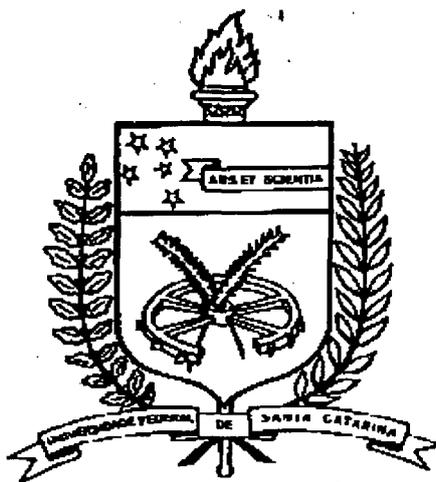


UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

**MULTIMÍDIA EM REDES : UM ESTUDO DA TRANSMISSÃO  
DE SINAIS DE VÍDEO E ÁUDIO EM UM BACKBONE FDDI**



DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA  
CATARINA PARA A OBTENÇÃO DE GRAU DE MESTRE EM CIÊNCIAS  
DA COMPUTAÇÃO.

CLÁUDIA JACY BARENCO

FLORIANÓPOLIS, ABRIL DE 1995.

MULTIMÍDIA EM REDES : UM ESTUDO DA TRANSMISSÃO DE SINAIS DE VÍDEO E ÁUDIO EM UM BACKBONE FDDI

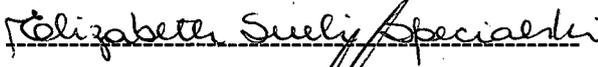
CLÁUDIA JACY BARENCO

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA OBTENÇÃO DO TÍTULO DE

MESTRE EM CIÊNCIAS DA COMPUTAÇÃO

ESPECIALIDADE SISTEMAS DE COMPUTAÇÃO E APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIAS DA COMPUTAÇÃO

ORIENTADOR

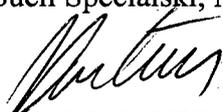
  
Profª Elizabeth Sueli Specialski, M.Sc.

COORDENADOR DO CURSO

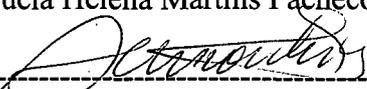
Prof. Rogério Cid Bastos, Dr.

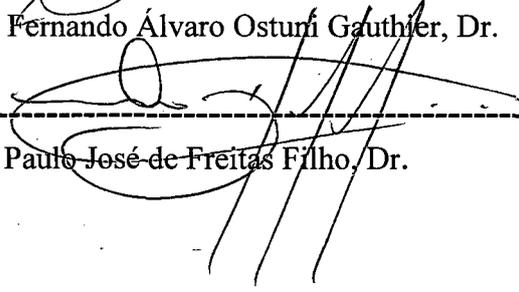
BANCA EXAMINADORA

  
Profª Elizabeth Sueli Specialski, M.Sc.  
PRESIDENTE

  
Prof. Luiz Lustosa Vieira, Dr.

  
Profª Lúcia Helena Martins Pacheco, M.Sc.

  
Prof. Fernando Álvaro Ostuni Gauthier, Dr.

  
Prof. Paulo José de Freitas Filho, Dr.

*Aos  
meus pais, Lindalva e Dalzenir,  
que sempre me deram exemplo  
de que não há conquistas valorosas  
na vida, sem trabalho e dedicação.*

## AGRADECIMENTOS

A pesquisa das informações necessárias para o desenvolvimento desse trabalho seria impossível sem a ajuda de muitas pessoas. Sinto-me muito feliz por poder pertencer a um grupo que tem um compromisso com o desenvolvimento de novas soluções na área da computação.

Agradeço primeiramente à minha orientadora, Beth, que aceitou o desafio de desenvolvermos um trabalho em uma área tão nova. Também agradeço ao Prof. Luiz Lustosa Vieira pelo contínuo esforço em me disponibilizar todos os recursos necessários para o desenvolvimento do projeto.

Gostaria de agradecer à Datamec Sistemas e Processamento de Dados S.A., empresa de que sou funcionária, e que me permitiu estar aqui na Universidade Federal de Santa Catarina para desenvolver esse trabalho de mestrado.

Agradeço à empresa 3COM do Brasil, em especial à Vânia Ferro e ao Laerte Pires, que nos forneceu os equipamentos necessários para concretização do projeto.

Aos inúmeros amigos que direta ou indiretamente contribuíram na elaboração desse trabalho, principalmente aqueles com quem convivi durante o curso de mestrado. Também agradeço ao meu amigo Alexandre (Brasília), amigo de muito tempo, que me influenciou na escolha da realização do mestrado na UFSC.

Ao pessoal do NPD da UFSC, Melo, Elvis, André, Wolmar, Richard, Jussara e Kátia, que incontáveis vezes tiveram paciência e me ajudaram nos problemas de rede.

Ao departamento de Enfermagem da UFSC, que me acolheu e permitiu a utilização de suas instalações físicas e dos equipamentos.

## RESUMO

Este trabalho apresenta uma análise sobre o comportamento do anel óptico FDDI, existente na rede local UFSC, quando são transmitidos sinais de vídeo e de áudio.

Para a realização desta análise, são apresentados conceitos básicos sobre multimídia em redes, padrão MPEG (Motion Picture Experts Group) de compressão de dados e sobre a arquitetura FDDI.

Apresenta-se, ainda, o ambiente de gerenciamento e o modelo de simulação utilizados para a coleta e análise de dados.

Uma interface para criação, visualização, transmissão e recepção de sinais de vídeo e áudio em computadores é apresentada como uma contribuição na melhoria da interação do usuário para a manipulação de recursos multimídia em redes.

## ABSTRACT

This study shows an analyse about the behaviour of the FDDI optical ring of the local area network of UFSC, when video and audio signals are transmitted.

To do this analyse, were presented basic concepts about multimedia networks, MPEG (Motion Picture Experts Group) standard to compress videos and audios and the FDDI architecture.

A management environment and a simulation model are presented too; these technologies were used to collect and to analyse data.

A interface to create, to reproduce, to transmit and to receive video and audio signals in computer networks is presented as a contribution to the improvement of the interaction of the user when using multimedia resources in networks.

## SUMÁRIO

Resumo.....	v
<i>Abstract</i> .....	v
Lista de Siglas.....	ix
Glossário.....	xi
Lista de Figuras.....	xiii
Lista de Fórmulas .....	vix
Lista de Tabelas.....	vii
<b>Capítulo 1 - INTRODUÇÃO</b>	
1.1 - Introdução.....	1
<b>Capítulo 2 - TECNOLOGIA MULTIMÍDIA EM REDES</b>	
2.1 - Introdução.....	4
2.2 - Fundamentos de Sistemas Multimídia em redes.....	5
2.2.1 - Texto.....	5
2.2.2 - Imagem gráfica.....	5
2.2.3 - Áudio.....	6
2.2.4 - Vídeo.....	7
2.3 - Limitações das tecnologias de redes de computadores existentes para transmissão de multimídia.....	8
2.4 - Protocolos Multimídia em redes.....	9
<b>Capítulo 3 - TECNOLOGIA FDDI</b>	
3.1 - Introdução.....	12
3.2 - Descrição da tecnologia de fibra ótica.....	12
3.3- Arquitetura FDDI .....	16
3.4 - O controle de acesso ao meio da arquitetura FDDI .....	18
3.5 - Formato dos quadros da arquitetura FDDI .....	20
3.6 - Alocação das mídias aos serviços síncronos e assíncronos da arquitetura FDDI	21
3.7 - FDDI II.....	22
3.8 - Alocação das mídias ao serviço isócrono.....	25
3.9 - FDDIII.....	25
<b>Capítulo 4 - ARQUITETURA WINDOWS NT SERVER 3.5 / FDDI DO PROJETO</b>	
4.1 -Introdução.....	27
4.2 - Padrão NDIS.....	29
4.3 - Protocolo NetBEUI.....	29
4.4 - NetBIOS sobre Windows NT.....	29

4.5 - Protocolo TCP/IP.....	30
4.6 - TCP/IP sobre o Windows NT.....	30
4.7 - Padrão TDI.....	32
4.8 - Resolução de nome do NetBIOS.....	32
4.9 - Transmissão de pacotes IP sobre uma rede FDDI.....	33
4.9.1 - Formato do pacote.....	33
4.9.2 - Resolução de endereço.....	34
4.9.3 - Endereço <i>broadcast</i> .....	35
4.9.4 - Suporte <i>multicast</i> .....	35
4.9.5 - Formato <i>trailer</i> .....	35
4.9.6 - Ordenação dos bytes.....	35
4.9.7 - Detalhes sobre o nível MAC.....	35
<b>Capítulo 5 - SISTEMAS DE TELEVISÃO E VÍDEO</b>	
5.1 - Introdução.....	37
5.2 - Descrição dos sistemas de televisão e vídeo.....	37
5.3 - Padrão de Compressão MPEG I.....	39
5.3.1 - Compressão de vídeos.....	40
5.3.2 - Compressão de áudio.....	42
5.4 - Padrão de compressão MPEG II.....	45
5.4.1 - Compressão de vídeo.....	45
5.4.2 - Compressão de áudio.....	46
5.5 - Padrão de compressão MPEG III.....	46
5.6 - Padrão de compressão MPEG IV.....	46
5.8 - O estado da arte do padrão MPEG.....	46
<b>Capítulo 6 - DESCRIÇÃO DO AMBIENTE DE REDES UTILIZADO NO DESENVOLVIMENTO DO PROJETO</b>	
6.1 - Introdução.....	49
6.2 - Descrição do ambiente de gerenciamento.....	52
6.3 - Descrição do modelo criado para gerenciamento do <i>backbone</i> FDDI e das estações multimídia.....	54
6.4 - Parâmetros a serem medidos para análise do desempenho da rede FDDI.....	59
6.5 - Descrição do software utilizado para compressão e reprodução dos vídeos e áudios no projeto.....	63
6.6 - Software desenvolvido como interface para criação /visualização /transmissão /recepção de vídeos e áudios.....	64
6.7 - Descrição do modelo de simulação.....	66

**Capítulo 7 - ANÁLISE DOS PARÂMETROS MEDIDOS**

7.1 - Introdução.....	71
7.2 - Descrição da análise dos dados coletados no ambiente de gerenciamento.....	72
7.3 - Descrição e análise dos resultados obtidos nas simulações.....	92

**Capítulo 8 - CONCLUSÃO**

8.1 - Conclusão.....	101
8.2 - Recomendações para trabalhos futuros.....	103
Referências bibliográficas.....	105
Anexo I - Listagem do projeto imagem.prj na linguagem C++.....	111
Anexo II - Listagem do modelo simulado FDDI.DOE.....	145
Anexo III - Endereços Internet utilizados no projeto.....	150
Anexo IV - Transformada Discreta do Cosseno (DCT).....	152
Anexo V - Descrição dos objetos da MIB FDDI [Case1993] e MIB II SNMP [McCloghrie1991] utilizados no projeto.....	157
Anexo VI - Gráficos gerados pelo software SunNet Manager.....	179

**LISTA DE SIGLAS**

ANSI	American National Standards Institute
APD	Avalanche Photodiode
ARP	Address Resolution Protocol
ASN.1	Abstract Syntax Notation.One
CD	Compact Disk
CS-HMUX	Circuit Switched Hybrid Multiplexer
DAC	Dual Attachment Concentrator
DARPA	Defense Advanced Research Projects Agency
DAT	Digital Audio Tape
DCT	Discrete Cosine Transform
DPG	Dedicated Packet Group
DSP	Digital Signal Processor
EISA	Extended Industry Standard Architecture
EPSL	Enhanced Performance and Services LAN
FDDI	Fiber Distributed Data Interface
FDM	Frequency Division Modulation
FTP	File Transfer Protocol
FTP	File Transfer Protocol
HDTV	High Definition Television
HMUX	Hybrid Multiplexer
ICMP	Internet Control Message Protocol
IEC	International Eletrotechnical Comission
ILD	Injection Laser Diode
IMAC	Isochronous MAC
IMC	Isochronous Maintenance Channel
IP	Internet Protocol
ISO	International for Standardization Organization
JTC1	Joint Technical Committee 1
LANs	Local Area Networks
LCF	Low Cost Fiber
LED	Light Emitting Diode
LLC	Logical Link Control
MAC	Medium Access Control
MIB	Management Information Base

MPEG	Motion Picture Experts Group
MS-DOS	Microsoft System
NBT	NetBios sobre oWindows NT
NDIS	Network Driver Interface Specification
NetBEUI	NetBIOS Extended User Interface
OS/2	Operating System/2
PHY	Physical Layer Protocol
PMD	Physical Medium Dependent
QOS	Quality of Service
RAM	Random Access Memory
RPC	Remote Procedure Call
SC29	Sub-Committee 29
SDH	Synchronous Digital Hierarchy
SMF	Single Mode Fiber
SMI	Structure of Management Information
SMT	Station Management
SNAP	Sub-Network Access Protocol
SNMP	Simple Network Management Protocol
SONET	Synchronous Optical Network
SPM	Sonet Physical Layer Mapping
ST II	Experimental Internet Stream Protocol : Version 2
TCP	Transmission Control Protocol
TDI	Transport Driver Interface
THT	Token Holding Timer
TP	Twisted Pair
TRT	Token Rotation Timer
TTRT	Target Token Rotation Time
VMTP	Versatile Message Transaction Protocol
WANs	Wide Area Networks
WBCs	Wideband Channels
WG 11	Working Group 11
XTP	Xpress Transfer Protocol

## GLOSSÁRIO

<i>Acknowledgment</i>	Resposta afirmativa enviada para o transmissor
<i>Anonymous</i>	Qualificação do usuário que acessa uma rede sem ter permissão. Como o próprio nome sugere, ele não precisa se identificar, conseqüentemente possui acesso a somente algumas informações..
<i>Backbone</i>	Ligação de alta velocidade onde várias redes se conectam através de <i>bridges</i>
<i>Binding</i>	Ligação ou conexão entre computadores, protocolos, etc ...
<i>Bridges</i>	Unidade que conecta duas redes que têm o mesmo procedimento de controle de conexão mas diferentes procedimentos de controle de acesso ao meio
<i>Broadcast</i>	Transmissão simultânea de dados para mais de um destino
<i>Buffer</i>	Local de armazenamento temporário
<i>Bypass</i>	Eliminação de uma estação de uma rede em anel permitindo o fluxo normal dos dados
<i>Cache</i>	<i>Buffer</i> de alta velocidade que contém instruções de dados mais freqüentemente acessados
<i>Checksum</i>	Somatório de um grupo de dados associado a um grupo e utilizado para o propósito de validação
<i>Color</i>	Dentro do reconhecimento óptico do caracter, representa a aparição spectral da imagem dependendo da imagem, da resposta spectral do observador e a composição spectral da luz incidente
<i>Default</i>	Um valor assumido quando nenhum valor for especificado
<i>Delay</i>	Quantidade de tempo que um evento é retardado
<i>Frame</i>	Área na qual uma imagem é visualizada
<i>Full color</i>	Todas cores permitidas pela visão do ser humano
<i>Gateway</i>	Uma unidade que conecta duas redes de computadores de diferentes arquiteturas
<i>Header</i>	Parte da mensagem que contém informações de controle como: nome estação origem, tipo de mensagem, prioridade, etc ...
<i>Host</i>	O computador primário ou computador que controla em uma instalação com múltiplos computadores
<i>Hub</i>	Unidade concentradora
<i>Multicast</i>	Transmissão de dados para um grupo selecionado de estações
<i>Performance</i>	Fator determinado pelo <i>throughput</i> , tempo de resposta e disponibilidade

<i>Pixels</i>	Picture Element
<i>Port ID</i>	Identificação do ponto de acesso para entrada de dados
<i>Proxy System</i>	Sistema procurador que permite a compatibilidade entre padrões de sistemas de gerenciamento diferentes
<i>Rank</i>	Número do nível
<i>Status</i>	Indicação da ocorrência de alguma condição
<i>Subnet mask</i>	Máscara de subrede
<i>Throughput</i>	Medida da quantidade de trabalho realizada por um sistema de computador durante um determinado período
<i>Timeout</i>	Extrapolação do limite de tempo para ganho de acesso a uma rede
<i>Trailer</i>	Informação de controle adicionada no final de um registro

## LISTA DE FIGURAS

Figura 2.1	Efeito da variação estatística do retardo.....	6
Figura 2.2	O parâmetro 'a'.....	8
Figura 2.3	Uma nova arquitetura multimídia [Chimiak1993].....	10
Figura 2.4	Outra arquitetura multimídia integrada [Chimiak1993].....	10
Figura 3.1	Guia de onda luminosa - fibra ótica [Giozza1986].....	13
Figura 3.2	Sistema de transmissão ponto-a-ponto em fibras óticas [Giozza1986].....	14
Figura 3.3	Protocolos da arquitetura FDDI [Soares1992].....	17
Figura 3.4	Formato dos quadros FDDI [Soares1992].....	20
Figura 3.5	Header do ciclo do FDDI II [Stallings1993].....	23
Figura 4.1	Arquitetura Windows NT/FDDI.....	28
Figura 4.2	Formato do pacote TCP/IP sobre FDDI.....	33
Figura 6.1	Interconexão do <i>Hub</i> com o <i>Backbone</i> FDDI.....	50
Figura 6.2	Configuração da conexão do <i>Hub</i> e das estações ao anel óptico.....	51
Figura 6.3	Ambiente de gerenciamento do projeto.....	53
Figura 6.4	Representação gráfica do ambiente de gerenciamento criado.....	59
Figura 6.5	Janela principal do software desenvolvido.....	65
Figura 6.6	Janela de transmissão e recepção de vídeos e áudios.....	66
Figura 6.7	Fluxograma do nível MAC FDDI.....	67
Figura 6.8	Representação geral do modelo simulado.....	68
Figura 6.9	Representação gráfica do modelo simulado.....	69
Figura 7.1	Simulação realizada com taxa de geração de dados a 100 Mbits/s (2 estações).....	95
Figura 7.2	Simulação realizada com taxa de geração de dados a 100 Mbits/s (10 estações).....	96
Figura 7.3	Simulação realizada com taxa de geração de dados a 4 Mbits/s (2 estações).....	97
Figura 7.4	Simulação realizada com taxa de geração de dados a 4 Mbits/s (10 estações).....	98
Figura 7.5	Simulação realizada com taxa de geração de dados a 4 Mbits/s (20 estações).....	99
Figura A.1	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 18/03/95 às 18:28 - Dados coletados de 30 em 30 segundos.....	180
Figura A.2	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 18/03/95 às 18:28 - Dados coletados de 30 em 30 segundos.....	181

Figura A.3	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 18/03/95 às 18:28 - Dados coletados de 30 em 30 segundos.....	182
Figura A.4	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 19/03/95 às 16:11 - Dados coletados de 30 em 30 segundos.....	183
Figura A.5	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 19/03/95 às 16:11 - Dados coletados de 30 em 30 segundos.....	184
Figura A.6	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 19/03/95 às 16:11 - Dados coletados de 30 em 30 segundos.....	185
Figura A.7	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 20/03/95 às 16:06 - Dados coletados de 30 em 30 segundos.....	186
Figura A.8	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 20/03/95 às 16:06 - Dados coletados de 30 em 30 segundos.....	187
Figura A.9	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 20/03/95 às 16:06 - Dados coletados de 30 em 30 segundos.....	188
Figura A.10	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 20/03/95 às 22:13 - Dados coletados de 30 em 30 segundos.....	189
Figura A.11	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 20/03/95 às 22:13 - Dados coletados de 30 em 30 segundos.....	190
Figura A.12	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 20/03/95 às 22:13 - Dados coletados de 30 em 30 segundos.....	191
Figura A.13	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 21/03/95 às 15:58 - Dados coletados de 30 em 30 segundos.....	192
Figura A.14	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 21/03/95 às 15:58 - Dados coletados de 30 em 30 segundos.....	193
Figura A.15	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 21/03/95 às 15:58 - Dados coletados de 30 em 30 segundos.....	194
Figura A.16	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 22/03/95 às 15:59 - Dados coletados de 30 em 30 segundos.....	195
Figura A.17	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 22/03/95 às 15:59 - Dados coletados de 30 em 30 segundos.....	196

Figura A.18	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 22/03/95 às 15:59 - Dados coletados de 30 em 30 segundos.....	197
Figura A.19	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 22/03/95 às 16:21 - Dados coletados de 30 em 30 segundos.....	198
Figura A.20	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 22/03/95 às 16:21 - Dados coletados de 30 em 30 segundos.....	199
Figura A.21	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 22/03/95 às 16:21 - Dados coletados de 30 em 30 segundos.....	200
Figura A.22	Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 22/03/95 às 18:26 - Dados coletados de 30 em 30 segundos.....	201
Figura A.23	Gráfico da taxa de erro de conexão na coleta de dados realizada no dia 22/03/95 às 18:26- Dados coletados de 30 em 30 segundos.....	202
Figura A.24	Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 22/03/95 às 18:26 - Dados coletados de 30 em 30 segundos.....	203

**LISTA DE FÓRMULAS**

Fórmula 5.1	Teorema de Nyquist.....	43
Fórmula A.1	Definição do algoritmo DFT.....	153
Fórmula A.2	Definição do algoritmo DFT com a parte real separada da imaginária.....	153
Fórmula A.3	Definição da parte real do algoritmo DFT.....	153
Fórmula A.4	Definição da parte imaginária do algoritmo DFT.....	153
Fórmula A.5	Definição do algoritmo DCT.....	155

## LISTA DE TABELAS

Tabela 5.1	Arquivo de vídeo WIND.AVI compactado pelo padrão MPEG.....	47
Tabela 5.2	Arquivo de vídeo HEYA.AVI compactado pelo padrão MPEG.....	48
Tabela 5.3	Arquivo de áudio TADA.WAV não compactado.....	48
Tabela 5.4	Arquivo de áudio Greeting.MP2 compactado pelo padrão MPEG.....	48
Tabela 6.1	Descrição das requisições de eventos.....	55
Tabela 6.2	Descrição das requisições de dados.....	55
Tabela 6.3	Descrição dos <i>traps</i> .....	58
Tabela 6.4	Parâmetros e características do ambiente de gerenciamento.....	61
Tabela 6.5	Características do ambiente Windows NT.....	61
Tabela 7.1	Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 18/03/95 às 18:28).....	72
Tabela 7.2	Transmissão de vídeos compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 18/03/95 às 18:28).....	73
Tabela 7.3	Transmissão de áudios compactados ( Dados coletados pelo software SunNet Manager - dia : 19/03/95 às 16:11 ).....	74
Tabela 7.4	Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 19/03/95 às 16:11 ).....	75
Tabela 7.5	Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 20/03/95 às 16:06).....	77
Tabela 7.6	Transmissão de vídeos compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 20/03/95 às 16:06).....	78
Tabela 7.7	Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 20/03/95 às 22:13).....	79
Tabela 7.8	Transmissão de vídeos compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 20/03/95 às 22:13).....	80
Tabela 7.9	Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 21/03/95 às 15:58).....	81
Tabela 7.10	Transmissão de vídeos compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 21/03/95 às 15:58).....	82
Tabela 7.11	Transmissão de áudios compactados ( Dados coletados pelo software SunNet Manager - dia : 22/03/95 às 15:59 ).....	84

Tabela 7.12	Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 22/03/95 às 15:59 ).....	85
Tabela 7.13	Transmissão de áudios compactados ( Dados coletados pelo software SunNet Manager - dia : 22/03/95 às 16:21 ).....	86
Tabela 7.14	Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 22/03/95 às 16:21 ).....	87
Tabela 7.15	Transmissão de áudios compactados ( Dados coletados pelo software SunNet Manager - dia : 22/03/95 às 18:26 ).....	88
Tabela 7.16	Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 22/03/95 às 18:26 ).....	89
Tabela 7.17	Parâmetros atribuídos para as coletas de dados e para a simulação.....	90
Tabela 7.18	Parâmetros atribuídos para a simulação.....	93

# Capítulo 1

## INTRODUÇÃO

### 1.1 - INTRODUÇÃO

Pode-se questionar porque a equipe deste projeto está preocupada com a transmissão de sinais de vídeo e áudio em uma rede de computadores. Uma vez que é sabido que existem algumas plataformas '*stand-alone*' que podem apresentar informações multimídia a partir de uma área de armazenamento local como CD-ROM, discos laser ou até dispositivos magnéticos. Para algumas aplicações esta plataforma pode até ser adequada, mas, a apresentação de informações multimídia através de uma rede se torna essencial quando o custo de aquisição e armazenamento de informações localmente faz com que a arquitetura '*stand-alone*' seja inviável.

As tecnologias que utilizam recursos multimídia em redes de computadores como a teleconferência, exigem uma transmissão síncrona e em tempo real e as características e os requisitos de comunicação dos diferentes tipos de informação (texto, imagem, som e vídeo) que variam enormemente. Esses requisitos são: retardo máximo de transferência, variação estatística do retardo, natureza do tráfego gerado (contínuo ou rajada), vazão média e taxas aceitáveis de erro de bit e pacotes de dados [Soares1992].

A mídia áudio caracteriza-se por gerar um tráfego contínuo. Mesmo quando se realiza a compactação por detecção de silêncio, o sinal deve ser reproduzido no destino a uma taxa constante. Quanto ao problema de erro de bit e pacote, o áudio por possuir sinais muito redundantes, ou seja, um sinal pode compensar facilmente a perda de outro sinal, fazendo com que esse problema não seja muito preocupante.

A mídia vídeo também caracteriza-se por ter um tráfego contínuo. Normalmente, como vídeo vem acompanhado de áudio, tendo obedecido os requisitos de retardo de um estão obedecidos o do outro. Em vídeo, a taxa de erro de pacote pode ser maior que a de bit, pois como em um vídeo são gerados vários quadros por segundo, um quadro pode compensar o erro do outro.

O projeto de multimídia em redes desenvolvido tem como objetivo o estudo do comportamento da rede com arquitetura FDDI (Fiber Distributed Data Interface) da

Universidade Federal de Santa Catarina. Como foi visto, as características dos recursos multimídia são diferentes das características dos recursos que transmite-se atualmente nas redes de computadores, que na sua grande maioria são recursos de texto, assim os novos sistemas de comunicação devem poder suportar isso.

Esse estudo engloba :

1) Avaliação do desempenho desta rede quando se transmite sinais de vídeo e áudio. Os parâmetros utilizados na avaliação referem-se à taxa de transmissão, a taxa de erro, a média do retardo de transmissão e controle de fluxo e congestionamento. Isso nos permite medir as limitações da transmissão desses recursos multimídia em uma rede FDDI;

2) Comprovação da viabilidade da utilização dessa arquitetura para o projeto de ensino à distância via redes de computadores [Vieira1994], [Barenco1994], [Santos1993].

Os testes de transmissão de sinais de vídeo e áudio são realizados entre duas estações multimídia conectadas a um anel óptico FDDI, utilizando como protocolo de transporte e de rede o TCP/IP ( Transmission Control Protocol / Internet Protocol ) e o sistema operacional Windows NT Server 3.5 da Microsoft. Foi desenvolvido um software para a criação/visualização e transmissão/recepção dos sinais de vídeo e áudio, melhorando a interface do usuário com o sistema com relação à manipulação desses recursos multimídia.

Para o gerenciamento e avaliação do desempenho da rede está sendo utilizado o protocolo padrão Internet SNMP (Simple Network Management Protocol).

Apesar do sistema operacional UNIX ser bem mais robusto para o desenvolvimento do estudo, a escolha de microcomputadores PC (Personal Computer) com arquitetura Intel 486 e 386 foi feita por várias razões :

- O microcomputador é uma máquina que é facilmente encontrada em qualquer instituição de ensino e empresa, viabilizando a nossa idéia de implantar o projeto de ensino a distância a nível nacional;
- Há uma grande variedade de recursos de hardware e software para manipulação de sinais de vídeo e áudio para o ambiente PC;
- O hardware disponível para compressão de vídeo e áudio está mais evoluído tecnologicamente para a plataforma PC do que para a plataforma UNIX.

A seguir descreve-se os assuntos abordados em cada capítulo.

No capítulo 2 busca-se mostrar os fundamentos dos sistemas multimídia e as características intrínsecas de cada recurso : texto, imagem, áudio e vídeo. Analisa-se também, o comportamento das redes atuais frente a esta tecnologia, identificando-se as necessidades básicas para transmissão de multimídia.

No capítulo 3 faz-se um estudo da tecnologia de fibra ótica, já que é o meio físico de transmissão utilizado no projeto.

O capítulo 4 apresenta a arquitetura de alta velocidade FDDI, que é a arquitetura utilizada no backbone de fibra ótica da Universidade Federal de Santa Catarina.

No capítulo 5 descreve-se a arquitetura Windows NT 3.5 sob a arquitetura FDDI, que é o ambiente dos microcomputadores utilizados no projeto.

No capítulo 6 explica-se alguns conceitos sobre os sistemas de televisão e vídeo, já que no capítulo subsequente serão utilizados muitos desses conceitos.

No capítulo 7 mostra-se o padrão de compressão MPEG (Motion Picture Experts Group), que é o padrão de compressão de vídeo e áudio utilizado no projeto.

O capítulo 8 mostra-se o ambiente de gerenciamento criado para a coleta dos parâmetros para a análise do comportamento do *Backbone* FDDI. Também descreve-se sobre o software desenvolvido para a melhora da interface com o usuário para transmissão de vídeos e áudios. E finalmente mostra-se o modelo de simulação desenvolvido para simulação do comportamento do *Backbone* FDDI para 'n' estações.

No capítulo 9 analisa-se estatisticamente os parâmetros medidos no ambiente de gerenciamento e os resultados obtidos com as simulações e no capítulo 10 descreve-se a conclusão do comportamento do *Backbone* FDDI para transmissão de sinais de vídeo e áudio.

## Capítulo 2

### TECNOLOGIA MULTIMÍDIA EM REDES

#### 2.1 - INTRODUÇÃO

De muitos estudos publicados depreende-se que o texto não é necessariamente o recurso mais eficiente para aprendizagem; ao contrário, o ambiente multisensorial que integra textos, gráficos e som oferece estímulos complementares a ambos os hemisférios cerebrais e cria melhores condições de atenção, motivação, aprendizagem e retenção. Destaca-se, entretanto, que o uso de som, imagens ou gráficos nos sistemas multimídia não é algo arbitrário e deve ser norteado pelas regras de percepção e comunicação visual. Grupos de produção de programas multimídia estão explicitando os fundamentos apropriados à formulação adequada de mensagens gráficas [Alder1993].

O uso de multimídia em microcomputador tem se tornado cada vez mais freqüente, devido a padronizações antes inexistentes. O padrão Multimedia PC (MPC) foi estabelecido e é regulado por um grupo de empresas, representando uma extensão ao padrão PC e servindo como base para o desenvolvimento de aplicações em multimídia. A adoção do padrão MPC como a especificação técnica a ser usada pelo mercado de informática, possibilita a garantia de integração de diferentes produtos de "hardware" e "software" de multimídia.

As especificações para uma máquina multimídia compatível com o padrão MPC 1 apresentam, como configuração básica, um microcomputador com processador 386SX, com memória RAM de, no mínimo, 2 MBytes, 30 MBytes de disco rígido, drive de CD-ROM com taxa de transferência mínima de 150 Kbytes/segundo e 1 segundo de tempo de acesso [Victoria1993].

Para se trabalhar no padrão MPC 2, algumas das exigências básicas são disco rígido de, no mínimo, 160 Mbytes, drives de CD-ROM com taxa de transferência mínima de 300 Kbytes/segundo e compatibilidade com o Photo-CD da Kodak [Victoria1993].

## 2.2 - FUNDAMENTOS DE SISTEMAS MULTIMÍDIA

As tecnologias que utilizam recursos multimídia como a teleconferência, exigem uma transmissão síncrona em tempo real e as características e requisitos de comunicação dos diferentes tipos de informação: texto, imagem, som e vídeo variam enormemente de um para outro. Esses requisitos são:

- Retardo máximo de transferência, que é o máximo tempo aceitável entre o envio e a recepção da informação no destino;
- Variação estatística do retardo, que determina estatisticamente de quanto esse tempo de retardo variará;
- Natureza do tráfego gerado, determina se o fluxo da informação é contínuo ou em rajada;
- Vazão média, determina a taxa de transmissão da rede;
- Taxas aceitáveis de erro de bit e pacotes de dados, determina a taxa de erro por bit e por pacote [Soares1992].

A seguir descreve-se o comportamento de cada mídia quando são transmitidas em uma rede de computadores.

### 2.2.1 - TEXTO

O tráfego gerado por informações em texto é em rajada, não existindo um fluxo contínuo de informação. Para o texto, o retardo máximo de transferência e a variação estatística do retardo não são problemas, sendo seus requisitos facilmente satisfeitos pelo sistema de comunicação hoje existente nas empresas e nas instituições de ensino. Quanto à tolerância a erros, na grande maioria das aplicações, não se pode tolerar erro nem em um único bit [Soares1992].

### 2.2.2 - IMAGEM GRÁFICA

No caso das imagens estáticas e animadas, o tráfego gerado é também em rajadas, já que em alguns períodos não temos fluxo de informações. Como nos textos, o retardo máximo e a variação estatística do retardo não causam problemas. Para imagens gráficas, a taxa de erro de bit pode ser bem maior que a taxa de erro de pacote, uma vez que não haverá nenhum problema se

um *pixel*, um ponto da tela, ficar azul ao invés de verde, mas a perda de um pacote pode significar o apagamento de um bloco de imagem na tela [Soares1992].

### 2.2.3 - ÁUDIO

O recurso de áudio caracteriza-se por gerar um tráfego contínuo. Mesmo quando no sinal de voz é realizada a compactação por detecção de silêncio, ou seja, nos períodos onde não temos nenhum sinal de áudio, este não é transmitido pela rede; os sinais devem ser reproduzidos no destino a uma taxa constante [Soares1992].

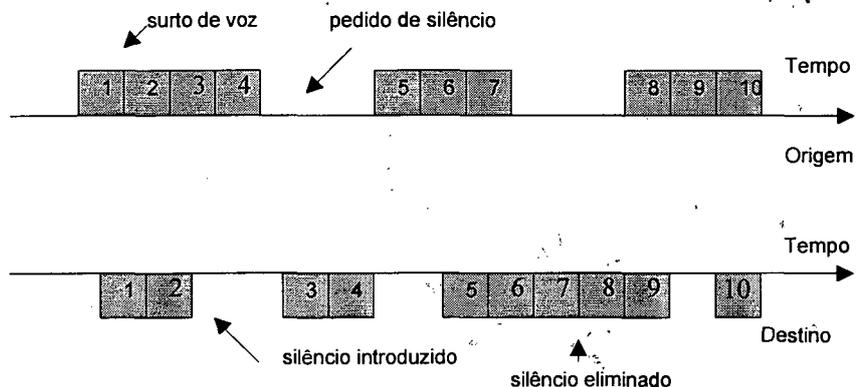


Figura 2.1 - Efeito da variação estatística do retardo [Soares1992]

Os surtos de voz e os surtos de silêncio são gerados na fonte a uma taxa constante e esses surtos de silêncio não são transmitidos. Assim, se os pacotes sofrerem retardos variados, chegarão ao destino não mais preservando a continuidade, podendo ainda gerar períodos de silêncio onde não existia, como podemos ver na figura 2.1, prejudicando conseqüentemente a inteligibilidade da informação no destino.

Os algoritmos de compensação têm como princípio básico assegurar uma reserva de pacotes antes de iniciar o processo de reprodução da voz, introduzindo um retardo inicial a cada surto de voz. Mas o problema não está resolvido, já que este retardo é limitado pelo máximo retardo de transferência permitido para o sinal de voz.

Se o retardo de transferência for muito grande, a conversação sentirá um efeito de ruptura, ou seja, serão gerados períodos de silêncio, tornando a percepção da conversação inviável. Um retardo maior que 200 ms já começa a incomodar os interlocutores [Soares1992].

Quanto às perdas, a taxa de erro de bits ou de pacotes podem ser altas devido ao alto grau de redundância dos sinais de áudio. O único requisito é que os pacotes não sejam muitos grandes, (no caso de voz, menores que uma sílaba). As perdas em torno de 1% são toleráveis [Gopal1984, Gruber1985], mas como a maioria das redes de alta velocidade utilizam fibra ótica, a taxa de erro típica é de  $10^{-9}$  ou menos. Sendo assim, a detecção de erro pode ser tranquilamente dispensada, trazendo assim um maior desempenho.

#### 2.2.4 - VÍDEO

A mídia de vídeo também se caracteriza por ter um tráfego contínuo. Normalmente, como o vídeo vem acompanhado de áudio, tendo obedecido os requisitos de retardo de um estão obedecidos os do outro. Em vídeo, a taxa de erro de bit pode ser maior que a taxa de erro de pacote, pelos mesmos motivos já citados e como são gerados vários quadros por segundo, a taxa de erro de pacote não é tão crítica [Soares1992].

Com todas essas características fica claro que um sistema de comunicação que suporte todos esses requisitos deve prover mecanismos de negociação para :

- Máximo retardo de transferência;
- Variação máxima de retardo para o atendimento aos requisitos de especificação de áudio e vídeo isócronos, isto é, variação do retardo determinística;
- Mecanismos para compensação da variação estatística do retardo;
- Valor da vazão necessária para abertura de uma conexão;
- Para todos os tipos de erros, qual estratégia será adotada : detecção, detecção e correção ou nada;
- Controle de fluxo e congestionamento;
- Condições para fechamento de conexão, caso não seja possível atender a todos os requisitos;
- Taxa de erro de bit e pacote toleráveis.

Diante das características intrínsecas de cada recurso multimídia, será comentado a seguir o comportamento das redes atuais quando se transmite esses recursos.

### 2.3- LIMITAÇÕES DAS REDES EXISTENTES

A rede telefônica, adequada para transmissão de voz, deixa a desejar mesmo quando utilizada para transmissão de outros sinais de áudio. Quando pensamos na melhoria das redes telefônicas, não chegamos nem perto do desejado em termos de banda passante.

A rede telefônica opera por comutação de circuitos, isto é aloca-se um canal de banda passante constante durante toda a conexão, e esta não pode ser usada por outro canal, sendo assim, ineficiente em rede de tráfego de rajadas (textos, voz com detecção de silêncio, etc...).

Nas redes comutadas por pacotes, a informação a ser transmitida é dividida em pacotes. O compartilhamento do sistema de comunicação é por demanda (multiplexação estatística), sendo muito eficiente no transporte de tráfego por rajada, já que se a banda passante do sistema não está sendo utilizada, e houver demanda de outra transmissão, ela é imediatamente alocada a esta.

As *Wide Area Networks* (WANs) existentes hoje no Brasil operam a uma velocidade em torno de 2400 bps a 9600 bps. As *Local Area Networks* (LANs) operam em velocidades maiores, podendo suportar alguns poucos canais de voz, mas não o suficiente para permitir aplicações com um número grande de usuários.

Uma solução seria estender os padrões IEEE 802.3 (Institute of Electrical and Electronics Engineers) e IEEE 802.5 para trabalharem com maiores velocidades e atingirem maiores distâncias.

A eficiência  $E$  de uma rede em termos de vazão máxima é dada por :

$E = 1/(1+f(a))$ , onde  $a$  é o retardo máximo de propagação na rede, normalizado pelo tempo de duração de um pacote **Trans** [Soares1986, Hammond1986], que está mostrado na figura 2.2.

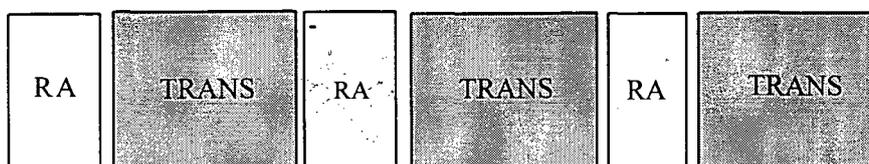


Figura 2.2 - O parâmetro "a" [Soares1992]

Na figura 2.2 :

**Trans** - Tempo de transmissão efetiva de um pacote;

**Ra** - Tempo em que o meio está sendo utilizado para a resolução de acesso.

Conforme a fórmula da eficiência de uma rede, mostrada anteriormente :

Se aumentar a velocidade (diminuir o tempo de transmissão do pacote) ou

aumentar a distância ( aumentar o retardo de propagação), então

o parâmetro  $a$  aumentará, diminuindo a eficiência ( $f(a) \sim a$ ).

Assim, estender os padrões de LANs para atuarem em maiores distâncias e maiores velocidades não faz sentido, pois a velocidade efetiva, a eficiência, ainda não satisfará à banda passante que as aplicações multimídias necessitam [Soares1992].

Duas estratégias estão sendo realizadas para o aumento da eficiência das redes baseadas em multimídia. A primeira é o aumento da largura de banda dos cabos e das ligações, onde aí aparecem as redes de alta velocidade sobre o meio físico de fibra ótica e a outra, é o desenvolvimento de melhores algoritmos de compressão de dados, como o MPEG.

## 2.4 - PROTOCOLOS MULTIMÍDIA EM REDES

A transmissão de recursos multimídia em redes de computadores requer o surgimento de novos protocolos de transporte e de rede para permitir a integração destes serviços com os já existentes, conforme mostra a figura 2.3. Como exemplo de protocolos em desenvolvimento, temos o RTC e o RTCP da Audio Visual Transport [Schulzrinne1992] e outros como o ST-II (Experimental Internet Stream Protocol : Version 2) cujo posicionamento na arquitetura INTERNET está mostrado na figura 2.4 [Chimiak1993].

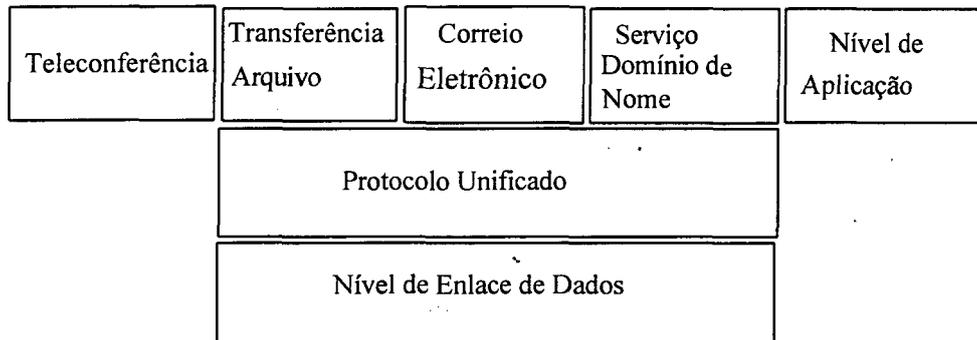


Figura 2.3 - Uma nova arquitetura Multimídia [Chimiak1993]

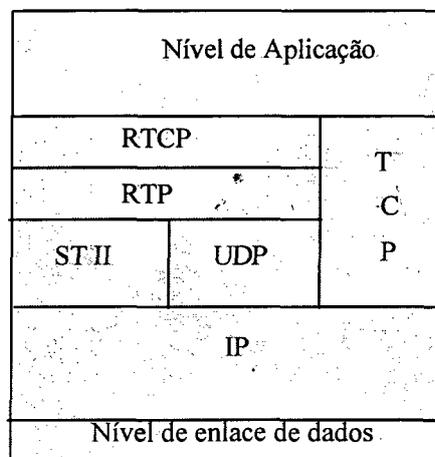


Figura 2.4 - Outra arquitetura multimídia integrada [Chimiak1993]

Protocolos INTERNET como Delta-t [Watson1983] , NETBLT [Clark1987] e Versatile Message Transaction Protocol (VMTP) [Cheriton1988] foram desenvolvidos, ajudando a surgir a Xpress Transfer Protocol (XTP) [XTP1992].

Um dos principais objetivos do XTP é a unificação dos protocolos de rede, de transporte e dos protocolos multimídia que são responsáveis pelo áudio, vídeo e seu sincronismo, como pode ser visto na figura 2.3.

O mecanismo de prioridade utilizado pelo XTP é o campo 32 bit SORT. O XTP somente oferece o mecanismo de prioridade, o controle de latência não é oferecido por si só. A combinação do mecanismo de prioridade com os mecanismos de controle de erro é que

possibilita o controle de latência e a sincronização, requeridos pelas transmissões multimídia em tempo real [Chimiak1993].

O reconhecimento seletivo, o rápido reconhecimento negativo e as retransmissões seletivas permitem que uma aplicação possa escolher um nível apropriado do controle da taxa de erro. Para um controle ótimo de sessão, é necessário haver múltiplas associações XTP.

O XTP foi criado para oferecer os requerimentos exigidos pelo nível de aplicação, isto é os requerimentos do QOS (Quality of Service).

Algumas experiências de aplicações multimídia sobre o XTP podem ser citadas [Adie 1993] :

- Na Universidade da Virgínia tem-se usado um demonstrativo de um correio eletrônico de imagens sobre o XTP/FDDI com padronização de compressão JPEG (Joint Picture Experts Group). Esta demonstração em tela inteira, suporta todas as cores, transmissão de 30 quadros por segundo de um disco na rede para um vídeo VGA remoto;
- Na Universidade técnica de Berlim, Alemanha, a coordenação, implementação e operação de serviços multimídia da R&D está utilizando o XTP como ponto de partida de seus projetos;
- No Comando naval americano, um serviço *multicast* é oferecido a um grupo específico de pessoas, esse serviço possui um tempo de latência em torno de 25 ms. Este tempo representa o atraso de um sinal de voz de um microfone até o sistema de áudio do palestrante;
- A Starlight Networks Inc. possui uma rede padrão Ethernet sobre XTP, com compressão de vídeo a uma velocidade de 1.2 Mbps, chegando a transmitir 10 quadros de vídeo por segundo.

No projeto desenvolvido não foi utilizado nenhum protocolo específico para tecnologia multimídia, já que os protocolos citados anteriormente, estão em fase de pesquisa e desenvolvimento. Na continuação do desenvolvimento do projeto, serão incorporados protocolos que se destinam especificamente para transmissão de multimídia em tempo real.

## Capítulo 3

### TECNOLOGIA FDDI

#### 3.1. INTRODUÇÃO

A arquitetura FDDI, uma arquitetura de redes de alta velocidade, é a arquitetura utilizada no projeto. Essa arquitetura utiliza como meio de transmissão a fibra ótica, que é um meio de transmissão rápido e confiável.

A utilização da luz como meio de comunicação e transmissão de informações é uma prática antiga. Lanternas piscando em meio a nevoeiros já sinalizavam perigos e avisos aos navegantes de séculos passados. A descoberta de que nada pode viajar mais rápido do que a luz no universo levou o homem a procurar dominá-la em seu benefício.

Assim como o desenvolvimento humano e tecnológico progride, a demanda por transmitir quantidades cada vez maiores de informações, a grandes velocidades e a longas distâncias, também aumenta e a tecnologia de fibra ótica surgiu para tornar isso uma realidade.

#### 3.2 - DESCRIÇÃO DA TECNOLOGIA DE FIBRA ÓTICA

A matéria-prima utilizada para a fabricação da fibra ótica é o cristal de quartzo, encontrado em abundância na crosta terrestre, particularmente no Brasil. Processos químicos transformam o quartzo num bastão que, "esticado", pode ser reduzido à espessura de um fio de cabelo.

A fibra ótica é um cilindro muito fino (de 50 a 150  $\mu\text{m}$ ) flexível e capaz de conduzir raios de luz. O uso de sílica pura dá a menor perda de potência ótica, sendo usada para maiores distâncias. Fibras de vidro têm um custo menor e um desempenho aceitável e as fibras de plástico são mais baratas ainda, mas seu desempenho é baixo.

As fibras óticas suportam transmissão para luz infravermelha (frequência entre  $10^{14}$  -  $10^{15}$  Hz) caracterizados como guias de onda luminosa. A variação dos índices de refração do núcleo e da casca é obtida através de dopagens diferentes.

A fibra ótica é constituída de um material cuja resistência ótica aumenta à medida que se afasta do seu eixo central. Assim, um raio de luz que caminhe exatamente pelo seu eixo central não sofre nenhuma refração. Os demais incidem sobre a parede reflexiva que reveste, internamente, a fibra ótica. Os raios que incidem em ângulos obtusos são refletidos e os que incidem em ângulos agudos são absorvidos.

Assim, pode-se notar que o raio de luz que caminha pelo eixo central chega ao destino antes que os raios de luz que sofreram várias reflexões durante o trajeto. Mas como são refletidos apenas os raios de luz que incidiram na parede reflexiva em ângulos obtusos, essa diferença é pequena [Ficap1992]. A figura 3.1 mostra o processo de transmissão da luz por fibra ótica.

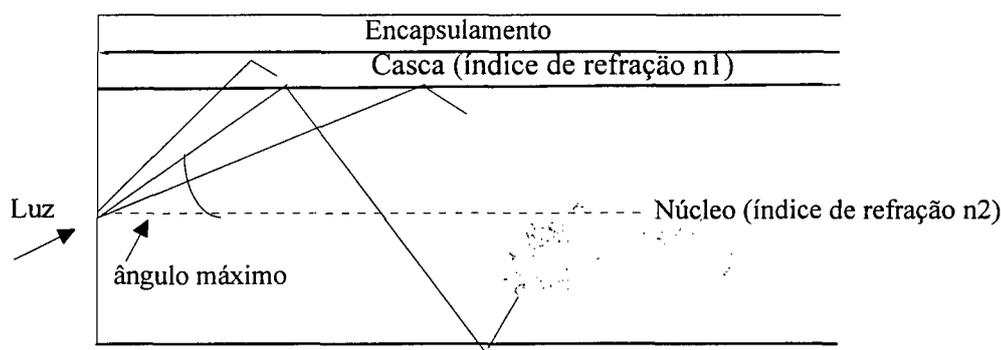


Figura 3.1 - Guia de onda luminosa - fibra ótica [Giozza1986]

Evidentemente, quanto maior for a distância, menos potente e mais difuso será o feixe de luz na saída da fibra ótica, o que dará exatamente, a limitação de distância de transmissão [Zakir1988].

Quanto maior a potência de entrada, maior será o comprimento da fibra ótica que ainda permite a leitura adequada do sinal. Quanto maior a taxa de transmissão, menor será o comprimento da fibra ótica [Zakir1988].

A capa isolante que envolve, externamente, as fibras óticas, previne a interferência de raios de luz externos além da proteção mecânica.

O sinal de luz é uma mistura de sinais eletromagnéticos de diferentes frequências. Quanto maior for o espectro de frequências do sinal de entrada, menor deve ser a taxa de transmissão e/ou menor será o comprimento da fibra ótica.

O acoplamento das fibras óticas com os dispositivos emissores de luz e os fotodetectores é bastante difícil devido às dimensões da fibra ótica serem microscópicas. As emendas também sofrem das mesmas restrições. Essas dificuldades fazem com que seja mais fácil a inserção de potência luminosa em uma única direção. Assim, a tecnologia de transmissão por fibra mais desenvolvida é a ponto-a-ponto. A realização de conexões multiponto com fibras óticas é limitada pela dificuldade em se obter conectores bidirecionais passivos (tipo T). A figura 3.2 mostra o sistema de transmissão ponto-a-ponto em fibras óticas.

