

T0163/98

# ESCALONAMENTO TEMPORAL EM REDE FIELD BUS



Afonso Celso Caldas Pacheco

DEDALUS - Acervo - EESC



31100016668

Dissertação apresentada à Escola de Engenharia de São Carlos da Universidade de São Paulo, como parte dos requisitos para obtenção do título de Mestre em Engenharia Mecânica.

ORIENTADOR: Prof. Dr. Mário Pinotti Jr.

São Carlos

1998

Class.	TESE/EESC
Cutt.	4335
Tombo	T0163198

31100016668

st 0979398

**Ficha catalográfica preparada pela Seção de Tratamento  
da Informação do Serviço de Biblioteca - EESC-USP**

P116e Pacheco, Afonso Celso Caldas  
Escalonamento temporal em rede Fieldbus /  
Afonso Celso Caldas Pacheco. -- São Carlos,  
1998.

Dissertação (Mestrado) -- Escola de Engenharia  
de São Carlos-Universidade de São Paulo, 1998.  
Área: Engenharia Mecânica.  
Orientador: Prof. Dr. Mário Pinotti Jr..

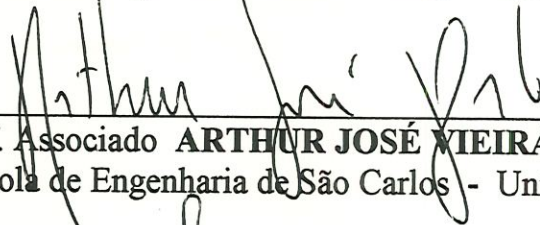
1. Fieldbus. 2. Field bus. 3. Escalonamento.  
4. Rede de campo. 5. Rede de chão de fábrica.  
I. Título.

## FOLHA DE APROVAÇÃO

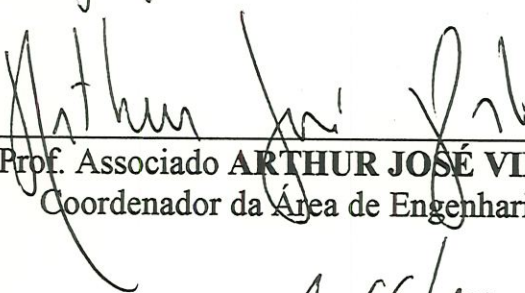
Candidato: Engenheiro **AFONSO CELSO CALDAS PACHECO**

Dissertação defendida e aprovada em 23-04-1998  
pela Comissão Julgadora:

  
\_\_\_\_\_  
Prof. Titular **MARIO PINOTTI JUNIOR (Orientador)**  
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

  
\_\_\_\_\_  
Prof. Associado **ARTHUR JOSÉ VIEIRA PORTO**  
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

  
\_\_\_\_\_  
Prof. Doutor **ADILSON GONZAGA**  
(Escola de Engenharia de São Carlos - Universidade de São Paulo)

  
\_\_\_\_\_  
Prof. Associado **ARTHUR JOSÉ VIEIRA PORTO**  
Coordenador da Área de Engenharia Mecânica

  
\_\_\_\_\_  
**JOSÉ CARLOS A. CINTRA**  
Presidente da Comissão de Pós-Graduação da EESC

Aos meus queridos Pais

## **Agradecimentos**

Meus sinceros agradecimentos a:

Todos que contribuíram para a conclusão deste trabalho, em especial ao Professor Pinotti pela orientação e apoio, à Cristina pela inestimável ajuda, à Ana Paula e Beth pela dedicação. E, por fim, à minha grande amiga Jaqueline pela força e compreensão.

## Sumário

1. Introdução .....	1
1.1 Contexto .....	1
1.2 Objetivos da pesquisa .....	4
2. Revisão Bibliográfica .....	5
2.1 Transmissão de dados em ambiente industrial.....	5
2.1.1 Evolução da transmissão de sinais .....	5
2.1.2 Evolução dos sistemas de transmissão .....	8
2.2 Controle, do clássico ao moderno .....	12
2.3 Controle em rede .....	14
2.3.1 Controle em rede SDCD.....	15
2.3.2 Controle em rede de campo Fieldbus .....	16
2.4 Redes locais industriais .....	17
2.4.1 O modelo de referência OSI/ISO.....	21
2.4.2 Arquitetura MAP .....	26
2.4.2.1 Projeto MAP/EPA.....	29
2.4.2.2 Projeto Mini-MAP.....	30
2.4.2.3 Protocolo MMS.....	31
2.4.3 Fieldbus - Barramento de Campo .....	32
2.4.3.1 Principais padrões para protocolos Fieldbus.....	35
2.5 Estratégias de Controle em Tempo Real .....	41
3. Escalonamento da comunicação no Fieldbus .....	43
3.1 Introdução.....	43
3.2 Tipo de comunicação no Fieldbus .....	44
3.3 Algoritmos de escalonamento.....	46
4. Características técnicas dos protocolos relativas à transmissão .....	51
4.1 PROFIBUS (PROcess FieldBUS) .....	52
4.2 FIP (Factory Instrumentation Protocol) .....	53

4.3 Protocolo IEC/ISA SP-50 .....	56
5. Solução para transferência de dados e cálculo de tempos envolvidos .....	59
5.1 Uma comparação entre monociclo e multiciclo .....	59
5.2 Cálculo de tempos envolvidos .....	61
6. Procedimento para medida de tempo em uma rede Fieldbus .....	67
6.1 Apresentação do aplicativo de simulação em rede.....	67
6.2 Medida do tempo global.....	73
7. Conclusão .....	77

