

MARCELO NASCIMENTO DOS SANTOS

**MEDIDAS DE QUALIDADE DE VOZ EM REDES IP**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná – UFPR.

Orientador: Prof. Dr. Eduardo Parente Ribeiro

Co-orientador: Prof. Dr. Marcus Vinicius Lamar

CURITIBA

2006

“Todos os livros do mundo não contém mais informação do que a que é transmitida em forma de vídeo para uma única grande cidade americana em apenas um ano. Nem todos os bits tem valor igual”.

Carl Sagan

## **AGRADECIMENTOS**

À minha família, amigos, professores e todos àqueles que contribuíram de alguma forma para todas as minhas conquistas e realizações.

## SUMÁRIO

<b>AGRADECIMENTOS .....</b>	<b>III</b>
<b>LISTA DE ACRÔNIMOS .....</b>	<b>VI</b>
<b>LISTA DE FIGURAS .....</b>	<b>VIII</b>
<b>RESUMO.....</b>	<b>IX</b>
<b>ABSTRACT.....</b>	<b>X</b>
<b>1 INTRODUÇÃO .....</b>	<b>11</b>
<b>2 CONCEITOS FUNDAMENTAIS.....</b>	<b>14</b>
2.1 TRANSPORTE DE VOZ SOBRE IP .....	14
2.1.1 FORMATO DO PACOTE VOIP .....	17
2.2 QUALIDADE DE SERVIÇO (QOS).....	18
2.2.1 PRINCIPAIS FATORES DE DEGRADAÇÃO .....	18
2.2.2 TRATAMENTO DE QOS .....	22
2.2.3 SERVIÇOS DIFERENCIADOS E INTEGRADOS .....	23
2.2.4 TÉCNICAS DE ENFILEIRAMENTO PARA QOS.....	25
<b>3 TRATAMENTO DA VOZ E MEDIÇÕES DE QUALIDADE.....</b>	<b>27</b>
3.1 CODIFICAÇÃO DOS SINAIS DE VOZ .....	27
3.2 SINAIS DE ÁUDIO .....	27
3.3 SINAIS DE VOZ .....	28
3.4 DIGITALIZAÇÃO DO SINAL DE VOZ.....	29
3.4.1 MODULAÇÃO POR CÓDIGO DE PULSO (PCM) .....	29
3.5 CODIFICADORES/DECODIFICADORES DE VOZ (CODECS).....	31
3.5.1 ITU G.711.....	32
3.5.2 PCM DIFERENCIAL ADAPTATIVO (ADPCM) .....	34
3.5.3 EXEMPLOS DE CODECS HÍBRIDOS.....	34
3.6 QUALIDADE DE VOZ.....	37
3.6.1 MEDIDAS DA QUALIDADE DE VOZ.....	37
3.6.2 MEDIÇÃO SUBJETIVA DA QUALIDADE .....	39
3.6.3 MEDIÇÃO OBJETIVA DA QUALIDADE .....	40
3.6.4 MODELOS DE OPINIÃO .....	41
3.6.5 MODELOS OBJETIVOS <i>SPEECH LAYER</i> .....	42
3.6.6 MODELOS OBJETIVOS <i>PACKET-LAYER</i> .....	47

<b>4 APRESENTAÇÃO DA METODOLOGIA.....</b>	<b>48</b>
4.1 O NETWORK SIMULATOR.....	49
4.2 A TOPOLOGIA DOS TESTES.....	50
4.3 O TRÁFEGO DE VOZ SIMULADO.....	51
4.4 O TRÁFEGO DE “BACKGROUND”.....	53
4.4.1 CARACTERÍSTICAS DO TRÁFEGO HTTP.....	53
4.4.2 GERAÇÃO DE TRÁFEGO WEB NO NS.....	55
4.5 ANÁLISE DO ARQUIVO DE DADOS.....	56
4.5.1 ANÁLISE DO ERRO.....	56
4.5.2 ANÁLISE DA VARIAÇÃO DO ATRASO.....	57
4.5.3 ANÁLISE ATRAVÉS DO E-MODEL.....	57
4.5.4 AVALIAÇÃO ATRAVÉS DA MEDIDA DE DISTORÇÃO.....	58
<b>5 RESULTADOS.....</b>	<b>65</b>
5.1 TESTE 1.....	65
5.2 TESTE 2.....	66
5.2.1 PERDA DE PACOTES.....	66
5.2.2 VARIAÇÃO DO ATRASO.....	68
5.2.3 MEDIÇÃO DA QUALIDADE.....	70
<b>6 CONCLUSÃO.....</b>	<b>75</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>77</b>
<b>ANEXO I.....</b>	<b>79</b>
<b>ANEXO II.....</b>	<b>82</b>

## LISTA DE ACRÔNIMOS

<b>Sigla</b>	<b>Definição</b>	<b>Tradução</b>
<b>ABS</b>	Analysis by syntesis	Análise por síntese
<b>ACELP</b>	Algebraic code excited linear prediction	Predição linear por excitação com código Algébrico
<b>ACR</b>	Absolute category rate	Classificação por categoria. absoluta
<b>ADPCM</b>	Adaptive differential pulse code modulation	Modulação adaptativa por código de pulsos diferenciais
<b>AF PHB</b>	Assured forwarding per hop behavior	Comportamento por nó de Encaminhamento Assegurado
<b>ATM</b>	Asynchronous transfer mode	Modo de transmissão assíncrono
<b>CBQ</b>	Class-based queuing	Enfileiramento baseado em classes
<b>CBR</b>	Constant bitrate	Taxa de bits constante
<b>CCR</b>	Comparison category rating	Classificação por categoria de comparação
<b>CDMA</b>	Code division multiplex access	Acesso múltiplo por divisão de código
<b>CLP</b>	Cell loss priority	Prioridade no descarte de célula
<b>DCR</b>	Degradation category rating	Classificação por categoria de degradação
<b>DiffServ</b>	Differentiated services	Serviços diferenciados
<b>DMOS</b>	Degradation mean opinion score	Nível de opinião médio por degradação
<b>DSCP</b>	Differentiated services code point	Ponto de código de serviços diferenciados
<b>EF PHB</b>	Expedited forwarding per hop behavior	Comportamento por nó de encaminhamento expresso
<b>FCFS</b>	First come first served	Primeiro a chegar primeiro a ser servido
<b>FFT</b>	Fast Fourier transform	Transformada rápida de Fourier
<b>FIFO</b>	First in first out	Primeiro a entrar primeiro a sair
<b>GSM</b>	Global mystem for mobile communications	Sistema global para comunicações móveis
<b>HTTP</b>	Hypertext transfer protocol	Protocolo de transferência de hipertexto
<b>IETF</b>	Internet engineering task force	Força tarefa de engenharia de internet
<b>IntServ</b>	Integrated services	Serviços integrados
<b>IP</b>	Internet protocol	Protocolo de internet
<b>IPv4</b>	Internet protocol version 4	Protocolo de internet versão 4
<b>IPv6</b>	Internet Protocol version 6	Protocolo de internet versão 4
<b>ISDN</b>	Integrated services digital network	Rede digital de serviços integrados
<b>ITU</b>	International telecommunications union	União internacional de telecomunicações
<b>MNB</b>	Measuring normalized blocks	Blocos de medida normalizados
<b>MOS</b>	Mean opinion score	Nível de opinião médio
<b>NS</b>	Network simulator	Simulador de redes
<b>NTT</b>	Nippon telegraph and telephone	Companhia telefônica e telegráfica do Japão
<b>OSI</b>	Open systems interconnect	Interconexão de sistemas abertos
<b>PAMS</b>	Perceptual analysis measurement system	Sistema de medida por análise perceptual
<b>PCM</b>	Pulse code modulation	Modulação por código de pulso
<b>PHB</b>	Per-hop behavior	Comportamento por nó
<b>PSQM</b>	Perceptual speech quality measurement	Medição da qualidade perceptual da fala
<b>QoS</b>	Quality of service	Qualidade de serviço
<b>RED</b>	Random early detection	Detecção randômica antecipada
<b>RFC</b>	Request for comments	Requisição para comentários
<b>RSVP</b>	Resource reservation protocol	Protocolo de reserva de recursos
<b>RTCP</b>	Real-time transport control protocol	Protocolo de controle de transporte em tempo real
<b>RTP</b>	Real-time transport protocol	Protocolo de transporte em tempo real

<b>SIP</b>	Session Initiation protocol	Protocolo de iniciação de sessão
<b>SURGE</b>	Scalable URL source generator	Gerador escalável de fontes URL
<b>TCP</b>	Transport control protocol	Protocolo de controle de transporte
<b>TOS</b>	Type of service	Tipo de serviço
<b>UDP</b>	User datagram protocol	Protocolo de datagrama do usuário
<b>ULP</b>	Unconditional loss probability	Probabilidade de descarte incondicional
<b>VAD</b>	Voice activity detector	Detector de atividade de voz

## LISTA DE FIGURAS

FIGURA 1 – EXEMPLO DE UMA REDE IP.....	14
FIGURA 2: REPRESENTAÇÃO DO MODELO OSI .....	15
FIGURA 3 – FORMATO DO PACOTE VOIP .....	17
FIGURA 4 – FUNCIONAMENTO DO <i>JITTER BUFFER</i> .....	20
FIGURA 5 – PARÂMETROS DO MODELO GILBERT.....	22
FIGURA 6 – CAMPOS IP TOS E DS .....	25
FIGURA 7 – PROCESSO DE AMOSTRAGEM .....	30
FIGURA 8 – PROCESSOS DE QUANTIZAÇÃO E CODIFICAÇÃO .....	31
FIGURA 9 – CODIFICAÇÃO G.711 .....	33
FIGURA 10 – COMPARAÇÃO ENTRE O FATOR R E ESCALA MOS .....	42
FIGURA 11 – DIAGRAMA DO PSQM DE ACORDO COM A RECOMENDAÇÃO P.861 .....	44
FIGURA 12 – DIAGRAMA DA METODOLOGIA.....	49
FIGURA 13 – TOPOLOGIA DA REDE SIMULADA .....	51
FIGURA 14 – FORMATO DO PACOTE A SER SIMULADO .....	52
FIGURA 15 – TRÁFEGO DE VOZ SIMULADO .....	53
FIGURA 16 – PARÂMETROS DO TRÁFEGO “BACKGROUND” .....	54
FIGURA 17 –CÁLCULO DA PERDA DE PACOTES .....	57
FIGURA 18 – OBTENÇÃO DO VETOR DE ERRO E MEDIÇÃO.....	58
FIGURA 19 – OBTENÇÃO DOS SINAIS A SEREM MEDIDOS.....	59
FIGURA 20 - CORRESPONDÊNCIA ENTRE AS ESCALAS DE FREQUÊNCIA EM HERTZ E BARK60	
FIGURA 21 – VALORES DAS DIFERENÇAS PARA O CODEC G.711 .....	62
FIGURA 22 – CURVAS DE MOS E IE PARA O CODEC G.711 .....	63
FIGURA 23 - VALORES DAS DIFERENÇAS PARA O CODEC G.729.....	64
FIGURA 24 – COMPARAÇÃO ENTRE O E-MODEL E O MÉTODO .....	65
FIGURA 25 – DENSIDADE DE PROBABILIDADE DA PERDA DE PACOTES.....	67
FIGURA 26 – DENSIDADE DE PROBABILIDADE DA VARIAÇÃO DO ATRASO .....	69
FIGURA 27 – MOS PARA 150 USUÁRIOS.....	71
FIGURA 28 – MOS PARA 200 USUÁRIOS.....	72
FIGURA 29 – MOS PARA 250 USUÁRIOS.....	73
FIGURA 30 – MOS E PERDAS MÉDIOS X E-MODEL.....	74

## RESUMO

Este trabalho apresenta um estudo da qualidade do tráfego de voz em redes IP. Aqui são apresentadas as características do transporte de voz sobre IP, os diversos fatores que afetam a qualidade do sinal de voz e as técnicas de medição utilizadas nessas redes. O estudo apresenta uma metodologia para medição da qualidade de voz utilizando uma rede simulada, construída com o Network Simulator, e fazendo uso dos arquivos de dados obtidos na simulação para realizar as medidas de qualidade. A utilização da rede simulada permite o teste de uma grande variedade de configurações de rede, sem a necessidade de mudança de hardware ou software. O método proposto pode ser utilizado quando não se dispõe de uma rede VoIP disponível para testes e oferece grande simplicidade pois utiliza softwares de fácil utilização e disponíveis gratuitamente.

Palavras-chave: Qualidade de Serviço, VoIP, Medições em redes

## **ABSTRACT**

This work shows a study of the quality of voice traffic over IP networks. Here we present the voice over IP characteristics, the various impairments over the voice signal and the assessment technics used on these networks. The study introduces a methodology to measure the voice quality using a simulated network with the trace files from the simulation in order to perform quality measurements. The use of a simulated network allows evaluating different configurations, without the need of changing hardware or software. The proposed method can be used when there are no real VoIP networks available for testing and offers great simplicity as it uses free and easy to handle softwares.

*Keywords: Quality of Service, VoIP, Network measurements*

## 1 INTRODUÇÃO

Com o crescimento da Internet, o *Internet Protocol* (IP) [1] passou a ter uma grande importância nas redes de transmissão de dados. Desta forma foi natural a utilização deste protocolo não apenas para o transporte de dados, como também para o transporte de voz entre outros serviços. Entretanto, por sua natureza, o protocolo IP não fornece meios eficientes para a transmissão de serviços de voz com qualidade, de forma que não poderia substituir uma rede de telefonia tradicional sem que sejam criadas soluções para tal.

Voz sobre IP (VoIP) pode ser definido como o transporte do tráfego de voz utilizando o protocolo IP. É importante observar, entretanto, que esta definição não é só aplicada para voz sobre Internet. A Internet é um conjunto de diversas redes interconectadas, que utiliza o protocolo IP, e que pode ser acessada por qualquer pessoa, e utilizada por uma grande variedade de aplicações [2]. O conceito de VoIP é aplicado para o transporte de voz em qualquer rede, pública ou privada, onde é utilizado o protocolo IP. A utilização de VoIP é uma oportunidade de se utilizar uma rede já existente, para oferecer um novo serviço. Entre as vantagens de uma rede integrada, podemos citar o baixo custo dos equipamentos, menores gastos com operação, oportunidade de criação de novos serviços.

O IP por si só não oferece garantias. É possível que diferentes pacotes que possuem mesma origem e destino tomem rotas distintas dentro da rede, o que pode ocasionar o recebimento de pacotes fora da ordem original, ou mesmo o não recebimento de alguns deles. Para evitar estes problemas, outros protocolos foram desenvolvidos para operar em conjunto com o IP, de forma a assegurar que os pacotes sejam entregues ao destino na sequência correta e sem perdas. Um protocolo que merece atenção especial é o TCP (*Transmission Control Protocol*) [3]. O TCP inclui funções para a retransmissão de pacotes perdidos ou que sofreram atrasos muito grandes e para o ordenamento de pacotes..

Apesar dos mecanismos utilizados pelo TCP serem apropriados e terem bastante sucesso na transmissão de dados, eles não são apropriados para a transmissão do tráfego de voz. A maioria do tráfego de dados é composta de dados

assíncronos e bastante sensível a erros, por outro lado, o tráfego de voz é síncrono por natureza e um pouco mais tolerante a erros.

Um dos grandes problemas na utilização de redes IP para o tráfego de voz é que normalmente essas redes são compartilhadas por vários outros serviços e a banda total disponível é utilizada por várias aplicações. Se não houver um controle da forma com que cada aplicação fará uso desta banda o tráfego de voz poderá ser seriamente afetado, tornando-se impossível uma garantia de qualidade. Mesmo com o grande aumento da capacidade de transporte das redes atuais, o aumento da utilização das redes também cresce rapidamente, desta forma, uma rede projetada para fornecer capacidade suficiente em um determinado momento, não garante que no futuro, poderá oferecer a mesma garantia. Também deve ser observado que no caso de falha na rede, essa garantia de banda não é mais válida.

Desta forma, é necessário conhecer mecanismos que possam fornecer ao tráfego de voz uma qualidade suficiente, e também os meios para que esta qualidade possa ser mensurada, pois as formas utilizadas nas redes tradicionais de telefonia, não podem ser diretamente aplicadas nas redes IP.

A avaliação da qualidade do tráfego de voz em uma rede IP pode ser realizada de diversas formas, uma medição subjetiva utiliza a avaliação feita por ouvintes, mas por se tratar de um método trabalhoso, na maioria das vezes é feita a avaliação objetiva da qualidade. Vários algoritmos vem sendo desenvolvidos para realizar medições objetivas de qualidade; o método apresentado neste trabalho utiliza uma medida de distorção do sinal de voz no domínio da frequência. Existem vários algoritmos que utilizam este princípio, alguns deles são descritos em recomendações do ITU-T. Como a maioria das ferramentas que utilizam estes algoritmos são produtos comerciais e são apenas utilizados para medição em redes reais, o método utilizado neste trabalho utiliza de forma simplificada, alguns conceitos encontrados nesses algoritmos como forma de medir a qualidade na rede simulada. Apesar de não utilizar a avaliação no domínio perceptual, como é feito nos algoritmos que tem maior correlação com os métodos objetivos, o método proposto pode ser utilizado para avaliar o impacto das diversas configurações de rede e dos codecs utilizados na qualidade da voz.

No capítulo 2 é feita uma apresentação da pilha de protocolos IP, suas características básicas e como o mesmo é utilizado para o transporte de voz. Também é feita apresentação do QoS, dos fatores que afetam a qualidade da voz e as técnicas para prover QoS em redes IP. No capítulo 3, são apresentadas as técnicas de codificação de voz, e alguns dos codecs utilizados no transporte de sinais de voz. Também são descritos os modelos utilizados para medições da qualidade de voz. O capítulo 4 faz uma apresentação da metodologia utilizada no trabalho, as ferramentas utilizadas e os procedimentos de medição. Os resultados das simulações são mostrados no capítulo 5, onde também são feitas as análises desses resultados. Por último, o capítulo 6 mostra as considerações finais e a conclusão.

## 2 CONCEITOS FUNDAMENTAIS

### 2.1 TRANSPORTE DE VOZ SOBRE IP

O protocolo IP foi projetado para permitir a interconexão de redes de computadores que utilizam a tecnologia de comutação de pacotes. Nele a transmissão dos dados é feita através de blocos de informação chamados datagramas, onde a origem e o destino são identificados por endereços de tamanho fixo [1].

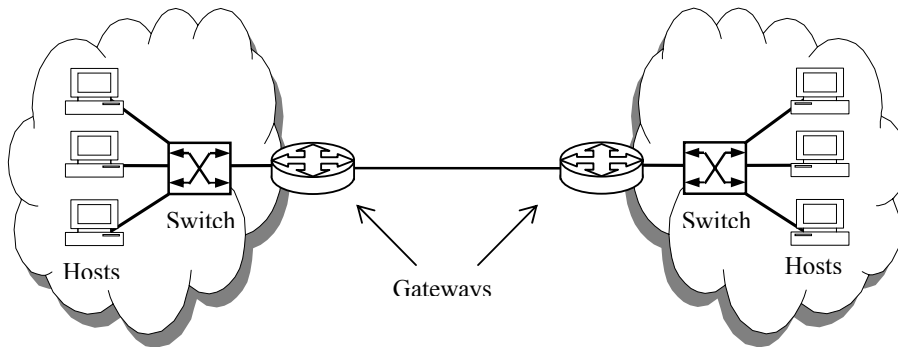


Figura 1 – Exemplo de uma rede IP

O IP trabalha em conjunto com protocolos de camadas superiores de forma a permitir a operação de aplicações ou serviços.

Entende-se por camada um componente do modelo OSI, este modelo está representado na Figura 2. Neste modelo o protocolo IP está localizado na camada 3, denominada camada de rede.

<b>Camada 7 - Aplicação</b>
<b>Camada 6 - Apresentação</b>
<b>Camada 5 - Sessão</b>
<b>Camada 4 - Transporte</b>
<b>Camada 3 - Rede</b>
<b>Camada 2 - Enlace</b>
<b>Camada 1 - Física</b>

Figura 2 – Representação do modelo OSI

Cada lado participante de uma comunicação envolve um certo número de camadas. A camada mais superior representa a informação que deve ser passada ao destino final. Entretanto, para que esta informação seja transferida, ela deve ser empacotada apropriadamente, e os pacotes por sua vez, devem ser roteados através de um meio físico.

O *Transmission Control Protocol* (TCP) [3] é um protocolo orientado à conexão que fornece um serviço confiável de transferência de dados fim a fim [3].

O TCP está localizado na camada de transporte, acima do IP. Ele tem como função básica, assegurar que todos os pacotes serão entregues às aplicações de destino em sequência, sem perdas e sem erros. Como o IP é um protocolo do tipo *best-effort*, ou seja, não garante que todos os pacotes serão entregues ao destino, o TCP tem a função de compensar essa falta de confiabilidade do IP através de uma confirmação fim a fim do recebimento de cada pacote. No caso em que pacotes não sejam recebidos no destino, o TCP garante uma retransmissão. O TCP também inclui um controle de fluxo, de forma que uma aplicação na origem, não sobrecarregue uma aplicação mais lenta no destino, com o envio de uma quantidade maior de pacotes do que ela pode tratar, além de um controle de congestionamento.

Também situado na camada de transporte, o *User Datagram Protocol* (UDP) [4], oferece um procedimento para o envio de mensagens para outros programas com um mínimo de mecanismos envolvidos, por exemplo o protocolo não garante a

entrega dos dados nem protege contra entregas duplicadas [3]. O UDP realiza uma função bastante simples, do lado transmissor ele simplesmente envia segmentos de dados de uma aplicação para o IP fazer o roteamento até o destino, do lado receptor, ele simplesmente repassa os segmentos do IP para a aplicação [5].

Diferente do TCP, o UDP não possui a funcionalidade de confirmação de mensagens, por isso não existe a garantia de entrega. No caso de perda de pacotes UDP, a aplicação é responsável pelo procedimento de identificação e retransmissão.

Quando desejamos transportar voz sobre IP, utilizamos o UDP como protocolo de transporte. Apesar do UDP ser um protocolo não confiável, em uma conversação, a perda ocasional de 1 ou 2 pacotes de voz embora não seja desejável, não prejudica de forma grave a transmissão. Por outro lado, o tráfego de voz é altamente sensível ao atraso, e a rotina de estabelecimento e confirmação de mensagens do TCP introduz atrasos. No caso de perda de pacotes, o TCP causaria ainda mais retardo na transmissão, devido à retransmissão. Desta forma, tolerar alguma perda de pacotes é uma estratégia melhor do que introduzir atraso na transmissão de voz [2].

