Universidade Federal do Rio de Janeiro

Núcleo de Computação Eletrônica

Renato Gomes do Nascimento

QUALIDADE DE SERVIÇOS EM REDES TCP/IP E IMPLANTAÇÃO DE VOIP:

Uma Visão Prática.

Rio de Janeiro

2006

Renato Gomes do Nascimento

QUALIDADE DE SERVIÇOS EM REDES TCP/IP E IMPLANTAÇÃO DE VOIP:

Uma Visão Prática.

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Orientador:

Prof. Paulo Henrique de Aguiar Rodrigues, Ph.D., Univ. da Califórnia, EUA

Rio de Janeiro

2006

Renato Gomes do Nascimento

QUALIDADE DE SERVIÇOS EM REDES TCP/IP E IMPLANTAÇÃO DE VOIP:

Uma Visão Prática.

Monografia apresentada para obtenção do título de Especialista em Gerência de Redes de Computadores no Curso de Pós-Graduação Lato Sensu em Gerência de Redes de Computadores e Tecnologia Internet do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro – NCE/UFRJ.

Aprovada em Maio de 2006.

Prof. Paulo Henrique de Aguiar Rodrigues, Ph.D., Univ. da Califórnia, EUA

Dedico este trabalho a todas as pessoas que dele fizeram parte direta e indiretamente.

E aos colegas, companheiros nas dúvidas, ansiedades, incentivos, alegrias e conquistas, o meu fraterno agradecimento.

AGRADECIMENTOS

Ao concluir este trabalho impossível esquecer daqueles que contribuíram para a sua realização. Agradeço...

A Deus, doce presença, sempre a iluminar o meu caminho.

A João e Ivaneide, meus pais, que me apoiaram, incentivaram e compreenderam minhas ausências.

A meus irmãos que contribuíram na minha trajetória.

À Sarah, sobrinha querida, luz da minha caminhada, motivo para seguir adiante.

Expresso, também, a minha mais sincera gratidão ao Professor Leandro Caetano Lustosa pelo esforço e dedicação prestados durante a realização dos experimentos.

E, por último, mas não menos importante ao Professor e Orientador Paulo Henrique de Aguiar Rodrigues, a me interessar pelo tema proposto, pela sua atenção, dedicação, disponibilidade e palavras.

RESUMO

NASCIMENTO, Renato Gomes do. **QUALIDADE DE SERVIÇOS EM REDES TCP/IP E IMPLANTAÇÃO DE VOIP: Uma Visão Prática.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2006.

Este trabalho investiga a aplicação de várias técnicas de QoS (Quality of Service – Qualidade de Serviço) em uma infraestrutura de rede TCP/IP, com a finalidade de garantir que os serviços mais sensíveis e importantes da Instituição possam operar adequadamente. O desenvolvimento de QoS torna-se necessário, pois, atualmente, a maioria das redes, inclusive a Internet, não oferecem garantias adequadas na operação de aplicações de tempo-real ou de missão-crítica. Nosso cenário é um Hospital envolvendo dois prédios diferentes que são interligados por uma rede sem fio.

ABSTRACT

NASCIMENTO, Renato Gomes do. **QUALIDADE DE SERVIÇOS EM REDES TCP/IP E IMPLANTAÇÃO DE VOIP: Uma Visão Prática.** Monografia (Especialização em Gerência de Redes e Tecnologia Internet). Núcleo de Computação Eletrônica, Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2006.

This work investigates the application of several QoS techniques in a TCP/IP network infrastructure in order to guarantee that most sensible and important services can work properly. QoS deployment is necessary, because, nowadays, most networks, including the Internet, do not provide adequate guarantees for execution of mission-critical or real-time applications. Our scenario is a hospital involving different buildings interconnected by a wireless LAN.

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

ACK	Acknowledgment
AF	Assured Forwarding
ATM	Asynchronous Transfer Mode
avg	average
BA	Behavior Aggregate
CBS	Committed Burst Size
CBWFQ	Class-Based Weighted-Fair Queuing
CIR	Committed Information Rate
CONPREV	Coordenação de Prevenção e Vigilância
CPU	Central Processing Unit
CQ	Custom Queuing
cwnd	congestion window
DiffServ	Differentiated Service
DS	Differentiated Service
DSCP	Differentiated Service Code Point
EBS	Excess Burst Size
EF	Expedited Forwarding
FCFS	First Come First Served
FIFO	First In First Out
FQ	Fair Queuing
FTP	File Transfer Protocol
HCI	Hospital do Câncer I
HT	Hyper-Threading
IEEE	Institute of Electrical and Electronics Engineers
INCA	Instituto Nacional de Câncer
IP	Internet Protocol
ITU	International Telecommunication Union
max_{th}	maximum threshold
MF	Multi-Field
min_{th}	minimum threshold
MOBVEM	Modified OpenH323 Based Voice Evaluation Module
MOS	Mean Opinion Score
NCE	Núcleo de Computação Eletrônica
OSI	Open Systems Interconnection
OWD	One Way Delay
PBS	Peak Burst Size
PDU	Protocol Data Unit

PHB	Per-Hop Behavior
PIR	Peak Information Rate
PQ	Priority Queuing
QoS	Quality of Service
recwnd	receive window
RED	Random Early Detection
RTT	Round Trip Time
SLA	Service Level Agreement
SN	Sequence Number
SNMP	Simple Network Management Protocol
SRTCM	Single Rate Three Color Marker
ssthresh	slow start threshold
TCA	Traffic Conditioning Agreement
TCP	Transmission Control Protocol
TRTCM	Two Rate Three Color Marker
UFRJ	Universidade Federal do Rio de Janeiro
VoIP	Voice over IP
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Detection

SUMÁRIO

1 **INTRODUÇÃO**

1.1 O PROBLEMA

O rápido desenvolvimento da tecnologia possibilitou a construção de computadores com maior capacidade de processamento, com custos mais acessíveis contribuindo com o crescimento de sua utilização nos diversos segmentos da sociedade.

A necessidade de conectar esses computadores em redes e o surgimento da rede mundial chamada Internet fez com que o modelo TCP/IP (*Transmission Control Protocol / Internet Protocol*) se tornasse um padrão aceito em todo mundo pela facilidade de sua implementação e a sua interoperabilidade em diferentes tipos de tecnologias.

Vegesna (2001) afirma que a maior rede IP (*Internet Protocol*), é com certeza a Internet. A Internet cresceu ao longo de poucos anos, assim como seu uso e o número de aplicações disponíveis. Como a Internet e as *Intranets* corporativas continuam a crescer, aplicações diferentes das tradicionais, tais como VoIP (*Voice over* IP - voz sobre IP) e vídeo-conferência, são vislumbradas. Mais e mais usuários e aplicações estão aparecendo na Internet a cada dia, e esta precisa da funcionalidade de suportar tanto as existentes como as emergentes aplicações e serviços. Hoje, entretanto, a Internet oferece apenas o serviço de melhor-esforço (*best-effort service*), que é caracterizado pela ausência de garantias na entrega dos pacotes de dados ao receptor.

Segundo Chiozzotto e Silva (1999) a Internet foi concebida para ser, entre outras coisas, uma rede tolerante a falhas, e a filosofia de operação que foi adotada no protocolo IP para que essa exigência fosse atendida fez com que o serviço oferecido para as aplicações seja do tipo não determinístico no que se refere ao comportamento do tráfego. E em redes com essa característica, é praticamente impossível pensar em executar aplicações que necessitem de um comportamento determinístico do tráfego, isto é, que possam ter garantia para si uma

determinada qualidade de serviço a ser respeitada pela rede. Todas as aplicações envolvendo áudio e/ou vídeo em tempo real se encaixam nessa categoria.

Desta forma, surge a Qualidade de Serviços (QoS – *Quality of Service*) em Redes TCP/IP como um mecanismo capaz de garantir os requisitos mínimos necessários para operação dessas novas aplicações, bem como a possibilidade de diferenciar os diversos tipos de tráfegos existentes em uma rede TCP/IP.

1.2 MOTIVAÇÕES

Observou-se no Instituto Nacional de Câncer (INCA), a necessidade de implantação de algumas aplicações multimídia, tais como vídeo-conferência e VoIP, que necessitam de Qualidade de Serviço; e que a instituição não possui implementado tipo algum de mecanismo de QoS para priorizar os fluxos de dados dessas novas e de outras aplicações.

1.3 OBJETIVOS

Este trabalho visa mostrar como o uso de QoS em redes TCP/IP é importante para que os serviços de tempo-real tenham boa operação, apresentar algumas técnicas de QoS para facilitar o tráfego de aplicações com necessidades especiais e responder a seguinte questão: Como aplicar Qualidade de Serviços para priorizar o tráfego de missão-crítica na rede de uma instituição?

Responder a esta questão e implementar Qualidade de Serviço de forma adequada beneficiará a instituição que desses serviços necessite.

1.4 RELEVÂNCIA DA PESQUISA

Qualidade de Serviços em redes TCP/IP vem ganhando grande importância atualmente, visto que muitos serviços utilizados precisam de que sejam garantidas algumas

características para uma boa operação. Serviços estes cada vez mais presentes no cotidiano de muitas empresas e pessoas.

Os recursos da rede TCP/IP são limitados, e por mais que se aumente a capacidade da rede, mais aumenta a demanda por novas aplicações e/ou novos recursos multimídia que necessitam de tratamento especial.

Assim é importante que se implemente Qualidade de Serviços na rede para que os novos serviços operem adequadamente.

1.5 ORGANIZAÇÃO DO TRABALHO

Este trabalho divide-se em quatro outros capítulos: no próximo, Referencial Teórico, a definição de Qualidade de Serviços e as diversas técnicas envolvidas em sua implementação serão apresentadas; no capítulo seguinte, Metodologia de Pesquisa, será apresentado o método utilizado para o desenvolvimento da monografia; no quarto capítulo, A Investigação, serão apresentados a instituição estudada e o ambiente de realização dos testes, também serão descritos em detalhe os métodos e ferramentas utilizados e os resultados obtidos; e, o último capítulo, Considerações Finais, descreverá as dificuldades e facilidades encontradas no decorrer da elaboração deste trabalho, apresentará propostas para trabalhos futuros e, ao final, uma conclusão sobre todo o trabalho.

2 REFERENCIAL TEÓRICO

2.1 INTRODUÇÃO

Qualidade de Serviços é um termo amplo que engloba um conjunto de técnicas que visam garantir, para as aplicações (ou serviços) classificadas como prioritárias, o controle de determinadas características das quais estas aplicações precisam para que sua operação seja adequada em diferentes tecnologias. Segundo Rodrigues (2005) estas características podem ser citadas como sendo:

- Disponibilidade do serviço (*availability*);
- Atraso fim-a-fim (*one way delay* – OWD);

  Tempo decorrido da ida do pacote do transmissor ao receptor mais a volta da resposta do receptor ao transmissor (*round trip time* – RTT).
- Variação do atraso (*jitter*);

  Variação da chegada dos pacotes ao receptor.
- Taxa de perda de pacotes;

  Quantidade de pacotes descartados ou que foram corrompidos durante seu trajeto na rede.
- Vazão (*throughput*).

  Número de *bits* que podem ser entregues com sucesso a cada segundo.

As técnicas de QoS tentam minimizar o atraso fim-a-fim, acabar com a variação de atraso e a perda de pacotes, e maximizar disponibilidade do serviço e a vazão. (RODRIGUES, 2005). As aplicações classificadas como prioritárias, devem receber tratamento especial, inclusive quando este tratamento especial prejudica o desempenho de outras menos

prioritárias. Sempre é bom ter em mente que a priorização de um determinado serviço só se dará em detrimento do desempenho de outros; é impossível priorizar tudo da mesma forma.

Para que seja melhor entendida a necessidade de QoS em redes TCP/IP, pode ser citado um exemplo: suponha dois serviços, um FTP (*File Transfer Protocol* – Protocolo de transferência de Arquivo) e o outro VoIP. Suponha também que, num determinado instante, vários usuários estejam fazendo uso do serviço FTP, causando a saturação da conexão. O que fará com que as chamadas de voz já existentes (ou as que ainda vão se estabelecer) tenham suas operações drasticamente prejudicadas pela perda de pacotes, longos atrasos e variações grandes destes atrasos. Com a implementação de uma boa política de QoS, ao tráfego de voz poderia ser reservada uma porcentagem da banda e também este tráfego poderia ser colocado em uma fila mais prioritária no(s) roteador(es) para garantir baixo atraso e baixas variações deste, garantindo a operação adequada do serviço VoIP.

Uma solução prática para este problema seria fornecer tanta capacidade de roteadores, tanto espaço de *buffers*[1] e tanta largura de banda que os pacotes simplesmente seriam transmitidos com enorme facilidade. Porém este tipo de solução esbarra num problema muito grande que é o custo. (TANENBAUM, 2003).

Neste capítulo serão abordados os seguintes tópicos: comportamento do protocolo TCP; os requisitos de QoS para as aplicações; os níveis de QoS; e a Arquitetura de Serviços Diferenciados.

2.2 COMPORTAMENTO DO PROTOCOLO TCP

O protocolo TCP (*Transmission Control Protocol*) é atualmente o protocolo de transporte mais utilizado na Internet. (VEGESNA, 2001). Assim, o estudo do seu comportamento torna-se muito importante para a criação de uma boa política de QoS.

[1] Memória para armazenamento temporário.

O TCP, sendo um protocolo que utiliza o conceito de janela deslizante, está sempre tentando aumentar a oferta de tráfego, aumentando a janela de transmissão a cada confirmação (ACK - *Acknowledgment*) que recebe do receptor, o que na maioria das vezes causa congestionamentos nas redes.

Segundo Stevens (1997), congestionamentos podem ocorrer quando dados chegam através de conexões de alta velocidade (rede de alta velocidade) e passam por conexões de baixa velocidade (rede de baixa velocidade), ou quando vários fluxos convergem num mesmo roteador e a capacidade do mesmo é menor que a soma da vazão desses fluxos.

Congestionamento é um fator de degradação do desempenho da rede como um todo, e a sua detecção se torna muito importante para que medidas possam ser tomadas a fim de minimizar ou até mesmo extinguí-los.

Perdas de pacotes causadas pela corrupção dos mesmos durante seu tráfego na rede são quase inexistentes. (STEVENS, 1997). Isto porque foram desenvolvidos meios físicos mais confiáveis como, por exemplo, as fibras ópticas.

Assim, as novas implementações do TCP assumem que a perda de pacotes está associada a descartes causados por um possível congestionamento na rede. Pois, quando um congestionamento ocorre, os *buffers* dos roteadores no caminho ficam totalmente ocupados, e como não há memória suficiente para os pacotes que chegaram com o *buffer* cheio, estes são descartados.

Segundo Stevens (1997), o TCP opera com quatro algoritmos: *slow start* (partida lenta), *congestion avoidance* (anti-congestionamento), *fast retransmit* (retransmissão rápida) e *fast recovery* (recuperação rápida).

Quando uma seção TCP é iniciada, o transmissor e o receptor trocam parâmetros através do *three-way-handshake*[2], sendo um destes parâmetros o *advertised receiver window*

[2] Negociação de parâmetros entre receptor e transmissor durante a abertura de uma conexão TCP.

(janela do receptor anunciada) ou *recwnd* (quantidade de *bytes* que o receptor pode receber num determinado espaço de tempo). Neste momento, o algoritmo que é executado é o *slow start*. Stevens (1997) afirma que este algoritmo adiciona dois parâmetros que o transmissor deve controlar, um chamado *congestion window* (janela de congestionamento) ou *cwnd*, que é a quantidade de *bytes* que o transmissor pode enviar num determinado espaço de tempo; e o outro chamado *slow start threshold* ou *ssthresh*, que limita o crescimento exponencial do parâmetro *cwnd*; o primeiro é iniciado com um segmento, tipicamente 536 ou 512 *bytes*, e o valor máximo de *bytes* que o transmissor poderá transmitir, será controlado pelo menor valor entre *recwnd* e *cwnd*, e, o *ssthresh*, iniciado com o valor 65535 *bytes*.

Em *slow start*, o transmissor inicia com a transmissão de um segmento, pois o *cwnd* inicialmente é igual a um. Supondo que esse segmento chegue corretamente ao receptor, este transmitirá uma confirmação (ACK). Também supondo que essa confirmação chegue corretamente ao transmissor, este incrementará o valor de *cwnd* em um segmento, fazendo com que o valor de *cwnd* passe de um para dois. Agora o transmissor poderá transmitir dois segmentos, e, quando receber os dois ACK*s* correspondentes, poderá incrementar o valor de *cwnd* em dois segmentos, passando de dois para quatro. Assim, segundo Stevens (1997), o valor de *cwnd* crescerá exponencialmente até que alcance o *ssthresh*.