## Lista de figuras

FIGURA 2.1 Evolução dos sistemas de transmissão em chão de fábrica.....	12
FIGURA 2.2 Sistema Digital de Controle Distribuído.....	16
FIGURA 2.3 Sistema digital de controle Fieldbus.....	17
FIGURA 2.4 Modelo de hierarquia para comunicação industrial. ....	19
FIGURA 2.5 Transferencia de dados no modelo OSI/ ISO. ....	26
FIGURA 2.6 Arquitetura MAP/EPA.....	29
FIGURA 2.7 Arquitetura Mini-MAP. ....	31
FIGURA 2.8 Alocação das camadas OSI/ISO no Profibus.....	36
FIGURA 2.9 Repartição do tempo entre três tarefas através da interrupção. ....	42
FIGURA 3.1 Ciclo com fase síncrona e fase assíncrona.....	44
FIGURA 4.1 Método de acesso ao meio físico no Profibus.....	53
FIGURA 4.2 Estrutura do ID-frame e VAR-frame do protocolo FIP.....	54
FIGURA 4.3 Envio de solicitação no protocolo FIP. ....	54
FIGURA 4.4 Resposta à solicitação no protocolo FIP. ....	55
FIGURA 4.5 Mecanismo de passagem de token no protocolo SP-50. ....	57
FIGURA 5.1 Mecanismos de monociclos e multiciclos. ....	60
FIGURA 5.2 Divisão do ciclo primário. ....	62
FIGURA 5.3 Fluxograma do algoritmo de varredura principal. ....	66
FIGURA 6.1 Curvas de consumo Ar/Combustível e Potência efetiva em função da variação da velocidade angular. ....	69
FIGURA 6.2 Diagrama das partes básicas do programa de simulação contido no slave. ....	70
FIGURA 6.3 Fluxograma da tarefa para geração gráfica. ....	71
FIGURA 6.4 Fluxograma da tarefa do protocolo de rede. ....	72
FIGURA 6.5 Representação gráfica do tempo global ( T ). ....	73
FIGURA 7.1 Exemplo de aplicações distribuídas num Fieldbus.....	78



## Lista de tabelas

TABELA 2.1 Evolução dos sistemas de transmissão e ganhos na precisão.....	10
TABELA 2.2 Descrição de alguns sistemas Fieldbus no mercado.....	34
TABELA 2.3 Comparação entre os barramentos SP-50, FIP, PROFIBUS.....	40
TABELA 3.1 Quadro comparativo entre Ciências da computação, Pesquisa Operacional e Comunicação no Fieldbus. ....	45
TABELA 6.1 Ensaio de potência efetiva líquida em um motor Johnson RD32C....	68
TABELA 6.2 Valores obtidos para o tempo global T. ....	75

## Resumo

A evolução das técnicas de controle em sistemas seqüenciais e distribuídos alcançou um nível onde o protocolo de comunicação analógico é, finalmente, substituído por um protocolo digital. Isso é realizado através da nova tecnologia de comunicação em chão de fábrica conhecida como Fieldbus. Nesta dissertação são apresentadas as etapas da evolução das técnicas de controle culminando com o surgimento do Fieldbus. É apresentada uma revisão das técnicas de controle no Fieldbus, a questão do escalonamento da comunicação, características técnicas dos protocolos mais importantes, principais algoritmos para solucionar a questão do escalonamento e, a realização de um procedimento de medida de tempo em uma rede Fieldbus.

## ABSTRACT

The development of the control systems obtain a level where the analogical communication protocol is substitute for another digital. This is realized with the new technology of low-level industrial communication, known as Fieldbus. This dissertation presents evolutions stages of control systems culminating in Fieldbus. Yours main characteristics are described, communication scheduling problems, technique characteristics of the most important protocols in the market, main algorithms to resolve the scheduling problems and execution of a time measurement behavior in a Fieldbus network.

# 1. Introdução

## 1.1 Contexto

No atual cenário mundial, culminando com a questão da globalização, diversas empresas têm passado por modificações em seus processos de fabricação, engenharia e manutenção com o objetivo de alcançar não só menores custos mas também melhorias na qualidade de seus processos e produtos e, conseqüentemente a obtenção de certificações (p.e. ISO 9000) que tornam-se cada vez mais mandatórias para todas as empresas, particularmente as exportadoras.

Diante destes fatos e do crescente aumento da complexidade das aplicações de controle industrial, com integração cada vez maior entre controladores e sensores/atuadores, as redes locais de comunicação de dados assumem uma posição de vital importância e destaque, uma vez que têm como objetivo o compartilhamento e a interconexão de recursos computacionais, tais como hardware e software, geograficamente dispersos em nível local.

Durante os últimos 20 anos, o grande desenvolvimento da microeletrônica possibilitou o aparecimento de circuitos integrados (CIs) de altíssima escala de integração. CRESTONI\* apud NETO (1995) relata que a aplicação desses CIs na indústria de instrumentação acarretou, por sua vez, o aparecimento de instrumentos de campo dotados de capacidades até então delegadas somente aos sistemas hierarquicamente superiores. Segundo STAROSWIECKI (1996), este desenvolvimento proporcionou o surgimento de numerosas aplicações que integram armazenamento e processamento de dados, revolucionando toda a tecnologia de controle.

---

\* CRESTONI, M.S. O *Fieldbus*: a nova tecnologia que revoluciona a automação e a disputa pelo seu padrão. *Eletricidade Moderna*, n.208, jul., 1991 apud Neto, E.F.S. et al. Comunicação em chão de fábrica. São Paulo, ISA, 1995. p.233.

