	25
4.1 Serviço Telefônico PSTN	25
4.2 Fundamentação da Voz sobre IP	26
4.2.1 Conversão AD e DA	27
4.2.2 Codificações e Decodificações da Voz	27
4.2.3 A Pilha IP: Protocolos TCP/IP	28
4.2.4 Protocolos de aplicação para VoIP	29
4.3 Telefonia IP	31
4.3.1 Servidores de telefonia IP e PBX	32
4.3.2 Gateways VoIP	33
4.3.3 Roteador VoIP	34
4.3.4 Endereçamento VoIP	35
4.3.5 Roteamento VoIP	35

4.3.6	Equipamentos do Usuário.....	36
4.4	Codificadores e Decodificadores de Voz.....	36
4.5	Qualidade de Serviço em VoIP.....	38
4.5.1	Mean Opinion Score.....	39
4.6	Codecs e a qualidade de voz.....	40
4.6.1	Escolha do codec apropriado.....	41
4.6.2	Qualidade de Voz Oferecida pela Rede.....	42
<i>CAPÍTULO 5 - FREEBSD.....</i>		44
5.1	História.....	44
5.2	Objetivos do Projeto FreeBSD.....	47
5.3	O Modelo de Desenvolvimento FreeBSD.....	47
5.4	A Versão Atual do FreeBSD.....	48
5.5	Características.....	49
5.6	QoS no FreeBSD.....	50
5.6.1	Dumynet.....	50
5.6.2	AltQ (Alternate Queueing).....	53
<i>CAPÍTULO 6 - AVALIAÇÃO DO SERVIÇO DE QOS NO FREEBSD.....</i>		55
6.1	Definição do Ambiente de Testes.....	55
6.2	Características do Equipamento Roteador.....	56
6.3	Métricas utilizadas na avaliação dos testes.....	57
6.4	Escolha de ferramentas para realizar os testes de desempenho.....	57
6.5	Definição do Tráfego de Teste.....	58
6.6	Diferenciação de Tráfego.....	59
6.7	Resultados dos testes experimentais de desempenho IP x QoS.....	60
6.8	Análise do Atraso fim-a-fim (<i>delay</i>).....	61
6.9	Análise da Variação de Atraso (<i>Jitter</i>).....	63
6.10	Análise da vazão.....	65
6.11	Análise da taxa de perdas.....	67
<i>CAPÍTULO 7 - CONCLUSÕES E TRABALHOS FUTUROS.....</i>		69
ANEXO 1 – Roteiro de Instalação e Configuração do AltQ.....		73
ANEXO 2 – Roteiro de Instalação e Configuração do RUDE & CRUDE.....		76
ANEXO 3 – Regras usadas para Simular o tráfego TCP constante.....		77
ANEXO 4 – Script para a decodificação do Tráfego recebido.....		78
ANEXO 5 – Sintaxe dos programas utilizados.....		80
ANEXO 6 – Artigo.....		81

LISTA DE FIGURAS

Figura 1 - A pilha de protocolos da Internet e as unidades de dados de protocolo [11]	5
Figura 2 - Operação da Fila FIFO [7].....	16
Figura 3 - Filas Fair Queueing [7]	17
Figura 4 - Operação do Algoritmo WFQ [7].....	18
Figura 5 - Filas WFQ [7]	18
Figura 6 - Operação do Enfileiramento Priority Queueing [7].....	19
Figura 7- Filas Priority Queueing [7]	20
Figura 8 - Operação do Enfileiramento Class Based Queueing [7]	21
Figura 9 - Filas Class Based Queueing [7]	21
Figura 10 - Métodos de Enfileiramento [7]	22
Figura 11 - Esquema para Seleção de Método de Enfileiramento [7].....	23
Figura 12 - O Funcionamento do WRED [7]	24
Figura 13 - Telefonia a pacotes de voz [5].....	26
Figura 14 - Elementos envolvidos na comunicação de voz [5].....	27
Figura 15 - Datagrama IP e seu cabeçalho [5].....	28
Figura 16 - Fases envolvidas na chamada VoIP [5]	29
Figura 17 - Pacote RTP [5].....	30
Figura 18- Elementos básicos da telefonia IP [16].....	31
Figura 19 - Uma rede VoIP com seus gateways conectados na PSTN [5].....	33
Figura 20 - Toll bypass usando gateways VoIP [5]	34
Figura 21 - Plataforma de Teste	56
Figura 22 - Delay rede não saturada.....	61
Figura 23 - Delay com a rede Saturada	62
Figura 24 - Jitter com a rede não saturada.....	63
Figura 25 - Jitter com a rede saturada	64
Figura 26 - Vazão com a rede não saturada.....	65
Figura 27 - Vazão com a rede saturada	66
Figura 28 - Perda de Pacotes	67

LISTA DE TABELAS

Tabela 1 - Codecs comuns em VoIP [17].....	37
Tabela 2 - Escala de pontuação de níveis de qualidade [5].....	40
Tabela 3 - Codecs e a Qualidade de Voz [5].....	41
Tabela 4 - Parâmetros de Referência de Rede [5].....	43
Tabela 5 - Estatísticas do Delay com a rede não saturada.....	62
Tabela 6 - Estatísticas do Delay com a rede Saturada.....	63
Tabela 7 - Estatísticas do Jitter com a rede não saturada.....	64
Tabela 8 - Estatísticas do Jitter com a rede Saturada.....	65
Tabela 9 - Estatísticas da Vazão com a rede não saturada.....	66
Tabela 10 - Estatísticas da Vazão com a rede saturada.....	67
Tabela 11 - Estatísticas da Perda com a Rede Saturada e sem QoS.....	68

CAPÍTULO 1 - INTRODUÇÃO

A utilização da rede, que inicialmente era utilizada apenas no âmbito local, foi-se tornando cada vez mais intensa devido à facilidade e a rapidez em que as informações poderiam ser trocadas e compartilhadas. Com isso, observou-se que as informações precisavam ultrapassar a barreira local e ter uma maior abrangência. A Internet é considerada um “Backbone” para possibilitar este tipo de comunicação. Graças a este recurso de troca de informações a grande distância, houve uma mudança geral em várias áreas, tais como: a área de pesquisa, que pode difundir e adquirir os seus conhecimentos de uma maneira mais rápida, mesmo fora do país; a do mercado mundial, cujas atividades e mercadorias foram apresentadas e negociadas de forma mais dinâmica e “transparentes”; a área de relações bancárias; e a própria área de informática, que utiliza este canal de divulgação de equipamentos, software, etc., e para tramitação de documentos, dados e mensagens sejam elas via e-mail ou comunicação on-line (via texto, voz ou vídeo).

O uso das aplicações em rede tem sido imprescindível na realização de tarefas em vários ambientes de trabalho. Atualmente utilizamos a internet para realizar inúmeras atividades, tais como: atividades bancárias, compras, pesquisas, troca de informações (confidenciais ou não), mensagens de e-mail, reuniões através de videoconferência e chat, dentre outras. Outro papel que a Internet vem ganhando é no campo do ensino onde pessoas podem se aperfeiçoar através de aplicações de *E-learning* e portais de informações onde profissionais renomados disponibilizam artigos das mais diversas áreas e acessíveis de qualquer lugar com acesso a Internet.

A rede Internet oferece basicamente um único tipo de serviço de comunicação dito do tipo "melhor esforço" (Best-effort Service), próprio do protocolo IP, onde não é provido nenhum mecanismo que garanta certos níveis de qualidade de serviços (QoS) e portanto não há limites em termos de parâmetros de desempenho de redes, do tipo taxa de bits, atraso fim-a-fim, variação de atraso e taxa de perda de pacotes. QoS pode ser definida como a especificação qualitativa e quantitativa dos requisitos de uma aplicação que um sistema deveria satisfazer a fim de obter a qualidade desejada [1].

As soluções de telefonia e videoconferência usando as redes IP, como também as aplicações científicas de banda larga têm uma demanda cada vez maior. Como os fluxos de áudio e vídeo correspondem a serviços em tempo real, eles exigem prazos de transporte muito curtos e uma transferência que apresente poucas variações, isto é, regular. As aplicações

tradicionais (web, correio eletrônico, transferência de arquivos) admitem intervalos de tempo maiores, mas por outro lado necessitam de uma baixa taxa de perda de pacotes.

Para enfrentar as novas restrições e fornecer uma qualidade de serviço adequada, os engenheiros podem utilizar o superdimensionamento da rede. Porém, esta é uma solução em curto prazo, pois o tráfego continua aumentando e tende a ocupar rapidamente toda a largura de banda disponível, além do que a aquisição de links de comunicação com alta largura de banda é bastante caro.

Mesmo nos links pouco carregados, os picos de tráfego muito curtos ocasionam às vezes degradações de desempenho inaceitáveis para as aplicações em tempo real, como aplicações de Telefonia IP, Vídeo sob demanda, Videoconferência, dentre outras.

Enfim, a expansão e a interconexão das redes tornam cada vez mais complexo o gerenciamento das tabelas de roteamento e torna cada vez mais necessário se ter mecanismos para priorizar o tráfego. Esta priorização permite dar preferência para tráfegos que necessitam de rigorosos parâmetros de qualidade, como nas aplicações de tempo-real, postergando o tráfego de pacotes que toleram maiores tempos de atraso, por exemplo.

Com a necessidade de suprir essas deficiências, novos protocolos e padrões estão sendo adotados para compor as chamadas Arquiteturas de Qualidade de Serviços. Dentre essas arquiteturas destacam-se, por sua grande aceitação, a de Serviços Integrados (IntServ) [18] e a de Serviços Diferenciados (Diffserv) [19]. Ambas as arquiteturas contam com grupos de trabalho no IETF (*Internet Engineering Task Force*).

Para que a rede suporte gerenciamento de QoS, os roteadores devem suportar os serviços necessários, e são estes, os responsáveis pelo encaminhamento de pacotes no nível de rede. Atualmente, a implementação das arquiteturas DiffServ e IntServ se torna demasiadamente caras, pelo fato de que o controle da maior parte dos roteadores intermediários está nas mãos das empresas de telecomunicações, as quais cobram por cada serviço requisitado para que se possa ter alguma garantia de qualidade sendo esta diretamente proporcional ao preço do SLA (*Service Level Agreement*) efetuado.

Diante desta situação, pequenas e médias empresas optam por controlar o tráfego apenas em seus próprios roteadores, dando preferência a tráfegos considerados críticos e deixando em segundo plano os demais tráfegos. Esta estratégia é adotada de forma a dar uma maior dinâmica ao fluxo de dados e otimizando os recursos de rede da empresa de forma a reduzir os custos com a aquisição de um link de dados com maior largura de banda, por exemplo. Neste contexto, é que surgem as ferramentas para diferenciação de tráfego, incluindo a classificação de pacotes, políticas de escalonamento de filas de encaminhamento nos roteadores e de descarte de pacotes quando estas filas estiverem cheias.

1.1 Objetivos do Projeto

O objetivo deste projeto é avaliar o uso de roteadores baseados em FreeBSD para realização de diferenciação de tráfego. Roteadores FreeBSD são computadores com S.O e que realizam as tarefas de roteamento, e no nosso caso, incluindo os serviços de diferenciação de tráfego.

Como objetivos específicos temos:

- Estudo de Qualidade de Serviço e das arquiteturas DiffServ e IntServ;
- Estudo dos serviços de roteamento implementados no S.O. FreeBSD;
- Estudo sobre VoIP e os requisitos de rede para cada um dos CODECs de voz disponíveis;
- Definição e implantação de um TestBed para realizar avaliações de desempenho com o roteador FreeBSD realizando serviços de diferenciação de tráfego;
- Realização de medidas de desempenho de redes usando um roteador FreeBSD. Estas medidas serão realizadas em diferentes cenários de configuração do roteador visando prover métricas para analisar algumas políticas de escalonamento de filas disponibilizadas no FreeBSD.

1.2 Motivação

A grande motivação para a realização deste trabalho é a situação atual das empresas, as quais estão explorando cada vez mais os recursos que a rede oferece, em termos conectividade (aplicações para comunicação tradicionais, como e-mail e programas de mensagens instantâneas, e de tempo real, como telefonia IP e Videoconferência), recursos estes que para ter um bom funcionamento precisam de algumas garantias de QoS.

Diante desta necessidade é que se pensou em efetuar estes estudos de forma a ajudar as empresas na utilização de seus recursos de rede, com a redução de custos com a aquisição de novos links de dados e principalmente com a aquisição de sistemas que efetuem o gerenciamento do tráfego, que no caso da utilização do Sistema Operacional FreeBSD e suas Ferramentas o custo de aquisição não existe em termos de licenças para sua utilização.

1.3 Estrutura do Documento

Este relatório está organizado na forma que segue:

O Capítulo 1 que trata da parte introdutória, mostrará a importância da internet como um canal de comunicação, os objetivos deste trabalho e as motivações.

O Capítulo 2 apresenta os principais conceitos das redes TCP/IP, ou seja, da pilha de protocolos utilizada na Internet atual, mostrando suas camadas e suas respectivas funções.

O Capítulo 3 apresenta os principais conceitos relacionados à Qualidade de Serviço (QoS) em redes e as principais arquiteturas definidas para o suporte a este serviço e os principais algoritmos que possibilitam o controle e condicionamento de tráfego de forma a tornar possível o controle da Qualidade para as aplicações que se utilizam da rede como meio de propagação de informações.

O Capítulo 4 apresenta os principais conceitos relacionados a VoIP e os mecanismos utilizados para a medição da Qualidade de Serviço, tendo como base a qualidade percebida pelo usuário.

O Capítulo 5 apresenta um panorama do sistema Operacional FreeBSD explorando sua história, objetivos, modelo de desenvolvimento e algumas de suas funcionalidades.

O Capítulo 6 apresenta as principais aplicações que fazem com que o FreeBSD, na função de roteador, implemente mecanismos para suporte a QoS em redes.

O Capítulo 7 apresenta a avaliação dos serviços de QoS no FreeBSD, mostrando os resultados da experimentação prática proposta pelo presente trabalho.

E finalmente o Capítulo 8 apresenta além das conclusões referentes ao trabalho em questão, sugestões para trabalhos futuros relacionados ao tema abordado pelo presente trabalho.

CAPÍTULO 2 - REDES TCP/IP

A pilha de protocolos da Internet consiste em cinco camadas: camada física, camada de enlace, camada de rede, camada de transporte e camada de aplicação. Em vez de usarmos a inconveniente terminologia n -PDU para cada uma das cinco camadas, vamos dar nomes especiais para quatro das cinco PDUs: quadro, datagrama, segmento e mensagem. Não vamos nomear a unidade de dados da camada física porque geralmente não se usa nenhum nome para essa camada. A pilha da Internet e os nomes das PDUs correspondentes estão ilustrados na Figura 1 [11].

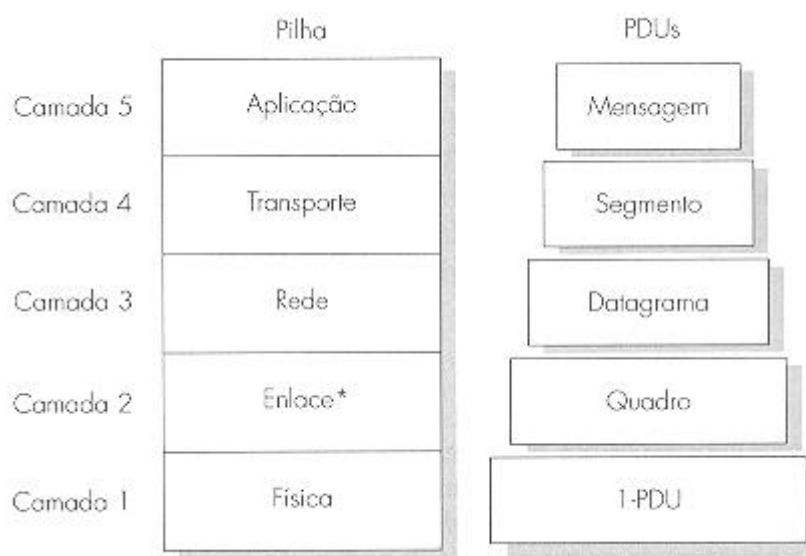


Figura 1 - A pilha de protocolos da Internet e as unidades de dados de protocolo [11]

A pilha da Internet, também conhecida como Arquitetura Internet ou ainda comumente chamada de Arquitetura TCP/IP (nome este devido ao nome de seus dois principais protocolos) permite que uma camada de protocolo seja implementada em software, hardware ou usando-se uma combinação dos dois. Os protocolos da camada de aplicação – como o http e SMTP – são quase sempre implementados nos softwares dos sistemas finais; isso também vale para os protocolos de camadas de transporte. Como a camada física e a camada de enlace são responsáveis pelo manuseio da comunicação por um enlace específico, elas são implementadas em uma placa de interface com a rede (por exemplo, as placas de interface da Ethernet ou das redes ATM) associada a um dado enlace. A camada de rede é,

freqüentemente, uma implementação mista de hardware e software. Apresentamos a seguir as camadas da Internet e os serviços que elas oferecem.

2.1 Camada de Aplicação

A camada de aplicação é responsável pelo suporte das aplicações de rede. Ela inclui muitos protocolos, dentre eles o HTTP, para suportar a Web, o SMTP, para suportar o correio eletrônico, o FTP, para suportar o transporte de arquivos e o RTP para suportar aplicações de tempo real como vídeo sob demanda, rádio por Internet, dentre outras.

2.2 Camada de Transporte

A camada de transporte fornece o serviço de transporte das mensagens da camada de aplicação entre os lados do cliente e do servidor de uma aplicação. Na Internet há dois protocolos de transporte: O TCP e o UDP.

2.2.1 Serviços do TCP

O modelo de serviço TCP inclui um serviço orientado à conexão e um serviço confiável de transferência de dados. Quando uma aplicação solicita o TCP como seu protocolo de transporte, ela recebe dele ambos os serviços.

Serviço orientado à conexão: o TCP faz com que o cliente e o servidor troquem informações de controle da camada de transporte antes que as mensagens da camada de aplicação comecem a fluir. Esse procedimento de apresentação, por assim dizer, alerta o cliente e o servidor, permitindo que eles se preparem para uma enxurrada de pacotes. Após a fase de apresentação, dizemos que existe uma conexão TCP entre a porta dos dois processos. A conexão é sempre full-duplex, na qual os dois processos podem enviar mensagens um ao outro pela conexão, ao mesmo tempo. Quando a aplicação termina de enviar as mensagens, ela deve interromper a conexão. Esse serviço é chamado de ‘orientado a conexão’, e não de serviço ‘de conexão’ (ou um serviço de ‘circuito virtual’), porque os dois processos estão conectados de um modo muito solto.

Serviço de transporte confiável: os processos comunicantes podem confiar no TCP para a entrega dos dados enviados sem erro e na ordem correta. Quando um lado da aplicação passa uma cadeia de bytes para dentro de uma porta, ele pode contar com o TCP para entregar a mesma cadeia de dados à porta receptora, sem falta de bytes ou bytes duplicados.

O TCP também inclui um mecanismo de controle de congestionamento, que é um serviço mais voltado ao bem estar geral da Internet do que ao benefício direto dos processos comunicantes. O mecanismo de controle de congestionamento do TCP limita a capacidade de transmissão de um processo (cliente ou servidor) quando a rede está congestionada. Em particular o controle de congestionamento do TCP tenta limitar cada conexão do TCP à justa porção de largura de banda da rede.

A limitação da velocidade de transmissão pode ter um efeito muito prejudicial sobre as aplicações de áudio e vídeo em tempo real que tenham uma exigência de largura de banda mínima. Além disso, as aplicações de tempo real são tolerantes às perdas e não necessitam de um serviço de transporte totalmente confiável. Por essas razões, os desenvolvedores de aplicações em tempo real rodam, em geral, suas aplicações em UDP, e não em TCP.

2.2.2 Serviços do UDP

O UDP é um protocolo de transporte simplificado, leve, com um modelo de serviço minimalista. Ele é um serviço não orientado à conexão; portanto, não há apresentação antes que os dois processos comecem a se comunicar. Ele fornece um serviço de transferência de dados não confiável, isto é, quando um processo envia uma mensagem para dentro de uma porta do UDP, este não oferece nenhuma garantia de que a mensagem alcançará a porta receptora. Além disso, as mensagens que, de fato, chegam à porta receptora podem chegar fora de ordem.

Por outro lado, o UDP não inclui um mecanismo de controle de congestionamento; portanto, um processo originador pode bombear dados para dentro de uma porta do UDP na taxa que quiser. Embora nem todos os dados consigam alcançar a porta receptora, uma grande fração deles consegue chegar. Os desenvolvedores de aplicações em tempo real freqüentemente preferem rodar suas aplicações em UDP. Da mesma maneira que o TCP, o UDP não oferece garantias em relação aos atrasos.

O UDP (*User Datagram Protocol*) opera no modo sem conexão e fornece um serviço datagrama não-confiável, sendo uma simples extensão do protocolo IP (*Internet*

Protocol) que é responsável pelo serviço da camada inter-rede na Arquitetura Internet TCP/IP.

O UDP recebe os pedidos de transmissão de mensagens entregues pelos processos de aplicação da estação de origem, e os encaminha ao IP que é responsável pela transmissão. Na estação de destino, o processo inverso ocorre. O IP entrega as mensagens (datagramas) recebidas ao UDP que as entrega aos processos de aplicação. A principal função do UDP é multiplexar (técnica utilizada para se compartilhar um meio de transmissão), na origem, e demultiplexar, no destino, o acesso ao nível inter-rede. [12]

2.3 Camada de Rede

A camada de rede é responsável pelo roteamento dos datagramas de uma máquina para outra. A camada de rede da Internet tem dois componentes principais. Ela possui um protocolo que determina os campos no datagrama IP, bem como o modo como agem os sistemas finais e os roteadores nesses campos. A camada de rede da Internet também contém protocolos de roteamento que determinam as rotas que os datagramas seguem entre origens e destinos. A Internet tem muitos protocolos de roteamento.

2.4 Camada de Enlace

A camada de rede roteia um pacote por meio de uma série de comutadores de pacote (chamados de roteadores na Internet) entre a origem e o destino. Para levar um pacote de um nó (sistema final ou comutador de pacote) ao nó seguinte na rota, a camada de rede depende dos serviços da camada de enlace. Na verdade, a cada nó, o IP passa o datagrama para a camada de enlace, que o entrega no nó seguinte ao longo da rota. Nesse nó seguinte, a camada de enlace passa o datagrama IP à camada de rede.

2.5 Camada Física

Enquanto a tarefa da camada de enlace é movimentar quadros inteiros de um elemento da rede até o elemento adjacente, a tarefa da camada física é movimentar os *bits individuais* interiores ao quadro de um nó para o nó seguinte. Os protocolos dessa camada novamente dependem do enlace; além disso, dependem do próprio meio de transmissão do enlace (por par de fios de cobre trançados ou fibra ótica monomodal). Por exemplo, a Ethernet tem muitos protocolos de camada física: um par de fios de cobre trançado, outro para cabo coaxial, outro para fibra e assim por diante. Em cada caso o bit é movimentado pelo enlace de maneira diferente.

CAPÍTULO 3 - QUALIDADE DE SERVIÇO (QOS)

Nos últimos anos, houve extraordinário desenvolvimento e ampla disseminação das aplicações em rede que transmitem e recebem conteúdo de áudio e vídeo pela Internet. Novas aplicações de rede multimídia (também conhecidas como aplicações de mídia de taxa constante) – vídeo de entretenimento, telefonia IP, rádio por Internet, sites WWW multimídia, teleconferência, jogos interativos, aprendizado à distância e muitas outras aplicações – são anunciadas continuamente. As exigências de serviço dessas aplicações diferem de modo significativo daquelas aplicações tradicionais orientadas a dados, como texto/imagem como: Web, e-mail, FTP e DNS [11].

Há algum tempo o modelo de controle de tráfego "melhor esforço" não atende mais aos requisitos de todas as aplicações que utilizam o protocolo IP. É necessário fornecer mecanismos à rede para que esta possa atender as novas necessidades das aplicações, oferecendo algum mecanismo que controle a qualidade nos serviços oferecidos.

Qualidade de Serviço é algo difícil de se definir, tendo significados diferentes para pessoas distintas. Há pouco tempo o termo QoS era utilizado apenas em tecnologias de rede como o ATM e Frame Relay, tecnologias estas que tem já na concepção de seus serviços oferecidos mecanismos para o controle da qualidade.

Em um sistema multimídia distribuído a qualidade de serviço pode ser definida como "a especificação qualitativa e quantitativa dos requisitos de uma aplicação que um sistema multimídia deveria satisfazer a fim de obter a qualidade desejada" [1]. Em redes de computadores o termo QoS é utilizado para definir o desempenho da rede, no que diz respeito ao bom funcionamento das aplicações.

Em termos gerais, QoS define uma política para dividir mais racionalmente a banda disponível e tentar garantir para certos serviços a latência e largura de banda, necessária especialmente em momentos de sobrecarga da rede, pois aqueles serviços definidos como prioritários deverão continuar funcionando perfeitamente, mesmo que isto prejudique a performance de outros de menor importância [2].

A implantação de qualidade de serviço (QoS) na rede é essencial para o sucesso de aplicações avançadas, como telemedicina, videoconferência e VoIP (voz sobre IP). Estas aplicações demandam, além de grande largura de banda, um serviço diferenciado. Em muitos casos, é preciso garantir que a transmissão de dados seja feita sem interrupção ou perda de pacotes.

QoS é fundamental para diversos tipos de aplicações, sobretudo as multimídia, pois é desejável que haja um sincronismo entre as diversas mídias. Por exemplo, numa videoconferência o som deve estar sincronizado com a imagem, ou seja, deve haver sincronia entre as palavras e os movimentos dos lábios das pessoas.

Outras aplicações podem negociar com a rede a probabilidade de queda da conexão, não se preocupando com a velocidade de transmissão nem o sincronismo, mas desejando-se que a conexão tenha um certo grau de confiabilidade.

São inúmeras as definições para qualidade de serviço encontradas nas literaturas, dentre elas pode-se destacar:

- "Qualidade de Serviço é definida como a capacidade da rede prover serviço de encaminhamento de dados de forma consistente e previsível" [3].
- "Habilidade de um elemento de rede, seja uma aplicação, host, roteador ou outro dispositivo, ter algum nível de garantia que seu tráfego e exigências de serviço podem ser satisfeitas" [4].

3.1 Parâmetros de Desempenho da Rede

Existem vários parâmetros que ajudam a inferir o desempenho de uma rede e conseqüentemente da qualidade de serviço oferecida. Dentre os parâmetros podemos destacar:

- **Taxa de bits:** a taxa de bits entre dois sistemas comunicantes é o número de dígitos binários que a rede é capaz de transmitir por unidade de tempo (expresso em Kbps, Mbps, Gbps, etc.) [5]. Neste contexto quando se refere ao termo taxa de bits pode ser a respeito da taxa de bits da rede interna, que atualmente com a tecnologia Ethernet tem-se taxas da ordem de 1Gbps, e a taxa de bits entre a rede interna e a internet ou com uma filial conectada a matriz através de um link Frame Relay, por exemplo, onde se concentram os esforços em otimizar o uso da rede, pois este ponto (Rede Interna - Internet) é geralmente considerado o gargalo de toda rede e é onde também os custos para elevação desta taxa se tornam muito altos.
- **Vazão (Throughput):** A vazão de uma rede é sua taxa de bits efetiva, ou a largura de banda efetiva. Assim, nós definimos a vazão como sendo a diferença entre a taxa de bits da ligação e os vários *overheads* (sobrecarga originada dos protocolos das demais camadas) associados à tecnologia de transmissão empregada.
- **Taxa de Erros:** É parâmetro importante para redes multimídia e que pode ser definido de várias maneiras. Uma taxa de erro de bits (BER – *Bit Error Rate*), que á

a razão entre o número médio de bits corrompidos ou errados e o número total de bits que são transmitidos. Outro é a taxa de erro de pacote (PER) definido como o anterior, substituindo bits por pacotes.