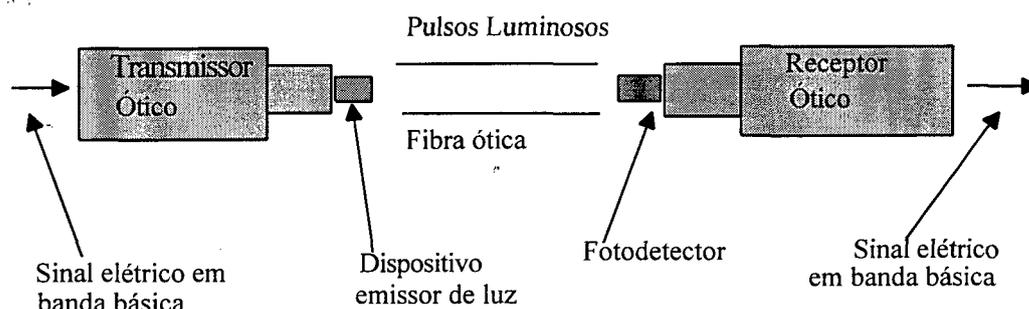


Figura 3.2 - Sistema de transmissão ponto-a-ponto em fibras óticas [Giozza1986]

Como a frequência da onda luminosa é muito alta e há grandes dificuldades e custos para se conseguir uma portadora de uma só frequência, usa-se a técnica ASK, que é o chaveamento por amplitude, para transmissão de sinais analógicos e a presença ou ausência de luz para a transmissão de sinais digitais. Está se pesquisando intensamente a possibilidade de fazer o equivalente ao *Frequency Division Modulation (FDM)*, que ficará *Wave Division Multiplexing (WDM)* quando falar-se em sinais luminosos [Hecht1993].

O desempenho das fibras óticas como suporte à transmissão digital é determinado por suas características de atenuação e dispersão na propagação dos pulsos de potência luminosa.

A atenuação em fibras óticas depende do comprimento de onda de luz utilizada na transmissão. A tecnologia atual de fibras óticas é caracterizada por espectros de baixa atenuação, em torno de  $0.8 \mu\text{m}$ ,  $1.3 \mu\text{m}$  e  $1.55 \mu\text{m}$ . Essas regiões são conhecidas por janelas de transmissão e correspondem à evolução dos sistemas de transmissão por fibra ótica [Giozza1986].

A técnica da fibra ótica foi desenvolvida nos anos 60 por Charles Kao, onde houve a purificação do material no seu estado natural e a atenuação foi diminuída de aproximadamente

1000 db/Km para 0.2 db/Km. Como comparação, pode ser mencionado que os cabos coaxiais, têm uma atenuação em torno de 5 db/Km [Giozza1986].

A quantidade de luz que não chega à extremidade oposta da fibra pode ter deixado a fibra ou ter sido absorvida (convertida em calor).

Os seguintes fatores contribuem para dispersão, ou seja, perda do sinal :

- Raios dispersos;
- Partículas não homogêneas;
- Irregularidade no diâmetro do núcleo;

Os seguintes fatores contribuem para a absorção :

- Raios infra-vermelho;
- Umidade, oxigênio;
- Íons metálicos;
- Danos no puxamento;
- Raios ultra-violetas [Ficap1992].

As perdas causadas por interferências eletromagnéticas não existem, já que a fibra ótica é imune a essas interferências.

Do ponto de vista de capacidade de transmissão, a tecnologia atual de fibras óticas é caracterizada por três tipos :

1. Multimodo com índice degrau - tem sua capacidade de transmissão limitada basicamente pela dispersão modal, isto é pelos diferentes tempos de propagação dos raios luminosos. A sua velocidade fica em torno de 15-25 MHz.Km. Esse é o tipo de fibra ótica utilizado no projeto. Apesar de não ser o tipo de fibra ótica ideal, é o mais viável economicamente.
2. Multimodo com índice gradual - são menos sensíveis à dispersão modal, acima citada, por isso possuem uma melhor capacidade de transmissão, que em média é de 400 MHz.Km.
3. Monomodo - insensíveis à dispersão modal, chegam a ter capacidade de transmissão da ordem de 100 GHz.Km, mas possuem o que chamamos de dispersão material ou cromática, que é causada pelo acoplamento das fibras com fontes luminosas pouco coerentes (policromáticas), tais como os diodos eletroluminescentes (*LED*). Uma maior capacidade de transmissão é conseguida com diodos laser. Nas fibras monomodo, o diâmetro do núcleo é tão pequeno que só existe um modo de propagação de luz.

Os mais importantes princípios para a propagação da luz nas fibras óticas são :

- Pureza do material que constitui o núcleo bastante elevada, a fim de que a atenuação seja mantida dentro de limites razoáveis;
- Feixes de luz, que por uma razão ou outra tendem a mudar sua direção de propagação, sejam mantidos dentro do núcleo da fibra.

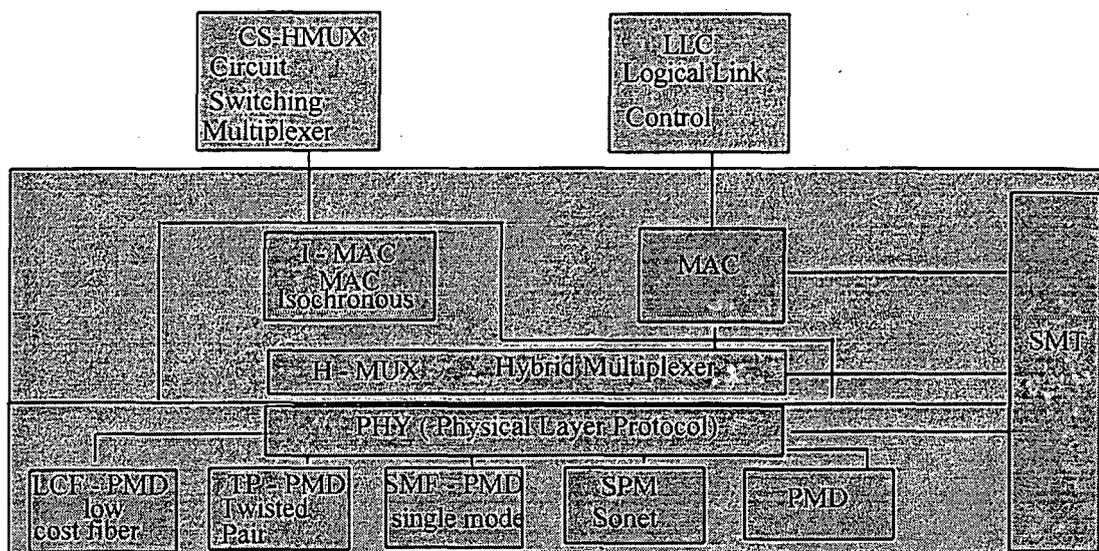
Uma certa perda por retro-espalhamento na fibra deve, por razões teóricas, não ser esquecida, é o que chamamos retro-espalhamento-de raios [Ficap1992].

Apesar de muito fina, a fibra ótica pode levar, na sua especificação convencional, oito vezes mais informações que o sistema atualmente em uso nas telecomunicações, isto significa que este meio de transmissão é ideal para a transmissão de recursos multimídia, que necessitam de uma grande largura de banda. A fragilidade física da fibra é contornada pelo tratamento com materiais especiais e pela superposição de materiais protetores. Os cabos submarinos de fibra ótica, depois de finalizados, têm a espessura de um dedo, mas guardam em seu interior um monumental poder de transmissão de todo tipo de informação, que chegam ao seu destino silenciosa e rapidamente [Ficap1992].

### **3.3 - ARQUITETURA FDDI**

A idéia de desenvolver uma interface de dados de alta velocidade baseada no uso de fibra ótica, começou a ser estudada por um grupo de trabalho da American National Standards Institute (ANSI) X3T9.5 num encontro em outubro de 1982.

A figura 3.3 mostra a arquitetura FDDI.



\* Figura 3.3 - Protocolos da arquitetura FDDI [Soares1992]

A escolha de fibra ótica como meio de transmissão era justificada pela alta taxa de transmissão : 100 Mbps. A escolha da arquitetura em anel era justificada pela facilidade de ligação ponto a ponto em fibra ótica e pelo excelente desempenho oferecido pelo anel com passagem de permissão.

A arquitetura FDDI caracteriza-se por ser uma rede em anel duplo usando fibra ótica como meio de transmissão de dados a uma taxa de 100 Mbps. Conforme o padrão PMD (Physical Layer Medium Dependent), a transmissão se faz com diodos emissores de luz (LED), transmitindo em um comprimento de onda nominal de 1,3  $\mu\text{m}$ .

A distância máxima entre dois repetidores adjacentes é de 2 quilômetros. No total do anel pode-se ter 1000 repetidores e a distância máxima do anel pode ser de até 200 Km.

Uma estação que pode conectar-se ao meio através da conexão aos dois anéis, são chamadas de estações classe A. As estações classe B são conectadas através de um simples cabo duplex diretamente ou por meio de um *hub*, concentrador, que se liga aos dois anéis.

Através da técnica de chaves de *bypass*, a estação se desconectará da rede caso falhe. A rede consiste em dois anéis em direções contrárias, com um dos anéis só funcionando em caso de falha da estação ou do enlace.

O PMD especifica o enlace de fibra ótica e os componentes óticos relacionados, incluindo os níveis de potência e características dos transmissores e receptores óticos, os requisitos de sinais de interface ótica e as taxas de erro permissíveis.

O PHY especifica os algoritmos de codificação/decodificação e de sincronismo de relógios e de quadros de dados.

O MAC especifica as regras de acesso ao meio, de endereçamento e verificação de dados. Possui dois componentes Hybrid-Multiplexer (HMUX) e Isochronous MAC (IMAC). O HMUX multiplexa os pacotes de dados do MAC e os dados isócronos do IMAC. O IMAC provê a interface entre FDDI e o serviço isócrono, representado pelo Circuit Switched Multiplexer (CS-HMUX).

O SMT especifica o controle requerido para operação apropriada das estações como: gerenciamento de configuração, gerenciamento de conexão e gerenciamento do anel.

O SMT monitora as atividades de controle do nível físico e de enlace de cada estação. A entidade SMT comunica-se com a outra entidade SMT na LAN para o controle das operações de redes.

### **3.4 - O CONTROLE DE ACESSO AO MEIO DA ARQUITETURA FDDI**

O protocolo FDDI possui três tipos de tráfego: síncrono, assíncrono restrito e assíncrono não restrito [Stallings1993].

No tráfego síncrono, os dados são transmitidos em intervalos regulares de tempo. O protocolo garante uma faixa passante e, também, um retardo de transferência limitado.

A banda passante não utilizada pelo tráfego síncrono é alocada para o tráfego assíncrono. Dois tipos de tráfego assíncrono são suportados pela rede : o restrito, no qual a banda passante assíncrona é usada por um número limitado de estações e o não restrito, no qual a banda passante assíncrona pode ser usada por todas as estações.

Quando da iniciação da rede, é negociado um parâmetro de protocolo chamado Target Token Rotation Time (TTRT). O TTRT deve ser baseado na metade do retardo máximo que cada aplicação de cada estação pode suportar, já que cada estação faz sua sugestão para o TTRT e o procedimento escolhe o menor valor estipulado. O protocolo também garante uma utilização do

anel igual à  $(TTRT - RL)/TTRT$ , onde  $RL$  é a latência do anel, tempo que uma permissão demora para circular o anel quando este não está submetido a qualquer carga. O protocolo também especifica um valor mínimo e máximo para o  $TTRT$ , pois um valor mal negociado do  $TTRT$  pode baixar o desempenho da rede.

Uma estação, ao capturar a permissão pode transmitir quadros da classe de serviço síncrona por um tempo negociado. A soma de toda a banda síncrona não pode ultrapassar o valor de  $TTRT - (D_{max} + F_{max} + Token\_time)$ , onde  $D_{max}$  é a latência máxima do anel,  $F_{max}$  é o tempo de transmissão de um quadro de tamanho máximo e  $Token\_time$  é o tempo de transmissão da permissão.

Para transmissão de quadros assíncronos, cada estação mantém um relógio chamado de Token Rotation Timer (TRT). O TRT é zerado toda vez que a permissão chega à estação tendo demorado um tempo menor que o  $TTRT$  desde a chegada anterior. Em caso contrário, TRT é zerado toda vez que atinge o valor  $TTRT$ . Uma permissão que chega à uma estação decorrido um tempo menor que  $TTRT$ , desde sua chegada anterior, permite a transmissão de quadros síncronos, pelo período negociado, e quadros assíncronos por um período igual à diferença entre o valor do  $TTRT$  e o valor de TRT. O valor do TRT é, então, carregado em um relógio chamado Token Holding Timer (THT). Se a permissão chega a uma estação decorrido um tempo maior que  $TTRT$ , então é permitida apenas a transmissão de mensagens classe síncrona.

A rede FDDI começa sua operação no modo assíncrono não restrito, que é o seu modo normal de funcionamento. A classe assíncrona não restrita pode possuir, opcionalmente, oito níveis de prioridade. A cada nível de prioridade  $i$  é associado um tempo  $T\_Pri(i)$ . Um quadro assíncrono de prioridade  $i$  só pode ser transmitido se  $THT < T\_Pri(i)$ . Caso as prioridades não sejam implementadas, quadros assíncronos podem ser transmitidos desde que  $THT < TTRT$ . Quando o THT estoura, as transmissões em progresso continuam até o fim.

Quando uma estação deseja criar um diálogo restrito, ela captura a permissão, transmite seus pacotes, e transmite a seguir uma permissão restrita, entrando no modo restrito. A estação de destino recebe os quadros e, ao capturar a permissão restrita, entra no modo restrito, começando a trocar quadros e permissões restritas com a origem, até que um quadro final de diálogo é transmitido, sendo seguido por uma permissão não restrita, terminando este modo de funcionamento. Só as estações no modo restrito podem transmitir quadros da classe assíncrona durante o período de permissões restritas. Uma vez que a permissão restrita é capturada, cada

estação no diálogo pode transmitir quadros assíncronos por um período alocado pelo protocolo SMT. Todas as estações podem transmitir quadros da classe de serviço síncrona. O suporte ao modo restrito é opcional.

Como podemos notar, o protocolo SMT garante para a classe síncrona um tempo médio de acesso máximo igual ao TTRT, um tempo máximo igual a duas vezes o TTRT e uma banda passante garantida, negociada com o SMT.

### 3.5 - FORMATO DOS QUADROS DA ARQUITETURA FDDI

Na arquitetura de protocolos FDDI, os dados são codificados para transmissão pelo protocolo PHY. Dos 32 símbolos gerados, 16 são símbolos de dados, 3 são usados para delimitação de início e fim de quadro, 2 são usados como indicadores de controle e 3 são usados na sinalização reconhecida pelo hardware PMD. Esta codificação vai exigir uma taxa de transmissão de 125 Mbaud, de forma a manter a taxa de dados de 100 Mbps.

A codificação usada para transmissão dos símbolos é a NRZI, onde o bit 1 é representado por uma transição e o 0 por não haver transição. Como nunca vai haver mais de três tempos de bits sem haver transição, o sincronismo é garantido.

O formato dos quadros de dados e da permissão podem ser vistos na figura 3.4.

10 símbolos 8 bytes PA	02 símbolos 1 byte SD	2 sím- bolos FC	6 (ou 2) bytes DA	6 (ou 2) bytes SA	< 4500 Bytes INFO	4 Bytes FCS	1 símbolo (1 byte) ED	3 símbolos FS
---------------------------------	--------------------------------	-----------------------	----------------------------	----------------------------	----------------------------	-------------------	--------------------------------	---------------------

A) Quadro de informação

PA	SD	FC	2 símbolos ED
----	----	----	---------------------

B) Permissão

Figura 3.4 - Formato dos quadros FDDI [Soares1992]

O preâmbulo precede cada transmissão e é usado para sincronismo entre transmissor e receptor. O delimitador de início de quadro (SD) consiste dos símbolos reservados JK. O campo FC define o tipo de quadro e sua característica. Ele distingue quadros síncronos de assíncronos, o comprimento do campo de endereço e o tipo de quadro: se de informação( vindo da camada *Logical Link Control* (LLC), de controle de acesso ao meio (MAC), de gerenciamento (SMT) ou se é de permissão. Um delimitador de fim de quadro (ED), dois símbolos delimitadores TT, completam o quadro da permissão.

Nos quadros de dados, os campos DA e SA representam os endereços de origem e destino, que podem ter dois formatos 16 ou 48 bits, sendo o bit mais à esquerda o bit que indica se o endereço é individual ou de um grupo, depois vem o endereço propriamente dito. No caso de endereço de 48 bits existe um bit indicando se é um endereço de administração local ou administração global. O FCS é um campo de 32 bits para detecção de erros através de cheque de redundância cíclica, usando um polinômio padrão *American National Standards Institute* (ANSI).

A estação de origem é quem remove os quadros do anel FDDI. A remoção é realizada quando o endereço de origem do quadro é igual ao da estação.

### **3.6 - ALOCAÇÃO DAS MÍDIAS AOS SERVIÇOS SÍNCRONO E ASSÍNCRONO DA ARQUITETURA FDDI**

A classe síncrona, por oferecer uma banda passante garantida, deve ser a preferida para a transmissão de dados contínuos. Mesmo para voz compactada, este tipo de classe de serviço deve ser o escolhido, pois pode-se garantir a banda passante quando da transmissão do surto de voz, além de se garantir um retardo máximo de transferência. A alocação da banda passante deve ser realizada pela taxa de pico, o que não diminui o desempenho da rede já que a banda passante não utilizada pode ser aproveitada pela transmissão assíncrona. A negociação do TTRT da rede deve levar em conta que é preciso manter, para a voz, um retardo menor que um certo limite [Soares1992].

Para sinais de vídeo e áudio compactado, e qualquer outra aplicação gráfica ou de texto em tempo real, a classe síncrona também é a mais adequada. Para os sinais que precisam manter

continuidade, como vídeo e áudio devemos compensar a variação estatística do retardo em algum nível superior de protocolo. Deve-se notar que o número de conexões que podem ser abertas deve ser menor do que no caso em que se pode transmitir tudo em classe assíncrona, reservando a classe síncrona só para o tráfego contínuo [Soares1992].

Conforme foi mostrado no capítulo 2, o áudio voz e o vídeo admitem um certo percentual de perdas. Neste caso, a classe assíncrona parece ser mais adequada, pois consegue-se assim, um maior número de conexões.

No caso de mídias com tráfego de rajadas sem restrições de tempo, o mais indicado é a classe de serviços assíncrona.

Nos casos de sinais contínuos a escolha deve recair sobre a classe síncrona, embora seja necessário realizar a compensação da variação estatística de retardo.

Para se obter uma variação de retardo determinístico, utiliza-se um outro serviço, chamado isócrona, da proposta FDDI II, que é descrita na próxima seção.

### 3.7 - FDDI II

FDDI II é uma rede com a mesma arquitetura física da FDDI, oferecendo os mesmos serviços, acrescidos do serviço isócrona. A rede possui 16 canais de banda larga que podem ser alocados como isócronos, usando no máximo 98.304 Mbps. Um canal residual não isócrona de capacidade de 1 Mbps permanece, para uso de serviços síncronos e assíncronos. Canais de banda larga podem ser alocados e desalocados em tempo real.

A rede FDDI II pode operar em modo básico e híbrido. No modo básico, somente o serviço de comutação de pacotes é permitido. No modo híbrido, ambos serviços de comutação de pacote e circuito são possíveis. A rede FDDI II tipicamente começa no modo básico para configurar os temporizadores e os parâmetros necessários. A presença na rede de qualquer estação que não opere o FDDI II impede a alocação de canais isócronos, ativando assim o modo FDDI.

A capacidade de chaveamento de circuito é imposta ao anel por um ciclo que se repete a cada 125 ms, criado por uma estação denominada Mestre do Ciclo (*Cycle Master*). A rede é iniciada no modo não isócrona e só muda para o modo híbrido depois que uma estação negocia e ganha o direito de ser o nó mestre do ciclo.

O formato do ciclo é o seguinte : um campo de preâmbulo (PA) com símbolos IDLE, seguido de um *header* formado pelo delimitador (SD) com símbolos JK. Os doze octetos do *cycle header* definem como o resto do ciclo é utilizado. O *Dedicated Packet Group* (DPG ) consiste de 12 octetos que são utilizados para o controle do token na comutação por pacote e, finalmente, tem-se os *Wideband Channels* (WBCs) onde cada um consiste em 96 octetos por ciclo.

Cada WBC provê uma capacidade de transferência de 6.144 Mbps ( 96 octetos por ciclo x 8 bits/octeto x 8000 ciclos/segundo ). Se o canal é utilizado para comutação de pacotes, todos os WBCs são agrupados para formar um canal único maior, dedicado somente a essa comutação de pacotes. A capacidade mínima em uma comutação por pacotes é de 768 Kbps, podendo chegar a 6.144 Mbps e no máximo a 99.072 Mbps.

O subnível IMAC determina quais WBCs serão utilizados na comutação por circuitos. Alternativamente, o IMAC pode dividir um WBC em vários subcanais permitindo um diálogo simultâneo, independente e isócrono entre os diferentes pares de estações do FDDI II. O *header* do ciclo FDDI II está representado na figura 3.5.

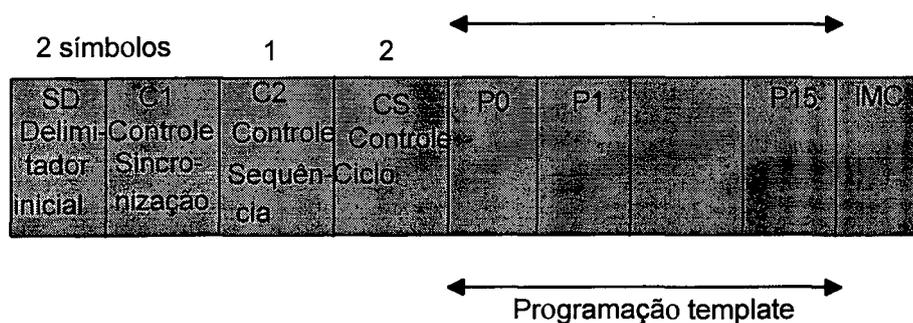


Figura 3.5 - Header do ciclo FDDI II [Stallings1993]

O campo SD indica o começo de um ciclo. O campo C1 estabelece o estado de sincronismo; S sem sincronismo e R com sincronismo. O campo C2 indica o *status* de sequenciamento do ciclo; o R indica que o sequenciamento do ciclo não foi estabelecido e S que foi estabelecido e finalmente o campo CS consiste no próprio controle de sequenciamento.

Se o C1 e o C2 contêm o valor R então o campo CS é interpretado como contendo a monitoração do *rank*, isto é a classificação . A estação com o maior *rank* torna-se a monitora do

ciclo. Quando o C1 e o C2 possuem o valor S então o campo CS conterá um valor entre 64 e 255 sendo incrementado a cada novo ciclo.

O campo de programação de *templates* ( P0 a P15 ) consiste de 16 símbolos, um para cada WBC. O valor R indica que o WBC correspondente é um canal para comutação de pacotes enquanto que o S indica que o WBC é dedicado ao tráfego isócrono.

O campo Isochronous Maintenance Channel (IMC) é de uso fora do escopo do FDDI II.

Duas estações podem se comunicar usando a comutação de circuitos através do compartilhamento de um canal isócrono dedicado ou podem se comunicar utilizando a comutação de pacotes sobre um canal de pacote de dados.

Um quadro ou permissão dentro do canal não isócrono tem o mesmo formato FDDI, exceto que o delimitador de início não usa o símbolo JK e sim os símbolos HALT e IDLE.

O protocolo I-MAC controla os canais isócronos e o protocolo H-MUX cuida da multiplexação dos serviços isócronos e não isócronos para entrega ao nível físico.

A alocação de um canal isócrono é feita por troca de mensagens no canal não isócrono. O protocolo SMT permite que as estações designadas como alocadoras de ciclo requeiram do mestre de ciclo a responsabilidade de gerenciamento de uma ou mais WBCs para aplicações isócronas. Ao receber a requisição, o mestre de ciclo espera pela permissão, transmite um novo cabeçalho de ciclo, contendo o novo gabarito, e avisa à estação alocadora de ciclo requisitante qual a identidade da WBC que ela adquiriu responsabilidade. As outras estações são avisadas sobre a banda passante do canal isócrono disponível, podendo assim requisitar banda passante de 8 , 16 , 32, 64 , 384, 1536, 1544, 1920, 2048 ou 6144 Kbps.

O protocolo SMT é usado também pelas estações para requisitar canais à estação alocadora de ciclo, que deve confirmar o pedido enviando à estação requisitante a identidade do canal alocado, ou então a recusa ao pedido de alocação e sua justificativa. Ao final da comunicação, a estação deve avisar a estação alocadora de ciclo, através do protocolo SMT, a liberação do canal.

No FDDI II, ao invés de se utilizar endereçamentos nos quadros, uma conexão é pré-estabelecida através de negociações utilizando pacotes de mensagens ou é estabelecida por uma convenção conhecida por ambas as estações.

### 3.8 - ALOCAÇÃO DAS MÍDIAS AO SERVIÇO ISÓCRONO DA ARQUITETURA FDDI

O serviço isócrono oferece um retardo de transferência constante, sendo o mais indicado para transmissão de dados contínuos. No entanto, se uma WBC não estiver sendo utilizada, ela não pode ser compartilhada por outros serviços não isócronos.

Um problema notado é que se tivermos, por exemplo, seis canais de 64 Kbits no byte 3,5,23,44,54 e 66 e for feita uma requisição de um canal de 384 Mbps, este canal não pode ser alocado pois não existem seis bytes contíguos na WBC, afetando com isso o desempenho da rede. A aceitação de canais com bits não contíguos exigiria uma manutenção de mapas de canais extremamente complexa [Soares1992].

Para mídias tais como áudio, onde o retardo de transmissão é importante e as perdas são suportáveis, o serviço isócrono vem trazer boas perspectivas, defendendo um serviço orientado à conexão e sem controle de erros. Agora para serviços como tráfegos em rajada, como voz com detecção de silêncio ou tráfegos sem interrupção mas com taxas variáveis, como vídeo compactado, cuja alocação do canal deve ser feita utilizando a taxa de pico, o serviço isócrono tem grandes desvantagens já que a banda passante não utilizada dos canais não pode ser utilizada pelos tráfegos não isócronos, ou mesmo por outros canais isócronos [Soares1992].

### 3.9 - FDDI III

FDDI follow-on é às vezes chamada de FDDI III ou Enhanced Performance and Services LAN (EPSL).

A FDDI III deve fornecer o serviço de chaveamento de circuito, bem como o serviço assíncrono e o modo de transferência assíncrona (ATM). A velocidade da rede deve ser compatível com a Synchronous Digital Hierarchy (SDH), 155.52 Mbps, 622.08 Mbps e 2.4 Gbps e deve utilizar o mesmo cabeamento FDDI. A definição completa do FDDI III está prevista para ser entregue em dezembro de 1995.

No projeto desenvolvido foi utilizada a arquitetura FDDI. Deve-se observar que a arquitetura apropriada para o desenvolvimento desse projeto é a FDDI II, já que esta arquitetura possui a facilidade do serviço isócrono, que possibilita a determinação da variação estatística do

retardo, muito importante para a transmissão síncrona de recursos multimídia. A escolha da arquitetura FDDI se deu porque esta arquitetura já se encontrava implantada no Campus da UFSC, e não tinha-se recursos para investimento na arquitetura FDDI II.

## Capítulo 4

### ARQUITETURA WINDOWS NT SERVER 3.5 / FDDI DO PROJETO

#### 4.1 - INTRODUÇÃO

Na figura 4.1 mostra-se a arquitetura Windows NT Server 3.5 / FDDI que está sendo utilizada no projeto.

O sistema operacional Windows NT Server 3.5 pode ser executado em sistemas com Intel x86, Reduced Instruction Set Computing (RISC) e DEC Alpha. Permite o multiprocessamento simétrico, onde pode-se adicionar mais processadores se for necessário ter-se melhor performance [Hecht1993].

A partir deste momento, quando for citado somente Windows NT, está se falando do Windows NT Server 3.5, isto foi feito para facilitar a leitura do trabalho.

Sua arquitetura interna é inteiramente de 32 bits. Com isso, os segmentos de memória passam dos 64 Kbits e ultrapassa-se a barreira dos 640 Kbits do sistema operacional MS-DOS. Oferece proteção de memória, já que o espaço reservado de memória para o sistema operacional é separado do espaço reservado para as aplicações. Com sua característica de multitarefa preemptiva, o sistema operacional aloca o tempo de processamento para cada aplicação da maneira mais eficiente. Além de que, a forma da multitarefa do Windows NT é baseada em *threads*. O conceito de *thread* é uma unidade de código executável. Assim, um processo Windows NT pode ter várias threads que podem ser executadas simultaneamente.

O Windows NT suporta o padrão Network Interface Specification (NDIS) e o Transport Driver Interface (TDI) [Microsoft1993].

O NDIS permite que se utilize vários tipos de placas de rede e protocolos sem precisar que se tenha uma versão diferente do protocolo de rede para cada tipo de placa. Quatro protocolos vêm embutidos no Windows NT Advanced Server : (1) NetBIOS Extended User Interface (NetBEUI); (2) Transmission Control Protocol / Internet Protocol (TCP/IP); (3) Microsoft NWLINK e (4) Data Link Control.

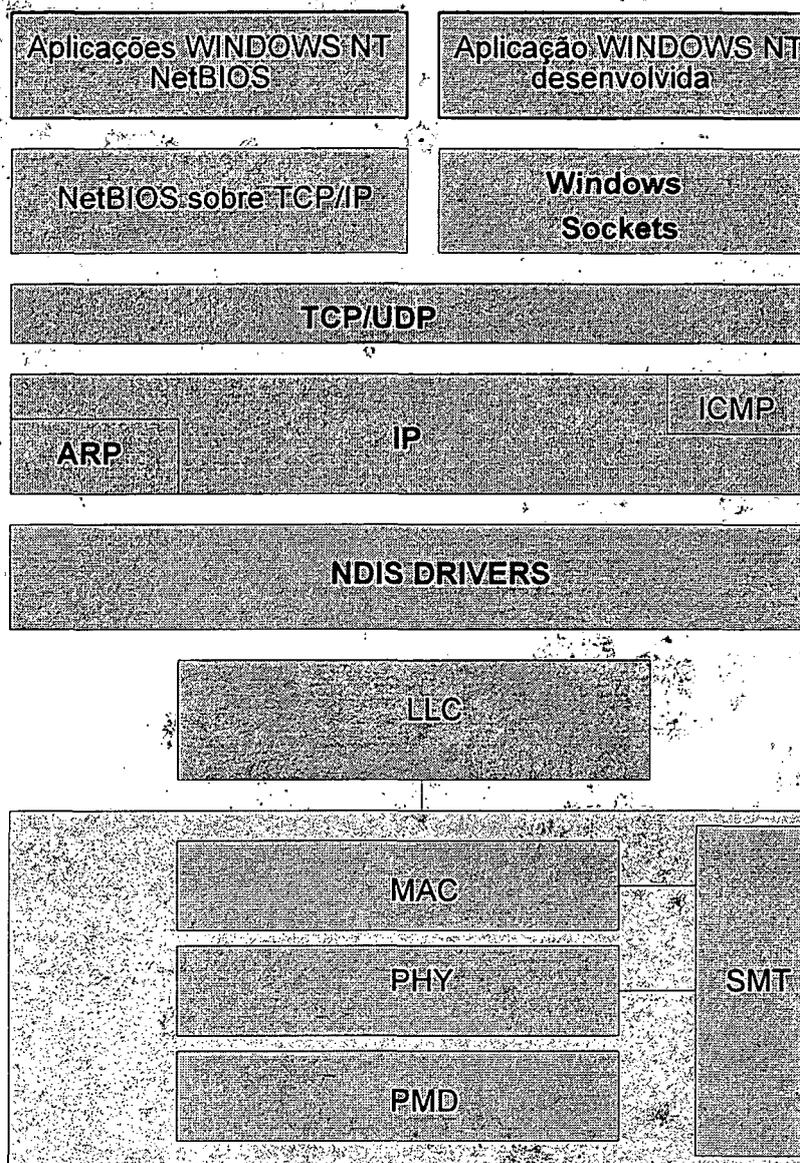


Figura 4.1 - Arquiteura Windows NT/FDDI

O TDI realiza a interface entre protocolo de rede e os níveis de rede mais acima. Assim, aplicações *Windows Sockets*, aplicações *Remote Procedure Call (RPC)* e aplicações *NetBIOS* não precisam ter uma interface para se comunicar com os protocolos de comunicação.

O *Windows NT Advanced Server* também suporta uma versão do *STREAMS* desenvolvido no sistema operacional *UNIX System V* versão 4.0. Isto significa que os protocolos originalmente desenvolvidos para *STREAMS* no *UNIX* podem ser facilmente portados para o *Windows NT*.

## 4.2 - PADRÃO NDIS

O padrão NDIS provê um conjunto de interfaces para que os protocolos e os *drivers* das placas de rede possam se comunicar. Nesse projeto os adaptadores de rede utilizados são o FDDI 5341 da Sysknect [Sysknect1993].

Quando utiliza-se múltiplos protocolos, a transmissão é realizada primeiro pelo protocolo primário e depois transmite-se nos outros protocolos.

No Windows NT cada ligação ou *binding* de um protocolo com a placa de rede, possui um número de adaptador de LAN associado a eles. O NDIS permite que se ligue dois ou mais protocolos a somente uma placa de rede.

## 4.3 - PROTOCOLO NetBEUI

O NetBIOS Extended User Interface (NetBEUI) foi desenvolvido na IBM em 1985. Quando este protocolo foi desenvolvido pensava-se que as LANs comportariam no máximo 200 computadores e seriam utilizados *gateways* para fazer-se a conexão a outras LANs ou a *mainframes*. Sendo assim, este protocolo possui um ótimo desempenho quando é utilizado em segmentos de redes locais, não sendo indicada sua utilização em redes de longa distância [Microsoft1993].

A versão do NetBEUI que acompanha o Windows NT é o NetBEUI 3.0. Na realidade, o NetBEUI 3.0 não é um verdadeiro NetBEUI porque ele possui adaptações para se compatibilizar com o TDI.

Como já foi citado, como o NetBEUI possui baixa performance quando se transmite em redes de longa distância, é recomendado que se utilize o NetBEUI e outro protocolo, como o TCP/IP, para acessar outros computadores através de um roteador ou de uma rede de longa distância.

## 4.4 - NetBIOS SOBRE WINDOWS NT

O NetBIOS sobre o Windows NT (NBT) permite que aplicações que estabelecem conexão para se comunicar, ou seja, orientadas à conexão e que utilizam o modo de resolução

por nome para encontrar o nó destinatário, se comuniquem com o TCP/IP, que utiliza endereços IP para encontrar o nó destinatário.

#### 4.5 - PROTOCOLO TCP/IP

O protocolo TCP/IP foi desenvolvido em 1970 no projeto de interconexão de redes do Defense Advanced Research Projects Agency (DARPA) [Microsoft1993].

A grande vantagem do TCP/IP é que este protocolo provê comunicação através de redes interconectadas com diferentes sistemas operacionais e arquiteturas de hardwares, como o UNIX, mainframes e o Windows NT.

O TCP/IP possibilita a utilização do protocolo de gerência de redes Simple Network Management Protocol (SNMP), que especifica um padrão para coleta de dados de gerenciamento de uma rede.

#### 4.6 TCP/IP SOBRE O WINDOWS NT

O protocolo TCP/IP é um padrão que estabelece como os dados são transmitidos entre os sistemas em uma rede de computadores. O protocolo TCP/IP sobre o Windows NT permite a conectividade entre computadores que utilizam o Windows NT como sistema operacional.

Apesar do protocolo TCP/IP ser na realidade dois protocolos, estes têm sido citados como um único protocolo da família de protocolos Internet.

O TCP é um protocolo de transporte orientado a conexão, o que significa que primeiro deve ser estabelecida uma conexão entre os dois sistemas que desejam transmitir os dados. Quando uma aplicação envia uma mensagem para o TCP, este divide a mensagem em pacotes com tamanhos apropriados, e os repassa para a camada IP a fim de que sejam enviados pela rede. O protocolo TCP designa um número sequencial para cada pacote, isto permite que o receptor reordene os pacotes na forma original. Também há uma verificação quanto a integridade dos dados através de uma operação matemática (*checksum*) aplicada pelo transmissor da mensagem. O receptor da mensagem realiza os mesmos cálculos e compara o resultado com o valor do *checksum* enviado pela origem. Se o resultado da comparação está correto, o receptor envia uma mensagem ACK (*acknowledgment*) para confirmar a recepção correta dos dados. Caso contrário,

o transmissor retransmite o pacote, já que não houve o recebimento do *acknowledgment*. O protocolo TCP utiliza as identificações das portas de comunicação (*port ID*) para especificar qual aplicação está enviando ou recebendo os dados. O *port ID*, o *checksum* e os números sequencias são inseridos nos pacotes TCP nos *headers* dos pacotes.

O IP basicamente endereça e envia os pacotes. O protocolo IP utiliza três informações para receber e enviar os pacotes de modo satisfatório : endereço IP, *subnet mask* e *default gateway*.

O endereço IP identifica o sistema dentro de uma rede TCP/IP. Este endereço ocupa 32 bits e é globalmente único em uma rede. Ele é subdividido em duas partes [Microsoft1993] :

- Identificador da rede : identifica um grupo de sistemas que estejam localizados logicamente na mesma rede;
- Identificador do *host* : identifica o sistema do usuário .

O *subnet mask* é utilizado para extrair estes dois últimos ítems citados de um endereço IP. O valor do *subnet mask* é determinado designando o valor do identificador da rede para 1's e o identificador do *host* para 0's.

O *default gateway* é utilizado somente quando utiliza-se sistemas que fazem parte da rede Internet. Quando um pacote é enviado ao receptor é verificado se o endereço de destino é compatível com o endereço de origem. Se é compatível, o pacote é enviado diretamente ao destino na rede local. Se o endereço não é compatível, o pacote é enviado para o *default gateway*. O pacote é enviado para outros *gateways* até encontrar um *gateway* que contenha o endereço especificado. Esse procedimento é o que é chamado de roteamento.

Existem dois outros protocolos não diretamente relacionados com o transporte dos dados das aplicações e do usuário, mas que pertencem ao protocolo IP. Estes são : (1) *Address Resolution Protocol* (ARP) e (2) *Internet Control Message Protocol* (ICMP) [Microsoft1993].

Apesar dos pacotes IP possuírem os endereços IP de origem e de destino, o endereço do *hardware* também deve ser conhecido. O protocolo IP adquire o endereço do *hardware* através do envio, por difusão, de um pacote especial, chamado ARP, que contém o endereço IP e de *hardware* do nó que deseja se comunicar e o endereço IP do nó destino. Todos os nós detectam esse pacote e o nó que contém o endereço IP em questão, responde enviando seu endereço de *hardware* . O mapeamento desse endereço é armazenado no nó que enviou o pacote ARP para posterior utilização.

O ICMP permite que dois nós que estejam em uma rede, possam compartilhar informações de erro e de estado do protocolo IP. Esta informação pode ser utilizada pelos protocolos dos níveis mais altos para recuperação de problemas de transmissão e para detecção de problemas na rede.

#### 4.7 - PADRÃO TDI

O TDI faz com que diferentes protocolos de rede se comuniquem com os níveis de rede mais acima utilizando uma interface comum. Assim, aplicações NetBIOS, aplicações Windows Sockets e aplicações RPC podem facilmente se comunicar com os protocolos de rede, não necessitando haver uma comunicação direta entre estas aplicações e os protocolos de comunicação.

#### 4.8 - RESOLUÇÃO DE NOME DO NetBIOS

Duas tarefas são requeridas quando utiliza-se o sistema operacional Windows NT com o protocolo TCP/IP [Microsoft1993] :

- Registro e atualização dos nomes NetBIOS;
- Descoberta dos nomes NetBIOS.

Quando há a configuração do sistema operacional Windows NT com o protocolo TCP/IP é necessário ser especificado o nome do computador, que é o nome NetBIOS e o endereço IP, que é o endereço ao qual todos os nós de uma rede reconhecerão a estação ou servidor.

Quando o NBT é inicializado, um pacote NetBIOS chamado *name registration request* é enviado a todas as estações informando o nome NetBIOS e o endereço IP sugerido pelo nó. Todas as estações que anteriormente solicitaram o nome NetBIOS devem responder com um pacote *negative name response*, caso não concorde com o nome NetBIOS sugerido. Não havendo nenhuma resposta negativa à solicitação, o nó adota o nome e o endereço sugerido. Sucessivos registros de nome NetBIOS são procedidos da mesma maneira.

O TCP/IP reconhece endereços IP e o Windows NT reconhece nomes, então para um nó suportando o sistema operacional Windows NT obter os endereços IP dos nós que deseja se comunicar, deve utilizar os serviços do NBT. Essas informações são obtidas durante o processo

de *b-node NetBIOS discovery*. Um pacote contendo o nome do computador destino é enviado a todos os nós. O nó que contém o nome correspondente, responde com um pacote *positive name query response* contendo seu endereço IP, assim, os nós podem se comunicar.

Em situações onde as sub-redes são conectadas através de um roteador IP, os pacotes podem não ser recebidos por um nó que esteja executando o sistema operacional Windows NT e o protocolo TCP/IP, fazendo com que a comunicação não seja possível. O arquivo LMHOSTS pode ser utilizado para especificar o mapeamento do nome NetBIOS e o endereço IP. Quando uma resolução de nome tem que ser realizada, o NBT verifica esse arquivo antes de executar o *b-node name discovery*.

## 4.9 - TRANSMISSÃO DE PACOTES IP E ARP SOBRE UMA REDE FDDI

### 4.9.1 - FORMATO DO PACOTE

Os pacotes IP e as requisições e respostas ARP enviadas em uma rede FDDI devem ser encapsulados conforme o padrão IEEE 802.2 LLC tipo 1 não numerado, com os campos DSAP e SSAP do header 802 contendo o valor global SAP do SNAP (Sub-Network Access Protocol) [IEEE1989]. Os primeiros 24 bits no SNAP devem ser zero e os 16 bits restantes são o EtherType [Reynolds1990] (IP=2048 e ARP = 2054). Os pacotes também devem estar de acordo com o nível FDDI MAC e o nível físico. Essa arquitetura pode ser observada na figura 4.2.

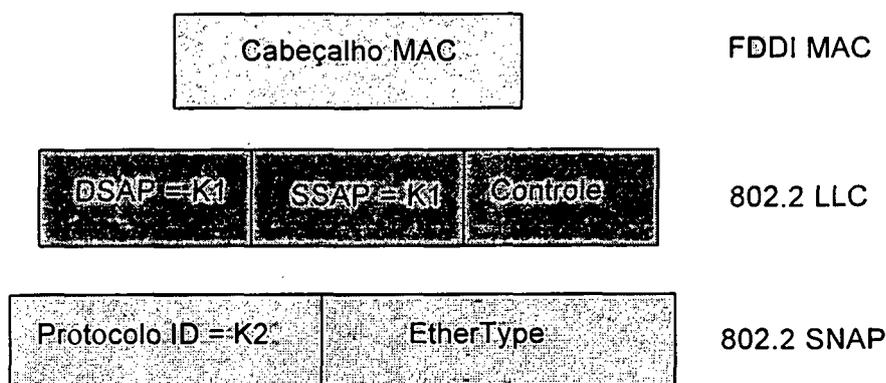


Figura 4.2 - Formato do pacote TCP/IP sobre FDDI

O tamanho do cabeçalho LLC e do cabeçalho SNAP é 8 octetos.

O valor de K1 é 170 (decimal), o valor de K2 é 0 (decimal) e o valor do controle é 3 (informação não numerada).

#### 4.9.2 - RESOLUÇÃO DE ENDEREÇO

O mapeamento dos 32 bits Internet para os 48 bits do endereço FDDI deve ser feito através do ARP.

Os endereços Internet são atribuídos arbitrariamente nas redês Internet. Cada *host* deve conhecer seu próprio endereço Internet e responder para o ARP apropriadamente. O ARP também é utilizado para traduzir um endereço Internet para um endereço FDDI quando for necessário.

O ARP possui campos que parametrizam sua utilização em vários contextos específicos. Esses campos são :

- hrd 16 bits - código do tipo de hardware;
- pro 16 bits - código do tipo de protocolo;
- hln 8 bits - octetos em cada endereço de hardware;
- pln 8 bits - octetos em cada endereço de protocolo;
- op 16 bits - código de operação.

O código de tipo de hardware para a rede FDDI é 6 [Reynolds1990]. O código para o tipo de hardware para redes Ethernet é 1. Infelizmente, diferentes valores entre Ethernet e FDDI fazem com que haja problemas de interoperabilidade nos ambientes interligados por bridges. Em muitas ocasiões os pacotes ARP são enviados com o código de tipo de hardware igual a 1.

O código do tipo de protocolo é 2048 [Reynolds1990]. O tamanho do endereço de hardware é 6. O tamanho do endereço de protocolo é 4 (endereço IP) e o código de operação é 1 para requisição e 2 para resposta.

Para se manter a interoperabilidade, já que o endereço FDDI normalmente é representado com o bit de grupo na posição de mais alta ordem dos bits, é realizada uma conversão para a ordem canônica, onde o bit de grupo fica posicionado no bit de mais baixa ordem.

### 4.9.3 - ENDEREÇO *BROADCAST*

O endereço *broadcast* Internet deve ser mapeado para o endereço broadcast FDDI [Braden1987].

### 4.9.4 - SUPORTE *MULTICAST*

O método de suporte ao *multicasting* pode ser encontrado no [Deering1989]. Neste trabalho não serão descritos muitos detalhes a respeito do suporte *multicasting*, pois nos testes realizados no projeto foram utilizadas apenas duas estações.

A utilização deste método deve possibilitar a cópia dos frames endereçados a cada um dos endereços arbitrários do *multicasting*. O endereço IP multicast é mapeado para o endereço FDDI colocando-se os 23 bits de mais baixa ordem do endereço IP nos 23 bits de mais baixa ordem do endereço de grupo (ordem canônica).

### 4.9.5 - FORMATO *TRAILER*

Os hosts diretamente conectados às redes FDDI não devem utilizar *trailer*.

### 4.9.6 - ORDENAÇÃO DOS BYTES

Os pacotes IP são transmitidos em redes FDDI como uma série de bytes de 8 bits. A ordenação da transmissão dos bytes é chamada de '*big-endian*' [Cohen1981].

### 4.9.7 - DETALHES SOBRE O NÍVEL MAC

A especificação FDDI MAC [ISO1989] define o tamanho máximo do pacote em 9000 símbolos (4500 octetos), incluindo os 4 símbolos (2 octetos) do preâmbulo.

São utilizados 4352 octetos para os dados, sendo 4096 octetos para os dados propriamente ditos e 256 octetos para o cabeçalho para o nível de rede e outros níveis acima. Os 148 octetos restantes são deixados como reserva.

Os *gateways* devem estar habilitados a aceitar pacotes no tamanho máximo dos pacotes FDDI, devendo fragmentá-los quando necessário.

As estações devem trabalhar com pacotes até o máximo permitido pelos pacotes FDDI. A preferência quanto ao tamanho do pacote deve ser comunicada ao nível TCP, através da opção *TCP Maximum Segment Size* [Postel1983].

Os datagramas nas redes FDDI não podem ser maiores que o tamanho máximo permitido pelo padrão Internet, 576 octetos.

Todos os frames IP e ARP devem ser transmitidos como frames assíncronos no modo não restrito. É recomendável que as implementações permitam a recepção e transmissão de frames IP e ARP no modo síncrono, assim como no modo assíncrono restrito.

Todas implementações FDDI devem suportar o padrão IEEE 802.2 Classe 1 para que haja compatibilidade com o padrão IEEE 802.2.

O padrão 802.2 classe 1 suporta comandos Unnumbered Information (UI), comandos e respostas eXchange IDentification (XID) e comandos e respostas TEST link (TEST) [IEEE1985]. As estações não necessitam estar aptas a transmitir comandos XID e TEST, mas necessitam transmitir respostas XID e TEST [Katz1990].

## Capítulo **5**

### **PADRÃO MPEG DE COMPRESSÃO**

#### **5.1 - INTRODUÇÃO**

A transmissão de vídeo é bem mais complexa do que a transmissão de sinais de voz ou dados. Os antigos telefones de voz analógicos simplesmente convertem as variações contínuas de intensidade sonora da voz em variações contínuas de um sinal elétrico. Os sinais de dados são dados binários : 1s e 0s. Os sinais de vídeo têm que codificar imagens continuamente mutáveis e acompanhadas de som.

Há quatorze padrões para televisão preto e branco, conhecidos pelas letras A,B até N. Ao mesmo tempo, há três grandes grandes padrões de televisão em cores : os padrões NTSC (National Television Systems Committee) encontrado nos Estados Unidos, Canadá, Japão, Chile e outros, o PAL (Phase Alternation by Line) encontrado no Brasil, Alemanha, a maior parte dos países da Europa Ocidental (menos a França), Austrália, Nova Zelândia, Argentina e vários outros e o SECAM (Séquentiel à Mémoire) encontrado na França, URSS, países da Europa Oriental, Oriente Médio e África [Chaves1991].

A frequência da corrente elétrica no Brasil é, como nos Estados Unidos, de 60 Hz. Por essa e outras razões, o sistema PAL-M brasileiro é, idêntico ao sistema NTSC-M americano, somente diferindo nos padrões de cores. O número de linhas desses padrões é 525, frequência de campo, ou seja número de linhas pares ou ímpares, de 60 por segundo, frequência horizontal (15.750 Hz), faixa de vídeo (4,2 MHz) e canal de vídeo (6MHz).

#### **5.2 - DESCRIÇÃO DOS SISTEMAS DE TELEVISÃO E VÍDEO**

Uma imagem nada mais é do que um arranjo de pontos chamados de *pixels* (*picture elements*). A câmera de televisão ou vídeo, para captar uma imagem, a divide em vários pontos

minúsculos e realiza um *raster scanning*, ou varredura linear da imagem, da esquerda para direita, de cima para baixo.

Uma câmera de televisão ou de vídeo, ao captar uma sequência de imagens, detecta as diferentes quantidades de luz e sombra que incidem sobre cada ponto. Um transdutor fotoelétrico converte as variações de luminosidade em variações de corrente elétrica, gerando um sinal que determina o grau de luminosidade, ou sinal Y.

Nas imagens coloridas, as três cores básicas são codificadas em um só sinal, que indica a matiz, ou tonalidade da cor e a sua saturação, ou intensidade. Esse sinal é chamado de crominância. Nas imagens preto e branco, as cores são as várias tonalidades de cinzas, limitadas, nas extremidades, pelo branco e pelo preto.

Os sinais de televisão e vídeo a cores são chamados de sinais compostos, já que combinam o sinal de luminância e crominância com a geração de pulsos de sincronização horizontal e vertical.

Quando o sinal é recebido no destino, o sinal de crominância é separado do sinal de luminância, nos seus elementos vermelho, verde e azul. É evidente que, nas imagens preto e branco, o sinal de crominância é ignorado. As informações de luminância, crominância e os pulsos de sincronização são levados ao tubo de imagem do aparelho de televisão ou monitor de vídeo de computador.

O que permite que os vários pontos exibidos sequencialmente na tela sejam percebidos como uma só imagem é a persistência visual. Bernard Grob [Chaves1991] disse : "A impressão causada nos olhos por uma luz persiste ainda por uma fração de segundo após a remoção da fonte de luz. Assim, se várias vistas são mostradas aos olhos durante esse intervalo de persistência visual, elas serão integradas pelo olho e o observador terá a impressão de estar vendo todas as imagens simultaneamente. É esta persistência que torna possível em televisão a exibição de apenas um elemento de imagem por vez. Quando a varredura é feita suficientemente rápida, os elementos de imagem aparecem aos olhos como uma imagem completa.

A persistência visual, citada anteriormente, opera na apresentação sequencial rápida de várias imagens isoladas, gerando a sensação de movimento.

Cada imagem é chamada de frame ou quadro. Para produzir a sensação de movimento numa forma natural, no mínimo deve-se gerar 16 quadros por segundo [Chaves1991]. Para

gerar-se *full motion video* gera-se no padrão americano e brasileiro, 30 quadros por segundo e nos padrões europeus, 25 quadros por segundo.

Existem dois modos de varredura de vídeo : (1) entrelaçada, onde transmite-se 60 campos (ou meios-quadros) do *frame* e a varredura progressiva que transmite 30 *frames* completos por segundo para evitar o efeito *flicker*, ou cintilação.

