UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE
CIÊNCIA DA COMPUTAÇÃO

Ricardo Henrique Bezerra Azoubel

UMA PLATAFORMA DE TESTES COM SERVIÇOS DIFERENCIADOS PARA MODELAGEM DE TRÁFEGO DE VOZ SOBRE IP: análises de desempenho e de impacto

São Luis
2004

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE
CIÊNCIA DA COMPUTAÇÃO

UMA PLATAFORMA DE TESTES COM SERVIÇOS DIFERENCIADOS PARA MODELAGEM DE TRÁFEGO DE VOZ SOBRE IP: análises de desempenho e de impacto

Dissertação submetida à Universidade Federal do Maranhão, como parte dos requisitos para a obtenção do grau de mestre em Ciência da Computação.

Ricardo Henrique Bezerra Azoubel

Prof. Dr. Zair Abdelouahab
Orientador

São Luis
2004

UMA PLATAFORMA DE TESTES COM SERVIÇOS DIFERENCIADOS PARA MODELAGEM DE TRÁFEGO DE VOZ SOBRE IP: análises de desempenho e de impacto.

Ricardo Henrique Bezerra Azoubel

Esta Dissertação, por meio da comissão julgadora dos trabalhos de defesa de Dissertação de Mestrado, foi julgada adequada para a obtenção do título de "Mestre em Engenharia Elétrica, Área de Concentração Ciência da Computação", e aprovada na sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Maranhão, em sessão pública realizada em / / .

Presidente _____

Prof. Dr. Zair Abdelouahab
(Orientador)

1º Examinador _____

Prof. Dr. Djamel Sadok

2º Examinador _____

Prof. Dr. Edson Nascimento

Dedicatória

A meu Pai, Dedé Azoubel, de qualquer forma e
n'alguma forma ..... feliz. Muito feliz, com certeza.
A Vadinho, meu amado e eterno filhote.

Agradecimentos

Sobretudo a **Deus** criador, pela saúde e possibilidade de realizar este trabalho.

A minha mãe, **Lourdinha**, pelo imensurável amor e ajuda dados em toda minha existência.

A minha esposa e eterna companheira, **Sheylinha**, pela paciência, compreensão e amor despejados durante o trabalho.

Aos meus irmãos **Drey**, **Anita** (mamainha), **Uziel**, **Tadeu**, **Lúcia** e **Zé Cláudio**, assim como aos meus **cunhados** e **sobrinhos**, pelo apoio e incentivo diários.

Ao **professor Zair**, pela disponibilidade, orientação e incentivo, importantes para o desenvolvimento desta dissertação.

Ao **professor Carlos Brandão**, pela orientação técnica e incentivo, essenciais para a realização deste trabalho.

Ao amigo **Wagner Elvio**, pelo grande apoio na construção da plataforma de testes.

A **Alcides**, **Ismael**, **professora Maria da Guia** e demais **amigos alunos** do programa de pós-graduação do departamento de energia elétrica da UFMA.

Resumo

Este trabalho apresenta uma plataforma de testes (*testbed*) sem custos, construída num ambiente controlado composto por microcomputadores e softwares livres. Implementa-se, em tal plataforma, o modelo de serviço diferenciado (DiffServ), com encaminhamento expresso (PHB EF). Propõe-se, fundamentalmente, a partir da obtenção das principais métricas de QoS (atraso, jitter, perda e vazão), a análise do desempenho de tráfego característico de voz, quando submetido a testes experimentais em vários cenários e condições. Inicialmente, num ambiente capaz de diferenciar tráfego, modelam-se fluxos gerados por codificadores/decodificadores de voz padronizados (G.711 e G.726), em que se varia o tamanho dos pacotes e a quantidade de fluxos agregados, em cenários com e sem QoS. Compara-se, em seguida, o comportamento de fluxos gerados por fontes atividade-silêncio (ON-OFF) e contínuas (CBR). Pode-se perceber nesse estudo o quanto a variação do tamanho dos pacotes impacta no desempenho dos pacotes mais prioritários. Realiza-se, na seqüência, uma análise específica do fator agregação em fluxos gerados por fontes ON-OFF e observa-se a atuação do princípio básico do modelo DiffServ, onde fluxos agregados recebem tratamento diferenciado. Estuda-se, por fim, através da utilização de protocolos de transporte (UDP e TCP) e de fluxos elásticos do tipo FTP, o quanto o tráfego de melhor esforço impacta no desempenho de fluxos modelados de voz.

Palavras-chaves: Serviços Diferenciados (DiffServ), Qualidade de Serviço e Voz sobre IP (VoIP).

Abstract

This work presents a platform of tests (testbed) inexpensive, constructed in a controlled environment composed by microcomputers and free softwares. It is implemented, in such platform, the differentiated service model (DiffServ), with expedited forwarding (PHB EF). It is basically considered, from the collection of metrics main of QoS (delay, jitter, loss and throughtput), the performance analysis of voice characteristic traffics, when submitted to experimental tests in some scenes and conditions. Initially, in an environment capable to differentiate traffics, flows generated by standardized voice coder/decoder (G.711 and G.726) are modeled, in which the packets size and the amount of aggregate flows are varied, in scenes with and without QoS. It is compared, after that, the behavior of flows generated by activity-silence (ON-OFF) and continuous (CBR) sources. Can be perceived in this study how much the packets size variation influence in the performance of the most priority packets. It is carried, in the sequence, a specific analysis of the aggregation factor in flows generated by ON-OFF sources, in which can be observed the action of the basic principle of the model DiffServ, where aggregate flows receive differentiated treatment. It is studied, finally, through the use of transport protocols (UDP and TCP) and of elastic flows of FTP type, how much the best effort traffic confuses the performance of voice modeled flows.

Keywords: Differentiated Services, Quality of Service and Voice over IP (VoIP).

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

ADPCM	Adaptive Differential Pulse Code Modulation
AF	Assured Forwarding
BA	Behavior Aggregate
BB	Bandwidth Broker
BE	Best Effort
CBQ	Class-Based Queuing
CBR	Constant Bit Rate
COPS	Commom Open Policy Service
DSCP	DiffServ Code Point
DSP	Digital Signal Processor
DWDM	Dense Wavelength Division Multiplexing
EF	Expedited Forwarding
FEC	Forward Error Correction
FIFO	First In First Out
FTP	File Transfer Protocol
GPS	Global Positioning System
GRED	Generic Random Early Detection
HTTP	Hipertext Transfer Protocol
IETF	Internet Engineering Task Force
IPDV	Instantaneous Packet Delay Variation
IPPM	IP Performance Metrics Working Group
ISO	International Standardization Organization
ISP	Internet Service Provider
ITU-T	International Telecommunications Union - Telecommunication Standardization Sector
MOS	Mean Opinion Score
MPEG	Moving Picture Experts Group
MTU	Maximum Transfer Unit
NFS	Network File System
NTP	Network Time Protocol
OWD	One-Way Delay
P2P	Peer-to-Peer
PCM	Pulse Code Modulation
PHB	Per-Hop Behavior
PQ	Priority Queuing
QDISC	Queue Discipline
QoS	Quality of Service
RED	Random Early Detection

RFC	Request for Comments
RSVP	Resource Reservation Protocol
RTP	Real Time Transport Protocol
SFQ	Stochastic Fairness Queuing
SLA	Service Level Agreement
SLS	Service Level Specifications
TC	Traffic Class
TCP	Transmission Control Protocol
ToS	Type of Service
UDP	User Datagram Protocol
VoIP	Voice over IP

1. Introdução

O surgimento de tipos de aplicações, como multimídia, voz sobre IP (VoIP), educação à distância e vídeo-conferência, por exemplo, exige o uso de técnicas e mecanismos avançados para atender às necessidades dos usuários. Por tratar-se de serviços que ocupam muito os recursos de uma rede, exigem, inclusive, controles rígidos para uma comunicação mínima e satisfatória. No mais, junta-se isso ao fato de que a Internet, já há algum tempo não mais restrita ao meio acadêmico, é um ambiente oportuno para o uso de tais aplicações, dadas sua escalabilidade e abrangência mundial. Tem-se, portanto, o objetivo maior de se disponibilizar sistemas críticos em ambientes de rede públicos, concorrentes e compartilhados (caso da Internet).

Vêem-se, com a rede mundial, tendências crescentes na necessidade de convergência das redes de computadores numa infra-estrutura possivelmente única, capaz de integrar serviços de dados, voz e até vídeo. Isto é desafiador, considerando tratar-se de uma rede IP construída originalmente para prover apenas serviços tradicionais, de melhor esforço (Best Effort), sem garantias de qualidade de serviço (QoS). Desta forma, para que haja eficiência na transmissão dos dados e satisfação dos usuários, presume-se que as redes não mais devam tratar as aplicações com os mesmos pesos, mas, sim, de formas reservada ou diferenciada, de acordo com o grau de tolerância e os requisitos métricos (atrasos, perdas, vazão) de cada uma delas. Pode-se afirmar então que aplicações de tempo real (VoIP, por exemplo) não funcionam eficientemente numa rede sem QoS (ainda mais se esta estiver carregada), cujos pacotes são manipulados com os mesmos requisitos aos de uma aplicação não sensível ao tempo (FTP e e-mail, por exemplo).

Qualidade de serviço em redes IP é, portanto, um aspecto fundamental para o desempenho fim-a-fim das aplicações, sendo crucial o entendimento de seus princípios, parâmetros, mecanismos, algoritmos e protocolos, desenvolvidos, testados e utilizados para a obtenção de resultados favoráveis.

De uma forma geral, o objetivo maior das pesquisas na área de QoS é a maximização da utilização de redes locais ou remotas, através do uso mais eficiente de seus recursos e de um melhor controle dos níveis de congestionamento do tráfego, adicionando

“alguma inteligência” à rede com o fim de reservar recursos ou diferenciar pacotes de dados, de acordo com os requisitos pré-determinados. Em redes IP, mais especificamente, os objetivos da qualidade de serviço são traduzidos em desafios, resumidos a seguir, segundo [2]:

- O IP, por default, como protocolo, não tem garantia de qualidade de serviço e;
- Com a Internet, a base instalada do IP tornou-se mundial, o que mantém inviável a mudança de protocolo.

O primeiro desafio resume-se na simplicidade do protocolo de rede, projetado originalmente para tratar as aplicações sem distinção, por melhor esforço. Já o segundo, obriga adequar-se ao novo paradigma, sem mudanças significativas no protocolo. Mesmo frente às inadequações e fragilidades do IP, pode-se afirmar que disponibilizar QoS nesse tipo de rede não é utopia, desde que os protocolos, as arquiteturas, os algoritmos e os mecanismos sejam adequadamente aplicados e mantidos. O IETF (*Internet Engineering Task Force*) vem mantendo, desde 1994, pesquisas para prover qualidade de serviço em redes de computadores, sendo que, para isso, criou grupos de trabalhos responsáveis em estabelecer padrões e tecnologias, como por exemplo: as arquiteturas de serviços integrados (IntServ)[12] e diferenciados[32], cujas atividades foram encerradas em maio/2001 e fevereiro/2003, respectivamente, e o protocolo RSVP, principal componente dos serviços IntServ.

Associados à pesquisa do desempenho de aplicações multimídia em ambientes com qualidade de serviço, mais especificamente com serviços diferenciados (DiffServ) [1], estão alguns trabalhos de simulação disponibilizados pelo Grupo de Teleinformática e Automação (GTA) da Universidade Federal do Rio de Janeiro, como [33], [56] e [57]. Em [58], Anurag Tyagi estuda, também através de simulações, por meio da ferramenta NS-2[61], o desempenho de tráfego gerado por diferentes padrões de vocodecs de áudio, usados para manipular sinais de voz. Já em [59], Kos usa, outra vez, simulação para avaliar o comportamento de uma rede diferenciada com aplicações com distintos níveis de sensibilidades, como HTTP, FTP, vídeoconferência e voz sobre IP. Por fim, em [60], o mesmo Kos compara o desempenho de aplicações VoIP numa rede com DiffServ,

utilizando enfileiramentos dos tipos prioritário (*priority queuing*) e FIFO (*First In First Out*) numa rede simulada.

1.1 Objetivos do trabalho

De uma forma bem sucinta, este trabalho tem como finalidade avaliar, através de uma série de experimentos e medições, os níveis de efetividade dos serviços diferenciados (DiffServ), tomando como base uma plataforma de testes **(*testbed*)** composta por *hardwares de baixo custo* (microcomputadores PC) e por *softwares de código aberto*. Tal avaliação é realizada a partir de diversos cenários, que são:

- Caracterização de fluxos gerados por codificadores/decodificadores de voz (VOCODERS) distintos, com a variação do tamanho dos pacotes e da quantidade de fluxos agregados;

- Modelagem de fluxos gerados tanto por fontes contínuas (CBR) como por fontes do tipo atividade-silêncio (ON-OFF) e;

- Geração de tráfego de melhor esforço em cima da variação de seus protocolos de transporte (UDP e TCP) e do uso de sessões FTP.

Em outras palavras, objetiva-se investigar quais modelagens de vocoders podem apresentar melhor e pior desempenhos, em ambientes com e sem qualidade de serviço. Pretende-se também avaliar, num ambiente com priorização de tráfego, qual é o tipo de fonte de tráfego (CBR ou ON-OFF) que alcança os melhores resultados. Por fim, procura-se perscrutar quais impactos o tráfego de fundo pode exercer sobre fluxos modelados de voz.

De uma forma mais específica, o trabalho tem a intenção não só de investigar e observar, mas, principalmente, de interpretar e justificar, a partir das variantes ambientais aplicadas na rede, as tendências comportamentais das métricas de QoS (atrasos, jitters e taxas de descarte de pacotes e de vazão). Tais medidas referem-se tanto ao tráfego de voz quanto aos fluxos de melhor esforço. Estudos de agregação de fluxos de voz também constituem os objetivos específicos da dissertação.

1.2 Organização dos tópicos

O texto está organizado com o Capítulo 2 apresentando uma fundamentação teórica sobre qualidade de serviço. O Capítulo 3 discute aspectos importantes do modelo diferenciado, como condicionamento de tráfego e o conceito de comportamentos por nó, e aborda o serviço no ambiente Linux. O Capítulo 4 mostra o ambiente de testes adotado, com a apresentação da topologia de rede, dos métodos e das ferramentas utilizadas. O Capítulo 5 confirma a capacidade do ambiente em diferenciar tráfego e apresenta estudo da modelagem de vocoders em ambientes com e sem serviço diferenciado. O Capítulo 6 estuda a modelagem de tráfego de voz, gerado por fontes contínuas e de atividade-silêncio. O Capítulo 7 avalia os impactos causados pela variação do tráfego de melhor esforço no desempenho de fluxos modelados de voz. Finalmente, o Capítulo 8 conclui o trabalho, destacando as contribuições e relacionando trabalhos futuros.

1.3 Conclusão do capítulo

O capítulo inicial se encarrega apenas em apresentar as primeiras considerações sobre o tema, introduzindo, basicamente, os desafios da área de qualidade de serviço e os objetivos do trabalho. A organização do texto também é mostrada.

2. Fundamentos Básicos de QoS

Serão apresentadas a seguir algumas abordagens importantes sobre qualidade de serviço, necessárias para a consistência da fundamentação teórica do trabalho. Serão discutidos vários aspectos, como a necessidade de QoS nas redes de computadores, os modelos propostos e as métricas essenciais para estudo.

2.1 O porquê da Qualidade de Serviço

A Internet tem evoluído de tal forma que praticamente todos os tipos de aplicações, como os sistemas críticos de voz e de vídeo, precisam convergir para aquele ambiente. Como já descrito no capítulo anterior, a Internet (ou o protocolo IP), por si só, não é um meio propício para as exigências de recursos que essas aplicações demandam. Elas têm, geralmente, como requisitos, a baixa latência, pequenos níveis de perdas de pacotes e um certo grau de desempenho.

A resposta pode estar então em fazer com que esses sistemas tenham garantias de ser executados, atendendo a seus requisitos de uma maneira satisfatória, sem necessidade de elevar-se a quantidade de recursos[1] da rede. De uma forma geral, prover qualidade de serviço numa rede como a IP, inadequada pela imprevisibilidade de resultados, pode ser justificado pela alternativa de poder-se aumentar a eficiência dessa rede sem ter-se que pagar mais por isso.

Na verdade, de acordo com a justificativa acima e a partir da idéia posta em [4], tem-se a seguinte discussão: com as novas tecnologias atualmente sendo disponibilizadas (fibras e DWDM, por exemplo), geralmente vê-se abundância e o barateamento de recursos de rede (como largura de banda). Com isso, não se faz necessário ter-se qualidade de serviço. Por outro lado, mesmo com o aumento substancial dos mesmos recursos (toma-se novamente a largura de banda como exemplo), sempre vão existir aplicações capazes de consumi-las e, desta forma, haverá, sempre, a necessidade de QoS na rede, conduzindo à opinião de que aumento na largura de banda não vai eliminar a necessidade de QoS. Sobre esta discussão, [3] abre a questão do gerenciamento de rede com QoS contra o aumento na

capacidade de recursos, com o segundo argumento podendo ser, muitas das vezes: i) mais barato, considerando-se a economia de tempo e a pouca necessidade de capacitação dos técnicos de suporte e ii) mais preciso, já que se pode estimar quanto gastar na compra de recursos. No entanto, nem sempre aumentar a capacidade da rede significa resolver os problemas do meio, quando então há a necessidade de inteligência para um bom gerenciamento dos recursos, justificando assim o uso de QoS. Para aplicações VoIP, por exemplo, o simples aumento na largura de banda pode não ser a solução, já que esse tipo de aplicação apresenta outros requisitos (o de tempo, por exemplo) mais críticos e significativos para o alcance de bons resultados.

Não é possível à rede dar algo que ela não tenha. Ou seja, QoS não pode criar largura de banda inexistente, apesar da disponibilidade desse recurso ser um ponto importante e um bom começo para se garantir uma execução satisfatória das aplicações. A qualidade de serviço apenas tenta gerenciar a largura de banda, de acordo com a demanda das aplicações e as configurações da rede, melhorando seu desempenho e garantindo a alocação de recursos aos fluxos de dados.

Em cima disto, é importante esclarecer que, além da idéia exposta acima, de que QoS não cria recursos, ainda há, segundo [3], algumas outras restrições que ainda persistem em torno do assunto, como por exemplo: i) “os mecanismos de QoS atuam preferencialmente em situações de contenção da rede, sendo irrelevantes nos casos de ociosidade”; ii) “QoS é elitista e injusta, na medida em que dá prioridade a algumas classes em detrimento de outras, forçando os usuários a pagarem mais caro para sentir que suas aplicações são mais eficientes”; e iii) “QoS não consegue compensar imperfeições da rede, como maus projetos e situações drásticas de congestionamento”.

O certo é que, apesar de tudo isso, mas frente aos desafios impostos pela necessidade de integração de aplicações na Internet, QoS é tema de extrema importância e utilidade, que alcança avançadas pesquisas em todo mundo e com alguns produtos (padronizados por organismos internacionais) já disponíveis no mercado. Esses são aspectos mais do que suficientes para justificar o porquê da qualidade de serviço nas atuais redes de computadores.

---

[1] Por *recursos* entendem-se, por exemplo, largura de banda no meio e memórias e processadores nos roteadores.

2.2 Múltiplas definições

O termo QoS pode assumir várias conotações, dependendo de onde poderá ser aplicado. Segundo a ISO (*International Standardization Organization*), QoS é o efeito do desempenho de um serviço para medir níveis de satisfação dos usuários daquele serviço[8]. Já segundo [9], QoS serve tanto para definir o desempenho da rede em relação às necessidades das aplicações, como para possibilitar a essas redes garantias de desempenho.

Outra definição pode ser aproveitada a partir de [10], em que se tem QoS como a capacidade de um elemento de rede ter algum nível de garantia de que requisitos de tráfego e serviços possam ser satisfeitos, com a rede provendo formas de encaminhamento de dados de maneira consistente e previsível. Já [11] se aproxima desta definição, mas explica que o termo "elemento de rede" pode ser uma aplicação, um hospedeiro (*host*), um roteador ou outro dispositivo.

De uma forma sucinta, e aproveitando as definições aqui utilizadas, pode-se dizer que qualidade de serviço seja a capacidade da rede de prover melhores serviços a tráfego selecionado (prioritário), sobre várias tecnologias de acesso, como, por exemplo, Ethernet, Frame Relay e ATM. No mais, tem como objetivo básico disponibilizar os recursos necessários aos pacotes de dados, de acordo com os níveis de relevância de cada serviço, oferecendo assim maior garantia às aplicações e maior satisfação aos usuários da rede.

2.3 O Melhor Esforço

Redes IP são geralmente descritas como redes de melhor esforço (*best effort*). Isto se refere ao modelo no qual a rede não diferencia o tratamento dos serviços que nela trafegam, sendo todos os pacotes manipulados de uma mesma forma, com uma mesma prioridade.

A partir da concepção do protocolo IP, sabe-se que este não garante entregar os pacotes num tempo mínimo necessário para as aplicações nem tampouco entregá-los no seu destino. Isto quer dizer que deve existir um esforço considerável dos elementos de conectividade, geralmente roteadores, em cumprir com tais entregas. Ou seja, a rede

encarrega-se de liberar todos os pacotes tão rapidamente quanto possível, mas sem garantias de manter a qualidade dos serviços, sendo constantemente incapaz de tomar medidas preventivas ou corretivas que possam manter tal qualidade em um patamar satisfatório para a maioria das aplicações. Além disso, dado que o tráfego de dados é imprevisível e em rajadas[5], há os problemas de congestionamentos, onde, por representar o esgotamento dos recursos disponíveis, pode provocar perdas ou atrasos dos pacotes. O resultado de tal escassez é uma sensível degradação da qualidade percebida pelos usuários, o que, mesmo para redes de melhor esforço, pode ser minimizado se políticas de gerenciamento de filas e técnicas de controle de congestionamento apropriadas forem utilizadas[6].

As redes de melhor esforço, representadas pelo protocolo IP, têm, como já levantado inicialmente, a vantagem da simplicidade. Necessitam então conviver com serviços mais completos e seguros, como é o caso do protocolo TCP, por exemplo, usado para suportar transferência de dados mais confiáveis, fim-a-fim, bidirecional e orientada à conexão, com mecanismos de controle de erros e seqüenciamento de pacotes[7]. O TCP é visto como a força motriz da Internet, com suporte à operação de suas principais (e pioneiras) aplicações, como a web (HTTP), correio eletrônico e transferência de arquivos (FTP). Partindo da inadequação do modelo de serviço de melhor esforço no fornecimento de garantias de QoS para as aplicações e da importância do TCP na eficiência fim-a-fim experimentada pelas aplicações que o utilizam, conclui-se pela necessidade de mais pesquisas voltadas às implicações do desempenho do TCP no contexto das emergentes arquiteturas para fornecimento de QoS em redes IP.

No entanto, nem todas as aplicações requerem os serviços confiáveis providos pelo TCP. Para esses sistemas o protocolo UDP foi desenvolvido com o intuito de oferecer um serviço de transporte orientado a datagramas, sem conexão e, portanto, não confiável, sem garantias do dado ser disponibilizado na rede ou, então, ser disponibilizado numa ordem incorreta. Ao contrário do TCP, o UDP não provê qualquer sistema de controle de congestão e adaptação, disponibilizando assim uma forma mais rápida e direta de enviar e receber dados, com um mínimo de sobrecarga, o que o torna propício para aplicações mais sensíveis ao tempo, como no caso das baseadas em voz e vídeo. Isto, aliado ao fato do UDP não ser confiável, faz com que este protocolo necessite do suporte de mecanismos de qualidade de serviço. Significa dizer que, por questões de eficiência, não é ideal o uso de

aplicações mais críticas a atrasos ou perdas de pacotes, sob UDP em redes de melhor esforço, onde não se tem garantia de QoS.

2.4 Classificação das Aplicações

O entendimento do tipo de tráfego gerado e o comportamento das aplicações são formas importantes para se determinar o modelo de QoS a ser disponibilizado aos usuários. Em [12], o grupo de trabalho IntServ[2], do IETF, classificou as aplicações de acordo com a divisão a seguir:

a) Aplicações elásticas

b) Aplicações de tempo real

   b.1) Tolerantes

   b.2) Intolerantes

Então, com base nessa abordagem, e de uma forma sucinta, pode-se definir as aplicações da seguinte forma, classificadas de acordo com os seus requisitos de QoS:

2.4.1 Aplicações Elásticas

São aquelas que sempre aguardam pela chegada dos dados, processando-os imediatamente quando isto acontece, sem necessidade de *buffer* para uso posterior. Ou seja, são aplicações que podem tolerar altas variações de vazão e perdas de pacotes, sem que isso prejudique sua qualidade[13]. Como exemplo, têm-se as aplicações de transferências de dados (FTP, e-mail), telnet e NFS, que são categorizadas de acordo com os requisitos de atraso e vazão, como mostrado a seguir:

a) *Asynchronous Bulk Traffic* – Quando os requisitos de atraso e vazão são bastante flexíveis, tal como e-mail e fax.

b) *Interactive Burst Traffic* – São aplicações com uma interação homem-máquina, cujos requisitos são aproximados aos das aplicações de tempo real

[2] http://www.ietf.org/html.charters/intserv-charter.html.

e idealmente têm atrasos entre 200-300ms ou menos, mas que podem tolerar algo ligeiramente acima disto[13]. Exemplos: telnet ou NFS.

c) *Interactive Bulk Traffic* – São aplicações interativas com exigências de atraso semelhantes às mostradas acima mas que, geralmente, seus usuários esperam uma taxa de vazão maior. Exemplos: transferência de dados (FTP) e web (HTTP).

2.4.2 Aplicações de Tempo Real

São as que não admitem atrasos, erros e variações, tanto nos atrasos entre os pacotes quanto na vazão, sendo sensíveis ao tempo, como aplicações de voz e vídeo, por exemplo. Cobrem situações em que os pacotes devem chegar no destino dentro de um período de tempo limitado, sendo que, de outra forma, o pacote certamente será descartado. Usam armazenamentos no destino para suavizar a variação no tempo de chegada do pacote, sendo, por isso, conhecidas como aplicações "*playback*", onde o transmissor empacota e envia os dados e o receptor desempacota-os e tenta reproduzir o mesmo sinal enviado. Há, com isso, um nível maior de fidelidade possível, através da disponibilização de *buffers*, tal como descreve [12], que subdivide ainda as aplicações de tempo real em tolerantes e intolerantes.

a) Aplicações de tempo real tolerantes – são aquelas que aceitam certos níveis de degradação dos requisitos de QoS (como perdas de pacotes), mantendo a qualidade aceitável e satisfatória através da operação dentro de uma escala de provisão de QoS. São, por sua vez, subdivididas em adaptativas e não-adaptativas.

    a.1 – Adaptativas – são aquelas capazes de ajustar as demandas de QoS dentro de um intervalo de valores aceitáveis, sendo provocadas por mecanismos que informam à aplicação o atual nível de eficiência da rede[13]. Adaptação a perdas de pacotes, por exemplo, pode indicar congestionamento na rede, tendo como conseqüência a redução das taxas de transmissão, tal como acontece no protocolo TCP.

a.2 – Não-adaptativas – são aquelas que não conseguem se ajustar da mesma forma que as adaptativas, mas que podem tolerar alguma variação nos requisitos de QoS. A qualidade de uma aplicação de vídeo, por exemplo, pode ser degradada pela perda de uma certa percentagem de pacotes, mas, mesmo assim, ainda poderá ser claramente visualizada (e entendida) pelos usuários.

b) Aplicações de tempo real intolerantes – são aplicações que necessitam que seus requisitos de QoS sejam rigorosamente obedecidos, pois, caso contrário, não realizarão suas tarefas suficientemente, resultando em distorções significativas na reprodução do sinal. Exemplos disso são as aplicações responsáveis pelo controle remoto de equipamentos de missão crítica, vistas em instrumentos de cirurgia e na robótica.

2.5 Modelos de QoS

Apresentam-se aqui os modelos (ou as arquiteturas) propostos no âmbito do IETF, comunidade internacional responsável pela padronização de tecnologias Internet, para a implementação de QoS na rede mundial, fundamentada nos serviços integrados e diferenciados, sendo o último o foco primordial desta dissertação.

2.5.1 Serviço Integrado

Visando estender o serviço de melhor esforço, na construção de uma infra-estrutura mais robusta para a Internet que pudesse suportar o transporte de informações em tempo real, foi criado o grupo de trabalho de serviços integrados do IETF. Tal força tarefa é responsável ainda em definir e padronizar interfaces e requisitos necessários para a implementação de um novo modelo de QoS. O grupo IntServ, como também ficou conhecido, responsabilizou-se também em dar garantias de QoS ao longo de um percurso de fluxo de dados, no qual todos os elementos de conectividade (roteadores) teriam que suportar, através de mecanismos de sinalização, a reserva de recursos. O RSVP é exemplo de protocolo de sinalização e será explicado ainda neste capítulo. Os elementos de rede devem, para tal reserva, estar em conformidade com um conjunto de recomendações (RFCs) relacionadas à arquitetura IntServ, dentre as quais, destacam-se [15], [16] e [12],

sendo este último o documento básico que aborda os elementos da arquitetura, o modelo em si, os mecanismos de controle de tráfego e o protocolo RSVP.

O modelo de serviço integrado é então marcado pela capacidade dos roteadores em reservar recursos, de forma a prover qualidade de serviço a fluxos de pacotes específicos e individuais (e não a agregados de fluxos), com informações sobre tais recursos[3] mantidas em cada roteador ao longo dos percursos tomados pelos fluxos[4]. E, por operar com fluxos individuais é que o modelo IntServ tem como característica mais particular a alta granularidade na reserva de recursos, com um controle fino das solicitações, que são iniciadas pelas aplicações que requeiram garantias de QoS. Isto é tido como vantagem do modelo, pois há uma maior precisão no uso e na alocação dos recursos. Entretanto, para que tal granularidade seja possível, é necessário que cada elemento da rede (roteadores e, inclusive, as estações que hospedam as aplicações) ao longo do percurso mantido pelo fluxo participe do processo de sinalização e alocação. Desta forma, a arquitetura torna-se pesada, pouco escalar e, conseqüentemente, imprópria para redes abrangentes, como a Internet.

O grupo IntServ desenvolveu suporte para duas amplas classes de aplicações[14], quais sejam:

a) Aplicações de tempo real, com restrições nos requisitos de QoS, como já apresentado recentemente;

b) Aplicações tradicionais, onde os usuários buscam níveis de eficiência similares aos providos pelas redes de melhor esforço.

A primeira classe de aplicações deu base para o grupo de trabalho padronizar o serviço garantido[15] (*guaranteed service*) e, a segunda classe, para o serviço de carga controlada[16] (*controlled load service*), com, ambas, representando a implementação dos serviços integrados.

[3] Também conhecidas como *informações de estado*.

[4] Qualquer conjunto de pacotes identificável, que parte de uma origem para um ou mais receptores, com um tratamento comum de qualidade de serviço[14]. De uma outra forma, tem-se um fluxo de dados quando um conjunto de pacotes é transportado de um ponto de origem a outro de destino, usando, para cada extremo, pelo menos, o mesmo endereço IP, a mesma porta de conexão e o mesmo protocolo de transporte.

2.5.1.1 Serviço Garantido (*Guaranteed Service*)

Serviço garantido é a implementação IntServ onde as métricas de QoS são asseguradas. Ou seja, é onde os níveis de vazão são garantidos, a proteção contra perdas de pacotes é mantida e os limites rígidos de atraso são propiciados. No entanto, a variação no atraso entre os pacotes (*jitter*), não é garantida[15].

O serviço é próprio para aplicações com requisitos rígidos de tempo real, que, para o alcance de resultados satisfatórios, necessitam que as métricas acima descritas sejam alcançadas. Basicamente, com este tipo de serviço, a taxa de transmissão dos pacotes é cumprida, sendo exigido que todos os nós intermediários o implemente, o que torna o serviço muito pouco escalável.

2.5.1.2 Serviço de Carga Controlada (*Controlled Load Service*)

De outra maneira, a implementação IntServ do tipo carga controlada é, fundamentalmente, aquela que, sob condições controladas (sem congestionamentos), se aproxima do comportamento de aplicações que recebam o serviço de melhor esforço. Assume-se, a partir de [16], que pequenas taxas de descarte de pacotes são permitidas e que a maioria dos pacotes deverão apresentar valores de atrasos próximos ao atraso mínimo dos pacotes bem sucedidos (não descartados). A grande diferença entre os serviços de carga controlada e de melhor esforço é que, no primeiro, os fluxos não sofrem grandes deteriorações com o aumento da carga na rede.

Como já sinalizado recentemente, as aplicações aqui definidas como tradicionais, em que níveis próximos aos oferecidos em redes de melhor esforço são providos, formam a base para a padronização do serviço de carga controlada. No entanto, como este serviço garante um certo atraso médio, com o atraso experimentado por alguns pacotes não podendo ser determinado, pode-se indicá-lo às aplicações do tipo tempo real tolerantes[18] [19], que funcionam bem quando a rede não está sobrecarregada.

2.5.1.3 Componentes para implementação do IntServ

O modelo de referência para implementação da arquitetura IntServ inclui as funções necessárias para o fornecimento dos serviços, sendo composto por quatro componentes, conforme descrição a seguir:

a) **Classificador** – permite identificar a quais fluxos os pacotes pertencem e a quais classes de QoS os mesmos serão atribuídos. Cada pacote recebido no roteador deve ser mapeado em alguma classe, com todos os pacotes da mesma classe recebendo o mesmo tratamento no escalonador. O conceito de classe é, na verdade, uma abstração, já que o mesmo pacote pode ser classificado de diferentes formas pelos roteadores que compõem o percurso do pacote, dependendo da concepção adotada. Por exemplo, roteadores de borda podem utilizar uma classe para cada fluxo e os roteadores internos podem mapear vários fluxos numa única classe.

b) **Escalonador de pacotes** – gerencia, através de políticas de enfileiramento, o encaminhamento dos fluxos na rede, de modo a satisfazer os requisitos de QoS. Um componente que pode ser considerado como parte do escalonador é o *avaliador*, responsável por medir as características de tráfego dos fluxos, de forma a auxiliar no escalonamento e no controle de admissão.

c) **Controle de admissão** – determina a existência de recursos, decidindo se um pedido de alocação pode ser garantido ou não. Compreende algoritmos de decisão, implementados pelos roteadores, para determinar se um fluxo pode ter sua qualidade de serviço garantida, sem que outros fluxos, já com seus recursos alocados, sejam afetados.

d) **Protocolo de reserva** – antes de encaminhar seus dados pela rede, as aplicações devem, através de um processo de sinalização, negociar a reserva de recursos necessários para tal transmissão. Para isto utilizam, geralmente, um protocolo de sinalização, que, na maioria das vezes, é o RSVP, desenvolvido para atuar em redes com serviço integrado. O RSVP é utilizado tanto pelos sistemas finais como pelos roteadores da rede. No primeiro caso, o protocolo solicita níveis específicos de QoS para as aplicações. Já referente ao uso pelos roteadores, o RSVP é responsável em repassar as requisições aos demais roteadores do percurso por onde os dados serão transmitidos, assim como estabelecer e manter as informações de estado suficientes para o serviço. Um fato importante é que todos os roteadores que compõem o caminho dos dados das aplicações devem participar do processo de reserva de recursos, sobrecarregando a rede e contribuindo para a

baixa escalabilidade da solução. Dentre as características do RSVP, podem ser destacadas as seguintes:

- O estilo de reserva de recurso adotado pelo RSVP é o *soft-state*, também conhecido como sem conexão (*connectionless*). Ou seja, as informações de estado têm um período de validade, são mantidas em cache e, periodicamente, são refrescadas pelos sistemas finais. As informações não utilizadas por um certo tempo são expiradas e as alteradas por um roteador são, automaticamente, difundidas pela rota;

- O RSVP utiliza o conceito de "*receiver-initiation*". Isto é, quem é responsável por iniciar e manter as reservas é o nó receptor, que sabe o que quer (ou pode) receber;

- RSVP é um protocolo de sinalização e, não, de roteamento. Da mesma forma, não é um protocolo de transporte e, portanto, não carrega dados. Para isto, trabalha em paralelo com os protocolos TCP ou UDP;

- Roteadores que compõem o percurso dos dados, mas que não implementam o protocolo RSVP, não inviabilizam o processo de qualidade de serviço, já que encaminham transparentemente as mensagens. No entanto, isto cria "pontos-fracos" na rede, onde o serviço retoma a concepção de melhor esforço, sem alocação de recursos nesses nós.

- O protocolo IP, nas versões 4 e 6, suporta o RSVP.

2.5.1.4 Críticas ao modelo IntServ

Na arquitetura de serviço integrado, a qualidade de serviço é obtida através da reserva de recursos por fluxo, o que, como já descrito, torna o modelo bastante granular, com uma grande precisão (controle fino) no uso e na alocação dos recursos. Esta é a vantagem do projeto, assim como o fato do IntServ ter sido projetado como extensão do serviço utilizado na Internet (melhor esforço) e, também, por não ter havido necessidade de modificações dos mecanismos de roteamento dos pacotes. No entanto, para que as garantias de QoS sejam efetivas, é necessário que todos os roteadores ao longo do percurso do tráfego implementem a classificação e o escalonamento de pacotes, o controle de

admissão e, principalmente, a sinalização de recursos. Isto tudo gera um grande problema de escalabilidade e torna o modelo impraticável para grandes redes, como a Internet, pois impõe, em termos de processamento e armazenamento, sobrecarga aos roteadores. Em outras palavras, quanto maior é a extensão da rede, maior é a quantidade de fluxos que os roteadores terão que tratar e, da mesma forma, maior será a quantidade de informações de estado que esses elementos terão que manter[5], impondo esforço excessivo sobre toda a rede. Foi justamente o problema de escalabilidade que possibilitou o estudo de soluções alternativas para a qualidade de serviço na Internet, sendo proposto pelo IETF o modelo de serviço diferenciado, com o qual os estudos experimentais deste trabalho são implementados.

