MARCOS ROBERTO ALVES MARTINS

INTEGRAÇÃO SISTÊMICA EM *MIDDLEWARE*

São Paulo

2010

MARCOS ROBERTO ALVES MARTINS

INTEGRAÇÃO SISTÊMICA EM *MIDDLEWARE*

Dissertação apresentada à Escola
Politécnica da Universidade de São
Paulo para obtenção do título de
Mestre em Engenharia Elétrica.

Área de concentração:
Engenharia Elétrica e de Sistemas Digitais

Orientador:
Prof. Dr. José Sidnei Colombo Martini

São Paulo

2010

Este exemplar foi revisado e alterado em relação à versão original, sob-responsabilidade única do autor com a anuência de seu orientador.

São Paulo, 29 de julho de 2010.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Martins, Marcos Roberto Alves

**Integração sistêmica em *middleware* / M.R.A. Martins. --
ed. rev. -- São Paulo, 2010.
233 p.**

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Middleware (Desenvolvimento; Análise) I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais II. t.

DEDICATÓRIA

*Para Marcos Filho, que a criatividade e a dedicação
façam parte de todos os seus dias.*

AGRADECIMENTOS

Ao Prof. Dr. José Sidnei Colombo Martini, pela motivação, dedicação, inspiração, que foram fundamentais na elaboração deste trabalho, mostrando sempre o caminho a ser seguido, mesmo quando a estrada se mostrava difícil.

A todos os Professores que contribuíram com conhecimento e dedicação.

A todos os colegas com os quais tive o privilégio de conviver durante o período de estudos.

À minha família pela motivação e compreensão e pelas horas de ausência.

Ao meu pai Osvaldo (*in memorian*) pelo exemplo de vida e perseverança.

À minha esposa Antonia pela dedicação e companheirismo.

Ao meu filho Marcos pelo incentivo.

À minha irmã Djanira pela motivação e orientação.

A DEUS sobretudo, pelo grande milagre da Vida.

“Não basta ensinar ao homem uma especialidade, porque se tornará assim uma máquina utilizável, mas não uma personalidade. É necessário que adquira um sentimento, um senso prático daquilo que vale a pena ser empreendido, daquilo que é belo, do que é moralmente correto. A não ser assim, ele se assemelhará, com seus conhecimentos profissionais mais a um cão ensinado do que a uma criatura harmoniosamente desenvolvida. Deve aprender a compreender as motivações dos homens, suas quimeras e suas angústias, para determinar com exatidão seu lugar preciso em relação a seus próximos e à comunidade.”

Albert Einstein (1879 – 1955)

Físico alemão

RESUMO

O presente trabalho estuda de forma analítica as diversas arquiteturas de redes utilizadas em ambientes industriais e administrativos, bem como os aspectos de convergência que levaram ao surgimento de plataformas de integração que definiram soluções de gestão corporativas.

Sob uma abordagem evolutiva serão analisados os conceitos e aplicabilidades dos padrões de comunicação de dados e de informação que definiram arquiteturas de sistemas distribuídos.

Serão abordados os aspectos de relevância em tecnologia da informação utilizados na gestão corporativa e industrial que possibilitaram o entendimento de soluções de integração baseadas em tecnologias de *middleware*.

Palavras-chave: *Middleware*, Redes, Integração.

ABSTRACT

This paper studies analytically the various network architectures used in industrial environments and administrative, as well as areas of convergence that led to the emergence of integration platforms that have defined corporate management solutions.

Under an evolutionary approach will consider the applicability of the concepts and communication patterns of data and information that defined architectures of distributed systems.

The aspects of relevance in information technology used in corporate management and industrial enabled the understanding of integration solutions based on middleware technologies.

Keywords: Middleware, Network Integration.

LISTA DE ILUSTRAÇÕES

Figura 01:	Modelo OSI	28
Figura 02:	<i>Splitting</i> camada (N)	32
Figura 03:	<i>Pass-through</i> dos serviços de sessão	35
Figura 04:	Relação entre processos, entidades e elementos de serviço de aplicação.	36
Figura 05:	Transmissão de dados no Modelo OSI	37
Figura 06:	Sistema distribuído organizado como <i>Middleware</i>	41
Figura 07:	Sistemas com múltiplos processadores	44
Figura 08:	Sistemas distribuídos	47
Figura 09:	Arquitetura em camadas – computação em grade	48
Figura 10:	Camadas de <i>software</i> e <i>hardware</i> em sistemas distribuídos	53
Figura 11:	Arquitetura de sistema operacional de núcleo monolítico	56
Figura 12:	Sistema cliente/servidor.	57
Figura 13:	Aplicação de banco de dados por arquivo compartilhado e cliente/servidor	58
Figura 14:	Relação entre Modelo de Referência OSI e Sistema Operacional de Rede (SOR).	60
Figura 15:	As “ondas” da arquitetura cliente/servidor	60
Figura 16:	Arquitetura <i>Peer-to-Peer</i>	62
Figura 17:	Estrutura de banco de dados hierárquico	66
Figura 18:	Sistema corporativo e seus subsistemas	71
Figura 19:	Tecnologia da informação e sistemas de informação	73
Figura 20:	Comunicação entre cliente e servidor com Novell	74
Figura 21:	Modelo Novell x Modelo OSI	75

Figura 22:	Encapsulamentos existentes para IPX	76
Figura 23:	<i>Sliding Window</i> - controle de fluxo em protocolo	76
Figura 24:	Interface <i>Shell</i> da rede <i>Novell</i>	77
Figura 25:	Modelo de Referência OSI e arquitetura SNA	79
Figura 26:	Modelo Rede SNA	80
Figura 27:	Projeto original da <i>Ethernet</i>	84
Figura 28:	Correspondência entre protocolos modelo OSI e protocolos LAN	86
Figura 29:	Formato <i>Frame Ethernet</i> DIX V1.0 e IEEE 802.3	87
Figura 30:	Variantes do <i>Frame Ethernet</i>	88
Figura 31:	Estrutura dos padrões IEEE 802.x	89
Figura 32:	IEEE 802.3z e IEEE 802.3ab	93
Figura 33:	<i>Frames</i> IEEE 802.3 e 802.3q	95
Figura 34:	Camadas da arquitetura TCP/IP	98
Figura 35:	Camadas do Modelo OSI e TCP/IP	99
Figura 36:	Camada de Rede - Serviços	100
Figura 37:	Serviços executados na camada de transporte	101
Figura 38:	Estrutura do segmento TCP	102
Figura 39:	Estrutura do segmento UDP	103
Figura 40:	Formato de datagrama Internet TCP/IP	104
Figura 41:	Formatos cabeçalhos IPv4 e IPv6	106
Figura 42:	Classificação geral dos sistemas de automação	107
Figura 43:	Linguagens de programação IEC 61131-3	112
Figura 44:	Estrutura da Pirâmide de Automação	114
Figura 45:	Base de Dados Comum	115

Figura 46:	Camadas <i>Fieldbus Foundation</i> e Modelo de Referência OSI	120
Figura 47:	Arquiteturas das Redes <i>Fieldbus</i>	124
Figura 48:	Comparação entre o Modelo OSI e o padrão <i>Fieldbus</i>	124
Figura 49:	Comparação entre Modelo OSI e padrão <i>Fieldbus</i> H1.	126
Figura 50:	Arquitetura do protocolo <i>Profibus</i> .	128
Figura 51:	Acesso ao meio do protocolo <i>Profibus</i>	130
Figura 52:	Configuração típica de sistema <i>Profibus</i> em automação de processo.	134
Figura 53:	Estrutura de dispositivo <i>Profinet</i>	136
Figura 54:	Integração <i>Profibus</i> com <i>Ethernet</i> e TCP/IP	137
Figura 55:	Transações entre dispositivos <i>Modbus</i>	139
Figura 56:	<i>Modbus</i> TCP/IP	140
Figura 57:	Formato padrão do Comando <i>HART</i>	141
Figura 58:	Camadas do padrão <i>HART</i> .	142
Figura 59:	Formato do <i>frame</i> FIP	144
Figura 60:	Arquitetura <i>EtherNet/IP</i>	145
Figura 61:	Encapsulamento de mensagens <i>EtherNet/IP</i>	146
Figura 62:	Estrutura do protocolo CAN	147
Figura 63:	Camadas do padrão <i>DeviceNet</i> .	148
Figura 64:	Aplicação de redes industriais	150
Figura 65:	Aplicação de Sistema Supervisório	151
Figura 66:	Geração de sistemas SCADA monolíticos	152
Figura 67:	Geração de sistemas SCADA distribuídos	153
Figura 68:	Geração de sistemas SCADA <i>Networked</i>	153
Figura 69:	Tela gráfica SCADA	155

Figura 70:	Integração de sistema SCADA	157
Figura 71:	Interface gráfica – SCADA++	160
Figura 72:	Arquitetura de sistema de gerenciamento tipicamente distribuído	163
Figura 73a:	Conexões não hierárquicas (<i>peer-to-peer</i>) entre gerentes	164
Figura 73b:	Conexões hierárquicas entre gerentes	164
Figura 74:	Modelo de gerenciamento OSI	167
Figura 75:	Aplicação do protocolo SNMP	169
Figura 76:	Troca de informações SNMP	171
Figura 77:	Formato de mensagens SNMP	173
Figura 78:	Relacionamento entre versões do SNMP	176
Figura 79:	Subárvore MIB II	179
Figura 80:	Resumo comparativo MIBs	180
Figura 81:	Elementos componentes de processo de gestão	182
Figura 82:	Distribuição dos Sistemas de Informação nos níveis hierárquicos	184
Figura 83:	Estrutura típica de sistema ERP	186
Figura 84:	Distribuição de fornecedores de sistemas ERP	187
Figura 85:	Fases de planejamento de sistemas ERP	188
Figura 86:	Níveis propostos para normatização das empresas industriais – MES	190
Figura 87:	Pirâmide de automação sem integração	191
Figura 88:	Aplicação típica de objetos distribuídos	193
Figura 89:	Invocação à métodos locais e remotos	197
Figura 90:	<i>Middleware</i> para comunicação em sistema distribuído	200

Figura 91:	Localização de <i>Middleware</i> no Modelo OSI	200
Figura 92:	<i>Middleware</i> – arquitetura TCP/IP	201
Figura 93:	Comunicação entre processos	202
Figura 94:	Modelo proposto de gestão produtiva.	206
Figura 95:	Arquitetura proposta	207
Figura 96:	ERP com inclusão de camada MES e SNMP4J	208
Figura 97:	Camada <i>Middleware</i>	209
Figura 98:	Estrutura da mensagem requisição/resposta	209
Figura 99:	Comunicação entre entidades RMI	210
Figura 100a:	Invocação remota	211
Figura 100b:	Interação entre componentes - Invocação remota	211
Figura 101:	Construção de aplicação RMI	212
Figura 102:	Arquitetura de infraestrutura de integração	214

LISTA DE TABELAS

Tabela 01:	Computadores com IP registrados na Internet	42
Tabela 02:	Características de sistemas NUMA	45
Tabela 03:	Comparação de arquiteturas de redes SNA e TCP/IP	82
Tabela 04:	Grupos de trabalhos – normatização padrão IEEE 802	85
Tabela 05:	Novas especificações IEEE 802	90
Tabela 06:	Meio de transmissão IEEE 802.3 <i>Ethernet</i> 10 Mbps	90
Tabela 07:	Meio de transmissão IEEE 802.3u <i>Ethernet</i> 100 Mbps.	91
Tabela 08:	Comparação entre IEEE 802.3u e IEEE 802.12	92
Tabela 09:	Resumo comparativo <i>Ethernet</i>	94
Tabela 10:	Meio físico para <i>Ethernet Gigabit</i>	94
Tabela 11:	Modelos de comunicação industrial	117
Tabela 12:	Velocidades de transmissão <i>Profibus DP</i>	129
Tabela 13:	Resumo de funções padrão <i>Profibus DP</i>	132
Tabela 14:	Formato mensagem ASCII	139
Tabela 15:	Formato mensagem RTU	139
Tabela 16:	Características de Supervisórios SCADA++	160
Tabela 17:	RFCs relacionadas com SNMPv1	171
Tabela 18:	RFCs relacionadas com SNMPv2	174
Tabela 19:	RFCs relacionadas com SNMPv2c	175
Tabela 20:	RFCs relacionadas com SNMPv3	175
Tabela 21:	Métodos utilizados da interface <i>Context</i>	213

LISTA DE SIGLAS

ALI	Application Layer Interface
AES	Advanced Encryption Standard
API	Application Program Interface
ASCII	American Code For Information Interchange
ASCE	Association Control Service Element
ASN	Abstract Syntax Notation
BD	Banco de dados
CCITT	Comité Consultatif International Télégraphique et Téléphonique
CMIP	Common Management Information Protocol
CSMA/CD	Carrier Sense Multiple Access/Collision Detection
CLP	Controlador Lógico Programável
CNC	Controle Numérico por Computador
CRC	Cyclic Redundancy Checy
DARPA	Defense Advanced Research Projects Agency
DNS	Domain Name System
DES	Data Encryption Standard
DDM	Distributed Database Maintenance
DIS	Data Independent Sublayer
DME	Distributed Management Environment
DLL	Data Link Layer

EIA	Electronic Industries Association
ER	Elemento de Rede
ERP	Enterprise Resource Planning
EPROM	Erasable Programmable Read Only Memory
FAS	Fieldbus Access Layer
FIP	Factory Instrumentation Protocol
FMS	Fieldbus Message Specification
FTP	File Transfer Protocol
FISCO	Fieldbus Intrinsically Safe Concept
HTTP	Hypertext Transfer Protocol
HDLC	High Level Data Link Control
HSE	High Speed Ethernet
ISP	Interoperable Systems Project
ICMP	Internet Control Message Protocol
IEC	International Electrotechnical Commission
IEEE	Institute Of Electrical And Electronics Engineers
IHM	Interface Homem Máquina
IP	Internet Protocol
ISA	International Society For Measurement And Control
ISO	International Organization For Standardization
I/O	Input / Output

IMSE	Industrial Message Service Element
LAN	Local Area Networks
LAS	Link Active Scheduler
LE	Layer Entity
LLA	Logical Layered Architecture
LLC	Logical Link Control
LLI	Lower Layer Interface
MAC	Medium Access Control
MAU	Medium Attachment Unit
MCSE	Message Common Service Element
MDS	Medium Dependent Sublayer
MIB	Management Information Base
MMS	Manufacturing Message Specification
MPS	Message Periodic / Aperiodic Services
MRP	Material Resource Planning
NMS	Network Management Stations
OID	Object Identifier
OSF	Open Software Foundation
OSI	Open Systems Interconnection
OPC - OLE	Object Linking And Embedding Process Control
PC	Personal Computer

PDU	Protocol Data Unit
PI	Profibus International
ROSE	Remote Operations Service Element
RMON	Remote Monitoring
RFC	Request For Comment
RTU	Remote Terminal Unit
SCADA	Supervisory Control And Data Acquisition
SDCD	Sistema Digital De Controle Distribuído
SGMP	Simple Gateway Management Protocol
SQL	Structure Query Language
SHA	Secure Hash Algorithm
SOR	Sistemas operacionais de rede
SMI	Structure Of Management Information
SNAP	Subnetwork Access Protocol
SNMP	Simple Network Management Protocol
SMTP	Simple Mail Transfer Protocol
SNAP	Subnetwork Access Protocol
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UCP	Unidade Central De Processamento
UDP	User Datagram Protocol

VCR Virtual Communication Relationship

VFD Virtual Field Device

SUMÁRIO

1. INTRODUÇÃO	21
1.1 MOTIVAÇÃO	22
1.2 OBJETIVO	22
1.3 METODOLOGIA	23
1.4 ORGANIZAÇÃO DA DISSERTAÇÃO	23
2. MODELO OSI	26
2.1 PROTOCOLOS DE INTERCONEXÃO	27
2.2 CAMADA FÍSICA.....	29
2.3 CAMADA DE ENLACE.....	29
2.4 CAMADA DE REDE	30
2.5 CAMADA DE TRANSPORTE.....	31
2.6 CAMADA DE SESSÃO	33
2.7 CAMADA DE APRESENTAÇÃO.....	34
2.8 CAMADA DE APLICAÇÃO.....	35
3. SISTEMAS DISTRIBUÍDOS	39
3.1 MODELOS DE ARQUITETURAS DE SISTEMAS DISTRIBUÍDOS	52
3.2 ARQUITETURAS CENTRALIZADAS	55
3.3 ARQUITETURAS DESCENTRALIZADAS	61
3.4 BANCO DE DADOS.....	64
4. TECNOLOGIA DA INFORMAÇÃO.....	70
4.1 NOVELL.....	74
4.2 SNA	78
4.3 ETHERNET	83
4.4 TCP/IP	96
5. AUTOMAÇÃO INDUSTRIAL	107
5.1 EVOLUÇÃO DO CONTROLE INDUSTRIAL.....	107
5.2 ARQUITETURA DE AUTOMAÇÃO INDUSTRIAL	111
5.3 MODELO PARA TROCA DE INFORMAÇÕES	116
6. REDES DE CAMPO INDUSTRIAIS	119

6.1	CAMADA FÍSICA.....	121
6.2	CAMADA LINK DE DADOS (ENLACE)	121
6.3	CAMADA DE APLICAÇÃO.....	122
6.4	CAMADA DE APRESENTAÇÃO (USUÁRIO)	122
7.	ARQUITETURAS DE REDES INDUSTRIAIS.....	123
7.1	FIELDBUS.....	123
7.2	PROFIBUS.....	128
7.3	MODBUS.....	138
7.4	HART	141
7.5	WORLDFIP.....	142
7.6	ETHERNET/IP (ETHERNET INDUSTRIAL PROTOCOL)	145
7.7	CAN	146
7.8	LONWORKS.....	148
7.9	BACNET	149
8.	SISTEMAS DE AQUISIÇÃO DE DADOS	151
8.1	COMPONENTES DE SISTEMAS SUPERVISÓRIOS SCADA.....	156
8.2	MODOS DE COMUNICAÇÃO SCADA.....	158
8.3	SISTEMA SUPERVISÓRIO SCADA ORIENTADO À OBJETO	159
9.	GERENCIAMENTO DE REDES	161
9.1	PROTOCOLO DE GERENCIAMENTO SNMP	168
9.2	MIB (MANAGEMENT INFORMATION BASE).....	177
10.	MODELO DE GESTÃO CORPORATIVA	182
10.1	ERP (ENTERPRISE RESOURCE PLANNING)	185
10.2	MES (MANUFACTURING EXECUTION SYSTEM).....	189
11.	INTEGRAÇÃO EM APLICAÇÕES DISTRIBUÍDAS	193
11.1	TECNOLOGIAS EM APLICAÇÕES DISTRIBUÍDAS	197
11.2	MIDDLEWARE.....	199
12.	ARQUITETURA PROPOSTA	206
13.	CONCLUSÃO E CONSIDERAÇÕES FINAIS.....	215
14.	REFERÊNCIAS.....	218

1. INTRODUÇÃO

O estudo da evolução de qualquer área da tecnologia proporciona uma melhor compreensão das principais realizações nestes segmentos, fornecendo subsídios para a compreensão de tais desenvolvimentos e suas tendências.

O aprimoramento da tecnologia possui uma relação intrínseca com o desenvolvimento corporativo. No campo das comunicações, desde a invenção do telefone aos atuais sistemas de processamento de dados, as tecnologias possibilitaram o surgimento de métodos de gestão corporativos cada vez mais abrangentes e eficientes.

As redes de computadores surgiram há relativamente pouco tempo, no final da década de 1960, elas herdaram diversas propriedades úteis de suas predecessoras, as mais antigas e amplamente adotadas redes telefônicas. Isto não deve causar surpresa, uma vez que tanto computadores quanto telefones são instrumentos de comunicação.

Os processos industriais tiveram desenvolvimento análogo ao das comunicações, e ambos possuem em comum a busca pela eficiência na gestão de seus processos. Os atuais sistemas de automação industrial baseados em sinais digitais tiveram como precursores os sistemas pneumáticos, e da mesma forma herdaram propriedades também muito úteis.

Contudo, atualmente existe um novo contexto, a presença dos sistemas de tecnologia da informação, que foram alavancados com o surgimento da Internet nos anos de 1990, que criou um cenário de integração e de convergência.

A influência das redes de computadores sobre outros tipos de telecomunicações resultou em uma convergência de redes que atualmente definiu o *VoIP*; sistemas de TV convergiram com os sinais digitais definindo a TV digital, entre muitos outros exemplos.

A Internet contribuiu de forma determinante para o surgimento de novas formas de relacionamento, entre pessoas e principalmente entre empresas, com o surgimento de serviços como *e-Commerce*, *e-Banking*, que propiciaram um ambiente de aplicações distribuídas como os *WebServices*.

O elemento comum de todo este enorme processo de convergência tem sido a tecnologia da informação, que no ambiente corporativo, possui como uma das principais metas a interoperabilidade.

O surgimento e aprimoramento de linguagens de programação de alto nível possibilitam que a interoperabilidade possa ser atingida com a utilização de níveis de abstração definidos como *middleware* entre as diversas aplicações legadas, as existentes e as novas tecnologias que certamente surgirão.

1.1 Motivação

A principal motivação para o desenvolvimento deste trabalho deve-se a falta de a interoperabilidade entre os sistemas destinados ao segmento de processos produtivos, inclusive em sistemas produtivos que herdaram plataformas legadas de automação.

Além dos problemas técnicos provenientes entre estas plataformas, onde a troca de informações ocorre mediante o desenvolvimento de soluções dedicadas, que comumente são onerosas e possuem aplicabilidade específica e que são desenvolvidas em sua grande maioria ao nível de *hardware*, inviabilizando a convergência das informações geradas com outros sistemas no ambiente corporativo.

A busca pela eficiência e padronização de Modelos de Gestão Empresariais inseriu o contexto de integração das informações como elemento fundamental na estratégia das corporações.

A aproximação entre processos nos ambiente da tecnologia de automação (TA) e da tecnologia da informação (TI), antes separados por barreiras culturais e tecnológicas, encontra na integração uma forma de dispor as informações aos níveis decisórios das empresas de forma ágil e consistente.

1.2 Objetivo

A presente dissertação tem por objetivo elaborar uma proposta de integração em ambientes heterogêneos, por meio da especificação de uma arquitetura aberta

baseada em *middleware* que possa prover o gerenciamento da informação no ambiente industrial integrado à gestão corporativa.

1.3 Metodologia

Para atingir o objetivo proposto para a elaboração deste trabalho, será adotada a seguinte metodologia:

- Pesquisa dos modelos e de padronização das arquiteturas de redes de computadores;
- Análise dos modelos e arquiteturas de sistemas distribuídos, e suas respectivas aplicações;
- Análise das tecnologias de informação que foram mais utilizadas no ambiente corporativo, e formas de integração dos protocolos que constituem tais soluções;
- Análise do contexto da automação industrial, arquiteturas e sistemas mais utilizados nas diversas camadas do processo produtivo;
- Avaliação dos sistemas de aquisição de dados utilizados na coleta de dados no ambiente industrial;
- Análise de arquiteturas destinadas ao gerenciamento de redes de computadores e suas plataformas;
- Análise dos modelos de Gestão Corporativas embasadas em tecnologias da informação e soluções disponíveis para integração de dados;
- Pesquisa das tecnologias utilizadas para integração em aplicações distribuídas, com aplicação de camadas de abstração entre processos;
- Elaboração de uma proposta de arquitetura baseada em camadas de *middleware* integrando processo produtivo a gestão corporativa.

1.4 Organização da dissertação

Além desta introdução, a dissertação compreende os seguintes capítulos:

CAPÍTULO 2: Modelo OSI, descreve as principais características da padronização das redes de computadores.

CAPÍTULO 3: Sistemas Distribuídos apresenta as principais características que definiram as arquiteturas de sistemas distribuídos, aspectos evolutivos relacionados às aplicações distribuídas e suas respectivas tecnologias.

CAPÍTULO 4: Tecnologia da Informação descreve as principais arquiteturas de redes de computadores destinadas ao ambiente corporativo, aspectos conceituais e evolutivos, arquiteturas e formas de integração entre protocolos.

CAPÍTULO 5: Automação Industrial analisa os principais aspectos relacionados à automação de processos.

CAPÍTULO 6: Redes de Campo Industriais; aborda os principais aspectos e características das redes de automação industrial, definição de serviços e camadas.

CAPÍTULO 7: Arquiteturas de Redes Industriais detalha as principais arquiteturas destinadas à automação de processos industriais, focando aspectos estruturais dos protocolos utilizados e suas respectivas arquiteturas de rede industrial, bem como a aplicabilidade de cada grupo de padrão de protocolo dentro da cadeia de processo produtivo.

CAPÍTULO 8: Sistemas de Aquisição de Dados analisa as características construtivas relacionadas a sistemas de aquisição de dados em ambientes de automação industrial.

CAPÍTULO 9: Gerenciamento de Redes descreve as principais arquiteturas de gerenciamento de redes, utilizadas em arquiteturas padronizadas como OSI e TCP/IP.

CAPÍTULO 10: Modelo de Gestão Corporativa descreve os conceitos de gestão corporativos e sua integração com a tecnologia da informação por plataformas ERP.

CAPÍTULO 11: Integração em Aplicações Distribuídas descreve os aspectos relacionados às tecnologias utilizadas em aplicações distribuídas e suas respectivas aplicabilidades e propriedades.

CAPÍTULO 12: Arquitetura Proposta, propõe um modelo de integração entre processos baseado em *middleware*.

CAPÍTULO 13: Conclusão e Considerações Finais, onde são apresentadas as considerações finais derivadas do desenvolvimento da dissertação e propostas de trabalhos futuros.

2. MODELO OSI

A comunicação entre os equipamentos implementados em uma rede ocorre por meio de modelos de arquiteturas de redes e de protocolos que viabilizem a comunicação entre estes equipamentos e dispositivos.

Vários modelos de arquiteturas foram propostos nas últimas décadas, porém o desenvolvimento de redes por meio de protocolos de comunicação tem-se mostrado o método mais eficiente devido às possibilidades de expansão e implementações. As redes proprietárias deixaram de ser soluções para as comunicações à medida que o número, o conjunto de serviços e os protocolos de cada camada apresentam variações de uma arquitetura para outra; e para que haja a interligação de redes heterogêneas proprietárias fizeram-se necessário a utilização de *drivers* de comunicação, que realizam uma coleção de rotinas do sistema operacional que efetivamente unem as pilhas de protocolos ao *hardware* da rede. Ele normalmente inclui rotinas para inicializar o dispositivo, transmitir quadros no enlace e interrupções de campo. O código utilizado no *driver* comumente é difícil de ler, pois é repleto de detalhes específicos do dispositivo, mas a lógica geral é relativamente simples.

Este fato foi determinante para a elaboração de modelos que iniciassem a padronização das redes; com a definição do modelo OSI (*Open Systems Interconnection*) proposto pela ISO (*International Organization for Standardization*) entre 1978 e 1984, conforme expõe [CARVALHO-94], que foi desenvolvido com o objetivo de padronizar a comunicação entre os equipamentos de uma rede por meio de protocolos de rede; a proposta do Modelo de Referência ISO OSI (*Open System Interconnection*) foi oferecer modelo de interconexão para sistemas abertos.

O Modelo de Referência para Interconexão de Sistemas Abertos e os diversos padrões correlacionados foram desenvolvidos por grupos de trabalho do *Telecommunications and Information Exchange Between Systems* e do *Information Retrieval, Transfer and Management for OSI*, ambos pertencentes ao *Information Processing Systems*, conforme expõe [SOARES-99].

A ISO foi uma das organizações que formalmente definiram um modo comum de conectar computadores em rede. A arquitetura estruturada pela organização ISO, definida como *Open System Interconnection* (OSI), define uma estrutura em sete

camadas, de forma aberta, onde protocolos implementam as funcionalidades de cada camada, viabilizando o tráfego da informação, desde o nível físico até os níveis de aplicação.

Conforme expõe [CARVALHO-94], "o funcionamento em camadas do Modelo de Referência OSI esta baseado no princípio de usuário e prestador de serviços, sendo que cada serviço representa um conjunto de funções, onde cada uma das camadas é prestadora de serviço à camada superior".

Cabe salientar que o Modelo de Referência OSI aplicado de forma isolada não define a arquitetura de uma rede, ele aponta o que cada camada deve executar. Conforme define [SOARES-99], "o fato de dois sistemas distintos atenderem o Modelo de Referência OSI não garante que eles possam trocar informações entre si, pois o modelo permite que sejam usadas diferentes opções de serviços/protocolos para as várias camadas".

2.1 Protocolos de Interconexão

O modelo OSI foi estruturado em pilhas, composto por sete camadas, onde cada camada deve ser vista como um programa, implementado por *hardware* ou *software*, que se comunica com o nível correspondente no outro equipamento. Os dados não são transferidos diretamente de forma horizontal entre os equipamentos, mas verticalmente pelos níveis, partindo do equipamento transmissor seguindo através do meio físico até o nível correspondente no equipamento receptor até o seu destino. Desta forma a comunicação percorre as sete camadas do modelo, sendo que cada camada executa uma função limitada e definida, dentro da arquitetura.

Os limites entre cada camada são chamados de interfaces, e estas interfaces são específicas para cada camada. Cada camada oferece serviços para a próxima camada, durante o transporte das informações, que são realizados por protocolos.

Apesar de o modelo OSI possuir sete camadas, um sistema de comunicação não precisa necessariamente implementar as sete camadas. A figura 01 a seguir ilustra a interação entre as camadas do modelo OSI.

Figura 01: Modelo OSI

Fonte: [PETERSON-04]

O modelo OSI é composto pelas seguintes camadas, segundo expõe [TANENBAUM-03], sendo:

- Camada física;
- Camada de enlace de dados;
- Camada de rede;
- Camada de transporte;
- Camada de sessão;
- Camada de apresentação;
- Camada de aplicação.

Entre cada nível adjacente (entre camadas) são definidas interfaces, que realizam a transmissão de dados verticalmente. O serviço fornecido por uma camada à outra é especificado pelo conjunto de primitivas de serviços, trocadas

entre si e pela ordem segundo a qual as primitivas são trocadas, conforme define [SOARES-99].

Segundo [OLIFER-08], há distinção entre o Modelo de Referência OSI e a pilha de protocolos OSI, onde o Modelo de Referência constitui um método conceitual entre sistemas abertos e a pilha OSI trata-se de um conjunto de especificações para protocolos determinados.

2.2 Camada Física

A camada física define as características elétricas, mecânicas, funcionais e procedurais para manter e ativar conexões físicas, onde uma das principais funções da camada física é permitir a ativação e desativação de conexões físicas. Os protocolos nesta camada se dedicam ao tipo de conexão e topologia física. Sua função é possibilitar o envio de uma cadeia de *bits* pela rede, não tratando o significado destes *bits* nem a forma como serão agrupados. Neste nível é definido se a transmissão será *half-duplex* ou *full-duplex*. Esta camada compreende as especificações do *hardware* utilizado na rede.

A camada física exerce a função de sequenciação, onde os *bits* são transmitidos e encaminhados ao nível da camada de enlace na mesma ordem em que são recebidos. No nível da camada física também é realizado o gerenciamento das conexões físicas, assegurando a qualidade de serviço, porém neste nível não são tratados problemas relacionados com erros de transmissão.

2.3 Camada de Enlace

Conforme expõe [TANENBAUM-03], a principal tarefa da camada de enlace é transformar um canal de transmissão bruto em uma linha que pareça livre de erros de transmissão não detectados para a camada superior (camada de rede). Neste nível, a camada de enlace organiza os *bits* transmitidos pelo meio físico, organizando-os na forma de quadros (*frames*). Este nível detecta erros durante o processo de transmissão, utiliza técnicas de redundância, controla o fluxo de transmissão impedindo desta forma que o transmissor envie mais dados que o receptor possa processar e identificar os dispositivos da rede; os protocolos desta

camada possuem como funcionalidade principal fazer com que os dados transmitidos alcancem o destino com integridade.

Conforme expõe [CARVALHO-94], as principais funções da camada de enlace são:

- Estabelecimento/liberação da conexão de enlace;
- Mapeamento um-para-um entre as unidades de dados;
- *Splitting* da conexão de enlace;
- Montagem e delimitação de quadros;
- Controle de seqüência;
- Controle de fluxo;
- Controle de erro;
- Controle de interconexão de dados;
- Gerenciamento da qualidade de serviços prestados.

O controle ao acesso do canal compartilhado, conforme define [TANENBAUM-03] pode ser obtido por meio de uma subcamada especial da camada de enlace de dados: a camada de controle de acesso, definida como subcamada MAC (*Medium Access Control*) e a subcamada LLC (*Logical Link Control*), que executa controle de seqüência, controle de fluxo e recuperação de erros em redes locais.

2.4 Camada de Rede

A camada de rede define a rota e o processo pelo qual os dados serão trafegados pela rede, fornecendo ao nível de transporte uma independência quanto as considerações de chaveamento e roteamento, onde as mensagens são encaminhadas na rede segundo algoritmos de roteamento e endereçamento. Prove também serviços pertinentes a duas filosofias:

- Circuitos virtuais. São serviços orientados à conexão, nesta técnica, os pacotes pertencentes a uma única conversaço, no so independentes;
- Datagramas. So servios no orientados conexo, onde o roteamento determinado toda vez que um pacote encaminhado por um n da rede.

O controle de congestionamento de pacotes também é realizado pela camada de rede, possibilitando que redes heterogêneas sejam interconectadas.

As principais funções da camada de rede, conforme define [CARVALHO-94] são:

- Roteamento e *relaying*;
- Multiplexação da conexão de rede;
- Segmentação e blocagem;
- Controle de erro;
- Sequenciação;
- Controle de fluxo;
- Transferência de dados expressos;
- Reinicialização;
- Seleção de serviço;
- Gerenciamento.

O objetivo do nível da camada de rede é oferecer ao nível da camada de transporte independência quanto ao roteamento, possibilitando entre outros serviços, controle de congestionamento.

2.5 Camada de Transporte

A camada de transporte tem a função de endereçamento e de organização das mensagens em segmentos, entregando-as à camada superior (camada de sessão); a camada de transporte também determina o tipo de serviço a ser fornecido à camada de sessão. Neste nível também é configurada a distribuição dos endereços dos nós, bem como os métodos de detecção de erros e recuperação. Caso os dados não sejam enviados corretamente ao dispositivo receptor, a camada de transporte inicia a retransmissão ou informa às camadas superiores para correção do problema.

No nível da camada de transporte são realizados o controle de fluxo das mensagens, a multiplicação e o *splitting* (uma conexão de transporte ligada a várias

conexões de rede). Conforme expõe [CARVALHO-94], o *splitting* “corresponde ao caso em que uma única conexão (N) é mapeada em diversas conexões da camada inferior (N-1).”

A figura 02 abaixo ilustra o mapeamento entre conexões.

Figura 02: *Splitting* camada (N)

Fonte: [CARVALHO-94]

Conforme define [TANENBAUM-03], “a camada de transporte é uma verdadeira camada fim-a-fim, que liga origem à destino”, ou seja, nas camadas inferiores, os protocolos são trocados entre cada uma das máquinas, nas camadas de transporte, de sessão, de apresentação e aplicação, as camadas são consideradas fim-a-fim, complementa [TANENBAUM-03].

Conforme expõe [CARVALHO-94], as principais funções da camada de transporte, definidas pela ISO, em função da qualidade de serviço, são:

- Mapeamento do endereço de transporte no endereço de rede;
- Estabelecimento e liberação de conexão de transporte;
- Multiplexação (fim-a-fim) de conexões de transporte sobre conexões de rede;
- Segmentação, blocagem e concatenação fim-a-fim;
- Controle de seqüência e controle de erro;

- Controle de fluxo;
- Monitoração de fluxo;
- Monitoração de qualidade de serviço;
- Transferência de unidades de dados expressos;
- Gerenciamento da qualidade de serviço.

Conforme define [SOARES-99], “o controle de fluxo possui função relevante neste nível, visto que nenhuma implementação tem espaço de armazenamento infinito, algum mecanismo deve ser assegurar que o transmissor envie mensagens numa taxa maior que a capacidade do receptor possa recebê-las.”

2.6 Camada de Sessão

Na camada de sessão é realizado o gerenciamento de *token*, controle do diálogo e gerenciamento de atividades de tráfego de informações. A definição de direção dos dados pelos modos:

- *Simplex*, comunicação em apenas uma direção;
- *Half duplex*, cada equipamento pode transmitir e receber, entretanto, um de cada vez;
- *Full duplex* permite comunicação simultânea, recebendo e transmitindo dados.

Este nível executa também a administração da sessão, efetuando o estabelecimento da conexão, transferindo dados e a liberação da conexão, permitindo que os usuários de diferentes máquinas possam estabelecer comunicação, realiza estabelecimento de conexão com outros usuários, definição de pontos de sincronização em um diálogo, interrupção de diálogos, negociação e utilização de *tokens* para permuta de dados. Conforme expõe [CARVALHO-94], estes serviços estão baseados nos conceitos de:

- Pontos de sincronização, onde são definidos o ponto de sincronização maior e ponto de sincronização menor;
- Atividade, que define um conjunto de unidades de diálogo;

- *Token*, propriedade de uma conexão de sessão dinamicamente atribuída aos serviços da camada de sessão. São definidos os *tokens* de: dados, sincronização menor, sincronização maior e de liberação ordenada.

O nível de sessão oferece mecanismos que possibilitam estruturar os circuitos oferecidos pelo nível de transporte. A camada de sessão utiliza o conceito de atividade, possibilitando aos usuários distinguir partes do intercâmbio de dados, definidos como atividades, que são constituídas de unidades de diálogo; que podem ser interrompidas e depois recomeçadas na mesma sessão, ou em conexões de sessão subseqüentes, conforme expõe [SOARES-99].

2.7 Camada de Apresentação

A camada de apresentação realiza a semântica e a transformação adequada dos dados. Antes do envio dos dados pelo nível de sessão, executa tarefas como compressão de textos e dados, descompressão de arquivos, criptografia de dados, conversão de padrões ASCII (*American Standard Code for Information Interchange*) e EBCDIC (*Extended Binary-Coded Decimal Interchange Code*) de terminais e arquivos para padrões de rede e vice-versa.

A camada de apresentação gerencia as estruturas de dados abstratos e possibilita o intercâmbio de dados de nível mais elevado. Nesta camada é realizada a codificação da informação especificada por meio de sintaxe abstrata em octetos para que os dados possam ser transferidos à camada de sessão.

Conforme define [CARVALHO-94], as principais funções da camada de apresentação são:

- Mapeamento da conexão de apresentação;
- Negociação e renegociação de contexto de apresentação;
- Transformação de sintaxe;
- Transferência de dados;
- *Pass-through* dos serviços de sessão, que repassa as primitivas de serviços diretamente à camada superior (camada de aplicação). A figura 03 a seguir ilustra esta operação.

- Gerenciamento da camada de apresentação.

Figura 03: *Pass-through* dos serviços de sessão.

Fonte: [CARVALHO-94]

Existe uma correspondência biunívoca entre os endereços de apresentação e de sessão; não existe nenhum tipo de multiplexação nesse nível de protocolo, conforme expõe [SOARES-99].

2.8 Camada de Aplicação

O nível da camada de aplicação oferece aos processos de aplicação os meios para que estes utilizem os meios de comunicação.

A camada de aplicação contém vários protocolos comumente necessários para os usuários. Neste nível são definidas as funções de gerenciamento da rede. O nível de aplicação possui todas as funções necessárias a comunicação entre sistemas abertos que não são executadas pelos outros níveis, além disto, o nível de aplicação realiza também os seguintes serviços: [CARVALHO-94]

- Definição da qualidade de serviço aceitável na conexão;
- Sincronização das aplicações participantes;
- Identificação de parceiros de comunicação;

- Mapeamento da associação de aplicação;
- Determinação da disponibilidade do parceiro de comunicação;
- Definição da responsabilidade na recuperação de erros;
- Especificação de aspectos relativos à segurança, integridade de dados, controle de acesso, etc.;
- Autenticação de parceiros de comunicação;
- Determinação da qualidade de serviço;
- Seleção de modo de diálogo (*full duplex* ou *half duplex*).

Conforme expõe [CARVALHO-94], “o conjunto de elementos de serviço de aplicação vinculados a uma associação e a relação existente entre tais elementos constituem o contexto de aplicação”.

A figura 04 abaixo ilustra a relação que ocorre entre processos, entidades e elementos de serviço de aplicação, onde cada entidade de aplicação é constituída por Elementos de Serviço de Aplicação, ASE (*Application Service Elements*).

ACSE - Association Control Service Element
 ASE - Application Service Element
 PSAP - Presentation Service Access Point

Figura 04: Relação entre processos, entidades e elementos de serviço de aplicação.

Fonte: [CARVALHO-94]

As seis camadas inferiores incluem funções e implementações que suportam os serviços de rede, enquanto que a camada de aplicação fornece os protocolos necessários para a realização das funções específicas dos serviços da rede, como o FTAM (*File Transfer, Access and Management*), o DS (*Directory Service*) e o MHS (*Message Handling System*), conforme menciona [SOARES-99].

O processo de transmissão de dados no Modelo de Referência OSI, pode ser ilustrado conforme a figura 05 abaixo.