- **Atraso:** Um dos principais parâmetros de desempenho de rede é o atraso. Ele pode ser definido de várias maneiras. Nós consideramos inicialmente o atraso fim-a-fim, que significa o tempo levado para transmitir um bloco de dados de um emissor a um receptor. Dentre os componentes que fazem parte do atraso fim-a-fim estão, o atraso na interface (tempo esperado de um pacote pronto para ser transmitido e a efetiva transmissão na rede), atraso de trânsito (tempo de propagação de um pacote de uma ponta a outra, no meio de transmissão, este parâmetro depende apenas da distância entre o ponto transmissor e o receptor) e atraso de transmissão (tempo para a transmissão de um bloco de dados fim-a-fim).
- **Variação de Atraso (Jitter):** Pode ser definido como a variação do atraso de transmissão entre os vários pacotes envolvidos em uma determinada comunicação. Esta variação quando alta prejudica aplicações como Vídeo Digital onde, os fluxos de vídeo e de áudio são enviados separadamente, ou seja se a Variação de Atraso é muito alta fica difícil manter a sincronização intermídia da aplicação em questão.

3.2 Arquiteturas de QoS

O IETF com o intuito de propor soluções para QoS e com isso garantir a interoperabilidade entre as soluções de diferentes fornecedores introduziu duas arquiteturas, tendo em vista a qualidade de serviço, o modelo *Integrated Services* (IntServ) e o *Differentiated Service* (Diffserv) que serão estudados abaixo.

3.2.1 IntServ

Na arquitetura Intserv [19], a aplicação reserva recursos (largura de banda, tempo de atraso e *Jitter*, por exemplo) que vai utilizar na rede, antes mesmo de iniciar o envio dos dados de áudio, vídeo, etc.. Esta reserva é feita através do protocolo de sinalização RSVP (*Resource Reservation Protocol*) [19], que poderá ser rejeitada, caso a rede não possa atender a solicitações da aplicação. Na sinalização RSVP existe troca de mensagens de controle entre emissor e receptor de forma que num determinado período de tempo possamos alocar uma

faixa da largura de banda para a transmissão dos dados. O RSVP utiliza o algoritmo *token bucket* para a suavização de tráfego, permitindo picos de tráfego por períodos curtos. O RSVP provê granularidade e controle fino das solicitações das aplicações. Sua desvantagem é a complexidade das suas operações nos roteadores, gerando *Overhead*. Com a reserva, os pacotes são identificados através de um rótulo. Faz-se uma solicitação sobre a QoS necessária e, de acordo com a reserva correspondente, recebe-se a QoS desejada. O Intserv aplica-se mais a redes corporativas, proprietárias e estruturadas. Por causa disso, na Internet, por ser uma rede muito extensa, ele não funciona corretamente. Além disso, os roteadores que fazem reserva de recurso terão que ser compatíveis.

Neste modelo temos alocação para dois tipos de serviço, além do "melhor esforço":

- Serviços Garantidos – para aplicações que necessitam de um atraso constante;
- Serviços de Carga Controlada – aplicações que necessitam de segurança e um limite de variação de atraso (*Jitter*), eliminando a idéia do “melhor esforço”.

Aplicações que exigem este tipo de serviço devem configurar caminhos e reservar recursos antes da transmissão dos dados. A implementação do Intserv é feita por quatro componentes:

- Protocolo de sinalização (RSVP);
- Rotina de controle de admissão;
- Classificador;
- Escalonador de pacotes.

Esses componentes tem por função organizar os pacotes de forma que a Qualidade de Serviço seja aplicada.

3.2.2 DiffServ

Na arquitetura do Diffserv [18], que foi proposta em 1998, não há reserva de recursos e ela é definida, principalmente, pelas RFCs 2474, 2475, dentre outras, sendo a qualidade garantida através de mecanismos de priorização de pacotes na rede, otimizando e minimizando o processamento necessário nos roteadores para os agregados que fazem os serviços. Os pacotes são classificados em classes de processamento QoS. Esse processamento é feito pelos roteadores de entrada (*edge routers*), reduzindo o nível de processamento

necessários nos outros roteadores (*core routers*) para fluxo de dados. Os roteadores *core* apenas processam, seguindo a classe de serviço.

São funcionalidades do Diffserv a classificação de pacotes, o monitoramento do tráfego, a marcação de pacotes, a suavização de tráfego, a classificação de comportamento e o escalonamento de pacotes. No Diffserv, há um campo específico para indicar o rótulo DSCP – *Differentiated Service Code Point* – que indica o agregado ou classe à qual o pacote pertence, que é o ToS (*Type of Service*), no IPv4, e o *Traffic Class*, no IPv6.

Os pacotes são processados e comutados segundo a marcação que receberam (DSCP), ou seja, recebem o serviço segundo o rótulo. Os serviços podem ser:

- EF – *Expedited Forwarding* – ou *Premium service*, fornece garantias mais rígidas de qualidade de serviço, para aplicações muito sensíveis a variações de características temporais da rede. Ele pode ser utilizado para implementar um serviço com pouco atraso, pouca variação do atraso e taxa de bits garantida. Neste tipo de serviço os pacotes que não estiverem dentro do perfil contratado são descartados diretamente não passando por uma reclassificação. Este serviço é mais indicado para aplicações multimídia e de tempo real.
- AF – *Assured Forwarding* – fornece quatro classes com diferentes níveis de garantias de encaminhamento de pacotes IP. Estas classes são referenciadas como AF_nm, onde "n" é o número da classe (1 a 4) e "m" é o valor de precedência de descarte (1 a 3). Assim, o AF define um grupo de PHBs (*Per-Hop Behaviors*) onde são definidas até 12 níveis de serviços AF distintos. Neste projeto chamaremos essas classes de AF1, AF2, AF3 e AF4. No caso de congestionamento dentro de uma classe AF, a precedência de descarte de um pacote determina a importância relativa daquele pacote. Um nó DS em situação de congestionamento, preferencialmente descarta pacotes com um maior valor de precedência de descarte, ao mesmo tempo em que evita descartar pacotes com um valor de precedência de descarte menor. a depender do grau de prioridade a ser empregado. Este serviço utiliza o algoritmo token bucket para suavização do tráfego e é indicado para aplicações que necessitam de uma melhor confiabilidade que a oferecida pelo modelo "melhor esforço".

3.2.3 MPLS

O MPLS (Multi Protocol Label Switching) [6] traz ao ambiente IP sem conexão um determinado controle que era um privilégio exclusivo das tecnologias baseadas em conexões como, por exemplo, frame relay e ATM. Baseada na comutação de pacotes de dados em função de uma etiqueta, ou "label", acrescentada a seu cabeçalho, esta tecnologia é flexível e versátil, pois um label pode definir o caminho, a fonte, o destino, a aplicação, a qualidade de serviço, etc.

Antes de tudo, o MPLS facilita o encaminhamento de pacotes através de rotas pré-configuradas, em função de critérios como, por exemplo, uma baixa taxa de engarrafamento, a divisão da carga em várias rotas ou a necessidade de restaurar um link em menos de 60 ms em caso de problema com o circuito, etc. Os sistemas intermediários situados no centro da rede processam as informações primárias contidas nos labels muito mais rapidamente, pois a decisão de roteamento é tomada antecipadamente. Por isso, os pacotes circulam mais rapidamente, e os recursos dos roteadores e comutadores são menos solicitados.

Um "label" MPLS pode ser associado ao fluxo de uma aplicação específica, o que permite distingui-lo dos outros, ao contrário do protocolo IP que, não diferencia entre aplicações. As aplicações que exigem uma largura de banda garantida e estável, como vimos acima, podem então ser processadas de modo prioritário.

Como um label MPLS pode ser associado a uma fonte e/ou destino, isso facilita a criação de circuitos virtuais privados compartilhando uma infra-estrutura física comum. Como veremos abaixo, estas VPNs permitem agregar os tráfegos que apresentam características comuns, o que é vantajoso tanto em termos de recursos da rede quanto em termos de segurança e gerenciamento da faturação. Por outro lado, a hierarquia dos labels MPLS permite construir VPNs que não necessitam de nenhuma modificação no plano de endereçamento IP dos clientes, coexistindo com as redes MPLS que alguns deles poderiam estabelecer entre seus diferentes sites.

Um dos principais conceitos do MPLS é a classe equivalente de encaminhamento FEC (*Forwarding Equivalence Class*) que define o caminho dos pacotes através de rótulos. Para uma FEC particular, a concatenação de caminhos comutados, forma um caminho unidirecional através da rede, conhecido como caminho comutado por rótulo LSP (*Label Switched Path*), ou seja, pacotes pertencentes a uma FEC comum, sempre pegam o mesmo caminho através do domínio MPLS. O LSP consiste de saltos comutados por rótulo "*Label Switched Hops*" entre pares de LSRs.

A arquitetura MPLS emprega dois tipos principais de roteadores, LSR (*Label Switching Router*) [8] que possuem somente interfaces puramente MPLS, possibilitando enviar o tráfego baseado puramente em rótulos, ou seja, trabalha somente com a camada 2. Para tratar os pacotes vindo das redes IP existe o LER (*Label Edge Routers*), que pode ser de ingresso ou egresso na rede MPLS, dependendo se o pacote está entrando ou saindo do domínio MPLS. Cada LSR e LER mantêm uma base de informação de rótulos conhecida como LIB (*Label Information Base*). A mesma é usada para dar encaminhamento aos pacotes.

3.3 Mecanismos de controle e inibição de congestionamento

Há vários mecanismos de enfileiramento para controle e prevenção de congestionamento em interfaces de roteadores (Ethernet, Seriais, Frame Relay, etc.), aplicáveis tanto em redes WAN como em LAN. Esta seção apresenta os principais mecanismos de enfileiramento e políticas de descarte.

3.3.1 FIFO - *First in First Out*

Em geral, o controle de tráfego nas conexões seriais dos roteadores é implementado através de filas FIFO (o primeiro a entrar é o primeiro a sair), conforme podemos observar na Figura 2. Uma fila FIFO é um mecanismo de armazenamento e repasse (*store and forward*) que não implementa nenhum tipo de classificação.

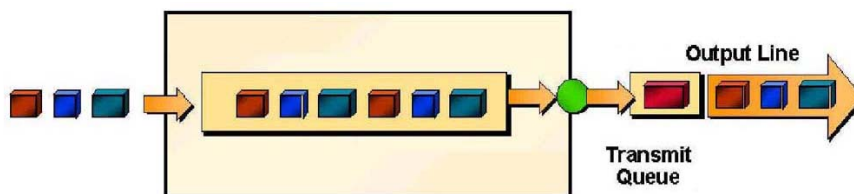


Figura 2 - Operação da Fila FIFO [7]

A ordem de chegada dos pacotes é que determina a alocação da banda, e o que chega primeiro é logo atendido. É o tratamento *default* da fila nos roteadores, já que não requer nenhuma configuração. O problema ocorre em tráfego de rajada, que pode causar

longos atrasos em aplicações sensíveis ao tempo. Por isso, filas FIFO não servem para aplicações que requerem QoS.

3.3.2 Enfileiramento Fair Queuing (FQ)

No algoritmo de Enfileiramento *Fair Queueing* (enfileiramento justo), as mensagens são ordenadas em sessões, e, para cada sessão, é alocado um canal (Figura 3). Essa operação provê uma alocação mais justa da banda entre os fluxos de dados e evita que fluxos mal comportados degrade o desempenho de outros fluxos. O escalonador visita as filas não vazias de modo circular, servindo um pacote de cada fila por vez. O objetivo da FQ é alocar a mesma quantidade de largura de banda para todos os fluxos: ela não foi projetada para dar suporte a fluxos com necessidades diferentes de largura de banda.

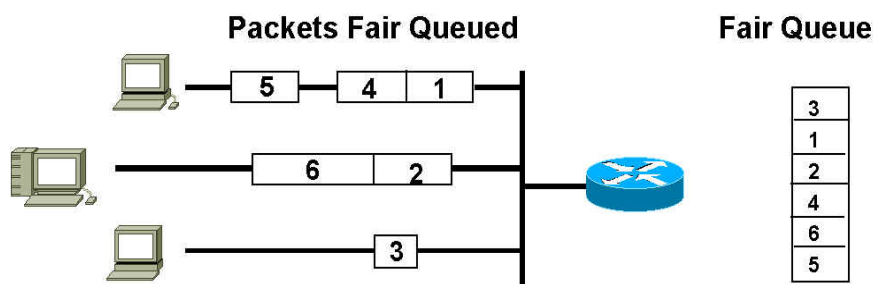


Figura 3 - Filas Fair Queueing [7]

3.3.3 Enfileiramento Weighted Fair Queueing (WFQ)

É uma implementação do FQ na qual é possível ponderar determinados tipos de fluxo. O algoritmo escalona o tráfego prioritário (interativo) para frente da fila, reduzindo o tempo de resposta. Ao mesmo tempo, compartilha o restante da banda com os outros tipos de fluxo de uma forma justa. O WFQ é dinâmico e se adapta automaticamente às mudanças das condições de tráfego, sendo bastante útil em conexões seriais de baixa velocidade até 2 Mbps. A implementação da WFQ exige uma alta carga computacional o que causa impacto na sua escalabilidade quando for oferecido suporte a um número grande de classes de serviço. Abaixo (Figura 4) seguem um modelo ilustrativo do funcionamento do WFQ.

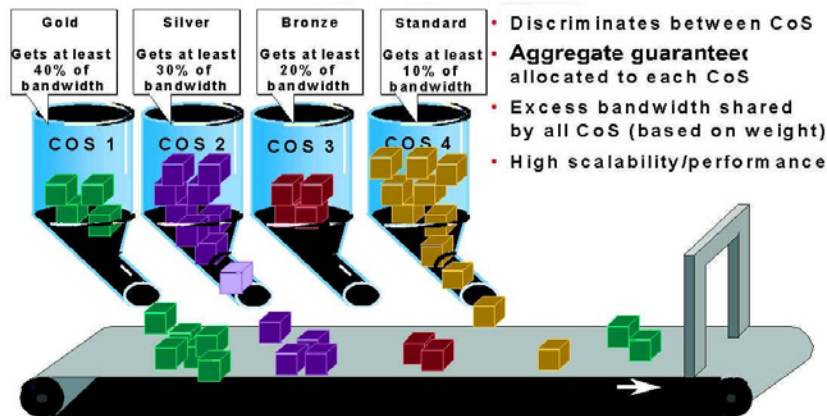


Figura 4 - Operação do Algoritmo WFQ [7]

Por apresentar um desempenho superior à fila FIFO, a fila WFQ já vem pré-configurada nas interfaces seriais de alguns roteadores disponíveis no mercado como por exemplo os roteadores Cisco.

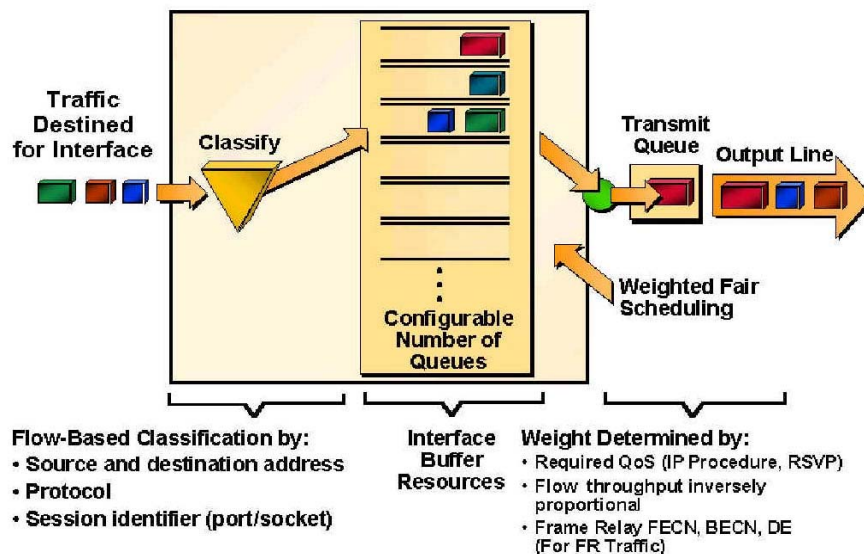


Figura 5 - Filas WFQ [7]

Como pode ser verificado na Figura 5, a classificação dos fluxos de dados pode ser realizada de diversas formas: por endereço fonte ou destino, por protocolo, pelo campo procedência IP, pelo par porta/socket, etc. A quantidade de filas é configurável e a ponderação pode ser estabelecida por precedência IP, ou em conjunto com outros protocolos de QoS como o RSVP, ou ainda em tráfego Frame Relay, como VoFR (*Voice over Frame Relay*) por

exemplo, através dos parâmetros FECN (*Forward Explicit Congestion Notification*), BECN (*Backward Explicit Congestion Notification*) e DE (*Discard Eligible*).

3.3.4 Enfileiramento *Priority Queueing* (PQ)

Numa fila com Enfileiramento PQ (Enfileiramento Prioritário), o tráfego de entrada é classificado em quatro níveis de prioridade: **alta**, **média**, **normal** e **baixa** (*high*, *medium*, *normal* e *low*). Os pacotes não classificados são marcados, por *default*, como **normal**.

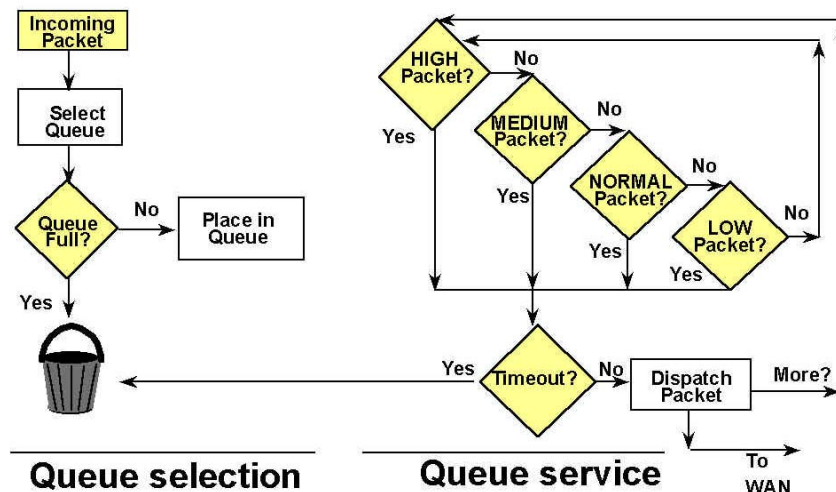


Figura 6 - Operação do Enfileiramento *Priority Queueing* [7]

Dentro de uma dada prioridade os pacotes são tratados pelo mecanismo de FIFO, ou seja, não existe nenhum mecanismo para diferenciar o tratamento entre pacotes de alta prioridade, por exemplo.

A disciplina PQ pode ser configurada para operar de dois modos [7]:

- SPQ (*Strict Priority Queuing*): assegura que os pacotes pertencentes a uma fila de alta prioridade sejam sempre servidos antes dos pacotes das filas de mais baixa prioridade. Por isso, este método deve ser utilizado com cuidado, para evitar longos atrasos e aumento de *jitter* nas aplicações de menor prioridade. Num caso extremo, o tráfego de menor prioridade pode

até nunca ser transmitido, se o de maior prioridade tomar toda a banda. Isso pode acontecer em conexões de baixa velocidade.

- RCPQ (*Rate-Controlled Priority Queuing*): permite que pacotes de maior prioridade sejam escalonados antes de pacotes de menor prioridade se, e somente se, a quantidade de tráfego na fila de alta prioridade permanecer abaixo de um limite configurado pelo usuário para condicionar o tráfego na rede. Porém, se torna difícil configurar limites para a porcentagem máxima de largura de banda alocada para uma determinada fila, quando não se conhece o tamanho dos pacotes, o volume e o comportamento do tráfego gerado por certas classes de serviço.

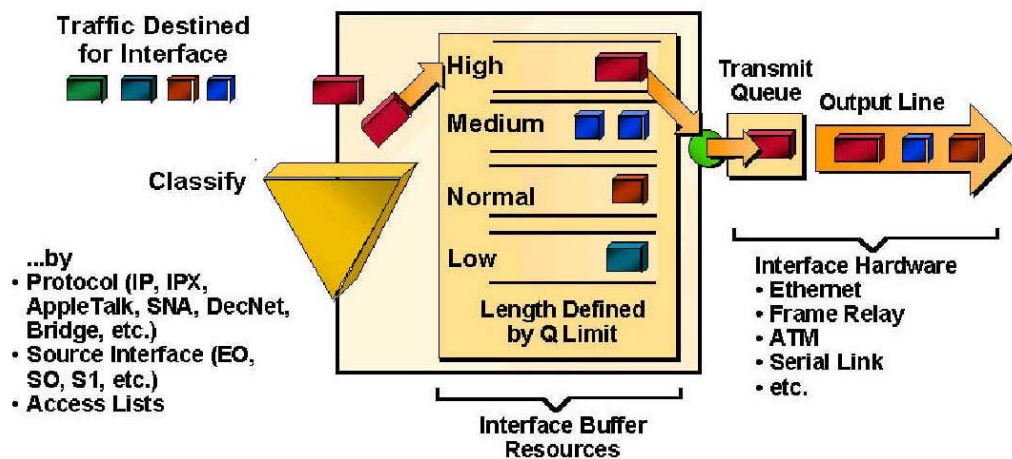


Figura 7- Filas Priority Queuing [7]

Há várias opções de classificação de tráfego numa fila PQ. A classificação pode ser por protocolo (IP, IPX, DecNet, SNA, etc), por interface de entrada ou por lista de acesso.

3.3.5 Enfileiramento *Class Based Queueing* (CBQ)

Também conhecido como CQ (*Custom Queueing*) ou ainda *Weighted Round Robin* (WRR), o CBQ permite especificar uma porcentagem da banda para uma determinada aplicação (alocação absoluta da banda). A banda reservada é compartilhada proporcionalmente, no percentual pré-definido, entre as aplicações e os usuários. O restante da banda é compartilhado entre os outros tipos de tráfego.

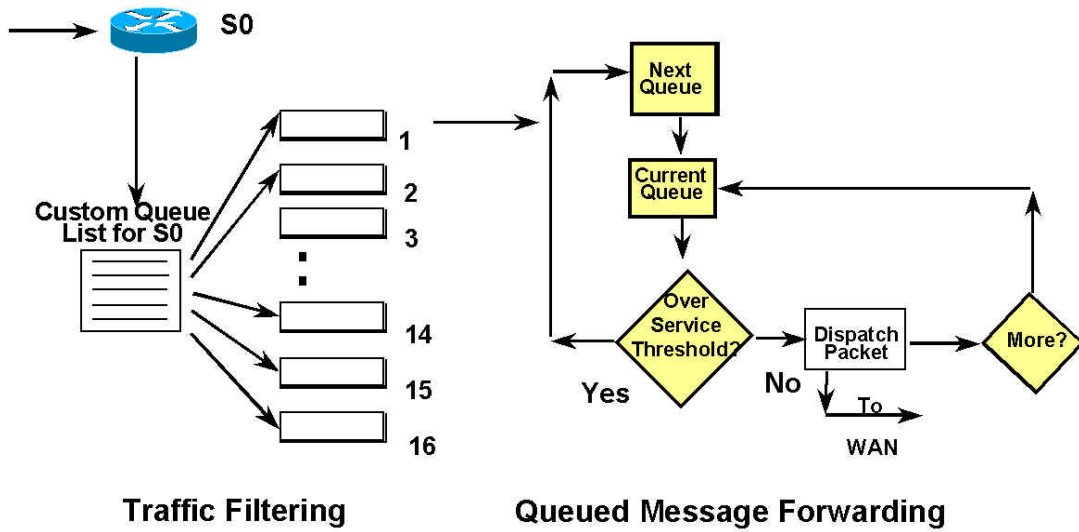


Figura 8 - Operação do Enfileiramento Class Based Queuing [7]

O algoritmo CBQ controla o tráfego alocando uma determinada parte da fila para cada fluxo classificado. As filas são ordenadas ciclicamente num esquema *round-robin*, onde, para cada fila, é enviada a quantidade de pacotes referente à parte da banda alocada antes de passar para a fila seguinte. Associado a cada fila, há um contador configurável que estabelece quantos bytes devem ser enviados antes de passar para a próxima fila.

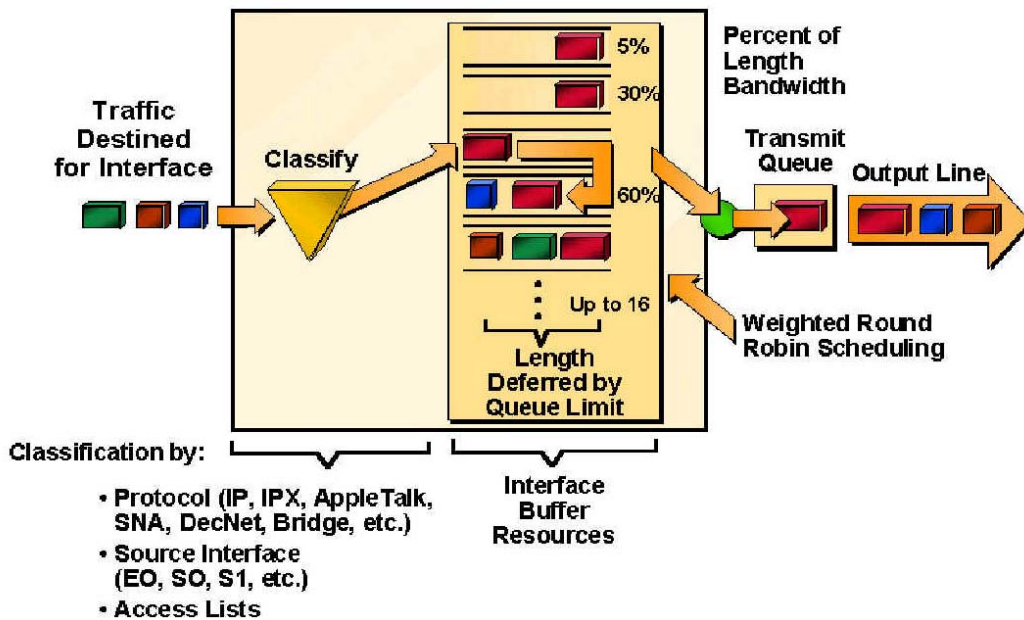


Figura 9 - Filas Class Based Queuing [7]

Até 17 filas podem ser definidas, mas a fila zero é reservada para mensagens do sistema como sinalização, *keep-alive*, etc. A classificação CQ pode ser feita por endereço fonte ou destino, por protocolo (IP, IPX, Appletalk, SNA, DecNet, etc), por precedência IP, por interface de entrada e ainda por listas de acesso.

