

Pedro Henrique Ramos Ribeiro

*Implantação de uma rede de testes para a
plataforma de QoS EuQoS sobre DiffServ
em Linux*

Trabalho de conclusão de curso apresentado
como parte dos requisitos para obtenção do
grau Bacharel em Ciências da Computação.

Orientador:

Prof. Dr. Roberto Willrich

UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIAS DA COMPUTAÇÃO

Florianópolis – SC

Dezembro / 2008

Trabalho de conclusão de curso sob o título “Implantação de uma rede de testes para a plataforma de QoS EuQoS sobre DiffServ em Linux”, defendido por Pedro Henrique Ramos Ribeiro, em Florianópolis, Santa Catarina, pela banca examinadora constituída:

Prof. Dr. Roberto Willrich
Departamento de Informática e Estatística Orientador

Banca Examinadora

Prof. Dr. Vitorio Bruno Mazzola
Departamento de Informática e Estatística

Prof. Dr. Rosvelter Coelho da Costa
Departamento de Informática e Estatística

“Um homem que fosse só homem, e dissesse as coisas que Jesus disse, não seria um grande mestre da moral: seria ou um lunático, em pé de igualdade com quem diz ser um ovo cozido, ou então seria o demônio. Cada um de nós tem de optar por uma das alternativas possíveis. Ou este homem era, e é, Filho de Deus, ou então foi um louco, ou algo pior. Podemos contraargumentá-lo, talhando-o de louco, ou cuspir nele e matá-lo como um demônio; ou podemos cair aos seus pés e chamá-lo de Senhor e Deus. Mas não venhamos com nenhuma bobagem paternalista sobre ele ser um grande ser humano. Ele não nos deu esta escolha. Nem nunca pretendeu. (...) O Filho de Deus tornou-se homem para possibilitar que os homens se tornem filhos de Deus.” C.S. Lewis

Agradecimentos

A Jesus Cristo, o Senhor. “Porque dele, por Ele e para Ele são todas as coisas”. (Rm. 11:36)

Ao meu pai, João Ribeiro Junior, e minha mãe, Vandarlui Serafim Ramos Ribeiro, que me amam com verdadeiro amor de pais.

Ao Prof. Dr. Roberto Willrich pela confiança e trabalho investidos.

Resumo

O Sistema EuQoS foi desenvolvido por um conjunto de empresas e instituições de pesquisa na Europa com o objetivo de desenvolver uma arquitetura que agregue aos provedores de serviço de internet a tecnologia necessária para prover garantia de QoS fim-a-fim entre dois usuários na rede.

Este trabalho tem como proposta estudar este sistema e implementar uma rede de testes onde ela possa ser executada e estudada. Como o sistema EuQoS é independente de tecnologia de rede este trabalho implementa também um domínio DiffServ sobre Linux para completar a estrutura de testes. Problemas de medição, como a variação de frequência de relógio, precisaram ser tratadas.

Palavras-chave: Qualidade de Serviço, EuQoS, Medição em redes

Abstract

The EuQoS system was developed by a European group of enterprises and research groups with the goal of developing a architecture that would aggregate to the Internet Service Provider the technology needed for them to offer their clients end-to-end quality of service guarantees in the network.

This work proposes to study this system and deploy a test network where it can be executed and studied. As the EuQoS system is network technology independent this work also deploys a DiffServ domain over Linux to complete this test network. Measurement problems, as clock drift, were also treated here.

Key-words: Quality of Service, EuQoS, Network Measurement

Sumário

Lista de Figuras

Lista de Tabelas

Lista de acrônimos e abreviações	p. 13
1 Introdução	p. 17
1.1 Objetivo	p. 17
1.1.1 Objetivos específicos	p. 18
2 Qualidade de Serviço - QoS	p. 19
2.1 Aplicações de tempo-real e a necessidade de QoS	p. 20
2.2 Classes de aplicações dependentes de QoS	p. 20
2.3 Parâmetro de QoS	p. 22
2.4 Controle de Admissão	p. 23
2.5 Blocos fundamentais para provisão de QoS em nível de rede	p. 24
2.5.1 Escalonamento	p. 24
2.6 Acordos de Nível de Serviço	p. 26
2.6.1 Gerenciamento de buffers	p. 27
2.6.2 Policiamento	p. 28
2.7 Arquiteturas de serviços integrados - IntServ	p. 29
2.7.1 O modelo IntServ	p. 29
2.7.2 Classes de serviço	p. 30

2.7.3	O protocolo RSVP	p. 31
2.7.4	Avaliação do modelo IntServ	p. 31
2.8	Arquiteturas de Serviços Diferenciados - DiffServ	p. 32
2.8.1	O modelo DiffServ	p. 33
2.8.2	O campo DS	p. 33
2.8.3	PHB CS - Case Selector	p. 34
2.8.4	EF PHB - Expedited Forwarding	p. 34
2.8.5	AF PHB - Assured Forwarding	p. 34
2.8.6	PDB - Per-Domain Behavior	p. 35
2.8.7	O funcionamento de uma arquitetura DiffServ	p. 36
2.8.8	Bandwidth Broker	p. 37
2.8.9	Avaliação do modelo DiffServ	p. 38
3	O Sistema EuQoS	p. 39
3.1	O cenário típico	p. 40
3.2	Visão geral da solução proposta pelo Sistema EuQoS	p. 41
3.3	A Arquitetura do Sistema EuQoS	p. 43
3.3.1	Camada de Aplicação	p. 43
3.3.2	Camada de Rede Virtual	p. 43
3.3.3	Camada de Transferência	p. 44
3.3.4	Principais Módulos	p. 44
3.3.4.1	Camada de Aplicação	p. 44
3.3.5	Camada de rede virtual - Independente de Tecnologia	p. 46
3.3.6	O protocolo EQ-BGP	p. 49
4	Implantação e testes do Sistema EuQoS	p. 51
4.1	Primeira implantação e Testes	p. 51

4.1.1	Primeiro teste UDP	p. 51
4.1.2	Segundo teste UDP	p. 53
4.1.3	VoIP	p. 55
4.1.4	Análise dos resultados	p. 56
4.2	Segunda Implantação e teste	p. 58
4.2.1	O problema da sincronização de relógios	p. 59
4.3	Medição do tempo de sinalização do Sistema EuQoS	p. 61
4.4	O domínio DiffServ	p. 62
4.5	Teste do domínio DiffServ	p. 63
4.5.1	Cenário 1	p. 65
4.5.2	Cenário 2	p. 68
4.5.3	Comentários sobre os testes	p. 68
4.6	Conclusão	p. 71
	Referências Bibliográficas	p. 72
5	Anexo A - Artigo	p. 75

Lista de Figuras

2.1	Estrutura de compartilhamento hierárquico de link.	p. 26
2.2	Suavização de tráfego com <i>Token Bucket Filter</i>	p. 28
2.3	Um exemplo de implementação dos blocos básicos da arquitetura DiffServ. . .	p. 35
3.1	Senário típico de utilização do sistema EuQoS	p. 40
3.2	Como os parâmetros de QoS fim-a-fim são formados a partir dos parâ- metros oferecidos por cada domínio	p. 42
3.3	Diferentes camadas da arquitetura EuQoS	p. 43
3.4	Diagrama dos módulos da Camada de Aplicação da arquitetura EuQoS . .	p. 45
3.5	Principais módulos da arquitetura EuQoS	p. 47
4.1	Arquitetura da primeira implantação do sistema EuQoS	p. 52
4.2	Atraso dos pacotes sentido UFSC para LAAS	p. 53
4.3	Atraso dos pacotes sentido LAAS para UFSC	p. 54
4.4	Diagrama da segunda implantação do sistema.	p. 58
4.5	Diagrama da rede montada para avaliar a diferença entre os relógios. . .	p. 60
4.6	Plotagem das diferenças entre estampas de tempo de cada pacote.	p. 60
4.7	Domínio DiffServ realizado no Linux.	p. 63
4.8	Primeiro cenário de testes.	p. 64
4.9	Segundo cenário de testes.	p. 65
4.10	Cenário 1: Variação do Atraso e Jitter sem de tráfego de fundo BE. . . .	p. 66
4.10	Cenário 1: Variação do Atraso e Jitter para 1Mbit de tráfego de fundo BE.	p. 66

4.10 Cenário 1: Variação do Atraso e Jitter para 2Mbit de tráfego de fundo BE.	p. 66
4.10 Cenário 1: Variação do Atraso e Jitter para 3Mbit de tráfego de fundo BE.	p. 67
4.10 Cenário 1: Variação do Atraso e Jitter para 4Mbit de tráfego de fundo BE.	p. 67
4.11 Cenário 2: Variação do Atraso e Jitter sem tráfego de fundo BE.	p. 68
4.11 Cenário 2: Variação do Atraso e Jitter para 1Mbit de tráfego de fundo BE.	p. 69
4.11 Cenário 2: Variação do Atraso e Jitter para 2Mbit de tráfego de fundo BE.	p. 69
4.11 Cenário 2: Variação do Atraso e Jitter para 3Mbit de tráfego de fundo BE.	p. 69
4.11 Cenário 2: Variação do Atraso e Jitter para 4Mbit de tráfego de fundo BE.	p. 70

Lista de Tabelas

4.1	Primeiro teste UDP - Dados de Atraso.	p. 53
4.2	Primeiro teste UDP - Dados de Variação no Atraso (<i>Jitter</i>).	p. 54
4.3	Primeiro teste UDP - Vazão.	p. 54
4.4	Segundo teste UDP - Atraso em um sentido.	p. 55
4.5	Segundo teste UDP - Variação no atraso (<i>Jitter</i>).	p. 55
4.6	Segundo teste UDP - Vazão.	p. 56
4.7	Teste pacotes VoIP - Atraso em um sentido.	p. 56
4.8	Teste pacotes VoIP - Variação no atraso (<i>Jitter</i>).	p. 57
4.9	Teste pacotes VoIP - Vazão.	p. 57
4.10	Dados das diferenças entre estampas de tempo de cada pacote.	p. 61
4.11	Dados dos testes de tempo para estabelecimento de sessão no Sistema EuQoS	p. 61
4.12	Resultados dos testes no Cenário 1	p. 65
4.13	Resultados dos testes no Cenário 2	p. 68

Lista de acrônimos e abreviações

A

AF - Assured Forwarding

AS - Autonomous System

ASIG - Application Signaling

ATM - Asynchronous Transfer Mode

B

BA - Behavior Agregate

BB - Bandwidth Brokers

BE - Best Effort

BGP - Border Gateway Protocol

C

CAC - Controle de Admissão de Conexões

CBQ - Class-Based Queueing

CHAR - Charging

CS - PHB Case Selector

CL - Controled Load

CoS - Class of Service

DiffServ - Differentiated Services

DSCP -DS Codepoint

D

DiffServ - Differentiated Services

DSCP -DS Codepoint

E

EF - PHB Expedited ForwardingDiffServ

G

GRE - Generic Routing Encapsulation

H

HTB - Hierarchical Token Bucket

I

IETF - Internet Engineering Task Force

IntServ - Integrated Services

ISP - Internet Service Provider

L

LAAS - Laboratoire d'Architecture et d'Analyse des Systèmes

M

MMFM - Monitoring, Measurement and Fault Management

MPLS - Multiprotocol Label Switching

N

NSIS - Next Steps in Signaling

NTP - Network Time Protocol

P

PDB - Per-Domain Behavior

PDP - Policy Decision Point

PEP - Policy Enforcement Point

PHB - Per-Hop Behavior

Q

QCM - Quality Control Module

R

RA - Resource Allocator

RFC - Request for Comments

RM - Resource Manager

RSVP - Resource reservation protocol

RTP - Real Time Protocol

S

SAAA - Security, Authentication, Authorization and Accounting

SIP - Session Initiation Protocol

SLA - Service Level Agreement

SLS - Service Level Specification

SSN - Signaling and Service Negotiation

T

TBF - Token Bucket Filter

TERO - Traffic Engineering and Resource Optimization

TI - Technology Independent

ToS - Type of Service

U

UFSC - Universidade de Santa Catarina

UMTS - Universal Mobile Telecommunications System

V

VoD - Video on Demand

1 *Introdução*

Apesar do que já foi alcançado, para que haja um próximo grande salto em direção a convergência de aplicações para a Internet é imprescindível um avanço na garantia de qualidade oferecida pela rede.

O serviço de transmissão de dados da Internet baseado no melhor-esforço faz com que a comunicação do usuário dependa do estado da rede e da maneira como ele a utiliza no momento da comunicação. Essa característica da rede, sua não previsibilidade, impede ou restringi, que ela seja adotada como um meio definitivo para a comunicação de diversas aplicações que são de uso crítico para seus usuários.

No ano de 2004 foi criado o projeto EuQoS, dentro de um fundo de fomento a pesquisa tecnológica europeu, o *Sixth Framework Programme* (FP6), com o objetivo de pesquisar, integrar, testar e validar novas tecnologias que agreguem à infra-estrutura da Internet a capacidade a oferecer garantia de serviço fim-a-fim.

Esta monografia surge como um fruto da parceria realizada entre a Universidade Federal de Santa Catarina e o LAAS – Laboratoire dAnalyse et dArchitecture des Systemes – na França, para colaboração em pesquisa e desenvolvimento para o sistema EuQoS.

O sistema EuQoS permite que dois usuários conectados a Internet iniciem um sessão entre eles com garantia de Qualidade de Serviço. Isto é, uma sessão EuQoS estabelecida por um usuário garante a ele uma reserva de recursos de rede em todo o caminho até o segundo usuário.

1.1 **Objetivo**

Seu objetivo é especificar e implantar uma rede de testes (*testbed*) onde possam ser realizadas avaliações de desempenho em redes com QoS. Em particular predende-se avaliar a arquitetura de QoS fim-a-im EuQoS.

1.1.1 Objetivos específicos

- Estudar a plataforma EuQoS;
- Instalar um domínio EuQoS integrado ao *testbed* europeu do projeto EuQoS;
- Realizar testes de desempenho no link entre a UFSC e o LAAS;
- Instalar e configurar um *testbed* para testes de avaliação de desempenho em redes propondo uma metodologia para atacar o problema da necessidade da sincronização de relógios para medições de atraso de sentido único;
- Implantar de dois domínios EuQoS sobre DiffServ em Linux neste TestBed;
- Realizar testes de medição de tempo para iniciação de sessões EuQoS.

2 *Qualidade de Serviço - QoS*

Para que se ofereça ao usuário da Internet garantias a respeito da qualidade do serviço que ele vai experimentar, são necessárias diferentes camadas de gerenciamento e de implementação. Uma *macro*-arquitetura de QoS é composta por conceitos gerais de como a rede será organizada e gerenciada. Em um nível mais baixo, temos os equipamentos de rede, os roteadores, e os mecanismos e algoritmos que impactam diretamente na performance da rede que se traduzirá na qualidade experimentada pelo usuário. Este capítulo tem o objetivo de dar uma visão geral destes diversos níveis do problema de suportar QoS na Internet.

O capítulo está organizado da seguinte maneira. Primeiramente, descrevemos diferentes tipos de aplicações de tempo-real e seus requisitos de QoS. A classe de aplicações de tempo-real é enfatizada pois tem ganho muitas aplicações nos últimos anos e é esperado que esta tendência continue. Uma descrição básica dos diferentes parâmetros de QoS é apresentada na seção 2.3 e os blocos básicos utilizados para implementar garantia desses parâmetros são apresentados nas seções 2.4 e 2.5. As seções 2.7 e 2.8 discorrem sobre as arquiteturas projetadas pela IETF para que as redes ofereçam QoS. Descrevemos a arquitetura de Serviços Integrados, IntServ, como sendo a primeira arquitetura desenvolvida com esses propósitos e discutimos seus problemas de escalabilidade, entre outros, e como eles levaram ao projeto de uma nova arquitetura. Discutimos então esta segunda tentativa de desenvolver uma arquitetura capaz de implementar QoS na Internet que seja escalável, a arquitetura de Serviços Diferenciados, DiffServ. Na última seção descrevemos como a necessidade de um gerenciamento para que as redes DiffServ ofereçam garantias ponta a ponta, e a aplicação dos denominados *bandwidth brokers* para este fim, conduziram a conceitos que formam a base da plataforma EuQoS.

2.1 Aplicações de tempo-real e a necessidade de QoS

Os oponentes dos estudos sobre QoS afirmam que o problema da qualidade na Internet pode ser facilmente resolvido aumentando a largura da banda (ex. *overprovisioning*), já que esta tem se tornando um recurso cada vez mais barato. No entanto muitas das aplicações de tempo-real como voz sobre IP (VoIP), vídeo-conferência, e tele-medicina requerem garantias de atraso, variação no atraso e perda de pacote e não somente de largura de banda. Além disto, a premissa de que é possível oferecer mais largura de banda do que o necessário é sempre ameaçada pelo crescimento do número de usuários cotidianos da Internet e das aplicações de multimídia avançada que consomem rapidamente toda banda disponível, o que sempre nos leva de volta ao problema de limitação de largura de banda. Quando o tráfego de outros tipos de aplicações (como transferência de arquivos por http) concorrem pelos mesmo recursos de rede com as aplicações de tempo-real, estas ultimas podem sofrer grandezas variáveis e imprevisíveis de atraso, variação do atraso e perda de pacotes, especialmente na ausência de qualquer priorização. Soma-se a isto o fato de que quase todos os componentes de rede experimentam picos de uso em determinados momentos. Tais características da Internet levam à uma das principais motivações do estudo de QoS: a necessidade de proteção e priorização de determinados fluxos de tráfego. Outra característica importante é a disparidade entre as larguras de banda disponíveis no centro e nas bordas da Internet que faz com que os roteadores da borda tenham tipicamente mais congestionamento do que os no centro, gerando portanto uma grande necessidade de realizar tal proteção e priorização nos roteadores de borda.

2.2 Classes de aplicações dependentes de QoS

Esta seção resume as principais classes de aplicações que utilizam a Internet segundo suas características e requisitos de QoS.

Interativas e não-interativas: Aplicações interativas são normalmente intermediárias de interações homem-homem ou homem-máquina e envolvem uma sequência de transferências de interações entre as pontas. Existem também algumas aplicação máquina-máquina que são interativas, como em controle automatizado e sensoriamento. Este tipo de aplicações podem ser sensíveis aos diferentes parâmetros de QoS como vazão, atraso, variação do atraso e perda.

Aplicações não-interativas não fazem múltiplas interações entre as pontas durante

uma sessão. Como no caso de uma transferência de arquivo ou um backup de dados. A qualidade do desempenho deste tipo de aplicações está mais ligada a vazão.

Elásticas e não-elásticas: Aplicações elásticas tem um bom desempenho independente das condições da rede que lhes são oferecidas. Este tipo de aplicação não necessita de qualquer garantia de QoS além daquela normalmente oferecida pelo modelo de melhor-esforço e a confiabilidade do transporte, como a oferecida pelo protocolo TCP. Claramente, a maioria das aplicação de tempo-real, que são a motivação maior de do estudo de QoS, são não-elásticas e possuem requisitos mínimos de performance de rede para operarem. Podemos ainda classificar as aplicações de tempo-real em críticas (“hard real-time”) e não-críticas (“soft real-time”). As não-críticas continuam operando ainda que os requisitos mínimos de qualidade não sejam atendidos, quando isto ocorre há uma degradação no serviço prestado pela aplicação como ocorre com aplicações multimídia. Já aplicações de tempo-real críticas tem seu serviço interrompido quando não tem seus requisitos mínimos de qualidade atendidos, como ocorre com algumas aplicações de tele-medicina.

Tolerantes e intolerantes: Enquanto aplicações elásticas não impõem qualquer requisito de QoS, aplicações tolerantes impõem um intervalo. Com os parâmetros dentro deste intervalo a aplicação tolerante continua funcionando ainda que o nível ótimo de QoS não esteja sendo oferecido. Pode-se dizer então que uma aplicação tolerante é elástica dentro de um intervalo, mas se seus limites de QoS são violados ela não funciona corretamente, como no caso da telefonia IP. Outro exemplo de aplicação tolerante é VoD, aplicações de vídeo sobre a Internet normalmente toleram uma certa taxa de perda e de atraso e continuam executando. Já qualquer aplicação que estabelece valores fixos para seus parâmetros de QoS afim de funcionar é intolerante, como as aplicações de tempo-real hard?.

Adaptativas e não-adaptativas: Esta classificação está relacionada a qualidade percebida pelo usuário da aplicação. Aplicações adaptativas utilizam artifícios para manter a qualidade percebida pelo usuário ainda que haja degradação na qualidade de serviço oferecida pela rede. Isto pode ser feito, por exemplo, diminuindo a taxa de envio ou a resolução da transmissão, no caso de vídeo, ou mesmo utilizando técnicas específicas de compressão e correção de erro. Uma solução de adaptatividade conhecida é a aplicação de buffers em aplicações multimídia. Eles permitem uma degradação suave quando há alteração na performance de rede. Entre as aplicações adaptativas estão a maioria das aplicações de streaming de vídeo e voz. Aplicações não-adaptativas podem lidar com degradações de QoS mas isto implica diretamente em degradação do serviço percebido pelo usuário. É comum que aplicações adaptativas sejam tolerantes e que aplicações não-adaptativas

sejam intolerantes, mas isto não é exclusivo.

Vídeo/Áudio em tempo-real e streaming: Aplicações multimídia de áudio/vídeo podem ser de tempo-real como radio/TV pela Internet, telefonia IP, e vídeo-conferência, ou de streaming e não tempo-real como aplicações VoD. Aplicações multimídia de tempo-real tem requisitos de QoS mais restritivos e geralmente utilizam técnicas de adaptatividade. Já aplicações de streaming não tem uma restrição tão estrita do momento absoluto em que sua execução deve iniciar e podem atrasá-lo além do tempo de atraso máximo da rede para evitar a variação no atraso, utilizando um buffer.

Multimídia e dados ou computação em larga escala: Existem muitas outras aplicações que dependem de QoS que não as multimídia. Algumas delas também dependem da entrega periódica e pontual de dados, como as grades computacionais, aplicações de transação eletrônica e instrumentos controlados remotamente. Outras dependem da transferência de uma grande quantidade de dados, como acontece com sistemas de armazenamento distribuído ou espelhado em que é necessário transportar uma enorme quantidade de dados entre servidores geograficamente distantes em um tempo determinado.

2.3 Parâmetro de QoS

Antes de discutirmos as diferentes arquiteturas e tecnologias aplicadas para a obtenção de QoS, vamos discutir os parâmetros de performance de rede para os quais requisitos de QoS são normalmente traduzidos. Os parâmetros de qualidade de serviço são quantificações de determinadas características de um fluxo de pacotes. Tais quantias devem estar dentro de limites especificados pelo nível de qualidade requerido pelas aplicações.

Os seguintes parâmetros formam a base da Qualidade de Serviço (qualquer outros podem ser mapeados para estes (EL-GENDY; BOSE; SHIN, 2003)).

Vazão é a quantidade efetiva de unidades de dados transmitidas por unidade de tempo (ex., bits/segundo). Normalmente este parâmetro é referenciado como “garantia de banda”. Garantia de vazão envolve alocação da capacidade do link e a capacidade de processamento dos nós intermediários.