Figura 05: Transmissão de dados no Modelo OSI

Fonte: [SOARES-99]

O processo de transmissão de dados tem início pelo nível de aplicação da estação transmissora; os dados do usuário são denominados SDU (*Service Data Unit*). O nível de aplicação agrega aos dados um cabeçalho definido, conforme expõe [SOARES-99] por PCI (*Protocol Control Information*), onde o resultado desta junção é denominado por PDU (*Protocol Data Unit*).

As informações são transmitidas pelas camadas do Modelo de Referência OSI, que acrescentam em cada nível um cabeçalho; este processo ocorre até o nível de enlace, que acrescenta dispositivo de checagem de erros definido por FCS (*Frame Check Sequence*). Quando os dados são recebidos pela camada física do destinatário, ocorre processo inverso, fazendo com que os dados sejam recebidos pelo usuário destinatário.

As aplicações podem implementar seus próprios protocolos de interação utilizando o conjunto de sete camadas de ferramentas do Modelo de Referência OSI, conforme expõe [OLIFER-08]; especificamente, para esse propósito é fornecida aos

programadores uma interface de programação de aplicações especial definida por API (*Application Programming Interface*).

3. SISTEMAS DISTRIBUÍDOS

Os sistemas de informação atravessaram nas últimas décadas revoluções importantes que certamente alavancaram o desenvolvimento de diversos segmentos da sociedade. Com o surgimento dos computadores, desde a “era moderna da computação” a partir da década de 1940, quando os primeiros computadores eletrônicos não possuíam sistemas operacionais até os sistemas *mainframes*, utilizados no processamento de informações e sistemas de grandes corporações e instituições, que atingiram o auge na década de 1980; possibilitou que diversos setores da economia fossem favorecidos com o surgimento e desenvolvimento de sistemas que agregaram confiabilidade, agilidade a processos executados de forma manual; quando em 1936, o primeiro computador eletromecânico foi desenvolvido pelo engenheiro alemão *Konrad Zuse*. [MACHADO-07]

A ocorrência da Segunda Guerra Mundial acelerou o desenvolvimento de equipamentos e sistemas computacionais que auxiliaram militares nos esforços de guerra. O ENIAC (*Electronic Numeric Integrator And Calculator*) foi o primeiro computador a válvulas desenvolvido pelo exército dos Estados Unidos, utilizado para cálculo de trajetórias balísticas, tendo sido desenvolvido pelos engenheiros *J. Presper Eckert* e *John W. Mauchly*, na Universidade da Pensilvânia, conforme expõe [MACHADO-07].

Nesta ocasião, a linguagem de programação era desenvolvida em função da linguagem da máquina, um *bit* por vez, em filas de chaves mecânicas, conforme expõe [DEITEL-05]. Os primeiros sistemas operacionais passaram a ser utilizados a partir dos anos 50, quando os laboratórios da *General Motors* utilizaram seu computador IBM 701. Em 1951, o MIT (*Massachusetts Institute of Technology*) colocou em funcionamento o primeiro computador voltado para aplicações em tempo real, o *Whirlwind I*, conforme expõe [MACHADO-07].

A partir dos anos 60, *Leonard Kleinrock*, *Paul Baran* e *Donald Davis*; conceituaram os sistemas de redes que utilizavam datagramas definindo a comutação de pacotes em sistemas de computadores. Desde então, o desenvolvimento e conseqüente aperfeiçoamento das redes de computadores locais (LANs - *Local Area Networks*) e mesmo até as redes de longa distância (WANs –

Wide Area Networks) possibilitaram que milhares de equipamentos se conectassem, trocando informações, dados, arquivos; enfim, acelerando o desenvolvimento social, cultural e tecnológico. Nesta ocasião também houve a introdução do conceito de multiprogramação, tornando os sistemas ociosos, em sistemas mais rápidos e eficientes.

A partir da década de 70, sistemas com multiprocessadores, aplicações de tempo compartilhado e de tempo real alcançaram soluções de integração em larga escala para sólidos produtos comerciais. Neste momento, a Internet, as redes locais (*intranets*) e as diversas tecnologias que emergiram em dispositivos de comunicação suportadas em redes exemplificam a gama de serviços e aplicações de sistemas distribuídos. Surgiram soluções proprietárias, como SNA (*System Network Architecture*), DECNet, NCP (*Netware Core Protocol*) e X.25¹.

Na década de 80, *softwares* de rede passam a estar relacionados com sistemas operacionais, dando início aos sistemas operacionais de rede, como a *Novell Netware* e a *Microsoft LAN Manager*, conforme expõe [MACHADO-07]. Ocorreu enorme difusão de aplicações voltadas à correio eletrônico, transferência de arquivos e acesso a bancos de dados remotos. A computação distribuída passou a integrar o ambiente corporativo, principalmente com soluções baseadas no modelo cliente/servidor.

A consolidação de sistemas operacionais baseados em interfaces gráficas ocorreu na década de 1990. O surgimento do WWW (*World Wide Web*), nesta ocasião causou enorme popularidade nas aplicações de computação distribuída. Esta década foi marcada pelo surgimento de soluções voltadas à *software* de fonte aberto, com a fundação da OSI (*Open Source Initiative*).

A partir da década de 2000, o conceito de processamento distribuído teve aplicação cada vez mais acentuada, conforme menciona [MACHADO-07]. A necessidade de interligar aplicações distintas e heterogêneas, simplificando a comunicação entre várias arquiteturas divergentes aliados à *WebService* fizeram de soluções como APIs (*Application Programming Interfaces*) e *middleware*, ferramentas indispensáveis no ambiente computacional.

¹ Uma norma do CCITT que descreve como os dados são tratados ao serem recebidos, e como os computadores podem acessar uma rede de comutação de pacotes. [DERFLER-95].

A divisão de responsabilidades entre os componentes de um sistema e a atribuição destes com os diversos computadores de uma rede talvez seja o aspecto mais evidente do projeto de sistemas distribuídos. Um sistema distribuído segundo [TANENBAUM-07] pode ser definido como “uma coleção de computadores independentes que se apresenta aos seus usuários como um sistema único e consistente”.

Para interligar computadores e redes heterogêneas de forma simultânea, os sistemas distribuídos comumente podem ser dispostos por meio de uma camada de *software* de alto nível, que propicia a comunicação de dados entre as aplicações. A figura 06 abaixo ilustra o sistema distribuído organizado como camada de *middleware*. [TANENBAUM-07].

Figura 06: Sistema distribuído organizado como *Middleware*

Fonte: [TANENBAUM-07]

A presença de computador autônomo associado com redes apresenta entre outras vantagens a possibilidade de integração de sistemas que executam aplicações distintas. [COULOURIS-07] define sistema distribuído como ambiente onde “os componentes de *hardware* ou *software*, localizados em computadores, interligados em redes, se comunicam e coordenam suas ações apenas enviando mensagens entre si”.

Cabe salientar que o conceito que define sistema distribuído não faz menção quanto à distância imposta aos componentes do sistema, que podem variar de computadores pessoais, sistemas destinados à CPD (Centros de Processamento de Dados) até mesmo sistemas indústrias automatizados e informatizados.

O desempenho de sistemas distribuídos analisado pelo aspecto da distância entre seus componentes tem sido continuamente aprimorado, devido aos conseqüentes esforços no desenvolvimento e aperfeiçoamento de redes e protocolos destinados às redes de computadores, que têm crescido de forma acentuada, conforme tabela 01 abaixo. Este crescimento, porém, traz uma característica relevante quanto à interoperabilidade destes sistemas, fazendo com que os mesmos sejam extensíveis, ou seja, se o sistema prove serviços em conformidade com regras padronizadas que descrevam a sintaxe e a semântica de tais serviços. [COULOURIS-07].

Tabela 01: Computadores com IP registrados na Internet

Data	Computadores	Servidores Web
Dezembro de 1979	188	0
Julho de 1989	130.000	0
Julho de 1999	56.218.000	5.560.866
Janeiro de 2003	171.638.297	35.424.956

Fonte: [COULOURIS-07]

Inicialmente, os computadores eram vistos como dispositivos seqüenciais, capazes somente de realizar operações de forma seqüencial, entretanto, com a implementação de sistemas multiprocessadores, elevou-se de forma exponencial o poder computacional, reduzindo-se o tempo de processamento de diversas aplicações. Segundo [DEITEL-05], que define sistema de multiprocessamento como “qualquer sistema que contenha mais de um processador. Entre os exemplos de multiprocessadores estão os computadores pessoais de dois processadores, servidores de alta capacidade que contém muitos processadores e grupos distribuídos de estações de trabalho que trabalham juntas para executar tarefas.”

Em 1966, conforme expõe [MACHADO-07], o modelo proposto por *Michael Flynn* define os sistemas computacionais em função do grau de paralelismo, sendo:

- SISD (*Single Instruction Stream, Single Data Stream*). Este tipo caracteriza o tipo mais elementar de arquitetura, que suporta fluxo único de instruções e fluxo único de dados. Esta arquitetura enquadra a maioria dos computadores pessoais que estão baseados em monoprocessadores. Técnicas como *pipeline*, VLIW (*Very Long Instruction Word*) e *Hyper-*

Threading (desenvolvida pela empresa *Intel*) possibilitam inserir paralelismo em computadores com arquitetura SISD.

- SIMD (*Single Instruction Stream, Multiple Data Stream*). Arquiteturas definidas como SIMD consistem em uma ou mais unidades de processamento. Estas arquiteturas possuem benefícios quando as instruções executadas e possuem elevado grau de paralelismo de dados, onde são efetuadas reiteradas operações aritméticas. Computadores providos com processadores vetoriais e processadores matriciais utilizam arquitetura SIMD e são conhecidos como supercomputadores, capazes de realizar elevadas quantidades de operações.
- MISD (*Multiple Instruction Stream, Single Data Stream*). Arquiteturas com fluxo múltiplo de instruções e fluxo único de dados não são comumente encontrados, onde diversas unidades de processamento atuam sobre o mesmo fluxo de dados.
- MIMD (*Multiple Instruction Stream, Multiple Data Stream*). Esta classe de arquitetura enquadra sistemas com fluxo múltiplo de instruções e fluxo múltiplo de dados, caracterizadas por unidades com multiprocessamento atuando de forma independente sobre fluxo de dados diversos e separados.

Conforme expõe [MACHADO-07], as arquiteturas baseadas em MIMD, podem ser classificadas quanto ao grau de acoplamento de um sistema, ou seja, fatores como compartilhamento e tempo de acesso à memória principal, métodos de comunicação e velocidade dos processadores, possibilitam que estas arquiteturas sejam classificadas em sistemas fortemente acoplados e fracamente acoplados.

A figura 07 a seguir ilustra estes sistemas e suas respectivas subdivisões.

Legenda:

MIMD - Multiple Instruction Stream, Multiple Data Stream

SMP – Symmetric Multiprocessors

NUMA – NonUniform Memory Access

CC-NUMA – Cache-Coherent NonUniform Memory Access

COMA – Cache Only Memory Architecture

Figura 07: Sistemas com múltiplos processadores

Fonte: Adaptado de [MACHADO-07]

Em sistemas caracterizados como fortemente acoplados (*tightly coupled system*), possuem uma única memória principal, que é compartilhada por todos os processadores que utilizam um único sistema operacional centralizado e os sistemas definidos como fracamente acoplados (*loosely coupled system*), são conectados por uma rede de comunicação, onde existem vários processadores, sistemas operacionais e memória principal. Estes sistemas apresentam características de maior flexibilidade e escalabilidade em relação aos sistemas fortemente acoplados. [MACHADO-07]

Os sistemas fortemente acoplados, denominados também por arquiteturas de multiprocessadores, se subdividem em: SMP e NUMA.

Os sistemas SMP (*Symmetric Multiprocessors*), que também são definidos UMA (*Uniform Memory Access multiprocessor*), são arquiteturas de multiprocessador de acesso uniforme à memória principal; este tipo de sistema

possui degradação de desempenho, conforme expõe [DEITEL-05], em sistemas de grande escala.

Outra vertente de sistemas fortemente acoplado são os sistemas NUMA (*NonUniform Memory Access*), que possuem elevado grau de escalabilidade, onde a memória principal é fisicamente distribuída entre os diversos processadores, existindo, porém um único espaço de endereçamento. A tabela 02 abaixo ilustra características de sistemas NUMA.

Tabela 02: Características de sistemas NUMA

Sistema	Fabricante	Número de processadores	Topologia
CM-2	<i>Thinking Machines</i>	1024 a 4096	Cubo
<i>Paragon</i>	<i>Intel</i>	4 a 2048	Grid 2D
T3D	<i>Cray</i>	16 a 2048	Torus 3D
SP-2	<i>IBM</i>	2 a 512	Árvore
CM-5	<i>Thinking Machines</i>	32 a 2048	Árvore

Fonte: [MACHADO-07]

Os sistemas de multiprocessadores de arquitetura de memória somente de *cache*, definidos por COMA (*Cache Only Memory Architecture*), são considerados uma variação da arquitetura NUMA, onde cada processador, com seu respectivo *cache*, está associado com uma fração da memória global compartilhada; não existe uma hierarquia de memória em cada processador, todos os *caches* formam um espaço de endereçamento global, conforme expõe [PITANGA-04]. Outra variante de sistemas NUMA são os sistemas de multiprocessadores escaláveis CC-NUMA, que possuem arquitetura de coerência com acesso não uniforme, onde a memória é dividida na mesma quantidade de blocos e de processadores, sendo que cada bloco é conectado via barramento a um processador com memória local.

Sistemas fracamente acoplados, conforme define [DEITEL-05] “comumente conectam componentes indiretamente por meio de enlace de comunicação.” Estes sistemas possuem como característica essencial maior flexibilidade e acentuada escalabilidade em comparação à sistemas fortemente acoplados. Estes sistemas também são designados como arquiteturas multicomputadores. Exemplos de sistemas desta natureza são Sistemas *Clusters*, que são constituídos por redes de interconexão, destinados à aplicações de alto desempenho.

Os Sistemas *Clusters* possuem elevada tolerância à falhas associados com alta disponibilidade; estes sistemas são utilizados em servidores de banco de dados, sistemas de *firewall* entre outras aplicações.

Outra exemplificação de sistemas fracamente acoplados são os Sistemas Operacionais de Rede (SOR), conforme expõe [MACHADO-07]. Atualmente estes sistemas são encontrados em redes locais (LANs) e redes metropolitanas, distribuídas geograficamente (WANs) e estão implementados em sistemas operacionais heterogêneos, possibilitando comunicação por meio da pilha de protocolos que formou a base da internet, os protocolos TCP/IP (*Transmission Control Protocol/ Internet Protocol*).

Outra vertente relevante de sistemas fracamente acoplados são os Sistemas Distribuídos; que são definidos por [PITANGA-04] “como conjunto de elementos que se comunicam através de uma rede de interconexão e que utilizam *software* de sistema distribuído. Cada elemento é composto por um ou mais processadores e uma ou mais memórias.”

Conforme salienta [MACHADO-07], um aspecto essencial na conceituação de sistemas distribuídos, onde se pode definir um sistema distribuído como fracamente acoplado pelo aspecto de *hardware* e fortemente acoplado pelo aspecto de *software*. Tais sistemas permitem elevada escalabilidade, implementação de sistemas de tolerância à falhas, que podem estar dispostos em vários níveis, além de permitir a implementação de aplicações distribuídas em função das necessidades dos usuários.

Cabe salientar a distinção existente entre sistemas de computação distribuídos, sistemas de informação distribuídos e sistemas embutidos distribuídos (também conhecidos por sistemas distribuídos perversivos), conforme expõe [TANENBAUM-07].

A figura 08 a seguir ilustra os tipos de sistemas distribuídos e suas respectivas subdivisões:

Figura 08: Sistemas distribuídos

Fonte: Adaptado de [TANENBAUM-07]

Sistemas de computação distribuídos, utilizados para computação de alto desempenho para sistemas distribuídos, que se subdivide em dois grupos:

- Sistemas de computação de *cluster*, que é utilizada na maioria das aplicações para programação paralela onde um único programa é executado em várias máquinas, tendo como aspecto típico a homogeneidade;
- Sistemas de computação em grade, definidos pelo alto grau de heterogeneidade, onde a colaboração entre aplicações distintas ocorre sob forma de uma organização virtual. Conforme propõe [FOSTER et al. 2001]¹ adup [TANENBAUM-07], uma arquitetura para este tipo de sistema é constituída em quatro camadas, sendo:
 - Camada base, que prove interfaces para recursos em locais em site específico;

¹ FOSTER, I.; KISHIMOTO, H. e SAVVA, A. "The Open Grid Service Architecture, Version 1.0" GGF Information Document GFD-I.030, janeiro 2005.

- Camada de conectividade, formada por protocolos de comunicação para suportar transações da grade que abranjam a utilização de múltiplos recursos;
- Camada de recursos, com subsídios da camada de conectividade, é responsável pelo gerenciamento de um único recurso, chama as interfaces que estão disponíveis pela camada base;
- Camada coletiva, que manipula o acesso a múltiplos recursos, além de realizar o escalocamento e alocação de tarefas para múltiplos recursos;
- Camada de aplicação, que realiza aplicações para o ambiente de computação em grade.

A figura 09 abaixo ilustra arquitetura computação em grade.

Figura 09: Arquitetura em camadas – computação em grade.

Fonte: [TANENBAUM-07]

O conjunto destas camadas, ou seja, camadas de coletividade, de conectividade e de recursos, definem conforme expõe [TANENBAUM-07], a essência de *middleware* em grade.

Sistemas de informação distribuídos caracterizam uma classe definida por sistemas distribuídos encontrados em corporações e instituições, que necessitaram buscar soluções para integração em aplicações de rede. Várias soluções de *middleware* existentes atualmente são procedentes do desenvolvimento com uma infraestrutura na qual era mais fácil integrar aplicações a um sistema de informações

de âmbito empresarial. Conforme expõe [TANENBAUM-07] pode-se distinguir vários níveis nos quais ocorreu a integração, algumas soluções propunham uma aplicação em rede, consistindo em um servidor que executava aplicação específica, freqüentemente incluindo um banco de dados e que a disponibilizava para programas remotos, definidos como clientes, que podiam enviar solicitações ao servidor para execução de tarefa determinada.

Ainda conforme [TANENBAUM-07], a integração ocorrida a baixo nível possibilitava que clientes empacotassem várias requisições possivelmente para diferentes servidores em uma única requisição maior, e posteriormente, as enviassem para execução como uma transação distribuída, onde todas as requisições deveriam ser executadas.

Com o surgimento de aplicações mais elaboradas, que eram gradativamente separadas em componentes independentes, distinguindo componentes de banco de dados de componentes de processamento, ficou evidente que a integração deveria ocorrer de modo que possibilitasse as aplicações se comunicar diretamente entre si, esta necessidade resultou nos modelos de integração entre aplicações empresariais definidos como EAI (*Enterprise Application Integration*), conforme expõe [TANENBAUM-07].

A EAI surgiu da necessidade de integrar aplicações COTS (*Commercial-Off-The-Shelf*), que surgiram em meados da década de 1990, oferecendo suporte às funções de negócios específicos, tais como fabricação, administração de pessoal e contabilidade, conforme aborda [CUMMINS-02]. As soluções de integração baseadas em EAI provem ferramentas, dispositivos e componentes para facilitação do processo de integração, por meio de adaptadores à sistemas legados.

Sistemas distribuídos pervasivos, também conhecidos como sistemas embutidos distribuídos, são sistemas que possuem como características a estabilidade e tamanho comumente reduzido; são compostos de um ou mais computadores pessoais que integram equipamentos eletrônicos de consumo, típicos como aparelhos de TV, equipamentos de áudio e vídeo, *smart phones*, PDAs (*Personal digital Assistants*) e outros equipamentos de uso pessoal, por isso são inerentemente distribuídos e que estão entorno da vida cotidiana, integrados em um único sistema.

Os requisitos para aplicações pervasivas devem prover mudanças contextuais, devido à modificação contínua do ambiente, como alteração entre estações-base, possibilitando conectar-se a outra rede; incentivar composição *ad hoc*, visto que usuários diversos utilizam de modos diferentes e reconhecer compartilhamento como padrão de forma facilitada.

Conforme expõe [MASCOLO et al. 2004]¹ adup [TANENBAUM-07] “na presença de mobilidade, tais dispositivos devem suportar a adaptação fácil e dependente de aplicação a seu ambiente local. Também devem ser capazes de descobrir serviços como eficiência e reagir de acordo.”

Exemplos de sistemas distribuídos pervasivos são encontrados em sistemas domésticos, onde atualmente a conexão, em muitos casos, é obtida por meio de padrões UPnP (*Universal Plug and Play*). No ambiente hospitalar, em sistemas e dispositivos eletrônicos destinados ao tratamento de saúde, freqüentemente são utilizados sistemas equipamentos com sensores organizados em rede de área corporal denominados por BAN (*Body-area Network*). Os dispositivos e redes de áreas corporais devem suportar processamento de dados na rede, onde os dados de monitoramento devem ser agregados, armazenados e enviados de forma permanente à centros médicos especializados. [TANENBAUM-07]

Outro exemplo de sistemas distribuídos pervasivos são as redes de sensores, que são utilizados na aquisição de dados e processamento de informações. Ainda conforme [TANENBAUM-07], que aborda rede de sensores, definindo-os como “as redes em malha, que em essência, formam um conjunto de nós que se comunicam por meio de ligações sem fio. Essas redes podem forma a base para muitos sistemas distribuídos de médio porte”. Tais dispositivos e sistemas são utilizados atualmente em sistemas *wireless* para comunicação e troca de dados e informações em uma rede LAN em ambientes corporativos; no contexto industrial, dispositivos com estas características são utilizados para interligar dispositivos de automação em ambientes hostis e de difícil acesso.

¹ MASCOLO, C.; CAPRA, L. e EMMERICH, W. “*Principles of Mobile Computing Middleware*”. In Mahamoud, Qusay H. (ed.), *Middleware for Communications*, capítulo 12. Nova York: John Wiley, 2004.

Assim, a computação distribuída consiste em adicionar o poder computacional de diversos computadores interligados por uma rede de computadores ou mais de um processador, atuando em conjunto no mesmo computador, para processar colaborativamente atividade determinada de forma coerente e transparente.

A união desses diversos computadores com o objetivo de compartilhar a execução de tarefas caracteriza os modernos sistemas distribuídos. O fator motivador para o desenvolvimento e aplicação de sistemas distribuídos é a necessidade de compartilhamento de recursos.

Usuários de sistemas operacionais distribuídos possuem a noção de que estão interagindo com uma máquina somente, enquanto, os sistemas operacionais de rede, devem estar conscientes da implementação distribuída. Conforme define [TANENBAUM-07], “um sistema distribuído deve ter por meta a abertura, onde o sistema deve prover serviços de acordo com regras padronizadas que descrevem a sintaxe e semântica desses serviços”. Estas regras são definidas em protocolos e descritas por meio de IDL (*Interface Definition Language*). Além deste requisito, é desejável que o sistema distribuído tenha características de:

- Transparência, em um sistema distribuído, os usuários não devem ser capazes de distinguir a localização dos recursos, por exemplo, sistema de arquivo distribuído remoto não deve ser diferente do sistema de arquivo local. A *ISO Open Distributed Processing Model*, define oito tipos de transparências de sistemas distribuídos, conforme aborda [DEITEL-05]:
 - Transparência de acesso;
 - Transparência de localização;
 - Transparência de falha;
 - Transparência de replicação;
 - Transparência de persistência;
 - Transparência de migração;
 - Transparência de relocação;
 - Transparência de transação.

- Flexibilidade, sistemas distribuídos devem ser construídos para redes, processos e linguagens de programação heterogêneas, desta forma, protocolos e sistemas *middleware* devem dissimular divergências existentes;
- Segurança, os recursos de um sistema distribuído devem ser protegidos contra agentes externos, e esta baseada em três componentes principais:
 - Confidencialidade,
 - Integridade,
 - Disponibilidade.
- Desempenho, apesar do custo computacional, a implementação de dispositivos de tolerância não devem afetar o desempenho de sistemas distribuídos, que podem ser mensurados por:
 - Número de mensagens;
 - Granulidade.
- Concorrência, característica relacionada a pedidos concorrentes para recursos, onde cada recurso deve ser definido para assegurar a consistência quando requerido;
- Escalabilidade, característica relacionada ao acréscimo de usuários e recursos de forma constante ao sistema, sendo, portanto uma das metas mais importantes no desenvolvimento de sistemas distribuídos. Esta característica pode ser mensurada sob aspectos de tamanho, em termos geográficos e administrativos.

3.1 Modelos de Arquiteturas de Sistemas Distribuídos

Os incrementos de velocidade e confiabilidade que a redes de dados obtiveram devido à constantes aprimoramentos, associados com o desenvolvimento de arquiteturas computacionais mais eficientes, tornaram os sistemas destinados à informação cada vez mais conectados. A necessidade de compartilhar recursos e dispor de sistemas integrados e tolerantes à falhas tornou os Sistemas Distribuídos difundidos em diversas arquiteturas.

A arquitetura de um sistema distribuído pode ser definida em função de sua estrutura, em termos de componentes especificados separadamente, conforme expõe [COULOURIS-07]. A organização de sistemas distribuídos, onde os componentes de *software* que compõem o sistema possuem como objetivo central assegurar que a estrutura atenda as demandas atuais e, provavelmente as solicitações futuras que serão impostas ao sistema, tendo como maior anseio tornar o sistema confiável, gerenciável, adaptável às implementações. Esta expectativa é atingida por meio das camadas de *software* que compõem os sistemas distribuídos.

Conforme aborda [COULOURIS-07], que define o termo arquitetura de *software* “como algo que se refere à estruturação do *software* em camadas ou módulos em um único computador e, mais recentemente, em termos de serviços oferecidos e solicitados entre processos localizados em um mesmo computador ou em computadores diferentes.” A necessidade de definição de uma arquitetura de *software* para sistemas distribuídos conduz necessariamente à determinação de uma arquitetura de sistema.

A figura 10 abaixo ilustra os principais componentes de *software* de sistemas distribuídos.

Figura 10: Camadas de *software* e *hardware* em sistemas distribuídos

Fonte: [COULOURIS-07]

A plataforma define os níveis baixos da arquitetura, oferecendo serviços aos níveis que estão localizados acima. Esta camada facilita a comunicação e coordenação entre processos. Os sistemas operacionais distribuídos fazem parte da

plataforma, provendo o gerenciamento de recursos localizados em vários computadores em rede, utilizando vários dos mesmos métodos de comunicação, estruturas e outros protocolos encontrados em sistemas operacionais de rede tornando a comunicação transparente. [COULOURIS-07]

A definição de arquitetura de sistemas distribuídos atende a estilos arquitetônicos, que podem ser especificados em função dos componentes que constituem as arquiteturas de sistemas distribuídos e pela forma de como estes componentes estão conectados. [TANENBAUM-07] classifica os estilos de arquiteturas de sistemas distribuídos em:

- Arquiteturas em camadas;
- Arquiteturas baseadas em objetos;
- Arquiteturas centradas em dados;
- Arquiteturas baseadas em eventos.

Os dispositivos que compõem sistemas distribuídos, comumente, possuem características heterogêneas, para dissimular a heterogeneidade faz-se necessária camada de programação, que ofereça abstração aos sistemas operacionais e linguagens de programação subjacentes assegurando portabilidade, transparência e interoperabilidade em sistemas distribuídos; estes requisitos são obtidos por meio de camadas de *software Middleware*.

A disposição dos componentes de *software* em sistemas distribuídos define arquiteturas de sistemas, classificadas em:

- Arquiteturas centralizadas;
- Arquiteturas descentralizadas;
- Arquiteturas híbridas.

Os sistemas distribuídos com arquiteturas centralizadas podem ser definidos pela simplicidade de implementação e gerenciamento, entretanto, são um estrangulamento em potencial, visto que, comumente o servidor central possui capacidade limitada e terá o desempenho afetado em virtude do aumento de demanda. Por outro lado, os sistemas descentralizados são escaláveis e robustos, porém isso demanda certa complexidade de implementação, principalmente nas

questões de tolerância à falhas e descoberta de recursos. Muitos sistemas distribuídos combinam características das duas arquiteturas, parte do sistema no tradicional modelo cliente-servidor e outra parte do modelo *peer-to-peer* definindo arquiteturas híbridas.

Estruturas híbridas são implantadas notavelmente em sistemas distribuídos colaborativos. A principal preocupação em muitos desses sistemas é como se juntar ao sistema, para o qual muitas vezes um esquema tradicional cliente-servidor é adotado. Uma vez que o nó se une ao sistema, pode-se utilizar um esquema totalmente descentralizado para colaboração.

3.2 Arquiteturas Centralizadas

Os sistemas distribuídos, com arquiteturas centralizadas, possuem como características, serem fortemente acoplados, tipo monolítico, onde não ocorre o particionamento núcleo, ou seja, os serviços do sistema e do núcleo fazem parte de um mesmo programa.

Conforme define [DEITEL-05], “o sistema operacional monolítico é a arquitetura de sistema operacional mais antiga e mais comum. Cada componente do sistema operacional é contido no núcleo e pode comunicar-se diretamente com qualquer outro. O núcleo normalmente é executado com acesso irrestrito ao sistema de computador.”

A figura 11 a seguir ilustra arquitetura de sistema operacional de núcleo monolítico.

Legenda:

GM – Gerenciador de Memória

EP – Escalonador de Processador

CEP – Comunicação entre Processos

SA – Sistema de Arquivos

E/S – Gerenciador de Entrada/Saída

GR – Gerenciador de Rede

Figura 11: Arquitetura de sistema operacional de núcleo monolítico

Fonte: [DEITEL-05]

Quando se aborda arquiteturas centralizadas, em sistemas distribuídos, a arquitetura cliente/servidor possui ampla utilização, tendo sido bastante implementada ao longo dos anos por diversas aplicações.

Conforme expõe [RENAUD-94], “sistemas cliente/servidor são mais interessantes quando os processos cliente e servidor são executados em máquinas diferentes conectadas por meio de uma rede.” Um servidor é um processo que implementa um serviço específico, e um cliente é um processo que solicita um serviço de um servidor por meio de uma requisição. Como exemplo de aplicação do modelo cliente-servidor, podem-se citar sistemas de servidor para banco de dados distribuídos, que interage continuamente com clientes, retransmitindo informações requisitadas.

A figura 12 a seguir ilustra arquitetura cliente/servidor, composta por três camadas.

Figura 12: Sistema cliente/servidor.

Fonte: [RENAUD-94]

Conforme expõe [TANENBAUM-07], várias aplicações cliente-servidor podem ser construídas em conformidade com partes distintas da arquitetura: uma parte que manipula a interação com um usuário, uma parte que atua sobre um banco de dados e uma parte que contém a funcionalidade central de uma aplicação.

Exemplificações deste modelo de arquitetura podem ser encontrados em dispositivos de busca na Internet, em sistemas de banco de dados implementados em redes locais; em função do ambiente de negócio da corporação onde tais sistemas estejam atuando, o nível dos dados pode ser organizado como banco de dados relacional.

A figura 13 a seguir ilustra aplicação de banco de dados por arquivo compartilhado e cliente/servidor.

Figura 13: Aplicação de banco de dados por arquivo compartilhado e cliente/servidor

Fonte: [RENAUD-94]

Neste modelo de arquitetura computação distribuída, o controle e gerenciamento das operações do sistema ocorrem de forma centralizada, provendo possibilidade de redução de custos comparativamente aos sistemas descentralizados, porém, conforme expõe [LOPES-00], estes modelos não são tão confiáveis quanto os sistemas descentralizados, em que as estações de trabalhos atuam de forma independente e com redundância.

3.2.1 Cliente/Servidor

O modelo cliente/servidor é uma arquitetura computacional que envolve requisições de serviços de clientes para servidores, desta forma, a definição para cliente/servidor seria a existência de uma plataforma base para que as aplicações, onde um ou mais clientes e um ou mais servidores, juntamente com o sistema operacional da rede, executem processamento distribuído. O sistema poderia ser entendido como a interação entre *hardware* e *software* em diferentes níveis, implicando na composição de diferentes computadores e aplicações.

O modelo cliente/servidor pode ser visto como um “paradigma de programação” que representa as interações entre os processos e a estrutura do

sistema. Este sistema é utilizado para melhorar a estrutura do sistema operacional, movendo-se a maior quantidade possível de funções para níveis mais altos do sistema, reduzindo, portanto o tamanho do núcleo.

A estação servidor disponibiliza recursos de uma estação aos usuários da rede, dentre alguns tipos de servidores, podem ser citados: [SOARES-99]

- Servidor de arquivos;
- Servidor de banco de dados;
- Servidor de impressão;
- Servidor de comunicação;
- Servidor de gerenciamento.

O processo cliente é ativo, emitindo solicitações ao servidor utilizando Sistema Operacional de Rede (SOR). A estação cliente pode interagir com único servidor ou com vários, conforme expõe [SILVERSCHATZ-99], o “cliente possui a responsabilidade de manter e processar o diálogo com o usuário, além de realizar a validação dos dados.” O processo servidor possui característica reativa, oferecendo serviços mediante solicitação às estações cliente, via SOR. Comumente, um servidor possui função específica, podendo ser acessado por várias estações clientes simultaneamente.

O modelo cliente/servidor pode ser implementado em camadas.

Nos sistemas dispostos em duas camadas, a GUI (*Graphical User Interface*) e a lógica de aplicação são executados na estação cliente, enquanto o servidor SQL (*Structure Query Language*) é executado na estação servidor.

Para a arquitetura disposta em três camadas, a GUI é processada na estação cliente, a lógica de aplicação é executada em um servidor de aplicação que atua como um cliente para o servidor SQL. [MENASCE-02]

A figura 14 a seguir relaciona os componentes da arquitetura cliente/servidor com Modelo de Referencia OSI.

Figura 14: Relação entre Modelo de Referência OSI e Sistema Operacional de Rede (SOR).

Fonte: Adaptado de [SOARES-99]

Em 1998, [ORFALI-98] abordava na ocasião que as próximas gerações de tecnologias cliente/servidor estarão baseadas em objetos distribuídos, como CORBA e JAVA; e define a próxima “onda” como *Web Technology*. As “ondas” da evolução da arquitetura cliente/servidor são ilustradas na figura 15 abaixo:

Figura 15: As “ondas” da arquitetura cliente/servidor

Fonte: [ORFALI-98]

3.3 Arquiteturas Descentralizadas

Nas últimas décadas, tem crescido de forma acentuada a necessidade de sistemas computacionais escaláveis e que apresentem maior tolerância à falhas, permitindo o compartilhamento de recursos em larga escala, fato considerado natural, devido ao aumento de demanda por serviços *online* associado diretamente ao crescimento da Internet que impulsionou diversas transações.

As redes de dados tornaram-se maiores e mais complexas, pela substituição de sistemas monolíticos por sistemas distribuídos; este crescimento está relacionado também ao processo de redução de custos que os serviços de telecomunicações atingiram nos últimos anos, tendo sido resultado de esforços no desenvolvimento de políticas de integração de usuários na Internet devido ao enorme potencial mercantil, social e cultural.

Conforme expõe [COULOURIS-07], “é esperado que a demanda por serviços na Internet cresça em uma escala limitada apenas pelo tamanho da população mundial.” Esta afirmação pode ser complementada pela inclusão de novas populações que atualmente não têm acesso a Internet, seja por motivos econômicos, políticos e sociais.

Em arquiteturas descentralizadas, a interação entre os sistemas operacionais de rede, que estabelecem a comunicação entre as estações da rede, não havendo necessidade de servidor central de gerenciamento; nesta arquitetura, todas as estações possuem o módulo cliente do sistema operacional (SORC) e o módulo servidor do sistema operacional (SORS). [SOARES-99]

Conforme expõe [TANENBAUM-07], o conceito de arquitetura de sistema distribuído descentralizado pode ser considerada uma evolução do modelo de arquitetura distribuída cliente/servidor multidividas (distribuição vertical/distribuição horizontal), onde o *peer-to-peer* define uma classe de arquitetura de sistemas que suporta distribuição horizontal.

Ainda conforme [TANENBAUM-07], “de uma perspectiva de alto nível, os processos que constituem um sistema *peer-to-peer* são todos iguais, o que significa que as funções que precisam ser realizadas são representadas por todo processo

que constitui o sistema distribuído; como consequência, grande parte da interação entre processo é simétrica.”

As redes onde os nós formam processos e os enlaces formam os caminhos, definem uma rede de sobreposição, na qual as redes *peer-to-peer* se desenvolveram.

Existem dois tipos de redes de sobreposição (*peer-to-peer*): estruturadas e não estruturadas. [TANENBAUM-07]

A figura 16 abaixo ilustra a arquitetura *peer-to-peer*.

Figura 16: Arquitetura *Peer-to-Peer*

Fonte: Adaptado de [SOARES-99]

Conforme expõe [COULOURIS-07], que definiu *peer-to-peer* como “aplicativos que exploram os recursos disponíveis nos limites da Internet – armazenamento, ciclos de processamento, conteúdo, presença humana.”

Diferente de arquiteturas centralizadas tradicionais, os sistemas *peer-to-peer* fornecem acesso e recursos de informação localizados em computadores de toda uma rede, seja uma rede corporativa ou mesmo a Internet. Os sistemas *peer-to-peer* compartilham as seguintes características: [COULOURIS-07]

- Garantia que cada usuário contribua com recursos ao sistema;
- Todos os nós em um sistema *peer-to-peer* possuem as mesmas capacidades e responsabilidades funcionais;

- A administração do sistema deve ocorrer de forma descentralizada;
- Devem ser projetados de forma a proporcionar um grau limitado de anonimato para provedores e usuários dos recursos;
- Equilíbrio da carga de trabalho e garantia de disponibilidade sem adição de sobrecargas indevidas por meio da escolha correta de algoritmo de disposição de dados em vários *hosts*.

3.3.1 Peer-to-Peer

A primeira utilização da expressão *peer-to-peer* ocorreu em 1984, com o desenvolvimento do projeto *Advanced Peer-to-Peer Networking Architecture*, na IBM. Conforme expõe [TANENBAUM-07], as redes *peer-to-peer* se desenvolveram em torno do paradigma do conceito de rede de sobreposição, ou seja, uma rede onde os nós são constituídos pelos processos e os enlaces representam os canais de comunicação, onde as estações são interconectadas por grafos, interconectados por arestas, que simbolizam as conexões entre as estações da rede.

A partir de 1999, as redes *peer-to-peer* surgiram de forma acentuada, quando número significativo de usuários adquiriram conexões de banda larga. Em meados do ano de 2004, o número de conexões de banda larga da internet ultrapassava 100 milhões. O desenvolvimento dos sistemas *peer-to-peer* podem ser classificados em três gerações: [COULOURIS-07]

- Primeira geração, definida pelo lançamento de serviços de músicas;
- Segunda geração, caracterizada pelo compartilhamento de arquivos em maior escalabilidade, tolerância de falhas, esta geração introduziu o conceito *Super-peers*. (*peers* com altas taxas de transmissão);
- Terceira geração, caracterizada pelo *Gnutella*¹ e *BitTorrent*².

¹ Gnutella – é um protocolo distribuído para buscas, modelado para redes não estruturadas e descentralizadas, cujo sistema é composto por *servents* que acumulam função de cliente e servidor simultaneamente. [FONSECA, Hubert. Compartilhamento de Conteúdo em Redes *Peer-to-Peer*]

² BitTorrent – é um sistema *peer-to-peer* de transferência de arquivos (*download*), que consiste em reunir arquivos a partir de porções fracionadas de outros usuários (conceito de colaboração) [TANENBAUM-07].

A arquitetura da rede *peer-to-peer* é constituída pelas conexões entre os *softwares* nas estações que pertencem ao sistema, e independe da arquitetura da rede física, podendo ser caracterizada em: não estruturada ou estruturada.

Alguns sistemas *peer-to-peer* possuem nós com funções específicas, onde os sistemas possam ser caracterizados pelo nível de centralização em arquiteturas puramente descentralizadas, parcialmente centralizadas ou híbridas.

[COULOURIS-07] cita que a função de um *middleware peer-to-peer* é simplificar a construção de serviços implementados em vários *hosts*, em uma rede amplamente distribuída.

Para que isto seja obtido, o *middleware* deve possibilitar aos clientes localizar e se comunicar com qualquer recurso disponível, mesmo que este recurso esteja amplamente distribuído entre os *hosts*.

Conforme aborda [COULOURIS-07], prover mecanismos que possibilitem aos clientes acessarem recursos de dados de forma rápida e segura é um problema relevante no projeto de aplicativos *peer-to-peer*. Além de requisitos funcionais, o *middleware peer-to-peer* deve tratar requisitos não funcionais, tais como:

- Escalabilidade global;
- Balanceamento de carga;
- Otimização das interações entre *peers*;
- Segurança de dados em ambientes heterogêneos;
- Anonimato.

Conforme finaliza [COULOURIS-07], expondo que “o projeto de uma camada *middleware* para suportar sistemas *peer-to-peer* em escala global é problema difícil; os requisitos de escalabilidade e disponibilidade tornam impraticável manter um banco de dados em todos os nós clientes....”.

3.4 Banco de dados

As informações geradas no processamento de dados em sistemas computacionais são armazenadas em repositórios definidos por banco de dados. Sistemas de bancos de dados são desenvolvidos para prover o gerenciamento de

grandes quantidades de informações, onde são utilizados Sistemas Gerenciadores de Banco de Dados (SGBD), conforme define [SILVERSCHATZ-99].

