UNIVERSIDADE CATÓLICA DE BRASÍLIA

PROGRAMA DE GRADUAÇÃO LATO SENSU
EM BACHARELADO EM REDES DE COMPUTADORES

Ciência da computação

FERRAMENTA GRÁFICA PARA O CONTROLE DE BANDA UTILIZANDO O HTB

Autores: Leonardo Otto Mussel Zanutto
Wagner Ferreira Nunes
Orientador: Eduardo Lobo

BRASÍLIA 2005

LEONARDO OTTO MUSSEL ZANUTTO
WAGNER FERREIRA NUNES

FERRAMENTA GRÁFICA PARA CONTROLE DE BANDA UTILIZANDO O HTB

Projeto Final apresentado ao Programa de Graduação "Lato Sensu" em Ciência da Computação da Universidade Católica de Brasília, como requisito para obtenção do Título de Bacharel em Ciência da Computação.

Orientador: Eduardo Lobo

Brasília

2005

TERMO DE APROVAÇÃO

Dissertação defendida e aprovada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, defendida e aprovada, em 7 de Dezembro de 2005, pela banca examinadora constituída por:

_____

Eduardo Lobo

_____

Mauro Tapajós

Brasília
UCB

A minha Família, pelo incentivo, apoio durante a minha vida acadêmica e compreensão nos momentos de minha ausência. Ao meu amigo e companheiro de projeto pela grande dedicação e esforços aplicados, bem como a compreensão em vários momentos difíceis enfrentados no projeto e no decorrer do curso.

Wagner Ferreira Nunes

Ao Prof., MSc, Eduardo Lobo,

Incentivador guia e mestre com importantes sugestões, sempre disposto em auxiliar nossa formação profissional e intelectual.

RESUMO

A comunicação e transferência de dados estão cada vez mais presentes no mundo atual, principalmente em empresas, onde se tornou um serviço essencial para competir no mercado. O controle de banda vem para ajudar estas empresas a terem uma melhor utilização de sua rede, evitando que invistam constantemente no aumento da largura de banda disponível sem necessidade. Frequentemente ocorrem casos em que os usuários de uma rede utilizam a banda para realizar tarefas não vinculadas à suas atividades na empresa ou instituição, onerando a infra-estrutura disponibilizada, diminuindo seu desempenho. Ferramentas gráficas para controle de banda, geralmente, são proprietários, e softwares livres para realizar estes tipos de tarefas são de difícil utilização. Foi proposto a criação de uma interface Web para uma ferramenta de controle de banda livre, o (HTB), que a torne mais intuitiva e de fácil utilização.

PALAVRAS-CHAVE: Desempenho, Controle, largura de banda, software livre, HTB.

ABSTRACT

Communication and data transfer are each time more present in the current world, mainly in companies, where its a essential service to survive in the market. The bandwidth control comes to help these companies to have a better use of his network, preventing them to invest constantly in connections of better quality without having necessity. Frequently occur cases that employees of the company are using the bandwidth to make tasks not entailed with the company, burdening the network and diminishing its performance. Graphical tools for bandwinth control, generally, are proprietors, and free softwares for these tasks are very hard to configure, then it. Was proposed the accomplishment of a Web interface for a free bandwidth control tool (HTB) that make this configuration simpler and much easier to use .

Keywords: *Performance, Bandwidth, Control, Free software, HTB.*

1. Introdução

1.1. Motivação

O desperdício de largura de banda com tráfego não relacionado aos negócios de uma empresa, leva a uma má qualidade dos serviços prestados no que se refere à velocidade de transferência de dados, ocasionando paradas constantes nas aplicações por falta de recursos. Sendo assim, é visível a falta de um controle de banda nos ambientes informatizados.

Porém, o que é controle de banda? O controle de banda é o meio que se utiliza para se alocar recursos de banda para aplicações críticas em uma rede e manter aplicações sem maior importância ou nocivas para rede sobre controle.

Para efeito de exemplificação será suposto que o responsável por um centro de informática, solicita à uma empresa prestadora de serviços de telecomunicação para que seja fornecido um link de dados para o fornecimento de serviço Internet. O propósito deste serviço é fornecer acesso à rede mundial de computadores a todos os funcionários do estabelecimento. Após o link de dados ser entregue pela prestadora de serviços, o roteador é configurado para que Internet propriamente dita seja disponibilizada.

O link disponibilizado possui um gerenciamento o pró ativo. Todas as máquinas de usuários são configuradas de modo que o serviço seja disponibilizado. Ao passar dos dias, o responsável pelo departamento de informática percebe que vários usuários reclamam de lentidão dos serviços. É verificado que a utilização do link está constantemente em 100% de uso da banda disponível. Um aumento da capacidade do serviço de Internet é disponibilizado gerando maior banda disponível para o tráfego de informações de Internet.

O circuito é instalado. Ao passar dos dias, novas reclamações de usuários são percebidas. Novamente é percebido que o problema é idêntico ao anterior sendo necessários novos aumentos de velocidade no circuito de dados, o que não será a melhor solução.

O responsável verifica, na estação de gerência, que uma porção significativa da largura da banda disponível está sendo utilizada por um grupo restrito de usuários, e em tarefas não associadas à atividade da empresa. Por exemplo: download arquivos de programas, filmes ou músicas. Diante desta situação o que pode ser feito?

O objetivo deste projeto é prover uma solução para este tipo de problema. Fornecendo uma interface gráfica de fácil operação para controle de banda.

Limites de banda são disponibilizados por classes (tipos de informações). Isto quer dizer que o gerente de redes deste ambiente poderá limitar a banda de download conforme um flexível conjunto de regras, permitindo uma melhor utilização da rede para um tráfego legítimo aos interesses da empresa.

1.1. Breve Histórico

Existem diversos programas livres com a finalidade de controlar a banda de uma rede de dados. As ferramentas, geralmente, funcionam de forma adequada, porém, na maioria dos casos, o manuseio destas não é intuitivo, são configuradas via linhas de comando dificultando bastante a sua utilização. Visando minimizar estes problemas, foi sugerido em 15 de junho de 2005, na Universidade católica de Brasília, o desenvolvimento de uma interface gráfica via Web de fácil manuseio para interagir com um destes controladores de banda em software livre, especificamente o HTB.

Em algumas décadas atrás, as redes de computadores eram de acesso restrito, ou seja, funcionava apenas em universidades, centros militares e instituição do governo, logo o seu conteúdo era ligado estritamente as atividades afins de onde era instalada.

Em tempos remotos da Guerra Fria nasceu a Arpanet para manter a comunicação das bases militares dos Estados Unidos, mesmo que o Pentágono fosse riscado do mapa por um ataque nuclear.

Quando a ameaça da Guerra Fria passou, Arpanet tornou-se tão inútil que os militares já não a consideravam tão importante para mantê-la sob a sua guarda. Foi assim

permitido o acesso aos cientistas que, mais tarde, cederam a rede para as universidades as quais, sucessivamente, passaram-na para as universidades de outros países, permitindo que pesquisadores domésticos a acessarem, até que mais de 5 milhões de pessoas já estavam conectadas com a rede e, para cada nascimento, mais 4 se conectavam com a imensa teia da comunicação mundial.

Era o início da World Wide Web, esse meio foi enriquecido. O conteúdo da rede ficou mais atraente com a possibilidade de incorporar imagens e sons. Um novo sistema de localização de arquivos criou um ambiente em que cada informação tem um endereço único e pode ser encontrada por qualquer usuário da rede denominada de localizador de recursos uniforme ou URL (*Uniform Resource Locator*).

Em síntese, a Internet é um conjunto de redes de computadores interligadas que tem em comum um conjunto de protocolos e serviços, de uma forma que os usuários conectados possam usufruir de serviços de informação e comunicação de alcance mundial.

Recursos tecnológicos estão cada vez mais baratos, permitindo que usuários domésticos com pequenas redes de computadores e empresas tenham acesso a Internet de banda larga. O crescimento descontrolado pode acarretar grandes custos para uma empresa, pois esta provavelmente estará sempre aumentando seus recursos sem muitas vezes ter real necessidade, um controlador de tráfego pode ser a solução.

Devido esta diversidade de informações e a suas facilidades de acesso, foi notado a falta de um controle dos dados que pela rede trafegavam, restringindo informações não interessantes para os serviços essenciais das empresas e reservando recursos para o que realmente fazia parte de suas atividades. Devido estas necessidades, surgiu o controle de banda. O controle de banda é a administração das informações que trafegam em uma rede de computadores.

2. Objetivos

2.1. Objetivo Geral

O projeto tem por finalidade a criação de um ambiente gráfico Web para configuração e gerenciamento de banda em uma rede. As duas interfaces de rede da máquina onde estará instalada a ferramenta de controle de banda são do padrão Ethernet podendo uma destas portas estarem conectadas a um roteador de saída para a Internet e outra à rede interna permitindo o controle de banda entre estes dois ambientes.

O software irá prover um ambiente que permita a, de forma prática, a configuração de uma ferramenta de controle de banda disponível para o sistema operacional linux, o HTB. Por se tratar de uma ferramenta Web, oferecerá mobilidade desta aplicação permitindo que um administrador de redes interaja com o aplicativo em qualquer espaço físico que possua acesso a Internet, evitando seu deslocamento em longas distâncias para acessar, de forma urgente, o controlador que a interface faz fronteira.

2.2. Objetivos Específicos

- Estudar a teoria e protocolos associados ao gerenciamento de consumo de banda.
- Pesquisar a existência de ferramentas de controle de banda em software livre.
- Pesquisar a existência de ferramentas gráficas proprietárias ou livres de controle banda.
- Selecionar ferramentas livres para geração e medição de tráfego.

A proposta é criar-se uma solução gráfica utilizando-se a linguagem de programação JSP em conjunto com programas escritos na linguagem C++ através da qual o usuário poderá criar, excluir e ativar várias regras para o controlador da banda HTB.

Os programas em C++ serão utilizados para a manipulação dos scripts a serem lidos pelo HTB.

A quantidade de computadores para o desenvolvimento e testes da aplicação será de três máquinas. A primeira atuará como um servidor do programa gerador de tráfego o Iperf, a segunda como um roteador, servidor Jsp e controlador do tráfego entre duas redes e a terceira como cliente Iperf. Em todas as máquinas, serão instalados os softwares de desenvolvimento da aplicação como o Netbeans, GCC++ e Java. Em todas as máquinas será instalado o sistema operacional Linux. A figura 1a ilustra a topologia da rede utilizada no desenvolvimento.

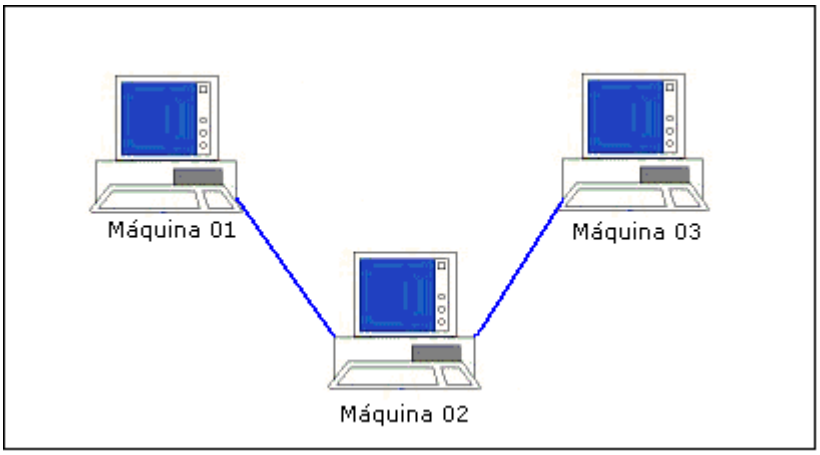


Figura 1a - Topologia da rede utilizada no desenvolvimento

O funcionamento ocorrerá da seguinte forma: Um dos Clientes/Servidor irá enviar dados para a outra estação Cliente/Servidor que estará em outra rede, à máquina que estará intermediando as duas redes, terá duas placas de comunicação, nesta máquina, além de ter instalado o software que controlará a banda, terá o servidor de aplicação Tomcat e o software que será desenvolvido.

Ao serem aplicadas as regras de controle de banda, o tráfego das informações entre os servidores deverá ficar limitado com um valor de banda próximo ao que foi estipulado nestas configurações.

3. Cronograma previsto

O desenvolvimento do projeto ocorreu de acordo com as seguintes etapas:	***Agosto***	***Setembro***	***Outubro***	***Novembro***	***Dezembro***
Estudar os conceitos de controle de banda	**P/R**	**P/R**			
Estudar os protocolos associados ao controle de banda	**P/R**	**P/R**	**P/R**		
Levantar e analisar as ferramentas de controle de banda existentes – ago/set	**P/R**	**P/R**			
Levantar e analisar as ferramentas para geração e medição de tráfego – set		**P/R**			
Levantar requisitos para o desenvolvimento da interface Web de gerenciamento de controle de banda – set./out.		**P/R**	**P/R**		
Desenvolver a solução web – out./nov.			**P/R**	**P/R**	
Teste de validação da solução – out./nov.			**P/R**	**P/R**	
Elaboração da monografia – ago./set./out./nov.	**P/R**	**P/R**	**P/R**	**P/R**	**P**

P Etapas previstas R Etapas realizadas P/R Etapas previstas e realizadas. Data da defesa: 07/12/05

Tabela 01 – Cronograma das atividades previstas e realizadas

4. PROPOSTA DA PESQUISA

4.1. Descrição da Pesquisa

Através do levantamento bibliográfico foi possível ter conhecimento dos materiais existentes relacionados ao controle de banda. Após o estudo destes materiais fez-se uma análise para identificar quais os assuntos são pertinentes ao estudo. Depois de identificados os assuntos, passou-se a realizar o desenvolvimento de acordo com as informações obtidas nos diversos materiais: registrando os resultados; analisando quais fatores e o grau de influência dos mesmos nos experimentos; classificando-os de acordo com a ordem de importância; e interpretando as execuções experimentais. Possibilitou a alteração da aplicação em função dos resultados obtidos nestes testes comparando com o material pesquisado.

4.2. Resultados Esperados

Pretende-se ao final deste projeto ter sido realizado todos os testes necessários para averiguar o funcionamento do controle de banda em uma rede e criar uma interface gráfica Web de fácil manuseio pelo administrador de redes, podendo, com ela, realizar a administração do tráfego de informações da rede de sua competência.

4.3. Restrições da Pesquisa Proposta

Por não ter sido encontrado, a pesquisa não contemplará testes de outras interfaces gráficas que auxiliam na configuração de controladores de banda.

4.4. Interessados e Beneficiados

São interessados e beneficiados todos os gerentes e administradores de redes que não possuam em suas redes de computadores uma ferramenta capaz de realizar o controle de banda de maneira eficaz e de fácil configuração.

4.5. Recursos Necessários

Para a realização da pesquisa foram necessários recursos de *hardware*, *software* e pessoal.

4.5.1. Recursos de *Hardware*

Durante a implementação, configuração e testes do serviço foram utilizados dois tipos de servidores:

- Servidor Web Tomcat com controlador HTB
  - 1 Computador Dell com processador Pentium IV de 2.8 Ghz
  - 512 Mb de Memória DDR
  - 1 HD de 80 Gb
- Máquinas Cliente/Servidor
  - 2 Computadores Dell com processador Pentium IV 2.8 Ghz
  - 512 Mb Memória DDR
  - 1 HD de 80 Gb

4.5.2. Recursos de *Software*

- Descrição dos *Softwares*

  Servidor Web TomCat

Software	***Versão***
Debian 3.1	Kernel 2.6.2
gcc	3.3.3-7
Htb	3
Java	5.0
TomCat	5.5

Tabela 2 – Versão dos produtos

Máquinas Cliente/Servidor

Software	Versão
Debian 3.1	Kernel 2.6.2
Iperf	2.0.2

Tabela 3 – Versão dos produtos

4.5.3. Recursos Físicos

Os experimentos e desenvolvimento da aplicação foram realizados no laboratório de redes C103 da Universidade Católica de Brasília.

4.5.4. Recursos Humanos

Participaram do desenvolvimento do projeto:

- 02 alunos de Graduação – Bacharelado em Ciência da Computação, Área de concentração Redes de Computadores.
- 01 orientador – Professor de Computação da UCB, Área de concentração Redes de Computadores.

4.6. Relação Custo Benefício

Como se trata de uma solução *open source*, não teremos lucro financeiro com a pesquisa, os dispêndios serão basicamente de recursos humanos, o benefício intelectual será para os envolvidos no projeto e a para a Universidade Católica de Brasília.

5. EMBASAMENTO TEÓRICO

5.1. O que é a largura de banda de uma rede?

A largura de banda de uma rede é a quantidade de dados, por unidade do tempo, que uma rede de dados pode transmitir. Sua medição é feita especificando um determinado espaço de tempo. Normalmente é medida em bits/s, bytes/s, kilobytes/s, megabytes/s ou gigabytes/s.

O tipo de medição que será utilizado depende do intervalo de tempo, por exemplo: Caso seja medido o total de banda consumido em um mês com acesso a internet provavelmente esta informação será apresentada em megabytes ou gigabytes pois a quantidade de informação é de nível elevado, mas se esta mesma informação for para um intervalo de tempo de 1(uma) hora será mostrado em kilobytes ou bytes. Porém existem casos em que o meio de transmissão é de alta capacidade, na ordem de Mbits, Gbits por segundo, logo não convém taxar estes links com unidade de medidas menores e tempos maiores que estas pois trabalharia com números enormes.

5.1.1. O que é controle de banda (*Bandwidth Control*)?

O controle de banda, também denominado *bandwidth control* é o controle da utilização da banda. É o meio que se utiliza para se alocar recursos de banda para aplicações críticas em uma rede e manter aplicações sem tanta importância ou nocivas para rede sob controle.

Aplicações de *download* de *softwares* estão cada vez mais comuns nas redes corporativas. Estas ferramentas podem ocupar completamente a banda, deixando aplicações de maior importância sem acesso ao canal de comunicação. Isso ocorre porque geralmente não são utilizadas ferramentas para controle de tráfego de informações no ambiente de comunicação de dados. Normalmente este controle funciona definindo o tráfego da rede em classes por aplicações e tipos de serviços onde é configurado um máximo e mínimo de banda que estas classes podem utilizar.

5.1.2. Porque Realizar controle de banda?

É notável que, nos últimos anos, as redes de computadores tiveram um aumento significativo gerando uma alta demanda de banda utilizada sem a realização de nenhum tipo de controle das informações que passam por estes links. Vários usuários e aplicações estão compartilhando um mesmo meio de transmissão de dados. Mediante este fato, deve ser utilizando o controle de banda para alocar recursos de rede para o usuário ou aplicação, podendo prevenir congestionamentos além de disponibilizar o canal de informações para aplicações com maior importância. A falta destas regras pode fazer com se tenha mais recursos que o necessário, gerando custos financeiros para a instituição.

"Mudanças estão ocorrendo na medida em que as empresas realizam seus negócios e processam suas informações além de estarem sendo guiadas pelas mudanças que ocorrem nas tecnologias de redes. A disponibilidade de novos serviços na Internet resultou no aumento do tráfego e do número total de usuários. Isto gerou a necessidade de aumentar a velocidade, o controle do tráfego de aplicações WEB"**(William Stallings)**

5.1.3. Protocolos associados ao controle de banda

TCP (*Transmission Control Protocol*)

A arquitetura TCP/IP surgiu com o objetivo principal de manter conectados, órgãos do governo e universidades dos Estados Unidos.

O protocolo TCP é orientado a conexão aplicando o algoritmo *three-way handshake* ou *three-fold* para iniciar uma comunicação com outro dispositivo de rede. É necessário saber qual o protocolo de aplicação da última camada que receberá os dados. [26]

Portas de comunicação da camada de transporte são disponibilizadas para tráfego de informações dos diversos serviços. Ao todo são 65.535 (64k) portas, sendo

que de 0 a 1024 chamadas de portas baixas, geralmente são portas padronizadas e portanto só podem ser usadas por aplicações que utilizem os serviços predefinidos. As portas de 1024 a 65535 chamadas de portas Altas e são atribuídas dinamicamente, sendo possível atribuir qualquer tipo de serviço/Aplicação a elas.

Camada Física

É o *meio* através do qual os dados são transmitidos, nela é tratada níveis de tensão, freqüência, níveis de luz (quando fibras óticas são utilizadas) . Inclui parâmetros que definem a transmissão, por exemplo: Quando a comunicação é feita por cabos metálicos, os níveis lógicos 0 e 1 são definidos por faixas de tensões e quando são utilizadas fibras óticas estes bits são representados, 1 pela presença de luz e 0 pela ausência. É importante observar que para cada tecnologia existe um padrão destes níveis.

Rede

Determina a rota que os dados seguirão da máquina de origem até o de destino. Tal rota dependerá das condições da rede, prioridade do serviço e outros fatores. Contém os protocolos IP e ICMP, e os protocolos de roteamento.