Atraso é o tempo transcorrido entre a saída do pacote da fonte e sua chegada no destino. Existe um atraso associado a cada uma das camadas de rede utilizadas entre a fonte e o destino - camada de aplicação, camada de transporte, camada de rede, camada de enlace e camada física. A qualidade requisitada é expressa em termos do limite

de maior atraso D_{max} . Um dos problemas enfrentados durante as etapas de medições deste trabalho é a medição do atraso em uma única direção (*one-way delay*), isto acontece devido a problemas de sincronização de relógio. Muitas vezes o intervalo total de ida e volta é utilizado (*round-trip delay*), mas não é uma métrica muito boa devido a prováveis diferenças entre as rotas de ida e de volta dos pacotes.

Variação de atraso, como no caso do atraso, também possui sua qualidade medida em relação a um limite J_{max} . Apresentamos aqui três definições da quantificação da variação do atraso, embora existam outras. Este parâmetro é comumente chamado “jitter”.

1. Pode ser calculado como as diferenças entre os tempos entre-partidas P_i e os tempo entre-chegadas C_i das i -ésima e $(i-1)$ -ésima unidades de dados, ex., $J_i = P_i - C_i$.
2. Pode ser calculado como a diferença entre os atrasos da i -ésima e a $(i+1)$ -ésima unidades de dados, ex., $J_i = D_{i+1} - D_i$.
3. Nos padrões RTP (GROUP et al., 1996), o atraso na variação é medido de maneira acumulativa através da seguinte equação: $J = J + \frac{|D(i-1,i) - J|}{16}$.

Perda é a razão entre o número de unidades de dados que não chegaram até o destino e o número de unidades de dados enviadas medida em um intervalo de tempo. Pode ser lida como uma “probabilidade” de perda. Retransmissões não alteram o valor da taxa de perda, são somente um meio de recuperá-las.

Confiabilidade está ligado com a perda mas representa um conceito diferente. Confiabilidade diz respeito a porcentagem de unidades de dados que enfim foram recebidas corretamente no destino. Muitos protocolos utilizam retransmissão para recuperar uma perda e oferecer confiabilidade à camada superior.

2.4 Controle de Admissão

Independente da abordagem de QoS aplicada, a rede ainda possui recursos limitados. Os links, os buffers dos equipamentos e suas capacidades de processamento são limitadas. Sendo assim, a chave de toda rede com QoS está na maneira como distribui seus recursos entre todas as requisições de serviços dos diferentes clientes. Controle de admissão é o processo de comparar os requisitos de serviço com os recursos disponíveis e decidir se aceita ou rejeita o pedido de serviço baseando-se no critério de que a admissão de um novo

pedido não pode levar a um nível de qualidade de serviço inaceitável para este ou para os serviços previamente admitidos. Controle de admissão pode ser realizado explicitamente, rejeitando o pedido de serviço, ou implicitamente como no caso da prática de controle de banda.

Controle de admissão explícito é um dos processos fundamentais na plataforma EuQoS, como discutiremos no capítulo sobre a plataforma.

2.5 Blocos fundamentais para provisão de QoS em nível de rede

Uma implementação de QoS em nível de rede requer funcionalidades adicionais dos dispositivos de rede além das funcionalidades básicas de encaminhamento e roteamento de pacotes. Entre estas estão os seguintes. Enfileiramento (*queuing*) e escalonamento (*scheduling*), policiamento (*policing*) e suavização (*shaping*) e gerenciamento de buffer. Vamos descrever aqui brevemente alguns desses blocos básicos. A combinação deles utilizada em uma rede pode variar dependendo das capacidades do hardware e das garantias de QoS que se deseja oferecer.

2.5.1 Escalonamento

Existe uma grande diversidade de algoritmos propostos para realizar escalonamento de pacotes na camada de rede. O objetivo deles é sempre distribuir a quantidade de vazão disponível no link entre diferentes classes de serviço de maneira que cada uma receba uma fatia previamente definida ou justa, oferecendo então algum tipo de garantia, normalmente estatística, dos valores dos parâmetros de rede como atraso, variação no atraso e perda de pacotes.

Escalonamento FIFO (primeiro a entrar, primeiro a sair): este é a maneira mais simples de escalonar pacotes. Os pacotes obtêm serviço da rede na sequência em que são recebidos. Neste caso, o atraso e a taxa de perda de pacotes são proporcionais ao tamanho do buffer disponível para a fila. Esta política de escalonamento funciona melhor quando todos os tráfegos possuem características semelhantes, já que nestes casos existe uma disputa equilibrada pelos recursos. No entanto ela não é capaz de priorizar um fluxo que naturalmente seria desfavorecido, o que faz com que não seja uma política passível de ser aplicada para prover diferenciação e garantias de QoS

para um tráfego particular.

Escalonamento por prioridade: um escalonador por prioridade estático é baseado em múltiplas filas FIFO com prioridades diferentes. As filas são servidas por ordem de sua prioridade. Ele é capaz de oferecer performance preditiva (EL-GENDY; BOSE; SHIN, 2003) se todos os fluxos entrando no escalonador (independente das prioridades de suas filas) tem o mesmo tamanho de pacotes e taxa de chegada. Sabemos que um pacote necessita esperar por todos os outros pacotes em filas de maior prioridade. Isto pode fazer gerar *starvation*¹ para classes de mais baixa prioridade.

Escalonamento por GPS (Generalized processor sharing) e variantes: Para resolver o problema de *starvation* do escalonador por prioridade o GPS propõe atribuir filas lógicas para cada fluxo e servir uma quantia de dados infinitesimal de cada fila com um dado período de tempo finito. Ele é eficiente em realizar um escalonamento justo mas não é implementável dado ao requisito de servir quantias infinitesimais de dados. Exemplo de implementações realizadas para o GPS são o *round robin* (RR) e o *weighted-round robin* (WRR). No escalonamento RR, cada fila envia um pacote em cada um dos períodos de tempo ou ciclos. O RR é justo já que independente da taxa de chegada os fluxos enviam somente um pacote por ciclo. No entanto, esta justiça só existe para fluxos com o mesmo tamanho de pacotes. No caso de existir uma diferença significativa entre os tamanhos dos pacotes dos diferentes fluxos, os fluxos com pacotes maiores consumirão uma fatia maior de recursos. O WRR (KATEVENIS; SIDIROPOULOS; COURCOUBETIS, 1991) foi desenvolvido para resolver o problema. Ele permite que cada fila envie n pacotes, ao invés de um único pacote, onde n é um peso atribuído à fila. Filas com pacotes menores recebem maior peso e assim recebem tratamento justo. No entanto o WRR não é eficiente para fluxos com tamanho de pacote variável, já que ele pressupõe o conhecimento do tamanho do pacote *a priori*. O *deficit round robin* (DDR) (SHREEDHAR; VARGHESE, 1996) ataca este problema fazendo com que o período de serviço oferecido varie em função do tamanho dos pacotes.

Escalonamento por WFQ (Weighted fair queueing): o escalonador WFQ (DEMERS; KESHAV; SHENKER, 1989) foi projetado para resolver o problema de *starvation* do escalonador por prioridades enquanto tenta aproximar o escalonador GPS. Este escalonador calcula o *tempo de fim de envio* de cada pacote como se ele fosse servido

¹O termo faz referência ao caso em que uma fila tem o acesso aos recursos de rede negado perpetuamente.

pelo GPS e utiliza este número para ordenar o envio dos pacotes. Por que o WFQ possibilita uma divisão justa entre todas as filas é um dos algoritmos mais implementados (EL-GENDY; BOSE; SHIN, 2003) nos roteadores comerciais. Variações deste escalonador que executam de maneira mais eficiente em hardware também foram propostas (REXFORD; GREENBERG; BONOMI, 1996).

Escalonamento por CBQ (Class-based queuing): o CBQ (FLOYD; JACOBSON, 1995) é um algoritmo de escalonamento que permite uma divisão hierárquica dos recursos entre diversas classes de tráfego para um link particular como mostrado na Figura 2.1. Esse algoritmo cria uma árvore de compartilhamento para todas as classes que serão suportadas pelo link. Cada classe, sendo interna ou folha, deve receber em intervalos de tempo a fatia de recursos estabelecida para ela. Uma classe pode ser configurada como *isenta*, *limitada* ou *isolada*. Uma classe isenta pode usufruir de 100% dos recursos do link. Um classe limitada não tem permissão para emprestar recursos das classes pais ainda que estas disponham de recursos excedentes. Uma classe isolada não permite que outras classes que não suas classes filhas emprestem seus recursos excedentes.

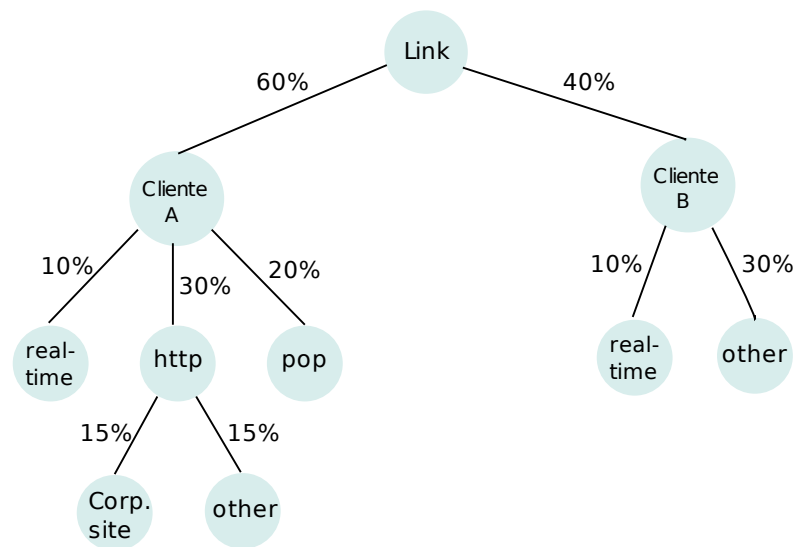


Figura 2.1: Estrutura de compartilhamento hierárquico de link.

2.6 Acordos de Nível de Serviço

Um acordo de nível de serviço (SLA - Service Level Agreement) é um documento que define formalmente a relação entre um cliente e um prestador de serviços. SLAs são

aplicadas nos mais diversos segmentos de mercado. Uma SLA contém (VERMA, 2004), em linhas gerais, uma descrição da natureza do serviço prestado; o nível de desempenho esperado dos serviços; o procedimento para se reportar problemas com o serviço; a duração da janela de tempo para a resolução de problemas; o processo para se monitorar e relatar o nível de serviço sendo oferecido; as consequências para o provedor de serviço quando não cumpre com suas obrigações; cláusulas de escape e restrições (casos especiais em que o provedor é eximido da responsabilidade pelo descumprimento de suas obrigações).

No contexto de redes IP, cláusulas que definem o nível de desempenho esperado são tipicamente definidas em termos de desempenho e disponibilidade, como pela média mensal do tempo de atraso e o tempo total de interrupções não agendadas do serviço do mês.

2.6.1 Gerenciamento de buffers

Em redes de alta velocidade que utilizam nós com um “produto do atraso de vazão” (*delay-bandwidth product*²) alto, os gateways são projetados com filas equivalentemente grandes para suportar congestionamentos transientes. O protocolo TCP detecta um congestionamento assim que um pacote é descartado pelo gateway. No entanto, não seria desejável ter filas grandes (possivelmente da ordem do produto de atraso de vazão (FLOYD; JACOBSON, 1993)) que estivessem cheias todo o tempo, o que ocasionaria um aumento do atraso na rede. Desta maneira, em redes de alta velocidade, é de maior importância manter a vazão alta mas o tamanho médio das filas baixo.

Gateways RED (*random early detection*) (FLOYD; JACOBSON, 1993) são muito empregados nestes cenários. Eles detectam congestionamentos incipientes computando o tamanho médio da fila. Quando o tamanho médio da fila ultrapassa um marco determinado, o gateway começa a descartar pacotes segundo uma probabilidade p onde p é uma função do tamanho médio da fila. Esses equipamentos mantêm o tamanho da fila baixo enquanto permitem queimas (*bursts*) ocasionais. Sua eficácia está em fazer com que protocolos de camada de transporte orientados a conexão, como o TCP, diminuam suas janelas quando há um congestionamento incipiente, por isso sua aplicação evita congestionamentos (ENHANCED..., 1999). O WRED (DISTRIBUTED..., 2008) (*weighted random early detection*) é uma variação do RED que descarta os pacotes seletivamente baseado no campo de precedência IP. Pacotes com uma precedência maior tem menor probabilidade de serem descartados. Este mecanismo é normalmente usado (DISTRIBUTED..., 2008) nos roteadores centrais da rede. Os roteadores de borda marcam o campo

²O número de bits que o emissor precisa transmitir antes que o primeiro bit chegue no destino.

de precedência IP conforme eles entram na rede e os roteadores centrais utilizam o WRED sobre esta marcação.

2.6.2 Policiamento

Policiamento de tráfego é normalmente aplicado (EL-GENDY; BOSE; SHIN, 2003) nas bordas da rede próximo à fonte. Este mecanismo tem o objetivo de verificar se as características do tráfego estão em conformidade com o acordo de nível de serviço negociado e garantir que somente os recursos contratados sejam utilizados. Quando um fluxo difere do esperado, ele pode ter pacotes descartados ou marcados para que recebam um tratamento de menor prioridade pela rede. O policiamento de tráfego pode se basear em um único parâmetro negociado, como taxa máxima de envio, ou em combinação de parâmetros, como taxa máxima de envio, tamanho dos pacotes, hora do dia, etc.

Os *token buckets* (e *leaky buckets*) são os mecanismos mais utilizados para policiamento de tráfego em um nó da rede. Um *token bucket* é com posto por um “balde” (*bucket*) de profundidade b que gera tokens a uma taxa r . Cada pacote que entra consome um token (ou um número de tokens diretamente proporcional ao tamanho do pacote dependendo da implementação) antes que possa ser encaminhado para o interior da rede. Se ao chegar não existem tokens no balde o pacote é descartado ou marcado para receber um tratamento de menor prioridade pela rede. Este mecanismo faz com que um fluxo seja conformante se sua taxa de envio média for menor ou igual a r e o tamanho de suas queimas (*bursts*) forem menor do que b . Para qualquer período de tempo t a quantia máxima de tráfego que pode trafegar é $r \cdot t + b$. *Token buckets* também podem ser utilizados para suavização (*shaping*) de tráfego como mostrado na Figura 2.2.

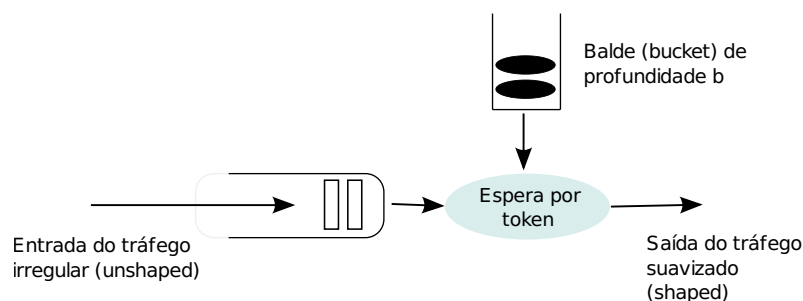


Figura 2.2: Suavização de tráfego com *Token Bucket Filter*.

Réplica de figura em (EL-GENDY; BOSE; SHIN, 2003).

2.7 Arquiteturas de serviços integrados - IntServ

Uma característica das redes IP é não serem orientadas a conexão. Elas funcionam com o conceito de “datagramas” que são roteados de um “nó” para o próximo, não havendo naturalmente ligado a elas o conceito de circuito virtual ou caminho virtual como é com as redes ATM. Cada pacote (datagrama) é uma unidade independente que contém todas as informações necessárias para que seja roteada da fonte ao destino. E como sabemos, a Internet foi construída sobre o modelo de melhor-esforço para a entrega destes datagramas. Como a maior parte do tráfego na Internet utiliza TCP ou UDP sobre IP, a IETF publicou (BRADEN; CLARK; SHENKER, 1994) em 1994 o padrão de Serviços Integrados IntServ. Uma aplicação que utiliza a arquitetura IntServ utiliza-se de um protocolo de reserva de recursos (RSVP) para sinalizar o pedido de reserva de recursos nos roteadores intermediários até o destino, fechando assim um túnel virtual de QoS. Este pedido de reserva contém o perfil do tráfego que será enviado e a qualidade requisitada, esses dados permitem a cada roteador calcular os recursos que necessita reservar (como espaço em buffer) e a política de escalonamento que deve instalar para que o QoS ponta a ponta requisitado seja atendido. Quando a camada de enlace suporta QoS, como no caso das redes ATM, o roteador é também responsável por negociar com a camada de enlace a instalação do suporte apropriado a QoS (WHITE; CROWCROFT, 1997). No caso da utilização de camadas de enlace passivas a QoS, esse mapeamento é trivial já que a transmissão é unicamente controlada pelo escalonador de pacotes do roteador.

A arquitetura IntServ introduziu novas classes de serviço. Essas diferentes classes são utilizadas de acordo com os requisitos de QoS necessários à sessão e configuradas nos roteadores. Uma sessão que não pertence a estas classes é tratada segundo o modelo melhor-esforço e estará sujeita a disponibilidade dos recursos no momento.

2.7.1 O modelo IntServ

A arquitetura IntServ foi projetada para prover QoS para cada fluxo individualmente. Uma sessão que deseja obter um QoS específico necessita iniciar um procedimento de configuração utilizando RSVP. O RSVP configura estados voláteis (*soft-states*) em cada roteador no caminho da fonte até o destino. Os estados definem a classe e os recursos necessários para o início da sessão. As reservas permanecem válidas enquanto a sessão está ativa, mas expiram caso não sejam atualizadas periodicamente (por isso voláteis). Neste modelo de serviço, caso existam N seções individuais com reservas de recursos configuradas

passando pelo roteador ao mesmo tempo, é necessário que o roteador mantenha uma tabela com as informações dos estados das N sessões. É importante notar que a reserva de recursos é realizada ponta a ponta em toda a rota da sessão, e que no caso de uma mudança de rota é necessário refazer a reserva na nova rota.

Este modelo é o mais eficaz possível já que controla os requisitos de QoS dos fluxos individuais.

Cada roteador necessita aplicar controle de admissão para as requisições de reserva para garantir que conseguirá atender aos requisitos do pedido. Uma vez que a reserva para a sessão foi feita de maneira apropriada em cada um dos roteadores no caminho, o fluxo de dados recebe da rede o compromisso de QoS fim a fim contanto que não haja alteração na rota dos pacotes durante a sessão e que as informações sobre as características do fluxo passadas ao roteadores sejam fieis. As ações específicas de policiamento e condicionamento aplicadas aos pacotes foram discutidas na seção 2.5.

2.7.2 Classes de serviço

A arquitetura IntServ introduziu duas novas classes de serviço além da tradicionalmente suportada melhor-esforço (BE): Serviço Garantido (GS), e serviço de carga controlada (CL).

Serviço garantido: esta classe provê à sessão um nível de vazão garantido, limites fixos para o atraso fim a fim e nenhuma perda de pacotes (desde que estes sejam conformantes). É destinada a aplicações com estritos requisitos de QoS fim a fim. Para instalar esta classe de serviço, o roteador necessita de informações das características do tráfego. Baseado nessas informações o roteador pode calcular, utilizando modelo matemáticos, a quantia de vazão e de espaço em *buffer* que será necessário reservar para a sessão.

Serviço de carga controlada: Esta classe não prove garantias quantitativas de QoS. O compromisso do roteador para com fluxos desta classe é de oferecer um serviço equivalente ao provido por uma rede melhor-esforço em situação de baixa carga. A diferença entre esta classe e a de melhor-esforço é que ela não permite que o serviço se deteriore caso a carga na rede aumente. Ela é destinada à aplicações que podem tolerar um certo nível de perda e de atraso contanto que estes fiquem dentro de determinados limites, como acontece com as aplicações chamadas tolerantes.

2.7.3 O protocolo RSVP

O protocolo de reserva de recursos RSVP foi projetado para permitir às pontas e roteadores envolvidos em uma sessão de tráfego interagir para configurar os estados necessários para a instalação dos serviços descritos anteriormente. O procedimento de reserva, como descrito na RFC2210 (WROCLAWSKI, 1997) que a define, inicia com o emissor enviando uma mensagem *PATH* para o destinatário que atravessará todos os roteadores no caminho. O conteúdo desta mensagem, de maneira resumida, é: o perfil de tráfego do fluxo que será gerado pelo emissor; dados utilizados para computar os parâmetros de QoS acumulados ao longo do caminho; as informações da rota percorrida para que a rota reversa seja utilizada para retornar a mensagem de confirmação *RESV* como explicado abaixo.

Quando a mensagem *PATH* chega ao destinatário, ele responde com uma mensagem *RESV* que contém a especificação do pedido de reserva, dado que o atraso ponta a ponta e outros parâmetros estejam dentro de limites aceitáveis. A mensagem *RESV*, conforme é aceita pelos roteadores no caminho, faz com estes instalem as devidas reservas e filtros. Caso ocorra um erro ou não haja recursos suficientes disponíveis, então ou uma mensagem *PATH_{err}* ou *RESV_{err}* é gerada pelo respectivo roteador que retorna ao emissor removendo qualquer reserva que já tenha sido instalada.

Quando a sessão chega encerra as mensagens *PATH_{tear}* e *RESV_{tear}* são enviadas e removem todas as reservas instaladas em todos os roteadores no caminho.

A IETF publicou a especificação de um novo protocolo de sinalização, o NSIS. Este protocolo será discutido mais detalhadamente adiante por ser o protocolo utilizado pela plataforma EuQoS.

2.7.4 Avaliação do modelo IntServ

Vale a pena aqui discorrer sobre as principais características deste modelo afim posteriormente entendermos por que a arquitetura proposta no capítulo seguinte foi proposta. O modelo IntServ aliado ao RSVP oferece flexibilidade em ir ao encontro das necessidades individuais de QoS de um fluxo já que as reservas são feitas individualmente para cada um deles; provê QoS garantido e determinístico já que os recursos calculados necessários são reservados em todos os roteadores no caminho; permite a reconfiguração de reservas no novo caminho em casos de mudança de rota; suporta sessões multicast, bastando que os diferentes destinatários respondam com a mensagem *RESV* para que os recursos sejam alocados devidamente.