O protocolo UDP, entretanto, não foi originalmente criado para o transporte de voz, ele é apenas a melhor escolha, em lugar do TCP. Para que o UDP seja utilizado no tráfego de voz, é necessário fazer uso de outros protocolos para que o transporte de voz seja feito de uma forma mais eficiente.

Para o transporte dos sinais de voz e vídeo sobre redes IP, foi desenvolvido o *Real-time Transfer Protocol* (RTP). O RTP fornece serviços de entrega fim a fim para dados com características de tempo real, tais como vídeo e áudio interativos. Estes serviços incluem a identificação do tipo de dado transportado, número de sequenciamento, marcadores de tempo e monitoramento da entrega. As aplicações rodam o RTP normalmente acima do UDP, para fazer uso dos seus serviços de multiplexação, verificação de erros nos dados e reordenamento de pacotes [6]. Por exemplo, os pacotes RTP incluem um número de sequência, desta forma a aplicação pode ao menos detectar a ocorrência de pacotes perdidos, e pode assegurar que os pacotes recebidos estão sendo entregues na ordem correta para o usuário. Os pacotes RTP também incluem um *time-stamp* que corresponde ao tempo que o pacote foi amostrado a partir da fonte. A aplicação de destino pode se

utilizar desta facilidade para assegurar um fluxo sincronizado de dados no usuário final, e também calcular o atraso e a variação do atraso (*jitter*). Deve-se notar que o RTP não trata desses problemas diretamente, ele apenas fornece informações adicionais para uma aplicação de camada superior, e dessa forma a aplicação pode realizar decisões sobre como o pacote deve ser tratado.

O RTP pode trabalhar em conjunto com outro protocolo, o *Real Time Control Protocol* (RTCP) [6]. Este protocolo permite a troca periódica de informações de controle entre os participantes de uma sessão, com o objetivo de fornecer informações de *feedback*. Estas informações podem ser utilizadas para detectar e corrigir problemas. Com a utilização do RTCP, um operador da rede pode, por exemplo, monitorar a qualidade da sessão e detectar problemas na rede.

### 2.1.1 FORMATO DO PACOTE VOIP

Considerando a necessidade do transporte de informações adicionais para o transporte da voz e a utilização de um modelo em camadas, os pacotes de voz transmitidos em uma rede IP apresentam o formato mostrado na Figura 3

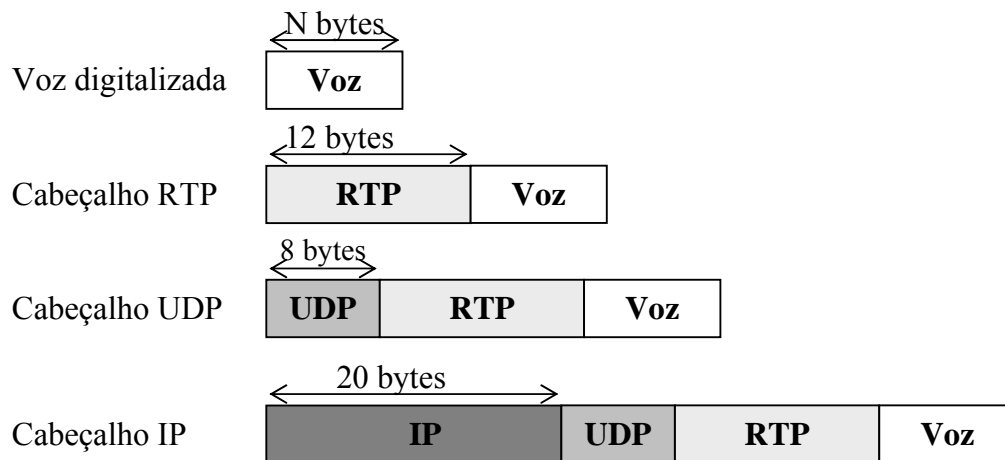


Figura 3 – Formato do pacote VoIP

As informações adicionais necessárias para o transporte de voz sobre IP representam uma transmissão de bits suplementares que pode ter grande influência

na qualidade final do sinal de voz. Da mesma forma o tamanho do quadro de voz digitalizada também é um parâmetro a ser considerado.

## 2.2 QUALIDADE DE SERVIÇO (QoS)

A Qualidade de Serviço é considerada por muitos como um dos principais problemas a serem tratados quando utilizamos voz sobre IP. A QoS, ou a falta dela, é uma das razões pelas quais são oferecidas várias soluções de VoIP gratuitamente pela Internet. A filosofia por trás disso, é que os usuários podem reclamar muito pouco sobre a pouca qualidade dos serviços, pois eles são gratuitos [2].

Quando se trata de utilizar VoIP para substituir a telefonia tradicional, a QoS torna-se um tópico importante.

A QoS é uma medida do nível de serviço entregue ao cliente. Ela pode ser considerada como o nível de garantia dado pela rede, de que uma aplicação particular terá suas exigências atingidas. De um ponto de vista mais técnico, a QoS pode ser caracterizada por diversos critérios de desempenho, tais como a disponibilidade, vazão, tempo de estabelecimento da conexão, percentual de transmissões com sucesso e a velocidade de detecção e correção de falhas.

### 2.2.1 PRINCIPAIS FATORES DE DEGRADAÇÃO

Do ponto de vista de uma rede IP, a QoS pode ser medida em termos de banda, perda de pacotes, atraso e variação do atraso. De forma a oferecer uma QoS alta, a rede IP precisa dar garantias de que, para uma determinada sessão ou conjunto de sessões, a medida dessas características estarão dentro de determinados limites [7].

- ATRASO (*DELAY*)

As aplicações de tempo real são um grupo de aplicações que possuem exigências estritas de atraso. Tipicamente elas possuem um tempo máximo para chegada de pacotes, após o qual os dados tornam-se pouco úteis. A voz está

incluída nesse grupo. No transporte de dados em uma rede IP o atraso não é um fator crítico quando comparamos com o transporte de voz.

O atraso em uma comunicação de voz é formado por vários componentes:

- Atraso de propagação – É o atraso provocado pelo meio físico, e é proporcional à distância entre os pontos finais;
- Atraso de transporte – Causado pelos diversos componentes da rede durante o roteamento dos pacotes.
- Atraso de codificação – É função do codec utilizado, pois cada tecnologia vai exigir um determinado tempo para converter os sinais em pacotes
- Atraso da fila de eliminação da variação do atraso (*jitter buffer*) – Causado pela utilização de uma fila para eliminação da variação do atraso na recepção.

Se considerarmos uma conversação, o atraso fim-a-fim (latência) é uma medida importante, pois leva em conta a interação entre os usuários, caso este atraso seja muito grande, a conversação pode tornar-se inviável. A recomendação G.114 do ITU-T estabelece que o atraso fim-a-fim deve ser de no máximo 300 ms.

- VARIAÇÃO DO ATRASO (*JITTER*)

Assim como o atraso, a variação do atraso também é um fator importante na transmissão da voz. Um atraso constante em uma conversação pode ser tolerado quando mantido dentro de certos níveis, pois os usuários se adaptam naturalmente ao atraso. Entretanto, se existe uma variação nesse atraso, esta adaptação torna-se difícil. A variação do atraso é um problema apenas para as aplicações de tempo real.

Uma técnica comum para remover a variação do atraso é a utilização de uma fila de compensação da variação de atraso. Com essa técnica, os pacotes recebidos, são armazenados em uma fila, e a variação do atraso é atenuada acrescentando-se um atraso adicional aos pacotes que sofreram menor atraso na rede. Os pacotes são entregues na saída com um atraso fixo (*offset*), em relação à origem. O termo ponto de *playback* refere-se ao ponto no tempo que serve de atraso fixo em relação ao tempo de envio original. As aplicações que utilizam filas de compensação de variação do atraso para reconstruir o sinal original, são chamadas de aplicações *playback*. A Figura 4 exemplifica o funcionamento da fila de

compensação. Na figura,  $L$  representa o atraso médio. A fila irá reter os pacotes até o tempo  $T=L+J$ , que representa o ponto de atraso fixo,  $p(x)$  representa a densidade de probabilidade do atraso.

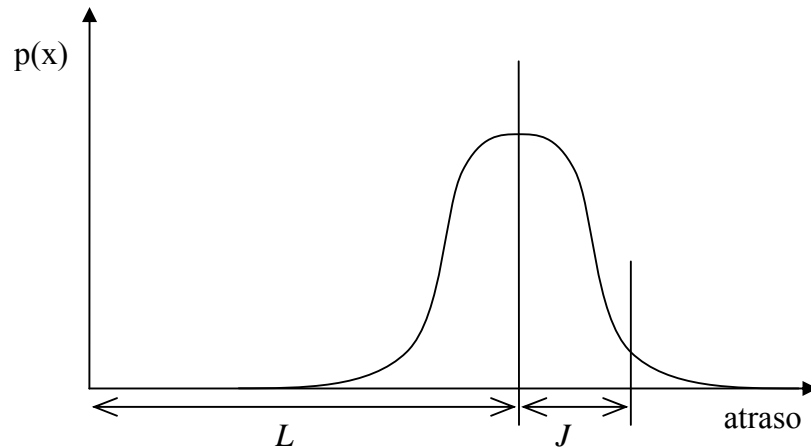


Figura 4 – Funcionamento do *jitter buffer*

O ponto de atraso fixo exerce um papel essencial nas aplicações do tipo *playback*. Os dados que chegam antes são simplesmente armazenados na fila, aguardando pelo ponto de atraso fixo. Os dados que chegam após este ponto estão muito atrasados para fazerem parte da reconstrução do sinal original, o receptor tem então duas opções: ele pode simplesmente descartar o pacote, ou pode mover o ponto de atraso fixo de forma que o pacote possa ser aceito. O ponto de atraso vai depender da tolerância da aplicação ao atraso e ao descarte de pacotes.

Em uma rede IP, o desvio  $D$  no tempo de transmissão e recepção entre dois pacotes  $i$  e  $j$ , pode ser calculado, pela seguinte equação:

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (1)$$

onde  $R_i$  e  $S_i$  são, respectivamente, os tempos de recepção e transmissão do pacote  $i$  e  $R_j$  e  $S_j$  são os tempos de recepção e transmissão do pacote  $j$ .

De acordo com a RFC 1889, a medida da variação do atraso é feita continuamente a cada chegada de pacote, de acordo com a equação:

$$J(i) = J(i-1) + (|D(i-1, i)| - J(i-1)) / 16 \quad (2)$$

- PERDA DE PACOTES

Os erros em uma rede IP são computados em termos de perda de pacotes, ou seja, através do número de pacotes perdidos. A maioria dos pacotes são perdidos devido ao congestionamento ou devido à utilização de filas de compensação da variação no atraso, que descartam os pacotes que sofreram um atraso superior ao desejado.

Caso a taxa de perda de pacotes seja obtida simplesmente através da razão entre número de pacotes perdidos e o número total transmitido, estamos admitindo que as perdas ocorrem com probabilidade  $p$  independente de perdas consecutivas, de acordo com o modelo de Bernoulli. Entretanto, em uma rede real, as perdas normalmente ocorrem em rajadas.

Uma forma de se avaliar a taxa de perdas de pacotes considerando-se as perdas em rajadas é através dos parâmetros do modelo Gilbert [8], onde  $p$  é a probabilidade de um pacote ser descartado dado que o pacote anterior foi recebido, e  $q$  é a probabilidade de um pacote ser recebido dado que o pacote anterior foi descartado. Caso  $p + q = 1$ , o modelo se reduz ao de Bernoulli. A Figura 5 mostra o diagrama de estados do modelo Gilbert, onde o estado 0 corresponde ao pacote recebido, e o estado 1 corresponde ao pacote descartado.

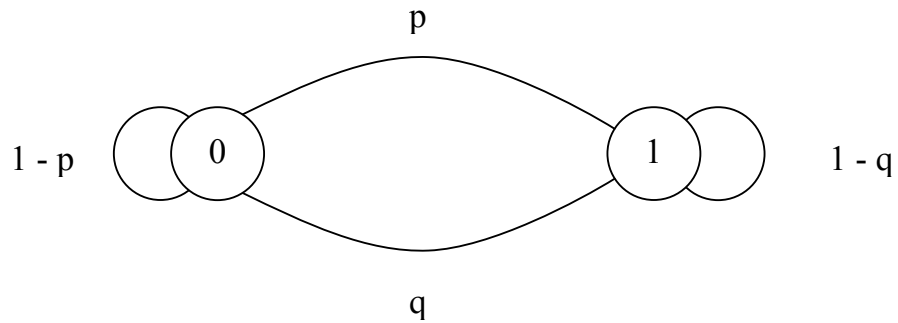


Figura 5 – Parâmetros do modelo Gilbert

Desta forma,  $1-q$  é denominado *conditional loss probability* (CLP), que serve de indicador para a perda em rajadas:

$$CLP = 1 - q \quad (3)$$

A média da perda de pacotes é representada pela *unconditional loss probability* (ULP):

$$ULP = \frac{P}{p + q} \quad (4)$$

### 2.2.2 TRATAMENTO DE QOS

Para tratar a QoS em redes IP, várias soluções foram desenvolvidas. Estas soluções tratam do problema por diversos ângulos. Um deles é assegurar que para cada sessão, os recursos por ela exigidos sejam reservados antes do estabelecimento da sessão. Esta aproximação possui certa similaridade com um serviço de circuito virtual, onde a banda necessária para uma chamada é reservada antes do completamento da chamada.

Uma outra aproximação é classificar o tráfego em diferentes classes, ou prioridades, e designar maiores prioridades para aplicações em tempo real, tais como voz, e menores prioridades para tráfego *best-effort*. De certa forma, esta aproximação é mais fácil de implementar, mais exige que nenhuma aplicação possa impedir totalmente o tráfego de outra.

Finalmente, existe uma técnica bastante comum, o superdimensionamento da rede, isto é, dar uma banda muito maior do que aquela que o tráfego irá necessitar. Alocar mais banda parece uma forma simples e cara de fornecer QoS, simples porque não exige maiores desenvolvimentos no sistema, mas cara porque significa superdimensionar a rede. Este superdimensionamento deve existir de forma que, os recursos da rede devem estar disponíveis nos momentos de rajada de tráfego. Infelizmente, a banda adicional ficará inutilizada na maior parte do tempo. Este método parece ser uma forma bastante ineficiente de resolver o problema do QoS, mas não deve ser desconsiderado totalmente [2].

Uma rede que suporta QoS precisa ter um controle sobre o processo de alocação de recursos, e decidir quem deve receber esses recursos, e quanto deve ser alocado. Uma rede IP, conforme foi concebida originalmente, não suporta nenhuma forma de alocação de recursos ativa. A rede trata cada pacote exatamente da mesma forma, servindo-os em uma base de serviços *first-come first-serve* (FCFS). Não existe um controle de admissão, desta forma os usuários podem injetar pacotes dentro da rede tão rápido quanto o possível.

### 2.2.3 SERVIÇOS DIFERENCIADOS E INTEGRADOS

As arquiteturas Serviços Integrados (*IntServ*) e Serviços Diferenciados (*DiffServ*) são soluções desenvolvidas pela *Internet Engineering Task Force* (IETF) para o provimento de QoS. Essas duas arquiteturas introduziram uma série de novos conceitos e primitivas importantes.

A arquitetura *IntServ* (*Integrated Services*) [9] foi a primeira proposta de padronização para prover QoS em redes IP. Basicamente trata-se de uma alocação por fluxo de tráfego, e tem por objetivo preservar o modelo de datagrama das redes baseadas em IP e ao mesmo tempo suportar a reserva de recursos [10].

Nesta arquitetura, um conjunto de mecanismos e protocolos é utilizado para fazer uma reserva explícita de recursos. Para receber a garantia de performance da rede, uma aplicação deve estabelecer a reserva de recursos ao longo do caminho, antes de começar a transmitir os pacotes. A fonte inicia o pedido de estabelecimento da reserva descrevendo inicialmente as características do fluxo e os recursos necessários. A rede pode aceitar este novo fluxo, apenas se existirem recursos suficientes de acordo com o que foi solicitado. Uma vez feita a reserva, a aplicação pode transmitir os pacotes ao longo do caminho reservado e a rede deverá garantir a reserva durante a sessão.

Para estabelecer a reserva é necessário um protocolo de reserva de recursos que irá a cada salto do caminho fazer a reserva em cada roteador. O protocolo também carrega informações sobre a caracterização do tráfego e necessidades de recursos, de forma que cada nó ao longo do caminho possa determinar se a reserva pode ser aceita ou não. O protocolo também deverá tratar com as mudanças na topologia da rede no caso de, por exemplo, uma queda de *link*. No *IntServ*, o protocolo RSVP foi desenvolvido como o protocolo de reserva de recursos a ser utilizado.

A arquitetura *DiffServ* [11] foi desenvolvida em resposta à necessidade de um método mais simples de prover diferentes níveis de serviço dentro de uma rede IP, para suportar vários tipos de aplicações e necessidades específicas de tráfego. Ela difere do *IntServ* em vários aspectos. O *IntServ* aloca recursos aos fluxos individuais enquanto o *DiffServ* divide o tráfego em um pequeno número de classes com diferentes prioridades. Essa abordagem oferece uma solução mais simples, do ponto de vista de implementação e utilização. O núcleo de uma rede *DiffServ* deve distinguir um pequeno número de classes de encaminhamento (*forwarding classes*) em vez de fluxos individuais. As classes de encaminhamento podem ser carregadas diretamente no cabeçalho do pacote, sem a exigência de uma pré-alocação de recursos. Além disso, uma classificação por fluxo e escalonamento complexos utilizados no *IntServ*, não são necessárias. O tratamento do encaminhamento que pode ser observado externamente em um nó, é descrito pelo termo *per-hop behavior* (PHB). Cada PHB é representado por um valor de 6 bits chamado de *Differentiated Services codepoint* (DSCP). Todos os pacotes com o mesmo DSCP são designados

a um comportamento específico (*behavior aggregate*), e recebem o mesmo tratamento de encaminhamento.

Dois tipos de PHB são definidos: Encaminhamento Expresso (EF) e Encaminhamento Assegurado (AF). Descritos nas RFCs 3246 e 2597 respectivamente [12][13].

O padrão *DiffServ* redefine o campo IP TOS, para indicar comportamentos de encaminhamento (*forwarding behaviors*). O campo chamado DS substitui as definições do octeto TOS do IPv4 e também do octeto *traffic class* do IPv6. Os primeiros 6 bits do campo DS são usados como DSCP para codificar o PHB de um pacote a cada nó. Os 2 bits remanescentes não são utilizados (CU). A Figura 6 mostra o formato dos campos.

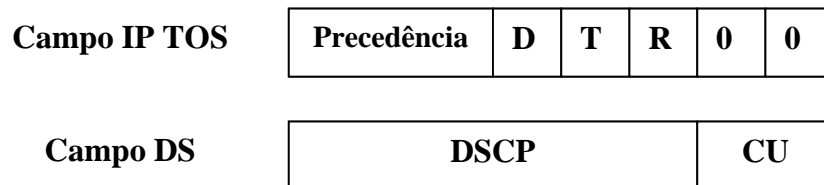


Figura 6 – Campos IP TOS e DS

#### 2.2.4 TÉCNICAS DE ENFILEIRAMENTO PARA QOS

Uma causa de problemas em uma rede de pacotes é que a cada nó, cada pacote pode ser recebido e processado imediatamente, mas ele apenas pode ser repassado para o próximo nó quando existir alguma capacidade disponível. O atraso entre a recepção e a transmissão de um pacote é desta forma, variável.

Uma maneira simples de reduzir o atraso e a variação do atraso para um determinado fluxo de dados é priorizar este fluxo em relação a outros. A idéia é que todos os fluxos devem receber o mesmo serviço, ou um serviço proporcional a sua prioridade, com os fluxos de mesma prioridade sendo tratados da mesma maneira. Existem várias técnicas para realizar uma política de enfileiramento, que decide em cada fila de saída correspondente a um link de transmissão, a ordem de transmissão de um pacote, ou eventualmente seu descarte, e tenta aproximar para cada fluxo de dados certos níveis de capacidade, variação do atraso, latência e perdas.

Essas técnicas podem ser agrupadas em duas categorias. A primeira categoria cuida da ordem dos pacotes nas filas de saída [7]:

- FIFO (*first in first out*), também conhecida como *first come first served* (FCFS) simplesmente transmite os pacotes na ordem em que foram recebidos;
- Class-based queuing (também chamada de *custom queuing* por alguns fornecedores de roteadores);
- Algoritmos de *Fair queuing* e *Weighted fair queuing* (WFQ).

As técnicas de enfileiramento podem ser combinadas com uma segunda categoria, que são técnicas de gerência de descarte de pacotes:

- Overflow simples
- Random early detection (RED) e weighted random early detection (WRED)

### 3 TRATAMENTO DA VOZ E MEDIÇÕES DE QUALIDADE

#### 3.1 CODIFICAÇÃO DOS SINAIS DE VOZ

A maioria dos sistemas para transporte de voz da atualidade utiliza algum padrão de codificação de voz. A voz analógica é codificada digitalmente e transportada através da rede. Uma das vantagens na implementação de VoIP é a oportunidade de se utilizar da eficiência das técnicas de codificação [2].

Inicialmente, pode-se pensar que o ideal seria a utilização da técnica mais eficiente para redução da banda através da codificação, desta forma também estaremos reduzindo os custos. Entretanto, existe uma relação entre a redução da banda e a qualidade da voz que está sendo transmitida. Desta forma, ao se reduzir demasiadamente a banda através de alguma técnica de codificação, estamos correndo o risco de perder substancialmente a qualidade da voz. A relação entre a redução da banda e a perda de qualidade, não é linear [2]. Ela depende também da técnica de codificação que está sendo utilizada.

#### 3.2 SINAIS DE ÁUDIO

Os sinais de áudio podem ser divididos em cinco categorias de qualidade [7]:

- Sinal telefônico de 300 a 3.400 Hz – Um sinal restrito a esta banda permanece claro e compreensível, e não altera o som natural da voz. Entretanto, esta banda não é suficiente para oferecer boa qualidade musical;
- Áudio banda larga de 30 Hz a 7.000 Hz – Voz é transmitida com uma qualidade e fidelidade excelente, mas não é boa o suficiente para a transmissão de música;
- Áudio de alta fidelidade de 20 Hz a 15 kHz – Excelente qualidade para voz e música. Sinais de alta fidelidade podem ser gravados em vários canais para dar uma característica espacial na reprodução;
- Qualidade de CD de 20 Hz a 20 kHz;
- Qualidade profissional de 20 Hz a 48 kHz.

