

**ALEXANDRE FÜRSTENBERGER**

**DESENVOLVIMENTO DE UM SISTEMA DE  
CONCENTRAÇÃO DE DADOS DE MEDIÇÃO  
FASORIAL SINCRONIZADA**

**FLORIANÓPOLIS – SC  
2010**

Catálogo na fonte pela Biblioteca Universitária  
da  
Universidade Federal de Santa Catarina

F991d Furstenberger, Alexandre  
Desenvolvimento de um sistema de concentração de dados de  
medição fasorial sincronizada [dissertação] / Alexandre  
Furstenberger ; orientador, Ildemar Cassana Decker.  
- Florianópolis, SC, 2010.  
188 p.: il., tabs.

Dissertação (mestrado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico. Programa de Pós-Graduação em  
Engenharia Elétrica.

Inclui referências

1. Engenharia elétrica. 2. Medição fasorial sincronizada.  
3. Concentrador de dados fasoriais. I. Decker, Ildemar  
Cassana. II. Universidade Federal de Santa Catarina. Programa  
de Pós-Graduação em Engenharia Elétrica. III. Título.

CDU 621.3

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA ELÉTRICA**

**DESENVOLVIMENTO DE UM SISTEMA DE  
CONCENTRAÇÃO DE DADOS DE MEDIÇÃO  
FASORIAL SINCRONIZADA**

Dissertação submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a  
obtenção de grau de Mestre em Engenharia Elétrica

**ALEXANDRE FÜRSTENBERGER**

Florianópolis, outubro de 2010.



# DESENVOLVIMENTO DE UM SISTEMA DE CONCENTRAÇÃO DE DADOS DE MEDIÇÃO FASORIAL SINCRONIZADA

Alexandre Fürstenberger

‘Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia Elétrica, Área de Concentração em Sistemas de Energia Elétrica, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.’



Ildemar Cassana Decker, D.Sc.  
Orientador

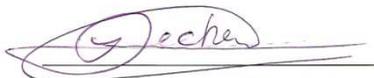


Marcelo Neujahr Agostini, Dr.  
Co-orientador



Roberto de Souza Salgado, Ph.D.  
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

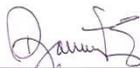
Banca Examinadora:



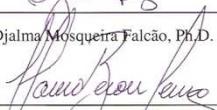
Ildemar Cassana Decker, D.Sc.  
Presidente



Marcelo Neujahr Agostini, Dr.



Djalma Mosqueira Falcão, Ph.D.



Flávio Beçon Lemos, Dr.



Aginaldo Silveira e Silva, Ph.D.



*Aos meus pais Renato e Valdelis, cuja fé em meu trabalho, dedicação na minha educação e orientação espiritual guiaram-me nos caminhos da vida, fazendo com que meus sonhos se tornassem realidade.*



## AGRADECIMENTOS

A Marcelo Agostini primeiramente, pois foi de sua responsabilidade a minha inserção dentro do grupo de pesquisa do Projeto MedFasee. Sem esta oportunidade, esta Dissertação de Mestrado não teria se tornado realidade. Agradeço também as horas dedicadas como co-orientador e, sobretudo, como amigo.

A Ildemar Cassana Decker, meu orientador pelo período em que estive no LabPlan até o fim deste trabalho. Sua didática e inspiração serviram de alicerces para a elaboração de um documento científico compatível com o nível de pesquisa desenvolvido no Projeto MedFasee.

Aos meus colegas de Projeto MedFasee, Marcelo Benetti, Ricardo Lira, Thiago Jeremias, Geyson Brustolin, Pedro César Cordeiro Vieira, Daniel Dotta, Rodolfo Bialecki Leandro e Valmor Zimmer, que direta ou indiretamente contribuíram para este trabalho.

Aos meus colegas de mestrado Diogo Bedin, Annelise Anderson, Emanoelli Cipriani e Daniel Collier, que me acompanharam na difícil jornada de provas e disciplinas.

Aos meus amigos de Labplan, Gelson Brigatto, Everthon Taghori Sica, Fabricio Takigawa, Ritchie Guder, Vitor Luiz de Matos, cuja convivência foi um privilégio.

A Banca Examinadora, Aguinaldo Silveira e Silva, Flávio Becon Lemos e Djalma Mosqueira Falcão, pelas suas inestimáveis contribuições.

Aos meus colegas de trabalho Maurício César Costa e Marcelo Luís Loureiro dos Santos. Sua paciência e compreensão, bem como amizade e apoio me ajudaram significativamente a conduzir o trabalho de mestrado paralelamente à rotina empresarial.

Agradeço também a Vanessa Virgínio de Araujo, cujo carinho, afeto, companhia, paciência, amor e compreensão me ajudaram nas jornadas de trabalho estendidas.

A todos os professores da Pós-Graduação e demais colegas de LabPlan. A convivência em um ambiente plural de conhecimentos e experiências serviu de grande aprendizado para o restante de minha vida profissional.

Por fim, agradeço aos órgãos institucionais de fomento: Universidade Federal de Santa Catarina (UFSC), Fundação de Ensino e Engenharia de Santa Catarina (FEESC) e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).



Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Elétrica

## **DESENVOLVIMENTO DE UM SISTEMA DE CONCENTRAÇÃO DE DADOS DE MEDIÇÃO FASORIAL SINCRONIZADA**

**Alexandre Fürstenberger**

Outubro / 2010

Orientador: Ildemar Cassana Decker, D.Sc.

Co-Orientador: Marcelo Neujahr Agostini, D.Eng.

Área de Concentração: Sistemas de Energia Elétrica.

Palavras-chave: Medição Fasorial Sincronizada; Concentrador de Dados Fasoriais.

Número de Páginas: 188.

O presente trabalho aborda o desenvolvimento de um sistema de concentração de dados de sistemas de medição fasorial sincronizada, de arquitetura descentralizada, contemplando dois níveis hierárquicos entre Concentradores de Dados Fasoriais (*Phasor Data Concentrator* – PDC), com objetivo de aumentar a escalabilidade do sistema em termos de processamento e comunicação. Em primeiro lugar são determinados os aspectos conceituais do desenvolvimento do Sistema de Concentração de Dados Fasoriais (*Phasor Data Concentrator System* – PDCS) proposto neste trabalho. São apresentadas as considerações sobre arquiteturas de aquisição de dados e arranjos entre equipamentos, bem como são descritas as lógicas de operação do sistema e forma de comunicação entre os equipamentos com funcionalidade de PDC. Em um segundo momento, apresenta-se o projeto de *software* implementado, cujo desenvolvimento baseia-se no paradigma da Modelagem Orientada a Objetos (MOO). São apresentadas classes desenvolvidas para representar equipamentos e processos relativos à operação do PDCS, visando permitir a comunicação entre PDCs.

Foram implementados protótipos de PDCS utilizando-se como plataforma de testes a estrutura do protótipo de Sistema de Medição Fasorial Sincronizada do Projeto MedFasee. Os resultados obtidos mostram o correto funcionamento do sistema, apresentando seu comportamento frente às perdas de dados nos canais de comunicação, mudanças de topologia da rede de medição e disponibilização de dados via interface de tempo real e base de dados histórica.



Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering

## **DEVELOPMENT OF A CONCENTRATION SYSTEM FOR SYNCHRONIZED PHASOR MEASUREMENT DATA**

**Alexandre Fürstenberger**

October / 2010

Advisor: Ildemar Cassana Decker, D.Sc.

Co-Advisor: Marcelo Neujahr Agostini, D.Sc.

Concentration Area: Electric Power Systems.

Keywords: Synchronized Phasor Measurement; Phasor Data Concentrator.

Number of Pages: 188.

The present work tackles the development of a synchronized phasor measurement data concentration system, with decentralized architecture, contemplating two hierarchical levels between Phasor Data Concentrators (PDC), aiming the system scalability increase in terms of processing and communication. At first, conceptual concepts concerning the development of the Phasor Data Concentration System (PDCS) proposed in this work is presented. Considerations about data acquisition architectures and equipment arrangement are addressed, along with system operational logics and messaging formats for data exchange between PDCs. In a second moment, the software project is presented, whose development is based on Oriented Object Modeling (OOM) paradigm. The project classes are presented, which design aim for equipment and process abstraction related to PDCS operation, allowing the communication between PDCs.

PDCS prototypes were implemented, using as testing platform the Synchronized Phasor Measurement System prototype of MedFasee Project. The results show the correct system operation, presenting its behavior facing data loss due communication channels, measurement topology change, real time data and historical data availability.



# SUMÁRIO

<b>Lista de Figuras .....</b>	<b>19</b>
<b>Lista de Tabelas .....</b>	<b>23</b>
<b>Lista de Siglas .....</b>	<b>25</b>
<b>1. INTRODUÇÃO .....</b>	<b>27</b>
1.1 MOTIVAÇÕES .....	30
1.2 DEFINIÇÃO E DELIMITAÇÃO DO PROBLEMA .....	32
1.3 OBJETIVO GERAL .....	33
1.3.1 Objetivos Específicos.....	33
1.4 ORGANIZAÇÃO DO TEXTO .....	34
<b>2. SISTEMAS DE MEDIÇÃO FASORIAL SINCRONIZADA.....</b>	<b>37</b>
2.1 ESTRUTURA FUNDAMENTAL DE UM SPMS.....	38
2.1.1 Sistema de Sincronização.....	39
2.1.2 Unidade de Medição Fasorial.....	40
2.1.3 Concentrador de Dados Fasoriais.....	41
2.1.4 Canais de Comunicação .....	42
2.2 DESENVOLVIMENTOS E APLICAÇÕES EM SPMS .....	42
2.2.1 Estados Unidos.....	42
2.2.2 China.....	45
2.2.3 Brasil.....	47
2.2.3.1 Projeto MedFasee.....	47
2.2.3.2 Projeto SIN.....	50
2.2.4 Outros Países.....	53
2.2.5 Comentários Finais.....	55
<b>3. SISTEMA DE CONCENTRAÇÃO DE DADOS FASORIAIS... 57</b>	
3.1 DESENVOLVIMENTOS E SOLUÇÕES EM CONCENTRADORES DE DADOS FASORIAIS .....	58
3.1.1 Soluções de Projetos .....	58
3.1.2 Soluções Comerciais .....	63
3.2 PRINCIPAIS REQUISITOS PARA O DESENVOLVIMENTO DE UM SISTEMA DE CONCENTRAÇÃO DE DADOS FASORIAIS .....	65
3.3 ASPECTOS DE ARQUITETURA DE CONCENTRAÇÃO DE DADOS FASORIAIS .....	67
3.3.1 Arquitetura Centralizada .....	67
3.3.2 Arquitetura Descentralizada.....	68

3.4	PROPOSTA DE UM SISTEMA DE CONCENTRAÇÃO DE DADOS	
	FASORIAIS - PDCS.....	70
3.4.1	Comunicação entre PDCs – Norma IEEE C37.118.....	71
3.4.1.1	Frame de Cabeçalho.....	72
3.4.1.2	Frame de Configuração.....	72
3.4.1.3	Frame de Dados.....	73
3.4.1.4	Frame de Comando.....	75
3.4.2	Lógica de Operação do PDCS.....	76
3.4.2.1	Configuração da Operação do PDCS.....	76
3.4.2.2	Operação do PDCS.....	77
3.4.3	Lógica de Operação do PDC.....	78
3.4.3.1	Visão geral da lógica operacional do PDC.....	78
3.4.3.2	Etapa de Inicialização.....	79
3.4.3.3	Etapa Principal de Operação do PDC.....	83
3.4.3.4	Processo de Acesso e Escrita no Banco de Dados.....	89
3.5	COMENTÁRIOS FINAIS.....	91

## **4. PROJETO DE SOFTWARE E IMPLEMENTAÇÃO COMPUTACIONAL..... 93**

4.1	PARADIGMA DA MODELAGEM ORIENTADA A OBJETOS.....	93
4.2	IMPLEMENTAÇÃO DO CÓDIGO-FONTE DO PDC – LINGUAGEM E COMPILADOR.....	95
4.3	CLASSES DO PROJETO DE SOFTWARE DO PDC.....	95
4.3.1	Projeto de software do PDC.....	96
4.3.2	Classe C_PDC.....	97
4.3.2.1	Configuração dos atributos da Classe C_PDC.....	100
4.3.2.2	Ciclo de aquisição de dados e <i>buffer</i> em HD.....	100
4.3.2.3	Disponibilização de dados em tempo real e <i>buffer</i> em memória.....	101
4.3.2.4	Monitoração do estado operativo dos equipamentos do SPMS.....	107
4.3.3	Classe C_PMU_PDC.....	108
4.3.3.1	Processamento dos <i>frames</i> IEEE C37.118.....	110
4.3.4	Classe C_PDC_PDC.....	112
4.3.4.1	Representação do PDC remoto.....	114
4.3.4.2	Estrutura do <i>frame</i> de configuração da classe C_PDC_PDC.....	114
4.3.5	Classe C_Conexao.....	115
4.3.5.1	Protocolos de transporte.....	116
4.3.5.2	Gerenciamento dos dados de rede.....	117
4.3.6	Classe C_Frame.....	118
4.3.7	Classe C_App.....	120
4.4	COMENTÁRIOS FINAIS.....	122

## **5. EXPERIMENTOS E RESULTADOS..... 123**

5.1	PROTÓTIPOS DO PDCS.....	123
5.1.1	Protótipo 1 - PDCS utilizando 2 PDCs Front End e PDC Mestre.....	124
5.1.2	Protótipo 2 – PDCS utilizando 3 PDCs Front End e PDC Mestre.....	125
5.1.3	Capacidade dos PDCS.....	127
5.1.3.1	Largura de Banda .....	127
5.1.3.2	Armazenamento em base de dados histórica .....	129
5.2	ANÁLISE DAS PERDAS DE DADOS .....	130
5.2.1	Perdas Nulas.....	130
5.2.2	Perdas Devido aos Canais de Comunicação – Internet .....	131
5.2.3	Perdas de Dados Entre PDCs Front End e PDC Mestre.....	133
5.3	VALIDAÇÃO DA ESTRUTURA DE DADOS EM TEMPO REAL .....	135
5.4	VALIDAÇÃO DA ESTRUTURA DE DADOS HISTÓRICA E ANÁLISE DE OCORRÊNCIA .....	138
5.4.1	Validação da base de dados entre PDCs Front End e PDC Mestre .....	138
5.4.2	Validação da base de dados em relação ao SIN .....	142
5.5	COMENTÁRIOS FINAIS.....	145
<b>6.</b>	<b>CONCLUSÕES .....</b>	<b>147</b>
6.1	SUGESTÕES PARA TRABALHOS FUTUROS .....	150
<b>APÊNDICE A. FERRAMENTAS DE DESENVOLVIMENTO</b>		<b>153</b>
.....		
A.1.	XML .....	153
A.1.1	DEFINIÇÕES .....	154
A.1.2	REGRAS DE FORMAÇÃO DO DOCUMENTO .....	155
A.1.3	ESTRUTURA EM ÁRVORE .....	156
A.1.4	VALIDAÇÃO.....	157
A.1.5	DOCUMENT TYPE DEFINITION.....	157
A.1.6	W3C SCHEMA .....	158
A.1.7	APLICAÇÃO DE XML NO PDC.....	160
A.2.	DOXYGEN .....	162
A.3.	LINGUAGEM DE PROGRAMAÇÃO C++ .....	163
<b>APÊNDICE B.</b>	<b>ARQUIVOS XML UTILIZADOS NO PDC</b>	<b>165</b>
B.1.	ARQUIVO DE CONFIGURAÇÃO DO PDC CONFIGPDC.XML .....	165
B.2.	ARQUIVO SCHEMA CONFIGPDC.XSD .....	168
<b>APÊNDICE C.</b>	<b>COMUNICAÇÃO MULTICAST .....</b>	<b>173</b>

C.1. UNICAST .....	174
C.2. BROADCAST.....	175
C.3. MULTICAST.....	176
C.4. ESCOPO DE ENDEREÇOS MULTICAST.....	177
C.5. PROTOCOLO IGMP .....	178
C.6. API PARA FILTRAGEM DE REMETENTES.....	179
C.7. APLICAÇÃO DE MULTICAST EM SPMS.....	180
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>183</b>

## LISTA DE FIGURAS

Figura 1.1 – Áreas tecnológicas principais para Smart Grids. Adaptado de (HOROWITZ, S. H., et al., 2010). .....	28
Figura 2.1 – Componentes básicos que formam um SPMS. Adaptado de (SANTOS, M. M., 2008). .....	38
Figura 2.2 – Arquitetura interna da PMU. ....	41
Figura 2.3 – Concentrador de Dados Fasoriais e suas funções. ....	41
Figura 2.4 – PMUs do sistema Norte Americano. Adaptado de (HU, Y., et al., 2010). .....	44
Figura 2.5 – Concentradores de Dados do SPMS Chinês. Adaptado de (BI, T., 2010). .....	46
Figura 2.6 – Protótipo de SMPS do implementado pelo Projeto MedFasee. ....	48
Figura 2.7 – Protótipo de SPMS instalado no sistema de transmissão da Eletrosul. ....	49
Figura 2.8 – Monitoração em tempo real dos sincrofasores do Projeto MedFasee Frequência. ....	50
Figura 2.9 – Arquitetura do Sistema de Medição Fasorial do SIN. ....	53
Figura 3.1 – Estrutura do PDC Stattnet - Adaptado de (LEIRBUKT, A., et al., 2008). .....	60
Figura 3.2 – Arquitetura do protótipo de SPMS do Projeto MedFasee - adaptado de (DECKER, I. C., et al., 2006). ....	62
Figura 3.3 – Arquitetura Centralizada de Concentração de Dados Fasoriais. ....	68
Figura 3.4 – Arquitetura Descentralizada de Concentração de Dados Fasoriais. ....	69
Figura 3.5 – Cenário de aplicação genérica do PDCS proposto. ....	70
Figura 3.6 – Estrutura geral dos frames da norma IEEE C37.118. Adaptado de (IEEE, 2005). ....	72
Figura 3.7 – Lógica de configuração do PDCS. ....	77
Figura 3.8 – Fluxo de sincrofasores na operação do PDCS. ....	77
Figura 3.9 – Lógica de operação do PDC. ....	79
Figura 3.10 – Fluxograma da Etapa de Inicialização do PDC. ....	82
Figura 3.11 – Etapa Principal de Operação do PDC. ....	83
Figura 3.12 – Etapa de Configuração do PDC. ....	85
Figura 3.13 – Processo de Recebimento do Frame de Dados IEEE C37.118 do PDC. ....	88
Figura 3.14 – Processos de Tempo Real e Armazenamento que compõe o PDC. ....	90

Figura 4.1 – PDC – Diagrama UML de do projeto de software do PDC. .....	97
Figura 4.2 – Diagrama UML da Classe C_PDC - Atributos e Métodos. .....	98
Figura 4.3 - Estrutura Interna do Buffer Circular em Memória do PDC. .....	103
Figura 4.4 - Diagrama de Classes do Buffer Circular em Memória....	104
Figura 4.5 – Interpretação gráfica dos tempos da re-sincronização de dados em memória. ....	105
Figura 4.6 – Diagrama UML da Classe C_PMU_PDC. ....	109
Figura 4.7 – Diagrama UML da Classe C_PDC_PDC. ....	113
Figura 4.8 – Diagrama UML da Classe C_Conexao.....	116
Figura 4.9 – Diagrama UML da Classe C_Frame.....	119
Figura 4.10 – Diagrama UML da Classe C_App.....	121
Figura 5.1 – Protótipo de PDCS com 2 PDCs Front End e PDC Mestre. .....	124
Figura 5.2 – Bancada de Testes do PDCS.....	125
Figura 5.3 – Protótipo de PDCS com 3 PDCs Front End e PDC Mestre. .....	126
Figura 5.4 – Equipamento executando o processo do openPDC. ....	127
Figura 5.5 – Interface gráfica do TVA PMU Connection Tester em funcionamento, recebendo dados do PDC Mestre. ....	137
Figura 5.6 – Interface gráfica do TVA PMU Connection Tester em funcionamento, recebendo dados do PDC Front End 1. ....	137
Figura 5.7 - Interface gráfica do TVA PMU Connection Tester em funcionamento, recebendo dados do PDC Front End 2. ....	138
Figura 5.8 – Frequência do SIN no momento da ocorrência . ....	143
Figura 5.9 – Tensões do SIN no momento da ocorrência. ....	144
Figura 5.10 – Diferença Angular entre PMUs USP-SC e UFSC. ....	145
Figura A.1 – Arquivo MedFasee.xml – Exemplo para estrutura do Projeto MedFasee.....	156
Figura A.2 – Estrutura em Árvore representando o arquivo MedFasee.xml.....	157
Figura A.3 – Arquivo MedFasee.DTD. Exemplo de DTD para o arquivo MedFasee.xml.....	158
Figura A.4 – Representação Gráfica do arquivo MedFasee.xsd. ....	159
Figura A.5 – Arquivo de Configuração pdc.cfg.....	160
Figura A.6 – Índice de Classes gerado pela aplicação do Doxygen....	163
Figura A.7 – Diagrama de colaboração entre Classes gerado pelo Doxygen.....	163
Figura C.1 – Rede comunicando-se em Unicast – Adaptado de (MA-	

KOFSKE, D., et al., 2003).....	174
Figura C.2 – Broadcast em uma LAN.....	175
Figura C.3 – Árvore de encaminhamento Multicast – Adaptado de (MAKOFS- KE, D., et al., 2003) .....	177



## LISTA DE TABELAS

Tabela 3.1 – Organização do Frame de Cabeçalho IEEE C37.118. ....	73
Tabela 3.2 – Frame de Configuração IEEE C37.118. ....	74
Tabela 3.3 – Frame de Dados IEEE C37.118. ....	75
Tabela 3.4 – Frame de Comando IEEE C37.118. ....	76
Tabela 5.1 – Demanda de largura de banda do frame de dados. ....	127
Tabela 5.2 – Evolução do requisito de largura de banda para a aquisição de dados pelo PDC. ....	128
Tabela 5.3 – Largura de banda requisitada pelo PDC Mestre. ....	129
Tabela 5.4 – Capacidade de disco rígido necessária para armazenamento dos dados em banco de dados histórico. ....	130
Tabela 5.5 – Resultados do MedPerdas durante as 24 horas do dia 01/06/2010 para a PMU UFSC. ....	132
Tabela 5.6 - Resultados do MedPerdas durante as 24 horas do dia 29/05/2010 para a PMU UnB. ....	133
Tabela 5.7 – Resultados do MedPerdas durante as 24 horas do dia 28/05/2010 para a PMU UNIFEI. ....	135
Tabela 5.8 – Dados da PMU UFPA presentes no PDC Front End 1 e Mestre. ....	139
Tabela 5.9 – Dados da PMU UFSC presentes no PDC Front End 2 e Mestre. ....	140
Tabela 5.10 – Dados da PMU UFPA armazenados no PDC Front End 1 e PDC Mestre. Período contemplando perda de dados. ....	141



## LISTA DE SIGLAS

AC – Alternate Current  
A/D – Processo de Conversão Analógico/Digital  
AEP – *American Electric Power*  
AMR – *Automatic Meter Reading*  
ANEEL – Agência Nacional de Energia Elétrica  
API – *Application Program Interface*  
ASCII – *American Standard Code for Information Interchange*  
BPA – *Bonneville Power Administration*  
CAISO – *California Independent System Operator*  
CEPRI – *Chinese Electric Power Research Institute*  
CERTS – *Consortium of Electric Reliability Technology Solutions*  
CIGRÈ – *International Council on Large Electric Systems*  
DFT – *Discrete Fourier Transform*  
DoE – *Department of Energy*  
EIPP – *Eastern Interconnect Phasor Project*  
EMS – *energy Management System*  
EPRI – *Electric Power Research Institute*  
FACTS – *Flexible AC Transmission System*  
FINEP – Financiadora de Estudos e Projetos  
GALILEO - *European Global Satellite Navigation System*  
GLONASS – *Global Navigation Satellite System*  
GPS – *Global Positioning System*  
HD – *Hard Drive* (Unidade de Disco Rígido de um Computador)  
HTML – *Hyper Text Markup Language*  
HVDC – *High Voltage Direct Current*  
IEEE – *Institute of Electrical and Electronic Engineers*  
IGMP – *Internet Group Management Protocol*  
IP – *Internet Protocol*  
IPDO – Informativo Preliminar Diário da Operação  
IRIG – *Inter Range Instrumentation Group*  
LAN – *Local Area Network*  
MOO – Modelagem Orientada a Objetos  
NETL – *National Energy Technology Laboratory*  
NERC – *North American Electric Reliability Corporation*  
NSF – *National Science Foundation*  
NTP – *Network Time Protocol*  
NPD – Núcleo de Processamento de Dados UFSC  
NYPA – *New York Power Authority*

OBDC - *Open Data Base Connectivity*  
PDC – *Phasor Data Concentrator*  
PDCS – *Phasor Data Concentrator System*  
PMU – *Phasor Measurement Unit*  
PPS – *Pulse Per Second*  
RTAI – *Real Time Application Interface*  
SCADA – *Supervisory Control and Data Acquisition System*  
SCDR – *Symmetric Component Distance Relay*  
SCE – *Southern California Edison*  
SEE – *Sistemas de Energia Elétrica*  
SGC – *State Grid Company*  
SGML - *Standard Generalized Markup Language*  
SIN – *Sistema Interligado Nacional*  
SPDC – *State Power Dispatching Center*  
SPDnet – *State Power Data Network*  
SPMS – *Synchronized Phasor Measurement System*  
SQL – *Structured Query Language*  
STL – *Standard Template Library*  
TC – *Transformador de Corrente*  
TI – *Tecnologia de Informação*  
TP – *Transformador de Potencial*  
TCP – *Transmission Control Protocol*  
TVA – *Tennessee Valley Authority*  
TVE – *Total Vector Error*  
UDP – *User Datagram Protocol*  
UML – *Unified Modeling Language*  
UTC – *Universal Time Coordinated*  
VPN – *Virtual Private Network*  
W3C – *World Wide Web Consortium*  
WAMS – *Wide Area Measurement System*  
WECC – *Western Electricity Coordinating Council*  
XML – *Extended Markup Language*

# CAPÍTULO 1

## 1. Introdução

Os mercados de energia elétrica passaram, nos últimos anos, por mudanças significativas, sobretudo no seu aspecto organizacional. Os processos de desregulamentação dos segmentos de geração, transmissão e distribuição permitiram o atendimento de requisitos de expansão da oferta e a introdução de competição no setor.

Do ponto de vista econômico, tais mudanças permitiram a existência de um ambiente competitivo, que busca minimizar o custo da energia elétrica, atrair capital privado para novos empreendimentos e otimizar os recursos presentes na matriz energética disponível (SILVA, E. L. D., 2001).

Como conseqüência, os Sistemas de Energia Elétrica (SEE) devem manter o compromisso de atendimento à demanda crescente, considerando-se requisitos de qualidade e continuidade, levando em consideração que sua expansão está sujeita a restrições sócio-ambientais. Ainda, como reflexo da viabilização econômica de novas tecnologias de geração de energia, as redes de transmissão devem estar preparadas para absorver a entrada de fontes de geração distribuída (conectadas aos sistemas de distribuição) e novas fontes de energia (eólica, solar, biomassa) de características intermitentes.

Todos estes aspectos remetem à necessidade da modernização dos sistemas de transmissão, sobretudo na incorporação de novas tecnologias que tiveram um salto evolutivo significativo nos últimos anos, tais como a Tecnologia de Informação (TI) e sistemas de comunicação, e que abrem caminho para novos paradigmas de estruturas de SEE como os *Smart Grids* ou Redes Inteligentes (FALCÃO, D. M., 2009).

O termo *Smart Grid* é mais bem compreendido se for observado como um conceito, não como equipamento único ou tecnologia específica. Sua principal premissa é a utilização de TI e sistemas de comunicação na rede elétrica, de forma que componentes da rede possam informar seu estado operativo, sendo traçadas estratégias de controle e operação eficientes. Uma rede elétrica sob este paradigma otimiza a produção e distribuição de energia, minimiza perdas, é auto-recuperável e possibilita a aplicação de novas tecnologias em eficiência energética e resposta

da demanda (EPRI, 2008;ETP, 2010).

Como consequência, novas características são adicionadas às redes, tais como:

- auto-recuperação (*self healing*);
- segurança cibernética;
- qualidade de energia;
- acomodação de diversas fontes de energia com diferentes características;
- redução de impacto ambiental;
- maior competição;
- resposta da demanda.

O *National Energy Technology Laboratory* (NETL), laboratório de pesquisa em energia vinculado ao Departamento de Energia dos Estados Unidos (*US Department of Energy - DoE*), identificou cinco áreas tecnológicas principais para a viabilidade da aplicação dos conceitos de *Smart Grids* nas redes elétricas (HOROWITZ, S. H., *et al.*, 2010;NETL, 2007). Essas áreas tecnológicas estão ilustradas pela Figura 1.1.



Figura 1.1 – Áreas tecnológicas principais para *Smart Grids*. Adaptado de (HOROWITZ, S. H., *et al.*, 2010).

Em primeiro lugar, têm-se os *Sistemas de Comunicação*. Canais de comunicação de banda larga, baixa latência e de elevada segurança cibernética permitem a integração entre subestações do sistema e centros de controle. Esta comunicação integrada permite ações de controle e correção rápidas e eficientes, bem como serve de alicerce para desenvolvimentos das demais áreas tecnológicas principais.

A *Medição e Monitoração* faz uso de equipamentos digitais que fornecem informações detalhadas do sistema. Do lado da demanda, os medidores eletrônicos, ou *Smart Meters* (*Automatic Meter Reading – AMR*), adicionam novas funcionalidades em relação aos medidores eletromecânicos. Dentre suas inovações, pode-se citar a medição auto-

mática do consumo de energia, possibilidade de analisar os fluxos de potência de determinada rede de consumo, disponibilização de sinais de preço, inclusão de resposta da demanda, etc.

Pelo lado da *Monitoração do Sistema*, considera-se o uso de tecnologias de monitoração em grandes áreas (*Wide Area Measurement Systems* - WAMS). Tal monitoração é possível por meio de dados fornecidos pelas Unidades de Medição Fasorial (*Phasor Measurement Unit* - PMU) (HOROWITZ, S. H., *et al.*, 2010). Estas unidades realizam o cálculo das tensões e correntes complexas do sistema em instalações distantes geograficamente, utilizando o sinal de sincronização de tempo do sistema GPS (*Global Positioning System*) como referência (PHADKE, A. G., 2002). A referência de tempo juntamente com a alta taxa de amostragem (em geral de trinta a sessenta medidas por segundo) são as principais características desta tecnologia, permitindo a monitoração da dinâmica dos SEE.

*Métodos de Controle e Proteção Avançados* incluem equipamentos e aplicações (algoritmos) que realizam a análise, diagnóstico e previsão de condições que possam afetar a operação do sistema elétrico. É atribuição destes métodos tomar ações corretivas a fim de mitigar, eliminar e prevenir cortes de suprimento.

A área de *Dispositivos Eletrônicos Avançados* abrange os avanços tecnológicos da eletrônica de potência, microeletrônica, materiais e supercondutividade. Dispositivos FACTS (*Flexible AC Transmission System*), sistemas de corrente contínua em alta tensão (HVDC – *High Voltage Direct Current*) e novas tecnologias de armazenamento de energia permitem responder à perturbações, por meio do uso de sinais de controle, em intervalos de tempo da ordem de milissegundos.

Tecnologias de *Suporte à Decisão* convertem dados de medição e monitoração do sistema elétrico em informações úteis aos operadores. Ferramentas de visualização sofisticadas, incluindo animações, gráficos em cores, simulações expeditas, fazem parte do arcabouço de informações disponibilizadas aos operadores, possibilitando a tomada de decisão de maneira rápida e segura.

Dentre as áreas tecnológicas apresentadas, a área de Medição e Monitoração exerce um papel importante no cenário de aplicação do conceito de *Smart Grids*. Os dados e informações obtidos do sistema dão subsídios para as outras áreas exercerem seus objetivos. De maneira particular, destaca-se o papel da monitoração em grandes áreas representada pelos Sistemas de Medição Fasorial Sincronizada (*Synchronized Phasor Measurement Systems* - SPMS).

Estes sistemas têm características diferenciadas em relação aos sistemas de supervisão e controle (*Supervisory Control and Data Acquisition* - SCADA) convencionais, elencadas a seguir:

- alta taxa de amostragem;
- sincronização temporal por meio de sistema GPS.

Tais características permitem que fenômenos dinâmicos sejam observados em maiores detalhes, possibilitando a extração de informações do sistema previamente desconhecidas.

Os SPMS têm por componente principal as PMUs. Entretanto, também fazem parte destes sistemas o Concentrador de Dados Fasoriais (*Phasor Data Concentrator* - PDC) e os canais de comunicação. Para a determinação das tensões e correntes complexas de um sistema, em um determinado instante de tempo, é necessário um sistema de sincronização suficientemente preciso como referência de tempo. O sistema GPS, por meio de seu sinal PPS (*Pulse Per Second*), possibilita sua utilização como fonte de sincronismo para os equipamentos do SPMS, dentro dos requisitos necessários (EHRENSPERGER, J. G., 2004; PHADKE, A. G., 2002).

O estudo e a aplicação de SPMS envolvem aspectos de diversas áreas de conhecimento, passando por conceitos de telecomunicações, tecnologia da informação, processamento digital de sinais, metodologias matemáticas aplicadas a SEE, projetos de desenvolvimento de *software*, etc.

Observa-se o papel da Medição Fasorial Sincronizada como uma área tecnológica fundamental para o avanço da modernização das redes elétricas, que caminha no sentido da viabilização da aplicação do conceito de *Smart Grids*. Desta forma, este trabalho concentra esforços no estudo e no desenvolvimento de SPMS, sendo suas motivações e seus objetivos discutidos a seguir.

## **1.1 Motivações**

Os primeiros desenvolvimentos envolvendo medição fasorial iniciaram em meados da década de 80, dentro do âmbito de pesquisa da *Virginia Polytechnic Institute and State University*. O primeiro desenvolvimento neste sentido se deu pela criação do SCDR (*Symmetrical Component Distance Relay*), relé de proteção para linhas de transmissão que utiliza componentes simétricas de tensão e corrente. O principal fator limitante na época era a falta de uma fonte de sincronização com precisão adequada, que pudesse servir de referência para o cálculo dos fasores (PHADKE, A. G., 2002).

Utilizando-se do sinal de sincronização do sistema GPS e dos al-

goritmos desenvolvidos para o SCDR, o primeiro protótipo de PMU foi desenvolvido no ano de 1988. Baseado neste protótipo, a *Macrodyne*, empresa fabricante de equipamentos para sistemas de medição e monitoração de SEE, desenvolveu em escala comercial o modelo de PMU *Macrodyne 1620*.

Considerando-se os aspectos evolutivos da tecnologia, em 1995, foi publicada a primeira norma relativa a sincrofasores pelo IEEE (*Institute of Electrical and Electronics Engineers*), a *IEEE Std 1344 – Standard for Synchrophasors for Power Systems* (IEEE, 1995). Este documento aborda, dentre seus principais tópicos, fontes para o sinal de sincronização das medidas fasoriais, formato de saída de dados enviados pelas PMUs e métodos para o cálculo de sincrofasores. Esta padronização permitiu a expansão da fabricação de equipamentos em nível comercial por diferentes empresas. Em 2005, com a evolução dos desenvolvimentos e o contexto de expansão no estudo da tecnologia, a norma IEEE 1344 foi completamente revisada, passando a ser denominada de *IEEE Std C37.118 – Standard for Synchrophasors for Power Systems* (IEEE, 2005). A nova norma revisita os formatos de mensagens com objetivo de prover maior compatibilidade entre diferentes aplicações de SPMS. Possibilita-se a inserção de dados de medidas analógicas nos frames de comunicação (medições de temperatura, por exemplo) e principalmente, estabelece parâmetros para testes de conformidade, quantificando o erro associado ao equipamento PMU através do conceito de Erro Total Vetorial (TVE – *Total Vector Error*). Estas ações traduzem a necessidade de estabelecer padrões que os equipamentos envolvidos em um SPMS devem obedecer, mantendo o compromisso de um ambiente de operação onde coexistem diversos fabricantes, independentes da solução tecnológica encontrada e aplicada nos seus equipamentos.

Devido à padronização bem definida para as PMUs, sua fabricação em escala comercial é realizada por diversas empresas que atuam no segmento de produção de equipamentos para medição e monitoração de SEE. Os desenvolvimentos atuais deste componente buscam aperfeiçoar a resposta dinâmica do equipamento, atentando também para as incertezas dos canais de instrumentação na exatidão da medição fasorial (HUANG, Z., *et al.*, 2007; LIRA, R., 2010).

Em contraparte, não existem especificações padronizadas para o sistema que realiza a concentração dos dados enviados pelas PMUs, o Concentrador de Dados Fasoriais (*Phasor Data Concentrator - PDC*). Este sistema tem por objetivo receber, processar, armazenar e disponibilizar os dados de medição fasorial, tanto para análise de dados históricos

(análises de estudos), quanto para processamento em tempo real (aplicações de monitoração).

A solução de concentração de dados depende do tamanho de cada SMPS, da taxa de aquisição de fasores, da largura de banda de rede utilizada, dentre outros fatores. Isto implica em soluções particulares para cada caso, dificultando a disponibilização de produtos comerciais no mercado de solução única, capazes de se adaptarem aos diversos casos presentes nos SEE.

Os tipos de solução, encontrados para concentração de dados fasoriais, na literatura, podem ser subdivididos em dois grupos: (a) soluções específicas de projetos em medição fasorial; (b) e pacotes de soluções comerciais, abrangendo diversos serviços de aquisição, armazenamento e utilização dos dados fasoriais.

A falta de padronização para a concentração de dados de SPMS, tanto no tocante aos equipamentos que são utilizados, quanto à arquitetura de maneira geral, deixa o tópico em aberto para discussões e novos desenvolvimentos. Neste contexto, a solução para aquisição e armazenamento de dados fasoriais é o tema central da presente dissertação.

## 1.2 Definição e Delimitação do Problema

A solução de aquisição e armazenamento de dados de SPMS deve contemplar diferentes aspectos, tais como (KARLSSON, D., *et al.*, 2004):

- infra-estrutura variável (adaptação a novos requisitos de uso do SPMS);
- portabilidade para aplicação de novas tecnologias;
- expansibilidade.

Para atender a estes requisitos, duas propostas de arquitetura se destacam:

- **Arquitetura Centralizada.** Esta arquitetura consiste de um PDC único, que recebe e processa os dados de todas as PMUs do SPMS.
- **Arquitetura Descentralizada.** Esta abordagem considera a presença de múltiplos PDCs que se comunicam entre si, podendo ser estabelecidos níveis hierárquicos para aquisição e envio de dados pelos equipamentos, dividindo-se então a função dos equipamentos entre Mestre e Escravo(s).

A primeira proposta se destaca como a solução primária da tecno-

logia, sendo aplicada em sistemas atualmente em operação, como por exemplo, em (DECKER, I. C., *et al.*, 2005) e (ZUO, J., *et al.*, 2008). A segunda aparece como uma evolução natural da arquitetura, tendo por principal objetivo agregar característica de expansibilidade aos SPMS, facilitando a adição de novas PMUs aos SEE.

É notório que, com a expansão da monitoração por meio de PMUs no sistema, um equipamento único seja sobrecarregado pela massa de dados enviada. Além disso, considerando-se que empresas diferentes tenham interesse em proporcionar o intercâmbio de dados fasoriais, especialmente entre interconexões que unem diferentes áreas de controle, faz-se necessária a viabilização da comunicação entre PDCs.

Para abranger esta solução, a Arquitetura Descentralizada é discutida (KARLSSON, D., *et al.*, 2004; MARTÍNEZ, E., 2008; MORAES, R. M., *et al.*, 2008) a fim de potencializar a escalabilidade (facilitando a adição de novas PMUs) e a confiabilidade (possibilidade de diferentes níveis de armazenamento de dados) do sistema.

### 1.3 Objetivo Geral

O presente trabalho tem por objetivo implementar um Sistema de Concentração de Dados Fasoriais (*Phasor Data Concentration System - PDCS*), de arquitetura descentralizada, contemplando dois níveis hierárquicos entre PDCs, visando assegurar escalabilidade ao sistema em termos de processamento e comunicação. Para isto utiliza-se, como plataforma de desenvolvimento, o protótipo de SPMS implementado no âmbito do Projeto MedFasee (MEDFASEE, 2003), atualmente composto por dez PMUs instaladas em redes de baixa tensão de universidades brasileiras.

#### 1.3.1 Objetivos Específicos

No desenvolvimento do trabalho, foram realizadas atividades que buscaram enriquecer o arcabouço de conhecimentos necessários para a base da escrita deste documento e prover o ferramental necessário para o desenvolvimento da proposta. Estas atividades compreendem os seguintes temas: (i) análise do estado da arte sobre tecnologias aplicadas na implementação de SPMS; (ii) estudo de técnicas de programação, tais como Modelagem Orientada a Objetos (MOO) e de ferramentas computacionais para o auxílio no desenvolvimento do *software* do PDC; (iii) adaptação do código-fonte do PDC do Projeto MedFasee, compatibilizando seu funcionamento no contexto da arquitetura do PDCS proposto.

Estes temas buscam atender aos seguintes objetivos específicos do trabalho:

- desenvolvimento conceitual do PDCS;
- projeto de *software* e implementação computacional;
- validação do sistema no âmbito do protótipo de SPMS do projeto MedFasee.

#### 1.4 Organização do Texto

O Capítulo 2 contém a revisão bibliográfica do trabalho, abrangendo-se, em um primeiro momento, o conceito de Sistemas de Medição Fasorial Sincronizada, sua estrutura básica, a descrição dos equipamentos envolvidos e suas funcionalidades. Na seqüência, aplicações de SPMS em diferentes países são apresentadas, abordando-se em maiores detalhes os desenvolvimentos dos Estados Unidos, China e Brasil.

No Capítulo 3, primeiramente são apresentadas soluções relativas a Concentradores de Dados Fasoriais, compreendendo-se soluções próprias de projetos em desenvolvimento e soluções comerciais disponíveis no mercado. Na seqüência, apresentam-se os requisitos para o desenvolvimento de um Sistema de Concentração de Dados Fasoriais (PDCS), juntamente com a proposta para este trabalho, abordando-se aspectos conceituais de arquitetura e de operação considerados durante seu desenvolvimento. Uma incursão é feita pelos formatos dos *frames* da norma IEEE C37.118, procurando-se ilustrar conceitos amplamente utilizados na comunicação entre equipamentos de SPMS. São apresentados os detalhes envolvendo a lógica de comunicação entre equipamentos e seus requisitos, referenciando-se, desde a aquisição de dados das PMUs, até a disponibilização de dados em tempo real pelo PDC de maior hierarquia. Diferentes etapas de execução do processo computacional do concentrador de dados são representadas através de fluxogramas, auxiliando a compreensão de todo o processo.

No Capítulo 4 é apresentado o projeto de *software* do PDCS, sob a ótica do equipamento PDC. São apresentados detalhes da modelagem dos equipamentos e dos processos envolvidos que viabilizam a comunicação entre PDCs. Para isto, são utilizados os fundamentos da Modelagem Orientada a Objetos (MOO), cujo objetivo é a representação de um projeto de *software* por meio de suas partes. Estas partes são denominadas *objetos*, com a descrição realizada em suas respectivas classes. Estas classes permitem a abstração dos elementos do mundo real, de forma a representá-los em escrita no código-fonte do *software*. As classes desenvolvidas no trabalho são apresentadas em conjunto com seus principais atributos e métodos, estabelecendo-se suas relações por meio de diagramas UML (*Unified Modeling Language*).

No Capítulo 5 são apresentados os experimentos e resultados do trabalho. São feitas avaliações do protótipo do sistema como um todo, buscando-se verificar o cumprimento dos requisitos considerados para o desenvolvimento do PDCS. Em um segundo momento avaliações são realizadas na operação do equipamento PDC verificando-se as perdas no processo de aquisição de dados, a fim de observar-se o processamento dos sincrofasores nos diferentes níveis de aquisição de dados. São avaliadas as estruturas de manipulação de dados em tempo real e em relação a bancos de dados históricos, sendo realizada a análise do registro de uma contingência no Sistema Interligado Nacional (SIN) a partir de dados fasoriais armazenados pelo PDCS.

No Capítulo 6 as conclusões do trabalho são apresentadas. Aspectos principais de cada capítulo são retomados de forma a contextualizar sua inserção no âmbito do documento. Uma análise sobre as contribuições do trabalho é realizada, apontando-se as implementações realizadas e os objetivos atingidos, bem como são elaboradas sugestões para trabalhos futuros.



# CAPÍTULO 2

## 2. Sistemas de Medição Fasorial Sincronizada

O conceito de Medição Fasorial Sincronizada foi apresentado inicialmente no final dos anos 80, a partir do desenvolvimento do primeiro protótipo de PMU, no âmbito do *Virginia Tech* nos Estados Unidos, cuja pesquisa contava com o apoio financeiro do *US Department of Energy* (DoE), do *US Electric Power Research Institute* (EPRI) e da *US National Science Foundation* (NSF). Baseado neste desenvolvimento, a empresa *Macrodyne* tornou o protótipo em um produto comercial, desenvolvendo-se também um PDC para armazenar os fasores medidos pelas PMUs.

A partir de 1994, aliado com a disponibilização para uso civil do sinal de sincronização do Sistema Global de Posicionamento (*Global Positioning System - GPS*), três empresas iniciaram seus projetos para instalações de equipamentos em seus respectivos sistemas, sendo elas: *Bonneville Power Administration* (BPA), *American Electric Power* (AEP) e *New York Power Authority* (NYPA). Todas as instalações foram implantadas e financiadas como projetos de pesquisa, sendo os esforços de desenvolvimento de aplicações para esta nova tecnologia liderados pela *Virginia Tech* e a Universidade de *Cornell* (PHADKE, A. G., 2002). Neste período, a potencialidade de se monitorar diretamente os estados do sistema, aliada a taxas de amostragem superiores aos sistemas de monitoramento convencionais, serviu de incentivo para novas pesquisas na área, registrando-se iniciativas em SEE em diferentes países.