No padrão NTSC, apesar de o sistema possuir 525 linhas horizontais, na prática, o número de linhas aproveitadas é de 330. As linhas restantes são utilizadas para a volta do sinal até o começo da linha seguinte, para apagamento, etc. A resolução horizontal é de 440 pontos.

A tela de monitor de computador no padrão VGA apresenta uma resolução bem maior (640 por 480 pontos), mas exibe apenas 16 cores simultaneamente. Já em um aparelho de televisão reproduz-se 16 milhões de cores. Um monitor compatível com o Super-VGA, a resolução é de 1024 por 768 pontos, mas exibe, no máximo, 256 cores.

Assim, conclui-se que o sistema com melhor resolução para receber uma imagem NTSC é o VGA de 640 por 480 pontos. Mesmo assim, quando é feita a digitalização da imagem, é necessário que se realize um processo de interpolação para ajustar os pontos que irão sobrar, 330 na direção vertical e cerca de 200 na horizontal [McQuillin1991].

Neste projeto os vídeos utilizados foram reproduzidos no padrão Super-VGA, já que estes tinham sido capturados neste padrão. Uma observação a ser feita é que a capturação dos vídeos não foi realizada neste projeto, utilizaram-se vídeos no padrão AVI (Audio Video Interlaced) capturados por empresas como Microsoft, Creative Labs e outras.

A resolução dos vídeos nesse projeto é de 160 por 120 pontos (horizontal x vertical).

### **5.3 - PADRÃO DE COMPRESSÃO MPEG I**

O MPEG (Motion Picture Experts Group) é um grupo de pessoas subordinadas à ISO com a função de gerar padrões de compressão de vídeo e áudio. O nome oficial desse padrão é o ISO/IEC (International for Standardization Organization/International Electrotechnical Commission) JTC1 (Joint Technical Committee 1) SC29 (Sub-Committee 29) WG11 (Working Group 11). Em particular, ele define a sequência de bits que irá gerar a compressão e conseqüentemente a descompressão [Popp1994].

O padrão MPEG é o padrão utilizado neste projeto para a compressão dos vídeos originalmente capturados no padrão AVI (Audio Video Interlaced).

O padrão MPEG I, padrão ISO 11172, possui uma taxa de compressão de 1.5 Mbits/s. Essa taxa é muito especial, já que é a taxa de transferência dos CD's audio e das fitas DAT's (Digital Audio Tape) [Popp1994].

### 5.3.1 - COMPRESSÃO DE VÍDEOS

O padrão MPEG I comprime vídeos com resolução de 352 por 240 *pixels*, com 30 *frames/segundo*. Deve-se notar que o padrão europeu é 352 por 288 *pixels* com 25 *frames/segundo*. Esse padrão é derivado do padrão para televisão digital CCIR-601, utilizado em equipamentos de vídeo digital. Como 288 multiplicado por 50 é exatamente igual a 240 multiplicado por 60, os dois formatos têm a mesma taxa de geração de *frames*.

utilização dos métodos de transformação e quantificação do padrão MPEG.

Alguns tipos de processamentos podem ser realizados antes de haver a compressão do vídeo :

- Entrelaçamento adaptativo : Esse método mapeia um vídeo com varredura entrelaçada com resolução de 720 x 480 *pixels* para um vídeo com varredura progressiva com resolução de 352 x 240 *pixels*. Esse método baseia-se basicamente na variação entre dois campos. Caso a variação seja pequena, utiliza-se a média entre os campos para formar um macrobloco, caso contrário, a área de um dos campos é selecionada.
- Pré-anti-aliasing e redução prévia dos blocos : Se uma prévia análise de um *frame* indica que algum sério problema no vídeo irá aumentar se o vídeo for codificado nas condições correntes, um filtro de pré-anti-aliasing, é aplicado ao vídeo, evitando-se que sejam gerados sinais com frequência que excedam o limite determinado pelo teorema de Nyquist . Esse filtro pode ser global, isto é, aplicado por toda a imagem ou local.

Outras técnicas de pré-filtragem são : análise da textura, mascaramento, análise dos limites e análise dos movimentos.

As imagens manipuladas são coloridas mas são convertidas para o espaço YUV (direção de luminância, direção de crominância horizontal e direção de crominância vertical) e depois os

dois canais de crominância (U e V) são reduzidos para 176 por 120 *pixels* . Apesar da perda da resolução, o olho humano não capta essa diferença.

A compressão primeiramente é realizada com a predição do próximo movimento dentro do espaço temporal e depois é utilizado um algoritmo DCT (Discrete Cosine Transform) para organizar a redundância. O algoritmo DCT trabalha com blocos 8 por 8, e a predição do próximo movimento é realizado no canal de luminância (Y) com blocos 16 por 16. Os coeficientes DCT são quantificados, isto significa que esses coeficientes são todos divididos por um mesmo valor. Essa quantificação modifica os três parâmetros de direção (16 por 16 da direção Y e 8 por 8 da direção U e V). No anexo V encontra-se a explicação do algoritmo DCT.

Existem três tipos de codificação de *frames*. Há os tipos "I" ou *intra frames*. Esses são *frames* codificados como imagens congeladas, não interessa os *frames* antigos. Depois temos os "P" ou *frames* preditos utilizando os frames I ou P mais recentes. Por último, há os "B" *frames* ou *frames* bidirecionais. Eles são gerados a partir dos "I" ou "P" *frames*, um no passado e outro no futuro. Primeiramente realiza-se uma média entre o bloco passado e o bloco futuro e subtrai-se essa média do bloco que está sendo codificado. Caso o *frame* gerado não seja satisfatório, gera-se um *frame* "I".

Como os "B" *frames* são gerados através de uma média entre dois blocos intermediários, o problema de ruído é minimizado quando manipula-se vídeos do padrão MPEG I (1.15 Mbits/s), geralmente tem-se um ganho de 2 dB na relação sinal/ruído (s/n). Entretanto, a altas taxas de transmissão os "B" *frames* não contribuem tanto para o refinamento progressivo dos vídeos.

Há muitas controvérsias sobre a eficiência na utilização dos "B" *frames* já que existem problemas como : (1) complexidade computacional, pois tem que ser realizada uma média entre dois macroblocos (16 x 16 blocos); (2) necessidade de uma largura de banda de 16 Mbytes/s para essa predição extra; (3) um *buffer* maior deve existir para o armazenamento dos *frames* utilizados como referência para as predições e (4) um atraso é gerado pois o *frame* que foi utilizado para a predição deve ser transmitido antes dos frames intermediários serem decodificados e mostrados.

A sequência gerada usualmente é representada assim : IBBPBBPBBPBBIBBPBBPB... O funcionamento é o seguinte : decodifica-se o *frame* "I", depois decodifica-se o *frame* "P", mantém-se os dois em memória e depois decodifica-se os dois *frames* "B". Provavelmente enquanto está se decodificando o *frame* "P", mostra-se o *frame* "I", mostra-se os "B's" enquanto

os decodifica, e depois mostra-se o "P" enquanto decodifica-se o próximo "P", e assim continua a sequência.

### 5.3.2 - COMPRESSÃO DE ÁUDIO

O som é o resultado de diferentes pressões no ar. Quando ele é captado por um microfone e alimentado para um amplificador, ele se transforma em níveis de voltagem. A voltagem é gerada pelo computador um certo número de vezes por segundo. Para o áudio com qualidade CD é necessário gerar-se amostras de áudio de 44 kHz, sendo que cada amostra tem uma resolução de 16 bits. No modo stereo isso nos dá uma taxa de 1.4 Mbits/s. Assim pode-se notar que deve existir uma boa compressão de dados.

A compressão do áudio é muito diferente da compressão de vídeo, já que o ouvido capta muito mais detalhes que os olhos, mas o ouvido é mais vagaroso que o olho.

O padrão IS 11172-3 descreve o padrão de compressão do áudio utilizando três esquemas de codificação : nível 1, 2 e 3, crescendo em complexidade e performance na mesma proporção. Esses três níveis são compatíveis na escala hierárquica, isto é, o nível 3 decodifica o nível 2 e o nível 1, e assim se segue. Para cada nível, o padrão define o formato da sequência de bits e o decodificador [Popp1994].

Para a compressão do áudio, utiliza-se modelos baseados em psicoacústica derivados de experiências com especialistas em ouvir sons que normalmente outro ser humano não ouviria. Existe o efeito chamado de "mascaramento", onde um componente de uma única frequência não permite que se ouça frequências que estejam próximas desta, fazendo com que a relação (energia x frequência) seja descrita por uma curva empírica.

O decodificador é muito menos complexo; sua única tarefa é decompor o sinal de áudio fora dos componentes codificados.

Todos os três níveis utilizam o mesmo filtro e utilizam a mesma informação no *header* para suportar a estrutura hierárquica do padrão. Esses níveis possuem uma estrutura que contém partes que são mais sensíveis a erros (*header*, alocação dos bits e fator de escala) e partes que são menos sensíveis (dados dos componentes do espectro).

Abaixo pode-se ver a qualidade de som de cada nível :

- Nível 1 : Sua taxa de compressão é de 192 kbits/s por cada canal de áudio. Essa é uma versão simplificada do nível 2;
- Nível 2 : Sua taxa de compressão é de 128 kbits/s por cada canal de áudio. Será utilizado nas futuras redes *broadcasting* digitais de áudio;
- Nível 3 : Sua taxa de compressão é de 64 kbits/s por cada canal de áudio. Esse nível já é utilizado em várias redes de telecomunicações (ISDN, transmissões via satélite e outras) e sistemas de áudio.

Para baixas taxas de bits (60 ou 64 Kbits/s por canal), o nível 2 gera sinais de áudio irreconhecíveis, já o nível 3 gera sinais mais perceptíveis. Quando se fala em taxas próximas de 120 kbits/s ou mais por canal, o nível 2 e nível 3 comportam-se de maneira bem semelhante, gerando os sinais reconstruídos não muito diferentes do original [Popp1994].

Nos sinais de áudio pode-se gerar amostras com baixa resolução (menos que 16 bits por amostra) ou pode-se gerar sinais com alta qualidade (mais que 16 bits por amostra).

Nyquist [Tanenbaum1988] provou que se um sinal arbitrário passa por um canal com largura de banda  $H$ , o sinal pode ser completamente reconstruído simplesmente gerando-se  $2H$  amostras de sinais por segundo. Esse valor  $2H$  deve ser exato já que o filtro pode não deixar passar amostras com frequências maiores que  $2H$ . Na fórmula 5.1 pode-se ver o teorema :

$$\text{Máxima taxa de transmissão} = 2 H \log_2 V \text{ bits/segundo;}$$

$$\text{onde } V = \text{número de níveis do sinal.} \quad (5.1)$$

Como pode-se ver no teorema acima, deve-se gerar sinais com frequência de pelo menos duas vezes a maior frequência que deseja-se reproduzir, no caso em estudo temos uma representação binária do sinal e conseqüentemente o número de níveis do sinal é igual a dois. Os seres humanos podem ouvir sons com frequências entre 20 Hz até 20 kHz. Assim, aplicando-se o teorema de Nyquist a frequência a ser gerada seria de 44 kHz. Então a solução encontrada pelo grupo MPEG foi a redução da resolução do sinal, isto é, o número de bits que representam uma amostra do sinal.

A real razão em se utilizar 16 bits para representar cada amostra do sinal de áudio é melhorar a relação sinal/ruído ( $s/n$ ). O ruído que se fala aqui é a quantificação de ruído gerado pelo processo de digitalização. Para cada bit que é adicionado, melhora-se 6 dB na relação sinal/ruído ( $s/n$ ) [Popp1994]. Para o ouvido, 6 dB corresponde na duplicação do nível do sinal. Um arquivo de áudio com qualidade de CD possui em média 90 dB da relação sinal/ruído.

Quando falamos em resolução de 8 bits, os ruídos são percebidos em momentos de silêncio ou entre palavras e sentenças, mas quando temos sons muito altos, esses ruídos não são percebidos. É justamente este o efeito de mascaramento e é a chave para codificação de áudio no padrão MPEG.

Se tem-se um tom muito forte com frequência aproximada de 1000 Hz e tem-se outro tom com frequência de 1100 Hz, esse segundo tom possui a relação sinal/ruído 18 dB mais baixa. Não será possível ouvir esse segundo tom pois ele será completamente mascarado pelo primeiro tom. Assim, como foi visto, qualquer som fraco que esteja perto de um som mais forte será mascarado.

O efeito de mascaramento significa que pode-se aumentar o ruído em torno de um sinal forte porque o ruído será mascarado, desta forma pode-se utilizar menos bits para representar o sinal, significando que houve compressão.

O padrão MPEG áudio nível 2 divide o espectro de frequência (20 Hz a 20 kHz) em 32 sub-bandas. Cada sub-banda conterá uma parte do espectro do som.

O decodificador também tem que considerar a sensibilidade do ouvido para várias frequências. O ouvido humano é muito menos sensível nas baixas e altas frequências. As maiores sensibilidades estão entre 2 kHz a 4 kHz, a mesma região que a voz humana ocupa.

Cada sub-banda deve possuir frequências que tenham as mesmas propriedades psicoacústicas. No padrão MPEG nível 2, cada sub-banda possui o tamanho de 750 Hz. O ideal seria se as sub-bandas fossem mais estreitas nas baixas frequências e mais largas nas frequências mais altas. O nível 3 possui uma resolução maior, aproximadamente 18 vezes mais e essa é a razão do nível 3 possuir maior performance do que o nível 2 a baixas taxas.

Outra técnica de redução de bits considera os componentes de tons e não tons do som. Em um sinal stereo pode-se ter muitas redundâncias entre os canais. Todos níveis do padrão MPEG exploram esses efeitos stereos utilizando o modo *joint-stereo*, isto é, os canais direito e esquerdo são codificados de forma matricial. Somente o nível 3 reduz a redundância aplicando a codificação Huffman.

Para o nível 2 e nível 3, o codificador trabalha com duração de 24 ms de som de cada vez. O mascaramento é calculado para os sons mais fortes e os sons mais fracos continuam com os ruídos, que é chamado de eco de ruído. O nível 3 trata esse problema utilizando uma janela para análise do som um pouco menor, por volta de 4 ms.

O padrão MPEG I permite a utilização de dois canais de áudio. Esses canais podem ser simples (mono), duplo (dois canais mono), *stereo* (nível 2 e nível 3) e *joint-stereo* (nível 3). O padrão MPEG II utiliza vários canais com as mesmas características acima citadas.

Os tempos de *delay* dependem do tipo da implementação que se está trabalhando. Abaixo mostra-se alguns tempos de *delay* aproximados para cada nível do padrão :

- Nível 1 - 19 ms ;
- Nível 2 - 35 ms ;
- Nível 3 - 59 ms.

O tempo de *delay* pode ser crítico para muitas aplicações, por exemplo em um sistema de teleconferência. Uma conversa em tempo real e de maneira bem natural deve possuir um tempo de *delay* em torno de 10 ms.

## 5.4 - PADRÃO DE COMPRESSÃO MPEG II

### 5.4.1 - COMPRESSÃO DE VÍDEO

O comportamento do padrão MPEG II, padrão ISO 13818, varia de acordo com o tipo de aplicação do usuário [Popp1994] :

- Para a resolução de 352 x 240 *pixels* com 30 *frames/s*, a taxa de compressão é em torno de 4 Mbits/s, sendo muito utilizada para aplicações de software;
- Para a resolução de 720 x 480 *pixels* com 30 *frames/s*, a taxa de compressão é de 15 Mbits/s, sendo muito utilizada em chips decodificadores, CATV, satélites. 95 % das aplicações encontram-se nessa categoria;
- Para a resolução de 1440 x 1152 *pixels* com 30 *frames/s*, a taxa de compressão é de 60 Mbits/s, as aplicações HDTV (High Definition Television ) utilizam essa categoria do padrão MPEG II;
- Para a resolução de 1920 x 1080 *pixels* com 30 *frames/s*, a taxa de compressão é de 80 Mbits/s.

A diferenciação entre o padrão MPEG I e o padrão MPEG II se dá através do *header* . O padrão MPEG II possui algumas extensões do *header* chamadas de *sequence\_scalable\_extension*.

### 5.4.2 - COMPRESSÃO DE ÁUDIO

O padrão de compressão de áudio do MPEG II é compatível com os dois canais do padrão MPEG I (direito e esquerdo).

### 5.5 - PADRÃO DE COMPRESSÃO MPEG III

O padrão MPEG III foi concebido para ser utilizado em aplicações HDTV com resolução de 1920 x 1080 *pixels* com 30 *frames/s* e com taxa de compressão entre 20 a 40 Mbits/s. Depois descobriu-se que o padrão High-1440 Level do MPEG II apenas com alguns poucos aperfeiçoamentos poderia atender aos requisitos das aplicações HDTV.

### 5.6 - PADRÃO DE COMPRESSÃO MPEG IV

O padrão MPEG IV trabalha com aplicações a baixas velocidades com resolução de 176 x 144 *pixels* com 10 *frames/s* e com taxa de compressão de 4800 a 64000 bits/s. Essa padrão é muito utilizado na aplicação de videofone em linhas telefônicas analógicas. Esse padrão ainda encontra-se em fase de identificação da aplicação.

### 5.7 - O ESTADO DA ARTE DO PADRÃO MPEG

A compressão de vídeo multiplexado com áudio no padrão MPEG encontra-se em fase de pesquisa e desenvolvimento. Para que a compressão de vídeo com áudio seja realizada de maneira satisfatória e viável, necessita-se, no mínimo, um processador 486 DX/66 MHz, decodificando quadros MPEG de tamanho (horizontal x vertical) de 160x120 com 10 quadros por segundo.

Nesse projeto, a compressão do vídeo foi realizada separadamente do áudio porque ainda não se encontra nenhum software para o sistema operacional Windows NT, que realize a compressão e multiplexação dessas duas mídias. Informações a respeito do desenvolvimento de um aplicativo que multiplexe vídeo e áudio compactado pode ser encontrado no [16]. Todas as

referências que possuem a letra 'I' acompanhada de um número inteiro são referências de endereços Internet que estão descritas no anexo III.

Para que haja uma decodificação de vídeo multiplexado com áudio em um vídeo full-screen, baseado em software, é necessário que se utilize máquinas com processadores bastante velozes, como as máquinas RISC. Um microcomputador com processador Intel DX-2 66 MHz, com vídeos multiplexados com áudio de tamanho horizontal x vertical (160 x 120) somente possui um desempenho satisfatório para vídeos com 10 quadros por segundo, o que está bem abaixo do padrão de televisão, que é 25 ou 30 quadros por segundo.

Na realidade, a decodificação baseada em hardware oferece o melhor desempenho. Especialistas acreditam que esse hardware venha a ficar economicamente mais viável em pouco tempo.

O software utilizado para a compressão dos vídeos é compatível com o padrão MPEG I e MPEG II. Nesse projeto, os vídeos por serem de baixa qualidade, foram codificados segundo o padrão MPEG I.

Alguns dos testes realizados com o padrão MPEG I, onde utilizou-se o software descrito na próxima seção, podem ser analisados nas tabelas 5.1, 5.2, 5.3 e 5.4.

Nome arquivo :	wind.avi		
Tamanho arquivo .AVI :	1.894 Kbytes		
Número de quadros :	160 quadros		
Número de quadros por segundo :	30 quadros/segundo		
Tempo de duração do vídeo :	05:33 segundos		
Tamanho da janela hxv :	160 x 120		
Áudio :	<input type="checkbox"/> Mono	<input checked="" type="checkbox"/> Stereo	<input type="checkbox"/> Joint Stereo
Áudio :	<input type="checkbox"/> 8 bits	<input checked="" type="checkbox"/> 16 bits	
Áudio :	<input type="checkbox"/> 11 KHz	<input type="checkbox"/> 22.1 KHz	<input checked="" type="checkbox"/> 44.1 KHz
Tamanho arquivo .MPG :	232 Kbytes		

Tabela 5.1 - Arquivo WIND.AVI compactado pelo padrão MPEG

Nome arquivo :	heya.avi		
Tamanho arquivo .AVI :	576.320 Kbytes		
Número de quadros :	110 quadros		
Número de quadros por segundo :	30 quadros/segundo		
Tempo de duração do vídeo :	03:67 segundos		
Tamanho da janela hxv :	160 x 120		
Áudio :	<input type="checkbox"/> 8 bits	<input checked="" type="checkbox"/> 16 bits	
Áudio :	<input type="checkbox"/> Mono	<input checked="" type="checkbox"/> Stereo	<input type="checkbox"/> Joint Stereo
Áudio :	<input type="checkbox"/> 11 KHz	<input type="checkbox"/> 22.1KHz	<input checked="" type="checkbox"/> 44.1 KHz
Tamanho arquivo .MPG	159.174 Kbytes		

Tabela 5.2 - Arquivo HEYA.AVI compactado pelo padrão MPEG

Nome do arquivo .WAV	Tada.WAV
Tamanho do arquivo	27.804 Kbytes
Tempo de duração do áudio	01:25 segundos

Tabela 5.3 - Arquivo não compactado TADA.WAV

Nome do arquivo .MP2	Greeting.MP2
Tamanho do arquivo	54.334 Kbytes
Tempo de duração do áudio	06:79 segundos

Tabela 5.4 - Arquivo Greeting.MP2 compactado pelo padrão MPEG

Como podemos verificar na tabela 5.1, o arquivo compactado ficou 8,16 vezes menor que o original. Já o arquivo da tabela 5.2, ficou 3,62 vezes menor que o original. Outros testes foram realizados e notou-se que quanto maior for o arquivo de vídeo padrão AVI, maior é a razão de compressão. Na prática, os arquivos de vídeo são geralmente maiores que 1 Mbyte, assim a razão de compressão será muito boa.

Quanto ao áudio, na tabela 5.3 mostrou-se um arquivo não compactado, com duração de 01:25 segundos, ocupando 27.804 Kbytes. Fazendo uma comparação com a tabela 5.4, observa-se que o arquivo de áudio compactado, com duração de 6:79 segundos, ocupou 54.334 Kbytes. Esse mesmo arquivo, não compactado, ocuparia 151.031 Kbytes, ou seja, 2,77 vezes maior:

## Capítulo **6**

### **DESCRIÇÃO DO AMBIENTE DE REDES UTILIZADO NO DESENVOLVIMENTO DO PROJETO**

#### **6.1 - INTRODUÇÃO**

A seguinte configuração foi instalada para realização dos testes do projeto :

- Microcomputador Intel 486 DX-2 66 Mhz, estação multimídia, com 20 Mbytes de memória RAM, 500 Mbytes de disco rígido com controladora IDE, placa Super VGA Trident de 1 Mbyte de memória, placa de som Sound Blaster Pro da Creative Labs e sistema operacional Windows NT Server 3.5.

Foi designado o codinome 'Picasso' para esta estação. Esta estação é a servidora FTP (File Transfer Protocol) de vídeo e áudio. É uma estação do tipo 'B' segundo a norma ANSI X3T9.5, já que está conectada ao anel óptico através de um concentrador (*Hub*).

Possui adaptador FDDI Syskonnnect SK5341, MAC (Medium Access Control) simples, conexão simples e barramento 32-bit EISA (Extended Industry Standard Architecture) [Syskonnnect1993].

O protocolo de transporte e de rede é o TCP/IP. No nível físico e de enlace encontramos os subníveis : PMD (Physical Medium Dependent), LLC (Logical Link Control), MAC (Medium Access Control), PHY (Physical Layer Protocol) e o protocolo de gerenciamento SMT 7.3 (Station Management). Esse subníveis estão definidos segundo a norma ANSI X3T9.5, que padroniza a arquitetura FDDI. Essa arquitetura pode ser visualizada na figura 3.1.

- Estação Sun SparcServer 10, sistema operacional Sun OS 4.1.

Foi designado o codinome 'Bach' para esta estação. É uma estação do tipo 'B' segundo a norma ANSI X3T9.5, já que está conectada através de um concentrador (*Hub*).

No desenvolvimento do projeto desejava-se transmitir os vídeos e áudios de um microcomputador com o sistema operacional Windows NT 3.5 para outro microcomputador com as mesmas configurações, mas não foi possível termos disponível para o desenvolvimento do

projeto dois microcomputadores com as características da estação 'Picasso', conectados ao anel óptico. Por esse motivo utilizamos uma estação Sun SparcServer 10 (estação 'Bach') transmitindo para um microcomputador (estação 'Picasso'). Como utilizamos a estação Sun somente para transmissão dos vídeos e áudios, isto é, não coletamos nenhum dado estatístico dessa estação, os resultados não foram influenciados por essa estação.

- *Hub Linkbuilder FDDI Workgroup* da 3Com, com módulo de gerenciamento para o protocolo SNMP. É um Hub com conexão DAC (Dual Attachment Concentrator), pois está conectado ao anel primário e secundário do anel FDDI [3ComA1993];
- Anel primário e secundário de fibra ótica multimodo 62.5/125  $\mu\text{m}$  alcançando aproximadamente 5 Km de distância cada anel;

A conexão da estação 'Picasso' e 'Bach' aos seus respectivos *Hubs* foi realizada através do cabeamento de fibra ótica multimodo 62.5/125  $\mu\text{m}$ .

Na figura 6.1 pode-se visualizar a interconexão dos microcomputadores 'Picasso' e 'Bach', através dos seus respectivos *Hubs*, ao *backbone* FDDI da universidade :

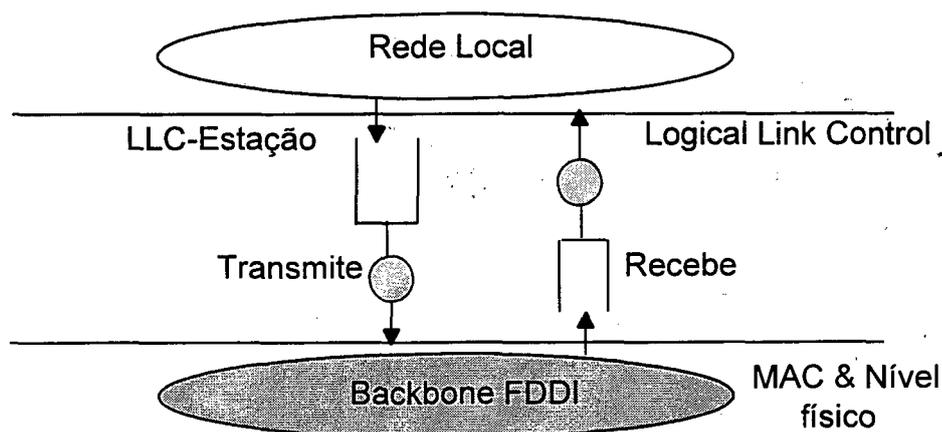


Figura 6.1 - Interconexão da Rede Local com o Backbone FDDI

Na figura 6.2 podemos visualizar a interconexão dos Hubs e das estações aos anéis ópticos primário e secundário.

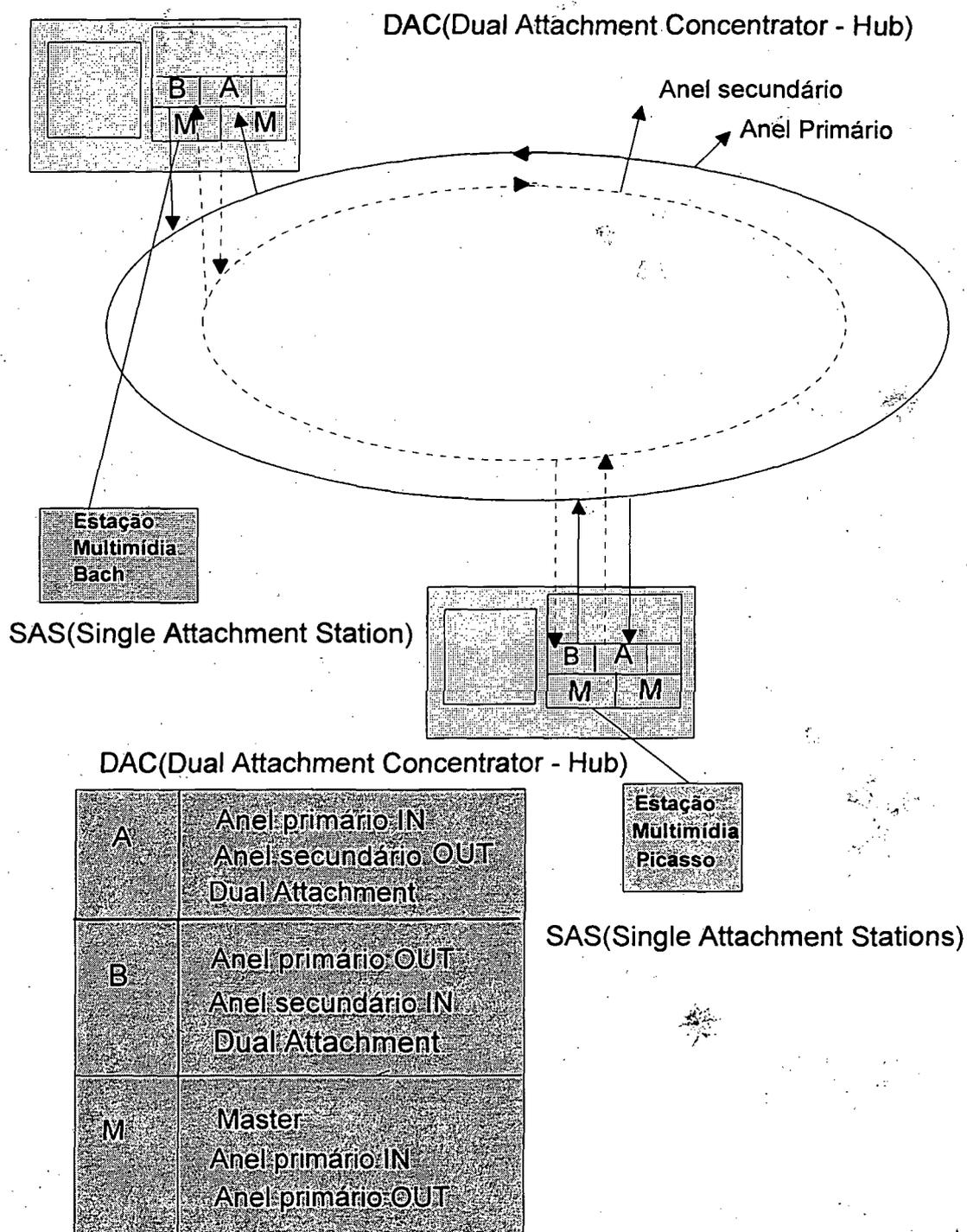


Figura 6.2 - Configuração da conexão do *Hub* e das estações ao anel óptico

## 6.2 - DESCRIÇÃO DO AMBIENTE DE GERENCIAMENTO DO PROJETO

No modelo de implementação da Internet, a obtenção das informações de gerenciamento e o seu posterior armazenamento na MIB (Management Information Base) fica a cargo de um elemento de software qualquer, não padronizado, e especificado a nível de implementação. Nesse projeto esse software é o SunNet Manager 2.2 da Sun Microsystems, Inc. Business. As informações contidas na MIB são lidas, alteradas e transferidas a partir de operações de gerenciamento especificadas em mensagens do protocolo SNMP transmitidas pelo gerente. Tal protocolo é usado para transportar, também, o resultado destas operações dos agentes ao gerente [Makron1994].

Os componentes do modelo Internet são :

- Um conjunto de objetos gerenciados, cada um contendo um agente e a MIB associada. Esses objetos serão descritos posteriormente dentro dessa mesma sessão;
- Uma estação de gerenciamento de rede, no caso desse projeto, representa uma estação de trabalho Sun Sparc Server10 ligada ao anel óptico;
- Um protocolo de gerenciamento de rede que é utilizado pela estação e pelos agentes, para a troca de informações de gerenciamento, nesse caso o SNMP e o ANSI FDDI SMT 7.3.

Nesse modelo de arquitetura são definidos tipos de objetos que contém cinco campos : nome textual identificando o objeto, uma sintaxe ASN.1 (Abstract Syntax Notation.One), a definição semântica associada ao tipo de objeto, o tipo de acesso (*read-only*, *read-write*, *write-only* e não acessível) e o *status* (obrigatório, opcional ou obsoleto).

Os tipos utilizados são [Makron1994]:

- Tipo simples, utilizando quatro tipos primitivos da ASN.1 : INTEGER, OCTET, STRING NULL e OBJECT IDENTIFIER;
- Tipos contexto-de-aplicação, utilizando tipos especiais definidos pela SMI (Structure of Management Information) : *IpAddress*, *NetworkAddress*, *Counter*, *Gauge*, *Timeticks* e *Opaque*.
- Tipos construídos, utilizando, para sua definição, tipos construtores da ASN.1 : *<list>* e *<table>*.

O ambiente do sistema de gerenciamento do projeto pode ser visualizado na figura 6.3 :

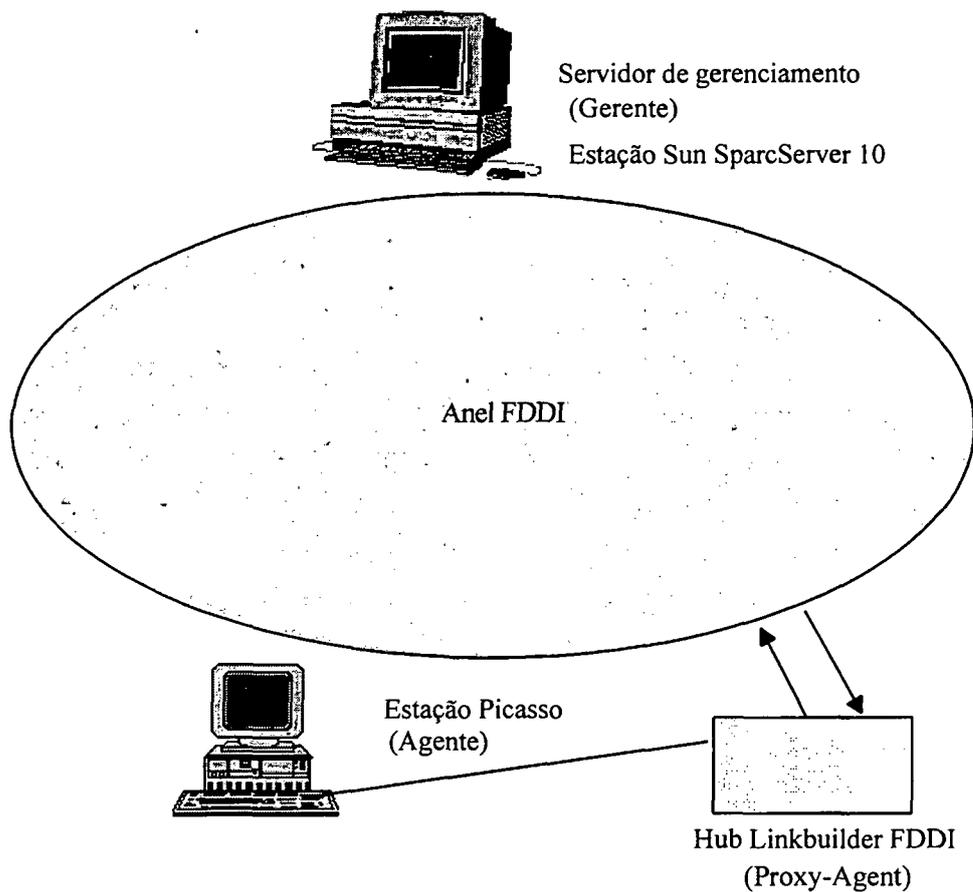


Figura 6.3 - Ambiente de gerenciamento do projeto

Com a utilização dos *proxy agents*, agentes procuradores, o protocolo SMT 7.3 dos agentes se comunica com o protocolo SNMP no gerente.

A descrição dos objetos para gerenciamento dos dispositivos que implementam a arquitetura FDDI baseada no padrão ANSI FDDI SMT 7.3, que no projeto representam a estação multimídia 'Picasso', a estação 'Bach' e o Hub Linkbuilder FDDI encontra-se no anexo VI. A MIB que está se utilizando é a MIB FDDI cuja definição pode ser encontrada em [Case1993].

### 6.3 - DESCRIÇÃO DO MODELO DE GERENCIAMENTO CRIADO PARA GERENCIAMENTO DO *BACKBONE* FDDI E DAS ESTAÇÕES MULTIMÍDIA

O software utilizado para o gerenciamento do Backbone FDDI da UFSC, das estações multimídia 'Picasso' e 'Bach' e do *Hub* Linkbuilder FDDI é o Sunnet Manager 2.2 da Sun Microsystems, Inc. Business. Esse software é compatível com o padrão SNMP de gerenciamento de redes e foi executado em uma estação Sun SparcServer 10, como pode-se verificar na figura 6.3.

No Hub Linkbuilder FDDI foi instalado o software Smart Agent, versão 1.1a do protocolo SNMP [3ComB1993], que suporta a MIB FDDI [Case1993] e a MIB II [McQuillin1991] do SNMP.

O SMT é o protocolo de gerenciamento dos níveis físico e de enlace de dados da arquitetura FDDI, que está sendo executado na estação 'Picasso'. Esse protocolo pode ser visualizado na figura 3.1.

Para estabelecer o relacionamento do SNMP com o SMT 7.3 utiliza-se *proxy agents* instalados no Hub Linkbuilder FDDI.

O Hub Linkbuilder FDDI se conecta ao anel óptico da UFSC, onde está conectada também a estação Sun SparcServer 10, que é a estação gerenciadora.

O direcionamento a um agente é feito através de requisições onde se especifica os parâmetros de uma determinada tarefa de gerenciamento. A requisição contém informações do objeto a ser gerenciado e a frequência com que o agente deve reportá-la.

As requisições de eventos provêm o envio dessas requisições periodicamente e as requisições de dados somente permitem o direcionamento ao agente quando determinada condição é alcançada.

Nas tabelas 6.1 e 6.2 estão mostradas as requisições de eventos e de dados utilizadas no projeto.

Nome requisição de evento	Proxy Agent	Intervalo de Polling	Atributos	Relação 1	Prioridade
EventoMAC	Hub Linkbuilder FDDI	900 seg.	fddimibMACTokenCts	> 10	Alta
			fddimibMACExpiredCts	> 10	Média
			fddimibMACNotCopiedCts	> 10	Média
			fddimibMACLateCts	> 10	Média
			fddimibMACRingOpCts	> 10	Média
ifSystemReboot	Host local	900 seg.	boottime	aumentou de 86400 seg. desde o último valor coletado	Alta
			uptime	aumentou de 86400 seg. desde o último valor coletado	Alta
When Disk is Full	Host local	900 seg.	used	> 300	Alta
When System is not Reachable	Host local	600 seg.	reachable	.not. true	Alta

Tabela 6.1 - Descrição das requisições de eventos

Nome da requisição de dados	Proxy System	Intervalo	Arquivo Log	Atributos
ConfigCaminho	Hub Linkbuilder FDDI	43200 seg.	ConfigCaminho	fddimibPATHConfigCurrent fddimibPATHConfigResource fddimibPATHConfigTokenOrder
Graph Host Performance	Host local	30 seg.	Performance	ierrs oerrs ipkts cpubusy opkts

Identificação MAC	Hub Linkbuilder FDDI	30 seg.	NumeroMAC	fddimibMACNumber
NumeroPorta	Hub Linkbuilder FDDI	30 seg.	NumeroPorta	fddimibPORTNumber
InputInterface	Hub Linkbuilder FDDI	30 seg.	InputInterface	ifNErrors ifNDiscards
TabelasMAC	Hub Linkbuilder FDDI	30 seg.	TabelasMAC	fddimibMACTMaxCapability fddimibMACCurrentPath fddimibMACSMTAddress fddimibMACTReq fddimibMACTNeg fddimibMACTMax fddimibMACTvxValue fddimibMACFrameCts fddimibMACCopiedCts fddimibMACTransmitCts fddimibMACLostCts fddimibMACErrorCts fddimibMACFrameErrorRatio fddimibMACFrameErrorFlag
TabelasPorta	Hub Linkbuilder FDDI	7200 seg.	TabelasPorta	fddimibPORTSMTIndex fddimibPORTMyType fddimibPORTCurrentPath fddimibPORTAvailablePaths fddimibPORTPMDClass fddimibLCTFailCts fddimibLemRejectCts fddimibLerCutoff fddimibPORTLerAlarm fddimibPORTConnectState fddimibPORTLerFlag fddimibPORTHardwarePresent fddimibPORTAction

TabelasSMT	Hub Linkbuil der FDDI	30 seg.	TabelasSMT	fddimibSMTIndex fddimibSMTStationID fddimibSMTUserData fddimibSMTMACCs fddimibSMTAvailablePaths fddimibSMTTNotify fddimibSMTStatRptPolicy fddimibSMTByPassPresent fddimibSMTCFState fddimibSMTRemoteDisconnect fddimibSMTStationStatus fddimibSMTTTimeStamp fddimibSMTTTransitionTime
RecordDiskSpace	Host local	108000 seg.		used capacity
NumeroSMT	Hub Linkbuil der FDDI	30 seg.		fddimibSMTNumber
OutputInterface	Hub Linkbuil der FDDI	30 seg.		ifOutQLen ifOutErrors
MediaErroConexao	Hub Linkbuil der FDDI	30 seg.	MediaerroConex ao	fddimibPORTLERestimate
NumeroCaminho	Hub Linkbuil der FDDI	30 seg.	NumeroCaminho	fddimibPATHNumber
NumeroMAC	Hub Linkbuil der FDDI	30 seg.	NumeroMAC	fddimibMACNumber
IndiceMACSMT	Hub Linkbuil der FDDI	30 seg.	IndiceMACSMT	fddimibMACSMTIndex
TabelasCaminho	Hub Linkbuil der FDDI	300 seg.	TabelasCaminho	fddimibPATHSMTIndex fddimibPATHIndex fddimibPATHvxLowerBound fddimibPATHMaxLowerBound fddimibPATHMaxTReq

StatusInterface	Hub	7200 seg.	StatusInterface	ifType
	Linkbuil			ifSpeed
	der			ifOperStatus
	FDDI			

Tabela 6.2 - Descrição das requisições de dados

Um *trap* é um relato não solicitado, enviado por um agente quando alguma condição de erro inesperada ocorre.

Na tabela 8.3 estão relacionados os *traps* considerados no projeto.

Nome do <i>trap</i>	Causa
ColdStart	Inicialização através do hardware
Authentication Failure	Violação de autenticação
a3comLowBattery	Bateria fraca
a3comHighTemp	Temperatura alta
a3comFanFailed	Falha no equipamento
a3comBadTelnetPasswd	Password do acesso via telnet incorreta
a3comBadConsolePasswd	Password de acesso via console incorreta

Tabela 6.3 - Descrição dos *traps*

A seguir será mostrada a representação gráfica do ambiente de gerenciamento criado no software Sunnet Manager 2.2.

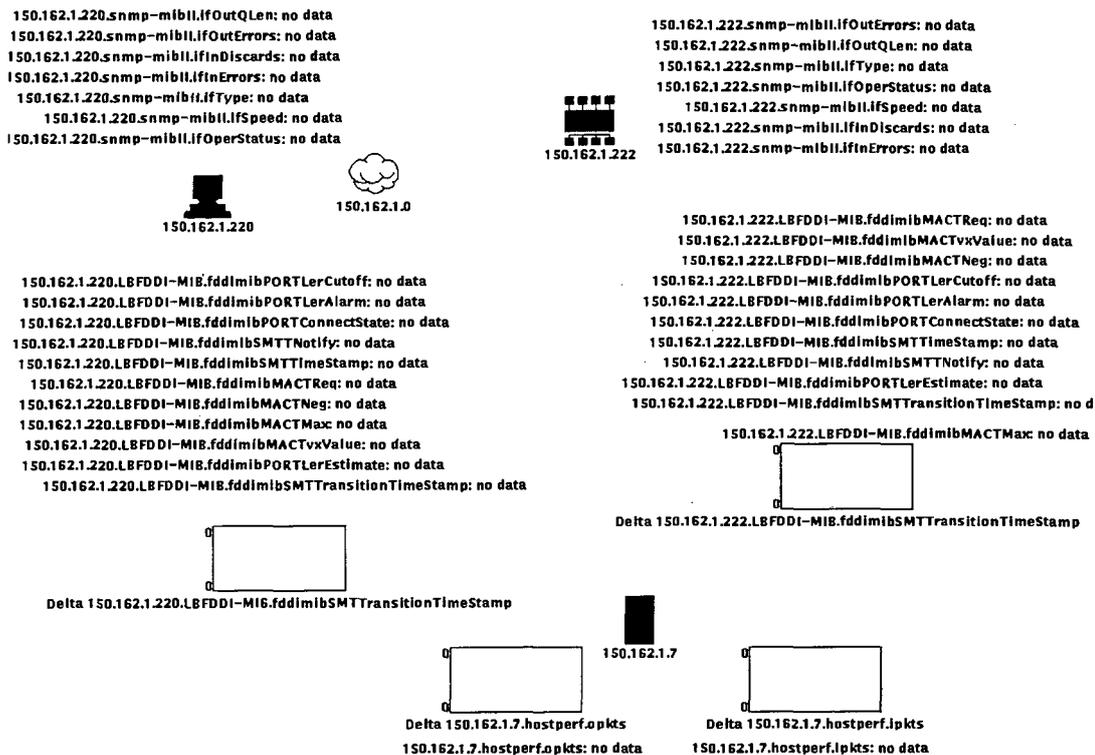


Figura 8.4 - Representação gráfica do ambiente de gerenciamento criado

## 6.4 - PARÂMETROS A SEREM MEDIDOS PARA ANÁLISE DO DESEMPENHO DA REDE FDDI

Nas situações de coleta de dados não foram consideradas situações de erros como : desconexão da rede, queda de luz e problemas de cabeamento, nem procedimentos de recuperação, por serem situações onde não se tinha interesse de estudo.

As duas estações multimídia operam de maneira mutuamente independentes e de forma idêntica, falando do ponto de vista estatístico.

Para se analisar o desempenho da arquitetura FDDI quanto a transmissão e recepção de vídeo e áudio, serão observados os parâmetros e as características do ambiente, descritos na tabela 6.4 e na tabela 6.5.

Para a realização da avaliação do desempenho utilizarem-se os softwares SunNet Manager 2.2 e o serviço de análise de *performance* do sistema operacional Windows NT Server 3.5.

<b>Características do anel para realização das avaliações</b>
Número de estações transmitindo
Distância total do anel (Kilômetro)
<b>Características do vídeo compactado</b>
Taxa de geração de vídeos (Mbits/s)
Taxa de geração de <i>frames</i> de vídeo ( Função exponencial com média em milisegundos)
Tamanho médio <i>frame</i> de vídeo
Média número <i>frames</i> /vídeo
<b>Características do áudio compactado</b>
Taxa de geração de áudio (Mbits/s)
Taxa de geração de <i>frames</i> de áudio (Função exponencial com média em milisegundos)
Tamanho médio <i>frame</i> de áudio
Média número de <i>frames</i> /áudio
<b>Características FDDI</b>
Valor TVX (milisegundos)
% alocação da largura de banda para transmissões síncronas
Valor TTRT (Target Token Rotation Timer em milisegundos)
Valor do T_Notify
D_MAX (Latência máxima do anel em milisegundos)
Token_Time (Tempo de transmissão da permissão em milisegundos)
LER_Cutoff
LER_Alarm
No. vezes que a estação recebeu o token
No. expirações do TRT
No. expirações do TVX
<b>Características do ambiente (vídeo compactado)</b>
Tempo máximo de espera para acesso ao meio de transmissão em milisegundos
Taxa de erro por pacote ( < 10 ** -9) [Legay1991]

Taxa de erro por bit ( $< 10^{-6}$ ) [Legay1991]
Taxa média de erro causado por <i>frames</i> que foram descartados por falta de espaço no buffer local
<b>Características do ambiente (áudio compactado)</b>
Tempo máximo de espera para acesso ( $\leq 1$ milissegundo) [Legay1991]
Taxa erro por pacote ( $< 10^{-9}$ ) [Legay1991]
Taxa erro por bit ( $< 10^{-6}$ ) [Legay1991]
Taxa média de erro causado por <i>pacotes</i> que foram descartados por falta de espaço no buffer local
<b>Comportamento do anel FDDI</b>
Média throughput ( Mbps )
% <i>frames</i> que tiveram timeout
% erro de estabelecimento de conexão

Tabela 6.4 - Parâmetros e características do ambiente de gerenciamento

<b>Características do ambiente Windows NT</b>
Capacidade do buffer (número de <i>frames</i> )
<b>Memória</b>
Páginas de memória escritas / segundo
Páginas de memória lidas / segundo
Número de bytes correntes na memória <i>cache</i> residente
Limite do número de bytes da memória <i>cache</i>
Número de bytes da memória <i>cache</i>
Número de bytes de memória disponíveis
<b>Disco físico</b>
% Tempo de disco
% Tempo de leitura do disco
% Tempo escrita no disco
Número de bytes / segundo
Número de escritas / segundo
Número de leituras / segundo
Tamanho da fila do disco

<b>Processador</b>
% Tempo utilizado pelo modo usuário
<b>Servidor FTP</b>
Número de bytes recebidos / segundo
Número de bytes enviados / segundo
Número total de bytes / segundo
Número de arquivos recebidos
Número de arquivos enviados
Número total de arquivos

Tabela 6.5 - Características do ambiente Windows NT

Através das medições aproximadas desses parâmetros mostrados nas tabelas 6.4 e 6.5 e através da técnica de simulação, que será descrita posteriormente, deseja-se analisar os seguintes fatores :

- Diferenças de comportamento de transmissão síncrona e assíncrona;
- Comportamento da transmissão em situações de baixo tráfego e alto tráfego;
- Comportamento da rede com relação aos requisitos de transmissão de vídeos e áudios em uma rede de computadores;
- Avaliação do comportamento da rede quando se transmite vídeo;
- Avaliação do comportamento da rede quando se transmite áudio;
- Número de estações suportadas quando se transmite sinais de vídeo e de áudio e há congestionamento na rede ;
- Resultados para tamanho de fila e atraso de *token*;
- Sensibilidade para dimensionamento do meio e latência das estações;
- Dimensionamento de *buffer* para recepção dos vídeos e áudios;
- Análise do método de acesso por *token* para transferência de vídeos e áudios compactados;
- Análise da *performance* de um *Backbone* FDDI em situações semelhantes às das coletas de dados e situações representadas nas simulações.

## 6.5 - DESCRIÇÃO DO SOFTWARE UTILIZADO PARA COMPRESSÃO E REPRODUÇÃO DOS VÍDEOS E ÁUDIOS NO PROJETO

O software utilizado para codificação dos sinais de vídeo foi o MPEG-2 Encoder / Decoder, Versão 1.1, Junho 1994. Uma das grandes vantagens deste software é que ele permite a utilização de múltiplos processadores no processo de codificação do vídeo, melhorando sensivelmente o tempo de processamento. Entretanto, a *performance* da codificação depende também da velocidade individual de cada processador, do sistema de arquivo e da rede de computadores.

Esse software foi desenvolvido conforme o padrão ISO/IEC 13818 Part 2 (CD: Nov. 1993, DIS : March 1994) pela ISO e IEC, com colaboração do ITU segundo recomendação H.261. Esse software suporta :

- Geração de taxas de bits constantes;
- Codificação baseada no modelo MPEG-2 test model 5 (TM5) rev. 2;
- Vídeos com varredura progressiva e entrelaçada;
- Geração de sequências de quadros no padrão ISO/IEC 11172-2 (MPEG-I);
- Formatos de arquivos de entrada : YUV separado, YUV combinado e PPM (*Portable PixMap*);
- Verificação da conformidade dos parâmetros do usuário com padrão MPEG.

Opções não suportadas :

- Extensões escaláveis;
- Verificação do número máximo de bits por macrobloco;
- Cancelamento de vetores de movimento;
- Edição do vídeo codificado;
- Taxa de bits variável;
- Controle de variação de taxa de cenas.