Esta camada pode ter um certo controle do tráfego nos canais de comunicação. Pode ser associada à função de fragmentação, ou seja, ter a subdivisão de pacotes grandes e um posterior reagrupamento destes pequenos pacotes nos dispositivos de destino é/ou interconexão.

No destino os dados são recompostos no seu formato original. Pode ser considerada uma das mais importantes, pois permitem que os dados cheguem ao destino da forma mais eficiente possível. O protocolo IP situa-se nesta camada.

Transporte

A camada de transporte inclui funções relacionadas com conexões entre a máquina fonte e máquina destino, segmentando os dados em unidades de tamanho

apropriado para utilização pelo nível de rede, seguindo ou não as orientações do nível de sessão.

As principais funções do nível de transporte são a criar conexões para cada requisição vinda do nível superior, multiplexar as várias requisições vindas da camada superior em uma única conexão de rede, dividir as mensagens em tamanhos menores, a fim de que possam ser tratadas pelo nível de rede e estabelecer e terminar conexões através da rede. [33]

O acesso das aplicações à camada de transporte é feito através de portas que recebem um número inteiro para cada tipo de aplicação, podendo também tais portas serem criadas ao passo em que novas necessidades vão surgindo com o desenvolvimento de novas aplicações.

Aplicação

São os protocolos reconhecidos pelos programas dos usuários finais da rede. Exemplo: correio eletrônico, FTP, navegador (browser), etc. No TCP esta camada representa as a camada de sessão, apresentação e aplicação como uma só. [27]

UDP (User Datagrama Protocol)

Este protocolo da camada de transporte é utilizado em casos que o dispositivo de origem não necessitar da garantia de chegada de dados ao receptor. Por exemplo, no caso de vídeo conferência, é utilizado o protocolo UDP, este protocolo não é orientado a conexão ou seja, não precisa estabelecer conexão entre origem e o destino antes de enviar os dados, não verificando ao menos se o dispositivo está ativo, ele apenas envia os dados.

O protocolo UDP empacota os dados e os envia para camada inferior (rede 3) para que o protocolo IP dê prosseguimento ao envio dos dados. Estes pacotes podem estar segmentos, porém são numerados antes de serem enviados, não sofrem nenhuma

verificação de chegada ao destino. Pode-se comparar o UDP com um correio, prepara-se uma carta e envia-se para o correio na esperança de que chegue ao seu destino.[28]

5.1.4. Controle de Congestionamento

É considerado congestionamento em rede de computadores, o excesso de informações em um determinado meio de comunicação de dados, fazendo com que haja descarte de informações gerando retransmissões. Estas retransmissões fazem com que as aplicações funcionem de maneira precária, pois as informações que estas necessitam para continuar o processamento chegam atrasadas ou não chegam.

O congestionamento que impede o a realização da demanda de tráfego de uma maneira eficiente e responsável. Se o congestionamento não é controlado, ele enche os *buffers* dos roteadores e *switchs* e em conseqüência pacotes passam a ser descartados. Para aplicações que toleram a perda de pacotes o descarte significa retransmissão, fazendo com que, as aplicações passem a funcionar de maneira insatisfatória gerando a conhecida lentidão das tarefas, logo, a perda de pacotes, significa uma perda na qualidade de serviço.

O controle de congestionamento pode-se resumido em dois grupos de soluções: loop-aberto e loop-fechado.

Ferramentas que utilizam o loop-aberto, incluem decisões de quando aceitar novos pacotes, decidindo quando deve descartá-los e quais deles. Todas as ferramentas que realizam este tipo de controle realizam estas decisões sem considerar o estado da rede.

No caso do loop-fechado é baseado na resposta que a rede fornece para o usuário isto é realizado em três partes, monitoramento da rede para detectar quando e onde o congestionamento ocorre, passa esta informação para o local aonde pode-se tomar alguma ação, ajustar a operação do sistema para corrigir o problema.

Várias métricas podem ser tomadas como referência para monitorar uma rede. Como exemplo: todos os pacotes que foram descartados por falta de espaço no buffer, o tamanho do enfileiramento, a quantidade de pacotes que tiveram que ser retransmitidos.

5.1.5. *DiffServ* e *IntServ*

Existem dois modelos de qualidade de serviço que podem ser aplicados em uma rede de dados, o *IntServ* e o *DiffServ*, serviços integrados e serviços diferenciados.

Serviços integrados

O modelo, conhecido por *IntServ* (*Integrated Services*) serviços integrados, refere-se a diversos trabalhos realizados pela IETF (*Internet Engineering Task Force*), que gerou diversas RFCs (*Request For Coments*), com a finalidade de criar uma arquitetura para dar suporte a aplicações de multimídia.

O *IntServ* baseia-se em reserva de recursos (largura de banda, atraso e *jitter*), antes do estabelecimento da comunicação. Para que haja esta reserva de recursos é utilizado um protocolo de reserva de recursos, o RSVP (*Reservation Protocol*). Antes de iniciar a comunicação entre dispositivos que necessitam de QoS, este protocolo reserva recursos fim a fim para que haja uma boa qualidade de comunicação. “O *IntServ* é implementado por quatro componentes, o protocolo de sinalização (RSVP), a rotina de controle de admissão, o classificador e o escalonador de pacotes. A função destes componentes é organizar os pacotes de forma que a QoS seja aplicada.”[41]

Serviços Diferenciados

Denominado de DiffServ (*Differentiated Services*), este tipo de serviço tem como característica a diferenciação dos pacotes, no qual são marcados de acordo com classes de serviços pré-determinadas. “Este modelo se destaca por oferecer uma característica indispensável: escalabilidade. Esta que pode ser obtida por meio de

agregação de fluxos e da separação das funções dos roteadores (borda e núcleo), em grandes redes de backbone. Redes que implementam DiffServ recebem o nome de Domínios DS. Estes Domínios DS negociam garantias mínimas de QoS, na qual fazem um “contrato” para a transmissão das aplicações dos usuários.

Os pacotes que trafegam entre domínios distintos são policiados nos roteadores de borda, que verificam sua conformidade com os contratos. Os roteadores do centro da rede apenas encaminham os pacotes aos seus destinos, oferecendo algumas garantias de QoS a determinados pacotes. Assim, diferentes fluxos podem ser tratados com distinção nos roteadores, que atendem seus requisitos de QoS. Tratamento este que é chamado PHB (*Per-Hop Behavior*).

Combinando o PHB com as regras de policiamento da borda, se permite a criação de diversos serviços numa rede *DiffServ*. Atualmente há dois PHBs sendo padronizados pelo IETF: Encaminhamento Expresso (*Expedited Forwarding* - EF) e Encaminhamento Assegurado (*Assured Forwarding* - AF). O PHB EF define garantias mais rígidas de QoS para aplicações que são mais sensíveis a variações de tempo na rede, já o PHB AF é utilizado por serviços que não necessitam de garantias rígidas, para obter diferenciação (preferência) no tráfego de seus pacotes na rede.”[41]

A qualidade de serviço pode ser observada de duas formas: do ponto de vista da aplicação ou da rede. Para uma aplicação, oferecer seus serviços com qualidade significa atender às expectativas do usuário em termos do tempo de resposta e da qualidade, muitas vezes subjetiva, do serviço que está sendo provido, ou seja, fidelidade adequada do som e/ou da imagem sem ruídos nem congelamentos.

A qualidade de serviço da rede depende das necessidades da aplicação, ou seja, do que ela requisita da rede a fim de que funcione bem e atenda, por sua vez, às necessidades do usuário. Estes requisitos são traduzidos em parâmetros indicadores do desempenho da rede como, por exemplo, o atraso máximo sofrido pelo tráfego da aplicação entre o computador origem e destino. [34]

A qualidade de serviço em redes de computadores está relacionada ao fato de que diversos dispositivos em uma rede, reservarem recursos fim a fim para uma

determinada aplicação/serviço. Adiante será detalhado como são realizadas estas reservas de acordo com o protocolo RSVP (*Reservation Protocol*).

Tradicionalmente, numa rede IP, todo tráfego é tratado de forma única. Isto quer dizer que todo pacote recebe o melhor esforço da rede para que este seja transmitido, sem garantias de que este chegue ao seu destino. O conceito do QoS é de que algumas aplicações ou usuários são mais importantes que outras, o que prevê um tratamento diferenciado.

O controle de banda geralmente é realizado por uma máquina na entrada/saída de uma rede de computadores. Somente este nó será responsável pelo controle de todo o tráfego de *download* e *upload.* Geralmente é realizado um controle baseado em classes e filtros. Este controle será detalhado no capítulo 6.

A seguir será mostrado como é o funcionamento de um protocolo de reserva de recursos, o RSVP. Aplicações multimídia são aplicações de tempo real onde o tempo de tráfego das informações na rede é muito importante. Estes são exemplos de aplicações que utilizam reservas de recursos. Para que estes requisitos fossem satisfeitos, foram desenvolvidos mecanismos em que as redes de comunicação de dados possam oferecer diversas qualidades de serviços exigidas para aplicações multimídia (vídeo e voz) que exigem uma qualidade mínima em termos de parâmetros temporais (como atraso e jitter) e capacidade efetiva de transmissão (como largura de banda).

Geralmente o RSVP funciona juntamente com algum software que possui mecanismos de escalonamento de pacotes, de classificação de fluxos de tráfego e de controle de admissão. Para efeitos de exemplificação será citado o ALTQ que foi desenvolvido para máquinas com sistema operacional FreeBSD e que possui interface com o RSVP.

Será mostrado como é o comportamento do trafego e o impacto que o RSVP juntamente com ALTQ oferece serviços de informações multimídia.

Para que o RSVP funcione de forma adequada, por ser um protocolo permite que as aplicações requisitem diferentes QoS para seus fluxos de dados, dois parâmetros devem ser levados em consideração:

Todos os elementos de redes tais como roteadores, devem suportar os mecanismos de controle de qualidade de serviço para garantir a entrega dos pacotes de dados, ou seja, devem ter implementado o RSVP em seu sistema operacional de entrada e saída (IOS).

É importante ser observado que este protocolo atua tanto em roteadores como em máquinas de usuários, fornecendo os recursos requisitados pelas aplicações. A figura 1 exemplifica este processo [14].

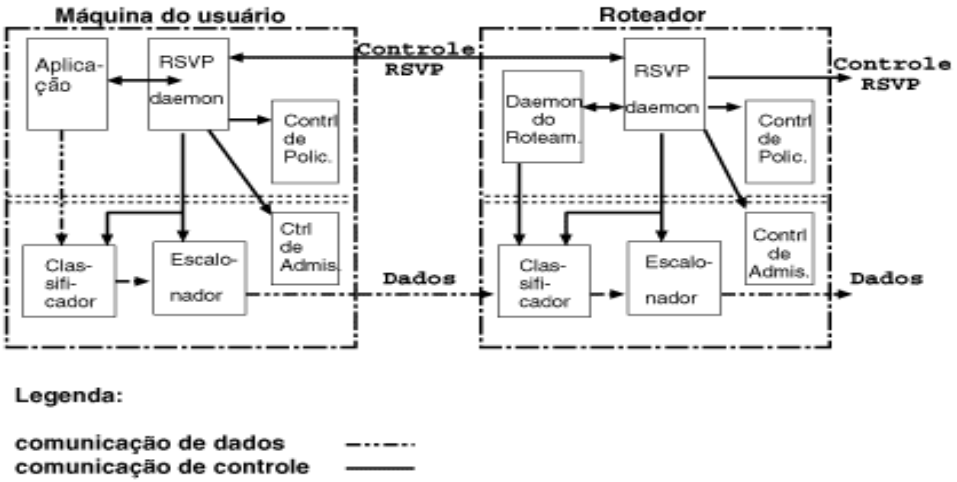


Figura 1 - RSVP em Máquinas do Usuário e Roteadores [14].

O RSVP trabalha de forma simplex, um único sentido de cada vez, ou seja, ele trata transmissões e recepções de formas distintas, operando juntamente com a camada de transporte. O RSVP não é um protocolo de transporte, ele trabalha no mesmo nível dos protocolos ICMP (*Internet Control Message Protocol*), o IGMP (*Internet Group Management Protocol*), e protocolo de roteamento (OSPF), conforme a figura 02.

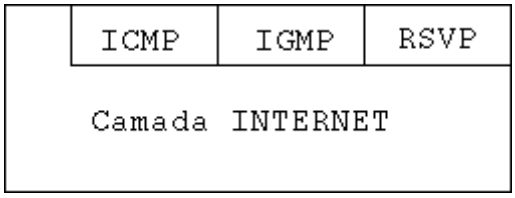


Figura 02 - camada de funcionamento do RSVP

Para melhor entendimento deste texto a seguir serão dadas algumas definições:

- **Sessão**

"O protocolo RSVP define como *sessão* todo enlace de comunicação pelo qual se relacionam as camadas de transporte de todos os participantes da comunicação, podendo ser ponto a ponto ou *multicast*. Cada sessão é tratada independentemente. O conceito de sessão é propositalmente genérico, pois uma sessão pode ser estabelecida baseando-se em valores de QoS diferentes daqueles requisitados pelo receptor inicialmente. Tal fato deve-se à liberdade que o gerenciador possui em unir recursos ao longo do caminho de dados da aplicação, sempre tendo o melhor aproveitamento dos recursos como objetivo. Ao efetuar essa política, os valores de QoS requisitados poderão sofrer alterações, desde que essas não acarretem perda de qualidade para uma comunicação já estabelecida.

- **Soft-state**

O protocolo RSVP é baseado na noção de *soft-state,* é um estado dos dispositivos que participarão da reserva dos recursos. O início do *soft-state* ocorre quando uma mensagem de reserva é recebida e realizada no elemento; este estado é periodicamente realimentado pelos receptores. Ao invés de entregar à rede a responsabilidade em detectar e responder à falhas, o RSVP delega aos receptores o trabalho de reenviar periodicamente suas requisições de serviços. Caso uma falha ocorra, somente uma nova requisição do serviço restabelecerá o *soft-state* nos roteadores. [14]

Mensagens

Mensagens são trocadas entre as aplicações e os diversos elementos de rede para que haja a correta qualidade de serviço prevista em uma determinada sessão.

Os objetivos RSVP são trocados após a definição da sessão, são elas mensagens de controle do RSVP. As principais mensagens são RESV e PATH. Cada uma deverá explicitamente requisitar a QoS através de objetos descritores do tráfego e filtros. Estes objetos podem sofrer alterações afim de comportar os diversos parâmetros de interesse.

Um transmissor, ao necessitar de recursos, envia uma mensagem PATH que percorre a rede por um caminho ponto a ponto ou *multicast* através de caminhos definidos pelos mecanismos de encaminhamento até os receptores.

A mensagem PATH, ao atravessar a rede em cada nó por quem a mensagem PATH passou, criará um estado chamado de PATH state. Esta mensagem armazena o estado de todos os nós por onde ela passou.

A mensagem RESV, é uma resposta do receptor em função da mensagem PATH, que contém um descritor de fluxo, informando os parâmetros de QoS aceitos por este receptor.

É através da troca destas mensagens que o RSVP toma uma serie de decisões, como por exemplo, aceitar ou não um novo fluxo, criando um ambiente para que os recursos sejam reservados. Os parâmetros utilizados para requisição de qualidade de serviço, estão representados nas mensagens do RSVP.

“Através da análise dos parâmetros trazidos pelos objetos RSVP, algumas simplificações podem ser feitas, de forma a agrupar fluxos distintos de uma mesma sessão que possuam características comuns. Tal tarefa, apesar da complexidade, proporciona, quando é possível, uma economia dos recursos utilizados, principalmente em termos de largura de banda. Para que isso possa ocorrer, é essencial a utilização dos estilos de reserva, assunto que será abordado em seguir.” [14]

A figura a seguir ilustra como é feita a reserva de recurso. Seja T1 uma máquina que contém uma aplicação que esteja solicitando recursos de QoS, R1 é uma outra máquina que receberá dados da aplicação da máquina T1 e G roteadores intermediários. A aplicação em T1 envia uma mensagem de controle chamada de mensagem de caminho (ou *path message* ou **PATH**), para iniciar a requisição de reserva de recursos. Esta mensagem irá percorrer diversos roteadores na rede até chegar à R1, de acordo com a figura 03[14]. A mensagem PATH contém em sua formação um cabeçalho RSVP além de todas as informações sobre o tráfego que a aplicação em **T1** espera gerar adicionada de valores para os parâmetros de QoS. A cada roteador **G** existente entre **T1** e **R1**, será criado um estado chamado de **PATH** *state*. [14]

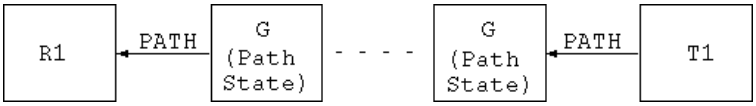


Figura 03- Mensagem PATH Seguindo de T1 a R1 [14].

Esta mensagem, ao chegar em R1, é analisada por este nó que seleciona os parâmetros de reserva desejados montando a mensagem de reserva (ou *reservation message* ou **RESV**). A mensagem RESV é retornada provocando, nos roteadores um estado chamado SOFT state, até retornar em T1 informando-o sobre as condições dos parâmetros de QoS da reserva realizada por **R1** e propriedades do caminho entre ambos, como é mostrado na figura 04[14].

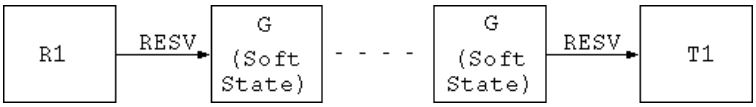


Figura 04- Reserva entre **R1** e **T1**, Criando o *Soft State* nos Elementos Intermediários.

5.1.6. Vantagens de utilizar controle de banda

Link de dados é um recurso de alto custo. Mediante este fato, torna-se necessário um controle sobre o que está ocupando a banda do meio de transmissão dos

dados. Surgem, logo, diversos métodos para que não haja informações dispensáveis e/ou indesejáveis ao bom funcionamento dos serviços da rede.

Ferramentas para este fim utilizam várias informações contidas nos pacotes para realizar esta filtragem. Campos dos pacotes de dados como endereço de origem, endereço de destino, portas dos serviços, assinaturas contidas nos pacotes. Estas informações são as mais utilizadas na seleção dos pacotes.

Geralmente é necessário limitar a quantidade de banda para determinada aplicação, pois algumas preenchem a banda na medida em que ela vai sendo disponibilizada, ou seja, *upgrade* nos links não é a solução. Nestes casos a única forma de represar estes tráfegos é identificando-os e limitando-os ou até mesmo eliminando-os.

Com o controle de banda é possível poupar recursos financeiros, pois geralmente não é necessário realizar alterações na capacidade do link, porém não deve ser descartada a necessidade de upgrade dos links de comunicação devido ao constante crescimento das redes de computadores.

5.1.7. Especificações de controle de banda

Para a especificação do controle de banda, foi utilizado o TSPEC que é uma especificação de tráfego (*Traffic SPECification* ou *TSPEC*).

Para entender as TSPECs, deverá ser explicado em que consiste o balde de *token*. Basicamente, existe um vasilhame que vai sendo preenchido lentamente com estes *tickets*. Cada pacote enviado retira uma unidade e, caso não exista mais, o pacote não poderá ser enviado. A velocidade em que ele preenchido regula a taxa média do fluxo de tráfego, enquanto o volume (profundidade) dita a variação permitida no fluxo. Tipicamente, As TSPECs apenas especificam estes dois parâmetros: a taxa dos *tokens* e a profundidade do balde. Por exemplo, um vídeo com taxa de amostragem de 75 quadros por segundo e se cada um deste necessitar de 10 pacotes, pode especificar uma taxa de 750Hz, e uma profundidade de balde de apenas 10. A profundidade seria

suficiente para acomodar o pico associado ao envio imediato de um quadro. Por outro lado, uma conversação iria necessitar uma taxa inferior, mas uma profundidade do vasilhame muito superior, dado que existem muitas pausas durante uma conversação e, portanto, pode ser satisfeita com menos *tokens* não enviados as pausas entre palavras e frases. Contudo, isto implica que a profundidade do balde seja superior para compensar a quantidade de palavras enviadas.[1]

O fluxograma da figura 05[1] e a figura 06, mostra como é realizado o controle de tráfego pelo balde de *tokens.*

Os termos deste fluxograma têm as seguintes definições:

- Tc: intervalo de tempo contratado.
- FIFO: primeiro que entra é o primeiro que sai da fila *(first-in-first-out)*
- CQ: *custom queuing*
- PQ: prioridade de enfileiramento

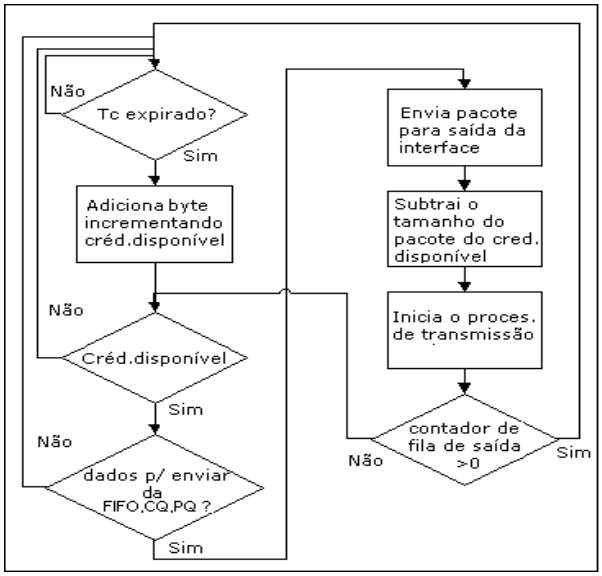


Figura 05 - Fluxo do Balde de tokens[1].