Atualmente a tecnologia de SPMS encontra-se em processo de evolução, com sistemas de fase experimental se transformando em aplicações de larga escala. Tal evolução traz requisitos tecnológicos não observados previamente, tais como (BEGOVIC, M., *et al.*, 2008):

- definição de normas e padronizações para precisão e interoperabilidade de equipamentos;
- precisão dos canais de instrumentação;
- necessidade de canais de comunicação de banda larga e baixa latência;

- capacidade computacional e desenvolvimento de tecnologias para gerenciar, processar e analisar grandes quantidades de dados.

Neste capítulo, primeiramente são abordados os conceitos da tecnologia de Medição Fasorial Sincronizada relativos à sua estrutura, juntamente com os elementos que compõem o sistema. Em um segundo momento, é apresentada uma revisão da literatura sobre os desenvolvimentos da tecnologia de SPMS em diversos países, com destaque aos desenvolvimentos dos Estados Unidos e China, bem como a aplicação da tecnologia no Brasil.

## 2.1 Estrutura Fundamental de um SPMS

De maneira conceitual, um SPMS pode ser descrito como um sistema para medição simultânea de grandezas elétricas distantes geograficamente, sendo estas adquiridas em uma mesma base de tempo através de um sistema de sincronização, que atua como referência para o cálculo dos fasores de corrente e de tensão. Fasores adquiridos desta maneira são denominados sincrofadores (IEEE, 2005).

Como descrito em (EHRENSPERGER, J. G., 2004), a estrutura fundamental de um SPMS é composta pelos seguintes componentes:

- Fonte de Sincronização;
- Unidade de Medição Fasorial – PMU;
- Concentrador de Dados Fasoriais – PDC;
- Canais de Comunicação.

Para ilustrar este conceito, na Figura 2.1 mostra-se a relação dos elementos dentro de um SPMS, observando-se a interação entre eles.



Figura 2.1 – Componentes básicos que formam um SPMS. Adaptado de (SANTOS, M. M., 2008).

### 2.1.1 Sistema de Sincronização

Com objetivo de garantir que as medidas sejam sincronizadas no tempo, é necessária a disponibilidade de uma fonte de sincronização, de precisão elevada e de grande abrangência geográfica. Desta forma, medidas adquiridas em instalações distantes entre si têm a possibilidade da associação de uma etiqueta de tempo. Para o caso de SPMS, a sincronização temporal é fundamental, pois define a referência para o cálculo da fase do sinal elétrico dos sincrofasores.

Para executar tal tarefa, o sistema GPS é a fonte de sincronização mais utilizada atualmente, sendo composto por uma constelação de 24 satélites principais e 5 reservas, distribuídos em seis planos orbitais, que completam uma volta na Terra em um período de doze horas. Estes satélites disponibilizam sinais de navegação de posição, velocidade e tempo (GPS, 2010; NAVSTAR, 2010). A precisão do sinal de tempo possui ordem de aproximadamente  $100\text{ ns}$ . Isto equivale  $0,0021^\circ$  elétrico de precisão em um sistema de  $60\text{ Hz}$ , sendo considerado suficiente para as aplicações em medição fasorial sincronizada (PHADKE, A. G., 2002). O sistema GPS disponibiliza o sinal de tempo *Pulse Per Second* (PPS), um sinal de tensão contínuo e de  $250\text{ ms}$  de duração, gerado toda vez que um segundo é completado dentro do sistema. Uma etiqueta de tempo acompanha este sinal, contendo o ano, o dia, a hora, o minuto e o segundo, sendo que o tempo utilizado é *Universal Time Coordinated* (UTC). Assim, o sistema GPS possibilita a medição sincronizada de fasores em instalações fisicamente remotas, pois fornece a referência temporal para os equipamentos.

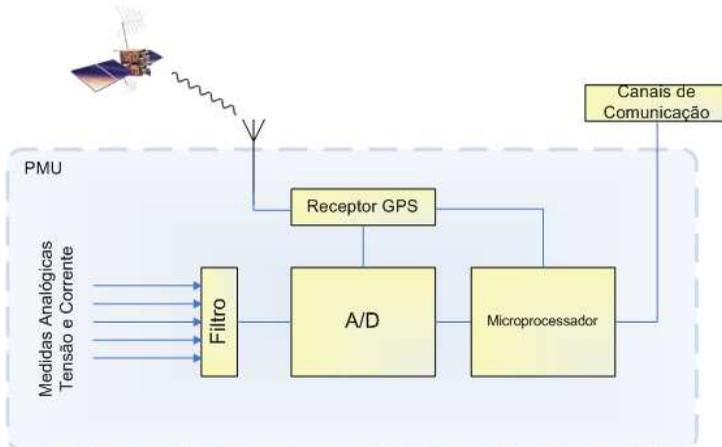
Outros sistemas de sincronização baseados em satélites são apontados na literatura, além do GPS. Atualmente em operação, o sistema russo *Global Navigation Satellite System* (GLONASS) é composto por 23 satélites (21 principais e 2 reservas), contemplando três planos de órbitas (ZAIDI, A. S., *et al.*, 2006). O GALILEO, sistema proposto pela União Européia, tem composição projetada de 30 satélites (27 principais e 3 reservas) em três planos de órbitas (ESA, 2010; VEJRAZKA, F., 2007). Este sistema se encontra em fase de testes, contando com 4 satélites construídos para validação do projeto, sendo o lançamento dos dois primeiros previsto para o início de 2011. A China caminha no sentido de implementar seu sistema de sincronização baseado em satélites. Batizado de Beidou I, atualmente possui quatro satélites em órbita (XIAOHUI, B., *et al.*, 2009). Diferentemente dos demais sistemas, a abrangência geográfica do sistema é restrita ao território chinês. O sinal de sincronização está disponível para uso civil e apresenta precisão suficiente para

aplicações em sistemas de medição fasorial sincronizada.

### 2.1.2 Unidade de Medição Fasorial

A PMU é o dispositivo responsável pelo cálculo dos sincrofasores referentes aos sinais analógicos de tensão e corrente adquiridos das barras do sistema. Ela utiliza o sinal do sistema GPS como referência, processando as medidas e as convertendo em grandezas fasoriais sincronizadas no tempo.

Os sinais analógicos são derivados dos circuitos secundários dos Transformadores de Potencial (TP) e dos Transformadores de Corrente (TC) das subestações, sendo processados por filtros “*anti-aliasing*”<sup>1</sup> e conversores Analógico/Digital (A/D). No processo de conversão A/D, o receptor GPS provê um sinal em padrão IRIG-B (IRIG, R. C. C., 2004), desenvolvido pelo grupo IRIG (*Inter Range Instrumentation Group*), que fornece as informações necessárias para se elaborar a etiqueta de tempo do sincrofasor. Após este procedimento, o *software* interno da PMU determina os fasores de cada um dos canais de medição utilizando algoritmo próprio, como por exemplo, a Transformada Discreta de Fourier (*Discrete Fourier Transform – DFT*) (PHADKE, A. G., *et al.*, 2006). A Figura 2.2 ilustra o diagrama de blocos representando a arquitetura interna da PMU.



<sup>1</sup> *Aliasing*: Presença de falsas frequências oriundas do processo de amostragem de um sinal analógico, devido à baixa taxa de amostragem adotada. Para evitar este fenômeno o sinal deve ser amostrado de acordo com a taxa de Nyquist, ou seja, utilizar uma taxa de amostragem duas vezes superior à componente de maior frequência do sinal.

Figura 2.2 – Arquitetura interna da PMU.

### 2.1.3 Concentrador de Dados Fasoriais

O PDC envolve, de maneira geral, todo o sistema de concentração de dados de um SPMS. Este equipamento, em sua essência, deve receber os fasores, enviados pelas PMUs do sistema através dos canais de comunicação, de forma contínua. Uma vez recebido os dados, o PDC deve correlacionar os diferentes sincrofasores, fazendo-se uso da etiqueta de tempo proporcionada pelo sistema GPS, e armazená-los em um banco de dados de forma coerente. O concentrador de dados deve proporcionar uma plataforma em tempo real, para aplicações de monitoração e/ou controle, e disponibilizar sua base histórica para subsidiar estudos. A base de dados pode ser cíclica (dados antigos sobrescritos após um determinado período de tempo) ou permanente (exigindo uma capacidade de armazenamento de dados de grande volume) (DECKER, I. C., *et al.*, 2004). A Figura 2.3 ilustra as funcionalidades do PDC.



Figura 2.3 – Concentrador de Dados Fasoriais e suas funções.

Utiliza-se, de maneira geral, um microcomputador como equipamento para a instalação do PDC. Podem ser utilizados microcomputadores de arquitetura PC convencionais, computadores de maior desempenho computacional, ou até mesmo *hardware* específico. Independente da opção utilizada, características como robustez, escalabilidade, simultaneidade e rapidez são requisitos básicos e indispensáveis para a executar a aquisição de fasores em tempo real (SANTOS, M. M., 2008).

Para a comunicação entre PMUs e PDCs é necessário que os equipamentos estejam de acordo com a norma IEEE C37.118, que estabelece diferentes tipos de formato de mensagens (dados, configuração e cabeçalho). Cada formato tem características distintas que carregam informações necessárias tanto para configuração da operação SPMS,

quanto para sua transferência de dados. Esta padronização permite a integração de equipamentos produzidos por diferentes fabricantes, garantindo a interoperabilidade e universalidade da tecnologia.

### **2.1.4 Canais de Comunicação**

Os canais de comunicação exercem a função de transporte da informação, adquirida pelas PMUs distribuídas no sistema, até o ponto de entrega destinado aos sincrofasores. No caso de um SPMS, o destino é o PDC.

Diversas soluções podem ser empregadas para satisfazer os critérios de comunicação de um SPMS, embora sua aplicação para SEE possa ser inviável financeiramente. Dentre estas soluções, podem ser consideradas tanto redes cabeadas (linhas telefônicas dedicadas, cabos de fibra óptica, *Power Line Communication* - PLC), como redes sem fio (enlaces de microondas, satélites *Low-earth Orbiting* - LEO) (BEGOVIC, M., *et al.*, 2008; NADUVATHUPARAMBIL, B., *et al.*, 2002).

Os requisitos de desempenho da tecnologia de comunicação utilizada dependem das aplicações. Uma aplicação destinada a controle em grandes áreas de monitoração demanda maior velocidade dos canais e uma alta taxa de atualização de dados. Por outro lado, uma aplicação de visualização pode abdicar da necessidade de uma maior latência, no que se refere à comunicação, porém mantendo-se uma elevada taxa de amostragem (DAGLE, J., 2009).

## **2.2 Desenvolvimentos e Aplicações em SPMS**

Esta seção tem por objetivo elencar algumas aplicações de SPMS documentadas na literatura, dando-se enfoque às soluções tecnológicas implementadas em diferentes sistemas e as características de cada implementação. Aplicações nos Estados Unidos, China e Brasil são apresentadas em maiores detalhes, citando-se também desenvolvimentos em demais países.

### **2.2.1 Estados Unidos**

Os Estados Unidos lideraram as primeiras iniciativas de pesquisa na área com o projeto *Wide Area Measurement System* (WAMS), iniciado em 1989. Este projeto tinha por objetivo principal determinar os requisitos de pesquisa e desenvolvimento para mercados de energia em ambiente desregulamentado (DECKER, I. C., *et al.*, 2004). A partir de 1994, com esforços conjuntos do EPRI, o projeto WAMS evoluiu, sendo realizadas instalações de PMUs nos subsistemas do *Western Electric Coordinating Council* (WECC), sobretudo nos subsistemas da *Southern*

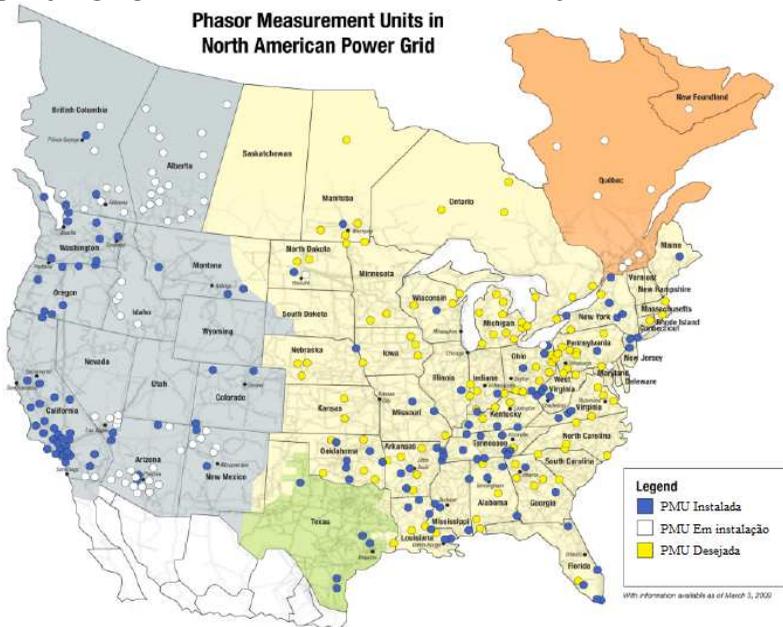
*California Edison* (SCE) e da *Bonneville Power Administration* (BPA). Estas duas empresas foram as primeiras a desenvolver ferramentas de monitoração baseadas em sincrofasores (BAHRGAVA, B., 1999; BALLANCE, J. W., *et al.*, 2003). Os objetivos destes desenvolvimentos eram: a monitoração do estado do sistema em tempo real; obter um registro de eventos durante contingências; estender a aplicação dos sincrofasores em aplicações de controle em tempo real.

O *blackout* de agosto de 2003, que atingiu o nordeste dos EUA e o Canadá, afetou aproximadamente 50 milhões de pessoas. Este evento motivou a busca por melhorias em relação à confiabilidade do sistema elétrico norte-americano. No mesmo ano, foi criado o *Eastern Interconnect Phasor Project* (EIPP), com o apoio do *US Department of Energy* (DoE), *Consortium of Electric Reliability Technology Solutions* (CERTS) e *North American Electric Reliability Corporation* (NERC), abrangendo empresas da região leste dos Estados Unidos. O projeto tinha por missão criar um SPMS amplamente acessível. Este sistema deveria ter características de robustez e segurança, e teria sua aplicação ao longo da Interconexão Leste. Com isto, buscava-se facilitar a ambientação e o desenvolvimento da tecnologia, criando ferramentas de registro e análise de eventos que ajudassem a aumentar a confiabilidade do sistema, bem como seu planejamento e operação (EIPP, 2006; PHADKE, A. G., *et al.*, 2008).

Em 2007, o DoE e o NERC, em conjunto com as empresas envolvidas, decidiram unir as iniciativas do WECC e EIPP, juntamente com o Canadá e México, formando então o *North American Synchronphasor Initiative* (NASPI), um grupo de trabalho composto por membros voluntários que trabalham dentro de diferentes ramos do setor elétrico da América do Norte. O grupo é organizado em cinco forças-tarefa que possuem diversas atribuições, tais como: implantar novas PMUs no sistema, estabelecer padrões para o desenvolvimento de equipamentos, criar ferramentas que façam uso dos sincrofasores em ambiente de operação, entre outras (DAGLE, J., 2009; NASPI, 2007). Abaixo são listadas as forças-tarefas que atualmente compõem os trabalhos do NASPI.

- *Operations Implementation Task Team;*
- *Planning Implementation Task Team;*
- *Performance & Standards Task Team;*
- *Data & Network Management Task Team;*
- *Research Initiatives Task Team.*

Até abril de 2010, o sistema norte americano contava com aproximadamente 200 PMUs ao longo de seu sistema. A Figura 2.4 ilustra a posição geográfica das PMUs do sistema de medição fasorial do NASPI.



*Figura 2.4 – PMUs do sistema Norte Americano. Adaptado de (HU, Y., et al., 2010).*

Em outubro de 2009, o governo dos Estados Unidos anunciou investimentos da ordem de US\$ 3,4 bilhões para modernização de suas redes de transmissão. No escopo destes investimentos está prevista a instalação de 850 PMUs por todo o sistema elétrico americano. Estes equipamentos serão posicionados em instalações com as seguintes características (FOSTER, T., 2010):

- principais troncos de transmissão;
- subestações de nível de tensão igual ou superior a 500kV;
- subestações com tensão maior ou igual a 200kV, em sua maioria;
- unidades geradoras com capacidade instalada igual ou superior a 500MW;
- grandes centros de carga;
- parques de geração eólica e solar;

- instalações remotas que possibilitem a observabilidade do sistema.

### 2.2.2 China

As pesquisas referentes à tecnologia de SPMS da China tiveram início em 1994. O principal aspecto motivador para o interesse pela tecnologia é a grande potencialidade na melhoria da supervisão, da operação, da proteção e do controle do SEE chinês.

O início das instalações de equipamentos de SPMS no sistema elétrico chinês é relatado desde 1995, com a introdução do módulo ADX3000 pelo *Chinese Electric Power Research Institute* (CEPRI). Este módulo é fabricado pela empresa *Ouhua Technology Co. Ltd.* de Taiwan e possui funcionalidade de PMU, sendo que 40 destas unidades foram instaladas nos diversos subsistemas chineses entre os anos de 1995 e 2002 (YANG, Q., *et al.*, 2007).

Entretanto, maiores desenvolvimentos utilizando PMUs só foram iniciados a partir de 2003. Isto se justificava pela necessidade de melhorias na infra-estrutura do sistema de comunicação chinês. Outro aspecto de relevância foi a publicação de uma especificação para o uso da tecnologia de SPMS, baseada na norma IEEE 1344 (IEEE, 1995). Este documento permitiu a produção em escala comercial de PMUs com a entrada de diversos fabricantes no mercado chinês (XIE, X., *et al.*, 2006).

Observando o ritmo dos desenvolvimentos em SPMS, bem como a necessidade de compatibilizar equipamentos para os requisitos do sistema elétrico chinês, o *Nation Electrical Power System Management and Exchange of Information Standardization Technical Committee* criou, em 2006, o *Power System Dynamic Monitoring Standardization Specialized Working Group* (WAMS/GPS). Este grupo de trabalho é formado por especialistas da área de pesquisa, indústria, departamentos de operação de sistemas elétricos e fabricantes de PMUs, e tem por objetivo coordenar e protocolar normas técnicas relacionadas à tecnologia de medição fasorial sincronizada (WU, J., *et al.*, 2008).

Estas normas padronizam funções operativas como:

- especificações de PMUs;
- guia de testes para PMUs;
- protocolos de comunicação;
- sistemas de sincronização.

Em termos de infra-estrutura, o SPMS chinês dispunha cerca de 400 PMUs em março de 2007 instaladas em seu sistema. Estas unidades foram instaladas em subestações de nível de tensão de 330 kV e 500 kV

e unidades geradoras com potência superior a 300MW.

O SPMS chinês é composto por 7 SPMS em nível regional nos subsistemas Norte, Nordeste, Central, Sul, Leste e o subsistema do *State Power Dispatching Center* (SPDC). A estes sistemas, soma-se ainda SPMS instalados nos subsistemas provinciais de Jiangsu, Henan, Guangdong, Yunnan e Ghizou.

Atualmente o sistema chinês possui um número superior a 1000 PMUs em operação em seus sistema. Estas unidades estão instaladas em todas as subestações de tensão igual ou superior a 500 kV e em todas as unidades geradoras com potência igual ou superior a 100 MW (BI, T., 2010).

A Figura 2.5 ilustra o diagrama esquemático do SPMS chinês, indicando os PDCs comissionados até o ano de 2010, que realizam a concentração de dados das diversas PMUs instaladas no sistema.



Figura 2.5 – Concentradores de Dados do SPMS Chinês. Adaptado de (BI, T., 2010).

As aplicações desenvolvidas no ambiente do sistema chinês possuem o objetivo de:

- Visualizar repostas dinâmicas e carregamento de linhas de transmissão;
- Sinalizar eventos para gravação e reprodução de fenômenos dinâmicos;
- Analisar oscilações eletromecânicas em tempo real;
- Validar modelos de simulação.

### **2.2.3 Brasil**

Atualmente existem duas vertentes de trabalho com relação a sistemas de medição fasorial sincronizada em andamento no Brasil: Projeto Medfasee (MEDFREQ, 2008) e Projeto Fasorial para o Sistema Interligado Nacional (SIN) (MORAES, R. M., *et al.*, 2008). O primeiro se destaca por ser o pioneiro nos estudos sobre a tecnologia no país, reunindo iniciativas do meio acadêmico e da indústria com objetivo de desenvolver, divulgar, compreender e contribuir para o desenvolvimento da tecnologia. O segundo, liderado pelo Operador Nacional do Sistema Elétrico (ONS), concentra esforços para a implementação de um SPMS aplicado ao SIN, objetivando uma operação do sistema mais eficiente e com maiores índices de confiabilidade.

Embora as duas iniciativas possuam especificamente objetivos distintos, no geral, busca-se contribuir para o desenvolvimento da tecnologia de medição fasorial sincronizada no Brasil, melhorando-se a performance do Sistema Elétrico Brasileiro, tanto do ponto de vista operacional (segurança, confiabilidade, continuidade de suprimento, aumento dos limites operativos da rede), quanto econômico (otimização dos recursos energéticos, modicidade tarifária). A seguir, é apresentada uma descrição detalhada sobre os dois projetos.

#### **2.2.3.1 Projeto MedFasee**

O Projeto MedFasee teve início em 2003 na Universidade Federal de Santa Catarina (UFSC), no âmbito do Laboratório de Planejamento de Sistemas de Energia Elétrica – LabPlan. Este projeto teve patrocínio da Financiadora de Estudos e Projetos (FINEP) e da fabricante de equipamentos de monitoração *Reason Tecnologia S.A* (DECKER, I. C., *et al.*, 2004). Os objetivos deste projeto eram: implementar um protótipo de SPMS; e desenvolver aplicações para monitoração e controle de sistemas elétricos de potência.

As primeiras atividades do projeto foram no sentido de documentar o estado da arte sobre medição fasorial, bem como desenvolver um simulador de SPMS (EHRENSPERGER, J. G., 2004). Em seguida, foram apresentados desenvolvimentos de ferramentas de monitoração e

resultados obtidos da instalação de um protótipo de SPMS, composto por três PMUs, adquirindo as medidas das tensões trifásicas da rede de baixa tensão das cidades de Curitiba, Florianópolis e Porto Alegre. Um PDC foi desenvolvido e alocado no LabPlan, com objetivo de realizar a aquisição e disponibilização dos dados do sistema (DECKER, I. C., *et al.*, 2005). Os sincrofasesores eram enviados através de *Internet* pública, a uma taxa de 60 sincrofasesores por segundo. Na Figura 2.6, ilustra-se a interface inicial de monitoração em tempo real dos fasores e a configuração do SPMS implementado pelo projeto MedFasee.

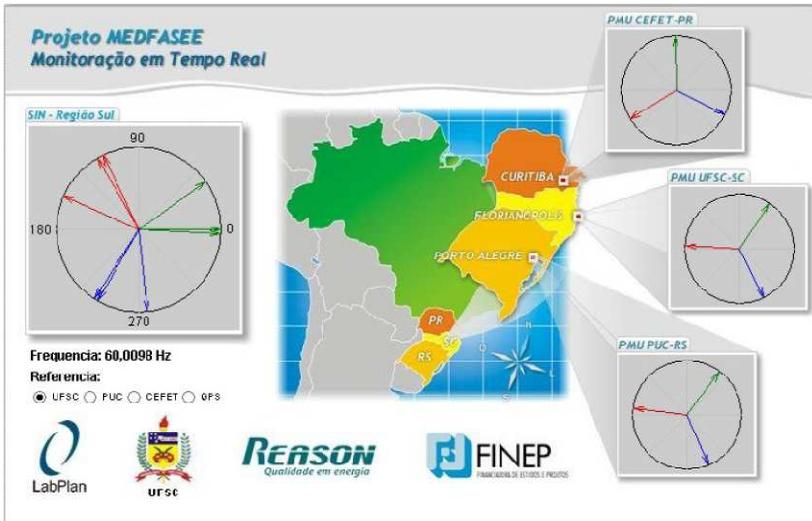


Figura 2.6 – Protótipo de SPMS do implementado pelo Projeto MedFasee.

Percebendo-se os resultados obtidos no Projeto MedFasee, juntamente com a experiência adquirida, novos projetos envolvendo medição fasorial foram iniciados, sendo eles: Projeto MedFasee Eletrosul, Projeto MedFasee Frequência e Projeto MedFasee CTEEP.

O projeto MedFasee Eletrosul teve início em 2007, em parceria entre Universidade Federal de Santa Catarina, Eletrosul Centrais Elétricas S.A. e Reason Tecnologia. Seus principais objetivos são: implementação de um protótipo de SPMS no sistema de transmissão da Eletrosul; desenvolver aplicações de monitoração e localização de faltas em linhas de transmissão (DECKER, I. C., *et al.*, 2008).

A instalação do protótipo de SPMS do projeto contempla quatro PMUs, localizadas nas subestações de Ivaiporã, Areia, Nova Santa Rita

e Campos Novos, em nível de tensão de 525kV. As PMUs adquirem dados de tensão e corrente de oito circuitos de transmissão, totalizando 45 fasores. As taxas de transmissão das PMUs são configuráveis entre 10, 12, 15, 20, 30 e 60 fasores por segundo, sendo 60 a taxa utilizada para o projeto. Os dados são enviados através de protocolo UDP/IP para o PDC localizado no escritório central da Eletrosul (DECKER, I. C., *et al.*, 2009). A Figura 2.7 ilustra a localização geográfica das PMUs do projeto.



Figura 2.7 – Protótipo de SPMS instalado no sistema de transmissão da Eletrosul.

No ano de 2008, o projeto MedFasee Frequência teve início, como expansão dos desenvolvimentos do Projeto MedFasee. O projeto, em parceria entre UFSC/LabPlan, Reason Tecnologia e Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), resultou com a instalação de 10 PMUs instaladas em universidades brasileiras, com objetivo de monitorar e avaliar a dinâmica do SIN, e ao mesmo passo divulgar e incentivar a pesquisa nesta área de medição fasorial sincrô-

zada (MEDFREQ, 2008).

A Figura 2.8 ilustra a plataforma de monitoração em tempo real em outubro de 2010 dos sincrofasores do projeto MedFasee Freqüência, juntamente com a localização das PMUs nas universidades do País.

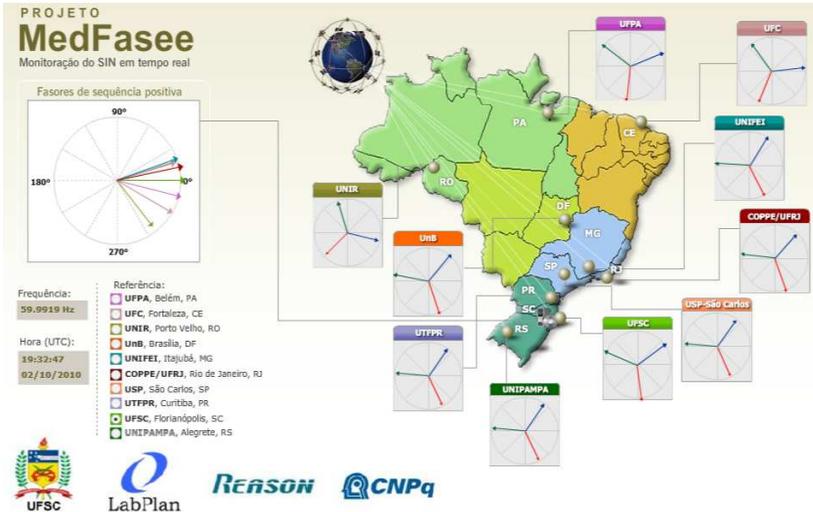


Figura 2.8 – Monitoração em tempo real dos sincrofasores do Projeto MedFasee Freqüência.

No ano de 2009, o projeto MedFasee CTEEP teve início. Esta iniciativa conta com a parceria entre Universidade Federal de Santa Catarina, Reason Tecnologia e Companhia de Transmissão de Energia Elétrica Paulista (CTEEP). Atualmente, o projeto encontra-se em fase de desenvolvimento, tendo por objetivo instalar um protótipo de SPMS contendo 5 PMUs e um PDC de arquitetura descentralizada ao longo do sistema de transmissão da CTEEP, em subestações de 345kV. As aplicações previstas envolvem a localização de faltas em linhas de transmissão, detecção de modos de oscilação, monitoração de aberturas angulares e monitoração das condições operacionais do sistema elétrico.

### 2.2.3.2 Projeto SIN

A iniciativa mais relevante de utilizar a tecnologia de medição fatorial sincronizada no SIN teve início em 2005 através do Projeto 6.2 – “Implantação de Sistema de Oscilografia de Longa Duração”, presente no Plano de Ação 2005/2007 do Operador Nacional do Sistema. Como objetivo, busca-se a implantação de um SPMS no SIN, a fim de capturar

dados do comportamento dinâmico do sistema, fornecendo suporte para estudos de análise sistêmica e proteção.

A Resolução Autorizativa nº170, publicada em 27 de abril de 2005 pela Agência Nacional de Energia Elétrica (ANEEL), delega atribuições ao ONS em relação ao projeto do SIN. Dentre elas destaca-se a coordenação da homologação das PMUs, por meio de ensaios de laboratório, e a avaliação do número de unidades a serem instaladas nas subestações do sistema. Esta resolução determina também que cabe às concessionárias a aquisição dos equipamentos para o SPMS e a disponibilização do canal de comunicação para envio dos sincrofasores ao centro de controle do ONS.

A partir de 2006, um segundo projeto foi adicionado ao Plano de Ação do ONS, definido como Projeto 11.11 – “Aplicação da tecnologia de Medição Fasorial a sistemas de apoio a tomadas de decisão em Tempo Real”. O objetivo principal deste projeto é fazer uso da tecnologia visando melhorias nos sistemas de supervisão e controle em tempo real, a fim de aumentar a segurança na operação, seja evitando contingências severas no sistema ou relaxando determinadas restrições operacionais (MORAES, R. M. D., *et al.*, 2007). Para a implantação do projeto, uma consultoria internacional foi contratada, elaborando especificações para os equipamentos e a infra-estrutura que deverá ser utilizada, determinando a arquitetura geral do SPMS e a identificação de aplicações que possam utilizar sincrofasores para tomada de decisão em tempo real.

Relativo à arquitetura a ser adotada, esta é composta por três níveis, com os equipamentos comunicando-se através de protocolo IP. O primeiro nível é composto pela rede local das subestações onde as PMUs serão alocadas, juntamente com um PDC de Subestação, que fará a coleta dos fasores correspondentes. O segundo nível é representado pela rede do centro de controle do agente, podendo opcionalmente conter um PDC de uso exclusivo. Finalmente, no terceiro nível estará a rede do centro de controle do ONS contendo dois Concentradores de Dados Centrais, um primário e outro secundário (MORAES, R. M. D., *et al.*, 2007). A Figura 2.9 ilustra a arquitetura do SPMS definido para o SIN.

Concerne às aplicações, quatro soluções foram propostas, descritas a seguir (RODRIGUES, M. A. M., *et al.*, 2009):

- *StressMon*: monitora a diferença angular entre barras do sistema predeterminadas, com objetivo de indicar o quão próximo aquela interligação está do limite de estabilidade. Estudos de contingências podem ser aplicados a condições operativas, observando a viola-

ção de limites operativos. Estes estudos proporcionam informações que os operadores podem usar a fim de ajustar a rede para um estado operativo seguro;

- *LoopAssist*: monitora a magnitude de tensão e a diferença angular entre terminais de sistemas de transmissão. Estas informações são utilizadas para determinação de níveis de estresse do sistema e limites de transferência dinâmicos;
- *SynchAssist*: monitora a magnitude de tensão, a diferença angular e o desvio de frequência entre barras de fronteiras que delimitam ilhas de operação do sistema, indicando os ajustes necessários de geração nas regiões, a fim de determinar o momento para o fechamento bem sucedido de uma interligação.
- *DampMon*: monitora grandezas elétricas para detectar e medir oscilações de baixa frequência. Em geral, de natureza eletromecânica, possibilitando-se assim aos operadores verificar o amortecimento do sistema para determinadas contingências a serem simuladas.

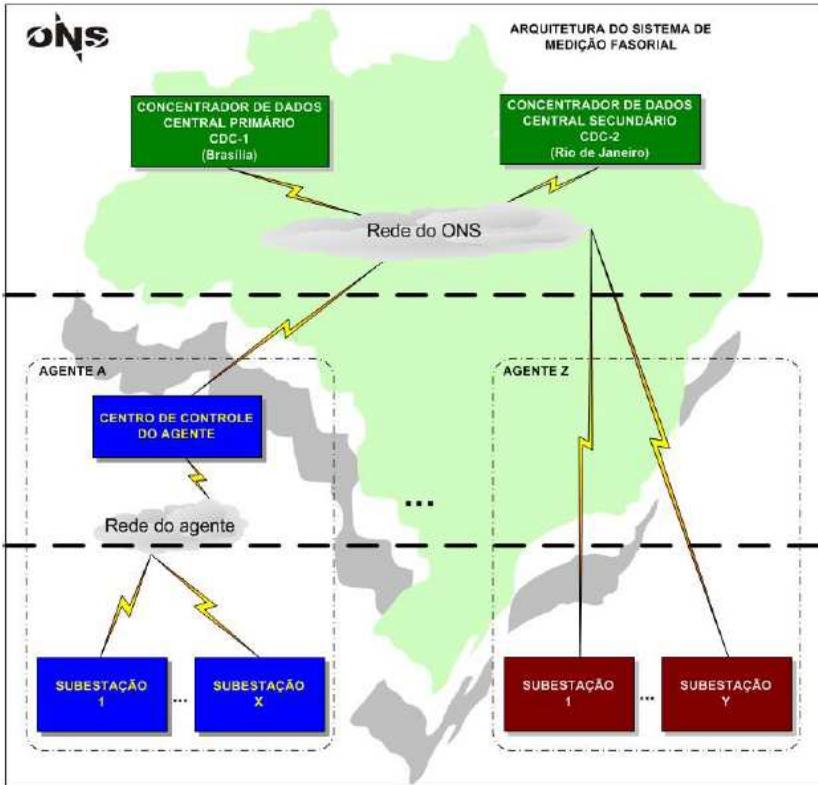


Figura 2.9 – Arquitetura do Sistema de Medição Fasorial do SIN.

## 2.2.4 Outros Países

Na última década, com a divulgação dos resultados obtidos através da monitoração dos sistemas elétricos por meio de sincrofasores, o número de iniciativas de estudo e implantação da tecnologia em diferentes países vem aumentando. Projetos em diferentes níveis de desenvolvimento podem ser encontrados em diversos países, desde aqueles em etapa de planejamento até SPMS já em operação.

Considerando o sistema elétrico dos países nórdicos, especificamente no leste da Dinamarca, tem-se instalado um total de 4 PMUs em subestações de 132kV e 400kV e outras 3 unidades em redes de baixa tensão. Os sincrofasores medidos são utilizados para o estudo e reprodução de eventos, bem como na análise das oscilações eletromecânicas entre áreas de controle (RASMUSSEN, J., *et al.*, 2006; XU, Z., *et al.*,

2008). A frequência do sistema nórdico também foi avaliada através dos sincrofasores medidos pelas PMUs (XU, Z., *et al.*, 2008). Descobriu-se, por meio deste estudo, a violação de um dos critérios de operação estabelecidos. Este critério determina que o número de minutos que o sistema elétrico nórdico deve operar abaixo do valor de  $59,90\text{ Hz}$ , no período de um ano, é de 1200. Neste estudo, foi verificado o valor de 1645 minutos. Em análises posteriores, concluiu-se que o despacho de curto prazo das usinas para atendimento à demanda é o responsável pelo comportamento observado.

Em 2005, a Stattnet, operador do sistema de transmissão noruegues, iniciou um projeto de pesquisa e desenvolvimento com objetivo de estudar e implementar a tecnologia de SPMS em seu sistema de 420kV. A motivação do projeto deu-se pela necessidade de soluções para a detecção preventiva de condições anormais de operação, que possam vir a liderar cortes de carga e *blackouts*. Foram instaladas 4 PMUs que fornecem dados para a monitoração do sistema, subsidiando análises para a estabilidade de pequenos sinais do sistema elétrico nórdico (LEIR-BUKT, A. B., *et al.*, 2006). Considerando ainda Suécia, Finlândia e Dinamarca, existem, no total, 21 PMUs instaladas no sistema de transmissão dos países nórdicos, havendo a perspectiva de expandir este número para 72 nos próximos anos (CHENINE, M., *et al.*, 2009). A Islândia possui seu sistema elétrico isolado dos outros países pertencentes ao NORDEL. Utilizam-se 7 PMUs na monitoração do sistema, que enviam dados para o centro de controle de operação do sistema. Neste local são calculados os amortecimentos de modos de oscilação, sendo apresentados em monitores em tempo real (PHADKE, A. G., *et al.*, 2008).

O projeto de instalação da tecnologia de SPMS no sistema elétrico russo teve início em 2005. O objetivo principal do projeto é a averiguação do comportamento dinâmico de seu sistema elétrico. Neste sistema foram instaladas 26 PMUs, instaladas em grandes geradores e subestações, medindo sincrofasores de corrente e tensão. Os sincrofasores são utilizados para análise e monitoração da performance do sistema juntamente com a validação de modelos de simulação dinâmica (PHADKE, A. G., *et al.*, 2008).

O SPMS do Japão compreende 11 PMUs instaladas na rede de baixa tensão de universidades parceiras do projeto. Os equipamentos estão inseridos nos sistemas de  $50\text{ Hz}$  (leste) e  $60\text{ Hz}$  (oeste) do sistema elétrico japonês. As PMUs monitoram tensões monofásica no nível de 100V, calculam os sincrofasores, enviando os dados através da internet

para os PDCs, alocados nas cidades de Nagoya e Kyushu. A taxa de aquisição utilizada é de 25 sincrofases por segundo, para o sistema de 50 Hz, e 30 sincrofases por segundo para o sistema de 60 Hz. Os dados coletados são utilizados para a estimação de parâmetros de identificação de oscilações eletromecânicas de baixa frequência, de ocorrência comum entre os sistemas do leste e oeste (OTA, Y., *et al.*, 2007).

Existem publicações com relação à aplicação da tecnologia de medição fasorial sincronizada em diversos outros países como México (MARTÍNEZ, E., 2008), Irlanda (DING, X., *et al.*, 2007), Índia (SOONEE, S. K., *et al.*, 2008), Romênia (BALASIU, F., *et al.*, 2009), Malásia (BIN, S. K., *et al.*, 2008), Croácia (SKOK, S., *et al.*, 2007), entre outros (CIGRÉ, 2007).

### **2.2.5 Comentários Finais**

Percebe-se que os desenvolvimentos na área da Medição Fasorial Sincronizada envolvem diferentes áreas de conhecimento. A infraestrutura utilizada para a implantação de SPMS demanda soluções das áreas de TI, sistemas de comunicação, processamento digital de sinais, entre outros.

São observados relatos de estudos e aplicação da tecnologia em diversos países. Na vanguarda das aplicações encontram-se os Estados Unidos e a China. A utilização de sincrofases para o desenvolvimento de ferramentas de apoio, tanto para análises e estudos sistêmicos como para a operação e controle em tempo real, vem adquirindo relevância estratégica para estes países.

No Brasil, projetos de pesquisa e desenvolvimento apresentam implementações de protótipos de SPMS em sistemas elétricos, bem como mostram os benefícios oriundos da tecnologia. A aplicação de SPMS de maior escala está prevista para o Sistema Interligado Nacional que, devido à magnitude e complexidade do sistema, encontra-se em fase de especificação.

Pode-se afirmar que, com a busca da modernização dos SEE dentro do paradigma dos *Smart Grids*, o aumento pela demanda da tecnologia de SPMS é inerente. Soluções, primeiramente adotadas para sistemas de pequeno porte, deverão ter seus conceitos revisitados. Neste contexto, pode-se citar como exemplo a adoção de novas arquiteturas de concentração de dados fasoriais, que possibilitam a facilidade de adição de novos equipamentos no SPMS. Esta característica agrega, principalmente, expansibilidade ao sistema, porém demanda soluções para problemas que previamente poderiam não ter se manifestado.



# CAPÍTULO 3

## 3. Sistema de Concentração de Dados Fasoriais

Este capítulo descreve as considerações, premissas e soluções adotadas para o desenvolvimento de um sistema de concentração de dados fasoriais. O escopo da arquitetura adotada, seu funcionamento, lógica de operação do sistema e PDC são apresentadas.

Primeiramente, apresenta-se a descrição de soluções em PDCs encontradas na literatura. Estas soluções são divididas em dois segmentos: desenvolvimento de projetos e pacotes de soluções comerciais. Em seguida, são apresentados os principais requisitos para o desenvolvimento de um sistema de concentração de dados fasoriais. Aspectos de arquitetura são apresentados, discutindo-se os conceitos de arquiteturas centralizadas e descentralizadas.

Uma vez estabelecidos os requisitos de desenvolvimento juntamente com os aspectos de arquitetura, a proposta do Sistema de Concentração de Dados Fasoriais (*ou Phasor Data Concentration System - PDCS*) é descrita, contemplando uma arquitetura de concentração de dados fasoriais descentralizada, organizada em dois níveis hierárquicos. São discutidas as vantagens envolvidas neste processo, caracterizando-se a evolução natural do processo, dado o grande volume de dados que são medidos por um SPMS. Estes dados necessitam manipulação adequada e crescem rapidamente com o acréscimo de novas PMUs ao sistema.

Após a definição da proposta do PDCS, a forma de comunicação entre equipamentos com funcionalidade de PDC é discutida. O uso de padrões de comunicação predefinidos, particularmente aqueles definidos na Norma IEEE C37.118, se faz necessário. Uma descrição sucinta dos formatos de mensagens é realizada, descrevendo-se a maneira que os dados são trocados entre equipamentos no âmbito do SPMS.

Finalmente, as premissas de funcionamento do PDCS são apresentadas. Estas premissas incluem: requisitos de operação que devem ser observados pelo sistema; lógica de configuração da operação do

PDCS; e lógica de operação do processo computacional do PDC.

### **3.1 Desenvolvimentos e Soluções em Concentradores de Dados Fasoriais**

Esta seção apresenta os principais desenvolvimentos adotados como solução para Concentradores de Dados. Apresentam-se casos publicados na literatura, propostos por projetos de desenvolvimento em (SPMS) e soluções disponíveis comercialmente.

#### **3.1.1 Soluções de Projetos**

##### ***TVA SuperPDC / openPDC***

O *TVA SuperPDC* começou a ser desenvolvido pela TVA, em 2004, ainda no âmbito do projeto EIPP. Esta solução é uma plataforma de aquisição de dados que atende a requisitos de confiabilidade e escalabilidade, é capaz de lidar com diferentes protocolos de comunicação, realiza consultas rápidas no banco de dados histórico, manipula e armazena uma extensa quantidade de dados e disponibiliza saídas de dados em tempo real (ZUO, J., *et al.*, 2008).

Desenvolvido utilizando o .NET framework, plataforma de programação em sistema operacional Windows em linguagem de programação C#, O *SuperPDC* recebe dados nos formatos das normas IEEE 1344 e C37.118, além de mensagens nos formatos *BPA Stream*, *Virginia Tech FNET*, *SEL Fast Message*, *OPC* e *Macrodyne*. Antes mesmo dos dados recém chegados serem armazenados em disco, eles são sincronizados e reenviados para monitores com aplicações em tempo real.

Para efetuar a configuração do sistema, informando ao *SuperPDC* as características do SPMS que ele irá gerenciar, são utilizados bancos de dados dos seguintes tipos: SQL Server; Access; MySQL; OleDb; Oracle; ODBC diversos.

O conteúdo de dados históricos é armazenado em um arquivo binário de formato próprio. Para conectar-se à base de dados, um banco de dados SQL, denominado *DatAware*, é utilizado. Este sistema provê a interface necessária para execução de funções de escrita e leitura, por meio de consultas eficientes e de performance computacional elevada.

Em outubro de 2009, com o objetivo de prover a indústria com tecnologia de concentração de dados fasoriais, promover a discussão na comunidade científica e apoiar as iniciativas da implementação do conceito de *Smart Grids*, a TVA anunciou a disponibilização do código-fonte do *SuperPDC*, como um projeto de código aberto. Esta iniciativa foi denominada de *openPDC* (OPENPDC, 2009). A partir do site do

projeto é possível acessar e obter o código fonte, bem como obter os guias de compilação e instalação do *software* e tutoriais, que apontam as configurações necessárias para a aquisição de dados de PMUs.

Há ainda um fórum de discussão, onde os desenvolvedores do projeto trocam informações com a comunidade. Este portal tem por objetivos solucionar dúvidas de usuários que estejam fazendo uso do sistema, coletar sugestões para novos desenvolvimentos e solucionar inconsistências que possam vir a ocorrer na operação do *software*.

Atualmente, a TVA opera um sistema de concentração de dados com base no *openPDC*, que processa 1,5 GBytes por hora, totalizando 36 GBytes e 3,6 bilhões de sincrofases diariamente, estando conectado a 120 PMUs (OPENPDC, 2009) do sistema do NASPI.

### ***NERC Phasor Data Concentration System***

O *openPDC* é a tecnologia núcleo utilizada para o desenvolvimento do NERC *Phasor Concentration System*. Este projeto encontra-se em estado de desenvolvimento, com objetivo de implementar um sistema concentrador de dados de grande escala, de arquitetura descentralizada, redundante e de capacidade elevada de processamento de dados. Neste sistema, as PMUs se comunicarão com PDCs locais, que reenviarão os dados coletados para PDCs regionais (nós regionais). Os dados são armazenados de maneira regional, referentes à área geográfica de monitoração. Estes nós regionais poderão se comunicar entre si, e enviarão seus dados para uma central armazenadora de dados global, utilizando-se de ferramentas de processamento paralelo para manipular grandes massas de dados (ROBERTSON, R., *et al.*, 2009).

### ***PDC Stattnet***

A *Stattnet*, operador do sistema de transmissão norueguês, vem trabalhando na implementação de um SPMS em seu sistema de transmissão. Este sistema teve duas fases de desenvolvimento. A primeira é relativa ao processo de alocação e instalação das PMUs em subestações consideradas estratégicas para o sistema. Nesta fase também foi utilizado um PDC desenvolvido em plataforma *Matlab*, realizando a concentração de dados para análise dos sincrofases calculados. A segunda fase compreende o desenvolvimento de um protótipo de PDC, baseado em *software* já desenvolvido na *Stattnet* para aquisição de dados.