### 3.3 SINAIS DE VOZ

Os sons de voz são caracterizados pelo formato do aparelho vocal, que consiste das cordas vocais, lábios e nariz. A voz é gerada quando o ar é empurrado a partir dos pulmões, passando ao longo do aparelho vocal. As vibrações básicas ocorrem nas cordas vocais, mas o som é alterado pela disposição da língua e lábios. O sistema vocal pode ser considerado como um filtro, e várias tecnologias de codificação tentam modelá-lo desta forma. Como o formato do sistema vocal varia relativamente devagar, a função de transferência do filtro precisa mudar da mesma forma. Representando o sistema vocal como um filtro, a excitação feita pelas vibrações das cordas vocais, correspondem ao sinal aplicado ao filtro.

Os sinais de voz podem ser classificados em três principais tipos [2]:

- Sons vozeados são produzidos quando as cordas vocais abrem e fecham, provocando uma vibração através da interrupção do fluxo de ar. O período de vibração das cordas vocais determina o *pitch* do som. Esta frequência está na faixa de 50 a 400 Hz e é geralmente menor nas vozes masculinas, e maior nas vozes femininas e infantis;
- Sons fricativos são produzidos quando a excitação é uma turbulência em forma de ruído, produzida forçando-se o ar em altas velocidades através de uma compressão do sistema vocal;
- Sons explosivos resultam de um completo fechamento do sistema vocal, o ar é então pressionado através desse fechamento que é então aberto repentinamente.

A combinação desses tipos básicos provoca uma grande variedade de sons, entretanto, a mudança no formato do sistema vocal e o tipo de vibração das cordas vocais variam relativamente devagar, e o sinal produzido é relativamente previsível ao longo do tempo. Esta previsibilidade pode ser utilizada no projeto de sistemas de codificação de voz, de forma a minimizar a quantidade de dados a serem transmitidos, diminuindo desta forma a banda necessária para a transmissão.

### 3.4 DIGITALIZAÇÃO DO SINAL DE VOZ

A conversão Analógica/Digital é o processo utilizado para representar uma quantidade de precisão infinita, que no caso dos sinais de voz corresponde à variação da amplitude de um sinal no tempo, por um conjunto finito de números com uma taxa de amostragem fixa ao longo do tempo. Este conjunto finito de valores de amplitude representa os níveis de quantização e cada amostra descreve a amplitude do sinal original em um instante específico. A conversão Analógica/Digital é caracterizada pela taxa de amostragem e pelo método de quantização [14].

#### 3.4.1 MODULAÇÃO POR CÓDIGO DE PULSO (PCM)

Na modulação por código de pulso, um sinal é representado por uma seqüência de pulsos codificados, o que é conseguido através da representação do sinal na forma discreta, no tempo e na amplitude. As operações básicas realizadas no transmissor de um sistema PCM são: amostragem, quantização e codificação. As operações de quantização e codificação são realizadas normalmente pelo mesmo circuito, que é chamado de conversor analógico/digital. As operações básicas realizadas no receptor são: regeneração, decodificação e reconstrução do trem de amostras quantizadas.

Matematicamente, o processo de amostragem pode ser definido como o resultado da multiplicação de um trem de pulsos periódicos de amplitude 1, pelo sinal original a ser amostrado. Este processo leva à representação do sinal na forma PAM (*pulse amplitude modulation*). A partir do sinal PAM, é possível regenerar um sinal contínuo no tempo, através de um conversor de sinal discreto para contínuo (D/C). A Figura 7 exemplifica o processo:

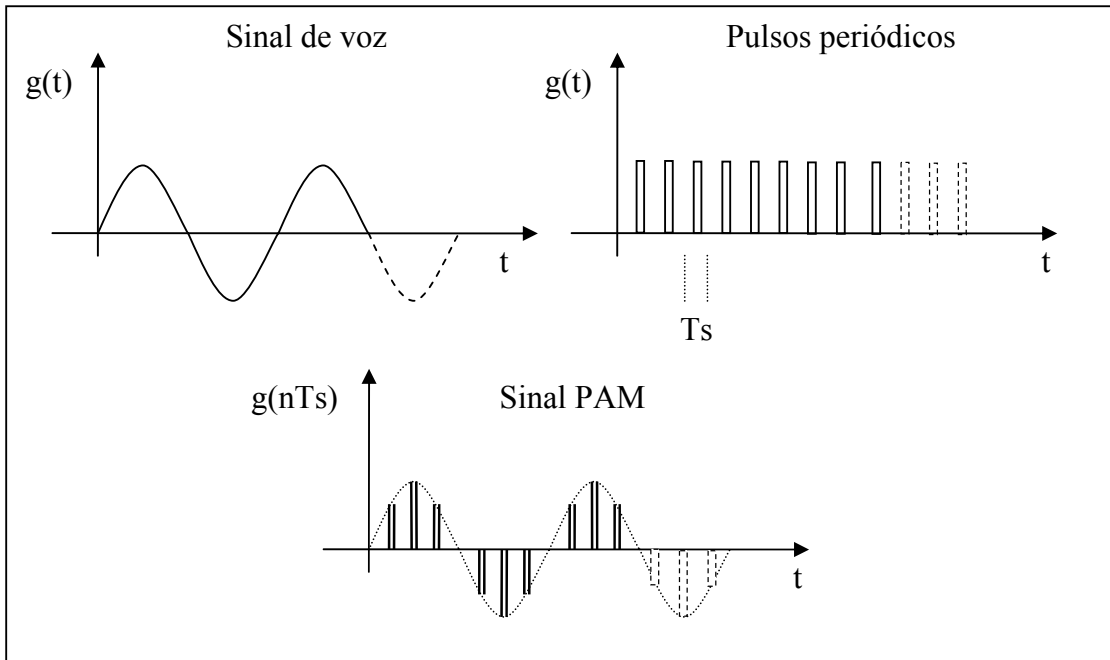


Figura 7 – Processo de amostragem

Após o processo de amostragem, a versão amostrada do sinal é então quantizada, obtendo-se desta forma, uma nova representação do sinal, que é discreta no tempo e na amplitude. O processo de quantização pode seguir uma lei uniforme, onde a separação entre os níveis do sinal é espaçada igualmente, ou pode utilizar uma separação variável.

Após os processos de amostragem e quantização, o sinal original passa a ser representado através de um conjunto de valores discretos, mas essa não é a melhor forma de transmiti-lo através do meio de transmissão. Para representar cada conjunto de valores amostrados, é utilizado um processo de codificação. Cada um dos eventos discretos de um código é chamado de símbolo. Um valor distinto do conjunto de valores discretos pode ser representado por uma seqüência de símbolos, que formam uma palavra ou caracter. A Figura 8 ilustra os processos de quantização e codificação.

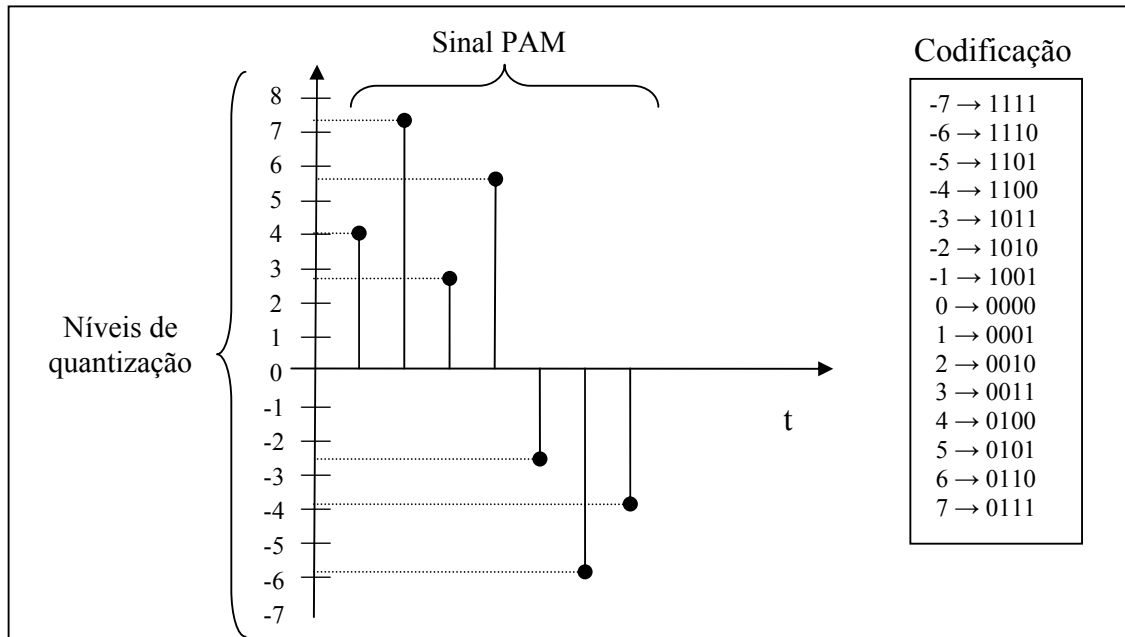


Figura 8 – Processos de quantização e codificação

### 3.5 CODIFICADORES/DECODIFICADORES DE VOZ (CODECS)

Os algoritmos básicos utilizados pela maioria dos codificadores de voz são [7]:

- Quantização adaptiva
- Quantização diferencial
- Predição Linear
- Predição de longo termo
- Quantização vetorial
- Codificação por entropia

Existem três categorias de codecs: forma de onda, paramétricos (vocoders) e híbridos [2].

Os codificadores de forma de onda basicamente fazem a amostragem e codificação do sinal analógico, sem se preocupar com a maneira com que o sinal foi produzido. Eles transmitem valores quantizados das amostras até o receptor, onde o

sinal é reconstruído. Geralmente eles produzem um sinal de alta qualidade e não são muito complexos. A maior desvantagem deste tipo de codec, é que eles consomem grande quantidade de banda, comparado com outros codecs. Quando um codificador de forma de onda funciona em taxas de transmissão mais baixas, a qualidade de voz diminui significativamente.

Os vocoders tentam casar o sinal de entrada com um modelo matemático de como o sinal é produzido. Eles normalmente utilizam um modelo de filtro para representar o sistema vocal, com um bit para representar sinais vozeados e não-vozeados. A informação é enviada até o receptor na forma de um conjunto de parâmetros que representam o sinal, e não na forma do próprio sinal. O receptor então, utiliza os valores recebidos e o mesmo modelo do transmissor para reconstruir o sinal analógico. Os vocoders operam em taxas de transmissão baixas, mas tendem a produzir uma voz que soa sintética.

Os codecs híbridos tentam utilizar o melhor das duas técnicas mostradas anteriormente. Apesar de tentar reproduzir em um certo nível, a forma de onda original, ele também utiliza o conhecimento sobre como os sons foram produzidos. Eles tentam oferecer uma boa qualidade em taxas mais baixas que os codecs de forma de onda.

### 3.5.1 ITU G.711

O G.711 [15], é a técnica mais comum de codificação de voz utilizada na atualidade, ele possui as seguintes características:

- 8 kHz de taxa de amostragem;
- Utiliza 8 bits para representar cada amostra;
- Quantização não-uniforme;
- 64 kbit/s de taxa de transmissão
- Atraso desprezível.

O processo da codificação começa com um sinal digital quantizado linearmente em 12 bits, no formato sinal + amplitude. Para cada amostra de 12 bits, o conversor G.711 gera uma saída no formato de um código de 8 bits, sendo 1 bit de

sinal, 3 bits de expoente ( que representa o segmento) , e 4 bits de mantissa. A Figura 9 mostra a representação deste código.

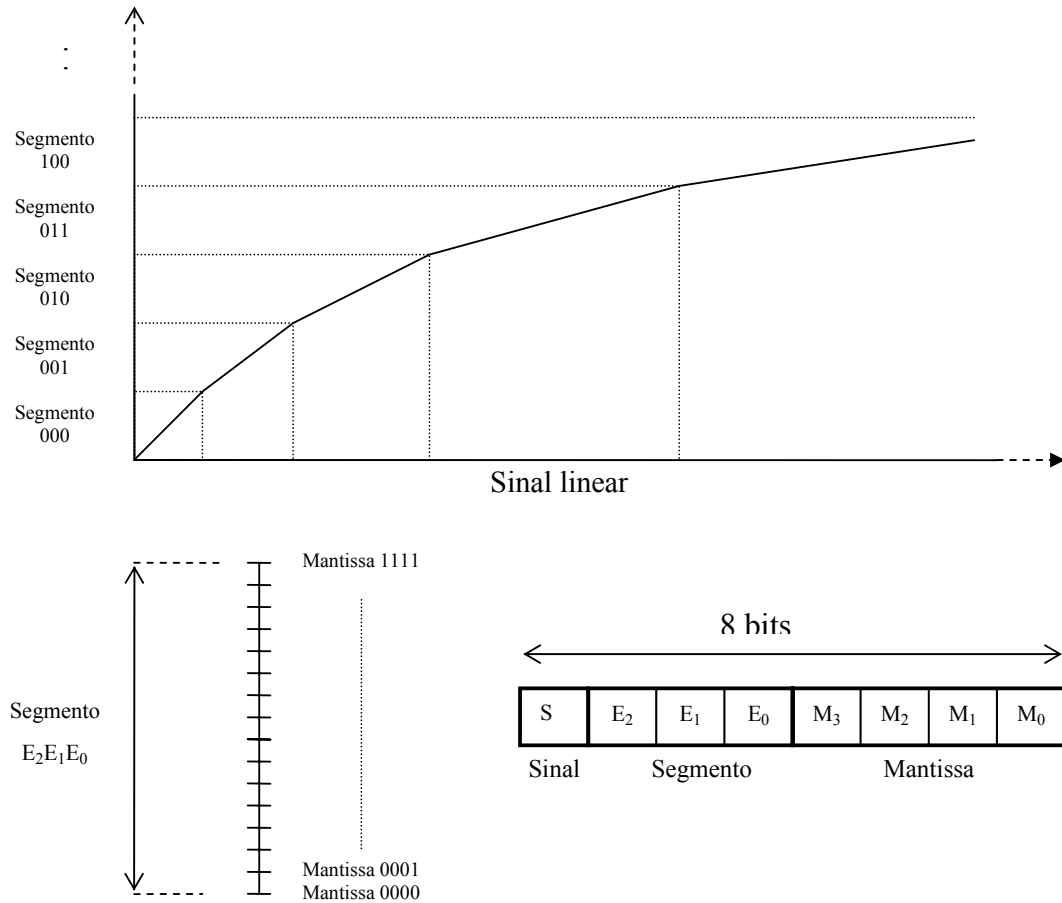


Figura 9 – Codificação G.711

Codificado desta forma, o sinal apresenta uma qualidade equivalente a de um sinal codificado linearmente em 12 bits (96 kbit/s), mas com uma taxa de transmissão de apenas 64 kbit/s. Esse ganho é conseguido em detrimento da relação sinal ruído nos níveis de potência mais altos, entretanto perceptualmente a qualidade não é afetada [2].

### 3.5.2 PCM DIFERENCIAL ADAPTATIVO (ADPCM)

Codecs PCM tais como o G.711, enviam amostras individuais até o destino, onde a forma de onda original é reconstruída. As variações de voz, entretanto, ocorrem relativamente devagar, e é possível prever o valor de uma determinada amostra baseando-se no valor de amostras anteriores. Neste caso, necessitamos transmitir apenas a diferença entre o valor previsto, e o valor real da amostra. Como no receptor está sendo realizada a mesma previsão, ele pode determinar o valor da amostra original, caso seja informado o valor da diferença entre o sinal previsto e o valor real da amostra. Esta técnica é conhecida como PCM diferencial.

Uma versão mais avançada do DPCM é o DPCM adaptativo (ADPCM). O ADPCM tipicamente faz a previsão das amostras baseando-se nas amostras anteriores, e também levando em conta o conhecimento de como a voz varia ao longo do tempo. O erro entre a amostra real e a prevista é quantizado e então enviado ao receptor. As recomendações G.721 e G.726 [16] são exemplos de ADPCM.

### 3.5.3 EXEMPLOS DE CODECS HÍBRIDOS

Os codecs híbridos tentam preencher o intervalo entre os codificadores de forma de onda e os vocoders. O objetivo dos codecs híbridos é oferecer voz de boa qualidade em uma taxa mais baixa que dos codecs de forma de onda.

Apesar de existirem outras formas de codecs híbridos, os mais utilizados são os codecs análise por síntese (AbS). Esses codecs utilizam o mesmo modelo de filtro de predição linear utilizado nos vocoders LPC (*Linear Predictive Coding*) mas transmitem informações adicionais que permitem uma melhor qualidade de voz.

- G.728

Os codificadores CELP (*Code Excited Linear Prediction*) implementam um filtro cujas características mudam ao longo do tempo, e contém uma lista de vetores acústicos. Nos codecs CELP, o que é transmitido para o receptor é um conjunto de

informações que indicam os coeficientes do filtro, ganho, e um ponteiro para o vetor de excitação escolhido. A ponta remota contém uma lista igual a do transmissor, e um filtro com as mesmas características, desta forma ela pode reconstruir o sinal original com grande grau de precisão. A Recomendação G.728 [17] especifica o *Low-Delay CELP* (LD-CELP), que é um codec adaptativo com retroação, pois ele utiliza amostras anteriores para determinar os coeficientes do filtro. Este codec apresenta as seguintes características:

- Trabalha com sinais de voz com 8 kHz de taxa de amostragem;
- Utiliza quadros de 5 amostras;
- Taxa de transmissão de 16 kbit/s;
- Provoca atraso menor que 1 ms

- G.723.1

A recomendação G.723.1 [18] especifica um codec do tipo *Algebraic Code Linear Prediction* (ACELP) que pode operar em taxas de transmissão variável, sendo que a taxa pode ser alterada durante uma conversação. Suas características principais são:

- Trabalha com sinais de voz com 8 kHz de taxa de amostragem;
- Utiliza quadros de 240 amostras (30ms);
- Taxa de transmissão de 6,3 ou 5,3 kbit/s;
- Provoca atraso de 37,5 ms.

O codificador trabalha com um sinal de voz limitado em banda, amostrado a 8.000 Hz, e que é quantizado linearmente utilizando 16-bits. São utilizados quadros de 240 amostras, correspondentes a 30 ms de voz. O codec utiliza também mais 7,5 ms de *look-ahead*, resultando em um atraso total de 37,5 ms. Cada quadro passa por um filtro passa altas para remover qualquer componente DC, e é dividido em 4 subquadros de 60 amostras. Várias operações são realizadas nestes subquadros, de forma a determinar os coeficientes de filtro adequados. *Algebraic Code-Excited Linear Prediction* (ACELP) é utilizada no caso da taxa de 5,3 kbit/s, e *Multipulse Maximum Likelihood Quantization* (MP-MLQ) no caso da taxa de 6,3 kbit/s.

A informação transmitida para o receptor inclui os coeficientes de predição linear, parâmetros de ganho, e os índices da tabela de vetores. Esta informação é composta de quadros de 24 octetos na taxa de 6,3 kbit/s e quadros de 20 octetos na taxa de 5,3 kbit/s.

No G.723.1 Anexo A, é especificado um mecanismo de supressão de silêncio, onde são utilizados quadros *Silence Insertion Description* (SID), de 4 octetos.

O G.723.1 possui uma qualidade que pode ser considerada boa, apesar da redução da taxa de transmissão, mas possui a desvantagem de possuir um atraso mínimo de 37,5 ms no codificador.

- G.729

A recomendação básica do G.729 [19] especifica um codec ACELP com as seguintes características:

- Trabalha com sinais com 8 Kbits de taxa de amostragem e 16 bits por amostra;
- Utiliza quadros de 10 ms, correspondentes a 80 amostras;
- Taxa de transmissão de 8 kbit/s;
- Provoca atraso de 15 ms.

O G.729 é um codec complexo, e de forma a reduzir esta complexidade no algoritmo, algumas simplificações foram introduzidas no G.729 Anexo A. Na transmissão é utilizada a mesma estrutura do G.729, e um decodificador G.729A pode ser utilizado em conjunto com um transmissor G.729. Esta simplificação no algoritmo produz uma ligeira redução de qualidade.

O G.729 anexo B é uma recomendação para a *voice activity detection* (VAD), *discontinuous transmission* (DTX) e *comfort noise generation* (CNG).

O VAD é uma decisão de transmitir, no caso da voz ou ruído estejam presentes na entrada. Esta decisão é realizada baseando-se na análise do quadro atual, e de dois quadros anteriores. Este mecanismo assegura que a transmissão ocorrerá por pelo menos dois quadros após o encerramento da atividade da voz.

A outra decisão a ser feita, é se o quadro SID será transmitido. O quadro SID contém alguma informação que permite ao decodificador simular o ruído de fundo. O quadro SID do G.729B possui 15 bits.

O G.729 anexo B é uma extensão para a produção de uma taxa de transmissão mais baixa. O G.729B trabalha com amostras de 10 ms, entretanto em vez de transmitir quadros de 80 bits, ele utiliza quadros de 64 bits, resultando em uma taxa de 6,4 kbit/s. A utilização do G.729B resulta em uma qualidade semelhante ao do G.723.1 a 6,3 kbit/s.

O G.729 anexo E oferece melhoras ao algoritmo G.729, em uma taxa mais elevada. A intenção é oferecer melhor desempenho na presença de ruído de fundo significativo na entrada. O G.729 utiliza um filtro de predição linear de 10 coeficientes, já no G.729E utiliza-se de um filtro de 30 coeficientes, além de um índice de 44 bits para os vetores, resultando na transmissão de 118 bits e uma taxa de 11,8 kbit/s.

### 3.6 QUALIDADE DE VOZ

Vários fatores podem influenciar a qualidade da voz. Esses problemas são bem conhecidos e comuns em redes de voz tradicionais. Essas redes são projetadas de forma a minimizar esses problemas e torná-los imperceptíveis para os usuários. Entre os fatores causadores de perda de qualidade podemos citar o atraso, eco, e ocorrência de erros. Para combater esses problemas diversas técnicas podem ser utilizadas tais como: utilização de supressores ou canceladores de eco, utilização de redundância e algoritmos de correção de erros.

Em redes VoIP alguns desses problemas podem ser causados por fatores diferentes dos encontrados em redes tradicionais.