2.5.2 Serviço Diferenciado

O modelo de diferenciação de serviço foi projetado como um mecanismo escalável para funcionar no centro de redes globais, como a Internet. O serviço diferenciado não guarda informações de estado sobre fluxos individuais e, por isso, não é granular como o IntServ. Ao contrário, o DiffServ é baseado na agregação de fluxos, com reservas sendo feitas para um conjunto de fluxos relacionados (conhecidos como BA – *Behavior Aggregate*), sem cargas de sinalização e, portanto, com menos sobrecarga. É esta a base da escalabilidade do modelo. Por não ser granular, o DiffServ não pode garantir os recursos para todos os fluxos que compõem o agregado. É muito importante saber que as reservas são alocadas para o BA e, desta maneira, um fluxo individual pode não atingir seus requisitos em termos de qualidade de serviço[3] (atraso, vazão, taxa de perdas). Entretanto, desde que a rede esteja corretamente provisionada, se pode alcançar, sim, níveis satisfatórios de garantia de recursos em redes com implementação dos serviços diferenciados.

O modelo baseia-se em agregar fluxos de dados com os mesmos requisitos, sendo que a informação de estado desse agregado é transportada no próprio cabeçalho IP e usada pelos roteadores como uma forma simples e eficiente de classificar os pacotes, com vistas a promover, entre eles, tratamentos diferenciados.

---

[5] Se a sinalização for realizada pelo RSVP, este, como já descrito, trabalha com o estado leve. Isto quer dizer que as informações deverão ser atualizadas periodicamente, o que prejudica ainda mais a escalabilidade.

Na verdade, a idéia de diferenciar tráfego, ao nível do protocolo IP, não é recente. Antes do aparecimento do DiffServ, bits já indicavam uma prioridade relativa ou o tipo de serviço desejado, de forma a influenciar no percurso que os pacotes poderiam seguir. Trata-se do campo ToS (*Type of Service*)[6] do cabeçalho IPv4, mostrado na Figura 2-1 e definido pelas RFCs 791[21] e 1349[22] para especificar como o pacote seria manipulado pelos roteadores. Os três primeiros bits são os níveis de precedência (ou prioridade relativa) e servem para os transmissores identificar a importância de cada pacote. Os 4 bits seguintes funcionam como chaves de classificação, servindo para especificar o tipo de transporte desejado. O bit 3 (D), quando ligado, significa que está sendo requisitado um baixo delay ao pacote; o bit 4 (T), por sua vez, sinaliza necessidade de alta vazão; o bit 5 (R) indica alta credibilidade na entrega do pacote e, por fim, o bit 6 (M), que especifica mínimo custo. Caso os quatro bits estejam desligados, isso vai significar que não há serviço especial a ser tratado. Ocorre que, na prática, a RFC 1349 não foi satisfatoriamente implementada pelos fabricantes de roteadores, por ser considerado um modelo limitado quanto à diferenciação de tráfego, uma vez que o sub-campo "Precedência" permite apenas prioridade relativa aos pacotes[7]. O IETF reviu o modelo que disponibilizava QoS por meio do uso de classes de tráfegos e de chaves de classificação e então remodelou o byte IPv4 TOS para o campo DiffServ (DS), o qual permite uma variedade maior dos tipos de pacotes que poderão ser enfileirados e escalonados na rede.

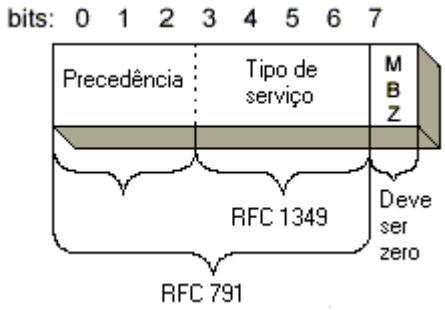


Figura 2-1 – O original byte TOS do IPv4.

A Figura 2-2 mostra a nova configuração, onde se pode observar que os seis bits do antigo byte TOS formam agora o DSCP (*DiffServ Code Point*), com a possibilidade de permitir, teoricamente, 2^6 (64) códigos distintos para representar comportamentos de enfileiramento e escalonamento dos pacotes. Os bits 6 e 7 não são utilizados.

[6] Com 1 byte (8 bits) de comprimento.
[7] Exemplo: pacotes com precedência 7 vêm antes dos com identificação 6 e assim por diante.

O fato é que o modelo DiffServ consegue diferenciar tráfego ao nível das filas, dos escalonadores, das rotas ou de outros fatores que possam interferir no desempenho dos serviços, o que o torna mais flexível do que um mecanismo de simples prioridade. Além disso, pode-se disponibilizar mais serviços com o DSCP do que a classificação por tipo de serviço (campo ToS). Vale lembrar ainda que, pelas normas, o campo DS, ao contrário do TOS, não mais sugere seleção de rotas[14]. Para informação, no IPv6, o campo redefinido pela RFC 2474 é o TC (*Traffic Class*), que, em relação ao IPv4, é utilizado exatamente para o mesmo fim.

Figura 2-2 – Campo DS do cabeçalho IPv4, definido pela RFC 2474.

Apesar de suas vantagens, o modelo DiffServ oferece também alguns problemas, como, por exemplo, não garantir recursos para todos os fluxos, com as reservas sendo feitas para a agregação como um todo (BA). Ao contrário do modelo IntServ, com o DiffServ não há o chamado controle fino das solicitações. Ou seja, a arquitetura apresenta baixa granularidade na reserva de recursos, com pouca precisão no uso e na alocação desses recursos.

2.5.3 Redes IntServ sobre DiffServ

Como se percebe, ambos os modelos apresentam vantagens e desvantagens. Com o intuito de aproveitar os benefícios das arquiteturas e aplicá-las em redes geograficamente extensas, como a Internet, estudos têm sido realizados para sugerir uma solução híbrida, através da combinação dos modelos de qualidade de serviço propostos pelo IETF. Na verdade, conhece-se a solução como “redes IntServ sobre DiffServ”, onde, para prover maior granularidade nas solicitações, os mecanismos de reserva e sinalização do serviço integrado são utilizados nas redes periféricas ou de acesso e, para maior alcance e escalabilidade, se emprega o modelo diferenciado na rede central, também conhecida como redes de trânsito, tal como mostra a Figura 2-3.

A complexidade da solução reside, principalmente, nos roteadores, que são os responsáveis pelo mapeamento entre as capacidades dos modelos, expressando aspectos

como: a transformação das classes do IntServ em comportamentos do serviço DiffServ (PHB, a ser visto no Capítulo 3); policiamentos adequados na região central e, ainda na rede diferenciada, o gerenciamento de recursos, traduzido pelo controle de admissão, efetuado a partir da disponibilidade de recursos no domínio DiffServ. Nesta linha de pesquisa, [24], por exemplo, apresenta uma plataforma de testes em Linux, que explora o mapeamento de fluxos sinalizados na rede IntServ em classes no interior do domínio DiffServ. [25], por sua vez, implementa, também através de um testbed, o provisionamento dinâmico de recursos num domínio diferenciado, por meio de negociadores de banda[8] e do protocolo COPS[26][27], que, trabalhando cooperativamente, são responsáveis pela troca dinâmica de informações sobre políticas de QoS.

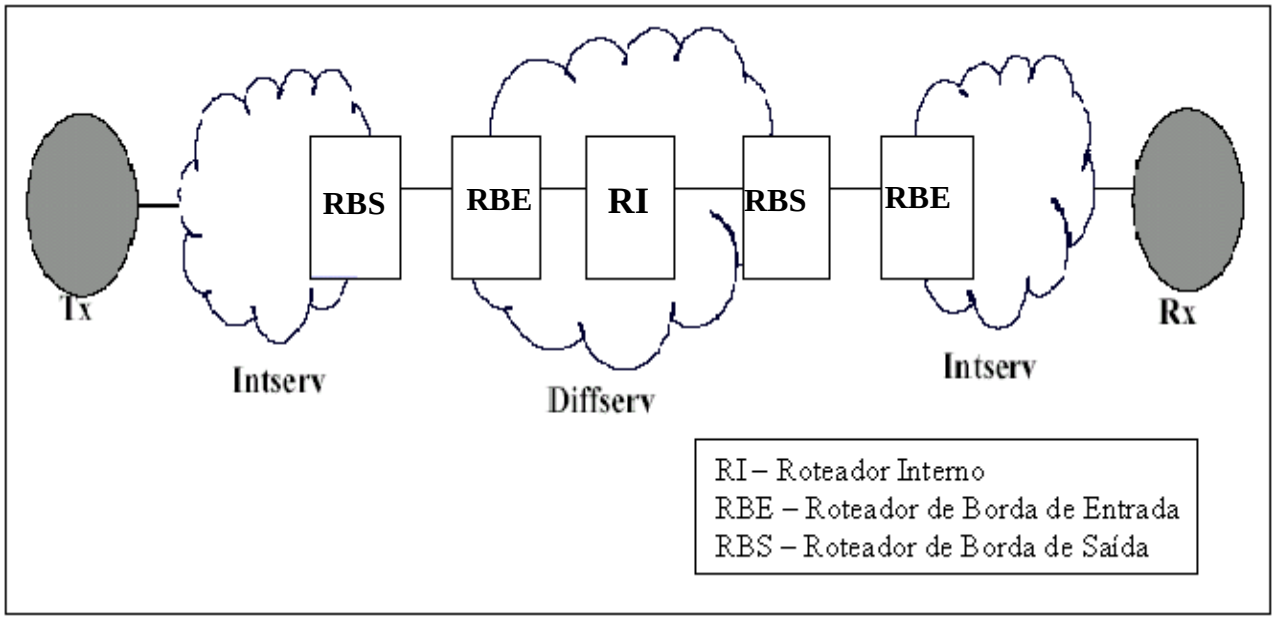


Figura 2-3 – Modelo que combina as arquiteturas integrada e diferenciada.

Uma descrição complementar dos serviços diferenciados pode ser encontrada mais adiante, no Capítulo 3.

2.6 Métricas da qualidade de serviço

As sessões seguintes definem as métricas essenciais para o estudo da qualidade de serviço provida pelas redes às aplicações. Estão inclusos neste contexto o atraso sofrido pelos pacotes e a variação do atraso entre esses mesmos pacotes, assim como a taxa de descarte, a vazão de dados e a largura de banda.

[8] Conhecidos como "*Bandwidth Broker*" ou, simplesmente, BB.

2.6.1 Atraso

De uma forma genérica, atraso (ou retardo) é o intervalo de tempo que os pacotes levam para percorrer seu caminho, do transmissor ao receptor, incluindo, obviamente, os retardos gerados pela própria rede, como os provocados pelas decisões de roteamento e pela permanência dos pacotes nas filas dos roteadores da rede.

O tipo de atraso considerado nos experimentos deste trabalho é o conhecido como atraso em um sentido (*one-way delay* ou OWD). Os conceitos de "*wire time*" e "*host time*", abordados pelo grupo de trabalho IPPM, do IETF, são fundamentais para o entendimento do OWD. A noção de "*wire time*" assume que o dispositivo de medição tem um posto de observação na ligação IP, onde se considera que o pacote tenha entrado no canal quando o primeiro bit aparecer no ponto de observação do transmissor e o pacote sai do canal quando o último bit do pacote passar pelo ponto de observação do receptor. O OWD, formalmente definido em [28], é medido justamente dentro deste conceito, a partir do tempo que o pacote tenha entrado no posto de observação do transmissor até o tempo que o último bit tenha sido observado pelo receptor, sendo a diferença desses dois valores o atraso em um sentido. A Figura 2-4 ajuda a compreender isto. A noção de "*host time*", por sua vez, é tida quando a medição é realizada por meio de software, através do registro dos tempos (*timestamps*) de transmissão e recepção dos pacotes. Esta é a noção adotada por uma das ferramentas de geração de pacotes utilizada nos estudos experimentais deste trabalho, como será visto na Sessão 4.2.3.

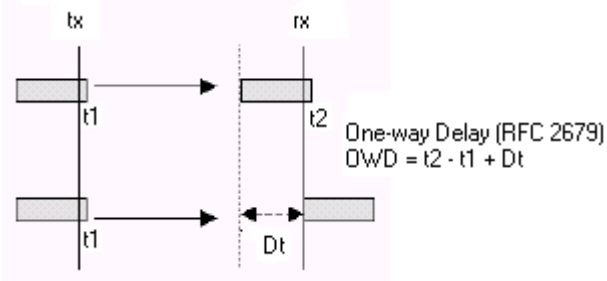


Figura 2-4 – Conceito de atraso em um sentido (OWD).

2.6.2 Variação do atraso entre pacotes (jitter)

O jitter, também conhecido como *Instantaneous Packet Delay Variation* ou IPDV-jitter, é formalmente definido em [29] e refere-se à variação do atraso entre pacotes

selecionados. De uma outra forma, o jitter baseia-se nas medidas do delay de um sentido e se define a partir de pares consecutivos de pacotes. Logo, o requisito para se medir o jitter é a existência de, pelo menos, dois pacotes.

Considerando-se D_i como o OWD do i-ésimo pacote, então o jitter relacionado àquele pacote é medido como o módulo de $(D_i - D_{i-1})$. Ou seja, jitter é a diferença modular entre os valores de atrasos de um sentido de pacotes selecionados, sendo calculado como $|(D_i - D_{i-1})|$.

Um importante uso do jitter está no estudo do dimensionamento de *buffers* para aplicações, como as de voz e de vídeo, que requerem uma entrega regular de pacotes. Ou seja, do lado do transmissor os pacotes são enviados continuamente, com um certo espaçamento entre eles. Devido aos congestionamentos da rede ou aos problemas de enfileiramento, este fluxo pode se tornar irregular, com atrasos variados entre cada pacote, conforme se pode visualizar na Figura 2-5. Quando um roteador recebe um fluxo de voz sobre IP, por exemplo, ele deve compensar a variação encontrada, armazenando esses pacotes em *buffer* (ou "*playout buffer delay*") e, depois, jogando-os, num fluxo estável, para o processador de sinais digitais (*Digital Signal Processor* ou DSP), onde será convertido para um fluxo analógico (Figura 2-6). Caso o jitter seja grande o suficiente para fazer com que pacotes sejam recebidos fora do limite do buffer, pacotes serão descartados e falhas serão escutadas no áudio. No entanto, pequenas taxas de perda podem ser, através de técnicas de interpolação de pacotes, recuperadas pelo DSP, sem prejuízo da escuta.

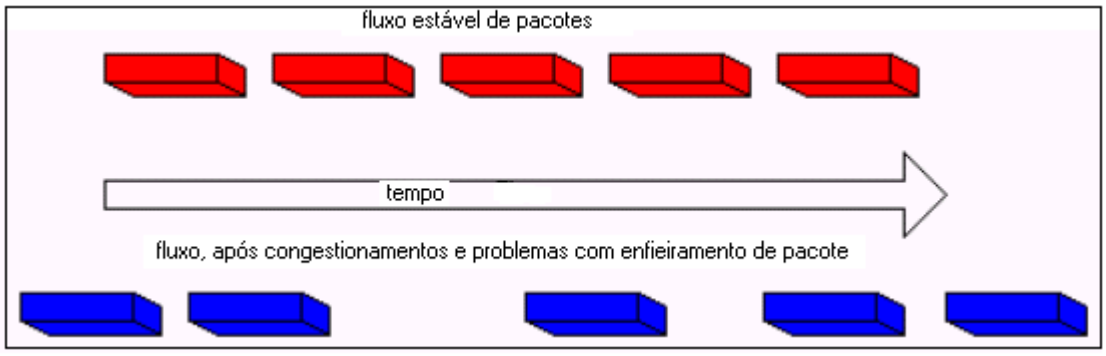


Figura 2-5 – Ocorrência de variações de delays entre pacotes.

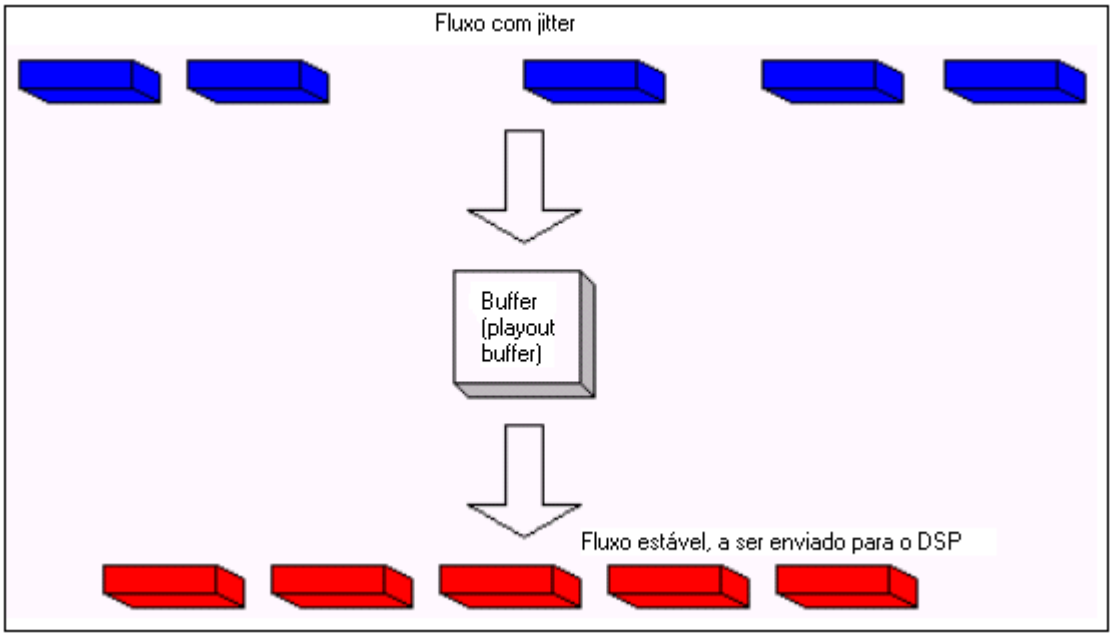


Figura 2-6 – Atuação do buffer na estabilização dos fluxos com jitter.

Como visto na sessão anterior, a perda de pacotes pode causar problemas sérios às aplicações de tempo real, como voz sobre IP, por exemplo, comprometendo a qualidade dessas aplicações. Por causa disto, técnicas de recuperação por redundância, como o FEC (*Forward Error Correction*), tentam minimizar a taxa de perda de quadros de voz, associada a uma certa taxa de perda de pacotes[30]. Protocolos como TCP se recuperam bem desses problemas, uma vez que efetuam retransmissão de pacotes em caso de perda. O protocolo UDP, mais comumente utilizado em transmissões de voz e vídeo, por sua vez, não recupera perdas e, desta forma, tende apresentar maior quantidade de pacotes perdidos.

O parâmetro estudado neste trabalho que se relaciona com as perdas é a "taxa de perda de pacotes", calculado, no lado do receptor, como a razão entre a quantidade de pacotes perdidos e a quantidade de pacotes transmitidos, dentro do período de análise[9].

2.6.3 Largura de banda e Vazão

Largura de banda de um *link* representa a capacidade máxima de transmissão de uma conexão, sendo comumente expressada em kilobits por segundo (Kbps) ou megabits por segundo (Mbps).

[9] Percentagem calculada através da fórmula: **Taxa de perda = (Nº de pacotes perdidos/ Nº de pacotes transmitidos)*100**.

A largura de banda pode ser entendida a partir da analogia com os canos d'água, com um certo diâmetro de espessura. Quanto maior for este diâmetro, maior será a quantidade de água por segundo que poderá passar através do cano. Aqui, no caso, o diâmetro é a largura de banda, o cano representa a ligação de rede e, a água, significa os dados.

Vazão é, justamente, a quantidade de dados que a rede é capaz de transmitir de um lado a outro, num certo período de tempo. Voltando à analogia, vazão seria a quantidade de água por segundo que passa através do cano. A vazão também é expressa em Kbps ou Mbps.

2.7 Conclusão do capítulo

Este capítulo mostrou aspectos básicos, mas imprescindíveis, sobre QoS. Foram apresentadas justificativas e questões restritivas para a implantação da qualidade de serviço nas redes de computadores. Destacou-se a importância do tema, que é alvo de estudos e pesquisas crescentes, responsáveis pelo estabelecimento de padrões internacionais e pela disponibilização de produtos de mercado.

Mostrou-se ainda a necessidade da classificação das aplicações para a determinação dos modelos de QoS (IntServ e DiffServ), os quais foram abordados dentro de uma visão crítica, com a apresentação de suas vantagens e desvantagens.

O capítulo foi finalizado com a definição das métricas essenciais para a pesquisa de QoS: atraso dos pacotes, a variação desses atrasos, o descarte de pacotes, a vazão de dados e a largura de banda.

3. Serviços Diferenciados (DiffServ)

Complementam-se aqui as definições apresentadas no capítulo anterior, abordando, dentre outras coisas, questões como o condicionamento de tráfego no modelo diferenciado (suas funções e seus mecanismos), o conceito de comportamento por nó, a implementação DiffServ no Linux e as aplicações de voz sobre IP.

3.1 Condicionamento de Tráfego no modelo DiffServ

O tráfego que ingressa numa rede com serviço diferenciado deve ser submetido a uma série de elementos funcionais, os quais são mantidos pela arquitetura e responsáveis pelo condicionamento daquele tráfego. Ou seja, os fluxos são classificados, condicionados, marcados e, então, associados a diferentes tipos de agregados (BA), através da gravação do valor DSCP correspondente. Percebe-se então que o modelo diferenciado adota um conjunto de funções, que são implementadas nos nós da rede. Incluem-se aí, além dos classificadores, medidores, marcadores, suavizadores e policiadores (ou descartadores), como se pode visualizar na Figura 3-1:

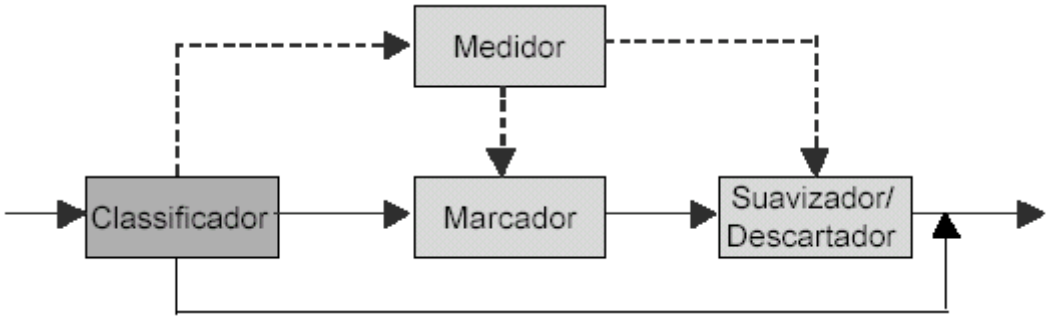


Figura 3-1 – Elementos de condicionamento de tráfego

O classificador é o primeiro elemento de condicionamento de tráfego e é responsável pela seleção dos pacotes, baseada no conteúdo de parte(s) do cabeçalho. Se tal seleção ocorrer na entrada de uma rede DS (ou seja, o pacote é originário de uma rede sem serviço diferenciado), a classificação pode ser avaliada em vários campos do pacote (endereço IP de destino, ToS e protocolo de transporte, por exemplo). Chama-se esse tipo de classificação de **multicampo** (*Mult-Field ou* MF). Caso a classificação seja baseada apenas no campo DSCP, ou seja, o pacote já veio marcado de um domínio DS, com o

campo DSCP correspondente a um determinado tipo de tráfego, a classificação é conhecida então como de **comportamento agregado** (*Behavior Aggregate* ou BA, como já citado anteriormente).

O medidor é responsável por verificar a conformidade do tráfego em relação ao perfil contratado, com base na medição de propriedades temporais dos fluxos de pacotes. Balde de fichas é uma das implementações de medidores conhecidas. A partir do resultado da medição, será tomada sobre o pacote uma ação de condicionamento, que pode ser marcação (ou remarcação), suavização ou descarte.

O marcador rotula o pacote, redirecionando-o para um determinado fluxo. É responsável também em realizar a remarcação dos pacotes (aumentando ou diminuindo sua prioridade). Atrasar os pacotes até que estes estejam em conformidade com o perfil contratado e possam ser liberados para transitarem na rede é a função do suavizador. Os pacotes que foram considerados pelo medidor como em "não conformidade" com o perfil contratado, podem ser descartados pelo policiador, que pode ser implementado como um caso especial de suavizador, sem *buffer* de espera[31].

3.1.1 Funções de condicionamento de tráfego

As funções de condicionamento de tráfego são realizadas pelo modelo DiffServ dentro do chamado domínio de serviços diferenciados (ou simplesmente domínio DS, exemplificado na Figura 3-2), que é composto por um conjunto de nós compatíveis com a arquitetura diferenciada. O domínio DS, geralmente, apresenta nós de entrada (INGRESSO), de saída (EGRESSO) e internos, sendo que os de entrada e saída são dados como nós de fronteira, já que atuam como pontos de demarcação entre domínios diferenciados ou entre estes e domínios não compatíveis com o modelo. Tipicamente, os nós de borda[10] realizam condicionamento de tráfego, classificando os pacotes que chegam, num agregado pré-definido; medindo-os para determinar a conformidade dos perfis de tráfego; marcando o campo DSCP apropriadamente, atrasando o tráfego quando necessário e até descartando os pacotes, em caso de congestionamentos. Realizam, na verdade, as funções mais complexas das executadas num domínio DS, uma vez que os nós interiores somente aplicam o comportamento (PHB) apropriado, empregando técnicas de

[10] Como também são conhecidos os nós de fronteira, representados, na Figura 3-2, pelos nós 1, 4 e 5.

policiamento ou suavização, e, algumas vezes, praticam remarcação de pacotes, dependendo da política do domínio. São as diferentes funcionalidades exercidas pelos roteadores, num domínio DS.

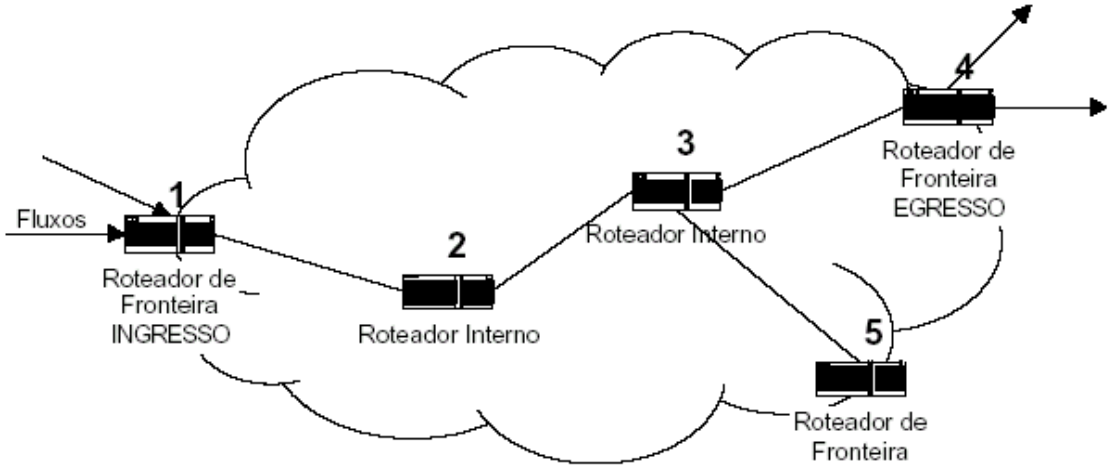


Figura 3-2 – Domínio de Serviços Diferenciados

3.1.2 Mecanismos de medição e policiamento de tráfego

A marcação e o policiamento de pacotes compartilham a função de medição, para determinar se cada pacote está ou não em conformidade com o perfil contratado (*in profile* e *out profile*, respectivamente). Um simples exemplo é o clássico método de balde de fichas (*token bucket*), que força um padrão de saída constante, mesmo em casos de rajadas de tráfego. O balde retém uma quantidade limitada de fichas (*tokens*), a que se chama tamanho do balde (***bucket size***), e essas fichas são geradas em ciclos de relógio do sistema operacional (*clocks*), a cada intervalo constante de tempo (Δt segundos), numa certa **taxa** de renovação, e removidas do balde à medida que os pacotes de rede chegam. Na verdade, cada ficha representa a permissão de enviar um certo número de bits na rede, sendo que, para transmitir um pacote, um número de fichas correspondendo ao tamanho daquele pacote deve ser removido do balde. Ou seja, para cada pacote que chega é verificado se há fichas disponíveis para ele. Havendo, elas são removidas e o pacote é considerado dentro do perfil. Por outro lado, se não houver qualquer ficha disponível no balde quando o pacote chegar, considera-se que o mesmo está fora do perfil, podendo ser reclassificado ou até mesmo descartado pelo mecanismo. Neste caso, quando há descarte de

pacotes, o *token bucket* funciona como um policiador de tráfego. A Figura 3-3 ilustra o mecanismo, que, para facilitar o entendimento, relaciona apenas uma ficha para cada pacote.

Como o tamanho do balde é finito e fixado pelo usuário, pequenas rajadas de rede podem ser toleradas, com pacotes pertencentes a essas rajadas sendo ainda considerados dentro do perfil. Na verdade, a seleção dos parâmetros “tamanho do balde” e “taxa de renovação de fichas” é que vai tanto determinar o perfil, forçando uma taxa média de encaminhamento de pacotes, quanto controlar o tamanho das rajadas permitidas.

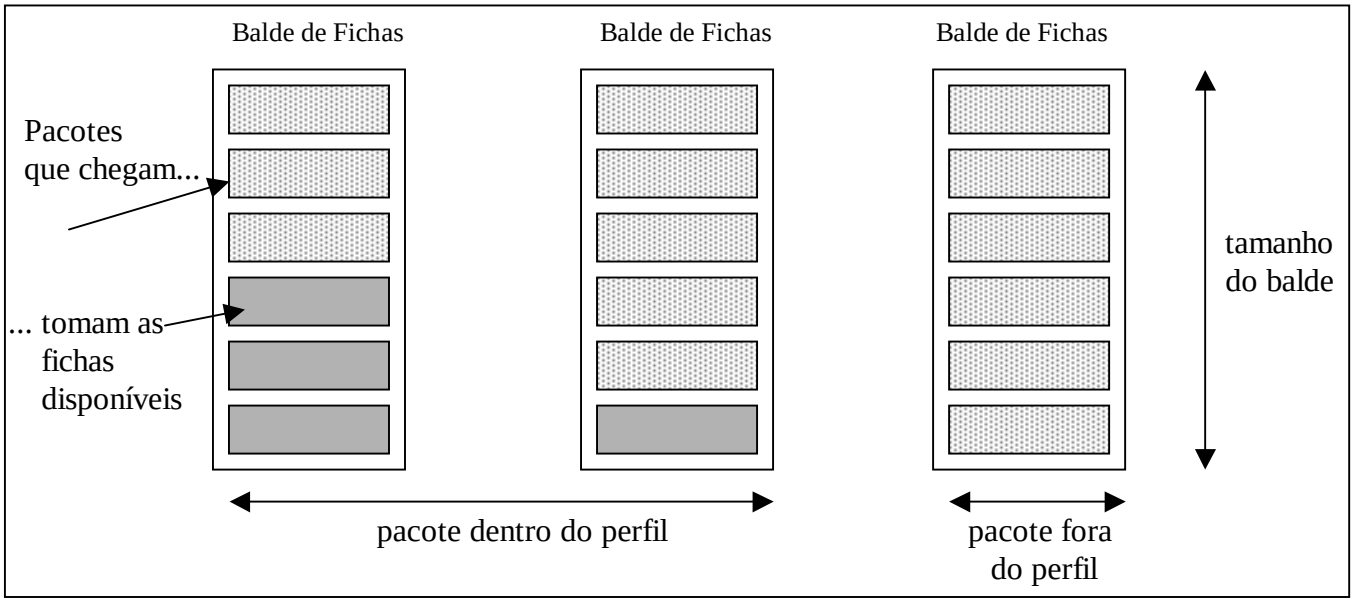


Figura 3-3 – Método do Balde de Fichas

Outra forma utilizada para medição e policiamento de tráfego é o de balde furado (*leaky bucket*), que consiste em encaminhar os pacotes para escalonamento numa mesma freqüência (taxa fixa de transmissão), não importando o intervalo de chegada entre eles e nem se chegam em rajadas. Ou seja, estabiliza a transmissão de fluxos de pacotes irregulares, diminuindo as chances de congestionamento nos nós posteriores.

O método acima funciona como um suavizador de tráfego, limitando a freqüência com que uma fila será servida, mesmo quando o escalonador esteja ocioso.

O balde furado apresenta um *buffer* (o *bucket*) para depositar os pacotes que chegam. Como esse tem comprimento limitado e, portanto, em momentos de rajadas maiores pode receber mais pacotes do que sua capacidade, “transbordando” o balde, o

mecanismo age também como policiador de tráfego, pois pacotes que chegarem com o *buffer* cheio certamente serão descartados.

3.2 Acordo de nível de serviço (SLA)

Acordo de nível de serviço, no contexto deste trabalho, é definido como um contrato entre provedores de serviços (provedor Internet ou ISP, por exemplo) ou entre um provedor de serviço e seus clientes. SLA especifica, geralmente em termos mensuráveis, o nível de qualidade de serviço que pode ser esperado ou, de outra forma, pode ser visto como um contrato que determina quais serviços o provedor de serviço irá fornecer e quais penalidades o mesmo irá sofrer, caso não cumpra com os compromissos firmados.

Na especificação dos serviços podem constar parâmetros relacionados a níveis de disponibilidade, de desempenho, de operação ou outros atributos do serviço, como suporte ao usuário e credibilidade.

Em termos de métricas de desempenho, são geralmente caracterizados parâmetros como os de tempo de resposta, vazão, atrasos, perdas e utilização dos sistemas. Métricas de vazão, por exemplo, definem as taxas com que os dados são liberados ao cliente, tal como: “no horário de trabalho, usuários da Intranet deverão carregar arquivos de imagem (.gif), de até 65Kbytes, em, no máximo, 10 segundos”. Já a métrica de tempo de resposta define o tempo máximo de resposta de uma aplicação, frente às requisições de um usuário, expressando acordos como: “90% dos usuários deverão experimentar um tempo de resposta de, no máximo, 2 segundos durante os horários de pico (10 às 12h e 16 às 18h)”.

Para facilitar especificações de QoS dentro de um contrato, um SLA deve conter, visando descrever o serviço, um conjunto de parâmetros mais técnicos, conhecidos como especificações de nível de serviço (SLS ou *Service Level Specifications*). Tais parâmetros caracterizam, por exemplo, na visão do serviço diferenciado, perfis de tráfegos agregados e as formas de encaminhamento de cada agregado de fluxos. O SLS pode especificar ainda, segundo [36], outros requisitos, como:

- ✓ mecanismos de autenticação e criptografia utilizados;
- ✓ ações (comandos) a serem executadas em caso de descontinuidade dos serviços;

- ✓ monitoração e auditoria do QoS provido e;
- ✓ restrições no roteamento de tráfego agregado.

Num ambiente de rede em que vários limites são atingidos (situação com interdomínios diferenciados, por exemplo), os SLAs podem ser gerenciados de duas maneiras distintas: estático ou dinamicamente. Se o número de serviços oferecidos é pequeno, os SLSs entre os domínios podem ser manualmente negociados e os roteadores de borda configurados por administradores de rede. Isto significa que o contrato SLA é feito estaticamente, entre partes humanas, e que seus termos não podem ser alterados sem uma intervenção do administrador (ver Figura 3-4). Entretanto, em redes muito extensas, como a Internet, é mais viável, entre os domínios, a implementação de mecanismos dinâmicos para manipular a sinalização de QoS e a provisão de recursos. Ou seja, são entidades que automatizam o processo de negociação de SLS e controle de admissão, de modo a configurar corretamente os dispositivos de rede e suportar os serviços de QoS acordados. Essa entidade é o negociador de largura de banda (BB ou *Bandwidth Broker*), responsável em garantir que os recursos dentro dos domínios diferenciados e em ligações conectando domínios adjacentes sejam corretamente alocados (ver Figura 3-5).

Um BB mantém informações relativas aos SLSs definidos entre o domínio diferenciado e seus clientes[11], e utiliza estas informações para gerenciar os recursos de rede, configurar os roteadores do domínio local e tomar decisões relativas ao controle de admissão.

Para verificar se uma solicitação de recursos de um usuário pode ser atendida através dos domínios, é preciso que o BB do domínio do usuário se comunique com os BBs dos demais domínios. Esta comunicação é geralmente implementada através de sinalização, utilizando protocolos para este fim, como o RSVP.

A implementação de gerenciamento dinâmico de SLA em redes extensas é, como se pode imaginar, bastante complexa. Para enfrentar este desafio, a iniciativa **Internet2 QBONE**[37] tem agrupado redes de pesquisa, grupos de universidades, engenheiros, pesquisadores, desenvolvedores de aplicação e redes de agências federais para

[11] O termo “clientes” inclui os usuários locais do domínio em que o BB se encontra, bem como as redes adjacentes.

construir um testbed em interdomínios com serviços diferenciados. Este foi, apesar de grandes dificuldades ainda encontradas para implementar o projeto[12], o primeiro esforço dessa natureza em testar o serviço diferenciado em redes geograficamente remotas e a primeira experiência com protocolos de sinalização em interdomínios DiffServ.