O protótipo é compatível com frame de dados das normas IEEE 1344 e C37.118. Pode atuar tanto na função de servidor como cliente. Isto possibilita a adoção da arquitetura descentralizada para concentra-

ção de dados de SPMS. São admitidas diferentes taxas de aquisição de sincrofasores (LEIRBUKT, A., *et al.*, 2008). O PDC *Stattnet* é dividido em quatro camadas distintas, como ilustrado na Figura 3.1.

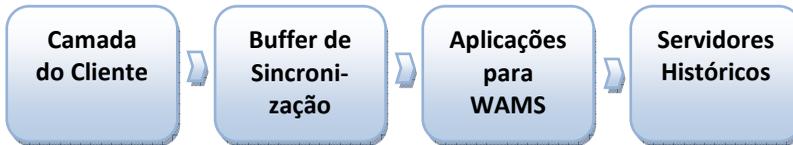


Figura 3.1 – Estrutura do PDC *Stattnet* - Adaptado de (LEIRBUKT, A., *et al.*, 2008).

A *Camada do Cliente* é responsável pela manipulação dos dados nos formatos 1344 e C37.118. Um cliente pode ser tanto uma PMU quanto um PDC, possibilitando a troca de informações entre diferentes SPMS.

O *Buffer de Sincronização* assegura que os dados sejam corretamente alinhados para serem repassados às aplicações. Isto se deve ao fato de o envio dos sincrofasores pela rede de comunicação ser realizado de maneira assíncrona.

A *Camada de Aplicações* possui bibliotecas desenvolvidas para efetuar cálculos algébricos envolvendo sincrofasores, bem como funções avançadas para monitoração da dinâmica do sistema, como, por exemplo, a detecção de modos de oscilação.

Finalmente, os dados são enviados para a quarta e última camada. Esta etapa disponibiliza os sincrofasores para sistemas SCADA/EMS e aplicações off-line que utilizam dados históricos. Duas soluções são adotadas para o armazenamento histórico de dados. A primeira utiliza um servidor com capacidade para armazenar 7 dias de histórico de operação. Os dados são particionados em pequenos períodos de tempo, onde podem ser observadas contingências no sistema de transmissão. Neste caso, considera-se uma alta taxa de amostragem no SPMS, com objetivo de observar os efeitos dinâmicos do sistema. A segunda alternativa utiliza amostragens inferiores, porém armazenando períodos maiores de tempo, compondo um histórico de operação do SPMS em base anual.

### ***PDC SPMS China***

O SPMS chinês é dividido em diversos subsistemas ao longo das regiões do país, considerando também suas províncias. Observando esta característica, foi atribuída a tarefa de concentração dos dados fasoriais

aos centros de controle desses subsistemas.

Nos centros de controle, servidores realizam a aquisição de dados que chegam através de rede privada, a *State Power Data Network* (SPDnet). Neste local, é realizado um processo de re-sincronização dos dados e atualiza-se a base de dados de tempo real. Os dados mais antigos são redirecionados para um novo servidor, dedicado a armazenar grandes volumes de dados históricos.

Os PDCs compartilham dados com os sistemas SCADA/EMS dos centros de controle, que utilizam os sincrofasores em suas aplicações, tais como: Controle Automático de Tensão; Sistema de Controle de Estabilidade Transitória; e Sistema Distribuído de Gerenciamento de Informações(XIE, X., *et al.*, 2006).

Atualmente alguns PDCs do sistema chinês recebem dados de PMUs com taxas de atualização de 100 sincrofasores por segundo. Isto caracteriza o cálculo de um fasor a cada metade de ciclo de 50 Hz. O maior desafio imposto ao projeto é estabelecer como desenvolver um PDC que esteja conectado com 100 PMUs, fornecendo medidas de 1000 canais fasoriais, que matenha o desempenho necessário para as aplicações de controle desejadas no futuro (YANG, Q., *et al.*, 2007).

### ***PDC MedFasee***

O início do desenvolvimento do PDC MedFasee deu-se a partir de 2003, no âmbito do Projeto MedFasee. Este equipamento faz parte do protótipo de SPMS projetado, tendo por objetivo receber, organizar, armazenar e disponibilizar os dados dos sincrofasores coletados pelas PMUs do projeto (DECKER, I. C., *et al.*, 2006).

A solução de concentração de dados contempla um único PDC, adotando-se, conseqüentemente, uma arquitetura centralizada. Este equipamento encarrega-se de receber todos os fasores enviados pelas PMUs do projeto. Em suas versões iniciais, o formato dos sincrofasores suportado era IEEE 1344. Entretanto, após a publicação da norma IEEE C37.118, todos os equipamentos do SPMS MedFasee foram migrados para o novo formato, com objetivo de operar em compatibilidade com esta norma. A Figura 3.2 ilustra a arquitetura do protótipo de SPMS desenvolvido pelo projeto MedFasee.

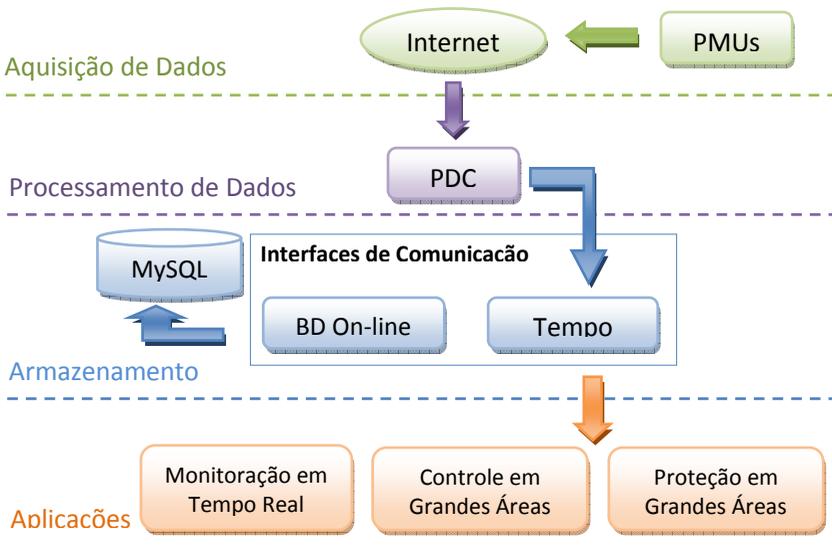


Figura 3.2 – Arquitetura do protótipo de SPMS do Projeto MedFasee - adaptado de (DECKER, I. C., et al., 2006).

Os dados são enviados por protocolo UDP/IP, dos pontos em que as PMUs estão conectadas, até a rede em que o PDC se encontra. Uma vez que os dados são recebidos pela interface de rede, extraem-se os valores de etiqueta de tempo, módulo e ângulo dos sincrofasores, para armazenamento na base histórica.

O banco de dados utilizado é o MySQL, que fornece uma ampla interface de funções para escrita e busca em tabelas de dados, sendo encarregado de correlacionar as etiquetas de tempo quando consultas são realizadas por aplicações. Seu tamanho é fixo, armazenando até sete dias de histórico, sobrescrevendo os dados após este período.

Para processos em tempo real existe a necessidade de correlacionar as etiquetas de tempo dos sincrofasores que chegam ao PDC. Isto é necessário pelo fato da inserção de atrasos no envio de dados entre os equipamentos do SPMS, característica inerente aos canais de comunicação. Sendo assim, um *buffer* em memória é destinado para o processamento das mensagens enviadas pelas PMUs. É definido um intervalo de tempo em que estes dados são alocados em memória, sendo processados com base em suas etiquetas de tempo e, por fim, disponibilizados para aplicações em tempo real.

O PDC MedFasee foi desenvolvido em linguagem de programa-

ção C++ com compilador GCC , utilizando o paradigma da Modelagem Orientada a Objetos (MOO). O sistema operacional utilizado é o GNU/Linux para garantir a operação do processo computacional do PDC em tempo real, utilizando-se o *Real Time Application Interface* (RTAI). Esta interface fornece uma biblioteca de funções para desenvolvimento de *software*, que permite, entre outras funcionalidades, conceder prioridade de tempo real para processos computacionais. Maiores detalhes são mostrados em (MANTEGAZZA, P., 1999).

### 3.1.2 Soluções Comerciais

#### ***PDC BPA e Enhanced Phasor Data Concentrator (ePDC)***

A BPA foi pioneira na implementação da tecnologia de SPMS em seu sistema elétrico. Para realizar a concentração de dados fasoriais foi desenvolvido um PDC capaz de processar os dados enviados das PMUs e enviá-los para aplicações que necessitam fazer uso dos mesmos.

Em 1997 a SCE adquiriu da BPA um PDC com objetivo de realizar a concentração dos sincrofasores de seu SPMS. Este PDC consistia de duas placas de aquisição de dados com capacidade para processar até 16 PMUs cada uma (BAHRGAVA, B., 1999).

Com a evolução dos estudos e desenvolvimentos, o PDC BPA se tornou um projeto comercial. Batizado de *Enhanced Phasor Data Concentrator – ePDC*, sua utilização é verificada na maioria das companhias que fazem parte do WECC e possuem PMUs em seus sistemas de transmissão.

Suas funcionalidades permitem a entrada de frames de dados de PMUs em formatos C37.118, *Macrodyne* e *PDCstream*. A configuração do sistema é realizada por meio de arquivo texto no formato XML, editado por aplicação dedicada para este fim. O *software* é independente de plataforma, podendo ser executado tanto em sistemas operacionais Windows ou GNU/Linux. Possui capacidade para operar com até 100 PMUs conectadas ao equipamento, com taxas de envio de fasores variando entre 6 e 60 frames por segundo (MARTIN, K., 2010).

Para o armazenamento dos sincrofasores utiliza-se um módulo de acompanhamento ao *ePDC*, denominado de *Phasor Archiver*. O *ePDC* envia os dados à este módulo, que possui uma base de dados compatível com MySQL, SQL e Oracle. O tamanho da base de dados é definido pelo usuário, dependendo da disponibilidade de *hardware*.

#### ***PDC Psymetrix***

A *Psymetrix* possui um pacote de serviços para aquisição e análise de dados coletados em SPMS. Batizado de *PhasorPoint*, os serviços podem ser separados em duas categorias: Concentradores de Dados (*PhasorPoint PDC*) e Aplicativos (*PhasorPoint Applications*) (PSY-METRIX, 2010).

O Concentrador de Dados pode realizar o papel de PDC de Subestação, em uma arquitetura descentralizada, bem como de PDC Central, processando e armazenando grandes volumes de dados. É capaz de receber múltiplos fluxos de dados em formato C37.118, podendo realizar conversão entre diferentes taxas de aquisição, sincronização de frames, múltiplos fluxos de saída de dados, armazenamento cíclico de dados.

Referente aos aplicativos são disponibilizadas ferramentas de monitoração para as seguintes grandezas e aspectos de operação do sistema: ângulo, magnitude de tensão, frequência, potências ativa e reativa, dinâmica do sistema, estabilidade transitória, recuperação/ressincronização de ilhas elétricas. O sistema é programado de forma modular, sendo possível a adição de novos aplicativos sem interferir naqueles previamente disponibilizados.

### ***PDC Schweitzer Engineering Laboratories – SEL PDC***

Como solução em sistemas de aquisição de dados fasoriais, a SEL propõe uma gama de produtos com objetivo de coletar dados fornecidos por PMUs. Entre eles, destaca-se o último desenvolvimento batizado *Station PDC* (SEL, 2010). Este PDC é compatível com arquiteturas descentralizadas. Possui capacidade para processar dados de aproximadamente 40 PMUs, mantendo compatibilidade com formatos de mensagens IEEE 1344, C37.118 e *SEL Fast Message*. Para o armazenamento histórico de dados é utilizado uma base de dados OBDC (*Open Data Base Connectivity*).

Outras soluções são apresentadas, tanto no que tange à coleta de dados das PMUs, quanto da visualização de sincrofasores em tempo real. O SEL-5073 *SynchroWave* é um *software* que adiciona funcionalidades de PDC a computadores comuns. Voltado à visualização de fasores e gerenciamento de dados históricos de eventos, utiliza um sistema de alarmes para a detecção de contingências no sistema. O aplicativo é desenvolvido para plataforma Windows, compatível com protocolos de comunicação *SEL Fast Message* e C37.118

### ***ABB PSGuard***

O *PSGuard* é o equipamento comercial desenvolvido pela ABB

com objetivo de realizar a aquisição de dados de PMUs nos formatos IEEE 1344 e C37.118. Um pacote de serviços é disponibilizado para a utilização dos sincrofasores contemplando equipamentos e aplicativos. Estes pacotes são divididos em três partes: *PSGuard 830*, *PSGuard 850* e *PSGuard 870*. O primeiro pacote fornece soluções para conectividade com PMUs, monitoração básica e armazenamento de dados com exportação manual. O segundo engloba as funcionalidades do primeiro, porém adicionando as seguintes ferramentas: monitoramento avançado, detecção e armazenamento automático de eventos, monitoração angular, monitoração térmica de linhas de transmissão, monitoração de estabilidade de tensão, detecção de oscilações eletromecânicas, monitoração da estabilidade de frequência e estimador de estados híbrido. O terceiro pacote de serviços complementa as funções do segundo, adicionando as seguintes ferramentas: recomendações de ação de controle, esquemas de ações restaurativas e controle avançado.

Referente ao equipamento, o *PSGuard* contém um disco rígido interno de 120 GB para armazenamento de dados locais. A base de dados é de formato específico do fabricante. Sua interface de comunicação suporta conectividade com até 16 PMUs, por meio de protocolo de comunicação TCP/IP, ou até 6 PMUs utilizando protocolo RS232 (protocolo de comunicação serial).

### **3.2 Principais requisitos para o desenvolvimento de um Sistema de Concentração de Dados Fasoriais**

Para o desenvolvimento de uma plataforma de aquisição de dados de medição fasorial sincronizada, existem diferentes aspectos a serem considerados. O equipamento encarregado desta tarefa deve ter as seguintes funcionalidades básicas: aquisição de medidas com sincronização temporal; capacidade de correlacionar etiquetas de tempo; armazenamento eficiente em base de dados; disponibilizar dados para aplicações em tempo real e para estudos *off-line* e operação contínua com prioridade de tempo real.

Como requisitos fundamentais de desempenho são necessários: alta performance computacional; sistema eficiente para armazenamento de dados; confiabilidade e disponibilidade; capacidade de comunicação; atendimento a variados tipos de aplicações; facilidade de integração; modularidade e expansibilidade.

Considerando o desempenho computacional, os dados recém chegados ao dispositivo de rede do equipamento devem ser disponibilizados às aplicações em tempo real e inseridos no banco de dados com a menor latência possível, caracterizando a necessidade de equipamento

dedicado e *software* desenvolvido com técnicas de programação que atendam estas demandas.

O sistema de armazenamento de dados históricos pode ser implementado de diversas maneiras, desde o desenvolvimento de formato próprio, como descrito em (ZUO, J., *et al.*, 2008), até a utilização de sistemas de gerenciamento de bancos de dados predefinidos (MySQL, Oracle, PostgreSQL, etc.). Independente da solução adotada, esta deverá ser estável e eficiente para que dados históricos possam ser rapidamente consultados.

A confiabilidade faz-se necessária uma vez que, como qualquer outro equipamento, eventos exógenos ao sistema podem ocorrer durante o processo de aquisição de dados, tais como: quedas de energia, falhas nos links de comunicação e mudanças de topologia do SPMS, onde existe a possibilidade da interrupção momentânea dos dados enviados pelas PMUs ao PDC, durante o regime de operação do sistema. A disponibilidade dos dados não deve ser prejudicada em função destes eventos, cabendo tanto ao *software* quanto ao *hardware* voltar ao estado de funcionamento prévio, sem para isso haver necessidade de intervenção do usuário.

O PDC deve ser capaz de se comunicar com outras interfaces por meio de protocolos de comunicação padronizados, permitindo troca de informações entre sistemas de diferentes plataformas operacionais e que podem ter sido desenvolvidos em arquiteturas de *hardware* diferentes. Desta forma um mesmo PDC pode adquirir dados de PMUs de diversos modelos comerciais, processar e disponibilizar os dados para diferentes sistemas que possam fazer uso das informações sejam eles sistemas SCADA/EMS, aplicativos em diferentes sistemas operacionais (GNU/Linux, Windows, MacOS X, FreeBSD, etc.) ou sistemas embarcados de solução específica.

O equipamento do PDC deve ser capaz de comportar mudanças na estrutura do SPMS ao longo do tempo. Novas PMUs são gradativamente adicionadas e as limitações para a aquisição destes novos dados devem ser somente ditadas pelo limite de *hardware* do equipamento. Portanto, o desenvolvimento do sistema de aquisição de dados deve comportar uma estrutura genérica no que se refere a número de PMUs e PDCs presente no SPMS, fazendo necessário maior utilização dos recursos computacionais do equipamento. Desta forma assegura-se a expansibilidade do SPMS do ponto de vista do processamento, armazenamento e aquisição de dados.

Por se tratar de um equipamento dedicado exclusivamente à concentração de dados, é comum que sua localização seja isolada, evitando riscos de acidentes e acessos indevidos ao sistema. Desta forma, toda comunicação e operação do concentrador devem ser planejadas de maneira a maximizar opções de controle e acesso externo, via ferramentas de acesso remoto. Estas opções garantem, não somente maior segurança ao equipamento, como facilitam o acesso remoto pelo usuário, caso sejam necessárias intervenções de manutenção, sejam elas corretivas ou preventivas, ou apenas para monitoração do desempenho operacional do PDC.

### **3.3 Aspectos de arquitetura de concentração de dados fasoriais**

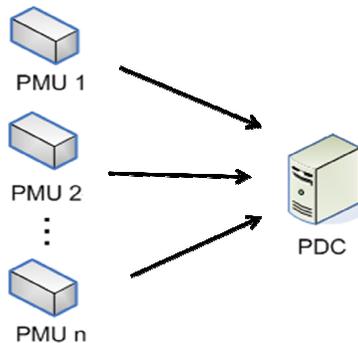
O principal aspecto de arquitetura para concentração de dados fasoriais envolve a disposição dos equipamentos que tem função de PDC ao longo do sistema. Considerando-se que cada SPMS tem características distintas (infra-estrutura disponível, requisitos de rede, taxa de amostragem, número de PMUs), e que aplicações com objetivos distintos farão uso dos mesmos sincrofasores, é natural que existam diferentes soluções viáveis para arquitetura. Neste contexto, observam-se dois tipos de arquitetura para concentração de dados fasoriais: a arquitetura centralizada e a arquitetura descentralizada. Estas soluções são apresentadas na seqüência.

#### **3.3.1 Arquitetura Centralizada**

Para o caso desta arquitetura, considera-se a presença de um único equipamento com funcionalidade de PDC. Todo o fluxo de dados enviado pelas PMUs do SPMS é redirecionado para este equipamento. A Figura 3.3 exemplifica a Arquitetura Centralizada.

Esta é a solução primogênita na tecnologia de medição fasorial sincronizada. Suas principais características são o gerenciamento da aquisição de dados de todo o sistema em apenas um equipamento e a simplicidade de instalação, uma vez que utiliza o número mínimo de equipamentos com função de PDC.

A latência envolvida no processo, a partir do cálculo dos sincrofasores pelas PMUs até a disponibilização para aplicações, depende apenas dos atrasos inseridos nos canais de comunicação e do tempo de processamento do PDC.



*Figura 3.3 – Arquitetura Centralizada de Concentração de Dados Fasoriais.*

Por outro lado, a fim de garantir redundância e confiabilidade no armazenamento de dados, se faz necessário a presença de um equipamento adicional que possa assumir a tarefa de aquisição de dados em caso de pane nos equipamentos do sistema de concentração de dados. Caso não existam links adicionais de comunicação entre PMUs e PDC, dados podem ser perdidos em situações onde a rede local do PDC, por algum motivo, pare de funcionar.

Outra característica a ser observada é a expansão do SPMS, onde a adição de novas PMUs no sistema faz com que a carga de processamento do PDC aumente gradativamente juntamente com a largura de banda necessária na rede local. A limitação de *hardware* do PDC é um fator que deve ser constantemente observado, sobretudo em sistemas de grande porte.

### **3.3.2 Arquitetura Descentralizada**

Nesta arquitetura considera-se a presença de múltiplos equipamentos com funcionalidade de PDC, com possibilidade de trocar informações entre si. Um grupo de PMUs do SPMS é direcionado a enviar dados a um PDC, que processa seus dados, podendo armazená-los localmente. Este PDC pode ainda enviar dados para outros PDCs do sistema, dependendo do critério de hierarquia adotado. A Figura 3.4 ilustra um exemplo de Arquitetura Descentralizada.

Esta arquitetura agrega maior capacidade de processamento por equipamento com função de PDC, uma vez que as PMUs do SPMS são distribuídas pelos diversos PDCs, permitindo também realizar aquisição distribuída de sincrofasores. A carga de gerenciamento de comunicação entre os equipamentos é dividida, de forma que cada PDC é responsável por administrar as conexões somente dos equipamentos que é responsá-

vel. No caso do exemplo apresentado, os PDCs A e B gerenciam apenas a comunicação com suas respectivas PMUs. No caso do PDC C, os equipamentos cuja comunicação ele deve gerenciar são os PDCs A e B, somente.

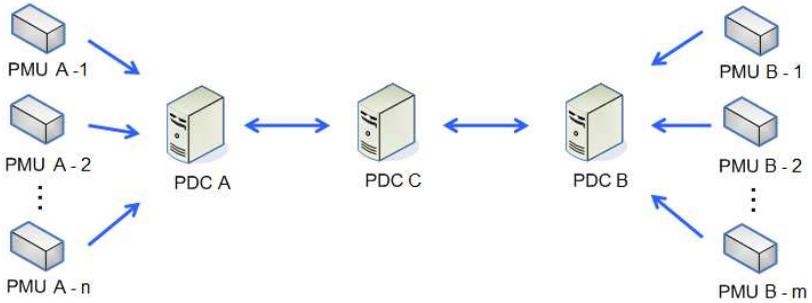


Figura 3.4 – Arquitetura Descentralizada de Concentração de Dados Fasoriais.

Considerando-se a presença de um maior número de PDCs no sistema, existe a possibilidade do armazenamento local de sincrofases. Esta característica agrega redundância no processo de concentração de dados, conseqüentemente aumentando sua confiabilidade.

Cabe ressaltar a característica modular presente nesta arquitetura. A possibilidade de adição de novos equipamentos com funcionalidade de PDC torna o sistema altamente expansível. Novas PMUs podem ser adicionados sempre que necessárias, podendo ser adicionado um PDC local com objetivo de distribuir o processamento dos novos sincrofases.

Como conseqüência da adição de um maior número de equipamentos na cadeia de aquisição de dados, a latência para disponibilização é afetada. Este valor aumenta conforme o número de PDCs envolvidos entre o cálculo do sincrofases pela PMU, e a disponibilização dos dados para as aplicações. O tempo de processamento individual de cada PDC é adicionado à latência de comunicação total, sendo necessária a sua quantificação para que as aplicações executem de maneira adequada. Esta característica é inerente à arquitetura descentralizada, podendo ser mitigada por meio do uso de equipamentos cujo tempo de processamento seja tanto menor quanto necessário, observando-se a disponibilidade de tecnologias de *hardware* disponíveis no mercado.

### 3.4 Proposta de um Sistema de Concentração de Dados Fasoriais - PDCS

A necessidade de uma arquitetura descentralizada de aquisição de dados surge a partir do momento de expansão que vive a tecnologia de medição fasorial sincronizada. Maiores níveis de armazenamento, processamento e disponibilização de dados são cada vez mais necessários. Dividir a tarefa de receber e armazenar dados de diferentes PMUs, não somente auxilia no alívio de processamento, como agrega redundância e confiabilidade no processo de aquisição de sincrofases.

A proposta deste trabalho tem por objetivo desenvolver um Sistema de Concentração de Dados Fasoriais (PDCS), considerando-se a Arquitetura Descentralizada, com arranjo hierárquico entre PDCs em dois níveis. Para isto foi necessária a interpretação e adaptação do PDC MedFasee (código-fonte e desenvolvimentos), para que se torne compatível com arquitetura planejada.

Os PDCs que fazem parte do PDCS são divididos de acordo com o nível hierárquico em que se encontram. Os PDCs alocados no primeiro nível são denominados *PDC Front End*, ao passo que o PDC presente no segundo nível é denominado de PDC Mestre. A Figura 3.5 ilustra um cenário genérico de aplicação do PDCS proposto.

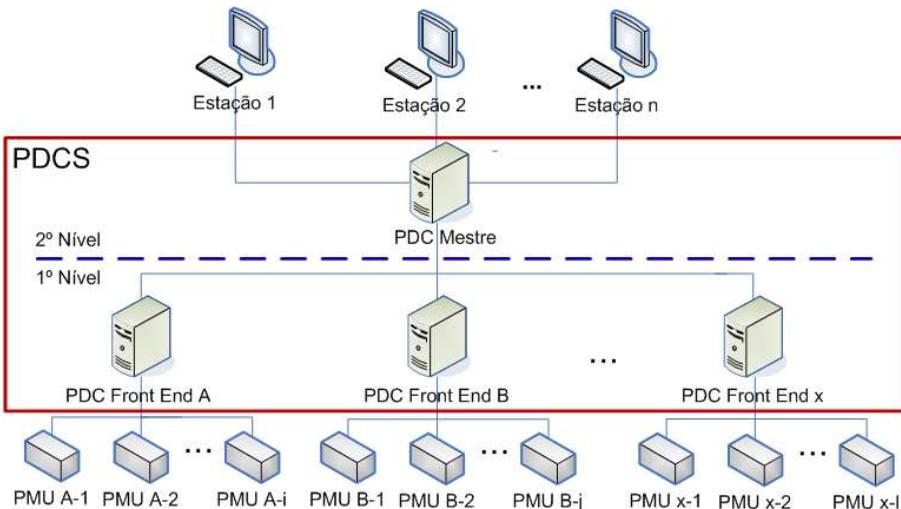


Figura 3.5 – Cenário de aplicação genérica do PDCS proposto.

O *PDC Front End* realiza a comunicação direta com as PMUs do SPMS. Este equipamento é responsável pela concentração de um determinado número de PMUs do sistema, bem como pelo gerenciamento da comunicação com suas respectivas PMUs, armazenando os sincrofasores em seu banco de dados local. É sua função também realizar a resincronização de dados, de forma que os sincrofasores sejam encaminhados para o segundo nível da hierarquia, ou seja, para o PDC Mestre.

O PDC Mestre será posicionado no segundo e maior nível da hierarquia. Suas responsabilidades são: realizar e gerenciar a comunicação com os *PDCs Front End*; armazenar os dados de todo o SPMS; disponibilizar os sincrofasores para aplicações em tempo real e para estudos *off-line*.

Nota-se que esta representação da arquitetura é genérica, exaltando a flexibilidade de conexão de diversos *PDCs Front End* ao PDC Mestre. Desta maneira é possível a divisão do SPMS por regiões de coletas de dados, agrupando PMUs próximas geograficamente em um único PDC específico para sua área. Outra consequência deste arranjo é o aumento da confiabilidade e redundância do sistema, dado que é possível o armazenamento local dos sincrofasores em suas respectivas bases históricas.

Considerando-se os fluxos de dados que são trocados entre os elementos desta arquitetura, percebe-se a importância da padronização dos formatos das mensagens que estabelecem a comunicação entre os equipamentos do SPMS, sobretudo no caso da troca de dados entre PDCs. Outro aspecto de relevância é a lógica operacional do PDCS. Utilizando-se de uma visão mais abrangente, é necessário observar a operação do PDCS de duas perspectivas: do sistema como um todo; do equipamento com funcionalidade de PDC. As seções a seguir apresentam estes aspectos de maneira aprofundada.

### **3.4.1 Comunicação entre PDCs – Norma IEEE C37.118**

Esta seção tem por objetivo descrever a estrutura dos formatos de mensagens previstos na norma IEEE C37.118 (IEEE, 2005). Estes formatos são denominados *frames*, e definem a interface de comunicação entre equipamentos que possuem aplicação a sistemas de medição fasorial sincronizada.

No caso de PMUs, a utilização das mensagens nos formatos da norma IEEE C37.118 é fundamental para subsidiar a interoperabilidade de diferentes fabricantes de equipamentos, deixando suas percepções sobre a tecnologia restritas a um conjunto interno de funções para aquisição de sinais analógicos e cálculo dos sincrofasores, proporcionando

um ambiente plural de soluções em um mesmo SPMS. Para o caso do PDCS, os *frames* da norma IEEE C37.118 possibilitam a comunicação entre PDCs de maneira organizada, dado que esta comunicação é um aspecto central para a viabilização do sistema proposto.

A estrutura geral dos *frames* definidos na norma IEEE C37.118 é ilustrada na Figura 3.6.

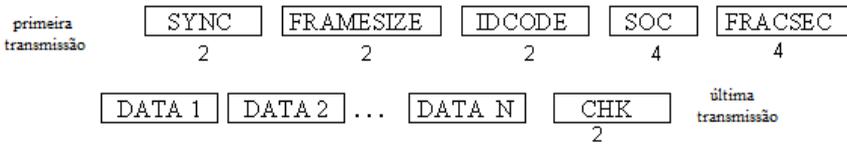


Figura 3.6 – Estrutura geral dos frames da norma IEEE C37.118. Adaptado de (IEEE, 2005).

O campo *SYNC* identifica o *frame* como sendo referente à norma. O campo *FRAMESIZE* informa o tamanho em bytes total do *frame*. Campo *SOC*<sup>2</sup> é preenchido com o valor da etiqueta de tempo do *frame*. O campo *FRACSEC* contém o valor da posição do *frame* dentro do segundo do SOC (em sistemas com taxa de 60 sincrofases por segundo, o campo *FRACSEC* pode variar entre 0 e 59). O campo *CHK* indica a integridade dos bytes do *frame*. Os campos *DATA* dependem do tipo de *frame*, que podem ser: cabeçalho; configuração; dados; e comando. Estes *frames* serão descritos a seguir.

### 3.4.1.1 *Frame de Cabeçalho*

Este *frame* possui dados em formatos ASCII (*American Standard Code for Information Interchange*) contendo informações sobre o equipamento tais como: número de série, versão de firmware, nome do dispositivo, filtros, algoritmos, entre outros. A Tabela 3.1 ilustra a ordem que os bytes contidos na mensagem devem ser encapsulados.

Este *frame* tem função informativa, sendo que os dados contidos nele são de responsabilidade do fabricante do equipamento.

### 3.4.1.2 *Frame de Configuração*

O *frame* de configuração tem por objetivo fornecer informações sobre como são enviados os sincrofases de um equipamento a outro. O dispositivo que recebe o *frame* de configuração saberá a ordem em que deve processar os sincrofases contidos no *frame* de dados, descrito na

<sup>2</sup> SOC – Sigla de *Second of Century*. O SOC é definido como o número de segundos corridos entre a data atual e o dia 1º de Janeiro de 1970.

seção a seguir. A configuração possui informações sobre o formato dos fasores (polar ou retangular), precisão das variáveis (valores inteiros ou ponto flutuante), nome do equipamento, número de sincrofases, informações sobre os canais que fornecem sincrofases, dados analógicos e dados digitais, entre outros. A estrutura do *frame* de configuração é ilustrada na

Tabela 3.1 – Organização do Frame de Cabeçalho IEEE C37.118.

Número	Campo	Tamanho [Bytes]	Descrição
1	SYNC	2	Bytes de sincronização
2	FRAMESIZE	2	Tamanho do frame
3	IDCODE	2	Código do Equipamento no SPMS
4	SOC	4	Etiqueta de tempo
5	FRACSEC	4	Fração de segundo
6	DATA 1	1	Caractere ASCII, 1º byte
k+6	DATA K	1	Caractere ASCII, k-ésimo byte, k>0
k+7	CHK	2	Verificação de integridade de dados

### 3.4.1.3 Frame de Dados

Este *frame* contém os dados fasoriais propriamente ditos. Conseqüentemente, dentro da operação normal de um SPMS, é o *frame* mais utilizado pelos equipamentos. Os dados estão ordenados de acordo com as informações fornecidas pelo *frame* de configuração. Desta forma, um *frame* de dados proveniente de uma PMU possui apenas dados de seus respectivos canais fasoriais, analógicos e digitais. Entretanto, se os dados forem oriundos de um PDC, este conterá dados de diversas PMUs em conjunto. Esta característica é possível através do campo NUM\_PMU que informa a quantidade de PMUs que estão presentes no mesmo *frame* de dados. A Tabela 3.3 ilustra a estrutura do *frame* de dados.

Destaca-se o campo SOC, que representa a etiqueta de tempo do frame. No caso da PMU, seu valor é trivial. Uma vez determinado o instante de tempo em que os sincrofases foram calculados, basta armazená-lo neste campo. Porém, no caso de dados transferidos entre PDCs, é necessário um processamento para a re-sincronização de dados recebidos, uma vez que o envio através da rede de comunicação é realizado de maneira assíncrona. O processo de re-sincronização será apresentado no Capítulo 4.

*Tabela 3.2 – Frame de Configuração IEEE C37.118.*

Número	Campo	Tamanho [Bytes]	Descrição
1	SYNC	2	Bytes de Sincronização
2	FRAMESIZE	2	Tamanho do frame
3	IDCODE	2	Código do Equipamento no SPMS
4	SOC	4	Etiqueta de tempo
5	FRACSEC	4	Fração de segundo
6	TIME_BASE	4	Resolução da fração de segundo
7	NUM_PMU	2	Número de PMUs no frame de dados
8	STN	16	Nome da estação – Caractere ASCII 16 Bytes
9	IDCODE	2	Código da PMU no SPMS
10	FORMAT	2	Formato dos canais no frame de dados
11	PHNMR	2	Número de canais fasoriais
12	ANNMR	2	Número de canais analógicos
13	DGNMR	2	Número de canais digitais
14	CHNAM	$16 \times (\text{PHNMR} + \text{ANNMR} + 16 \times \text{DGNMR})$	Nomes dos fasores e dos respectivos canais Analógicos e digitais
15	PHUNIT	$4 \times \text{PHNMR}$	Fator de conversão dos canais fasoriais
16	ANUNIT	$4 \times \text{ANNMR}$	Fator de conversão dos canais analógicos
17	DIGUNIT	$4 \times \text{DGNMR}$	Máscara para as informações dos canais digitais
18	FNOM	2	Frequência nominal
19	CFGCNT	2	Contador de mudança de configurações
	Repetir 8-19		Repetir os campos 8-18 igual ao nº de PMUs do campo 7
20	DATA_RATE	2	Taxa de transmissão de dados
21	CHK	2	Verificação de integri-

dade de dados

Tabela 3.3 – Frame de Dados IEEE C37.118.

Número	Campo	Tamanho [Bytes]	Descrição
1	SYNC	2	Bytes de Sincronização
2	FRAMESIZE	2	Tamanho do frame
3	IDCODE	2	Código do Equipamento no SPMS
4	SOC	4	Etiqueta de tempo
5	FRACSEC	4	Fração de segundo
6	STAT	2	Flags de estado da PMU/PDC
7	PHASORS	4xPHNMR ou 8xPNHMR	Fasores – inteiro u ponto flutuante
8	FREQ	16	Frequência – inteiro/ponto flutuante
9	DFREQ	2 ou 4	Taxa de variação da frequência – inteiro ou ponto flutuante
10	ANALOG	2xANNMR ou 4xANNMR	Dados analógicos – inteiro ou ponto flutuante
11	DIGITAL	2xDGNMR	Dados digitais
	Repetir 6-11		Campos 6-11 são repetidos de acordo com o campo NUM_PMU do frame de configuração
21	CHK	2	Verificação de integridade de dados

### 3.4.1.4 Frame de Comando

Este *frame* tem por função permitir que os dispositivos dentro de um SPMS possam responder a comandos de origem externa. Estes comandos podem ter por origem tanto nos equipamentos do SPMS quanto nos centros de controle. São definidos seis tipos diferentes de comandos, mostrados a seguir:

- Comando 1: Desligar a transmissão de *frames* de dados;
- Comando 2: Ligar a transmissão de *frames* de dados;
- Comando 3: Envio de *frame* de cabeçalho;
- Comando 4: Envio de *frame* de configuração 1;
- Comando 5: Envio de *frame* de configuração 2;
- Comando 6: Envio de *frame* estendido.

A Tabela 3.4 ilustra a estrutura do *frame* de comando.

Tabela 3.4 – *Frame de Comando IEEE C37.118.*

Número	Campo	Tamanho [Bytes]	Descrição
1	SYNC	2	Bytes de sincronização
2	FRAMESIZE	2	Tamanho do frame
3	IDCODE	2	Código do Equipamento no SPMS
4	SOC	4	Etiqueta de tempo
5	FRACSEC	4	Fração de segundo
6	CMD	2	Comando enviado para PMU/PDC
7	EXTFRAME	0-65518	Dados do frame estendido, palavras de 16 bits máximo de 65518 bytes definidos pelo usuário
8	CHK	2	Verificação de integridade de dados

O *frame* estendido provê a possibilidade de o usuário definir qualquer tipo de dado que se deseje trocar entre os equipamentos, desde mensagens de log do sistema até dados diversos sobre monitoração de variáveis de desempenho do sistema. Seu uso e conteúdo é completamente arbitrário, podendo inclusive servir de meio para trocas de dados entre equipamentos do SPMS.

### 3.4.2 Lógica de Operação do PDCS

Considerando-se o ambiente de funcionamento do PDCS, os PDCs devem seguir uma lógica de operação preestabelecida. Esta lógica é dividida em dois modos: (i) configuração da operação dos PDCS; (ii) operação do PDCS. As seções a seguir apresentam em detalhes as referidas lógicas.

#### 3.4.2.1 Configuração da Operação do PDCS

Esta etapa visa prover os PDCs do sistema com os parâmetros de configuração dos equipamentos com o qual eles se comunicam. Estes parâmetros que necessitam ser informados ao PDC para que se possa realizar a interpretação do *frame* de dados de maneira correta. No caso de *PDC Front End*, são solicitados os parâmetros das PMUs pelas quais ele é responsável. No caso de PDC Mestre, são solicitados os parâmetros dos *PDCs Front End* do PDCS. A Figura 3.7 ilustra a lógica de configuração da operação do PDCS.

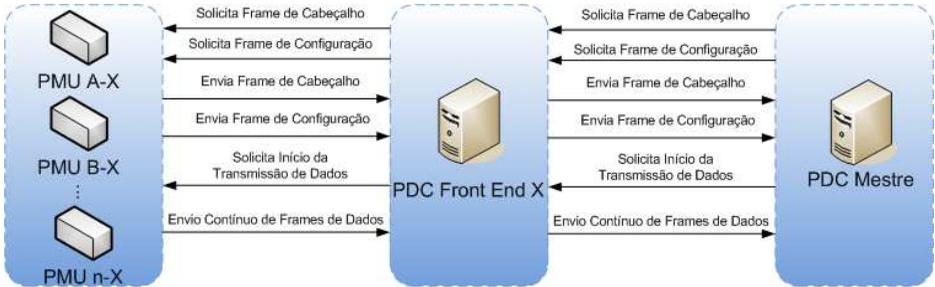


Figura 3.7 – Lógica de configuração do PDCS.

A solicitação dos parâmetros é realizada por meio de *frames* de comando. Em um primeiro momento, solicitam-se, para o equipamento de destino, os *frames* de cabeçalho e configuração, passando-se a aguardar a chegada dos dados requeridos. Uma vez recebidos os *frames*, um novo *frame* de comando é enviado, solicitando o início da transmissão de dados. A partir deste momento, inicia-se a concentração de dados fasoriais, com transmissão contínua de *frames* de dados entre os equipamentos do PDCS. Este procedimento é idêntico tanto para o *PDC Front End* quanto para o PDC Mestre.

### 3.4.2.2 Operação do PDCS

A operação contínua do PDCS ocorre quando os PDCs do sistema contêm os parâmetros de configuração dos equipamentos pelos quais são responsáveis. A Figura 3.8 ilustra o fluxo dos dados ao longo do PDCS, desde a aquisição dos dados das PMUs, até a disponibilização dos sincrofases pelo PDC Mestre.

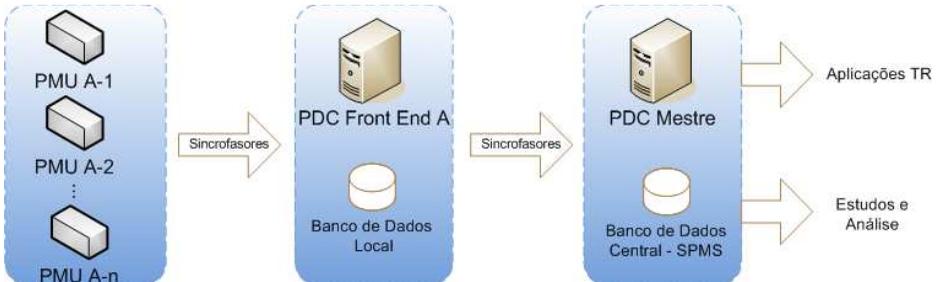


Figura 3.8 – Fluxo de sincrofases na operação do PDCS.

Na operação do PDCS, primeiramente os sincrofases são enviados das PMUs ao *PDC Front End*. Esta operação é realizada no primei-

ro nível da hierarquia do sistema. Os dados são armazenados em base de dados local, porém existe a necessidade de redirecionar os sincrofasores ao PDC Mestre. Isto é tarefa da interface de tempo real do *PDC Front End*, que terá a atribuição de regularizar os atrasos inseridos pelos canais de comunicação, correlacionando os sincrofasores às mesmas etiquetas de tempo em que foram calculados. Quando o processo de re-sincronização é finalizado, um novo pacote de dados é disponibilizado para envio PDC Mestre.

O PDC Mestre, ao receber os dados enviados pelos *PDCs Front End* do sistema, ocupa-se de armazenar os valores no seu respectivo banco de dados e, aplicando o mesmo processo de re-sincronização de dados dos *PDCs Front End*, disponibiliza um fluxo de dados para aplicações em tempo real, contendo dados de todas as PMUs do SPMS.

### **3.4.3 Lógica de Operação do PDC**

Esta seção descreve a lógica de operação do equipamento PDC. Para isso, foram separadas etapas e processos de forma a descrever as ações realizadas internamente pelo equipamento, desde a configuração dos parâmetros de operação, até a disponibilização dos dados coletados.

Primeiramente, uma visão geral sobre a lógica operacional do PDC é apresentada. Em seguida, descreve-se a configuração dos parâmetros de operação fornecidos pelo usuário, a fim de esclarecer como as particularidades de cada SPMS são adaptadas ao PDC. Isto garante sua operação de forma adequada. Em seguida, são apresentadas, com o auxílio de fluxogramas, as Etapas de Inicialização e Principal de Operação do PDC.

#### **3.4.3.1 Visão geral da lógica operacional do PDC**

A lógica de operação do PDC pode ser separada de acordo com as etapas presentes na Figura 3.9.

A Etapa de Inicialização carrega os atributos do PDC com os parâmetros de configuração do SPMS, iniciando os protocolos necessários para a operação do PDC. Em seguida, dá-se início a Etapa de Operação, compreendendo o gerenciamento dos parâmetros de configuração dos equipamentos do SPMS, bem como o recebimento de dados. Iniciada esta etapa, o processo do PDC executa de maneira contínua, disponibilizando os dados em dois diferentes *buffers*, um em disco rígido (HD), e outro em memória. Os dados em HD são acessados pelo Processo de Armazenamento, processo computacional executado em paralelo ao processo do PDC, responsável pela inserção dos sincrofasores na base de dados do PDC. Uma rotina interna ao processo do PDC acessa o

*buffer* em memória, alinhando as etiquetas de tempo dos sincrofasores, disponibilizando um pacote de dados em tempo real. Este pacote é encaminhando ao PDC Mestre, caso o PDC seja *Front End*, ou para as aplicações em tempo real, caso o PDC seja Mestre.

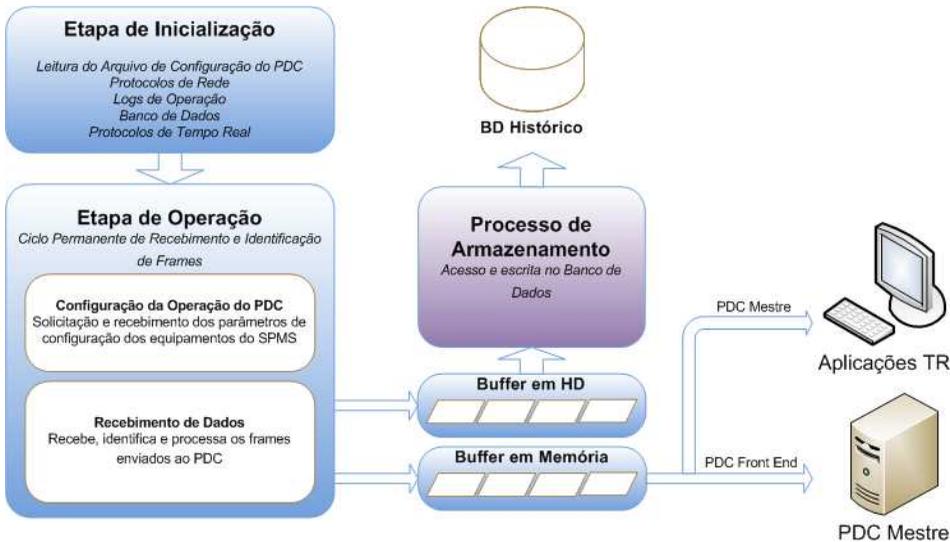


Figura 3.9 – Lógica de operação do PDC.

As seções a seguir apresentam em detalhes as etapas e processos que fazem parte da lógica de operação do PDC.

### 3.4.3.2 Etapa de Inicialização

Primeiramente, existe a necessidade de informar ao PDC os parâmetros de configuração para sua operação, bem como os parâmetros dos equipamentos que lhe enviarão dados. Isto equivale a informar, por exemplo, a quantidade de PMUs e PDCs que o equipamento deverá se comunicar, a seleção de *logs* para acompanhamento de sua operação, o tamanho da base de dados histórica, o tamanho do *buffer* em memória, se sua função na hierarquia é de Mestre ou *Front End*, dentre outros. Além disso, estes parâmetros precisam ser acessados de forma externa ao código-fonte do PDC. Para isso utiliza-se um arquivo de configuração.