Porém, essa nova geração de instrumentos dotados de “inteligência” passam a manipular e transmitir dados referentes ao processo digitalmente, não tendo, desta forma, suas potencialidades totalmente utilizadas devido ao padrão analógico de transmissão por corrente (4 a 20 mA) utilizado no chão de fábrica. Desta forma, apesar de amplamente difundido na indústria, este padrão passa a ser uma limitação para os novos instrumentos já que impossibilita a transmissão de uma maior variedade de sinais bem como das variáveis do processo tais como: sinais de controle e status do equipamento, que possibilitam a calibração remota e facilita o diagnóstico de falhas. Faltava, desta forma, uma maneira segura de se promover a troca de informações entre os instrumentos de campo (sensores/atuadores, transmissores, etc.) e entre eles e os sistemas hierarquicamente superiores, tais como CLP's, CNC's, etc. .

Nesse cenário tecnológico surgiram as primeiras propostas para comunicação digital capaz de interligar tais instrumentos de campo. Contudo são soluções proprietárias específicas de cada fabricante, desenvolvidas para servir a seu próprio mercado consumidor e, devido a utilização de protocolos fechados tais soluções tornaram os equipamentos de fabricantes diferentes incompatíveis. Ou seja, não há preocupação com uma linguagem universal normalizada que permita a verdadeira capacidade de interação entre sistemas de fabricantes diversos, que traduz-se na capacidade de comunicação entre equipamentos de diferentes fornecedores. Devido a esse fato torna-se inevitável e vital a necessidade da elaboração de padrões de comunicação no nível do chão de fábrica que propicie o desenvolvimento de novos produtos com a desejada capacidade de interação.

De acordo com FURNESS (1994) é dentro desse contexto que em meados dos anos 80 surge o conceito de Fieldbus. Seu conceito e aceitação é baseado em um duro trabalho feito por inúmeros grupos de reputação nacional (EUA) e internacional, como ISA (International Society for Measurement and Control), IEC (International Electrotechnical Commission), grupo alemão (PROFIBUS), grupo francês (FIP), e um determinado número de protocolos específicos de diversos fabricantes. Tais grupos desenvolveram um estudo sobre o Fieldbus e lançaram os requisitos que o novo padrão deveria preencher.

Conforme PELLE (1991), o barramento de campo (Fieldbus) é constituído por um sistema de comunicação de dados digital, serial, bidirecional, multiponto,

associado a protocolos de controle de enlace de dados e de interface com o usuário, capaz de promover a comunicação dos instrumentos de campo entre si, bem como com os equipamentos hierarquicamente superiores, aproveitando ao máximo as potencialidades da geração de instrumentos de campo “inteligentes”, proporcionando assim a descentralização das tarefas. Diante destes fatos percebemos que o Fieldbus veio para substituir o padrão de comunicação analógico (4-20 mA) com inúmeras vantagens sobre este último.

Em relação aos trabalhos de normalização temos que, após terem sido iniciados com a criação do Consórcio Internacional de Fieldbus com a junção do grupo da ISA denominado SP-50, do IEC e com a participação da quase totalidade das empresas fabricantes de instrumentos para controle de processos, os trabalhos vinham se desenrolando normalmente até que em Setembro de 1992, devido a morosidade dos trabalhos, houve um “racha” neste consórcio com a criação de três grupos:

- ISA SP-50 / IEC
- ISP (Interoperable System)
- WORLDFIP

Sendo que o grupo ISP era oriundo de empresas que possuíam o protocolo PROFIBUS com alguns itens retirados do protocolo HART (Highway address remote transducer) e o outro grupo vindo do FIP, de empresas de origem francesa, agora estendido para nível mundial através da WORLDFIP.

Portanto a partir desta data passamos a contar com 3 grupos distintos trabalhando de maneira diversa na padronização do protocolo de campo. Em Agosto de 1994 chegou-se novamente a um consenso através da junção do grupo SP-50 da ISA / IEC e dos grupos ISP e de uma parte do grupo WORLDFIP denominado WORLDFIP NORTH AMERICA, sendo que uma outra parte deste grupo o WORLDFIP Europa não se juntou e continua trabalhando em separado, mas com um pequeno grupo de empresas. O cronograma do grupo denominado Fieldbus Foundation, que é um grupo apoiado pela ISA e inúmeras empresas, prevê o término dos trabalhos no início do ano de 1998, vindo portanto ao encontro com a grande expectativa de todos os usuários.

## **1.2 Objetivos da pesquisa**

A garantia de utilização adequada de um Fieldbus numa aplicação do usuário, segundo FRANCO (1995), depende do atendimento aos tempos exigidos por esta aplicação, que por sua vez é função, principalmente, do escalonamento da comunicação entre os dispositivos a ele acoplados. Ou seja, uma questão de suma importância em redes Fieldbus é o escalonamento temporal do controle de processo, para tal é necessário o conhecimento dos tempos envolvidos no mesmo e um algoritmo de escalonamento temporal que otimize a utilização destes.

Esse trabalho tem como proposta, além de fornecer material de referência para um prévio conhecimento a respeito de algoritmos de escalonamento temporal, a implementação de um procedimento de medida de tempo que auxilie o escalonamento temporal para controle de processos em uma rede Fieldbus. Para tal será utilizada uma rede Fieldbus desenvolvida no Laboratório de Eletrônica Embarcada do Departamento de Engenharia Mecânica.

## 2. Revisão Bibliográfica

### 2.1 Transmissão de dados em ambiente industrial

Inicialmente era feita a instrumentação de um experimento científico para obter dados qualitativos e quantitativos que permitissem compreender e modelar o comportamento da natureza ou dos fenômenos observáveis. A medida de determinadas grandezas físicas tais como temperatura, força, deslocamento, velocidade, aceleração, corrente elétrica e outras tantas, permitiu que o homem pudesse propor modelos físicos e matemáticos que traduzissem a realidade. A partir da máquina a vapor construída por James Watt em 1782, a Revolução Industrial passou a exigir o *controle* de determinadas grandezas físicas (variáveis) envolvidas no processo de produção. Desde então a instrumentação e controle têm andado juntos, e em decorrência desse fato surge a importância da transmissão da informação.

