

**UNIVERSIDADE DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA**  
**DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**USO DE TDMoIP<sup>®</sup> COMO ALTERNATIVA  
PARA *BROADCASTING* EM APLICAÇÕES IPTV**

**WILSON DUTRA SAMPAIO**

**ORIENTADOR: DOUTOR PAULO HENRIQUE PORTELA DE  
CARVALHO**

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA**

**PUBLICAÇÃO PPGENE.DM – 283 A/06**

**BRASÍLIA/DF: OUTUBRO /2006**

**UNIVERSIDADE DE BRASÍLIA  
FACULDADE DE TECNOLOGIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**USO DE TDMoIP<sup>®</sup> COMO ALTERNATIVA  
PARA *BROADCASTING* EM APLICAÇÕES IPTV**

**WILSON DUTRA SAMPAIO**

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE  
ENGENHARIA ELÉTRICA DA FACULDADE DE TECNOLOGIA DA  
UNIVERSIDADE DE BRASÍLIA COMO PARTE DOS REQUISITOS  
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE.**

**APROVADA POR:**

---

**Prof<sup>o</sup> Paulo Henrique Portela de Carvalho, Docteur, ENE/UnB  
(Orientador)**

---

**Rodrigo Pinto Lemos, Doutor, EEE, UFG  
(Examinador Externo)**

---

**Paulo Roberto de Lira Gondim, Doutor, ENE/UnB  
(Examinador Interno)**

**BRASÍLIA, 30 DE OUTUBRO 2006.**

## FICHA CATALOGRÁFICA

SAMPAIO, WILSON DUTRA

Uso de TDMoIP<sup>®</sup> como alternativa para *broadcasting* em aplicações IPTV [Distrito Federal] 2006.

xxii, 253p., 297 mm (ENE/FT/UnB, Mestre, Engenharia Elétrica, 2006).

Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Elétrica.

1. TDM sobre IP

2. Emulação de Circuitos

3. Convergência de Redes

4. Transmissão Digital de Vídeo

5. IPTV

6. Vídeo *on-demand*

I. ENE/FT/UnB

II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

SAMPAIO, WILSON DUTRA (2006). Uso de TDMoIP<sup>®</sup> como alternativa para *broadcasting* em aplicações IPTV, Dissertação de Mestrado. Publicação PPGENE.DM-283 A/06. Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 190 p.

## CESSÃO DE DIREITOS

AUTOR: Wilson Dutra Sampaio

TÍTULO: Uso de TDMoIP<sup>®</sup> como alternativa para *broadcasting* em aplicações IPTV

GRAU: Mestre

ANO: 2006

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

---

Wilson Dutra Sampaio

QRSW4 BL A-3 - Apto 306, Setor Sudoeste

70.675-403 – Brasília – DF - Brasil

## **AGRADECIMENTOS**

À Brasil Telecom e aos meus gerentes e diretores ao longo dos últimos três anos, pelo apoio e flexibilização no horário de trabalho, que me proporcionaram essa grande oportunidade de enriquecimento pessoal e profissional.

Ao meu orientador Paulo Henrique Portela de Carvalho, pelo apoio durante a execução do trabalho e pela grande paciência, aceitando que o trabalho fosse conduzido dentro do ritmo possível em função das atividades da Brasil Telecom.

Aos colegas do LabCom Roque, Georges, Priscila, Martin e Eduardo, pela ajuda com os equipamentos, configuração das redes e utilização das instalações, além do companheirismo e da convivência diária.

Aos graduandos Diego, Fabrício, Márcio, Sérgio e Breno, pelo apoio na configuração dos roteadores, desenvolvimento das classes e rotinas em Java e utilização do Analisador.

À RAD do Brasil, na pessoa do Sr. Oscar Calderón Prager, pela cessão dos equipamentos IP-mux para demonstração, bem como acesso à documentação associada.

Para Julhiana e Viviani, cada uma na sua época, que se viram privadas de mim por tanto tempo e de tantas formas durante a realização desse trabalho.

## **RESUMO**

### **USO DE TDMoIP<sup>®</sup> COMO ALTERNATIVA PARA *BROADCASTING* EM APLICAÇÕES IPTV**

**Autor: Wilson Dutra Sampaio**

**Orientador: Paulo Henrique Portela de Carvalho**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, outubro de 2006**

O presente trabalho estuda as tecnologias de emulação de circuitos TDM sobre redes de pacotes, desenvolvidas como alternativa às aplicações VoIP para preservar os investimentos nas redes atuais. Com base nesse estudo, é proposta a utilização de circuitos TDM emulados como alternativa de baixo custo para transmissão vídeo digital, caracterizando uma aplicação IPTV com características diferentes das usuais.

Em seu desenvolvimento, são propostos novos mecanismos para o seqüenciamento de pacotes e a recuperação do relógio de transmissão no receptor, que são descritos e validados através de simulação no MatLab<sup>®</sup> e Simulink, bem como parcialmente implementados através de uma aplicação desenvolvida em Java, submetida à avaliação experimental na rede IP do LabCom/UnB.

O trabalho apresenta também os resultados experimentais obtidos com a utilização de uma aplicação comercial TDMoIP<sup>®</sup>, desenvolvida pela RAD, utilizada para emular um circuito E1 no entroncamento de uma central de comutação Trópico-RA, analisado sob diversas condições, incluindo o ponto de vista da central em relação à qualidade do enlace e o estudo comparativo com a aplicação Java desenvolvida.

## **ABSTRACT**

### **TDMoIP<sup>®</sup> AS AN ALTERNATIVE FOR *BROADCASTING* IN IPTV APPLICATIONS**

**Author: Wilson Dutra Sampaio**

**Supervisor: Paulo Henrique Portela de Carvalho**

**Programa de Pós-graduação em Engenharia Elétrica**

**Brasília, outubro de 2006**

This work presents new technologies for TDM emulation over packet networks, as an alternative to VoIP applications, saving network investments. After that, it proposes TDM pseudo-wires as an option to reduce digital video transmission costs, introducing a new kind of IPTV application.

The work proposes new algorithms to improve solutions to TDM over IP main challenges: packet-sequence processing and clock-recovery at receiver. These algorithms are described and implemented using simulation at MatLab<sup>®</sup> and Simulink, and tested using a simple Java-based application to deploy TDM over IP on a real network at LabCom/UnB.

Moreover, it also presents experimental results from a commercial solution for TDMoIP<sup>®</sup> technology, developed by RAD Data Communications, Inc. and used to deploy trunk E1 circuits for a *Trópico-RA* voice switch, in several conditions, analyzing pseudo-wire behavior and link quality as viewed by that switch. Finally, this work provides a comparison between that and proposed solution.

# SUMÁRIO

<b>1 - INTRODUÇÃO E MOTIVAÇÕES.....</b>	<b>1</b>
<b>2 – EMULAÇÃO DE CIRCUITOS TDM EM REDES IP.....</b>	<b>19</b>
<b>2.1 - A EMULAÇÃO DE CIRCUITOS EM REDES DE PACOTES.....</b>	<b>19</b>
2.1.1 - O conceito de emulação de circuitos em redes de pacotes.....	20
2.1.2 - Modelo de referência PWE3 para sincronização de rede.....	21
2.1.3 - As diversas propostas de emulação de circuitos TDM sobre PSN....	26
<b>2.2 - SERVIÇOS TDM SOBRE IP.....</b>	<b>31</b>
2.2.1 - Diferenças essenciais entre as redes TDM e redes IP.....	31
2.2.2 - Transporte de serviços TDM através de redes IP.....	32
2.2.3 – Implementação de pseudo-circuitos TDM.....	34
<b>2.3 - PROPOSTA DE EVOLUÇÃO DA IMPLEMENTAÇÃO TDMOIP® .....</b>	<b>41</b>
2.3.1 - A Implementação TDMoIP® da RAD.....	41
2.3.2 - O novo mecanismo proposto para seqüenciamento de pacotes.....	47
2.3.3 - O mecanismo alternativo proposto para sincronização TDM.....	53
<b>2.4 - COMPARAÇÃO ENTRE AS TECNOLOGIAS TDMOIP® E VOIP.....</b>	<b>59</b>
2.4.1 - Semelhanças entre as tecnologias.....	59
2.4.2 - Sensibilidade às perdas de pacotes.....	60
2.4.3 - Eficiência de utilização da largura de banda.....	62
2.4.4 – Complexidade computacional.....	64
2.4.5 - Latência e requisitos para transporte do tráfego de voz.....	67
<b>2.5 - PRINCIPAIS APLICAÇÕES TDM SOBRE IP E ETHERNET.....</b>	<b>70</b>
2.5.1 - Entroncamento de voz e extensão de circuitos.....	70
2.5.2 - Circuitos E1/T1 sobre redes de pacotes como rede de acesso local...	75
2.5.3 - Serviços de linha dedicada E1/T1 sobre redes de pacotes.....	77
2.5.4 - <i>Backbone</i> para redes móveis sobre rede se pacotes.....	79

<b>3 - ALTERNATIVA TDMoIP<sup>®</sup> PARA BROADCASTING IPTV.....</b>	<b>81</b>
<b>3.1 - A TRANSMISSÃO DE VÍDEO DIGITAL.....</b>	<b>81</b>
<b>3.2 - OS PADRÕES DE CODIFICAÇÃO DE VÍDEO.....</b>	<b>82</b>
3.2.1 – Os padrões H.261 e MPEG-1.....	83
3.2.2 – O padrão H.262 ou MPEG-2 Vídeo.....	83
3.2.3 – O padrão H.263.....	85
3.2.4 – O padrão H.264 ou MPEG-4 Vídeo.....	85
<b>3.3 - SERVIÇOS IPTV: CARACTERÍSTICAS E REQUISITOS.....</b>	<b>86</b>
<b>3.4 - TECNOLOGIAS PARA TRANSMISSÃO IPTV E VoD.....</b>	<b>88</b>
<b>3.5 - ADEQUAÇÃO TDMOIP<sup>®</sup> PARA TRANSMISSÃO DE VÍDEO.....</b>	<b>91</b>
3.5.1 - Fragmentação do fluxo de vídeo em pacotes.....	94
3.5.2 - Absorção de PDV na transmissão IPTV.....	96
3.5.3 - Sincronização dos receptores na transmissão IPTV.....	97
<b>4 - IMPLEMENTAÇÃO E RESULTADOS EXPERIMENTAIS.....</b>	<b>103</b>
<b>4.1 - MODELAGEM TDMoIP<sup>®</sup> UTILIZANDO MATLAB<sup>®</sup> E SIMULINK.....</b>	<b>103</b>
4.1.1 – Modelagem TDMoIP <sup>®</sup> no MatLab <sup>®</sup> e Simulink.....	103
4.1.2 – Simulação MatLab <sup>®</sup> do algoritmo de seqüenciamento de pacotes...	104
4.1.3 – Simulação Simulink do PLL para sincronização do receptor.....	111
<b>4.2 - EXPERIMENTAÇÃO TDMoIP<sup>®</sup> NA REDE LABCOM/UnB.....</b>	<b>132</b>
4.2.1 – Definição da topologia e configuração da rede LabCom.....	133
4.2.2 – Estudo e configuração dos equipamentos RAD IPmux-11.....	136
4.2.3 – Estudo e configuração da central Trópico-RA.....	141
4.2.4 – Ensaios de desempenho TDMoIP <sup>®</sup> .....	143
<b>4.3 - IMPLEMENTAÇÃO TDMOIP<sup>®</sup> EM JAVA.....</b>	<b>151</b>
4.3.1 – Desenvolvimento das ferramentas de transmissão e recepção.....	151
4.3.2 – Ensaios de desempenho TDMoIP <sup>®</sup> .....	154
<b>4.4 - ANÁLISE DOS RESULTADOS EXPERIMENTAIS.....</b>	<b>157</b>
4.4.1 – Desempenho do algoritmo de seqüenciamento de pacotes.....	157
4.4.2 – Desempenho do PLL para sincronização do receptor.....	158
4.4.3 – Desempenho do IPMux-11 na emulação de um enlace TDM.....	159
4.4.4 – Análise comparativa ente IPmux-11 e aplicação Java.....	161

<b>5 – CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>165</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>169</b>
<b>APÊNDICE A – ARQUITETURA PWE3.....</b>	<b>A1</b>
<b>A.1 - TERMINOLOGIA UTILIZADA NA ARQUITETURA PWE3.....</b>	<b>A1</b>
<b>A.2 - PROTOCOLOS UTILIZADOS NA ARQUITETURA PWE3.....</b>	<b>A9</b>
<b>A.3 - TIPOS DE FLUXOS DE DADOS NA ARQUITETURA PWE3.....</b>	<b>A10</b>
<b>A.4 - PILHA DE PROTOCOLOS NA ARQUITETURA PWE3.....</b>	<b>A13</b>
<b>A.5 - DETECÇÃO DE ERROS EM PWE3.....</b>	<b>A22</b>
<b>A.6 - CONGESTIONAMENTO EM PWE3.....</b>	<b>A23</b>
<b>A.7 - PLANO DE CONTROLE DA ARQUITETURA PWE3.....</b>	<b>A24</b>
<b>A.8 - IMPLEMENTAÇÃO DA ARQUITETURA PWE3 SOBRE IP.....</b>	<b>A25</b>
<b>APÊNDICE B – REQUISITOS PWE3.....</b>	<b>A29</b>
<b>B.1 - REQUISITOS GERAIS PARA PWE3.....</b>	<b>A29</b>
<b>B.2 - REQUISITOS PARA PWE3 EMULANDO SERVIÇOS TDM.....</b>	<b>A35</b>
<b>APÊNDICE C – ESPECIFICAÇÕES TDMoIP®.....</b>	<b>A41</b>
<b>C.1 – SUBCAMADA DE ENCAPSULAMENTO.....</b>	<b>A41</b>
<b>C.2 – SUBCAMADA DE ADAPTAÇÃO.....</b>	<b>A52</b>
<b>C.3 – SERVIÇOS TDM EMULADOS.....</b>	<b>A60</b>
<b>C.4 – GERENCIAMENTO DE FALHAS.....</b>	<b>A64</b>
<b>APÊNDICE D – PROCESSAMENTO DO NÚMERO DE SEQUÊNCIA.....</b>	<b>A69</b>
<b>APÊNDICE E – LISTAGEM DE FUNÇÕES MATLAB®.....</b>	<b>A73</b>
<b>E.1 – FUNÇÃO PARA SIMULAÇÃO DE PACOTES COM O ALGORITMO IETF.....</b>	<b>A73</b>
<b>E.2 – FUNÇÃO PARA SIMULAÇÃO DE PACOTES COM O NOVO ALGORITMO.....</b>	<b>A75</b>

## LISTA DE TABELAS

Tabela 1.1 - Escala de notas da avaliação subjetiva MOS.....	1
Tabela 1.2 - Custos mensais dos produtos Brasil Telecom para voz e dados.....	14
Tabela 2.1 - Comparação entre as propostas para transporte TDM sobre PSN.....	31
Tabela 2.2 - Duração das amostras TDM para cada tamanho do pacote de dados.....	55
Tabela 2.3 - Amostras de voz por tecnologia e codificador utilizado.....	62
Tabela 2.4 - <i>Overhead</i> e taxa efetiva por tecnologia e codificador utilizado.....	63
Tabela 2.5 - Efeito do atraso na qualidade do sinal de voz.....	67
Tabela 2.6 - Especificações de latência máxima para redes públicas e privadas.....	68
Tabela 2.7 - Latência fixa por tecnologia e codificador utilizado.....	68
Tabela 2.8 - Comparação de requisitos PWE3 com relação a VoIP.....	69
Tabela 3.1 - <i>Overhead</i> na PSN x tamanho dos pacotes na tecnologia SAToP.....	94
Tabela 3.2 - <i>Overhead</i> na PSN x tamanho dos pacotes na tecnologia TDMoIP®.....	95
Tabela 4.1 - Matriz de simulação da transmissão de pacotes através da PSN.....	105
Tabela 4.2 - Equivalência entre parâmetros experimentais e os utilizados no PLL Simulink.....	126
Tabela 4.3 - Regras de Zieger-Nichols em malha fechada para ajuste de controladores PID.....	129
Tabela 4.4 - Configuração de parâmetros para os <i>gateways</i> IPmux-11.....	139
Tabela 4.5 - Registro de eventos do IPmux-11, acesso HTTP.....	140
Tabela 4.6 - Regime de ocupação do tráfego concorrente.....	145
Tabela 4.7 - Condições de alarme para enlaces PCM na Trópico-RA.....	148
Tabela 4.8 - Comparação perdas efetivas de pacote nos dois algoritmos simulados....	157
Tabela 4.9 - Duração dos dados TDM e quantidade de pacotes para PDUs AAL1.....	162
Tabela 4.10 -Comparação do desempenho entre o IPmux-11 e a aplicação Java.....	163

## LISTA DE FIGURAS

Figura 1.1 - Penetração das tecnologias de comunicação na população mundial (%)	5
Figura 1.2 - Penetração das tecnologias de comunicação nos domicílios brasileiros...	5
Figura 1.3 - <i>Broadcasting</i> de TV utilizando uma rede IP.....	6
Figura 1.4 - Interconexão de PBXes e LANs, utilizando um <i>backbone</i> IP.....	7
Figura 1.5 - Modelo de referência CES sobre PSN.....	9
Figura 1.6 - Comunicação entre dois telefones convencionais usando VoIP.....	10
Figura 1.7 - Comunicação entre dois telefones convencionais usando TDMoIP.....	11
Figura 2.1 - Modelo de referência PWE3 para sincronização de rede.....	22
Figura 2.2 - Cenário de rede sincronizada pelo PE.....	23
Figura 2.3 - Cenário de rede sincronizada pelo CE.....	24
Figura 2.4 - Cenário de sincronização relativa.....	25
Figura 2.5 - Cenário de sincronização adaptativa.....	26
Figura 2.6 - Correntes da emulação TDM sobre redes de pacote.....	27
Figura 2.7 - Acomodação da variação no atraso dos pacotes pelo <i>jitter buffer</i> .....	34
Figura 2.8 - Recuperação do relógio no destino pelo método síncrono.....	36
Figura 2.9 - Recuperação do relógio no destino pelo método relativo SRTS.....	37
Figura 2.10 - Recuperação do relógio no destino pelo método adaptativo utilizando <i>timestamp</i> através do protocolo RTP.....	38
Figura 2.11 - Recuperação do relógio no destino pelo método adaptativo utilizando a ocupação, no tempo, do <i>jitter buffer</i> .....	38
Figura 2.12 - Esquemas de sincronização para <i>gateways</i> TDMoIP® RAD.....	46
Figura 2.13 - Ilustração do mecanismo de seqüenciamento proposto.....	48
Figura 2.14 - Exemplo de reconfiguração dinâmica do <i>jitter buffer</i> .....	51
Figura 2.15 - Mecanismo de recuperação do relógio baseado em <i>timestamp</i> e PLL.....	57
Figura 2.16 - Recuperação do relógio no receptor pelo método adaptativo proposto.....	58
Figura 2.17 - Entroncamento TDM convencional.....	71
Figura 2.18 - Entroncamento de serviços através de VoIP.....	73
Figura 2.19 - Entroncamento IP estático utilizando TDMoIP®.....	73

Figura 2.20 - Entroncamento IP com comutação utilizando TDMoIP®.....	74
Figura 2.21 - Entroncamento IP utilizando TDMoIP® com interoperabilidade VoIP.....	75
Figura 2.22 - Utilização de <i>metroethernet</i> como rede de acesso multi-serviços.....	76
Figura 2.23 - Serviços de linha dedicada sobre redes de dados.....	77
Figura 2.24 - <i>Backbone</i> IP para uma rede móvel celular.....	79
Figura 3.1 - Acesso IPTV numa configuração <i>triple play</i> .....	89
Figura 3.2 - Modelo convencional ponto a ponto para transmissão IPTV.....	90
Figura 3.3 - Alternativa de <i>broadcasting</i> IPTV usando TDMoIP® e <i>multicast</i> .....	93
Figura 3.4 - Esquema usual para controle de qualidade em IPTV.....	98
Figura 3.5 - Proposta de controle de qualidade baseado exclusivamente no receptor..	99
Figura 4.1 - Histograma de freqüências na medição do RTT entre o LabCom/UnB e os sítios Internet das universidades federais e portais dos governos estaduais brasileiros.....	106
Figura 4.2 - Gráfico da simulação MatLab®, apresentando os pacotes transmitidos (azul) e recebidos (verde).....	107
Figura 4.3 - Gráfico da simulação MatLab®, apresentando os instantes de transmissão (azul) e recepção (verde) de cada pacote.....	104
Figura 4.4 - Gráfico da simulação MatLab® utilizando o algoritmo proposto pelo IETF para seqüenciamento, apresentado no Apêndice A.....	109
Figura 4.5 - Gráfico da simulação MatLab® utilizando o novo mecanismo proposto de gravação no <i>jitter buffer</i> para seqüenciamento.....	110
Figura 4.6 - Princípio de operação da sincronização adaptativa usando PLL.....	111
Figura 4.7 - Diagrama de blocos do PLL em malha fechada.....	113
Figura 4.8 - Diagrama de blocos do filtro EWMA duplo.....	116
Figura 4.9 - Diagrama de blocos do controlador PID.....	119
Figura 4.10 - Diagrama de blocos do DCO.....	121
Figura 4.11 - Operação do PLL na recuperação do sincronismo no receptor.....	123
Figura 4.12 - Implementação do PLL com filtro EWMA no Simulink.....	124
Figura 4.13 - Evolução do erro entre $T(n)$ e $R(n)$ para o filtro EWMA no Simulink.....	127
Figura 4.14 - Evolução do erro $T(n) - R(n)$ para o filtro EWMA c/desvio de $-7,5\text{Hz}$ ....	128
Figura 4.15 - Sinal de controle para o filtro EWMA c/desvio de $-7,5\text{Hz}$ .....	129

Figura 4.16 - Implementação do PLL com o controlador PID proposto no Simulink....	130
Figura 4.17 - Evolução do erro $T(n) - R(n)$ para o controlador PID c/desvio de $-7,5\text{Hz}$	131
Figura 4.18 - Sinal de controle para o controlador PID c/desvio de $-7,5\text{Hz}$ .....	132
Figura 4.19 - Modelo conceitual do ambiente experimental .....	133
Figura 4.20 - Topologia da plataforma de testes PWE3.....	134
Figura 4.21 - Tela principal da ferramenta “Analisador”, do LabCom/UnB.....	135
Figura 4.22 - <i>Gateway</i> IPmux-11 da RAD utilizado nos testes.....	136
Figura 4.23 - Tela de configuração do <i>gateway</i> IPmux-11 utilizado nos testes.....	137
Figura 4.24 - Central Trópico-RA utilizada nos testes para geração de feixe E1.....	142
Figura 4.25 - Caminho utilizado pelos pacotes no pseudo-circuito TDM.....	144
Figura 4.26 - Desempenho de um pseudo-circuito E1 para enlaces de 4Mbps.....	146
Figura 4.27 - Desempenho de um pseudo-circuito E1 para enlaces de 6Mbps.....	146
Figura 4.28 - Desempenho de um pseudo-circuito E1 para enlaces de 8Mbps.....	147
Figura 4.29 - Desempenho de um pseudo-circuito E1 para enlaces de 10Mbps.....	147
Figura 4.30 - Desempenho de um pseudo-circuito E1 para 48 octetos por pacote.....	149
Figura 4.31 - Desempenho de um pseudo-circuito E1 para 240 octetos por pacote.....	149
Figura 4.32 - Desempenho de um pseudo-circuito E1 para 720 octetos por pacote.....	150
Figura 4.33 - Desempenho de um pseudo-circuito E1 para 1440 octetos por pacote.....	150
Figura 4.34 - Diagrama de blocos do Mux-IP PC.....	152
Figura 4.35 - Telas das aplicações Mux-IP PC.....	154
Figura 4.36 - Ensaio comparativo entre IPmux-11 e Mux-IP PC.....	155
Figura 4.37 - Desempenho do IPmux-11 no teste comparativo.....	156
Figura 4.38 - Desempenho do Mux-IP PC no teste comparativo.....	156
Figura A.1 - Multi-quadro com 16 quadros e sinalização CAS.....	A6
Figura A.2 - Utilização do canal 16 para sinalização CCS de um feixe E1.....	A7
Figura A.3 - Modelo lógico da pilha de protocolos PWE3.....	A9
Figura A.4 - Modelo de referência para a pilha de protocolos PWE3.....	A14
Figura A.5 - Detalhamento da camada de encapsulamento PWE3.....	A15
Figura A.6 - Arquitetura PWE3 sobre uma rede IP.....	A25
Figura A.7 - Arquitetura PWE3 sobre rede MPLS com palavra de controle.....	A26
Figura B.1 - Modelo de referência PWE3.....	A30
Figura B.2 - Cenário original: pseudo-circuito PE a PE.....	A31

Figura B.3 - Cenário alternativo: pseudo-circuito CE a CE.....	A32
Figura C.1 - Formato geral de um pacote TDMoIP <sup>®</sup> .....	A41
Figura C.2- Palavra de controle TDMoIP <sup>®</sup> de 32 bits.....	A42
Figura C.3 - Formato de um pacote TDMoIP <sup>®</sup> sobre UDP/IP.....	A45
Figura C.4 - Formato de um pacote TDMoIP <sup>®</sup> sobre MPLS.....	A48
Figura C.5 - Formato de um pacote TDMoIP <sup>®</sup> sobre <i>Ethernet</i> .....	A50
Figura C.6 - Estrutura AAL1 de 48 octetos.....	A54
Figura C.7- Exemplo de alocação dos canais TDM nos octetos das PDUs AAL1 para emulação de circuitos estruturada.....	A55
Figura C.8 - Estrutura de uma <i>P-format</i> PDU AAL1 para emulação de circuitos estruturada.....	A56
Figura C.9 - Exemplo de alocação de canais nos octetos das PDUs AAL1.....	A57
Figura C.10- PDUs AAL1 dentro do s pacotes TDMoIP <sup>®</sup> .....	A58
Figura C.11- Hierarquia digital plesiócrona.....	A61
Figura C.12- Fluxo TDM estruturado com canalização.....	A62
Figura C.13- Hierarquia digital síncrona.....	A63
Figura C.14- Configuração típica de uma rede TDMoIP <sup>®</sup> .....	A65

## LISTA DE NOMENCLATURAS E SIGLAS

<b>3G</b>	Terceira Geração das Redes Móveis
<b>AC</b>	<i>Attachment Circuit</i> – Circuito de Acesso
<b>ADPCM</b>	<i>Adaptive Differential Pulse Code Modulation</i> – Modulação por Código de Pulsos Diferencial Adaptativa
<b>ADSL</b>	<i>Asymmetric Digital Subscriber Line</i> – Linha Digital de Assinante Assimétrica
<b>AIS</b>	<i>Alarm Indication Signal</i> - Sinal de Indicação de Alarme
<b>ANATEL</b>	Agência Nacional de Telecomunicações
<b>ANSI</b>	<i>American National Standards Institute</i> – Instituto Nacional de Padrões dos Estados Unidos da América
<b>ATM</b>	<i>Asynchronous Transfer Mode</i> - Modo de Transferência Assíncrono
<b>ATMF</b>	<i>Asynchronous Mode Transfer Forum</i> – Fórum de discussão para ATM
<b>AU</b>	<i>Administrative Unit</i> – Unidade Administrativa (SDH)
<b>AUG</b>	<i>Administrative Unit Group</i> – Grupo de Unidades Administrativas (SDH)
<b>AVC</b>	<i>Advanced Video Coding</i> – Codificação Avançada de Vídeo
<b>BE</b>	<i>Best Effort</i> – Melhor Esforço
<b>BSC</b>	<i>Binary Synchronous Communication</i> – Comunicação Binária Síncrona
<b>BSC</b>	<i>Base Station Controller</i> - Controlador de Estação Base
<b>BTS</b>	<i>Base Transceiver Station</i> - Estação Base Transceptora de rádio
<b>C++</b>	Linguagem de Programação C orientada a objetos
<b>CA</b>	Corrente Alternada
<b>CAS</b>	<i>Channel-Associated Signaling</i> – Sinalização por Canal Associado
<b>CBR</b>	<i>Constant Bit Rate</i> – Taxa de Bits Constante
<b>CCS</b>	<i>Common Channel Signaling</i> – Sinalização por Canal Comum
<b>CD</b>	<i>Compact Disc</i> – Disco Compacto
<b>CE</b>	<i>Customer Edge</i> – Equipamento Cliente
<b>CES</b>	<i>Circuit Emulation Services</i> – Serviços de Emulação de Circuitos
<b>CESoPSN</b>	<i>Circuit Emulation Services over Packet-Switched Network</i> – Serviços de Emulação de Circuitos sobre Redes Comutadas em Modo Pacotes
<b>CODEC</b>	<i>COder/DECoder</i> – Codificador/Decodificador de Sinais de Voz e Vídeo

<b>CRC</b>	<i>Cyclic Redundancy Check</i> – Verificação de integridade de dados através de Código de Redundância Cíclica
<b>CS-ACELP</b>	<i>Conjugate Structure – Algebraic Code-Excited Linear Prediction</i> – Preditor Linear Excitado por Código Algébrico de Simetria Complementar
<b>CSN</b>	<i>Circuit-Switched Network</i> – Rede Comutada em Modo Circuitos
<b>CW</b>	<i>Control Word</i> – Palavra de Controle TDMoIP®
<b>DCO</b>	<i>Digitally Controlled Oscillator</i> – Oscilador Controlado Digitalmente
<b>DSL</b>	<i>Digital Subscriber Line</i> – Linha de Assinante Digital
<b>DSLAM</b>	<i>Digital Subscriber Line Access Module</i> – Módulo de Acesso para Linha de Assinante Digital
<b>DSP</b>	<i>Digital Signal Processor</i> – Processador Digital de Sinais
<b>DTH</b>	<i>Direct-To-Home</i> – Direto para Casa (TV por assinatura)
<b>DVB</b>	<i>Digital Video Broadcasting</i> – Difusão de Vídeo Digital
<b>DVD</b>	<i>Digital Versatile Disk</i> – Disco Versátil Digital
<b>EILD</b>	Exploração Industrial de Linha Dedicada
<b>ENE</b>	Departamento de Engenharia Elétrica/UnB
<b>EWMA</b>	<i>Exponentially Weighted Moving Average</i> – Média Móvel Exponencialmente Ponderada ou Alisamento Exponencial
<b>FAS</b>	<i>Frame Alignment Signal</i> – Sinal de Alinhamento de Quadro (TDM)
<b>FDM</b>	<i>Frequency Division Multiplexing</i> – Multiplexação por Divisão de Frequência
<b>FLL</b>	<i>Frequency-Locked Loop</i> – Enlace de Frequência Capturada
<b>FM</b>	Frequência Modulada
<b>FR</b>	<i>Frame Relay</i> – Protocolo de Chaveamento de Quadros
<b>FT</b>	Faculdade de Tecnologia/UnB
<b>FXO</b>	<i>Foreign eXchange Office</i> – Interface de linha analógica para operadora
<b>FXS</b>	<i>Foreign eXchange Subscriber</i> – Interface de linha analógica para assinante
<b>G.114</b>	Recomendação ITU-T que define os valores aceitáveis para o atraso em redes de comunicação de voz, orientado a redes públicas nacionais
<b>G.131</b>	Recomendação ITU-T que caracteriza e define as necessidades de controle de eco em redes de comunicação de voz.
<b>G.702</b>	Recomendação ITU-T que especifica a estrutura de taxas de bits da hierarquia digital PDH
<b>G.703</b>	Recomendação ITU-T que especifica as características físicas e elétricas das interfaces digitais PDH para taxas até 140 Mbps
<b>G.704</b>	Recomendação ITU-T que especifica as estruturas síncronas de quadro utilizadas nas interfaces G.703 até 45 Mbps
<b>G.706</b>	Recomendação ITU-T que especifica os procedimentos de alinhamento de quadro e cálculo de CRC relacionados às estruturas básicas de quadro definidas pela recomendação G.704

<b>G.707</b>	Recomendação ITU-T que especifica as taxas de bits das interfaces utilizadas pelos nós da hierarquia digital SDH
<b>G.711</b>	Recomendação ITU-T que especifica a codificação PCM para frequências de voz, utilizando taxa de 64 kbps
<b>G.723.1</b>	Recomendação ITU-T que especifica um codificador de voz de dupla taxa para comunicações multimídia, transmitindo a 5,3 kbps e 6,3 kbps
<b>G.726</b>	Recomendação ITU-T que especifica a codificação ADPCM para frequências de voz, utilizando taxas de 40 kbps, 32 kbps ou 16 kbps
<b>G.729</b>	Recomendação ITU-T que especifica a codificação de voz a taxas de 8 kbps, utilizando o algoritmo CS-ACELP.
<b>G.751</b>	Recomendação ITU-T que especifica os equipamentos de multiplexação digital TDM operando com taxas de 3 <sup>a</sup> . ordem (34,368 Mbps) e 4 <sup>a</sup> .ordem (139,264 Mbps), utilizando justificação positiva
<b>G.775</b>	Recomendação ITU-T que especifica os critérios para a detecção e liberação dos defeitos indicados pelas condições LOS e AIS
<b>G.810</b>	Recomendação ITU-T que apresenta as definições e a terminologia utilizada para redes de sincronismo
<b>G.823</b>	Recomendação ITU-T que especifica o controle do <i>jitter</i> e <i>skew</i> em redes digitais baseadas na hierarquia digital PDH, para taxas de 2,048 Mbps (E1)
<b>G.824</b>	Recomendação ITU-T que especifica o controle do <i>jitter</i> e <i>skew</i> em redes digitais baseadas na hierarquia digital PDH, para taxas de 1,544 Mbps (T1)
<b>G.826</b>	Recomendação ITU-T que especifica os objetivos e parâmetros de performance de erros para circuitos digitais internacionais operando a taxa de bits constante (CBR), na taxa primária (E1/T1) ou acima desta
<b>GW</b>	<i>Gateway</i> – Equipamento de controle e/ou acesso de uma rede ou serviço
<b>H.248</b>	Recomendação ITU-T que define o protocolo de comunicação entre <i>media gateways</i> e o <i>media gateway controller</i> , também conhecido como MEGACO em função de seu desenvolvimento cooperativo com a IETF
<b>H.261</b>	Recomendação ITU-T para codificação de vídeo em sistemas audiovisuais de faixa estreita
<b>H.262</b>	Designação ITU-T para o padrão MPEG-2 de vídeo: codificação genérica para imagens em movimento e informação de áudio associada
<b>H.263</b>	Recomendação ITU-T para codificação otimizada de vídeo para uso com taxas de bits inferiores àquelas tipicamente utilizadas em H.261
<b>H.264</b>	Recomendação ITU-T para codificação avançada de vídeo para serviços audiovisuais genéricos
<b>H.323</b>	Recomendação ITU-T para serviços de comunicação multimídia baseados em pacotes, incluindo videoconferência
<b>HDLC</b>	<i>High-Level Data Link Control</i> – Controle de Enlace de Dados de Alto Nível
<b>HDSL</b>	<i>High-bit-rate Digital Subscriber Line</i> – Linha Digital de Assinante de para Altas taxas de bits, de características simétricas
<b>HTTP</b>	<i>HyperText Transfer Protocol</i> – Protocolo de Transferência de HiperTexto
<b>IANA</b>	<i>Internet Assigned Numbers Authority</i> – Autoridade de Associação de Números para Internet
<b>IBGE</b>	Instituto Brasileiro de Geografia e Estatística

<b>ICMP</b>	<i>Internet Control Message Protocol</i> – Protocolo Internet de Mensagens de Controle
<b>IEEE</b>	<i>Institute of Electrical and Electronics Engineers</i> – Instituto dos Engenheiros Eletricistas e Eletrônicos
<b>IETF</b>	<i>Internet Engineering Task Force</i> – Força Tarefa para Engenharia da Internet
<b>iLBC</b>	<i>Internet Low Bit-rate CODEC</i> – Codificador/Decodificador Internet a Baixas Taxas
<b>IP</b>	<i>Internet Protocol</i> – Protocolo de Internet
<b>IPTV</b>	<i>Internet Protocol TeleVision</i> – Televisão baseada em Protocolo de Internet
<b>ISDN</b>	<i>Integrated Services Digital Network</i> – Rede Digital de Serviços Integrados
<b>ISO</b>	<i>International Organization for Standardization</i> – Organização Internacional para Padronização
<b>ISUP</b>	<i>Integrated Services User Part</i> – Protocolo de Sinalização para Usuário de Serviços Integrados (SS7)
<b>ITU-T</b>	<i>International Telecommunications Union – Telecommunication Standardization Sector</i> – União Internacional de Telecomunicações – Setor de Padronização para Telefocomunicações
<b>IWF</b>	<i>InterWorking Function</i> – Função de Interoperabilidade
<b>HDTV</b>	<i>High-Definition TeleVision</i> – Televisão de Alta Definição
<b>JVT</b>	<i>Joint Video Team</i> – Equipe Conjunta de Vídeo
<b>LABCOM</b>	Laboratório de Pesquisa e Desenvolvimento em Telecomunicações do Departamento de Engenharia Elétrica/UnB
<b>LAPB</b>	<i>Link Access Procedure Balanced</i> – Procedimento Balanceado de Acesso para Enlace de Dados
<b>LAN</b>	<i>Local Area Network</i> – Rede Local
<b>LDP</b>	<i>Label Distribution Protocol</i> – Protocolo de Distribuição de Etiquetas
<b>LEMOM</b>	Laboratório de Estruturas de Microondas e Ondas Milimétricas
<b>LER</b>	<i>Label Edge Router</i> – Roteador de Borda MPLS
<b>LOS</b>	<i>Loss of Signal</i> – Perda de Sinal (TDM)
<b>LSR</b>	<i>Label-Switching Router</i> – Roteador de Comutação de Etiquetas MPLS
<b>MAN</b>	<i>Metropolitan Area Network</i> – Rede Metropolitana
<b>MEF</b>	<i>Metro Ethernet Forum</i> – Fórum de discussão para <i>Ethernet</i> Metropolitana
<b>MEGACO</b>	<i>MEdia GAteway COntrol</i> – Protocolo de Controle de MGC
<b>MGC</b>	<i>Media Gateway Controller</i> – Controlador de <i>Gateways</i> de Mídia
<b>MGCP</b>	<i>Media Gateway Control Protocol</i> – Protocolo de Controle de MGC
<b>MOS</b>	<i>Mean Opinion Score</i> – Pontuação pela Opinião Média
<b>MP-ACELP</b>	<i>Multi Pulse – Algebraic Code-Excited Linear Prediction</i> – Preditor Linear Excitado por Código Algébrico Multi-pulsos

<b>MP3</b>	<i>MPEG-1 Audio Layer 3</i> – Camada 3 de Áudio do Padrão MPEG-1
<b>MPEG</b>	<i>Moving Picture Experts Group</i> – Grupo de Especialistas em Imagens em Movimento (Vídeo)
<b>MPLS</b>	<i>Multi-Protocol Label Switching</i> – Protocolo de Comutação de Etiquetas Multiprotocolos
<b>MP-MLQ</b>	<i>Multi Pulse – Maximum Likelihood Quantization</i> – Quantização pela Máxima Semelhança Multi-pulsos
<b>MTU</b>	<i>Maximum Transmission Unit</i> – Unidade Máxima de Transmissão
<b>MUX</b>	<i>MUltipleX</i> – Equipamento de Multiplexação
<b>NEA</b>	Número Equivalente de Assinante
<b>NS</b>	Número de Sequência
<b>NSP</b>	<i>Native Service Processing</i> – Processamento de Serviço Nativo
<b>NTSC</b>	<i>National Television Systems Committee</i> – Comitê Nacional de Sistemas de Televisão, padrão norte-americano para televisão em cores
<b>ONG</b>	Organização Não-Governamental
<b>ONU</b>	Organização das Nações Unidas
<b>OOF</b>	<i>Out of Frame Alignment Synchronization</i> – Ausência de Sincronismo de Quadros
<b>OSPF</b>	<i>Open Shortest Path First</i> – Primeiro pelo Caminho Livre Mais Curto
<b>PABX</b>	<i>Private Automatic Branch eXchange</i> – Equipamento Automático de Comutação Privada
<b>PBX</b>	<i>Private Branch eXchange</i> – Equipamento de Comutação Privada
<b>PAL</b>	<i>Phase-Alternating Line</i> – Linha de Fase Alternada, padrão europeu para televisão em cores
<b>PC</b>	<i>Personal Computer</i> – Computador Pessoal ou Microcomputador
<b>PCM</b>	<i>Pulse Code Modulation</i> – Modulação por Código de Pulsos
<b>PCS</b>	Ponto de Controle de Sinalização (SS7)
<b>PDV</b>	<i>Packet Delay Variation</i> – Variação no Atraso dos Pacotes
<b>PE</b>	<i>Provider Edge</i> – Equipamento Provedor
<b>PESQ</b>	<i>Perceptual Evaluation of Speech Quality</i> – Avaliação Perceptiva da Qualidade de Voz
<b>PDH</b>	<i>Plesiochronous Digital Hierarchy</i> – Hierarquia Digital Plesiócrons
<b>PDU</b>	<i>Protocol Data Unit</i> – Unidade de Dados de Protocolo
<b>PID</b>	Controle Proporcional, Integral e Derivativo
<b>PLC</b>	<i>Packet-Loss Concealment</i> – Mitigação de Perdas de Pacote
<b>PLL</b>	<i>Phase-Locked Loop</i> – Enlace de Fase Capturada
<b>PNAD</b>	Pesquisa Nacional por Amostra de Domicílios
<b>Ppm</b>	Partes por milhão

<b>PSN</b>	<i>Packet-Switched Network</i> – Rede Comutada em Modo Pacotes
<b>PTS</b>	Ponto de Transferência de Sinalização (SS7)
<b>PW</b>	<i>Pseudo-Wire</i> – Pseudo-Circuito
<b>PWE3</b>	<i>Pseudo-Wire Emulation Edge-to-Edge</i> – Emulação Fim a Fim de Pseudo-Circuitos
<b>QoS</b>	<i>Quality of Service</i> – Qualidade de Serviço
<b>RAI</b>	<i>Remote Alarm Indication</i> – Sinal de Indicação de Alarme Remoto
<b>RFI</b>	<i>Remote Defect Indication</i> – Sinal de Indicação de Defeito Remoto
<b>REMAV</b>	Rede Metropolitana de Alta Velocidade
<b>RFC</b>	<i>Request for Comments</i> – Nomenclatura dos Padrões IETF
<b>RNP</b>	Rede Nacional de Pesquisa
<b>RTFC</b>	Rede Telefônica Fixa Comutada
<b>RTP</b>	<i>Real Time Protocol</i> – Protocolo para Dados em Tempo Real
<b>RTCP</b>	<i>Real Time Control Protocol</i> – Protocolo de Controle para Dados em Tempo Real
<b>RTSP</b>	<i>Real Time Streaming Protocol</i> – Protocolo de Transmissão de Vídeo em Tempo Real
<b>RTT</b>	<i>Round-Trip Time</i> – Tempo de ida e volta
<b>Rx</b>	Receptor
<b>SAT</b>	<i>Structure-Agnostic Transport</i> – Transporte sem Consciência da Estrutura
<b>SAToIP</b>	<i>Structure-Agnostic Transport over Internet Protocol</i> – Transporte sem Consciência da Estrutura sobre Protocolo de Internet
<b>SDH</b>	<i>Synchronous Digital Hierarchy</i> – Hierarquia Digital Síncrona
<b>SECAM</b>	<i>SÉquentiel Couleur Á Mémoire</i> – Cor seqüencial com Memória, padrão francês para televisão em cores
<b>SIGTRAN</b>	<i>SIGnaling TRANsport Protocol</i> – Protocolo de Transporte de Sinalização
<b>SG</b>	<i>Signaling Gateway</i> – Gateway de Sinalização
<b>SIP</b>	<i>Session Initiation Protocol</i> – Protocolo de Iniciação de Sessão
<b>SMP</b>	Serviço Móvel Pessoal
<b>SNA</b>	<i>Synchronous Network Architecture</i> – Arquitetura de Rede Síncrona
<b>SOHO</b>	<i>Small Office/Home Office</i> – Pequeno Escritório/Escritório Doméstico
<b>SONET</b>	<i>Synchronous Optical Network</i> – Rede Óptica Síncrona
<b>SS7</b>	<i>Signaling System Number 7</i> – Sinalização por Canal Comum Número 7
<b>STM</b>	<i>Synchronous Transport Module</i> – Módulo de Transporte Síncrono
<b>STS</b>	<i>Synchronous Transport Signal</i> – Sinal de Transporte Síncrono
<b>TDM</b>	<i>Time Domain Multiplexing</i> – Multiplexação no Domínio do Tempo

<b>TDMoIP</b>	<i>Time Division Multiplexing over Internet Protocol</i> – Multiplexação no Domínio do Tempo sobre Protocolo de Internet
<b>TELENOR</b>	<i>Telecommunications of Norway</i> – Telecomunicações da Noruega
<b>TIC</b>	Telefones fixos e celulares, Internet, microComputadores, rádio e televisão
<b>TG</b>	<i>Trunking Gateway</i> – Gateway de Entroncamento
<b>TU</b>	<i>Tributary Unit</i> – Unidade Tributária (SDH)
<b>TUG</b>	<i>Tributary Unit Group</i> – Grupo de Unidades Tributárias (SDH)
<b>Tx</b>	Transmissor
<b>UDP</b>	<i>User Datagram Protocol</i> – Protocolo de Datagramas de Usuário
<b>UFAC</b>	Universidade Federal do Acre
<b>UMTS</b>	<i>Universal Mobile Telecommunication System</i> - Sistema Universal de Telecomunicações Móveis
<b>UN</b>	<i>United Nations</i> – Nações Unidas
<b>UnB</b>	Universidade de Brasília
<b>UNIR</b>	Fundação Universidade Federal de Rondônia
<b>UTFORS</b>	Operadora de Telecomunicações da Suécia, adquirida pela TELENOR
<b>UTP</b>	<i>Unshielded Twisted Pair</i> – Par Trançado sem Blindagem
<b>VBR</b>	<i>Variable Bit Rate</i> – Taxa de Bits Variável
<b>VC</b>	<i>Virtual Container</i> – Contêiner Virtual (SDH)
<b>VCD</b>	<i>Video Compact Disc</i> – Disco Compacto de Vídeo
<b>VCEG</b>	<i>Video Coding Experts Group</i> – Grupo de Especialistas Codificação de Vídeo
<b>VHF</b>	<i>Very High Frequency</i> – Frequência Muito Alta
<b>VHS</b>	<i>Video Home System</i> – Sistema de Vídeo Doméstico
<b>VLC</b>	<i>VideoLan Client</i> – Cliente de Vídeo para LAN
<b>VoD</b>	<i>Video on Demand</i> – Transmissão de Vídeo sob Demanda
<b>VoIP</b>	<i>Voice over IP</i> – Voz sobre Protocolo de Internet
<b>VRML</b>	<i>Virtual Reality Modelling Language</i> – Linguagem para Modelamento de Realidade Virtual
<b>VSB</b>	<i>Vestigial-Sided Band</i> – Banda Lateral Vestigial, modulação utilizada na transmissão de TV aberta no Brasil
<b>WAN</b>	<i>Wide Area Network</i> – Rede de Grande Alcance
<b>xDSL</b>	<i>x-Digital Subscriber Line</i> – Família de Linhas Digitais de Assinantes
<b>Y.1413</b>	Recomendação ITU-T que especifica a interoperabilidade entre redes TDM e MPLS no plano de usuário
<b>Y.1414</b>	Recomendação ITU-T que especifica a interoperabilidade entre redes MPLS e serviços de voz

# 1 - INTRODUÇÃO E MOTIVAÇÕES

As Redes de Telecomunicações estão completando 150 anos, e vêm se desenvolvendo de uma forma fascinante em diversos aspectos. Nas últimas décadas, em especial surgiram e se consolidaram novas tecnologias que incrementam a capacidade dessas Redes de forma assustadora, ao mesmo tempo em que as tornam mais simples e, portanto, mais confiáveis, mais eficientes e fáceis de operar – logo, mais baratas.

A última grande inovação nesse cenário foi a disseminação das redes de transporte de alta velocidade baseadas no protocolo IP (*Internet Protocol*), que trouxeram a simplicidade, robustez, eficiência no uso da banda e baixo custo operacional - popularizadas com a difusão da grande rede mundial de computadores, a Internet - para o mundo das telecomunicações tradicionais, originalmente dominado por tecnologias proprietárias, com grande período de desenvolvimento e maturação, suportadas essencialmente por alguns grandes fabricantes mundiais de equipamentos e construídas pelas grandes operadoras ao longo das últimas décadas, em virtude dos vultosos investimentos associados.

No aspecto “voz”, serviço para a qual foi desenvolvida, a infra-estrutura tradicional de telecomunicações, caracterizada pela famosa confiabilidade dos “cinco noves” (99,999%) e pela boa qualidade de conversação, auferida por uma nota MOS (*Mean Opinion Score*) superior a 4,0 segundo padrões internacionais reconhecidos – ver Tabela 1.1, tem penetração praticamente universal no mercado mundial, permitindo o estabelecimento de intercomunicações planetárias em poucos segundos e quase sempre de forma completamente automática. Sem dúvida, essa é uma característica fundamental do serviço e da qual nenhum usuário abrirá mão, devendo ser considerada quando da análise de qualquer alternativa às redes tradicionais.

Tabela 1.1 - Escala de notas da avaliação subjetiva MOS

Nota	1,0	2,0	3,0	4,0	5,0
Conceito	Ruim	Insatisfatório	Regular	Bom	Excelente

No aspecto “dados”, serviço que passou a fazer parte das redes de telecomunicações nos últimos 40 anos, com o desenvolvimento dos primeiros terminais interativos na década de 1960; também existem tecnologias tradicionais consagradas, que permitem a construção de redes de transporte extremamente confiáveis e seguras, sejam para o acesso remoto a grandes centros de processamento, com supercomputadores para uso militar, pesquisas científicas ou *data centers* de grandes corporações, ou para uma simples interconexão de redes locais, utilizadas para suportar aplicações comerciais, educacionais ou de entretenimento, como controlar remotamente o estoque de uma rede de farmácias, assistir a uma aula por videoconferência ou receber mensagens eletrônicas dos amigos.

Dentre as tecnologias desenvolvidas para a transmissão de dados, evoluímos da jurássica SNA (*Synchronous Network Architecture*) desenvolvida pela IBM para seus famosos *mainframes*, reis absolutos do chamado “processamento de dados” até a década de 1980, dos protocolos precursores de enlace como BSC (*Binary Synchronous Communication*) HDLC (*High-Level Data Link Control*) e do LAPB (*Link Access Procedure Balanced*), também conhecido como X.25, que consolidou a comutação de pacotes, para ficar entre os mais famosos; até as tecnologias de transmissão de dados ainda hoje consagradas nas redes de telecomunicações das Operadoras, como o FR (*Frame-Relay*), uma evolução do X.25, e o ATM (*Asynchronous Transfer Mode*), grande promessa dos anos 1990 para as redes de comunicação de dados, que acabou não alcançado nem a abrangência e nem os resultados esperados; ambas implantadas sobre redes ópticas de transporte de grande capacidade e confiabilidade, baseadas na arquitetura SDH/SONET (*Synchronous Digital Hierarchy/Synchronous Optical NETWORK*). Maiores informações sobre a evolução das redes de dados e as características de cada uma dessas tecnologias podem ser encontradas em [SOARES1995].

Contudo, uma rede desenvolvida sob conceitos relativamente simples, de arquitetura extremamente flexível, em função dos seus requisitos originais de controle descentralizado e capacidade para manter-se razoavelmente operacional sob extremas condições de falha em boa parte de seus nós – afinal, foi concebida para aplicações militares; acabou se tornando a grande “vedete” do final do século passado, e praticamente o padrão atual para transporte de dados, e uma alternativa crescente para todos os demais serviços de

telecomunicações, incluindo a sesquicentenária “voz” e o cinqüentenário “vídeo”, que juntos modificaram a civilização humana no século XX.

O que torna as redes IP tão atraentes, a ponto de provocar uma revolução em mercados consolidados como o *broadcasting* de vídeo, onde a televisão representa um consultor doméstico, o entretenimento mais acessível, a babá eletrônica, o grande agregador (ou seria desagregador?) da sociedade atual. No Brasil, não seria exagero dizer que a televisão aberta, que entra todos os dias na casa de cada um de nós, constitui um verdadeiro “Quarto Poder” da República, com influência até mais direta e imediata na nossa vida cotidiana do que os tradicionais Executivo, Legislativo e Judiciário, nascidos em seu equilíbrio nos ideais libertários franceses há 250 anos, 100 anos antes das Telecomunicações.

Da mesma forma, no lado oposto, a miniaturização e redução nos preços dos equipamentos de vídeo permite hoje a praticamente qualquer cidadão de classe média a geração de conteúdos de áudio e vídeo de boa qualidade, da forma e com o objetivo que desejar, trazendo cada vez mais o protagonismo para a mídia de informação e entretenimento.

No meio desse caminho, temos as aplicações que associam a facilidade na geração de conteúdo baseado em vídeo com a capacidade de transmissão e difusão desse conteúdo a grandes distâncias, de forma simples e barata. Essas aplicações abrem um universo completamente novo para serviços de utilidade pública, como tele-educação, tele-medicina e tele-monitoramento; e inovações praticamente ilimitadas no campo do entretenimento.

Essa revolução é sentida a cada instante, através de novos hábitos e costumes, como os jogos eletrônicos em grupo e à distância, inimagináveis para a geração, hoje na faixa dos trinta anos, que passava suas tardes na companhia de um Atari 2600; a busca de relacionamentos “virtuais” na “segurança” psicológica do mundo digital, talvez a aplicação que mais traga novos usuários para a Internet hoje em dia, como atestam o sucesso de sítios como o *Orkut*, o *Myspaces* e o *Messenger*, sejam eles pré-adolescentes curiosos ou pessoas da “melhor idade” que encontram na tela de seus micros uma nova vida, longe de suas limitações físicas; ou mesmo o velho cinema do século XIX, grande disseminador, no século XX, do chamado *american way of life* para a cultura ocidental, e com ele boa parte do poder da única superpotência mundial; que agora precisa ser repensado, uma vez que

toda a produção mundial, incluindo um obscuro e independente filme iraniano ou indiano, poderia estar disponível, em nossa casa, ao toque de um botão do controle remoto.

Câmeras de segurança e monitoramento estão hoje integradas em nossas vidas, localizadas na maioria dos grandes edifícios, tanto públicos como privados, e mesmo nas ruas das grandes cidades, concretizando a onipresença do *big-brother*, preconizado por George Orwell em seu livro “1984”, escrito em 1948. Essa realidade é tão presente e concreta que pessoas se dispõem, em todo o mundo, a serem observadas 24 horas por dia por milhões, num programa de televisão muito popular no Brasil que as eleva à condição de celebridade instantânea, e cujo nome é muito apropriadamente inspirado na obra de Orwell.

Nunca foi tão fácil e barato, em toda a história da humanidade, gerar conteúdo, seja voz, vídeo, textos, imagens, músicas, sem depender de emissoras de rádio ou TV, editoras, jornais, revistas ou gravadoras; da mesma forma, a Internet trouxe a possibilidade de democratização total desse conteúdo, inicialmente através dos *e-mails*, *homepages* e grupos de discussão, depois dos *blogs*, *fotologs* e, agora, “*videologs*” e sítios como *YouTube*, *MySpaces* e *Orkut*, canais independentes e disponíveis a quem estiver interessado e tiver a paciência de procurar. E mesmo essa paciência hoje não precisa ser muito grande, com o aperfeiçoamento constante das ferramentas de busca para achar qualquer coisa em qualquer lugar dentro da grande rede mundial de computadores.

Contudo, esse admirável mundo interconectado e digital, a princípio livre e democrático, ainda é acessível para somente 15% da população mundial, a reduzida fatia que detém o privilégio de acesso a um computador conectado à rede Internet, conforme atestam os indicadores da ONU criados em 2000 para acompanhamento das Metas do Milênio [ONU2006], cujo gráfico de penetração percentual das principais tecnologias de comunicação é apresentado Figura 1.1.

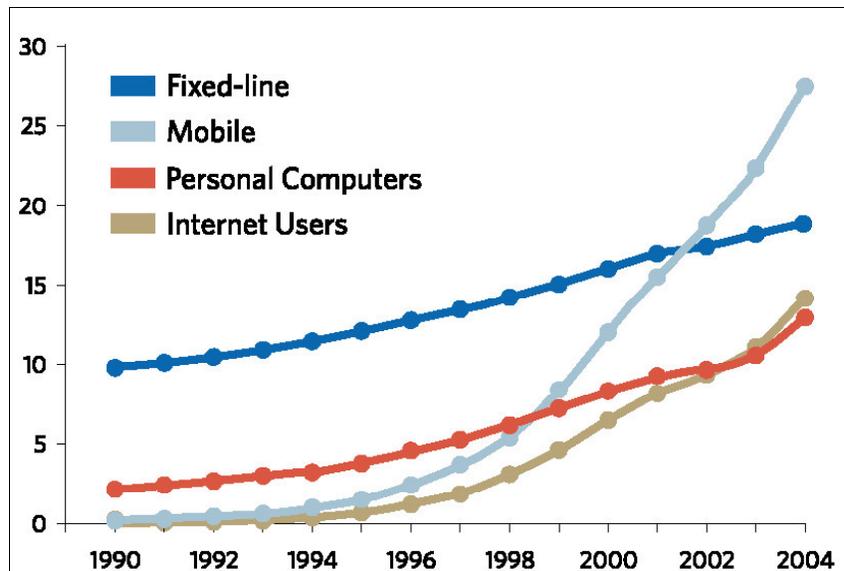


Figura 1.1 - Penetração das tecnologias de comunicação na população mundial (%).

No Brasil, cerca de 32 milhões de pessoas ou apenas 21% da população acima de 10 anos estão incluídos nesse seleto grupo segundo a PNAD (Pesquisa Nacional por Amostra de Domicílios), realizada em 2005 pelo IBGE (Instituto Brasileiro de Geografia e Estatística) e recentemente divulgada [IBGE2006], como mostra o gráfico de penetração das tecnologias, segundo essa pesquisa, apresentado na Figura 1.2.

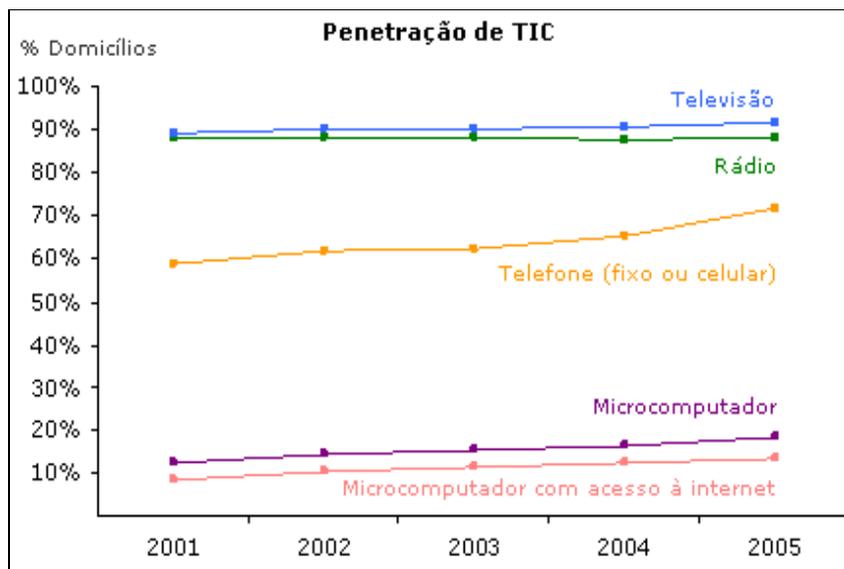


Figura 1.2 - Penetração das tecnologias de comunicação nos domicílios brasileiros.

Entretanto, o mesmo gráfico da Figura 1.2 demonstra que a quase totalidade dos lares brasileiros tem pelo menos uma televisão e um equipamento de rádio, cerca de 40 milhões de famílias têm também telefones fixos [ANATEL2006] e quase 96 milhões de pessoas andam todos os dias com seus telefones celulares no bolso (ou na bolsa) [ANATEL2006a], sendo esses os meios efetivamente disponíveis para a divulgação massiva de conteúdo e/ou prover serviços de telecomunicações à população.

Assim, a difusão (*broadcasting*) de vídeo e áudio, dominada por grandes empresas/grupos econômicos e dependentes de pesados investimentos; e as redes tradicionais de telecomunicações, fixas e móveis, ainda são os melhores instrumentos para alcançar a grande maioria da população, no Brasil e no mundo. Isso torna premente a necessidade de buscar alternativas simples e baratas, como a mostrada na Figura 1.3, para levar o grande universo de conteúdo disponível no mundo Internet, bem como serviços mais baratos de telecomunicações, aos equipamentos disponíveis, preservando os investimentos realizados e concentrando as evoluções tecnológicas na rede de transporte, onde beneficiam toda a população e não apenas um pequeno grupo que pode pagar pelos novos serviços.

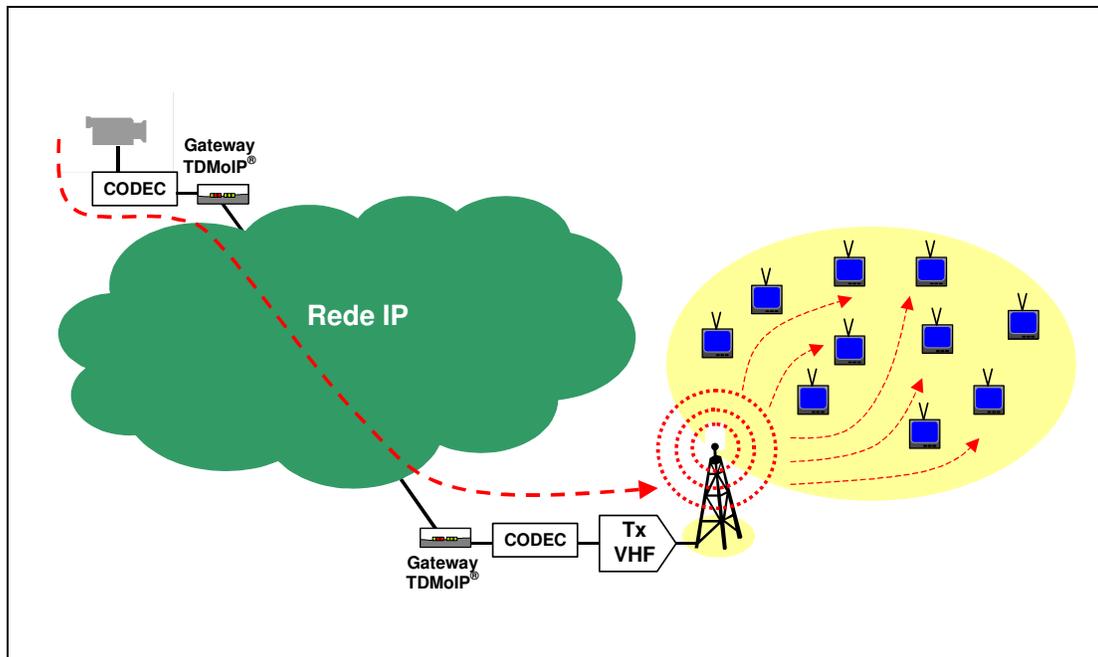


Figura 1.3 - *Broadcasting* de TV utilizando uma rede IP.

Considerando inicialmente o *broadcasting* de TV, é fato consumado que somente uma grande rede de televisão, sustentada por sua rede de patrocinadores, possui capacidade para

manter transmissões nacionais via satélite, mas é bastante provável que uma grande quantidade de emissoras comunitárias de baixo custo e cobertura local pudesse receber programação nacional consistente, de caráter educativo e social, gerada de diversas formas e por diversas entidades, através da condição mostrada na Figura 1.3, para então retransmiti-la para sua comunidade.

Da mesma forma, somente uma grande operadora de telecomunicações tem capacidade para construir uma rede de abrangência nacional utilizando as tecnologias tradicionais, como SDH, mas uma pequena operadora poderia oferecer serviços de voz e dados a baixos custos, “alugando” determinada largura de banda de um provedor nacional que detenha uma rede de transporte em modo pacotes de boa cobertura, como mostrado na Figura 1.4.

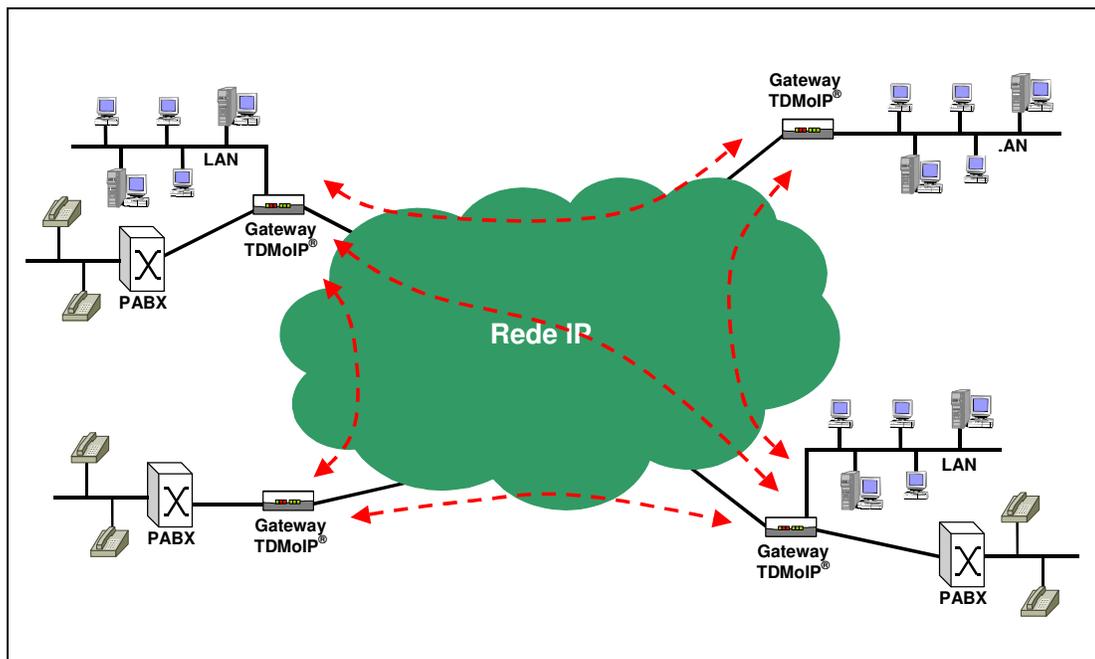


Figura 1.4 - Interconexão de PBXes e LANs, utilizando um *backbone* IP.

Nessa linha, considerando a grande penetração das redes IP nos últimos anos, como forma de aproveitar sua popularização, abrangência e eficiência, esforços significativos vêm sendo feitos no desenvolvimento de novos serviços que tragam aplicações tradicionais da rede de telecomunicações para dentro dessas redes, tais como o mais difundido deles, VoIP (*Voice over IP*), ou transmissão de voz sobre redes IP, com o objetivo de oferecer serviços de voz a tarifas bem mais baixas que as praticadas pelas Operadoras convencionais de telecomunicações, em função do custo reduzido de transporte das chamadas nas redes IP

quando comparado com as redes TDM (*Time Division Multiplexing*) tradicionais, decorrentes da maior eficiência de ocupação de banda e simplicidade.

Em essência, a tecnologia VoIP representa uma mudança revolucionária para as redes de telecomunicações atuais, exigindo a substituição de boa parte de seus componentes por novos módulos baseados na arquitetura IP, incluindo os equipamentos terminais, com vantagens ainda um pouco nebulosas, padrões não totalmente consolidados, tecnologias ainda não completamente maduras e expectativas de longo prazo para retorno dos investimentos. Dessa forma, as grandes Operadoras mundiais ainda caminham a passos lentos e cautelosos em direção a essa migração, sobretudo em função da grande capacidade ociosa das redes de transmissão implantadas com o *boom* do final da década de 90, onde se projetava uma demanda exponencial por serviços maravilhosos e ávidos por banda larga, que acabou não acontecendo de fato, comprometendo bastante os investimentos do setor nos últimos anos. Uma tecnologia cujo principal benefício é a redução das tarifas de serviço aos clientes, em nome de uma melhoria na eficiência de aproveitamento das redes de transporte, cuja capacidade atual deve suprir facilmente a demanda pelos próximos 10 anos sem maiores investimentos, não parece ser a prioridade atual das Operadoras convencionais, sobretudo das concessionárias, também denominadas *incumbents*.

Nesse cenário, aparecem oportunidades para tecnologias alternativas, menos revolucionárias e não tão dependentes de grandes investimentos, dentre as quais a emulação de circuitos tradicionais através de redes de transporte em modo pacotes, sobretudo redes IP, conciliando as vantagens oferecidas pelas redes baseadas na comutação de pacotes com a confiabilidade e estabilidade das redes baseadas na comutação de circuitos, traçando o chamado caminho “evolucionário”, que prevê a evolução progressiva das redes existentes em direção ao chamado “Mundo IP”.

Nessas tecnologias, onde o TDMoIP® (*Time Division Multiplexing over Internet Protocol*), ou seja, o transporte de circuitos tradicionais multiplexados no tempo sobre redes IP, representa uma das alternativas, a rede IP é efetivamente utilizada como rede de transporte, mas de forma transparente para os equipamentos terminais, que mantêm integralmente suas funcionalidades e características, tais como qualidade de áudio, protocolos de sinalização e interoperabilidade, preservando todos os investimentos realizados e permitindo a

comunicação de voz, dados ou vídeo como se estivessem interligados por uma rede TDM tradicional baseada em circuitos dedicados ou de alocados de forma determinística. Isso, é claro, desde que sejam garantidos recursos suficientes dentro da rede IP para garantir a taxa de transmissão (largura de banda, perdas de pacote) e a qualidade do canal (baixo atraso e pequena variação nesse atraso).

Essa transparência é obtida através de Serviços de Emulação de Circuitos ou CES (*Circuit Emulation Services*), conceito onde um equipamento de interface é colocado junto aos dispositivos terminais, interagindo com os mesmos como se fosse um equipamento nativo da rede TDM, empacotando os fluxos de dados e inserindo-os na rede de transporte em modo pacotes ou PSN (*Packet-Switching Network*), bem como extraindo os pacotes recebidos e recompondo o fluxo original TDM. O modelo de referência desse conceito CESoPSN (*Circuit Emulation Services over Packet-Switched Network*) é apresentado na Figura 1.5, para a emulação de um circuito TDM.

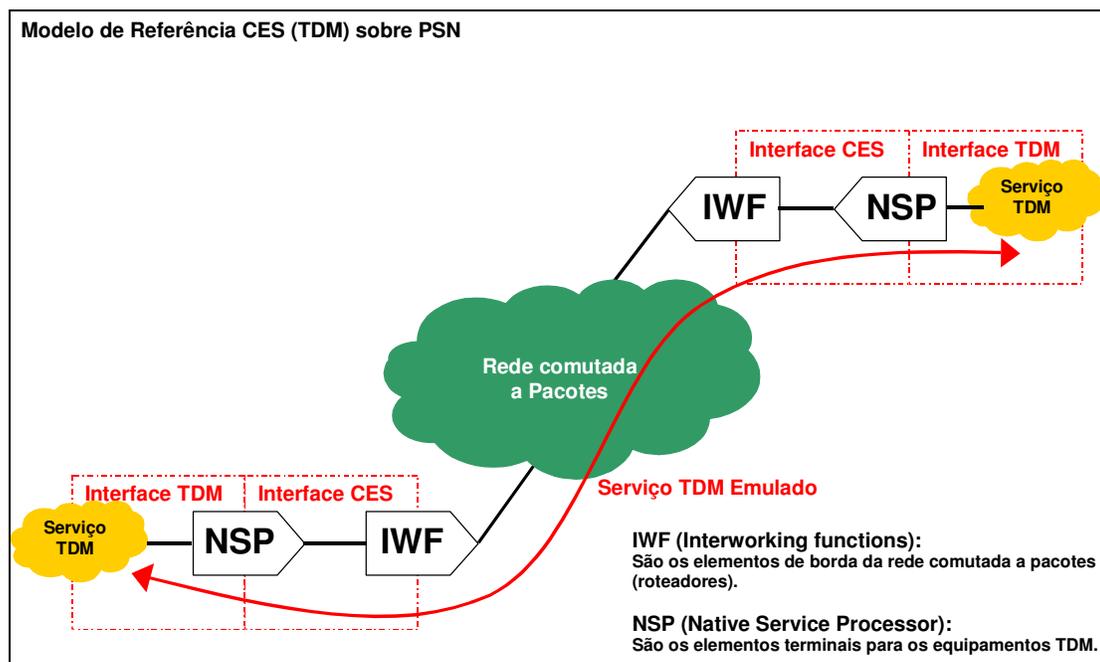


Figura 1.5 - Modelo de referência CES sobre PSN.

As Figuras 1.6 e 1.7 ilustram os fluxos de transporte e controle envolvidos em uma conversação entre dois telefones convencionais analógicos através de uma rede IP, do ponto de vista da Operadora de Telecomunicações. Elas ilustram claramente a diferença de

complexidade e de investimentos entre as duas abordagens apresentadas: revolucionária (serviço VoIP) e evolucionária (serviço TDMoIP<sup>®</sup>).

Na Figura 1.6 são apresentados os elementos de rede e protocolos necessários para o estabelecimento de uma chamada entre dois telefones convencionais da RTFC (Rede Telefônica Fixa Comutada) utilizando VoIP.

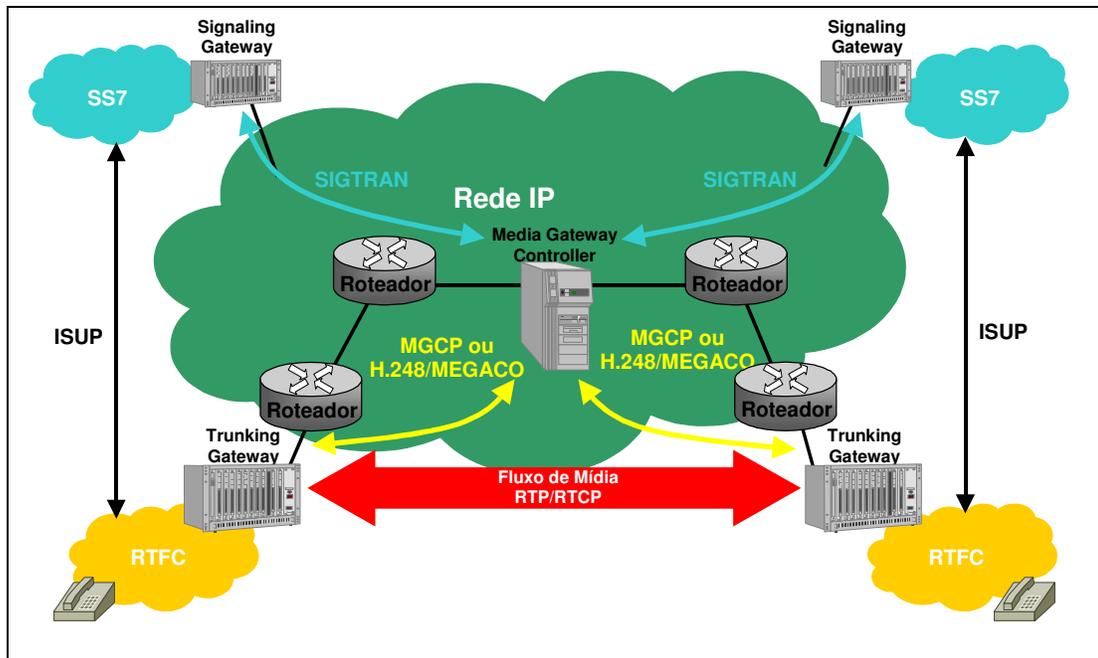


Figura 1.6 - Comunicação entre dois telefones convencionais usando VoIP.

Além da rede de transporte baseada no Protocolo IP e seus roteadores, responsáveis pelo encaminhamento dos pacotes IP, são necessários, pelo menos, os seguintes elementos [COLLINS2001]:

- Dois *Trunking Gateways* (TG), um conectado à RTFC de origem e outro à rede de destino, para realizar a conversão do tráfego de voz, codificado em PCM (*Pulse Code Modulation*) dentro de um canal do circuito TDM (feixe DS0), em um fluxo de pacotes IP correspondente, na origem, e a conversão inversa, no destino;
- Dois *Signaling Gateways* (SG), um conectado à rede de sinalização de origem e outro à rede de destino, para realizar a conversão das mensagens de sinalização SS7 do circuito TDM (canal 16 do feixe E1), em um fluxo de pacotes IP correspondente;
- Um *Media Gateways Controller* (MGC) ou *Softswitch*, conectado aos *Trunking Gateways* através da rede IP, para realizar o encaminhamento e acompanhamento da

chamada, controlando a troca de pacotes e o tipo de codificação utilizado entre os dois *Trunking Gateways*.

Para o encaminhamento, controle e sinalização da chamada, além do protocolo de rede IP (*Internet Protocol*) e dos protocolos de transporte UDP (*User Datagram Protocol*) e RTP/RTCP (*Real Time Protocol/Real Time Control Protocol*), responsáveis pela troca de pacotes em tempo real entre origem e destino, são necessários, pelo menos, mais dois protocolos adicionais dentro da rede:

- a) MGCP (*Media Gateway Control Protocol*) ou MEGACO/ H.248, para permitir ao *Softswitch* o controle dos *Trunking Gateways*;
- b) SIGTRAN (*SIGnaling TRANsport Protocol*) ou similar, para permitir o transporte das mensagens de sinalização dentro da rede IP de modo que possam ser interpretadas de forma adequada pelo *Softswitch* e pelos *Signaling Gateways*.

Na Figura 1.7 são apresentados os elementos de rede e protocolos necessários para o estabelecimento de uma chamada entre dois telefones convencionais da RTFC (Rede Telefônica Fixa Comutada) utilizando TDMoIP®.

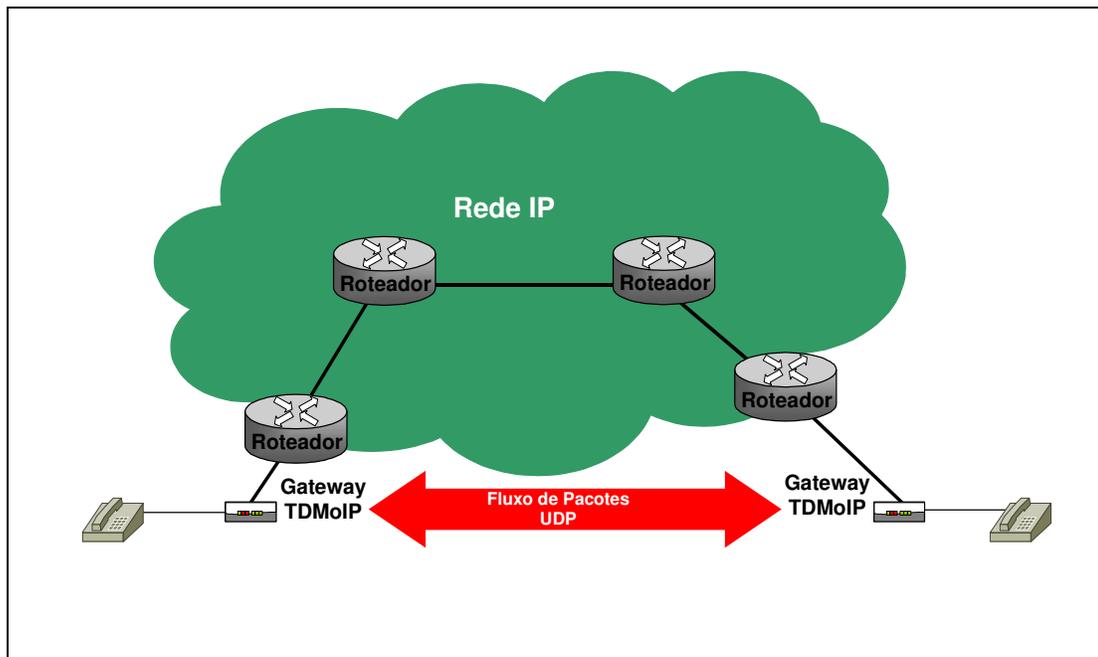


Figura 1.7 - Comunicação entre dois telefones convencionais usando TDMoIP.

Além da rede de transporte baseada no Protocolo IP e seus roteadores, são utilizados apenas dois equipamentos conectados aos terminais, denominados Multiplex (MUX) ou *Gateways* TDMoIP<sup>®</sup>, que fazem a conversão do tráfego de voz, codificado em PCM (realizando também essa codificação, no caso de conexão de um telefone analógico) dentro de um canal do circuito TDM (feixe DS0), em um fluxo de pacotes IP correspondente para o *Gateway* TDMoIP<sup>®</sup> localizado na outra extremidade da rede IP, estabelecendo um circuito ponto-a-ponto entre ambos, que corresponde ao circuito TDM emulado sobre a rede IP.

A diferença de complexidade entre ambas as tecnologias é evidente, caracterizando a natureza evolucionária da abordagem TDMoIP<sup>®</sup> quando comparada à tecnologia VoIP.

Evidentemente, uma outra opção é a solução integralmente VoIP, com a utilização de telefones IP ou *softphones* e protocolos como o SIP (*Session Initiation Protocol*) e o H.323 apresenta maior potencial, sobretudo em relação à economia de banda em função da flexibilidade de codificação da voz e serviços como videoconferência, mensagens integradas e multi-localização, mas implica necessariamente em investimentos na substituição dos equipamentos terminais, além de não apresentar a maturidade das redes TDM.

Considerando a situação de uma operadora entrante, que está construindo a sua rede e visando a conquistar novos clientes ou a situação de um novo provedor de serviços buscando nichos de mercado que rentabilizem seu negócio, a utilização da tecnologia VoIP dentro de uma rede completamente convergente para voz, dados e vídeo pode parecer mais interessante; mas para uma grande Operadora, cuja rede TDM implantada tem valor significativo, possuindo clientes consolidados onde o investimento em telecomunicações, através de PBX (*Private Branch eXchange*) de grande porte e equipamentos similares é significativo, a extensão de circuitos utilizando TDMoIP<sup>®</sup> pode ser uma forma bem mais segura e interessante de oferecer tarifas competitivas.

Adicionando a isso a expansão observada das redes metropolitanas MAN (*Metropolitan Area Networks*) e mesmo redes WAN (*Wide Area Networks*) baseadas na simplicidade e robustez da tecnologia *Ethernet* (*Metro Ethernet, Gigabit Ethernet e 10-Gigabit Ethernet*),

a interconexão direta de equipamentos TDM a essas redes, através de *gateways* TDMoIP<sup>®</sup>, pode ser um negócio bastante atrativo para os clientes finais, integrando suas redes de voz e dados de forma simples e sem grandes investimentos adicionais, substituindo integralmente todas as linhas dedicadas alugadas da Operadora tradicional, que apresentam tarifas elevadas e independentes da sua utilização efetiva e do volume de tráfego cursado, pelo acesso à rede de transporte em modo pacotes de um provedor independente, onde a largura de banda pode ser contratada de forma flexível adequada às reais necessidades de comunicação do cliente, com evidentes vantagens econômicas.

As tecnologias *Metro Ethernet* como rede de transporte metropolitana e TDMoIP<sup>®</sup> como interface de acesso podem assim ser consideradas complementares entre si, representando real possibilidade de competição dentro do mercado das linhas dedicadas ponto a ponto – as chamadas *leased lines* - para a transmissão de voz e/ou dados em grandes corporações, hoje praticamente um monopólio das Operadoras locais, as chamadas *incumbents*, favorecendo bastante a criação de redes e expansão de pontos de presença para Operadoras entrantes e Provedores Internet.

A disseminação do acesso em banda larga, que alcançou 4,7 milhões de linhas no Brasil em junho de 2006 segundo estudos da CISCO Systems [CISCO], ainda com forte predominância das tecnologias xDSL (*Digital Subscriber Lines*), é outro importante fator que pode alavancar a tecnologia TDMoIP<sup>®</sup> como alternativa ao serviço convencional dedicado e mesmo aos serviços VoIP, sobretudo para usuários empresariais de pequeno e médio porte que desejem preservar seus pequenos PBX, reduzindo os custos com telecomunicações sem investir muito ou se aventurar em soluções duvidosas oferecidas por pequenos provedores VoIP, com planos de negócio ainda não consolidados ou tecnologias de pouca maturidade.

Dessa forma, inclusive pela total eliminação dos dispendiosos circuitos dedicados interurbanos através do transporte pelas redes IP, os custos dessas linhas para os clientes finais tendem a cair muito nos próximos anos, favorecendo a formação de redes privadas de baixo custo baseadas em TDM.

Como exemplo disso, a Tabela 1.2 apresenta uma comparação entre o custo mensal de alguns circuitos interurbanos baseados em linhas dedicadas da operadora Brasil Telecom, representados pelos produtos da família BrT Link, com o serviço de conectividade IP com qualidade assegurada, o produto Vetor; e o serviço IP Dedicado de conexão, ambos também da Brasil Telecom, para velocidades equivalentes a um circuito E1 (2Mbps). Os valores utilizados correspondem aos preços máximos praticados, vigentes em novembro/2006, obtidos com a Gerência de Produtos e Serviços daquela operadora.

Tabela 1.2 – Custos mensais dos produtos Brasil Telecom para voz e dados.

<b>Produto</b>	<b>Circuito Brasília – Porto Alegre</b>	<b>Circuito Brasília - São Paulo</b>	<b>Circuito Brasília – Rio de Janeiro</b>	<b>Circuito Brasília – Belo Horizonte</b>	<b>Circuito Brasília – Cuiabá</b>
<b>VoiceLink G.703 30 canais</b>	R\$ 13.000,00	R\$ 10.787,40	R\$ 11.500,00	R\$ 11.500,00	R\$ 13.000,00
<b>DataLink 2 Mbps</b>	R\$ 23.507,91	R\$ 9.727,71	R\$ 9.484,51	R\$ 8.322,74	R\$ 24.061,95
<b>VideoLink 2 Mbps</b>	R\$ 15.500,00	R\$ 14.000,00	R\$ 15.000,00	R\$ 15.000,00	R\$ 15.500,00
<b>Vetor (MPLS) 2 Mbps Classe Dados*</b>	R\$ 7.605,02	R\$ 7.227,84	R\$ 7.529,87	R\$ 6.870,54	R\$ 7.671,16
<b>Vetor (MPLS) 2 Mbps Classe Dados Expressos*</b>	R\$ 9.489,21	R\$ 9.057,69	R\$ 9.428,67	R\$ 8.618,65	R\$ 9.569,96
<b>Vetor (MPLS) 2 Mbps Classe Multimídia**</b>	R\$ 12.116,27	R\$ 11.802,14	R\$ 12.051,05	R\$ 11.289,26	R\$ 11.030,45
<b>Vetor (MPLS) 2 Mbps Classe Voz**</b>	R\$ 9.063,20	R\$ 9.847,89	R\$ 10.126,93	R\$ 8.541,76	R\$ 10.268,23
<b>IP Dedicado (100% Banda assegurada) 2 Mbps</b>	R\$ 4.700,00	R\$ 4.700,00	R\$ 4.700,00	R\$ 4.700,00	R\$ 4.700,00

\* Inclui a locação mensal do roteador, no valor de R\$ 270,00.

\*\* Inclui a locação mensal do roteador para voz e dados, no valor de R\$ 3.500,00.

Dessa tabela, pode ser observado que a utilização do serviço IP Dedicado para o entroncamento de PBX, ao invés do VoiceLink tradicional, representaria uma redução de 60% nos custos de transporte. Da mesma forma, a utilização do Vetor (classe Dados) ou do próprio IP Dedicado, ao invés do VideoLink oferecido às grandes redes de televisão para o transporte de vídeo digital, representaria uma redução, respectivamente, de 50% ou 70% nos custos de transporte.

Considerando ainda a opção, bem mais arriscada, de utilização da Internet como rede de transporte, o custo mensal médio seria de R\$ 253,00, para acessos entre 2 e 8Mbps, conforme estudo apresentado em [CISCO2006] para os meses de abril a junho/2006, mais R\$ 100,00 com a locação do par de modems HDSL, ou seja, uma redução superior a 90% nos custos de transporte.