Este arquivo engloba valores que, do ponto de vista do *software* em execução, são constantes. Entretanto, ao longo do tempo de operação do SPMS, pode ser necessária a alteração de algum destes parâmetros. A utilização do arquivo de configuração permite sua alteração, quando

necessária, sem haver necessidade de nova compilação do código-fonte do PDC.

Para o caso do PDC MedFasee, a configuração do sistema é feita por meio de dois arquivos texto em formato ASCII, possuindo características próprias sem padronização. O primeiro arquivo contém parâmetros exclusivamente para a operação do PDC, enquanto que, no segundo, encontram-se os dados pertencentes às PMUs do SPMS. Dois aspectos prejudicam a utilização desta solução: a possibilidade de erros em tempo de execução (supondo que algum dado incorreto seja inserido pelo usuário, não existe mecanismo para interpretação deste erro, que é percebido apenas quando da execução do *software*); a dificuldade de acesso individual de algum parâmetro, caso o programador necessite acesso a um valor particular, a leitura do arquivo deverá ser feita de maneira seqüencial até o encontro do valor desejado. Este procedimento utiliza linhas de código e processamento de dados desnecessários.

Tendo isto em vista, percebe-se a necessidade de se utilizar uma solução de configuração que possua uma sintaxe padrão, de fácil entendimento ao usuário, com capacidade de informar erros de digitação, validar os tipos de dados fornecidos nos parâmetros de configuração e que unifique as informações operacionais do PDC e a estrutura que compõe o SPMS como um todo.

Para satisfazer as necessidades citadas acima, optou-se pela montagem de um arquivo de configuração em formato XML (*Extended Markup Language*). Este arquivo, assim como o anterior, é escrito por meio de caracteres em formato ASCII, e pode ser modificado utilizando editores de texto padrão.

O principal objetivo da linguagem é o transporte de dados e o que eles representam, utilizando para isso marcações de texto, denominados etiquetas (*tags*). Estas etiquetas envolvem os dados presentes no arquivo, identificando-os e adicionando propriedades caso haja necessidade. Um documento XML possui regras padronizadas de acordo com o *World Wide Web Consortium* (W3C), consórcio internacional entre empresas, desenvolvedores próprios e público geral, destinado à elaboração de regras e normas para a Internet.

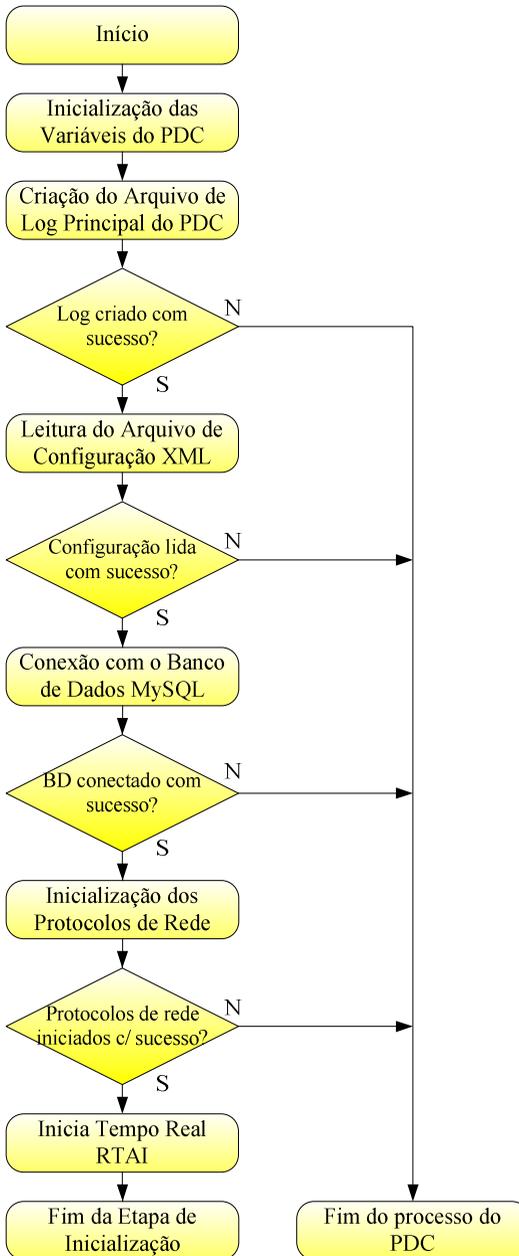
Qualquer aplicação compatível com as regras de formação de um documento XML deve, antes de processar seus dados, analisar o arquivo e validá-lo, indicando os erros que possam estar presentes. Esta característica é importante para o processo de configuração do PDC, uma vez que fontes de erros podem ser devidamente mitigadas. O acesso aos dados do arquivo pode ser realizado de acordo com as necessidades do

desenvolvedor, reduzindo o código-fonte, evitando processamento em excesso de dados.

Observando os aspectos desta linguagem, a vasta disponibilização de bibliotecas de desenvolvimento em diversas linguagens de programação, aplicativos de edição, desenvolvimento e validação de documentos XML e documentação para estudo, percebe-se que a sua aplicação ao processo de desenvolvimento do PDC cumpre os requisitos necessários. Maiores detalhes sobre a linguagem XML, suas definições, conceitos e regras de formação serão abordados em detalhes no Apêndice A.

Após a determinação dos parâmetros de operação do PDC e dos elementos que irão compor o SPMS por meio do arquivo de configuração XML, uma série de procedimentos devem ser realizados de forma a possibilitar a troca de *frames* no formato IEEE C37.118 através dos links de comunicação.

Inicialmente, é necessário atribuir valores iniciais aos atributos do PDC. Este procedimento garante que nenhum campo contenha valores espúrios devido à sua declaração inicial em tempo de execução. Em seguida cria-se o log principal do PDC. Seu objetivo é realizar a interface com o usuário, permitindo acompanhar os principais eventos na execução do programa do PDC. Após este processo, é realizada a leitura do arquivo de configuração XML, que fornecerá os parâmetros necessários para operação do concentrador de dados. A conexão com o banco de dados MySQL é realizada, criando as tabelas que irão armazenar os dados históricos do SPMS. Os protocolos de rede são inicializados, determinando-se os parâmetros de rede para que o PDC possa se comunicar com os demais equipamentos do SPMS. Finalmente, é dado início à interface de tempo real, RTAI, informando ao sistema operacional que o processo do PDC deve ser tratado com prioridade máxima em relação aos outros processos executados simultaneamente. Após esta última etapa, o PDC está pronto para seguir à próxima Etapa. A Figura 3.10 ilustra o fluxograma da Etapa de Inicialização do PDC.



*Figura 3.10 – Fluxograma da Etapa de Inicialização do PDC.*

### 3.4.3.3 Etapa Principal de Operação do PDC

A Etapa Principal de Operação inicia após a Etapa de Inicialização. Nela, inicia-se um ciclo responsável por realizar a constante monitoração da chegada de *frames* na interface de rede do equipamento do PDC. Este ciclo é encarregado de receber, identificar e processar *frames* que chegam, tomando as decisões cabíveis em cada caso. A Figura 3.11 ilustra, de maneira geral, a etapa principal de operação do PDC.

Toda vez que uma varredura de dados for realizada, e os protocolos de rede acusarem o recebimento de um *frame*, o sistema deve identificar seu formato, verificando a compatibilidade com a norma IEEE C37.118. Em caso positivo, o *frame* é processado para a determinação do seu tipo (configuração, cabeçalho, comando ou dado), passando-se para rotinas específicas de tratamento das informações.

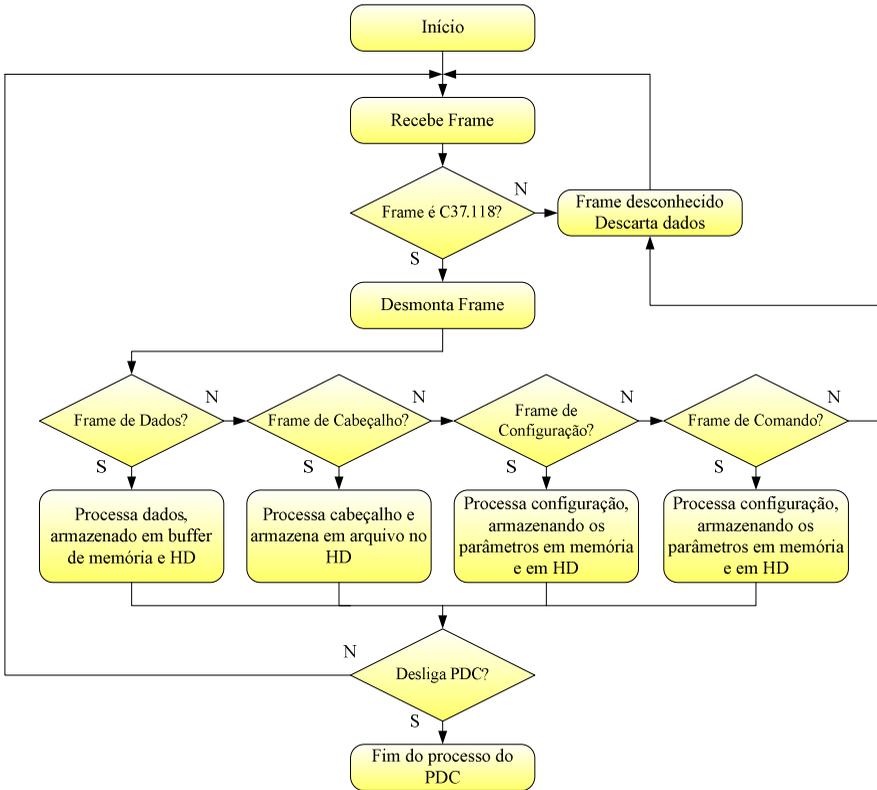


Figura 3.11 – Etapa Principal de Operação do PDC.

Caso não seja identificado o formato IEEE C37.118, o pacote de dados é considerado desconhecido e descartado pelo sistema. Ao final de cada ciclo, o sistema realiza uma verificação com objetivo de determinar se a finalização do processo do PDC foi solicitada. Se tal evento não foi apontado, um novo ciclo principal de varredura de dados é iniciado. Este procedimento ocorre continuamente, até uma solicitação explícita de parada ser realizada, por meio de *frame* de comando.

A Etapa de Operação, apresentada na Figura 3.11, mostra de maneira geral o tratamento aos *frames* que chegam a um PDC. Entretanto, diversos procedimentos ocorrem ao longo de sua operação, sendo que esta etapa pode ser dividida em duas partes. A primeira compreende a configuração da operação do PDC, a segunda é referente ao processo de recebimento dos *frames* de dados dos equipamentos do SPMS. A seguir estas duas partes serão apresentadas em detalhes.

### ***Etapa de Configuração da Operação do PDC***

A etapa de configuração da operação é iniciada dentro da etapa principal do PDC, fazendo uso do ciclo principal de operação. Seu princípio básico é informar o PDC com os dados de configuração dos equipamentos com o(s) qual(is) ele se comunica. Isto possibilita a troca de *frames* de maneira correta, assegurando o processamento dos sincrofasores que são enviados pelas(os) PMUs/PDCs. Isto é realizado por meio do envio e recebimento de *frames* de cabeçalho e configuração, que são enviados para todos os equipamentos que o PDC deve se comunicar. Uma vez recebidos os parâmetros de configuração, solicita-se o início do envio de *frames* de dados. A etapa é considerada completa quando houver o recebimento do primeiro *frame* de dados pelo PDC. A Figura 3.12 ilustra o fluxograma da etapa de configuração do PDC.

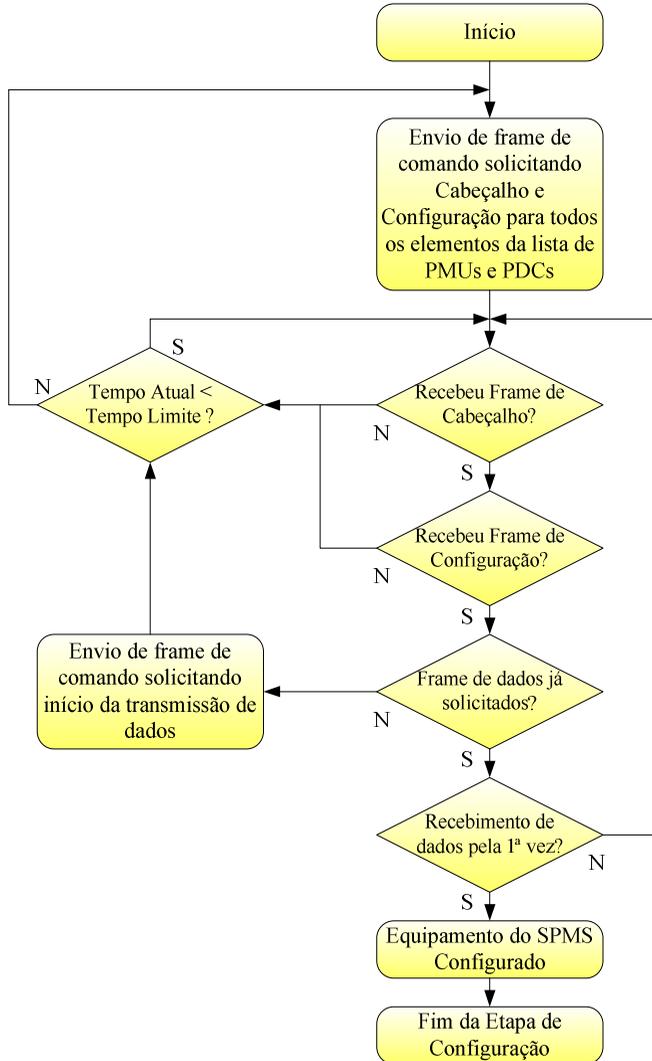


Figura 3.12 – Etapa de Configuração do PDC.

Onde:

*Tempo Atual*: Contador que acumula o tempo decorrido desde o início da configuração do equipamento.

*Tempo Limite*: Tempo máximo tolerado para a configuração do equipamento.

O PDC possui uma lista de PMUs e uma lista de PDCs armazenadas em memória. Tais listas são criadas a partir da leitura do arquivo de configuração XML do PDC. Elas carregam pares de informações contendo IDCODE (código de referência do equipamento no SPMS, parâmetro da norma IEEE C37.118) e endereço IP (*Internet Protocol*). Com estas informações, são enviados *frames* de comando para cada um dos elementos da lista, solicitando em um primeiro momento *frames* de configuração e cabeçalho.

O dispositivo que recebe os comandos deve estar preparado para interpretar corretamente a solicitação do PDC, respondendo prontamente os dados requeridos. Uma vez atendidas estas solicitações, o PDC armazena as informações contidas nos *frames*, e novo *frame* de comando é enviado para o dispositivo, requisitando o início da transmissão de dados. A partir da chegada do primeiro *frame* de dados ao PDC enviado pelo equipamento, o processo de configuração estará completo.

Mesmo a operação do PDC estando devidamente configurada, devem-se considerar eventos de falhas com a comunicação dos equipamentos do SPMS. Caso uma PMU ou PDC estejam fora de funcionamento, ou venham a parar de se comunicar subitamente, não poderão enviar seus respectivos *frames* para o equipamento de destino. Da mesma maneira, um *frame* de comando enviado pelo PDC pode não chegar ao seu destino, seja por falha no link de comunicação ou defeitos próprios de *hardware*.

Estes casos são previstos e tratados de forma a não alterar o funcionamento normal do PDC, sendo para isso determinado um tempo máximo de espera pelos *frames*. Uma vez ultrapassado este tempo, uma nova etapa de configuração da operação é iniciada, repetindo-se enquanto o PDC estiver em funcionamento, ou o equipamento responder de acordo com o esperado. Isto garante que o PDC opere continuamente, independente de falhas externas que são de acontecimento frequentes.

### ***Processo de Recebimento do Frame de Dados***

Considerando o ciclo principal de operação do PDC, existem quatro possibilidades de *frames* a serem recebidos. Nos casos se tratando dos *frames* de cabeçalho, configuração e comando, seu processamento é direto e o principal objetivo é preparar o PDC para recebimento de dados.

Depois de estabelecidos os parâmetros de configuração, o *frame* de dados é o *frame* mais utilizado pelo sistema. No caso do SPMS do Projeto MedFasee, a taxa de atualização do sistema é de 60 *frames* por

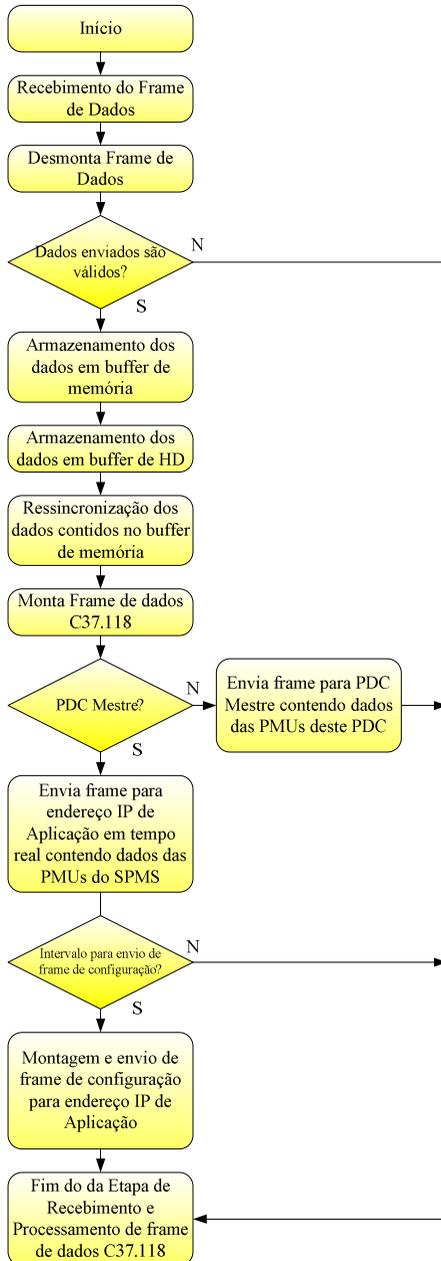
segundo, ou seja, 1 *frame* de dados enviado ao PDC por ciclo de 60 Hz, para cada PMU do SPMS.

Desta forma, faz-se necessário descrever os processos que envolvem o recebimento do *frame* de dados. Estes procedimentos permitem o processamento, armazenamento e disponibilização dos sincrofasores que chegam ao PDC. A Figura 3.13 ilustra o fluxograma do processo de recebimento de *frame* de dados do PDC.

O processo de recebimento do *frame* de dados é iniciado quando um *frame* de dados é identificado na Etapa Principal da Operação do PDC. Após esta identificação, verifica-se a validade dos dados do ponto de vista da qualidade da medida fasorial. Em caso positivo, inicia-se a varredura dos parâmetros contidos nele, baseado na seqüência fornecida pelo *frame* de configuração, fornecido pela(o) PMU/PDC. Desta forma, o PDC saberá identificar a ordem em que os dados devem ser interpretados, associando-se os equipamentos e seus respectivos canais fasoriais, analógicos e digitais.

Extraindo-se os valores de módulo, ângulo e etiqueta de tempo contidos no *frame*, estes são armazenados em *buffer* de memória, para posterior re-sincronização. Da mesma maneira, o *buffer* em HD é acessado, armazenando-se os dados em arquivos texto de formato específico. Um processo independente, executado simultaneamente ao PDC é encarregado de ler os arquivos do *buffer* e inserir os dados nas tabelas do banco de dados MySQL.

Os dados contidos no *buffer* em memória são encaminhados a um processo de re-sincronização, que tem por objetivo alinhar os dados recebidos em diferentes instantes de tempo. Isto é necessário dado que o envio de dados pela rede é feito de maneira assíncrona. Como resultado deste processo, um pacote de dados é disponibilizado em memória. Na posse destes dados, um novo *frame* de dados é criado pelo PDC. O destino destes dados depende da função hierárquica do PDC. Se for *PDC Front End*, os dados são enviados para o PDC Mestre. No caso de PDC Mestre, os dados são enviados para a lista de aplicações pré-definidas. Esta lista é presente no arquivo de configuração XML.



*Figura 3.13 – Processo de Recebimento do Frame de Dados IEEE C37.118 do PDC.*

#### 3.4.3.4 Processo de Acesso e Escrita no Banco de Dados

De acordo com o apresentado anteriormente, o processo computacional do PDC, no momento em que processa o frame de dados, armazena os valores dos fasores em arquivos de *buffer* em HD e em memória. O uso do *buffer* em memória justifica-se pela necessidade de resincronização dos dados e na maior velocidade de processamento, uma vez que estes dados possuem prioridade de tempo real.

Por outro lado, para o armazenamento histórico, não existe necessidade de disponibilização imediata, dado que não é necessário resincronizar os dados para a inserção na base histórica. Isto se deve pelo fato do PDC MedFasee utilizar um banco de dados MySQL, que realiza o gerenciamento dos dados de forma que, quando consultado, a disponibilização é apresentada com fasores na mesma base de tempo.

A estratégia de utilizar um *buffer* em HD, ao invés de acessar diretamente o banco de dados no processo principal do PDC, tem por objetivo separar o processo de aquisição e processamento dos fasores do processo de acesso e inserção de dados no banco. Devido às características de tempo real do processo principal do PDC, não se deseja que a aquisição dos fasores seja interrompida devido a consultas na base de dados histórica, ou pela própria atualização da base com dados recém chegados das PMUs, uma vez que o acesso ao banco de dados é feita de maneira seqüencial, sendo necessária a finalização de uma consulta para o início de outra.

Desta forma, um processo executado em conjunto com o PDC se encarrega de acessar os arquivos do *buffer* em HD, conectar-se ao banco de dados, inserir dados nas tabelas e apagar os arquivos de origem do *buffer*. Com isso, garante-se que cada processo tenha sua devida prioridade em relação ao outro, e o processo de aquisição e processamento do PDC não seja interrompido em nenhuma etapa. A Figura 3.14 apresenta a relação entre os processos de tempo real e de armazenamento do PDC.

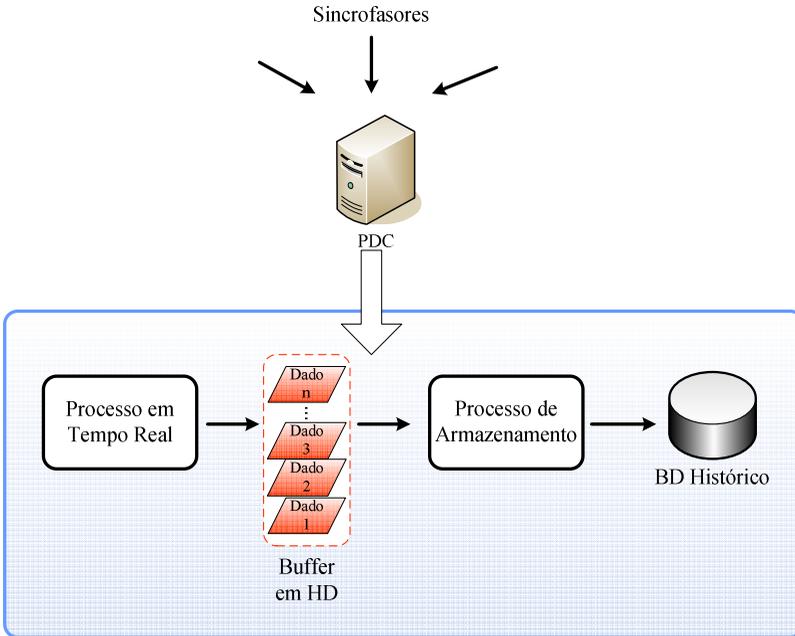


Figura 3.14 – Processos de Tempo Real e Armazenamento que compõe o PDC.

Para o processo principal, uma vez gravado os dados no *buffer* em HD, sua execução continua sem observar os acontecimentos posteriores aos dados ali gravados. Do ponto de vista do Processo de Armazenamento, os arquivos de *buffer* são dados de entrada do processo, que se encarrega de executar a consulta ao banco de dados MySQL, inserindo os dados nas respectivas colunas das tabelas pré-definidas.

Cabe ressaltar que o *buffer* de dados em HD é limitado pelo espaço livre em disco do computador em que o PDC está sendo executado. No caso de consultas ao banco de dados histórico para estudos, o processo armazenador pára as inserções nas tabelas, esperando o fim do processamento do pedido. Enquanto isso, o *buffer* continua sendo preenchido, ocupando maior espaço em disco progressivamente, até o momento em que seja finalizada a consulta. Em seguida, o armazenador retoma suas atividades, inserindo os dados acumulados no HD a taxas maiores que o preenchimento do *buffer*. Isto elimina o risco de ultrapassagem do limite de armazenamento de dados do HD.

### 3.5 Comentários Finais

Neste capítulo foram apresentados os principais aspectos conceituais de desenvolvimento e operação do PDCS. Primeiramente foram mostradas soluções encontradas pela literatura e no mercado para a solução de aquisição de dados de SPMS, que fazem parte do conteúdo de conhecimento para a elaboração do PDCS proposto no trabalho.

Os requisitos e premissas consideradas no desenvolvimento do trabalho foram apresentados, observando-se a necessidade de um sistema de concentração de dados de maior confiabilidade, flexibilidade e expansibilidade. Para atender estes requisitos, a proposta do PDCS contempla uma arquitetura descentralizada de aquisição de dados, organizada de maneira hierárquica em dois níveis. Para viabilizar a implementação do PDCS, apresentam-se os formatos de dados utilizados na comunicação entre os PDCs do sistema, padrões estabelecidos pela norma IEEE C37.118.

O capítulo aborda os aspectos de operação do PDCS, separando as lógicas de operação do PDCS e do PDC. Dentro do aspecto de operação do PDC, são apontadas suas divisões internas de tarefas. Primeiramente um processo em tempo real é responsável pela aquisição de dados e processamento dos *frames* enviados pelas PMUs, tomando-se os procedimentos cabíveis para cada caso. Um *buffer* em memória e outro em HD são utilizados para aplicações que façam uso dos sincrofasores em tempo real, e para inserção na base histórica, respectivamente.



# CAPÍTULO 4

## 4. Projeto de *Software* e Implementação Computacional

O presente capítulo tem por objetivo apresentar o projeto de *software* desenvolvido neste trabalho. Este projeto é adicionado aos desenvolvimentos do PDC MedFasee, de forma que as novas funcionalidades permitam a comunicação entre equipamentos com função de PDC. Esta comunicação é característica fundamental para a viabilização da arquitetura proposta do PDCS.

São apresentados os conceitos do paradigma da Modelagem Orientada a Objetos (MOO), bem como sua aplicação no desenvolvimento do *software* do PDC. As classes desenvolvidas para o projeto são descritas. Primeiramente é apresentado o conjunto de classes e sua relação, por meio de diagrama UML. Em seguida, uma descrição individual de cada classe é realizada, abrangendo os principais aspectos de cada caso.

### 4.1 Paradigma da Modelagem Orientada a Objetos

Ao se desenvolver um *software*, espera-se que este seja capaz de realizar uma determinada tarefa, de acordo com rotinas e instruções padronizadas, criadas pelo desenvolvedor. Estas rotinas, a fim de garantir a correta organização do código-fonte, devem ser distribuídas e classificadas de acordo com suas funcionalidades e seus objetivos.

Para cumprir a meta descrita, procura-se com os conceitos da MOO desenvolver e organizar um projeto em sua totalidade, partindo da modelagem de suas partes. Estas partes são denominadas *objetos*, que têm por objetivo representar (abstrair) o mundo real que está sendo estudado.

Os objetos são a parte central do paradigma, definidos de forma a manterem relações com outros objetos, estabelecendo um fluxo organizado de troca de informações, característica que facilita o desenvolvimento, manutenção e expansão do projeto de *software* (RUMBAUGH, J., *et al.*, 1994).

A fase de desenvolvimento de um projeto de *software*, que utiliza os conceitos da MOO, concentra boa parte de seus esforços em suas

fases iniciais. Nesta fase são definidos os objetos que fazem parte do *software*, sendo estabelecidas suas inter-relações, para finalmente produzir uma aplicação computacional capaz de realizar as tarefas desejadas. Como consequência, a implementação dos conceitos será facilitada, reduzindo o tamanho do código fonte e melhorando a qualidade da organização dos desenvolvimentos.

Projetos baseados em MOO têm por característica organizar os objetos em duas partes fundamentais: *atributos* e *métodos*. Os atributos tratam das características do objeto. São variáveis que armazenam os valores ou estados do sistema. Por sua vez, os métodos são funções, que determinam a forma de manipular os valores contidos nos atributos.

Um objeto só existe por meio de sua descrição. Ela é realizada dentro de sua respectiva *classe*. As classes podem ser entendidas como a forma para os objetos de um determinado projeto, sendo que uma *instância* de uma classe dá origem a um objeto. Esta diferença conceitual é de extrema importância, dado que, embora a definição de uma classe seja individual, os objetos podem ser instanciados tantas vezes quanto necessário.

Outro aspecto a ser observado é o conceito de *encapsulamento* de dados. Esta característica da MOO introduz o controle de acesso de informações entre diferentes objetos, garantindo que procedimentos internos e exclusivos de cada objeto sejam protegidos de ações externas de outras partes do projeto de *software*. Isto agrega independência ao desenvolvimento, resultando em um *software* mais robusto e menos sensível a modificações.

Também fazem parte dos conceitos de MOO as *ligações* e *associações*. Assim como objetos estão para classes, ligações são instâncias de associações. As associações descrevem os relacionamentos conceituais entre objetos que estão ligados entre si. Existem tipos especiais de associações, tais como *agregação*, *composição* e *herança*.

A agregação é uma associação que define um objeto por meio de suas partes. Trata-se de uma coleção de objetos menores agregados de maneira a formar um objeto maior. Porém, instâncias dos objetos menores podem existir independentes do objeto agregado, inclusive sendo utilizadas em outras associações ou agregações. Para casos onde diversos objetos são partes de um objeto maior, porém as partes não tem sentido de existir isoladamente, existe aí uma associação de composição. A herança se trata da possibilidade de uma classe herdar características em comum de outra classe. A classe de origem é chamada de classe base, onde podem ser agrupadas as interseções conceituais de um con-

junto de diferentes classes, que serão desenvolvidas a partir da derivação da classe base.

## 4.2 Implementação do código-fonte do PDC – Linguagem e compilador

O código-fonte do PDC foi desenvolvido em sua plenitude em linguagem de programação C++. Esta linguagem tem suporte para os conceitos da MOO e disponibiliza ferramentas para o desenvolvimento de *software*. Entre elas, destaca-se a *Standard Template Library* (STL), biblioteca que contém classes, algoritmos, operadores e iteradores. Esta biblioteca tem por objetivo servir de alicerce para qualquer desenvolvimento em C++, inclusive para novas bibliotecas que possam vir a ser desenvolvidas.

As implementações realizadas pela STL são focadas na facilidade de uso de seus contêineres e na otimização do uso de memória do equipamento e maximização da velocidade de processamento dos dados, bem como seu código-fonte é de padrão universal, podendo ser usado qualquer compilador. Desta forma, procurou-se sempre que possível fazer uso dos elementos presentes na STL a fim de agregar poder de processamento ao PDC, que juntamente com as implementações de classes da linguagem C++, originaram um projeto de *software* robusto, expansível, modular e eficiente, compatível com o paradigma da Modelagem Orientada a Objetos.

Referente ao compilador utilizado, tendo em vista que o processo computacional do PDC foi desenvolvido para executar em sistemas operacionais GNU/Linux, utilizou-se o GCC (GNU Compiler Collection) para geração do arquivo executável. O GCC é uma coletânea de compiladores compatível com as linguagens C, C++, Objective-C, Fortran, Java e ADA.

## 4.3 Classes do projeto de *software* do PDC

O paradigma da MOO foi utilizado na concepção do PDC Med-Fasee, sendo estendido para comportar a comunicação entre equipamentos com funcionalidade de PDC, viabilizando a arquitetura proposta para o PDCS. Desta maneira é possível compor um sistema modular e expansível, preparado para receber novas atribuições sem necessitar grandes alterações nos desenvolvimentos previamente realizados.

Os conceitos foram aplicados no sentido de representar os equipamentos envolvidos na medição e aquisição de sincrofasores, e processos internos à operação do PDC. A descrição das classes utilizadas para o desenvolvimento do PDC será apresentada nas seções a seguir, abor-

dando seus aspectos conceituais e de implementação. Tais classes podem ser classificadas da seguinte maneira:

- Classes de representação de estruturas físicas: C\_PDC, C\_PMU\_PDC e C\_PDC\_PDC;
- Classes de representação de processos internos ao PDC: C\_Conexao, C\_Frame e C\_App.

Para a representação gráfica das classes e seus relacionamentos, utilizam-se diagramas UML (FOWLER, M., *et al.*, 2000). Primeiramente é apresentado o diagrama UML contendo o escopo geral do projeto de *software* desenvolvido. Em seguida, são descritos os objetivos específicos de cada classe, seguido das motivações e necessidades para cada caso, passando para a apresentação de suas funcionalidades e estabelecendo as relações com outras classes envolvidas no processo. Ao final da descrição das classes, uma representação individualizada em diagrama UML será apresentada, procurando ilustrar os principais aspectos a serem observados em cada caso.

#### **4.3.1 Projeto de *software* do PDC**

O diagrama UML ilustrado na Figura 4.1 constitui o combinado das classes que são descritas neste capítulo. Este diagrama estabelece suas respectivas relações.

A estrutura apresentada é parte do projeto de *software* desenvolvido para viabilizar a aquisição dos sincrofasores calculados pelas PMUs. O objetivo deste diagrama é ilustrar os desenvolvimentos que foram adicionados ao código-fonte do PDC MedFasee, e que foram o escopo da implementação do PDCS. Este projeto permite a comunicação entre equipamentos com funcionalidade de PDC, característica necessária para viabilizar a arquitetura descentralizada, premissa básica nos desenvolvimentos deste trabalho.

Observa-se, pela figura, a relação entre as classes do projeto. A classe C\_PDC é, em essência, a composição das instâncias das classes C\_PMU\_PDC, C\_PDC\_PDC, C\_App e C\_Conexao. Ela representa o equipamento PDC, onde são processados, armazenados e disponibilizados os sincrofasores.

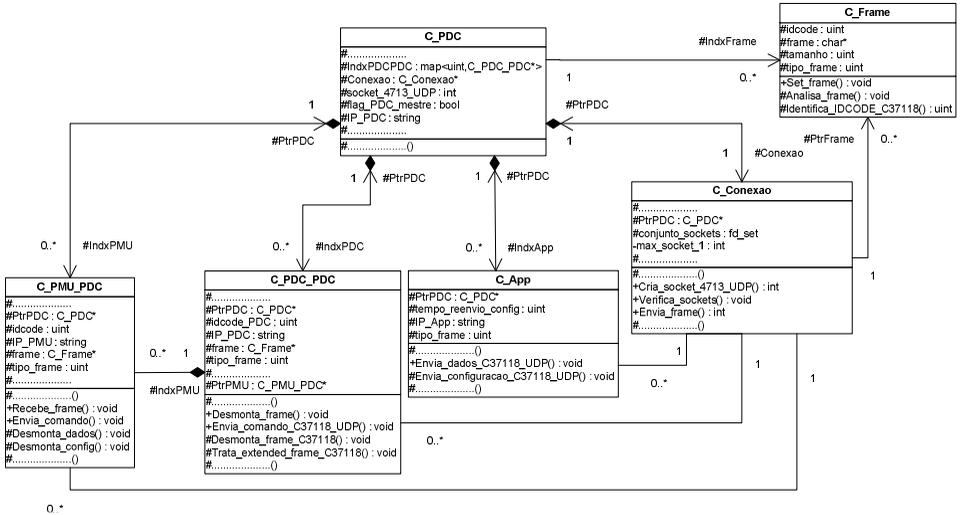


Figura 4.1 – PDC – Diagrama UML de do projeto de software do PDC.

As classes C\_PMU\_PDC e C\_PDC\_PDC representam equipamentos físicos que se comunicam com o PDC. O número de instâncias destas classes depende do SPMS, e são definidos no arquivo de configuração XML do PDC. A classe C\_Conexao possui apenas uma instância, responsável pelo gerenciamento dos protocolos de rede do PDC. A classe C\_Frame representa os *frames* enviados ao PDC, com número de instâncias igual à quantidade de *frames* recebidos em um ciclo de varredura de dados. A classe C\_App representa as aplicações que fazem uso dos dados em tempo real, disponibilizados pelo PDC. Seu número de instâncias é definido no arquivo de configuração XML do PDC.

As seções a seguir apresentam em detalhes os objetivos, funcionalidades e aspectos de cada classe do projeto de *software* do PDC.

### 4.3.2 Classe C\_PDC

Considerando-se o fato de o PDC ser composto por um equipamento único, que receberá dados de diversos equipamentos dentro do SPMS, é notória a necessidade da representação deste elemento de forma individualizada. Desta forma, armazenam-se, em seu escopo, as informações que irão subsidiar as tomadas de decisão pelo *software* no momento em que *frames* chegam ao equipamento. Ainda, por se tratar de um dispositivo protagonista no sistema de aquisição de dados, as

interfaces de tempo real, de configuração, de comunicação e de monitoração do sistema devem ser gerenciadas por este equipamento.

A classe `C_PDC` é a principal classe deste projeto de *software*, com apenas um objeto instanciado. Ela contém os parâmetros necessários para a coordenação e execução dos processos envolvidos para a concentração de dados. O diagrama UML da classe é apresentado na Figura 4.2.

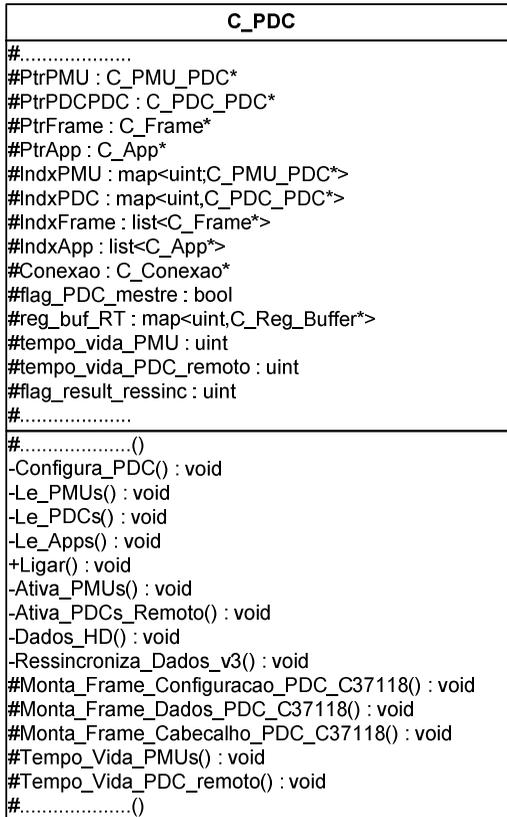


Figura 4.2 – Diagrama UML da Classe `C_PDC` - Atributos e Métodos.

Os principais métodos e atributos são:

- *PtrPMU*: ponteiro para objeto da classe `C_PMU_PDC`;
- *PtrPDCPDC*: ponteiro para objeto da classe `C_PDC_PDC`;
- *PtrFrame*: ponteiro para objeto da classe `C_Frame`;
- *PtrApp*: ponteiro para objeto da classe `C_App`;

- *IndxPMU*: lista contendo os ponteiros para objetos da classe *C\_PMU\_PDC*;
- *IndxPDC*: lista contendo os ponteiros para objetos da classe *C\_PDC\_PDC*;
- *IndxFrame*: lista contendo ponteiros para objetos da classe *C\_Frame*;
- *IndxApp*: lista contendo ponteiros para objetos da classe *C\_App*;
- *Conexao*: ponteiro para objeto da classe *C\_Conexao*
- *flag\_PDC\_mestre*: flag indicador da função hierárquica do PDC.
- *reg\_buf\_RT*: lista contendo os dados re-sincronizados em memória;
- *tempo\_vida\_PMU*: tempo limite, em segundos, para objetos da classe *C\_PMU\_PDC* serem considerados inativos;
- *tempo\_vida\_PDC\_remoto*: tempo limite, em segundos, para objetos da classe *C\_PDC\_PDC* serem considerados inativos;
- *Configura\_PDC()*: método de leitura dos parâmetros de configuração da operação do PDC fornecidos pelo arquivo de configuração XML;
- *Le\_PMUs()*: método de leitura dos parâmetros relativos às PMUs do arquivo de configuração XML;
- *Le\_PDCs()*: método de leitura dos parâmetros relativos aos PDCs do arquivo de configuração XML;
- *Le\_Apps()*: método de leitura dos parâmetros relativos às aplicações em tempo real do arquivo de configuração XML;
- *Ligar()*: método que inicia o processo do PDC;
- *Ativa\_PMUs()*: método que solicita os parâmetros de configuração das PMUs;
- *Ativa\_PDCs\_remoto()*: método que solicita os parâmetros de configuração dos PDCs;
- *Dados\_HD*: método de injeção de dados no *buffer* em HD;
- *Ressincroniza\_Dados\_V3()*: método de re-sincronização dos dados contidos nos *buffers* em memória dos objetos das classes *C\_PMU\_PDC* e *C\_PDC\_PDC*;
- *Monta\_Frame\_Configuracao\_C37118*: método de criação do *frame* de configuração da norma IEEE C37.118;
- *Monta\_Frame\_Dados\_C37118*: método de criação do *frame* de dados da norma IEEE C37.118;
- *Monta\_Frame\_Cabecalho\_C37118*: método de criação do *frame* de cabeçalho da norma IEEE C37.118;
- *Tempo\_Vida\_PMUs()*: método que determina o tempo em que os objetos da classe *C\_PMU\_PDC* estão inativos;

- *Tempo\_Vida\_PDC\_remoto()*: método que determina o tempo em que os objetos da classe *C\_PDC\_PDC* estão inativos;

As seções a seguir apresentam as principais características e funcionalidade da classe *C\_PDC*, abordando a utilização dos atributos e métodos apresentados previamente.

#### **4.3.2.1 Configuração dos atributos da Classe *C\_PDC***

Os atributos da classe *C\_PDC* envolvem parâmetros de configuração e dados de entrada, fornecidos a partir do arquivo de configuração XML. Este arquivo fornece parâmetros relativos ao SPMS (por exemplo, taxa de aquisição de fasores, número de PMUs/*PDCs Front End*, sinalizador indicativo da função hierárquica do PDC) e parâmetros de funcionalidades específicas (registro de logs da operação do sistema, utilização de protocolos de tempo real, endereço IP da interface de rede local, entre outros). O método *C\_PDC::Configura\_PDC()* realiza o acesso a estas informações. Neste momento o objeto da classe *C\_PDC* passa a utilizar métodos específicos para gerenciar os protocolos de tempo real, solicitar parâmetros de configuração dos equipamentos, executar constante varredura da interface de rede em busca de sincrofasores e identificar a origem dos *frames* encaminhados ao PDC.

Especificamente com relação à representação dos equipamentos do SPMS, três métodos fazem acesso aos campos do arquivo XML que contêm os dados das PMUs, PDCs e Aplicações em tempo real, sendo eles: *C\_PDC::Le\_PMUs()*, *C\_PDC::Le\_PDCs()*, *C\_PDC::Le\_Apps()*. Estes métodos preenchem os atributos *C\_PDC::IndxPMU*, *C\_PDC::IndxPMU* e *C\_PDC::IndxPMU*, respectivamente. Estes atributos são listas que servem de base para coordenar o envio e recebimento de sincrofasores.

Importante destacar a característica modular da classe *C\_PDC*. Caso seja necessária a inclusão de novos equipamentos ou modificação de parâmetros operativos, basta alterar o arquivo XML, incluindo ou modificando os parâmetros necessários, de forma que o objeto PDC saiba a quantidade de memória a ser alocada dinamicamente para cada objeto instanciado em seu escopo.

#### **4.3.2.2 Ciclo de aquisição de dados e *buffer* em HD**

Após a configuração dos atributos da classe *C\_PDC*, os equipamentos do SPMS, referentes aos objetos instanciados das classes *C\_PMU\_PDC* e *C\_PDC\_PDC*, necessitam ser contatados. Este proce-

dimento é necessário para que o PDC possa ter acesso aos respectivos parâmetros de configuração. Isto é feito utilizando *frames* de comando, de forma a solicitar o envio, pelos equipamentos, dos seus respectivos *frames* de cabeçalho e configuração. Para isto, são utilizados os métodos *C\_PDC::Ativa\_PMUs()* e *C\_PDC::Ativa\_PDCs\_remoto()*.

Uma vez que os objetos estejam configurados, pode-se iniciar o envio de *frames* de dados. A partir deste momento é dado início ao ciclo permanente de varredura na interface de rede do computador, identificando o recebimento de *frames* enviados por equipamentos do SPMS ao PDC, de acordo com o apresentado no Capítulo 3, ilustrado pela Figura 3.11.

O método utilizado para receber os dados da rede é *C\_Conexao::Verifica\_sockets()*, chamado a partir do escopo da classe *C\_PDC*. Quando um *frame* é detectado, sua origem é identificada, preenchendo-se o atributo *C\_PDC::IndxFrame*. Este atributo é acessado, sendo que os dados são repassados aos objetos específicos das classes *C\_PMU\_PDC* e *C\_PDC\_PDC*, para tratamento. Os sincrofases são extraídos dos *frames* e armazenados no *buffer* em HD, por meio do método *C\_PDC::Dados\_HD*, e no *buffer* em memória, por meio de métodos específicos das classes *C\_PMU\_PDC* e *C\_PDC\_PDC*.

O *buffer* em HD tem por objetivo disponibilizar os dados recebidos para inserção no banco de dados histórico. Sua composição se trata de arquivos texto ASCII de formatação própria contendo dados relativos ao período de um segundo de aquisição de todas as *PMUs/PDCs Front End* que enviam dados ao PDC. Estes arquivos são posteriormente acessados pelo Processo de Armazenamento, sendo inseridos na base dados MySQL.

Cabe ressaltar que, por se tratar de um procedimento que envolve disponibilização de dados históricos, o armazenamento, tanto no *buffer* em HD, quanto na base histórica, é feito sem qualquer processo de sincronização de etiquetas de tempo contidas nos sincrofases. Esta atribuição é deixada para o sistema de gestão proprietário do banco de dados MySQL.