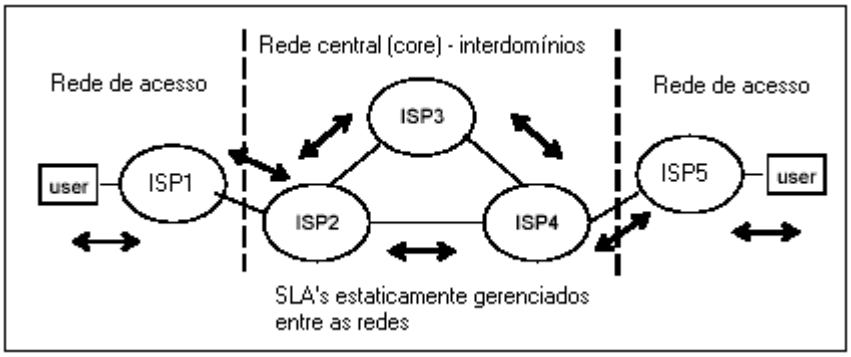


Figura 3-4 – SLA estático

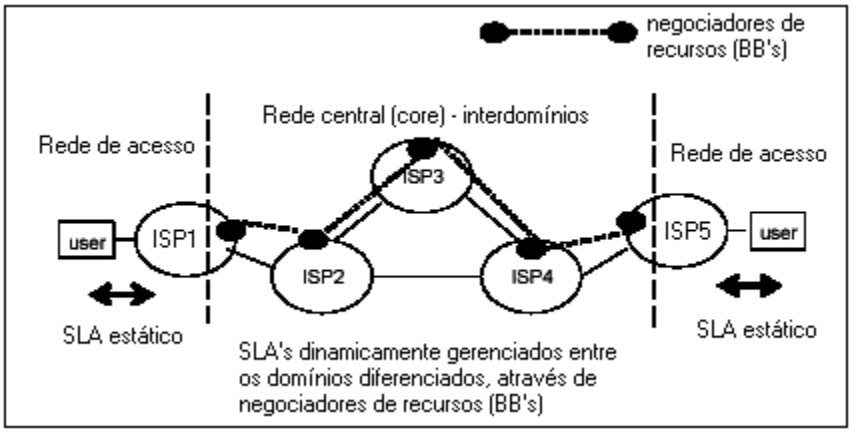


Figura 3-5 – SLA dinâmico, com uso de BBs

3.3 Per-Hop Behavior (PHB)

Um dos conceitos mais importantes em serviços diferenciados é o de comportamentos por nó (*Per-Hop Behavior* – PHB), definido como um tratamento, observado externamente, que é dado aos pacotes de um certo agregado[32] de fluxos, marcados com o mesmo código DS (DSCP) e com o mesmo tratamento no encaminhamento dos pacotes. O PHB define o tratamento a ser oferecido ao pacote, até quando este for encaminhado ou descartado em cada nó, sendo selecionado por meio de um mapeamento entre o DSCP (que identifica a agregação) e o tratamento que os agregados devem receber no domínio DS. Os fluxos pertencentes a uma agregação serão servidos,

[12] Um dos serviços originalmente projetados foi o *Internet2 Qbone Premium Service*, que se encontra atualmente inativo, em função de barreiras, como: complexidade no policiamento dos roteadores em cada

num domínio DS, de acordo com a combinação de todos os PHBs concedidos a eles ao longo do caminho percorrido no interior do domínio[33]. Ou seja, o resultado de uma transmissão (o serviço fim a fim) é conseguido através da combinação dos PHBs entre os domínios, ao longo do percurso[17]. De uma outra forma ainda, os comportamentos (PHBs) individuais aplicados em cada roteador, por si só, não garantem uma qualidade de serviço fim-a-fim, mas conseguem fazê-lo se forem aplicados aos roteadores do domínio DS.

Além do *PHB default*, que serve para representar o tráfego de melhor esforço e garantir a compatibilidade em todos os roteadores, dois PHBs são padronizados pelo IETF: o de **encaminhamento assegurado** (PHB AF – *Assured Forwarding*)[35] e o de **encaminhamento expresso** (PHB EF – *Expedited Forwarding*)[34]. O primeiro, também conhecido como *serviço premium* ou de canal dedicado virtual, pode ser usado para tráfego com requisitos de baixa perda, baixo atraso, baixo jitter (variação dos atrasos) e garantia de largura de banda, alcançados desde que, aos fluxos agregados sejam assegurados pouco ou nenhum enfileiramento. Para isso, ou seja, para garantir pouca ocupação nas filas com tráfego EF, é requerido que a taxa de serviço dada ao agregado de pacotes EF, na saída do roteador, exceda sua taxa de entrada sobre longos e curtos intervalos de tempos, independentemente da carga de outros tipos de tráfego[34]. O PHB EF é propício para tráfego de tempo real (como VoIP) e, por isso, foi aplicado nos testes aqui explanados. O PHB AF, por sua vez, baseia-se numa expectativa de serviço, concebida em momentos de congestionamento, sem garantias de que os requisitos (perdas de pacotes, por exemplo) sejam atingidos. Com este comportamento podem ser providas 04 classes independentes, cada uma com 03 níveis de descarte, o que conduz ao total de 12 diferentes códigos AF (DSCPs).

3.3.1 Encaminhamento Assegurado (PHB AF)

O comportamento AF provê classes distintas de tráfegos, com variação de níveis de probabilidades de perda. Desta forma, como ligeiramente sinalizado acima, dois distintos contextos de classificação são codificados dentro do DSCP: a classe de serviço do pacote e seu respectivo nível de precedência para descarte. Ou seja, são disponibilizadas 4

---

domínio, carência na disponibilidade de aplicações e custos operacionais.

classes diferentes, tendo, cada uma, 3 níveis de precedência, correspondendo, dentro de cada classe, a uma maior ou menor probabilidade de perda[13]. Desta forma, um roteador congestionado vai dar mais prioridade aos pacotes, de uma certa classe, que tiverem os menores valores de precedência de descarte.

A partir da Tabela 3-1, dos bits **CCCDD0**, CCC representa a codificação da classe de serviço, que irá determinar a fila de transmissão do pacote, e DD codifica o nível de precedência do pacote dentro das classes.

0	1	2	3	4	5	**DSCP**
C	**C**	**C**	**D**	**D**	0	

Tabela 3-1 – Códigos AF para classe de serviço e nível de precedência

A Tabela 3-2 mostra, para cada grupo PHB AF, os valores dos DSCPs recomendados pela RFC 2597[35].

Precedência de descarte	**Classe AF1**	**Classe AF2**	**Classe AF3**	**Classe AF4**
Nível 1 (baixa)	001010	010010	011010	100010
Nível 2 (média)	001100	010100	011100	100100
Nível 3 (alta)	001110	010110	011110	100110

Tabela 3-2 – Valores DSCP definidos pela RFC 2597

Para cada uma das classes, são alocados, nos roteadores, certos níveis de recursos de rede, como largura de banda e armazenamento (*buffers*). Congestionamentos ocorrem quando os roteadores recebem uma carga de tráfego maior do que sua capacidade de tratá-los, o que poderá exceder os limites de armazenamento e levar a um cenário em que o roteador deverá decidir quais pacotes serão descartados. Nesses casos, é o nível de precedência do pacote que irá determinar a probabilidade de seu descarte.

O condicionamento de tráfego é o responsável pela marcação do pacote em um dos doze códigos recomendados pelo IETF, de acordo com o resultado da medição. Um perfil associado a cada tráfego define o serviço esperado por ele. Se, na chegada do domínio DS, os pacotes estiverem em conformidade com o perfil para eles contratado, os mesmos terão, enquanto a agregação não exceder a taxa contratada definida no perfil de serviço, alta probabilidade de ser entregues. Esses pacotes usarão recursos suficientes e serão marcados com níveis baixos de precedência para descarte. Caso contrário, ou seja, se

[13] O nível de precedência determina a importância do pacote.

os pacotes não estiverem em conformidade com o perfil contratado[14], eles provavelmente serão rebaixados no nível de classificação, sendo marcados com níveis de descarte mais elevados e, portanto, terão menores probabilidades de serem encaminhados. Dependendo de suas marcações atuais, os pacotes já serão imediatamente descartados. Mecanismos de descarte de pacote baseados em níveis de precedência e em classes, como o GRED[38], são adequados para a implementação dos grupos AF.

3.3.2 Encaminhamento Expresso (PHB EF)

O PHB EF é a versão de DiffServ para encaminhar tráfego de aplicações multimídia e de tempo real. Neste tipo de serviço não há reclassificação de pacotes, como no serviço assegurado. Ou seja, os pacotes que não estiverem dentro do perfil contratado serão obrigatoriamente descartados, sem possibilidade de remarcação.

Como já abordado inicialmente nesta sessão, o serviço expresso caracteriza-se pelos baixos valores de perda, jitter e atraso, o que significa dizer que, em cada nó do domínio DS, os pacotes agregados encontrarão as filas dos roteadores vazias ou com pequenas taxas de ocupação. Para alcançar isto, o nó com PHB EF deve garantir que a agregação de fluxos receba, em qualquer instante, uma taxa de chegada inferior à taxa contratada, independentemente da intensidade de outros tráfegos que cheguem ao nó.

Para permitir a essência do serviço expresso no interior de um domínio DS, como no da Figura 3-2, ou seja, que a taxa de chegada da agregação EF aos nós internos desse domínio esteja em conformidade com a taxa que foi contratada, os roteadores de entrada (INGRESSO) têm papel fundamental. São eles que precisam condicionar o tráfego expresso que chega ao domínio DS, de forma a minimizar recondicionamentos naqueles pontos internos EF e permitir, conseqüentemente, que a definição do PHB seja válida a qualquer instante e em todo o percurso dos fluxos. Engenharia de tráfego, às vezes, é necessária, porque tráfego agregado, vindo de vários roteadores de fronteira de entrada, se aglomera nos roteadores internos, formando novas agregações.

[14] Exceder a largura de banda a eles associada, por exemplo.

O encaminhamento expresso pode ser implementado num domínio DS através da aplicação de disciplinas de escalonamento com enfileiramento prioritário (*Priority Queuing* – PQ), com limitação de taxas nas classes. Nesse caso, é muito importante o controle de tráfego na fronteira do domínio, para evitar que fluxos EF, injustamente, excedam na utilização dos recursos, em detrimento à atuação do tráfego menos prioritário (como os de melhor esforço e classes AF). E isto pode ocorrer, principalmente, se a disciplina aplicada for a prioritária de cunho estrito. Escalonadores, como PQ, serão abordados mais adiante.

3.4 Diffserv em ambiente Linux

O núcleo do sistema operacional Linux já inclui um conjunto flexível de funções de controle de tráfego, tornando-o propício para a realização de trabalhos experimentais na área de qualidade de serviço. O princípio básico envolvido no suporte a essas funções pode ser observado na Figura 3-6, em que o núcleo do sistema processa os dados recebidos da rede e gera novos dados para serem encaminhados.

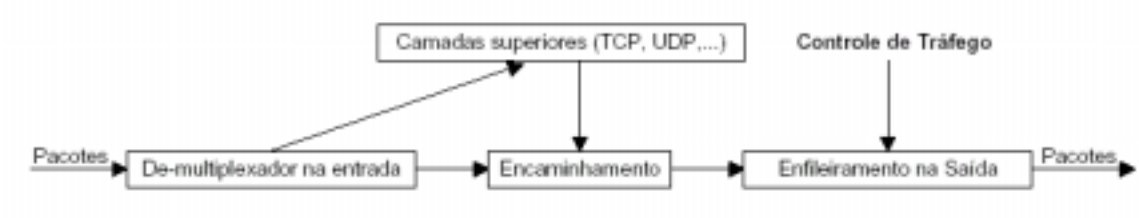


Figura 3-6 – Princípio básico de controle de tráfego no Linux

A função de "encaminhamento" é a responsável em selecionar a interface de saída aos pacotes que entram no nó. Quando enfileirados, os pacotes passam a ser monitorados pelo módulo de controle de tráfego (aplicação TC – *Linux Traffic Controller* [38][39]), que pode decidir, por exemplo, se os pacotes serão descartados; quais suas prioridades no encaminhamento e quando enviá-los (motivado pela função de suavização do tráfego).

Quatro componentes enfocam a arquitetura de controle de tráfego no Linux:

- ✓ Disciplinas de serviço (ou *qdisc*);
- ✓ Classes;

- ✓ Filtros;
- ✓ Policiamento.

Muito sinteticamente, já que não é o objetivo deste trabalho aprofundar esses conceitos, uma disciplina de serviço deve estar sempre associada a um dispositivo de rede. As mais complexas disciplinas, podem, no entanto, conter classes, tendo os filtros como os responsáveis pela atribuição dos pacotes às classes correspondentes e pela atribuição de prioridades de uma classe em relação às outras. O controle de tráfego em Linux é dito flexível porque, a princípio, as classes não obrigatoriamente mantêm as filas. Elas podem anexar outra disciplina de serviço[15], que, por sua vez, pode, por exemplo, ter outra classe e esta, finalmente, manter uma fila de pacotes. É uma hierarquia de componentes, como se pode observar na Figura 3-7.

O policiamento pode, sobre os fluxos que excederem os limites, manipular pacotes nas filas, de acordo com determinadas ações. Tais ações podem ser a eliminação e a remarcação de pacotes.

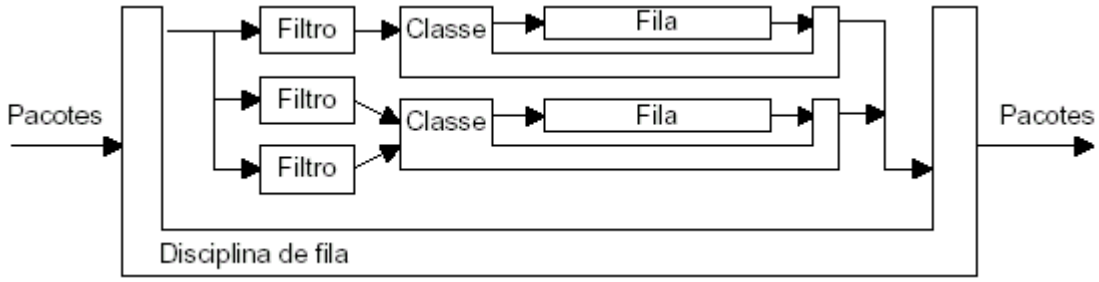


Figura 3-7 – Exemplo de uma simples disciplina de serviço no Linux, com várias classes.

3.4.1 Disciplinas de escalonamento

Serão sucintamente abordados, a seguir, aspectos de alguns tipos de escalonadores de pacotes, que são, inclusive, implementados no código do Linux e utilizados na plataforma de testes deste trabalho. As sintaxes e os detalhes de uso dessas disciplinas no Linux são amplamente abordados em [41] e [42].

[15] Disciplina de serviço interior.

a) *First In First Out* (FIFO)

Conforme [40], enfileiramento do tipo FIFO é a mais básica disciplina de escalonamento, onde todos os pacotes são tratados da mesma forma (sem prioridades de uns sobre os outros), sendo os mesmos colocados numa única fila e servidos na mesma ordem em que são enfileirados.

O enfileiramento FIFO tem como vantagens a simplicidade e uma baixa carga computacional. No entanto, não toma decisões sobre prioridades dos pacotes e, por isso, é impróprio para aplicações de tempo real, uma vez que, em casos de congestionamentos, tende degradar as métricas de QoS (atrasos, jitters e perdas de pacotes) dessas aplicações.

No Linux, a disciplina FIFO é apresentada em duas versões: *pfifo* e *bfifo*. No primeiro caso, a fila é limitada pelo número de pacotes e, na segunda opção, pelo número total de bytes de todos os pacotes na fila.

b) Enfileiramento baseado em classes (CBQ – *Class-Based Queuing*)

O CBQ provê o particionamento e o compartilhamento da largura de banda disponível num *link*, através do uso de classes hierarquicamente estruturadas, onde cada classe terá sua própria fila e receberá uma "fatia" da banda. Ou seja, os pacotes pertencentes às aplicações são, baseados num esquema de pesos e prioridades, classificados em várias classes de serviço e, então, associados a uma fila, que é especificamente dedicada àquela classe de serviço. Para as classes com a mesma prioridade, as respectivas filas são servidas na ordem conhecida como *round-robin*, em que, para transmissão, as filas são percorridas de forma circular. É transmitida então, a partir de cada fila, uma quantidade de tráfego proporcional ao peso que lhe foi atribuído. Desta forma, pode-se garantir, com CBQ, a maior parte da largura de banda do link a certas aplicações, consideradas mais críticas, com a parte remanescente sendo disponibilizada ao restante do tráfego. Além disso, na estrutura hierárquica, uma classe filha pode emprestar largura de banda da classe mãe, desde que haja banda disponível.

O CBQ é uma solução consideravelmente razoável quanto ao compartilhamento de banda passante pelas classes de serviço. A possibilidade de se particionar, de forma hierárquica, o tráfego em tipos distintos de classes de serviço, torna a disciplina *cbq* a mais flexível das atualmente implementadas no núcleo do Linux[42]. No entanto, o CBQ não é um método simples de enfileiramento e, por isso, apresenta problemas de sobrecarga computacional. Na implementação do Linux, por exemplo, isso é percebido, especificamente, durante as atividades de classificação de pacotes, de determinação das filas a serem servidas e dos cálculos de estimativa de banda a ser usada pelas classes do *tc*, o que pode gerar degradação das métricas de qualidade de serviço.

c) Enfileiramento justo do tipo SFQ (*Stochastic Fairness Queuing*)

O SFQ é uma simples implementação da família de algoritmos de enfileiramento justo, sendo projetado para garantir que cada fluxo tenha um acesso justo aos recursos da rede e, também, para evitar que fluxos variáveis (em rajadas) consumam mais largura de banda que os demais. Ou seja, o SFQ não visa diferenciação de serviço entre os pacotes. Ao contrário, distribui eqüitativamente a largura de banda aos fluxos do *link*. Logo, teoricamente, o SFQ não seria próprio para tráfego de voz, o que poderá ser confirmado mais adiante, nos testes experimentais realizados em laboratório.

Com SFQ, os pacotes são inicialmente classificados em fluxos pelo sistema e, depois, associados a uma fila, do tipo FIFO, que passa a ser dedicada àquele fluxo. As filas ativas[16] são servidas de forma circular (*round-robin*), sendo transmitido um certo volume de dados de cada fila em cada passagem do algoritmo. O SFQ é chamado “estocástico” porque ele implementa algoritmo com função hash, que divide o tráfego num número limitado de filas e as associa aos pacotes de cada fluxo.

A vantagem da disciplina é que, dentre os escalonadores justos, é a que requer menos cálculos[41], imprimindo uma baixa carga computacional. No entanto, o SFQ apresenta limitações. Uma delas é a possibilidade de que a função *hash* possa selecionar a mesma fila para mais de um fluxo (colisões), prejudicando a eqüidade da disciplina. Uma vez que isso aconteça, os fluxos selecionados para o mesmo valor de *hash* serão tratados

injustamente, como se fossem um fluxo apenas. Isto ocorre quando o número de fluxos ativos se aproxima do número de filas disponibilizado pela disciplina, que, no Linux, é limitado em 128 filas.

d) Enfileiramento prioritário (*Priority* Queuing - PQ)

O escalonamento por prioridades, como o nome mesmo sugere, foi desenhado para que, dentre várias filas, o encaminhamento seja realizado a partir da atribuição de prioridades a cada uma delas. Trata-se de um método simples para suportar serviço diferenciado, em que os pacotes são inicialmente classificados e associados a filas com diferentes níveis de prioridade.

Há duas implementações desta disciplina: a primeira é a **estrita**, em que somente serão transmitidos pacotes de uma certa fila quando todas as de prioridade superior estiverem ociosas (vazias); a segunda é a **controlada por taxa**[40], a qual permite que pacotes de uma fila mais prioritária somente sejam encaminhados antes que os de uma fila com menor prioridade quando o volume de tráfego da mais prioritária estiver abaixo de um certo limite configurado pelo usuário. Usou-se, nos estudos experimentais, o enfileiramento prioritário estrito, que apresenta como desvantagem a possibilidade de fluxos mais prioritários alocar todo o recurso de banda disponível no *link*, impedindo, ou pelo menos dificultando, a transmissão de fluxos de menor prioridade. No entanto, o PQ tem a vantagem de, por efetuar apenas classificação de pacotes, apresentar baixa carga computacional, sendo reconhecido como um algoritmo simples e eficiente na diferenciação de tráfego.

No Linux, a disciplina prioritária é conhecida como *prio*, a qual implementa um escalonador de prioridade estrita, com um número de bandas (filas)[17] configurável pelo usuário. Por *default*, na construção da disciplina, 03 classes automaticamente são criadas e, em cada classe dessa, uma fila FIFO é anexada[18]. Quando um pacote chega numa *interface* de rede com a disciplina *prio*, há o processo de classificação e o mesmo é encaminhado para uma das filas FIFO criada. Para extrair um pacote da fila, as classes mais prioritárias serão

[16] As filas que têm pacotes pendentes para serem transmitidos.
[17] Quanto maior a identificação da banda, menor será a prioridade.

sempre varridas primeiro e as bandas maiores somente serão visitadas quando não houver pacotes nas bandas mais prioritárias. No entanto, se um pacote menos prioritário estiver sendo servido no momento da chegada de um mais prioritário, este deverá aguardar a transmissão do primeiro.

A Figura 3-8 ilustra a estrutura automaticamente construída no Linux após a inclusão de uma disciplina de fila *prio*, na qual se tem 3 bandas, cada qual correspondendo a uma fila, sendo a banda superior a mais prioritária e a inferior a com menor prioridade para ser servida.

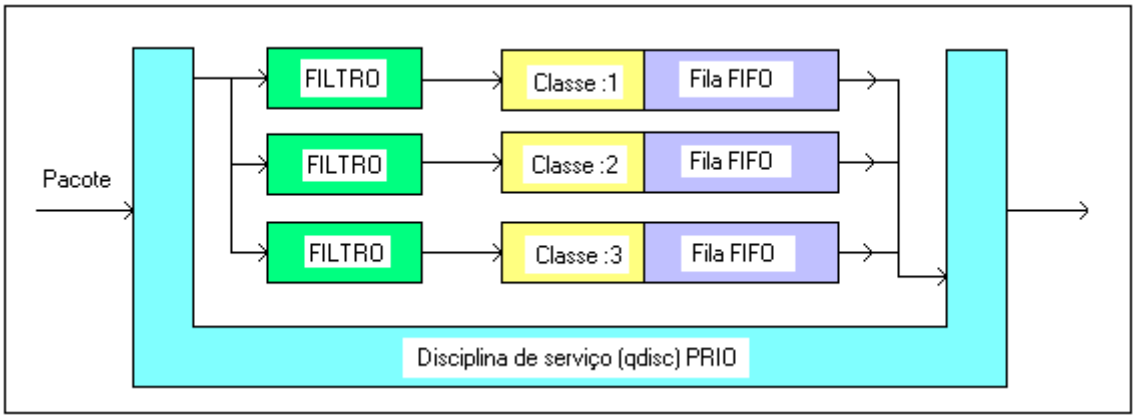


Figura 3-8 – Estrutura básica da disciplina *prio* no Linux

3.5 Voz sobre IP como uma aplicação diferenciada

O entendimento do tipo de tráfego gerado e o comportamento das aplicações são aspectos importantes para se determinar o modelo de QoS a ser disponibilizado aos usuários.

Voz sobre IP é uma aplicação sobre a qual ainda é gerada uma grande expectativa, considerando que já se consegue reduzir custos com telefonia e proporcionar melhores qualidades no uso da voz em redes locais e remotas. Para isso, várias técnicas de codificação para telefonia e para pacotes de voz têm sido padronizadas pelo ITU-T na série G, com taxas de geração que podem variar de 5.3Kbps[19] a 64Kbps[20], dependendo do tipo de codificador/decodificador de voz (vocoder) utilizado. Dentre esses, pode-se mencionar

[18] Pode-se substituir as filas FIFO por outras estruturas de enfileiramento.
[19] Taxa gerada, para um único fluxo, por codificador de voz do tipo G.723.1.
[20] Taxa gerada, para um único fluxo, por codificador de voz do tipo G.711.

aqui[21] o G.711, que, para comprimir, descomprimir, codificar e decodificar a fala, usa o algoritmo PCM (*Pulse Code Modulation*), responsável em gerar amostras de 8 bits a cada 0.125ms[22], o que leva a uma taxa de 64Kbps[23]. É padrão também o G.726, que usa o algoritmo ADPCM (*Adaptive Differential Pulse Code Modulation*) para codificar um fluxo de bits G.711 em palavras de dois, três ou quatro bits para gerar taxas de 16, 24, 32 ou 40Kbps[30, 43]. Recentes codificadores promovem uma drástica redução na taxa de geração, às custas de adicional complexidade e atrasos na codificação, além de apresentar baixa qualidade[44]. Como exemplos, tem-se o G.729, com taxas de 8Kbps, e o G.723.1, que gera os pacotes a uma taxa de 6.4Kbps. Há vários vocoders disponibilizados atualmente, haja vista serem utilizados apropriadamente para lidar com as mais adversas condições da rede. Pode-se, por exemplo, adotar um certo vocoder num *link* que apresente altas taxas de perda ou atrasos elevados.

Cada vocoder fornece som com uma certa qualidade, que é subjetiva porque depende da preferência de cada pessoa. Desta forma, a avaliação de conversações sobre redes IP pode ser feita através do nível de opinião médio (MOS - *Mean Opinion Score*), que é uma métrica subjetiva comumente utilizada para isso e que expressa o julgamento dos usuários, através de uma escala de 1 (ruim) até 5 (excelente). Dos vocoders julgados, o G.711 é o que alcança o maior índice, com o MOS de 4.1 pontos. O vocoder G.726 tem a opinião média de 3.85 pontos.

Os parâmetros de QoS mais importantes para tráfego de tempo real, como o de VoIP, são os atrasos (delays), a variação desses atrasos (jitters) e a perda de pacotes. São eles que vão determinar a qualidade dos fluxos de voz. O delay, por exemplo, pode dar origem a ecos na recepção e, por isso, é importante reduzi-lo ao máximo possível, devendo-se também empregar algoritmos de cancelamento de eco. Como os pacotes experimentam atrasos diversos na rede, a variação do tempo de chegada desses pacotes no receptor (atrasos entre os pacotes ou jitter) geralmente não é constante, mesmo que o seja no lado do transmissor. Por isso, como já mostrado anteriormente, a necessidade de utilização de *buffers* nos receptores, para uma reprodução mais contínua e fiel da voz. Quanto às perdas

[21] Foram experimentalmente caracterizados neste trabalho, fluxos de voz gerados pelos vocoders G.726 e G.711.
[22] Conhecido como ***frame size*** ou ***frame length***, que é o período de tempo necessário para gerar um pacote de voz.
[23] A cada segundo são geradas 8.000 amostras de 1 byte, resultando em 8.000bytes/seg ou 64Kbits/seg.

de pacotes, taxas relativamente baixas podem ser toleradas, sem que haja grandes distorções na recepção. O número máximo de pacotes consecutivamente perdidos também é uma métrica importante, sendo um fator que está diretamente relacionado com o maior período durante o qual a voz é perdida[45]. Referente a isto, conforme [46], para um fluxo de voz, é considerada aceitável uma taxa de perda de pacotes de até 21% e um número máximo de 2 pacotes consecutivamente perdidos.

O atraso percebido num ambiente com VoIP é, na verdade, gerado por uma série de pequenos atrasos, como o de geração/formação do pacote e o de rede. O primeiro diz respeito ao tempo gasto para amostrar a voz e gerar um pacote de dados que represente a amostra gerada, sendo que está relacionado ao *frame size* mantido nos vocoders. Já o atraso de rede representa o tempo necessário para o transporte do pacote, do terminal de origem ao de destino, compreendendo o tempo de transmissão dos pacotes, atrasos com roteamento e o tempo de permanência dos pacotes nas filas dos roteadores da rede, que podem se minimizados com a implementação de serviços diferenciados. Ou seja, pode-se perceber que os atrasos enfrentados pelos pacotes podem ter uma parte fixa e outra variável. A parte fixa é a que não depende da carga da rede, mas, sim, do distanciamento entre transmissor/receptor, da velocidade dos links e da capacidade dos elementos de rede (memória e processamento, por exemplo). A parte variável é justamente aquela que resulta do comprimento da fila e da carga momentânea da rede. Desta forma, a diferenciação/priorização do tráfego não consegue interferir na parte fixa dos atrasos, mas, o que é mais importante, ajuda a minimizar sua parte variável.

A recomendação G.114[47], do ITU-T, define, conforme mostra a Figura 3-9, uma classificação de valores, em milissegundos, para a viabilidade do atraso em um sentido (OWD) nas aplicações de voz:

De acordo com o esquema a seguir, atrasos em um sentido de até 150ms são perfeitamente aceitáveis. No entanto, até 250ms, considera-se um atraso aceitável, pressupondo-se menor qualidade na conversação. Em ligações internacionais, que envolvem satélites, com maiores retardos, é aceitável um atraso de até 600ms[45].

Em aplicações de voz, adotando-se somente o UDP como protocolo de transporte, os pacotes podem tomar rumos distintos na rede e chegar fora da seqüência no

destino, o que seria resolvido caso se adotasse o TCP para verificar o seqüenciamento e a correção das mensagens. No entanto, em sua essência, apesar do uso de aplicações com tecnologia P2P, como o **Skype**[63][24], o TCP não suporta transmissão de voz em tempo real, porque utiliza mecanismo de recuperação dos dados perdidos por retransmissão. Ou seja, no caso da perda de um pacote, por exemplo, a liberação dos dados para a aplicação deverá esperar por todas as retransmissões, o que acarretaria atrasos intoleráveis. Desta forma, as aplicações de tempo real utilizam o UDP, ao invés do TCP, pela simplicidade e menor sobrecarga que o primeiro tem sobre o segundo. O problema de seqüenciamento é resolvido com o uso do UDP em conjunto com o protocolo RTP, definido pela RFC 1889[48], o qual provê serviço de entrega fim-a-fim para dados com características de tempo real, tais como áudio e vídeo interativos.

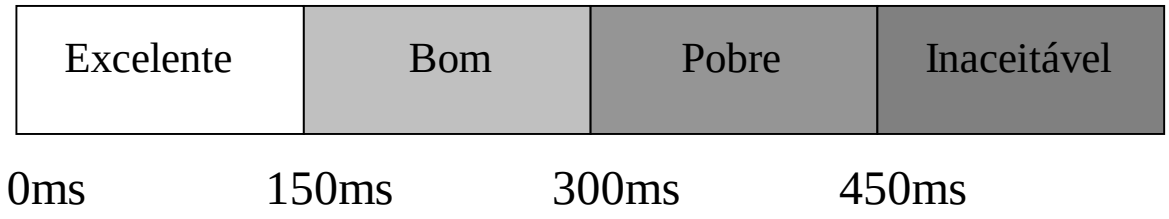


Figura 3-9 – Classificação da viabilidade de operação das aplicações de voz, segundo o ITU-T.

A voz codificada é montada em pacotes, que serão transportados pela rede, utilizando-se os protocolos IP, UDP e RTP para realizarem esse transporte. Como já comentado, por VoIP ser uma aplicação de tempo real, recomenda-se o uso conjunto desses protocolos, haja vista a redução do tempo perdido com confirmações e retransmissões.

Um pacote típico de voz é composto então por cabeçalhos daqueles protocolos e pela voz codificada em bits (parte também conhecida como *frame* de voz, carga útil ou *payload*), sendo que, os cabeçalhos, juntos, totalizam 40bytes[25] e contém informações necessárias para o transporte do pacote, como, por exemplo, endereços IP de origem e de destino, portas de serviços utilizadas, identificação do fluxo de voz, número de seqüência

[24] O Skype pode utilizar portas TCP, com valores acima de 1024.

do pacote, etc. É importante salientar, no entanto, que, por questões de eficiência, já que os frames de voz geralmente são pequenos, os pacotes podem carregar um ou mais desses frames. Por exemplo, um pacote contendo voz com 24bytes[26] de carga útil representa apenas 37,5% de eficiência[27]. A solução seria então incluir vários outros frames no mesmo pacote, de forma a aumentar esta eficiência.

Pode-se utilizar, ainda, a técnica de compressão de cabeçalho, que, como introduzido em [62] e aperfeiçoado em [20], ajuda a aumentar a eficiência do RTP, comprimindo os cabeçalhos RTP/UDP/IP de 40bytes para 2 a 4bytes. Tal compressão é especialmente benéfica para pacotes pequenos em links muito lentos, com a redução significativa dos atrasos e da sobrecarga.

3.6 Conclusão do capítulo

Este capítulo concluiu as definições iniciais descritas no capítulo anterior sobre o modelo de serviços diferenciados. Foram apresentados, além dos elementos funcionais da arquitetura e os aspectos acerca do condicionamento de tráfego presentes num domínio diferenciado, as formas de implementação de mecanismos de medição e policiamento de tráfego.

O capítulo abordou como QoS pode ser visto pelo prisma de contratos entre os usuários e os provedores de serviço, a partir dos conceitos de acordo e de especificação de nível de serviço (SLA e SLS, respectivamente). O conceito de PHB, um dos mais importantes em DiffServ, também foi contextualizado, sendo reproduzidos os aspectos mais importantes dos serviços expresso (PHB EF) e assegurado (PHB AF).

Questões no Linux responsáveis pelo controle e pela diferenciação de tráfego foram descritas neste capítulo de uma forma bem sucinta, já que o aprofundamento do assunto não é objetivo deste trabalho. As disciplinas de escalonamento aplicadas no *testbed* foram destacadas: FIFO, CBQ, SFQ e PQ.

---

[25] 20bytes para o IP + 8bytes para o UDP + 12bytes para o RTP.
[26] É o caso de pacote contendo voz no formato G.723.1.
[27] Percentagem resultante a partir do seguinte cálculo: (24/(24+40)*100), onde 24 é o tamanho da carga útil (em bytes) e 40 (também em bytes) é o tamanho do cabeçalho.

O capítulo abordou, por fim, aspectos relacionados às aplicações de voz sobre IP, com ênfase nos codificadores/decodificadores de voz (vocoders) caracterizados neste trabalho: G.711 e G.726.

4. O ambiente de testes

Apresenta-se, neste capítulo, o ambiente em que a fase experimental do trabalho foi realizada. Serão mostrados a topologia da rede e o domínio diferenciado, assim como os métodos utilizados nos experimentos e as ferramentas de software adotadas no testbed.

4.1 Topologia da rede

A plataforma de testes é composta por cinco microcomputadores Pentium, nos quais foram instalados sistemas operacionais Linux[28], com pacotes iproute2 e suporte à qualidade de serviço (DiffServ). Mostra-se com isso que, num ambiente sem custos e sem equipamentos sofisticados, composto apenas por microcomputadores, alguns dos quais exercendo funções de roteadores, e por softwares livres[29], se pôde estudar o desempenho de fluxos modelados de voz sobre IP com qualidade de serviço, utilizando o modelo diferenciado – mais especificamente o encaminhamento expresso (PHB EF).

A Figura 4-1 mostra a topologia que se adotou, montada numa rede Ethernet (IEEE 802.3 10BaseT/100BaseT) e disposta numa área local e completamente controlada, sem qualquer outro tráfego presente nos links, os quais apresentam velocidades de 10Mbps ou 100Mbps. Como o enfoque é trabalhar com fluxos característicos de voz, que geralmente ocupam muito pouca banda passante, e como cada fluxo desses representa um programa em execução no sistema operacional, seria completamente inviável preencher uma parte representativa dos links com fluxos de baixas taxas. Seria necessária para isso uma quantidade enorme de fluxos e, conseqüentemente, um número impróprio de processos no Linux competindo por recursos de máquina. Os resultados poderiam ser muito variáveis e duvidosos. Por exemplo, num link de 10Mbps, para se alcançar ¼ desse valor com fluxos modelados de voz de 32Kbps (geração característica de um vocoder G.726), seria necessário utilizar-se setenta e oito fluxos de voz e, portanto 78 processos nas máquinas geradoras. Como solução para tal problema, teve-se que configurar o controlador de tráfego para limitar consideravelmente o link. Isto foi conseguido através da utilização da

[28] Distribuição RedHat versão 9, com kernel 2.4.31.
[29] Os softwares livres utilizados nos experimentos foram os sistemas operacionais e as ferramentas de suporte a tráfego de rede, como os geradores, os analisadores e os medidores.

disciplina de serviço ***cbq*** e será mais bem abordado durante as explicações dos experimentos.

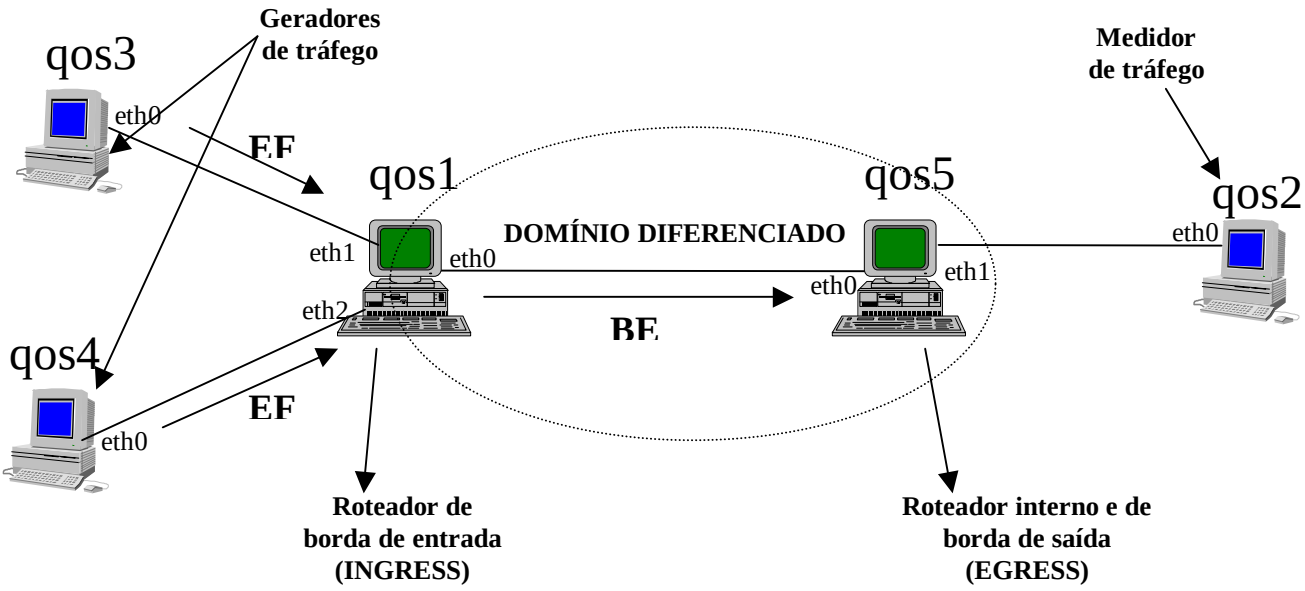


Figura 4-1 – Topologia da rede

O domínio diferenciado também pode ser visualizado na figura acima. Conforme explicado no Capítulo 3, num domínio DS os roteadores podem executar, em relação ao tráfego, funções de condicionamento distintas. No caso do domínio montado no testbed, os nós da rede exercem os seguintes papéis:

a) As máquinas qos3 e qos4 não fazem parte do domínio DS e servem exclusivamente para gerar tráfego prioritário (de voz), sem executar qualquer espécie de marcação ou classificação nos pacotes.