A alternativa TDMoIP<sup>®</sup> está disponível comercialmente desde 1999, quando sua primeira geração foi implantada pela operadora sueca UTFORS (adquirida em 2002 pela gigante norueguesa TELENOR) como solução para oferecimento de linhas dedicadas e serviços similares baseados em IP/*Ethernet* no mercado da Suécia. Nos anos seguintes, a tecnologia TDMoIP<sup>®</sup> foi implantada nos Estados Unidos, França, Inglaterra, Áustria, África, Rússia, Turquia, China e Austrália, possuindo mais de 40.000 portas E1/T1 nesses países segundo os dados apresentados por seu principal fabricante, a *RAD Data Communications*, responsável pelo registro da marca “*TDMoIP<sup>®</sup> driven*” e por uma linha de equipamentos para a construção de redes e serviços baseados nessa tecnologia.

A tecnologia TDMoIP<sup>®</sup> permite que pequenas empresas de telecomunicações, assim como provedores de serviços IP em geral, possam ser bastante competitivos na disputa com as grandes Operadoras pelo chamado Mercado Empresarial, constituído pelas pequenas e médias empresas; e pelos pequenos escritórios e o número crescente de usuários que trabalham em suas residências, conhecidos como SOHO (*Small Office/Home Office*). Segundo analistas, esse é o segmento do mercado de telecomunicações onde a competição é de fato acirrada, sendo o grande foco corrente das Operadoras devido ao potencial de rentabilidade existente, uma vez que o Mercado Corporativo encontra-se saturado e com baixíssimas margens, em virtude da política predatória de preços praticada ao longo dos últimos anos, quando as redes de dados viraram uma *commoditie*, e o atendimento ao Mercado Massa implica em vultosos investimentos em capilaridade, o que praticamente garante o seu completo domínio pelas grandes Operadoras, restringindo-se a competição ao pequeno universo de clientes mais rentáveis e concentrados nos grandes aglomerados urbanos.

Outra grande possibilidade da alternativa TDMoIP<sup>®</sup> é a redução de custos operacionais para as Operadoras móveis, sobretudo aquelas que operam de forma independente, uma

vez que o uso de linhas dedicadas alugadas, tipicamente feixes T1 (1,544 Mbps, USA, Japão) ou E1 (2,048 Mbps, Brasil, Europa) para a interligação de suas estações rádio-base às centrais de comutação e controle respondem por grande parte de seus custos de rede. Em áreas metropolitanas cobertas por redes de transporte comutadas em modo pacotes, como *Metro Ethernet* e *Gigabit Ethernet*, a substituição dessas linhas dedicadas por *gateways* TDMoIP<sup>®</sup> para a interconexão das estações rádio-base pode ser relativamente simples e economicamente muito vantajosa.

Considerando todo esse cenário, o objetivo deste trabalho é aprofundar o estudo sobre a alternativa TDMoIP<sup>®</sup> e suas abordagens similares dentro do conceito de emulação de circuitos sobre redes de transporte em modo pacotes (CESoPSN), em especial os circuitos TDM sobre redes IP, com a finalidade de propor, discutir e validar essa tecnologia como uma possível alternativa de baixo custo para *broadcasting* de conteúdo de vídeo/áudio integrado a baixas taxas de transmissão, tipicamente até 1,5Mbps, permitindo a sua produção e distribuição independente para fins sociais, educacionais e de entretenimento, possibilitando, quiçá, dois grandes desdobramentos:

- a) A disseminação de canais de serviços IPTV (*Internet Protocol TeleVision*) com baixo custo de transmissão e reduzido investimento em equipamentos, utilizando o mesmo conceito das chamadas TVs corporativas, existentes nas grandes empresas, a fim de levar conteúdo customizado e interativo a escolas, programas sociais de inclusão digital, ONGs (Organizações Não Governamentais) que promovam educação e cidadania e outras instituições de caráter educacional e social que desejem utilizar esses conteúdos em seus programas de atendimento às comunidades;
- b) A criação de uma rede nacional, interligada pela Internet ou outra rede capaz de prover conectividade IP com maior disponibilidade de largura de banda, de emissoras locais de TV para distribuição desse conteúdo, via interface de rádio convencional VHF (*Very High Frequency*), com modulação analógica VSB (*Vestigial-Sided Band*) e canais de 6MHz, compatíveis com a totalidade dos receptores existentes no Brasil e em boa parte do mundo, com foco nas comunidades de baixa renda e nas regiões menos atendidas, contribuindo para a democratização de acesso à informação disponível através da Internet ou gerada de forma livre por produtores independentes, permitindo que esse conteúdo chegue aos usuários finais desejados sem a necessidade de competição por espaço na programação das grandes redes de TV aberta do país;

Para tanto, foram definidas duas grandes Metas para direcionamento desse estudo, a fim de delimitar o escopo a ser abordado na dissertação e definir ferramentas, cenários e critérios de análise utilizados na implementação e avaliação dos resultados experimentais:

- a) Demonstrar que a tecnologia TDMoIP<sup>®</sup> com sincronização adaptativa é mais barata que PDH/SDH dedicado e mais simples e eficiente que as principais tecnologias VoIP (*Voice over IP*) utilizando o protocolo RTP, no transporte fim a fim de troncos de voz TDM convencionais;
- b) Validar (ou não) a tecnologia TDMoIP<sup>®</sup> com sincronização adaptativa pelo receptor como alternativa simples e barata ao ATM/SDH para transporte de TV digital de baixa definição, em qualidade compatível com a TV analógica atual, e para *broadcasting* IPTV a baixas taxas (até 1,5Mbps).

Com esses objetivos, o trabalho está organizado da seguinte forma:

No capítulo 2 é aprofundado o conceito de emulação de circuitos, sendo propostos novos mecanismos para melhorar a robustez das soluções existentes para os desafios enfrentados: o seqüenciamento dos pacotes e a recuperação do relógio de transmissão no receptor. Na seqüência, é feita uma comparação entre algumas características da emulação TDM e VoIP, encerrando com aplicações da tecnologia TDM sobre IP.

No capítulo 3 é abordada a transmissão de vídeo digital, com seus padrões de codificação, introduzindo em seguida o conceito de IPTV e suas principais tecnologias, para então propor as adequações necessárias à abordagem TDM sobre IP a fim de permitir sua utilização como alternativa para *broadcasting* de serviços IPTV similares.

No capítulo 4 é realizada a validação dos novos mecanismos propostos, através da simulação dos algoritmos e sua implementação em uma aplicação simples para TDM sobre IP, utilizando Java. Além disso, é feita a experimentação de uma solução comercial para essa tecnologia, analisando seu desempenho e comparando-o com a solução proposta.

Finalmente, no capítulo 5, são feitas as conclusões do trabalho, com base nos resultados de validação e experimentação, apresentando também propostas para trabalhos futuros.



## **2 - EMULAÇÃO DE CIRCUITOS TDM EM REDES IP**

### **2.1 - A EMULAÇÃO DE CIRCUITOS EM REDES DE PACOTES**

O tráfego mundial de telecomunicações, em especial o de telefonia, é convencionalmente transportado através de circuitos dedicados, que consistem em enlaces bidirecionais orientados a conexão, operando de forma síncrona ou plesiócrona, e organizados em agrupamentos de diferentes níveis, denominados hierarquias. Essas hierarquias são conhecidas por redes ou serviços TDM (Time-Divison Multiplexing), em função da característica de compartilhamento do meio de transmissão utilizada, ou seja a multiplexação por divisão de tempo, onde cada canal é transmitido dentro de um intervalo de tempo (*timeslot*) pré-definido e fixo durante a conexão.

A emulação de circuitos, ou CES (*Circuit Emulation Services*) é uma tecnologia que permite o transporte de serviços TDM, tais como feixes E1/T1/E3/T3 da hierarquia digital plesiócrona, ou PDH (*Plesiochronous Digital Hierarchy*) e circuitos SONET/SDH (*Synchronous Optical Network/Synchronous Digital Hierarchy*), através de redes de transporte comutadas em modo pacotes, ou PSN (*Packet-Switching Networks*). A CES foi originalmente concebida na década de 1990, dentro do chamado “Mundo” ATM (*Assynchronous Transfer Mode*) [ATM1997]. A idéia foi trazida para dentro das redes comutadas a pacotes por uma série de organismos internacionais ligados às redes de telecomunicações, como a IETF (*Internet Engineering Task Force*), o MEF (*Metro Ethernet Forum*) e o MPLS (*Multi-Protocol Label Switching*) Fórum.

Os principais padrões CES estão sendo definidos pelo grupo de trabalho PWE3 (*Pseudo-Wire Emulation Edge-to-Edge*), constituído pela IETF em 2001, com o objetivo de desenvolver métodos para transportar serviços das camadas 1 e 2 através de redes comutadas em modo pacotes, em especial IP e MPLS. Como principal trabalho desse grupo foram geradas três RFCs (*Request for Comments*) básicas, documentos normativos publicados pela IETF, estabelecendo os Requisitos para PWE3 [RFC3916], a Arquitetura para PWE3 [RFC3985] e os Requisitos Específicos para Emulação de circuitos TDM

através de PWE3 sobre redes comutadas em modo pacotes [RFC4197]; além de quatro RFCs abordando questões complementares, como o estabelecimento e manutenção de pseudo-circuitos através do protocolo LDP (*Label Distribution Protocol*) [RFC4447], métodos de encapsulamento para o transporte *Ethernet* sobre redes MPLS [RFC4448], alocações IANA (*Internet Assigned Numbers Authority*) para PWE3 [RFC4446] e definição da palavra de controle PWE3 para uso sobre redes MPLS [RFC4385]; bem como alguns *Internet-Drafts* associados [IETF2006] [IETF2006a] [IETF2006b], já em chamada final para publicação como RFCs, que vêm orientando a maioria dos estudos sobre o assunto e serão descritos na seqüência do trabalho.

### **2.1.1 - Os requisitos da emulação PWE3 de circuitos TDM sobre PSN**

A RFC4197 define requisitos específicos para a Emulação de Pseudo-Circuitos Fim a Fim (PWE3) no transporte de sinais digitais multiplexados TDM, tanto da Hierarquia Digital Plesiócrona (PDH) como da Hierarquia Digital Síncrona/Rede Óptica Síncrona (SDH/SONET), estabelecidos sobre redes de transporte comutadas em modo pacotes (PSN), de forma alinhada com a Arquitetura PWE3 [RFC3985], descrita no Apêndice A e fazendo referência aos requisitos aplicáveis à PWE3 [RFC3916], listados no Apêndice B.

Esses requisitos específicos decorrem das características particulares dos serviços TDM:

- Necessidade de sincronização entre os relógios dos Circuitos de Acesso (AC) de origem e destino em cada uma das direções do pseudo-circuito;
- Necessidade de manter o *jitter* e a flutuação (*wander*) do relógio, para o fluxo de dados entregue ao AC de destino, dentro dos limites impostos pelos documentos normativos, na presença das grandes variações no atraso dos pacotes produzidas pela PSN.

Além de características intrínsecas às aplicações que utilizam circuitos TDM, como, por exemplo, aplicações de voz:

- Necessidade de especial ênfase na redução dos atrasos de transmissão;
- Relativa tolerância a erros nos dados transmitidos.

Que são diferentes das características de outras aplicações, como, por exemplo, o transporte de informações de sinalização:

- Relativa tolerância aos atrasos de transmissão;
- Extrema sensibilidade a erros nos dados transmitidos.

## 2.1.2 - Modelo de referência PWE3 para sincronização de rede

A Figura 2.1 apresenta um Modelo de Referência genérico para a sincronização de redes dentro da Arquitetura PWE3, envolvendo a seguinte notação:

- CE1, CE2** são os Equipamentos Clientes onde terminam os circuitos TDM emulados;
- PE1, PE2** são os Equipamentos Provedores que adaptam o serviço nativo TDM ao pseudo-circuito;
- S1, S2** são os roteadores de núcleo da rede comutada em modo pacotes;
- Phy** é a interface física onde termina o circuito TDM;
- Enc** é a interface no PE de origem do pseudo-circuito, onde o encapsulamento dos fluxos de dados é realizado;
- Dec** é a interface no PE de destino do pseudo-circuito, onde o desencapsulamento dos fluxos de dados é realizado. Essa interface contém um *buffer* de compensação, também conhecido como *jitter buffer*, de tamanho limitado;
- =====>** são os Circuitos de Acesso (AC) para os circuitos TDM;
- >** são os Pseudo-Circuitos (PW) provendo emulação fim a fim para os circuitos TDM.
- Ck<sub>CE1</sub>** é o relógio local (oscilador) disponível e utilizado pelo CE1 para transmissão do fluxo de dados TDM através do circuito de acesso em direção a CE2;
- Ck(R)<sub>CE1</sub>** é o relógio recuperado por PE1 a partir do fluxo de dados TDM que chega pelo circuito de acesso. **Ck<sub>CE1</sub>** e **Ck(R)<sub>CE1</sub>** têm sempre a mesma frequência;
- Ck(Rp)<sub>PE2</sub>** é o relógio utilizado pelo PE2 para transmissão do fluxo de dados TDM através do circuito de acesso em direção a CE2, recuperado a partir dos pacotes recebidos através do pseudo-circuito;
- Ck(RpR)<sub>PE2</sub>** é o relógio novamente recuperado por CE2 a partir do fluxo de dados TDM que chega de PE2 pelo circuito de acesso. **Ck(Rp)<sub>PE2</sub>** e **Ck(RpR)<sub>PE2</sub>** têm sempre a mesma frequência;
- Ck<sub>CE2</sub>** é o relógio local (oscilador) disponível utilizado pelo CE2 para transmissão do fluxo de dados TDM através do circuito de acesso em direção a CE1;

- $Ck(R)_{CE2}$  é o relógio recuperado por PE2 a partir do fluxo de dados TDM que chega pelo circuito de acesso.  $Ck_{CE2}$  e  $Ck(R)_{CE2}$  têm sempre a mesma frequência;
- $Ck(Rp)_{PE1}$  é o relógio utilizado pelo PE1 para transmissão do fluxo de dados TDM através do circuito de acesso em direção a CE1, recuperado a partir dos pacotes recebidos através do pseudo-circuito;
- $Ck(RpR)_{PE1}$  é o relógio novamente recuperado por CE1 a partir do fluxo de dados TDM que chega de PE1 pelo circuito de acesso.  $Ck(Rp)_{PE1}$  e  $Ck(RpR)_{PE1}$  têm sempre a mesma frequência;
- $Ck_{PE1}$  é o relógio (oscilador) local disponível em PE1;
- $Ck_{PE2}$  é o relógio (oscilador) local disponível em PE2;
- $Ck_{PSN}$  é o relógio comum de referência da rede PSN, se existente, disponível em PE1 e PE2;
- $Ck_{TDM}$  é o relógio comum de referência da rede TDM, se existente, disponível em CE1 e CE2, representando um caso particular, pois diferentes pares de Equipamentos Cliente podem utilizar diferentes relógios de referência, sobretudo se pertencem a redes TDM distintas.

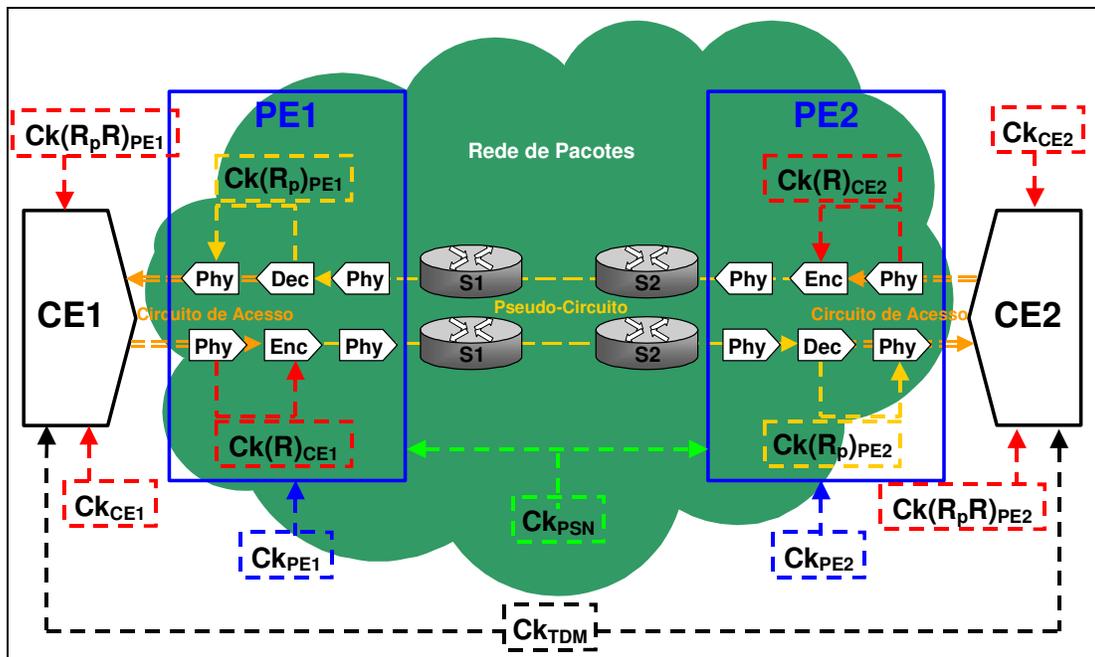


Figura 2.1 - Modelo de referência PWE3 para sincronização de rede.

O requisito básico da emulação fim a fim de um circuito TDM é que tanto os relógios  $Ck(R)_{CE1}$  e  $Ck(Rp)_{PE2}$ , como os relógios  $Ck(R)_{CE2}$  e  $Ck(Rp)_{PE1}$  estejam na mesma

frequência. O método mais adequado para garantir isso depende do esquema de sincronização da rede. Podem ser considerados os seguintes cenários de sincronização:

**a) Cenários de Sincronização através de Redes Síncronas:**

Dependendo de qual parte da rede é sincronizada através de um relógio comum, existem dois cenários distintos:

**a.1) Redes Sincronizadas pelo Equipamento Provedor (PE):**

A Figura 2.2 é a versão do Modelo de Referência para sincronização da rede, apresentando o cenário de sincronização pelo PE:

- O relógio comum de referência da rede PSN  $Ck_{PSN}$  está disponível para todos os equipamentos PE, e os osciladores locais  $Ck_{PE1}$  e  $Ck_{PE2}$  são fixados em  $Ck_{PSN}$ .
- Os relógios  $Ck(Rp)_{PE2}$  e  $Ck(Rp)_{PE1}$  são os mesmos que  $Ck_{PE2}$  e  $Ck_{PE1}$ , respectivamente;
- Os relógios  $Ck_{CE1}$  e  $Ck_{CE2}$  são os mesmos que  $Ck(RpR)_{PE1}$  e  $Ck(RpR)_{PE2}$ , respectivamente, ou seja, CE1 e CE2 utilizam um enlace fechado para sincronização.

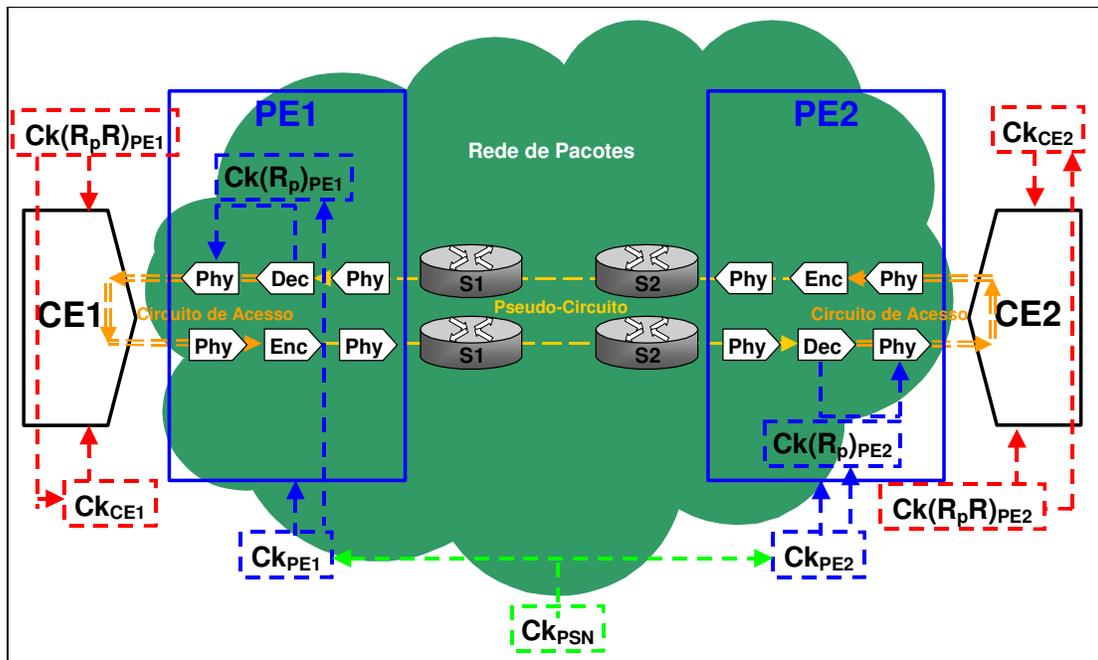


Figura 2.2 - Cenário de rede sincronizada pelo PE.

**a.2) Redes Sincronizadas pelo Equipamento Cliente (CE):**

A Figura 2.3 é a versão do Modelo de Referência para sincronização da rede, apresentando o cenário de sincronização pelo CE:

- O relógio comum de referência da rede TDM  $Ck_{TDM}$  está disponível para todos os equipamentos CE, e os osciladores locais  $Ck_{CE1}$  e  $Ck_{CE2}$  são fixados em  $Ck_{TDM}$

- Os relógios  $Ck(Rp)_{PE2}$  e  $Ck(Rp)_{PE1}$  são os mesmos que  $Ck_{CE2}$  e  $Ck_{CE1}$ , respectivamente, ou seja, PE1 e PE2 utilizam um enlace fechado para sincronização.

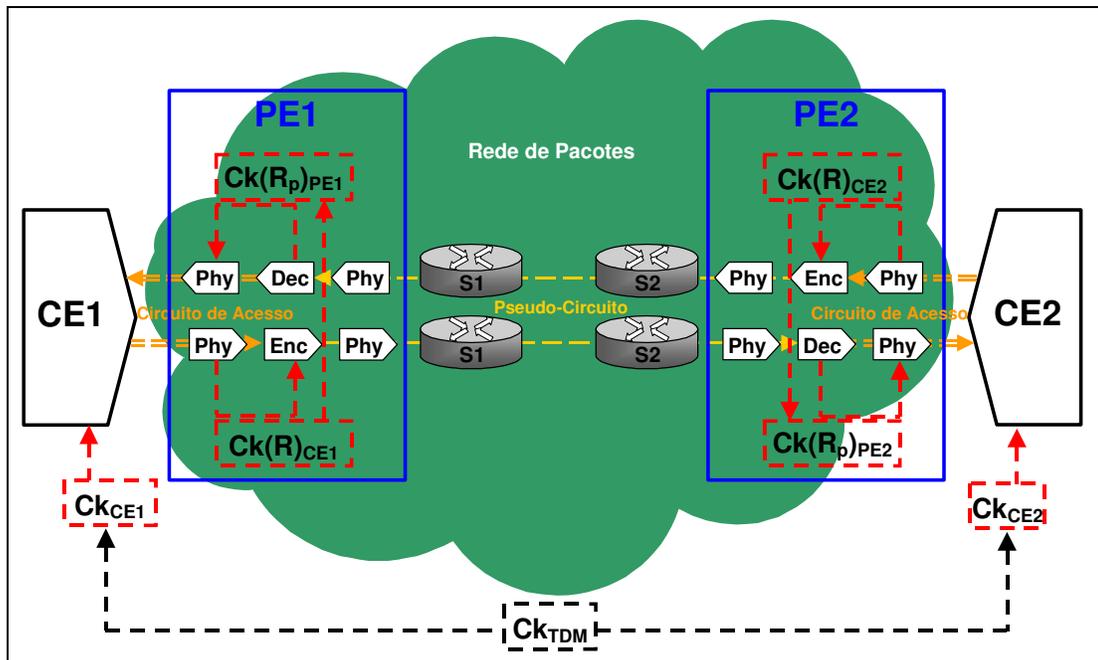


Figura 2.3 - Cenário de rede sincronizada pelo CE.

Nenhuma informação de sincronização precisa ser transmitida através do pseudo-circuito para esses dois casos.

#### b) Cenário de Sincronização Relativa:

Neste caso, cada CE utiliza a sua própria fonte de relógio de transmissão, que é transportada através da PSN e recuperada pelo respectivo PE remoto. A Figura 2.4 é a versão do Modelo de Referência para a sincronização relativa da rede nesse cenário:

- O relógio comum de referência da rede PSN  $Ck_{PSN}$  está disponível para todos os equipamentos PE, e os osciladores locais  $Ck_{PE1}$  e  $Ck_{PE2}$  são fixados em  $Ck_{PSN}$ .
- Os relógios  $Ck_{CE1}$  e  $Ck_{CE2}$  são gerados localmente, sem referência a um relógio comum;
- Os relógios  $Ck(Rp)_{PE2}$  e  $Ck(Rp)_{PE1}$  são os mesmos que  $Ck_{PE2}$  e  $Ck_{PE1}$ , respectivamente, que por sua vez são gerados com referência ao relógio comum disponível para todos os equipamentos PE,  $Ck_{PSN}$ ;

Em uma pequena modificação desse cenário, um – e apenas um - dos equipamentos CE pode utilizar o relógio recuperado a partir do fluxo de dados TDM que chega do respectivo PE pelo circuito de acesso  $Ck(R_pR)_{PE2}$ , por exemplo, como seu relógio de transmissão  $Ck_{CE2}$ , no exemplo, utilizando um enlace fechado para sincronização.

Neste caso, a informação de sincronização, correspondente à diferença entre o relógio de referência comum  $Ck_{PSN}$  e o relógio independente  $Ck_{CE1}$ , por exemplo, precisa ser explicitamente transferida do PE de origem para o PE de destino.

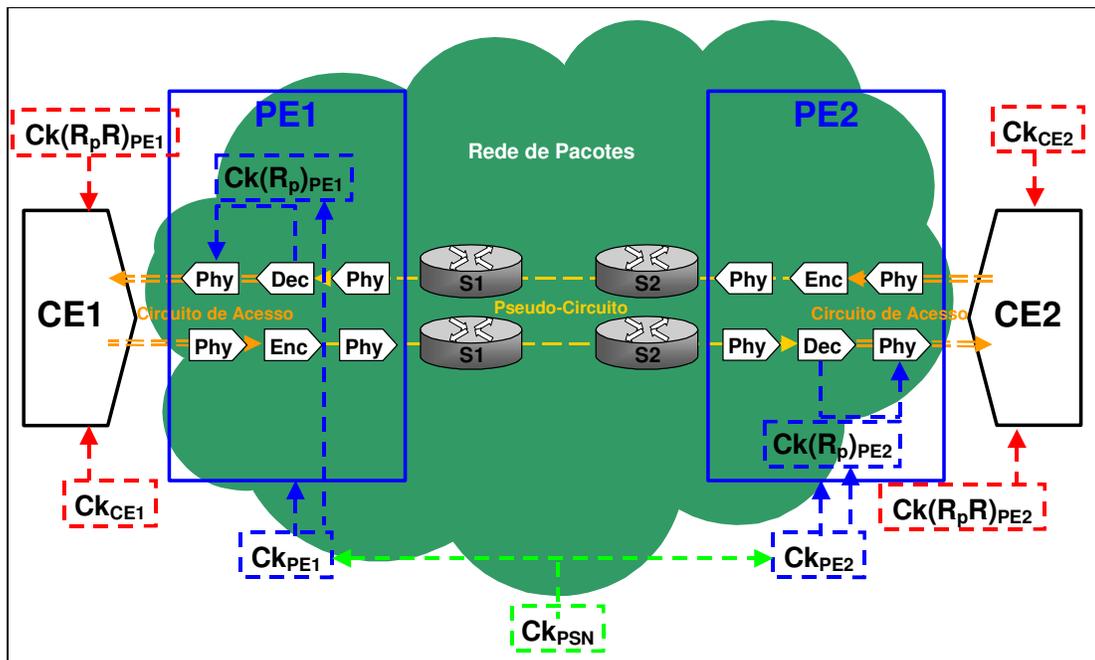


Figura 2.4 - Cenário de sincronização relativa.

### c) Cenário de Sincronização Adaptativa:

O cenário de sincronização adaptativa é caracterizado pela completa independência entre os relógios de origem e destino, tanto dos Equipamentos Clientes (CEs) e circuitos TDM nativos como dos Equipamentos Provedores (PEs) e a rede PSN, tornando a sincronização entre o relógio de transmissão utilizado pelo CE de origem e o relógio recuperado pelo PE de destino muito mais difícil que nos outros cenários. A Figura 2.5 é a versão do Modelo de Referência para a sincronização adaptativa da rede nesse cenário:

- Nenhum relógio comum de referência da PSN  $Ck_{PSN}$  está disponível para PE1 e PE2;
- Nenhum relógio comum de referência TDM  $Ck_{TDM}$  está disponível para CE1 e CE2.

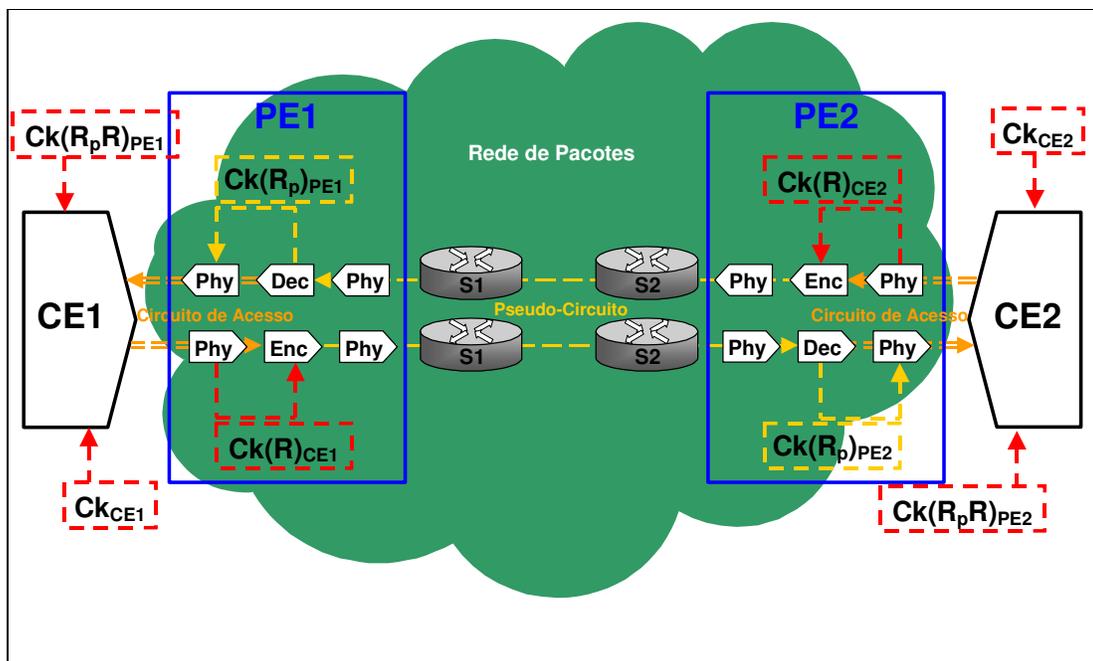


Figura 2.5 - Cenário de sincronização adaptativa.

Deve ser observado que a tolerância entre os relógios  $Ck_{CE1}$  e  $Ck(Rp)_{PE2}$  e entre os relógios  $Ck_{CE2}$  e  $Ck(Rp)_{PE1}$  deve ser pequena o bastante para assegurar que a *jitter buffer* não tenha sua capacidade esgotada (*overflow*) ou seja rapidamente esvaziado (*underflow*), causando a perda de continuidade do fluxo de dados TDM.

Neste caso, a informação de sincronização, correspondente à diferença entre o relógio de transmissão  $Ck_{CE1}$ , por exemplo, e o relógio recuperado a partir dos pacotes  $Ck(Rp)_{PE2}$ , no mesmo exemplo, pode ser explicitamente transferida do PE de origem para o PE de destino através do protocolo RTP ou mecanismo similar.

### 2.1.3 - As diversas propostas para emulação de circuitos TDM sobre PSN

Os esforços de padronização das tecnologias para Emulação de Serviços TDM sobre redes de transporte com comutação em modo pacotes vem sendo desenvolvidos em diversos fóruns técnicos e organismos internacionais de padronização. Na IETF, isso está sendo conduzido pelo Grupo de Trabalho PWE3, que aborda a emulação de serviços fim a fim através de pseudo-circuitos, descrito anteriormente e fonte dos principais documentos abordados. No ITU-T, o Grupo de Estudo 13 (SG13) conduz o assunto, tendo desenvolvido a Recomendação ITU-T Y.1414 para serviços de voz sobre MPLS [ITU-T Y1414] e a

Recomendação ITU-T Y.1413 sobre a interoperabilidade entre redes TDM e MPLS [ITU-T Y1413]. Esse assunto também foi discutido pelo *MPLS Forum* [STEIN2003].

Dessa forma, diversos documentos têm sido gerados e revisados, caracterizando diversas propostas distintas, cada uma abordando aspectos particulares do transporte TDM sobre as redes de pacotes, conforme mostrado na Figura 2.6.

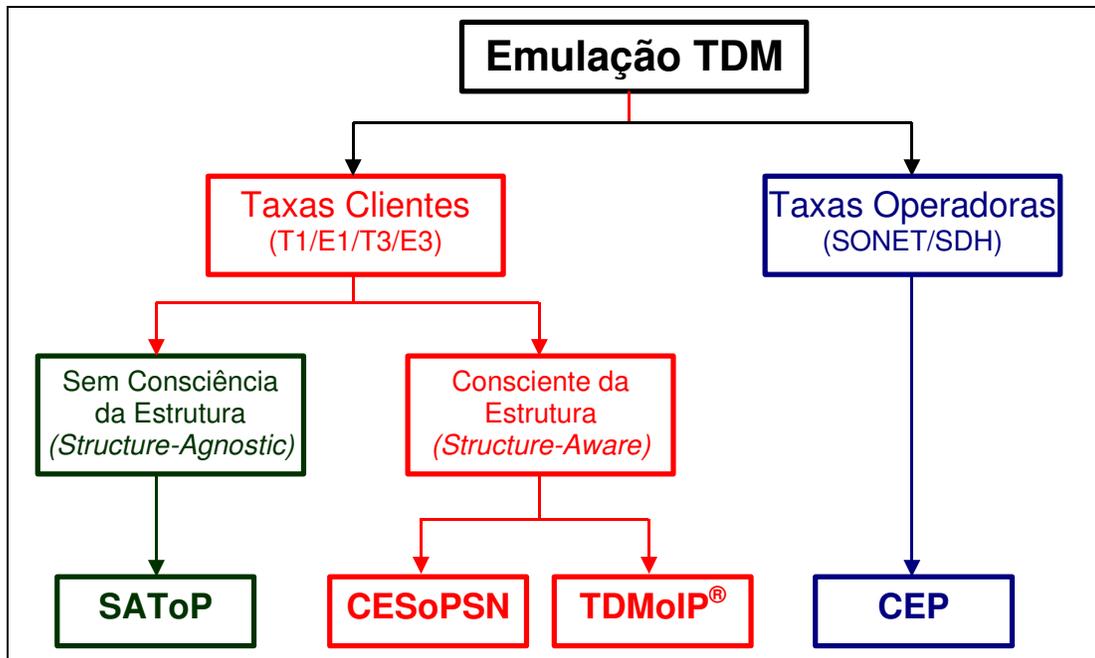


Figura 2.6 - Correntes da emulação TDM sobre redes de pacote.

Existem atualmente duas linhas de estudo dentro do grupo IETF PWE3, uma focada no transporte TDM com elevadas taxas de transmissão, compatíveis com os *backbones* das grandes Operadoras, denominado **CEP** (*SONET/SDH Circuit Emulation over Packet*), ou emulação de circuitos SONET/SDH sobre pacotes, definido pelo respectivo *Internet-Draft* [IETF2006b]; e outra com foco no transporte TDM com taxas mais baixas, utilizadas pelos clientes dessas Operadoras, por sua vez subdividida em três propostas distintas:

- **SAToP** (*Structure-Agnostic TDM over Packet*), ou transporte de serviços TDM sobre pacotes sem consciência da estrutura, definido pelo respectivo *Internet-Draft* [RFC4553], que encontra-se na fila para publicação como RFC pela IETF ainda em 2006.

- **CESoPSN** (*Structure-aware TDM Circuit Emulation Service over Packet Switched Network*), ou serviço de emulação de circuitos TDM, consciente da estrutura, sobre rede de pacotes, definido pelo respectivo *Internet-Draft* [IETF2006a].
- **TDMoIP<sup>®</sup>** (*TDM over IP*), ou transporte consciente da estrutura de serviços TDM sobre redes IP, posteriormente generalizada para outras redes, como MPLS, L2TPoIP e Ethernet, criando o termo **TDMoPW** (*TDM over Pseudo-Wire*), ou transporte de serviços TDM sobre pseudo-circuitos, definido pelo respectivo *Internet-Draft* [IETF2006]. A nomenclatura original TDMoIP<sup>®</sup> vem sendo mantida nos documentos por consistência histórica, mas é hoje marca registrada da RAD Data Communications, que cedeu à IETF os direitos perpétuos de reprodução para fins de desenvolvimento da padronização pelo grupo PWE3.

O SAToP é um protocolo para transporte de serviços TDM sobre pseudo-circuitos que trata os dados TDM como um fluxo arbitrário de bits, dividido em pacotes com tamanho fixo e bem definido, conforme configurado entre os PEs de origem e destino, desprezando completamente qualquer estrutura que possa existir dentro dele. Dessa forma, esse protocolo é adequado para o transporte de fluxos TDM efetivamente não estruturados, como canais de vídeo codificados de forma digital; e para o transporte de fluxos TDM estruturados onde não existe necessidade de preservar a integridade da estrutura, interpretar ou manipular canais individuais durante esse transporte.

A abordagem SAToP é adequada para PSNs que apresentam perda de pacotes em níveis desprezíveis e aplicações que não requerem discriminação entre canais e nem intervenção na sinalização TDM. Quando um pacote SAToP é perdido, um padrão “todos os bits iguais a um” é enviado à interface TDM, o qual é interpretado pelo terminal TDM de destino como uma indicação AIS, que imediatamente dispara uma condição de falha para o circuito. Essa condição é admissível por apenas 0,2% do tempo de utilização do circuito, conforme estabelecido pela Recomendação ITU-T G.826, de forma que o protocolo SAToP, apesar de encontrar-se em estágio mais avançado de padronização em virtude da publicação iminente da respectiva RFC, tem sua aplicação limitada às redes de pacotes extremamente confiáveis e dotadas de disponibilidade excedente de largura de banda.

Como essas condições não podem ser assumidas de maneira geral para as redes de pacotes, o transporte TDM consciente de estrutura pode ser uma alternativa mais adequada para muitas aplicações, uma vez que nesse caso é possível preservar explicitamente a estrutura TDM durante o transporte sobre a PSN, tornando possível a mitigação efetiva das eventuais perdas de pacotes.

O transporte consciente de estrutura leva em consideração pelo menos algum elemento da estrutura TDM para aumentar a sua robustez em relação às perdas de pacote e outros elementos não-determinísticos associados à PSN. Os pseudo-circuitos TDM conscientes de estrutura não necessitam transportar o *overhead* dessa estrutura através da rede de pacotes, permitindo a eliminação de elementos como o FAS, que pode ser regenerado no PE de destino. Outras vantagens dessa abordagem são a separação dos *timeslots* individuais relativos a cada canal, permitindo a utilização de técnicas avançadas para a mitigação de perdas de pacote; a identificação clara da sinalização TDM, simplificando os mecanismos para a sua utilização e manutenção; e finalmente a conservação de largura de banda, através de mecanismos para detecção de atividade de voz e/ou interpretação da sinalização, evitando a transmissão de *timeslots* ociosos.

Existem três métodos para assegurar a integridade fim a fim da estrutura TDM, cada um com as suas vantagens e características específicas:

- **Fixação de estrutura (*structure-locking*)** – necessita que cada pacote seja montado a partir do começo de uma determinada estrutura TDM, seja ela um canal, um quadro ou um multi-quadro, contendo sempre uma ou mais dessas estruturas completas. É o método definido pela abordagem CESoPSN, que pressupõe a transmissão independente de cada canal Nx64Kbps, de forma a permitir a sua alocação dinâmica, otimizando a utilização de largura de banda da rede e recompondo o fluxo TDM no PE de destino.
- **Indicação de estrutura (*structure-indication*)** – permite que os pacotes contêm fragmentos arbitrários das estruturas básicas, mas emprega ponteiros para indicar onde cada estrutura começa. É o método definido pela abordagem TDMoIP<sup>®</sup> quando os canais podem ser alocados de forma estática e/ou é necessária a interoperabilidade com circuitos emulados baseados em AAL1 (*ATM Adaptation Layer number 1*), uma vez que representa um fluxo de dados do tipo CBR, com taxa de transmissão constante.

- **Regeneração de estrutura (*structure-reassembly*)** – faz sentido apenas para TDM canalizado, onde o PE de origem extrai e armazena os canais individuais, encapsula esses canais em pacotes e os transmite através da rede; enquanto o PE de destino regenera a estrutura original com base nos pacotes recebidos, introduzindo o *overhead* necessário. É o método definido pela abordagem TDMoIP<sup>®</sup> quando é desejada a alocação dinâmica de canais e/ou é necessária a interoperabilidade com circuitos emulados baseados em AAL2 (*ATM Adaptation Layer number 2*), uma vez que representa um fluxo de dados do tipo VBR, ou seja, com taxa de transmissão variável.

Apesar do nome histórico, o protocolo em desenvolvimento segundo a proposta TDMoIP<sup>®</sup> pode operar sobre diversos tipos de redes de transporte comutadas em modo pacotes, incluindo UDP (User Datagram Protocol) sobre IPv4 (*Internet Protocol version 4*) ou IPv6 (*Internet Protocol version 6*), MPLS (*Multi-Protocol Label Switching*), L2TPv3 (*Layer 2 Tunneling Protocol version 3*) sobre IP, ou mesmo *Ethernet* pura, cada qual representando uma das implementações particulares definidas no respectivo *Internet-Draft* [IETF2006].

Em função de suas características mais aderentes aos objetivos propostos, ou seja, o transporte transparente de fluxos TDM utilizando redes IP como *backbone* para transmissão de aplicações de vídeo envolvendo taxas até 34Mbps (E3), serão aprofundadas neste trabalho apenas as propostas SAToP, por ser a alternativa mais simples para o transporte não consciente de estrutura através de redes homogêneas com garantia de QoS, e TDMoIP<sup>®</sup>, por ser uma alternativa mais abrangente e robusta, permitindo o transporte através de redes menos confiáveis, em função da camada de adaptação e da possibilidade de utilizar sincronização adaptativa.

A Tabela 2.1 apresenta um resumo das principais características de cada uma dessas propostas para o transporte de circuitos TDM sobre redes comutadas em modo pacotes:

Tabela 2.1 – Comparação entre as propostas para transporte TDM sobre PSN.

	<b>SAToP</b>	<b>CESoPSN</b>	<b>TDMoIP/ CBR</b>	<b>TDMoIP/ VBR</b>	<b>CEP</b>
<b>Taxas de Transmissão</b>	T1/E1 e T3/E3	N x 64kbps	T1/E1 e T3/E3	N x 64kbps	STM-N N=1,4,16,64
<b>Consciência da Estrutura</b>	Não	Sim	Sim	Sim	Sim
<b>Adaptação do Fluxo de Dados</b>	Inexistente	Inexistente	AAL1	AAL2	Dependente do Serviço
<b>Alinhamento de Quadros</b>	Inexistente	Fixação de Estrutura	Indicação de Estrutura	Regeneração de Estrutura	Indicação de Estrutura
<b>Tratamento de Perdas de Pacote</b>	Padrão de “uns”	Permite Mitigação	Permite Mitigação	Permite Mitigação	Sinalização de Erro
<b>Palavra de Controle</b>	32 bits	32 bits	32 bits	32 bits	Mínimo 64 bits
<b>Número de Seqüência</b>	16 bits sem sinal	16 bits sem sinal	16 bits sem sinal	16 bits sem sinal	16 bits sem sinal
<b>Utilização Protocolo RTP</b>	Opcional (Se relógio comum PEs)	Opcional (Para relógio comum PEs)	Opcional (Para relógio comum PEs)	Opcional (Para relógio comum PEs)	Opcional
<b>Redes de Pacotes Compatíveis</b>	UDP/IPv4-6 L2TPv3 MPLS	UDP/IPv4-6 L2TPv3 MPLS	UDP/IPv4-6 L2TPv3 MPLS Ethernet	UDP/IPv4-6 L2TPv3 MPLS Ethernet	UDP/IPv4-6 L2TPv3 MPLS
<b>Tamanho por Pacote (E1)</b>	<i>Overhead</i> + N x 256 bytes	<i>Overhead</i> + N x 32 bytes (1ms a 5ms)	<i>Overhead</i> + N x 48 bytes	<i>Overhead</i> + N x 48 bytes	<i>Overhead</i> + N x 783 bytes (STS-N)
<b>Overhead para 256 octetos TDM (UDP/IP s/ RTP)</b>	11,1%	12,5%	11,9% a 13,8%	15,6%	Dependente do Serviço
<b>Compatibilidade ATM</b>	Complexa	Complexa	Simplex	Simplex	Complexa

## 2.2 - SERVIÇOS TDM SOBRE IP

### 2.2.1 - Diferenças essenciais entre as redes TDM e redes IP

As redes TDM convencionais baseadas em comutação de circuitos, também conhecidas como CSN (*Circuit-Switching Networks*), como PDH e SONET/SDH, são fortemente determinísticas, transmitindo um ou mais octetos desde o equipamento de origem até o equipamento de destino através de um canal dedicado e com largura de banda definida, em intervalos de 125us.

O atraso sofrido pelos dados que trafegam dentro de uma rede TDM é previsível, baixo e constante ao longo de todo o período da conexão, sendo a sincronização entre origem e destino provida através dos dados transmitidos e as variações admitidas no relógio (*clock*) TDM são estritamente definidas nas especificações normativas. Além disso, a infraestrutura das redes TDM suporta um rico conjunto de facilidades para os usuários através de diversos protocolos de sinalização, com grande maturidade e elevada confiabilidade.

Por sua vez, as redes de transporte em modo pacotes, conhecidas como PSN (*Packet-Switching Networks*), como as redes IP e os sistemas MPLS (*Multi-Protocol Label Switching*), apresentam-se mais eficientes que as redes TDM devido ao compartilhamento inerente da largura de banda disponível. Entretanto, essa capacidade de compartilhamento leva as redes de pacotes a tornarem-se inerentemente não-determinísticas.

Os dados – ou pacotes – enviados através dessas redes precisam competir pelos recursos oferecidos, como largura de banda, filas e interfaces dos roteadores, o que causa variação no atraso sofrido por cada um dos pacotes transmitidos e também a sua eventual perda devido a descarte dentro da rede, em caso de congestionamento ou esgotamento de recursos. Um equipamento de origem pode introduzir pacotes na rede a intervalos regulares, à semelhança de uma rede TDM, mas a PSN não oferece garantia de que esses pacotes chegarão ao equipamento de destino nesses mesmos intervalos, na mesma ordem, ou mesmo que todos os pacotes introduzidos efetivamente chegarão ao destino. Além disso, as redes IP e PSN em geral foram projetadas para o transporte de dados arbitrários, de forma que qualquer sinalização relacionada às redes TDM não é suportada.

### **2.2.2 - Transporte de serviços TDM através de redes IP**

Existem duas maneiras fundamentais pelas quais os serviços tradicionais TDM podem ser integrados às redes PSN baseadas em IP: A substituição completa da rede TDM e dos equipamentos dos usuários por uma nova infra-estrutura com capacidade para prover mecanismos inovadores para o transporte e sinalização de dados de voz em tempo real; e uma segunda alternativa que mantém intactos os equipamentos dos usuários, protocolos e aplicações, realizando o tunelamento dos dados TDM através da rede de transporte comutada em modo pacotes.

Dessa forma, a segunda alternativa representaria, em princípio, um caminho de migração mais simples e eficiente, sob o ponto de vista dos investimentos necessários para operadoras e fabricantes de equipamentos, sendo a essência da proposta historicamente conhecida como TDMoIP<sup>®</sup> e suas evoluções. [STEIN 2003]

As tecnologias SAToP e TDMoIP<sup>®</sup> permitem a emulação de circuitos tradicionais TDM (T1, E1, T3, E3 e Nx64Kbps) através da adaptação e encapsulamento, pelo nó de origem, do tráfego TDM em pacotes a serem transportados através de redes IP, MPLS ou *Ethernet*, conforme especificado no Apêndice C.

Por adaptação entende-se o mecanismo que modifica o fluxo de dados TDM original (*payload*) de forma a permitir a sua regeneração, pelo nó de destino, no momento da retirada dos pacotes da rede IP. Utilizando um processo adequado de adaptação, a sinalização e a sincronização do fluxo TDM original podem ser recuperadas, e mesmo uma certa quantidade de pacotes porventura perdidos pode ser acomodada, sem perda significativa de qualidade para a comunicação estabelecida.

Por encapsulamento entende-se a acomodação do fluxo de dados adaptado em pacotes dentro do formato e das características impostas pela tecnologia PSN utilizada. Formas de encapsulamento TDMoIP<sup>®</sup> estão hoje definidas para redes IP com transporte UDP (User Datagram Protocol) - UDP/IP, redes MPLS, redes IP com L2TP (Layer 2 Tunneling Protocol) - L2TP/IP e redes *Ethernet*, conforme estabelecido no *Internet-Draft* proposto pelo Grupo PWE3 da IETF para padronização da tecnologia TDMoIP<sup>®</sup>. [IETF2006], cuja publicação como RFC já foi requisitada.

O equipamento de interoperabilidade que conecta o circuito TDM tradicional à rede PSN é chamado *gateway* TDMoIP<sup>®</sup> ou IP-Mux, e pode estar localizado tanto no Equipamento Provedor (PE) como no Equipamento Cliente (CE). O *gateway* que encapsula os dados TDM e insere pacotes na rede PSN é chamado de “extremidade PSN” (*PSN-bound*), enquanto o *gateway* que extrai os dados TDM dos pacotes e gera tráfego para a rede TDM é chamado de “extremidade TDM” (*TDM-bound*). Os circuitos TDM emulados através dessa tecnologia são sempre ponto-a-ponto, bidirecionais, e transportam a mesma taxa de bits em ambas as direções.

### 2.2.3 - Implementação de pseudo-circuitos TDM

O aspecto crítico na implementação de pseudo-circuitos TDM é a recuperação do relógio, pois nas redes TDM tradicionais a camada física transporta informações de temporização através do fluxo de dados, síncrono ou quase-síncrono, existindo em algum ponto da rede um relógio primário com precisão da ordem de  $10^{-11}$ ; enquanto na grande maioria das redes de pacotes essa sincronização entre os nós não existe. Dessa forma, o relógio deve ser extraído a partir dos pacotes recebidos, que sofrem diferentes atrasos em função do caminho percorrido e do tráfego existente na PSN, resultando numa componente aleatória conhecida como PDV (*Packet Delay Variation*). Isso implica no desenvolvimento de mecanismos eficientes para reproduzir o relógio e a taxa constante do fluxo TDM.

O mecanismo mais utilizado, descrito em [CISCO2004], é a utilização de um *buffer* para armazenamento temporário dos dados recebidos em cada pacote, conhecido como *jitter buffer*, onde os dados recebidos da PSN são escritos à taxa variável com que são recebidos, mas lidos e enviados para o circuito TDM de destino a uma taxa constante, determinada pela velocidade de extração dos dados do *buffer*, que é controlada pelo relógio do PE de destino, conforme apresentado na Figura 2.7.

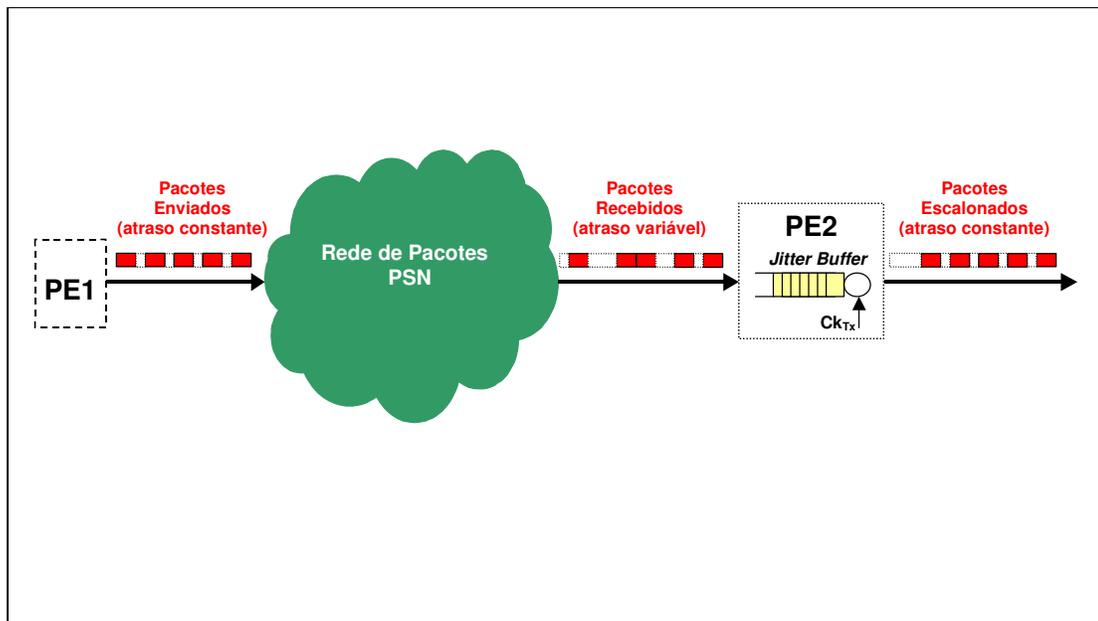


Figura 2.7 - Acomodação da variação no atraso dos pacotes pelo *jitter buffer*.

Dessa forma, a componente aleatória de alta frequência da variação no atraso sofrido pelos pacotes (PDV) é absorvida, acomodando essas variações e tornando o atraso constante do ponto de vista da rede TDM de destino, em troca da latência adicional introduzida pelo processo de armazenamento e espera. Nas condições ideais, o *jitter buffer* deveria operar com metade de sua capacidade nominal preenchida, minimizando de forma idêntica os riscos de ser totalmente preenchido, obrigando ao descarte de pacotes (*overflow*) ou não possuir pacotes para encaminhamento ao circuito TDM de destino num dado instante (*underflow*). Nessa premissa, a latência adicional introduzida no pseudo-circuito para acomodação das variações no atraso sofrido pelos pacotes é equivalente à metade do tamanho desse *buffer*. Nas implementações, o tamanho do *jitter buffer* deve ser configurável, ou ainda dinâmico, aumentando ou diminuindo de acordo o atraso fim a fim observado na PSN.

Além disso, o relógio utilizado para a regeneração do fluxo TDM pelo PE de destino precisa estar alinhado com o relógio utilizado pelo PE de origem, correspondente à taxa do fluxo TDM original, a fim de que a sincronização seja mantida entre os CEs. Como não existe, na grande maioria das PSN, um relógio comum aos PEs de origem e destino, a frequência de extração dos dados do *buffer* costuma ser apenas nominalmente igual à taxa do fluxo TDM na origem, gerando flutuações de baixa frequência no circuito TDM de destino, conhecidas como *wander*, que prejudicam a qualidade da conversação bidirecional, podendo torná-la completamente inviável.

Como discutido anteriormente no Modelo de Referência PWE3 para Sincronização da Rede, apresentado na seção 2.1.2, existem três alternativas para resolver o problema de sincronização entre as redes TDM de origem e de destino em um pseudo-circuito TDMoIP<sup>®</sup>: a utilização de redes síncronas, a sincronização relativa e a sincronização adaptativa [AWEYA2003].

No primeiro caso, correspondente aos cenários (a.1) e (a.2) do Modelo de Referência PWE3, a informação de relógio é gerada diretamente através do PE de destino (a.2), ou por enlace fechado a partir do circuito TDM de destino (a.1), os quais fazem parte ou têm outras interfaces com uma rede síncrona comum. Essa situação, representada na Figura 2.8, tem pequena abrangência, sendo limitada ao caso de expansão de uma rede TDM existente,

onde os circuitos de origem e destino já se encontram interligados por uma rede síncrona, mas são desejadas novas rotas para transporte TDM através de uma rede de pacotes que atenda ambas as extremidades; ou ao caso onde existe um relógio comum dentro da PSN, distribuído a todos os PE.

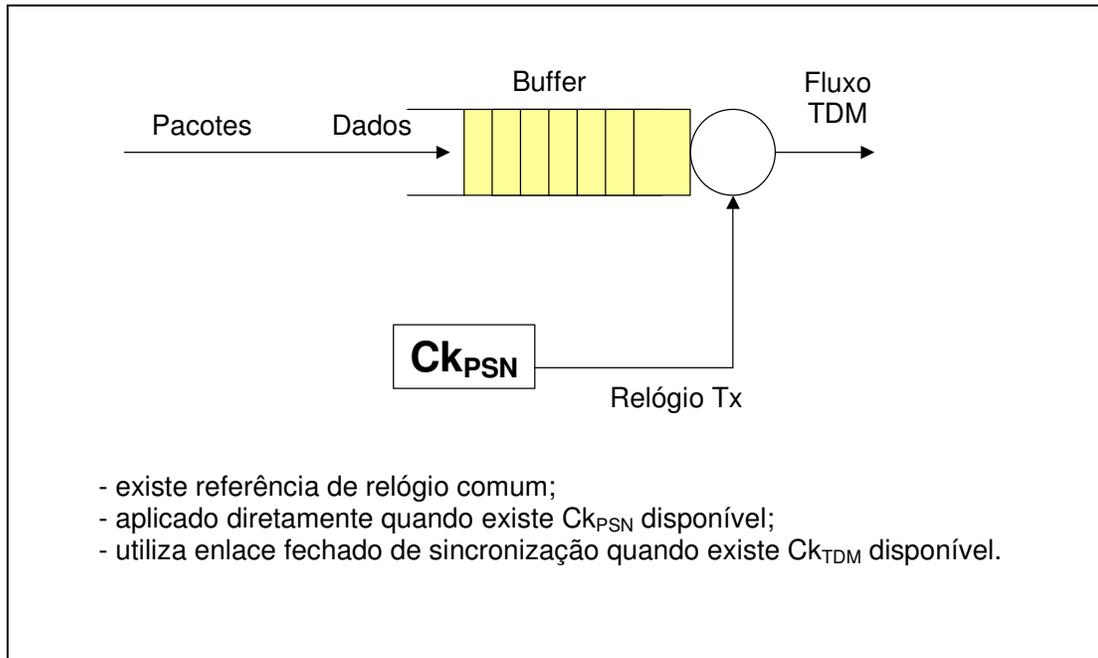


Figura 2.8 - Recuperação do relógio no destino pelo método síncrono.

No segundo caso, correspondente ao cenário (b) do Modelo de Referência PWE3, um relógio comum também está disponível para ambos os *gateways* TDMoIP<sup>®</sup>, através da PSN, mas como o relógio do circuito TDM de origem é completamente independente, a relação entre o relógio de origem e esse relógio comum é inserida no pacote de dados, utilizando o protocolo RTP (*timestamp*) ou os bits SRTS (*Synchronous Residual TimeStamp*) no *overhead* AAL1. No PE de destino, essa relação inserida no pacote é somada ao valor do relógio comum, re-estabelecendo o relógio do circuito TDM de origem, conforme representado na Figura 2.9. Essa situação, típica no caso da emulação de circuitos em redes ATM, também apresenta pequena abrangência no caso das redes de pacotes utilizadas em TDMoIP<sup>®</sup>, como IP e *Ethernet*, uma vez que as mesmas não apresentam, como característica geral, um relógio comum a todos os seus nós.

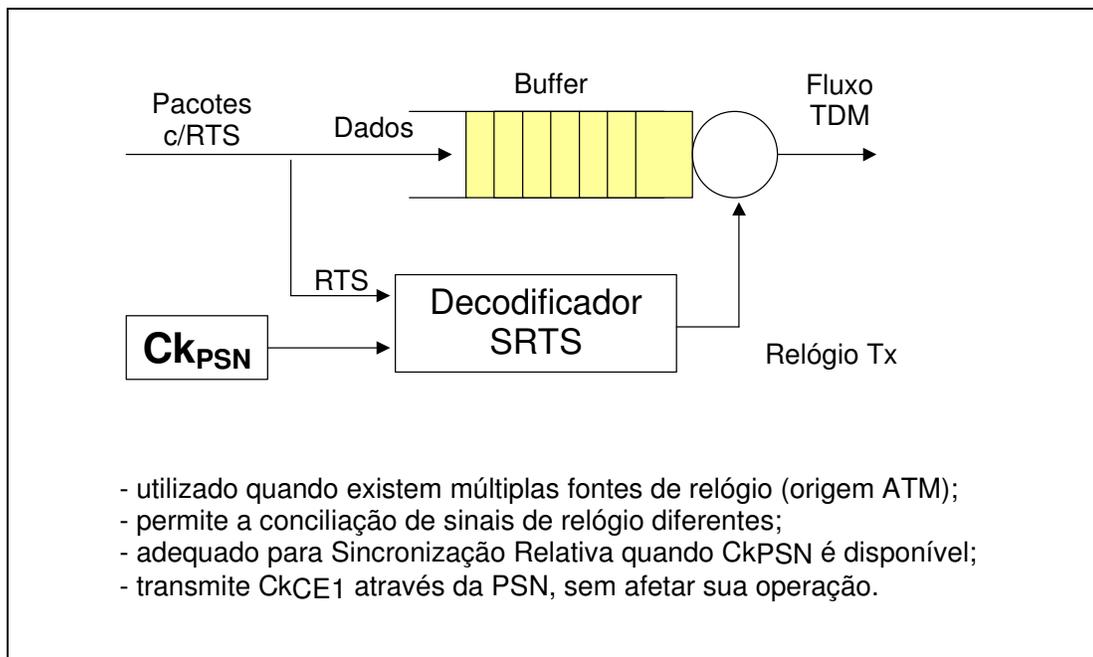


Figura 2.9 - Recuperação do relógio no destino pelo método relativo SRTS

No último caso, correspondente ao cenário (c) do Modelo de Referência PWE3, não existe um relógio comum, devendo o relógio TDM de origem ser regenerado exclusivamente com base em características observáveis nos pacotes recebidos da PSN, como por exemplo o instante exato de chegada ao PE de destino, associado ao instante de geração do pacote pelo PE de origem, que é transmitido pela rede através do protocolo RTP (*timestamp*); ou o nível de preenchimento do *jitter buffer* em função do tempo.

Em função da variação no atraso dos pacotes, processos de filtragem que eliminem a natureza estatística dessas características devem ser empregados, tais como FLLs (*Frequency Locked Loops*) e PLLs (*Phase Locked Loops*). Essa situação apresenta abrangência mais geral, sendo aplicável a todas as redes de pacotes utilizadas em TDMoIP<sup>®</sup>, incluindo aquelas baseadas em datagramas IP e quadros *Ethernet*.

Qualquer que seja o mecanismo empregado, o fluxo TDM a partir do PE de destino deve apresentar conformidade com as restrições para *jitter* e *wander* estabelecidas nas Recomendações ITU-T G.823 e G.824 para os circuitos TDM tradicionais. Na Figura 2.10 é representada a recuperação do relógio utilizando a *timestamp* transmitida através do protocolo RTP.

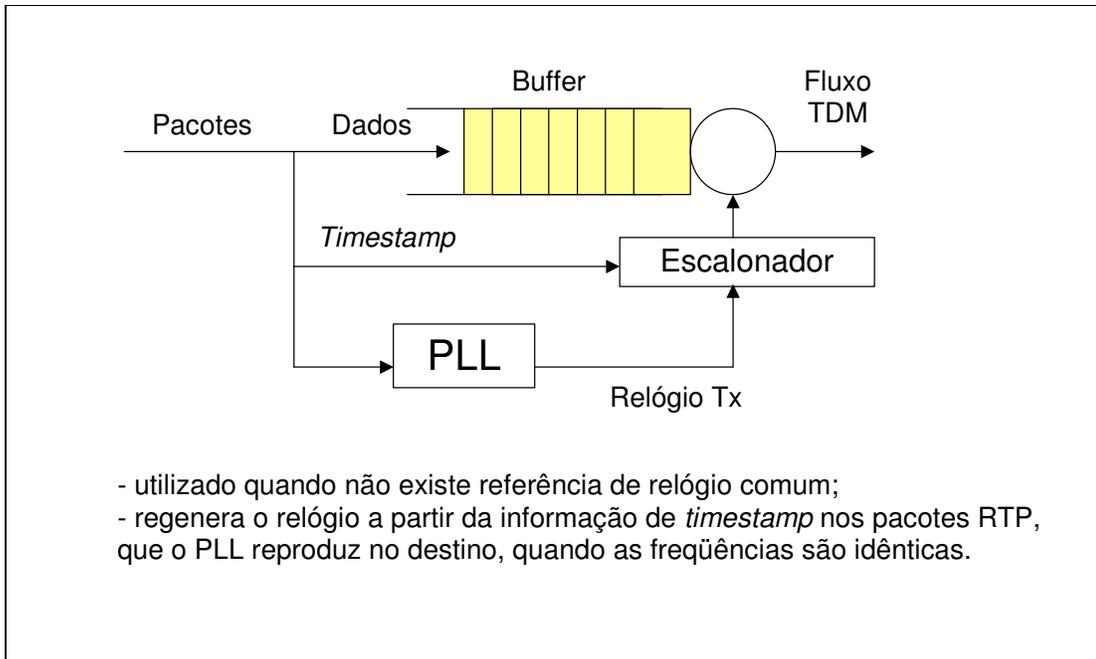


Figura 2.10 - Recuperação do relógio no destino pelo método adaptativo utilizando *timestamp* através do protocolo RTP.

Enquanto na Figura 2.11 é representada a recuperação do relógio utilizando a ocupação, no tempo, do *jitter buffer*.

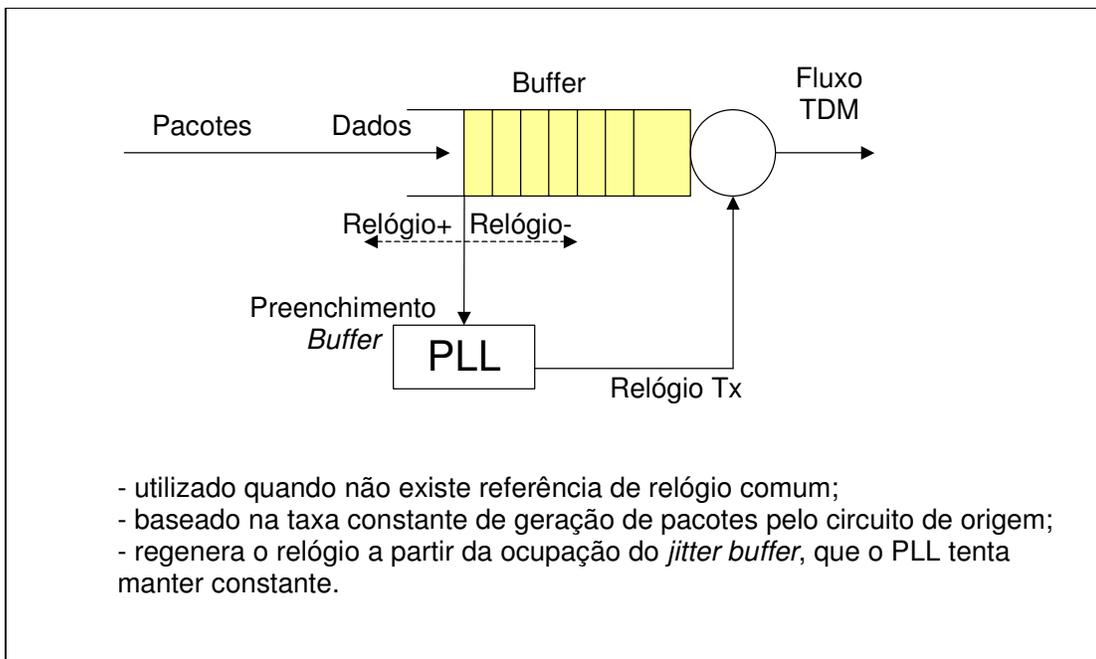


Figura 2.11 - Recuperação do relógio no destino pelo método adaptativo utilizando a ocupação, no tempo, do *jitter buffer*.

Contudo, os métodos baseados na ocupação do *jitter buffer* ao longo do tempo são sensíveis às perdas de pacotes na PSN, pois a frequência de saída é fortemente dependente do fluxo de chegada de pacotes.

Dessa forma, a sincronização adaptativa utilizando *timestamp* representa uma forma mais eficiente de regeneração do relógio utilizado pelo circuito TDM de origem, lembrando ainda que o protocolo RTP pode ser dispensado caso seja garantido que o tamanho dos pacotes e sua taxa de geração sejam constantes, conforme já discutido. Nesse caso, como o número de seqüência da Palavra de Controle TDMoIP<sup>®</sup> é gerado de forma monotônica e cada pacote contém sempre o mesmo número de amostras, a *timestamp* seria simplesmente uma função linear do número de seqüência (*NS*), calculada através da Expressão (2.1), considerando o número de seqüência inicial igual a zero, onde *TamanhoPacote* deve ser fornecido em quadros TDM:

$$\text{Timestamp} = NS \cdot (\text{TamanhoPacote}) \cdot (125 \mu s) \quad (2.1)$$

Outro aspecto fundamental é a mitigação das perdas de pacotes inerentes às PSN, uma vez que o fluxo de dados TDM pressupõe a entrega de bits a uma taxa constante, tipicamente sobre um canal dedicado, sendo tolerados alguns erros de bit, mas jamais a perda de dados em trânsito. Qualquer rede de pacotes apresenta algum grau de desordenamento e perda de pacotes, decorrente da possibilidade de caminhos múltiplos, enfileiramento e descarte em seus roteadores, sobretudo em situações de carga excessiva ou congestionamento de enlaces críticos, e essas perdas e/ou recebimento fora de ordem precisam ser adequadamente compensados para a continuidade do serviço TDM, sobretudo em aplicações como voz interativa e transmissão de vídeo.

O mecanismo para mitigação da perda e recebimento de pacotes fora de ordem é o rastreamento dos números de seqüência recebidos em cada pacote, armazenados dentro do *jitter buffer*, identificando anomalias e realizando ações para cada caso. Quando a perda de um ou mais pacotes é detectada, esse mecanismo deve gerar pacotes de preenchimento a fim de manter a sincronização TDM, enquanto pacotes com números de seqüência incorretos ou com erros de cabeçalho devem ser descartados e substituídos, assegurando que os padrões necessários à regeneração do fluxo TDM no destino, como o FAS, sejam

sempre encaminhados. Em caso de recebimento fora de ordem, os pacotes devem ser reordenados em tempo hábil para que sejam encaminhados ao circuito TDM de destino no instante correto dentro do fluxo original.

Enquanto a simples inserção de pacotes arbitrários de preenchimento pode ser suficiente para manutenção da sincronização no circuito TDM de destino, para o tráfego de telefonia isso pode causar problemas como voz entrecortada, ruído desconfortável ou mesmo uma conversação ininteligível, enquanto para o tráfego de vídeo isso pode causar distorções visíveis nas imagens recebidas, com queda de qualidade. Dessa forma, quando a probabilidade de perda for significativa, devem ser implementados mecanismos mais apurados para a mitigação das perdas de pacotes, como a escolha adequada dos valores a serem substituídos em caso de perda, com o objetivo de minimizar esses erros de percepção.

Algumas possibilidades são a repetição dos dados contidos no pacote anteriormente recebido e a geração, quando existem recursos computacionais disponíveis, de uma estimativa dos dados perdidos com base na interpolação entre o pacote anterior ao perdido e o próximo pacote do fluxo, uma vez que as perdas são usualmente detectadas pelo recebimento bem sucedido do pacote seguinte ao faltante.

Evidentemente, mecanismos de Qualidade de Serviço (QoS) também podem e devem ser empregados sempre que possível no estabelecimento dos pseudo-circuitos, a fim de assegurar que a PSN ofereça o melhor desempenho possível para o fluxo TDMoIP<sup>®</sup> transportado, reduzindo as perdas de pacotes e melhorando a qualidade dos mesmos:

- Serviços de priorização na Camada 2 podem ser solicitados utilizando o campo de prioridade VLAN, desde que todos os comutadores (*switches*) envolvidos estejam aptos a oferecê-la;
- A priorização em redes MPLS deve ser realizada utilizando os bits EXP, sendo sugerido um LSP específico e adequado às taxas envolvidas no fluxo TDM;
- A prioridade na camada IP deve ser controlada utilizando o campo ToS, desde que os roteadores envolvidos estejam aptos a oferecê-la;

- Em caso de utilização de mecanismos *DiffServ*, a classe EF (*Expedited forwarding*) deve ser utilizada, a fim de oferecer baixa latência e a menor variação possível no atraso dos pacotes (*jitter*);
- Em caso de utilização de mecanismos IntServ, o serviço GS (Guaranteed Service), com reserva de largura de banda adequada às taxas envolvidas no fluxo TDM deve ser utilizado, e o atraso introduzido pela PSN avaliado de forma a assegurar compatibilidade com os requisitos de latência para o pseudo-circuito TDM.

### **2.3 - PROPOSTA DE EVOLUÇÃO DA IMPLEMENTAÇÃO TDMoIP®**

Em função das fragilidades ainda encontradas nas implementações correntes da tecnologia TDMoIP®, esse trabalho propõe novos mecanismos para as Subcamadas de Seqüenciamento e Sincronização, buscando, na primeira, um tratamento mais completo e eficiente da reordenação de pacotes com base em um novo algoritmo para o gerenciamento do *jitter buffer*; e na segunda, uma abordagem menos sensível à QoS oferecida pela PSN para a recuperação adaptativa do relógio de transmissão no receptor, utilizando filtragem PLL e técnicas clássicas de controle para garantir o acompanhamento estrito, pelo oscilador local do receptor, da frequência de transmissão utilizada no CE de origem do fluxo TDM.

Assim, essa proposta busca estender a possibilidade de regeneração adequada do fluxo TDM no CE de destino, atendendo aos requisitos impostos pelos padrões das redes TDM tradicionais, até condições mais adversas do que as hoje toleradas no transporte de pseudo-circuitos TDM sobre redes IP, assegurando o transporte de voz e vídeo nessas redes de forma eficiente, confiável e completamente transparente aos equipamentos terminais.

#### **2.3.1 - A implementação TDMoIP® da RAD**

A RAD Data Communications Ltd. é a principal responsável pelo desenvolvimento da tecnologia TDMoIP®, tanto no que se refere ao processo de padronização, pois alguns de seus engenheiros são os principais contribuidores do grupo PWE3, como também no desenvolvimento de produtos comerciais, tendo colocado no mercado diversos equipamentos com a marca *TDMoIP® Driven*. Dessa forma, foram realizados contatos com

seus representantes no Brasil, no sentido de buscar maiores informações sobre seus produtos e também obter equipamentos para demonstração e análise da tecnologia utilizada nos mesmos.

Analisando a documentação a que tivemos acesso [RAD2005], foram descritas algumas características da implementação TDMoIP<sup>®</sup> da RAD, as quais foram confrontadas com a arquitetura e os requisitos estabelecidos pelo grupo PWE3 para emulação de circuitos TDM sobre PSN, descritos respectivamente, nos Apêndices A e B.

Nesse confronto, foram enfatizados os dois aspectos identificados como os mais críticos para o transporte de pseudo-circuitos TDM sobre redes de pacotes, ambos relativos à Camada de Encapsulamento: as Subcamadas de Seqüenciamento e Sincronização.

Para o seqüenciamento, a implementação RAD utiliza um *jitter buffer*, que é preenchido pelos pacotes IP recebidos e esvaziado pela regeneração do fluxo TDM em direção ao CE de destino. Essa regeneração é iniciada somente após o preenchimento de metade do *buffer*, a fim de acomodar a variação no atraso sofrida pelos pacotes na PSN dentro de um limite máximo de tolerância, que também corresponde ao atraso adicional introduzido pelo *jitter buffer* no pseudo-circuito. Assim, o tamanho do *buffer* é configurável entre 3ms e 300ms, permitindo a acomodação de um atraso total, correspondente ao provocado pela PDV da rede mais o atraso de empacotamento, entre 1,5ms e 150ms.

O atraso de empacotamento para uma dada configuração de canais, designada *bundle* na documentação RAD, é dado pela expressão (2.2).

$$AtrasoEmpacotamento = \frac{47.N.(125\mu s)}{TS} \quad (2.2)$$

Onde:

$N$  é o número de octetos TDM por pacote, considerando o tamanho da célula AAL (48 octetos), como indicado na expressão (2.3). Em função da MTU típica na *Ethernet*,  $N$  pode variar entre 1 (48 octetos) e 30 (1440 octetos);

$TS$  é o número de *timeslots* válidos existentes no fluxo TDM (32 para E1 completo).

$$N = \frac{\text{Octetos TDM por pacote}}{48} \quad (2.3)$$

Evidentemente, existe um compromisso entre a largura de banda ocupada pelo pseudo-circuito, que é reduzida pela utilização de um número maior de octetos TDM por pacote, em função do menor *overhead* correspondente aos cabeçalhos; e o atraso de empacotamento, que aumenta pela necessidade de aguardar a chegada de mais quadros TDM para montagem de um pacote.

O tamanho do *jitter buffer* deve ser configurado para o pseudo-circuito de acordo com a expressão (2.4).

$$\text{TamanhoBuffer} = \text{AtrasoEmpacotamento} + PDV \quad (2.4)$$

Onde *PDV* é a máxima variação esperada para o atraso sofrido pelos pacotes, estimada ou medida sobre a rede PSN, utilizando metade do valor RTT (*Round-Trip Time*) médio, por exemplo.

As funções de reordenação da subcamada de seqüenciamento dependem do formato escolhido para o fluxo de dados TDM, designadas V1 (*Version 1*) e V2 (*Version 2*) na documentação:

No caso da Versão 1, onde é utilizada AAL1 para adaptação do fluxo TDM, a reordenação de pacotes recebidos fora de ordem de transmissão é suportada apenas para valores ímpares de N, ou seja, 1x48, 3x48, ..., 29x48 octetos TDM por pacote, permitindo a recuperação da ordenação original para até sete pacotes, em função do limite que pode ser endereçado pelo ponteiro AAL1, de 3 bits.

No caso da Versão 2, onde é utilizada AAL2 para adaptação do fluxo TDM, a reordenação de pacotes recebidos fora de ordem de transmissão é suportada para qualquer valor de N, permitindo a recuperação da ordenação original para até 64 pacotes.

Contudo, esses valores representam os limites máximos para a reordenação dos pacotes, sendo os valores reais aplicáveis a um determinado pseudo-circuito dependentes do tamanho do *jitter buffer* e determinados pela expressão (2.5), permanecendo limitados às quantidades máximas 7 (V1) e 64 (V2) mesmo que o resultado da expressão seja superior.

$$PacotesReordenaveis = \frac{[(TamanhoBuffer(ms) - 1)] \cdot (TS \cdot 8)}{47 \cdot N} \quad (2.5)$$

Ainda com relação ao seqüenciamento, em caso de recebimento de um pacote duplicado, a implementação RAD considera apenas aquele recebido por último, e as perdas de pacote são detectadas através dos erros de seqüência observados nos pacotes armazenados no *jitter buffer*.

Assim, pode ser deduzido que existe um processo específico de gerenciamento do *jitter buffer*, que implementa a reordenação, quando possível, substitui os pacotes duplicados e monitora/trata a ocorrência de erros. Tal característica é reforçada pela descrição dos erros de seqüência, *underflow* e *overflow* na documentação, que pressupõe esse gerenciamento explícito:

- **Erros de Seqüência:** o *gateway* de destino verifica os números de seqüência através do mecanismo de recepção e espera que cada um dos novos pacotes recebidos esteja na seqüência, em relação ao anterior (pacote 5 recebido após o 4 e assim sucessivamente). Quando, por alguma razão, isso não acontece, significa que houve um problema com a integridade do fluxo de pacotes, e portanto com o transporte de voz/vídeo/dados, que é indicado através do incremento do contador *Sequence Errors*.
- **Erros de Underflow:** após a inicialização do pseudo-circuito, o *jitter buffer* armazena pacotes até que esteja preenchido com metade da sua capacidade máxima, correspondente ao espaço ocupado pelo equivalente em pacotes do tamanho configurado, em milissegundos. Somente após esse ponto o *gateway* começa a gerar o fluxo TDM em direção ao CE de destino. Os pacotes armazenados asseguram que o equipamento TDM será alimentado com um fluxo contínuo de dados mesmo se os pacotes subseqüentes sofrerem atrasos variáveis na PSN. Obviamente, se esses pacotes sofrem atrasos demasiado longos, o *buffer* será gradualmente esvaziado até que não existam mais pacotes disponíveis para continuidade do fluxo TDM, situação que

provoca o incremento do contador *Jitter Buffer Underflow* e indica um problema de integridade fim a fim no transporte de voz/vídeo/dados.

- **Erros de *Overflow*:** em regime permanente, o *jitter buffer* está preenchido até a metade, o que significa que existe espaço para armazenar uma quantidade equivalente de novos pacotes recebidos. O *overflow* é o fenômeno oposto ao *underflow*, isto é, quando uma grande rajada de pacotes é recebida pelo *gateway*, em número superior à sua capacidade de armazenamento, o buffer é totalmente preenchido. Neste caso, um número desconhecido de pacotes excedentes é descartado e é gerado um *underflow* forçado, em virtude do esvaziamento do *jitter buffer* para que o processo seja reiniciado desde o princípio. Um *overflow* provoca o incremento do contador *Jitter Buffer Overflow* e sempre resulta num *underflow* imediato, forçado pelo *gateway*, com os mesmos efeitos já descritos acima.

Para sincronização do fluxo TDM no PE de destino, a implementação RAD não utiliza o protocolo RTP, por isso permite a utilização de apenas três cenários previstos no Modelo de Referência para Sincronização de Rede descrito na seção 2.1.2, da seguinte forma:

- Para o cenário (a.1), ou seja, sincronização através dos PEs, o relógio comum disponível na PSN é aplicado às entradas EXT CLK dos *gateways*, que é configurado para utilização de relógio externo, sendo os equipamentos TDM nas extremidades do pseudo-circuito configurados para enlace fechado (*loopback*) de sincronização, como apresentado na Figura 2.12 (a.1).
- Para o cenário (a.2), ou seja, sincronização através dos CEs, os *gateways* são simplesmente configurados para enlace fechado (*loopback*) de sincronização, sendo o relógio comum garantido pelos equipamentos TDM nas extremidades do pseudo-circuito, como apresentado na Figura 2.12 (a.2).
- Para o cenário (c), ou seja, sincronização adaptativa, um dos *gateways* é configurado para enlace fechado (*loopback*) de sincronização, obtendo o relógio a partir do equipamento TDM a ele conectado, enquanto o *gateway* na outra extremidade do pseudo-circuito é configurado para sincronização adaptativa, como apresentado na Figura 2.12 (c).

O cenário (b), ou seja, sincronização relativa, não é implementado.

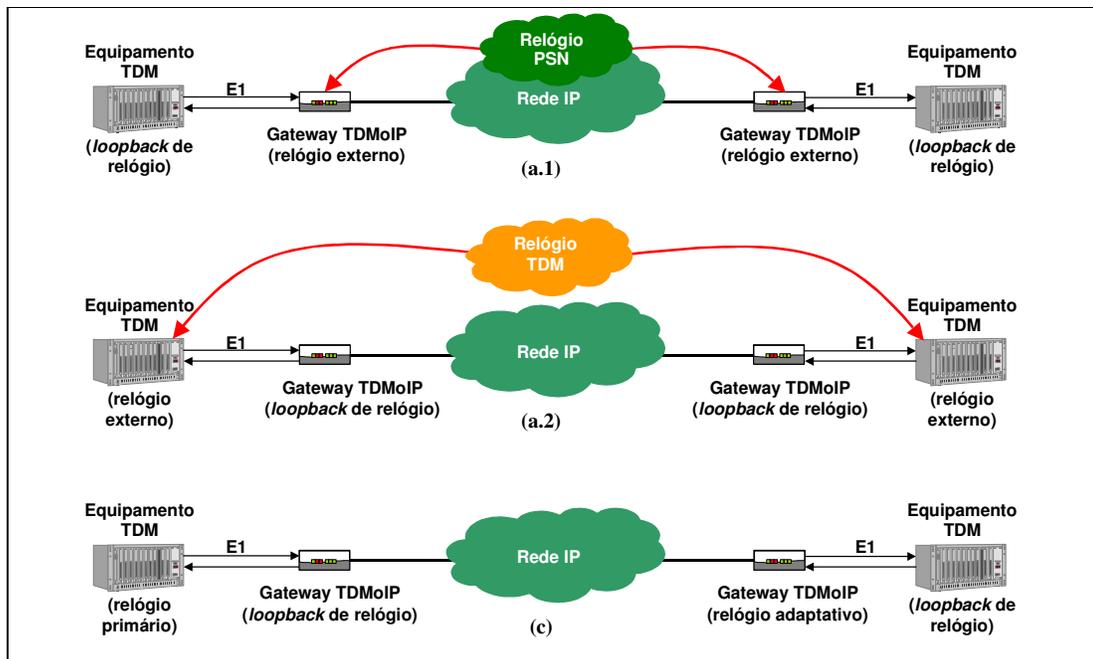


Figura 2.12 - Esquemas de sincronização para gateways TDMoIP<sup>®</sup> RAD.

O algoritmo utilizado na sincronização adaptativa é a regeneração do relógio de transmissão utilizado pelo CE de origem, através da monitoração do estado de preenchimento do *jitter buffer* ao longo do tempo, como mostrado na Figura 2.11 acima. Quando a ocupação do *buffer* cresce, a frequência do relógio de transmissão gerado pelo oscilador interno do *gateway* de destino é incrementada, a fim de evitar um *overflow* do mesmo; da mesma forma, quando o *buffer* começa a esvaziar, a frequência do relógio de transmissão é reduzida, evitando o seu *underflow*.

Esse algoritmo, como já discutido, é bastante sensível a perdas e/ou variação no atraso sofrido pelos pacotes que atravessam a PSN, de forma que a qualidade do relógio regenerado está intimamente ligada à qualidade oferecida pela rede ao tráfego que transporta o fluxo TDM, o que implica em necessidade de utilização de ferramentas de QoS, nem sempre disponíveis, para sua operação satisfatória, comprometendo a robustez da implementação RAD para o transporte de voz interativa e transmissão de vídeo nas redes onde não é possível uma garantia de QoS fim a fim.

Um último aspecto a destacar na implementação RAD é a chamada Conectividade de Operação e Manutenção (OAM *Connectivity*), que não é definida pelo *Internet-Draft* do

padrão TDMoIP<sup>®</sup> e, portanto, pode ser configurada ou não nos *gateways*. Essa funcionalidade implementa, através de um número de *bundle* exclusivo, um canal de controle entre os equipamentos de origem e destino, na forma do Canal Tipo 1 descrito pela Arquitetura PWE3, permitindo que o *gateway* de destino seja detectado e uma conexão válida reconhecida antes de iniciar o fluxo de dados TDM, evitando sobrecarga na PSN pelo envio descontrolado de pacotes, em taxas elevadas, sem que o pseudo-circuito esteja estabelecido e operacional. Através desse mecanismo, um *gateway* TDMoIP<sup>®</sup> RAD somente começa a transmitir os pacotes contendo dados TDM à taxa plena quando detecta um *gateway* remoto em condições para recebimento do fluxo de dados, evitando o desperdício de largura de banda na rede de pacotes.

### **2.3.2 – O novo mecanismo para seqüenciamento de pacotes proposto**

Na implementação RAD, são observadas duas limitações na subcamada de seqüenciamento: a existência de fortes restrições para o processo de reordenação dos pacotes recebidos no destino e a necessidade de gerenciamento do *jitter buffer*, levando à necessidade de reiniciar o pseudo-circuito, com o esvaziamento desse *buffer*, sempre que sua capacidade é excedida (*overflow*) durante a emulação de um pseudo-circuito.