Quando isso ocorre o algoritmo *congestion avoidance* entra em execução, e, de acordo com Stevens (1997), o *cwnd* é incrementado com o seguinte valor a cada ACK recebido: tamanho do segmento vezes o tamanho do segmento dividido por *cwnd*, fazendo com que o *cwnd* cresça linearmente, pois a cada RTT (*round-trip time*[3]) o *cwnd* é incrementado de um segmento, até que um congestionamento seja detectado, já no *slow start*, o *cwnd* é incrementado pelo número de ACKs recebidos a cada RTT.

[3] Refere-se ao tempo que decorre do envio de uma informação, da sua chegada ao seu destino, do envio da resposta do destino para a origem e da chegada desta resposta à origem.

Congestionamentos são detectados pelo TCP através de *time-outs* (tempo expirado) ou da recepção de ACKs duplicados.

Time-outs ocorrem quando um pacote é transmitido e sua confirmação não é enviada, pois o receptor não recebeu o pacote, ou quando a confirmação é enviada, mas não é recebida pelo transmissor, pois se perdeu na rede, ou quando a confirmação é enviada, mas não chega ao transmissor no tempo determinado. Pois quando cada pacote é transmitido, um cronômetro é disparado e a confirmação deste deverá chegar até esse cronômetro atingir um limite, que é calculado durante toda a conexão, do contrário ocorrerá um *time-out* e o pacote deverá ser retransmitido.

A recepção de ACKs duplicados ocorre quando vários pacotes são transmitidos e o receptor deixa de receber um dos pacotes, mas recebe seus subseqüentes. A cada pacote subseqüente que o receptor recebe, envia uma confirmação informando que ele está "esperando" pelo pacote perdido.

Quando três ACKs duplicados são recebidos pelo transmissor, o algoritmo *fast retransmit* entra em execução e faz com que o *ssthresh* receba a metade do valor de *cwnd* atual, o segmento esperado pelo receptor seja transmitido antes que o *time-out* deste ocorra, e o *cwnd* recebe o novo valor de *ssthresh* somado ao valor do tamanho de três segmentos. Depois que o *fast retransmit* termina sua execução, o *fast recovery* entra em execução, utilizando os mesmos valores de *ssthresh* e *cwnd* calculados no *fast retransmit*, e o TCP opera como no *congestion avoidance*, ou seja, incrementando *cwnd* de um segmento a cada RTT. Quando ocorre um *time-out*, o TCP volta a operar com o algoritmo *slow start*, com o *cwnd* igual a um segmento e *ssthresh* igual ao seu último valor obtido nos algoritmos acima. (STEVENS, 1997).

Como conseqüência, a vazão do fluxo TCP tem seu comportamento como mostrado na figura a seguir, ou seja, parecendo um dente de serra.

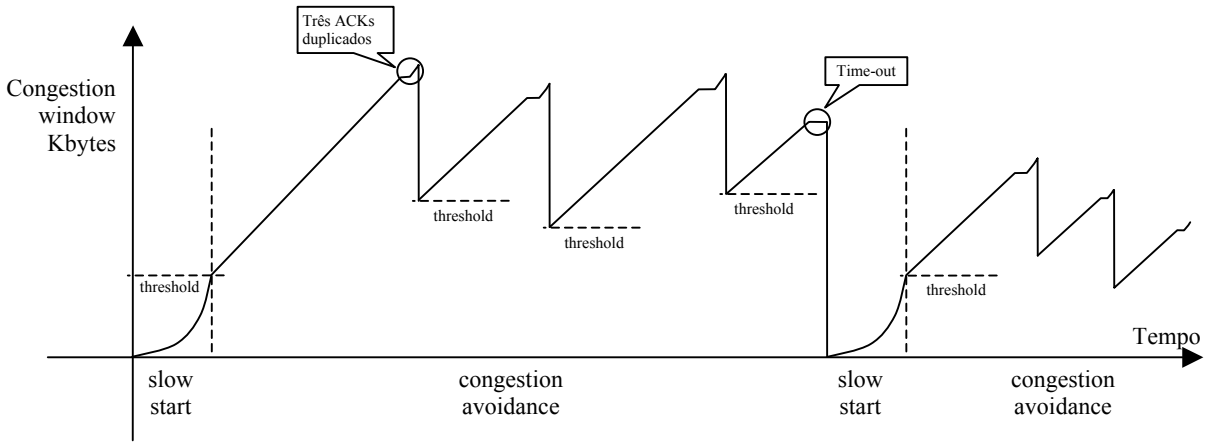


Figura 1 – Comportamento do TCP perante congestionamentos. Fonte: Rodrigues (2005).

Um fenômeno conhecido como sincronização global, ocorre em uma rede com alta utilização, pois os tráfegos chegam em rajadas nos roteadores causando falta de *buffers*, e, conseqüentemente, descartes simultâneos ocorrem nas conexões TCP que compartilham o gargalo, fazendo com que todas reajam ao mesmo tempo e recuem diminuindo o tráfego. Instantes depois, todas se recuperam, causando novamente a falta de *buffers*, e, novamente, com a falta de *buffers*, todas recuam. A sincronização global faz com que o tráfego no enlace oscile, a utilização do *link* caia e a vazão individual de cada conexão TCP fique abaixo da fatia equânime (*fair share*), causando ineficiência na utilização do enlace. Outro fato que ocorre nessa situação é que algumas conexões recebem mais do que outras por um período longo de tempo (desigualdade na captura de banda). (MORRIS, 1997).

2.2.1 **O Algoritmo RED**

Braden, et. al. (1998) afirma que os mecanismos para evitar congestionamentos descritos por Stevens (1997), apesar de serem poderosos e necessários, não são suficientes para prover um bom serviço sob todas as circunstâncias.

O mecanismo tradicional de gerenciamento de filas nos roteadores é configurar uma quantidade máxima de pacotes que podem ser aceitos por cada fila, aceitar pacotes até que se

atinja o tamanho máximo da fila e quando isso acontecer, os pacotes subseqüentes são descartados até que um ou mais pacotes que estejam ocupando a fila sejam transmitidos, fenômeno conhecido como *"tail drop"*. Esse mecanismo possui duas desvantagens: a primeira é que, em algumas situações, um ou poucos fluxos podem monopolizar o espaço da fila e a segunda é que as filas trabalham cheias ou quase cheias por um longo período de tempo, e, assim, na ocorrência de rajadas, situação comum na Internet, descartes múltiplos ocorrerão, resultando na sincronização global. (BRADEN, et. al., 1998).

Assim o RED (*Random Early Detection* – Detecção antecipada aleatória.) é proposto por Braden, et. al. (1998), sendo um algoritmo para gerenciamento ativo de filas nos roteadores que evita o monopólio de filas para certos fluxos e a utilização máxima de *buffers*, e, conseqüentemente, a sincronização global, através de descartes antecipados à ocupação máxima do *buffer* e aleatórios através de cálculos probabilísticos.

O RED consiste de duas partes: uma para estimar o tamanho médio da fila e, a outra, para decidir se um pacote que chega na fila será descartado ou não (BRADEN, et. al., 1998), e, neste último algoritmo, são definidos dois limiares min_{th} (*minimum threshold*) e max_{th} (*maximum threshold*). (FLOYD, JACOBSON, 1993).

A utilização média da fila (*avg* – *average*) é calculada e comparada com os limiares min_{th} e max_{th}. Os pacotes que chegam com *avg* menor que min_{th}, não serão marcados para descarte. Todos os pacotes que chegam *avg* maior que max_{th}, são descartados com probabilidade um (1), ou seja, cem por cento. Pacotes que chegam *avg* maior que min_{th} e menor que max_{th} são marcados ou descartados de acordo com a probabilidade p_a, como mostra a figura a seguir. (FLOYD, JACOBSON, 1993).

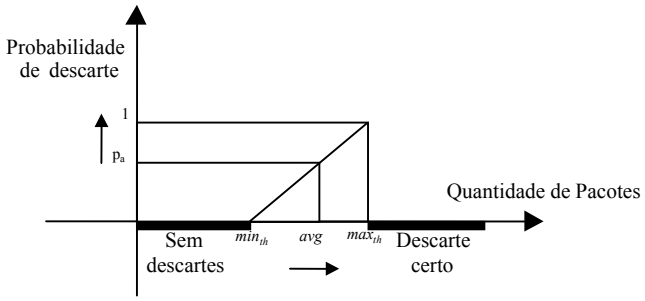


Figura 2 – O algoritmo RED. Fonte: Rodrigues (2005)

O RED ainda dispõe de uma variável chamada *count* que garante que não se espera muito tempo até marcar ou descartar um pacote, pois ele conta a quantidade de pacotes que passaram (que não foram marcados ou descartados) desde a última marcação ou descarte. Isso faz com que os pacotes sejam marcados ou descartados em intervalos com distribuição uniforme, evitando a sincronização global. (RODRIGUES, 2005).

Uma variação do RED, conhecida como WRED (*Weighted* RED), garante que os pacotes com mais prioridade, de acordo com a precedência IP, sejam marcados ou descartados com menos rigor do que outros com menos prioridade. (RODRIGUES, 2005). Um exemplo de WRED é ilustrado a seguir.

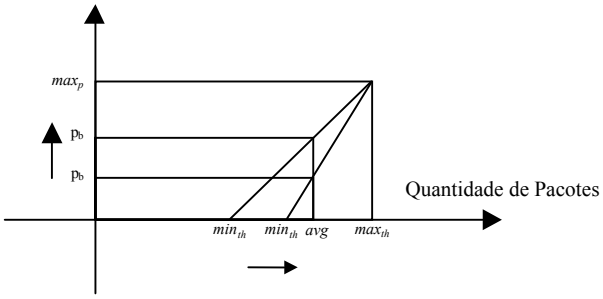


Figura 3 – O algoritmo WRED. Fonte: Rodrigues (2005)

2.3 REQUISITOS DE QOS PARA AS APLICAÇÕES

Atualmente na Internet existem diferentes tipos de aplicações, tais como transferência de arquivos, vídeo sob demanda, VoIP, entre outros, que necessitam de diferentes requisitos para obterem um funcionamento adequado.

Qualidade de serviços requer que um conjunto de requisitos quantitativos, como os vistos anteriormente (disponibilidade de serviço, atraso fim-a-fim, variação de atraso, taxa de perdas de pacotes e vazão) sejam respeitados. Porém, segundo Braden, Clarck e Shenker (1994), esses requisitos quantitativos estão limitados ao atraso máximo e mínimo, e, o grau que o desempenho de cada aplicação depende de um serviço de baixo atraso varia enormemente.

Assim, uma série de distinções qualitativas entre as aplicações podem ser feitas baseadas no grau de dependência das aplicações. Uma classe de aplicações, conhecida como aplicações de tempo real, precisa que os dados de cada pacote cheguem num determinado tempo, caso contrário, os dados se tornam inúteis. Outra classe, conhecida como aplicações elásticas, sempre esperarão pela chegada dos dados (BRADEN, CLARCK e SHENKER, 1994). A figura a seguir mostra a classificação das aplicações segundo Braden, Clarck e Shenker (1994).

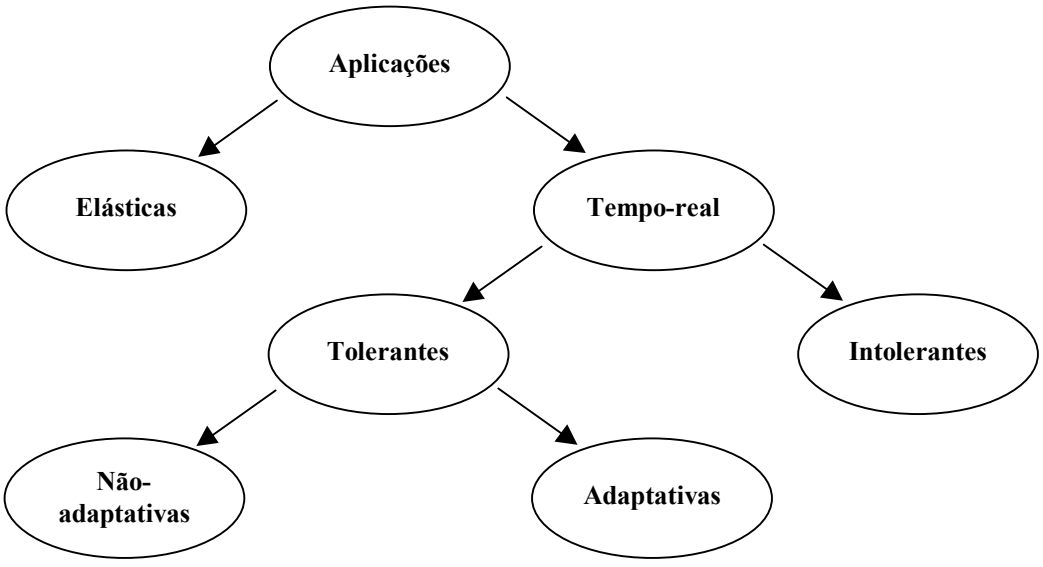


Figura 4 – A classificação das aplicações de acordo com seus requisitos.

As aplicações de tempo-real são aplicações sensíveis ao tempo, onde o atraso de chegada dos pacotes ao destino deve ser minimizado ao máximo.

O desempenho de uma aplicação de tempo-real pode ser medido através de dois parâmetros: atraso e fidelidade. (BRADEN, CLARCK e SHENKER, 1994).

Algumas aplicações de tempo-real permitem algum atraso na entrega de pacotes e são conhecidas como tolerantes (Exemplo: áudio e vídeo sob demanda). Outras, que necessitam da interação entre o transmissor e o receptor, não permitem estes atrasos e são conhecidas com intolerantes (Exemplo: VoIP), porém todas são sensíveis à perda de fidelidade em sua qualidade. (BRADEN, CLARCK e SHENKER, 1994).

As aplicações tolerantes podem ser: adaptativas e não adaptativas. As adaptativas têm a propriedade de se adaptarem aos atrasos de pacotes devido às condições da rede. Essa capacidade de adaptação fica limitada ao nível de perda de fidelidade permitido pelo usuário. (BRADEN, CLARCK e SHENKER, 1994).

E, finalmente, as aplicações elásticas são aquelas que admitem maiores atrasos na chegada de pacotes ao destino, porém elas não podem ser consideradas insensíveis a atrasos, pois atrasos muito grandes podem degradar o desempenho dessas aplicações. (BRADEN, CLARCK e SHENKER, 1994).

A figura 9 mostra alguns exemplos de aplicações e seus respectivos requisitos.

Aplicação	**Confiabilidade**	**Retardo**	**Variação de Atraso**	**Vazão**
Correio Eletrônico	Alta	Baixa	Baixa	Baixa
Transferência de Arquivos	Alta	Baixa	Baixa	Média
Acesso à Web	Alta	Média	Baixa	Média
Login remoto	Alta	Média	Média	Baixa
Áudio por demanda	Baixa	Baixa	Alta	Média
Vídeo por demanda	Baixa	Baixa	Alta	Alta
Telefonia	Baixa	Alta	Alta	Baixa
Videoconferência	Baixa	Alta	Alta	Alta

Figura 5 – Rigidez dos requisitos de qualidade de serviços. Fonte: Tanenbaum (2003)

2.4 NÍVEIS DE QOS

Como visto na seção anterior, as aplicações possuem diferentes requisitos de QoS para que sua operação seja adequada.

Vegesna (2001) diz que, entender os diferentes tipos de aplicações existentes, é a chave para entender os diferentes níveis de QoS que os fluxos de dados precisam em uma rede.

Segundo Vegesna (2001), a capacidade da rede de garantir o controle dos requisitos de QoS das aplicações é dividida em três níveis de serviço: Serviço de melhor-esforço (*Best-effort service*); Serviço diferenciado (*Differentiated service*); e Serviço garantido (*Garanteed service*).

- Serviço de melhor-esforço

É o tradicional meio de entrega de pacotes que foi originalmente concebido com o protocolo TCP/IP. Caracteriza-se pela não garantia de quando ou se o pacote será entregue ao seu destino. (VEGESNA, 2001). Por este motivo, Vegesna (2001) afirma que este tipo de serviço não é considerado parte dos serviços de QoS.

- Serviço diferenciado

Neste tipo de serviço, os fluxos de pacotes são agrupados em classes baseadas em seus requisitos de serviço. Cada classe de tráfego será tratada de acordo com a QoS que estiver configurada para cada uma. (VEGESNA, 2001).

O serviço diferenciado não dá garantias de cumprimento dos requisitos de QoS por si só. Ele somente diferencia os fluxos e permite um tratamento especial para os fluxos prioritários sobre os outros. (VEGESNA, 2001).

- Serviço garantido

Requer que a rede estabeleça uma reserva dos recursos que as aplicações necessitam, antes que a comunicação aconteça. Segundo Vegesna (2001), este tipo de serviço também requer que rígidos requisitos de serviço sejam garantidos pela rede.