Em testes onde são tratados atrasos na ordem de milissegundos, a quantidade de processos sendo executados no sistema operacional de uma máquina pode influenciar nos resultados finais. Ou seja, cada fluxo gerado representa um processo em execução no Linux. Logo, num cenário onde precisam ser gerados, por exemplo, 32 fluxos de voz, em vez de executá-los todos numa mesma máquina, sobrecarregando-a, mais seguro fazê-lo distribuindo os processos entre as máquinas geradoras. E exatamente por isso, para se causar um balanceamento na carga total de geração dos fluxos de voz e se chegar a resultados mais confiáveis, é que foram adotadas duas máquinas para a mesma função.

b) A máquina qos1 opera como o roteador de entrada do domínio DS (ingresso), sendo responsável, essencialmente, pelas funções de marcação dos pacotes e classificação MF dos fluxos gerados. Tais funcionalidades foram teorizadas no Capítulo 3 deste trabalho. Além disso, para perturbar o meio, qos1 realiza também a geração de tráfego de fundo (melhor esforço ou apenas BE).

c) A máquina qos5 é vista tanto como o roteador interno como o de fronteira (tipo egresso) do domínio diferenciado. Ou seja, é um nó interno porque condiciona o tráfego vindo de um nó ingresso e é um roteador de fronteira porque, depois dele, nenhuma outra máquina realiza qualquer condicionamento de tráfego. O nó qos5 ignora os dois bits menos significativos do campo DSCP[30] e gera uma nova marcação nos pacotes, classificando-os única e exclusivamente através desse campo. Ou melhor, a máquina qos5 realiza classificação do tipo BA e, além disso, é onde, nos estudos experimentais, são aplicadas as disciplinas de escalonamento, com vistas à diferenciação de tráfego.

d) A máquina qos2 é sempre o destino dos pacotes. É nela em que são feitos os cálculos e as medições de tráfego, com base nos valores de QoS de interesse (delays, jitters, taxas de perdas e de vazão).

4.2 Metodologia adotada

Para que se chegue a um trabalho experimental confiável, é extremamente importante que as etapas sejam previstas a partir de planejamentos constantes e que sejam adotados métodos e padrões coerentes. Serão descritas nesta sessão quais métricas de QoS foram adotadas, como os resultados foram alcançados e os tipos de tráfego utilizados, além de se abordar também a sincronização de tempo na rede e as ferramentas de software aplicadas no testbed.

4.2.1 Aspectos gerais

As métricas de QoS colhidas para análise no trabalho foram as consideradas pela literatura como as mais relevantes para o estudo de voz em redes IP, que são o delay e

[30] Sem uso pelos roteadores do domínio DS, conforme RFC 2474[23].

sua variação (jitter)[33][58]. Foram também investigadas as taxas de perda de pacotes e de vazão.

Teve-se a preocupação de se seguir a base do modelo diferenciado, através do estudo do comportamento de agregados de fluxos caracterizados de voz. Na verdade, agregação é uma palavra-chave em redes com DS, introduzida para aumentar a escalabilidade da arquitetura. Neste contexto, dado o grande nível de variações dos resultados e, portanto, a pouca confiabilidade alcançada, não se adotou medições em cima de *fluxo de referência*, muito utilizado em trabalhos de simulação, como em [45] e em [49], por exemplo. Ao contrário disso, as métricas de QoS foram calculadas, para cada fluxo, através da média aritmética dessas medidas, avaliada sobre todos os pacotes daquele fluxo. Como se está trabalhando com agregados de fluxos, os valores de uma repetição são conseguidos, outra vez, pelo cálculo da média aritmética sobre os valores médios de cada fluxo, o que traz mais confiança e menos variações dos resultados durante as demais repetições de cada estudo experimental. Desta maneira, os testes foram repetidos várias vezes, de forma a garantir a validade estatística dos resultados, os quais foram sumarizados através do cálculo da média aritmética das métricas sobre o total de 12 repetições. Tal número foi considerado suficiente, diante das poucas variações observadas entre os resultados das repetições, uma vez terem sido realizadas, como já citado, num ambiente controlado.

Na noção de serviços diferenciados foram aplicados, sempre, o **encaminhamento expresso (PHB EF)**, para corresponder ao tráfego modelado de voz, e o **melhor esforço (PHB default)**, relativo ao tráfego de perturbação do meio. Ademais, como forma de se aproximar da situação real, os fluxos de voz foram gerados, em todos os testes, por fontes do tipo ON-OFF, com a utilização do UDP como protocolo de transporte. Já para o tráfego de fundo, foram empregados, dependendo da situação, os protocolo UDP ou TCP, sendo o primeiro mais comumente presente nos estudos, uma vez ter-se maior controle das taxas sendo geradas, já que a vazão resultante do TCP, por ser um protocolo adaptativo, sempre depende das condições momentâneas da rede, fugindo, de alguma forma, do controle do autor.

4.2.2 Sincronização do tempo

O estudo do tempo em que os pacotes percorrem de um lado a outro da rede (atraso fim-a-fim em um sentido ou *one-way delay*) requer muita precisão nos relógios das máquinas transmissoras e receptoras[31]. Isto ocorre porque as ferramentas que realizam tal medição calculam o delay tomando como base a diferença numérica entre o tempo em que o pacote chega no receptor e o tempo em que o mesmo é transmitido[32]. Se os relógios das máquinas envolvidas no processo de geração/recepção do tráfego estiverem alguns milissegundos fora de sincronia, isso, com certeza, como observado em laboratório, vai influenciar significativamente nos resultados finais. Visando não se ter problemas dessa natureza, configurou-se a rede para que, automaticamente, na virada de cada minuto, as máquinas do testbed sincronizassem seus relógios, pois se percebeu que 1 minuto apenas já era um tempo suficiente para desajustar, em pouquíssimos milissegundos, os relógios dos nós[33]. Foram usadas, para isso, as ferramentas "*ntpdate*" e "*at*", próprias do sistema operacional. Além disso, elegeu-se a máquina qos1 como servidora de tempo (NTP *server*), a quem as máquinas geradoras (qos3 e qos4) e receptora de tráfego (qos2) solicitavam, a cada minuto, sincronização do tempo. Uma forma possível de se obter a sincronização perfeita do tempo numa rede de computadores seria através de uma interface especial de relógio baseada em GPS, que forneceria um padrão de tempo global único, com precisão. No entanto, trata-se de um hardware incomum e com custos adicionais[50].

Ademais, ainda com relação à sincronização de tempo, percebeu-se, durante os testes iniciais, dois fatos importantes: i) quando os experimentos foram disparados no início ou pouco além do meio do minuto, os atrasos sofriam uma elevação muito discreta, pois são momentos em que os relógios não estavam totalmente sincronizados[34]; ii) quando os experimentos duraram mais de 60 segundos, os atrasos dos pacotes tenderam a uma certa

[31] Principalmente neste testbed, pois, como a rede é local e controlada, os valores de delay e jitter tendem a ser inferiores aos encontrados em redes de longas distâncias, com satélite, links de baixas velocidades, roteamentos excessivos, etc.

[32] Quando o pacote chega no destino é marcado com um **timestamp de recepção** e quando o mesmo pacote sai da máquina de origem ele é rotulado com um **timestamp de transmissão**.

[33] Isso é percebido através do comando "ntpdate", do Linux, que força a sincronização do tempo entre duas máquinas na rede e que usa o protocolo NTP (*Network Time Protocol*).

[34] Isso se dá porque, nos primeiros momentos após a sincronização das máquinas (que ocorre na virada do minuto), os relógios ainda não estão totalmente sincronizados. Logo, há um momento ótimo em que as máquinas têm seus relógios sincronizados, mas, depois, aproximadamente no quadragésimo segundo do minuto, as máquinas tendem a perder novamente a sincronização total.

elevação, já que a virada do minuto é um período crítico, pois é o momento em que as máquinas realizam a sincronização do tempo, carregando discretamente a rede. A solução então foi encontrar um momento ótimo para o início de cada experimentação e fazer com que os testes tivessem uma curta duração. Desta forma, padronizou-se que todo teste seria inicializado no décimo terceiro segundo do minuto e que duraria apenas 20 segundos.

4.2.3 Ferramentas de software

Para suporte fundamental à realização do trabalho, foram necessárias algumas ferramentas de software, responsáveis pela geração, medição e análise de tráfego.

Os componentes *mgen*, *drec* e *mcalc*, da família MGEN[51], foram aplicados, respectivamente, para a geração, a recepção e a medição do tráfego UDP. Foi através do *mcalc* que se pôde sumarizar os dados para a geração dos gráficos aqui apresentados e analisados. O MGEN permite a geração controlada de fluxos UDP, sendo possível a transmissão de tráfego *unicast* ou *multicast*, a partir da taxa de bits desejada. A ferramenta permite ainda controlar o tempo do experimento e a variação do tamanho dos pacotes UDP estudados, além de aplicar a noção de "*host time*", definida em [28], em que há o registro do "*timestamp*" nos pacotes imediatamente antes de serem transmitidos na rede e assim que sejam recebidos no destino. Os dois tempos servem, como já destacado mais acima, para calcular o OWD. As versões do MGEN utilizadas nos testes foram a 3.2 para o componente *drec* e a 3.0 para o componente *mgen*[35], já que, com tráfego do tipo ON-OFF, a execução do *mgen* 3.2, sempre usada por meio de *scripts*, abortava com "falhas de segmentação". Trabalhou-se com o *mcalc* 3.2 para o processamento dos valores métricos de QoS.

Especificamente para o uso de tráfego do tipo TCP, utilizou-se o NETPERF[52], definido como uma ferramenta de *benchmark*, desenvolvida pela divisão de redes da Hewlett-Packard Company e útil para medir aspectos de desempenho de rede. O NETPERF satura o meio e retorna a vazão alcançada pelo fluxo analisado, sendo composto por dois programas básicos: *netperf* e *netserver*. A ferramenta é baseada no esquema **monitor-refletor**, onde o tráfego é gerado numa estação (a com o *netperf*), transmitido para o nó destino (o com o *netserver*), refletido e retornado à estação transmissora, a qual

[35] Mesmo numa rede muito bem sincronizada, a versão 3.0 do *drec* apresentou valores de delay e jitter incoerentes e muito elevados.

apresenta os resultados. Tal esquema é especialmente necessário em testes onde se deseje calcular atrasos de ida-e-volta dos pacotes[36], com menos custos. Significa dizer nesse caso que a maior carga de processamento se restringe no lado transmissor (monitor) dos dados, com o lado refletor efetuando apenas a função de retransmissão dos pacotes, sem qualquer processamento extra de informação. A versão do NETPERF aqui empregada foi a 2.2.

A análise de rede foi realizada através da ferramenta *tcpdump*[53], indispensável para um acompanhamento preciso dos pacotes durante os experimentos e para a obtenção de conclusões importantes. É utilizada, algumas vezes, para confirmar tamanho dos pacotes (*heads* e *paylods*) e para computar o número total de pacotes e a vazão. Com *tcpdump*, os dados são capturados durante as sessões de testes e analisados após o processamento, em arquivos de registro (log).

Por fim, para o controle de tráfego, como já sinalizado anteriormente, utilizou-se a ferramenta *tc* do Linux, ao nível de usuário, para dar o suporte suficiente e necessário aos serviços diferenciados, basicamente criando-se e associando-se filas aos dispositivos de saída de rede.

4.3 Conclusão do capítulo

Foram abordados, neste capítulo, itens importantes referentes ao ambiente de testes. A topologia de rede, disposta numa área local e controlada, e o domínio diferenciado, foram apresentados, especificando-se as funções de condicionamento de cada nó do *testbed*.

Definiu-se a metodologia utilizada nos testes, descrevendo-se a forma de cálculo das métricas de QoS e os serviços DiffServ utilizados.

Apresentou-se também como o tempo na rede foi tratado, destacando-se a importância da sincronização entre as máquinas para o alcance de resultados confiáveis.

Finalizou-se o capítulo com a exposição das ferramentas de software adotadas nos experimentos.

[36] Conhecido como "*round-trip delay*".

Mostram-se, nos próximos capítulos, detalhes dos estudos experimentais realizados na plataforma de testes e as análises correspondentes aos valores extraídos.

5. Modelagem de vocoders de áudio

Teve-se como primeira preocupação, antes mesmo de qualquer tipo de modelagem, a certificação de que o ambiente de testes montado era capaz de prover QoS, com diferenciação de serviço entre classes, através da aplicação, no roteador interno (QoS5), de disciplinas de escalonamento de tráfego. Concebeu-se, então, um estudo experimental para isso, determinado como pré-requisito para os demais estudos.

O intuito desse primeiro estudo era também comparar os valores métricos de QoS entre as implementações, sendo empregados três tipos de escalonadores disponíveis no Linux: ***prio***, ***sfq*** e ***pfifo***. Para cada caso foi reservada, através da criação de uma disciplina de serviço ***cbq*** e de uma classe (também cbq) a ela instalada, a banda passante de 2.048Kbps no roteador interno e, sob tal disciplina, foram aplicados os escalonadores correspondentes. Ou seja, no primeiro caso, instalou-se a disciplina ***prio*** na classe cbq já criada, utilizando-se duas bandas[37] (das três criadas automaticamente[41]) para o escoamento dos fluxos e, em cada banda dessa, anexou-se uma fila crua do tipo FIFO, com limite de 10 pacotes em cada uma. Para o caso da disciplina ***sfq***, esta foi anexada diretamente na classe cbq, com os fluxos sendo classificados, cada um, para uma das 128 filas criadas automaticamente[42] e escalonados, em ordem *round-robin*, por meio de algoritmo de *hashing*. A opção *quantum*[38] utilizada foi a default do sistema (1.514bytes) e o valor da opção *perturb*[39] escolhido foi de 15 segundos[40]. É vista como uma disciplina projetada para garantir que cada fluxo tenha um acesso justo aos recursos da rede, prevenindo que rajadas de um fluxo monopolizem a rede, sendo este o motivo pelo qual a disciplina ***sfq*** não é aconselhada para tráfego de tempo real. Por fim, para a disciplina ***pfifo***, fez-se algo semelhante, anexando-a à classe cbq criada e configurando-a para comportar o limite máximo de 20 pacotes. Neste caso, os pacotes dos fluxos concorrem na mesma fila, sem qualquer tratamento especial. Pode-se ver a utilização deste escalonador como a ausência de QoS na rede.

---

[37] A banda 0 foi direcionada para o tráfego mais prioritário e a banda 1 para os pacotes de menor prioridade. Como já se sabe, o PRIO age com **prioridade estrita**, na qual os pacotes direcionados para as bandas mais altas (de menor prioridade) somente são servidos quando não houver tráfego nas bandas de maior prioridade.

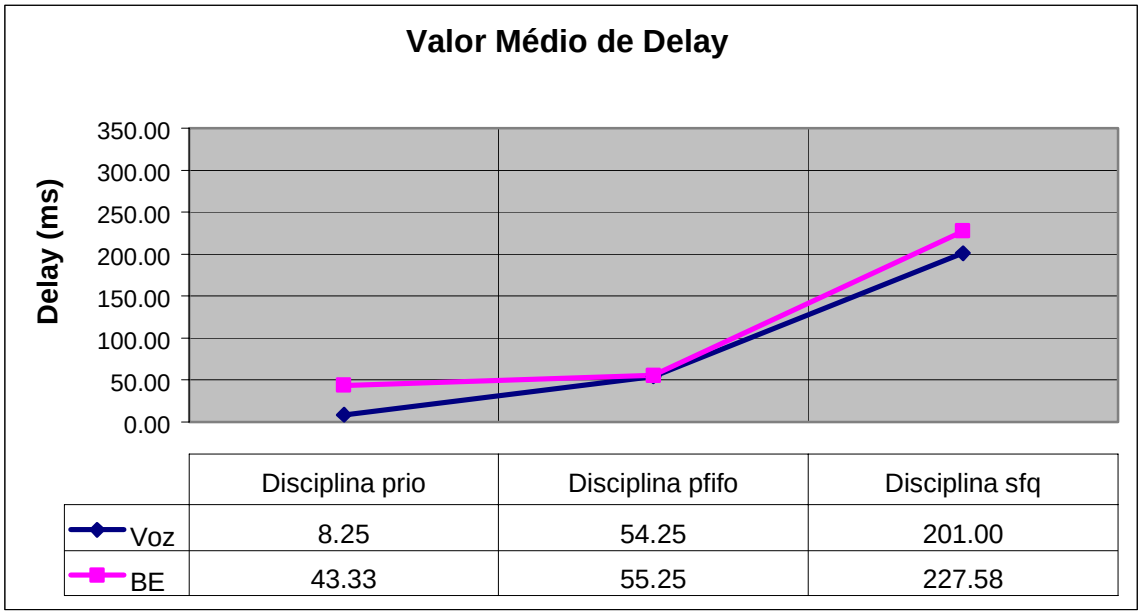
[38] Especifica a quantidade de bytes que pode ser enviada, de cada fila, durante a passagem do round-robin.

[39] Introduz perturbação periódica no cálculo de *hash*, para evitar colisões de pacotes.

[40] Em testes preliminares, percebeu-se que, quanto maior o valor desta opção maior os valores de atrasos e jitters encontrados. Escolheu-se então um valor abaixo da duração dos testes.

Como o objetivo era fundamentalmente o de mostrar a capacidade do testbed de poder diferenciar tráfego, foram injetados na rede 02 fluxos exatamente idênticos, tendo, cada um deles, as seguintes características: UDP como protocolo de transporte; pacotes com carga útil de 520bytes; fonte geradora do tipo contínua (CBR) e agregado composto por oito fluxos de 160Kbps (38.5 pacotes por segundo – pps) cada um, com taxa total de geração de 1.280Kbps. Utilizou-se, para isso, a ferramenta *mgen*, responsável pela geração/ medição dos agregados.

Os gráficos da Figura 5-1 e da Figura 5-2 exibem, para cada disciplina, os respectivos valores de atraso e jitter médios alcançados pelos dois fluxos idênticos injetados na rede, com um deles representando o fluxo de voz e o outro o fluxo de melhor esforço (BE).



	Disciplina prio	Disciplina pfifo	Disciplina sfq
Voz	8.25	54.25	201.00
BE	43.33	55.25	227.58

Figura 5-1 – Valor médio de delay

Vê-se, claramente através dos valores obtidos, como a disciplina ***prio*** consegue tratar, diferentemente, ambos os fluxos, priorizando estritamente o de voz em detrimento ao de melhor esforço. O tráfego BE com ***prio*** tem, ainda assim, conseguido melhor desempenho do que os fluxos de voz e de fundo das demais disciplinas (***pfifo*** e ***sfq***), dada a eficiência com que os pacotes de voz são encaminhados com a disciplina prioritária. Ou seja, os valores de BE com ***prio*** são baixos (em relação aos valores das demais disciplinas) porque os números referentes à voz já são significativamente pequenos.

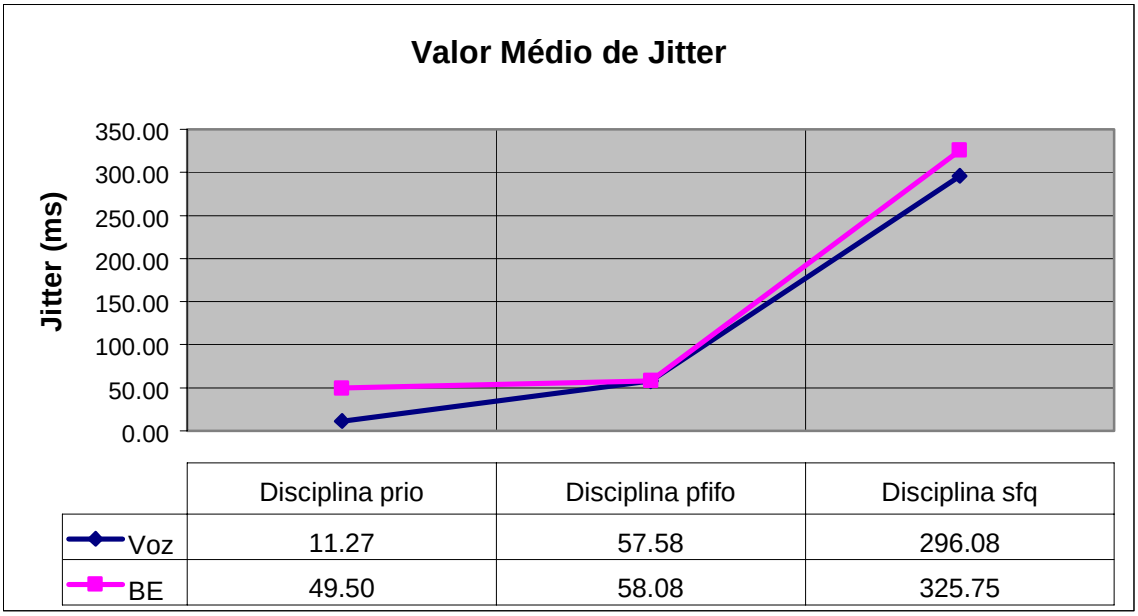


Figura 5-2 – Valor médio de jitter

Chama a atenção também a proximidade dos valores de voz e BE nas disciplinas ***pfifo*** e ***sfq***, explicada pela eqüidade no tratamento dado aos fluxos. No caso do ***pfifo***, isto é traduzido pela competição dos pacotes dos fluxos pelos espaços (20 pacotes) de uma mesma fila. Já no caso do ***sfq***, os valores próximos são confirmados pela natureza justa com que os fluxos são tratados. Os valores bastante elevados com ***sfq*** foram alcançados, muito provavelmente, por causa da configuração da disciplina, mais especificamente devido ao grande tamanho de cada fila (128 pacotes[41]), elevando os atrasos e suas variações. Ademais, a variação do limite da fila com ***pfifo*** também influi diretamente nos valores de jitter e delay. Adotou-se o limite de 20 pacotes para efeito de comparações, já que a disciplina ***prio***, embora em 2 filas, como já descrito, igualmente apresenta 20 pacotes.

As figuras a seguir mostram os gráficos relativos às taxas de perdas de pacotes e de vazão conseguidas por cada uma das disciplinas testadas. Destaca-se, na Figura 5-3, a percentagem de pacotes perdidos na banda 0 (referente à voz) da disciplina ***prio***.

Demonstrando, outra vez, a eqüidade com que os fluxos são tratados, vê-se a proximidade das taxas alcançadas com ***sfq*** e ***pfifo***.

[41] Este limite pode ser alterado, mas apenas no próprio código-fonte, antes da compilação[42].

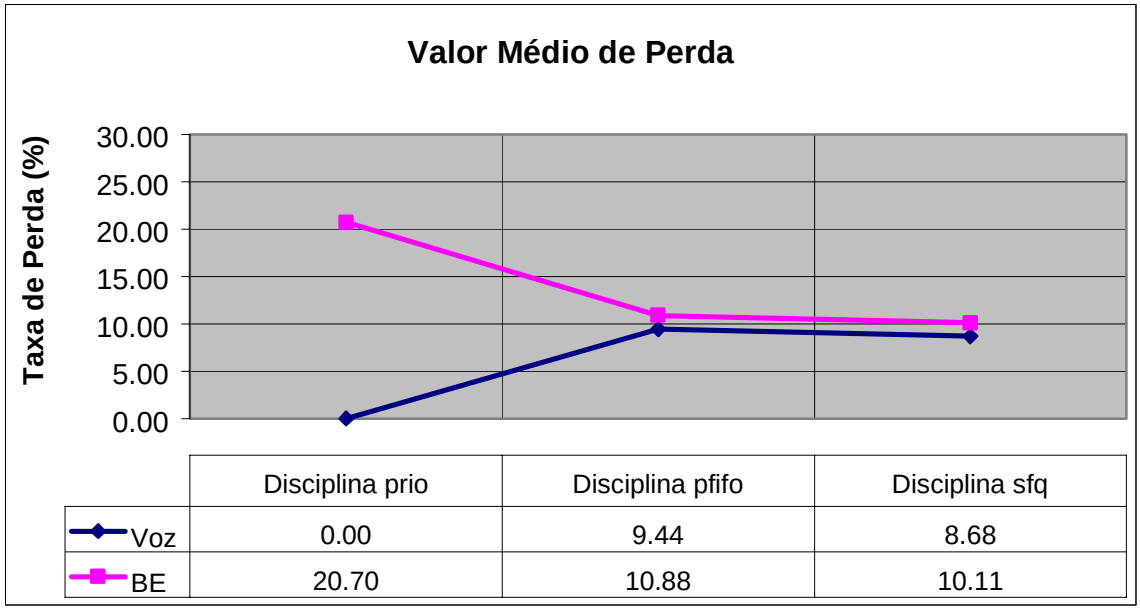


Figura 5-3 – Taxa de perda de pacotes.

Estabeleceu-se, em todo o trabalho, que, para vazão, a métrica adotada não seria simplesmente o valor da vazão atingido pelos fluxos, mas, sim, a razão entre este valor e a taxa gerada para os fluxos. Isto se deu porque há experimentos que comparam cenários em que as taxas de geração são distintas, dada, principalmente, à natureza de tráfego do tipo ON-OFF, como poderá ser percebido mais adiante. Tal métrica ficou definida como "**taxa de vazão alcançada**".

A Figura 5-4 mostra os resultados conseguidos pelas disciplinas e, curiosamente, percebe-se que é a imagem inversa da taxa de perdas. A princípio, é possível compreender-se que quanto maior a "vazão alcançada" por um fluxo menor será a resistência (perdas) que o mesmo vai enfrentar em seu percurso, e vice-versa. Talvez o que não seja óbvio é que isso se desse de uma forma inversamente proporcional, como mostrou os resultados.

Tendo-se provado que o ambiente de teste é capaz de diferenciar tráfego, através da aplicação e configuração de disciplinas de serviço como o ***prio***, partiu-se para estudar os efeitos alcançados por fluxos modelados de áudio, característicos de diferentes tipos de codificadores/decodificadores de voz, com a variação de tamanhos de pacotes e número de fluxos agregados prioritários. Foram modelados fluxos ON-OFF característicos dos seguintes vocoders:

a) G.726, com *frame size* de 30ms, respectivos pacotes com tamanhos de 120bytes e fluxos ON-OFF gerados a uma taxa de 32Kbps[42] cada um;

b) G.711, com *frame size* de 25ms, respectivos pacotes com tamanhos de 200bytes e fluxos ON-OFF gerados a uma taxa de 64Kbps cada um;

c) G.711, com *frame size* de 65ms, respectivos pacotes com tamanhos de 520bytes e fluxos ON-OFF gerados, também, a uma taxa de 64Kbps cada um.

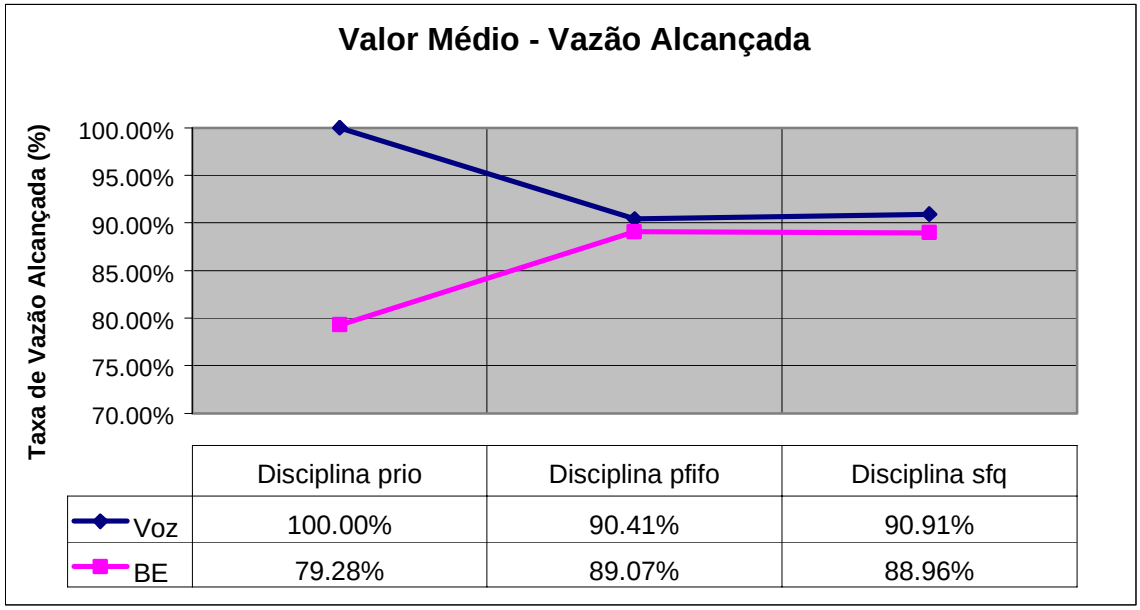


Figura 5-4 – Taxa de "vazão alcançada".

À primeira vista pode-se imaginar que os tamanhos dos *frames* são grandes demais para representar VoIP. Na verdade, a escolha se deu em função de uma limitação do ambiente de testes, já que, se fossem empregados frames bem menores[43], isso acarretaria na necessidade de um número proporcionalmente maior de fluxos e, conseqüentemente, de mais processos Linux. Como já citado, isso comprometeria os resultados.

Os fluxos de voz foram testados no intervalo de 20 segundos, sendo que os momentos ON-OFF foram respeitados seguindo, para cada segundo de teste, 400ms de atividade e 600ms de silêncio. Como não se controla a quantidade exata em gerações com fluxos do tipo ON-OFF, foram usadas taxas com **percentagens proporcionais**, onde, para

[42] Em momentos de atividade de cada fluxo gerado.
[43] Se fosse modelado o vocoder G.729, por exemplo, os fluxos deveriam ser gerados a uma taxa de apenas 8Kbps cada um. Isso significa que, para um estudo comparativo em relação ao vocoder G.711, seria necessário gerar 8 fluxos (processos Linux) G.729 para equivaler a 1 fluxo G.711.

cada tipo de vocoder analisado, as taxas de geração de voz ficaram em 20% do total gerado, com o restante (80%) sendo preenchido por tráfego de melhor esforço (BE). Para esclarecer essas proporções, a Tabela 5-1 mostra, para cada tipo de vocoder, a vazão de cada fluxo ON-OFF, quantos fluxos foram necessários gerar e as diferentes vazões dos agregados alcançadas, com todas discretamente acima da linha dos 500Kbps. Os valores referentes aos fluxos de fundo (BE) também são mostrados na Tabela 5-1, com as respectivas vazões correspondendo a 80% da carga total da rede.

Tipo de Vocoder		Taxa gerada em 1 fluxo nos momentos ON	Tamanho útil do pacote (payload)	Vazão efetiva em cada fluxo	Nº de fluxos utilizados	Vazão total efetiva[44]
G.726 (120b)	Voz	32Kbps	120b	13.86Kbps	37	512.7Kbps
	BE (CBR)	-	1470b	2.051Kbps	1	2.051Kbps
G.711 (200b)	Voz	64Kbps	200b	26.42Kbps	19	502.0Kbps
	BE (CBR)	-	1470b	2.008Kbps	1	2.008Kbps
G.711 (520b)	Voz	64Kbps	520b	30.02Kbps	17	510.5Kbps
	BE (CBR)	-	1470b	2.042Kbps	1	2.042Kbps

Tabela 5-1 – Vazão agregada por vocoder.

Com base nos valores de geração, estabeleceram-se, também proporcionalmente, os valores de reserva de banda, configurados no roteador interno através da disciplina ***cbq***, partindo-se do princípio de se limitar aproximadamente **2.048Kbps** para uma geração conjunta em torno de 512Kbps para voz e 2048Kbps para tráfego de melhor esforço[45]. Dessa regra básica, têm-se as reservas na Tabela 5-2.

Tipo de Vocoder	Reserva de banda em QoS5 (Kbps), através da configuração da disciplina *cbq*
G726-120b	2.050,96Kbps
G711-200b	2.008,00Kbps
G711-520b	2.042,07Kbps

Tabela 5-2 – Reservas de banda no roteador interno do domínio DS.

O estudo comparou o desempenho dos fluxos de voz sobre dois tipos de escalonadores Linux: ***prio*** e ***bfifo***. No primeiro caso, anexou-se ao ***prio***, nas bandas de voz e de melhor esforço, filas do tipo FIFO, limitada por bytes[46] (tipo *bfifo*) e fixadas em 1.600bytes para voz e 14.700bytes para o tráfego default. No segundo caso, com a

[44] Valor conseguido com a ausência de mecanismos de diferenciação de serviço.
[45] Ver valores da coluna "Vazão total efetiva", da Tabela 5-1.
[46] Em testes com filas FIFO do tipo ***pfifo***, anexadas às bandas do ***prio***, não foi possível observar-se diferenciação nos valores de atraso entre os vocoders, já que a variação dos tamanhos dos pacotes não influenciou nos resultados, porque o limite da fila era fixo e estipulado por pacotes. Por exemplo, numa

disciplina ***bfifo***, estabeleceu-se, para efeito de comparações com a disciplina prioritária, o tamanho da fila única FIFO fixada em 16.300bytes, que corresponde à soma dos tamanhos das duas bandas utilizadas com ***prio*** (ou seja, 1.600 + 14.700). É importante frisar também que o uso da disciplina ***bfifo*** serve para representar a ausência de qualidade de serviço, já que, nela, não há qualquer tratamento especial ou diferenciado. Os resultados serão mostrados nas próximas sessões.

5.1 Aplicação da disciplina *prio*

5.1.1 Estudo do delay

Observando-se a Figura 5-5, percebe-se que o atraso dos agregados de voz aumenta discretamente com o tamanho dos pacotes. Esse crescimento é causado, basicamente, pela variação do tamanho dos pacotes em estudo e pela priorização do tráfego de voz.

A variação no tamanho dos pacotes afeta diretamente no atraso de geração e, conseqüentemente, no tempo em que esses pacotes vão gastar para alcançarem o nó receptor (conhecido como “atraso de transmissão”). Deve-se perceber que pacotes maiores levam mais tempo para serem gerados e isso repercute no valor médio de delay daqueles pacotes. No caso em questão, conforme a Tabela 5-3, mantém-se praticamente o mesmo valor da taxa de geração do tráfego BE, mas, à medida que se aumenta o tamanho dos pacotes modelados dos vocoders (G.726-120bytes, G.711-200bytes e G.711-520bytes – nessa ordem), diminui-se, necessariamente, a taxa de pacotes EF gerados por segundo. Logo, aumenta-se o tempo de geração (pela ferramenta MGEN) e de transmissão de cada pacote de voz, o que contribui para elevação do atraso médio desses pacotes.

---

pfifo limitada em 10 pacotes, com o aumento do tamanho dos pacotes a fila sempre terá os mesmos 10 pacotes, o que mantém, praticamente, os mesmos resultados.

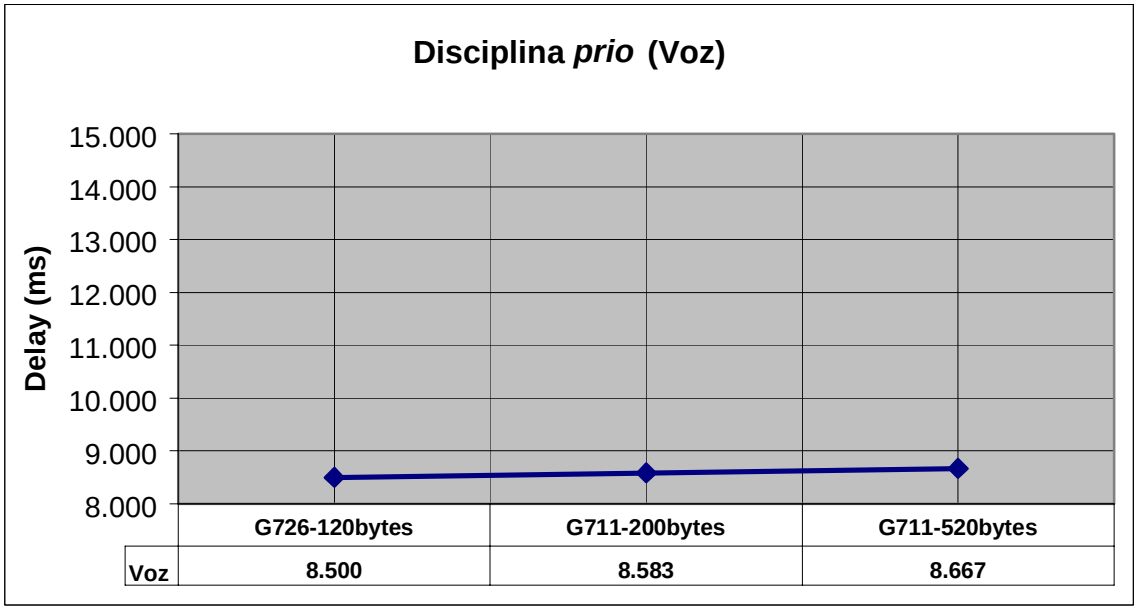


Figura 5-5 – Delay de voz com a disciplina *prio*.