O suporte da comunicação de dados em instalações de medição e controle de processos, em antigos sistemas controladores mecânicos, era feito através de sensores e transmissores que estavam geralmente unidos, formando uma unidade capaz de ler e transmitir a grandeza física. Entretanto, com o advento da eletrônica digital surgiram novos sistemas de controle sofisticados onde sensores, atuadores e controladores formam conjuntos separados e distantes entre si, são os sistemas de controle digital, aumentando dessa forma a importância e complexidade dos sistemas de transmissão de dados. Pois “a qualidade de todo o *loop* de controle é diretamente afetada pela precisão dos processos de medida” (O'BRIEN, 1990, p.1937) conseqüentemente temos que ter um meio de transmissão bastante confiável para obtermos maior segurança com relação às medidas realizadas.

#### 2.1.1 Evolução da transmissão de sinais

A evolução da transmissão de sinais teve início com transmissão analógica de sinais realizada através da variação contínua, em relação ao tempo, de uma grandeza física. Tal variação ocorre dentro de uma faixa preestabelecida onde a intensidade do sinal representa o valor da variável de processo transmitida.



“O primeiro padrão analógico surgido foi o sinal pneumático , que variava na faixa de 3 a 15 Psi (20.7 a 103.4 Kpa) representando 0 a 100% da extensão do transmissor” (O'BRIEN, 1990, p.1937), ou seja, correspondendo aos valores máximos e mínimos lidos para as variáveis de processo. Esse sistema de transmissão possuía graves deficiências como o atraso causado em linhas longas de transmissão e os efeitos do volume dos dispositivos terminais, por exemplo uma válvula atuadora. A questão dos atrasos é de vital importância, já que a qualidade do *loop* de controle pode ser prejudicada quando a constante de tempo do sinal transmitido for de significada relevância se comparada com a constante de tempo do sistema em questão.

Com o desenvolvimento dos semicondutores, inicialmente o transistor e depois os circuitos integrados, a eletrônica começou a oferecer equipamentos com baixo consumo e condições de custo/tamanho físico cada vez mais competitivos em relação aos instrumentos pneumáticos ao mesmo tempo em que praticamente resolviam o problema de atraso na transmissão do sinal. Estes transmissores utilizavam sinais elétricos para representar o valor da variável transmitida. As formas mais utilizadas para o sinal elétrico são as seguintes: tensão, deslocamento de fase ou frequência e corrente.

Nos sistemas onde é utilizado a transmissão por tensão o sinal das variáveis do processo é representado por uma tensão contínua. Entretanto, em linhas de transmissão com distâncias relativamente grandes temos a atenuação do sinal elétrico pela impedância da linha de transmissão e, no que diz respeito à sua aplicação em sistemas de aquisição mais complexos ou em plantas de controle cujo ambiente seja marcado pela presença de ruídos eletromagnéticos não é recomendado devido à sua susceptibilidade à ruídos.

Devido a problemas com as distâncias, como citado anteriormente, foram desenvolvidas técnicas de transmissão por frequência e por pulsos. Na transmissão por frequência os dados referentes ao processo são transmitidos por um sinal de amplitude constante modulado em frequência, sendo esta proporcional à variável transmitida. Já na transmissão por pulsos, que podem ser de tensão ou corrente, os dados transmitidos são proporcionais à amplitude, largura ou frequência dos pulsos. As transmissões, tanto para transmissões por frequência como por pulsos, podem ser feitas via cabo, método mais comum, ou por ondas de rádio. A transmissão por rádio é muito utilizada em locais onde não é possível

ou adequado a passagem de fios ou sinais elétricos, como áreas de difícil manutenção ou com perigo de explosão.

Já no caso da transmissão por corrente, que é dentre as transmissões analógicas o método mais difundido e que apresenta melhor desempenho, uma corrente direta proporcional ao valor da variável medida é transmitida. Para tal transmissão foram utilizadas faixas de valores com variações de correntes diversas como; 1 a 5 mA, 0 a 10 mA, 4 a 20 mA e 10 a 50 mA.

“Em 1975 o comitê ISA - SP 50 definiu a faixa de 4 a 20 mA como padrão, sendo adotado pela maioria das indústrias de equipamentos de controle” (BABB, 1986, p.120). Este fato proporcionou uma rápida infiltração no mercado consumidor por parte do padrão anteriormente citado, possibilitando o surgimento de equipamentos compatíveis mesmo sendo de diferentes fabricantes, podendo ser estes utilizados em uma mesma planta de controle. Além dessa podemos também citar outras consideráveis vantagens como: o fato de não haver problemas com a atenuação do sinal pela linha de transmissão, no caso de longas linhas de transmissão; uma menor susceptibilidade a ruídos eletromagnéticos; energia suficientemente baixa para ser utilizado em plantas onde é necessário segurança intrínseca; entre outras. Resolvendo assim problemas anteriormente apresentados pelos outros métodos de transmissão. A interconexão de vários equipamentos a um controlador central é feita através de pares de fios trançados, e ficou conhecida como transmissão por loop de corrente.

É crescente o desenvolvimento de microprocessadores e em consequência disso, ainda na década de setenta, surge a transmissão digital de sinal. Que passa a ser cada vez mais utilizada devido ao fato desta ser superior à transmissão analógica em vários aspectos importantes tais como aponta TANENBAUM (1996): uma taxa de erro muito baixa, possibilidade de multiplexação de vários sinais com um melhor aproveitamento dos equipamentos utilizados, elevadas taxas de transmissão de dados e preços dos equipamentos cada vez mais baixos.