Weighted Fair Queueing	Priority Queueing	Custom Queueing
• No queue lists	• 4 queues	• 16 queues
• Low volume given priority	• High queue serviced first	• Round-robin service
• Session dispatching	• Packet dispatching	• Threshold dispatching
• Interactive traffic gets priority	• Critical traffic gets through	• Allocation of available bandwidth
• File transfer gets balanced access	• Designed for low bandwidth links	• Best suited for high bandwidth links
• Enable by default		

Figura 10 - Métodos de Enfileiramento [7]

A Figura 10 acima apresenta um resumo dos três métodos de enfileiramento e a Figura 11, a seguir, apresenta algumas considerações sobre a escolha de qual método utilizar. De fato, estas são as diretrizes básicas nas considerações iniciais de um projeto para a implantação de QoS em uma empresa, mas, como todo projeto tem suas próprias peculiaridades, na prática o que prevalecerá será o método e a configuração dos respectivos parâmetros que se enquadrem nas condições do projeto (disponibilidade de banda nas conexões WAN, topologia do backbone, roteamento estático ou dinâmico, etc.) e que produza uma boa qualidade para as aplicações mais importantes da empresa.

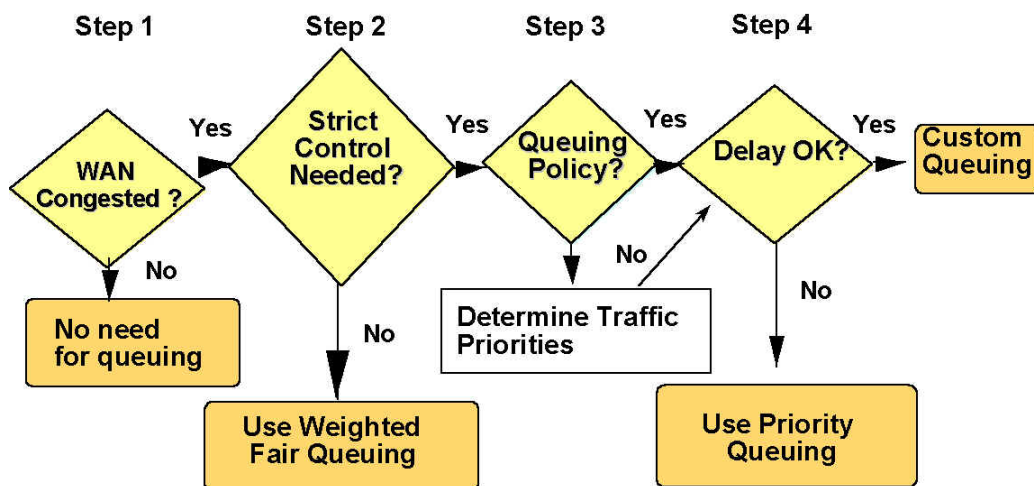


Figura 11 - Esquema para Seleção de Método de Enfileiramento [7]

3.3.6 A detecção RED

A detecção RED - *Random Early Detection* (detecção randômica antecipada) é um mecanismo para prevenção e inibição de congestionamento ou *congestion avoidance*. O algoritmo monitora o tráfego antecipadamente utilizando as funções de controle de congestionamento TCP, descartando pacotes aleatoriamente e indicando para a fonte reduzir a taxa de transmissão, evitando assim situações de congestionamento antes que ocorra picos de tráfego. Quando habilitado numa interface, o RED começa a descartar pacotes a uma taxa que pode ser previamente configurada.

3.3.7 A detecção WRED

A detecção WRED ou *Weighted RED*, é uma implementação que combina as funcionalidades do RED com a classificação de pacotes por precedência IP. Baseado nessa classificação, o WRED descarta pacotes seletivamente, descartando inicialmente os pacotes de menor prioridade, com diferentes pesos para cada classe.

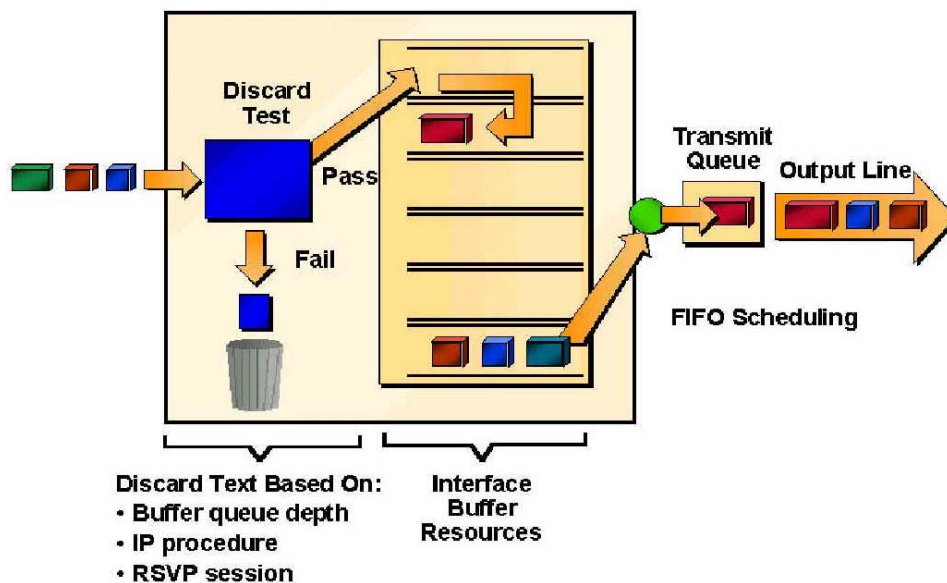


Figura 12 - O Funcionamento do WRED [7]

É possível desabilitar a classificação precedência IP e habilitar o descarte com base apenas no tamanho do buffer da fila; ou ainda utilizar o WRED em conjunto com o RSVP para se obter um descarte mais seletivo. Nesse caso, antes que ocorra uma situação de congestionamento, os fluxos de menor prioridade nas sessões RSVP serão descartados antes dos outros de maior prioridade.

WRED é útil em qualquer interface na qual a possibilidade de congestionamento seja eminente. Entretanto, é geralmente utilizado em roteadores centrais de backbone (*core routers*), com a precedência IP habilitada pelos roteadores de acesso (*edge routers*).

CAPÍTULO 4 - VOZ SOBRE IP (VOIP)

Este capítulo apresenta uma das mais importantes aplicações multimídia, a voz sobre IP, ou simplesmente telefonia IP. VoIP é um termo que engloba a transmissão de voz sobre redes IP (na rede corporativa da empresa ou na Internet). Quando o conceito de VoIP surgiu, este representou uma revolução na maneira em que chamadas telefônicas de longa distância poderiam ser conduzidas. Hoje em dia, entretanto, a VoIP engloba muito mais do que ligações mais baratas de longa distância para amigos e familiares. VoIP é um termo geral para tecnologias que utilizam protocolos Internet, permitindo a conexão entre rede de pacotes e redes comutadas para troca de voz, fax e outras formas de informação que tradicionalmente vinham sendo transportada através de conexões dedicadas de circuito comutado, disponibilizados pela *Public Switched Telephone Network* – PSTN. O desafio da VoIP é o de entregar um fluxo de pacotes de voz de forma confiável ao cliente.

O uso da VoIP possibilita o desenvolvimento de novas aplicações impossíveis de serem implementadas na telefonia tradicional. Além disso, ela permite a redução de custos pela convergência de duas redes tradicionalmente separadas, reduzindo custos de instalação e manutenção, e a economia em chamadas de longa distância. Entretanto, os investimentos para a implementação da VoIP são altos e existe o risco da mudança em uma área tão estratégica como a telefonia.

4.1 Serviço Telefônico PSTN

O serviço telefônico fornecido pela telefonia tradicional, a PSTN (*Public Switched Telephone Network*) usa conexões por comutação de circuito, o que significa que quando uma chamada é feita, é estabelecido um circuito dedicado de um telefone até o outro. Décadas de conhecimento e experiência permitiram ao serviço fornecido pela PSTN alcançar a alta qualidade e a disponibilidade que possui hoje. As redes a comutação a circuitos garante taxa de bits, atraso limitado e constante (sem variação). O nível de qualidade esperada da rede PSTN é denominado de “cinco-noves”, o que significa que a rede como um todo deve estar disponível e funcional para 99,999% do tempo [16].

O serviço telefônico fornecido pela VoIP ocorre de maneira bem diferente daquele oferecido pelas PSTNs. A VoIP faz a transferência de voz como dados em uma rede

IP. Ao invés da comutação a circuito, a conexão telefônica em VoIP é por comutação de pacotes, onde múltiplos dispositivos computacionais compartilham uma rede de dados. O serviço oferecido pela rede geralmente é do tipo melhor esforço, que, ao contrário da comutação a circuito, não fornece garantias de qualidade, gerando um problema na implementação da VoIP.

4.2 Fundamentação da Voz sobre IP

Esta classe de aplicação multimídia apresenta similaridades, em termos de funções, com a telefonia, mas há uma grande diferença: a conversação a pacotes aproveita a infra-estrutura computacional de uma organização para servir também à comunicação entre o pessoal (Figura 13).

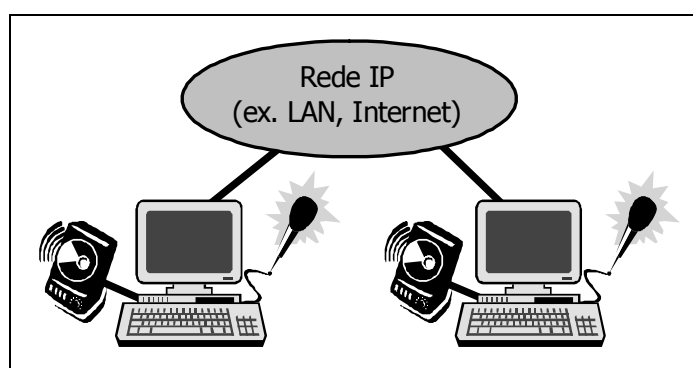


Figura 13 - Telefonia a pacotes de voz [5]

A possibilidade de comunicação de voz através da Internet, em vez da rede de telefonia comutada (PSTN), se tornou realidade em fevereiro de 1995 quando a Vocaltec, Inc. introduziu seu software Internet Phone. Projetado para rodar em um PC 486/33Mhz (ou maior) equipado com placa de som, alto-falante, microfone e acesso à rede, o software compacta o sinal de voz e translada este em pacotes IP para transmissão sobre a Internet.

A Figura 14 apresenta o caminho da mídia nas chamadas telefônicas IP considerando uma comunicação de voz usando computadores.

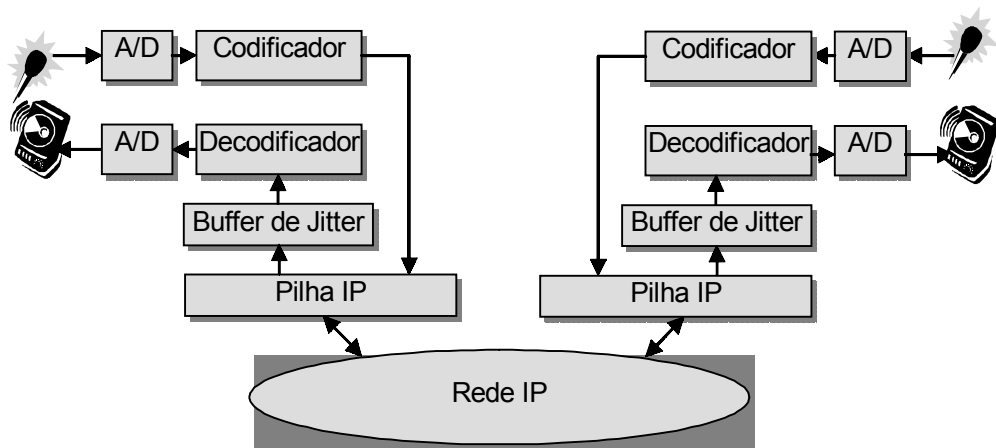


Figura 14 - Elementos envolvidos na comunicação de voz [5]

4.2.1 Conversão AD e DA

A conversão Analógica para Digital (A/D), do sinal elétrico gerado pelo microfone, e a conversão Digital para Analógico (D/A), para geração do sinal elétrico que atinge a caixa de som, são realizadas pela placa de áudio do computador.

4.2.2 Codificações e Decodificações da Voz

Um codec (abreviação de codificador/decodificador ou compactador/descompactador) realiza a compactação e descompactação do áudio para reduzir a largura de banda. O codec pode ser implementado via software, realizando apenas a compressão/descompressão. Ele também pode ser implementado via hardware que realiza as conversões CAD e CDA, além da compactação/descompactação. Normalmente, a codificação via hardware é realizada para codificação de áudio e vídeo de qualidade, por exemplo usando o padrão MPEG, ou então na implementação de terminais de voz (telefone IP).

Existem uma dezena de codecs disponíveis, cada um com suas características. No capítulo 3 foram apresentados vários destes codecs (G.711, G.723, G.728, etc.).

4.2.3 A Pilha IP: Protocolos TCP/IP

A família de protocolos TCP/IP forma a base da Internet e muito presente nas corporações de hoje. Programas enviam e recebem dados sobre a rede IP fazendo chamadas ao software TCP/IP, conhecida como pilha de protocolo. A informação trocada entre hosts na rede é colocada em segmentos de dados (os datagramas IP), tendo um tamanho (tamanho do datagrama) e a identificação associada a cada datagrama (o cabeçalho do datagrama).

Na arquitetura TCP/IP existem dois protocolos de transporte:

- TCP: um protocolo orientado a conexão que garante confiabilidade via retransmissão de pacotes perdidos e realiza o controle de fluxo e de congestionamento.
- UDP: um protocolo que não é orientado a conexão e não garante confiabilidade.

O protocolo UDP é mais adequado para a transmissão de voz, pois este tipo de aplicação é tolerante a perdas de pacotes. Além disso, o UDP é mais adequado para garantir as taxas de transmissão geradas pelos codecs.

A aplicação monta datagramas que contem informações específicas de protocolo. O datagrama de aplicação é encapsulado em um pacote TCP ou UDP. O pacote TCP ou UDP é encapsulado em um pacote IP.

A Figura 15 apresenta o datagrama IP e seu cabeçalho. Para uma rede Ethernet, o tamanho máximo do datagrama IP é de 1500 bytes. Sendo que o tamanho do cabeçalho IP é normalmente 20 bytes, restam 1480 bytes para o nível transporte. O tamanho do cabeçalho TCP é de 20 bytes (M=20) e o tamanho do cabeçalho UDP é de 8 bytes (M=8). No caso do UDP, o tamanho máximo do payload (parte de dados da aplicação) é de 1472 bytes.

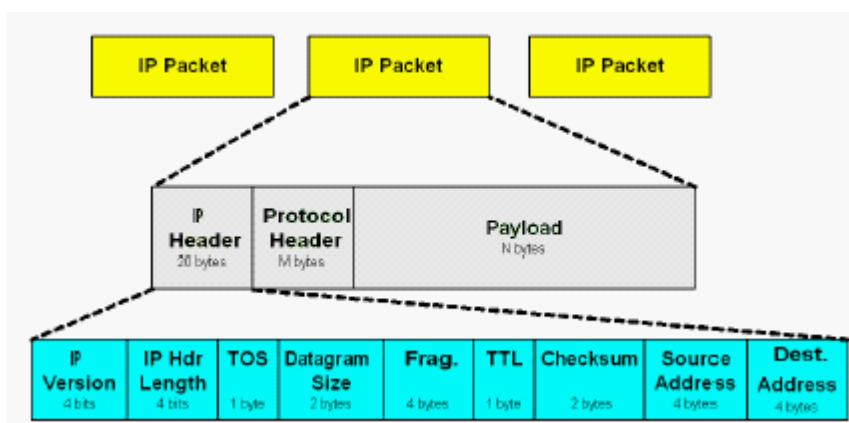


Figura 15 - Datagrama IP e seu cabeçalho [5]

4.2.4 Protocolos de aplicação para VoIP

Programas de aplicações implementam os chamados protocolos de aplicação que utilizam protocolos de transporte (TCP ou UDP). Implementando uma chamada de voz em uma rede digital envolve duas fases (Figura 16): a configuração de chamada (*call setup*) – que é o equivalente VoIP da obtenção do tom de chamada da telefonia, digitação do número do telefone, soar o telefone ou um sinal de ocupado, e tirar o fone do gancho no outro lado; e então a conversação telefônica em si. Protocolos VoIP são necessários durante estas duas fases:

- Vários protocolos podem executar as configurações de chamada e encerramentos de chamada, incluindo H.323, SIP, MGCP, e Melaco. Os programas que implementam o protocolo de configuração de chamada usam TCP e UDP para encapsular o dado trocado durante a configuração da chamada e seu encerramento.
- A troca de dados de voz codificados ocorre após o estabelecimento da chamada, usando dois fluxos de dados – um em cada direção – para deixar os dois participantes falarem ao mesmo tempo. Cada um destes fluxos de dados usa um protocolo de alto nível chamado RTP (*Real-time Transport Protocol*), que é encapsulado em datagramas UDP e IP para transferência.

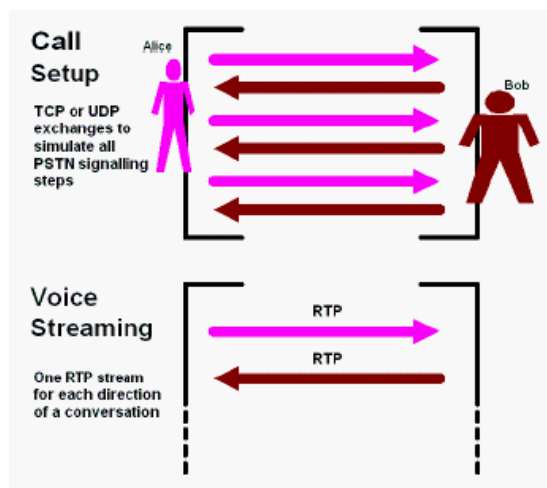


Figura 16 - Fases envolvidas na chamada VoIP [5]

Existem duas famílias de protocolos de configuração de chamadas:

- Os protocolos H.323 e MGCP (Media Gateway Control Protocol) oriundos da comunidade da telefonia. O H.323 é o protocolo mais usual e é na realidade uma família de padrões baseados em telefonia para multimídia, incluindo voz e videoconferência. MGCP é uma versão menos flexível, para uso em dispositivos mais baratos como telefones domésticos.
- SIP (Session Initiation Protocol) e Megaco (Media Gateway Control Protocol) são protocolos leves desenvolvidos pela IETF (comunidade de redes). SIP é um protocolo bem mais leve (que o H.323) e praticamente faz a mesma coisa. Ele é suportado pela Cisco e pela Nortel, e a Microsoft tem recentemente iniciado o desenvolvimento de interfaces clientes SIP com o sistema operacional Windows XP.

Embora a família H.323 de protocolos de configuração de chamada é predominante hoje, o é esperado que os quatro protocolos discutidos acima (H.323, MGCP, Megaco e SIP) deverão usados em iguais proporções nos próximos anos.

RTP é muito usado para *streaming* de áudio e vídeo. Ele é projetado para aplicações que enviam dados em uma direção sem reconhecimento. A Figura 17 mostra o datagrama RTP. Em especial, o cabeçalho de cada datagrama RTP contém uma marca temporal (timestamp), de modo que as aplicações recebem o datagrama pode reconstruir a temporização do dado original. Este cabeçalho também contém um número de seqüência, de modo que o lado receptor possa tratar datagramas faltantes, duplicados e fora de ordem.

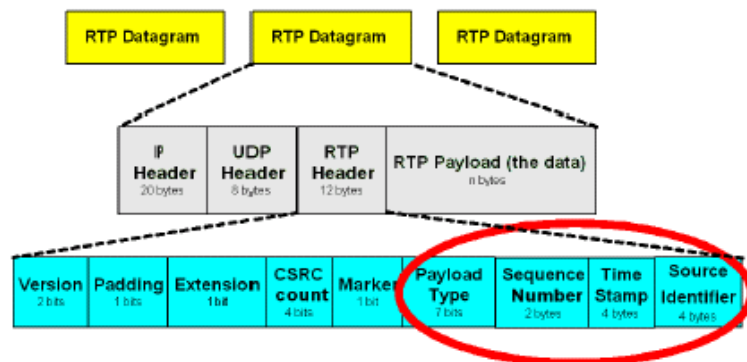


Figura 17 - Pacote RTP [5]

4.3 Telefonia IP

Telefonia sobre a Internet é uma técnica de transmissão de voz sobre rede de dados, que permite a sistemas de telefonia comerciais serem usados para chamadas de pessoas via redes IP. Os componentes essenciais da telefonia IP, além daqueles apresentados na seção 4.2, são (Figura 18): servidores de telefonia IP e PBX, Gateways VoIP, e fones IP.

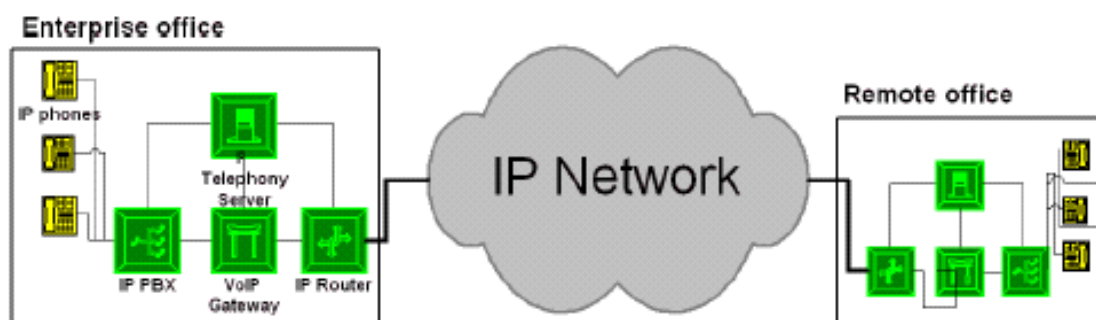


Figura 18- Elementos básicos da telefonia IP [16]

4.3.1 Servidores de telefonia IP e PBX

Muitas transações em redes de dados são baseados no modelo cliente-servidor. Computadores clientes fazem pedidos de serviços a computadores servidores, que desempenham o serviço e retorna o resultado (por exemplo servidores web, email, banco de dados). Com a inclusão de dados de voz nas redes IP adiciona outro conjunto de servidores projetados para prover serviços de voz.

Na PSTN, o PBX é normalmente um sistema “Caixa Fechada” que fornece todas as funções de voz e funcionalidades necessárias de maneira proprietária. O gerenciamento da plataforma fechada é específico do vendedor do PBX. Com a implantação de VoIP, um PBX IP serve como um servidor principal da telefonia IP, fornecendo funções e características semelhantes aos providos por um PBX tradicional. Além destas, os PBXs IP estão provendo novas funções (visto mais adiante). Um PBX IP pode ser construído com um PC rodando um sistema operacional, como o Microsoft Windows, Linux ou Solaris.

Outros tipos de servidores de telefonia IP fornecem novos e interessantes serviços. A possibilidade de serviço de mensagem unificado (*Unified messaging*) – a convergência de mail de voz e e-mail – pode ser considerado um benefício da implementação de VoIP. Servidores de serviço de mensagem unificado também executam sobre plataformas PC e se comunicam com servidores de e-mail e PBXs IP para fornecer acesso a mensagens de vários modos.

Outro novo conceito introduzido com os servidores de telefonia IP é o *clustering*, em que vários destes servidores são agrupados em um *cluster* para oferecer aumento da escalabilidade, confiabilidade e redundância. Servidores em *cluster* funcionam em conjunto e podem ser visto como uma unidade, provendo poder de processamento combinado que logicamente aparece como um único servidor. *Clustering* não é disponível atualmente com PBXs tradicionais em uma PSTN.

Gatekeepers (porteiros) são outro tipo de servidores. Segundo o padrão H.323, os *Gatekeepers* são usados para fornecer características de controle de admissão e outras funções de gerenciamento para serviços multimídia. Estas características são apresentadas mais adiante.

4.3.2 Gateways VoIP

Gateways VoIP e roteadores IP transmitem datagramas de voz RTP por uma rede IP. Gateways VoIP fornecem uma conexão entre a rede VoIP e a PSTN ou a um PBX. Estes dispositivos realizam um papel importante na migração para a VoIP.

É necessário conectar a rede IP a PSTN para colocar chamadas para/de usuários PSTN. Para tanto, Gateways VoIP devem suportar o protocolo SS7 (protocolo de sinalização do PSTN), que é usado pelo gateway VoIP para sinalizar comutações no PSTN quando uma chamada telefônica é originária da rede VoIP com o chamado na PSTN. O gateway executa a digitalização (no caso o sinal de entrada ser um sinal analógico), compressão, demodulação (no caso do sinal de entrada se tratar de um sinal de fac-símile) e funções de empacotamento IP sobre o sinal de voz recebido.

Gateways VoIP podem também fornecer conversação entre diferentes codec, que é chamado *transcoding*. Se um codec diferente do G.711 (por exemplo, G.729), for usado em uma rede VoIP, os dados de voz devem ser convertidos para G.711 antes de serem transferidos para a PSTN.

Em um ambiente corporativo, os gateways VoIP podem interconectar PBXs tradicionais para fornecer um caminho de migração e permitir uma migração gradual ao VoIP. Um exemplo desta utilização de gateways pode ser visto na Figura 19.

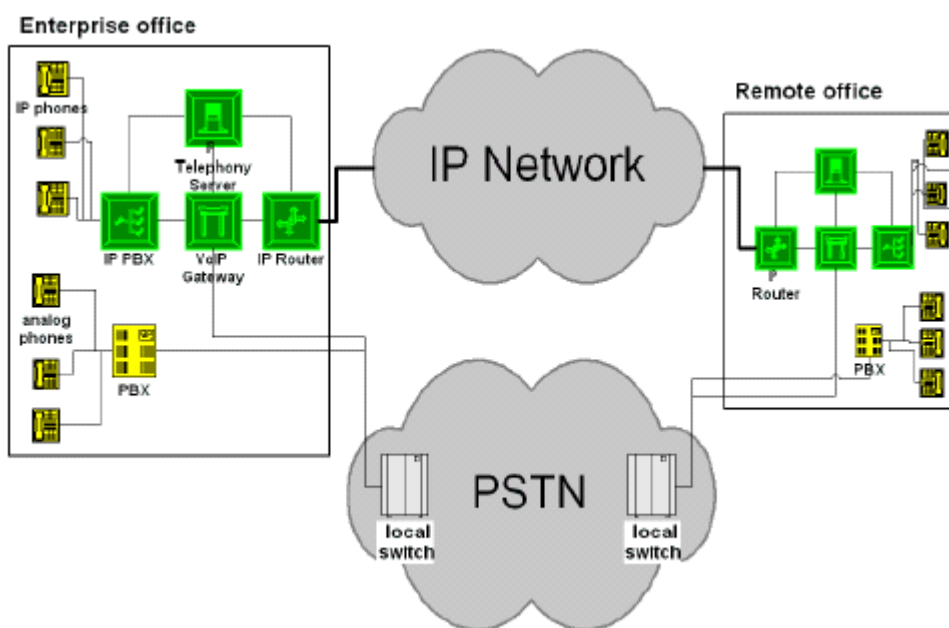


Figura 19 - Uma rede VoIP com seus gateways conectados na PSTN [5]

Na realidade, existem várias opções de empregar telefonia IP [16]:

- Toll bypass usando gateways VoIP: os gateways VoIP são usados para conectar PBXs tradicionais. O tráfego PSTN pode passar através do PBX, para o gateway VoIP, e sobre um backbone de rede IP. Esta opção pode prover redução de custos para chamadas entre sítios das corporações e oferece um primeiro passo para a migração de um emprego de VoIP maior. Alguns cenários adicionais deste tipo de implementação de VoIP podem ser encontrados em [17].
- VoIP com PBXs IP: é um passo a mais para a rede convergente. Integradores oferecem serviços para instalar, configurar, e testar VoIP usando PBX IP em conjunto ou substituindo os PBXs tradicionais.