Além disso, sabe-se que, com a disciplina ***prio***, os pacotes EF são classificados para a banda mais prioritária e escalonados imediatamente para transmissão enquanto houver pacotes na fila correspondente (forma estrita). Há, com isso, um nível mínimo de concorrência com o tráfego de fundo, mesmo que este tenha probabilidade de ser servido, já que os agregados de voz são do tipo ON-OFF e, em momentos de silêncio desses fluxos, o tráfego BE aumenta suas chances de ocupar a fila de transmissão[47]. Mesmo assim, entretanto, a afluência inibe a atuação dos pacotes de melhor esforço.

Atrasos com geração de pacotes				
	Número de pacotes efetivamente gerados por segundo (pps)[48]			
Tipo de Vocoder	**Agregado de Voz (ON-OFF)**	**Fluxo BE (CBR)**	**Tempo para gerar 1 pacote EF (ms)**	**Tempo para gerar 1 pacote BE (ms)**
G726-120b	517.55	174.40	**1.93**	5.73
G711-200b	303.87	170.75	**3.29**	5.85
G711-520b	119.50	173.64	**8.36**	5.75

Tabela 5-3 –Atrasos de geração de pacotes.

Com base nas linhas adotadas acima, pode-se concluir que, quanto menor o tamanho dos pacotes: menos tempo será gasto para se gerar cada pacote; para manter a

[47] Estágio adicional de armazenagem de pacotes. Geralmente é uma fila FIFO, limitada por número de pacotes (configurável pelo comando *ifconfig* do Linux) e localizada após as decisões de escalonamento do sistema operacional (TC).
[48] Valores reais médios, extraídos em cada experimento.

mesma carga na rede, mais pacotes serão injetados; os momentos de ociosidade da banda 0 (voz) serão menores; a possibilidade do tráfego de fundo atrapalhar o encaminhamento dos pacotes de voz será menor; e, conseqüentemente, os atrasos médios de uma via serão menores.

Em síntese, justificam-se os resultados extraídos com ***prio*** pelo fato dos pacotes maiores sofrerem tempos mais elevados para serem gerados e transmitidos na rede e, também, em decorrência dos pacotes de voz praticamente não enfrentarem concorrência com o tráfego de perturbação.

A Figura 5-6 inclui, à Figura 5-5, a linha dos fluxos de melhor esforço contínuos (CBR), que, ao contrário dos fluxos de voz, melhoram seus atrasos à proporção que os pacotes aumentam de tamanho, quando, mesmo com o método estrito da disciplina ***prio***, mais pacotes BE concorrem com o tráfego de voz. Com o vocoder G.711 (520bytes), por exemplo, isso é mais evidente. É quando então o tráfego de fundo enfrenta menos concorrência com o número de pacotes de voz injetados na rede – cada pacote BE, nesse caso, concorre com menos de 1 pacote prioritário (apenas 0.68, como mostra a Tabela 5-5 mais adiante, na Sessão 5.2.1).

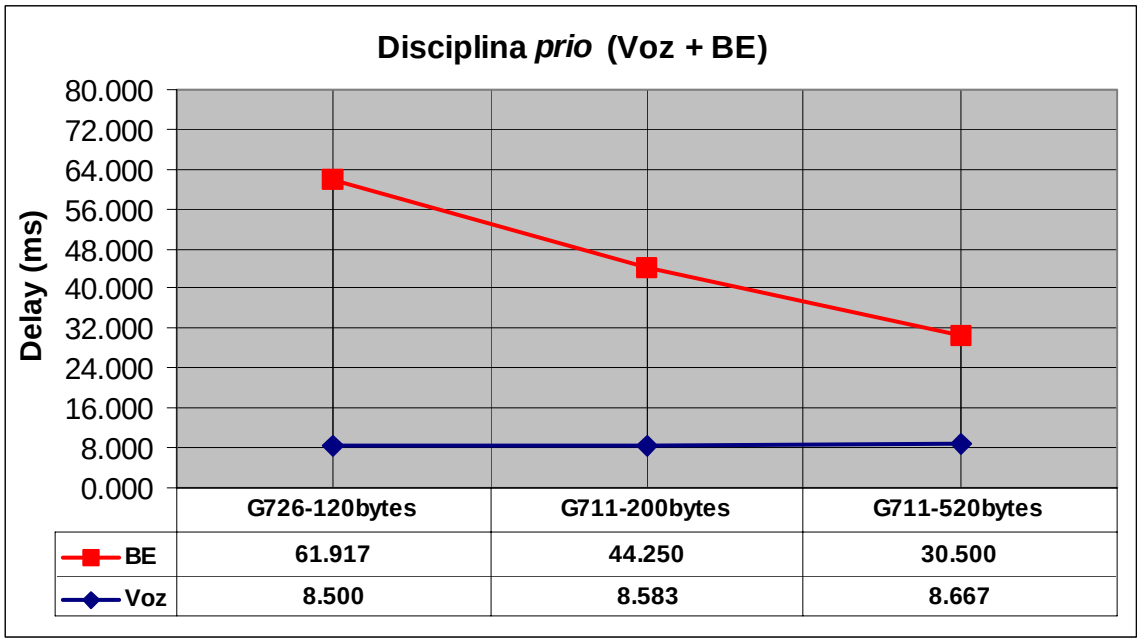


Figura 5-6 – Delay do tráfego de perturbação com a disciplina *prio*.

É bom entender que a modelagem dos vocoders aqui experimentada se baseou apenas em parâmetros possíveis de serem caracterizados no testbed, tais como a variação do tamanho dos pacotes, a variação do número de fluxos e a geração de fluxos formados

por uma variação regular de atividade-silêncio. Isto quer dizer que atrasos relacionados à geração/formação dos pacotes de VoIP, próprios do processo de codificação/decodificação realizado pelos vocoders de áudio, foram ignorados nos testes, obviamente porque não foram empregados vocoders reais, sendo considerados apenas os atrasos de geração de dados e de transmissão dos pacotes.

Vale destacar, no entanto, que os atrasos resultantes já incluem a sobrecarga introduzida por processamentos adicionais da disciplina de escalonamento ***cbq***, especificamente quanto às atividades de classificação de pacotes, determinação das filas a serem servidas e cálculos de estimativa de banda a ser usada pelas classes do *tc*.

5.1.2 Estudo do jitter

Os valores de jitter são vistos na Figura 5-7, com o vocoder G.726 marcando, agora, os valores mais elevados e o G.711 (520bytes) apresentando o melhor resultado.

Vê-se que o jitter decai com o aumento do tamanho dos pacotes de voz. As cargas dos tráfegos de BE e de EF são sempre proporcionais, com o número de fluxos e o tamanho dos pacotes sendo, praticamente, constantes com BE e variados com EF, dependendo do vocoder em estudo. Como já abordado, com EF, reduzindo-se o tamanho dos pacotes, para manter a carga proporcional, deve-se aumentar o número de pacotes injetados na rede[49]. Assim aumenta a probabilidade de que esses pacotes enfrentem maiores variações nos atrasos de enfileiramento (*queuing delay*) ao longo do percurso, sendo mais possível que se tenha, mesmo com a atuação estrita da disciplina ***prio***, uma mistura variável maior de pacotes de voz com pacotes de melhor esforço.

De uma outra forma, agregados com pacotes de tamanhos maiores apresentam melhor desempenho quanto ao jitter e isto pode ser explicado pelo comprimento das rajadas. Ou seja, em disciplina como ***prio***, a presença de rajadas mais longas (pacotes maiores) reduz o jitter, considerando que os pacotes da mesma rajada experimentam atrasos de enfileiramento similares. Como conseqüência, tais pacotes atingem também valores de jitters menores[54], conforme se vê o comportamento do tráfego de voz na Figura 5-7. Como já explicado, pode-se atribuir ainda os resultados ao fator agregação, que, conforme

[49], afeta também o jitter médio experimentado por fluxos prioritários, que aumenta com a elevação do número de fluxos.

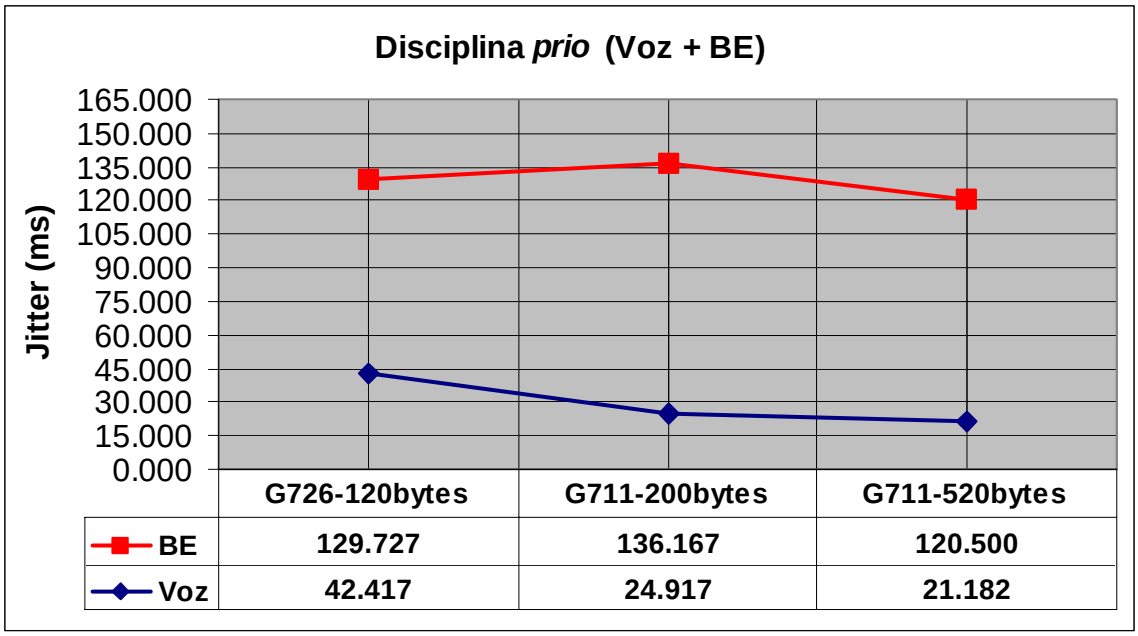


Figura 5-7 – Jitter com a disciplina *prio*.

5.1.3 Estudo das taxas de perdas de pacotes e de vazão

A Figura 5-8 mostra a taxa de descarte de pacotes que os fluxos conseguiram no estudo experimental, tendo a modelagem do vocoder G.726 obtido o melhor desempenho e a do G.711, com 520 bytes, apresentando as piores taxas.

Pode-se atribuir tais resultados à eficiência com que a disciplina ***prio*** consegue encaminhar um número mais evasivo de pacotes menores. Isto é motivado pelo fato do tamanho da fila referente à banda prioritária (fila do tipo FIFO, limitada em apenas 1.600bytes) ser fixado em número de bytes (BFIFO), o que faz com que pacotes maiores, mesmo que gerados em menor quantidade, apresentem taxas de descarte mais elevadas, uma vez que ocupam mais espaço da fila. Isto é confirmado através de testes preliminares, com filas anexadas às bandas do ***prio*** sendo do tipo FIFO e limitada, desta vez, por número de pacotes (PFIFO), em que, nas mesmas condições de rede (carga da rede, nº de fluxos, etc), se percebeu um comportamento contrário, com taxas de descarte muito baixas e

[49] São injetados, em média, ao final de cada experimento, 10.351 pacotes na modelagem com G.726, 6.077 pacotes com G.711 (200bytes) e apenas 2.389 pacotes com G.711 (520bytes).

tendendo a 0% com o aumento do tamanho dos pacotes[50]. Ou seja, pode-se perceber, neste caso, que o tamanho dos pacotes não exerceu qualquer influência na taxa de descarte, uma vez que o comprimento da fila era inflexível e limitado por um número fixo de pacotes. Houve uma pequena influência nas perdas, sim, mas do número de pacotes gerados por segundo, com taxas de descarte maiores para vocoders com taxas de geração maiores (com pacotes menores).

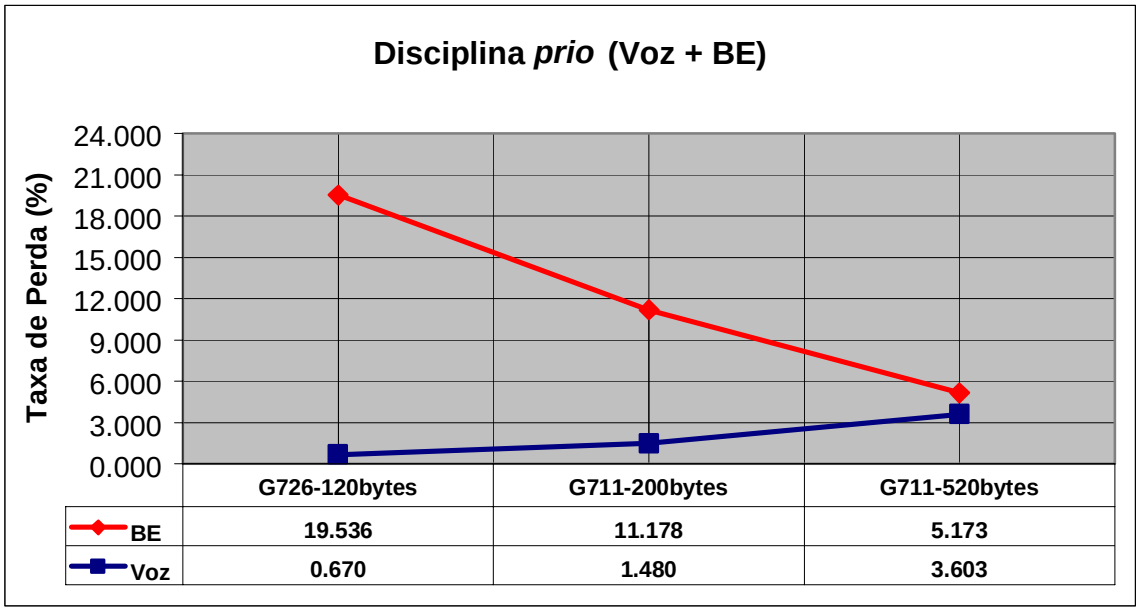


Figura 5-8 – Taxa de descarte de pacotes com a disciplina *prio*.

Um outro fator que ajuda a explicar o quadro mostrado na Figura 5-8 é a atuação, na banda 0 da disciplina ***prio***, de fluxos de natureza ON-OFF. Ou seja, com uma agregação maior, momentos de silêncio de um fluxo de voz muito provavelmente serão preenchidos por outros fluxos igualmente prioritários, com menor possibilidade dos pacotes de melhor esforço ocuparem a fila da interface de saída. Isto contribui para que tráfego com maior número de fluxos apresente melhor desempenho (caso mais bem sucedido é a modelagem do vocoder G.729, com pacotes de apenas 120bytes e distribuídos num total de 37 fluxos).

As justificativas acima servem também para explicar as taxas de "vazão alcançada" (Figura 5-9), uma vez que tem melhor vazão o tráfego que encontrou menos perdas e resistência durante seu percurso. Como no experimento anterior, vê-se, aqui, que a

[50] As taxas de descarte dos pacotes de voz com **filas PFIFO** sendo anexadas à disciplina ***prio*** foram as seguintes: 0.6708% para o vocoder G.726, 0.0591% para o vocoder G.711 (200bytes) e 0% para o vocoder G.711 (520bytes).

taxa de “vazão alcançada” é exatamente o reflexo do que ocorre com a taxa de descarte de pacotes, com a modelagem do G.726 obtendo o melhor desempenho.

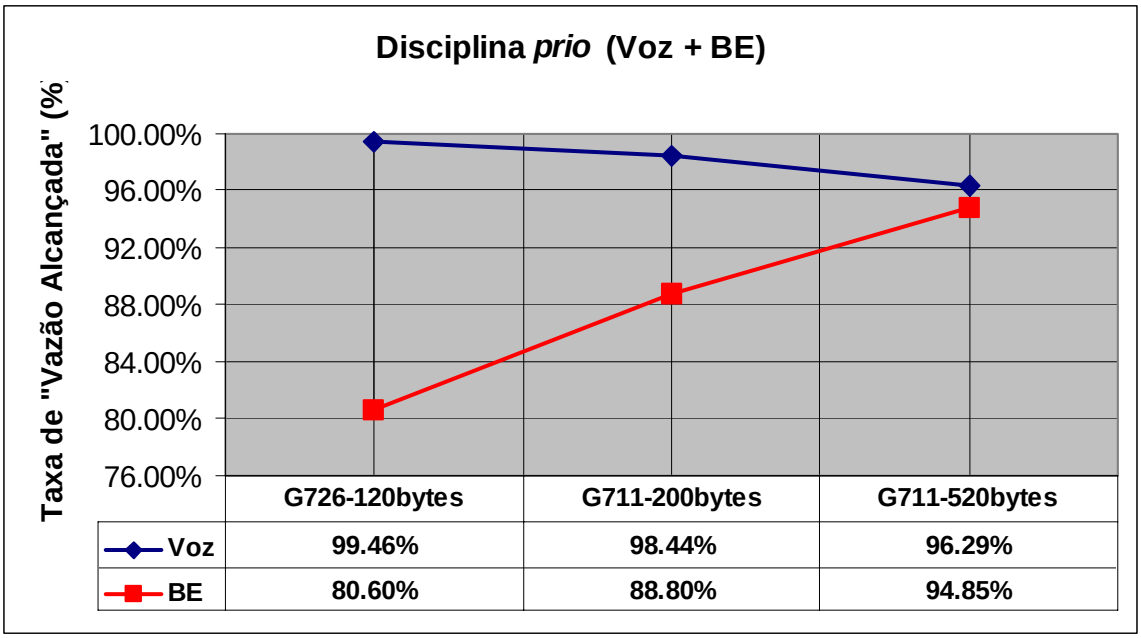


Figura 5-9 – Taxa de “vazão alcançada”, com a disciplina *prio*.

5.1.4 A disciplina *prio* num ambiente de sobrecarga

Como já descrito, os resultados de delay entre as modelagens, mostrados na Figura 5-5, apresentaram, por conta do ambiente controlado e com poucos pontos geradores de atrasos (roteamento, distanciamento da rede, etc), valores discretamente diferentes. Buscou-se então uma forma onde se pudesse visualizar melhor esses valores. Imaginou-se que, sobrecarregando a rede, isso poderia ocorrer, uma vez que a sobrecarga força mais a presença dos mecanismos de qualidade de serviço. Optou-se então em manter as mesmas configurações anteriores, reduzindo-se apenas os valores de reserva de banda no roteador interno em 100%. O cenário de sobrecarga passou então a ter a configuração mostrada na Tabela 5-4.

Os valores experimentados no cenário de sobrecarga são apresentados na Figura 5-10 e Figura 5-11, servindo também para ratificar os resultados inicialmente analisados num ambiente menos carregado. Vale comparar a Figura 5-5 e a Figura 5-10(a) e observar como é possível visualizar, na última, uma acentuação maior nos valores de atraso de voz. A Figura 5-10(b) mostra, em comparação à Figura 5-6, o quão o tráfego de melhor esforço se degrada com taxas sobrecarregadas de transmissão.

Ambiente de sobrecarga	
Tipo de Vocoder	**Reserva de banda em QoS5 (Kbps), através da configuração da disciplina *cbq***
G726-120b	**1025.48**
G711-200b	**1004.00**
G711-520b	**1021.03**

Tabela 5-4 – Reservas de banda no roteador interno do domínio DS – Cenário de sobrecarga.

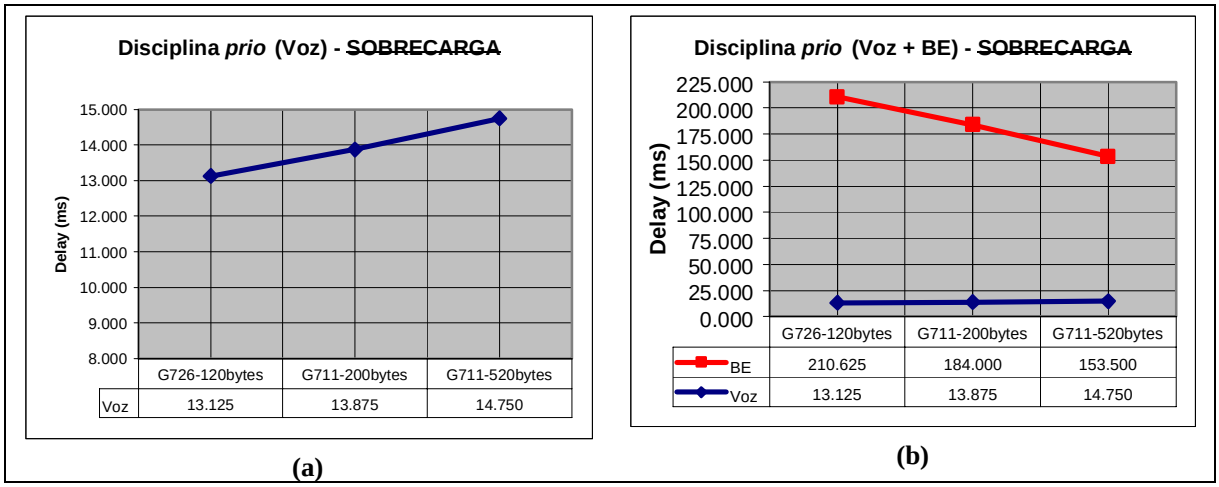


Figura 5-10 – Delay com a disciplina *prio*. – Cenário de sobrecarga.

Confrontada à Figura 5-7, a Figura 5-11(a) demonstra valores mais elevados na variação dos atrasos entre os pacotes (principalmente em relação ao tráfego de *background*), motivados pela diminuição dos recursos de banda no roteador interno.

Já a Figura 5-11(b) e a Figura 5-11(c) mostram como os fluxos prioritários, num ambiente de sobrecarga, perdem mais pacotes e, conseqüentemente, não conseguem atingir melhores níveis de vazão, não ultrapassando o limite de 91% (taxa de “vazão alcançada”).

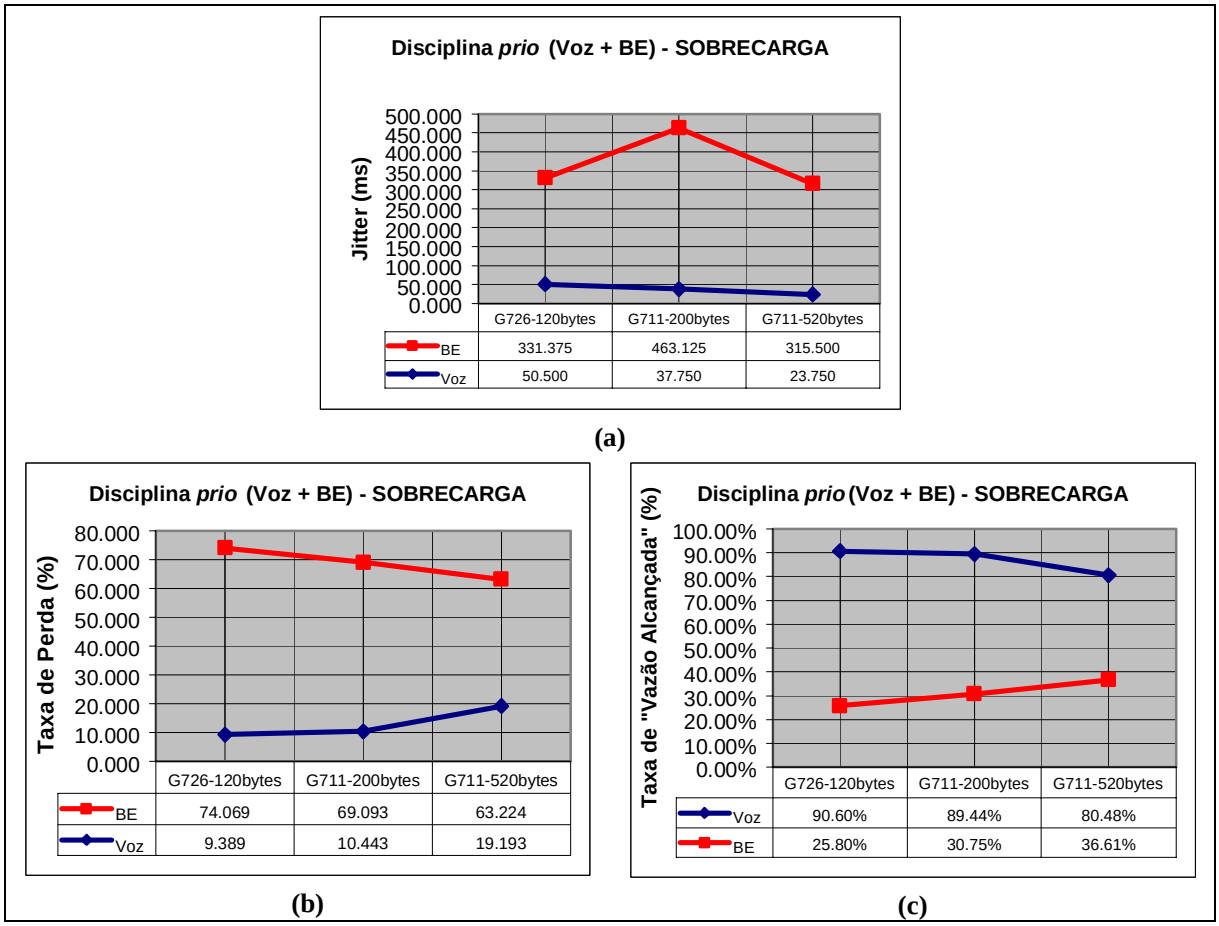


Figura 5-11 – Jitter (a), taxas de perda (b) e de "vazão alcançada" (c), com *prio*. – Cenário de sobrecarga.

5.2 Aplicação da disciplina *bfifo*

5.2.1 Estudo do delay

Os atrasos sofridos pelo tráfego de voz ganham, com a disciplina *bfifo*, uma nova configuração. Ou seja, atingem valores significativamente elevados e são decrescentes com o aumento do tamanho dos pacotes, tendo a modelagem do vocoder G.711 (520bytes) apresentado o melhor desempenho. A Figura 5-12(a) ilustra isso, onde as linhas referentes aos tráfegos EF e BE apresentam a mesma tendência, com os valores de voz sendo ligeiramente piores que os valores de melhor esforço.

Aqui, ao contrário da disciplina ***prio***, o tráfego de voz não obteve tratamento diferenciado, o que quer dizer que não foi dada prioridade no escalonamento de seus

pacotes, que competiram, de forma direta nas filas de tamanhos fixos[51], com os pacotes de melhor esforço. A falta de uma priorização dos fluxos de voz é um fator importante para justificar a mesma tendência decrescente das linhas que representam o tráfego e, principalmente, os altos valores de atraso (acima dos 30ms) sofridos pelos agregados de voz, demonstrados na Figura 5-12(b). Pode-se, nela, comparar os resultados obtidos pela aplicação das disciplinas ***prio*** e ***bfifo***.

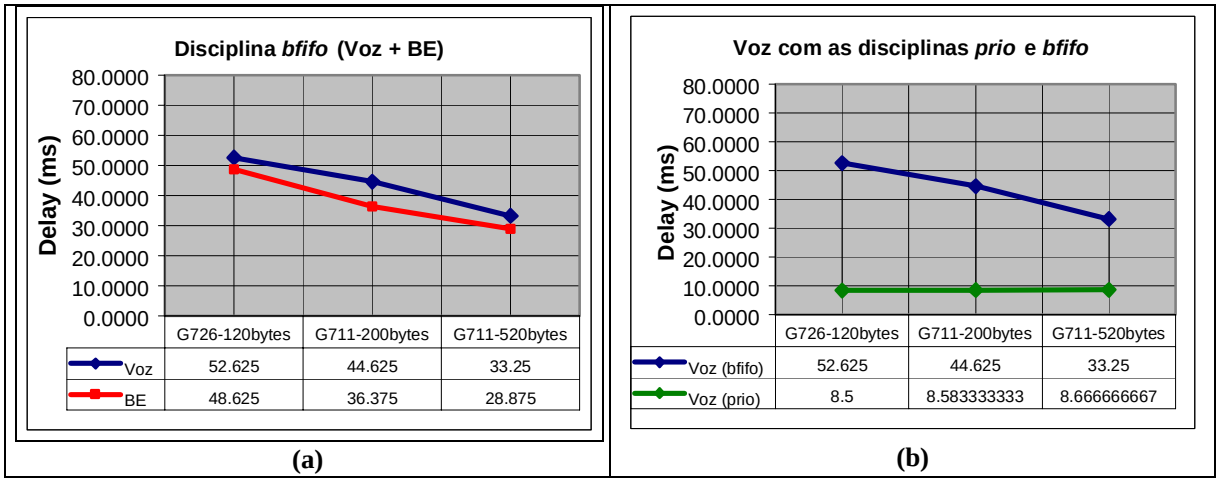


Figura 5-12 – (a) Delay com *bfifo*; (b) Comparação de delays entre as disciplinas *prio* e *bfifo*

Além disso, pode-se explicar o decrescimento dos valores em função da concorrência entre pacotes de voz e BE injetados na rede. A Tabela 5-5 mostra, para cada tipo de vocoder modelado, o número de pacotes de voz e de melhor esforço efetivamente gerados por segundo e o número de pacotes de voz gerados para cada pacote BE. Percebe-se nessa tabela que o grau de concorrência de pacotes EF para cada pacote BE diminui com o aumento do tamanho dos pacotes de voz. Por exemplo, a modelagem do vocoder G.711 (520bytes) é a que tem um menor número de pacotes de voz concorrendo com cada pacote de melhor esforço. Este o caso em que é gerado o menor número de pacotes na rede (voz e BE), com menos disputa entre eles, o que contribui para que os pacotes característicos desse vocoder cheguem ao nó destino num tempo menor em relação aos demais.

A taxa de geração do tráfego de perturbação é sempre maior que a dos agregados de voz. Com isso, é lógico imaginar que, sem priorização de tráfego, as filas sejam mais ocupadas por pacotes BE do que por pacotes de voz, contribuindo para que os atrasos médios do tráfego de melhor esforço sejam menores do que os de voz.

[51] Fila fixa BFIFO gerenciada pelo TC, com comprimento de 16.300bytes, e fila de transmissão da interface de saída, limitada, por default, em 100 pacotes e com MTU de 1500bytes.

Um outro fator que ajuda a caracterizar isto é a natureza da fonte de geração do tráfego. Considerando que os fluxos de voz são ON-OFF (gerados em 20% da banda total reservada) e que o fluxo de melhor esforço é CBR (gerado em 80% da banda total reservada), os momentos de silêncio do tráfego de voz são parcial ou inteiramente preenchidos pelo fluxo BE, o que atrasa os pacotes de voz. A intensidade de tal atraso vai depender do número momentâneo de fluxos em estado OFF.

Pacotes gerados na rede			
	Nº de pacotes efetivamente gerados por segundo (pps)		**.**
Tipo de Vocoder	**Agregado de Voz (ON-OFF)**	**Fluxo BE (CBR)**	**Nº de pacotes EF gerados para cada pacote BE**
G726-120b	517.55	174.40	**2.95**
G711-200b	303.85	170.75	**1.77**
G711-520b	119.45	173.64	**0.68**

Tabela 5-5 – Razão entre pacotes EF e BE gerados na rede.

5.2.2 Estudo do jitter

A Figura 5-13(a) demonstra o gráfico da variação média do delay, onde se pode perceber um decréscimo nos valores de voz à medida que os pacotes aumentam de tamanho. Na verdade, voltando aos experimentos com a disciplina ***prio***, vê-se, claramente, uma semelhança nos pontos de jitter entre as disciplinas estudadas – vide Figura 5-13(b). A diferença básica é que, com ***bfifo***, os valores são mais elevados, já que, como explicado antes, os pacotes modelados de voz não são prioritários. Logo, competem bem mais com os pacotes de melhor esforço que com a disciplina ***prio***, o que contribui para uma maior variação média dos atrasos entre os pacotes.

Diante disso, pode-se perfeitamente atribuir os resultados desta sessão às mesmas justificativas levantadas para a disciplina ***prio***. Ou seja, quanto mais pacotes injetados na rede (pacotes menores) maior é a probabilidade de que esses pacotes sofram variações nos atrasos de enfileiramento, com uma miscelânea maior de pacotes de voz com os de melhor esforço. Não se deve esquecer que pacotes maiores, como têm rajadas mais longas, lidam com atrasos de enfileiramento mais próximos e, por isso, apresentam melhor desempenho quanto ao jitter.

Ademais, há ainda a contribuição da natureza variável da fonte geradora dos fluxos de voz (atividade-silêncio), que possibilita que o tráfego contínuo de fundo preencha a banda nos momentos de não atuação dos fluxos de voz. Isto permite que, quanto menor o tamanho dos pacotes de voz, mais pacotes estarão misturados na rede, aumentando o valor médio de jitter daqueles pacotes.

Conforme Figura 5-13(a), o jitter de melhor esforço, por sua vez, praticamente não sofre maiores variações nos testes, provavelmente porque o tráfego BE tenha a mesma configuração em todos os experimentos.

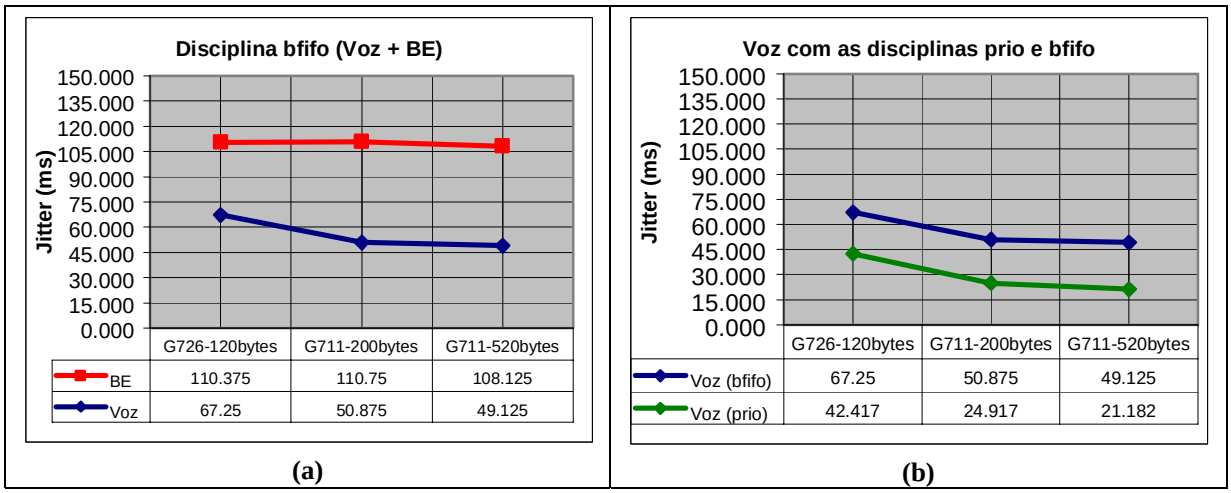


Figura 5-13 – (a) Jitter com *bfifo*; (b) Comparação de jitters entre as disciplinas *prio* e *bfifo*

5.2.3 Estudo da taxa de perda de pacotes

Os valores das taxas de descarte de pacotes de voz e de BE (Figura 5-14-a), assim como no estudo do delay médio, decrescem com o aumento do tamanho dos pacotes. Também nesta métrica o vocoder G.711 (520bytes) apresenta o de melhor desempenho entre as modelagens analisadas na disciplina ***bfifo***. Essa redução das taxas pode ser explicada em função do transbordamento das filas de transmissão de tamanho fixo, mais evidente à medida que os pacotes diminuem de tamanho, quando então, para garantir a mesma carga de rede, um maior número de pacotes por segundo é injetado na rede.

Além disso, a falta de um disciplinamento prioritário aos fluxos de voz os mantém na mesma tendência decrescente que o tráfego de melhor esforço, já que ambos são tratados indistintamente.

Percebe-se ainda na Figura 5-14(a) que as taxas de perdas de voz são, em comparação às de melhor esforço, consideravelmente mais elevadas, principalmente para o vocoder G.726, que gera um maior número de pacotes. Isso se justifica em função de dois fatores, intrinsecamente relacionados: não há tratamento diferenciado entre o tipo de tráfego e a carga de transmissão dos fluxos BE é sempre superior à dos agregados de voz. É o que faz com que os pacotes de melhor esforço tenham uma maior ocupação das filas e, conseqüentemente, apresentem menores taxas de descarte. Além disso, os pontos de voz e de BE mais se aproximam na modelagem G.711 (520bytes), quando então há menos pacotes na rede, menor relação de concorrência pela ocupação das filas e, com isso, menores taxas de descarte.

A Figura 5-14(b) compara as perdas de voz das disciplinas ***prio*** e ***bfifo***, onde se pode observar comportamentos bastante distintos e discrepâncias nos resultados.