#### **4.3.2.3 Disponibilização de dados em tempo real e *buffer* em memória**

Considerando-se o fornecimento de dados em tempo real pelo PDC, têm-se como principais requisitos a disponibilização dos sincrofases no menor tempo possível e a necessidade de sincronismo entre dados de diferentes equipamentos.

Contudo, apesar do cálculo dos sincrofasores ser processado de maneira sincronizada, a transmissão de dados nos canais de comunicação é realizada de maneira assíncrona. Atrasos são inseridos no processo de transporte pela rede, e dependem basicamente do caminho lógico percorrido pelos *frames* entre as PMU e os PDCs. Em se tratando de canais compartilhados por outras aplicações, tal como a Internet, ou no caso das redes corporativas das empresas, ou mesmo as redes de operação, estes atrasos podem ser significativos.

É necessário um método que disponibilize os dados dos equipamentos do SPMS em uma mesma base de tempo, alinhando as etiquetas de tempo dos *frames* recebidos pelo PDC. O método encarregado desta tarefa é denominado de *C\_PDC::Ressincroniza\_Dados\_V3()*. Este método acessa o *buffer* em memória dos objetos das classes *C\_PMU\_PDC* e *C\_PDC\_PDC*, observando as etiquetas de tempo dos *frames* armazenados, tendo como resultado o preenchimento do atributo *C\_PDC::reg\_buf\_RT*. Este atributo é o pacote de dados res-sincronizados, que subsidia a montagem do *frame* de dados pelo método *C\_PDC::Monta\_Frame\_Dados\_C37118()*.

Os tópicos a seguir abordam em detalhes o *buffer* em memória e o processo de re-sincronização de dados. Tais elementos são fundamentais para a disponibilização dos sincrofasores em tempo real, e também são responsáveis por viabilizar a comunicação entre PDCs, necessária para a implementação do PDCS.

### ***Buffer circular em memória***

Cada objeto tipo *C\_PMU\_PDC* e *C\_PDC\_PDC* tem um *buffer* em memória, contendo os  $n$  últimos frames de dados recebidos do respectivo equipamento real. A Figura 4.3 apresenta a estrutura interna de armazenamento do *buffer* circular em memória.

Este *buffer* é representado pela classe *C\_Buffer*, a qual armazena  $n$  objetos tipo *C\_Reg\_Buffer* em um vetor. Cada objeto do tipo *C\_Reg\_Buffer* armazena um *frame* de dados recebido do equipamento real, seja uma PMU ou um *PDC Front End*. A classe possui um atributo que armazena a etiqueta de tempo, representada pela classe *C\_Tempo\_Frame*; um vetor de objetos do tipo *C\_Phaseor*, cada um representando um fasor (módulo e ângulo) dentro dos *frames* de dados enviados pelo equipamento real; e mais duas variáveis representando a frequência (*freq*) e a variação de frequência (*dfreq*) enviadas no *frame* de dados.

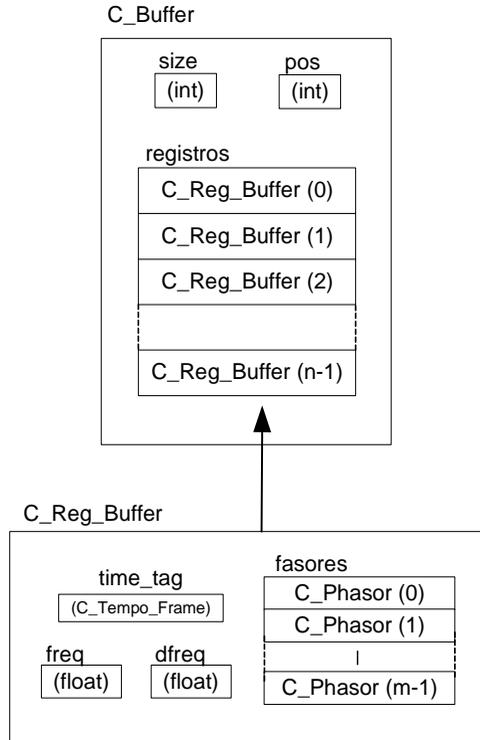


Figura 4.3 - Estrutura Interna do Buffer Circular em Memória do PDC.

O tamanho do *buffer*, ou seja, a quantidade de frames (objetos tipo *C\_Reg\_Buffer*) armazenados em cada objeto tipo *C\_Buffer*, depende do atraso máximo permitido para a re-sincronização dos dados. O atraso máximo é configurado por meio do arquivo de Configuração XML do PDC. O *buffer* é dimensionado automaticamente, de forma a armazenar *frames* de dados suficientes para que o PDC opere adequadamente com os atrasos configurados.

Um diagrama de classes, mostrando de maneira simplificada as relações entre as classes que formam o *buffer*, é apresentado na Figura 4.4.

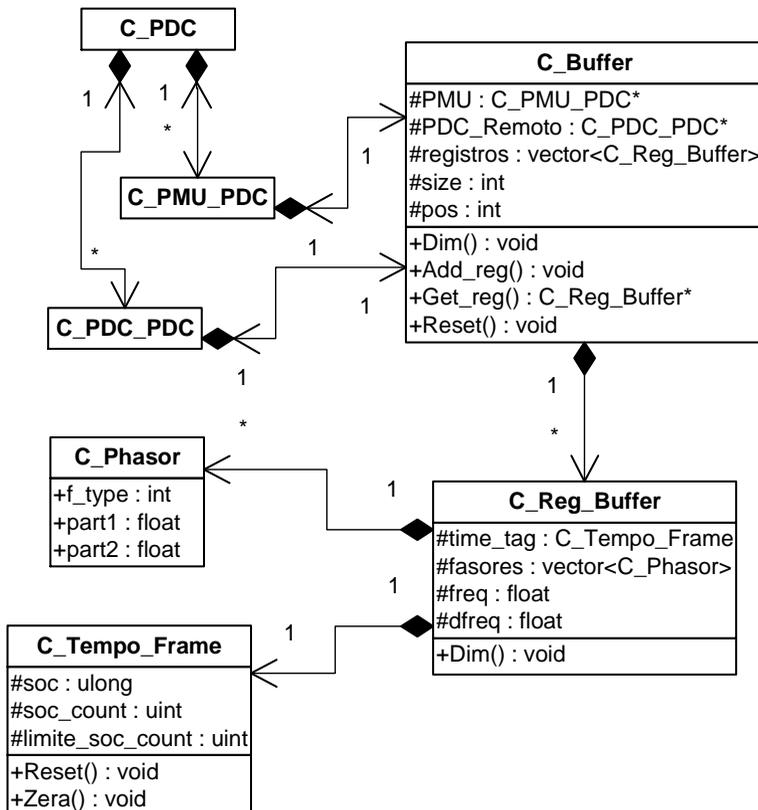


Figura 4.4 - Diagrama de Classes do Buffer Circular em Memória.

A seção a seguir é dedicada para o detalhamento do processo de re-sincronização de dados. Este processo utiliza a estrutura do *buffer* para disponibilizar pacotes de dados em uma mesma base de tempo, oriundos de diferentes equipamentos do SPMS.

### Método de Re-sincronização de Dados

O processo de re-sincronização de dados é parte fundamental no desenvolvimento do PDCS, dado que sem este procedimento não seria possível alocar em um mesmo *frame* de dados os sincrofases de diferentes PMUs/PDCs.

O PDC utiliza atualmente seis parâmetros para realizar a re-sincronização dos dados: *tempo-base*, *tempo-de-busca*, *tempo-enviado*, *atraso-mínimo* e *atraso-máximo*. O *tempo-base* é atualizado cada vez que um novo *frame* de dados chega ao PDC (*Front End* ou Mestre),

caso a etiqueta de tempo do novo frame seja mais atual que o *tempo-base* corrente. O *tempo-de-busca* é a etiqueta de tempo corrente que será buscada nos *buffers* em memória do PDC. O *tempo-enviado* contém a etiqueta de tempo do pacote de dados disponibilizado mais recentemente. A atualização do *tempo-de-busca* é realizada incrementando-se uma unidade o valor do *tempo-enviado*. A Figura 4.5 ilustra uma interpretação gráfica dos intervalos de tempo utilizados no acesso ao *buffer* em memória dos objetos C\_PMU\_PDC e C\_PDC\_PDC para execução do processo de re-sincronização de dados.

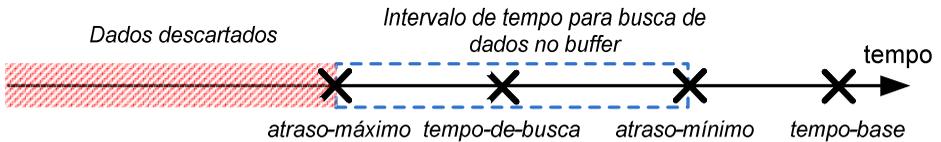


Figura 4.5 – Interpretação gráfica dos tempos da re-sincronização de dados em memória.

Em condições normais de operação, o PDC não disponibiliza dados com etiquetas de tempo maiores (mais atuais) que a diferença entre o *tempo-base* e o *atraso-mínimo*, tampouco com etiquetas menores (mais atrasadas) que a diferença entre o *tempo-base* e o *atraso-máximo*. As etiquetas de tempo dos pacotes de dados disponibilizados em tempo real pelo PDC têm atrasos que variam na faixa [*atraso-mínimo* ; *atraso-máximo*], de acordo com o atraso na chegada dos *frames* de dados de cada equipamento ao PDC.

Casos excepcionais de operação foram observados, em momentos onde todos os equipamentos remotos param de enviar dados por certo período de tempo. Desde que este período seja menor que *atraso-máximo*, ao chegar o primeiro novo frame de dados válido de um equipamento remoto, após a interrupção, o PDC recomeçará a enviar os dados recebidos antes da interrupção, armazenados nos respectivos *buffers*, observando o limite da diferença entre *tempo-base* atualizado e *atraso-máximo*. Se o período de interrupção for maior que o atraso máximo configurado, os dados recebidos antes da interrupção são considerados atrasados demais, e automaticamente descartados do processo de disponibilização em tempo real. Observe-se, porém, que estes dados ainda são armazenados normalmente no banco de dados do PDC, dado que o *buffer* em HD não considera atrasos de recebimento de dados. O

processo de re-sincronização é realizado a cada vez que um novo *frame* de dados é recebido pelo PDC, tenha ele uma etiqueta de tempo mais atual que o *tempo-base* ou não.

A estrutura *C\_PDC::reg\_buf\_RT* é uma lista de s para objetos tipo *C\_Reg\_Buffer*, indexada pelo código de identificação IDCODE do equipamento do qual veio o *frame* de dados. Os pares da lista são criados automaticamente durante a leitura da lista de PMUs (para o caso de *PDC Front End*) e da lista de PDCs (para o caso de PDC Mestre), sendo que a lista contém um par para cada equipamento remoto com o qual o PDC estabelecerá comunicação.

O objeto tipo *C\_PDC*, ao executar o método *C\_PDC::Ressincroniza\_Dados\_V3()*, preenche seu atributo *C\_PDC::reg\_buf\_RT*, a medida que percorre as suas listas de PMUs e/ou *PDCs Front End* do PDC, buscando os dados relativos a um determinado tempo de busca. Caso existam todos os dados em uma determinada execução do processo de re-sincronização, a lista é completada. Caso haja perda de dados de algum equipamento, sobram pares em branco na lista, o que não afeta o desempenho do restante do PDC.

O *flag C\_PDC::flag\_result\_ressinc* indica o resultado do processo de re-sincronização. No método *C\_PDC::Ressincroniza\_Dados\_V3()* atribui-se o valor deste *flag* de acordo com a situação atual do conjunto de dados a ser disponibilizado em tempo real. Seus valores possíveis são:

- 1 → indica a finalização da ressincronização sem a disponibilização de um pacote de dados em tempo real;
- 2 → indica a montagem de um pacote de dados ressincronizados com todos os equipamentos em falta (pacote vazio);
- 3 → indica a montagem de um pacote de dados ressincronizados com equipamentos em falta (pacote incompleto);
- 4 → indica a montagem de um pacote de dados ressincronizados completo.

Com o procedimento descrito acima, o PDC tenta operar sempre com o menor atraso especificado (atraso mínimo). Caso os dados de um ou mais elementos do SPMS passarem a chegar com atrasos maiores em relação ao elemento mais rápido, o PDC aumenta automaticamente o atraso até *atraso-medio* especificado. Quando os dados dos elementos

mais atrasados voltarem a chegar ao PDC com atrasos menores, automaticamente o PDC passa a usar atrasos menores para a disponibilização de dados de tempo real.

Quanto maior o atraso máximo, por exemplo, maior será o tamanho do *buffer* de cada objeto, e, portanto, maior a exigência de memória para a operação de cada PDC. O atraso de chegada de frames de dados de cada equipamento no SPMS é variável, dependendo de fatores tais como carregamento da rede entre os equipamentos, largura de banda disponível, nível de compartilhamento da rede com outras aplicações, etc. Portanto, não se pode definir a priori o atraso do equipamento conectado ao caminho lógico mais lento, para uma definição ideal do atraso médio.

No caso de o PDC fornecer dados em tempo real para aplicações de controle, pode ser necessário ainda que o atraso dos *frames* de dados re-sincronizados seja conhecido, com um valor fixo ou ao menos dentro de uma pequena faixa de variação. Para que o PDC opere com um atraso fixo, os atrasos mínimo e médio devem ser configurados iguais. Desta forma, em condições normais de operação, o PDC sempre disponibilizará um *frame* de dados re-sincronizados em tempo real, com um atraso igual ao atraso mínimo e médio configurado, mesmo que o frame esteja incompleto. Observe-se que o atraso máximo somente é usado em casos eventuais, após um período de perda de dados de todos os equipamentos simultaneamente.

Ressalte-se que o atraso envolvido no processo de re-sincronização é parte do atraso total dos dados, desde a sua medição nas PMUs até a sua disponibilização final para as aplicações. Para se obter o atraso total deve-se somar ao atraso de re-sincronização de cada nível hierárquico na estrutura do PDCS, os atrasos devido ao processamento em si dos equipamentos, tanto nas PMUs quanto nos PDCs, e o atraso devido às redes de comunicação entre os equipamentos.

A ordem de grandeza do atraso total depende de cada situação de instalação e de operação de um SPMS. Uma avaliação realista do atraso total deve ser realizada em campo, durante a operação contínua do sistema.

#### **4.3.2.4 Monitoração do estado operativo dos equipamentos do SPMS**

Outra característica presente nas atribuições da classe *C\_PDC* é a constante monitoração do estado operativo dos equipamentos do SPMS com os quais o PDC se comunica.

Objetos do tipo `C_PMU_PDC` e `C_PDC_PDC` contêm atributos que definem a quantidade de tempo em segundos que um equipamento pode ficar sem enviar qualquer tipo de *frame* ao PDC, denominados `C_PMU_PDC::tempo_vida_PMU` e `C_PDC_PDC::tempo_vida_PDC_remoto`. Estes parâmetros são definidos no arquivo de configuração XML. Após a ultrapassagem do limite de tempo adotado, é considerado que o equipamento perdeu comunicação e seu objeto é reiniciado, com todos os seus atributos trazidos aos valores originais do início da operação do sistema.

Esta ação dá início a um novo procedimento de configuração, sendo solicitados *frames* de cabeçalho e configuração. Neste caso, até nova resposta do equipamento, é considerada uma mudança de topologia do SPMS, que deverá ser informada ao PDC Mestre e às aplicações em tempo real, utilizando-se *frames* de configuração enviados imediatamente após a identificação do estouro do tempo de vida do equipamento em questão.

O procedimento acima descrito garante a constante atualização do estado de comunicação dos equipamentos, tornando o sistema autônomo para gerenciar eventuais falhas de comunicação ou qualquer contingência de ordem técnica, assegurando a operação contínua e ininterrupta necessária para um sistema de concentração de dados.

### **4.3.3 Classe `C_PMU_PDC`**

A classe `C_PMU_PDC` tem por objetivo representar o equipamento PMU, cujos objetos são instanciados dentro do escopo da classe `C_PDC`. Estes objetos contêm os parâmetros que representam sua configuração física, e que são necessários para a interpretação de seus dados. Com isso, o objeto da classe `C_PDC` tem um mapeamento dos parâmetros das PMUs com as quais ele se comunica, possibilitando a correta concentração e processamento dos dados enviados a ele. A Figura 4.6 apresenta o diagrama UML da classe `C_PMU_PDC`.

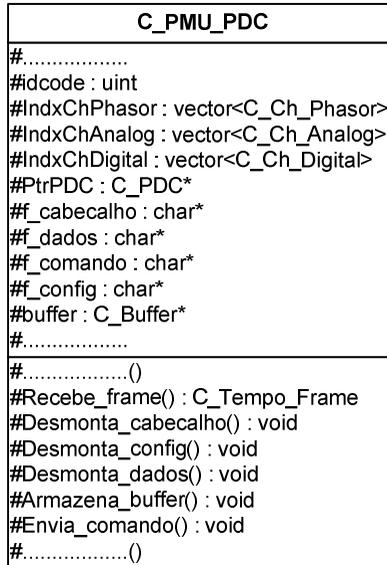


Figura 4.6 – Diagrama UML da Classe C\_PMU\_PDC.

Os principais atributos e métodos são:

- *idcode*: código de identificação numérico do equipamento no SPMS;
- *PtrFrame*: ponteiro para objeto da classe C\_Frame;
- *IndxChPhasor*: vetor contendo a ordem em que os dados dos canais fasoriais são enviados;
- *IndxChAnalog*: vetor contendo a ordem em que os dados dos canais analógicos são enviados;
- *IndxChDigital*: vetor contendo a ordem em que os dados dos canais digitais são enviados;
- *PtrPDC*: ponteiro para o objeto da classe C\_PDC;
- *f\_cabeçalho*: ponteiro para a posição de memória dos dados do *frame* de cabeçalho;
- *f\_dados*: ponteiro para a posição de memória dos dados do *frame* de dados;
- *f\_comando*: ponteiro para a posição de memória dos dados do *frame* de comando;
- *f\_config*: ponteiro para a posição de memória dos dados do *frame* de configuração;
- *buffer*: ponteiro para objeto da classe C\_Buffer;

- *Recebe\_frame()*: método de identificação do tipo de *frame* da norma IEEE C37118 recebido;
- *Desmonta\_cabecalho()*: método de processamento das informações contidas no *frame* de cabeçalho;
- *Desmonta\_config()*: método de processamento das informações contidas no *frame* de configuração;
- *Desmonta\_dados()*: método de processamento das informações contidas no *frame* de dados;
- *Armazena\_buffer()*: método de inserção dos dados no *buffer* de memória pertencente ao objeto da classe *C\_PMU\_PDC*.
- *Envia\_comando()*: método de envio de comando ao equipamento da PMU.

A principal funcionalidade da classe *C\_PMU\_PDC* é a interpretação dos *frames* em formato C37.118, enviados por PMUs ao PDC. Neste caso, o objeto da classe *C\_PDC* encaminha o *frame* a ser processado por meio da chamada do método *C\_PMU\_PDC::Recebe\_frame()*. Este método identifica o tipo de *frame* enviado, e processa os dados com auxílio de métodos específicos. As seções a seguir apresentam o tratamento dado a cada tipo de *frame* recebido pelo PDC.

#### **4.3.3.1 Processamento dos *frames* IEEE C37.118**

Toda vez que um *frame* é detectado pelo objeto da classe *C\_PDC*, ele é processado e identificado, extraindo-se seu parâmetro *IDCODE*.

O número do *IDCODE* é comparado com os valores presentes na lista de PMUs, atributo *C\_PMU\_PDC::IdxPMU*, da classe *C\_PDC*. Se este número existir, o *frame* é encaminhado para o respectivo objeto da classe *C\_PMU\_PDC*, que por meio de seus métodos deve tratar e processar as informações recebidas. Para cada tipo de *frame*, um tratamento distinto é dedicado, conforme descrito a seguir.

##### ***Frame de Cabeçalho***

Caso o tipo do *frame recebido* seja identificado como Cabeçalho, o método chamado é o *C\_PMU\_PDC::Desmonta\_config()*. Os valores dos campos em caracteres ASCII são lidos e em seguida armazenados no disco rígido, em diretório específico do computador em que processo do PDC está sendo executado.

O *frame* de cabeçalho traz informações inseridas pelo desenvolvedor tais como, nome do equipamento, versão de *firmware*, dados técnicos, entre outros.

### **Frame de Configuração**

O *frame* de configuração traz todas as informações que subsidiam a interpretação do *frame* de dados. Este *frame* é processado pelo método `C_PMU_PDC::Desmonta_config()`. Uma vez que os dados são tratados pelo método, os parâmetros são armazenados nos atributos do objeto da classe, permitindo a criação de um retrato da estrutura física da PMU que o originou.

Entre as principais informações, destacam-se o número de canais fasoriais, analógicos e digitais. Estes valores, quando extraídos do *frame*, são armazenados nos atributos `C_PMU_PDC::IndxChPhasor`, `C_PMU_PDC::IndxChAnalog` e `C_PMU_PDC::IndxChDigital`. Com isto, é possível dimensionar o tamanho do *buffer* em memória do objeto, que é posteriormente acessado pelo método de re-sincronização de dados da classe `C_PDC`. Caso seja alterada a configuração da PMU, essa mudança é informada através do *frame* de configuração, e o *buffer* é automaticamente redimensionado.

Os nomes dos canais da PMU também são fornecidos. Eles permitem a identificação dos fasores que estão sendo calculados nos canais fasoriais, os tipos de medidas analógicas que estão sendo realizadas e o significado dos canais digitais. É definido, de acordo com a norma IEEE C37.118, que a ordem em que forem lidos os parâmetros dos canais presentes no *frame* de configuração, é a mesma ordem em que são concatenados os dados enviados para o PDC, pelo *frame* de dados.

Os parâmetros do *frame* de configuração, assim como feito para o *frame* de cabeçalho, são gravados em arquivo em formato ASCII, separado no disco rígido do computador. Desta maneira, o usuário pode acessar a configuração atual do objeto da classe `C_PMU_PDC` em tempo de execução.

### **Frame de Dados**

O *frame* de dados contém efetivamente as medidas fasoriais, analógicas e digitais. Estes dados, ao chegarem ao PDC, são passados para o método `C_PMU_PDC::Desmonta_dados()`, para a interpretação de seus valores.

Primeiramente, são analisados os valores do campo `STAT`, presente no *frame*. Este campo traz diversas informações, tais como *flag* indicativo de validade dos dados enviados pela PMU, *flag* indicativo de erro do equipamento, *flag* de sincronismo entre PMU e GPS, entre ou-

tros. Caso a medida esteja de acordo, o *frame* é encaminhado para o processamento dos dados.

A partir disso, o método passa a percorrer a estrutura do *frame*, cujo tamanho e ordem de leitura foram previamente fornecidos pelo *frame* de configuração, e os dados são extraídos e armazenados nos atributos da classe. Em seguida a este procedimento, os atributos da classe são acessados para a inserção dos dados no *buffer* em memória, por meio do método *C\_PMU\_PDC::Armazena\_buffer()*, e no *buffer* em HD, por meio do método *C\_PDC::Dados\_HD*.

### ***Frame de Comando***

O *frame* de comando tem por objetivo levar as requisições que precisam ser feitas pelo PDC para o equipamento da PMU. Como o sentido da comunicação, neste caso, é inverso, este *frame* é montado dentro do escopo da classe *C\_PMU\_PDC*, por meio do método *C\_PMU\_PDC::Envia\_comando()*. Os possíveis comandos, previstos em norma, são aqueles apresentados na seção 3.4.1.4.

### **4.3.4 Classe C\_PDC\_PDC**

Dado a adoção da arquitetura descentralizada do PDCS, o PDC compatível com esta estrutura deve estabelecer a troca de informações com demais equipamentos do sistema, que possuem funcionalidade de concentração de dados. Considerando-se que, os requisitos para estabelecer a comunicação entre equipamentos com função de PDC não diferem dos requisitos de comunicação entre PMUs e PDCs, a classe *C\_PDC\_PDC* é, conceitualmente, similar à classe *C\_PMU\_PDC*.

De maneira geral, a classe *C\_PDC\_PDC* representa um equipamento PDC remoto, que será contatado para troca de dados. Este PDC remoto pode ter tanto a função de *PDC Front End* quanto a função de PDC Mestre. Isto depende da função hierárquica do PDC representado pelo objeto da classe *C\_PDC*.

O desenvolvimento da estrutura desta classe, tendo em vista os conceitos da MOO, segue as mesmas premissas adotadas na classe *C\_PMU\_PDC*, agregando-se novas funcionalidades quando necessárias. Estas funcionalidades foram implementadas a partir da extrapolação de conceitos já utilizados e bem consolidados, alcançando abstração, reaproveitamento de código, modularidade e expansibilidade. O diagrama UML na Figura 4.7 apresenta os principais métodos e atributos desta classe.

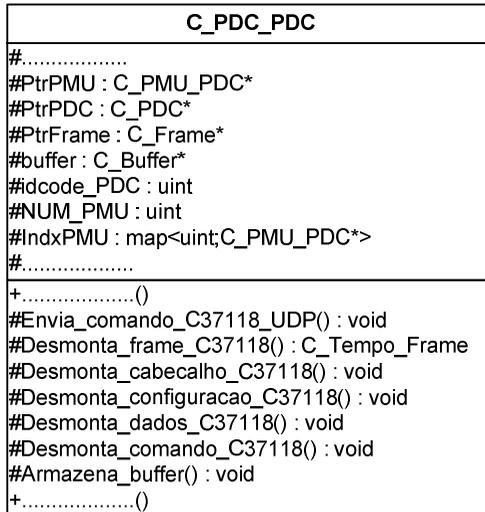


Figura 4.7 – Diagrama UML da Classe C\_PDC\_PDC.

Os principais atributos e métodos são:

- *PtrPMU*: ponteiro para objetos da classe C\_PMU\_PDC;
- *PtrPDC*: ponteiro para objeto da classe C\_PDC;
- *PtrFrame*: ponteiro para objeto da classe C\_Frame;
- *buffer*: ponteiro para objeto da classe C\_Buffer;
- *idcode*: código de identificação numérico do equipamento no SPMS;
- *NUM\_PMU*: número de PMUs contidas dentro do escopo do *frame* de configuração;
- *IndxPMU*: lista de referências de objetos da classe C\_PMU\_PDC
- *Envia\_comando\_C37118\_UDP*: método de envio de *frame* de comando da norma IEEE C37.118 via protocolo de transporte UDP;
- *Desmonta\_frame\_C37118()*: método de identificação do tipo de *frame* da norma IEEE C37118 recebido;
- *Desmonta\_cabecalho\_C37118()*: método de processamento das informações contidas no *frame* de cabeçalho;
- *Desmonta\_configuracao\_C37118()*: método de processamento das informações contidas no *frame* de configuração;
- *Desmonta\_dados\_C37118()*: método de processamento das informações contidas no *frame* de dados;
- *Desmonta\_comando\_C37118()*: método de processamento das informações contidas no *frame* de cabeçalho;

- *Armazena\_buffer()*: método de inserção dos dados no *buffer* de memória pertencente ao objeto da classe *C\_PDC\_PDC*.

As seções a seguir apresentam os aspectos conceituais da aplicabilidade da classe *C\_PDC\_PDC*, sobretudo no seu papel de representação do equipamento PDC remoto. Apresenta-se também a estrutura do *frame* de configuração utilizado para a comunicação entre PDCs.

#### **4.3.4.1 Representação do PDC remoto**

De maneira geral, a classe *C\_PDC\_PDC* representa um PDC remoto, isto é, um equipamento do SPMS que realiza a aquisição de dados fasoriais e que possui a habilidade de se comunicar com outro de mesma função. Entretanto, objetos da classe *C\_PDC\_PDC* executam diferentes ações que dependem do nível hierárquico do equipamento PDC dentro do PDCS.

No caso de PDC Mestre, os objetos instanciados da classe *C\_PDC\_PDC* modelam *PDCs Front End*, encarregados de enviar os dados das PMUs pelas quais ele é responsável de realizar a concentração de dados. Do ponto de vista do *PDC Front End*, o objeto da classe *C\_PDC\_PDC* representa um PDC Mestre. Quanto aos processos de execução internos ao *software* do PDC, pode-se dizer que as atribuições desta classe são análogas àquelas da classe *C\_PMU\_PDC*. Sendo assim, objetos da classe *C\_PDC\_PDC* processam todos os diferentes tipos de *frames* da norma C37.118, que são repassados pelo objeto da classe *C\_PDC*.

Os objetos instanciados desta classe contêm um *buffer* de dados em memória. Este *buffer* é acessado pelo método de re-sincronização de dados, cujos pacotes de dados disponibilizados subsidiam a montagem de um *frame* de dados IEEE C37.118. Este *frame* é encaminhado para o PDC Mestre ou para a lista de aplicações em tempo real, dependendo da função hierárquica do PDC.

Esta estrutura de *software* permite viabilizar a comunicação entre PDCs, garantindo o envio de dados das PMUs do SPMS ao PDC Mestre, intermediadas pelo processamento dos *PDCs Front End*, em tempo real e de forma contínua.

#### **4.3.4.2 Estrutura do *frame* de configuração da classe *C\_PDC\_PDC***

Como apresentado na descrição da classe *C\_PMU\_PDC*, toda vez que um *frame* de configuração é recebido pelo PDC, este é interpretado

e armazenado pelo respectivo objeto que corresponde ao equipamento de origem dos dados.

Contudo, no caso da classe *C\_PDC\_PDC*, a utilização do *frame* de configuração é feita de maneira diferenciada, dado que, neste caso, o *frame* de dados a ser enviado do *PDC Front End* para o PDC Mestre e do PDC Mestre para as aplicações em tempo real podem conter um número de PMUs maior do que um.

Esta possibilidade está prevista na norma, observando-se o parâmetro *NUM\_PMU* constante no *frame* de configuração, apresentado na Tabela 3.2. Este parâmetro designa quantas vezes os campos compreendidos no intervalo de 8 a 18 da referida tabela são repetidos, definindo a estrutura completa do *frame* de dados. Na classe *C\_PDC\_PDC*, o atributo correspondente a este campo é o *C\_PDC\_PDC::NUM\_PMU*. Seu valor depende do número de PMUs que enviam dados ao PDC, em um determinado momento. No caso de *PDC Front End*, o número máximo de PMUs possíveis em um *frame* de configuração é igual ao número total de PMUs gerenciadas por este PDC. No caso de PDC Mestre, este número máximo é igual ao total de PMUs contidos no SPMS.

#### 4.3.5 Classe *C\_Conexao*

De acordo os desenvolvimentos previstos para a implementação do projeto de *software* do PDC, observou-se a necessidade de se separar os processos que envolvem protocolos de rede daqueles que envolvem aquisição e processamento de dados.

Esta separação permite um tratamento isolado (encapsulado) de necessidades de comunicação de rede, comuns aos objetos *C\_PMU\_PDC* e *C\_PDC\_PDC*, haja vista que a troca de mensagens via rede é feita nos dois sentidos.

Optou-se pela representação dos parâmetros e operações do PDC, que envolvem canais de comunicação, em uma classe denominada *C\_Conexao*. Esta classe tem como principal atribuição gerenciar os protocolos de rede *ethernet*, que têm por função a função de canal de comunicação da estrutura do SPMS. A Figura 4.8 apresenta o diagrama UML da classe *C\_Conexao*.

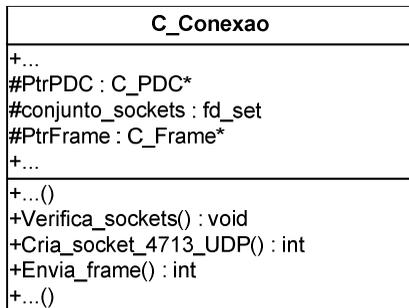


Figura 4.8 – Diagrama UML da Classe C\_Conexao.

Os principais métodos e atributos são:

- *PtrPDC*: ponteiro para o objeto da classe C\_PDC;
- *conjunto\_sockets*: conjunto de *sockets* monitorados pelo PDC;
- *PtrFrame*: ponteiro para objeto da classe C\_Frame;
- *Verifica\_sockets()*: método para verificação da chegada de *frames* na interface de rede do computador;
- *Cria\_socket\_4713\_UDP()*: método de criação do *socket* de comunicação;
- *Envia\_frame()*: método de envio de dados para equipamentos, via protocolos de rede.

Esta classe possui apenas um objeto instanciado durante a execução do processo do PDC. Durante o processo de aquisição de dados, todos os pacotes são gerenciados por este objeto, tanto para envio quanto para recebimento de dados.

Com objetivo de apresentar em maiores detalhes os desenvolvimentos aplicados nesta classe, as seções a seguir fazem uma incursão sobre os conceitos utilizados dos protocolos de rede *ethernet*, bem como os aspectos de funcionalidade levados em consideração.

#### 4.3.5.1 Protocolos de transporte

Os protocolos de rede utilizados no desenvolvimento da classe C\_Conexao são fornecidos pela Interface de Programação de Aplicação (API – *Application Program Interface*) *socket.h*. Esta biblioteca contém os métodos que possibilitam o *software* do PDC acionar a interface de rede do computador, a fim de criar um caminho para a troca de dados entre os equipamentos do SPMS.

As ferramentas disponibilizadas pela biblioteca são utilizadas dentro da classe C\_Conexao, onde foram desenvolvidos métodos especí-

ficos para atender as necessidades do PDC. Estes métodos procuram ter ampla abrangência, sem distinguir o tipo do objeto que está lhe acionando.

A API *socket.h* é desenvolvida para enviar e receber pacotes de dados em redes de computadores através de protocolos de transporte UDP/IP (*User Datagram Protocol/Internet Protocol*) e TCP/IP (*Transmission Control Protocol/Internet Protocol*).

O protocolo UDP/IP é um protocolo de transporte simplificado, tendo por característica a ausência de conexão entre os aplicativos cliente e servidor, aferindo à transmissão de dados a característica de baixa latência (tempo de envio de dados) em rede. Conseqüentemente, pacotes de dados UDP podem ser perdidos ao longo da transmissão, sem possibilidade de recuperação, o que prejudica sua confiabilidade.

Por outro lado, mensagens que utilizam o protocolo TCP/IP recebem diversos benefícios do ponto de vista de confiabilidade. Isto se deve pelo fato do estabelecimento efetivo de uma conexão entre processo cliente e servidor, garantindo a entrega dos pacotes ao destino e, no caso de perda de dados, existe a possibilidade de solicitar a retransmissão dos mesmos. A descrição detalhada das características e funcionalidades dos protocolos UDP e TCP está disponível em (STEVENS, W. R., *et al.*, 2004).

Considerando o escopo de SPMS, dado a alta taxa de amostragem das medidas coletadas no sistema, o volume de dados a ser transmitido é significativo. Desta forma abre-se mão da confiabilidade da entrega de dados ao PDC em detrimento da menor latência na transmissão dos sincrofasores, uma vez que, dependendo das aplicações (por exemplo, controle e/ou proteção), esta pode ser crítica e deve ser minimizada. Unindo estes requisitos com as características dos protocolos de transporte, observa-se a aplicabilidade do protocolo UDP/IP no envio/recebimento de fasores dentro do SPMS.

A classe *C\_Conexao* foi desenvolvida contemplando a utilização do protocolo UDP/IP. Entretanto sua estrutura permite a inserção das funcionalidades aplicadas ao protocolo TCP/IP, caso necessário. Esta característica permite que o código fonte seja expandido sem necessidade de adaptações nos desenvolvimentos já existentes.

#### **4.3.5.2 Gerenciamento dos dados de rede**

Por definição, a classe *C\_Conexao* não processa os formatos de mensagens da norma IEEE C37.118. O tratamento destes formatos é deixado para os métodos das classes *C\_PMU\_PDC* e *C\_PDC\_PDC*, de acordo com as designações da classe *C\_PDC*. Por sua vez, os objetos

destas classes desconhecem o tratamento dos dados dos protocolos de rede. Por definição, as atribuições da classe *C\_Conexao* limitam-se a realizar a interface entre rede e *software*, disponibilizando os dados para interpretação e tratamento dos métodos das demais classes do PDC.

Para efetuar início dos protocolos de rede do PDC, é necessária a criação do seu respectivo *socket*. O *socket* é a ponte de comunicação entre o sistema operacional do computador com sua interface de rede, e é por onde passam todos os fluxos de dados trocados com os equipamentos do SPMS. São atrelados ao *socket* os seguintes parâmetros: protocolo de transporte (TCP/UDP), endereço IP, porta de comunicação. Este procedimento é realizado pela classe *C\_PDC*, acionando-se o método *C\_Conexao::Cria\_socket\_4713\_UDP()*.

Considerando-se o recebimento de dados pela classe *C\_Conexao*, esta é feita por meio do método *C\_Conexao::Verifica\_sockets()*. A cada ciclo de varredura de dados, este método é acionado pelo objeto da classe *C\_PDC* e, como resultado, uma lista de *frames* é disponibilizada. Esta lista contém elementos que são referências para objetos da classe *C\_Frame*. Cada referência é equivalente a um pacote de dados, recebidos no *socket* da interface de rede do PDC. Estes dados são interpretados no escopo da classe *C\_PDC*.

Para a atribuição de envio de dados via os protocolos de rede, um método desenvolvido apenas com essa funcionalidade foi elaborado, denominado *C\_Conexao::Envia\_frame()*. Este método contém em seus argumentos os parâmetros necessários para viabilizar o envio de dados, sendo eles: pacote de dados em memória (independente de formato), tamanho do pacote, endereço IP de destino e protocolo de transporte (TCP/UDP). Sendo assim, independente do objeto que executa a chamada do método, basta fornecer os parâmetros dentro da forma especificada, que os dados são enviados ao destino especificado.

#### **4.3.6 Classe *C\_Frame***

A classe *C\_Frame* visa modelar os *frames* utilizados na comunicação entre equipamentos do sistema. Como apresentado no Capítulo 2, nos desenvolvimentos iniciais da tecnologia de SPMS, diferentes tipos de *frames* foram adotados para troca de dados entre equipamentos, dado que no período não haviam normas que definissem padronizações de formatos.

Embora, atualmente os formatos dos *frames*, incluindo seus tipos, são ditados pela norma IEEE C37.118, é razoável admitir que revisões e/ou novos documentos possam vir a suceder tais regras. Considerando este aspecto, para o projeto de *software* do PDC, optou-se pelo desen-

volvimento de uma classe exclusiva para representar *frames* aplicados à SPMS. Caso novos *frames* venham fazer parte dos requisitos de comunicação entre equipamentos de SPMS, é possível realizar a expansão do código-fonte, adequando o projeto aos novos formatos de *frames*, sem necessidade de alterações nos desenvolvimentos já realizados. A Figura 4.9 apresenta o diagrama UML da classe *C\_Frame*.

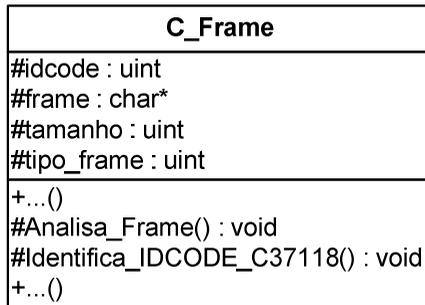


Figura 4.9 – Diagrama UML da Classe *C\_Frame*..

Os principais atributos e métodos são:

- *idcode*: código identificador do equipamento que originou o *frame*;
- *frame*: ponteiro para posição de memória em que está armazenada a seqüência de dados do *frame*;
- *tamanho*: tamanho, em *bytes*, do *frame*;
- *tipo\_frame*: identificação do tipo de *frame*. Seu valor padrão é igual a 1, para identificar *frames* no formato IEEE C37.118;
- *Analisa\_Frame()*: método de análise do *frame*, responsável pela sua identificação, bem como a extração das informações que são guardadas nos atributos da classe;
- *Identifica\_IDCODE\_C37118()*: método de extração do parâmetro *IDCODE*, necessário para identificar a origem do *frame*.

Durante a execução do processo do PDC, a cada ciclo de varredura de dados, são criadas *n* instâncias desta classe, onde *n* é equivalente ao número de *frames* que chegam na interface de rede do computador.

No momento em que uma instância da classe *C\_Frame* é realizada, o processo de identificação do *frame* é iniciado, estabelecendo seu tipo. Para os atuais desenvolvimentos, este tipo pode ser apenas referen-

te ao formato IEEE C37.118. Assim que confirmado seu formato, o IDCODE é extraído e armazenado no atributo *C\_FRAME::idcode*.

Os objetos instanciados são referenciados no escopo da classe *C\_PDC*, por meio do atributo *C\_PDC::IdxFrame*. Este atributo é uma lista com ponteiros aos objetos da classe *C\_Frame*. É de responsabilidade da classe *C\_PDC* acessar esta lista e encaminhar os dados contidos nos *frames* para processamento. Após este procedimento, a lista é apagada em conjunto com as instâncias da classe *C\_Frame*, realizadas para o ciclo de varredura de dados corrente. Desta forma, garante-se que a memória alocada para este processo seja liberada para o próximo ciclo.

#### **4.3.7 Classe C\_App**

No escopo dos desenvolvimentos do PDC MedFasee, a disponibilização de dados para aplicações em tempo real é feita por meio de tabelas em banco de dados MySQL, atualizadas a cada segundo. A aplicação que desejasse fazer uso destes dados deveria acessar o banco de dados e solicitar explicitamente tais valores.

Embora a solução acima apresentada fosse válida, dois aspectos devem ser considerados, tais como: requisitos de latência e amostragem dos dados.

Considerando a latência, percebe-se que, por mais rápido que seja feito o processamento dos dados, existe um atraso inerente ao processo de inserção dos valores na tabela de tempo real e a requisição dos dados pelo lado das aplicações. Sobre a amostragem dos dados, como a tabela de tempo real da base dados do PDC possui atualizações a cada segundo, todos os outros pacotes de dados re-sincronizados disponibilizados durante este segundo são descartados. Desta forma, observa-se a necessidade de modificar o mecanismo de disponibilização de dados em tempo real do PDC. Para isso foi desenvolvida a classe *C\_App*.

Esta classe modela as aplicações computacionais que fazem uso dos dados em tempo real do PDC. Esta abordagem é semelhante àquela utilizada nas classes *C\_PMU\_PDC* e *C\_PDC\_PDC*. A diferença é que esta representação se refere a um processo interno do PDC, e não a equipamentos físicos. A Figura 4.10 abaixo apresenta a estrutura da classe *C\_App*.

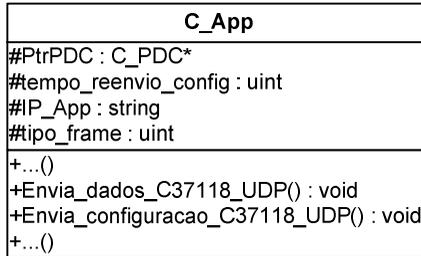


Figura 4.10 – Diagrama UML da Classe C\_App.

Os principais atributos e métodos são:

- *PtrPDC*: ponteiro para o objeto da classe *C\_PDC*;
- *IP\_App*: endereço IP do equipamento que faz uso dos dados em tempo real do PDC;
- *tipo\_frame*: variável que armazena o tipo de *frame* que é enviado. Seu valor é igual a 1, no caso de envio de dados no formato C37.118;
- *tempo\_reenvio\_config*: tempo, em segundos, para o reenvio de *frame* de configuração para o equipamento que faz uso dos dados em tempo real do PDC;
- *Envia\_dados\_C37118\_UDP()*: método para envio de *frame* de dados no formato C37.118, utilizando protocolo UDP;
- *Envia\_configuracao\_C37118\_UDP()*: método de envio de *frame* de configuração no formato C37.118, utilizando protocolo UDP.

Existem  $n$  instâncias desta classe, onde  $n$  é igual ao número de aplicações que fazem uso dos dados em tempo real do PDC. Os valores que preenchem os atributos *IP\_App*, *tipo\_frame* e *tempo\_reenvio\_config* são informados no arquivo de configuração XML do PDC.

A utilização de um atributo específico para identificar o tipo de *frame* tem por justificativa a expansibilidade do código fonte. Caso novos formatos de mensagens venham a existir, classe pode ser rapidamente adaptada, sem necessidade de modificação dos desenvolvimentos já realizados.

Para que a aplicação possa interpretar os dados em tempo real enviados pelo PDC, é necessário o envio de *frames* de configuração, para que o *frame* de dados seja interpretado corretamente. Devido ao fato de o PDC não receber comandos externos, o envio dos *frames* de configuração é realizado de maneira sistemática. A cada intervalo de

*tempo\_reenvio\_config*, um novo *frame* de configuração, contendo os parâmetros mais atuais considerados pelo PDC, é enviado ao endereço IP da aplicação. Desta forma, um fluxo de dados em tempo real é enviado continuamente para as aplicações, dispensando a necessidade de solicitação de informações ao PDC.

#### **4.4 Comentários Finais**

Este capítulo apresentou em detalhes os aspectos mais relevantes adotados na filosofia do projeto de *software* desenvolvido neste trabalho, cujo objetivo é viabilizar a comunicação entre equipamentos com função de PDC. Esta comunicação é característica fundamental na implementação do PDCS.

Utilizou-se para o desenvolvimento do projeto conceitos da MO-O, buscando, entre outros aspectos, o reaproveitamento do código-fonte, abstração (representação) dos elementos físicos do SPMS e de características funcionais do PDC por meio de classes e garantir o encapsulamento de dados, que permite a escrita de um código fonte organizado e expansível. Durante os desenvolvimentos, procurou-se manter a característica de expansibilidade do código fonte. Este paradigma possibilita a expansão dos desenvolvimentos para novos requisitos, que possam vir a fazer parte da tecnologia de SPMS.

As classes desenvolvidas no projeto representam equipamentos físicos do SPMS (classes *C\_PMU\_PDC* e *C\_PDC\_PDC*) e características funcionais que fazem parte do PDC (classes *C\_Conexao*, *C\_Frame*, *C\_App*). Ao longo do capítulo, foram apresentadas as suas principais características e funcionalidades, apontando aspectos centrais do desenvolvimento das classes.