- **Policiamento e modelagem**

A modelagem ocorre na saída de informações (*upload*) da rede. É utilizada para: controlar o tráfego; reduzir a largura da banda disponível e também para diminuir o congestionamento de tráfico quando há uma grande quantidade de usuários.

O policiamento é semelhante à modelagem, porém este é realizado sobre entrada e saída de informações. O CAR [3] (*Committed Access Rate*) é uma forma de policiamento que realiza basicamente duas tarefas, em redes IP:

1ª - Controle de banda com limites para a taxa de transferência *(policing)* – este policiamento permite que em uma determinada interface, haja controle de entrada ou saída de dados (transmissão ou recepção de dados). Os dados transmitidos ou recebidos estão, neste caso, limitados a regras preestabelecidas, ficando os pacotes excedentes descartados ou reclassificados com menores prioridades de transmissão.

2ª - Classificação de pacotes segundo regras de precedência IP ou de grupos de QoS (regras internas no roteador permite o particionamentos da rede em múltiplas classes de serviços ou níveis de priorização)

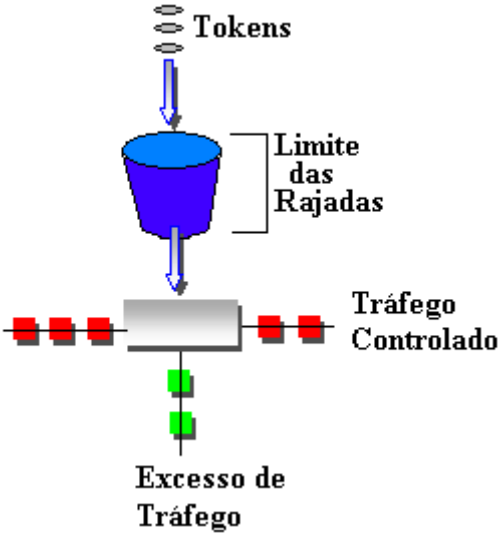


Figura 6 - *Commited Access Rate* com balde de tokens[3]

"A profundidade do balde é o tamanho da rajada (*burst size*). Se houver *tokens* suficientes quando o tráfego chega ao balde, então o tráfego é dito estar em conformidade e a quantidade correspondente de *tokens* é removida. Se não houver *tokens* suficientes, então o tráfego é dito excessivo, neste caso o pacote pode ser descartado, reclassificado com outro nível de prioridade, etc."[3]

Neste mecanismo, o balde armazena *tokens* gerados por um relógio em um certo intervalo de tempo. Para que um pacote possa ser transmitido é necessário capturar e destruir um *token*. Neste processo é possível realizar alteração na quantidade de *tokens* que são depositados na unidade do tempo, permitindo desta forma, um melhor gerenciamento das explosões de tráfego que ocorrem na rede.

5.2. Ferramentas de controle de banda existentes

5.2.1. O Dummynet

Apesar de não ter sido utilizado no projeto, devido ao tempo escasso, será descrito o funcionamento do controle de banda utilizando o FreeBSD juntamente com o Dummynet[4].

O Dummynet[4] é uma ferramenta do sistema operacional FreeBSD, muito flexível para gerenciamento de banda para impor ou simular algumas condições desejadas no tráfego da rede. Ele funciona interceptando os pacotes podendo efetuar a limitação de banda, perdas de pacotes, retardos de propagação etc. Todo o gerenciamento deste controlador é feito através do *firewall* deste sistema operacional, o *ipfirewall*, que deverá estar configurado e funcionando. Os próximos tópicos mostrarão em detalhes sua configuração.

O controle de tráfego, ou seja, controle do roteamento dos pacotes, no sistema operacional FreeBSD pode ser realizado pelo seu *firewall*, o *ipfirewall*, em conjunto de outros aplicativos, o Dummynet. Com a inclusão deste controlador no FreeBSD, o controle de tráfego deixou de ser apenas filtragem de pacotes segundo regras preestabelecidas filtrando quem e quando pode acessar determinado serviço, rede e/ou estação. A inclusão desta nova ferramenta possibilitou o controle de tráfego extensivo, diferentemente do *IPFilter* (pacote de programas utilizados para realização de Nat e *Firewall*) [37].

Todo o roteamento de tráfego na infra-estrutura básica de uma rede pode ser gerenciado pela implementação do *ipfirewall* e o *dummynet*, porém, por questões de

desempenho, atrasos em filas de cada regra a ser criada, deseja-se que nenhuma regra seja criada de forma dinâmica.

Probabilidade de Ocorrências (Probability Matching)

Para provar a grande funcionalidade do *ipfirewall*, será detalhado o parâmetro "prob" de grande funcionalidade para auditar e testar uma rede. Com ele é possível simular a restrição de pacotes de forma aleatória sob várias taxas de probabilidade. Trata-se de uma ferramenta estatística que utiliza a opção "prob" nas regras do *firewall*. Esta opção é seguida de uma probabilidade estatística representada por um número entre 0 e 1. Trata-se da probabilidade dos pacotes que serão liberados pelo firewall. Dessa maneira, uma probabilidade 0.9 (indicada pelo comando "prob 0.9") permitirá que o tráfego de 90% dos pacotes que forem restringidos por aquela regra, de maneira semelhante que uma "prob 0.1" permitirá apenas 10% de probabilidade. A seguir será representado como é a sintaxe para este tipo de aplicação do ipfirewall.

<comando> [<no. regra>] [prob <prob_ocorrencia>] <acão> [log [logamount <número>]] **<proto>** from **<origem>** to **<destino> [<interface-spec>] [<opcoes>]**

Para melhor exemplificar, se a intenção for restringir 20% dos pedidos de echo (*echo requests*) do ICMP, pode ser utilizada a seguinte regra:

Add 1000 prob 0.8 allow icmp from any to any in icmptypes 8

Poderá também querer negar 50% dos pacotes TCP SYN para o servidor Web, dessa forma simulando um tráfego pesado na interface ep0. Para isto, será feito da seguinte forma:

add 1000 prob 0.5 allow tcp from any to any in setup via ep0

A utilização de probabilidade de ocorrências é uma função nativa do *ipfirewall.*

Opções do *Dummynet*

Para que se possa implementar todas outras funcionalidade do *Traffic Shapping*, será necessário o uso do *dummynet*, que foi incorporado na versão 2.2.8 do FreeBSD. Logo será necessário adicionar uma opção ao *kernel* do sistema operacional.

Após ter compilado o *kernel* do FreeBSD com a opção *dummynet* adicionada as opções típicas do *ipfirewall*. Terá, agora, que especificar a criação de túneis, chamados de "*pipes*", para controle de tráfego. Um túnel nada mais é que uma regra que permite a canalização das informações segundo estas regras. Através do comando "pipe <pipe #>" no *ipfirewall* é possível realizar a criação destes túneis. Para melhor ilustrar o funcionamento deste comando, será exemplificado sua utilização através da criação de um túnel simples:

pipe 10 config bw 100Kbit/s

É importante ser observado que o comando "pipe" é apenas mais uma opção de configuração do *ipfirewall* assim como "*add"* ou "*delete*", porém neste caso antes de cada comando é feita uma chamada ao *ipfirewall* (/sbin/ipfw pipe 10 por exemplo).

No exemplo do comando anterior, será criado um túnel que vai limitar o tráfego de informações para uma taxa de 100 Kilobits por segundo. Esta medida foi utilizada para efeito de exemplificação, porém existem outras que podem ser utilizadas, são elas: bit/s, Byte/s, Kbit/s, Kbyte/s, Mbit/s, Mbyte/s. O parâmetro obrigatório "bw" indica que a seguir será especificado o *bandwidth* (banda).

Existem outras formas de controlar o tráfego. Outra delas é o *"delay"* (atraso) , que forca um atraso na comunicação, tendo como efeito o que se chama de *"lag"* do sistema. A regra a seguir exemplifica o uso deste comando:

pipe 10 config delay 100

Neste caso, valor 100 significa o tempo, em milissegundos, que serão atrasados os dados. Isso significa que todos dados roteado através desse túnel terão um atraso de 100ms. Alem estes artifícios para controle de banda, pode-se lançar mão do sistema de perda de pacotes. Como exemplo, será suposto que se deseja 20% dos pacotes que passarão por determinado caminho se percam (equivalente a "prob 0.8"). Para isto será criado o seguinte túnel:

O parâmetro "**plr**" significa "*packet loss rate*" (taxa de perda de pacotes), este parâmetro força a perda de pacotes, portanto o valor indica à que taxa os pacotes não serão roteados, oposto da opção "prob" do *ipfirewall*, que indica a taxa dos pacotes que serão roteados. É interessante que ao se utilizar uma das possibilidades, que utilize somente esta para evitar confusões. Ambas são igualmente efetivas, porém o prob é nativo do *ipfirewall* e o segundo faz parte do *dummynet*.[4]

5.2.2. Brmultiaccess [38]

O Brmultiaccess é uma ferramenta gráfica que além de outras funcionalidades, realiza o controle de banda. É uma ferramenta proprietária com alto custo de aquisição.

Alguns recursos do **BRmultiaccess** são: compartilhamento da conexão, gerenciamento através de interface Web (relatórios, estatísticas e gráficos), *firewall* com alarmes, mascaramento de IP, *Proxy* transparente, servidor de e-mail, *web-mail*, controle de banda, DMZ, multiplataforma, antivírus, cluster, VPN e servidor DHCP.

5.3.2. CBQ (*Class Based Queuing*)

CBQ é um algoritmo que realiza gerenciamento de tráfego foi desenvolvido por um grupo de estudiosos em rede no Laboratório Nacional de *Lawrence Berkeley* mantido pela Universidade da Flórida, Estados Unidos. [35]

O controle de banda é realizado através da divisão do tráfego em classes hierárquicas com qualquer combinação de IP´S, protocolos e tipos de aplicações. Desta forma o CBQ permite que administradores de redes possam gerenciar a banda de maneira hierárquica. [7]

Dispositivos que utilizam CQB permitem o gerenciamento e classificação do tráfego de acordo com alguma exigência das aplicações, garantindo que aquele tipo de tráfego receba a banda necessária. Veja a figura 07.

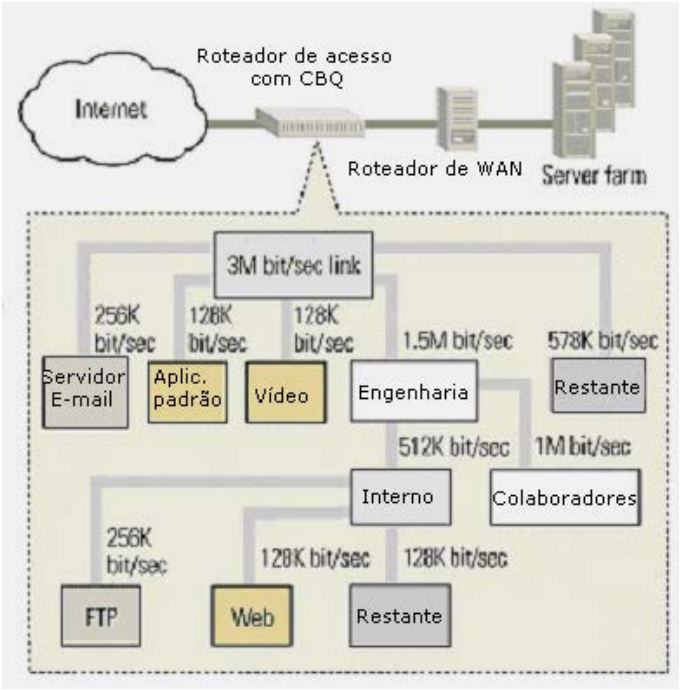


Figura 07 – Classes hierárquica para controle de banda[6]

O algoritmo do CBQ faz utilização de algumas variáveis de controle, são elas:

- *Idle* – A variável *idle* é a diferença entre o tempo desejado e o tempo medido entre os pacotes mais recentes. Quando a conexão está mandando mais tráfego que a banda alocada o *idle* tem valor negativo quando a conexão está mandando de forma perfeita o valor da variável é zero.

- *Avgidle*: esta variável representa a média da variável *idle*, quando *avgidle* é zero ou negativa a classes está com limite ultrapassado

- *Offtime* : este parâmetro informa o tempo que o limite ultrapassado deve esperar para poder enviar outro pacote.3.[29]

5.3.3. TC (Traffic Control)

TC é uma ferramenta que vem normalmente acoplada ao *kernel* do linux, consistindo das seguintes características: modelagem, programação.

- **Modelagem**

A modelagem ocorre na saída de informações (*upload*) da rede servindo para: controlar o tráfego; reduzir a largura da banda disponível e também para diminuir o congestionamento de tráfico quando há uma grande quantidade de usuários.

- **Policiamento**

Policiar possui a mesma função da modelagem só que este ocorre na entrada de informações.

- **Descarte**

O tráfego excedente em uma banda pode ser descartado preventivamente, tanto na saída como na entrada. Processamento de tráfego é controlado por três tipos de objetos: qdiscs, classes e *filters.*

- **Qdiscs**

O *kernel* do linux oferece diversos recursos que permitem o total controle no envio de pacotes. Este conjunto de recursos é denominado *Queueing Discipline* (qdisc) - são regras de enfileiramento.

Quando o núcleo do sistema operacional necessita enviar um pacote a uma interface, este é enfileirado para o qdisc que está configurado naquela interface. Imediatamente após, o *kernel* tenta pegar o maior número de pacotes possíveis do qdisc para entregar ao adaptador de rede.

A mais simples regra do qdisc é um pfifo (*First in First Out*), pois ela não consome alto poder computacional, é simplesmente uma fila em que o primeiro pacote que entra é o primeiro que sai, contudo pode armazenar tráfego quando a interface de rede não estiver disponível por curtos períodos de tempo.

- **Qdiscs com Classes**

Alguns qdiscs podem conter classes, estas contêm outros qdiscs. Tráfego pode ser enfileirado em qualquer um destes qdiscs. Quando o *kernel* tenta retirar os pacotes do enfileiramento onde existem classes configuradas o qdisc pode priorizar certos tipos de tráfego baixando primeiramente os pacotes de classes especificas.

- **Filtros**

O filtro é utilizado pela classe para determinar a que classe um pacote pertence. Quando o tráfego chega a uma classe que tem várias subclasses, é necessário classificar este tráfego. Vários métodos podem ser aplicados para realizar esta tarefa, um destes métodos é o filtro. Todos os filtros que acompanham as classes são chamados até que um retorne uma informação verdadeira, caso não exista nenhum filtro para o tipo de tráfego pode-se adotar outros tipos de critérios isto muda de acordo com o qdisc.

- **Qdiscs sem Classes**

[p|b]fifo:
Mais simples de todos, primeiro a entrar na fila é o primeiro a sair.
pfifo_fast:
Qdisc padrão para roteadores avançados. Consiste em três enfileiramentos que funcionam a partir de *flags* que marcam a prioridade de cada pacote.
Red:

Detecção Aleatória Antecipada (*Random Early Detection*) simula congestionamentos descartando os pacotes randomicamente quando estiverem perto da banda alocada que foi configurada. Funciona bem para bandas de alta velocidade.

Sfq:

Stochastic Fairness Queueing reordena o enfileiramento do tráfego para que cada sessão mande um pacote por vez.

Tbf:

Filtro por balde de *token* (*Token Bucket Filter*) reduz a velocidade do tráfego para uma taxa pré-configurada. Funciona bem para rede de alta velocidade.

Qdiscs sem classes podem apenas serem anexados para a raiz da interface, pfifo_fast é o padrão no caso de não ter sido configurado nada.

- **Classes Para o Qdisc**

  - **CBQ**

Implementa o compartilhamento de link hierarquizado de classes. Contém elementos de modelagem e capacidade de priorizar tráfego. A modelagem é feita utilizando cálculos de tempo baseados no tamanho do pacote.

  - **HTB**

Implementa o compartilhamento de link hierarquizado de classes. Facilita a garantia de banda para classes, enquanto permite o empréstimo de banda entre classes. Contém elementos de modelagem é baseado no TBF e pode priorizar classes.

- **PRIO**

Não utiliza modelagem, permite configurar várias classes que são analisadas em ordem, isto proporciona uma maior facilidade na priorização do tráfego, onde classes mais baixas só podem mandar dados se as mais altas já tiverem enviado.

- **Funcionamento**

As classes formam uma árvore, onde cada classe tem um único pai. Uma classe pode ter múltiplos filhos. Alguns qdiscs permitem adição de classes em tempo de execução (CBQ, HTB), enquanto outros (PRIO) são criados com um numero fixo de filhos.

Qdiscs que permite a adição dinâmica de classes podem não ter nenhuma classe como ter várias subclasses em quais o tráfego vai ser enfileirado. Além disso, cada classe contém uma folha que por padrão tem o comportamento do pfifo, esta folha pode ter outras classes, mas uma classe pode ter apenas uma folha.

Quando um Pacote entra em um qdisc pode ser classificado para uma das classes por três tipos de critérios, mas nem todos os qdiscs implementam os três: tc *filters, type of service*, skb->priority

- ***tc filters***

Se um filtro é anexado a uma classe, ele é consultado primeiro para instruções que devem ser tomadas. Os Filtros podem comparar todos os campos nos cabeçalhos dos pacotes, e também por marcações realizadas pelo *firewall*, *iptables* e *ipchains.*

- ***Type of Service***

Alguns qdiscs tem regras internas para classificar pacotes baseados no campo tipo de serviço.

- **skb->*priority***

Aplicações de usuários podem codificar um identificador no campo skb->priority usando a opção SO_PRIORITY. Cada nó na árvore pode ter seus próprios filtros mas filtros com níveis mais altos podem apontar para diretamente para classes mais baixas.[7]

5.3 Ferramentas de Monitoramento

5.3.1 MRTG

O MRTG [39] é software livre distribuído nos termos da *GNU General Public License* [8]. Esta não é exatamente uma ferramenta que controla o tráfego, e sim, uma ferramenta que realiza o monitoramento da rede. A principal função desta ferramenta é gerar gráficos relativos à utilização de diversos dispositivos.

Como esta ferramenta, contempla geração de gráficos relativos à utilização de portas de comunicação de dados, ela passa a ser um item indispensável no gerenciamento do controle de banda. É através deste tipo de ferramenta que são obtidos gráficos do comportamento de uma rede com o auxílio de outras ferramentas que se têm informações de quais serviços, usuários e portas estão com maior utilização do link. A partir destas informações gráficas, é possível decidir quais regras deverão ser inseridas no controlador de banda.

Além da monitoração da rede, é possível, também, medir taxar de erro no link que está sendo monitorado. Isto é útil, pois, ás vezes, tem-se uma falsa impressão de que a rede está com alto tráfego, devido à lentidão nas transferências. O fato é que nem sempre isto é verdade, pode ocorrer uma alta taxa de erro nas informações que atravessam este link de acesso, fazendo com que várias retransmissões sejam realizadas, dando uma falsa impressão de alta utilização da banda.

O MRTG gera páginas em html, disponibilizando os gráficos alem de outras informações pertinentes a estes gráficos.

Características:

- “Mede sempre dois valores, no caso de tráfego, pode ser Entrada e Saída”.
- Faz as leituras via SNMP ou através de *script* que retorne um formato padrão.
- Coleta dados a cada 5 minutos por padrão, mas este tempo pode ser alterado para mais.
- Cria página HTML com quatro gráficos (diário, semanal, mensal e anual). Se algum deles não for necessário pode ser suprimido.
- O MRTG pode avisar caso o valor do gráfico atinja um valor preestabelecido. Por exemplo: se determinado servidor atinge 95% do espaço do disco, o MRTG pode mandar um e-mail para o administrador informando o ocorrido.
- Possui uma ferramenta para gerar os arquivos de configuração: o CFGMAKER.
- Possui uma ferramenta para gerar um página de índice para os casos em que muitos itens são monitorados: o INDEXMAKER.”[5]

As figuras 08,09 e 10 , mostram como é visualizado um gráfico gerado no MRTG.