No entanto a sua adoção tem sido restringida pelas seguintes razões:

- Problemas de escalabilidade. O fato de que cada roteador precisa guardar informações sobre os estados de cada um dos fluxos faz com que o modelo não escale para tráfegos em backbones de redes.
- O atraso para o início das sessões, resultado da necessidade de esperar até que a reserva seja feita em todo o caminho, é restritivo para algumas aplicações de tempo real e em outras aplicações cujo tempo para esta configuração é da mesma grandeza do tempo de duração da sessão.
- Por que os cálculos dos parâmetros a serem reservados são realizados antes do início da sessão com base na especificação do perfil de tráfego esperado, uma alteração no funcionamento da aplicação que gere mudanças neste perfil de tráfego pode causar problemas imprevisíveis a não ser que o novo perfil de tráfego seja conhecido de utilizado nas novas reservas.
- O modelo IntServ não é compatível com o protocolo IPSec. Os roteadores necessitam ler campos da camada de transporte do pacote para poder identificar o fluxo a qual ele pertence e estes campos são cifrados neste protocolo. Este problema foi resolvido com a introdução do IPv6, no qual é possível identificar o fluxo a qual o pacote pertence pelas informações da camada de rede.
- O RSVP, sendo um protocolo baseado no receptor, não é naturalmente compatível com o modelo cliente-servidor em que o cliente inicia a comunicação mas o servidor é quem envia os dados. Isto requer que o protocolo seja aplicado na direção inversa à da comunicação: o servidor envia a mensagem PATH e o cliente responde com a RESV.

2.8 Arquiteturas de Serviços Diferenciados - DiffServ

Com o objetivo de encontrar uma alternativa que contornasse os problemas do modelo IntServ, a IETF montou um grupo de trabalho para desenvolver e padronizar os princípios de uma nova arquitetura, chamada de Serviços Diferenciados ou DiffServ (BLAKE et al., 1998). Esta arquitetura foi adotada pelo Consórcio da Internet2 (INTERNET2. . . ,) para provimento de QoS na rede QBone³.

³QBone é uma “cama de testes” experimental para estudos relacionados a implementação de DiffServ na Internet2

2.8.1 O modelo DiffServ

Diferente do seu predecessor, IntServ, o modelo DiffServ não se baseia em sinalização por fluxo e reservas de recursos em todo o caminho, ele separa todo o tráfego em um número de agregados⁴ diferentes, cada um com diferentes requisitos de QoS, e faz com que os diferentes agregados sejam tratados diferentemente pelos dispositivos da rede. Isso, com efeito, faz com que seja impossível oferecer garantias determinísticas e quantitativas à um determinado fluxo, como no modelo IntServ, o que se oferece é uma garantia qualitativa da diferenciação no seu tratamento pela rede. Outra diferença é que este modelo define o padrão das políticas de encaminhamento que devem ser aplicadas de maneira independente por cada um dos roteadores, e não serviços ponta a ponta como no caso das classes de serviço garantido e controlado do modelo IntServ. E finalmente, o controle e configuração do nível de QoS oferecido pela rede tem sua ênfase nos acordos de nível de serviço (SLA) definidos entre domínios e não nas sinalizações dinâmicas por sessão como no modelo IntServ.

As próximas seções tratarão de descrever os principais componentes da arquitetura DiffServ e seus objetivos na provisão de QoS.

2.8.2 O campo DS

Para realizar a agregação do tráfego é necessário utilizar alguns bits no cabeçalho dos pacotes que permitam diferenciá-los. Estes bits, no contexto DiffServ, constituem o “campo DS”. A RFC 2474 (NICHOLS et al., 1998) especifica o campo DS que coincide com o byte ToS no cabeçalho IP. O conteúdo deste campo é chamado *DS codepoint* (DSCP), e ocupa somente seis bits do byte ToS.

O DSCP é o campo utilizado para definir o tipo de tratamento que o pacote receberá dos equipamentos da rede. Estes diferentes tipos de tratamento são denominados de *per-hop behavior* (PHB). A arquitetura DiffServ é baseada sobre a definição de alguns poucos PHBs diferentes, que cobrem os agrupamentos básicos dos diferentes requisitos de QoS. A marcação dos bits do campo DS dos pacotes define por qual dessas PHBs ele será tratado. Atualmente estão definidos três PHBs diferentes, em adição ao serviço padrão de melhor-esforço, que descreveremos na sequência.

⁴Um agregado de tráfego é um grupo de fluxos de dados que recebem o mesmo tratamento da rede.

2.8.3 PHB CS - Case Selector

Este PHB é definido na RFC 2474 (NICHOLS et al., 1998) para manter compatibilidade com os bits de precedência do byte ToS do IP. Ele pode ser utilizado para definir oito diferentes níveis de prioridade sendo que o maior valor indica maior prioridade de encaminhamento. Entre as tecnologias que permitem implementar esta PHB estão o WFQ e o CBQ descritos na seção 2.5.

2.8.4 EF PHB - Expedited Forwarding

Definido na RFC 3246 (DAVIE et al., 2002), este PHB foi projetado para oferecer um serviço de baixo atraso, baixa variação no atraso e baixa perda sobre uma vazão garantida. O princípio é tentar fazer com que os pacotes que tem seu DSCP marcados como EF encontrem sempre filas de encaminhamento pequenas no roteadores. Isto normalmente é atingido fazendo com que os recursos de encaminhamento sejam alocados para esses pacotes em uma taxa sempre maior do que a taxa com que eles chegam. Esta PHB é utilizada por aplicações com requisitos estritos de atraso e variação no atraso.

A RFC 3246 define formalmente esta PHB em termos de tempo ideal e real de saída de um pacote EF como função da taxa de envio da interface do dispositivo, do tempo de chegada e do tamanho do pacote. Define também limites para o atraso dos pacotes. No mesmo documento que o especifica são feitas considerações sobre quais mecanismos podem ser utilizados para implementar este PHB, dentre eles filas de prioridades e WRR.

2.8.5 AF PHB - Assured Forwarding

Este PHB foi projetado para oferecer serviços de pouca perda e vazão garantida e no entanto não oferecer quaisquer garantias de atraso e variação no atraso. Ele foi definido pelo grupo de trabalho DiffServ da IETF na RFC 2597 (HEINANEN et al., 1999). Este PHB consiste em três diferentes comportamentos de encaminhamento de pacotes, AFx1, AFx2 e AFx3 em ordem crescente de precedência de descarte (o último tem maior probabilidade de sofrer descartes do que o primeiro). O símbolo “x” representa uma entre as diferentes classes AF. Os projetistas recomendam o uso de quatro classes AF independentes e três ordens de precedência de descarte por classe. Sugerem também que os serviços mais prioritários sejam mapeados para a classe de maior índice por questões de compatibilidade com a precedência IP (SILVA, 2005). Os pacotes não podem sofrer reordenamento entre as classes. Esta PHB oferece então diferentes níveis de serviços dadas as possíveis

combinações de classes e precedência.

A PHB AF é normalmente implementada através de técnicas de gerenciamento de buffers, WRED, e se torna mais proeminente em casos de congestionamento de rede, quando pacotes precisam ser descartados.

A Figura 2.3 mostra como alguns dos mecanismos discutidos na seção 2.5 podem ser integrados para implementar três diferentes PHBs.

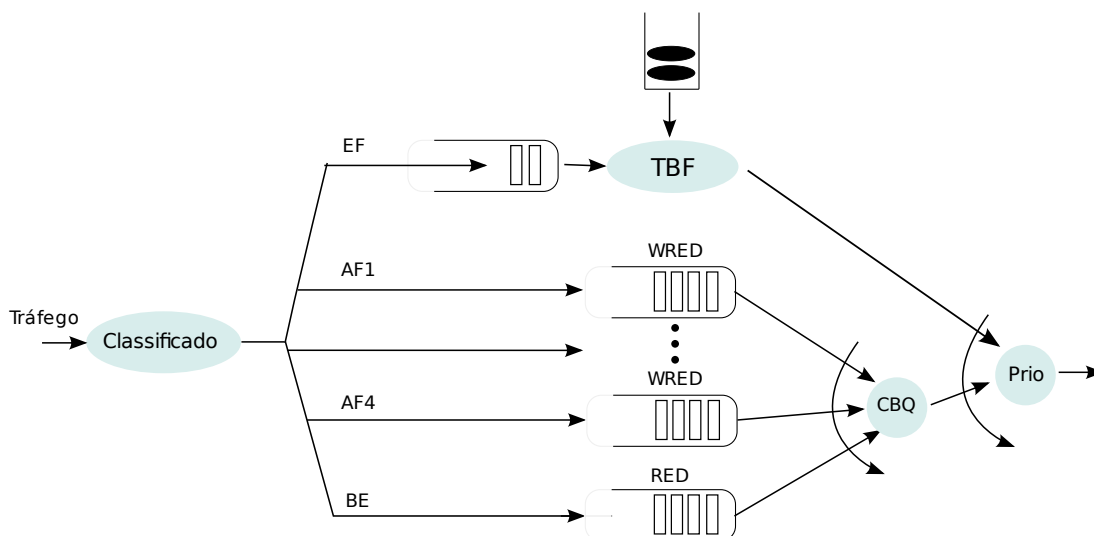


Figura 2.3: Um exemplo de implementação dos blocos básicos da arquitetura DiffServ.

Réplica de figura em (EL-GENDY; BOSE; SHIN, 2003).

2.8.6 PDB - Per-Domain Behavior

Os PHBs são instalados individualmente por cada roteador em um domínio⁵ DiffServ. Um grupo de pacotes que são tratados pelos mesmos PHBs em todos os nós enquanto atravessam um domínio DiffServ é denominado de *behavior aggregate* (BA). O termo BA se torna então um sinônimo do que vamos definir aqui. O termo PDB (*per-domain behavior*) é utilizado para definir um serviço global, que oferece determinados parâmetros de QoS, que um grupo de pacotes marcados com o mesmo DSCP recebem do domínio. Sendo assim um PDB define o serviço oferecido entre a entrada e a saída dos pacotes em um domínio. São os atributos dos diferentes PDBs (vazão, taxa de perda e limites de atraso) que são anunciados como especificações de nível de serviço (SLSs) por um domínio.

⁵Por domínio DiffServ, queremos referenciar a parte de uma rede que está sob uma administração única e que implementa DiffServ

Estes atributos, no caso de domínios DiffServ, são normalmente limites estatísticos e porcentagens, não valores fixos.

Cada PDB é construído como um conjunto de PHBs instalados nos nós do domínio de maneira que o resultado final das políticas de encaminhamento aplicadas por cada um deles resulte no cumprimento do nível de serviço para o qual ele foi projetado. O grupo de trabalho DiffServ da IETF propõe alguns PDBs, mas é incumbência de cada domínio definir o grupo de PDBs que ele oferecerá como diferentes tipos de serviço.

2.8.7 O funcionamento de uma arquitetura DiffServ

Tendo conhecimento dos blocos fundamentais que formam uma arquitetura DiffServ e para que possamos compreender a dinâmica de seus relacionamentos vamos discutir como uma implementação dessa arquitetura funciona.

Uma rede que suporta DiffServ é composta por um conjunto de domínios DiffServ, esses domínios são, no contexto mais geral da Internet, ASs⁶. Cada domínio tem se relaciona com seus vizinhos através de acordos de nível de serviço.

Vamos descrever aqui a jornada típica de um pacote em uma rede DiffServ desde sua saída da fonte até atingir o destino. O pacote é primeiramente medido e comparado com o perfil de tráfego negociado entre o cliente e seu provedor de serviços de rede, pacotes considerados não conformantes são ou descartados ou não marcados na etapa seguinte. O pacote tem então o seu campo DSCP marcado conforme o nível de serviço contratado. A marcação do pacote pode ser feita ou pelo próprio emissor, ou no primeiro roteador na entrada da rede do provedor de serviço (*leaf router*), ou mesmo nos roteadores de borda (podendo inclusive ocorrerem remarcações sucessivas nos diferentes domínios).

Uma vez que o pacote está marcado ele se torna parte de um agregado (BA) juntamente com outros pacotes com a mesma marcação no DSCP. No roteador de entrada de cada domínio o pacote é sujeitado à políticas de condicionamento de tráfego conforme o contrato estabelecido sofrendo remarcações quando preciso.

Os roteadores internos ao domínio implementam as políticas de encaminhamento para os diferentes PHBs que o domínio utiliza. No roteador da borda de saída o pacote pode ser novamente sujeitado a políticas de condicionamento para que o tráfego de saída se adeque ao contrato feito com o domínio vizinho.

⁶Um Autonomous System é um conjunto de redes IP e roteadores sobre o controle de uma única entidade.

É visível então que a maior complexidade na arquitetura DiffServ é atribuída aos roteadores de borda - que verificam todo o tráfego segundo os contratos, executam a marcação e condicionamento quando preciso - e aos roteadores internos resta a simples execução do encaminhamento conforme os PHBs. Embora a agregação reduza a flexibilidade e o poder de aplicar QoS a fluxos específicos, faz com que a arquitetura seja escalável.

2.8.8 Bandwidth Broker

Para que se possa oferecer serviços de QoS ponta a ponta sobre múltiplos domínios DiffServ é necessário uma camada superior de gerenciamento. A arquitetura DiffServ de dois bits (NICHOLS; JACOBSON; ZHANG, 1999) propôs a implementação de mediadores, os Bandwidth Brokers (BB), para este propósito. Os BBs residem dentro dos domínios ou entre domínios adjacentes. Eles são responsáveis por negociar com os outros BBs o estabelecimento de serviços ponta a ponta e por manter as informações de estado de cada um destes serviços (ao invés de atribuir isto individualmente aos roteadores, como no modelo IntServ). Nesta arquitetura um cliente entra em contato com o BB de sua rede e requisita um nível de serviço; caso o pedido seja conformante, este BB vai contactar o BB adjacente e assim por diante até o BB da rede do destino. Cada BB informa o perfil do tráfego e o nível de serviço requerido ao vizinho. Um vez que as requisições são confirmadas pelos BBs eles configuram a classificação e o condicionamento nos roteadores de borda de seus respectivos domínios para que o tráfego do cliente seja mapeado para os PDBs apropriados.

Um outro método de estabelecer serviços ponta a ponta é utilizar o protocolo de sinalização RSVP, mas diferente do modelo IntServ, com agregação (BAKER et al., 2001). Nesta caso a sinalização não tem o objetivo de estabelecer estados de reserva de recursos em todos os nós no caminho do fluxo, como no IntServ, mas o de sinalizar cada a entrada de cada uma das redes no caminho sobre com qual classe de agregação deverá tratar um determinado fluxo que a atravessa.

Os *Bandwidth Brokers* foram desenvolvidos como parte da chamada estrutura de policiamento (“policy framework”) desenvolvida pela IETF (EL-GENDY; BOSE; SHIN, 2003). O policiamento é utilizado para regular o acesso aos recursos da rede e serviços conforme critérios administrativos. Esta estrutura de policiamento realiza controle de admissão e provisão de recursos através de duas entidades principais: os PDPs (*Policy Decision Point*) e os PEPs (*Policy Enforcement Point*). Um PDP armazena em uma base de dados todas as informações administrativas necessárias para as tomadas de decisões

sobre os serviços prestados pelo domínio e distribui essas decisões aos PEPs (normalmente utilizando COPS (DURHAM et al., 2000)), responsáveis por aplica-las ao tráfego de rede. Um *bandwidth broker* é considerado um PDP na estrutura de policiamento.

2.8.9 Avaliação do modelo DiffServ

A arquitetura DiffServ apresenta as seguintes vantagens sobre a IntServ. Esta arquitetura é escalável já que trabalha com QoS para agregados de tráfego e não para fluxos individuais, e não necessita armazenar informações de estados nos roteadores; não há atraso gerado pela necessidade de sinalizações já que os serviços de QoS são construídos sobre SLAs e políticas pré-definidas entre os domínios; não há necessidade de controle de admissão nos roteadores já que isto é realizado pelo policiamento e remarcação nos roteadores de borda; é compatível com o protocolo IPsec já que necessita ler somente o cabeçalho IP dos pacotes (embora existam problemas (BLAKE et al., 1998) (NICHOLS et al., 1998) em lidar com os campos DS “interno” e “externo” em túneis IPSec).

Apesar destas vantagens, algumas restrições necessitam ser superadas para que o modelo seja adotado na prática:

- Como proposto, o modelo não provê garantias de QoS ponta a ponta para o tráfego na Internet, mas somente especifica como os domínios podem implementar diferenciação do tratamento do tráfego através de diferentes classes.
- Para que seja adotado, é necessário desenvolver sua interoperabilidade com outros padrões como MPLS e ATM.
- Para que os usuários possam usufruir do modelo DiffServ é necessário que suas aplicações implementes APIs com suporte a DiffServ.

3 *O Sistema EuQoS*

Este capítulo cuida de descrever em um nível de arquitetura o sistema EuQoS. Acreditamos ser importante citar aqui a motivação dos investimentos no desenvolvimento do sistema:

Concernente ao modelo de negócio, este se baseia na motivação de que os provedores de serviço e de rede aumentem suas receitas pela provisão de serviços com níveis adicionais de QoS, além do nível atualmente oferecido pela Internet atual. (ENRÍQUEZ; ANDRÉS, 2005, p. 11)

Com base nos tópicos discutidos no capítulo anterior fica claro que o problema de oferecer Qualidade de Serviço na Internet permanece. Enquanto a arquitetura IntServ se mostrou não ser escalável, a arquitetura Diffserv define somente como cada rede no caminho tratará determinado fluxo, não oferecendo assim uma resposta ao problema de garantia de Qualidade de Serviço fim-a-fim.

O projeto EuQoS foi realizado por um consórcio de centros de pesquisa de provedores de serviços e universidades da União Européia. Este projeto estava inserido dentro de um fundo de fomento a pesquisas europeu na áreas de Qualidade de Serviço na Internet. O objetivo era projetar e implementar um sistema de Garantia de QoS, o sistema EuQoS, que seria composto por tecnologias já existentes e pelo resultado do desenvolvimento de novas tecnologias que unidas resolvessem o problema global de garantir Qualidade de Serviço fim-a-fim sobre redes heterogêneas.

Uma decisão inicial importante foi que o Sistema EuQoS não teria como alvo implementar Garantia de QoS para toda a Internet, mas somente em um conjunto domínios que a suportassem e para um número usuários registrados no sistema. Isto, contanto que se conseguisse desenvolver uma arquitetura que fosse representativa, simples e escalável. Esta decisão também se baseou na premissa de que nem todas as aplicações necessitam de garantia de QoS, mas que o número dessas aplicações é pequeno em relação ao todo e crescerá suavemente.

3.1 O cenário típico

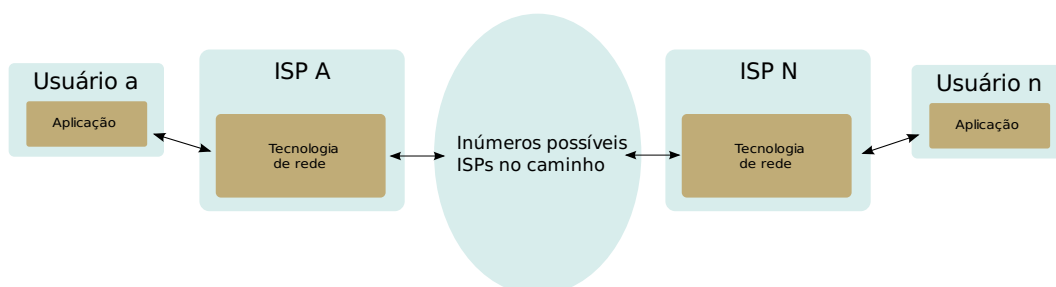


Figura 3.1: Senário típico de utilização do sistema EuQoS

Para que se possa ter uma compreensão clara da descrição da arquitetura que faremos neste capítulo vamos definir aqui um cenário típico da utilização do Sistema EuQoS e definir a terminologia empregada.

Um *usuário* do Sistema EuQoS é um *cliente* de um provedor de serviços que paga para utilizar um serviço de garantia de qualidade. O Sistema EuQoS é orientado ao usuário, o que quer dizer que toda seção é iniciada por um *usuário a* em um domínio *A* tendo em vista obter uma seção com garantia de qualidade entre ele e um *usuário n* em um domínio *N* (ou possivelmente com um *usuário a'* no mesmo domínio *A*). Note que entre o domínio do *usuário a* e o domínio do *usuário n* podem existir alguns outros domínios. Como um usuário aqui é sempre um cliente de um provedor de serviços que implementa o sistema EuQoS, utilizamos aqui os termos *usuário* e *cliente* intercâmbialmente.

Uma *seção* ou *conexão* representam aqui um estado da rede entre os dois usuários em que foi configurada uma garantia da qualidade de serviço no caminho entre eles. Este conceito não se associa diretamente ao tipo de aplicação utilizada pelos usuários, estas podem ser das mais variadas (como aplicativos multimídia ou sensores remotos). A sessão de QoS é iniciada antes de iniciar o tráfego de dados.

Um *domínio EuQoS* é uma rede de acesso que implementa suporte ao Sistema EuQoS. No caso comum é um *Autonomous System*¹ (AS) que implementa suporte ao Sistema. Como do ponto de vista do usuário isto se traduz em um provedor de acesso a Internet (*ISP - Internet Service Provider*) utilizaremos, dependendo da ocasião, os termos *domínio EuQoS*, *AS* e *ISP* intercâmbialmente.

¹Um Autonomous System é uma rede sob administração de uma única entidade, como a rede de um provedor de serviços

3.2 Visão geral da solução proposta pelo Sistema EuQoS

A diferença principal entre a proposta do projeto EuQoS e as arquiteturas propostas anteriormente está no empenho realizado na definição de uma arquitetura global que quebrassem a complexidade do problema maior entre diferentes subsistemas.

Entre as regras fundamentais do projeto (ENRÍQUEZ; ANDRÉS, 2005, p. 19) estava o entendimento de que a infraestrutura de rede da Internet já está bem consolidada (ex. xDSL, UMTS, ATM) e que por isso o sistema precisaria utilizar as ferramentas que cada uma dessas tecnologias já implementam para obter QoS.

A estratégia utilizada pelo sistema EuQoS segue o conceito de estabelecer na rede um número de classes de serviço de QoS (CoS), cada uma orientada para tratar diferentes tipos de tráfego IP com diferentes objetivos de QoS.

O objetivo do sistema EuQoS é então estabelecer um número de classes de serviço de rede em todo o caminho entre o *usuário a* e o *usuário b*. Essas classes de serviço de rede são visíveis pelas aplicações dos usuários e mantidas através de múltiplos domínios de rede, ainda que estas implementem diferentes tecnologias de rede - lembrando que o objetivo principal do sistema é permitir que usuários em diferentes redes de acesso como xDSL, UMTS, WLAN, LAN/Ethernet, possam estabelecer uma conexão com garantia de QoS. É importante então que as classes de serviço definidas no sistema satisfaçam as expectativas de QoS das aplicações dos usuários e possam ser implementadas em tecnologias de rede.

Diferentes tecnologias implementam diferentes mecanismos para prover as classes de serviços definidas. Esses mecanismos são responsáveis por garantir as expectativas de QoS; o que quer dizer que a rede deve oferecer aos fluxos de pacotes as características (como atraso e perda), determinadas para a classe de serviço para o qual foram submetidos. Entre esses mecanismos estão os descritos na seção 2.5, como escalonadores de pacotes, policiadores e classificadores instalados nos equipamentos.

A maneira como o sistema EuQoS propõe resolver o problema da garantia de QoS fim-a-fim é definindo um caminho entre as duas pontas de maneira que a soma dos parâmetros de qualidade oferecidos por cada um dos domínios no caminho satisfaça os requisitos de QoS fim-a-fim (veja a Figura 3.2). Desta maneira, o tráfego do usuário submetido a uma CoS obterá os requisitos de QoS fim-a-fim para esta classe.