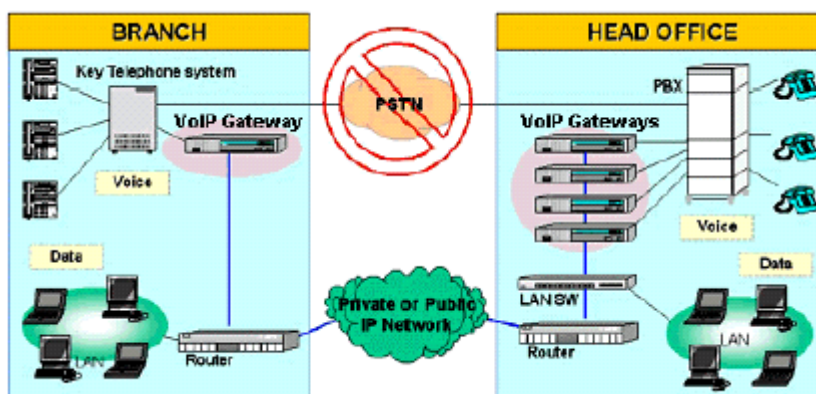


Figura 20 - Toll bypass usando gateways VoIP [5]

4.3.3 Roteador VoIP

Examinando cabeçalhos de pacotes IP, roteadores IP tomam a decisão necessária para mover pacotes para o próximo roteador ou hop ao longo do caminho para o destino, traçando a rota do pacote de voz através da rede.

Um roteador pode ter integrado as funcionalidades de Gateway VoIP, por exemplo um SwitchRouter Gateway IP, as quais a partir da adição de módulos de voz (placas) são capazes de prover a interface de conexão entre o dispositivo de voz (PBX) e a rede de dados, habilitando o transporte de chamadas VoIP.

4.3.4 Endereçamento VoIP

Uma Internet corporativa existente deveria ter um plano de endereçamento IP. Para o esquema de numeração IP, as interfaces de voz aparecem como hosts IP adicionais, ou como uma extensão do esquema existente ou como novos endereços IP.

A translação do número telefônico do PBX para um endereço IP é desempenhada por um DPM (Dial Plan Mapper), onde o número do telefone destino é mapeado para o endereço IP de destino. Quando o número é recebido do PBX, o roteador compara o número com aquele mapeado na tabela de roteamento. Se um casamento é encontrado, a chamada é roteada para o host IP. Após a conexão ser estabelecida, a conexão da Intranet corporativa é transparente para o usuário.

Fones IP geralmente utilizam o DHCP. O servidor DHCP fornece um endereço IP quando um host de rede, neste caso um fone IP, é ativo na rede. Usando o serviço DHCP, fones IP podem ser movidos com relativa facilidade. Quando você realoca um fone IP, movendo ele para outra subrede, por exemplo, o servidor DNS para aquela subrede pode ser capaz de encontrá-lo, amenos que você desabilite o DHCP do fone e coloque um endereço de rede estático. Neste caso, você terá um problema de configuração antes de mover. Por razões similares, se seu servidor DNS não funcionar, você poderia perder o serviço de telefonia. A disponibilidade do servidor de DNS e DHCP necessita ser monitorado. Inclusive, é interessante instalar servidores redundantes aumentar a disponibilidade deste serviço.

4.3.5 Roteamento VoIP

Um dos pontos fortes das redes IP é a maturidade e sofisticação dos seus protocolos de roteamento. O uso de um protocolo de roteamento moderno, tal como Enhanced Interior Gateway Routing Protocol (EIGRP), permite considerar o atraso quando do cálculo do melhor caminho. Além disso, arquiteturas de Qualidade de Serviços (QoS), como Serviços Diferenciados e Serviços Integrados/RSVP (Resource Reservation Protocol) podem ser usados para reservas de recursos para garantir a qualidade de voz.

O uso de tecnologias tag switching e MPLS (Multiprotocol Label Switching) também são interessantes para VoIP, permitindo estender o roteamento IP e facilitam o provimento de QoS, além de permitir a realização da Engenharia de Tráfego.

4.3.6 Equipamentos do Usuário

O equipamento de usuário é o dispositivo usado pelo usuário para solicitar e fazer uso dos serviços de voz. Este equipamento pode ser tanto um aparelho telefônico, sendo ligado a uma central PBX, direto ao gateway ou direto à rede local (telefone IP), ou um telefone baseado em PC, onde o telefone é ligado direto a um microcomputador via uma placa especial ou simplesmente um microcomputador com placa de som e software para telefonia IP (tipo netmeeting).

Para possibilitar a VoIP, o áudio analógico deve primeiro ser convertido em datagramas digitais, realizados pelos codecs. Os codecs podem estar situados no PBX IP ou no próprio telefone. Caso forem utilizados telefones convencionais, os codecs estão localizados no PBX IP. Chamadas que chegam são digitalizadas no PBX IP antes de serem retransmitidos para a rede IP. Outra alternativa é posicionar os codecs no próprio telefone. Estes novos telefones digitais são chamados fones IP. Em vez de ter conectores de telefone de 4 linhas, eles usualmente têm uma conexão LAN Ethernet. Um fone IP faz conexões de dados para um servidor de telefonia IP, que fazem o processamento de configuração de chamada.

Um computador pessoal também pode servir como telefone IP, bastando uma placa de áudio, microfone, fone de ouvidos e uma conexão à LAN. Este computador deve executar um software que realiza a codificação. Como um fone IP, o computador provavelmente se conecta a um servidor de telefonia IP para fazer o processamento da configuração de chamadas.

4.4 Codificadores e Decodificadores de Voz

Existem uma dezena de codecs disponíveis, cada um com suas características. Como por exemplo, o G.711u e G.711a que convertem de um sinal analógico para digital e vice-versa com relativamente alta qualidade. Codecs de baixa velocidade, como o G.726, G.729 e os da família G.723.1, consomem menos largura de banda. Mas codecs lentos reduzem a qualidade do áudio muito mais que codecs de alta velocidade. Isto, pois eles compactam a transmissão digital com perdas.

A Tabela 1 descreve codecs comuns para VoIP. Ela apresenta a taxa de bits gerada pelo codec, o tamanho do pacote de voz típico em ms, o tamanho do pacote de voz em bytes, o tamanho do pacote IP contendo o pacote de voz (incluindo cabeçalhos RTP, UDP e

IP), a taxa de bits a nível de rede desconsiderando supressão de silêncio e a taxa a nível de rede com um VAD de 65%. Note que as taxas apresentadas correspondem a um fluxo de voz unidirecional, se considerado uma conversação com dois fluxos de voz simultâneos, as taxas apresentadas devem ser multiplicadas por dois. Note que as taxas apresentadas são ao nível de rede. A banda aumenta no nível de enlace devido a sobrecarga do cabeçalho desta camada. Por exemplo, se considerado uma rede local Ethernet, com um cabeçalho de 4 bytes, a taxa do G.711 aumenta para 81,6 kbps (desconsiderando supressão de silêncio).

Codec	Taxa nominal do codec (kbps)	Tamanho do pacote de voz típico (ms)	Byte de dados por pacote	Tamanho total do datagrama IP (bytes)	Taxa a nível IP (kbps)	Taxa com VAD de 65% (kbps)
G.711u	64	20	160	200	80,00	57,60
G.711a	64	20	160	200	80,00	57,60
G.726-32	32	20	98	138	55,20	41,48
G.726-32	32	15	60	100	53,33	42,13
G.728	16	20	40	80	32,00	26,40
G.729	8	20	20	60	24,00	21,20
G.723.1m	6,3	30	24	64	17,07	14,83
G.723.1a	5,3	30	20	60	16,00	14,13

Tabela 1 - Codecs comuns em VoIP [17]

Uma seqüência de amostras de voz, de tamanho típico de 20 a 30 ms, são enviadas para a rede na forma de um datagrama RTP. Um dos problemas do uso do RTP é o incremento da sobrecarga de protocolos para envio da voz codificada. Por exemplo, um tamanho típico do payload quando usado o codec G.729 é 20 bytes, isto significa que as saídas do codec são string de 20 bytes da chamada de VoIP que são geradas em uma predeterminada taxa. Com o RTP, dois terços do datagrama é o cabeçalho porque a sobrecarga do cabeçalho consiste de: RTP (12 bytes) + UDP (8 bytes) + IP (20 bytes) = 40 bytes. O consumo da largura de banda real de chamadas de VoIP é maior que a pensada inicialmente. O codec G.729, por exemplo, tem uma taxa de payload de dados de 8 kbps. Seu uso de largura de banda real é maior que isto, quando enviado a intervalos de 20 ms, seu tamanho de payload é 20 bytes por datagrama. Para isto, adiciona-se 40 bytes de cabeçalho RTP e alguns cabeçalhos de camada 2. Por exemplo, drivers Ethernet geralmente adicionam mais 18 bytes. Além disso, a taxa consumida deve ser multiplicada por dois, pois em uma chamada VoIP há geralmente dois fluxos RTP concorrentes (um em cada direção).

O codec G.723.1 tem duas taxas associadas, 5,3 e 6,3 kbps, onde o modo de operação pode mudar dinamicamente em cada quadro. Seu tamanho de quadro de voz é 30 ms, mas um atraso adicional de 7,5 ms é necessário para seu buffer look-ahead, resultando em um atraso total do algoritmo de 37,5 ms.

O codec G.729/G.729A produz uma taxa de 8 kbps e custa um atraso de algoritmo de 15ms (10 ms de tamanho do pacote de voz e 5ms de tempo de look-ahead). G.729A é uma versão de complexidade reduzida da especificação original do G.729.

Alguns telefones IP deixam você setar o “atraso entre pacotes” ou o “tamanho do pacote de voz”, isto é, a taxa na qual o emissor enviará datagramas para a rede. Por exemplo, a 64 kbps, um “datagrama de voz de 20 milissegundos” implica que o emissor criará um datagrama de 160 bytes a cada 20 ms. Existe uma equação simples que relaciona a velocidade do codec, o atraso entre datagramas de voz, e o tamanho do payload do datagrama:

$$\begin{aligned} \text{Tamanho do payload (em bytes)} = \\ \frac{\text{VelocidadeCodec(embps)} * \text{AtrasodoDatagrama}}{8(\text{bits / byte}) * 100(\text{ms / s})} \end{aligned}$$

Neste exemplo: 160 bytes = (64000 x 20)/8000.

Para uma dada taxa, incrementando o atraso causa o aumento do datagrama, desde que os datagramas são enviados com menos frequência para o transportar a mesma quantidade de dados. Um atraso de 30 ms na taxa de dados de 64 kbps poderia significar enviar datagramas de 240 bytes.

4.5 Qualidade de Serviço em VoIP

A qualidade da chamada telefônica é a questão central da telefonia IP. A qualidade oferecida por um serviço a muito tempo estável e que não sofreu profundas mudanças em décadas, a telefonia por comutação, fornece um padrão de qualidade a ser pretendido na telefonia VoIP.

O padrão de Qualidade de Serviço (QoS – *Quality of Service*) que a telefonia IP deve oferecer tem como referência o padrão que a telefonia convencional oferece hoje, padrão ao qual os usuários estão acostumados. A telefonia por comutação (a tradicional) pode fornecer 99,999% de disponibilidade do sistema quando se deseja uma linha telefônica para efetuar uma chamada [16].

Em VoIP, é necessária uma série de ações para se garantir uma boa qualidade de chamada. O requisito de boa qualidade da fala percebida pelo usuário final pode não ser fácil de ser alcançado e mantido, exigindo planejamento da implementação VoIP e, talvez, o

gerenciamento do tráfego VoIP após a implementação do serviço. A satisfação desse requisito pode determinar o sucesso da implementação da voz sobre IP.

Portanto, a telefonia IP traz algumas questões novas que não existem na telefonia convencional, como a garantia da qualidade de serviço. O resultado da QoS é a qualidade de voz percebida pelo usuário, sendo que os requisitos, sob o ponto de vista de um usuário da telefonia IP, são:

- A disponibilidade do serviço deve ser próxima a 100% quando se desejar efetuar uma chamada;
- Deve fornecer boa qualidade da voz;
- Deve fornecer as mesmas funcionalidades às quais os usuários estão acostumados, como por exemplo, sinal de ocupado e transferência de ramal.

Antes de implantar VoIP em uma rede, é recomendado à realização de testes para verificar a qualidade da chamada. É quase certo que uma rede de dados precisará de modificações para garantir uma boa qualidade de chamada telefônica VoIP. Para auxiliar no planejamento da implementação do serviço VoIP, algumas empresas desenvolveram aplicativos para avaliar a qualidade que uma rede de dados pode oferecer. Essa avaliação prévia possibilita o preparo da rede para implementação do serviço de VoIP.

A seguir serão apresentadas algumas formas de avaliar a qualidade de voz percebida pelos usuários dos serviços de telefonia clássica e de VoIP.

4.5.1 Mean Opinion Score

Desde que a telefonia foi inventada, o teste da qualidade de chamada é subjetivo: tirar o fone do gancho e ouvir a qualidade da voz. A principal medida subjetiva da qualidade de é o MOS (Mean Opinion Score), descrito na recomendação ITU P.800 [P.800, 1996]. Ao usar MOS com ouvintes humanos, um grupo de pessoas ouve o áudio e dá suas opiniões sobre a qualidade da chamada.

No MOS, o nível de qualidade de voz é medido por uma escala de cinco pontos cujos valores estão listados e descritos na Tabela 2. É importante notar que os valores obtidos através do MOS são subjetivos e a experiência e herança cultural das pessoas, assim como as mensagens que elas ouvem, são fatores importantes na determinação da medida do MOS.

Nota	Significado
1	Ruim: ininteligível, não é possível entender a mensagem decodificada. Possui interrupções severas devido às degradações
2	Pobre: o sinal possui interrupções devido às degradações, tem-se que fazer um esforço considerável para entender alguns trechos
3	Moderado: a qualidade da voz é ruim, as degradações incomodam, porém não tem interrupções e ainda consegue-se entender a mensagem (requer esforço moderado)
4	Bom: a voz é agradável de se ouvir, ou seja, percebe-se de gradações mas não chegar a incomodar, pois são mínimas (nenhum esforço severo é requerido)
5	Excelente: não se consegue diferenciar o trecho original como corrompido, ou seja, não se percebe a degradação do sinal (nenhum esforço é requerido)

Tabela 2 - Escala de pontuação de níveis de qualidade [5]

O MOS tem o inconveniente de ser difícil e dispendioso de executar. Também há o inconveniente da necessidade de se formar um grupo de pessoas toda vez que for preciso fazer um ajuste na qualidade da chamada. Este procedimento tornou-se desnecessário, pois o padrão de comportamento humano foi extensamente pesquisado e registrado. Para evitar estes problemas, vários padrões para a medição da qualidade da chamada:

- PSQM [P.861, 1998] / PSQM+: Perceptual Speech Quality Measure;
- MNB [P.861, 1998]: Measuring Normalized Blocks;
- PESQ [P.862, 2001]: Perceptual Evaluation of Speech Quality;
- PAMS [British Telecom, 2001]: Perceptual Analysis Measurement System;
- E-Model [G.107, 2002].

Os padrões PSQM, PSQM+, MNB, PESQ E PAMS não são apropriados para medir a qualidade da chamada em uma rede de dados. São baseados na telefonia mais tradicional, tratando a rede de dados como uma grande caixa preta analógica. Não sendo baseados em rede de dados, não conseguem lidar com atraso, variação de atraso e perda de pacote. Também não possuem escalabilidade, o que permitiria avaliar a qualidade de centenas ou milhares de chamadas simultâneas. O E-Model é o padrão recomendado para a avaliação da qualidade da chamada VoIP [16].

4.6 Codecs e a qualidade de voz

É razoável se esperar que quanto mais o sinal de voz é compactado maior é a perda da sua inteligibilidade. Em outras palavras, codecs com baixa velocidade de conversão

de sinal degradam muito mais a qualidade da fala percebida do que codecs de alta velocidade, devido à compactação que fazem do sinal. Como são enviados poucos bits, o lado que recebe faz o possível para reproduzir o sinal o mais próximo do sinal original. Desde o padrão G.711 até o padrão G.728 tem-se um aumento de compressão de 2:1, com a respectiva queda de qualidade, segundo o critério MOS de 4,3 para 4,0.

A Tabela 3 descreve codecs comuns para VoIP, apresentando a taxa do codec, o atraso de empacotamento e a qualidade de voz oferecida. O Atraso do algoritmo refere-se ao atraso de compactação. O atraso total é o atraso gerado apenas nos sistemas finais, considerando o atraso de codificação, decodificação e o atraso do buffer de jitter. Este atraso influencia na qualidade, como se pode constatar pela coluna MOS (*Mean Opinion Scope*), a medida subjetiva da qualidade de voz.

Codec	Taxa de bits no codec	Tamanho do pacote de voz	Atraso do algoritmo	Atraso de bufferização	Atraso na fonte	MOS
G.711	64 kbps	20 ms	0,125 ms	40 ms	20 ms	4,2 a 4,7
G.726-32	32 kbps	15 ms	0,125 ms	40 ms	20 ms	3,9 a 4,2
G.728	16 kbps	20 ms	0,625 ms	40 ms	20 ms	3,7 a 4,3
G.729	8 kbps	20 ms	5,0 ms	40 ms	25 ms	3,9 a 4,2
G.729A	8 kbps	20 ms	5,0 ms	40 ms	25 ms	3,7 a 4,2
G.723.1m	6,3 kbps	30 ms	7,5 ms	60 ms	67,5ms	3,8 a 4,0
G.723.1a	5,3 kbps	30 ms	7,5 ms	60 ms	67,5ms	3,3 a 3,7

Tabela 3 - Codecs e a Qualidade de Voz [5]

4.6.1 Escolha do codec apropriado

Embora tudo aponte para opção de se selecionar um algoritmo de compressão de voz que use a menor taxa de bits, o uso deste tipo de métrica pode não resultar na seleção de um algoritmo apropriado. A escolha de um codec que gera menor taxa de bits resulta em uma qualidade inferior. Um dos componentes que baixam a qualidade é o atraso fim-a-fim destes

codecs. Neste caso, a escolha de um codec que gera maior taxa de bits permite obter uma melhor qualidade de voz (e o atraso fim-a-fim).

O G.711 é o codec recomendado para uso em VoIP por [16]. Uma vantagem é que o G.711 é o codec que causa menos degradação e adiciona o menor atraso. Outra vantagem é que, como a PSTN usa o G.711, ao se usar o mesmo codec na VoIP, evita-se o *transcoding*, que é a conversão da codificação do sinal entre diferentes tipos de codecs. O *transcoding* deve ser suportado pelo *Gateway VoIP*.

Cancelamento de perda de pacotes (PLC - *Packet Loss Concealment*) é uma característica adicional dos codecs G.711u e G.711a. As técnicas PLC reduzem ou mascaram os efeitos da perda de dados durante a conversação de voz. O PLC não adiciona atraso e não tem efeitos maléficis, mas tornam os codecs G.711 mais caros para produzir. Por causa deste custo, esta técnica é relativamente rara hoje.

A recomendação ITU-T G723 foi originalmente projetada para comunicação multimídia em videofones, a uma taxa de 6,3 e 5,3Kbps, com retardo de 67,5ms. Para as aplicações onde o retardo é crítico, este padrão torna-se intolerável. O G.723.1 é, muitas vezes, preferido na VoIP, devido a redução da taxa de bits, importante em redes de pequena largura de banda, como é o caso geral da Internet.

4.6.2 Qualidade de Voz Oferecida pela Rede

Como visto nos capítulos anteriores, aplicações tradicionais em rede e aplicações VoIP têm diferentes requisitos de rede. Aplicações VoIP consomem relativamente pequena largura de banda, mas não pode tolerar grandes atrasos e variações de atraso, além de permitir uma reduzida perda de pacotes. Mesmo quando eles são transportados na mesma rede, o tráfego de voz e de dados não podem ser manipulados do mesmo modo [16]:

- Eles têm diferentes tamanhos de pacotes;
- Eles são enviados em diferentes taxas;
- Eles são buferizados e liberados para o destino de modo diferente;
- Eles devem satisfazer expectativas de usuários muito diferentes.

Um e-mail ou uma transferência de arquivo pode atrasar meia hora sem que ninguém note isto, mas atrasos de poucas centenas de milissegundos podem incomodar uma chamada de voz. E quando se usa VoIP sobre uma dada rede corporativa, os atrasos causados por outras aplicações, roteadores sobrecarregados, e falhas de switches podem ser inevitáveis.

Muitas redes de dados não estão prontas para fornecer o desempenho necessário para chamadas com qualidade e confiabilidade do nível da PSTN. Devido à meta de fornecer boa qualidade de chamada, o tráfego de voz impõe um novo conjunto de novos requisitos ao nível de redes. Voz tem características tempo-real, que tem requisitos muito rígidos para o desempenho da rede.

A Tabela 4 define os níveis de qualidade de voz com base apenas nos parâmetros de desempenho no nível de rede. Note que além destes, outros componentes do equipamento terminal (computadores dos usuários finais) influenciam na qualidade de voz percebida. A tabela abaixo considera apenas a influência da rede na qualidade final da voz.

Categorias de Qualidade	Atraso fim a fim	Média das perdas de pacotes	Varição do atraso	Satisfação do Usuário
Melhor	< 150 ms	00%	0 ms	Muito Satisfeito
Alto	< 250 ms	03%	75 ms	Satisfeito
Médio	< 350 ms	15%	125 ms	Alguns Usuários Insatisfeitos
Baixo	< 450 ms	25%	225 ms	Muitos Usuários Insatisfeitos

Tabela 4 - Parâmetros de Referência de Rede [5]

CAPÍTULO 5 - FREEBSD

A seção a seguir oferece algumas informações básicas sobre o projeto, incluindo uma breve história, seus objetivos, seu modelo de desenvolvimento e algumas de suas principais aplicações.

5.1 História

O projeto FreeBSD teve seu nascimento no início de 1993, em parte como uma consequência do conjunto de manutenção não-oficial do 386BSD (“Unofficial 386BSD Patchkit”) pelos seus 3 últimos coordenadores: Nate Williams, Rod Grimes e Jordan Hubbard.

O objetivo original era produzir um *snapshot* intermediário do 386BSD, de forma a poder corrigir uma série de problemas com este sistema, que o mecanismo de manutenção não era capaz de resolver. O nome inicial do projeto que era “386BSD 0.5” ou “386BSD Ínterim” em referência a este fato.

386BSD era o sistema operacional de Bill Jolitz, que já estava naquele instante sofrendo quase um ano de negligência. Como o mecanismo de manutenção *patchkit* se tornava mais e mais desconfortável a cada dia que passava, os desenvolvedores decidiram ajudar Bill oferecendo a ele este *snapshot* “ínterim”. Tais planos foram bruscamente interrompidos quando Bill Jollitz repentinamente decidiu retirar sua sanção ao projeto sem nenhuma indicação clara do que deveria ser feito.

Não levou muito para ser decidido que o objetivo continuava a valer a pena, mesmo sem a ajuda de Bill, e então se adotou o nome “FreeBSD”, sugerido por David Greenman. Os objetivos iniciais foram definidos depois de consultar os usuários recentes do sistema e, uma vez estando claro que o projeto estava na estrada para, talvez, tornar-se uma realidade, Jordan Hubbard entrou em contato com a Walnut Creek CDROM, com o olho aberto á possibilidade de aperfeiçoar os canais de distribuição do FreeBSD para as pessoas que não tinham acesso à Internet. Walnut Creek CDROM não apenas aprovou a idéia de distribuir o FreeBSD em CD, mas também foi mais longe, ao ponto de oferecer ao projeto uma máquina para trabalho dedicado e uma conexão rápida com a Internet. Sem esta confiança, sem precedentes, da Walnut Creek CDROM no que era, naquele momento, um projeto completamente desconhecido, é muito provável que o FreeBSD não tivesse chegado tão longe e tão rápido ao ponto em que está hoje.

A primeira distribuição em CDROM (e na Internet em geral) foi o FreeBSD 1.0, lançado em Dezembro de 1993. Era baseado na fita 4.3BSD-Lite (“Net/2”) da Universidade da Califórnia, Berkeley (U.C. Berkeley), com muitos componentes originados do 386BSD e da Fundação do Software Livre (*Free Software Foundation*). Foi um sucesso razoavelmente grande para uma primeira aparição, e os continuaram o ciclo com uma versão altamente bem sucedida, o FreeBSD 1.1 release de Maio de 1994.

Por volta desta época, algumas nuvens de tempestade inesperadas começaram a se formar no horizonte, conforme a Novell e U.C. Berkeley acertaram ao longo do processo penal entre ambas, a respeito da situação legal da fita contendo o Net/2 de Berkeley. Uma das condições do acordo eram as concessões da U.C. Berkeley que implicava que grandes trechos do Net/2 fossem códigos “impedidos” e de propriedade da Novell, que havia por sua vez adquirido-os da AT&T algum tempo antes. O que Berkeley recebeu em retribuição foi a “benção” da Novell para o lançamento da versão 4.4BSD-Lite, que quando acontecesse, seria declarado como “impedido” e todos os usuários do Net/2 seriam fortemente encorajados a mudar de sistema para a nova versão. Isso incluiu o FreeBSD, ao projeto foi dado o prazo final de Julho de 1994 para parar de distribuir seu produto baseado na versão Net/2. Sob tais termos de acordo, o projeto poderia lançar uma última versão antes do prazo em questão, o que originou o FreeBSD 1.1.5.1.

O FreeBSD definiu então a árdua tarefa de literalmente se reinventar a partir de um sistema completamente novo e consideravelmente incompleto, o 4.4BSD-Lite. As versões “Lite” continham grandes blocos de código a menos, removidos pelo CSRG de Berkeley (devido a várias decisões legais), códigos necessários para a construção de um sistema inicializável e que podia ser utilizado em produção e o fato é que a conversão do 4.4 para a plataforma Intel era altamente incompleta. O projeto levou até Novembro de 1994 para concluir esta transição, quando lançou a versão 2.0 do FreeBSD na rede mundial e em CDROM (em Dezembro). Apesar de um pouco bruta naquele instante, a versão teve um sucesso significativo, e foi seguida pelo FreeBSD 2.0.5, mais robusto e de mais fácil instalação, em Junho de 1995.

Em Agosto de 1996 foi lançado o FreeBSD 2.1.5, que foi bastante popular entre os provedores de internet (ISP) e as empresas a ponto de justificar a viabilidade de outra versão no ramo 2.1-STABLE. Esta versão foi o FreeBSD 2.1.7.1, lançado em Fevereiro de 1997, que marcou o término do desenvolvimento *mainstream* do 2.1-STABLE. Agora em manutenção, apenas aperfeiçoamentos de segurança e outras correções críticas são realizadas neste ramo (RELENG_2_1_0).

O ramo 2.2 do FreeBSD foi iniciado a partir da série parcial de desenvolvimento (“-CURRENT”) em Novembro de 1996, foi intitulado ramo RELENG_2_2, e a primeira versão completa (2.2.1) foi lançada em Abril de 1997. Versões posteriores ao longo do ramo 2.2 foram criadas no verão e outono de 1998, sendo a última delas (2.2.8) lançada em Novembro de 1998, marcando o início do fim do ramo 2.2.