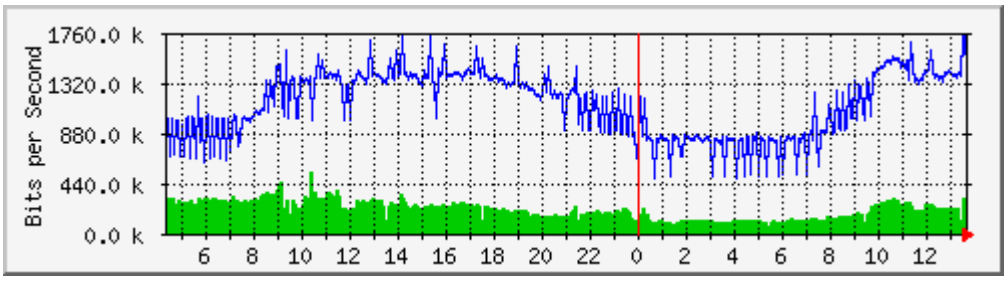


Figura 08 – Exemplo de gráfico gerado pelo MRTG

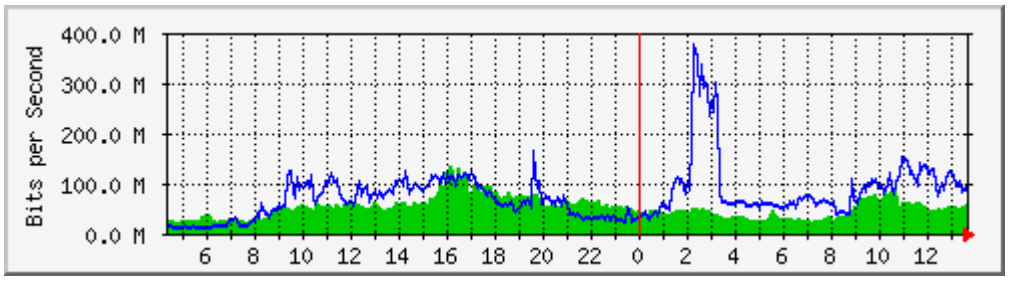


Figura 09 – Exemplo de gráfico gerado pelo MRTG

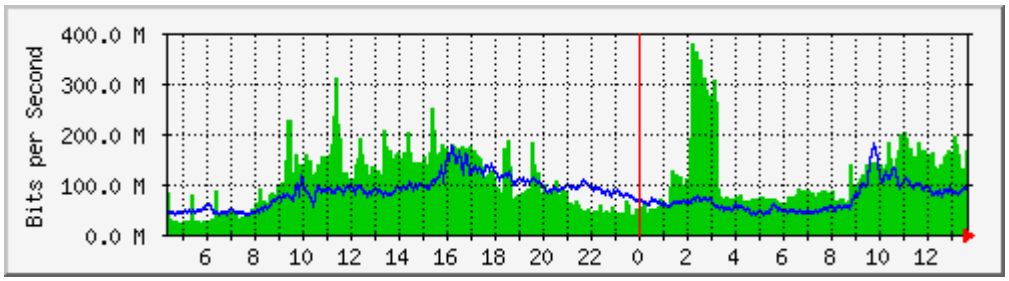


Figura 10 – Exemplo de gráfico gerado pelo MRTG

5.3.2 Iperf

Depois de inúmeros testes realizados com ferramentas de geração de tráfego chegamos a conclusão de que o Iperf seria a melhor solução para o nosso caso, pois alem de gerar o tráfego em qualquer tipo de porta com protocolo TCP e UDP, permitir a configuração do tamanho do arquivo a ser enviado pela rede ele informa ao final do envio qual a velocidade de transmissão que este arquivo foi enviado.

O Iperf [39] é um software mantido pela Universidade de Illinois[9] nos Estados Unidos sob licença GPL (*General Public License*)[8]. Sua utilização e realizada em linhas de comando. Através desta ferramenta é possível uma simulação de tráfego na rede, ou seja, ele é capaz de inserir na rede uma quantidade enorme de pacotes fazendo com que a banda de utilização de uma rede seja testada. Em vários casos são contratados acessos de dados em que neles, geralmente, não são realizados testes de sua taxa real de utilização. Caso exista problema na largura de banda contratada (não corresponder à taxa real adquirida), o usuário tomará conhecimento do problema no momento em que o link contratado chegar a 100% de utilização. Neste

momento será notada uma lentidão no acesso as informações remotas, logo será notado que o link não atingiu o velocidade contratada. Porém, a altura do processo, vários meses se passaram sem que a banda contratada estivesse sido disponibilizada.

Para que inconvenientes como estes não aconteçam em um ambiente de rede, é importante o uso de um gerador de tráfego, para que seja confirmada a largura de banda contratada.

Para o projeto da interface do controlador de banda abordado neste documento, o gerador de tráfego foi de grande importância para que diversas ferramentas de controle de controle de banda fossem testadas e que resultou na escolha do controlador de banda HTB.

O Iperf é uma ferramenta *free* e de código aberto. Ela é capaz de gerar tráfegos tanto UDP como TCP. Algumas características e comandos são descritos a seguir:

- **-d --dualtest** Teste bidirecional simultaneamente
- **-n --num** Numero de bytes para transmissão do teste
- **-r --tradeoff** Teste bidirecional executado individualmente
- **-t --time** Tempo em segundos da transmissão (10 segundos o padrão)
- **-T --ttl** Tempo de vida para MultiCast (padrão 1)
- **-F --fileinput <arquivo>** Entrada de dados para transmissão por arquivo
- **-L --listenport** Entrada de dados para transmissão por stdin
- **-P --parallel** Numero de cliente para execução em paralelo
- **-s ou --server** Roda em modo servidor
- **-U --single-udp** Roda em modo único usando UDP
- - **D --daemon** Roda o servidor como Daemon "serviço"

Instalação do Iperf

Na instalação do Iperf, foram utilizadas máquinas com:

- SO: Debian Gnu/Linux / Kernel 2.6.2

- Iperf 2.0.1
- Dispositivo de Rede: Realtek 10/100/1000 Fast Ethernet (Chipset RTL-8169)

O Iperf pode ser baixado no seguinte endereço: http://dast.nlanr.net/Projects/iperf/

Após feito ter feito o *download* do arquivo, será necessário que este seja descompactado e compilado. A seguir são mostrados o passos para que estas tarefas sejam realizadas com êxito:

```
#tar -zxvf iperf-2.0.1.tar.gz
#cd iperf-2.0.1
#./configure ; make ; make install
```

A porta padrão do Iperf é a 5001, logo é importante certificar se o *firewall* não está bloqueando esta porta. Porém se não for interessante desbloquear a porta no *firewall*, outra poderá ser escolhida utilizado para isso o parâmetro –p. Supondo que a porta esteja desbloqueada, o servidor será executado da seguinte maneira:

```
# iperf -s
```

Para este exemplo, será suposto que o endereço IP da máquina onde está executado o servidor seja 10.10.10.1 e do cliente seja 10.10.10.2. O seguinte comando deverá ser executado em linha de comando:

```
# iperf -c 10.10.10.1
```

Como resultado destes procedimentos terá na console da máquina cliente as seguintes informações:

```
maquinaCliente:~# iperf -c 192.168.0.1
 ------------------------------------------------------------
Client connecting to 192.168.0.1, TCP port 5001
TCP window size: 128 KByte (default)
------------------------------------------------------------
```

5 local 10.10.10.2 port 32926 connected with 10.10.10.1 port 5001
5 0.0-10.0 sec 1.86 MBytes 96.1 Mbits/sec

Com estes resultados poderá concluir que a taxa de transmissão de dados para este caso foi de 96.1 Mbits/sec.

6. HTB (*Hierarchy Token Bucket*)

HTB[31] é uma ferramenta mais nova, intuitiva e rápida que o CBQ, as duas ferramentas ajudam a controlar o tráfego de saída de um determinado link.

6.1. Características

- Realiza o controle da saída de uma determinada interface;
- HTB está na versão 3, sua versão mais antiga está disponível o código fonte HTB versão 2;
- Sua configuração é realizada por linhas de comando;
- O HTB tem por finalidade dar continuidade e melhorar o CBQ. Ambos têm como base de funcionamento o TC (*traffic control*) do Linux.

6.2.Requisitos de Software

O HTB necessita que alguns pacotes de *software* estejam pré-instalados. Com o sistema operacional utilizado neste trabalho, Debian Sarge 3.1 kernel 2.6.2, foi necessário fazer a instalação do IpRoute. A instalação deste pacote é detalhada no ANEXO E.

6.3.Compartilhamento de link

Com o HTB é necessário ter criado uma classe denominada de “parente”, esta classe parente se refere à uma regra “principal”. Abaixo desta, de forma hierárquica, são criadas subclasses. Se algumas destas subclasses não estiverem utilizando toda a banda da classe pai então outras classes poderão “pegar emprestado” a banda que está sobrando, este processo e denominado de *borrowing.* É bom lembrar que classes parente não podem pegar emprestado umas das outras.

O total de serviço provido para cada classe é no mínimo o total requerido por esta e conforme o que foi designado para cada classe ou seja se você tem um mínimo de

banda requerido para uma classe, sempre receberá este mínimo antes de emprestar banda para alguém.

Exemplo de Regra Principal

tc qdisc add dev eth0 root handle 1: htb default 12

tc qdisc add dev eth0 : define qual a placa de rede está sendo referenciada e cria a regra como principal.

root : parâmetro para definição da classe principal da placa eth0.

Handle: 1: é o mesmo que 1:0 quer dizer que esta é a regra 1 da placa eth0.

Htb default 12: diz que se existir algum tipo de tráfego que não esta definido em nenhum filtro ele irá tomar como default a classe 1:12.

Exemplo de classe parente:

tc class add dev eth0 parent 1: classid 1:1 htb rate 512kbps ceil 512kbps

tc class add dev eth0: define que a classe se refere a placa eth0.

Parente 1: : define que a regra 1:0 é sua classe Pai (raiz).

Classid 1:1 : numeração da classe parente.

Htb rate 512kbit : esta regra informa qual o link garantido para esta placa neste caso 512 Kilo bites..

Ceil 512kbit : esta regra informa qual o máximo que esta classe pode atingir.

Exemplos de Sub-classes Parente (filhas):

tc class add dev eth0 parent 1:1 classid 1:10 htb rate 200kbit ceil 512kbit
tc class add dev eth0 parent 1:1 classid 1:20 htb rate 300kbit ceil 512kbit

Estas regras acima são bem parecidas com o exemplo da regra parente a diferença é que no parâmetro *parent* está 1:1 informando que esta classe é uma sub-parente da anterior.

As próximas regras são responsáveis pelo controle de banda efetivamente, que são chamados de filtros, eles que analisam o tráfego e sabem qual regra devem aplicar para cada uma das classes.

Exemplo de Filtros:

*tc filter add dev eth0 protocol ip parent 1:0 prio 1 u32 *
match ip src 10.10.10.4 match ip dport 80 0xffff flowid 1:10

*tc filter add dev eth0 protocol ip parent 1:0 prio 1 u32 *
match ip src 10.10.10.4 flowid 1:20

tc filter add dev eth0: define que o filtro irá para a placa eth0.
Protocol ip parente 1:0 : define que a classe raiz é a 1:0 na interface eth0.
Prio 1 u32 \ : define que tipo de prioridade este filtro tem sobre os outros.
Match ip src 10.10.10.4 : a regra será ativada em cima do IP que esta enviando o pacote.
match ip dport 80 0xffff : sobre qual porta estará sendo aplicado o filtro.
Flowid 1:10 : Por qual classe estará executando.

Quando muitas classes querem tomar emprestado largura de banda, a classe pai fornece a cada uma delas um determinado número de bytes antes de servir as outras classes concorrentes. Este número é denominado cota. Não é necessário especificar cotas manualmente porque HTB escolhe valores pré-computados. A figura 11 ilustra como é hierarquizada esta árvore.

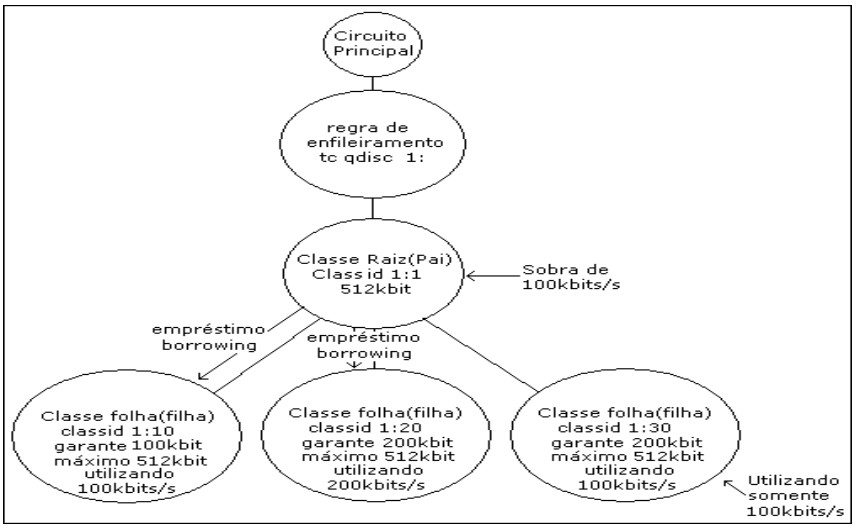


Figura 11 – Exemplo da Hierarquia da Árvore HTB

6.4. Priorizando a largura de banda compartilhada

Priorizar tráfego apresenta dois aspectos. O primeiro está relacionado como o excedente de largura de banda que é distribuído entre os tipos de serviços. E o segundo pertence as prioridades das classes, a que tiver uma maior prioridade receberá primeiro a oferta de excedente de banda.

Para este exemplo, será suposto que existem três classes concorrendo uma banda de 512kbits/s conforme é mostrado na figura 12. Para a classe 1:10 é garantida uma banda de 100kbits/s com limite de 512kbits/s e prioridade 1, para a 1:20 200kbits/s com limite de 512kbits/s com prioridade 2 e para 1:30 200kbits/s com limite de 512kbits/s com prioridade 3. Em determinado momento, as classes 1:10 e 1:20 estão utilizando o máximo de suas bandas garantidas, ou seja, 100kbits/s para a 1:10 e 200kbits/s para a 1:20, porém a 1:30 está utilizando somente 50% de sua banda garantida (100kbits/s). Neste caso é tido uma sobra de recurso de banda de 100kbits/s. Este recurso disponível será entregue a classe com maior prioridade no caso a 1:10 e o restante para as de menores prioridades, neste caso a 1:20.

Geralmente são priorizadas classes que não podem ter muitos atrasos e que sejam de maior importância, por exemplo no tráfego de áudio e vídeo, normalmente precisam estar priorizados para evitar atrasos na transmissão evitando má qualidade nestas informações de multimídia.

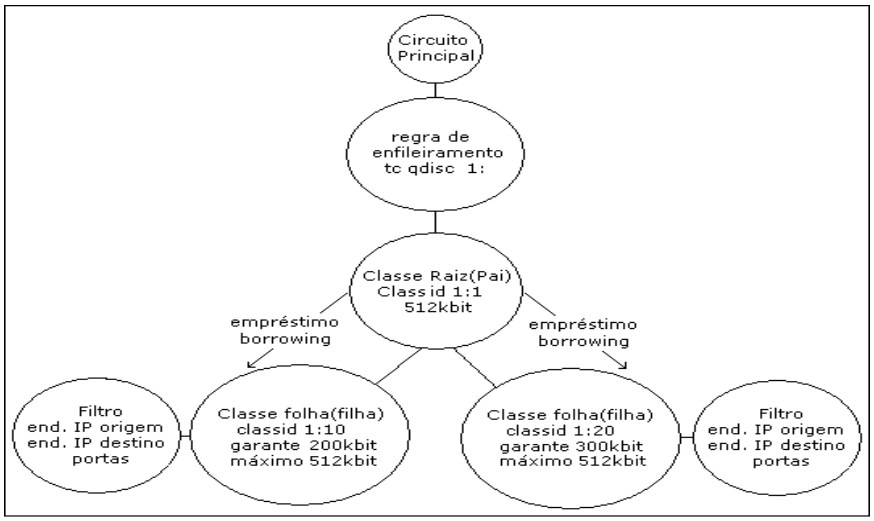


Figura 12 – Concorrência de banda e empréstimo de banda

É importante ser observado que apesar deste processo de utilização de sobra de banda, não utilizada por outras classes, ser chamado de empréstimo, este não se assemelha a um empréstimo propriamente dito, pois não há devolução da banda que foi dita emprestada. Este termo foi utilizado nesta documentação para preservar o utilizado pelo desenvolvedor do HTB.

7. Outras Aplicações utilizadas

7.1.Tomcat

O Tomcat [40] é um servidor que possibilita rodar aplicações JAVA para web. É um software livre e *OpenSource* foi desenvolvido pela Apache Jakarta e oficialmente endossado pela Sun. Utiliza tecnologias *Java Servlet e JavaServer Pages* (JSP).[10]

O Tomcat pode ser utilizado como servidor web/HTTP, ou pode funcionar integrado a um servidor web dedicado como o Apache httpd ou o Microsoft IIS. A versão utilizada neste projeto é a 5.5, ele servirá para realizar a comunicação entre o usuário final e o HTB, esta comunicação será feita através da linguagem de programação JSP.

7.2.JSP (Java Servers Page)

A tecnologia JSP permite que desenvolvedores de Web e designers, desenvolvam rapidamente e consigam manter facilmente páginas dinâmicas para Web sendo independente de plataforma, permite que desenvolvedores alterem o layout da página sem terem muita dificuldade.

JSP utiliza XML com *tags* que são encapsulados gerando o conteúdo para a página. A lógica da aplicação pode permanecer no servidor o usuário final terá apenas acesso ao código final encapsulado, sendo assim separando a parte lógica do design. [11]

7.3. Netbeans IDE

NetBeans[12] é um projeto *OpenSource* dedicado a prover um software seguro que visa suprir as necessidades dos desenvolvedores, usuários e em negócios

O NetBeans IDE, permite o desenvolvimento sobre a plataforma J2EE, ele suporta a criação de aplicações JAVA, Web, tecnologias móveis/sem fio. O *open source*

Java IDE, tem tudo que um desenvolvedor necessita. Nele já vem acoplado uma série de outras funcionalidades como *plug-ins*, debug, alem de funcionar em múltiplas plataformas. Algumas das funcionalidades da versão mais nova do NetBeans IDE.[12]

- Componente de navegação através das classes.
- Permite configurar configure o CLASSPATH do projeto usando usas Bibliotecas disponíveis.
- Permite utilizar o gerador automático para “Debugar” e Compilar arquivos com scripts existentes.
- Permite configurar o projeto para ter múltiplas fontes de root.
- Permite criar Registros e teste serviços para Web.
- Possui serviços Web e Componentes Web.
- Permite a validação de aplicações utilizando o verificador J2EE.

7.4. C++, C (GCC)

Linguagem de programação nativa do Linux, que permitirá a criação de scripts para fazer a comunicação entre o JSP e o Linux.

O GCC (*GNU C Compiler*) criado por Richard StallMan na universidade de *Berkley*[36] – Estados Unidos. É um compilador de alta qualidade e de bastante portabilidade para as linguagens C, C++ e C orientado a objeto. O compilador foi desenvolvido para suportar várias plataformas sendo que primeiramente ele traduz o código para uma linguagem de Registro e depois para a linguagem *assembler* especifica para aquela arquitetura. [13]

8. Aplicação desenvolvida

8.1. Casos de uso

Na tabela 03 consta a lista de todos os casos de uso utilizados no desenvolvimento da interface gráfica

Caso de uso	*Nome*
01	Acessar Programa
02	Verificar Placas
03	Cadastrar Regra Geral
04	Cadastrar Regra Principal
05	Cadastrar Sub-Regra
06	Cadastrar Filtro
07	Executar as regras pré-definidas no HTB
08	Alterar regras previamente cadastradas
09	Deletar
10	Desativar

A figura 13 mostra o diagrama de caso de uso utilizado no desenvolvimento

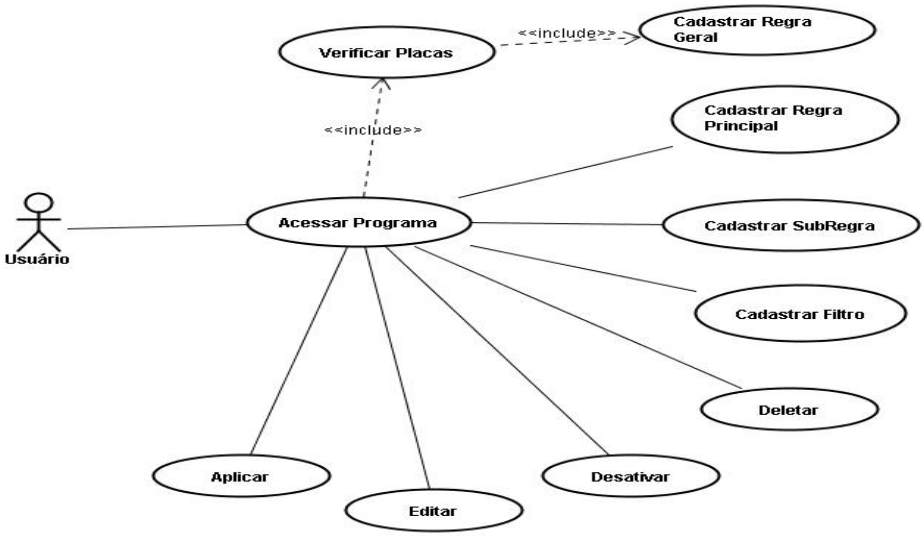


Figura 13 – casos de uso utilizados no desenvolvimento

8.1.1. UC01_Acessar_Programa

Nome: Acessar Programa.