Para que haja garantia de qualidade de serviço em cada domínio dois conceitos são aplicados: o dimensionamento (também chamado provisionamento) e controle de admissão

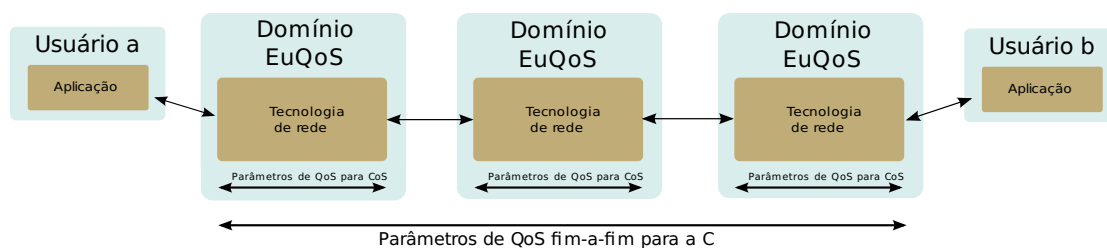


Figura 3.2: Como os parâmetros de QoS fim-a-fim são formados a partir dos parâmetros oferecidos por cada domínio

de conexões (descrito na seção 2.4).

Cada domínio de rede que implementa o sistema EuQoS necessita dimensionar, ou provisionar, a quantidade de recursos (como capacidade de link e tamanho de buffers) que reservará para as diferentes classes de serviço.

Com o conhecimento da quantidade de recursos disponível para cada uma das classes de serviço e dos requisitos de recursos de cada uma das conexões, o sistema EuQoS utiliza controle de admissão de conexões para limitar o número de conexões que o domínio aceitará em cada classe. Desta maneira garante o cumprimento dos requisitos de qualidade para as conexões já aceitas.

O sistema EuQoS é a implementação de uma infraestrutura que permite que os domínios se comuniquem com as aplicações dos clientes, com domínios vizinhos e com a tecnologia de rede para configurar sessões com garantia de qualidade sob demanda.

De maneira superficial, configurar uma sessão significa: receber a requisição do cliente para iniciar uma sessão; descobrir um caminho até o destino que permita o cumprimento da qualidade requerida; configurar sua rede para tratar o fluxo de dados da sessão do cliente pela CoS selecionada e requisitar do próximo domínio no caminho que faça o mesmo.

Note que a maneira como cada domínio fará para garantir que o tráfego submetido a uma classe de serviço seja servida com os requisitos mínimos de qualidade é independente do sistema como um todo. Cada domínio aplicará, dependendo da tecnologia de rede que utiliza, técnicas como engenharia de tráfego, policiamento e escalonamento para garantir o cumprimento dos requisitos em sua rede.

3.3 A Arquitetura do Sistema EuQoS

A prática de dividir para conquistar foi muito utilizada no projeto do sistema EuQoS. O sistema está dividido em camadas conceituais e essas camadas tiveram suas funções implementadas por grupos de módulos.

Vamos descrever o sistema a partir de seus três níveis conceituais: a camada de aplicação, a camada de rede virtual e a camada de transferência. A Figura 3.3 permite visualizar esses três níveis.

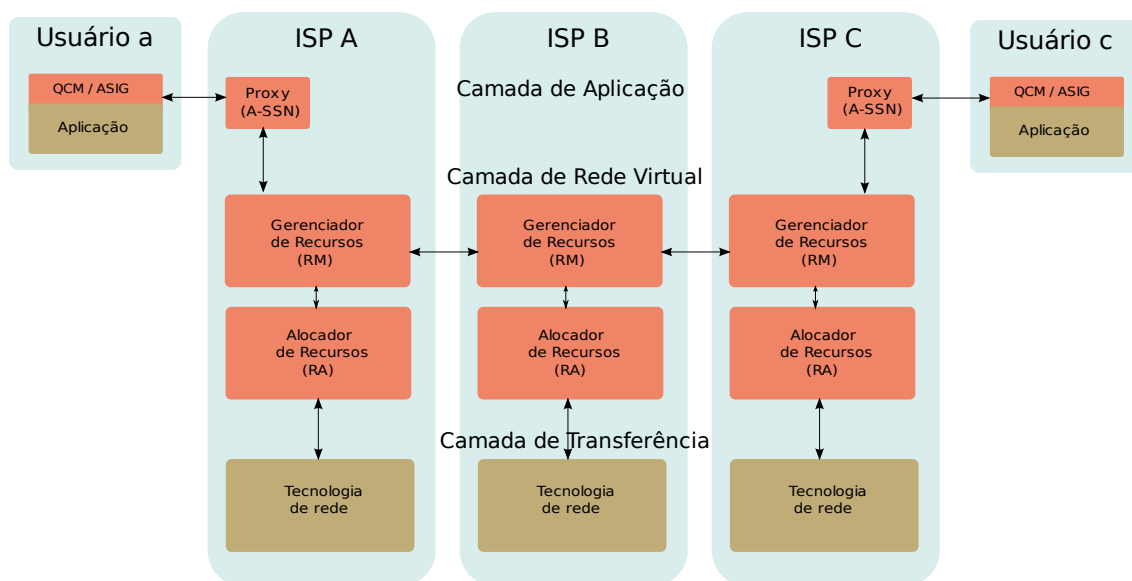


Figura 3.3: Diferentes camadas da arquitetura EuQoS

Vamos agora descrever melhor cada uma dessas camadas.

3.3.1 Camada de Aplicação

É através da camada de aplicação que a aplicação do usuário a , com suporte ao EuQoS, se comunica com o sistema EuQoS de seu provedor de serviços para sinalizar o desejo de iniciar uma nova sessão com QoS. Esta camada propaga esta comunicação até o domínio N e em seguida ao usuário n .

3.3.2 Camada de Rede Virtual

A camada de rede virtual é responsável pela gerência e alocação dos recursos tanto internamente aos ASs como globalmente em todo o caminho do usuário a até o usuário

n. Ela é composta por uma parte independente de tecnologia (*Technology Independent - TI*) controlada por um gerenciador de recursos, o RM (*Resource Manager*), e uma parte dependente de tecnologia (*Technology Dependent*) controlada por um alocador de recursos, o RA (*Resource Allocator - RA*).

O RM gerencia os recursos de rede em um nível mais alto que o RA. Ele abstrai a tecnologia de rede implementada pelo AS e trabalha sobre dados nominais que são quantificações dos recursos que este AS possui. Para efetuar este gerenciamento o RM implementa uma série de funcionalidades, entre elas o controle de admissão de novas conexões (CAC) no domínio, o gerenciamento de SLAs com domínios vizinhos, as decisões de roteamento para a obtenção de um caminho com QoS entre os usuários, e a comunicação com RMs dos domínios vizinhos para efetuar solicitações de QoS. Quando o RM toma uma decisão sobre efetuar uma reserva de recurso em seu domínio, ele se comunica com o RA para que este aplique as configurações nos dispositivos de rede específicos. As decisões tomadas pelo RM se baseiam nos valores nominais dos recursos da rede, nos SLAs estabelecidos e no número de conexões já aceitas.

O RA recebe requisições do RM e possui uma parte dependente de tecnologia que é responsável por implementar nos equipamentos específicos medidas necessárias para oferecer o QoS requerido à sessão. Tanto a reserva como a liberação de recursos nos equipamentos (como banda e buffer) são realizados pelo RA. Dessa maneira dissocia-se as decisões de rede das decisões de tecnologia.

3.3.3 Camada de Transferência

A camada de transferência é por onde os dados das aplicações sujeitos a QoS trafegam. É composta pelas tecnologias de rede implementadas pelos domínios.

3.3.4 Principais Módulos

A seguir descrevemos os principais módulos que compõem a arquitetura do sistema EuQoS agrupados de acordo com a camada em que estão localizados.

3.3.4.1 Camada de Aplicação

A camada de aplicação se distribui entre módulos nas aplicações dos usuários e no sistema EuQoS instalados nos ASs. A Figura 3.4 exhibe como esses módulos se distribuem.

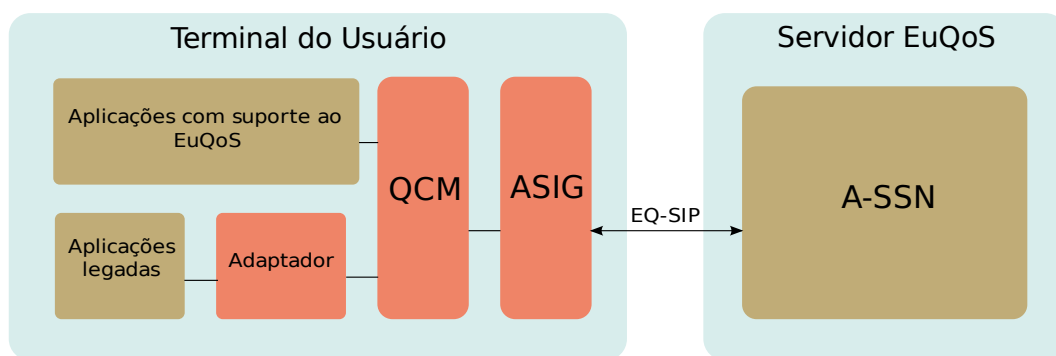


Figura 3.4: Diagrama dos módulos da Camada de Aplicação da arquitetura EuQoS

Do lado das aplicações dos usuários temos o **QCM** (*Quality Control Module*) e o **ASIG** (*Application Signaling*), estes podem ser implementados na forma de bibliotecas utilizadas pelas aplicações que desejam suportar o sistema EuQoS. O módulo QCM fornece uma interface comum para o sistema EuQoS, contendo nele os meios de gerencia de sessões com QoS, acesso a informações da conta do usuário e tratamento de eventos recebidos. Este módulo também realiza um mapeamento do tipo de aplicação (ex. um codec de vídeo) para requisitos de QoS.

Para a comunicação entre a aplicação do usuário e o sistema EuQoS instalado no AS, o projeto propôs uma extensão do protocolo de controle de sessões SIP que suportasse a definição de requisitos de QoS. A esta extensão foi dado o nome de EQ-SIP. O módulo ASIG implementa este protocolo e permite que a aplicação do usuário se comunique com um Proxy EuQoS em seu provedor de serviço para requerer o início de uma sessão.

Do lado do ISP, estão os módulos A-SSN (*Application-level SSN*), CHAR (*Charging*), SAAA (*Security, Authentication, Authorization and Accounting*).

O módulo **A-SSN** (*Application-level Signalling and Service Negotiation*) provê suporte para a comunicação do sistema com o módulo ASIG dos clientes. A sinalização no nível de aplicação é responsável pelo estabelecimento, manutenção e finalização das seções de QoS com os níveis especificados pelos usuários. Este módulo interage com os seguintes outros módulos no sistema EuQoS:

- ASIG, para que possa trocar com as aplicações dos usuários mensagens de estabelecimento de seções através do protocolo EQ-SIP;
- SAAA, no momento do registro e deregistro de usuários para efetuar autenticação e durante o processo de início e fim de seções para repassar informações de contabi-

lidade;

- A-SSN de domínios vizinhos, para repassar mensagens EQ-SIP de estabelecimento de seção entre os módulos ASIG das aplicações dos usuários;
- CallController, para que este inicie o processo de reserva de recursos após o acordo das características da seção no nível de aplicação.

O módulo **CHAR** é responsável pela gerência da cobrança dos usuários do sistema EuQoS e pela geração de suas faturas baseadas na sua utilização da rede. Este módulo possui nele as políticas de cobrança e contratos dos clientes que regem a tarifação das sessões efetuadas pelo cliente. Critérios como tempo de seção e nível de qualidade requerida são utilizados.

O módulo **SAAA** é responsável por controlar o acesso dos recursos da rede aos usuários (Autenticação), a verificar e autorizar o nível de QoS requisitado pelo usuário (Autorização) e por coletar dados das seções para contabilidade.

O módulo SAAA possui interfaces com os seguintes outros módulos:

- Com o módulo A-SSN, para que possa receber e responder pedidos de autenticação de usuários e autorização de suas seções, e também para receber informações de contabilidade;
- com o módulo CHAR para que possa trocar informações de contabilidade;
- com o módulo TERO, descrito na seção 3.3.5, para fornecer estatísticas dos usuários.

3.3.5 Camada de rede virtual - Independente de Tecnologia

Esta seção descreverá os principais módulos que compõem a camada de rede virtual da arquitetura do sistema EuQoS. A Figura 3.5 exhibe como esses módulos se relacionam dentro de um domínio EuQoS.

O módulo **RM-SSN** (*Resource Manager SSN*) provê suporte para a reserva e gerenciamento de recursos no plano de transferência entre os diversos domínios EuQoS no caminho. Esta função realiza a comunicação, nó a nó, entre os RMs dos domínios adjacentes utilizando um protocolo do projeto NSIS² para que ao final do processo de

²*Next Step in Signaling* é um conjunto de protocolos de sinalização proposta pela IETF. Este tipo de sinalização em nível de rede tem como objetivo a instalação, a manutenção e a remoção de estados de

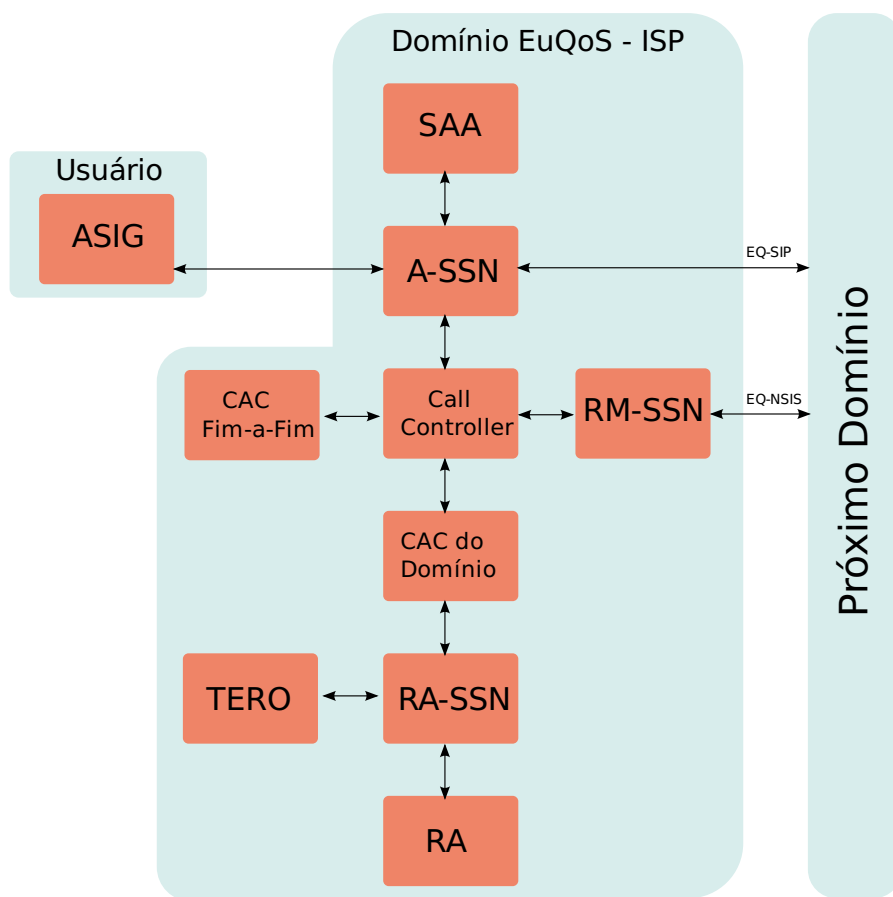


Figura 3.5: Principais módulos da arquitetura EuQoS

estabelecimento de uma seção a reserva de recursos tenha sido instalada em todos os domínios no caminho. Este módulo interage com os seguintes outros módulos do sistema EuQoS:

- CallController. O CallController se comunicará com o RM-SSN para que seja iniciado o processo de reserva de recursos nó a nó para a nova conexão. Isto é realizado após o CallController efetuar o controle de admissão e reserva de recursos no domínio local;
- O sub-módulo RM-SSN do domínio vizinho, para que a reserva de recursos seja instalada ao longo do caminho.

O módulo **CAC** (*Connection Admission Control*) é responsável por limitar o número de conexões a um nível em que a rede possa manter o QoS necessário para as controle em elementos de rede (FU et al., 2005). No sistema EuQoS esta sinalização percorre todos os RMs comunicando um pedido de reserva de recursos para uma sessão que está iniciando. Posteriormente comunica também o fim da sessão para que os recursos sejam desalocados.

conexões já estabelecidas, como vimos na seção 2.4.

Como o sistema EuQoS ataca o problema da garantia de qualidade de serviço fim-a-fim sobre redes heterogêneas, a função CAC assume um papel triplo:

- CAC fim-a-fim (independente de tecnologia de rede): verifica se um caminho inter-domínio fim-a-fim, que ofereça o QoS requerido, pode ser estabelecido;
- CAC do domínio (dependente de tecnologia): verifica os recursos de seu domínio a fim de comunicar ao CAC fim-a-fim para que este calcule o QoS total que pode ser oferecido no caminho.
- CAC da rede de suporte - *Undelying-Network CAC* (dependente de tecnologia): é responsável por aplicar a reserva na tecnologia específica do seu domínio para que este possa atender aos requisitos de QoS.

Um quarto módulo existe e é responsável por coordenar todos os outros sub-módulos CAC e também o módulo RM-SSN. Este último é denominado **CallController**.

O módulo **MMFM** (*Monitoring, Measurement and Fault Management*) é responsável pelas seguintes sub-funções do sistema EuQoS:

- Monitorar o estado da rede naquele momento para dar suporte as decisões do RM de aceitar ou rejeitar novos pedidos de seções de QoS;
- prover informações que auxiliam a função TERO;
- prover suporte a detecção de falhas através do monitoramento dos parametros dos elementos de rede;
- monitora o nível de QoS sendo oferecido para determinar se os requisitos das SLAs estão sendo satisfeitos.

Este módulo oferece uma interface web para o administrador de rede e possui uma interface com o módulo CAC.

O módulo **TERO** (*Traffic Engineering and Resource Optimization*) tem dois objetivos principais. O primeiro é o provisionamento de recursos. Isto significa distribuir os recursos de rede disponíveis entre as várias classes de serviço que se deseja oferecer. A demanda por cada uma das classes e os recursos disponíveis são parâmetros para o desempenho desta função. Os recursos sendo provisionados podem ser intra-domínio ou

inter-domínio. O provisionamento inter-domínio (TERO Independente de Tecnologia de Rede) parte da troca de especificações de nível de serviço (SLSs) entre domínios adjacentes. Posteriormente o TERO necessita também provisionar os recursos da rede intra-domínio de maneira a atender essas SLSs (configurando os roteadores de borda da rede para executarem, por exemplo, policiamento de tráfego e escalonamento) e auxiliar na análise de tráfego e nas previsões dando suporte ao processo de estabelecimento destas SLSs.

O segundo objetivo deste módulo é exercer Engenharia de Tráfego. No sistema EuQoS isto diz respeito a controlar o processor de determinação das rotas entre domínios, auxiliando o processo de decisão do EQ-BGP, descrito na seção 3.3.6, de maneira que os melhores caminhos fim-a-fim para cada uma das classes de serviço sejam determinadas. Para isto o TERO coleta e analisa informações do histórico das sessões realizadas, informações requisitadas ao módulo MMFM de medidas de tráfego nas interfaces de saída dos roteadores de borda do domínio (como por exemplo, ocupação de buffer e utilização do link) e informações do módulo SAA sobre o número de assinaturas de clientes para cada uma das classes de serviço, e fornece uma interface web para que o administrador da rede tome decisões baseadas no resultado das análises.

O módulo **RA-SSN** (*Resource Allocation SSN*) provê o suporte para a reserva e gerenciamento de recursos locais. Este sub-módulo interage com os seguintes outros módulos do sistema EuQoS:

- TERO, utilizado no processo de provisionamento, para a definição das CoSs suportadas e seus recursos;
- CAC do domínio. Após o módulo CAC decidir sobre a admissão da conexão, ele se comunica com o RA-SSN para enviar pedidos de reserva de recursos locais e receber suas confirmações;
- O módulo de alocação de recursos, RA, com o propósito de efetuar a instalação da reserva de recursos ou configurar as CoSs suportadas. As interações entre o RA-SSN e o RA utilizam o protocolo COPS-PR(CHAN et al., 2001).

3.3.6 O protocolo EQ-BGP

O EQ-BGP (*EuQoS-QoS Border Protocol*) é uma extensão dos protocolos BGP-4 e Q-BGP. Ele permite a descoberta de um caminho de roteamento inter-domínio levando em conta os parâmetros de QoS oferecidos por cada domínio no caminho para as diferentes

classes de serviço. Os roteadores EQ-BGP estão dispostos nas bordas dos domínios e anunciam a seus roteadores vizinhos informações sobre os endereços destinos alcançáveis por eles acrescentadas de informações sobre as classes de serviço disponíveis e sobre os parâmetros de desempenho que oferecem para cada uma delas (como atraso, variação no atraso e perda). Note que diferentes rotas podem existir entre a fonte e o destino, e o que o roteador EQ-BGP faz é, baseando-se naquelas informações, aplicar critérios para decidir qual a melhor rota para um destino em uma determinada classe de serviço.

4 *Implantação e testes do Sistema EuQoS*

Este capítulo descreve as etapas de instalação da plataforma e os testes realizados. As metodologias de medições também estão descritas aqui juntamente com os resultados obtidos.

4.1 **Primeira implantação e Testes**

Em um primeiro momento começamos a implantação de um domínio EuQoS com o objetivo de integrá-lo a cama de testes do sistema instalado na Europa. A Figura 4.1 exibe em um diagrama como foi realizada a integração. Foram utilizados um computador para realizar um túnel GRE com o LAAS, dois computadores, o *Host A* e o *Host B*, onde o sistema EuQoS foi instalado e um último computador para executar a aplicação cliente do sistema EuQoS.

Como parte dos requisitos para nossa integração a cama de testes precisaríamos realizar uma sequência de testes do link da UFSC até o LAAS. Esta seção descreve os resultados obtidos em diferentes testes realizados entre uma máquina no domínio UFSC até uma máquina no domínio LAAS.

O software utilizado para gerar os tráfegos de teste e realizar as medições foi o MGEN (MEGEN..., 2008) através da interface Netmeter (GRACIÀ et al., 2005). Os relógios de ambos dos computadores de ambas as pontas foram sincronizados utilizando servidores de tempo NTP.

A seguir estão descritos os resultados desses testes iniciais no link entre a UFSC e o LAAS.