Além disso, o tamanho do *jitter buffer* é fixado pelo operador no momento de estabelecimento do pseudo-circuito, com base em estimativa ou medição do atraso provocado pela PSN, não permitindo reconfiguração enquanto o mesmo estiver ativo.

Assim, como forma de melhorar a robustez da implementação, tornando-a menos sensível à QoS oferecida pela PSN ao fluxo de pacotes contendo dados TDM, é proposto um novo mecanismo integrado de armazenamento, reordenação e mitigação de perdas de pacote, implementado através de um *buffer* circular auto-gerenciável, que será detalhado a seguir.

Nessa proposta, ao invés da abordagem tradicional de implementação do *jitter buffer*, onde os pacotes recebidos são inicialmente armazenados em sua ordem de chegada, cabendo a um segundo mecanismo a sua reordenação, quando possível, é substituída por um mecanismo integrado onde o pacote recebido é sempre armazenado diretamente em sua posição correta dentro do *buffer*, de característica circular, com base, exclusivamente, em

seu número de seqüência; eliminando a necessidade de qualquer processo adicional de reordenação ou gerenciamento desse *buffer*. Dessa forma, o processo de leitura dos dados para regeneração do fluxo TDM realiza simplesmente uma varredura seqüencial e contínua do *buffer* circular, atrasada no tempo em relação ao processo de escrita, uma vez que a ordem dos dados já foi assegurada pelo processo de escrita, tornando a operação no receptor muito mais eficiente, como ilustrado na Figura 2.13, assumindo pacotes de 1ms e um *jitter buffer* de 16 posições (16ms).

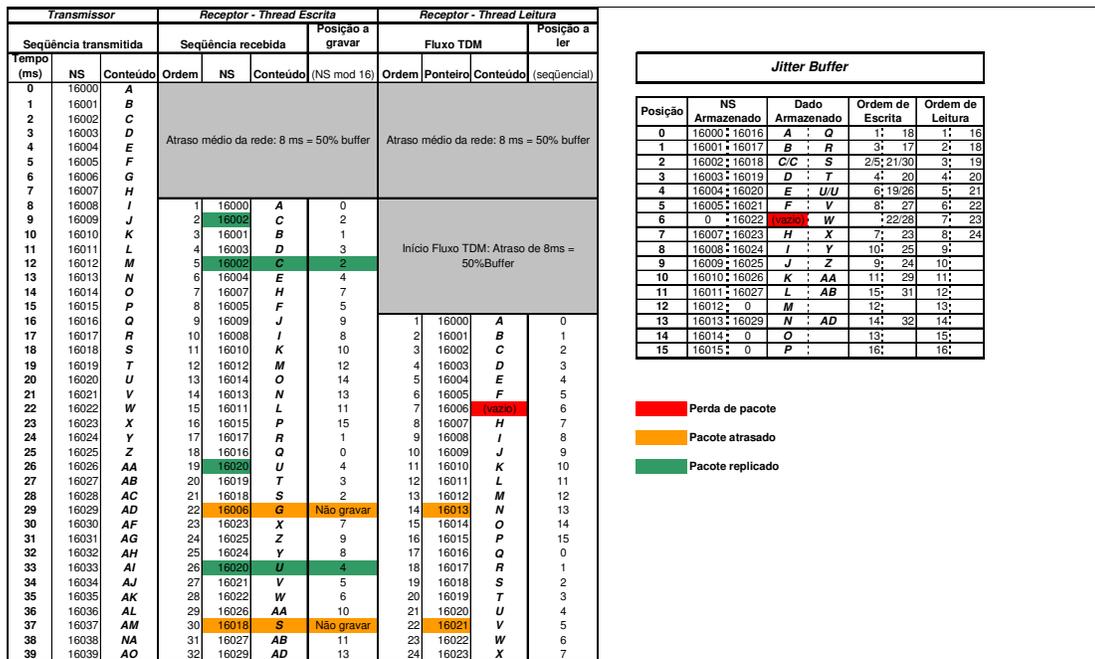


Figura 2.13 - Ilustração do mecanismo de seqüenciamento proposto.

As primeiras três colunas representam o comportamento do transmissor, com pacotes sendo transmitidos na seqüência a cada milissegundo, contendo os dados representados pela seqüência alfabética A, B, C, ..., AA, AB, AC, ... As quatro colunas seguintes, com linhas deslocadas 8 ms do início da transmissão, correspondente ao atraso provocado pela rede, representam os pacotes recebidos no destino e tratados pela *thread* de escrita, mostrando a ordem de chegada na primeira, o NS recebido na segunda e o dado do pacote na terceira, indicando na quarta coluna a posição em que os mesmos são escritos no *buffer* de acordo com o algoritmo proposto. Nas outras quatro colunas seguintes, é representado o processo de reprodução dos pacotes TDM executado pela *thread* de leitura, novamente deslocado de metade do tamanho do *buffer* utilizado (8 ms) e mostrando na primeira coluna

a ordem de reprodução, na segunda o NS do pacote reproduzido e na terceira o respectivo dado, indicando na quarta coluna a posição onde este foi lido através do processo de varredura seqüencial. Um diagrama de ocupação de cada uma das 16 posições do *buffer* é também reproduzido na Figura 2.13, à direita, mostrando as respectivas ordens de escrita e leitura dos NS e dados dos pacotes em cada posição.

O primeiro pacote recebido é o 16000, que pelo algoritmo é gravado na posição “0”, enquanto o segundo é o 16002, que é gravado na posição “2”; o terceiro pacote a chegar é o 16001, que é corretamente colocado na posição “1” de acordo com o algoritmo. O quinto pacote recebido, no entanto, é novamente o 16002 (indicado em verde), que sobrepõe o dado armazenado anteriormente de forma transparente, conforme definido pelo novo algoritmo proposto. Esse processo continua indefinidamente, mas quando 50% do *buffer* é ocupado, é também iniciado o processo de reprodução dos dados armazenados, varrendo de forma seqüencial e cíclica as 16 posições. Pode ser ainda observado na Figura 2.13 que quando chega o instante de reprodução do pacote 16006 (indicado em vermelho), este ainda não foi recebido e o *buffer* está vazio, obrigando o receptor a executar um processo de mitigação da perda. Quando esse pacote finalmente chega, como o vigésimo segundo recebido, ele não é mais válido pois está demasiado atrasado e portanto não é sequer gravado no *buffer* (indicado em laranja).

Em caso de perda de pacotes na PSN, existe o risco de que os dados contidos na posição corrente de leitura não sejam válidos, mas referentes a um instante de tempo anterior, de forma que sua validade precisa ser verificada pela continuidade do número de seqüência armazenado e, em caso de inconsistência, deve ser executado o processo de mitigação previamente definido, como a transmissão de um padrão constante, a reprodução dos dados do pacote anterior ou a reconstrução dos dados com base na interpolação entre as posições adjacentes do *jitter buffer*, por exemplo, garantindo assim uma taxa constante de bits na saída do receptor: o fluxo TDM regenerado em direção ao CE de destino.

Nessa abordagem, não existe limitação para a quantidade de pacotes reordenáveis, uma vez que os mesmos podem ser escritos em suas posições corretas no momento em que são recebidos, seja ele qual for, desde que ainda sejam válidos com relação ao processo de leitura e reprodução. Pacotes atrasados em relação ao processo de leitura, cuja perda já foi mitigada, são automaticamente descartados (não armazenados) e pacotes que tenham sido

replicados pela rede e cheguem novamente ao receptor vão automaticamente substituindo seus antecessores, assegurando que não exista duplicidade no fluxo TDM de saída, tudo isso de forma integrada, dentro do processo de escrita, que é extremamente simples.

Da mesma forma, não é necessário que o pseudo-circuito seja reiniciado quando ocorrerem rajadas de pacotes que excedam a capacidade do *jitter buffer (overflow)*, com seu esvaziamento e introdução de novo atraso até que seja novamente alcançada a metade de sua capacidade, como na implementação RAD, uma vez que o armazenamento de pacotes além da posição corrente de leitura é automaticamente tratado pelo processo de escrita, substituindo todos os pacotes antigos previamente armazenados pelos que acabam de ser recebidos, mais recentes. Isso não evita, evidentemente, os erros introduzidos nos dados pela perda dos pacotes sobrepostos, que acabam tratados pelo processo de mitigação definido, mas assegura a continuidade do pseudo-circuito após essas rajadas, uma vez que não existe condição de *overflow*, minimizando o impacto desses erros na disponibilidade do circuito TDM, já que não são introduzidos atrasos adicionais nem existe descontinuidade no fluxo de bits.

Essa característica permite ainda que o tamanho do *buffer* circular possa ser alterado enquanto o pseudo-circuito permanece ativo, acomodando automaticamente as perdas de pacote ou de seqüência que possam ser introduzidas durante a reconfiguração, de forma a que não haja perda de continuidade no fluxo de bits. Dessa forma, é possível introduzir técnicas de ajuste adaptativo nos parâmetros de recepção do PE de destino, o que confere maior robustez ao pseudo-circuito TDM em relação a grandes variações de desempenho da PSN, característica extremamente desejável para a grande variabilidade de condições esperada numa aplicação de *broadcasting* IPTV. Uma vez que um dado receptor detecte crescimento apreciável no atraso sofrido pelos pacotes recebidos, o tamanho do *jitter buffer* pode ser automaticamente incrementado, assegurando melhor acomodação do novo patamar de atraso; da mesma forma, quando detectada a melhoria de desempenho da rede em relação ao atraso, o tamanho do *jitter buffer* pode ser imediatamente reduzido por esse receptor, preservando seus recursos, sem prejuízo da continuidade do fluxo TDM. A Figura 2.14 ilustra o comportamento transparente do mecanismo de seqüenciamento durante uma reconfiguração, supondo a mudança de tamanho no *jitter buffer* de 16 para 24 posições.

Transmissor			Receptor - Thread Escrita			Receptor - Thread Leitura			
Seqüência transmitida			Seqüência recebida			Fluxo TDM			
Tempo (ms)	NS	Conteúdo	Ordem	NS	Conteúdo	Posição a gravar (NS mod 16)	Ordem	Ponteiro	Conteúdo (seqüencial)
0	16000	A							
1	16001	B							
2	16002	C							
3	16003	D							
4	16004	E							
5	16005	F							
6	16006	G							
7	16007	H							
8	16008	I							
9	16009	J							
10	16010	K							
11	16011	L							
12	16012	M							
13	16013	N							
14	16014	O							
15	16015	P							
16	16016	Q							
17	16017	R							
18	16018	S							
19	16019	T							
20	16020	U							
21	16021	V							
22	16022	W							
23	16023	X							
24	16024	Y							
25	16025	Z							
26	16026	AA							
27	16027	AB							
28	16028	AC							
29	16029	AD							
30	16030	AE							
31	16031	AF							
32	16032	AG							
33	16033	AH							
34	16034	AI							
35	16035	AJ							
36	16036	AK							
37	16037	AL							
38	16038	AM							
39	16039	AN							

Jitter Buffer				
Posição	NS Armazenado	Dado Armazenado	Ordem de Escrita	Ordem de Leitura
0	16000	16016	A : Q	1: 18
1	16001	16017	B : R	3: 17
2	16002	16018	C : S	2:5: 21
3	16003	16019	D : T	4: 20
4	16004	16020	E : U	6: 19
5	16005	16021	F : V	8: 6:
6	16014	(vazio)		22 7:
7	16007	16023	H : X	7: 23
8	16008	16024	I : Y	10: 9:
9	16009	16025	J : Z	9: 24 10: 18
10	16010	16026	K : AA	11: 30 11: 19
11	16011	16027	L : AB	15: 12: 20
12	16012	16028	M : AC	12: 26 13: 21
13	16013	16029	N : AD	14: 27 14: 22
14	16014	16030	O : AE	13: 28 15: 23
15	16015	16031	P : AF	16: 16: 24
16	16024	16032	Q : AG	25: 2:
17	16025	16033	R : AH	26: 3:
18	16026	16034	S : AI	27: 4:
19	16027	16035	T : AJ	28: 5:
20	16028	16036	U : AK	29: 6:
21	16029	16037	V : AL	30: 7:
22	16030	16038	W : AM	31: 8:
23	16031	16039	X : AN	32: 9:

Figura 2.14 - Exemplo de reconfiguração dinâmica do jitter buffer.

A seguir, é apresentado um pseudo-código para implementação do novo mecanismo de seqüenciamento de pacotes proposto nesse trabalho:

- Aplicação para Recepção de dados TDM sobre PSN:

// Objetos globais compartilhados entre as threads:

```

PonteiroTDM      Inteiro 16 bits ; ponteiro de leitura dos pacotes no buffer
TamanhoBuffer   Inteiro      ; tamanho, em pacotes, do buffer de recepção
TempoPacote     Inteiro      ; duração, em ms, dos dados TDM em cada pacote
Buffer:         Estrutura    ; buffer de armazenamento dos pacotes recebidos
    Buffer.NS      (0..TamanhoBuffer-1) de Inteiro 16 bits
    Buffer.Payload (0..TamanhoBuffer-1) de Pacote

```

// Inicialização do buffer de recepção:

```
Posicao      Inteiro (0)
```

```
ENQUANTO Posicao < TamanhoBuffer
```

```
    Buffer.NS(Posicao) = 0
```

```
    Incrementa(Posicao)
```

```
FIM ENQUANTO ; assegura que NS = 0 para inicialização do buffer
```

```
AguardaPacotes() ; aguarda a chegada do primeiro pacote na porta 2142 (TDMoIP®)
```

```
EscritaBuffer.inicio() ; inicia thread de escrita no buffer
```

```
Espera(TamanhoBuffer*TempoPacote*50%) ; aguarda ocupação de 50% do buffer
```

```
LeituraBuffer.inicio() ; inicia thread de leitura no buffer
```

- *Thread* de Escrita:

FUNCAO EscritaBuffer()

// Armazenamento dos pacotes no *buffer* de recepção:

```
NS          Inteiro 16 bits ; número de seqüência a escrever
Posicao     Inteiro      ; posição de escrita no buffer
Payload    Pacote      ; dados a escrever
PacoteRx:  Estrutura    ; objeto para tratamento dos pacotes recebidos
           PacoteRx.NS  Inteiro 16 bits
           PacoteRx.Payload Pacote
```

```
PacoteRx = BuscaPacote() ; obtém primeiro pacote recebido da interface de rede
NS = PacoteRx.NS         ; obtém o número de seqüência desse pacote
Payload = PacoteRx.Payload ; obtém os dados desse pacote
Posicao = NS mod TamanhoBuffer ; define posição de escrita do primeiro pacote
Buffer.grava(Posicao, NS, Payload) ; grava primeiro pacote no buffer
PonteiroTDM = PacoteRx.NS ; define posição inicial do ponteiro de leitura
ENQUANTO AplicacaoAtiva
    NS = PacoteRx.NS ; obtém o número de seqüência do pacote
    Payload = PacoteRx.Payload ; obtém os dados do pacote
    Posicao = NS mod TamanhoBuffer ; define posição de escrita
    // Grava o pacote no buffer de recepção se este não estiver atrasado demais:
    SE NS > PonteiroTDM ENTAO Buffer.grava(Posicao, NS, Payload)
    PacoteRx = BuscaPacote() ; obtém novo pacote da interface de rede
FIM ENQUANTO
```

- *Thread* de Leitura:

FUNCAO LeituraBuffer()

// Leitura dos pacotes do buffer de recepção e geração do fluxo TDM de saída:

```
NS          Inteiro 16 bits ; número de seqüência lido do buffer
Posicao     Inteiro      ; posição de leitura do buffer
Payload    Pacote      ; dados lidos do buffer
PacoteRx:  Estrutura    ; objeto para tratamento dos pacotes recebidos
           PacoteRx.NS  Inteiro 16 bits
           PacoteRx.Payload Pacote
```

```
Posicao=PonteiroTDM mod TamanhoBuffer ; obtém posição de leitura no buffer
ENQUANTO AplicacaoAtiva
    Payload = Buffer.le(Posicao, Payload) ; obtém os dados do pacote
    TransmiteTDM(Payload) ; reproduz dados para fluxo TDM
    Incrementa(PonteiroTDM) ; avança pacote a ser lido
    Posicao=PonteiroTDM mod TamanhoBuffer ; obtém posição de leitura no buffer
    NS = Buffer.le(Posicao, NS) ; obtém o número de seqüência do
    ; próximo pacote armazenado
```

```

// Mitiga perda se o pacote armazenado não for o esperado pelo ponteiro:
SE NS <> PonteiroTDM ENTAO MitigaPerda()
// Executa processo de sincronização do fluxo TDM:
Espera(TempoPacote)
FIM ENQUANTO

FUNCAO MitigaPerda()
// Exemplo de mitigação de pacotes perdidos:
Buffer.grava(Posicao, PonteiroTDM, Payload)      ; grava os dados do pacote anterior na
                                                  ; próxima posição do buffer a ser lida

```

Cabe observar que a implementação do mecanismo proposto é extremamente simples, e assegura robustez adicional para a utilização de técnicas avançadas para a mitigação das perdas de pacote, uma vez que a verificação da validade do próximo pacote é realizada imediatamente após a transmissão do último pacote válido, preparando a próxima transmissão para os dados fluxo TDM, o que assegura um intervalo equivalente à duração dos dados em cada pacote para o processamento dessa mitigação. Considerando um fluxo E1, para 256 octetos de dados por pacote, a regeneração dos dados de um pacote perdido poderia consumir até 1ms de processamento, sem afetar a continuidade desse fluxo.

### 2.3.3 – O mecanismo alternativo para sincronização TDM proposto

Na implementação RAD com sincronização adaptativa, é utilizada a monitoração do estado de ocupação do *jitter buffer* ao longo tempo como mecanismo de sincronismo para a regeneração do fluxo TDM, atuando diretamente sobre a frequência de relógio do oscilador local que controla a taxa de envio dos bits TDM à interface de saída, conforme apresentado anteriormente na Figura 2.11.

Como já evidenciado, essa característica torna a precisão do relógio de saída fortemente dependente do desempenho da PSN, uma vez que as perdas de pacote têm influência direta na ocupação do *jitter buffer*, tendendo a retardar o relógio do receptor mesmo que este esteja reproduzindo fielmente o relógio do fluxo TDM de origem. Da mesma forma, uma rajada de pacotes gerada por melhoria nas condições de enfileiramento em um dado nó da PSN, que alcance o receptor num intervalo relativamente curto, tende a acelerar o relógio do receptor de forma equivocada. Ambas as situações são indesejáveis quando da utilização da tecnologia TDMoIP<sup>®</sup> como alternativa para *broadcasting* IPTV, pois essas

alterações indevidas na taxa do fluxo de bits de saída tendem a ser agravadas pelos algoritmos de decodificação de áudio e vídeo nos receptores conectados ao CE de destino, afetando negativamente a qualidade do serviço emulado através do pseudo-circuito.

Desse forma, como a premissa da alternativa PWE3 para *broadcasting* de IPTV é estabelecer um fluxo de bits confiável, a taxa constante e transparente aos protocolos e codificadores de áudio e vídeo utilizados, é necessário utilizar uma técnica de sincronização adaptativa diferente, que assegure a estabilidade e a precisão do relógio no PE de destino mesmo em condições de QoS não assegurada pela PSN, como por exemplo aquelas baseadas em transmissão de uma marcação de tempo (*timestamp*) explícita para os pacotes, utilizando o protocolo RTP, conforme apresentado anteriormente na Figura 2.10.

Em [AWEYA2004] foi realizado um estudo dessa alternativa, incluindo o projeto de um circuito PLL para garantia da sincronização do relógio no receptor, de forma a atender aos requisitos estabelecidos para a regeneração do fluxo TDM. Esse estudo foi validado através de resultados experimentais, obtidos com a implementação do PLL na emulação de um circuito T3 não estruturado, utilizando placas wanPMC-2T3E3 da SBE Inc. e uma rede *Ethernet* com parâmetros controlados, conforme apresentado em [AWEYA2004a].

Contudo, a utilização do protocolo RTP para a transmissão explícita de *timestamps* através da PSN não é desejável para nossa proposta de *broadcasting* IPTV, pois implica em maior complexidade de implementação e num *overhead* adicional de 12 octetos para cada pacote transmitido, reduzindo a eficiência de banda do pseudo-circuito.

Para fluxos onde a taxa de bits é constante (CBR), a quantidade de octetos em cada pacote, embora configurável, é invariante para o pseudo-circuito estabelecido, representando um intervalo de tempo bem definido e fixo, como pode ser observado na Tabela 2.2.

Tabela 2.2 - Duração das amostras TDM para cada tamanho do pacote de dados.

<b>Tamanho do Pacote (Payload)</b>	<b>SAToP</b>	<b>CESoPSN</b>	<b>TDMoIP/ CBR</b>
32 octetos		0,125 ms	
47 octetos			0,184 ms
64 octetos		0,250 ms	
94 octetos			0,367 ms
96 octetos		0,375 ms	
128 octetos		0,500 ms	
141 octetos			0,551 ms
160 octetos		0,625 ms	
188 octetos			0,734 ms
192 octetos		0,750 ms	
224 octetos		0,875 ms	
235 octetos			0,918 ms
256 octetos	1,000 ms	1,000 ms	
282 octetos			1,102 ms
288 octetos		1,125 ms	
320 octetos		1,250 ms	
329 octetos			1,285 ms
352 octetos		1,375 ms	
376 octetos			1,469 ms
384 octetos		1,500 ms	
416 octetos		1,625 ms	
423 octetos			1,652 ms
448 octetos		1,750 ms	
470 octetos			1,836 ms
480 octetos		1,875 ms	
512 octetos	2,000 ms	2,000 ms	
517 octetos			2,020 ms
564 octetos			2,203 ms
611 octetos			2,387 ms
640 octetos		2,500 ms	
658 octetos			2,570 ms
705 octetos			2,754 ms
752 octetos			2,938 ms
768 octetos	3,000 ms	3,000 ms	
799 octetos			3,121 ms
846 octetos			3,305 ms
893 octetos			3,488 ms
896 octetos		3,500 ms	
940 octetos			3,672 ms
987 octetos			3,855 ms
1024 octetos	4,000 ms	4,000 ms	
1034 octetos			4,039 ms
1081 octetos			4,223 ms
1128 octetos			4,406 ms
1152 octetos		4,500 ms	
1175 octetos			4,590 ms
1222 octetos			4,773 ms
1269 octetos			4,957ms
1280 octetos	5,000 ms	5,000 ms	
1316 octetos			5,141 ms
1363 octetos			5,324 ms
1410 octetos			5,508 ms

Dessa forma, como o número de seqüência de cada pacote é incrementado de forma monotônica para cada pacote transmitido, e os pacotes precisam ser gerados a uma taxa constante para acomodar a característica CBR do fluxo TDM, tem-se que a *timestamp* é simplesmente uma função linear do número de seqüência do pacote, dada pela expressão (2.6). , o que torna a sua transmissão completamente dispensável.

$$Timestamp_n = (NS_n - NS_0).TempoPacote + Timestamp_0 \quad (2.6)$$

Onde:

$NS_0$  é o número de seqüência inicial da sessão, aleatório;

$NS_n$  é o número de seqüência do  $n$ ésimo pacote recebido;

$TempoPacote$  é o intervalo de dados TDM contido no pacote (Tabela 2.2);

$Timestamp_0$  é a *timestamp* inicial definida para a sessão, aleatória;

$Timestamp_n$  é a *timestamp* do  $n$ ésimo pacote recebido.

Assim, a expressão (2.6) pode ser utilizada para determinação da *timestamp* a partir do número de seqüência recebido, permitindo a utilização da técnica de sincronização adaptativa proposta em [AWEYA2004] sem necessidade de utilização do protocolo RTP, conforme apresentado na Figura 2.15, adaptada daquele trabalho, e descrito a seguir:

- a) O receptor gera a *timestamp* para cada pacote recebido, a partir do NS recebido, do NS inicial e da *timestamp* inicial, e armazena a mesma no *buffer* de *timestamps*;
- b) Essa *timestamp* é utilizada como entrada do PLL para a regeneração do relógio no receptor, de forma sincronizada com o relógio de transmissão no PE de origem;
- c) O relógio regenerado é utilizado para controle da taxa de bits de saída e também para incremento do contador do PLL, carregado inicialmente com a *timestamp* gerada a partir do primeiro NS recebido;
- d) O comparador associado ao mecanismo de controle do fluxo de bits de saída compara a *timestamp* regenerada pelo contador do PLL, deslocada pelo valor de atraso esperado, com a *timestamp* gerada a partir do NS recebido, que foi armazenada no *buffer*, liberando os dados TDM para a interface de saída quando estas forem coincidentes.

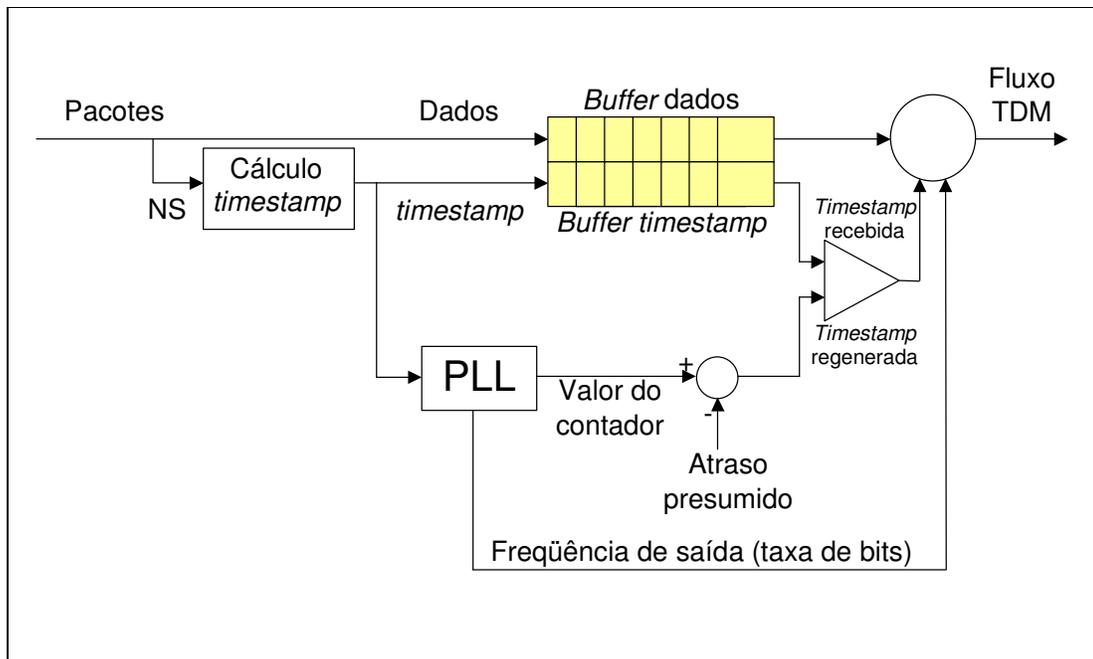


Figura 2.15 - Mecanismo de recuperação do relógio baseado em *timestamp* e PLL.

Entretanto, uma análise mais aprofundada do mecanismo acima permite observar que não é efetivamente necessário o cálculo explícito da *timestamp*, uma vez que esta é uma função linear do número de seqüência tanto quando é recebida como quando é regenerada, sendo possível realizar todo o processo de sincronização, incluindo a operação do PLL e a comparação final para liberação do fluxo de bits de saída, exclusivamente com base nesse número recebido. Associando esse fato ao novo mecanismo de seqüenciamento proposto na seção 2.3.2, onde a análise do número de seqüência determina diretamente a posição de escrita no *jitter buffer*, é possível simplificar ainda mais o mecanismo de sincronização do fluxo TDM no receptor, conforme apresentado na Figura 2.16 e descrito a seguir:

- a) O NS recebido é utilizado diretamente como entrada do PLL para a regeneração do relógio no receptor, sendo também analisado para definição da posição de armazenamento no *buffer*, conforme algoritmo de seqüenciamento proposto;
- b) O relógio regenerado é utilizado para controle da taxa de bits de saída, através de um escalonador, e também para incremento do contador do PLL, carregado inicialmente com o primeiro NS recebido;
- c) O comparador associado ao escalonador compara o NS regenerado ( $N_{\text{PRESUMIDO}}$ ) pelo contador do PLL, deslocado por uma determinada quantidade de pacotes  $N_{\text{ESPERA}}$ , com o NS recebido em cada pacote, armazenado no *buffer*, liberando os dados TDM para a

interface de saída quando os mesmos forem coincidentes, o que assegura o sincronismo entre o fluxo TDM em direção ao CE de destino e aquele recebido pelo PE de origem.

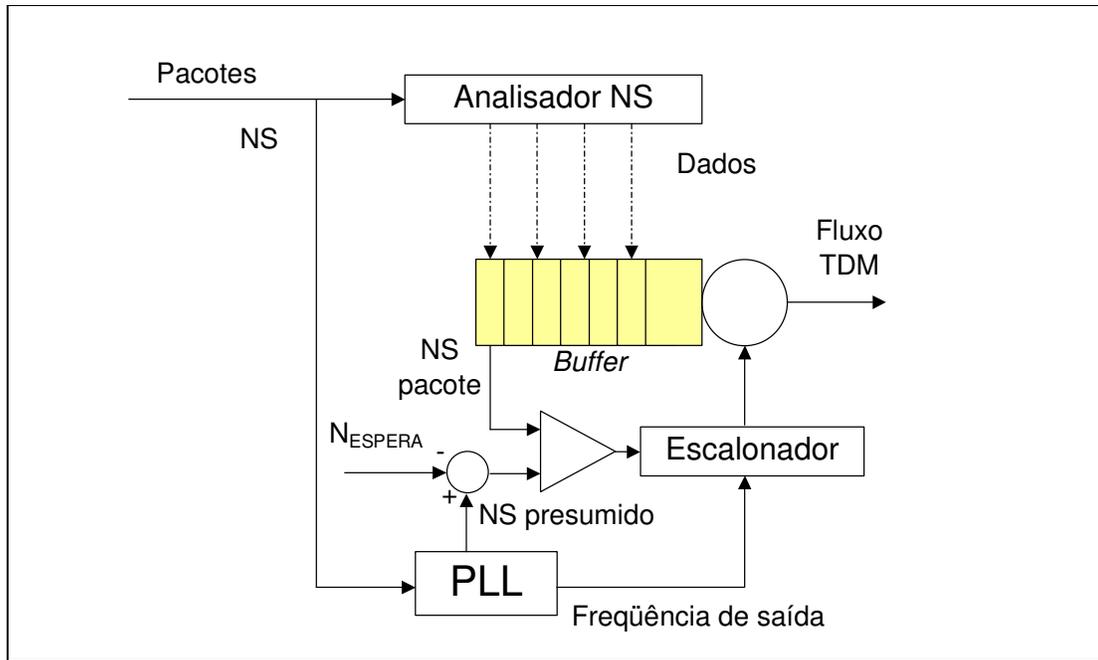


Figura 2.16 - Recuperação do relógio no receptor pelo método adaptativo proposto.

A quantidade de pacotes  $N_{ESPERA}$  define quantos pacotes devem ser armazenados no *jitter buffer* antes que seja iniciada a regeneração do fluxo TDM na saída, correspondendo ao atraso que deve ser introduzido para acomodação da PDV oferecida pela PSN. Tipicamente, como o *jitter buffer* deve operar na condição central em regime permanente, é dado pela Expressão (2.7)

$$N_{ESPERA} = 50\% \cdot TamanhoBuffer(\text{em pacotes}) \quad (2.7)$$

No Capítulo 4 será descrita em maiores detalhes a operação do mecanismo de sincronização apresentado na Figura 2.16, incluindo a descrição detalhada dos blocos que compõe o PLL e sua fundamentação matemática, bem como a proposta desenvolvida nesse trabalho para utilização de técnicas de controle clássico visando ao aprimoramento do seu desempenho, a fim de atender aos requisitos de sincronismo para a utilização da tecnologia TDMoIP<sup>®</sup> como alternativa para *broadcasting IPTV*.

## **2.4 - COMPARAÇÃO ENTRE AS TECNOLOGIAS TDMoIP® E VoIP**

Em princípio, a tecnologia TDMoIP® é bem mais simples que VoIP por ser completamente transparente à sinalização e aos protocolos de voz e dados, mesmo quando estes são proprietários, bem como em função da inexistência de modificações nos equipamentos terminais. Por mais que venha sendo rapidamente desenvolvida nos últimos anos, a tecnologia VoIP ainda apresenta algumas questões em aberto em relação a diversidade de protocolos, compatibilização entre os formatos de mídia existentes e desenvolvimento de mecanismos eficientes para tradução entre os mecanismos de sinalização envolvendo a RFTC tradicional, o que vem limitando a criação de serviços e funcionalidades às quais os clientes já estão acostumados, enquanto TDMoIP® aproveita todas as funcionalidades existentes das centrais públicas e privadas para comutação de voz, de forma transparente.

De maneira geral, entende-se que VoIP seja uma alternativa interessante para as novas implementações no transporte de voz, onde é necessário construir uma outra rede e adquirir novos equipamentos, enquanto TDMoIP® é recomendável para a racionalização dos custos de operação das redes existentes, sobretudo de transporte, bem como para a expansão dessas redes com pequenos investimentos, quando não são desejados serviços e funcionalidades distintas daquelas já existentes.

### **2.4.1 – Semelhanças entre as tecnologias**

Ambas as tecnologias têm como objetivo fundamental a melhoria da eficiência de ocupação de banda dentro da rede de transporte, através da substituição dos circuitos dedicados, utilizados nas redes tradicionais, pela comutação de pacotes, que permite o compartilhamento eficiente dos meios de transmissão entre pacotes de diversas origens e destinos, contendo voz, dados e vídeo, reduzindo drasticamente os custos de transporte.

Os *gateways* VoIP são mais complexos, incluindo DSPs (*Digital Signal Processors*) que permitem facilidades como compressão de voz e supressão de silêncio, que reduzem significativamente o consumo de largura de banda na transmissão e, portanto, apresentam eficiência bem maior que o transporte TDM convencional. Contudo, isso é feito às expensas de uma menor qualidade de voz e maior latência nessa transmissão, pela

necessidade de armazenamento e processamento das amostras de voz, e não quase sempre introduzindo pesado *overhead* na transmissão dos pacotes de voz propriamente ditos.

Já a relativa simplicidade dos *gateways* TDMoIP<sup>®</sup> também permite um melhor aproveitamento da banda de transmissão que o transporte TDM convencional, eliminando parte do *overhead* de estrutura e canais ociosos dentro do entroncamento, mas preserva a qualidade de voz, apresenta latência menor que o VoIP, e é completamente transparente para os terminais, evitando assim a substituição dos equipamentos TDM utilizados, como centrais e PBX digitais, conservando os investimentos realizados e evitando novos custos com treinamento, operação e manutenção dos equipamentos VoIP.

No que tange aos desafios enfrentados, as duas tecnologias buscam soluções bastante similares, pois ambas utilizam redes comutadas a pacotes para transporte de voz, sendo fortemente afetadas pelas características não-determinísticas das mesmas: a perda de pacotes e o atraso variável ao atravessar a rede. Assim, são feitas algumas comparações diretas entre VoIP e as abordagens PWE3, considerando seu comportamento frente a essas características, bem como aspectos importantes do ponto de vista de implementação, como a eficiência de utilização da largura de banda e a complexidade computacional envolvida, que apresentam impacto direto nos custos da solução adotada.

#### **2.4.2 – Sensibilidade às perdas de pacotes**

A perda de pacotes durante o transporte dos sinais de voz provoca falhas ou intervalos significativos no fluxo de amostras, produzindo sons entrecortados, desconfortáveis ou mesmo ininteligíveis. O efeito preciso das perdas de pacotes na qualidade da voz transmitida, bem como o desenvolvimento de algoritmos para mitigação dessas perdas, conhecidos como PLC (*Packet Loss Concealment*) tem sido objeto de estudos avançados dentro da comunidade VoIP, como o trabalho de análise dos efeitos dessa mitigação para diversos codificadores de voz, realizado pelo LabCom/UnB e disponível em [OBANDO2005]. Alguns resultados significativos desses estudos podem ser resumidos, conforme apresentado em [STEIN2003]:

- a) Cerca de 1% de perda de pacotes causam queda na qualidade de voz percebida, baixando do padrão normalmente reconhecido em telefonia fixa para níveis

- equivalentes à qualidade observada em telefonia móvel celular, que utiliza inerentemente algoritmos de compressão;
- b) Acima de 2%, a perda de pacotes é a causa dominante para a deterioração da qualidade de voz, tornando comparáveis entre si, em termos de qualidade, sinais de voz com e sem utilização de algoritmos de compressão;
  - c) Utilizando algoritmos apropriados para mitigação da perda de pacotes, até 5% de perda para sinais de voz não comprimidos ainda permitem qualidade comparável ou superior àquela observada em telefonia móvel celular, com seus algoritmos de compressão.

Contudo, esses resultados não são diretamente aplicáveis à tecnologia TDMoIP<sup>®</sup>, uma vez que o pacote VoIP contém tipicamente entre 80 e 240 amostras do sinal de voz, representando entre 10 e 30 ms de conversação; enquanto os pacotes TDMoIP<sup>®</sup> podem conter uma única amostra, ou um número pequeno delas, representando até 0,625 ms de conversação (para 5 amostras por canal, o limite para um tronco E1 considerando a MTU *Ethernet*), o que reduz bastante sua sensibilidade às perdas de pacote, em função da elevada redundância inerente aos sinais de voz. Assim, de um modo geral, é esperado que os canais de voz transportados através de um entroncamento TDMoIP<sup>®</sup> suportem, individualmente, perdas de pacotes bem superiores a esses limites, sem prejuízo considerável da qualidade de conversação.

Na Tabela 2.3 é apresentado um comparativo entre algumas configurações VoIP e as três abordagens para transporte de circuitos TDM sobre redes de pacotes: TDMoIP<sup>®</sup>, SAToP e CESoPSN. Esse comparativo considera o codificador utilizado; sua designação usual; sua taxa nominal de bits por canal; a avaliação MOS obtida segundo o algoritmo PESQ (*Perceptual Evaluation of Speech Quality*), determinada de acordo com os experimentos realizados no LabCom/UnB, apresentados em [OBANDO2005] - com exceção do codificador G.726, onde foi utilizado o valor nominal apresentado em [COLLINS2001]; a quantidade de amostras de voz por pacote, assumindo pacotes de 1 ms para as tecnologias TDM; e o tempo de convergência, em milissegundos, definido como o tempo que o estado do decodificador leva, após uma ocorrência de perda de quadros (decorrente da perda de pacotes na rede), para convergir ao mesmo estado em que se encontrava o codificador para uma dada amostra, também de acordo com os experimentos realizados no LabCom/UnB.

Tabela 2.3 - Amostras de voz por tecnologia e codificador utilizado.

Tecnologia	Codificador	Designação	Taxa Nominal por Canal (kbps)	MOS (PESQ)	Amostras por pacote	Tempo de Convergência (ms)
VoIP	G.711	PCM	64,00	4,41	160	0
	G.723.1	MP-MLQ	6,30	3,52	60	128
	G.723.1	MP-ACELP	5,30	3,38	60	132
	G.726	ADPCM	32,00	4,00	80	
	G.729	CS-ACELP	8,00	3,78	240	80
	iLBC		13,33	3,85	160	30
	iLBC		15,20		240	
TDMoIP	G.711	PCM	64,00	4,41	7,83	0
CESoPSN	G.711	PCM	64,00	4,41	8	0
SAToP	G.711	PCM	64,00	4,41	8	0

Cabe observar, nessa tabela, que um pacote VoIP utilizando o codificador G.729 transporta trinta vezes mais informação de voz que um pacote TDMoIP<sup>®</sup> típico, para um mesmo canal, o que o torna bem mais sensível a qualquer perda. Além disso, a utilização da codificação G.711, sem compressão, torna as amostras de voz completamente isoladas entre si, não existindo propagação da perda de um pacote para as demais amostras, como descrito pelo tempo de convergência nulo, contra um intervalo de 30ms a 132 ms para os codificadores que utilizam compressão, o que também confere mais robustez às tecnologias PWE3 em relação a VoIP quanto à sensibilidade às perdas de pacote.

#### 2.4.3 – Eficiência de utilização da largura de banda

A tecnologia VoIP apresenta uma maior eficiência de utilização da largura de banda da PSN para a transmissão de um único canal de voz, em virtude da possibilidade de utilização de técnicas de compressão de voz e supressão de silêncio, presentes na nos codificadores utilizados pela maioria das aplicações desse tipo.

Contudo, quando consideramos a questão sob o ponto de vista de um entroncamento, onde existem, por exemplo, 30 canais de voz sendo transportados, essa eficiência deixa de ser tão premente, uma vez que, por característica, os canais VoIP são transportados individualmente sobre a PSN, pois os pacotes de voz são gerados pelos equipamentos terminais, que são conectados diretamente a essa rede de pacotes. Enquanto isso, na tecnologia TDMoIP<sup>®</sup>, o efeito de multiplexação das amostras de voz é intrínseco, pois a

mesma foi desenvolvida exatamente para o transporte de circuitos de entroncamento, contendo múltiplos canais, através de um pseudo-circuito emulado dentro da PSN (PWE3). Com isso, todo o *overhead* necessário para o transporte dos pacotes dentro da rede acaba sendo compartilhado por todos os canais transportados, reduzindo a distância entre as duas tecnologias no que se refere à largura de banda total ocupada no transporte através da PSN, como pode ser observado na Tabela 2.4, que apresenta as taxas efetivas de bits considerando 30 canais independentes VoIP, com cada um dos codificadores e um circuito E1 com os mesmos 30 canais transportados através de PWE3 para as três abordagens, considerando pacotes de 1 ms. Em todos os casos é considerado o protocolo IP (cabeçalho de 20 octetos), com transporte UDP (cabeçalho de 8 octetos), lembrando que para as aplicações VoIP e a abordagem CESoPSN é utilizado também o protocolo RTP (cabeçalho de 12 octetos).

Cabe observar, nessa tabela, que para o mesmo codificador G.711, as três tecnologias baseadas em PWE3 são mais eficientes que a tecnologia VoIP em decorrência do efeito da multiplexação sobre o *overhead*. Além disso, a eficiência do codificador VoIP G.729 (CS-ACELP), sobre o G.711 utilizado em TDMoIP®, que é nominalmente oito vezes superior, cai para menos de quatro vezes quando é considerada a taxa efetiva de bits por canal, efeito que se repete para os demais codificadores.

Tabela 2.4 - *Overhead* e taxa efetiva por tecnologia e codificador utilizado.

Tecnologia	Codificador	Designação	Taxa Nominal por Canal (kbps)	Bits de Voz por Canal	Bits de <i>Overhead</i> por Canal	Pacotes por segundo (pps)	<i>Overhead</i> por Canal (%)	Taxa Efetiva na PSN por Canal (kbps)
VoIP	G.711	PCM	64,00	1280	320	50,0	20%	80,0
	G.723.1	MP-MLQ	6,30	189	320	33,3	63%	17,0
	G.723.1	MP-ACELP	5,30	158	320	33,5	67%	16,0
	G.726	ADPCM	32,00	640	320	50,0	33%	48,0
	G.729	CS-ACELP	8,00	240	320	33,3	57%	18,7
	iLBC		13,33	304	320	43,8	51%	27,4
	iLBC		15,20	400	320	38,0	44%	27,4
TDMoIP	G.711	PCM	64,00	62,7	8,8	1.021,3	12%	73,0
CESoPSN	G.711	PCM	64,00	64,0	11,7	1.000,0	15%	75,7
SAToP	G.711	PCM	64,00	64,0	8,5	1.000,0	12%	72,5

Evidentemente, a situação apresentada é um caso particular de análise, podendo ser utilizadas técnicas de compressão do cabeçalho IP/UDP, que reduziriam o *overhead* para o

transporte dos canais VoIP, da mesma forma que existe um tráfego adicional, não considerado, do protocolo RTCP nas tecnologias que utilizam RTP.

Esses resultados apontam problemas com a utilização de aplicações VoIP em substituição aos circuitos E1 convencionais no ambiente dos clientes, à medida que cresce a quantidade de canais transportados, em função do excesso de *overhead* gerado sobre a PSN, pois cada canal é tratado de forma independente. Assim, têm surgido soluções contemplando a multiplexação desses canais no nível do protocolo RTP, acopladas diretamente a equipamentos PBX, permitindo que os mesmos sejam tratados como um arranjo único, o que melhoraria o *overhead* de transporte em cerca de 15% [RAD2003]. Dentro das redes IP das operadoras, no entanto, o problema é menos significativo, pois os canais de voz são previamente agregados em troncos TDM convencionais e encaminhados à PSN através de *trunking gateways*.

#### **2.4.4 – Complexidade computacional**

Em uma análise preliminar das duas situações apresentadas no Capítulo 1, Figuras 1.6 e 1.7, onde é realizada a comunicação entre dois telefones convencionais utilizando, respectivamente, as tecnologias VoIP e TDMoIP<sup>®</sup>, pode-se observar que a complexidade inerente a uma aplicação VoIP é bastante superior, conforme será descrito a seguir.

Em princípio, para que a comunicação com a aplicação VoIP seja possível entre duas RTFC distintas, em um ambiente de operadora, são necessários, pelo menos, cinco equipamentos, além dos roteadores e demais elementos da própria PSN:

- dois *trunking gateways*, um em cada RTFC, que recebem o tráfego de voz analógica gerado pelos telefones, já codificado pelas centrais da RTFC em canais digitais a 64kbps (G.711), dentro de um circuito TDM (tronco) de entrada. Uma vez recebidos, os canais de voz são convertidos, através de um arranjo de DSPs apropriado, em amostras de voz correspondentes ao codificador desejado (G.729, por exemplo), que por sua vez são encapsuladas em pacotes de voz para transporte dentro da rede IP. Chegando ao *trunking gateway* de destino, as amostras de voz são retiradas dos pacotes e reconvertidas pelos DSPs em canais digitais a 64kbps, que são transmitidos

através de um circuito TDM (tronco) de saída para a central de destino, onde o canal é decodificado em voz analógica que é recebida pelo outro telefone.

- dois *signaling gateways*, um em cada RTFC, que recebem a sinalização ISUP (SS7) correspondente ao circuito TDM (tronco) de entrada e realizam a sua conversão, também através de DSPs, gerando uma sinalização equivalente para transporte dentro da rede IP e reconvertendo os pacotes de sinalização recebidos em mensagens ISUP (SS7) correspondentes ao circuito TDM (tronco) de saída, de forma a assegurar a continuidade de sinalização entre nas duas RTFC.
- um *media gateway controller* ou *softswitch*, para controlar o fluxo de pacotes, mapear números de telefone em endereços e definir o tipo de codificação utilizado entre os dois *trunking gateways*, bem como acompanhar o estado da chamada em curso, interpretando as mensagens recebidas das RTFC através dos *signaling gateways*.

Além disso, para que esses cinco equipamentos possam comunicar-se adequadamente entre si no desempenho das funções necessárias para assegurar a comunicação entre os dois telefones, são necessários, pelo menos, mais três protocolos, que precisam estar implementados em seus processadores, além dos protocolos IP e UDP da PSN:

- RTP/RTCP, para controlar o fluxo de pacotes de voz entre os dois *trunking gateways*, assegurando a sincronização das amostras de voz durante a conversação;
- MGCP ou MEGACO/H.248, para realizar a comunicação entre os dois *trunking gateways* e o *softswitch*, assegurando o controle total deste sobre a chamada em curso;
- SIGTRAN, para realizar a comunicação entre os dois *signaling gateways* e o *softswitch*, assegurando que mensagens de sinalização possam ser recebidas e interpretadas por estes, a fim de acompanhar e controlar o estado da chamada em curso.

Enquanto isso, para que a comunicação com a aplicação TDMoIP<sup>®</sup> seja possível entre as mesmas duas RTFC distintas, são necessários, além dos roteadores e demais elementos da própria PSN, apenas dois *gateways* TDMoIP<sup>®</sup>, um em cada RTFC, que também recebem o tráfego de voz analógica gerado pelos telefones, codificado em canais digitais a 64kbps (G.711), dentro de um circuito TDM (tronco) de entrada. Uma vez recebido em cada *gateway* TDMoIP<sup>®</sup>, o canal de voz associado à conversação compõe, juntamente com os demais canais presentes no circuito TDM, incluindo aqueles destinados à sinalização entre as RTFC, um único fluxo contínuo de bits, que é subdividido a intervalos regulares

previamente configurados pelo processador do *gateway*. Essas seqüências de bits são encapsuladas em pacotes para transporte através da PSN, sendo agregada a cada pacote uma Palavra de Controle. Chegando ao *gateway* TDMoIP<sup>®</sup> de destino, as seqüências de bits recebidas são retiradas dos pacotes e reagrupadas pelo processador do mesmo de acordo com o seqüenciamento estabelecido pela palavra de controle. Assim, o fluxo contínuo de bits presente na respectiva origem é recuperado para os circuitos TDM (trancos) de saída em cada *gateway* TDMoIP<sup>®</sup>, dentro dos quais estão inseridos os canais digitais a 64kbps correspondentes aos telefones em conversação, que são transmitidos até a respectiva central de destino e decodificados em voz analógica, recebida pelo outro telefone.

Dessa forma, são realizados apenas processamentos locais e independentes entre si, como montagem/desmontagem de pacotes, inserção da Palavra de Controle, ponteiros AAL1, e cálculo de CRCs, não sendo necessários protocolos de comunicação entre os dois *gateways* ou entre esses e a RTFC, uma vez que a sinalização é transportada de forma transparente.

Existe, entretanto, um protocolo de Conectividade de Operação e Manutenção (OAM *Connectivity*), com objetivo de reconhecer uma conexão válida antes de iniciar o fluxo de dados TDM, evitando sobrecarga de pacotes na PSN sem que o pseudo-circuito esteja estabelecido e operacional com o *gateway* de destino para seu recebimento, mas esse protocolo não é essencial para funcionamento da tecnologia.

Deve ser observado que, na Figura 1.7, a comunicação não acontece através das RTFC, mas pela conexão direta dos telefones ao *gateway* TDMoIP<sup>®</sup>. Evidentemente, isso é possível apenas quando o *gateway* possui uma interface FXS onde é conectado o telefone. Caso contrário, é também necessária a codificação prévia em canais digitais a 64kbps (G.711), funcionalidade presente nos equipamentos PBX digitais existentes.

Existe razoável complexidade computacional na recepção dos fluxos TDMoIP<sup>®</sup> com sincronização adaptativa, em virtude dos mecanismos de recuperação de relógio, assim como no caso de utilização de técnicas especiais para a mitigação das perdas de pacote durante a regeneração do fluxo TDM, mas essa complexidade nas aplicações baseadas em VoIP demonstra-se bem superior.

A maior complexidade computacional é refletida nos investimentos necessários para implantação das duas soluções, estimados [RAD2003], para um enlace T1 (24 canais de voz) com qualidade de voz equivalente à oferecida por uma operadora, em:

- VoIP: US\$ 15.000,00/porta;
- TDMoIP®: US\$ 1.500,00/porta (confirmado pela RAD do Brasil).

#### 2.4.5 – Latência e requisitos para transporte do tráfego de voz

A latência (*delay*) ou atraso total fim a fim experimentado pelos pacotes em uma dada direção de conversação, assim como a variação desse atraso (PDV), têm impacto significativo sobre a qualidade do transporte dos sinais de voz interativos utilizados em telefonia, influenciando diretamente na percepção dos interlocutores. Um atraso muito elevado compromete o dinamismo da conversação e gera colisões, ou seja, ambos os interlocutores acabam falando ao mesmo tempo, prejudicando o entendimento na conversação. Um estudo aprofundado dos efeitos do atraso e variação do atraso em aplicações de voz é encontrado em [BARRETO2001], mas pode ser considerada, para análise do seu impacto em telefonia, a escala apresentada na Tabela 2.5, elaborada a partir dos requisitos apresentados na Recomendação ITU-T G.114.

Tabela 2.5 – Efeito do atraso na qualidade do sinal de voz.

Latência (ms)	Qualidade do sinal reproduzido
0	Excelente
< 150	Bom
< 250	Regular
< 350	Pobre
< 450	Inaceitável

Assim, são definidas pela Recomendação ITU-T G.114 três faixas distintas para o atraso experimentado em uma dada direção de conversação, assumindo, nesses casos, um controle adequado do eco no circuito, que se torna necessário para atrasos superiores a 25 ms. Essas informações, juntamente com as faixas normalmente aplicadas às redes privadas de comunicação, estão compiladas na Tabela 2.6, e constituem os requisitos de latência para quaisquer aplicações de transporte de voz através de redes de pacotes, independentemente da tecnologia utilizada.

Tabela 2.6 – Especificações de latência máxima para redes públicas e privadas.

<b>Latência (ms)</b>	<b>Descrição com base na Recomendação G.114</b>
< 25	Aceitável sem utilização de canceladores de eco (G.131).
< 150	Aceitável para a maioria das aplicações dos usuários, presumindo controle adequado de eco.
150-400	Aceitável desde que os administradores da rede tenham consciência do tempo de transmissão e do impacto nas aplicações dos usuários.
> 400	Inaceitável para planejamento de rede, pode ser excedido excepcionalmente.
<b>Latência (ms)</b>	<b>Descrição normalmente aplicada para redes privadas</b>
< 200	Aceitável para redes privadas de comunicação.
< 250	Limite para redes privadas de comunicação.

Na Tabela 2.7 é apresentado novo comparativo entre algumas configurações VoIP e as três abordagens para transporte de circuitos TDM sobre redes de pacotes: TDMoIP<sup>®</sup>, SAToP e CESoPSN, descrevendo seus atrasos característicos. Esse comparativo considera o codificador utilizado; sua designação usual; o atraso de codificação (COD), o atraso algorítmico associado a alguns tipos de codificadores (ALG), o atraso de decodificação (DEC) e o atraso de processamento (PROC), conforme valores típicos apresentados em [CISCO2005] para cada uma dessas etapas, na tecnologia VoIP; e pacotes de 1 ms com atraso de processamento também de 1 ms [RAD2005], para as três abordagens PWE3. Nas duas últimas colunas são apresentados o atraso total, correspondente à soma dos anteriores, que representa o atraso fixo introduzido na comunicação, exclusivamente em decorrência da tecnologia e codificador utilizado; e a parcela da máxima latência admissível, de acordo com a Recomendação ITU-T G.114, para uma comunicação de voz unidirecional.

Tabela 2.7 - Latência fixa por tecnologia e codificador utilizado.

<b>Tecnologia</b>	<b>Codificador</b>	<b>Designação</b>	<b>Atraso COD (ms)</b>	<b>Atraso ALG (ms)</b>	<b>Atraso DEC (ms)</b>	<b>Atraso de PROC (ms)</b>	<b>Atraso Fixo (ms)</b>	<b>% Latência Admissível (150 ms)</b>
<b>VoIP</b>	<b>G.711</b>	PCM	20,000	-	2,000	2,500	24,5	16%
	<b>G.723.1</b>	MP-MLQ	30,000	7,500	3,000	5,000	45,5	30%
	<b>G.723.1</b>	MP-ACELP	30,000	7,500	3,000	5,000	45,5	30%
	<b>G.726</b>	ADPCM	20,000	-	2,000	2,500	24,5	16%
	<b>G.729</b>	CS-ACELP	10,000	5,000	1,000	2,500	18,5	12%
	<b>ILBC</b>		20,000	-	2,000	2,500	24,5	16%
	<b>ILBC</b>		30,000	-	3,000	2,500	35,5	24%
<b>TDMoIP</b>	<b>G.711</b>	PCM	0,979	-	0,979	1,000	3,0	3%
<b>CESoPSN</b>	<b>G.711</b>	PCM	1,000	-	1,000	1,000	3,0	3%
<b>SAToP</b>	<b>G.711</b>	PCM	1,000	-	1,000	1,000	3,0	3%

Cabe observar, nessa tabela, que os codificadores utilizados nas aplicações VoIP comprometem, com atrasos fixos, entre 12% e 30% da latência máxima admissível no transporte, enquanto as tecnologias PWE3 comprometem apenas 3%, entre quatro e dez vezes menos, permitindo assim a utilização de *jitter buffers* significativamente maiores, com capacidade para acomodação de uma faixa mais larga de variações no atraso sofrido pelos pacotes dentro da PSN, o que lhes confere maior robustez em condições de congestionamento ou ausência de QoS.

Entretanto, dadas as características de agregação inerentes às tecnologias PWE3 para o transporte de voz, na verdade feita através de emulação dos circuitos convencionais, existem outros requisitos rígidos a serem atendidos, definidos pelos padrões dos circuitos TDM convencionais, que não são de fato críticos para a maioria das aplicações VoIP, conforme apresentado na Tabela 2.8.

Tabela 2.8 – Comparação de requisitos PWE3 com relação a VoIP

<b>Requisito</b>	<b>PWE3 (Circuitos TDM)</b>	<b>VoIP</b>
<b>Taxa de bits</b>	2,048 Mbps (E1)	admite maior flexibilidade de taxas
<b>Desvio de frequência</b>	máximo admissível: +/- 50ppm (gera LOS ou OOF)	menos crítico, pois afeta apenas o decodificador (impacto na qualidade MOS)
<b>jitter</b>	máximo: <0,1 us	não existe compromisso com <i>jitter</i> , apenas com PDV

Assim, como não poderia ser diferente para tecnologias com abordagens tão distintas para o mesmo problema, são observadas vantagens e desvantagens em cada uma das soluções, recaindo a escolha nos requisitos e condições da aplicação desejada.

De maneira geral, a tecnologia VoIP apresenta maior flexibilidade, podendo operar de forma integrada com diferentes classes de terminais individuais, como telefones IP, *softphones* em microcomputadores pessoais, terminais analógicos conectados a *gateways* FXS (*Foreign eXchange Subscriber*), PBX conectados a *gateways* FXO (*Foreign eXchange Office*) ou mesmo redes inteiras de comutação convencionais conectadas através de *trunking gateways*, assegurando total acessibilidade e permitindo conversações

múltiplas entre eles, mas isso a custo de uma grande quantidade de elementos e protocolos e, não raramente, excessivo *overhead* e latência, exigindo um gerenciamento adequado da QoS e dos recursos oferecidos pelas PSN utilizadas no transporte da voz em pacotes.

Por outro lado, as abordagens baseadas em PWE3, como a TDMoIP<sup>®</sup>, apresentam menos flexibilidade, mas oferecem em troca uma complexidade significativamente menor, maior robustez em relação à QoS e maior eficiência de recursos, sobretudo investimentos, quando empregadas em soluções ponto a ponto para emulação dos circuitos TDM convencionais através das PSN, como o entroncamento e extensão de circuitos de voz para redes pré-existentes formadas por terminais, PBX ou centrais convencionais, conforme será ilustrado na seção 2.5.1.

## **2.5 - PRINCIPAIS APLICAÇÕES TDM SOBRE IP E *ETHERNET***

Com o objetivo de ilustrar o grande potencial existente para a emulação de circuitos TDM sobre redes de comutação em modo pacotes, são apresentadas a seguir algumas importantes aplicações das tecnologias PWE3 nas redes de comunicação atuais, com especial atenção para o entroncamento e extensão de circuitos de voz, uma vez que a demonstração da tecnologia TDMoIP<sup>®</sup> como alternativa de menor custo, maior simplicidade e melhor eficiência para esse tipo de aplicação constitui um dos objetivos desse trabalho.

### **2.5.1 – Entroncamento de voz e extensão de circuitos**

O crescente interesse em utilização das tecnologias IP para o transporte de serviços TDM evidencia a necessidade de aplicações que atendam exatamente aquelas necessidades para as quais esses circuitos TDM foram originalmente concebidos: o entroncamento entre centrais e a transmissão dos canais de voz entre redes interurbanas. Isso vem acontecendo tanto através da substituição das redes dedicadas de transmissão por alternativas mais baratas, baseadas na comutação em modo pacotes, como pela extensão desses circuitos a novos pontos de presença, com menores investimentos decorrentes do compartilhamento das redes com serviços de dados e vídeo.

A Figura 2.17 apresenta uma configuração típica de entroncamento interurbano entre duas centrais telefônicas, através de uma rede TDM tradicional baseada em comutação de circuitos, onde os enlaces digitais de transmissão, tipicamente  $n \times E1$  (PDH), chegam ao nó da rede interurbana mais próximo e são inseridos na estrutura STM-1 que trafega dentro de um anel SDH, através de dispositivos *cross-connects* existentes nesses nós. Além disso, a sinalização entre essas centrais é trocada através de uma rede SS7, normalmente já operando em modo pacotes, formada pelos respectivos PTS (Ponto de Transferência de Sinalização) a elas conectados.

Dessa forma, cada enlace E1 de entroncamento possui um circuito dedicado entre as duas centrais, devidamente provisionado dentro das unidades tributárias que trafegam dentro do anel SDH, independentemente de estar ou não sendo utilizado num determinado momento, reduzindo a eficiência da banda de transmissão SDH, pois mesmo que outra central tenha, num dado momento, necessidade de tráfego para ocupação desses canais, eles não podem ser ocupados até que a configuração de provisionamento do STM-1 seja explicitamente modificada.

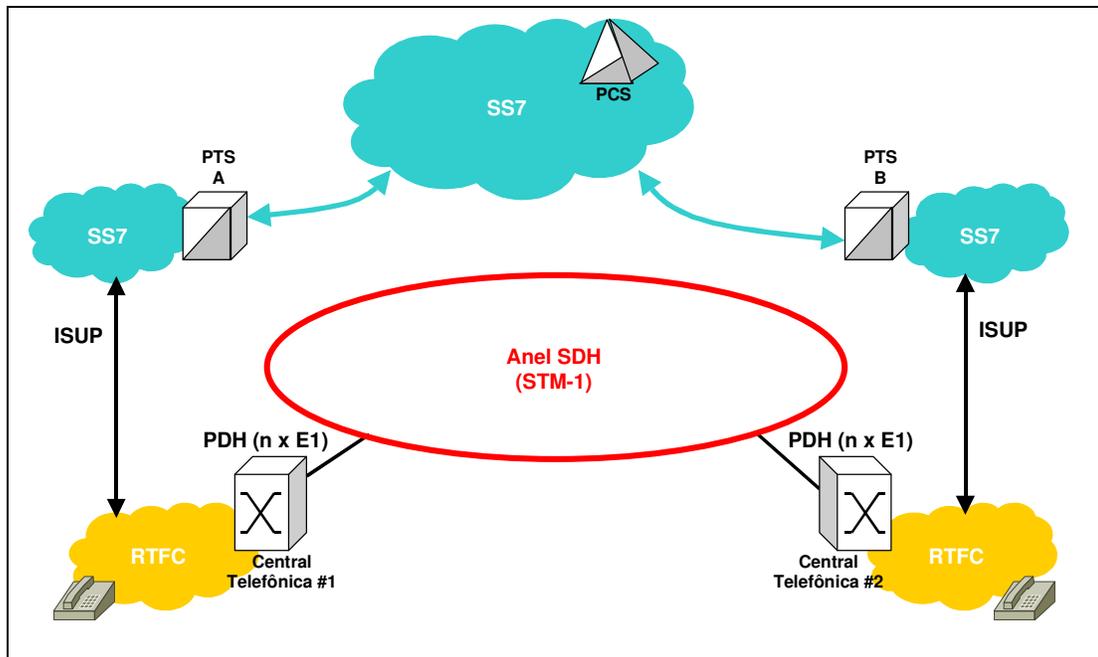


Figura 2.17 - Entroncamento TDM convencional.

Assim, essa configuração representa uma grande oportunidade de melhoria da eficiência de transmissão pela utilização de uma rede comutada por células ou pacotes, ao invés do

entroncamento convencional comutado a circuitos, mesmo que esta rede seja suportada pelo próprio anel SDH original. Esse foi o grande objetivo e a essência do desenvolvimento das redes ATM, baseadas no chaveamento de células através do meio de transmissão e permitindo a integração dos diversos serviços de voz, dados e vídeo na mesma rede, com melhor aproveitamento da largura de banda disponível.

Contudo, enquanto as redes ATM estavam sendo desenvolvidas e implantadas, envolvendo vultosos investimentos por parte das operadoras mundiais, o crescimento das aplicações e redes baseadas na comutação em modo pacotes, como IP e *ethernet*, atingia uma velocidade e abrangência nunca vistas até então. Assim, nada mais natural que surgissem propostas para o entroncamento de circuitos TDM de voz e outros serviços de faixa estreita sobre essas redes de pacotes, de forma a assegurar um melhor aproveitamento da banda de transmissão disponível sem incorrer na alta complexidade e nos custos elevados oferecidos pelas redes ATM.

Surgiram então tecnologias como VoIP e PWE3, desenvolvidas especificamente para esse tipo de aplicação, cujas abordagens detalhadas são realizadas em [AWEYA2003] e serão brevemente descritas nessa seção.

A Figura 2.18 ilustra a abordagem do problema de entroncamento interurbano entre duas centrais utilizando a tecnologia VoIP e seus elementos fundamentais nas camadas de mídia e controle, incluindo conversão da sinalização SS7, que já foram abordados anteriormente; oferecendo como principal característica a grande flexibilidade, representada pela possibilidade de acesso, para os terminais localizados nessas centrais, a qualquer telefone IP ou *softphone* conectado à PSN utilizada nesse entroncamento, que poderiam estar, virtualmente, em qualquer lugar do mundo. Contudo, cabe esclarecer que aplicações dessa natureza são suportadas, na quase totalidade dos casos, por redes de pacotes desenvolvidas e controladas com essa finalidade, onde é assegurada a QoS necessária, tais como o *backbone* IP/MPLS de uma grande operadora, não sendo concebível a utilização da Internet pública, por exemplo, como rede de transporte para entroncamento de voz.

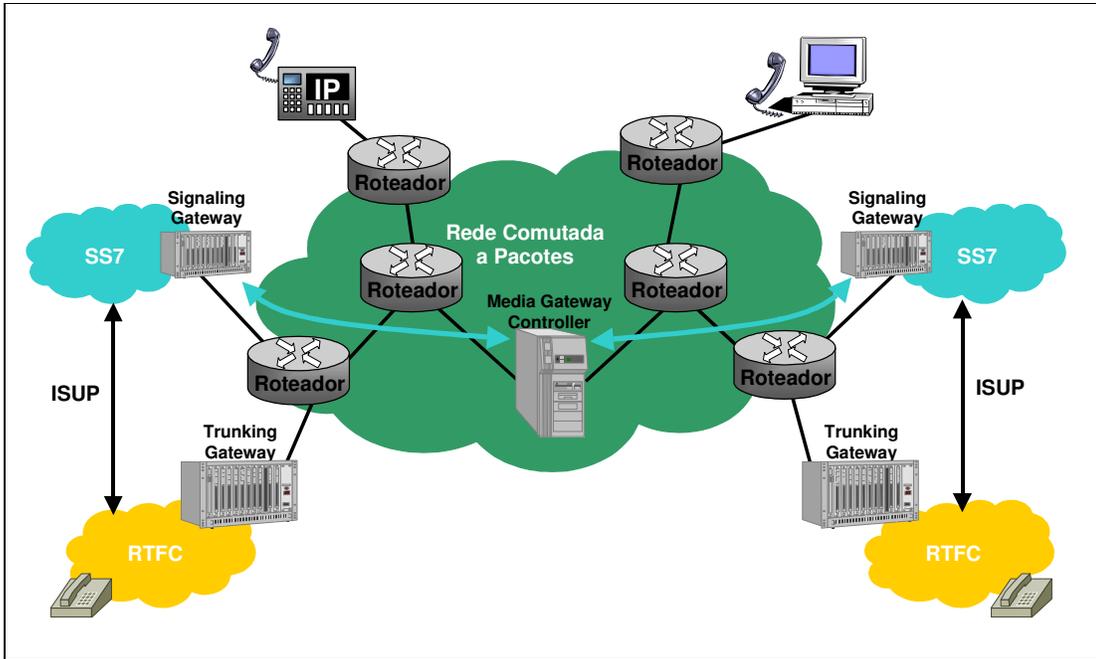


Figura 2.18 - Entroncamento de serviços através de VoIP.

A aplicação da tecnologia TDMoIP<sup>®</sup> para o problema de entroncamento interurbano entre duas centrais é ilustrada na Figura 2.19, destacando que a emulação fim a fim dos circuitos TDM a tornam completamente transparente para os equipamentos terminais, que continuam recebendo apenas enlaces E1.

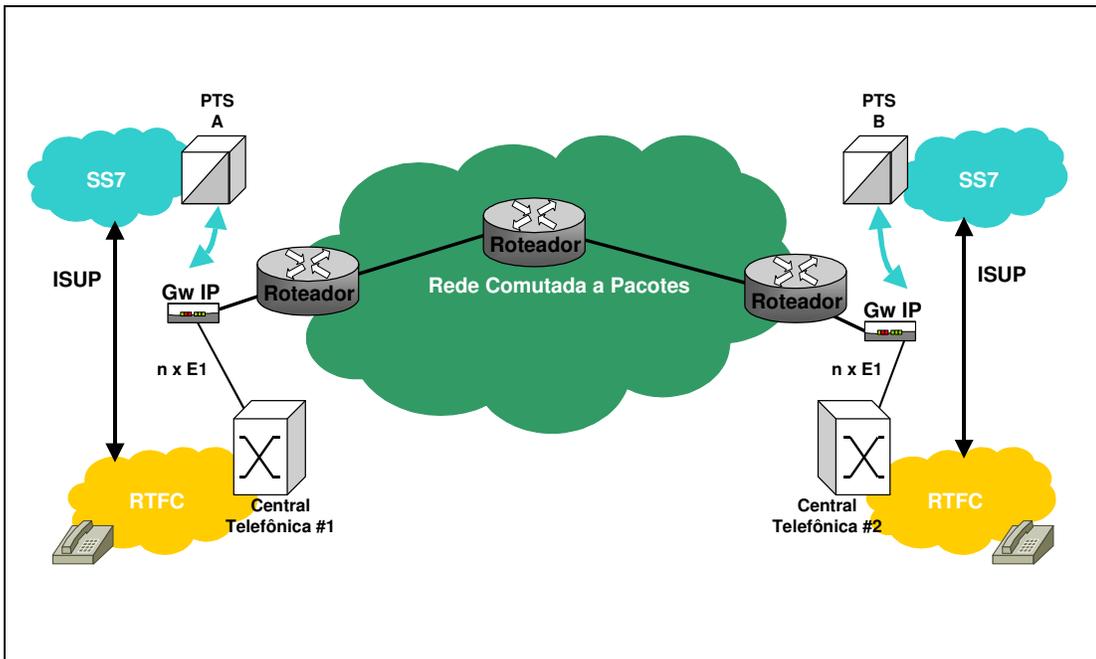


Figura 2.19 - Entroncamento IP estático utilizando TDMoIP<sup>®</sup>.

Todas as funções necessárias para a emulação dos circuitos, como a conversão do fluxo contínuo de bits em pacotes, a regeneração desse fluxo no destino e a recuperação do relógio de transmissão são desempenhadas exclusivamente pelos respectivos *gateways* TDMoIP<sup>®</sup>, instalados nas extremidades do entroncamento.

A limitação dessa configuração é a ausência de flexibilidade decorrente da configuração estática dos circuitos ponto-a-ponto, pois não existe comutação dentro do entroncamento, sendo necessário um circuito TDM emulado para cada par de centrais entroncadas. A seleção entre os circuitos fica a cargo da matriz de comutação de cada central, da mesma forma que aconteceria no entroncamento por circuitos TDM convencionais.

Outra alternativa, mais complexa, seria a utilização de *gateways* TDMoIP<sup>®</sup> com capacidade de comutação (*switches*), transferindo para o PE de origem parte da inteligência de seleção entre os pseudo-circuitos para a determinação da rota de entroncamento, através de endereçamento múltiplo dos pacotes encaminhados, como ilustrado na Figura 2.20.

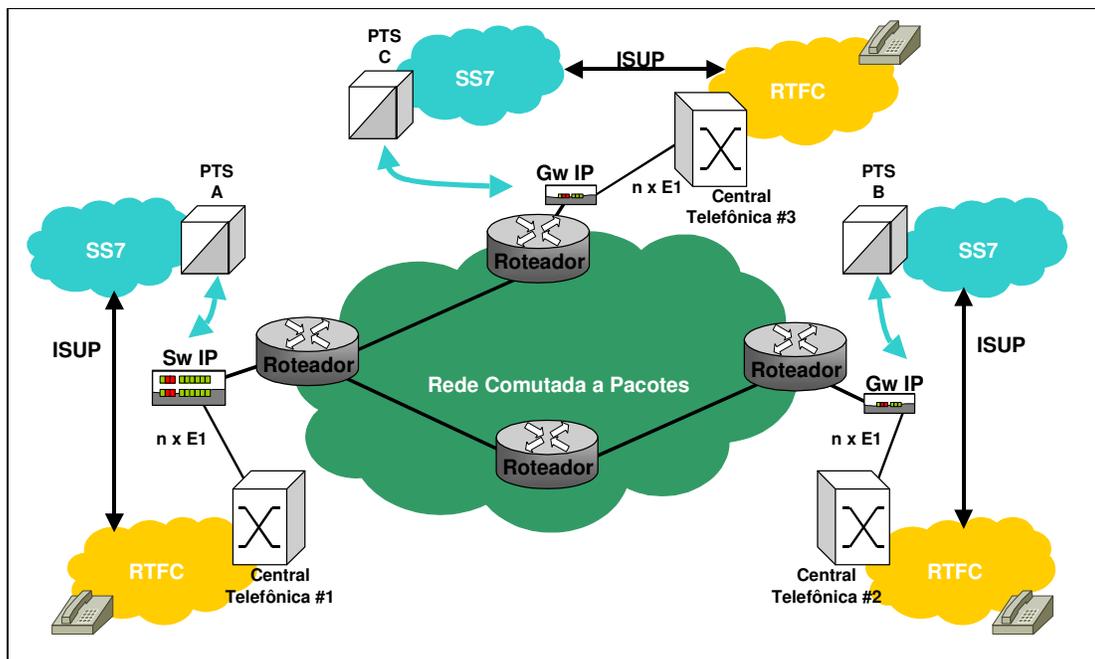


Figura 2.20 - Entroncamento IP com comutação utilizando TDMoIP<sup>®</sup>.

Finalmente, em função da ausência de flexibilidade da abordagem PWE3 para as aplicações de entroncamento, é proposta em [AWEYA2003] uma alternativa envolvendo interoperabilidade entre entroncamento TDMoIP<sup>®</sup> e serviços VoIP, ilustrada na Figura

2.21. Nessa proposta seria feito, no Plano de Controle executado pelos *gateways* PWE3, o mapeamento dos protocolos de sinalização e controle de chamadas utilizados pelas aplicações VoIP, como o SIP, em mensagens de sinalização compatíveis com os terminais ISDN (*Integrated Services Digital Network*), que assim poderiam ser interpretadas diretamente pelas centrais de comutação, assegurando a acessibilidade dos terminais conectados às mesmas a telefones IP e *softphones*, como se fossem terminais ISDN.

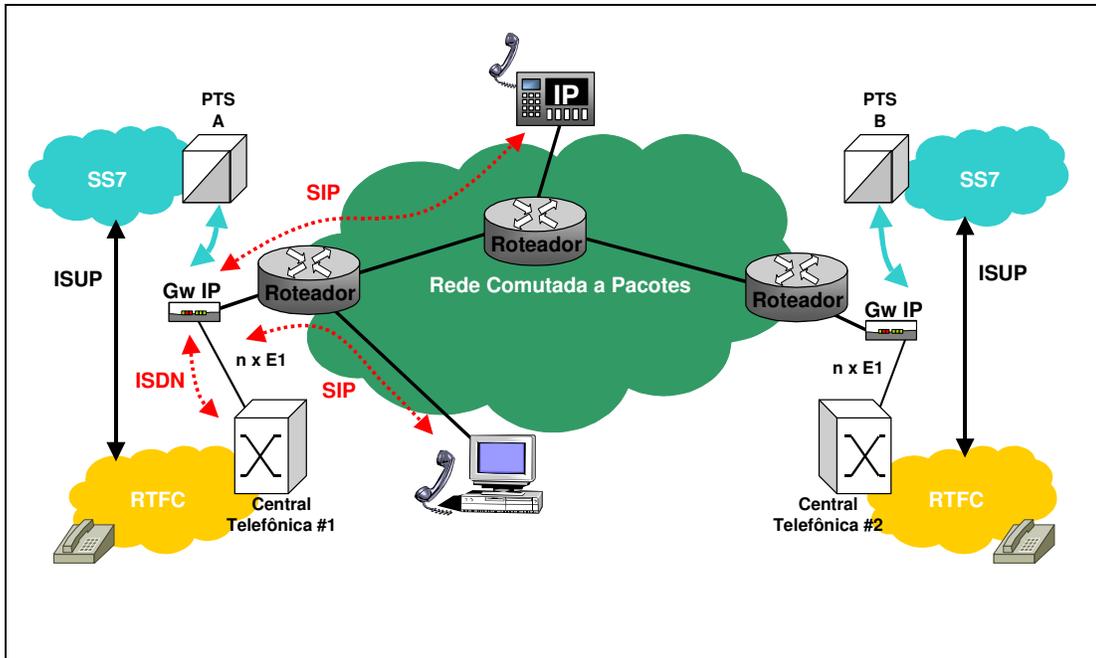


Figura 2.21 - Entroncamento IP utilizando TDMoIP<sup>®</sup> com interoperabilidade VoIP.

### 2.5.2 – Circuitos E1/T1 sobre redes de pacotes como rede de acesso local

No cenário competitivo atual, onde as redes de acesso de maior capilaridade estão, em sua quase totalidade, sob o controle das concessionárias ou *incumbents*, torna-se complexo e dispendioso para qualquer nova empresa entrante o oferecimento de serviços de voz além da abrangência de sua própria rede, freqüentemente limitada em função dos elevados investimentos necessários para construção das redes de acesso.

Nas condições da regulamentação brasileira, esses serviços são oferecidos através do aluguel de enlaces dedicados, normalmente E1, na modalidade conhecida como EILD (Exploração Industrial de Linha Dedicada), através de contratos de interconexão exigidos e supervisionados pela ANATEL, que representam, na prática, o equilíbrio de interesses

entre as empresas. Dessa forma, as empresas que não controlam a rede de acesso acabam tendo suas estratégias restritas pelas condições de disponibilidade de suas principais concorrentes, limitando suas ofertas e a própria capacidade de atendimento ao mercado.

Com a utilização das tecnologias PWE3 para emulação transparente de circuitos TDM, qualquer rede de pacotes existente pode ser utilizada para prover esses serviços, a custo reduzido em função do inerente compartilhamento e sem necessidade de modificação nos equipamentos dos clientes ou da própria operadora. A expansão das redes que oferecem conectividade IP e as *ethernets* metropolitanas (*metroethernets*), implantadas originalmente para prover serviços de dados, representam novas e importantes opções de rede de acesso para as operadoras entrantes. Essas operadoras têm, inclusive, a opção de contratar conectividade IP com seus clientes das próprias concessionárias, a custos mais baixos que as linhas dedicadas, construindo sobre esse novo acesso desagregado (*unbundled*) a sua rede multi-serviços, incluindo o transporte de voz através da tecnologia TDMoIP®, como apresentado na Figura 2.22, onde os serviços de voz são oferecidos aos clientes através de uma rede do tipo gigabit *ethernet* metropolitana, completamente integrados aos demais serviços de dados.

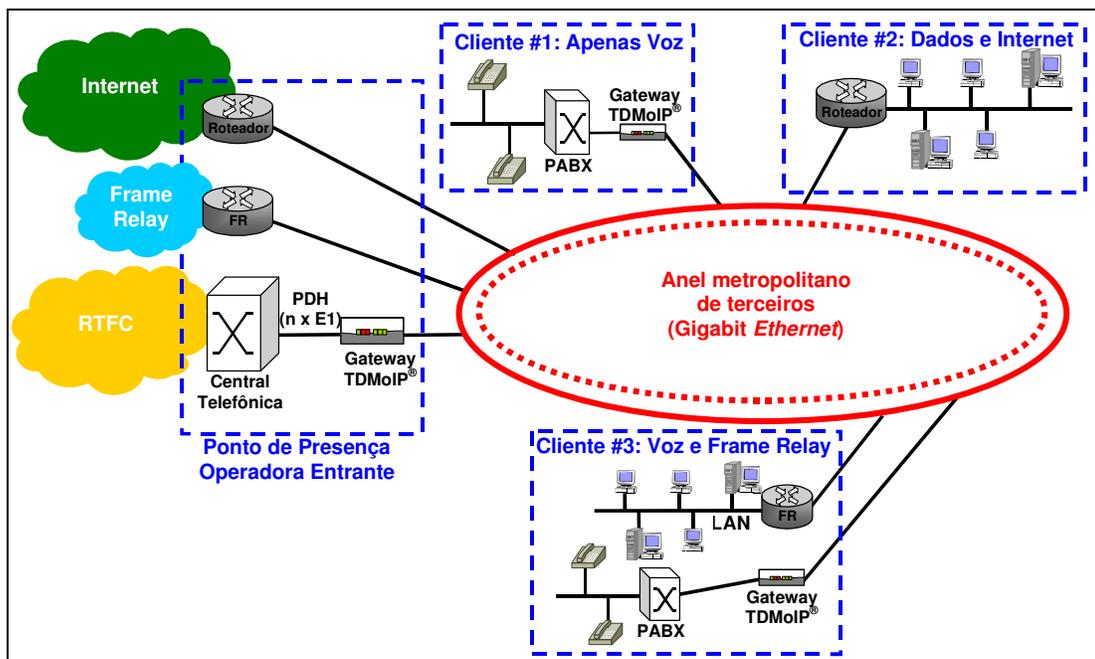


Figura 2.22 – Utilização de *metroethernet* como rede de acesso multi-serviços.

### 2.5.3 – Serviços de linha dedicada E1/T1 sobre redes de pacotes

As operadoras que tradicionalmente não oferecem serviços de voz em virtude dos elevados investimentos envolvidos, tendo focado seus negócios em serviços como conectividade IP, Internet ou formação de redes, têm na tecnologia de emulação de circuitos TDM uma grande oportunidade de rentabilizar sua infra-estrutura.

Essas operadoras possuem, tipicamente, *backbones* metropolitanos ou interurbanos de alta velocidade baseados em tecnologias como IP e MPLS, e não raramente apresentam largura de banda subutilizada em suas redes, em função da forte expansão das redes de dados nos últimos anos, muito além da demanda que vem sendo colocada pelos clientes. Dessa forma, se for possível para a operadora o controle do atraso e da perda de pacotes em seus *backbones*, assegurando uma QoS adequada, a oferta de serviços de linha dedicada pela emulação de circuitos TDM ponto a ponto, como os apresentados na Figura 2.23, é uma excelente alternativa para otimização do uso de suas redes, agilizando o retorno dos seus investimentos e permitindo que sejam praticados preços bastante competitivos com relação aos enlaces E1/T1 tradicionais.

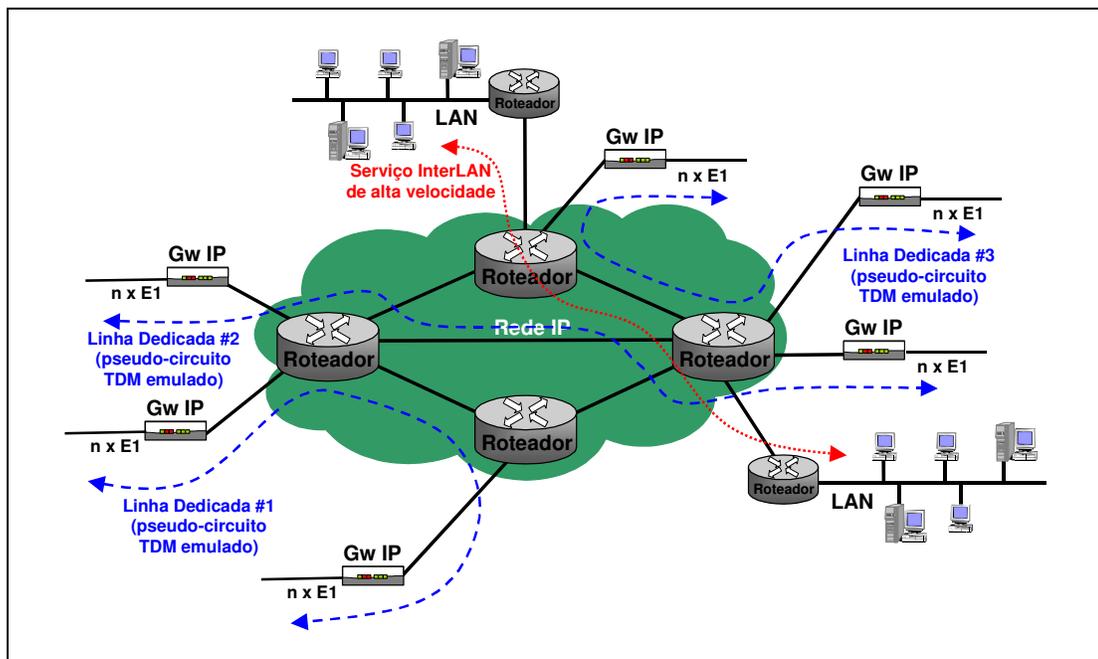


Figura 2.23 – Serviços de linha dedicada sobre redes de dados.

Além disso, as operadoras de comunicação de dados podem posicionar-se de forma exclusiva no mercado, oferecendo enlaces E1 nos Estados Unidos ou enlaces T1 na Europa ou no Brasil, permitindo a implantação imediata de serviços até então impossíveis pela incompatibilidade entre as duas hierarquias.

#### **2.5.4 – Backbone para redes móveis sobre redes de pacotes**

O sucesso da telefonia móvel celular em todo o mundo é o principal fenômeno do século XXI. Na maioria dos países, a quantidade de acessos móveis já supera, com larga vantagem, a quantidade de acessos fixos em serviço, o que significaria o fim da telefonia fixa ao longo dos próximos anos, segundo alguns analistas.

Contudo, as redes de transporte das operadoras móveis são integralmente baseadas nos enlaces TDM convencionais. Todas as conexões entre os principais elementos dessas redes, como BTSs (*Base Transceiver Stations*), BSCs (*Base-Station Controllers*) e MSCs (*Mobile Switching Centers*), são realizadas por circuitos TDM (E1/T1), através de linhas dedicadas alugadas das concessionárias locais, a custos elevados; ou ainda de enlaces de microondas ponto a ponto, que representam investimentos adicionais e enfrentam problemas como as questões ambientais ligadas ao uso de radiofrequência e a saturação do espectro de frequências nos grandes centros urbanos.

Segundo informações do setor [RAD2003], o custo dos enlaces TDM representaria entre 40% e 60% do custo operacional das redes móveis, o que deixa essas operadoras particularmente interessadas em soluções que reduzam seus custos de transporte. Nesse aspecto, a crescente expansão da conectividade IP nas regiões metropolitanas também coloca a tecnologia TDMoIP<sup>®</sup> como uma excelente alternativa para essa aplicação, permitindo o estabelecimento de pseudo-circuitos TDM através das PSN existentes, como por exemplo uma rede de distribuição de TV a cabo que ofereça serviços IP, de forma completamente transparente para os equipamentos de transmissão utilizados pelas estações-base, que continuariam, em suas interfaces, trafegando apenas enlaces E1/T1 convencionais.

Dessa forma, a abordagem PWE3 permite que todo o *backbone* de uma rede móvel seja implementado através de redes IP ou *metroethernet*, como mostrado na Figura 2.24, sem qualquer modificação nos elementos existentes da rede, reduzindo significativamente os custos associados ao transporte dos canais de voz entre as estações-base, seus controladores e a MSC. Essa solução facilitaria também a migração dessa rede para a chamada terceira geração, conhecida como UMTS (*Universal Mobile Telecommunication System*), uma vez que várias das soluções em desenvolvimento, nesse caso, apontam para a utilização de conectividade IP entre seus elementos, possibilitando o compartilhamento total da rede de transporte entre os elementos das duas gerações, assegurando melhor eficiência na utilização da banda disponível.