No atual ambiente corporativo, os bancos de dados podem ser vistos como sistemas de gestão que possibilitam que diversos processos e usuários utilizem a mesma base de informação sob necessidades diversas.

Conforme expõe [SILVERSCHATZ-99], um sistema de banco de dados oferece o benefício de proporcionar aos usuários deste sistema uma visão abstrata dos dados, ocultando detalhes sobre maneiras de armazenamento e manutenção destes dados.

Existem vários tipos de bancos de dados disponíveis, dos quais se destacam:

- Banco de dados relacional. Este sistema de banco de dados se consolidou como pioneiro em aplicações comerciais. A estrutura do modelo de banco de dados relacional se constitui de coleção de tabelas. Conforme expõe [LAUDON-98] define banco de dados relacional “como o modelo relacional algo como tipo de modelo lógico de banco de dados que tratam dados como se tivessem sido armazenados em tabelas bidimensionais.” A metodologia de consulta de dados utiliza linguagens procedurais e não procedurais e modelos definidos por álgebra relacional. Na década de 1970, a IBM desenvolveu a linguagem SQL (*Structure Query Language*), inicialmente utilizado para consulta em bancos de dados relacional, esta linguagem foi incorporada a outros construtores de bancos de dados. O sistema utiliza uma linguagem declarativa que baseada em três comandos básicos: *Select*, *From* e *Where*.
- Banco de dados orientado à objetos é uma adequação ao paradigma de programação orientada à objetos, este modelo surgiu para atender as exigências destas aplicações como: Banco de Dados Hipertexto, Banco de Dados Multimídia, *Office information systems* (OIS), *Computer-Aided Design* (CAD). Segundo [LAUDON-98], banco de dados orientado à objetos “é uma abordagem ao gerenciamento de dados que armazenam tanto dados quanto procedimentos atuando sobre os dados como objetos que podem automaticamente ser recuperados e compartilhados; os objetos podem conter multimídia.”

- Banco de dados hierárquico é uma coleção de registros conectados uns aos outros por meio de rede de comunicação. O conjunto dos registros é organizado na forma de uma árvore estruturada em segmentos: raiz, pai e filhos.

A figura abaixo 17 ilustra este tipo de arquitetura.

Figura 17: Estrutura de banco de dados hierárquico

- Banco de dados em rede é considerado uma extensão do modelo hierárquico, é uma coleção de registros conectada uns aos outros também por redes de comunicação, o modelo em rede estabelece uma relação membro-proprietário. Segundo [LAUDON-98], define banco de dados em rede “como modelo lógico de banco que é útil para lidar com relacionamentos muitos-para-muitos.”

Conforme expõe [SILVERSCHATZ-99], a arquitetura de um sistema de banco de dados é fortemente influenciada pelo sistema básico computacional sobre o qual o sistema de banco de dados é executado. Este fato influencia na arquitetura dos bancos de dados, que atualmente pode ser classificados em:

- Sistema centralizado, caracterizados por sistemas de banco de dados que atuam em um único computador. Computadores pessoais e estações de trabalhos enquadram-se nesta categoria. Nesta configuração o *front-end*¹ e *back-end*² são executados em um único sistema. [SILVERSCHATZ-99]
- Sistema cliente/servidor, caracterizados pelo desenvolvimento das redes de computadores e dos protocolos de compartilhamento de recursos que aproximaram aplicações de *front-end* e *back-end* para estações cliente e servidor por meio da linguagem SQL ou de um programa de aplicação de comunicação (*Middleware*).
- Sistemas paralelos, caracterizados por múltiplos processadores e múltiplos discos interconectados por redes de alta velocidade e que possuem capacidade elevada de processamento em relação aos sistemas centralizados e cliente/servidor. O desempenho de sistemas paralelos pode ser avaliado por meio de *throughput* e pelo tempo de resposta do sistema para concluir uma tarefa específica. Parâmetros como aceleração (redução de tempo de execução de uma tarefa por meio do aumento do grau de paralelismo) e escalabilidade (manuseio de maior número de tarefas por meio de aumento do grau de paralelismo) são considerados na avaliação de sistemas paralelos. [SILVERSCHATZ-99].

¹ *Front-end* – consiste em ferramentas como formulários, gerador de relatórios e recursos de interface gráfica. [SILVERSCHATZ-99].

² *Back-end* – executa o gerenciamento das estruturas de acesso, desenvolvimento e otimização de consultas, controle de ocorrências e recuperação. [SILVERSCHATZ-99].

- Sistemas distribuídos são caracterizados pelo armazenamento de dados em diversos computadores como um conjunto de banco de dados parciais independentes que compartilha um esquema comum e que coordena o processamento de transações com acesso de dados não local. O banco de dados em sistemas distribuídos distingue-se dos sistemas paralelos pela distância geográfica que separam os componentes do sistema. Os processadores comunicam-se uns aos outros por meio de redes de comunicação que definem o roteamento e as estratégias de comunicação. Neste modelo de banco de dados, fatores como replicação, fragmentação devem ser considerados no armazenamento distribuído de dados.

Sistemas de bancos de dados distribuídos robustos necessitam detectar falhas e possuir capacidade de reconfiguração do sistema, de forma que o sistema recupere a condição original após restabelecimento do processamento de informações, isto se deve ao fato de que sistemas distribuídos trabalham com algoritmos distribuídos que conseqüentemente necessitam de um coordenador; caso este coordenador venha a falhar em conseqüência de uma falha no site onde este esteja instalado; após o reinício do processamento, o sistema deverá restabelecer as condições de funcionamento.

No segmento de tecnologia da informação, existem atualmente bancos de dados específicos desenvolvidos por *Business Information Warehouse*, destinados ao conceito de *Consumer Oriented*, como o *Operating Data Storage* (ODS), *Data Warehouse*, *Data Mart*, *Data Mining*, *On-line Analytical Processing* (OLAP) e sistemas ERP (*Enterprise Resource Planning*); todos desenvolvidos para integrar dados e informações nos diversos segmentos no ambiente corporativo voltado as necessidades específicas.

Para [BERRY-97], sistemas *Data Warehouse* devem ser visto como “um conjunto de dados de diferentes fontes, em um formato comum com definições consistentes de campos e chaves”, ainda conforme [BERRY-97], que também define *Data Mining* como “exploração e análise, por meios automáticos ou semiautomáticos, de grande quantidade de dados no sentido de descobrir meios e regras significativas.”

Sistemas OLAP segundo [ARAUJO-07] se referem “a um conjunto de ferramentas voltadas para acesso e análise *ad hoc* de dados, com o objetivo final de transformar dados em informações capazes de dar suporte às decisões gerenciais de forma amigável e flexível ao usuário e em tempo hábil.”

Conforme expõe [CORNACHIONE-01], com “o avanço da tecnologia de informação associada ao substancial aumento da complexidade dos processos empresariais em ambientes corporativos, a integração dos dados das corporações passa a ser condição necessária para suportar as necessidades de planejamento e decisões estratégicas.” Neste contexto a integração de sistemas e dados apresenta-se como solução para obtenção, tratamento de informações e dados corporativos; tende-se a observar uma solução que privilegia um banco de dados único, compartilhado pelas diversas aplicações presentes nas empresas.

Conforme afirma [STAIR-04]¹ adup [MENASCE-02] que “se uma organização não possui dados ou a capacidade de processá-los, não terá condições de obter sucesso em grande parte de suas atividades empresariais.”

¹ STAIR, M. R. Princípios de sistemas de informação: uma abordagem gerencial. 4ª edição. Rio de Janeiro: LTC, 2004.

4. TECNOLOGIA DA INFORMAÇÃO

O atual panorama dos Sistemas de Informação destinados ao ambiente corporativo é resultado da evolução dos negócios e da tecnologia aplicada com o passar dos anos. A tecnologia da informação traz para empresas e organizações a capacidade de melhorar a qualidade e disponibilidade de informações relevantes para a produtividade, gerando eficiência aos processos.

A gestão estratégica da informação transformou-se numa parte essencial e crítica a qualquer estrutura organizacional de sucesso. Esta gestão deve ocorrer de forma integrada e que possibilite a inserção de novas tecnologias que certamente serão desenvolvidas.

Conforme expõe [BIO-08], “um sistema aberto pode ser compreendido como um conjunto de partes em constante interação (o que ressalta um dos aspectos fundamentais da idéia de sistemas: a interdependência entre as partes), constituindo um todo orientado para determinados fins e em permanente relação de interdependência com o ambiente externo.” Este conceito apresenta com propriedade a relevante necessidade de abertura de sistemas, especificamente nos Sistemas de Informação destinados às aplicações corporativas; tais sistemas tornam iminente a necessidade de reavaliação de processos para que a implementação da TI (Tecnologia da Informação) possa proporcionar um aumento da satisfação de usuários.

Vários aspectos devem ser analisados para contextualizar a constituição de organizações corporativas, dentre as quais podem ser citados a interação entre subsistemas que compõe grande parte dos modelos de gestão corporativas, que atualmente estão apoiados em sistemas de informação e comunicação: [BIO-08]

- Sistema físico-operacional;
- Sistema de informação;
- Sistema social;
- Sistema de gestão.

A figura 18 a seguir apresenta a interação entre estes segmentos dentro do contexto de sistema organizacional.

Figura 18: Sistema corporativo e seus subsistemas

Fonte: Adaptado de [BIO-08]

Cada um dos subsistemas do processo corporativo compõe-se de uma rede de subsistemas e processos interdependentes. Em uma empresa do segmento industrial, por exemplo, o subsistema de Planejamento e Controle de Produção deve interagir junto às áreas de vendas, controle de materiais, estoque e expedição de produtos, subsidiando o nível gerencial da empresa por meio de dados e informações. Conforme expõe [BIO-08], no contexto atual, os recursos computacionais são imprescindíveis no processo de gestão administrativa, onde “o computador é um recurso transformador de dados e informações...”.

A evolução da tecnologia contribuiu também para o processo de fragmentação de sistemas, ou seja, quando novos sistemas foram desenvolvidos, antigos sistemas foram abandonados ou não foram integrados de forma correta a nova tecnologia. Deve haver, portanto a definição e adoção de um Plano Estratégico em Tecnologia da Informação. Segundo [SIMCSIK-02], que define MIS (*Management Information System*) como “todo o processo metodológico de coleta de dados e geração de informação, comunicadas para os níveis gerenciais e de direção da empresa, que os utilizam nos processos decisórios e nas comunicações organizacionais.”

Os Sistemas em Informação podem ser classificados em dois principais grupos: [BIO-08]

- Sistemas de apoio às operações;
 - Processos de transações;
 - Tomada de decisões direcionadas à operação.
- Sistemas de apoio à gestão.
 - Tomada de decisões estratégicas.

Segundo [SIMCSIK-02] “dentro das empresas existem dois tipos de processamento de informações. Um baseia-se em transações e é utilizado operacionalmente, o outro trabalha com sistemas de apoio às decisões.” Os sistemas de apoio às decisões operam atualmente com uma tecnologia que já existe há um bom tempo e que agora ganha força impulsionada pela evolução do *hardware* e *software*. Trata-se do conceito denominado *Data Warehouse*, no qual as informações são preparadas para o uso de pessoas que tomam decisões.

O ambiente de *Data Warehouse* integra informações provenientes de uma grande variedade de bancos de dados operacionais em um único banco de dados lógico, que pode ser visto com um repositório central desenvolvido para facilitar o processo de suporte à decisão. Sistemas reduzidos de *Data Warehouse* são definidos por *Data Marts*, tendo sido concebidos para aplicações específicas de armazenamento e análise de dados. [SIMCSIK-02]

Dada à diversidade de soluções de sistemas de informação, especialmente as empresariais, a base de sustentação pelo âmbito da Tecnologia da Informação pode ser composta por vários grupos, dos quais se destaca *hardware*, *software*, banco de dados e telecomunicações.

A figura 19 a seguir apresenta estes quatro componentes dentro do contexto da Tecnologia da Informação.

Figura 19: Tecnologia da informação e sistemas de informação

Fonte: Adaptado de [BIO-08]

Conforme expõe [SIMCSIK-02], analisando o âmbito industrial, o controle do sistema industrial implica em um sistema de informações qualificado, com ferramentas de assistência à tomada de decisões, que devem proporcionar:

- Sistema de planejamento a longo e médio prazo, incorporando o máximo de informações, relacionando vendas provisionais e capacidade de produção;
- Planejamento em curto prazo e sistema provisional para organizar a fabricação;
- Ferramentas de consulta em tempo real aos dados do sistema produtivo;
- Análise estatística e ferramentas de simulação para situações envolvendo múltiplos parâmetros e que requerem rápida tomada de decisões.

Segundo [WANG-98], expõe que “a informação tecnologia pode ser a maior ferramenta dos tempos modernos, mas é o julgamento de negócios dos humanos que a faz poderosa.”

4.1 Novell

O sistema operacional *NetWare*, da empresa *Novell*, considerado um dos primeiros sistemas operacionais de rede a serem implementados em larga escala. O sistema foi desenvolvido como sistema operacional proprietário em 1983.

A concepção do sistema *NetWare* possibilita que o mesmo possa ser implementado em redes com configuração *token-ring*, estrela e barramento. Os servidores *Netware* provêm serviços a seus usuários como: arquivo, impressão, mensagens, aplicação e banco de dados. Para ter acesso aos recursos clientes *NetWare* enviam GNS (*GetNearestServer*) para localizar o Servidor, que estabelecem comunicação e confeccionam tabela SAP (*Service Advertising Protocol*), com recursos disponíveis. A figura 20 abaixo ilustra esta comunicação.

Figura 20: Comunicação entre cliente e servidor com Novell

Fonte: Adaptado de [DIOGENES-01]

Desde a concepção do sistema operacional *Netware*, a *Novell* desenvolveu soluções para comunicação para o sistema operacional baseado nos protocolos proprietários SPX (*Sequenced Packet Exchange*) e IPX (*Internetwork Packet Exchange*). Estes protocolos serviram de bases para outras soluções para redes, como o Windows NT. A figura 21 a seguir ilustra estes protocolos.

Legenda:

RIP – Routing Information Protocol

NLSP – NetWare Link Services Protocol

SAP – Service Advertising Protocol

NCP – NetWare Core Protocol

Figura 21: Modelo Novell x Modelo OSI

Fonte: Adaptado de [DIOGENES-01]

O protocolo IPX é utilizado pela *Novell* e atua na camada de rede, fornece um serviço de datagrama não orientado à conexão a seus usuários, isto é, seus pacotes são transmitidos sem que seja necessário estabelecer conexões e não são reconhecidos pelo destinatário.

Conforme expõe [DIOGENES-01], o IPX define a inter-rede e o endereço dos nós baseados nos números de rede que são atribuídos a cada segmento de rede. Os endereços IPX são constituídos por três componentes: o endereço da rede onde esta conectada a estação, o endereço da estação na rede e o endereço de uma porta (*socket*) que identifica o processo que esta executando na estação que enviou ou que ira receber o pacote.

O IPX implementa um esquema de roteamento inter-redes baseado em tabelas de rotas localizadas nos *gateways*. O *frame* IPX pode ser encapsulado para trafegar nos meios: *Ethernet*, *Token Ring* e FDDI (*Fiber Distributed Data Interface*); a figura 22 abaixo ilustra os tipos de encapsulamentos existentes para *Ethernet*.

Figura 22: Encapsulamentos existentes para IPX

Fonte: [DOWNES-98]

O protocolo SPX da arquitetura *NetWare* é um protocolo orientado à conexão, atua no nível de transporte. O protocolo implementa serviço de circuito virtual, ou seja, mecanismos de controle de erros, de fluxo e de sequenciação; o protocolo é capaz de supervisionar as transmissões de dados informando ao operador a falha em retransmissões.

O SPX utiliza algoritmo *sliding window* para controle de erros e de fluxo, a figura 23 ilustra este controle de fluxo por meio deste algoritmo, que realiza o controle de erros por meio de janelas deslizantes.

Figura 23: *Sliding Window* - controle de fluxo em protocolo

Fonte: Adaptado de [SOARES-99]

A concepção da rede *Novell Netware* esta baseada no modelo cliente-servidor; inicialmente o conceito deste sistema foi atuar como suporte de servidor de arquivos, conforme expõe [FOROUZAN-06], o *software* do servidor de arquivos *Netware* reside na camada de aplicação, tendo como parâmetro o Modelo de Referência OSI. A Novell foi a primeira empresa a lançar um sistema operacional de rede para compartilhamento real de arquivos [DERFLER-95].

A figura 24 abaixo ilustra a interface do servidor *Netware* com rede, utilizando em sua concepção de 1986, comandos em DOS (*Disk Operating System*).

Figura 24: Interface *Shell* da rede Novell

Fonte: [SCHATT-93]

O Novell *NetWare* oferece sistema de segurança para servidor de arquivos de quatro formas diferentes: [SCHATT-93]

- O procedimento de acesso (*login*), onde o controle de acesso é realizado por nome e senha válida do usuário;
- Os direitos de síndico, o administrador da rede estabelece direitos de acesso personalizado para cada usuário, definido a partir de então como síndico, em função das necessidades quanto aos recursos disponíveis na rede;
- Os direitos de diretórios;

- Os atributos de arquivos, onde a rede *NetWare* possibilita que usuários definidos executem alterações de arquivos.

Os sistemas de tolerância a falhas (STF – *System Fault Tolerance*) possuem funções de proteção dispostos em três níveis caso ocorram falhas no servidor de arquivos: [SHELDON-93]

- STF nível I (*Hot fix*);
- STF nível II (Disk mirroring/Disk duplexing);
- STF nível III (*Server duplexing*).

O sistema operacional de rede *Novell NetWare* possui duas formas de apresentação aos usuários: [CUSTODIO-95]

- *NetWare Life*, que pode ser instalado em todos os dispositivos da rede, onde todas as estações podem desempenhar funções de cliente e servidor simultaneamente, caracterizando a arquitetura *peer-to-peer*;
- *NetWare* v2.x, v3.x, v4.x, a serem instalados em estações servidores e estações clientes, caracterizando a arquitetura cliente/servidor, estabelecendo desta forma o compartilhamento dos recursos de rede.

4.2 SNA

A arquitetura SNA (*Systems Network Architecture*), foi anunciada em 1974 para comunicação de dados em sistemas de larga escala. O SNA sofreu processo evolutivo para acomodar as redes de computadores, que passou a incluir suporte para redes *peer-to-peer* e outros meios de protocolos de comunicação padronizados como *Ethernet*, FDDI, *frame relay*, entre outros.

O modelo SNA é uma coleção de componentes de *hardware* e *software* integrados para criar uma rede funcional. A quantidade e o tipo de *hardware* e *software* instalado definem a funcionalidade da rede SNA.

Esta arquitetura foi uma solução desenvolvida pela IBM para a evolução continua que sofrem os produtos de comunicação de dados, tanto em *hardware* como *software*; isso se deve a necessidade de padronização de uma linha de produtos, sem levar em consideração que a IBM estava focando uma conectividade

universal entre seus produtos, para que houvesse ideal compartilhamento de recursos, sem necessidade de reconfiguração da instalação. Antes da solução SNA, os terminais de usuários de sistemas *mainframe* eram conectados diretamente a programas de aplicativos específicos no computador central. A arquitetura SNA é o principal mecanismo de comunicação de dados para computadores desenvolvidos pela IBM, utilizado em aplicações de larga escala, o SNA é um modelo de referência que define como que os sistemas de informação podem transportar dados.

A figura 25 abaixo ilustra as operações fundamentais do modelo SNA, comparando-as aos níveis definidos pelo Modelo de Referência OSI. [DOMINGUES-02]

Figura 25: Modelo de Referência OSI e arquitetura SNA

Fonte: Adaptado de [DOWNES-98]

Conforme expõe [DOMINGUES-02], o SNA consiste de:

- Uma arquitetura que define regras e responsabilidades de entidades de comunicação dentro de uma rede;
- Uma especificação de protocolo, que define os formatos e fluxos de mensagens dentro dessas entidades.

SNA baseia-se nos seguintes conceitos fundamentais:

- SNA é uma rede hierárquica (*master-slave*)
- SNA utiliza rotas pré-definidas para transportar dados;
- ACF/VTAM (Access Control Field / *Virtual Telecommunications Access Method*) é o master da rede.

SNA é uma arquitetura que habilita comunicação entre unidades endereçáveis de rede. A comunicação tradicional do SNA envolve quatro entidades físicas: *Host*, *Front-end Processor*, controladores de Cluster e Terminais.

Conforme expõe [DOMINGUES-02], “o conceito de redes hierárquicas obedece à existência de dois elementos na arquitetura, uma entidade principal, definida por *master* e as entidades secundárias, denominadas por *slave*.” A arquitetura tradicional SNA centraliza o processamento, com características determinísticas no tráfego de dados na rede.

A figura 26 abaixo ilustra a arquitetura SNA.

Figura 26: Modelo Rede SNA

Fonte: Adaptado de [DOMINGUES-02]

Na arquitetura SNA, é utilizado Classe de Serviço, definida por COS (*Class of Service*) para assegurar características de prioridade de transmissão de dados, possibilitando tempo de resposta de alta prioridade.

A COS SNA designa as características do serviço de transporte de rede de uma determinada sessão e em função das necessidades do usuário, diferentes COSs podem ser especificadas em uma rede SNA. A COS fornece o mecanismo para determinar todas as rotas SNA e descreve os níveis de serviço aceitáveis para

uma sessão e especifica também a coleção de sessões características, incluindo o tempo de resposta, segurança e disponibilidade. [DOMINGUES-02]

O gerenciamento de redes na arquitetura SNA é realizado por meio do sistema *NetView*, que foi inserido em 1986; é constituída por uma estrutura de gerenciamento desenvolvida pela IBM para o gerenciamento de redes de computadores com grande capacidade de processamento, com base no protocolo de gerenciamento SNMP (*Simple Network Management Protocol*). O *NetView* é capaz de obter gerenciamento de dados solicitados como dados não solicitados, permitindo monitorar, controlar e reconfigurar redes SNA.

O roteamento de SNA tradicional é realizado por VTAM (*Virtual Telecommunications Access Method*) no *mainframe*, devido a incompatibilidades com fornecedores de roteadores, a IBM adequou a estrutura de roteamento por meio da Comunicação Não Hierárquica Avançada, definida por APPC (*Advanced Program-to-Program Communication*). A capacidade de roteamento no APPC é conhecida como Formação de Rede Não Hierárquica Avançada APPN (*Advanced Peer-to-Peer Networking*). [BENNETT-98a]

O roteamento SNA é realizado pelo *APPN IBM Advanced Peer-to-Peer Networking Routing*, que é uma arquitetura *peer IBM* que apóia a comunicação de serviços de roteamento em sistemas APPC (*Advanced Program-a-Program Comunnications*). Conforme expõe [DOWNES-98], o roteamento é dinâmico e esta baseado em um caminho de peso mínimo estabelecido a partir de sugestões recebidas de todos os nós de rede APPN. Cada nó da rede APPN é responsável por informar alterações em sua topologia local (isto é, o próprio nó e os links em anexo). A informação de topologia é transmitida até que todos os nós APPN possam recebê-los. Quando um nó recebe informações que já possui, este interrompe a transmissão de dados para outros nós e informações duplicadas são reconhecidas por meio de um sistema de verificação.

O Roteamento de Alto Desempenho, (HPR - *High Performance Routing*) é uma extensão da arquitetura APPN. Constituído de um conjunto de melhorias para melhorar o roteamento de dados APPN, aumentar a confiabilidade, facilitar a compatibilidade e migração, o HPR proporciona funcionalidade adicional pelos componentes: [DOMINGUES-02]

- RTP (*Rapid Transport Protocol*), protocolo orientado à conexão, desenvolvido para redes de alta velocidade, com baixas taxas de erro;
- ANR (*Automatic Networking Routing*), algoritmo desenvolvido para roteamento em redes de alta velocidade; por meio de rápida comutação de pacotes, o algoritmo elimina o armazenamento em nós intermediários de uma conexão, reduzindo tempo de roteamento.

As redes baseadas em SNA podem ser integradas a outras tecnologias de redes, como o TCP/IP.

Conforme aborda [DOMINGUES-02], os protocolos de rede tem sido determinados pelas aplicações escolhidas pelas organizações, fato que tem levado a uma inevitável propagação de protocolos de rede. Visto que arquiteturas de redes possuem comportamentos diferentes em situações adversas, o desempenho e aspectos para integração devem ser analisados sob aspectos específicos. Ainda conforme [DOMINGUES-02] propõe que “a análise de integração seja analisada considerando: perdas de comunicação, perda de *frame* de dados e retransmissão de *frames* recebidos de forma incorreta.”

A tabela 03 abaixo apresenta comparação entre redes SNA e TCP/IP.

Tabela 03: Comparação de arquiteturas de redes SNA e TCP/IP

Parâmetro	TCP/IP	SNA
Definição	Pré-definição.	Topologia dinâmica
Roteamento	Rotas pré-definidas.	Rotas são dinamicamente determinadas; Roteamento determinado por diversos tipos de algoritmos.
Integridade	Garantida pela rede e pela camada de transporte;	Determinação de erros no ponto final da transmissão;
	Capacidade de gerar resposta definida.	Baseada em datagramas, sem recuperação nas camadas de rede e transporte.
Priorização	Tabelas de classes de serviço;	Sem priorização,
	Prioridade baseada em sessão.	Priorização de roteamento em alguns roteadores,
Overhead	Nó de roteamento = 29 bytes; Nó periférico = 9 bytes	44 bytes de header.

Fonte: Adaptado de [DOMINGUES-02]

A integração entre SNA e TCP/IP pode ocorrer de várias formas por meio de soluções disponíveis no mercado atuando em diversos níveis na arquitetura de

comunicação, entretanto, o roteamento apresenta-se como forma viável de integração devido aos protocolos desenvolvidos, apesar de causar aumento de *overhead*; o TCP/IP utiliza para roteamento os protocolos RIP (*Routing Information Protocol*) e OSPF (*Open Shortes Path First*), a arquitetura SNA dispõe dos protocolos de roteamento VTAM e NCP.

4.3 Ethernet

A *Ethernet* esta enquadrada nas tecnologias de redes locais LAN (*Local Area Network*), sendo uma das redes mais bem sucedida dos últimos anos, caracterizada também como o padrão de rede mais utilizada para a transmissão de dados em redes corporativas. A simplicidade foi um dos principais motivadores do sucesso desta rede, que foi pioneira a sugerir a utilização de um meio compartilhado para acesso às redes. [OLIFER-08]

Vários outros fatores são associados ao êxito desta arquitetura de rede, [ROSS-06] enfatiza a velocidade da rede *Ethernet* de alta velocidade e complexidade das tecnologias existentes, como *token ring*, FDDI (*Fiber Distributed Data Interface*) e ATM (*Assynchronous Transfer Mode*), que além de complexas eram mais onerosas.

A flexibilidade da arquitetura *Ethernet* devido à ampla divulgação, possibilitou o conhecimento da estrutura dos protocolos que compõem a arquitetura, tornando-o um padrão acessível e passível de implementações, apresentando grande viabilidade quanto à adoção do padrão *Ethernet*. Conforme expõe [OLIFER-08], sobre sistemas abertos, que define tais sistemas como “qualquer sistema, seja computador isolado, uma rede de computadores, um sistema operacional, um aplicativo ou qualquer outro *hardware* ou *software*; constituído de acordo com as especificações abertas, onde tais especificações, após amplamente divulgadas e disponibilizadas, sendo submetidas a uma ampla e diversificada discussão.”

Dentro deste contexto, pode-se considerar a *Ethernet*, um sistema aberto bem sucedido, pela participação de instituições acadêmicas, institutos de pesquisa e corporações no desenvolvimento e aprimoramento do padrão, e pela aceitação dos usuários. Ainda, conforme expõe *Robert M. Metcalfe*, “a tecnologia *Ethernet*

transformou os PCs de unidades de processamento de dados em dispositivos de comunicação.” [HANCOCK-03]

A figura 27 ilustra o projeto original da *Ethernet* desenvolvido por *Robert Metcalfe* e *David Borges* na década de 1970. [ROSS-06]

Figura 27: Projeto original da *Ethernet*

Fonte: [ROSS-06].

O surgimento e desenvolvimento das redes *Ethernet* ocorreram nos laboratórios da *Xerox (Palo Alto Reserch Center)* e da *Intel*, em 1976. Tendo sido desenvolvida originalmente por *Robert Metcalfe* e *David Boggs*, o sistema desenvolvido então foi definido como *Ethernet*, em referência ao éter luminoso, através do qual os antigos diziam que a radiação eletromagnética se propagava. (Físico britânico *James Maxwell*, século XIX descobriu que a radiação eletromagnética podia ser descrita por uma equação de onda).

Em 1978, a *Digital Equipment Corporation (DEC)* e a *Intel Corporation* uniram-se à *Xerox* para definir o padrão *Ethernet DIX V1.0*, que operava a velocidade de 2,94 Mbps. Os procedimentos utilizados para a elaboração do padrão foram divulgados numa publicação conhecida por Livro Azul da *Ethernet*. [HANCOCK-03]

Como continuidade deste desenvolvimento, foi elaborado o padrão cooperativo *Ethernet Versão 2.0*, que contribuiu para formar a base para o padrão 802.3. Conforme expõe [HANCOCK-03], as duas especificações, *Ethernet V2.0* e IEEE 802.3 são similares, visto que o IEEE utilizou as detalhes tecnológicos da *Ethernet V2.0* como base para o padrão 802.3, porém, existem as diferenças entre os padrões, tornam-os incompatíveis entre si, tais como cabos, diferenças entre

funções de transceptores, formato de quadros. Utilizando um método de acesso para gerenciar demandas simultâneas, este padrão difundiu-se como a tecnologia mais utilizada em ambientes corporativos, evoluiu para um protocolo utilizado em mais de 85% das conexões instaladas. [ROSS-06]

A rede *Ethernet* foi oficialmente aceita como padrão em 1985, definido no IEEE 802.3 (*Institute of Electrical and Electronics Engineers*), que operava a taxa de 10 Mbps, atualmente as redes *Ethernet* operam a taxas de transmissão de até 10 Gbps. A normatização IEEE 802.3, definiu a camada física e de *software* para a rede local LAN. A tabela 04 abaixo apresenta os principais grupos para a normatização do padrão IEEE 802.

Tabela 04: Grupos de trabalhos – normatização padrão IEEE 802.

Padrão	Especificação
802.1	Avaliação e arquitetura de LANs
802.2	LLC (<i>Logical Link Control</i>) Controle de link lógico
802.3	<i>Ethernet</i>
802.4	<i>Token bus</i>
802.5	<i>Token Ring</i> (anel de símbolos - representa entrada da empresa IBM nas LANs)
802.6	<i>Fila Dual / Barramento Dual</i> (primeira rede metropolitana)
802.7	Grupo técnico consultivo para tecnologias de banda larga
802.8	Grupo técnico consultivo para tecnologias de fibra óptica
802.9	LANs isócronas, para aplicações de tempo real
802.10	LANs virtuais e segurança
802.11	LANs sem fio (<i>Wireless Networks</i>)
802.12	Prioridade de demanda (<i>AnyLAN da Hewlett Packard</i>)
802.13	Vago, número não utilizado.
802.14	Modens a cabo, acabou extinto (consórcio industrial conseguiu chegar primeiro)
802.15	Redes pessoais, <i>Bluetooth</i> .
802.16	Radio de banda larga
802.17	Anel de pacote elástico

Fonte: [TANENBAUM-03]

O padrão 802.11, que trata as redes *Wireless*, sob nome comercial de *WiFi*, oferece diversos produtos no mercado, dedicados à dispositivos móveis; o padrão 802.15 também engloba o *ZigBee*, destinado ao fornecimento de comunicação de dados para equipamentos domésticos de largura de banda baixa com baixas potências; o padrão 802.16, com denominação comercial de *WiMax*, foi ratificado em 2004/2005 como alternativa para os enlaces a cabo e DSL (*Digital Subscriber Line*) para aplicação em residências e escritórios. [COULOURIS-07]

Conforme expõe [OLIFER-08], atualmente a *Ethernet* engloba as tecnologias *Token Ring*, FDDI (*Fiber Distributed Data Interface*), IEEE 802.11, entre outras, onde todas estas tecnologias, apesar de terem características específicas, destinam-se a construção de redes LANs. Entretanto, na ocasião do desenvolvimento inicial das tecnologias de redes locais (LANs), a integração com redes de longa distância WAN não foi tratada como prioritária, desta forma, as redes LANs implementam somente as funções descritas para as duas camadas mais baixas do modelo OSI, ou seja, camada física e camada de enlace de dados; a figura 28 ilustra esta correspondência.

Figura 28: Correspondência entre protocolos modelo OSI e protocolos LAN

Fonte: [OLIFER-08]

Esta característica inicial ocorreu devido à funcionalidade destas duas camadas atenderem os requisitos para entrega dos quadros na topologia de padrões LAN. Contudo, posteriormente a *Ethernet* foi integrada a arquitetura TCP/IP (*Transmission Control Protocol/Internet Protocol*), caracterizada como rede WAN, que enfatiza a interligação de redes de tecnologias heterogêneas formando uma inter-rede.

Os *frames* definidos pela norma IEEE 802.3 e o padrão elaborado pela *Intel*, *Digital* e *Xerox* possuem diferenças, conforme ilustra a figura 29 a seguir.

Legenda:

G / I - Endereço de grupo ou individual

FCS - Frame check sequence

SFD - Start of Frame Delimiter

L / G - Bit de endereço Local ou Global

Figura 29: Formato *Frame Ethernet* DIX V1.0 e IEEE 802.3

Fonte: Adaptado de [TANENBAUM-03] e [PETERSON-04]

A constituição do quadro *Ethernet* (formato *frame* padrão *Ethernet Xerox, Intel* e *Digital*), conforme figura 29, mostra preâmbulo de 64 bits, permite que o equipamento receptor seja sincronizado com o sinal, sendo uma seqüência de 0 e 1 alternados. Os *hosts* de origem e destino são identificados por um endereço de 48 bits. O campo de tipo de pacote serve como uma chave de demultiplexação, identificando os possíveis protocolos de nível mais alto no qual este quadro deverá ser entregue. Cada quadro contém até 1500 *bytes* de dados. No mínimo, um quadro precisa conter pelo menos 46 bytes de dados, mesmo que isso signifique que o *host* deve completar o quadro antes de transmiti-lo. O motivo para esse tamanho mínimo do quadro é que ele precisa ter tamanho suficiente para detectar uma colisão. [PETERSON-04].

As divergências quanto ao padrão IEEE 802.3 e a proposta DIX elaborada pelo consórcio *Xerox, Intel* e *DEC* levaram ao surgimento de variantes do *frame Ethernet*, sendo: [OLIFER-08]

- Segunda variante - 802.3/LLC (802.3/802.2 ou *Novell* 802.2);
- Terceira variante - *Raw* 802.3/*Novell* 802.3;
- Quarta variante - *Ethernet* SNAP (SNAP – *Subnetwork Access Protocol*).

Os formatos das variantes do *frame Ethernet* são ilustrados na figura 30.

Quadro 802.3 / LLC

6	6	2	1	1	1(2)	46 - 1497 (1496)	4
D A	S A	L	DSAP	SSAP	Controle	Dados	FCS
Cabeçalho LLC							

Quadro Raw 802.3 / Novell 802.3

6	6	2	46 - 1500	4
D A	S A	L	Dados	FCS

Quadro Ethernet DIX (II)

6	6	2	46 - 1500	4
D A	S A	T	Dados	FCS

Quadro Ethernet SNAP

6	6	2	1	1	1	3	2	46 - 1492	4
D A	S A	L	DSAP	SSAP	Controle	OUI	T	Dados	FCS
AA					AA	3	0		
Cabeçalho LLC					Cabeçalho SNAP				

Figura 30: Variantes do *Frame Ethernet*

Fonte: [OLIFER-08]

A partir da década de 1980, o Comitê IEEE 802 padronizou as tecnologias de redes LAN, baseado nos padrões IEEE 802.x, conforme ilustra a figura 31 e tabela 04. A primeira versão do padrão IEEE 802.3, foi publicada em 1985, tendo como título formal “*IEEE 802.3 Carrier Sense Multiple Access with Collision Detection Access Method and Physical Layer Specifications.*”

Figura 31: Estrutura dos padrões IEEE 802.x

Fonte: [OLIFER-08]

Ainda, conforme [OLIFER-08], vários outros grupos participaram da padronização dos protocolos LAN, como o *ANSI* (*American National Standards Institute*), desenvolvedor do padrão *FDDI* (*Fiber Distributed Data Interface*), definindo em 1986 o padrão *ANSI X3T9.5*, utilizado para interconexão de sistemas de computadores em rede de topologia em anel de fibra, com taxas de transmissão de 100 Mbps. A relação de especificações que atendem ao IEEE 802.1 continua a ganhar novos padrões, recentemente, foram incorporados o 802.1Q que define modelos para LANs virtuais; o 802.1p, que determina os modelos de priorização de tráfego, oferecendo suporte para aplicações de QoS (*Quality of Service*).

A tabela 05 a seguir apresenta relação das principais especificações.

Tabela 05: Novas especificações IEEE 802.

Padrão	Função	Observação
IEEE 802.P	Priorização de mensagens	256 níveis de prioridade
IEEE 802.12d	Redundância de <i>Links</i>	Maior confiabilidade para a rede
IEEE 802.3x	<i>Full duplex</i>	Comunicação bidirecional simultânea
IEEE 802.3z	<i>Gigabit Ethernet</i>	Utilizado como backbone corporativo

Fonte: Adaptado de [TANENBAUM-03]

Conforme expõe [OLIFER-08], inicialmente, as redes *Ethernet* foram criadas com base em cabos coaxiais, posteriormente, com o desenvolvimento de outros meios físicos para transmissão; foram elaboradas especificações que atendem a estes meios de transmissão para a rede *Ethernet* 10 Mbps, em função da distância e meio físico. As principais implementações são ilustradas na tabela 06, sendo:

Tabela 06: Características de meio de transmissão IEEE 802.3 *Ethernet* 10 Mbps

Rede	Meio	Distância (m)
10Base5	Baseband coaxial grosso	500
10Base2	Baseband coaxial fino	185
10Base-T	Baseband par trançado	100
10Base-F	Baseband fibra óptica	Variável
10Broad 36	Broadband	3600

Fonte: Adaptado de [OLIFER-08]

O padrão *Ethernet* 10 Mbps atendeu aos principais requisitos para transmissão de dados por muitos anos, porém, a partir dos anos 90, com o aumento das taxas de velocidade dos barramentos internos dos computadores, que superaram 1000 Mbps, tornou-se necessário o surgimento de novas tecnologias *Ethernet* para atender a esta nova demanda, culminando com o surgimento da *Fast Ethernet*.

Em 1993, a organização FEA (*Fast Ethernet Alliance*), propôs a adoção do *Fast Ethernet* como padrão ao IEEE. A organização FEA contava com fabricantes como *Intel*, *SynOptics*, *3Com*, *Sun Microsystems*, entre outros. O padrão proposto foi definido como 802.3u, especificado por 100Base-T.

As especificações de meio para o padrão *Fast Ethernet* (100 Mbps) são ilustradas na tabela 07.

Tabela 07: Características de meio de transmissão IEEE 802.3u *Ethernet* 100 Mbps.

Rede	Meio	Distância (m)
100Base-TX	UTP EIA/TIA Cat. 5 / STP IBM tipo 1	100
100Base-T4	UTP EIA/TIA Cat. 5 / STP IBM tipo 1	100
100Base-FX	Fibra multimodo 62,5/125 μ	412 - 2000

Fonte: Adaptado de [HANCOCK-03]

O padrão IEEE 802.3u (*Fast Ethernet* 100Base-T) manteve o CSMA/CD para o acesso ao meio de transmissão, mantendo portando compatibilidade com a *Ethernet Convencional* 10 Mbps, porém, a coalizão formada pela *HP* e *AT&T* defendiam novos métodos de acesso ao meio, definindo o método de prioridade por demanda, similar padrão *token ring* (IEEE 802.5).

O padrão proposto pela *HP* e *AT&T*, denominado 100VG-AnyLAN foi definido em 1995 como IEEE 802.12, entretanto apresenta incompatibilidade com padrão *Fast Ethernet* IEEE 802.3u. O protocolo proposto pelo padrão 100VG-AnyLAN define prioridade por demanda, especifica como os *hubs* disponibilizam suas portas para identificar os nós com dados para transmitir e a ordem das transmissões. [HANCOCK-03]

A tabela 08 apresenta resumo comparativo entre padrões IEEE 802.3u (*Fast Ethernet*) e IEEE 802.12 (100VG-AnyLAN).