O software utilizado para visualização dos vídeos durante os testes realizados no desenvolvimento do projeto foi o MPEGPLAY V1.50 de Michael Simmons [I1].

Este software foi executado no ambiente Windows NT podendo mostrar arquivos do padrão MPEG que incluem *frames* tipo "P" e *frames* tipo "B", com resolução de até 352 x 288

*pixels*. Esse software possui várias opções quanto a cor do vídeo : (1) mono; (2) tons de cinza; (3) *color* e (4) *full color*. O software manipula arquivos com tamanhos maiores que 1 Mbyte.

O software utilizado para reprodução do áudio é o Xing Sound MPEG Audio Player da Xing Technology Corporation [Xing1994]. Esse software é executado no ambiente Windows e reproduz áudio com melhor qualidade caso a placa de som utilizada, possua o chip Analog Device's 2115 DSP (Digital Signal Processor). Neste caso o áudio MPEG é enviado diretamente para a placa de áudio não passando pelo processador.

Para codificação do áudio para o padrão MPEG, o software que se utilizou foi o MPEG Encoder e o MPEG Recorder, também da Xing Technology Corporation. O esquema de codificação utilizado foi o nível 2 do padrão de áudio do MPEG.

## **6.6 - SOFTWARE DESENVOLVIDO COMO INTERFACE PARA CRIAÇÃO / VISUALIZAÇÃO / TRANSMISSÃO / RECEPÇÃO DE VÍDEOS E ÁUDIOS**

Foi desenvolvido nesse projeto um software na linguagem Borland C++ 4.0. Este software foi programado para o sistema operacional Windows NT (32 bits). Seu objetivo é fornecer um ambiente onde os usuários possam criar e manipular vídeos, áudios e imagens, além de poder transmiti-los e recebê-los, desde que esteja utilizando o protocolo de transporte e de rede TCP/IP [Notare1994].

Para criação do arquivo utiliza-se a opção (Arquivo/Novo) e obtém-se um *menu* com as seguintes opções : Desenho/Scanner/Animação/Vídeo/Áudio. Assim, dependendo do tipo de criação que se deseja realizar, é chamado o software apropriado.

Para visualização do arquivo não é necessário saber onde se encontra o arquivo ou qual o software que o gerou. Através da opção (Arquivo/Abrir) escolhe-se o arquivo e através da extensão deste é chamado automaticamente o software que o gerou, sendo mostrado imediatamente o vídeo, áudio ou imagem.

Para a transmissão do arquivo, escolhe-se a opção Transmissão que possui como sub menu Envia/Recebe. Após a escolha é chamado o protocolo FTP (File Transfer Protocol) para que seja enviado ou recebido o arquivo.

Com esse software, um usuário pode, facilmente, acessar um servidor *Anonymous* Windows NT de vídeo e áudio que está sendo criado na Universidade Federal de Santa Catarina. Para isso, o usuário deve possuir os softwares Windows, Windows for Workgroup ou Windows NT da Microsoft instalado em sua máquina, o software descrito anteriormente para manipulação e transmissão dos vídeos e áudios, placa VGA, de preferência Super VGA colorida e qualquer placa de som (8 bits ou 16 bits) para que sejam reproduzidos os áudios.

A listagem do projeto desenvolvido pode ser encontrado no anexo I.

Na figura 6.5 e 6.6 pode-se visualizar a janela principal do software e a janela de transmissão e recepção de vídeos e áudios.

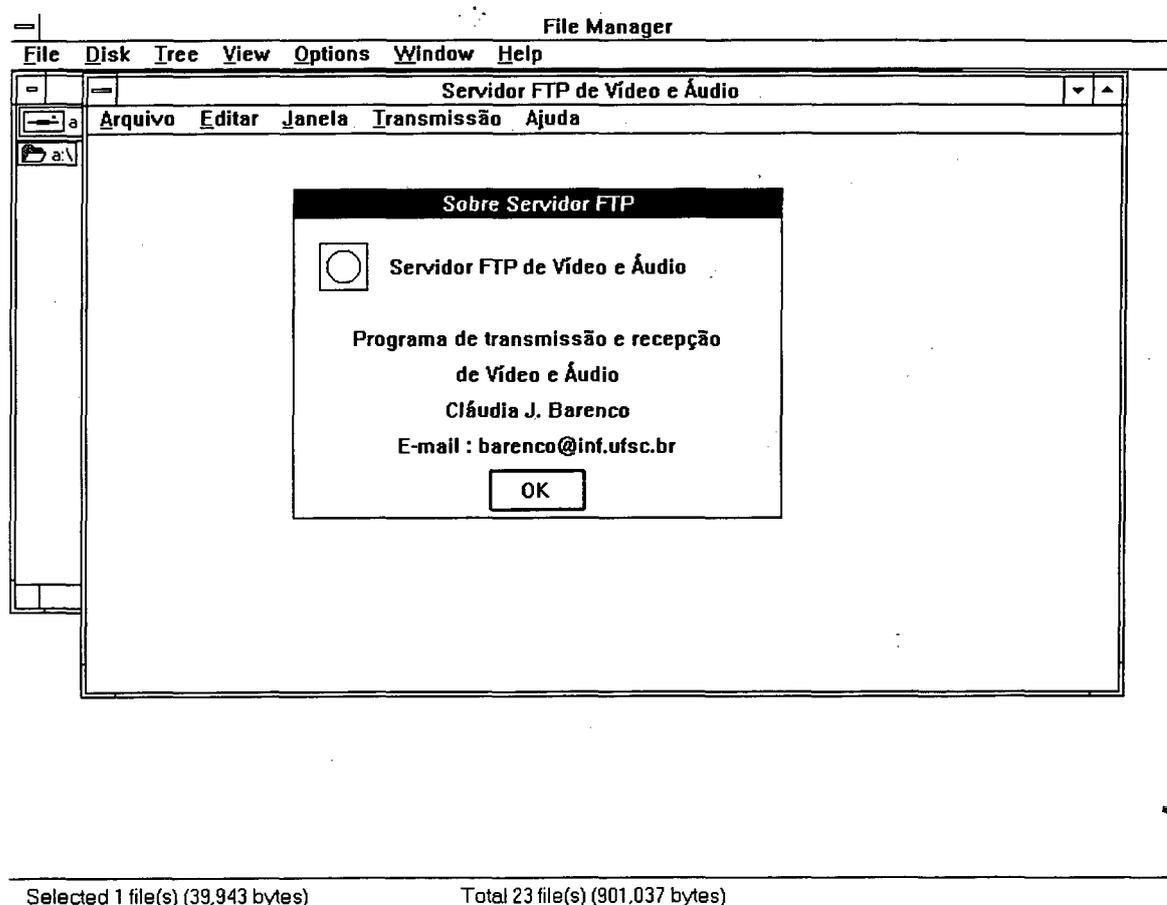


Figura 6.5 - Janela principal do software desenvolvido

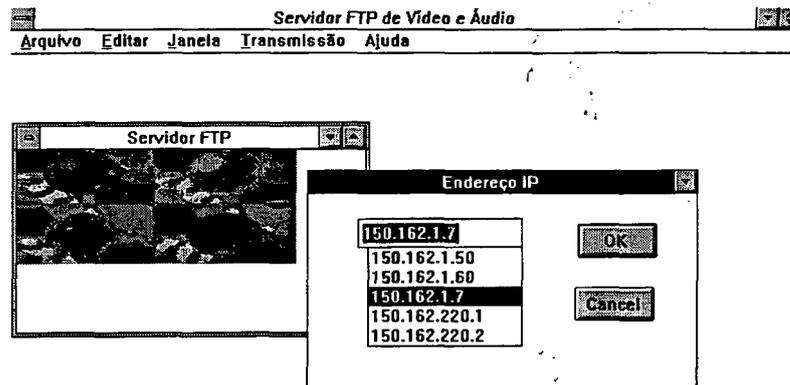


Figura 6.6 - Janela de transmissão e recepção de vídeos e áudios

## 6.7 - DESCRIÇÃO DO MODELO DE SIMULAÇÃO

Na arquitetura FDDI dois tipos de transmissões são possíveis :

- Síncrona, onde é garantida a largura de banda e conseqüentemente é ideal para transmissões onde a variável tempo é crítica e há sensibilidade quanto ao atraso de transmissão;
- Assíncrona, onde sua alocação só é realizada depois de terem havido as transmissões síncronas, sendo assim muito dependente da situação de congestionamento ou não da rede.

Por existirem essas classes, a tecnologia FDDI é muito conveniente para transmissões de multimídia, onde dados como áudio e vídeo, que são variáveis críticas com relação ao tempo, competem com a transferência de dados, como transferência de arquivos, que possuem uma prioridade menor.

O modelo simulado segue as características descritas na tabela 7.18 e o ambiente descrito na figura 6.8. Esse modelo está baseado nos estudos de [Ranai1994].

A figura 6.7 mostra um fluxograma das operações do FDDI MAC :

Como consequência dos estudos de [Sevcik1987], duas propriedades fundamentais do FDDI foram observadas :

- O máximo TRT do anel não pode exceder duas vezes o TTRT;
- A média do TRT não pode exceder o TTRT.

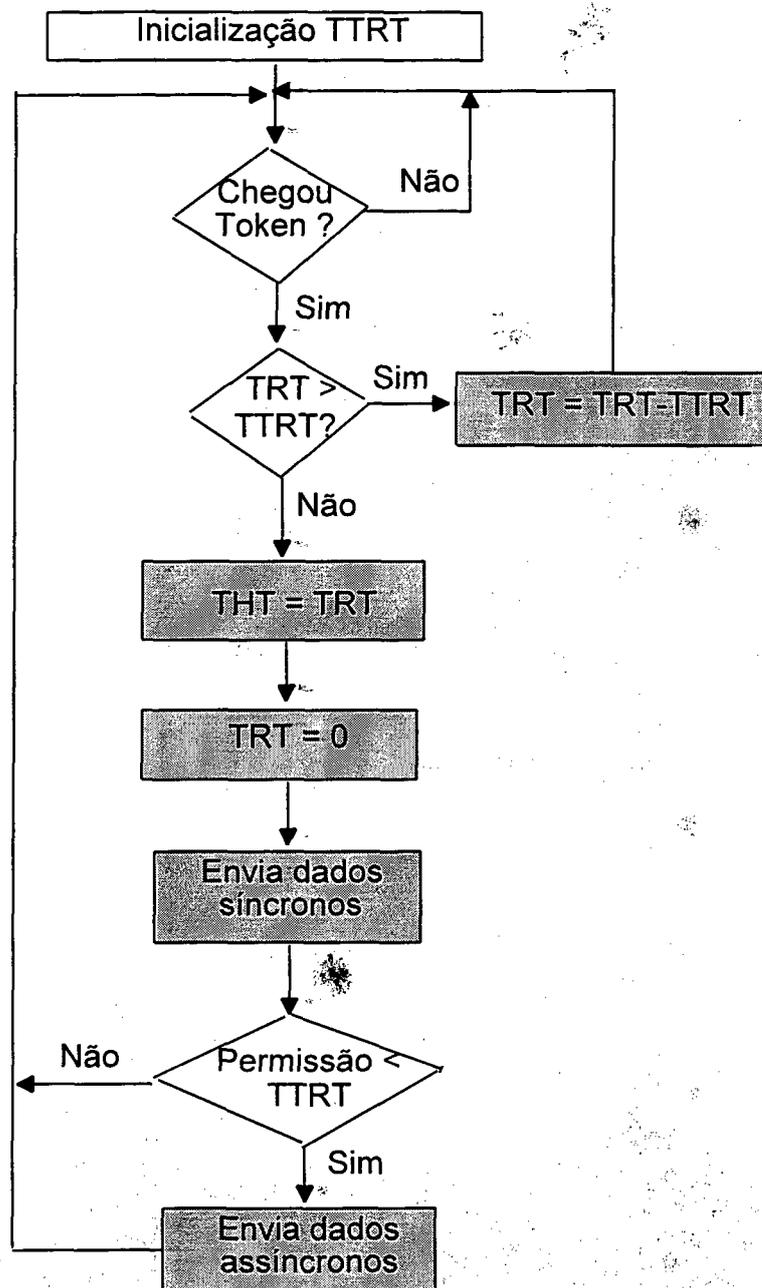


Figura 6.7 - Fluxograma do nível MAC FDDI

Para a transmissão assíncrona a simulação realizada foi no modo não restrito, ou seja, todas as estações podem transmitir nesse modo.

Para as transmissões assíncronas não foi simulada a característica de prioridades, já que a arquitetura do projeto não possui esta facilidade.

No modelo criado não se levou em consideração erros como : erro de conexão, erro de bit, erro de pacote, etc ..., já que não tínhamos interesse de estudar essas situações.

O modelo foi descrito nos softwares de simulação Arena Versão 1.2 e SIMAN V [Pegden1990] da Systems Modeling Corporation.

Esses softwares suportam :

1. Funções estatísticas e distribuições de probabilidade para a geração dos dados;
2. Filas com controle de acesso para gerenciamento do limite de alocação dos recursos;
3. Funções para obtenção precisa de resultados estatísticos;
4. Características automáticas de filas (Ex : tempo de fila, tamanho da fila, etc ...);
5. Capacidade gráfica interativa para construção de modelos;
6. Um depurador interativo para controle e monitoria da execução da simulação;

A noção de tempo no modelo é realizada através da passagem sequencial dos pacotes e seu enfileiramento.

Para cada objeto criado, tem-se um conjunto de parâmetros que os representa. Por exemplo temos : tempo de chegada, tempo de fila, tempo máximo de espera para acesso ao meio, etc ...

Uma observação a ser feita é que nesse modelo de simulação não se levou em consideração o tempo gasto para empacotamento dos vídeos e áudios.

Uma representação geral do modelo, pode ser visualizada na figura 6.8.

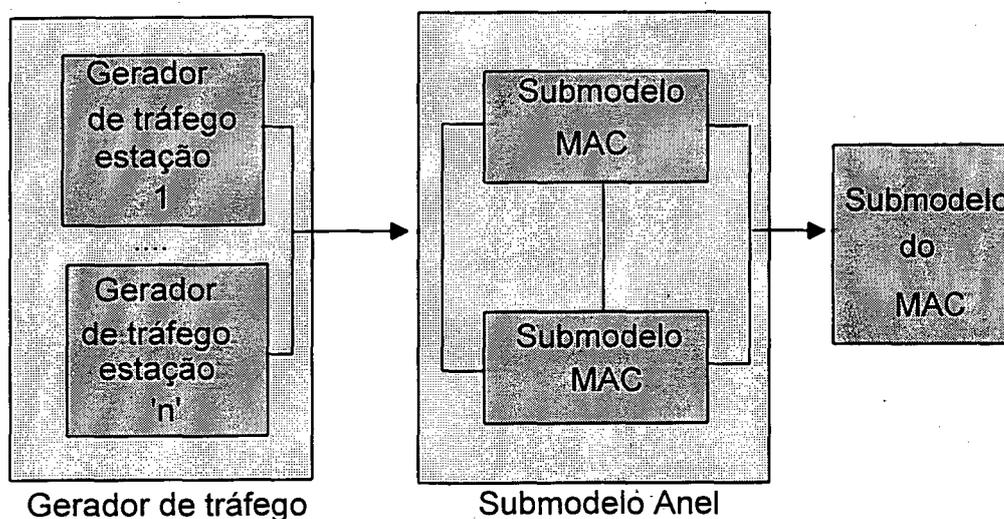


Figura 6.8 - Representação geral do modelo

Na figura 6.9 mostra-se a representação gráfica do modelo simulado.

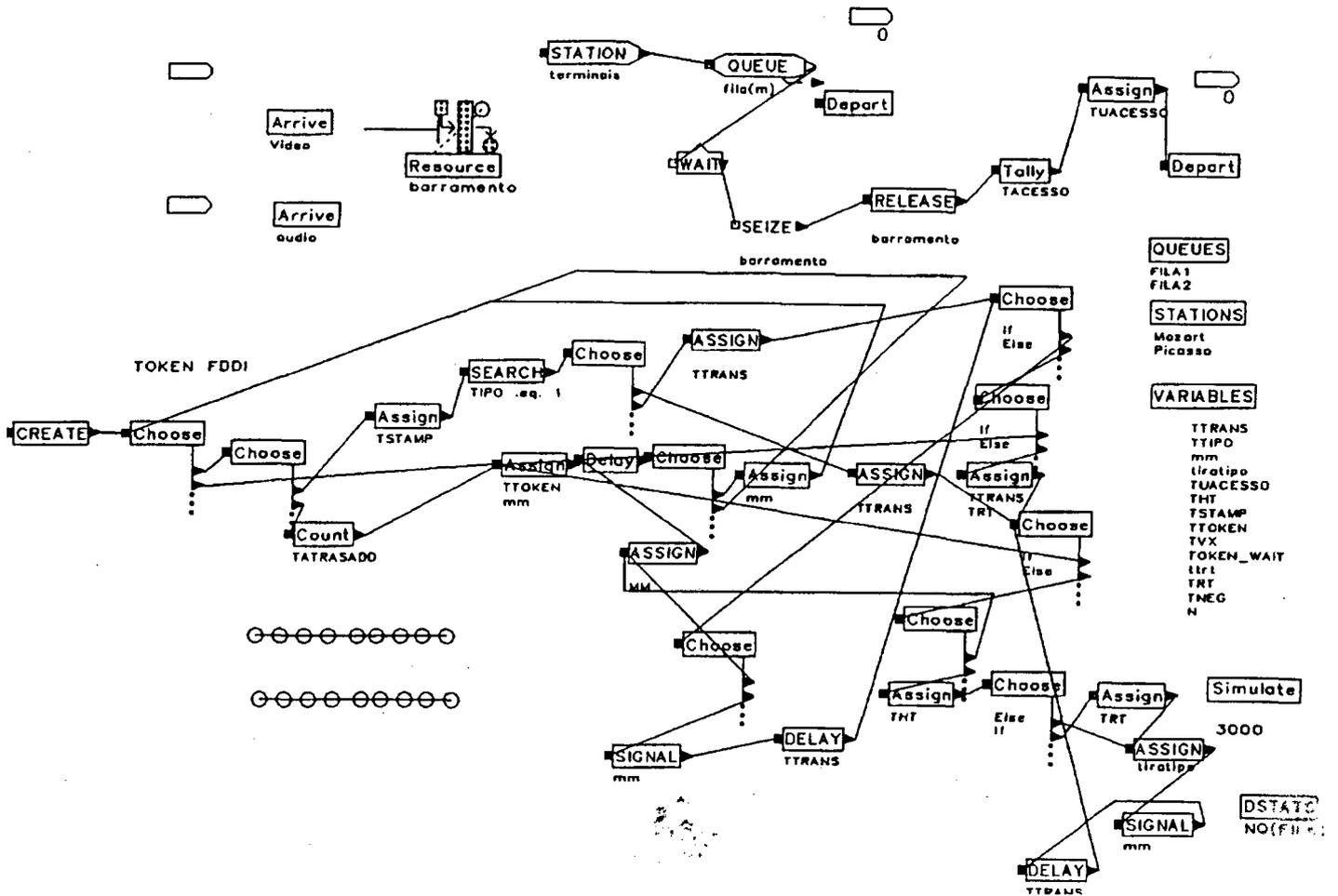


Figura 6.9 - Representação gráfica do modelo simulado

O modelo consiste em um gerador de tráfego, um submodelo do anel e um modelo do MAC da estação que está acessando o meio de transmissão.

O gerador de tráfego gera os dados com um fluxo de aproximadamente 100 Mbits/s. O submodelo anel, simula a passagem do token e a transmissão física do token e o submodelo MAC manipula o mecanismo de prioridade da rede FDDI de cada estação simulada.

No anexo III pode-se visualizar o código fonte do modelo simulado.

Através desse modelo de simulação conseguiu-se obter resultados sobre o comportamento do *Backbone* FDDI onde estejam conectadas 2, 10 e 20 estações multimídias, que estejam gerando vídeos e áudios a uma taxa de 4 Mbits/s, taxa de compactação de vídeos segundo o padrão MPEG II.

Não foi possível realizar a simulação com mais de 20 estações porque o software Arena somente permite 19.000 entidades presentes no modelo ao mesmo tempo, valor que foi extrapolado quando colocou-se 30 estações no modelo. Apesar disso, 20 estações é uma quantidade razoável, levando-se em consideração que foram simuladas situações onde 2, 10 ou 20 fossem pontos de um sistema de teleconferência [VIDICOMA1993], [VIDICOMB1993], [VIDICOMC1993],[VIDICOMD1993] que gerassem vídeos compactados a 4 Mbits/s.

## Capítulo 7

### ANÁLISE DOS PARÂMETROS MEDIDOS

#### 7.1 - INTRODUÇÃO

Para a coleta dos parâmetros a serem medidos, foram criados vários ambientes diferentes. Coletou-se dados em períodos considerados como de baixo tráfego: à noite e final de semana e no período da tarde, que é considerado um período de alto tráfego.

A escolha dos vídeos e áudios a serem transmitidos ocorreu aleatoriamente e suas transmissões aconteceram de maneira contínua durante o período de 15 minutos. Foi escolhido o período de 15 minutos porque a unidade de tempo de muitas variáveis a serem coletadas é em nanosegundos, assim ficaria impraticável de se analisar manualmente os dados se o período de transmissão fosse muito longo.

O tamanho médio em bytes dos vídeos compactados era de 830,40 Kbytes e dos áudios compactados era de 359,079 Kbytes. Os vídeos tinham duração média de 25 segundos e resolução de 160 por 120 pontos (horizontal x vertical).

As transmissões foram realizadas da estação 'Bach' para a estação 'Picasso'. A conexão dessas estações ao *backbone* FDDI está mostrada na figura 6.2. Essas estações estão distantes aproximadamente de 1 Km.

No momento das transmissões a estação 'Picasso' estava dedicada ao serviço de servidor FTP, ou seja, nenhum outro processo do usuário estava sendo executado.

Em nenhuma das transmissões ocorreu algum problema de descontinuidade de transmissão.

Nas tabelas a seguir, quando se cita a palavra *frames*, quer se dizer os quadros transmitidos pelo nível MAC da arquitetura FDDI.

Para a realização das três simulações alguns parâmetros a serem atribuídos, foram obtidos no ambiente de gerenciamento e no sistema operacional Windows NT. Esses parâmetros se

mantiveram constantes durante todas as simulações. Na tabela 7.18 estão descritos esses parâmetros.

## 7.2 - DESCRIÇÃO E ANÁLISE DOS DADOS COLETADOS NO AMBIENTE DE GERENCIAMENTO

A seguir nas tabelas 7.1 à 7.16 serão mostrados os dados coletados em diversas situações diferentes.

Nas tabelas 7.1 e 7.2 o ambiente para a coleta dos dados era o seguinte :

- Transmissão realizada dia 18/03/95 às 18:28;
- Transmitiu-se vídeos compactados pelo padrão MPEG I;
- A duração da transmissão foi de 15 minutos;
- Por ser um dia de sábado, final de semana, foi classificada como uma situação de baixo tráfego.

<b>Características do anel para realização das avaliações</b>	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
<b>Características do vídeo compactado</b>	
Taxa de geração de vídeos (Mbits/s)	1,64736
Taxa de geração de <i>frames</i> de vídeos ( Função exponencial com média em milisegundos)	E XP(1.9)
Tamanho médio <i>frames</i> de vídeo (bytes)	427
Média número <i>frames</i> /vídeo	791
<b>Características FDDI</b>	
Valor TVX (milisegundos)	1,34220
% alocação da largura de banda para transmissões síncronas	70
Valor operacional do TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0,167772160
Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,042066869

LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8
No. vezes que a estação recebeu o token	13.683.479
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (vídeo compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milisegundos)	2,621440
Taxa de erro por pacote ( < 10 -9) [Legay1991]	0
Taxa de erro por bit ( < 10 -6) [Legay1991]	0
Taxa média de erro causado por pacote que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	
Média throughput ( Mbps )	100
% frames que tiveram timeout	0,00001
Taxa de erro de estabelecimento de conexão	10 ** -12

Tabela 7.1 - Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 18/03/95 às 18:28)

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de vídeo)	137.155
<b>Memória</b>	
Páginas de memória escritas / segundo	0,034
Páginas de memória lidas / segundo	0,496
Número de bytes correntes na memória cache residente	704.512,00
Limite do número de bytes da memória cache	7.749.632,00
Número de bytes da memória cache	1.642.492,00
Número de bytes de memória disponíveis	9.834.496,00
<b>Disco físico</b>	
% Tempo de disco	73,571
% Tempo de leitura do disco	3,213

% Tempo escrita no disco	70,358
Número de bytes / segundo	215.456,375
Número de escritas / segundo	5,186
Número de leituras / segundo	0,526
Tamanho da fila do disco	0,00
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	42,947
<b>Servidor FTP</b>	
Número de bytes recebidos / segundo	205.842,156
Número de bytes enviados / segundo	142,047
Número total de bytes / segundo	205.984,203
Número de arquivos recebidos	665,00
Número de arquivos enviados	0,00
Número total de arquivos	665,00

Tabela 7.2 - Transmissão de vídeos compactados (Dados coletados pelo sistema operacional Windows NT - dia : 18/03/95 às 18:28)

Nas tabelas 7.3 e 7.4 o ambiente para a coleta dos dados era o seguinte :

- Transmissão realizada dia 19/03/95 às 16:11;
- Transmitiu-se áudios compactados pelo padrão MPEG I;
- A duração da transmissão foi de 15 minutos;
- Por ser um dia de domingo, final de semana, foi classificada como uma situação de baixo tráfego.

<b>Características do anel para realização das avaliações</b>	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
<b>Características do áudio compactado</b>	
Taxa de geração de áudios (Mbits/s)	1,198568

Taxa de geração de <i>frames</i> áudios ( Função exponencial com média em milisegundos)	EXP(2.7)
Tamanho médio <i>frame</i> de áudio (bytes)	438
Média número <i>frames</i> /áudio	839
<b>Características FDDI</b>	
Valor TVX (milisegundos)	1.34220
% alocação da largura de banda para transmissões síncronas	70
Valor TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0,167772160
Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,081948258
LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8
No. vezes que a estação recebeu o token	13.422.548
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (áudio compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milisegundos)	2,621440
Taxa de erro por pacote ( < 10 -9) [Legay1991]	0
Taxa de erro por bit ( < 10 -6) [Legay1991]	0
Taxa média de erro causado por pacotes que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	
Média throughput ( Mbps )	100
% frames que tiveram timeout	0,00001
Taxa de erro de estabelecimento de conexão	10 ** -13

Tabela 7.3 - Transmissão de áudios compactados ( Dados coletados peio software SunNet Manager - dia : 19/03/95 às 16:11 )

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de áudio)	184.172
<b>Memória</b>	

Páginas de memória escritas / segundo	0,017
Páginas de memória lidas / segundo	0,067
Número de bytes correntes na memória <i>cache</i> residente	2.584.576,00
Limite do número de bytes da memória <i>cache</i>	7.749.632,00
Número de bytes da memória <i>cache</i>	4.476.928,00
Número de bytes de memória disponíveis	5.984.256,00
<b>Disco físico</b>	
% Tempo de disco	14,395
% Tempo de leitura do disco	0,192
% Tempo escrita no disco	14,203
Número de bytes / segundo	68.700,531
Número de escritas / segundo	1,977
Número de leituras / segundo	0,069
Tamanho da fila do disco	0,00
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	43,471
<b>Servidor FTP</b>	
Número de bytes recebidos / segundo	149.821,922
Número de bytes enviados / segundo	366,757
Número total de bytes / segundo	150.188,688
Número de arquivos recebidos	1.445,00
Número de arquivos enviados	0,00
Número total de arquivos	1.445,00

Tabela 7.4 - Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 19/03/95 às 16:11)

Nas tabelas 7.5 e 7.6 o ambiente para a coleta dos dados era o seguinte :

- Transmissão realizada dia 20/03/95 às 16:03;
- Transmitiu-se vídeos compactados pelo padrão MPEG I;

- A duração da transmissão foi de 15 minutos;
- Foi classificada como uma situação de alto tráfego, por ter sido realizada em um horário já conhecido como de maior tráfego.

<b>Características do anel para realização das avaliações</b>	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
<b>Características do vídeo compactado</b>	
Taxa de geração de vídeos (Mbits/s)	1,601584
Taxa de geração de frames de vídeo ( Função exponencial com média em milisegundos)	EXP(2.0)
Tamanho médio <i>frame</i> de vídeo (bytes)	421
Média número <i>frames</i> /vídeo	802
<b>Características FDDI</b>	
Valor TVX (milisegundos)	1,34220
% alocação da largura de banda para transmissões síncronas	70
Valor TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0 ,167772160
Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,0
LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8
No. vezes que a estação recebeu o token	14.628.429
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (vídeo compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milisegundos)	2,621440
Taxa de erro por pacote ( < 10 -9) [Legay1991]	0
Taxa de erro por bit ( < 10 -6) [Legay1991]	0
Taxa média de erro causado por <i>frames</i> que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	

Média throughput ( Mbps )	100
% frames que tiveram timeout	0
Taxa de erro de estabelecimento de conexão	10 ** -13

Tabela 9.5 - Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 20/03/95 às 16:03 )

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de áudio)	143.216
<b>Memória</b>	
Páginas de memória escritas / segundo	0,025
Páginas de memória lidas / segundo	0,819
Número de bytes correntes na memória <i>cache</i> residente	1.294.336,00
Limite do número de bytes da memória <i>cache</i>	8.888.320,00
Número de bytes da memória <i>cache</i>	3.063.808,00
Número de bytes de memória disponíveis	6.344.704,00
<b>Disco físico</b>	
% Tempo de disco	80,830
% Tempo de leitura do disco	6,335
% Tempo escrita no disco	74,494
Número de bytes / segundo	213.336,969
Número de escritas / segundo	5,349
Número de leituras / segundo	0,859
Tamanho da fila do disco	0,00
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	30,377
<b>Servidor FTP</b>	
Número de bytes recebidos / segundo	200.298,969

Número de bytes enviados / segundo	127,327
Número total de bytes / segundo	200.426,29
Número de arquivos recebidos	2.116,00
Número de arquivos enviados	3,00
Número total de arquivos	2.119,00

Tabela 7.6 - Transmissão de vídeos compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 20/03/95 às 16:03)

Nas tabelas 7.7 e 7.8 o ambiente para a coleta dos dados era o seguinte :

- Transmissão realizada dia 20/03/95 às 22:13;
- Transmitiu-se vídeos compactados pelo padrão MPEG I;
- A duração da transmissão foi de 15 minutos;
- Foi classificada como uma situação de baixo tráfego, por ter sido realizada à noite, considerado um horário de menor tráfego.

<b>Características do anel para realização das avaliações</b>	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
<b>Características do vídeo compactado</b>	
Taxa de geração de vídeos (Mbits/s)	2,242752
Taxa de geração de <i>frames</i> de vídeo ( Função exponencial com média em milisegundos)	EXP(1.5)
Tamanho médio <i>frame</i> de vídeo (bytes)	466
Média número <i>frames</i> /vídeo	725
<b>Características FDDI</b>	
Valor TVX (milisegundos)	1,34220
% alocação da largura de banda para transmissões síncronas	70
Valor TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0,167772160
Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,101919080
LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8

No. vezes que a estação recebeu o token	12.757.985
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (vídeo compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milissegundos)	2,621440
Taxa de erro por pacote ( < 10 -9) [Legay1991]	0
Taxa de erro por bit ( < 10 -6) [Legay1991]	0
Taxa média de erro causado por <i>frames</i> que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	
Média throughput ( Mbps )	100
% frames que tiveram timeout	0,0000079
Taxa de erro de estabelecimento de conexão	10 ** -13

Tabela 7.7 - Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 20/03/95 às 22:13 )

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de vídeo)	92.502
<b>Memória</b>	
Páginas de memória escritas / segundo	0,034
Páginas de memória lidas / segundo	0,781
Número de bytes correntes na memória <i>cache</i> residente	6.819.840,00
Limite do número de bytes da memória <i>cache</i>	8.888.320,00
Número de bytes da memória <i>cache</i>	7.749.632,00
Número de bytes de memória disponíveis	3.387.392,00
<b>Disco físico</b>	
% Tempo de disco	100,00
% Tempo de leitura do disco	5,103
% Tempo escrita no disco	100,00
Número de bytes / segundo	291.427,219

Número de escritas / segundo	7,197
Número de leituras / segundo	0,815
Tamanho da fila do disco	1,00
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	2,961
<b>Servidor FTP</b>	
Número de bytes recebidos / segundo	280.344,844
Número de bytes enviados / segundo	208,284
Número total de bytes / segundo	280.553,125
Número de arquivos recebidos	2.480,00
Número de arquivos enviados	5,00
Número total de arquivos	2.485,00

Tabela 7.8 - Transmissão de vídeos compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 20/03/95 às 22:13)

Nas tabelas 7.9 e 7.10 o ambiente de coleta de dados era o seguinte :

- Transmissão realizada dia 21/03/95 às 15:58;
- Transmitiu-se vídeos compactados pelo padrão MPEG I;
- A duração da transmissão foi de 15 minutos;
- Foi classificada como uma situação de alto tráfego, por ter sido realizada à tarde, considerado um horário de grande tráfego.

<b>Características do anel para realização das avaliações</b>	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
<b>Características do vídeo compactado</b>	
Taxa de geração de vídeos (Mbits/s)	1,033024
Taxa de geração de <i>frames</i> de vídeo ( Função exponencial com média em milisegundos)	EXP(3.6)
Tamanho médio <i>frame</i> de vídeo (bytes)	496

Média número <i>frames</i> /vídeo	741
<b>Características FDDI</b>	
Valor TVX (milisegundos)	1,34220
% alocação da largura de banda para transmissões síncronas	70
Valor TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0,167772160
Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,023047664
LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8
No. vezes que a estação recebeu o token	11.464.616
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (vídeo compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milisegundos)	2,621440
Taxa de erro por pacote ( < 10 <sup>-9</sup> ) [Legay1991]	0
Taxa de erro por bit ( < 10 <sup>-6</sup> ) [Legay1991]	0
Taxa média de erro causado por <i>frames</i> que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	
Média throughput ( Mbps )	100
% frames que tiveram timeout	0,0000168
Taxa de erro de estabelecimento de conexão	10 ** -13

Tabela 7.9 - Transmissão de vídeos compactados ( Dados coletados pelo software SunNet Manager - dia : 21/03/95 às 15:58 )

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de vídeo)	188.269
<b>Memória</b>	
Páginas de memória escritas / segundo	0,017
Páginas de memória lidas / segundo	0,092

Número de bytes correntes na memória <i>cache</i> residente	1.904.640,00
Limite do número de bytes da memória <i>cache</i>	5.775.360,00
Número de bytes da memória <i>cache</i>	3.235.840,00
Número de bytes de memória disponíveis	5.955.584,00
<b>Disco físico</b>	
% Tempo de disco	13,902
% Tempo de leitura do disco	0,244
% Tempo escrita no disco	13,657
Número de bytes / segundo	73.853,547
Número de escritas / segundo	1,971
Número de leituras / segundo	0,099
Tamanho da fila do disco	0 ,000
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	36,144
<b>Servidor FTP</b>	
Número de bytes recebidos / segundo	165.679,14
Número de bytes enviados / segundo	402,228
Número total de bytes / segundo	166.081,359
Número de arquivos recebidos	741,00
Número de arquivos enviados	2,00
Número total de arquivos	743,00

Tabela 7.10 - Transmissão de vídeos compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 21/03/95 às 15:58)

Nas tabelas 7.11 e 7.12 o ambiente de coleta de dados era o seguinte :

- Transmissão realizada dia 22/03/95 às 15:59;
- Transmitiu-se áudios compactados pelo padrão MPEG I;
- A duração da transmissão foi de 15 minutos;
- Foi classificada como uma situação de alto tráfego, por ter sido realizada à tarde, considerado um horário de grande tráfego.

<b>Características do anel para realização das avaliações</b>	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
<b>Características do áudio compactado</b>	
Taxa de geração de áudios (Mbits/s)	1,425432
Taxa de geração de <i>frames</i> de áudio ( Função exponencial com média em milisegundos)	EXP(3.0)
Tamanho médio <i>frame</i> de áudio (bytes)	567
Média número <i>frames</i> /áudio	648
<b>Características FDDI</b>	
Valor TVX (milisegundos)	1,34220
% alocação da largura de banda para transmissões síncronas	70
Valor TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0,167772160
Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,055178259
LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8
No. vezes que a estação recebeu o token	9.515.299
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (áudio compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milisegundos)	2,621440
Taxa de erro por pacote ( < 10 <sup>-9</sup> ) [Legay1991]	0
Taxa de erro por bit ( < 10 <sup>-6</sup> ) [Legay1991]	0
Taxa média de erro causado por <i>frames</i> que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	
Média throughput ( Mbps )	100
% frames que tiveram timeout	0,000240
Taxa de erro de estabelecimento de conexão	10 ** -12

Tabela 7.11 - Transmissão de áudios compactados ( Dados coletados pelo software SunNet Manager - dia : 22/03/95 às 15:59 )

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de áudio)	128.562
<b>Memória</b>	
Páginas de memória escritas / segundo	0,017
Páginas de memória lidas / segundo	0,092
Número de bytes correntes na memória <i>cache</i> residente	1.904.640,00
Limite do número de bytes da memória <i>cache</i>	5.775.360,00
Número de bytes da memória <i>cache</i>	3.235.840,00
Número de bytes de memória disponíveis	5.955.584,00
<b>Disco físico</b>	
% Tempo de disco	13,902
% Tempo de leitura do disco	0,244
% Tempo escrita no disco	13,657
Número de bytes / segundo	73.853,547
Número de escritas / segundo	1,971
Número de leituras / segundo	0,099
Tamanho da fila do disco	0,000
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	36,144
<b>Servidor FTP</b>	
Número de bytes recebidos / segundo	165.679,14
Número de bytes enviados / segundo	402,228
Número total de bytes / segundo	166.081,359
Número de arquivos recebidos	741,00
Número de arquivos enviados	2,00
Número total de arquivos	743,00

Tabela 7.12 - Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 22/03/95 às 15:59)

Nas tabelas 7.13 e 7.14 o ambiente de coleta de dados era o seguinte :

- Transmissão realizada dia 22/03/95 às 16:21;
- Transmitiu-se áudios compactados pelo padrão MPEG I;
- A duração da transmissão foi de 15 minutos;
- Foi classificada como uma situação de alto tráfego, por ter sido realizada à tarde, considerado um horário de grande tráfego.

<b>Características do anel para realização das avaliações</b>	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
<b>Características do áudio compactado</b>	
Taxa de geração de áudios (Mbits/s)	1,751272
Taxa de geração de <i>frames</i> de áudio ( Função exponencial com média em milisegundos)	EXP(2.5)
Tamanho médio <i>frame</i> de áudio (bytes)	582
Média número <i>frames</i> /áudio	631
<b>Características FDDI</b>	
Valor TVX (milisegundos)	1,34220
% alocação da largura de banda para transmissões síncronas	70
Valor TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0 ,167772160
Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,066156174
LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8
No. vezes que a estação recebeu o token	10.578.900
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (áudio compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milisegundos)	2,621440
Taxa de erro por pacote ( < 10 -9) [Legay1991]	0
Taxa de erro por bit ( < 10 -6) [Legay1991]	0

Taxa média de erro causado por <i>frames</i> que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	
Média throughput ( Mbps )	100
% frames que tiveram timeout	0,0000148
Taxa de erro de estabelecimento de conexão	10 ** -12

Tabela 7.13 - Transmissão de áudios compactados ( Dados coletados pelo software SunNet Manager - dia : 22/03/95 às 16:21 )

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de áudio)	94.814
<b>Memória</b>	
Páginas de memória escritas / segundo	0,007
Páginas de memória lidas / segundo	0,006
Número de bytes correntes na memória <i>cache</i> residente	2.928.640,00
Limite do número de bytes da memória <i>cache</i>	5.775.360,00
Número de bytes da memória <i>cache</i>	4.345.856,00
Número de bytes de memória disponíveis	5.500.928,00
<b>Disco físico</b>	
% Tempo de disco	17,970
% Tempo de leitura do disco	0,002
% Tempo escrita no disco	17,968
Número de bytes / segundo	95.440,922
Número de escritas / segundo	2,518
Número de leituras / segundo	0,005
Tamanho da fila do disco	0 ,000
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	42,156

Servidor FTP	
Número de bytes recebidos / segundo	218.909,500
Número de bytes enviados / segundo	530,945
Número total de bytes / segundo	219.440,453
Número de arquivos recebidos	1.280,00
Número de arquivos enviados	2,00
Número total de arquivos	1.282,00

Tabela 7.14 - Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 22/03/95 às 16:21)

Nas tabelas 7.15 e 7.16 o ambiente de coleta de dados era o seguinte :

- Transmissão realizada dia 22/03/95 às 18:26;
- Transmitiu-se áudios compactados pelo padrão MPEG I;
- A duração da transmissão foi de 15 minutos;
- Foi classificada como uma situação de baixo tráfego, por ter sido realizada à noite, considerado um horário de menor tráfego.

Características do anel para realização das avaliações	
Número de estações transmitindo	1
Distância total do anel (Kilômetro)	5
Características do áudio compactado	
Taxa de geração de áudios (Mbits/s)	1,543624
Taxa de geração de <i>frames</i> de áudio ( Função exponencial com média em milisegundos)	EXP(2.9)
Tamanho médio <i>frame</i> de áudio (bytes)	590
Média número <i>frames</i> /áudio	623
Características FDDI	
Valor TVX (milisegundos)	1,34220
% alocação da largura de banda para transmissões síncronas	70
Valor TTRT (Target Token Rotation Timer em milisegundos)	0,7987200
Valor do T_Notify (milisegundos)	30000
D_MAX (Latência máxima do anel em milisegundos)	0,167772160

Token_Time (Tempo médio de transmissão da permissão em milisegundos)	0,029464788
LER_Cutoff	10 ** -7
LER_Alarm	10 ** -8
No. vezes que a estação recebeu o token	10.588.007
No. expirações do TRT	0
No. expirações do TVX	0
<b>Características do ambiente (áudio compactado)</b>	
Tempo máximo de espera para acesso ao meio de transmissão (milisegundos)	2,621440
Taxa de erro por pacote ( < 10 -9) [Legay1991]	0
Taxa de erro por bit ( < 10 -6) [Legay1991]	0
Taxa média de erro causado por <i>frames</i> que foram descartados por falta de espaço no buffer local	0
<b>Comportamento do anel FDDI</b>	
Média throughput ( Mbps )	100
% frames que tiveram timeout	0,0000700
Taxa de erro de estabelecimento de conexão	10 ** -12

Tabela 7.15 - Transmissão de áudios compactados ( Dados coletados pelo software SunNet Manager - dia : 22/03/95 às 18:26 )

<b>Características do ambiente Windows NT</b>	
Capacidade do buffer (número de frames de áudio)	106.650
<b>Memória</b>	
Páginas de memória escritas / segundo	0,001
Páginas de memória lidas / segundo	0,006
Número de bytes correntes na memória <i>cache</i> residente	2.543.616,00
Limite do número de bytes da memória <i>cache</i>	5.775.360,00
Número de bytes da memória <i>cache</i>	3.960.832,00
Número de bytes de memória disponíveis	6.299.648,00
<b>Disco físico</b>	
% Tempo de disco	16,994

% Tempo de leitura do disco	0,008
% Tempo escrita no disco	16,985
Número de bytes / segundo	86.46,906
Número de escritas / segundo	2,222
Número de leituras / segundo	0,005
Tamanho da fila do disco	0 ,000
<b>Processador</b>	
% Tempo utilizado pelo modo usuário	45,928
<b>Servidor FTP</b>	
Número de bytes recebidos / segundo	192.953,969
Número de bytes enviados / segundo	498,759
Número total de bytes / segundo	193.452,734
Número de arquivos recebidos	1.924,00
Número de arquivos enviados	2,00
Número total de arquivos	1.926,00

Tabela 7.16 - Transmissão de áudios compactados ( Dados coletados pelo sistema operacional Windows NT - dia : 22/03/95 às 18:26)

Na tabela 7.17 mostram-se os valores atribuídos para alocação da largura de banda síncrona e para os parâmetros FDDI dos adaptadores, das estações e do *Hub*.

Esses valores foram utilizados para a coleta de dados das transmissões e para a realização da simulação.

Parâmetro	Descrição	Valor atribuído
SbaPayLoad	Define a largura de banda síncrona para alocação manual e estática	1094 bytes. Valor que corresponde a 70% da largura de banda alocada para transmissões síncronas
T_TransAssinc	Máximo tempo que pode ser negociado com o protocolo SMT para transmissões síncronas	0,025 milissegundos
Min_Segment_Size	Tamanho mínimo do <i>frame</i>	1 byte

Tabela 7.17 - Parâmetros atribuídos para as coletas de dados e para a simulação

Conforme as figuras A.2, A.6, A.8, A.11, A.14, A.17, A.20 e A.24, mostradas no anexo IV, verifica-se que a taxa de erro de conexão durante as diversas transmissões manteve-se entre  $10^{-9}$  e  $10^{-13}$ , uma taxa de erro considerada baixa.

Em nenhuma das situações de transmissão o tempo máximo para recebimento do token foi extrapolado. Conforme as figuras A.1, A.5, A.9, A.12, A.15, A.18, A.21 e A.23, mostradas no anexo III, verifica-se que esse tempo de transição do token também manteve-se na maior parte do tempo constante. Conclui-se que mesmo em situações consideradas como de alto tráfego, não houve congestionamento no *Backbone* FDDI.

Comparando-se o número de vezes que a estação recebeu o token, podemos verificar que nos períodos de alto tráfego, quando se transmitiu áudios gerou-se menor congestionamento na rede do que quando se transmitiu vídeo. Nos períodos de baixo tráfego o comportamento foi idêntico para as transmissões de áudio e de vídeo.

Quanto à taxa de erro de bit e de pacote, podemos verificar nas tabelas 7.1 a 7.16 que não houve nenhum erro, ou seja, a qualidade dos vídeos e áudios continuou a mesma depois de serem transmitidos. Apesar do erro de pacote não ser considerado um grande problema para as mídias de vídeo e áudio, essas taxas de erros serem zero comprovam que o meio físico de fibra ótica ajuda a minimizar erros dessa natureza. No capítulo 2, encontram-se maiores detalhes sobre problemas de erro de bit e pacote em vídeos e áudios.

Nas transmissões realizadas, o *buffer* alocado na estação 'Picasso' para a recepção das transmissões foi de 1024 Kbytes, uma alocação muito boa, já que não houve nenhuma situação de *frames* que foram descartados por falta de espaço no *buffer*.

O percentual de erro causada por *timeout* nas transmissões dos *frames* foi de 0.00003 %. Um percentual considerado muito bom.

Com as análises acima descritas, confirma-se que o meio físico de fibra ótica é um meio de transmissão de baixa taxa de erro e muito rápido.

Em situações semelhantes às analisadas nas coletas de dados, conclui-se que o *Backbone* FDDI atende perfeitamente às necessidades para a transmissão de vídeos e áudios compactados pelo padrão MPEG. No item 7.3, encontram-se novas conclusões em situações diversas às apresentadas aqui.

É importante lembrar que as transmissões foram realizadas em um ambiente local, no campus da UFSC, em um *Backbone* FDDI onde encontram-se instalados apenas 6 concentradores FDDI.

Através dos dados coletados pelo sistema operacional Windows NT da estação 'Picasso' e descritos nas tabelas 7.2, 7.4, 7.6, 7.8, 7.10, 7.12, 7.14 e 7.16 podemos fazer uma análise do desempenho da estação 'Picasso', cuja configuração está descrita no capítulo 6. Essa análise abrange o desempenho do processador, a utilização de memória e o desempenho do acesso à disco rígido.

O percentual médio de tempo utilizado para acesso ao disco rígido durante as coletas de dados, ou seja, durante as transmissões dos vídeos e áudios foi de 32,24 %. Esse percentual representa quase que totalmente o percentual do tempo de escrita no disco, já que os vídeos estavam sendo transmitidos da estação 'Bach' para essa estação.

O processador ficou em média 35,01% do tempo sendo utilizado pelo modo usuário. O modo usuário engloba todas as aplicações e códigos de subsistemas. Nos momentos das coletas de dados a única aplicação que estava sendo executada era o serviço de servidor FTP.

O percentual médio de utilização da memória RAM foi de 70%.

Pode-se concluir que um microcomputador com as configurações da estação 'Picasso' talvez não atenda a demanda de recepção de vídeos e áudios quando muitos usuários estiverem gerando vídeos e áudios a uma taxa de 1.3 Mbits/s e os transmitindo para essa mesma estação.

Uma observação importante é que a taxa média de geração dos vídeos e áudios, durante as coletas de dados foi de 1.3 Mbits/s. No padrão de compressão de vídeo MPEG II a taxa de compressão de dados é de 4 Mbits/s, taxa bem maior do que a utilizada durante as coletas de dados.

### **7.3 - DESCRIÇÃO E ANÁLISE DOS RESULTADOS OBTIDOS NAS SIMULAÇÕES**

Nas simulações foram representadas situações onde existiam 2, 10 ou 20 pontos de teleconferência gerando vídeos a 4 Mbits/s, que é a taxa de compressão segundo o padrão MPEG II. Não se levou em consideração o tempo de captura dos vídeos por uma câmara de vídeo ou acesso a esses vídeos em um disco rígido ou CD-ROM, já que as compactações dos vídeos e áudios desse projeto foram realizadas através de um *software*, inviabilizando a compactação em tempo real, que somente é viável através de *hardware*.

Na tabela 7.18 mostra-se os parâmetros atribuídos para a realização das simulações. Esses dados foram atribuídos baseados nos resultados obtidos no ambiente de gerenciamento criado no software SunNet Manager e no sistema operacional Windows NT 3.5. O ambiente de gerenciamento pode ser visualizado na figura 6.4.

Parâmetro	Descrição	Valor atribuído
Taxa média de geração de vídeo	Taxa média de geração de um <i>frame</i> de vídeo conforme padrão MPEG II (4 Mbits/s).	0,94 milisegundos
Capacidade média do buffer das estações envolvidas na simulação	Capacidade média do buffer das estações para a recepção das transmissões. Dado coletado no Windows NT.	2105 <i>frames</i>
Token_Time	Tempo médio de transmissão do <i>token</i> de uma estação para outra. Dado coletado no software SunNet Manager.	0,04975625 milisegundos (fddimibSMTTransitionTime Stamp)
SbaPayLoad	Define a largura de banda síncrona para alocação manual e estática. Dado coletado no Windows NT.	1094 bytes. Valor que corresponde a 70% da largura de banda alocada para transmissões síncronas
TVX	Máximo tempo permitido para o nível MAC receber <i>frames</i> válidos ou <i>tokens</i> . Dado coletado no software SunNet Manager.	1,34220 milisegundos (fddimibMACTvxValue)
T-Notify	Frequência com que frames com informações dos vizinhos são enviados no anel. Dado coletado no software SunNet Manager.	30000 milisegundos (fddimibMACTNotify)
T-TTRT	Valor operacional do TTRT. Dado coletado no software SunNet Manager.	0,6711 milisegundos
T_Max	Tempo máximo para o <i>token</i> se mover por todo o anel quando não está submetido a nenhuma carga. Dado coletado no software SunNet Manager.	0,167772160 milisegundos (fddimibMACTMax)
T_TransAssinc	Máximo tempo que pode ser negociado com o protocolo SMT para transmissões síncronas. Dado coletado no Windows NT.	0,025 milisegundos
T_Assinc	Tempo entre gerações de <i>frames</i> nas transmissões síncronas. Parâmetro da própria arquitetura.	0,000125 milisegundos

F_Max	Tempo máximo de transmissão de um <i>frame</i> de tamanho máximo	0,0040 milisegundos
-------	--	---------------------

Tabela 7.18 - Parâmetros atribuídos para a simulação

Conforme descrito no capítulo 4, a soma de toda a banda síncrona não pode ultrapassar o valor de  $T\_TTRT - (T\_Max + F\_Max + Token\_Time)$ , assim com os valores atribuídos segundo tabela 7.18, esse limite foi respeitado. Segundo [Dykeman1987], esse valor máximo para a banda síncrona garante o tempo máximo de acesso igual a duas vezes  $T\_TTRT$ .