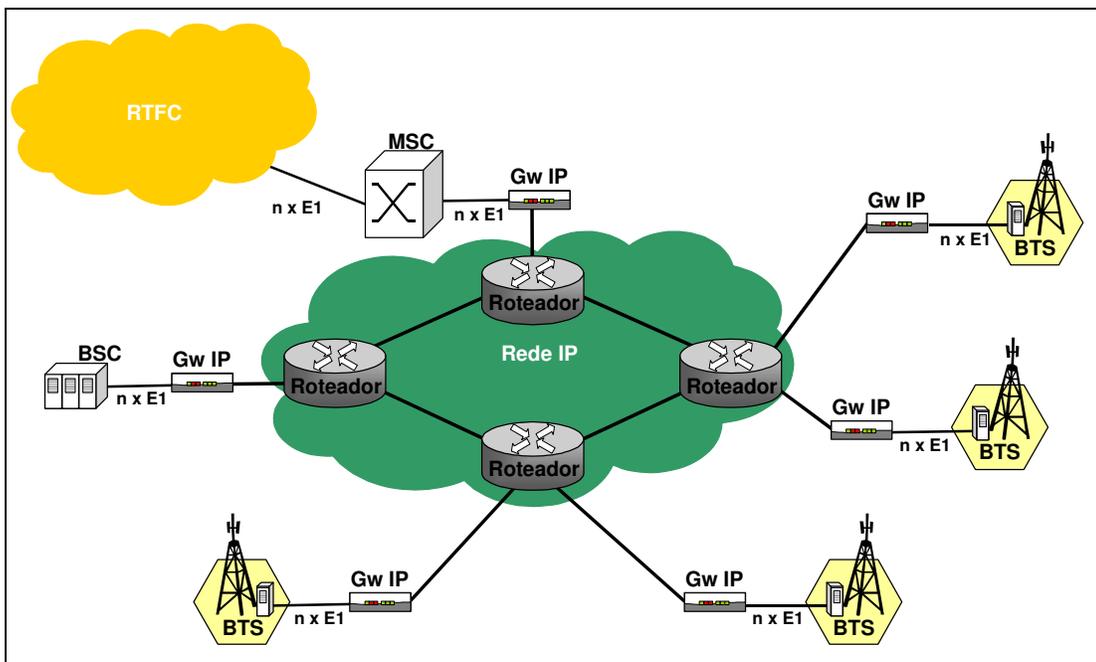


Figura 2.24 - *Backbone IP* para uma rede móvel celular.

Como conclusão da exposição realizada nesse capítulo, mostrou-se que os desafios enfrentados pelo transporte de TDM sobre PSN podem ser superados, utilizando dois novos mecanismos para aprimoramento das tecnologias existentes. Além disso, foi observado que a implementação de pseudo-circuitos é uma boa alternativa ao VoIP e uma realidade comercial para diversas aplicações. No capítulo 3 será feita a proposta de utilização desses pseudo-circuitos na transmissão de vídeo digital, criando aplicações similares a IPTV suportadas através dessa tecnologia, aprimorada pelos novos mecanismos apresentados.



### **3 - ALTERNATIVA TDMoIP® PARA *BROADCASTING* IPTV**

Nesse capítulo é discutida a questão da transmissão de vídeo digital, com uma rápida revisão dos principais padrões de codificação existentes e sua indicação para o transporte através dos pseudo-circuitos propostos. Na seqüência é apresentado o conceito usual de IPTV, com suas características e requisitos, além de uma breve descrição da tecnologia empregada pelas aplicações atuais. Finalmente, é feita a correlação entre os novos mecanismos propostos de seqüenciamento e recuperação do relógio no receptor e as adaptações necessárias para a utilização dos pseudo-circuitos em *broadcasting* de serviços IPTV similares.

#### **3.1 - A TRANSMISSÃO DE VÍDEO DIGITAL**

A utilização de sinais digitais para a transmissão de informações revolucionou as redes de telecomunicações a partir da metade do século XX, por oferecer solução para o principal problema enfrentado por qualquer sistema de comunicação a grandes distâncias: o ruído térmico presente no canal utilizado e/ou introduzido pelos componentes do sistema, que limitava o processo de regeneração dos sinais ao longo do enlace, já que o ruído, presente na mesma banda do sinal original, é amplificado em cada uma das etapas de repetição, degradando a relação sinal/ruído do canal a níveis inaceitáveis a partir de determinada distância. Com a digitalização dos sinais, o processo de regeneração foi reduzido

Contudo, apesar da forte digitalização observada nas redes de telefonia e em outras aplicações de áudio como o *broadcasting* de rádio, a transmissão digital de sinais de vídeo enfrentou grandes obstáculos, em função da enorme largura de banda necessária para a sua transmissão digital, permanecendo quase totalmente sob domínio analógico até meados da década de 1980. Como ilustração, a transmissão de um canal de TV analógico no Brasil ocupa uma faixa de 6MHz no espectro VHF, enquanto para a transmissão digital do mesmo canal, com qualidade similar e assumindo uma resolução de 640x480 pontos, 30

quadros por segundo e 16 bits para quantização e codificação, seriam necessários mais de 147Mbps de taxa de transmissão.

Assim, a efetiva utilização do vídeo digital somente tornou-se possível com a expansão da microeletrônica e o advento dos DSPs, que permitiram o desenvolvimento de algoritmos para a compressão e codificação mais eficiente dos sinais de vídeo, explorando suas propriedades e a redundância existente entre quadros sucessivos. Esses algoritmos foram sendo aperfeiçoados e deram origem aos padrões de codificação utilizados atualmente, que permitiram o desenvolvimento de aplicações como VCDs, DVDs, VoD, videotelefonia, videoconferência, TV por assinatura via satélite, *broadcasting* de TV Digital e IPTV.

### **3.2 - OS PADRÕES DE CODIFICAÇÃO DE VÍDEO**

A codificação eficiente dos sinais de vídeo apresenta dois grandes objetivos, a redução do tamanho ocupado pelas amostras codificadas e a otimização da banda utilizada em sua transmissão. Esses objetivos são alcançados pelas técnicas de compressão, que têm por base duas premissas básicas: a comprovação, através de estudos científicos, de que a visão humana é incapaz de perceber toda a informação presente em uma imagem complexa em movimento, permitindo que sejam desenvolvidos processos que descartem as informações imperceptíveis; e o fato de que a maioria das seqüências de imagens apresenta muitas regiões repetidas, onde não acontece movimento, que podem ser transmitidas com velocidade muito menor que a necessária para a imagem como um todo.

Com base nessas técnicas, desde início da década de 1990 foram desenvolvidos alguns padrões para a codificação digital de sinais de vídeo, sempre buscando uma melhor qualidade de transmissão com maior eficiência de utilização de banda, destacando-se as Recomendações ITU-T da série H.2XX e a família desenvolvida pelo grupo MPEG.

Assim, serão descritos os padrões mais utilizados nas aplicações de transmissão de vídeo, com o objetivo de mostrar as suas principais características e a sua possibilidade de utilização dentro da alternativa proposta para *broadcasting* IPTV.

### 3.2.1 – Os padrões H.261 e MPEG-1

O padrão de codificação de vídeo definido pela Recomendação ITU-T H.261, para utilização em sistemas audiovisuais de faixa estreita, foi desenvolvido em 1990 para a transmissão de sinais digitais de vídeo sobre linhas ISDN, então a grande promessa para a convergência das redes de comunicação, operando em taxas múltiplas de 64 kbps, definidas entre 40 kbps e 2Mbps e suportando quadros de vídeo com resoluções de 352x288, 176x144 ou 88x72 pixels, para 25 quadros por segundo.

Da mesma forma, o padrão MPEG-1 foi desenvolvido na mesma época com o objetivo de oferecer uma qualidade aceitável de vídeo utilizando taxas de até 1,5Mbps, suportando quadros com varredura progressiva (não entrelaçados) com resolução de 352x240 pixels, para 30 quadros por segundo (sistema NTSC), ou 352x288 pixels, para 25 quadros por segundo (sistemas PAL e SECAM); com qualidade de imagem comparável à oferecida pelas antigas fitas VHS (*Video Home System*), que difundiram o conteúdo de vídeo para uso doméstico na década de 1980. O MPEG-1 é considerado o formato de maior compatibilidade dentro da família MPEG, sendo reproduzido sem problemas pela grande maioria dos aplicativos disponíveis para computadores pessoais e aparelhos de VCD e DVD. É também um elemento desse padrão o *MPEG-1 Audio Layer 3*, CODEC para compressão de sinais de áudio popularmente conhecido como MP3, que modificou as estruturas da indústria fonográfica mundial a partir do final da década passada, permitindo de forma simples e eficiente o oferecimento e o intercâmbio de músicas através da Internet.

Considerando a emulação de um circuito E1 sobre uma PSN utilizando a tecnologia TDMoIP<sup>®</sup>, esses padrões de codificação, que operam a taxas inferiores a 2Mbps, podem ser utilizados para *broadcasting* de IPTV de forma transparente através desse pseudo-circuito.

### 3.2.2 – O padrão H.262 ou MPEG-2 Vídeo

Conforme as tecnologias de processamento digital de sinais se tornaram mais poderosas e baratas, dando condições para que equipamentos capazes de decodificar algoritmos mais complexos fossem acessíveis aos clientes residenciais, permitindo mais qualidade de vídeo, foram desenvolvidos padrões de codificação mais avançados. Em 1994 foi criado o MPEG-

2, mais tarde definido na Recomendação ITU-T H.262, como também é conhecido, cujo principal objetivo foi a adaptação do MPEG-1 para dar suporte à HDTV (*High Definition TeleVision*), com resolução típica de 1280x720 pixels. Esse padrão é tipicamente utilizado na codificação de sinais de áudio e vídeo para *broadcasting*, exigindo equipamentos mais poderosos, mas alcançando maior eficiência na codificação, ou seja, oferecendo melhor qualidade por kbps utilizado na transmissão dos sinais.

O MPEG-2 é o padrão utilizado pelos DVDs, assim como nas transmissões diretas via satélite (DTH) utilizando a banda *Ku* das TVs por assinatura, bem como nas transmissões digitais de TV a cabo. Esse padrão é dividido em três partes:

A parte “Sistemas” define dois formatos de encapsulamento, o chamado *Transport Stream*, projetado para transportar sinais de áudio e vídeo através de meios não confiáveis, como transmissão através do ar para TV digital; e o chamado *Program Stream*, projetado para meios confiáveis, como os discos utilizados nos padrões DVD e SVCD.

A parte “Vídeo” é semelhante ao MPEG-1, mas ao contrário deste também suporta quadros com varredura entrelaçada, como os utilizados nos sistemas de transmissão de TV analógica. O MPEG-2 não é, como o primeiro, otimizado para taxas abaixo de 1Mbps, mas supera a performance do MPEG-1 em taxas acima de 3Mbps, e seus decodificadores, quando totalmente compatíveis com o padrão, são capazes de reproduzir MPEG-1.

Finalmente, a parte “Áudio” é novamente uma evolução do MPEG-1, com codificação de sinais de áudio em mais de dois canais, mas compatível com os decodificadores MPEG-1 (MP3) através da reprodução dos dois canais principais em estéreo.

Considerando suas características e a qualidade superior oferecida, a emulação de um circuito E1 sobre uma rede de pacotes não seria suficiente para *broadcasting* de IPTV utilizando o padrão MPEG-2, e o desempenho apresentado pelos padrões anteriores em taxas inferiores a 2Mbps os credencia como a melhor escolha para a aplicação proposta.

### 3.2.3 – O padrão H.263

O padrão definido pela Recomendação ITU-T H.263 é um codificador de vídeo desenvolvido em 1995 como solução de compressão para baixas taxas de transmissão, em aplicações de videoconferência. Esse padrão foi desenvolvido como uma evolução do H.261, operando de forma otimizada em taxas inferiores às utilizadas pelo mesmo, considerando também diversos aspectos dos padrões MPEG-1 e MPEG-2, que o tornam um substituto adequado para o primeiro em todas as taxas utilizadas, sobretudo em virtude das suas evoluções H.263v2/H.263+, desenvolvida em 1998, e H.263v3/H.263++, de 2000. O padrão H.263 foi projetado para utilização em sistemas baseados na Recomendação ITU-T H.324, para videotelefonia e videoconferência sobre redes comutadas a circuitos, mas também é utilizado nas redes baseadas na Recomendação ITU-T H.323, para videoconferência em redes IP; e H.320, para videoconferência em redes ISDN. Além disso, a maioria do conteúdo em Flash Video presente em sítios Internet como o *YouTube*, *MySpace* e similares é apresentado com essa codificação.

Dessa forma, o H.263 apresenta-se como um excelente candidato para a aplicação proposta de *broadcasting* de IPTV através da emulação de circuitos E1 sobre uma rede de pacotes, utilizando a tecnologia TDMoIP<sup>®</sup>.

### 3.2.4 – O padrão H.264 ou MPEG-4 Vídeo

O padrão definido pela Recomendação ITU-T H.264, também conhecido como AVC (*Advanced Video Coding*), que corresponde à Parte 10 do padrão MPEG-4 teve como principal objetivo em seu desenvolvimento a capacidade de oferecer uma boa qualidade de vídeo utilizando taxas 50% inferiores às utilizadas pelos padrões anteriores, como o H.263, o MPEG-2 e o próprio MPEG-4 Parte 2, fazendo isso sem grandes incrementos de complexidade que tornassem a solução impraticável ou cara demais. Um segundo objetivo era realizar tal tarefa de modo flexível, permitindo que o padrão pudesse ser aplicado a uma grande variedade de aplicações, envolvendo taxas e resoluções de quadro tanto baixas como elevadas, e que operasse sobre diversas redes e sistemas, como videotelefonia comutada a circuitos, redes IP, transmissão de TV digital e armazenamento em DVDs.

Esse padrão tem como característica a elevada compressão de dados, sendo produto de um esforço conjunto entre os grupos VCEG (*Video Coding Experts Group*), patrocinado pelo ITU-T, e o MPEG (*Moving Picture Experts Group*), formando a parceria denominada JVT (*Joint Video Team*), que apresentou sua primeira versão em 2003.

O padrão MPEG-4 absorveu muitas características de seus antecessores MPEG-1 MPEG-2, incluindo novas funcionalidades como suporte a renderização tridimensional, objetos VRML (*Virtual Reality Modelling Language*), suporte para a gerência digital de direitos autorais e interatividade. Foram também desenvolvidas extensões para suporte à codificação de vídeo com fidelidade superior, denominadas FRExt (*Fidelity Range Extensions*), com maior precisão de amostragem, incluindo a codificação em 10 e 12 bits, e uma melhor resolução de cores, além de técnicas avançadas de processamento de sinais.

Em função da premissa de simplicidade e baixos custos assumida para a proposta de broadcasting IPTV discutida nesse trabalho, o padrão MPEG-4, embora possa ser utilizado em taxas compatíveis com a emulação de um circuito E1 sobre uma rede de pacotes, não é recomendável dada a baixa maturidade da tecnologia e ao elevado custo dos codificadores.

### **3.3 - SERVIÇOS IPTV: CARACTERÍSTICAS E REQUISITOS**

A chamada IPTV (*Internet Protocol TeleVision*) é um sistema de televisão digital por assinatura onde é utilizada uma rede de transporte IP, normalmente a Internet nas aplicações não corporativas, utilizando os acessos de banda larga dos clientes, normalmente baseados em tecnologias do tipo xDSL.

Para o Mercado Residencial, onde o entretenimento é o grande motivador para os usuários de segmento superior o objetivo das operadoras é prover aos clientes a chamada oferta tríplice, ou *triple play*, constituída pelos serviços de voz (telefonia fixa), dados (acesso Internet) e TV por assinatura (IPTV, VoD), sobre uma mesma conexão de acesso, com o objetivo de manter fidelizada a sua base de clientes. No caso das operadoras tradicionais brasileiras de telecomunicações, a oferta tríplice tem avançado da telefonia convencional,

através da rede de cobre, para o ADSL, oferecendo serviços Internet sobre essa rede, e agora chega à IPTV, uma vez que Telemar, Telefônica e Brasil Telecom já possuem testes-piloto em andamento com essa tecnologia, com previsão de lançamento comercial próximo, embora de forma cautelosa em virtude das questões em aberto no que tange à regulamentação. Na outra ponta, as operadoras de TV por assinatura também passaram a oferecer, nos últimos anos, serviços internet através de sua rede cabeada, e já estão consolidadas ofertas comerciais de telefonia fixa através de VoIP, como a oferecida pela Net/Embratel, qualificando-as, da mesma forma, ao oferecimento da oferta tríplice.

Além disso, está sendo adicionado agora mais um serviço a esse pacote, a telefonia móvel, caracterizando a chamada “oferta quádrupla”, buscando a convergência total dos serviços de telecomunicações, com benefícios tangíveis para clientes e operadoras, como um único canal de relacionamento, faturamento na mesma conta, promoções e benefícios e cruzados.

Para os Mercados Corporativos e Governo, que já são potencialmente clientes de serviços como a videoconferência e as chamadas TVs Executivas, a utilização de IPTV através dos serviços de conectividade IP já contratados para suas redes de dados, normalmente mais bem poderosos que as conexões ADSL residenciais, permite a criação, a custos bastante reduzidos, de novas ferramentas poderosas para o ambiente corporativo, como o treinamento à distância e auto-conduzido, eliminando o custo de viagens e permitindo a administração do tempo de cada participante; e a divulgação direta e instantânea de conteúdos de interesse ao negócio, como palestras, eventos, seminários e vídeos institucionais, no próprio computador pessoal utilizado diariamente pelos seus colaboradores, localizado em qualquer ponto de suas redes internas.

Como requisitos, as aplicações IPTV devem apresentar qualidade semelhante àquela oferecida pelos serviços tradicionais de TV por assinatura, como qualidade de transmissão e confiabilidade da programação, assegurando conforto aos clientes. A maioria dos sistemas IPTV implantados ainda não suportam padrões de alta definição, como o HDTV, mas somente outros sistemas de televisão digital como o DVB, que utiliza codificação MPEG-2 e oferece qualidade muito próxima aos DVDs domésticos.

Como todo o processo de transmissão é baseado no protocolo IP, as aplicações IPTV são bastante sensíveis às perdas de pacotes e aos atrasos sofridos dentro da rede, devendo haver disponibilidade de banda suficiente para os serviços desejados e se assegurada QoS suficiente para um fluxo estável de pacotes em cada um dos receptores, sendo esse o principal desafio das tecnologias desenvolvidas para essa finalidade.

### **3.4 - TECNOLOGIAS PARA TRANSMISSÃO IPTV E VoD**

Os serviços IPTV e VoD, que diferem entre si apenas com relação ao conteúdo e grau de interatividade oferecido, são tipicamente suportados por redes de transporte em modo pacotes de alta velocidade, como backbones IP corporativos, *Ethernet* a 100Mbps ou *GigabitEthernet*, às quais os clientes são conectados diretamente ou através de uma infraestrutura de acesso de banda larga, que pode ser compartilhada com a Internet pública.

Nessa condição, também denominada TV sobre Internet, o conteúdo é normalmente público, obtido através de qualquer servidor dentro da rede mundial, não sendo oferecida nenhuma garantia de qualidade, que fica dependente da disponibilidade de banda e do tráfego existente na rota entre o servidor e o cliente. Esse tipo de aplicação não pode ser considerado um efetivo concorrente à TV por assinatura, pois um usuário dificilmente pagaria por um serviço tão precário, estando mais próxima da simples e descompromissada visualização em HTTP, a partir de um computador pessoal, de vídeos curiosos e de curta duração, em sítios como o *YouTube*. Nesse mesmo grupo podem ser incluídos os cerca de 1.300 canais de TV sobre Internet gratuitos oferecidos em junho/2006 pelas grandes redes mundiais de televisão aberta, que retransmitem sua programação tradicional e podem ser acessados através de qualquer conexão Internet, utilizando computadores pessoais, TVs conectadas a computadores, Vídeo iPods ou telefones celulares 3G. Contudo, sua utilização é limitada em virtude da baixa qualidade de transporte oferecida pela Internet.

Nas aplicações comerciais IPTV, embora o acesso possa ser compartilhado com a Internet, como mostrado na Figura 3.1, a distribuição de conteúdo é feita através de redes privadas, da mesma forma que nas redes tradicionais de TV assinatura, com a diferença fundamental de que, ao invés de utilizar cabos ou receptores de sinais de satélite, onde todos os

conteúdos disponíveis são transmitidos de forma contínua para todos os assinantes, sendo selecionados através de multiplexação FDM nos receptores (ou *set-top boxes*); o fluxo de áudio/vídeo é entregue em pacotes IP, através de fluxos individuais estabelecidos diretamente entre o servidor e o receptor, transportando exclusivamente o conteúdo selecionado pelo cliente. Isso oferece duas vantagens: a redução de largura de banda, pois só é transmitida a programação que efetivamente deseja ser recebida por algum dos clientes conectados no momento, e a maior disponibilidade de conteúdos, pois não existe limitação na quantidade de canais, oferecendo uma liberdade de escolha nunca antes disponível nas TVs por assinatura, exatamente o que caracteriza o termo “Vídeo sob Demanda”.

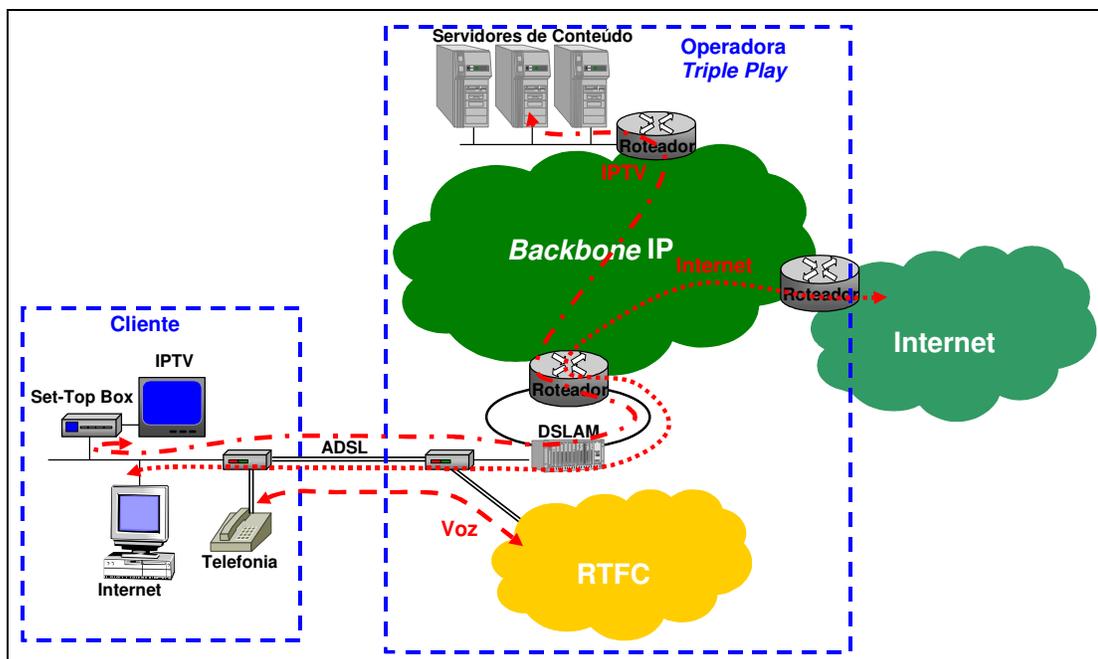


Figura 3.1 – Acesso IPTV numa configuração *triple play*.

Ao invés da restrição de conteúdo, a limitação dessa tecnologia é a capacidade da rede de transporte para estabelecer e gerenciar fluxos independentes, assegurando recursos suficientes para manter a qualidade de vídeo. Cada cliente estabelece a sua sessão, ao invés de “aderir” à recepção do conteúdo escolhido, permitindo que um programa seja iniciado exatamente no instante desejado e não em horário pré-determinado. Isso permite que sejam integradas ao serviço ferramentas de interatividade, como a escolha de programação entre milhares de opções disponíveis, a consulta a bancos de dados relacionados ao programa escolhido na tela do televisor ou a escolha de ângulos de câmera, recursos hoje disponíveis nos DVDs, mas com o inconveniente do deslocamento até a locadora.

Quando são abordadas as questões de escolha de filmes e interatividade, o conceito IPTV acaba confundido com o Vídeo sob Demanda (VoD), uma aplicação um pouco mais antiga, criada originalmente para redes comutadas a circuitos, como a ISDN, mas que também foi absorvida pela revolução do “Mundo IP” na última década. Nesse serviço, que ilustra bem uma das aplicações IPTV, o cliente navega em um catálogo de filmes apresentado em seu televisor, assiste a *trailers* e escolhe o filme a que deseja assistir. Quase instantaneamente, o filme é iniciado em sua tela, com qualidade DVD, pois o fluxo de pacotes contendo os dados de áudio e vídeo do mesmo passa a ser recebido pelo decodificador MPEG-2 localizado no *set-top box* do cliente, através de uma conexão IP ponto a ponto estabelecida entre este e o servidor da operadora onde está armazenado o filme. A fim de assegurar o mesmo conforto de um DVD, funções usuais como pausa, avanço e retrocesso de imagem são permitidas, utilizando o protocolo RTSP (*Real Time Streaming Protocol*).

A Figura 3.2 apresenta um serviço IPTV comercial sendo oferecido a uma grande corporação, com funcionários espalhados por diversos ambientes geograficamente distantes, através da rede IP (com QoS) contratada junto a uma operadora de dados, utilizando o modelo ponto a ponto (*unicast*) descrito acima.

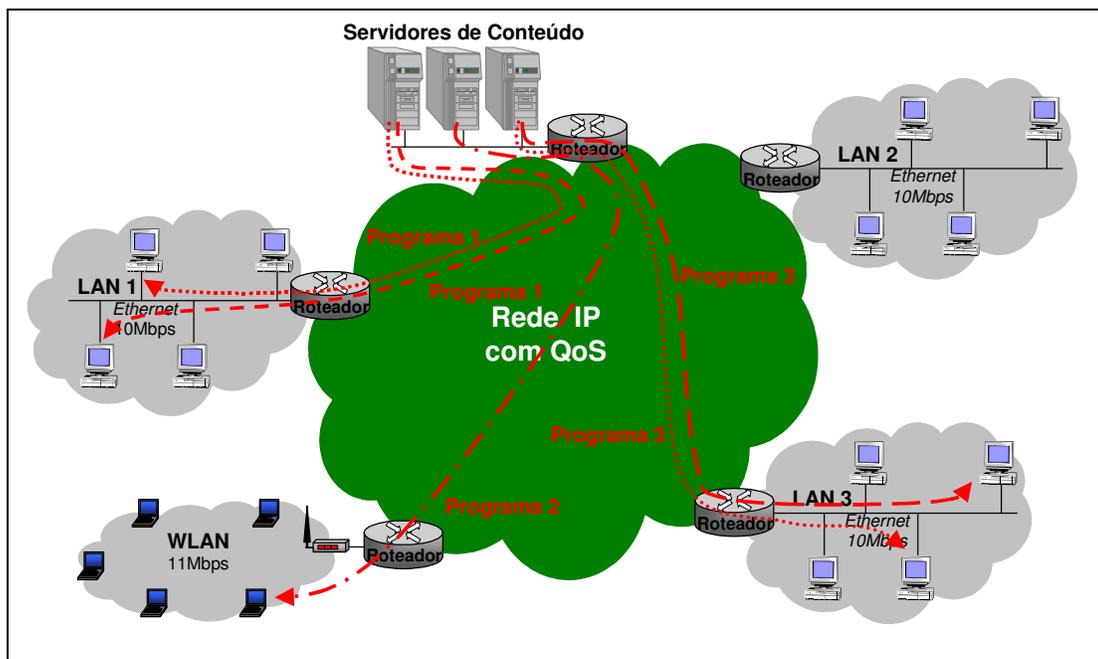


Figura 3.2 – Modelo convencional ponto a ponto para transmissão IPTV.

No exemplo, o programa 1 é o treinamento em vídeo para o pessoal da contabilidade, enquanto o programa 2, um filme de divulgação do novo produto, está sendo assistido por um consultor de vendas, em sua base remota. Ao mesmo tempo, está sendo assistida por dois colaboradores, no programa 3, a mensagem de fim de ano do presidente da empresa.

### **3.5 - ADEQUAÇÃO TDMOIP® PARA TRANSMISSÃO DE VÍDEO**

A tecnologia IPTV apresentada vem sendo desenvolvida por grandes empresas de tecnologia, como a gigante Microsoft®, com foco nas aplicações comerciais de entretenimento voltadas para clientes de alto poder aquisitivo. Assim, a utilização dos equipamentos, protocolos e sistemas de gestão disponíveis no mercado exigiria a justa remuneração desse desenvolvimento, através da aquisição de licenças de uso ou do pagamento direto de *royalties*. Dessa forma, a criação de uma rede nacional de IPTV para educação à distância, com objetivos puramente sociais, permitindo a transmissão ao vivo de eventos, palestras ou seminários, bem como a distribuição livre de conteúdos de vídeo associados a questões acadêmicas ou culturais, dependeria de pesados investimentos governamentais, que restringiriam a sua abrangência e capilaridade. Da mesma maneira, a contratação de uma rede de transporte com a QoS necessária para aplicações IPTV na forma apresentada tornaria o custo operacional do projeto, para uma abrangência nacional, quase proibitivo, como pode ser deduzido a partir da Tabela 1.1, onde a formação dos cinco circuitos propostos, ligando Brasília a São Paulo, Rio de Janeiro, Belo Horizonte, Cuiabá e Porto Alegre, utilizando o serviço Vetor para dados multimídia, o mais indicado nesse tipo de aplicação, teria um custo mensal estimado em quase R\$ 60 mil, apenas para a rede de transporte e considerando um único ponto de acesso em cada uma dessas cidades.

Assim, este trabalho propõe a utilização da emulação de pseudo-circuitos E1, utilizando as tecnologias TDMoIP® ou SAToP com os aprimoramentos propostos para o seqüenciamento de pacotes e a sincronização adaptativa dos receptores, como alternativa para a formação dessa rede nacional de tele-educação, utilizando a conectividade IP oferecida pelas redes de alta velocidade espalhadas pelas comunidades acadêmicas do país, como o Projeto REMAV (Redes Metropolitanas de Alta Velocidade), interconectadas através do *backbone* RNP2, da RNP (Rede Nacional de Pesquisa) ou, quando necessário,

dos *backbones* das operadoras nacionais, utilizando conectividade IP sem QoS assegurada, mas com garantia de banda, como o IP Dedicado da Brasil Telecom, muito mais baratos. Como referência, a conectividade nesse serviço para as mesmas cidades acima, considerando um único ponto de acesso em cada uma, teria um custo mensal estimado em menos de R\$ 12 mil, cinco vezes menos, para a mesma velocidade de 2Mbps.

Dessa forma, a tecnologia PWE3 proposta permitiria a emulação de circuitos E1 dedicados entre os pontos desejados para geração e recepção de conteúdo dentro da comunidade acadêmica interligada através dessa rede nacional, completamente transparentes aos serviços e protocolos nativos dos equipamentos de áudio e vídeo disponíveis, desde que utilizando taxas inferiores a 1,5Mbps, como os codificadores MPEG-1, H.261, e H.263, assegurando qualidade de vídeo comparável aos sistemas VHS e VCD.

A Figura 3.3 apresenta essa alternativa de *broadcasting* IPTV utilizando TDMoIP<sup>®</sup>, para diversas redes geograficamente distantes, ilustrando também a redução na ocupação de banda dentro do *backbone* pela utilização de roteadores com suporte a conexões ponto-multiponto (*multicast*). Os programas são gerados na origem, em codificação nativa, e encaminhados dentro dos pseudo-circuitos E1 de forma transparente até os roteadores de destino, considerando o endereço *multicast* de cada um dos grupos desejados. A partir de então, os pacotes são encaminhados a cada um dos destinos envolvidos na recepção do programa, ficando cada receptor encarregado de fazer o seqüenciamento e a sincronização adaptativa de seu próprio fluxo, como será discutido na seção 3.6.3. Esses receptores podem ser microcomputadores implementando a função *gateway* TDMoIP<sup>®</sup>, como indicado, que já fariam também a decodificação do vídeo sem nenhum *hardware* adicional, através de programas como o VLC (*VideoLan Client*) utilizado no ambiente descrito em [SCHULTER2002] e desenvolvido em código aberto por pesquisadores da École Centrale de Paris; ou ainda por um *gateway* TDMoIP<sup>®</sup> propriamente dito, entregando o sinal E1 regenerado para um CODEC e permitindo a sua exibição através de sistemas de vídeo analógico, como diversos televisores em um circuito interno de vídeo ou o transmissor VHF da TV educativa local, para transmissão em maior alcance, como *campi* remotos, postos avançados de extensão ou escolas municipais ou estaduais, incluindo a zona rural.

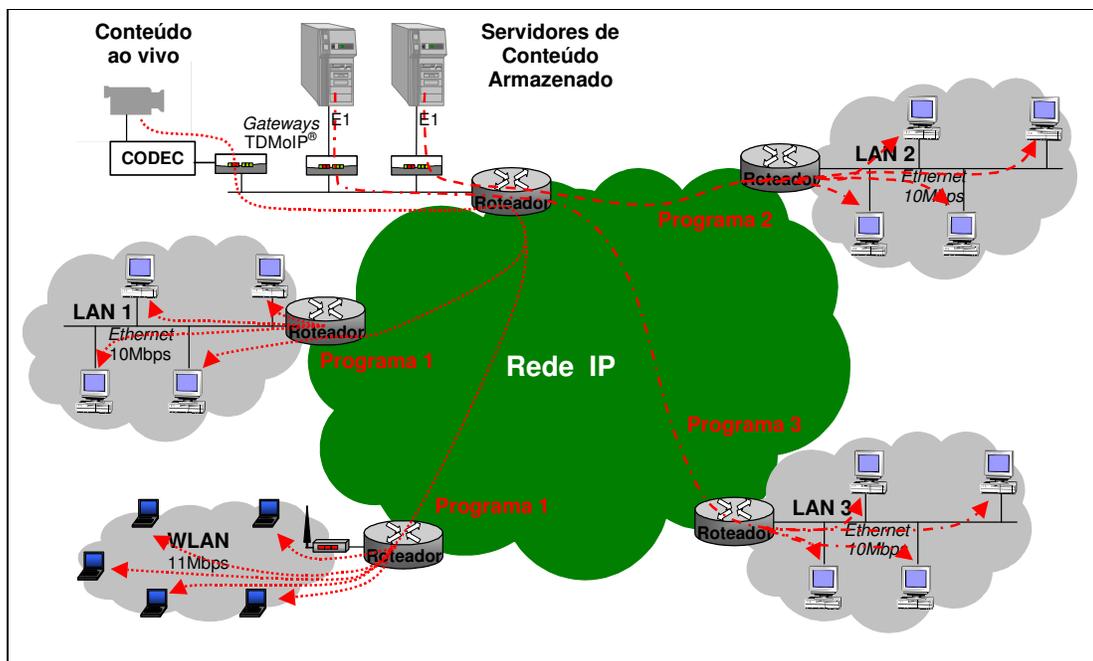


Figura 3.3 – Alternativa de *broadcasting* IPTV usando TDMoIP® e *multicast*.

Fazendo um paralelo com a aplicação IPTV corporativa descrita, os estudantes do laboratório de sistemas da universidade LAN1, assim como os alunos espalhados pelo *campus* da universidade WLAN estariam assistindo a uma palestra sobre redes de alta velocidade, transmitida ao vivo pela UnB, enquanto os professores de uma grande escola estadual vizinha LAN2 estariam assistindo à defesa de doutorado em educação de adultos de uma de suas colegas, gravada há duas semanas; e na universidade LAN3 estaria sendo exibida aos alunos uma cirurgia crítica gravada num hospital de Brasília.

Na proposta desse trabalho, todos os desafios associados à transmissão IPTV através da rede, que convencionalmente são abordados utilizando protocolos como o RTP e técnicas de mitigação de perdas dentro da camada de aplicação, são transferidos para o que seria, do ponto de vista de uma aplicação de vídeo, a “camada de enlace”, representada pelo pseudo-circuito emulado, exatamente como no paralelo apresentado anteriormente entre as abordagens VoIP e TDMoIP®. A emulação PWE3 proposta representa assim um “tubo” para a aplicação de transmissão e recepção de vídeo, exatamente como se os equipamentos de origem e destino estivessem conectados por um circuito dedicado, como acontece nas aplicações comerciais baseadas em ATM.

Contudo, para que a tecnologia PWE3 possa efetivamente ser utilizada, são necessárias algumas adaptações baseadas nos mecanismos propostos nas seções 2.3.2 e 2.3.3, a fim de assegurar maior robustez em relação ao atraso e perda dos pacotes transportados pela PSN.

### 3.5.1 - Fragmentação do fluxo de vídeo em pacotes

O primeiro aspecto a considerar na transmissão de sinais de vídeo digital codificado, que consiste num fluxo contínuo de bits, através de um pseudo-circuito emulado sobre uma rede de pacotes, onde os dados fluem de natureza discreta, é exatamente a fragmentação desse fluxo para montagem dos pacotes. Na transmissão de canais de voz, os pontos de fragmentação apresentam-se de forma mais intuitiva na abordagem SAToP, em função da fixação de estrutura, mas não são determinantes para a abordagem TDMoIP<sup>®</sup>, que tratam a separação dos canais por indicação de estrutura, mas ambas permitem a retirada, do fluxo dos pacotes, dos dados associados a canais ociosos. Em termos práticos, como nos sinais de vídeo toda a redundância já foi absorvida no processo de codificação, a fragmentação do fluxo contínuo em pacotes deve observar duas premissas básicas:

- a) O compromisso entre a latência fim a fim dentro do circuito, que cresce com o uso de pacotes maiores; e o *overhead* transmitido através da rede, que cresce com o uso de pacotes menores, conforme indicado nas Tabelas 3.1 e 3.2.
- b) A propagação de erros no caso de perda de pacotes, uma vez que os algoritmos de codificação de vídeo apresentam interdependência entre as amostras transmitidas, já que mesmo nas abordagens envolvendo a transmissão de um único quadro por pacote pode haver necessidade de fragmentação, quando o tamanho do quadro supera a MTU da rede. Essa questão é complexa e foge ao escopo do trabalho, cabendo apenas registrar que são aproveitadas algumas características de fracionamento existentes em cada tipo de codificador para otimizar o ponto de fragmentação dos pacotes.

Tabela 3.1 – *Overhead* na PSN x tamanho dos pacotes na tecnologia SAToP.

PDU's	Total Octetos	Octetos TDM	<i>Overhead</i>	Tempo (ms)	Pacotes/s	Taxa TDM (bps)	Taxa Real (bps)
1	288	256	11,11%	1,000	1.000,000	2.048.000	2.304.000
2	544	512	5,88%	2,000	500,000	2.048.000	2.176.000
3	800	768	4,00%	3,000	333,333	2.048.000	2.133.333
4	1056	1024	3,03%	4,000	250,000	2.048.000	2.112.000
5	1312	1280	2,44%	5,000	200,000	2.048.000	2.099.200

Tabela 3.2 – *Overhead* na PSN x tamanho dos pacotes na tecnologia TDMoIP®.

PDU's	Total Octetos	Octetos TDM	Overhead	Tempo (ms)	Pacotes/s	Taxa TDM (bps)	Taxa Real (bps)
1	80	47	41,25%	0,184	5.446,809	2.048.000	3.485.957
2	128	94	26,56%	0,367	2.723,404	2.048.000	2.788.766
3	176	141	19,89%	0,551	1.815,603	2.048.000	2.556.369
4	224	188	16,07%	0,734	1.361,702	2.048.000	2.440.170
5	272	235	13,60%	0,918	1.089,362	2.048.000	2.370.451
6	320	282	11,88%	1,102	907,801	2.048.000	2.323.972
7	368	329	10,60%	1,285	778,116	2.048.000	2.290.772
8	416	376	9,62%	1,469	680,851	2.048.000	2.265.872
9	464	423	8,84%	1,652	605,201	2.048.000	2.246.506
10	512	470	8,20%	1,836	544,681	2.048.000	2.231.013
11	560	517	7,68%	2,020	495,164	2.048.000	2.218.337
12	608	564	7,24%	2,203	453,901	2.048.000	2.207.773
13	656	611	6,86%	2,387	418,985	2.048.000	2.198.835
14	704	658	6,53%	2,570	389,058	2.048.000	2.191.173
15	752	705	6,25%	2,754	363,121	2.048.000	2.184.533
16	800	752	6,00%	2,938	340,426	2.048.000	2.178.723
17	848	799	5,78%	3,121	320,401	2.048.000	2.173.597
18	896	846	5,58%	3,305	302,600	2.048.000	2.169.040
19	944	893	5,40%	3,488	286,674	2.048.000	2.164.963
20	992	940	5,24%	3,672	272,340	2.048.000	2.161.294
21	1040	987	5,10%	3,855	259,372	2.048.000	2.157.974
22	1088	1034	4,96%	4,039	247,582	2.048.000	2.154.956
23	1136	1081	4,84%	4,223	236,818	2.048.000	2.152.200
24	1184	1128	4,73%	4,406	226,950	2.048.000	2.149.674
25	1232	1175	4,63%	4,590	217,872	2.048.000	2.147.350
26	1280	1222	4,53%	4,773	209,493	2.048.000	2.145.205
27	1328	1269	4,44%	4,957	201,734	2.048.000	2.143.218
28	1376	1316	4,36%	5,141	194,529	2.048.000	2.141.374
29	1424	1363	4,28%	5,324	187,821	2.048.000	2.139.657
30	1472	1410	4,21%	5,508	181,560	2.048.000	2.138.054

O tamanho dos pacotes a ser utilizado, definido pela quantidade de PDU's em cada pacote, depende das características da PSN, da distância entre as extremidades do pseudo-circuito e do codificador, devendo considerar como limite máximo a MTU da rede, normalmente 1500 octetos quando existe um segmento *ethernet*, e o *overhead* máximo admissível em função da banda disponível. Para a abordagem TDMoIP®, pacotes com menos de 5 PDU's, indicados em cinza na tabela 3.1, são considerados proibitivos pela baixa eficiência no uso de banda, além de necessitarem de elevadas taxas de processamento, em pacotes por segundo, nos roteadores, e não serem compatíveis com a banda proposta de 2Mbps, quando é utilizada a taxa de 1,5Mbps de dados TDM pelo codificador. Para a abordagem SAToP, todos os tamanhos indicados são possíveis.

### 3.5.2 - Absorção de PDV na transmissão IPTV

O segundo aspecto a considerar é a necessidade de absorção da variação no atraso sofrida pelos pacotes (PDV) que chegam ao receptor, através da utilização de um *jitter buffer*, como na maioria das aplicações em tempo real sobre redes de pacotes.

Entretanto, é exatamente nesse ponto que o novo mecanismo proposto nesse trabalho para seqüenciamento dos pacotes, apresentado na seção 2.3.2 oferece a grande contribuição para o aprimoramento da tecnologia PWE3 para o *broadcasting* de IPTV, pois permite que sejam dispensados os *timestamps*, reduzindo o *overhead*, e realiza o controle do reordenamento dos pacotes exclusivamente através do número de seqüência recebido, dentro do próprio processo de escrita no *buffer*, que é circular e também não necessita de qualquer tipo de gerenciamento; assegurando um excelente desempenho e permitindo a recuperação automática de qualquer pacote atrasado ou adiantado que seja recebido, respectivamente, até o instante anterior ao de reprodução do mesmo; e imediatamente após a reprodução da posição corrente.

Essa característica, bem superior àquela utilizada pela implementação RAD, que define a possibilidade de reordenação entre 0 e 7 pacotes devido ao ponteiro AAL1, é exatamente o que melhora a robustez do pseudo-circuito para utilização em redes PSN que assegurem a banda, mas não ofereçam garantia de entrega ou controle de atraso. Em função da aplicação de *broadcasting* IPTV proposta ser basicamente unidirecional, com interatividade básica exclusivamente para o controle da reprodução do vídeo e não contemplando funções de videotelefonia ou videoconferência, o *jitter buffer* pode ter tamanho elevado, não existindo a restrição de 150ms para a latência fim a fim estabelecida como requisito para o transporte de canais de voz TDM. Dessa forma, cada um dos receptores envolvidos pode se ajustar, dinamicamente, às características de atraso dos pacotes que são recebidos, assegurando a reprodução de um fluxo contínuo de dados para o codificador de vídeo

Além disso, como várias das técnicas propostas para mitigação das perdas de pacote em transmissão multimídia através de redes IP estão baseadas em solicitação de retransmissão [RFC2354], o mecanismo de seqüenciamento desenvolvido, que grava os pacotes dentro do *buffer* em função do número de seqüência recebido, permite um tratamento simples e

eficiente das retransmissões pelo receptor, pois pacotes recebidos a qualquer momento, desde que anteriores ao instante programado para a sua reprodução, são diretamente acomodados na seqüência correta. Assim, embora a retransmissão não seja uma situação desejada em PWE3, na aplicação de *broadcasting* IPTV pode ser criado um processo adicional, monitorando as falhas dentro do “buffer”, em intervalos compatíveis com o RTT do pseudo-circuito, e solicitando a retransmissão de um pacote perdido a tempo de que possa ser reproduzido normalmente, no instante programado, sem que seja necessária nenhuma modificação no processo de gravação ou leitura do *buffer* no receptor.

Finalmente, em função da total independência entre o processo de seqüenciamento dos pacotes, realizado durante a escrita no *buffer*, do processo de reprodução à taxa constante, as funções típicas de parada, avanço, atraso, exibição quadro a quadro e em velocidade mais lenta ou mais acelerada podem ser facilmente implementadas, envolvendo exclusivamente cada receptor, desde que exista capacidade de armazenamento suficiente no *buffer*, o que confere à aplicação um desempenho bem mais próximo da reprodução de vídeos em VHS ou DVD do que os usuais vídeos reproduzidos da Internet através do protocolo http, onde o acionamento de qualquer dessas funções prejudica o fluxo de vídeo, obrigando a que os pacotes sejam novamente requisitados do servidor e gerando atrasos desconfortáveis na continuidade da reprodução.

### **3.5.3 - Sincronização dos receptores na transmissão IPTV**

O esquema normalmente utilizado para controle da qualidade de vídeo em IPTV, incluindo a regeneração do relógio para controle do fluxo de bits entregue ao decodificador, é demonstrado na Figura 3.4, para uma transmissão unidirecional ponto a ponto (*unicast*). Nesse tipo de esquema de controle, para o qual é apresentada em [GENEL2003] uma proposta com o codificador H.263+, utilizando a Internet, existe uma interdependência muito grande entre o receptor e o transmissor, uma vez que a qualidade percebida no destino é explicitamente sinalizada para a origem, que por sua vez precisa adequar as condições de transmissão a fim de melhorar a qualidade no receptor em função das variações de desempenho da PSN no transporte dos pacotes.

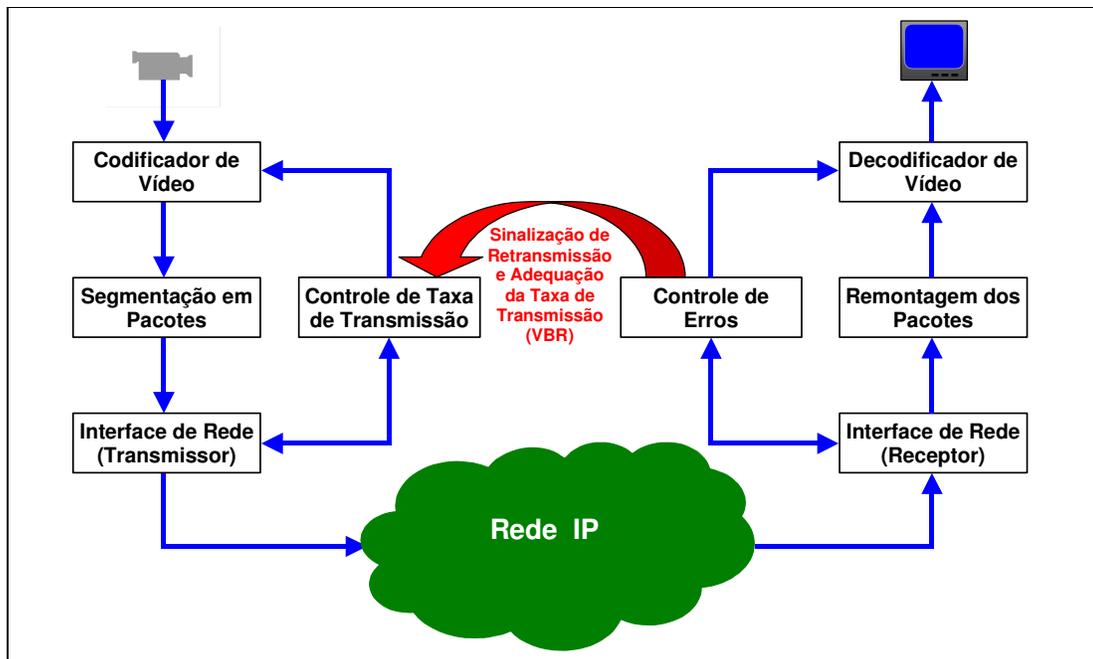


Figura 3.4 – Esquema usual para controle de qualidade em IPTV.

Como é observado na Figura 3.4, nesse esquema usual o codificador realiza a compressão dos sinais de vídeo e os entrega para o processo de segmentação, onde o fluxo é acomodado em pacotes e marcações de tempo (*timestamps*) são agregadas aos mesmos através do protocolo RTP, para então serem encaminhados através da rede IP. O receptor, recebendo os pacotes, extrai os dados e marcações de tempo e efetua a regeneração do fluxo contínuo de bits do sinal de vídeo codificado, que é entregue ao decodificador. Enquanto isso, pacotes de controle são trocados entre o transmissor e o receptor através do protocolo RTCP, em portas IP separadas, contendo informações e estatísticas do enlace coletadas no destino para realimentação do transmissor, incluindo a perda de pacotes, a fim de que o mesmo possa adequar sua operação, seja retransmitindo o pacote perdido, seja reduzindo a taxa de transmissão, no caso de codificadores operando em VBR, ou ainda atuando diretamente no codificador para modificar a forma de tratamento dos quadros, tornando a codificação mais robusta a perdas.

Na alternativa de *broadcasting* IPTV proposta, essa interação direta entre receptor e transmissor para controle do fluxo de pacotes e do processo de codificação, que se apresenta como grande complicador em aplicações ponto-multiponto, não é necessária, pois existe um circuito E1 dedicado entre transmissor e receptor; ou entre este e cada um dos receptores, se o fluxo de pacotes for replicado pelos roteadores ao longo da rota em

função da utilização de protocolos *multicast*. Assim, os equipamentos podem ser configurados para processos de codificação/decodificação mais simples, pressupondo a existência de meios de transmissão confiáveis, como no caso do *Program Stream* ao invés do *Transport Stream* para o MPEG2.

Essa alternativa é apresentada na Figura 3.5, onde toda a atividade de controle é executada exclusivamente no receptor, através do mecanismo de sincronização adaptativa proposto, cujo principal objetivo é assegurar a estabilidade do fluxo de bits dentro do pseudo-circuito, e não a qualidade da decodificação em si. Dessa forma, toda a responsabilidade pela qualidade da conexão, do pseudo-circuito, que seria uma “camada de enlace” para a aplicação, sendo a codificação/decodificação realizada com taxas constantes e de forma completamente independente. Mecanismos de retransmissão de pacotes, se existentes, são também tratados ao nível do pseudo-circuito, de forma transparente para o decodificador.

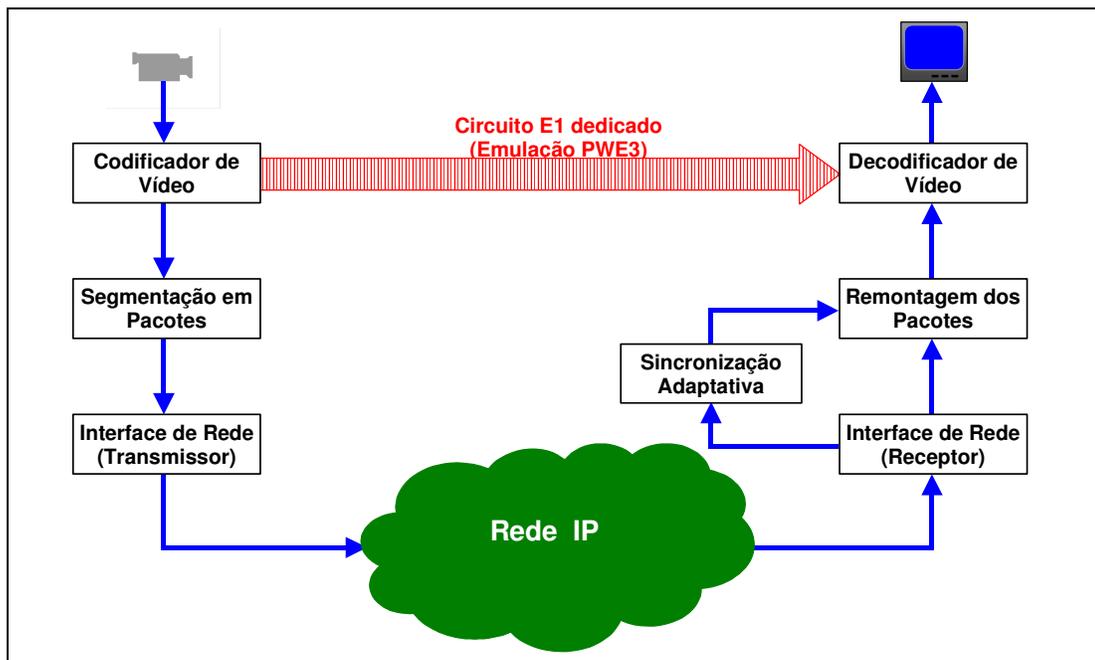


Figura 3.5 – Proposta de controle de qualidade baseado exclusivamente no receptor.

Como mostrado na Figura 3.5, o transmissor opera de forma livre, gerando um fluxo contínuo de pacotes contendo o sinal de vídeo codificado, cabendo ao receptor a função de regenerar esse sinal para o decodificador, baseado exclusivamente na informação de número de seqüência dos pacotes recebidos, através do novo mecanismo de

seqüenciamento desenvolvido, já discutido acima, e da recuperação do relógio do transmissor através do mecanismo de sincronização adaptativa baseado em PLL apresentado na seção 2.3.3.

Do ponto de vista da aplicação, o circuito E1 emulado pode ser interpretado como um simples cabo de conexão entre o codificador e o decodificador, não importando se ambos os equipamentos encontram-se no mesmo prédio ou a centenas de quilômetros de distância.

Felizmente, em função do caráter unidirecional da transmissão, os requisitos de precisão para recuperação do relógio, assegurando boa qualidade na reprodução do sinal de vídeo para o decodificador, são bastante inferiores aos necessários nas aplicações TDM para transmissão de voz definidos pela Recomendação ITU-T G.823, o que simplifica bastante essa tarefa para as taxas até 1,5Mbps definidas no escopo desse trabalho.

Em função das características definidas para o *broadcasting* IPTV proposto, com objetivos educacionais ou sociais, não existiriam grandes problemas se a recepção de um programa iniciasse alguns segundos mais tarde para alguns receptores, em decorrência de um tempo de aquisição mais elevado do PLL ou da necessidade de um pré-armazenamento maior de pacotes no *buffer*, desde que, a partir do início da reprodução, a qualidade de vídeo se mantivesse adequada, sem cortes ou interrupções.

Cabe observar ainda que, mesmo em aplicações comerciais domésticas, sobretudo quando não é feita a conexão aos televisores através de entradas “vídeo componente”, não existe grande criticidade, do ponto de vista do decodificador de vídeo, para a recuperação do relógio no receptor, sendo comum a utilização de osciladores locais nos *set-top boxes* para controlar o processo de reprodução analógica dos sinais de vídeo.

Embora não baseada em PWE3, uma abordagem similar à proposta, considerando o uso de PLL para recuperação do relógio no receptor através de *timestamps*, é apresentada em [TRYFONAS1999] para sistemas de transmissão MPEG-2, demonstrando excelente desempenho em aplicações onde existem requisitos críticos para a recuperação do relógio.

Como conclusão das discussões realizadas nesse capítulo, foi observado que existem padrões de codificação disponíveis e bastante difundidos para atendimento à restrição de taxa de transmissão, em 1,5Mbps, como o MPEG-1 e o grupo H.323, sendo possível oferecer qualidade similar à TV analógica existente com os pseudo-circuitos E1 propostos. Além disso, pôde ser caracterizada a transferência da responsabilidade, pela qualidade do vídeo, da camada de aplicação para a “camada de enlace” representada pelo pseudo-circuito, desacoplando o processo do transmissor e permitindo o controle dessa qualidade em cada receptor, pela utilização dos novos mecanismos de melhoria propostos no trabalho. No capítulo 4 será feita a validação desses mecanismos, através da sua implementação utilizando simulação e uma aplicação simples desenvolvida em Java, além da experimentação de um pseudo-circuito utilizando equipamentos comerciais, dentro da rede LabCom/UnB.



## **4 - IMPLEMENTAÇÃO E RESULTADOS EXPERIMENTAIS**

Nesse capítulo é realizada a implantação dos mecanismos propostos, inicialmente através de simulação, com a descrição detalhada da modelagem utilizada para recuperação do relógio no transmissor, em especial o PLL. Em seguida, são apresentados os resultados experimentais da utilização de um produto comercial para estabelecimento de um pseudo-circuito para entroncamento de voz em uma central telefônica. Na seqüência, é apresentada a tentativa de implementação de uma aplicação em Java para a emulação de circuitos TDM sobre redes IP, utilizando os mecanismos propostos, comparando seu desempenho com o produto comercial através de experimentação dentro da rede LabCom/UnB. Finalmente, são analisados os resultados obtidos na simulação e nos experimentos realizados.

### **4.1 - MODELAGEM TDMoIP<sup>®</sup> UTILIZANDO MATLAB E SIMULINK**

Com o objetivo de validar e avaliar o desempenho do novo mecanismo de seqüenciamento de pacotes, proposto na seção 2.3.2, e do mecanismo alternativo de sincronização TDM, proposto na seção 2.3.3, foram realizadas suas implementações em ambiente de simulação, através das ferramentas MatLab<sup>®</sup> e Simulink, na versão 6.0.0.88 Release 12.

#### **4.1.1 – Modelagem TDMoIP<sup>®</sup> no MatLab<sup>®</sup> e Simulink**

Em função das características distintas dos mecanismos propostos, foi analisada a melhor alternativa para sua simulação dentro das ferramentas de simulação disponíveis.

Para o novo mecanismo de seqüenciamento, por tratar-se de um algoritmo passível de implementação matemática, foram criadas funções dentro do MatLab<sup>®</sup> que executassem o tratamento desejado dos dados, sendo os principais desafios a modelagem dos pacotes como variáveis matemáticas e, fundamentalmente, a modelagem do comportamento de

uma rede PSN, refletindo o atraso sofrido pelos pacotes e, por conseqüência, sua perda e/ou recebimento fora de ordem.

Para o mecanismo alternativo de sincronização, contudo, embora a operação PLL também seja baseada em fundamentação matemática, a complexidade de modelagem através de simples funções de manipulação dos dados seria demasiado grande, e não permitiria uma correta visualização das questões envolvidas. Assim, optou-se pela utilização da ferramenta Simulink, inclusa no pacote MatLab<sup>®</sup>, que possui modelos disponíveis de todos os blocos básicos necessários para a construção dos circuitos que implementam o mecanismo proposto, caracterizando assim uma simulação do *hardware* associado ao mesmo.

#### **4.1.2 – Simulação MatLab<sup>®</sup> do algoritmo de seqüenciamento de pacotes**

A análise do problema de variação do atraso e perda de pacotes em uma rede PSN, realizada para a Internet em [BOLOT1993], apresenta o modelo da rede, do ponto de vista de um pacote transmitido entre dois pontos, como uma fila onde o tempo de serviço é uma variável aleatória que segue uma determinada estatística, característica da PSN analisada.

Esse modelo foi entendido como satisfatório para a validação do algoritmo proposto para seqüenciamento de pacotes, de modo que a modelagem matemática, dentro do MatLab<sup>®</sup>, foi totalmente baseada nessa abordagem. Dessa forma, cada pacote gerado e transmitido através da rede foi modelado como a linha de uma matriz, ao qual foram associados três atributos, armazenados em cada uma das colunas da mesma:

- O número de seqüência (NS), cujo valor inicial é determinado randomicamente dentro do intervalo  $[0; 2^{16}-1]$  e incrementado a cada novo pacote (nova linha da matriz);
- O instante de transmissão do pacote (t), cujo valor inicial é zero para o primeiro pacote, somando-se aos demais a duração dos dados TDM dentro de cada pacote (1ms);
- O instante de recepção do pacote, correspondente à soma do instante de transmissão (t) com um valor aleatório, com média equivalente à metade do RTT (*Round-Trip Time*) estabelecido para a simulação e distribuição estatística definida.

A Tabela 4.1 ilustra matriz utilizada na modelagem da transmissão dos pacotes na PSN.

Tabela 4.1 – Matriz de simulação da transmissão de pacotes através da PSN.

<b>Pacote Simulado</b>	<b>Número de Seqüência</b>	<b>Instante de Transmissão</b>	<b>Instante de Recepção</b>
<b>1</b>	16327	0 ms	20 ms
<b>2</b>	16328	1 ms	26 ms
<b>3</b>	16329	2 ms	39 ms
<b>4</b>	16330	3 ms	24 ms
<b>5</b>	16331	4 ms	23 ms
<b>6</b>	16332	5 ms	Infinito (perdido)
<b>7</b>	16333	6 ms	38 ms
<b>8</b>	16334	7 ms	29 ms
<b>9</b>	16335	8 ms	Infinito (perdido)
<b>10</b>	16336	9 ms	37 ms
...			
<b>500</b>	16827	499 ms	528 ms

Com o objetivo de permitir um controle mais efetivo do comportamento dos algoritmos de seqüenciamento na ocorrência de perdas de pacote dentro da simulação, foi incluída ainda uma outra variável, independente da estatística de distribuição definida para a PSN, com o objetivo de levar o atraso ao infinito, simulando um pacote perdido, para determinado percentual de pacotes previamente definido.

Dessa forma, para que a simulação fosse realizada, comparando o desempenho entre o algoritmo apresentado no *Internet-Draft* TDMoIP<sup>®</sup> (reproduzido no Apêndice A) e o proposto na seção 2.3.2, era preciso definir, nesse modelo, um comportamento estatístico para o atraso dos pacotes condizente com aquele observado nas redes de interesse.

Assim, foram definidas duas estratégias para a obtenção de amostras do atraso sofrido pelos pacotes em uma WAN de abrangência nacional, que permitisse a caracterização das situações a serem enfrentadas pelo problema de *broadcasting* PTV:

- 1) A medição do RTT para pacotes gerados na rede LabCom/UnB e encaminhados através da Internet para destinos nas principais capitais do país, utilizando o protocolo ICMP (*Internet Control Message Protocol*) com pacotes de 300 octetos (representando 1ms de tráfego TDM), em diversos horários e dias da semana ao longo do mês de agosto/2006, a fim de refletir o comportamento em redes sem controles ou garantias, baseadas apenas no serviço de melhor esforço, ou BE (*Best-Effort*);

- 2) A medição dos atrasos sofridos pelo tráfego real dos clientes, em diversos nós do *backbone* IP da Brasil Telecom e para as diversas classes do serviço Vetor, a fim de refletir o comportamento de uma rede bem controlada e com garantia de QoS aos clientes, além do uso sistemático de técnicas de engenharia de tráfego.

Infelizmente, a segunda estratégia não pôde ser executada, em função de mudanças na configuração da gerência da rede IP da Brasil Telecom, que impossibilitaram as medições, de forma que a estatística acabou sendo gerada exclusivamente com base nos RTTs medidos através da Internet, utilizando como endereço de destino os sítios das principais universidades federais do país e os portais dos governos estaduais. Evidentemente, essa abordagem foi impactada por outras variáveis além do atraso oferecido pela rede, como o desempenho dos servidores de domínio, a velocidade dos enlaces no destino e a carga dos servidores de hospedagem desses sítios, mas a quantidade de medições foi razoável e o comportamento estatístico obtido, apresentado na Figura 4.1, ficou dentro das expectativas previstas, com um RTT médio geral de 84,1 ms e 30% de perda de pacotes para todo o conjunto de medições realizadas.

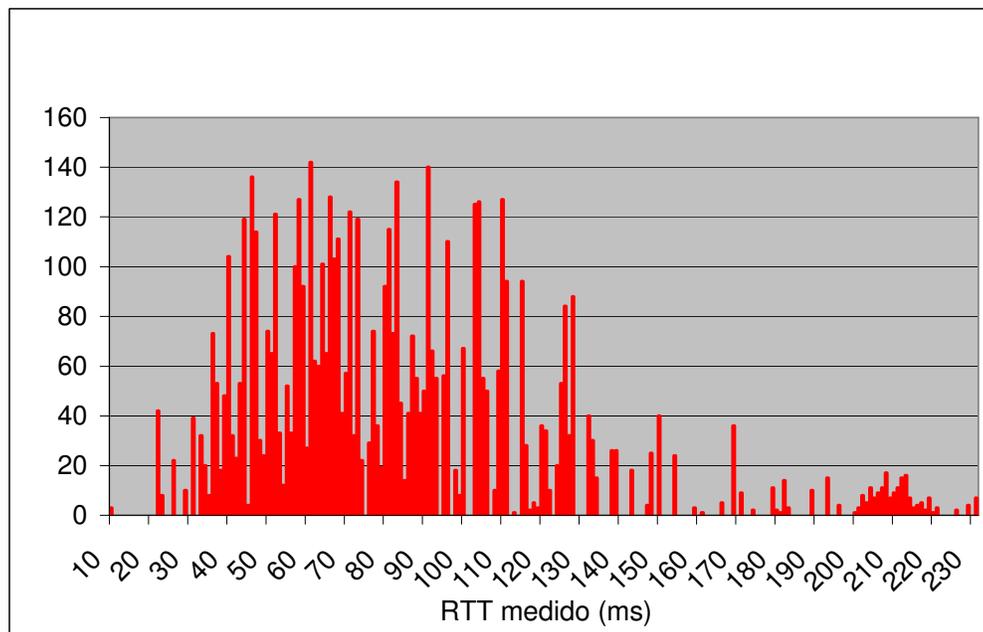


Figura 4.1 – Histograma de frequências na medição do RTT entre o LabCom/UnB e os sítios Internet das universidades federais e portais dos governos estaduais brasileiros.

A inspeção visual do histograma obtido para os RTTs medidos permite ajustar o comportamento do atraso a uma distribuição lognormal, em função da “cauda” representada pelos RTTs acima de 200ms, medidos para sítios na região norte do país, da UFAC (Universidade Federal do Acre) e UNIR (Fundação Universidade de Rondônia).

Contudo, considerando pouco representativa a média encontrada para o RTT, na faixa dos 80ms, e a concentração dos valores medidos no intervalo [40ms; 120ms], optou-se por considerar distribuição uniforme dentro desse intervalo na simulação, entendendo que isso representa bem o comportamento do atraso de pacotes para comparação entre os algoritmos de seqüenciamento, além de ser mais facilmente implementado no MatLab®.

Assim, o atributo “Instante de Recepção” foi modelado como uma variável aleatória de distribuição uniforme entre os valores  $RTT/2$  e  $RTT/2+RTT$ , para um RTT de 80 ms, sendo os resultados da simulação apresentados na Figura 4.2, onde os pontos em azul representam o número de seqüência de cada pacote transmitido, no instante de transmissão; enquanto os pontos em verde representam o mesmo número de seqüência, no respectivo instante de recebimento, função do atraso aleatório sofrido por cada pacote dentro da rede.

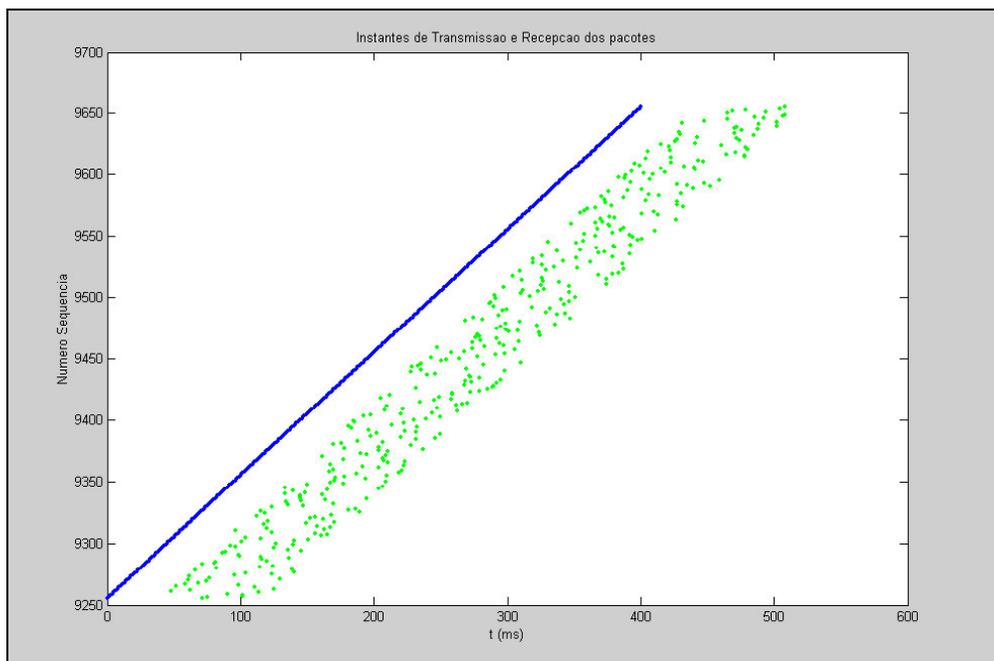


Figura 4.2 – Gráfico da simulação MatLab®, apresentando os pacotes transmitidos (azul) e recebidos (verde).

Nesse gráfico, o atraso aleatório sofrido por cada um dos pacotes é indicado pela distância, no eixo horizontal, entre o ponto em azul e o ponto em verde.

Na Figura 4.3 é mostrado o instante de recebimento de cada pacote, por ordem de chegada, apresentando uma visão mais intuitiva do processo. Pode ser observado que os primeiros pacotes são recebidos na ordem correta, formando uma pequena reta de inclinação positiva, a partir do pacote de número de seqüência 9.265 começam a haver degraus, indicando mudança na ordem de natural de recebimento, e a partir do pacote 9.275, a inclinação da reta torna-se bruscamente negativa, indicando que uma grande quantidade de pacotes com número de seqüência inferior a este foi recebido. Esse fenômeno se repete inúmeras vezes para os 400 pacotes da simulação, cuja taxa de pacotes perdidos foi de 10%, em função da elevada dispersão do atraso gerada pela distribuição uniforme utilizada.

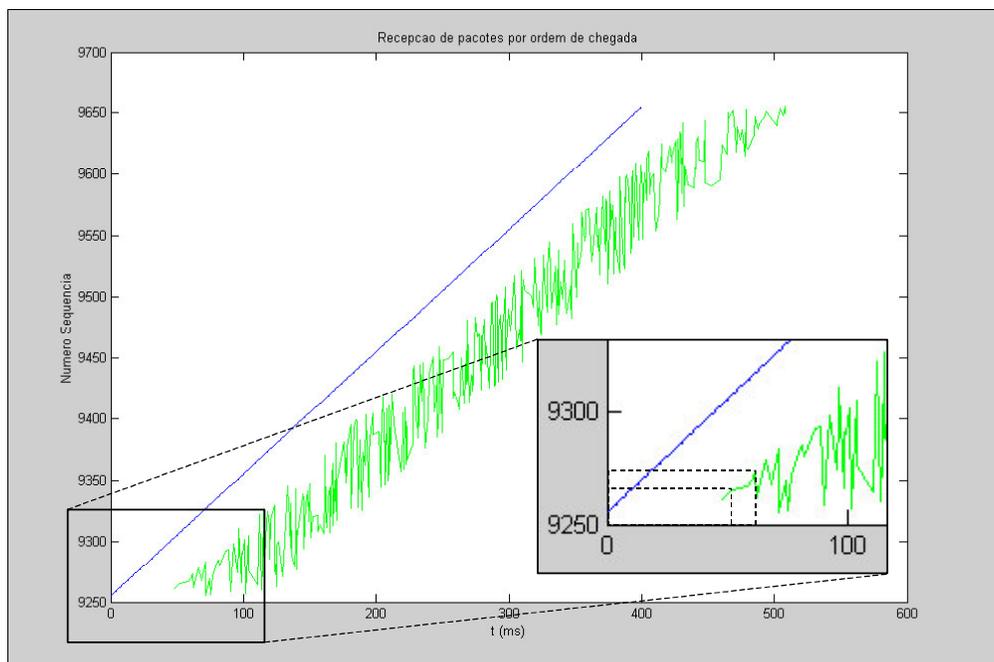


Figura 4.3 – Gráfico da simulação MatLab<sup>®</sup>, apresentando os instantes de transmissão (azul) e recepção (verde) de cada pacote.

A partir da situação mostrada nos gráficos anteriores, representada na simulação pela matriz descrita anteriormente, são aplicados os dois algoritmos de simulação em análise, com o objetivo de restabelecer o seqüenciamento dos pacotes, com a menor perda de informação possível.

Assim, na Figura 4.4 é apresentado o resultado da aplicação do algoritmo apresentado em [IETF2006], reproduzido no Apêndice D, utilizando como mitigação para os pacotes considerados perdidos a reprodução do último pacote recebido, considerando o parâmetro  $R=40$ , ou seja, aceitando pacotes atrasados com desvio de até quarenta posições em relação à esperada, conforme definido pelo algoritmo. Esse valor representa o máximo possível para a simulação realizada, onde a reprodução dos pacotes começa 40 ms após o recebimento do primeiro. Os pontos em vermelho representam o número de seqüência de cada pacote transmitido, no respectivo instante de reprodução para o fluxo TDM de saída.

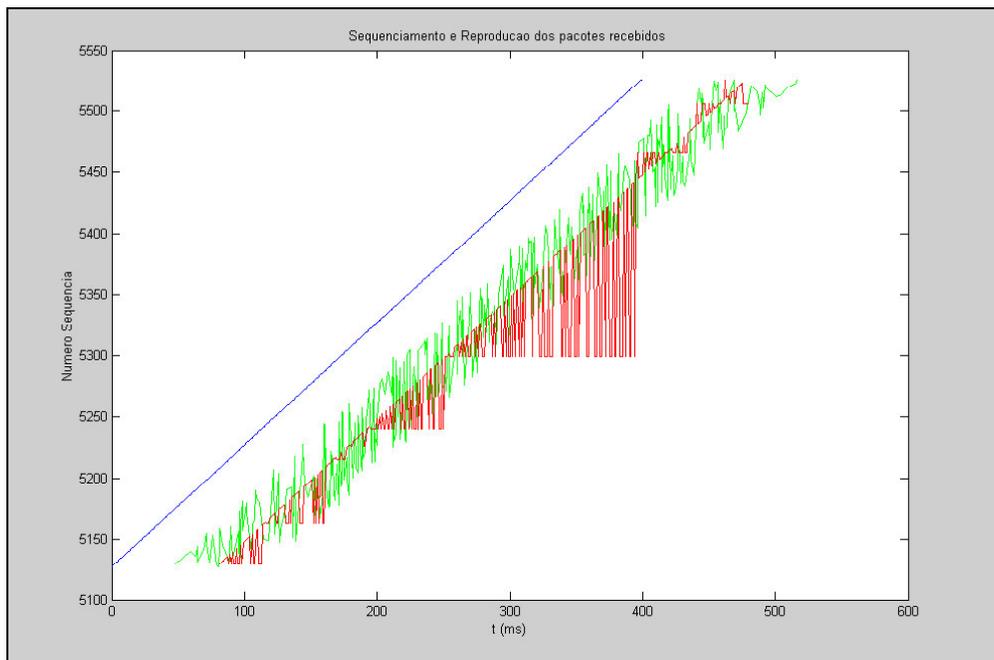


Figura 4.4 – Gráfico da simulação MatLab<sup>®</sup>, utilizando o algoritmo proposto pelo IETF para sequenciamento, apresentado no Apêndice A.

Como pode ser observado pelos pontos em vermelho, o desempenho desse algoritmo é muito ruim para o comportamento fortemente aleatório do atraso assumido, mesmo admitindo-se a recuperação de pacotes atrasados em até 40ms. A existência dos “patamares” mostrados na figura é decorrente da estratégia de preenchimento integral das posições remanescentes do *buffer*, utilizando o conteúdo do último pacote válido, sempre que for recebido um pacote de número de seqüência superior ao esperado pelo algoritmo. Essas posições preenchidas, em princípio, teriam seu conteúdo substituído pelos dados reais à medida que os pacotes atrasados fossem chegando, limitados ao instante de reprodução, mas isso, em função da elevada dispersão do atraso no modelo de rede

simulado, não ocorre com a frequência necessária para que a linha vermelha reproduzisse fielmente o seqüenciamento original dos pacotes, apresentado na linha em azul.

Finalmente, na Figura 4.5 é apresentado o resultado da aplicação do algoritmo proposto nesse trabalho, utilizando como regra de mitigação, para os pacotes considerados perdidos, também a reprodução do último pacote recebido sem problemas. Os pontos em vermelho novamente representam o número de seqüência de cada pacote transmitido, no respectivo instante de reprodução para o fluxo de saída.

Como pode ser observado pela reta em vermelho, o algoritmo absorveu completamente os pacotes recebidos fora da ordem esperada, em verde, recolocando-os com sucesso na ordem original em que foram transmitidos, indicada pela reta em azul. O deslocamento de aproximadamente 80 ms na horizontal entre as retas vermelha e azul é decorrente dos 40ms de atraso médio definido na simulação da rede, mais o atraso introduzido pelo *buffer* de reprodução, configurado para um tamanho de 80ms e início da reprodução dos pacotes, contendo 1ms de dados TDM, a partir de 50% de ocupação.

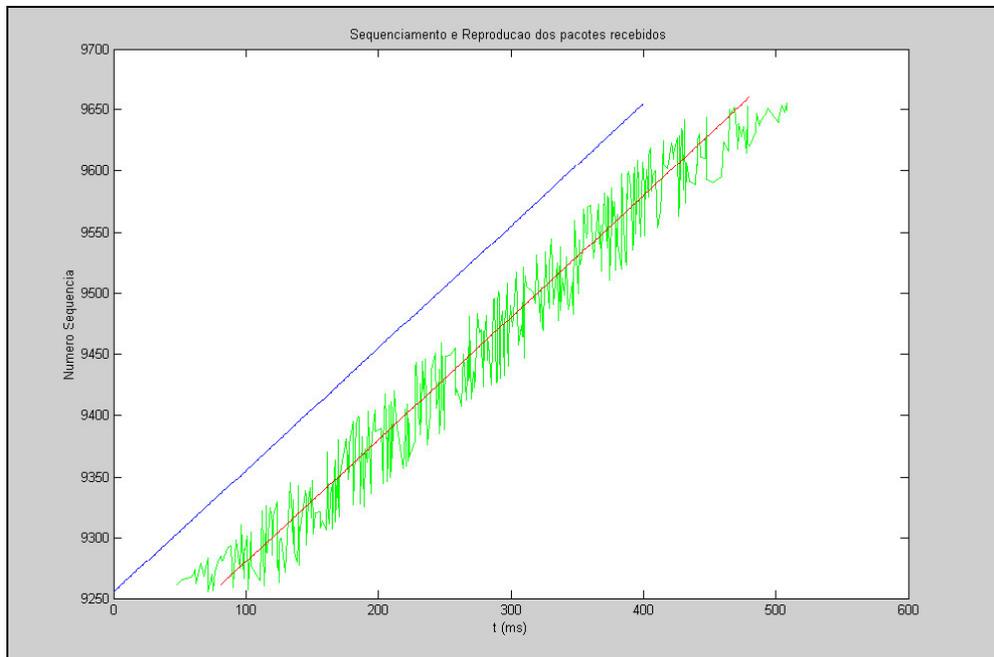


Figura 4.5 – Gráfico da simulação MatLab<sup>®</sup>, utilizando o novo mecanismo proposto de gravação no *jitter buffer* para seqüenciamento.

Nesse exemplo mostrado em particular, foram considerados perdidos e mitigados 42 dos 400 pacotes simulados, em função do seu recebimento não ter ocorrido a tempo de serem reproduzidos no fluxo de saída.

As funções MatLab<sup>®</sup> desenvolvidas para a implementação dos dois algoritmos apresentados acima estão listadas no Apêndice B.

#### 4.1.3 – Simulação Simulink do PLL para sincronização do receptor

O princípio geral de operação do PLL para sincronização adaptativa do receptor na emulação de circuitos TDM sobre uma rede de pacotes, conforme proposto na seção 2.3.3, é ilustrado pela Figura 4.6, adaptada de [AWEYA2004].

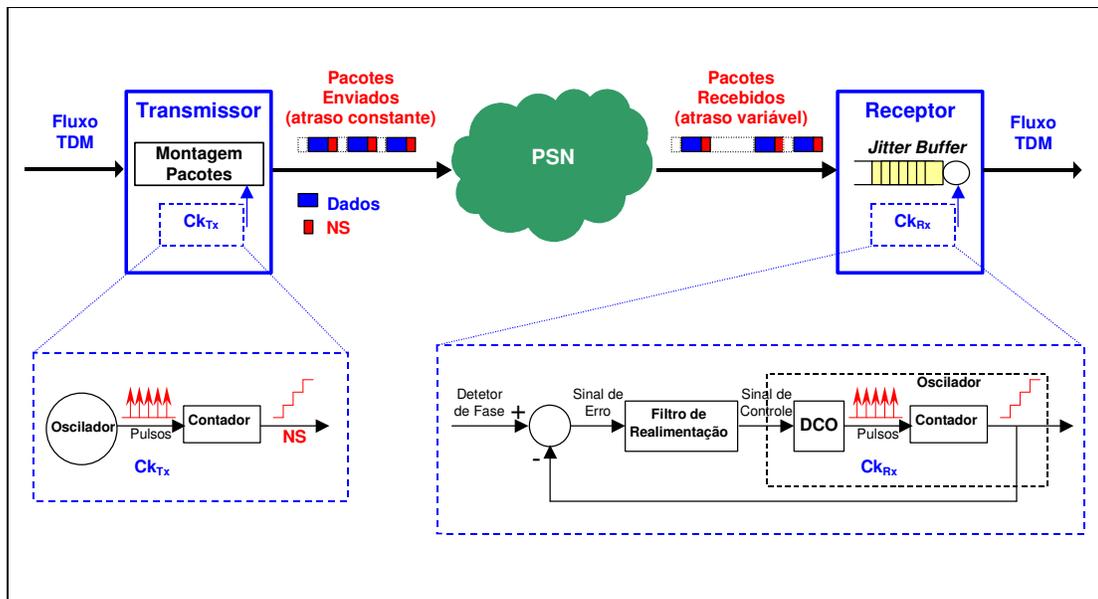


Figura 4.6 – Princípio de operação da sincronização adaptativa usando PLL.

O transmissor realiza a montagem dos pacotes, a uma taxa constante determinada por seu oscilador local, normalmente sincronizado com o relógio regenerado a partir do fluxo TDM original, que por sua vez estaria sincronizado com o restante da rede TDM. A cada um desses pacotes, cujo tamanho em duração dos dados TDM é fixo, é adicionado um número de seqüência representativo do mesmo, que na figura é gerado por um contador de pulsos conectado ao relógio local  $Ck_{Tx}$ . Na seqüência, os pacotes assim montados são encaminhados, através da PSN, ao receptor.

Quando os pacotes chegam ao receptor, são desmontados e armazenados no *jitter buffer*, aguardando o instante de sua reprodução no fluxo TDM de saída, que é determinado pelo seu próprio relógio local  $Ck_{Rx}$ . A função do PLL é exatamente fazer com que  $Ck_{Tx}$  mantenha-se sempre preso (*locked*) a  $Ck_{Rx}$ , independentemente das flutuações dos osciladores e, fundamentalmente, do atraso variável sofrido pelos pacotes que trafegam através da rede.

O circuito PLL proposto, conceitualmente bastante similar àquele utilizado em aplicações de demodulação de sinais de áudio, como em receptores de rádio FM (Frequência Modulada), é composto por quatro módulos básicos, mostrados na Figura 4.6:

- a) **Detetor de fase:** é responsável pelo cálculo da diferença entre um sinal de referência, no caso o número de seqüência recebido em cada pacote, e a saída do PLL, no caso o número de seqüência presumido para o pacote, determinando o chamado Sinal de Erro;
- b) **Filtro de realimentação:** é um filtro de característica passa-baixas, que elimina os efeitos indesejados presentes no sinal de referência, como ruídos e, no caso, as flutuações decorrentes da diferença de atraso sofrida pelos pacotes dentro da PSN (PDV), gerando o chamado Sinal de Controle;
- c) **DCO (Digitally Controlled Oscillator):** é um oscilador controlado digitalmente, que gera o relógio utilizado no receptor, exatamente na freqüência determinada pelo Sinal de Controle;
- d) **Contador de Pulsos:** é um contador idêntico ao utilizado pelo transmissor para a geração dos números de seqüência dos pacotes, conectado à saída do DCO, que é responsável pela geração do número de seqüência presumido para os pacotes, que constitui a saída do PLL.

Em sua operação na sincronização de circuitos TDM emulados sobre redes de pacote, o PLL enfrenta dois grandes desafios opostos na sua implementação:

- Apresentar uma freqüência de corte bem definida, a fim de suavizar de forma adequada os efeitos da variação no atraso sofrido pelos pacotes (PDV) dentro da rede;
- Responder rapidamente às variações na freqüência do transmissor, percebida a partir dos pacotes recebidos, a fim de reduzir o chamado “tempo de aquisição”, ou seja, o

intervalo de tempo necessário para que a frequência gerada pelo DCO seja exatamente a frequência do oscilador local do transmissor, em regime permanente de operação.

A chave para compatibilização desses dois desafios mutuamente excludentes é exatamente o projeto adequado do Filtro de Realimentação, cujos requisitos fundamentais são:

- Eliminar de forma satisfatória os efeitos da variação no atraso dos pacotes (PDV) sobre o Sinal de Controle do DCO;
- Não retardar a resposta do PLL a flutuações na frequência do transmissor;
- Não elevar o tempo de recuperação do relógio durante o restabelecimento do pseudo-circuito, após mudanças na PSN como o rearranjo de rotas ou a recuperação de congestionamentos, uma vez que isso levaria a *overflow* do *jitter buffer*, com descarte de pacotes e geração de AIS para o enlace TDM.

Para melhor compreensão desses requisitos, é feita uma descrição matemática da operação do PLL, utilizando o desenvolvimento em [AWEYA2004], com as devidas adaptações e a ajuda do diagrama de blocos do PLL em malha fechada apresentado na Figura 4.7.

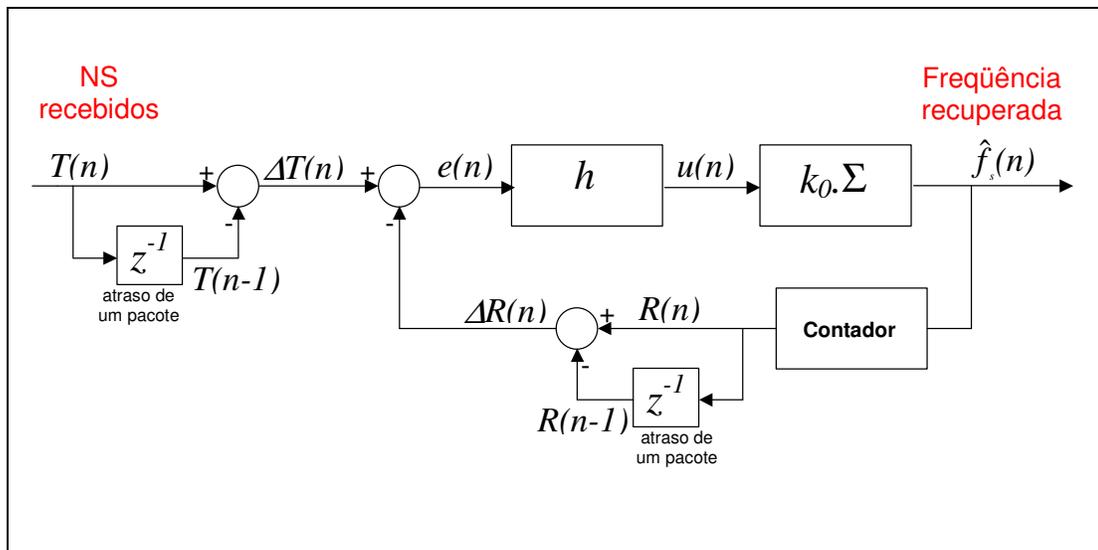


Figura 4.7 – Diagrama de blocos do PLL em malha fechada.