Tabela 08: Comparação entre IEEE 802.3u e IEEE 802.12

	IEEE 802.3u	IEEE 802.12
Meio		
UTP categoria 3	4 pares (100 m) - 100Base T4	4 pares (100 m)
UTP categoria 4	4 pares (100 m) - 100Base T4	4 pares (100 m)
UTP categoria 5	2 pares (100 m) - 100Base TX	2 pares (não disponível)
	4 pares (100 m) - 100Base T4	4 pares (200 m)
UTP de 25 pares	Sem suporte	Suporte
Meio		
STP IBM Tipo 1	Sim (100 m) - 10Base - T4 / TX	Sim (100 m)
Fibra óptica (62,5/125)	412 m duplex parcial - 100Base-FX	Sim (200 m)
	2 km duplex completo - 100Base-FX	
Topologia		
Diâmetro da rede	Varia de 200 m a 320 m dependendo do tipo de cabo e das repetidoras	8 km
Repetidoras em cascata	Dois níveis	Cinco níveis
Subcamada MAC		
Acesso ao meio	CSMA/CD	Prioridade de demanda
Quadros IEEE 802.3	Sim	Sim
Quadros IEEE 802.5	Não	Sim
Suporte a aplicações		
Dados sensíveis à passagem de tempo	Não	Sim
Desempenho		
Taxa de transferência de 100 m	80%	95%
Taxa de transferência de 2500 m	Sem suporte	80%

Fonte: Adaptado de [HANCOCK-03]

O padrão IEEE 802.12, possui características determinísticas, livre de colisões, utiliza com eficiência a largura de banda com estabilidade, entretanto, apesar destas características, não foi enquadrado como uma rede *Ethernet*, onde foi definida outra designação para enquadrá-lo (802.12) por não utilizar o meio de acesso CSMA/CD (*Carrier Sense Multiple Access/Collision Detection*). Conforme expõe [HANCOCK-03]; o processo de consulta e determinação da ordem de transmissão dos dados, denominado de arbitragem cíclica priorizada, é o coração do protocolo de prioridade por demanda, que desempenha a função de controlar e tomar decisões quanto ao acesso das estações ao meio compartilhado. A partir de 1996, o IEEE elaborou

grupo de estudo para elevar as taxas de transmissão do *Fast Ethernet*, o Grupo de Estudos de Alta Velocidade (*HSSG – Higher Speed Study Group*) conduziu estudos com objetivo de elevar as taxas de transmissão a 1000 Mbps, definindo a *Gigabit Ethernet Alliance* (GEA), que rapidamente atingiu mais de 100 membros após seis meses de criação. [HANCOCK-03]

Como resultados destes estudos foram elaborados dois padrões para a *Gigabit Ethernet*, o IEEE 802.3z, tendo como meio físico a fibra óptica e o IEEE 802.3ab, que focou o cobre como meio para transmissão. Em 1998, o padrão IEEE 802.3z foi aprovado e em 1999 o IEEE 802.3ab foi confirmado como padrão. A condução de grupos de desenvolvimentos distintos ocorreu para atender tecnologias diferentes, conforme ilustra a figura 32 abaixo.

Figura 32: IEEE 802.3z e IEEE 802.3ab

Fonte: Adaptado de [OLIFER-08]

A tabela 09 apresenta resumo comparativo entre os três grupos de redes *Ethernet*: a Convencional (10 Mbps), *Fast Ethernet* (100 Mbps) e *Gigabit Ethernet* (1000 Mbps).

Tabela 09: Resumo comparativo: *Ethernet* convencional, *Fast Ethernet* e *Gigabit Ethernet*.

Comprimento máximo por segmento			
	<i>Ethernet</i> Convencional	<i>Fast Ethernet</i>	<i>Gigabit Ethernet</i>
Taxa de dados	10 Mbps	100 Mbps	1000 Mbps
UTP categoria 5	100 m	100 m	100 m
STP IBM tipo 1	500 m	100 m	25 m
Fibra multimodo	2 Km	412 m - duplex parcial 2 Km - duplex completo	260 - 550 m
Fibra modo único	25 Km	20 Km	3 Km

Fonte: Adaptado de [HANCOCK-03]

O padrão *Ethernet* Convencional requer um tempo máximo de 1,2 milissegundos para a transmissão de um *frame* de 1518 *bytes*. Atualmente, esse tempo pode ser reduzido a 12 microssegundos utilizando *Gigabit Ethernet*. No entanto, as possibilidades para *Ethernet* a 10Gbit/s já estão sob consideração, onde um pacote pode ser transmitido dentro de 1,2 microssegundos. A *Ethernet Gigabit* admite utilização de cabos de cobre e de fibra óptica, para as derivações do padrão, conforme apresenta a tabela 10.

Tabela 10: Meio físico para *Ethernet Gigabit*

	1000Base-SX	1000Base-LX	1000Base-CX	1000Base-T
Meio	Fibra de multimodo (50 / 62,5 microns)	Modo único (10 microns) ou multimodo (50 / 62,5 microns)	Par trançado blindado	UTP padrão da categoria 5
Tamanho máximo de segmento	550 metros	500 metros	25 metros	100 metros

Fonte: Adaptado de [TANENBAUM-03]

O conceito principal dos desenvolvedores da *Gigabit Ethernet* foi manter as características essenciais da *Ethernet Convencional*, portanto a *Gigabit Ethernet* não oferece suporte direto a QoS (*Quality of Service*) para isto, o IEEE inseriu dois protocolos para a camada de enlace, o IEEE 802.3p e o IEEE 802.3q. O IEEE 802.3p define um esquema de prioridade de 03 bits com oito níveis de prioridade, que atende aos padrões 802.4, 802.5 802.6 e 802.12. Para definição de prioridade no padrão IEEE 802.3 (*Ethernet*) o IEEE 802.q fornece priorização de dados que pode ser configurado e ativado pelo usuário.

Conforme expõe [HANCOCK-03], outro protocolo relacionado com esse conceito é o controle de fluxo IEEE 802.3x duplex completo, que possibilita que portas de chave duplex completo enviem comandos de controle de fluxo para estações de trabalho a elas conectadas. Estendendo esse conceito às camadas superiores, o IETF (*Internet Engineering Task Force*) desenvolveu o RSVP (*Resource Reservation Protocol*) que opera na camada 03 e permite que nos terminais reservem uma quantidade específica de largura de banda através de uma rede IP para uma transmissão particular.

A figura 33 ilustra os frames IEEE 802.3 e IEEE 802.3q.

Figura 33: *Frames* IEEE 802.3 e 802.3q

Fonte: Adaptado de [HANCOCK-03]

O padrão *Ethernet* é um meio compartilhado, onde a ocorrência de disputa para transmissão de dados é inerente; a definição de priorização de tráfego, remete a Qualidade de Serviços (QoS) que implementada de forma correta, assegura que todas as transmissões sejam entregues. Conforme expõe [DONAHUE-08], a implantação da QoS ocorre inicialmente por meio da marcação de pacotes, posteriormente o controle e finalmente a programação destes pacotes. A marcação refere-se a decidir que prioridade deve possuir; o controle esta relacionado as medidas tomadas pelo roteador e a programação se refere a interface que esta atendendo os pacotes, ou seja, na ordem determinada pela prioridade definida.

A *Gigabit Ethernet* possibilita redes comutadas, roteadas e compartilhadas. Todas as tecnologias de interconexão atuais, como *IP switching* e *Layer 3 switching*, são completamente compatíveis com *Gigabit Ethernet* e *Fast Ethernet*. Portanto, a escolha de *Gigabit Ethernet* como uma rede de alta velocidade, não restringe a escolha da tecnologia de interconexão ou a topologia da rede.

A *Ethernet* utilizada no ambiente Industrial executa a transmissão de dados críticos e dados rotineiros, distinguindo e priorizando-os conforme tipo de serviço, assegurando a eficácia da transmissão. O *switch* industrial utilizado na *Ethernet* assegura confiabilidade e determinismo para processos produtivos, evitando que CLPs e outros dispositivos processem informações desnecessárias, priorizando o tráfego crítico em tempo real comparado às mensagens menos importantes, assegurando o controle em tempo real do processo produtivo e postergando as transmissões configuradas como secundárias.

4.4 TCP/IP

A DARPA (*Defense Advanced Research Projects Agency*) iniciou os trabalhos em direção a uma tecnologia de interconexão de redes em meados da década de 1970, com a arquitetura e os protocolos tomando forma entre 1977 e 1979, entretanto, os primeiros passos para a definição de uma rede de comunicação iniciaram-se na década de 1960. Estudos conduzidos no MIT (*Massachusetts Institute of Technology*), no *Rand Insitute* e no *National Physical Laboratory* forneceram os alicerces para a atual Internet. [ROSS-06]

Na década de 1970, a DARPA era reconhecida como a principal agência financiadora de pesquisa de redes de comutação de pacotes com a ARPANET, que inicialmente era uma rede fechada.

A partir de meados da mesma década iniciaram novas redes de comutação por pacotes: *ALOHAnet*, *Telenet*, *Cyclades*, *SNA*. [ROSS-06]

Conforme expõe [COMER-06], a Internet global começou por volta de 1980, quando a DARPA iniciou processo de conversão de computadores conectados às redes de pesquisa aos recém-desenvolvidos protocolos TCP/IP; a transição para a tecnologia de Internet foi concluída em 1983, momento em que a Secretaria de Defesa dos Estados Unidos determinou que todos os computadores de redes de

longa distância utilizassem os protocolos TCP/IP; simultaneamente, a DCA (*Defense Communication Agency*) dividiu a ARPANET em duas redes distintas: uma destinada à pesquisa e outra com a finalidade de comunicação militar.

A DARPA, por meio de uma política extensiva de divulgação, disponibilizou os protocolos TCP/IP para serem implementados em universidades dos Estados Unidos, que rapidamente atingiram cerca de 90% dos departamentos de ciências da computação das universidades. Após sete anos de seu início, a Internet cresceu por milhares de redes localizadas nos Estados Unidos e Europa; em 2005, a Internet global atingiu cerca de 300 milhões de computadores em 209 países. [COMER-06]

O surgimento da Internet implementada pelos protocolos TCP/IP ocorreu de forma paralela ao desenvolvimento das redes de comunicação, e seu desenvolvimento não seguiu as padronizações como o Modelo de Referência OSI ou IEEE.

O modelo TCP/IP, foi concebido para situações extremas e para ter capacidade de enviar mensagens, mesmo que houvesse interrupções no trajeto, por caminhos ou rotas alternativas, o que permite interligar redes de tecnologias heterogêneas, permitindo a formação de uma inter-rede.

Os dois principais protocolos que implementam este modelo acabaram por fornecer o nome ao modelo. Esta arquitetura baseia-se em um serviço de transporte orientado à conexão, serviço que é fornecido pelo TCP (*Transmission Control Protocol*) orientado à conexão, e, pelo IP (*Internet Protocol*) não orientado à conexão.

A arquitetura do modelo TCP/IP é organizada em quatro camadas: *host/rede*, *inter-redes*, *transporte* e *aplicação*.

A figura 34 a seguir ilustra as camadas e tráfego de dados entre as mesmas.

Figura 34: Camadas da arquitetura TCP/IP

Fonte: Adaptado de [SOARES-99]

A arquitetura TCP/IP enfatiza a interligação de redes de tecnologias diferentes, o conceito desta necessidade está associada ao fato de que não há nenhuma tecnologia que atenda às necessidades de todos os usuários.

Para que sejam conectadas duas ou mais redes faz-se necessário a utilização de Internet *gateway* ou *Internet router*, o usuário enxerga a inter-rede como uma rede virtual única, onde todas as máquinas estão conectadas, não importando a forma física de como isto ocorre.

Como comparação entre o modelo TCP/IP e o Modelo OSI, o ponto de fundamental relevância que pode ser estabelecido entre os dois é o motivo de suas concepções: a arquitetura TCP/IP foi elaborada para interligar redes com tecnologias distintas, a simplicidade quanto ao número de camadas também tornou o TCP/IP mais abrangente; o Modelo de Referência OSI teve como finalidade a padronização para redes, onde inicialmente houve o desenvolvimento das camadas e posteriormente a implementação dos protocolos da arquitetura, apresenta complexidade na implementação de protocolos e definição de serviços inter-relacionados [TANENBAUM-03].

A figura 35 ilustra as camadas do modelo OSI e TCP/IP.

Legenda:

TELNET - Prove serviço de terminal virtual

FTP - File Transfer Protocol

SMTP - Simple Mail Transfer Protocol

DNS - Domain Name System

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

IP - Internet Protocol

Figura 35: Camadas do Modelo OSI e TCP/IP

Fonte: Adaptado de [TANENBAUM-03]

Esta solução simples e prática mostrou-se muito eficiente para a interconexão de redes de sistemas abertos, oferecendo uma opção de solução não proprietária. O modelo TCP/IP passou a ser visto como padrão de *facto*. Outro aspecto relevante a Arquitetura TCP/IP está associado ao número de camadas; na subcamada de rede, somente a utilização do protocolo IP, cujo serviço e datagrama não confiável, trouxeram inflexibilidade na arquitetura neste nível, sendo um dos motivos que trouxeram êxito à arquitetura TCP/IP. [SOARES-99]

4.4.1 Camadas TCP/IP

A camada *Host/Rede* é um grande vácuo, não sendo especificamente determinado o que ocorre nesta camada que possibilita conectividade com diversos tipos de redes.

A camada Inter-Rede, responsável pela integração da rede, oferece transferência de dados, recebe pedidos de transporte para transmitir os pacotes que

são encapsulados em um datagrama IP (que é definido nesta camada) que juntamente com algoritmos de roteamento determinam como as mensagens serão entregues. A figura 36 abaixo ilustra a posição da camada de inter-rede e serviços oferecidos.

Figura 36: Camada de Rede - Serviços

Fonte: [FOROUZAN-06]

A camada de Transporte tem como principal responsabilidade possibilitar que as entidades (*hosts*) de origem e destino estabeleçam comunicação. Nesta camada, estão implementados dois protocolos que atuam para viabilizar a fluidez da comunicação: o protocolo TCP e o protocolo UDP. O TCP oferece um serviço de circuito virtual, orientado à conexão confiável, assegura a entrega sem erros de *bytes* dentro da inter-rede; trata o fluxo de dados, impedindo que o receptor seja sobrecarregado com um volume maior de informações que possa manipular. O UDP é um protocolo sem conexão e sem confiabilidade, que é utilizado em situações que não requerem controle do fluxo de informações. É também utilizado em circunstâncias em que a entrega imediata de informações é mais importante que a precisão.

A figura 37 a seguir ilustra os serviços executados na camada de transporte.

Figura 37: Serviços executados na camada de transporte

Fonte: Adaptado de [FOROUZAN-06]

Na camada de Aplicação estão contidos os protocolos de nível mais elevado da arquitetura TCP/IP, onde são implementados as aplicações e processos que utilizam a rede; diversos protocolos foram implementados nesta camada para atender várias necessidades, TELNET, FTP (*File Transfer Protocol*), SMTP (*Simple Mail Transfer Protocol*), DNS (*Domain Name System*), HTTP (*Hypertext Transfer Protocol*), entre outros.

4.4.2 Protocolo TCP

Conforme expõe [HANCOCK-03], o TCP é um protocolo duplex completo orientado por conexão, que desempenha funções como: transmissão confiável de dados, detecção e correção de erros, garantia de transmissão de dados na ordem correta, retransmissão de dados não recebidos, garantia de não duplicação de dados entre os nós origem e destino.

O formato e conteúdo do cabeçalho TCP são apresentados na figura 38 a seguir.

Porta da fonte #					Porta do destino #				
Número de sequência									
Número de reconhecimento									
Comprimento do cabeçalho	NU	URG	ACK	PSH	RST	SYN	FIN	Janela de recepção	
Valor de verificação da Internet					Ponteiro para dados urgentes				
Opções									
Dados									

Legenda:

RST / SYN / FIN - Estabele e encerra conexão

PSH - Informações para destinatário

ACK - Indicação de valor carregado em campo de reconhecimento

URG - Mostra existência de dados

NU - Não utilizado

Figura 38: Estrutura do segmento TCP

Fonte: Adaptado de [ROSS-06]

O TCP/IP utiliza na comunicação entre entidades de rede o processo conhecido por *three-way handshake*¹, que envolve uma estação inicial (cliente) requisitando que um *link* seja estabelecido com o servidor para iniciar a comunicação. O TCP é caracterizado como um protocolo tipo PAR (*positive acknowledgment with retransmission*) e provê também controle de fluxo fim-a-fim, onde utiliza *buffers* para controle de dados: transmitidos, recebidos e não confirmados.

Em comparação as redes locais, as redes destinadas à grandes distâncias possuem latência elevada, a janela de 16 *bits* que inicialmente implementava o TCP restringe o fluxo de dados entre nós. Esta restrição foi corrigida com a introdução da opção escala de janela definida nas RFCs 1072 e 1323.

Outra modificação relevante ocorrida TCP foi a função *time-stamp* (tempo de espera de confirmação). O PAR do TCP requer que o emissor retransmita todos os segmentos de dados não confirmados, se as confirmações não chegarem após tempo definido. A especificação de *time-stamp* apropriado consta na RFC 1323. [HANCOK-03]

¹ Comunicação em três fases: 1) o cliente inicia uma solicitação de conexão para o servidor; 2) o servidor confirma o pedido; 3) o cliente confirma a confirmação do servidor. [HANCOCK-03]

O número de seqüência para um segmento do TCP também sofreu modificações por meio da RFC 2018, com a inclusão de confirmação seletiva, onde se houver um intervalo entre grupos de segmentos recebidos, o receptor envia confirmações de dados recebidos especificando os dados recebidos fora de ordem.

4.4.3 Protocolo UDP

O protocolo UDP (*User Datagram Protocol*) é um protocolo não orientado à conexão que provê um serviço de datagrama não confiável, esta definido na RFC 768. [ROSS-06].

A figura 39 abaixo ilustra estrutura do segmento UDP.

Porta da fonte #	Porta do destino #
Comprimento	Soma de verificação
Dados da aplicação (mensagem)	

Figura 39: Estrutura do segmento UDP

Fonte: Adaptado de [ROSS-06]

O protocolo UDP apresenta estrutura simples, cujo propósito é a rapidez na entrega de segmentos, entretanto, não executa detecção e / ou correção de erros, não retransmite os dados que eventualmente não foram entregues e também não executa controle de erros ou fluxo de dados.

Estas características representam rapidez na entrega de mensagens, porém atribui aos níveis de aplicação a tarefa pelo controle de erros e controle de fluxo de pacotes.

Conforme expõe [PETERSON-04], embora o UDP não implemente controle de fluxo confiável, o mesmo assegura a exatidão da mensagem pela utilização de uma soma de verificação, conforme figura 39. A soma de verificação, na Internet baseada em IPv4 é opcional, porém, passará a ser obrigatório na Internet IPv6.

4.4.4 Protocolo IP

O protocolo IP está definido na RFC 791; foi desenvolvido para possibilitar a interconexão de redes baseados na tecnologia de comutação de pacotes (datagramas IP). Na arquitetura TCP/IP o ambiente inter-redes (camada de atuação do protocolo IP) consiste em *hosts* conectados a redes que por sua vez são interligados por meio de *gateways* (roteadores).

O IP é um protocolo não orientado à conexão, sua função possibilitar a transferência de datagramas do *host* origem para o *host* destino. O protocolo também provê serviços de fragmentação e remontagem de datagramas longos.

A figura 40 abaixo apresenta datagrama IP.

VERS	HLEN	TIPO SERVIÇO	TAMANHO TOTAL	
IDENTIFICAÇÃO			FLAGS	OFFSET DO FRAGMENTO
TEMPO DE VIDA		PROTOCOLO	CHECKSUM DO CABEÇALHO	
ENDEREÇO IP ORIGEM				
ENDEREÇO IP DESTINO				
OPÇÕES IP				PREENCHIMENTO
DADOS				
DADOS				

Legenda:

VERS - Versão do protocolo IP

HLEN - Tamanho do cabeçalho

Figura 40: Formato de datagrama Internet TCP/IP

Fonte: [COMER-06]

Algumas das principais características do protocolo IPv4 são: [SOARES-99]

- Serviço de datagrama não confiável;
- Endereçamento hierárquico;
- Facilidade de fragmentação e remontagem de pacotes;
- Identificação da importância do datagrama e do nível de confiabilidade exigido;
- Identificação da urgência de entrega e da ocorrência futura ou não de pacotes na mesma direção (pré-alocação, controle de congestionamento);

- Campo especial indicando qual o protocolo de transporte a ser utilizado no nível superior;
- Roteamento adaptativo distribuído nos *gateways*;
- Descarte e controle de tempo de vida dos pacotes inter-redes no *gateway*.

Os endereços IPv4 são compostos por 32 bits. Os endereços IP podem ser utilizados para redes e para *host* individual. O protocolo IP utiliza três classes de endereços para redes distintas: classe A, classe B e classe C.

Conforme expõe [HANCOCK-03], o controle de endereços IP foi elaborado inicialmente pelo IANA (*Internet Assigned Numbers Authority*), que na ocasião possuía autoridade sobre os números utilizados na Internet. O IANA integrou o Instituto de Ciência da Informação da *University of Southern California*. A partir de 1999, foi constituída nos Estados Unidos a organização ICANN (*Internet Corporation for Assigned Names and Numbers*) para assumir e conduzir as atribuições do IANA.

As atribuições de endereço IP são administradas de forma distribuída e hierárquica por meio de um sistema de Registros da Internet (IR – *Internet Registry*), o qual possui uma estrutura com várias organizações em diversos países.

Os endereços IP são utilizados para identificação de *hosts* na Internet. Atualmente predominante na Internet o IPv4. Conforme expõe [HANCOCK-03], o processo de resolução de nomes mais eficiente é desempenhado por servidores de nomes de domínios.

As limitações impostas ao IPv4 devido ao crescimento da Internet nas últimas décadas culminaram no desenvolvimento do IPv6 descrito na RFC 1752, que foi projetado para substituir o IPv4, solucionar os problemas de endereçamento de IPv4 e prover novas funções para suprir as demandas com as diversas aplicações de endereço IP. Com a utilização de endereços de 128 bits, o IPv6 disponibiliza vários tipos de endereços: *unicast* (uma origem e um destino), *anycast* (define um único computador) e *multicast* (uma origem e muitos destinos); provê suporte à roteamento prioritário, oferece configuração e reconfiguração automáticas de endereços e mecanismos relacionados à segurança como: autenticação, privacidade e confidencialidade de dados via criptografia. [FOROUZAN-06]

A figura 41 a seguir ilustra formato dos cabeçalhos IPv4 e IPv6.

4	4	8	16	16	3	13	8	8	16	32	32	Variável	
V	HL	ST	TL	ID	F	FO	TTL	P	HC	S A	DA	OPT	PAD

Formato cabeçalho IPv4

4	4	24	16	8	8	128	128
V	P	FL	PL	NH	HL	S A	DA

Formato cabeçalho IPv6

Legenda:

V - Versão do protocolo

HL - Comprimento do cabeçalho IPv4 / Especificação pacote ativo (tempo) IPv6

ST - Especificação de pacote para roteamento

TL - Comprimento total do pacote

FO - Informação para remontagem de pacotes

TTL - Especifica tempo que pacote permanece ativo

P - Especifica criação de dados do usuário IPv4 / prioridade de pacotes IPv6

HC - Checagem de soma do cabeçalho

S A - Endereço IP de origem

DA - Endereço IP de destino

OPT - Especifica opções de controle

PAD - Auxílio especificação tamanho cabeçalho

FL - Especifica rotulo de controle para tratamento especial

PL - Especifica comprimento de dados do usuário no cabeçalho

NH - Especifica tipo de cabeçalho IPv6

Figura 41: Formatos dos cabeçalhos IPv4 e IPv6

Fonte: Adaptado de [HANCOCK-03] e [ROSS-06]

Segundo [DOMINGUES-02], que aponta a consolidação de redes como solução para integração entre arquiteturas distintas, como SNA e TCP/IP, onde cita: “todos os recursos e técnicas tendentes à consolidação SNA/IP não são excludentes, o que significa dizer; que não há um projeto padrão de integração apontando para o uso de uma técnica ou outra. Todas as formas podem ser utilizadas.”

5. AUTOMAÇÃO INDUSTRIAL

A Automação Industrial pode ser definida como um conjunto de técnicas e de sistemas de produção, principalmente fabris, baseado em máquinas e dispositivos (sensores, atuadores, transdutores, etc.) que atualmente interagem de forma integrada com capacidade de executar tarefas previamente definidas pelo homem e de controlar seqüências de operações sem a intervenção humana.

A Automação Industrial tem como objetivo a produtividade, qualidade e segurança na execução de processos.

5.1 Evolução do Controle Industrial

A Revolução Industrial ocorrida no século XIX alavancou os meios de produção da sociedade moderna, transformando os processos industriais e formas de produção em sistemas complexos, capazes de atuar sobre diversas variáveis, a figura 42 abaixo ilustra a atual classificação de sistemas de automação.

Figura 42: Classificação geral dos sistemas de automação

Fonte: [MORAES-07]

Conforme expõe [PEREIRA-03], “os primeiros passos para a automação industrial e a formação das atuais redes industriais, tiveram início durante a Revolução Industrial, no final do século XIX.”

À medida que a produtividade de máquinas e equipamentos crescia com o aumento do consumo das sociedades, tornou-se necessário o conseqüente surgimento de sistemas que possibilitassem aumento de produção, com tarefas repetitivas que se tornaram cada vez mais produtivas.

Em 1568, *Jacques Besson*, engenheiro francês desenvolveu o primeiro torno de abrir roscas que se tem notícia; em 1770, surgiram os mecanismos automáticos de produção em serie que revolucionaram a indústria, *James Watt* desenvolveu os primeiros reguladores mecânicos em 1788. Desde então o desenvolvimento e aperfeiçoamento de novos equipamentos e sistemas têm sido contínuo, passando por sistemas cada vez mais confiáveis, como o controle numérico, que começou a implementar as máquinas de bens de consumo da década de 1950, chegando aos robôs das atuais linhas de produção, que utilizavam chaves fim-de-curso e cames, que foram introduzidos comercialmente em 1959 pela *Planet Corporation*. [SILVEIRA-98]

A tecnologia com robótica baseada no conceito de automação programável passou a ser incorporada aos meios e processos de produção. Estas tecnologias de processo de produção passaram a ser utilizadas pelas indústrias automobilísticas a partir da década de 1960, quando a *Ford Motor* implementa seu processo produtivo com robô *Unimate*.

O conceito de automação baseada em controladores programáveis foi um avanço em relação ao controle numérico. Esta tecnologia passou a ser definitivamente incorporada aos processos industriais, substituindo quadros de comando a reles. [SILVEIRA-98]

No final dos anos 60, com o desenvolvimento da eletrônica, os circuitos integrados possibilitaram o surgimento de sistemas microprocessados que foram utilizados para controle de processos industriais. Em 1969, baseados numa especificação da *General Motors*, surgiram os primeiros Controladores Lógicos Programáveis (CLP).

O CLP foi idealizado, entre outros requisitos, para atender a necessidade de alteração de linhas de produção e industrialização sem que fossem realizadas grandes modificações mecânicas e elétricas em instalações, não havendo necessidade de impactos prolongados e paralisações de produção e sem grandes modificações no *layout* das fábricas. O CLP surgiu originalmente para atender a demanda da indústria automobilística, especificamente na *Hydronic Division da General Motors*, sob o comando do engenheiro *Richard Morley*. O objetivo inicial do CLP foi de reunir em um único equipamento as seguintes características funcionais: [MORAES-07]

- Facilidade de programação;
- Facilidade de intervenções para manutenção;
- Confiabilidade;
- Robustez, devido aos ambientes industriais;
- Possibilidade de redução de painéis e componentes elétricos;
- Possibilidade de envio de dados para processamento e supervisão centralizada;
- Possibilidade de expansão modular.

Ao longo do surgimento e desenvolvimento, os CLPs puderam ser classificados em função do sistema de programação utilizado e posteriormente pela interligação entre os dispositivos em redes de comunicação. Os equipamentos de 1ª geração foram caracterizados pela programação, sendo que nesta geração de dispositivos estava intimamente ligada à arquitetura do *hardware* do equipamento; nesta fase a linguagem utilizada era o *Assembly*, que variava de acordo com o processador utilizado na arquitetura do *hardware* do CLP, havendo a necessidade de conhecimento do circuito eletrônico do equipamento para a tarefa da programação do CLP; nesta geração, os dispositivos eram geralmente programados em fábrica, por meio da implementação do programa diretamente no equipamento. Com a evolução das linguagens de programação, surge então a 2ª geração, onde a linguagem de programação estava menos dependente do *hardware*, devido à inclusão de programas que realizavam a compilação das instruções de programa. Atualmente, devido aos modernos sistemas operacionais e às linguagens de

programação de alto nível, as metodologias de programação são desenvolvidas de forma independente do *hardware*.

Os controladores lógicos de 3ª geração passaram a dispor de dispositivos externos para comunicação portátil, que possibilitavam a programação via teclados ou por consoles.

A popularização dos microcomputadores ocorrida com o surgimento do PC IBM (*Personal Computer*) fez com que surgisse a 4ª geração, onde os CLPs dispunham de porta serial de entrada, que com auxílio dos microcomputadores viabilizou a programação, que passou a ser realizado no próprio equipamento. Nesta fase iniciou-se a utilização de computadores nos processos de automação industrial.

O desenvolvimento acelerado das comunicações industriais aliado à necessidade de padronização dos protocolos utilizados pelos dispositivos de automação caracterizou a 5ª geração, que passaram a dispor de redes de comunicação industrial que possibilitaram a interligação de dispositivos, compondo arquiteturas de automação.

Necessidades provenientes dos processos econômicos de globalização ocorridos na década de 1990 fizeram com as corporações buscassem formas mais flexíveis de integração de processos produtivos.

Com a consolidação da tecnologia de controladores lógicos programáveis, dos sistemas de comunicação e transmissão de dados, a busca pelo desenvolvimento de novos recursos a serem implementados nos sistemas de controle tem sido constante com o objetivo de aumentar a eficiência no monitoramento e controle de processos. Atualmente as principais características atualmente implementadas nos modernos controladores lógicos programáveis são: [MORAES-07]

- Linguagem de programação de alto nível;
- Simplificação de quadros elétricos;
- Confiabilidade operacional;
- Possibilidade de implementação de funções matemáticas avançadas;
- Possibilidade de interconexão por meio de redes de comunicação;

- Possibilidade de interligação a sistemas de supervisão e gerenciamento centralizado.

Tais características tornaram os sistemas de automação confiáveis e produtivos, sendo utilizados para controle em diversos segmentos, principalmente na indústria.

Conforme expõe [PRUDENTE-07], “o mercado de CLP cresceu de um volume de cerca de 120 milhões de dólares em 1978 para cerca de 100 bilhões de dólares nos primeiros anos da década de 1990 e continua em crescimento.”

5.2 Arquitetura de Automação Industrial

A arquitetura de automação industrial pode ser descrita como qualquer sistema, apoiado em tecnologia computacional, que substitua o trabalho humano, e que vise soluções rápidas e econômicas para atingir os complexos objetivos das industriais e dos serviços de automação. [MORAES-07]

Este conceito aborda de forma abrangente todos os sistemas que podem ser otimizados por meio de processos automatizados, podendo ser aplicado aos diversos sistemas e setores que utilizam automação, tais como industrial, bancário, comercial, etc., visando à produtividade, confiabilidade e eficiência de processos.

Conforme expõe [STEMMER-01], “verifica-se tendência de descentralização da capacidade decisória dos componentes do sistema de automação industrial, aproximando-os do processo, sem, entretanto perder as vantagens da supervisão e controle centralizados.”

Em automação industrial, os atuais controladores programáveis disponíveis, permitem o controle lógico e dinâmico de variáveis do processo com grandes vantagens e relativa simplicidade de programação e reprogramação, devido às linguagens utilizadas atualmente, conforme Norma IEC 61131-3, baseados em linguagens gráficas e textuais.

A figura 43 a seguir ilustra estas linguagens.

Figura 43: Linguagens de programação IEC 61131-3

Fonte: [PRUDENTE-07]

Nos atuais processos de automação são utilizados computadores de processos, que permitem a coleta e processamento de informações dos sistemas controlados, que possibilitam implementação de modelos matemáticos de engenharia de controle, simulando desempenho do sistema, regras de segurança, operações em tempo real, além de facilitar a interface com sistemas de supervisão e gerenciamento, conforme expõe [MORAES-07].

A utilização dos atuais sistemas automatizados implica necessariamente também na aplicação de sistemas interligados, monitorados e que possam ser supervisionados. As redes de comunicação surgem como forma viável de assegurar o acompanhamento dos processos. Estas redes de comunicação podem ser implementadas por sistemas supervisórios, sistemas de gerenciamento de falhas, sistemas de interface homem-máquina; com o objetivo de facilitar a interpretação de dados, possibilitando integração entre equipamentos e usuários. Esta integração por meio de redes assegura o monitoramento do desempenho dos sistemas produtivos e equipamentos utilizados para a automação destes sistemas.

Conforme expõe [MORAES-07], os sistemas de automação que envolvem a informatização apresentam vantagens significativas que são fundamentais para a competitividade das corporações, como a possibilidade de expansão e comunicação, com utilização de recursos de fácil acesso, sem os quais, as plantas

industriais podem apresentar uma grande quantidade de alarmes, por exemplo, inviabilizando a atuação do operador de tais sistemas e equipamentos.

No âmbito industrial, a automação atende às necessidades como maiores níveis de qualidade de conformação e de flexibilidade aos processos. Esta viabilização na utilização da automação pode ser mensurada por menores custos nos processos produtivos, redução ou até mesmo extinção de retrabalhos ocasionados por desconformidades no processo.

O planejamento adequado dos meios de produção assegura redução de perdas de matérias primas, redução do custo de capital pela gestão eficiente de estoques e recursos humanos; o maior controle de informações relativas ao processo e servirá de subsídio para decisões de planejamento e controle dos processos de produção.

Uma gestão corporativa eficiente compreende o gerenciamento de todas as etapas do processo produtivo. Devido ao grau de desenvolvimento das redes industriais e das redes corporativas de dados esta tarefa pode ser atingida, possibilitando o acompanhamento eficiente da planta, isto atualmente é possível devido aos avanços ocorridos nas áreas de tecnologia de automação e também nos segmentos voltados à tecnologia da informação.

A automação industrial contempla a realização de várias tarefas e funções que tem suas atribuições definidas em níveis de interligação e atribuição de forma hierarquizada, onde estas tarefas podem ser definidas em um organograma, definido como Pirâmide de Automação [MORAES-07], com diferentes níveis de automação encontrados numa planta industrial, conforme figura 44 a seguir:

Legenda:

SNA - Systems Network Architecture

TCP/IP - Transmission Control Protocol/Internet Protocol

MAP - Manufacturing Automation Protocol

CAN - Controller Area Network

CLP - Controlador Lógico Programável

PC - Personal Computer

CNC - Controle Numérico por Computador

SDCD - Sistemas Digitais de Controle Distribuído

Figura: 44 – Estrutura da Pirâmide de Automação

Fonte: Adaptado de [MORAES-07] e [STEMER-01]

A utilização de sistemas supervisórios aplicados aos controladores lógicos programáveis permitiu que fossem integrados os sistemas produtivos, fazendo com que houvesse intercâmbio de informações entre diferentes processos em setores distintos de uma empresa, otimizando o tempo e recursos despendidos para confecção de bens e/ou produtos.

Dentre as razões para a aplicação de sistemas automatizados, podem-se citar algumas, tais como: [MORAES-07]

- Tarefas que exigem grande repetibilidade e qualidade;
- Execução de tarefas agressivas;
- Possibilidade de redução de custos de produção;
- Restabelecimento rápido do processo produtivo;

- Possibilidade de introdução de sistemas produtivos integrados;
- Utilização de computadores tipo PC na programação dos dispositivos.

Conforme expõe [TOVAR-03], que aborda a integração como a solução para interligar as *Automated Islands* aos segmentos administrativos, ou seja, a possibilidade de os subsistemas da empresa poderem interagir por meio de sistemas de comunicação de dados comuns.

A figura 45 ilustra interação entre subsistemas e base de dados comum.

Figura 45: Base de Dados Comum

Fonte: Adaptado de [TOVAR-03]

Empresas do segmento de desenvolvimento soluções para automação industrial têm focado esforços no aprimoramento de produtos e de sistemas que possibilitem a integração de sistemas computacionais com confiabilidade e robustez aplicada aos níveis de fabricação (controladores lógicos programáveis). A integração dos processos produtivos associados aos sistemas corporativos possibilitou o surgimento de sistemas integrados de gestão empresarial, conhecidos por ERP (*Enterprise Resource Planning*) que tem sido utilizado na gestão de empresas. Os

sistemas ERP surgiram a partir da evolução dos sistemas MRP (*Material Resource Planning*). Os sistemas ERP são compostos por uma base de dados única e por módulos que suportam diversas atividades das empresas.

Um dos principais desafios da automação industrial eficiente está voltado para o preenchimento da lacuna existente entre os processos de gestão e os processos fabris quanto à disponibilidade de modelos eficazes para interação e troca de informações entre estes sistemas. Conforme expõe [CUNHA-04], as dificuldades para integração entre redes industriais são muitas, dentre as quais:

- Falta de visão integrada das empresas;
- Projetos refletem falta de visão integrada;
- Os sistemas não estão preparados para esta integração;
- Faltam padrões para facilitar a integração;
- Justificativas econômicas tradicionais;
- Falta de experiência e metodologia para essa integração.

5.3 Modelo para Troca de Informações

As arquiteturas para tráfego de dados aplicados aos sistemas de automação e sistemas informatizados baseiam-se na troca de informações entre diversos dispositivos de campo, tais como sensores, controladores lógicos programáveis e sistemas de supervisão e monitoramento.

Os principais modelos utilizados para troca de informações em redes industriais são: [LOPES-00] [MORAES-07]

- Mestre/escravo, conceito baseado em hierarquia de comunicação;
- Cliente / servidor, conceito baseado em filas;
- Produtor / consumidor, conceito baseado em tabelas de comunicação;
- Distribuição de relatórios, conceito baseado em filas.

A tabela 11 a seguir apresenta estes modelos de comunicação relacionados com DLL (*Data Link Layer*) da arquitetura de redes de campo.

Tabela 11: Modelos de comunicação industrial

Modelo	Data Link Layer DLL	Programado por	Direção
Cliente-Servidor	Enfileirada	Usuário	Bidirecional
Distribuição de relatório	Enfileirada	Usuário	Unidirecional
Produtor-consumidor	Bufferizada	Rede	Unidirecional

Fonte: Adaptado de [LOPES-00]

No modelo cliente/servidor, a comunicação ocorre de forma enfileirada não escalonada. Enfileirada implica que as mensagens são enviadas na ordem fornecida para a transmissão, onde é estabelecida e obedecida prioridade para transmissão sem que seja sobrescrita das mensagens anteriores. No protocolo de comunicação de cada estação, é implementado um conjunto de filas para envio e recebimento de arquivos, fazendo com que uma estação leia o valor registrado por um sensor de campo e direcione uma mensagem para sua interface de comunicação ler esta variável do processo em outra estação, ocorrendo à seguinte seqüência: [LOPES-00]

- A mensagem solicitada aguarda numa fila de saída e será lançada para a rede na próxima vez que a outra estação se comunicar;
- O sensor de campo recebe a solicitação, que ficará aguardando em uma fila de recepção;
- O sensor define o valor de tempo solicitado e retorna este valor utilizando a mesma seqüência.

Este modelo assegura a preservação da seqüência de tempo, sendo apropriado para o tráfego de grande volume de dados que são transmitidos em pequenos pacotes.

O tempo de espera torna-se o fator determinante, uma vez que uma estação com baixo desempenho refletirá na performance do sistema inteiro, por isto, conforme expõe [LOPES-00], sistemas cliente/servidor são de difícil configuração.

O modelo produtor/consumidor apresenta como principal característica a comunicação via *buffer*, onde apenas a última versão da informação é mantida, o dado mais recente sobrescreve o anterior. Este sistema é conceituado como sendo

um sistema que utiliza grupo de *buffers* no caminho de comunicação de cada estação, sendo que cada buffer: [LOPES-00]

- Corresponde a uma variável da aplicação;
- É identificado dentro do grupo de aplicação por um rótulo lógico;
- Mantém o valor de uma variável da aplicação, esperando para ser transmitido via rede, para ser utilizado pela aplicação no processo.

Este método de comunicação de dados é complementado por 03 processos distintos: o produtor, a rede e o consumidor; que são independentes entre si, que podem operar no modo cíclico ou no modo dirigido por evento. [LOPES-00]

Sistemas com esta concepção apresentam facilidade de configuração; a principal aplicação esta no nível de controle e supervisão, conforme expõe [LOPES-00], porém possui limitação no gerenciamento de eventos e também para transmissão de quantidades elevadas de informações críticas.

Segundo [MORAES-07] expõe que estes sistemas possuem economia na transmissão de dados e oferece determinismo ao sistema, visto que o tempo de entrega de dados independe do número de dispositivos que os solicitam.

No padrão *Fieldbus Foundation*, a subcamada FAS (*Fieldbus Access Sub Layer*) utiliza os métodos de comunicação entre dispositivos cliente/servidor, produtor/consumidor e distribuição de relatório. A camada FAS utiliza funções DLL. Estes serviços estão previstos por VCRs (*Virtual Communication Relationship*) no padrão *Fieldbus*.

O método Distribuição de Relatório (*Report Distribution*) utiliza comunicação por filas, não escalonada para comunicação entre dispositivos no padrão *Fieldbus Foundation*.