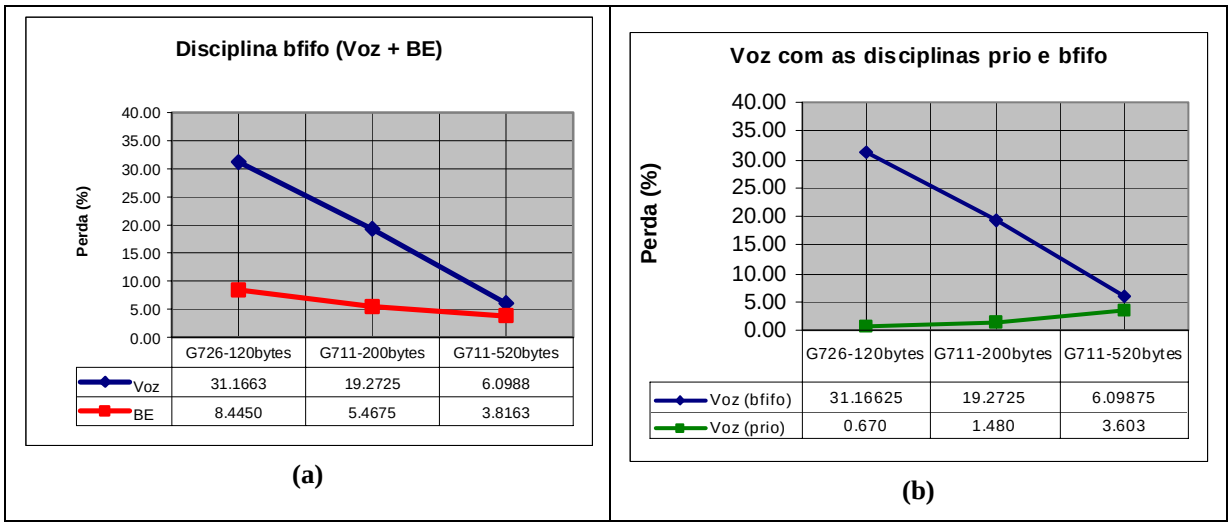


Figura 5-14 – (a) Taxa de perdas com *bfifo; (b)* Comparação de taxas de perdas entre as disciplinas *prio* e *bfifo*

5.3 Conclusão do capítulo

Iniciou-se este capítulo com um estudo experimental em que disciplinas de escalonamento foram aplicadas (*prio*, *sfq* e *pfifo*), com o intuito principal de comprovar a capacidade do ambiente em diferenciar tráfego. Tal estudo serviu como pré-requisito para os próximos experimentos. Após isso, partiu-se para estudar o desempenho de fluxos padronizados de distintos vocoders de áudio, em ambientes com e sem qualidade de serviço.

A aplicação do escalonador *prio* significou a presença de um ambiente diferenciado, no qual se percebeu que a modelagem do vocoder G.726, com pacotes de 120bytes, apresentou os menores atrasos e os menores índices de descarte de pacotes, mas, no entanto, apresentou as mais altas variações de atraso entre pacotes. Ou seja, o aumento do tamanho dos pacotes afetou diretamente no atraso e inversamente no jitter do tráfego modelado de voz. Conclui-se que, com DiffServ no ambiente montado, os melhores desempenhos foram observados com a modelagem de tráfego com pacotes de tamanhos menores.

De uma outra forma, mantendo-se as mesmas condições de transmissão do tráfego do experimento com *prio*, aplicou-se no *testbed* a disciplina de escalonamento do tipo FIFO (bfifo), que representou a ausência de qualidade de serviço. Observou-se um comportamento diferente, onde o vocoder G.711, com pacotes de 520bytes, alcançou os melhores resultados em todas as métricas de QoS estudadas. Isto é, sem diferenciação de serviço, vocoders com pacotes maiores se sobressaem em relação às modelagens com pacotes menores.

6. Estudo da continuidade dos fluxos

Este capítulo analisa aspectos relacionados à continuidade dos fluxos, abordando mais precisamente dois estudos: o primeiro investiga o desempenho de tipos de fontes geradoras distintas e o segundo analisa especificamente o comportamento de agregados gerados por fontes do tipo atividade-silêncio.

6.1 Desempenho de fontes geradoras distintas

Pretende-se, neste estudo, comparar os níveis de eficiência alcançados por fluxos prioritários de voz, gerados tanto por origens **contínuas (CBR)** como por fontes do tipo **atividade-silêncio (ON-OFF)**.

Para os fluxos prioritários de voz, variou-se o tamanho dos pacotes em **80, 160, 320 e 800 bytes**, sendo que, para cada variação desta, foi gerado apenas 01 fluxo de perturbação, do tipo contínuo (CBR), com pacotes de tamanhos fixos de 1.470 bytes. Aplicou-se um ambiente de diferenciação, através da utilização da disciplina de serviço ***prio***, com 2 bandas. À banda prioritária foi anexada uma fila FIFO de 1.600 bytes, para voz, e à banda de menor prioridade, para acomodar o tráfego de fundo, se anexou também uma fila FIFO, mas com 14.700 bytes. Vale ressaltar ainda que todos os fluxos aqui analisados foram transportados através do protocolo UDP.

Da mesma forma que no estudo dos vocoders, como não se controla a quantidade exata em gerações do tipo ON-OFF, foram usadas nesta análise taxas de geração baseadas em **percentagens proporcionais**. Ou seja, os fluxos de voz ocupam sempre 20% da taxa total gerada e o restante é preenchido por tráfego de perturbação do meio. Para garantir igualdade entre as variações, as reservas de banda efetuadas no roteador interno foram proporcionais às taxas totais geradas (voz + BE). Para o tráfego de voz do tipo CBR, para efeito de comparação com as gerações ON-OFF, utilizou-se o mesmo raciocínio, injetando-se 20% de voz e 80% de tráfego BE.

A Tabela 6-1 apresenta o *planejamento do estudo experimental*, mostrando, para cada tamanho de pacote, conforme o princípio proporcional já descrito, as taxas de geração e a reserva de banda no roteador interno, assim como a quantidade de fluxos

compondo os agregados de voz. Pode-se perceber, a partir da relação entre a vazão total gerada (voz + BE) e a banda reservada no roteador interno, que o cenário adotado para este estudo foi o de **sobrecarga**, já que foram reservados, em todos os casos, apenas 39.06% de banda em relação à taxa total gerada. Uma melhor visualização das métricas de QoS foi o que motivou a adoção deste cenário de sobrecarga da rede.

		Tipo de tráfego	Tipo de fonte	Tamanho dos pacotes (bytes)	Nº de fluxos utilizados	Vazão efetiva em cada fluxo (Kbps)	Vazão total efetiva por tráfego (Kbps)	Reserva de banda no roteador interno – *cbq/prio* (Kbps)
1) 80 bytes	**1.1**	**Voz**	**ON-OFF**	**80**	19	26.40	**501.60**	**979.53**
		BE	**CBR**	**1470**	1	2006	**2006**	
	1.2	**Voz**	**CBR**	**80**	8	64.00	**512**	**1000**
		BE	**CBR**	**1470**	1	2048	**2048**	
2) 160 bytes	**2.1**	**Voz**	**ON-OFF**	**160**	19	26.40	**502.00**	**980.47**
		BE	**CBR**	**1470**	1	2008	**2008**	
	2.2	**Voz**	**CBR**	**160**	8	64.00	**512**	**1000**
		BE	**CBR**	**1470**	1	2048	**2048**	
3) 320 bytes	**3.1**	**Voz**	**ON-OFF**	**320**	19	26.40	**502.46**	**981.04**
		BE	**CBR**	**1470**	1	2009	**2009**	
	3.2	**Voz**	**CBR**	**320**	8	64.00	**512**	**1000**
		BE	**CBR**	**1470**	1	2048	**2048**	
4) 800 bytes	**4.1**	**Voz**	**ON-OFF**	**800**	19	26.53	**503.90**	**983.95**
		BE	**CBR**	**1470**	1	2015	**2015**	
	4.2	**Voz**	**CBR**	**800**	8	64.00	**512**	**1000**
		BE	**CBR**	**1470**	1	2048	**2048**	

Tabela 6-1 – Planejamento do estudo

É importante lembrar ainda que, para os fluxos do tipo ON-OFF, em cada segundo de teste, 400ms foram utilizados para geração de pacotes e o restante (600ms) foi de silêncio.

6.1.1 Estudo do delay

A Figura 6-1 apresenta os atrasos médios em uma direção dos agregados de voz, gerados por fontes ON-OFF e CBR, em tamanhos de pacotes variados. São

percebidos, de uma maneira geral, dois fatores importantes: o primeiro é que fluxos de voz com origem atividade-silêncio sofrem mais atrasos que os gerados por fontes contínuas e o segundo é que o tamanho dos pacotes de voz, em sua essência, influencia nos resultados.

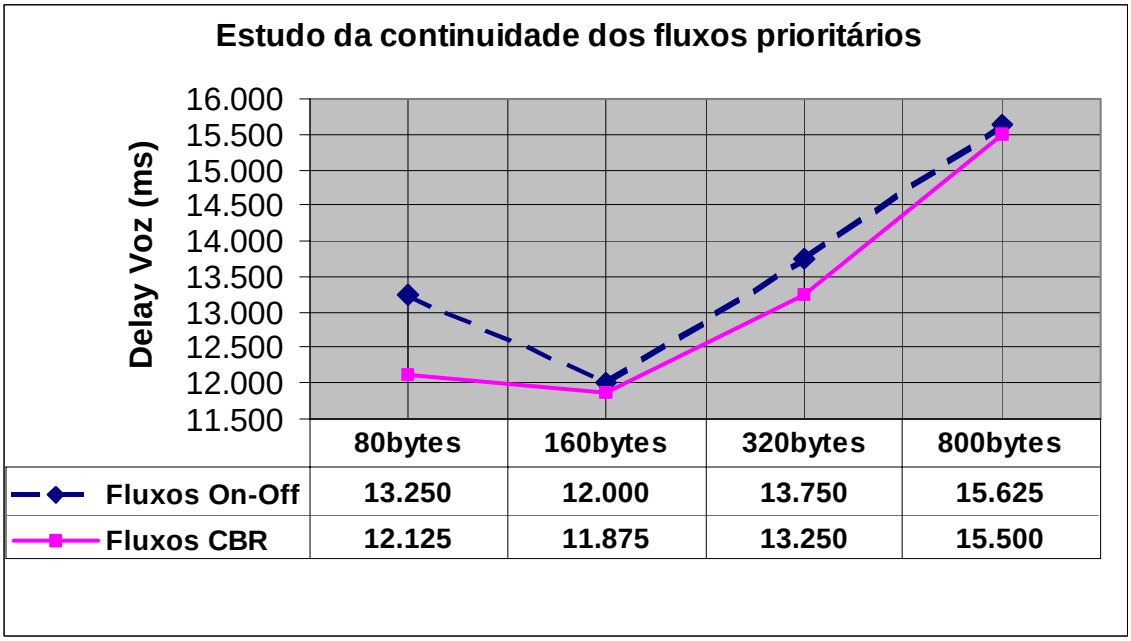
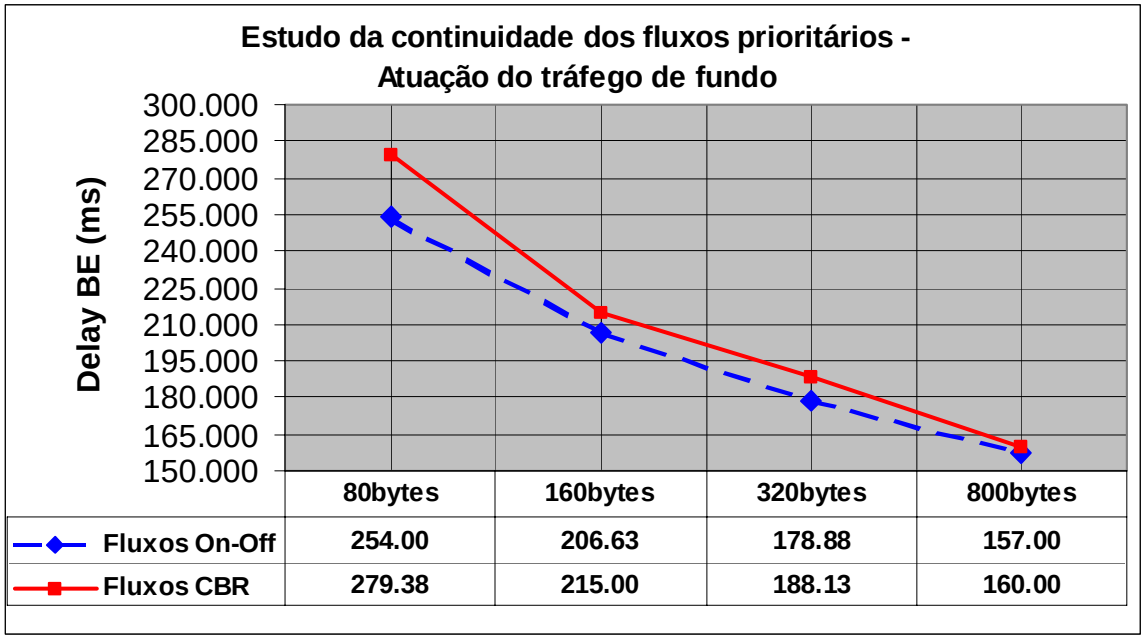


Figura 6-1 – Delay médio dos fluxos de voz.

Como já se sabe, o escalonador utilizado no roteador interno é o de enfileiramento com prioridade estrita, tendo os fluxos de voz maior preferência do que o fluxo de melhor esforço. Na verdade, isso quer dizer que os pacotes de fundo são servidos apenas quando a fila mais prioritária estiver ociosa. No entanto, pacotes de voz chegando em fila vazia devem aguardar o encaminhamento de pacote de fundo, se houver, e isso contribui para a elevação do atraso médio dos pacotes EF. Percebe-se então que, de qualquer forma, há uma certa afluência entre os fluxos. Tendo-se tráfego de voz com origem ON-OFF, os pacotes BE apresentam mais chances de estar ocupando a fila de transmissão, uma vez que pacotes EF não são continuamente injetados na rede. Isso contribui para uma concorrência maior entre os pacotes e, conseqüentemente, para a ocorrência de atrasos médios mais elevados com tráfego de voz do tipo atividade-silêncio do que com tráfego de voz com origem CBR.

Agora, com relação à atuação dos pacotes de melhor esforço, estes, conforme se pode perceber na Figura 6-2, quando associados ao tráfego EF do tipo ON-OFF, apresentam menores atrasos do que quando injetados juntamente com fluxos contínuos de voz. É a comprovação de que, na companhia de fluxos prioritários de origem atividade-

silêncio, há uma maior ocupação de pacotes BE na rede, prejudicando a atuação do tráfego prioritário de voz[52]. Na verdade, os altos valores médios de atrasos dos pacotes maiores de voz refletem positivamente nos atrasos dos fluxos de fundo, que, não importando a fonte de geração EF, decrescem com o aumento do tamanho dos pacotes de voz.



	80bytes	160bytes	320bytes	800bytes
Fluxos On-Off	254.00	206.63	178.88	157.00
Fluxos CBR	279.38	215.00	188.13	160.00

Figura 6-2 – Delay médio do tráfego de fundo (BE).

Nota-se também na Figura 6-1 que o tamanho dos pacotes influencia diretamente no desempenho do agregado de voz. Assim como no estudo dos vocoders (Capítulo 5), quanto maior é o tamanho dos pacotes, mais tempo será gasto para se gerar e, conseqüentemente, para se transmitir esses pacotes na rede, contribuindo assim para a ocorrência de atrasos maiores, independentemente do tipo da fonte geradora dos fluxos. Com base no mesmo raciocínio da Tabela 5-3, isso é demonstrado na Tabela 6-2 e na Tabela 6-3, que, a partir do número de pacotes efetivamente gerados por segundo[53], apresenta os tempos necessários para a geração dos pacotes, os quais, como se pode ver nas colunas referentes aos pacotes EF (com CBR e com ON-OFF), crescem com o aumento do tamanho dos pacotes.

[52] Além dos atrasos, o tráfego BE, quando injetado juntamente com o agregado de voz do tipo ON-OFF, apresentou taxas de descarte menores que com os agregados CBR, ajudando a comprovar a maior ocupação de pacotes de fundo quando injetados com fluxos de voz atividade-silêncio.

[53] Pode-se, teoricamente, calcular o número de pacotes por segundo gerados na rede. No entanto, na prática, os números de pacotes gerados podem, por ineficiência da ferramenta, não coincidir exatamente com os valores calculados. O termo "*efetivamente gerados*" significa o registro da quantidade de pacotes gerados em laboratório e, não, valores teóricos.

Percebe-se ainda na Figura 6-1 que, com pacotes de voz com tamanhos de 80 bytes, os atrasos sofrem uma certa elevação. Isto pode ter sido uma conseqüência das significativas taxas de descarte de pacotes de 80 bytes (ver Figura 6-4), os quais foram gerados, para manter a mesma carga, em maior quantidade do que com os demais tamanhos.

Fluxos de voz do tipo ON-OFF				
	Número de pacotes efetivamente gerados por segundo (pps)			
Tamanho de pacote (bytes)	Agregado de Voz **ON-OFF**	Fluxo BE (CBR)	**Tempo para gerar 1 pacote EF (ms)**	Tempo para gerar 1 pacote BE (ms)
80bytes	763.01	170.58	**1.311**	5.862
160bytes	379.37	170.75	**2.636**	5.857
320bytes	189.35	170.92	**5.281**	5.851
800bytes	74.53	171.33	**13.417**	5.837

Tabela 6-2 – Cálculo de tempo para geração de pacotes, com agregado de voz do tipo ON-OFF

Fluxos de voz do tipo CBR				
	Número de pacotes efetivamente gerados por segundo (pps)			
Tamanho de pacote (bytes)	Agregado de Voz CBR	Fluxo BE (CBR)	**Tempo para gerar 1 pacote EF (ms)**	Tempo para gerar 1 pacote BE (ms)
80bytes	800.00	174.17	**1.250**	5.742
160bytes	400.00	174.17	**2.500**	5.742
320bytes	200.00	174.17	**5.000**	5.742
800bytes	75.89	174.17	**13.177**	5.742

Tabela 6-3 – Cálculo de tempo para geração de pacotes, com agregado de voz do tipo CBR

6.1.2 Estudo do jitter

A variação média do atraso entre pacotes é mostrada na Figura 6-3. Observa-se que as curvas são decrescentes com o aumento do tamanho dos pacotes e que há uma significante diferença nos valores, tendo os fluxos de voz do tipo CBR alcançado os mais elevados níveis de jitter.

As mesmas explicações encontradas para justificar o decrescimento do valor de jitter no estudo dos vocoders serão utilizadas aqui neste estudo, já que se tratam de cenários

semelhantes, no sentido de que se varia o tamanho dos pacotes de voz e se deixa o fluxo de perturbação inalterado em todos os experimentos. Na verdade, dois fatores justificam as curvas decrescentes do jitter: o primeiro é que, com pacotes menores, vai se tendo mais pacotes na rede e, com isso, aumenta a probabilidade de se ter uma mistura variável maior de pacotes de voz e de melhor esforço na fila de transmissão; o segundo fator fica por conta de que, com pacotes maiores, as rajadas são mais longas e os atrasos de enfileiramento são mais parecidos, contribuindo para que os jitters sejam menores.

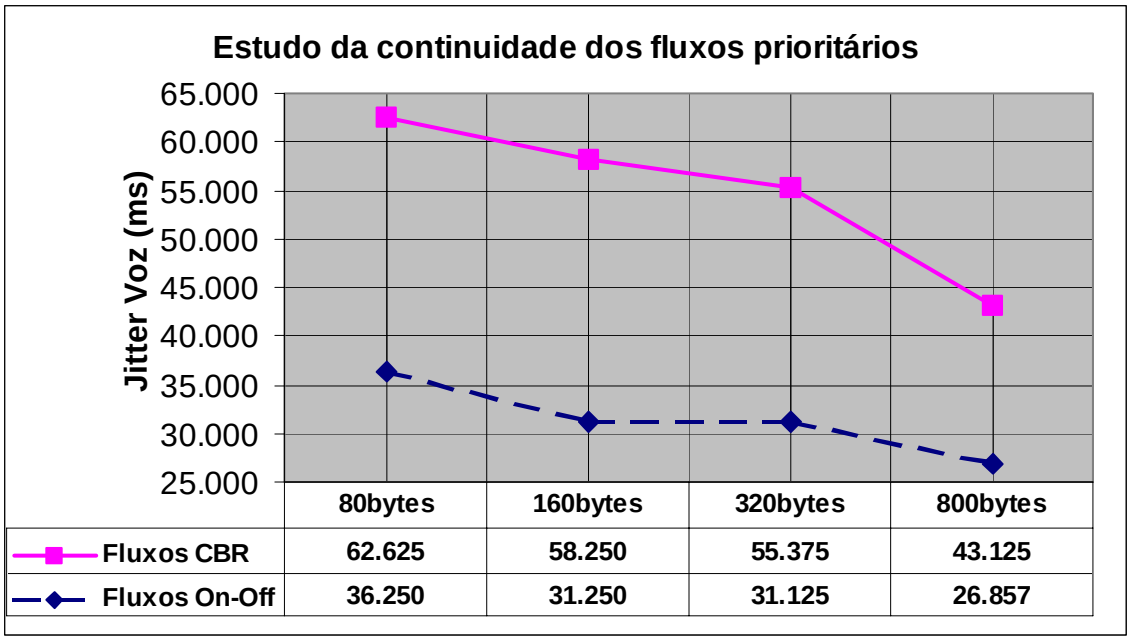


Figura 6-3 – Jitter médio do tráfego de voz.

Conforme a Figura 6-3, os valores de jitter relacionados aos fluxos de origem CBR são, para qualquer tamanho de pacote, maiores que os gerados por fontes ON-OFF. Da Figura 6-4, pode-se observar perfeitamente que os pacotes de voz do tipo CBR alcançaram taxas médias de perda razoavelmente menos elevadas que os pacotes de voz do tipo atividade-silêncio. Pode-se afirmar com isso que foram encaminhados bem mais pacotes de voz CBR do que ON-OFF (ver Tabela 6-4[54]). Ou seja, como a carga de tráfego de fundo é praticamente a mesma em ambos os casos (ver coluna “Fluxo BE” na Tabela 6-2 e na Tabela 6-3), foram tratados mais pacotes na rede com tráfego CBR do que com agregado ON-OFF. Conclui-se assim que há uma maior mistura variável entre pacotes EF e de fundo com fluxos de voz do tipo contínuo, sendo, esta, a causa dos valores de jitter com tráfego de voz do tipo CBR serem mais elevados.

[54] Pacotes encaminhados na rede são, obviamente, todos os que não foram descartados. Logo, podem ser calculados a partir das taxas de perda de pacotes da Figura 6-4 (= 100% – “taxa de descarte”).

Taxa de pacotes de voz encaminhados na rede (%)		
	Tipo de fonte de voz	
Tamanho do pacote	ON-OFF	CBR
80bytes	86.82	94.96
160bytes	93.23	99.41
320bytes	90.21	97.92
800bytes	81.94	88.82

Tabela 6-4 – Percentuais de pacotes CBR e ON-OFF encaminhados na rede.

6.1.3 Estudo da taxa de perda de pacotes

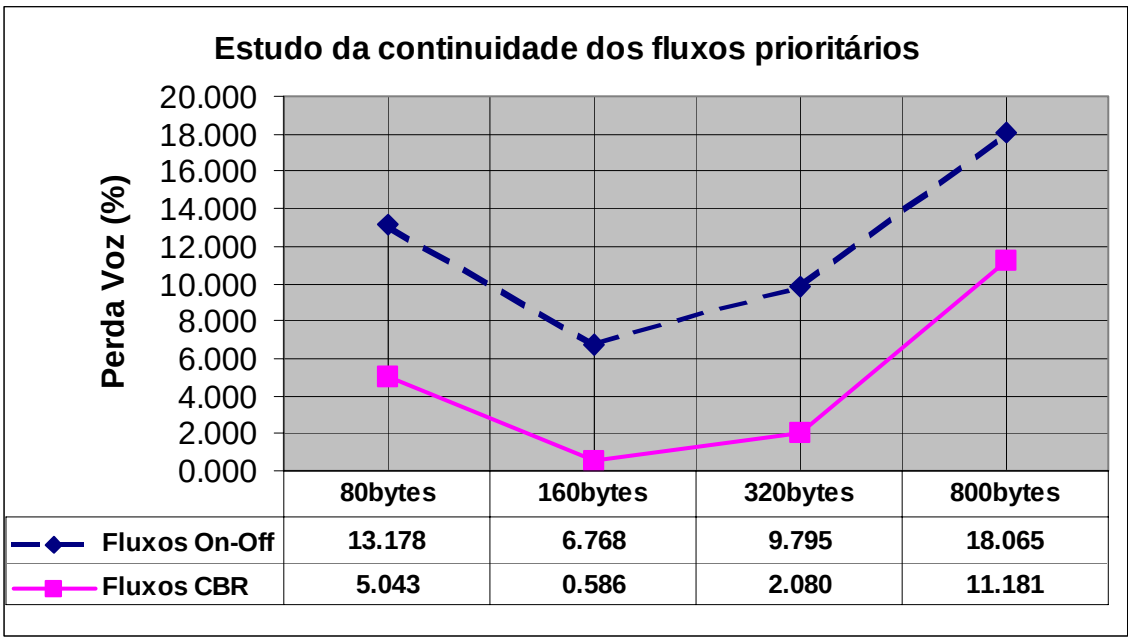


Figura 6-4 – Taxa média de descarte de pacotes do tráfego de voz.

Pacotes de voz do tipo ON-OFF foram os mais descartados, conforme Figura 6-4, muito possivelmente devido à natureza das gerações dos pacotes. Como já se sabe, com o princípio estrito da disciplina ***prio***, pacotes de fundo serão servidos apenas nos momentos de ociosidade da fila mais prioritária. Fluxos de voz gerados por fontes atividade-silêncio, mesmo que em forma de agregação, dão mais possibilidade dos pacotes BE preencherem a fila de transmissão do que fluxos CBR, porque estes, por não sofrerem interrupções momentâneas, estarão sempre sendo encaminhados pela disciplina e ocuparão mais constantemente a fila de transmissão.

6.2 Agregação ON-OFF de fluxos prioritários

Este é um estudo particular, que analisa especificamente, num ambiente com DiffServ, o desempenho de fluxos agregados, gerados por fontes do tipo atividade-silêncio. Para isso, foram comparados, na ausência de pacotes de melhor esforço, tráfegos equivalentes com agregação e sem agregação (com apenas 01 fluxo). A disciplina de serviço aplicada foi a ***prio***, sendo utilizada apenas a banda 0, com uma fila FIFO de 1.600 bytes, já que fluxos de perturbação não foram gerados. O protocolo de transporte adotado nos experimentos foi o UDP.

Tal como no estudo anterior, os pacotes sofreram alteração de tamanho (80, 160, 320 e 800 bytes) e, para manter a igualdade entre as variações[55], as cargas injetadas foram equivalentes entre os cenários com e sem agregação, conforme mostra a Tabela 6-5, que apresenta também, para comprovar tal correspondência, o número de pacotes gerados na rede. Para possíveis comparações futuras, as reservas de banda utilizadas aqui seguiram os mesmos valores do estudo da continuidade dos fluxos. Como não há concorrência com tráfego BE, os valores reservados no roteador interno são suficientemente grandes para a passagem dos fluxos estudados, os quais percorrem os nós da rede sem maiores resistências.

As Figura 6-5 e a Figura 6-6 apresentam, respectivamente, os valores médios de delay e jitter para cada tipo de tráfego, onde se pode perceber claramente o melhor desempenho dos fluxos agregados em relação ao tráfego sem agregação. As taxas médias de descarte de pacotes, mostradas na Figura 6-7, também confirmam isso. Pode-se atribuir os resultados ao fundamento básico do modelo DiffServ, em que microfluxos constituem um tráfego agregado maior, para o qual as reservas de recursos são alocadas, com este conjunto inteiro de fluxos recebendo um tratamento diferenciado.

[55] Entre os tipos de tráfego (com e sem agregação) e entre os tamanhos dos pacotes.

	Tipo de tráfego	Nº de fluxos	Taxa de pacotes efetivamente gerados por segundo	Nº total de pacotes gerados na rede[56]	Reserva de banda no roteador interno – *cbq/prio* (Kbps)
1) 80 bytes	Agregado	19	40	15.162	979.53
	1 Fluxo	1	760	15.200	
2) 160 bytes	Agregado	19	20	7.600	980.47
	1 Fluxo	1	380	7.600	
3) 320 bytes	Agregado	19	10	3.800	981.04
	1 Fluxo	1	190	3.800	
4) 800 bytes	Agregado	19	4	1.520	983.95
	1 Fluxo	1	76	1.520	

Tabela 6-5 – Estudo da agregação ON-OFF – Planejamento.

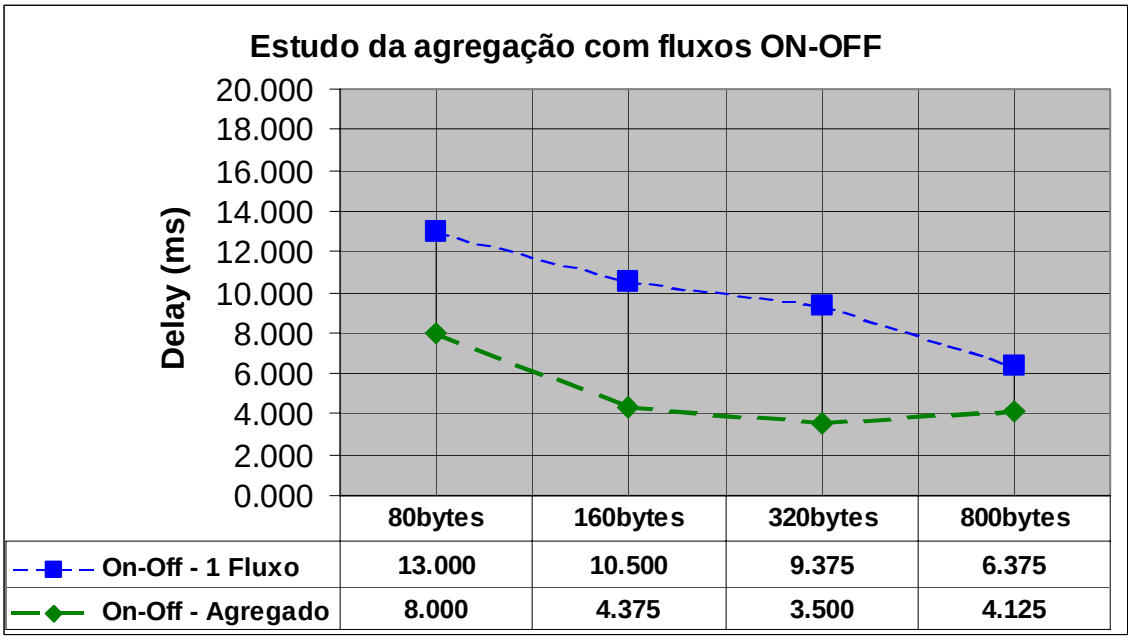


Figura 6-5 – Estudo da agregação com fluxos ON-OFF – Valores médios de delay.

[56] **Valor médio**, calculado a partir das repetições.

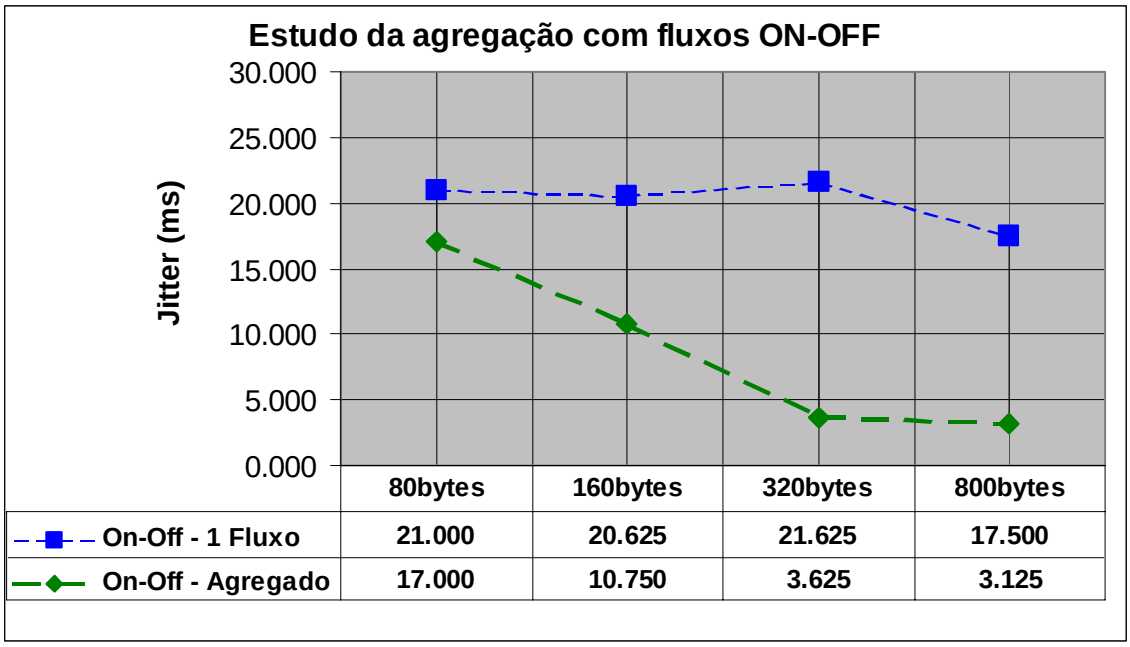


Figura 6-6 – Estudo da agregação com fluxos ON-OFF – Valores médios de jitter.

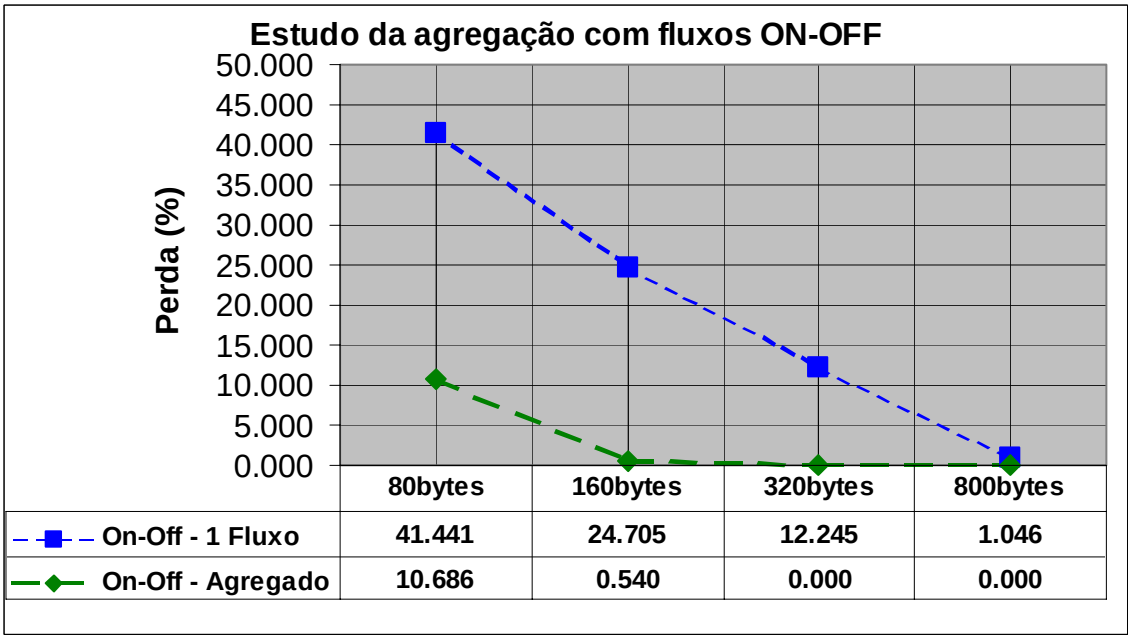


Figura 6-7 – Estudo da agregação com fluxos ON-OFF – Taxas médias de perda de pacotes.

Na verdade, chama a atenção os altos percentuais de pacotes descartados em 1 fluxo, o que acaba refletindo nas demais métricas. Ou seja, sem agregação, quanto maior é a taxa de geração de pacotes (ver Tabela 6-5), significativamente maior é a quantidade de pacotes perdidos (Figura 6-6, “On-Off - 1 fluxo”). Com 1 fluxo de 80 bytes, por exemplo, são gerados efetivamente na rede 760 pacotes em cada segundo e a taxa de descarte é de 41.4%. Já com pacotes de 800 bytes (ainda com relação ao tráfego de 1 fluxo), são gerados

efetivamente apenas 76 pacotes num segundo, sendo a taxa de descarte próxima de zero e pouco pior do que a atingida com agregação.

Do outro lado, com fluxos agregados, pode-se perceber as baixas taxas de descarte de pacote e, conseqüentemente, melhores desempenhos de delay e de jitter. Ocorre que, com tráfego agregado do tipo ON-OFF, cada fluxo que compõe esse tráfego gera poucos pacotes por segundo e os momentos de inatividade de alguns fluxos certamente são preenchidos por outros que estejam no estado ON. Desta forma, há uma saturação muito menor da fila FIFO (anexada à disciplina ***prio***) com agregado de fluxos do que com tráfego de apenas 1 fluxo, o qual, ao contrário da agregação: gera rajadas bem maiores[57] nos momentos de atividade em cada segundo, não utiliza os momentos OFF de cada segundo (600ms) e, como conseqüência, sobrecarrega as filas do *tc* e de transmissão, causando maior degradação no desempenho. É importante lembrar que isto é mais evidente à medida que as taxas de geração aumentam (pacotes menores), não importando se o tráfego é agregado ou não.

6.3 Conclusão do capítulo

Estudou-se inicialmente, neste capítulo, o comportamento de fluxos gerados por fontes contínuas e ON-OFF, numa rede com priorização de tráfego.