O valor do  $T\_TTRT$  é sugerido por todas as estações que estão no anel no momento de inicialização da rede e o protocolo SMT escolhe o menor deles. Esse tempo é baseado na metade do retardo máximo que as aplicações das estações podem suportar. Como na simulação todas as estações estão transmitindo vídeos compactados pelo padrão MPEG II, esse valor é único e constante igual à 0,6711, que é a metade do retardo máximo permitido, TVX, que é 1,34220.

O protocolo garante, também, que a utilização do anel seja igual a  $(T\_TTRT - T\_MAX) / T\_TTRT$ , que com esse modelo de simulação conseguiu-se 75%.

Para a validação do modelo de simulação, isto é, comprovação de que este modelo representa o ambiente criado na realidade, foram realizadas duas simulações, gerando tráfego de 100 Mb/s e comparou-se os resultados obtidos com os valores coletados nas transmissões reais, já citadas anteriormente neste mesmo capítulo. Nas figuras 7.1 e 7.2 pode-se visualizar os resultados dessas simulações.

Conforme figura 7.1, durante 1000 milisegundos transmitiu-se 269.568 *frames* a partir de 10 estações, valor que é proporcional ao número médio de *frames* transmitidos durante as coletas de dados descritas anteriormente, onde somente uma estação estava transmitindo e o tempo de transmissão era de 900.000 milisegundos.

Conforme figura 7.2, durante 1000 milisegundos transmitiu-se 53.921 *frames* a partir de duas estações, valor que é proporcional ao número médio de *frames* transmitidos durante as coletas de dados descritas anteriormente, onde somente uma estação estava transmitindo e o tempo de transmissão era de 900.000 milisegundos.

Conclui-se que o modelo representa a estrutura real do *Backbone* FDDI onde foram realizadas as coletas de dados.

Nas figuras 7.3, 7.4 e 7.5 mostram-se os resultados obtidos com a simulação de duas estações, 10 estações e 20 estações respectivamente.

SIMAN V - License #9410807

Systems Modeling Corporation

Summary for Replication 1 of 1

Project: Analise desempenho FDDI      Run execution date : 3/28/1995

Analyst: Claudia Barenco              Model revision date: 3/26/1995

Replication ended at time : 1000.0

Statistics were cleared at time: 2.11987e-311

Statistics accumulated for time: 1000.0

#### TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
------------	---------	-----------	---------	---------	--------------

TACESSO	.03709	3.2099	.00000	.51556	269568
---------	--------	--------	--------	--------	--------

Tsaida	.25450	.57821	1.8887E-06	.51766	269568
--------	--------	--------	------------	--------	--------

#### DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
------------	---------	-----------	---------	---------	-------------

Tamanho fila	5.2004	.72267	.00000	22.000	1.0000
--------------	--------	--------	--------	--------	--------

barramento Available	1.0000	.00000	1.0000	1.0000	1.0000
----------------------	--------	--------	--------	--------	--------

#### COUNTERS

Identifier	Count	Limit
------------	-------	-------

saida_C	269568	Infinite
---------	--------	----------

TATRASADO	0	Infinite
-----------	---	----------

bufcheio	0	Infinite
----------	---	----------

RECEBET	19624	Infinite
---------	-------	----------

FILAVAZIA	39247	Infinite
-----------	-------	----------

Execution time: 108.60 minutes.

Simulation run complete.

Figura 7.1 - Simulação realizada com taxa de geração de dados a 100 Mbits/s (10 estações)

SIMAN V - License #9410807  
 Systems Modeling Corporation  
 Summary for Replication 10 of 10

Project: Analise desempenho FDDI      Run execution date : 6/23/1995  
 Analyst: Claudia Barenco              Model revision date: 3/28/1995  
 Replication ended at time : 1000.0  
 Statistics were cleared at time: 1.37293e-311  
 Statistics accumulated for time: 1000.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
TACESSO	.03709	1.1847	.00000	.44831	53921
Tsaida	.04990	.57593	5.8911E-07	.10024	53921

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Tamanho fila	1.0185	1.1559	.00000	11.000	.00000
barramento Available	1.0000	.00000	1.0000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
saida_C	53921	Infinite
TATRASADO	0	Infinite
bufcheio	0	Infinite
RECEBET	20003	Infinite
FILAVAZIA	35593	Infinite

Execution time: 110.10 minutes.  
 Simulation run complete.

Figura 7.2 - Simulação realizada com taxa de geração de dados a 100 Mbits/s (2 estações)

SIMAN V - License #9410807  
Systems Modeling Corporation  
Summary for Replication 10 of 10

Project: Analise desempenho FDDI                      Run execution date : 6/23/1995  
Analyst: Claudia Barenco                                Model revision date: 3/31/1995  
Replication ended at time : 1000.0  
Statistics were cleared at time: 1.37293e-311  
Statistics accumulated for time: 1000.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
TACESSO	.96671	.96849	.00000	6.7674	2068
Tsaida	.05042	.82436	1.9623E-04	.98988	2068

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Tamanho fila	.03578	5.2673	.00000	2.0000	.00000
barramento Available	1.0000	.00000	1.0000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
saida_C	2068	Infinite
TATRASADO	0	Infinite
bufcheio	0	Infinite
RECEBET	20076	Infinite
FILAVAZIA	14454	Infinite

Execution time: 10.25 minutes.  
Simulation run complete.

Figura 7.3 - Simulação realizada com taxa de geração de dados a 4 Mbits/s (2 estações)

SIMAN V - License #9410807  
 Systems Modeling Corporation  
 Summary for Replication 10 of 10

Project: Analise desempenho FDDI                      Run execution date : 6/23/1995  
 Analyst: Claudia Barenco                                Model revision date: 3/31/1995  
 Replication ended at time : 1000.0  
 Statistics were cleared at time: 2.11987e-311  
 Statistics accumulated for time: 1000.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
ACESSO	.96703	.98516	.00000	6.9700	10336
Tsaida	.25330	.58310	2.8930E-05	1.8874	10336

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Tamanho fila	.17517	2.4317	.00000	3.0000	.00000
barramento Available	1.0000	.00000	1.0000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
saida_C	10336	Infinite
TATRASADO	1	Infinite
bufcheio	0	Infinite
RECEBET	20061	Infinite
FILAVAZIA	14401	Infinite

Execution time: 20.55 minutes.  
 Simulation run complete.

Figura 7.4 - Simulação realizada com taxa de geração de dados a 4 Mbits/s (10 estações)

SIMAN V - License #9410807  
Systems Modeling Corporation  
Summary for Replication 1 of 1

Project: Analise desempenho FDDI                      Run execution date : 3/31/1995  
Analyst: Claudia Barenco                                      Model revision date : 3/26/1995  
Replication ended at time : 1000.0  
Statistics were cleared at time: 2.62915e-311  
Statistics accumulated for time: 1000.0

TALLY VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Observations
TACESSO	1.0791	1.1441	1.2500E-04	8.5461	17589
Tsaida	.60642	.60017	2.3894E-07	4.0304	17589

DISCRETE-CHANGE VARIABLES

Identifier	Average	Variation	Minimum	Maximum	Final Value
Tamanho fila	.42132	1.6179	.00000	4.0000	1.0000
barramento Available	1.0000	.00000	1.0000	1.0000	1.0000

COUNTERS

Identifier	Count	Limit
saida_C	17589	Infinite
TATRASADO	11	Infinite
bufcheio	0	Infinite
RECEBET	16359	Infinite
FILAVAZIA	5827	Infinite

Execution time: 305.05 minutes.  
Simulation run complete.

Figura 7.5 - Simulação realizada com taxa de geração de dados a 4 Mbits/s (20 estações)

Os tempos de acesso ao meio de transmissão médios foram de 0,96671, 0,96703 e 1,0791 milissegundos para 2, 10 e 20 estações respectivamente, tempos considerados ótimos segundo [Legay1991], que estipula o tempo máximo de acesso de 1 milissegundo.

A partir do momento em que o *frame* foi gerado até o início de sua transmissão demorou-se em média 0,05042, 0,25330 e 0,60642 milissegundos para 2, 10 e 20 estações respectivamente. O [Legay1991] diz que o atraso máximo para um vídeo compactado deve ser de 250 milissegundos. Considerando-se que o número médio de *frames* por vídeo transmitidos durante as coletas de dados, mostradas nesse mesmo capítulo, foi de 730 *frames*, obteremos em média 36,8066, 183,6424 e 442,6866 milissegundos, mostrando que para 20 estações o atraso máximo permitido já é extrapolado. Uma observação a ser feita é que 250 milissegundos é o retardo máximo para chegar até o usuário destino, assim a esses tempos obtidos pela simulação tem-se que adicionar o tempo de captura ou acesso ao vídeo, o tempo de compactação, o tempo de transmissão e o tempo de descompactação dos vídeos.

Em média ficaram no *buffers* das estações esperando para serem transmitidos : 0,03578, 0,17517 e 0,42132 *frames* e no máximo 2, 3 e 4 *frames* para 2, 10 e 20 estações respectivamente. Assim, conclui-se que nas três situações houve uma boa vazão dos vídeos gerados e nenhum *frame* foi descartado por falta de espaço no *buffer* local.

O *token* chegou atrasado 0%, 0,0049% e 0,06% das vezes para 2, 10 e 20 estações respectivamente. Isso significa que o congestionamento na rede foi baixo.

Nas três simulações o percentual de vezes em que o *token* chegou na estação mas não havia *frames* de vídeo para serem transmitidos foram : 72%, 71% e 25% para 2, 10 e 20 estações respectivamente. Esses percentuais indicam que a vazão dos vídeos é alta mas com o aumento do número de estações essa vazão tende a diminuir. Resultado que pode ser também visto pelo aumento da média do número de *frames* nos *buffers* das estações, esperando para serem transmitidos.

Para 2 e 10 estações o número médio de *frames* transmitidos por cada estação em 1000 milissegundos foram quase os mesmos : 1034 e 1033 respectivamente. Para 20 estações esse número diminuiu para 879,45 *frames*. Novamente confirma-se que a vazão dos vídeos gerados diminuiu bastante quando 20 estações estavam transmitindo.

## Capítulo 8

### CONCLUSÃO

#### 8.1 - CONCLUSÃO

A inexistência de recursos de hardware obrigou o presente estudo a ficar restrito a utilização de compressão de vídeo e áudio via software.

O estudo de multimídia apresentou a carência de softwares que efetuem a multiplexação de sinais de vídeo com sinais de áudio, dificultando o estudo proposto, no sentido de avaliar o desempenho da rede com transmissão sincronizada de áudio e vídeo.

Pelo exposto, a análise de desempenho da rede não pode ser efetuada em tempo real.

Com a utilização de um modelo de simulação, obteve-se uma visão aproximada do comportamento da rede FDDI, sob diferentes cargas de tráfego e número de usuários conectados. Foram realizadas duas simulações, gerando tráfego de 4 Mbits/s e os resultados obtidos, após a comparação com os valores coletados nas transmissões reais, permitem afirmar que :

Quando duas estações estavam transmitindo simultaneamente sinais de vídeo e áudio no Backbone FDDI da UFSC, o tempo médio de acesso ao meio de transmissão foi de 0,96671 milissegundos, tempo considerado ótimo segundo [Legay1991]. O tempo médio entre a geração do vídeo até o início de sua transmissão foi de 36,8066 milissegundos, um estudo feito por [Legay1991] diz que o máximo atraso permitido para transmissão de vídeos compactados é 250 milissegundos, significando que nesse estudo esse tempo está bem abaixo do limite máximo permitido. Em nenhuma das simulações realizadas entre essas duas estações o *token* chegou atrasado. A vazão na rede foi alta, resultado comprovado pelo percentual alto do número de vezes que o *token* chegou na estação mas não havia *frames* de vídeo a serem transmitidos.

Para dez estações, o comportamento foi considerado muito bom, já que o tempo médio de acesso ao meio foi de 0,96703 milissegundos, o tempo médio entre a geração do vídeo e o início de sua transmissão foi de 183,6424 milissegundos, o *token* chegou atrasado 0.0049% das vezes,

percentual considerado baixo e 71% das vezes o *token* chegou na estação mas não havia *frames* a serem transmitidos.

Para vinte estações, o tempo médio para acesso ao meio foi de 1,0791 milissegundos, tempo já considerado ruim segundo [Legay1991]. Entre a geração do vídeo e início de sua transmissão transcorreu-se em média 442,6866 milissegundos, ou seja, o atraso máximo permitido foi extrapolado e a vazão na rede diminuiu bastante, comparando-se com a vazão de duas e dez estações na rede.

Considera-se a seguinte situação : vinte departamentos de uma universidade participarão de uma teleconferência sobre 'Ensino à Distância' e um grande especialista internacional será o palestrante. O *Backbone* FDDI que está sendo utilizado para a teleconferência possui as mesmas características descritas no capítulo 6. Todos os departamentos estão utilizando microcomputadores com a configuração da estação multimídia 'Picasso', descrita no capítulo 6 e com a arquitetura de redes mostrada na figura 4.1. Os vídeos estão sendo capturados por uma placa que possui compressão de dados em tempo real de 4 Mbits/s. Essa teleconferência é interativa, ou seja, os participantes da teleconferência podem interferir na conferência, fazendo perguntas e discutindo idéias, sendo gerado, com isso, vídeos de todos os pontos da conferência.

Com base em tal situação fictícia poderia-se observar aspectos como : tempo de acesso ao meio físico, fluxo de dados, tempo máximo de retardo permitido para que a teleconferência não incomode os receptores da conferência, taxa de erros de *frames* e congestionamento. Baseando-se nas análises descritas no capítulo 7, vinte estações participando de um sistema de teleconferência com as mesmas características desse projeto, não teria o desempenho necessário para que a teleconferência ocorresse da maneira mais natural possível. O aspecto retardo máximo permitido foi extrapolado quando simulou-se 20 estações transmitindo vídeos a 4 Mbits/s simultaneamente, como pode-se verificar no capítulo 7, na seção 7.3 e o fluxo de dados diminuiu muito. As qualidades dos vídeos e dos áudios foram mantidas quase que integralmente, já que como foi descrito no capítulo 7, a taxa de erros de bits e de pacotes foram quase nulas.

Em um mesmo cenário onde tivéssemos 10 estações, a teleconferência ocorreria de maneira satisfatória, atendendo os aspectos citados no parágrafo anterior.

É importante lembrar que 20 estações que estejam em um sistema de teleconferência, podem ser aproximadamente 600 pessoas que estejam participando na conferência. Também deve ficar

claro que a situação fica crítica somente no caso em que as estações transmitem simultaneamente para o mesmo destino. Em uma situação real de teleconferência, a transmissão se dá no sentido de um para muitos (e não de muitos para um).

## 8.2 - RECOMENDAÇÕES PARA TRABALHOS FUTUROS

O projeto desenvolvido faz parte de uma área de estudo onde as pesquisas começaram recentemente. Por esse motivo, muitos trabalhos ainda devem ser desenvolvidos a fim de dar suporte à aplicações mais sofisticadas.

A possibilidade de se criar uma rede mundial de computadores, a chamada *Super Highway* para a transmissão de todos os tipos de dados : texto, voz, áudio e vídeo, necessita antes do desenvolvimento de novas tecnologias em várias áreas : compressão de dados, redes de computadores, softwares mais interativos e que utilizem todos os recursos oferecidos pelo hardware, melhores hardwares e de custo mais baixo, meios de transmissão como a fibra ótica instalados em todos os pontos do mundo e principalmente a disseminação maior da utilização de computadores em todos os lares e empresas.

O objetivo deste projeto foi dar um início ao estudo da área de redes de alta velocidade, com a intenção de tornar viável a transmissão de dados multimídia, principalmente vídeo, já que como pôde-se verificar no capítulo 2, é a mídia que requer maiores controles quando esta é transmitida em uma rede de computadores.

Outro aspecto importante a ser citado, é que a tecnologia multimídia e a tecnologia de redes de alta velocidade são tecnologias ainda de custo alto, fazendo com que em fases do projeto utilizássemos de recursos que não eram os ideais. Como exemplo, pode-se citar a utilização da compressão de vídeo e áudio por meio de software e não por hardware, o que seria ideal, e que permitiria o estudo da transmissão em tempo real.

A primeira sugestão a ser feita é a aquisição de hardware para compressão e descompressão de vídeo em tempo real. Foi realizada uma pesquisa no mercado e encontrou-se o produto Xing Sound para captura e compressão de áudio em tempo real no padrão MPEG [Xing1994]. Outro produto encontrado foi a placa ReelMagic para compressão e descompressão de vídeo no padrão MPEG [Reel1994].

O software InVision e VisionGraphics da Invision Systems Corporation [I13] é um aplicativo para o ambiente Windows para realização de videoconferência, compatível com a arquitetura FDDI. Os problemas encontrados nesse aplicativo são : (1) O driver para compressão de vídeo no padrão MPEG, ainda está em desenvolvimento; (2) Suporta somente o protocolo TCP/IP e o protocolo de transporte IPX e (3) Não possui versão para Windows NT; espera-se que, até o final do ano, essa versão esteja disponível.

Estudos devem ser realizados sobre os novos protocolos de transporte e rede para comunicação multimídia em tempo real. Alguns desses protocolos que estão sendo desenvolvidos, são os já citados anteriormente : RTC, RTCP, ST-II e XTP.

Um projeto muito interessante desenvolvido na Universidade de Mannheim, Alemanha é o XMovie [Adie1993], onde foi criado um ambiente de teste para transmissão e apresentação de vídeos digitais no ambiente Unix e que estão utilizando protocolos específicos para transmissões multimídia.

A tecnologia FDDI II é mais apropriada para transmissão de multimídia em tempo real, já que possui a característica de ser isócrona, ou seja, o retardo de transmissão é determinístico, facilitando a sincronização do vídeo com o áudio. Assim, seria necessário portar da tecnologia FDDI, utilizada nesse projeto, para a tecnologia FDDI II, quando se pesquisar as transmissões multimídia em tempo real.

O sistema operacional Windows NT é recente e é 32 bits; assim, novos aplicativos devem surgir para aproveitarem todas as facilidades oferecidas pelo Windows NT, tais como : multi-thread, portátil para várias plataformas, endereçamento de até 4 Gbytes de memória, sendo 2 Gbytes utilizado pelo usuário e 2 Gbytes utilizado pelo sistema operacional e outras. Aplicativos originalmente desenvolvidos para o Windows 3.1 são executados mais lentamente sob o Windows NT. A pesquisa realizada no mercado mostrou que existem poucos aplicativos já portados para o Windows NT.

No nível físico da arquitetura FDDI, pode-se também utilizar o SPM (SONET Physical Layer Mapping) que possibilita velocidades de 155.52 Mbps (STS-3), 622.08 (STS-48) e 2.4 Gbps (STS-48). O padrão SPM já está sendo incorporado à arquitetura FDDI III, cuja definição completa esta prevista para o final de 1995.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [3COM1994] - **3Com**; "*FDDILink Network Adapters - Installation Guide*". Manual Part No. 09-0313-003. 1994
- [3COMA1993] - **3Com**; "*LinkBuilder FDDI Workgroup Hub - User Guide*". Manual Part No. 09-0447-000. 1993.
- [3COMB1993] - **3Com**; "*Transcend LinkBuilder FDDI SmartAgent Software*". Manual Part No. 09-0537-000. 1994.
- [Adams1985] - **C. Adams, S. Ades**; "*Voice Experiments in the UNIVERSE Project*". Proceedings of the International Conference on Communications, 29.4.1-29.4.9, 1985.
- [Adie1993] - **C. Adie**; "*A Survey of Distributed Multimedia Research, Standards and Products*", RARE Project, Amsterdam, January 1993.
- [Adie1994] - **C. Adie**; "*Network Access to Multimedia Information*", RFC1614, Edinburgh University, May 1994.
- [Alder1992] - **M. Alder**; "*MPEG-Video*". Arquivo Internet mpegfa31.txt, October, 1992.
- [Allend1993] - **D. Allend**; "*The Real Multimedia*". Magazine Byte, Pp. 10, July, 1993.
- [Almeida1993] - **R. A. F. Almeida, J. E. de A. Moreira, J. E. F. Neto, A. F. Almeida**; "*Avanços da Tecnologia Instrucional em Telecomunicações - Uma Experiência Brasileira*". VI Reunião Interregional sobre Recursos Humanos e Treinamento da UIT; Pp. 1-20, junho de 1993.
- [Barenco1994] - **C. J. Barenco, L. L. Vieira, E. S. Specialski**; "*Multimídia em Rede FDDI : Uma solução para o ensino à distância e teleconferências*". 14o Encontro Nacional de Engenharia de Produção, outubro 1994.
- [Braden1987] - **R. Braden and J. Postel**; "*Requirements for Internet Gateways*", RFC 1009, USC/Information Sciences Institute, 1987.
- [Case1993] - **J. Case**; "*FDDI MIB*", RFC1512, The University of Tennessee and SNMP Research, September 1993.
- [Chaves1991] - **E. O. C. Chaves**, "*Multimídia conceituação, aplicações e tecnologia*". Editora People Computação, setembro 1991.
- [Cheriton1988]- **D. Cheriton**, "*VMTP: Versatile Message Transaction Protocol Specification*", RFC 1045, Stanford University, February 1988.
- [Chimiak1993] - **W. J. Chimiak**; "*A Comment on Packet Video Remote Conferencing and the Transport/Network Layers*". RFC1453 - Request for Comments, Pp. 1-10, abril de 1993.
- [Cho1990] - **Cho & Lee**; "*Fast Algorithm and Implementation of 2-D DCT*". Tr. CAS Vol. 38 Pp. 297
- [Clark1987]- **Clark, D. Lambert, M. and L. Zhang**, "*NETBLT : A Bulk Data Transfer Protocol*", RFC 998, MIT, March 1987.

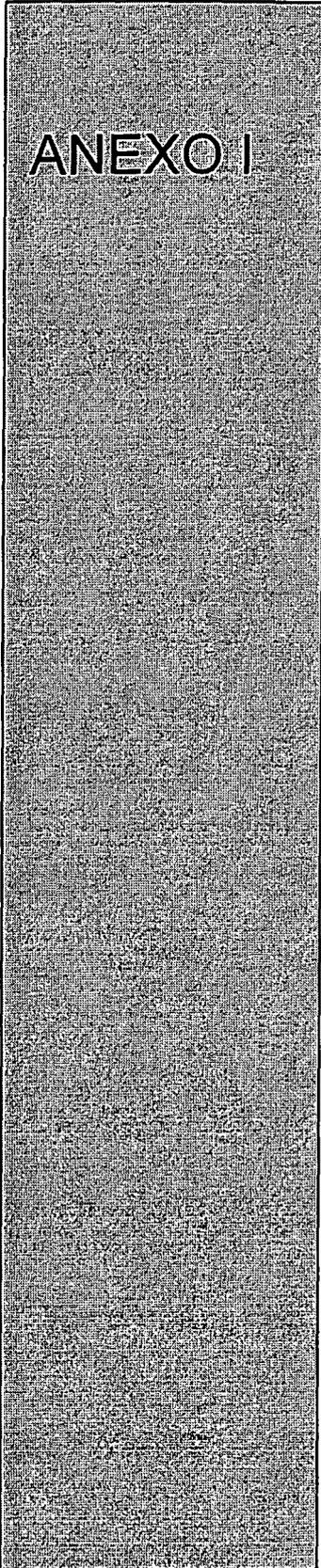
- [ClarksonA1993] - **M. Clarkson**; "*ALL-Terrain Networking*". Magazine Byte, Pp. 111-116, August, 1993.
- [ClarksonB1993] - **M. A. Clarkson**; "*Hitting Warp Speed for LANs*". Magazine Byte; Pp. 123-128, March, 1993.
- [Cohen1981] - **D. Cohen**; "*On holy wars and a plea for peace*", Computer, IEEE, October 1981.
- [Cole1981]- **Cole, R.**, "*PVP - A Packet Video Protocol*", W-Note 28, Information Sciences Institute, University of California, Los Angeles, CA, August 1981.
- [Compuadd1993] - **Compuadd**; "*Multimídia em Rede de Telefonia*". Pifcon System; Pp. 01-05, 1993.
- [Corner1993] - **J. B. Corner**; "*What's the Deal with the DCT ?*". IEEE Computer Graphics & Applications, Pp. 78-83, july 1993.
- [Cronk1993] - **R. D. Cronk**; "*Unlocking Data's Content*". Magazine Byte, Pp. 111-120; september 1993.
- [Deering1989] - **S. Deering**; "*Host Extensions for IP Multicasting*", RFC 1112, Stanford University, August 1989.
- [Derfler1992] - **F. J. Derfler Jr.**; "*Guia para a Interligação de Redes Locais*". PC Magazine - Editora Campus, 1992.
- [Dykeman1987] - **D. Dykeman, W. Bux**; "*An Investigation of the FDDI Media Access Control Protocol*". Research Report, RZ 1591, IBM Zurich, maio de 1987.
- [Embratel1993] - **Embratel**; "*O futuro passa por aqui*". Guia de Serviços; Pp. 08-10, 1993.
- [Ferrari1990] - **D. Ferrari**; "*Client Requirements for Real-Time Communications Services*". RFC1193 - Request for Comments, Pp. 01-24, november, 1990.
- [Ficap1992] - **Ficap - Fios e Cabos Plásticos do Brasil S.A.**; "*Informativo Técnico de Fibras óticas*", 1992.
- [Fogg1993] - **C. Fogg**, "*MPEG - 2*". Arquivo Internet mpegfa31.txt, october, 1993.
- [Fogg1994] - **C. Fogg**; "*MPEG Frequently Asked Questions List*". MPEG Software Simulation Group, Draft 3.4, june 1994.
- [Friese1994] - **J. Friese**; "*Safeguarding high-speed transmission*". Telcom Report, Siemens Telecommunications; Pp. 43-44, january/february, 1994.
- [Furiati1993] - **G. Furiati**; "*TELERJ Faz Teste na Rede Metropolitana*". RNT 166, Pp. 23, Junho de 1993.
- [Gaggioni1987] - **H. P. Gaggioni**, "*The Evolution of Video Technologies*". IEEE Communications Magazine, Pp. 20-36, november 1987.
- [Gall1991] - **D. Le. Gall**; "*MPEG : A video Compression Standard for Multimedia Applications*". Communications of the ACM, april 1991, Commitee Draft.

- [Giozza1986] - **W. F. Giozza, J. F. M. de Araújo, J. A. B. Moura e J. P. Sauvé;** "*Redes Locais de Computadores - Tecnologia e Aplicações*". Embratel - Editora McGraw-Hill, Pp. 91-94, 1986.
- [Goldacker1993] - **G. Goldacker, T. Luckenbach;** "*ATM and ATM Adaptation Layer - Overview and Codings*". Berlin; Pp. 01-15, April, 1993.
- [Gopal1984] - **P. M. Gopal, J. W. Wong, J. C. Majithia;** "*Analysis of Playout Strategies for Voice Transmission Using Switching Techniques*". Performance Evaluation, n. 4; fevereiro de 1984.
- [Gruber1985] - **J. G. Gruber;** "*Subjective Effects of Variable Delay and Speech Loss in Dinamically Managed Voice Systems*". IEEE Transactions on Communications, Vol. com-33, agosto de 1985.
- [Hammond1986] - **J. L. Hammond, J. P. O'Reilly;** "*Peformance Analyses of Local Computer Networks*". Addison-Wesley, 1986.
- [Hecht1993] - **J. Hecht;** "*Entendendo Fibras Óticas*". Editora Berkeley , 1993.
- [IBM1987] - **IBM;** "*Dictionary of Computing*", March 1987.
- [IEEE1985] - **IEEE,** "*IEEE Standards for Local Area Networks : Logical Link Control*", IEEE, New York, 1985.
- [IEEE1989] - **IEEE,** "*IEEE Standard P802.1A - Overview and Architecture*", 1989.
- [Irvén1988] - **J. H. Irvén, M. E. Nilson, T. H. Judd, J. F. Petterson e Y. Shibata;** "*Multimedia Information Services : A Laboratory Study*". IEEE Communications Magazine, Vol. 26, Pp. 27-44, june 1988.
- [ISO1989] - **ISO;** "*Fiber Distributed Data Interface (FDDI) - Media Access Control*", ISO 9314-2, 1989.
- [JPEG1991] - **JPEG;** "*JPEG in Communications of the ACM*", april 1991, Committe Draft.
- [Kano1991] - **S. Kano, K. Kitami, M. Kawarasaki;** "*ISDN Standardization*". Proceedings of IEEE, Vol. 79, No. 2, fevereiro 1991.
- [Karmouch1993] - **A. Karmouch;** "*Multimedia Communications*". Lab. University of Ottawa, Canada, SBRC maio de 1993.
- [Katz1990] - **D. Katz,** "*A Proposed Standard for the Transmission of IP Datagrams over FDDI Networks*". RFC 1188, october 1990.
- [Legay1991] - **P. Legay and G. Roulet;** "*Lion and MAX, the experiences of two Esprit projects on High-Speed Mans*". NATO ASI Series, vol. F72, Pp. 119-145.
- [Lobato1992] - **R. M. F. Lobato;** "*Comunicação Homem(Máquina) x Máquina = Computação Gráfica + Sistemas Multimídia*". Revista Tecnologia, Pp. 48-59, Dezembro de 1992.
- [Makron1994] - **Makron Books;** "*Arquiteturas de Redes de Computadores OSI e TCP/IP*". São Paulo, 1994.

- [Markopoulos1990] **J. G. Markopoulos, C. I. Costopoulou, M. E. Theologou e E. N. Protonotarios;** "*Object-Oriented Modeling for Quality of Service and Network Performance*". Proceedings of the Fifth RACE TMN Conference; Pp. 01-13, 1990.
- [Martini1988] - **P. Martini, O. Spaniol and T. Welzel;** "*File Transfer in High-Speed Token Ring Networks : Performance Evaluation by Approximate Analysis and Simulation*". IEEE Journal on Selected Areas in Communications, Vol. 6, No. 6, July 1988.
- [McCloghrie1991] - **K. McCloghrie;** "*Management Information Base for Network Management of TCP/IP-based internets : MIB-II*". RFC 1213, march, 1991.
- [McQuillin1991] - **L. McQuillin,** "*Multimedia and Video*". Jornal MacUser, Vol.7, No. 2, Pp. 112, february 1991.
- [Mello1993] - **J. P. Mello Jr.;** "*Future Communications*". Magazine Byte; Pp. 94-106, August, 1993.
- [Microsoft1993] - **Microsoft;** "*TCP/IP*". Manual Microsoft Windows NT 3.1. 1993.
- [Montoro1990] - **F. A. Montoro;** "*Transmissão de dados e modem*". Pp. 32, 1990.
- [Murdoch1972] - **D. C. Murdoch,** "*Álgebra Linear*". Livros técnicos e científicos, Rio de Janeiro, 1972.
- [Notare1994] - **M. S. M. A. Notare, C. J. Barenco, L. Lustosa Vieira;** "*Tratamento e Manipulação de imagens no ensino a distância*". INFOIMAGEM '94, outubro 1994.
- [Pegden1990] - **C. D. Pegden and R. E. Shannon ;** "*Introduction to Simulation Using SIMAN*". McGraw-Hill, Inc. 1990
- [Popp1994] - **H. Popp;** "*MPEG-Audio*". Arquivo Internet mpegfa31.txt, march, 1994.
- [Postel1983] - **J. Postel;** "*The TCP Maximum Segment Size Option and Related Topics*", RFC 879, USC/Information Sciences Institute, November 1983.
- [Ranai1994] - **K. Ranai, K. S. Lim and R. H. Deng,** "*Simulating a Multimedia FDDI Backbone Network*". The Computer Journal, Vol. 37, No. 3, Pp. 187-198, 1994.
- [Reel1994] - **Reel Magic;** *Unioni Comércio e Indústria Ltda.* Fax : 011-744-3134; São Paulo, S.P.
- [Reynolds1990] - **J. K. Reynolds and J. Postel,** "*Assigned Numbers*". RFC 1060, USC/Information Sciences Institute, march 1990.
- [Roschelsen1993] - **F. Roschelsen;** "*Giving the Telephone Vision*". Telecom Report International, Siemens Private Communication System, Inc.; Pp. 7-10, 1993.

- [Santos1993] - **N. Santos, L. Lustosa, A. P. Fialho;** "*Ensino à distância e multimídia, uma realidade nacional*". Projeto de pesquisa do departamento de engenharia de produção da Universidade Federal de Santa Catarina, 1993.
- [Schildt1994] - **H. Schildt;** "*Windows NT - Manual de programação*". Editora Makron Books, 1994.
- [Schulzrinne1992]- **Schulzrinne, H.,"***A Transport Protocol for Audio and Video Conferences and other Multiparticipant Real-Time Applications*", Internet Engineering Task Force, Internet-Draft, October 1992.
- [Sevcik1987] - **Sevcik, K. C. and Johnson, M. J.;** "*Cycle Time Properties of the FDDI Token Ring Protocol*". IEEE Transm. Softw. Engr. SE-13, 00-00. 1987.
- [Siemens1993] - **Siemens Private Communications System Inc. ;** "*Metropolitan Area Network, EWSM High-Speed Communications Networks*", 1993.
- [Soares1986] - **L. F. G. Soares,** "*Redes Locais*", Editora Campus, Rio de Janeiro, 1986.
- [Soares1990] - **L. F. G. Soares, S. L. Martins;** "*Um algoritmo para Compensação da Variação Estatística do Retardo em Redes Comutadas por Pacotes*". Anais do 8o Simpósio Brasileiro de Redes de Computadores, pp. 435-445, 1990.
- [Soares1991] - **L. F. G. Soares, S. L. Martins, T. L. P. Bastos;** "*Um algoritmo para Compensação da Variação Estatística do Retardo em Redes Comutadas por Pacotes*". Anais do 8o. Simpósio Brasileiro de Redes de Computadores, 1991.
- [Soares1992] - **L. F. G. Soares, L. Tucheman, M. A. Casanova e P. R. R. L. Nunes;** "*Fundamentos de Sistemas Multimídia*". VII Escola de Computação, Gramado - RS, agosto de 1992.
- [Spaniol1988] - **Spaniol, O., Danthine, A.;** "*High Speed Local Area Networks*", Proceedings of First IFIP WG 6.4 Workshop on HSLAN, Aachen 1987.
- [Stallings1993] - **W. Stallings;** "*FDDI Speaks*". Magazine Byte, Pp. 197-200, April, 1993.
- [SunNet1993] - **SunNet Manager 2.2;** "*User's Guide Manual*". Sun Microsystems, Inc. Business. 1993.
- [Syskonnnect1993] - **Syskonnnect;** "*Innovation in FDDI connectivity*". Installation Manual, 1993.
- [Tanenbaum1988] - **A. S. Tanenbaum;** "*Computer Networks*". Second Edition, Pp. 56, 1988.
- [Telecom1991] - **Telecom;** "*High Speed Communications- The Pulse of the Economy*". Telecom Report International, Pp. 01-05, july/august, 1991.
- [Topolcic1990]- **C. Topolcic,** "*Experimental Internet Stream Protocol : Version 2 (ST II)*", RFC 1190, CIP Working Group, October 1990.

- [Verma1990]- **P. de K. Verma**; "*ISDN Systems*". Architecture, Technology, and Applications; Pp. 262-304, 1990.
- [Victoria1993] - **R. Victoria**; "*Guia de Multimídia*". Berkeley Brasil Editora, 1993.
- [VIDICOMA1993] - **Informe VIDICOM**; "*Aplicações de Videocomunicação Digital no Brasil*". Pp. 1, julho de 1993.
- [VIDICOMB1993] - **Informe VIDICOM**; "*Circuitos Digitais Comutados : Ampliam-se as Aplicações*". Pp. 02 , julho, 1993.
- [VIDICOMC1993] - **Informe VIDICOM**; "*Videocomunicação Digital - Tipos de Redes de Videoconferência*". Pp. 04, junho de 1993.
- [VIDICOMD1993] - **Informe VIDICOM**; "*Videocomunicação Digital - CODECs menores, mais baratos e melhores*". Pp. 02, setembro de 1993.
- [Vieira1994] - **L. L. Vieira, C. J. Barenco**; "*The Methodology for the design and implementation of multimedia systems training in the firm*". SEHE'94 - First International Conference on Software Engineering in Higher Education, november 1994.
- [Watson1983]- **R. W. Watson**, "*Delta-t Protocols Specification*", Lawrence Livermore Laboratory, April 15, 1983.
- [Xing1994] - **Xing Sound**. *Xing Technology Corporation* . Fax : 805-473-0147; Arroyo Grande, CA.
- [XTP1992]- **Xpress Transfer Protocol, version 3.6**; *XTP Forum*, 1900 State Street, Suit D, Santa Barbara, California 93101, USA, January 11, 1992.
- [Yager1993] - **Tom Yager**; "*Better than Being there*". Magazine Byte, Pp. 129-134, março de 1993.
- [Zakir1988] - **J. Zakir Júnior**; "*Redes Locais - O estudo de seus elementos*". Livros técnicos e científicos LTDA; Pp. 23-27, 1998.



ANEXO I

---

LISTAGEM DO PROJETO  
IMAGEM.PRJ NA  
LINGUAGEM C++

```
\*****/
```

FTP.C -- Servidor FTP de Vídeo e Áudio

Copyright() Cláudia Barenco, 1994

```
\*****/
```

```
#include <windows.h>
#include <stdlib.h>
#include <commdlg.h>
#include "demomdi.h"
#include "string.h"

#define min(a,b) (((a) < (b)) ? (a) : (b))
#define max(a,b) (((a) > (b)) ? (a) : (b))
#define SEMTITULO "(semtitulo)"
#define EDITID 1

long FAR PASCAL _export ProcJanBase (HWND, UINT, UINT, LONG);
BOOL FAR PASCAL _export ProcFecharEnum (HWND, LONG);
long FAR PASCAL _export ProcJanOla (HWND, UINT, UINT, LONG);
long FAR PASCAL _export ProcJanPaste (HWND, UINT, UINT, LONG);
long FAR PASCAL _export ProcJanRetang (HWND, UINT, UINT, LONG);
BOOL FAR PASCAL _export ProcDlgSobre (HWND, UINT, UINT, LONG);
BOOL FAR PASCAL _export ProcDlgAjuda (HWND, UINT, UINT, LONG);
BOOL FAR PASCAL _export ProcDlgEnvia (HWND, UINT, UINT, LONG);
BOOL FAR PASCAL _export ProcDlgRecebe (HWND, UINT, UINT, LONG);
BOOL FAR PASCAL _export ProcDlgDemo (HWND, UINT, UINT, LONG);

static HWND hjanEditar,hwnd;
void PoeTitulo (HWND, char *);
static long TamanhoArquivo ( int hArquivo);
static UINT msgProcuraTroca;
static OPENFILENAME ofn;
void InicializaArq (HWND);
BOOL DlgAbrirArquivo (HWND, LPSTR, LPSTR);
BOOL LerArquivo (HWND, LPSTR);
HWND hjanBase, hjanCliente;
char aux[80];
char szMsg[80];
HINSTANCE hinstMod;
int a;
WORD awShow[2] = { 2, SW_SHOW};
struct LOADPARMS {
WORD segEnv;
LPSTR lpszCmdLine;
```

```

LPWORD lpwShow;
LPWORD lpwReserved;
} parms;
char szClasseBase [] = "MdiBase";
char szClasseOla [] = "MdiOlaFilha";
char szClasseRetang [] = "MdiRetangFilha";
char szClassePaste [] = "MdiPaste1";
HANDLE hCop;
RECT retang;
HMENU hMenuInic, hMenuOla, hMenuRetang;
HMENU hMenuInicJanela, hMenuOlaJanela, hMenuRetangJanela;

```

```
// Estrutura para armazenar os dados de cada janela-filha
```

```

typedef struct
{
    short nCor;
    COLORREF corTexto;
}
DADOSOLA;

```

```
typedef DADOSOLA NEAR *NPDADOSOLA;
```

```
// Estrutura para armazenar os dados de cada janela-filha Retang
```

```

typedef struct
{
    short cxCliente;
    short cyCliente;
}
DADOSRETANG;

```

```
typedef DADOSRETANG NEAR *NPDADOSRETANG;
```

```
// Variaveis Globais
```

```

static FARPROC lpfnProcDlgSobre;
static FARPROC lpfnProcDlgAjuda;
static FARPROC lpfnProcDlgEnvia;
static FARPROC lpfnProcDlgRecebe;
static FARPROC lpfnProcDlgDemo;
static char szNomeAplic [] = "Imagem";

```

```
int PASCAL WinMain (HANDLE hCopia, HANDLE hCopiaAnterior,
```

```

LPSTR lpszLinhaCmd, int nCmdMostrar)
{
HANDLE hAcelera ;
MSG msg ;
WNDCLASS classejan ;

hCop = hCopia ;

if (!hCopiaAnterior)
    {
        // Registra a classe da janela de base

classejan.style = CS_HREDRAW | CS_VREDRAW ;
classejan.lpfnWndProc = ProcJanBase ;
classejan.cbClsExtra = 0 ;
classejan.cbWndExtra = 0 ;
classejan.hInstance = hCopia ;
classejan.hIcon = LoadIcon (NULL, IDI_APPLICATION) ;
classejan.hCursor = LoadCursor (NULL, IDC_ARROW) ;
classejan.hbrBackground = COLOR_APPWORKSPACE + 1 ;
classejan.lpszMenuName = NULL ;
classejan.lpszClassName = szClasseBase ;

RegisterClass (&classejan) ;

        // Registra a classe da janela-filha OLA

classejan.style = CS_HREDRAW | CS_VREDRAW ;
classejan.lpfnWndProc = ProcJanOla ;
classejan.cbClsExtra = 0 ;
classejan.cbWndExtra = sizeof (LOCALHANDLE) ;
classejan.hInstance = hCopia ;
classejan.hIcon = LoadIcon (NULL, IDI_APPLICATION) ;
classejan.hCursor = LoadCursor (NULL, IDC_ARROW) ;
classejan.hbrBackground = GetStockObject (WHITE_BRUSH) ;
classejan.lpszMenuName = NULL ;
classejan.lpszClassName = szClasseOla ;

RegisterClass (&classejan) ;

        // Registra a classe da janela-filha PASTE

classejan.style = CS_HREDRAW | CS_VREDRAW ;
classejan.lpfnWndProc = ProcJanPaste ;
classejan.cbClsExtra = 0 ;
classejan.cbWndExtra = sizeof (LOCALHANDLE) ;

```

```

classejan.hInstance = hCopia ;
classejan.hIcon     = NULL;
classejan.hCursor   = LoadCursor (NULL, IDC_ARROW) ;
                    classejan.hbrBackground = GetStockObject (WHITE_BRUSH) ;
classejan.lpszMenuName = NULL ;
classejan.lpszClassName = szClassePaste;

RegisterClass (&classejan) ;

                    // Registra a classe da janela-filha RETANG

classejan.style      = CS_HREDRAW | CS_VREDRAW ;
classejan.lpfnWndProc = ProcJanRetang ;
classejan.cbClsExtra  = 0 ;
classejan.cbWndExtra  = sizeof (LOCALHANDLE) ;
classejan.hInstance  = hCopia ;
classejan.hIcon      = NULL ;
classejan.hCursor    = LoadCursor (NULL, IDC_ARROW) ;
                    classejan.hbrBackground = GetStockObject (WHITE_BRUSH) ;
                    classejan.lpszMenuName = NULL ;
classejan.lpszClassName = szClasseRetang ;

RegisterClass (&classejan) ;
}

                    // Obtem os handles de tres menus e submenus possiveis

hMenuInic  = LoadMenu (hCop, "MdiMenuInic") ;
hMenuOla   = LoadMenu (hCop, "MdiMenuOla") ;
hMenuRetang = LoadMenu (hCop, "MdiMenuRetang") ;
hMenuInicJanela = GetSubMenu (hMenuInic, INIC_MENU_POS) ;
hMenuOlaJanela  = GetSubMenu (hMenuOla,  OLA_MENU_POS) ;
hMenuRetangJanela = GetSubMenu (hMenuRetang, RETANG_MENU_POS) ;

                    // Carrega a tabela de aceleradores

hAcelera = LoadAccelerators (hCop, "MdiAcelera") ;

                    // Cria a janela de base

hjanBase = CreateWindow (szClasseBase, "Servidor FTP de Vídeo e Áudio",
                        WS_OVERLAPPEDWINDOW | WS_CLIPCHILDREN,
                        CW_USEDEFAULT, CW_USEDEFAULT,
                        CW_USEDEFAULT, CW_USEDEFAULT,
                        NULL, hMenuInic, hCopia, NULL) ;

hjanCliente = GetWindow (hjanBase, GW_CHILD) ;

```

```

ShowWindow (hjanBase, nCmdMostrar) ;
UpdateWindow (hjanBase) ;

    // Entra no laço de mensagem modificado

while (GetMessage (&msg, NULL, 0, 0))
    {
        if (!TranslateMDISysAccel (hjanCliente, &msg) &&
            !TranslateAccelerator (hjanBase, hAcelera, &msg))
            {
                TranslateMessage (&msg) ;
                DispatchMessage (&msg) ;
            }
    }

    // Limpa tudo apagando os menus nao-anexados

DestroyMenu (hMenuOla) ;
DestroyMenu (hMenuRetang);
return msg.wParam ;
    }

void PoeTitulo (HWND hjanCliente, char * szNomeTitulo)
    {
        char szTitulo [64 + _MAX_FNAME + _MAX_EXT];
        sprintf (szTitulo, "%s - %s", (LPSTR) szNomeAplic, (LPSTR) (szNomeTitulo [0] ?
szNomeTitulo : SEMTITULO));

        SetWindowText (hjanCliente, szTitulo);
    }

long FAR PASCAL _export ProcJanBase (HWND hjan, UINT mensagem, UINT wParam,
LONG lParam)
    {
        WORD wReturn;
        char szMsg[80];
        int tamanho;
        static HWND    hjanCliente ;
        static HWND    hjanProxVis;
        static BOOL    bPrecisaSalvar = FALSE;
        static char szNomeArq [_MAX_PATH];
        char    arqabrir;
        static char szNomeTitulo [_MAX_FNAME + _MAX_EXT];
        CLIENTCREATESTRUCT criacliente ;
        FARPROC    lpfnEnum ;
        static HWND    hjanFilha ;
        MDICREATESTRUCT criamdi ;

```

```

lpfnProcDlgSobre = MakeProcInstance ((FARPROC) ProcDlgSobre,hCop) ;
lpfnProcDlgAjuda = MakeProcInstance ((FARPROC) ProcDlgAjuda,hCop) ;
lpfnProcDlgEnvia = MakeProcInstance ((FARPROC) ProcDlgEnvia,hCop) ;
lpfnProcDlgRecebe = MakeProcInstance ((FARPROC) ProcDlgRecebe,hCop) ;
lpfnProcDlgDemo = MakeProcInstance ((FARPROC) ProcDlgDemo,hCop) ;

switch (mensagem)
    {
        case WM_CREATE:          // Cria a janela-cliente

            criacliente.hWindowMenu = hMenuInicJanela ;
            criacliente.idFirstChild = IDM_PRIMFILHA ;
            hjanCliente = CreateWindow ("MDICLIENT", NULL,
                WS_CHILD | WS_CLIPCHILDREN | WS_VISIBLE,
                0, 0, 0, 0, hjan, 1, hCop,
                (LPSTR) &criacliente) ;
            hjanEditar = CreateWindow ("edit", NULL, WS_CHILD | WS_VISIBLE |
                WS_HSCROLL | WS_VSCROLL |
                WS_BORDER | ES_LEFT | ES_MULTILINE |
                ES_NOHIDSESEL | ES_AUTOHSCROLL | ES_AUTOVSCROLL,
                0, 0, 0, 0, hjanCliente, EDITID, hCop, NULL);
            SendMessage (hjanEditar, EM_LIMITTEXT, 32000, 0L);
            InicializaArq (hjanCliente);

            // InicializaFonte (hjanEditar);

            msgProcuraTroca = RegisterWindowMessage (FINDMSGSTRING);
            lstrcpy (szNomeArq, (LPSTR)
                (((LPCREATESTRUCT) lParam) -> lpCreateParams));
            if (lstrlen (szNomeArq) > 0)
                {
                    GetFileTitle (szNomeArq, szNomeTitulo, sizeof (szNomeTitulo));
                    if (!LerArquivo (hjanEditar, szNomeArq))
                        MessageBox (hjanCliente, "O arquivo não pôde ser lido", "Tratamento de
Imagens",
                            MB_ICONEXCLAMATION | MB_OK);
                }
            PoeTitulo (hjanCliente, szNomeTitulo);
            return 0;

        case WM_COMMAND:
            switch (wParam)
                {
                    case IDM_NOVOOLA:

                        // Cria a janela-filha Ola

```

```

criamdi.szClass = szClasseOla ;
criamdi.szTitle = "Ola" ;
criamdi.hOwner = hCop ;
criamdi.x      = CW_USEDEFAULT ;
criamdi.y      = CW_USEDEFAULT ;
criamdi.cx     = CW_USEDEFAULT ;
criamdi.cy     = CW_USEDEFAULT ;
criamdi.style  = 0 ;
criamdi.lParam = NULL ;
hjanFilha = (HWND) SendMessage (hjanCliente,
                               WM_MDICREATE, 0,
                               (long) (LPMDICREATESTRUCT) &criamdi) ;
return 0 ;

case IDM_NOVORETANG:

    // Cria a janela-filha Retang

    criamdi.szClass = szClasseRetang ;
    criamdi.szTitle = "Retangulos" ;
    criamdi.hOwner = hCop ;
    criamdi.x      = CW_USEDEFAULT ;
    criamdi.y      = CW_USEDEFAULT ;
    criamdi.cx     = CW_USEDEFAULT ;
    criamdi.cy     = CW_USEDEFAULT ;
    criamdi.style  = 0 ;
    criamdi.lParam = NULL ;
hjanFilha = (HWND) SendMessage (hjanCliente,
                               WM_MDICREATE, 0,
                               (long) (LPMDICREATESTRUCT) &criamdi) ;
return 0 ;

case IDM_FECHAR:

    // Fecha a janela ativa

    hjanFilha = LOWORD (SendMessage (hjanCliente,
                                     WM_MDIGETACTIVE, 0, 0L)) ;
if (SendMessage (hjanFilha, WM_QUERYENDSESSION, 0, 0L))
    SendMessage (hjanCliente, WM_MDIDESTROY,
                hjanFilha, 0L);
return 0 ;

case IDM_SAIR:

    // Sai do programa

```

```

SendMessage (hjan, WM_CLOSE, 0, 0L);
    return 0;

    // Mensagens para organizar as janelas

case IDM_LADO:
    SendMessage (hjanCliente, WM_MDITILE, 0, 0L);
    return 0;

case IDM_CASCATA:
    SendMessage (hjanCliente, WM_MDICASCADE, 0, 0L);
    return 0;

case IDM_ORGANIZAR:
    SendMessage (hjanCliente, WM_MDIICONARRANGE, 0, 0L);
    return 0;

case IDM_FECHARTUDO: // Tenta fechar todas as filhas

    lpfnEnum = MakeProcInstance ((FARPROC) ProcFecharEnum,
        hCop);
        EnumChildWindows (hjanCliente, lpfnEnum, 0L);
        FreeProcInstance (lpfnEnum);
    return 0;

case IDM_SOBRE:

if (DialogBox (hCop, "CaixaSobre", hjanCliente, lpfnProcDlgSobre));
    InvalidateRect (hjanCliente, NULL, TRUE);
    return 0;

case IDM_AJUDA:

if (DialogBox (hCop, "CaixaNao", hjanCliente, lpfnProcDlgAjuda));
    InvalidateRect (hjanCliente, NULL, TRUE);
    return 0;

case IDM_DEMO:

if (DialogBox (hCop, "CaixaDemo", hjanCliente, lpfnProcDlgDemo));
    InvalidateRect (hjanCliente, NULL, TRUE);
    return 0;

case IDM_ENVIA:

if (DialogBox (hCop, "CaixaEnvia", hjanCliente, lpfnProcDlgEnvia));