Considerando que existem dois osciladores completamente independentes no transmissor e no receptor, cujas frequências são definidas nas equações (4.1) e (4.2):

$$f_s = \frac{1}{\tau_s} \quad (4.1)$$

$$\hat{f}_s = \frac{1}{\hat{\tau}_s} \quad (4.2)$$

E definindo as seguintes variáveis discretas:

$n = 0,1,2,\dots \rightarrow$  valores discretos que representam os pacotes na PSN

$T(n) \rightarrow$  número de seqüência gerado no transmissor

$R(n) \rightarrow$  número de seqüência gerado no receptor

$d(n) \rightarrow$  atraso em pacotes sofrido pelo enésimo pacote na PSN

$d(n-1) \rightarrow$  atraso em pacotes sofrido pelo pacote (n - 1) na PSN

Tem-se a equação (4.3), que relaciona os números de seqüência do transmissor e receptor:

$$R(n) = T(n) + d(n) \quad (4.3)$$

Pode ser também definida matematicamente na equação (4.4) a variação no atraso sofrido por cada pacote  $j(n)$ , aqui denominada *jitter*, como sendo:

$$j(n) = d(n) - d(n-1) \quad (4.4)$$

Considerando ainda os intervalos entre pacotes gerados pelo transmissor, dados pela equação (4.5), que são constantes em função das características do circuito TDM emulado:

$$\Delta T(n) = T(n) - T(n-1) = K_{TDM} \quad (4.5)$$

E os intervalos entre os pacotes que chegam ao receptor, dados pela equação (4.6):

$$\Delta R(n) = R(n) - R(n-1) \quad (4.6)$$

A equação (4.6) pode se reescrita da seguinte forma, gerando a equação (4.7):

$$\Delta R(n) = R(n) - R(n-1)$$

$$\Delta R(n) = T(n) + d(n) - [T(n-1) + d(n-1)] = T(n) - T(n-1) + d(n) - d(n-1)$$

$$\Delta R(n) = \Delta T(n) + j(n)$$

$$\Delta R(n) = \Delta T(n) + j(n) = K_{TDM} + j(n) \quad (4.7)$$

Ou seja, o intervalo de chegada dos pacotes ao receptor apresenta duas componentes, uma constante, correspondente à duração dos dados TDM dentro do pacote, e outra variável, que corresponde ao *jitter* (ou PDV), no sentido definido acima.

Assim, o problema fundamental do PLL consiste em controlar a frequência de oscilação no receptor,  $\hat{f}_s$ , de modo que os dois intervalos entre pacotes sejam idênticos:

$$\Delta R(n) = \Delta T(n) \therefore j(n) = 0$$

Para isso, é calculado pelo Comparador de Fase o Sinal de Erro  $e(n)$  entre ambos, definido na equação (4.8), que corresponde ao próprio  $j(n)$  medido, como indicado na Figura 4.7:

$$e(n) = \Delta T(n) - \Delta R(n) = j(n) \quad (4.8)$$

Esse Sinal de Erro é aplicado ao Filtro de Realimentação, modelado como um sistema linear com função de transferência  $h$ , em princípio invariante no tempo, de modo que o Sinal de Controle  $u(n)$  é a resultante da convolução entre ambos, dada pela equação (4.9):

$$u(n) = h * e(n) \quad (4.9)$$

O Sinal de Controle, por sua vez, é aplicado ao DCO, que pode ser modelado, numa primeira análise, pela equação (4.10), onde  $K_o$  é o ganho do DCO:

$$\hat{f}_s = K_o \cdot \sum u(n) \quad (4.10)$$

Assim, na saída do DCO, é obtida a frequência do oscilador no receptor, como função direta do número de seqüência gerado através do Contador para os pacotes que chegam ao mesmo, conforme dado pela equação (4.11):

$$\begin{aligned} \hat{f}_s &= K_o \cdot \sum u(n) = K_o \cdot \sum h * e(n) = K_o \cdot \sum h * (\Delta T(n) - \Delta R(n)) = K_o \cdot \sum h * (K_{TDM} - \Delta R(n)) \\ \hat{f}_s &= K_o \cdot \sum h * [K_{TDM} - (R(n) - R(n-1))] \\ \hat{f}_s &= K_o \cdot \sum h * [K_{TDM} - (R(n) - R(n-1))] \end{aligned} \quad (4.11)$$

Como a frequência do DCO no receptor  $\hat{f}_s$ , utilizada para gerar  $R(n)$  através do contador na saída do PLL, nada mais é que uma estimativa da frequência  $f_s$  do oscilador utilizado no transmissor para geração de  $T(n)$  através do respectivo contador, tem-se a condição descrita

pela equação (4.12), que assegura a “captura” da frequência do transmissor pelo PLL, garantindo a sincronização do fluxo TDM de saída no receptor:

$$R(n) - R(n-1) = \Delta R(n) = \Delta T(n) = K_{TDM} \Leftrightarrow f_s = \hat{f}_s \quad (4.12)$$

Logo, fica provado matematicamente que o fluxo TDM que chega ao transmissor pode ser regenerado fielmente na saída do receptor através, exclusivamente, dos números de seqüência  $T(n)$  originais, recebidos dentro de cada pacote, e dos números de seqüência  $R(n)$  presumidos, gerados pelo próprio receptor, não havendo necessidade de transmissão explícita de marcações de tempo (*timestamps*) nos pacotes TDM.

Entretanto, a equação (4.11), apesar de direta, não pode ser implementada de maneira simples, já que os elementos representados pela função de transferência  $h$  e pelo integrador discreto  $\Sigma$ , são relativamente complexos, assim como a operação de convolução. Dessa forma, esses dois elementos e suas implementações precisam ser descritos em maiores detalhes antes que possa ser discutida a sua modelagem dentro do Simulink.

Assim, na Figura 4.8 é apresentado o diagrama de blocos para um filtro de realimentação do tipo EWMA (Exponencialmente Weighted Moving Average), acompanhado por um breve desenvolvimento de sua função de transferência em  $Z$ , conforme proposto em [AWEYA2004], onde podem ser encontradas maiores informações a respeito desse filtro, assim como as motivações do autor para sua utilização.

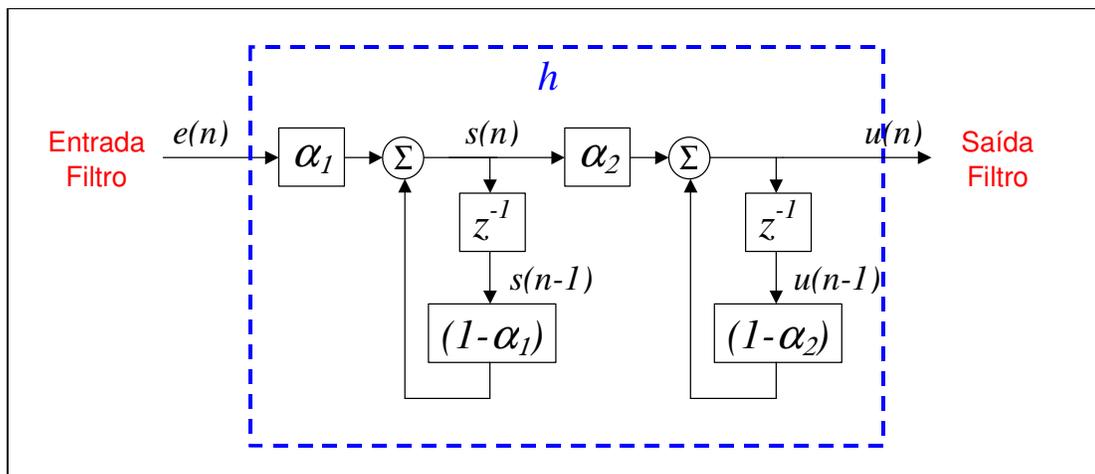


Figura 4.8 – Diagrama de blocos do filtro EWMA duplo.

Como é observado na seqüência dos blocos na Figura 4.8, o filtro é caracterizado por dois estágios idênticos em cascata, que podem ser matematicamente descritos pelo par de equações a diferenças (4.13) e (4.14), onde  $0 < \alpha_1, \alpha_2 < 1$ :

$$s(n) = (1 - \alpha_1).s(n-1) + \alpha_1.e(n) \quad (4.13)$$

$$u(n) = (1 - \alpha_2).s(n-1) + \alpha_2.s(n) \quad (4.14)$$

Aplicando a transformada Z a essas equações ou seguindo o fluxo dos blocos, é obtida a função de transferência apresentada na equação (4.15), onde  $\beta_1 = 1 - \alpha_1$  e  $\beta_2 = 1 - \alpha_2$ :

$$H(z) = \frac{U(z)}{E(z)} = \frac{\alpha_1.z}{z - (1 - \alpha_1)} \cdot \frac{\alpha_2.z}{z - (1 - \alpha_2)} = \frac{\alpha_1.\alpha_2.z^2}{z^2 - (\beta_1 + \beta_2).z + \beta_1.\beta_2} \quad (4.15)$$

Esse filtro, utilizando a parametrização apresentada em [AWEYA2004a], foi implementado no Simulink como referência para as simulações realizadas com o mecanismo alternativo de sincronização adaptativa proposto neste trabalho, utilizando uma técnica de controle clássico, o controlador PID, para geração do sinal de controle dentro do PLL, a partir do sinal de erro medido.

O controlador PID é a técnica de controle clássico de estrutura fixa mais utilizada nas aplicações industriais, em processos com funções de transferência aproximadas por sistemas lineares de 1ª ou 2ª ordem, permitindo assim a utilização de controladores baseados em poucos parâmetros, muitas vezes determinados de forma empírica. Para processos mais complexos, são utilizadas normalmente técnicas do chamado controle moderno, baseadas em realimentação de estado e implementadas por controladores digitais. Maiores informações sobre essas técnicas de controle podem ser encontradas em [DAZZO1988]. O termo PID significa Proporcional+Integral+Derivativo, que são as três ações independentes executadas por esse controlador sobre a variável determinada:

- **Ação Proporcional (P):** consiste basicamente na utilização de um ganho  $K_p$  aplicado ao sinal de erro, oferecendo resultados razoáveis para sistemas lineares de 1ª ordem que proporcionam, em malha fechada, um erro estático pequeno e uma resposta transitória bastante rápida. Em sistemas de ordem superior, no entanto, são necessários valores elevados de ganho para redução do erro estático, e isso costuma levar os sistemas à instabilidade, o que torna a ação proporcional pouco recomendável, isoladamente.

- **Ação Integral (I):** consiste basicamente em introduzir capacidade de memorização no processo, ponderada pelo ganho  $K_I$ , através da utilização de um integrador (pólo na origem do plano 's') no controlador, fazendo com que o sistema, em malha fechada, tenha resposta oscilatória e mais lenta, oferecendo bons resultados quando as características do sistema não atendem às especificações de erro nulo para determinadas entradas ou perturbações. Costuma ser sempre acompanhada por uma ação Proporcional.
- **Ação Derivativa (D):** consiste basicamente em introduzir sensibilidade à taxa de variação do processo, ponderada pelo ganho  $K_D$ , com o objetivo de melhorar as características transitórias do comportamento do sistema, oferecendo respostas rápidas às variações bruscas detectadas e ação nula sobre o regime permanente. Costuma ser sempre acompanhada pelas ações Proporcional e Integral.

Dessa forma, analisando os requisitos impostos ao Filtro de Realimentação utilizado pelo PLL, foi feita uma associação imediata com as características de ação independente de um controlador PID, sendo proposta a sua utilização como alternativa às abordagens tradicionais baseadas nos diversos tipos de filtros passa-baixas, como o EWMA citado anteriormente.

O controlador PID oferece a ação proporcional necessária para o ajuste de escala entre a variabilidade dos números de seqüência, dependente do volume de pacotes recebidos fora de ordem da PSN, com o sinal de controle do DCO e sua capacidade de deslocamento em torno da sua freqüência central. Ao mesmo tempo, a ação integral assegura, através da memorização, o amortecimento do ruído gerado pelas grandes variações no intervalo de recebimento dos pacotes, característica básica da função desempenhada pelos filtros passa-baixas. Finalmente, a ação derivativa possibilitaria uma resposta rápida quando do restabelecimento do pseudo-circuito em decorrência de variações na configuração da PSN ou escorregamento da freqüência do transmissor, garantindo um tempo de aquisição reduzido para o PLL. Assim, foi entendido que, com a sua utilização ao invés de um simples filtro, seria possível ajustar cada uma dessas características de forma independente, assegurando o melhor desempenho para o PLL em todas as situações.

O diagrama de blocos do controlador PID discreto utilizado é apresentado na Figura 4.9, assim como suas equações a diferenças e a respectiva função de transferência em Z utilizada na implementação Simulink.

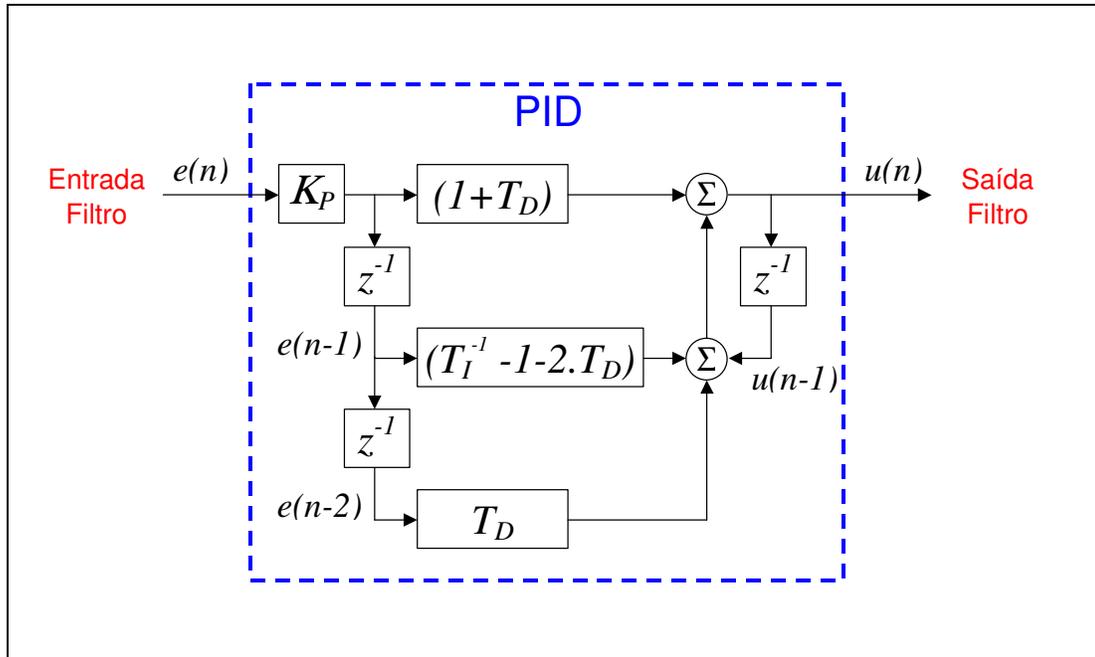


Figura 4.9 – Diagrama de blocos do controlador PID.

Através da simples álgebra de blocos aplicada à Figura 4.9, o controlador PID é matematicamente descrito pela equação a diferenças (4.16), onde  $K_p$  é o ganho Proporcional  $K_i = K_p \cdot \frac{1}{T_i}$  é o ganho Integral e  $K_d = K_p \cdot T_d$  é o ganho Derivativo:

$$u(n) = K_p \left[ e(n) + \frac{1}{T_i} \cdot \sum_{i=0}^{n-1} e(i) + T_d \cdot [e(n) - e(n-1)] \right] \quad (4.16)$$

Essa equação a diferenças representa literalmente as características do controlador, mas tem o inconveniente de necessitar, devido à componente integral, da armazenagem de todas as amostras anteriores. Contudo, como é observado pela seqüência de blocos na Figura 4.9, ela pode ser reescrita na forma da equação (4.16a), em função da saída anterior e de apenas duas das amostras anteriores, uma vez que o controlador PID é um sistema de 2ª. ordem.

$$\begin{aligned}
u(n) &= K_p \cdot e(n) + \frac{K_p}{T_I} \cdot \sum_{i=0}^{n-1} e(i) + K_p \cdot T_D \cdot [e(n) - e(n-1)] \\
u(n-1) &= K_p \cdot e(n-1) + \frac{K_p}{T_I} \cdot \sum_{i=0}^{n-2} e(i) + K_p \cdot T_D \cdot [e(n-1) - e(n-2)] \\
u(n) - u(n-1) &= K_p \cdot e(n) + \frac{K_p}{T_I} \cdot \sum_{i=0}^{n-1} e(i) + K_p \cdot T_D \cdot [e(n) - e(n-1)] - \\
&\quad - \left[ K_p \cdot e(n-1) + \frac{K_p}{T_I} \cdot \sum_{i=0}^{n-2} e(i) + K_p \cdot T_D \cdot [e(n-1) - e(n-2)] \right] \\
u(n) &= u(n-1) + K_p \cdot e(n) + \frac{K_p}{T_I} \cdot e(n-1) + \frac{K_p}{T_I} \cdot \sum_{i=0}^{n-2} e(i) + K_p \cdot T_D \cdot e(n) - K_p \cdot T_D \cdot e(n-1) - \\
&\quad - K_p \cdot e(n-1) - \frac{K_p}{T_I} \cdot \sum_{i=0}^{n-2} e(i) - K_p \cdot T_D \cdot e(n-1) + K_p \cdot T_D \cdot e(n-2) \\
u(n) &= u(n-1) + K_p \cdot [1 + T_D] \cdot e(n) + K_p \cdot \left[ \frac{K_p}{T_I} - 1 - 2 \cdot T_D \right] \cdot e(n-1) + K_p \cdot [T_D] \cdot e(n-2) \\
u(n) &= u(n-1) + K_p \cdot [1 + T_D] \cdot e(n) + K_p \cdot \left[ \frac{K_p}{T_I} - 1 - 2 \cdot T_D \right] \cdot e(n-1) + K_p \cdot [T_D] \cdot e(n-2) \quad (4.16a)
\end{aligned}$$

Aplicando a transformada Z à equação (4.16a) e reagrupando os termos, é obtida a função de transferência para o controlador PID, apresentada na equação (4.17):

$$\begin{aligned}
Z\{u(n)\} &= Z \left\{ u(n-1) + K_p \cdot [1 + T_D] \cdot e(n) + K_p \cdot \left[ \frac{K_p}{T_I} - 1 - 2 \cdot T_D \right] \cdot e(n-1) + K_p \cdot [T_D] \cdot e(n-2) \right\} \\
Z\{u(n)\} &= Z\{u(n-1)\} + Z\{Kp \cdot [1 + T_D] \cdot e(n)\} + Z \left\{ Kp \cdot \left[ \frac{1}{T_I} - 1 - 2 \cdot T_D \right] \cdot e(n-1) \right\} + Z\{Kp \cdot [T_D] \cdot e(n-2)\} \\
U(z) &= z^{-1} \cdot U(z) + Kp \cdot [1 + T_D] \cdot E(z) + Kp \cdot \left[ \frac{1}{T_I} - 1 - 2 \cdot T_D \right] \cdot z^{-1} \cdot E(z) + Kp \cdot [T_D] \cdot z^{-2} \cdot E(z) \\
(1 - z^{-1}) \cdot U(z) &= Kp \cdot \left[ [1 + T_D] + \left[ \frac{1}{T_I} - 1 - 2 \cdot T_D \right] \cdot z^{-1} + [T_D] \cdot z^{-2} \right] \cdot E(z) \\
\frac{U(z)}{E(z)} &= Kp \cdot \frac{[1 + T_D] + \left[ \frac{1}{T_I} - 1 - 2 \cdot T_D \right] \cdot z^{-1} + [T_D] \cdot z^{-2}}{(1 - z^{-1})} \cdot \frac{z^2}{z^2} \\
\frac{U(z)}{E(z)} &= Kp \cdot \frac{[1 + T_D] \cdot z^2 + \left[ \frac{1}{T_I} - 1 - 2 \cdot T_D \right] \cdot z + [T_D]}{(z - 1) \cdot z} \\
H(z) = \frac{U(z)}{E(z)} &= K_p \cdot \frac{[(1 + T_D) \cdot z^2 - (1 + 2 \cdot T_D - \frac{1}{T_I}) \cdot z + T_D]}{(z - 1) \cdot z} \quad (4.17)
\end{aligned}$$

O segundo elemento de maior complexidade para a construção do PLL é o DCO, que na verdade acaba sendo implementado de forma bastante simples, considerando as características disponíveis nos contadores digitais. Um DCO é na verdade um divisor de frequência por  $N$ , sendo este programável, ao qual é aplicado um relógio de frequência mais elevada, tipicamente acima de 100 para o caso do PLL, conforme apresentado na Figura 4.10 e no desenvolvimento matemático correspondente.

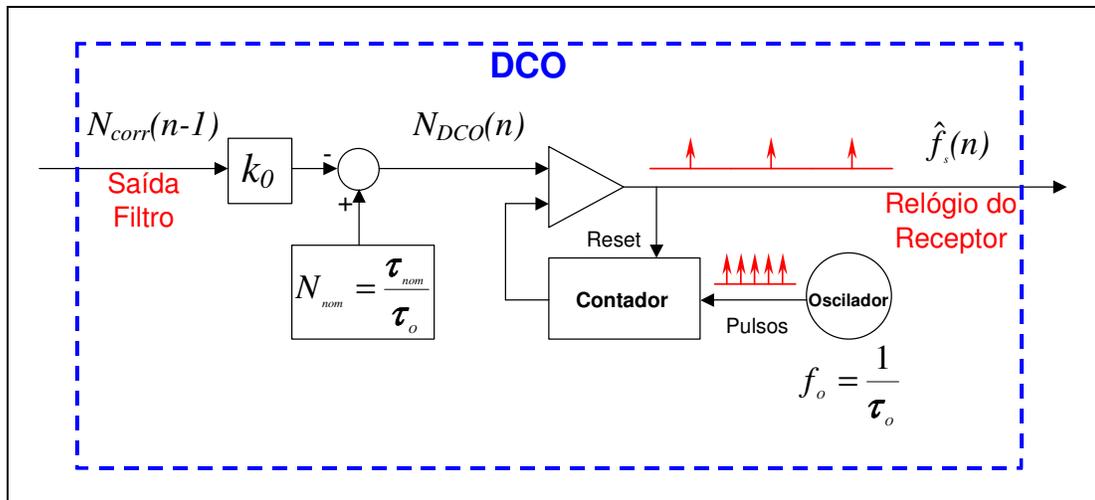


Figura 4.10 – Diagrama de blocos do DCO.

O DCO é construído a partir de um contador, ao qual são aplicados os pulsos produzidos por um oscilador fixo independente de alta frequência, definida pela equação (4.18):

$$f_o = \frac{1}{\tau_o} \quad (4.18)$$

A saída desse contador é aplicada a um comparador, que realiza a comparação da contagem com um valor  $N_{DCO}(n)$  de referência, correspondente à entrada do DCO, enquanto sua saída é ligada à entrada de reinicialização (*reset*) do contador, como mostrado na Figura 4.10.

Assim, sempre que a contagem alcança o valor de referência  $N_{DCO}(n)$ , o contador é reiniciado, começando um novo ciclo. Isso significa que a frequência do oscilador fixo aplicado na entrada do contador aparece, na saída do comparador, dividida pelo valor  $N_{DCO}(n)$ . Tomando-se a saída do comparador como sinal de saída do DCO e assumindo que

o valor de referência  $N_{DCO}(n)$  é uma entrada digital de controle, é implementado um oscilador digitalmente controlável, cuja frequência de saída é definida pela equação (4.19).

$$f_{DCO}(n) = \frac{1}{\tau_{DCO}(n)} \quad (4.19)$$

Assumindo para esse DCO uma frequência central, definida pela equação (4.20), cujo valor corresponde à divisão da frequência do oscilador fixo  $f_o$  por um valor  $N_{nom}$ , definido pela equação (4.21), é caracterizada a condição demonstrada na Figura 4.10:

$$f_{nom} = \frac{1}{\tau_{nom}} \quad (4.20)$$

$$N_{nom} = \frac{\tau_{nom}}{\tau_o} \quad (4.21)$$

Considerando ainda um sinal de correção  $N_{corr}(n-1)$  aplicado quando da chegada do pacote anterior, definido na equação (4.22) como um ajuste  $\tau_{corr}(n-1)$  no período de oscilação do DCO, dado pelo ganho  $K_o$ , que resulta no novo período  $\tau_{DCO}(n)$ , dado pela equação (4.23):

$$N_{corr}(n-1) = \frac{1}{K_o} \cdot \frac{\tau_{corr}(n-1)}{\tau_o} \quad (4.22)$$

$$\tau_{DCO}(n) = \tau_{nom} - \tau_{corr}(n-1) \quad (4.23)$$

Aplicando as definições anteriores em (4.23), são obtidas as equações (4.24) e (4.25), para o período  $\tau_{DCO}(n)$  e o divisor  $N_{DCO}(n)$  em função do sinal de correção  $N_{corr}(n-1)$ :

$$\tau_{DCO}(n) = \tau_{nom} - \tau_{corr}(n-1)$$

$$\tau_{DCO}(n) = N_{nom} \cdot \tau_o - K_o \cdot N_{corr}(n-1) \cdot \tau_o$$

$$\tau_{DCO}(n) = [N_{nom} - N_{corr}(n-1)] \cdot \tau_o$$

$$\frac{\tau_{DCO}(n)}{\tau_o} = [N_{nom} - N_{corr}(n-1)] = N_{DCO}(n)$$

$$\tau_{DCO}(n) = [N_{nom} - K_o \cdot N_{corr}(n-1)] \tau_o \quad (4.24)$$

$$N_{DCO}(n) = N_{nom} - K_o \cdot N_{corr}(n-1) \quad (4.25)$$

Dessa forma, como  $N_{corr}(n-1)$  corresponde à saída do filtro de realimentação, ou seja, ao sinal de controle  $u(n-1)$ , o período de oscilação resultante é, na verdade, uma estimativa do período do relógio utilizado pelo transmissor  $\tau_s$ , conforme indicado pela expressão (4.26):

$$\tau_{DCO} = \hat{\tau}_s = \frac{1}{\hat{f}_s} \xleftarrow{\text{estimativa}} \tau_s = \frac{1}{f_s} \quad (4.26)$$

Descrito o funcionamento de cada um de seus blocos componentes, na Figura 4.11, é apresentada a operação completa do PLL para a recuperação do sincronismo no receptor.

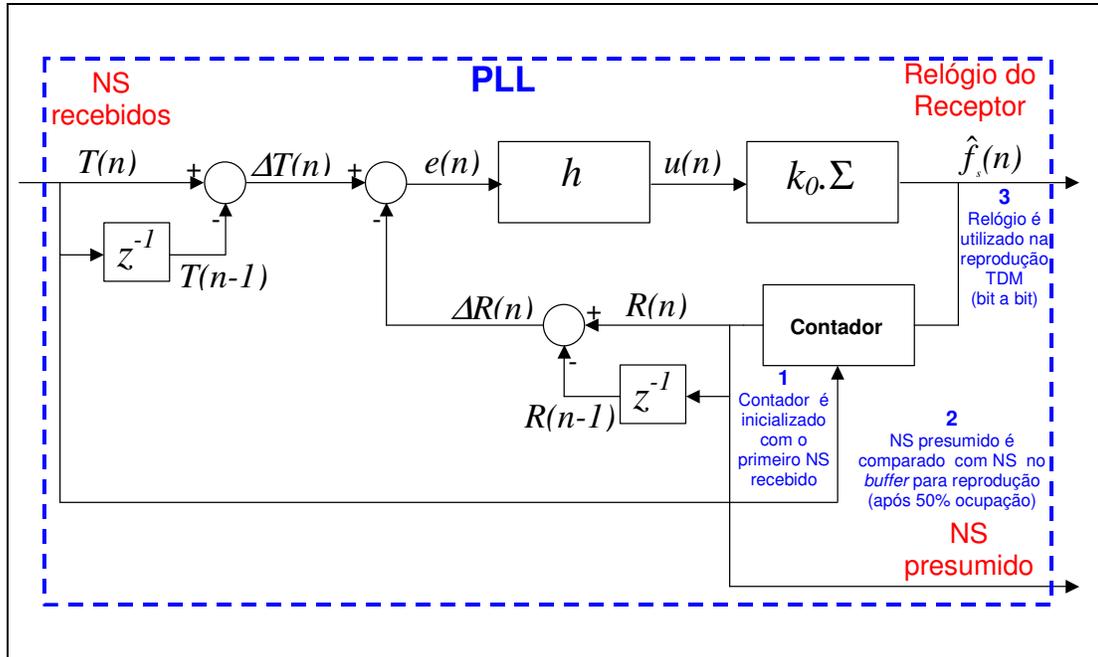


Figura 4.11– Operação do PLL na recuperação do sincronismo no receptor.

Quando o primeiro pacote é recebido, o contador é inicializado com o valor do seu número de seqüência (1); conforme os demais pacotes vão sendo recebidos, o PLL calcula a diferença entre o número de seqüência do pacote que chega  $T(n)$  e o anterior  $T(n-1)$  e inicia a operação em malha fechada, comparando essa diferença  $\Delta T(n)$  com a saída atual  $R(n)$  do contador  $R(n)$ , subtraída da contagem anterior  $R(n-1)$ , ou seja,  $\Delta R(n)$ . A diferença entre os dois valores gera um sinal de erro  $e(n)$ , que é então filtrado por  $h$ , cuja saída  $u(n)$  é utilizada para ajustar a freqüência do DCO,  $\hat{f}_s$ , que gera os pulsos para o contador, fechando assim a malha de realimentação. Quando o intervalo entre números de seqüência recebidos  $\Delta T(n)$  é maior que o intervalo na contagem  $\Delta R(n)$ , significa que a freqüência do DCO está mais rápida que a de transmissão, devendo ser reduzida através do aumento do divisor  $N_{DCO}$ ; caso contrário, a freqüência do DCO deve ser acelerada pela redução desse divisor, até que ambos os intervalos sejam iguais. Nessa condição, os números de seqüência gerados no

contador serão idênticos aos armazenados no *jitter buffer* para reprodução de cada conjunto de dados TDM, após o período de espera necessário para acomodação do atraso médio introduzido pela PSN, tipicamente 50% da ocupação do *buffer* (2) em regime permanente. Ao mesmo tempo, a frequência na saída do DCO corresponde exatamente ao relógio utilizado na montagem dos pacotes com o fluxo TDM no transmissor, permitindo desmontagem sincronizada e regeneração exata desse fluxo, bit a bit, no receptor (3).

O mecanismo de regeneração do relógio no receptor através do PLL foi implementado no Simulink, conforme apresentado na Figura 4.12, fazendo as devidas adaptações para o ambiente de simulação, com o objetivo de avaliar seu desempenho nas condições encontradas na emulação de circuitos TDM sobre PSN, como variação no atraso e perda de pacotes, bem como leves desvios na frequência do transmissor, analisando o tempo de captura do PLL para condições de *jitter* e o *wander* presentes no fluxo TDM.

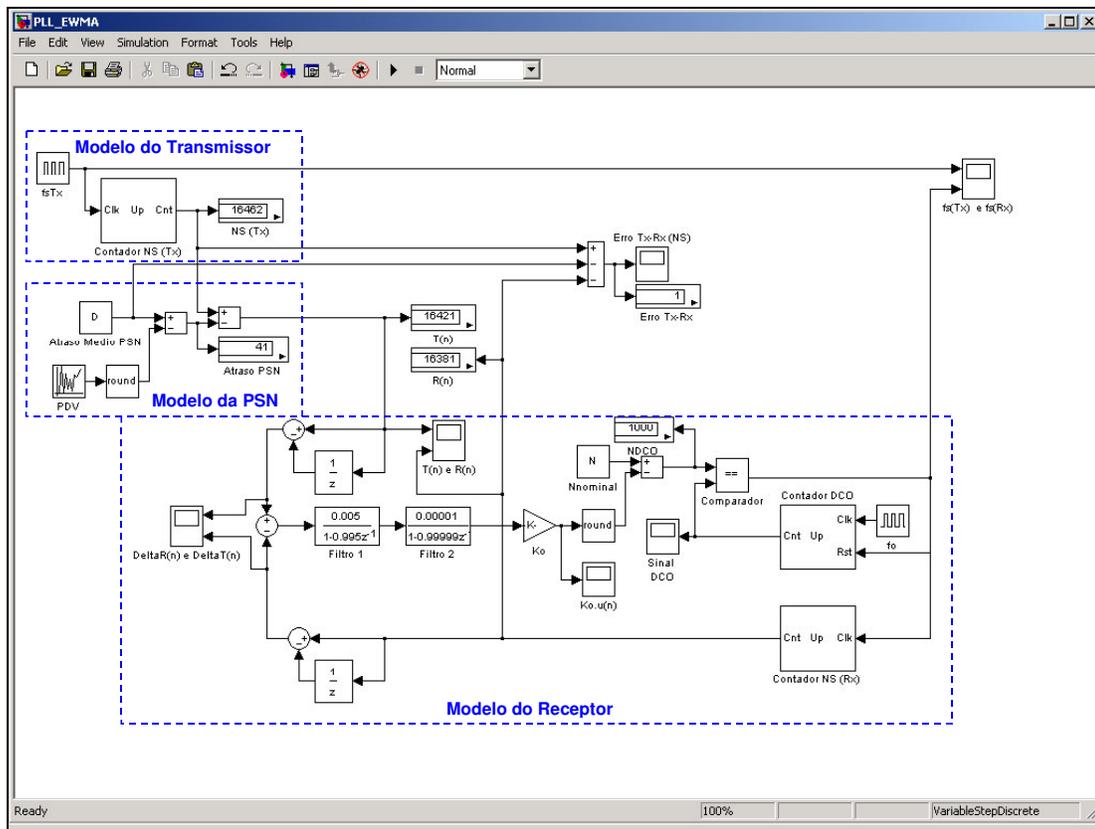


Figura 4.12 – Implementação do PLL com filtro EWMA no Simulink.

O modelo foi construído dentro do Simulink utilizando os blocos disponíveis para a implementação do PLL, e também modelos de osciloscópio e *display* para a monitoração contínua dos principais sinais envolvidos., como o atraso nos pacotes provocado pela PSN, a saída do DCO, os sinais  $\Delta T(n)$  e  $\Delta R(n)$ , o valor dos contadores no transmissor, recebido  $T(n)$  e regenerado  $R(n)$  no receptor, a diferença entre ambos  $d(n)$ , as frequências no transmissor  $f_s$  e no receptor  $\hat{f}_s$ , o sinal de controle do DCO,  $K_o.u(n)$ , e o valor do divisor de frequência corrente,  $N_{DCO}(n)$ .

O modelo criado utiliza quatro parâmetros de configuração, que precisam ser previamente definidos antes que a simulação seja executada:

- **Intervalo entre geração de pacotes (T)**, em milissegundos, que corresponde ao período do oscilador no transmissor  $\tau_s$ ;
- **Divisor nominal de frequência (N)**, o valor de  $N_{nom}$  que determina a frequência central do DCO, configurando o número de vezes que o período  $\tau_o$  do oscilador de alta frequência é superior a  $\tau_s$ , e conseqüentemente o passo disponível para ajuste do DCO;
- **Atraso médio PSN (D)**, o valor médio de atraso sofrido pelos pacotes que atravessam a PSN, utilizado para compensação da diferença entre os contadores do receptor e do transmissor;
- **Número de seqüência inicial (NS1)**, o número de seqüência do primeiro pacote recebido, que é carregado como valor inicial do contador do receptor.

A simulação do mecanismo de sincronização desenvolvida no Simulink é composta por três modelos, indicados na Figura 4.12 através de linhas tracejadas e descritos a seguir:

- **Modelo do transmissor**, bastante simples, implementado através de um contador, cuja saída é o número de seqüência dos pacotes transmitidos, alimentado por um oscilador cuja frequência corresponde à taxa constante de geração de pacotes do fluxo TDM;
- **Modelo da PSN**, implementado através de um deslocamento negativo do número de seqüência gerado pelo transmissor, correspondente ao atraso médio sofrido pelos pacotes dentro da rede de transporte, mais uma componente aleatória de distribuição normal em torno do atraso médio e variância de modo a assegurar que 99% dos atrasos estarão compreendidos no intervalo  $[D/2; 3D/2]$ ; com o objetivo de simular o comportamento variável do número de seqüência dos pacotes que chegam ao receptor;

- **Modelo do receptor**, o PLL em si, implementado através da composição dos blocos que realizam as funções anteriormente descritas para cada um de seus componentes. Para o filtro EWMA, foram utilizados os coeficientes experimentais apresentados em [AWEYA2004a], entendidos como os melhores ajustes obtidos para esse filtro pelo autor na aplicação PLL, sendo também buscada uma equivalência, dentro das limitações do Simulink, entre os parâmetros de configuração utilizados naquela implementação e aqueles configurados no modelo, conforme mostrado na Tabela 4.2.

Tabela 4.2 – Equivalência entre parâmetros experimentais e os utilizados no PLL Simulink.

Parâmetro	Unidade	wanPMC-2T3E3	Simulink
NS inicial (NS1)		aleatório	16000
$\tau_s$ (T)	ms	0,000022	0,2
$f_o/f_{nom}$ (N)		20.000.000	1.000
Atraso PSN (D)	ms	100	100
Precisão ajuste DCO	ppm	0,05	1000
Desvio $f_{DCO}$ max	ppm	125,25	111
Desvio $f_{DCO}$ min	ppm	-79,5	-90
$u(n)$ para $f_{DCO}$ max		0,00444	0,00444
$u(n)$ para $f_{DCO} = f_{nom}$		0,00000	0,00000
$u(n)$ para $f_{DCO}$ min		-0,00360	-0,00360
Frequência $f_{nom}$	kHz	44.736	5
Desvio min de $f_{nom}$	Hz	5.603	0,5
Desvio min de $f_{nom}$	Hz	-3.557	-0,4
Ganho DCO ( $K_o$ )		25.466	25

Na implementação em referência, é utilizado um feixe T3, de 44,736 Mbps, sendo controlada diretamente a frequência de relógio do fluxo de bits. Na simulação, é utilizada a taxa de geração dos pacotes e não a frequência da linha, que é de aproximadamente 5000 (4.723) pacotes/s, considerando os dois quadros T3 por pacote, com 592 octetos cada um. Dessa forma, a  $f_{nom}$  utilizada na simulação é 5 kHz, considerando um pacote gerado a cada 200us. Considerando um  $N_{nom}$  de 1000, o erro de quantização para a frequência é inferior a  $0,5^\circ$ , permitindo passos de ajuste de 0,5Hz, suficiente para os objetivos aqui propostos. Dessa forma, o ganho DCO é ajustado para que a saída do filtro de realimentação produza efeitos similares no desvio de frequência, assegurando a equivalência entre os parâmetros.

Inicialmente, o modelo Simulink foi validado utilizando as mesmas condições simuladas no MatLab® para o mecanismo de seqüenciamento de pacotes, ou seja, considerando a emulação de um circuito E1, com 1ms de dados TDM por pacote e, portanto, taxa de geração de 1000 pacotes/s. Da mesma forma, foi considerado um atraso médio na PSN de 80ms, admitindo-se a variação desse atraso dentro do intervalo [40ms;120ms], embora nesse caso usando uma distribuição normal ao invés de uniforme, sendo o resultado obtido para o erro entre as contagens  $T(n)$  e  $R(n)$  apresentado na Figura 4.13.

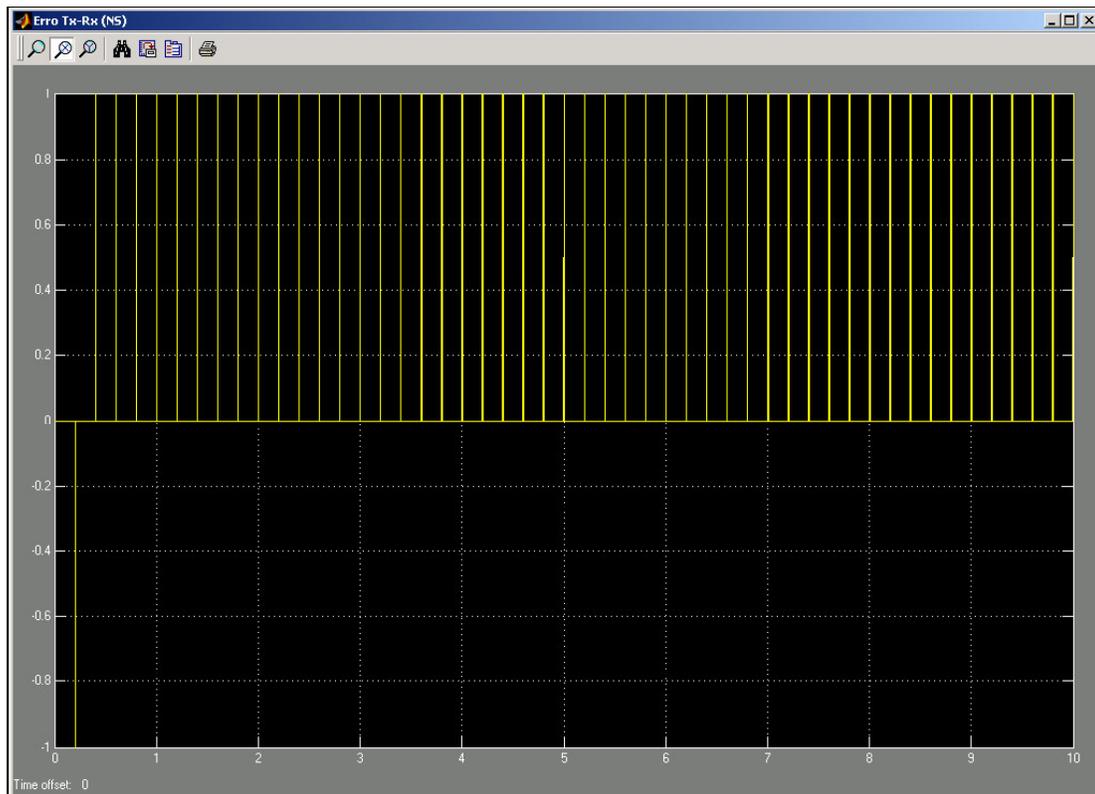


Figura 4.13 –Evolução do erro entre  $T(n)$  e  $R(n)$  para o filtro EWMA no Simulink.

Para a simulação, foram utilizados os parâmetros de equivalência da Tabela 4.2, sendo criado um experimento semelhante a um dos testes realizados em [AWEYA2004a], para avaliação da resposta do PLL à flutuação na frequência do transmissor. Assim, foi gerado um desvio de  $-7,5\text{Hz}$  (0,15% ou 15000ppm) na frequência de 5kHz utilizada para a montagem de pacotes no transmissor, e monitorado o erro entre  $T(n)$  e  $R(n)$  durante o processo de reação do PLL para compensação desse desvio no receptor.

O resultado obtido para o erro entre as contagens  $T(n)$  e  $R(n)$ , utilizando o PLL com filtro EWMA, para PDV de 15ms no modelo da rede, é mostrado na Figura 4.14.

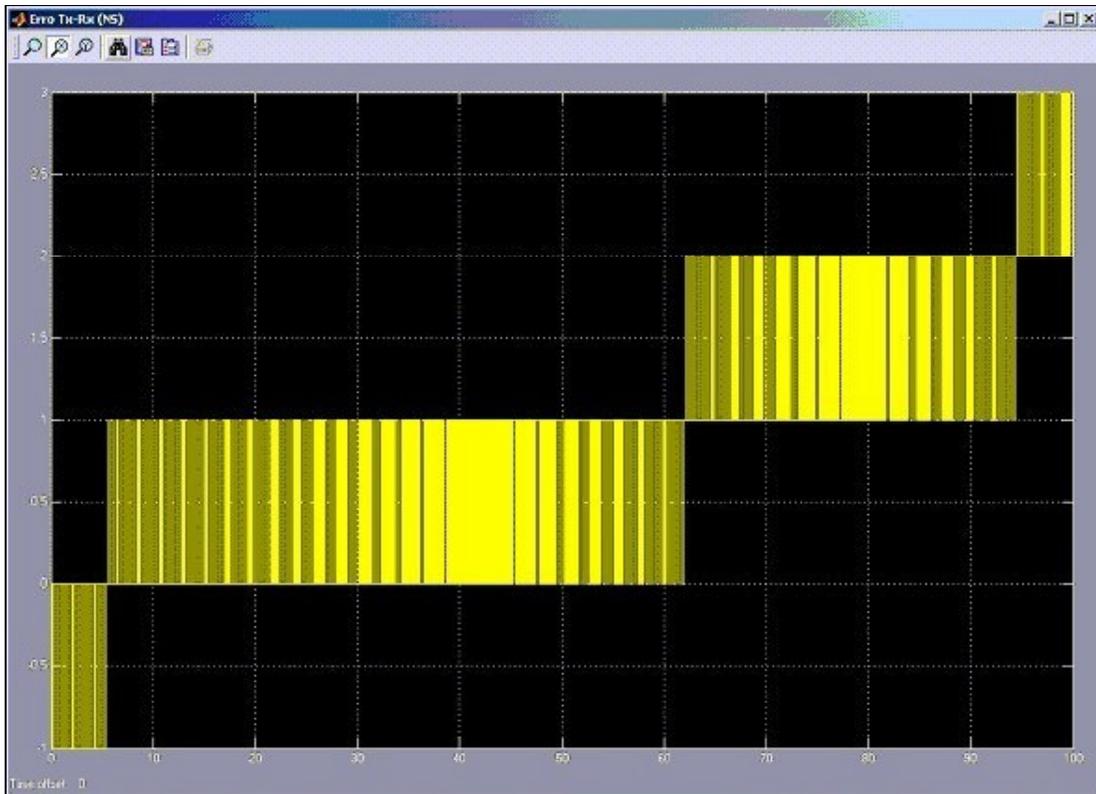


Figura 4.14 – Evolução do erro  $T(n) - R(n)$  para o filtro EWMA c/desvio de  $-7,5\text{Hz}$ .

Pode ser observado na Figura 4.14 que o PLL responde à variação intencional da frequência no transmissor, percebida como um erro negativo entre as contagens  $T(n)$  e  $R(n)$ , desacelerando a frequência no receptor. Contudo, esse efeito é forte demais, gerando um erro positivo crescente que não consegue ser revertido nos 100ms de simulação.

Na Figura 4.15 é mostrada a evolução do sinal na saída do filtro EWMA, utilizado para controle do DCO, onde pode ser verificada a resposta lenta desse sinal em função de sua característica superamortecida. Para que a frequência do transmissor seja efetivamente capturada, o sinal de controle deve estar entre +1 e +2, ao invés dos  $-7,5$  indicados ao final dos primeiros 100ms, o que levará pelo menos mais 150ms, assumindo que não exista nova ultrapassagem. Com isso, o tempo de captura do PLL é projetado em mais de 250ms para um desvio de  $-0,15\%$  na frequência, quando é utilizado o filtro EWMA.

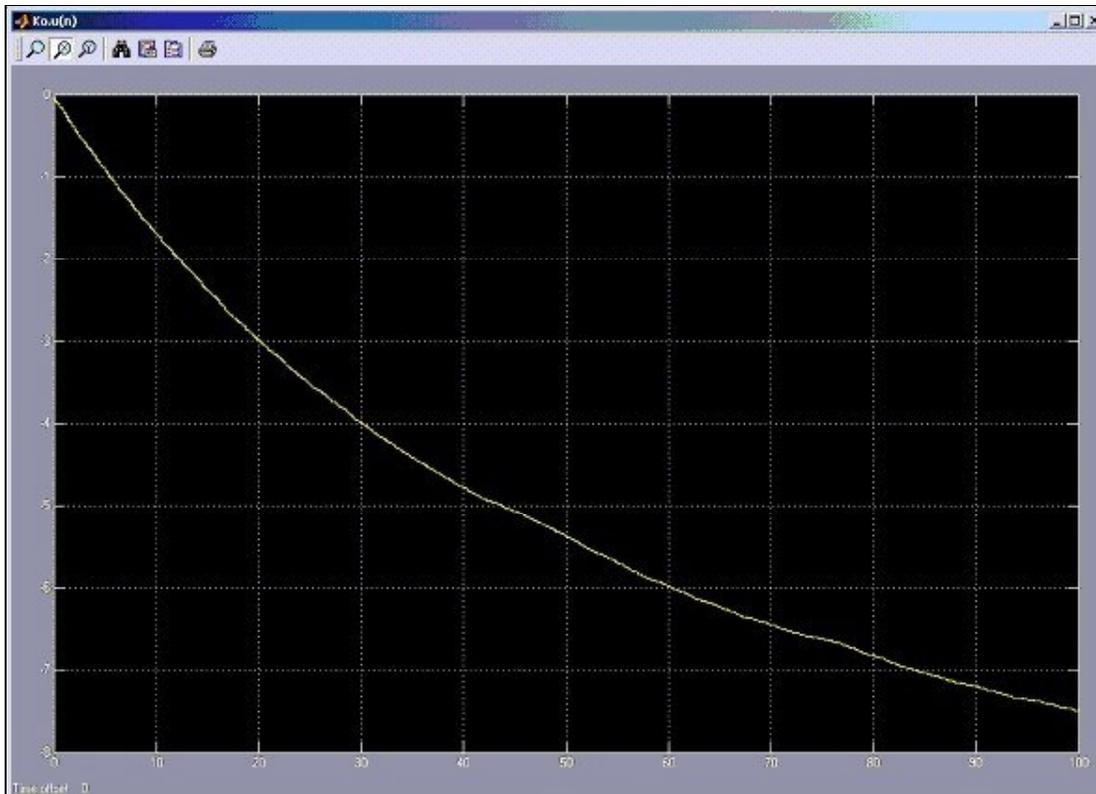


Figura 4.15 – Sinal de controle para o filtro EWMA  $c/\text{desvio}$  de  $-7,5\text{Hz}$

Uma vez obtido, através da simulação, o desempenho do PLL com o filtro EWMA apresentado em [AWEYA2004a], foi feita a substituição desse filtro pelo controlador PID proposto nesse trabalho.

Para ajuste inicial dos parâmetros utilizados pelo controlador,  $K_P$ ,  $T_I$  e  $T_D$ , foram utilizadas as regras empíricas definidas pelo método Zieger-Nichols (1942) para sistemas de controle em malha fechada, determinando através da variação de  $K_P$ , com  $T_I$  e  $T_D$  nulos, o limiar de estabilidade do sistema, caracterizado pelo comportamento oscilatório do erro entre as contagens; considerando  $K_O = K_P$  (oscilatório),  $T_O$  o período de oscilação e utilizando a Tabela 4.3.

Tabela 4.3 – Regras de Zieger-Nichols em malha fechada para ajuste de controladores PID.

Controlador	$K_P$	$T_I$	$T_D$
P	$0,5 K_O$	-	-
PI	$0,4 K_O$	$0,8 T_O$	-
PID	$0,6 K_O$	$0,5 T_O$	$0,12 T_O$

O modelo Simulink para o PLL com o controlador PID é apresentado na Figura 4.16.

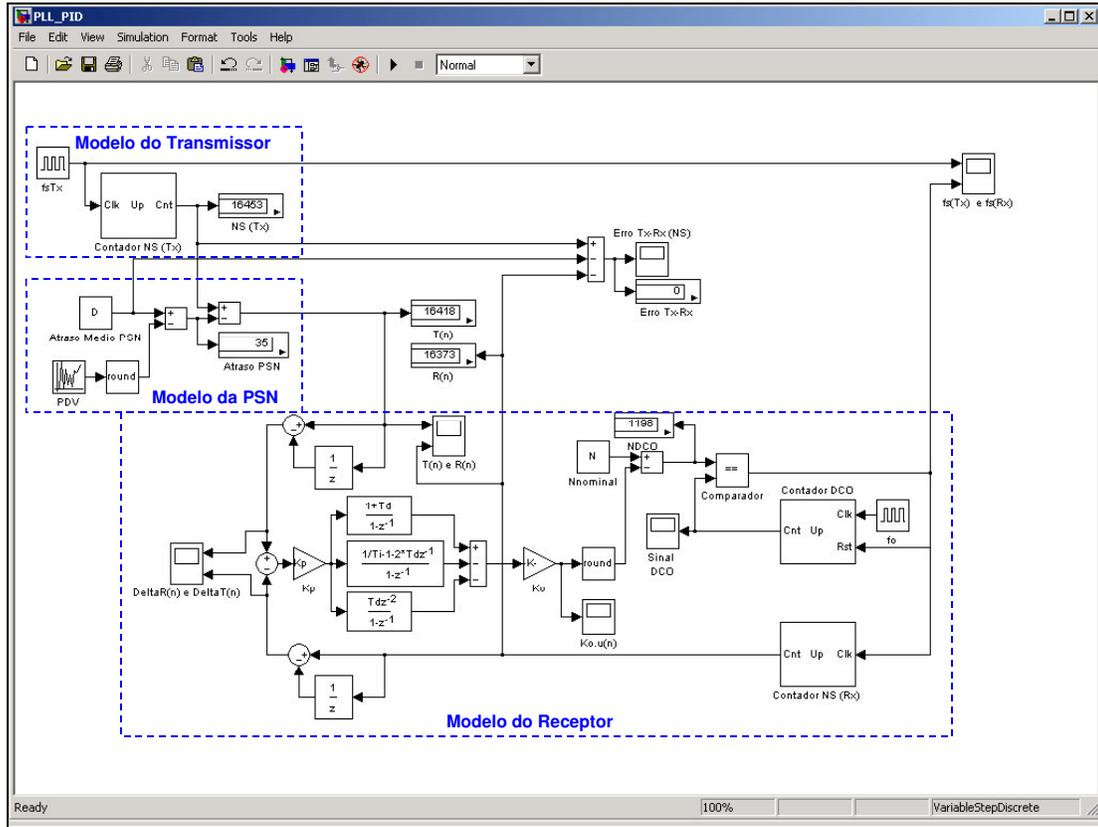


Figura 4.16 – Implementação do PLL com o controlador PID proposto no Simulink.

Esse modelo utiliza, além dos mesmos quatro parâmetros de configuração do modelo anterior, os três parâmetros adicionais referentes ao controlador PID, que precisam ser definidos antes que a simulação seja executada:

- **Ganho proporcional ( $K_p$ )**, valor correspondente ao ganho da ação proporcional;
- **Termo integral ( $T_i$ )**, valor que, dividindo o ganho proporcional, pondera a ação integral do controlador PID;
- **Termo derivativo ( $T_d$ )**, valor que, multiplicando o ganho proporcional, pondera a ação derivativa do controlador PID.

Após diversas análises de desempenho apresentado segundo os critérios definidos na Tabela 4.3, com ajustes empíricos, os parâmetros do controlador foram ajustados para:

$$\begin{aligned}
 K_P &= 1 \\
 T_I &= 0,002 \\
 T_D &= 0,000001
 \end{aligned}$$

Dessa forma, foi repetido para esse modelo o experimento anterior, nas mesmas condições. Os resultado obtido para o erro entre as contagens  $T(n)$  e  $R(n)$ , também com PDV de 15ms no modelo da rede, é mostrado na Figura 4.17.

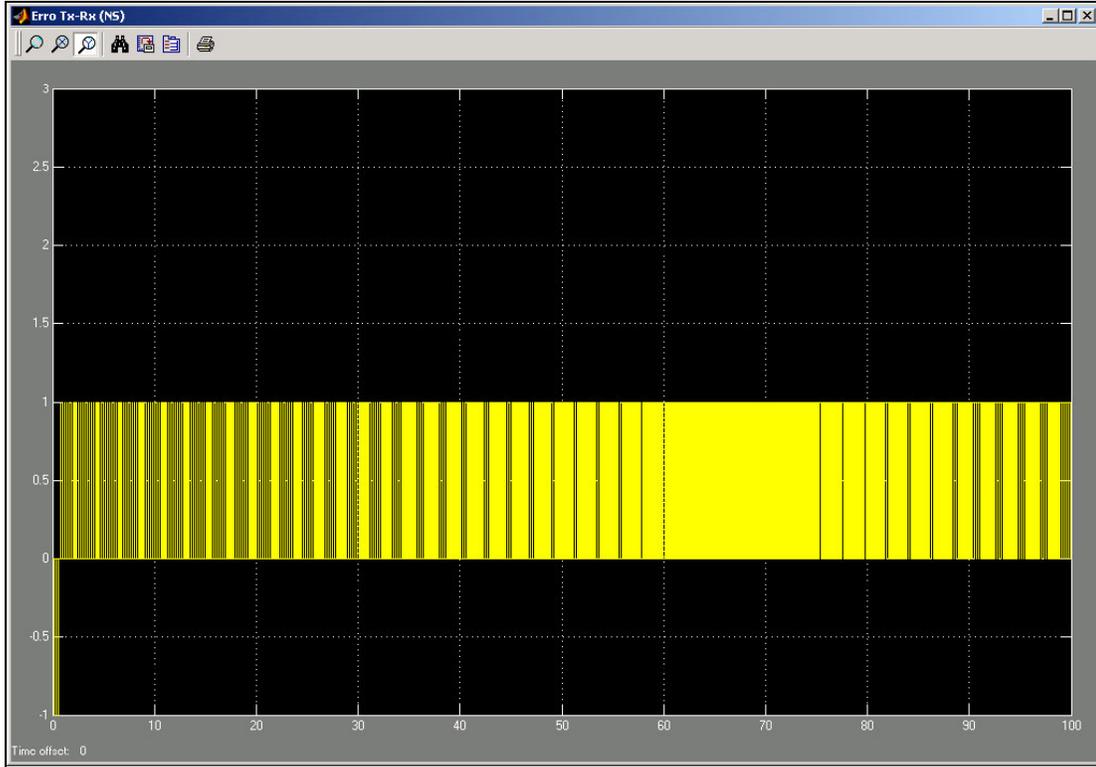


Figura 4.17 – Evolução do erro  $T(n) - R(n)$  para o controlador PID c/desvio de  $-7,5\text{Hz}$ .

Como pode ser observado na Figura 4.17, o erro manteve-se entre 0 e +1 ao longo de praticamente todo o período de simulação, após a rápida correção do valor  $-1$ , em virtude do receptor estar com a frequência  $7,5\text{Hz}$  mais rápida que o transmissor, permanecendo nesse intervalo mesmo com a variação aleatória no atraso sofrido pelos pacotes, inerente ao modelo utilizado para a PSN.

Na Figura 4.18 é mostrada a evolução do sinal na saída do filtro controlador PID, utilizado para controle do DCO de modo a assegurar a manutenção das diferenças entre as contagens  $T(n)$  e  $R(n)$ . Como a resolução utilizada é de  $5\text{ Hz}$ , pois  $N=1000$ , a oscilação entre  $+1$  e  $+3$  para esse sinal de controle pode ser considerada normal em regime permanente, o que significa um tempo de captura do PLL de cerca de  $1\text{ms}$  para um desvio de  $-0,15\%$  na frequência, quando é utilizado o controlador PID.

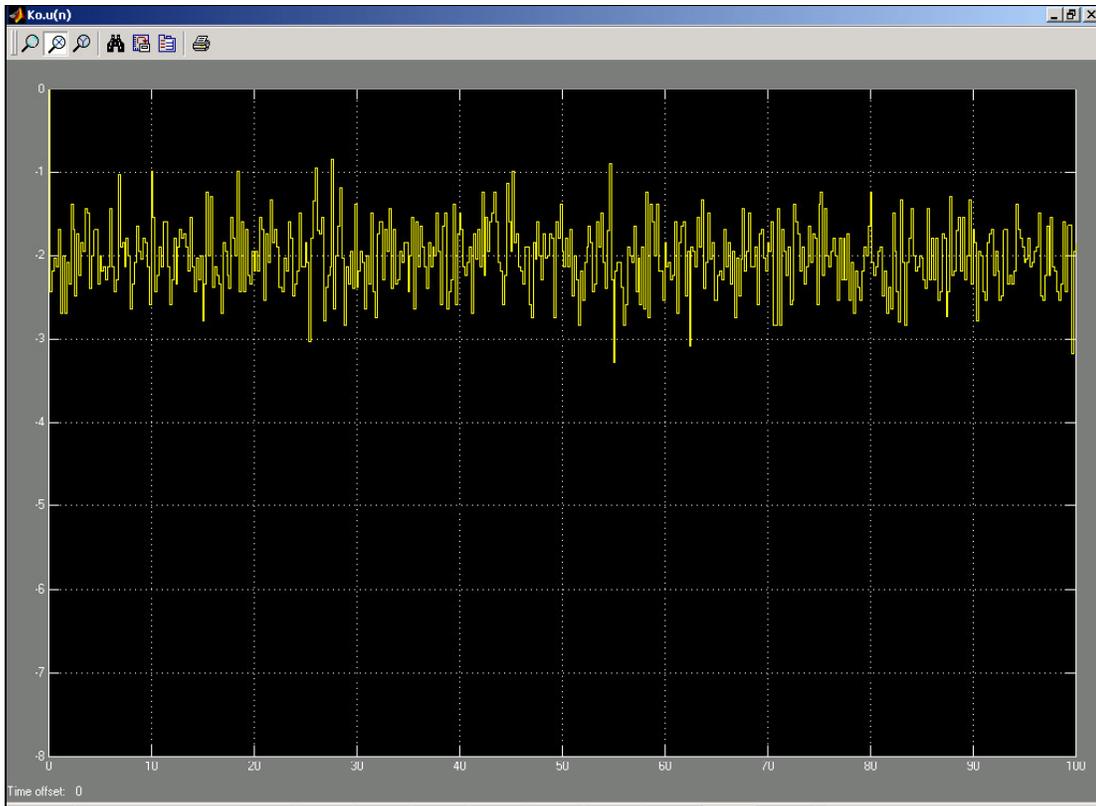


Figura 4.18 – Sinal de controle para o controlador PID c/desvio de  $-7,5\text{Hz}$ .

Comparando os resultados obtidos entre as duas implementações PLL, através das Figuras 4.14 e 4.17, que apresentam a evolução do erro entre as contagens  $T(n)$  e  $R(n)$ , tomada na mesma escala, é verificado um desempenho bastante superior do controlador PID na simulação.

#### 4.2 - EXPERIMENTAÇÃO TDMoIP® NA REDE LABCOM/U<sub>n</sub>B

Após o estudo realizado para compreensão das características, pontos fortes e limitações da emulação de circuitos TDM sobre redes comutadas em modo pacotes, propondo novos mecanismos para tratamento das questões-chave envolvidas e validando esses mecanismos através de simulação, foram buscadas formas de experimentação da tecnologia PWE3 e dos novos algoritmos propostos, a fim de assegurar a sua validade sob condições reais de operação nas redes existentes. Para isso, foi concebido um ambiente experimental, cujo modelo conceitual é apresentado na Figura 4.19.

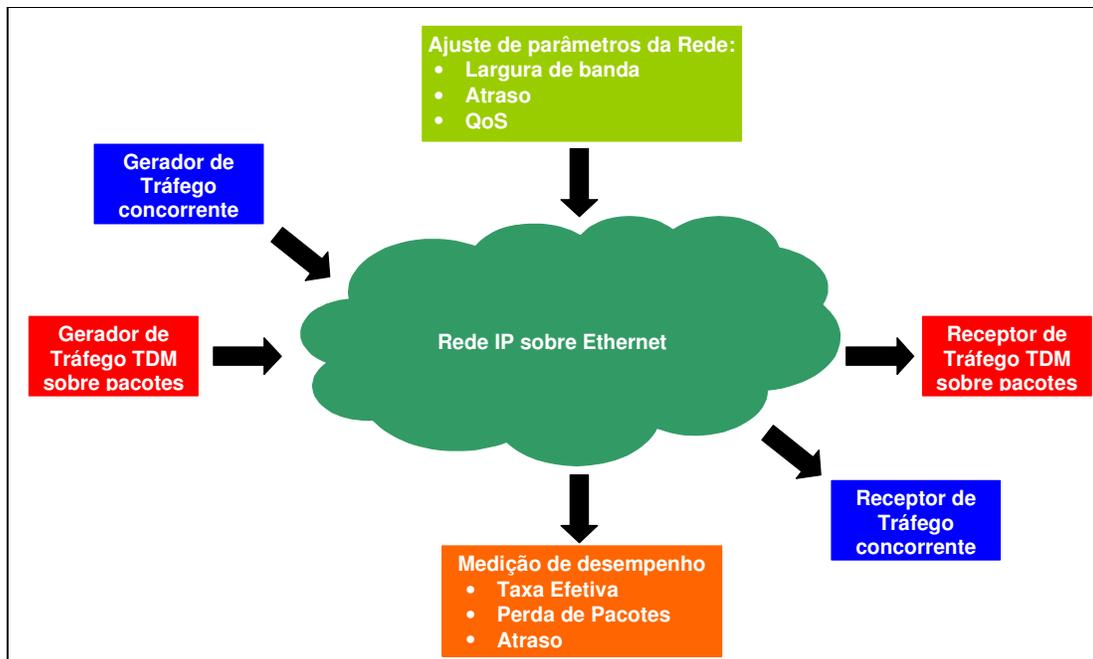


Figura 4.19 – Modelo conceitual do ambiente experimental.

Nesse modelo conceitual, é utilizada uma rede IP, construída sobre *Ethernet*, na qual seja possível, de alguma forma, controlar os parâmetros associados ao transporte dos pacotes, como a largura de banda oferecida, o atraso sofrido e, eventualmente, mecanismos de QoS. Além disso, devem existir formas de monitorar o desempenho da rede, através da medição da taxa efetiva oferecida aos fluxos, da contagem da perda de pacotes e da determinação dos parâmetros de atraso, como sua média e variação (PDV).

Para a geração do fluxo TDM sobre IP de interesse, é utilizado um gerador e um transmissor de pacotes, construídos de acordo com as especificações PWE3, de forma a caracterizar uma aplicação real de transporte. Finalmente, deve haver condições de perturbar esses fluxos TDM sobre IP, através da geração e recepção, controlável, de tráfego concorrente para competir com os mesmos pelos recursos limitados da rede IP, como a banda disponível e as filas nos roteadores, gerando situações de congestionamento.

#### 4.2.1 – Definição da topologia e configuração da rede LabCom

O próximo passo para transformar esse modelo conceitual em realidade, criando uma plataforma de testes para a tecnologia PWE3, foi construir esse ambiente experimental a partir do ambiente de teste e medição para redes convergentes criado no LabCom/UnB



A essas duas redes foram conectados diversos equipamentos, utilizados para gerar e receber os fluxos de pacotes contendo dados TDM e o tráfego concorrente, como microcomputadores, notebooks e o par de *gateways* IPmux-11, fornecidos pela RAD do Brasil como demonstração para a realização do trabalho.

Na geração do tráfego TDM nativo para esses *gateways* IPmux-11, foi utilizada a central Trópico-RA, de fabricação Alcatel, instalada no LabCom , através de um feixe E1 da rota de entroncamento com a central Sistema-12 também existente na rede LabCom.

Para a geração do tráfego concorrente, foi utilizada a ferramenta “Analisador”, desenvolvida no LabCom/UnB para análise de redes IP [BEZERRA2004], que apresenta facilidades para a configuração e agendamento de diversos tipos de tráfego. A tela principal dessa ferramenta, com alguns tráfegos configurados, é apresentada na Figura 4.21.

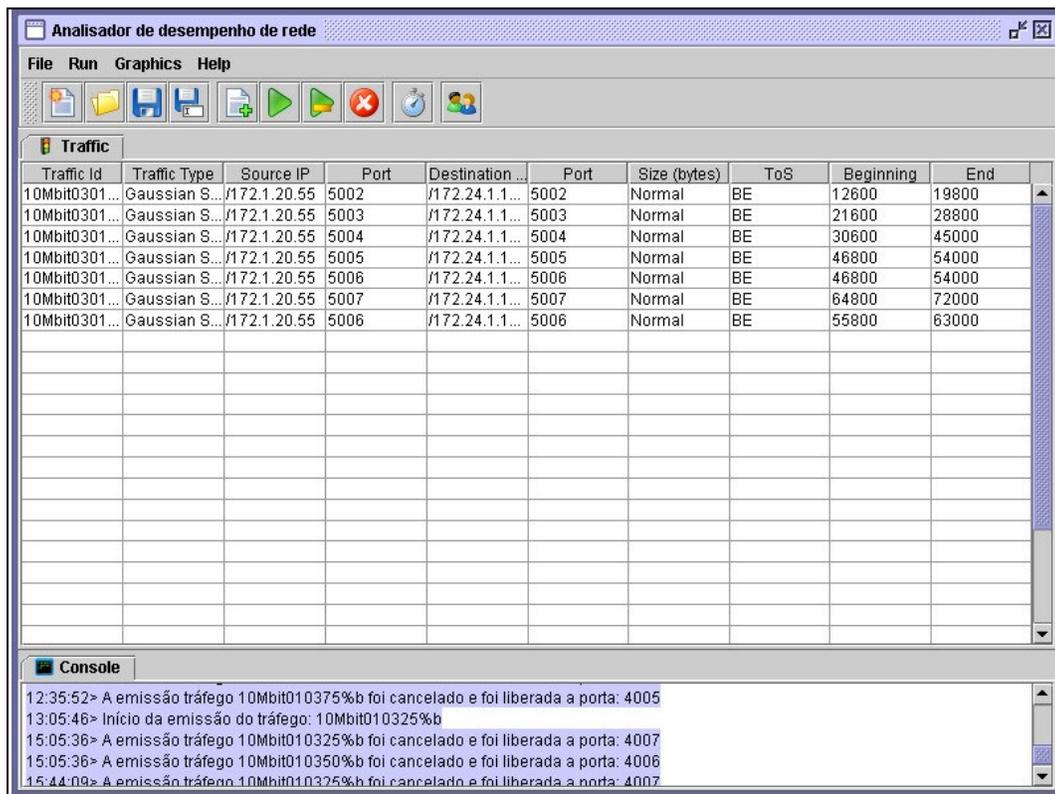


Figura 4.21 – Tela principal da ferramenta “Analisador”, do LabCom/UnB.

Finalmente, para monitoração do desempenho da rede, foi utilizada a ferramenta *tcpdump*, disponível nos roteadores Linux, para captura dos pacotes de cada um dos fluxos e medição dos respectivos instantes de recebimento, além da contagem de pacotes perdidos diretamente pelas aplicações nos microcomputadores e nos *gateways* IPmux-11 .

#### 4.2.2 – Estudo e configuração dos equipamentos RAD IPmux-11

Assim que os *gateways* IPmux-11 enviados pela RAD do Brasil chegaram ao LabCom, foi realizado um estudo de seus manuais, a fim de entender seus procedimentos de instalação e configuração. Um dos gateways utilizados é mostrado na Figura 4.22.



Figura 4.22 - Gateway IPmux-11 da RAD utilizado nos testes.

A instalação dos mesmos foi simples, pois podem ser alimentados em 220V CA, consistindo exclusivamente na montagem dos conectores utilizados nos cabos coaxiais instalados na central, ativada semanas antes, no padrão G.703. Esses cabos são conectados ao *gateway* através de um adaptador, fornecido com o mesmo, que converte os dois conectores coaxiais, de transmissão e recepção, num único conector RJ-45, como utilizado nos cabos UTP. A conexão do respectivo cabo *ethernet* concluiu o processo de instalação, sendo que o IPmux-11 possui uma *bridge* interna com duas portas *ethernet* adicionais,

utilizadas para conectar os microcomputadores na rede 172.1.20.0, evitando a necessidade de outros equipamentos.

A configuração também foi bastante simples, pois os IPmux-11 possuem uma interface serial RS-232C, para conexão a terminais VT100, através da qual é feito o acesso inicial à sua configuração, conforme tela apresentada na Figura 4.23.

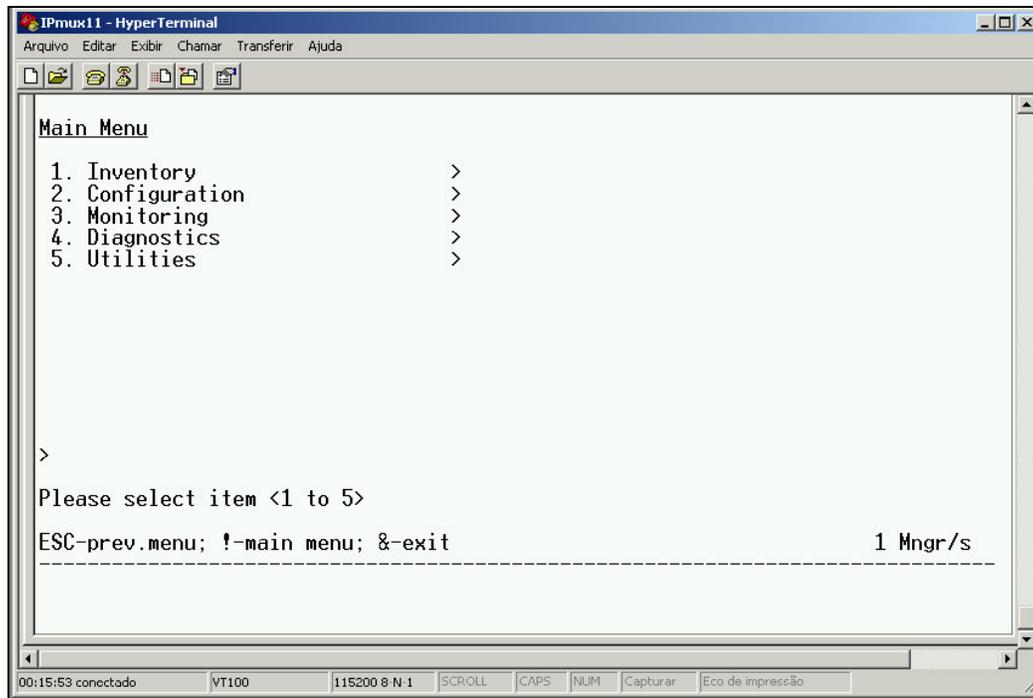


Figura 4.23 – Tela de configuração básica do gateway IPmux-11 utilizado nos testes.

A configuração inicial é realizada através da seqüência *1.System* e *1.Host IP*. Uma vez configurados seus endereços IP, 172.24.1.100 e 172.1.20.100 respectivamente, as máscaras de subrede e os respectivos roteadores, os gateways podem ser acessados através de TELNET, via terminal, ou HTTP, via qualquer navegador, com acesso integral a todas as suas facilidades. Assim, a configuração básica foi concluída com a atualização dos relógios internos, que foram sincronizados manualmente no mesmo segundo através da seqüência *1.System* e *4.Date/Time*.

A próxima etapa foi a configuração física do fluxo TDM em cada gateway, utilizando a seqüência *2.Configuration*, *3.Physical Layer* e *1.TDM configuration*, sendo definido:

*Configuration>Physical layer>TDM configuration (E1)*

*Channel ID (1)*  
*Restoration time (CCITT)*  
*Signaling mode (CAS disabled)*

*1.Admin status: (Enable)*  
*2.Transmit clock source (Adaptive) ou (Loopback)*  
*3.Rx sensitivity (Short haul)*  
*4.Line type (Framed G.704 CRC)*  
*5.Idle code[0 - ff] (7E)*  
*6.Send upon fail (AIS)*  
*7.OOS code[0 - ff] (FF)*

Como tipo de linha foi definido o sinal TDM utilizado pela rota da Trópico-RA, G.704 com uso de CRC, estruturado com canalização e CRC: (*Framed G.704 CRC*).

Em relação ao relógio de transmissão, o *gateway* localizado na rede 172.1.20.0, fisicamente mais próximo da central e recebendo dados da rota de saída, foi configurado para recuperação do relógio a partir do fluxo TDM recebido da central (*Loopback*), enquanto o *gateway* localizado na rede 172.24.1.0 foi configurado para recuperação de relógio adaptativa (*Adaptive*) utilizando o fluxo de pacotes recebido, estabelecendo o esquema indicado na Figura 2.12c, apresentada na seção 2.3.1.

Na seqüência, foi realizada a configuração da conexão *ethernet*, utilizando a seqüência *2.Configuration*, *3.Physical Layer* e *2.Eth configuration*, sendo definida a velocidade de 10Mbps, *full-duplex*:

*Configuration>Physical layer>Eth configuration*

*Channel (Network-Eth1)*

*1. Channel state (Enable)*  
*2. Auto negotiation (Enable)*  
*3. Max capability advertised (10baseT full duplex)*  
*4. Default type (10baseT half duplex)*

Finalmente, foi realizada a configuração da conexão TDMoIP<sup>®</sup> propriamente dita, utilizando a seqüência *2.Configuration*, *3.Connection* e *2.DS0Bundle*, para definir os canais a transmitir; e *2.Configuration*, *3.Connection* e *3.Bundle connection*, para definir os parâmetros de emulação do enlace, conforme estabelecido na Tabela 4.4:

Tabela 4.4 – Configuração de parâmetros para os gateways IPmux-11.

Parâmetro	IPmux-11 (local)	IPmux-11 (remoto)
Endereço IP	172.24.1.100	172.1.20.100
Sinalização	G.704 CRC	G.704 CRC
Relógio	adaptativo	loopback
Adaptação dos dados	AAL1	AAL1
Número de canais	31	31
Tamanho do <i>buffer</i>	50ms	50ms

*Configuration>Connection>Bundle connection*

1. Destination IP address (172.1.20.100) ou (172.24.1.100)
2. Next hop (172.24.1.111) ou (172.1.20.1)
3. IP TOS[0 - 255] (0)
4. Connection status (Enable)
5. Destination bundle[1 - 8063] (1)
6. TDM bytes in frame(x48 bytes)[1 - 30] (5)
7. Payload format (V1)
8. OAM connectivity (Disable)
9. Jitter buffer [msec][3 - 300] (50)
10. VLAN tagging (Disable)

Assim, foi concluído o processo de configuração, ativando a conexão TDMoIP<sup>®</sup> assim que os parâmetros foram salvos. Dessa forma, foi criado um pseudo-circuito TDM, interligando com sucesso as rotas de entrada (1024) e saída (0) da central Trópico-RA através da rede IP do LabCom, que passou a ser monitorado, em relação aos sinais TDM, através da seqüência *3.Monitoring, 1.Statistics e 1.TDM physical layer*, apresentando:

*Monitoring>Statistics>TDM physical layer (E1)*

<i>Channel ID</i>	(1)		
<i>LOS:</i>	(0)	<i>DM:</i>	(0)
<i>LOF (Red):</i>	(0)	<i>ES:</i>	(0)
<i>LCV:</i>	(0)	<i>SES:</i>	(0)
<i>RAI (Yellow):</i>	(0)	<i>UAS:</i>	(0)
<i>AIS:</i>	(0)	<i>LOMF:</i>	(0)
<i>FEBE:</i>	(0)		
<i>BES:</i>	(0)		

*Time start:* (21:30:00 2006-09-09)  
*1. Interval* ... (1)

Em relação à conexão TDMoIP<sup>®</sup>, a monitoração é feita através da seqüência *3.Monitoring, 1.Statistics e 2.Connection*, apresentando:

*Monitoring>Statistics>Connection*

```

Sequence errors:                (0)
Jitter buffer underflows:      (0)
Jitter buffer overflows:      (0)
Max Jitter buffer deviation [msec]: (4)
Time since [sec]:              (487)
1. Bundle ID[1 - 1]           (1)
2. Interval                    (0)

```

Em ambos os casos, a monitoração é feita dentro de uma “janela” (*Interval*) de 15 minutos (900s), por até 24h, totalizando 96 janelas que ficam armazenadas no *gateway*.

Para efeito de monitoramento, existe ainda mais uma opção, através da seqüência Em relação à conexão TDMoIP<sup>®</sup>, a monitoração é feita através da seqüência *3.Monitoring, 2.Status e 2.Connection*, que permite a verificação do estado da conexão, apresentando:

*Monitoring>Status>Connection*

```

Destination IP address:  (172.1.20.100)
Next hop MAC address:   (00E07DF5A8EE)
Connectivity status:    > (OK      )
Sequence errors:        (0)
Jitter buffer underflows: (0)
Jitter buffer overflows: (0)
1. Bundle ID[1 - 1]     (1)

```

E, finalmente, o arquivo de registro de eventos (*event log*), acessado através da seqüência *3.Monitoring, 3.Event log e 2.Read log file*, conforme mostrado na Tabela 4.5.

Tabela 4.5 – Registro de eventos do IPmux-11, acesso HTTP.

Index	Log entry
56	2006-09-09 19:52:15 LOGIN VIA TELNET
55	2006-09-08 20:38:03 SN ERRORS TDM SLOT BUNDLE 1 1 EVENTS
54	2006-09-08 20:38:02 SN ERRORS TDM SLOT BUNDLE 1 5 EVENTS
53	2006-09-08 20:37:57 SN ERRORS TDM SLOT BUNDLE 1 7 EVENTS
52	2006-09-08 20:37:23 SN ERRORS TDM SLOT BUNDLE 1 9 EVENTS

Em seguida, foram feitos testes de conversação através dos dois telefones analógicos instalados na central, não sendo percebida qualquer diferença de qualidade entre as chamadas transportadas pelo enlace TDM nativo, com conexão direta do juntor de saída ao juntor de entrada pelo cabo coaxial; e as transportadas através do enlace TDMoIP®.

#### **4.2.3 – Estudo e configuração da central Trópico-RA**

A ativação da central Trópico-RA do LabCom/UnB foi a fase mais difícil da montagem do ambiente experimental, pois a mesma não era utilizada desde 2004 e apresentava problemas no sistema configurado de acesso TELNET através do concentrador Cyclades. Assim, foi necessário o acesso direto via interface RS-232C ao processador 2, de O&M.

Resolvido o problema de acesso, a central também não respondeu às rotinas normais de inicialização, apresentando vários processadores que não carregavam, os quais tiveram que ser levantados individualmente, obrigando a um estudo relativamente profundo da documentação disponível [PROMOM1999] [PROMOM1999a] [PROMOM1999b] [PROMOM1999c] [ALCATEL2002] [ALCATEL2002a] [ITS2002] [ITS2002a]. No caso do processador 21, responsável pelo gerenciamento do módulo de assinantes, houve necessidade de substituição da placa processadora por uma sobressalente, e alguns módulos não estavam operacionais, como a sinalização SS7, mas foi possível utilizar os assinantes e os jutores de entrada e saída, condição suficiente para os objetivos do trabalho.

Como problema adicional, o sistema de baterias mostrou-se inoperante, fazendo com que a central perdesse a alimentação quando das quedas de energia no prédio SG-11, obrigando a sua reativação, felizmente agora bem mais simples, por diversas vezes ao longo dos cerca de 45 dias de teste.

Nessa central Trópico-RA, foram instalados telefones em dois assinantes analógicos que já estavam previamente criados, utilizando os NEAs 00210000 e 00210114, ou seja, processador 21, placa 0, posição 0, para o assinante 5582-1000; e processador 21, placa 1, posição 14, para o assinante 5582-1030.

A Figura 4.24 mostra a central Trópico-RA do LabCom/UnB, ativada, com os três bastidores abertos.



Figura 4.24 – Central Trópico-RA utilizada nos testes para geração de feixe E1.

Após realizar diversos testes com esses assinantes, utilizando diretamente a comutação local, foi configurado um plano de encaminhamento alternativo, usando o plano #1 (Reserva) armazenado, que direcionava para a rota de saída 0, localizada fisicamente no juntor 00080000, processador 8, placa 0, enlace 0; as chamadas cuja marcação fosse iniciada com o dígito “1”. Da mesma forma, foi reconfigurada a rota de entrada 1024, localizada fisicamente no juntor 00070000, processador 7, placa 0, enlace 0, de forma que os dígitos recebidos através da mesma fossem considerados como rota rápida para os terminais da própria central.

Assim, através dos cabos coaxiais, foi ligado o sinal Tx do enlace E1 do juntor 00080000 no sinal Rx do enlace E1 do juntor 00070000, e vice-versa, de modo que as rotas 0 e 1024 fossem externamente interligadas, não mais através da central Sistema-12, mas diretamente. Os demais enlaces dos mesmos jutores permaneceram como estavam, em *loopback*.

Dessa forma, quando do terminal 5582-1000 era teclado 11030, a chamada era encaminhada à rota 0 de saída, sendo recebida pela rota 1024 de entrada e reencaminhada ao assinante 5582-1030, fazendo com que a conversação fosse conduzida através do enlace E1. Quando uma chamada para o destino 11000 era originada do terminal 5582-1030, ocorria exatamente o inverso.

Mais tarde, quando foram ativados os *gateways* IPmux-11, os cabos Tx e Rx do juntor 00080000 foram conectados ao *gateway* remoto e os cabos do juntor 0007000 ao *gateway* local, fazendo com que o entroncamento fosse realizado através do núcleo MPLS, utilizando a emulação do circuito TDM E1 através de TDMoIP®.

#### **4.2.4 – Ensaios de desempenho TDMoIP®**

Concluída a fase de configuração dos terminais, assegurando que os fluxos TDM gerados pela central eram recebidos pelos *gateways* IPmux-11, encapsulados em pacotes e encaminhados através das respectivas interfaces *Ethernet* aos roteadores de borda (LER) do núcleo, foi realizada a configuração desse núcleo, de forma a permitir a realização dos ensaios de desempenho.

Como o objetivo desse trabalho é a utilização de PWE3 em PSN onde existe garantia de banda, mas não QoS, as disciplinas de fila para implementação de MPLS não foram utilizadas; contudo, a fim de aumentar o número de saltos dentro do caminho percorrido pelos pacotes dentro do núcleo, aumentando sua sensibilidade aos parâmetros da rede, foram fixados limites muito estreitos para o tráfego em rajadas no *token bucket* configurado para as interfaces eth0 do LSR02 e eth2 do LSR04, que assim provocaram, pela atuação do OSPF, mudanças nas tabelas de roteamento, modificando a rota original entre as redes 172.24.1.0 e 172.1.20.0 através do núcleo, como mostrado na Figura 4.25:

- Roteamento Original OSPF:  
LER03->LSR02->LER01->172.24.1.100 (3 saltos);
- Novo Roteamento OSPF:  
LER03->LSR02->LSR04->LER01->172.24.1.100 (4 saltos).



progressivo dos enlaces, até a ocupação total da banda disponível. Para alguns ensaios, foi feito também um descarregamento progressivo, obedecendo à regra inversa.

Tabela 4.6 – Regime de geração do tráfego concorrente.

Distribuição do Tamanho dos Pacotes	Normal	Normal	Normal	Normal
- Média (octetos)	256	512	768	1024
- Variância	655,36	2.621,44	5.898,24	10.485,76
Tamanho Mínimo (95%)	179,2	358,4	537,6	716,8
Tamanho Máximo (95%)	332,8	665,6	998,4	1.331,2
Taxa na PSN (bps)	1.024.000	2.048.000	3.072.000	4.096.000
Ocupação do Enlace 4Mbps	25%	50%	75%	100%

Definidas todas as características para a rede, o fluxo TDM e o tráfego concorrente, foram realizados algumas medições, buscando determinar a estabilidade do pseudo-circuito e a sua sensibilidade à ocupação da rede, bem como o efeito da quantidade de PDUs AAL1 utilizadas em cada pacote, ou seja, do tamanho do pacote contendo dados TDM, nesse desempenho, conforme abordado em [PILATI2004], a fim de tecer algumas comparações.

a) Ensaio de desempenho em 24h, considerando a largura de banda variável:

Nesse ensaio, foram configurados ciclos de 2 horas para cada uma das condições de tráfego concorrente, carregando e descarregando a rede, com intervalo de 30 minutos entre cada uma delas, a fim de assegurar que houvesse pelo menos uma “janela” de medição de 15 minutos, pelos IPmux-11, sem a presença desse tráfego. O tamanho dos pacotes foi fixado em 5 PDUs (240 octetos AAL1) e as medições foram realizadas para taxas nominais nos enlaces de 4Mbps, 6Mbps, 8Mbps e 10Mbps.

Em cada cenário, foram monitoradas as perdas de pacotes, a quantidade de *underflows* e *overflows* e ocupação máxima do *buffer* dentro de cada janela de 15 minutos.

As Figuras 4.26, 4.27, 4.28 e 4.29 apresentam, respectivamente, o desempenho observado ao longo das 24h de teste para cada uma dessas variáveis, para enlaces de 4Mbps, 6Mbps, 8Mbps e 10Mbps. Nessas figuras é indicado o intervalo de geração do tráfego de perturbação e a qualidade do enlace TDM percebida pela central Trópico-RA.