A árvore foi ramificada mais uma vez, em 20 de Janeiro de 1999, iniciando os ramos 4.0-CURRENT e 3.X-STABLE. A partir da 3.X-STABLE, a versão 3.1 foi lançada, em 15 de Fevereiro de 1999; a versão 3.2 foi lançada em 15 de Maio de 1999; a 3.3 em 16 de Setembro de 1999; a versão 3.4 em 20 de Dezembro de 1999, e a 3.5 em 24 de Junho de 2000, que foi complementada um pouco depois com uma pequena atualização de segurança, o 3.5.1, que incorporava algumas correções de segurança de última hora para o Kerberos. Esta se tornou a versão final para o ramo 3.X.

Outro ramo foi iniciado em 13 de Março de 2000, de forma emergencial na metade do ramo 4.X-STABLE, considerado agora o “ramo -stable corrente”. Posteriormente houveram várias versões desta série: 4.0-RELEASE foi apresentado ao mundo em Março de 2000, e a versão mais recente, 4.10-RELEASE surgiu em Maio de 2004.

A versão 5.0-RELEASE, muito aguardada, foi anunciada em 19 de Janeiro de 2003. O resultado culminante de aproximadamente três anos de trabalho, esta versão colocou o FreeBSD no caminho do suporte avançado a multiprocessamento simétrico, suporte avançado a aplicações multithread e apresentou ao público suporte às plataformas UltraSPARC® e ia64. Esta versão foi seguida pela 5.1 em Junho de 2003. Além de um número muito grande de novas funcionalidades, as versões 5.X do FreeBSD contem ainda uma série de trechos em desenvolvimento em todas as arquiteturas de sistemas relacionadas. Por tal razão, as versões 5.X são consideradas versões de “Nova Tecnologia”, enquanto a série 4.X atua como versões de “Produção”.

No momento apropriado, a série 5.X será declarada estável e o trabalho mais recente será destinado ao próximo ramo de desenvolvimento, o 6.0-CURRENT.

Por hora, projetos de desenvolvimento que requerem mais tempo continuam sendo realizados no ramo 5.X-CURRENT, e versões *SNAPSHOT* da série em CDROM (e lógico, na rede) estão continuamente sendo disponibilizados à partir do servidor de *snapshots* e apresentado como o trabalho em progresso.

5.2 Objetivos do Projeto FreeBSD

Os objetivos do Projeto FreeBSD são de oferecer *software* que pode ser utilizado para qualquer propósito, sem restrições impeditivas. A principal “missão” é oferecer código para qualquer usuário, e para qualquer propósito, de forma que o código em questão chegue o mais longe que puder e que traga os maiores benefícios que forem possíveis.

O código em nossa árvore fonte sob a Licença Pública Geral GNU (GPL) ou Licença Pública Menos Geral GNU (LGPL) contém consideravelmente mais restrições impeditivas, contudo, felizmente mais pelo lado de forçar o acesso irrestrito ao código do que o oposto tradicional. Devido às complexidades adicionais que podem envolver o uso comercial de *software* GPL, tem-se preferência pelos programas lançados sob a licença de direito autoral BSD sempre que possível, por ser uma licença consideravelmente mais flexível.

5.3 O Modelo de Desenvolvimento FreeBSD

O desenvolvimento do FreeBSD é um processo muito aberto e flexível, literalmente composto pela contribuição de centenas de pessoas ao redor do mundo. A infraestrutura de desenvolvimento do FreeBSD permite que centenas de desenvolvedores colaborem mutuamente pela Internet.

Informações úteis de se conhecer sobre o projeto FreeBSD e seu processo de desenvolvimento, seja trabalhando de forma independente ou em cooperação mútua:

- O repositório CVS – A árvore de fontes central do FreeBSD é mantida pelo CVS (Sistema de Versões Concorrentes), uma ferramenta de controle de código fonte disponível gratuitamente, e que vem junto com o FreeBSD. O repositório CVS primário está localizado em Santa Clara, Califórnia, nos EUA, onde ele é replicado em várias máquinas espelho espalhadas ao redor do mundo. A árvore CVS, que contém os ramos -CURRENT e -STABLE, podem ser completamente replicadas, de forma extremamente fácil, à sua própria máquina local.
- A lista de *committers* – Os *committers* são as pessoas que tem acesso de *escrita* na árvore CVS, e tem autorização para realizar modificações diretas na árvore fonte do

FreeBSD (o termo “committer” se origina no comando commit do cvs, que é utilizado para incluir novas mudanças no repositório CVS). Ou seja, os *committers* são os desenvolvedores do FreeBSD.

- O Grupo Central (Core Team) do FreeBSD – O *FreeBSD core team* seria equivalente ao corpo de diretores, se o Projeto FreeBSD fosse uma empresa. A tarefa primária do core team é garantir que o projeto, de forma geral, esteja em boa forma e caminhando nas direções corretas, bem como o recrutamento de novos membros para o core team conforme outros vão se desligando do cargo. O core team atual foi eleito a partir de um grupo de desenvolvedores que se candidataram ao cargo, em Junho de 2002. Eleições para o core team acontecem a cada 2 anos.
- Contribuidores externos – Por último, mas definitivamente não em último, o maior grupo de desenvolvedores são os próprios usuários do sistema, que oferecem retorno e correções de problemas ao projeto de maneira consideravelmente constante.

Em resumo, o modelo de desenvolvimento é um conjunto de círculos concêntricos. O modelo centralizado é projetado com o objetivo de oferecer conveniência aos *usuários* do FreeBSD, que são agraciados com um modelo simples para acompanhar uma base centralizada de códigos, e não para afastar contribuidores em potencial.

5.4 A Versão Atual do FreeBSD

FreeBSD é um sistema cujos fontes foram inteiramente baseados no 4.4BSD-Lite, disponível livremente para computadores baseados em Intel i386™, i486™, Pentium®, Pentium Pro, Celeron®, Pentium II, Pentium III, Pentium 4 (ou compatível), Xeon™, DEC Alpha™ e Sun UltraSPARC. O sistema é baseado primariamente em programas do CSRG (Grupo de Pesquisa em Sistemas Computacionais) da U.C. Berkeley, com algumas melhorias incorporadas do NetBSD, OpenBSD, 386BSD e da Fundação do Software Livre (FSF).

Desde o lançamento do FreeBSD 2.0 no final de 1994, a performance, as funcionalidades, e a estabilidade do FreeBSD foram dramaticamente incrementados. A maior modificação foi à melhoria do sistema de memória virtual com a inclusão do *merged VM/file buffer cache* que não apenas melhora a performance, mas também reduz pedaços perdidos de memória no FreeBSD, tornando a configuração mínima de 5 MB um requerimento mais aceitável. Outras melhorias incluem suporte completo à cliente e servidor NIS, suporte à

transaction TCP, discagem PPP por demanda, suporte DHCP integrado, um subsistema SCSI aperfeiçoado, suporte ISDN, suporte ATM, FDDI, adaptadores de rede *Fast Ethernet* e *Gigabit Ethernet* (1000 Mbit), suporte aperfeiçoado às adaptadoras Adaptec mais recentes, e de correções de outros problemas.

Além da distribuição base, o FreeBSD oferece ainda uma coleção de aplicações (*ports*) composta de milhares de programas populares. A lista de ports varia de servidores http (WWW) à jogos, linguagens de programação, editores de texto e praticamente tudo o mais que se encaixe entre eles. A coleção de ports inteira requer aproximadamente 300 MB de espaço para armazenamento, sendo todos ports identificados como “deltas” à seus fontes originais. Isto torna o processo de atualização dos ports consideravelmente mais fácil para nós, e reduz muito a demanda por espaço em disco, especialmente quando comparado à versão 1.0 da *Coleção de Ports*.

Uma série de documentos adicionais que eventualmente você pode julgar muito úteis durante o processo de instalação e utilização do FreeBSD podem ser encontrados também no diretório `/usr/share/doc` em qualquer estação FreeBSD recente.

5.5 Características

O **FreeBSD** é um sistema operacional **UNIX Multitarefa de 32 bits gratuito**, para computadores pessoais baseados na tecnologia **INTEL** e compatíveis (Arquitetura PC). É um sistema operacional completo, robusto e distribuído com o seu código fonte, o que faz dele um excelente sistema operacional para estudantes de Engenharia ou Ciências da Computação.

O robusto suporte **TCP/IP** do FreeBSD faz dele um sistema ideal e de baixo custo para aplicações **Inter/Intranet**. Veja alguns dos serviços que ele permite implementar: Servidor WWW; Servidor FTP; Servidor Proxy; Servidor Telnet; Servidor IRC; Servidor DNS; Servidor de Mail (POP3 / SMTP); Servidor PPP; Roteador para LAN / WAN; Firewall; Servidor para jogos multiplayer via Internet (QUAKE , etc); e vários outros tipos de aplicações (Bancos de Dados, VPN, e assim por diante).

O FreeBSD pode também ser utilizado com uma interface gráfica Xwindows, transformando seu PC numa verdadeira **Workstation UNIX**. É importante ressaltar que sistemas Unix e Linux no geral utilizam um conceito diferente onde se tem o Servidor X que dá todo o suporte para que seja instalado no sistema um ambiente gráfico, ficando a cargo do usuário a escolha do ambiente que lhe for mais amigável e que trazer mais utilidades, como

por exemplo: KDE, Gnome, Window Maker (desenvolvido no Brasil), Qwvm (Idêntica a interface dos Sistemas Windows 9X), dentre outras.

5.6 QoS no FreeBSD

O FreeBSD quando utilizado na função de Roteador de uma rede utiliza como mecanismo de controle e enfileiramento de pacotes o padrão oferecido pelo protocolo IP o FIFO conforme explicado na sessão 3.3.1 , ou seja, fornece um serviço de melhor esforço para todos os datagramas que transporta. Em outras palavras, o roteador vai tentar entregar os datagramas que recebe o mais rápido possível. Contudo o serviço de melhor esforço não faz nenhuma promessa sobre o atraso fim a fim para um pacote individual. Tampouco faz promessa quanto à variação do atraso de pacote dentro de uma corrente de pacotes.

O maior problema para as aplicações multimídia é quando um dos enlaces (LAN ou WAN) do Roteador está congestionado, neste caso os pacotes começam a ser enfileirados utilizando-se o mecanismo FIFO o qual não disponibiliza de nenhum mecanismo para controle da qualidade de serviço conforme visto anteriormente. É neste momento que entram as aplicações para suporte a QoS no FreeBSD onde com base na instalação e sua configuração pode-se mudar a dinâmica do roteamento dos pacotes dando-se, por exemplo, prioridade de passagem para pacotes de aplicações com maiores exigências como as aplicações multimídia e deixando em “segundo plano” as aplicações que se utilizam do protocolo TCP.

No presente projeto pesquisou-se dois principais softwares para tal finalidade o Dummynet e o AltQ conforme segue abaixo:

5.6.1 Dummynet

O Dummynet foi proposto por Luigi Rizzo [9] como uma alternativa aos simuladores de rede existentes na época. A justificativa dada era que, parâmetros operacionais como largura de banda, atrasos e tamanho de filas eram fáceis de se controlar nos simuladores, porém eles só ofereciam essas possibilidades no modelo simulado não havendo possibilidade de simular em um sistema real.

O Dummynet simula/implementa filas e limitações de banda, delays, perda de pacotes e efeito de multicanaís (*pipes*). Ele também implementa uma variação do WFQ (ver

sessão 3.3.3) chamada de WF2Q+. Ele pode ser usado em estações de trabalho ou em Maquinas atuando como Roteadores ou Bridge.

A lógica de sua utilização consiste na criação de canais (*pipes*) com determinada largura de banda disponível para determinado tipo de tráfego, podendo também atribuir a um canal várias filas (*queues*) com tamanhos, pesos, delay e perda de pacotes, se julgar necessário, diferentes. O Dummynet trabalha em conjunto com o firewall padrão do FreeBSD, o IPFW. Para utiliza-lo basta adicionar regras ao firewall com a seguinte sintaxe:

```
# ipfw add [N1] pipe {N2} tráfego
# ipfw pipe {N2} config bw {B} delay [D] queue [Q] plr [P] weight [W]
```

Onde (as opções entre chaves são obrigatórias e as entre colchetes são de livre escolha):

- N1 é o número da regra no firewall (é muito importante atentar para este número devido ao fato de que os pacotes quando chegam no roteador passam pela seqüência das regras do firewall até que o pacote em questão seja libera ou bloqueado por uma das regras);
- N2 é o número do canal (pipe) este número não tem relação alguma com N1;
- B é a banda disponível pode ser expressa em Bit/s ou Byte/s e pode-se usar o prefixo K ou M, por exemplo, 1000Bit/s = 1Kbit/s;
- D é o delay que o tráfego em questão receberá no roteador é medido em milisegundos (ms);
- Q é o tamanho da fila, pode ser expresso também em Bits ou em Bytes;
- P é a porcentagem de perda de pacotes (plr = Packet Loss Rate) no caso para se ter uma perda de pacotes de 10% seta-se como valor de D 0.1;
- W é o peso, prioridade que esta fila (queue) terá sobre as outras dentro de um mesmo canal (pipe) o valor varia entre 0 e 100 e o valor default é 1, quanto maior o W maior a prioridade para o tráfego em questão.

A seguir mostraremos alguns exemplos de canais:

- Situação 1 – a empresa X com um link qualquer com largura de banda de 400Kbits/s quer priorizar os pacotes udp e limitar o tráfego

FTP para fora da rede em 20Kbps tanto de Download como de Upload (para isso necessita-se das seguintes regras):

```
# ipfw pipe 1 config bw 400Kbits/s
# ipfw queue 1 config pipe 1 weight 95
# ipfw queue 2 config pipe 1 weight 5
# ipfw add queue 1 udp from any to any via fxp0
# ipfw add queue 2 tcp from any to any 21 via fxp0
```

Onde, fxp0 é a interface de rede que está ligada a rede pública (link dedicado).

- Situação 2 – o provedor de internet a rádio (ISP) X quer controlar a banda de seus clientes de forma a distribuir melhor o seu link de 2Mbps entre os clientes e também de poder disponibilizar aos mesmos pacotes de velocidade, por exemplo acesso *Light* de 64Kbps e acesso *Plus* de 256Kbps, neste caso teria-se a disposição as seguintes regras:

```
# ipfw pipe 1 config bw 64Kbit/s
# ipfw pipe 2 config bw 64Kbit/s
# ipfw pipe 3 config bw 256Kbit/s
# ipfw pipe 4 config bw 256Kbit/s
# ipfw add pipe 1 ip from 192.168.1.2 to any in via fxp1
# ipfw add pipe 2 ip from any to 192.168.1.2 out via fxp1
# ipfw add pipe 3 ip from 192.168.1.50 to any in via fxp1
# ipfw add pipe 4 ip from any to 192.168.1.50 out via fxp1
```

Onde, fxp1 é a interface de rede da LAN e o cliente com endereço IP 192.168.1.2 contratou a acesso *Light* e o cliente com o endereço IP contratou o acesso *Plus*.

Esses exemplos foram apenas ilustrativos do que se pode fazer com o Dummynet pode-se explorá-lo muito mais conforme as necessidades em questão(Vide [14]).

5.6.2 AltQ (Alternate Queueing)

O AltQ não é apenas uma ferramenta para possibilitar mecanismos de QoS, é também um framework que permite gerenciar as disciplinas de fila de pacotes na rede [13]. O AltQ introduz novas funções para a manipulação das filas de pacotes em um roteador ou em uma estação de trabalho, permitindo que se escolha como mecanismos de Enfileiramento de pacotes o: PQ, WFQ, CBQ citados no Capítulo 3, dentre outros além de algoritmos de controle de congestionamento como: RED (Item 3.3.6) RIO, BLUE e o CDNR (não abordados no presente relatório).

Para a experimentação prática proposta no presente relatório optou-se por utilizar o AltQ devido à possibilidade de utilização de outros mecanismos de controle da fila de pacotes, ao contrário do Dummynet (Item 5.1) que permitia apenas a implementação de Filas WF2Q+.

A lógica de sua utilização consiste na criação de classes com determinada largura de banda disponível e a criação de filtros (tipos de tráfego) com determinada porcentagem de tráfego relativo à classe a que pertence, podendo também atribuir a uma classe atributos como prioridade, algoritmo de controle de congestionamento, seqüência de filtragem, dentre outros. O AltQ é uma solução para fornecimento de QoS e trabalha independente do sistema de firewall, ao contrário do Dummynet que permite apenas o uso do IPFW, o AltQ trabalha com o PF, IPFW, e qualquer outro sistema de firewall existente na plataforma em questão. Para utiliza-lo basta criar um arquivo de configuração, ou seja, um arquivo texto qualquer com as regras em seu conteúdo com a seguinte sintaxe:

Definição da interface a ser controlada:

```
# interface if_name [ bandwidth bps ] [ tbrsize bytes ] [ sched_type ]
```

Onde:

if_name – nome da interface de rede onde o AltQ irá controlar o fluxo de pacotes;

bandwidth – é a taxa de Bits da Interface de rede, por exemplo, *bandwidth 1M*;

tbrsize – especifica o tamanho do “balde” (bucket) para o algoritmo de controle de congestionamento utilizado seja o token bucket;

sched_type – especifica o algoritmo de enfileiramento de pacotes, por exemplo, blue, cbq, fifoq, hfsc, pq ou wfq.

Definição das classes de tráfego (utilizada para os algoritmos CBQ, PQ e HFSC), onde neste caso será abordado apenas os comandos relativos a filas controladas pelo algoritmo CBQ, informações sobre a utilização dos demais algoritmos consultar [15]:

O Comando class especifica a classe CBQ. As classes são organizadas de forma hierárquica, todas as classes, exceto a classe root, tem que ter uma superclasse.

```
# class sched_type if_name class_name parent_name [ priority pri ] [ pbandwidth percent ] [ exactbandwidth bps ] [borrow] [default] [maxdelay msec] [red|rio]
```

Onde:

sched_type – neste caso será utilizado o valor cbq;

if_name – interface de rede a qual o AltQ irá controlar o fluxo de pacotes;

class_name – nome da classe a qual se está criando;

parent_name – nome da superclasse;

priority – prioridade que esta classe terá sobre as outras (valor mínimo é 0 e o máximo é 7);

pbandwidth – porcentagem da banda utilizada (valor entre 0 e 100);

exactbandwidth – o valor da banda em bits por segundo;

borrow – caso o parâmetro borrow seja utilizado a classe em questão poderá usar o tráfego da superclasse para a banda que exceder seu valor de pbandwidth ou exactbandwidth, caso contrário à banda excedente sofrerá atraso ou perda dos pacotes;

default – a classe definida como default atenderá os tráfegos não especificados pelo conjunto de regras do AltQ;

maxdelay – é o delay máximo que os pacotes pertencentes a classe em questão sofrerão, este valor é utilizado para definir o tamanho da fila e caso não seja setado o tamanho padrão da fila será 30 pacotes.

red | rio – algoritmos de controle de congestionamento.

Maiores informações a respeito da sintaxe do AltQ podem ser obtidas em [15].

CAPÍTULO 6 - AVALIAÇÃO DO SERVIÇO DE QoS NO FREEBSD

Este capítulo apresenta uma análise de desempenho de encaminhamento de datagramas IP em uma rede provida de mecanismos de QoS e sem QoS considerando o uso de roteadores FreeBSD em uma rede local. Para isso, o ambiente de testes será usado em dois momentos distintos: primeiro momento utilizando o roteamento IP clássico; e segundo momento usando roteamento com QoS, ou seja, o roteador irá controlar o fluxo de pacotes no enlace L2 priorizando os pacotes de aplicações VoIP e controlando o fluxo das demais aplicações de forma a não interferirem no uso das aplicações de VoIP mais sensíveis a variações de tráfego na rede. Nestes dois momentos, serão realizados 2 testes, o primeiro enlace não-saturado e o outro com o enlace saturado, visando medir o desempenho e os resultados são confrontados. A duração de cada teste será de 60s.

Esta análise visa comparar, em situações de tráfego igual e trafegando pelos mesmos nós com as mesmas condições, é revelar se um Roteador FreeBSD é adequado para efetuar o do Controle de Banda de forma a priorizar o tráfego VoIP de uma empresa.

6.1 Definição do Ambiente de Testes

O primeiro passo para a realização da avaliação de desempenho pretendida foi definir a topologia de rede mais adequada ao nosso propósito. O principal requisito dessa topologia é dispensar o uso de mecanismos de sincronização de relógio convencionais como, NTP de forma que as estatísticas geradas no experimento não sejam influenciadas por este tipo de problema e também a simulação de um link de saída com 1Mbps (Enlace L2 conforme podemos observar na Figura 21) alcançado através do uso do AltQ primeiramente controlando o tráfego e utilizando o mecanismo de enfileiramento de pacotes FIFO e no segundo teste além de controlar o tráfego utilizou-se o mecanismo de enfileiramento CBQ, devido à possibilidade da criação de regras que otimizem o uso da banda ao máximo.

A topologia criada está disposta conforme a figura 21. O ambiente de testes definido tem 1 roteador e 2 computadores sendo um deles para gerar e o outro para gerar e receber tráfego, todos com Sistema Operacional FreeBSD.

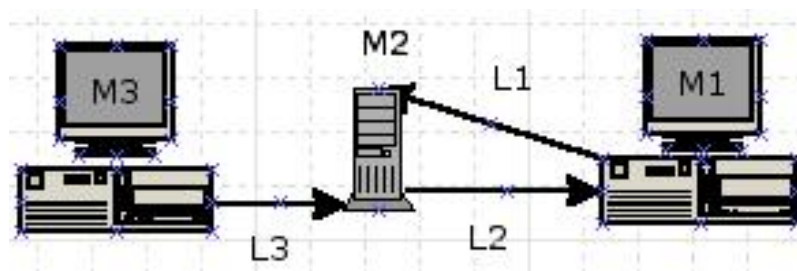


Figura 21 - Plataforma de Teste

A máquina 1 (M1) é geradora/receptora de tráfego, simulando neste caso o uso de aplicações VoIP, para o *testbed*. O roteador 2 (M2) efetua além do roteamento o condicionamento de tráfego e recebe tráfego de fundo vindo da máquina 3, tráfego TCP simulando o uso de aplicações HTTP, FTP, POP e SMTP, por exemplo.

6.2 Características do Equipamento Roteador

Definida a topologia, passa-se para a etapa de montagem do ambiente de teste. Como pressuposto inicial, optou-se pelo uso de um roteador implementado na forma de computador pessoal usando FreeBSD e a tecnologia de rede Ethernet. No ambiente de teste implantado, o roteador FreeBSD tem a versão 4.9 RELEASE e possui o kernel configurado para ter suporte ao AltQ 3.0 configuração esta disponível no Anexo 1.

Por padrão, o Sistema Operacional FreeBSD possui uma fila com tamanho igual a 10 pacotes por interface. E, como o tamanho das filas não foi objeto de análise e investigação desse projeto, foram adotados esses valores padrões.

Como o sistema FreeBSD é de propósito geral, ele não é uma opção ideal para servir como roteador e ao mesmo tempo como estação de trabalho para uso geral. Para evitar esses problemas, foram adotadas algumas políticas para inibir grande parte dos serviços não necessários do FreeBSD e deixá-lo apenas como roteador.

O roteador FreeBSD está provido de placas de rede Intel 82559 10/100. Um fator que merece destaque para análise de desempenho em uma rede é a utilização de placas de rede bem projetadas e de boa qualidade. Em um estágio inicial foram utilizadas placas de redes de baixo custo, onde se deparou com vários problemas na realização dos testes. Um dos principais problemas detectados é a ocorrência de um valor excessivo de perda de pacotes mesmo com uma taxa de transmissão abaixo do limite da capacidade dos enlaces. Esse

problema foi resolvido a partir do momento em que se utilizaram placas de rede de melhor qualidade, permitindo realizar avaliações que estavam de acordo com a configuração da ferramenta geradora de tráfego.

6.3 Métricas utilizadas na avaliação dos testes

Para a avaliação do desempenho nesse projeto, foram adotadas as seguintes métricas:

- **Atraso por pacote:** É o intervalo temporal entre a transmissão do pacote pela fonte e a recepção do mesmo no destino. Um dos grandes responsáveis por este atraso é o processamento nos roteadores.
- **Variação no atraso (*jitter*):** É a variação do atraso sofrido por dois pacotes subjacentes. Caso tenha um valor baixo, ele indica que os pacotes estão sofrendo praticamente o mesmo atraso de processamento nos roteadores. Por meio dessa métrica, consegue-se analisar quão constante é o roteamento de pacotes processado nos roteadores.
- **Vazão:** Indica a taxa de transmissão obtida pela fonte dos pacotes.
- **Perdas:** Indica o percentual de pacotes que foram descartados por fluxo numa transmissão de dados.

6.4 Escolha de ferramentas para realizar os testes de desempenho

Para realização das medições, foram avaliadas várias ferramentas para geração e análise de tráfego, dentre elas estão a Netperf [20] e RUDE & CRUDE [21].

O Netperf (*Network Performance*) utiliza um esquema de gerador e receptor de tráfego. Onde na máquina geradora (M3) chama-se um arquivo binário passado como parâmetros às características dos pacotes a serem enviados como duração da transmissão e tamanho dos pacotes e na máquina receptora (M1) roda-se apenas um processo *Daemon*, que fica rodando e escutando em uma porta TCP aguardando o envio de tráfego por uma outra máquina.

A ferramenta RUDE & CRUDE utiliza um esquema gerador e receptor de tráfego. O gerador gera um tráfego UDP configurável para um receptor especificado. O receptor armazena os pacotes UDP sem decodificá-los, de forma a não gerar carga de processamento na máquina receptora, em arquivo para posterior cálculo de desempenho. Pó ser unidirecional, ela não necessita de uma comunicação *full-duplex*. Entretanto, seu uso implica na necessidade de sincronização de relógios do transmissor e do receptor. O RUDE & CRUDE deixa a decodificação do tráfego de rede para depois, consistindo nesta característica sua principal vantagem perante as demais ferramentas, dos experimentos sem influenciar na avaliação do tráfego. Desta forma, o RUDE & CRUDE foi o escolhido para realizar as medições no ambiente de testes. Outra vantagem desta ferramenta é a análise de tráfego por pacote, podendo medir informações de atraso por pacote, quesito mais importante no contexto de velocidade de encaminhamento.

Para a sincronização do relógio, efetuou-se a configuração da máquina 1 (M1) para que não efetuasse o *Loopback* ao enviar tráfego da interface de rede pertencente ao enlace L1 para a interface de rede pertencente ao enlace L2 passando pelo roteador 1 (M2), onde será efetuado o controle do tráfego.

Uma constatação observada na utilização das ferramentas em geral foi à influência do não determinismo do FreeBSD na máquina receptora no momento de coletar o tráfego da rede. O processo de armazenamento do tráfego em disco rígido causava variações na análise do tráfego. Isto, pois os acessos a disco geravam uma concorrência no tratamento das interrupções, influenciando o desempenho da placa de rede e do sistema em si. Em consequência disso, ocorriam algumas perdas de pacotes aleatórios prejudicando assim a análise do teste. A solução adotada para armazenar o tráfego de rede foi utilizar uma das partições do disco com o sistema de UFS juntamente com Softupdate, o qual trabalha com os arquivos primeiramente em memória para depois sincronizá-lo com o disco, solucionando esse problema de acesso a disco.