Na transmissão digital os valores correspondentes às variáveis do processo são relacionados a valores preestabelecidos para uma posterior transmissão. Junto com o código em que são representadas as variáveis do processo podem ser incluídas informações adicionais ou redundantes, que em caso de erro na transmissão permitam a detecção deste erro e ainda recuperação correta da informação transmitida. Algumas técnicas foram desenvolvidas para realizar tais

tarefas, podemos citar o “check-sum” e bit de paridade para detecção de erros e o código de Hamming para a recuperação e correção da informação.

A forma mais simples e mais utilizada para realizar a transmissão digital é a codificação binária, onde os sinais transmitidos podem assumir somente dois valores distintos conhecidos como “0” e “1” . As três formas básicas de se transmitir a informação binária são o método por amplitude(amplitude-shift keying, ASK), o deslocamento de frequência (frequency-shift keying, FSK) e o deslocamento de fase (phase-shift keying, PSK). Sendo o método por amplitude a forma mais utilizada, neste os níveis binários representam a ausência ou presença do sinal. O deslocamento de frequência utiliza duas frequências diferentes e não múltiplas e o deslocamento de fase utiliza uma mudança de fase do sinal para representar os mesmos.

Também podem ser utilizados códigos multi níveis para a transmissão de informações, com o sinal podendo assumir três ou mais valores discretos. A razão para a utilização de sinais digitais multi níveis é o ganho de tempo na transmissão e economia da largura de banda da linha de transmissão. Em contrapartida exige uma difícil decodificação e apresenta uma maior susceptibilidade a ruídos.

### **2.1.2 Evolução dos sistemas de transmissão**

Os sistemas de transmissão de sinais até alguns anos atrás utilizavam-se de vários amplificadores de sinal espalhados por toda a malha de controle, esses amplificadores possibilitavam a chegada dos sinais até um controlador central onde este tratava de realizar as correções necessárias nos sinais. Essas correções se tornavam desejáveis devido ao fato, relatado por BROWN (1989), destes sinais serem afetados por mudanças de temperatura, variações na pressão estática e pelas próprias não linearidades do sensor, contribuindo para a ocorrência de erros nas medidas.

No início da década de 80 surgem transmissores dotados de microprocessadores, conhecidos como transmissores inteligentes (smart transmitters). Os primeiros dispositivos deste tipo eram transmissores de pressão, posteriormente vieram os transmissores de temperatura e fluxo magnético. Esses transmissores tinham a capacidade de armazenar dados, obtidos através de cálculos estatísticos refinados, que serviriam de parâmetro para comparação com os dados obtidos dos sensores, ou seja, eles tinham a capacidade de tratar

previamente os erros presentes na variável a ser transmitida fazendo com que esta assumisse um valor mais próximo do valor real, contribuindo para o alívio do controlador central e o aumento da qualidade do controle do processo como um todo. Ou seja, a malha de controle que era realizada no dispositivo central passa a ser executada no dispositivo de controle local. Isto é particularmente sentido em sistemas de médio a grande porte, que podem possuir centenas de medidores, onde o alívio do controlador central é de vital importância.

Paralelamente ao avanço da nova tecnologia surgem as críticas à mesma, que questionam a necessidade da substituição de uma tecnologia já conhecida, e que desempenha de forma satisfatória suas funções, por uma mais cara, devido ao alto custo de seus equipamentos, e que realizava o tratamento dos dados digitalmente mas que os retornam à forma analógica (4 a 20 mA) para uma posterior transmissão.

Naturalmente o desenvolvimento da microeletrônica fez com que o custo dos transmissores inteligentes fosse reduzido a níveis praticamente iguais a de seus correspondentes analógicos. Porém o que se sabe de fato é que o custo total, "incluindo manutenção preventiva, tempo médio entre falhas, tempo de reparo e custos de imprecisão (que influenciam na qualidade do processo e do produto), é mais baixo que nas unidades analógicas" (O'BRIEN, 1990, p.1937).

Com o desenvolvimento desses equipamentos e o surgimento de sensores e atuadores também dotados de "inteligência" inicia-se a implantação de sistemas de controle distribuído (SCD), que passa a distribuir certas funções elementares do algoritmo de controle por toda a planta.

Apesar de todo o avanço proporcionado pelos transmissores inteligentes ainda havia a questão das conversões para os dados que eram enviados do sensor para o transmissor (A/D), tratados e enviados para serem transmitidos (D/A) analogicamente, seguindo o padrão de transmissão de 4 a 20 mA. É dentro desse contexto que surge a segunda geração de transmissores inteligentes, que deu início a um importantíssimo avanço eliminando a transmissão analógica através da utilização de um protocolo digital de transmissão. Ao mesmo tempo em que os sistemas de controle distribuídos (SCD) desenvolveram a possibilidade de entradas de comunicação digital.

Segundo BROWN (1989) dentre as principais melhorias da segunda geração podemos citar três: a precisão de todo o "loop" de controle é dobrada, e não

somente a do transmissor; não há mais necessidade de redefinição manual dos valores de operação do transmissor toda vez que os parâmetros de operação do sistema são mudados, pois os novos valores podem ser transferidos do controlador ou de um terminal de operação para o transmissor; e finalmente, a transmissão digital permite que o sistema de controle monitore continuamente a validade da variável medida.