Objetivo: Acessar a página inicial na WEB do programa de controle de banda

Atores: Usuário.

Prioridade: Alta.

Pré-condições: servidor Tomcat deve estar funcionando.

Fluxo Principal:

P1) Esse caso de uso começa quando o usuário acessa, via WEB, a página de nome "princ.jsp" no servidor em que esta estiver localizada.
P2) O usuário irá visualizar uma janela em que mostrará todas as regras que estão cadastradas para o funcionamento do HTB.
P3) O usuário terá a opção de selecionar várias opções se este clicar em "APLICAR" o executa A1.
P4) Caso o usuário clique em "EDITAR", executa A2.
P5) Caso o usuário clique em "DELETAR", executa A3.
P6) Caso o usuário clique em "DESATIVAR", executa A4.
P7) Caso o usuário clique em "CADASTAR", o sistema o redirecionará para os casos de uso UC04, ou UC05, ou UC06, finalizando o caso do uso.

Fluxo Alternativo (A1):

A1.1) O sistema irá executar as regras descritas acima, fazendo com que o HTB entre em funcionamento, para isso executará o caso de uso UC07.

Fluxo Alternativo (A2):

A2.1) O sistema irá redirecionar o usuário para o caso de uso UC08.

Fluxo Alternativo (A3):

A3.1) O sistema irá redirecionar o usuário para o caso de uso UC09.

Fluxo Alternativo (A4):

A4.1) O sistema irá redirecionar o usuário para o caso de uso UC10

Protótipo Visual (PTV1): Figura 14

Figura 14 – Protótipo visual

8.1.2. UC02_Verificar_Placas

Nome: Verificar Placas.

Objetivo: Verificar as placas de rede que estão ativas no Linux.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC01.

Fluxo Principal:

P1) Esse caso de uso inicia quando o site principal é acessado, automaticamente ele irá realizar uma busca nas placas que estão ativas no sistema.
P2) O sistema realiza a execução de um script de nome “inter” este script irá criar um arquivo de nome “interfaces”, este conterá informações de periféricos do sistema.
P3) Logo após o sistema realizará a execução do sub-programa “buscap”, este irá varrer o arquivo “interfaces” em busca das placas que estão ativas.
P4) O sistema grava as placas encontradas e um arquivo chamado “placas” encerrando assim o caso de uso.

Não possui fluxo alternativo.

Não possui protótipo visual

8.1.3. UC03_Cadastrar_Regra_Geral

Nome: Cadastrar Regra Geral

Objetivo: Através do arquivo “placas” cadastrar as regras principais do HTB.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC02.

Fluxo Principal:

P1) Esse caso de uso começa quando UC02 termina.
P2) O sistema que está executando o sub-programa “buscap” que cria regras gerais do HTB para cada uma das placas encontradas.
P3) As regras gerais são gravadas em um arquivo denominado “regrapai”.

Não possui fluxo alternativo.

Não possui protótipo visual

8.1.4. UC04_Cadastrar_Regra_Principal

Nome: Cadastrar Regra Principal

Objetivo: Cadastrar as regras parentes do HTB.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC03.

Fluxo Principal:

P1) Esse caso de uso inicia quando o usuário clica no item “CADASTAR” referente às regras principais.

P2) O sistema irá redirecionar o usuário para a página WEB “regrapar.jsp”.

P3) Nesta página o usuário poderá escolher qual a placa se refere o tráfego e qual a velocidade do seu link.

P4) O usuário, após estes passos, irá clicar no botão “CADASTRAR” e o sistema irá gravar os dados em um arquivo temporário de nome “tparent”.

P5) O sistema irá executar o subprograma “criaregras”, este irá varrer o arquivo temporário e irá realizar a gravação destes dados no arquivo denominado “parent”.

P6) Se tudo ocorrer sem problemas o sistema irá redirecionar o usuário de volta para a página principal encerrando o caso de uso.

Não possui fluxo alternativo.

Protótipo Visual (PTV1): Figura 15

Figura 15 - Protótipo Visual do UC04

8.1.5. UC05_Cadastrar_Sub-Regra

Nome: Cadastrar Sub-Regra

Objetivo: Cadastrar as sub-regras que estão relacionadas com as regras parente do HTB.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC03.

Fluxo Principal:

P1) Esse caso de uso inicia após o usuário clicar no item “CADASTAR”, referente às sub-regras.
P2) O sistema irá redirecionar o usuário para a página WEB “regrasub.jsp”.
P3) Nesta página o usuário poderá escolher qual a placa se refere ao tráfego e qual a velocidade garantida do link e qual a velocidade máxima que este link pode atingir.
P4) O usuário , após estes passos, irá clicar no botão “CADASTRAR” e o sistema irá gravar os dados em um arquivo temporário de nome “tchild”, caso o usuário coloque banda maior do que a comportada pela regra principal, executa A1.
P5) O sistema irá executar o subprograma “criaregras2”, este irá varrer o arquivo temporário e irá realizar a gravação destes dados no arquivo denominado “child”.
P6) Se tudo ocorrer sem problemas o sistema irá redirecionar o usuário de volta para a página principal encerrando o caso de uso.

Fluxo Alternativo (A1):

A1.1) O sistema irá reapresentar a página “regrasub.jsp” mostrando a mensagem de erro “Regra ultrapassa o limite de banda definido a regra principal selecionada”.

Protótipo Visual (PTV1): Figura 15

Figura 16 - Protótipo Visual do UC05

8.1.6. UC06_Cadastrar_Filtro

Nome: Cadastrar Filtro

Objetivo: Cadastrar os filtros que serão referentes às sub-regras.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC03.

Fluxo Principal:

P1) Esse caso de uso inicia quando o usuário clica no item “CADASTAR” referente às regras filtro.
P2) O sistema irá redirecionar o usuário para a página WEB “filtros.jsp”.
P3) Nesta página o usuário poderá escolher qual a placa se refere ao tráfego e qual o IP ou rede que entrará neste filtro, qual a porta e qual a sub-regra ele se refere.
P4) O usuário, após estes passos, irá clicar no botão “CADASTRAR” e o sistema irá gravar os dados em um arquivo temporário de nome “tfilter”.
P5) O sistema irá executar o subprograma “criafiltros”, este irá varrer o arquivo temporário e irá realizar a gravação destes dados no arquivo denominado “filter”, caso o usuário coloque o valor 0 (zero) como porta, executa A1.
P6) Se tudo ocorrer sem problemas o sistema irá redirecionar o usuário de volta para a página principal encerrando o caso de uso.

Fluxo Alternativo (A1):

A1.1) O sistema irá executar o subprograma “criafiltros2”, este irá varrer o arquivo temporário e irá realizar a gravação destes dados no arquivo denominado “*filter*”.

Protótipo Visual (PTV1): Figura 17

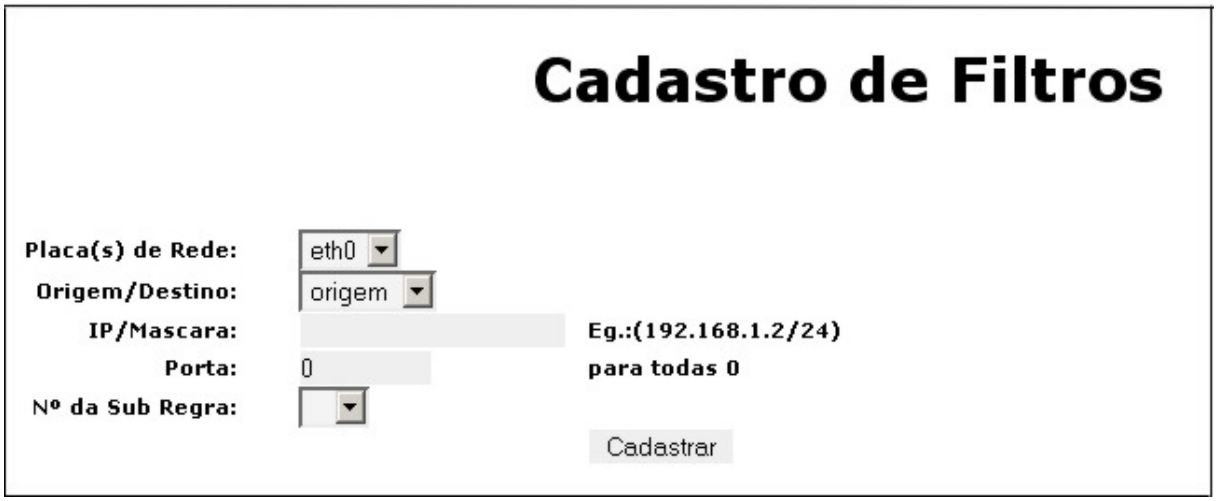


Figura 17 - Protótipo Visual do UC06

8.1.7. UC07_Aplicar

Nome: Aplicar.

Objetivo: Executar as regras pré-definidas no HTB.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC04, UC05, UC06 pelo menos uma vez.

Fluxo Principal:

P1) Esse caso de uso inicia quando o usuário clica no item “APLICAR” na página principal.
P2) O sistema irá redirecionar o usuário para a página WEB “grr.jsp”.
P3) Nesta página o sistema irá executar o arquivo de nome “roda” este arquivo irá executar os arquivos “regrapai”, “parent”, “child” e “filter”, desta forma colocando em execução o controle de banda.
P4) Se tudo ocorrer sem problemas, o sistema irá redirecionar o usuário de volta para a página principal encerrando o caso de uso.

Não existe Fluxo Alternativo

Não existe Protótipo Visual

8.1.8. UC08_Editar (Não Implementado)

Nome: Editar.

Objetivo: Alterar regras previamente cadastradas.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC04, UC05, UC06 pelo menos um deles.

Fluxo Principal:

P1) Esse caso de uso inicia quando o usuário clica no item “Editar” na página principal.
P2) O sistema irá redirecionar o usuário para a página WEB referente ao tipo de regra que ele esta editando, caso seja regra parente ele ira para “regrapar.jsp”.
P3) O usuário visualizará a página igual a de cadastro mas os campos já estarão preenchidos.
P4) Se tudo ocorrer sem problemas o sistema irá redirecionar o usuário de volta para a página principal encerrando o caso de uso.

Não existe Fluxo Alternativo

Não existe Protótipo Visual

8.1.9. UC09_Deletar

Nome: Deletar.

Objetivo: Remover uma regra especifica.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC04, UC05, UC06 pelo menos um deles.

Fluxo Principal:

P1) Esse caso de uso inicia quando o usuário clica no item “Deletar” na página principal.
P2) O sistema irá fazer um pedido de confirmação ao usuário.
P3) Se o usuário clicar em OK o sistema irá deletar a regra em que o usuário clicou, sendo que se a regra for uma regra parente, irá deletar todas as regras referentes a ela, caso o usuário clique em “Cancelar”, executa A1.
P4) Se tudo ocorrer bem o sistema irá realizar um *reload* na página principal encerrando o caso de uso.

Fluxo Alternativo (A1):

A1.1) O sistema irá ignorar o pedido que o usuário fez.

Não existe Protótipo Visual

8.1.10. UC10_Desativar

Nome: Desativar.

Objetivo: Desativar o controle de banda.

Atores: Usuário.

Prioridade: Alta.

Pré-condições: realizar o UC01.

Fluxo Principal:

P1) Esse caso de uso inicia quando o usuário clica no item “DESATIVAR” na página principal.
P2) O sistema irá fazer um pedido de confirmação ao usuário.
P3) Se o usuário clicar em OK o sistema irá desativar o controle de banda deixando o sistema livre sem nenhuma regra implícita nele, caso o usuário clique em “Cancelar”, executa A1.
P4) Se tudo ocorrer sem problemas o sistema irá realizar um *reload* na página principal encerrando o caso de uso.

Fluxo Alternativo (A1):

A1.1) O sistema irá ignorar o pedido que o usuário fez.

Não existe Protótipo Visual

8.2. Funcionamento do Software

Para facilitar o entendimento segue significado de alguns termos:

- Regras ROOT: são regras raiz (pai) que dá origem a outras regras chamadas de regras parentes principal;
- Regras parente principal: são regras/classes que darão origem as regras/classes filhas propriamente ditas, chamadas de regras parentes;
- Regra parente (regras filhas): São as regras/classes que estão nas folhas da árvore hierárquica do HTB.
- Velocidade Garantida: é a velocidade que não importa qual seja o tráfego da rede, esta velocidade deve ser mantida para um determinado tipo de tráfego.
- Velocidade Máxima: é a velocidade mais alta que determinado tráfego pode atingir.

A figura 18 exemplifica como é a hierarquia da árvore HTB [31].

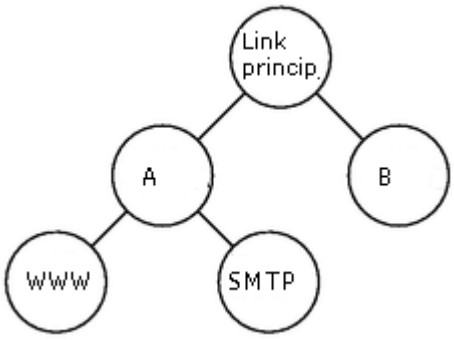


Figura 18 - Árvore hierárquica do HTB [31]

Primeiramente irá entrar na página inicial do Software, será permitido a escolha de criação de classe. Quando a página principal é acessada, automaticamente é lido quais as placas existentes no computador e logo são criadas as regras ROOT (principais) para o funcionamento do HTB.

8.2.1. Cadastro de regras Parentes (Classes)

A criação de uma regra parente é necessária para que se saiba qual o tamanho do link a ser controlado. Será suposto que se tenha um link de 100Mbits e é necessário controlar 10Mbits destes 100Mbits, isto irá ser informado nesta regra na forma de kbits, portanto para informar uma regra "parente" de 10 megabits é necessário informar o tamanho de 10000Kbits.

Para se cadastrar uma regra "parente" é necessário acessar o arquivo "regrapar.jsp" no servidor, este arquivo pode ser acessado pela página principal clicando no botão cadastrar referente a regras principais. A figura 19 mostra como isto é feito pela interface.

Depois de o arquivo ter sido acessado o usuário deve informar em que Placa de Rede ocorrerá o controle de banda, como no HTB o controle normalmente ocorre na saída do tráfego é necessário averiguar exatamente qual será a placa que será selecionada. Logo após a seleção da placa o usuário deve informar a velocidade total do link, este link total poderá ser subdividido mais á frente.

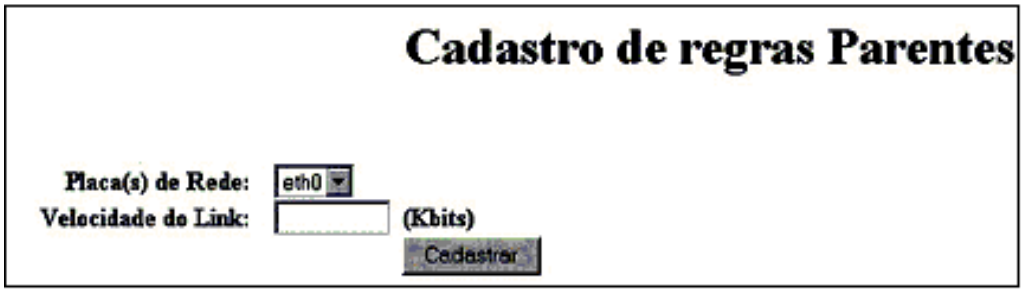


Figura 19 - Cadastro de regras parentes

BufferedReader in2 = new BufferedReader(new FileReader("//home//leo//Projeto//tparent"));

O comando *BufferedReader* Realiza a leitura de um arquivo temporário(tparent) para verificar quais regras já foram criadas, para assim poder dar continuidade na numeração.

```
Writer escreve = new BufferedWriter(new
FileWriter("//home//leo//Projeto//tparent"));
```

Depois que foi verificado qual o próximo número na seqüência a função *BufferedWriter* irá escrever neste mesmo arquivo temporário qual a placa que se destina o controle, qual o número da regra a ser criada e qual a velocidade ficando o arquivo formatado da seguinte forma:

```
Eth0
1
600
Process r =
Runtime.getRuntime().exec("//home//leo//Projeto//criaregras");
r.waitFor();
```

Logo após a criação do arquivo temporário o JSP irá chamar o programa *criaregras,* este programa realiza a criação da regra parente. Ele irá ler o arquivo "tparent" e a partir dos dados encontrados neste arquivo irá criar outro arquivo denominado "parent" neste arquivo irá conter todos os scripts, por exemplo, (tc class add dev eth0 parent 1: classid 1:1 htb rate 600kbit ceil 600kbit).

8.2.2. Cadastro de Sub-Regras (Classes)

É necessário criar as sub-regras para dividir o link especificado na regra anterior. Por exemplo, se a regra anterior for configurada para um link de 10Mbits, pode-se criar inúmeras regras para dividir este link em sub-links de menor capacidade. Ainda fornecer taxas de garantia de tráfego bem como uma taxa máxima que este link poderá atingir.

Para criar uma sub-regra pertencente a uma regra "parente" é necessário acessar o arquivo regrassub.jsp, no servidor, este acesso pode ser realizado através da página principal do software e clicando no botão cadastrar referente a sub-regras.

Na página de configuração da sub-regra o usuário irá se deparar com algumas regras a mais do que na anterior, ele terá que informar a placa de rede novamente e o usuário deve manter a mesma atenção que teve na regra anterior, depois informar qua a regra "parente" e será associada esta regra parente, deverá informar qual a velocidade garantida e qual a velocidade máxima para esta regra. A figura 20 ilustra a tela da interface para cadastro das regras subparentes.

Figura 20 - cadastro de Subparentes

Writer escreve = new BufferedWriter(new FileWriter("//home//leo//Projeto//tchild"));

A função *BufferedWriter* irá escrever no arquivo "tchild" todas as informações necessárias para a criação de um arquivo com o script o arquivo temporário ficara da seguinte forma:

```
eth0
1
10
30
100
Process r =  Runtime.getRuntime().exec("//home//leo//Projeto//criaregras2");
r.waitFor();
```

Logo após a criação do arquivo temporário o JSP irá chamar o programa *criaregras2*, este programa realiza a criação da regra subparente. Ele irá ler o arquivo "tchild" e a partir dos dados encontrados neste arquivo irá criar outro arquivo denominado "*child"*. Este arquivo irá conter todos os scripts referentes a subregras

parente, por exemplo, (tc class add dev eth0 parent 1:1 classid 1:10 htb rate 30kbits ceil 100kbits).

8.2.3. Cadastro de Regras Filtro

O cadastro das regras filtro, que irão definir que tipo de tráfego será filtrado ou não, por exemplo, deseja-se limitar o tráfego de acesso à páginas WEB de todos os usuários então cria-se uma regra para todos os IP's na porta 80 e referenciando uma sub-regra que irá informar qual será a velocidade do tráfego

Para criar uma regra filtro, é necessário acessar o arquivo "filtros.jsp", no servidor, este arquivo pode ser acessado através da página principal clicando no botão cadastrar referente ao cadastro de regras filtro

Na página de configuração de regras filtro o usuário deverá informar qual a placa de rede em que o tráfego de saída esta ocorrendo, informará qual o IP de destino ou de origem, no caso do IP poderá também informar uma rede completa, informará qual a porta que ocorrerá o controle de banda e qual é a sub-regra que este filtro irá se aplicar. A figura 21 descreve como é a tela de cadastro dos filtros.

É bom lembrar que no caso de o tráfego se aplicar a duas regras filtro fica valendo a regra que foi cadastrada primeiramente.