O aspecto de disponibilização de dados em tempo real foi apresentado em detalhes, dado que esta característica é fundamental para a viabilização do PDCS. A estrutura do *buffer* em memória das classes *C\_PMU\_PDC* e *C\_PDC\_PDC* foi apresentada. O método de resincronização de dados foi discutido aprofundadamente, de forma a estabelecer a forma em que os dados em tempo real são disponibilizados pelo PDC.

# CAPÍTULO 5

## 5. Experimentos e Resultados

Este capítulo tem por objetivo apresentar os experimentos e resultados da operação do PDCS. Primeiramente, são apresentadas configurações montadas em bancada de testes, utilizando a estrutura do Projeto MedFasee Frequência de forma a validar a arquitetura do PDCS apresentada no Capítulo 3.

Em seguida, é realizada uma análise da operação do PDCS, considerando as perdas ocorridas na transmissão de dados. A disponibilização de dados em Tempo Real também é avaliada, sendo para isso utilizado *software* de teste dedicado a validação de dados enviados por equipamentos compatíveis com a tecnologia de medição fasorial sincronizada.

Análises são realizadas do comportamento do sistema elétrico utilizando dados disponibilizados pela base dados histórica dos PDCs. As bases são comparadas, em um primeiro momento, entre os equipamentos, com objetivo de verificar a aderência dos dados armazenados nas bases de dados locais (*PDCs Front End*) e na base de dados central (PDC Mestre). Em seguida, é avaliada uma ocorrência no Sistema Interligado Nacional capturada pelas PMUs do Projeto MedFasee Frequência, servindo como referência para estabelecer se os dados armazenados no banco de dados do PDC Mestre reproduzem, de maneira coerente, o comportamento dinâmico do sistema elétrico.

### 5.1 Protótipos do PDCS

Esta seção apresentará os protótipos que foram instalados em laboratório, com duas estruturas distintas, tendo por objetivo validar a estrutura hierárquica de aquisição de dados proposta para este trabalho. Estas estruturas procuram também exemplificar a expansibilidade do projeto, característica principal da proposta do sistema de concentração de dados proposto.

### 5.1.1 Protótipo 1 - PDCS utilizando 2 PDCs Front End e PDC Mestre

O protótipo apresentado na Figura 5.1 ilustra um diagrama esquemático da implementação prática do sistema proposto para este trabalho. As PMUs utilizadas nesta configuração são acessadas do protótipo de SPMS do Projeto MedFasee Frequência (MEDFREQ, 2008).

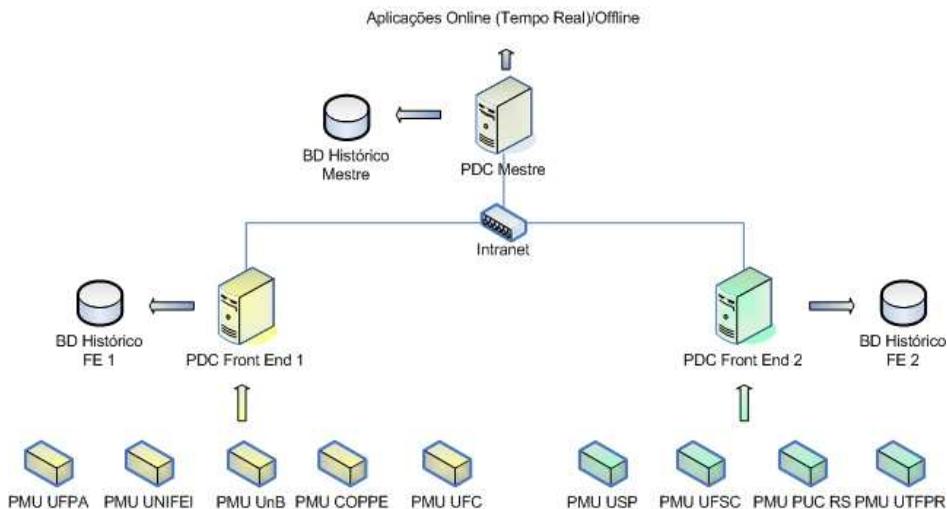


Figura 5.1 – Protótipo de PDCS com 2 PDCs Front End e PDC Mestre.

O protótipo de SPMS do Projeto MedFasee tem papel fundamental para o desenvolvimento desta proposta, sendo utilizado como plataforma de validação dos conceitos utilizados para a comunicação entre PDCs, elaboradas para este trabalho. A estrutura implementada está nos padrões do PDCS apresentado na Figura 3.5, considerando  $n$  igual a 2.

Observa-se é a presença de bancos de dados históricos nos PDCs Front End. A utilização de bases de dados locais traz o benefício da redundância no armazenamento dos sincrofases e, conseqüentemente, aumenta a confiabilidade do sistema de aquisição de dados.

Os equipamentos utilizados como PDCs para o protótipo exposto se tratam de microcomputadores DELL Power Edge R410, contendo as seguintes características:

- Processador: Intel Xeon Quad E5506 64bits, 2,13 GHz, 4 Mb Cache

- Memória RAM de 2GB
- Dois discos rígidos de 250Gb, operando em RAID 1 (espelhamento)
- Sistema Operacional GNU/Linux CentOS 5.3 64bits

As PMUs do projeto são do modelo RPV 304 da fabricante *Reason Tecnologia S/A*, sendo a comunicação entre os equipamentos realizado por meio de conexão VPN (*Virtual Private Network*), estabelecida pelos firewalls instalados no LabPlan/UFSC e em cada ponto de instalação das PMUs remotas, sobre a rede Internet.

A Figura 5.2 ilustra a bancada de testes montada dentro do Laboratório de Planejamento de Sistemas de Energia Elétrica – LabPlan, da Universidade Federal de Santa Catarina.



Figura 5.2 – Bancada de Testes do PDCS.

### 5.1.2 Protótipo 2 – PDCS utilizando 3 PDCs Front End e PDC Mestre

Com intuito de validar a escalabilidade do PDCS, considerando maior número de *PDCs Front End* e, ao mesmo tempo, verificar sua compatibilidade com sistemas desenvolvidos independentemente ao Projeto MedFasee, adicionou-se à estrutura apresentada na seção anterior, um terceiro *PDC Front End*, cujo equipamento executa o *software* de aquisição de dados fasoriais desenvolvido pela TVA, o *openPDC*. A

Figura 5.3 ilustra a estrutura expandida, contendo três equipamentos com função de *PDC Front End* e um no papel de *PDC Mestre*. Assim como no caso anterior, todos os *PDCs Front End* possuem base de dados local.

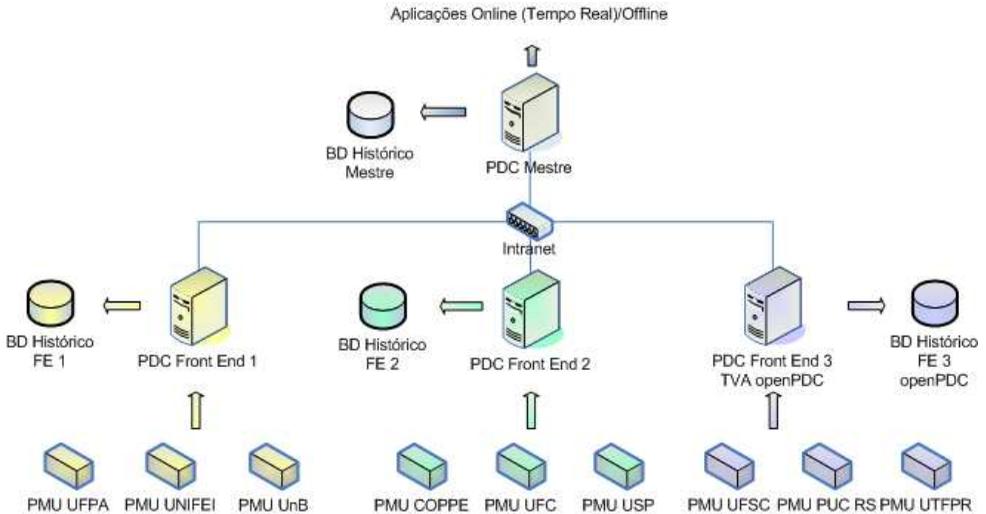


Figura 5.3 – Protótipo de PDCS com 3 PDCs Front End e PDC Mestre.

O equipamento utilizado para o *openPDC* é ilustrado pela Figura 5.4. Sua configuração de *hardware* é a seguinte:

- Processador: Intel Core 2 Duo E68550 64bits, 3,0 GHz, 4 Mb Cache
- Memória RAM de 2GB
- Dois discos rígidos de 250Gb, operando em RAID 1 (espelhamento)
- Sistema Operacional Windows Server 2003



Figura 5.4 – Equipamento executando o processo do openPDC.

### 5.1.3 Capacidade dos PDCS

Durante o processo de implementação dos protótipos de PDCS, duas grandezas devem ser observadas visando compatibilizar o uso do sistema com suas capacidades. A primeira é a largura de banda utilizada pelos PDCs do PDCS e a segunda é a capacidade de armazenamento da base de dados histórica.

#### 5.1.3.1 Largura de Banda

Considerando-se os requisitos de largura de banda, a Tabela 5.1 ilustra a banda necessária para um *frame* de dados.

Tabela 5.1 – Demanda de largura de banda do *frame* de dados.

<i>Campo</i>	<i>Tamanho Bytes</i>
SYNC	2
FRAME_SIZE	2
IDCODE	2
SOC	4
FRAC_SEC	4
STAT	2
PHASORS	4 / 8 * N° CANAIS FASORIAIS
FREQ	2 / 4
DFREQ	2 / 4
ANALOG	3 / 4 * N° CANAIS ANALÓGICOS
DIGITAL	2 * N° CANAIS DIGITAIS
CHK	2

O tamanho do *frame* depende de três fatores. O primeiro é a precisão das grandezas fasoriais e analógicas, que podem ser tanto valores inteiros quanto valores em ponto flutuante. O segundo fator é o número de canais fasoriais, analógicos e digitais. Por último, o *frame* de dados pode conter valores de mais de uma PMU, sendo que seu tamanho é proporcional a este parâmetro. Considerando-se uma PMU do projeto MedFasee, que utiliza 3 canais fasoriais com precisão em ponto flutuante, tem seu *frame* de dados com tamanho de 50 bytes. Tendo em vista que os dados são enviados por protocolo UDP/IP, são adicionados ao pacote de dados mais 54 bytes (26 referente a camada Ethernet II, 20 bytes referentes a camada IP e 8 bytes do protocolo UDP), totalizam-se 104 bytes para um *frame* de dados. Admitindo a taxa de amostragem do projeto MedFasee, de 60 sincrofasores por segundo, o requisito de largura de banda é igual a 49,92 kbps. A Tabela 5.2 ilustra a evolução da demanda por largura de banda quando novas PMUs (com características iguais as do Projeto MedFasee) são adicionadas a um *PDC Front End*.

*Tabela 5.2 – Evolução do requisito de largura de banda para a aquisição de dados pelo PDC.*

<i>Nº de PMUs</i>	<i>Largura de Banda [kbps]</i>
1	49,92
2	99,84
3	149,76
4	199,68
5	249,60
6	299,52
7	349,44
8	399,36
9	449,28
10	499,20

Do ponto de vista do PDC Mestre, existe um ganho em escala da utilização da banda de rede, devido ao fato que o *frame* de dado enviado pelo *PDC Front End* contém todas as PMUs pelo qual é responsável. Desta forma, compartilham-se os campos fixos contidos no *frame* de dados. A Tabela 5.3 ilustra a largura de banda demanda pelo PDC Mestre no contexto dos dois protótipos de PDCS implementados.

Tabela 5.3 – Largura de banda requisitada pelo PDC Mestre.

<b>Largura de Banda PDC Mestre</b>			
Protótipo 1	Largura de Banda [kbps]	Protótipo 2	Largura de Banda [kbps]
PDC Front End 1	115,20	PDC Front End 1	82,56
PDC Front End 2	98,8	PDC Front End 2	82,56
		PDC Front End 3	82,56
<b>Total</b>	214,00	<b>Total</b>	247,68

Observa-se que, se fosse utilizado apenas um PDC para receber os dados de todas as PMUs do Projeto MedFasee, a banda necessária seria de 449,28 kbps. Com a presença dos *PDCs Front End*, é notório o ganho relativo ao compartilhamento dos campos fixos do *frame* de dados, pois existe uma redução da ordem de 52% da utilização da rede, para o caso do Protótipo 1 de PDCS.

### 5.1.3.2 Armazenamento em base de dados histórica

Considerando-se a base de dados histórica, são armazenados em suas tabelas os seguintes parâmetros recebidos dos *frames* de dados: IDCODE (2 bytes), número do canal (1 byte), etiqueta de tempo (SOC - 4 bytes; FRACSEC - 1 bytes), fasor (módulo - 4 bytes; ângulo - 4 bytes). Desta forma, o registro de um canal fasorial de uma PMU equivale a 16 bytes de registro no banco de dados histórico. O requisito de tamanho de disco rígido depende da quantidade de dias de histórico desejado. A Tabela 5.4 ilustra a necessidade de capacidade de disco rígido necessária em função dos dias de histórico requeridos. Estes dados consideram as PMUs do Projeto MedFasee, enviando dados a 60 sincrofases por segundo, referentes a três canais fasoriais em cada equipamento.

Considerando o caso do *hardware* utilizado para os PDCs do PDCS, cujo disco rígido tem capacidade de 250 Gb, é possível ter uma base histórica de até 111 dias.

*Tabela 5.4 – Capacidade de disco rígido necessária para armazenamento dos dados em banco de dados histórico.*

<b>Tamanho em disco necessário para o banco de dados histórico</b>		
<b>Nº de PMUs</b>	<b>Dias de Hist</b>	<b>Tamanho em Gbytes</b>
9	1	2,24
Bytes/s inseridos na base de dados	2	4,48
	3	6,72
25920	4	8,96
	5	11,20
	6	13,44
	7	15,68
	8	17,92
	9	20,16
	10	22,39

## 5.2 Análise das Perdas de Dados

Tendo em vista da utilização do protocolo UDP para o transporte de dados pela rede de comunicação, que conecta os equipamentos do SPMS, é necessário estabelecer o volume de perdas de sincrofasores que ocorre durante o período de funcionamento do PDC.

Para tanto, foi utilizado o programa MedPerdas, *software* desenvolvido no âmbito do projeto MedFasee, o qual realiza buscas diárias seqüenciais na base de dados de cada PDC, com discretização horária, analisando cada PMU individualmente. Considerando-se uma taxa de 60 fasores por segundo, durante uma hora são esperados 216.000 fasores de cada PMU. A partir do número de fasores existentes no banco de dados, o programa calcula e apresenta a porcentagem de perda de dados de cada PMU, a cada hora. Foram verificados três casos de comportamento das perdas no funcionamento do PDCS, que são descritos a seguir.

### 5.2.1 Perdas Nulas

Considerando o caso da PMU UFSC, as perdas verificadas durante os testes foram nulas, tanto no *PDC Front End 2*, quanto no PDC Mestre. Este resultado é esperado, uma vez que a PMU UFSC está conectada diretamente na rede local do LabPlan/UFSC, juntamente com os *PDCs Front End* e Mestre durante os testes. Desta forma, não existem perdas de dados devido à rede de comunicação, conforme já mostrado em resultados anteriores de avaliação de perdas de dados, realizadas no

âmbito do Projeto MedFasee. Na Tabela 5.5 ilustram-se as avaliações de perdas da PMU UFSC durante o dia 01/06/2010, tanto no *PDC Front End 2* quanto no PDC Mestre.

### **5.2.2 Perdas Devido aos Canais de Comunicação – Internet**

No caso das demais PMUs, pequenas porcentagens de perdas de dados foram verificadas ao longo dos dias, devido à rede de comunicação entre as PMUs, instaladas nas respectivas universidades, e o PDCS, instalado no LabPlan/UFSC.

Conforme descrito na seção 5.1.1, a comunicação entre os equipamentos se dá através de uma conexão VPN, estabelecida pelos firewalls instalados no LabPlan/UFSC e em cada ponto de instalação das PMUs remotas, sobre a rede Internet. Os dados são enviados pelas PMUs usando protocolo UDP/IP e, portanto, perdas de dados são esperadas de acordo com o nível de carregamento da rede, tanto da Internet entre as universidades, quanto em suas respectivas redes internas.

Na Tabela 5.6 ilustram-se as avaliações de perdas de dados da PMU UnB durante o dia 29/05/2010 (sábado), em ambos os PDCs. Observam-se pequenas porcentagens de perdas ao longo de todo o dia, com um pico no período das 17h (UTC) (14h no horário local de Brasília).

Porém observa-se que não foram registradas perdas de dados entre os *PDCs Front End* e Mestre, ou seja, mesmo em um regime de operação com algumas falhas na chegada de dados ao *PDC Front End*, a comunicação entre *PDCs Front End* e Mestre funcionou de maneira correta.

*Tabela 5.5 – Resultados do MedPerdas durante as 24 horas do dia 01/06/2010 para a PMU UFSC.*

<b>PDC Front End 2 - PMU em Análise -&gt; UFSC</b>			<b>PDC Mestre - PMU em Análise -&gt; UFSC</b>		
<b>Tabela: 20100601</b>			<b>Tabela: 20100601</b>		
<b>Hora</b>	<b>Dados recebidos</b>	<b>Perdas (%)</b>	<b>Hora</b>	<b>Dados recebidos</b>	<b>Perdas (%)</b>
0h	216000	0	0h	216000	0
1h	216000	0	1h	216000	0
2h	216000	0	2h	216000	0
3h	216000	0	3h	216000	0
4h	216000	0	4h	216000	0
5h	216000	0	5h	216000	0
6h	216000	0	6h	216000	0
7h	216000	0	7h	216000	0
8h	216000	0	8h	216000	0
9h	216000	0	9h	216000	0
10h	216000	0	10h	216000	0
11h	216000	0	11h	216000	0
12h	216000	0	12h	216000	0
13h	216000	0	13h	216000	0
14h	216000	0	14h	216000	0
15h	216000	0	15h	216000	0
16h	216000	0	16h	216000	0
17h	216000	0	17h	216000	0
18h	216000	0	18h	216000	0
19h	216000	0	19h	216000	0
20h	216000	0	20h	216000	0
21h	216000	0	21h	216000	0
22h	216000	0	22h	216000	0
23h	216000	0	23h	216000	0

Tabela 5.6 - Resultados do MedPerdas durante as 24 horas do dia 29/05/2010 para a PMU UnB.

PDC Front End 1 - PMU em Análise -> UnB			PDC Mestre - PMU em Análise -> UnB		
Tabela: 20100529			Tabela: 20100529		
Hora	Dados recebidos	Perdas (%)	Hora	Dados recebidos	Perdas (%)
0h	216000	0,0000000	0h	216000	0,0000000
1h	215997	0,0013889	1h	215997	0,0013889
2h	215994	0,0027778	2h	215994	0,0027778
3h	215930	0,0324074	3h	215930	0,0324074
4h	215972	0,0129630	4h	215972	0,0129630
5h	216000	0,0000000	5h	216000	0,0000000
6h	215279	0,3337963	6h	215279	0,3337963
7h	215525	0,2199074	7h	215525	0,2199074
8h	214723	0,5912037	8h	214723	0,5912037
9h	215948	0,0240741	9h	215948	0,0240741
10h	215850	0,0694444	10h	215850	0,0694444
11h	215946	0,0250000	11h	215946	0,0250000
12h	215999	0,0004630	12h	215999	0,0004630
13h	215997	0,0013889	13h	215997	0,0013889
14h	215994	0,0027778	14h	215994	0,0027778
15h	215844	0,0722222	15h	215844	0,0722222
16h	215957	0,0199074	16h	215957	0,0199074
17h	212832	1,4666667	17h	212832	1,4666667
18h	215797	0,0939815	18h	215797	0,0939815
19h	215288	0,3296296	19h	215288	0,3296296
20h	214559	0,6671296	20h	214559	0,6671296
21h	215194	0,3731481	21h	215194	0,3731481
22h	214882	0,5175926	22h	214882	0,5175926
23h	215571	0,1986111	23h	215571	0,1986111

### 5.2.3 Perdas de Dados Entre *PDCs Front End* e *PDC Mestre*

Foram observados períodos de operação do PDCS em que pequenas perdas de dados ocorrem entre os *PDCs Front End* e o *PDC Mestre*. Analisando a Tabela 5.7, onde são apresentadas as perdas de dados da PMU UNIFEI durante o dia 28/05/2010, é possível observar, durante os períodos das 2h, 7h, 8h, 12h e 23h, perdas de dados um pouco maiores no *PDC mestre* em relação ao *PDC frontend* que recebia os dados desta PMU.

A ocorrência de perdas de dados entre um *PDC Front End* específico e o *PDC Mestre* está relacionada a períodos em que ocorrem perdas de dados simultâneas de todas as PMUs daquele *PDC Front End*, durante um período maior que o atraso máximo configurado para este

*PDC Front End*. Isto pode acontecer, por exemplo, em períodos momentâneos de falha ou sobrecarga da conexão entre a rede local onde está conectado o tal PDC e a rede Internet. Nestes casos, o *PDC Front End* permanece um determinado período de tempo sem enviar dados ao PDC Mestre, pois dados novos não são recebidos no *PDC Front End*, e, portanto, seu *tempo-base* não é atualizado, bloqueando temporariamente a disponibilização de pacotes de dados em tempo real.

Ao chegar o primeiro frame de dados válido após o "período de silêncio", o *tempo-base* do *PDC Front End* é atualizado, e o processo de re-sincronização de dados em tempo real é retomado. Caso o período de silêncio tenha sido maior que o limite definido pelo atraso máximo configurado no *PDC Front End*, os últimos dados das PMUs que chegaram à este PDC, pouco antes do período de silêncio, e que estavam armazenados no seu *buffer* em memória, são automaticamente descartados.

Observe-se, porém, que os dados descartados do envio para o PDC Mestre já foram devidamente armazenados no banco de dados local do *PDC Front End*, existindo em sua base dados local, mas não na base de dados do PDC Mestre. Já no caso de períodos de silêncio pequenos, ou seja, períodos de perda de dados de todas as PMUs menores que o atraso máximo, o *PDC Front End* consegue retomar sua operação normal, utilizando todos os dados existentes no *buffer*, sem descartes.

Ressalta-se que foram registrados poucas ocorrências do caso descrito acima, todas elas associadas a interrupções da conexão da rede do LabPlan/UFSC ao NPD (Núcleo de Processamento de Dados da UFSC), ou mesmo falhas temporárias da conexão da rede UFSC com a Internet.

Para a operação do PDCS em campo, deve ser escolhido um atraso máximo compatível com a rede a ser utilizada para a conexão entre os equipamentos do SPMS, considerando-se ainda a tolerância a atrasos das aplicações que farão uso dos sincrofasores.

Tabela 5.7 – Resultados do MedPerdas durante as 24 horas do dia 28/05/2010 para a PMU UNIFEI.

PDC Front End 1 - PMU em Análise -> UNIFEI			PDC Mestre - PMU em Análise -> UNIFEI			Diferença (Dados PDC FE 1 - Dados PDC Mestre)
Tabela: 20100528			Tabela: 20100528			
Hora	Dados recebidos	Perdas (%)	Hora	Dados recebidos	Perdas (%)	
0h	215091	0,4208333	0h	215091	0,4208333	0
1h	214714	0,5953704	1h	214714	0,5953704	0
2h	214736	0,5851852	2h	214607	0,6449074	129
3h	215565	0,2013889	3h	215565	0,2013889	0
4h	205260	4,9722222	4h	205260	4,9722222	0
5h	215584	0,1925926	5h	215584	0,1925926	0
6h	215595	0,1875000	6h	215595	0,1875000	0
7h	212912	1,4296296	7h	212734	1,5120370	178
8h	215084	0,4240741	8h	215007	0,4597222	77
9h	214483	0,7023148	9h	214483	0,7023148	0
10h	215993	0,0032407	10h	215993	0,0032407	0
11h	215341	0,3050926	11h	215341	0,3050926	0
12h	210695	2,4560185	12h	210493	2,5495370	202
13h	215798	0,0935185	13h	215798	0,0935185	0
14h	215259	0,3430556	14h	215259	0,3430556	0
15h	215258	0,3435185	15h	215258	0,3435185	0
16h	209986	2,7842593	16h	209986	2,7842593	0
17h	215677	0,1495370	17h	215677	0,1495370	0
18h	215596	0,1870370	18h	215596	0,1870370	0
19h	215735	0,1226852	19h	215735	0,1226852	0
20h	215406	0,2750000	20h	215406	0,2750000	0
21h	214757	0,5754630	21h	214757	0,5754630	0
22h	215685	0,1458333	22h	215685	0,1458333	0
23h	215538	0,2138889	23h	215240	0,3518519	298

### 5.3 Validação da estrutura de dados em Tempo Real

Como apresentado na seção 4.3.2.3, a estrutura de dados em tempo real do PDC funciona por meio de um *buffer* de dados em tempo real. Os dados presentes neste *buffer* são acessados e referenciados a uma mesma etiqueta de tempo. Assim que o método de re-sincronização de dados disponibiliza um pacote de dados, o PDC se encarrega de formatá-lo em um *frame* de dados no IEEE C37.118, repassando-o para as aplicações em tempo real ou PDC Mestre, dependendo da funcionalidade do equipamento do PDC.

Com objetivo de validar esta estrutura, testes foram realizados utilizando o *software TVA PMU Connection Tester*, desenvolvido pela TVA (*Tennessee Valley Authority*), cujo objetivo é subsidiar os estudos da tecnologia de sincrofasores no âmbito do NASPI. Este *software* tem por objetivo verificar se os dados enviados de qualquer dispositivo estão

sendo recebido, em diversos formatos de mensagens (IEEE C37.118, BPA *PDCstream*, IEEE 1344).

Para que os dados sejam enviados para a aplicação, foi adicionado o endereço IP do computador à lista de aplicações do arquivo de Configuração XML, de forma que o fluxo de dados possa ser direcionado corretamente. Desta forma, um objeto da classe *C\_App* será instanciado a fim de representar esta aplicação remota que faz uso dos dados em tempo real do PDC. Na Figura 5.5 ilustra-se a interface gráfica do programa em tempo de execução, a fim de ilustrar o recebimento dos dados enviados pelo PDC Mestre.

Este teste indica o correto funcionamento do PDC, tanto do ponto de vista de funcionamento da estrutura de dados em tempo real, por meio de seu *buffer* em memória, quanto pelo correto repasse de dados no formato IEEE C37.118. Observa-se, em destaque, no canto superior esquerdo a lista de PMUs cujos dados estão sendo disponibilizados em tempo real. Para o caso do PDC Mestre, todas as PMUs do SPMS estão presentes nesta lista.

Cabe ressaltar que a utilização de um *software* externo aos desenvolvimentos do projeto MedFasee, para esta validação, procurou eliminar os efeitos de erros sistemáticos que poderiam estar presentes nos desenvolvimentos do PDC e que não são detectados.

Ainda se tratando da validação da estrutura de tempo real, o envio de dados para aplicações é, conceitualmente, feito apenas pelo PDC Mestre. Entretanto, a fim de testar a validade dos dados em tempo real enviados pelos *PDCs Front End*, aplicou-se o mesmo teste a fim de capturar os sincrofasores e comprovar que a transmissão é feita também no formato IEEE C37.118. A Figura 5.6 e a Figura 5.7 ilustram a tela capturada em tempo de execução do TVA PMU Connection Tester, recebendo dados dos *PDCs Front End* 1 e 2 respectivamente.

Dado o correto funcionamento do programa, pode-se observar que os *PDCs Front End* enviam um fluxo de dados no formato IEEE C37.118 ao PDC Mestre, utilizando a estrutura de *buffer* em memória do projeto de *software*. Observam-se as respectivas listas de PMUs em destaque em cada tela, contendo as PMUs que enviam dados aos respectivos *PDCs Front End*.

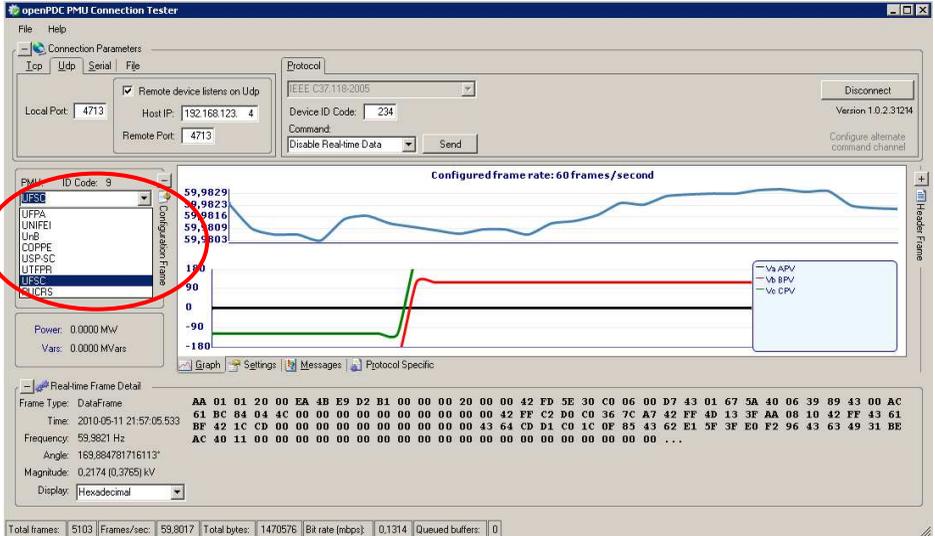


Figura 5.5 – Interface gráfica do TVA PMU Connection Tester em funcionamento, recebendo dados do PDC Mestre.

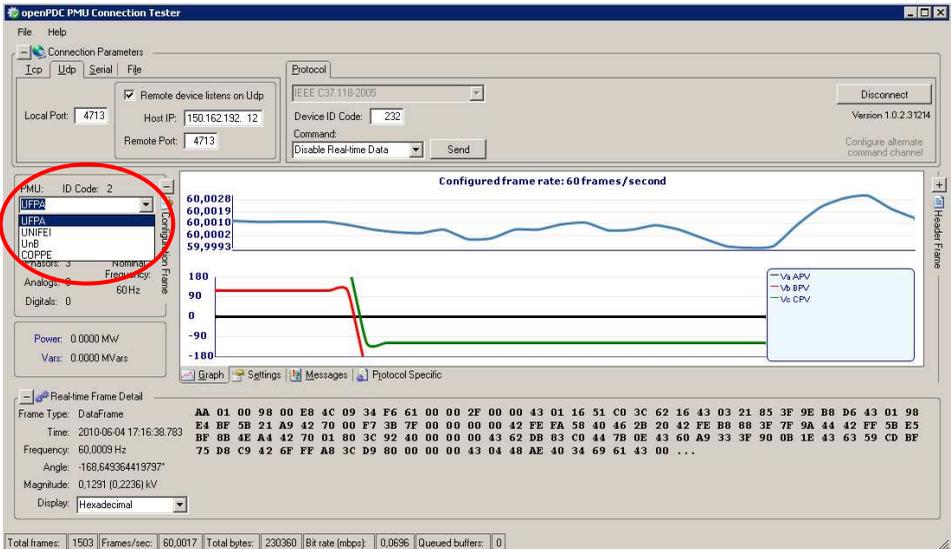


Figura 5.6 – Interface gráfica do TVA PMU Connection Tester em funcionamento, recebendo dados do PDC Front End 1.

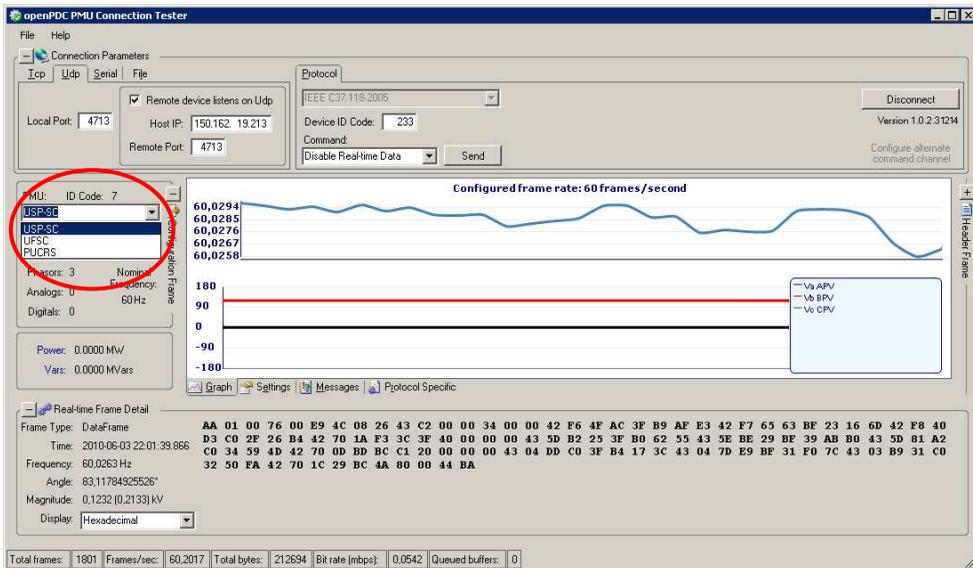


Figura 5.7 - Interface gráfica do TVA PMU Connection Tester em funcionamento, recebendo dados do PDC Front End 2.

## 5.4 Validação da estrutura de dados histórica e análise de ocorrência

A validação da inserção de dados na base histórica do PDC é realizada por meio de duas maneiras: (i) validação entre bases de dados dos *PDCs Front End* e PDC Mestre, buscando verificar a aderência dos dados nos diferentes equipamentos; (ii) validação contra ocorrências no Sistema Interligado Nacional (SIN), buscando observar a reprodução dos eventos com os dados armazenados no PDC Mestre. A seguir são discutidos estes dois cenários.

### 5.4.1 Validação da base de dados entre *PDCs Front End* e PDC Mestre

Primeiramente busca-se verificar a coerência dos dados armazenados nos PDCs, tanto na base de dados local do *PDC Front End* quanto na base de dados central do PDC Mestre. Para isso foram recuperados das respectivas bases de dados os valores dos fasores em um mesmo instante de tempo, sendo realizada a comparação de seus valores absolutos, na expectativa de ocorrência nula de diferenças entre estes valores.

A fim de realizar o resgate destes dados, utilizou-se o programa MySQL Control Center, *software* cliente para acesso à bases de dado em formato MySQL. Este *software* é disponibilizado gratuitamente, da mesma maneira que o próprio banco de dados MySQL, no endereço [HTTP://www.mysql.com](http://www.mysql.com). Este cliente possui diversas funcionalidades, destacando-se a possibilidade do usuário executar buscas e filtragens aos dados disponibilizados na base, exibindo os dados de maneira ordenada e com possibilidade de exportação para tratamento em outros aplicativos de uso padrão.

A

Tabela 5.8 e a Tabela 5.9 ilustram as comparações entre os dados coletados das PMU UFSC e PMU UFPA, presentes no Projeto MedFasee Frequência. O período selecionado corresponde ao primeiro segundo das 20:00:00 GMT (17:00:00 horário de Brasília) do dia 02/06/2010, com os valores de módulo e ângulo correspondentes à Fase A (canal 0) das PMUs. Cabe lembrar que a taxa de amostragem do protótipo de SPMS do projeto MedFasee Frequência é de 60 sincrofasores por segundo. Com objetivo de melhor apresentar os resultados dos dados históricos, optou-se por demonstrar apenas os valores relativos aos 30 primeiros sincrofasores do segundo em análise, dado que uma vez verificado a aderência dos dados deste período infere-se o correto processamento dos demais dados.

*Tabela 5.8 – Dados da PMU UFPA presentes no PDC Front End 1 e Mestre.*

PDC Front End 1 - PMU UFPA						PDC Mestre - PMU UFPA						Variação	
idp	phas	tempo	cont	modulo	angulo	dp	phas	tempo	cont	modulo	angulo	Δ módulo	Δ ângulo
2	0	1275508800	0	122.793	-79.6682	2	0	1275508800	0	122.793	-79.6682	0,000	0,000
2	0	1275508800	1	122.799	-79.4644	2	0	1275508800	1	122.799	-79.4644	0,000	0,000
2	0	1275508800	2	122.797	-79.2565	2	0	1275508800	2	122.797	-79.2565	0,000	0,000
2	0	1275508800	3	122.799	-79.0476	2	0	1275508800	3	122.799	-79.0476	0,000	0,000
2	0	1275508800	4	122.794	-78.8428	2	0	1275508800	4	122.794	-78.8428	0,000	0,000
2	0	1275508800	5	122.772	-78.657	2	0	1275508800	5	122.772	-78.657	0,000	0,000
2	0	1275508800	6	122.753	-78.4703	2	0	1275508800	6	122.753	-78.4703	0,000	0,000
2	0	1275508800	7	122.750	-78.2696	2	0	1275508800	7	122.750	-78.2696	0,000	0,000
2	0	1275508800	8	122.761	-78.0513	2	0	1275508800	8	122.761	-78.0513	0,000	0,000
2	0	1275508800	9	122.772	-77.8295	2	0	1275508800	9	122.772	-77.8295	0,000	0,000
2	0	1275508800	10	122.775	-77.6652	2	0	1275508800	10	122.775	-77.6652	0,000	0,000
2	0	1275508800	11	122.773	-77.4596	2	0	1275508800	11	122.773	-77.4596	0,000	0,000
2	0	1275508800	12	122.775	-77.2149	2	0	1275508800	12	122.775	-77.2149	0,000	0,000
2	0	1275508800	13	122.783	-77.0060	2	0	1275508800	13	122.783	-77.0060	0,000	0,000
2	0	1275508800	14	122.789	-76.8023	2	0	1275508800	14	122.789	-76.8023	0,000	0,000
2	0	1275508800	15	122.789	-76.6023	2	0	1275508800	15	122.789	-76.6023	0,000	0,000
2	0	1275508800	16	122.788	-76.4036	2	0	1275508800	16	122.788	-76.4036	0,000	0,000
2	0	1275508800	17	122.793	-76.1967	2	0	1275508800	17	122.793	-76.1967	0,000	0,000
2	0	1275508800	18	122.796	-75.9897	2	0	1275508800	18	122.796	-75.9897	0,000	0,000
2	0	1275508800	19	122.801	-75.7779	2	0	1275508800	19	122.801	-75.7779	0,000	0,000
2	0	1275508800	20	122.809	-75.6087	2	0	1275508800	20	122.809	-75.6087	0,000	0,000
2	0	1275508800	21	122.815	-75.4017	2	0	1275508800	21	122.815	-75.4017	0,000	0,000
2	0	1275508800	22	122.817	-75.1902	2	0	1275508800	22	122.817	-75.1902	0,000	0,000
2	0	1275508800	23	122.821	-74.9846	2	0	1275508800	23	122.821	-74.9846	0,000	0,000
2	0	1275508800	24	122.836	-74.7681	2	0	1275508800	24	122.836	-74.7681	0,000	0,000
2	0	1275508800	25	122.849	-74.5502	2	0	1275508800	25	122.849	-74.5502	0,000	0,000
2	0	1275508800	26	122.850	-74.3427	2	0	1275508800	26	122.850	-74.3427	0,000	0,000
2	0	1275508800	27	122.842	-74.1049	2	0	1275508800	27	122.842	-74.1049	0,000	0,000
2	0	1275508800	28	122.838	-73.9097	2	0	1275508800	28	122.838	-73.9097	0,000	0,000
2	0	1275508800	29	122.840	-73.7027	2	0	1275508800	29	122.840	-73.7027	0,000	0,000

Tabela 5.9 – Dados da PMU UFSC presentes no PDC Front End 2 e Mestre.

PDC Front End 2 - PMU UFSC						PDC Mestre - PMU UFSC						Variação	
dp	hchpha	tempo	cont	modulo	angulo	dp	hchpha	tempo	cont	modulo	angulo	Δ módulo	Δ ângulo
9	0	1275508800	0	219.857	-104.973	9	0	1275508800	0	219.857	-104.973	0,000	0,000
9	0	1275508800	1	219.865	-104.742	9	0	1275508800	1	219.865	-104.742	0,000	0,000
9	0	1275508800	2	219.863	-104.502	9	0	1275508800	2	219.863	-104.502	0,000	0,000
9	0	1275508800	3	219.853	-104.267	9	0	1275508800	3	219.853	-104.267	0,000	0,000
9	0	1275508800	4	219.847	-104.037	9	0	1275508800	4	219.847	-104.037	0,000	0,000
9	0	1275508800	5	219.845	-103,81	9	0	1275508800	5	219.845	-103,81	0,000	0,000
9	0	1275508800	6	219.844	-103.573	9	0	1275508800	6	219.844	-103.573	0,000	0,000
9	0	1275508800	7	219.843	-103.338	9	0	1275508800	7	219.843	-103.338	0,000	0,000
9	0	1275508800	8	219.839	-103.098	9	0	1275508800	8	219.839	-103.098	0,000	0,000
9	0	1275508800	9	219.840	-102.863	9	0	1275508800	9	219.840	-102.863	0,000	0,000
9	0	1275508800	10	219.849	-102.627	9	0	1275508800	10	219.849	-102.627	0,000	0,000
9	0	1275508800	11	219.852	-102.383	9	0	1275508800	11	219.852	-102.383	0,000	0,000
9	0	1275508800	12	219.848	-102.145	9	0	1275508800	12	219.848	-102.145	0,000	0,000
9	0	1275508800	13	219.847	-101.904	9	0	1275508800	13	219.847	-101.904	0,000	0,000
9	0	1275508800	14	219.849	-101.669	9	0	1275508800	14	219.849	-101.669	0,000	0,000
9	0	1275508800	15	219.846	-101,44	9	0	1275508800	15	219.846	-101,44	0,000	0,000
9	0	1275508800	16	219.844	-101.207	9	0	1275508800	16	219.844	-101.207	0,000	0,000
9	0	1275508800	17	219.842	-100,97	9	0	1275508800	17	219.842	-100,97	0,000	0,000
9	0	1275508800	18	219.835	-100.737	9	0	1275508800	18	219.835	-100.737	0,000	0,000
9	0	1275508800	19	219.828	-100.502	9	0	1275508800	19	219.828	-100.502	0,000	0,000
9	0	1275508800	20	219,81	-100.272	9	0	1275508800	20	219,81	-100.272	0,000	0,000
9	0	1275508800	21	219.794	-100.038	9	0	1275508800	21	219.794	-100.038	0,000	0,000
9	0	1275508800	22	219.794	-997.943	9	0	1275508800	22	219.794	-997.943	0,000	0,000
9	0	1275508800	23	219.791	-995.564	9	0	1275508800	23	219.791	-995.564	0,000	0,000
9	0	1275508800	24	219.780	-993.225	9	0	1275508800	24	219.780	-993.225	0,000	0,000
9	0	1275508800	25	219.768	-990.957	9	0	1275508800	25	219.768	-990.957	0,000	0,000
9	0	1275508800	26	219.763	-988.662	9	0	1275508800	26	219.763	-988.662	0,000	0,000
9	0	1275508800	27	219.765	-986.321	9	0	1275508800	27	219.765	-986.321	0,000	0,000
9	0	1275508800	28	219.771	-983.999	9	0	1275508800	28	219.771	-983.999	0,000	0,000
9	0	1275508800	29	219.773	-98.165	9	0	1275508800	29	219.773	-98.165	0,000	0,000

Pode ser observado que para os dois casos os dados dos *PDCs Front End 1 e 2* estão em concordância com aqueles apresentados no *PDC Mestre*. Isto é mostrado pelo cálculo da variação entre os valores, apresentados em destaque nas tabelas. Desta forma verifica-se a correta operação do *PDCS* em seus dois níveis de aquisição de dados.

Obtém-se também como resultado desta análise a correta operação de repasse dos dados do *PDC Front End* para o *PDC Mestre*, que ocorre por meio de frames IEEE C37.118 utilizando a estrutura de *buffer* de dados em tempo real, cuja interface foi apresentada na Seção 5.3.

Com intuito de apresentar o comportamento do *PDCS* face ao tratamento da perda de dados, a Tabela 5.10 ilustra trinta valores de sincronizadores presentes no segundo do instante de tempo das 22:30:14 GMT (19:30:14 horário de Brasília) do dia 02/06/2010, referente a *PMU UFPA do PDC Front End 1*.

Tabela 5.10 – Dados da *PMU UFPA* armazenados no *PDC Front End 1* e *PDC Mestre*. Período contemplando perda de dados.

PDC Front End 1 - PMU UFPA					PDC Mestre - PMU UFPA						
hcodepm	hchpha	tempo	cont	modulo	angulo	hcodepm	hchpha	tempo	cont	modulo	angulo
2	0	1275517814	0	128.578	-34.2182	2	0	1275517814	0	128.578	-34.2182
2	0	1275517814	1	128.572	-34.1967	2	0	1275517814	1	128.572	-34.1967
2	0	1275517814	3	128.569	-34.132	2	0	1275517814	2	0	0
2	0	1275517814	4	128.571	-34.0906	2	0	1275517814	3	128.569	-34.132
2	0	1275517814	5	128.572	-34.0539	2	0	1275517814	4	128.571	-34.0906
2	0	1275517814	6	128.573	-34.0196	2	0	1275517814	5	128.572	-34.0539
2	0	1275517814	7	128.573	-33.9876	2	0	1275517814	6	128.573	-34.0196
2	0	1275517814	8	128.569	-33.9636	2	0	1275517814	7	128.573	-33.9876
2	0	1275517814	9	128.572	-33.931	2	0	1275517814	8	128.569	-33.9636
2	0	1275517814	10	128.573	-33.8949	2	0	1275517814	9	128.572	-33.931
2	0	1275517814	11	128.57	-33.8675	2	0	1275517814	10	128.573	-33.8949
2	0	1275517814	12	128.572	-33.8319	2	0	1275517814	11	128.57	-33.8675
2	0	1275517814	13	128.579	-33.7958	2	0	1275517814	12	128.572	-33.8319
2	0	1275517814	14	128.583	-33.7594	2	0	1275517814	13	128.579	-33.7958
2	0	1275517814	15	128.586	-33.7213	2	0	1275517814	14	128.583	-33.7594
2	0	1275517814	16	128.59	-33.6897	2	0	1275517814	15	128.586	-33.7213
2	0	1275517814	17	128.589	-33.663	2	0	1275517814	16	128.59	-33.6897
2	0	1275517814	18	128.586	-33.6308	2	0	1275517814	17	128.589	-33.663
2	0	1275517814	19	128.585	-33.602	2	0	1275517814	18	128.586	-33.6308
2	0	1275517814	20	128.582	-33.5751	2	0	1275517814	19	128.585	-33.602
2	0	1275517814	21	128577	-33.5458	2	0	1275517814	20	128582	-33.5751
2	0	1275517814	22	128.572	-33.5212	2	0	1275517814	21	128.577	-33.5458
2	0	1275517814	23	128.573	-33.489	2	0	1275517814	22	128.572	-33.5212
2	0	1275517814	24	128.576	-33.4494	2	0	1275517814	23	128.573	-33.489
2	0	1275517814	25	128.564	-33.4128	2	0	1275517814	24	128.576	-33.4494
2	0	1275517814	26	128.55	-33.3788	2	0	1275517814	25	128.564	-33.4128
2	0	1275517814	27	128.552	-33.3422	2	0	1275517814	26	128.55	-33.3788
2	0	1275517814	28	128.547	-33.3185	2	0	1275517814	27	128.552	-33.3422
2	0	1275517814	29	128.541	-33.291	2	0	1275517814	28	128.547	-33.3185
						2	0	1275517814	29	128.541	-33.291

Observa-se em destaque a perda do sincrofasor referente à posição 2 do segundo em análise. Considerando a operação do *PDC Front End*, um sincrofasor perdido é detectado pela sua ausência na base de dados. Este comportamento deverá ser contemplado pelas aplicações que farão uso dos dados disponibilizados na base MySQL.