6. REDES DE CAMPO INDUSTRIAIS

As redes de campo ou barramento de campos para aplicação em plantas industriais foram desenvolvidas com intuito de interligar dispositivos aplicados na automação industrial, como: sensores, atuadores, módulos de I/O (*input/output*), controladores lógicos, sistemas de supervisão e monitoramento. O conceito inicial foi estabelecer um ambiente que pudesse ser compartilhado, agregando flexibilidade, distribuição de processamento nos dispositivos no parque fabril.

O controle industrial que inicialmente era centralizado, que com a evolução das redes de campo passou a ocorrer de forma distribuída nos CLPs, definindo os Sistemas de Controle Distribuídos. [ALBUQUERQUE-07]

Nas últimas décadas, os processos de instrumentação e automatização migraram de sistemas pneumáticos, chegando aos atuais sistemas baseados em microprocessadores. No início dos anos da década de 1940, eram utilizados sinais de pressão de 3 a 15 PSI para aplicações industriais. Na década de 1960, os sistemas de controle industrial passaram a aplicar sinais elétricos analógicos de 4 a 20 mA para instrumentação e controle. Esta técnica trouxe diversas vantagens aos sistemas industriais, como redução de ruído e melhoria na confiabilidade dos dispositivos, diminuição do tempo intervenções para manutenção, e agilidade nos processos de expansão e implementação de equipamentos da fábrica.

Durante a década de 1970, com o desenvolvimento da eletrônica digital, os processadores passaram a serem utilizados para o monitoramento e controle de vários dispositivos de controle e automação.

Na década de 1980 foram desenvolvidos e implementados sensores que possuíam capacidade de processamento (dispositivos inteligentes) para aplicações no controle digital. Estes sensores conseguiram aliar baixo custo com relativa flexibilidade de instalação. Entretanto, tornava-se evidente a necessidade de padronização para os dispositivos e sistemas de automação, estas necessidades levaram a realização de fóruns de âmbito internacional que objetivaram a padronização entre as tecnologias desenvolvidas.

As organizações ISA (Instrumentation Society of América), IEC (International Electrotechnical Commission), Profibus (German National Standard) e FIP (French National Standard), formaram o comitê IEC/ISA SP50 (Standards & Practices 50), com o objetivo desenvolver um padrão de integração para a grande variedade de instrumentos de controle e automação; providenciarem interfaces para operação simultânea de vários dispositivos e suportar um protocolo de comunicação para todos os dispositivos e redes de campo.

Em 1992, o mercado de instrumentos voltados para automação industrial era liderado por dois grandes grupos, que lideravam soluções para integração dos dispositivos e sistemas de automação, a ISP (*Interoperable Systems Project*) e a *WorldFIP (Factory Instrumentation Protocol)*. Em 1994, a ISP e *WordFIP* uniram-se e criaram a *Fieldbus Foundation*; esta iniciativa teve como objetivo agilizar o processo de padronização das redes de campo. [STEMMER-01]

Conforme expõe [ALBUQUERQUE-07], as quatro principais camadas do *Fieldbus Foundation* foram padronizadas pelo ISA SP 50 em 1996, ou seja, camadas: física, enlace de dados, subcamada de acesso *fieldbus* e especificação de mensagem *fieldbus*; e sua constituição básica é demonstrada na figura 46, que ilustra as camadas *Fieldbus* e a relação destas camadas com o modelo OSI.

Figura 46: Camadas *Fieldbus Foundation* e Modelo de Referência OSI

Fonte: [FOUNDATION-07]

6.1 Camada Física

A camada física, proposta na ISA SP 50 define o meio de transmissão utilizada na interligação de dispositivos, compondo-se de outras 03 subcamadas: [STEMMER-01]

- DIS (*Data Independent Sublayer*), atuação como interface com a camada de enlace, recebe e repassa solicitações de serviços;
- MDS (*Médium Dependent Sublayer*), atuação na codificação dos dados que serão enviados em formato compatível com meio físico de transmissão;
- MAU (*Médium Attachment Unit*), descreve o transceptor para o meio físico.

Conforme expõe [ALBUQUERQUE-07], o nível físico define o meio para transporte das mensagens, sendo definidos três meios físicos para o *Fieldbus Foundation*: cabos, fibra óptica e sinal de radio (ISA -550-02).

6.2 Camada Link de Dados (Enlace)

Esta camada detecta erros de transmissão, monitorando a comunicação, assegurando integridade de mensagem por meio de CRC (*Cyclical Redundancy Check*); oferece também quatro classes de funções a serem implementadas nas estações, conforme estas implementações sejam requeridas, sendo as funções de: [STEMMER-01]

- *Responder*, estação atua como estação escrava, respondendo as solicitações;
- *Initiator*, estação utiliza o meio de acesso (por meio de *token*), enviando e requisitando dados a outras estações;
- *Link-Master*, atuação semelhante ao barramento FIP, incluindo as funções de *Responder* e *Initiator* utilizando também *token* e tempo interno do sistema;
- *Bridge*, estação atua com capacidade de interligar outras entidades de enlace diferentes.

A camada de *link* de dados atua também fornecendo quatro classes de serviços à camada de aplicação: gerenciamento de *buffers* e filas, transferência de dados com conexão, transferência de dados sem conexão e o escalonamento de transações.

Conforme aborda [ALBUQUERQUE-07], o acesso ao meio é definido pelo LAS (*Link Active Scheduler*), por meio de uma requisição de *token* por dispositivos conectados ao barramento, onde o LAS transmite o *token* a outro dispositivo requisitante, via pré-configuração ou escalonamento.

6.3 Camada de Aplicação

Nesta camada são definidas as formas de transmissão das mensagens de dados, ou seja, modos: cíclico, acíclico ou requisitadas pelo consumidor (tempo real); para que todos os dispositivos conectados possam interagir, fornecendo os serviços de controle aplicados aos processos. Nesta camada estão previstos os serviços de: [STEMMER-01]

- MCSE (Message Common Service Element);
- MSE (Industrial Message Service Element);
- DDM (Distributed Database Maintenance).

A camada de aplicação prove levantamento estatístico da rede, com detecção de falhas, adição e remoção de estações na rede.

6.4 Camada de Apresentação (Usuário)

Na camada de apresentação, também designada como camada de usuário, conforme define [ALBUQUERQUE-07] é estabelecido a estrutura de mensagens para implementações estratégicas de controle por meio de conexões em blocos funcionais. Conforme aborda [FOUNDATION-07], que complementa citando que os Blocos Funcionais representam as funções de automação básicas, que são executadas por aplicações do bloco funcional. “Cada bloco funcional processa parâmetros de entrada, de acordo com um algoritmo específico e um conjunto interno de parâmetros de controle.”

7. ARQUITETURAS DE REDES INDUSTRIAIS

As arquiteturas de padrões de Redes Industriais destinam-se a integrar os componentes de um sistema de automação e são caracterizadas em função do protocolo utilizado, e pela atuação nos níveis da cadeia produtiva (sensores, processo, controle); possuem características específicas para atender as necessidades determinadas do ambiente e do processo, tais como:

- Comportamento temporal;
- Confiabilidade;
- Conectividade;
- Requisitos de meio ambiente.

A integração de componentes e sistemas de automação atende as aplicações de CIM (*Computer Integrated Manufacturing*).

7.1 FieldBus

O protocolo *Fieldbus* foi desenvolvido pela *Fieldbus Foundation*, foi um padrão desenvolvido para automação de sistemas industriais, elaborado pela *Fieldbus Foundation* e normalizado pela ISA SP 50 (*International Society for Measurement and Control*) e pela IEC 61158.

Este padrão tem como objetivo a interligação de instrumentos e dispositivos de controle e automação de processos industriais. O *Fieldbus* interliga dispositivos primários de automação, por meio de uma linha de comunicação serial, de forma bidirecional a um sistema integrado de automação e controle, conforme ilustra a figura 47 a seguir.

O protocolo *Fieldbus* pode ser utilizado com sistemas supervisórios de aquisição de dados conhecidos por SCADA (*Supervisory Control and Data Acquisition*).

Arquitectura das redes FIELDBUS

Figura 47: Arquiteturas das Redes *Fieldbus*

Fonte: [FOUNDATION-07]

O tipo de tráfego de dados no *Fieldbus* é determinístico para variáveis de processo e não determinístico para parametrização e diagnósticos. O padrão *Fieldbus* foi desenvolvido com base no padrão OSI, porém nem todas as camadas do *Fieldbus* foram implementadas. Segue na figura 48, comparação entre o Modelo OSI e o *Fieldbus*.

Figura 48: Comparação entre o Modelo OSI e o padrão *Fieldbus*.

Fonte: [FOUNDATION-07]

O padrão *Fieldbus* esta dividido em 02 níveis principais:

- Nível físico;
- Nível *software*.

7.1.1 Nível Físico

Nível responsável pela interligação física entre os dispositivos e instrumentos de campo, controladores lógicos programáveis e sistemas de supervisão e aquisição de dados (SCADA). São definidos também neste nível os padrões elétricos para ligações, com suas respectivas características elétricas que estarão integrando a rede *Fieldbus*.

Neste nível também é definido a divisão de velocidade de transmissão, sendo: [CUNHA-04]

- H1 – (*Lower Speed Fieldbus*) com 31,25 Kbps;
- H2 – (*Higher Speed Fieldbus*) com 100 Mbps ou 1Gbps.

O padrão *Fieldbus* H1 de 31,25 Kbps podem operar nas mesmas instalações onde são utilizados os padrões 4-20 mA, permitindo atualização gradual do sistema de automação da planta, é utilizado para transmissão de dados em tempo real de comunicação com equipamentos e instrumentos.

O padrão H1 é um protocolo interoperável, suportado por grande parte dos fornecedores de instrumentos, sendo vinte e cinco vezes mais rápido que a maioria dos protocolos dedicados para transmissores inteligentes. A figura 49 a seguir ilustra comparação entre o Modelo OSI e o padrão *Fieldbus* H1.

Figura 49: Comparação entre Modelo OSI e padrão *Fieldbus* H1.

Fonte: [FOUNDATION-07]

O padrão *Ethernet* foi incorporado pela *Foundation Fieldbus* na especificação do padrão H2. A rede utiliza o padrão UDP/IP (*User Datagram Protocol* / *Internet Protocol*) sobre as camadas de enlace *Ethernet*. O padrão foi definido como HSE e opera nas aplicações de nível mais elevado dentro da rede *Fieldbus* e com velocidade de 100 Mbps, sendo que o padrão H1 destina-se aos dispositivos de campo.

A interligação entre a rede padrão H1 e o padrão H2 *Fieldbus* é feito por meio de *linking device*, que realiza conversão entre os dados na rede H1 em mensagens, utilizando os protocolos padrões da *Ethernet*.

A especificação dos padrões é definida pela ANSI/ISA - Fieldbus Standard for Use in Industrial Control System Part 2 Physical Layer Specification and Service Definition.

Alguns itens da especificação são destacados no padrão devido a sua relevância:

- Transmissão de dados somente de forma digital;
- Comunicação na forma bi-direcional;
- Utilização do código *Manchester*;
- Modulação de voltagem;

- Transmissão com ou sem energização;
- Tráfego determinístico para variáveis de processo e não determinístico para parametrização e diagnósticos.

Especificações quanto à tolerância de falhas, redundâncias, segurança, velocidade, bem como aplicações em funcionamentos críticos são também tratados pela norma, conforme alguns tópicos relacionados abaixo:

- Segurança intrínseca, impedância maior que 400 ohms para frequência variando de 7,8 a 39 KHz; e por meio de barreiras e equipamentos intrinsecamente seguros de acordo com o padrão FISCO (*Fieldbus Intrinsically Safe Concept*);
- Não pode ocorrer interrupção no funcionamento da rede *Fieldbus* durante a conexão/desconexão de qualquer dispositivo;
- Redundância de fonte, de *host*, de condicionador de sinal e de *Link Active Scheduler* (LAS), redundância via acopladores redundantes;
- Na ocorrência de falhas em elementos de transmissão, a comunicação não poderá ser prejudicada por mais de 1ms.

7.1.2 Nível de Software

Este nível do padrão *Fieldbus* é tratado pelo *software* supervisor de aquisição de dados (SCADA), transparente para o usuário, que se divide em camadas:

- Subnível de enlace (*data link layer*); neste subnível é assegurada a transmissão da mensagem ao destinatário, onde é corrigido e garantido o controle de rede e roteamento de mensagens e definição das transmissões;
- Subnível de aplicação (*application layer*); neste subnível é definida a sintaxe das mensagens e modo de transmissão de cada mensagem podendo ser cíclica, imediata, uma única vez ou quanto for requisitada;
- Subnível do usuário (*user layer*); neste subnível é definida a forma pela qual pode ser feito o acesso a informações dentro de equipamentos *Fieldbus*, podendo-se distribuir as informações para outros dispositivos da rede.

7.2 Profibus

O *Profibus* (*Process Fiel Bus*) DIN 91, barramento de campo líder na Europa, foi desenvolvido pelo projeto *Fieldbus* com participação do Ministério Alemão de Investigação e Tecnologia. Sua tecnologia foi desenvolvida e administrada pela *Profibus User Organization*, cujos membros formam a *Profibus International*.

O padrão está definido pelas normas EN50170 e EN50254, que asseguram a independência de fabricante, sendo considerado sistema aberto e pelo IEC 61158 e IEC 61784 e pela DIN 19245 (padrão alemão).

O *Profibus*, segundo sua própria definição, especifica as características técnicas e funcionais de um sistema *fieldbus* serial no qual controladores digitais descentralizados podem ser interligados do nível de campo ao nível de célula.

A arquitetura do *Profibus* está baseada no modelo OSI, sendo utilizadas apenas as camadas 1, 2 e 7, ou seja, físicas, enlace e de aplicação, respectivamente. A figura 50 ilustra as camadas do padrão *Profibus*

Figura 50: Arquitetura do protocolo *Profibus*.

Fonte: [PROFIBUS-06]

A camada física do padrão *Profibus* está baseado na norma EIA RS-485 (*EIA – Eletronic Industries Association*), sendo que a interligação dos dispositivos pode ocorrer por meio de par trançado blindado, podendo ser utilizado também fibra

óptica. Podem ser conectados no barramento até quatro segmentos, sendo que cada segmento pode ter até trinta e dois dispositivos.

De acordo com a aplicação, pode-se utilizar como meios de transmissão os seguintes padrões:

- RS 485 para utilização universal, em especial sistemas de automação de manufatura;
- IEC 61158-2 para aplicações em sistemas de automação em controle de processo;
- Fibra Óptica para aplicações em sistemas que requerem grande imunidade a interferências e grandes distâncias.

A tabela 12 abaixo mostra as possíveis velocidades de transmissão com suas respectivas distâncias correspondentes.

Tabela 12: Velocidades de transmissão *Profibus* DP

Baud rate (Kbit/s)	9.6	19.2	93.75	187.5	500	1500	12000
Distância/segmento (m)	1200	1200	1200	1000	400	200	100

Fonte: [PROFIBUS-06]

A codificação utilizada no padrão é binária NRZ. Na camada de enlace, o *Profibus* define outras duas subcamadas:

- Subcamada de controle e acesso ao meio (MAC);
- Subcamada de controle de ligação lógica (LLC).

A especificação MAC (*Medium Access Control*) combina dois métodos determinísticos de acesso ao meio e assegura que somente uma estação tem o direito de transmissão de dados a cada intervalo de tempo; durante a comunicação entre controladores lógicos programáveis complexos deve ser assegurado tempo suficiente para que sejam executadas tarefas dentro de intervalo de tempo definido e que a transmissão de dados em tempo real de forma cíclica deve ser complementada em velocidade e simplicidade na comunicação entre dispositivos mestre e escravo.

O padrão *Profibus* possui grande aceitação na Europa e possui as seguintes versões:

- Profibus-DP (Decentralized Periphery);
- Profibus-PA (Process Automation);
- Profibus-FMS (Fieldbus Message Specification);
- Profinet (Powerfull Real-time Open Flexible Integrated Net Convergence Enterprisewide Transparent).

7.2.1 Profibus - DP

Versão *Profibus* que reúne características de alta velocidade com baixo custo. As funções básicas do *Profibus DP* são resumidas na tabela 13 a seguir, sendo: [PROFIBUS-06].

Tabela 13: Resumo de funções padrão *Profibus DP*

Tecnologia de transmissão	RS-485 (par trançado cabo de dois fios) ou fibra óptica
	<i>Baudrate</i> de 9,6 Kbits/seg. a 12 Mbits/seg.
Acesso ao BUS	Procedimento de passagem de token entre mestres e procedimento de mestre-escravo para escravos
	Possíveis sistemas mono-mestre ou multi-mestre
	Dispositivos mestre e escravo, máximo de 126 estações em um barramento de comunicação
Comunicação	<i>Peer-to-peer</i> (transmissão de dados de usuário) ou <i>multicast</i> (comandos de controle)
	Transmissão de dados do usuário mestre-escravo cíclica e transmissão de dados não cíclica mestre-escravo
Modos de operação	<i>Operate</i> : transmissão cíclica de entrada e saída de dados
	<i>Clear</i> : entradas são lidas, e saídas são mantidas em estado seguro
	<i>Stop</i> : transmissão de dados só é possível em mestre-mestre
Sincronização	Comandos de controle permitem sincronização de entradas e saídas
	<i>Sync mode</i> : saídas são sincronizadas
	<i>Freeze mode</i> : (modo de congelamento) entradas são sincronizadas
Funcionalidade	Transmissão de dados cíclica entre mestre DP e escravo(s) DP
	Ativação ou desativação dinâmica de escravos individualmente
	Verificação da configuração do escravo DP
	Poderosas funções de diagnósticos, 3 níveis hierárquicos de mensagens de diagnósticos.
	Sincronização de entradas e/ou saídas
	Designação de endereços para escravos DP via o barramento
	Configuração de mestre DP (DPM1) sobre o <i>bus</i>
	Máximo de 246 bytes de entrada e saída por escravo DP
Funções de segurança e proteção	Todas as mensagens são transmitidas com <i>Hamming distance</i> HD=4
	<i>Watchdog timer</i> no escravo DP
	Proteção de acesso para I/O (<i>input/output</i>) dos escravos DP
	Monitoração da transmissão de dados com temporizados configuráveis pelo Mestre
Tipos de dispositivos	<i>Class-2 DP master</i> (DPM2): programação/configuração/DP diagnóstico de dispositivos
	<i>Class-1 DP master</i> (DPM1): controlador programável central tai com PLCs, PCs, etc.
	<i>DP slave</i> : dispositivo com I/O (<i>input/output</i>) binário ou analógico, <i>drivers</i> , válvulas, etc.

Fonte: Adaptado de [PROFIBUS-06]

O *Profibus DP* é utilizado principalmente para transferência rápida de dados cíclicos entre controladores centrais (PCs e CLPs) e dispositivos de I/O descentralizados e distribuídos (válvulas, atuadores, drivers, etc), utilizado também para transmissões paralelas com 24 V ou 0-20 mA.

7.2.2 Profibus - PA

O padrão *Profibus PA* foi desenvolvido especialmente para automação de processos, permite que sejam conectados sistemas de automação e sistemas de controle de processo com os dispositivos de campo (tais como transmissores de pressão, temperatura e nível). Em comparação com métodos convencionais proporciona redução em torno de 40%, possibilita também que sensores e atuadores possam ser conectados a um barramento comum em área de segurança intrínseca. O padrão *Profibus PA* possui segurança intrínseca por meio da utilização dos modelos EX para os padrões DP/PA (*link/couplers*), de acordo com modelo FISCO.

O padrão *Profibus PA* foi elaborado juntamente com usuários de indústrias de processos e possui os seguintes requisitos especiais para atuação nas áreas de aplicação:

- Perfil de aplicação padronizado para automação e controle de processo;
- Possibilita inserção e remoção de estações e dispositivos, em áreas de segurança intrínseca sem afetar outras estações;
- Alimentação elétrica dos dispositivos via próprio barramento, conforme padrão IEC 61158-2, que possibilita a comunicação de dados e transporte de energia no mesmo barramento. Utiliza par-trançado blindado ou sem blindagem;
- Permite utilização em áreas com potencial explosivo;
- Permite topologia em barramento ou árvore. O padrão utiliza velocidade de 31,25 Kbps, com até trinta e duas estações por segmento e um máximo de cento e vinte e sete estações (com repetidores).

O padrão *Profibus PA* foi preparado com base no perfil de comunicação DP. A descrição das funções e comportamentos dos dispositivos esta baseada no modelo de Blocos Funcionais (*Function Block Model*). As definições do padrão PA torna-o um substituto para sistemas que utilizam transmissão analógica 4 a 20 mA ou *HART*. A figura 52 a seguir ilustra a aplicação do *Profibus PA* em sistema de automação de processo.

Figura 52: Configuração típica de sistema *Profibus* em automação de processo.

Fonte: [PROFIBUS-06]

O padrão *Profibus PA* oferece intercambiabilidade e interoperabilidade aos dispositivos de campo PA de fabricante diferentes por meio da utilização de blocos funcionais que descrevem os parâmetros e funções do dispositivo, tais como:

- Bloco Físico (*Physical Block*) contém informações gerais do dispositivo;
- Bloco Transdutor (*Transducer Block*) contém dados específicos do dispositivo;
- Bloco de Entrada Analógica (*Analog Input Block – AI*) fornece o valor medido pelo sensor, com estado e escala;
- Bloco de Saída Analógica (*Analog Output Block – AO*) fornece o valor da saída analógica;
- Bloco de Entrada Digital (*Digital Input Block – DI*) fornece o valor da entrada digital para o sistema;
- Bloco de Saída Digital (*Digital Output Block – DO*) fornece o valor da saída digital para o sistema de controle.

7.2.3 Profibus-FMS

O *Profibus FMS* é considerado uma solução de uso geral para tarefas extensas em comunicação, utilizado para comunicação em nível de célula. O padrão FMS

oferece um amplo leque de aplicações com grande flexibilidade, sendo utilizado na comunicação entre controladores lógicos programáveis.

As versões *Profibus DP* e *FMS* podem ser utilizadas simultaneamente no mesmo cabo, o que traz grande flexibilidade para o padrão em termos de expansões da planta. Ambos os padrões (*DP* e *FMS*) podem ainda executar o mesmo dispositivo de campo. No *Profibus FMS* são implementadas as camadas 1,2 e 7 e são de extrema importância. O padrão *FMS* define uma ampla seleção de serviços de comunicação mestre-mestre ou mestre-escravo.

A camada de aplicação (nível camada 7) do padrão *FMS* é composta por:

- Fieldbus Message Specification (*FMS*);
- Lower Layer Interface (*LLI*).

O modelo de comunicação do padrão *FMS* permite que aplicações distribuídas sejam unificadas em um processo comum por meio da utilização de relacionamentos de comunicação, a parte da aplicação situada no dispositivo de campo pode ser acessada via comunicação é denominada de dispositivo virtual de campo (*VFD – Virtual Field Device*). Os objetos de comunicação de dispositivos *FMS* são registrados em um dicionário de objetos. Os objetos de comunicação podem ser classificados em 02 tipos:

- Objetos de comunicação estática são registrados no dicionário de objetos estáticos e configurados uma única vez e não podem ser alterados durante a operação;
- Objetos de comunicação dinâmica são registrados na seção dinâmica do dicionário de objetos e podem ser modificados durante a operação.

Os serviços do padrão *FMS* são um subconjunto dos serviços *MMS*, que foram otimizados e estendidos, estão divididos nos seguintes grupos:

- Serviços de gerenciamento do contexto;
- Serviços de acesso a variáveis;
- Serviços de gerenciamento do domínio;
- Serviços de gerenciamento de chamada de programa;
- Serviços de gerenciamento de eventos;

- Serviços *VFD Support* para identificação e *status*;
- Serviços de gerenciamento OD para dicionário de objetos.

Além dos serviços citados acima, o padrão FMS dispõe de funções de gerenciamento de rede FMA7 (*Fieldbus Management Layer 7*), que podem ser ativadas de forma local ou remota. As funções de gerenciamento permitem:

- Gerenciamento do contexto, utilizado para estabelecer ou desconectar uma conexão;
- Gerenciamento da configuração, utilizado para acessar variáveis, contadores e parâmetros das camadas 1 e 2;
- Gerenciamento de falha, utilizado para indicar eventos, falhas e reiniciar dispositivos.

7.2.4 Profinet

Para a elaboração do *PROFINET* (*Powerfull Real-time Open Flexible Integrated Net Convergence Enterprisewide Transparent*), vários esforços foram conduzidos no sentido de reunir em uma única solução de automação as características do padrão *Profibus* e da versatilidade da arquitetura TCP/IP.

O *Profinet* foi definido em conformidade com o *Physical Layer* ISO/IEC 8802-3, cujo *datalink layer* esta de acordo com o TCP/UDP/IP/*Ethernet*, conforme expõe [PROFIBUS-06]. A figura 53 abaixo ilustra estrutura de dispositivo *Profinet*.

Figura 53: Estrutura de dispositivo *Profinet*

Fonte: [PROFIBUS-06]

O *Profinet* foi o resultado do desenvolvimento de um padrão que atende:

- Roteamento direto do TCP/IP para *Profibus*;
- Configuração e monitoramento de dispositivos *Profibus* via *Ethernet* e Internet (TCP/IP);
- Acesso de dados de dispositivos de campo em ambientes de TI;
- Suporte a todas as configurações *Fieldbus*;
- Acessibilidade e utilização de ferramentas de ambiente TI em dispositivos de campo;

Comunicação em protocolo considerado totalmente aberto. A figura 54 ilustra a aplicação de padrão *Profinet* com acesso aos dispositivos via TCP/IP e *Ethernet*.

Figura 54: Integração *Profinet* com *Ethernet* e TCP/IP

Fonte: [PROFIBUS-06]

A comunicação pelo padrão *Profinet* suporta 03 níveis de atuação:

- Dados de engenharia e dados não urgentes são transmitidos pelo padrão *Ethernet* e TCP/IP. Esta comunicação padrão ocorre entre dois dispositivos de campo inteligentes em estrutura de automação distribuída;
- Para transmissão de dados de processo urgentes, existe canal disponível de Tempo Real;
- Para aplicações de mecanismos síncronos tais como controle de movimento, existe uma comunicação de tempo real que, á uma taxa de

pulso horário de menos de 1milisegundo, suporta uma taxa de exatidão de salto de 1microsegundo.

O *Profinet* é considerado o primeiro padrão aberto da *Ethernet* Industrial com segurança integrada para aplicações seguras tipo sem-fio (*Wireless*) que podem ser adotadas em WLAN Industrial. [PROFIBUS-06]

7.3 Modbus

O padrão *Modbus* é um protocolo de comunicação de dados orientado à caracter, utilizado em sistemas de automação industrial. Foi desenvolvido pela *Modicon Industrial Automation Systems*, atual *Schneider*, na década de 1970.

O *Modbus* tem sido largamente utilizado para redes de industriais pela simplicidade e facilidade de implementação. A *Schneider Electric* disponibilizou as normatizações que definem o padrão para domínio público, difundindo a tecnologia do padrão.

O padrão *Modbus* apresenta como características:

- Utiliza RS 232, RS 485 ou *Ethernet* TCP/IP para transmissão;
- Estrutura de troca de mensagens/comunicação mestre/escravo entre dispositivos inteligentes;

Possui comandos para envio de dados discretos (entradas e saídas digitais) ou numéricos (entradas e saídas analógicas).

No modelo de comunicação mestre/escravo, um único dispositivo mestre inicia a comunicação, solicitando aos demais dispositivos (considerados escravos) dados e informações requisitadas. Geralmente o dispositivo mestre é o sistema supervisor.

A figura 55 a seguir apresenta o fluxo de troca de informações no padrão *Modbus*.

Figura 55: Transações entre dispositivos *Modbus*

Fonte: [ALBUQUERQUE-07]

No padrão *Modbus* existem 02 modos de transmissão de dados:

- ASCII (American Code for Information Interchange);
- RTU (Remote Terminal Unit).

No modo ASCII, cada byte de mensagem é encaminhado como dois caracteres ASCII. A tabela 14 apresenta o formato da mensagem no modo ASCII.

Tabela 14: Formato mensagem ASCII

<i>Start</i>	Endereço	Função	Dados	CRC	<i>End</i>
(0x3A)	2 chars	2 chars	N chars	2 chars	CRLF

Fonte: Adaptado de [ALBUQUERQUE-07]

No modo RTU, cada *byte* de mensagem é enviado como dois caracteres hexadecimais de 4 bits. A tabela 15 mostra o formato da mensagem no modo RTU.

Tabela 15: Formato mensagem RTU

<i>Start</i>	Endereço	Função	Dados	CRC	<i>End</i>
3..5 chars	8 bits	8 bits	N x 8 bits	16 bits	3..5 chars

Fonte: Adaptado de [ALBUQUERQUE-07]

O *Modbus* está baseado no modelo mestre-escravo, onde o dispositivo mestre requisita informações às estações escravo. No modo requisição/resposta, o dispositivo mestre encaminha uma requisição ao dispositivo escravo específico.

O padrão *Modbus* sofreu variações, que foram feitas para integrá-lo com outros meios de transporte, tais como:

- *Modbus* TCP/IP;
- *Modbus* PLUS.

No padrão *Modbus* TCP/IP, o *Modbus* está implementado na camada de aplicação da arquitetura TCP/IP. Quando o *Modbus*/TCP é utilizado, o mecanismo de controle e acesso é o CSMA/CD e as estações utilizam o modelo cliente/servidor. [ALBUQUERQUE-07].

A figura 56 ilustra esta implementação.

Figura 56: *Modbus* TCP/IP

Fonte: [MODBUS-06]

Neste modelo, os dados são encapsulados em formato binário, em *frames* TCP para a utilização do meio físico *Ethernet*. O algoritmo de controle de acesso ao meio de transmissão utilizado é o CSMA/CD. [ALBUQUERQUE-07]

O padrão *Modbus*PLUS dispõe de recursos adicionais para roteamento, diagnóstico e endereçamento, o modelo é utilizado para controladores de processo, para comunicação entre controladores lógicos programáveis. O meio físico de utilizado é o RS 485 com controle de acesso por meio do HDLC (*High Level Data Link Control*) com taxas de transmissão de 1Mbps.

7.4 HART

O protocolo *HART* (*Highway Addressable Remote Transducer*) foi desenvolvido pela *Fisher Rosemount* durante a década de 1980. O padrão *HART* é uma forma de transmissão de dados em alta fidelidade para grandes distâncias.

O protocolo foi aberto à comunidade em 1990, definido pela norma HFC 5.0 [CUNHA-04]. Em 1993, foi criada a organização HFC (*Hart Communication Foundation*) para difundir a tecnologia.

A comunicação do protocolo *HART* tradicional (analógico) ocorre por meio de sinal de 4 a 20 mA e por sinal digital, baseado em FSK (*Frequency Shift Key* – padronização telefônica 202 Bell).

A topologia do padrão *HART* possibilita diversos modos de comunicação: multi-mestre, mestre/escravo, multidrop, conforme expõe [ALBUQUERQUE-07].

A figura 57 abaixo ilustra formato padrão do comando *HART*.

PR	SD	AD	CD	BC	Status	Data	Parity
----	----	----	----	----	--------	------	--------

Legenda:

PR - Preâmbulo

SD - Start byte

AD - Endereço

CD - Comando

BC - Número de Bytes da Mensagem

Figura 57: Formato padrão do Comando *HART*

Fonte: Adaptado de [HART-03].

A especificação do protocolo *HART* esta baseado no Modelo de Referência OSI; a camada física possibilita transmissão de *bits* entre dispositivos da rede e na camada de aplicação definem-se os comandos do protocolo.

A figura 58 a seguir ilustra o as camadas do padrão *HART*.

Figura 58: Camadas do padrão *HART*.

Fonte: Adaptado de [HART-03]

O padrão *HART*, inseriu a tecnologia *wireless* para o padrão que ficou definido como *WirelessHART*, sendo composto pelas camadas: física, enlace, rede, transporte e aplicação e opera na faixa de frequência de 2,4 GHz a 250Kbps.

Conforme expõe [CAMPOS-09], “o subcomitê ISA100.12 analisa a possibilidade do *WirelessHART* ser adotado como padrão ISA100 para instrumentos de campo.”

7.5 WorldFip

O padrão *WorldFIP* (*World Factory Instrumentation Protocol*) foi elaborado para interligar sensores e atuadores; o desenvolvimento do padrão levou a elaboração da Norma Francesa NFC 46-600.

A proposta permite a interligação de sensores e atuadores, levando em consideração as restrições de tempo real impostas por um grande número de aplicações em nível de chão-de-fábrica. No padrão, definiu-se o modo de transmissão produtor/consumidor, que possui mecanismo de controle centralizado na comunicação. O padrão possui as características de um barramento de campo.

Atualmente, o *WorldFIP* é um padrão europeu (EN50170) e utiliza o padrão da camada física IEC 1158-2, possuindo ainda um padrão de mensagem MMS da ISO. O padrão utiliza as três primeiras camadas do modelo OSI, ou seja, camada física, camada *link* de dados e camada de rede.

Na camada física, o padrão tem como meio de transmissão par trançado (blindado ou não) e fibra óptica. O número de *host* por controlador pode ser de sessenta e quatro por segmento, com até quatro segmentos com repetidores, os cabos devem ser terminados em cada extremidade e a redundância é opcional.

O modo de transmissão é síncrono e o método de acesso ao meio é centralizado. Conforme expõe [STEMMER-01], para o par trançado estão padronizadas três velocidades de transmissão de dados:

- S1: 31,25 Kbps, distância de até 2000 m;
- S2: 1 Mbps, distância de até 500 m, com par trançado blindado;
- S3: 2,5 Mbps, distância de até 200 m, com par trançado blindado.

Para utilizações com fibra ótica, a velocidade permitida é de 5 Mbps.

A velocidade S2 é padrão do modelo, as velocidades S1 e S3 são utilizadas apenas para aplicações especiais, onde S1 é utilizada em área sujeitas à explosão e o S3 é utilizado em aplicações de desempenho elevado.

O padrão suporta até duzentas e cinquenta e seis estações e o barramento principal pode ser decomposto em segmentos, podendo ser interligado por repetidores. A camada de enlace do *WorldFIP* não faz uma distinção formal entre as subcamadas LLC e MAC, como proposto na norma IEEE 802.

O método de acesso ao meio é baseado na difusão (*broadcasting*), organizada por uma entidade centralizada denominada árbitro do barramento. O padrão *WorldFIP* teve como base o fato de que a informação em rede industrial pode ser compartilhada para varias outras estações na rede. A maioria dos dados transmitidos pelo barramento de campo é representada por objetivos variáveis com um único nome no sistema. Um objeto é definido por um único transmissor, classificado pelo modelo de transmissão como produtor e os receptores da rede são definidos como consumidores, sendo que a comunicação ocorre quando: [STEMMER-01]

- Num primeiro momento, o árbitro difunde na rede o nome da variável (objeto) a ser transmitida (quadro de identificação);
- O produtor da variável difunde, em seguida, a informação ligada ao identificador (quadro de dados);

Todos os consumidores interessados passam a copiá-la na fase final.

Devido à utilização da difusão, os endereços de transmissores e receptores não precisam ser conhecidos pela aplicação. Cada estação é completamente autônoma e o único requisito imposto às estações é o de difundir, por solicitação do árbitro de barramento, as variáveis por elas produzidas.

A transmissão de mensagens atende a norma IEEE 802.2 LLC (protocolo desenvolvido para controle de transmissões de dados no nível de enlace do protocolo MAC, inclui sistema de endereçamento e verificação de erros) tipos 1 e 3.

A figura 59 ilustra o formato do *frame* FIP.

Legenda:

PRE - Preâmbulo
 FSD - Frame Start Delimiter
 EB - Equalization Bit
 DFS - Data Frame Sequence
 FSS - Frame Start Sequence
 FED - Frame End Delimiter
 FES - Frame End Sequence

Figura 59: Formato do *frame* FIP

Fonte: [STEMMER-01]

Na camada de enlace, são especificados os seguintes serviços: [STEMMER-01]

- Atualização cíclica de dados;
- Atualização não periódica de dados;
- Transferência de mensagem com ACK;
- Transferência de mensagem sem ACK.

Na camada de aplicação do padrão *WorldFIP*, foi proposto à implementação de um subconjunto do MMS, onde foram definidos serviços para mensagem industrial, além da criação de uma segunda classe de serviços, a MPS (*Message Periodic / Aperiodic Services*).

O *WorldFIP* possui também implementações de gerenciamento de rede, que permitem configurar rede, implementações de testes, detecção e correção de falhas, atualizações de lista de objetos e tabelas além de funções de gerenciamento de operações de partida e parada.

7.6 EtherNet/IP (Ethernet Industrial Protocol)

O protocolo EtherNet/IP (Ethernet Industrial Protocol) foi desenvolvido pelo ODVA (Open DeviceNet Vendor Association).

Conforme expõe [SOUSA-03], o protocolo define o modelo de comunicação entre dispositivos produtor/consumidor. O padrão está baseado na *Ethernet*, definida pela especificação IEEE 802.3 juntamente com protocolo CIP (*Control Information Protocol*).

A figura 60 abaixo ilustra a estrutura do protocolo *EtherNet/IP*.

Figura 60: Arquitetura *EtherNet/IP*

Fonte: Adaptado de [ODVA-07b]

Na arquitetura *EtherNet/IP*, as mensagens são encapsuladas via protocolos TCP ou UDP na camada de transporte. Conforme expõe [ODVA-07a], devido ao método cíclico possuir melhor desempenho, o *EtherNet/IP* comumente utiliza o UDP para transporte de mensagens.

A figura 61 a seguir ilustra a estrutura de encapsulamento de mensagens.

Estrutura	Nome do Campo	Tipo de Dados	Descrição
Cabeçalho	Command	UINT	Comando
	Length	UINT	Tamanho em bytes dos dados da mensagem
	Session handle	UDINT	Identificador de sessão
	Status	UDINT	Código de estado
	Sender context	ARRAY de octetos	Informação somente para o remetente da mensagem
	Options	UINT	Opções
Dados	Encapsulated data	ARRAY de 0 a 65511 USINT	Dados da mensagem

Figura 61: Encapsulamento de mensagens *EtherNet/IP*

Fonte: Adaptado de [ODVA-07a]

No cabeçalho de encapsulamento de mensagens não é disposto campo para distinção entre mensagens de requisição e mensagens de respostas, portanto as transmissões são definidas em: [LUGLI-07]

- Implícitas, que utilizam o UDP e a camada CIP;
- Explícitas, que utilizam o TCP e a camada CIP.

Conforme expõe [FELSER-04]¹ [ETHERNET-01]² apud [LUGLI-07], a camada de aplicação CIP não oferece comunicação com a camada de aplicação *Profinet*, não havendo, portanto, interoperabilidade entre as diversas redes *Ethernet*.

7.7 CAN

O CAN foi normalizado pela ISO em 1993 e esta implementado em larga escala em diversos produtos comerciais, onde o padrão ISO 11898 (até 1 Mbits/seg) é destinado para comunicação serial de alta velocidade e o padrão ISO 11519 (até 125 Kbits/seg) utilizado em baixas velocidades. [ALBUQUERQUE-07]

¹ FELSER, Max; SAUTER Thilo. **Standardization of Industrial Ethernet – the Next Battlefield?** IEEE Article, 6th IEEE International Workshop on Factory Communication Systems, Vasteras, Sweden, 2004, 413-421 p. [FELSER-04]

² **Ethernet/IP Specification Release 1.0** – June, 2001. Norma *Ethernet/IP*. ODVA, 2001. [ETHERNET-01].

Conforme expõe [CABRAL-05], o protocolo CAN implementa um mecanismo de acesso ao meio MAC (*Medium Access Control*), do tipo CSMA-CA (*Carrier Sense Multiple Access – Collision Avoidance*), onde o barramento deve estar livre para a estação iniciar a transmissão, livre de colisões.

O mecanismo de resolução de colisão é baseado no campo *Arbitration Field*. O barramento utiliza o NRZ (*non-return to zero*) para *bit-stuffing*¹.

A estrutura do protocolo CAN é ilustrada na figura 62 abaixo.

Legenda:

SOF - Start Of Frame

RTR - Remote Transmission Request

CRC - Cyclic Redundancy Code

EOF - End Of Frame

DLC - Data Length Code

Figura 62: Estrutura do protocolo CAN

Fonte: Adaptado de [CABRAL-05]

O protocolo CAN serviu como base para o desenvolvimento do padrão *DeviceNet*.

O *DeviceNet* foi elaborado pela *Allen Bradley* e possui especificação aberta, define uma rede baseada no método de transmissão produtor/consumidor. [STEMMER-01].

Para o desenvolvimento de *hardware* baseado no protocolo *DeviceNet*, os fabricantes devem ser licenciados pelo ODVA (*Open DeviceNet Vendor Association*).

¹ No caso de protocolos orientados a *bit*, a técnica *Bit Stuffing* permite que o usuário envie qualquer sequência de *bits* como informações que contêm sequências de bits de flag não sejam confundidas com os delimitadores da mensagem. [<http://penta.ufrgs.br/tp951/protocolos/6bitstu.html>]

A figura 63 abaixo as camadas do protocolo *DeviceNet*.

Figura 63: Camadas do padrão *DeviceNet*.

Fonte: Adaptado de [STEMMER-01]

O padrão *DeviceNet* prove tráfego de dados por meio de comunicação: produtor/consumidor, *polling*, *cyclic*, *multi-master* e *multicast*.