#### 3.6.1 MEDIDAS DA QUALIDADE DE VOZ

A maioria dos codecs de voz tenta oferecer a melhor qualidade possível dentro dos limites impostos pela taxa de transmissão, entretanto vários fatores podem influir na qualidade final percebida pelo usuário. A medida da qualidade de

voz em uma rede de telefonia pode ser realizada por métodos subjetivos, baseados na avaliação feita por um conjunto de ouvintes, e objetivos baseados em uma série de medidas sobre o sinal ou parâmetros da rede. Para medir o nível de qualidade, medidas objetivas não são confiáveis para um novo codec. Para a utilização de uma ferramenta de medida automatizada, é necessário ter conhecimento dos valores de qualidade já conhecidos e medidos através de métodos subjetivos. As ferramentas utilizadas para medição de qualidade em redes tradicionais também não oferecem precisão quando utilizadas para medir a qualidade em redes VoIP, pois as propriedades do meio de transmissão são diferentes nas duas redes, por exemplo as degradações que numa rede TDM são medidas em termos de erros de bit, em uma rede IP são medidas em termos de perda de pacotes, além de outros novos fatores tais como a variação do atraso e atraso causado pela utilização de filas [7].

Medidas subjetivas são desse modo indispensáveis, pois uma medição através da percepção do ouvido humano é o meio mais preciso. Entretanto, medidas subjetivas precisas exigem um grande esforço para sua realização, pois devem seguir várias normas, como as que seguem:

- Assegurar que o número total de ouvintes é suficiente para um resultado estatisticamente confiável;
- Assegurar que a percepção auditiva dos ouvintes é normal;
- Instruir corretamente os ouvintes sobre a metodologia dos testes;
- Assegurar que o material utilizado é diversificado;
- Assegurar que os testes são realizados em várias línguas, por um número de organizações experientes;
- Assegurar que todas as condições de uso do codec são testadas;
- Escolha adequada das condições em que são realizados os testes.

Estes testes são especificados nas recomendações P.800 e P830 do ITU-T, e alguns dos métodos são descritos a seguir.

### 3.6.2 MEDIÇÃO SUBJETIVA DA QUALIDADE

Para codecs de taxas mais baixas, entre 4 kbit/s e 32 kbit/s, o teste ACR (*Absolute Category Rate*) é o mais utilizado método de medição subjetiva. Nos testes ACR, os ouvintes são questionados para que classifiquem a qualidade absoluta do sinal de voz, sem saber qual é a referência de áudio que está sendo utilizada. Este método produz o *mean opinion score* (MOS), que é uma escala utilizada para medir a qualidade, mostrada na Tabela 1.

<b>Excelente</b>	<b>5</b>
<b>Boa</b>	<b>4</b>
<b>Fraca</b>	<b>3</b>
<b>Pobre</b>	<b>2</b>
<b>Ruim</b>	<b>1</b>

Tabela 1 – Escala MOS

O MOS é um julgamento absoluto sem referências, mas para que haja uma coerência e calibração entre testes sucessivos, é necessária alguma referência. Para isso, um sinal de áudio de referência é inserido entre os sinais que estão sendo julgados pelos ouvintes. Frequentemente é utilizada a *modulated noise reference unit* (MNRU), que é um equipamento que simula a degradação de voz e nível de ruído equivalente ao produzido pela codificação PCM.

Além do ACR, também são usados o *degradation category rating* (DCR) e o *comparison category rating* (CCR). O método DCR é utilizado quando sinais de boa qualidade estão sendo comparados. Este método produz um *degradation mean opinion score* (DMOS). A metodologia é semelhante ao ACR, exceto pelo fato do sinal de referência ser conhecida pelos ouvintes, e apresentada em primeiro lugar. O CCR é similar ao DCR, mas a ordem do sinal de referência e do codec avaliado, é escolhida aleatoriamente. O resultado é o *comparison mean opinion score* (CMOS).

Para sistemas de comunicação interativos, especialmente VoIP, testes conversacionais também são bastante úteis, pois eles tentam reproduzir as mesmas condições do serviço que é prestado aos usuários. As degradações introduzidas pelo eco e atrasos, e que não estão presentes nos testes MOS, podem ser então

levadas em consideração. Os resultados também são dados em uma escala de 1 a 5, e produzem o MOSc. Estes testes apresentam grande dificuldade de realização, e a consistência e repetibilidade são difíceis de serem obtidas. A

Codec	Tipo de codec	Tamanho do frame (ms)	Atraso total (ms)	Taxa de transmissão (kbit/s)	MOS
G.711	PCM	0,125	0,125	64	4.2
G.721	ADPCM	0,125	0,125	32	4.0
G.726	ADPCM	0,125	0,125	16/24/32/40	4.0 (32 kbit/s)
G.728	LD-CELP	0,625	5	16	4.0
G.729A	CS-ACELP	10	15	8	3.7
G.723	ACELP	30	37,5	6,3/5,3	3,9/3,7

**Tabela 2** mostra os valores de MOS para diversos codecs [7]:

Codec	Tipo de codec	Tamanho do frame (ms)	Atraso total (ms)	Taxa de transmissão (kbit/s)	MOS
G.711	PCM	0,125	0,125	64	4.2
G.721	ADPCM	0,125	0,125	32	4.0
G.726	ADPCM	0,125	0,125	16/24/32/40	4.0 (32 kbit/s)
G.728	LD-CELP	0,625	5	16	4.0
G.729A	CS-ACELP	10	15	8	3.7
G.723	ACELP	30	37,5	6,3/5,3	3,9/3,7

Tabela 2 – MOS para diversos codecs

### 3.6.3 MEDIÇÃO OBJETIVA DA QUALIDADE

Como os métodos para medição subjetiva são dispendiosos e demandam tempo, são necessárias outras formas de avaliação que possam ser utilizadas mais amplamente, através da medição das características físicas dos terminais e redes.

As metodologias de medição objetiva de qualidade podem ser classificadas em vários grupos a partir do ponto de vista do objetivo, procedimento de medida, informação de entrada e MOS. As metodologias que exploram aos parâmetros de qualidade da rede e terminais, e produzem uma estimativa do MOS conversacional, podem ser classificadas como *opinion models*. As metodologias que necessitam de

um sinal de voz como entrada e produzem uma estimativa do MOS, podem ser classificadas como *speech-layer objective models*. Já aquelas que exploram as características do pacote IP e produzem uma estimativa do listening MOS podem ser chamadas de *packet-layer objective models* [20].

#### 3.6.4 MODELOS DE OPINIÃO

Os modelos de opinião (*opinion models*) têm sido estudados pelo ITU-T há bastante tempo, e vários modelos foram propostos como candidatos a tornarem-se um padrão.

O E-model foi proposto para padronização nos anos 90. Ele é baseado no modelo OPINE da NTT, em que todos os fatores responsáveis pela degradação da qualidade são somados em uma escala psicológica.

O E-model mede os efeitos combinados de vários parâmetros de transmissão que afetam a qualidade da voz. Ele assume que esses parâmetros são aditivos, e podem ser transformados em uma escala de fatores psicológicos, gerando uma escala representada pelo fator R, que mede a qualidade da transmissão [21]:

$$R = R_0 - I_s - I_d - I_e + A \quad (5)$$

onde  $R_0$  representa a relação sinal-ruído,  $I_s$  representa perdas que ocorrem simultaneamente ao sinal de voz,  $I_d$  representa degradação provocada pelo atraso (delay),  $I_e$  representa a degradação provocada pelo equipamento e A é um fator de compensação, ou expectativa do usuário. Tanto  $R_0$  quanto  $I_s$  descrevem parâmetros que são relacionados apenas com o sinal, não dependendo da rede de transporte. Em uma rede IP podemos dessa forma computar o fator R em função dos fatores  $I_d$  e  $I_e$  que são relacionados respectivamente ao delay e à taxa de perda de pacotes [22].

O Fator R varia em uma escala de 0 a 100, e pode ser mapeado para a escala MOS, a Figura 10 mostra uma comparação dessas duas escalas e o nível de satisfação do usuário.

<b>R</b>	<b>Satisfação do usuário</b>	<b>MOS</b>
100 -----	Muito Satisfatório	----- 4.5
90 -----	Satisfatório	----- 4.3
80 -----	Alguns usuários insatisfeitos	----- 4.0
70 -----	Muitos usuários insatisfeitos	----- 3.6
60 -----	Quase todos usuários insatisfeitos	----- 3.1
50 -----	Não recomendado	----- 2.6
0 -----		----- 1

Figura 10 – Comparação entre o fator R e escala MOS

Os parâmetros dos fatores de degradação são avaliados a partir de testes subjetivos reais, de forma que o modelo seja calibrado. Desta forma, a validade do E-model para uma determinada tecnologia depende da calibração feita previamente para esta tecnologia. O E-model é útil apenas se usado corretamente. Por exemplo, um fator de degradação medido sob determinado perfil de perda é válido apenas para este mesmo perfil. Novas versões do E-model tem sido desenvolvidas para se adaptar às degradações que são específicas de uma rede IP [23].

### 3.6.5 MODELOS OBJETIVOS *SPEECH LAYER*

O estudo dos modelos objetivos do tipo *speech-layer* começaram com o uso da relação sinal-ruído (SNR) como meio de avaliar os sinais de voz codificados em PCM. Na segunda metade dos anos 80, vários modelos objetivos que exploravam a distorção espectral em vez da distorção da forma de onda foram propostos como métodos objetivos de qualidade mais aplicáveis para a avaliação dos codecs de taxa baixa. Entretanto, devido à falta de precisão na estimação, nenhum deles foi

padronizado como uma recomendação do ITU-T. Mais tarde, um modelo baseado na distorção espectral Bark, ofereceu uma precisão adequada, e formou a base da recomendação P.861, *Perceptual Speech Quality Measure* (PSQM), em 1998.

Entretanto, devido às características de perda de pacotes das redes IP, esta aproximação não é apropriada para as medidas em VoIP. Um algoritmo mais sofisticado foi criado, e padronizado na recomendação P.862, *Perceptual Evaluation of Speech Quality* (PESQ), em 2001 [24].

- PSQM

O PSQM (*Perceptual Speech Quality Measure*) é um método para estimar a qualidade de codecs de voz que faz parte da recomendação ITU-T P.861, que foi posteriormente substituída pela recomendação P.862. Ele foi desenvolvido pela empresa de telecomunicações holandesa KPN em 1997 [25]. Neste algoritmo, a medida da qualidade é feita através de um modelo psico-acústico que procura reproduzir a percepção do ouvido humano. A Figura 11 mostra um diagrama da filosofia utilizada no desenvolvimento do PSQM onde um modelo da percepção auditiva humana é construído e é feita uma comparação entre os sinais codificados [26].

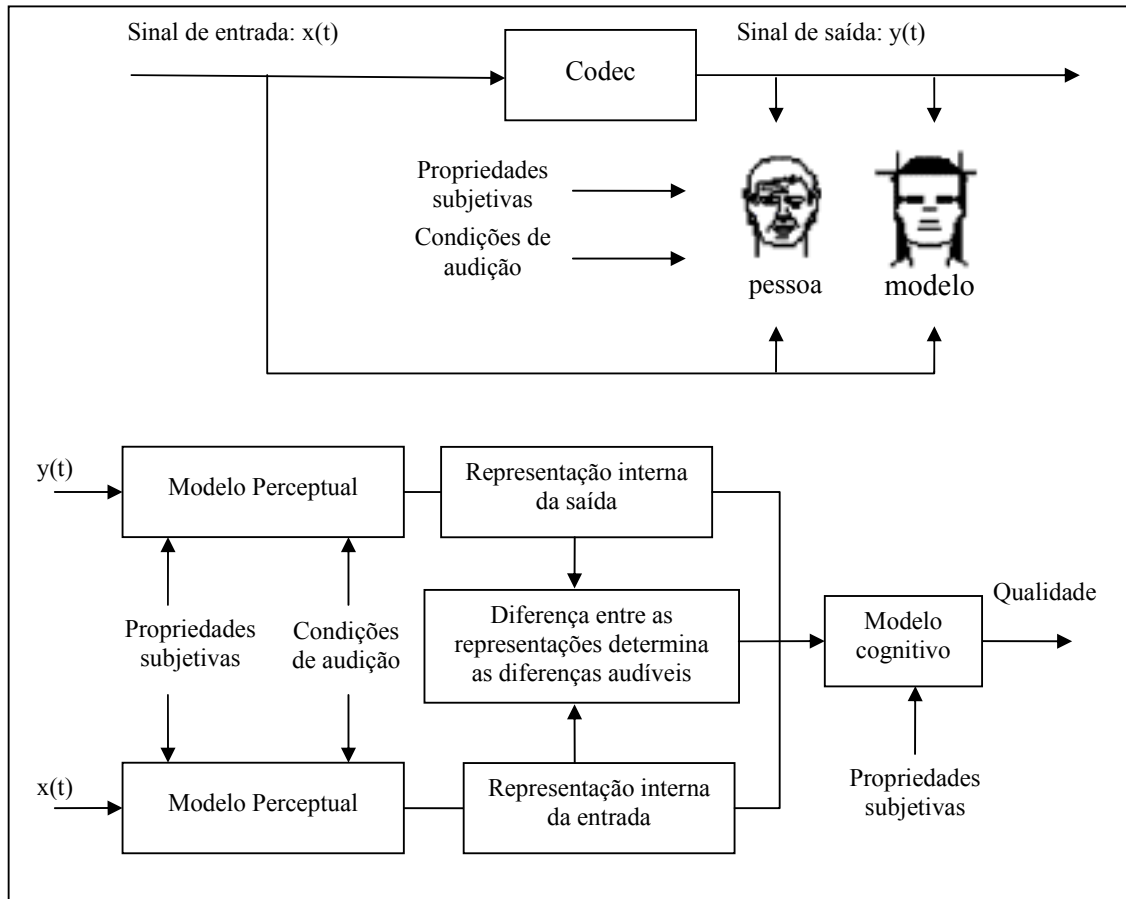


Figura 11 – Diagrama do PSQM de acordo com a recomendação P.861

O sinal é convertido para o domínio psico-acústico através de três operações [26]:

- Mapeamento tempo-freqüência – é implementado através de uma FFT com uma janela Hanning;
- Mudança na escala de freqüência – é realizada através da conversão da escala em Hertz para uma escala em Bark;
- Mudança na escala de amplitude – é feita uma compressão de amplitude de acordo com a sensibilidade auditiva (*loudness*).

Através da comparação entre o sinal original e o distorcido, é obtido um fator chamado de perturbação de ruído. A distorção é computada para cada quadro com

tamanho de 256 amostras e 50% de overlap. O resultado mostra a perturbação de ruído com relação ao tempo e à frequência, e a média desses valores é relacionada com a qualidade do codec. O resultado obtido pelo algoritmo é denominado valor PSQM. Ele indica o grau de degradação através de uma escala que varia de 0 a 6.5, sendo que 0 corresponde a um sinal sem degradação, e 6.5 a degradação máxima. O valor PESQ pode ser convertido para a escala MOS de acordo com a equação a seguir [27]:

$$MOS = \frac{4}{1 + e^{0.66PSQM - 2.2}} + 1 \quad (6)$$

O PSQM+ foi proposto como uma melhora ao algoritmo PSQM. Ele leva em consideração o valor PSQM e a potência do sinal, de forma que variações no volume do sinal são contabilizadas no valor final. O maior problema do PSQM é o alinhamento do sinal original com o sinal medido, pois o delay da rede é desconhecido e pode sofrer variações, mas para que o algoritmo compare os dois sinais, é necessário o perfeito alinhamento. Desta forma, o algoritmo também não considera a degradação devida ao *delay* e a variação do atraso [28].

- MNB

O MNB (*Measuring Normalizing Blocks*) foi desenvolvido pelo instituto de ciências de telecomunicações do departamento de comércio americano em 1997. Ele modela o julgamento humano da qualidade através da análise no tempo e na frequência. Essas duas análises são combinadas de forma que seja obtido um valor chamado *Auditory Distance* (AD), que mede a distorção. Este algoritmo também faz parte da recomendação P.861 do ITU [26].

A medida é feita através do alinhamento entre o sinal original e o sinal a ser medido. São então eliminados os componentes DC, e posteriormente é estimada a potência média dos dois sinais, para que sejam normalizados. O passo seguinte é a transformação do sinal para o domínio da frequência utilizando uma FFT com janela Hamming de 128 amostras (16 ms) e overlap de 50 %. Os quadros resultantes são comparados entre os dois sinais, de forma que são eliminados os quadros cujas

diferenças estão abaixo de um determinado valor, e também os quadros com componentes de frequência com potência zero. Os quadros que não foram eliminados são transformados de acordo com uma escala de *loudness*, e comparados no domínio do tempo e da frequência. São então obtidos valores que representam as diferenças entre os sinais para diversos intervalos de frequências e é feita uma combinação desses valores, que representa o valor medido [25].

- PAMS

Em 1998, a British Telecom desenvolveu o PAMS (*Perceptual Analysis Measurement System*). Este algoritmo também faz uma análise no tempo e na frequência e inclui o sincronismo no tempo, que era um problema que não havia sido resolvido nos algoritmos apresentados na recomendação P.861.

- PESQ

O algoritmo PESQ (Perceptual Evaluation of Speech Quality) é um padrão do ITU para medida de qualidade descrito na recomendação P.862. Ele foi criado em conjunto pela KPN e British Telecom através da combinação do PSQM+ e do PAMS, e foi desenvolvido especificamente para poder ser utilizado em redes como VoIP e ISDN, já que os algoritmos da recomendação P.861 não eram eficientes para tratar dos problemas específicos destas redes. O PESQ apresenta a medida da qualidade diretamente na escala MOS [29]

O algoritmo PESQ basicamente segue os mesmos passos usados no PSQM, entretanto são introduzidas modificações de forma a melhorar sua performance. Primeiramente há uma compensação do ganho nos sinais original e degradado, de forma que apresentem o mesmo nível de potência, essa calibração é feita tanto no domínio do tempo quanto na frequência. Após essa etapa, é realizada uma filtragem nos sinais, de forma que o sinal a ser analisado tenha a mesma característica de um sinal ouvido após atravessar uma rede telefônica, os sinais são então alinhados no tempo de forma que é definido o intervalo em que vai ser feita a análise. A conversão para o domínio da frequência é feita utilizando uma janela Hanning sobre quadros de 32 ms e 50% de *overlap*. A análise é feita sobre as

diferenças no domínio da frequência na escala Bark, após serem realizadas as compensações de loudness, e de variações de ganho.

Devido a sua melhor performance em relação aos outros modelos apresentados para a medição da qualidade dos sinais na faixa de voz, o PESQ tornou-se uma nova recomendação P.862, substituindo a anterior P861 [30].

### 3.6.6 MODELOS OBJETIVOS *PACKET-LAYER*

O ITU está também tentando padronizar uma metodologia de medição de qualidade baseada apenas nas informações do pacote IP. Esta metodologia está sendo provisoriamente chamada de P.VTQ. Este procedimento inicia com a estimativa dos parâmetros de qualidade intermediários como a taxa de perda de pacotes, padrão da perda e variação do atraso obtidos a partir do cabeçalho RTP e das informações do RTCP. O listening MOS é então estimado. Estes parâmetros intermediários formam um subconjunto de entidades definidas no RTCP-XR (*extended report*) proposto pelo IETF. A qualidade é estimada através da aplicação de um segundo estágio do algoritmo P-VTQ, utilizando as informações do RTCP-XR [20].

#### 4 APRESENTAÇÃO DA METODOLOGIA

A metodologia de medição da qualidade de serviço proposta aqui consiste em avaliar a qualidade de um tráfego de voz simulado, através da simulação de congestionamento com características semelhantes às encontradas em uma rede real onde existe tráfego de voz e dados. Através dos arquivos de dados obtidos nesta rede simulada, são feitas medições objetivas de qualidade que possam resultar em uma avaliação próxima àquela obtida caso fossem utilizados métodos de medição subjetivos para avaliar a qualidade da voz em uma rede real.

Primeiramente, são gerados pacotes que simulam o tráfego de voz, este tráfego é transmitido através de um link, onde também trafegam pacotes que simulam o tráfego Web, de forma a ocasionar congestionamento que irá provocar variação do atraso e perda de pacotes no tráfego de voz.

Os arquivos de dados gerados na simulação, contêm as informações de tempo e perdas que serão analisadas de forma que possa ser feita a avaliação de qualidade. A partir das informações contidas no arquivo de dados, serão utilizados dois métodos para a análise objetiva de qualidade. O primeiro método é o E-model, um método computacional definido na recomendação G.107 do ITU [21]. O segundo método foi desenvolvido a partir de medidas de distorção do sinal. Os dois métodos serão comparados tendo como referência valores tabelados na escala MOS.

A Figura 12 mostra um diagrama da metodologia utilizada neste trabalho, onde um sinal de voz codificado é multiplicado por um vetor de erros formado por zeros e uns que correspondem respectivamente aos pacotes recebidos e perdidos. No primeiro teste foi utilizado um vetor de erros representando perdas uniformes gerado através do matlab. No segundo teste foi utilizado um vetor de erros gerado a partir de um arquivo de dados obtido na rede simulada. O sinal de voz degradado através dessa multiplicação é decodificado e convertido para o domínio da frequência através de uma FFT. Em seguida é feita a conversão para a escala de Bark sendo a mesma conversões também feita no sinal original. Os dois sinais são então comparados no domínio da frequência de forma a se obter uma medida da diferença entre os sinais que é finalmente convertida para a escala MOS.

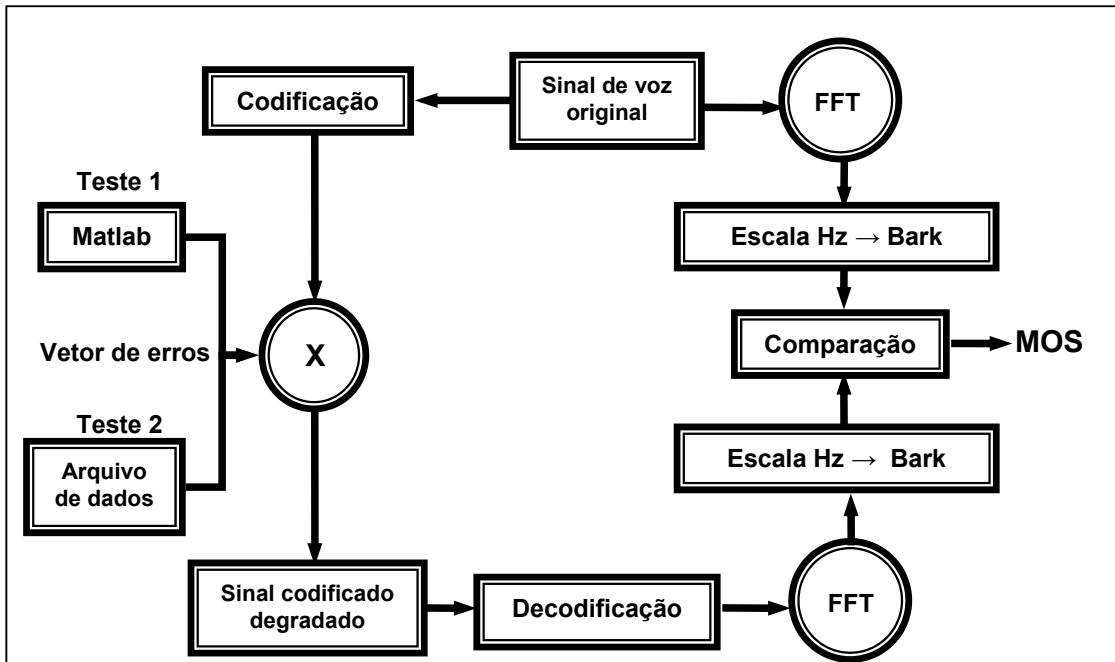


Figura 12 – Diagrama da metodologia

#### 4.10 NETWORK SIMULATOR

Para construir a rede simulada, foi utilizado o Network Simulator (NS) que é um simulador baseado nas linguagens C++ e OTcl. O simulador possui uma grande biblioteca de objetos de rede e protocolos, o que permite simular várias situações encontradas em redes reais.