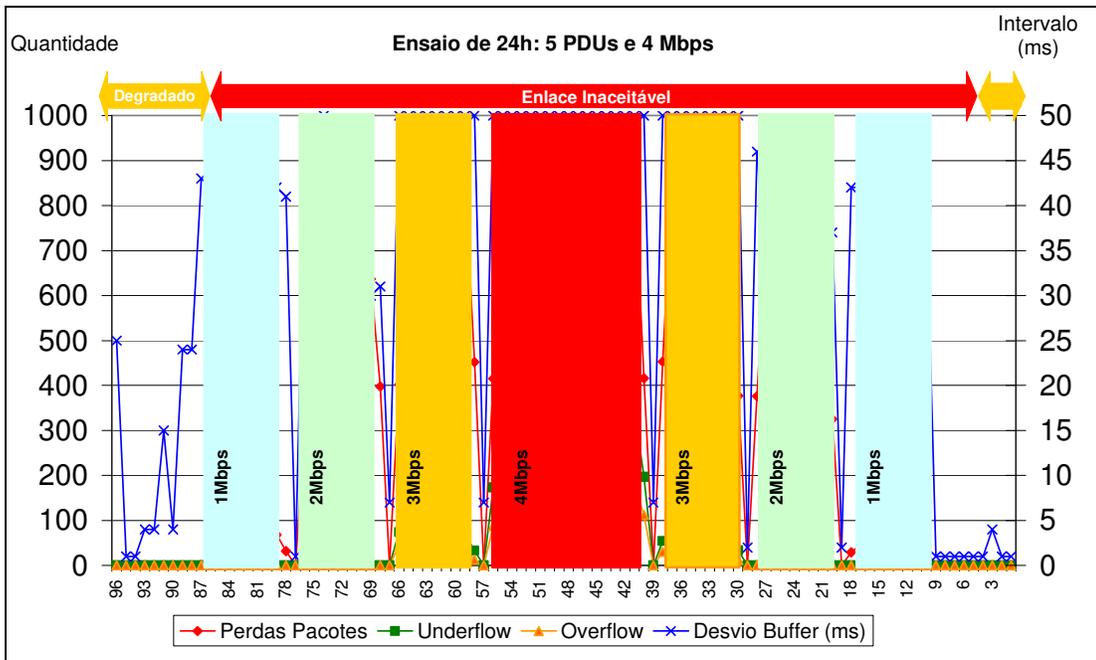


Figura 4.26 – Desempenho de um pseudo-circuito E1 para enlaces de 4Mbps

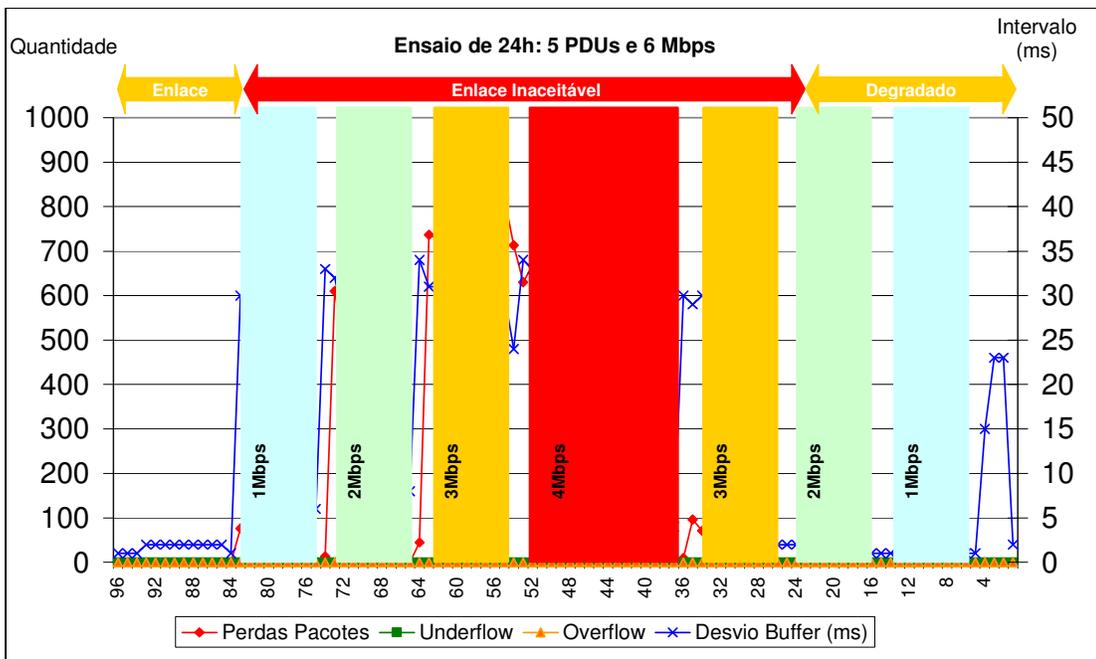


Figura 4.27 – Desempenho de um pseudo-circuito E1 para enlaces de 6Mbps

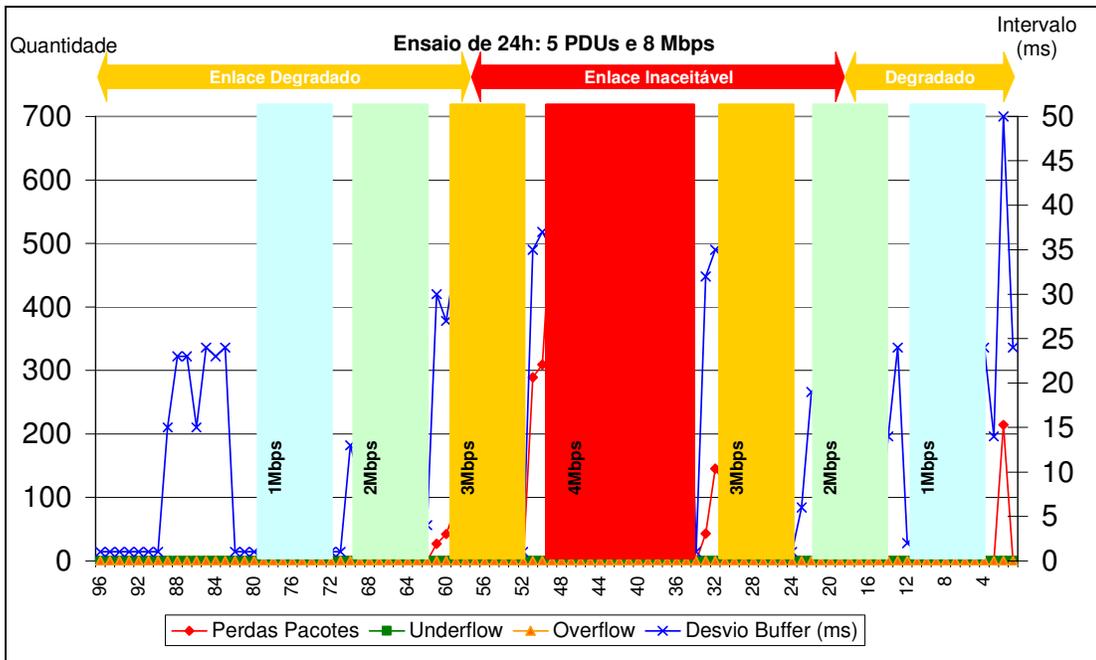


Figura 4.28 – Desempenho de um pseudo-circuito E1 para enlaces de 8Mbps

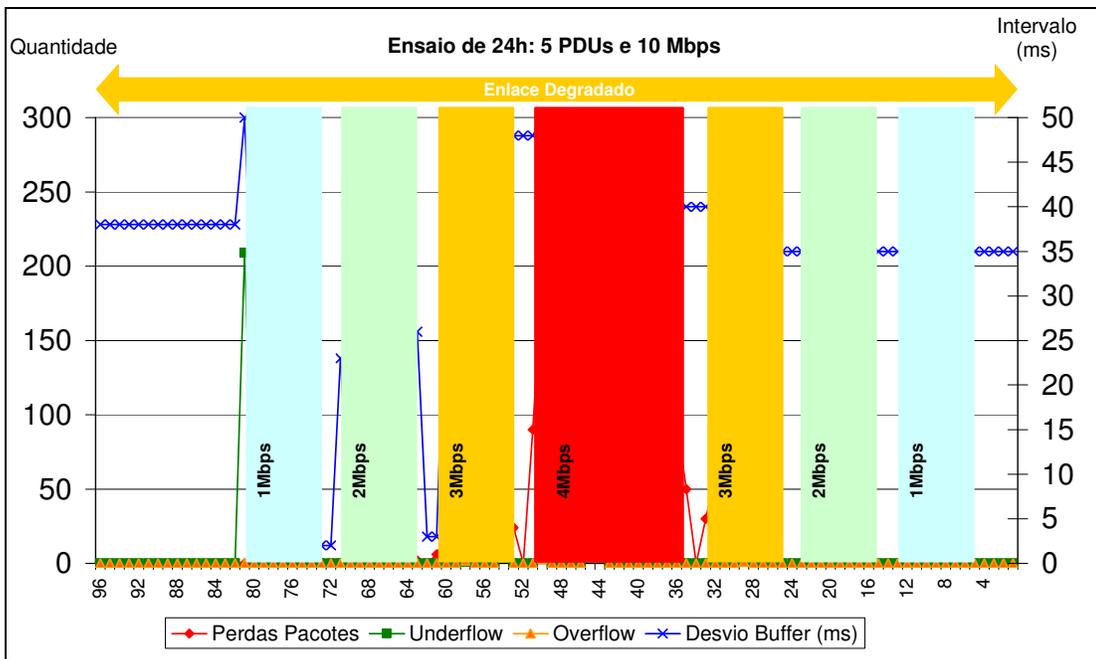


Figura 4.29 – Desempenho de um pseudo-circuito E1 para enlaces de 10Mbps

A qualidade do enlace PCM medida pela Trópico-RA é obtida através da monitoração de seu arquivo de alarmes após a realização dos testes, conforme apresentado na Tabela 4.7.

Tabela 4.7 – Condições de alarme para enlaces PCM na Trópico-RA .

<b>Alarme</b>	<b>Situação</b>	<b>Condição</b>	<b>Prioridade</b>
GSINC_00	Enlace PCM com taxa de escorregamento na condição <b>degradada</b>	> 5 escorregamentos em 24 horas	Não Urgente
GSINC_01	Enlace PCM com taxa de escorregamento na condição <b>inaceitável</b>	>30 escorregamentos em 1 hora	Semi-Urgente

Cabe salientar que as condições de monitoração da central utilizam intervalos demasiado elásticos, se comparados aos intervalos de medição considerados nos ensaios, sendo apenas uma referência durante os testes. Para uma análise efetiva da qualidade dos enlaces PCM nesse tipo de teste seria necessário utilizar um analisador de protocolo, a fim de medir com precisão os escorregamentos e o *jitter* presente no fluxo TDM de saída do *gateway* remoto.

b) Ensaio de desempenho considerando a quantidade de PDUs variável:

Nesse ensaio, foram configurados ciclos de 1 minuto para cada uma das condições de tráfego concorrente, com intervalo de pelo menos 15 minutos entre cada uma delas, a fim de assegurar que os efeitos de cada condição de ocupação fossem restritos a uma única “janela”. A taxa nominal nos enlaces foi fixada em 4Mbps e as medições foram realizadas para tamanhos de pacotes com 1 PDU (48 octetos), 5 PDUs (240 octetos), 15 PDUs (720 octetos) e 30 PDUs (1440 octetos).

Em cada cenário, foram monitoradas as perdas de pacotes, a quantidade de *underflows* e *overflows* e ocupação máxima do *buffer* dentro de cada janela de 15 minutos.

As Figuras 4.30, 4.31, 4.32 e 4.33 apresentam, respectivamente, o desempenho observado para cada uma dessas variáveis, considerando pacotes de 1, 5, 15, e 30 PDUs. Nessas figuras também é indicado o intervalo de geração do tráfego de perturbação e a qualidade do enlace TDM percebida pela central Trópico-RA.

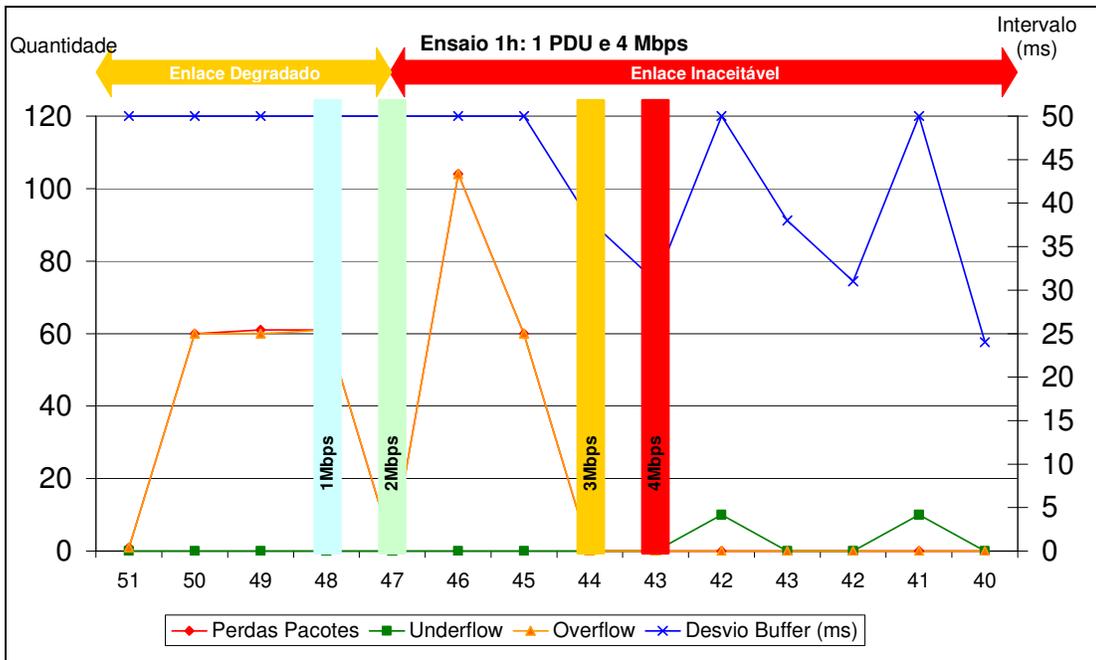


Figura 4.30 – Desempenho de um pseudo-circuito E1 para 48 octetos por pacote.

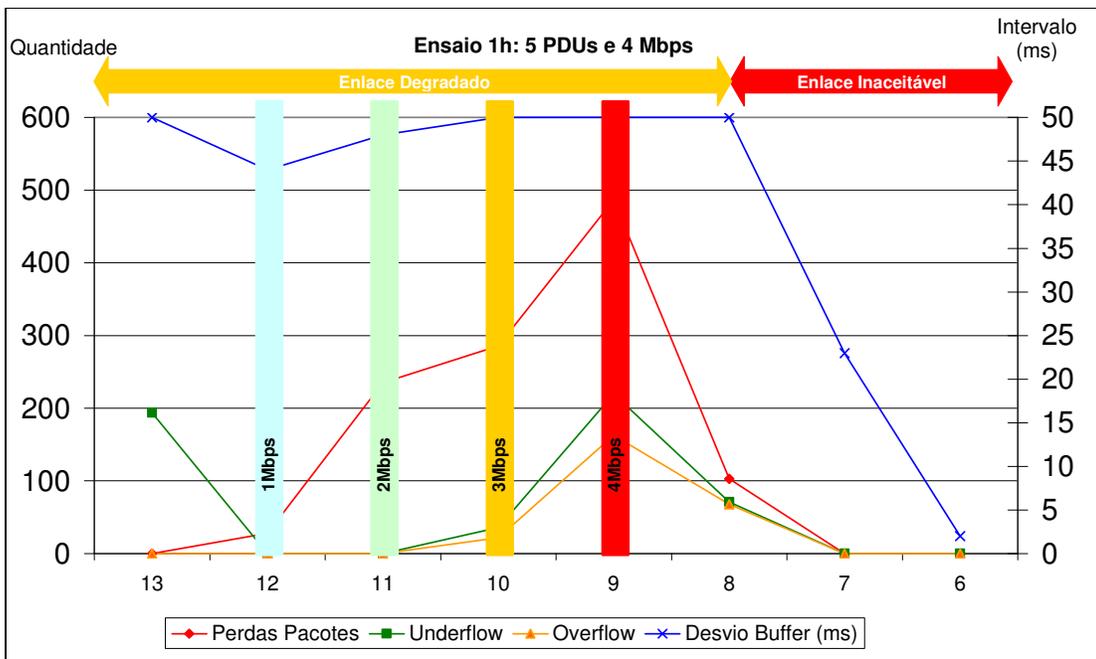


Figura 4.31 – Desempenho de um pseudo-circuito E1 para 240 octetos por pacote.

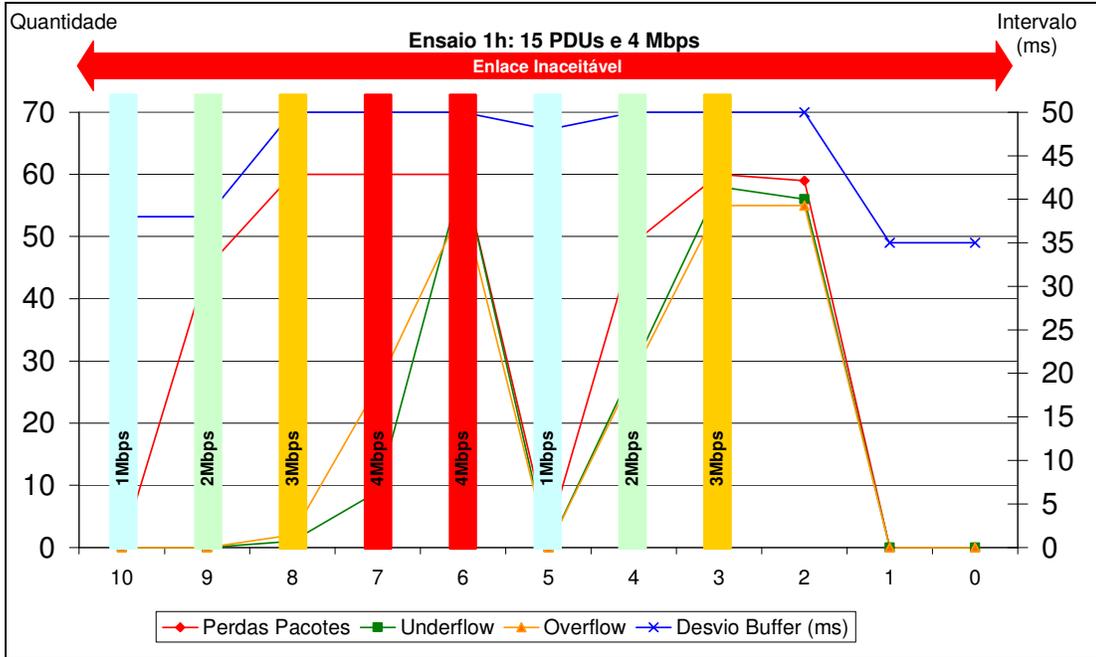


Figura 4.32 – Desempenho de um pseudo-circuito E1 para 720 octetos por pacote.

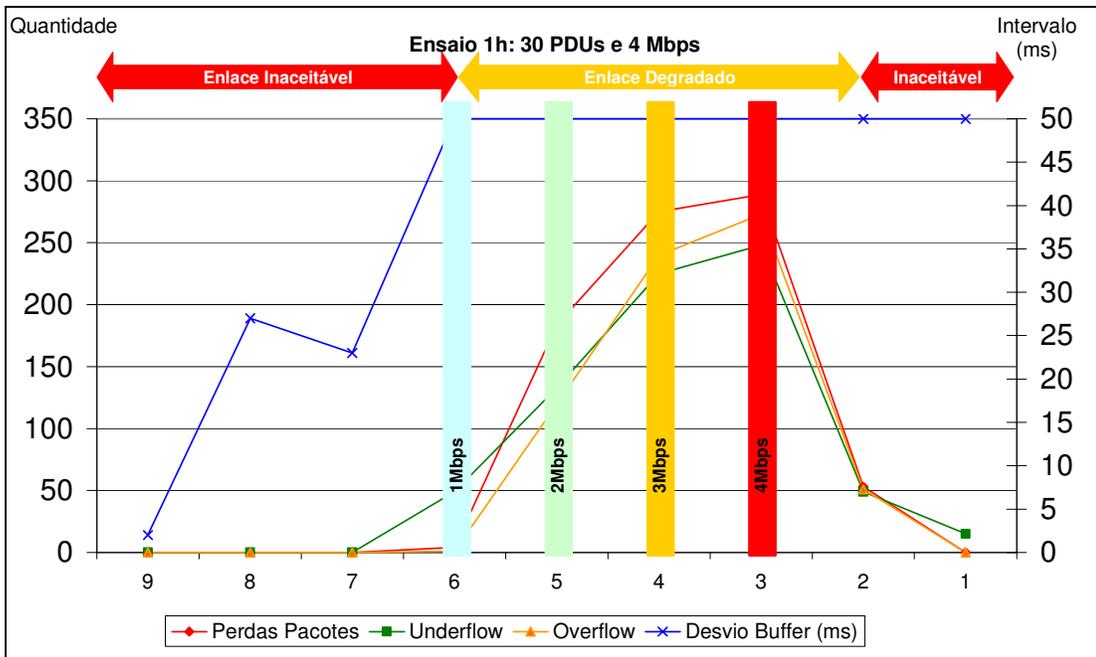


Figura 4.33 – Desempenho de um pseudo-circuito E1 para 1440 octetos por pacote.

Além dos ensaios monitorados, o enlace ficou quase o tempo inteiro ativo durante 45 dias, até que a central fosse desligada em função da elevada temperatura do laboratório, devido à falha nos sistemas de ar condicionado. Nesse período, salvo quando os enlaces foram sobrecarregados pelos testes, o desempenho dos *gateways* IPmux-11 foi excelente, com baixíssima perda de pacotes e sem indicação de escorregamento degradado ou inaceitável pela central Trópico-RA, ficando as falhas limitadas às quedas de energia.

Quando a central foi desativada, os enlaces permaneceram ativos, tendo sido modificada a configuração de relógio do *gateway* remoto, de *loopback* para interno e ligados os terminais Tx e Rx de cada IPmux-11 através de um cabo coaxial. Nessa condição, que foi utilizada nos ensaios com a aplicação em Java, foram registrados 11 dias de funcionamento ininterrupto, não sendo registrada nenhuma perda de pacotes no *gateway* remoto.

### **4.3 - IMPLEMENTAÇÃO TDMOIP<sup>®</sup> EM JAVA**

Com o objetivo de comprovar o desempenho dos algoritmos propostos, e em função da incerteza inicial da possibilidade de testes com equipamentos comerciais para validação da tecnologia, foi iniciado em 2005 um projeto envolvendo alunos de graduação do Curso de Engenharia de Redes da UnB, vinculados ao LEMOM (Laboratório de Estruturas de Microondas e Ondas Milimétricas), com o objetivo de desenvolver uma aplicação TDM sobre IP, a ser utilizada na obtenção dos resultados experimentais desse trabalho. Essa aplicação seria o trabalho de conclusão do curso desses alunos.

#### **4.3.1 – Desenvolvimento das ferramentas de transmissão e recepção**

Dentro das premissas de desenvolvimento da aplicação de emulação TDM sobre IP, foi adotado o nome Mux-IP PC, tendo sido escolhida a linguagem Java em função de outros projetos existentes no LEMOM e do interesse dos alunos no aprendizado da linguagem.

Durante o projeto, foram concebidas duas aplicações. Uma delas é responsável pela geração de pacotes conforme definido pelos *Internet-Drafts*, sendo adotada a abordagem SAToP por simplicidade, uma vez que a proposta de utilização dos pseudo-circuitos para

*broadcasting* IPTV não implica na manipulação isolada dos canais de voz. E a outra é responsável pela recepção desses pacotes, onde foi implementado o novo mecanismo proposto para seqüenciamento. A especificação original dessas duas aplicações é descrita pelo diagrama de blocos funcionais apresentado na Figura 4.34.

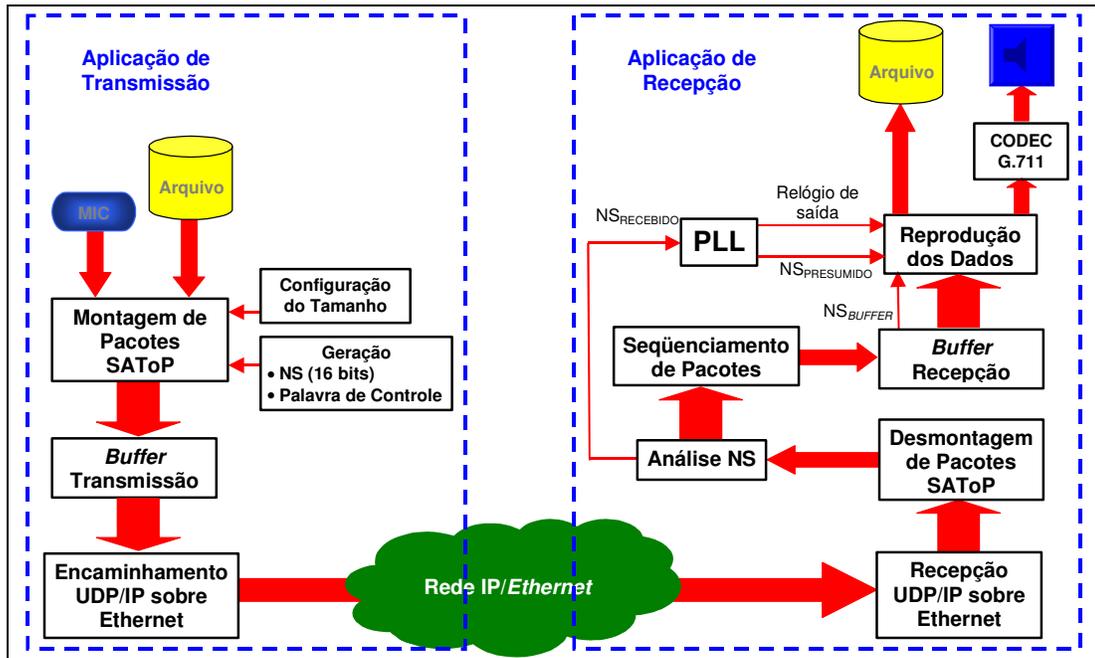


Figura 4.34 – Diagrama de blocos do Mux-IP PC.

Na aplicação de transmissão, a idéia original do projeto era que os pacotes fossem montados a partir de um arquivo de voz ou vídeo, lido pela aplicação utilizando base de tempo fixa, equivalente a um sinal TDM efetivo. No caso de voz, as amostras de 8 bits, já codificadas seguindo a lei A da Recomendação ITU-T G.711, seriam acomodadas dentro de um determinado canal do quadro E1, configurável pela aplicação. No caso de vídeo, a transmissão do arquivo ocuparia todo o fluxo de bits, de forma contínua. Em uma segunda abordagem, seria utilizado o microfone da placa de som instalada no PC para codificação G.711 (lei A) em tempo real do sinal de voz, a fim de que este fosse acomodado em um dos canais. Na seqüência a aplicação faz a montagem dos pacotes, utilizando a abordagem SAToP, permitindo a configuração do seu tamanho entre 1 e 5 quadros E1, ou 1 ms de dados TSM, correspondentes a 256, 512, 768, 1024 ou 1280 octetos; gerando um número de seqüência de 16 bits, com valor inicial configurável, que é acomodado dentro da palavra de controle. Uma vez montados, os pacotes são armazenados num *buffer* de transmissão, aguardando o envio pela interface *ethernet*, que é realizado por uma classe nativa do Java.

Na aplicação de recepção, os pacotes são obtidos da interface *ethernet* e entregues ao processo também por uma classe nativa do Java, que realiza a sua desmontagem, retirando a palavra de controle e dela extraindo o número de seqüência, que é utilizado pelo algoritmo de seqüenciamento para determinar a posição de escrita no buffer circular auto-gerenciável de recepção, assegurando assim a seqüência correta dos dados quando forem retirados pelo processo de leitura do *buffer*. Da mesma forma nesse caso, a idéia original era que o processo de recuperação do relógio a partir do número de seqüência fosse implementado na aplicação, construindo o PLL em blocos de *software*, ou mesmo utilizando algum *hardware* para facilitar o processo de geração da freqüência de relógio para reprodução dos dados de cada pacote no fluxo de saída. Além disso, estava prevista a gravação dos dados de áudio ou vídeo recebidos em arquivo, para que pudessem ser comparados com a informação transmitida, evidenciando a eventual perda de qualidade. No caso dos sinais de voz, estava prevista a utilização do algoritmo PESQ para avaliação da qualidade MOS do sinal recebido, utilizando a implementação realizada em [OBANDO2005]. Finalmente, em uma segunda abordagem, esperava-se poder reproduzir o conteúdo de um determinado canal G.711 diretamente na placa de som do PC, permitindo assim uma experiência de conversação em tempo real, utilizando o circuito EI emulado, da mesma forma que foi realizado para os telefones analógicos utilizando o roteamento da central Trópico-RA.

Contudo, apenas parte dessa especificação original pôde ser implementada, uma vez que a escolha inicial do Java demonstrou-se desastrosa para as necessidades desse tipo de aplicação, onde seriam necessárias bases de tempo precisas e o controle total da máquina, eventualmente com a incorporação de módulos de *hardware* no receptor ou a utilização de interrupções. Foram analisadas, pelos alunos envolvidos no projeto, diversas técnicas para melhorar a temporização no Java, incluindo códigos nativos em C<sup>++</sup>, sem obter sucesso.

A leitura, manipulação e gravação de arquivos chegou a ser implementada em algumas versões, mas tornava o processo tão lento dentro da máquina virtual Java que comprometia totalmente as taxas de transmissão, tendo que ser abandonada.

Entretanto, o desempenho do algoritmo de seqüenciamento implementado na aplicação, que independe de bases de tempo precisas, demonstrou-se muito bom, sendo testada, inclusive, a sua resposta em transmissões através da Internet pelos alunos do projeto. A fim de assegurar, de forma definitiva, a validade desse algoritmo, foram realizados testes provocando desordenamento intencional dos pacotes pelo próprio transmissor, sendo a ordem seqüencial correta re-estabelecida perfeitamente na saída. A Figura 4.35 apresenta as duas aplicações independentes do MUX-IP PC desenvolvido.

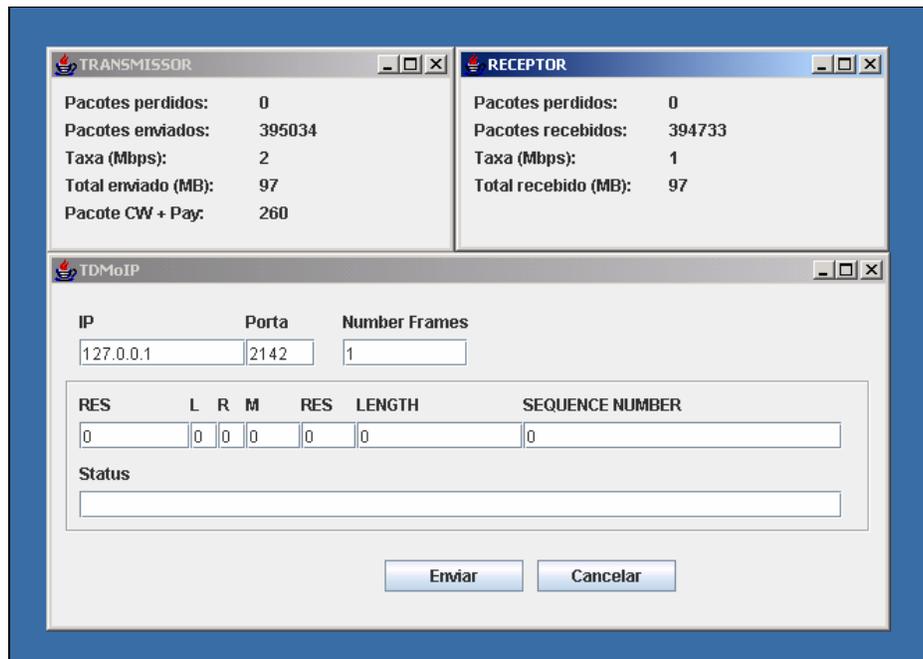


Figura 4.35 – Telas das aplicações Mux-IP PC.

Em função dessas dificuldades com o Java, que implicariam em refazer o desenvolvimento utilizando outra linguagem que permitisse o domínio mais preciso das bases de tempo, como o C++, o projeto inicial foi abandonado, mantendo-se a aplicação apenas com as funcionalidades básicas, o que limitou a realização dos ensaios de desempenho.

#### 4.3.2 – Ensaios de desempenho TDMoIP®

Como a aplicação MUX-IP PC desenvolvida consegue implementar apenas o novo mecanismo de seqüenciamento, foram realizados ensaios para comparação direta do desempenho das duas aplicações, fazendo com que ambos os fluxos trafegassem ao mesmo

tempo através do núcleo, a fim de determinar qual delas apresenta melhor desempenho na competição pelos recursos da rede com o tráfego concorrente e entre si, conforme mostrado na Figura 4.36, onde em vermelho com linha pontilhada é indicado o fluxo 2Mbps entre os IPmux-11, em vermelho com linha contínua é indicado o tráfego entre as aplicações Mux-IP PC e em azul o tráfego concorrente gerado pelo Analisador.

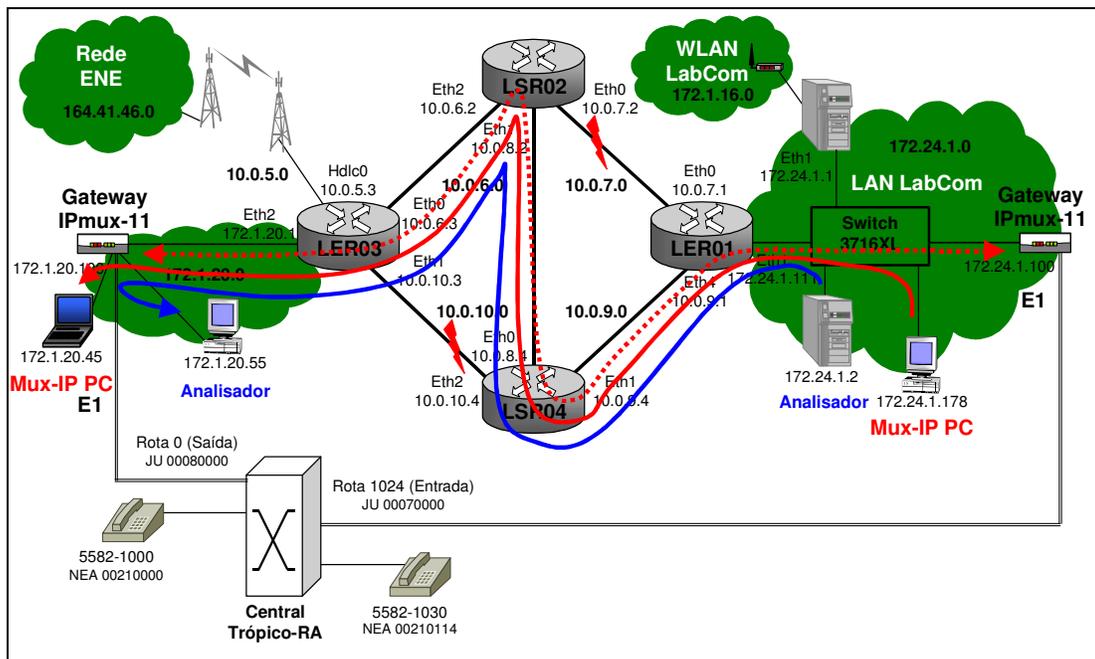


Figura 4.36 – Ensaio comparativo entre IPmux-11 e Mux-IP PC.

Como a premissa da emulação de circuitos TDM sobre as redes de pacotes é a garantia da banda, foi definida a taxa nominal nos enlaces de 6Mbps, assegurando 2Mbps de banda para cada uma das aplicações, restando mais 2Mbps para o tráfego concorrente.

Para o ensaio, foram configurados ciclos de 1 minuto para cada uma das condições de tráfego concorrente, com intervalo de pelo menos 15 minutos entre cada uma delas, a fim de assegurar que os efeitos de cada condição de ocupação fossem restritos a uma única “janela” do IPmux-11. O tamanho dos pacotes para o Mux-IP PC foi fixado em 1ms (256 octetos), sendo monitoradas apenas as perdas de pacotes, enquanto para os gateways foram configuradas 30 PDUs (1440 octetos AAL1), e em seguida 5 PDUs (240 octetos AAL1), sendo monitoradas as perdas de pacotes, a quantidade de *underflows* e *overflows* e ocupação máxima do *buffer* dentro de cada janela de 15 minutos.

O padrão de variação do tráfego concorrente foi o apresentado na tabela 4.6. O fluxo entre os IPmux-11 demandava 2,077 Mbps (176 pacotes/s) para 30 PDUs e 2,308 Mbps (1.058 pacotes/s) para 5 PDUs; e o Mux-IP PC demandava 2,304 Mbps (1.000 pacotes/s). As Figuras 4.37 e 4.38 apresentam, respectivamente, o desempenho de cada um.

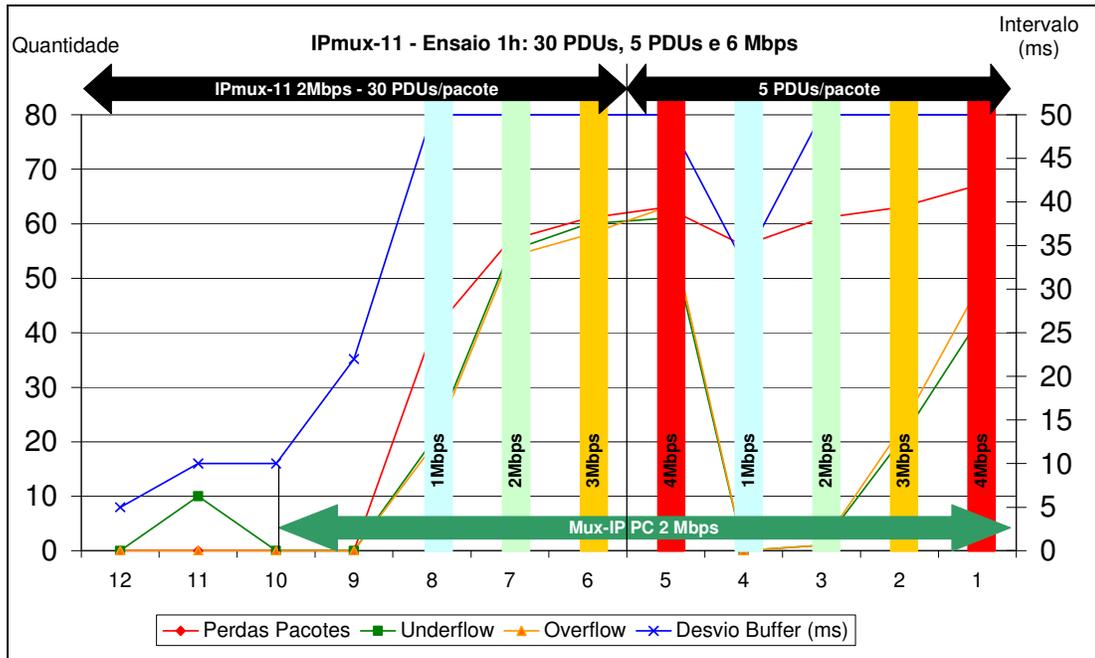


Figura 4.37 – Desempenho IPmux-11 no ensaio comparativo para 6Mbps.

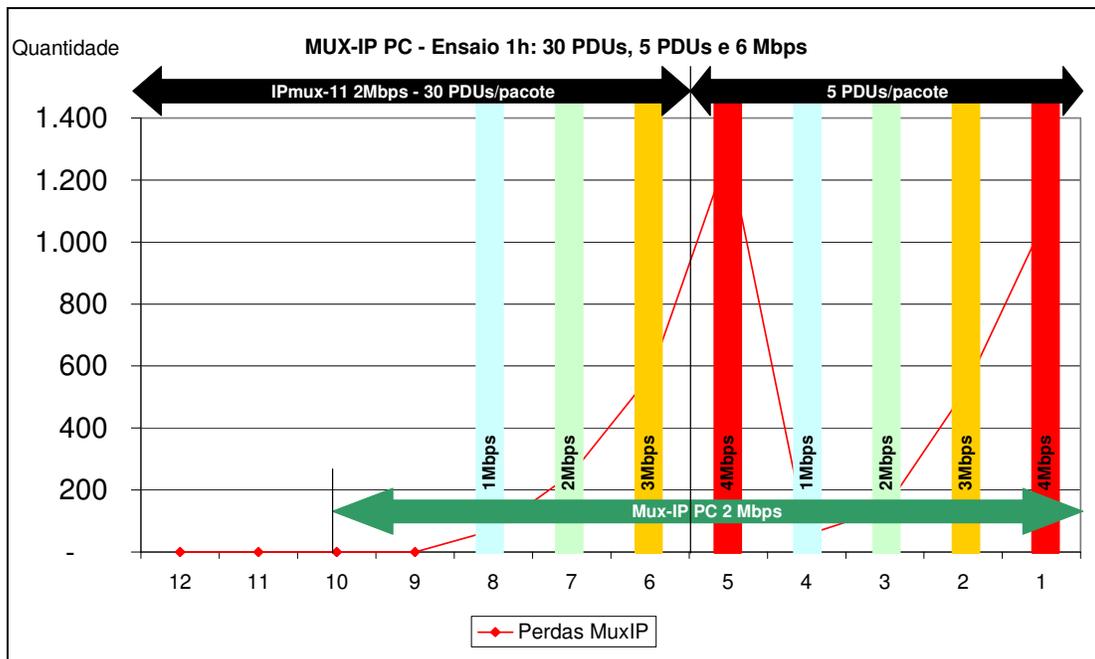


Figura 4.38 – Desempenho Mux-IP PC no ensaio comparativo para 6Mbps.

## 4.4 -ANÁLISE DOS RESULTADOS EXPERIMENTAIS

### 4.4.1 – Desempenho do algoritmo de seqüenciamento de pacotes

O desempenho comparativo entre o novo algoritmo de seqüenciamento proposto neste trabalho e aquele originalmente apresentado no *Internet-Draft* [IETF2006], considerando pacotes efetivamente perdidos e mitigados, de acordo com as condições de simulação descritas na seção 4.1.2 e sem perda induzida de pacotes, é apresentado na Tabela 4.8, para as quantidades de pacotes e probabilidades de recebimento fora de ordem indicadas, sendo a melhoria de desempenho comparada entre o novo algoritmo e o IETF com R=40.

Tabela 4.8 – Comparação perdas efetivas de pacote nos dois algoritmos simulados.

Simulação	400 pacotes				1600 pacotes			
	1%	5%	10%	20%	1%	5%	10%	20%
<b>Probabilidade de Pacotes Desordenados</b>								
<b>Algoritmo IETF R=0</b>	6 pacotes 1,5%	20 pacotes 5,0%	40 pacotes 10,0%	87 pacotes 21,8%	24 pacotes 1,5%	80 pacotes 5,0%	163 pacotes 10,2%	320 pacotes 20,0%
<b>Algoritmo IETF R=10</b>	5 pacotes 1,3%	16 pacotes 4,0%	34 pacotes 8,5%	58 pacotes 14,5%	18 pacotes 1,1%	74 pacotes 4,6%	149 pacotes 9,3%	281 pacotes 17,6%
<b>Algoritmo IETF R=20</b>	3 pacotes 0,8%	9 pacotes 2,3%	25 pacotes 6,3%	54 pacotes 13,5%	6 pacotes 0,4%	67 pacotes 4,2%	119 pacotes 7,4%	253 pacotes 15,8%
<b>Algoritmo IETF R=40</b>	2 pacotes 0,5%	7 pacotes 1,8%	19 pacotes 4,8%	42 pacotes 10,5%	2 pacotes 0,1%	41 pacotes 2,6%	74 pacotes 4,6%	153 pacotes 9,6%
<b>Novo Algoritmo Proposto</b>	1 pacote 0,3%	5 pacotes 1,3%	13 pacotes 3,3%	28 pacotes 7,0%	1 pacote 0,1%	30 pacotes 1,9%	64 pacotes 4,0%	134 pacotes 8,4%
<b>Melhoria de Desempenho</b>	<b>-0,3%</b>	<b>-0,5%</b>	<b>-1,5%</b>	<b>-3,5%</b>	<b>-0,1%</b>	<b>-0,7%</b>	<b>-0,6%</b>	<b>-1,2%</b>

A análise desses dados demonstra que o novo algoritmo de seqüenciamento proposto, além de ser conceitualmente mais simples, pois não necessita de processos especiais para gerenciamento do *buffer* e monitoração dos *flags* de esgotamento e esvaziamento; demonstra desempenho bem superior ao apresentado pelo algoritmo sugerido no *Internet-Draft* para R = 0, 10 e 20, permitindo a recuperação transparente de uma quantidade maior

de pacotes, em todas as situações analisadas, inclusive  $R=40$ . Cabe salientar também que as diferenças entre o novo algoritmo e o sugerido no documento IETF são mais significativas para probabilidades mais elevadas de recebimento fora de ordem, indicando o melhor desempenho exatamente nas piores condições oferecidas pela PSN.

Dessa forma, o algoritmo desenvolvido pode ser considerado uma ótima alternativa de evolução para a tecnologia TDMoIP<sup>®</sup>, permitindo sua utilização em redes PSN com perdas de pacotes mais elevadas e piores desempenhos em relação ao atraso e, sobretudo, à variação desse atraso (PDV), indo ao encontro dos requisitos de utilização dessa tecnologia como alternativa de *broadcasting* para aplicações IPTV.

#### **4.4.2 – Desempenho do PLL para sincronização do receptor**

O desempenho comparativo entre o PLL com filtro de realimentação EWMA e aquele utilizando um controlador PID para realimentação do sinal de controle para o DCO é evidenciado pelo comportamento do erro entre as contagens  $T(n)$  e  $R(n)$ , mostrados, respectivamente nas figuras 4.14 e 4.17, juntamente com o sinal  $u(n)$  na saída do controlador ao longo de toda a simulação, nas Figuras 4.15 e 4.18.

O filtro EWMA apresenta uma característica superamortecida, respondendo lentamente à variação intencional na frequência do transmissor, como pode ser observado pelo sinal de controle na Figura 4.15. Dessa forma, mesmo transcorridos 100ms de simulação, o filtro ainda não conseguiu responder à inversão no sentido original do erro pela sua primeira atuação, tendo continuado a desacelerar o relógio do receptor, chegando a um erro de três contagens, e agora levará um tempo muito maior para voltar à condição de equilíbrio.

Enquanto isso, o PLL, por implementar um sistema subamortecido, respondeu quase instantaneamente ao erro inicial, desacelerando o relógio do receptor em cerca de 1 ms, devido à componente derivativa. Na sequência, a componente integral passou a atuar, impedindo que o relógio desacelerasse demais como no caso anterior, e ficou mantendo o erro oscilando entre zero e uma contagem, ao longo dos 99 ms seguintes, independentemente da PDV associada aos pacotes que foram recebidos nesse período, como pode ser observado na Figura 4.18.

Assim, o desempenho do PLL com o controlador PID demonstra-se muito superior à abordagem com o filtro EWMA, para leves desvios na frequência do transmissor, tendo feito o ajuste preciso e quase instantâneo do desvio intencional provocado.

#### **4.4.3 – Desempenho do IPMux-11 na emulação de um enlace TDM**

a) Desempenho para largura de banda da rede variável:

Analisando o entroncamento TDM realizado através do pseudo-circuito entre os *gateways* IPmux-11, nos ensaios de 24 horas utilizando taxas nominais de 4Mbps, 6Mbps, 8Mbps e 10Mbps nos enlaces do núcleo MPLS, pode ser observado que a qualidade da conexão é bastante sensível à taxa efetivamente disponível na rede para esse fluxo, lembrando que para as 5 PDUs consideradas nos ensaios a taxa efetiva é de 2,3Mbps.

Observando a Figura 4.26, para 4Mbps nos enlaces, a ocupação do *buffer* mantém-se abaixo dos 50ms quando são gerados tráfegos concorrentes de 1Mbps e 2Mbps, apesar de crescer bastante; e a perda de pacotes cresce violentamente quando a concorrência cresce de 1Mbps para 2Mbps, embora não cheguem a ocorrer fenômenos de *overflow* ou *underflow*. Contudo, para os tráfegos concorrentes de 3Mbps e 4Mbps, onde claramente a rede não consegue dar vazão a toda a demanda e congestionada, esses fenômenos aparecem fortemente, com a saturação do *buffer* e a perda sistemática de dados. É ainda interessante observar a simetria do gráfico, indicando que a o enlace consegue se recuperar após o congestionamento, e o fato da condição de escorregamento do enlace PCM já ser considerada inaceitável pela central Trópico-RA mesmo para o tráfego de 1Mbps.

Observando a Figura 4.27, para 6Mbps nos enlaces, verifica-se que já não ocorre a saturação do *buffer* para nenhum tráfego, mantendo-se na faixa dos 30 ms; a perda de pacotes existe para todos os casos e cresce com o congestionamento, mas não ocorrem fenômenos de *overflow* ou *underflow* mesmo quando o tráfego chega a 4Mbps. A simetria do gráfico não é mantida nesse caso, pois a segunda aplicação dos tráfegos de 3Mbps, 2Mbps e 1Mbps não provocam tanta perda de pacotes, provavelmente pela fixação, devido ao tráfego de 4Mbps, do ponto de leitura numa posição avançada, mais próxima de 100% do *buffer*. Isso se reflete na condição de escorregamento do enlace PCM percebida pela central Trópico-RA, que deixa de ser inaceitável ao final do tráfego de 3Mbps.

Observando as Figuras 4.28 e 4.29, para 8Mbps e 10Mbps nos enlaces, respectivamente, verifica-se que o comportamento é bem melhor, também não ocorre a saturação do *buffer* para nenhum tráfego; só existe perda de pacotes para 3Mbps e 4Mbps, não havendo também *overflow* ou *underflow*. O gráfico volta a ter comportamento simétrico e tudo isso aparece na condição de escorregamento do enlace PCM percebida pela central Trópico-RA, que fica inaceitável apenas para os tráfegos de 3Mbps e 4Mbps, para o enlace de 8Mbps, permanecendo apenas degradada para todos os tráfegos com o enlace de 10Mbps.

Assim, é confirmada a premissa de que a existência de banda suficiente para a taxa de transmissão exigida pelo fluxo TDM nativo é condição fundamental para a emulação de circuitos TDM sobre as redes de pacotes, uma vez que os mecanismos de controle podem compensar fatores como atraso e PDV, mas a manutenção de um fluxo constante de bits somente é possível se os pacotes gerados à taxa constante encontrarem capacidade de vazão dentro da rede.

Evidentemente, o desempenho do fluxo de pacotes TDM em relação ao tráfego concorrente gerado seria muito superior se fossem utilizados túneis MPLS no núcleo para sua priorização, como é usual em VoIP, mas isso foge aos objetivos desse trabalho, por envolver custos adicionais na rede de transporte.

b) Desempenho para quantidade de PDUs variável:

Analisando o entroncamento TDM realizado através do pseudo-circuito entre os *gateways* IPmux-11, em ensaios de 1h com enlaces de 4Mbps, para vários tamanhos de pacote (quantidade de PDUs), observa-se um comportamento similar nas Figuras 4.31, 4.32 e 4.33, indicando, como esperado, que a qualidade cai conforme o congestionamento da rede aumenta, com aumento das perdas de pacote e eventos de *overflow* e *underflow* crescentes com o aumento do tráfego concorrente, incluindo a saturação do *buffer* em 50 ms.

No que tange à percepção da qualidade do enlace TDM pela central Trópico-RA, em princípio o escorregamento é considerado inaceitável quando a perda de pacotes cresce e o *buffer* satura, mas essa medição não é tão confiável como no caso anterior pois o intervalo de coleta dos dados pela central é de uma hora, exatamente o tempo total de cada ensaio.

Contudo, cabe destacar a Figura 4.30, envolvendo pacotes com apenas 1 PDU. Nesse caso, observa-se uma surpreendente melhoria do comportamento para os tráfegos de 3Mbps e 4Mbps, com redução a quase zero da perda de pacotes e dos eventos de *overflow* e *underflow*, em função da saída do *buffer* da condição de saturação nesses casos. Isso acontece em função do mecanismo utilizado no ensaio para a geração do tráfego concorrente, que é baseado no aumento da média do tamanho do pacote, no caso para 768 e 1024 octetos. Quando esse tráfego de pacotes grande é colocado para competir pelos recursos da rede com os pacotes TDM, contendo apenas 48 octetos, a vantagem para este nas filas dos roteadores é evidente, deixando presos no congestionamento os pacotes concorrentes.

Assim, é também confirmada a expectativa de que pacotes com dados TDM sobre IP, por apresentarem, em princípio, tamanhos inferiores aos utilizados pelas aplicações VoIP, conseguem melhor desempenho em redes congestionadas e sem garantia de QoS. Evidentemente, o *overhead* para transmissão de apenas uma PDU, é proibitivo na maioria das aplicações com banda restrita.

As conclusões obtidas estão coerentes com os resultados apresentados em [PILATI2004], tomados como referência para a criação dos cenários experimentais.

#### **4.4.4 – Análise comparativa entre IPMux-11 e aplicação Java**

Numa primeira análise das Figuras 4.37 e 4.38, pode parecer que o desempenho da aplicação Java foi muito ruim, uma vez que apresentou perdas superiores a 1.000 e 1.200 pacotes, quando o tráfego concorrente foi, respectivamente, de 3Mbps e 4Mbps, enquanto o gateway IPmux-11 perdeu, diretamente, entre 50 e 65 pacotes na mesma condição.

Contudo, essa análise simplista não representa a realidade dos problemas dentro do pseudo-circuito, uma vez que a implementação da RAD está sujeita a processos de *overflow* e *underflow*, que também passaram a acontecer quando o carregamento da rede cresceu, como está indicado na Figura 4.37.

Quando um *overflow* acontece, diversos pacotes são perdidos, pois não existe mais espaço para seu armazenamento. Além disso, como descrito na seção 2.3.1, a implementação da RAD aciona um processo automático de limpeza do *buffer*, sempre que esse evento ocorre, reiniciando o processo de espera para a reprodução do fluxo TDM de saída. Dessa forma, considerando que o *buffer*, para o pseudo-circuito TDM em regime permanente, teria 50% de ocupação, de forma que o processo de reprodução estaria defasado do processo de escrita dessa mesma ordem, é razoável supor que, quando um *overflow* ocorre, pelo menos metade dos dados armazenados ainda não foram reproduzidos no fluxo de saída, e são perdidos. Assim, a perda efetiva de pacotes para o IPmux-11 deve ser considerada através da expressão (4.27):

$$Perdas_{RAD} = Perdas_{NS} + N_{Overflow} \cdot (50\% \cdot T_{Buffer}) \quad (4.27)$$

Onde:

$Perdas_{RAD}$  é a quantidade efetiva de pacotes perdidos;

$Perdas_{NS}$  é a quantidade de pacotes perdidos detectados através do número de seqüência;

$N_{overflow}$  é a quantidade de eventos de *overflow* no período;

$T_{buffer}$  é o tamanho do buffer, em pacotes.

Como no IPmux-11 o tamanho do *buffer* é alocado em tempo, tendo sido configurado para 50 ms, a Tabela 4.9 precisa ser utilizada para determinar  $T_{buffer}$ .

Tabela 4.9 – Duração dos dados TDM e quantidade de pacotes para PDUs AAL1.

PDUs	Dados TDM (ms)	Pacotes (50ms)
1	0,184	272
5	0,918	54
10	1,836	27
15	2,754	18
20	3,672	14
25	4,590	11
30	5,625	9

Além disso, como os pacotes não transportam a mesma quantidade de dados TDM para os dois fluxos, sobretudo quando são utilizadas 30 PDUs no IPmux-11, é necessário determinar a quantidade efetiva de dados perdidos com cada um dos pacotes.

Assim, a Tabela 4.10 apresenta os resultados efetivos do ensaio para comparação mais justa do desempenho obtido entre os dois fluxos, onde as Perdas RAD foram calculadas através da expressão (4.27).

Tabela 4.10 – Comparação do desempenho entre o IPmux-11 e a aplicação Java.

PDUs AAL1	Tráfego Concorrente	Em Perda de Pacotes			Em Perda de Dados TDM		
		Perdas RAD	Perdas Mux-IP	% Melhoria	Perdas RAD (ms)	Perdas Mux-IP (ms)	% Melhoria
30	1Mbps	124	71	-42,9%	700	71	-89,9%
	2Mbps	297	253	-14,8%	1.671	253	-84,9%
	3Mbps	319	569	78,5%	1.793	569	-68,3%
	4Mbps	343	1.289	275,8%	1.929	1.289	-33,2%
5	1Mbps	56	55	-1,8%	51	55	7,0%
	2Mbps	88	137	55,3%	81	137	69,1%
	3Mbps	662	511	-22,8%	608	511	-15,9%
	4Mbps	1.374	1.048	-23,7%	1.262	1.048	-16,9%

A análise desses dados demonstra que a aplicação Java, utilizando o novo algoritmo de seqüenciamento proposto, apresenta melhora de desempenho em sete das oito situações, mostrando-se inferior ao IPmux-11 apenas quando a quantidade de pacotes foi praticamente a mesma, uma vez que cada pacote SAToP usado no Java carrega 1ms de dados TDM, enquanto 5 PDUs AAL carregam 0,918 ms. Esse melhor desempenho é justificado pela maior quantidade de pacotes atrasados recuperáveis, mas sobretudo pelo problema que o *overflow* representa quando o fluxo é prejudicado por um congestionamento da rede, quando o tráfego supera demais a taxa dos enlaces.

Deve ser considerada, de fato, apenas a melhoria obtida quando os fluxos possuem pacotes semelhantes, ou seja, as linhas para 5 PDUs AAL1, pois pacotes de maior tamanho são mais prejudicados pelo congestionamento nos roteadores que os pacotes menores, além de conterem quantidade bem maior de dados, não podendo ser feita uma comparação direta.

Como conclusão das implementações e experimentos desse capítulo, foi validado o novo mecanismo de seqüenciamento, tanto na simulação MatLab<sup>®</sup> como na implementação em Java, demonstrando resultados melhores que o algoritmo IETF e a aplicação comercial. Da mesma forma foi validada, através de simulação, a proposta de PLL com controlador PID para a regeneração do relógio no receptor. Finalmente, foram comprovadas algumas premissas do transporte TDM sobre IP, como a necessidade de garantia de banda e o bom

desempenho decorrente da utilização de pacotes de menor tamanho. Finalmente, foi atestada a possibilidade de utilização da tecnologia TDMoIP<sup>®</sup> da RAD para emulação de circuitos E1 no entroncamento de centrais, com bastante sucesso.

## 5 – CONCLUSÕES E TRABALHOS FUTUROS

O estudo aprofundado das tecnologias de emulação de circuitos TDM sobre redes de pacotes realizado nesse trabalho permite concluir que a tecnologia de emulação fim a fim de circuitos TDM sobre redes IP é viável, e realmente apresenta vantagens em relação à tecnologia VoIP em algumas aplicações, sobretudo quando pré-existentes e não houver interesse em fazer novos investimentos, aproveitando apenas a melhor eficiência de utilização da banda de transmissão na rede de transporte, permitindo assim a redução de custos com esses serviços.

Essa viabilidade foi comprovada através da demonstração de que os principais desafios enfrentados no transporte através de rede de pacotes de serviços extremamente sensíveis a atraso e variação nesse atraso, como os circuitos TDM, podem ser superados, mesmo em redes onde não existam garantias de QoS, desde que exista largura banda disponível para assegurar a vazão de dados requerida por esse tipo de serviço, caracterizado pelo envio de pacotes a taxas constantes.

A superação desses desafios foi comprovada pelo desenvolvimento de um novo algoritmo, bastante simples, que se demonstrou eficiente e robusto na simulação e experimentação realizada, para o seqüenciamento dos pacotes no receptor, um dos grandes problemas enfrentados pelas redes IP com transporte UDP. A base desse algoritmo é a utilização de um *buffer* circular, completamente auto-gerenciável, onde a ordenação dos pacotes é definida exclusivamente com base no número de seqüência recebido na palavra de controle do encapsulamento utilizado para os pacotes, já durante seu armazenamento nesse *buffer*, o que torna trivial o processo de leitura e reprodução do fluxo TDM na saída do receptor, assim como permite o tempo equivalente a pelo menos um pacote de duração dos dados TDM para a mitigação de eventuais perdas, sendo compatível, também, com processos de retransmissão comumente utilizados na transmissão de vídeo. A melhoria de desempenho com a utilização desse algoritmo chegou, na simulação, a 20% em relação ao algoritmo de processamento do número de seqüência proposto no *Internet-Draft* TDMoIP<sup>®</sup>, reproduzido no Apêndice D.

Outro resultado interessante foi obtido com a simulação da proposta de utilização de um PLL para a recuperação do relógio de transmissão dos pacotes TDM, no receptor, utilizando exclusivamente os números de seqüência recebidos e dispensando completamente, para circuitos à taxa constante, a utilização de *timestamps* de qualquer espécie, com a conseqüente otimização no uso de banda pela ausência do *overhead* do protocolo RTP. Infelizmente, não foi possível a implementação desse mecanismo para testes em uma rede real, mas os resultados de simulação foram bastante promissores, incorporando as características robustas de um controlador PID à necessidade de estabilização e baixo tempo de aquisição do PLL.

Ainda sobre a tecnologia em si, a comparação com a abordagem VoIP, para aplicações específicas de substituição dos entroncamentos de voz convencionais buscando redução de custos, demonstrou que o transporte de TDM sobre IP apresenta menor latência, possibilitando a utilização de *buffers* maiores, e assim acomodar maiores variações no atraso sofrido pelos pacotes, assim como menor sensibilidade à perda de pacotes, em virtude da menor quantidade de informação dentro de cada um deles e à grande redundância existente na codificação padrão G.711, que também assegura uma qualidade MOS muito próxima da telefonia normal.

Dessa forma, o trabalho atingiu sua primeira grande Meta, tendo demonstrado que a tecnologia TDMoIP<sup>®</sup>, com sincronização adaptativa, é mais barata que PDH/SDH dedicado, conforme comparativo de custos realizado no Capítulo 1, e mais simples e eficiente que as principais tecnologias VoIP (*Voice over IP*) utilizando o protocolo RTP, no transporte fim a fim de troncos de voz TDM convencionais, conforme demonstrado, com base em resultados VoIP obtidos nas referências, no Capítulo 2.

Considerando a proposta-título do trabalho, o estudo dos padrões de codificação para transmissão de vídeo digital demonstrou que é possível atender, mantendo qualidade de vídeo compatível com os objetivos propostos, à restrição de taxa de transmissão, em 1,5Mbps, permitindo o “broadcasting” de vídeo sobre IP através de pseudo-circuitos E1.

Nesse cenário, foi proposta modificação do mecanismo de controle de qualidade para as transmissões de vídeo digital, eliminando o acoplamento receptor/transmissor e deixando a

atividade para o pseudo-circuito emulado, implementado como uma “camada de enlace” do ponto de vista da aplicação, fornecendo a esta um “tubo de bits” fim a fim confiável para o transporte dos sinais de vídeo digital através da rede comutada em modo pacotes, e utilizando os mecanismos propostos de seqüenciamento e recuperação do relógio de transmissão exatamente para assegurar essa qualidade desde que, evidentemente, exista banda disponível suficiente na rede para dar vazão à quantidade de pacotes necessária para a transmissão.

Quanto à segunda grande Meta, validar a tecnologia TDMoIP<sup>®</sup> com sincronização adaptativa pelo receptor como alternativa simples e barata ao ATM/SDH para transporte de TV digital de baixa definição, em qualidade compatível com a TV analógica atual, e para *broadcasting* IPTV a baixas taxas (até 1,5Mbps); infelizmente os estudos realizados não podem ser utilizados para uma conclusão definitiva, mas podemos considerar essa alternativa ainda promissora, em virtude do comportamento apresentado pelos pseudo-circuitos nas simulações e experimentos realizados.

Como base nos resultados obtidos, existe perspectiva de publicação de partes desse trabalho, envolvendo principalmente os novos mecanismos desenvolvidos e a avaliação da tecnologia TDMoIP<sup>®</sup> no entroncamento de voz da central Trópico-RA.

Como perspectiva de continuidade dos trabalhos, é sugerida a retomada do desenvolvimento da aplicação Mux-IP PC de acordo com a sua especificação original, utilizando uma linguagem de programação mais adequada às necessidades de controle de tempo e acesso aos recursos da máquina, podendo envolver o desenvolvimento de hardware dedicado para permitir a construção de uma interface compatível com G.703, a fim de que circuitos E1 possam ser conectados diretamente à aplicação. Uma alternativa seria a utilização de placas comerciais, similares à citada em [AWEYA2004a], para realização da interface.

Um grande aspecto não foi esgotado nesse trabalho, é a validação desse “tubo de bits” criado através da emulação de circuitos TDM para testes de transporte de vídeo, sob demanda e em tempo real. Isso pode ser feito, de forma mais simples, utilizando arquivos de vídeo codificado carregados no próprio computador, fazendo a sua segmentação em

pacotes e transmitindo sobre a rede, verificando o efeito da transmissão sob diversas condições de enlace e *buffer*. Outra alternativa, dependente do desenvolvimento da interface física para a aplicação Mux-IP PC, ou da utilização de produtos comerciais, é a aplicação direta do sinal de saída dos equipamentos codificadores, como câmeras e dispositivos de videoconferência, comparando seu desempenho com as tecnologias usuais de transmissão de vídeo sobre IP.

Finalmente, podem ser desenvolvidos estudos comparativos mais detalhados entre as aplicações TDM sobre IP e VoIP, utilizando a plataforma VoIP implantada na rede LabCom/UnB, fazendo a conectividade entre as duas aplicações e a central Trópico-RA, a fim de criar um ambiente similar de convergência ao existente nas operadoras.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [ABDALA2005] ABDALA Jr., H., SOARES, A. M., CARVALHO, P.H.P, BARRETO, P.S. ANVAME-N-ZÉ, G. e LAMBERT, R.; “*Implementação de um Ambiente de Teste e Medição para Redes Convergentes*”, XXII Simpósio Brasileiro de Telecomunicações, SBRT’05, Campinas, Setembro, 2005;
- [ALCATEL2002] ALCATEL S.A.; “*Projeto de Instalação LABCOM-CDT-UNB-BRASÍLIA*”, Tomo A, Alcatel S.A., Brasília, Março, 2002;
- [ALCATEL2002a] ALCATEL S.A.; “*Programação da Central – PRO LABCDT*”, Alcatel S.A., Brasília, Março, 2002;
- [ANATEL2006] ANATEL; “*Indicadores do Plano Geral de Metas de Universalização: Resultados Mensais*”, Setembro, 2006, disponível em: [http://www.anatel.gov.br/index.asp?link=/Telefonia\\_Fixa/stfc/indicadorespgmu/2006/tabela.htm?Cod=1820](http://www.anatel.gov.br/index.asp?link=/Telefonia_Fixa/stfc/indicadorespgmu/2006/tabela.htm?Cod=1820);
- [ANATEL2006a] ANATEL; “*SMC e SMP – Controle de Estações Móveis: Brasil – Estações em Operação*”, Setembro, 2006, disponível em: <http://sistemas.anatel.gov.br/smp/administracao/consulta/acompanhamentoestacoes/tela.asp?CodTopico=2440&CodArea=31&CodTemplate=413&CodMenuServico=43>;
- [ANSI105] ANSI T1.105 - 2001; “*Synchronous Optical Network (SONET) - Basic Description including Multiplex Structure, Rates, and Formats*”, Maio, 2001;
- [ANSI107] ANSI T1.107 - 1995; “*Digital Hierarchy - Format Specification*”, 1995;

- [ATM1997] ATM Forum; “*Circuit Emulation Service Interoperability Specification 2.0*”, Janeiro, 1997;
- [AWEYA2003] AWEYA, James; “*Trunking of TDM and Narrowband Services over IP Networks*”, International Journal of Network Management nº13, páginas 33-60, John Wiley & Sons, Ltd., Novembro, 2003;
- [AWEYA2004] AWEYA, James; “*Circuit Emulation Services over Ethernet – Part 1: Clock Synchronization using Timestamps*”, International Journal of Network Management nº14, páginas 29-44, John Wiley & Sons, Ltd., 2004;
- [AWEYA2004a] AWEYA, James; “*Circuit Emulation Services over Ethernet – Part 2: Prototype and Experimental Results*”, International Journal of Network Management nº14, páginas 45-58, John Wiley & Sons, Ltd., 2004;
- [BARRETO2001] BARRETO, Luis A. N., “*Desenvolvimento de Metodologias de Projeto de Redes de Voz sobre IP*”, Dissertação de Mestrado, Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Distrito Federal, 2001;
- [BARRETO2005] BARRETO, Priscila S., “*Otimização de Roteamento Adaptativo em Redes Convergentes com Tráfego Auto-Similar*”, Exame de Qualificação de Doutorado, Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Distrito Federal, Abril, 2005, trabalho em andamento;
- [BEZERRA2004] BEZERRA, R.S., ABDALA Jr., H., SOARES, A. M., CARVALHO, P.H.P, BARRETO, P.S.; “*Uma Ferramenta em Código Aberto para Análise de Redes Convergentes*”, 11th International Conference on Telecommunications poster session, ICT 2004, Fortaleza, Agosto, 2004;

- [BOLOT1993] BOLOT, J. C., “*Characterizing End-to-End Packet Delay and Loss in the Internet*”, Journal of High-Speed Networks, vol.2, nº3, páginas 305-323, December, 1993;
- [CISCO2006] CISCO Systems S.A.; “*Barômetro CISCO da Banda Larga: Análise de Mercado 2º.Trimestre/2006*”, IDC Brasil, 2006;
- [CISCO2004] White Paper; “*Understanding Jitter in Packet Voice Networks*”, CISCO Systems, Inc, 2004;
- [CISCO2005] White Paper; “*Understanding Delay in Packet Voice Networks*”, CISCO Systems, Inc, 2001;
- [COLLINS2001] COLLINS, Daniel; “*Carrier Grade Voice over IP*”, The McGraw-Hill Companies, Inc., USA, 2001;
- [DAZZO1988] D’AZZO, John J. e HOUPIS, Constantine H.; “*Análise e Projeto de Sistemas de Controle Lineares*”, 2ª.edição, Editora Guanabara., Rio de Janeiro, 1988;
- [GENEL2003] GENEL, Hayrettin e ERTEN, Y.M.; “*An End-to-End QoS Control Scheme for Video Transmission over Internet*”, 8th International Symposium on Computers and Communication Proceedings, ISCC 2003, Kiris-Kemer, Turkey, Junho, 2003;
- [IBGE2006] IBGE; “*PNAD – Pesquisa Nacional por Amostragem de Domicílios, Síntese de Indicadores*”, Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro, 2006 p. 205 e 206;
- [IEEE802.3] IEEE 802.3; “*IEEE Standard Local and Metropolitan Area Networks - Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*”, 2002;

- [IEEE802.1Q] IEEE 802.1Q; “*IEEE Standard Local and Metropolitan Area Networks – Virtual Bridged Local Area Networks*”, 2003;
- [IETF2006] IETF PWE3 Working Group Internet-Draft; “*TDM over IP*”, Junho, 2006, trabalho em andamento, disponível em:  
<http://www.ietf.org/internet-drafts/draft-ietf-pwe3-tdmoip-05.txt>;
- [IETF2006a] IETF PWE3 Working Group Internet-Draft; “*Structure-aware TDM Circuit Emulation Service over Packet Switched Network (CESoPSN)*”, Maio, 2006, trabalho em andamento, disponível em:  
<http://www.ietf.org/internet-drafts/draft-ietf-pwe3-cesopsn-07.txt>;
- [IETF2006b] IETF PWE3 Group Internet-Draft; “*SONET/SDH Circuit Emulation over Packet (CEP)*”, Janeiro, 2006, trabalho em andamento, disponível em:  
<http://www.ietf.org/internet-drafts/draft-ietf-pwe3-sonet-12.txt>;
- [ITS2002] ITS; “*Introdução Trópico RA*”, Apostila de treinamento, Instituto de Tecnologia de Software de São Paulo - ITS, Campinas, Julho, 2002;
- [ITS2002a] ITS; “*Operação, Manutenção e Supervisão Trópico RA*”, Apostila de treinamento, Instituto de Tecnologia de Software de São Paulo - ITS, Campinas, Julho, 2002;
- [ITU-T G702] ITU-T Recommendation G.702; “*Digital Hierarchy Bit Rates*”, Novembro, 1988;
- [ITU-T G704] ITU-T Recommendation G.704; “*Synchronous Frame Structures used at 1,544, 6,312, 2,048, 8,448 and 44,736 kbit/s Hierarchical Levels*”, Outubro, 1998;
- [ITU-T G706] ITU-T Recommendation G.706; “*Frame Alignment and Cyclic Redundancy Check (CRC) Procedures relating to Basic Frame Structures defined in Recommendation G.704*”, Abril, 1991;

- [ITU-T G707] ITU-T Recommendation G.707; “*Network Node Interface for the Synchronous Digital Hierarchy (SDH)*”, Outubro, 2000;
- [ITU-T G751] ITU-T Recommendation G.751; “*Digital Multiplex Equipments operating at the Third Order bit rate of 34,368 kbit/s and the Fourth Order bit rate of 139,264 kbit/s and using Positive Justification*”, Novembro, 1988;
- [ITU-T G810] ITU-T Recommendation G.810; “*Definitions and Terminology for Synchronization Networks*”, Agosto, 1996;
- [ITU-T Y1413] ITU-T Recommendation Y.1413; “*TDM-MPLS Network Interworking – User Plane Interworking*”, Março, 2004;
- [ITU-T Y1414] ITU-T Recommendation Y.1414; “*Voice Services – MPLS Network Interworking*”, Julho, 2004;
- [OBANDO2005] OBANDO, Martin D. Bravo, “*Comparação do Desempenho dos Codificadores de Voz G.711, G.729, G.723.1 e iLBC em transmissão com perda de pacotes*”, Dissertação de Mestrado, Universidade de Brasília, Faculdade de Tecnologia, Departamento de Engenharia Elétrica, Distrito Federal, 2005;
- [ONU2006] Organização das Nações Unidas, “*The Millenium Development Goals Report 2006*”, United Nations, New York, 2006, p.25;
- [PROMOM1999] PROMOM Ltda.; “*Trópico RA: Manual de Comandos – Versão S10.0, Código 45001049-6*”, PROMOM Eletrônica Ltda, Junho, 1999;
- [PROMOM1999a] PROMOM Ltda.; “*Trópico RA: Manual de Atividades de Manutenção Corretiva – Versão S10.0, Código 45001047-7*”, PROMOM Eletrônica Ltda, Junho, 1999;

- [PROMOM1999b] PROMOM Ltda.; “*Trópico RA: Manual de Atividades de Manutenção Preventiva – Versão S10.0, Código 45009268-9*”, PROMOM Eletrônica Ltda, Junho, 1999;
- [PROMOM1999c] PROMOM Ltda.; “*Trópico RA: Manual de Atividades de Manuseio – Versão S10.0, Código 45009267-4*”, PROMOM Eletrônica Ltda, Junho, 1999;
- [PILATI2004] PILATI, Tarcisio B. e ZUCCHI, Wagner L.; “*TDMoIP over Ethernet Backbones*”, International Workshop on Telecommunications Proceedings, IWT2004, Santa Rita do Sapucaí, Agosto, 2004;
- [RAD2003] White Paper.; “*TDMoIP x VoIP: Matching Technology to Requirements*”, RAD Data Communications Ltd., 2005;
- [RAD2005] RAD Data Communications Ltd.; “*IPmux-11 TDMoIP® Gateway – Installation and Operation Manual*”, RAD Data Communications Ltd., 2005;
- [RFC768] IETF RFC768; “*User Datagram Protocol (UDP)*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc768.txt>, Agosto, 1980;
- [RFC791] IETF RFC791; “*Internet Protocol (IP)*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc791.txt>, Setembro, 1981;
- [RFC2354] IETF RFC2354; “*Options for Repair of Streaming Media*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc3550.txt>, Junho, 1998;
- [RFC3550] IETF RFC3550; “*RTP: A Transport Protocol for Real-Time Applications*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc3550.txt>, Julho, 2003;

- [RFC3916] IETF RFC3916; “*Requirements for Pseudo-Wire Emulation Edge-to-Edge (PWE3)*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc3916.txt>, Setembro, 2004;
- [RFC3985] IETF RFC3985; “*Pseudo-Wire Emulation Edge-to-Edge (PWE3) Architecture*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc3985.txt>, Março, 2005;
- [RFC4197] IETF RFC4197; “*Requirements for Edge-to-Edge Emulation of Time Division Multiplexed (TDM) Circuits over Packet Switching Networks*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc4197.txt>, Outubro, 2005;
- [RFC4385] IETF RFC4385; “*Pseudowire Emulation Edge-to-Edge (PWE3) Control Word for Use over an MPLS PSN*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc4385.txt>, Fevereiro, 2006;
- [RFC4446] IETF RFC4446; “*IANA Allocations for Pseudowire Edge to Edge Emulation(PWE3)*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc4446.txt>, Abril, 2006;
- [RFC4447] IETF RFC4447; “*Pseudowire Setup and Maintenance Using the Label Distribution Protocol (LDP)*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc4447.txt>, Abril, 2006;
- [RFC4448] IETF RFC4448; “*Encapsulation Methods for Transport of Ethernet over MPLS Networks*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc4448.txt>, Abril, 2006;
- [RFC4553] IETF RFC4553; “*Structure-Agnostic TDM over Packet (SAToP)*”, conteúdo on-line: <http://www.ietf.org/rfc/rfc4553.txt>, Junho, 2006;

- [SCHULTER2002] SCHULTER, A., RIBEIRO, L., BECKER, V. e CAETANO, M.; “*Um ambiente de Distribuição de Vídeo MPEG2 com Suporte Multicast em Código Aberto para o Projeto I2TV*”, 4º.Workshop RNP2, Natal, 2002;
- [SOARES1995] SOARES, L. F. G., LEMOS, G. e COLCHER, S.; “*Redes de Computadores: das LANs, MANs e WANs às Redes ATM*”, Editora Campus, Rio de Janeiro, 1995;
- [STEIN2003] STEIN, Yaakov e STROEHLEIN, Brian; “*Taking an Inside Look at TDMoIP: A Tutorial*”, Março, 2003; disponível em:  
[http://www.commsdesign.com/design\\_corner/OEG20030326S0012](http://www.commsdesign.com/design_corner/OEG20030326S0012);
- [STEIN2003a] STEIN, Yaakov; “*In-depth Comparision of TDMoIP and CESoPSN*”, Junho, 2003, disponível em:  
[www.dspsp.com/tdmoip/compare.pdf](http://www.dspsp.com/tdmoip/compare.pdf);
- [STEIN2005] STEIN, Yaakov e SCHWARTZ, Eitan; “*Circuit Extension over IP: The Evolutionary Approach to Transporting Voice and Legacy Data over IP Networks – version 2.0*”, Maio 2004 , disponível em:  
<http://www.dspsp.com/tdmoip/tdmoip-wp.pdf>
- [TRYFONAS1999] TRYFONAS, Christos e VARMA, Anujan; “*Timestamping Schemes for MPEG-2 Systems Layer and Their Effect on Receiver Clock Recovery*”, IEEE Transactions on Multimedia, Vol.1, No.3, páginas 251-263, September, 1999;
- [WEIHUA2003] WEIHUA, Zhou et al.; “*A New Architecture of Converged Networks*”, International Conference on Communication Technology 2003 Proceedings, ICCT 2003, Beijing, China, 2003.

*TDMoIP*<sup>®</sup> e *TDMoIP driven* são marcas registradas da RAD Data Communications, Inc.  
*MatLab*<sup>®</sup> é marca registrada da The MathWorks, Inc.  
*Microsoft*<sup>®</sup> é marca registrada da Microsoft Corporation.

## **APÊNDICE A – ARQUITETURA PWE3**

A RFC3985 descreve a arquitetura utilizada para PWE3, ou seja, a Emulação por Pseudo-Circuitos Fim a Fim, para emulação de serviços como ATM, *Frame-Relay*, *Ethernet*, TDM e SONET/SDH, sobre redes de transporte em modo pacotes (PSN), usando especificamente redes IP e MPLS. Essa RFC apresenta a arquitetura dos quadros, define a terminologia e especifica os diversos elementos de protocolo envolvidos e suas respectivas funções.

Os pseudo-circuitos (PW) devem prover as seguintes funções com o objetivo de emular o comportamento e as características do serviço nativo:

- a) encapsulamento das PDU específicas do serviço emulado ou fluxos de dados que chegam à porta (física ou lógica) do PE de origem;
- b) transporte dos dados encapsulados através de tunelamento dentro da rede de transporte em modo pacotes;
- c) estabelecimento do pseudo-circuito, incluindo a troca e/ou distribuição de identificadores do PW utilizado pelos nós terminais do túnel dentro da rede de transporte em modo pacotes;
- d) gerenciamento de sinalização, sincronização, ordenação e quaisquer outros aspectos do serviço dentro dos limites do pseudo-circuito;
- e) gerenciamento de alarmes e condições específicas do serviço emulado.

### **A.1 - TERMINOLOGIA UTILIZADA NA ARQUITETURA PWE3**

A Arquitetura da Emulação por Pseudo-Circuitos Fim a Fim (PWE3) utiliza a seguinte terminologia formal, definida na RFC3985, cujos principais elementos já foram apresentados no Modelo de Referência PWE3 e são agora descritos em maiores detalhes:

### ***Attachment Circuit (AC)***

#### **Circuito de Acesso**

É o circuito físico ou virtual de acesso que conecta um CE a um PE, utilizando o serviço nativo que é emulado pelo pseudo-circuito (PW). Um Circuito de Acesso pode ser, por exemplo, um DLCI (*Data-Link Connection Identifier*) *Frame Relay*, um VPI/VCI (*Virtual Path Identifier/Virtual Circuit Identifier*) ATM, uma porta Ethernet, uma VLAN (*Virtual Local Area Network*), uma conexão PPP (*Point-to-Point Protocol*) sobre uma interface física, uma sessão PPP através de um túnel L2TP, ou um LSP (*Label-Switching Path*) MPLS.

#### ***CE-bound***

#### **Sentido de Destino do Pseudo-Circuito**

É a direção de tráfego onde as PW-PDUs são recebidas em um pseudo-circuito, a partir da rede de pacotes, processadas e então encaminhadas ao equipamento cliente (CE) de destino.

#### ***CE-Signaling***

#### **Sinalização CE**

É o conjunto de mensagens enviadas e recebidas pelo Plano de Controle, no serviço nativo, do equipamento do cliente (CE). Pode ser desejável, ou mesmo necessário para o equipamento provedor (PE) do pseudo-circuito, monitorar ou interagir com essa sinalização a fim de emular efetivamente o serviço.

#### ***Control Word (CW)***

#### **Palavra de Controle**

É um cabeçalho de quatro octetos utilizado em alguns encapsulamentos para transportar informações associadas a cada pacote quando a rede de pacotes é MPLS.

#### ***Customer Edge (CE)***

#### **Equipamento Cliente**

É o dispositivo de origem ou destino do serviço nativo suportado pelo pseudo-circuito de forma transparente. O CE não tem consciência de que está usando um serviço emulado ao invés do serviço nativo.

#### ***Forwarder (FWRD)***

#### **Módulo de Encaminhamento**

É um subsistema do equipamento provedor (PE) que seleciona o pseudo-circuito a utilizar na transmissão de um fluxo de dados (*payload*) recebido a partir do Circuito de Acesso.

### ***Fragmentation***

#### **Fragmentação**

É a ação de subdividir uma única PDU do serviço nativo em múltiplas PDUs antes da sua transmissão através da PSN, esperando que a PDU original seja reconstruída em algum outro lugar na rede. Os pacotes podem ser submetidos à fragmentação quando tiverem tamanho superior à MTU da rede através da qual eles serão encaminhados.

### ***Maximum Transmission Unit (MTU)***

#### **Unidade Máxima de Transmissão**

É o tamanho máximo do pacote (excluindo o cabeçalho do protocolo de enlace) que um determinado tipo de interface pode transmitir sem necessidade de fragmentação.

### ***Native Service Processing (NSP)***

#### **Processamento do Serviço Nativo**

É o processamento do fluxo de dados recebido pelo PE de origem a partir do CE de origem, antes do encaminhamento ao pseudo-circuito para transmissão através do núcleo da rede de pacotes, ou o processamento dos dados recebidos pelo PE de destino a partir do pseudo-circuito estendido sobre a PSN, antes de seu encaminhamento através do circuito de acesso (AC) para o CE de destino, ambos considerando as funcionalidades originais do serviço nativo, que são definidas por outros órgãos de padronização que não o IETF, como ITU-T (*International Telecommunications Union – Telecommunication Standardization Sector*), ANSI (*American National Standards Institute*) e AMTF (*Asynchronous Mode Transfer Forum*).

### ***Packet-Switched Network (PSN)***

#### **Rede de Transporte comutada em Modo Pacotes**

Dentro do contexto PWE3, é uma rede que utiliza IP ou MPLS como mecanismo de encaminhamento de pacotes.

### ***PE-bound***

#### **Sentido de Origem do Pseudo-Circuito**

É a direção de tráfego onde as informações recebidas do equipamento cliente (CE) de origem são adaptadas a um pseudo-circuito, e as PW-PDUs correspondentes são encaminhadas através da rede de pacotes.

### ***PE/PW Maintenance***

#### **Módulo de Manutenção do Pseudo-Circuito a partir do Provedor (PE)**

É utilizada pelo PE para estabelecer, manter e desconectar um pseudo-circuito. Pode ser acoplada à sinalização CE a fim de permitir o gerenciamento efetivo desse pseudo-circuito.

### ***Protocol Data Unit (PDU)***

#### **Unidade de Dados de Protocolo**

É a unidade de dados encaminhada/recebida para/de uma rede por uma determinada camada de protocolo.

### ***Provider Edge (PE)***

#### **Equipamento Provedor**

É o equipamento provedor, de origem ou destino, que estabelece um pseudo-circuito fim a fim (PWE3) entre um CE de origem e um CE de destino, para a emulação de um determinado serviço.

### ***Pseudo Wire (PW)***

#### **Pseudo-Circuito**

É um mecanismo que transporta os elementos essenciais de um serviço emulado a partir de um determinado PE de origem até outro(s) PE(s) de destino, através de uma rede de transporte comutada em modo pacotes.

### ***Pseudo Wire Emulation Edge-to-Edge (PWE3)***

#### **Emulação por Pseudo-Circuitos Fim a Fim**

É um mecanismo que emula os atributos essenciais de um serviço de transporte, como um feixe E1 sobre linha dedicada ou *Frame Relay*, estabelecendo conectividade fim a fim entre a origem e o destino do fluxo de dados associado a esse serviço através de uma rede de transporte comutada em modo pacotes.

### ***Pseudo Wire PDU (PW-PDU)***

#### **PDU associada ao Pseudo-Circuito**

É uma PDU encaminhada através do pseudo-circuito que contém todo o fluxo de dados e as informações de controle necessárias para emular o serviço desejado.

### ***PSN Tunnel***

#### **Túnel PSN**

É um túnel estabelecido através da rede de pacotes, dentro do qual um ou mais pseudo-circuitos podem ser transportados.

### ***PSN Tunnel Signaling***

#### **Sinalização do Túnel PSN**

É a sinalização utilizada para estabelecer, manter e eliminar um túnel PSN.

### ***PW Demultiplexer***

#### **Demultiplexador de pseudo-circuitos**

É um método definido no Plano de Dados para identificar um pseudo-circuito estabelecido até um PE.

### ***Time Division Multiplexing or Time Domain Multiplexing (TDM)***

#### **Multiplexação por Divisão no Tempo ou no Domínio do Tempo**

É uma designação freqüentemente utilizada em referência ao fluxo síncrono de bits de dados estabelecido às taxas definidas pela Recomendação ITU-T G.702 (feixes T1, E1, T3, E3).

### ***Tunnel***

#### **Túnel**

É um método para transportar informações de forma transparente através de uma rede.