Variou-se o tamanho dos pacotes prioritários e se pôde perceber quanto isso influenciou diretamente no desempenho desses pacotes, com reflexos nas métricas de QoS. O aumento do tamanho dos pacotes significou, basicamente, a degradação do atraso dos fluxos de voz e a melhora dos valores de jitters desses fluxos. Nessas métricas, o tráfego ON-OFF apresentou, para todos os tamanhos de pacotes, melhor desempenho do que os fluxos contínuos (CBR).

Aplicou-se também um estudo específico para analisar, sem perturbação do meio, o fator agregação em fluxos do tipo atividade-silêncio. Observou-se que tráfego agregado (composto por 19 fluxos) alcançou, em todas as métricas e em pacotes de todos os tamanhos (80, 160, 320 e 800bytes), melhor desempenho que tráfego com apenas 1 fluxo.

[57] No exemplo estudado, tráfego com 1 fluxo gera rajadas 19 vezes maiores que o tráfego agregado.

7. Impactos do tráfego de fundo

Estudam-se, neste capítulo, os efeitos da variação do tráfego de melhor esforço no tráfego modelado de voz. Para isso, caracterizou-se, em todos os experimentos, o vocoder G.711 com *frame size* de 60ms e correspondentes pacotes com 520bytes de tamanho. Utilizou-se a disciplina ***prio*** para o escalonamento dos pacotes.

Este estudo experimental compreendeu a implementação de duas etapas. A primeira teve a finalidade de analisar a implicação, no desempenho do tráfego de voz, de fluxos de melhor esforço com protocolos de transporte e tamanhos de pacotes distintos. A fase foi complementada, para fins de comparação, com a observação do desempenho do tráfego de voz sem perturbação do ambiente. A segunda etapa do estudo teve o objetivo de avaliar os efeitos no desempenho de fluxos de voz causados pela injeção na rede de tráfego de melhor esforço do tipo FTP.

7.1 Primeira etapa

Foram aplicados, como tráfego de fundo, fluxos dos tipos UDP e TCP, sendo que, para cada um deles, variou-se o tamanho dos pacotes BE em 256, 512 e 1500bytes. Reservou-se, em todos os casos, a banda de 1500Kbps no roteador interno da rede (QoS5). 80% do tráfego total gerado na rede foi do tipo melhor esforço e, o restante, destinado aos agregados de voz, conforme se pode observar na Tabela 7-1:

Na verdade, o objetivo aqui foi comparar a influência dos protocolos de transporte com a rede, experimentando, em cada tipo de protocolo, as mesmas condições de carga. Ou seja, planejou-se inicialmente o estudo com o protocolo TCP (sem qualquer outro tráfego) e, depois, foram aplicadas exatamente as mesmas condições com o protocolo UDP. Na Tabela 7-1, as taxas geradas no tráfego de melhor esforço são distintas, dada a natureza cooperante ou adaptativa do protocolo TCP, onde não se tem controle sobre a vazão resultante, a qual depende do estado momentâneo da rede. As vazões BE conseguidas foram as que mais se aproximaram da reserva de banda fixada no roteador

interno do testbed, que foi 1.500Kbps. Também por causa da natureza do protocolo TCP é que este valor permaneceu constante para todos os tamanhos de pacotes BE[58].

Tamanho dos pacotes BE (*payload*)		**BE**	**Voz**	**Taxa total gerada na rede**	**Reserva de banda em QoS5 (Kbps), através da configuração da disciplina *cbq***
256bytes	**Taxa gerada (Kbps)**	**1334.61**	333.65 **(11 fluxos)**	**1668.26**	**1500.00**
	%	80.00%	20.00%	**100%**	
512bytes	**Taxa gerada (Kbps)**	**1560.58**	390.14 **(13 fluxos)**	**1950.72**	**1500.00**
	%	80.00%	20.00%	**100%**	
1500bytes	**Taxa gerada (Kbps)**	**1620.00**	420.40 **(14 fluxos)**	**2040.40**	**1500.00**
	%	79.40%	20.60%	**100%**	

Tabela 7-1 – Planejamento da 1ª etapa – Taxas de geração usadas nos protocolos TCP e UDP em BE e em Voz

Para a geração dos 20% restantes de tráfego da rede, equivalente aos agregados de voz – caracterizados como o G.711 (520bytes), foram necessários 11, 13 e 14 fluxos, respectivamente utilizados nos testes com pacotes BE de 256, 512 e 1500bytes. Ainda com relação ao tráfego de voz, é importante frisar que o protocolo de transporte empregado foi o UDP e que a fonte de geração utilizada foi do tipo ON-OFF.

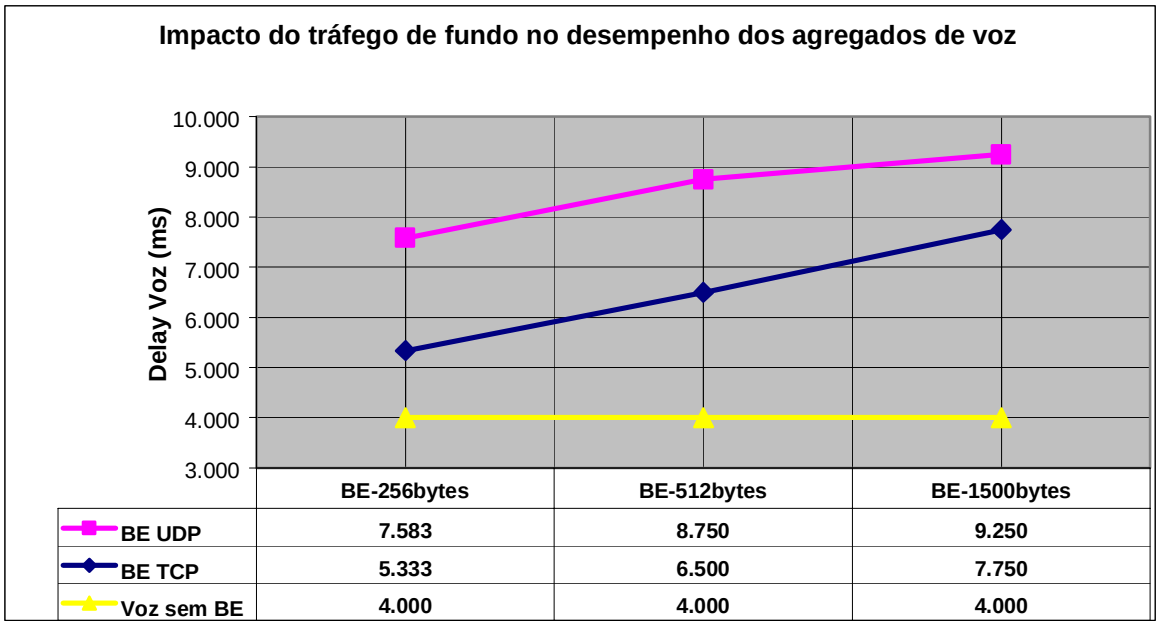
A ferramenta de geração/medição do tráfego TCP utilizada nos testes foi, como já descrito anteriormente, o NETPERF, responsável em calcular a vazão resultante (desempenho) do fluxo em estudo, a partir da medição do número de transações TCP (request/response)[59]. É importante saber que cada transação é contabilizada assim que o nó de origem receber, do nó de destino, um pacote de confirmação (*acknowledge* ou ACK), em resposta à transmissão de um pacote. Os pacotes de transmissão tiveram seus tamanhos variados (256, 512 ou 1500bytes), mas os pacotes de confirmação foram fixados num menor tamanho possível (1 byte), haja vista ter-se o mínimo de impacto na comparação com o tráfego UDP, que não tem confirmações.

[58] Alterando-se o valor de reserva, automaticamente a vazão TCP resultante se adapta em novos valores.
[59] Uma transação TCP é definida como a troca de uma única solicitação (*request*) e uma única resposta (*response*)[52] – Teste de desempenho do Netperf do tipo TCP_RR.

7.1.1 Estudo do delay

Observou-se, inicialmente, o desempenho no atraso dos agregados de voz sem a interferência de qualquer outro tráfego na rede. Injetou-se, então, de forma isolada, as mesmas cargas de fluxos de voz que a gerada, posteriormente, com o tráfego de fundo. Isto é, foram introduzidos, a princípio, apenas os agregados de voz, compostos por 11, 13 e 14 fluxos, e, diante disso, verificou-se o atraso médio de cada um deles. Pode-se visualizar, na Figura 7-1, que os valores alcançados foram pequenos e os mesmos em cada caso (4ms), já que, em nenhum dos experimentos, se gerou mais do que 29% da banda reservada no roteador interno (1.500Kbps).

Passou-se então a analisar o quão o tráfego de fundo exerce influência no desempenho dos mesmos agregados de voz observados inicialmente de forma isolada. A Figura 7-1 apresenta os resultados dos atrasos médios, diante da variação do tipo de protocolo de transporte e do tamanho dos pacotes do tráfego de melhor esforço.



	BE-256bytes	BE-512bytes	BE-1500bytes
BE UDP	7.583	8.750	9.250
BE TCP	5.333	6.500	7.750
Voz sem BE	4.000	4.000	4.000

Figura 7-1 – Impacto do tráfego BE – Valores médios de delay de voz.

Diante da elevação dos valores, pode-se perceber, claramente, como o tráfego de melhor esforço exerce uma significativa influência no desempenho dos fluxos prioritários de voz. Vê-se, ainda, que os atrasos são crescentes com o tamanho dos pacotes BE e que a geração de tráfego de fundo com protocolo UDP degrada mais a atuação dos fluxos EF que com o protocolo TCP.

Pacotes de fundo, apesar de menos prioritários, são transmitidos na rede juntamente com o tráfego de voz. Na verdade, exercem influência porque os fluxos prioritários têm origem ON-OFF e, nos momentos de menor atividade, a banda é preenchida por pacotes oportunistas do tipo BE. Quanto maior forem esses pacotes de fundo, conseqüentemente mais tempo levarão os pacotes de voz, a eles misturados, a atravessarem a rede até a máquina de destino, o que resulta na tendência crescente das linhas (BE UDP e BE TCP).

Justifica-se uma menor influência do tráfego TCP em função da própria natureza desse protocolo, uma vez que o mesmo consegue ajustar-se, através de retransmissões e de acordo com o grau de ocorrência de descartes de pacotes, sua taxa de geração, causando menos congestionamentos na rede e fazendo com que os pacotes EF permaneçam menos tempo nas filas. De uma outra forma, o protocolo UDP, sem esquema de retransmissões e adaptação ao meio, gera os pacotes a uma taxa constante de bits, sem se preocupar com congestionamentos, causando maiores atrasos nos fluxos prioritários de voz.

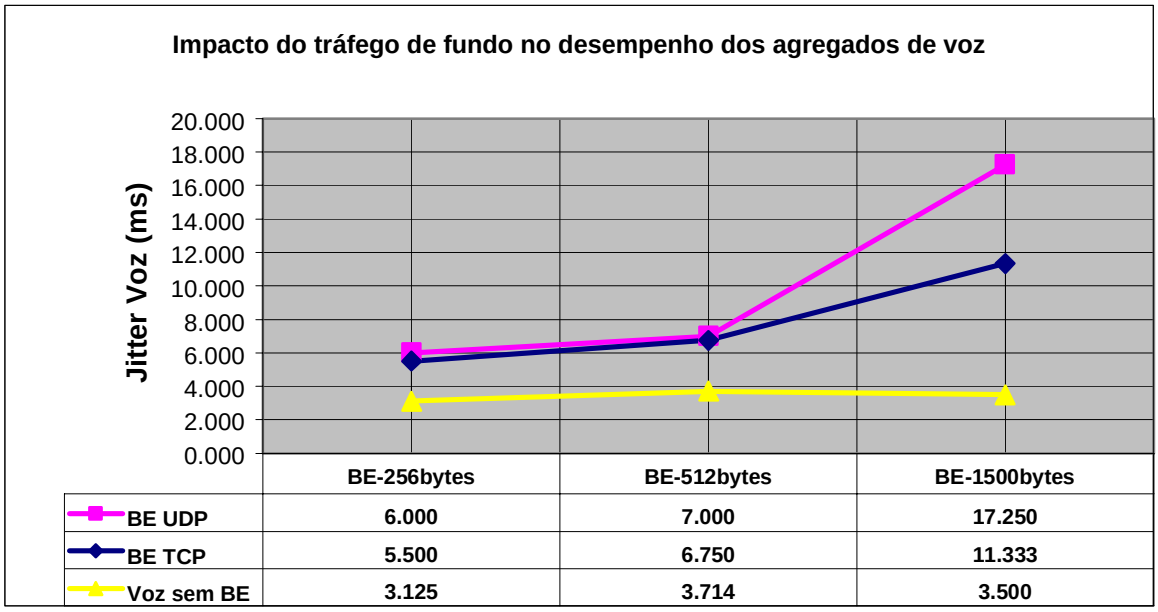
É importante lembrar ainda que, com pacotes BE com carga útil de 1500 bytes, como são transmitidos mais bytes do que a rede ethernet pode transmitir numa unidade de transmissão (MTU de 1500bytes, para *payload* e cabeçalhos), ocorre fragmentação dos pacotes e, com isso, se enfrenta mais congestionamento de rede. Esse fator é registrado na Figura 7-1, onde o atraso médio de fluxos de voz, com BE de 1500 bytes, alcança os mais altos valores. Isso vai ser refletido também nas demais métricas.

7.1.2 Estudo do jitter

A variação dos atrasos entre pacotes pode ser observada na Figura 7-2, em que se percebe também como o tráfego de fundo influencia no comportamento dos fluxos prioritários. Tal como no estudo do delay, os valores são crescentes com o aumento do tamanho dos pacotes BE e a linha referente ao protocolo UDP se posiciona sempre acima da que representa o protocolo TCP, sendo que, em BE com os tamanhos de 256 e 512bytes, essa diferença é muito discreta. No entanto, chama a atenção o aumento desproporcional no valor do jitter quando se injetam pacotes de fundo com tamanhos de 1500bytes, tanto com o protocolo UDP quanto com TCP, registrando assim como a

fragmentação de quadros ethernet pode degradar (mais no sentido de variação dos atrasos) a qualidade de serviço de fluxos de tempo real.

Nas mesmas condições, em que há fragmentação de quadros ethernet, e numa rede mais extensa geograficamente, com mais pontos de roteamento, maior número de filas e links de baixa velocidade, esses valores tenderiam a se propagar, chegando, talvez, a valores inviáveis para a execução de aplicações VoIP, o que é foco de investigação em testbeds com outras topologias.



	BE-256bytes	BE-512bytes	BE-1500bytes
BE UDP	6.000	7.000	17.250
BE TCP	5.500	6.750	11.333
Voz sem BE	3.125	3.714	3.500

Figura 7-2 – Impacto do tráfego BE – Valores médios de jitter de voz.

Pode-se observar ainda no gráfico que, com a ausência de tráfego de fundo, praticamente não há variação nos valores de jitter de voz com o aumento do número de fluxos EF, já que, além de terem banda extra à disposição, os pacotes não encontram concorrência com outros tráfegos.

Nos estudos anteriores (com a disciplina ***prio***), manteve-se o tamanho dos pacotes de fundo constante, variou-se apenas o tamanho dos pacotes de voz e percebeu-se que, quanto maior eram os pacotes de voz menor era o valor de jitter médio. A linha do gráfico foi decrescente e, como se deve lembrar, se atribuiu isso ao número de pacotes prioritários injetados na rede em relação ao tamanho desses pacotes e ao comprimento das rajadas. Neste experimento, de uma outra forma, mesmo que em condições de carga distintas, mantendo-se o tamanho dos pacotes prioritários e variando-se agora o tamanho dos pacotes BE, encontrou-se um crescimento no valor médio do jitter de voz. Aqui, a

causa é que, quanto maior o tamanho dos pacotes de perturbação, mais variações de atrasos nas filas (fila FIFO do ***prio,*** gerenciada pelo *tc*, e fila de transmissão da interface) os pacotes prioritários de voz irão sofrer. De qualquer forma, da mesma maneira que em [49], pode-se concluir sucintamente que o jitter médio é então uma função tanto do tamanho dos pacotes EF quanto do tamanho dos pacotes de fundo.

7.1.3 Estudo das taxas de perda de pacotes e de vazão

A Figura 7-3 e a Figura 7-4 demonstram, respectivamente, as taxas de "descarte" e de "vazão alcançada" para este experimento. São comprovadas as tendências de desempenho discutidas nas métricas anteriores, com as taxas sofrendo maior degradação com o aumento do tamanho dos pacotes de fundo e com o tráfego de melhor esforço com UDP exercendo maior impacto nos fluxos de voz do que o tráfego do tipo TCP. Além disso, vê-se, mais uma vez, a inversão na tendência dos valores entre essas métricas.

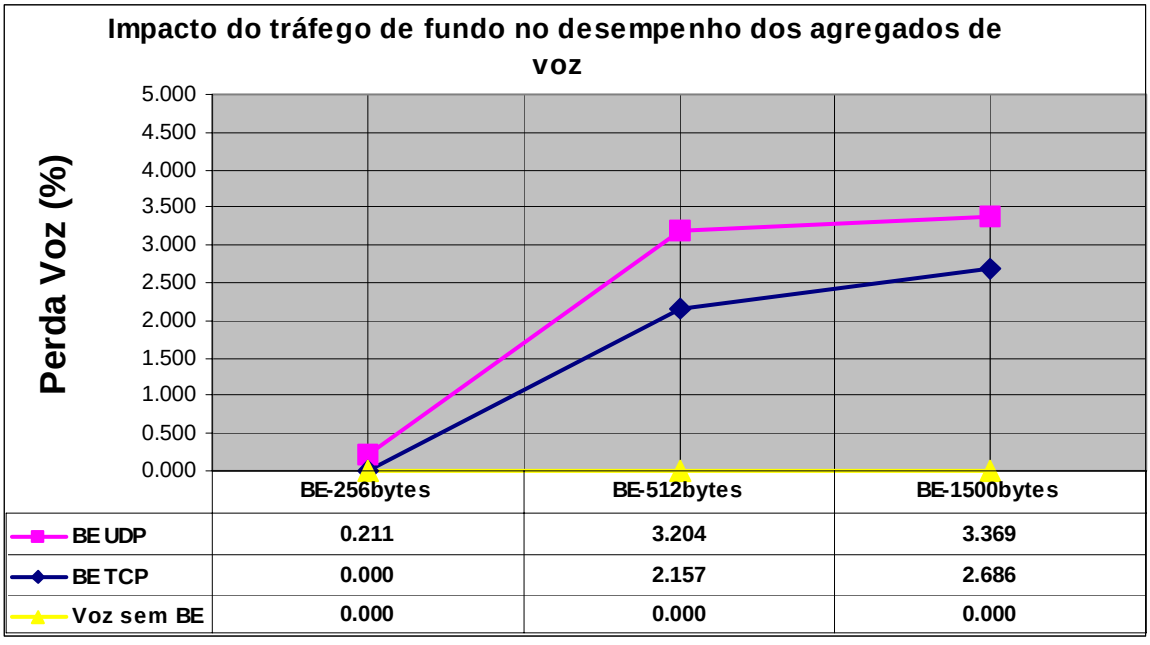
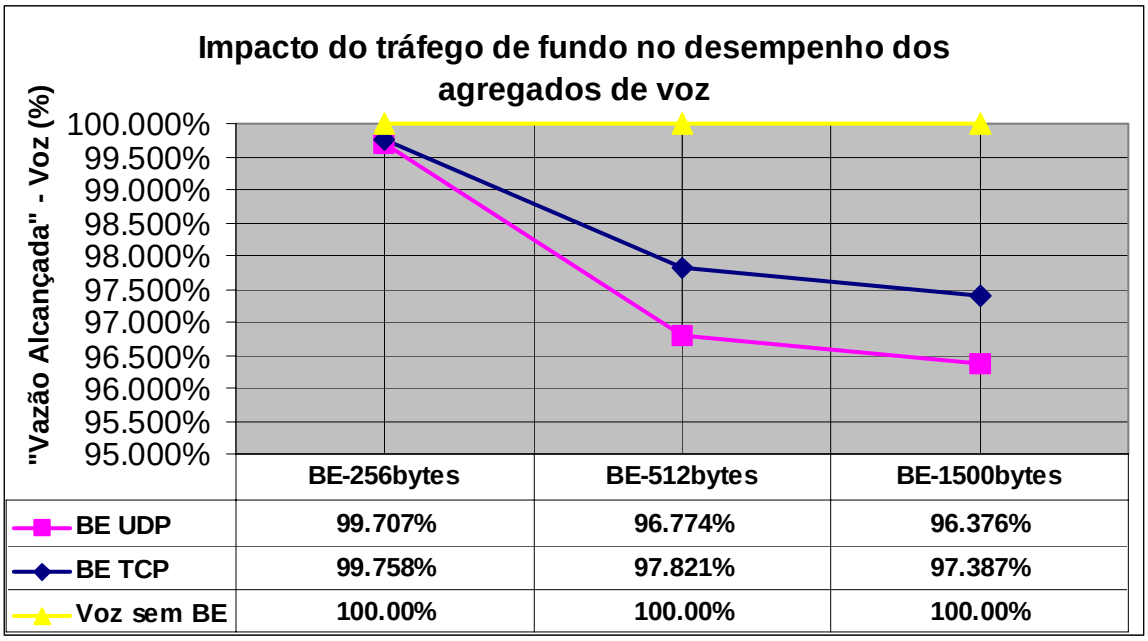


Figura 7-3 – Impacto do tráfego BE – Valores médios de perda de pacotes prioritários.

Quanto às perdas, pode-se observar na Tabela 7-1 que, assim como o número de fluxos prioritários, as taxas de geração dos tráfegos de voz e de fundo aumentam com o tamanho dos pacotes BE, mas, no entanto, a reserva efetuada no roteador interno permanece inalterada em todo o estudo (em 1500Kbps). Com isto, à medida que se aumenta a taxa total de vazão (EF + BE) da rede e conserva-se a reserva de banda no

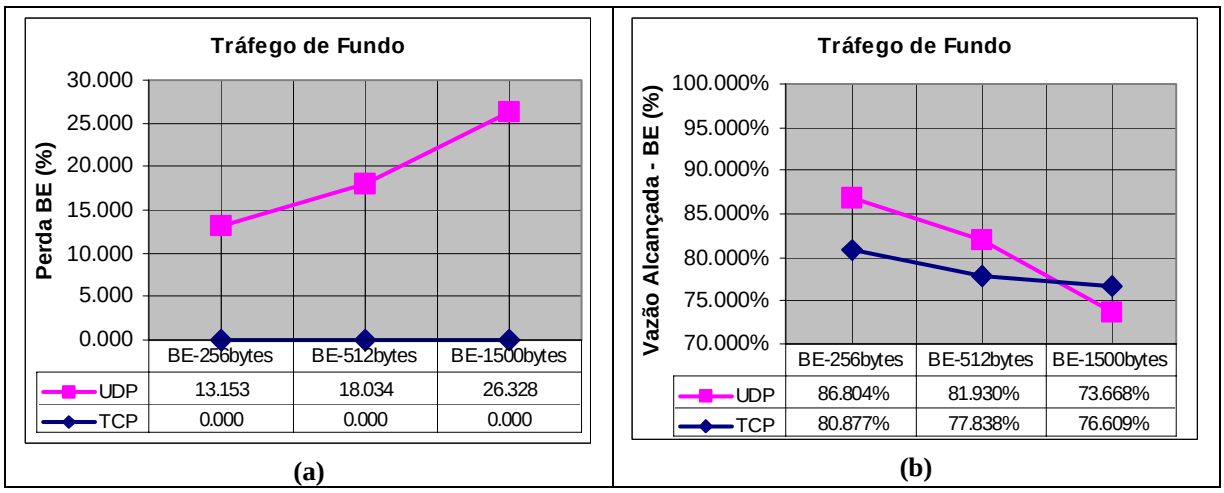
interior do domínio, maior tende a ser o número de pacotes prioritários descartados, independentemente do tipo de protocolo de transporte utilizado no tráfego BE.



	BE-256bytes	BE-512bytes	BE-1500bytes
BE UDP	99.707%	96.774%	96.376%
BE TCP	99.758%	97.821%	97.387%
Voz sem BE	100.00%	100.00%	100.00%

Figura 7-4 – Impacto do tráfego BE – Taxas médias de "vazão alcançada" de pacotes prioritários.

Da mesma forma, quanto mais pacotes de voz injetados, maior deve ser a atuação da disciplina ***prio*** em dar preferência a esses pacotes. Como conseqüência, maior é a degradação do fluxo de melhor esforço com fontes UDP, por este ser contínuo e não cooperante. A Figura 7-5(a) mostra esta tendência e a Figura 7-5(b) apresenta a taxa de "vazão alcançada" dos fluxos de perturbação injetados neste estudo. O tráfego de fundo do tipo TCP, dada sua natureza adaptativa, não apresentou pacotes descartados.



	BE-256bytes	BE-512bytes	BE-1500bytes
UDP	13.153	18.034	26.328
TCP	0.000	0.000	0.000

	BE-256bytes	BE-512bytes	BE-1500bytes
UDP	86.804%	81.930%	73.668%
TCP	80.877%	77.838%	76.609%

(a) **(b)**

Figura 7-5 – Tráfego BE – (a) Taxa de perda de pacotes; (b) Taxa de "vazão alcançada".

Vale observar que a taxa de "vazão alcançada" para o tráfego BE/UDP (Figura 7-5-b) é exatamente a inversão dos valores relativos à taxa de perdas para o mesmo tráfego

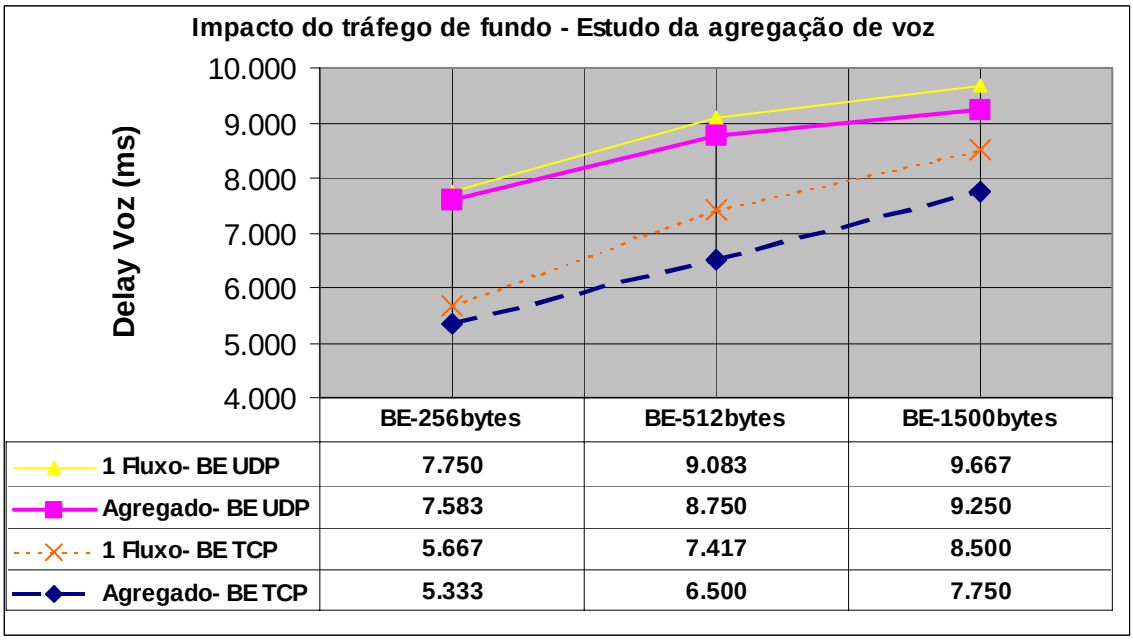
(Figura 7-5-a). No entanto, não foi o caso com os fluxos TCP, que apresentaram 0% de taxa de descarte (em todos os tamanhos) e deveriam alcançar, pelo mesmo raciocínio tomado com UDP, 100% de taxa de "vazão alcançada". Isto não aconteceu devido ao controle de transmissão do protocolo, que permanentemente oscila a taxa de bits gerados, para que pacotes nunca sejam perdidos.

7.1.4 Estudo da agregação de voz (uso de apenas 1 fluxo prioritário)

Faz-se, nesta sessão, uma análise comparativa dos resultados apresentados nos itens anteriores deste capítulo, em que foi utilizado tráfego agregado de voz. Aqui, ao contrário, monta-se um cenário marcado por apenas 1 fluxo prioritário de voz. São, na verdade, casos extremos. Ou seja, no primeiro momento se usou, dependendo do tamanho do pacote de fundo, 11, 13 ou 14 fluxos EF. No segundo caso, reproduzindo-se as mesmas condições de carga do tráfego agregado de voz (n° de pacotes gerados e vazão), se aplicou apenas 01 fluxo EF. Isto é, a configuração do tráfego de voz utilizada nesta sessão, em termos de tamanho dos pacotes[60], carga do tráfego de fundo e reserva de banda no roteador interno, foi a mesma aplicada com as agregações.

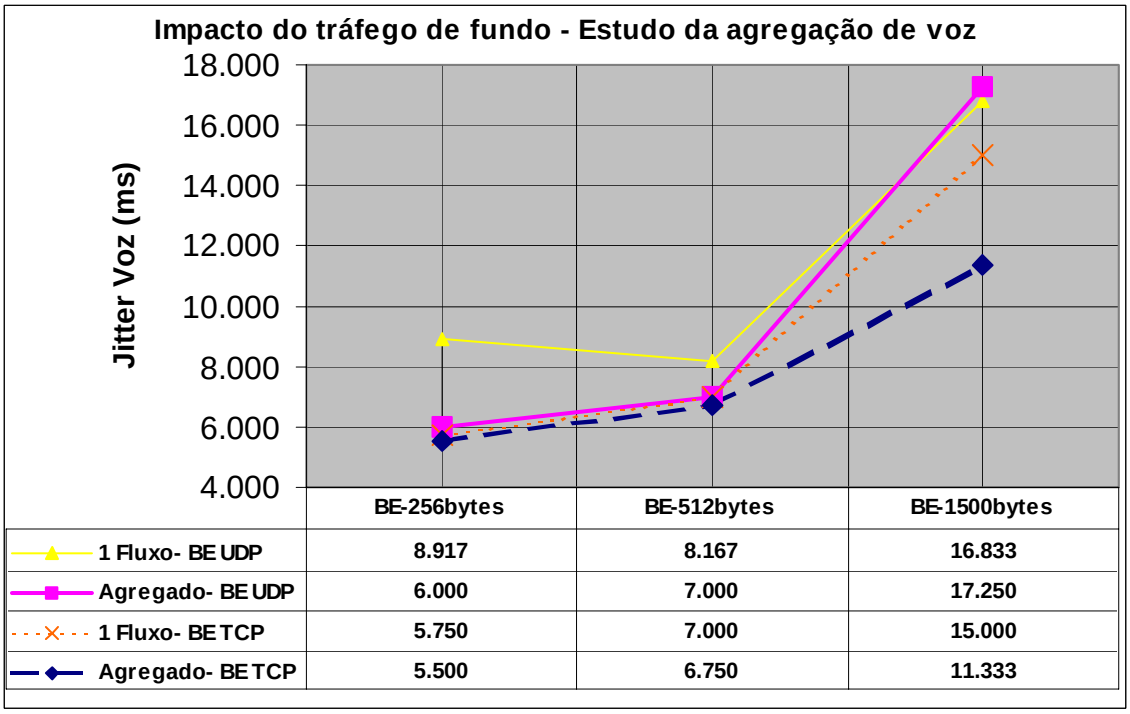
Os gráficos repetem as linhas referentes aos agregados de voz, utilizando os protocolos UDP e TCP, e incluem os valores relativos ao tráfego com 1 fluxo prioritário, também com os mesmos protocolos de transporte. A Figura 7-6 mostra, para cada caso e com base nessas informações, os valores médios de delay. Pode-se observar nessa figura que os agregados de voz, da mesma forma que no estudo da "agregação ON-OFF de fluxos prioritários" (Sessão 6.2), obtiveram melhores desempenhos que o tráfego com fluxos únicos, independentemente do tipo de protocolo de transporte. Isto se repete para os valores de jitter (Figura 7-7) e é justificado em decorrência dos fluxos agregados do tipo ON-OFF gerarem, cada um deles, poucos pacotes por segundo, com uns fluxos preenchendo as filas nos momentos de inatividade (OFF) dos outros. Assim sendo, essas filas são muito menos saturadas do que o tráfego com apenas 1 fluxo ON-OFF, o qual, por sua vez, gera rajadas maiores no estado de atividade e, conseqüentemente, tem seu desempenho mais degradado.

[60] Caracterização do vocoder G.711, com frame size de 65ms e correspondentes pacotes com tamanhos de 520bytes.



	BE-256bytes	BE-512bytes	BE-1500bytes
1 Fluxo- BE UDP	7.750	9.083	9.667
Agregado- BE UDP	7.583	8.750	9.250
1 Fluxo- BE TCP	5.667	7.417	8.500
Agregado- BE TCP	5.333	6.500	7.750

Figura 7-6 – Impacto do tráfego de fundo (estudo da agregação de voz) - Valores médios de delay.

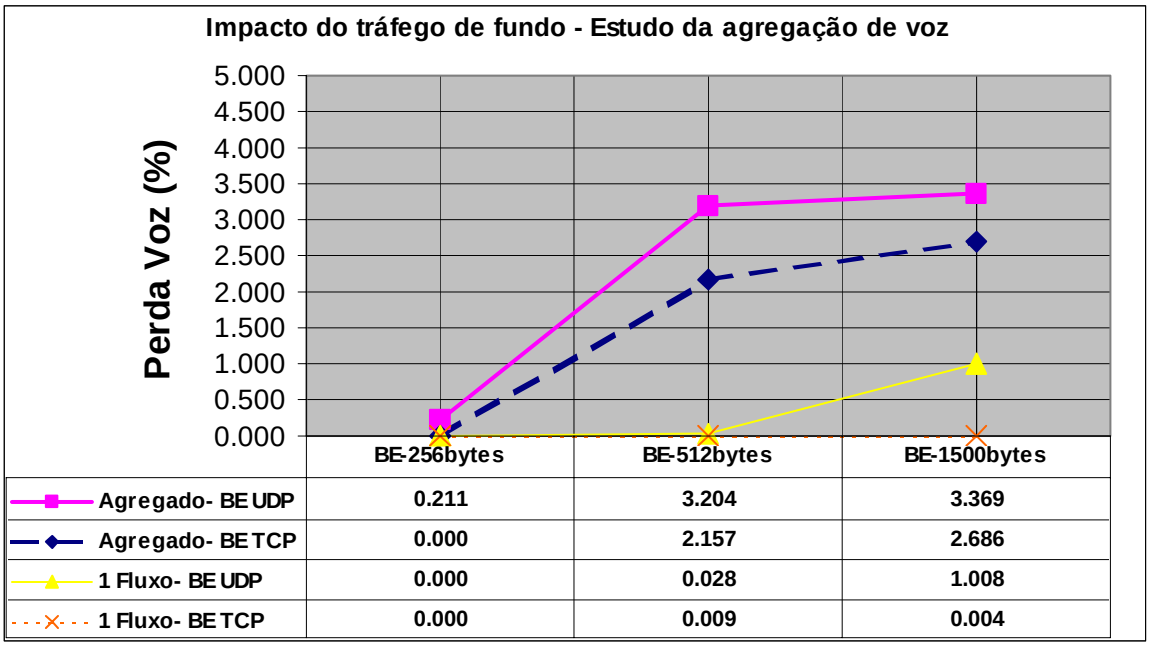


	BE-256bytes	BE-512bytes	BE-1500bytes
1 Fluxo- BE UDP	8.917	8.167	16.833
Agregado- BE UDP	6.000	7.000	17.250
1 Fluxo- BE TCP	5.750	7.000	15.000
Agregado- BE TCP	5.500	6.750	11.333

Figura 7-7 – Impacto do tráfego de fundo (estudo da agregação de voz) - Valores médios de jitter.

As taxas médias de descarte de pacotes (Figura 7-8), no entanto, não acompanharam as tendências das métricas anteriores, tendo o tráfego com apenas 1 fluxo apresentado os menores índices. Na verdade, não necessariamente as taxas de descarte devem ter a mesma inclinação das métricas de delay e jitter. Resultados de testes individuais mostraram, por exemplo, tráfego de voz (principalmente agregado) com taxa de perda consideravelmente alta (apresentando até uma certa distorção, em relação às demais

repetições) e com valores razoáveis de delay e jitter. Recíprocas também foram percebidas, com taxas de perda pequenas e valores de delay e jitter acima da média.



	BE-256bytes	BE-512bytes	BE-1500bytes
Agregado- BE UDP	0.211	3.204	3.369
Agregado- BE TCP	0.000	2.157	2.686
1 Fluxo- BE UDP	0.000	0.028	1.008
1 Fluxo- BE TCP	0.000	0.009	0.004

Figura 7-8 – Impacto do tráfego de fundo (estudo da agregação de voz) - Taxas médias de perda de pacotes.

7.2 Segunda etapa

O intuito desta fase é avaliar efeitos no desempenho dos fluxos prioritários de voz, causados pela injeção na rede de tráfego de melhor esforço de origem FTP, que é uma aplicação elástica (do tipo "*bulk traffic*") e de rajadas mais longas do que serviços como *web* (http), por exemplo.

Da mesma forma que na 1ª etapa, a disciplina de escalonamento utilizada aqui foi a ***prio***, tendo a banda 0 sido preenchida por uma fila do tipo BFIFO, com 1.600bytes, e à banda 1 sendo anexada, também, uma fila BFIFO, com 14.700bytes. A reserva de banda efetuada no roteador interno, através da configuração da disciplina ***cbq***, foi de 1.500Kbps.