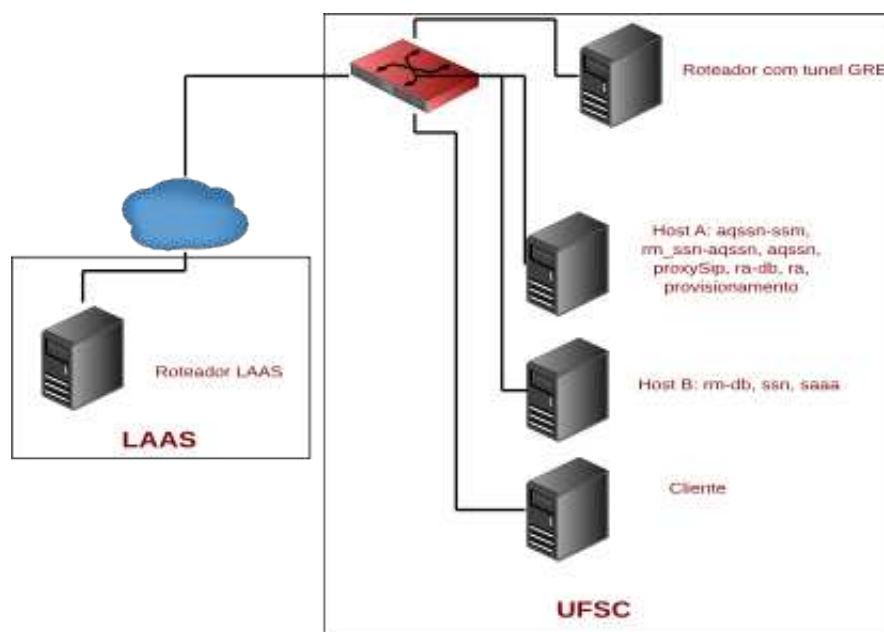


Figura 4.1: Arquitetura da primeira implantação do sistema EuQoS

4.1.1 Primeiro teste UDP

Este primeiro teste teve as seguintes características:

- Taxa de emissão: 96pps
- Tamanho dos pacotes: 1420 bytes (UDP) / 1448 (Total com IP)
- Duração do teste: 10 min (600000 ms)

A Tabela 4.1 exibe os dados de atraso e variação de atraso para cada um dos sentidos. Durante os primeiros 32.92 segundos do teste, os pacotes no sentido UFSC para LAAS foram perdidos. Essa perda se repeliu durante rodadas diferentes deste teste e sua causa não foi detectada. O tráfego no sentido UFSC para LAAS possui um maior atraso médio (28.18% a mais). Uma causa provável é que esta direção faz um caminho diferente ou está mais carregada.

Algo interessante visto neste primeiro teste é que o atraso diminuiu com o tempo de uma maneira quase linear (como pode ser visto na Figura 4.2). Isto indica uma provável variação na diferença entre os relógios (*clock drift*), mesmo com a utilização do NTP. Este problema é agravado em testes em redes locais, como está exposto na seção seguinte e necessita ser tratado.

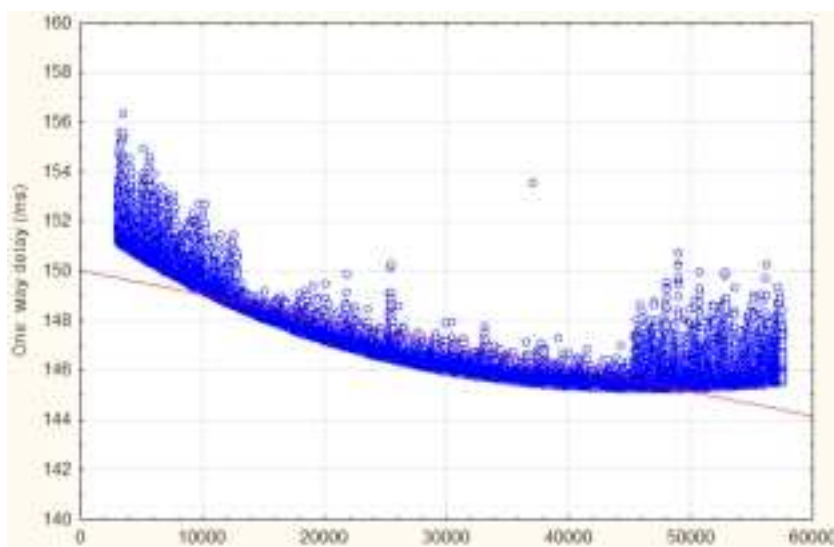


Figura 4.2: Atraso dos pacotes sentido UFSC para LAAS

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	57600	57600
Perda de Pacotes	5.54% (Pacote 0 ao 3159) 32.92 seg	0%
Perda de sequência	0%	0%
Média (ms)	147.06	104.14
Intervalo de confiança (95%) (ms)	[147.09, 146.27]	[104.12, 104.15]
Mediana (ms)	146.27	104.65
Mínimo (ms)	145.22	99.44
Máximo (ms)	348.37	108.39
Variância	14.33	3.65
Desvio Padrão (ms)	3.79	1.91

Tabela 4.1: Primeiro teste UDP - Dados de Atraso.

A Figura 4.3 que mostra o atraso medido no sentido LAAS para UFSC mostra outro fenômeno, a existência de dois padrões diferentes de atraso, supostamente devido a existência de mais de duas rotas diferentes neste caminho.

As Tabelas 4.1, 4.2 e 4.3 mostram, respectivamente, os dados de atraso, variação no atraso (*jitter*) e vazão obtidos neste teste.

4.1.2 Segundo teste UDP

Este segundo teste teve as seguintes características:

- Taxa de emissão: 897pps

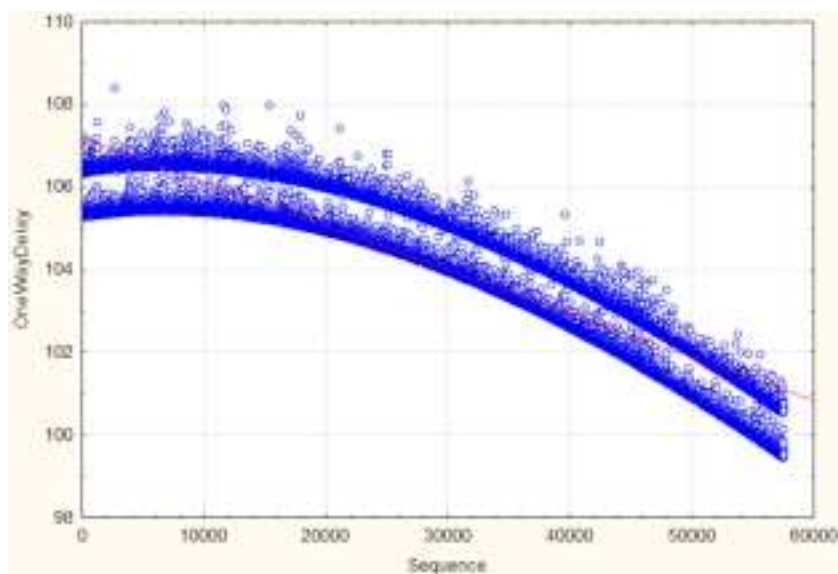


Figura 4.3: Atraso dos pacotes sentido LAAS para UFSC

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	57600	57600
Perda de Pacotes	5.54% (Pacote 0 ao 3159) 32.92 seg	0%
Perda de sequência	0%	0%
Média (ms)	0.31	1.06
Intervalo de confiança (95%)	[0.274, 0.339]	[1.056, 1.059]
Mediana (ms)	0.12	1.06
Mínimo (ms)	0.0	0.0
Máximo (ms)	201.88	3.01
Variância	15.22	0.02
Desvio Padrão (ms)	3.90	0.14

Tabela 4.2: Primeiro teste UDP - Dados de Variação no Atraso (*Jitter*).

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	54439	57600
Perda de Pacotes	5.54% (Pacote 0 ao 3159) 32.92 seg	0%
Perda de sequência	0%	0%
Média (bps)	1090300	109390
Intervalo de confiança (95%)	[1089688, 1090911]	[1089977, 1090902]
Mediana (bps)	1090560	1090560
Mínimo (bps)	931520	976960
Máximo (bps)	1101920	1101920
Variância	54880723	26470249
Desvio Padrão (bps)	7408.15	5144.92

Tabela 4.3: Primeiro teste UDP - Vazão.

- Tamanho dos pacotes: 160 bytes (UDP) / 188 (Total com IP)
- Duração do teste: 10 min (600000 ms)

O aumento no número de pacotes que sofreram perda de sequência no sentido LAAS para UFSC, exibido na Tabela 4.4, em relação ao teste anterior se deve a diminuição significativa do tempo entre envio de pacotes e tende a confirmar a hipótese levantada anteriormente da existência de duas rotas diferentes neste sentido.

As Tabelas 4.5 e 4.6 exibem respectivamente os dados relativos a variação do atraso e vazão durante o teste.

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	538200	538200
Perda de Pacotes	5.79% (Pacote 0 ao 29456)	0%
Perda de sequência	3.24%	20.62%
Média (ms)	139.60	107.69
Intervalo de confiança (95%)	[139.625, 139.628]	[107.821, 107.832]
Mediana (ms)	139.60	107.69
Mínimo (ms)	138.16	105.42
Máximo (ms)	145.85	259.60
Variância	0.49	3.64
Desvio Padrão (ms)	0.70	1.91

Tabela 4.4: Segundo teste UDP - Atraso em um sentido.

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	538200	538201
Perda de Pacotes	5.79% (Pacote 0 ao 29456)	0%
Perda de sequência	3.24%	20.62%
Média (ms)	0.082	0.201
Intervalo de confiança (95%)	[0.081, 0.082]	[0.200, 0.202]
Mediana (ms)	0.008	0.021
Mínimo (ms)	0.0	0.0
Máximo (ms)	4.225	153.237
Variância	0.031	0.193
Desvio Padrão (ms)	0.176	0.439

Tabela 4.5: Segundo teste UDP - Variação no atraso (*Jitter*).

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	538200	538201
Perda de Pacotes	5.79% (Pacote 0 ao 29456)	0%
Perda de sequência	3.24%	20.62%
Média (bps)	1147880	1147945
Intervalo de confiança (95%)	[1147862, 1147899]	[1147931, 1147959]
Mediana (bps)	1148160	1148160
Mínimo (bps)	992000	1022720
Máximo (bps)	1153280	1153280
Variância	45385654	27423365
Desvio Padrão (bps)	6736,888	5236.732

Tabela 4.6: Segundo teste UDP - Vazão.

4.1.3 VoIP

Neste teste o desempenho da rede em ambos os sentidos foi avaliado para pacotes e vazão relativos aos valores médios de uma sessão VoIP utilizando um codec comum. As características foram as seguintes:

- Taxa de emissão: 20pps
- Tamanho dos pacotes: 60 bytes (UDP) / 88 (Total com IP)
- Duração do teste: 10 min (600000 ms)

As Tabelas 4.7, 4.8 e 4.9 exibem respectivamente os dados resultantes do testes relativos a respectivamente atraso em um sentido, variação no atraso e vazão.

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	12000	12000
Perda de Pacotes	5.47% (Pacote 0 ao 657)	0%
Perda de sequência	0%	0%
Média (ms)	145.40	112.13
Intervalo de confiança (95%)	[145.40, 145.41]	[112.12, 112.15]
Mediana (ms)	145.32	112.05
Mínimo (ms)	144.96	110.49
Máximo (ms)	150.51	114.52
Variância	0.13	0.49
Desvio Padrão (ms)	0.36	0.70

Tabela 4.7: Teste pacotes VoIP - Atraso em um sentido.

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	12000	12000
Perda de Pacotes	5.47% (Pacote 0 ao 657)	0%
Perda de sequência	0%	0%
Média (ms)	0.26	1.04
Intervalo de confiança (95%)	[0.25, 0.26]	[1.04, 1.04]
Mediana (ms)	0.09	1.06
Mínimo (ms)	0.001	0.0
Máximo (ms)	5.08	2.8
Variância	0.16	0.03
Desvio Padrão (ms)	0.40	0.18

Tabela 4.8: Teste pacotes VoIP - Variação no atraso (*Jitter*).

Direção	UFSC -> LAAS	LAAS -> UFSC
Número de pacotes	12000	12000
Perda de Pacotes	5.47% (Pacote 0 ao 657)	0%
Perda de sequência	0%	0%
Média (bps)	9598.31	9598.40
Intervalo de confiança (95%)	[9594.23, 9602.38]	[9595.26, 9601.54]
Mediana (bps)	9600.00	9600.00
Mínimo (bps)	8640.00	8640.00
Máximo (bps)	10080.00	9600.00
Variância	2439.53	1536.00
Desvio Padrão (bps)	49.39	39.19

Tabela 4.9: Teste pacotes VoIP - Vazão.

4.1.4 Análise dos resultados

Os testes de desempenho da rede UFSC-LAAS executados durante esta primeira implantação serviram para nos introduzir às técnicas de medições em rede e a problemas como a variação na diferença entre relógios para medição de atraso em único sentido.

Durante esta primeira implantação não conseguimos vencer os obstáculos impostos pela complexidade do Sistema EuQoS para executar com sucesso um domínio na UFSC integrado a rede de testes na Europa. No entanto em uma segunda etapa de implantação descrita a seguir o objetivo de executar com sucesso dois domínios EuQoS locais foi satisfeito.

4.2 Segunda Implantação e teste

Com o término do programa de desenvolvimento do Sistema EuQoS no início de 2008, encerramos nosso trabalho de tentativa de integração de um domínio EuQoS na UFSC com a rede de testes na Europa e passamos para uma segunda fase de implantação na qual o objetivo era então instalar dois domínios em uma rede local para testes.

O processo de instalação foi demorado e trabalhoso devido a complexidade do sistema. Um dos aspectos desta complexidade é o número de softwares que necessitam ser instalados e configurados devidamente e como o projeto foi desenvolvido de uma maneira distribuída entre os vários integrantes na Europa, a documentação também é muito segmentada. Todo este processo de instalação não será exposto aqui pois se trata de instalação e configuração de parâmetros dos diversos programas o que foge do escopo deste trabalho.

A Figura 4.4 exibe o diagrama de como foram dispostas as máquinas em uma rede local para formar dois domínios EuQoS.

4.2.1 O problema da sincronização de relógios

Com a configuração exibida da Figura 4.4 onde ambos os domínios estão conectados em um mesmo switch Gigabit e com todas as interfaces de rede dos computadores sendo também Gigabits, o tempo de atraso entre as redes é demasiadamente pequeno e a metodologia de medição se tornou crítica. Após configurar as placas de rede envolvidas no caminho dos dados entre os dois domínios para 10Mbit o tempo aumentou um pouco com primeiros testes de atraso de sentido único, do *Cliente a* ao *Cliente b*, exibindo valores da

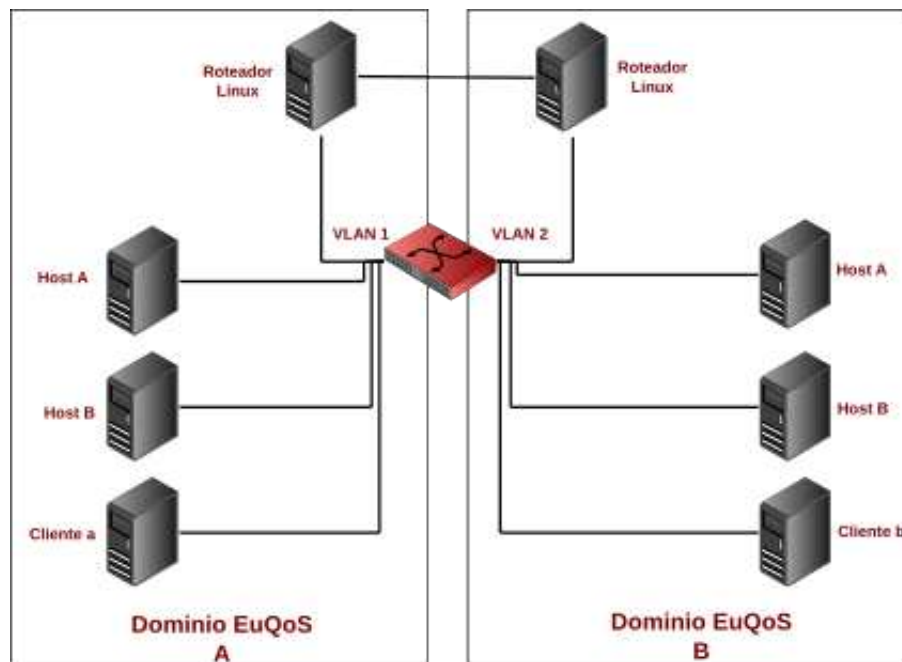


Figura 4.4: Diagrama da segunda implantação do sistema.

ordem de poucos milissegundos (menos de uma dezena).

Com quantias de tempo desta ordem, o problema da variação da diferença entre os relógios fonte-destino, como observados na Figura 4.2 necessitava ser atacado.

A principal fonte de variação entre a frequência de relógios em computadores diferentes é o fato de que a frequência dos osciladores de cristal utilizados pelos computadores comuns variam de acordo com diferenças construtivas e também com a temperatura. Uma solução imaginada, e experimentada aqui, foi a utilização de um ambiente de virtualização onde ambos *Cliente a* e *Cliente b* pudessem ser executados sobre o mesmo hardware.

A solução de virtualização precisaria ser testada já que a infra-estrutura de virtualização acrescenta latências no sistema. Precisaríamos saber se conseguiríamos contornar o problema dos relógios e se, mesmo sobre um sistema de virtualização, conseguiríamos obter uma boa precisão nas medições.

Conforme proposto em (PEUHKURI; GROHN; SIMOLA, 2007), um possível método para a avaliação do sincronismo entre relógios de dois computadores é fazer com que ambos capturem pacotes simultaneamente em um mesmo ponto da rede e então analisar a diferença entre as estampas de tempo atribuídas por ambos.

O teste foi realizado emitindo pacotes para um endereço broadcast em uma taxa fixa na rede que eram capturados por ambos os clientes.

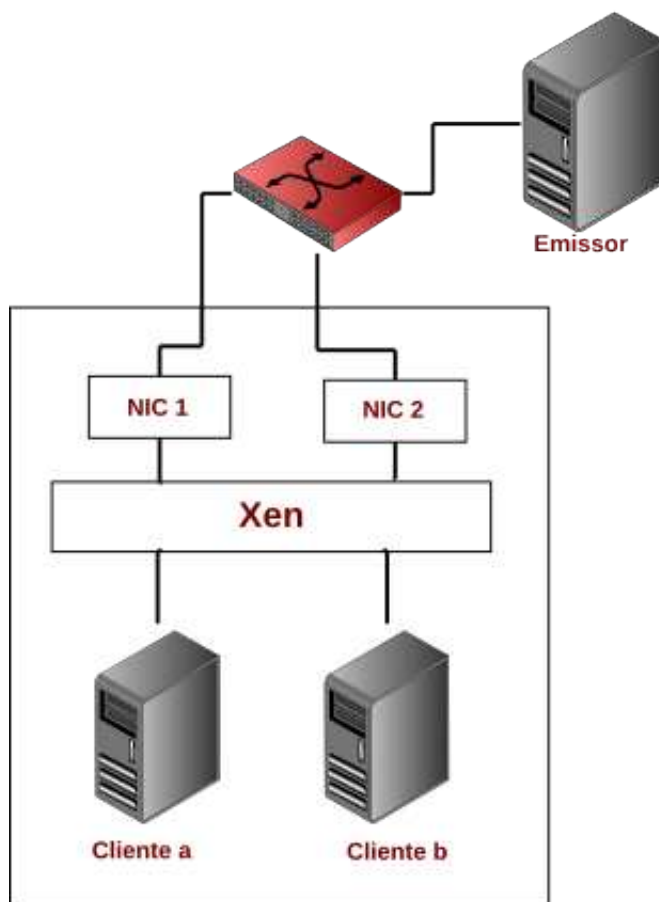


Figura 4.5: Diagrama da rede montada para avaliar a diferença entre os relógios.

A Figura 4.5 mostra como foi montada a rede para os testes.

Foram emitidos 10.000 pacotes de 64 bytes em intervalos fixos em uma taxa de 110 pacotes por segundo. A duração aproximada do teste foi de 90 segundos. A taxa escolhida foi baixa para que não houvesse perda nas placas ou no switch. O tráfego de teste era o único na rede durante sua duração.

A Figura 4.6 e a Tabela 4.10 exibem dados das medições da diferença entre as estampas de tempo atribuídas pelas duas máquinas para cada um dos pacotes.

Diferença Média (μs)	-1000,13
Diferença Mínima (μs)	-1032,00
Diferença Máxima (μs)	-989,00
Desvio Padrão (μs)	1,522

Tabela 4.10: Dados das diferenças entre estampas de tempo de cada pacote.

O resultado do teste foi satisfatório. Neste caso a distância entre os relógios permane-

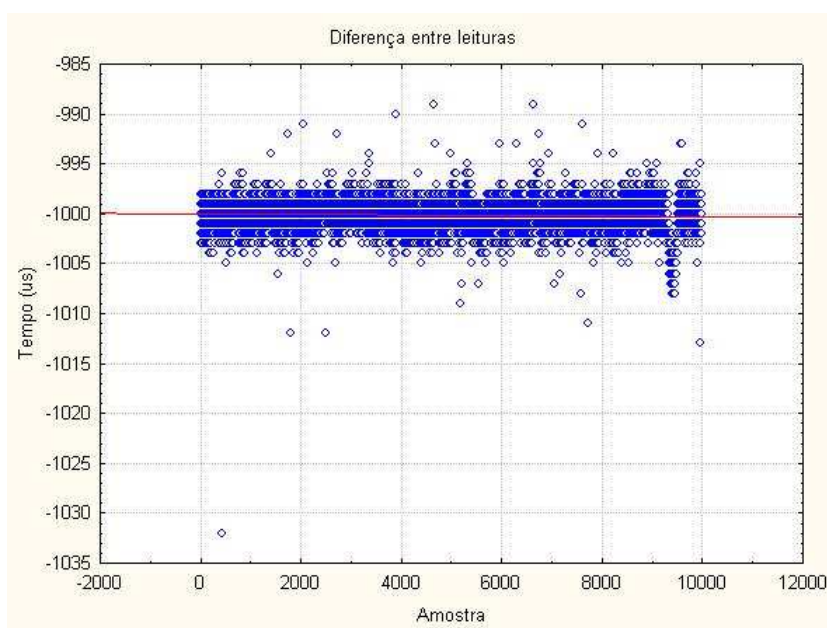


Figura 4.6: Plotagem das diferenças entre estampas de tempo de cada pacote.

ceu muito próximo de 1ms durante todo o teste. O valor do desvio padrão e dos intervalos de confiança mostram que esta metodologia oferece uma precisão da ordem de dezenas de microsegundos. Esta metodologia de testes foi então utilizada em todos os testes descritos a partir daqui neste trabalho.

4.3 Medição do tempo de sinalização do Sistema EuQoS

Com a testbed instalada realizamos uma medição para conhecer o tempo necessário para o estabelecimento de uma sessão entre dois usuários EuQoS. Isto é, a quantia de tempo decorrido desde o pedido de estabelecimento de sessão por um *Usuário a* no *Domínio A* até o recebimento desse pedido pelo *Usuário b* no *Domínio B*.