A aplicabilidade da PWE3, ou seja, da Emulação por Pseudo-Circuitos Fim a Fim, a um serviço em particular depende da sensibilidade desse serviço (ou de sua implementação no equipamento cliente) aos efeitos inerentes às redes de transporte por comutação em modo pacotes, tais como perda de pacotes, atraso, variação no atraso e reordenação; e da habilidade da camada de adaptação definida na arquitetura em minimizar esses efeitos indesejáveis.

Alguns serviços, como IP sobre *Frame Relay* sobre PWE3, podem adaptar-se de forma bastante robusta às características das redes de pacotes baseadas em IP e MPLS. Outros, como a interconexão de sistemas PBX, necessitam de considerações mais cuidadosas na camada de adaptação e dentro da rede de pacotes, como funções de engenharia de tráfego. Além disso, em alguns casos, as restrições impostas pela PSN podem tornar as garantias requeridas pelo serviço nativo impossíveis de serem providas pelo pseudo-circuito.

Complementando a terminologia, são apresentados alguns termos e acrônimos utilizados com os Serviços TDM, nos seus processos particulares de sinalização, sincronização e indicação de alarmes. A sinalização é um elemento fundamental dos circuitos TDM, sendo utilizada para supervisionar e indicar o estado das aplicações de telefonia, prover alertas a estas aplicações (como requisições de conexão e desconexão), e transferir informações de roteamento e endereçamento. Esses sinais precisam ser transportados de forma confiável

sobre os pseudo-circuitos estendidos sobre as redes de pacote para que os serviços nativos de telefonia, nas extremidades de origem e destino, funcionem de maneira adequada.

### Channel-Associated Signaling (CAS)

#### Sinalização por Canal Associado

A sinalização CAS é transportada através de um canal específico (ou *timeslot*, normalmente o 16, para E1) no mesmo quadro T1 ou E1 dos sinais de voz, alocando 4 bits de sinalização para cada um dos canais de voz do feixe, o chamado Canal Associado (normalmente com taxa de 2kbps). Como a sinalização não é feita na mesma faixa de voz (*out-band*), ela pode ser transferida numa taxa mais baixa que o tráfego de dados TDM, não sendo necessária a atualização de todos os bits CAS em cada quadro TDM transmitido. Assim, um ciclo CAS atualizando todos os bits de sinalização acontece somente após um determinado número de quadros, o que define uma nova estrutura conhecida como multi-quadro ou superquadro, sendo as mais comuns as que compreendem 12, 16 ou 24 quadros, correspondendo a 1,5 ms, 2 ms e 3 ms de duração do fluxo de bits. Na transmissão de canais telefônicos no Brasil é utilizada uma estrutura de 16 quadros por multi-quadro, com duração de 2 ms, para os 30 canais de voz de um feixe E1 típico, conforme mostrado na Figura A.1.

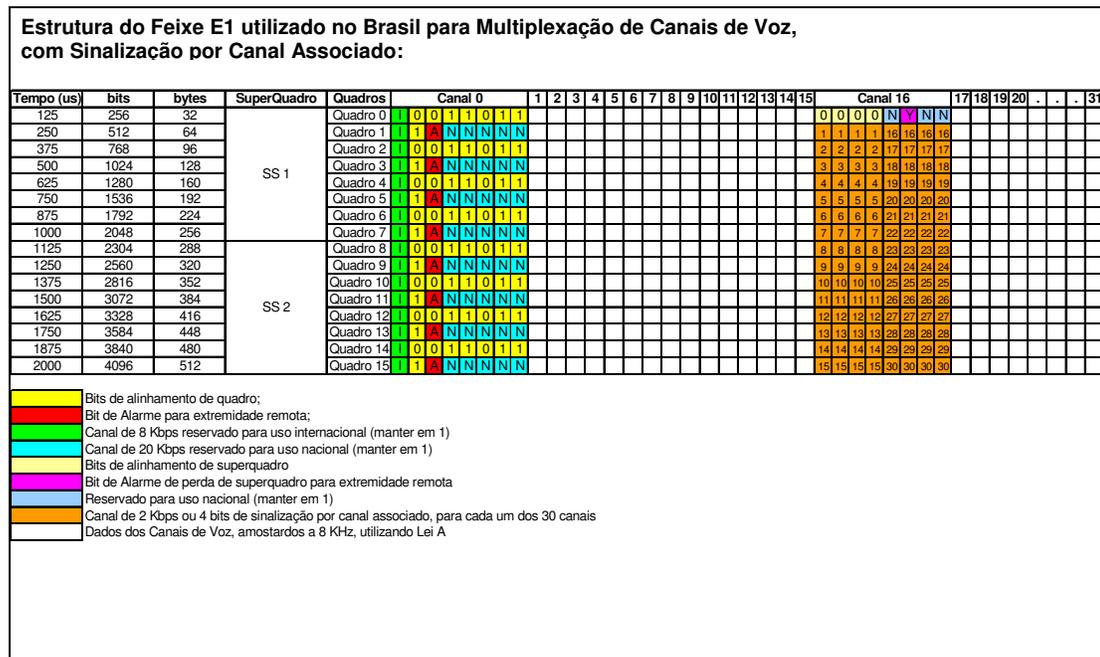


Figura A.1 – Multi-quadro com 16 quadros e sinalização CAS.

## Common Channel Signaling (CCS)

### Sinalização por Canal Comum

A sinalização CCS utiliza um canal digital em separado para transportar mensagens assíncronas relacionadas ao estado das aplicações de telefonia em cada canal de um determinado “tronco” TDM. Esse canal de sinalização pode estar fisicamente localizado em um ou mais *timeslots* adjacentes do mesmo tronco “TDM”, denominado CCS associado ao tronco, como mostrado na Figura A.2, onde é utilizado o canal 16 do feixe E1 como canal de sinalização; ou ser transportado através de uma rede independente, específica para sinalização e comutada em modo pacotes. A sinalização CCS é tipicamente baseada no protocolo HDLC, com códigos ociosos ou mensagens do tipo “manter-vivo” sendo enviadas até que um evento que necessite de sinalização, como a mudança de estado de um terminal de “no gancho” para “fora do gancho” ocorra. Alguns exemplos de sistemas de sinalização por canal comum são a SS7 (*Signaling System Number 7*), definida pela Recomendação ITU-T Q.700, e a sinalização ISDN-PRI (*Integrated Services Digital Network-Primary Rate Interface*) ou RSDI-PRI (Rede Digital de Circuitos Integrados – Interface Primária), definida pela Recomendação ITU-T Q.931.

**Estrutura do Feixe E1 utilizado no Brasil para Multiplexação de Canais de Voz, com Sinalização por Canal Comum:**

Tempo (us)	bits	bytes	Quadros	Canal 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	16	Canal 16	17	18	19	20	. . .	31
125	256	32	Quadro 0	1 0 0 1 1 0 1 1																CCS						
250	512	64	Quadro 1	1 1 A N N N N N																CCS						
375	768	96	Quadro 2	1 0 0 1 1 0 1 1																CCS						
500	1024	128	Quadro 3	1 1 A N N N N N																CCS						
625	1280	160	Quadro 4	1 0 0 1 1 0 1 1																CCS						
750	1536	192	Quadro 5	1 1 A N N N N N																CCS						
875	1792	224	Quadro 6	1 0 0 1 1 0 1 1																CCS						
1000	2048	256	Quadro 7	1 1 A N N N N N																CCS						
1125	2304	288	Quadro 8	1 0 0 1 1 0 1 1																CCS						
1250	2560	320	Quadro 9	1 1 A N N N N N																CCS						
1375	2816	352	Quadro 10	1 0 0 1 1 0 1 1																CCS						
1500	3072	384	Quadro 11	1 1 A N N N N N																CCS						
1625	3328	416	Quadro 12	1 0 0 1 1 0 1 1																CCS						
1750	3584	448	Quadro 13	1 1 A N N N N N																CCS						
1875	3840	480	Quadro 14	1 0 0 1 1 0 1 1																CCS						
2000	4096	512	Quadro 15	1 1 A N N N N N																CCS						

Bits de alinhamento de quadro;  
 Bit de Alarme para extremidade remota;  
 Canal de 8 Kbps reservado para uso internacional (manter em 1)  
 Canal de 20 Kbps reservado para uso nacional (manter em 1)  
 Canal de 64 Kbps para mensagens CCS  
 Dados dos Canais de Voz, amostrados a 8 KHz, utilizando Lei A

Figura A.2 - Utilização do canal 16 para sinalização CCS de um feixe E1.

### ***Packet Delay Variation (PDV)***

#### **Variação no Atraso dos Pacotes**

É o termo geral utilizado pela IETF para descrever as variações observadas no atraso sofrido por cada um dos pacotes que atravessam a rede (PSN). Os termos específicos *jitter* (escorregamento de bits) e *wander* (flutuação de relógio) serão utilizados apenas no contexto das redes TDM, conforme definidos na Recomendação ITU-T G.810 [ITU-T G810], para descrever as variações de curta e longa duração, respectivamente, observadas no fluxo de bits do sinal digital.

### ***Loss of Signal (LOS)***

#### **Perda de Sinal**

É um termo utilizado para designar a condição na qual um sinal TDM válido não pode ser extraído da camada física, cujos critérios efetivos de detecção e liberação estão descritos na Recomendação ITU-T G.775.

### ***Frame Alignment Signal (FAS)***

#### **Sinal de Alinhamento de Quadro**

É um termo utilizado para designar o padrão especial e periódico de bits que impõe a estrutura dos quadros E1 e T1, conforme descrito na Recomendação ITU-T G.704.

### ***Out of Frame Alignment Synchronization (OOF)***

#### **Ausência de Sincronismo de Quadros**

É um termo utilizado para designar o estado de um receptor TDM quando este não encontra um FAS válido, cujos critérios efetivos de detecção e liberação estão descritos na Recomendação ITU-T G.706. Essa condição invalida os dados TDM recebidos.

### ***Alarm Indication Signal (AIS)***

#### **Sinal de Indicação de Alarme**

É um termo utilizado para designar um padrão especial de bits dentro do fluxo TDM que indica a presença de falha em um circuito de origem, cujos critérios efetivos de indicação e liberação estão descritos na Recomendação ITU-T G.775.

### ***Remote Alarm Indication (RAI) e Remote Defect Indication (RDI)***

#### **Sinais de Indicação de Alarme ou Defeito Remoto**

São termos (normalmente sinônimos) utilizados para designar um padrão especial dentro do quadro de um serviço TDM, que é enviado de volta à origem pelo receptor que recebe um sinal AIS. Esse sinal não pode ser detectado quando uma condição LOS ou OOF é detectada ou um sinal AIS é recebido. As regras específicas para a codificação desse padrão no enquadramento TDM são discutidas na Recomendação ITU-T G.775.

## A.2 - PROTOCOLOS UTILIZADOS NA ARQUITETURA PWE3

A RFC3985 estabelece um modelo lógico de Pilha de Protocolos com o objetivo de minimizar as diferenças entre pseudo-circuitos operando sobre diferentes tipos de redes de transporte comutadas em modo pacotes (PSN). A definição dessa pilha de protocolos, cujo modelo lógico é mostrado na Figura A.3, tem como objetivos tornar a definição de cada pseudo-circuito independente da rede sobre a qual o mesmo é estabelecido e maximizar a reutilização das definições de protocolo do IETF e suas respectivas implementações.

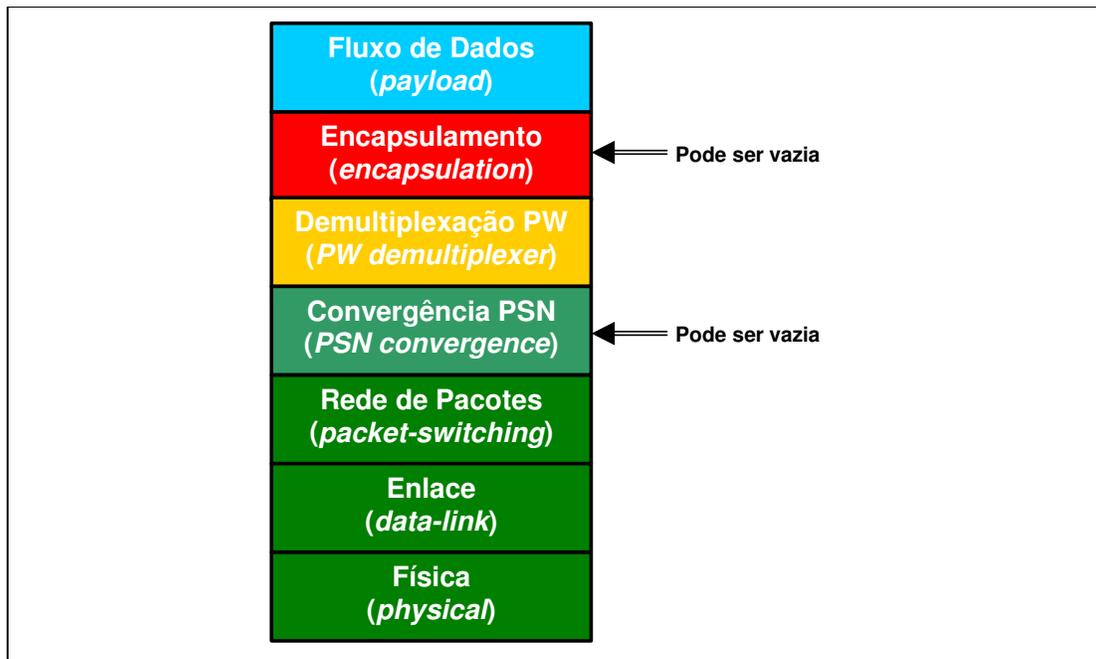


Figura A.3 - Modelo lógico da pilha de protocolos PWE3.

O fluxo de dados (*payload*) é transportado sobre a Camada de Encapsulamento, que por sua vez transporta qualquer outra informação que não esteja presente nesse fluxo de dados, mas seja necessária para que o PE de destino do pseudo-circuito (PW) possa encaminhar esse fluxo de dados ao CE de destino, através da interface física. Caso nenhuma informação complementar seja necessária além do fluxo de dados em si, essa camada é vazia.

A Camada de Encapsulamento também provê suporte para processamento em tempo real e, se necessário, para seqüenciamento dos pacotes.

A Camada de Demultiplexação dos pseudo-circuitos provê a habilidade de oferecer múltiplos PWs sobre um único túnel através da PSN. O valor do Demultiplexador PW utilizado para identificar um determinado PW dentro do Plano de Dados deve ser único para cada PE, mas esse não é um requisito PWE3. Ele deve, contudo, ser único para cada extremidade de destino do túnel. Caso seja necessário identificar um túnel em particular, isso é responsabilidade da Camada de Rede (PSN).

A Camada de Convergência PSN provê os acréscimos necessários para estabelecer a conformidade da rede de pacotes utilizada com os requisitos assumidos para a PSN pelo serviço emulado. Assim, essa camada estabelece uma interface consistente para o pseudo-circuito, tornando-o independente do tipo de rede de pacotes utilizada. Caso a PSN já atenda inerentemente aos requisitos do serviço emulado, essa camada é vazia.

As últimas três camadas são tradicionalmente encontradas nas redes de transporte e dependem do tipo de rede, de enlace e de interface física utilizada, sendo inerentes à rede de transporte em modo pacotes onde o serviço é emulado. Essa PSN pode ser uma rede IPv4 (*Internet Protocol version 4*), IPv6 (*Internet Protocol version 6*) ou MPLS.

Assim, PWE3 define a Camada de Encapsulamento, os métodos de transporte para os diversos tipos de fluxos de dados dos serviços nativos e a interface com as Camadas de Demultiplexação PW e Convergência PSN, esperando que os métodos de tunelamento sobre a PSN, como L2TP ou MPLS, sejam providos pelas demais camadas inferiores.

### **A.3 - TIPOS DE FLUXOS DE DADOS NA ARQUITETURA PWE3**

Os tipos de fluxos de dados (*payloads*) são classificados dentro dos seguintes grupos de dados nativos, correspondentes aos tipos específicos de serviço:

- **Pacotes** – Ethernet (todos), HDLC *framing*, *Frame Relay*, ATM AAL5 PDU;
- **Células** – ATM, de modo geral;
- **Fluxo de bits** – Feixes não estruturados E1, T1, E3, T3;
- **Fluxo estruturado de bits** – SONET/SDH.

Cada um desses tipos é caracterizado e tem suas funcionalidades particulares estabelecidas pela RFC3985:

#### **a) Dados em Pacotes**

Um *payload* baseado em pacotes é uma unidade de dados de tamanho variável, entregue ao Equipamento Provedor (PE) através do Circuito de Acesso (AC), que pode ser relativamente grande quando comparado à MTU da rede de pacotes. A definição dos limites do pacote depende do encapsulamento utilizado, e na grande maioria das aplicações o *overhead* de transmissão, como os *flags* e *stuffing* bits do protocolo HDLC, é eliminado antes do encaminhamento através do pseudo-circuito.

Os pacotes são normalmente encaminhados através do pseudo-circuito como um único bloco. Entretanto, podem existir casos onde o tamanho combinado entre o pacote de dados e os cabeçalhos PWE3 e PSN excede o MTU de um determinado caminho dentro da rede de pacotes. Nestes casos, algum método de fragmentação precisa ser aplicado, como quando é utilizada uma rede Ethernet para conexão ao provedor do serviço PWE3, ou quando estão envolvidos pseudo-circuitos aninhados.

Um fluxo de dados em pacotes pode necessitar de suporte para seqüenciamento (números de seqüência), permitindo o controle da sua ordenação, ou de suporte para aplicações em tempo real.

Em algumas situações, os pacotes podem ser selecionados a partir do universo de pacotes apresentado ao circuito emulado, com base em alguma técnica de multiplexação, para compartilhamento do pseudo-circuito existente. Por exemplo, um ou mais PDUs *Frame Relay* podem ser selecionadas para transporte sobre um pseudo-circuito particular com base no identificador DLCI ou, em caso de pacotes *Ethernet*, pela utilização de um filtro adequado através da camada MAC (*Media Access Control*). Essa é uma função do módulo de encaminhamento, e portanto a seleção é realizada antes do pacote ser entregue à Camada de Encapsulamento do pseudo-circuito, ou seja, na Camada de Demultiplexação.

#### **b) Dados em Células**

Um *payload* baseado em células é criado pela captura, transporte e reprodução de grupos de octetos apresentados ao circuito em um formato de tamanho fixo. A delimitação do grupo de bits que compreende a célula é específica para o tipo de encapsulamento. Dois exemplos comuns de células são as células de 53 octetos do ATM, e as grandes células de 188 octetos utilizadas no fluxo de transporte MPEG (*Moving Picture Experts Group*) utilizado no padrão europeu de TV Digital DVB (*Digital Video Broadcasting*).

Com o objetivo de reduzir o *overhead* em cada pacote encaminhado através da PSN, múltiplas células podem ser concatenadas em um único fluxo de dados. A Camada de Encapsulamento pode considerar o pacote de dados completo ao fim de um determinado intervalo de temporização, após um determinado número de células ser recebido ou quando uma célula especial - uma célula ATM OAM (*Operation and Maintenance*), por exemplo - for recebida. A vantagem de concatenar múltiplas PDUs deve ser ponderada pelo possível aumento na variação do atraso sofrido pelos pacotes e pelo maior potencial de perda de dados em caso de perda de pacotes. Em alguns casos, para minimizar esse impacto e melhorar o desempenho do pseudo-circuito, pode ser apropriado para a Camada de Encapsulamento implementar algum tipo de compressão, tal como supressão de silêncio e compressão de voz.

Esse serviço de transporte sob a forma de células normalmente necessita de suporte para seqüenciamento e também pode necessitar de suporte para aplicações em tempo real, mas usualmente não necessita de fragmentação, podendo implementar algum tipo de compressão, como a supressão de células vazias.

Em algumas situações, as células a serem incorporadas ao fluxo de dados podem ser selecionadas por filtragem a partir do fluxo de células apresentado ao circuito emulado. Por exemplo, um serviço ATM PWE3 pode selecionar células para transporte sobre um pseudo-circuito com base em seus campos VCI ou VPI.

Essa é uma função do módulo de encaminhamento, e portanto a seleção é realizada antes do pacote ser entregue à Camada de Encapsulamento do pseudo-circuito, ou seja, na Camada de Demultiplexação.

### **c) Dados em Fluxo de Bits**

Um *payload* baseado em fluxo de bits é criado pela captura, transporte e reprodução de padrões de bits sobre o circuito emulado, sem obter vantagens a partir de qualquer estruturação que, através de inspeção, possa estar disponível dentro do tráfego transportado; ou seja, a estrutura interna do fluxo não tem efeito sobre a fragmentação em pacotes.

Esse serviço baseado em fluxo de bits necessita de suporte para seqüenciamento e para aplicações em tempo real.

Em algumas situações, é possível aplicar supressão ao fluxo de bits. Por exemplo, E1 e T1 enviam “sempre uns” para indicar falha. Essa condição pode ser detectada sem nenhum conhecimento pré-determinado da estrutura do fluxo de bits, e a transmissão desses dados pode ser suprimida.

#### **d) Dados em Fluxo Estruturado de Bits**

Um *payload* baseado em fluxo estruturado de bits é criado pelo uso de algum tipo de conhecimento da estrutura interna do fluxo de bits para captura, transporte e reprodução de padrões de bits sobre o circuito emulado.

Duas importantes características distinguem os fluxos de bits estruturados, com relação aos fluxos não-estruturados:

Algumas componentes do fluxo de bits original podem ser eliminadas na direção de origem (*PE-bound*) por um bloco NSP. Por exemplo, em um fluxo SONET estruturado, os *overheads* de seção e linha (e, possivelmente, mais), podem ser eliminados, através de um bloco montador/desmontador de quadros. Esse bloco também é necessário para alinhamento de quadros em aplicações com feixes E1/T1 fracionários;

O pseudo-circuito deve preservar a estrutura através da rede de pacotes, de forma que o bloco NSP na direção de destino (*CE-bound*) possa inseri-la corretamente dentro do fluxo de bits não-estruturado reconstruído. A informação eliminada (como justificações de ponteiro SONET) pode aparecer na Camada de Encapsulamento para facilitar essa reconstituição.

Esse serviço baseado em fluxo estruturado de bits necessita de suporte para seqüenciamento e para aplicações em tempo real, podendo implementar na Camada de Encapsulamento, como opção, a supressão de silêncio/bits ociosos ou compressão similar.

Os fluxos estruturados de bits são distintos dos fluxos de dados em células, pois as suas estruturas podem ser grandes demais para serem transportadas em um único pacote. Quando essas estruturas são pequenas, a distinção desaparece, podendo ser utilizados no transporte os mesmos métodos desenvolvidos para os fluxos de dados em células.

#### **A.4 - PILHA DE PROTOCOLOS NA ARQUITETURA PWE3**

O Modelo de Referência para a pilha de protocolos de um pseudo-circuito emulado fim a fim, como estabelecida na RFC3985, é ilustrado na Figura A.4, seu objetivo é reduzir as diferenças entre pseudo-circuitos estendidos sobre diferentes tipos de redes de pacotes [WEIHUA2003]. O pseudo-circuito provê ao Equipamento Cliente (CE) uma conexão emulada física ou virtual desde a sua porta de saída até o destino remoto. As PDUs nativas provenientes do CE de origem são entregues à Camada de Encapsulamento no Equipamento Provedor (PE) de origem e então encaminhadas através do túnel formado

dentro da rede de pacotes. O PE de destino remove o encapsulamento e restaura as PDUs ao seu formato nativo para transmissão ao CE de destino.

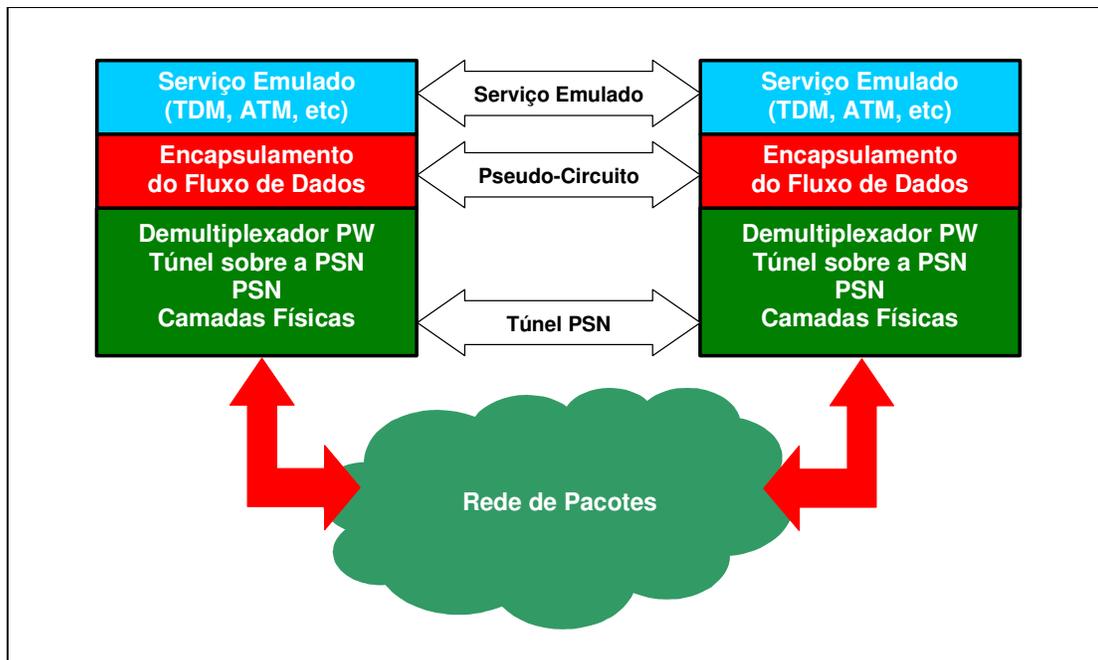


Figura A.4 - Modelo de referência para a pilha de protocolos PWE3.

A seguir são detalhadas as características da Camada de Encapsulamento e o papel das camadas inferiores na implementação de um pseudo-circuito através da Arquitetura PWE3:

a) Camada de Encapsulamento:

A Camada de Encapsulamento provê a infra-estrutura necessária para adaptar o tipo específico de fluxo de dados que está sendo transportado através do pseudo-circuito às condições esperadas pela Camada de Demultiplexação PW utilizada para estabelecer esse pseudo-circuito através da rede de pacotes, sendo constituída de três subcamadas, conforme apresentado na Figura A.5:

- Convergência do Fluxo de Dados;
- Seqüenciamento;
- Sincronização.

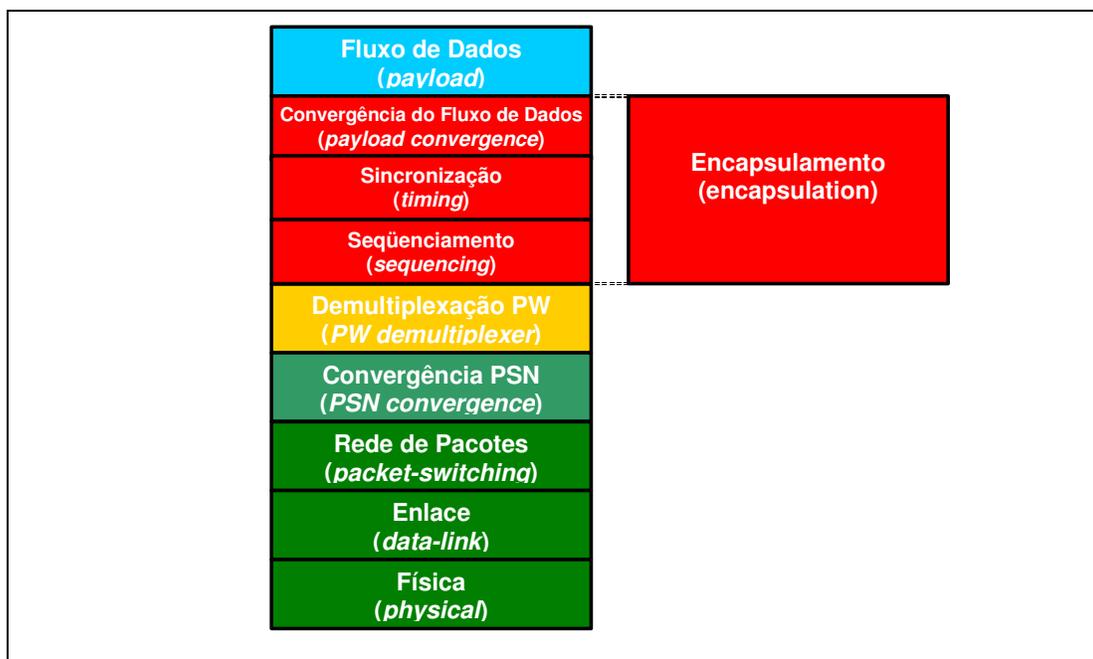


Figura A.5 - Detalhamento da camada de encapsulamento PWE3.

A subcamada de Convergência do Fluxo de Dados é fortemente ajustada ao tipo específico de fluxo de dados, de forma que é desejável agrupar um certo número de tipos desejados de fluxo em uma classe genérica e estabelecer um único tipo de subcamada de convergência para esse grupo, limitando o número de tipos de subcamada necessários e reduzindo a complexidade de implementação. A sinalização por pacote ou qualquer outra informação do tipo “fora da faixa” (*out-band*), exceto sincronização e seqüenciamento, deve ser provida por essa camada.

As subcamadas de Sincronização e Seqüenciamento provêm serviços para a subcamada de Convergência do Fluxo de Dados para todos os tipos de fluxo onde existe essa necessidade.

#### a.1) Subcamada de Convergência do Fluxo de Dados:

O objetivo principal dessa subcamada é o encapsulamento do fluxo de dados dentro das PW-PDUs. As PDUs nativas a serem encapsuladas podem conter cabeçalhos introduzidos pelas camadas 1 e 2 do serviço emulado, necessárias para a conectividade original de forma específica para cada serviço, enquanto o cabeçalho da subcamada de Convergência do Fluxo de Dados carrega a informação adicional necessária para reproduzir essas PDUs nativas para a interface física do CE de destino. Nem todas as informações adicionais

necessárias para reproduzir as PDUs nativas precisam ser transportadas no cabeçalho das PW-PDUs, uma vez que algumas informações (como, por exemplo, o tipo de pseudo-circuito) podem ser armazenadas como informações de estado no PE de destino durante o estabelecimento do pseudo-circuito.

A Camada de Encapsulamento e sua sinalização associada necessitam de um ou mais dos quatro tipos seguintes de canais, a serem providos pelas camadas inferiores de Demultiplexação PW e da PSN, sendo necessários um canal do tipo “1” e um ou mais canais dos tipos “2”, “3” e “4”:

**Tipo 1:** Um canal de controle confiável para sinalização de eventos de linha, indicação de estado e, em casos excepcionais, eventos entre os CEs que necessitem ser convertidos e transmitidos de forma confiável entre os PEs. PWE3 pode necessitar deste tipo de canal de controle para prover emulação confiável de protocolos de enlace mais complexos.

**Tipo 2:** Um canal seqüencial de alta prioridade, não confiável, para uso típico na sinalização entre os CEs. A alta prioridade pode ser indicada simplesmente através dos bits DSCP (*Differentiated Services CodePoint*) no protocolo IP ou dos bits EXP (*Expedited*) no MPLS, dando prioridade ao pacote durante a transmissão. Esse canal pode ser indicado por uma classe diferenciada, para indicar que os pacotes recebidos pelo PE devem ser processados com maior prioridade.

**Tipo 3:** Um canal seqüencial comum, para o tráfego de dados sensíveis à reordenação de pacotes, como qualquer tráfego não-IP.

**Tipo 4:** Um canal não seqüencial, para o tráfego de dados insensíveis à ordem dos pacotes, como datagramas IP.

Os canais de dados (tipos 2, 3 e 4) devem ser transportados nas PW-PDUs de dados (*in-band*), estendendo esse conceito o tanto quanto possível para uma rede de pacotes. Onde possível, é desejável que sejam empregados mecanismos de suporte à Qualidade de Serviço, usualmente designada QoS (*Quality of Service*) na transmissão das PW-PDUs através da rede de transporte comutada em modo pacotes.

Nos casos em que a conectividade fim a fim possa ser interrompida por nós NAT (*Network Address Translation*), listas de controle de acesso, *firewalls*, etc. o canal de controle deve

estar apto a manter a conectividade e estabelecer o pseudo-circuito, apesar do tráfego de dados ser bloqueado por um ou mais desses mecanismos. Nesses casos, a menos que o canal de controle seja transportado no mesmo pseudo-circuito dos dados (*in-band*), a sinalização para estabelecimento de um pseudo-circuito não confirmará a existência de um caminho de dados fim a fim. Em alguns casos, existe ainda a necessidade de sincronizar eventos CE com os dados transportados no pseudo-circuito. Esse é especialmente o caso dos circuitos TDM, onde eventos do tipo “no gancho/fora do gancho” nos comutadores da RTFC (Rede Telefônica Fixa Comutada) precisam ser transportados através um canal de controle confiável enquanto o fluxo de bits associado é transportado através de um canal seqüencial de dados.

As Subcamadas de Seqüenciamento Sincronização são opcionais e somente utilizadas quando um serviço emulado em particular necessita dos respectivos serviços oferecidos. Caso contrário, o cabeçalho associado pode ser omitido a fim de preservar recursos de processamento e da própria rede.

Algumas vezes, um tipo específico de fluxo de dados irá demandar o transporte com ou sem seqüenciamento e/ou suporte para aplicações em tempo real. Por exemplo, uma garantia do transporte em serviços *Frame Relay* é a preservação da ordem dos pacotes, de forma que muitas aplicações que o utilizam esperam a entrega dos mesmos em ordem e não estão preparadas para realizar, por si mesmas, a reordenação dos quadros, sendo fundamental que o serviço de seqüenciamento seja provido pelo pseudo-circuito emulado. Contudo, caso o serviço *Frame Relay* seja utilizado somente para transportar tráfego IP, pode ser desejável ignorar essa necessidade a fim de reduzir o custo de processamento por pacote.

O princípio básico é que, sempre que possível, um protocolo IETF existente deve ser usado para prover estes serviços. Quando um protocolo adequado não estiver disponível, o protocolo existente deve ser estendido ou modificado para atingir os requisitos PWE3, tornando-o então disponível para outras utilizações dentro do IETF.

#### a.2) Subcamada de Seqüenciamento:

A Subcamada de Seqüenciamento deve prover três serviços: ordenação de quadros, detecção de quadros duplicados e detecção de perda de quadros. Esses serviços permitem a

emulação das propriedades intrínsecas de um circuito físico, e o suporte a esses serviços pode ser omitido da implementação PWE3 caso não seja necessário.

O tamanho da faixa de números de seqüência depende da velocidade do serviço emulado e da duração máxima das condições transientes dentro da rede transporte comutada em pacotes utilizada. Uma faixa superior a  $2^{16}$  (65.536) pode, portanto, ser necessária para evitar a sobreposição de números de seqüência durante a duração dessa condição que representa o tempo de vida do pacote, tipicamente o atraso máximo sofrido pelo mesmo dentro da rede na transmissão fim a fim, considerado como a metade do RTT (*Round-Trip Time*) máximo medido na rede para os nós de origem e destino (PEs) do pseudo-circuito.

#### a.2.1) Ordenação de Quadros:

Quando pacotes transportando as PW-PDUs atravessam uma rede de pacotes, podem naturalmente chegar fora de ordem ao PE de destino. Para alguns serviços, os quadros (de controle, dados ou ambos) devem ser entregues em ordem, de forma que algum mecanismo deve ser implementado para assegurar a entrega na ordem, da mesma forma que no serviço nativo. Definir um número de seqüência no cabeçalho da Subcamada de Seqüenciamento de cada pacote é uma das aproximações possíveis. Alternativamente, pode-se observar que o seqüenciamento é uma parte do problema da entrega de pacotes para aplicações em tempo real, e um único mecanismo combinado como o protocolo RTP (*Real Time Protocol*), definido pela RFC3550 [RFC3550], poderia ser empregado.

Existem duas estratégias possíveis para assegurar a ordenação dos quadros:

- Descartar as PW-PDUs recebidas fora da ordem esperada;
- Tentar classificar as PW-PDUs recebidas, na ordem correta.

A escolha da estratégia mais adequada a um determinado serviço emulado através de um pseudo-circuito PWE3 dependerá de alguns fatores, tais como:

- Criticidade da perda de pacotes para a operação do pseudo-circuito, determinada, por exemplo, pela taxa de erros de bit, ou BER (*Bit Error Rate*) aceitável;
- Velocidades do pseudo-circuito e da rede de transporte comutada em modo pacotes utilizada;

- Atraso aceitável para os pacotes, uma vez que mais atraso precisa ser introduzido para permitir a reordenação;
- A incidência esperada de pacotes recebidos fora de ordem.

#### a.2.2) Detecção de Quadros Duplicados:

Em alguns casos raros, os pacotes transportando as PW-PDUs podem ser duplicados pela rede de pacotes sobre a qual é estabelecido o pseudo-circuito. Para alguns serviços, isso não é aceitável, sendo necessária a implementação de algum mecanismo que assegure que os quadros duplicados não serão entregues ao CE de destino. Esse mecanismo pode ser o mesmo empregado para assegurar a entrega de quadros na ordem correta.

#### a.2.3) Detecção de Perda de Quadros:

O Equipamento Provedor (PE) de destino pode determinar quando um quadro foi perdido pelo rastreamento dos números de seqüência das PW-PDU recebidas. Em algumas implementações, se uma PW-PDU não é entregue até o instante de tempo correto, o PE de destino irá presumir que a mesma foi perdida. Caso essa PW-PDU processada como perdida seja entregue posteriormente, ela deve ser simplesmente descartada pelo PE de destino.

#### a.3) Subcamada de Sincronização:

A Subcamada de Sincronização provê os serviços necessários para reproduzir, da forma mais próxima possível, as características de sincronização projetadas para as redes onde os serviços nativos seriam transmitidos, uma vez que um grande número desses serviços tem expectativas de sincronização baseadas nessas redes.

Essa camada, fundamental para diversos serviços emulados, deve permitir a entrega do tráfego nativo com uma taxa de transmissão de bits (*bit rate*), um atraso percebido pelos quadros (*delay*), uma variação máxima desse atraso (*jitter*) e uma flutuação de relógio (*wander*) similar àquela recebida pelo Equipamento Provedor (PE) de origem a partir do Equipamento Cliente (CE) de origem.

Nestes casos, o PE de destino deve reproduzir o tráfego nativo exatamente como ele foi recebido pelo PE de origem. Isso implica no encaminhamento da informação de

sincronização entre os dois PEs, ou, em alguns casos, no seu recebimento a partir de uma referência externa.

Portanto, a Subcamada de Sincronização deve prover dois serviços: recuperação de relógio e sincronização do fluxo de dados reproduzido, ou seja, entrega de cada quadro no intervalo de tempo esperado pelo serviço nativo. Cada serviço específico emulado através de um pseudo-circuito pode necessitar de um deles ou de ambos simultaneamente.

#### a.3.1) Recuperação de Relógio:

A Recuperação de Relógio é a extração da informação de temporização, ou relógio, utilizada na transmissão do fluxo de bits de saída do PE para o CE de destino, tendo como base o fluxo de pacotes entregue pela PSN a esse PE de destino, e necessita de um mecanismo adequado. Um circuito físico carrega a informação de relógio de forma intrínseca, mas a sua extração a partir de uma fonte altamente afetada pelo *jitter*, como um fluxo de pacotes, é uma tarefa relativamente complexa. Assim, é desejável que um protocolo existente para suporte a aplicações em tempo real, como o RTP, seja utilizado com esse propósito, a menos que possa ser demonstrado que o mesmo é inadequado ou desnecessário para um tipo particular de fluxo de dados.

#### a.3.2) Sincronização do Fluxo de Dados Reproduzido:

A Sincronização do Fluxo de Dados Reproduzido é a entrega das PW-PDUs não adjacentes recebidas através da PSN à interface de saída do pseudo-circuito, com um atraso constante em relação à interface de entrada do mesmo, a fim de preservar a sincronização do fluxo de dados para o serviço nativo. A temporização dessa entrega pode ser relativa a um relógio recuperado a partir do fluxo de pacotes recebido através da PSN, ou relativa a um relógio externo.

#### a.4) Fragmentação:

O ultimo aspecto associado à Camada de Encapsulamento diz respeito à Fragmentação: Nas condições ideais, um fluxo de dados seria transmitido através do pseudo-circuito em uma única PW-PDU. Entretanto, existirão casos onde o tamanho combinado do fluxo de dados e seus cabeçalhos PWE3 e PSN associados excederão a MTU de um determinado caminho dentro da rede de transporte comutada a pacotes. Quando o tamanho do pacote excede a MTU de uma dada rede, a fragmentação e posterior remontagem precisam ser

implementadas para que esse pacote seja entregue. Uma vez que ambas as operações consomem consideráveis recursos adicionais da rede, quando comparadas à simples comutação de um pacote inteiro, a necessidade de fragmentação e remontagem ao longo da rede deve ser reduzida ou eliminada tanto quanto possível, sendo uma preocupação particular em pontos de agregação onde um grande número de pseudo-circuitos são processados, como por exemplo nos PEs.

No caso ideal, o equipamento de origem do tráfego encaminhado através do pseudo-circuito possui mecanismos locais adaptativos para assegurar que pacotes com necessidade de fragmentação não sejam enviados. Quando isso não acontece, o ponto mais próximo do nó de origem dentro da rede, dotado de capacidade para fragmentação e remontagem, deve tentar reduzir o tamanho dos pacotes a fim de satisfazer à MTU da PSN. Portanto, no Modelo de Referência PWE3 apresentado na Figura 2.1, a fragmentação deve ser primeiramente realizada no CE de origem, se possível. Somente quando o CE não possa garantir um tamanho aceitável para a MTU do pseudo-circuito, o PE de origem deve tentar o seu próprio método de fragmentação.

Nos casos onde o gerenciamento da MTU falha em limitar o fluxo de dados a um tamanho adequado para transmissão pelo pseudo-circuito, o PE pode assumir tanto um método genérico de fragmentação da PW-PDU como, se disponível, utilizar o serviço de fragmentação da PSN das camadas inferiores, sendo aceitável uma implementação PE sem suporte à fragmentação, que simplesmente descarte os pacotes que excedem a MTU da rede de pacotes, notificando essa ocorrência ao Plano de Gerência do Equipamento Provedor (PE) de origem.

Caso o comprimento de um quadro das camadas 1 e 2 no PE de destino, restaurado a partir de uma PW-PDU, exceda a MTU do Circuito de Acesso (AC) de destino, ele deve ser descartado. Neste caso, o Plano de Gerência do Equipamento Provedor (PE) de destino pode ser notificado.

b) Demais Camadas inferiores:

A Arquitetura PWE3 apresenta três requisitos de serviço às camadas inferiores de protocolo utilizadas para transportar os pseudo-circuitos através de uma rede de transporte comutada em modo pacotes:

- Multiplexação, implementada na Camada de Demultiplexação PW;
- Fragmentação;
- Entrega e Informação de Comprimento das PW-PDUs.

b.1) Camada de Demultiplexação dos Pseudo-Circuitos (PW):

O objetivo da Camada de Demultiplexação PW é permitir que múltiplos pseudo-circuitos sejam carregados em um único túnel, a fim de reduzir complexidade e preservar recursos.

Alguns tipos de serviços nativos são capazes de agrupar múltiplos circuitos em um único “tronco”, como por exemplo múltiplos ATM VCs (*Virtual Circuits*) em um VP (*Virtual Path*), múltiplas *Ethernet VLANs* em um único meio físico ou múltiplos feixes independentes DS0 (64 kbps) em um único feixe T1 ou E1. Um pseudo-circuito pode realizar a interconexão entre esses troncos, e cada um dos circuitos nele contidos teria seu próprio identificador de multiplexação.

b.2) Fragmentação:

Caso a rede de pacotes possa prover os serviços de fragmentação e remontagem com performance adequada, ela pode ser utilizada para obtenção de um MTU efetivo grande o bastante para transportar as PW-PDUs, conforme discutido anteriormente.

b.3) Camada de Rede (PSN):

A entrega das PW-PDUs ao Equipamento Provedor (PE) de destino é um serviço da Camada de Rede, no caso, da rede de transporte comutada em modo pacotes. Da mesma forma, a informação de comprimento dessa PW-PDU é carregada no cabeçalho PSN. Caso a Camada de Rede inferior não provenha toda a informação necessária para determinar o comprimento de uma PW-PDU, a Camada de Encapsulamento deve provê-la.

## **A.5 - DETECÇÃO DE ERROS EM PWE3**

É uma prática comum utilizar algum mecanismo de detecção de erros para validação das PW-PDUs, tal como um Código de Redundância Cíclica, CRC (*Cyclic Redundancy Checks*) ou um algoritmo verificador similar, conhecido genericamente como *checksum* para assegurar a integridade fim a fim dos quadros. Os mecanismos específicos de cada serviço emulado através do pseudo-circuito devem definir quais *checksums* dos pacotes

devem ser preservados através do pseudo-circuito e quais devem ser removidos das PW-PDUs no PE de origem e recalculados para inserção, pelo PE de destino, dentro do fluxo de dados encaminhado para o CE de destino.

A primeira hipótese minimiza o processamento, enquanto a última economiza largura de banda. Para uma dada implementação, a escolha pode ser ditada por restrições de *hardware*, que pode não permitir a preservação do *checksum* ao longo do pseudo-circuito. Para protocolos como ATM e Frame Relay, a abrangência do checksum está restrita a um único enlace, uma vez que os identificadores de circuito DLCI (*Frame Relay*) e VPI/VCI (ATM) têm apenas significado local e mudam a cada salto. Caso o identificador do circuito (e, portanto, o *checksum*) deva mudar em função de um salto através do pseudo-circuito, seria mais eficiente eliminá-lo e recalcular o *checksum* no destino.

O documento que define o serviço específico para cada protocolo deve descrever o esquema de validação das PW-PDUs a ser utilizado.

## **A.6 - CONGESTIONAMENTO EM PWE3**

A rede de pacotes que transporta o pseudo-circuito pode estar sujeita a congestionamento, cujas características podem variar de acordo com o tipo de rede, sua arquitetura e configuração, bem como com a carga oferecida à mesma.

Caso o tráfego transportado através do pseudo-circuito seja conhecido (através de inspeção nos pacotes, por exemplo) e de característica “TCP-amigável”, ou seja, baseado no protocolo TCP (*Transmission Control Protocol*), que permite retransmissão, o descarte de pacotes pela rede irá promover a redução necessária na carga oferecida à rede, e nenhuma ação adicional para evitar congestionamento é necessária.

Caso o pseudo-circuito esteja estabelecido sobre uma rede de pacotes que provenha serviços de entrega mais confiáveis, como *DiffServ*, o PE de destino deve monitorar a perda de pacotes para assegurar que o serviço requisitado está realmente sendo oferecido. Caso não esteja, o PE deve assumir que a PSN está provendo um serviço de melhor esforço, ou BE (*Best-Effort*) e utilizar as medidas correspondentes para evitar congestionamento, descritas a seguir.

Caso um serviço BE esteja sendo usado e o tráfego não seja conhecido como “TCP-amigável”, o PE de destino deve monitorar a perda de pacotes para assegurar que a taxa de perdas encontra-se dentro do limite aceitável. A perda de pacotes é considerada aceitável se um fluxo de pacotes TCP através do mesmo caminho de rede e experimentando as mesmas condições atingiria uma vazão média, medida em escala de tempo razoável, não inferior àquela que o fluxo do pseudo-circuito está atingindo. Essa condição pode ser satisfeita pela implementação de medidas de limitação de taxa de transmissão no NSP, ou pela desconexão de um ou mais pseudo-circuitos, dependendo do tipo de tráfego que está sendo transportado. Quando o congestionamento é evitado pela desconexão de um pseudo-circuito, um mecanismo adequado deve ser implementado para evitar que o mesmo retorne imediatamente ao serviço, causando uma série de picos de congestionamento.

A comparação ao TCP não pode ser especificada exatamente, mas pretende estabelecer uma ordem de grandeza em termos de vazão e escala de tempo, estabelecida como o intervalo de ida e volta de um pacote, ou RTT (*Round-Trip Time*) numa conexão fim a fim. Em essência, esse requisito estabelece que não é aceitável implantar uma aplicação, usando PWE3 ou qualquer outro protocolo de transporte, na Internet baseada no melhor esforço, quando este consome largura de banda de forma arbitrária e não compete de forma justa com os pacotes TCP na mesma ordem de magnitude.

#### **A.7 - PLANO DE CONTROLE DA ARQUITETURA PWE3**

Como aspecto final abordado na Arquitetura PWE3, temos o Plano de Controle, que define o estabelecimento e desconexão de pseudo-circuitos. Um pseudo-circuito deve ser estabelecido antes que um serviço emulado possa ser estabelecido e deve ser desconectado quando o serviço emulado não for mais necessário.

O estabelecimento e desconexão de um pseudo-circuito pode ser provocado por um comando manual a partir do Plano de Gerência de um Equipamento Provedor (PE), pela sinalização de estabelecimento ou desconexão de um Circuito de Acesso (AC), como por exemplo um quadro ATM SVC (*Signaling Virtual Channel*), ou por um mecanismo de auto-descoberta.

Durante o processo de estabelecimento, os PEs precisam trocar informações, como a apresentação das capacidades de cada um, de forma que um túnel de sinalização precisa ser estendido para prover os mecanismos que habilitem os PEs a trocar todas as informações necessárias no interesse do pseudo-circuito a ser estabelecido. A configuração manual dos pseudo-circuitos é considerada um tipo especial de sinalização e é permitida na Arquitetura PWE3.

### A.8 - IMPLEMENTAÇÃO DA ARQUITETURA PWE3 SOBRE IP

Considerando que a definição da Arquitetura PWE3 deve empregar os protocolos IETF onde possível, a Figura A.6 apresenta uma implementação sobre uma rede de pacotes baseada no protocolo IP, descrevendo a arquitetura de protocolos IETF utilizados em cada uma das camadas estabelecidas pelo modelo de referência da pilha de protocolos.

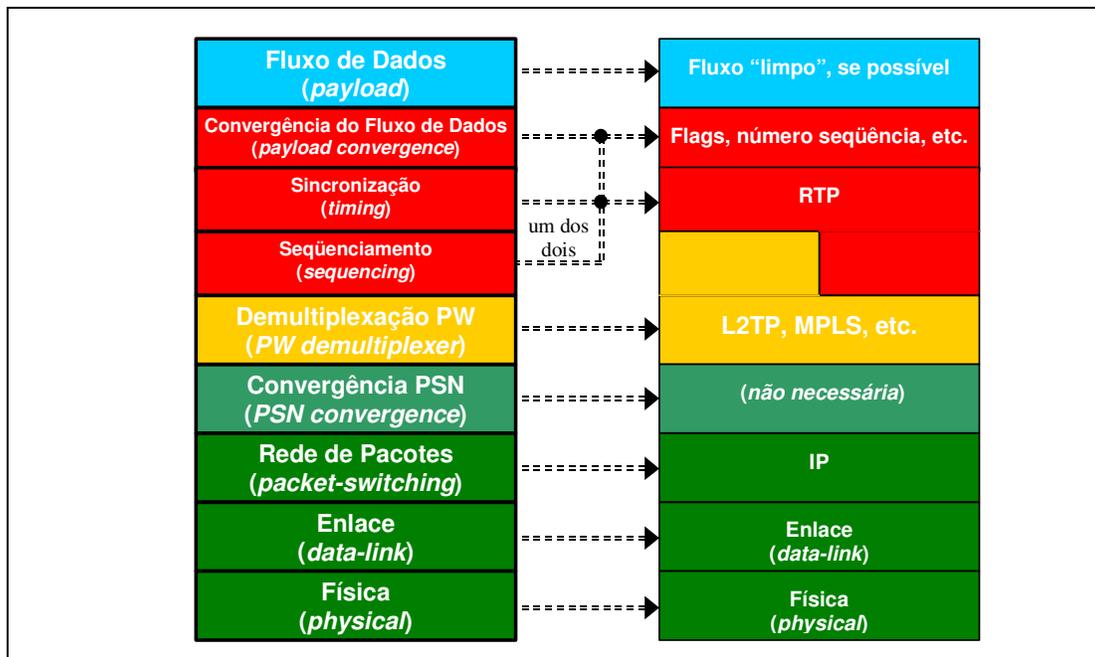


Figura A.6 - Arquitetura PWE3 sobre uma rede IP.

Como regra, o fluxo de dados deve ser transportado da forma como recebido do serviço nativo, com a adequação pela Subcamada de Convergência do Fluxo de Dados, quando necessário. Contudo, em certas circunstâncias pode ser justificável transmitir esse fluxo após alguma forma de processamento, cujas razões devem estar descritas na definição da Camada de Encapsulamento para o tipo específico de fluxo de dados.

Onde apropriado, a sincronização explícita é implementada pelo protocolo RTP que, quando utilizado, também provê o serviço de seqüenciamento. Quando a rede de pacotes é UDP/IP (*User Datagram Protocol/Internet Protocol*), o cabeçalho RTP segue o cabeçalho UDP e precede o cabeçalho PW. Para todos os outros casos o cabeçalho RTP segue o cabeçalho PW. A Camada de Encapsulamento pode, adicionalmente, conter um número de seqüência, de forma que o seqüenciamento pode ser provido tanto pelo RTP como pela Camada de Encapsulamento, mas não por ambos.

A Camada de Demultiplexação PW é provida pela etiqueta PW, que pode assumir formatos específicos em um grande número de protocolos IETF: uma etiqueta MPLS, um *Session ID* L2TP ou uma porta UDP. Quando o pseudo-circuito é transportado através de uma rede IP, nenhuma Camada de Convergência PSN é necessária. No caso especial onde a Camada de Demultiplexação PW é uma etiqueta MPLS, pode ser usada a arquitetura de protocolos apresentada na Figura A.7, em substituição àquela apresentada na Figura A.6.

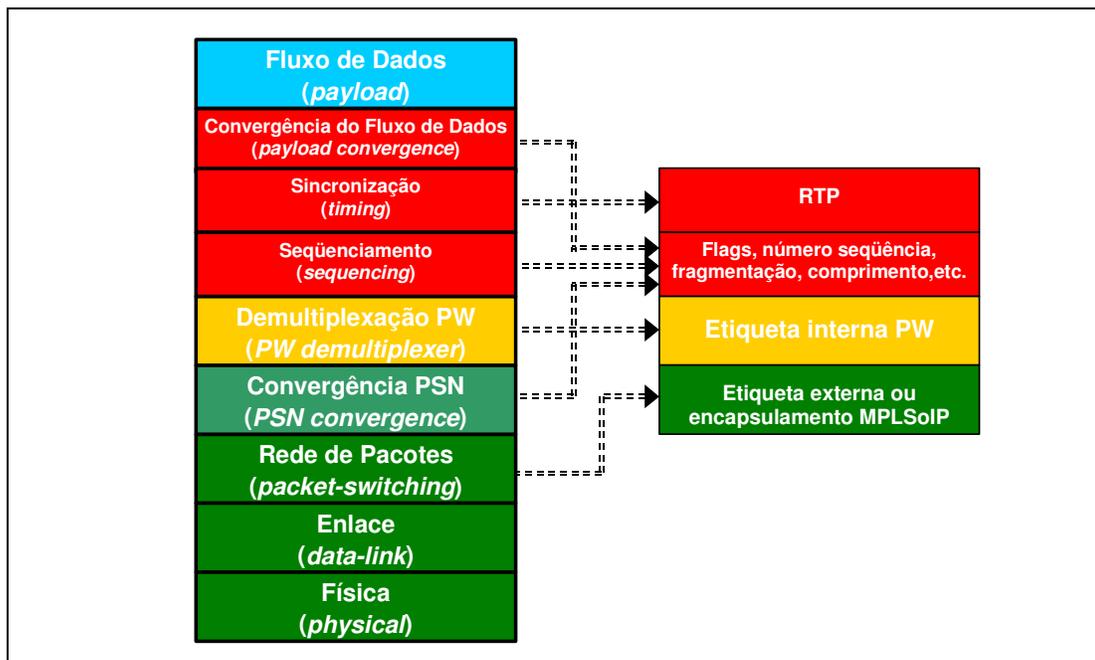


Figura A.7 - Arquitetura PWE3 sobre rede MPLS com palavra de controle.

Como pode ser observado na Figura A.7, o uso de MPLS agrega eficiência ao circuito emulado, uma vez que diversos componentes das camadas da Arquitetura PWE3 podem ser comprimidos para aumentar essa eficiência.

Uma etiqueta MPLS interna é usada para prover a função da Camada de Demultiplexação PW, enquanto uma Palavra de Controle, ou CW (*Control Word*) é utilizada para transportar a maior parte das informações necessárias para as Camadas de Encapsulamento e de Convergência PSN, em um formato compacto: Os *flags* na Palavra de Controle realizam a convergência necessária para o fluxo de dados, e um campo de número de seqüência suporta tanto a entrega ordenada dos pacotes como o serviço de fragmentação dentro da Camada de Convergência PSN (suportada por um método de controle de fragmentação). Contudo, a *Ethernet* alinha todos os quadros para o tamanho mínimo de 64 octetos, e o cabeçalho MPLS não inclui um indicador de comprimento. Portanto, para permitir que pseudo-circuitos PWE3 transportados em uma rede MPLS atravessem corretamente um enlace *Ethernet*, um campo para correção de comprimento é necessário na Palavra de Controle. Além disso, como nas redes IP, a sincronização é implementada, quando necessário, pelo protocolo RTP.

Em algumas redes, pode ser necessário transportar pseudo-circuitos PWE3 sobre MPLS sobre IP. Nessas circunstâncias, o pseudo-circuito é encapsulado para transporte MPLS como descrito, então um método de transporte de MPLS sobre redes IP é aplicado à PW-PDU resultante.

Para redes MPLS, existe uma restrição adicional para o formato do pacote PW (PW-PDU): Alguns roteadores detectam pacotes IP com base nos quatro bits iniciais do conteúdo do pacote, de forma que, para facilitar a operação, esses bits na PW-PDU não devem ser idênticos os números de versão IP em uso corrente.



## APÊNDICE B – REQUISITOS PWE3

### B.1 - REQUISITOS GERAIS PARA PWE3

A RFC3916 define PWE3, ou seja, a Emulação por Pseudo-Circuitos Fim a Fim como um mecanismo que emula os atributos essenciais de um serviço como ATM, *Frame-Relay* ou *Ethernet* sobre uma rede de transporte em modo pacotes, estabelecendo que as funções requeridas para os pseudo-circuitos incluem:

- a) o encapsulamento das PDU (*Protocol Data Units*) específicas de cada serviço que chegam à porta de ingresso na PSN, usualmente um roteador de borda da rede de transporte em modo pacotes;
- b) o transporte dessas PDU através de um caminho de rede ou túnel;
- c) o gerenciamento da sincronização e ordenação das PDU no destino;
- d) quaisquer outras operações necessárias para emular o comportamento e as características do serviço de forma tão fiel quanto possível.

A RFC3916 estabelece, ainda que, do ponto de vista do usuário, o pseudo-circuito deve ser percebido como um enlace ou circuito dedicado do serviço desejado, podendo, no entanto, haver algumas deficiências que impedem algumas aplicações de serem transportadas através do pseudo-circuito, as quais devem estar completamente descritas nos documentos específicos de cada serviço.

O modelo de referência estabelecido pela IETF na RFC3916, Figura B.1, define um pseudo-circuito, ou *pseudo-wire*, da seguinte forma:

**Pseudo-Wire (PW)** é uma conexão fim a fim entre dois equipamentos provedores do serviço emulado, ou PE (*Provider Edges*), conectados aos equipamentos dos clientes, ou CE (*Customer Edges*), através de circuitos de acesso dedicados, ou AC (*Attachments Circuits*). Pode ser um enlace *Frame Relay*, um ATM VPI/VCI, uma porta *Ethernet*, uma VLAN, um enlace HDLC, uma conexão PPP, um túnel L2TP, um LSP MPLS, etc.

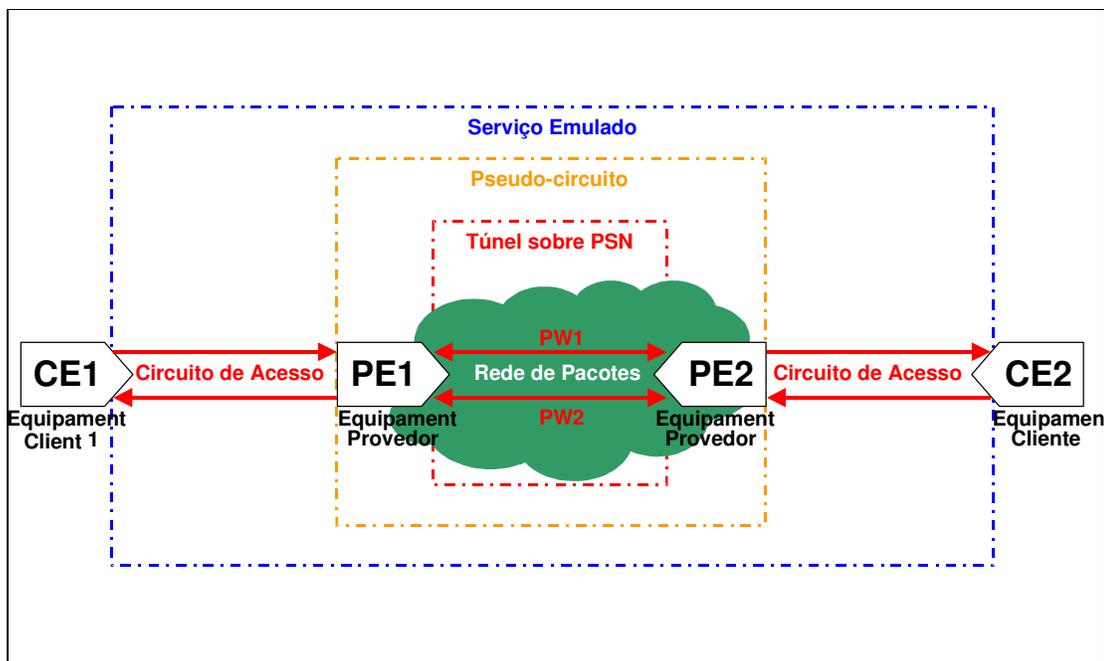


Figura B.1 - Modelo de referência PWE3.

Durante o estabelecimento de um pseudo-circuito (PW), os dois PE de origem e destino são configurados ou trocam automaticamente informações sobre o serviço a ser emulado, de forma a entender como os pacotes provenientes da origem e encaminhados ao destino devem ser processados.

Depois que o PW é estabelecido entre os dois PE, os quadros recebidos pelo PE de origem, provenientes do CE de origem pelo circuito de acesso no serviço nativo, são encapsulados e encaminhados via pseudo-circuito através da rede de pacotes, até o PE remoto de destino, onde os quadros nativos são reconstruídos e encaminhados pelo circuito de acesso até o CE remoto de destino.

No cenário original, a emulação de pseudo-circuitos é um serviço oferecido pelo provedor da rede de transporte comutada a pacotes, através de um dispositivo de interoperabilidade, ou IWF (*InterWorking Functions*), que aceita a conexão direta a partir do serviço nativo, utilizando a rede de pacotes apenas como *backbone*, como mostrado na Figura B.2.

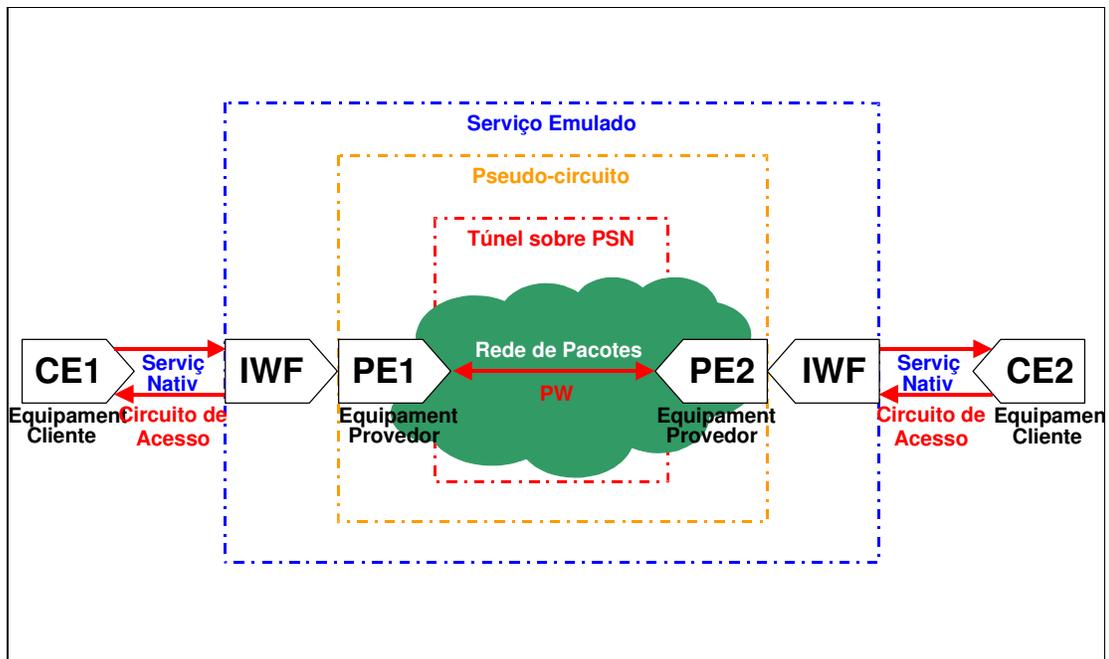


Figura B.2 - Cenário original: pseudo-circuito PE a PE.

Essa abordagem apresenta duas vantagens principais:

- Não necessita de qualquer modificação/ investimento nos equipamentos do cliente;
- A Responsabilidade de instalação/manutenção IWF é do provedor do serviço, permitindo o compartilhamento dos IWF para atender a múltiplos clientes.

E duas grandes desvantagens potenciais:

- Os custos do circuito de acesso, no serviço nativo, continuam sob a responsabilidade do cliente;
- Não permite o agrupamento de serviços sobre uma mesma conexão, no ambiente do cliente.

Dessa forma, esse cenário torna-se mais adequado para clientes de pequeno porte, onde existem poucas portas/filiais e o circuito de acesso ao provedor é largamente disponível ou relativamente barato.

Com a evolução do modelo PWE3, foi proposto um cenário alternativo, onde a emulação de pseudo-circuitos pode ser implantada pelo próprio cliente, que contrata um serviço PSN do provedor da rede de transporte comutada a pacotes, trazendo o dispositivo de interoperabilidade, ou IWF (*InterWorking Functions*), para as suas próprias instalações, como mostrado na Figura B.3, eliminando as linhas dedicadas de acesso ao provedor.

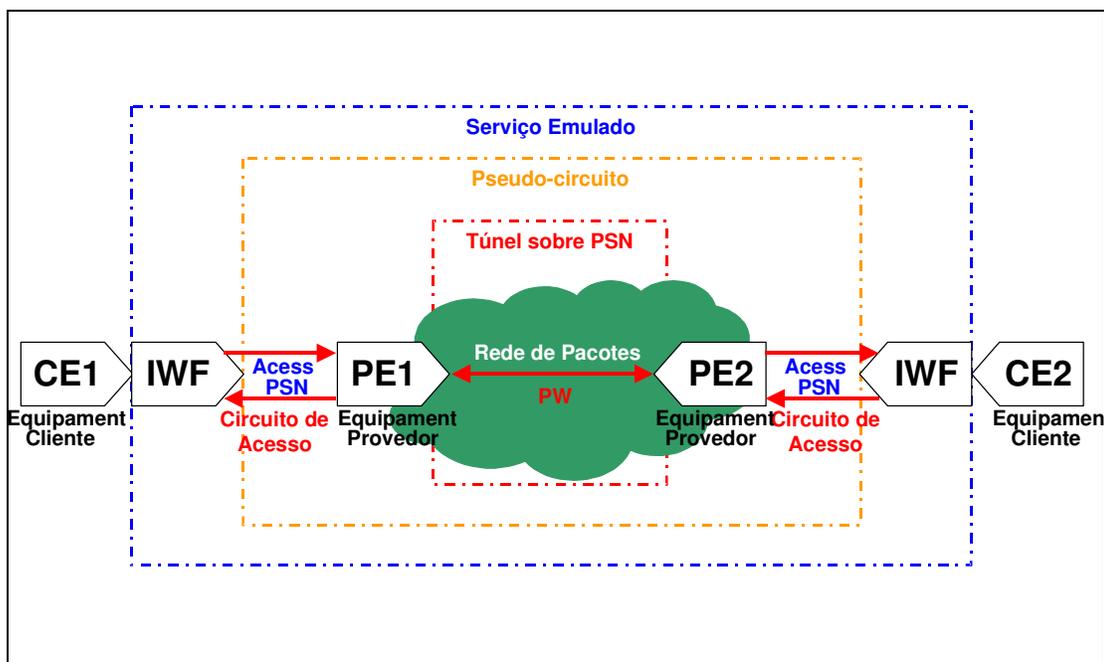


Figura B.3 - Cenário alternativo: pseudo-circuito CE a CE.

Essa nova abordagem apresenta outras duas vantagens principais:

- O acesso local em modo pacotes, tipicamente IP ou Metro Ethernet, é em princípio bem mais barato que uma linha dedicada TDM ou conexões FR e ATM;
- Essa arquitetura oferece a possibilidade de agrupamento de diferentes serviços sobre o mesmo acesso, tipicamente dos serviços de voz e vídeo com a interconexão da rede de dados do cliente.

E uma desvantagem potencial:

- A Responsabilidade de instalação/manutenção IWF é do cliente, existindo investimentos diretos em equipamentos.

Dessa forma, esse cenário torna-se mais adequado para clientes corporativos ou de porte médio, onde existem muitas portas/filiais, uma vez que a conexão PSN pode ser implementada de forma simples e integrada, utilizando, por exemplo, tecnologias xDSL.

Além dessas definições e cenários, a IETF estabelece alguns outros requisitos gerais para as aplicações PWE3, independentemente do tipo particular de serviço que esteja sendo emulado.

Os requisitos de processamento de pacotes devem prever mecanismos para:

- a) o encapsulamento dos quadros do serviço nativo, correspondentes à PDU desse serviço, tipicamente envolvendo protocolos de camada 2 do modelo de referência OSI, de forma a possibilitar a sua transmissão através da PSN e a reconstrução das mesmas no destino;
- b) a reordenação de quadros do serviço nativo, com base nos pacotes recebidos, sendo sugerida a inclusão de um número de seqüência no cabeçalho PW de cada pacote, a RFC3916 define a necessidade desses mecanismos, mas considera que a sua implementação foge ao escopo dos requisitos;
- c) a eliminação de quadros duplicados antes da entrega dos dados ao serviço nativo no destino, que podem ser os mesmos utilizados para assegurar a reordenação dos quadros;
- d) o controle de fragmentação, quando o quadro do serviço nativo (*payload*) e os cabeçalhos PW e da PSN propriamente dita excedem a MTU (*Maximum Transmission Unit*) da rede de transporte em modo pacotes utilizada, gerando necessidade de fragmentação do fluxo de dados, com impactos na performance do pseudo-circuito;
- e) o controle do *overhead* por pacote encaminhado através da PSN, quando o quadro do serviço nativo (*payload*) é relativamente pequeno, podendo ser agrupado em múltiplas unidades dentro de um mesmo pacote para redução do *overhead* introduzido pelo cabeçalho da PSN, característica que deve ser ponderada pelo impacto causado pelo atraso de processamento para montagem/desmontagem dos pacotes e maior potencial de perda de dados em caso de perda de pacotes.

Os requisitos de manutenção dos serviços emulados são claramente definidos, devendo prever mecanismos para:

- a) o estabelecimento e desconexão de pseudo-circuitos para a emulação dos serviços, que pode ser provocado por um comando através do plano de gerência do PE, automaticamente pelo estabelecimento/desconexão de um circuito de acesso (AC), ou por algum mecanismo de auto-descoberta; devendo prever a troca de informações entre os PE de origem e destino para esse estabelecimento e a coordenação entre os dois PW criados, caso o circuito correspondente do serviço nativo seja bidirecional;
- b) o tratamento de mensagens de manutenção dos serviços nativos;
- c) a iniciação de mensagens de manutenção para o serviço emulado, a partir do PE.

Os requisitos de gerência dos serviços emulados são claramente definidos, devendo prever mecanismos para:

- a) criação, leitura e atualização de MIB (Management Information Bases);
- b) configuração e provisionamento dos canais de acesso;
- c) monitoração de performance;
- d) gerenciamento de falhas e notificações;
- e) verificação de conexão e rastreamento (*traceroute*) de pseudo-circuitos.

Os requisitos de fidelidade dos serviços emulados são também abordados na RFC3916, estabelecendo:

- a) as características dos serviços emulados que devem corresponder, de forma fiel, às características dos serviços nativos, tais como:
  - a.1) tipo, velocidade e MTU do serviço emulado, que devem ser definidas para o serviço emulado, caso o sejam para o serviço nativo;
  - a.2) transparência, para os serviços nativos emulados entre dois CE de origem e destino, de um eventual compartilhamento de caminhos por dois pseudo-circuitos entre dois PE de origem e destino, dentro da rede de pacotes;
  - a.3) garantia de notificação de ambos os CE do serviço nativo em caso de falha do circuito emulado, seja no trecho do PW, seja no trecho dos AC, caso essa notificação exista no serviço nativo;
  - a.4) transparência, sobre os circuitos dos serviços emulados, a protocolos de roteamento que possam ser estabelecidos sobre os circuitos do serviço nativo;
- b) que não existe requisito de que os serviços emulados devam prover a mesma qualidade de serviço que os serviços nativos.

Além desses requisitos, são realizadas na RFC3916 algumas breves considerações sobre Qualidade de Serviço, ou QoS (*Quality of Service*), tráfego inter-domínios e segurança.

## B.2 - REQUISITOS PARA PWE3 EMULANDO SERVIÇOS TDM

Os requisitos complementares aos definidos pela RFC3916 para a utilização de pseudo-circuitos emulando serviços TDM, descritos na RFC4197, são os seguintes:

- a) Transporte de informações necessárias para reconstrução do fluxo no destino:
  - Para transporte sem consciência de estrutura (SAT), essa funcionalidade pode ser provida pela Camada de Fluxo de Dados;
  - Para transporte consciente de estrutura, a informação precisa ser provida pela Camada de Encapsulamento;
  - Para transporte de circuitos SDH/SONET, deve ser preservado o *overhead* de caminho como parte do fluxo de dados. Os componentes relevantes do *overhead* de transporte podem ser carregados na Camada de Encapsulamento.
- b) Suporte à multiplexação e demultiplexação:

Deve existir caso essas funções sejam suportadas pelos serviços nativos. Isso é relevante para NxDS0 circuitos (com ou sem sinalização) e NxVT-x em um único STS-1 ou VC-4.

  - Para esses circuitos, a combinação entre as Camadas de Encapsulamento e Fluxo de Dados deve prover tratamento separado para cada sub-circuito;
  - O pseudo-circuito deve prover suficiente informação para permitir multiplexação e demultiplexação pelo próprio NSP. A redução de complexidade da emulação PWE3 através do uso das capacidades do NSP para esse fim pode ser a solução preferencial.
- c) Interpretação ou transferência transparente de mensagens de manutenção:

Quando geradas pelos serviços nativos, dependendo do cenário em particular.
- d) Detecção e tratamento de defeitos no pseudo-circuito.
- e) Fragmentação:

Indicações de fragmentação podem ser utilizadas quando excedem o atraso de empacotamento desejado e/ou a MTU da rede entre PEs de origem e destino. Contudo, o suporte a PDUs de tamanho variável [RFC3916], não é aplicável à emulação TDM.
- f) Seqüenciamento:

Esse serviço deve ser provido pela Camada de Encapsulamento, associado a serviços de sincronização quando necessários. A informação de comprimento da PW-PDU também pode ser provida por essa camada, mas não é necessária para serviços TDM.

g) Adaptação do Fluxo de Dados:

Caso necessário, pode ser utilizada alguma forma de adaptação do fluxo TDM nativo, a fim de atingir objetivos específicos e bem documentados, devendo ser utilizadas técnicas padronizadas de adaptação.

h) Conectividade:

- A emulação deve suportar o transporte de sinais entre Circuitos de Acesso (ACs) de mesmo tipo e, sempre que apropriado, mesma taxa de transmissão.
- A Camada de Encapsulamento não deve ser afetada características particulares de conexão entre o Circuito de Acesso (AC) e o Equipamentos Provedor (PE), em ambas as extremidades do pseudo-circuito.

i) Sincronização:

A Camada de Encapsulamento deve prover serviço de sincronização suficiente para:

- Alinhar os relógios de origem e destino do serviço emulado, sem considerar o cenário específico de sincronização da rede;
- Manter o *jitter* e o *wander* do relógio de destino do serviço emulado dentro dos limites específicos definidos pelas referências normativas apropriadas;

Caso uma mesma fonte de sincronismo de alta precisão esteja disponível para todos os PEs de um determinado domínio, a Camada de Encapsulamento deve ser capaz de fazer uso disso para recuperação do relógio do serviço nativo.

j) Robustez:

A robustez do serviço emulado depende não apenas do protocolo PWE3, mas também da implementação adequada dos seguintes procedimentos:

j.1) Controle da perda de pacotes:

A emulação fim a fim de circuitos TDM pressupõe baixa probabilidade de perda de pacotes entre os PEs de origem e destino, não sendo requerido nenhum mecanismo de retransmissão.

Para minimizar o efeito da perda de pacotes, a Camada de Encapsulamento deve:

- Habilitar a interpretação independente dos dados TDM em cada pacote pelo PE de destino;
- Permitir detecção confiável dos pacotes perdidos, estimando o intervalo de chegada do próximo pacote e baseando a detecção nessa estimativa;
- Minimizar possíveis efeitos da perda de pacotes na recuperação do relógio do circuito pelo PE de destino;

- Aumentar a resiliência da interface TDM do CE à perda de pacotes, permitindo a substituição, pelo PE de destino, dos dados apropriados.

j.2) Tratamento da entrega de pacotes fora de ordem:

A Camada de Encapsulamento deve prover os mecanismos necessários para garantir a entrega ordenada de pacotes que estejam transportando dados TDM sobre a rede. Os pacotes que chegam ao PE de destino fora da ordem devem:

- Ser corretamente detectados;
- Ser reordenados, caso não sejam recebidos tarde ou cedo demais para a reprodução no fluxo TDM encaminhado ao CE;
- Ser tratados como perdidos, caso não possam ser reordenados.

k) Sinalização de aplicações CE:

Circuitos TDM não estruturados normalmente não necessitam de qualquer mecanismo especial para transporte da sinalização do Equipamento Cliente (CE), uma vez que esta é transportada como parte do serviço emulado. Contudo, algumas aplicações CE envolvendo circuitos TDM estruturados, como telefonia, precisam de sinalização específica para indicação das mudanças de estado dessas aplicações com relação ao fluxo TDM.

A Camada de Encapsulamento deve suportar a sinalização de estado das aplicações CE para os circuitos relevantes, envolvendo:

- Capacidade de suportar diferentes esquemas de sinalização com mínimo impacto sobre o encapsulamento do fluxo TDM;
- Multiplexação de sinais específicos da aplicação CE e dados do serviço TDM no mesmo pseudo-circuito;
- Sincronização, dentro dos limites de tolerância específicos da aplicação, entre os sinais CE e os dados na saída do pseudo-circuito;
- Recuperação probabilística contra perdas ocasionais de pacotes na PSN;
- Recuperação determinística do estado da aplicação CE após estabelecimento do pseudo-circuito ou falhas da rede;

A sinalização CE utilizada para propósitos de manutenção, como comandos *loopback*, recuperação de dados de monitoração de performance, etc. deve utilizar o protocolo genérico de manutenção PWE3.

l) Utilização de largura de banda da rede de pacotes:

A Camada de Encapsulamento deve buscar compromisso eficiente entre esses requisitos:

- Utilização eficiente da largura de banda.

Assumindo que o tamanho do cabeçalho da camada de encapsulamento não depende do tamanho do conjunto de dados encapsulados, um aumento no tamanho desse conjunto de dados por pacote resulta em aumento de eficiência;

- Baixa latência fim a fim.

Esse é um requisito comum para aplicações de voz sobre serviços TDM. O atraso de empacotamento é um dos componentes da latência fim a fim, que diminui com a redução do tamanho do conjunto de dados por pacote. O buffer de compensação utilizado no PE de destino aumenta a latência do circuito emulado, e os atrasos adicionais introduzidos por esse buffer não devem exceder a variação no atraso dos pacotes (PDV) observada na PSN.

A Camada de Encapsulamento pode também economizar largura de banda:

- Deixando de enviar dados corrompidos do fluxo TDM através da PSN;
- Deixando de enviar através da PSN os canais que se encontram permanentemente inativos, no caso de transporte consciente de estrutura;
- Habilitando a supressão dinâmica de canais temporariamente ociosos, também no caso de transporte consciente de estrutura, desde que isso não viole a integridade das estruturas entregues através do pseudo-circuito;

Para circuitos Nx64kbps, a Camada de Encapsulamento deve prover a capacidade de manter o atraso fim a fim independente da taxa do serviço emulado.

m) Variação no atraso dos pacotes:

A Camada de Encapsulamento deve prover a capacidade de compensação das variações no atraso sofrido pelos pacotes, enquanto mantém *jitter* e *wander* do relógio do serviço emulado dentro das tolerâncias especificadas nas referências normativas no PE de destino.

Essa camada pode permitir a adaptação dinâmica do atraso introduzido pelo *jitter buffer*, caso as variações no atraso dos pacotes se modifiquem com o tempo; tal adaptação pode introduzir um pequeno nível de erros, dentro dos limites tolerados pela aplicação, mas não deve introduzir *wander* adicional para o relógio do serviço emulado, no PE de destino.

n) Compatibilidade com a infra-estrutura das PSN existentes:

A combinação das Camadas de Encapsulamento e Túnel PSN para emulação fim a fim de circuitos TDM deve ser compatível com a infra-estrutura das PSN existentes, em particular com mecanismos de compressão de cabeçalho onde essa capacidade é um diferencial.

o) Controle de congestionamento:

Os circuitos TDM operam a taxas constantes, oferecendo carga fixa para a rede de pacotes. Assim, o mecanismo de variação da taxa de transmissão utilizado pelo protocolo TCP para ajustar a demanda ao estado de congestionamento da PSN não é aplicável, sendo necessária a capacidade de desconectar um pseudo-circuito TDM quando o congestionamento é detectado. Além disso, devem ser evitadas situações onde múltiplos PWs TDM sejam desconectados ou restabelecidos, levando a PSN à instabilidade.

p) Detecção e tratamento de defeitos:

A Camada de Encapsulamento para emulação fim a fim de serviços TDM deve, isoladamente ou em conjunto com as camadas inferiores da pilha de protocolos PWE3, prover a detecção, tratamento e notificação dos seguintes defeitos:

- Conexão indevida, ou pacotes extraviados.

A importância desse requisito é baseada na expectativa dos usuários em função da detecção bastante confiável de conexões indevidas nas redes SONET/SDH;

- Perda de pacotes.

A detecção de pacotes perdidos é requisito para manter a integridade do relógio, bem como prover a localização do problema no pseudo-circuito.

- Pacotes com problemas de montagem.

q) Monitoração de performance:

A Camada de Encapsulamento para emulação de serviços TDM deve prover coleta de dados de performance compatível com os parâmetros definidos para o TDM nativo.

r) Segurança:

Além das considerações gerais definidas na RFC3916 e totalmente aplicáveis, os serviços TDM são especialmente sensíveis à variação no atraso dos pacotes, e portanto precisam ser protegidos contra esse método de ataque.



## APÊNDICE C – ESPECIFICAÇÕES TDMoIP®

### C.1 – SUBCAMADA DE ENCAPSULAMENTO

Nos pacotes TDMoIP®, cujo formato geral é apresentado na Figura C.1, a informação contida no fluxo de dados adaptado é imediatamente precedida por uma palavra de controle, ou CW (*Control Word*), de 32 bits, contendo indicadores de alarme, o comprimento do pacote e o número de seqüência do mesmo, utilizado para detectar perdas e proceder a sua reordenação, conforme definido em [IEFT2006].

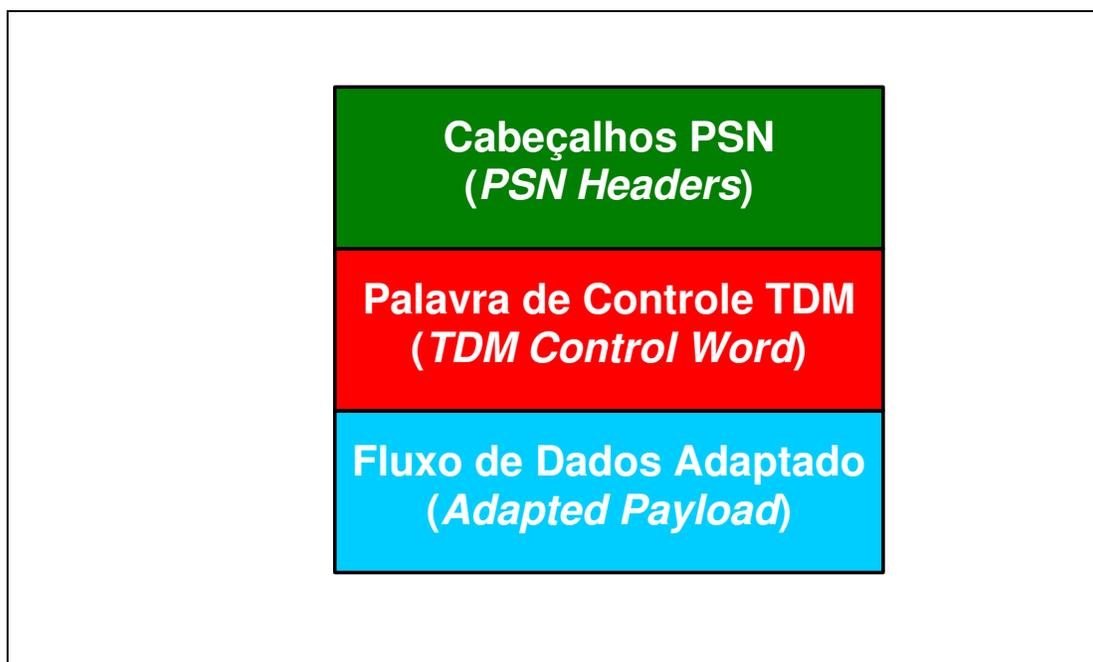


Figura C.1 - Formato geral de um pacote TDMoIP®.

Os cabeçalhos PSN são aqueles previstos pelos protocolos UDP/IP, L2TPv3/IP, MPLS ou *Ethernet*, contendo todas as informações necessárias para encaminhar o pacote através da PSN, desde a extremidade *PSN-bound* até a *TDM-bound*. A rede de pacotes é assumida pela tecnologia como razoavelmente confiável e com largura de banda suficiente para permitir o transporte dos dados TDM desejados.

Além desses, um cabeçalho opcional RTP de 12 octetos pode ser utilizado para transmitir de forma explícita a informação de temporização, através de uma *timestamp* que indica o instante de criação do pacote em unidades de um relógio comum disponível aos *gateways* TDMoIP<sup>®</sup>. Quando utilizado, o cabeçalho fixo RTP deve seguir imediatamente a palavra de controle (CW) para todos os tipos de PSN, com exceção de UDP/IP, na qual o cabeçalho RTP deve preceder a palavra de controle (CW), conforme definido pelo *Internet-Draft* [IEFT2006]. A *timestamp* deve ser gerada de acordo com as regras definidas para o RTP [RFC3550] e o relógio deve ser um inteiro múltiplo de 8kHz, escolhido de modo a permitir a recuperação de sincronismo dentro dos padrões estabelecidos para o fluxo TDM.

A recuperação do sincronismo com base *timestamp* transportada através do RTP depende da existência de um relógio comum a ambas as extremidades do pseudo-circuito, o que não é muito comum nas PSNs, o que restringe a sua utilização. Além disso, como a *timestamp* é função linear do número de seqüência replicado na palavra de controle, o cabeçalho e o protocolo RTP podem ser dispensados na maioria das aplicações envolvendo a emulação de pseudo-circuitos TDM, reduzindo o *overhead*.

A palavra de controle (CW) de 32 bits deve aparecer em todos os pacotes TDMoIP<sup>®</sup> gerados, sendo seu formato apresentado na Figura C.2.