O NS é um simulador de eventos discretos, onde cada evento possui um tempo para ocorrência. O evento é um objeto do C++ com um identificador único, um tempo pré agendado, e um ponteiro para um objeto que manipula este evento [31].

O interpretador OTcl é utilizado para executar scripts feitos pelo usuário, onde podemos definir uma topologia, protocolos específicos, aplicações que desejamos simular e a forma de saída que será obtida. A linguagem OTcl pode então utilizar os objetos compilados em C++ através de uma correspondência entre estes e seus próprios objetos. Desta forma o compilador C++ permite uma grande

eficiência e rapidez na execução, enquanto a utilização do interpretador OTcl permite uma sintaxe simples e flexível para o usuário. Os dados da simulação podem ser armazenados em um arquivo *trace* de onde podemos retirar as informações necessárias para nossa análise, desta forma podemos obter entre outras informações, os tempos de transmissão e recepção de cada pacote, número de pacotes perdidos e tipo de dado transmitido. Essas informações serão então, utilizadas para analisar o comportamento do tráfego simulado.

#### 4.2 A TOPOLOGIA DOS TESTES

Nos testes foi criada no NS uma topologia semelhante à apresentada em [32], desta forma foi possível observar eventos o mais próximo aos de uma simulação com equipamentos reais. Foram criados quatro nós simulando os PCs geradores de tráfego, e mais dois nós simulando os roteadores. No caso do simulador, não foi necessário criar os *switches*, pois os nós que representam os PCs foram ligados diretamente aos nós que representam o roteador, e o objeto de estudo é o link entre os dois roteadores, que representa o gargalo da rede. A Figura 13 mostra a topologia utilizada.

Na figura, os nós  $B_1$  a  $B_n$  são as fontes geradoras de tráfego “*background*”, o tráfego de dados que irá gerar congestionamento, cada fonte representa um usuário. O nó  $V_1$  representa a fonte de tráfego de voz,  $R_1$  e  $R_2$  são roteadores e  $D_1$  e  $D_2$  são, respectivamente, os destinos do tráfego de dados e voz. Os dados gerados na simulação são registrados em um arquivo de dados, de onde serão retiradas as informações para gerar o vetor de erros, que é um arquivo de registro dos pacotes de voz transmitidos e perdidos. Adicionalmente, também são retiradas do arquivo de dados, informações sobre o tempo de envio e chegada dos pacotes de voz, que servirão para o cálculo da variação do atraso.

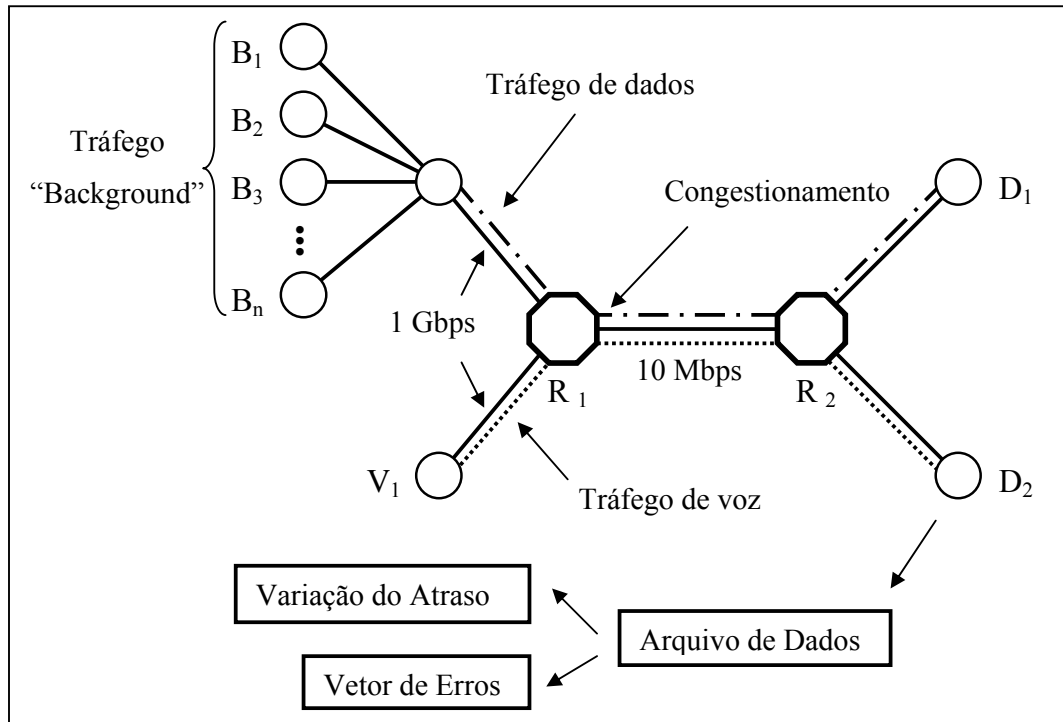


Figura 13 – Topologia da rede simulada

#### 4.30 TRÁFEGO DE VOZ SIMULADO

Para os testes estão sendo simulados o tráfego de voz utilizando os codecs G.711 e G729A [33]. No G.711 cada amostra de 8 bits é transmitida em um intervalo de 125  $\mu$ s (64 kbps). Para este codec, estamos simulando um tráfego de voz onde cada pacote IP/UDP/RTP carrega 80 amostras, ou 80 bytes, correspondentes à 10 ms de voz. Juntamente com os cabeçalhos IP (20 bytes), UDP (12 bytes) e RTP (8 bytes) no total são transmitidos pacotes de 120 bytes, a 96 kbps. No G.729 foram utilizados pacotes de 10 e 20 ms de voz, que correspondem respectivamente à quadros de 10 e 20 bytes, juntamente com os cabeçalhos IP/UDP/RTP temos um tamanho total de 50 bytes no primeiro caso, e 60 bytes no segundo. A Figura 14 mostra o formato dos pacotes simulados.

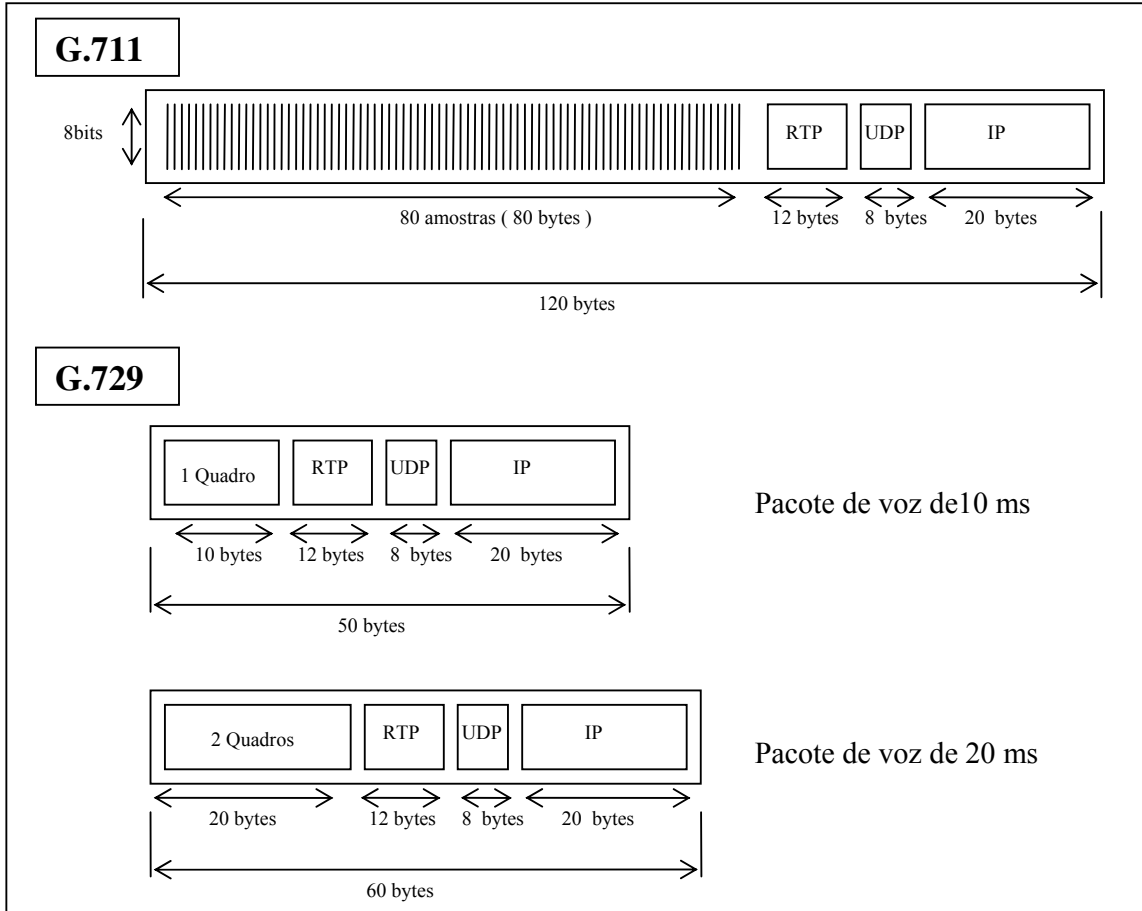


Figura 14 – Formato do pacote a ser simulado

No NS, o tráfego de voz é gerado por um nó, representando o PC que gera pacotes CBR/UDP. No caso do codec G.711, pacotes de 120 bytes a cada intervalo de 10 ms. Este tráfego é enviado para o nó que representa o roteador onde ocorrerá o congestionamento. A Figura 15 mostra o tráfego a ser simulado para os codec G.711 e G.729.

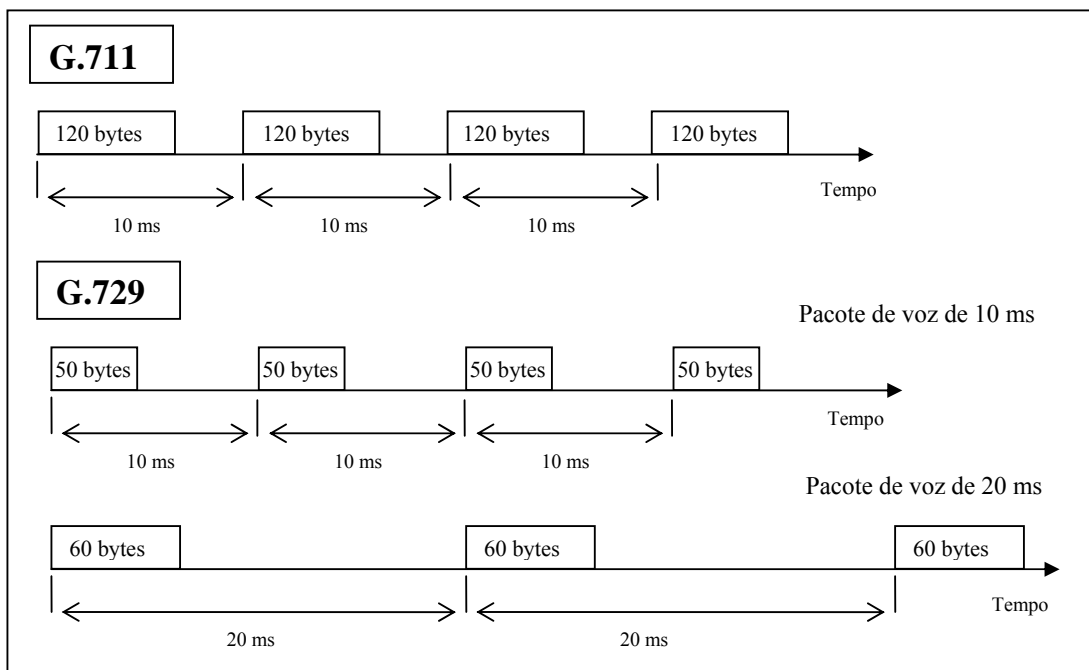


Figura 15 – Tráfego de voz simulado

#### 4.4 O TRÁFEGO DE “BACKGROUND”

Na simulação apresentada em [32] foi utilizado um gerador de tráfego baseado no modelo SURGE [34], que é um gerador de tráfego HTTP baseado no estudo das características de tráfego de redes *web* reais. Na geração do tráfego de background em nossa rede simulada, foi utilizado um modelo semelhante ao utilizado no modelo SURGE. Mas com a utilização de pacotes UDP, este tráfego simula a utilização de um arquivo de dados de um tráfego HTTP.

##### 4.4.1 CARACTERÍSTICAS DO TRÁFEGO HTTP

O tráfego gerado na simulação é baseado no tamanho dos arquivos, quantidade de arquivos por objeto *web* e intervalo de transmissão entre arquivos do tráfego HTTP. Estas características seguem os modelos de distribuição semelhantes aos apresentados em [34][35].

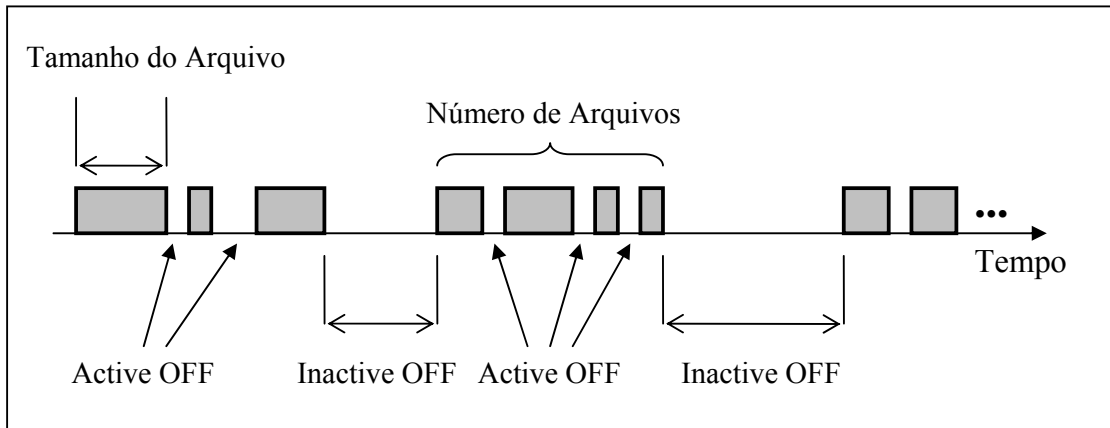


Figura 16 – Parâmetros do tráfego “Background”

A Figura 16 mostra o tráfego gerado por cada uma das fontes cujos parâmetros são detalhados a seguir:

- Tamanho dos arquivos – de acordo com [34] os arquivos transferidos seguem uma distribuição do tipo Pareto. O tamanho médio dos arquivos é de 10000 bytes. A função densidade de probabilidade desta distribuição é definida por:

$$p(x) = \alpha k^\alpha x^{-(\alpha+1)} \quad (7)$$

com parâmetros  $\alpha=1.0$  e  $k=1000$ ;

- Intervalos de transmissão – é o período em que não existe transmissão. Segundo [34] são definidos dois tipos de intervalos:
  - Active OFF Times – definido como o período de tempo entre a transmissão de componentes de um único objeto *web*. Estes intervalos seguem uma distribuição do tipo Weibull, cuja função densidade de probabilidade é definida por:

$$p(x) = \frac{bx^{b-1}e^{-\left(\frac{x}{a}\right)^b}}{a^b} \quad (8)$$

com parâmetros  $a=1,46$  e  $b=0.382$ , sendo o período de tempo médio igual a 1,4 segundos ;

- Inactive OFF Times – definido como o período de tempo entre a transmissão de objetos Web diferentes. Estes intervalos seguem uma distribuição do tipo Pareto, com parâmetros  $\alpha=1.5$  e  $k=1$ , sendo o tempo médio desse intervalo igual a 3 segundos..

- Número de arquivos por objeto Web – em [34] este número é definido como o número de *Embedded References*, que representa os arquivos que são transferidos dentro de cada objeto *web*. Este número segue uma distribuição do tipo Pareto, com  $k=1$  e  $\alpha=2.43$ , com média igual a 5,43 arquivos.

#### 4.4.2 GERAÇÃO DE TRÁFEGO WEB NO NS

O Network Simulator é capaz de gerar distribuições do tipo Pareto, entretanto, por problemas no algoritmo utilizado pelo simulador, não foi possível utilizar esta distribuição com parâmetro  $\alpha=1$ , como é o caso do tamanho dos arquivos do modelo SURGE, por isso para gerar esta distribuição no simulador foi utilizada a seguinte equação [36]:

$$x = \frac{k}{(1-y)^\alpha}, y \in [0,1] \quad (9)$$

onde  $y$  segue uma distribuição uniforme.

Para gerar os *Active OFF Times*, também foi utilizada uma equação para gerar uma distribuição Weibull, pois o NS não pode criar diretamente esta distribuição. Para gerar a distribuição Weibull foi utilizada a seguinte equação [36]:

$$x = a(-\ln(y))^{\frac{1}{b}}, y \in [0,1] \quad (10)$$

onde  $y$  segue uma distribuição uniforme.

Para o número de arquivos transmitidos por objeto web, foi utilizada a distribuição Pareto, onde neste caso não foram encontrados problemas na geração pelo NS. Como o número de arquivos é sempre um número inteiro, foi feita uma aproximação para o inteiro mais próximo.

#### 4.5 ANÁLISE DO ARQUIVO DE DADOS

O arquivo de dados gerado pelo NS contém informações sobre todos os eventos ocorridos na simulação. No nosso caso estamos apenas interessados na análise dos pacotes CBR do link entre os dois roteadores, onde ocorre o congestionamento e onde temos as informações sobre os pacotes descartados, e o tempo de partida e chegada dos pacotes.

##### 4.5.1 ANÁLISE DO ERRO

Os testes consistem basicamente em analisar a qualidade de voz de acordo com a ocorrência de perda de pacotes devido ao congestionamento. Foi utilizada em um primeiro teste uma distribuição randômica uniforme para as perdas, este teste serviu para comparar os resultados do método com os valores fornecidos pelo modelo E. No teste utilizando a rede simulada foi utilizado o modelo de Gilbert para a análise das perdas de pacotes, sendo estas representadas pelo ULP. A Figura 17 mostra o cálculo das perdas para cada sinal de voz.

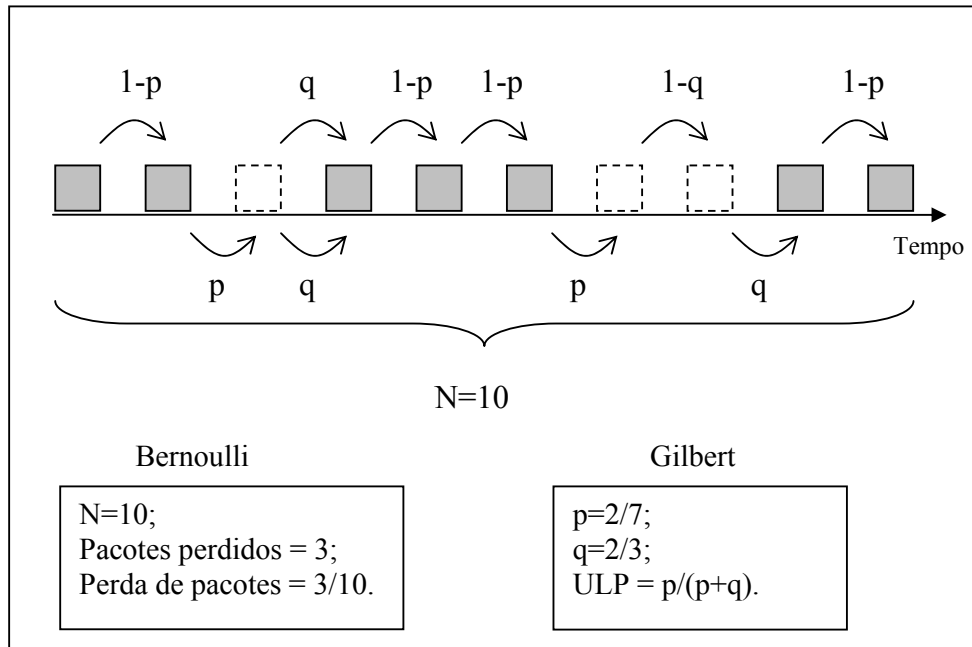


Figura 17 –Cálculo da perda de pacotes

#### 4.5.2 ANÁLISE DA VARIAÇÃO DO ATRASO

Através da análise dos tempos de transmissão e recepção de cada pacote, é possível fazer um cálculo da variação do atraso de acordo com a Equação 3. Neste trabalho não estão sendo consideradas nas medições de qualidade, as degradações ocasionadas pela variação do atraso, mas foram realizadas medidas desta variação através dos dados fornecidos pela simulação. Estas degradações podem ser interpretadas como um delay adicional, ou em perda de pacotes quando utilizamos um *jitter buffer*.

#### 4.5.3 ANÁLISE ATRAVÉS DO E-MODEL

No nosso caso, como os valores  $R_0$  e  $I_s$  presentes na equação para obtenção do fator  $R$ , têm relação apenas com o sinal codificado não dependendo da rede de transporte, podemos então utilizar valores *default*, essa equação pode ser simplificada para [21]:

$$R = 94,2 - I_e \quad (11)$$

Nesta equação simplificada, o valor 94,2 representa o fator  $R$  do codec G.711 na ausência de degradação ( $R_0 + I_s$ ) e como não estamos considerando perdas devido ao atraso, o fator  $I_d$  da equação original foi suprimido.