Este teste foi realizado sem tráfego de fundo na rede. Seu objetivo foi obter uma medida de primeira ordem do tempo de processamento e sinalização para estabelecimento de sessões no sistema EuQoS.

Durante o teste foi utilizada a ferramenta tcpdump para capturar os pacotes de rede nas interfaces de ambos os computadores *Cliente a* e *Cliente b* (veja Figura 4.4) enquanto se iniciava sessões. Os resultados do teste estão dispostos na Tabela 4.11.

Este teste mostra como a testbed instalada pode ser utilizada para avaliações de desempenho do Sistema EuQoS em busca de oportunidades de trabalhos futuros na ot-

Número do teste	Envio	Recebimento	Diferença
1	11:28:26.38	11:28:26.66	00:00.27
2	11:29:06.34	11:29:06.89	00:00.55
3	11:29:45.95	11:29:46.50	00:00.55
4	11:30:34.83	11:30:35.38	00:00.55
5	11:43:07.92	11:43:08.46	00:00.54

Tabela 4.11: Dados dos testes de tempo para estabelecimento de sessão no Sistema EuQoS

mização do sistema.

4.4 O domínio DiffServ

Para completar a implantação dos domínios precisávamos de uma implementação de QoS na camada de rede. Dentro do projeto EuQoS foi também desenvolvido um script que implementa esta camada como um domínio DiffServ em Linux. Utilizamos este script para configurar um domínio DiffServ na saída da interface do roteador do *Domínio A* que conecta ao roteador do *Domínio B* de maneira que todo o tráfego que saísse do *Domínio A* em direção ao *Domínio B* estivesse sujeito ao domínio DiffServ.

Para entendermos como as classes do domínio DiffServ foram dimensionadas, imaginemos o caso em que um Domínio EuQoS possui um link com capacidade de 9Mbit com um segundo Domínio e deseja provisionar esses recursos (o processo de provisionamento é descrito na sessão 3.2) de maneira que as seguintes classes de serviço fossem oferecidas:

Classe 1 - Uma primeira classe para oferecer garantia de QoS para aplicações de tempo real;

Classe 2 - Uma segunda classe para oferecer garantia de QoS para aplicações que não são de tempo real;

Classe 3 - Uma terceira classe para oferecer garantias de QoS menos estritas do que a anterior para aplicações que não são de tempo real;

Restante - Permitir também que clientes utilizem a rede sem garantia de QoS (no modelo *Best Effort*).

De maneira análoga ao exibido na Figura 2.3 do capítulo sobre Qualidade de Serviço, aqui são implementados diferentes PHBs DiffServ para atender cada classe de serviço

utilizando mecanismos como filas FIFO e o escalonador HTB (uma variação do CBQ). A Figura 4.7 exibe como isto foi realizado.

Foram realizados quatro PHBs diferentes: EF, AF1, AF3 e BE. TBFs, subclasses do escalonador HTB, foram configurados para garantir a reserva de recursos de cada um dos PHBs. Um TBF foi utilizado também para limitar o total de tráfego no link a 9Mbit (a rede está configurada para 10Mbit então desta maneira garantimos que os mecanismos aqui implementados controlarão o desempenho da rede e não haverá perda de pacotes pelos dispositivos de rede).

A seguinte correspondência foi utilizada entre as PHBs implementadas e as classes de serviço oferecidas:

- EF para o tráfego da classe 1;
- AF1 para o tráfego da classe 2;
- AF3 para o tráfego da classe 3;
- BE para o tráfego restante.

Para que possamos gerenciar não só o recurso de vazão mas também o de armazenamento filas *FIFO* foram utilizadas, como exibido na Figura 4.7. Todas estas filas foram configuradas com o tamanho de 10.000 pacotes.

4.5 Teste do domínio DiffServ

Executamos uma sequência de testes para avaliar o comportamento da PHB EF implementada no domínio DiffServ. Supomos um caso em que um cliente deseja manter um tronco de ligações VoIP entre dois escritórios de sua empresa. Utilizamos um tráfego de teste aproximando o tráfego gerado por 20 chamadas VoIP simultâneas utilizando um codec comum. Supondo uma média de 50 pacotes por segundo de 200 bytes cada em uma chamada, em 20 chamadas teríamos um tráfego de 1.795Kbs. Portanto nosso tráfego de teste é um tráfego de 1.795Kbps UDP com pacotes de tamanho de 200bytes gerado entre o computador *Cliente a* e o computador *Cliente b*.

Parâmetros como atraso, variação no atraso (*jitter*) e perda de pacotes foram medidos em dois cenários de testes diferentes. Em um primeiro cenário o tráfego foi enviado sem diferenciação, utilizando a classe BE. Em um segundo cenário o tráfego foi submetido a

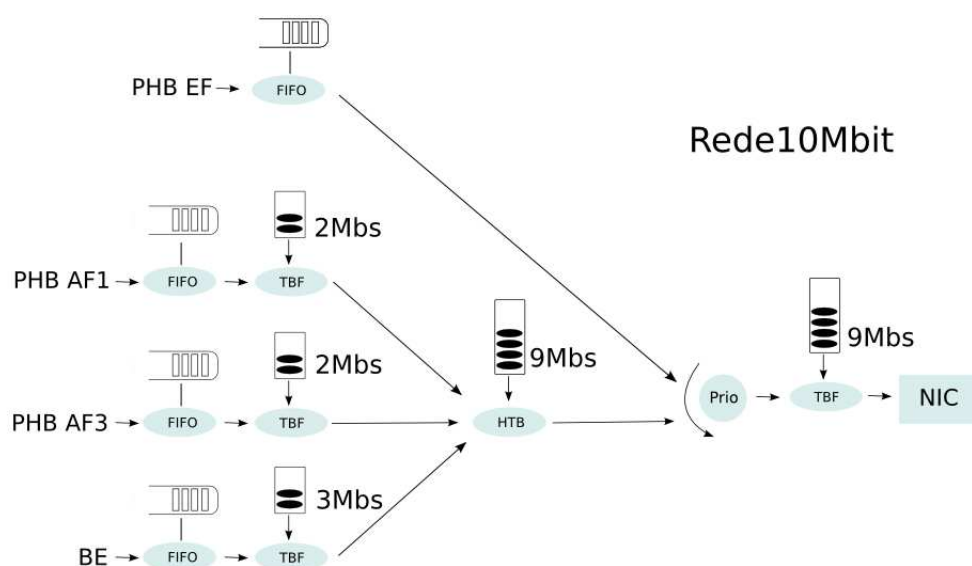


Figura 4.7: Domínio DiffServ realizado no Linux.

PHB EF. Em ambos os cenários os parâmetros foram medidos na presença de diferentes tráfegos de fundo.

Para cada cenário foram executados cinco testes. Em todos eles um tráfego constante de 2Mbps de pacotes de 1500 bytes foi submetido a PHB AF1 e um tráfego constante de 2Mbps de pacotes de 500 bytes foi submetido a PHB AF3. No primeiro teste de cada cenário estes eram os únicos tráfegos de fundo. Para cada um dos testes seguintes um tráfego de respectivamente 1Mbps, 2Mbps, 3Mbps e 4Mbps foi submetido a PHB BE.

As Figuras 4.8 e 4.9 ajudam a compreender os dois cenários de testes.

4.5.1 Cenário 1

A Tabela 4.12 exhibe os resultados obtidos nos diferentes testes neste cenário. Como esperado, já que não há nenhum tratamento do tráfego, os parâmetros degradam rapidamente. A sequência de resultados exibida na Figura 4.10 permite visualizar este processo. Por fim, como visto no item (i) da Figura 4.10 o tempo de atraso começa a aumentar vertiginosamente pela utilização dos buffers implementados que fazem com que os pacotes continuem sendo entregados mas com um atraso cada vez maior. No final vemos que o atraso supera 1 segundo.

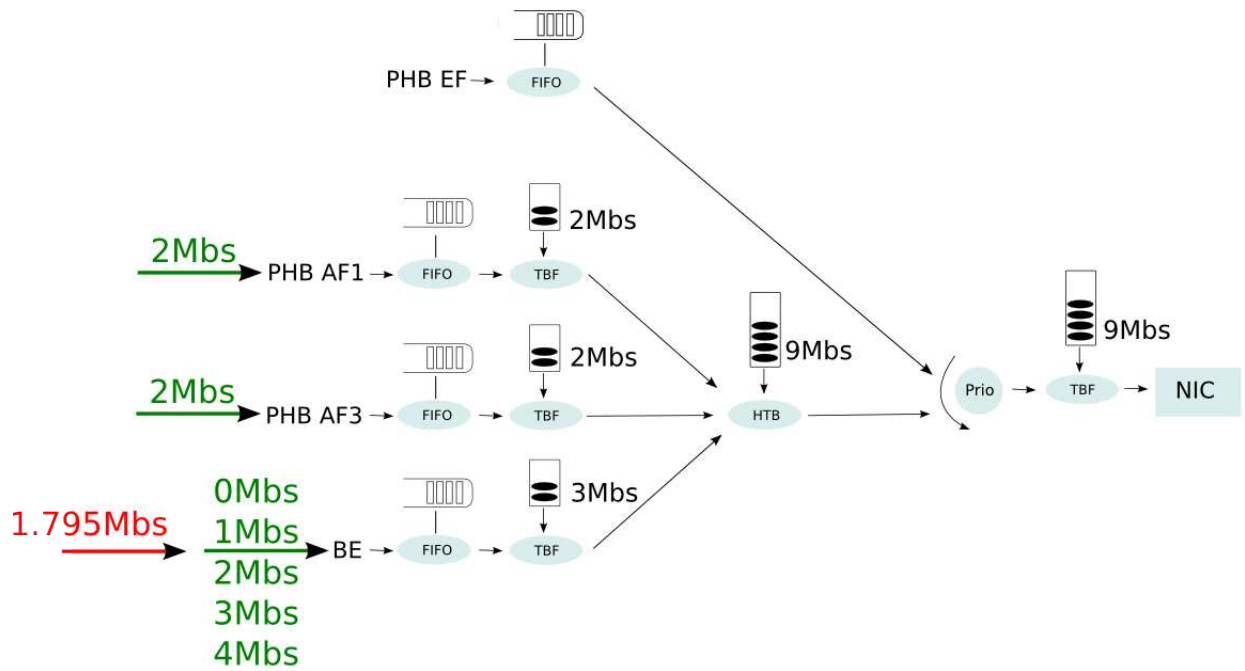


Figura 4.8: Primeiro cenário de testes.

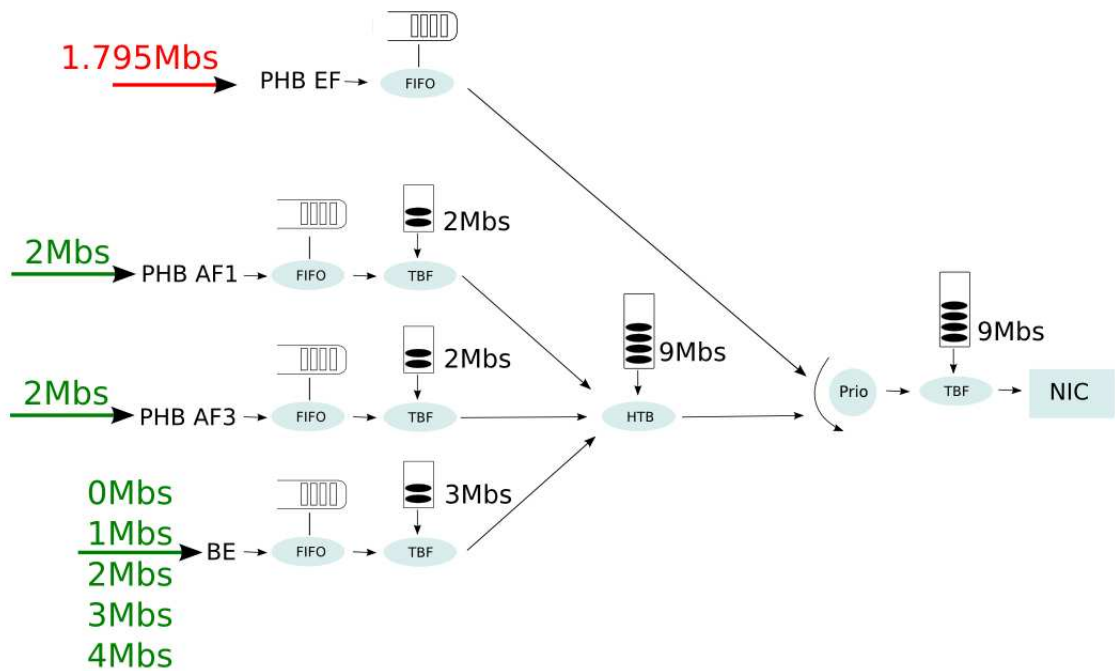


Figura 4.9: Segundo cenário de testes.

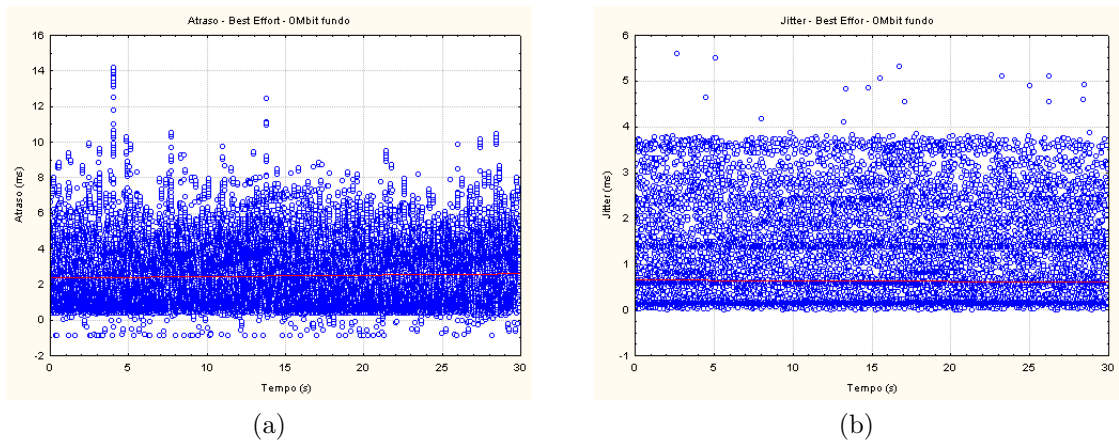


Figura 4.10: Cenário 1: Variação do Atraso e Jitter sem de tráfego de fundo BE.

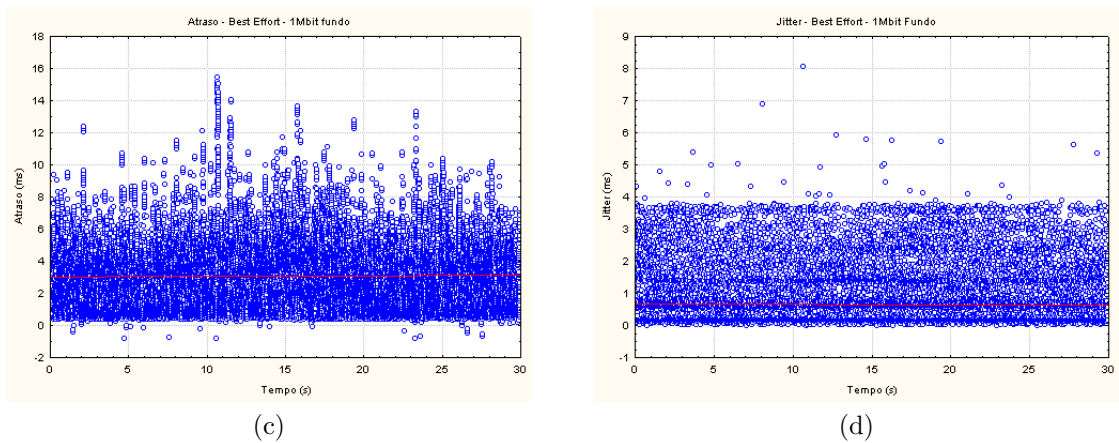


Figura 4.10: Cenário 1: Variação do Atraso e Jitter para 1Mbit de tráfego de fundo BE.

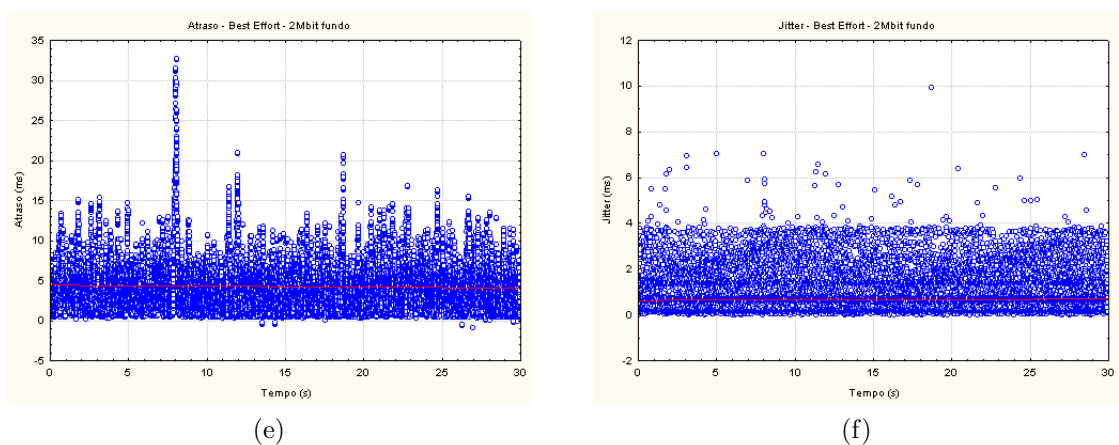


Figura 4.10: Cenário 1: Variação do Atraso e Jitter para 2Mbit de tráfego de fundo BE.

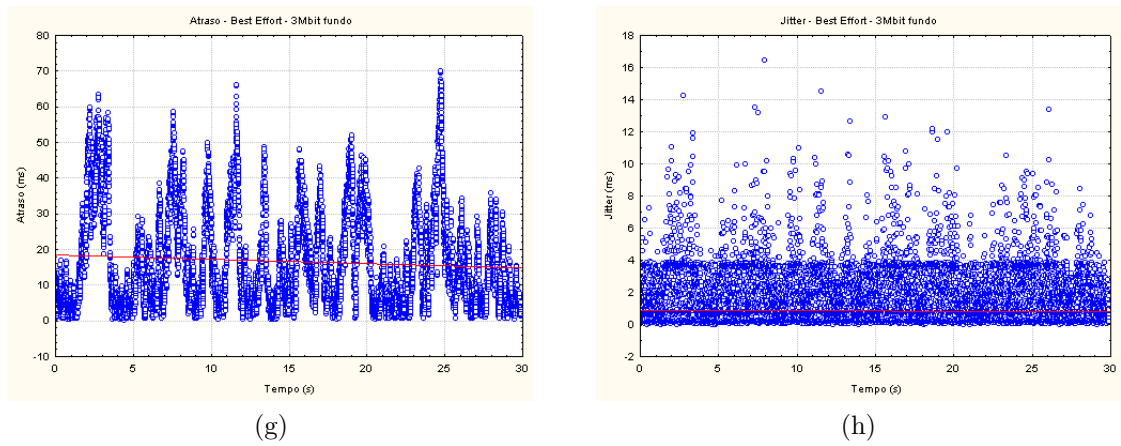


Figura 4.10: Cenário 1: Variação do Atraso e Jitter para 3Mbit de tráfego de fundo BE.

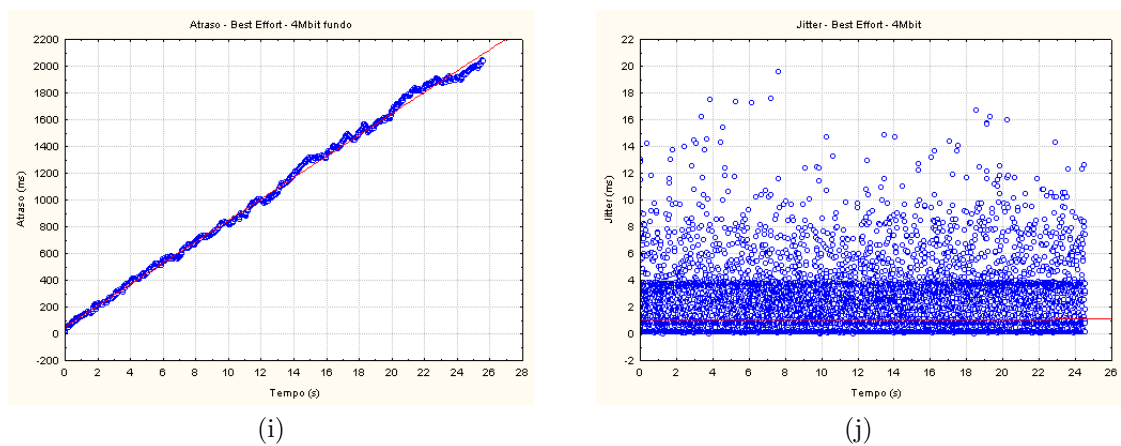


Figura 4.10: Cenário 1: Variação do Atraso e Jitter para 4Mbit de tráfego de fundo BE.

Tráfego de fundo BE	Média do atraso (ms)	Desvio padrão do atraso (ms)	Média do Jitter (ms)	Desvio padrão do Jitter (ms)	Perda
0Mbps	2.482	1.7425	0.630	0.884	0.00%
1Mbps	3.074	2.017	0.6445	0.870	0.00%
2Mbps	4.254	2.895	0.663	0.871	0.04%
3Mbps	16.645	13.595	0.837	1.311	0.30%
4Mbps	1071.703	593.6582	1.054	1.785	24.55%
5Mbps	1415.837	468.1909	0.940	1.590	26.34%

Tabela 4.12: Resultados dos testes no Cenário 1

Tráfego de fundo BE	Média do atraso (ms)	Desvio padrão do atraso (ms)	Média do Jitter (ms)	Desvio padrão do Jitter (ms)	Perda
0Mbps	2.247	1.376	0.521	0.777	0.00%
1Mbps	2.677	1.350	0.489	0.711	0.00%
2Mbps	3.069	1.275	0.447	0.650	0.00%
3Mbps	7.880	3.316	0.465	0.746	0.00%
4Mbps	10.594	2.202	0.490	0.803	1.68%
5Mbps	6.57	5.360	0.408	0.710	5.00%

Tabela 4.13: Resultados dos testes no Cenário 2

4.5.2 Cenário 2

Neste cenário o tráfego é submedito a PHB EF. A Tabela 4.13 e a Figura 4.11 exibem os resultados. Nota-se que neste cenário apesar do comportamento do tráfego variar os parâmetros permanecem dentro de limites baixos.