2.5 ARQUITETURA DE SERVIÇOS DIFERENCIADOS

O maior objetivo de QoS é desempenhar serviços garantidos e diferenciados na internet ou em qualquer outra rede baseada no protocolo IP. (VEGESNA, 2001).

Nesta seção, será discutida a Arquitetura de Serviços Diferenciados, conhecida como *Differentiated Service Architecture* ou *DiffServ Architecture*, responsável, como o próprio nome sugere, pelos serviços diferenciados.

Esta arquitetura define o campo DS (*Differentiated Service*) que substituirá os campos TOS do cabeçalho do protocolo IP versão quatro e o *Traffic Class* (Classe de Tráfego) do cabeçalho do protocolo IP versão seis. (NICHOLS, et. al., 1998).

Nichols, et. al. (1998), afirma que os seis *bits* do campo DS são utilizados como um código (DSCP – *DiffServ Code Point*), para selecionar a forma como o pacote será tratado em cada nó da rede, como pode ser visto na figura a seguir. Os dois últimos *bits*, o seis e o sete, são ignorados pelos nós que possuem suporte a serviços diferenciados quando determinam o comportamento nó-a-nó (*per-hop behavior*) a ser aplicado a cada pacote recebido.

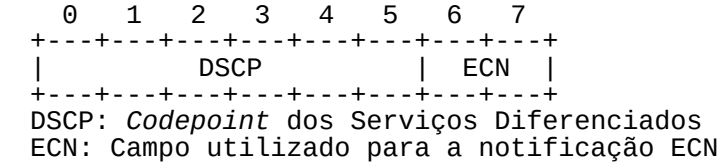


Figura 6 – O campo DSCP no cabeçalho IP. Adaptado de Rodrigues (2005)

Segundo Nichols, et. al. (1998), o *codepoint* (código DSCP) deverá ser utilizado como um índice de uma tabela de associação a um determinado PHB (*Per-Hop Behavior*). E cada valor do *codepoint* poderá está associado a um PHB, que poderá ser diferente em diferentes domínios administrativos DS, que, de acordo com Blake, et. al. (1998), são uma ou mais redes sob a mesma administração onde os nós DS operam com uma política de provisão de serviços comum e um conjunto de grupos PHB implementados em cada nó.

Cada domínio DS deve ter um *codepoint* padrão, geralmente o PHB melhor-esforço é representado com o DSCP igual a "000000". Se um nó recebe um pacote com *codepoint* inválido, ou seja, que não existe um PHB associado, o pacote deverá receber um tratamento equivalente ao comportamento padrão do domínio DS, mas não deverá ter seu DSCP remarcado para o DSCP padrão ou qualquer outro. (NICHOLS, et. al., 1998).

Foram criadas oito classes de serviços representadas com DSCP igual a "XXX000", onde "XXX" são números binários que variam de 000 (0 em decimal) a 111 (7 em decimal), visando compatibilizar com o uso do campo de precedência IP (IP *precedence*), onde apenas os três *bits* mais significativos são analisados por roteadores que não suportam *DiffServ*. Assim, os roteadores antigos tratarão a prioridade do pacote analisando apenas os três *bits* mais significativos, que corresponderão a precedência IP, e, por outro lado, os roteadores que suportam *DiffServ* operarão normalmente usando o DSCP integralmente. (RODRIGUES, 2005).

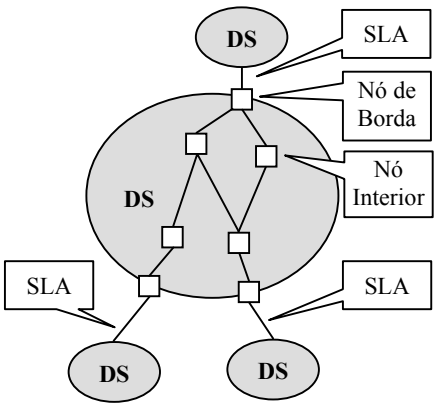


Figura 7 – Arquitetura de Serviços Diferenciados. Adaptado de Rodrigues (2005).

No exemplo descrito na figura 11, pode ser notada a presença de um domínio *DiffServ* ligado a três outros domínios DS através de SLAs (*Service Level Agreement* – Acordo de Nível de Serviço), que, segundo Rodrigues (2005), são acordos de serviços entre domínios DS, que especificam as características de desempenho requeridas pelo serviço oferecido bem como o TCA (*Traffic Conditioning Agreement* – Acordo de Condicionamento de Tráfego), que especifica a classificação dos pacotes, a política de marcação, os perfis de tráfego permitidos e as ações que serão aplicadas aos fluxos dentro e fora destes perfis.

São observados também no mesmo exemplo, dois tipos de nós: os interiores e os de borda. Cada um desempenha uma função diferente.

Os nós de borda podem ser classificados ainda como sendo nós de entrada (*Ingress Node*) ou nós de saída (*Egress Nodes*). Esta classificação dependerá da direção do tráfego, isto é, se o tráfego está entrando em um domínio DS, o nó que o está recebendo é classificado como de entrada, e, logicamente, se estiver saindo, o nó é classificado como de saída. (BLAKE, et. al., 1998).

Os nós de entrada são responsáveis pela classificação e verificação do cumprimento dos acordos de condicionamento de tráfego (TCA) entre os domínios DS. Os de saída podem aplicar condicionamento no tráfego em direção a um domínio parceiro, de acordo com o TCA acordado. (BLAKE, et. al., 1998).

Já os nós interiores podem desempenhar funções limitadas de condicionamento de tráfego como, por exemplo, remarcação de um código DS. (BLAKE, et. al., 1998).

Na Arquitetura de Serviços Diferenciados são identificados quatro elementos: classificadores, perfis de tráfego, condicionadores e comportamento nó-a-nó (PHB).

O classificador de pacotes tem por objetivo identificar os fluxos de dados que receberão um determinado serviço diferenciado, sendo mapeados em um dos PHBs oferecidos pelo domínio DS no qual irão trafegar (RODRIGUES, 2005) e, segundo Blake, et. al. (1998),

podem ser de dois tipos: o BA (*Behavior Aggregate* – comportamento de um agregado) e o MF (*Multi-Field* – multi-campo). A diferença entre eles é o que o BA classifica os pacotes levando em consideração somente o DS *codepoint*, já o MF leva em consideração a combinação de um ou mais campos do cabeçalho, como por exemplo o endereço de origem, o de destino, o campo DS, a identificação do protocolo, portas de origem e destino, e outras informações como a interface de entrada.

O perfil de tráfego define as propriedades temporais de um fluxo de tráfego selecionadas pelo classificador e define regras para determinar se um pacote está dentro ou fora do perfil. (Blake, et. al., 1998).

O condicionador é responsável pela medição, marcação, moldagem e policiamento do tráfego que entra e sai de um domínio DS. (RODRIGUES, 2005). Segundo Heinanen e Guerin (1999), dois algoritmos podem ser usados como componentes de um condicionador de tráfego DS, o SRTCM (seção 2.5.1) e o TRTCM (seção 2.5.2).

A figura a seguir mostra um diagrama do bloco classificador e condicionador de tráfego.

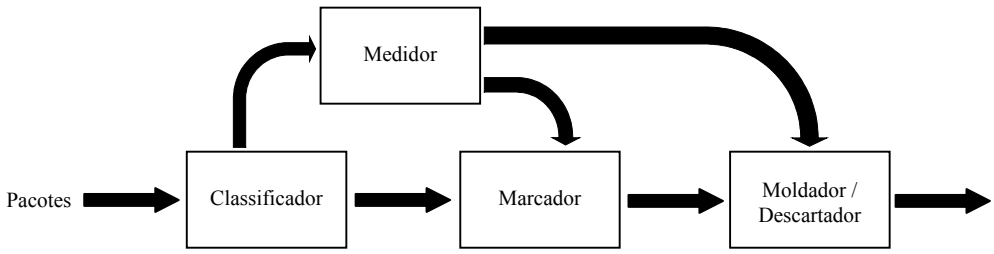


Figura 8 – Classificador e Condicionador de Tráfego. Adaptado de Blake, et. al. (1998)

O medidor checa a aderência aos parâmetros de tráfego e informa o resultado ao marcador ou moldador/descartador para que estes disparem ações para pacotes dentro ou fora do perfil. O marcador escreve ou reescreve o valor do código DSCP. Os moldadores atrasam alguns ou todos os pacotes de um tráfego para estarem em conformidade com o perfil e os

descartadores descartam alguns ou todos os pacotes de um tráfego que excedem o perfil, operação conhecida como policiamento. (VEGESNA, 2001).

O comportamento nó-a-nó (PHB) é a descrição do comportamento, observável externamente, de um encaminhamento (tais como: perda, atraso, variações de atraso, etc) dado a um agregado por um determinado nó DS, ou seja, é a maneira pela qual cada nó DS aloca recursos (banda, buffers, etc) a um determinado tráfego ou agregado. (Blake, et. al., 1998).

Segundo Blake, et. al. (1998), PHB é implementado nos nós DS através de alguns mecanismos de gerenciamento de *buffer* e de escalonamento de pacotes. Alguns dos mecanismos de gerenciamento de *buffer* já foram discutidos neste trabalho, e podem ser citados como sendo: *tail drop*, RED e WRED.

Nas seções 2.5.3 a 2.5.7, serão discutidos alguns dos mecanismos de escalonamento de pacotes onde, alguns dos mesmos, foram utilizados nos testes de QoS propostos neste trabalho.

2.5.1 **Algoritmo SRTCM**

O algoritmo SRTCM (*Single Rate Three Color Marker* – marcador de três cores baseado em uma taxa) mede um fluxo de pacotes IP e marca os pacotes com as cores verde, amarelo e vermelho de acordo com três parâmetros de tráfego, o CIR (*Committed Information Rate* – taxa média em *bytes* por segundo acordada), o CBS (*Committed Burst Size* – rajada máxima em *bytes* acordada) e o EBS (*Excess Burst Size* – rajada máxima em *bytes* tolerada). (HEINANEN e GUERIN, 1999a).

Segundo Heinanen e Guerin (1999), os pacotes são marcados com a cor verde se não excederem a taxa CBS, com a cor amarelo se excederem a taxa CBS mas não excederem a taxa EBS e vermelho se excederem as duas.

Este algoritmo usa dois baldes de fichas (*tokens*) que são servidos de fichas na taxa CIR: o Balde C (Bc), que tem o tamanho igual a CBS e o Balde E (Be), que tem o tamanho igual a EBS. No início da operação do algoritmo, os baldes estão cheios e somente são aceitos pacotes com o tamanho máximo igual ou menor ao valor máximo entre CBS e EBS. (RODRIGUES, 2005). Por isso, Heinanen e Guerin (1999a), recomendam que o valor de CBS ou EBS seja maior ou igual ao maior pacote que possivelmente seja encontrado no fluxo.

O medidor do algoritmo opera de dois modos: o modo *Color-Blind* (insensível à cor), que assume que os pacotes chegam sem cor; e, o modo *Color-Aware* (sensível à cor), que assume que algum processo anterior já coloriu os pacotes que chegam. (HEINANEN e GUERIN, 1999a).

No modo *Color-Blind*, quando um pacote de B *bytes* chega no momento T e se Tc (quantidade de fichas no Balde Bc), no momento T, menos B é maior ou igual a zero, o pacote é marcado de verde e Tc é decrementado de B fichas. Senão, se Te (quantidade de fichas no Balde Be) no momento T menos B é maior ou igual a zero, o pacote á marcado de amarelo e Te é decrementado de B fichas. Senão o pacote é marcado de vermelho e os baldes ficam inalterados, como pode ser visto na figura a seguir. (HEINANEN e GUERIN, 1999a).

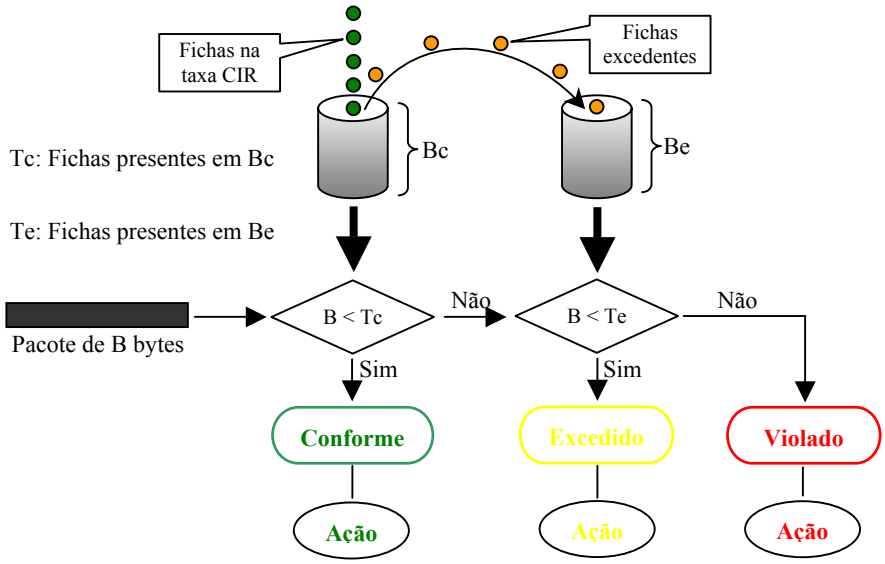


Figura 9 – Funcionamento do SRTCM no modo *Color-Blind*. Adaptado de Rodrigues (2005)

Já no modo *Color-Aware*, quando um pacote de B *bytes* chega no momento T e se o pacote é verde e Tc, no momento T, menos B é igual ou maior que zero, o pacote permanece verde e Tc é decrementado de B fichas. Senão, se o pacote é verde ou amarelo e Te, no momento T, menos B é igual ou maior que zero, o pacote é marcado de amarelo e Te é decrementado de B fichas. Senão, o pacote é marcado de vermelho e os baldes ficam inalterados. (HEINANEN e GUERIN, 1999a).

2.5.2 **Algoritmo TRTCM**

O algoritmo TRTCM (*Two Rate Three Color Marker* – marcador de três cores baseado em duas taxas) mede um fluxo de pacotes IP e marca os pacotes com as cores verde, amarelo e vermelho de acordo com quatro parâmetros de tráfego, o PIR (*Peak Information Rate* – taxa máxima de pico em *bytes* por segundo), o PBS (*Peak Burst Size* – tamanho máximo da rajada em *bytes*) o CIR (*Committed Information Rate* – taxa média em *bytes* por segundo acordada) e o CBS (*Committed Burst Size* – rajada máxima em *bytes* acordada). (HEINANEN e GUERIN, 1999b).

Segundo Heinanen e Guerin (1999b), os pacotes são marcados com a cor vermelha se excederem a taxa PIR, senão são marcados com a cor amarelo ou verde dependendo se eles excedem ou não a taxa CIR respectivamente.

Este algoritmo usa dois baldes de fichas (*tokens*): o Balde P (P), que tem o tamanho igual a PBS e é servido de fichas na taxa PIR, e o Balde C (C), que tem o tamanho igual a CBS e é servido de fichas na taxa CIR. No início da operação do algoritmo, os baldes estão cheios de fichas. (RODRIGUES, 2005). Heinanen e Guerin (1999b) recomendam que o valor de PBS e CBS seja maior ou igual ao maior pacote que possivelmente seja encontrado no fluxo.

O medidor do algoritmo opera de dois modos: o modo *Color-Blind* (insensível à cor), que assume que os pacotes chegam sem cor; e, o modo *Color-Aware* (sensível à cor), que assume que algum processo anterior já coloriu os pacotes que chegam. (HEINANEN e GUERIN, 1999b).

No modo *Color-Blind*, quando um pacote de B *bytes* chega, no momento T, e se Tp (quantidade de fichas no Balde P), no momento T, menos B é menor que zero, o pacote é marcado de vermelho e os baldes ficam inalterados. Senão, se Tc (quantidade de fichas no Balde C), no momento T, menos B é menor que zero, o pacote á marcado de amarelo e Tp é decrementado de B fichas. Senão, o pacote é marcado de verde e ambos os baldes são decrementados de B fichas, como pode ser visto na figura a seguir. (HEINANEN e GUERIN, 1999b).