```

```
        InvalidateRect (hjanCliente, NULL, TRUE) ;
    return 0;

case IDM_RECEBE:

    if(DialogBox (hCop, "CaixaEnvia", hjanCliente, lpfProcDlgRecebe));
        InvalidateRect (hjanCliente, NULL, TRUE) ;
    return 0;

case IDM_SCANNER:

    wReturn = WinExec("iphoto",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;

case IDM_VIDEO:

    wReturn = WinExec("vblaster",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;

case IDM_PAINTB:

    wReturn = WinExec("pbrush.exe",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;

case IDM_EXCEL:

    wReturn = WinExec("excel.exe",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;
```

```
case IDM_WAV :
    wReturn = WinExec("mplayer.exe",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;

case IDM_MPEG :

    wReturn = WinExec("xing_ply.exe",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;

case IDM_COREL:

    wReturn = WinExec("coreldr.exe",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;

case IDM_ACTION:

    wReturn = WinExec("action.exe",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
    return 0;

case IDM_AUTHOR:

    wReturn = WinExec("asw.exe",SW_SHOW);
    if (wReturn < 32)
    {
        sprintf (szMsg, "WinExec falhou. Código de erro = %d",wReturn);
        MessageBox(hjan,szMsg,"Erro",MB_ICONSTOP);
    }
}
```

```

    return 0;

case IDM_PASTE:

    criamdi.szClass = szClassePaste;
    criamdi.szTitle = "Tratamento de Imagens - Editar/Paste" ;
    criamdi.hOwner = hCop ;
    criamdi.x      = CW_USEDEFAULT ;
    criamdi.y      = CW_USEDEFAULT ;
    criamdi.cx     = CW_USEDEFAULT ;
    criamdi.cy     = CW_USEDEFAULT ;
    criamdi.style  = CS_HREDRAW | CS_VREDRAW;
    criamdi.lParam = NULL ;
    hjanFilha = (HWND) SendMessage (hjanCliente,
        WM_MDICREATE, 0,
        (long) (LPMDCREATESTRUCT) &criamdi);
    SendMessage (hjanCliente, WM_MDITILE, 0, 0L) ;
return 0;

case IDM_ABRIR:

    if (DlgAbrirArquivo(hjanCliente,szNomeArq, szNomeTitulo));
    {
        PoeTitulo (hjanCliente, szNomeTitulo);
        bPrecisaSalvar = FALSE;
        tamanho = strlen(szNomeArq);
        if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] == 'B' &
szNomeArq[tamanho-2] == 'M' & szNomeArq[tamanho-1] == 'P')
        {
            parms.segEnv = 0;
            strcpy(aux, " ");
            strcat(aux,szNomeArq);
            parms.lpszCmdLine = aux;
            parms.lpwShow = (LPWORD) awShow;
            parms.lpwReserved = (LPWORD) NULL;
            hinstMod = LoadModule("pbrush.exe", &parms);
            if ((UINT) hinstMod < 32)
            {
                sprintf(szMsg, "LoadModule failed; error code = %d",
                    hinstMod);
                MessageBox(hwnd, szMsg, "Error", MB_ICONSTOP);
            }
        }
        if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] == 'P' &
szNomeArq[tamanho-2] == 'C' & szNomeArq[tamanho-1] == 'X')
        {

```

```

        strcpy(aux, " ");
        strcat(aux, szNomeArq);
        parms.segEnv = 0;
        parms.lpszCmdLine = aux;
        parms.lpwShow = (LPWORD) awShow;
parms.lpwReserved = (LPWORD) NULL;
        hinstMod = LoadModule("pbrush.exe", &parms);
    }

    if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] == 'A' &
szNomeArq[tamanho-2] == 'V' & szNomeArq[tamanho-1] == 'I')
    {
        strcpy(aux, " ");
        strcat(aux, szNomeArq);
        parms.segEnv = 0;
        parms.lpszCmdLine = aux;
        parms.lpwShow = (LPWORD) awShow;
        parms.lpwReserved = (LPWORD) NULL;
        hinstMod = LoadModule("mplayer.exe", &parms);
    }

    if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] == 'T' &
szNomeArq[tamanho-2] == 'I' & szNomeArq[tamanho-1] == 'F')
    {
        strcpy(aux, " ");
        strcat(aux, szNomeArq);
        parms.segEnv = 0;
        parms.lpszCmdLine = aux;
        parms.lpwShow = (LPWORD) awShow;
        parms.lpwReserved = (LPWORD) NULL;
        hinstMod = LoadModule("iphoto.exe", &parms);
    }

    if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] == 'X' &
szNomeArq[tamanho-2] == 'L' & szNomeArq[tamanho-1] == 'S')
    {
        strcpy(aux, " ");
        strcat(aux, szNomeArq);
        parms.segEnv = 0;
        parms.lpszCmdLine = aux;
        parms.lpwShow = (LPWORD) awShow;          /* mostra filha
como ícone */
        parms.lpwReserved = (LPWORD) NULL;        /* deve ser
NULL */
        hinstMod = LoadModule("excel.exe", &parms);
    }

    if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] == 'M' &
szNomeArq[tamanho-2] == 'P' & szNomeArq[tamanho-1] == 'G')

```

```

        {
            strcpy(aux, " ");
            strcat(aux, szNomeArq);
            parms(segEnv) = 0;
            parms(lpCmdLine) = aux;
            parms(lpShow) = (LPWORD) awShow;          /* mostra filha
como ícone */
            parms(lpReserved) = (LPWORD) NULL;       /* deve ser
NULL */
            hinstMod = LoadModule("mpegply.exe", &parms); /* Versao
WING */
            if ((UINT) hinstMod < 32)
                {
                    sprintf(szMsg, "LoadModule failed; error code = %d",
                        hinstMod);
                    MessageBox(hwnd, szMsg, "Error", MB_ICONSTOP);
                }
        }

        if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] ==
'W' & szNomeArq[tamanho-2] == 'A' & szNomeArq[tamanho-1] == 'V')
            {
                strcpy(aux, " ");
                strcat(aux, szNomeArq);
                parms(segEnv) = 0;
                parms(lpCmdLine) = aux;
                parms(lpShow) = (LPWORD) awShow;          /* mostra filha
como ícone */
                parms(lpReserved) = (LPWORD) NULL;       /* deve ser
NULL */
                hinstMod = LoadModule("mplayer.exe", &parms);
                if ((UINT) hinstMod < 32) {
                    sprintf(szMsg, "LoadModule failed; error code = %d",
                        hinstMod);
                    MessageBox(hwnd, szMsg, "Error", MB_ICONSTOP);
                }
            }
    }

        if (szNomeArq[tamanho-4] == '.' & szNomeArq[tamanho-3] ==
'M' & szNomeArq[tamanho-2] == 'P' & szNomeArq[tamanho-1] == '2')
            {
                strcpy(aux, " ");
                strcat(aux, szNomeArq);
                parms(segEnv) = 0;
            }

```

```

        parms.lpszCmdLine = aux;
        parms.lpwShow = (LPWORD) awShow;          /* mostra filha
como ícone */
        parms.lpwReserved = (LPWORD) NULL;       /* deve ser
NULL */
        hinstMod = LoadModule("xing_ply.exe", &parms);
        if ((UINT) hinstMod < 32) {
            sprintf(szMsg, "LoadModule failed; error code = %d",
                hinstMod);
            MessageBox(hwnd, szMsg, "Error", MB_ICONSTOP);
        }
    }

    szNomeArq[0] = '\0';
    szNomeTitulo[0] = '\0';
    bPrecisaSalvar = FALSE;
    return 0;

default:    // Passa `a janela ativa
hjanFilha = LOWORD (SendMessage (hjanCliente,
    WM_MDIGETACTIVE, 0, 0L)) ;
if (IsWindow (hjanFilha))
    SendMessage (hjanFilha, WM_COMMAND,
        wParam, lParam) ;
break ;    // e depois `a DefFrameProc
    }
break ;

case WM_QUERYENDSESSION:
case WM_CLOSE:    // Tenta fechar todas as filhas

SendMessage (hjan, WM_COMMAND, IDM_FECHARTUDO, 0L) ;

if (NULL != GetWindow (hjanCliente, GW_CHILD))
    return 0 ;

break ; // ou seja, chama DefFrameProc ;
}
// Passa as mensagens nao processadas `a DefFrameProc (nao DefWindowProc)

return DefFrameProc (hjan, hjanCliente, mensagem, wParam, lParam) ;
}

BOOL FAR PASCAL _export ProcFecharEnum (HWND hjan, LONG lParam)
{

```

```

if (GetWindow (hjan, GW_OWNER)) // verifica o titulo do icone
    return 1 ;

SendMessage (GetParent (hjan), WM_MDIRESTORE, hjan, 0L) ;

if (!SendMessage (hjan, WM_QUERYENDSESSION, 0, 0L))
    return 1 ;

SendMessage (GetParent (hjan), WM_MDIDESTROY, hjan, 0L) ;
return 1 ;
}

long FAR PASCAL _export ProcJanOla (HWND hjan, UINT mensagem, UINT wParam,
                                   LONG lParam)
{
    static COLORREF MatrizCorTxt [] = { RGB (0, 0, 0), RGB (255, 0, 0),
                                        RGB (0, 255, 0), RGB (0, 0, 255),
                                        RGB (255, 255, 255) } ;
    static HWND hjanCliente, hjanBase ;
    HDC hdc ;
    HMENU hMenu ;
    LOCALHANDLE hDadosOla ;
    NPDADOSOLA npDadosOla ;
    PAINTSTRUCT ps ;
    RECT retang ;

    switch (mensagem)
    {
        case WM_CREATE:
            // Aloca memoria para os dados privados da janela

            hDadosOla = LocalAlloc (LMEM_MOVEABLE | LMEM_ZEROINIT,
sizeof (DADOSOLA)) ;

            npDadosOla = (NPDADOSOLA) LocalLock (hDadosOla) ;
            npDadosOla->nCor = IDM_NEGRO ;
            npDadosOla->corTexto = RGB (0, 0, 0) ;
            LocalUnlock (hDadosOla) ;
            SetWindowWord (hjan, 0, hDadosOla) ;

            // Salva alguns handles da janela

            hjanCliente = GetParent (hjan) ;
            hjanBase = GetParent (hjanCliente) ;
            return 0 ;
    }
}

```

```

case WM_COMMAND:
    switch (wParam)
    {
        case IDM_NEGRO:
        case IDM_VERMELHO:
        case IDM_VERDE:
        case IDM_AZUL:
        case IDM_BRANCO:
            // Altera a cor do texto

            hDadosOla = GetWindowWord (hjan, 0) ;
            npDadosOla = (NPDADOSOLA) LocalLock (hDadosOla) ;
            hMenu = GetMenu (hjanBase) ;
            CheckMenuItem (hMenu, npDadosOla->nCor,
                F_UNCHECKED) ;
            npDadosOla->nCor = wParam ;
            CheckMenuItem (hMenu, npDadosOla->nCor,
                MF_CHECKED) ;
            npDadosOla->corTexto =
                MatrizCorTxt [wParam - IDM_NEGRO] ;
            LocalUnlock (hDadosOla) ;
            InvalidateRect (hjan, NULL, FALSE) ;
        }
    return 0 ;

case WM_PAINT:

    // Pinta a janela

    hdc = BeginPaint (hjan, &ps) ;
    hDadosOla = GetWindowWord (hjan, 0) ;
    npDadosOla = (NPDADOSOLA) LocalLock (hDadosOla) ;
    SetTextColor (hdc, npDadosOla->corTexto) ;
    LocalUnlock (hDadosOla) ;
    GetClientRect (hjan, &retang) ;
    DrawText (hdc, "Ola, Pessoal!", -1, &retang,
        DT_SINGLELINE | DT_CENTER | DT_VCENTER) ;
    EndPaint (hjan, &ps) ;
    return 0 ;

case WM_MDIACTIVATE:

    // Define o menu OLA se estiver ganhando o foco

    if (wParam == TRUE)
        SendMessage (hjanCliente, WM_MDISETMENU, 0,
            MAKELONG (hMenuOla, hMenuOlaJanela)) ;

```

```

// marca o desmarca o item do menu

hDadosOla = GetWindowWord (hjan, 0) ;
npDadosOla = (NPDADOSOLA) LocalLock (hDadosOla) ;
CheckMenuItem (hMenuOla, npDadosOla->nCor,
    wParam ? MF_CHECKED : MF_UNCHECKED) ;
LocalUnlock (hDadosOla) ;

// Define o menu Inic se estiver perdendo o foco

if (wParam == FALSE)
    SendMessage (hjanCliente, WM_MDISETMENU, 0,
        MAKELONG (hMenuInic, hMenuInicJanela)) ;

DrawMenuBar (hjanBase) ;
return 0 ;

case WM_QUERYENDSESSION:
case WM_CLOSE:
    if (IDOK != MessageBox (hjan, "Quer fechar a janela", "Ola",
        MB_ICONQUESTION | MB_OKCANCEL))
        return 0 ;
    break ; // ou seja, chama DefMDIChildProc

case WM_DESTROY:
    hDadosOla = GetWindowWord (hjan, 0) ;
    LocalFree (hDadosOla) ;
    return 0 ;
}

// Passa a mensagem nao processada `a DefMDIChildProc

return DefMDIChildProc (hjan, mensagem, wParam, lParam) ;
}

long FAR PASCAL _export ProcJanRetang (HWND hjan, UINT mensagem, UINT wParam,
    LONG lParam)
{
    static HWND hjanCliente, hjanBase ;
    HPEN hPincel ;
    HDC hdc ;
    LOCALHANDLE hDadosRetang ;
    NPDADOSRETANG npDadosRetang ;
    PAINTSTRUCT ps ;
    short xEsq, xDir, yTopo, yBaixo, nVermelho, nVerde, nAzul ;

    switch (mensagem)

```

```

    {
case WM_CREATE:
    // Aloca memoria para os dados privados da janela

    hDadosRetang = LocalAlloc (LMEM_MOVEABLE | LMEM_ZEROINIT,
        sizeof (DADOSRETANG)) ;

    SetWindowWord (hjan, 0, hDadosRetang) ;

    // Inicializa o temporizador

    SetTimer (hjan, 1, 250, NULL) ;

    // Salva alguns handles de janela

    hjanCliente = GetParent (hjan) ;
    hjanBase = GetParent (hjanCliente) ;
    return 0 ;

case WM_SIZE: // Salva o tamanho da janela

    hDadosRetang = GetWindowWord (hjan, 0) ;
    npDadosRetang = (NPDADOSRETANG) LocalLock (hDadosRetang) ;
    npDadosRetang->cxCliente = LOWORD (lParam) ;
    npDadosRetang->cyCliente = HIWORD (lParam) ;
    LocalUnlock (hDadosRetang) ;

    break ; // WM_SIZE precisa ser processada por DefMDIChildProc

case WM_TIMER: // Exibe um retangulo aleatorio

    hDadosRetang = GetWindowWord (hjan, 0) ;
    npDadosRetang = (NPDADOSRETANG) LocalLock (hDadosRetang) ;
    xEsq = rand () % npDadosRetang->cxCliente ;
    xDir = rand () % npDadosRetang->cxCliente ;
    yTopo = rand () % npDadosRetang->cyCliente ;
    yBaixo = rand () % npDadosRetang->cyCliente ;
    nVermelho = rand () & 255 ;
    nVerde = rand () & 255 ;
    nAzul = rand () & 255 ;
    hdc = GetDC (hjan) ;
    hPincel = CreateSolidBrush (RGB (nVermelho, nVerde, nAzul)) ;
    SelectObject (hdc, hPincel) ;
    Rectangle (hdc, min (xEsq, xDir), min (yTopo, yBaixo),
        max (xEsq, xDir), max (yTopo, yBaixo)) ;
    ReleaseDC (hjan, hdc) ;
    DeleteObject (hPincel) ;

```

```

        LocalUnlock (hDadosRetang) ;
                return 0 ;

case WM_PAINT:      // Limpa a janela

        InvalidateRect (hjan, NULL, TRUE) ;
        hdc = BeginPaint (hjan, &ps) ;
        EndPaint (hjan, &ps) ;
        return 0 ;

case WM_MDIACTIVATE: // Define o menu apropriado
        if (wParam == TRUE)
                SendMessage (hjanCliente, WM_MDISETMENU, 0,
                        MAKELONG (hMenuRetang, hMenuRetangJanela)) ;
        else
                SendMessage (hjanCliente, WM_MDISETMENU, 0,
                        MAKELONG (hMenuInic, hMenuInicJanela)) ;
        DrawMenuBar (hjanBase) ;
        return 0 ;

case WM_DESTROY:
        hDadosRetang = GetWindowWord (hjan, 0) ;
                LocalFree (hDadosRetang) ;
        KillTimer (hjan, 1) ;
        return 0 ;
}
// Passa a mensagem nao processada `a DefMDIChildProc

return DefMDIChildProc (hjan, mensagem, wParam, lParam) ;
}

BOOL FAR PASCAL _export ProcDlgSobre (HWND hDlg, UINT mensagem, UINT wParam,

                                LONG lParam)

{

switch (mensagem)
{

case WM_COMMAND:
        switch (wParam)
                {

case IDOK:
                EndDialog (hDlg, TRUE) ;
                return TRUE ;

case IDCANCEL:

```

```

        EndDialog (hDlg, FALSE);
        return TRUE;
    }
    break;
}
return FALSE;
}

```

```

BOOL FAR PASCAL _export ProcDlgAjuda (HWND hDlg, UINT mensagem, UINT wParam,
    LONG lParam)

```

```

{
    switch (mensagem)
    {
        case WM_COMMAND:
            switch (wParam)
            {
                case IDOK:
                    EndDialog (hDlg, TRUE);
                    return TRUE;
                case IDCANCEL:
                    EndDialog (hDlg, FALSE);
                    return TRUE;
            }
            break;
    }
    return FALSE;
}

```

```

BOOL FAR PASCAL _export ProcDlgDemo(HWND hDlg, UINT mensagem, UINT wParam,
    LONG lParam)

```

```

{
    switch (mensagem)
    {

        case WM_COMMAND:
            switch (wParam)

```

```

    {
    case IDOK:
        EndDialog (hDlg, TRUE) ;
        return TRUE ;

    case IDCANCEL:
        EndDialog (hDlg, FALSE) ;
        return TRUE ;

    }

    break ;
}

return FALSE ;
}

void MostraTecla (HWND hDlg, int iTipo, char * szMensa, UINT wParam, LONG lParam)
{
    char szBuffer[80];
    HDC hdc;

    ScrollWindow (hDlg, 0, -40, &retang, &retang);
    hdc = GetDC(hDlg);
    SelectObject (hDlg, GetStockObject (SYSTEM_FIXED_FONT));
    TextOut (hDlg, 10, retang.bottom - 40, szBuffer, 80);
    strncpy (parms.lpszCmdLine, szBuffer, strlen(szBuffer));
    ReleaseDC (hDlg, hdc);
    ValidateRect (hDlg, NULL);
}

BOOL FAR PASCAL _export ProcDlgEnvia(HWND hDlg, UINT mensagem, UINT wParam,
    LONG lParam)

{

    DWORD dwIndex, cbItemString;

    WORD cListItems;
    PSTR psz;
    static char szBuf[16]= "";
    WORD wReturn;

    switch (mensagem)
        {

```

```

case WM_CHAR :
    MostraTecla (hDlg, 1, "WM_CHAR", wParam, lParam);
    return 0;

case WM_COMMAND :
    switch (wParam)
    {
        case IDOK :
            parms.lpszCmdLine = szBuf;
            parms.lpwShow = (LPWORD) awShow;
            parms.lpwReserved = (LPWORD) NULL;
            hinstMod = LoadModule("ftp.exe", &parms);
            EndDialog (hDlg, TRUE) ;
            return FALSE ;

            case IDCANCEL:
                EndDialog (hDlg, TRUE) ;
                return FALSE ;

    }
case 101:
    switch HIWORD((DWORD)(lParam))
    {

        case CBN_DBLCLK :
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_GETCURSEL, 0, 0);
            if (dwIndex != CB_ERR)
                SendDlgItemMessage(hDlg, 101, CB_GETLBTEXT, (WPARAM)
dwIndex, (LPARAM) ((LPSTR) psz));
                strcpy (szBuf,psz);
                return FALSE;

            case CBN_SETFOCUS :
                dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.3"));
                dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.7"));
                dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.50"));
                dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.60"));
                dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.220"));
                dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.220.1"));
    }

```

```

        dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPPARAM)((LPSTR) "150.162.220.2"));
        dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPPARAM)((LPSTR) "150.162.1.222"));
        return FALSE;
    }

    break ;
}

return FALSE ;
}

```

```

BOOL FAR PASCAL _export ProcDlgRecebe(HWND hDlg, UINT mensagem, UINT
wParam,
        LONG lParam)

```

```

{

DWORD dwIndex, cbItemString;

WORD cListItems;
PSTR psz;
static char szBuf[16]= "";
WORD wReturn;

    switch (mensagem)
    {
    case WM_CHAR :
        MostraTecla (hDlg, 1, "WM_CHAR", wParam, lParam);
        return 0;

    case WM_COMMAND :
        switch (wParam)
        {
            case IDOK :
                parms.lpszCmdLine = szBuf;
                parms.lpwShow = (LPWORD) awShow;
                parms.lpwReserved = (LPWORD) NULL;
                hinstMod = LoadModule("ftp.exe", &parms);
                EndDialog (hDlg, TRUE) ;
                return FALSE ;

            case IDCANCEL:
                EndDialog (hDlg, TRUE) ;

```

```

        return FALSE ;

    }
case 101:
    switch HIWORD((DWORD)(lParam))
    {

        case CBN_DBLCLK :
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_GETCURSEL, 0, 0);
            if (dwIndex != CB_ERR)
                SendDlgItemMessage(hDlg, 101, CB_GETLBTEXT, (WPARAM)
dwIndex, (LPARAM) ((LPSTR) psz));
                strcpy (szBuf,psz);
                return FALSE;

        case CBN_SETFOCUS :
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.3"));
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.7"));
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.50"));
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.60"));
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.220"));
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.220.1"));
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.220.2"));
            dwIndex = SendDlgItemMessage(hDlg, 101, CB_ADDSTRING, 0,
(LPARAM)((LPSTR) "150.162.1.222"));
            return FALSE;

    }

    break ;
}

return FALSE ;
}

```

BOOL DlgAbrirArquivo (HWND hjanCliente, LPSTR lpstrNomeArq, LPSTR lpstrNomeTitulo)

```

    {

ofn.hwndOwner = hjanCliente;
ofn.lpstrFile = lpstrNomeArq;
ofn.lpstrFileName = lpstrNomeTitulo;
ofn.Flags = OFN_CREATEPROMPT;

return GetOpenFileName (&ofn);
}

```

```

BOOL LerArquivo (HWND hjanEditar, LPSTR lpstrNomeArq)

```

```

{
    long lTamanho;
    HANDLE hBuffer;
    int hArquivo;
    LPSTR lpstrBuffer;

if(-1 == (hArquivo = _lopen (lpstrNomeArq, OF_READ | OF_SHARE_DENY_WRITE)))
return FALSE;

    if ((lTamanho = TamanhoArquivo (hArquivo)) >= 32000)
    {
        _lclose (hArquivo);
return FALSE;
    }

if (NULL == (hBuffer = GlobalAlloc (GHND, lTamanho + 1 )))
{
    _lclose (hArquivo);
return FALSE;
}

lpstrBuffer = GlobalLock (hBuffer);
_lread (hArquivo, lpstrBuffer, (WORD) lTamanho);
_lclose (hArquivo);
lpstrBuffer [(WORD) lTamanho] = '\0';

SetWindowText (hjanEditar, lpstrBuffer);
GlobalUnlock (hBuffer);
GlobalFree (hBuffer);

return TRUE;
}

```

```

static long TamanhoArquivo ( int hArquivo)

```

```

{
long lPosicaoAtual = _llseek (hArquivo, 0L, 1);
long lTamanhoArquivo = _llseek (hArquivo, 0L, 2);

_llseek (hArquivo, lPosicaoAtual, 0);

return lTamanhoArquivo;
}

void InicializaArq (HWND hjanCliente)
{
static char *szFiltro[] = {
    "Arquivos [*.BMP]", "*.BMP",
    "Arquivos [*.PCX]", "*.PCX",
    "Arquivos [*.GIF]", "*.GIF",
    "Arquivos [*.TIF]", "*.TIF",
    "Arquivos [*.GIF]", "*.GIF",
    "Arquivos [*.GLI]", "*.GLI",
    "Arquivos [*.FLI]", "*.FLI",
    "Arquivos [*.AVI]", "*.AVI",
    "Arquivos [*.MPG]", "*.MPG",
    "Arquivos [*.WAV]", "*.WAV",
    "Arquivos [*.MP2]", "*.MP2",
    "Arquivos [*.XLS]", "*.XLS",
    "Todos Arquivos [ *.* ]", " *.*",
    "" };

ofn.lStructSize      = sizeof (OPENFILENAME);
ofn.hwndOwner        = hjanCliente;
ofn.hInstance        = NULL;
ofn.lpstrFilter      = szFiltro [0];
ofn.lpstrCustomFilter = NULL;
ofn.nMaxCustFilter   = 0;
ofn.nFilterIndex     = 0;
ofn.lpstrFile        = NULL;
ofn.nMaxFile         = _MAX_PATH;
ofn.lpstrFileTitle   = NULL;
ofn.nMaxFileTitle    = _MAX_FNAME + _MAX_EXT;
ofn.lpstrInitialDir  = NULL;
ofn.lpstrTitle       = NULL;
ofn.Flags            = 0;
ofn.nFileOffset      = 0;
ofn.nFileExtension   = 0;
ofn.lpstrDefExt      = "BMP";
ofn.lCustData        = 0L;
ofn.lpfnHook         = NULL;
ofn.lpTemplateName  = NULL;
}

```

```
long FAR PASCAL _export ProcJanPaste (HWND hjanFilha, UINT mensagem, UINT wParam,
LONG lParam)
```

```
{
static HWND    hjanProxVis;
PAINTSTRUCT   ps;
HDC           hdc, hdcMem;
HANDLE        hGMem;
LPSTR         lpGMem;
RECT          retang;
static HBITMAP hMapabits;
BITMAP        mb;
static char    cxCliente, cyCliente;
POINT         ptTamanho, ptOrg;

switch (mensagem)
{

case WM_SIZE:
cxCliente = LOWORD (lParam);
cyCliente = HIWORD (lParam);
return 0;

case WM_CREATE:
hjanFilha = SetClipboardViewer (hjanProxVis);
OpenClipboard (hjanFilha);
hMapabits = GetClipboardData (CF_BITMAP);
CloseClipboard ();
return 0;

case WM_CHANGECHAIN :
if (wParam == hjanProxVis)
hjanProxVis = LOWORD(lParam);
else if (hjanProxVis)
SendMessage(hjanProxVis,mensagem,wParam,lParam);
return 0;

/* case WM_DRAWCLIPBOARD:
if (hjanFilha)
SendMessage(hjanFilha,mensagem,wParam,lParam);
InvalidateRect(hjanFilha,NULL,TRUE);
return 0;*/

case WM_PAINT :
hdc = BeginPaint (hjanFilha, &ps);
if (hMapabits != NULL)
```

```

        {
            SetCursor (LoadCursor (NULL, IDC_WAIT));
            hdcMem = CreateCompatibleDC (hdc);
            SelectObject (hdcMem, hMapabits);
            GetObject(hMapabits, sizeof(BITMAP),(LPSTR) &mb);
            SetStretchBltMode (hdc, COLORONCOLOR);
            StretchBlt (hdc, 0, 0, cxCliente+200,cyCliente+200, hdcMem, 0, 0,
                mb.bmWidth,mb.bmHeight,SRCCOPY);
            SetCursor(LoadCursor (NULL, IDC_ARROW));
            DeleteDC (hdcMem);
        }
        EndPaint (hjanFilha, &ps);
        ChangeClipboardChain (hjanFilha, hjanProxVis);
        return 0;

    case WM_DESTROY :
        ChangeClipboardChain (hjanFilha, hjanProxVis);
        PostQuitMessage (0) ;
        return 0 ;

    }
    return DefWindowProc (hjanFilha, mensagem, wParam, lParam);
}

/*-----
    IMAGEM.RC Lista de Recursos
    -----*/

#include <windows.h>
#include "imagem.h"

imagem ICON imagem.ico
Sos  ICON help.ico

MdiMenuInic MENU
{
    POPUP "&Arquivo"
    {
        POPUP "&Novo"
        {
            POPUP "&Desenho"
            {
                MENUITEM "&Paintbrush",    IDM_PAINTB
                MENUITEM "&Excel",        IDM_EXCEL
                MENUITEM "&Corel Draw",    IDM_COREL
            }
        }
    }
}

```

```

    MENUITEM "&Instant Artist",    IDM_NAO
    MENUITEM "&Outros",            IDM_NAO
}
POPUP "&Scanner"
{
    MENUITEM "&Iphoto",            IDM_SCANNER
    MENUITEM "&ScanEdit",          IDM_NAO
    MENUITEM "&Paintbrush",        IDM_PAINTB
    MENUITEM "&Outros",            IDM_NAO
}
POPUP "&Animação"
{
    MENUITEM "&AuthorWare",        IDM_AUTHOR
    MENUITEM "Auto&desk Animator",  IDM_NAO
    MENUITEM "A&ction",            IDM_ACTION
    MENUITEM "&Santa Fé Media Master", IDM_NAO
    MENUITEM "&Outros",            IDM_NAO
}
POPUP "&Vídeo"
{
    MENUITEM "&Vídeo Blaster",      IDM_VIDEO
    MENUITEM "Vídeo &Spigot",       IDM_NAO
    MENUITEM "Outros",             IDM_NAO
}
POPUP "Á&udio"
{
    MENUITEM "&Mplayer",            IDM_WAV
    MENUITEM "M&PEG Player",        IDM_MPEG
    MENUITEM "&Outros",            IDM_NAO
}
    MENUITEM SEPARATOR
MENUITEM "&Sair",                IDM_SAIR
}
    MENUITEM "&Abrir"              IDM_ABRIR
    MENUITEM SEPARATOR
    MENUITEM "&Sair",              IDM_SAIR
}

POPUP "&Editar"
{
    MENUITEM "&Colar",              IDM_PASTE
}

POPUP "&Janela"
{
    MENUITEM "&Cascata\tShift+F5",  IDM_CASCATA
    MENUITEM "&Lado a Lado\tShift+F4", IDM_LADO
    MENUITEM "&Fechar Tudo",        IDM_FECHARTUDO
}

```

}

POPUP "&amp;Transmissão"

```

{
  MENUITEM "&Envia",          IDM_ENVIA
  MENUITEM "&Recebe",       IDM_RECEBE
}

```

POPUP "A&amp;juda"

```

{
  MENUITEM "Aj&juda",        IDM_AJUDA
  MENUITEM SEPARATOR
  MENUITEM "&Sobre ...",    IDM_SOBRE
}

```

}

MdiMenuOla MENU

{

POPUP "&amp;Cor"

```

{
  MENUITEM "&Negro",          IDM_NEGRO
  MENUITEM "&Vermelho",      IDM_VERMELHO
  MENUITEM "Ve&rde",         IDM_VERDE
  MENUITEM "&Azul",          IDM_AZUL
  MENUITEM "&Branco",        IDM_BRANCO
}

```

POPUP "&amp;Janela"

```

{
  MENUITEM "&Cascata\tShift+F5",  IDM_CASCATA
  MENUITEM "&Lado a Lado\tShift+F4",  IDM_LADO
  MENUITEM "Organizar &Icones",      IDM_ORGANIZAR
  MENUITEM "&Fechar Tudo",          IDM_FECHARTUDO
}

```

}

MdiMenuRetang MENU

{

POPUP "&amp;Arquivo"

```

{
  MENUITEM "Novo &Ola",          IDM_NOVOOLA
  MENUITEM "Novo &Retangs",      IDM_NOVORETANG
  MENUITEM "&Fechar",            IDM_FECHAR
  MENUITEM SEPARATOR
  MENUITEM "&Sair",              IDM_SAIR
}

```

POPUP "&amp;Janela"

```

{
  MENUITEM "&Cascata\tShift+F5",    IDM_CASCATA
  MENUITEM "&Lado a Lado\tShift+F4",  IDM_LADO
  MENUITEM "Organizar &Icones",     IDM_ORGANIZAR
  MENUITEM "&Fechar Tudo",          IDM_FECHARTUDO
}
}

```

#### MdiAcelera ACCELERATORS

```

{
  VK_F5, IDM_CASCATA, VIRTKEY, SHIFT
  VK_F4, IDM_LADO,   VIRTKEY, SHIFT
}
#define TABGRP (WS_TABSTOP | WS_GROUP)

```

#### CaixaSobre DIALOG 20, 20, 160, 100

```

STYLE WS_POPUP | WS_CAPTION
CAPTION "Sobre Servidor FTP"
{
  CTEXT    "Servidor FTP de Vídeo e Áudio"  -1, 0, 12, 160, 8
  ICON     "Imagem"      -1,      8, 8, 0, 0
  CTEXT    "Programa de transmissão e recepção" -1, 0, 36, 160, 8
  CTEXT    "de Vídeo e Áudio"      -1, 0, 48, 160, 8
  CTEXT    "Cláudia J. Barenco"    -1, 0, 60, 160, 8
  CTEXT    "E-mail : barenco@inf.ufsc.br" -1, 0, 72, 160, 8
  DEFPUSHBUTTON "OK"      IDOK,    64, 84, 32, 14, WS_GROUP
}

```

#### CaixaAjuda DIALOG 20, 20, 160, 80

```

STYLE WS_POPUP | WS_CAPTION
CAPTION "Ajuda Servidor FTP"
{
  CTEXT    "Ajuda Imagem" -1, 0, 12, 160, 8
  ICON     "Sos"          -1, 41, 9, 0, 0
  CTEXT    "Opção ainda não disponível" -1, 0, 36, 160, 8
  DEFPUSHBUTTON "OK"      IDOK,    64, 60, 32, 14, WS_GROUP
}

```

#### CaixaNao DIALOG 20, 20, 160, 80

```

STYLE WS_POPUP | WS_CAPTION
CAPTION "Servidor FTP"
{
  CTEXT    "Servidor FTP de Vídeo e Áudio" -1, 0, 12, 160, 8
  ICON     "Imagem"      -1, 8, 8, 0, 0
  CTEXT    "Opção ainda não disponível" -1, 0, 36, 160, 8
}

```

```

DEFPUSHBUTTON "OK"    IDOK,    64, 60, 32, 14, WS_GROUP
}

```

```

CaixaDemo DIALOG 20, 20, 160, 80
STYLE WS_POPUP | WS_CAPTION
CAPTION "Servidor FTP"
{
ICON    "Imagem"      -1, 8, 8, 0, 0
CTEXT   "Versão demo não disponibiliza esta opção" -1, 0, 36, 160, 8
DEFPUSHBUTTON "OK"    IDOK,    64, 60, 32, 14, WS_GROUP
}

```

```

CaixaEnvia DIALOG 20, 20, 157, 80
STYLE WS_POPUP | WS_CAPTION | WS_GROUP
CAPTION "Endereço IP"
{
COMBOBOX 101, 20, 10, 67, 60, CBS_SIMPLE | CBS_HASSTRINGS | CBS_SORT
DEFPUSHBUTTON "OK", IDOK, 109, 12, 32, 14, BS_DEFPUSHBUTTON | WS_GROUP |
WS_TABSTOP
PUSHBUTTON "Cancel", IDCANCEL, 108, 38, 32, 14, WS_GROUP | WS_TABSTOP
}

```

```

/*-----
IMAGEM.H Arquivo de Cabecalho
-----*/

```

```

#define INIC_MENU_POS 0
#define OLA_MENU_POS 2
#define RETANG_MENU_POS 1

```

```

#define IDM_NOVOOLA 10
#define IDM_NOVORETANG 11
#define IDM_FECHAR 12
#define IDM_SAIR 13
#define IDM_ABRIR 14

```

```

#define IDM_NEGRO 20
#define IDM_VERMELHO 21
#define IDM_VERDE 22
#define IDM_AZUL 23
#define IDM_BRANCO 24

```

```

#define IDM_LADO 30
#define IDM_CASCATA 31
#define IDM_ORGANIZAR 32
#define IDM_FECHARTUDO 33

```

```
#define IDM_PRIMFILHA 100
```

```
#define IDM_SOBRE 34
```

```
#define IDM_AJUDA 35
```

```
#define IDM_PASTE 36
```

```
#define IDM_NAO 37
```

```
#define IDM_PAINTB 38
```

```
#define IDM_SCANNER 39
```

```
#define IDM_VIDEO 40
```

```
#define IDM_EXCEL 41
```

```
#define IDM_COREL 42
```

```
#define IDM_ACTION 43
```

```
#define IDM_AUTHOR 44
```

```
#define IDM_OPcoes 45
```

```
#define IDM_ENVIA 46
```

```
#define IDM_RECEBE 47
```

```
#define IDM_DEMO 48
```

```
#define IDM_WAV 49
```

```
#define IDM_MPEG 50
```

**ANEXO II**

**LISTAGEM DO MODELO  
SIMULADO FDDI.DOE**

```

33$    CREATE,      1,0.94::NEXT(42$);

42$    TRACE,      -1,"-Choosing from 2 options\n";
55$    BRANCH,     1:If,(Tnow-TToken(MM))<(TVX(MM)),22$,Yes:
        Else,21$,Yes;

22$    TRACE,      -1,"-Updating counter RECEBET \n";
57$    COUNT:     RECEBET,1:NEXT(25$);

25$    TRACE,      -1,"-Choosing from 2 options\n";
58$    BRANCH,     1:If,NQ(MM*2-1) == 0 .AND. NQ(MM*2) == 0,24$,Yes:
        Else,46$,Yes;

24$    TRACE,      -1,"-Making assignments\n";
59$    ASSIGN:     TTOKEN(MM)=TNOW:NEXT(53$);

53$    TRACE,      -1,"-Making assignments\n";
60$    ASSIGN:     mm=mm+1:NEXT(2$);

2$     TRACE,      -1,"-Choosing from 2 options\n";
61$    BRANCH,     1:If,MM>N,1$,Yes:
        Else,52$,Yes;

1$     TRACE,      -1,"-Making assignments\n";
62$    ASSIGN:     mm=1:NEXT(52$);

52$    TRACE,      -1,"-Delaying for time TOKEN_WAIT\n";
63$    DELAY:     TOKEN_WAIT:NEXT(42$);

46$    TRACE,      -1,"-Making assignments\n";
64$    ASSIGN:     TSTAMP(MM)=TNOW:NEXT(32$);

32$    TRACE,      -1,"-Choosing from 2 options\n";
65$    BRANCH,     1:If,NQ(MM*2-1) == 0,0$,Yes:
        Else,31$,Yes;

0$     TRACE,      -1,"-Making assignments\n";
66$    ASSIGN:     TTOKEN(MM)=TNOW:NEXT(53$);

31$    ASSIGN:     TTRANS=.000125:NEXT(50$);

50$    TRACE,      -1,"-Choosing from 2 options\n";
67$    BRANCH,     1:If,TNEG>TNOW-TSTAMP(MM),29$,Yes:
        Else,49$,Yes;

29$    TRACE,      -1,"-Choosing from 2 options\n";
68$    BRANCH,     1:If,NQ(MM*2-1) == 0,15$,Yes:
        Else,26$,Yes;

15$    TRACE,      -1,"-Updating counter FILAVAZIA \n";
70$    COUNT:     FILAVAZIA,1:NEXT(49$);

49$    TRACE,      -1,"-Choosing from 2 options\n";

```

71\$ BRANCH, 1:If,(TSTAMP(MM)-TTOKEN(MM))>TTRT,23\$, Yes:  
 Else,47\$, Yes;

23\$ ASSIGN: TTOKEN(MM)=TSTAMP(MM):NEXT(53\$);

47\$ TRACE, -1,"-Making assignments\n";

72\$ ASSIGN: TASSINC(MM)=TNOW:  
 TTRANS=0:  
 TRT(mm)=TSTAMP(MM)-TTOKEN(MM):NEXT(48\$);

48\$ TRACE, -1,"-Choosing from 2 options\n";

73\$ BRANCH, 1:If,(TTRT-TRT(MM))<(TNOW-TASSINC(MM)),20\$, Yes:  
 Else,18\$, Yes;

20\$ ASSIGN: TTOKEN(MM)=TSTAMP(MM):NEXT(53\$);

18\$ TRACE, -1,"-Choosing from 2 options\n";

74\$ BRANCH, 1:If,NQ(MM\*2) == 0,14\$, Yes:  
 Else,45\$, Yes;

14\$ TRACE, -1,"-Updating counter FILAVAZIA \n";

76\$ COUNT: FILAVAZIA,1:NEXT(30\$);

30\$ ASSIGN: TTOKEN(MM)=TSTAMP(MM):  
 MM=MM+1:NEXT(2\$);

45\$ TRACE, -1,"-Making assignments\n";

77\$ ASSIGN: THT(MM)=TRT(MM):NEXT(44\$);

44\$ TRACE, -1,"-Choosing from 2 options\n";

78\$ BRANCH, 1:Else,34\$, Yes:  
 If,TRT(MM)>=TTRT,43\$, Yes;

34\$ ASSIGN: tiratipo=2;

36\$ SIGNAL: MM\*2;

35\$ DELAY: TTRANS:NEXT(48\$);

43\$ TRACE, -1,"-Making assignments\n";

79\$ ASSIGN: TRT(MM)=0:NEXT(34\$);

26\$ ASSIGN: tiratipo=1;

38\$ SIGNAL: MM\*2-1;

37\$ DELAY: TTRANS:NEXT(50\$);

21\$ TRACE, -1,"-Updating counter RECEBET \n";

81\$ COUNT: RECEBET,1:NEXT(41\$);

41\$ TRACE, -1,"-Updating counter TATRASADO \n";

83\$ COUNT: TATRASADO,1:NEXT(24\$);

108\$ CREATE, 1,0:EXPO(0.94):MARK(Tentrada);

```

107$   ASSIGN:   MARCA=0:
          TIPO=DISC(.7,1,1,2):
          nm=6;
148$   STATION,   Audio;
99$    TRACE,     -1,"-Arrived to system at station Audio\n";
145$   ASSIGN:   Picture=Default;
124$   DELAY:    0.;
94$    TRACE,     -1,"-Transferred to next module\n";
118$   DELAY:    0.00:NEXT(13$);

13$    WHILE:    NN<21:NEXT(12$);

12$    TRACE,     -1,"-Duplicating entities\n";
149$   DUPLICATE: 1,VAIAUDIO:NEXT(9$);

9$     TRACE,     -1,"-Making assignments\n";
150$   ASSIGN:    NN=NN+1:NEXT(11$);

11$    ENDWHILE;
10$    DISPOSE;
VAIAUDIO STATION,   ESTAUDIO;
157$   TRACE,     -1,"-Arrived to station ESTAUDIO\n";
175$   ASSIGN:    SetIndex=MEMIDX(ESTAUDIO,M);
177$   STORE:     ESTAUDIO_SI;
178$   DELAY:    0.;
163$   UNSTORE:   ;
158$   DELAY:    0.000:NEXT(3$);

3$     TRACE,     -1,"-Making assignments\n";
179$   ASSIGN:    M=NN:NEXT(19$);

19$    TRACE,     -1,"-Choosing from 2 options\n";
180$   BRANCH,    1:If,tipo==1,17$,Yes:
          Else,16$,Yes;
17$    ASSIGN:    MARCA=m*2-1;
Filas  QUEUE,     fila(MARCA),94091,cheio;
51$    WAIT:     MARCA,1;
28$    SEIZE,    1:barramento,1;
27$    RELEASE:  barramento,1:NEXT(40$);

40$    TRACE,     -1,"-Updating Tally TACESSO \n";
182$   TALLY:    TACESSO,TNOW-TUACESSO(MM),1:NEXT(39$);

39$    TRACE,     -1,"-Making assignments\n";
183$   ASSIGN:    TUACESSO(MM)=TNOW:NEXT(54$);

54$    STATION,   saida;

```

```

191$    TRACE,      -1,"-Arrived to station saida\n";
221$    DELAY:      0.;
199$    COUNT:      saida_C,1;
194$    TALLY:      Tsaida,Interval(Tentrada),1;
184$    TRACE,      -1,"-Disposing entity\n";
192$    DISPOSE;
cheio   STATION,      saicheio;
229$    TRACE,      -1,"-Arrived to station saicheio\n";
259$    DELAY:      0.;
237$    COUNT:      bufcheio,1;
222$    TRACE,      -1,"-Disposing entity\n";
230$    DISPOSE;
16$     ASSIGN:      MARCA=m*2:NEXT(Filas);

291$    CREATE,      1:EXPO(0.94):MARK(Tentrada);
290$    ASSIGN:      MARCA=0:
                TIPO=DISC(.7,1,1,2):
                nn=1;
331$    STATION,      Video;
282$    TRACE,      -1,"-Arrived to system at station Video\n";
328$    ASSIGN:      Picture=Default;
307$    DELAY:      0.;
277$    TRACE,      -1,"-Transferred to next module\n";
301$    DELAY:      0.00:NEXT(8$);

8$     WHILE:      NN<11:NEXT(7$);

7$     TRACE,      -1,"-Duplicating entities\n";
332$    DUPLICATE:   1,VAIVIDEO:NEXT(4$);

4$     TRACE,      -1,"-Making assignments\n";
333$    ASSIGN:      nn=NN+1:NEXT(6$);

6$     ENDWHILE;
5$     DISPOSE;
VAIVIDEO STATION,  ESTVIDEO;
340$    TRACE,      -1,"-Arrived to station ESTVIDEO\n";
358$    ASSIGN:      SetIndex=MEMIDX(ESTVIDEO,M);
360$    STORE:      ESTVIDEO_S1;
361$    DELAY:      0.;
346$    UNSTORE:    ;
341$    DELAY:      0.000:NEXT(3$);

```

**ANEXO III**

**ENDEREÇOS INTERNET  
UTILIZADOS NO PROJETO**

- [I1] - decel.ecel.uwa.edu.au /users/michael
- [I2] - ftp.netcom.com /pub/cfogg/mpeg2
- [I3] - MPEG-L@netcom.com
- [I4] - s2k-ftp.cs.berkeley.edu /pub/multimedia/mpeg/movies
- [I5] - IMAGE-L@VM3090.EGE.EDU.TR
- [I6] - ftp.informatik.tu-muenchen.de /pub/comp/graphics/mpeg/mplex
- [I7] - ftp.cs.tu-berlin.de
- [I8] - ftp.crs4.it
- [I9] - ftp.comp.lancs.ac.uk
- [I10] - sunsite.unc.edu /pub/electronic-publications/iuma/audio\_utils
- [I11] - snm-people@zippy.telcom.arizona.edu
- [I12]- ftp.microsoft.com /peropsys/windows
- [I13]- info@invision.com

ANEXO IV

TRANSFORMADA DISCRETA  
DO COSSENO (DCT)

O DCT a partir de uma função contínua no tempo (como as ondas sonoras) ou no espaço (como uma imagem), chama  $S_x$ , a transforma em uma função contínua de frequência  $F_f$ .

O padrão DCT divide uma imagem em blocos de 8 por 8 *pixels* e executa uma transformação bi-dimensional com oito elementos DCT em cada bloco.

Neste algoritmo, as frequências são representadas com unidades que equivalem a quantidade de ciclos de oito amostras. Assim, uma frequência de um é o seno ou cosseno com um ciclo através das oito amostras.

Primeiramente será demonstrado o algoritmo da DFT (Discrete Fourier Transform), para a partir deste algoritmo entender-se o algoritmo DCT.

A definição oficial do DFT é :

$$F_f = \sum_{x=0}^7 S_x e^{-2\pi f_x/8} \quad (\text{A1})$$

Agora separando a parte real da imaginária :

$$F_f = \sum_{x=0}^7 S_x \cos\left(\frac{2\pi f_x}{8}\right) - i \sum_{x=0}^7 S_x \sin\left(\frac{2\pi f_x}{8}\right) \quad (\text{A2})$$

Nomeando cada parte separadamente :

$$F_{Rf} = \sum_{x=0}^7 S_x \cos\left(\frac{2\pi f_x}{8}\right) \quad (\text{A3})$$

$$F_{If} = \sum_{x=0}^7 S_x \sin\frac{2\pi f_x}{8} \quad (\text{A4})$$

Nós podemos escrever a transformada de Fourier na forma de uma matriz de cossenos e senos multiplicados pelo vetor das amostras  $S_x$ .

Sabe-se que  $\frac{2\pi}{8} = 45^\circ$  e que  $\sin(45^\circ) = \cos(45^\circ) = \frac{\sqrt{2}}{2}$ , que a partir de agora será chamado de  $r$ .

A representação da parte real em forma de matriz é :

$$\begin{array}{rcccccccc}
 F_{R0} & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & S_0 \\
 F_{R1} & & 1 & r & 0 & -r & -1 & -r & 0 & r & S_1 \\
 F_{R2} & & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & S_2 \\
 F_{R3} & & 1 & -r & 0 & r & -1 & r & 0 & -r & S_3 \\
 F_{R4} & = & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & S_4 \\
 F_{R5} & & 1 & -r & 0 & r & -1 & r & 0 & -r & S_5 \\
 F_{R6} & & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & S_6 \\
 F_{R7} & & 1 & r & 0 & -r & -1 & -r & 0 & r & S_7
 \end{array} \times$$

Podendo ser condensada a :

$$F_R = \mathbf{R} \times \mathbf{s}$$

A parte imaginária ficará :

$$\begin{array}{rcccccccc}
 F_{I0} & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 F_{I1} & & 0 & r & 1 & r & 0 & -r & -1 & -r \\
 F_{I2} & & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\
 F_{I3} & & 0 & r & -1 & r & 0 & -r & 1 & -r \\
 F_{I4} & = & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 F_{I5} & & 0 & -r & 1 & -r & 0 & r & -1 & r \\
 F_{I6} & & 0 & -1 & 0 & 1 & 0 & -1 & 0 & 1 \\
 F_{I7} & & 0 & -r & -1 & -r & 0 & r & 1 & r
 \end{array}$$

Simplificando, ficará :

$$F_I = \mathbf{S} \times \mathbf{s}$$

A linha 0 representa a frequência 0 (também chamada de componente DCT), a linha 1 representa a frequência 1, e assim por diante.

Os oito elementos DFT produzem 16 números (8 reais e 8 imaginários). Assim, algumas transformações têm que ser realizadas para se produzir 8 números independentes.

Pode-se notar que, na matriz  $\mathbf{R}$ , a linha 3 é igual à linha 5, a linha 2 é igual à linha 6 e a linha 1 é igual à linha 7. Assim, as linhas 5, 6 e 7 produzem informações redundantes, havendo somente 5 linhas independentes.

Da mesma maneira, na matriz  $\mathbf{S}$ , as linhas 0 e 4 são nulas, a linha 3 é igual à linha 5 negativa, a linha 2 é igual à linha 6 negativa e a linha 1 é igual à linha 7 negativa. Somente as linhas 1, 2 e 3 representam informações interessantes.