Conforme citado por BROWN (1989) no que diz respeito à diminuição de erros temos ainda que, com a remoção do hardware de conversão analógica é reduzida à metade a sensibilidade à variações de temperatura, fator gerador de boa parte dos erros em muitas aplicações. A possibilidade de monitoração do transmissor aumentou a confiabilidade e a segurança no sistema, já que passa a ser possível a imediata detecção e correção de falhas.

Logo a seguir mostraremos os ganhos obtidos na transmissão e no loop de controle, em relação à diminuição do erro, com a evolução dos sistemas de transmissão (transmissor/padrão de transmissão):

Sistema de transmissão		Precisão (% erro)	
Transmissor	Padrão	Transmissão	Loop
analógico	analógico	0.150	0.225
digital	analógico	0.075	0.150
digital	digital	0.075	0.075

TABELA 2.1 Evolução dos sistemas de transmissão e ganhos na precisão.

Para implementação do sistema o tempo gasto na especificação do instrumento foi sensivelmente diminuído devido à maior faixa de operação dos transmissores de segunda geração. Esta maior faixa de operação (range) é também de grande valia devido às correções necessárias ao projeto e que são executadas na prática.

Na questão da configuração, BROWN (1989) cita que durante a utilização dos primeiros transmissores inteligentes era necessário que fossem configuradas separadamente as bases de dados do SCD e do transmissor. Com os transmissores de segunda geração esse problema deixou de existir pois o transmissor é incorporado à base de dados do SCD. Ao se instalar o transmissor é feito o carregamento de sua base de dados através da via de comunicação. Esta

integração da base de dados permite também ao operador/engenheiro a verificação de toda a base de dados do transmissor e não apenas parte dela. Podem também ser feitas mudanças nas funções do transmissor, com o armazenamento destas para recuperação posterior.

Com relação à operação podemos citar alguns benefícios tais como possibilidade de configuração da faixa de operação e diagnóstico remotos. Possibilidade de transmissão de uma segunda variável, podendo ser outra variável do processo ou mesmo dados referentes ao equipamento como temperatura interna, umidade, etc.. Bem como a proteção contra base de dados inadequada que é de grande valia durante processos de expansão do sistema de controle, quando novos transmissores são adicionados a este.

Como percebemos existe também uma redução significativa no tempo gasto com a manutenção devido a possibilidade de definição dos parâmetros de operação do transmissor a partir de uma estação de controle remota.

“O melhor desempenho e características dos transmissores inteligentes de segunda geração em relação a seus antecessores de primeira geração e aos analógicos, resultam em menores custos de implementação, operação e manutenção” (BROWN, 1989, p.1625).

A seguir mostraremos a evolução dos sistemas de transmissão, dos primeiros sensores analógicos aos transmissores inteligentes de segunda geração:

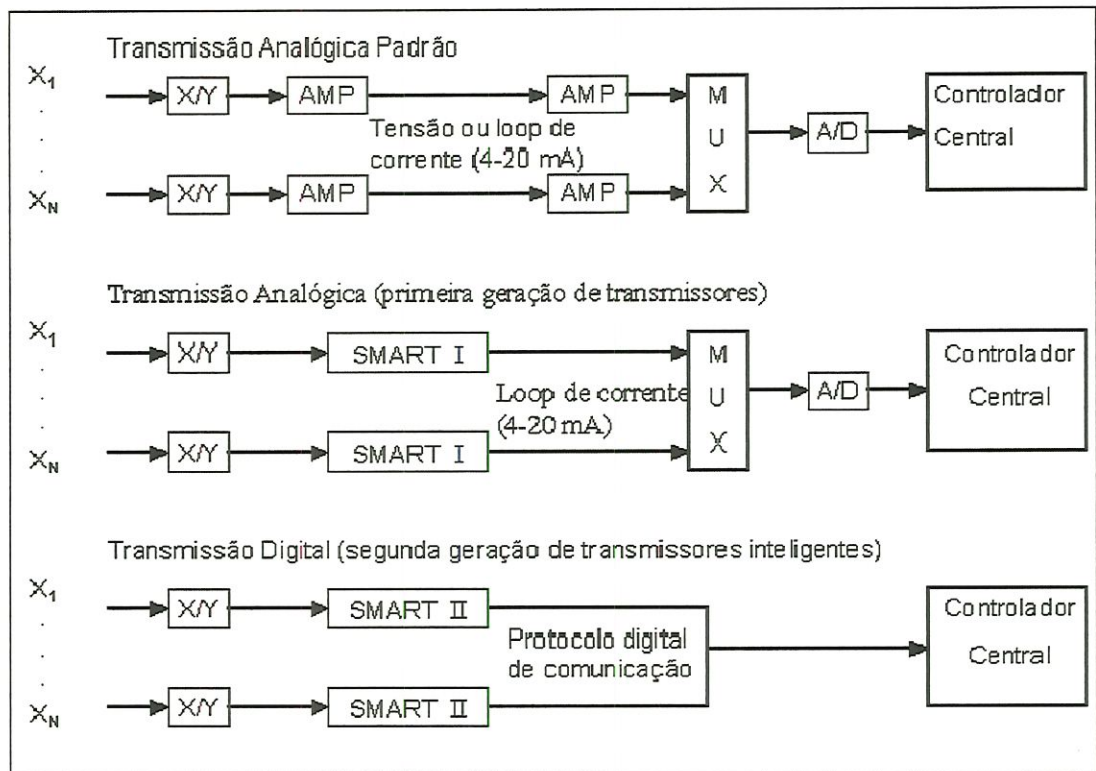


FIGURA 2.1 Evolução dos sistemas de transmissão em chão de fábrica.