O fator  $R$  pode ser transformado para a escala MOS através da seguinte equação:

$$\begin{cases} MOS = 1 & R < 0 \\ MOS = 1 + 0.035R + R(R - 60)(100 - R)7 \cdot 10^{-6} & 0 < R < 100 \\ MOS = 4,5 & R > 100 \end{cases} \quad (12)$$

#### 4.5.4 AVALIAÇÃO ATRAVÉS DA MEDIDA DE DISTORÇÃO

A medida da distorção é feita através da comparação entre um sinal de referência e o mesmo sinal já degradado. A degradação é obtida através de um vetor de erro, gerado na simulação de uma rede congestionada no NS, este vetor de erro é formado por zeros e uns, onde os zeros correspondem aos pacotes perdidos no congestionamento. A Figura 18 figura mostra o método utilizado.

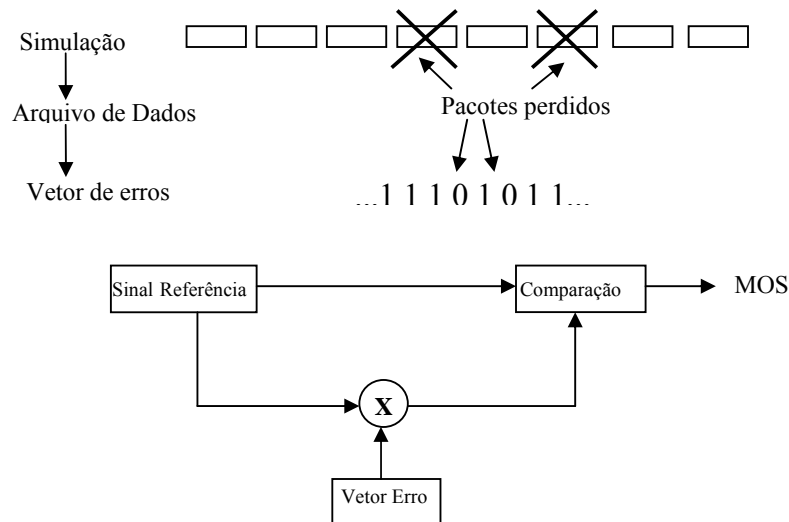


Figura 18 – Obtenção do vetor de erro e medição

Este método é o mesmo utilizado em [32], mas no nosso caso utilizamos o arquivo de dados gerado pelo NS. A forma como são obtidas os diversos sinais degradados é mostrada na Figura 19. Nesta figura podemos ver um sinal de voz a ser medido que corresponde a um tamanho de  $n$  pacotes e o vetor obtido a partir da simulação, que possui tamanho  $N$ , os sinais de voz degradados são obtidos “deslizando-se” o sinal de voz original ao longo do vetor, os pacotes perdidos, representados na cor preta, são deletados da amostra de voz de acordo com a posição relativa do pacote perdido. Obtemos desta forma, o número de  $k$  sinais de voz degradados a serem medidos.

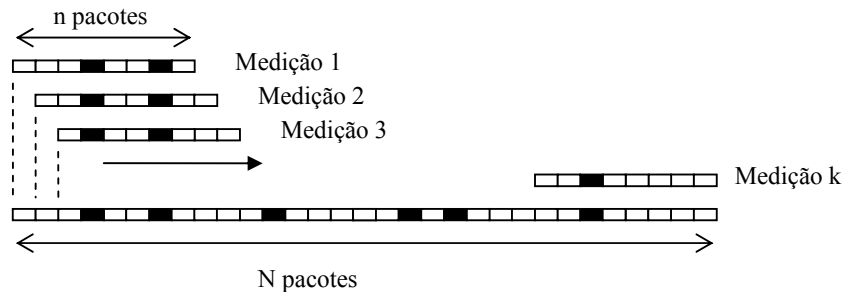


Figura 19 – Obtenção dos sinais a serem medidos

O sinal original e o sinal degradado são transformados para o domínio da frequência através do cálculo da FFT, utilizando uma janela Hanning de 32 ms, com 50% de *overlap*, no caso do sinal amostrado em 8 kHz esta janela corresponde a 256 amostras, conforme descrito em [24]. É calculada então, a potência do sinal.

O sinal no domínio da frequência em Hz é convertido para uma escala de frequência Bark, que de acordo com o conceito das bandas críticas, melhor representa a sensação auditiva do ouvido humano. Segundo este conceito o sistema auditivo processa os sons em subbandas chamadas bandas críticas, a largura de cada banda difere de acordo com a frequência, sendo maior nas faixas de frequência mais altas, cada banda tem a largura de um Bark. A conversão da escala em Hertz para Barks é feita através da equação [25]:

$$z = 13 \cdot \arctan(0,76 \cdot f) + 3,5 \arctan\left(\frac{f}{7,5}\right)^2 \quad (13)$$

onde  $z$  é a taxa da banda crítica em Barks, e  $f$  é a frequência em Hertz.

A largura de cada banda é dada pela equação [25]:

$$\Delta f_G = 25 + 75 \cdot (1 + 1,4 f^2)^{0,69} \quad (14)$$

onde  $f$  é a frequência e  $\Delta f_G$  é a largura da banda, ambas em Hertz.

A Figura 20 mostra a curva que relaciona as escalas em Hertz e Bark.

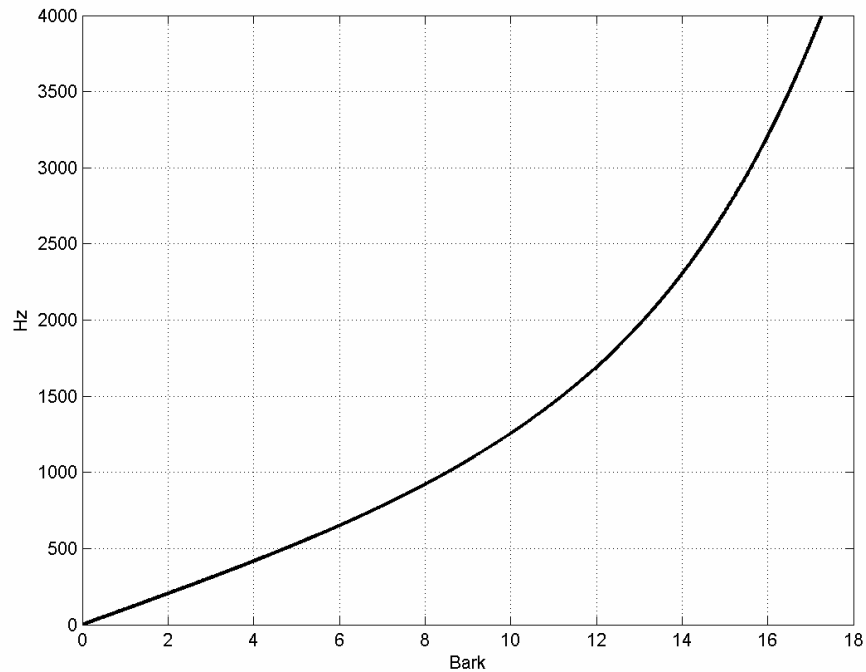


Figura 20 - Correspondência entre as escalas de frequência em Hertz e Bark

É feita então, uma subtração entre a potência dos dois sinais no domínio da frequência para cada uma das bandas, como mostra a equação:

$$D(i) = \sum_{k=1}^n [S_x^{(k)}(i) - S_y^{(k)}(i)]^2 \quad (15)$$

onde  $S_x^{(k)}(i)$  é o espectro do frame  $k$  do sinal de voz original na banda crítica  $i$  e  $S_y^{(k)}(i)$  é o espectro do frame  $k$  do sinal de voz degradado,  $D(i)$  é o somatório das diferenças para a banda crítica  $i$ .

A equação 16 mostra o somatório dessas subtrações que representa a diferença  $D$  entre os sinais, esta diferença será convertida para a escala MOS.

$$D = \sum_{i=1}^n D(i) \quad (16)$$

Por simplificação, foram computados apenas as bandas críticas de 5 a 8 pois foram as que apresentaram maior linearidade na curva  $D \times perdas$ .

Como sinal original foi utilizado um sinal de referência da base de dados TIMIT disponível em [37], com duração de 2 segundos e amostrado a 8000 Hz e 16-bit no formato PCM linear. Este sinal foi convertido para amostras codificadas em G.711 e G729. Para simular a degradação em vários níveis, este sinal foi multiplicado por um vetor contendo zeros e uns, onde os zeros correspondem aos erros. Como o objetivo é simular perdas de pacotes, cada zero nesse vetor é multiplicado por um número de amostras correspondentes ao quadro, no nosso caso um quadro de 10 ms, ou 80 amostras. O sinal resultante é um sinal degradado que será comparado com o sinal original através do método. Os resultados desse teste para o codec G.711 são mostrados na Figura 21.

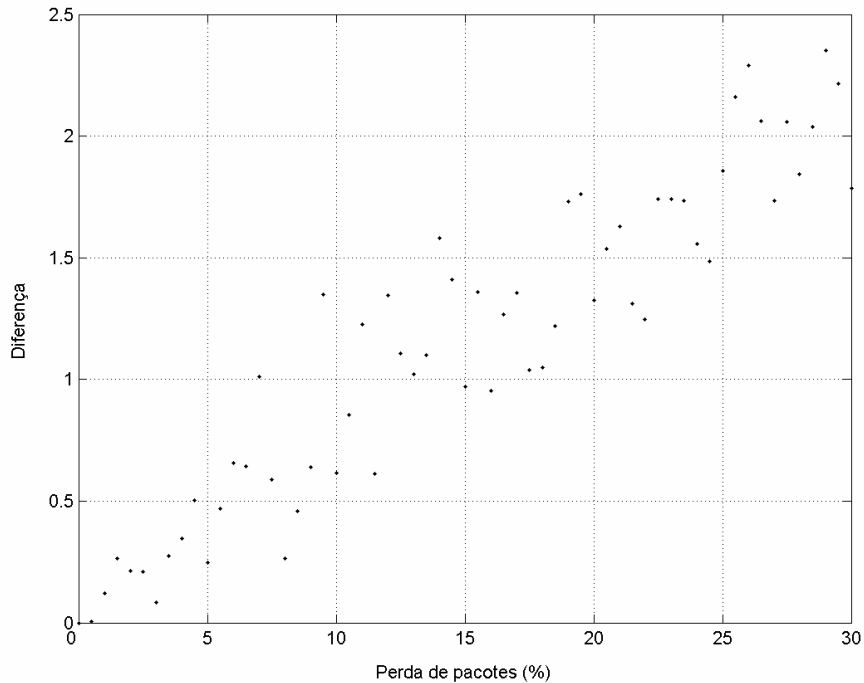


Figura 21 – Valores das diferenças para o codec G.711

Na figura, os pontos mostram valores normalizados da média do valor absoluto das diferenças ( $D$ ) entre os dois sinais, obtida a partir de 10 medidas para cada valor de erro, de forma a diminuir o efeito da posição do quadro errado no sinal de áudio. A linha mostra uma aproximação obtida através do MATLAB.

De acordo com [38][39], existe uma relação entre o valor do fator  $l_e$  utilizado no E-model, e a taxa de perda de pacotes, esta relação segue a seguinte equação:

$$I_e = a \ln(1 + b\rho) + c \quad (17)$$

onde  $\rho$  é o percentual da taxa de perda de pacotes e os parâmetros  $a, b$  e  $c$  são constantes que dependem apenas do codec utilizado.

Para o codec G.711, os valores das constantes  $a, b$  e  $c$  foram obtidos a partir da curva  $l_e \times \rho$  para o E-model. Os valores de  $l_e$  por sua vez, foram obtidos a partir da curva MOS x perdas segundo o E-model em [40]. Essas curvas são mostradas na Figura 22.

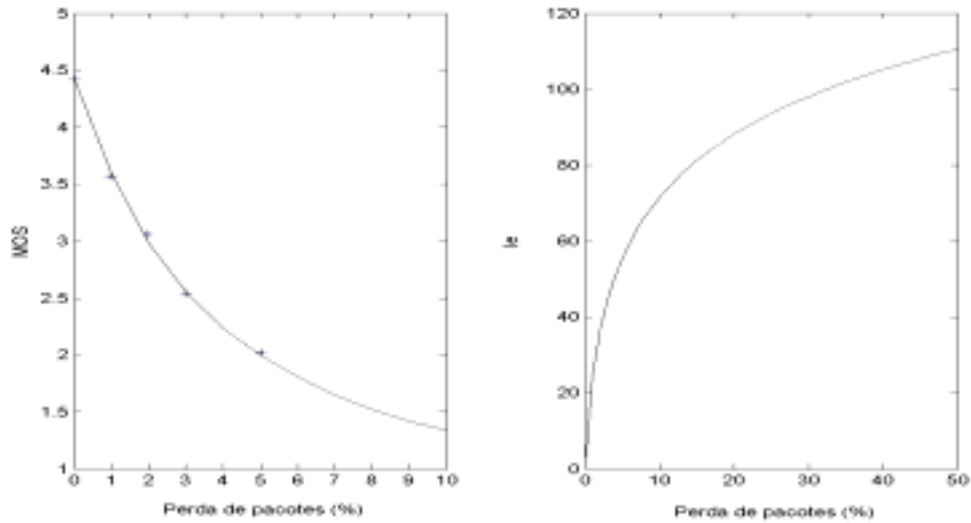


Figura 22 – Curvas de MOS e le para o codec G.711

Através destas curvas foram obtidos os valores de  $a=25$  e  $b=165$ . Para o codec G.711 o valor de  $c=0$ , ou seja, quando não existem perdas o valor de  $le$  também é 0.

Tendo em mãos a relação entre os valores de erro e diferença entre sinais, e a relação entre o MOS e o erro, foi obtido um gráfico da relação entre o MOS e o valor das diferenças entre sinais. A partir deste gráfico e através uma ferramenta de aproximação de curvas, foi obtida a seguinte equação que representa esta relação para o codec G.711:

$$MOS = 3,42 \cdot e^{-3,207 \cdot D} + 1 \quad (18)$$

Para se obter a equação na ferramenta de aproximação de curvas do MATLAB foi utilizado o algoritmo região de confiança (*trust-region*) que é o método não-linear padrão da ferramenta.

A Figura 23 mostra os valores das diferenças ( $D$ ) para o codec G.729, as constantes  $a$ ,  $b$  e  $c$  foram obtidas diretamente a partir de [39], e foi utilizado o mesmo processo utilizado no codec G.711 para obter a partir das curvas  $MOS \times perdas$  e  $D \times perdas$ , resultando na equação a seguir:

$$MOS = 0,95 \cdot e^{-2,626 \cdot D} + e^{-308,7 \cdot D} - 0,0473 \cdot D + 1,95 \quad (19)$$

Com base nessas equações, podemos obter os valores de MOS a partir da comparação entre os sinais.

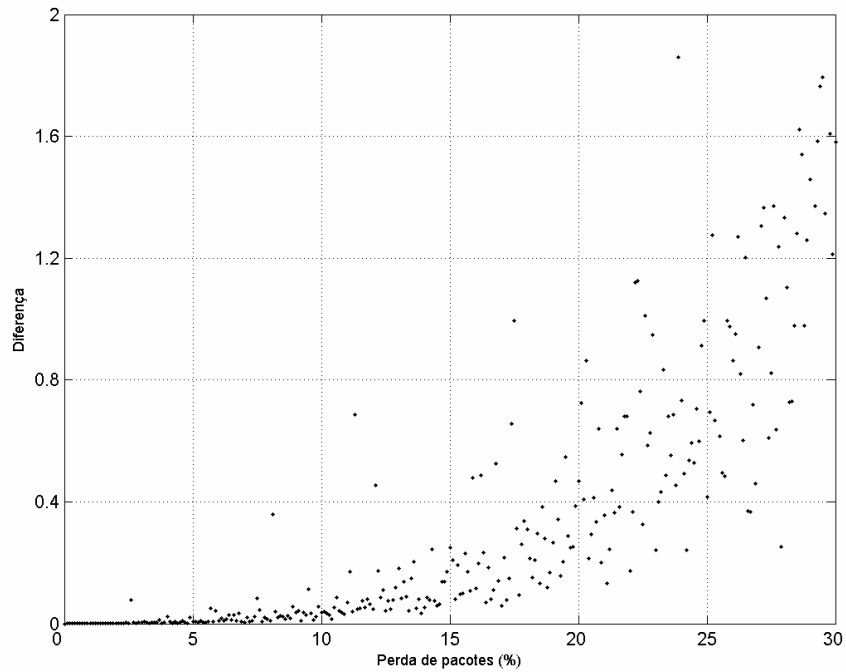


Figura 23 - Valores das diferenças para o codec G.729

## 5 RESULTADOS

Foram realizados dois testes utilizando o método apresentado, no primeiro foram obtidas as equações para a conversão da medida da distorção para o MOS através do E-model e no segundo o método foi utilizado para medir a qualidade de voz na rede simulada, os resultados dos dois testes são apresentados a seguir. O programa que foi escrito para o software MATLAB encontra-se no Anexo I.

### 5.1 TESTE 1

O primeiro teste consistiu em avaliar a capacidade do método em medir a qualidade da voz. A Figura 24 mostra o gráfico obtido para valores de perda de pacotes uniforme, variando de 0% a 30% para os dois codecs e a comparação com os valores obtidos a partir do E-model.

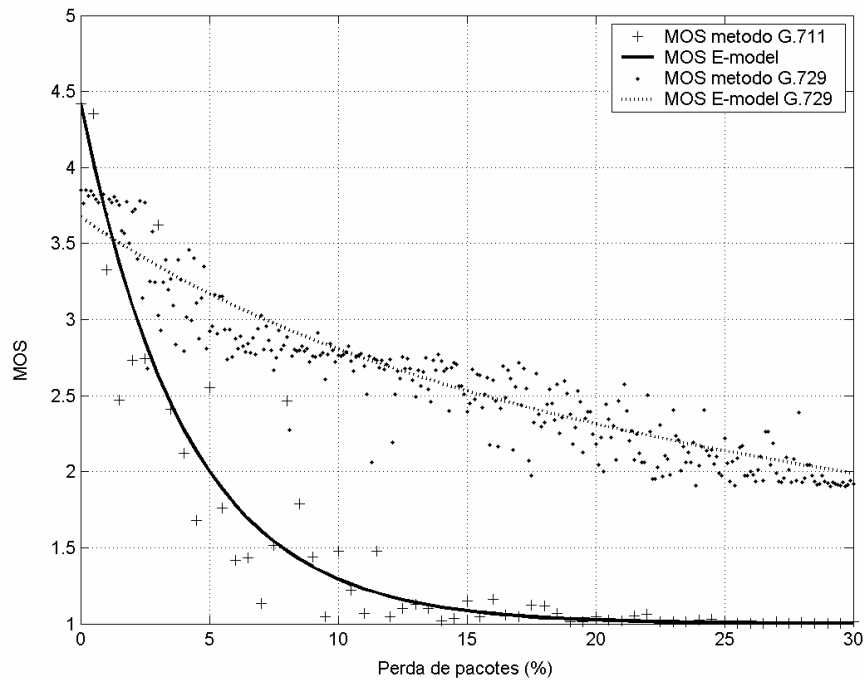


Figura 24 – Comparação entre o E-model e o método

Nessa figura podemos observar que o valor de MOS do método, calculado através da medição das diferenças entre os sinais segue aproximadamente a curva obtida através do E-model. Para o codec G.711, onde não foi utilizada nenhuma forma de compensação de perdas de pacotes, vemos uma queda acentuada no valor do MOS de acordo com o aumento da perda. Já para o codec G.729, que apresenta um valor de MOS inicial ligeiramente menor que o G.711, essa diminuição do MOS com o aumento das perdas é menor, pois o codec possui internamente uma forma de compensar as perdas. Para um mesmo valor de erro os valores de MOS obtidos a partir do método podem variar, o que não ocorre no E-model, já que o mesmo não leva em consideração a posição relativa dos pacotes perdidos.

## 5.2 TESTE 2

O segundo teste foi a avaliação da qualidade do tráfego de voz utilizando a rede simulada. O programa desenvolvido para o NS encontra-se no Anexo II. Após a obtenção do vetor gerado a partir do arquivo de dados, o sinal de voz original foi então multiplicado por este vetor de forma a se obter o sinal degradado. A simulação foi feita para pacotes carregando 10 e 20 ms de voz, e foi utilizado o mesmo sinal de referência utilizado no teste anterior. Para este teste foi utilizado o codec G.729.

O teste foi realizado com a geração de tráfego de background simulando 150, 200 e 250 usuários. Como estamos simulando o tráfego de uma rede real, foi utilizado o modelo Gilbert para a análise da perda de pacotes, devido à ocorrência de erros em rajadas. Foram feitas 1000 medições ao longo de cada vetor de erro, os resultados obtidos são mostrados a seguir:

### 5.2.1 PERDA DE PACOTES

A Figura 25 mostra a curva aproximada da função densidade de probabilidade da perda de pacotes, representada por  $f(x)$ . O primeiro gráfico mostra a distribuição obtida na simulação de pacotes de voz de 10 ms. O segundo gráfico mostra o resultado para simulação de pacotes de voz de 20 ms.