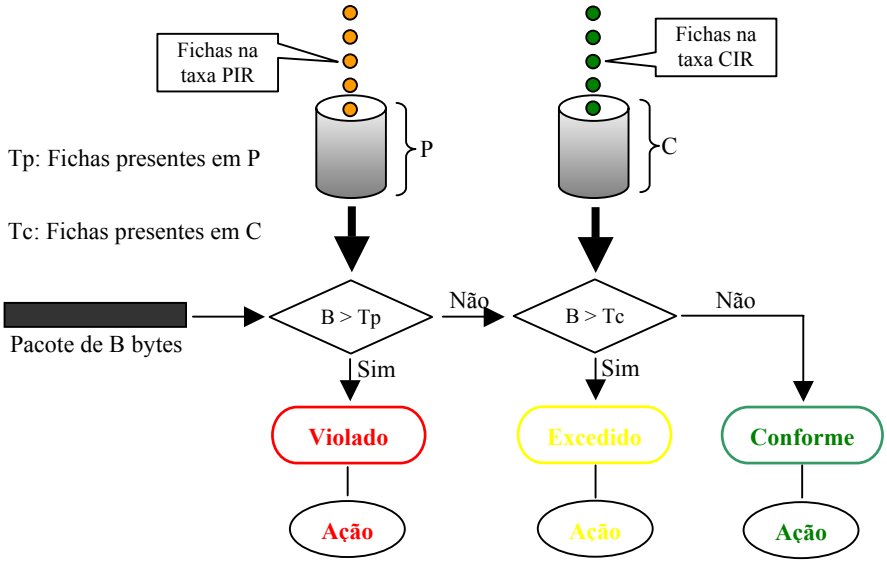


Figura 10 – Funcionamento do TRTCM no modo *Color-Blind*.

No modo *Color-Aware*, quando um pacote de B *bytes* chega no momento T e se o pacote é vermelho ou Tp no momento T menos B menor que zero, o pacote é marcado de vermelho e baldes ficam inalterados. Senão, se o pacote é amarelo ou Tc no momento T menos B é menor que zero, o pacote é marcado de amarelo e Tp é decrementado de B fichas.

Senão, o pacote é marcado de verde e ambos os baldes são decrementados de B fichas. (HEINANEN e GUERIN, 1999b).

2.5.3 **Escalonamento FIFO (*First In First Out)***

O escalonamento FIFO (primeiro a entrar, primeiro a sair), também conhecido como FCFS (*First Come First Served* – primeiro a chegar, primeiro a ser servido), serve os pacotes na ordem de chegada, isto é, a ordem de saída da fila FIFO é a mesma ordem de saída.

É o mecanismo de escalonamento mais comumente encontrado nos roteadores hoje. Caracteriza-se por não possuir nenhum método de diferenciação entre os fluxos e, conseqüentemente, não realiza priorização entre eles, como pode ser visto na figura a seguir. (VEGESNA, 2001).

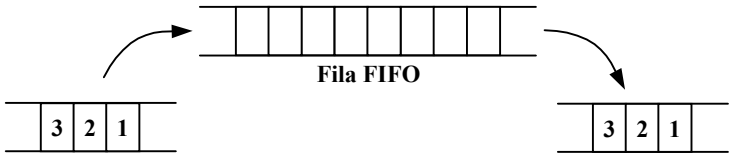


Figura 11 – Escalonamento FIFO. Fonte: Vegesna (2001).

2.5.4 **Escalonamento FQ (*Fair Queuing*)**

O escalonamento FQ (escalonamento justo) serve os pacotes de acordo com um número de seqüência (SN – *Sequence Number*) que o pacote recebe ao entrar na fila FQ.

Os números de seqüência definem a ordem relativa em que os pacotes serão servidos pelo escalonamento FQ. (VEGESNA, 2001).

Estes números são calculados da seguinte forma: um parâmetro chamado número do ciclo é utilizado para representar quantos ciclos *byte* a *byte* o escalonador já completou. Pacotes chegam em um fluxo inativo, ou seja, fila de espera vazia, recebem um SN igual ao valor do seu tamanho em *bytes* somado ao número do ciclo no qual chegou, já os pacotes que

chegam em um fluxo ativo, ou seja, fila de espera ocupada, recebem um SN igual ao valor do seu tamanho em *bytes* somado ao maior SN dos pacotes da fila a que fora destinado. O parâmetro número do ciclo é atualizado pelo escalonador FQ de acordo com o SN do último pacote transmitido ou preparado para transmissão. (RODRIGUES, 2005).

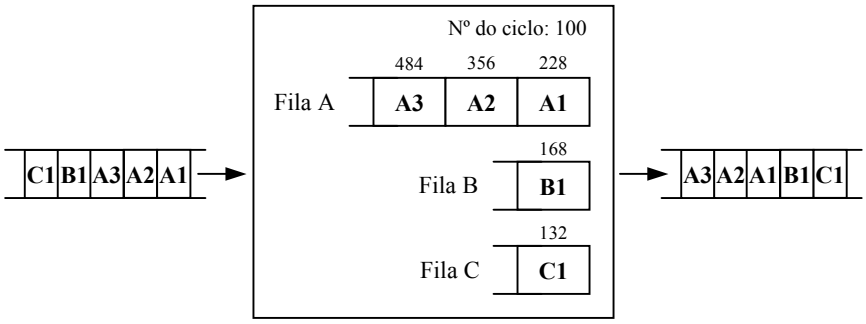


Figura 12 – Escalonamento FQ. Fonte: Vegesna (2001).

A figura acima mostra um exemplo de um escalonador FQ. Os pacotes do fluxo A (A1, A2 e A3) têm seu tamanho igual a 128 *bytes*, os do fluxo B (B1) têm seu tamanho igual a 68 *bytes* e os do fluxo C (C1) têm seu tamanho igual a 32 *bytes*. Todos os pacotes de todos os fluxos chegaram ao escalonador FQ no ciclo de número 100. Os pacotes A1, B1 e C1 encontraram suas respectivas filas vazias e receberam o SN igual aos seus respectivos tamanhos em *bytes* somados ao número do ciclo, que no exemplo é igual a 100. Já o pacote A2 encontrou sua fila ocupada pelo pacote A1 e recebe um SN igual ao seu tamanho em *bytes* somado ao número do SN do pacote A1. O mesmo aconteceu com o pacote A3 que recebeu seu SN de acordo com seu tamanho e com o SN do pacote A2.

O pacote C1, por ter o menor SN entre todos os pacotes, será o primeiro a ser transmitido pelo escalonador e o número do ciclo será atualizado de acordo com o SN de C1, ou seja, passará a ser igual a 132. Supondo que um pacote D1, de tamanho igual a 100 *bytes*, chegue neste instante, receberá um SN igual ao valor do seu tamanho em *bytes* somado ao novo número do ciclo. Assim seu SN será igual a 232.

Nota-se que neste tipo de escalonamento pacotes de tamanho pequeno recebem um SN menor que outro com um tamanho grande e, como conseqüência, são transmitidos antes.

Com a finalidade de priorizar os dados mais importantes (críticos) em relação a outros foi introduzida uma variação do escalonamento FQ chamada de WFQ (*Weighted Fair Queuing* – escalonamento justo baseado em peso).

Neste escalonamento, pesos são atribuídos por fluxo sendo cada fluxo servido na proporção do seu peso. Os pesos são atribuídos baseando-se na precedência do cabeçalho IP de cada pacote e os números SN que cada pacote receberá dependerão do peso atribuído a cada pacote. (VEGESNA, 2001).

Como exemplo de cálculo dos pesos do escalonamento WFQ, será utilizada a implementação da empresa Cisco. Segundo Vegesna (2001), nos roteadores Cisco com a versão do IOS anterior a 12.0(5)T os pesos têm o valor igual a 4096, já os roteadores com versão do IOS igual ou superior a 12.0(5)T os pesos têm o valor igual a 32768.

No escalonamento WFQ os pesos são calculados da seguinte forma: valor do peso (4096 ou 32768, no exemplo da Cisco) dividido pela precedência IP mais 1. (VEGESNA, 2001). Então, se é utilizado o valor 32768 para o peso e o pacote tem a precedência IP igual a 5, este pacote receberá um peso igual a 5461.

Para calcular o valor do SN que um pacote receberá neste escalonamento, procede-se da mesma maneira que no escalonamento FQ (com a única diferença a introdução do peso), isto é, para um pacote que encontra sua fila vazia multiplica-se o peso (calculado como descrito anteriormente) pelo valor do tamanho do pacote em *bytes* e soma-se ao número do ciclo. Para um pacote que encontra sua fila cheia, multiplica-se o peso (calculado como descrito anteriormente) pelo valor do tamanho do pacote em *bytes* e soma-se ao maior SN dos pacotes da fila a que foi destinado. Como pode ser visto na figura a seguir.

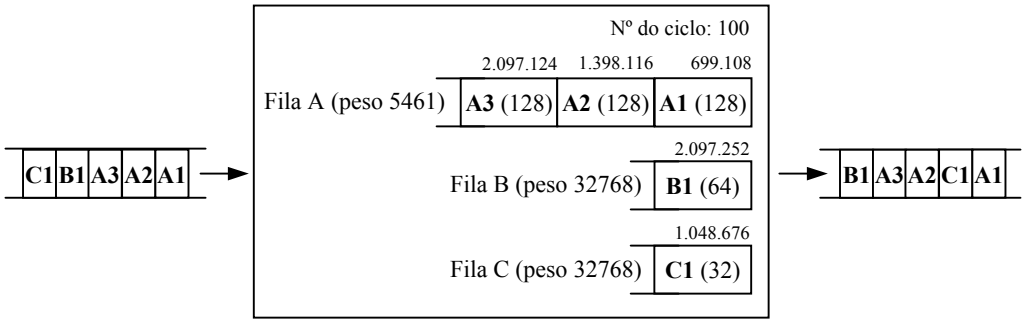


Figura 13 – Escalonamento WFQ. Fonte: Rodrigues (2005).

A figura acima mostra uma implementação do escalonamento WFQ com um peso igual a 32768. Os pacotes do fluxo A possuem uma precedência IP igual a 5, assim seus pesos têm o valor igual a 5461; já os pacotes do fluxo B e C possuem a precedência IP igual a 0 e, conseqüentemente, seus pesos têm o valor igual a 32768.

2.5.5 **Escalonamento PQ (*Priority Queuing*)**

No escalonamento PQ (escalonamento prioritário), o tráfego entrante é classificado e enfileirado nas filas PQ. Esta classificação pode levar em conta o tipo de protocolo, a interface de entrada, o tamanho dos pacotes, fragmentos ou listas de acesso. Pacotes que não são classificados para a fila PQ são colocados em outras filas, tais como: FIFO, WFQ, entre outras. (RODRIGUES, 2005).

O escalonamento PQ funciona da mesma maneira que o FIFO, sendo que todo o tráfego que se encontra numa fila PQ é servido até a exaustão, ou seja, se existirem pacotes na fila PQ, nenhum outro pacote de outras filas é servido até que a fila PQ se esvazie; e, um pacote que chega em uma fila PQ vazia, esperará somente o tempo de uma transmissão em andamento para ser transmitido.

Vegesna (2001) afirma que nos roteadores Cisco foram implementadas quatro filas PQ, uma de alta prioridade chamada de *high***,** uma de prioridade média chamada de *medium***,** uma normal chamada de *normal* e outra de baixa prioridade chamada de *low*.

Pacotes enfileirados na fila *high* são transmitidos primeiro, os enfileirados na fila *medium* são transmitidos em seguida e assim por diante. (VEGESNA, 2001).

2.5.6 **Escalonamento CQ (*Custom Queuing*)**

O escalonamento CQ (escalonamento personalizado) mantém filas separadas para cada classe de tráfego. A cada fila é alocado um número de *bytes* que poderão ser transmitidos por cada uma, o que garante uma banda mínima por fila. O escalonador CQ circula entre as filas e apanha o número de *bytes* configurado para a fila, se a mesma estiver vazia, passa para a próxima. (RODRIGUES, 2005).

Por exemplo, para alocar 20% da banda para o protocolo A, 60% para o protocolo B e 20% para o protocolo C onde o tamanho dos pacotes do protocolo A é igual a 1086, do protocolo B igual a 291 e do protocolo C igual a 831, procede-se da seguinte forma: cada porcentagem de alocação é dividida pelo respectivo tamanho do pacote, isto é, 20 divido por 1086, 60 por 291 e 20 por 831. Logo depois, normalizam-se os resultados pelo menor valor, que no exemplo é igual a 0,01842 e teremos os seguintes resultados respectivamente 1, (0,20619 dividido por 0.01842) e (0,02407 dividido por 0.01842). Arredonda-se para o inteiro mais próximo superiormente, chegando aos resultados 1, 12 e 2; e multiplica-se pelos respectivos tamanhos dos pacotes, chegando-se ao número máximo de *bytes* que serão transmitidos pelo escalonador em cada visita a cada fila, isto é, na fila do protocolo A serão transmitidos 1086 *bytes*, na do protocolo B 3492 *bytes* e na do protocolo C 1662 *bytes*, em cada visita do escalonador.

2.5.7 **Escalonamento CBWFQ (*Class-Based Weighted-Fair Queuing*)**

O escalonamento CBWFQ (escalonamento WFQ baseado em classe) combina a característica de garantia de banda do escalonamento CQ com a característica de distribuição

justa entre diversos fluxos do escalonamento WFQ. (RODRIGUES, 2005).

A figura a seguir mostra um exemplo deste escalonamento. Observa-se a presença de três classes: a mais prioritária com 40% da banda reservada e com a garantia de entrega e uma menor latência, uma de prioridade média com 25% da banda reservada e com somente a garantia de entrega e uma terceira classe, a menos prioritária, com 10% da banda reservada e sem nenhum tipo de garantia e estará sujeita ao serviço de melhor-esforço.

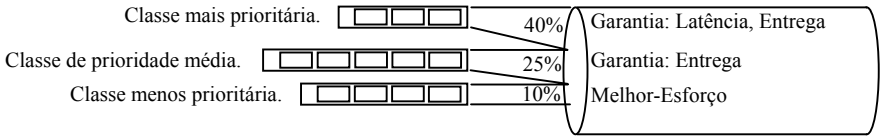


Figura 14 – Escalonamento CBWFQ. Adaptado de Rodrigues (2005)

Nota-se, na figura acima, que a soma das reservas das três classes atinge somente 75% da capacidade total do enlace. Vegesna (2001) afirma que é recomendável que a soma das bandas das filas dos escalonamentos PQ e CBWFQ não excedam esse valor, pois os 25% restantes ficam destinados aos tráfegos não classificados em nenhum dos dois escalonamentos (tráfego de controle, roteamento, entre outros) e aos encapsulamentos da camada dois.

Segundo Vegesna (2001), este escalonamento permite especificar uma banda mínima que cada classe irá receber, diferenciando-se do escalonamento WFQ onde a banda mínima do fluxo é obtida indiretamente baseando-se nos pesos de todos os fluxos ativos. Permite também que a capacidade reservada para um determinado fluxo seja utilizada por outros quando este não a estiver utilizando.

3 **METODOLOGIA DA PESQUISA**

3.1 TIPO DE PESQUISA

Para que uma pesquisa seja classificada como científica, é necessário escolher o paradigma de pesquisa que será a base do processo de investigação. (SILVA, 2003). Creswell (1994 apud SILVA, 2003), define os principais paradigmas de pesquisa existentes como sendo: qualitativo e quantitativo.

O paradigma de pesquisa utilizado foi o qualitativo, onde foram analisadas algumas informações conseguidas com o departamento de informática da instituição. E utilizou-se o método de estudo de caso, descrito por Creswell (1994) (apud SILVA, 2003, p. 21) como sendo um dos métodos do paradigma, por ser o método mais adequado à pergunta que o estudo coloca.

Yin (1994) define o estudo de caso como sendo uma investigação empírica que também necessita de revisão de literatura. Pode oferecer diversas variáveis porque baseia-se em várias fontes de evidências. Representa uma maneira de se investigar um fenômeno e suas intervenções na vida real, através de levantamento e experiências e procedimentos já testados. O estudo de casos, como estratégia de pesquisa, é preferencialmente usado nas ciências sociais e gerenciais, onde há uma clara necessidade de se compreender fenômenos sociais complexos. Não sendo um fim, mas um caminho de pesquisa.

O estudo de caso fora usado para estudar o funcionamento da instituição e determinar focos de mudança ou de intervenção. Esta pesquisa enfatizou o conhecimento da instituição para compreendê-la melhor no seu contexto e suas inter-relações internas e externas para desenvolver uma solução de tráfego de informações.

Vergara (1997 apud SILVA, 2003) classifica um trabalho científico conforme seu objetivo e a forma de investigação (os meios utilizados durante a pesquisa). Conforme os fins a serem atingidos, uma pesquisa pode ser: exploratória, descritiva, explicativa, metodológica, aplicada ou intervencionista.

Assim, esta pesquisa foi exploratória, analisou o uso dos mecanismos de QoS, e será também descritiva, visando expor as características do fenômeno.

Pesquisas bibliográficas e documentais, descritas por Vergara (1997 apud SILVA, 2003), foram alguns dos meios utilizados durante a pesquisa a fim de atingir os objetivos a que se propôs.

3.2 SELEÇÃO DOS SUJEITOS

Foram entrevistados um Analista de Rede e um Gerente da área de Tecnologia da Informação.