Entretanto, como o *PDC Front End* recebe dados de outras PMUs do SPMS, o *frame* de dados que será enviado para o PDC Mestre deverá conter o valor dos sincrofasores das demais PMUs deste mesmo instante de tempo. Desta forma, toda vez que é detectado a ausência dos dados em função de perdas, os respectivos valores são preenchidos com zero e repassados ao PDC Mestre, como pode ser observado em destaque na Tabela 5.10, onde o sincrofasor relativo à posição 2 possui valores de módulo e ângulo iguais a zero. Esta estratégia pressupõe que as aplicações que farão uso dos dados históricos deverão saber lidar com valores nulos dentro da base de dados, tomando as ações necessárias.

#### **5.4.2 Validação da base de dados em relação ao SIN**

Uma vez validada a estrutura de dados histórica no que tange à aderência entre os *PDCs Front End* e Mestre, é necessário verificar a adequação dos dados armazenados pelo PDCS face à eventos ocorridos no SIN.

Para isso são analisados dados armazenados na base de dados histórica do PDC Mestre, tendo como período de análise o intervalo de 03:00h a 03:05h, horário de Brasília, período presente no Informativo Preliminar Diário da Operação (IPDO) divulgado pelo ONS referente ao dia 02/06/2010, relatando a seguinte ocorrência:

```
"Às 03h00min ocorreu o desligamento automático geral do setor de 345kV da subestação SUL (CTE-EP). Causa: a ser identificada. No momento do desligamento houve manobras de normalização do disjuntor de paralelo DJP1 que estava sob intervenção programada. Consequência: interrupção de 540MW de carga da Eletropaulo na região do ABC Paulista e nas zonas sul e central da cidade de São Paulo. Normalização: iniciada
```

às 03h08min e concluída às 03h11min.”

Com o auxílio das ferramentas desenvolvidas no âmbito do Projeto MedFasee, é possível consultar o banco de dados MySQL do PDC Mestre, com objetivo de gerar gráficos que são analisados em seqüência.

Primeiramente, foi realizada a busca na base histórica do PDC Mestre, no momento descrito pelo IPDO, a fim de recuperar o comportamento da frequência do SIN durante a ocorrência, como ilustrado na Figura 5.8. Destaca-se a magnitude máxima de 60,17 Hz durante o evento, destacado na figura, voltando aos valores em torno da frequência nominal nos minutos seguintes.

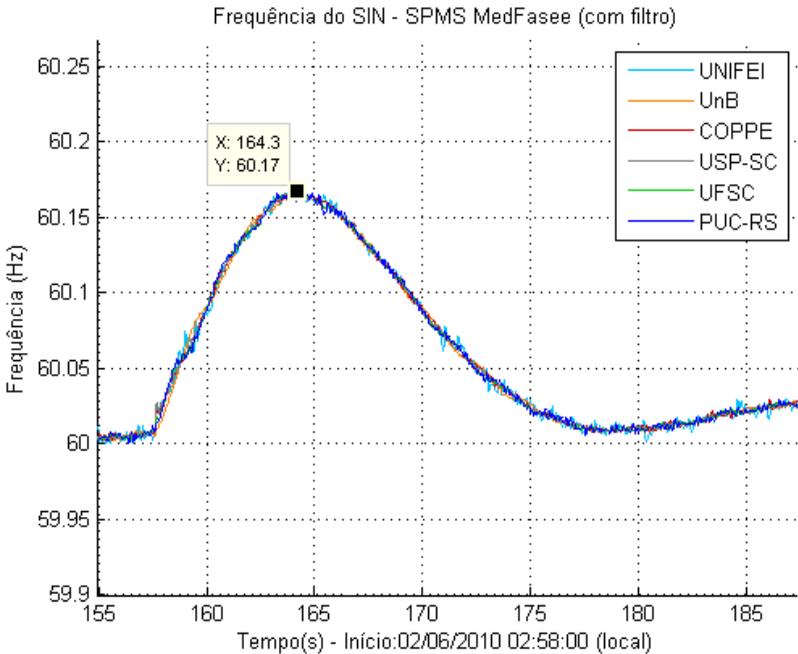
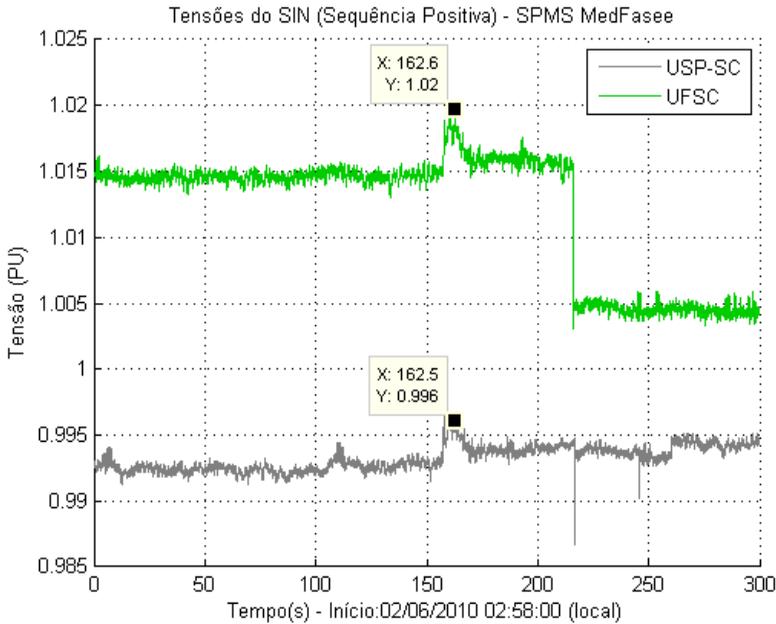


Figura 5.8 – Frequência do SIN no momento da ocorrência .

Em seguida, a Figura 5.9 apresenta o comportamento das tensões verificadas pelas PMUs da USP-SC e UFSC. Observa-se indicado nesta figura a variação de 0,05pu na rede de baixa tensão da PMU UFSC, tendo em seguida o acionamento de um tap regulador de tensão, agindo no sentido de reduzir o nível de tensão da rede. Já para a PMU USP-SC,

observa-se o comportamento crescente na tensão, também indicando na figura, porém em valores pouco significativos.



*Figura 5.9 – Tensões do SIN no momento da ocorrência.*

A Figura 5.10 ilustra as diferenças angulares entre as PMUs USP-SC e UFSC. Observa-se no momento da ocorrência que houve um pico de diferença de  $4,875^\circ$ , caracterizando as variações de fluxo de potência ativa pela rede. Tais variações foram corrigidas assim que as ações corretivas no sistema foram adotadas.

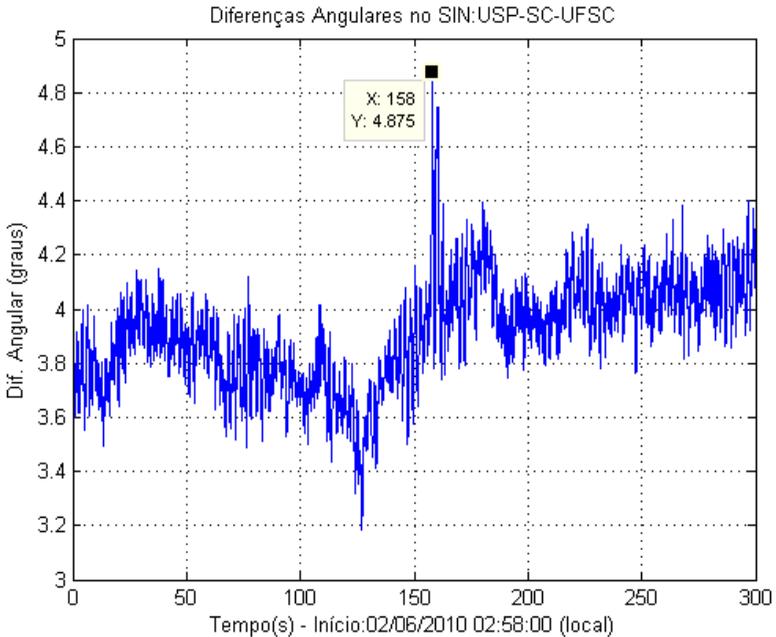


Figura 5.10 – Diferença Angular entre PMUs USP-SC e UFSC.

Percebe-se pelo momento inicial indicado pelo IPDO e pelos gráficos de frequência, tensão e diferença angular, que o PDCS teve êxito em representar a ocorrência relatada no SIN, a partir de dados coletados da rede de baixa tensão das universidades participantes do Projeto MedFasee Frequência. Desta forma, se valida a estrutura de armazenamento histórico de sincrofasores da nova estrutura do PDC no que tange à reprodução de eventos no sistema elétrico.

## 5.5 Comentários Finais

Verifica-se que, baseado nas análises apresentadas anteriormente, a estrutura de aquisição de dados de medição fasorial sincronizada em dois níveis hierárquicos opera de maneira coerente. Foi possível observar o comportamento do sistema em diferentes níveis de análise, compreendendo desde sua performance operacional até a validação dos dados processados em sua base de dados histórica.

É importante salientar os benefícios oferecidos pela utilização do protótipo de Sistema de Medição Fasorial Sincronizada do Projeto MedFasee Frequência, que dispõe de nove PMUs instaladas na rede de

baixa tensão em universidades brasileiras, abrangendo todo o Sistema Interligado Nacional.

A utilização deste SPMS para validação do PDCS permitiu verificar as mais diversas condições operacionais, sobretudo no que tange à continuidade e disponibilidade das PMUs no sistema. Observaram-se momentos em que se teve a total disponibilidade dos equipamentos, e momentos onde somente alguns estavam enviando dados para os PDCs.

Fatores como utilização da rede local onde as PMUs estão instaladas, falhas nos equipamentos, falha na alimentação de energia local e problemas de sincronização com o relógio GPS fazem com que a topologia de medição do SPMS mude a todo instante. Esta condição de operação é propícia para o desenvolvimento do PDC, pois permite a identificação e correção de problemas operacionais que possam vir a aparecer, agregando robustez e estabilidade à aquisição e processamento dos dados fasoriais.

# CAPÍTULO 6

## 6. Conclusões

O objetivo deste trabalho foi desenvolver um sistema de concentração de dados de medição fasorial sincronizada, de arquitetura descentralizada (utilizando mais de um equipamento com funcionalidade de PDC) e arranjo hierárquico, visando assegurar ao sistema escalabilidade em termos de processamento e comunicação. Os desenvolvimentos em PDC realizados no âmbito do Projeto MedFasee e apresentados em (DECKER, I.C., 2006) serviram de referência para a implementação do sistema, sendo realizada a adaptação e modificação de seu código-fonte de forma a tornar o PDC compatível com a solução de PDCS proposto neste trabalho. Estes desenvolvimentos, em conjunto com a implementação e operação do PDCS dentro da estrutura do protótipo de SPMS do Projeto MedFasee Frequência, formam a principal contribuição deste trabalho.

O desenvolvimento inicial do projeto exigiu a separação conceitual do sistema em dois aspectos: (i) operação do sistema e (ii) operação dos equipamentos com funcionalidade de PDC do sistema. Devido ao fato da adoção da arquitetura descentralizada, primeiramente estabeleceu-se a forma como os PDCs do sistema deveriam se comunicar. A lógica de operação do sistema apresentada no Capítulo 3 permite que os equipamentos troquem informações na forma e no instante corretos. Em um segundo momento, a lógica de operação do equipamento com funcionalidade de PDC foi estabelecida, sendo dividida em Etapa de Inicialização e Etapa Principal de Operação. Fluxogramas foram apresentados de forma a ilustrar estas etapas e os processos internos executados nas durante a operação do *software* do PDC. Esta lógica permite que, dependendo da função hierárquica exercida pelo PDC (*Front End* ou *Mestre*), o fluxo de dados recebido pelo equipamento tenha o destino correto. Esta separação conceitual é fundamental para o entendimento da estrutura e serviu de diretriz para o desenvolvimento de *software* do projeto.

O desenvolvimento de *software* deste trabalho fez uso dos conceitos da Modelagem Orientada a Objetos. Foram apresentadas seis

classes que têm por objetivo abstrair elementos físicos presentes nos SPMS (C\_PDC, C\_PMU\_PDC e C\_PDC\_PDC) e características funcionais internas à operação do PDC (C\_Conexão e C\_App). Estas classes em conjunto fazem parte da implementação de *software* que viabiliza a comunicação entre PDCs, característica necessária para o funcionamento do PDCS proposto.

Os formatos de comunicação entre PDCs não são especificados em documentos ou normas técnicas. A utilização da norma IEEE C37.118 foi adotada como solução, porém não se trata de solução única, sendo que novos formatos podem vir a ser determinados para esta finalidade. A Classe C\_Frame permite a expansão do código-fonte de forma que o PDC possa ser flexível para a compatibilização com novos formatos de comunicação.

Detalhes sobre a disponibilização de dados em tempo real são apresentados no Capítulo 4. Um *buffer* de dados em memória é utilizado para armazenar os dados que chegam ao PDC por um período de tempo pré-determinado. Este *buffer* é acessado pelo método de resincronização de dados que disponibiliza pacotes de dados de diferentes equipamentos do SPMS em uma mesma etiqueta de tempo. O pacote de dados é processado para o formato do *frame* de dados da norma IEEE C37.118 e encaminhado para seu destino final (PDC Mestre ou aplicações em tempo real). O entendimento e adaptação da interface de disponibilização de dados em tempo real presente no código-fonte do PDC MedFasee foi fundamental, dado que sem ela, a comunicação entre PDCs não poderia ser viabilizada.

A implementação do projeto de *software* foi realizada em código-fonte único, de forma que a utilização dos métodos e atributos para cada função hierárquica fosse determinada por um parâmetro de configuração definido pelo usuário. Esta estratégia tem por objetivo concentrar os desenvolvimentos em códigos que fossem compilados sempre em conjunto, evitando problemas ao longo do desenvolvimento e proporcionando que um mesmo pacote de instalação pudesse ser disponibilizado para PDCs, independente de sua função dentro dos níveis hierárquicos do PDCS. Com os desenvolvimentos encapsulados em um mesmo código, o desenvolvimento de *software* é simplificado, bem como sua configuração e instalação pelo usuário final.

Utilizando como plataforma de testes o protótipo de SPMS do Projeto MedFasee Freqüência, foram implementados dois protótipos de PDCS conforme apresentados no Capítulo 5. O primeiro contendo apenas equipamentos executando o *software* desenvolvido a partir do PDC

MedFasee e um segundo contemplando a presença de um equipamento executando o *software* do openPDC. O funcionamento destes dois protótipos permitiu verificar a correta comunicação entre os PDCs presentes no PDCS, mostrando também a escalabilidade do sistema, validando a estrutura hierárquica proposta no Capítulo 3. Como resultado desta operação também pode ser observado a interoperabilidade com sistemas de diferentes plataformas de desenvolvimento, uma vez que o sistema operacional compatível com o *openPDC* é o *MS Windows*. Esta operação conjunta é importante, pois é possível verificar que, independente das soluções adotadas internamente aos *softwares* dos PDCs, sua comunicação é realizada de maneira padronizada via *frames* da norma IEEE C37.118.

Referente ao desempenho operacional dos PDCS implementados, foram avaliadas características de capacidade de rede e armazenamento, perdas de dados, disponibilização de dados em tempo real e em base histórica. Verificou-se que, relativo à capacidade do PDCS, a demanda de rede é distribuída pelos PDCs *Front End*, uma vez que não somente o número de PMUs que enviam dados a um ponto da rede é diminuído, como os dados das PMUs são concatenados em um único *frame* de dados, compartilhando os campos fixos presentes nos *frames* da norma IEEE C37.118. Estas características reduzem significativamente a necessidade de utilização de largura de banda. No tocante à demanda de armazenamento em base de dados históricos, o limite está associado ao *hardware* utilizado no sistema, sendo necessário expandir a capacidade física de acordo com o tamanho da base histórica que se deseja utilizar.

A análise de perdas de dados constatou que o *software* do PDC não introduz perdas artificiais no sistema, uma vez que o *PDC Front End* que recebe os dados da PMU UFSC, conectados em rede local, apresentou perdas nulas. As perdas verificadas em PMUs conectadas aos *PDCs Front End* com conexão VPN pela Internet apresentaram os níveis esperados, dado a utilização do protocolo de transporte UDP/IP. Foram identificados casos onde perdas de dados ocorreram entre *PDCs Front End* e PDC Mestre. Estas perdas ocorrem em casos específicos de operação do PDCS, em que todas as PMUs de um determinado PDC *Front End* param de enviar dados. Estes dados, que estão demasiadamente atrasados, são descartados do *buffer* em memória, porém são armazenados em base de dados histórica.

Para a análise da disponibilização de dados em tempo real e base histórica foram realizados testes com *softwares* desenvolvidos fora do

âmbito do Projeto MedFasee. Para a disponibilização de dados de tempo real, utilizou-se o *TVA PMU Connection Tester* para receber os dados enviados pelo PDC Mestre. O correto recebimento dos dados pelo aplicativo, apresentado no Capítulo 5, valida a implementação da Classe *C\_App* descrita no Capítulo 4. Do ponto de vista da base histórica, observou-se que os dados armazenados localmente pelos *PDCs Front End* são os mesmos armazenados no PDC Mestre, caracterizando o correto processamento dos dados pelos PDCs. Foi verificado também o registro de uma ocorrência no SIN, indicando que os dados armazenados reproduzem o comportamento do sistema elétrico.

Com base nos resultados do Capítulo 5 e na performance operacional dos PDCS implementados verifica-se que o objetivo principal do trabalho foi atingido de maneira satisfatória. Além do correto funcionamento dos protótipos de PDCS, cuja principal contribuição é agregar escalabilidade em termos de processamento e comunicação, pode-se destacar as seguintes contribuições do trabalho em relação aos desenvolvimentos presentes no PDC MedFasee: viabilização da comunicação entre PDCs; disponibilização de dados em tempo real em formato de *frames* da norma IEEE C37.118; a possibilidade de expansão para novos formatos de comunicação por meio da Classe *C\_Frame*; utilização de arquivo no formato XML para armazenar os dados de configuração do PDC fornecidos pelo usuário.

Cabe ressaltar que, apesar da vasta literatura encontrada sobre SPMS, poucas referências são destinadas aos Concentradores de Dados Fasoriais. A documentação dos desenvolvimentos deste trabalho, considerando o nível de detalhamento abordado, é outra contribuição deste trabalho.

## **6.1 Sugestões para Trabalhos Futuros**

Dentre os aspectos que podem seguir além dos trabalhos apresentados até este ponto, destaca-se a expansão do PDCS, atualmente com dois níveis hierárquicos, para compreender um número  $n$  de níveis. Desta forma o PDC pode se movimentar ao longo da cadeia de aquisição de dados, sendo posicionado de maneira que melhor convém às necessidades do SPMS. Outro aspecto a ser observado é a utilização do banco de dados MySQL para armazenamento dos parâmetros de configuração do SPMS, atualmente realizado pelo arquivo de configuração XML. Isto permitiria a utilização de interfaces padronizadas na configuração do sistema, restringindo ao máximo o usuário lidar com soluções proprietárias.

Considerando o desempenho dos bancos de dados, uma análise em função do tipo de banco de dados utilizado pelo PDC carece de investigação, dado a presença dos mais variados tipos de bancos de dados disponíveis para utilização, tais como Oracle, PostgreSQL, Access, entre outros. Cada banco tem características distintas, podendo ser avaliado seu desempenho quanto ao acesso, escrita e recuperação dos síncronos das respectivas bases de dados.

Adicionalmente, pode-se citar a utilização de outros sistemas operacionais para a execução do processo do PDC. Atualmente utiliza-se o sistema GNU/Linux, porém é interessante avaliar o desenvolvimento de um código-fonte que procure ser independente de plataforma operacional e que possa manter as características necessárias para a aquisição de fasores em tempo real.



# APÊNDICES

## Apêndice A. Ferramentas de Desenvolvimento

Esta seção apresenta as ferramentas de desenvolvimento estudadas e utilizadas no estudo e implementação do projeto de *software* do PDC.

### A.1. XML

XML é a sigla para EXtensible Markup Language, linguagem que tem por objetivo descrever dados de maneira padronizada em formato de texto. XML foi projetada para o transporte e o armazenamento de dados, tendo em foco o que estes dados representam.

XML é derivada da *Standard Generalized Markup Language* (SGML), linguagem pioneira desenvolvida para o compartilhamento comum de documentos técnicos. A primeira adaptação do SGML foi o *Hyper Text Markup Language* (HTML), linguagem semelhante ao XML, dedicada a organização e apresentação padronizada de documentos, presente na maior parte das aplicações na Internet atualmente (BENZ, B., *et al.*, 2003).

Apesar das semelhanças entre si, HTML e XML diferem não somente nos seus objetivos, mas também na maneira de utilização. Tendo em vista que a principal preocupação do XML é o transporte de dados, as etiquetas que envolvem os dados não são padronizadas, fornecendo livre arbítrio ao programador para representá-las da maneira que quiser, como no exemplo a seguir.

```
<?xml version="1.0" encoding="UTF-8"?>
<Clientes>
    <Nome>João Silva</Nome>
    <Nome>Adriano Carvalho</Nome>
</Clientes>
```

*Exemplo de Arquivo XML.*

Percebe-se claramente que os dados deste documento referem-

se a **nomes de clientes**. Uma aplicação que deseja utilizar esta informação estará interessada no conteúdo do elemento <Nome>, que uma vez acessada será tratada como um dado em formato padrão, neste caso uma string.

Para o HTML, não há interesse se “João Silva” é um nome de um cliente, e sim como ele será apresentado (listas, tabelas, figuras, etc.). Desta forma existe a necessidade que as etiquetas HTML sejam padronizadas, indicando a ação que uma aplicação deve tomar em relação ao arquivo descrito.

Considerando a liberdade para a criação de etiquetas disponibilizada pelo XML, pode se pensar que qualquer tarefa pode ser executada através dele. Porém é importante salientar que um arquivo XML é apenas informação envolvida em etiquetas, que em si não realiza ou executa tarefa alguma. Cabe ao programador interpretar as informações contidas no arquivo para o uso em sua aplicação. XML se tornou recomendação (W3C, 2008) e atualmente encontra-se na sua quinta edição.

#### A.1.1.1. **Definições**

Esta seção tem por objetivo apresentar as definições básicas para se entender um documento escrito no formato XML. Exemplos são elaborados a fim de explicitar os conceitos abordados.

##### ***Elemento***

Um elemento pode ser definido como tudo aquilo que está compreendido desde a etiqueta de abertura até a etiqueta de fechamento, possuindo o seguinte formato:

```
<Elemento></Elemento>
```

Os nomes dos elementos podem conter letras, números, hífen, sublinhado, ponto e dois pontos. É permitido iniciar um nome de elemento com dois pontos, letras e sublinhado, porém nenhum outro caractere que não seja alfabético pode ser utilizado, nem as letras “XML”.

A fim de substituir os espaços para nomes de elementos, comumente se utiliza o caractere sublinhado “\_”, por exemplo:

```
<fasor_corrente></fasor_corrente>
```

##### ***Atributos***

Atributos possuem valores que são diretamente associados ao elemento que lhe diz respeito. Estão sempre localizados na etiqueta de abertura, possuindo o seguinte formato:

```
<PMU no_canais="48"></PMU>
```

Os valores dos atributos são sempre associados aos nomes dos atributos através do sinal de igual (=) e devem estar envolvidos por aspas duplas (“”).

### ***Texto***

O texto de um elemento são todos os caracteres que se encontram entre as etiquetas, representando os dados de fato a serem transportados. O texto não precisa obedecer às regras impostas aos Elementos e Atributos, como por exemplo:

```
<IP_PMU>148.165.49.25</IP_PMU>
```

### ***Elementos vazios***

Elementos que não possuem nem Atributos nem Texto podem ser abreviados da seguinte maneira:

```
<PMU />
```

## **A.1.2. Regras de formação do documento**

Embora não haja restrições quanto à nomenclatura das etiquetas, existem algumas regras para a formação do arquivo XML.

1. Uma vez que uma etiqueta for aberta, ela deve necessariamente ser fechada. Por exemplo:

```
<PMU>UFSC</PMU>
```

2. As etiquetas são sensíveis a caixa. Portanto um elemento `<pmu>UFSC</pmu>` é diferente do elemento `<PMU>UFSC</PMU>` embora ambas tenham o mesmo conteúdo.

3. Os elementos devem estar devidamente aninhados, como no exemplo a seguir:

```
<SPMS><num_PMU>10</num_PMU></SPMS>
```

4. É necessário existir um elemento raiz único dentro do documento. Todos os outros elementos são aninhados dentro dele.

5. Um arquivo XML sempre deve começar com uma etiqueta padrão para identificação da versão XML utilizado e do conjunto de caracteres que se está trabalhando, como no caso a seguir.  
`<?xml version="1.0" encoding="UTF-8" ?>`

6. Comentários possuem sintaxe própria, como apresentado abaixo:

```
<!-- Comentario -->
```

Um documento que possui sintaxe correta, ou seja, obedece aos pré-requisitos descritos acima é dito “*well-formed*”, em outras palavras possui “*boa-formação*”. Esta definição é importante, pois se difere do conceito de **validação**, que será discutido mais adiante.

### A.1.3. Estrutura em árvore

Uma das regras de sintaxe é presença de um elemento raiz, onde todos os outros elementos estão contidos nele. Desta forma um documento XML pode ser representado por uma árvore de elementos hierarquizados.

Um elemento é chamado *parent* quando existem elementos entre sua etiqueta de abertura e fechamento, estando um nível acima na estrutura hierárquica do arquivo XML. Conseqüentemente o elemento que possui um *parent* é definido como *child* e está um nível hierárquico abaixo. No caso de elementos que estejam no mesmo nível, estes são *siblings*.

A fim de exemplificar os conceitos apresentados acima, a Figura A.1 ilustra o arquivo MedFasee.xml, que descreve a estrutura básica de equipamentos do protótipo de medicação fasorial sincronizada do projeto MedFasee. Em seguida a Figura A.2 ilustra a respectiva representação em árvore, contendo as relações entre os elementos.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Estrutura de Equipamentos do Prototipo MedFasee -->
<SPMS>
  <PMUs_do_Sistema>
    <PMU idcode="1">UFSC</PMU>
    <PMU idcode="2">CEFET-PR</PMU>
    <PMU idcode="3">PUC-RS</PMU>
  </PMUs_do_Sistema>
  <PDCs_do_Sistema>
    <PDC idcode="1">PDCLABPAN</PDC>
  </PDCs_do_Sistema>
</SPMS>
```

Figura A.1 – Arquivo MedFasee.xml – Exemplo para estrutura do Projeto MedFasee.

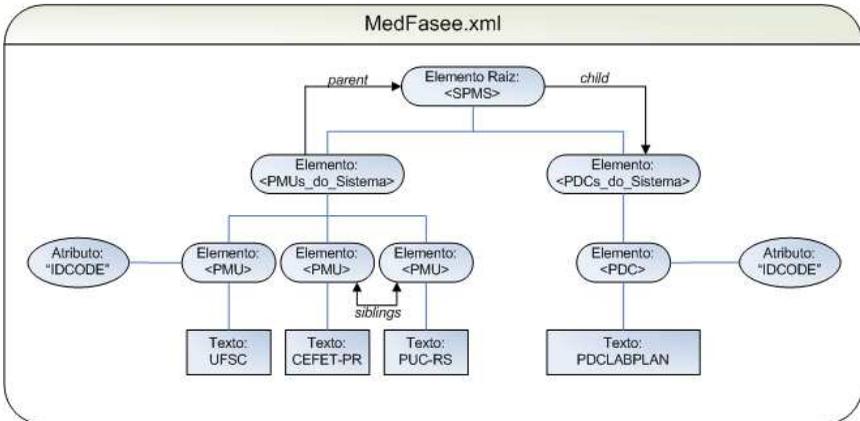


Figura A.2 – Estrutura em Árvore representando o arquivo MedFasee.xml.

#### A.1.4. Validação

Previamente foi definido que um documento possui “boa-formação” quando sua sintaxe está correta, considerando as regras estabelecidas pela recomendação do W3C. Entretanto, pode ocorrer a necessidade de que elementos tenham certas regras de formação, como por exemplo, constantes numéricas necessárias para um cálculo.

Desta maneira, fica claro que se o programador acidentalmente inserir um caractere nesta constante, sua aplicação terá comportamento indesejado e um erro de execução ocorrerá. Portanto nota-se a necessidade de um instrumento que estabeleça regras de validação para um determinado arquivo XML.

Atualmente, o W3C possui dois formatos padrão para validação de documentos, sendo eles o DTD (Document Type Definition) e W3C Schema. Desta forma, quando um arquivo XML de boa-formação for validado contra um DTD ou um Schema, ele será dito **Válido**.

#### A.1.5. Document Type Definition

O objetivo de uma DTD é especificar blocos de construção através de listas de elementos e atributos, permitindo moldar o arquivo às necessidades do programador e da aplicação, representando apenas os dados que se deseja transportar.

Uma DTD pode ser utilizada de duas maneiras. A primeira como arquivo separado, contendo sintaxe específica com extensão .dtd. A segunda adicionando o código ao início do arquivo XML.

Através de DTD, arquivos XML podem carregar descrições próprias, realizar verificações durante o desenvolvimento, permite que gru-

pos de desenvolvimento independente possam realizar trabalhos considerando um tipo padrão de definição de documento, compatibilizando a troca de dados. A seguir é apresentado um exemplo de DTD para o arquivo XML MedFasee.xml, mostrado na Figura A.3.

```
<!ELEMENT SPMS (PMUs_do_Sistema, PDCs_do_Sistema)>
<!ELEMENT PMUs_do_Sistema (PMU, PMU, PMU)>
<!ELEMENT PDCs_do_Sistema (PDC)>
<!ELEMENT PMU (#PCDATA)>
<!ELEMENT PDC (#PCDATA)>
<!ATTLIST PMU      idcode CDATA #REQUIRED>
<!ATTLIST PDC      idcode CDATA #REQUIRED>
```

Figura A.3 – Arquivo MedFasee.DTD. Exemplo de DTD para o arquivo MedFasee.xml.

Para utilizar o arquivo DTD externo, basta inseri-lo através de uma declaração no início do documento XML, como apresentado a seguir.

```
<!DOCTYPE SPMS SYSTEM "MedFasee.dtd">
```

Pode-se observar que para cada elemento deve-se especificar seu nome e o bloco de construção que ele receberá. Se existirem *children*, todos, sem exceção, deverão ser listados. Da mesma forma os atributos deverão todos ser declarados de acordo com o elemento de que fazem parte.

Apesar de sua forma sucinta, DTDs não são aninhadas como no caso de etiquetas XML, cabendo ao programador editar o arquivo de acordo com seu estilo a fim de facilitar a leitura e interpretação.

#### A.1.6. W3C Schema

Os Schemas são arquivos de extensão .xsd com o mesmo objetivo das DTDs, porém com diferentes e novas funcionalidades. A formação deste tipo de arquivo segue as mesmas regras de sintaxe impostas para a escrita de arquivos XML com boa-formação.

Schemas definem os nomes, atributos, tipo de texto, número e seqüência de *children*, valores fixos e padrão e principalmente tipos de dados facilitando assim a maneira de descrever o conteúdo do documento, validação da compatibilidade dos dados, entre outros.

Devido a esta atribuição, arquivos XSD possuem estrutura de maior complexidade e tamanho comparado aos arquivos aos quais representam. Por outro lado, comparado às DTDs sua interpretação é rela-

tivamente simples, pois possui estrutura menos ambígua (BENZ, B., *et al.*, 2003).

A referência a um arquivo Schema se dá através da adição de atributos ao elemento raiz do arquivo XML, como mostrado abaixo.

```
<Elemento_Raiz xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Schema_Arquivo.xsd">
  </Elemento_Raiz>
```

Primeiramente define-se o espaço de nome `xmlns:xsi` apontando o endereço <http://www.w3.org/2001/XMLSchema-instance>, endereço contendo um documento com a descrição de como o W3C Schema deve ser referenciado.

O segundo parâmetro `xsi:noNamespaceSchemaLocation` indica que não existem namespaces predefinidos para este arquivo, informando que os elementos do arquivo XML deverão ser validados contra um Schema especificado, neste caso o arquivo `Schema_Arquivo.xsd` presente no diretório raiz do arquivo XML.

A Figura A.4 ilustra a interpretação gráfica para o Schema do arquivo `MedFasee.xml`.

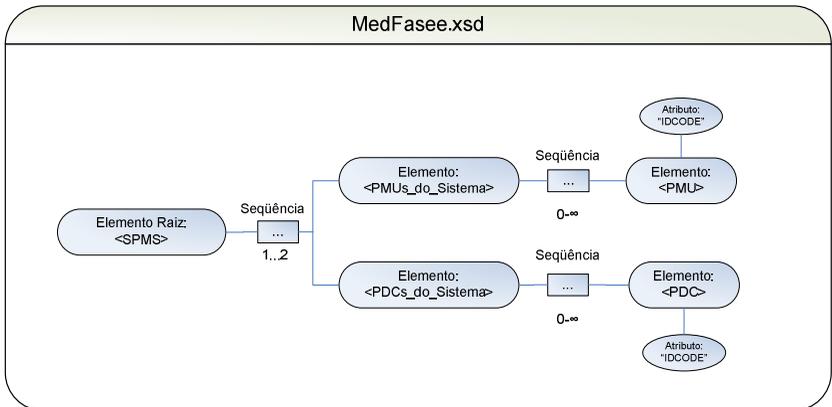


Figura A.4 – Representação Gráfica do arquivo `MedFasee.xsd`.

Elementos que possuem *children* não necessitam declarar todos aqueles que possuem nome repetido, como no caso das DTDs. O limite

pode ser definido através de um parâmetro compreendido no intervalo de 0 até  $\infty$ .

Importante salientar que a definição dos elementos em si não possui ordem específica, a não ser que este seja o desejo do programador, sendo que para a validação ocorrer, basta o arquivo XML corresponder à estrutura determinada pelo arquivo de extensão .xsd

W3C Schema está em processo de se tornar uma recomendação da W3C, tendo em vista o maior número de recursos disponíveis, passando assim a ser o sucessor das DTDs.

#### A.1.7. **Aplicação de XML no PDC**

O PDC possui parâmetros que necessitam de ajuste a partir de um arquivo de configuração externo ao arquivo executável do programa. Esta metodologia foi adotada a fim de tornar o Concentrador de Dados compatível com as particularidades de diferentes Sistemas de Medição Fasorial Sincronizada.

Parâmetros como números de identificação dos elementos do SPMS, endereços de rede específicos, taxa de envio de fasores, dados relativos ao(s) Banco(s) de Dados, são valores que varia de acordo com o sistema que se deseja instalar. Ao se adotar um arquivo externo para este fim, evita-se que toda modificação realizada no SPMS necessite uma nova compilação do código-fonte do PDC.

No desenvolvimento do PDC MedFasee, um arquivo ASCII simples descrevia tais parâmetros, sendo os valores de interesse separados do restante do arquivo através do caractere “:”. Desta forma, ao realizar a leitura dos dados, o código do PDC busca na linha respectiva este caractere, e tudo aquilo situado após ele é considerado um parâmetro de configuração. Na Figura A.5 a seguir é ilustrado um trecho do arquivo de configuração do PDC MedFasee, “pdc.cfg”.

```
IDCODE do PDC: 11
Tamanho da base historica local (dias): 30
Taxa de transmissao de fasores (frames/s): 60
Ativa operacao em tempo real (RTAI): 1
...
```

*Figura A.5 – Arquivo de Configuração pdc.cfg.*

Apesar da clareza com que os parâmetros são apresentados e a facilidade para leitura, existem dois aspectos principais que prejudicam sua utilização.

A primeira é em nível de desenvolvimento, onde para programar o acesso dos parâmetros, estes devem ser varridos de maneira seqüencial, não podendo ser ignorada nenhuma linha do arquivo, mesmo se esta não for utilizada.

O segundo aspecto é que nada impede do usuário que esteja configurando o arquivo, de cometer um erro de digitação, inserindo valores em formatos inconsistentes com o esperado pelo PDC. O erro só seria observado em tempo de execução, podendo tornar a tarefa de correção onerosa.

Da mesma forma, para configuração do PDC MedFasee, um arquivo separado seguindo as mesmas diretrizes do arquivo “*pdccfg*”, é utilizado para carregar os dados referentes às PMUs do sistema.

Sendo assim, a configuração do PDC era feita por meio de dois arquivos de texto ASCII distintos. No caso da expansão para a nova arquitetura hierárquica, seguindo a mesma lógica, um novo arquivo deveria ser adicionado, contendo os dados dos PDCs remotos que fazem parte da comunicação entre PDCs. Conseqüentemente, três arquivos distintos seriam destinados para executar tarefas similares.

Observa-se a necessidade de uma solução que pudesse integrar todos os dados de configuração em único arquivo, que sinalizasse erros no processo de preenchimento dos valores a serem lidos, acessando-os aleatoriamente e sem precisar varrer todos os elementos do arquivo.

Com estes aspectos em vista, aliando a potencialidade do XML, observou-se a oportunidade da utilização da linguagem para descrever os parâmetros de configuração do PDC.

Com a disponibilidade de uso do W3C Schema, o problema de formato de dados é resolvido imediatamente, pois não somente ele define a estrutura do arquivo XML como um todo, mas também especifica o tipo dos dados a serem incluídos em cada elemento.

Para a questão do acesso aleatório de dados e integração dos diferentes arquivos de configuração, o analisador sintático DOM se mostrou o mais apto a estas necessidades, pois através da criação de uma estrutura em árvore montada na memória permite que o arquivo XML seja percorrido na maneira que o programador necessite, na ordem que lhe seja conveniente. O arquivo de configuração XML do PDC, bem como seu respectivo arquivo Schema, estão disponibilizados no Apêndice B.

## A.2. Doxygen

Para dar início aos estudos do código fonte do PDC MedFasee, estudou-se a utilização de uma ferramenta que auxiliasse em dois aspectos fundamentais: a compreensão lógica dos desenvolvimentos (relacionamento entre classes e funções) e o entendimento técnico da escrita (atributos e variáveis e suas representações).

É comum para programadores que trabalham com extensos códigos fonte, que comentários sejam inseridos ao mesmo passo que as funcionalidades são implementadas. Esta estratégia permite que a documentação no presente possa orientar desenvolvimentos no futuro, dado que com o passar do tempo, conceitos utilizados ora no passado possam não ter sentido no futuro, quando determinado trecho de código seja revisitado e não esteja bem comentado.

Desta forma, entende-se que para os estudos e desenvolvimentos do PDC, é necessário utilizar uma ferramenta visual que possa orientar os estudos das funcionalidades, utilizando para isso os comentários já inseridos no código fonte do PDC MedFasee.

A ferramenta que satisfaz os requisitos acima citados é denominada Doxygen. Esta ferramenta é composta por um gerador de documentação de códigos fonte, que utiliza marcações inseridas nos comentários do programa, e as utiliza como referência para gerar os arquivos que compõe a documentação final.

O Doxygen é compatível com as linguagens de programação C, C++, Java, Fortran, C#, IDL, VHDL, PHP, Python, entre outros. Trata-se de um *software* livre, disponível para download em <http://www.stack.nl/~dimitri/doxygen/> juntamente com sua documentação. Seu desenvolvimento é realizado em plataforma GNU/Linux e Mac OS, porém portátil para uso em diversos sistemas operacionais como Windows e sistemas baseados em UNIX.

Como produto final, é possível obter a documentação gerada pelo aplicativo em diversos formatos de exibição, tais como: HTML, Latex, RTF, PostScript, páginas de manuais em formato UNIX e PDF com hyperlinks.

Para subsidiar o estudo do código fonte do PDC MedFasee, confluindo para posterior aplicação no código fonte do PDC, utilizou-se a saída no formato HTML, que permite a utilização de qualquer navegador padrão (Internet Explorer, Netscape, Mozilla Firefox, etc.) para leitura da documentação.

Ademais, as facilidades operacionais para navegação entre os métodos e classes, bem como a possibilidade de visualização gráfica cola-

boração entre classes, que faz parte da documentação HTML, são de importância didática relevante no processo de estudo do código fonte.

A Figura A.6 e a Figura A.7 ilustram a interface gráfica da aplicação do Doxygen sobre os arquivos contendo o código fonte do PDC.

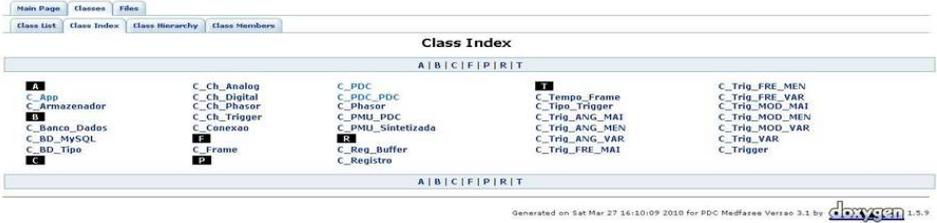


Figura A.6 – Índice de Classes gerado pela aplicação do Doxygen.

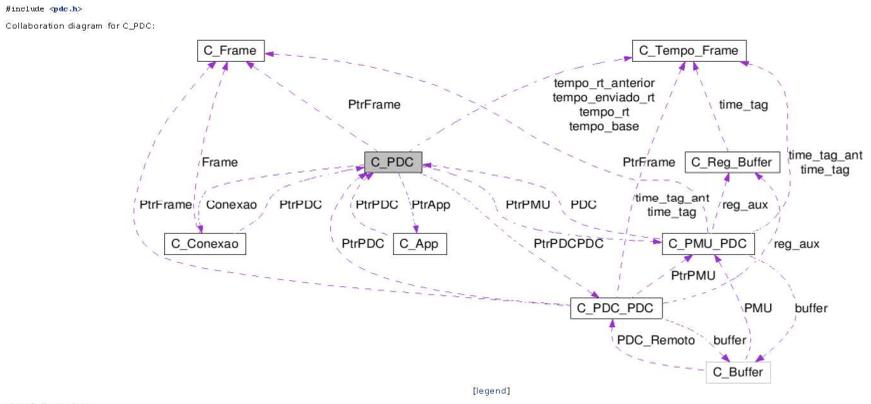


Figura A.7 – Diagrama de colaboração entre Classes gerado pelo Doxygen.

Percebe-se a forte funcionalidade visual da ferramenta. No caso da Figura A.6 cada classe representada na lista é um link, que uma vez acessado abre a descrição completa da respectiva classe. Já para a Figura A.7, pode ser feita uma visualização geral da colaboração entre classes, observando os atributos pelos quais é possível ser feita a troca de dados entre seus respectivos objetos.

### A.3. Linguagem de programação C++

A linguagem de programação C++ foi desenvolvida com base na linguagem C, porém com suporte a diversas ferramentas adicionais para o programador, bem como o suporte aos conceitos da MOO. Seu uso é

bastante difundido entre os desenvolvedores de *software* pelo mundo, o que dá origem a um grande volume de documentação sobre o assunto.

Em suma, a linguagem C++ provê a flexibilidade e facilidade do usuário desenvolvedor de definir classes. De maneira geral, para o C++, as classes são tipos de dados definidos pelo usuário, que permitem, além de outras funcionalidades, a aplicação dos conceitos de abstração e encapsulamento, fundamentais para o desenvolvimento de projetos baseados em MOO (STROUSTRUP, B., 2000).

O projeto de *software* do PDC faz uso das ferramentas disponibilizadas pela linguagem C++, sobretudo aquelas presentes na *Standard Template Library* (STL), biblioteca contendo classes contêineres, algoritmos, operadores e iteradores. Esta biblioteca tem por objetivo fornecer os alicerces para qualquer desenvolvimento na linguagem C++, inclusive para demais bibliotecas a serem desenvolvidas.

As implementações realizadas pela STL são focadas tanto na facilidade de uso de seus contêineres quanto pela otimização do uso de memória do equipamento e maximização da velocidade de processamento dos dados.

Portanto, procurou-se sempre que possível fazer uso dos elementos presentes na STL a fim de agregar poder de processamento ao PDC, que juntamente com as implementações de classes da linguagem C++, originaram um projeto de *software* robusto, expansível, modular e eficiente, compatível com o paradigma da Modelagem Orientada a Objetos.

# Apêndice B. Arquivos XML Utilizados no PDC

## B.1. Arquivo de Configuração do PDC ConfigPDC.xml

A seguir ilustra-se o arquivo de configuração XML desenvolvido para o PDC. Este arquivo contém os parâmetros configuráveis pelo usuário, de forma que as modificações que possam vir a ocorrer no âmbito do SPMS possam ser implementadas sem necessidade de recompilação do código-fonte.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
=====
Arquivo XML para a configuração do PDC

Projeto MedFasee
LabPlan - UFSC
http://www.medfasee.ufsc.br

Autores:
Alexandre Furstenberger
Leandro Percebon
MNAgostini
06/2009
=====
-->
<ConfigPDC xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="ConfigPDC.xsd">
  <!-- IDCODE do PDC - IEEE Std C37.118 -->
  <idcode_PDC>234</idcode_PDC>
  <!-- IP da interface de rede local do PDC - IP da placa de rede efetivamente utilizada-->
  <IP_local>150.162.19.214</IP_local>
  <parametros>
    <!-- Tamanho do buffer em HD, em segundos -->
    <tam_buff_HD>1</tam_buff_HD>
    <!-- Taxa de transmissao de sincrofases, em frames/s -->
    <taxa_fasores>60</taxa_fasores>
    <!-- Tempo de vida para declarar PMU desconectada, em segundos -->
    <tempo_vida_PMU>60</tempo_vida_PMU>
    <!-- Tempo de vida para declarar PDC Remoto desconectado, em segundos -->
    <tempo_vida_PDC>60</tempo_vida_PDC>
    <!-- Atraso mínimo e máximo para o processo de resincronizacao -->
    <atraso_ressinc>
      <min>300</min>
      <max>300</max>
    </atraso_ressinc>
  </parametros>
</ConfigPDC>
```

```

<flags>
<!-- Ativa/desativa a operacao do PDC em tempo real -->
<flag_tempo_real>1</flag_tempo_real>
<!-- Ativa/desativa o armazenamento de frames que chegam no PDC - opcao apenas para debug -->
<flag_armaz_frames_chegantes>0</flag_armaz_frames_chegantes>
<!-- Ativa/desativa a verificacao de triggers -->
<flag_verif_triggers>0</flag_verif_triggers>
<!-- Determina se este PDC tem por funcao Mestre ou Escravo -->
<flag_PDC_mestre>1</flag_PDC_mestre>
</flags>
<logs>
<!-- Ativa/desativa o log de perdas de cada PMU, individualmente -->
<flag_log_perdas_PMU>1</flag_log_perdas_PMU>
<!-- Ativa/desativa o log de perdas de dados resincronizados, em tempo real -->
<flag_log_perdas_dados_RT>1</flag_log_perdas_dados_RT>
<!-- Ativa/desativa o log de disponibilizaçao de dados resincronizados, em tempo real -->
<flag_log_dados_RT>0</flag_log_dados_RT>
<!-- Ativa/desativa o log de atrasos de cada PMU, individualmente -->
<flag_log_atraso_PMUs>0</flag_log_atraso_PMUs>
<!-- Ativa/desativa o log de atrasos de cada PDC, individualmente -->
<flag_log_atraso_PDCs>1</flag_log_atraso_PDCs>
<!-- Ativa/desativa o log de atrasos relativos entre PMUs (em frames) -->
<flag_log_atraso_relativo>0</flag_log_atraso_relativo>
<!-- Ativa/desativa o log de insercoes do armazenador -->
<flag_log_insercoes_armaz>1</flag_log_insercoes_armaz>
</logs>
<!-- Lista de bancos de dados do PDC -->
<lista_BD>
<BD>
<!-- Banco de dados 1 do PDC - normalmente, mas não necessariamente, o local -->
<idcode>1</idcode>
<!-- Tipo de banco de dados.
      0 -> estrategia nao atribuida
      1 -> MySQL (padrao)
      x -> outros (implementacoes futuras) -->
<tipo>1</tipo>
<!-- Flag de uso do banco de dados:
      0 -> nao usa o BD local
      1 -> armazena apenas os dados RT
      2 -> armazena apenas os dados históricos
      3 -> armazena dados RT e históricos (padrao) -->
<flag_uso_BD>3</flag_uso_BD>
<!-- IP do banco de dados
      | padrao: localhost -->
<IP_BD>localhost</IP_BD>
<!-- Usuario para a conexao -->
<usuario>u_medfase</usuario>
<senha>pdcteep</senha>
<!-- Nome da base de dados no servidor do banco -->
<nome_base>db_smf</nome_base>
<!-- Tamanho da base de dados, em dias -->
<tam_base_hist>30</tam_base_hist>
</BD>
<BD>
<idcode>2</idcode>
<tipo>1</tipo>
<flag_uso_BD>0</flag_uso_BD>
<IP_BD>150.162.19.129</IP_BD>
<usuario>u_medfase</usuario>
<senha>pdcteep</senha>
<nome_base>db_smf</nome_base>
<tam_base_hist>30</tam_base_hist>
</BD>
</lista_BD>
<!-- Lista de PMUs que o PDC deve conectar -->
<lista_PMU>
</lista_PMU>
<!-- Lista de PDCs que o PDC deve se comunicar -->
<lista_PDC>
<PDC>
| | | | <idcode_PDC>233</idcode_PDC>