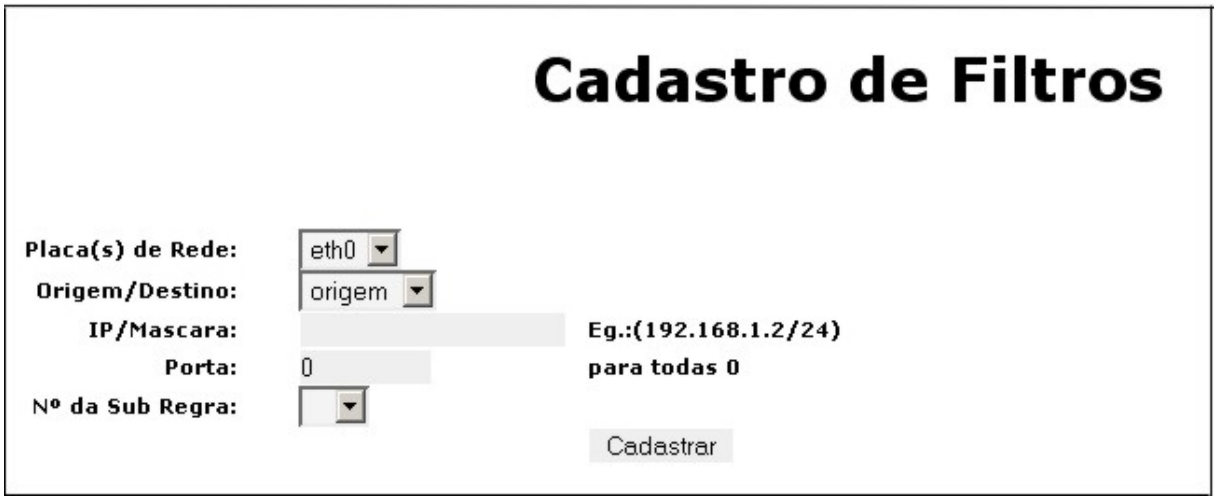


Figura 21 – Cadastro de Filtros

```
Writer escreve = new BufferedWriter(new
FileWriter("//home//leo//Projeto//tfilter"));
```

A função *BufferedWriter* irá escrever no arquivo “tfilter” todas as informações necessárias para a criação de um outro arquivo que conterá o script, o arquivo temporário ficara da seguinte forma:

```
eth0
192.168.200.4
80
10
Process r =  Runtime.getRuntime().exec("//home//leo//Projeto//criaregras2");
r.waitFor();
```

Logo após a criação do arquivo temporário, o JSP irá chamar o programa *criafiltro,* este programa realiza a criação da regra de filtro. Ele irá ler o arquivo “tfilter” e a partir dos dados encontrados neste arquivo irá criar outro arquivo denominado “filter” este arquivo irá conter todos os scripts referentes aos filtros, por exemplo:

```
tc filter add dev eth0 protocol ip parent 1:0 prio 1 u32 \ match ip src
192.168.200.30 match ip dport 80 0xffff flowid 1:10
```

8.3 Ambiente esperado

Para obter os resultados esperados, o software de controle de banda deverá fornecer um meio bastante acessível e simples ao administrador da rede. Nesse contexto, todos os processos referentes ao controle de banda serão realizados através da ferramenta gráfica não ocorrendo configuração via console.

Os administradores terão acesso total ao sistema cabendo a eles restringir o acesso a esta maquina. Os usuários comuns da rede não terão acesso nenhum a este sistema cabendo ao administrador tomar providencias caso alguma regra esteja interferindo no trabalho diário dos funcionários.

9. Roteiro para execução de testes

Os testes de garantia de banda realizados pela ferramenta, foram realizados utilizando três máquinas. O Iperf foi utilizado para gerar tráfego em várias portas simultaneamente através de vários terminais do Linux. A seguir são mostrados os resultados dos testes através de captura das telas no momento em que estes eram realizados.

A figura 21.a ilustra a topologia da rede utilizada nos testes.

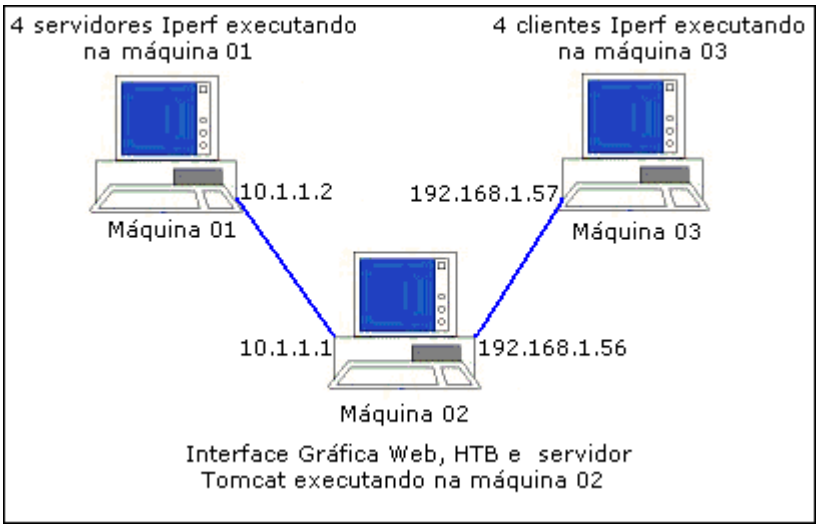


Figura 21.a – Topologia da rede utilizada nos testes

9.1. Primeiro teste

Neste teste não é realizado nenhum controle de banda, apenas os geradores de tráfego (Iperf cliente e servidor) são utilizados para a geração do tráfego. Como pode ser visto na figura 22, não existe nenhum controle (classes) e filtros cadastrados na aplicação gráfica

As figuras 23.a, 23b, 23.c, 23.d, 24.a, 24.b, 24.c e 24.d mostram as telas capturadas no momento dos testes (execução do Iperf) nas máquinas em que os servidores e clientes do Iperf estavam sendo executados. São mostradas oito telas, em

cada uma é mostrado o resultado dos testes do Iperf. É percebido que o tráfego foi praticamente dividido entre os quatro servidores e clientes.

Figura 22 – Tela principal sem classes e filtros cadastrados.

Através do resultado dos testes é percebido uma pequena diferença na largura de banda utilizada entre estes servidores e clientes, esta diferença é inerente ao modo com que o Iperf realiza o cálculo de banda utilizada. Este cálculo é realizado por uma média aritmética dos pacotes que trafegam do cliente para o servidor, logo, quanto maior o tempo dos testes, maior será a precisão dos resultados.

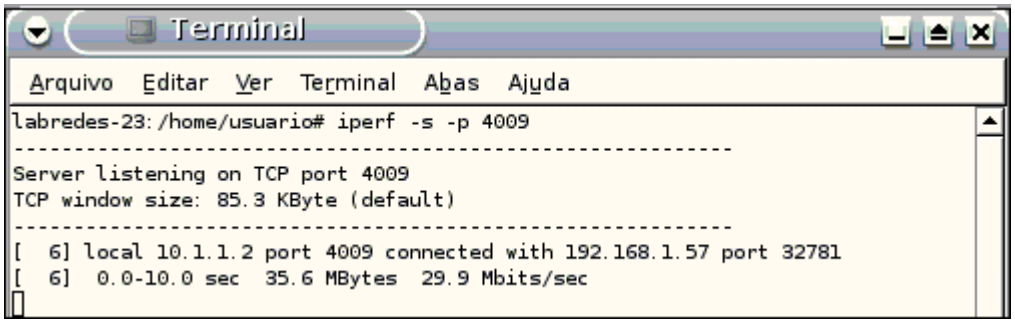


Figura 23.a - resultados dos testes do lado do servidor Iperf 1, sem controle de tráfego.

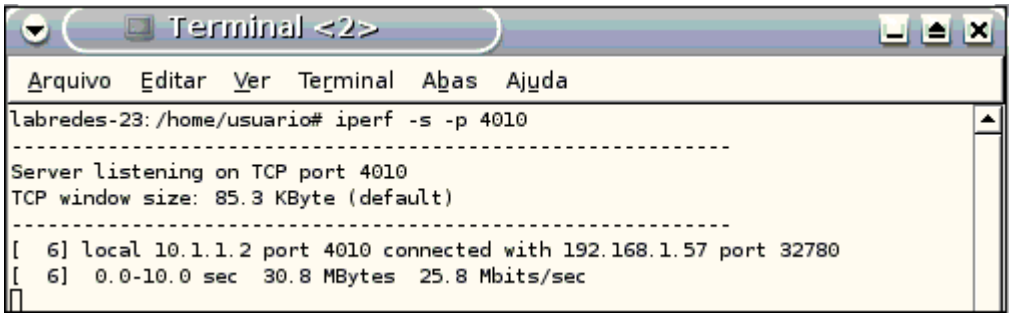


Figura 23.b - resultados dos testes do lado do servidor Iperf 2, sem controle de tráfego.

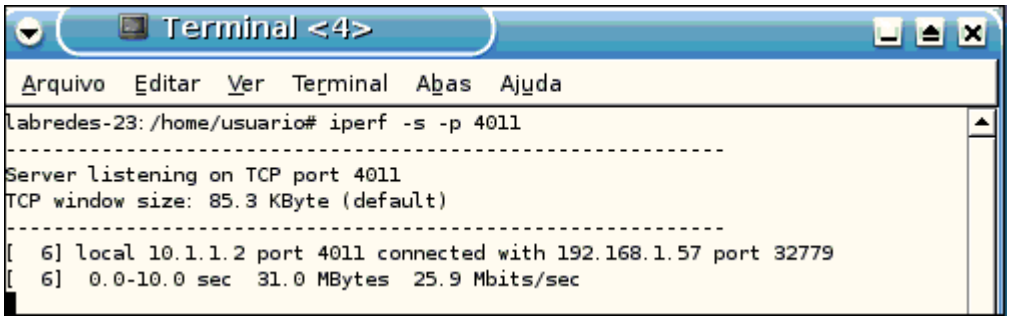


Figura 23.c - resultados dos testes do lado do servidor Iperf 3, sem controle de tráfego.

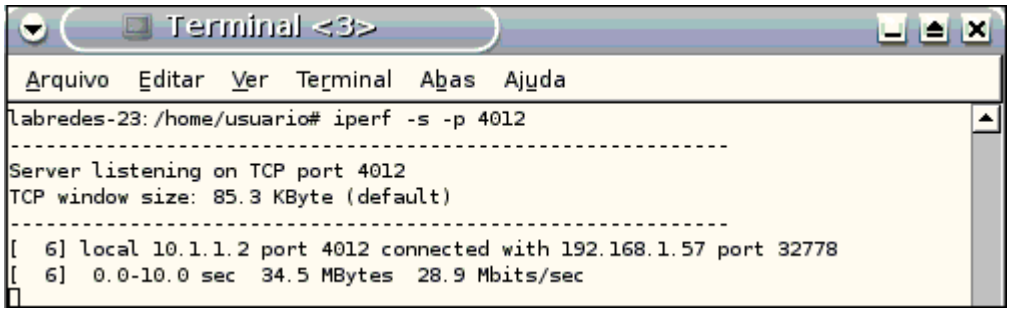


Figura 23.d - resultados dos testes do lado do servidor Iperf 4, sem controle de tráfego.

Foram utilizados quatro servidores cada um aberto em uma porta diferente em uma mesma máquina que tinha o IP 10.1.1.2 , as portas foram 4009, 4010, 4011 , 4012 e para cada porta foi obtido um tráfego médio de 29Mbits/s.

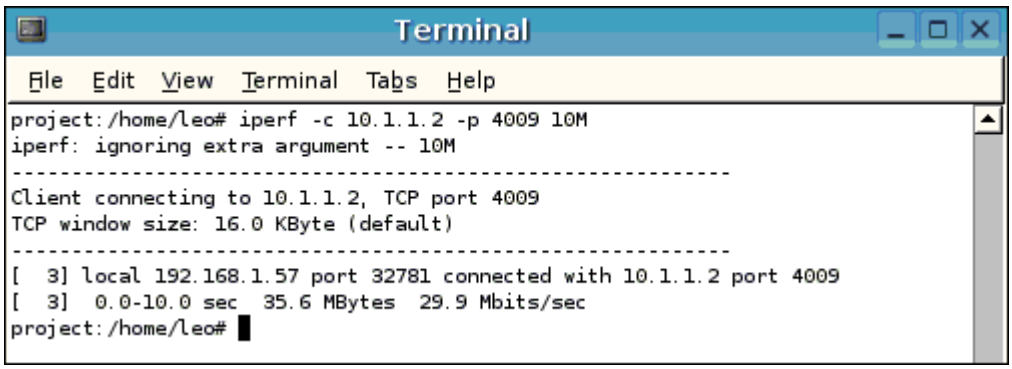


Figura 24.a - resultados dos testes do lado do cliente Iperf 1, sem controle de tráfego.

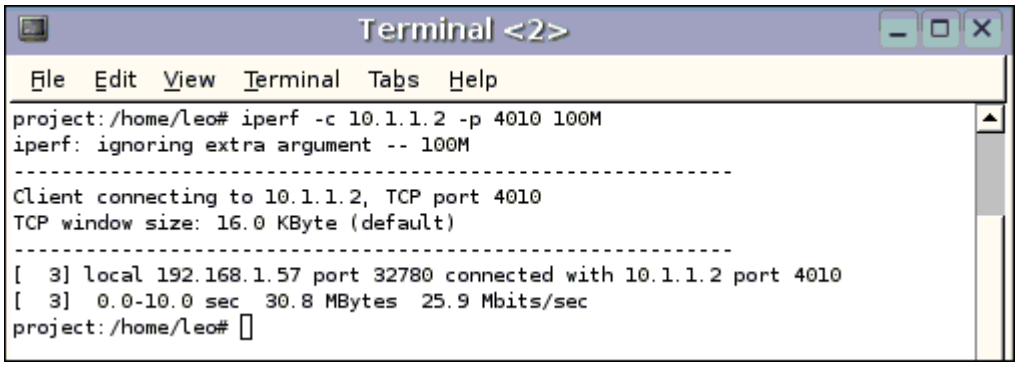


Figura 24.b - resultados dos testes do lado do cliente Iperf 2, sem controle de tráfego.

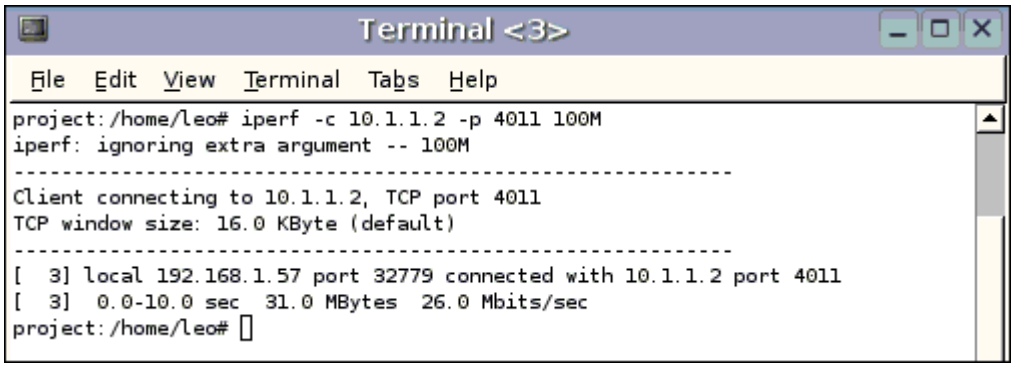


Figura 24.c - resultados dos testes do lado do cliente Iperf 3, sem controle de tráfego.

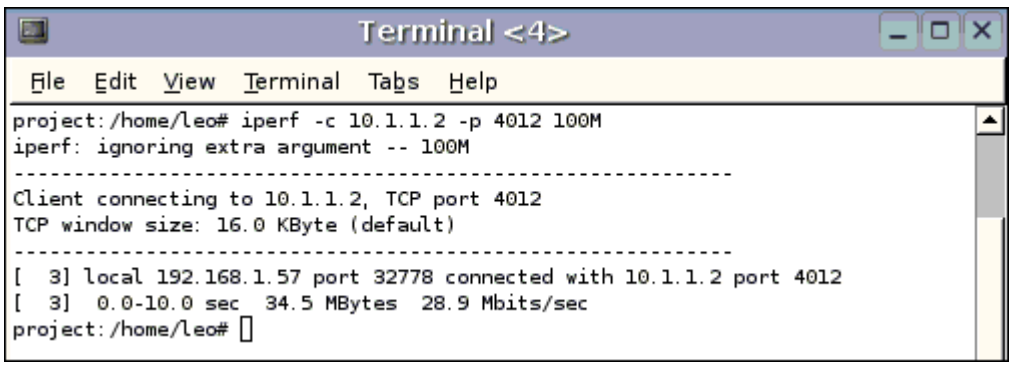


Figura 24.d - resultados dos testes do lado do cliente Iperf 4, sem controle de tráfego.

Foram utilizados quatro clientes cada um enviando pacotes para uma das portas dos servidores abertos na máquina 10.1.1.2. Esta máquina contém o IP 192.168.1.57, as duas máquinas estão em redes diferentes e realizavam a comunicação através de um de uma terceira que está realizando o papel de roteador, controlador de tráfego (HTB) e

interface gráfica Web. Os IP's que esta máquina possui é 192.168.1.56 na eth1 e 10.1.1.6 na eth0.

9.2. Segundo teste

A seguir, serão mostrados os resultados dos testes nos servidores e nos clientes mediante regras de controle aplicadas no servidor HTB através da ferramenta gráfica Web. A figura 25 ilustra a tela após cadastro dos filtros e das classes.

As figuras 26.a, 26.b, 26.d, 27.a, 27.b, 27.c e 27.d ilustram os resultados dos testes após a geração de tráfego pelo Iperf. Neste caso a classe principal reserva 600kits/s da banda total da porta eth0. Notem que nos resultados dos testes, a quantidade de tráfego de um dos servidores foi limitado/reservado pela classe parente 2. Foi conseguida esta taxa próxima de 400kbits/s e não os 100kbits/s garantidos pois não existe nenhuma classe concorrente com a parente 2.

Figura 25 – Tela principal com o filtro e classes cadastradas.

A partir das figuras 26.a, 26.b, 26.d, 27.a, 27.b, 27.c e 27.d é verificado que o tráfego no primeiro servidor, por se enquadrar no filtro configurado, foi limitado a quantidade de banda configurada no HTB pela interface gráfica e o restante da banda da porta eth0 foi distribuída entre os outros três servidores que fogem dos filtros do controlador.

Nesta tela pode-se verificar que foi cadastrada uma classe parente raiz (pai) com tamanho máximo de 600Kbits/s e também foi cadastrada uma subclasse (filha) com velocidade garantida de 100Kbits/s e uma máxima de 400Kbits/s e logo após foi cadastrado uma regra filtro para todo tráfego que tiver destino na porta 4009 e como origem o IP 192.168.1.57.

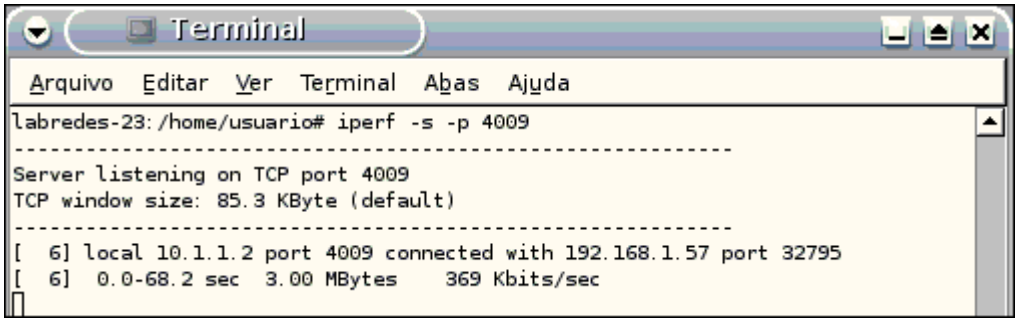


Figura 26.a – resultados dos testes do lado do servidor Iperf 1, com controle de tráfego.

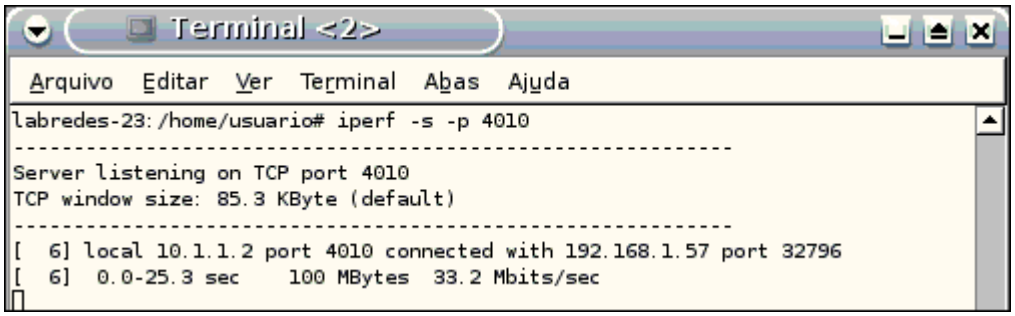


Figura 26.b – resultados dos testes do lado do servidor Iperf 2, com controle de tráfego.