A escolha do Analista de Rede se deu pelo fato de ser a pessoa que detém o conhecimento da infra-estrutura de rede do hospital como um todo, bem como os equipamentos que a compõem e a interligação destes. E, por fim, a do Gerente, por ter uma visão gerencial do negócio e saber quais os projetos em que o hospital está envolvido e os futuros, o que proporcionou uma melhor adequação da pesquisa à instituição.

3.3 COLETA DE DADOS

Eco (2004) define dois tipos de fontes de dados, uma primária e a outra secundária, que o pesquisador deve utilizar para instrumentalizar a pesquisa. E recomenda a primária por obter os dados diretamente da fonte, não estando sujeito às interpretações de terceiros, diminuindo o risco de resultados tendenciosos ou equivocados.

Esta pesquisa utilizou fontes diretas, iniciando seu estudo buscando informações com um Analista de Rede, bem como, com um Gerente e alguns usuários através de entrevistas. As entrevistas do Analista de Rede e do Gerente serão informais ou abertas possibilitando conhecimento maior da instituição e seguiram os roteiros sugeridos pelos Apêndices A e B respectivamente.

Buscaram-se soluções para facilitar o tráfego de dados através de informações técnicas, observando as prioridades e as necessidades da instituição e seus principais serviços.

3.4 ANÁLISE E INTERPRETAÇÃO DOS DADOS

Os dados coletados através da entrevista do Analista de Rede foram analisados e utilizados para que fosse montado um cenário da instituição. Este cenário foi útil para que fossem identificadas as técnicas de QoS que melhor atendessem às necessidades da instituição e as que melhor se adaptassem aos equipamentos, serviços e aplicações do hospital.

A visão gerencial, formada a partir da análise da entrevista do Gerente, possibilitou que fossem identificadas as reais necessidades e prioridades do hospital, bem como a possibilidade de adequar o projeto às necessidades futuras deste.

3.5 LIMITAÇÕES DO MÉTODO

Segundo Yin (1994) o estudo de casos possui limitações, entre elas cita-se a possibilidade de manipulação dos dados pelo autor. O autor pode influenciar as respostas e conduzir a pesquisa de acordo com suas idéias, favorecendo este ou aquele resultado.

Os dados também podem ser manipulados, sem a articulação do autor, pelos entrevistados. Neste caso o autor não pode ser negligente, devendo checar as fontes buscando a maior autenticidade possível dos dados.

Outra limitação citada por Yin (1994) é que o estudo de casos é uma estratégia não generalizável, apresentando poucos dados para uma generalização científica, pois seu foco é apenas um objeto. Assim, os resultados obtidos nesta estratégia necessariamente não se aplicam a todas as situações, devendo ser adaptada ou, em algumas vezes, completamente reestruturada.

Uma outra preocupação, citada por Yin (1994), é a de que o estudo de casos fica limitado ao acesso às fontes de dados. Se todas as informações necessárias ao estudo não estiverem disponíveis ao pesquisador, este estudo pode não apresentar uma solução adequada ao problema da instituição.

Porém, sabendo-se das limitações e dificuldades do método escolhido, procurou-se utilizar todo o potencial qualitativo do mesmo, evitando-se a manipulação dos dados pelo autor e, diminuindo-se a manipulação pelos entrevistados, conseguindo assim, resultados mais científicos.

4 **A INVESTIGAÇÃO**

4.1 A INSTITUIÇÃO

O Instituto Nacional de Câncer (INCA) é o órgão do Ministério da Saúde, vinculado à Secretaria de Atenção à Saúde, responsável por desenvolver e coordenar ações integradas para a prevenção e controle do câncer no Brasil.

O INCA é composto de quatro unidades hospitalares e três unidades assistenciais, das quais duas foram objetos da presente investigação, a unidade hospitalar principal, o Hospital do Câncer I (HCI), localizado na Praça Cruz Vermelha, 23, Rio de Janeiro e a unidade assistencial CONPREV (Coordenação de Prevenção e Vigilância), localizada na Rua dos Inválidos, 212, Rio de Janeiro.

4.2 AMBIENTE DE REALIZAÇÃO DOS TESTES

O INCA, como dito anteriormente, não possui implementado tipo algum de mecanismo de priorização de tráfego.

Assim, foi montado um ambiente para realização dos testes, descrito na figura 20, onde foi escolhida a aplicação VoIP para realizar os testes por dois motivos: por ser uma aplicação de tempo-real intolerante, isto é, necessita de que requisitos mínimos sejam rigorosamente respeitados, pois atrasos e perdas de pacotes influem diretamente em seu desempenho; e, o outro motivo, foi que na entrevista com o Gerente de informática, foi verificado um interesse da instituição em relação a este serviço.

Na figura 15, observa-se o esquema de interligação das duas unidades em estudo. As duas antenas, que utilizam a tecnologia Wireless 802.11b, ligam os dois prédios a uma distância de aproximadamente quinhentos metros. Foi colocado em cada prédio um roteador Cisco com a intenção de empregar algumas técnicas de QoS para priorizar o tráfego de voz.

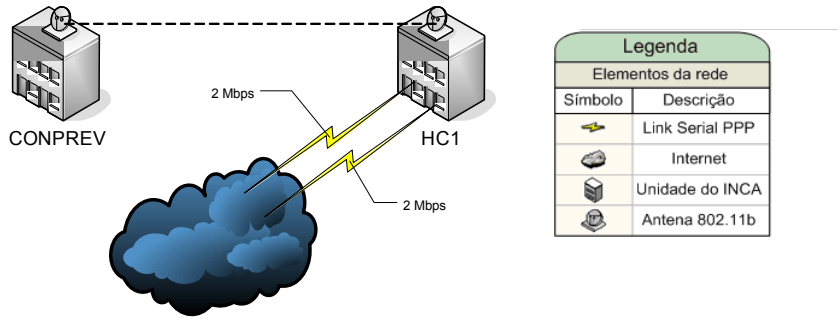


Figura 15 – O ambiente de teste.

Sendo esse *link* 802.11b o único meio de interligação das duas unidades, todo o tráfego de internet, e-mail, FTP e de alguns sistemas da unidade CONPREV, passam por ele, fazendo com que, em alguns momentos, se faça necessária a utilização de QoS para garantir o desempenho das aplicações de tempo-real.

A rede utilizada é formada por quatro nós, dois para gerar o tráfego de voz (chamador e chamado) e dois para gerar o tráfego de fundo[4] (gerador e refletor), que podem ser visualizados na figura 16.

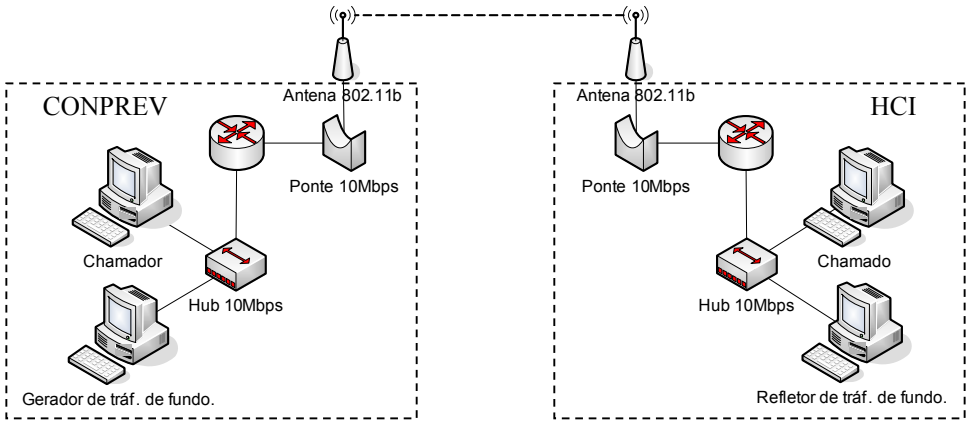


Figura 16 – O ambiente de teste em detalhe.

Utilizou-se uma ferramenta chamada MOBVEM (*Modified OpenH323 Based Voice Evaluation Module*) para geração do tráfego de voz, desenvolvida pelo Laboratório de VoIP da UFRJ (www.voip.nce.ufrj.br). Consiste de *scripts* escritos em *Perl* responsáveis pela

[4] Ou tráfego concorrente. Refere-se ao tráfego que disputará o meio de comunicação com o tráfego de voz.

geração de tráfego de voz baseado no protocolo OpenH323 e de um módulo genérico para avaliação objetiva da qualidade de voz.

Para geração do tráfego de fundo, utilizou-se uma ferramenta chamada Tangram, desenvolvida originalmente pelo Laboratório de VoIP da UFRJ para medição de capacidade de circuitos ATM (*Asynchronous Transfer Mode* – Modo de Transferência Assíncrona) e enlaces IP. Sendo hoje utilizada como geradora de tráfego de fundo, adicionou-se um mecanismo para geração de tráfego com espaçamento entre os pacotes exponencialmente distribuído com a finalidade de gerar um tráfego concorrente mais fiel à situação cotidiana.

Outra ferramenta utilizada foi a *Voice Quality Library*, também desenvolvida pelo Laboratório de VoIP, que junto com o *software* gratuito OpenPhone, mantido pelo projeto OpenH323 (www.openh323.org), transforma-se numa ferramenta chamada de VQOpenPhone, capaz de gerar chamadas de voz baseadas no protocolo OpenH323, permitindo ainda uma avaliação da qualidade das chamadas em tempo real.

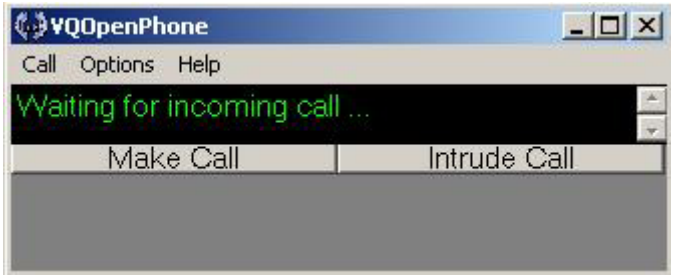


Figura 17 – O VQOpenPhone.

O VQOpenPhone tem o auxílio do VQPlot, também desenvolvido pelo Laboratório de VoIP da UFRJ, para gerar gráficos dos dados do Modelo E, que será explicado mais adiante, e dos dados estatísticos (tais como: *jitter*, atrasos da rede, etc.) colhidos no decorrer de toda a chamada.

Nas figuras 18 e 19 podem ser visualizados o VQPlot e os gráficos nele gerados.

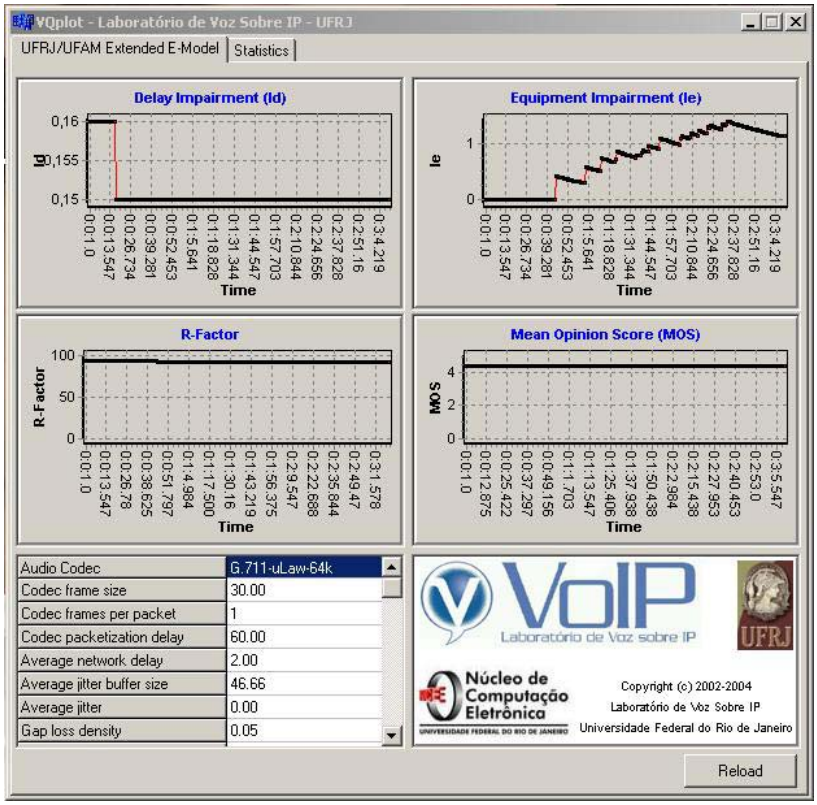


Figura 18 – Gráficos gerados pelo VQPlot com os dados do Modelo E.

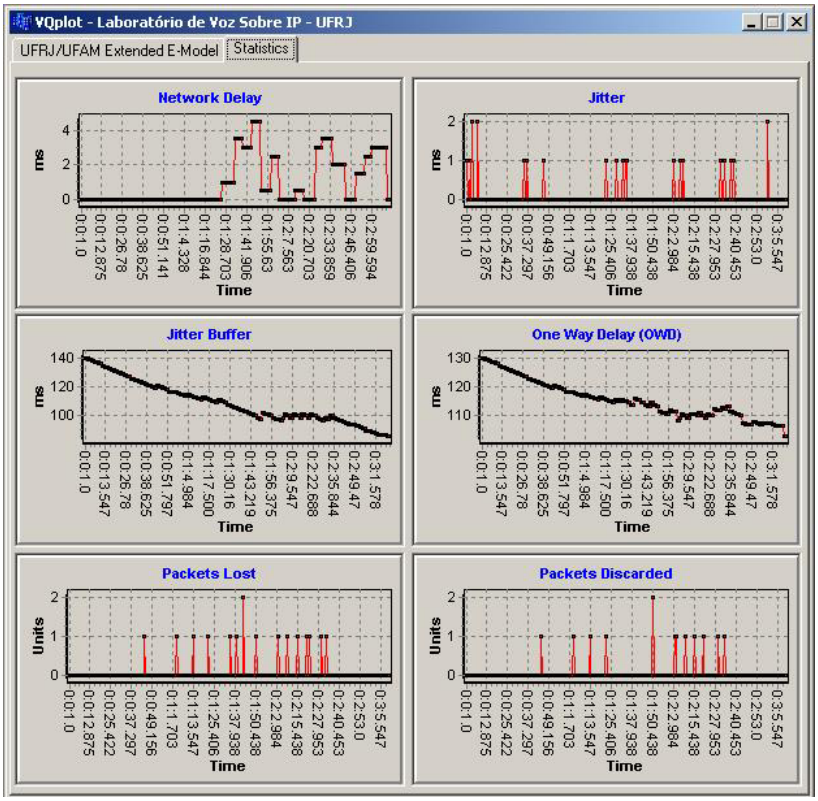


Figura 19 – Gráficos gerados pelo VQPlot com os dados estatísticos.

A avaliação da qualidade das chamadas de voz feita pelas ferramentas acima, baseia-se no *E-Model* (Modelo E), que é um modelo objetivo para avaliação da qualidade da voz em sistemas híbridos (rede de comutação de circuito/pacote). Esse modelo retorna um fator-R que varia de 0 a 100 que é convertido pelas ferramentas para a escala de pontuação MOS (*Mean Opinion Score* – média das opiniões) que é um método subjetivo de avaliação da qualidade, definido pelas recomendações P.800 e P.830 da ITU-T[5], onde uma média aritmética é calculada a partir das opiniões de um grupo de voluntários que avaliam trechos de conversas telefônicas e que varia de 1 (péssimo) a 5 (ótimo). A forma de conversão utilizada pelas ferramentas é apresentada na figura 20.

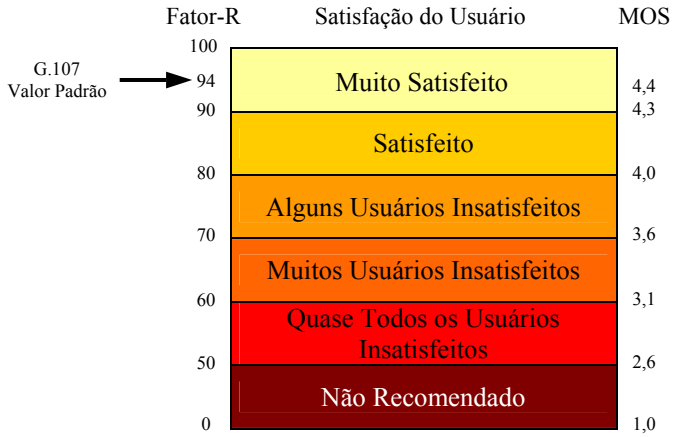


Figura 20 – Conversão do Fator R para a escala de pontuação MOS.

O tempo total de simulação de cada experimento foi de 160 segundos, tempo considerado suficiente para se observar os resultados desejados. Para cada experimento foram realizadas 30 replicações, que representam um compromisso aceitável entre o custo computacional e a confiança estatística nos resultados.