6.5 Definição do Tráfego de Teste

Como já foi citado, os testes tiveram duas etapas: uso apenas do roteamento clássico IP e o condicionamento de pacotes utilizando o mecanismo de Enfileiramento de Pacotes CBQ. Para ambos os testes, a máquina Receptora 1 foi o destino dos dados gerados

pelas máquinas Geradoras 1 e 2 (figura 21). Os dados gerados foram do tipo CBR (Constant Bit Rate) adotando o protocolo de transporte UDP.

O caso avaliado configurou-se a máquina M1 para gerar tráfego UDP, simulando o uso de aplicações VoIP neste caso o tráfego gerado é de 320Kbps (4 ligações simultâneas com CODEC G.711 sem supressão de silêncio, neste caso gerando uma taxa constante (CBR) de 80kbps para cada ligação) e M3 para gerar tráfego TCP, pacotes com 1024Bytes.

Em todos os testes apresentados neste relatório, tanto o tráfego de teste como o tráfego de fundo foram CBR (Constant Bit Rate) usando o protocolo UDP (tráfego enviado por M1) e TCP (tráfego enviado por M3). Cabe aqui ressaltar que o tráfego CBR não é o tráfego majoritário na Internet. Em geral, na Internet o tráfego é VBR (Variable Bit Rate) usando TCP. O tráfego tipo CBR foi adotado para facilitar a análise. Devido ao fato de não se ter uma ferramenta que gerasse um tráfego TCP constante a uma determinada taxa, foi utilizado o seguinte mecanismo que M3 pudesse gerar o tráfego em 2 níveis 200Kbps e 800Kbps respectivamente. Para o primeiro caso criou-se um *pipe*, através do uso do DUMMYNET, de 200Kbps com isso obteve-se um fluxo constante de 200Kbps e para o segundo caso realizou-se o mesmo procedimento só que com um *pipe* de 800Kbps (as regras utilizadas para a criação destes *pipes* podem ser analisadas no Anexo3).

Foram feitos 2 testes sendo o primeiro teste com o roteador IP normal e o segundo com o Roteador efetuando o controle do tráfego e enfileiramento dos pacotes seguindo o mecanismo CBQ e cada teste com dois momentos, o primeiro momento com o enlace não-saturado (onde M3 gerava um tráfego de 200Kbps) e segundo momento com o enlace saturado (onde M3 gerava um tráfego do 800Kbps).

6.6 Diferenciação de Tráfego

Para efetuar os testes se teve que pensar em uma solução para que o ambiente de simulação ficasse o mais próximo possível do descrito, ou seja, utilizando-se a tecnologia Ethernet teríamos que simular o link de uma empresa com 1Mbps (Enlace L2) de *Upsteam*, durante o teste sem QoS. Para isso efetuou-se a configuração do Roteador para operar com o Atq usando como mecanismo de enfileiramento de pacotes o FIFO. A configuração do roteador, no caso, do AltQ pode ser observada abaixo (criou-se o arquivo de configuração com o conteúdo abaixo, neste caso chamado de `altq.conf.fifo`):

```
#Simula um link com upstream de 1Mbps usando FIFO
```

```
interface fxp2 bandwidth 1M fifoq
```

Para os testes em que o enlace L2 operava com controle de tráfego, com o mecanismo de enfileiramento de pacotes CBQ, utilizou-se a seguinte configuração (criou-se outro arquivo de configuração com o conteúdo abaixo, neste caso chamado de altq.conf):

```
#Simula um link com upstream de 1Mbps
```

```
interface fxp2 bandwidth 1M cbq
```

```
class cbq fxp2 root NULL pbandwidth 100
```

```
# meta classe
```

```
class cbq fxp2 def_class root borrow pbandwidth 100 default
```

```
# Tráfego VoIP
```

```
class cbq fxp2 voip def_class priority 7 borrow pbandwidth 40 red
```

```
filter fxp2 voip 0 5001 0 0 17
```

```
filter fxp2 voip 0 0 0 5001 17
```

```
# Tráfego TCP
```

```
class cbq fxp2 tcp_class def_class priority 1 borrow pbandwidth 55
```

```
filter fxp2 tcp_class 0 0 0 0 6
```

Para melhor entendimento das configurações mostradas anteriormente, ver Item 6.2.

6.7 Resultados dos testes experimentais de desempenho IP x QoS

Esta seção apresenta os resultados da análise comparativa do uso das tecnologias IP convencional (FIFO) e IP/AltQ alterando o mecanismo de enfileiramento de pacotes e efetuando o condicionamento de tráfego em duas situações de rede: uma a rede não saturada e

a outra com a rede saturada. As medidas apresentadas neste relatório são referentes ao tráfego gerado pela máquina Geradora 1 (tráfego VoIP). As medidas foram obtidas pela máquina 1 e decodificadas utilizando-se o script conforme se pode observar no Anexo 4.

6.8 Análise do Atraso fim-a-fim (*delay*)

As Figuras 22 e 23 apresentam o atraso sofrido pelos pacotes UDP, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X AltQ/CBQ com a rede não saturada (Figura 22) e com a rede saturada (Figura 23). Os resultados obtidos neste experimento são apresentados resumidamente nas tabelas 5 e 6.

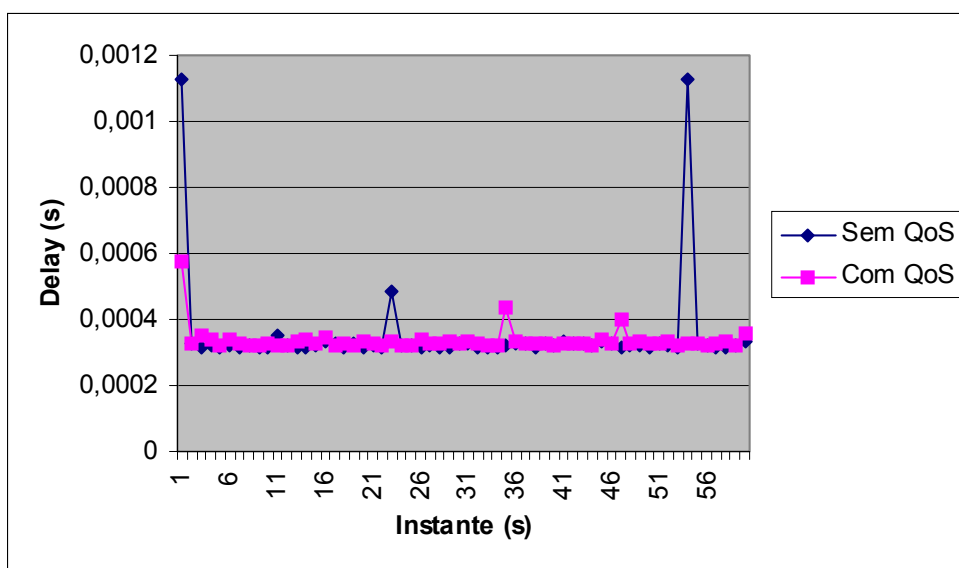


Figura 22 - Delay rede não saturada

Analisando o Gráfico (Figura 22), pode-se observar que o atraso dos pacotes (*delay*) nas duas situações se mantêm praticamente iguais sendo que com QoS ficou com poucas oscilações, porém inferiores se comparado à rede operando sem QoS. Podendo-se concluir que a rede em estado não saturado possui uma melhor estabilidade em seu tráfego quando o roteador opera controlando o tráfego (com QoS).

A Tabela 5 mostra maiores informações, como forma de complementar as informações analisadas e visualizadas na Figura 22.

	Sem QoS	Com QoS
Total	0,021069	0,020144
Mínimo	0,000313	0,00032
Máximo	0,001127	0,000577
Média	0,000351	0,000336

Tabela 5 - Estatísticas do Delay com a rede não saturada

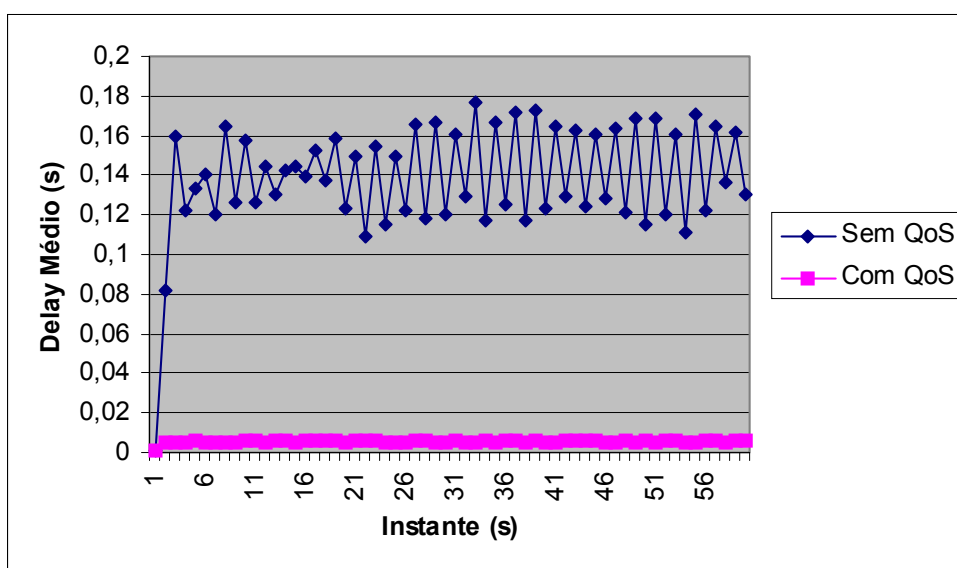


Figura 23 - Delay com a rede Saturada

Analisando o Gráfico (Figura 23), pode-se observar que o atraso dos pacotes (*delay*) manteve-se praticamente constante e com valores bastante inferiores para a rede operando com QoS, enquanto que com rede operando sem QoS o delay além de ser alto, o que inviabilizaria o uso de aplicações VoIP durante este período, oscilou bastante durante o período de teste. Podendo-se concluir que a rede em estado saturado é imprescindível para uma melhor estabilidade em seu tráfego VoIP o uso de um roteador controlando o tráfego (com QoS).

A Tabela 6 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 23.

	Sem QoS	Com QoS
Total	8,321908	0,330191
Mínimo	0,00129	0,001045
Máximo	0,176748	0,006359
Média	0,138698	0,005503

Tabela 6 - Estatísticas do Delay com a rede Saturada

Concluindo o parâmetro atraso fim-a-fim, constatou-se que, no ambiente de teste definido, o QoS teve um ótimo desempenho, tornando, mesmo em situações em que a rede esteja saturada, viável o uso de aplicações VoIP.

6.9 Análise da Variação de Atraso (*Jitter*)

As Figuras 24 e 25 apresentam a variação de atraso (*jitter*) sofrido pelos pacotes UDP, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X AltQ/CBQ com a rede não saturada (Figura 24) e com a rede saturada (Figura 25). Os resultados obtidos neste experimento são apresentados resumidamente nas tabelas 7 e 8.

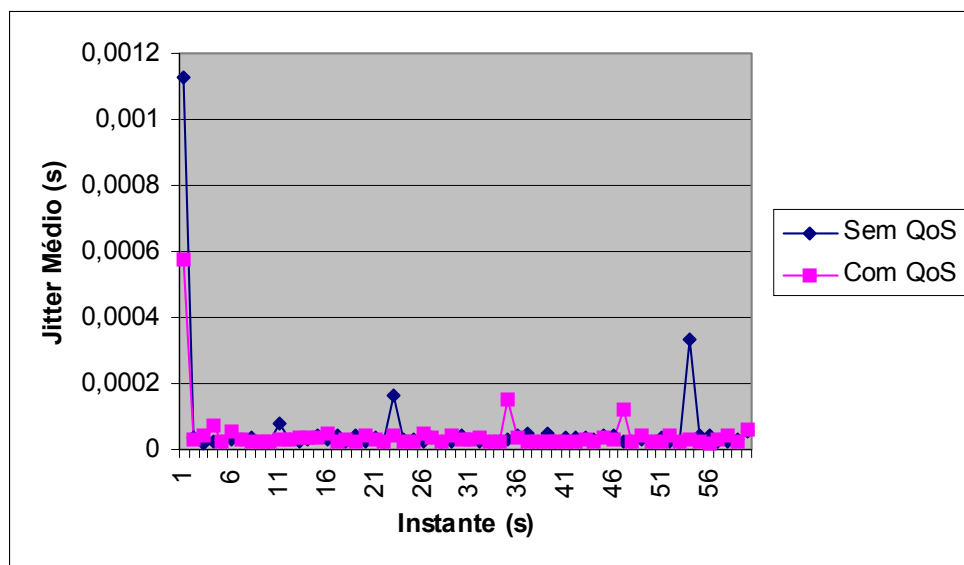


Figura 24 - Jitter com a rede não saturada

Analisando o Gráfico (Figura 24), pode-se observar que a variação do atraso dos pacotes (*jitter*) nas duas situações se mantêm praticamente iguais sendo que com QoS ficou com poucas oscilações, porém inferiores se comparado à rede operando sem QoS. Podendo-se concluir que a rede em estado não saturado possui uma melhor estabilidade em seu tráfego quando o roteador opera controlando o tráfego (com QoS).

A Tabela 7 mostra maiores informações, como forma de complementar as informações analisadas e visualizadas na Figura 24.

	Sem QoS	Com QoS
Total	0,003464	0,002665
Mínimo	0,00002	0,000018
Máximo	0,001125	0,000577
Média	5,77E-05	4,44E-05

Tabela 7 - Estatísticas do Jitter com a rede não saturada

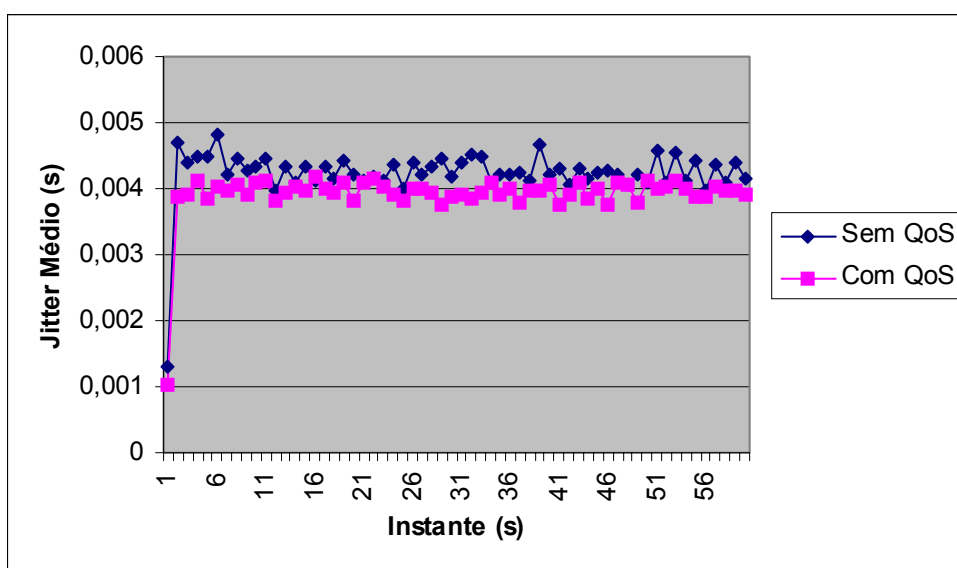


Figura 25 - Jitter com a rede saturada

Analisando o Gráfico (Figura 25), pode-se observar que a variação do atraso dos pacotes (*jitter*) manteve-se praticamente constante e com valores ligeiramente inferiores para a rede operando com QoS, enquanto que com rede operando sem QoS o *jitter* oscilou bastante e com valores sempre superiores a rede operando com QoS, o que é prejudicial para o uso de aplicações VoIP. Podendo-se concluir que a rede em estado saturado é imprescindível para uma melhor estabilidade em seu tráfego VoIP o uso de um roteador controlando o tráfego (com QoS).

A Tabela 8 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 25.

	Sem QoS	Com QoS
Total	0,254055	0,235085
Mínimo	0,00129	0,001045
Máximo	0,004807	0,004173
Média	4,23E-03	3,92E-03

Tabela 8 - Estatísticas do Jitter com a rede Saturada

6.10 Análise da vazão

As Figuras 26 e 27 apresentam a vazão que os pacotes UDP obtiveram durante os testes, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X AltQ/CBQ com a rede não saturada (Figura 27) e com a rede saturada (Figura 28). Os resultados obtidos neste experimento são apresentados resumidamente nas tabelas 9 e 10.

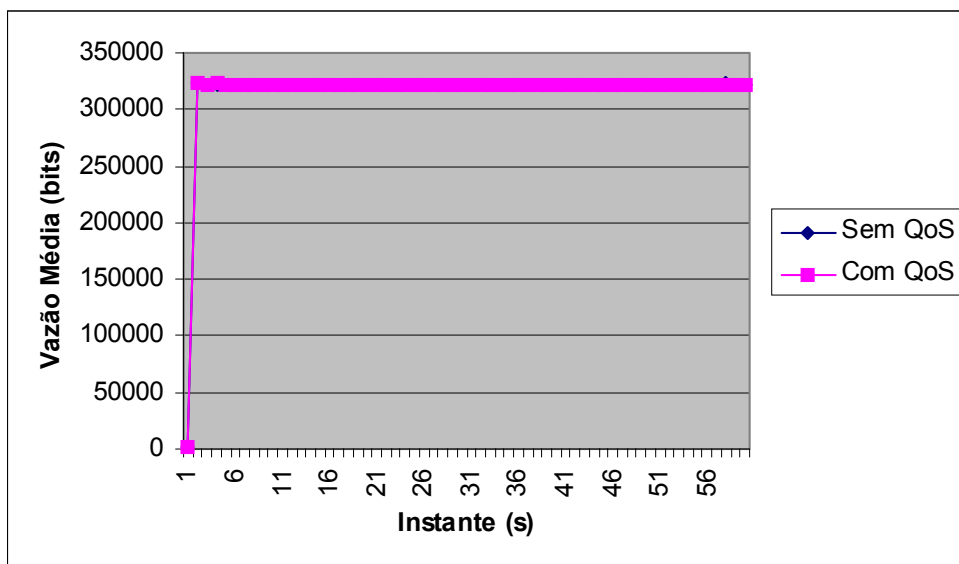


Figura 26 - Vazão com a rede não saturada

Analisando o Gráfico (Figura 26), pode-se observar que a vazão dos pacotes nas duas situações se mantêm iguais, ou seja, as aplicações VoIP com a rede não saturada alcançam sua taxa de transmissão de pacotes independente de o roteador estar ou não

controlando o tráfego. Podendo-se concluir que a rede em estado não saturado é indiferente a presença de um roteador controlando o tráfego (com QoS).

A Tabela 9 mostra maiores informações, como forma de complementar as informações analisadas e visualizadas na Figura 26.

	Sem QoS	Com QoS
Total	19035360	19035360
Mínimo	1440	1440
Máximo	324000	324000
Média	3,17E+05	3,17E+05

Tabela 9 - Estatísticas da Vazão com a rede não saturada

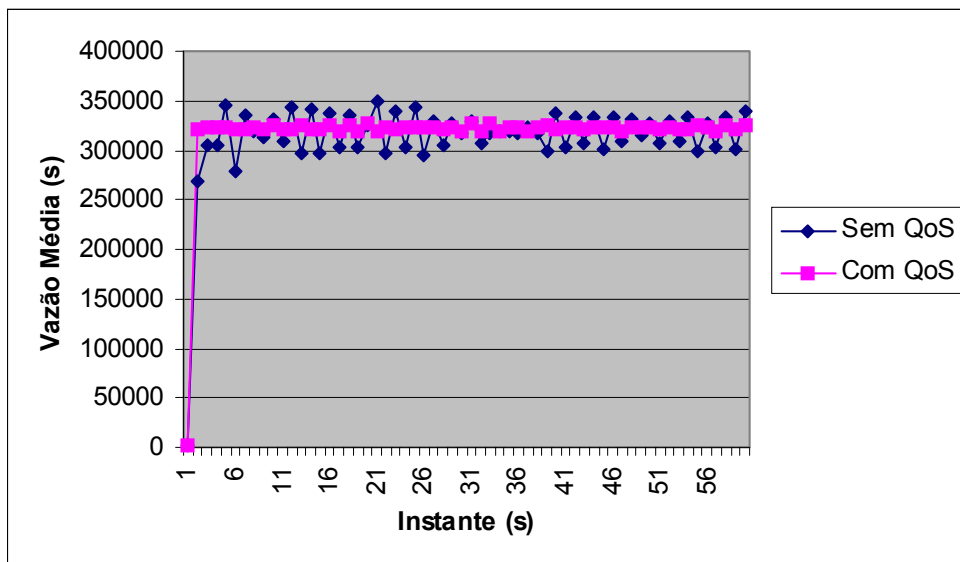


Figura 27 - Vazão com a rede saturada

Analisando o Gráfico (Figura 27), pode-se observar que a vazão dos pacotes manteve-se praticamente constante e com vazão igual a taxa necessária às aplicações VoIP para a rede operando com QoS, enquanto que com rede operando sem QoS a vazão oscilou bastante, o que é prejudicial para o uso de aplicações VoIP, considerando que em vários momentos a vazão é inferior ao necessário para o uso das aplicações VoIP. Podendo-se concluir que a rede em estado saturado é imprescindível para o bom funcionamento em seu tráfego VoIP o uso de um roteador controlando o tráfego (com QoS).

A Tabela 10 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 27.

	Sem QoS	Com QoS
Total	18757440	19035360
Mínimo	1440	1440
Máximo	349920	326880
Média	312624	317256

Tabela 10 - Estatísticas da Vazão com a rede saturada

6.11 Análise da taxa de perdas

A Figura 28 apresenta a perda que os pacotes UDP sofreram durante os testes, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X AltQ/CBQ com a rede não saturada (Figura 28). Para os casos de rede não saturada, não ocorreu a perda de pacotes VoIP, em função disto não apresentamos aqui o gráfico referente a este cenário. A perda ocorreu apenas no caso da rede estar saturada e o roteador não efetuando o controle do tráfego (QoS) momento em que os pacotes UDP (referentes ao tráfego VoIP) concorriam, através do mecanismo FIFO, pela banda disponível com os pacotes TCP (referentes às demais aplicações). Os resultados obtidos neste experimento são apresentados resumidamente na tabela 11.

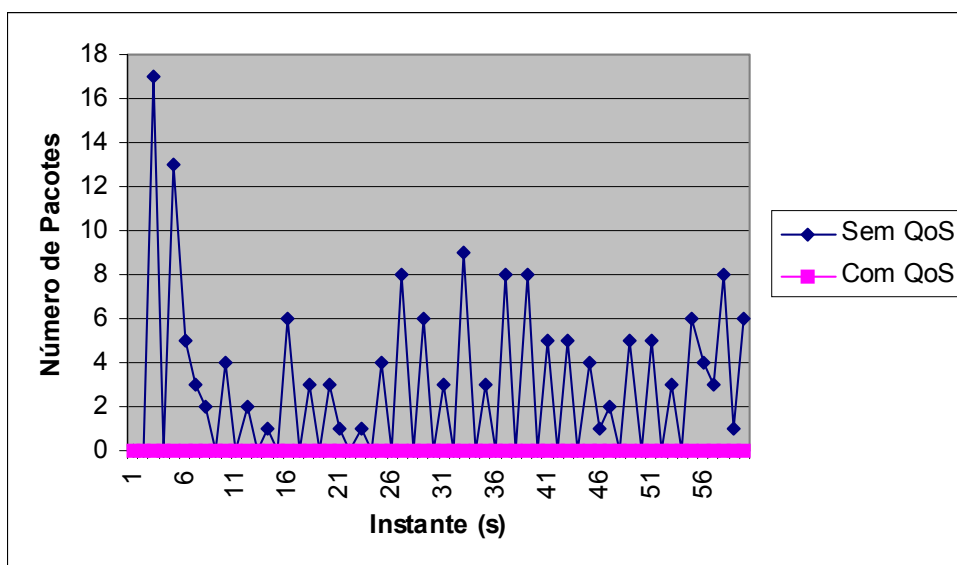


Figura 28 - Perda de Pacotes

Analisando o Gráfico (Figura 28), pode-se observar que a perda dos pacotes ocorridas com rede operando sem QoS inviabiliza o uso de aplicações VoIP. Podendo-se concluir que a rede em estado saturado é imprescindível para o bom funcionamento em seu tráfego VoIP, ou seja, sem perda de pacotes é necessário o uso de um roteador controlando o tráfego (com QoS).

A Tabela 11 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 28.

	Sem QoS	Com QoS
Total	168	0
Mínimo	0	0
Máximo	17	0
Média	2,8	0

Tabela 11 - Estatísticas da Perda com a Rede Saturada e sem QoS

CAPÍTULO 7 - CONCLUSÕES E TRABALHOS FUTUROS

O presente trabalho possibilitou o estudo mais aprofundado na área de redes de computadores, mais especificamente em mecanismos para se possibilitar o controle da Qualidade de Serviço (QoS) em redes, com o objetivo de otimizar ao máximo a largura de banda das empresas e dar preferência para tráfegos que possuem certas exigências quando a restrições em sua transmissão como é o caso do tráfego gerado por aplicações multimídia em geral.

Ao longo dos estudos pesquisou-se desde conceitos básicos, os quais são imprescindíveis para o entendimento global da área em questão, até conceitos mais avançados e direcionados a área de QoS em redes como Mecanismos para Controle de Enfileiramento de Pacotes, Conceitos de QoS e suas respectivas arquiteturas, além dos principais parâmetros de desempenho de uma rede.

Da parte prática foi muito importante para se ter à efetivação do aprendizado, agregando o estudo do Sistema Operacional FreeBSD o qual vem ganhando cada vez mais espaço entre as empresas devido a sua estabilidade, segurança e disponibilidade de aplicações, conhecimento este que poderá ser utilizado perfeitamente no ambiente profissional.

Pode-se concluir também ao fim dos experimentos que o FreeBSD e suas ferramentas de suporte a QoS são bastante eficientes e podem ser utilizadas pelas empresas de forma a reduzir custos com a aquisição de enlaces com maior largura de banda e licenças de softwares proprietários além de evitar que o uso abusivo de determinadas aplicações interfiram no funcionamento das aplicações que necessitam de requisitos mínimos para ter um bom funcionamento.

Com a realização trabalho de conclusão de curso pode-se confirmar a importância desta atividade, a qual completa o conhecimento teórico obtido ao longo de todo o curso, além de servir como uma importante fonte de pesquisa e geração de novos conhecimentos tanto para o aluno quanto para a Instituição, com o conhecimento prático adquirido através das atividades realizadas para o desenvolvimento do projeto.