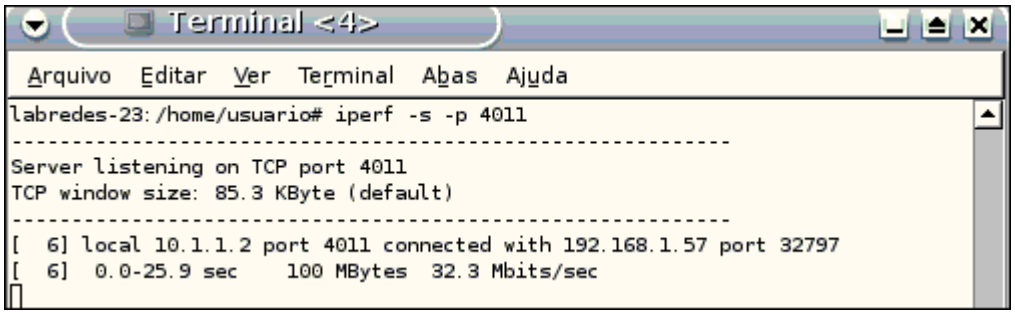


Figura 26.c – resultados dos testes do lado do servidor Iperf 3, com controle de tráfego.

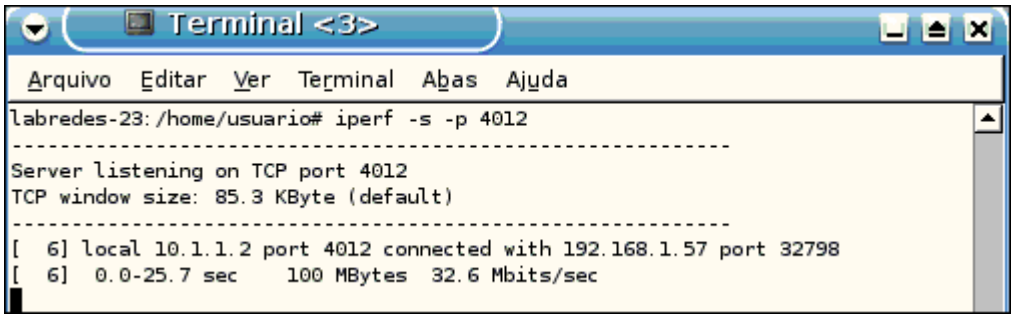


Figura 26.d – resultados dos testes do lado do servidor Iperf 4, com controle de tráfego.

O controle de tráfego foi realizado na porta 4009 pois se enquadrou nas regras do filtro. Verifique que esta porta teve uma taxa de transmissão média de 400Kbits/s, a taxa máxima que foi especificada na subclasse. As outras portas mantiveram seu tráfego com cera de 33Mbits/s na transmissão.

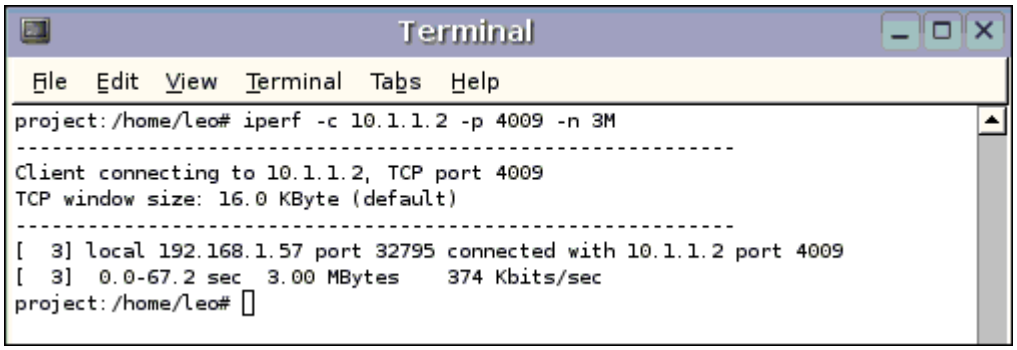


Figura 27.a – resultados dos testes do lado do cliente Iperf 1, com controle de tráfego.

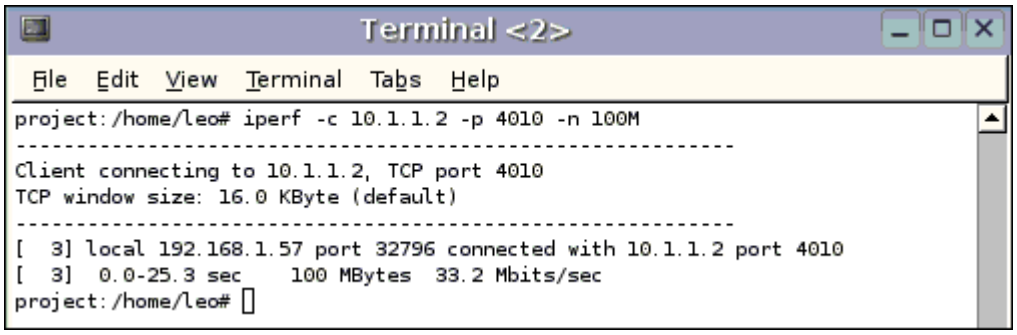


Figura 27.b – resultados dos testes do lado do cliente Iperf 2, com controle de tráfego.

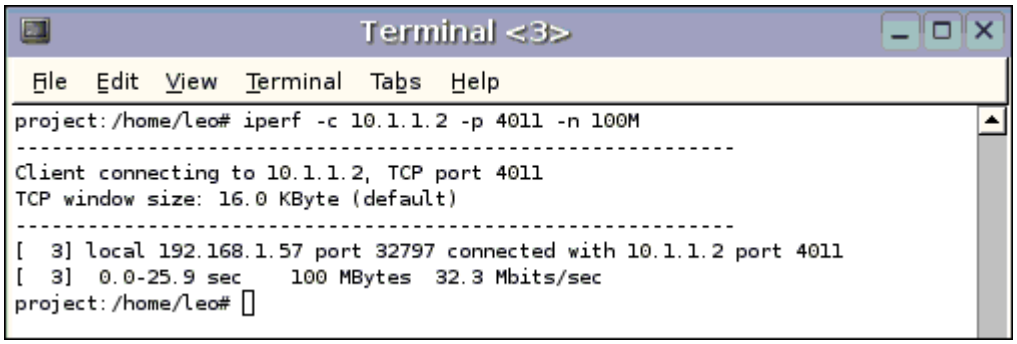


Figura 27.c – resultados dos testes do lado do cliente Iperf 3, com controle de tráfego.

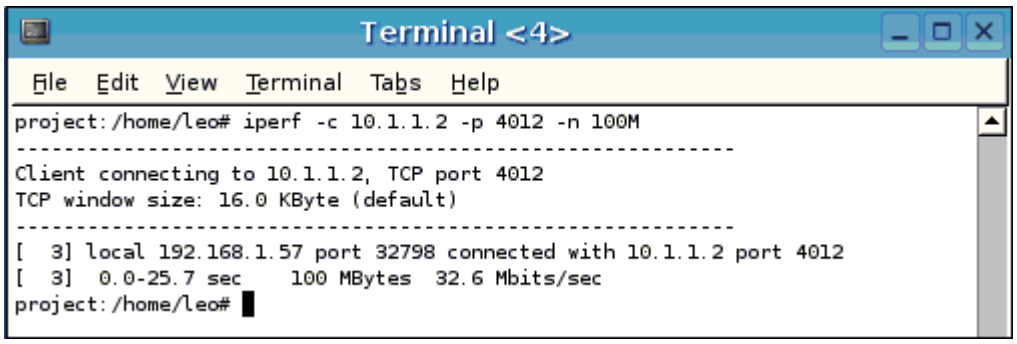


Figura 27.d – resultados dos testes do lado do cliente Iperf 4, com controle de tráfego.

9.3. Terceiro teste

Neste teste além das regras já previamente cadastradas no teste anterior (segundo teste) foi cadastrado mais uma subclasse com taxa garantida de 400Kbits/s e com taxa máxima de 600Kbits/s e também um filtro a mais para a porta 4011. A figura 29.a, 29.b, 29.c, 29.d, 30.a, 30.b, 30.c e 30.d ilustra o resultado dos testes após aplicação destas regras.

Foi observado, nestes testes, que houve a garantia de banda para cada classe cadastrada e o restante da banda foi distribuída para todas as portas. A forma de distribuição é realizada por um algoritmo de distribuição interno que decide a quantidade que será doada a cada regra através de suas prioridades. A figura 28 ilustra a tela da interface após cadastro dos filtros classes.

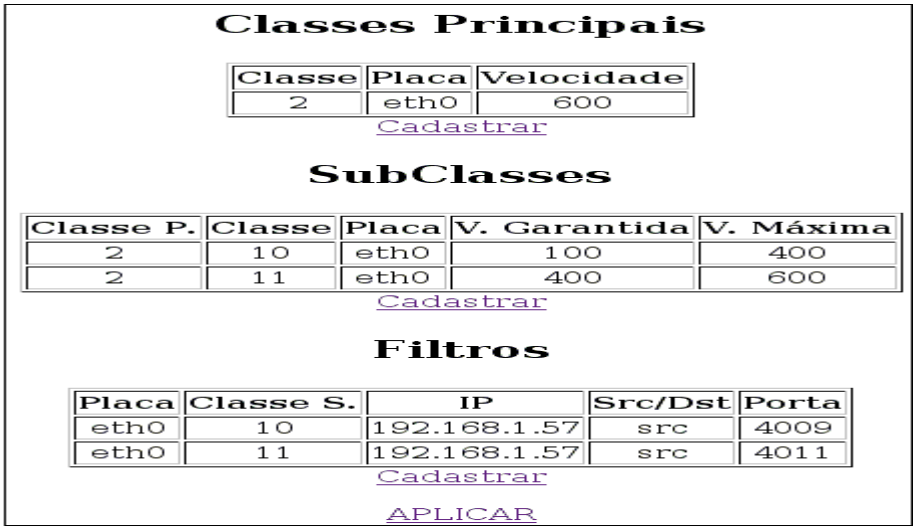


Figura 28– Tela principal com o filtro e classes cadastradas.

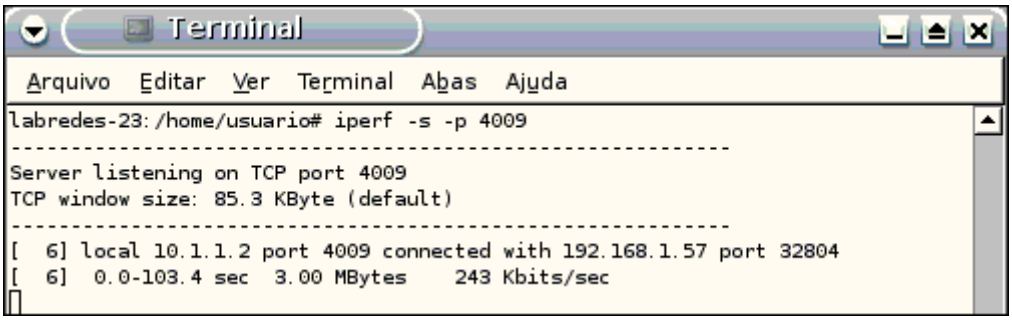


Figura 29.a– resultados dos testes do lado do servidor Iperf 1, com controle de tráfego, com duas classes parentes configuradas.

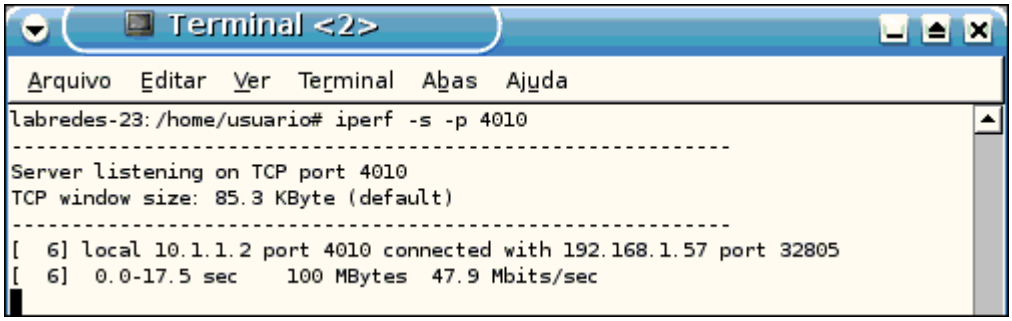


Figura 29.b– resultados dos testes do lado do servidor Iperf 2, com controle de tráfego, com duas classes parentes configuradas.

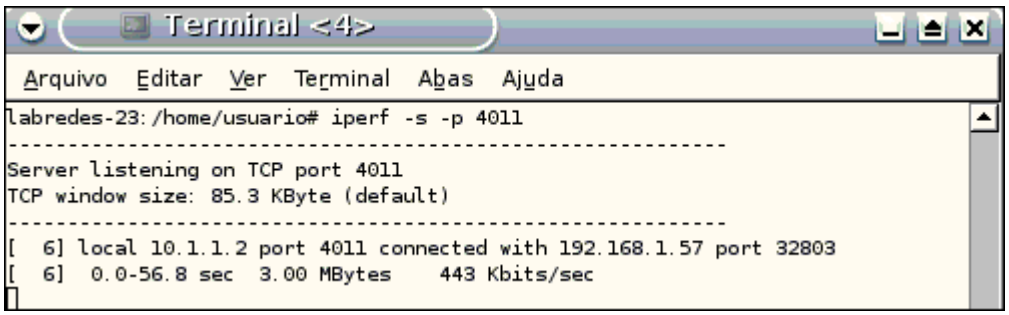


Figura 29.c– resultados dos testes do lado do servidor Iperf 3, com controle de tráfego, com duas classes parentes configuradas.

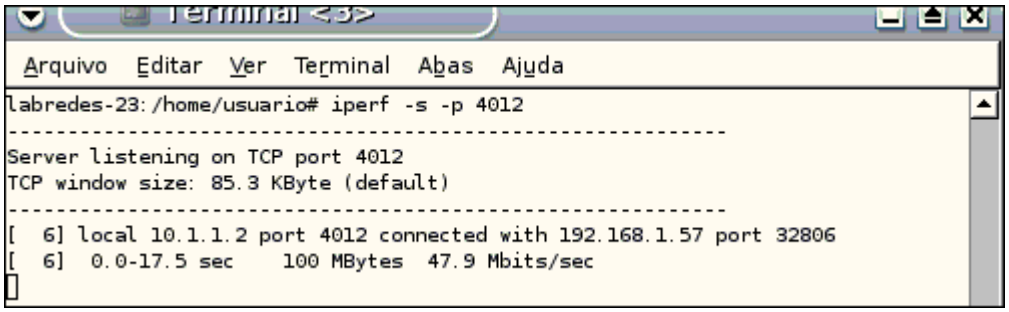


Figura 29.d– resultados dos testes do lado do servidor Iperf 4, com controle de tráfego, com duas classes parentes configuradas.

A partir das figuras 29.a, 29.b, 29.c, 29.d, 30.a, 30.b, 30.c e 30.d, verifica-se que nos dois servidores em que as regras foram aplicadas, o controle de banda ocorreu conforme o configurado no HTB através da interface gráfica. No servidor de porta 4009, a banda se manteve a 243Kbits/s estando esta taxe entre sua banda garantida e máxima. Na porta 4011 o tráfego ficou com uma média de 443Kbits/s também se mantendo dentro da média esperada as outras duas portas dividiram o tráfego que sobrou, ficando com uma média de 47.9Mbits/s.

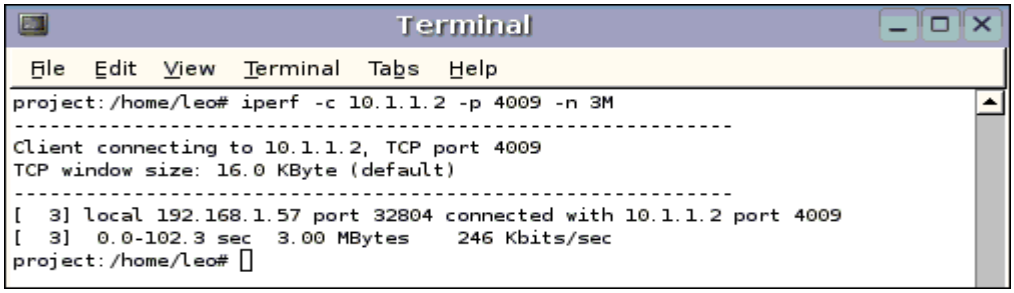


Figura 30.a– resultados dos testes do lado do cliente Iperf 1, com controle de tráfego, com duas classes parentes configuradas.

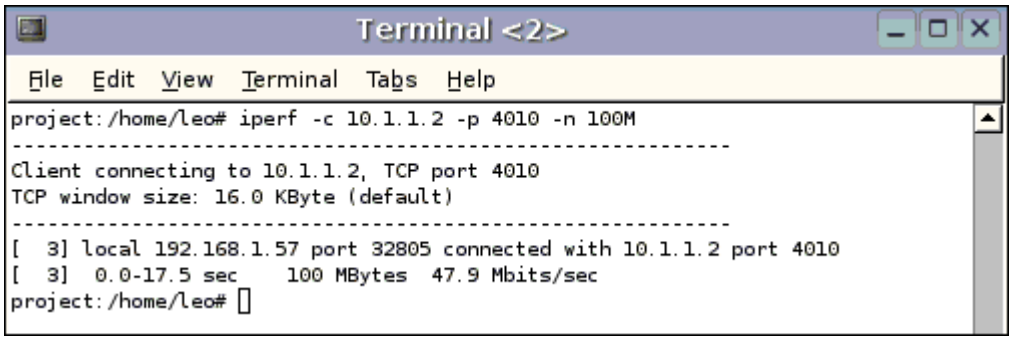


Figura 30.b– resultados dos testes do lado do cliente Iperf 2, com controle de tráfego, com duas classes parentes configuradas.

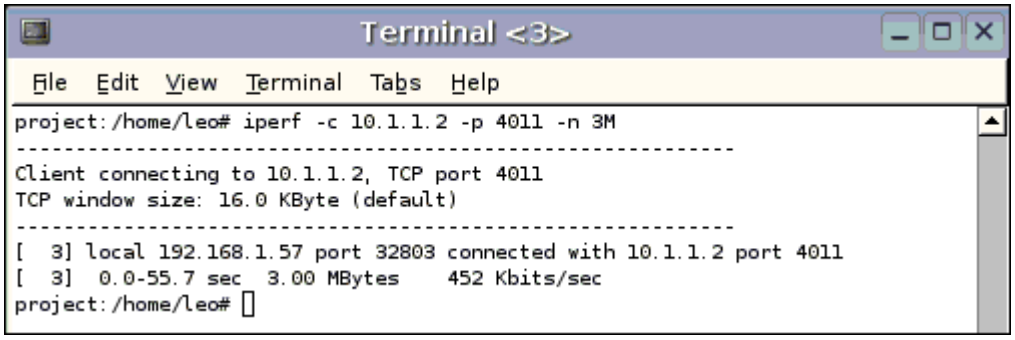


Figura 30.c– resultados dos testes do lado do cliente Iperf 3, com controle de tráfego, com duas classes parentes configuradas.

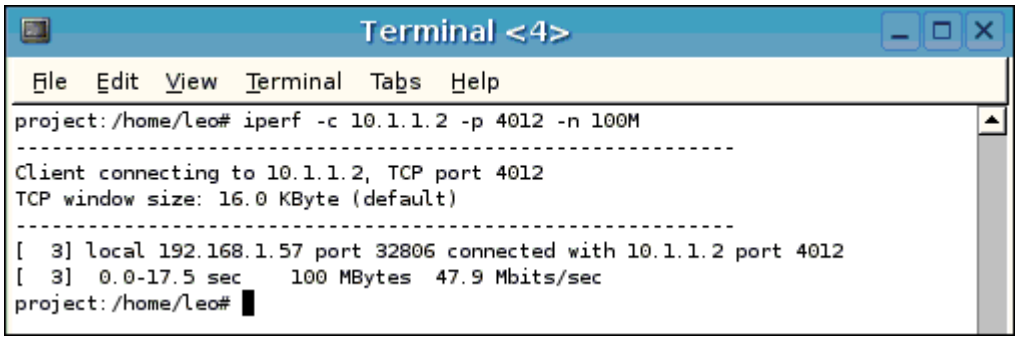


Figura 30.d– resultados dos testes do lado do cliente Iperf 4, com controle de tráfego, com duas classes parentes configuradas.

10. Conclusão

Ao longo deste projeto foi possível perceber que, através de ferramentas gráficas, é possível minimizar as dificuldades de configuração de diversos programas acessados via linhas de comandos.

Foi percebido que os resultados obtidos com a medida de tráfego do Iperf ficam bem próximos do configurados no HTB. Existem diferenças inerentes aos *overheads* dos pacotes, além das medidas realizadas pelo Iperf serem por meios estatísticos através de médias. Pelo fato destas medidas serem realizadas por médias, nos testes realizados, foi percebido que quanto maior a amostra de pacotes, maior a precisão no valor de banda medido pela ferramenta. Portanto é interessante colocar analisadores de banda de alto desempenho, que gerem gráficos em tempo real, para acompanhar o controle de banda.

Será esperado que, com a ferramenta desenvolvida, estudantes e conhecedores da área de redes tenham este produto como um objeto de trabalho e fonte de estudo. Existem outros problemas relacionados ao controle de banda, pode ser considerado o tráfego de informações *peer-to-peer* em que, em sua maioria, não existem portas padrões para seu funcionamento. Neste caso é necessário um estudo das características dos pacotes que fazem parte da comunicação destas aplicações, ou seja, determinar um padrão de assinatura que estes *softwares* utilizam .