Para todos os resultados, calculou-se um intervalo de confiança de 90% da distribuição *t-Student*, que é representado através dos gráficos das figuras 20 e 21 da seção 4.4. Em alguns

[5] É um dos setores da ITU, *International Telecommunication Union*, responsável pela elaboração de padrões de alta qualidade (recomendações), envolvendo todos os campos de telecomunicações.

casos, no entanto, os intervalos não são visíveis, pois os resultados não apresentaram variações muito significativas (ou seja, os intervalos de confiança apresentaram tamanho reduzido).

As medições foram realizadas considerando-se ambos os sentidos do tráfego, da fonte para o destino. Os resultados estão apresentados na seção 4.4.

4.3 VALIDAÇÃO DO AMBIENTE DE TESTE

Para realização das chamadas de voz é necessária a escolha de um dos *codecs*[6] de voz, que, no caso dos testes realizados neste trabalho, foi escolhido o *codec* G.711, que é um dos *codecs* padronizados pelo ITU-T.

Todo *codec* demanda uma certa carga no processador da máquina que está realizando a chamada de voz. Em uma única chamada essa demanda por processamento não é acentuada, mas quando a máquina começa a realizar várias chamadas simultaneamente, como realizado nos testes deste trabalho, essa demanda começa a ser muito significativa.

Dois fatos foram considerados na escolha pelo *codec* G.711: o primeiro foi que o MOBVEM já estava configurado e funcionando com este *codec* e sua reconfiguração para trabalhar com outro demandaria um período de tempo do pessoal do laboratório VoIP, o outro foi que o G.711 é o *codec* que menos demanda processamento, pois não realiza compressão da voz, assim mais chamadas poderiam ser realizadas simultaneamente pelas máquinas sem perda de qualidade da voz por falta de capacidade de processamento e, como conseqüência da não compressão, menos chamadas simultâneas seriam necessárias para saturar o *link* pois é o *codec* que mais demanda capacidade de banda.

[6] Elemento responsável por codificar e decodificar a voz.

Com a finalidade de validar o ambiente de teste, as máquinas responsáveis pelo tráfego de voz foram colocadas em rede através de um cabo *cross-over* (cabo cruzado), possibilitando uma banda passante de 100Mbps *full-duplex*, conforme visto na figura a seguir, e assim, poderiam ser realizadas várias chamadas simultâneas sem que o limite fosse a banda passante e, conseqüentemente, fosse testada a capacidade das máquinas em realizar chamadas simultâneas.

Figura 21 – Interligação das máquinas geradoras do tráfego de voz.

Vale ressaltar que as máquinas responsáveis pelo tráfego de voz são máquinas com boa capacidade de processamento e memória, com a seguinte configuração: Processador Intel® Pentium 4 HT (*Hyper-Threading*) de 2.8 GHz, 512 MB de memória RAM DDR, disco rígido de 40 GB SATA 7200 rpm e placa mãe Intel i865G.

Como o MOBVEM roda somente em máquinas com sistema operacional Linux, foi instalado nas máquinas o Fedora Core versão 4 da distribuição Red Hat, pois foi nessa distribuição que foi construído o MOBVEM. Tomou-se o cuidado de instalar o mínimo possível nas máquinas e de utilizá-las com a única finalidade de gerar o tráfego de voz para que seus recursos não fossem compartilhados com outros serviços e/ou aplicativos.

O G.711, como dito anteriormente, é o *codec* que mais demanda capacidade de banda por não realizar compressão da voz. Ele gera uma amostra de voz de 8 *bits* a cada 125μs, ou seja, a cada 1ms são gerados 8 *bytes* em amostras de voz. A cada 30 ms ele gera uma PDU (*Protocol Data Unit* – Unidade de Dados do Protocolo) de voz, o que resulta em uma PDU com o tamanho de 240 *bytes* em amostras de voz como pode ser visto na figura a seguir.

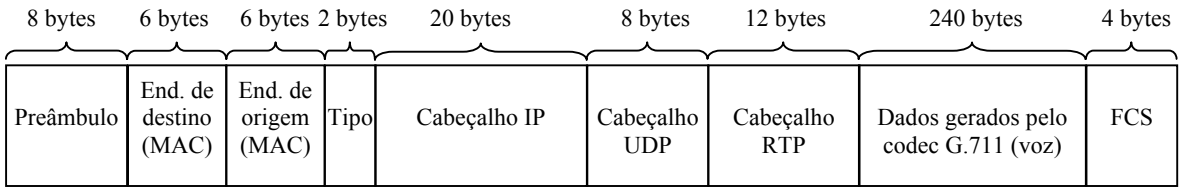


Figura 22 – Formato do quadro de voz no meio físico.

Uma PDU de 240 *bytes* gerada a cada 30ms demanda uma banda de 64 Kbps em cada chamada. Levando em consideração não só a PDU da camada de aplicação mas de todas as outras camadas, temos um quadro com tamanho de 306 *bytes* no meio físico, como pode ser visto na figura 22, o que resulta em um consumo de banda de 81,6 Kbps por chamada.

Nos testes foram conseguidas até 39 chamadas simultâneas sem perda de qualidade (com MOS acima de 4.2), o que soma 3182,4 Kbps de utilização de um *link* com a capacidade de 100 Mbps.

Nesse teste aplicou-se o mesmo critério de cálculo do intervalo de confiança de 90%, onde também realizaram-se 30 experimentos, e foi levada em consideração a utilização do processador da máquina próxima a 100%, como critério de parada do aumento do número de chamadas.

Assim, como poderá ser visto na seção que se segue, a capacidade de processamento das máquinas envolvidas na geração das chamadas de voz não influenciou os resultados obtidos na seção 4.4.

4.4 RESULTADOS OBTIDOS

Num primeiro momento, realizaram-se testes com o ambiente descrito na seção 4.2 sem o uso de QoS para se obter uma visão do comportamento das chamadas de voz em uma situação cotidiana.

Na figura 23 observa-se o MOS dos grupos de uma, cinco, dez, treze, quatorze, quinze e dezesseis chamadas respectivamente.

Chamadas com MOS abaixo de 3.1 são consideradas de qualidade ruim. Assim, verificou-se, na figura 20, que um grupo de até quinze chamadas possui uma qualidade aceitável, num modo tolerante de observar, e que até 13 chamadas poderiam ser realizadas com boa qualidade, ou seja, de uma forma bem compreensível.

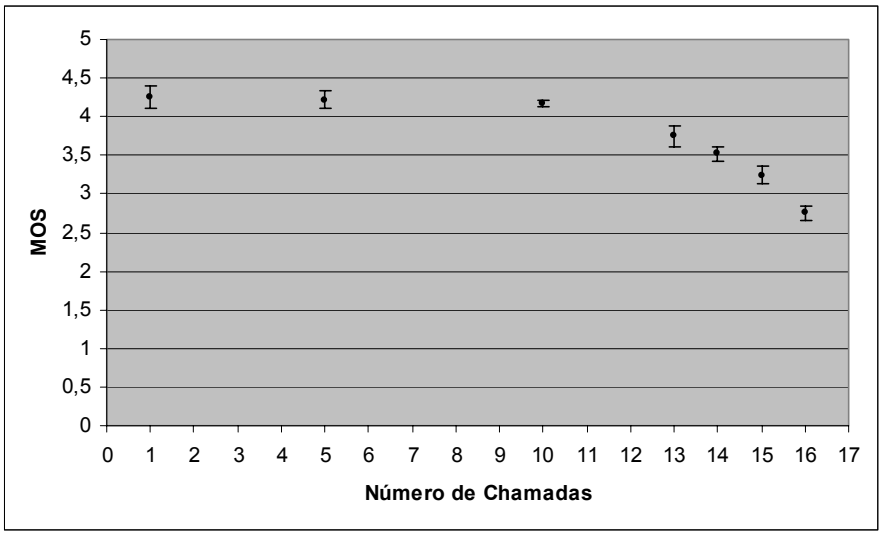


Figura 23 – Representação gráfica do MOS das chamadas.

Se estivesse sendo implantado um controle de admissão de chamadas nesse *link*, um grupo de até treze chamadas simultâneas poderia ser o limite a ser admitido para assegurar uma boa qualidade às chamadas.

Fazendo uma comparação com um sistema tradicional de telefonia, o *link* em questão (802.11b) poderia ser comparado com um tronco de treze linhas em que até treze ligações poderiam ser admitidas simultaneamente, e, levando em consideração uma relação de dez para um, até cento e trinta usuários poderiam ser bem atendidos por esse tronco.

Para obter uma melhor avaliação dos resultados obtidos, mediu-se o valor do tráfego de fundo em cada um dos grupos de chamadas, que fora coletado através do protocolo SNMP (*Simple Network Management Protocol* – Protocolo de Gerência Simples de Rede) configurado nas pontes 802.11b de cada prédio, e pode ser visualizado na figura 24.

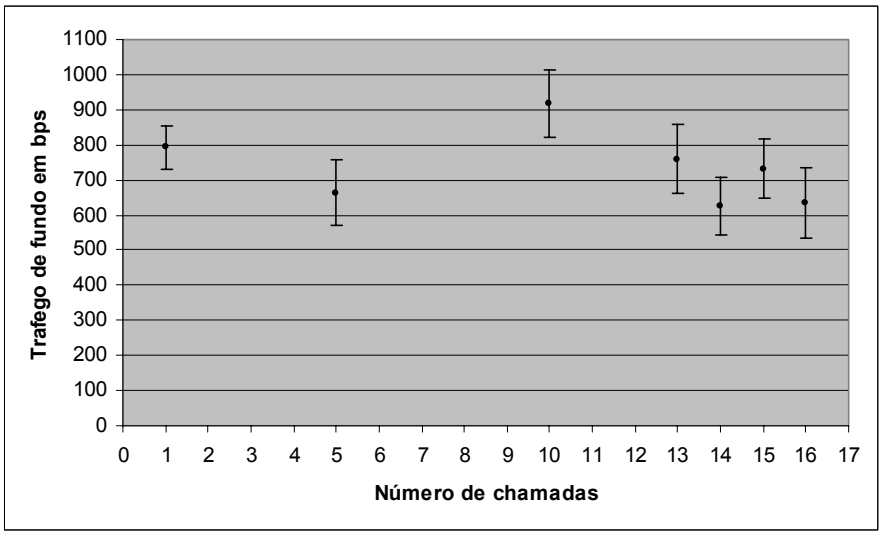


Figura 24 – Representação gráfica do tráfego de fundo.

As figuras 25 a 28, mostram uma representação gráfica do MOS de uma das chamadas dentro do conjunto de dez, quatorze, quinze e dezesseis chamadas simultâneas respectivamente no decorrer dos 160 segundos de duração de cada teste. Esses gráficos foram gerados a partir do VQPlot.

Para a realização destes testes foram utilizadas quatro máquinas: duas utilizando o MOBVEM (chamador e chamado) e duas utilizando o VQOpenPhone (chamador e chamado), pois o MOBVEM, como dito anteriormente, roda em Linux e o VQOpenPhone em Windows.

Os testes foram realizados como descrito a seguir: para o teste com dez chamadas (figura 25) por exemplo, foram disparadas nove chamadas pelo MOBVEM e uma chamada utilizando o VQOpenPhone disparada simultaneamente; para o teste com quatorze chamadas (figura 26), foram disparadas treze chamadas através do MOBVEM e uma com o VQOpenPhone, e assim sucessivamente.

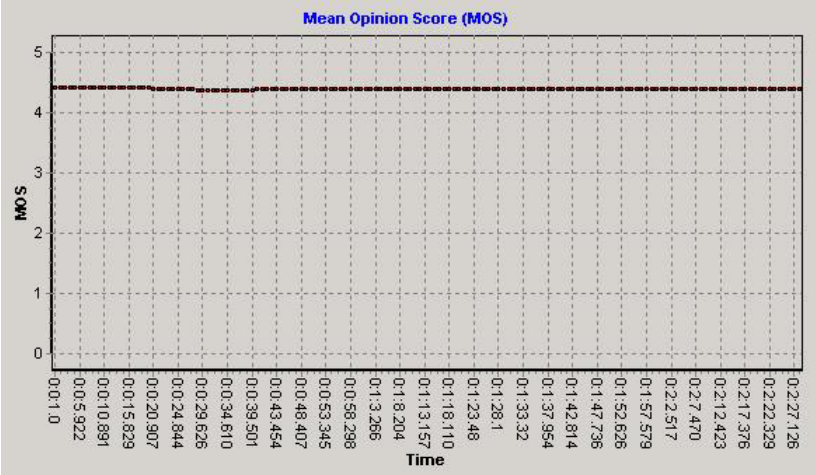


Figura 25 – MOS de uma das 10 chamadas no decorrer do tempo.

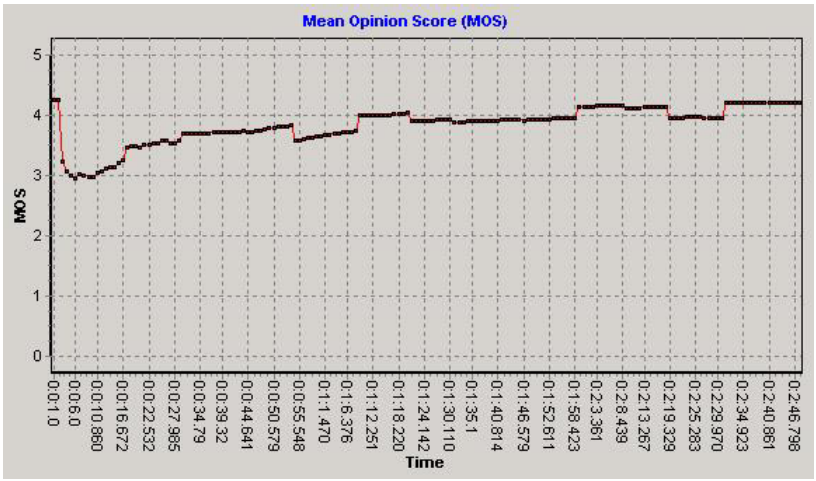


Figura 26 – MOS de uma das 14 chamadas no decorrer do tempo.

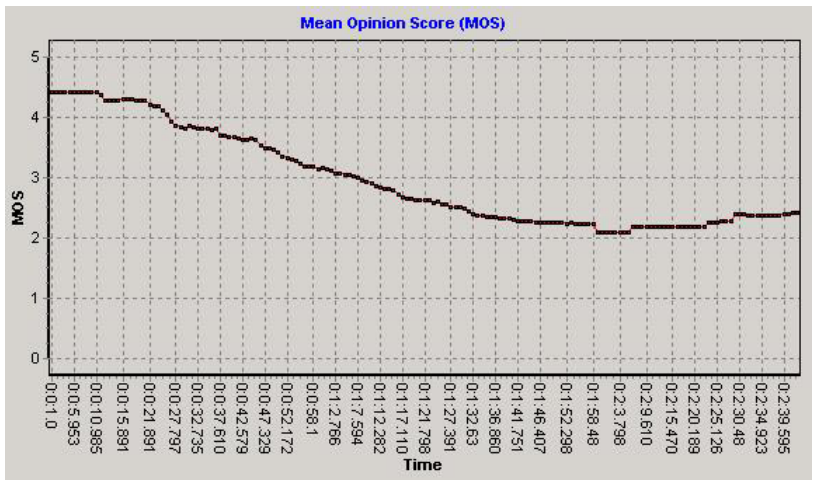


Figura 27 – MOS de uma das 15 chamadas no decorrer do tempo.

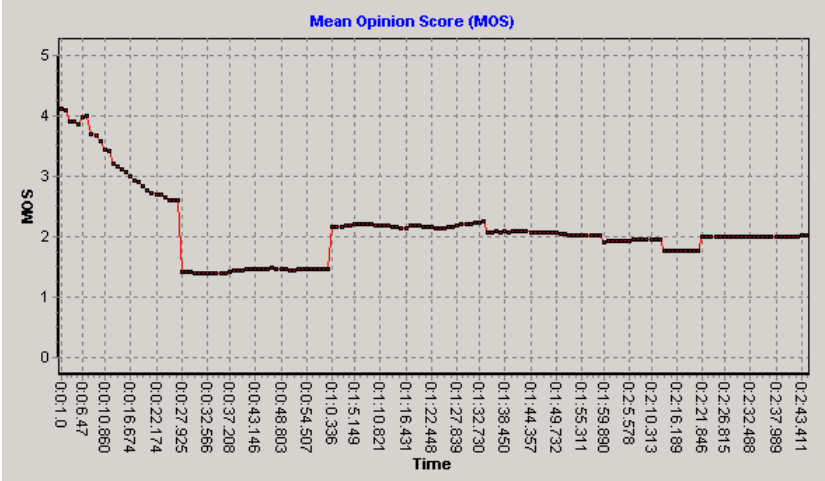


Figura 28 – MOS de uma das 16 chamadas no decorrer do tempo.