7.8 LONWorks

A rede operacional local *LONWorks* (LON – *Local Operating Network*) foi desenvolvida pela *Echlon* em 1990. O protocolo possui grande utilização na automação predial e de sistemas de escritórios (edifícios). Conforme expõe [CHERMONT-07], a arquitetura de protocolos *LonWorks*, define uma arquitetura aberta.

As redes *LonWorks* são caracterizadas como redes *peer-to-peer*.

Os principais componentes da de *LONWorks* são: [ALBUQUERQUE-07]

- Protocolo de *LonTalk*;
- Chips Neurais;
- Transceptores;
- Administração de cadeia;
- *Software* de aplicação.

O protocolo *LonTalk* esta especificado pela norma ANSI/EIA/CEA 709.1, sendo um conjunto de serviços que viabilizam a comunicação entre os dispositivos da rede LON. Conforme expõe [STEMMER-01], o protocolo *LonTalk* esta baseado no Modelo de Referencia OSI e implementa as sete camadas previstas.

Conforme cita [LONWORKS-09], em 2009, o protocolo *LonWorks* foi ratificado como um padrão global para sistemas de controle em edificações, sendo definida a norma ISO / IEC 14908-1.

O protocolo provê um conjunto de serviços que possibilita que um programa de aplicação do dispositivo enviar e receber mensagens de e para outros dispositivos de rede sem a necessidade de conhecer a topologia da rede ou a outros dispositivos nomes, endereços ou funções.

O protocolo *LonWorks* (ISO/IEC14908-1) e as redes *LonWorks*, podem ser implementadas através de qualquer meio físico, incluindo a linha de força, par trançado, rádio frequência (RF), infravermelho (IR), cabos coaxiais e fibras ópticas.

7.9 BACNet

O padrão *BACNet* (*Building Automation and Control NETWORKS*) foi elaborado em 1987. O protocolo esta especificado pela normas ISO16484 e ANSI/ASHRAE 135/1995, tendo sido definido inicialmente pela ASHRAE (*American Society of Heating, Refrigerating and Air-Conditioning Engineers*), definido como um protocolo aberto e de alto desempenho.

A comunicação *BACNet* é baseada no modelo cliente/servidor; o equipamento denominado cliente envia um pedido de alguma função para o servidor que a processa e envia de volta o resultado obtido. As redes *BACNet* definem trinta e cinco tipos de mensagens divididas em cinco grupos de classes que vão desde alarmes, eventos e funções definidas.

O padrão *BACNet* opera a taxas de 9600 a 76800 bps, com possibilidade de ate 127 dispositivos por segmento. As principais aplicações do padrão são realizadas em automação predial, sistemas de climatização, inversores de frequência e instrumentação.

Os diversos protocolos e padrões para comunicação industrial foram desenvolvidos para aplicações específicas destinadas ao tráfego de dados no ambiente industrial. Nota-se o surgimento de vários padrões que incorporaram a *Ethernet* e o TCP/IP em suas arquiteturas como forma de torná-los flexíveis quanto a atender todos os níveis para o tráfego de informações, inclusive no ambiente corporativo.

A figura 64 ilustra e situam alguns padrões citados neste trabalho dentro do contexto corporativo e industrial.

Figura 64: Aplicação de redes industriais

Fonte: Adaptado de [GEORGINI-00]

8. SISTEMAS DE AQUISIÇÃO DE DADOS

Os sistemas de monitoramento dedicados à automação industrial têm como objetivo a elevação do desempenho dos dispositivos interligados ao sistema automatizado, assegurando modularidade e escalabilidade da planta industrial, além de facilidades de programação e integração de novos dispositivos aliados à segurança, confiabilidade e robustez do sistema industrial.

Os elementos que compõem o sistema de automação: sensores, transdutores, atuadores, dispositivos de aquisição de dados, unidades remotas, CLPs; devem ser supervisionados para que tais objetivos possam ser atingidos. Conforme define [ALBUQUERQUE-07], o controle supervisório “denota o processo de monitorar à distância uma atividade, transmitindo diretrizes de operação aos controladores localizados à distância e recebendo de volta a indicação da realização das ações de controle.”

Em sistemas baseados em redes de barramentos de campo, são utilizados sistemas de controle e aquisição de dados (SCADA – *Supervisory Control & Data Acquisition Systems*) para esta finalidade. A figura 65 ilustra uma aplicação de sistema supervisório.

Figura 65: Aplicação de Sistema Supervisório

Fonte: [SILVA-05]

A evolução dos sistemas SCADA esta ligada ao desenvolvimento dos atuais sistemas de informação e computação, e conforme expõe [NCS-04], os sistemas SCADA podem ser classificados em três gerações, sendo:

- Primeira geração - sistemas monolíticos;
- Segunda geração - sistemas distribuídos;
- Terceira geração - sistemas *Networked*.

Os sistemas SCADA monolíticos estão associados ao conceito de computação centralizada no *mainframe*, oferecendo poucas possibilidades de comunicação com outros sistemas; a figura 66 ilustra este tipo de sistema.

Figura 66: Geração de sistemas SCADA monolíticos

Fonte: [NCS-04]

Os sistemas SCADA distribuídos evoluíram a partir dos sistemas monolíticos com utilização das redes LAN, onde a maioria dos protocolos de comunicação utilizados eram proprietários. A utilização de sistemas distribuídos permitiu a elevação da capacidade de processamento, aumento da confiabilidade com utilização de técnicas de redundância.

A figura 67 a seguir ilustra este tipo de sistema.

Figura 67: Geração de sistemas SCADA distribuídos

Fonte: [NCS-04]

Os atuais sistemas de terceira geração caracterizam-se pelas arquiteturas abertas de comunicação, utilizando padrões abertos de comunicação, que possibilita que múltiplos sistemas compartilhem dos recursos de estações *master*.

A inclusão de sistemas SCADA em arquiteturas abertas de comunicação tornou possível a expansão destes sistemas em redes WAN e não apenas em LAN com utilização de protocolos IP (*Internet Protocol*) e redes *Ethernet*. A figura 68 mostra este tipo de arquitetura, que permitiu aumento da confiabilidade dos sistemas por meio de alternativas de roteamento de comunicação.

Figura 68: Geração de sistemas SCADA *Networked*

Fonte: [NCS-04]

Nos sistemas atuais sistemas SCADA a interação com o usuário é obtida por meio de interface gráfica que representa a planta ou processo industrial monitorado, sendo necessários *hardware* e *software* específicos para esta finalidade.

Implementações realizadas em sistemas SCADA definem a próxima geração de sistemas supervisórios que se baseiam no modelo de Orientação à Objeto, que possui abordagem promissora para especificação de arquiteturas complexas, visto que: [ZOLDAN-04]

- Um sistema baseado em modelo orientado à objetos pode ser descrito como uma rede de dispositivos interconectados, provendo uma arquitetura;
- As propriedades do modelo orientado à objeto podem ser herdadas de outro objeto, fundamentais para o desenvolvimento de arquiteturas complexas, que caracteriza o recurso da herança.

O conceito de orientação à objeto no desenvolvimento de sistemas industriais de tempo real tem sido utilizado, uma vez que permite a estruturação de informações de forma lógica para sistemas complexos. Conforme expõe [ZOLDAN-04], “o reuso de *software* é a principal vantagem na orientação a objetos, eliminando a redundância de código, que é uma das principais dificuldades dos implementadores ou mantenedores de *softwares* CLP e SCADA.”

As arquiteturas convencionais de sistemas supervisórios SCADA provem suporte duas hierarquias de rede, sendo:

- Redes de Informação;
- Redes de controle.

O sistema supervisório convencional possibilita que informações do processo produtivo possam ser monitoradas, coletadas por meio de dispositivos de aquisição de dados e os dados manipulados, analisados e armazenados em banco de dados para posterior utilização de usuários do sistema. Um sistema supervisório deve apresentar funcionalidades básicas, entre as quais se destacam: [PINHEIRO-06]

- Aquisição de dados, processo que envolve a coleta e transmissão de dados desde a planta industrial (estações remotas) até as estações centrais de monitoração;

- Visualização de dados consiste na apresentação das informações através de interfaces gráficas. A figura 69 ilustra exemplo de visualização de processo industrial [JURIZATO-02];

Figura 69: Tela gráfica SCADA

Fonte: [JURIZATO-02]

- Processamento de alarmes, onde os alarmes são classificados por níveis de prioridade em função da sua gravidade. Na ocorrência de falha do servidor ou da rede de comunicações, é possível efetuar o armazenamento das mensagens de alarme em *buffer*;
- Tolerância a falhas, para atingir níveis aceitáveis de tolerância a falhas é usual a existência de informação redundante na rede e de estações de monitoramento.

Os controladores lógicos programáveis espalhados na planta industrial enviam sinais elétricos para o sistema supervisor SCADA, por meio de TAGs (mensagens digitais). As mensagens digitais TAGs podem ser do tipo: [MORAES-07]

- *Device*; significa que os dados se originaram de CLPs para o sistema supervisor;
- *DDE (Dynamic Data Exchange)*; significa que os dados se originaram de um servidor;
- *Memory*; mostra que os dados existem localmente no sistema supervisor.

Os sistemas supervisórios SCADA operam, conforme expõe [MORAES-07] de forma geral em dois modos: o modo de desenvolvimento (*Development Time*) e o modo de execução (*Run Time*).

8.1 Componentes de Sistemas Supervisórios SCADA

Os principais componentes físicos de um sistema supervisório SCADA podem ser descritos como: [SILVA-05]

- Sensores;
- Atuadores;
- Estações remotas, (RTU - *Remote Terminal Units*), onde ocorre o processamento dos dados colhidos. Os CLPs desempenham funções análogas às RTUs: a coleta, processamento e armazenamento das informações em base de dados no controle central do sistema SCADA.
- Sistemas de comunicação, definido pelas redes de comunicação, que constitui a plataforma por onde as informações do sistema são transferidas para o controle central.
- Estação de controle central, (estação *master*), que executa a supervisão e monitoramento do sistema SCADA, constituindo-se na unidade principal do sistema.

Existem três componentes de *software* para a estação *master*: o sistema operacional, a plataforma SCADA e o aplicativo de supervisão. As estações *master* podem ser centralizadas em único computador ou distribuídas em rede, possibilitando compartilhamento de recursos do sistema.

A figura 70 a seguir ilustra integração de sistemas SCADA.

Figura 70: Integração de sistema SCADA

Fonte: [EFACEC-06]

Conforme expõe [SILVA-05], os sistemas SCADA podem ter divididas as principais tarefas em blocos ou módulos, que vão permitir maior ou menor flexibilidade e robustez, de acordo com a solução desejada. Em linhas gerais, estas tarefas podem ser divididas em:

- Núcleo e sistema de processamento;
- Comunicação com outros sistemas SCADA e com CLPs/RTUs;
- Lógicas de programação interna (*Scripts*) ou controle;
- Comunicação com clientes externos / sistemas externos;
- Protocolos de comunicação.

O funcionamento dos atuais sistemas SCADA parte dos processos de comunicação com os equipamentos de campo, cujas informações são enviadas ao sistema que constitui o núcleo principal do *software*. A comunicação também possibilita a troca de dados entre estações do sistema, compartilhamento de dados, além de acesso a outras redes.

O núcleo principal (estação *master*) é responsável por distribuir e coordenar o fluxo de informações para os demais módulos, até atingirem a forma desejada pelo

operador do sistema, geralmente em forma gráfica; as estações *master* utilizam sistemas operacionais, que podem ser UNIX, LINUX, DOS e *Windows*.

Conforme expõe [NCS-04], atualmente, no segmento industrial existem diversos protocolos de comunicação para sistemas supervisórios SCADA, dentre os quais o IEC 60870.5 (*International Electrotechnical Commission*) e o DNP 3 (*Distributed Network Protocol version 3.0*).

O protocolo IEC foi elaborado com base no modelo OSI, tendo sido publicado em 1995 como padrão IEC 60870.5; o DNP 3, considerado como protocolo aberto, conforme expõe [NCS-04], desenvolvido em novembro de 1993, e publicado pela *Harris Distributed Automation Products*, o protocolo transporta dados de valores genéricos, com conjunto extensivo de funções tendo sido desenvolvido inicialmente para sistemas de distribuição de energia elétrica, que posteriormente teve aceitação na industrial, atuando em diversos sistemas SCADA.

8.2 Modos de Comunicação SCADA

A comunicação de um sistema SCADA convencional está diretamente relacionada à troca de informações entre os dispositivos de campo e estações máster, por meio de protocolos abertos ou proprietários, que podem ser: [MORAES-07]

- Comunicação com IED (*Intelligent Electronic Devices*, CLP/RTU);
- Comunicação com outras estações SCADA;
- Comunicação com outros sistemas.

Os protocolos de comunicação entre estações *master* e dispositivos de campo atendem a dois métodos: por *polling* ou por interrupção (também designado por *Report by Exception*).

A comunicação pelo método *polling* (*master/slave*) faz com que a estação central (*master*) tenha controle absoluto das comunicações, efetuando seqüencialmente o *polling* aos dados das estações remotas (*slave*), que apenas respondem à estação central após a recepção da solicitação, ou seja, em *half-duplex*. O método traz simplicidade no processo de aquisição de dados devido à redução de colisões no tráfego da rede; define comportamento determinístico;

facilidade na detecção de falhas de ligação e uso de estações remotas sem processamento. [MORAES-07]

No método de comunicação por interrupção (*Report by Exception*), ocorre quando um IED (CLP ou RTU) monitora os seus parâmetros de entrada e, ao detectar alterações significativas de valores que ultrapassem os limites definidos, envia as informações para a estação central. Isto evita a transferência de informação desnecessária, diminuindo o tráfego na rede, além de permitir uma rápida detecção de informação urgente e a comunicação entre estações remotas escravo para escravo (*slave-to-slave*). [MORAES-07].

8.3 Sistema Supervisório SCADA Orientado à Objeto

Conforme expõe [ZOLDAN-04], a partir de 1989 iniciaram-se estudos para o desenvolvimento de sistemas de controle orientado à objetos. Projetos de desenvolvimento foram conduzidos na Europa pela ERSF (*European Synchrotron Radiation Facility*) e deram origem ao TACO (*Telescope and Accelerator Controller with Objects*).

Utilizando tecnologias consolidadas como RPC (*Remote Procedure Call*) o TACO propôs a aplicação de um sistema de controle orientado à objeto.

Com base no padrão CORBA (*Common Object Request Broker Architecture*), foi elaborado o sistema TANGO (*Taco Next Generation Object*), baseado em linguagens *Java* e *C++*, o TANGO atua nas plataformas Linux, NT, entre outros sistemas operacionais.

Baseado na programação orientada à objeto, foram desenvolvidos , conforme aborda [ZOLDAN-04], sistemas supervisórios como EPICS (*Experimental Physics and Industrial Control System*), IOC (*Input-Output Controller*) e SCADA ++. Utilizando linguagem de programação de alto nível, o SCADA++ se difere dos sistemas SCADA convencionais que utilizam linguagens de baixo nível.

O SCADA++ esta especificado no IEEE C37.1 para aplicação em sistemas supervisórios e de automação.

A tabela 16 a seguir ilustra as principais características do sistema supervisório SCADA++.

Tabela 16: Características de Supervisórios SCADA++

	SCADA Convencional	SCADA ++
Linguagem de programação	Baixo nível nos CLPs	Alto nível
Integração de dados	Via banco de dados	Padrões RPC / SOAP / CORBA
Acesso ao hardware de rede	Via TAGs	Via Objetos
Sistema operacional	Dependência SO	Independência SO
Comunicação/varredura	Polling / Report by Exception	varredura ciclica

Fonte: Adaptado de [ZOLDAN-04]

A figura 71 abaixo ilustra interface gráfica desenvolvida por meio de *packages* do JAVA, que implementa o sistema supervisorio SCADA++.

Figura 71: Interface gráfica – SCADA++

Fonte: [ZOLDAN-04]

9. GERENCIAMENTO DE REDES

O gerenciamento de redes se constitui num sistema de extrema importância para o rastreamento e correção de problemas oriundos em redes de transmissão de dados e em seus componentes, bem como monitorar o desempenho da rede de comunicação; possibilita a geração de gráficos e relatórios para análises e implementações.

O modelo clássico de gerenciamento de redes pode ser sintetizado em 03 etapas, sendo: [CARVALHO-93]

- Coleta de dados: consiste no processo de monitoramento dos recursos disponíveis na rede;
- Diagnóstico: etapa que realiza a análise e tratamento a partir de dados coletados;
- Ação ou controle: uma vez detectado e diagnosticado o evento, o gerenciamento deve possibilitar ação corretiva sobre o dispositivo gerenciado.

Devido ao crescente desenvolvimento de redes heterogêneas de computadores e à diversidade de arquiteturas de redes, a tarefa de gerenciamento torna-se imprescindível. Vários esforços têm sido intensificados desde os anos de 1980 para definição de arquiteturas para gerenciamento de redes heterogêneas de computadores em ambientes de TI de forma padronizada e aberta e que ofereçam confiabilidade.

O gerenciamento de dispositivos de rede é muito amplo, envolvendo áreas funcionais de:

- Gerenciamento de falhas;
- Gerenciamento de configuração;
- Gerenciamento de contabilização;
- Gerenciamento de desempenho;
- Gerenciamento de segurança.

Conforme expõe [CARVALHO-93], o gerenciamento de redes deve coordenar os recursos materiais, *hardwares* (equipamentos, estações, *switches*, *hub*, etc.) e recursos lógicos, *softwares*, (protocolos de arquiteturas, protocolos de comunicação, aplicativos, sistemas operacionais, etc.) assegurando confiabilidade, tempos de resposta aceitáveis e segurança às informações. A arquitetura de gerenciamento que esteja incumbida de gerenciar redes e seus dispositivos deve incorporar as seguintes características:

- O gerenciamento deve ser parte integrante da rede;
- Possibilitar vários pontos de acesso ao gerenciamento da rede (estações de controle e/ou gerenciamento);
- Disponibilizar informações sobre a rede, estatísticas do desempenho, eventos ocorridos, sejam estes normais ou com problemas, deverão ser passíveis de obtenção de forma centralizada e organizados para facilitar a compreensão;
- Possibilitar adoção de esquema de tratamento prioritário que possibilite que as mensagens de controle da rede precedam outros tipos de tráfego;
- Possibilitar existência de mecanismos de segurança que limitem e controle e acesso a rede e detectem uso não autorizado à rede;
- Possibilitar que funções de gerenciamento de rede operem independentemente do meio de transmissão utilizado;
- Deve possibilitar existência de banco de dados que contenha informações sobre todos os componentes e dispositivos da rede e de seus usuários;

Alterações e redirecionamentos na rede devem ser efetivados de forma flexível e simples.

Para que o gerenciamento de redes torne-se ser eficiente, este deve permitir o gerenciamento em ambientes diversos e heterogêneos, independente dos dispositivos que compõem tal sistema. O gerenciamento de falhas utiliza *hardware* e *software* implementados, que possibilitem o gerenciamento para alertar os administradores (ou programas gerentes, que são aplicativos implementados com atribuição de gerenciamento propriamente dito em uma rede) a respeito de falhas e auxiliar no reparo.

A figura 72 abaixo ilustra sistema de gerenciamento em arquitetura distribuída.

Figura 72: Arquitetura de sistema de gerenciamento tipicamente distribuído.

Fonte: Adaptado de [STALLINGS-99]

Conforme expõe [OLIFER-08], a disponibilidade de vários gerentes possibilita que a carga seja distribuída, permitindo a escalabilidade do sistema. Normalmente são utilizados dois tipos de *enlaces* para a conexão de gerentes: não hierárquico e hierárquico.

A figura 73 a seguir ilustra estas conexões.

Figura 73a: Conexões não hierárquicas (*peer-to-peer*) entre gerentes

Figura 73b: Conexões hierárquicas entre gerentes

Fonte: Adaptado de [OLIFER-08]

Em virtude da complexidade que as redes possam atingir, bem como a diversidade de dispositivos implementados, o sistema de gerenciamento deve permitir que sejam utilizados sistemas de tolerância a falhas e/ou redundâncias de *hardware* e *software*, que podem continuar a oferecer serviços na rede, mesmo na ocorrência de eventual falha da rede. [CARVALHO-93] cita que as principais ferramentas utilizadas para gerenciamento de falhas devem atender as seguintes características:

- Sistema de gerenciamento da rede, composto por *hardware* e *software* que possibilite monitorar funcionamento dos componentes da rede;
- Analisador de protocolo, constituído por implementações de *hardware* e *software* que monitorem o tráfego de rede;

- Implementação que possibilite verificar meio físico de transmissão;
- Sistemas redundantes de *hardware* e *software*;
- Implementações para gerenciamento de desempenho; enquanto o gerenciamento de falhas possui característica reativa, o gerenciamento de desempenho possui um aspecto ativo essencial, visto que coleta e interpreta informações periódicas de indicadores de desempenho, avaliando tendências do sistema;
- Implementações para gerenciamento de segurança, que consiste em proteger dados e seus respectivos equipamentos, sejam *hardware* e *software*.

A integração de implementações e funções de gerenciamento de redes não pode ser tratada somente como adição de recursos aos equipamentos, mas como ações necessárias que agregarão segurança, eficiência e confiabilidade aos sistemas.

Nas arquiteturas de redes a estratificação de funções em níveis, onde o gerenciamento deve ser parte das funções inerentes para cada nível. Conforme expõe [CARVALHO-93], avanços da área de gerenciamento de rede ocorreram com a utilização de sistemas especialistas; na área de gerenciamento de falhas, por exemplo, sempre que um problema ocorre o sistema tenta tomar decisões, analisa o histórico de ocorrências do sistema, reduz o trabalho rotineiro executado por operadores de rede.

Por meio de análise de tráfego das redes, o sistema sugere alterações que visam aperfeiçoar os custos das redes, mantendo disponíveis os níveis de serviço.

Algumas soluções de gerenciamento de redes utilizam ferramentas e aplicativos proprietários, de diferentes fornecedores, que não se comunicam entre si, aumentando os custos e ineficiência do gerenciamento de redes, fazendo com que a integrações entre tais sistemas tornem-se onerosas e pouco eficientes devido à abrangência dos recursos que estarão sendo disponibilizados para integração.

Os sistemas de gerenciamento de um único fornecedor trabalham de forma integrada, porém, com limites de abrangência, fazendo com que os usuários utilizem somente os produtos deste fornecedor independente de custos e eficiência técnica.

Uma maneira de resolver o problema de abrangência, em que apenas um único fornecedor tende a gerenciar o sistema seria a adoção de APIs (*Application Program Interfaces*) ou sistemas que atenuem a heterogeneidade.

A utilização de APIs trouxe, além de uma solução para o problema de integração, a possibilidade dos usuários se adaptarem aos sistemas proprietários de gerenciamento e suas características. Evidente que, nenhum sistema de gerenciamento de redes atende às diferentes necessidades específicas de usuários e suas respectivas peculiaridades, portanto a utilização de APIs possibilita que cada usuário elabore seu ambiente de gerenciamento atendendo estas necessidades.

Os esforços para o desenvolvimento de APIs são enormes, visto que cada componente e dispositivo de rede apresenta características diferentes, que adicionado a rede implica no desenvolvimento de uma nova API. Cabe, contudo salientar que as limitações do sistema como um todo estão sempre relacionadas ao componente de menor capacidade em fornecer dados. Como forma de contornar este problema, muitos fornecedores desenvolveram aplicações baseadas em padrões de *facto* de mercado.

Dentre várias soluções que foram desenvolvidas, entretanto, as que se destacaram devido à aceitação foram o SNMP (*Simple Network Management Protocol*), protocolo aplicado à arquitetura TCP/IP e o CMIP (*Common Management Information Protocol*), baseado na arquitetura OSI.

Para a implementação de um sistema de gerenciamento, uma das primeiras etapas é a análise da arquitetura de rede de comunicação existente e a seleção do modelo de gerenciamento.

O protocolo CMIP atua no nível de aplicação do modelo OSI, sendo protocolo orientado a conexão, utiliza serviços do ASCE (*Association Control Service Element*), ROSE (*Remote Operations Service Element*). O padrão CMIP para gerenciamento no modelo OSI tem sido apoiado pela OSF/DME (*Open Software Foundation / Distributed Management Environment*).

A figura 74 a seguir ilustra modelo de gerenciamento OSI/CMIP.

Legenda:

CMIP - Common Management Information Protocol

LME - Layer Management Entity

MIB - Management Information Base

SMAE - System Management Application Entity

Figura 74: Modelo de gerenciamento OSI

Fonte: [CARVALHO-93]

O sistema de gerenciamento CMIP está baseado em modelo orientado à conexão, utiliza filosofia de aquisição de informações dos dispositivos gerenciados por *polling* e por técnica de notificação. Para realizar atividades de gerenciamento o CMIP necessita de maior capacidade de processamento e memória em relação a outras soluções de gerenciamento como o SNMP. [CARVALHO-93]

O SNMP opera no padrão TCP/IP e *Ethernet*, com modo de acesso não orientado a conexão e possui comandos que permitem ao usuário requisitar informações de seus objetos ou até mesmo agir sobre eles. A arquitetura SNMP pressupõe a existência de estações de gerenciamento, onde as aplicações de gerenciamento e elementos gerenciados da rede (roteadores, estações e dispositivos de comunicação) operam. Pela concepção, a aquisição de informações no SNMP é feita por *polling*, isto é, periodicamente indaga cada recurso sobre o seu *status*, adicionalmente é definido o mecanismo de TRAP, por meio do qual um recurso informa ao gerente que precisa ser submetido ao *polling*.

As implementações em SNMP tendem a ser rápidas, visto pouca capacidade de processamento e memória.

O SNMP não é um padrão internacional de *jure*, embora tenha se tornado de *facto*, e os fabricantes verificam suas implementações somente por meio de testes de interoperabilidade.

Os primeiros a implementarem a solução SNMP foram os fabricantes de *gateways*, *bridges*, *switches* e roteadores. Atualmente esta solução implementa vários dispositivos como *no-breaks*, sistemas de automação, como CLPs e sistemas de telemetria.

9.1 Protocolo de Gerenciamento SNMP

Uma interligação de redes heterogêneas necessita de um sistema de gerenciamento que possibilite aos administradores detectar problemas, controlar o tráfego de informações e localizar componentes que estejam violando os padrões e parâmetros estabelecidos. Estas atividades são fundamentais no gerenciamento, além de permitir a troca de informações e dados entre os dispositivos que compõe a rede.

O protocolo SNMP atende às necessidades de gerenciamento de redes baseadas no padrão TCP/IP e *Ethernet*. O protocolo realiza o gerenciamento utilizando associação com MIB (*Management Information Base*).

Muitas redes incluíam implementações de gerenciamento como parte de seus protocolos na estrutura de gerenciamento de redes. Ao contrário das redes remotas, uma interligação em redes TCP/IP não possui um único protocolo no nível de enlace. Em vez disto, a interligação em redes consiste em várias redes físicas interconectadas por meio de roteadores IP, onde os roteadores formam os comutadores ativos que os administradores precisam analisar e controlar. Como os roteadores conectam-se à redes heterogêneas, os protocolos para gerenciamento da interligação em redes operam em nível de aplicação e para sua comunicação utilizam os protocolos de nível da camada de transporte TCP/IP.

Conforme expõe [CARVALHO-93] existem três tipos de gerenciamento: gerenciamento de sistemas, gerenciamento de camada e operação da camada. Por

meio da utilização de protocolos de gerenciamento de sistemas da camada de aplicação que se realiza o gerenciamento de sistemas.

A atuação de gerenciamento com *software* aplicados ao nível de camada de aplicação apresenta algumas vantagens. Os protocolos podem ser desenvolvidos sem levar em consideração o *hardware* básico da rede, uma pilha de protocolos pode ser utilizada para todas as redes. Ainda, os protocolos podem ser projetados sem considerar o *hardware* do equipamento gerenciado, eles podem ser utilizados para todos os dispositivos gerenciados. [TOVAR-96]

O protocolo de gerenciamento de redes SNMP foi lançado em 1988 para atender a necessidade cada vez maior de um padrão para gerenciar os dispositivos IP. O SNMP provê ao usuário um conjunto de operações simples que possibilitam o gerenciamento remoto destes dispositivos implementados com arquiteturas TCP/IP e *Ethernet*.

Inicialmente, o SNMP esteve associado ao gerenciamento de roteadores, *switches*, *racks* de modem, entre outros equipamentos, entretanto, a arquitetura do protocolo permite que possa ser utilizado para gerenciar vários tipos de dispositivos, como sistemas de energia, *no-breaks* e controladores lógicos programáveis. A figura 75 ilustra a aplicação do SNMP.

Figura 75: Aplicação do protocolo SNMP

Fonte: [LOPES-02]

O SNMP difundiu-se e popularizou-se nas décadas de 1980 e 1990, essa popularização conduziu a reavaliação de suas deficiências, principalmente ligadas à falta de implementações de mecanismos de segurança, autenticação e privacidade; as implementações levaram ao desenvolvimento do protocolo com a versão 3 do SNMP.

O SNMP é um protocolo destinado ao gerenciamento de redes, que permite de forma simples, em tempo real conhecer o *status* da rede e dispositivos conectados.

A arquitetura inicial SNMP derivou do SGMP (*Simple Gateway Management Protocol*), protocolo desenvolvido para gerenciar roteadores e equipamentos de interconexão de uma rede TCP/IP. A primeira versão do SNMP apresentava informações para gerenciamento e controle desses dispositivos de interconexão, o que determinou as escolhas dos elementos componentes da arquitetura SNMP.

O protocolo está implementado na camada de aplicação da arquitetura TCP/IP, e utiliza o UDP (definido na RFC 768) como protocolo de transporte na comunicação entre cliente e servidor. O SNMP é um protocolo não orientado à conexão, uma vez que utiliza o TCP em sua arquitetura, que por não ter nenhuma conexão ponto-a-ponto estabelecida entre agentes e gerente da rede.

Conforme expõe [SCHMIDT-01], o protocolo utiliza a porta 161 do UDP para enviar e receber solicitações e a porta 162 para receber *traps* (mensagens de alarmes/eventos) de dispositivos gerenciados.

Todo dispositivo que implementa o SNMP deve utilizar estes números como portas *default*, porém alguns fornecedores permitem modificar as portas na configuração do agente. Quase todos os fabricantes de computadores, estações de trabalho, roteadores, *hubs*, etc., oferecem a solução SNMP como ferramenta de gerenciamento.

A tabela 17 a seguir apresenta a lista de RFCs que especificam o SNMPv1.

Tabela 17: RFCs relacionadas com SNMPv1

RFC	Data	Título
1155	Mai/90	<i>Structure and Identification of Management Information (SMI) for TCP/IP based Internet</i>
1157	Mai/90	<i>A Simple Network Management Protocol (SNMP)</i>
1212	Mar/91	<i>Concise MIB Definitions</i>
1213	Mar/91	<i>Management Information Base for Network Management of TCP/IP based Internet: MIB II</i>
1643	Jul/94	<i>Definition of Managed Objects for the Ethernet-like Interface Types</i>

Fonte: Adaptado de [SCHMIDT-01]

O funcionamento do protocolo SNMP basicamente esta relacionado com a troca de informações entre 02 dispositivos da rede, ou seja, o gerente (*software* implementado na estação de gerenciamento) e o agente (*software* implementado em estações gerenciadas).

A figura 76 ilustra o modelo de troca de informações entre estas entidades

Figura 76: Troca de informações SNMP

Fonte: [STALLINGS-98]

O SNMPv1 utiliza o conceito de comunidades para definir uma confiabilidade entre gerenciadores e agentes. Um gerente (*manager*) pode ser descrito como um programa executado em uma estação servidora que possibilita a obtenção e envio

de informações de gerenciamento junto aos dispositivos gerenciados. O agente é um programa executado na estação gerenciada que é responsável pela manutenção das informações da estação gerenciada. Um agente é configurado com três nomes de comunidade: *Read-only*, *Read-write* e *Trap*.

A comunidade *read-only* permite ler os valores de dados sem modificá-los. A comunidade *read-write* possibilita ler e modificar o valor de dados, sendo também possível ler contadores, e redefinir seus valores; a comunidade *trap* possibilita o recebimento de notificações assíncronas dos agentes.

Ao contrário de outras soluções de gerenciamento, o protocolo SNMP apresenta um conjunto resumido de comando baseado no conceito de busca/alteração. Este conceito apresenta basicamente duas operações, uma que permite ao usuário alterar atributos de um objeto de uma MIB (*Operação Set*) e outra para obter os valores dos atributos de um objeto (*Operação Get*). Existem também as operações *Trap* e *Get-Next*. A operação *Get-Next* é utilizada para ler o valor da próxima variável, o gerente fornece o nome de uma variável e o cliente obtém o valor e nome da próxima variável; a operação *Trap* é utilizada para comunicação de um evento previamente definido.

A figura 77 a seguir demonstra o formato das mensagens no protocolo SNMP.

Apesar da boa aceitação, o protocolo SNMPv1 apresentava deficiências relacionadas à segurança, visto que as mensagens não eram criptografadas e não utilizavam qualquer tipo de autenticação, além disto, o SNMP não se enquadrava para o gerenciamento de grandes redes de computadores devido às limitações de desempenho para obtenção de requisições explícitas e por falta de suporte para troca de informações gerente-gerente.

Em função da necessidade de implementações em segurança, foi desenvolvido em 1993, a versão 2 do protocolo, o SNMPv2, com vários avanços, tais como:

- Estrutura de informação;
- Primitivas de comunicação;
- Comunicação gerente-gerente;
- Implementações de segurança.

Version	Community		SNMP PDU			
SNMP Message						
PDU type	request-id	0	0	variablebindings		
GetRequest PDU, GetNextRequest PDU and SetRequest PDU						
PDU type	request-id	error-status	error-index	variablebindings		
GetResponse PDU						
PDU type	enterprise	agent-addr	generic-trap	specific-trap	time-stamp	variablebindings
Trap PDU						
name 1	value 1	name 2	value 2		name n	value n
Variablebindings						

Campo	Descrição
Version	Versão do SNMP, por exemplo, RFC 1157 versão 1
PDU	Protocol Data Units (unidade de dados de protocolo)
Community	Nome da comunidade de modo a identificar o gerente para o agente
Request-Id	Utilizado para identificar cada mensagem de request
Error-status	Utilizado para indicar que ocorreu alguma exceção no processamento da solicitação
Error-index	Quando o Error-Status não é zero, este campo fornece informações como índice da variável que causou a exceção.
Variablebindings	Lista de nomes de variáveis e valores correspondentes.
Enterprise	Tipo de objeto que gerou a trap
Agent-addr	Endereço do agente que gerou o trap
Generic-trap	Tipo de trap genérica, informa os tipos de valores padrões definidos para traps, ou valor specific.
Specific-trap	Código de trap específica, no caso do valor do campo generic-trap ser specific
Time-stamp	Contém o valor do sysUpTime, ou seja, tempo desde a última reinicialização da entidade de rede da geração do trap.

Figura 77: Formato de mensagens SNMP

Fonte: Adaptado de [SCHMIDT-01]

A versão 2 do SNMP (SNMPv2) procurou corrigir deficiências da versão 1 (SNMPv.1). Ele basicamente surgiu da evolução do SNMPv1 e do RMON (*Remote Monitoring*). Utiliza a SMI 2 (*Structure of Management Information*), que permite a presença de novos tipos de ASN.1 (*Abstract Syntax Notation*)¹.

¹ Meio de especificar o modo como os dados são representados e transmitidos entre gerenciadores e agentes no contexto SNMP [SCHMIDT-05]

Além disto, possibilita a criação e exclusão de objetos, juntamente com a comunicação entre gerentes (*Manager to Manager MIB*).

A tabela 18 demonstra as RFCs que foram relacionadas com a versão 2 do protocolo SNMP.

Tabela 18: RFCs relacionadas com SNMPv2

RFC	Data	Título
1441	1993	<i>Introduction to SNMPv2</i>
1442	1993	<i>SMI for SNMPv2</i>
1443	1993	<i>Textual conventions for SNMPv2</i>
1444	1993	<i>Conformance Statements for SNMPv2</i>
1445	1993	<i>Administrative model for SNMPv2</i>
1446	1993	<i>Security Protocols for SNMPv2</i>
1447	1993	<i>Party MIB for SNMPv2</i>
1448	1993	<i>Protocol Operations for SNMPv2</i>
1449	1993	<i>Transport mappings for SNMPv2</i>
1450	1993	<i>MIB for SNMPv2</i>
1451	1993	<i>Manager to manager MIB</i>
1452	1993	<i>Coexistence Between SNMPv1 and SNMPv2</i>

Fonte: Adaptado de [SCHMIDT-05] e [STALLINGS-99]

A edição realizada em 1993 para o SNMPv2 incluiu várias implementações de segurança, porém estas não foram aceitas extensamente por causa de falta do consenso entre os grupos responsáveis pela análise do SNMP. Uma edição revisada de SNMPv2 foi emitida dentro 1996, com os realces funcionais e operacionais, mas sem as facilidades de segurança implementadas da versão anterior.

A tabela 19 a seguir relaciona as RFCs vinculadas à versão SNMPv2c.

Tabela 19: RFCs relacionadas com SNMPv2c

RFC	Data	Título
1901	Jan/96	<i>Introduction to Community-based SNMPv2</i>
1902	Jan/96	<i>Structure of Management Information for SNMPv2</i>
1903	Jan/96	<i>Textual Conventions for SNMPv2</i>
1904	Jan/96	<i>Conformance Statements for SNMPv2</i>
1905	Jan/96	<i>Protocol Operations for SNMPv2</i>
1906	Jan/96	<i>Transport Mappings for SNMPv2</i>
1907	Jan/96	<i>Management Information Base for SNMPv2</i>
1908	Jan/96	<i>Internet-Standard Network Management Framework</i>

Fonte: Adaptado de [SCHMIDT-05] e [STALLINGS-99]

Para compensar a falta de implementações e corrigir problemas relacionados com a segurança na versão SNMPv2c, foram elaboradas as versões SNMPv2u e SNMPv2*. As mensagens e a segurança foram melhor desenvolvidas e implementadas nestas versões (denominadas SNMPv2u e SNMPv2*), também foi implementada a segurança feita por usuário (*User-Based*), o que só permite a realização de operações por usuários específicos, impedindo o acesso de qualquer usuário. Estas duas propostas foram unificadas, dando origem a versão 3 do protocolo em 1998, o SNMPv3.

A tabela 20 ilustra as RFCs relacionadas com a versão 3 do SNMP.

Tabela 20: RFCs relacionadas com SNMPv3

RFC	Data	Título
2271	Jan/98	<i>Architecture for Describing SNMP Management Frameworks</i>
2272	Jan/98	<i>Message Processing and Dispatching for SNMP</i>
2273	Jan/98	<i>SNMPv3 Applications</i>
2274	Jan/98	<i>User Based Security Model for SNMPv3</i>
2275	Ago/98	<i>View-Based Access Control Model (VACM) for SNMPv3</i>

Fonte: Adaptado de [SCHMIDT-05] e [STALLINGS-99]

O desenvolvimento da versão 3 do SNMP (SNMPv3) trouxe aspectos relevantes relacionados à segurança, que implementa mecanismos para evitar a alteração das mensagens enviadas; além disto, barra-se o acesso a elementos estranhos à execução de operações de controle, que são realizadas por meio de uma operação *SetRequest*. Evita-se também a leitura das mensagens por parte de elementos não autorizados, além de se assegurar ao gerente o direito de alteração da senha dos agentes.

A segurança é conseguida por meio da implementação de mecanismos de criptografia com o DES (*Data Encryption Standard*) e de algoritmos de autenticação que podem ser tanto o MD5 quanto o SHA (*Secure Hash Algorithm*); posteriormente, as técnicas de criptografia foram atualizadas com utilização de AES (*Advanced Encryption Standard – RFC 3826*). [STALLINGS-08]. Portanto, as implementações definidas para SNMPv3 estão relacionadas a:

- Autenticação;
- Privacidade;
- Controle de acesso.

A versão 3 do protocolo SNMP encapsula as PDUs (*Protocol Data Unit*) das versões anteriores, ou seja, SNMPv1 e SNMPv2, a figura 78 abaixo descreve o relacionamento entre as diferentes versões do SNMP por meio dos formatos elaborados.

Figura 78: Relacionamento entre versões do SNMP

Fonte: Adaptado de [STALLINGS-99]

9.2 MIB (Management Information Base)

No gerenciamento em uma rede baseado na pilha de protocolos TCP/IP, as MIBs que podem ser consideradas como bancos de dados de objetos gerenciados e são utilizados para obter informações de servidores e estações SNMP. Os dados são obtidos por requisições de estações gerente a um ou mais estações agentes da rede utilizando os serviços do protocolo de transporte UDP para enviar e receber suas mensagens pela rede.

Dentre as variáveis que podem ser requisitadas, as MIBs, são a base de informações de gerenciamento. Todo tipo de informação sobre o status ou estatística é acessado pela NMS (*Network Management Stations*). Uma NMS é responsável pela operação de *polling* e por receber *traps* de agentes na rede, esta definida em uma MIB. [SCHMIDT-05].