É intenção deste, que seja continuado o desenvolvimento da aplicação, tornando-a mais eficaz, e de fácil utilização.

11. Proposta de trabalho futuro

Outras funcionalidades como *plotagem* de gráfico em tempo real realizado pela própria aplicação é interessante, pois irá permitir um retorno dos resultados obtidos após a aplicação das regras pelo usuário.

Realizar inserção de scripts para controlar automaticamente banda referente aos programas *peer-to-peer*.

Como trabalhos futuros é sugerido a realização de testes com o DummyNet e o HTB para ver como os dois se portam e em que casos deve-se utilaza-los.

Fica, também, proposto trabalhos futuros a fim de incorporar esta ferramenta ao Nagios.

12. Bibliografia

[1] **INTSEV** .20 Ago.2005. Disponível em:
<http://pt.wikipedia.org/wiki/IntServ> Acesso em:20 Ago.2005

[2] **WHAT IS A TOKEN BUCKET?** . 22 Ago.2005. Disponível em:
<http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart4/qcpolts.htm#xtocid241931> Acesso em: 22 Ago.2005

[3] **NEWSGENERATION.** 23 Ago.2005. Disponível em:
<http://www.rnp.br/newsgen/0009/qos_voip2.html> Acesso em: 23 Ago.2005

[4] **FREEBSB BRASIL.** 23 Ago.2005. Disponível em:
<http://www.freebsdbrasil.com.br/guia-ipfw.php?cap=7> Acesso em: 23 Ago.2005

[5] **MULTI ROUTER TRAFFIC GRAPHER.** 27 Ago. 2005. Disponível em:
<http://pt.wikipedia.org/wiki/MRTG> Acesso em: 27 Ago.2005

[6] **NETWORKWORLD.** 05 Set.2005. Disponível em:
<http://www.networkworld.com/details/656.html?def> Acesso em: 05 Set.2005

[7] **LINUX MAN PAGE.** 07 Set.2005. Disponível em:
<http://www.die.net/doc/linux/man/man8/tc.8.html> Acesso em: 07 Set.2005

[8] **GNU GENERAL PUBLIC LICENSE.** 15 Set. 2005. Disponível em:
<http://www.gnu.org/copyleft/gpl.html> Acesso em: 15 Set.2005

[9] **UNIVERSITY OF ILLINOIS.** 20 Set.2005. Disponível em:
<http://www.uillinois.edu/> Acesso em: 20 Set.2005

[10] **MÁRCIO D'ÁVILA WEB SITE.** 22 Set.2005. Disponível em: <http://www.mhavila.com.br> Acesso em: 22 Set.2005

[11] **JAVASERVER PAGES OVERVIEW.** 27 Set.2005. Disponível em: <http://java.sun.com/products/jsp/overview.html> Acesso em: 27 Set.2005

[12] **NETBEANS.ORG**. 30 Set.2005. Disponível em: <http://www.netbeans.org/products/ide/> Acesso em: 30 Set.2005

[13] **ABOUT COMPUTING & TECHNOLOGY.** 03 Out.2005. Disponível em: <http://linux.about.com/cs/linux101/g/gcc.htm> Acesso em: 03 Out.2005

[14] **O PROTOCOLO RSVP E O DESEMPENHO DE APLICAÇÕES** . 05 Out.2005. Disponível em: <www.rnp.br/newsgen/0005/rsvp.html> Acesso em: 05 Out.2005

[15] R. Braden, D. Clark, S. Shenker, Integrated Services in the Internet Architecture: an Overview, RFC 1633, julho 1994.

[16]L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, ``RSVP: A New Resource ReSerVation Protocol", IEEE Network, pp. 8-17, setembro 1993.

[17] B. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, Resource ReSerVation Protocol - Version 1 Functional Specification, setembro 1997, http://www.isi.edu/div7/rsvp/rsvp.html .

[18] J. Wroclawski, The use of RSVP with IETF Integrated Services, Internet-Draft, outubro 1996.

[19] D. Clark, ``The Design Philosophy of the DARPA Internet Protocols", Proceedings of ACM SIGCOMM' 88, agosto 1988.

[20] S.Floyd, V. Jacobson, ``Link-sharing and Resource Management Models for Packet Network", IEEE/ACM Transactions on Networking, vol. 3 No. 4, agosto 1995.

[21] K. Cho, ``A Framework for Alternate Queueing: Towards Traffic Management by PC-UNIX Based Routers'', Proceedings of USENIX 1998, Annual Conference, New Orleans LA, June 1998.

[22] K. Cho, ALTQ - TIPS.txt, draft contido no pacote ALTQv.1.1.3.

[23] H. Zhang, ``Services Disciplines for Guaranteed Performance Service in Packet-Switching Networks'', Proceedings of the IEEE, 83(10), outubro 1995.

[24] S. Berson e S. Vincent, ``Aggregation of Internet Integrated Services State'', Procceedings of JWQoS, Maio 1998.

[25] Ana L.P.Schmidt. ``Análise de Qualidade de Serviço para Aplicações de Tempo Real Utilizando Reserva de Recursos na Internet'', Tese de Mestrado, UFRJ, Setembro 1999.

[26] **CLUBES DAS REDES**. 01 Nov.2005.Disponível em:
< http://www.clubedasredes.eti.br> Acesso em: 01 Nov.2005

[27] **REDES DE COMPUTADORES I.** 05 Nov.2005. Disponível em:
<http://myspace.eng.br/info/net/net1.asp> Acesso em 05 Nov.2005

[28] **SERVIDOR MESONPI.** 10 Nov.2005. Disponível em:
<http://mesonpi.cat.cbpf.br/naj/tcpipf.pdf> Acesso em: 05 Nov.2005

[29] **NETWORKWORLD.** 17 Nov.2005. Disponível em:<http://www.networkworld.com/details/656.html?def> Acesso em:17 Nov.2005

[30] **LINUX NEWS**. 17 Nov.2005. Disponível em:
<http://www.linuxnews.com.br/artigosredes/rede23.php> Acesso em: 17 Nov.2005

[31] **HTB HOME.** 07 Jul.2005. Disponível em:
<http://luxik.cdi.cz/~devik/qos/htb/> Acesso em: 07 Jul.2005

[32] **COSMO.** 28 Out.2005. Disponível em:
<http://kplus.cosmo.com.br/materia.asp?co=11&rv=Vivencia> Acesso em:28 Out.2005

[33] **PRO.** 28 Out.2005. Disponível em:
<http://www.projetoderedes.com.br/artigos/artigo_modelo_osi.php > Acesso em:28 Out.2005

[34] **RNP.** 28 Out.2005. Disponível em:
< http://www.rnp.br/noticias/2003/not-031017b-coord.html > Acesso em:28 Out.2005

[35] **BERKELEY LAB.** 28 Out.2005. Disponível em:
< http://www.lbl.gov/> Acesso em:28 Out.2005

[36] **BERKELEY UNIVERSITY OF CALIFORNIA.** 28 Out.2005. Disponível em:
< http://www.berkeley.edu/> Acesso em:28 Out.2005

[37] **IPFILTER.** 28 Out.2005. Disponível em:
< http://coombs.anu.edu.au/~avalon/> Acesso em:28 Out.2005

[38] **BRCONNECTION.** 26 Nov.2005. Disponível em:
< http://www.brc.com.br> Acesso em:26 Nov.2005

[39] **NLANR.** 26 Jul.2005. Disponível em:
<http://dast.nlanr.net/Projects/Iperf/ > Acesso em:26 Jul.2005

[40] **TOMCAT.** 23 Jul.2005. Disponível em:
< http://tomcat.apache.org/ > Acesso em:23 Jul.2005

[41] **QOS QUALITY OF SERVICE** 23 Jul.2005. Disponível em:
< http://www.inf.lasalle.tche.br/~tasso/redes2/QoS%20-%20Artigo.doc > Acesso em:23 Jul.2005

13. Glossário

ATM - é uma tecnologia de comunicação de dados de alta velocidade usada para interligar redes locais, metropolitanas e de longa distância para aplicações de dados, voz, áudio, e vídeo.

Bandwidth – largura de banda.

Bit - é a menor unidade de informação eletrônica.

Byte – é um conjunto de oito bits.

BSD – é uma licensa de software. A licença BSD cobre as distribuições de software da *Berkeley* Software *Distribution,* além de outros programas. Esta é uma licença considerada 'permissiva' porque impõe poucas restrições sobre a forma de uso, alterações e redistribuição do software licenciado. O software pode ser vendido e não há obrigações quanto a inclusão do código fonte, podendo o mesmo ser incluído em software proprietário. Esta licença garante o crédito aos autores do software mas não tenta garantir que trabalhos derivados permanecem como software livre.

CBQ-Class Based Queuing – controle de banda baseado em classes.

Cluster - Um cluster é um tipo especial de sistema distribuído construído a partir de computadores convencionais (desktops). Vários computadores são ligados em rede e comunicam através de núcleos do sistema operativo alterados para o efeito, utilizando protocolos de redes específicos.

Delay – atraso no envio de informações.

DHCP - O DHCP, *Dynamic Host Configuration Protocol*, é um protocolo de serviço TCP/IP segura, que oferece configuração dinâmica com concessão de endereços IP de *host* e distribui outros parâmetros de configuração para clientes de rede.

DMZ – Zona desmilitarizada. *Demilitarized Zone* ou Zona desmilitarizada, termo de guerra correspondente. É uma area da rede onde se colocam os servidoes que tem que ficar expostos a Internet com servidores de Web, E-Mail, Terminal e outros. No *WandAccess* esta área é configurada em uma terceira interface de rede.

Download - mesmo que baixar ou transferir dados de uma página para seu computador.

Dummynet - *Dummynet* é uma ferramenta para gerenciamento de banda.

Chipset - De uma forma geral, um *chipset* (anglicismo que significa grupo de chips) é o cérebro de uma placa de circuitos.

Debian – nome dado a uma distribuição do linux.

Ethernet – padrão de interface de comunicação de dados

First-in-first-out – termo utilizado em filas, primeiro que entra é o primeiro que sai.

Frame Relay - tecnologia de comutação de pacotes que assegura com eficiência a entrega de pacotes ou tramas através de circuitos virtuais.

Gcc - C, C++, Objective-C, Fortran, Java, and Ada,

GPL - A Licença Pública Geral GNU (GN*U General Public License GPL*) é a licença que acompanha os pacotes distribuídos pelo Projeto GNU, e mais uma grande variedade de software, incluindo o núcleo do sistema operacional Linux. A formulação da GPL é tal que ao invés de limitar a distribuição do software por ela protegido, ela de fato impede que este software seja integrado em software proprietário. A GPL é baseada na legislação internacional de *copyright*, o que deve garantir cobertura legal para o software licenciado com a GPL. (veja também a recém publicada licença CC-GNU GPL [Brasil]).

Htb - *Hierachical token bucket* (balde de tokens hierárquico).

IP - IP pode ser um acrónimo para *Internet Protocol*, o protocolo sob o qual assenta a infra-estrutura da Internet; Endereço IP, um modelo de endereçamento utilizado pelo protocolo IP.

IGMP - Internet Group Management Protocol.

Iperf – é uma ferramenta para simulação de tráfego na rede.

Ipfirewall – é o firewall do sistema operacional FreeBSD.

IOS – *Internetwork Operating System* – Proprietário da Cisco.

Open source - O termo Software Livre se refere aos softwares que são fornecidos aos seus usuários com a liberdade de executar, estudar, modificar e repassar (com ou sem alterações) sem que, para isso, os usuários tenham que pedir permissão ao autor do programa.

Java – linguagem de programação de alta portabilidade.

Jitter - é um desvio ou deslocamento de algum aspecto dos pulsos de um sinal digital. O desvio pode ocorrer em termos de amplitude, do tempo da fase ou extensão do pulso do sinal.

JSP - é a abreviação de Java Server Pages, que em português seria algo como Páginas de Servidor Java. É então, uma tecnologia orientada a criar páginas web com programação em Java.

Kernel - pode ser entendido com uma série de arquivos escritos em linguagem C e em linguagem *Assembly* que constituem o núcleo do sistema operacional;

Kilobyte - Equivalente a 1.024 bytes e sua sigla é kb.

Linguagem assembler - O *assembly* é uma notação legível por humanos para o código de máquina que uma arquitectura de computador específica usa. A linguagem-máquina, que é um mero padrão de bits, é tornada legível pela substituição dos valores em bruto por símbolos denominados mnemónicas.

Netbeans – ferramenta case para desenvolvimento java.

Overhead – é o cabecalho de uma determinado protocolo

On-line - significa estar conectado na Internet.

Policing – processo de controle de entrada de informações em um dispositivo de rede.

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

QoS - qualidade de serviço.

Queueing Discipline (qdisc) -são regras de enfileiramento.

Free - livre, liberar; software cujo código fonte está disponível publicamente e que pode ser modificado e distribuído por qualquer pessoa.

RSVP- *reservation protocol.*

Scripts – seqüências de comandos utilizados geralmente para configuração.

Softwares - é uma seqüência de instruções a serem seguidas e/ou executadas, na manipulação, redirecionamento ou modificação de um dado/informação ou acontecimento.

Switchs – são equipamentos de chaveamento que criam uma espécie de canal de comunicação exclusiva entre a origem e o destino.

Traffic Control – controlador de tráfego do Linux.

Traffic Shapping - é o condicionamento do débito de redes, com a finalidade de priorizar o tráfego e gerir a largura de banda disponível.

Three-way handshake – é o mecanismo utilizado pelos dispositivos de rede TCP/IP para estabelecimento de conexão.

Token - Uma mensagem específica ou padrão de bit que significa permissão para transmissão.

TomCat -Tomcat é um container de Servlets que é usada na implementação das tecnologias *Java Servlets e Java Server Pages* (JSP).

Upload – processo de envio de informações.

VPN - Uma Rede Privada Virtual (*Virtual Private Network* - VPN) é uma rede de comunicações privada normalmente utilizada por uma empresa ou um conjunto de empresas e/ou instituições, construída em cima de uma rede de comunicações pública (como por exemplo, a Internet). O tráfego de dados é levado pela rede pública utilizando protocolos padrão, não necessariamente seguros.

X.25 – protocolo de comunicação de dados.

ANEXO A Especificação detalhada de *hardware*

Controlador de tráfego – Dell

Componente	Especificação
Processador	Tecnologia: Intel Pentium IV Quantidade: 1 processador Clock: 1.8 GHz Cache L2: 256kB FSB (Front Side Bus): 400 MHZ
Memória	Tecnologia: DDR (Double Data Rate) SDRAM Capacidade: 512kB
Chipset	Via (Gx 412)
BIOS	Chip: Flash ROM de 8Mbit
Slots	Slots PCI: - 1 slots AGP 8x - 2 slots PCI 32 bits/33 MHz
Conexões de Entrada e Saída	- 2 Conectores DB-9 para interface seriais, padrão RS-232 (UART16550A) - 1 Conector Mini-Din para interface de teclado padrão PS/2 - 1 Conector Mini-Din para interface de mouse padrão PS/2 - 6 (4traseiros + 2 frontal) Conectores de 4 pinos para interface USB (Universal Serial Bus) padrão 2.0 - 2 Conectores padrão RJ-45 para interfaces de rede integrada - 1 Conector DB-15 para interface de vídeo integrada.
Controladoras de Discos (Integrada)	IDE Velocidade: 80 MB/s Canais: 2 canais
Controladoras de Rede (Integradas)	LAN 1 Chipset: Intel 82550 Padrão: 10 Base-T/100 base TX Fast Ethernet LAN 2 Chipset: Intel 82554 Padrão: 10 Base-T/100 base TX e base 1000-TX
Controladora de Vídeo (Integrada)	Chipset: ATI® RAGE® XL Memória: 4MB SDRAM Resolução Máxima: 1600 x 1200, 8/16/24 e 32 bit / pixel
Discos	Externo – USB - Seagate Capacidade: 80 GB Rotação: 7.200 rpm padrão
Unidade de Disco Flexível	Padrão: 3 ½” Slim (combo) Capacidade: 1.44MB Características: Compatível com discos Dupla Face / Dupla Densidade
Dispositivos Óticos	CD-ROM Padrão: IDE Slim ½” Tecnologia: UDMA2 Velocidade: 48X máxima Tempo de Acesso: 110ms Buffer:128Kb
Gabinete	Rack - Horizontal Baias de 5 1/4": 1 baia Dimensões: 400mm x 300mm x 500mm (LxPxA) Peso: 4,5 Kg Max
Fonte de Alimentação	Potência de saída: 400Watts Tensões de Entrada: 110/220V
Teclado	Padrão: ABNT2, 107 teclas + Teclas Multimídia Conexão:Mini-DIN
Mouse	Resolução: 400dpi Botões: 3 botões, sendo um com scroll Conexão: Padrão PS/2 Mini-DIN

ANEXO B – Instalação do Debian 3.1 Kernel 2.6.2

Obtenção do *Software*

Para obter o *software*, basta entrar no *site* oficial:

- Endereço : http://www.debian.org/
- Selecionar o atalho: *obtain a copy;*
- Selecionar Download images of Debian CD/DVDs;
- Será dada 3 opções para salvar as imagens:
  Download a *minimal bootable CD image* – consiste em salvar uma versão básica para instalação do sistema operacional;

  * Comprar os CDs/DVDs.

  * Baixar as imagens dos CDs/DVDs utilizando o jigdo – programa para baixar arquivos na Internet;

  * Baixar as imagens dos CDs/DVDs utilizando o BitTorrent – programa para baixar arquivos na Internet;

  * Baixar as imagens dos CDs/DVDs utilizando o HTTP ou FTP.

- Deve ser escolhido uma das quatro formas de obtenção dos arquivos para a instalação.
- Após a obtenção dos CDs/DVDs pode acessar o manual de instalação on line em http://www.debian.org/releases/stable/i386/

ANEXO C – Instalação do NetBeans 4.1 no Debian 3.1

Obtenção do *Software*

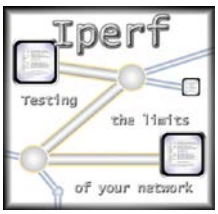
Para obter o *software*, basta entrar no site da NetBeans IDE 4.1 Release download:

- no endereço: http://www.netbeans.info/downloads/download.php
- É necessário aceitar o acordo para obter o acesso ao *software*
- Após aceitar o contrato selecione *Download your file here*.
- Será dada a opção de salvar pelo navegador, salve no diretório desejado

Instalação do *Software*

- Execute o arquivo obtido, netbeans_4.1-linux.bin, após isso o netbeans estará instalado

ANEXO D – Instalação do Iperf no Debian 3.1



Obtenção do *Software*

Para obter o *software*, basta entrar no site da Nlanr:

- Endereço: http://dast.nlanr.net/Projects/Iperf/
- Selecionar o atalho Iperf Version 2.0.2 download source
- Será dada a opção de salvar pelo navegador, salve no diretório desejado
- Descompacte o arquivo
- Compile o código fonte com o gcc: gcc Iperf2.0.2 – o Iperf

Descompactar o Programa

Escolha o diretório em que serão armazenados os arquivos, e descompacte o programa.

Instalação do Iperf

Não é necessária a instalação do programa, basta colocá-lo no Path do Linux.

Pode ser instalado *online*, também, no Debian, através do comando apt-get install Iperf.

ANEXO E – Instalação do HTB no Debian 3.1

Obtenção do *Software*

Para obter o *software*, basta entrar no site da HTB:

- Endereço: http://luxik.cdi.cz/~devik/qos/htb/
- Selecionar o atalho htb3.6-020525.tgz
- Será dada a opção de salvar pelo navegador, salve no diretório desejado

Descompactar o Programa

Escolha o diretório em que serão armazenados os arquivos, e descompacte o programa de acordo com os comandos a seguir:

tar –zxpvf htb3.6-020525.tgz -C /

Instalação do Iperf

Não é necessária a instalação do programa, após a descompactação basta copiar o tc para /sbin.

Pode ser instalado *online*, também, no Debian, através do comando apt-get install route

ANEXO F – Instalação do TomCat no Debian 3.1

Obtenção e instalação do *Software*

O software pode ser instalado através de uma instalação de pacotes *online* conforme indicado a seguir:

‘apt-get install Tomcat4’ para instalar o compilador GCC.

ANEXO G – Instalação do GCC no Debian 3.1

Obtenção e instalação do *Software*

O software pode ser instalado através de uma instalação de pacotes *online* conforme indicado a seguir:

‘apt-get install gcc’ para instalar o compilador GCC.

ANEXO H – Instalação do Java no Debian 3.1

Obtenção e instalação do *Software*

O software pode ser instalado através de uma instalação de pacotes *online* conforme indicado a seguir:

‘apt-get install j2sdk1.4’ para instalar o SDK ou 'apt-get install j2re1.4' para instalar o Runtime Environment.