Agora gera-se uma nova matriz que contém somente as informações interessantes : a parte real e imaginária de F1, F2 e F3 e a parte real de F4 e F0.

$$\begin{array}{rcccccccc}
 F_{R0} & & 1 & 1 & 1 & 1 & 1 & 1 & 1 & S_0 \\
 F_{R1} & & 1 & r & 0 & -r & -1 & -r & 0 & r & S_1 \\
 F_{I1} & & 0 & r & 1 & r & 0 & -r & -1 & -r & S_2 \\
 F_{R2} & & 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 & S_3 \\
 F_{I2} & = & 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 & S_4 \\
 F_{R3} & & 1 & -r & 0 & r & -1 & r & 0 & -r & S_5 \\
 F_{I3} & & 0 & r & -1 & r & 0 & -r & 1 & -r & S_6 \\
 F_{R4} & & 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & S_7
 \end{array} \times$$

A definição oficial do algoritmo DCT é :

$$G_{u/2} = \frac{1}{2} K_u \sum_{x=0}^7 S_x \text{Cos} \left( \frac{(2x+1)u\pi}{16} \right), \quad (\text{A5})$$

Onde  $k_0 = \frac{\sqrt{2}}{2}$  e  $K_u = 1$ , para  $u \neq 0$ .

Definindo  $c_j = \text{Cos} \left( \frac{j\pi}{16} \right)$ , pode-se representar o algoritmo DCT como uma matriz assim :  
Simplicando,  $G = C \times s$ .

A matriz  $C$  não é simétrica mas ela possui a propriedade de ortogonalidade. Pode-se representar a linha 0 como  $K_0 \times C_0$ , onde  $K_0 = C_4$  e  $C_0 = 1$ , assim é verificado que se multiplicarmos cada linha por ela mesma obtém-se o valor 1. Em outras palavras, a multiplicação da matriz  $F$  pela transposta dela mesma, resulta em uma matriz identidade [Murdoch1972], significando que a matriz DCT é uma matriz rotação em um espaço octodimensional.

Comparando-se os algoritmos DFT e DCT pode-se verificar a maior simplicidade do algoritmo DFT com relação ao algoritmo DCT. No algoritmo DFT trabalha-se apenas com multiplicação de números simples : 0, 1, -1,  $r$  e  $-r$ . No algoritmo DCT são realizadas 64 multiplicações de números bastante complexas.

O algoritmo DFT necessita de coeficientes de altas frequências para realizar a aproximação das descontinuidades de uma cópia de uma onda para outra. O algoritmo DCT trabalha com ondas triangulares, uma onda é gerada através da onda original alternada pela sua reflexão, assim não há descontinuidade e, conseqüentemente, há menor necessidade de haver coeficientes de altas frequências.

As imagens não possuem muita energia nos coeficientes de alta frequência [Corner1993]; sendo assim, o algoritmo DCT é o mais apropriado para a compactação das imagens.

**ANEXO V**

**DESCRIÇÃO DOS OBJETOS  
DA MIB FDDI [Case1993] E  
MIB II SNMP [McCloghrie1991]  
UTILIZADOS NO PROJETO**

```
FDDI-SMT73-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    Counter
```

```
    FROM RFC1155-SMI
```

```
OBJECT-TYPE
```

```
    FROM RFC1212;
```

```
fddi      OBJECT IDENTIFIER ::= {transmission 15}
```

```
fddimib   OBJECT IDENTIFIER ::= {fddi 73}
```

```
-- Definição de alguns tipos de dados especiais
```

```
FddiTimeNano ::= INTEGER (0..2147483647)
```

```
-- Esse tipo de dado especifica 1 nanosegundo como um valor inteiro
```

```
FddiTimeMilli ::= INTEGER (0..2147483647)
```

```
-- Esse tipo de dado é utilizado em muitos FDDI timers.
```

```
FddiResourceId ::= INTEGER (0..65535)
```

```
-- Esse tipo de dado é utilizado para referenciar uma instância MAC, PORT ou PATH
```

```
Resource ID
```

```
FddiSMTStationType ::= OCTET STRING (SIZE(8))
```

```
-- É o único identificador de uma estação FDDI. Está definido na forma IEEE (ANSI/IEEE P802.1A), notação canônica para 48 bits onde o primeiro octeto contém os primeiros 8 bits de um endereço, com o bit do endereço I/G (Individual/Group) sendo o menos significativo e o U/L (Universal/Local) sendo o próximo mais significativo e assim sucessivamente.
```

```
FddiMACLongAddressType ::= OCTET STRING (SIZE(6))
```

```
-- A representação de um endereço MAC está também na notação canônica de 48 bits.
```

```
-- A FDDI MIB é composta de 5 grupos relacionados a seguir. De cada um destes grupos, foram selecionados alguns objetos de interesse para representarem recursos do projeto em questão :
```

```
fddimibSMT      OBJECT IDENTIFIER ::= {fddimib1}
```

```
fddimibMAC      OBJECT IDENTIFIER ::= {fddimib2}
```

```
fddimibMACCounters OBJECT IDENTIFIER ::= {fddimib3}
```

```
fddimibPATH     OBJECT IDENTIFIER ::= {fddimib4}
```

**fddimibPORT** OBJECT IDENTIFIER ::= {fddimib5}

**-- Objetos selecionados do grupo SMT :**

**fddimibSMTNumber** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Identifica o número de implementações SMT"

::= {fddimibSMT 1}

**fddimibSMTIndex** OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Valor único identificador de cada SMT"

::= {fddimibSMTEntry 1}

**fddimibSMTStationId** OBJECT-TYPE

SYNTAX FddiStationIdType -- OCTET STRING (SIZE(8))

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Usado para identificar unicamente uma estação FDDI"

REFERENCE "ANSI {fddiSMT 11}"

::= {fddimibSMTEntry 2}

**fddimibSMTUserData** OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (32))

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Essa variável contém 32 octetos para definição de informações do usuário"

REFERENCE "ANSI {fddiSMT 17}"

::= { fddiSMTEntry 6}

**fddimibSMTMACCs** OBJECT-TYPE

SYNTAX INTEGER (0..255)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O número de MACs em uma determinada estação ou concentrador".

REFERENCE "ANSI {fddiSMT 21}"

::= {fddimibSMTEnterY 8}

**fddimibSMTAvailablePaths** OBJECT-TYPE

SYNTAX INTEGER (0..7)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Um valor que indica o tipo de PATH disponível em uma determinada estação. Esse valor começa com o valor zero, depois para cada tipo de path soma-se o valor 2 elevado à potência segundo a seguinte tabela : (1) Primário - 0; (2) Secundário - 1 (3) Local - 2."

REFERENCE "ANSI {fddiSMT 24}"

::= {fddimibSMTEnterY 11}

**fddimibSMTTNotify** OBJECT-TYPE

SYNTAX INTEGER (2..30)

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Um timer, expresso em segundos, utilizado no protocolo de notificação da estação vizinha. Varia de 0 a 30 segundos. O valor default é 30 segundos.

REFERENCE "ANSI {fddiSMT29}"

::= {fddimibSMTEnterY 15}

**fddimibSMTStatRptPolicy** OBJECT-TYPE

SYNTAX INTEGER { true(1), false(2)}

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Se verdadeiro, indica que o nodo gerará um frame de relatório de Status de algum evento ou condição"

REFERENCE "ANSI {fddiSMT 30}"

::= {fddimibSMTEEntry 16}

**fddimibSMTBypassPresent** OBJECT-TYPE

SYNTAX INTEGER { true (1), false (2)}

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Um flag que indica se a estação possui um bypass na porta AB"

REFERENCE "ANSI {fddiSMT 34}

::= {fddiSMTEEntry 18}

**fddimibSMTCFState** OBJECT-TYPE

SYNTAXE INTEGER {

cf0 (1), -- isolated

cf1 (2), -- local\_a

cf2 (3), -- local\_b

cf3 (4), -- local\_ab

cf4 (5), -- local\_s

cf5 (6), -- wrap\_a

cf6 (7), -- wrap\_b

cf7 (8), -- wrap\_ab

cf8 (9), -- wrap\_s

cf9 (10), -- c\_wrap\_a

cf10 (11), -- c\_wrap\_b

cf11 (12), -- c\_wrap\_s

cf12 (13), -- thru

}

ACCESS read-only

STATUS obrigatório

DESCRIPTION "A configuração de conexão da estação ou concentrador"

REFERENCE "ANSI {fddiSMT 42}"

::= {fddimibSMTEEntry 20}

**fddimibSMTRemoteDisconnectFlag** OBJECT-TYPE

SYNTAX INTEGER { true (1), false (2)}

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Um flag que indica que a estação foi remotamente desconectada da rede por receber um fddiSMTAction disconnect"

REFERENCE "ANSI {fddiSMT 44}"

::= {fddimibSMTEntry 21}

**fddimibSMTStationStatus** OBJECT-TYPE

SYNTAX INTEGER {concatenated (1), separated (2), thru (3)}

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O estado corrente do anel primário e secundário com relação a determinada estação"

REFERENCE "ANSI {fddiSMT 45}"

::= {fddimibSMTEntry 22}

**fddimibSMTTimeStamp** OBJECT-TYPE

SYNTAX FddiTimeMilli

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Esta variável assume o valor do TimeStamp"

REFERENCE "ANSI {fddiSMT 51}"

::= {fddimibSMTEntry 24}

**fddimibSMTTransitionTimeStamp** OBJECT-TYPE

SYNTAX FddiTimeMilli

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Essa variável contém o valor do TransitionTimeStamp"

REFERENCE "ANSI {fddiSMT 52}"

::= {fddimibSMTEntry 25}

**-- Objetos seleccionados do grupo MAC :**

**fddimibMACNumber** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS obrigatório

**fddimibMACTMaxCapability** OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Indica o valor máximo do fddiMACTvxValue que um determinado MAC suporta"

REFERENCE "ANSI {fddiMAC 14}"

::= {fddimibMACEntry 6}

**fddi mibMACCurrentPath** OBJECT-TYPE

SYNTAX INTEGER { isolated (1), local (2), secondary (3), primary (4), concatenated (5), thru (6)}

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Indica que PATH um determinado MAC está correntemente inserido"

REFERENCE "ANSI {fddiMAC 23}"

::= {fddimibMACEntry 8}

**fddimibMACSMTAddress** OBJECT-TYPE

SYNTAX FddiMACLongAddressType -- OCTET STRING (SIZE (6))

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Endereço individual de 48-bits do MAC utilizado pelos SMT frames"

REFERENCE "ANSI {fddiMAC 41}"

::= {fddimibMACEntry 16}

**fddimibMACTReq** OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-only

STATUS obrigatório  
 DESCRIPTION "Essa variável é o T\_Req\_value passado para o MAC"  
 REFERENCE "ANSI {fddiMAC 51}"  
 ::= {fddimibMACEntry 17}

**fddimibMACSMTAddress** OBJECT-TYPE  
 SYNTAX FddiMACLongAddressType -- OCTET STRING (SIZE (6))  
 ACCESS read-only  
 STATUS obrigatório  
 DESCRIPTION "É um endereço individual de 48-bit do MAC, utilizado pelos frames SMT"  
 REFERENCE "ANSI {fddiMAC 41}"  
 ::= {fddimibMACEntry 16}

**fddimibMACTNeg** OBJECT-TYPE  
 SYNTAX FddiTimeNano  
 ACCESS read-only  
 STATUS obrigatório  
 DESCRIPTION "É reportado como um número FddiTimeNano"  
 REFERENCE "ANSI {fddiMAC 52}"  
 ::= {fddimibMACEntry 18}

**fddimibMACTMax** OBJECT-TYPE  
 SYNTAX FddiTimeNano  
 ACCESS read-only  
 STATUS obrigatório  
 DESCRIPTION "É a variável T\_Max\_Value passada ao MAC"  
 REFERENCE "ANSI {fddiMAC 53}"  
 ::= {fddimibMACEntry 19}

**fddimibMACTvxValue** OBJECT-TYPE  
 SYNTAX FddiTimeNano  
 ACCESS read-only  
 STATUS obrigatório  
 DESCRIPTION "É a variável TVX\_Value passada ao MAC"

```

REFERENCE "ANSI {fddiMAC 54}
 ::= {fddimibMACEntry 20}
fddimibMACFrameCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS obrigatório
    DESCRIPTION "Um contador do número de frames recebidos pelo MAC"
    REFERENCE "ANSI {fddiMAC 71}"
 ::= {fddimibMACEntry 21}
fddimibMACCopiedCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS obrigatório
    DESCRIPTION "Um contador com valor aproximado do número de frames
endereçados para uma estação e copiado com sucesso para seu buffer"
    REFERENCE "ANSI {fddiMAC 72}"
 ::= {fddimibMACEntry 22}
fddimibMACTransmitCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS obrigatório
    DESCRIPTION "Uma contador com valor aproximado do número de frames
transmitidos pelo MAC"
    REFERENCE "ANSI {fddiMAC 73}"
 ::= {fddimibMACEntry 23}
fddimibMACErrorCts OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS obrigatório
    DESCRIPTION "Um contador do número de frames que foi detectado erro pelo
MAC não foi detectado erro por outro MAC"

```

REFERENCE "ANSI {fddiMAC 81}"  
 ::= {fddimibMACEntry 81}

**fddimibMACLostCts** OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Um contador do número de vezes que o MAC detectou erro de formato durante a recepção do frame e que o frame foi descartado"

REFERENCE "ANSI {fddiMAC 82}"  
 ::= {fddimibMACEntry 25}

**fddimibMACFrameErrorRatio** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Essa variável é o valor de :  $((\Delta \text{fddiMACLostCts} + \Delta \text{fddiMACErrorCts}) / (\Delta \text{fddiMACFrameCts} + \Delta \text{fddiMACLostCts})) * 2^{**} 16$ "

REFERENCE "ANSI {fddiMAC 96}"  
 ::= {fddimibMACEntry 27}

**fddimibMACFrameErrorFlag** OBJECT-TYPE

SYNTAX INTEGER { true (1), false (2) }

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Indica que a condição de erro do frame MAC está presente"

REFERENCE "ANSI {fddiMAC 114}"  
 ::= {fddimibMACEntry 31}

**fddimibMACFrameErrorThreshold** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Uma medida que determina quando deve der gerado um relatório sobre as condições do MAC"

REFERENCE "ANSI {fddiMAC 95}"

::= {fddimibMACEntry 26}

**-- Objetos selecionados do grupo MAC Counters**

**fddimibMACTokenCts**                    OBJECT-TYPE

SYNTAX     Counter

ACCESS     read-only

STATUS     obrigatório

DESCRIPTION "Um contador que é uma aproximação do número de vezes que uma estação recebeu um token (restrito ou não restrito) no MAC. Este contador é utilizado para determinar congestionamento na rede"

REFERENCE   "ANSI {fddiMAC 74}"

::= {fddimibMACCountersEntry 1}

**fddimibMACTvxExpiredCts**            OBJECT-TYPE

SYNTAX     Counter

ACCESS     read-only

STATUS     obrigatório

DESCRIPTION "Um contador que é uma aproximação do número de vezes que o Tvx expirou"

REFERENCE   "ANSI {fddiMAC 83}"

::= {fddimibMACCountersEntry 2}

**fddimibMACNotCopiedCts**            OBJECT-TYPE

SYNTAX     Counter

ACCESS     read-only

STATUS     obrigatório

DESCRIPTION "Um contador que é uma aproximação do número de frames que foi endereçado para esse MAC e não foi copiado para o buffer. Isto pode ocorrer por problemas de congestionamento no buffer local"

REFERENCE   "ANSI {fddiMAC 84}"

::= {fddimibMACCountersEntry 3}

**fddimibMACLateCts**                    OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Um contador que é uma aproximação do número de expirações do TRT desde que o MAC foi reinicializado ou um token foi recebido"

REFERENCE "ANSI {fddiMAC 85}"

::= {fddimibMACCountersEntry 4}

**fddimibMACRingOpCts** OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Um contador do número de vezes que o anel passou do estado não operacional para operacional"

REFERENCE "ANSI {fddiMAC 86}"

::= {fddimibMACCountersEntry 5}

**fddimibMACNotCopiedRatio** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Essa variável é o valor de  $(\Delta fddiMACNotCopiedCts / (\Delta fddiMACCopiedCts + \Delta fddiMACNotCopiedCts)) * 2^{16}$ "

REFERENCE "ANSI {fddiMAC 105}"

::= {fddimibMACCountersEntry 6}

**fddimibMACNotCopiedThreshold** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Um limite que determina quando um relatório da condição do MAC deve ser gerado"

REFERENCE "ANSI {fddiMAC 103}"

::= {fddimibMACCountersEntry 8}

**-- Objetos selecionados do grupo PATH****fddimibPATHNumber** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O total de PATHs possíveis em uma determinada entidade de gerenciamento."

::= {fddimibPATH 1}

**fddimibPATHSMTIndex** OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O valor do índice SMT associado a esse PATH"

::= {fddimibPATHEntry 1}

**fddimibPATHIndex** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Variável que identifica unicamente o PATH primário, secundário e local"

REFERENCE "ANSI {fddiPATH 11}"

::= {fddimibPATHEntry 2}

**fddimibPATHTVXLowerBound** OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Especifica o valor mínimo do fddiMACTvxValue que deve ser utilizado para cada MAC que está configurado em um determinado PATH. O valor inicial dessa variável é 2500 nsec (2.5 ms). Essa variável deve ser :  $0 < \text{fddimibPATHTVXLowerBound} < \text{fddimibPATHMaxTReq}$  e  $\text{fddimibPATHTVXLowerBound} \leq \text{fddimibMACTVXCapability}$ ."

REFERENCE "ANSI {fddiPATH 21}"

::= {fddimibPATHEntry 3}

**fddimibPATHMaxLowerBound** OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Especifica o valor mínimo do fddiMACTMax que deve ser utilizado para cada MAC que está configurado em um determinado PATH. O valor inicial dessa variável é 165000 nseg (165 ms). Essa variável deve ser : 10000 nseg (10 mseg) <= fddimibPATHMaxLowerBound < fddimibMACTMaxCapability"

REFERENCE "ANSI {fddiPATH 22}"

::= {fddimibPATHEntry 4}

**fddimibPATHMaxTReq** OBJECT-TYPE

SYNTAX FddiTimeNano

ACCESS read-write

STATUS obrigatório

DESCRIPTION "Especifica o valor máximo para o fddiMACT-Req que dever ser utilizado para cada MAC que está configurado em um determinado PATH. O valor default é 165000 (165 ms). Essa variável deve ser : fddimibPATHTVXLowerBound < fddimibPATHMaxTReq <= fddimibPATHMaxLowerBound"

REFERENCE "ANSI {fddiPATH 23}"

::= {fddimibPATHEntry 5}

**fddimibPATHConfigTokenOrder** OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Um objeto que associa a ordem do token. Se o token passa pelos recursos a,b,c e d nessa ordem, o valor desse objeto para esses recursos é 1,2,3 e 4 respectivamente"

::= {fddimibPATHConfigEntry 3}

**fddimibPATHConfigResourceIndex** OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O valor do índice do recurso SMT utilizado para referenciar uma instância de um determinado MAC ou PORT"

::= {fddimibPATHConfigEntry 5}

**fddimibPATHConfigCurrentPath** OBJECT-TYPE

SYNTAX INTEGER { (1) isolado, (2) local, (3)secundário, (4) primário, concatenado (5), thru (6) }

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O status de inserção corrente no PATH de um determinado recurso"

::= {fddimibPATHConfigEntry 6}

## -- Objetos selecionados do grupo PORT

**fddimibPORTNumber** OBJECT-TYPE

SYNTAX INTEGER (0..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O número total de portas em uma entidade de aplicação de gerenciamento de rede"

::= {fddimibPORT 1}

**fddimibPORTSMTIndex** OBJECT-TYPE

SYNTAX INTEGER (1..65535)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O valor do índice do SMT associado a uma determinada porta"

::= {fddimibPORTEEntry 1}

**fddimibPORTMyType** OBJECT-TYPE

SYNTAX INTEGER { (1) a, (2) b, (3) s, (4) m, (5) nenhum }

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O valor do PC\_TYPE PORT"

REFERENCE "ANSI {fddiPORT 12}"

::= {fddimibPORTEntree 3}

**fddimibPORTCurrentPath** OBJECT-TYPE

SYNTAX INTEGER { ce0 (1) -- isolado, ce1 (2) -- local, ce2 (3) -- secundário, ce3 (4) -- primário, ce4 (5) concatenado, ce5 (6) thru. }

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Indica o PATH em que uma determinada porta está correntemente inserida"

REFERENCE "ANSI {fddiPORT 16}"

::= {fddimibPORTEntree 7}

**fddimibPORTAvailablePaths** OBJECT-TYPE

SYNTAX INTEGER (0..7)

ACCESS read-only

STATUS obrigatório

DESCRIPTION "Indica o PATH que está disponível para uma determinada PORT. Inicialmente esse valor é zero. Para cada tipo de PATH que uma determinada porta está disponível é somado um valor, de acordo com a tabela a ser mostrada, elevado à potência de dois. A tabela é a seguinte : (0) primário, (1) secundário, (2) local"

REFERENCE "ANSI {fddiPORT 19}"

::= {fddimibPORTEntree 10}

**fddimibPORTPMDCClass** OBJECT-TYPE

SYNTAX INTEGER { (1) multimodo, (2) modo simples 1, (3) modo simples 2, (4) sonet, (5) fibra ótica de baixo custo, (6) par trançado, (7) desconhecido, (8) não especificado }

ACCESS read-only

STATUS obrigatório  
 DESCRIPTION "Indica o tipo da entidade PMD associada com uma determinada porta"  
 REFERENCE "ANSI {fddiPORT 22}"  
 ::= {fddimibPORTEntry 11}

**fddimibPORTLerEstimate** OBJECT-TYPE  
 SYNTAX INTEGER (4..15)  
 ACCESS read-only  
 STATUS obrigatório  
 DESCRIPTION "Média da taxa de erro de conexão. Essa variável varia de  $10^{-4}$  a  $10^{-15}$  e é reportado como o valor absoluto de um logaritmo na base 10"  
 REFERENCE "ANSI {fddiPORT 51}"  
 ::= {fddimibPORTEntry 15}

**fddimibPORTLemRejectsCts** OBJECT-TYPE  
 SYNTAX Counter  
 ACCESS read-only  
 STATUS obrigatório  
 DESCRIPTION "Contador do número de vezes que uma conexão foi rejeitada"  
 REFERENCE "ANSI {fddiPORT 52}"  
 ::= {fddimibPORTEntry 16}

**fddimibPORTLerCutoff** OBJECT-TYPE  
 SYNTAX INTEGER (4..15)  
 ACCESS read-write  
 STATUS obrigatório  
 DESCRIPTION "Uma estimativa de quando uma conexão será desfeita. Varia de  $10^{-4}$  a  $10^{-15}$  e é reportado em valor absoluto de um logaritmo na base 10 ( valor default 7) "  
 REFERENCE "ANSI {fddiPORT 58}"  
 ::= {fddimibPORTEntry 18}

**fddimibPORTLerAlarm** OBJECT-TYPE

SYNTAX INTEGER (4..15)  
 ACCESS read-write  
 STATUS obrigatório  
 DESCRIPTION "Uma estimativa de quando um alarme será gerado. Varia de  $10^{-4}$  a  $10^{-15}$  e é reportado em valor absoluto de um logaritmo na base 10 ( valor default 8) "

REFERENCE "ANSI {fddiPORT 59}"

::= {fddimibPORTEntry 19}

**fddimibPORTConnectState** OBJECT-TYPE

SYNTAX INTEGER { (1) desabilitada, (2) conectada, (3) standby, (4) ativa}  
 ACCESS read-only  
 STATUS obrigatório

DESCRIPTION "Indica o estado de conexão de uma determinada porta"

REFERENCE "ANSI {fddiPORT 61}"

::= {fddimibPORTEntry 20}

**fddimibPORTLerFlag** OBJECT-TYPE

SYNTAX INTEGER { (1) true, (2) false}  
 ACCESS read-only  
 STATUS obrigatório

DESCRIPTION "Essa condição se torna ativa quando o valor de fddiPORTLerEstimate é menor ou igual a fddiPORTLerAlarm. Será reportado com o Status Report Frames (SRF)"

REFERENCE "ANSI {fddiPORT 64}"

::= {fddimibPORTEntry 23}

**fddimibPORTHardwarePresent** OBJECT-TYPE

SYNTAX INTEGER { (1) true, (2) false}  
 ACCESS read-only  
 STATUS obrigatório

DESCRIPTION "Essa variável indica a presença de um hardware em uma determinada porta"

REFERENCE "ANSI {fddiPORT 65}"

```

 ::= {fddimibPORTEntry 24}
fddimibPORTAction      INTEGER { (1) outro, (2) mantém estado da porta, (3)
habilita porta, (4) desabilita porta, (5) inicializa porta, (6) finaliza porta}
      ACCESS      read-write
      STATUS      obrigatório
      DESCRIPTION  "Causa a geração de um sinal de controle para uma
determinada porta"
      REFERENCE    "ANSI {fddiPORT 70}"
 ::= {fddimibPORTEntry 25}
END

```

Alguns objetos da MIB II SNMP [McCloghrie1991] foram utilizados para gerenciar as interfaces de redes instaladas nas estações multimídia 'Picasso' e do *Hub* LinkbuilderFDDI.

**-- A seguir descreve-se os objetos de interesse para gerenciamento dos recursos do projeto do grupo INTERFACES**

```

ifType                OBJECT-TYPE
      SYNTAX      INTEGER {
          (1) outra,
          (2) regular1822(2),
          (3) hdh1822(3),
          (4) ddn-x25 (5),
          (5) rfc877-x25 (5),
          (6) ethernet-csmacd (6),
          (7) iso88023-csmacd (7),
          (8) iso88024-tokenBus (8),
          (9) iso88025-tokenRing (9),
          (10) iso88026-man,
          (11) starLan,
          (12) proteon-10Mbit,

```

```

(13) proteon-80Mbit,
(14) hyperchannel,
(15) fddi,
(16) lapb,
(17) sdlc,
(18) ds1,          -- T1
(19) e1,          -- Equivalente a T-1 na Europa
(20) basicISDN,
(21) primaryISDN, -- Proprietário serial
(22) PropPointToPointSerial,
(23) ppp,
(24) SoftwareLoopback,
(25) eon,
(26) ethernet-3Mbit,
(27) nsip,        -- XNS sobre IP
(28) slip,        -- SLIP genérico
(29) ultra,       -- Tecnologias Ultra
(30) ds3,         -- T-3
(31) sip,         -- SMDS
(32) frame-relay
    }
ACCESS      read-only
STATUS      obrigatório
DESCRIPTION "Descrição do tipo de interface de acordo com o nível físico"
::= { ifEntry 3 }
ifSpeed          OBJECT-TYPE
SYNTAX      Gauge
ACCESS      read-only
STATUS      obrigatório
DESCRIPTION "Uma estimativa da largura de banda corrente da interface em
bits por segundo"

```

::= {ifEntry 5}

**ifOperStatus** OBJECT-TYPE

SYNTAX INTEGER {

(1) up, -- pronto para passar pacotes

(2) down,

(3) testando, -- em algum modo de teste

}

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O estado operacional corrente da interface. No estado de teste nenhum pacote pode passar"

::= {ifEntry 8}

**ifOutErrors** OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O número de pacotes que não foram transmitidos por ter ocorrido erros"

::= {ifEntry 20}

**ifOutQLen** OBJECT-TYPE

SYNTAX Gauge

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O tamanho da fila de pacotes na saída (em número de pacotes)"

::= {ifEntry 21}

**ifInDiscards** OBJECT-TYPE

SYNTAX Counter

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O número de pacotes que chegaram e foram descartados por falta de espaço no buffer"

::= {ifEntry 13}

**ifINErrors**

**OBJECT-TYPE**

SYNTAX Counter

ACCESS read-only

STATUS obrigatório

DESCRIPTION "O número de pacotes que continham erro e não puderam ser passados para os níveis mais acima"

::= {ifEntry 14}

ANEXO VI

GRÁFICOS GERADOS PELO  
SOFTWARE SunNet Manager

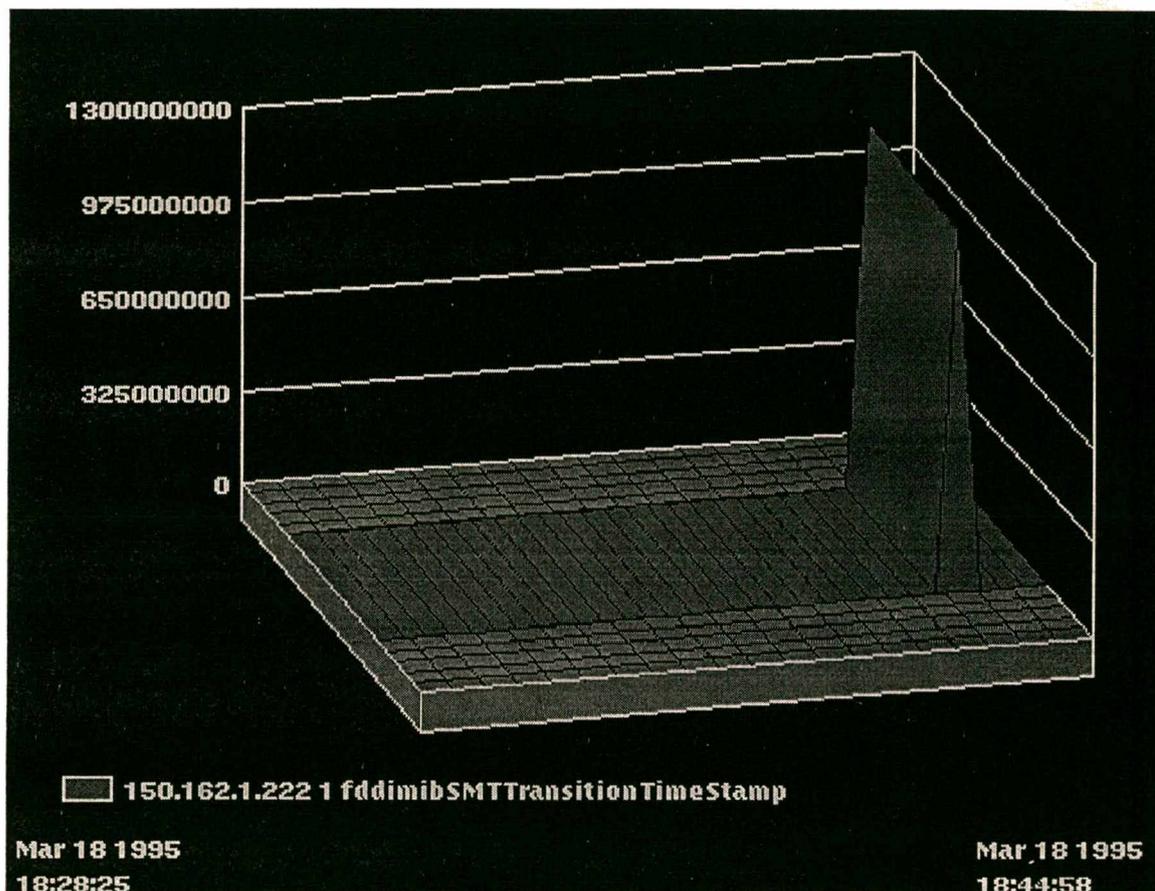


Figura A.1 - Gráfico das diferenças entre tempos de transição do Token na coleta de dados do dia 18/03/95 às 18:28 - Dados coletados de 30 em 30 segundos

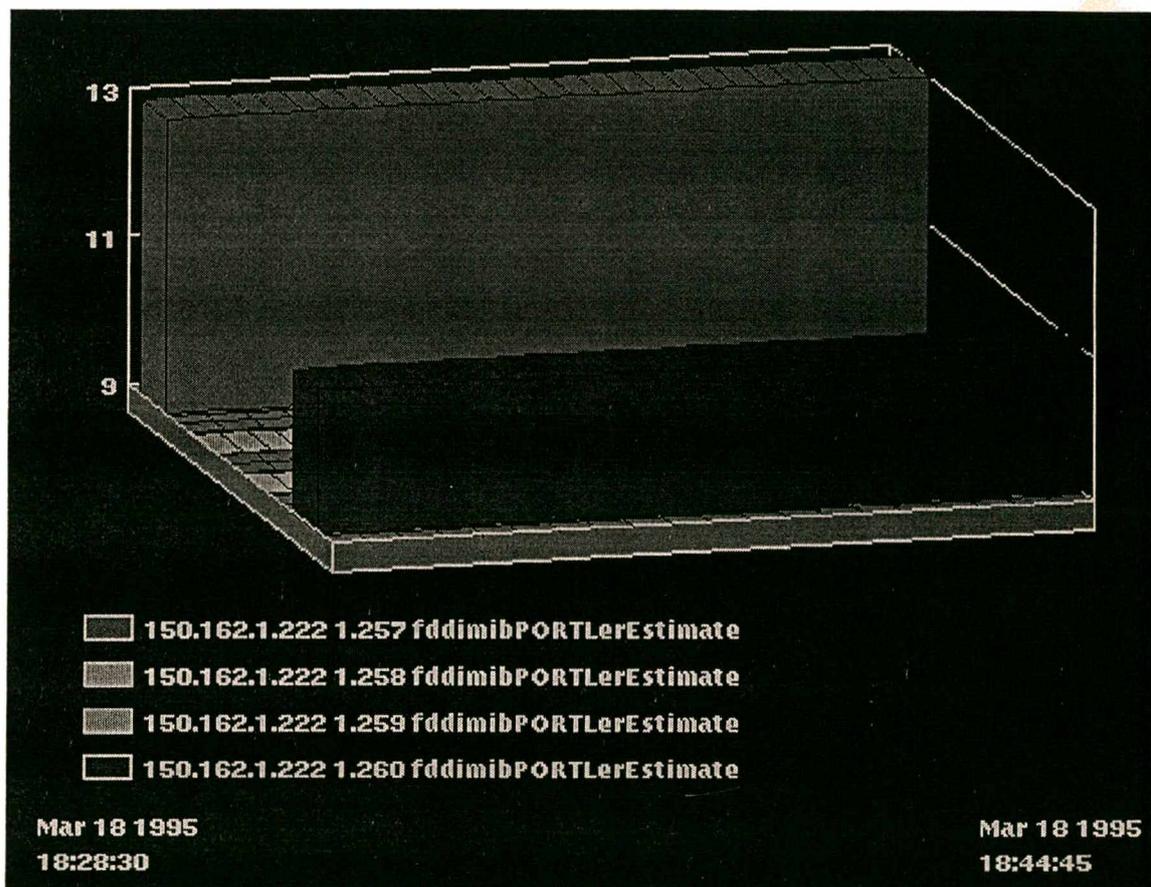


Figura A.2 - Gráfico da taxa de erro de conexão da coleta de dados realizada no dia 18/03/95 às 18:28 - Dados coletados de 30 em 30 segundos

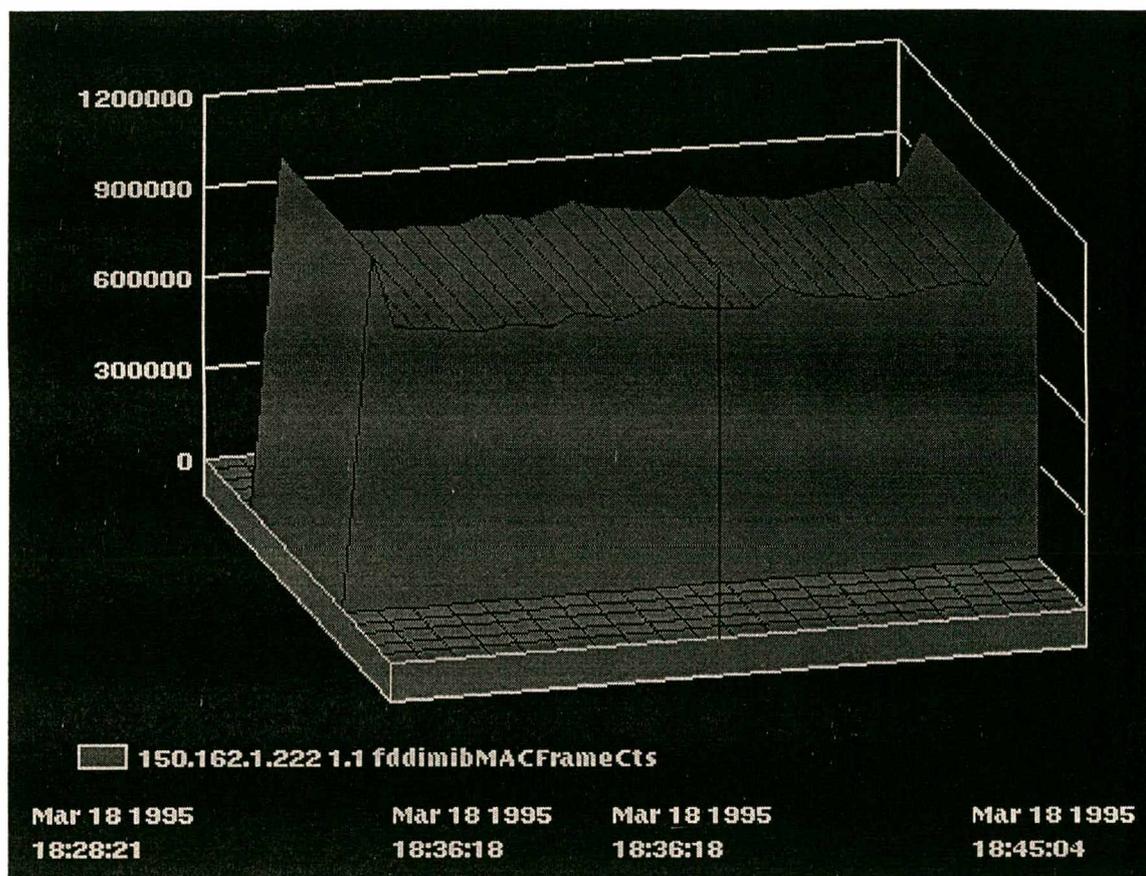


Figura A.3 - Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 18/03/95 às 18:28 - Dados coletados de 30 em 30 segundos

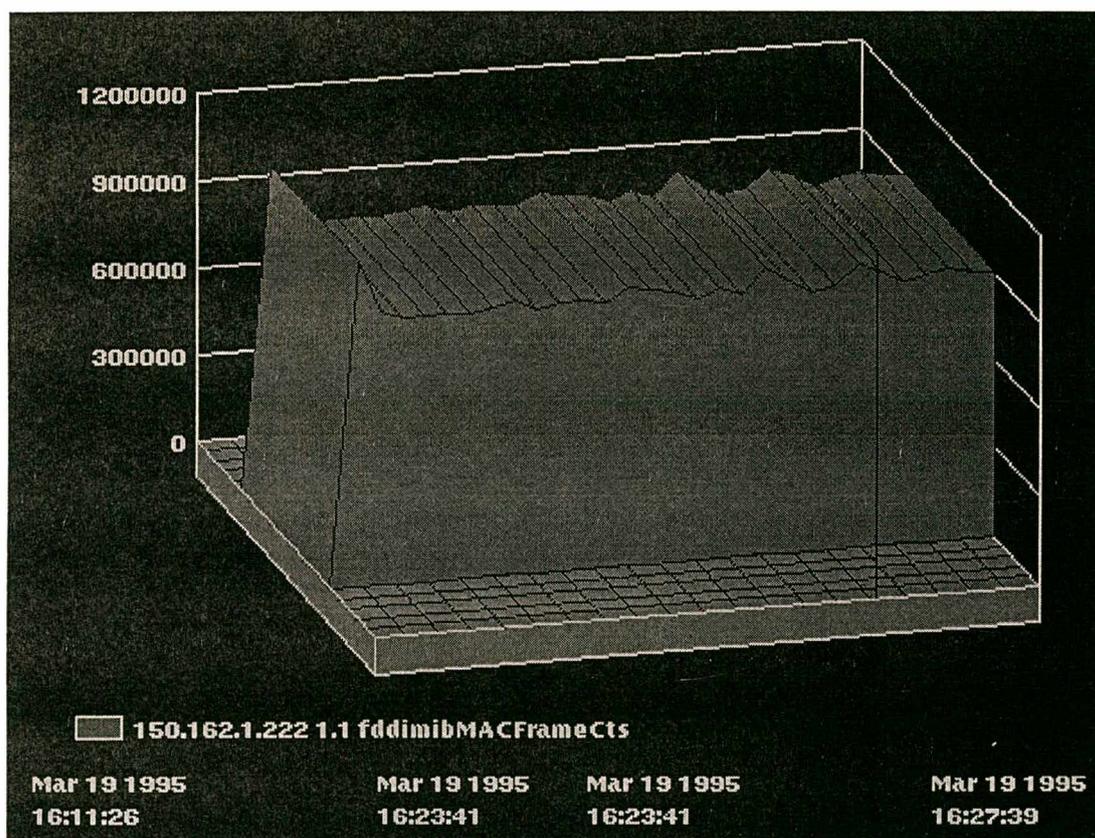


Figura A.4 - Gráfico das diferenças do número de frames transmitidos no teste realizado no dia 19/03/95 às 16:11 - Dados coletados de 30 em 30 segundos

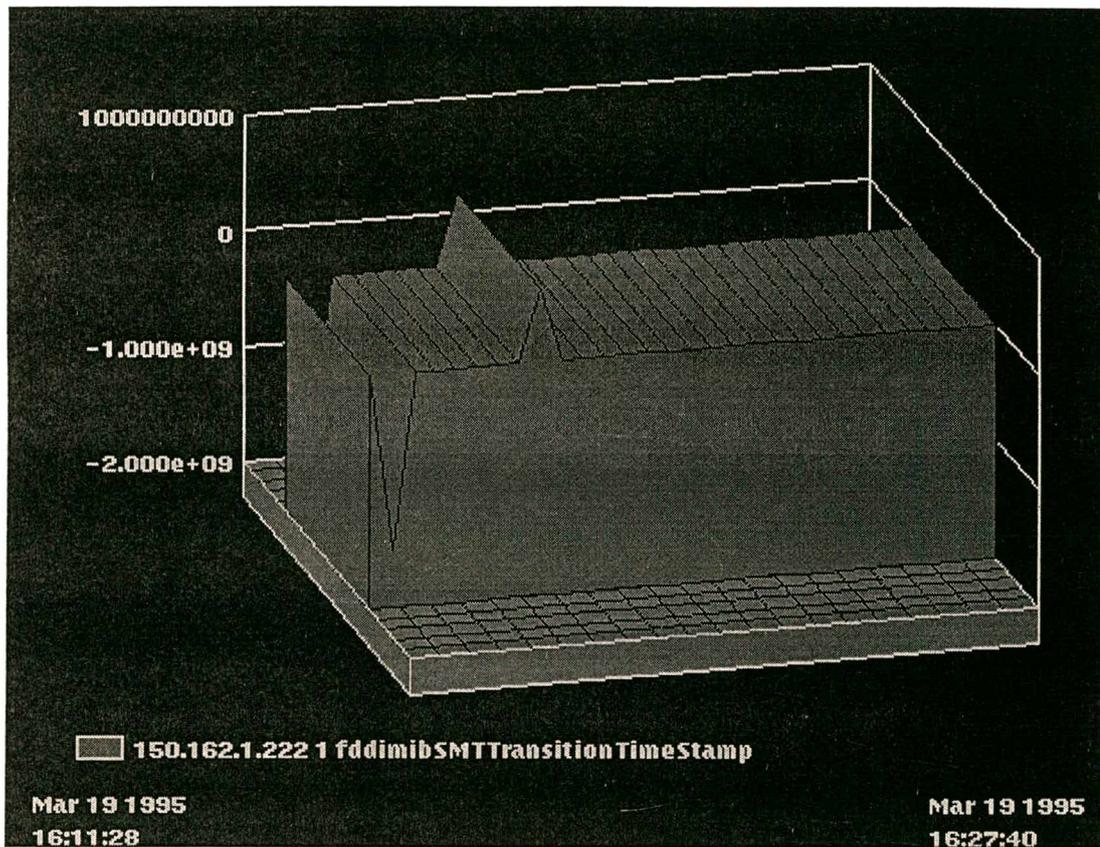


Figura A.5 - Gráfico das diferenças dos tempos de transição do Token na coleta de dados do dia 19/03/95 às 16:11 - Dados coletados de 30 em 30 segundos

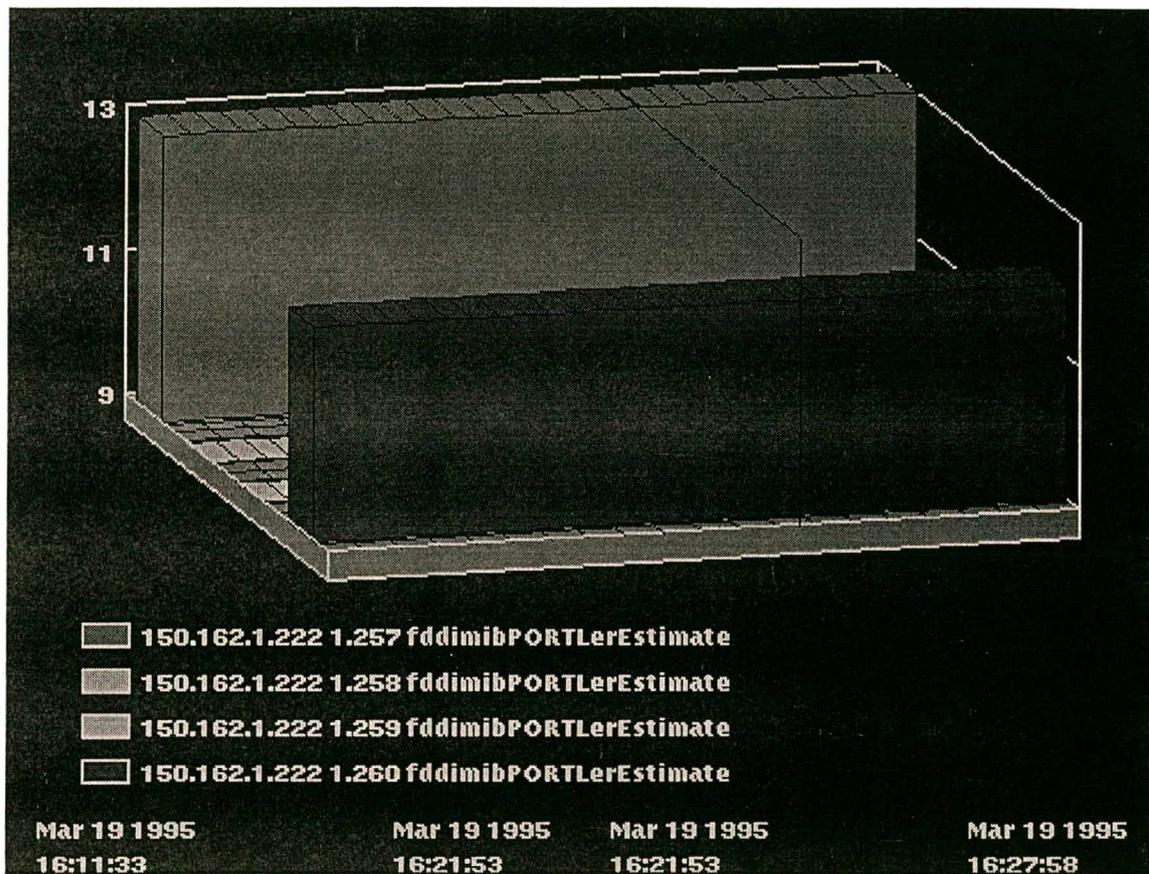


Figura A.6 - Gráfico da taxa de erro de conexão da coleta de dados realizada no dia 19/03/95 às 16:11 - Dados coletados de 30 em 30 segundos

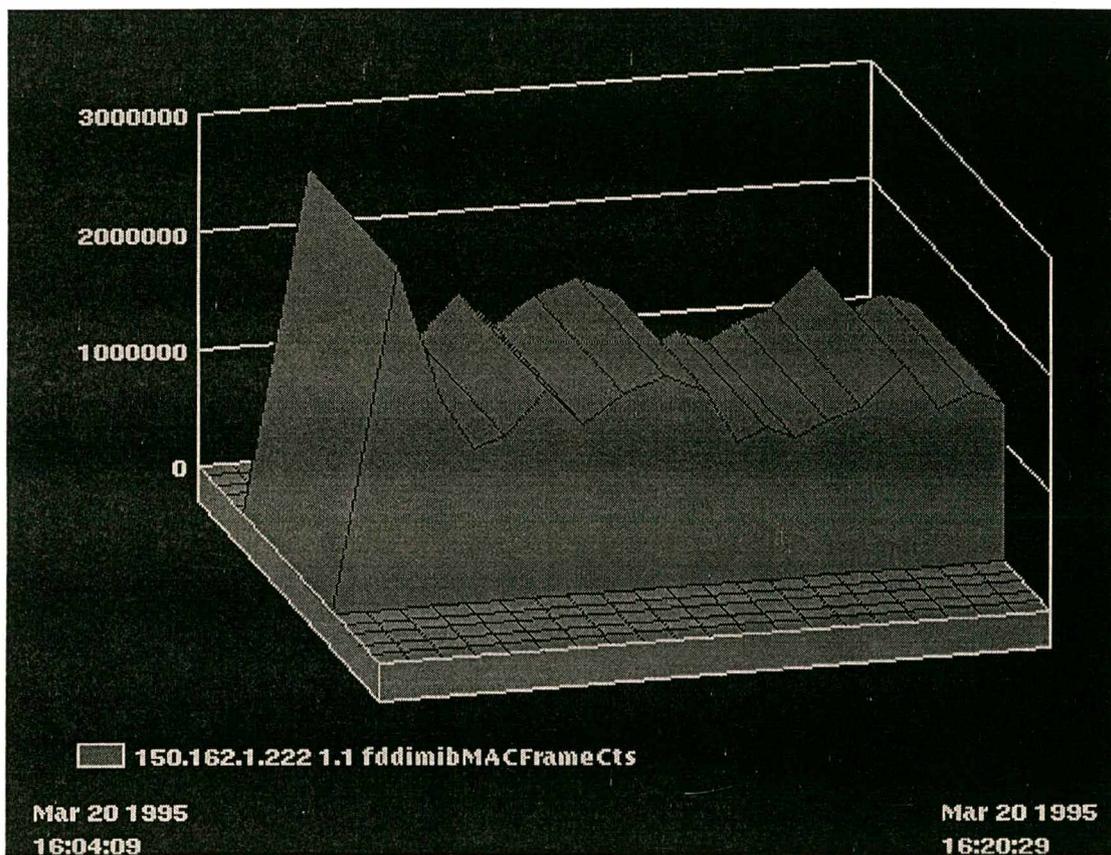


Figura A.7 - Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 20/03/95 às 16:06 - Dados coletados de 30 em 30 segundos