A partir de dezesseis chamadas simultâneas observa-se uma queda acentuada do MOS das chamadas fazendo com que sua compreensão fique bastante prejudicada, chegando em alguns momentos a ficar incompreensível, por falta de qualidade das chamadas.

O teste com 16 chamadas, por ter sido o teste em que o MOS das chamadas ficou mais prejudicado, foi detalhado através das figuras 29 a 35.

Na figura 29 é apresentado um gráfico do Fator-R do Modelo E, que se assemelha ao gráfico do MOS apresentado na figura 29, comprovando a eficiência da ferramenta em converter o Fator-R em escala de pontuação MOS.

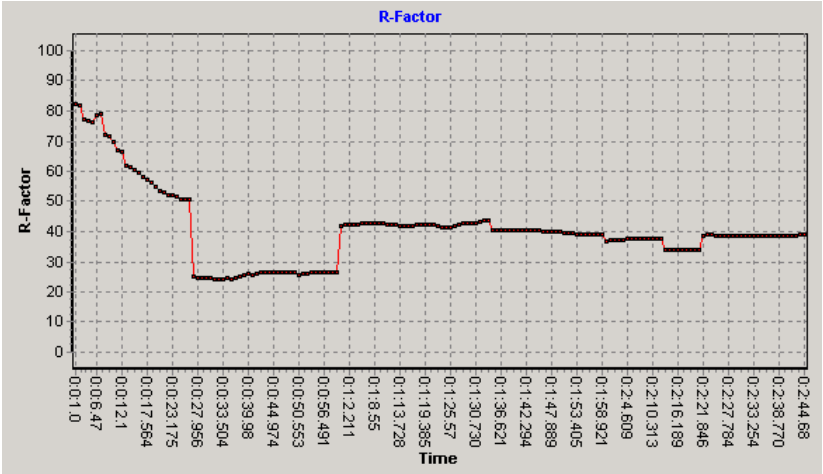


Figura 29 – Fator-R de uma das 16 chamadas no decorrer do tempo.

Antes de atingir o tempo de 27s, ocorre um declínio significativo no MOS e no Fator-R da chamada, como observado nas figuras 28 e 29, que pode se explicado pelo aumento expressivo do atraso da rede e do atraso fim-a-fim, como visto nas figuras 30 e 31 respectivamente; pois VoIP é uma aplicação muito sensível a atrasos como toda aplicação de tempo-real. Assim que os atrasos diminuem no tempo aproximado de 1m e 1s, tanto o MOS quanto o Fator-R aumentam, mostrando uma relação inversamente proporcional aos atrasos da rede e fim-a-fim.

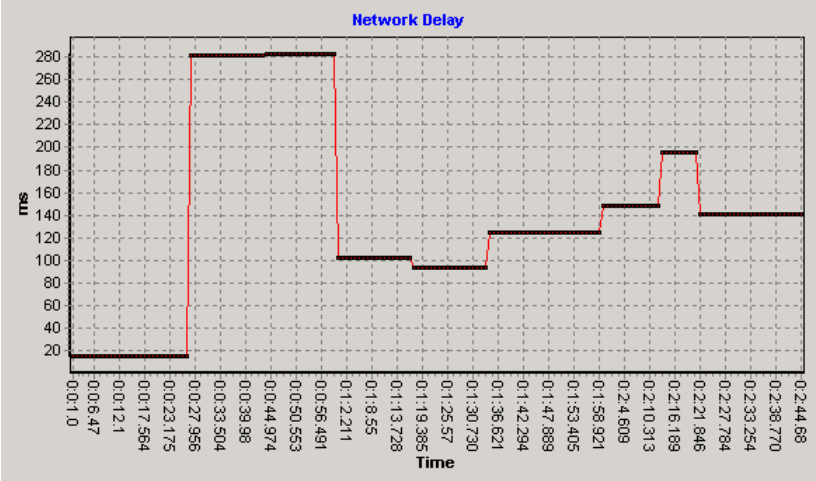


Figura 30 – Atraso da rede medido no decorrer do teste com 16 chamadas.

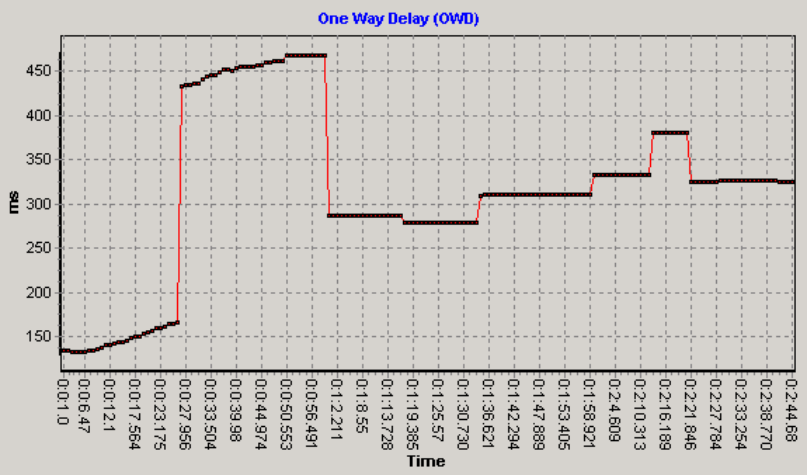


Figura 31 – Atraso em um sentido medido no decorrer do teste com 16 chamadas.

A variação do atraso (*jitter*), vista na figura 32, mostrou-se significativa para utilizar o *buffer* de compensação de *jitter* por completo (configurado com 250ms) a partir dos primeiros 50s da chamada, como apresentado na figura 33; o que, de uma maneira não tão significativa quanto os atrasos, prejudicou a qualidade da chamada.

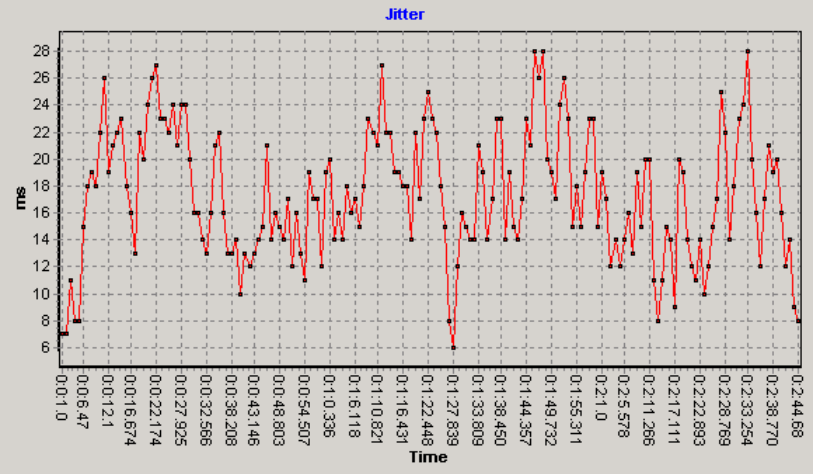


Figura 32 – Jitter medido no decorrer do teste com 16 chamadas.

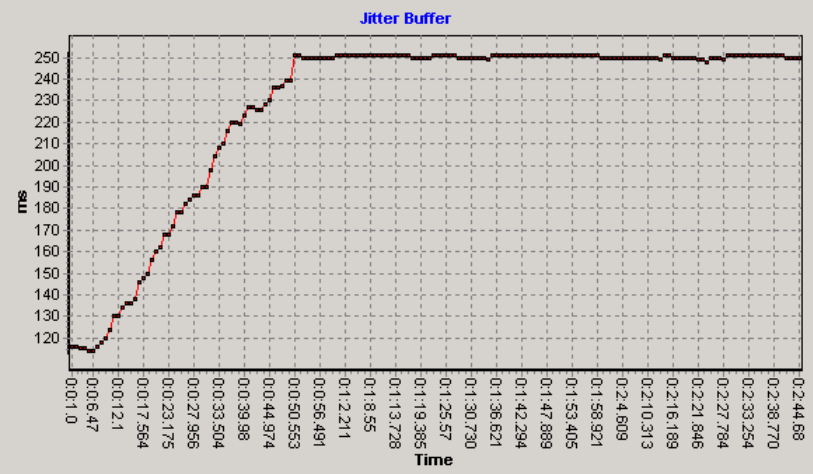


Figura 33 – Utilização do buffer de compensação de jitter da ferramenta.

A taxa de perda de pacotes e de descartados, vistas nas figuras 34 e 35, assim como o *jitter*, contribuíram para a diminuição da qualidade da chamada, porém não tanto quanto os atrasos.

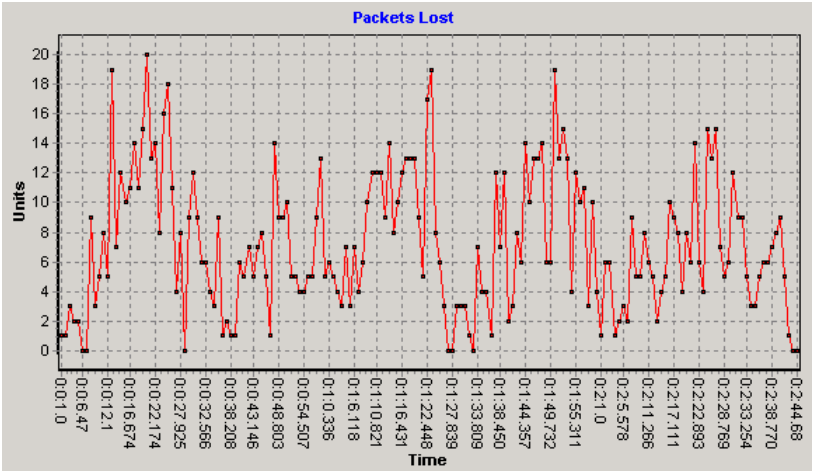


Figura 34 – Quantidade de pacotes perdidos no decorrer do teste com 16 chamadas.

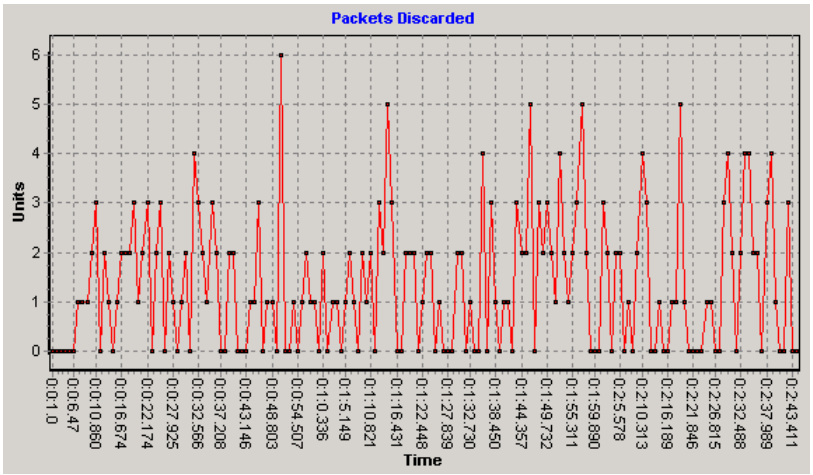


Figura 35 – Quantidade de pacotes descartados no decorrer do teste com 16 chamadas.

Logo depois foram realizados testes utilizando algumas técnicas de QoS, tais como: PQ, WFQ e CBWFQ. Porém os resultados obtidos com a realização desses últimos testes foram os mesmos obtidos nos testes sem QoS, e, como conseqüência, não foram conseguidas mais que quinze chamadas com qualidade aceitável (MOS acima de 3.1).

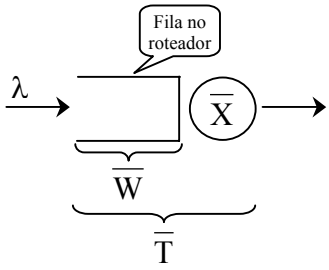
Com esses resultados obtidos nos testes com QoS, foram feitas análises para descobrir o que ocorrera para que as técnicas de QoS não obtivessem êxito na priorização do tráfego de voz.

Como as técnicas de QoS aqui estudadas são técnicas de priorização de filas no nível três ou nível de rede do modelo de referência OSI (*Open Systems Interconnection*), verificou-

se a existência de filas de nível três nos roteadores. Através dos cálculos realizados a seguir, observou-se que a formação de filas só seria significativa numa rede de 10Mbps com uma taxa de utilização do *link* acima de 95%.

Para realizar os cálculos foi utilizado o Modelo M|D|1, pois os pacotes que trafegam nesse *link* em geral são de tamanho grande (igual ou próximo a 1500 *bytes*) e fixo.

Modelo M|D|1 de formação de filas:



$$\overline{W} = \frac{\rho\overline{X}}{1-\rho} = \frac{\rho}{1-\rho} \times \frac{\overline{X}}{2}$$

Onde $\overline{X}$ = tempo médio para transmissão do pacote.

$\overline{X} = \frac{L}{10\text{Mbps}} = \frac{1500 \times 8}{10^7} = \frac{12000}{10^7} = 1{,}2\text{ms}$, onde L = tamanho do pacote em *bits*.

$$\overline{T} = \overline{W} + \overline{X}$$

Onde $\overline{T}$ = tempo total do pacote no roteador.

Para voz:

$$\overline{T_{voz}} = \overline{W_{M|D|1}} + \overline{X_{voz}}$$

$$= \frac{\rho}{1-\rho} \times \frac{L}{2C} + \frac{L_{voz}}{C}$$

Onde $\frac{L_{voz}}{C} = \frac{306 \times 8}{10\text{Mbps}} \cong 0{,}245\text{ms}$ e $L = 1500 \times 8$

$$= \frac{\rho}{1-\rho} \times 0{,}6 + 0{,}245$$

Para ρ = 0,5, ou seja, utilização do *link* a 50%, tem-se $\overline{T_{voz}} = 0{,}845\text{ms}$.

Para ρ = 0,95, tem-se $\overline{T_{voz}} = 11{,}645\text{ms}$.

Para ρ = 0,98, tem-se $\overline{T_{voz}} = 29{,}645\text{ms}$.

Número de pacotes na fila:

$$\overline{Nq} = \lambda \overline{W} = \lambda \times \frac{\rho}{1-\rho} \times \frac{\overline{X}}{2} = \frac{\rho^2}{1-\rho} \times \frac{1}{2}$$

Para $\rho = 0{,}95$, temos $\frac{0{,}95^2}{0{,}05} \times \frac{1}{2} = 18{,}05 \times \frac{1}{2} \cong 9$ pacotes.

Para $\rho = 0{,}97$, temos $\frac{0{,}97^2}{0{,}03} \times \frac{1}{2} = 31{,}3633 \times \frac{1}{2} \cong 16$ pacotes.

Para $\rho = 0{,}98$, temos $\frac{0{,}98^2}{0{,}02} \times \frac{1}{2} = 48{,}02 \times \frac{1}{2} \cong 24$ pacotes.

Para alcançar essa taxa acima de 95% de utilização do *link*, utilizaram-se as máquinas de geração de tráfego de fundo. Porém, com uma taxa de utilização tão alta, o número de pacotes logicamente aumentou muito, conseqüentemente os roteadores demandaram mais de seus processadores para tratar a quantidade de pacotes, pois além de realizar suas tarefas normais, como por exemplo: rotear pacotes, deveriam tratar a demanda por processamento que cada técnica de QoS exigia, fazendo com que os mesmos começassem a descartar os pacotes que não conseguiam tratar, e, conseqüentemente, degradar a qualidade das chamadas, pois são roteadores de pequeno porte e antigos, que não foram construídos para controlar e tratar tamanha quantidade de pacotes juntamente com as técnicas de QoS.

Outro fato importante é que uma taxa de utilização tão alta não é comum nesses *links* de velocidade mais alta. Desse modo, não vale a pena aplicar QoS nesses *links*.

Num ambiente de teste montado no Laboratório de VoIP da UFRJ, visualizado na figura 36, em que se colocaram dois roteadores Cisco serie 2500 ligados através de suas *interfaces* seriais (*interface* serial 0), formando um *link* de 1Mbps; também utilizaram-se quatro máquinas: duas para as chamadas de voz e duas para a geração do tráfego de fundo. Este ambiente teve o objetivo de testar três políticas de fila: FIFO, WFQ e CBWFQ.

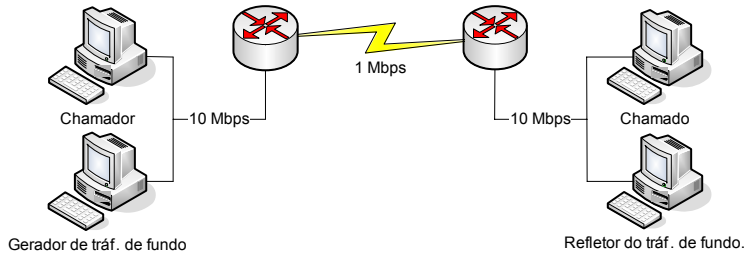


Figura 36 – Ambiente de teste montado no Laboratório de VoIP.