Conforme expõe [SCHMIDT-01], um agente pode implementar várias MIBs, porém todos os agentes implementam uma MIB específica, a MIB II, definida na RFC 1213, onde o principal objetivo desta MIB é fornecer informações gerais sobre o gerenciamento de dispositivos em rede TCP/IP. Vários padrões de MIBs foram desenvolvidos para redes, como ATM (MIB - RFC 2515), *Frame Relay* (DTE *Interface Type* MIB RFC 2115), DNS (*Server MIB* RFC 1611) entre outros.

O gerenciamento de rede utilizando o SNMP permite que seja realizado o acompanhamento simples e fácil do estado em tempo real, da rede, podendo ser utilizado para gerenciar diferentes tipos de sistemas.

Cada estação gerenciada pelo SNMP deve possuir um agente e uma base de informações MIB. Conforme expõe [SCHMIDT-05], um objeto gerenciado é a visão abstrata de um recurso real do sistema; desta forma, todos os recursos da rede que devem ser gerenciados são modelados e as estruturas dos dados resultantes são os objetos gerenciados.

Os objetos gerenciados podem ter permissões para serem lidos ou alterados, sendo que cada leitura representará o estado real do recurso e, cada alteração também será refletida no próprio recurso. Dessa forma, a MIB é o conjunto dos objetos gerenciados, que procura abranger todas as informações necessárias para a gerência da rede.

A RFC 1066 apresentou a primeira versão da MIB, a MIB I. Este padrão definiu a base de informação necessária para monitorar e controlar redes baseadas na pilha de protocolos TCP/IP. A evolução aconteceu com o RFC 1213 que propôs uma segunda MIB, a MIB II, para uso baseado na pilha de protocolos TCP/IP. Basicamente são definidos três tipos de MIBs: MIB II, MIB experimental, MIB privada.

A MIB II, é considerada uma evolução da MIB I, fornece informações gerais de gerenciamento sobre um determinado equipamento gerenciado. Através das MIB II podem ser obtidas informações como número de pacotes transmitidos, estado da *interface*, entre outras.

A MIB experimental é definida em virtude de seus componentes (objetos) estão em fase de desenvolvimento e teste, em geral, eles fornecem características mais específicas sobre a tecnologia dos meios de transmissão e equipamentos utilizados.

MIB privada é aquela em que seus componentes fornecem informações específicas dos equipamentos gerenciados, como configuração, colisões e também é possível reinicializar, desabilitar uma ou mais portas de um roteador.

As regras de construção das estruturas da MIB são descritas através da SMI (*Structure of Management Information*).

A SMI define com exatidão como os objetos gerenciados são nomeados e especifica os respectivos tipos de dados associados. Os objetos são organizados em uma hierarquia em árvore e são reconhecidos por um OID (*Object Identifier*). Essa estrutura forma a base do esquema de atribuição de nomes do SNMP. Um OID de objeto é formado por uma seqüência de inteiros baseada nos nós das árvores, separada por pontos (.).

A figura 79 a seguir ilustra a subárvore MIB-II.

Nome da subárvore	OID	Descrição
system	1.3.6.1.2.1.1	Define uma lista de objetivos pertencentes à operação do sistema
interfaces	1.3.6.1.2.1.2	Rastreia o status de cada interface em uma entidade gerenciada
at	1.3.6.1.2.1.3	Mantém compatibilidade com versões anteriores
ip	1.3.6.1.2.1.4	Rastreia os aspectos do IP, incluindo o roteamento IP
icmp	1.3.6.1.2.1.5	Rastreia aspectos como erros do ICMP, exclusões, etc.
tcp	1.3.6.1.2.1.6	Rastreia o estado das conexões do TCP
udp	1.3.6.1.2.1.7	Rastreia dados estatísticos do UDP, datagramas internos e externos
egp	1.3.6.1.2.1.8	Rastreia dados estatísticos do EGP e mantém uma tabela de vizinhos do EGP
transmission	1.3.6.1.2.1.10	Não existem objetos definidos para este grupo
snmp	1.3.6.1.2.1.11	Avalia o desempenho da implementação básica do SNMP na entidade gerenciada e rastreia pacotes enviados e recebidos

Figura 79: Subárvore MIB II:

Fonte: [SCHMIDT-01].

Conforme cita [OLIFER-08] a especificação RMON MIB melhora a capacidade de comunicação remota com a MIB e implementa acréscimo aos recursos funcionais do SNMP, o banco de dados RMOM MIB contém informações agregadas sobre o dispositivo e que não requer que grandes volumes de informações trafeguem na rede.

Foram especificados dois padrões para RMON: RMON 1 e RMON2. O RMON 1 atua na camada MAC (*Media Access Control*) e o RMON 2 opera na camada de rede e em camadas superiores (RFCs 1513, 1757, 2021 e 2074). [STALLINGS-99]

A RMOM MIB possui independência do protocolo da camada de rede, diferente dos padrões MIB I e MIB II, sendo conveniente para redes heterogêneas que utilizam vários protocolos na camada de rede, finaliza [OLIFER-08].

A figura 80 abaixo apresenta resumo comparativo entre MIBs.

	MIB II	HOST MIB	BRIDGE MIB	HUB MIB	RMON MIB
Interface statistics					
IP, TCP, UDP statistics					
SNMP statistics					
Host job counts					
Host file system information					
Spanning-tree performance					
Wide-area link performance					
Link testing					
Network traffic statistics					
Host table of all addresses					
Hosts statistics					
Historical data					
Alarm thresholds					
Configurable statistics					
Traffic matrix with all nodes					
"Host top N" tables					
Packet capture / protocol analysis					
Distributed logging of events					

Figura 80: Resumo comparativo MIBs

Fonte: [GABOS-09]

Devido à versatilidade do protocolo de gerenciamento SNMP que passou a ser integrado em diversas plataformas com a utilização de APIs. Conforme expõe [MELLQUIST-94], o desenvolvimento destas APIs requerem a utilização de biblioteca de funções com as quais o programador deve estar familiarizado, onde estas APIs definem uma plataforma específica, resultando em um código específico não oferecendo portabilidade para a solução.

O SNMP passou a dispor do SNMP++ que provê programação orientada à objeto com um conjunto de classes de objetos C++ e bibliotecas de classes reutilizáveis. O protocolo utiliza os mecanismos existentes em outras APIs aliadas as vantagens da programação orientada à objetos para aplicações de gerenciamento SNMP.

Conforme expõe [MELLQUIST-94] esta solução provê:

- Facilidade de utilização;
- Portabilidade;
- Extensibilidade;
- Segurança.

Utilizando o conceito de programação orientada à objeto em soluções de gerenciamento e inspirado no SNMP++ foi desenvolvido o SNMP4J API *Java*¹.

¹ www.snmp4j.org/

10. MODELO DE GESTÃO CORPORATIVA

Segundo expõe [BIO-08], que define sistema de gestão “como conjunto, interdependente, dos processos decisório-gerenciais (planejar, organizar, controlar, etc.) que visa levar a empresa aos resultados desejados. O sistema de informação, por meio das informações gerenciais, interage com o sistema de gestão, ao suportar os processos decisórios por meio de tais informações.”

Ainda conforme [BIO-08], as empresas definem modelos de gestão ao optarem por metodologias administrativas que podem variar de um modelo único (aplicado comumente às empresas de grande porte e multinacionais) ou mesmo pela competência individual dos dirigentes das empresas. Basicamente, os componentes que compõem o Modelo de Gestão Corporativa são:

- Filosofia empresarial;
- Visão dos dirigentes sobre os resultados empresariais;
- Estrutura organizacional;
- Processo gerencial de planejamento/direção/controle;
- Informações gerenciais geradas pelo sistema de informação.

A figura 81 abaixo apresenta o inter-relacionamento entre estes componentes.

Figura 81: Elementos componentes de processo de gestão

Fonte: [BIO-08]

Os Sistemas de Informação Gerencial constituem o elo entre o Processo Gerencial além de subsidiar diretamente os resultados empresariais. Conforme aborda [BIO-08], os Sistemas de Informação envolvem: informações mercadológicas, informações de produção e informações econômico-financeiras.

Segundo expõe [LAUDON-98], sistemas de informação “podem ser definidos tecnicamente como um conjunto de componentes inter-relacionados que coletam (ou recuperam), processam, armazenam e distribuem informação com a finalidade de dar suporte à tomada e controle em uma organização. Os sistemas de informação podem auxiliar gerentes e trabalhadores a analisar problemas...”.

O crescimento de uma empresa, conseqüentemente acarretará no aumento de volume de informações que devem ser sistematizadas e distribuídas pela organização, facilitando o compartilhamento e processamento.

Segundo expõe [CAIÇARA-08], “um Sistema de Informação é parte integrante da empresa e é um produto de três componentes: Tecnologia, Organização e Pessoas.” Não se pode entender ou usar SI em empresas de forma eficiente sem o conhecimento de suas dimensões em termos de organização e de pessoas, assim como suas dimensões tecnológicas.

Os sistemas de informação (SI) podem ser classificados de várias formas, dentre as maneiras comumente utilizadas, tem-se a classificação de acordo com a área funcional e de acordo com o nível hierárquico. [CAIÇARA-08].

A classificação em função da área funcional compreende sistemas de informação: financeira, contábil (controladoria), industrial (produção), *marketing* (vendas) e de recursos humanos. Outra forma de classificação são os níveis hierárquicos, que abrange e define os níveis: operacional, tático e estratégico. (LAUDON-98]

A figura 82 a seguir ilustra a distribuição dos Sistemas de Informação e respectivos níveis hierárquicos de uma empresa.

Legenda:

TPS - Transaction Processing System
 KWS - Knowledge Work Systems
 OAS - Office Automation Systems
 MIS - Management Information Systems
 DSS - Decision Support Systems
 ESS - Executive Support Systems

Figura 82: Distribuição dos Sistemas de Informação nos níveis hierárquicos

Fonte: Adaptado de [LAUDON-98]

Segundo expõe [CAIÇARA-08], “grande parte das organizações são orientadas por funções, ou seja, cada processo empresarial é suportado por um sistema.”

Esta separação entre processos leva ao desenvolvimento de soluções de gestão baseadas em sistemas de informação que não possibilitem a comunicação entre segmentos da empresa.

Ainda conforme [CAIÇARA-08], “a integração dos sistemas de informação acaba com as barreiras existentes entre os próprios departamentos e entre as sedes e os departamentos, e reduz a duplicação de esforços.”

Segundo [CUMMINS-02], cita que a automação do processo de negócio, definido por gerenciamento de fluxo de trabalho, “dinamiza as comunicações entre os sistemas e as pessoas.”

10.1 ERP (Enterprise Resource Planning)

Conforme expõe [MARANGONI-08], sistemas ERPs (*Enterprise Resource Planning*) podem ser definidos como:

Um conjunto integrado de *softwares* que proporcionam o gerenciamento e a visão completa das empresas, facilitando a integração de áreas que, mesmo sendo alvos de tratamentos distintos, podem trabalhar em conjunto, gerando uma maior sinergia à empresa, como finanças, recursos humanos, compras e vendas.

O ERP atua além do planejamento, provém controle e suporte a todos os segmentos operacionais, administrativos e comerciais de corporações. Possibilita a integração com todos os segmentos da empresa oferecendo um Sistema Integrado de Informação que se comunicam e utilizam a mesma base de dados.

A necessidade de processos eficientes de solicitação e gestão de materiais na indústria deu origem na década de 1969 ao *Material Requeriment Planning* (MRP). Os sistemas MRP eram "pacotes que interagiam entre si" e, desta forma possibilitavam o planejamento da utilização de insumos e a administração das diversas etapas dos processos produtivos. [MARANGONI-08]

Esta técnica mostrou-se eficiente na gestão de inventários, conforme expõe [MIÃO-07], quando na década de 1980, ocorreu a aprimoramento deste sistema e o surgimento do MRP II.

O desenvolvimento acentuado das redes de computadores a partir da década de 1980 tornou evidente a necessidade de sistemas mais abrangentes. A partir da década de 1990, surgem os sistemas ERP com a finalidade de integração entre os diversos setores corporativos.

A arquitetura de um sistema ERP, conforme define [DAVENPORT-98] é constituído utilizando-se arquitetura cliente-servidor.

A figura 83 a seguir ilustra estrutura típica de sistema ERP.

Figura 83: Estrutura típica de sistema ERP

Fonte: Adaptado de [DAVENPORT-98]

Conforme aborda [MARANGONI-08], com o acelerado crescimento da internet, a evolução do ERP tradicional foi o ERP II.

Devido ao rápido crescimento na demanda por soluções de integração, a partir da década de 1990, desenvolvedores e fornecedores focaram esforços em sistemas ERP voltados à Internet, fazendo com que os módulos pudessem ser atualizados pela *Web*, e que usuários remotos conseguissem acessar e suprir o sistema. [MARANGONI-08].

As principais características de sistemas ERP são: [SOUZA-00]

- Pacotes comerciais de *software*;
- Desenvolvidos a partir de modelos-padrão de processos;
- Sistemas ERP são integrados;
- Possuem grande abrangência funcional;
- Utilizam um banco de dados corporativo;
- Requerem procedimentos de ajuste.

O ERP II provê a integração além dos sistemas corporativos, enfatizou a colaboração comercial com a internet, definindo o *e-commerce*. Além de desenvolver

produtos e formas de comercialização específicas, o ERP II possibilita o incremento do fluxo de informações entre corporações, interligando sistemas e módulos ERPs.

Segundo [MIÃO-07], a distribuição entre os fornecedores de sistemas ERP, com base em dados do IPM¹ (Impacta Pesquisa Periódica de Mercado – abril/2005), apresenta a distribuição entre as plataformas de mercado, conforme ilustra a figura 84 abaixo.

Legenda:

SAP - Systemanalyse and Programmentwicklung ²

Figura 84: Distribuição de fornecedores de sistemas ERP

Fonte: (IPM¹ – Impacta Pesquisa Periódica de Mercado, 2005)

Conforme aborda [MIÃO-07], o processo de planejamento para implementação de sistemas ERP provê etapas de planejamento prévio, para que a implementação seja bem sucedida. Estes processos estão ilustrados na figura 85 a seguir e são compreendidos de:

- Levantamento das necessidades de informação;
- Definição das informações a serem extraídas do sistema;
- Entendimento da abrangência dos vários módulos do sistema;

¹ IPM - Impacta Pesquisa Periódica de Mercado, mantém informações sobre as pesquisas que realiza a respeito do Mercado de Tecnologia da Informação no Brasil. [<http://www.impacta.com.br/a-impacta/pesquisa-mercado.php>]

² A SAP foi fundada na Alemanha em 1972, desenvolveu o conceito original de ERP com o produto R/2. [CAIÇARA-08]

Figura 85: Fases de planejamento de sistemas ERP

Fonte: Adaptado de [MIÃO-07]

- Identificação das simplificações e eliminações dos vários módulos do sistema;
- Definição da seqüência de implantantação, resultados e prazos;
- Estabelecimento de responsabilidades;
- Treinamento;
- Preocupação com relação ao *Middleware*.

[MIÃO-07] cita a importância do *middleware* como elemento essencial na integração de sistemas de gestão ERP.

10.2 MES (Manufacturing Execution System)

Conforme cita [SHIRASUNA-08], o conceito de *Manufacturing Execution System* (MES), foi inserido em 1992 pela empresa *AMR Research Inc*¹.

Um MES coleta e reúne informações dos processos de chão-de-fabrica e as realimenta para o sistema de planejamento localizado na camada ERP/MRP II. Segundo a *AMR*, “o MES representa a camada de execução que existe entre a área corporativa e o sistema de controle que provê funcionalidades de controle e visibilidade.”

Segundo aborda [CORREA-08], a *Manufacturing Execution System Association International*² (*MESA International*) designa o sistema MES como “sistema de chão-de-fabrica orientado para a melhoria de desempenho que complementa e aperfeiçoa os sistemas integrados de gestão (planejamento e controle) da produção.”

O MES foi desenvolvido para servir de elo entre o sistema de planejamento e a fábrica, ainda conforme expõe [CORREA-08], o MES/SFC (*Shop Floor Control*) cumpre dois papéis: controle da produção, comparando o que efetivamente foi produzido com o planejamento, e libera ordens de produção.

As principais funcionalidades do MES, conforme modelo elaborado pela *MESA International* são: [SHIRASUNA-08]

- Alocação de recursos e *status*;
- Operações/Programação Detalhada;
- Unidades de Produção e Despacho;
- Controle de Documento;
- Coleta de Dados/Aquisição;
- Gerenciamento de mão-de-obra;

¹ *AMR Research Inc.* – empresa de consultoria com foco em *supply chain* (cadeia de suprimentos) [www.amrresearch.com]

² *MESA INTERNACIONAL (Manufacturing Execution System Association International)* – grupo de empresas desenvolvedoras e fornecedoras de sistemas MES [www.mesa.org]

- Gerenciamento de Qualidade;
- Gerenciamento de Processo;
- Gerenciamento de Manutenção;
- Rastreabilidade de Produto e Genealogia;
- Análise de Performance.

O MES está especificado pela norma ISA S95, que define cinco níveis nas empresas industriais. A figura 86 abaixo ilustra os níveis propostos na normatização. [SHIRASUNA-08].

Figura 86: Níveis propostos para normatização das empresas industriais – MES

Fonte: Adaptado de [SHIRASUNA-08]

Apesar de ocorrerem desenvolvimentos com foco na integração entre processos administrativos e produtivos, a falta de elementos padronizados, o surgimento de novos projetos e o desenvolvimento de sistemas inadequados levam a falta de integração entre as estruturas de gestão. Conforme expõe [NEVES-08], que cita que “na busca pelo aumento da eficiência dos processos produtivos, o setor industrial realizou nos últimos anos grandes investimentos na automação e informatização do chão-de-fábrica.”

Ainda, conforme complementa [NEVES-08], “se de um lado as máquinas *inteligentes* passam a dominar a manufatura, as outras áreas da produção e da empresa também investem na modernização, automação e implantação de TI; no entanto, a implantação geralmente se dá de forma desordenada, localizada e com TIs que nem sempre *conversam* entre si...”

Esta falta de integração entre processos (gestão administrativa e processos produtivos) estabelece uma lacuna que leva a conseqüente falta de eficiência na gestão da Informação.

A figura 87 abaixo ilustra lacuna entre processos de gestão.

Figura 87: Pirâmide de automação sem integração

Fonte: Adaptado de [FERNANDES-06]

O MES tem a função de realizar o elo entre a gestão produtiva e os sistemas ERP, provendo arquitetura integrada e eficiente. Entretanto, conforme cita [SHIRASUNA-08] existe uma participação reduzida de sistemas MES nos processos de integração, e complementa mencionando que esta redução deve-se à:

- Nível de automação das empresas;
- Desconhecimento do MES;
- Barreiras culturais entre engenharia e TI;
- Custo de investimento em MES;
- Complexidade para implantar o MES.

[SHIRASUNA-08], finaliza citando os inúmeros benefícios obtidos com a implantação de sistemas MES em plantas indústrias, e salienta que além dos fatores citados, atribui a falta de convergência entre os departamentos de Engenharia/Manutenção e TI para a reduzida implantação destes sistemas.

Segundo aponta [CERRI-04], deixa evidente que “ os sistemas de informação ou tecnologia da informação não devem ser tratados isoladamente, isto é, sem que se busque atender aos negócios da organização.”

Conforme expõe [SEIXAS-09] cita que a norma ISA-95 define o termo MOM (*Manufacturing Operations Management*) em substituição a designação feita pela empresa ARM e expõe que a abrangência do MES/MOM possui atribuições que compreendem além da produção, operações de: produção, manutenção, qualidade e inventário.

Segundo [SEIXAS-09], deve ser elaborado “mapa de funcionalidades” da planta, analisando as interfaces do MES/MOM com o ERP, e estabelece que este seja um dos parâmetros que deve ser considerado na adoção do sistema adequado a planta industrial.

Devido ao contexto atual de convergência de tecnologias no ambiente produtivo, prover a gestão adequada dos processos e operações fabris passou a ser discernido como ponto de fundamental relevância. Conforme aborda [PAVA-09], que as modernas soluções de Gerenciamento de Operações (MOM), devem ser capazes de atender:

- Escalabilidade – Uma abordagem em plataforma de solução;
- Flexibilidade – Convergência de TA e TI no contexto da manufatura;
- Especialização – soluções focadas por indústria;
- Inteligência Operacional – Enterprise Manufacturing Intelligence (EMI).

11. INTEGRAÇÃO EM APLICAÇÕES DISTRIBUÍDAS

As tecnologias de informação surgidas a partir de soluções desenvolvidas utilizando objetos distribuídos tem sido implementadas por diversas empresas possibilitando a rápida troca de dados e informações, aproximando filiais de empresas, criando alternativas cômodas para clientes e reduzindo custos de logística para clientes e fornecedores; integrando enfim, a cadeia produtiva, logística, administrativa, entre outros segmentos que compõem empresas.

Conforme expõe [CUMMINS-02], “a tecnologia de objetos distribuídos oferece suporte ao desenvolvimento de componentes distribuídos que interagem como objetos que trocam mensagens em uma rede. Os objetos podem ser serviços ou objetos compartilhados de uma aplicação de negócio.”

Os objetos podem ser distribuídos em uma série de computadores dispostos na rede onde a comunicação ocorre por meio de *middleware*, definido como requisitor de objetos, conforme complementa [SOMMERVILLE-07].

A figura 88 abaixo ilustra uma aplicação típica de objetos distribuídos.

Figura 88: Aplicação típica de objetos distribuídos

Fonte: Adaptado de [CUMMINS-02]

As aplicações distribuídas podem ser vistas atualmente no ambiente corporativo, em implementações tais como:

- Sistema de gestão via *web*;
- e-Commerce;
- e-Banking;
- e-Business
- Outros.

Nota-se, portanto, a expansão de atividades em um ambiente heterogêneo como a Internet. Segundo [CUMMINS-02], “o comércio eletrônico é basicamente a integração corporativa que se prolonga para clientes e parceiros de negócios.”

Ainda, conforme menciona [CUMMINS-02], na década de 1990, as aplicações de integração corporativas foram baseadas em produtos definidos por COSTS (*Commercial-Off-the-Shelf*), tais como sistemas *Data Warehouse*, *Data Mart* e *Data Mining*, porém, necessidades de adaptação e integração aliadas à expansão inerentes das corporações associadas à baixos custos tornou-se uma necessidade.

Conforme expõe [SOMMERVILLE-07], as vantagens do modelo de objetos distribuídos são:

- Possibilita que o projetista do sistema posponha decisões sobre onde e como os serviços devem ser fornecidos, ou seja, os objetos que fornecem serviços podem ser executados em qualquer nó da rede;
- Sistema de arquitetura aberta, que permite que novos recursos sejam adicionados quando necessário;
- Sistema flexível e escalonável;
- Possibilita reconfigurar dinamicamente o sistema com objetos que migram através da rede, conforme necessário.

Os sistemas computacionais distribuídos possuem como características serem heterogêneos e escalonáveis, onde a escalabilidade pode ocorrer de diversas formas por meio de várias plataformas de computação. Conforme expõe [CUMMINS-02], “com padrões de interoperabilidade apropriados e o *middleware*, as aplicações podem interoperar, mesmo que estejam sendo executadas em diferentes plataformas, usando diferentes gerenciadores de banco de dados e escritos em diferentes linguagens.”

Os sistemas de objeto distribuído são a base de arquiteturas de três camadas, na qual a lógica de apresentação ou primeira camada reside em um cliente, a lógica central comercial reside em uma camada central e o banco de dados (*back end*) reside na terceira camada. A tecnologia de objetos distribuídos estende a camada central, permitindo a acessibilidade para vários objetos de aplicativo, em vez de somente apenas um.

O *middleware* desempenha papel essencial para assegurar a comunicação contínua de objetos, em uma arquitetura de objetos distribuídos, onde os objetos podem ser implementados em linguagens de programação diversas. Conforme complementa [SOMMERVILLE-07], o *middleware* atua em dois níveis para apoiar a comunicação, sendo:

- Nível de comunicação lógica, o *middleware* fornece a funcionalidade que possibilita aos objetos de diferentes computadores trocarem informações de dados e de controle;
- Nível de componente, o *middleware* prove uma base para o desenvolvimento de componentes compatíveis, possibilita implementações com métodos padronizados.

Conforme menciona [CUMMINS-02] há três principais tecnologias de objetos distribuídos:

- A CORBA (Common Object Request Broker Architecture);
- O COM+ (Component Object Model);
- O EJB (Enterprise JavaBeans).

Conforme expõe [COULOURIS-07], existem atualmente vários padrões de *middleware* orientados à objetos, dentre os quais:

- O CORBA;
- RMI *Java*;
- Web Services;
- DCOM (Distributed Component Object Model).

Os protocolos de objeto distribuído são construídos na mesma arquitetura básica que tem por base uma camada de comunicação de rede que compreende três partes: [NIRVA-01]

- O servidor de objeto;
- O *skeleton*;
- O *stub*.

O servidor de objeto e o *skeleton* comumente residem na camada central. O *stub* reside na camada cliente e manipula a comunicação entre processos entre o objeto no lado do cliente e no lado do servidor; agindo como *proxy* no cliente e é responsável por comunicar pedidos do cliente para o servidor de objeto, por meio do *skeleton*.

O *stub* e o *skeleton* são responsáveis por fazer o servidor de objeto (que pode ficar na camada central ou na terceira camada) parecer como se a execução estivesse ocorrendo na estação do cliente.

Para encaminhar dados entre diferentes espaços de endereços, o *stub* e o *skeleton* utilizam dois processos, definidos por *marshaling*¹ e *unmarshaling*².

Conforme expõe [COULOURIS-07] os sistemas de objetos distribuídos podem adotar a arquitetura cliente/servidor; os objetos são gerenciados pelos servidores e seus clientes invocam seus métodos utilizando invocação a método remoto (RMI - *Remote Method Invocation*). A invocação é realizada pela execução de um método do objeto no servidor e o resultado é retornado para o cliente em outra mensagem.

Outros métodos e modelos de arquiteturas podem ser impostas à objetos distribuídos, sendo: [COULOURIS-07]

- Replicar os objetos, elevando as vantagens de tolerância à falhas;
- RMI concorrentes, a partir de objetos em diferentes computadores;

Cada processo contém um conjunto de objetos, alguns dos quais podem receber invocações remotas e locais. [COULOURIS-07] define as invocações à métodos entre objetos em diferentes processos, estejam estes na mesma estação ou não, como “invocações a métodos remotos” e finaliza mencionando que as invocações a métodos no mesmo processo são “invocações a métodos locais.”

¹ *marshaling* – empacota os parâmetros de uma chamada de método ou valores de retorno em um formato padrão para transmissão. [NIRVA-01]

² *unmarshaling* – operação inversa do *marshaling*, desempacota o formato padrão de uma apresentação de dados no espaço de endereço de um processo receptor. [NIRVA-01]

Tratar o estado compartilhado de um programa distribuído como um conjunto de objetos e que estes podem ser acessados via RMI ou mesmo serem copiados em um *cache* local trazem grandes vantagens para sistemas heterogêneos, visto que diferentes formatos de dados podem ser utilizados em instalações divergentes sem que estes formatos sejam notados pelos clientes que utilizam RMI para acessar estes objetos. [COULOURIS-07].

A figura 89 abaixo ilustra a invocação à métodos locais e remotos.

Figura 89: Invocação à métodos locais e remotos

Fonte: Adaptado de [COULOURIS-07]

11.1 Tecnologias em aplicações distribuídas

A arquitetura CORBA (*Common Object Request Broker Architecture*) foi desenvolvida na década de 1990 pela OMG (*Object Management Group*) baseado em um modelo de abstração definido por OMA (*Object Managemet Architecture*) que utiliza um ORB (*Object Request Broker*) para criar e gerenciar a comunicação entre objetos no lado do cliente e no lado do servidor; o acesso é realizado por meio do protocolo IIOP (*Internet Inter-ORB*), que torna possível que programas escritos em diferentes linguagens de programação se comuniquem através da Internet.

Os objetos e interfaces CORBA são especificados utilizando a OMG IDL (*OMG Interface Definition Language*), que possibilitam a operação mútua entre objetos no lado do cliente e no lado do servidor, escritos em diferentes linguagens de programação; conforme cita [BORSOI-04], a utilização do padrão CORBA possibilita ter aplicações completamente distribuídas com partes do *software* sendo executado em diferentes plataformas e com localização em rede.

O EJB (*Enterprise JavaBeans*) é um modelo de componente que permite aos desenvolvedores distribuírem componentes no servidor (servidores de aplicativos e

servidores de banco de dados). Em aplicativos EJB, a ativação remota segue a especificação RMI (*Remote Method Invocation*), porém os fornecedores não estão limitados ao protocolo de transporte RMI. Os componentes EJB são componentes *Java* que são executados em um servidor (servidor de aplicativos ou de banco de dados), que podem funcionar ou serem executados em qualquer ambiente que possua um interpretador *Java* (JVM – *Java Virtual Machine*) e um *contêiner* EJB. O *Enterprise JavaBeans* possibilita aos aplicativos se comunicarem em ambientes de cliente e servidor de várias camadas e através da Internet e de estruturas de intranet.

O RMI (*Remote Method Invocation*) *Java* foi desenvolvido pela empresa *Sun Microsystems* e possibilita que um objeto *Java* sendo executado em uma JVM (*Java Virtual Machine*) chame métodos presentes em outro objeto *Java* sendo executado em outra JVM. Quando ocorre a invocação de um método de um objeto remoto, o cliente RMI atua sobre um objeto local como se fosse um objeto remoto; este objeto local é chamado *stub* e age como um *proxy* do objeto remoto, provendo ao cliente a transparência na comunicação. O *stub* é gerado a partir de um compilador RMIC (*Remote Method Invocation Compiler*).

Conforme expõe [BORSOI-04], o RMI constitui uma interface que possibilita a intercomunicação entre objetos *Java* localizados em diferentes *hosts*; cada objeto remoto implementa uma interface remota que especifica quais métodos podem ser invocados remotamente pelos clientes, da mesma forma que é realizada a invocação remota.

O RMI utiliza o JRMP (*Java Remote Method Protocol*) para chamadas de ativação remotas e foi projetado para operar no ambiente *Java*. Em 1999, a *Sun* desenvolveu o RMI através da especificação IIOP (*RMI-IIOP*), que foi elaborado em conjunto com a IBM, possibilita que objetos *Java* se comuniquem com objetos CORBA.

O DCOM (*Distributed Component Object Model*) é uma arquitetura desenvolvida pela Microsoft® para aplicações distribuídas. O DCOM é uma extensão do COM que suporta a comunicação entre objetos em computadores distintos.

O COM+ (*Component Object Model*), possibilita que os clientes ativem serviços fornecidos por componentes compatíveis com COM. Os objetos e interfaces COM são especificados utilizando a IDL (*Interface Definition Language*) da empresa Microsoft® por meio de uma coleção de objetos ActiveX.

Os *WebServices* são aplicações de *software* identificadas por um URI (*Uniform Resource Identifier*) e estão enquadrados como qualquer tipo de serviço disponível na internet e que utiliza o XML (*Extensible Markup Language*) de forma independente de sistema operacional ou linguagem de programação.

O XML é utilizado para possibilitar que clientes se comuniquem com serviços *web* e para definir as interfaces e outras propriedades deste mesmo serviço. O XML possui a propriedade de ser extensível (as *tags* são definidas pelos usuários) em contraste com o HTML (*Hypertext Markup Language*), que possui conjunto de *tags* fixo.

Os *WebServices* utilizam em sua arquitetura protocolos abertos como HTTP (*Hypertext Transfer Protocol*) e SOAP (*Simple Object Access Protocol*). Para representação e estruturação das mensagens recebidas e enviadas é utilizado o XML; as operações incluindo os parâmetros de entrada e saída são codificadas no protocolo SOAP. Em *WebServices*, os serviços de operação, mensagens, parâmetros são descritos utilizando a linguagem baseada em XML definida por WSDL (*Webservice Description Language*).

11.2 Middleware

Segundo definição de [COULOURIS-07] “um *middleware* é composto por um conjunto de processos ou objetos, em um grupo de computadores, que interagem entre si de forma a implementar comunicação e oferecer suporte para compartilhamento de recursos distribuídos.”

A camada de *software* definida pelo *middleware* possibilita a comunicação entre aplicações distribuídas, tendo por objetivo diminuir a complexidade e heterogeneidade dos diversos sistemas existentes, provendo serviços que realizam a comunicação entre esta categoria de aplicações de forma transparente. [MACIEL-04]

A figura 90 abaixo ilustra a comunicação através de *middleware*.

Figura 90: *Middleware* para comunicação em sistema distribuído

Fonte: Adaptado de [SERIAN-02]

Conforme expõe [SERIAN-02], “*middleware* constitui uma estrutura de comunicação que é independente do sistema operacional e da natureza do sistema de transmissão.”

No modelo OSI que define os diferentes níveis de comunicação entre sistema de informação, *middleware* está localizado na parte superior e define os protocolos de comunicação entre aplicações.

A figura 91 ilustra este processo no Modelo de Referência OSI.

Figura 91: Localização de *Middleware* no Modelo OSI

Fonte: Adaptado de [SERIAN-02]

Na pilha de protocolos TCP/IP, a camada de abstração *middleware* interage com os protocolos de transporte UDP e TCP, viabilizando a comunicação entre aplicações e serviços.

A figura 92 a seguir ilustra esta camada na arquitetura TCP/IP.

Figura 92: *Middleware* – arquitetura TCP/IP

Fonte: Adaptado de [COULOURIS-07]

A interface de programa aplicativo para TCP fornece a abstração de um fluxo (*stream*) bidirecional entre pares de processos; “a informação transmitida consiste em um fluxo contínuo de dados sem dar a noção de limites de mensagem, os fluxos fornecem um bloco de construção para a comunicação produtor-consumidor”, conforme expõe [BACON-02]¹ apud [COULOURIS-07].

Um produtor e um consumidor constituem um par de processos, onde a função do produtor é produzir itens de dados para o consumidor.

Segundo ainda [COULOURIS-07], menciona que a interface de programa aplicativo para UDP fornece uma abstração de passagem de mensagem.

Definido na camada de *middleware* para comunicação TCP/IP existe os protocolos de requisição e resposta, elaborados para prover comunicação cliente/servidor nas formas de RMI (*Remote Method Invocation*) ou RPC (*Remote Procedure Call*).

¹ BACON, J. (2002). **Concurrent Systems**, terceira edição. Harlow, Inglaterra: Addison-Wesley.

Na comunicação com os protocolos de transporte TCP ou UDP, são utilizados a abstração de *soquete*¹, para a comunicação entre processos, conforme aborda [COULOURIS-07], que complementa citando que “a comunicação entre processos consiste em transmitir uma mensagem entre um soquete de um processo e um *soquete* de outro processo.”

A figura 93 abaixo ilustra este processo de comunicação.

Figura 93: Comunicação entre processos

Fonte: Adaptado de [COULOURIS-07]

Na comunicação por fluxo TCP, a API do protocolo TCP fornece a abstração de um fluxo de *bytes* no qual os dados podem ser lidos (*receive*) e escritos (*send*), sendo ocultas pela abstração de fluxo (*stream*) as seguintes características:

- Tamanho das mensagens;
- Mensagens perdidas;
- Controle de fluxo;
- Duplicação e ordenamento de mensagens;
- Destinos de mensagens.

No protocolo UDP, a comunicação é estabelecida com datagrama enviado pelo protocolo UDP transmitido de um processo remetente para um processo destino, sem necessidade de confirmações, que define a forma mais simples de comunicação entre processos, conforme expõe [COULOURIS-07].

¹ Soquete – API de comunicação para sistemas UNIX BSD, foram desenvolvidas em linguagem C, cujo objetivo é viabilizar interface na transmissão TCP e UDP. [SOARES-99]

As plataformas de *middleware* formam uma engrenagem vital para a construção de sistemas distribuídos robustos; tais plataformas definem camadas de abstração que facilitam o desenvolvimento de sistema de *software* de larga escala, aliviando a carga ao desenvolver aplicações que exigem um número complexo de serviços de infraestrutura.

Cabe salientar, conforme expõe [COULOURIS-07] que o desenvolvimento de uma *middleware* para suportar sistemas *peer-to-peer* em escala global torna-se um problema difícil, o conhecimento das localizações dos objetos deve ser particionado por toda a rede, onde cada nó se responsabiliza por manter o conhecimento de tais localizações e objetos, portanto, manter os requisitos de escalabilidade e disponibilidade tornam-se impraticáveis, manter um banco de dados em todos os nós clientes fornecendo tais localizações.

Segundo [MAHMOUD-04], cita que as próximas gerações de sistemas de sistemas distribuídos irão requerer comportamento previsível em áreas tais como *throughput*, escalabilidade, dependabilidade e segurança.

As técnicas adaptativas e reflexivas são apontadas por [MAHMOUD-04] como um paradigma emergente fundamental para o desenvolvimento de plataformas dinâmicas de *middleware* da próxima geração. Estas técnicas capacitam um sistema para adaptar-se automaticamente ao ambiente para cumprir às necessidades do usuário dentro das políticas definidas.

Um sistema adaptativo tem a capacidade de alterar seu comportamento e funcionalidade. Segundo [MAHMOUD-04] um *middleware* adaptativo “ é um *software* cujo comportamento funcional pode ser modificado dinamicamente para aperfeiçoar-se às alterações nas condições ou exigências ambientais.”

[MAHMOUD-04] menciona ainda que tradicionalmente, as plataformas de *middleware* são projetadas para um domínio específico do aplicativo ou cenário de implantação. Na realidade, vários domínios se sobrepõem, e ambientes de implantação são dinâmicos e não estáticos, a tecnologia de *middleware* atual não fornece suporte para lidar com essas condições.

Conforme aborda [MACIAL-04], “o *middleware* adaptativo deve ser reconfigurável, possibilitar que novos serviços sejam acoplados e ter as seguintes características:”

- Suporte a aplicações sensíveis ao contexto e adaptativas;
- Suporte em tempo de execução de comunicações *ad hoc* entre as aplicações;
- Serviços adaptativos específicos de aplicação;
- Interoperabilidade com *middleware* de outros domínios.

[MAHMOUD-04] cita que nos últimos anos, têm surgido plataformas que suportam reconfigurabilidade, permitindo que plataformas sejam personalizadas para uma tarefa específica, este trabalho tem levado ao desenvolvimento de plataformas de *middleware* adaptativo multiuso.”

Um sistema reflexivo é definido por [MAHMOUD-04] como “aquele que pode analisar e *raciocinar* sobre as suas capacidades e ambiente operacional, permitindo que se autoadaptar em tempo de execução”. Ainda conforme expõe [MAHMOUD-04], o *middleware* reflexivo baseia-se no *middleware* adaptativo, fornecendo os meios necessários para permitir que um sistema seja manipulado e adaptado em tempo de execução.

Plataformas reflexivas suportam avançado comportamento adaptativo; a adaptação pode ocorrer de forma autônoma, com base no *status* do sistema, do ambiente, ou nas políticas definidas de seus usuários ou administradores, conforme expõe [MAHMOUD].

Um *middleware* reflexivo utiliza o conceito da refletividade computacional, ou seja, uma aplicação pode acessar algumas partes do estado do sistema logo abaixo e modificá-lo dinamicamente, modificando, então, sua própria semântica, conforme aborda [AGHA-02]¹ apud [MACIEL-04].

Os serviços oferecidos pelas camadas de abstração *middleware* comumente são: [MACIEL-04]

¹ AGHA, Gul A. **Adaptive Middleware**. *Communication on the ACM*. Junho, 2002 v. 45. nº 6 pp. 31-32.

- Gerenciamento e apresentação;
- Computação;
- Gerenciamento de informação;
- Comunicação;
- Gerenciamento de sistema;
- Sistema de entrega.

Segundo expõe [SCHANTZ-01], para aplicações distribuídas, o *middleware Distributed Object Computing* (DOC), atua como um protocolo de rede que possui uma arquitetura decomposta em camadas, que provê:

- *Distribution middleware*, define os modelos de programação de alto nível dos quais as APIs podem estender a capacidade de programação no sistema operacional;
- *Domain-specific middleware services*, são serviços específicos solicitados por domínios determinados, que provêm mecanismos de reutilização. Os serviços oferecidos por esta camada permitem maior crescimento da qualidade do sistema e diminuem o esforço de vida necessários para o desenvolvimento de aplicações distribuídas específicas;
- *Common middleware service*, define serviços de alto nível que podem ser utilizados por programadores para construção de aplicações distribuídas.
- *Host Infrastructure middleware*, encapsula a comunicação proveniente ao sistema operacional estabelecendo componentes reutilizáveis de programação distribuída.

12. ARQUITETURA PROPOSTA

Neste capítulo, será abordada a arquitetura proposta que contempla a necessidade de integração no ambiente corporativo industrial, analisando as ferramentas necessárias para aproximação entre a gestão dos processos administrativos e a gestão produtiva.

O modelo proposto para a integração em aplicações distribuídas contempla a utilização de abstração *middleware* implementado nos níveis de:

- Execução (RTU e CLP);
- Aquisição de dados (SCADA++)
- Integração industrial (MES);
- Gerenciamento (SNMP4J);
- Gestão/planejamento (ERP).

A figura 94 ilustra o modelo proposto.

Figura 94: Modelo proposto de gestão produtiva.