Como trabalhos futuros sugere-se:

- Repetir os experimentos em um ambiente empresarial;
- Repetir os experimentos utilizando outro sistema Operacional;
- Repetir os experimentos com o AltQ no OpenBSD com o objetivo de comparar a implementação do AltQ neste sistema, o qual tem suporte nativo, com a do FreeBSD.
- Construção de uma Interface Gráfica para o Gerenciamento/Criação das de controle de Banda do IPFW e/ou AltQ de forma a tornar mais acessível a usuários menos especializados;
- Utilizar Ambiente de Alta-Disponibilidade para prover maior confiabilidade do Sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Lu, Guojun. **Communication and Computing for Distributed Multimedia Systems**. Artech House Inc, 1996.
- [2] Melo, Edison Tadeu Lópes. **Qualidade de Serviço em Redes IP com Diffserv: Avaliação através de medições**. Dissertação 2001
- [4] Ferguson, P & Huston, G., **Qualify of Service: Delivering QoS on the Internet and Corporate Networks**. Wiley Computer Publishing. 1999.
- [5] Willrich, Roberto. **Apostila da Disciplina de Sistemas Multimídia**, 2004, Disponível em <http://www.inf.ufsc.br/~willrich/Ensino/INE5639>
- [6] Awduche, D.; Malconm, J. Requirements for Traffic Engineering Over MPLS, Request for Comments 2702. Disponível em <http://www.ietf.org/rfc/rfc2702.txt>
- [7] Silva, Adailton J. S. Qualidade de Serviço em VoIP – Parte I, 2004, Disponível em http://www.rnp.br/newsgen/0005/qos_voip1.html
- [8] E. Rosen, A. Viswanathan, R. callon, Multiprotocol Label Switching Architecture, <http://www.ietf.org/rfc/rfc3031.txt?number=3031>, - Maio 2004.
- [9] Rizzo, Luigi, “DUMMYNET”, Disponível em http://info.iet.unipi.it/~luigi/ip_dummynet/ 2004.
- [10] Simão, Eugênio. SERVIÇOS DIFERENCIADOS EM REDES IP: diferenciação de serviços de rede utilizando um roteador CBQ – Dissertação de Mestrado – UFSC / Março 2002
- [11] Kurose, James F.. Redes de Computadores e a Internet: Uma Nova Abordagem. Pearson Education 2004

- [12] Soares, Luis F. Gomes. Redes de Computadores. Campos 1995
- [13] Cho, Kenjiro. A Framework for Alternate Queueing: Towards Traffic Management by PC-UNIX Based Routers, Artigo encontrado na Internet <http://www.csl.sony.co.jp/person/kjc/kjc/papers/usenix98/altq.html#ALTQ>
- [14] Tracanelli, Patrick. Ipfirewall (IPFW) How-To, Disponível em <http://www.freebsdbrasil.com.br/guia-ipfw.php>
- [15] Schneider, Wolfram. Manual do AltQ, Disponível em <http://www.openbsd.org/cgi-bin/man.cgi?query=altq.conf>
- [16] J.Q. Walker, J.T. Hicks. The Essential Guide to VoIP Implementation and Management. NetIQ.
- [17] L.C. Soares, V.A. Freire. Redes Convergentes. Rio de Janeiro: Altabooks. 2002.
- [18] Black, D et al Na Architecture for Differentiated Services, Rquest for Comments 2475. Disponível em: <http://www.ietf.org/rfc/rfc2475.txt> 1998
- [19] Braden, R.; Clark, D. & Shenker, S. Integrated Services in the Internet Architecture: an Overview. Request for Comments 1633. Disponível em: <http://www.ietf.org/rfc/rfc1633.txt> 1994
- [20] Jones, Rick. “Netperf”, Disponível em <http://www.netperf.org> 2004
- [21] Laine, J., Saaristo, S., and Prior, R. (2002) “Rude & crude”, Disponível em <http://rude.sourceforge.net> 2004.

ANEXO 1 - Roteiro de Instalação e configuração do AltQ

Para instalação

Kernel:

1 – Copiado os fontes do AltQ com o seguinte comando:

```
wget http://www.rofug.ro/projects/freebsd-altq/altq-freebsd-5.2-release-beta2.tar.gz
```

2 – Descompactado o arquivo baixado (diretório /root)

```
tar -zxvpf altq-freebsd-5.2-release-beta2.tar.gz
```

3 – Copia da arvore de diretórios /usr/src/sys para a /usr/src/sys.altq

```
cd /usr/src
```

```
mkdir sys.altq
```

```
cd sys.altq
```

```
(cd ../sys ; tar cf - .) | tar xvf -
```

4 – Aplicar o patch (arquivo sys-altq-current.diff) do AltQ nos fontes do kernel do novo diretório (/usr/src/sys.altq):

```
patch -p0 < <altq_dir>/sys-altq/sys-altq-freebsd-5.2-release.diff
```

(onde <altq_dir> é o diretório onde os fontes do AltQ foram descompactados)

5 - Cópia dos arquivos do AltQ para o diretório /usr/src/sys.altq/

```
cp -R <altq_dir>/sys-altq/altq .
```

(onde <altq_dir> é o diretório onde os fontes do AltQ foram descompactados)

6 - Copiar o arquivo *i386/conf/ALTQ* para um novo Arquivo e acrescentar as demais opções que julgar necessário:

Neste caso foi criado o arquivo *SRV.ALTQ*

E Utilizadas as seguintes opções no *KERNEL* (Além das default):

```
# ALTQ options
options ALTQ # Alternate queueing
options ALTQ_CBQ # Class Based Queueing
options ALTQ_WFQ # Weighted Fair Queueing
options ALTQ_FIFOQ # FIFO queueing
options ALTQ_PRIQ # Priority Queue
options ALTQ_NOPCC # Don't use Processor Cycle Counter
# Enable this for SMP
options PFIL_HOOKS # Packetfilter hooks
options RANDOM_IP_ID # Used by pf, good for security
```

Demais opções que necessitam estar no arquivo de configuração do kernel (verificar a existência delas, caso contrário é só adicionar no arquivo):

```
device bpf
options PFIL_HOOKS
options INET6 ( pf não funciona sem o ipv6 )
options RANDOM_IP_ID (ou make WITH_RANDOM_ID=1)
```

7 Recompilado o Novo Kernel

```
cd /usr/src/sys.altq/i386/conf
/usr/sbin/config SRV.ALTQ
cd ../compile/SRV.ALTQ
make depend
make
make install
```

8 – Finalizando:

- criar um link simbólico /usr/include/altq -> /usr/src/sys.altq/altq

```
ln -s /usr/src/sys.altq/altq /usr/include/altq
```

- Entrar no diretorio usr.sbin

○ `cd <altq_dir>/usr.sbin`

○ `make`

○ `make install`

- Fazer os testes com os arquivos de exemplo dentro do diretório

`<altq_dir>/usr.sbin/altq/altqd/altq.conf.samples/ :`

```
altqd -f <config file>
```

- Você pode monitorar o funcionamento do ALTQ com o comando altqstat:

```
altqstat -w 2
```

ANEXO 2 - Roteiro de Instalação e configuração do Rude & Crude

1 – Atualização do Perl

```
cd /usr/ports/lang/perl5.8  
make  
make install  
use.perl port
```

2 – Instalação do Rude/Crude

```
cd /usr/ports/net/rude  
make  
make install
```

Observação: estes procedimentos foram realizados após a atualização da árvore do ports com o seguinte comando:

```
cvsup -g -L 2 /etc/cvsupfile
```

O arquivo /etc/cvsupfile tem o seguinte conteúdo:

```
*default host=cvsup2.br.FreeBSD.org  
*default base=/usr  
*default prefix=/usr  
*default release=cvs tag=RELENG_5  
*default tag=.  
*default delete use-rel-suffix  
*default compress  
ports-all
```

ANEXO 3 – Regras usadas para simular o tráfego TCP constante

- Regras utilizadas para simular o tráfego constante a 200Kbps entre M3 e M1, estas regras foram colocadas em M3

```
$fwcmd add 2 pipe 1 all from any to 192.168.1.2 out via fxp0
```

```
$fwcmd pipe 1 config bw 200kbit/s
```

```
$fwcmd add 3 pipe 2 all from 192.168.1.2 to any in via fxp0
```

```
$fwcmd pipe 2 config bw 200kbit/s
```

- Regras utilizadas para simular o tráfego constante a 800Kbps entre M3 e M1, estas regras foram colocadas em M3

```
$fwcmd add 2 pipe 1 all from any to 192.168.1.2 out via fxp0
```

```
$fwcmd pipe 1 config bw 800kbit/s
```

```
$fwcmd add 3 pipe 2 all from 192.168.1.2 to any in via fxp0
```

```
$fwcmd pipe 2 config bw 800kbit/s
```

- Sendo que fxp0 é a interface de rede pertencente a M3 e ao enlace L3 e 192.168.1.2 é IP de M1 pertencente ao enlace L2 (vide Figura 22).

ANEXO 4 – Script para a decodificação do tráfego recebido

```
#!/usr/local/bin/bash

# decodifica os logs no diretorio atual, com a extensao .log, e os processa.

FILES=`ls -l $1`
PPATH=$PWD
CRUDE="/home/bunn/experimentacao/crude"

SPATH="/home/bunn/experimentacao/scripts"
DELAY="$SPATH/delay"
DELAYM="$SPATH/delay-medio"
JITTER="$SPATH/jitter"
JITTERM="$SPATH/jitter-medio"
PERDAS="$SPATH/perdas"
PERDAST="$SPATH/perdas-tot"
VAZAO="$SPATH/vazao"

mkdir delay jitter vazao perdas

for i in $FILES; do {
    $CRUDE -d "$1/$i" > "$PPATH/$i.tmp"
    cat "$i.tmp" | sed "1 D" > "$i.log"
} done

rm -f "$PPATH/*.tmp"
FILES=`ls -l *.log`

for i in $FILES; do {
    cd "$SPATH/vazao"
    $VAZAO "$SPATH/$i" > "v-$i"
    cd "$SPATH/delay"
    $DELAY "$SPATH/$i" > "d-$i"
    $DELAYM "$SPATH/$i" > "dm-$i"
}
```

```
cd "$PPATH/jitter"  
$JITTER "$PPATH/$i" > "j-$i"  
$JITTERM "$PPATH/$i" > "jm-$i"  
cd "$PPATH/perdas"  
$PERDAS "$PPATH/$i" > "p-$i"  
$PERDAST "../$i" >> "p-tot"  
} done
```

ANEXO 5 – Sintaxe dos programas utilizados

1 – Rude

```
rude -s rude.conf
```

Onde o conteúdo do arquivo rude.conf é:

```
START NOW
```

```
#Simulando 4 ligações simultâneas
```

```
1000 0010 ON 3001 192.168.1.2:5001 CONSTANT 224 180
```

```
61000 0010 OFF
```

2 – Crude

```
crude -p 5001 -l arquivo.txt
```

3 – Netperf - server

```
netserver
```

4 – Netperf – para gerar tráfego

```
netperf -H 192.168.1.2 -l 120
```


Avaliação do Serviço de QoS no FreeBSD

Roberto Willrich¹, Mário A. R. Dantas¹, Vitório B. Mazzola¹, Alexandre Bunn¹

¹INE – Departamento de Informática e Estatística da UFSC

Caixa Postal 476 – 88.040-900 – Florianópolis – SC – Brasil

{willrich, mario, mazzola, albunn}@inf.ufsc.br

RESUMO: A Internet, desde seu início, utiliza-se de um mecanismo de escalonamento de pacotes do tipo FIFO (*First In First Out*) também chamado de modelo Melhor Esforço. As soluções de telefonia e videoconferência usando as redes IP, como também as aplicações científicas de banda larga têm uma demanda cada vez maior. Como os fluxos de áudio e vídeo correspondem a serviços em tempo real, eles exigem prazos de transporte muito curtos e uma transferência que apresente poucas variações, isto é, regular. As aplicações tradicionais (web, correio eletrônico, transferência de arquivos) admitem intervalos de tempo maiores, mas por outro lado necessitam de uma baixa taxa de perda de pacotes.

O objetivo deste trabalho é avaliar, com base em algumas métricas de desempenho (*delay*, *jitter*, vazão e perda de pacotes), o serviço de QoS oferecido por Roteadores baseados em FreeBSD. Com base nos resultados obtidos pode-se concluir que o FreeBSD juntamente com suas ferramentas para suporte a QoS em redes atendem os requisitos das empresas em definir políticas para a priorização de pacotes a serem transmitidos pelo seu enlace de saída o qual geralmente tem uma largura de banda limitada se comparada a largura de banda da rede local.

1 INTRODUÇÃO

O uso das aplicações em rede tem sido imprescindível na realização de tarefas em vários ambientes de trabalho. Atualmente utilizamos a internet para realizar inúmeras atividades, tais como: atividades bancárias, compras, pesquisas, troca de informações (confidenciais ou não), mensagens de e-mail, reuniões através de videoconferência e chat, dentre outras. Outro papel que a Internet vem ganhando é no campo do ensino onde pessoas podem se aperfeiçoar através de aplicações de *E-learning* e portais de informações onde profissionais renomados disponibilizam artigos das mais diversas áreas e acessíveis de qualquer lugar com acesso a Internet.

A rede Internet oferece basicamente um único tipo de serviço de comunicação dito do tipo "melhor esforço" (Best-effort Service), próprio do protocolo IP, onde não é provido nenhum mecanismo que garanta certos níveis de qualidade de serviços (QoS) e portanto não há limites em termos de parâmetros de desempenho de redes, do tipo taxa de bits, atraso fim-a-fim, variação de atraso e taxa de perda de pacotes. QoS pode ser definida como a especificação qualitativa e quantitativa dos requisitos de uma aplicação que um sistema deveria satisfazer a fim de obter a qualidade desejada [1].

As soluções de telefonia e videoconferência usando as redes IP, como também as aplicações científicas de banda larga têm uma demanda cada vez maior. Como os fluxos de áudio e vídeo correspondem a serviços em tempo real, eles exigem prazos de transporte muito curtos e uma transferência que apresente poucas variações, isto é, regular. As aplicações tradicionais (web, correio eletrônico, transferência de arquivos) admitem intervalos de tempo maiores, mas por outro lado necessitam de uma baixa taxa de perda de pacotes [4].

Para enfrentar as novas restrições e fornecer uma qualidade de serviço adequada, os engenheiros podem utilizar o superdimensionamento da rede. Porém, esta é uma solução em curto prazo, pois o tráfego continua aumentando e tende a ocupar rapidamente toda a largura de banda disponível, além do que a aquisição de links de comunicação com alta largura de banda é bastante caro.

Enfim, a expansão e a interconexão das redes tornam cada vez mais complexo o gerenciamento das tabelas de roteamento e torna cada vez mais necessário se ter mecanismos para priorizar o tráfego. Esta priorização permite dar preferência para tráfegos que necessitam de rigorosos parâmetros de qualidade, como nas aplicações de tempo-real, postergando o tráfego de pacotes que toleram maiores tempos de atraso, por exemplo.

Para que a rede suporte gerenciamento de QoS, os roteadores devem suportar os serviços necessários, e são estes, os responsáveis pelo encaminhamento de pacotes no nível de rede. Atualmente, a implementação das arquiteturas DiffServ [7] e IntServ [8] se torna demasiadamente caras, pelo fato de que o controle da maior parte dos roteadores intermediários está nas mãos das empresas de telecomunicações, as quais cobram por cada serviço requisitado para que se possa ter alguma garantia de qualidade sendo esta diretamente proporcional ao preço do SLA (*Service Level Agreement*) efetuado.

Diante desta situação, pequenas e médias empresas optam por controlar o tráfego apenas em seus próprios roteadores, dando preferência a tráfegos considerados críticos e deixando em segundo plano os demais tráfegos. Esta estratégia é adotada de forma a dar uma maior dinâmica ao fluxo de dados e otimizando os recursos de rede da empresa de forma a reduzir os custos com a aquisição de um link de dados com maior largura de banda, por

exemplo. Neste contexto, é que surgem as ferramentas para diferenciação de tráfego, incluindo a classificação de pacotes, políticas de escalonamento de filas de encaminhamento nos roteadores e de descarte de pacotes quando estas filas estiverem cheias.

Este artigo tem por objetivo avaliar o uso de roteadores baseados em FreeBSD para realização de diferenciação de tráfego. Roteadores FreeBSD são computadores com S.O e que realizam as tarefas de roteamento, e no nosso caso, incluindo os serviços de diferenciação de tráfego através da utilização do Software AltQ [5] [6] instalado e configurado no Roteador em questão

O restante deste artigo está organizado na forma que segue. A seção 2 descreve as características gerais do FreeBSD e do AltQ. Em seguida, a seção 3 apresenta a Avaliação do Serviço de QoS no FreeBSD e a metodologia de testes. Por fim, a seção 4 apresenta as conclusões deste trabalho.

2 FREEBSD

O FreeBSD é um sistema operacional UNIX Multitarefa de 32 bits gratuito, para computadores pessoais baseados na tecnologia INTEL e compatíveis (Arquitetura PC). É um sistema operacional completo, robusto e distribuído com o seu código fonte, o que faz dele um excelente sistema operacional para estudantes de Engenharia ou Ciências da Computação.

O robusto suporte TCP/IP do FreeBSD faz dele um sistema ideal e de baixo custo para aplicações Inter/Intranet. Veja alguns dos serviços que ele permite implementar: Servidor WWW; Servidor FTP; Servidor Proxy; Servidor Telnet; Servidor IRC; Servidor DNS; Servidor de Mail (POP3 / SMTP); Servidor PPP; Roteador para LAN / WAN; Firewall; Servidor para jogos multiplayer via Internet (QUAKE , etc); e vários outros tipos de aplicações (Bancos de Dados, VPN, e assim por diante).

2.1 QoS no FreeBSD

O FreeBSD quando utilizado na função de Roteador de uma rede utiliza como mecanismo de controle e enfileiramento de pacotes o padrão oferecido pelo protocolo IP o FIFO, ou seja, fornece um serviço de melhor esforço para todos os datagramas que transporta. Em outras palavras, o roteador vai tentar entregar os datagramas que recebe o mais rápido possível. Contudo o serviço de melhor esforço não faz nenhuma promessa sobre o atraso fim

a fim para um pacote individual. Tampouco faz promessa quanto à variação do atraso de pacote dentro de uma corrente de pacotes.

O maior problema para as aplicações multimídia é quando um dos enlaces (LAN ou WAN) do Roteador está congestionado, neste caso os pacotes começam a ser enfileirados utilizando-se o mecanismo FIFO o qual não disponibiliza de nenhum mecanismo para controle da qualidade de serviço conforme visto anteriormente. É neste momento que entram as aplicações para suporte a QoS no FreeBSD onde com base na instalação e sua configuração pode-se mudar a dinâmica do roteamento dos pacotes dando-se, por exemplo, prioridade de passagem para pacotes de aplicações com maiores exigências como as aplicações multimídia e deixando em “segundo plano” as aplicações que utilizam o protocolo TCP.

No presente artigo pesquisou-se dois principais softwares para tal finalidade o Dummynet [3] e o AltQ [5] sendo que exploraremos apenas o AltQ devido ao seu maior número de opções e aos mecanismos de Escalonamento de Pacotes implementados.

2.2 AltQ (Alternate Queueing)

O AltQ não é apenas uma ferramenta para possibilitar mecanismos de QoS, é também um framework que permite gerenciar as disciplinas de fila de pacotes na rede [5]. O AltQ introduz novas funções para a manipulação das filas de pacotes em um roteador ou em uma estação de trabalho, permitindo que se escolha como mecanismos de Enfileiramento de pacotes o: PQ, WFQ, CBQ, dentre outros além de algoritmos de controle de congestionamento como: RED, RIO, BLUE e o CDNR.

Para a experimentação prática proposta no presente relatório optou-se por utilizar o AltQ devido à possibilidade de utilização de outros mecanismos de controle da fila de pacotes, ao contrário do Dummynet que permitia apenas a implementação de Filas WF2Q+.

A lógica de sua utilização consiste na criação de classes com determinada largura de banda disponível e a criação de filtros (tipos de tráfego) com determinada porcentagem de tráfego relativo à classe a que pertence, podendo também atribuir a uma classe atributos como prioridade, algoritmo de controle de congestionamento, seqüência de filtragem, dentre outros. O AltQ é uma solução para fornecimento de QoS e trabalha independente do sistema de firewall, ao contrário do Dummynet que permite apenas o uso do IPFW, o AltQ trabalha com o PF, IPFW, e qualquer outro sistema de firewall existente na plataforma em questão. Para utiliza-lo basta criar um arquivo de configuração, ou seja, um arquivo texto qualquer com as regras em seu conteúdo.

A configuração do AltQ basicamente consiste na definição das classes de tráfego (utilizada para os algoritmos CBQ, PQ e HFSC), onde neste caso será abordado apenas os comandos relativos a filas controladas pelo algoritmo CBQ, informações sobre a utilização dos demais algoritmos consultar [6].

3 AVALIAÇÃO DO SERVIÇO DE QOS NO FREEBSD

Este item apresenta uma análise de desempenho de encaminhamento de datagramas IP em uma rede provida de mecanismos de QoS e sem QoS considerando o uso de roteadores FreeBSD em uma rede local. Para isso, o ambiente de testes será usado em dois momentos distintos: primeiro momento utilizando o roteamento IP clássico; e segundo momento usando roteamento com QoS, ou seja, o roteador irá controlar o fluxo de pacotes no enlace L2 priorizando os pacotes de aplicações VoIP e controlando o fluxo das demais aplicações de forma a não interferirem no uso das aplicações de VoIP mais sensíveis a variações de tráfego na rede. Nestes dois momentos, serão realizados 2 testes, o primeiro enlace não-saturado e o outro com o enlace saturado, visando medir o desempenho e os resultados são confrontados. A duração de cada teste será de 60s.

Esta análise visa comparar, em situações de tráfego igual e trafegando pelos mesmos nós com as mesmas condições, é revelar se um Roteador FreeBSD é adequado para efetuar o do Controle de Banda de forma a priorizar o tráfego VoIP de uma empresa.

3.1 Definição do Ambiente de Testes

O primeiro passo para a realização da avaliação de desempenho pretendida foi definir a topologia de rede mais adequada ao nosso propósito. O principal requisito dessa topologia é dispensar o uso de mecanismos de sincronização de relógio convencionais como, NTP de forma que as estatísticas geradas no experimento não sejam influenciadas por este tipo de problema e também a simulação de um link de saída com 1Mbps (Enlace L2 conforme podemos observar na Figura 1) alcançado através do uso do AltQ primeiramente controlando o tráfego e utilizando o mecanismo de enfileiramento de pacotes FIFO e no segundo teste além de controlar o tráfego utilizou-se o mecanismo de enfileiramento CBQ, devido à possibilidade da criação de regras que otimizem o uso da banda ao máximo.

A topologia criada está disposta conforme a figura 1. O ambiente de testes definido tem 1 roteador e 2 computadores sendo um deles para gerar e o outro para gerar e receber tráfego, todos com Sistema Operacional FreeBSD.

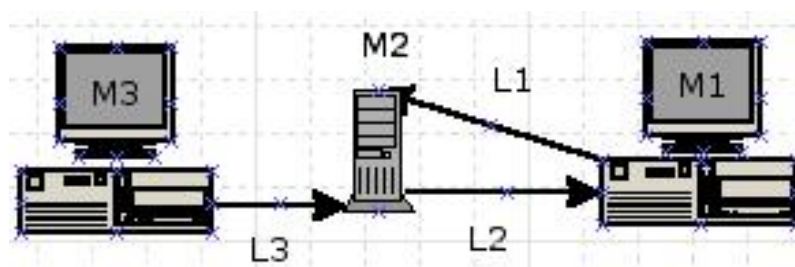


Figura 29 - Plataforma de Teste

A máquina 1 (M1) é geradora/receptora de tráfego, simulando neste caso o uso de aplicações VoIP, para o *testbed*. O roteador 2 (M2) efetua além do roteamento o condicionamento de tráfego e recebe tráfego de fundo vindo da máquina 3, tráfego TCP simulando o uso de aplicações HTTP, FTP, POP e SMTP, por exemplo.

3.2 Características do Equipamento Roteador

Definida a topologia, passa-se para a etapa de montagem do ambiente de teste. Como pressuposto inicial, optou-se pelo uso de um roteador implementado na forma de computador pessoal usando FreeBSD e a tecnologia de rede Ethernet. No ambiente de teste implantado, o roteador FreeBSD tem a versão 4.9 RELEASE e possui o kernel configurado para ter suporte ao AltQ 3.0.

O roteador FreeBSD está provido de placas de rede Intel 82559 10/100. Um fator que merece destaque para análise de desempenho em uma rede é a utilização de placas de rede bem projetadas e de boa qualidade. Em um estágio inicial foram utilizadas placas de redes de baixo custo, onde se deparou com vários problemas na realização dos testes. Um dos principais problemas detectados é a ocorrência de um valor excessivo de perda de pacotes mesmo com uma taxa de transmissão abaixo do limite da capacidade dos enlaces. Esse problema foi resolvido a partir do momento em que se utilizaram placas de rede de melhor qualidade, permitindo realizar avaliações que estavam de acordo com a configuração da ferramenta geradora de tráfego.

3.3 Métricas utilizadas na avaliação dos testes

Para a avaliação do desempenho nesse projeto, foram adotadas as seguintes métricas [2]:

- **Atraso por pacote:** É o intervalo temporal entre a transmissão do pacote pela fonte e a recepção do mesmo no destino. Um dos grandes responsáveis por este atraso é o processamento nos roteadores.
- **Varição no atraso (*jitter*):** É a variação do atraso sofrido por dois pacotes subjacentes. Caso tenha um valor baixo, ele indica que os pacotes estão sofrendo praticamente o mesmo atraso de processamento nos roteadores. Por meio dessa métrica, consegue-se analisar quão constante é o roteamento de pacotes processado nos roteadores.
- **Vazão:** Indica a taxa de transmissão obtida pela fonte dos pacotes.
- **Perdas:** Indica o percentual de pacotes que foram descartados por fluxo numa transmissão de dados.

3.4 Escolha de ferramentas para realizar os testes de desempenho

Para realização das medições, foram avaliadas várias ferramentas para geração e análise de tráfego, dentre elas estão a Netperf [9] e RUDE & CRUDE [10].

O Netperf (*Network Performance*) utiliza um esquema de gerador e receptor de tráfego. Onde na máquina geradora (M3) chama-se um arquivo binário passado como parâmetros às características dos pacotes a serem enviados como duração da transmissão e tamanho dos pacotes e na máquina receptora (M1) roda-se apenas um processo *Daemon*, que fica rodando e escutando em uma porta TCP aguardando o envio de tráfego por uma outra máquina.

A ferramenta RUDE & CRUDE utiliza um esquema gerador e receptor de tráfego. O gerador gera um tráfego UDP configurável para um receptor especificado. O receptor armazena os pacotes UDP sem decodificá-los, de forma a não gerar carga de processamento na máquina receptora, em arquivo para posterior cálculo de desempenho. Pó ser unidirecional, ela não necessita de uma comunicação *full-duplex*. Entretanto, seu uso implica na necessidade

de sincronização de relógios do transmissor e do receptor. O RUDE & CRUDE deixa a decodificação do tráfego de rede para depois, consistindo nesta característica sua principal vantagem perante as demais ferramentas, dos experimentos sem influenciar na avaliação do tráfego. Desta forma, o RUDE & CRUDE foi o escolhido para realizar as medições no ambiente de testes. Outra vantagem desta ferramenta é a análise de tráfego por pacote, podendo medir informações de atraso por pacote, quesito mais importante no contexto de velocidade de encaminhamento.

Para a sincronização do relógio, efetuou-se a configuração da máquina 1 (M1) para que não efetuasse o *Loopback* ao enviar tráfego da interface de rede pertencente ao enlace L1 para a interface de rede pertencente ao enlace L2 passando pelo roteador 1 (M2), onde será efetuado o controle do tráfego.

3.5 Definição do Tráfego de Teste

Como já foi citado, os testes tiveram duas etapas: uso apenas do roteamento clássico IP e o condicionamento de pacotes utilizando o mecanismo de Enfileiramento de Pacotes CBQ. Para ambos os testes, a máquina Receptora 1 foi o destino dos dados gerados pelas máquinas Geradoras 1 e 2 (figura 1). Os dados gerados foram do tipo CBR (Constant Bit Rate) adotando o protocolo de transporte UDP.