Vale dizer que neste ambiente de teste utilizou-se a mesma aplicação (VoIP) e as mesmas ferramentas do anterior (MOBVEM, Tangram). Diferentemente do ambiente da figura 16, este ambiente não foi montado em um ambiente de produção, e, conseqüentemente, houve a necessidade de criar um tráfego que concorresse com as chamadas de voz.

Testaram-se os seguintes cenários neste ambiente de teste:

Cenário 1 – Geraram-se 5 chamadas de voz concorrendo com um tráfego de fundo igual a 800Kbps, utilizando uma política FIFO.

Resultado: Gerou-se um tráfego de fundo de 800Kbps e também 5 chamadas de voz que equivalem a aproximadamente 400Kbps, o que somados resultam em aproximadamente 1200Kbps, ou seja, houve uma demanda maior que a capacidade do *link* que era de 1Mbps e, conseqüentemente, houve perdas tanto no tráfego de fundo quanto no tráfego das chamadas de voz, causando diminuição do MOS, ficando em aproximadamente 2,6, pois a política FIFO não prioriza nem discrimina tráfego qualquer.

Cenário 2 – Geraram-se 2 chamadas e um tráfego de fundo de 800Kbps, ainda utilizando a política FIFO.

Resultado: Como soma do tráfego de fundo com o tráfego das chamadas resulta em 960Kbps, uma demanda abaixo da capacidade do *link*, não houve perdas tanto no tráfego de fundo quanto no tráfego das chamadas, e as chamadas tiveram um MOS igual a 4,41.

Cenário 3 – Geraram-se 5 chamadas e um tráfego de fundo igual a 800Kbps, utilizando a política WFQ.

Resultado: Apesar da demanda da soma dos dois tráfegos ser maior que a capacidade do *link*, a qualidade das chamadas ficou ótima (MOS igual a 4,41), pois no sistema WFQ prioriza pacotes menores, como é o caso do pacote de voz (306 *bytes*), em detrimento do pacote maior, como é o caso dos pacotes do tráfego de fundo (1500 *bytes*). Neste experimento observou-se uma taxa de perda de 25% no tráfego de fundo, que representa os 200Kbps necessários para o tráfego de voz.

Cenário 4 – Tráfego de fundo de 800Kbps e 6 chamadas simultâneas, utilizando a política WFQ.

Resultado: A qualidade das chamadas foi boa (MOS 4,22), mas houve uma perda não significativa em uma das chamadas. A perda do tráfego de fundo foi de aproximadamente de 33% que representa 264Kbps que somado aos 200Kbps disponíveis resulta em 464Kbps, bem próximo do valor exigido pelas 6 chamadas (480Kbps).

Cenário 5 – Tráfego de fundo de 800Kbps e 6 chamadas simultâneas, ainda utilizando a política WFQ. Marcou-se o tráfego de mídia de voz com DSCP EF (*Expedited Forwarding* – Encaminhamento Expedido) e o tráfego de sinalização de voz com DSCP AF31 (*Assured Forwarding* – Encaminhamento Assegurado).

Resultado: O resultado foi o mesmo obtido que no último cenário, já que a política WFQ prioriza pacotes de tamanho menor. A marcação utilizada neste cenário somente fará diferença no caso de haver necessidade de priorizar o tráfego de voz em relação ao tráfego de outra aplicação, cujos pacotes gerados sejam de um tamanho bem próximo do tamanho dos pacotes do tráfego de voz.

Cenário 6 – Tráfego de fundo de 1000Kbps e 2 chamadas simultâneas, utilizando uma política CBWFQ com reserva de 256Kbps para o tráfego de voz (mídia e sinalização). Marcou-se o tráfego de mídia de voz com DSCP EF e o de sinalização com DSCP AF31.

Resultado: As duas chamadas de voz tiveram uma excelente qualidade, MOS 4,41. Houve 16% de perda no tráfego de fundo, o que equivale a 160Kbps. Como esta política reserva uma banda de até 256Kbps para o tráfego de voz, assegurou-se uma reserva de banda necessária para o tráfego das chamadas de voz.

Cenário 7 – Sem tráfego de fundo e 5 chamadas simultâneas, utilizando uma política CBWFQ com reserva de 256Kbps para o tráfego de voz (mídia e sinalização). O tráfego de mídia de voz marcou-se com DSCP EF e de sinalização com DSCP AF31.

Resultado: A qualidade das chamadas ficou ruim (MOS em média ficou igual a 2,8), pois a soma das demandas de banda de todas as chamadas é de aproximadamente 400Kbps e a reserva de banda para esse tráfego foi de 256Kbps; tudo que excedeu esse limite foi descartado, prejudicando a qualidade das chamadas.

Cenário 8 – Tráfego de fundo de 1000Kbps e 5 chamadas simultâneas, utilizando uma política CBWFQ com reserva de 256Kbps para o tráfego de voz (mídia e sinalização). Marcou-se o tráfego de mídia de voz com DSCP EF e o de sinalização com DSCP AF31. Implementou-se também um controle de taxa SRTCM com CIR configurado com 256Kbps, CBS configurado com 64000 *bytes* e EBS configurado também com 64000 *bytes*. O tráfego não conforme, tráfego marcado de “vermelho”, teve seu DSCP desmarcado, isto é, DSCP igual a “000000” (melhor-esforço).

Resultado: A qualidade das chamadas ficou muito boa, aproximadamente 4,38 na média. A reserva de banda configurada representou um pouco mais que a banda necessária para o tráfego de 3 chamadas. O tráfego das outras duas, que excedeu o limite, não foi descartado como aconteceu no cenário anterior, foi "jogado" para uma fila WFQ através do controle de taxa SRTCM para disputar com o tráfego de fundo (DSCP marcado com "000000"). Como a política WFQ prioriza os pacotes de tamanho menor, o tráfego das chamadas de voz ganhou prioridade, assegurando uma boa qualidade para as mesmas.

Para marcação dos pacotes de voz (mídia e sinalização) com DSCP EF e AF31 nos cenários de 5 a 8, foi utilizado o *IPTables* do Linux instalado nas máquinas responsáveis pelo tráfego de voz. Para geração das regras do *IPTables*, foi criado um usuário "voip" nas duas máquinas e as chamadas de voz foram geradas a partir desse usuário. Utilizaram-se as seguintes regras de *IPTables* para marcação dos pacotes de voz:

```
#iptables -t mangle -A POSTROUTING -m owner --uid-owner voip -p UDP -j DSCP
--set-dscp-class EF
#iptables -t mangle -A POSTROUTING -m owner --uid-owner voip -p TCP -j DSCP
--set-dscp-class AF31
```

As configurações das políticas de QoS nos roteadores realizadas na *interface* serial 0 de cada um, serão descritas a seguir:

– Configuração da política FIFO

```
#configure terminal
#interface serial 0
#no fair-queue
```

– Configuração da política WFQ

```
#configure terminal
#interface serial 0
#fair-queue
```

– Configuração da política CBWFQ

```
#configure terminal
#class-map match-any voice-flows
#description voip signaling and media
#match ip dscp ef
#match ip dscp af31
#exit
#policy-map voip
#class voice-flows
#priority 256
#exit
#exit
#interface serial 0
#no fair-queue
#service policy output voip
```

– Configuração do controle SRTCM

```
#configure terminal
#access-list 101 permit ip any any dscp ef
#access-list 101 permit ip any any dscp af31
#interface serial 0
#rate-limit output access-group 101 256000 64000 64000 conform-action
transmit exceed-action set-dscp-transmit 0
```

Neste ambiente de teste com um *link* de 1Mbps, todas as políticas de fila mostraram eficiência na priorização do tráfego de voz. No ambiente escolhido anteriormente, com um *link* de 10Mbps, as técnicas de QoS não obtiveram sucesso em priorizar o tráfego de voz, pois a formação de fila era praticamente inexistente pela alta velocidade do *link*.

5 CONSIDERAÇÕES FINAIS

5.1 DIFICULDADES

Muitas foram as dificuldades encontradas no decorrer da elaboração deste trabalho. Dentre as principais são citadas:

- O tempo necessário disponibilizar para construção do trabalho foi grande;
- A dificuldade de conseguir a disponibilidade necessária dos entrevistados;
- Tempo para montagem e preparação do ambiente para os testes realizados;
- Disponibilidade da Instituição para execução dos testes, pois alguns testes só puderam ser realizados nos finais de semana;
- Dificuldade de acesso à literatura;
- Resultados inesperados nos testes.

5.2 FACILIDADES

Para tornar a realização deste trabalho algo concreto, algumas facilidades foram de grande valia, sendo citadas a seguir:

- A grande ajuda do Professor Leandro e a do orientador Professor Aguiar;
- Disponibilização dos roteadores pelo NCE / UFRJ e pelo orientador, sem os quais os testes não poderiam ser realizados;
- Disponibilização das ferramentas utilizadas nos testes;
- A colaboração da instituição na realização dos testes, pois em alguns testes, foi preciso modificar o ambiente de produção para realizá-los.

5.3 TRABALHOS FUTUROS

Como trabalho futuro poderia ser explorada a implantação de QoS no ambiente proposto na figura 16, porém com roteadores com capacidade de processamento maior e que pudessem ser instalados no lugar das pontes 802.11b, ou seja, que tivessem uma interface 802.11b para que fossem ligados diretamente.

Outra proposta interessante seria que esses roteadores fossem robustos o suficiente para suportarem tanto a quantidade de pacotes necessária para realizar os testes, quanto para trabalharem com QoS no nível 3 do modelo OSI, como também no nível 2, através do padrão IEEE (*Institute of Electrical and Electronics Engineers* – Instituto de Engenheiros Elétricos e Eletrônicos) 802.11e, onde, por exemplo, poderia ser garantido um tempo maior de acesso ao meio físico de comunicação às aplicações mais prioritárias.

5.4 CONCLUSÕES

Em redes a partir de 10Mbps a formação de filas nos roteadores somente será significativa quando a utilização do enlace estiver acima de 95%, conseqüentemente, a discriminação entre um ambiente com QoS e outro sem QoS só será importante para valores muito altos de utilização, acima de 95%.

Nesta situação, os roteadores utilizados nos testes não suportaram a quantidade grande de pacotes por falta de capacidade de processamento e acabaram por descartar tanto os pacotes de dados quanto os pacotes de voz, fazendo com que, em alguns casos, a utilização de QoS piorasse a qualidade das chamadas de voz. Deste modo, não pode ser verificada a eficiência das técnicas em priorizar o tráfego de voz no link testado.

Porém, o impacto de QoS neste caso seria menos significativo que num ambiente com uma rede serial de baixa velocidade.

Entretanto isso é inviável na prática, de modo que nos enlaces de maior velocidade, não será preciso aplicar QoS nos roteadores, a menos em situações atípicas, pois, em uma situação cotidiana, esses enlaces não teriam uma utilização tão alta.

Como verificado nos cenários montados no Laboratório de VoIP, QoS é essencial em enlaces com velocidade igual ou abaixo de 2Mbps; para que em situações críticas de intensa utilização, os tráfegos mais importantes possam ser priorizados. Verificou-se também que, a partir de 50% de utilização média deste enlace de 1Mbps, houve momentos de picos de utilização máxima do enlace, intensificando ainda mais a necessidade de utilização de QoS nesses enlaces de baixa velocidade.

Algumas vezes, disputas internas no próprio computador podem implicar em piora na qualidade da voz, por falta de poder computacional em atender a demandas de programas CPU (*Central Processing Unit* – Unidade Central de Processamento) intensivos e altas transferências de arquivos, provocando perda no desempenho do SO (Sistema Operacional) no tratamento da voz.

Vale dizer aqui que este trabalho não tem um fim em si mesmo, ou seja, não pretende finalizar o assunto proposto. Espero que ele provoque novos questionamentos para que trabalhos futuros mais aprofundados possam ser realizados para o benefício da instituição e/ou de outras.

REFERÊNCIAS BIBLIOGRÁFICAS

BLAKE, Steven; BLACK, David L.; CARLSON, Mark A.; DAVIES, Elwyn; WANG, Zeng; WEISS, Walter. RFC 2475 – *An Architeture for Differentiated Services*. Dezembro de 1998.

BRADEN, Bob; CLARCK, David D.; CROWCROFT, Jon; DAVIE, Bruce; DEEREING, Steve; ESTRIN, Deborah; FLOYD, Sally; JACOBSON, Van; MINSHALL, Greg; PARTRIDGE, Craig; PETERSON, Larry; RAMAKRISHNAN, K. K.; SHENKER, Scott; WROCLAWSKI, John, ZHANG, Lixia. RFC 2309 – *Recomendations on Queue Management and Congestion Avoidance in the Internet*, Abril de 1998.

BRADEN, Bob; CLARCK, David; SHENKER, Scott. RFC 1633 – *Integrated Services in the Internet Architecture:* an Overview, Julho de 1994.

CHIOZZOTTO, Mauro; SILVA, Luís Antonio Pinto da. *TCP/IP Tecnologia e Implementação*. Érica, 1999.

CRESWELL, John W. *Research design:* qualitative & quantitative approaches. Thousand Oaks: Sage, 1994.

ECO, Umberto. *Como se faz uma tese*. São Paulo: Perspectiva, 2004.

FLOYD, Sally; JACOBSON, Van. *Random Early Detection Gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, V.1 N.4, Agosto 1993, p. 397-413.

HEINANEN, Juha; GUERIN, Roch. RFC 2697 – *A Single Rate Three Color Marker*. Setembro de 1999.

HEINANEN, Juha; GUERIN, Roch. RFC 2698 – *A Two Rate Three Color Marker*. Setembro de 1999.

MORRIS, Robert. *TCP Behavior with Many Flows*. IEEE International Conference on Network Protocols. Outubro de 1997.

NICHOLS, Kathleen; BLAKE, Steven; BAKER, Fred; BLACK, David L.. RFC 2474 – *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. Dezembro de 1998.

RAMAKRISHNAN, K. K.; FLOYD, Sally; BLACK, David L. RFC 3168 – *The Addition of Explicit Congestion Notification (ECN) to IP*. Setembro de 2001.

RODRIGUES, Paulo Henrique de Aguiar. *QoS em Redes*. Apostila da Disciplina de Qualidade de Serviços em Rede do Curso de Pós-Graduação em Gerência de Redes de Computadores e Tecnologia Internet do Programa MOT CN, 2005.

SILVA, Mônica Ferreira da. *Tecnologia da Informação***:** Um Estudo sobre a Influência do Contexto Social na XXX. Projeto de Pesquisa de Doutorado. Instituto COPPEAD de Administração, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2003.

STEVENS, W. Richard. RFC 2001 – *TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms*, Janeiro de 1997.

TANENBAUM, Andrew S. *Redes de Computadores***.** 4 ed. Rio de Janeiro: Campus, 2003.

VEGESNA, Srinivas. *IP Quality of Service* – The complete resource for understanding and deploying IP quality of service for Cisco networks. Cisco Press, 2001.

VERGARA, Sylvia Constant. *Projetos e relatórios de pesquisa em Administração*. São Paulo: Atlas, 1997.

YIN, R.K. *Applications of Case Study Research*. Newbury Park: Sage, 1994.

APÊNDICE A

Questionário 1

Entrevistado: _____
Função: _____
Data: _____

1) As unidades do INCA estão interligadas? Como?

_____
_____

2) Qual a velocidade dos *links* que ligam os hospitais?

_____

3) Qual a tecnologia envolvida em cada *link*?

_____
_____

4) Quais são as operadoras de cada *link*? Qual o tipo de contrato junto à elas?

_____
_____

5) Quais são os tipos de equipamentos e o mecanismo de QoS que cada um oferece?

_____
_____
_____

6) Quais são os tipos de roteadores ligados a cada *link*?

_____
_____
_____

7) Quais os serviços mais utilizados atualmente? E os que se pretende utilizar?

_____
_____

8) Quais os serviços têm mais importância para a instituição?

_____

APÊNDICE B

Questionário 2

Entrevistado: _____
Função: _____
Data: _____

1) Quantas unidades o INCA possui?

_____

2) Quantos usuários o INCA possui?

_____

3) Em quais projetos a instituição está envolvida? E quais são os futuros?

_____
_____

4) Quais serviços devem merecer prioridade máxima?

_____
_____

5) Quantos e quais são os setores que necessitam de prioridade em sua comunicação?

_____
_____

6) Quais são os problemas mais freqüentemente enfrentados?

_____
_____
_____

7) Como são solucionados?

_____
_____

8) Na sua opinião, o que está faltando para melhorar o serviço?

_____
_____
_____
_____