## 2.2 Controle, do clássico ao moderno

“O controle automático tem representado um papel vital no avanço da engenharia e da ciência. (...) tornou-se uma parte importante e integral dos modernos processos industriais e de fabricação” (OGATA, 1993, p. 1).

Da máquina a vapor de James Watt (1782) aos dias atuais o controle tem evoluído bastante. Tendo experimentado várias descobertas durante esse período, segundo FRANKLIN (1994), como o primeiro estudo sistemático da estabilidade de controle em “*feedback*” que foi apresentado no artigo “On Governors” por J. C. Maxwell (1868), conseguindo e E. J. Routh determinar o critério de estabilidade em 1877. Logo depois da apresentação do trabalho de Routh, o matemático russo A. M. Lyapunov (1883) inicia estudos sobre a questão da estabilidade de movimento, seus estudos foram baseados em equações diferenciais não lineares que são equivalentes ao critério de Ruth. O trabalho de Lyapunov foi fundamental para o que é agora conhecido como variável de estado para teoria de controle.

A análise da equação característica continua sendo a base da teoria de controle até a invenção do “*electronic feedback amplifier*” por H. S. Black em 1927.

Pois, com a introdução de amplificadores eletrônicos, nas décadas seguintes à primeira guerra mundial tornou-se possível o uso de longas linhas telefônicas. Em consequência disso aumentou-se os diâmetros dos cabos e as perdas de energia, sendo necessário vários amplificadores para repor a energia perdida resultando em muitas distorções devido a pequenas não linearidades dos tubos de vácuo usados nos amplificadores. Foi aí que Black propôs o “*feedback amplifier*” utilizando-se do conceito de resposta em frequência e de variáveis complexas voltando-se para a análise em frequência. Em 1932 H. Nyquist publica um artigo descrevendo como determinar estabilidade a partir de um gráfico da resposta em frequência de malha fechada.

Simultaneamente ao desenvolvimento do “*feedback amplifier*”, o controle com realimentação de processos industriais vem a se tornar padrão. Neste cenário, caracterizado por processos que não somente são altamente complexos como também não lineares e sujeitos a grandes defasamentos no tempo entre sensores e atuadores, desenvolve-se o controle proporcional-integral-derivativo (PID). Durante a segunda guerra mundial foi dado um grande impulso no campo do controle e desenvolve-se técnicas de projetos de **servomecanismos**, é como os mecanismos de controle passam a ser chamados. Em 1948 W. R. Evans introduz um outro enfoque para o projeto de sistemas de controle, o método conhecido como lugar das raízes (*root locus*) é apropriado para projeto bem como para análise de estabilidade e permanece como uma importante técnica até hoje.

“Os métodos de resposta em frequência e lugar das raízes, que constituem o coração da teoria de controle clássica, levam a sistemas que são estáveis e satisfazem a um conjunto de requisitos de desempenho mais ou menos arbitrários” (OGATA, 1993, p. 1).

“Em virtude dos modernos processos (*plants*) com muitas entradas e saídas tornarem-se cada vez mais complexos, a descrição de um sistema de controle moderno requer um grande número de equações. A teoria de controle clássica, que trata apenas de sistemas de entrada simples e saída simples, torna-se impotente para os sistemas de entrada múltipla e saída múltipla” (OGATA, 1993, p. 1).

Segundo FRANKLIN (1994) durante os anos 50 muitos autores, incluindo R. Bellman e R. E. Kalman nos E.U.A. e L. S. Portryagin na U.R.S.S., retomam as equações diferenciais parciais (EDO) como um modelo para sistemas de controle.



Muitos desses trabalhos são estimulados pelo novo campo do controle de satélites artificiais em volta da Terra, para o qual a EDO é uma forma natural para se escrever o modelo. Esses novos modelos tiveram o suporte dos computadores digitais. Assim o trabalho de Lyapunov foi passado para uma nova e possível linguagem de controle, e os estudos de controle ótimo, iniciados por Wiener e Phillips durante a segunda guerra mundial, foram estendidos para otimização de trajetórias de sistemas não lineares baseados nos cálculos de variações. Este método não usa a resposta em frequência ou a equação característica mas trabalha diretamente com a EDO na forma “normal” ou de “estado” e normalmente utiliza-se de computadores para realização dos cálculos. Apesar das bases do estudo de EDOs terem iniciado no século passado, esse enfoque é conhecido como Controle Moderno para distingir-se do Controle Clássico.

### 2.3 Controle em rede

Uma fábrica automatizada é normalmente composta de várias células, cada uma contendo robôs, sensores, e mecanismos de transportes. Um barramento de multi-acesso conecta todos os equipamentos na célula. *Bridges* (pontes) então conectam múltiplas células para formar a malha de controle. Portanto, a rede de uma fábrica consiste de vários *links* (conexões), onde cada um é normalmente um barramento multi-acesso. Segundo SHIN (1996) a habilidade de proporcionar previsíveis comunicações interprocessos é de grande significado para uma fábrica, porque atrasos imprevisíveis na comunicação devem levar a perdas no *deadline* (tempo de resposta máximo de cada processo) de uma ou mais tarefas de comunicação que coletivamente monitoram e controlam equipamentos de manufatura e processos.

De acordo com LUTZE (1995) o *backbone* (tronco central) da maioria dos sistemas de controle é o Controlador Lógico Programável (CLP). Ao mesmo tempo que ele é capaz de lidar com tarefas de controle, tem como grande desvantagem conter todos os pontos de E/S (entrada e saída)<sup>#</sup> dentro do centro de controle. Numerosas conexões, aglomerados de cabos e fios devem passar para terminação de todos os componentes de E/S. Isso acaba acarretando maior consumo de tempo, trabalho e espaço. Deve-se frisar que apesar de termos no

CLP um equipamento digital, sua comunicação com sensores/atuadores é feita através de um protocolo analógico (4-20 mA).