Este estudo compreendeu a observação de agregados de voz compostos por 14 fluxos, modelados, novamente, com base no vocoder G.711, tendo o *frame size* de 60ms (520bytes de *payload*). Foram gerados dois tipos distintos de tráfego de background: **TCP**, gerado pela ferramenta NETPERF e que será aqui reconhecido como "tráfego não-FTP"; e **FTP**, gerado pelo programa ftp do Linux. Ambos têm pacotes com tamanhos totais de 1.448bytes (payload + cabeçalhos).

Comparou-se o desempenho dos agregados de voz frente à injeção de tráfego BE, marcado por: 01 fonte TCP, 01 sessão FTP e, finalmente, 04 sessões FTP. Além do tráfego prioritário, a atuação desses fluxos de fundo também foi analisada.

No esquema FTP, o cliente foi a máquina qos2 e o servidor a máquina qos1. Logo, o fluxo de fundo (FTP) percorreu a rede de qos1 a qos2, atravessando, com DiffServ ativado, a interface de saída do roteador interno (eth1 de qos5), tal como nos experimentos anteriores.

Houve a preocupação de se trabalhar com a mesma caracterização para tráfegos de fundo não-FTP e FTP (com 01 sessão). Ou seja, foram avaliadas, primeiramente, as características do tráfego de FTP e, só depois, é que se modelou o tráfego não-FTP, de forma a apresentarem as mesmas peculiaridades: tamanho dos pacotes, tempo de geração e vazão. Para 01 sessão FTP, utilizou-se arquivo único com 5.090Kbytes, transferido em torno de 28.8 segundos (com a presença de tráfego de voz[61]). Tomou-se então o mesmo tempo para se gerar o tráfego não-FTP, de maneira que correspondesse ao tempo de transferência FTP. Para 04 sessões FTP, fragmentou-se o arquivo empregado inicialmente (de 5.090Kbytes) em 04 arquivos menores, de forma que a soma de seus tamanhos fosse exatamente igual ao arquivo utilizado com 01 sessão[62].

Os cenários foram comparados, tomando-se como base o desempenho dos fluxos prioritários e de melhor esforço. A Figura 7-9 e a Figura 7-10 mostram o desempenho do tráfego de voz frente à perturbação do meio através de fluxo TCP e da transferência de arquivos de dados. Vê-se, nas figuras, como a aplicação FTP agride mais os fluxos de voz do que uma geração puramente TCP, tendo ambos, como já colocado antes, as mesmas características de transmissão, em termos de tamanho dos pacotes e tempo de geração. Pode-se imaginar os resultados em função das naturezas elástica do tráfego FTP e oscilatória do protocolo TCP, com controle variado de congestionamentos. Os resultados levam a crer que a geração NETPERF (não-FTP) é, mesmo que adaptativa às condições da rede, mais previsível e menos variável que o serviço FTP. Isto faz com que, com FTP, o tráfego concorrente (no caso aqui o de voz) ocupe mais tempo nas filas e

[61] Tendo-se somente o tráfego FTP na rede, a transmissão do arquivo durou 24 segundos.
[62] Um com 1.250Kbytes e os outros três com 1.280Kbytes cada.

apresente maior variação entre os períodos de enfileiramento, sendo isso mais evidente ainda quando várias sessões de longas transferências ocupam a rede.

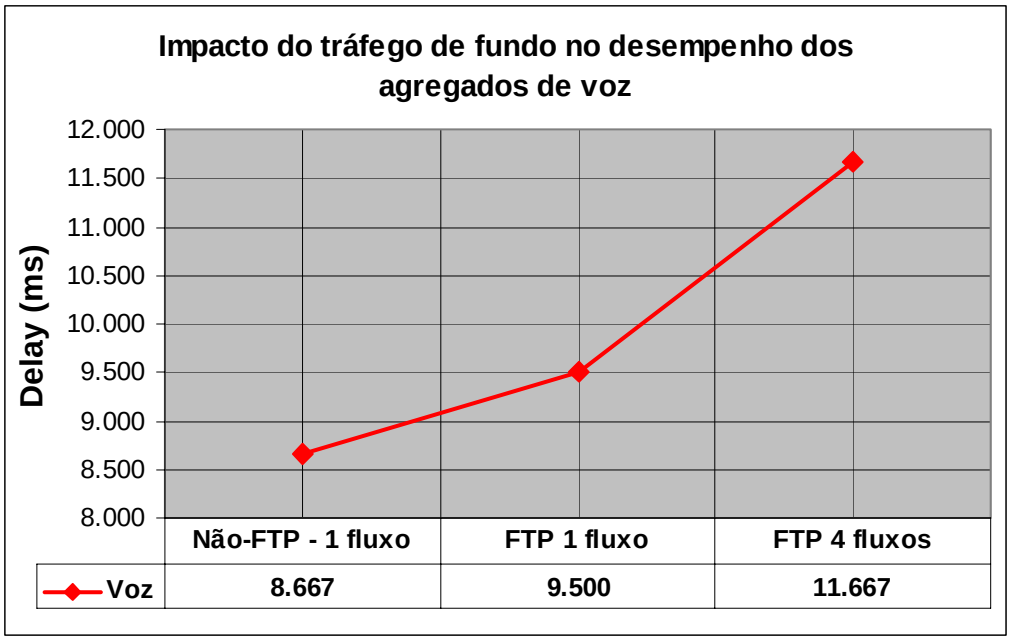


Figura 7-9 – Delay médio do tráfego de voz.

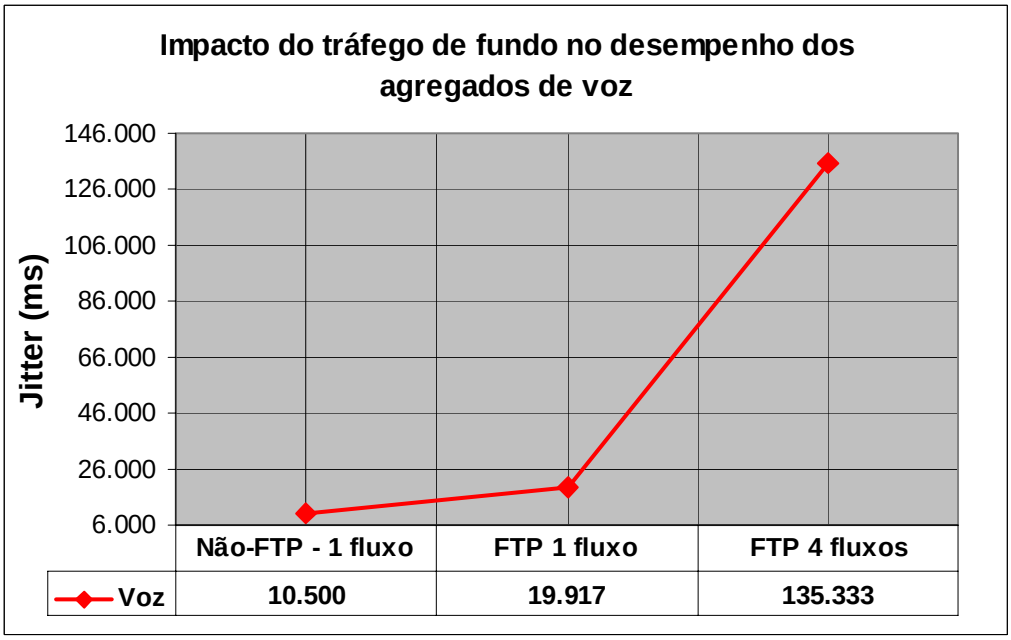
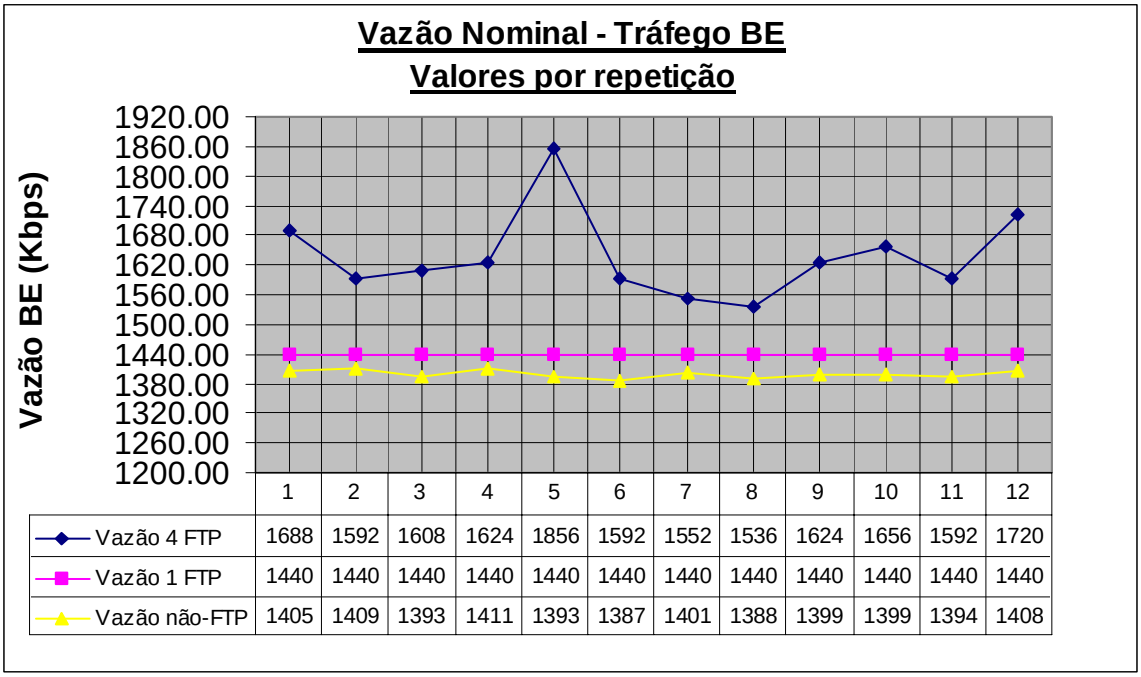


Figura 7-10 – Jitter médio do tráfego de voz.

A vazão média do tráfego de fundo FTP ficou acima da alcançada pelo fluxo não-FTP. Como este teve, em todas as repetições, valores de vazão inferiores aos de tráfego FTP (Figura 7-11), então isto contribui para justificar o melhor desempenho do agregado de voz com o tráfego puramente TCP, já que, com tráfego não-FTP, houve menor ocupação da banda pelo tráfego BE.

Com 4 sessões FTP, durante as repetições dos testes, chamou a atenção, ao contrário dos demais experimentos desta sessão, a grande variação dos valores de jitter atribuídos ao tráfego prioritário, ora com valores bem expressivos (acima de 280ms) e ora com valores abaixo do 24ms[63]. Essa instabilidade motiva o alcance dos piores resultados de delay e jitter, podendo ser observada na Figura 7-11. Ao contrário, com 1 sessão FTP, os valores de vazão foram invariáveis, tendo, em todas as repetições, o valor de 1.440Kbps.



	1	2	3	4	5	6	7	8	9	10	11	12
Vazão 4 FTP	1688	1592	1608	1624	1856	1592	1552	1536	1624	1656	1592	1720
Vazão 1 FTP	1440	1440	1440	1440	1440	1440	1440	1440	1440	1440	1440	1440
Vazão não-FTP	1405	1409	1393	1411	1393	1387	1401	1388	1399	1399	1394	1408

Figura 7-11 – Seqüências da vazão de melhor esforço.

Como já comentado, os cenários em que se trabalha com tráfego de fundo do tipo FTP são, para efeito de comparação, semelhantes, com os arquivos transferidos tendo exatamente os mesmos tamanhos, sendo que, com 4 sessões FTP, fragmentou-se o arquivo utilizado com 1 sessão em 4 arquivos de tamanhos menores. No entanto, o que se percebeu, ainda na Figura 7-11, foi que a vazão total dos 4 fluxos FTP ficou consideravelmente acima da praticada por apenas 1 fluxo FTP[64]. Isto abre um leque para investigações futuras, uma vez que, dado o fenômeno da ***sincronização global***, deveria acontecer justamente o contrário, pois, teoricamente, com várias sessões TCP/FTP, o controle ativo do protocolo TCP tem maior atuação e as vazões de cada sessão tendem a diminuir, ainda mais se for considerado que esse tráfego concorre com mais 14 fluxos prioritários de voz.

[63] O jitter médio com 4 fluxos FTP foi calculado em 135,3ms.
[64] A vazão média com 4 fluxos FTP ficou em 1.636,67Kbps e com 1 fluxo FTP em 1.440,00Kbps.

Os gráficos de taxas de “perda” e de “vazão alcançada”, referentes ao comportamento dos fluxos de voz, são mostrados na Figura 7-12 e na Figura 7-13, respectivamente. Eles praticamente confirmam os resultados das métricas de delay e jitter, com os níveis maiores de deterioração da voz ocorrendo com tráfego FTP. No entanto, deve-se observar que, com várias sessões de transferência, há, em relação ao tráfego com 1 sessão FTP, uma melhora muito discreta nas taxas de perda e de “vazão alcançada” do tráfego de voz. Isto demonstra uma estagnação dos resultados, provavelmente ocasionada pela atuação da disciplina ***prio*** em evitar a tendência de uma maior taxa de perda de pacotes prioritários de voz.

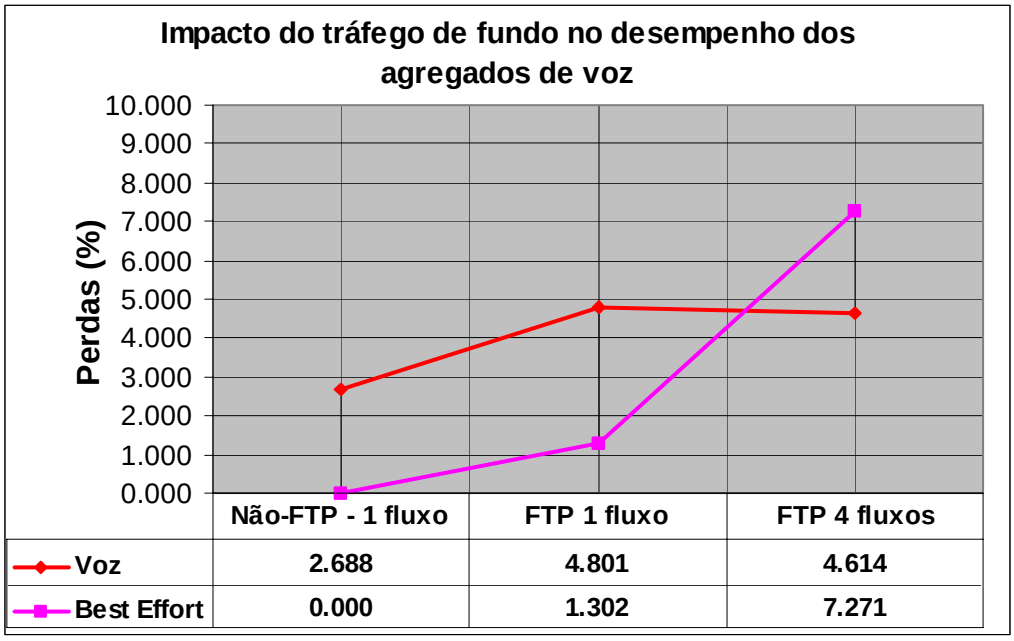


Figura 7-12 – Taxa média de perda de pacotes dos tráfegos de voz e de BE.

Quanto ao tráfego de fundo, observa-se que o mesmo se degrada com o aumento do número de sessões de transferência, mas, conforme já sinalizado na sessão anterior, não mantém, como o tráfego UDP, coerência na relação “perda x vazão”. O fluxo não-FTP, por exemplo, não sofreu descarte de pacotes e, no entanto, em relação aos demais tráfegos, foi o que atingiu o menor índice na métrica de “vazão alcançada” (80.42%). O controle de fluxo e a elasticidade do comportamento dos tráfegos TCP e TCP/FTP são os responsáveis pelo contra-senso dos valores médios encontrados nos fluxos de fundo.

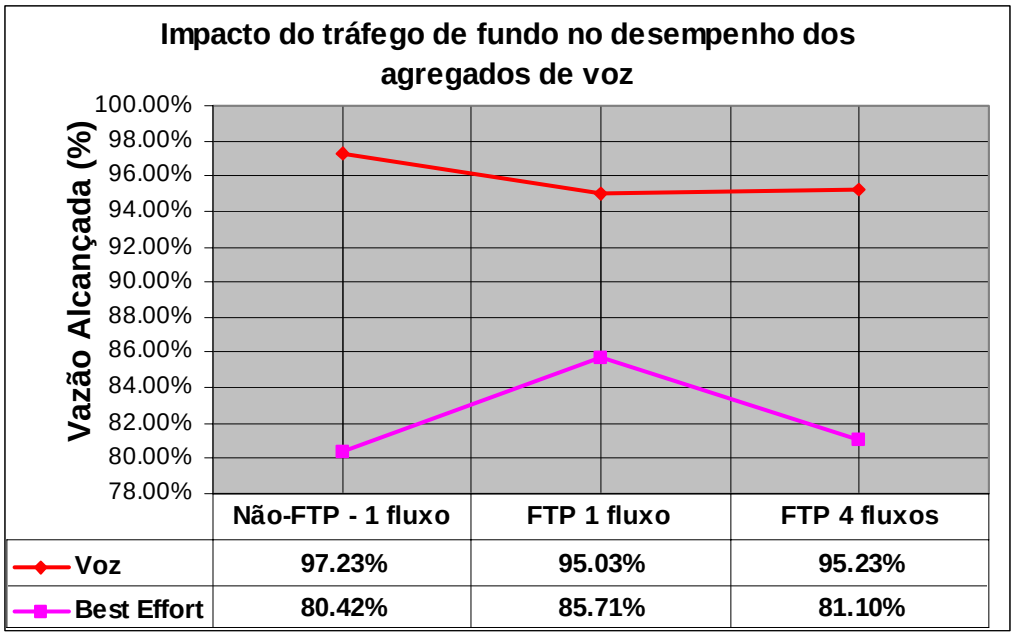


Figura 7-13 – Taxa média de "vazão alcançada" dos tráfegos de voz e de BE.

7.3 Conclusão do capítulo

Variaram-se, neste capítulo, as características do tráfego de fundo e observou-se o impacto dessas mudanças no tráfego modelado de voz. Os estudos foram realizados em duas etapas.

Na 1ª etapa, foram alterados tanto o tamanho dos pacotes (256, 512 e 1500bytes) quanto o tipo de protocolo de transporte (TCP e UDP) do tráfego de melhor esforço. Os teste mostraram que, em todas as métricas de QoS, o tráfego de fundo do tipo UDP agrediu mais o agregado de voz do que o protocolo TCP, sendo que, quanto maior o tamanho dos pacotes BE, maior foi a degradação do tráfego prioritário. Estudou-se, ainda nesta etapa, o fator agregação, onde, nas mesmas condições de rede do estudo anterior, o tráfego de voz foi gerado sem agregação (com apenas 1 fluxo). Percebeu-se que os testes ratificaram as tendências apresentadas no Capítulo 6, tendo os agregados obtido melhor desempenho (em delay e em jitter) do que tráfego sem agregação, independentemente do tipo de protocolo de transporte utilizado.

A 2ª etapa avaliou a influência de tráfego elástico de fundo do tipo FTP, com a variação da quantidade de fontes. Viu-se que, com várias sessões FTP, houve mais degradação do tráfego de voz do que com apenas 1 fluxo FTP. Tráfego BE não-FTP (TCP apenas) também foi testado e foi o que menos influenciou no desempenho do tráfego

prioritário. Em virtude do fenômeno da *sincronização global*, alertou-se, com múltiplos fluxos FTP, para a necessidade de futuras investigações, uma vez que a vazão total desses fluxos alcançou valores consideravelmente acima da praticada por apenas 1 fluxo FTP.

8. Conclusão

O trabalho contribuiu na construção de uma plataforma de testes (*testbed*), com estudos, em rede local composta por hardwares de baixo custo e softwares livres, de impactos no desempenho de fluxos modelados de voz, a partir da utilização do modelo de diferenciação de serviço em sistemas Linux.

Fez-se, inicialmente, uma contextualização teórica do assunto, abrangendo-se aspectos básicos da qualidade de serviço e dos modelos sugeridos pelo IETF, sendo que, para o serviço diferenciado, houve um certo aprofundamento. Foram apresentados, logo após tal fundamentação, os itens relacionados ao ambiente de testes, com destaque para a topologia da rede e para a metodologia adotada nos experimentos.

Verificando-se, através de experimentos com escalonadores diferentes, de que o meio era capaz de diferenciar tráfego, partiu-se para a modelagem de vocoders distintos de áudio, onde se percebeu que, utilizando-se o escalonador ***prio***, a modelagem do vocoder G.726, com pacotes de 120bytes, apresentou os menores atrasos e os menores índices de descarte de pacotes, mas, no entanto, apresentou as mais altas variações de atraso entre pacotes. O tamanho dos pacotes afetou diretamente no atraso e inversamente no atraso entre pacotes do tráfego modelado de voz. Um ambiente de sobrecarga de rede também foi aplicado e as mesmas tendências foram encontradas, confirmando assim os resultados inicialmente colhidos. Ainda no estudo dos vocoders, agora com disciplina de escalonamento do tipo FIFO (***bfifo***), que representou a ausência de qualidade de serviço na rede, sem priorização de tráfego, embora se tivesse a preocupação de manter-se as mesmas condições de transmissão de tráfego do experimento com ***prio***, os efeitos foram bem diferentes. A modelagem do vocoder G.711, com pacotes de 520 bytes, obteve os melhores resultados, em todas as métricas.

Utilizou-se o testbed para um estudo comparativo do desempenho de agregados de voz, gerados a partir de fontes distintas, com características contínuas (CBR) e de atividade-silêncio (ON-OFF). Observou-se que o tráfego CBR obteve os menores valores médios de atraso e de descarte de pacotes. No entanto, o tráfego ON-OFF conseguiu o melhor desempenho no jitter. Foram analisadas, ainda neste estudo, os efeitos da agregação

de fluxos do tipo atividade-silêncio, onde se variou o número de fluxos prioritários de voz. Viu-se que, com um maior número de fluxos, os melhores resultados foram atingidos, confirmando assim o fator agregação como o princípio básico do modelo DiffServ.

Foram avaliados também, num ambiente com priorização de tráfego (disciplina ***prio***), impactos no desempenho da modelagem de agregados de voz, causados pela variação do tráfego de fundo. Duas etapas foram realizadas:

- Na primeira etapa, o tráfego BE recebeu variações tanto no tipo de protocolo de transporte utilizado quanto no tamanho dos pacotes. Constatou-se que o tráfego UDP, pela falta de mecanismos de controle de congestionamento, causa maior influência no desempenho do agregado de voz que fluxos do tipo TCP, indiferentemente do tamanho do pacote BE. Em ambos os protocolos, quanto maior o pacote de fundo, menor o desempenho dos fluxos de voz. Ou seja, pacotes de fundo maiores, misturados aos de voz, fazem com que estes levem mais tempo para atravessar a rede, influindo em sua degradação. Realizou-se ainda nesta etapa um estudo de agregação, no qual se repetiram os testes, agora com apenas 1 fluxo de voz sendo perturbado por tráfegos de fundo UDP e TCP. Observou-se que, para os atrasos e jitters médios, os agregados conseguiram melhor desempenho que o tráfego de voz com apenas 1 fluxo, ratificando os resultados obtidos no estudo da Sessão 6.2.(“Agregação ON-OFF de fluxos prioritários”).

- Na segunda etapa, avaliaram-se os efeitos de aplicações FTP, com uma e quatro sessões, nos fluxos de voz. A maior degradação do tráfego prioritário foi observada com a injeção de 4 sessões FTP, o que possibilitará uma investigação futura, uma vez que, dada a *sincronização global*, quanto maior o número de fluxos TCP, como há uma maior controle de transmissão das fontes geradoras, menor deveria ser a vazão total na rede. Valores métricos elevados e uma grande instabilidade no tráfego com 4 sessões FTP foram observados.

O trabalho definiu a métrica “**taxa de vazão alcançada**”. Trata-se da razão entre a vazão atingida pelos fluxos e a respectiva taxa de geração aplicada. Os resultados indicaram, em todos os experimentos com tráfego UDP, uma complementação entre os

valores das taxas de perda de pacotes e de “vazão alcançada”, uma vez que, para qualquer tráfego não cooperante e numa mesma proporção, quanto maior é o número de pacotes descartados, menor é a quantidade de bits recebidos no destino (e vice-versa). Tráfego TCP, ao contrário, não apresentou coerência na relação entre as taxas de perda e de “vazão alcançada”, dada a própria natureza do protocolo, que é sempre influenciado por mecanismos de controle de transmissão.

Houve uma sensível preocupação no trabalho pela exposição das justificativas de cada resultado, com a tradução dos gráficos que expressam as métricas de interesse. Além disso, imaginou-se que, para explicar o comportamento do tráfego modelado de voz e como ele é influenciado pela rede, seria necessário, também, justificar as tendências dos fluxos de perturbação do meio. Isto ocorreu em várias experimentações.

Pode-se levantar aqui algumas sugestões de trabalhos futuros, como:

- A topologia construída, numa rede local e controlada, não permite concluir os resultados, segundo a classificação do ITU-T (ver Figura 3-9) para atrasos em um sentido. Para uma visualização mais realística dessa métrica, seria necessário estender geograficamente a rede, mantê-la controlada, recalcular os atrasos médios e analisar a viabilidade sobre os resultados obtidos;

- O tráfego modelado de voz deste trabalho foi estudado de uma forma quantitativa. No entanto, fica a necessidade de analisá-lo qualitativamente, no mesmo testbed ou numa topologia mais estendida (o que seria mais interessante ainda), com a aplicação de ferramentas específicas para isso, tais como *Netmeeting*, *Real Player* ou outras;

- Pode-se utilizar a mesma plataforma de teste para uma avaliação, através do serviço diferenciado, de tráfego modelado de vídeo no formato MPEG, com a utilização de mecanismos de gerenciamento ativo de filas, responsáveis pela associação de níveis de precedências de descarte de pacotes, de acordo com o tipo de quadro trafegado, tal como simulado em [55].

- O efeito da *sincronização global* consiste na atuação simultânea de mecanismos de controle de congestionamentos de protocolos, como o TCP. Tal fenômeno

pode criar uma situação em que a rede oscila freqüentemente entre taxas elevadas de transferência, com perdas de pacotes, e vazões muito aquém da capacidade da rede, podendo prejudicar o desempenho dos fluxos. Mecanismos de gerenciamento ativo de filas, como os providos pela família de algoritmos RED, em conjunto com o protocolo TCP, impedem, através da monitoração do tamanho das filas de espera e da notificação de ocorrências de perdas às fontes geradoras, a *sincronização global* e evitam o congestionamento da rede. Desta maneira, necessita-se estudar com mais profundidade, num ambiente com diferenciação de serviço, a influência da *sincronização global* e dos mecanismos ativos de filas no desempenho de tráfego multimídia (voz e vídeo).

9. Referências Bibliográficas

[1] CROLL, A. e PACKMAN, E. **Managing Bandwidth: deploying QoS in Enterprise Networks**. Prentice Hall, 1999.

[2] MARTINS, JOBERTO. **Qualidade de Serviço (QoS) em redes IP: Princípios Básicos, Parâmetros e Mecanismos**. UNIFACS, São Paulo, 2000.

[3] KAMIENSKI, CARLOS ALBERTO e SADOK, DJAMEL. **Qualidade de Serviço na Internet**. Minicurso SBRC 2000, maio 2000.

[4] XIAO, XIPENG. **Providing Quality of Service in the Internet**. Department of Computer Science, Michigan State University, 2000.

[5] LEFELHOCZ, C. *et al*. **Congestion Control for Best-Effort Service: Why We Need A New Paradigm**. IEEE network, janeiro 1996.

[6] BRADEN, R. *et al*. **Recommendations on Queue Management and Congestion Avoidance in the Internet**. RFC 2309, abril 1998.

[7] SOARES, L. *et al*. **Redes de Computadores: das LANs, MANs e WANs às Redes ATM**. Editora Campus, 1997.

[8] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO). **Information Technology – Quality of Service – Framework**. ISO/IEC 13236, julho 1995.

[9] TEITELMAN, B. e HANSS, T. **QoS Requirements for Internet2**. Internet2 QoS Work Group Draft, abril 1998.

[10] STARDUST.COM, INC. **The Need for QoS**. White Paper, julho 1999.

[11] FERGUSON, P. e HUSTON, G. **Quality of Service: delivering QoS on the Internet and in Corporate Networks**. Wiley Computer Publishing, 1999.

[12] R. BRADEN; D. CLARK e S. SHENKER. **Integrated Services in the Internet Architecture: an Overview**. RFC 1633, junho 1994.

[13] MIRAS, DIMITRIOS. **Network QoS Needs of Advanced Internet Applications: A Survey**. Internet2 QoS Work Group, dezembro 2002.

[14] ARMITAGE, GRENVILLE. **Quality of Service in IP Networks: Foundations for a Multi-Service Internet**. USA: Macmillan Technical Publishing, abril 2000.

[15] SHENKER, S.; PARTRIDGE, C. e GUERIN, R. **Specification of guaranteed quality of service**. RFC 2212, 1997.

[16] WROCLAWSKI, J. **Specification of the controlled-load network element service**. RFC 2211, 1997.

[17] OLIVEIRA, RENATO DONIZETE. **Serviços Diferenciados em redes IP – medições e testes para aplicações envolvendo mídias contínuas**. Abril 2001. Dissertação de Mestrado – Universidade Federal de Santa Catarina (UFSC). Santa Catarina.

[18] MOTA, OSCAR. **Uma arquitetura adaptável para provisão de QoS na Internet**. Maio 2001. Dissertação de Mestrado – Pontifica Universidade Católica do Rio de Janeiro. Rio de Janeiro.

[19] ARINDAM, PAUL. **QoS in data networks: Protocols and Standards**. 1999.

[20] KOREN, T. *et al*. **Enhanced Compressed RTP (CRTP) for Links with High Delay, Packet Loss and Reordering**. RFC 3545, 2003.

[21] POSTEL, J. **Internet Protocol – DARPA Internet Program Protocol Specification**. RFC 791, 1981.

[22] ALMQUIST, P. **Type of Service in the Internet Protocol Suite**. RFC 1349, 1992.

[23] NICHOLS, K.; BLAKE, S.; BAKER, F. e BLACK, D. L. **Definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers**. RFC 2474. 1998.

[24] ALMESBERGER, WERNER *et al*. **A Prototype Implementation for the IntServ Operation over DiffServ Networks**. In *Proceeding of the IEEE Globecom 2000*, S. Francisco, USA, novembro 2000

[25] MAMELI, ROBERTO e SALSANO, STEFANO. **Use of COPS for IntServ operations over Difffserv: Architectural Issues, Protocol design and Test-Bed implementations**. *In Proceeding of the IEEE ICC 2001*, Helsinki, junho 2001.

[26] DURHAM, D.; BOYLE, J.; COHEN, R. *et al*. **The COPS (Common Open Policy Service) Protocol**. RFC 2748, 2000.

[27] RAJAN, RAJU et al. **A Policy Framework for Integrated and Differentiated Services in the Internet**. IEEE Network, pp.36-41, setembro-outubro 1999.

[28] ALMES, G.; KALIDINDI, S.; ZEKAUSKAS, M. **One-way Delay Metric for IPPM**. RFC 2679, setembro 1999.

[29] DEMICHELIS C; CHIMENTO P. **IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)**. RFC 3393, 2002.

[30] HERSENT, OLIVER. **Telefonia IP**. São Paulo: Prentice Hall, 2002.

[31] WAGNER, KURT. **Short Evaluating of Linux's Token-Bucket-Filter (TBF) Queuing Discipline**. 2001.

[32] BLAKE, R.; BLACK, D. *et al*. **An Architecture for Differentiated Services**. RFC 2475, dezembro 1998.

[33] ZIVIANI, ARTUR; REZENDE, JOSÉ F. DE; DUARTE, OTTO CARLOS. **Evaluating Voice Traffic in a differentiated services network**. In *Proceeding of the 17th International Teletraffic Congress* – ITC17, pp. 907-918, Salvador, Brazil, December 2001.

[34] B. DAVIE *et al*. **An expedited forwarding PHB (Per-Hop Behavior)**. RFC 3246, março 2002.

[35] J. HEINANEN *et al*. **Assured forwarding PHB group.** RFC 2597, junho 1999.

[36] MOTA, OSCAR THYAGO JOSÉ DUARTE DANTAS LISBÔA. **Uma arquitetura adaptável para provisão de QOS na Internet**. 2001. Dissertação de Mestrado – Pontifica Universidade Católica do Rio de Janeiro. Rio de Janeiro.

[37] QBONE home page. Disponível em http://qbone.internet2.edu/.

[38] ALMESBERGER, WERNER; SALIM, JAMAL HADI.; KUZNETSOV, ALEXEY. **Differentiated Services on Linux**. In *Proceeding of the Globecom'99*, pp. 831-836, dezembro 1999.

[39] RADHAKRISHNAN, S. **Linux – Advanced Networking Overview, Version 1**. The University of Kansas,1999.

[40] SEMERIA, CHUCK. **Supporting Differentiated Service Classes: Queue Scheduling Disciplines**. USA: Juniper Networks, dezembro 2001.

[41] HUBERT, BERT. **Linux Advanced Routing & Traffic Control HOWTO**. 2002.

[42] PRIOR, RUI PEDRO DE MAGALHÃES CLARO. **Qualidade de Serviço em Redes de Comutação de Pacotes**. 2001. Dissertação de Mestrado – Faculdade de Engenharia da Universidade do Porto. Porto, Portugal.

[43] GAO RESEARCH INC. **G.726 Vocoder**. Disponível em http://www.gaoresearch.com.

[44] MARKOPOULOU, ATHINA P.; TOBAGI, FOUAD A.; KARAM, MANSOUR J. **Assessment of VoIP Quality over Internet Backbones**. In *Proceeding of the IEEE INFOCOM 2002*, New York, junho 2002.

[45] ZIVIANI, ARTUR. **Voz sobre serviços diferenciados na Internet**. Setembro 1999. Dissertação de Mestrado – Universidade Federal do Rio de Janeiro (UFRJ). Rio de Janeiro.

[46] SU, D.; SRIVASTAVA, J.; YAO, J. **Investigating factors influencing QoS of Internet phone**. In *Proceeding of the IEEE International Conference on Multimedia Computing and System*, pp. 308-313, junho 1999.

[47] ITU-T. **One-way transmission time**. Recomendação G.114. Março 1993.

[48] SCHULZRINNE, H. *et al*. **RTP: A Transport Protocol for Real-Time Applications**. RFC 1889, 1996.

[49] FERRARI, TIZIANA; PAU GIOVANNI; RAFAELLI, CARLA. **Measurement Based Analysis of Delay in Priority Queuing**. In *Proceeding of the IEEE Global Telecommunication Conference Globecom 2001*, S. Antonio, Texas, novembro 2001.

[50] LEOCÁDIO, MÁRCIO PARENTE; RODRIGUES, PAULO AGUIAR. **Uma ferramenta para Geração de Tráfego e Medição em Ambiente de Alto Desempenho**. *Anais do 18º SBRC*, Belo Horizonte, MG, pp. 321-336, maio 2000.

[51] NAVAL RESEARCH LABORATORY (NRL). **The Multi-Generator (MGEN) Toolset**. Disponível em http://manimac.itd.nrl.navy.mil/MGEN.

[52] JONES, R. **Netperf**. 1995. Disponível em http://www.netperf.org/ netperf/ NetperfPage.html.

[53] V. JACOBSON; C. LERES; S. MCCANNE. **Tcpdump, a network packet capture program**. 1989. Disponível em http://www.tcpdump.org.

[54] FERRARI, TIZIANA. **End-to-End Performance Analysis with Traffic Aggregation**. Computer Network Journal, vol. 34, no. 6, pp. 905-914, dezembro 2000.

[55] ZIVIANI, ARTUR; REZENDE, JOSÉ FERREIRA DE; DUARTE, OTTO CARLOS. **Vídeo MPEG sobre serviços diferenciados**. *Anais do 19º Simpósio Brasileiro de Telecomunicações 2001*, Fortaleza, CE, 2001.

[56] ZIVIANI, ARTUR; REZENDE, JOSÉ FERREIRA DE; DUARTE, OTTO CARLOS. **Evaluating the expedited forwarding of voice traffic in a differentiated services network**. International Journal of Communication Systems, vol. 15, no. 9, pp. 799-813, novembro 2002.

[57] ZIVIANI, ARTUR; REZENDE, JOSÉ FERREIRA DE; DUARTE, OTTO CARLOS. **Towards a differentiated services support for voice traffic**. 1999. *Anais do IEEE Global Telecommunications 1999*, Rio de Janeiro, RJ, 1999.

[58] TYAGI, ANURAG; MUPPALA JOGESH K.; MEER, HERMANN DE. **VoIP support on differentiated services using Expedited Forwarding**. In *Proceeding of the IPCCC 2000*, Phoenix, AZ, USA, pp. 574-580, fevereiro 2000.

[59] KOS, ANTON. **Real-Time Applications Performance in Differentiated Service Network**. In *Proceeding of the IEEE Region 10 International Conference on Electrical and Electronic Technology*, vol. 1, pp. 96-102, agosto 2001.

[60] KOS, ANTON. **Performance of VoIP Applications in a Simple Differentiated Services Network Architecture**. In *Proceeding of the International Conference on Trends in Communications*, 2001.

[61] **NETWORK SIMULATION (NS)**. Disponível em http://www-mash.cs.berkeley.edu/ns/ ns.html.

[62] CASNER, S.; JACOBSON V. **Compressing IP/UDP/RTP Headers for Low-Speed Serial Links**. RFC 2508, 1999.

[63] Skype home page. Disponível em http://www.skype.com/.