O caso avaliado configurou-se a máquina M1 para gerar tráfego UDP, simulando o uso de aplicações VoIP neste caso o tráfego gerado é de 320Kbps (4 ligações simultâneas com CODEC G.711 sem supressão de silêncio, neste caso gerando uma taxa constante (CBR) de 80kbps para cada ligação) e M3 para gerar tráfego TCP, pacotes com 1024Bytes.

Em todos os testes apresentados neste relatório, tanto o tráfego de teste como o tráfego de fundo foram CBR (Constant Bit Rate) usando o protocolo UDP (tráfego enviado por M1) e TCP (tráfego enviado por M3). Cabe aqui ressaltar que o tráfego CBR não é o tráfego majoritário na Internet. Em geral, na Internet o tráfego é VBR (Variable Bit Rate) usando TCP. O tráfego tipo CBR foi adotado para facilitar a análise. Devido ao fato de não se ter uma ferramenta que gerasse um tráfego TCP constante a uma determinada taxa, foi utilizado o seguinte mecanismo que M3 pudesse gerar o tráfego em 2 níveis 200Kbps e 800Kbps respectivamente. Para o primeiro caso criou-se um *pipe*, através do uso do

DUMMYNET, de 200Kbps com isso obteve-se um fluxo constante de 200Kbps e para o segundo caso realizou-se o mesmo procedimento só que com um *pipe* de 800Kbps.

Foram feitos 2 testes sendo o primeiro teste com o roteador IP normal e o segundo com o Roteador efetuando o controle do tráfego e enfileiramento dos pacotes seguindo o mecanismo CBQ e cada teste com dois momentos, o primeiro momento com o enlace não-saturado (onde M3 gerava um tráfego de 200Kbps) e segundo momento com o enlace saturado (onde M3 gerava um tráfego do 800Kbps).

3.6 Diferenciação de Tráfego

Para efetuar os testes se teve que pensar em uma solução para que o ambiente de simulação ficasse o mais próximo possível do descrito, ou seja, utilizando-se a tecnologia Ethernet teríamos que simular o link de uma empresa com 1Mbps (Enlace L2) de *Upsteam*, durante o teste sem QoS. Para isso efetuou-se a configuração do Roteador para operar com o Atq usando como mecanismo de enfileiramento de pacotes o FIFO. A configuração do roteador, no caso, do AltQ pode ser observada abaixo (criou-se o arquivo de configuração com o conteúdo abaixo, neste caso chamado de `altq.conf.fifo`):

```
#Simula um link com upstream de 1Mbps usando FIFO
```

```
interface fxp2 bandwidth 1M fifoq
```

Para os testes em que o enlace L2 operava com controle de tráfego, com o mecanismo de enfileiramento de pacotes CBQ, utilizou-se a seguinte configuração (criou-se outro arquivo de configuração com o conteúdo abaixo, neste caso chamado de `altq.conf`):

```
#Simula um link com upstream de 1Mbps
```

```
interface fxp2 bandwidth 1M cbq
```

```
class cbq fxp2 root NULL pbandwidth 100
```

```
# meta classe
```

```
class cbq fxp2 def_class root borrow pbandwidth 100 default
```

```
# Tráfego VoIP
class cbq fxp2 voip def_class priority 7 borrow pbandwidth 40 red
filter fxp2 voip 0 5001 0 0 17
filter fxp2 voip 0 0 0 5001 17

# Tráfego TCP
class cbq fxp2 tcp_class def_class priority 1 borrow pbandwidth 55
filter fxp2 tcp_class 0 0 0 0 6
```

3.7 Resultados dos testes experimentais de desempenho IP x QoS

Esta seção apresenta os resultados da análise comparativa do uso das tecnologias IP convencional (FIFO) e IP/AltQ alterando o mecanismo de enfileiramento de pacotes e efetuando o condicionamento de tráfego em duas situações de rede: uma a rede não saturada e a outra com a rede saturada. As medidas apresentadas neste relatório são referentes ao tráfego gerado pela máquina Geradora 1 (tráfego VoIP). As medidas foram obtidas pela máquina 1 e decodificadas utilizando-se o próprio CRUDE.

3.8 Análise do Atraso fim-a-fim (*delay*)

As Figuras 2 e 3 apresentam o atraso sofrido pelos pacotes UDP, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X AltQ/CBQ com a rede não saturada (Figura 2) e com a rede saturada (Figura 3). Os resultados obtidos neste experimento são apresentados resumidamente nas tabelas 1 e 2.

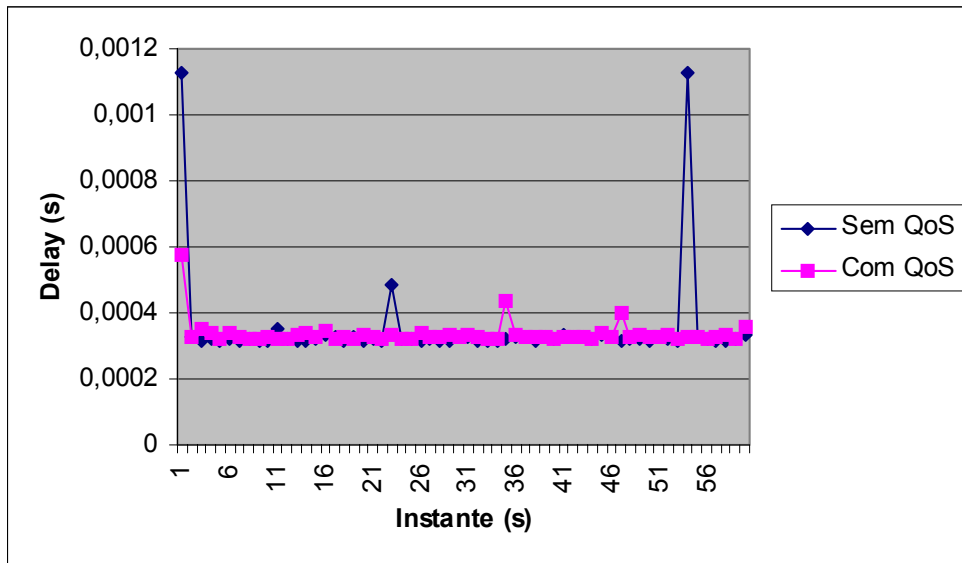


Figura 30 - Delay rede não saturada

Analisando o Gráfico (Figura 2), pode-se observar que o atraso dos pacotes (*delay*) nas duas situações se mantêm praticamente iguais sendo que com QoS ficou com poucas oscilações, porém inferiores se comparado à rede operando sem QoS. Podendo-se concluir que a rede em estado não saturado possui uma melhor estabilidade em seu tráfego quando o roteador opera controlando o tráfego (com QoS).

A Tabela 1 mostra maiores informações, como forma de complementar as informações analisadas e visualizadas na Figura 2.

	Sem QoS	Com QoS
Total	0,021069	0,020144
Mínimo	0,000313	0,00032
Máximo	0,001127	0,000577
Média	0,000351	0,000336

Tabela 12 - Estatísticas do Delay com a rede não saturada

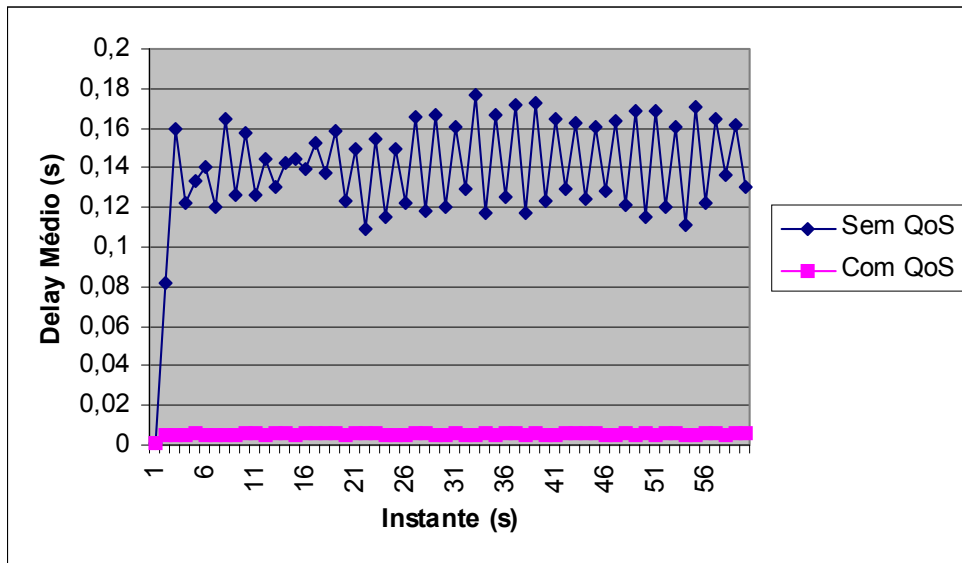


Figura 31 - Delay com a rede Saturada

Analisando o Gráfico (Figura 3), pode-se observar que o atraso dos pacotes (*delay*) manteve-se praticamente constante e com valores bastante inferiores para a rede operando com QoS, enquanto que com rede operando sem QoS o delay além de ser alto, o que inviabilizaria o uso de aplicações VoIP durante este período, oscilou bastante durante o período de teste. Podendo-se concluir que a rede em estado saturado é imprescindível para uma melhor estabilidade em seu tráfego VoIP o uso de um roteador controlando o tráfego (com QoS).

A Tabela 2 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 3.

	Sem QoS	Com QoS
Total	8,321908	0,330191
Mínimo	0,00129	0,001045
Máximo	0,176748	0,006359
Média	0,138698	0,005503

Tabela 13 - Estatísticas do Delay com a rede Saturada

Concluindo o parâmetro atraso fim-a-fim, constatou-se que, no ambiente de teste definido, o QoS teve um ótimo desempenho, tornando, mesmo em situações em que a rede esteja saturada, viável o uso de aplicações VoIP.

3.9 Análise da Variação de Atraso (*Jitter*)

As Figuras 4 e 5 apresentam a variação de atraso (*jitter*) sofrido pelos pacotes UDP, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X AltQ/CBQ com a rede não saturada (Figura 4) e com a rede saturada (Figura 5). Os resultados obtidos neste experimento são apresentados resumidamente nas tabelas 3 e 4.

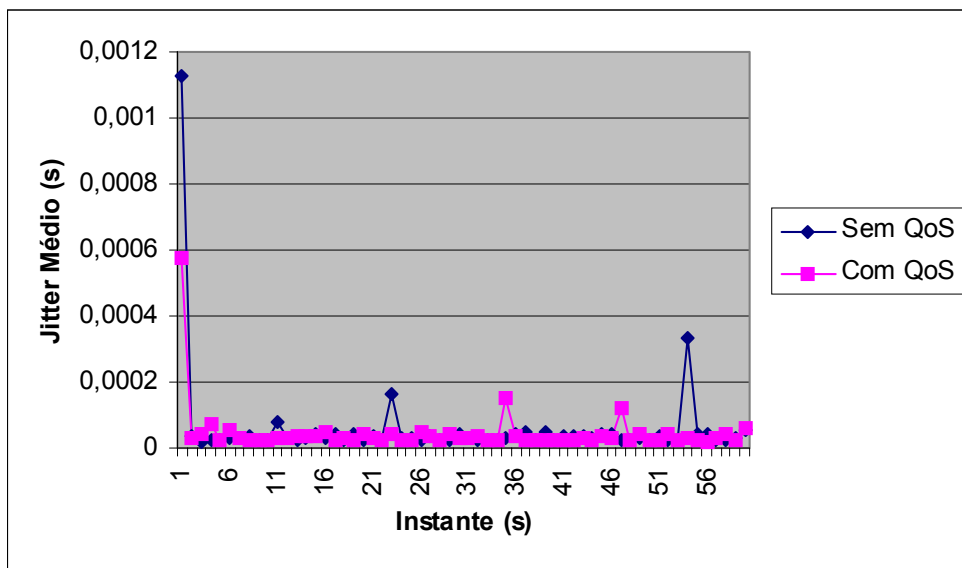


Figura 32 - Jitter com a rede não saturada

Analisando o Gráfico (Figura 4), pode-se observar que a variação do atraso dos pacotes (*jitter*) nas duas situações se mantêm praticamente iguais sendo que com QoS ficou com poucas oscilações, porém inferiores se comparado à rede operando sem QoS. Podendo-se concluir que a rede em estado não saturado possui uma melhor estabilidade em seu tráfego quando o roteador opera controlando o tráfego (com QoS).

A Tabela 3 mostra maiores informações, como forma de complementar as informações analisadas e visualizadas na Figura 4.

	Sem QoS	Com QoS
Total	0,003464	0,002665
Mínimo	0,00002	0,000018
Máximo	0,001125	0,000577
Média	5,77E-05	4,44E-05

Tabela 14 - Estatísticas do Jitter com a rede não saturada

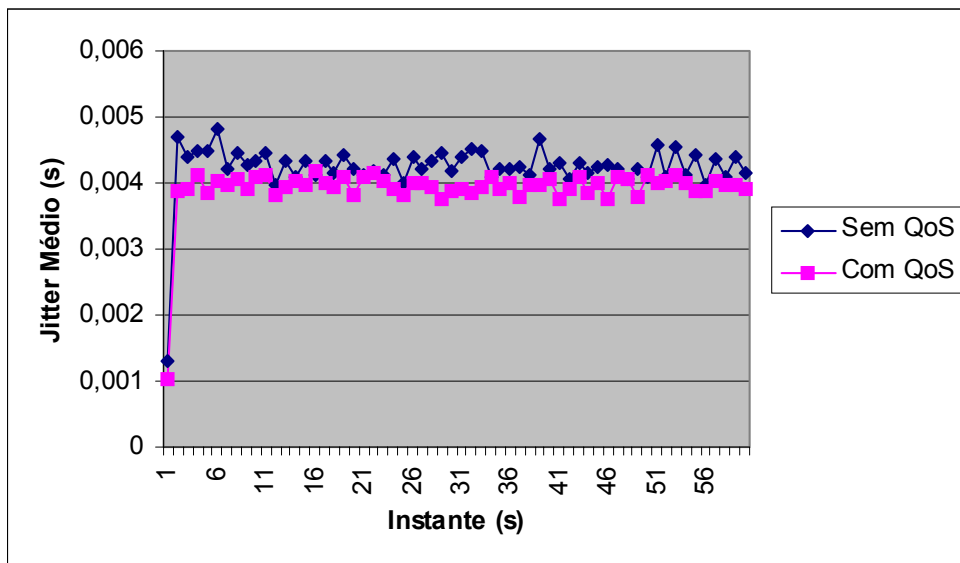


Figura 33 - Jitter com a rede saturada

Analisando o Gráfico (Figura 5), pode-se observar que a variação do atraso dos pacotes (*jitter*) manteve-se praticamente constante e com valores ligeiramente inferiores para a rede operando com QoS, enquanto que com rede operando sem QoS o *jitter* oscilou bastante e com valores sempre superiores a rede operando com QoS, o que é prejudicial para o uso de aplicações VoIP. Podendo-se concluir que a rede em estado saturado é imprescindível para uma melhor estabilidade em seu tráfego VoIP o uso de um roteador controlando o tráfego (com QoS).

A Tabela 4 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 5.

	Sem QoS	Com QoS
Total	0,254055	0,235085
Mínimo	0,00129	0,001045
Máximo	0,004807	0,004173
Média	4,23E-03	3,92E-03

Tabela 15 - Estatísticas do Jitter com a rede Saturada

3.10 Análise da vazão

As Figuras 6 e 7 apresentam a vazão que os pacotes UDP obtiveram durante os testes, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X

AltQ/CBQ com a rede não saturada (Figura 7) e com a rede saturada (Figura 8). Os resultados obtidos neste experimento são apresentados resumidamente nas tabelas 5 e 6.

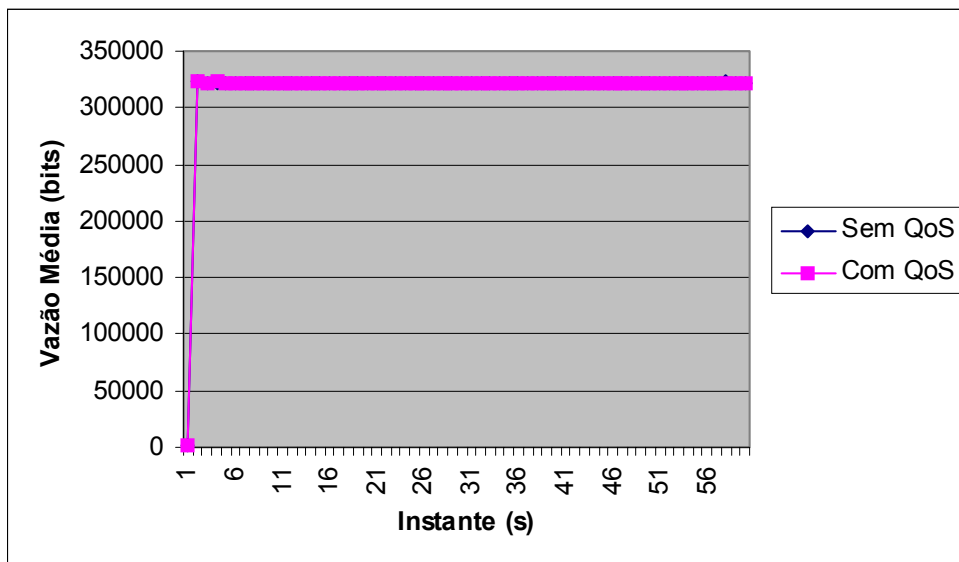


Figura 34 - Vazão com a rede não saturada

Analisando o Gráfico (Figura 6), pode-se observar que a vazão dos pacotes nas duas situações se mantêm iguais, ou seja, as aplicações VoIP com a rede não saturada alcançam sua taxa de transmissão de pacotes independente de o roteador estar ou não controlando o tráfego. Podendo-se concluir que a rede em estado não saturado é indiferente a presença de um roteador controlando o tráfego (com QoS).

A Tabela 5 mostra maiores informações, como forma de complementar as informações analisadas e visualizadas na Figura 6.

	Sem QoS	Com QoS
Total	19035360	19035360
Mínimo	1440	1440
Máximo	324000	324000
Média	3,17E+05	3,17E+05

Tabela 16 - Estatísticas da Vazão com a rede não saturada

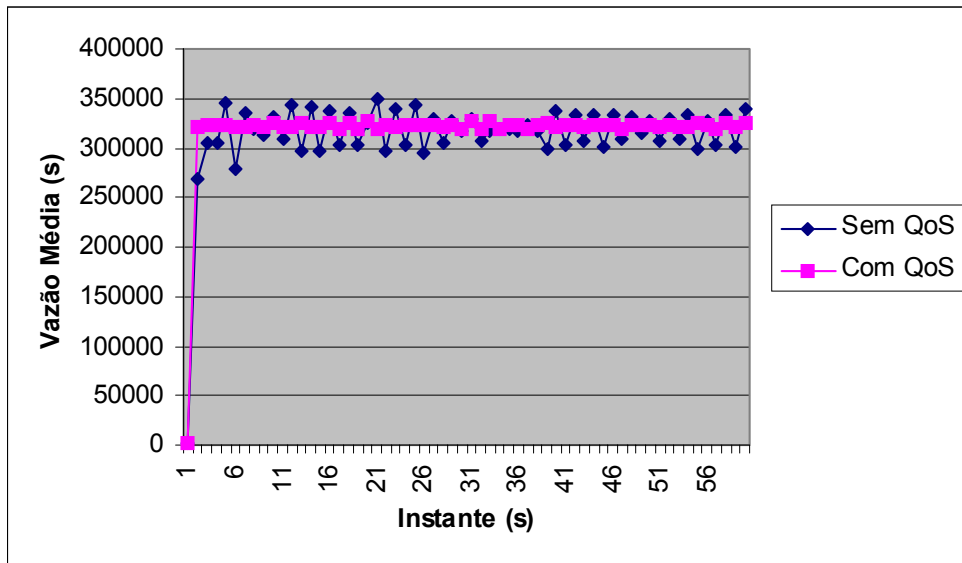


Figura 35 - Vazão com a rede saturada

Analisando o Gráfico (Figura 7), pode-se observar que a vazão dos pacotes manteve-se praticamente constante e com vazão igual a taxa necessária às aplicações VoIP para a rede operando com QoS, enquanto que com rede operando sem QoS a vazão oscilou bastante, o que é prejudicial para o uso de aplicações VoIP, considerando que em vários momentos a vazão é inferior ao necessário para o uso das aplicações VoIP. Podendo-se concluir que a rede em estado saturado é imprescindível para o bom funcionamento em seu tráfego VoIP o uso de um roteador controlando o tráfego (com QoS).

A Tabela 6 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 7.

	Sem QoS	Com QoS
Total	18757440	19035360
Mínimo	1440	1440
Máximo	349920	326880
Média	312624	317256

Tabela 17 - Estatísticas da Vazão com a rede saturada

3.11 Análise da taxa de perdas

A Figura 8 apresenta a perda que os pacotes UDP sofreram durante os testes, quando utilizado o mecanismo de enfileiramento de pacotes convencional (FIFO) X

AltQ/CBQ com a rede não saturada (Figura 8). Para os casos de rede não saturada, não ocorreu a perda de pacotes VoIP, em função disto não apresentamos aqui o gráfico referente a este cenário. A perda ocorreu apenas no caso da rede estar saturada e o roteador não efetuando o controle do tráfego (QoS) momento em que os pacotes UDP (referentes ao tráfego VoIP) concorriam, através do mecanismo FIFO, pela banda disponível com os pacotes TCP (referentes às demais aplicações). Os resultados obtidos neste experimento são apresentados resumidamente na tabela 7.

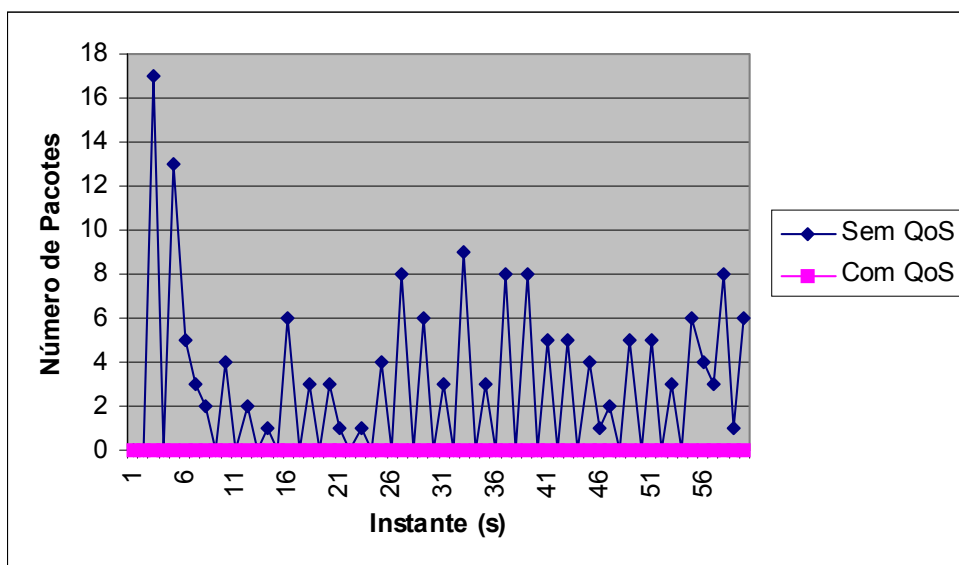


Figura 36 - Perda de Pacotes

Analisando o Gráfico (Figura 8), pode-se observar que a perda dos pacotes ocorridas com rede operando sem QoS inviabiliza o uso de aplicações VoIP. Podendo-se concluir que a rede em estado saturado é imprescindível para o bom funcionamento em seu tráfego VoIP, ou seja, sem perda de pacotes é necessário o uso de um roteador controlando o tráfego (com QoS).

A Tabela 7 mostra maiores informações, como forma de complementar e confirmar as conclusões alcançadas com a análise da Figura 8.

	Sem QoS	Com QoS
Total	168	0
Mínimo	0	0
Máximo	17	0
Média	2,8	0

Tabela 18 - Estatísticas da Perda com a Rede Saturada e sem QoS

4 CONCLUSÕES E TRABALHOS FUTUROS

Esse artigo apresentou uma análise prática quanto ao impacto na eficiência do uso do AltQ para prover serviços de Diferenciação de Tráfego em test-beds FreeBSD usando enlaces Ethernet. Neste trabalho, procurou-se adaptar o ambiente de teste para ser o mais próximo possível de um ambiente real de uma pequena ou média empresa com um enlace de saída para a internet limitada onde para que se evite gastos com aquisição de links com maior largura de banda se possa definir esquemas de prioridade para seus pacotes e com isso otimizando o uso da rede.

Pode-se concluir também ao fim dos experimentos que o FreeBSD e suas ferramentas de suporte a QoS são bastante eficientes e podem ser utilizadas pelas empresas de forma a reduzir custos com a aquisição de enlaces com maior largura de banda e licenças de softwares proprietários além de evitar que o uso abusivo de determinadas aplicações interfiram no funcionamento das aplicações que necessitam de requisitos mínimos para ter um bom funcionamento.

Como trabalhos futuros sugere-se:

- Repetir os experimentos em um ambiente empresarial;
- Repetir os experimentos utilizando outro sistema Operacional;
- Repetir os experimentos com o AltQ no OpenBSD com o objetivo de comparar a implementação do AltQ neste sistema, o qual tem suporte nativo, com a do FreeBSD;
- Construção de uma Interface Gráfica para o Gerenciamento/Criação das de controle de Banda do IPFW e/ou AltQ de forma a tornar mais acessível a usuários menos especializados;
- Utilizar Ambiente de Alta-Disponibilidade para prover maior confiabilidade do Sistema.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Lu, Guojun. **Communication and Computing for Distributed Multimedia Systems**. Artech House Inc, 1996.
- [2] Willrich, Roberto. **Apostila da Disciplina de Sistemas Multimídia**, 2004, Disponível em <http://www.inf.ufsc.br/~willrich/Ensino/INE5639>
- [3] Rizzo, Luigi, “DUMMYNET”, Disponível em http://info.iet.unipi.it/~luigi/ip_dummysnet/ 2004.
- [4] Kurose, James F.. **Redes de Computadores e a Internet: Uma Nova Abordagem**. Pearson Education 2004
- [5] Cho, Kenjiro. **A Framework for Alternate Queueing: Towards Traffic Management by PC-UNIX Based Routers**, Artigo encontrado na Internet <http://www.csl.sony.co.jp/person/kjc/kjc/papers/usenix98/altq.html#ALTQ>
- [6] Schneider, Wolfram. **Manual do AltQ**, Disponível em <http://www.openbsd.org/cgi-bin/man.cgi?query=altq.conf>
- [7] Black, D et al **Na Architecture for Differentiated Services**, Request for Comments 2475. Disponível em: <http://www.ietf.org/rfc/rfc2475.txt> 1998
- [8] Braden, R.; Clark, D. & Shenker, S. **Integrated Services in the Internet Architecture: an Overview**. Request for Comments 1633. Disponível em: <http://www.ietf.org/rfc/rfc1633.txt> 1994
- [9] Jones, Rick. “Netperf”, Disponível em <http://www.netperf.org> 2004
- [10] Laine, J., Saaristo, S., and Prior, R. (2002) “Rude & Crude”, Disponível em <http://rude.sourceforge.net> 2004.