O modelo propõe a adoção de controladores lógicos programáveis implementados com MIB (*Management Information Basic*), tornando-os agentes e definindo uma arquitetura de gerenciamento de objetivos distribuídos na planta industrial, possibilitando o gerenciamento destes dispositivos, conforme ilustra a figura 95 a seguir.

Figura 95: Arquitetura proposta

As atribuições de arquitetura de automação de processo, são definidas pelas implementações das lógicas de automação e controle, programadas nos dispositivos CLPs/RTUs juntamente com os sistemas de supervisão e aquisição de dados (SCADA).

O modelo proposto adota como sistema de supervisão e aquisição de dados o SCADA++, desenvolvido como um sistema de controle baseado em programação orientada à objetos, onde o acesso aos recursos dos *hardwares* dispostos na rede industrial passa a ser realizado via objeto, não mais por *tags*, como nos sistemas convencionais, além de assegurar a independência em relação aos sistemas operacionais dos computadores.

A plataforma de gerenciamento atuará no monitoramento de rede e dos dispositivos da planta industrial, fornecendo parâmetros relacionados aos dispositivos de automação (CLPs, RTUs, *switches* industriais) e parâmetros de rede. O gerenciamento prevê a implementação de SNMP4J, por meio de API *Java* para gerentes e agentes da arquitetura.

Como requisitos de segurança, o protocolo de gerenciamento dispõe de recursos de privacidade e autenticação.

O MES atuará na arquitetura de gestão, provendo a integração industrial entre os sistemas de aquisição de dados (SCADA); realizando coleta/aquisição de dados e parâmetros (processo industrial – sensores e atuadores); no detalhamento e gestão de ordens de serviço e de produção; gestão de recursos materiais e humanos.

As atribuições de manutenção dos dispositivos de automação e da rede, passarão a dispor de informações e dados gerados a partir do gerenciamento SNMP4J por meio dos dados obtidos das MIB implementadas nos controladores lógicos programáveis.

A figura 96 abaixo ilustra a esquema ERP obtido.

Figura 96: ERP com inclusão de camada MES e SNMP4J

A camada de abstração *middleware* proposta deverá prover transparência na comunicação, independente dos protocolos de comunicação do sistema, das plataformas e sistemas operacionais dos computadores, bem como de *hardwares* e dispositivos empregados na arquitetura de gestão. O *middleware* será baseado no modelo de invocação e método remoto RMI (*Remote Method Invocation*); este método de invocação, juntamente com uma API (*Application Program Interface*) possibilita que o *software de middleware* chame métodos de objetos remotos como se estes fossem objetos locais, salientando que a invocação à objetos remotos possui maior latência do que a invocação à objetos locais.

O RMI provê suporte para registro, que possibilita aos clientes realizarem monitoramento em um serviço específico.

A figura 97 a seguir ilustra detalhamento da arquitetura de abstração proposta.

Figura 97: Camada *Middleware*

Como a arquitetura pressupõe objetos remotos, a camada de abstração prove utilização de nível destinado a protocolos de requisição/resposta, para invocação à objetos remotos. Estes protocolos são elaborados para integrar as respostas às correspondentes requisições.

A estrutura da mensagem do protocolo requisição/resposta é ilustrada na figura 98 abaixo.

Tipo da mensagem	<i>int (0=Request, 1=Reply)</i>
Identificador Requisição (Request Id)	<i>int</i>
Referência do Objeto	<i>RemoteObjectRef</i>
Identificador de método (method Id)	<i>int ou Method</i>
Argumentos	<i>// array de bytes</i>

Figura 98: Estrutura da mensagem requisição/resposta

Fonte: [COLOURIS-07]

Na arquitetura proposta será sugerido o *Java RMI*.

Conforme expõe [COSTA-06] a tecnologia *Java* foi desenvolvida pela *Sun Microsystems* para propiciar a escrita de *software* altamente confiável e robusto com *strong typing*¹.

¹ *strong typing* – tipagem forte viabiliza uma abrangente checagem em tempo de compilação para evitar problemas em potencial no “casamento” dos tipos. [COSTA-06]

Os programas RMI apresentam uma arquitetura cliente-servidor, onde os clientes invocam métodos que atuam sobre objetos nos servidores, tanto dos clientes RMI quanto os servidores RMI são codificados na linguagem *Java*. [ALBUQUERQUE-01]

Ao invocar um método sobre um objeto remoto, o cliente RMI atua na realidade sobre um objeto local, que se “faz passar” por objeto remoto, este objeto local é definido como *stub*. O *stub* atua como um *proxy* do objeto remoto e “esconde” do cliente o uso dos serviços providos pelos protocolos de transporte. [ALBUQUERQUE-01].

A figura 99 abaixo ilustra a comunicação entre entidades RMI.

Figura 99: Comunicação entre entidades RMI

Fonte: Adaptado de [COSTA-08a] e [COSTA-06]

O programa cliente faz invocações de métodos pelo objeto *proxy*, o RMI envia a requisição para JVM (*Java Virtual Machine*) remota e redireciona para a implementação, qualquer valor retornado pela implementação é devolvido ao *proxy* e então ao programa cliente.

A figura 100 a seguir ilustra a Invocação Remota.

Figura 100a: Invocação remota

Figura 100b: Interação entre componentes – Invocação remota.

Ao invocar um método remoto, um cliente RMI necessita identificar o objeto remoto sobre o qual o método atua. Para obter uma referência para esse objeto, o cliente recebe a referência como o valor retornado por método ou utiliza um serviço definido por *registry*, que é utilizado tanto por clientes quanto servidores RMI. O servidor de registro atende na porta 1099 (*default*), porém pode ser alterada.

Conforme expõe [COSTA-08a], a operação RMI é definida por um ambiente de comunicação composto por três elementos: cliente RMI, servidor RMI e servidor de registro. Os métodos são acessados pelos clientes no servidor RMI, onde este deve definir uma interface contendo tais métodos.

Ainda conforme [COSTA-08a], o “servidor RMI atua como um servidor TCP, embora os detalhes da conexão os detalhes da conexão não sejam percebidos pelos usuários...”

A figura 101 a seguir ilustra procedimento para construção de aplicação RMI.

Figura 101: Construção de aplicação RMI

Fonte: [ROCHA-03]

A etapa inicial consiste em estabelecer a interface que define os métodos que estarão disponíveis para acesso remoto. Desta forma, é fundamental criar uma interface *Java* com relação de herança, declarando os métodos visíveis (pacote *Java.RMI*).

A tabela 21 a seguir apresenta a lista dos métodos mais utilizados da interface *Context*.

Tabela 21: Métodos utilizados da interface *Context*

Metodo	Funcionalidade
Object addToEnvironment (String, Object)	Adiciona uma nova propriedade ao Context
void bind (String, Obejct)	Associa um nome a um elemento
void close ()	Fecha o Context
Context creatSubcontext (String)	Cria um novo escopo de acordo com a configuração do Context
void destroySubcontext (String)	Remove um escopo de acordo com a configuração do Context
Hashtable getEnvironment ()	Retorna as informações de configuração do Context
NamingEnumeration list (String)	Retorna todas as informações de nomes relacionados a um escopo
NamingEnumeration listBindings (String)	Retorna todas as informações de nomes relacionadas a um escopo, com seus atributos
Object lookup (String)	Recupera uma informação de acordo com o parametro especificado
void rebind (String, Object)	Reassocia um nome a um elemento
Object removeFromEnvironment (String, Object)	Remove uma propriedade do Context
voi rename (String, String)	Associa um novo nome a um elemento
void unbind (String)	Desassocia um nome de um elemento

Fonte: [COSTA-08b]

As fases subsequentes consistem de implementar e compilar os objetos remotos, gerar *stubs* e *skeletons*, implementar aplicações servidor e cliente e compila-los. (ROCHA-03)

Os servidores RMI possuem os objetos remotos com os quais os clientes RMI interagem. Após designados, os objetos remotos permanecem enquanto existirem referências para estes mantidas pelos clientes, conforme expõe [ALBUQUERQUE-01], além dos métodos remotos, os servidores contém o código para:

- Receber as chamadas dos clientes;
- Identificar os métodos a serem chamados;
- Chamar os métodos passando os parâmetros necessários;
- Enviar os resultados aos clientes.

13. CONCLUSÃO E CONSIDERAÇÕES FINAIS

Um aspecto de fundamental relevância observado no desenvolvimento desta dissertação foi a aplicação de arquiteturas de sistemas distribuídos utilizados em diversos sistemas, especialmente no ambiente industrial; onde os padrões de barramentos de campo passaram a dispor em suas estruturas de implementações de *Ethernet* e TCP/IP. Conclui-se desta forma que os padrões de gestão industrial e corporativa serão convergentes e abertos, possibilitando maior flexibilidade, escalabilidade e interoperabilidade.

A rápida ascensão das arquiteturas de sistemas distribuídos proporcionou o surgimento de modelos de gestão empresariais dinâmicos que necessitam de dispositivos ágeis e que deem subsídios aos processos de tomada de decisões com rapidez e eficiência. A integração da informação dentro do ambiente corporativo torna-se fator de essencial proeminência no contexto estratégico e decisório das corporações.

A tecnologia da informação tem contribuído de forma incisiva neste processo de integração da informação compartilhada, ao dispor de modelos e de linguagens computacionais que oferecem mecanismos que possam ser utilizados diretamente em dispositivos fabris, proporcionando acompanhamento em tempo real destes equipamentos com a possibilidade de integração ao ambiente de dados das corporações.

A aplicação de programação orientada à objeto atrelada aos dispositivos e sistemas de automação, como sistemas de aquisição de dados, (SCADA++) e até mesmo em sistemas direcionados inicialmente para ambientes de tecnologia da informação como o gerenciamento de redes (SNMP++ e SNMP4J) mostram a versatilidade e aplicabilidade deste paradigma na integração e acesso de dados em novas plataformas e em sistemas legados. Desta forma são asseguradas características como encapsulamento, herança e polimorfismo à programação de sistemas de monitoramento industrial.

Diversas plataformas surgiram com o propósito de integração, entretanto, a linguagem Java esta sendo implementada por diversos fornecedores e poderá ser a principal arquitetura de objetos distribuídos do futuro [CUMMINS-02].

O segmento de empresas de manufatura que dispõem de sistemas automatizados e informatizados demandará nos próximos tempos de soluções que visem a “Inteligência Operacional” provendo a plena integração de dados em tempo real e de forma segura.

A utilização de soluções de integração comercial como ERP, com módulos direcionados ao ambiente fabril como o MES implementados, com sistemas de gerenciamento e camadas de abstração (*middleware*) possibilitam acessibilidade e o compartilhamento informações e dados gerados.

Por meio da utilização de *middleware JavaRMI*, torna-se possível atingir a integração e interoperabilidade dos dispositivos da planta com robustez e segurança. Esta relação pode ser estabelecida por meio da programação orientada a objeto que é um paradigma de programação avançado.

A integração de dados e informações subsidia a consolidação de arquiteturas integradas como por exemplo SOA (*Service-oriented architecture*), conforme ilustra a figura a seguir.

Legenda:

B2B – *Business to Business*

CRM - *Customer Relationship Management*

Figura: Arquitetura orientada à Serviços

13.1 Trabalhos futuros

Como trabalhos futuros sugere-se a realização de *benchmark* em dispositivos não TI, como *switches* industriais e CLPs implementados com *JavaRMI* e outras plataformas de *middleware* como DCOM e CORBRA. Recomenda-se avaliar a aplicabilidade de *middleware* não somente para monitoramento e gerenciamento de dados, mas também para controle em sistemas de automação.

Também se sugere a aplicação de virtualização sobre os dados integrados, que poderão compor soluções como *CloudComputing*.

14. REFERÊNCIAS

- [ALBUQUERQUE-01] ALBUQUERQUE, Fernando. **TCP/IP Internet Programação de sistemas distribuídos HTML, JAVASCRIPT e JAVA**. Rio de Janeiro: Axcel Books, 2001.
- [ALBUQUERQUE-07] ALBUQUERQUE, Pedro Urbano Braga de; ALEXANDRIA, Auzuir Ripardo de. **Redes Industriais**. Fortaleza: Edições Livro Técnico, 2007.
- [ALMEIDA-08] ALMEIDA, André Gustavo Duarte. **Um Ambiente Multi-Middleware para Desenvolvimento de Aplicações Distribuídas**. Dissertação (Mestrado). Universidade Federal do Rio Grande do Norte, Natal, 2008.
- [ALVES-04] ALVES, Luiz Antonio Antunes. **Gerador de Aplicações: Uma Ferramenta para Gerencia de Sistemas Embutidos utilizando SNMP**. Monografia (Graduação). Universidade Estadual de Montes Claros, Montes Claros, 2004.
- [ALVES-05] ALVES, Jose Luiz Loureiro. **Instrumentação controle e automação de processos**. Rio de Janeiro: LTC, 2005.
- [ARAÚJO-07] ARAÚJO, Erika Maria Teixeira; BATISTA, Mônica de Lourdes Souza; MAGALHÃES, Teresinha Moreira de. **Um estudo sobre as ferramentas OLAP**. Faculdades Integradas Viana Junior, Juiz de Fora, 2007. Disponível em:
http://www.devmedia.com.br/articles/viewcomp_forprint.asp?comp=6691 Acesso em 10 jan. 2010
- [BARBOSA-01] BARBOSA, Álvaro Cesar Pereira. **Middleware para Integração de Dados Heterogêneos**. Tese (Doutorado). Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2001.

- [BENNETT-98a] BENNETT, Geoff. **Internet working com TCP/IP: protocolos, serviços, segurança e performance**. São Paulo: IBPI PRESS, 1998.
- [BENNETT-98b] BENNETT, Geoff. **Internet working com TCP/IP: Tecnologia e Infraestrutura**. São Paulo: IBPI PRESS, 1998.
- [BERNAL-02] BERNAL, Paulo Sergio Milano; FALBRIARD, Claude. **Redes Banda Larga. Roteamento IP, ATM e Interworking em banda larga**. São Paulo: Érica, 2002.
- [BERRY-97] BERRY, Michael J. A.; LINOFF, Gordon. **Data Mining Techniques**. New York: John Wiley, 1997.
- [BIO-08] BIO, Sergio Rodrigues. **Sistema de Informação. Um Enfoque Gerencial**. São Paulo: Atlas, 2008.
- [BORSOI-04] BORSOI, Beatriz Terezinha; SCHULTZ, Rúbia Eliza de Oliveira. **Estudo Comparativo entre CORBA e JAVA RMI**. Anais do Congresso Anual de Tecnologia da Informação, CATI, 2004.
- [BREBNER-05] BREBNER, Paul. et al. **Middleware benchmarking: Approaches, results, experiences**. Wiley InterScience, 2005.
- [CABRAL-05] CABRAL, Jose Manuel Tavares Vieira. **Uma Arquitectura para Adaptação de Tráfego de Baixo Debito em Aplicações de Controlo**. 2005. Tese (Doutorado) – Universidade do Minho, Portugal, 2005.
- [CAMEIRA-03] CAMEIRA, Renato Flório. **Arquitetura Integrada de Sistemas: Modelo de Referencia em um Contexto de (hiper) integração de processos e sistemas nas organizações**. Universidade Federal do Rio de Janeiro, 2003.
- [CAMPOS, 2009] CAMPOS Jr, Lellis do Amaral. **WirelessHART – Tecnologia Wireless aplicada a instrumentos de campo**. Controle & Instrumentação, São Paulo, n. 144, p.74-78, fevereiro 2009.

- [CARMONA-05] CARMONA, Tadeu; HEXSEL, Roberto A. **Universidade redes**. São Paulo: Digerati Books, 2005.
- [CARVALHO-93] CARVALHO, Tereza Cristina Melo de Brito. **Gerenciamento de Redes. Uma Abordagem de Sistemas Abertos**. São Paulo: Makron Books, 1993.
- [CARVALHO-94] CARVALHO, Tereza Cristina Melo de Brito. **Arquiteturas de redes de computadores OSI e TCP/IP**. São Paulo: Makron Books, 1994.
- [CERRI-04] CERRI, M. L.; CAZARINI, E. W. **Diretrizes para implantação de ERPs**, in ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, ENEGEP, Florianópolis, 2004.
- [CISCO-09] CISCO SNA. **Technology Brief**. Disponível em http://www.cisco.com/en/US/tech/tk331/tk341/tech_brief09_186a00800b4258.html Acesso em 10 de janeiro de 2010.
- [CHERMONT-07] CHERMONT, Marlon Gripp. **Proposta de Desenvolvimento de um Agente Proxy SNMP para Gerenciamento de Redes LonWorks**. Dissertação (Mestrado). Universidade de São Paulo, São Paulo, 2007.
- [CHOWDHURY-02] CHOWDHURY, Dhiman D. **Projetos avançados de redes IP**. Rio de Janeiro: Campus, 2002.
- [COMER-06] COMER, Douglas E. **Interligação de redes com TCP/IP. Princípios, protocolos e arquitetura**. Rio de Janeiro: Elsevier, 2006.
- [CORNACHIONE-01] CORNACHIONE, Jr. Edgard B. **Sistemas Integrados de gestão**. São Paulo: Atlas, 2001.
- [CORREA-08] CORREA, Henrique Luiz; GIANESI, Irineu Gustavo Nogueira; CAON, Mauro. **Planejamento, Programação e Controle da Produção**. São Paulo: Atlas, 2008.
- [CORREA-08] CORREA, Henrique Luiz; GIANESI, Irineu Gustavo Nogueira; CAON, Mauro. **Planejamento, Programação e Controle da Produção**. São Paulo: Atlas, 2008.

- [CORTES-03] CORTES, Pedro Luz. **Sistemas Operacionais Fundamentos**. São Paulo: Érica, 2003.
- [COSTA-05] COSTA, Daniel Gouveia. **SCTP Uma alternativa aos tradicionais protocolos de transporte da Internet**. Rio de Janeiro: Ciência Moderna, 2005.
- [COSTA-06] COSTA, Luis Carlos Moreira. **Java avançado**. Rio de Janeiro: Ciência Moderna, 2006.
- [COSTA-08a] COSTA, Daniel G. **JAVA em rede. Programação Distribuída na Internet**. Rio de Janeiro: Brasport, 2008.
- [COSTA-08b] COSTA, Daniel Gouveia. **Java em Rede: Recursos Avançados de Programação**. Rio de Janeiro: Brasport, 2008.
- [COULOURIS-07] COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. **Sistemas distribuídos. Conceitos e Projetos**. Porto Alegre: Bookman, 2007.
- [CUMMINS-02] CUMMINS, Fred A. **Integração de Sistemas. EAI – Enterprise Application Integration: Arquiteturas para integração de sistemas e aplicações corporativas**. Rio de Janeiro: Campus, 2002.
- [CUNHA-04] CUNHA, Judson Michel. **Modelo de gerenciamento integrado para ambientes de redes industriais**. Dissertação (Mestrado). Universidade Federal de Santa Catarina, Florianópolis, 2004.
- [CUSTODIO-95] CUSTÓDIO, João. **Novell NetWare 4.1 Planejamento, Implementação e Gerenciamento**. Rio de Janeiro: Brasport, 1995.
- [CYCLADES-04] CYCLADES Brasil. **Guia Internet de conectividade**. São Paulo: Editora SENAC, 2004.
- [DAVENPORT-98] DAVENPORT, Thomas H. **Putting the enterprise into the enterprise system**. Harvard Business Review, Boston, n.1, jul.1998. 13 p. Disponível em: <http://www.hbsp.harvard.edu>. Acesso em: 20 out. 2007.

- [DATE-95] DATE, C.J. **An Introduction to database systems**. 6.ed. Reading: Addisison-Wesley, 1995, p. 2-9.
- [DERFLER-95] DERFLER, Frank J. **Guia de Conectividade**. Rio de Janeiro: Campus, 1995.
- [DEITEL-05] DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. **Sistemas Operacionais**. 3ª ed. São Paulo: Pearson Prentice Hall, 2005.
- [DIOGENES-01] DIOGENES, Yuri. **Certificação CISCO**. Rio de Janeiro: Axcel Books, 2001.
- [DOMINGUES-02] DOMINGUES, Edi; MILHOMEM, Fabiana. **Integrando redes SNA & TCP/IP**. Rio de Janeiro: Alta Books, 2002.
- [DOWNES-98] DOWNES, Kevin; FORD, Merilee; LEW, H. Kim; SPAINER, Steve; STEVENSON, Tim. **Internetworking Technologies Handbook**. Indianapolis: Cisco Press, 1998.
- [DONAHUE-08] DONAHUE, Garry A. **Redes Robustas**. Rio de Janeiro: Alta Books, 2008.
- [EFACEC-06] EFACEC Sistemas de Electronica. **Automação de Sistemas de Energia**. Scatex. 2006. Disponível em: http://www.efacec.com.br/docs/TDSC_SCATEXEMS_PT.pdf. Acesso em 30 mar 2007.
- [EDWARDS-99] EDWARDS, Jeri; HARKEY, Dan; ORFALI, Robert. **Client/Server Survival Guide**. 3ª Edition. New York: Wiley, 1999.
- [FERNANDES-06] FERNANDES, V. **Visão atual da TI no chão de fábrica ERP**, Automação e Controle, Anais Congresso e Exposição Visual atual Internacional de Produção Industrial, ProIndústria, 2006.

- [FERREIRA-01] FERREIRA, Claudio Luís. **Maestro: Um Middleware para Suporte a Aplicações Distribuídas Baseadas em Componentes de Software**. Dissertação (Mestrado). Escola Politécnica da Universidade de São Paulo, 2001.
- [FOROUZAN-06] FOROUZAN, Behrouz A. **Comunicação de dados e redes de computadores**. Porto Alegre: Bookman, 2006.
- [FOUNDATION-07] FOUNDATION FIELDBUS Part 4 Communication. Smar Automação Industrial. Disponível em: <http://www.smar.com/brasil2/fieldbus.asp> . Acesso em: 24 fev 2007.
- [FREITAS-04] FREITAS, Clayton Araújo; MONTEIRO, João W. Alves. **Análise de protocolos na área de gerência de redes (SNMP/RMON)**. Monografia (Graduação). Universidade Federal de Goiás, Goiânia, 2004.
- [GABOS-09] GABOS, Denis. **Gerenciamento de Redes**. Laboratório de Arquitetura e Redes de Computadores. Escola Politécnica da Universidade de São Paulo, Notas de aulas, 2009.
- [GABRIEL-01] GABRIEL, Torres. **Redes de computadores: Curso Completo**. Rio de Janeiro: Axcel Books, 2001.
- [GEORGINI-00] GEORGINI, Marcelo. **Automação Aplicada. Descrição e Implementação de Sistemas Seqüenciais com PLCs**. São Paulo, Érica, 2000.
- [GLOVER-07] GLOVER, Bill; BHATT, Himanshu. **Fundamentos de RFID**. Rio de Janeiro: Alta Books, 2007.
- [HALL-99] HALL, Jon. **LINUX**. Rio de Janeiro: Campus, 1999.
- [HANCOCK-03] HANCOCK, William M. GALLO, Michael A. **Comunicação entre computadores e tecnologias de rede**. São Paulo: Pioneira Thomson Learning, 2003.
- [HARRISON-94] HARRISON, Peter G.; PATEL, Naresh M. **Performance Modelling of Communication Networks and Computer Architectures**. Addison Wesley, 1994.

- [HART-03] **O protocolo digital Hart**, 2003. Disponível em http://www.smar.com/PDFs/ApplicationNotes/AppNotes_HART.pdf Acesso em 14 nov. 2009.
- [HOPCROFT-02] HOPCROFT, John; ULLMAN, Jeffrey D.; MOTWANI, Rajjev. **Introdução à Teoria de Autômatos, Linguagens e Computação**. Rio de Janeiro: Elsevier, 2002.
- [HUBBARD-03] HUBBARD, John R. **Programação em C++**. Porto Alegre: Bookman, 2003.
- [JURIZATO-02] JURIZATO, Luis Augusto; PEREIRA, Paulo Sergio R. **Sistemas Supervisórios**. Network Technologies, Nova Odessa, v.1/2, p. 105-114, 2002.
- [KRISHNAMURTHY-01] KRISHNAMURTHY, Balachander; REXFORD, Jennifer. **Redes para a Web**. Rio de Janeiro. Campus. 2001.
- [LAUDON-98] LAUDON, Kenneth C.; LAUDON, Jane P. **Management Information Systems: New approaches to organization & technology**. 5ª ed. New Jersey: Prentice Hall, 1998.p 280-282.
- [LAUREANO-06] LAUREANO, Marcos. **Máquinas Virtuais e Emuladores. Conceitos, Técnicas e Aplicações**. São Paulo: Novatec Editora, 2006.
- [LEE-01] LEE, Richard C.; TEPFENHART, William M. **UML e C++ Guia Prático de Desenvolvimento Orientado à Objeto**. São Paulo: Makron Books, 2001.
- [LOPES-00] LOPES, Aldabó Ricardo. **Sistemas de redes para controle e automação**. Rio de Janeiro: Book Express, 2000.
- [LONWORKS-09] LonWorks **Technology Overview**, 2009. Disponível em: <http://www.echelon.com/communities/energycontrol/developers//lonworks/default.htm> Acesso em 22 nov. 2009.
- [LOPES-02] LOPES, Raquel Vigolvino. **Melhores Práticas para a Gerência de Redes de Computadores**. Dissertação

(Mestrado). Universidade Federal da Paraíba, Campina Grande, 2002.

- [LUGLI-07] LUGLI, Alexandre Baratella. **Uma Ferramenta Computacional para Análise de Topologia e Tráfego para Redes Ethernet Industriais**. Dissertação (Mestrado). Universidade Federal de Itajubá, Itajubá, 2007.
- [MACHADO-07] MACHADO, Francis Berenger; MAIA, Luiz Paulo. **Arquitetura de sistemas operacionais**. 4^a ed. Rio de Janeiro: LTC, 2007.
- [MACIEL-04] MACIEL, Rita Suzana Pitangueira; ASSIS, Semírames Ribeiro de. **Uma solução para o desenvolvimento de aplicações distribuídas**, Salvador, 2004.
- [MAHMOUD-04] MAHMOUD, Qusay. **Middleware for Communications**. USA: Wiley, 2004.
- [MARANGONI-08] MARANGONI, Matheus. **Software de Gestão ERP**. Relatório Completo. SEBRAE Serviço Brasileiro de Apoio às Micros e Pequenas Empresas, 2008.
- [MELLQUIST-94] MELLQUIST, Peter E. **SNMP++ Na Object Oriented Approach For Network Management Programming Using C++**. Revision 2.1. Hewlett Packard, 1994.
- [MENASCE-02] MENASCE, Daniel A.; ALMEIDA, Virgílio A. F. **Planejamento de capacidade para serviços na Web. Métricas, Modelos e Métodos**. Rio de Janeiro: Campus, 2002.
- [MIÃO-07] MIÃO, Rodolfo. **Implementação de Sistemas ERP ("Enterprise Resource Planning") SAP R/3® e suas Tecnologias Middleware**. Dissertação (MBA). Universidade de São Paulo, São Paulo, 2007.
- [MODBUS-96] Modicon Modbus **Protocol Reference Guide**, 1996. Disponível em http://www.modbus.org/docs/PI_MBUS_300.pdf Acesso em 05 de nov. 2009.

- [MODBUS-06] Modbus **Application Protocol Specification** v1.1b, 2006. Disponível em http://www.modbus.org/docs/modbus_application_protocol_v1_1b.pdf. Acesso em 12 dez. 2009.
- [MORAES-07] MORAES, Cícero Couto de; CASTRUCCI, Plínio de Lauro. **Engenharia de Automação Industrial**. 2ª ed. Rio de Janeiro: LTC, 2007.
- [MOURA-86] MOURA, J. A. B.; GIOZZA, W. F.; SAUVÉ, J. P.; ARAÚJO, J. F. M. de. **Redes Locais de Computadores**. São Paulo: McGraw-Hill, 1986.
- [MUELLER-02] MUELLER, Stephen M. **APIs AND PROTOCOLS FOR CONVERGENT NETWORK SERVICES**. New York: McGraw-Hill, 2002.
- [MULLENDER-95] MULLENDER, Sape. **Distributed Systems**. New York: Addison-Wesley, 1995.
- [NCS-04] NCS TIB 04-1. TECHNICAL Information Bulletin 04-1, 2004. NATIONAL COMMUNICATIONS, **SYSTEM - SCADA Supervisory Control and Data Acquisition Systems**. Disponível em: http://www.ncs.gov/library/tech_bulletins/2004/tib_04-1.pdf Acesso em: 17 set 2007.
- [NETO-08] NETO, Pedro Otávio Alves; SILVA, Robson Soares. **Tecnologias de Sistemas Distribuídos Implementados em JAVA: Sockets, RMI, RMI-IIOP e CORBA**. Anuário da Produção Acadêmica Docente, vol. II, nº. 3, 2008.
- [NEVES-08] NEVES, José Manoel Souza das; SANTOS, Fernando César Almada. **Implantação de tecnologias de informação utilizadas na integração entre o chão-de-fábrica e os sistemas ERP**. Controle & Instrumentação, São Paulo, n. 143, p. 56-61, 2008.
- [NIRVA-01] NIRVA, Morisseau-Leroy. **Oracle 8i Programação de Componentes Java com EJB, CORBA e JSP**. Rio de Janeiro: Campus, 2001.

- [ODVA-07a] ODVA Profile File. **Network Infrastructure for EtherNet/IP**: Introduction and Considerations. ODVA Publication, 2007.
- [ODVA-07b] Especificação EtherNet/IP. Disponível em http://www.odva.org/Portals/0/Library/Publications_Numbered/PUB00035R0_Infrastructure_Guide.pdf Acesso em 18 out. 2009.
- [OLIFER-08] OLIFER, Victor; OLIFER, Natália. **Redes de Computadores: Princípios, Tecnologias e Protocolos para o Projeto de Redes**. Rio de Janeiro: LTC. 2008.
- [ORFALI-98] ORFALI, Robert; HARKEY, Dan. **Client/Server Programming with JAVA and CORBA**. 2ª ed. New York: Wiley, 1998.
- [PAVA-09] PAVA, Pablo; ELIAS, Fábio Henrique. **MOM – A terceira geração de sistemas ERP**. InTech, São Paulo, n. 116, p. 26-29, 2009.
- [PEREIRA-03] PEREIRA, Carlos Eduardo. **Evolução dos Sistemas de Controle**. Porto Alegre, Brasil, 2003. Disponível em: <http://www.eleto.ufrgs.br/~cpereira>. Acesso em 22 out 2008.
- [PETERSON-04] PETERSON, Larry L. BRUCE S. Davie. **Redes de Computadores. Uma abordagem de sistemas**. Rio de Janeiro: Elsevier, 2004.
- [PINHEIRO-06] PINHEIRO, José Mauricio Santos. **Introdução às Redes de Supervisão e Controle**, 2006.
- [PITANGA-04] PITANGA, Marcos. **Construindo supercomputadores com Linux**. 2ª ed. Rio de Janeiro: Brasport, 2004.
- [PRESSMAN-06] PRESSMAN, Roger S. **Engenharia de Software**. São Paulo: McGraw-Hill, 2006.

- [PROFFITT-01] PROFFITT, Brian; ZUPAN, Ann. **XHTML Desenvolvimento WEB**. São Paulo: Makron Books, 2001.
- [PRUDENTE-07] PRUDENTE, Francesco. **Automação Industrial**. Rio de Janeiro: LTC, 2007.
- [PROFIBUS-06] PROFIBUS **Descrição Técnica. Associação PROFIBUS Brasil** 2006. Disponível em <http://www.profibus.org.br>. Acesso em 12 mar. 2008.
- [PUPO-02] PUPO, Maurício Santos. **Interface homem - máquina para supervisão de um CLP em controle de processo através da WWW**. Dissertação (Mestrado) – Escola Politécnica Universidade de São Paulo, São Paulo, 2002.
- [RALHA-04] RALHA, Claudio. **Segredos do Visual Studio.Net**. São Paulo: Digital Books, 2004.
- [RENAUD-94] RENAUD, Paul E. **Introdução aos sistemas cliente/servidor**. Rio de Janeiro: Infobook, 1994.
- [RISO-04] RISO, Bernardo Gonçalves; MACIEL, Cristiano; SOUZA, Ivanise Volpato; ROSSI, Leila Lisiane; NOTARE, Mirela Sechi Moretti; TONIN, Neilor Avelino. **Engenharia de Protocolos com LOTOS/ISO**. Florianópolis: Editora UFSC, 2004.
- [ROCHA-03] ROCHA, Helder da. **Fundamentos de Objetos Remotos**. Java 2 Standard Edition, 2003. Disponível em <http://www.argonavis.com.br>: Acesso em 12 de jul 2010.
- [RODRIGUES-07] RODRIGUES, Gustavo Viana. **Disponibilização de Serviços de Segurança para Sistemas Distribuídos através de WebServices**. Monografia (Graduação). Universidade Regional de Blumenau, Blumenau, 2007.
- [ROSÁRIO-05] ROSÁRIO, João Mauricio. **Princípios de Mecatrônica**. São Paulo: Prentice Hall, 2005.

- [ROSS-06] ROSS, Keith W.; KUROSE, James F. **Redes de Computadores e a Internet. Uma abordagem top-down**. 3ª ed. São Paulo: Pearson Addison Wesley, 2006.
- [SANTOS-06] SANTOS, Breno Henrique Vivarelli Diego Lima. **Automação e Controle da Usina Hidrelétrica de Itaipu utilizando o Sistema SCADA/MES**. Monografia (Graduação). Universidade Estadual de Ponta Grossa, Ponta Grossa, 2006.
- [SANTOS-07] SANTOS, Alfredo Luiz dos. **Integração de Sistemas com Java**. Rio de Janeiro: Brasport, 2007.
- [SCHACH-09] SCHACH, Stephen. **Engenharia de Software: Os Paradigmas Classico & Orientado à Objetos**. 7ª ed. São Paulo: McGraw-Hill, 2009.
- [SCHANTZ-01] SCHANTZ, Richard E.; SCHMIDT, Douglas. **Middleware for Distributed Systems** – Evolving the Common Structure for Network-centric Applications Encyclopedia of Software Engineering, 2001.
- [SCHATT-93] SCHATT, Stan. **Como Funcionam as REDES LOCAIS**. Rio de Janeiro: Berkeley, 1993.
- [SCHMIDT-01] SCHMIDT, Kevin J. **SNMP essencial**. Rio de Janeiro: Campus, 2001
- [SCHMIDT-05] SCHMIDT, Kevin J.; DOUGLAS R. Mauro. **SNMP Essential**. USA: O'Reilly, 2005.
- [SEIXAS-03] SEIXAS, Constantino. **Programação Concorrente em Ambiente Windows. Uma Visão de Automação**. Belo Horizonte: Editora UFMG, 2003.
- [SEIXAS-09] SEIXAS, Constantino. **MES. Funcionalidades**. InTech, São Paulo, n. 116, p. 20-25, 2009.
- [SERAIN-02] SERAIN, Daniel. **Middleware and Enterprise Application Integration**. London: Springer, 2002

- [SHELDON-93] SHELDON, Tom. **Novell NetWare 4: The Complete Reference**. Berkeley: McGraw-Hill, 1993
- [SHIRASUNA-08] SHIRASUNA, Mauro. **MES: Situação presente e expectativa do futuro**. Controle & Instrumentação, São Paulo, n. 143, p. 66-71, 2008.
- [SIEGEL-96] SIEGEL; J. CORBA: **Fundamentals and Programming**, John Wiley & Sons Inc. 1996.
- [SILBERSCHATZ-99] SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**. São Paulo: Makron Books, 1999.
- [SILVA-03] SILVA, Lino Sarlo. **Virtual Private Network**. São Paulo: Novatec Editora, 2003.
- [SILVA-05] SILVA, Ana Paula Gonçalves da; SALVADOR, Marcelo. **O que são Sistemas Supervisórios?** Elipse, 2005.
- [SILVA-06] SILVA, Luiz Antonio Ferreira da. **Análise de Desempenho de Protocolos de Transporte para Redes de Alta Velocidade**. Dissertação (Mestrado) - Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2006.
- [SILVEIRA-98] SILVEIRA, Paulo R. da; SANTOS, Winderson Eugenio dos. **Automação e Controle Discreto**. 6ª. Ed. São Paulo: Érica, 1998.
- [SIPSER-07] SIPSER, Michael. **Introdução à teoria da computação**. São Paulo: Thompson Learning, 2007.
- [SIMCSIK-02] SIMCSIK, Tibor; POLLONI, Enrico G. F. **Tecnologia da Informação Automatizada**. São Paulo: Berkeley Brasil, 2002.
- [SOARES-99] SOARES, Luis Fernando Gomes; LEMOS, Guido; COLCHER, Sergio. **Redes de computadores das LAN's MAN's e WAN's as redes ATM**. 2ª ed. Rio de Janeiro: Campus, 1999.

- [SOMMERVILLE-07] SOMMERVILLE, Ian. **Engenharia de Software**. 8ª ed. São Paulo: Pearson Addison-Wesley, 2007.
- [SOUZA-00] SOUZA, Cesar Alexandre de. **Sistemas Integrados de Gestão Empresarial**. Dissertação (Mestrado). Faculdade de Economia, Administração e Contabilidade USP, São Paulo, 2000.
- [SOUSA-03] SOUSA, Paulo Manuel Baltarejo de. **EtherNet/IP**. Monografia (Graduação). Instituto Superior de Engenharia do Porto, Portugal, 2003.
- [SOUZA-05] SOUZA, Rodrigo Barbosa. **Uma Arquitetura para Sistemas Supervisórios Industriais e sua Aplicação em Processos de Elevação Artificial de Petróleo**. Dissertação (Mestrado). Universidade Federal do Rio Grande do Norte, Natal, 2005.
- [STALLINGS-98] STALLINGS, William. **SNMP v3 A Security enhancement for SNMP**. IEE communications Surveys & tutorials, v.1, n.1, 1998. p.2-17. Disponível em: <http://www.dl.comsoc.org/cocoon/consoc/html/index.htm> Acesso em: 20 dez. 2007
- [STALLINGS-99] STALLINGS, William. **SNMP, SNMPv2, SNMPv3, RMON 1 and 2**. New York: Addison Wesley, 1999.
- [STALLINGS-08] STALLINGS, Willian. **Criptografia e Segurança de Redes: Princípios e Práticas**. São Paulo: Pearson Prentice Hall, 2008.
- [STEMMER-01] STEMMER, Marcelo Ricardo. **Sistemas Distribuídos e Redes de Computadores para Controle e Automação Industrial**. Universidade Federal de Santa Catarina, 2001.
- [TAKAHASHI-90] TAKAHASHI, Tadao; LIESEMBERG, Hans K. E.; XAVIER, Daniel Tavares. **Programação Orientada à Objetos: Uma Visão Integrada do Paradigma de Objetos**. São Paulo: IME-USP, 1990.
- [TANG-92] TANG, Adrian; SCOGGINS, Sophia. **Open Networking with OSI**. New Jersey: Prentice Hall, 1992.

- [TANENBAUM-03] TANENBAUM, Andrew S. **Redes de computadores**. 4. Ed. Rio de Janeiro: Campus, 2003.
- [TANENBAUM-07] TANENBAUM, Andrew S.; STEEN, Maarten Van. **Sistemas distribuídos. Princípios e paradigmas**. 2ª ed. São Paulo: Pearson Prentice Hall, 2007.
- [THIEME-05] THIEME, Moises. **Modelo de Governança em Facilidades Prediais para Centros de Tecnologia da Informação de Instituições Financeiras**. Dissertação (MBA). Universidade de São Paulo, São Paulo, 2005.
- [THOMPSON-02] THOMPSON, Marco Aurélio. **Proteção e Segurança na Internet**. São Paulo: Érica, 2002.
- [TOVAR-96] TOVAR, Eduardo. **Introdução à Informática Industrial**. Lisboa: IPP-ISEP, 1996.
- [TOVAR-03] TOVAR, Eduardo. **Produção Integrada por Computador**. Instituto Politécnico do Porto, Lisboa, Portugal, 2003. Disponível em: <http://www.dei.isep.ipp.pt/~emt/infind/apontamentos.html>. Acesso em 23 de jan. 2010.
- [VIANA-08] VIANA, Eliseu Ribeiro Cherene. **Virtualização de servidores LINUX para redes corporativas**. Rio de Janeiro: Editora Ciência Moderna, 2008.
- [YOURDON-99] YOURDON, Edward; ARGILA, Carl. **Análise e Projeto Orientados à Objetos. Estudos de Casos**. São Paulo: Makron Books, 1999.
- [WADLOW-00] WADLOW, Thomas A. **Segurança de redes. Projeto e gerenciamento de redes seguras**. Rio de Janeiro: Campus, 2000.
- [WANG-98] WANG, Charles B. **Techno Vision II**. São Paulo: Makron Books, 1998.
- [WATANABE-06] WATANABE, Edson Hiroshi. **Aplicação de Software Aberto em Redes Industriais**. Dissertação (Mestrado). Universidade Federal do Paraná, Curitiba, 2006.

[ZOLDAN-04]

ZOLDAN, Alan David; SÁ, Claudio Cesar; AGNOL, Everson Dall; PARPINELLI, Rafael; DECKER, Rangel. **SCADA ++ A Distributed Object Oriented Control and Data Acquisition System**, Joinville, 2004.