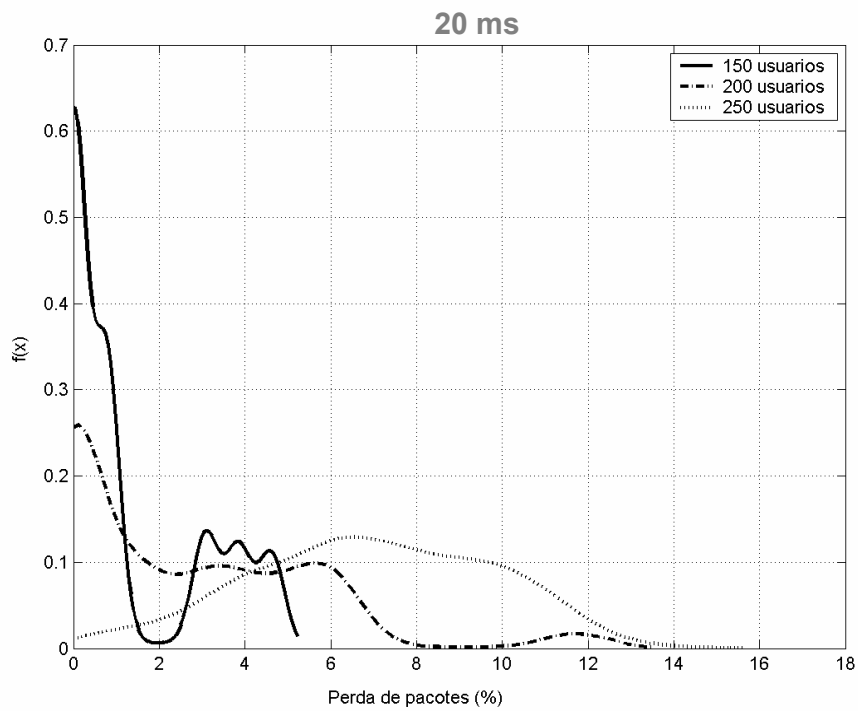
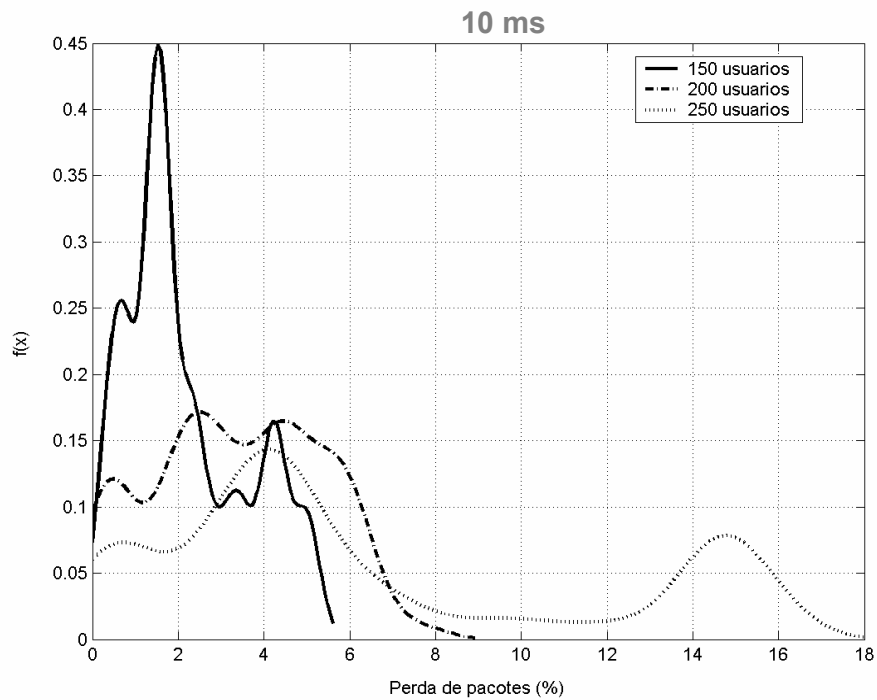


Figura 25 – Densidade de probabilidade da perda de pacotes

Nesses gráficos pode ser observado o aumento da perda de pacotes de acordo com o aumento de tráfego, e observamos a ocorrência relativamente menor

no caso da simulação de pacotes de 20 ms, pois nesse caso a quantidade de pacotes transmitidos é menor, o que diminui a probabilidade de perdas.

### 5.2.2 VARIAÇÃO DO ATRASO

A Figura 26 mostra a função densidade de probabilidade do *jitter* representada por  $f(x)$ , o valor do *jitter* foi calculado de acordo com a Equação 3. O primeiro gráfico mostra o resultado para a simulação de pacotes de voz de 10 ms, e o segundo para 20 ms.

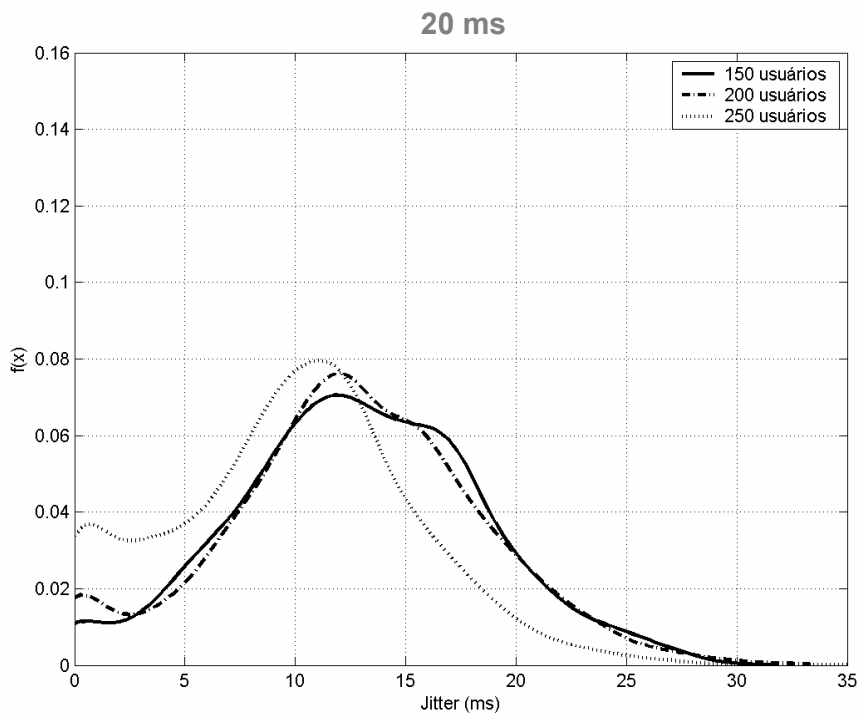
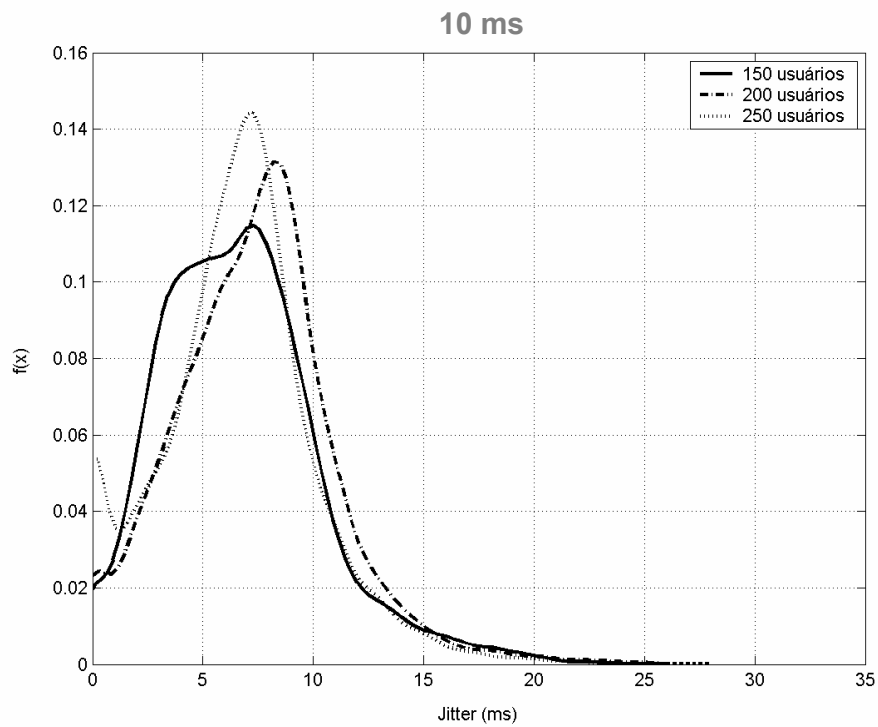


Figura 26 – Densidade de probabilidade da variação do atraso

Nestes gráficos também podemos observar a influência maior do aumento do congestionamento sobre o tráfego de pacotes de 10 ms, para o tráfego de pacotes 20 ms, o aumento do congestionamento não se refletiu diretamente no valor da variação do atraso medida.

### 5.2.3 MEDIÇÃO DA QUALIDADE

Os resultados dos testes de estimação do MOS utilizando a metodologia são mostrados nas figuras a seguir: Também nestes testes, devido à localização relativa do erro, podemos observar a medição de diferentes valores de MOS para um mesmo valor de perda de pacotes.

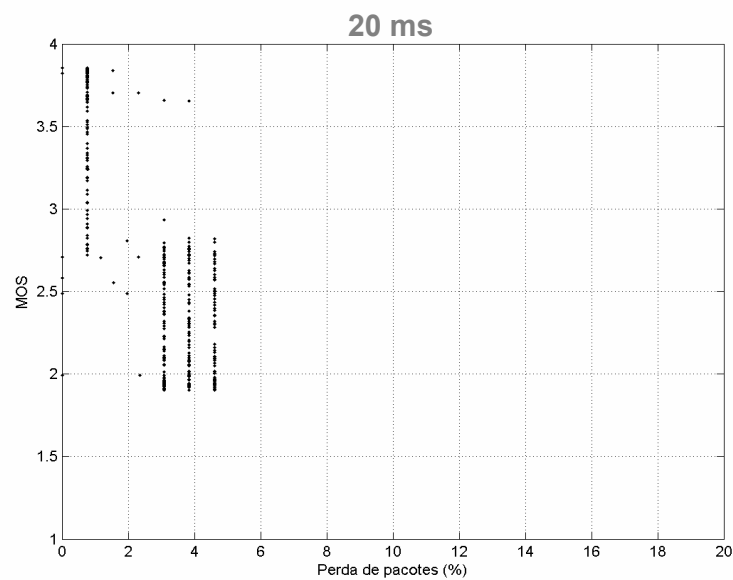
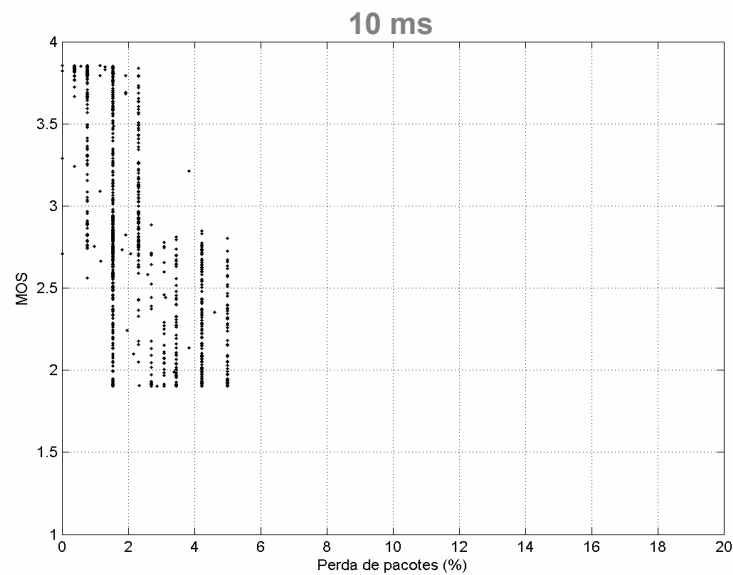


Figura 27 – MOS para 150 usuários

Para o tráfego *background* simulado de 150 usuários, observamos que o valor de MOS calculado varia bastante, mesmo para pacotes de voz de 20 ms, onde o valor das perdas é relativamente menor. Isso ocorre porque os pacotes maiores carregam uma quantidade maior de informação, e no caso de perda vai representar um dano maior à qualidade.

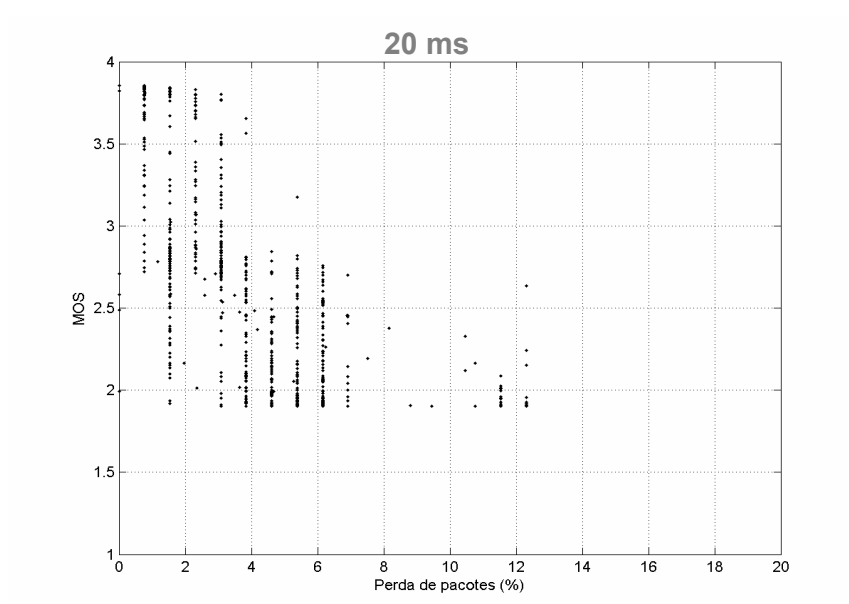
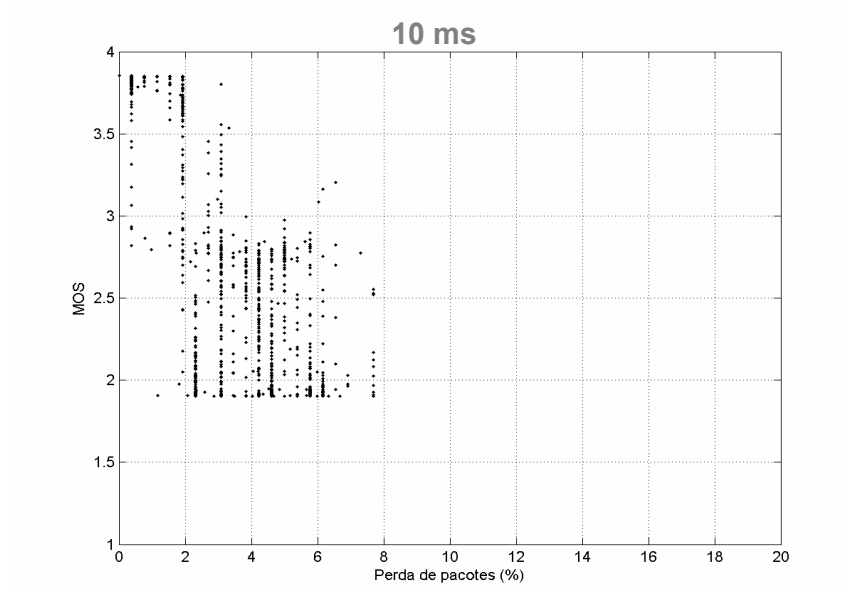


Figura 28 – MOS para 200 usuários

No caso da simulação de 200 e 250 usuários, podemos observar um comportamento semelhante ao observado no teste anterior, nesse caso com uma degradação maior.

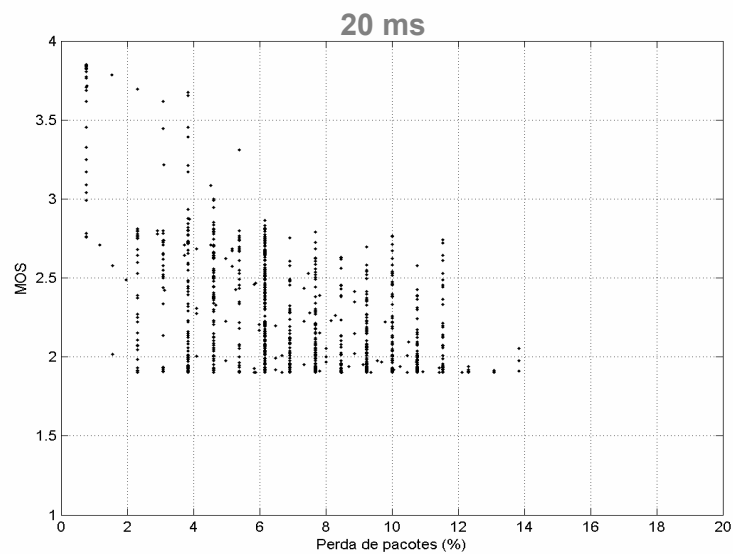
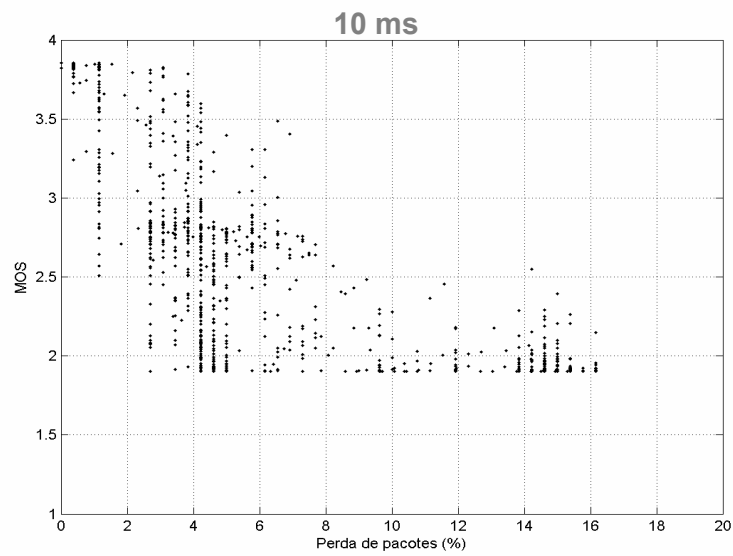


Figura 29 – MOS para 250 usuários

Observamos que tanto para pacotes com 10 ms quanto para pacotes com 20 ms, há uma grande ocorrência de medições onde a qualidade está abaixo de um nível de qualidade considerado aceitável, isto justifica a utilização de métodos de controle de QoS.

A Figura 30 mostra um gráfico do MOS e perdas médias para cada um dos casos testados acima, nele podemos ver que exceto no caso de 250 usuários, as perdas e o MOS medido foi maior para os pacotes de 10 ms. No gráfico também temos a curva do e-model para perdas uniformes, e podemos observar que a curva apresenta valores sempre superiores à média para perdas em rajadas, medidas pelo método.

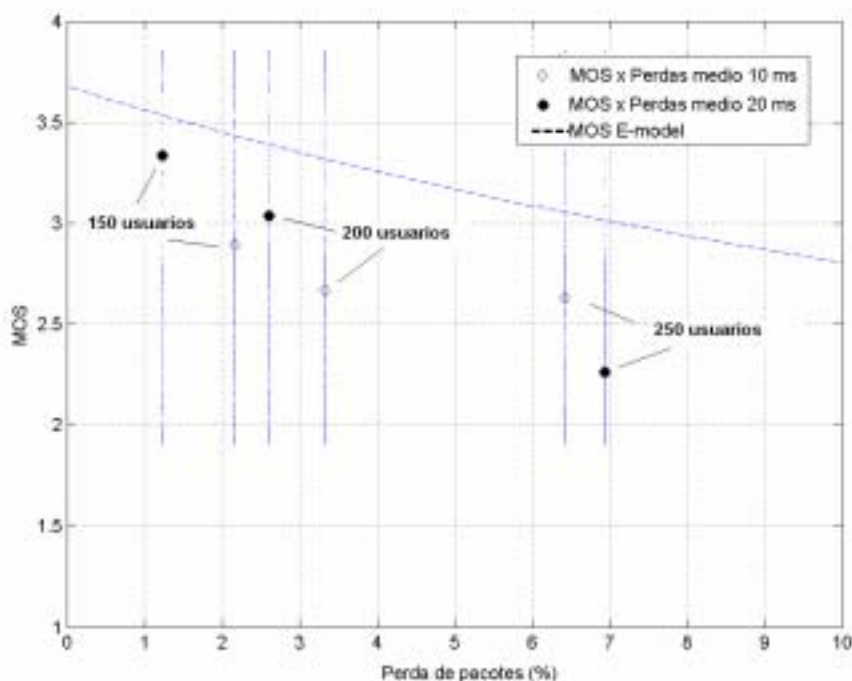


Figura 30 – MOS e perdas médios x E-model

Como podemos observar através dos testes, a qualidade da voz em uma rede onde não foi utilizada nenhuma forma de tratamento de QoS pode variar bastante na presença de congestionamento provocado por outras fontes de tráfego. Na prática, o controle da QoS é feito atribuindo-se uma prioridade maior aos pacotes de voz, através da utilização de uma fila específica, desta forma a probabilidade de perda devido ao congestionamento torna-se menor. Através do Network Simulator, podem ser reproduzidas redes utilizando várias técnicas tais como: DiffServ, filas de prioridade e MPLS. Desta forma podemos mensurar o impacto causado no tráfego de voz em várias condições de congestionamento através do método proposto.

## 6 CONCLUSÃO

Neste trabalho foi apresentada uma metodologia de medição da qualidade de voz através da utilização de uma rede simulada no Network Simulator e da utilização dos os *traces* gerados por este simulador para gerar um sinal de voz com características semelhantes às de um sinal que foi degenerado devido ao congestionamento ocorrido em uma rede real. O método de medição de qualidade leva em consideração as diferenças no domínio da frequência entre o sinal original e o sinal degenerado.

O método foi utilizado para avaliar a qualidade de um sinal de voz que foi codificado de acordo com as recomendações G.711 e G.729 e submetida à degradação. Os resultados foram comparados com valores fornecidos pelo modelo E. Foi observada nos resultados fornecidos pelo método, a influência da posição relativa dos pacotes perdidos na qualidade do sinal. Também foi verificada a qualidade do sinal de voz quando submetido a perdas em rajadas, característica de redes IP. Foi observado neste caso a maior degradação provocada por este tipo de perda. Não foram realizados testes para verificar a correlação dos resultados obtidos pela metodologia com outros métodos de medição objetivos, o que serviria para avaliar a precisão dos valores obtidos.

Utilizando uma rede simulada, o método apresentado tem a vantagem de ser flexível e barato pois além de utilizar um *software* distribuído gratuitamente podem ser simuladas diversas topologias apenas com alteração do programa sem a necessidade de mudanças em equipamentos. Por ser um *software* amplamente utilizado em ambientes de pesquisa o Network Simulator é continuamente atualizado de forma que podem ser simuladas a maior parte das novas tecnologias utilizadas em redes IP.

Portanto, a contribuição do trabalho é oferecer uma alternativa para o estudo da qualidade de voz quando não há uma rede ou softwares de medição disponíveis para testes.

Trabalhos futuros podem mostrar uma melhora no método incorporando um modelo perceptual à medição de forma à apresentar uma melhor correspondência

com a sensibilidade auditiva de acordo com os métodos utilizados em redes reais. Também pode ser feita uma comparação com outros métodos perceptuais e a simulação com outros *codecs* e características de tráfego.