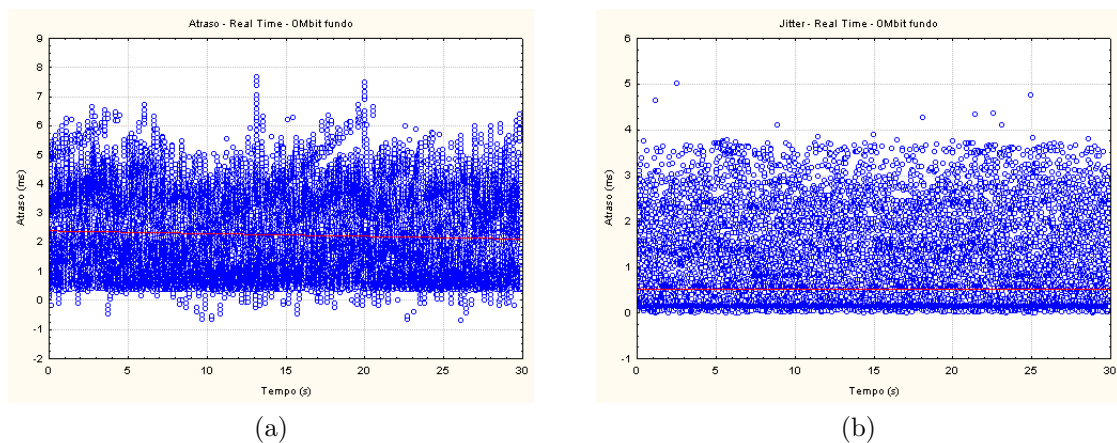
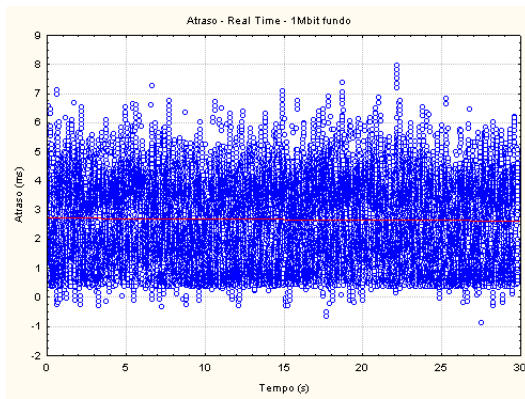
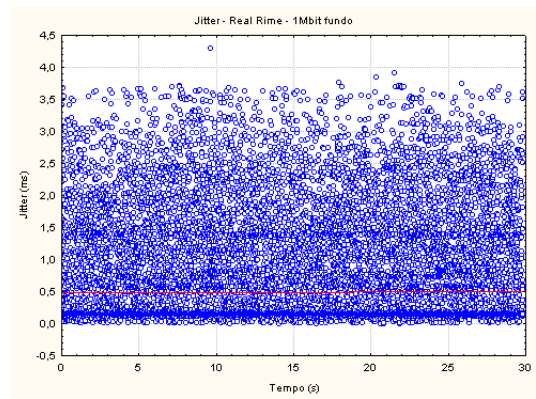


Figura 4.11: Cenário 2: Variação do Atraso e Jitter sem tráfego de fundo BE.

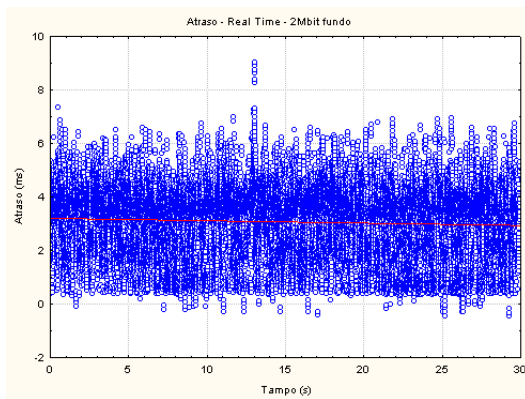


(c)

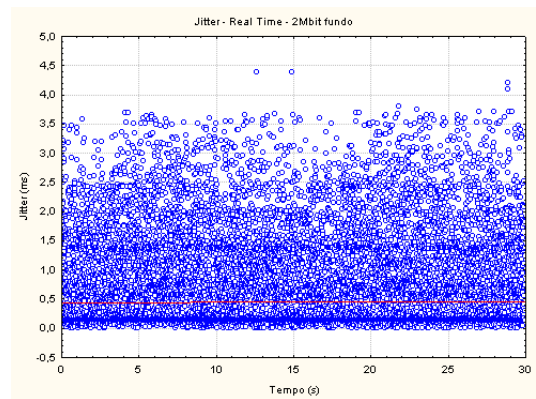


(d)

Figura 4.11: Cenário 2: Variação do Atraso e Jitter para 1Mbit de tráfego de fundo BE.

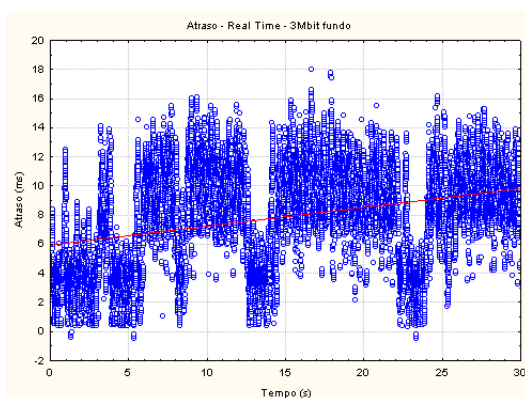


(e)

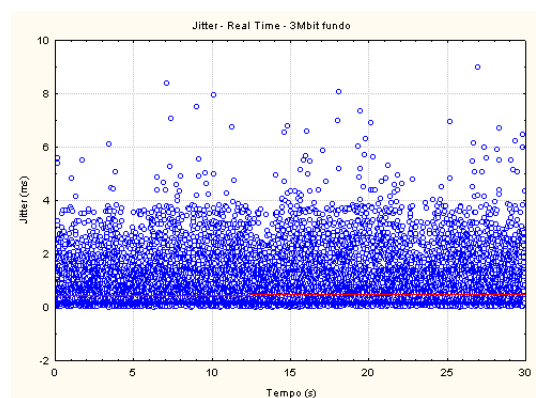


(f)

Figura 4.11: Cenário 2: Variação do Atraso e Jitter para 2Mbit de tráfego de fundo BE.



(g)



(h)

Figura 4.11: Cenário 2: Variação do Atraso e Jitter para 3Mbit de tráfego de fundo BE.

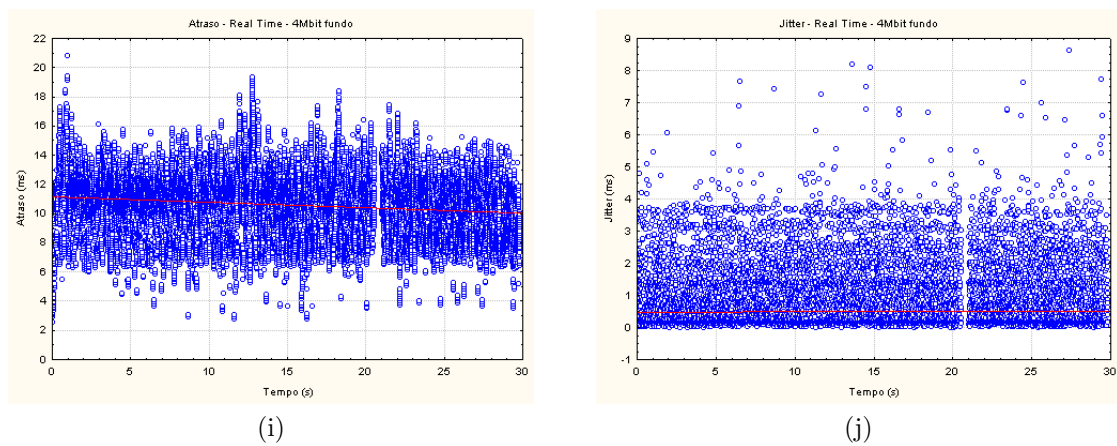


Figura 4.11: Cenário 2: Variação do Atraso e Jitter para 4Mbit de tráfego de fundo BE.

4.5.3 Comentários sobre os testes

Os resultados dos testes aqui realizados não possuem relevância já que a comparação entre os dos cenários é obviamente absurda. Seu objetivo não é outro senão permitir a experimentação da rede de testes e da metodologia de medição descrita na seção 4.2.1.

4.6 Conclusão

Durante a realização deste trabalho muitos conceitos fundamentais relacionados com redes de computadores que antes eram conhecidos superficialmente são hoje mais profundamente conhecidos e dominados. Não bastasse o ganho de conhecimento, o processo de desenvolver um trabalho de maior porte é também em si mesmo um aprendizado valioso. Assim, é certo dizer aqui que a realização deste trabalho é de grande contribuição para a formação do estudante como um profissional da Ciência da Computação.

Compreender a arquitetura e a dinâmica de um sistema do porte do EuQoS, constituído de mais de duas dezenas de softwares interagindo em camadas diferentes e desenvolvido por diversos grupos de trabalho, foi algo trabalhoso e uma compreensão mais madura veio somente com algum tempo de trabalho. Foi possível no entanto atingir o ponto em que é possível compreender a função básica de cada componente no sistema e como eles interagem para oferecer o serviço desejado. Desde ponto já podemos nos propor a estudar algum componente específico sustentados pelo entendimento do todo.

Apesar de grande parte do trabalho ter consistido em estudo sobre Qualidade de Serviços em redes de computadores e o sistema EuQoS, a instalação de uma rede de testes nos permitiu alguns desafios que adicionaram alguma criatividade ao projeto. A utilização de máquinas virtuais para execução de testes em redes é algo que experimentamos aqui e que pode ser estudado mais profundamente.

Os testes realizados e descritos no capítulo anterior mostram que a rede de testes está pronta para ser utilizada e permitir mais estudos sobre o sistema EuQoS.

Referências Bibliográficas

- BAKER, F. et al. *Aggregation of RSVP for IPv4 and IPv6 Reservations*. IETF, set. 2001. RFC 3175 (Proposed Standard). (Request for Comments, 3175). Disponível em: <<http://www.ietf.org/rfc/rfc3175.txt>>.
- BLAKE, S. et al. *An Architecture for Differentiated Service*. IETF, dez. 1998. RFC 2475 (Informational). (Request for Comments, 2475). Updated by RFC 3260. Disponível em: <<http://www.ietf.org/rfc/rfc2475.txt>>.
- BRADEN, R.; CLARK, D.; SHENKER, S. *Integrated Services in the Internet Architecture: an Overview*. IETF, jun. 1994. RFC 1633 (Informational). (Request for Comments, 1633). Disponível em: <<http://www.ietf.org/rfc/rfc1633.txt>>.
- CHAN, K. et al. *COPS Usage for Policy Provisioning (COPS-PR)*. IETF, mar. 2001. RFC 3084 (Proposed Standard). (Request for Comments, 3084). Disponível em: <<http://www.ietf.org/rfc/rfc3084.txt>>.
- DAVIE, B. et al. *An Expedited Forwarding PHB (Per-Hop Behavior)*. IETF, mar. 2002. RFC 3246 (Proposed Standard). (Request for Comments, 3246). Disponível em: <<http://www.ietf.org/rfc/rfc3246.txt>>.
- DEMERS, A.; KESHAV, S.; SHENKER, S. Analysis and simulation of a fair queueing algorithm. In: *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*. New York, NY, USA: ACM, 1989. p. 1–12. ISBN 0-89791-332-9.
- DISTRIBUTED Weighted Random Early Detection. Cisco Systems, Inc., 2008. Disponível em: <http://www.cisco.com/en/US/docs/ios/11/_1/feature/guide/WRED.html>.
- DURHAM, D. et al. *The COPS (Common Open Policy Service) Protocol*. IETF, jan. 2000. RFC 2748 (Proposed Standard). (Request for Comments, 2748). Updated by RFC 4261. Disponível em: <<http://www.ietf.org/rfc/rfc2748.txt>>.
- EL-GENDY, M.; BOSE, A.; SHIN, K. Evolution of the internet qos and support for soft real-time applications. *Proceedings of the IEEE*, v. 91, n. 7, p. 1086–1104, July 2003. ISSN 0018-9219.
- ENHANCED IP Services for Cisco Networks. Indianapolis: Cisco Press, 1999.
- ENRÍQUEZ, J.; ANDRÉS, J. *Definition of Business, Communication and QoS models - Intermediate*. [S.l.], March 2005.
- FLOYD, S.; JACOBSON, V. Random early detection gateways for congestion avoidance. *Networking, IEEE/ACM Transactions on*, v. 1, n. 4, p. 397–413, Aug 1993. ISSN 1063-6692.

- FLOYD, S.; JACOBSON, V. Link-sharing and resource management models for packet networks. *Networking, IEEE/ACM Transactions on*, v. 3, n. 4, p. 365–386, Aug 1995. ISSN 1063-6692.
- FU, X. et al. Nsis: a new extensible ip signaling protocol suite. *Communications Magazine, IEEE*, v. 43, n. 10, p. 133–141, Oct. 2005. ISSN 0163-6804.
- GRACIÀ, R. et al. Active measurement tool for the euqos project. *IPS-MoMe*, Mar 2005.
- GROUP, A.-V. T. W. et al. *RTP: A Transport Protocol for Real-Time Applications*. IETF, jan. 1996. RFC 1889 (Proposed Standard). (Request for Comments, 1889). Obsoleted by RFC 3550. Disponível em: <<http://www.ietf.org/rfc/rfc1889.txt>>.
- HEINANEN, J. et al. *Assured Forwarding PHB Group*. IETF, jun. 1999. RFC 2597 (Proposed Standard). (Request for Comments, 2597). Updated by RFC 3260. Disponível em: <<http://www.ietf.org/rfc/rfc2597.txt>>.
- INTERNET2 QBone. Disponível em: <<http://qbone.internet2.edu/>>.
- KATEVENIS, M.; SIDIROPOULOS, S.; COURCOUBETIS, C. Weighted round-robin cell multiplexing in a general-purpose atm switch chip. *Selected Areas in Communications, IEEE Journal on*, v. 9, n. 8, p. 1265–1279, Oct 1991. ISSN 0733-8716.
- MEGEN - The Multi-Generator. Naval Research Laboratory of the US, 2008. Disponível em: <<http://pf.itd.nrl.navy.mil/mgen/mgen.html>>.
- NICHOLS, K. et al. *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. IETF, dez. 1998. RFC 2474 (Proposed Standard). (Request for Comments, 2474). Updated by RFCs 3168, 3260. Disponível em: <<http://www.ietf.org/rfc/rfc2474.txt>>.
- NICHOLS, K.; JACOBSON, V.; ZHANG, L. *A Two-bit Differentiated Services Architecture for the Internet*. IETF, jul. 1999. RFC 2638 (Informational). (Request for Comments, 2638). Disponível em: <<http://www.ietf.org/rfc/rfc2638.txt>>.
- PEUHKURI, M.; GROHN, A.; SIMOLA, O. Clock synchronisation in industry-standard computer hardware. *Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on*, p. 1–8, May 2007.
- REXFORD, J.; GREENBERG, A.; BONOMI, F. Hardware-efficient fair queueing architectures for high-speed networks. *INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, v. 2, p. 638–646 vol.2, Mar 1996.
- SHREEDHAR, M.; VARGHESE, G. Efficient fair queueing using deficit round-robin. *Networking, IEEE/ACM Transactions on*, v. 4, n. 3, p. 375–385, Jun 1996. ISSN 1063-6692.
- SILVA, M. A. Um sla para voip e seu mapeamento em uma rede diffserv/mpls. julho 2005.

VERMA, D. Service level agreements on ip networks. *Proceedings of the IEEE*, v. 92, n. 9, p. 1382–1388, Sept. 2004. ISSN 0018-9219.

WHITE, P.; CROWCROFT, J. The integrated services in the internet: state of the art. *Proceedings of the IEEE*, v. 85, n. 12, p. 1934–1946, Dec 1997. ISSN 0018-9219.

WROCLAWSKI, J. *The Use of RSVP with IETF Integrated Services*. IETF, set. 1997. RFC 2210 (Proposed Standard). (Request for Comments, 2210). Disponível em: <<http://www.ietf.org/rfc/rfc2210.txt>>.

5 Anexo A - Artigo

Instalação de uma rede de testes para a plataforma EuQoS

Pedro H. Ribeiro¹

¹Departamento de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC) Florianópolis – SC Brazil

pribeiro@inf.ufsc.br

***Abstract.** This paper describes the network testbed deployed by the author as his bachelor's degree conclusion work. This network deployed at the Universidade de Santa Catarina is intended to allow the study of the EuQoS platform and the execution of network measurement tests. A methodology for one way delay measurements had to be developed in order to avoid the clock drift problems between different computers. This paper also reviews the main QoS and EuQoS concepts.*

***Resumo.** Este artigo discorre sobre a rede de testes desenvolvida durante o trabalho de conclusão de curso do autor. A rede implantada na UFSC tem como objetivo permitir o estudo da plataforma EuQoS e a execução de experimentos de medição em redes. Uma metodologia para testes de atraso de único sentido precisou ser desenvolvida para resolver o problema da variação da frequência entre relógios de máquinas diferentes. Este artigo faz também uma revisão sobre os principais conceitos de QoS e sobre o sistema EuQoS.*

1. Introdução

Para que se ofereça ao usuário da Internet garantias a respeito da qualidade do serviço que ele vai experimentar, são necessárias diferentes camadas de gerenciamento e de implementação. Uma macro-arquitetura de QoS é composta por conceitos gerais de como a rede será organizada e gerenciada. Em um nível mais baixo, temos os equipamentos de rede, os roteadores, e os mecanismos e algoritmos que impactam diretamente na performance da rede que se traduzirá na qualidade experimentada pelo usuário.

A o sistema EuQoS é um exemplo de uma macro-arquitetura projetada para agregar a Internet a tecnologia necessária para oferecer garantia de QoS por sessão fim-a-fim. Isto quer dizer; permitir que dois usuários na Internet estabeleçam uma sessão para transferência de dados com garantia estrita de QoS.

A proposta deste artigo é descrever a rede de testes configurada e instalada no Laboratório de Pesquisa em Sistemas Distribuídos (LAPESD) na UFSC como trabalho de conclusão de curso do autor. Esta rede foi montada com o objetivo de oferecer um ambiente onde a plataforma de QoS EuQoS pudesse ser executada e testada.

Além de executar o sistema, a estrutura necessitava ser apta para permitir medições de parâmetro de rede, como atraso em único sentido, e vazão. Esta rede foi instalada com sucesso e está disponível para estudos do sistema EuQoS naquele laboratório.

2. Qualidade de Serviço

Esta sessão revisa os conceitos de qualidade de serviço e desempenho de rede necessários para a compreensão do processo de instalação da rede.

Requisitos de QoS são normalmente traduzidos para diferentes parâmetros de desempenho. Estes parâmetros são quantificações de determinadas características do fluxo de dados. Garantia de qualidade de serviço em redes é garantir que tais parâmetros estejam dentro de limites especificados pelo nível de qualidade requerido.

2.1. Parâmetros de QoS

Os seguintes parâmetros de desempenho são básicos (os demais são derivados destes [El-Gendy, Bose e Shin 2003]).

Vazão é a quantidade efetiva de unidades de dados transmitidas por unidade de tempo (ex., bits/segundo). Normalmente este parâmetro é referenciado como "garantia de banda". Garantia de vazão envolve alocação da capacidade do link e a capacidade de processamento dos nós intermediários.

Atraso é o tempo transcorrido entre a saída do pacote da fonte e sua chegada no destino. Existe um atraso associado a cada uma das camadas de rede utilizadas entre a fonte e o destino - camada de aplicação, camada de transporte, camada de rede, camada de enlace e camada física. A qualidade requisitada é expressa em termos do limite de maior atraso D_{max} . Um dos problemas atacados durante as etapas de medições neste trabalho foi a medição do atraso em uma única direção (*one-way delay*). O problema é derivado das dificuldades de se sincronizar os relógios da fonte e do destino do teste. Muitas vezes o intervalo total de ida e volta é utilizado (*round-trip delay*), mas não é uma métrica muito boa devido a possíveis diferenças entre as rotas de ida e de volta dos pacotes.

Varição de atraso, como no caso do atraso, também possui sua qualidade medida em relação a um limite J_{max} . Ele representa a variação do valor do parâmetro de atraso durante o tempo e pode ser quantificado de diferentes maneiras. Este parâmetro é comumente chamado "jitter".

Perda é a razão entre o número de unidades de dados que não chegaram até o destino e o número de unidades de dados enviadas medida em um intervalo de tempo. Pode ser lida como uma "probabilidade" de perda. Retransmissões não alteram o valor da taxa de perda, são somente um meio de recuperá-las.

Confiabilidade está ligado com a perda mas representa um conceito diferente. Confiabilidade diz respeito a porcentagem de unidades de dados que enfim foram recebidas corretamente no destino. Muitos protocolos utilizam retransmissão para recuperar uma perda e oferecer confiabilidade à camada superior.

2.2. Blocos básicos para provisão de QoS em nível de rede

Uma implementação de QoS em nível de rede requer funcionalidades adicionais dos dispositivos de rede além das funcionalidades básicas de encaminhamento e roteamento de pacotes. Entre estas estão os seguintes.

Controle de Admissão. Independente da abordagem de QoS aplicada, a rede ainda possui recursos limitados. Os links, os buffers dos equipamentos e suas

capacidades de processamento são limitadas. Sendo assim, a chave de toda rede com QoS está na maneira como distribui seus recursos entre todas as requisições de serviços dos diferentes clientes. Controle de admissão é o processo de comparar os requisitos de serviço com os recursos disponíveis e decidir se aceita ou rejeita o pedido de serviço baseando-se no critério de que a admissão de um novo pedido não pode levar a um nível de qualidade de serviço inaceitável para este ou para os serviços previamente admitidos. Controle de admissão pode ser realizado explicitamente, rejeitando o pedido de serviço, ou implicitamente como no caso da prática de controle de banda. O controle de admissão explícito é um dos processos fundamentais na plataforma EuQoS.

Escalonamento. Existe uma grande diversidade de algoritmos propostos para realizar escalonamento de pacotes na camada de rede. O objetivo deles é sempre distribuir a quantidade de vazão disponível no link entre diferentes classes de serviço de maneira que cada uma receba uma fatia previamente definida ou justa, oferecendo então algum tipo de garantia, normalmente estatística, dos valores dos parâmetros de rede como atraso, variação no atraso e perda de pacotes.

Técnicas de gerenciamento de buffers. Em redes de alta velocidade que utilizam nós com um "produto do atraso de vazão" (*delay-bandwidth product*) alto - o número de bits que o emissor precisa transmitir antes que o primeiro bit chegue no destino - os gateways são projetados com buffers equivalentemente grandes para suportar congestionamentos transientes. O protocolo TCP detecta um congestionamento assim que um pacote é descartado pelo gateway. No entanto, não seria desejável ter filas grandes (possivelmente da ordem do produto de atraso de vazão [Floyd e Jacobson 1993]) que estivessem cheias todo o tempo, o que ocasionaria um aumento do atraso na rede. Desta maneira, em redes de alta velocidade, é de maior importância manter a vazão alta mas o tamanho médio dos buffers baixo. Diferentes técnicas, como o RED e o WRED [Floyd e Jacobson 1993] são utilizados para efetuar controle do tamanho desses buffers realizando descartes de pacotes.