```

```
<!-- Tamanho da base de dados, em dias -->
<tam_base_hist>30</tam_base_hist>
</BD>
<BD>
  <idcode>2</idcode>
  <tipo>1</tipo>
  <flag_uso_BD>0</flag_uso_BD>
  <IP_BD>150.162.19.129</IP_BD>
  <usuario>u_medfase</usuario>
  <senha>pdccateep</senha>
  <nome_base>db_smf</nome_base>
  <tam_base_hist>30</tam_base_hist>
</BD>
</lista_BD>
<!-- Lista de PMUs que o PDC deve conectar -->
<lista_PMU>
</lista_PMU>
<!-- Lista de PDCs que o PDC deve se comunicar -->
<lista_PDC>
<PDC>
  <idcode_PDC>233</idcode_PDC>
  <IP_PDC>150.162.19.213</IP_PDC>
</PDC>
</lista_PDC>
<!-- Lista de Aplicacoes que o PDC deve enviar frames de dados C37.118 -->
<lista_App>
<App>
  <IP_App>150.162.19.125</IP_App>
  <!-- Este tempo (em segundos) determina o intervalo em que o PDC mandara um arquivo de CFG para o ip da Aplicacao -->
  <tempo_envio_config>30</tempo_envio_config>
  <!-- Tipo de frame que a aplicacao espera receber. Preencher com 1 para C37.118 -->
  <tipo_frame>1</tipo_frame>
</App>
</lista_App>
</ConfigPDC>
```

## B.2. Arquivo Schema ConfigPDC.xsd

A seguir ilustra-se o arquivo XML Schema que serve de base para a formação do arquivo de configuração XML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2009 sp1 (http://www.altova.com) by firsten (EMBRACE) -->
<!-- =====
Esquema do arquivo XML para a configuração do PDC

Projeto MedFasee
LabPlan - UFSC
http://www.medfasee.ufsc.br

Autores:
Alexandre Furstenberger
Leandro Percebon
MNAgostini
05/2009
----->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="ConfigPDC">
    <xs:complexType>
      <xs:sequence>
        <!-- IDCODE do PDC - IEEE Std C37.118 -->
        <xs:element name="idcode_PDC" type="xs:unsignedInt"/>
        <xs:element name="IP_local" type="xs:string"/>
        <xs:element name="parametros">
          <xs:complexType>
            <xs:sequence>
              <!-- Tamanho do buffer em HD, em segundos -->
              <xs:element name="tam_buff_HD" type="xs:unsignedInt" default="1"/>
              <!-- Taxa de transmissao de sincrofases, em frames/s -->
              <xs:element name="taxa_fases" type="xs:unsignedInt" default="60"/>
            
```

```

<!-- Tempo de vida para declarar PMU desconectada, em segundos -->
<xs:element name="tempo_vida_PMU" type="xs:unsignedInt" default="10"/>
<!-- Atrasos minimo e maximo para o processo de resincronizacao -->
<xs:element name="tempo_vida_PDC" type="xs:unsignedInt" default="10"/>
<xs:element name="atraso_ressinc">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="min" type="xs:unsignedInt" default="0"/>
      <xs:element name="max" type="xs:unsignedInt" default="15"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="flags">
  <xs:complexType>
    <xs:sequence>
      <!-- Ativa/desativa a operacao do PDC em tempo real -->
      <xs:element name="flag_tempo_real" type="xs:boolean" default="1"/>
      <!-- Ativa/desativa o armazenamento de frames que chegam no PDC - opcao apenas para debug -->
      <xs:element name="flag_armaz_frames_chegantes" type="xs:boolean" default="0"/>
      <!-- Ativa/desativa a verificacao de triggers -->
      <xs:element name="flag_verif_triggers" type="xs:boolean" default="0"/>
      <xs:element name="flag_PDC_mestre" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="logs">
  <xs:complexType>
    <xs:sequence>
      <!-- Ativa/desativa o log de perdas de cada PMU, individualmente -->
      <xs:element name="flag_log_perdas_PMU" type="xs:boolean" default="1"/>
      <!-- Ativa/desativa o log de perdas de dados resincronizados, em tempo real -->
      <xs:element name="flag_log_perdas_dados_RT" type="xs:boolean" default="1"/>
      <!-- Ativa/desativa o log de disponibilização de dados resincronizados, em tempo real -->
      <xs:element name="flag_log_dados_RT" type="xs:boolean" default="1"/>
      <!-- Ativa/desativa o log de atrasos de cada PMU, individualmente -->
      <xs:element name="flag_log_atraso_PMUs" type="xs:boolean" default="1"/>
      <!-- Ativa/desativa o log de atrasos de cada PDC, individualmente -->
      <xs:element name="flag_log_atraso_PDCs" type="xs:boolean" default="1"/>
      <!-- Ativa/desativa o log de atrasos relativos entre PMUs (em frames) -->
      <xs:element name="flag_log_atraso_relativo" type="xs:boolean" default="1"/>
      <!-- Ativa/desativa o log de insercoes do armazenador -->
      <xs:element name="flag_log_insercoes_armaz" type="xs:boolean" default="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="lista_BD">
  <xs:complexType>
    <xs:sequence>
      <!-- Banco de dados 1 do PDC -->
      <xs:element name="BD" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <!-- Flag de uso do banco de dados:

```

```

<xs:complexType>
  <xs:sequence>
    <!-- Flag de uso do banco de dados:
    0 -> nao usa o BD local
    1 -> armazena apenas os dados RT
    2 -> armazena apenas os dados históricos
    3 -> armazena dados RT e históricos -->
    <xs:element name="idcode" type="xs:unsignedInt"/>
    <xs:element name="tipo" type="xs:unsignedInt" default="1"/>
    <xs:element name="flag_uso_BD" type="xs:unsignedInt" default="3"/>
    <!-- IP do banco de dados -->
    <xs:element name="IP_BD" type="xs:string" default="localhost"/>
    <!-- Usuario para a conexao -->
    <xs:element name="usuario" type="xs:string"/>
    <xs:element name="senha" type="xs:string"/>
    <!-- Nome da base de dados no servidor do banco -->
    <xs:element name="nome_base" type="xs:string" default="db_smf"/>
    <!-- Tamanho da base de dados, em dias -->
    <xs:element name="tam_base_hist" type="xs:unsignedInt" default="15"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
<!-- Banco de dados 1 do PDC -->
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- Lista de PMUs que o PDC deve conectar -->
<xs:element name="lista_PMU">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PMU" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <!-- IDCODE da PMU -->
            <xs:element name="idcode_PMU" type="xs:unsignedInt"/>
            <!-- IP da PMU -->
            <xs:element name="IP_PMU" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!-- Lista de PDCs que o PDC deve conectar -->
<xs:element name="lista_PDC">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="PDC" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <!-- IDCODE do PDC -->
            <xs:element name="idcode_PDC" type="xs:unsignedInt"/>
            <!-- IP do PDC -->
            <xs:element name="IP_PDC" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>

```

```
<!-- IP do PDC -->
<xs:element name="IP_PDC" type="xs:string"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
<!-- Lista de Aplicacoes que o PDC deve enviar frame de dados -->
<xs:element name="lista_App">
<xs:complexType>
<xs:sequence>
<xs:element name="App" minOccurs="0" maxOccurs="unbounded">
<xs:complexType>
<xs:sequence>
<!-- IP da Aplicacao -->
<xs:element name="IP_App" type="xs:string"/>
<!-- Tempo que determina o intervalo de reenvio do frame de configuracao C37
<xs:element name="tempo_envio_config" type="xs:unsignedInt"/>
<xs:element name="tipo_frame" type="xs:unsignedInt" default="1"/>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```



## Apêndice C. Comunicação Multicast

Sincrofasores em um SPMS são enviados das PMUs ao PDC através de protocolos de comunicação IP, empacotados em um frame no formato descrito pela norma IEEE C37.118.

Para o caso do PDC proposto, a fim de manipular as conexões entre equipamentos e o envio/recebimento de pacotes pela Internet, é utilizada a API de Sockets, também conhecido como “Berkeley Sockets” (STEVENS, W. R., *et al.*, 2004). Esta interface provê ferramentas para desenvolvedores para programar rotinas destinadas à redes de comunicação, utilizando protocolos de entrega *Transmission Control Protocol* (TCP) e *User Datagram Protocol* (UDP).

A comunicação TCP, por se tratar de um protocolo que realiza conexões entre os processos que enviam/recebem dados, dispõe de diversos instrumentos para obter informação sobre o tráfego de dados na rede, permitindo o gerenciamento de conexões, entrega ordenada de pacotes, confiabilidade e controle de congestionamento.

Paralelamente, comunicação UDP não exige conexão entre as partes envolvidas, e disponibiliza ao programador números de portas. Com este parâmetro, é possível a realização de multiplexagem e demultiplexagem. Isto significa que números de portas permitem que diversas aplicações possam manter seus pacotes de dados separados utilizando a mesma camada de transporte (MAKOFKSKE, D., *et al.*, 2003).

Entretanto, apesar de não haver a garantia de entrega de dados nem a gama de recursos providos pelo TCP, a comunicação UDP tem por característica a maior velocidade no transporte de dados. Este aspecto é de extrema importância para o SPMS, devido à taxa de amostragem de até 60 sincrofasores por segundo. Para o caso do PDC, que recebe essa taxa de frames por PMU e necessita tratar os dados em tempo real, é preferível a velocidade do protocolo UDP, embora este acarrete em perdas na transmissão de dados.

Tendo isto em vista, as seções a seguir têm por objetivo introduzir os conceitos de comunicação em redes com os paradigmas Unicast, Broadcast e Multicast, descrevendo e discutindo seus principais aspectos e sua aplicação em Sistemas de Medição Fasorial Sincronizada, especi-

almente no que se refere ao sistema de aquisição de dados, ou seja, o PDC.

Neste apêndice são apresentados e discutidos alguns aspectos sobre comunicação em redes no que tange o paradigma de transporte de dados sobre a rede, apresentando as características principais de Unicast, Broadcast e Multicast, dando ênfase a este último, por se tratar de um conceito atual e ainda em aberto para discussões no meio científico.

### C.1. Unicast

Comunicação Unicast trata-se da difusão de dados ponto-a-ponto. Um processo envia informação através da rede com destino único, identificando o receptor através do seu respectivo endereço IP.

Desta forma, se um determinado servidor necessita enviar mensagens para diversos clientes, uma cópia destes dados será realizada para cada processo e enviada individualmente. Percebe-se que em redes que possuem um grande número de usuários, é natural a ocorrência de congestionamentos e colisões de dados durante o transporte.

Protocolos TCP só suportam Unicast, ao passo que UDP permite o uso de diferentes paradigmas. A Figura C.1 ilustra um exemplo de comunicação unicast em uma rede de vários usuários.

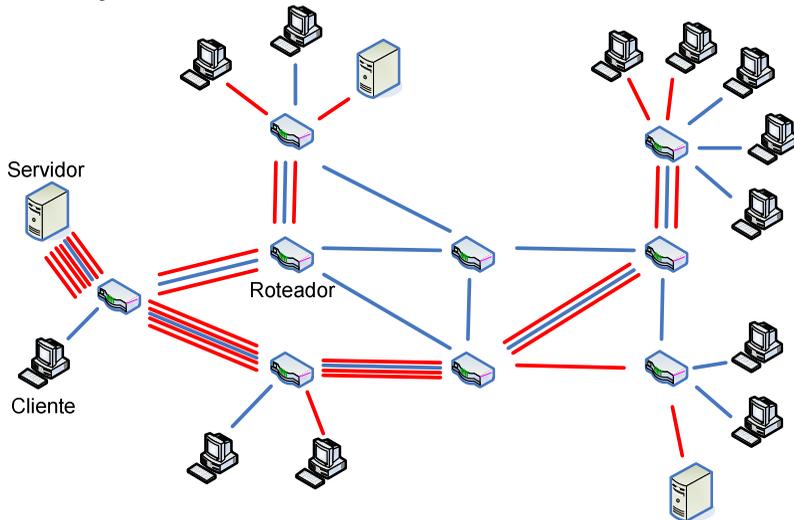


Figura C.1 – Rede comunicando-se em Unicast – Adaptado de (MAKOFKSKE, D., et al., 2003).

Percebe-se que em processos onde o fluxo de dados é requerido intensivamente, caso a largura de banda projetada para a rede for excedida, atrasos, congestionamentos e colisões de pacotes poderão acontecer, mesmo que no lado do cliente a rede não esteja em alta demanda.

## C.2. Broadcast

Broadcast define a difusão de mensagens para todos os elementos contidos na rede, sem exceção. O processo envia o pacote de dados Unicast para o endereço broadcast da rede, que se encarrega de entregar aos endereços finais, sendo que a camada de transporte que suporta tal comunicação é o UDP (STEVENS, W. R., *et al.*, 2004).

Em redes que não possuem sub-redes, o endereço IP broadcast é sempre o maior endereço IP possível dentro de uma LAN. Por exemplo, supondo uma LAN cujo IP é dado por 145.165.2.0, o endereço Broadcast é dado por 145.165.2.255.

Esta definição pode ser estendida para um escopo maior de endereços na LAN que se está trabalhando. Considerando, por exemplo, a rede de IP 145.0.0.0, pode-se enviar uma mensagem broadcast para toda a rede através do endereço 145.255.255.255. Normalmente, roteadores ao receber um datagrama externo com destino ao endereço broadcast, descartam a mensagem.

A Figura C.2 ilustra o envio de um datagrama broadcast, considerando o escopo de uma LAN.

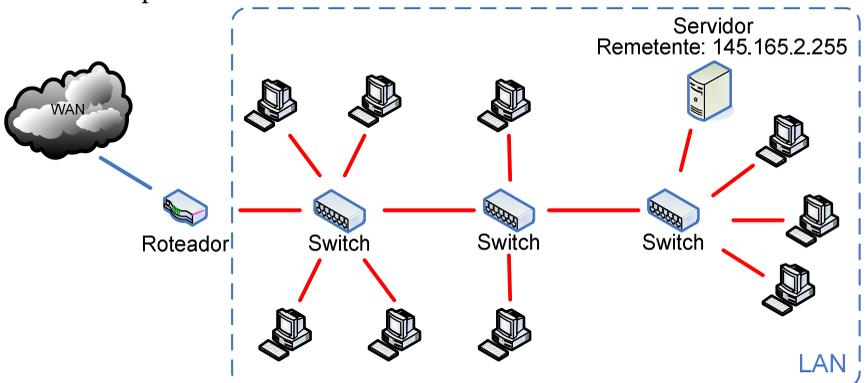


Figura C.2 – Broadcast em uma LAN.

A utilização de comunicação broadcast possui algumas aplicações, por exemplo: assumindo que exista um servidor em uma LAN sendo necessária a obtenção do seu endereço de IP. Em alguns casos, quando existem diversos clientes conectando-se ao servidor, a utilização da rede pode ser otimizada através do uso de datagramas broadcast.

Alguns protocolos que podem ser citados pelo uso de broadcast são: *Address Resolution Protocol (ARP)*, *Bootstrap Protocol (BOOTP)*, *Network Time Protocol (NTP)* (MOGUL, J., 1984; STEVENS, W. R., *et al.*, 2004).

### **C.3. Multicast**

A comunicação Multicast situa-se na posição intermediária entre Unicast e Broadcast. Enquanto Unicast a comunicação é feita ponto-a-ponto, Broadcast inclui todos os elementos dentro do escopo da rede que se está trabalhando. Multicast também se trata de um método de comunicação que abrange múltiplos destinatários, porém não todos.

Em essência, o que se espera é que apenas processos interessados em determinada mensagem sejam capazes de recebê-la, sendo que o remetente necessita realizar este envio apenas uma vez, deixando que o trabalho de cópia e encaminhamento seja depositado na estrutura da rede (roteadores e switches).

Desta forma, procura-se diminuir a carga de dados transmitidos por aplicações que envolvem um grande número de destinatários ao mesmo tempo. Da mesma maneira que o Broadcast, Multicast utiliza a camada de transporte UDP.

Para tornar possível a comunicação Multicast, uma funcionalidade necessita estar presente durante este processo, chamada de “árvore de encaminhamento Multicast”, responsável pela ponte entre o processo remetente e as aplicações, de acordo com a Figura C.3.

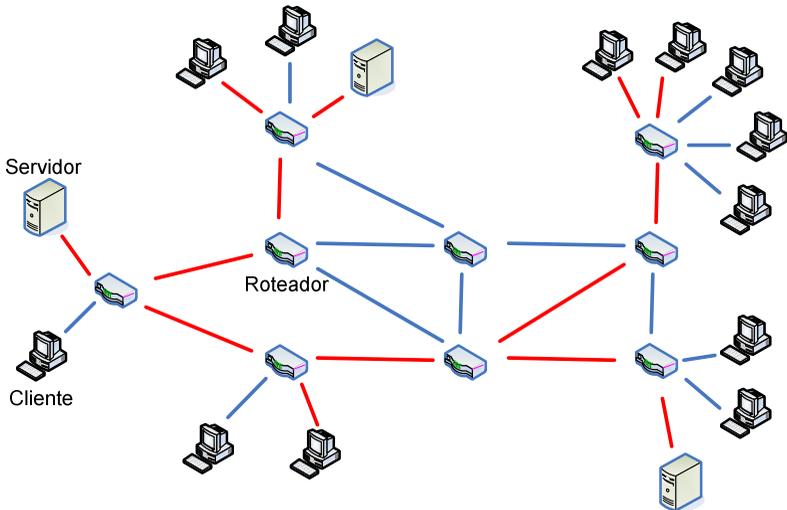


Figura C.3 – Árvore de encaminhamento Multicast – Adaptado de (MAKOF-SKE, D., et al., 2003).

Neste caso, o Servidor está no papel de remetente e posicionado na raiz da árvore, enviando apenas um datagrama por vez. Este dado é copiado e encaminhado pelos roteadores e switches da rede, apenas para os processos interessados em recebê-lo.

Nota-se a flexibilidade e a minimização do tráfego de dados na rede, bem como no esforço reduzido do servidor para enviar seus dados. Multicast também possui característica de escalabilidade, ou seja, habilidade de manipular de maneira estável e constante a demanda crescente requerida do sistema.

Entretanto, estes benefícios adicionam complexidade tanto para a rede, que necessita de equipamentos dedicados para o envio/recebimento de dados, quanto para o programador que desenvolve programas com fins de utilizar as funcionalidades do Multicast (MAKOF-SKE, D., et al., 2003).

#### C.4. Escopo de Endereços Multicast

Para os desenvolvimentos e testes da comunicação Multicast em IPv4, IPs específicos são designados, denominados Classe D. Esta classe abrange todos os IPs do intervalo de 224.0.0.0 até 239.255.255.255.

Um IP compreendido dentro deste intervalo identifica um **Grupo Multicast**. Toda vez que uma aplicação deseja enviar uma mensagem

Multicast, o IP de destino será um grupo, iniciando uma sessão Multicast, e processos interessados em receber a mensagem deverão requerer sua entrada, se tornando **membros**.

### **C.5. Protocolo IGMP**

Para efetuar a entrada em um grupo Multicast, o processo que quer se tornar membro necessita informar à rede o interesse de participar da sessão e receber os dados. Para isto é usado o protocolo IGMP (*Internet Group Management Protocol*), que realiza a interface de comunicação entre o sistema operacional da aplicação e o roteador mais próximo pertencente à árvore Multicast. Desta forma são efetuadas operações de gerenciamento no grupo, controlando a entrada e saída de membros.

Desde o início da concepção do protocolo, três versões foram publicadas, a citar: IGMP (DEERING, S., 1989), IGMPv2 (FENNER, W., 1997) e IGMPv3 (CAIN, B., *et al.*, 2002). Este último é projetado para substituir os anteriores, porém sem perder interoperabilidade com os mesmos.

A utilização de um protocolo em detrimento de outro depende do tipo do modelo de serviço Multicast utilizado, que podem ser dois: *Any Source Multicast* (ASM) ou *Source Specific Multicast* (SSM). ASM utiliza o protocolo IGMPv2 e até a publicação da terceira versão, este modelo era o único suportado para aplicações Multicast.

As características e diferenças destes dois modelos de serviço são explicitadas nas seções a seguir.

#### ***Any Source Multicast***

Neste modelo, para se tornar parte do grupo, o processo interessado necessita apenas enviar seu pedido de entrada e, tão logo for concluído, começará a receber os dados enviados para esta sessão. Da mesma forma, um processo que deseja transmitir seus dados para um grupo, basta enviá-los ao endereço IP que o representa.

Assim sendo, a aplicação receberá todos os dados enviados àquela sessão, independente do processo remetente. Em outras palavras a difusão de dados pode ser tanto de um processo para muitos receptores, quanto de muitos processos para muitos receptores.

Como consequência direta deste modelo de serviço, pode-se citar colisões de endereços de grupo. No caso de dois processos independentes venham, ao acaso, enviarem seus dados a um mesmo grupo Multicast, estes são recebidos igualmente pelas aplicações. Desta forma, não há possibilidade para filtragem de remetentes.

### ***Source Specific Multicast***

Este modelo considera que além da informação do grupo, a aplicação também forneça o endereço IP do remetente ao qual deseja receber pacotes de dados Multicast.

Desta forma existe o par (Remetente, Grupo), definindo um “canal”. Esta é a principal característica que difere SSM do ASM. Sempre que uma aplicação entrar em um canal, ela explicita o desejo de receber datagramas Multicast apenas de determinado remetente. Qualquer outro processo que deseje enviar dados para o grupo terá seus dados descartados, dado que este processo estará formando um novo canal Multicast.

A fim de distinguir processos que trabalham com modelo SSM, um intervalo dedicado de IPs dentro da classe D é disponibilizado, que vai de 232.0.0.0 até 232.255.255.255.

As principais vantagens associadas a este modelo são:

- Filtragem de remetentes
- Prevenção de colisão de endereços
- Redução de infra-estrutura
- Otimização de roteamento

Diretrizes normativas para o uso e desenvolvimento de SSM para protocolo IP são apresentadas em (HOLBROOK, H., *et al.*, 2006), sendo que a condição necessária para sistemas que desejem utilizar SSM é a compatibilidade com o protocolo IGMPv3.

### **C.6. API para filtragem de remetentes**

Considerando aspectos de programação, as diferenças entre ASM e SSM são pequenas, bastando o programador informar o endereço IP do remetente o qual se deseja receber pacotes multicast. Isto é feito através de opções passadas para a API de sockets em sistemas que suportam IGMPv3.

Em (THALER, D., *et al.*, 2004) são determinadas extensões na interface de sockets com objetivo de controlar os remetentes de um processo Multicast, destacando-se duas APIs: Basic (Delta-based) e Advanced (Full-state).

#### ***Basic (Delta-based) API***

Esta API permite a inclusão ou remoção (filtragem) de remetentes de maneira incremental (deltas), ou seja, um de cada vez. Esta simplicidade pode ser requerida dependendo do tipo de aplicação envolvida, pois sua programação é simplificada.

Neste escopo existem dois modos de operação do filtro: **exclusivo** e **inclusivo**. Para o modo exclusivo, em princípio, todos os remetentes são aceitos para enviar dados para um determinado grupo multicast. O filtro então age excluindo um a um aqueles remetentes que não se deseja receber dados. Esta operação define o ASM, previamente comentado.

Por outro lado, o modo inclusivo opera inicialmente excluindo todos os remetentes de um determinado grupo, posteriormente adicionando um a um aqueles em que se quer receber mensagens Multicast. Este modo, portanto, define o SSM, como visto anteriormente.

Para manipular o filtro utiliza-se a função `setsockopt()`, presente nas API de sockets para programação em redes, que para sistemas UNIX está presente na biblioteca `sys/socket.h`.

### ***Advanced (Full-state) API***

Assim como a Delta-based API, Full-state API opera com os modos exclusivo e inclusivo. Porém, sua estrutura permite configurar o filtro de uma sessão Multicast a partir de uma lista de remetentes, adicionando-os ou removendo-os com uma simples chamada de função. Aplicações que necessitem trocar o modo de operação sem sair da sessão, ou precisem manipular uma grande quantidade de remetentes devem usar esta API (THALER, D., *et al.*, 2004).

Diferentemente do caso anterior, a função utilizada para manipular o socket é `ioctl()`, função de manipulação de eventos de entrada e saída (I/O) do sistema operacional, utilizada quando uma determinada operação não se encaixa completamente em nenhuma outro conjunto de funções. Esta função está presente na biblioteca `sys/ioctl.h` para sistemas UNIX.

## **C.7. Aplicação de Multicast em SPMS**

Conceitualmente é notória a aplicabilidade dos conceitos da comunicação Multicast em SPMS. O principal benefício da utilização deste paradigma está na otimização do intenso fluxo de dados favoriais que compõe a principal característica desta tecnologia, a alta taxa de amostragem.

Desta forma, procurou-se desenvolver o PDCS compatível com os conceitos apresentados previamente, sendo que a externalização de dados em tempo real pode ser feita para sessões Multicast.

Entretanto, um aspecto importante deve ser salientado. Devido à incipiência da tecnologia Multicast, juntamente com o constante desenvolvimento de novas tecnologias e padrões que comportem tais desenvolvimentos, o número de dispositivos presentes nas redes utilizadas atualmente é escasso.

De fato, roteadores compatíveis com o protocolo IGMPv3 normalmente possuem configurações de fábrica apontadas para o bloqueio de mensagens Multicast, e estas configurações são mantidas ao longo de seu uso em redes públicas. Estes fatores dificultam a utilização de Multicast fora de redes locais de computadores, se tratando dos principais fatores limitantes para a implementação de aplicações compatíveis com esta tecnologia.

Sendo assim, para aplicações em SPMS, é visto a necessidade de redes dedicadas, devidamente configuradas, para a correta circulação dos pacotes de dados e, conseqüentemente, o bom uso dos benefícios da comunicação Multicast.



# Referências Bibliográficas

- BAHRGAVA, B. "Synchronized Phasor Measurement System Project at Southern California Edison Co." In *IEEE Power Engineering Society Summer Meeting*, 1999.
- BALASIU, F., LAZAR, F. M., BALAURESCU, R. "Automatic Security Measures Foreseen in the Romanian Electrical Power System in Case of Major Disturbances." In *IEEE Bucharest Power Tech Conference*, 2009.
- BALLANCE, J. W., BHARGAVA, B., RODRIGUEZ, G. D. "Monitoring Power System Dynamics Using Phasor Measurement Technology for Power System Dynamic Security Assessment." In *IEEE Power Tech*. Bologna, 2003.
- BEGOVIC, M., NOVOSEL, D., DJOKIC, B. "Issues Related to the Implementation of Synchrophasor Measurements." In *41st Hawaii International Conference on System Sciences*, 2008.
- BENZ, B., DURANT, J. R. *Xml Programming Bible*: Wiley Publishing Inc, 2003.
- BI, T. "Wams Implementation in China: Part I Current Status - Naspi Presentation." In *NASPI Work Group Meeting June 8-9, 2010 - disponível em <http://www.naspi.org> - último acesso em 29/08/2010*, 2010.
- BIN, S. K., ABDULLAH, S., YUSUF, N. S. B. N. "Tenaga Nasional Berhad Wide Area Measurement System Based Applications." In *2nd IEEE International Conference on Power and Energy* 2008.
- CAIN, B., DEERING, S., KOUVELAS, I., FENNER, B., THYAGARAJAN, A. "Rfc 3376 - Internet Group Management Protocol, Version 3." 2002.
- CHENINE, M., ZHU, K., NORDSTROM, L. "Survey on Priorities and Communication Requirements for Pmu-Based Applications in the Nordic Region." In *IEEE Bucharest Power Tech Conference*, 2009.
- CIGRÉ. "Wide Area Monitoring and Control for Transmission Capability Enhancement." Working Group C4.601, 2007.
- DAGLE, J. "North American Synchrophasor Initiative - an Update of Progress." In *42nd Hawaii International Conference on System Sciences*, 2009.
- DECKER, I. C., AGOSTINI, M. N., DOTTA, D., ZIMATH, S. L. "Sistema De Medição Fasorial Sincronizada: Desenvolvimento E Aplicação

- Na Monitoração De Frequência Do Sistema Elétrico Brasileiro." In *XVIII Seminário Nacional de Produção e Transmissão de Energia Elétrica*. Curitiba, 2005.
- DECKER, I. C., AGOSTINI, M. N., MEYER, B. T., TEODORO, F. H., DOTTA, D., ZIMATH, S. L. "Estudo E Avaliação Do Desempenho De Um Protótipo De Sistema De Medição Fasorial Sincronizada Instalado No Sistema De Transmissão Da Eletrosul." Paper presented at the XX Seminário Nacional de Produção e Transmissão de Energia Elétrica - SNPTEE 2009.
- DECKER, I. C., DOTTA, D., AGOSTINI, M. N., SILVA, A. S. E., MEYER, B. T., ZIMATH, S. L. "Installation and Monitoring Experiences of the First Synchronized Measurement System in the Brazilian National Grid." In *Transmission and Distribution Conference and Exposition: Latin America: IEEE/PES*, 2008.
- DECKER, I. C., DOTTA, D., AGOSTINI, M. N., ZIMATH, S. L., SILVA, A. S. E. "Performance of a Synchronized Phasor Measurements System in the Brazilian Power System." In *IEEE PES General Meeting 2006*. Canada, 2006.
- DECKER, I. C., EHRENSPERGER, J. G., AGOSTINI, M. N., SILVA, A. S. E., BETTIOL, A. L., ZIMATH, S. L. "Synchronized Phasor Measurement System: Development and Applications." In *IX Simpósio de Especialistas em Planejamento da Operação e Expansão Elétrica - SEPOPE* Rio de Janeiro, 2004.
- DEERING, S. "Rfc 1112 - Host Extensions for Ip Multicasting." 1989.
- DING, X., LITTLER, T., MORROW, J., CROSSLEY, P. A., TERZIJA, V., O'MALLEY, M. "Synchronized Phasor Measurement on the All-Ireland Electrical Network." In *IEEE Lausanne Power Tech*, 2007.
- EHRENSPERGER, J. G. "Sistemas De Medição Fasorial Sincronizada: Análise Do Estado Da Arte E Aplicações No Monitoramento De Sistemas De Energia Elétrica." Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2004.
- EIPP. "Eastern Interconnection Phasor Project." Paper presented at the IEEE PES Power Systems Conference and Exposition, 2006. PSCE '06. 2006.
- EPRI. "The Green Grid. Energy Savings and Carbon Emissions Reductions Enabled by a Smart Grid. Technical Update June 2008." 2008.
- ESA. "European Space Agency - Disponível Em [Http://Www.Esa.Int/Esacp/Index.Html](http://www.esa.int/Esacp/Index.html) - Último Acesso Em 29/08/2010." 2010.
- ETP. "European Technology Platform for the Electricity Networks of the Future - [Http://Www.Smartgrids.Eu](http://www.smartgrids.eu) - Último Acesso Em 15 De Novembro De 2010." (2010).

- FALCÃO, D. M. "Smart Grids E Microredes: O Futuro Já É Presente." In *VIII SIMPASE*, 2009.
- FENNER, W. "Rfc 2236 - Internet Group Management Protocol, Version 2." 1997.
- FOSTER, T. "Measurement Tools at Heart of Smart Grid Need Calibration to Ensure Reliability." *Electric Light & Power - disponível em <http://www.elp.com>* 2010.
- FOWLER, M., SCOTT, K. *Uml Essencial*. 2 Edição ed: Bookman, 2000.
- GPS. "Global Positioning System - Disponível Em <Http://Www.Gps.Gov/Systems/Gps/> - Último Acesso Em 29/08/2010." 2010.
- HOLBROOK, H., B.CAIN. "Rfc 4607 - Source-Specific Multicast for Ip." 2006.
- HOROWITZ, S. H., PHADKE, A. G., RENZ, B. A. "The Future of Power Transmission." *IEEE Power & Energy Magazine* 2010.
- HU, Y., NOVOSEL, D. "Considerations and Requirements of Pdc's for Wampac Systems - Presentantion." In *Brazil CIGRE PDC Workshop*, 2010.
- HUANG, Z., HAUER, J. F., MARTIN, K. E. "Evaluation of Pmu Dynamic Performance in Both Lab Environments and Field Operating Conditions." Paper presented at the IEEE Power Engineering Society General Meeting 2007.
- IEEE. "1344 Standard for Synchrophasors for Power Systems." 1995.
- IEEE. "C37.118 Standard for Synchrophasors for Power Systems ", 2005.
- IRIG, R. C. C. "Irig - Inter-Range Instrumentation Group Timecodes - Standard 200-04." 2004.
- KARLSSON, D., HEMMINGSSON, M., LINDAHL, S. "Wide Area System Monitoring and Control." *IEEE Power & Energy Magazine* 2004.
- LEIRBUKT, A., BREIDABLIK, O., GJERDE, J. O., KORBA, P., UHLEN, K., VORMEDAL, L. K. "Deployment of a Scada Integrated Wide Area Monitoring System." In *IEEE/PES Transmission and Distribution Conference and Exposition: Latin America*, 2008.
- LEIRBUKT, A. B., GJERDE, J. O., KORBA, P., UHLEN, K., VORMEDAL, L. K., WARLAND, L. "Wide Area Monitoring Experiences in Norway." In *Power Systems Conference and Exposition: IEE/PES*, 2006.
- LIRA, R. "Influência Dos Canais De Instrumentação Na Exatidão Da Medição Fasorial Sincronizada." Mestrado, Universidade Federal de Santa Catarina, 2010.
- MAKOFKSKE, D., ALMEROOTH, K. *Multicast Sockets: The Practical Guide for Programmers*. 1 Ed ed, *The Practical Guides Series*: Morgan Kaufmann Publishers, 2003.

- MANTEGAZZA, P. "Real Time Application Interface." In [www.rtai.org](http://www.rtai.org), 1999.
- MARTIN, K. "Enhanced Phasor Data Concentrator." In [http://www.naspi.org/toolRepository/tool\\_details.aspx?tid=75](http://www.naspi.org/toolRepository/tool_details.aspx?tid=75), 2010.
- MARTÍNEZ, E. "Simefas: A Phasor Measurement System for the Security and Integrity of Mexico's Electric Power System." In *IEEE PES General Meeting 2008*, 2008.
- MEDFASEE. "Sistema De Medição Fasorial Sincronizada Com Aplicações Em Sistemas De Energia Elétrica - [Www.Labplan.Ufsc.Br/Medfasee](http://Www.Labplan.Ufsc.Br/Medfasee) - Acessado Em 09 De Fevereiro De 2009." 2003.
- MEDFREQ. "Sistema Nacional De Medição Fasorial Sincronizada Em Baixa Tensão Para a Monitoração E Estudos Do Sistema Elétrico - [Www.Labplan.Ufsc.Br/Medfreq](http://Www.Labplan.Ufsc.Br/Medfreq) - Acessado Em 09 De Fevereiro De 2009." 2008.
- MOGUL, J. "Rfc 919 - Broadcasting Internet Datagrams." 1984.
- MORAES, R. M., VOLSKIS, H. A. R., HU, Y. "Deploying a Large-Scale Pmu System for the Brazilian Interconnected Power System." In *III Conference on Electric Utility Deregulation and Restructuring and Power Technologies*. China, 2008.
- MORAES, R. M. D., VOLSKIS, H. A. R., GIOVANINI, R., HU, Y., MANTO, R., SARDINHA, C., NOVOSEL, D., CENTENO, V. "Arquitetura Do Sistema De Medição Sincronizada De Fasores Do Sin. Requisitos E Aplicações." Paper presented at the XIX Seminário Nacional de Produção e Transmissão de Energia Elétrica - SNP-TEE 2007.
- NADUVATHUPARAMBIL, B., VALENTI, M. C., FELIACHI, A. "Communication Delays in Wide Area Measurement Systems " In *34th Southeastern Symposium on System Theory*, 2002.
- NASPI. "North American Synchrophasor Initiative - [Www.Naspi.Org](http://Www.Naspi.Org) - Acessado Em 09 De Fevereiro De 2009." 2007.
- NAVSTAR. "Navigation Signal Timing and Ranging Global Positioning System - Disponível Em [Http://Www.Spaceandtech.Com/Spacedata/Constellations/Navstar-Gps\\_Consum.Shtml](http://Www.Spaceandtech.Com/Spacedata/Constellations/Navstar-Gps_Consum.Shtml) - Último Acesso Em 29/08/2010." 2010.
- NETL. "A Systems View of the Modern Grid - Disponível Em [Http://Www.Netl.Doe.Gov/Smartgrid/Refshelf.Html](http://Www.Netl.Doe.Gov/Smartgrid/Refshelf.Html) - Último Acesso Em 13 De Setembro De 2010." (2007).
- OPENPDC. "[Http://Openpdc.Codeplex.Com/](http://Openpdc.Codeplex.Com/)." 2009.
- OTA, Y., HASHIGUCHI, T., UKAI, H., SONODA, M., MIWA, Y., TAKEUCHI, A. "Monitoring of Interconnected Power System Para-

- meters Usign Pmu Based Wams." In *IEEE Lausanne Power Tech*, 2007.
- PHADKE, A. G. "Synchronized Phasor Measurements - a Historical Overview." In *Transmission and Distribution Conference and Exhibition: Asia Pacific: IEEE/PES*, 2002.
- PHADKE, A. G., THORP, J. S. "History and Applications of Phasor Measurements." In *IEEE Power Systems Conference and Exposition*, 2006.
- PHADKE, A. G., VOLSKIS, H., MORAES, R. M. D., BI, T., NAYAK, R. N., SEHGAL, Y. K., SEN, S., MARTÍNEZ, E., SATTINGER, W., SAMUELSSON, O., NOVOSEL, D., MADANI, V., KULIKOV, Y. A. "The Wide World of Wide-Area Measurement." *IEEE Power & Energy Magazine* 2008.
- PSYMETRIX. "Phasor Point Solutions." In [www.psymetrix.com](http://www.psymetrix.com), 2010.
- RASMUSSEN, J., JORGENSEN, P. "Synchronized Phasor Measurements of a Power System Event in Eastern Denmark." *IEEE Transactions on Power Systems* 21 (2006): 278-84.
- ROBERTSON, R., CARROLL, R. "Nerc Pcs Project Update Presentation." In disponível em [www.naspi.org](http://www.naspi.org), 2009.
- RODRIGUES, M. A. M., VOLSKIS, H. A. R., LIMA, L. C., SOLLERO, R. B., MOREALE, M. S., MANO, R., HU, Y. "Synchrophasor Data Applications for Power System Control Center in Brazil - a Proof-of-Concept Initiative for Control Center Candidate Functions." In *Study Committee B5 Colloquium*. Jeju Island - Korea, 2009.
- RUMBAUGH, J., BLAHA, M., PRELERMANI, W., EDDY, F., LORENSSEN, W. *Modelagem E Projetos Baseados Em Objetos*: Editora Campus, 1994.
- SANTOS, M. M. "Análise Da Dinâmica De Sistemas Elétricos De Potência Usando Dados De Medição Fasorial Sincronizada." Universidade Federal de Santa Catarina, 2008.
- SEL. "Schweitzer Engineering Laboratories Pdc." In [www.selinc.com](http://www.selinc.com), 2010.
- SILVA, E. L. D. *Formação De Preços Em Mercados De Energia Elétrica*. 1 ed: Sagra Luzzatto, 2001.
- SKOK, S., IVANKOVIC, I., CERINA, Z., ORSULIC, V. "Using Wam in Croatian Power System for Energy Transit Utilization." In *2007 IEEE Lausanne Power Tech*, 2007.
- SOONEE, S. K., NARASIMHAN, S. R., PORWAL, R. K., KUMAR, S., KUMAR, R., PANDEY, V. "Application of Phase Angle Measurement for Real Time Security Monitoring of Indian Electric Power System - an Experience." In *CIGRÉ Session*, 2008.
- STEVENS, W. R., FENNER, B., RUDOFF, A. M. *Unix Network Programming - the Sockets Networking Api*. 3 Edição ed. Vol. 1: Addison-Wesley, 2004.

- STROUSTRUP, B. *The C++ Programming Language*. 3 Edição ed: Addison-Wesley, 2000.
- THALER, D., FENNER, B., QUINN, B. "Rfc 3678 - Socket Interface for Multicast Source Filters." 2004.
- VEJRAZKA, F. "Galileo and Other Satellite Navigation Systems." In *Radioelektronika 17th International Conference* 2007.
- W3C. "Extensible Markup Language (Xml) 1.0 (Fifth Edition) - Disponível Em <http://www.w3.org/Tr/Rec-Xml/> - Último Acesso Em 02/05/2010." (2008).
- WU, J., ZHOU, J., ZHANG, D., ZHOU, J., WANG, Z., XU, S. "Pmu Standard of China." In *IEEE Asia Pacific Conference on Circuits and Systems - APCCAS*, 2008.
- XIAOHUI, B., HUI, H. "Research on Phasor Measurement Unit Based On "Beidou I" Satellite Navigation System." In *International Conference on E-Business and Information System Security - EBISS '09*, 2009.
- XIE, X., XIN, Y., XIAO, J., WU, J., HAN, Y. "Wams Applications in Chinese Power System." *IEEE Power & Energy Magazine* 2006, 54-63.
- XU, Z., OSTERGAARD, J., TOGEBY, M., ISLEIFSSON, F. R. "Evaluating Frequency Quality of Nordic System Using Pmu Data." In *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century* 2008.
- YANG, Q., BI, T., WU, J. "Wams Implementation in Cihna and the Challenges for Bulk Power System Protection." In *IEEE Power Engineering Society General Meeting*, 2007.
- ZAIDI, A. S., SUDDLE, M. R. "Global Navigation Satellite Systems: A Survey." In *International Conference on Advances in Space Technologies*, 2006
- ZUO, J., CARROL, R., TRACHIAN, P., DONG, J., AFFARE, S., ROGERS, B., BEARD, L., LIU, Y. "Development of Tva Superpdcc: Phasor Applications, Tools and Event Replay." In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century: IEEE/PES*, 2008.