Na simulação poderão ser utilizadas redes com outras técnicas de QoS para que seja avaliada a eficiência das mesmas.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] J. Postel, RFC-791- Internet Protocol (IP), Internet Engineering Task Force. 1981. Disponível em: <http://www.ietf.org/rfc/rfc0791.txt>
- [2] Daniel Collins, "Carrier Grade Voice Over IP," McGraw-Hill, 2003.
- [3] J. Postel, RFC-793-Transmission Control Protocol, Internet Engineering Task Force. 1981. Disponível em: <http://www.ietf.org/rfc/rfc0793.txt>.
- [4] J. Postel, RFC-768- User Datagram Protocol, Internet Engineering Task Force. 1981. Disponível em: <http://www.ietf.org/rfc/rfc0768.txt>.
- [5] Soares, Luiz Fernando G., Redes de Computadores: das LAN's, MAN's e WAN's às redes ATM, Campus, 1995.
- [6] Schulzrinne, H, Casner, S., Frederick, R. and V. Jacobson, RFC-1889- RTP: A Transport Protocol for Real-Time Applications, Internet Engineering Task Force. 1981. Disponível em: <http://www.ietf.org/rfc/rfc1889.txt>
- [7] Olivier Hersent, Jean-Pierre Petit and David Gurle, "Beyond VOIP Protocols: Understanding Voice Technology and Networking Techniques for IP Telephony," John Wiley & Sons Ltd., West Sussex, 2005.
- [8] Wenyu Jiang and Henning Schulzrinne, "Comparison and Optimization of Packet Loss Repair Methods on VoIP Perceived Quality under Bursty Loss," *NOSSDAV'02*, Miami Beach, Florida, May 2002.
- [9] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, June 1994.
- [10] A. Wang, Zheng, Internet QoS: Architectures and mechanisms for Quality of Service, Lucent Technologies, 2001.
- [11] S. Black et al., "An Architecture for Differentiated Services," RFC2475, Dec. 1998.
- [12] Davie, B., Charny, A., Bennet, J.C.R., Benson, K., Le Boudec, J.Y., Courtney, W., Davari, S., Firoiu, V. and D. Stiliadis, RFC-3246 - An Expedited Forwarding PHB (Per-Hop Behavior), Internet Engineering Task Force. 2002. Disponível em: <http://www.ietf.org/rfc/rfc3246.txt>.
- [13] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski, RFC-2597 - Assured forwarding goup, Internet Engineering Task Force. 1999. Disponível em: <http://www.ietf.org/rfc/rfc2597.txt>.
- [14] Haykin, Simon, Communication Systems, John Wiley & Sons, Inc.,1994.
- [15] International Telecommunications Union, "Pulse Code Modulation (PCM) of voice frequencies," *ITU-T Recommendation G.711*, 1988.
- [16] International Telecommunications Union, "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM)," *ITU-T Recommendation G.726*,1990.
- [17] International Telecommunications Union, "Coding of speech at 16 kbits/s using low-delay code excited linear prediction," *ITU-T Recommendation G.728*, 1992.
- [18] International Telecommunications Union, "Dual rate speech coder for multimedia communications transmitting at 5.3 and 6.3 kbit/s," *ITU-T Recommendation G.723*, 1996.
- [19] International Telecommunications Union, "Coding of speech at 8 kbit/s using conjugate structure algebraic-code-excited linear prediction (CS-ACELP)," *ITU-T Recommendation G.729*, 1996.
- [20] A. Takahashi and Hideaki Yoshino, "Perceptual QoS assessment technologies for VoIP", *IEEE Communications Magazine*, July 2004.
- [21] International Telecommunications Union, "The E-Model, a computational model for use in transmission planning," *ITU-T Recommendation G.107* July 2000.

- [22] C. Boutremans, G. Lannaccone and C. Diot, "Impact of link failures on VoIP performance," *NOSSDAV'02*, Miami Beach, Florida, May 2002.
- [23] William C. Hardy, "VoIP Service Quality – Measuring and Evaluating Packet-Switched Voice," McGraw-Hill, 2003.
- [24] International Telecommunications Union, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *ITU-T Recommendation P.862*, 2001.
- [25] W. Yang, *Enhanced Modified Bark Spectral Distortion (EMBSD): An Objective Speech Quality Measure Based on Audible Distortion and Cognition Model*, Temple University, 1999.
- [26] International Telecommunications Union, "Objective quality measurement of telephone band (300–3400 Hz) speech codecs," *ITU-T Recommendation P.861*, 1996.
- [27] Renshou Dai, "A Technical White Paper on Sage's PSQM Test," Sage Instruments white paper, August, 2000.
- [28] Spirent Communications - Zarak Systems Division, "Using PSQM to Test Packetized Speech," Spirent Communications, March, 2000.
- [29] European Telecommunications Standards Institute, "ETSI Guide EG 201 377-1, 2002.
- [30] J.G. Beerends, A.P. Hekstra, A.W Rix and M. P. Hollier, "Perceptual Evaluation of Speech Quality (PESQ), the new ITU standard for end-to-end speech quality assessment. Part II – Psychoacoustic model," For publication in the *J.Audio Eng. Soc.*, October, 1998.
- [31] E. Altman e T. Jiménez, *NS Simulator for beginners*, Univ. de Los Andes e ESSI, 2003-2004.
- [32] F. Hammer, P.Reichl e T. Ziegler, *Where Packet Traces Meet Speech Samples: An Instrumental Approach to Perceptual QoS Evaluation of VoIP.*, Telecommunications Research Center (ftw.), Vienna.
- [33] Voice Age Corporation, *Open G.729 Initiative -Technical Documentation Version 1.0 – Revision 2004-07-15*, VoiceAge Corporation, Quebec, CA, 2004.
- [34] P. Barford e M. Crovella, *Generating Representative Web Workloads for Network and Server Performance Evaluation*, Computer Science Department, Boston University, Boston.
- [35] J. Cao et al., *Stochastic Models for Generating Synthetic HTTP Source Traffic*, IEEE Infocom, 2004.
- [36] P. Clifford, "Simulating Random Variables," Jesus College, Oxford. Disponível em: <http://www.jesus.ox.ac.uk/~clifford/a5/chap1/node5.html>.
- [37] TIMIT corpus. Disponível na internet em [http://www.ee.columbia.edu/~dpwe/sounds/sents/sf1\\_cln.wav](http://www.ee.columbia.edu/~dpwe/sounds/sents/sf1_cln.wav) arquivo consultado em dezembro/2005
- [38] Lijing Ding e Rafik A. Goubran, *Speech quality Prediction in VoIP Using the Extended E-Model*, Dep. Of Systems and Computer Engineering, Carleton University, Ottawa.
- [39] L. Sun e E. Ifeachor, *New Models for Perceived Voice Quality Prediction and their Applications in Playout Buffer Optimization for VoIP Networks*, University of Plymouth, Plymouth, UK, 2004.
- [40] A. Markopoulou, F Tobagi e M. Karam, *Assessing the Quality of Voice Communications Over Internet Backbones*, IEEE/ACM Transactions on Networking, vol.11, NO.5, October 2003.

# ANEXO I

## Programas do MATLAB para cálculo do MOS

### Programa utilizado no Teste 1

```

%% Este programa calcula um valor medio das diferenças
%%entre um sinal
%% codificado em G.729 e o mesmo sinal degradado em
%% varios níveis de erro
%
%%Leitura do sinal codificado
fid = fopen('bit729');
[x,a2] = fread(fid,'ubit1');
fclose(fid);
d=0;
x=x';
media=0;

for n=1:10 %Numero de repetições

%Gera matriz com vetores contendo 0's e 1's de acordo
com o percentual de
%erro
a1=0;
a=0.3:0.001:0.5; % Variação do erro
b=1-a;
cc=ceil(a2/80);
[d1,d2]=size(a);
sum(1:d2)=0;

for i=1:d2
ve(:,i) = randerr(cc,1,[0 1; a(i) b(i)]); %Gera matriz com
varios vetores
end

%Multiplica o sinal original pelos varios vetores da matriz
de erro
for i=1:d2
kk=1;
for j=1:cc
for k=1:80 %Tamanho do quadro=80
if (kk <= a2)
xe(kk,i)=x(kk)*ve(j,i);
kk=kk+1;
end
end
end
end

% Converte os sinais codificados em G.729 em sinais de
audio
for i = 1:d2
fid = fopen(['P' int2str(i)], 'w');
fwrite(fid,xe(:,i),'ubit1');
fclose(fid);
end

for u=1:d2
S = ['copy P' int2str(u) ' PX'];
[status,result] = dos(S);
result;
[status,result] = dos('conv');

[xa,Fs,bits]=wavread('sampleorig');
[xb,Fs,bits]=wavread('sampletest');
[a3,a4]=size(xb);

% Calcula um valor medio das diferenças entre um sinal
% de audio e o mesmo sinal degradado em varios níveis
de erro

%Calculo da diferença entre os sinais
%Filtering
N=floor(a3/128)-1;
w=hann(256);
t = 0:0.000125:0.6;
w=w';
c=0;
c1=0;

for i = 1:N
for j = 1:256
x1(i,j) = xa(c+j);
xb1(i,j) = xb(c+j);
end
c=c+128;
end

y=x1;
yb=xb1;

%calcula fft
for i = 1:N
y(i,:)=y(i,:).*w;
yb(i,:)=yb(i,:).*w;

Y(i,:) = fft(y(i,:),256);
Yb(i,:) = fft(yb(i,:),256);

Pyy(i,:) = Y(i,:).*conj(Y(i,:))/256;
Pyyb(i,:) = Yb(i,:).*conj(Yb(i,:))/256;

f1(i,:) = 8000*(0:128)/256;
end

%vetor bark
z=13*atan(0.76*f1/1000)+3.5*atan((f1/7.5/1000).^2);

%somatorio das bandas
BSD(1:17)=0;
for i = 1:N
for j = 1:129
if (floor(z(i,j)) > 0)
c = floor(z(i,j));
for k = 1:2
BSD(c)= BSD(c) + (Pyy(i,k)-Pyyb(i,k))^2;
end
end
end
end

BSD1(u,:)=BSD; %Matriz com os resultados para cada
vetor testado

end %loop u

```

```

for m=1:d2

for i=5:8
sum(m)=sum(m)+BSD1(m,i);
end

end %loop m

clear xe

```

```

clear ve

media=media+sum; % Soma de cada um dos testes

end %fim do loopn

media=(media/n); %Media da soma de todos os testes

```

## Programa utilizado no Teste 2

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Este programa calcula o valor de MOS obtido a partir
% das diferenças entre um %sinal codificado em G.729 e o
% mesmo sinal degradado em varios níveis de erro
% de acordo com a multiplicação do sinal original pelo
% vetor gerado no NS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

fid = fopen('bit729');
[x,a2] = fread(fid,'ubit1');
fclose(fid);
d=0;
x='x';
media=0;

```

```

vv=load('2925010b');
v1=vv(:,2);
[b2,a2]=size(x);

cc=ceil(a2/80);

```

```

d2=1000; %numero de vetores erro

```

```

sum(1:d2)=0;
k=1;
ce(1:d2)=0;
for i=1:d2
    for j=1:cc
        ve(j,i)=v1(k);
        k=k+1;
    end
    k=k-cc+1;
end

for i=1:d2
    for j=1:cc
        if (ve(j,i)==0)
            ce(i)=ce(i)+1;
        end
    end
    e(i)=ce(i)/cc;
end

```

```

%for m=1:d2

```

```

%Gera uma matriz com varios sinais degradados de
acordo com o vetor erro

```

```

for i=1:d2
    kk=1;
    for j=1:cc
        for k=1:80 %Tamanho do quadro=80
            if (kk <= a2)
                xe(kk,i)=x(kk)*ve(j,i);
                kk=kk+1;
            end
        end
    end
end
end
end

```

```

% Grava cada um dos arquivos degradados
for i = 1:d2
    fid = fopen(['P' int2str(i)],'w');
    fwrite(fid,xe(:,i),'ubit1');
    fclose(fid);
end

```

```

%-----

```

```

%Teste para cada um dos sinais
for u=1:d2
    S = ['copy P' int2str(u) ' PX'];
    [status,result] = dos(S);
    result;
    [status,result] = dos('conv');

```

```

[xa,Fs,bits]=wavread('sampleorig');
[xb,Fs,bits]=wavread('sampletest');
[a3,a4]=size(xb);

```

```

%Filtering
N=floor(a3/128)-1;
w=hann(256);
t = 0:0.000125:0.6;
w=w';
c=0;
c1=0;

```

```

for i = 1:N
    for j = 1:256
        x1(i,j) = xa(c+j);
        xb1(i,j) = xb(c+j);
    end
    c=c+128;
end

```

```

y=x1;
yb=xb1;

```

```

%calcula fft
for i = 1:N
    y(i,:)=y(i,:).*w;
    yb(i,:)=yb(i,:).*w;

```

```

Y(i,:) = fft(y(i,:),256);
Yb(i,:) = fft(yb(i,:),256);

```

```

Pyy(i,:) = Y(i,:).* conj(Y(i,:)) / 256;
Pyyb(i,:) = Yb(i,:).* conj(Yb(i,:)) / 256;

```

```

f1(i,:) = 8000*(0:128)/256;
end

```

```

%vetor bark
z=13*atan(0.76*f1/1000)+3.5*atan((f1/7.5/1000).^2);
%somatorio das bandas

```

```

BSD(1:17)=0;
for i = 1:N
    for j = 1:129
        if (floor(z(i,j)) > 0)
            c = floor(z(i,j));
            for k = 1:2
                BSD(c)= BSD(c) + (Pyy(i,k)-Pyyb(i,k))^2;
            end
        end
    end
end

BSD1(u,:)=BSD; %Matriz com os resultados para cada
vetor testado
u
end %loop u

%end %loop m
for m=1:d2
    for i=5:8
        sum(m)=sum(m)+BSD1(m,i);
    end
end
sum=sum*100000;
for i=1:d2
    MOS(i) = 0.95*exp(-2.626*sum(i)) + exp(-308.7*sum(i))-
0.0473 + 1.95;
    MOS2(i) = 2.694*exp(-0.0112*e(i)*100) + 0.9788*exp(-
0.09021*e(i)*100);
end

%%%%%%%%%%%%%%
% Este programa calcula o ULP de cada fragmento do
%vetor de erro do NS
%utilizado na multiplicação com o sinal de voz
%
%%%%%%%%%%%%%%
k=1;
d2=1000;
ce(1:d2)=0;
cclp(1:d2)=0;
cp(1:d2)=0;
cc1(1:d2)=0;

```

```

CLP(1:d2)=0;
p(1:d2)=0;
q(1:d2)=0;
for i=1:d2
    for j=1:cc
        ve(j,i)=v1(k);
        k=k+1;
    end
    k=k-cc+1;
end

for i=1:d2
    for j=1:cc
        if (ve(j,i)==0)
            ce(i)=ce(i)+1;
            if (j<cc)
                if (ve(j+1,i)==0)
                    cclp(i)=cclp(i)+1;
                end
            end
        end
        if (ve(j,i)==1)
            cc1(i)=cc1(i)+1;
            if (j<cc)
                if (ve(j+1,i)==0)
                    cp(i)=cp(i)+1;
                end
            end
        end
    end
    if (ce(i)~=0)
        CLP(i)=cclp(i)/ce(i);
    end
    q(i)=1-CLP(i);
    if (cc1(i)~=0)
        p(i)=cp(i)/cc1(i);
    end
    if ((p(i)+q(i))~=0)
        ULP(i)=p(i)/(p(i)+q(i));
    end
    e(i)=ce(i)/cc;
end

```

## ANEXO II

### Programa em tcl do NS para a rede simulada

```
#####
#
# Este programa cria uma rede simulada e gera tráfego
# simulando #congestionamento entre pacotes VoIP e
# tráfego web.
#
#####

#Criação do objeto de simulação
set ns [new Simulator]

#Abertura dos arquivos trace
set tf [open out.tr w]
set Out [open Out.ns w]

#Link onde ocorre congestionamento
set n0 [$ns node]
set n1 [$ns node]
$n0 color blue
$n1 color blue
$ns duplex-link $n0 $n1 10Mb 100ms DropTail
$ns queue-limit $n0 $n1 300

#Fonte de tráfego CBR
set n3 [$ns node]
$n3 color red
$ns duplex-link $n3 $n0 1000Mb 20ms DropTail
$ns queue-limit $n3 $n0 20

#Fonte de tráfego background
set n4 [$ns node]
$ns duplex-link $n4 $n0 1000Mb 20ms DropTail
$ns queue-limit $n4 $n0 20

#Destino do tráfego CBR
set n5 [$ns node]
$n5 color red
$ns duplex-link $n1 $n5 1000Mb 20ms DropTail
$ns queue-limit $n1 $n5 20

#Destino do tráfego de background
set n6 [$ns node]
$ns duplex-link $n1 $n6 1000Mb 20ms DropTail
$ns queue-limit $n1 $n6 20

#Dados para o arquivo trace
$ns trace-queue $n0 $n1 $tf

#Número de fontes de tráfego background
set NodeNb 100
set NumberFlows 20
set tempo [expr $NumberFlows * 1.5 ]

#Criação da conexão UDP
set udp [new Agent/UDP]
$ns attach-agent $n3 $udp
set null [new Agent/Null]

$ns attach-agent $n5 $null
$ns connect $udp $null

#Criação do tráfego CBR sobre UDP
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set packetSize_ 200
$cbr set type_ CBR
$cbr set interval_ 0.02
$ns at 2 "$cbr start"
$ns at $tempo "$cbr stop"

#Links entre a fonte CBR e o roteador
for {set j 1} {$j<=$NodeNb} {incr j} {
    set S($j) [$ns node]
    $ns duplex-link $S($j) $n4 100Mb 1ms DropTail
    $ns queue-limit $S($j) $n4 1000
}

#Criação do tráfego de background
for {set i 1} {$i<=$NodeNb} {incr i} {
    for {set j 1} {$j<=$NumberFlows} {incr j} {
        set tcp_src($i,$j) [new Agent/UDP]
        set tcp_snk($i,$j) [new Agent/Null]
    }
}

for {set i 1} {$i<=$NodeNb} {incr i} {
    for {set j 1} {$j<=$NumberFlows} {incr j} {
        $ns attach-agent $S($i) $tcp_src($i,$j)
        $ns attach-agent $n6 $tcp_snk($i,$j)
        $ns connect $tcp_src($i,$j) $tcp_snk($i,$j)
        $tcp_src($i,$j) set packetSize_ 1500
    }
}

for {set i 1} {$i<=$NodeNb} {incr i} {
    for {set j 1} {$j<=$NumberFlows} {incr j} {
        set ftp($i,$j) [new Application/FTP]
        set ftp($i,$j) [$tcp_src($i,$j) attach-source FTP]
        $ftp($i,$j) set type_ FTP
    }
}

#geração de números randômicos para o tráfego de
background
set rng1 [new RNG]
$rng1 seed 0
set rng2 [new RNG]
$rng2 seed 0
set rng3 [new RNG]
$rng3 seed 0
set rng4 [new RNG]
$rng4 seed 0
set rng5 [new RNG]
$rng5 seed 0

#Criação das variáveis "Active OFF Times"
set RV [new RandomVariable/Exponential]
$RV set avg_ 0.13
$RV use-rng $rng1
```

```

# Criação das variáveis "Request Sizes"
set RVSize [new RandomVariable/Pareto]
$RVSize set avg_ 10000
$RVSize set shape_ 1.2
$RVSize use-rng $rng2

#Criação da variável auxiliar para "Inactive off Times"
set RVOff [new RandomVariable/Pareto]
$RVOff set avg_ 3
$RVOff set shape_ 1.5
$RVOff use-rng $rng3

#Criação das variáveis auxiliar para "Embedded
References"
set RVEmb [new RandomVariable/Pareto]
$RVEmb set avg_ 1.7
$RVEmb set shape_ 2.43
$RVEmb use-rng $rng4

#Criação de variável para o tamanho máximo de pacote
background
set r1 [new RandomVariable/Uniform]
$r1 use-rng $rng5
$r1 set min_ 0
$r1 set max_ 1

#Criação das variáveis Weibull e Pareto
for {set i 1} {$i<=$NodeNb} {incr i} {
for {set j 1} {$j<=$NumberFlows} {incr j} {
set y1 [$r1 value]
set x1 [expr 1000/(1-$y1)]
set x2 [expr 1.46*pow(-log($y1),1/0.382)]
set Size($i,$j) $x1
while {$x2>=1} {
set y1 [$r1 value]
set x2 [expr 1.46*pow(-log($y1),1/0.382)]
}
set AcOff($i,$j) $x2
}
}

#Grava as variáveis do tráfego background para análise
dos dados
for {set i 1} {$i<=1000} {incr i} {
set y1 [$r1 value]
set x1 [expr 1000/(1-$y1)]
set x2 [expr 1.46*pow(-log($y1),1/0.382)]
set FSize($i) $x1
set AcOf($i) $x2
set lnOf($i) [$RVOff value]
set EmRef($i) [$RVEmb value]
puts $Out "FSize($i) $AcOf($i) $lnOf($i) $EmRef($i)"
}

#Criação do numero de "Embedded References" e "Active
OFF Times"
for {set i 1} {$i<=$NodeNb} {incr i} {
for {set j 1} {$j<=$NumberFlows} {incr j} {
set EmbRef($i,$j) [expr 1 + [ $RVEmb value]]
set EmbRef($i,$j) [expr round($EmbRef($i,$j))]
set ActOFF($i,$j) [$RVOff value]
}
}

```

```

#Contador do numero de paginas
for {set i 1} {$i<=$NodeNb} {incr i} {
set tot($i) 0
for {set j 1} {$j<=$NumberFlows} {incr j} {
set tot($i) [ expr $tot($i) + $EmbRef($i,$j)]
set tot1($i,$j) $tot($i)
}
}

#Intervalo de transmissão entre pacotes
set c1 0
set t1 $stempo
for {set i 1} {$i<=$NodeNb} {incr i} {
set k 1
set t [$ns now]
for {set j 1} {$j<=$NumberFlows} {incr j} {
set c [expr $tot1($i,$k) + 1]
if {$j == $c} {
set t [expr $t + [$RVOff value]]
set k [expr $k + 1]
} else {
set t [expr $t + $AcOff($i,$j)]
}
set c1 [expr $c1 + 1]
set Conct($i,$j) $t

if {$j == $NumberFlows} {
if {$t < $t1} {
set t1 $t
}
}
$ns at $Conct($i,$j) "$ftp($i,$j) send $Size($i,$j)"
}
}

#Calculo velocidade de transmissão de pacotes
set som 0
set t2 30.0
set c2 0.0
for {set i 1} {$i<=$NodeNb} {incr i} {
for {set j 1} {$j<=$NumberFlows} {incr j} {
if {$Conct($i,$j) <= $t1} {
set som [expr $som + $Size($i,$j)]
set c2 [expr $c2+1]
}
}
}
set velbps [expr $som*8/$t1/1000000]
set velpkts [expr $c2/$t1]
puts $Out "$NodeNb $t1 $velbps Mbps $velpkts pkts/sec"

proc finish {} {
global ns tf nf
$ns flush-trace
close $tf
exit 0
}

$ns at $t1 "finish"
$ns run

```