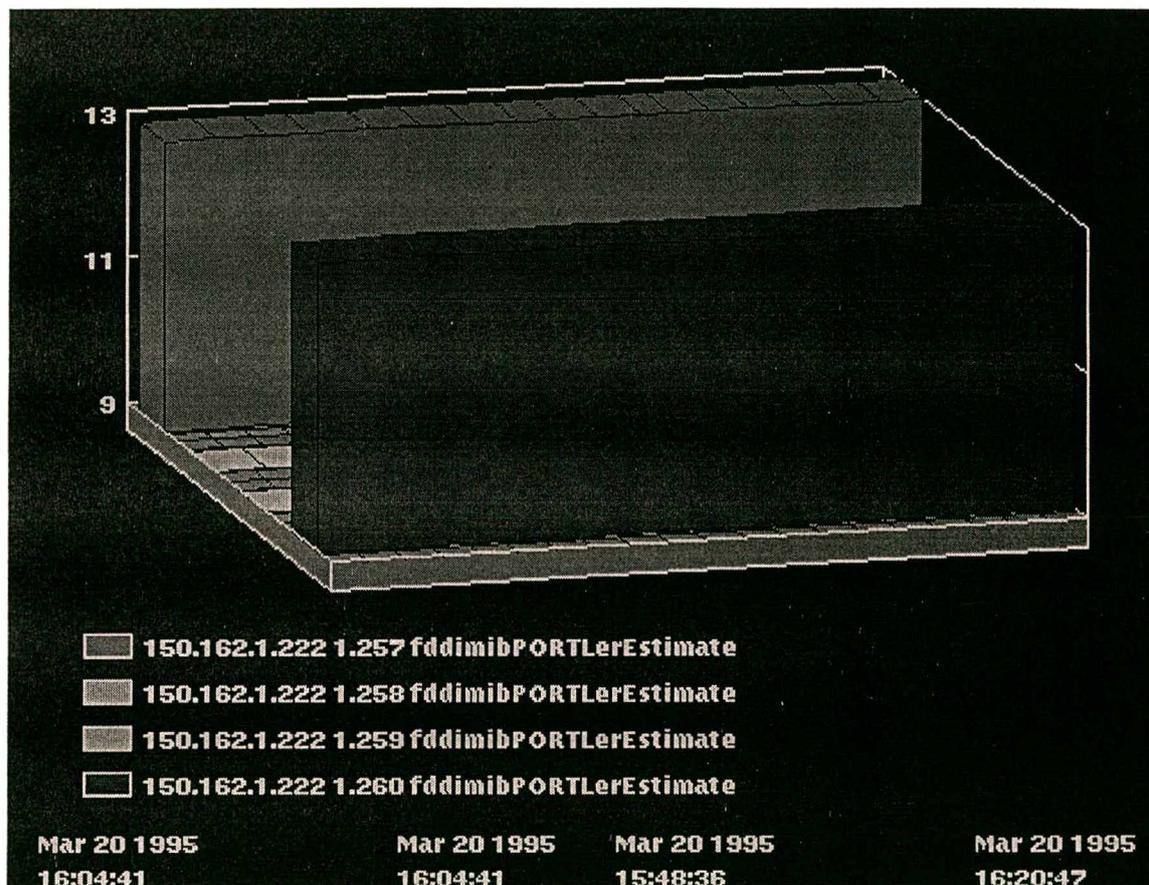


Figura A.8 - Gráfico da taxa de erro de conexão da coleta de dados realizada no dia 20/03/95 às 16:06 - Dados coletados de 30 em 30 segundos

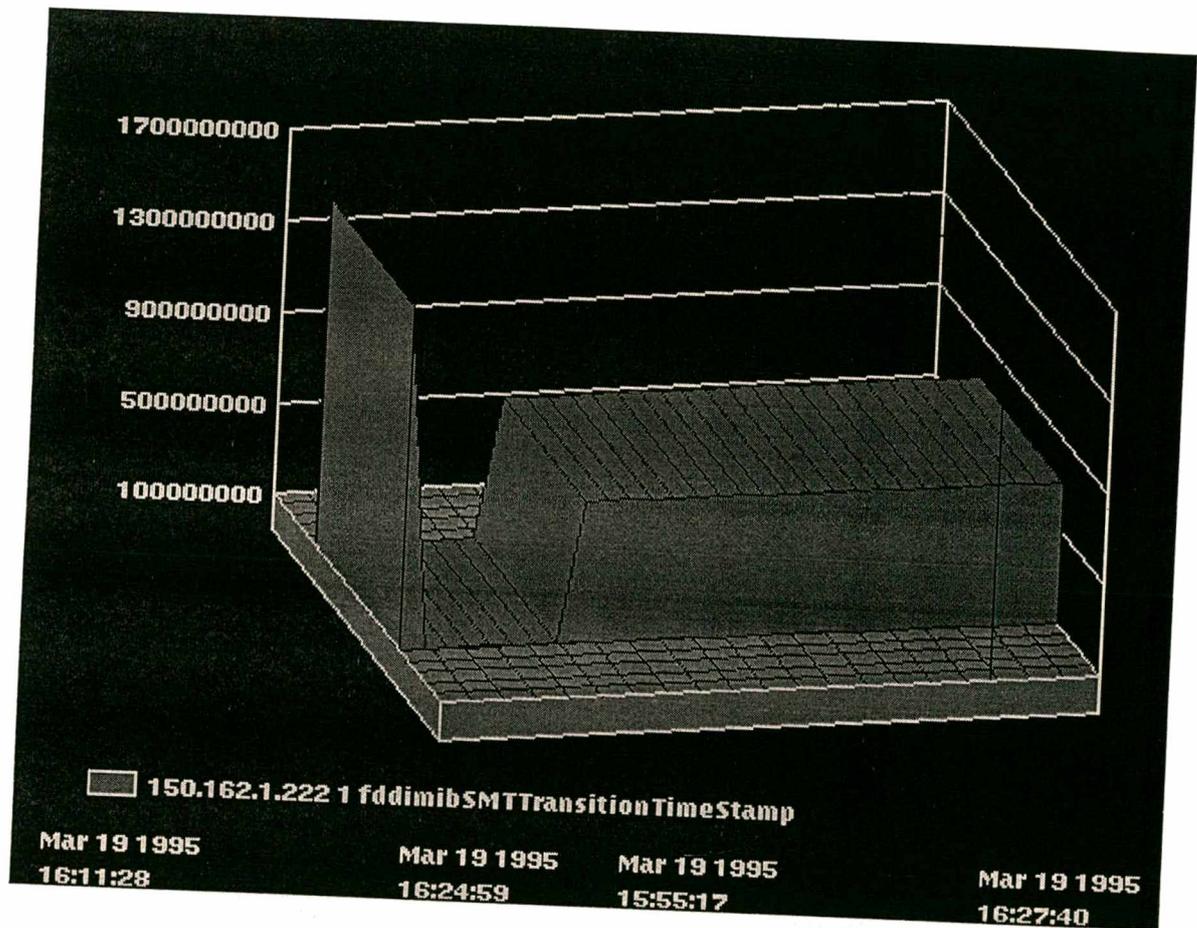


Figura A.9 - Gráfico do tempo de transição do Token na coleta de dados do dia 20/03/95 às 16:06 - Dados coletados de 30 em 30 segundos

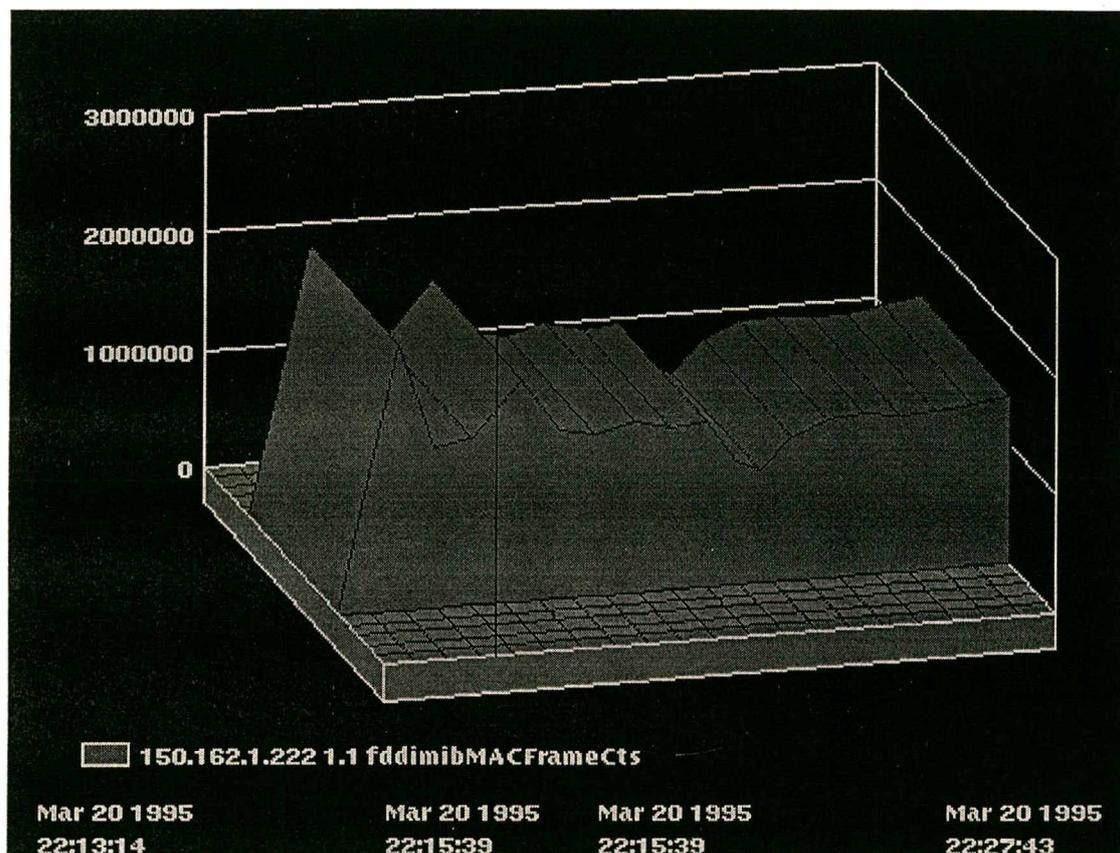


Figura A.10 - Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 20/03/95 às 22:13 - Dados coletados de 30 em 30 segundos

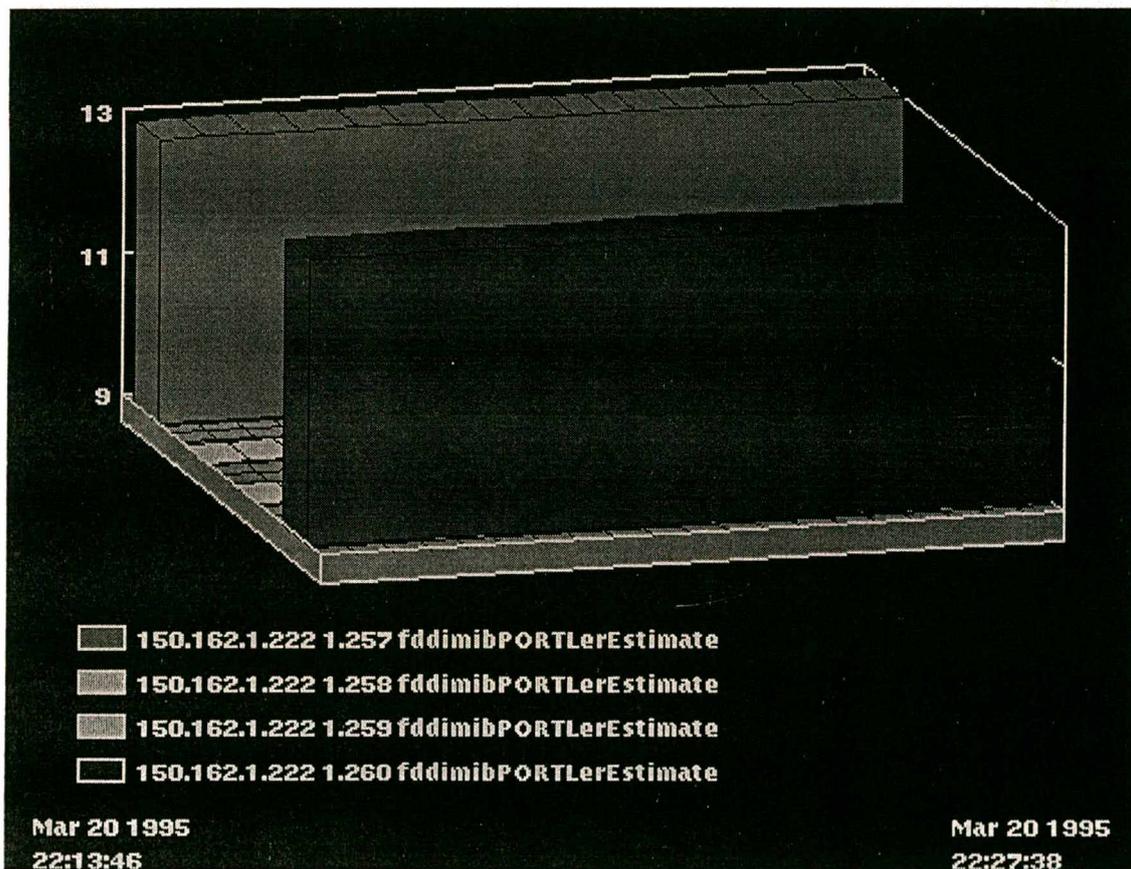


Figura A.11 - Gráfico da taxa de erro de conexão da coleta de dados realizada no dia 20/03/95 às 22:13 - Dados coletados de 30 em 30 segundos

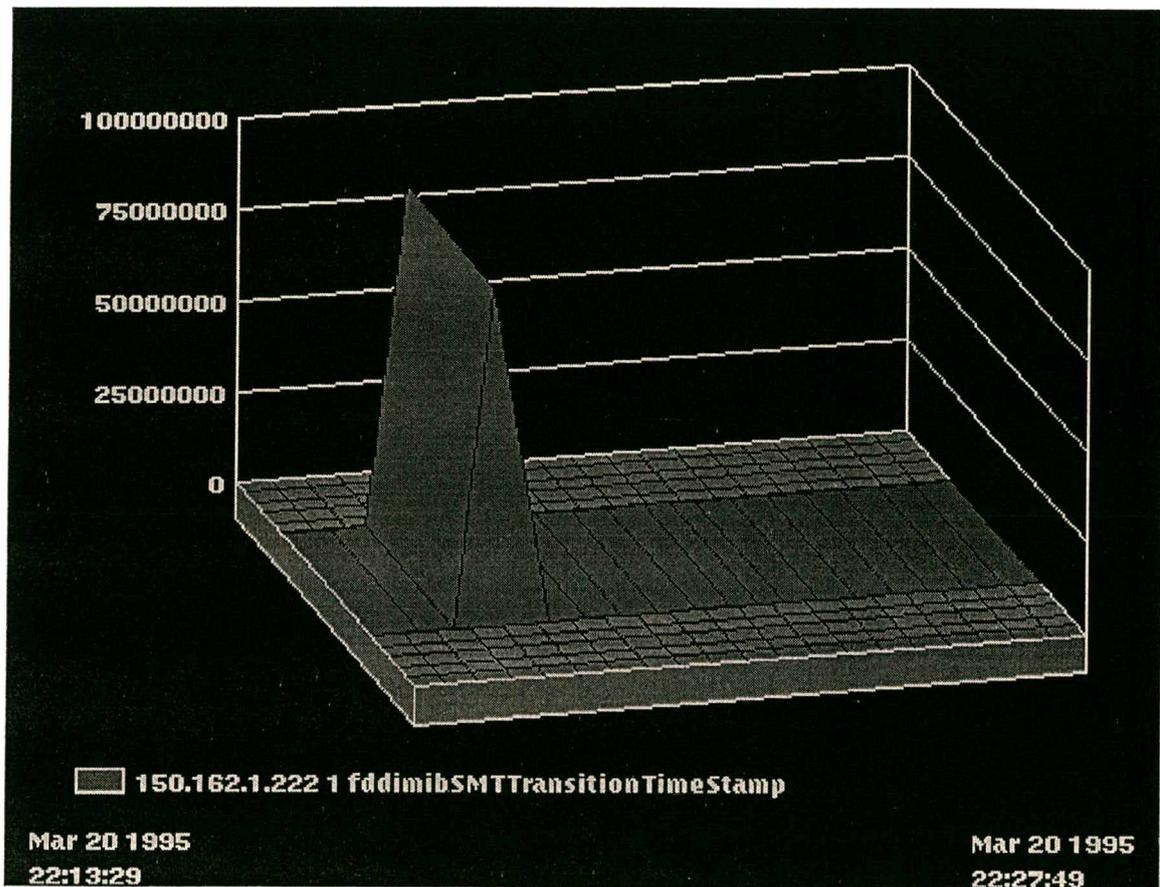


Figura A.12 - Gráfico das diferenças do tempo de transição do Token na coleta de dados do dia 20/03/95 às 22:13 - Dados coletados de 30 em 30 segundos

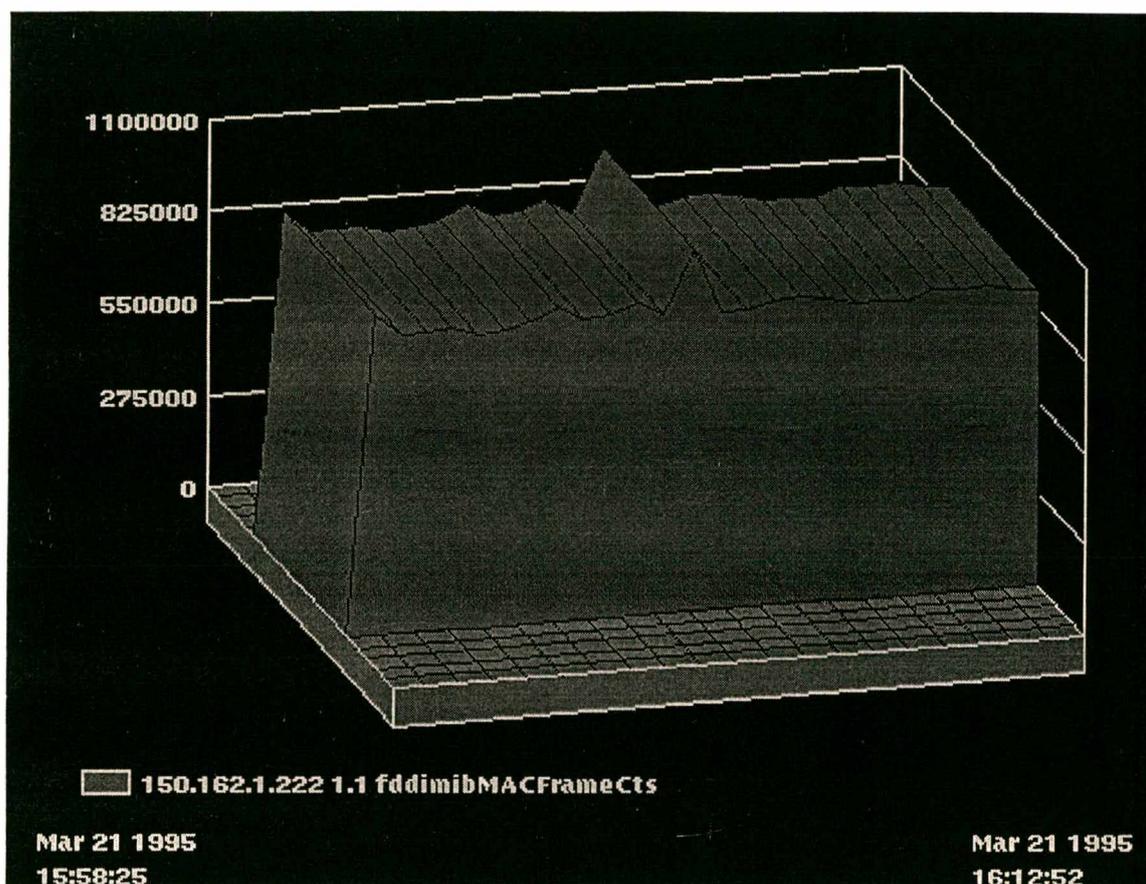


Figura A.13 - Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 21/03/95 às 15:58 - Dados coletados de 30 em 30 segundos

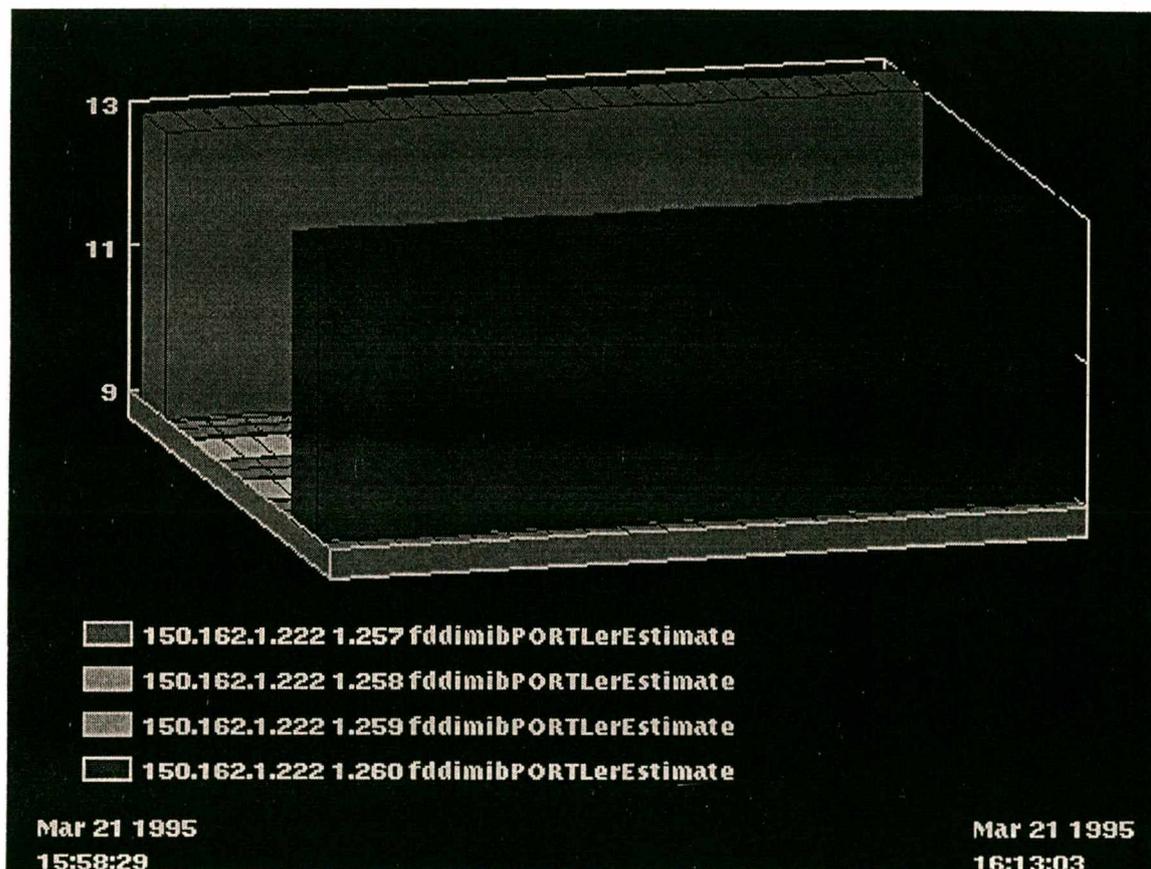


Figura A.14 - Gráfico da taxa de erro de conexão da coleta de dados realizada no dia 21/03/95 às 15:58 - Dados coletados de 30 em 30 segundos

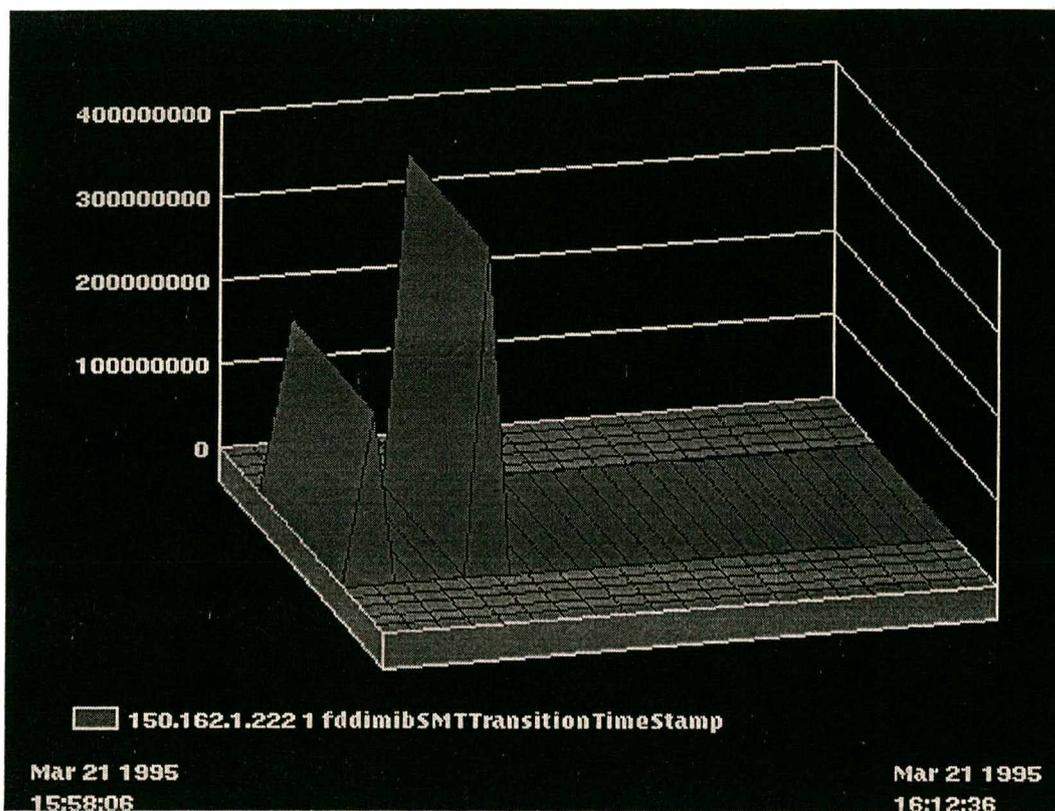


Figura A.15 - Gráfico das diferenças do tempo de transição do Token na coleta de dados do dia 21/03/95 às 15:58 - Dados coletados de 30 em 30 segundos

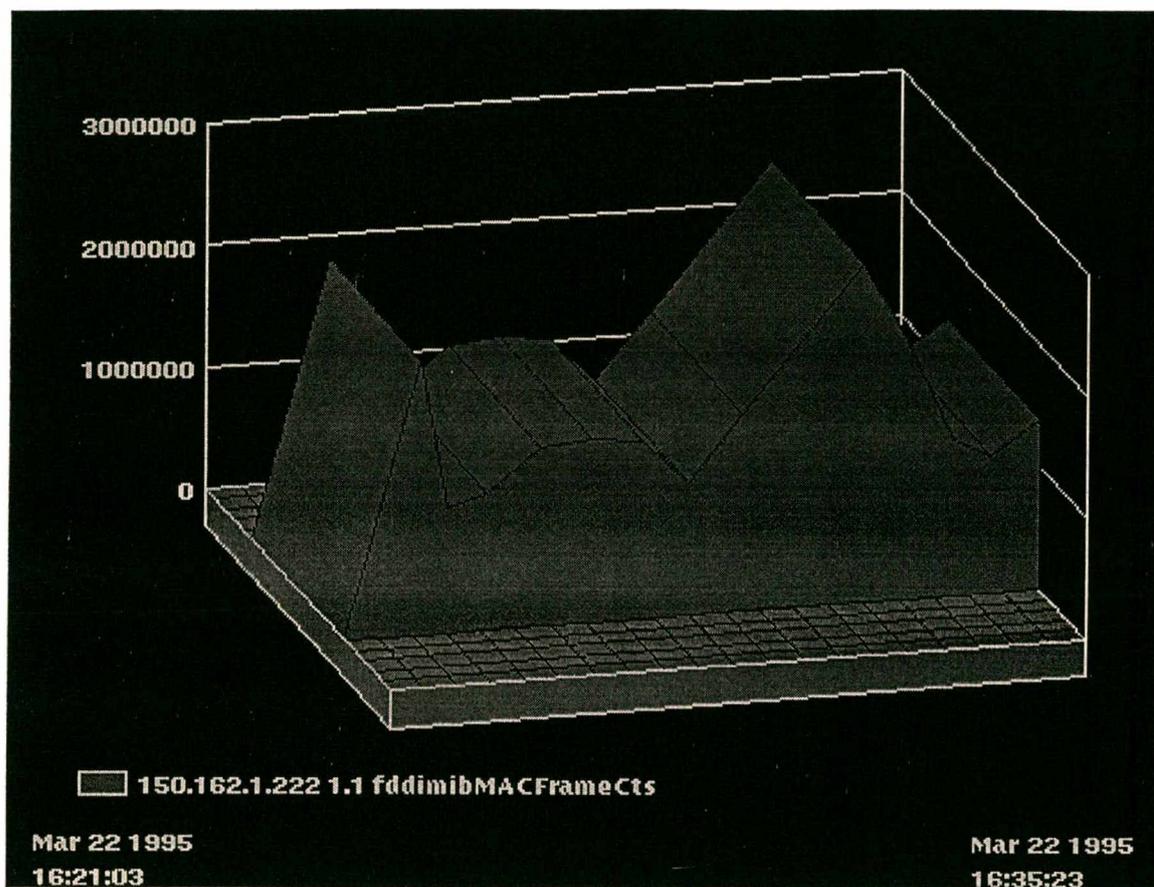


Figura A.16 - Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 22/03/95 às 15:59 - Dados coletados de 30 em 30 segundos

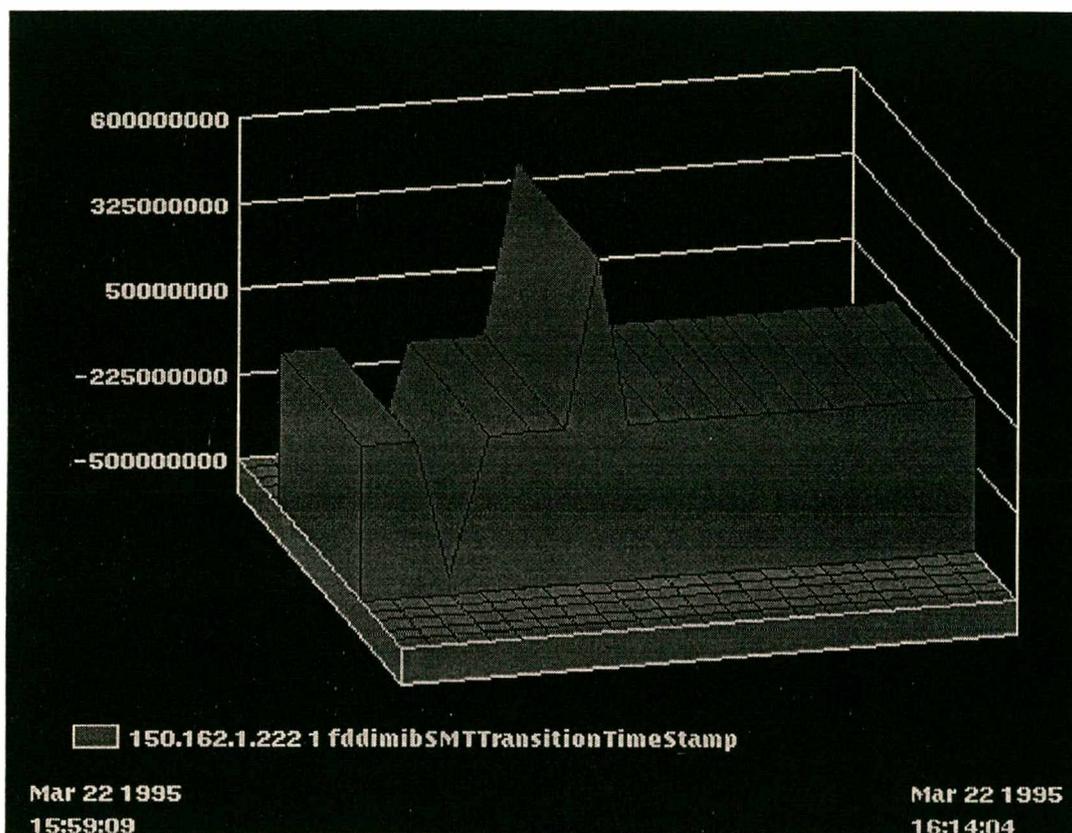


Figura A.18 - Gráfico das diferenças do tempo de transição do Token na coleta de dados do dia 22/03/95 às 15:59 - Dados coletados de 30 em 30 segundos

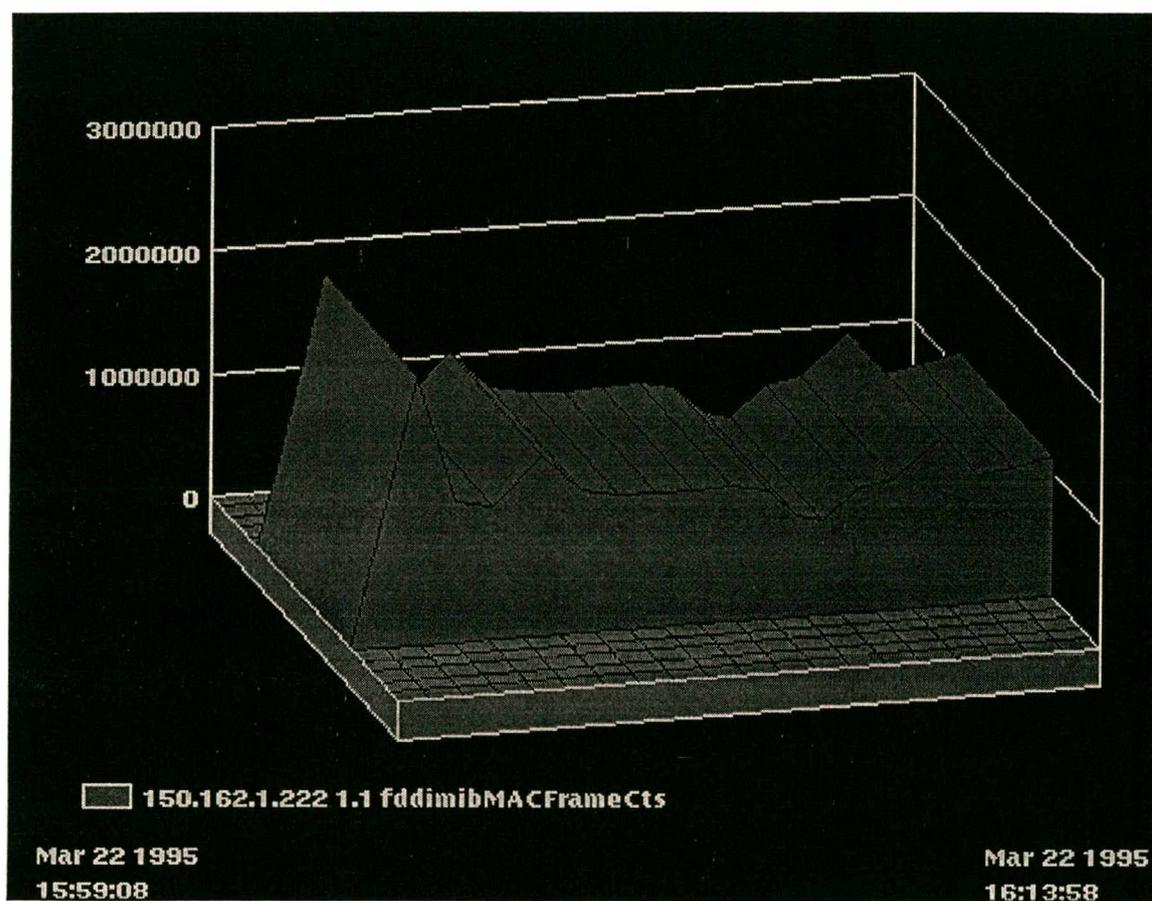


Figura A.19 - Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 22/03/95 às 16:21 - Dados coletados de 30 em 30 segundos

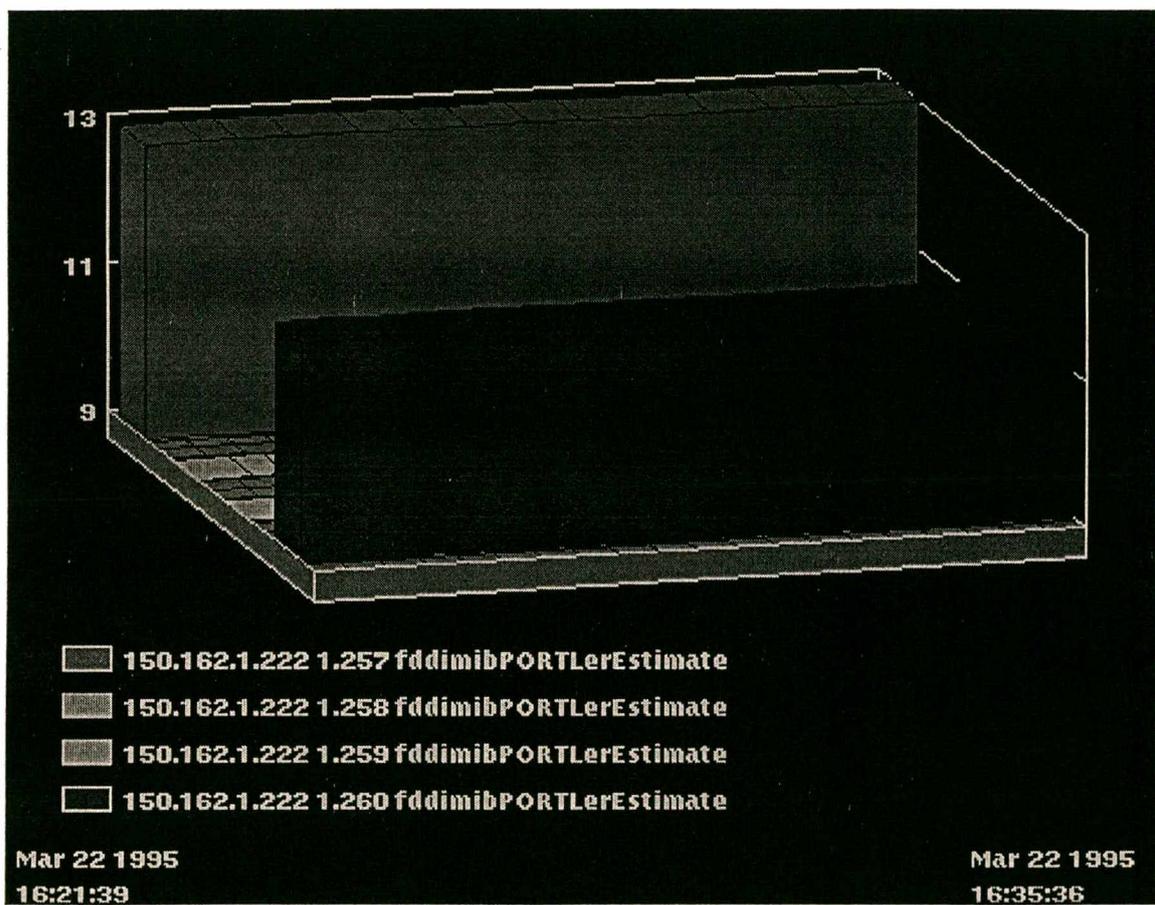


Figura A.20 - Gráfico da taxa de erro de conexão da coleta de dados realizada no dia 22/03/95 às 16:21 - Dados coletados de 30 em 30 segundos

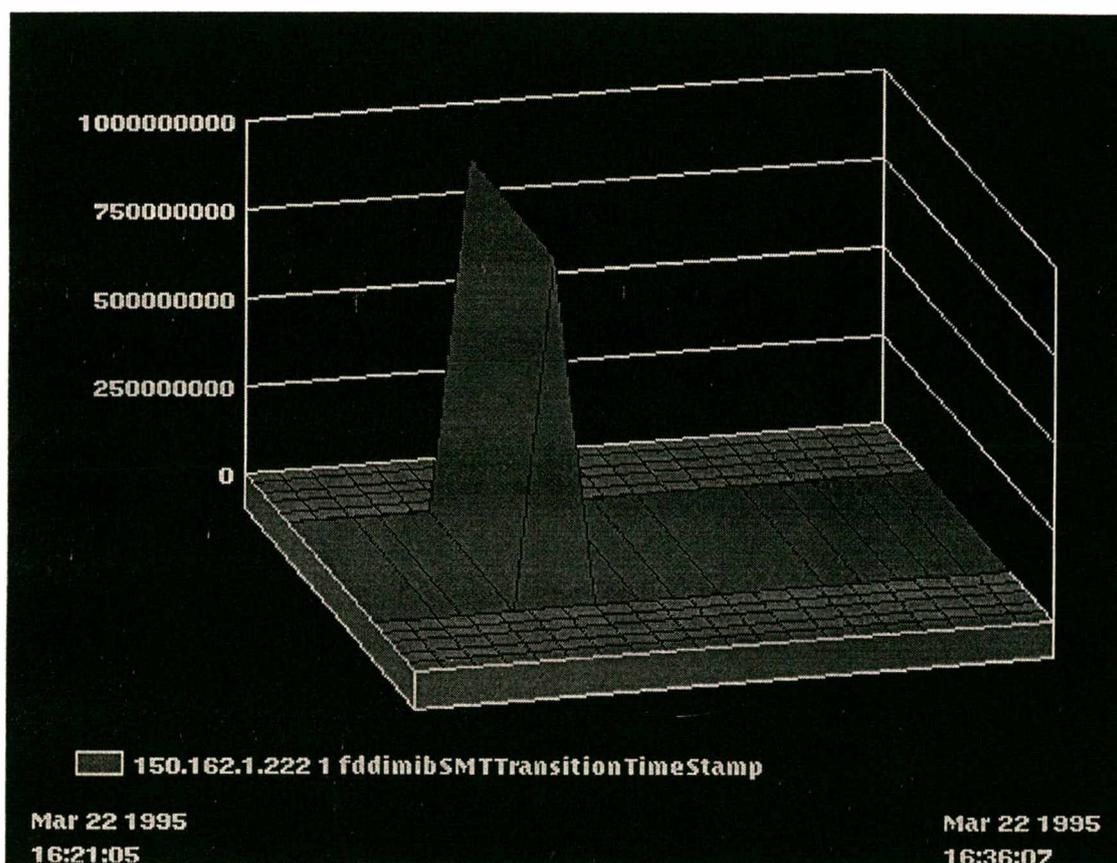


Figura A.21 - Gráfico das diferenças do tempo de transição do Token na coleta de dados do dia 21/03/95 às 16:21 - Dados coletados de 30 em 30 segundos

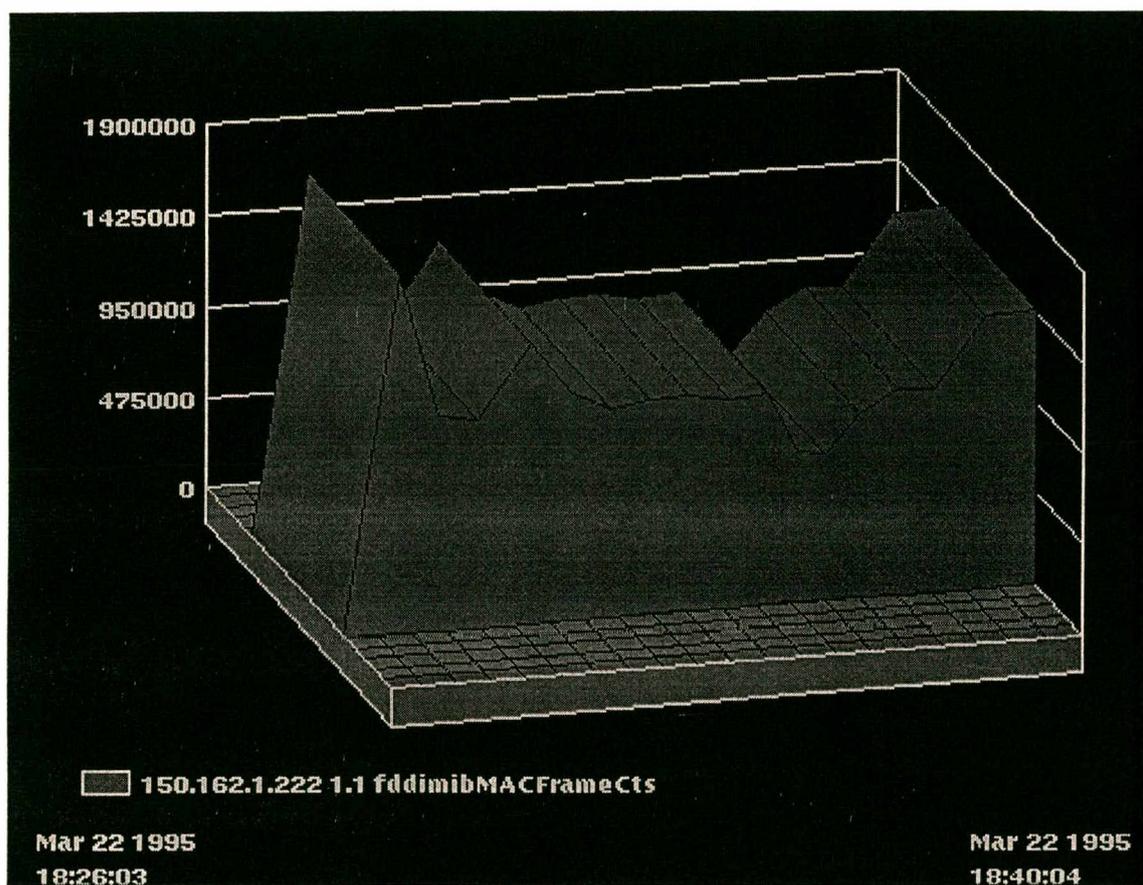


Figura A.22 - Gráfico das diferenças do número de frames transmitidos na coleta de dados realizada no dia 22/03/95 às 18:26 - Dados coletados de 30 em 30 segundos

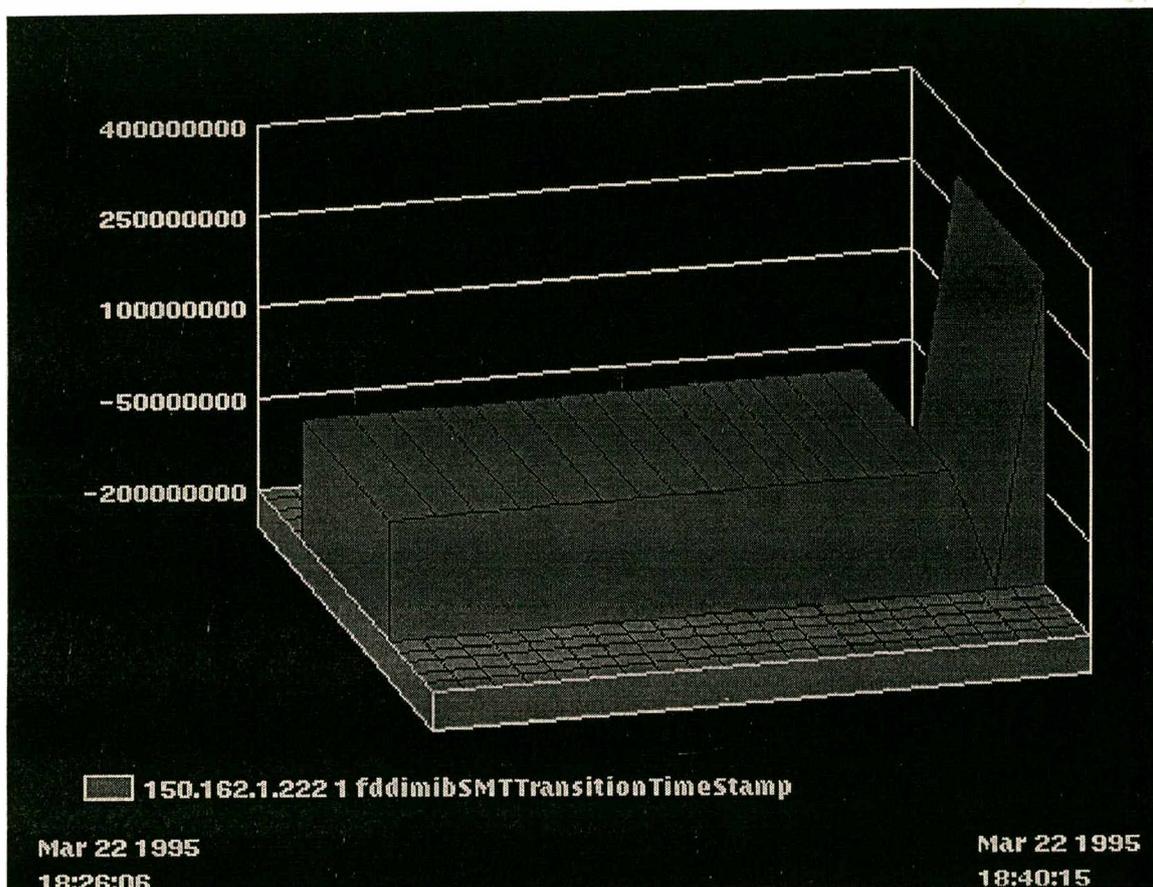


Figura 9.23 - Gráfico das diferenças do tempo de transição do Token na coleta de dados do dia 21/03/95 às 18:26 - Dados coletados de 30 em 30 segundos

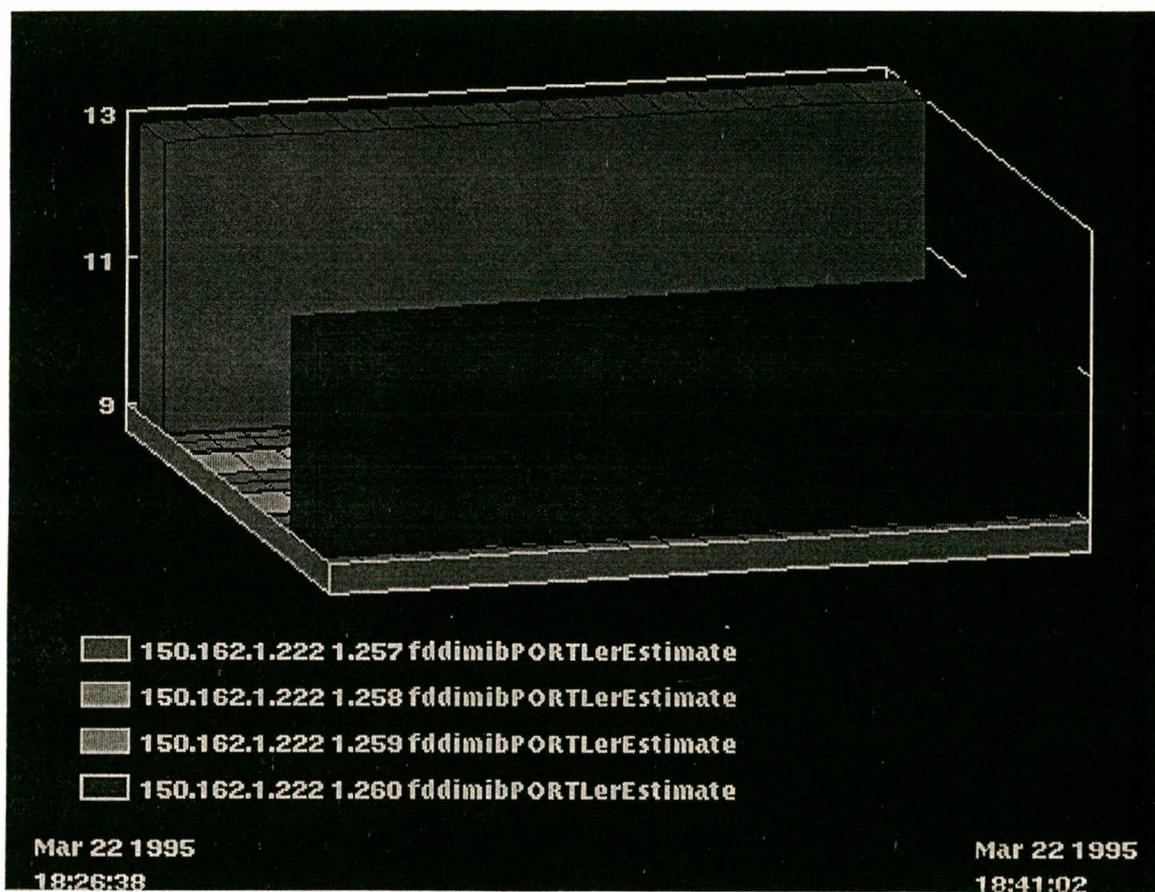


Figura 9.24 - Gráfico da taxa de erro de conexão da coleta de dados realizada no dia 22/03/95 às 18:26 - Dados coletados de 30 em 30 segundos