Policimento. É normalmente aplicado nas bordas da rede próximo à fonte. Este mecanismo tem o objetivo de verificar se as características do tráfego estão em conformidade com o acordo de nível de serviço negociado e garantir que somente os recursos contratados sejam utilizados. Quando um fluxo difere do esperado, ele pode ter pacotes descartados ou marcados para que recebam um tratamento de menor prioridade pela rede. *Token Buckets* são uma possível implementação deste mecanismo.

2.3. Arquitetura de Serviços Diferenciados para QoS (DiffServ)

Em 1998 a IETF iniciou um grupo de trabalho com o objetivo de desenvolver e padronizar os princípios de uma nova arquitetura de QoS [Blake et al. 1998].

O modelo DiffServ separa todo o tráfego em um número de agregados - grupos de fluxos de dados que recebem o mesmo tratamento da rede - cada um com diferentes requisitos de QoS e faz com que os diferentes agregados sejam tratados diferentemente pelos dispositivos da rede. Define também os tipos de tratamento desejados dos dispositivos de rede para cada tipo de agregado.

Para realizar esta agregação do tráfego é necessário utilizar alguns bits no cabeçalho dos pacotes que permitam diferenciá-los. Estes bits, no contexto DiffServ, constituem o "campo DS". A RFC 2474 [Nichols et al. 1998] especifica o campo DS que coincide com o byte ToS no cabeçalho IP. O conteúdo deste campo é chamado *DS codepoint* (DSCP), e ocupa somente seis bits do byte ToS.

O DSCP é o campo utilizado para definir o tipo de tratamento que o pacote receberá dos equipamentos da rede. Estes diferentes tipos de tratamento são denominados de *per-hop behavior* (PHB). A arquitetura DiffServ é baseada sobre a definição de alguns poucos PHBs diferentes, que cobrem os agrupamentos básicos dos diferentes requisitos de QoS. A marcação dos bits do campo DS dos pacotes define por qual dessas PHBs ele será tratado. Dos três PHBs diferentes atualmente definidos, em adição ao serviço padrão de melhor-esforço, este trabalho faz o uso de dois.

O PHB *Expedited Forwarding* (EF), definido na RFC 3246 [Davie et al. 2002], foi projetado para oferecer um serviço de baixo atraso, baixa variação no atraso e baixa perda sobre uma vazão garantida. O princípio é tentar fazer com que os pacotes que tem seu DSCP marcados como EF encontrem sempre filas de encaminhamento pequenas nos roteadores. Isto normalmente é atingido fazendo com que os recursos de encaminhamento sejam alocados para esses pacotes em uma taxa sempre maior do que a taxa com que eles chegam. Esta PHB é utilizada por aplicações com requisitos estritos de atraso e variação no atraso.

O PHB *Assured Forwarding* (AF) foi projetado para oferecer serviços de pouca perda e vazão garantida e no entanto não oferecer quaisquer garantias de atraso e variação no atraso. Ele foi definido pelo grupo de trabalho DiffServ da IETF na RFC 2597 [Heinanen et al., 1999]. Este PHB consiste em três diferentes comportamentos de encaminhamento de pacotes, AFx1, AFx2 e AFx3 em ordem crescente de precedência de descarte (o último tem maior probabilidade de sofrer descartes do que o primeiro). O símbolo "x" representa uma entre as diferentes classes AF. Os pacotes não podem sofrer reordenamento entre as classes. Esta PHB oferece então diferentes níveis de serviços dadas as possíveis combinações de classes e precedência.

Em um domínio com suporte a DiffServ um tráfego tem seu campo DS alterado na entrada da rede de acordo com o tratamento que se deseja que este receba. Desta maneira este será tratado de acordo por todos os nós do domínio.

3. O Sistema EuQoS

O projeto EuQoS foi realizado por um consórcio de centros de pesquisa de provedores de serviços e universidades da União Européia. Este projeto estava inserido dentro de um fundo de fomento a pesquisas europeu na áreas de Qualidade de Serviço na Internet. O objetivo era projetar e implementar um sistema de Garantia de QoS, o sistema EuQoS, que seria composto por tecnologias já existentes e pelo resultado do desenvolvimento de novas tecnologias que unidas resolvessem o problema global de garantir Qualidade de Serviço fim-a-fim sobre redes heterogêneas.

3.1. Visão geral da solução proposta pelo sistema

A diferença principal entre a proposta do projeto EuQoS e as arquiteturas propostas anteriormente está no empenho realizado na definição de uma arquitetura global que quebrasse a complexidade do problema maior entre diferentes subsistemas.

Entre as regras fundamentais do projeto [Enríquez e Andrés 2005] estava o entendimento de que a infra-estrutura de rede da Internet já está bem consolidada (ex. xDSL, UMTS, ATM) e que por isso o sistema precisaria utilizar as ferramentas que cada uma dessas tecnologias já implementam para obter QoS.

A estratégia utilizada pelo sistema EuQoS segue o conceito de estabelecer na rede um número de classes de serviço de QoS (CoS), cada uma orientada para tratar diferentes tipos de tráfego IP com diferentes objetivos de QoS.

O objetivo do sistema passa a ser então estabelecer um número de classes de serviço de rede em todo o caminho entre um *usuário a* e um *usuário b*. Essas classes de serviço de rede são visíveis pelas aplicações dos usuários e mantidas através de múltiplos domínios de rede, ainda que estas implementem diferentes tecnologias de rede - lembrando que o objetivo principal do sistema é permitir que usuários em diferentes redes de acesso como xDSL, UMTS, WLAN, LAN/Ethernet, possam estabelecer uma conexão com garantia de QoS. É importante então que as classes de serviço definidas no sistema satisfaçam as expectativas de QoS das aplicações dos usuários e possam ser implementadas em tecnologias de rede.

Diferentes tecnologias implementam diferentes mecanismos para prover as classes de serviços definidas. Esses mecanismos são responsáveis por garantir as expectativas de QoS; o que quer dizer que a rede deve oferecer aos fluxos de pacotes as características (como atraso e perda), determinadas para a classe de serviço para o qual foram submetidos. Entre esses mecanismos estão os descritos na seção anterior, como escalonadores de pacotes, policiadores e classificadores instalados nos equipamentos.

A maneira como o sistema EuQoS propõe resolver o problema da garantia de QoS fim-a-fim é definindo um caminho entre as duas pontas de maneira que a soma dos parâmetros de qualidade oferecidos por cada um dos domínios no caminho satisfaça os requisitos de QoS fim-a-fim (Figura 1). Desta maneira, o tráfego do usuário submetido a uma CoS obterá os requisitos de QoS fim-a-fim para esta classe.



Figura 1. Os parâmetros de QoS fim-a-fim são formados a partir dos parâmetros oferecidos por cada domínio

Para que haja garantia de qualidade de serviço em cada domínio dois conceitos são aplicados: o dimensionamento (também chamado provisionamento) e controle de admissão de conexões (descrito na seção anterior).

Cada domínio de rede que implementa o sistema EuQoS necessita dimensionar, ou provisionar, a quantidade de recursos (como capacidade de link e tamanho de buffers) que reservará para as diferentes classes de serviço.

Com o conhecimento da quantidade de recursos disponível para cada uma das classes de serviço e dos requisitos de recursos de cada uma das conexões, o sistema EuQoS utiliza controle de admissão de conexões para limitar o número de conexões que o domínio aceitará em cada classe. Desta maneira garante o cumprimento dos requisitos de qualidade para as conexões já aceitas.

O sistema EuQoS é a implementação de uma infra-estrutura que permite que os domínios se comuniquem com as aplicações dos clientes, com domínios vizinhos e com a tecnologia de rede para configurar sessões com garantia de qualidade sob demanda.

De maneira superficial, configurar uma sessão significa: receber a requisição do cliente para iniciar uma sessão; descobrir um caminho até o destino que permita o cumprimento da qualidade requerida; configurar sua rede para tratar o fluxo de dados da sessão do cliente pela CoS selecionada e requisitar do próximo domínio no caminho que faça o mesmo.

Note que a maneira como cada domínio fará para garantir que o tráfego submetido a uma classe de serviço seja servida com os requisitos mínimos de qualidade é independente do sistema como um todo. Cada domínio aplicará, dependendo da tecnologia de rede que utiliza, técnicas como engenharia de tráfego, policiamento e escalonamento para garantir o cumprimento dos requisitos em sua rede.

3.2. Arquitetura do Sistema EuQoS

A prática de dividir para conquistar foi muito utilizada no projeto do sistema EuQoS. O sistema está dividido em camadas conceituais e essas camadas tiveram suas funções implementadas por grupos de módulos.

Vamos descrever aqui o sistema a partir de seus três níveis conceituais: a camada de aplicação, a camada de rede virtual e a camada de transferência. A Figura 2 permite visualizar esses três níveis.

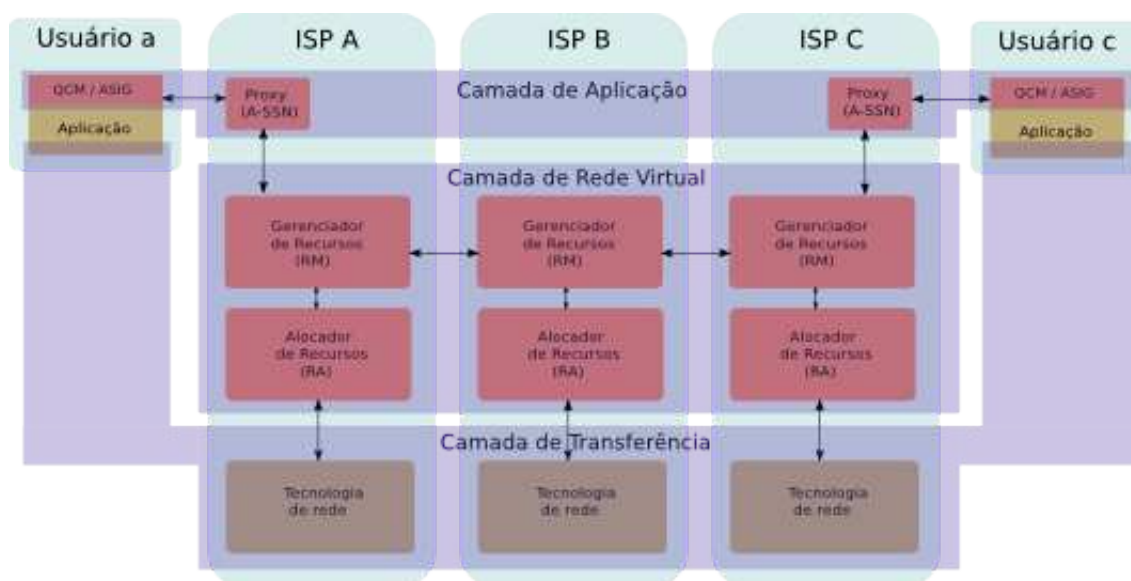


Figura 2. Diferentes camadas da arquitetura EuQoS

É através da camada de aplicação que a aplicação do *usuário a*, com suporte ao EuQoS, se comunica com o sistema EuQoS de seu provedor de serviços para sinalizar o desejo de iniciar uma nova sessão com QoS. Esta camada propaga esta comunicação até o domínio alvo e em seguida ao usuário alvo.

A camada de rede virtual é responsável pela gerência e alocação dos recursos tanto internamente aos ASs como globalmente em todo o caminho do *usuário a* até o

usuário alvo. Ela é composta por uma parte independente de tecnologia (*Technology Independent - TI*) controlada por um gerenciador de recursos, o RM (*Resource Manager*), e uma parte dependente de tecnologia (*Technology Dependent*) controlada por um alocador de recursos, o RA (*Resource Allocator*).

O RM gerencia os recursos de rede em um nível mais alto que o RA. Ele abstrai a tecnologia de rede implementada pelo AS e trabalha sobre dados nominais que são quantificações dos recursos que este AS possui. Para efetuar este gerenciamento o RM implementa uma série de funcionalidades, entre elas o controle de admissão de novas conexões (CAC) no domínio, o gerenciamento de SLAs com domínios vizinhos, as decisões de roteamento para a obtenção de um caminho com QoS entre os usuários, e a comunicação com RMs dos domínios vizinhos para efetuar solicitações de QoS. Quando o RM toma uma decisão sobre efetuar uma reserva de recurso em seu domínio, ele se comunica com o RA para que este aplique as configurações nos dispositivos de rede específicos. As decisões tomadas pelo RM se baseiam nos valores nominais dos recursos da rede, nos SLAs estabelecidos e no número de conexões já aceitas.

4. Implantação da rede

A Figura 3 mostra como foi montada a rede para instalação de dois domínios EuQoS locais. Os módulos que compõem o RA foram instalados no Host A de cada domínio. Os módulos que compõem o RM foram instalados no Host B de cada domínio. Dois roteadores Linux foram utilizados para fazer a função de roteadores de borda dos domínios. Em cada domínio foi instalado computadores para desempenharem o papel de clientes de cada domínio.

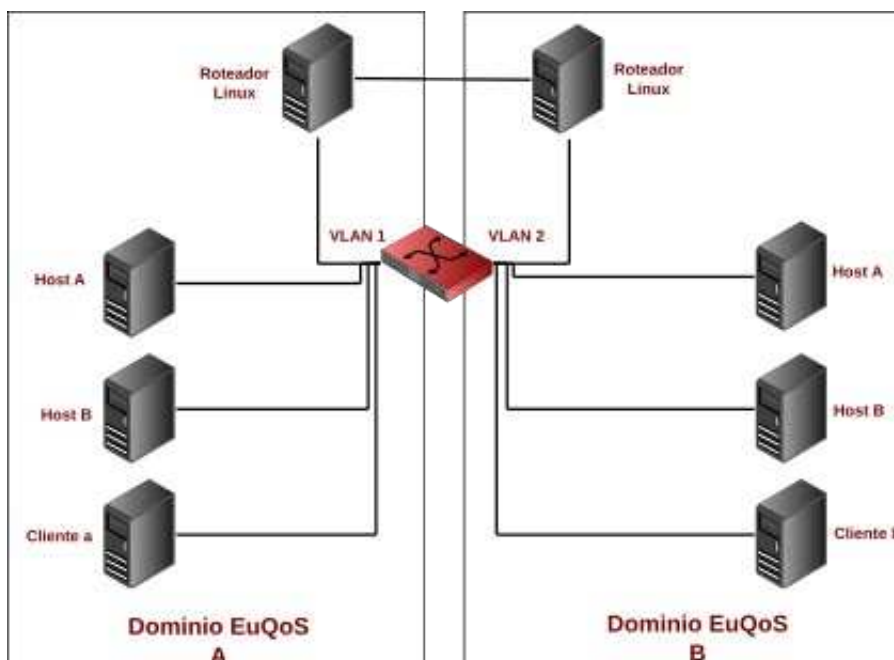


Figura 3. Diagrama da rede de testes

4.1. O problema da sincronização de relógios

Com a configuração exibida da Figura 3, onde ambos os domínios estão conectados em um mesmo switch Gigabit e com todas as interfaces de rede dos computadores sendo também Gigabits, o tempo de atraso entre as redes é demasiadamente pequeno e a metodologia de medição se tornou crítica. Após configurar as placas de rede envolvidas no caminho dos dados entre os dois domínios para 10Mbit o tempo aumentou um pouco com primeiros testes de atraso de sentido único entre os clientes de ambos os domínios exibindo valores da ordem de poucos milissegundos (menos de uma dezena).

Com quantias de tempo desta ordem é preciso garantir uma sincronização muito precisa dos relógios da máquina fonte e destino.

A principal fonte de variação entre a frequência de relógios em computadores diferentes é o fato de que a frequência dos osciladores de cristal utilizados pelos computadores comuns variam de acordo com diferenças construtivas e também com a temperatura. Uma solução imaginada, e experimentada aqui, foi a utilização de um ambiente de virtualização onde ambos *Cliente a* e *Cliente b* pudessem ser executados sobre o mesmo hardware.

A solução de virtualização precisaria ser testada já que a infra-estrutura de virtualização acrescenta latências no sistema. Precisaríamos saber se conseguiríamos contornar o problema dos relógios e se, mesmo sobre um sistema de virtualização, conseguiríamos obter uma boa precisão nas medições.

Conforme proposto em [Peuhkuri, Grohn e Simola 2007], um possível método para a avaliação do sincronismo entre relógios de dois computadores é fazer com que ambos capturem pacotes simultaneamente em um mesmo ponto da rede e então analisar a diferença entre as estampas de tempo atribuídas por ambos.

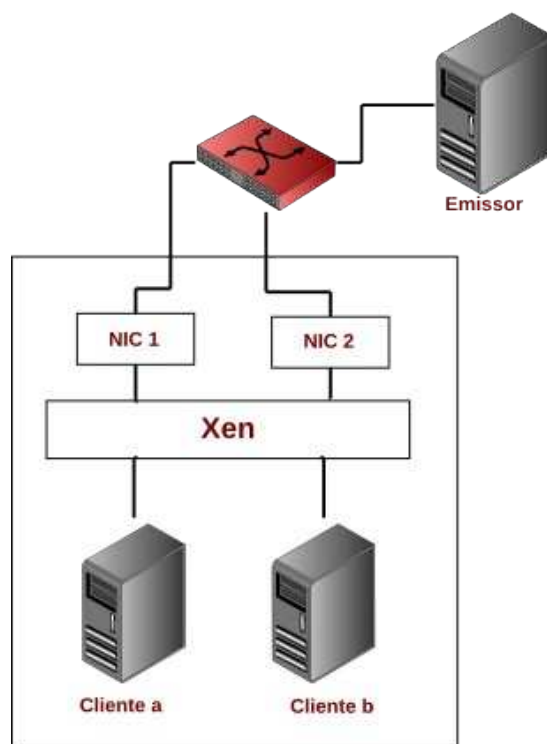


Figura 4. Estrutura de teste para medição de diferenças entre relógios

Este teste foi realizado emitindo pacotes para um endereço broadcast em uma taxa fixa na rede que eram capturados por ambos os clientes. A Figura 4 mostra como foi montada a rede para os testes.

Foram emitidos 10.000 pacotes de 64 bytes em intervalos fixos em uma taxa de 110 pacotes por segundo. A duração aproximada do teste foi de 90 segundos. A taxa escolhida foi baixa para que não houvesse perda nas placas ou no switch. O tráfego de teste era o único na rede durante sua duração.

A Figura 5 e a Tabela 1 exibem dados das medições da diferença entre as estampas de tempo atribuídas pelas duas máquinas para cada um dos pacotes.

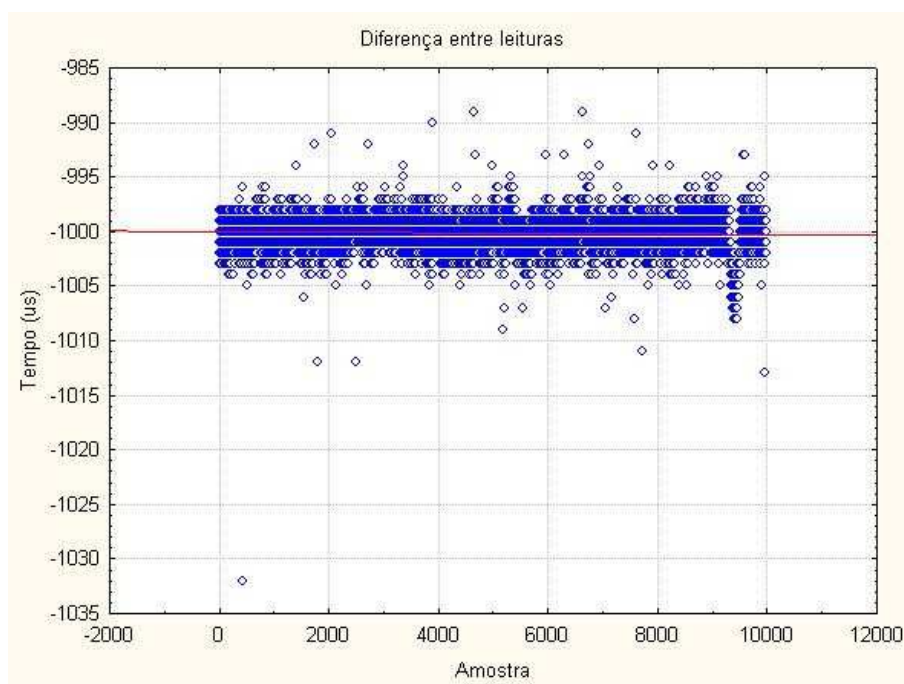


Figura 5. Plotagem das diferenças entre leituras

Diferença Média (μs)	-1000,13
Diferença Mínima (μs)	-1032,00
Diferença Máxima (μs)	-989,00
Desvio Padrão (μs)	1,522

Tabela 1. Resultados do teste

O resultado do teste foi satisfatório. Neste caso a distância entre os relógios permaneceu muito próximo de 1ms durante todo o teste. O valor do desvio padrão e dos intervalos de confiança mostram que esta metodologia oferece uma precisão da ordem de dezenas de microssegundos. Esta metodologia de testes foi então utilizada em todos os testes descritos a partir daqui neste trabalho.

5. Conclusão

A rede com dois domínios EuQoS locais está instalada e funcional no LAPESD. Com a arquitetura de testes de medição descrita acima a rede está pronta para servir de base para estudos e testes da plataforma EuQoS.

Referencias

- Blake, S. et al. “An Architecture for Dierentiated Service”. IETF, dez. 1998. RFC 2475(Informational). (Request for Comments, 2475). Atualizado pela RFC 3260. <http://www.ietf.org/rfc/rfc2475.txt>
- Davie, B. et al. An Expedited Forwarding PHB (Per-Hop Behavior). IETF, mar. 2002. RFC 3246 (Proposed Standard). (Request for Comments, 3246). <http://www.ietf.org/rfc/rfc3246.txt>
- Enríquez, J.; Andrés, J. Denition of Business, Commnication and QoS models -Intermediate. [S.l.], March 2005. <http://www.euqos.eu>
- El-Gendy, M.; Bose, A.; Shin, K. “Evolution of the internet qos and support for softreal-time applications”. Proceedings of the IEEE, v. 91, n. 7, p. 1086-1104, Julho 2003.
- Floyd, S.; Jacobson, V. “Random early detection gateways for congestion avoidance”. *Networking*, IEEE/ACM Transactions on, v. 1, n. 4, p. 397-413, Agosto 1993.
- Heinanan, J. et al. Assured Forwarding PHB Group. IETF, jun. 1999. RFC 2597(Proposed Standard). (Request for Comments, 2597). Updated by RFC 3260. <http://www.ietf.org/rfc/rfc2597.txt>
- Nichols, K.; Jacobson, V.; Zhang, L. “A Two-bit Dierentiated Services Architecture for the Internet”. IETF, jul. 1999. RFC 2638 (Informational). (Request for Comments, 2638). <http://www.ietf.org/rfc/rfc2638.txt>
- Peuhkuri, M.; Grohn, A.; Simola, O. “Clock synchronisation in industry-standard computer hardware”. Testbeds and Research Infrastructure for the Development of Networks and Communities, 2007. TridentCom 2007. 3rd International Conference on, p. 1{8, May 2007.