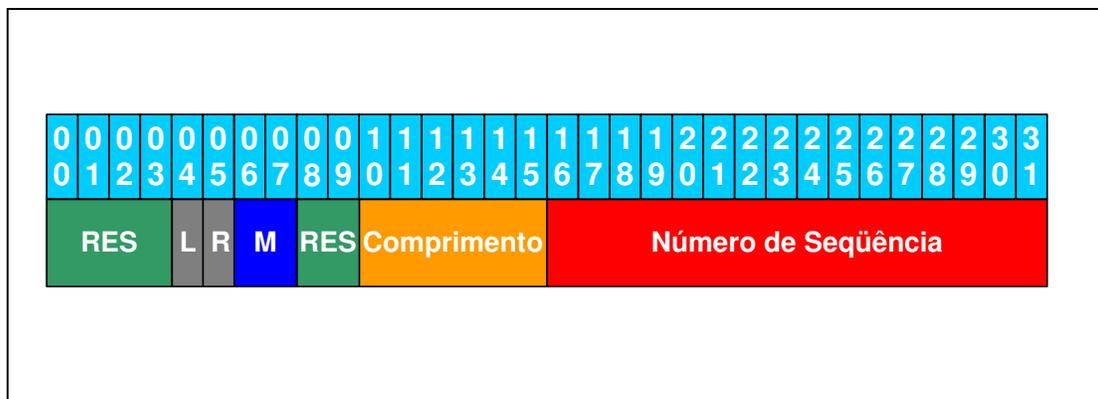


Figura C.2 - Palavra de controle TDMoIP<sup>®</sup> de 32 bits.

A distribuição dos 32 bits da palavra de controle é feita da seguinte forma:

- **RES (Reservados)**, 4 bits - o primeiro *nibble* da palavra de controle deve ser zero quando a rede de pacotes utiliza MPLS, a fim de assegurar que o pacote não seja interpretado com um pacote IP quando executada uma inspeção profunda pelos roteadores. Nas demais redes, é recomendável que esses bits sejam zero.
- **L (Falha Local)**, 1 bit – o *flag* L é ativado quando o *gateway* TDMoIP<sup>®</sup> detecta ou é informado de uma falha no circuito TDM de origem com impacto sobre o fluxo de dados TDM sendo encaminhado, como perda de sinal, perda da sincronização de quadro ou recebimento de um AIS. Quando o *flag* L é ativado o conteúdo do pacote pode não ser significativo, e o fluxo de dados pode ser suprimido a fim de conservar largura de banda. Após a remoção da falha, o *flag* L deve ser desativado.
- **R (Falha Remota)**, 1 bit – o *flag* R é ativado quando o *gateway* TDMoIP<sup>®</sup> detecta ou é informado que os dados TDM não estão sendo recebidos pelo circuito TDM de destino (rede TDM remota), indicando uma falha na direção reversa da conexão bidirecional. O *gateway* deve gerar um RDI após receber um pacote com o *flag* R ativo, e o *flag* R deve ser ativado quando do recebimento de um RDI através do circuito TDM.
- **M (Modificador de Falha)**, 2 bits – o uso do campo M é opcional, e quando utilizado complementa o significado do *flag* L.
- Quando L não está ativado, indicando dados TDM válidos, o campo M é usado com o seguinte significado:
  - 00 indica a não utilização local da modificação de falha;
  - 01 é reservado;
  - 10 é reservado;
  - 11 é reservado;Quando L está ativado, indicando dados TDM inválidos, o campo M é usado com o seguinte significado:
  - 00 indica uma falha no circuito TDM que deve disparar a geração de um AIS no PE de destino (*TDM-bound*);
  - 01 indica dados TDM ociosos ,que não devem disparar nenhum alarme.Caso o fluxo de dados tenha sido suprimido, os códigos de ociosidade pré-configurados devem ser gerados pelo PE de destino para o circuito TDM;
  - 10 indica dados TDM corrompidos mas potencialmente recuperáveis;
  - 11 é reservado;

- **RES (Reservados)**, 2 bits – esses bits são reservados e devem ser colocados em zero.
- **Comprimento (*Length*)**, 6 bits – é usado para indicar o comprimento do pacote TDMoIP<sup>®</sup> (palavra de controle e dados encapsulados), em caso de bits de alinhamento (*padding*) serem empregados para atingir requisitos de unidade mínima de transmissão da rede de pacotes. Ele deve ser usado se o comprimento total do pacote, incluindo cabeçalhos PSN, cabeçalho RTP opcional, palavra de controle e dados encapsulados, for menor que 64 octetos, e deve ser colocado em zero quando não utilizado.
- **Número de Seqüência**, 16 bits – é usado para prover a função de seqüenciamento do pseudo-circuito descrita nos requisitos PWE3, habilitando a detecção de perdas ou desordenamento de pacotes. A janela de seqüenciamento é uma faixa circular de 16 bits, sem sinal; o valor inicial do número de seqüência deve ser randômico e imprevisível, para fins de segurança e o mesmo deve ser incrementado em módulo  $2^{16}$ , de forma independente para cada pseudo-circuito.

Para o encapsulamento do fluxo de dados TDM dentro dos pacotes TDMoIP<sup>®</sup>, o PE de origem (PSN-bound) extrai octetos consecutivos do fluxo TDM contínuo, preenchendo cada octeto a partir de seu bit mais significativo. Os octetos extraídos são então adaptados utilizando os algoritmos AAL1 ou AAL2 e o fluxo de dados resultante é colocado dentro do pacote, acompanhados dos respectivos cabeçalhos característicos da PSN utilizada:

#### a) TDM sobre UDP/IP

Quando a rede comutada em modo pacotes utilizada é IP, deve ser utilizado o protocolo UDP, uma vez que retransmissões não fazem sentido dadas as características de tempo real dos circuitos TDM.

Os cabeçalhos IP [RFC3550] e UDP [RFC768] estão descritos nas respectivas RFCs, devendo preceder os dados TDMoIP<sup>®</sup>, sendo o formato completo do pacote TDM sobre UDP/IPv4, incluindo o cabeçalho do protocolo RTP opcional, apresentado na Figura C.3.

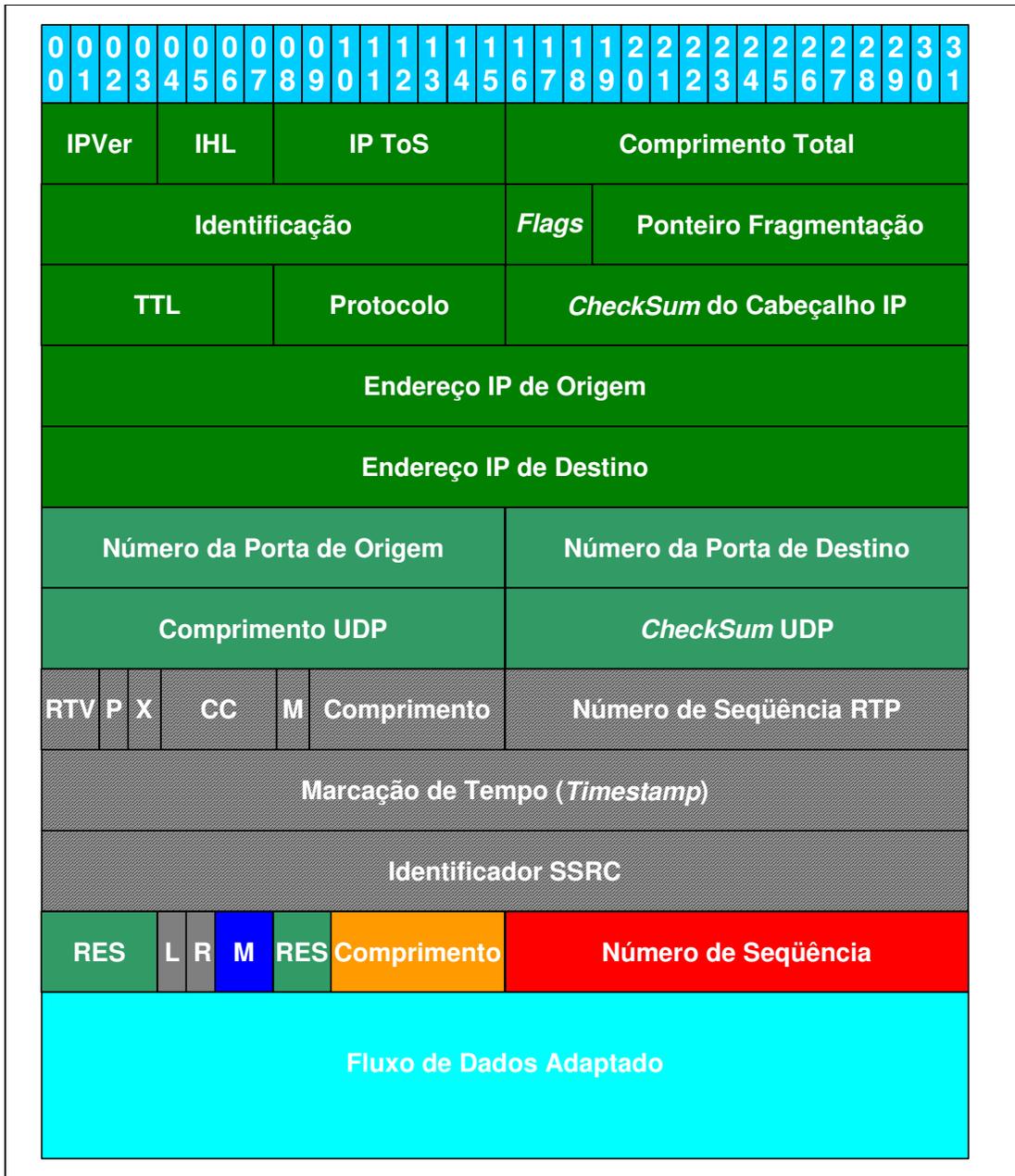


Figura C.3 - Formato de um pacote TDMoIP<sup>®</sup> sobre UDP/IP.

As primeiras 5 linhas de 32 bits são o cabeçalho IP, totalizando 20 octetos; a sexta e sétima linhas são o cabeçalho UDP, totalizando 8 octetos; as linhas 8, 9 e 10 são o cabeçalho opcional RTP, constituído por 12 octetos; a linha 11 é a Palavra de Controle TDMoIP<sup>®</sup>, de 4 octetos; e as seguintes representam os dados TDM encapsulados utilizando AAL1 ou AAL2. Segue a descrição dos campos desse pacote, com exceção da Palavra de Controle, já descrita anteriormente:

- **IPver** (*IP version*), 4 bits – é a versão do protocolo IP; no caso, IPver=4;
- **IHL** (*IP Header Length*), 4 bits – é o comprimento, em palavras de 32 bits, do cabeçalho IP; no caso, IHL=5;
- **IP ToS** (*IP Type of Service*), 8 bits – é campo com o tipo de serviço IP;
- **Comprimento Total**, 16 bits – é o comprimento, em octetos, do pacote, incluindo cabeçalhos e dados;
- **Identificação**, 16 bits – é o campo de identificação de fragmentação IP;
- **Flags**, 3 bits – são os *flags* de controle IP e devem ser definidos como *Flags*=010 para evitar fragmentação;
- **Ponteiro Fragmentação**, 13 bits – localiza o fragmento dentro do datagrama IP em não é utilizado para TDMoIP®;
- **TTL** (*Time to Live*), 8 bits – é campo IP que indica o tempo de vida do pacote, datagramas com zero neste campo são descartados pela rede;
- **Protocolo**, 8 bits – indica o protocolo utilizado, deve ser definido como 0x11 ou 17, que significa UDP;
- **Checksum do cabeçalho IP**, 16 bits – é a soma de verificação (*checksum*) do cabeçalho IP;
- **Endereço IP de Origem**, 32 bits – é o endereço IP do PE de origem;
- **Endereço IP de Destino**, 32 bits – é o endereço IP do PE de destino;
- **Números de Porta de Origem e Destino**, 16 bits cada – indica a porta UDP utilizada pelos PEs de origem e destino para o serviço PWE3, devendo ser manualmente configuradas de modo a indicar o rótulo do pseudo-circuito criado, de forma que o endereço IP de destino e um dos números de porta identifique unicamente o fluxo TDM sendo transportado. A escolha do número de porta de origem ou de destino para esse rótulo é dependente da implementação, mas deve ser acordada pelos dois PEs. Quando utilizada como identificador do fluxo TDM, o número de porta UDP deve estar compreendido na faixa de alocação dinâmica, ou seja, de 49152 a 65535. O valor 0 é reservado, e o valor 8191 (0x1FFF) é pré-configurado para o pseudo-circuito de operação e manutenção (OAM). Além disso, quando a porta de origem é utilizada para identificar o fluxo TDM, a porta de destino deve ser fixada em 2142 (0x085E), o número alocado pela IANA (*Internet Assigned Numbers Authority*) para TDMoIP®.
- **Comprimento UDP**, 16 bits – é o comprimento, em octetos, do cabeçalho UDP e dados;

- **Checksum UDP**, 16 bits – é a soma de verificação (*checksum*) dos cabeçalhos UDP/IP e dados, caso não calculado, deve ser definido como zero;
- **RTV (RTP Version)**, 2 bits – é a versão do protocolo RTP; no caso, RTV=2;
- **P (Padding)**, 1 bit – indica que o pacote contém um ou mais octetos de alinhamento ao final do fluxo de dados, deve ser colocado em zero nas aplicações TDMoIP®;
- **X (Header Extension)**, 1 bit – indica a existência de uma extensão do cabeçalho fixo RTP, deve ser colocado em zero nas aplicações TDMoIP®;
- **CC (CSRC count)**, 4 bits – contém o número de identificadores CSRC (*Contributing SouRCe*) que seguem o cabeçalho fixo RTP, deve ser colocado em zero nas aplicações TDMoIP®;
- **M (Marker)**, 1 bit – permite a marcação de eventos significativos dentro do fluxo de pacotes, como o limite dos quadros, deve ser colocado em zero nas aplicações TDMoIP®;
- **PT (Payload Type)**, 7 bits – identifica o formato do fluxo de pacotes RTP e determina a sua interpretação pela camada de aplicação, deve ser alocado na faixa de mapeamento dinâmico de códigos;
- **Número de Seqüência RTP**, 16 bits – é um número incrementado de um a cada pacote de dados RTP enviado, que pode ser utilizado pelo receptor para detectar a perda e/ou restaurar a seqüência dos pacotes recebidos, seu valor inicial é aleatório e deve ser idêntico ao número de seqüência utilizado na palavra de controle TDMoIP®;
- **Timestamp**, 32 bits – a marcação de tempo (*timestamp*) reflete o instante de amostragem do primeiro octeto no pacote de dados RTP, derivada do relógio do sistema e incrementada monotônica e linearmente no tempo para permitir a sincronização e o cálculo do *jitter*. A resolução do relógio deve ser suficiente para assegurar a precisão de sincronização desejada;
- **Identificador SSRC (Synchronization SouRCe)**, 32 bits – identifica a fonte de sincronização, ou seja, a referência para o número de seqüência e a *timestamp* do fluxo de pacotes RTP, deve ser escolhido de forma aleatória para evitar que duas fontes possuam o mesmo identificador numa mesma sessão RTP.

**b) TDM sobre MPLS**

Quando a rede de pacotes utilizada é MPLS, seu cabeçalho, incluindo as etiquetas do túnel e do pseudo-circuito, deve preceder diretamente a Palavra de Controle TDMoIP<sup>®</sup>, seguida pelo cabeçalho do protocolo RTP opcional, conforme apresentado na Figura C.4.

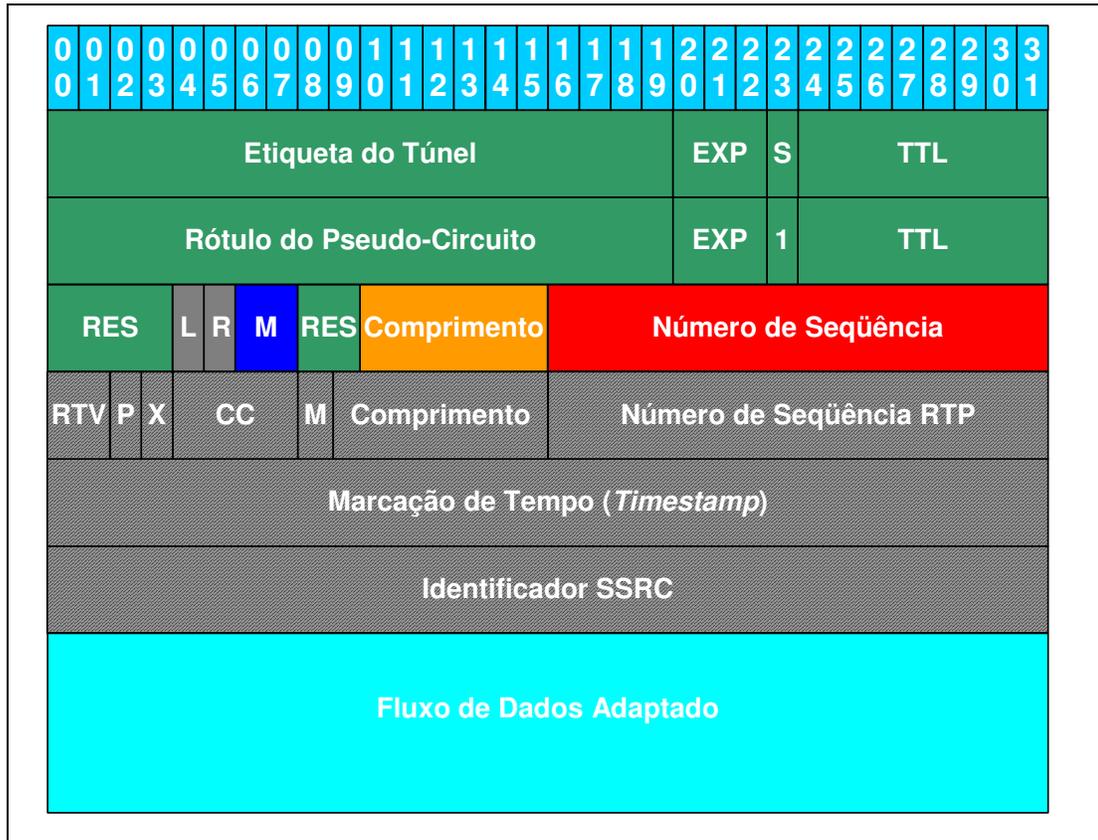


Figura C.4 - Formato de um pacote TDMoIP<sup>®</sup> sobre MPLS.

As primeiras 2 linhas de 32 bits são o cabeçalho MPLS, totalizando 8 octetos; a terceira linha é a Palavra de Controle TDMoIP<sup>®</sup> de 4 octetos; as linhas 4, 5 e 6 são o cabeçalho opcional RTP, de 12 octetos; e as seguintes são os dados TDM encapsulados utilizando AAL1 ou AAL2. Segue a descrição dos campos desse pacote ainda não abordados:

**Etiqueta do Túnel**, 20 bits – é a etiqueta que identifica os LSPs (*Label Switching Paths*), também denominado “túneis MPLS” através dos quais os pacotes contendo dados TDM devem ser encaminhados através da rede MPLS. Essa etiqueta pode ser associada tanto através de provisionamento manual como via um protocolo de controle MPLS, como o LDP (*Label Distribution Protocol*). Enquanto o pacote é encaminhado pela rede MPLS podem haver zero, uma ou múltiplas etiquetas empilhadas.

- **EXP (Experimental)**, 3 bits – é um campo experimental, pode ser utilizado para transportar a classificação *DiffServ* para as etiquetas dos túneis.
- **S (Stacking)**, 1 bit – o bit de pilha indica o fundo da pilha MPLS, S=0 para todas as etiquetas associadas a túneis, e S=1 para o rótulo do pseudo-circuito.
- **TTL (Time to Live)**, 8 bits – é campo MPLS que indica o tempo de vida do pacote dentro do túnel (LSP).
- **Rótulo do Pseudo-Circuito**, 20 bits – é o rótulo que identifica o pseudo-circuito estendido entre os PEs de origem e destino através da rede MPLS; esse rótulo deve ser uma etiqueta MPLS válida e pode ser configurado manualmente ou através de um protocolo de sinalização.

### c) TDM sobre *Ethernet*

Um pacote TDMoIP<sup>®</sup> (Palavra de Controle, cabeçalho do protocolo RTP opcional e dados adaptados através de AAL1 ou AAL2) pode ser encapsulado diretamente em um quadro *Ethernet*, quando precedido pelos respectivos endereços MAC (*Media Access Control*) de destino e origem, e sucedido pelos 4 bytes FCS (*Frame Check Sequence*) da seqüência de verificação de quadro. As implementações TDMoIP<sup>®</sup> devem estar aptas a receber quadros tanto no padrão *Ethernet* DIX (Digital, Intel e Xerox - o consórcio de desenvolvedores originais), como no padrão IEEE 802.3 [IEEE802.3], devendo transmitir apenas quadros no padrão *Ethernet* DIX. Esse pacote é apresentado na Figura C.5.

O encapsulamento *Ethernet* introduz restrições para o tamanho dos pacotes: Caso o pacote contendo dados TDM tenha menos de 64 octetos, o tamanho mínimo, bits de alinhamento devem ser introduzidos e o comprimento real indicado através do campo “Comprimento” na Palavra de Controle. Da mesma forma, a fim de evitar fragmentação, a quantidade de dados TDM no pacote deve ser restrita a um tamanho máximo, evitando que o MTU de 1500 octetos seja superado.

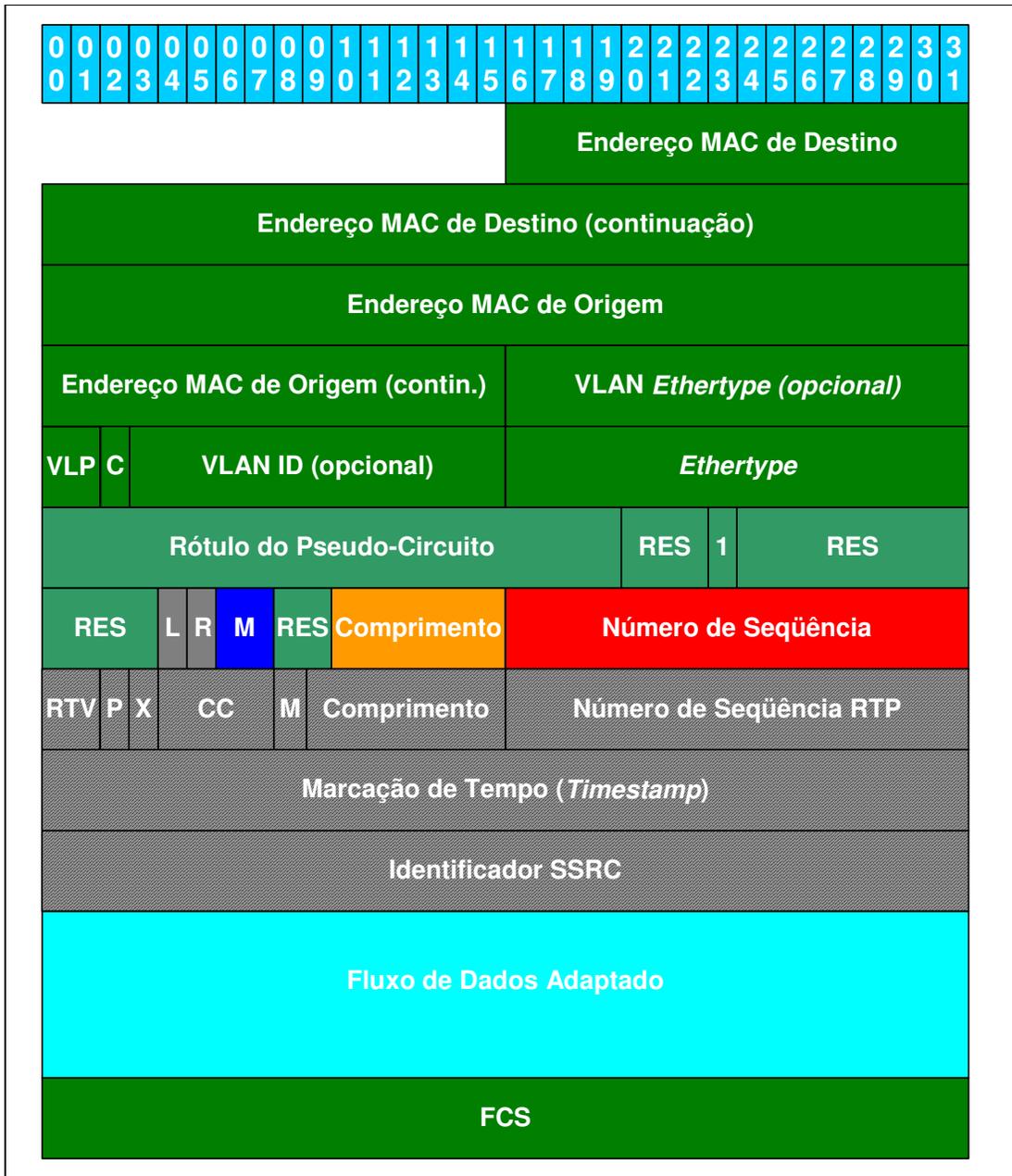


Figura C.5 - Formato de um pacote TDMoIP<sup>®</sup> sobre *Ethernet*.

As primeiras 5 linhas de 32 bits são o cabeçalho *Ethernet* (DIX), totalizando 18 octetos, podendo haver campos adicionais no caso do padrão IEEE 802.3; a sexta linha é um rótulo no formato da etiqueta MPLS; a sétima linha é a Palavra de Controle TDMoIP<sup>®</sup> de 4 octetos; as linhas 8, 9 e 10 são o cabeçalho opcional RTP, de 12 octetos; e as seguintes representam os dados TDM encapsulados utilizando AAL1 ou AAL2, com exceção da última linha, que corresponde à Seqüência de Verificação de Quadro (FCS) *Ethernet*, com 4 octetos. Segue a descrição dos campos ainda não abordados que formam esse pacote:

- **Endereço MAC de Destino**, 48 bits – é o endereço global único da interface de rede que deve receber o pacote, no PE de destino; seu formato é definido em IEEE802.3;
- **Endereço MAC de Origem**, 48 bits – é o endereço global único da interface de rede que originou o pacote, no PE de origem; seu formato é definido em IEEE802.3.
- **VLAN *Ethertype***, 16 bits – um valor 33024 (0x8100) nesse campo indica uma marcação de LAN virtual (VLAN) de acordo com o padrão IEEE 802.1.Q, e que os próximos dois octetos contém os campos VLP, C e VLAN ID, conforme definido em [IEEE802.1Q]. As marcações VLAN podem ser empilhadas, de forma que o campo de dois octetos que sucede o VLAN ID pode ser novamente um VLAN *Ethertype*.
- **VLP (*VLAN Priority*)**, 3 bits – indica a prioridade da VLAN, conforme definido pelo padrão IEEE 802.1.Q;
- **C (*Canonical Format Indicator*)**, 1 bit – quando ativado, indica que os descritores de rota estão presentes, conforme definido pelo padrão IEEE 802.1.Q;
- **VLAN ID (*VLAN Identifier*)**, 12 bits – identifica de forma unívoca a VLAN a que o quadro pertence. Caso seja definido como zero, apenas o campo VLP é significativo. Os valores de 1 até 4095 (0xFFFF) são reservados, os outros 4193 valores são válidos como identificadores VLAN.
- ***Ethertype***, 16 bits – identifica o protocolo utilizado, conforme valores alocados pelo IEEE. O valor alocado para *CES over Ethernet* é 35032 (0x88D8), mas também pode ser utilizado o 34887 (0x8847), correspondente ao valor *Ethertype* para o MPLS.
- **Rótulo do Pseudo-Circuito**, 20 bits – é o rótulo que identifica o pseudo-circuito estendido entre os PEs de origem e destino através da rede MPLS. Esse rótulo deve ser manualmente configurado e o restante da linha deve ser formatado para que seja semelhante a uma etiqueta MPLS.
- **FCS (*Frame Check Sequence*)**, 32 bits – é o campo contendo o código CRC para detecção de erros no quadro *Ethernet*, calculado conforme estabelecido pelo padrão IEEE 802.3 [IEEE802.3].

Cabe observar que quadros Ethernet podem ser utilizados para o transporte de pseudo-circuitos TDM sem necessidade das camadas IP ou MPLS; contudo, uma etiqueta no formato MPLS deve estar sempre presente, com seu cabeçalho de quatro octetos, o bit S=1 e todos os outros bits que não fazem parte da etiqueta reservados, sendo definidos como zero na direção *PSN-bound* e ignorados na direção *TDM-bound*.

## C.2 – SUBCAMADA DE ADAPTAÇÃO

A tecnologia SAToP [RFC4553], trata o pseudo-circuito como um fluxo contínuo e arbitrário de bits, não prevendo qualquer adaptação dos dados TDM e transportando, em cada pacote, apenas um número fixo de octetos recebidos do circuito de acesso TDM, a fim de facilitar o tratamento das perdas de pacotes na PSN. Esse tamanho do pacote é definido durante o estabelecimento do pseudo-circuito e precisa ser o mesmo em ambas as direções, permanecendo inalterado durante toda a conexão. As regras para a acomodação dos bits TDM dentro dos pacotes SAToP são as seguintes:

- A ordem dos octetos dentro do pacote é a mesma do circuito de acesso TDM;
- Os bits consecutivos provenientes do fluxo TDM preenchem cada octeto do pacote iniciando pelo bit mais significativo e seguindo até o bit menos significativo;
- Todas as implementações SAToP devem ser capazes de suportar os seguintes tamanhos de pacote para o fluxo de dados:
  - E1: 256 octetos;
  - T1: 192 octetos;
  - E3/T3: 1024 octetos.

A tecnologia TDMoIP<sup>®</sup> [IETF2006] pode utilizar diferentes técnicas de adaptação, dependendo das características do tráfego TDM. Sempre que possível, devem ser utilizados mecanismos de adaptação originalmente desenvolvidos para ATM, o que traz como benefício a interoperabilidade com serviços de emulação de circuitos transportados sobre redes ATM. Essa foi uma escolha natural no desenvolvimento da tecnologia, pois os mecanismos da família de protocolos AAL (*ATM Adaptation Layer*), desenvolvidos originalmente para adaptar os vários tipos de dados ao rígido formato do ATM, são soluções gerais para o problema de transporte de fluxos de dados em tempo real, a taxas constantes ou variáveis, sobre uma rede comutada em modo pacotes.

Para os troncos TDM tradicionais, estaticamente alocados e com taxa de transmissão de bits constante, ou CBR (*Constant Bit-Rate*), é utilizada uma camada de adaptação AAL1 (*ATM Adaptation Layer number 1*), desenvolvida para o transporte de serviços CBR sobre redes ATM. Essa camada realiza a segmentação do fluxo contínuo de bits TDM em pequenas células de 48 octetos, inserindo informações de seqüenciamento, temporização, recuperação de erros e sincronização. Por exemplo, caso o fluxo TDM seja composto por

um feixe E1 com sinalização por canal associado (CAS), a camada AAL1 insere um ponteiro no começo do próximo superquadro, de forma que, mesmo se algumas células forem perdidas, o ponteiro permite a recuperação da sinalização no próximo superquadro.

Essa técnica permite a concatenação de qualquer número de células AAL1 em um pacote, de modo a permitir flexibilidade no compromisso entre o atraso de empacotamento, que diminui com a redução de células por pacote, e a eficiência de utilização da largura de banda, que aumenta com seu incremento, em virtude da redução do *overhead*.

Para enlaces TDM dinamicamente alocados, onde a taxa de transmissão de bits varia com a ativação de *timeslots* e/ou mecanismos de detecção de atividade de voz, a fim de evitar a transmissão de canais ociosos e economizar largura de banda, é utilizada uma camada de adaptação AAL2 (*ATM Adaptation Layer number 2*), desenvolvida para o transporte de serviços de taxa variável, ou VBR (*Variable Bit-Rate*) sobre redes ATM. Essa camada acomoda cada *timeslot* TDM em pequenas mini-células, inserindo informações de identificação, definição de comprimento e seqüenciamento, enviando somente aquelas que carregam informações válidas, dentro de um único pacote. Para *timeslots* carregando informações HDLC, como aqueles utilizados na sinalização por canal comum (CCS), uma adaptação especial é provida a fim de assegurar o seu encapsulamento direto.

#### **a) Formato AAL1**

Para situações onde é predominante o uso de TDM não canalizado, ou TDM canalizado com alocação de canais estática, conhecida como “emulação de circuitos”, o fluxo de dados pode ser encapsulado de forma eficiente utilizando AAL1, desenvolvida para fluxos com taxa de bits constante (CBR), cujas características fundamentais são aqui apresentadas.

O primeiro octeto de uma PDU AAL1 contém um número de seqüência de três bits, protegido contra erros, como apresentado na Figura C.6.



fixo de pacotes, mas pode ser desejável quando da interoperabilidade com circuitos emulados ATM que a utilizem, ou para permitir mecanismos de recuperação de relógio baseados na AAL1.

Para o transporte consciente de estrutura, a PDU para emulação estruturada demonstra-se adequada para circuitos TDM canalizados de acordo com a Recomendação ITU-T G.704 [ITU-T G704], em particular quando estão envolvidos circuitos E1/T1 fracionários. A AAL1 estruturada considera os dados não apenas como um fluxo contínuo de bits, mas como um arranjo de canais. Os N octetos dos N canais a transportar são primeiro ordenados pelo número do canal; assim, se os canais 2, 3, 5, 7 e 11 devem ser transportados, seus cinco octetos correspondentes são colocados na PDU imediatamente após o número de seqüência, sendo esse procedimento repetido até que todos os 47 octetos estejam ocupados, como mostrado na Figura C.7(a). A próxima PDU é iniciada a partir do próximo canal dessa seqüência que ficou de fora, como mostrado na Figura C.7(b). e assim por diante. O conjunto de canais 2, 3, 5, 7 e 11 constitui a estrutura básica e o ponto onde essa estrutura termina e a próxima inicia é o limite da estrutura do fluxo TDM. O problema com esse arranjo é a ausência de indicação explícita da identidade de cada octeto, pois cada PDU AAL1 é iniciada com um canal diferente, de forma que uma simples perda de pacote resulta no embaralhamento dos canais, sem possibilidade de recuperação.

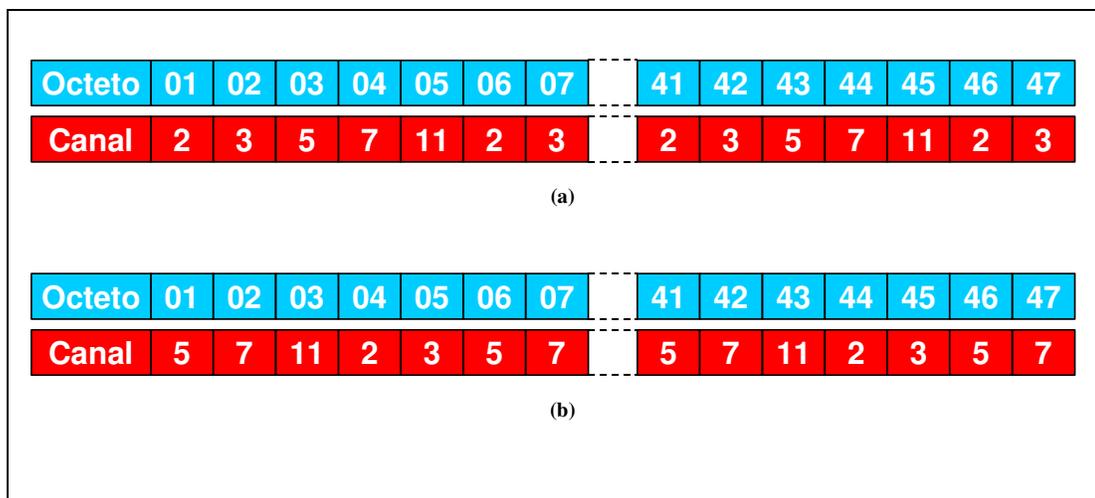


Figura C.7 - Exemplo de alocação dos canais TDM nos octetos das PDUs AAL1 para emulação de circuitos estruturada.

A solução para o problema é a introdução periódica de um ponteiro para o próximo limite da estrutura, que não precisa ser muito freqüente, pois a identificação de cada canal é unívoca a menos que pacotes sejam perdidos. Assim, a sincronização de quadro TDM é mantida pela inserção desse ponteiro no cabeçalho da PDU AAL1, indicando o início do próximo quadro, uma vez a cada ciclo de 8 PDUs. Esse ponteiro possui 7 bits, suficientes para representar um deslocamento de 128 posições, e é protegido por um bit de paridade par no bit mais significativo do respectivo octeto. Dessa forma, os ponteiros podem ser encaminhados somente nas PDUs de número de seqüência par, já que o mesmo pode indicar qualquer octeto entre os 46 da PDU que o contém e os 47 da próxima, totalizando 93 posições possíveis. Essas PDUs são denominadas *P-format* PDUs e seu formato é apresentado na Figura C.8.

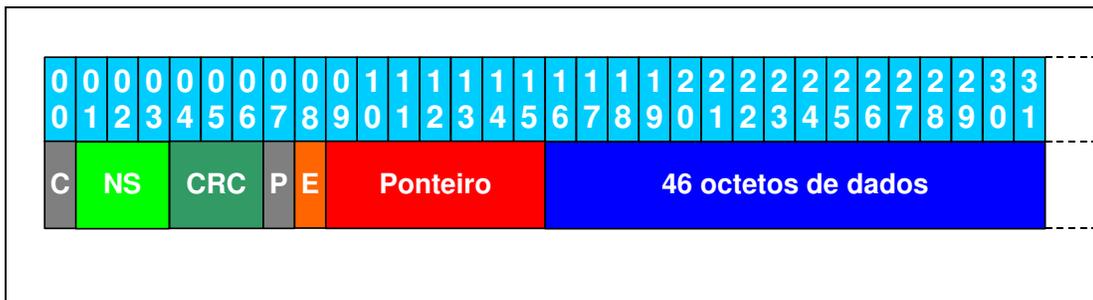


Figura C.8 - Estrutura de uma *P-format* PDU AAL1 para emulação de circuitos estruturada.

- **C (Indicação de Convergência)**, 1 bit – C=1, para *P-format* PDUs;
- **NS (Número de Seqüência)**, 3 bits – é o número de seqüência AAL1, incrementado a cada PDU, formando um ciclo de 8 PDUs;
- **CRC (Código de Redundância Cíclica)**, 3 bits – é o *checksum* para C+SN (4 bits);
- **P (Paridade)**, 1 bit – é o bit de paridade par do octeto formado por C+SN+CRC.
- **E (Paridade)**, 1 bit – é o bit de paridade par do octeto que contém o ponteiro.
- **Ponteiro**, 7 bits – é o ponteiro para o próximo limite da estrutura do fluxo TDM.

Se o Ponteiro é igual a zero, significa que a estrutura se inicia no próximo octeto, isto é, o próximo octeto de dados pertence ao canal de numeração mais baixa (no exemplo anterior, o canal 2); ponteiro igual a 93 significa que o último octeto da próxima PDU é o último da estrutura, e a PDU que o segue inicia uma nova estrutura. P valor especial 127 indica que não existe limite a ser indicado nesse ciclo, necessário quando estruturas muito grandes estão sendo transportadas, cujos limites se estendem além das 8 PDUs do ciclo. A *P-format*

PDU é sempre colocada na primeira posição possível do ciclo de 8 PDUs (indicado pelo número de seqüência) onde o limite da estrutura ocorre, e somente pode ocorrer uma vez a cada ciclo.

A PDU de emulação estruturada com CAS é similar à anterior, tendo como diferença apenas a definição de estrutura: enquanto naquela a estrutura é composta apenas pelos N canais, nesta a estrutura engloba o multi-quadro composto pela múltipla repetição dos N canais e também seus respectivos bits de sinalização CAS. Os 4 bits CAS de cada canal são agrupados em octetos e o octeto final é completado com zeros, se necessário, fechando assim a estrutura. Assim, a sincronização de quadro TDM é mantida pela inserção de ponteiros no cabeçalho da PDU AAL1, indicando o início do próximo multi-quadro, além da transmissão de uma sub-estrutura incluindo os bits CAS dos respectivos canais.

Por exemplo, em circuitos E1 os bits de sinalização CAS são atualizados a cada multi-quadro de 16 quadros. Assim, uma estrutura N x 64kbps derivada de um circuito E1 utilizando sinalização CAS consiste de 16 repetições dos N octetos correspondentes a cada canal, seguida por N conjuntos com os 4 bits de sinalização ABCD de cada canal, e finalmente 4 zeros se N for ímpar, conforme indicado na Figura C.9, assumindo a transmissão dos canais 2, 3 e 5.

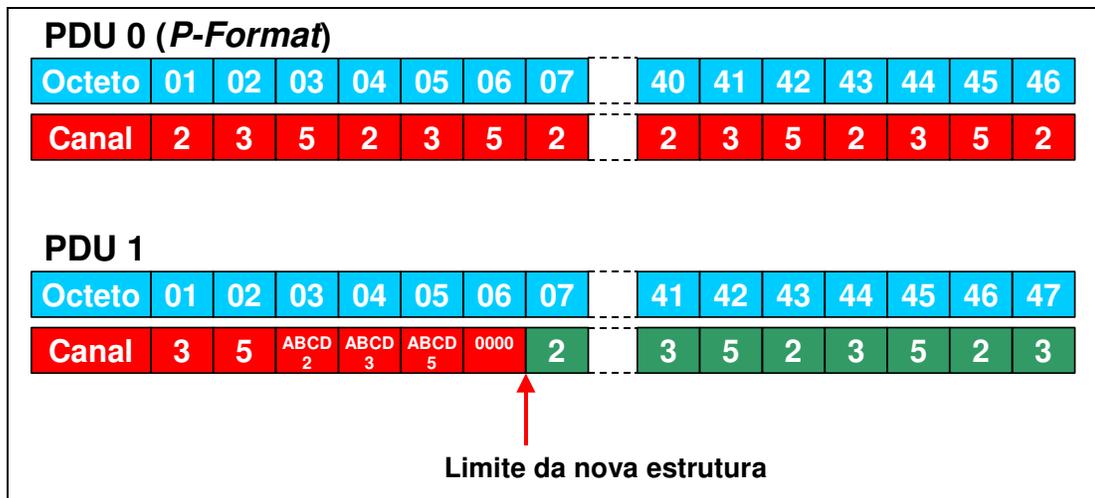


Figura C.9 - Exemplo de alocação de canais nos octetos das PDUs AAL1.

No modo AAL1, o fluxo de dados TDMoIP<sup>®</sup> consiste em pelo menos uma PDU de 48 octetos encapsulada dentro de cada pacote, conforme mostrado na Figura C.10. O número de PDUs deve ser pré-configurado e escolhido de forma que o tamanho total do pacote não exceda a MTU da rede. Para redes *Ethernet*, onde o MTU é de 1500 octetos, 30 é o limite. O número exato de PDUs por pacote é escolhido considerando as restrições de latência e largura de banda existentes, pois uma única PDU oferece a mínima latência, mas implica em elevado *overhead*. Todas as implementações TDMoIP<sup>®</sup> devem suportar entre 1 e 8 PDUs por pacote para circuitos E1 e T1, e entre 5 e 15 PDUs para circuitos E3 e T3.



Figura C.10 - PDUs AAL1 dentro dos pacotes TDMoIP<sup>®</sup>.

## b) Formato AAL2

Apesar do formato AAL1 poder ser configurado para o transporte de circuitos E1 e T1 fracionários ( $N \times 64\text{kbps}$ ), a alocação dos canais transportados precisa ser estática, uma vez que AAL1 transporta apenas fluxos a taxas de bit constantes (CBR). Contudo, pode ser freqüente a situação onde nem todos os canais se encontram ativos simultaneamente, o que é detectado pela observação da sinalização dos canais TDM; e mesmo em canais ativos cerca de metade do tempo de conversação é ocupado por silêncio, que pode ser identificado através técnicas de detecção de atividade de voz. Nesses casos, o fluxo de dados é encapsulado de forma mais eficiente utilizando a adaptação AAL2, desenvolvida para fluxos de dados com taxa de bits variável (VBR), que permite a alocação dinâmica dos canais efetivamente utilizados, conhecida como “emulação de enlaces”, conservando largura de banda.

No modo AAL2, o fluxo de dados TDMoIP<sup>®</sup> consiste em uma ou mais PDUs AAL2 de tamanho variável. Cada PDU AAL2 contém três octetos de *overhead* e entre 1 e 64 octetos de dados. Um pacote pode ser construído através da inserção de PDUs correspondentes aos canais ativos, adicionando todas as PDUs que estiverem completas num certo instante, ou utilizando qualquer outra técnica semelhante, de forma que mais de uma PDU pertencente a um único canal pode estar contida em um dado pacote.

Como a proposta do trabalho é a emulação de circuitos TDM e o transporte transparente de sinais de vídeo digital através dos mesmos, não é necessário descrever adaptação AAL2.

### **c) Formato HDLC**

A motivação para tratamento de HDLC em TDMoIP<sup>®</sup> é o transporte eficiente de sinalização por canal comum (CCS), como SS7 ou a utilizada em RDSI-PRI, quando inserida no fluxo TDM. O mecanismo proposto não é genérico para todos os dados HDLC, e assume que as mensagens transportadas são sempre menores que o tamanho máximo do pacote.

O modo HDLC deve ser usado somente quando é esperado que a maioria da banda utilizada pelo fluxo HDLC seja ocupada por bits ociosos, caso contrário o canal CCS deve ser tratado como um canal convencional. Esse modo é baseado na existência de um pseudo-circuito em separado, para transporte transparente dos dados e mensagens de controle associados à CCS:

O PE de origem (*PSN-bound*) monitora os *flags* até que um quadro seja detectado. O conteúdo desse quadro é coletado e tem seu FCS testado; se o FCS é incorreto, o quadro é descartado, caso contrário é enviado através do pseudo-circuito em separado, tendo seu FCS descartado e os bits ociosos removidos para economia de banda. Quando esse quadro é recebido pelo PE de destino (*TDM-bound*), seu FCS é recalculado e o quadro original CCS é reconstruído.

### C.3 – SERVIÇOS TDM EMULADOS

Os serviços TDM que podem ser emulados através da Arquitetura PWE3, segundo a RFC4197, são os seguintes:

a) SAT (Transporte Não-consciente de Estrutura) para Sinais da Hierarquia PDH:

O transporte sem consciência de estrutura é pressuposto para os seguintes sinais:

- E1 como descrito na Recomendação ITU-T G.704;
- T1 (ou DS1) como descrito na Recomendação ITU-T G.704;
- E3 como descrito na Recomendação ITU-T G.751;
- T3 (ou DS3) como descrito na Recomendação ANSI T1.107;

b) Transporte Consciente de Estrutura para Sinais da Hierarquia PDH:

O transporte com consciência de estrutura é pressuposto para os seguintes sinais:

- E1 com estruturas de quadro, como descrito na Recomendação ITU-T G.704;
- NxDS0 (64 kbps), como ou sem sinalização CAS;

c) Transporte Consciente de Estrutura de Circuitos da Hierarquia SDH/SONET:

O transporte com consciência de estrutura é pressuposto para os seguintes circuitos SONET STS (*Synchronous Transport Signal*) e VT (*Virtual Tributaries*) ou SDH VCs (*Virtual Containers*), não fazendo grande sentido o transporte SAT para SDH/SONET:

- SONET STS-1 SPE (*Synchronous Payload Envelope*) ou SDH VC-3;
- SONET STS-Nc (N=3,12,48,192) ou SDH VC-4, VC-4-4c, VC-4-16c e VC-4-64c;
- SONET VT-N (N=1,5, 2, 3, 6) ou SDH VC-11, VC-12, VC-2;
- SONET NxVT-N ou SDH NxVC-11, NxVC-12, NxVC-2, NxVC-3.

Revisando as duas hierarquias:

**a) Hierarquia PDH:**

A Hierarquia Digital Plesiócrona (PDH) apresenta diversidade de padronização. As taxas de transmissão tradicionalmente utilizadas são detalhadas na Recomendação ITU-T G.702 [ITU-T G702] e mostradas na Figura C.11. Na América do Norte, os fluxos de bits designados como T1 (1,544 Mbps) e T3 (44,736 Mbps) são mandatórios, enquanto no Brasil e na Europa os fluxos de bits designados como E1 (2,048 Mbps) e E3 (34,368 Mbps) são os mais utilizados.

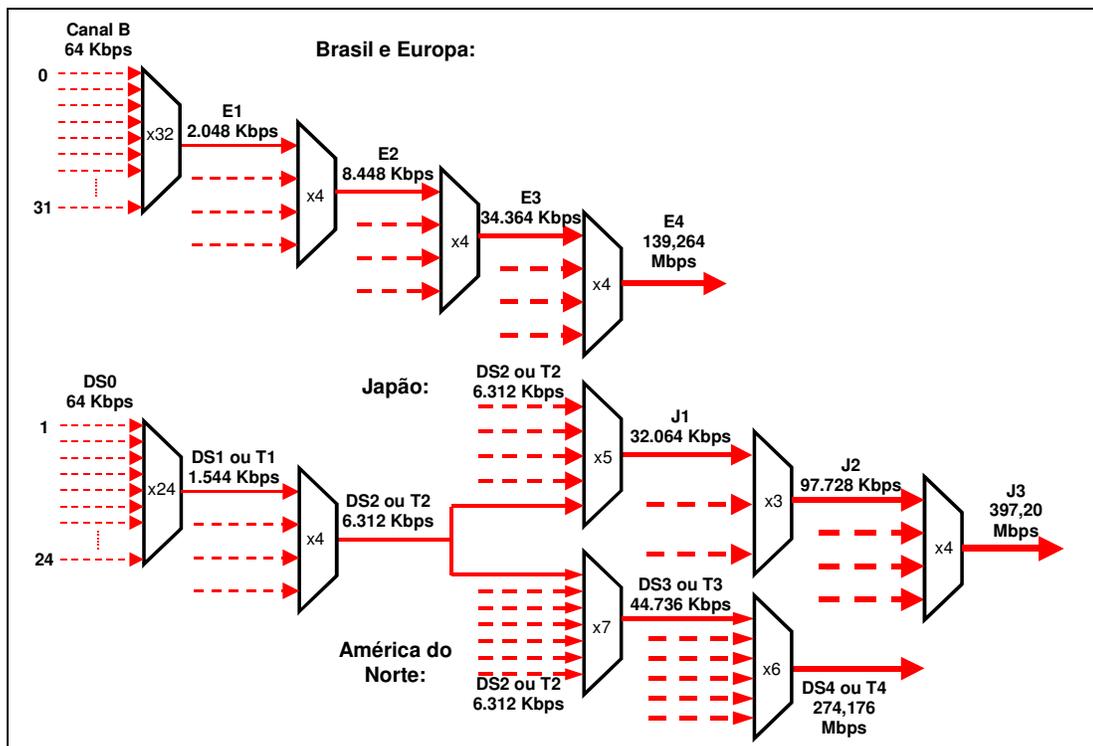


Figura C.11 - Hierarquia digital plesiócrona.

Apesar dos circuitos TDM poderem ser utilizados para transportar qualquer fluxo de bits com as taxas definidas na Recomendação G.702, existe um método padronizado para transporte desses fluxos em unidades chamadas quadros, contendo sempre o mesmo número de bits. Em função da frequência de amostragem de 8 kHz para o tráfego de voz, com base no Teorema de Shannon, a taxa de bits é sempre um múltiplo de 8.000, de forma que um quadro T1 consiste em 193 bits e um quadro E1 é formado por 256 bits. Nos fluxos TDM sem enquadramento, ou não-estruturados, todos esses bits estão disponíveis para o fluxo de dados.

Quando existe enquadramento, o mesmo é imposto pela introdução de um padrão periódico dentro do fluxo de bits para identificação dos limites do quadro. Esse padrão é formado por um bit para quadros T1 e oito bits para quadros E1, maiores detalhes sobre a geração e utilização desses bits são encontrados nas Recomendações ITU-T G.704 [ITU-T G704] , G.706 [ITU-T G706] e G.751 [ITU-T G751] e no Padrão ANSI T1.107 [ANSI107].

Os fluxos TDM com enquadramento são utilizados para multiplexar canais, usualmente voz com 8.000 amostras de 8 bits por segundo – 64 kbps, em uma seqüência de *timeslots*

recorrentes, sempre na mesma ordem dentro de cada quadro. Esse arranjo é chamado “TDM canalizado” e introduz uma estrutura adicional. Em alguns casos, são também definidos grupos de quadros consecutivos, os superquadros, impondo mais um nível na estrutura.

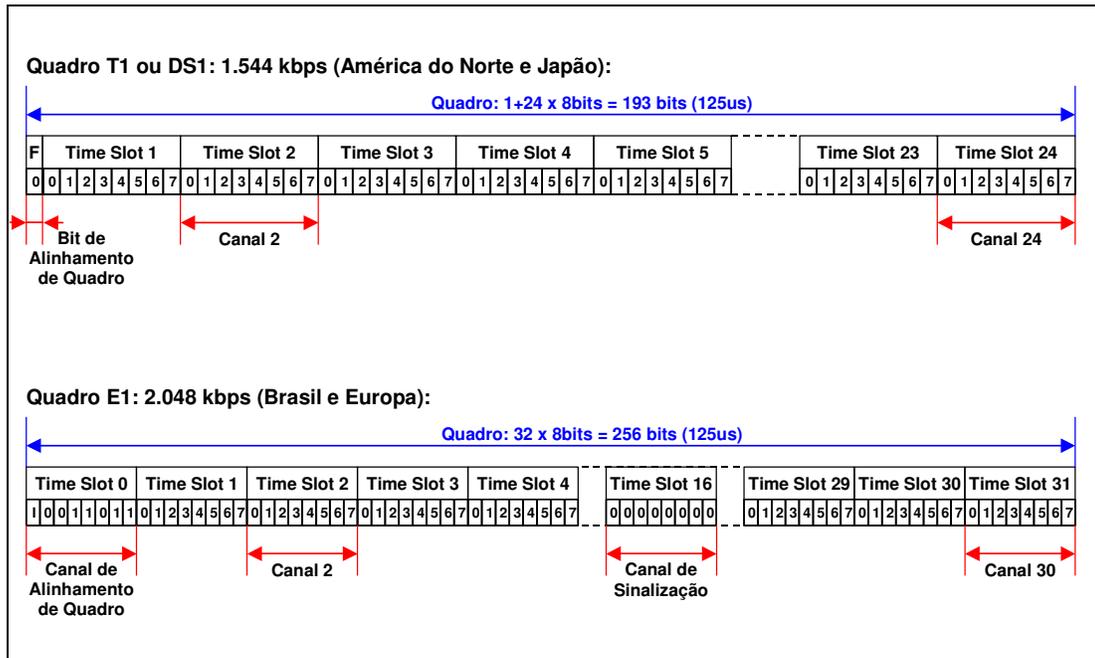


Figura C.12 - Fluxo TDM estruturado com canalização.

Em relação ao transporte TDM dentro de pseudo-circuitos, existem duas abordagens:

- **SAT (Structure-Agnostic Transport)** – é o transporte de circuitos TDM não-estruturados, ou de circuitos TDM estruturados quando a estrutura é considerada irrelevante do ponto de vista do transporte. Nesse transporte não-consciente de estrutura, qualquer *overhead* estrutural que esteja presente no fluxo de dados é transportado de forma transparente, e o encapsulamento não provê nenhum mecanismo para sua localização ou utilização.
- **Structure-Aware Transport** – é o transporte de circuitos TDM estruturados, levando em consideração pelo menos algum dos níveis da estrutura. Nesse transporte consciente de estrutura, não existe garantia de que todos os bits do fluxo TDM serão transportados através da rede de pacotes (especificamente, os bits de sincronização e *overhead* relacionado podem ser eliminados na origem e regenerados no destino) ou que os bits transportados estarão localizados no pacote em sua ordem original dentro do fluxo (nesse caso, a ordem dos bits é geralmente recuperada no destino).

**b) Hierarquia SDH/SONET:**

O termo SONET se refere à Rede Óptica Síncrona utilizada na América do Norte e especificada pelo Padrão ANSI T1.105 [ANSI105], baseada no conceito de um contêiner de dados de Nx783 octetos, repetido a cada 125us. Esse fluxo básico de dados se refere a um STS-1 (*Synchronous Transport Signal – level 1*), comumente utilizado para transportar circuitos T3 ou E3, mas podendo ser concatenado em circuitos de maior ordem e largura de banda, como STS-Nc (*Synchronous Transport Signal – level N, concatenado*), utilizados para transportar qualquer tipo de dados, como células ATM e sinais de vídeo digital; ou subdividido em circuitos de menor ordem, denominados tributários virtuais ou VTs (*Virtual Tributaries*), comumente utilizados para transportar circuitos T1 ou E1.

A Hierarquia Digital Síncrona (SDH) é o equivalente internacional aprimorado da rede SONET, como tentativa de padronização das hierarquias digitais existentes em um esquema único de multiplexação, especificada pela Recomendação ITU-T G.707 e mostrada na Figura C.13. Maiores informações sobre essas hierarquias podem ser encontradas em [SOARES1988], ou nos documentos normativos citados.

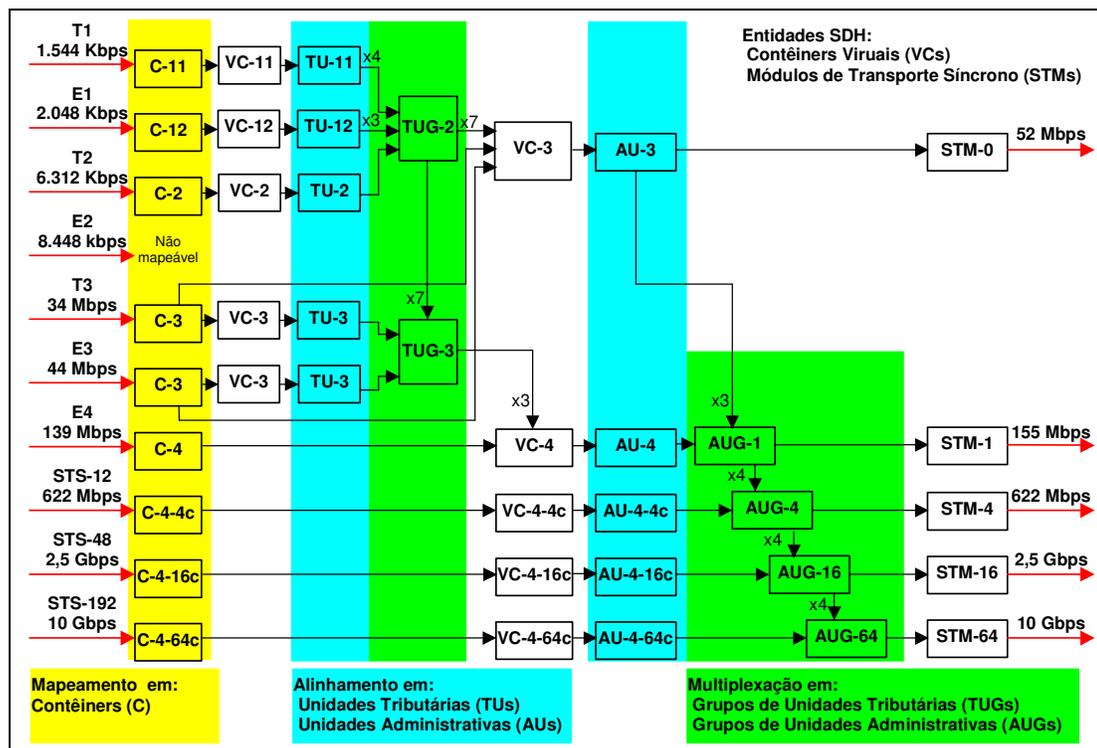


Figura C.13 - Hierarquia digital síncrona.

As redes SDH/SONET têm por característica a inclusão de uma quantidade substancial de *overhead*, utilizado para monitoração de performance, isolamento de defeitos e outras funções de manutenção ao longo dos diferentes enlaces ópticos ou elétricos. Isso inclui também um mecanismo de ponteiros para o transporte de dados de forma assíncrona. Adicionalmente, cada contêiner inclui *overhead* para monitoramento da performance fim a fim, isolamento de defeitos e manutenção para o serviço que está sendo transportado. Caso esse contêiner seja subdividido em circuitos de menor taxa, como T1/E1, novo *overhead* é incluído para monitoramento fim a fim dos circuitos T1/E1 individuais.

#### **C.4 – GERENCIAMENTO DE FALHAS**

As redes TDM nativas representam as falhas detectadas através das indicações AIS (para falha local) e RDI (para falha remota), que são padrões especiais inseridos no fluxo contínuo de bits. Os mecanismos de transporte sem consciência de estrutura, como o SAToP, carregam essas indicações de forma transparente, mas para mecanismos conscientes da estrutura, onde o PE de origem pode remover o *overhead* do fluxo TDM, incluindo essas indicações, é necessária uma sinalização explícita das condições de defeito nos circuitos TDM nativos, em ambas as extremidades.

Essa sinalização é transportada através dos *flags* existentes na Palavra de Controle, contida em cada um dos pacotes transmitidos através da PSN, o que de certa forma simula o comportamento dos mecanismos nativos de Operação e Manutenção dos circuitos TDM, baseados em padrões de bits inseridos no fluxo. Estes *flags* foram projetados para o envio de mensagens urgentes, ou seja, mensagens cujo conteúdo não pode ser significativamente atrasado em relação aos dados TDM que são potencialmente impactados pelas mesmas.

A operação do tratamento de falhas para TDMoIP<sup>®</sup> é demonstrada na Figura C.14, considerando um fluxo TDM sendo transmitido do Equipamento Cliente 1 (CE1) para o Equipamento Cliente 2 (CE2), utilizando *gateways* TDMoIP<sup>®</sup> (IWF1 e IWF2), que constituem os Equipamentos Provedores (PE1 e PE2), através de uma rede comutada em modo pacotes (PSN). O tratamento das falhas ocorridas nos circuitos TDM1, TDM2 ou na PSN não deve restringir-se apenas à sua detecção, mas também permitir a sua correta localização, gerando alarmes somente dentro da rede afetada.

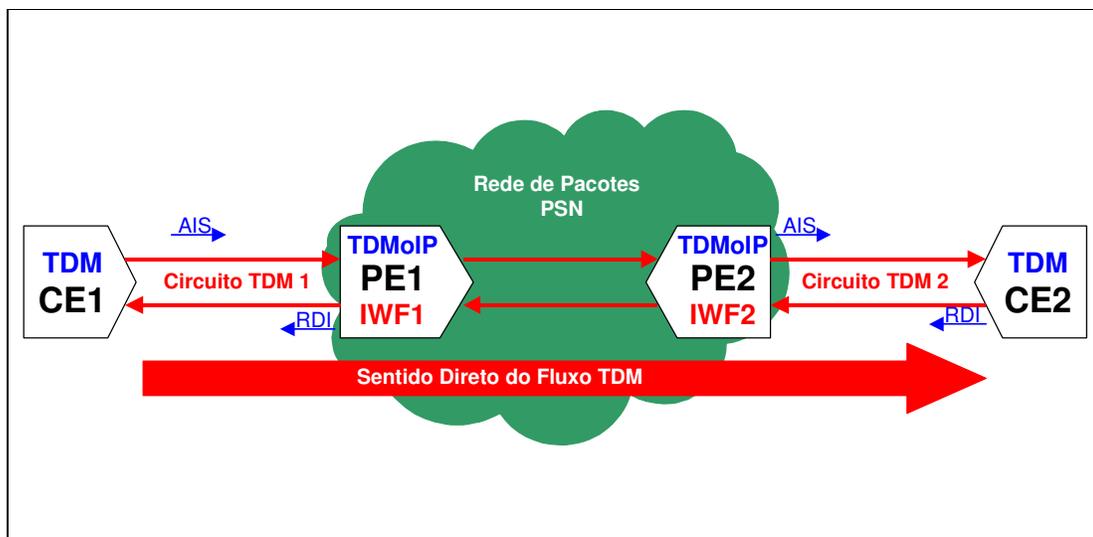


Figura C.14 - Configuração típica de uma rede TDMoIP®.

Nesse contexto, dependendo das características atribuídas ao mesmo, podem existir dois cenários de Operação e Manutenção para o pseudo-circuito TDM entre CE1 e CE2: terminação local e terminação estendida.

No cenário de terminação local, somente os dados válidos são transmitidos entre os Circuitos TDM1 e TDM2, os *gateways* IWF1 e IWF2 são considerados como terminação de cada circuito, e nenhuma falha detectada no Circuito TDM1 é transmitida ao Circuito TDM2, ou o contrário. Assim, os sinais AIS não devem ser propagados através da PSN.

No cenário de terminação estendida, caso ocorra uma falha no Circuito TDM1, como perda total de sinal ou perda de sincronismo de quadro, em localização distinta do último enlace até PE1, o nó TDM de destino do enlace em falha irá gerar um AIS em direção a PE1; caso a falha ocorra no último enlace, o próprio PE1 irá detectar o problema. Em ambos os casos, o *gateway* IWF1 ativa o *flag* de Falha Local (L) na Palavra de Controle TDMoIP® dos pacotes que estão sendo encaminhados através da rede PSN, o que estende a informação da falha até o Circuito TDM2.

Enquanto as indicações diretas de falha, ocorridas na direção do fluxo TDM, podem ser geradas por quaisquer elementos da rede que as detecte, as indicações reversas são geradas somente pelos equipamentos terminais. Assim, em um cenário de terminação local, o *gateway* IWF1 representa uma terminação para o Circuito TDM1, e quando detecta a perda de validade do sinal TDM ou recebe informações de problemas em outro ponto da rede,

pode gerar um sinal TDM RDI em direção ao Equipamento Cliente CE1. Contudo, se o cenário for de terminação estendida, o *gateway* IWF1 não é uma terminação para o fluxo TDM e portanto não deve gerar nenhum sinal RDI, ativando o *flag* L ao invés disso. Esse *flag*, ao ser recebido pelo IWF2, provoca a geração de um sinal AIS no fluxo TDM em direção ao Equipamento Cliente CE2, que é a terminação efetiva nesse cenário, o qual pode então gerar o sinal TDM RDI no fluxo reverso.

Quando o *flag* L é ativado, o pacote pode ser tratado de quatro formas:

- a forma padrão é o preenchimento, pelo IWF1 do conteúdo com uma quantidade apropriada de dados correspondente a um sinal TDM AIS (usualmente, todos os bits iguais a “1”) ou o simples encaminhamento dos dados TDM referentes ao AIS recebido;
- alternativamente, para o transporte consciente de estrutura de TDM canalizado, o conteúdo deve ser preenchido com uma mensagem de “condicionamento de tronco”, que envolve um padrão de dados previamente configurado como “fora de serviço”, sobretudo quando estiver sendo utilizada a agregação canais provenientes de diversos troncos TDM tradicionais num único pseudo-circuito TDM;
- a terceira possibilidade é simplesmente suprimir o conteúdo para esses pacotes, economizando largura de banda na rede de pacotes;
- finalmente, caso o IWF1 presume que o impacto da falha sobre os dados é reduzido ou passível de correção, como uma perda de sincronismo de multi-quadro, o conteúdo pode ser preenchido com os dados TDM recebidos, indicando através do campo M que os dados estão corrompidos, mas são potencialmente recuperáveis.

Quando o IWF2 recebe uma indicação de falha local sem o campo M, ele encaminha (ou gera, se os dados TDM tiverem sido suprimidos do pacote) um sinal AIS ou de condicionamento de tronco, se assim previamente configurado, em direção ao CE2, emulando o que aconteceria numa rede TDM convencional. Essa situação indica exclusivamente que uma falha ocorreu no Circuito TDM1, estando o restante da rede em perfeitas condições operacionais.

Se o campo M indica que os dados TDM são potencialmente recuperáveis, algoritmos específicos podem ser utilizados para minimizar os impactos da falha sobre o pseudo-circuito. Caso esse campo indique que os dados TDM são ociosos, nenhum alarme deve ser

gerado e o IWF2 pode tratar o conteúdo do pacote da mesma maneira que os demais dados TDM, gerando, se necessário, um sinal de condicionamento de tronco ao invés de um AIS.

Quando ocorre uma falha no Circuito TDM2, é gerado um sinal TDM AIS em direção a CE2, que pode ser respondido pelo envio de um sinal TDM RDI na direção reversa do fluxo. No cenário de terminação local esse sinal é restrito ao Circuito TDM2, enquanto no cenário de terminação estendida o IWF2, detectando o sinal RDI inserido como dado TDM válido, deve ativar o *flag* de Falha Remota (R) na Palavra de Controle TDMoIP<sup>®</sup> dos pacotes enviados de volta ao IWF1 através da rede de pacotes. O IWF1, recebendo essa indicação, gera um sinal TDM RDI em direção a CE1, emulando o comportamento de uma rede TDM convencional.

A última situação possível é a ocorrência de falha unidirecional na rede de pacotes, caso em que o IWF1 encaminha os pacotes em direção ao IWF2, mas esses não são recebidos. Nesse caso, o IWF2 deve alarmar o problema para a gerência da PSN, gerando sinais TDM AIS em direção a CE2. Este, por sua vez, pode responder com um sinal TDM RDI, que seria propagado ao Circuito TDM1 através da ativação, pelo IWF2, do *flag* R no cenário de terminação estendida. Uma vez recebido esse *flag* pelo IWF1, o mesmo deve gerar um sinal TDM RDI em direção a CE1.

Em todos esses casos, se a condição de falha persiste por um dado intervalo de tempo, cujo valor padrão é 2,5s, uma falha no serviço é declarada. Como os pseudo-circuitos são inerentemente bidirecionais, uma falha dessas em qualquer das direções resulta em falha bidirecional, e o pseudo-circuito é desconectado, passando ambos os *gateways* TDMoIP<sup>®</sup> a encaminhar sinais AIS para seus respectivos circuitos TDM, até que a conectividade seja re-estabelecida pelo estabelecimento de um novo pseudo-circuito.



## APÊNDICE D – PROCESSAMENTO DO NÚMERO DE SEQUÊNCIA

O algoritmo para processamento do número de seqüência para aplicações TDMoIP® apresentado em [IETF2006] é descrito a seguir, através da tradução e adaptação integral do respectivo apêndice A. Esse algoritmo não especifica o gerenciamento do *buffer* de reprodução utilizado, mas sua análise permite deduzir que isso é realizado por um outro processo externo executado de forma assíncrona, uma vez que esse gerenciamento é fundamental para a operação do algoritmo descrito, pois são utilizados dois *flags* externos indicativos de esgotamento e esvaziamento do *buffer*, e seriam necessários ponteiros para indicar as posições de escrita e leitura dos pacotes que estão sendo recebidos ou reproduzidos, também não indicados.

O campo número de seqüência na Palavra de Controle permite a detecção de pacotes perdidos e recebidos fora de ordem. Aqui é fornecido o pseudo-código de um algoritmo de exemplo para seu tratamento, a fim de esclarecer as questões envolvidas. Essas questões são específicas de cada implementação e nenhuma explicação simples pode apreender todas as possibilidades envolvidas.

A fim de simplificar a descrição, a aritmética em módulo (resto da divisão entre inteiros) é consistentemente utilizada no tratamento da ciclicidade. Todas as diferenças entre índices são explicitamente convertidas para o intervalo  $[-2^{15} \dots + 2^{15} - 1]$  com o objetivo de assegurar que a simples verificação do sinal da diferença permita a previsão correta da ordem de chegada dos pacotes

Além disso, é introduzida a noção de um *buffer* de reprodução, a fim de definir sem ambigüidade os pacotes que chegam atrasados. Quando um pacote chega, após ter sido previamente assumido como perdido, o *gateway* de destino pode descartá-lo e continuar tratando-o como perdido. Alternativamente, se o dados de preenchimento que foi inserido em seu lugar ainda não foi reproduzido no fluxo de saída, existe a opção de inserir o dado real no *buffer* de reprodução. Evidentemente, o dado de preenchimento pode ser gerado no momento da detecção inicial de um pacote perdido ou em qualquer instante até a sua

reprodução. A descrição anterior pressupõe um *buffer* de reprodução orientado a pacotes, ao invés de orientado a octetos TDM; contudo, este não é um requisito real para as implementações de reordenação, uma vez que o segundo poderia ser utilizado sem problemas, considerando ponteiros indicando o início de cada pacote dentro do fluxo TDM.

Uma vez introduzido o buffer de recepção, serão tratadas explicitamente as condições de esgotamento e esvaziamento desse buffer. O esgotamento (*overflow* ou *over-run*) ocorre quando os pacotes chegam tão rapidamente que não podem ser armazenados para posterior reprodução. Isso é usualmente uma indicação de imprecisão grosseira de sincronismo ou configuração incorreta. O esvaziamento (*underflow* ou *under-run*), é usualmente um sinal de paralização da rede, como resultado de congestionamento ou falha.

// Variáveis externas utilizadas no pseudo-código:

Received	Inteiro 16 bits ; número de seqüência do pacote recebido
Played	Inteiro 16 bits ; número de seqüência do pacote sendo reproduzido (Nota 1)
Filler	Pacote ; dados de preenchimento para pacotes perdidos (“1s”)
OverRun	Booleano ; o <i>buffer</i> de reprodução está cheio (Nota 3)
UnderRun	Booleano ; o <i>buffer</i> de reprodução está vazio (Nota 3)

// Variáveis internas utilizadas no pseudo-código:

Expected	Inteiro 16 bits ; próximo número de seqüência esperado
D	Inteiro 16 bits ; diferença entre esperado e recebido (Nota 2)
L	Inteiro 16 bits ; diferença entre os números de seqüência do pacote que está sendo reproduzido e aquele que acabou de ser recebido ; (Notas 1 e 2)

// Parâmetro do algoritmo:

R	; máximo atraso recuperável para um pacote ; se R=0, o pacote que chega atrasado é sempre descartado.
---	---

Nota 1: Somente requeridos quando existe reordenação (opcional)

Nota 2: Estes números estão sempre no intervalo  $[-2^{15} \dots + 2^{15} - 1]$

Nota 3: O *buffer* de reprodução é esvaziado pelo processo de regeneração do fluxo TDM, que é executado de forma assíncrona com o processo de chegada de pacotes, não especificado nesse documento.

```

// Algoritmo para Processamento do Número de Sequência:
ENQUANTO AplicacaoAtiva
    Aguarda(RecebimentoPacote) ; aguarda recebimento do próximo pacote
    SE Received = Expected ; trata pacote como em ordem
        IF NOT OverRun ENTAO
            GRAVA(Pacote) ; coloca conteúdo do pacote no buffer
        SENAO
            DESCARTA(Pacote) ; despreza o pacote
            Expected = (Received + 1) mod 2^16
    SENAO ; trata pacote como fora de ordem
        D = ((Expected - Received) mod 2^16) - 2^15
        SE D > 0 ENTAO ; pacotes Expected, Expected+1, ...,
            ; Received-1 foram perdidos
            ENQUANTO NOT OverRun
                GRAVA(Filler) ; coloca seqüência de preenchimento("1s")
                ; ou dados interpolados no buffer
            IF NOT OverRun ENTAO
                GRAVA(Pacote) ; coloca conteúdo do pacote no
                ; buffer
            SENAO
                DESCARTA(Pacote) ; despreza o pacote
                Expected = (Received + 1) mod 2^16
            FIM ENQUANTO
        SENAO ; não incrementa Expected
        SE NOT UnderRun ENTAO
            L = ( (Played - Received) mod 2^16 ) - 2^15
            SE 0 < L <= R ENTAO ; substitui os dados do pacote
                GRAVA(Pacote) ; previamente marcado como
                ; perdido
        SENAO
            DESCARTA(Pacote) ; despreza o pacote
    FIM ENQUANTO

```

Caso seja assumido pela aplicação que não é necessária a reordenação, ou seja, os pacotes que chegam atrasados não são recuperados ( $R = 0$ ), o final do algoritmo (indicado em itálico), pode ser substituído, simplesmente, por:

```

        SENAO ; não incrementa Expected
        DESCARTA(Pacote) ; despreza o pacote
    FIM ENQUANTO

```



## APÊNDICE E – LISTAGEM DE FUNÇÕES MATLAB®

### E.1 – FUNÇÃO PARA SIMULAÇÃO DE PACOTES COM O ALGORITMO IETF

```
function SimulaPacotesIETFDC(Quantidade,OcupacaoMinima,Recuperaveis,ProbDesordem)
%%% Arquivo: SimulaPacotesIETFDC.m
%%% Autor: Eng. Wilson Dutra Sampaio - LabCom - ENE/UnB
%%% Objetivo: Simulação do atraso sofrido pelos pacotes em uma PSN e seqüenciamento
%%% desses pacotes na recepcao
%%% Algoritmo: Buffer linear (por simplicidade de gerenciamento) escrita e leitura sequencial
%%% (Apresentado IETF-PWE3 TDMoIP Internet-Draft)
% Parâmetros:
% Round-trip time da rede (em ms):
RTT=80;
% Tamanho do jitter buffer (em ms)- assumido linear, com capacidade para todos os pacotes, por
simplicidade de gerenciamento:
BufferSize=Quantidade;
% Ocupacao para inicio da reproducao:
Ocupacao=OcupacaoMinima;
% Duração de cada pacote (em ms):
Tpacote=1;
% Numero de pacotes da simulacao:
Pacotes=Quantidade;
% Intervalo de pacotes atrasados passíveis de recuperacao:
R=Recuperaveis;
% Base de tempo:
t = [0:Tpacote:Pacotes*Tpacote];
N = length(t);
% Gera pacotes:
Tx = zeros(3,N);
NS=round(rand(1)*2^16);
Tx(1,1)=NS;
for k=2:N;
    Tx(1,k)=NS+k-1;
end
Tx(2,:)=t;
% Transmite pacotes, atribuindo um instante de chegada aleatorio entre [RTT/2; RTT/2+RTT]:
for k=1:N;
    Tx(3,k)=t(:,k)+RTT/2;
    if rand(1) < ProbDesordem
        Tx(3,k)=Tx(3,k)+rand(1)*RTT;
    end
end
% Plota instantes de transmissao e recepcao
figure(1);
plot (Tx(2,:),Tx(1,:),'b.',Tx(3,:),Tx(1,:),'g.');
```

title('Instantes de Transmissao e Recepcao dos pacotes');

xlabel('t (ms)'); ylabel('Numero Sequencia');

disp('Pressione uma tecla para continuar.');

pause;

% Torna o sistema causal, gerando Rx com os pacotes recebidos por ordem de chegada:

Rx = zeros(2,N);

Cx=Tx;

for k=1:N;

    Rx(2,k)=100\*RTT; % Forca um instante de recepcao muito grande

```

for i=1:N;
    if Cx(3,i)<=Rx(2,k); % Se o instante de recepcao de Cx (imagem de Tx) e' menor que Rx
corrente, troca.
        Rx(1,k)=Cx(1,i); % Assume o NS recebido de Cx
        Rx(2,k)=Cx(3,i); % Assume o instante de recepcao de Cx
        ok=i;
    end
end
Cx(3,ok)=200*RTT; % Forca um instante de recepcao maior ainda para o Cx ja' carregado em
Rx
end
% Plota transmissao x recepcao
figure(2);
plot (Tx(2,:),Tx(1,:),'b',Rx(2,:),Rx(1,:),'g');
title('Recepcao de pacotes por ordem de chegada');
xlabel('t (ms)'); ylabel('Numero Sequencia');
disp('Pressione uma tecla para continuar. ');
pause;
% Acomoda pacotes no buffer e gera fluxo de saida:
Buffer=zeros(1,BufferSize);
Payout=zeros(1,Pacotes);
% Inicializacao da Thread de Escrita:
% Grava o primeiro pacote recebido:
Buffer(1,1)=Rx(1,1);
% Inicializa Ponteiro Payout com primeiro pacote recebido
PonteiroTDM=Rx(1,1);
% Inicializacao da Thread de Leitura:
Perdas=0;
Expected=PonteiroTDM+1;
% Simula Aplicacao ativa no Receptor:
for k=2:Pacotes+RTT*Ocupacao;
    % Simula Thread de Escrita:
    % Escreve pacotes de forma reordenada
    if k<=Pacotes;
        Received=Rx(1,k);
        if Received==Expected;
            % Pacote em ordem
            % Grava pacote no buffer de reproducao
            Buffer(1,k)=Rx(1,k);
            Expected=Received+1;
        else
            % Pacote fora de ordem
            D=Received-Expected;
            if D>0;
                % Recebido e' mais novo (maior) que o esperado
                % Interpreta que anteriores foram perdidos
                % Mitiga Perdas de pacote
                i=k;
                while i<=BufferSize;
                    % Coloca ultimo pacote recebido com sucesso n as posicoes remanescentes do buffer
                    Buffer(1,i)=Buffer(1,k-1);
                    i=i+1;
                end
                Expected=Received+1;
                % Incrementa contador de perdas em 'D' unidades, do recebido ao esperado
                Perdas=Perdas+D;
            else
                % Recebido e' mais antigo (menor) que o esperado
                L=Received-PonteiroTDM;
                % Verifica se esta no intervalo admitido para recuperacao de pacotes atrasados
                if L>0 & L<=R;
                    % Coloca o pacote atrasado recebido na posicao correta no buffer
                    Buffer(1,(Received-Rx(1,1)+1))=Rx(1,k);

```

```

        % Decrementa contador de perdas para o pacote recuperado
        Perdas=Perdas-1;
    end
end
end
end
% Simula Thread de Leitura:
% Le pacotes de forma sequencial
if k>RTT*Ocupacao;
    % Reproduz pacote corrente
    Playout(1,k-RTT*Ocupacao)=Buffer(1,PonteiroTDM-Buffer(1,1)+1);
    PonteiroTDM=PonteiroTDM+1;
end
end
% Base de tempo para Reproducao:
tp=[RTT/2+(RTT*Ocupacao+1)*Tpacote:Tpacote:RTT/2+(Pacotes+RTT*Ocupacao)*Tpacote];
% Informa a quantidade de pacotes perdidos nao recuperados como saida da funcao
SimulaPacotesIETFDC=Perdas
% Plota processo de Reproducao a partir do buffer
figure(3);
plot (Tx(2,:),Tx(1,:),'b',Rx(2,:),Rx(1,:),'g',tp,Playout(1,:),'r');
title('Sequenciamento e Reproducao dos pacotes recebidos ');
xlabel('t (ms)'); ylabel('Numero Sequencia');
drawnow;

```

## E.2 – FUNÇÃO PARA SIMULAÇÃO DE PACOTES COM O NOVO ALGORITMO

```

function SimulaPacotesNADC(Quantidade,OcupacaoMinima,ProbDesordem)
%%% Arquivo: SimulaPacotesNADC.m
%%% Autor: Eng. Wilson Dutra Sampaio - LabCom - ENE/UnB
%%% Objetivo: Simulação do atraso sofrido pelos pacotes em uma PSN e sequenciamento
%%% desses pacotes na recepcao
%%% Algoritmo: Buffer circular auto-gerenciavel com escrita ordenada e leitura sequencial
%%% (Proposta inédita para Dissertação de Mestrado)
% Parâmetros:
% Round-trip time da rede (em ms):
RTT=80;
% Tamanho do jitter buffer (em ms):
BufferSize=RTT;
% Ocupacao para inicio da reproducao:
Ocupacao=OcupacaoMinima;
% Duração de cada pacote (em ms):
Tpacote=1;
% Numero de pacotes da simulacao:
Pacotes=Quantidade;
% Base de tempo:
t = [0:Tpacote:Pacotes*Tpacote];
N = length(t);
% Gera pacotes:
Tx = zeros(3,N);
NS=round(rand(1)*2^16);
Tx(1,1)=NS;

for k=2:N;
    Tx(1,k)=NS+k-1;
end
Tx(2,:)=t;
% Transmite pacotes, atribuindo um instante de chegada aleatorio entre [RTT/2; RTT/2+RTT]:
for k=1:N;
    Tx(3,k)=t(:,k)+RTT/2;

```

```

if rand(1) < ProbDesordem
    Tx(3,k)=Tx(3,k)+rand(1)*RTT;
end
end
% Plota pacotes transmitidos e recebidos
figure(1);
plot (Tx(2,:),Tx(1,:),'b.',Tx(3,:),Tx(1,:),'g.');
```

title('Instantes de Transmissao e Recepcao dos pacotes');

xlabel('t (ms)'); ylabel('Numero Sequencia');

disp('Pressione uma tecla para continuar.');

pause;

% Torna o sistema causal, gerando Rx com os pacotes recebidos por ordem de chegada:

Rx = zeros(2,N);

Cx=Tx;

for k=1:N;

    Rx(2,k)=100\*RTT; % Forca um instante de recepcao muito grande

    for i=1:N;

        if Cx(3,i)<=Rx(2,k); % Se o instante de recepcao de Cx (imagem de Tx) e' menor que Rx corrente, troca.

            Rx(1,k)=Cx(1,i); % Assume o NS recebido de Cx

            Rx(2,k)=Cx(3,i); % Assume o instante de recepcao de Cx

            ok=i;

        end

    end

    Cx(3,ok)=200\*RTT; % Forca um instante de recepcao maior ainda para o Cx ja' carregado em Rx

end

% Plota instantes de transmissao e recepcao

figure(2);

plot (Tx(2,:),Tx(1,:),'b',Rx(2,:),Rx(1,:),'g');

title('Recepcao de pacotes por ordem de chegada');

xlabel('t (ms)'); ylabel('Numero Sequencia');

disp('Pressione uma tecla para continuar.');

pause;

% Acomoda pacotes no buffer e gera fluxo de saida:

Buffer=zeros(2,BufferSize);

Playout=zeros(1,Pacotes);

% Inicializacao da Thread de Escrita:

% Grava o primeiro NS e pacote recebido:

NS=Rx(1,1);

PosicaoE=mod(NS,BufferSize)+1; % Porque nao existe indice 0 na matriz

Buffer(1,PosicaoE)=Rx(1,1);

Buffer(2,PosicaoE)=Rx(2,1);

% Inicializa Ponteiro Playout com primeiro NS recebido

PonteiroTDM=NS;

% Inicializacao da Thread de Leitura:

PosicaoL=mod(PonteiroTDM,BufferSize)+1; % Porque nao existe indice 0 na matriz

Perdas=0;

% Simula Aplicacao ativa no Receptor:

for k=2:Pacotes+BufferSize\*Ocupacao;

    % Simula Thread de Escrita:

    % Escreve pacotes de forma reordenada

    if k<=Pacotes;

        NS=Rx(1,k);

        PosicaoE=mod(NS,BufferSize)+1; % Porque nao existe indice 0 na matriz

        if NS>PonteiroTDM;

            % Grava NS e instante de envio no buffer de reproducao

            Buffer(1,PosicaoE)=Rx(1,k);

            Buffer(2,PosicaoE)=Rx(2,k);

        end

    end

    % Simula Thread de Leitura:

    % Le pacotes de forma sequencial

```

if k>BufferSize*Ocupacao;
    % Reproduz pacote corrente
    Playout(1,k-BufferSize*Ocupacao)=Buffer(1,PosicaoL);
    PonteiroTDM=PonteiroTDM+1;
    PosicaoL=mod(PonteiroTDM,BufferSize)+1; % Porque nao existe indice 0 na matriz
    NSLido=Buffer(1,PosicaoL);
    if NSLido ~= PonteiroTDM;
        % Mitiga Perdas
        % Grava PonteiroTDM na posicao corrente, pois NS nao e' o esperado, indicando que esta
atrasado ou perdido
        Buffer(1,PosicaoL)=PonteiroTDM;
        % Incrementa contador de perdas
        if k< Pacotes+BufferSize*Ocupacao;
            Perdas=Perdas+1;
        end
    end
end
end
end
end
% Base de tempo para Reproducao:
tp=[RTT/2+(BufferSize*Ocupacao+1)*Tpacote:Tpacote:RTT/2+(Pacotes+BufferSize*Ocupacao)*Tpacote];
% Informa a quantidade de pacotes perdidos nao recuperados como saida da funcao
SimulaPacotesNADC=Perdas
% Plota processo de Reproducao a partir do buffer
figure(3);
plot (Tx(2,:),Tx(1,:),'b',Rx(2,:),Rx(1,:),'g',tp,Playout(1,:),'r');
title('Sequenciamento e Reproducao dos pacotes recebidos ');
xlabel('t (ms)'); ylabel('Numero Sequencia');
drawnow;

```