### 2.3.1 Controle em rede SDCD

Os Sistemas Digitais de Controle Distribuído (SDCD) possuem estações de controle local (ECLs) interligados por uma rede de comunicação digital (com um protocolo proprietário) e, apesar de termos equipamentos digitais espalhados por toda a planta, como sensores/atuadores inteligentes (microprocessados), transmissores, etc. a comunicação entre eles e as ECLs fica por conta de um padrão analógico de 4-20 mA. Ou seja, os sensores/atuadores realizam a medição e manipulação digital dos dados e em seguida convertem o sinal para analógico que é transmitido às ECLs, ver figura 2.2, nas quais o sinal é novamente convertido para a forma digital. Nessas estações são realizados, através de algoritmos de controle das variáveis de processo, as alterações necessárias para correção do sinal. O sinal corrigido é mais uma vez convertido para analógico e é transmitido ao campo para alterarem, através dos elementos finais de controle, os valores das variáveis manipuladas. Só pode ser enviada uma única informação num único sentido do transmissor para o controlador e vice-versa.

Os valores das variáveis, oriundos dos sensores/atuadores, armazenados na memória das ECLs trafegam através da rede de comunicação digital onde podem ser manipulados através de softwares de supervisão. Podendo, desta maneira, retornar às ECLs valores recalculados das variáveis para se obter um melhor nível de otimização da planta.

Com isso PEREIRA (1995) observa que, o termo distribuído se refere às diversas ECLs que compõem o sistema (praticamente uma ECL por malha de controle) mas na realidade apesar da existência dos transmissores inteligentes, alguns deles com capacidade de controle, esta característica não é explorada e, portanto temos um sistema centralizado na disposição das ECLs por conseguinte não se utiliza a distribuição do controle no campo.

---

# mais conhecidos como I/O points (input/output).

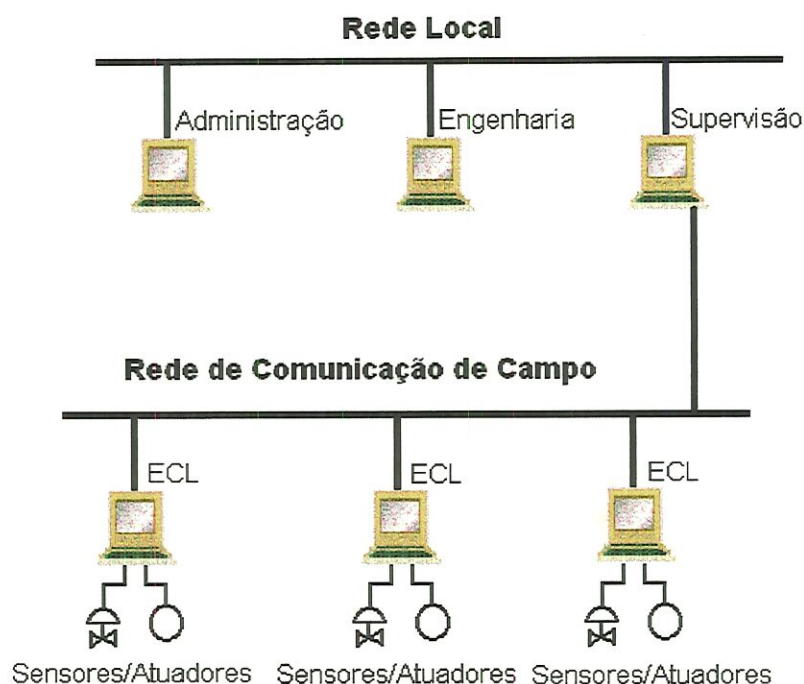


FIGURA 2.2 Sistema Digital de Controle Distribuído.

### 2.3.2 Controle em rede de campo Fieldbus

Fieldbus é um típico exemplo de uma verdadeira rede em tempo real. De acordo com RAJA (1993) o barramento de campo, mais conhecido como Fieldbus é uma rede local de tempo real do nível mais baixo na hierarquia da automação de fábrica, ligando os dispositivos de campo aos dispositivos de controle através de uma topologia de barramento, onde todas as informações são transmitidas serialmente e multiplexadas no tempo.

Com o Fieldbus foram conseguidas algumas vantagens que listaremos a seguir:

- significativa redução dos custos de cablagem e interfaces, já que haverá um cabo (barramento) principal ao qual são ligados vários instrumentos bem como eliminação de dispositivos A/D e D/A;
- facilidade na instalação, já que temos uma topologia multidrop ao invés de ponto a ponto;
- rapidez na inicialização, operação, e manutenção do sistema: detecção de falhas, diagnósticos, reparos .Devido ao fato do Fieldbus possibilitar a

realização de tarefas on-line. Informações disponíveis em qualquer ponto do sistema;

- redução do tamanho do sistema de controle principal, possibilitando o deslocamento para o campo de alguns algoritmos e processos de controle (P.e.: processamento de sinais);
- permissão para comunicação direta entre os instrumentos de campo e deste com os instrumentos de controle;
- melhoria do desempenho da transmissão digital em relação a analógica;
- expansão e evolução mais flexível, bastando conectar um novo instrumento à rede e configurá-lo.

O sistema de controle passa a ser verdadeiramente distribuído, devido ao fato de que não necessita-se de ECLs para servir de intermediadoras entre os instrumentos de campo e a estação de supervisão, já que temos um protocolo puramente digital dando uma maior autonomia aos equipamentos “inteligentes” (sensores/atuadores, transmissores, etc.), a seguir mostraremos um esquemático de um sistema Fieldbus.

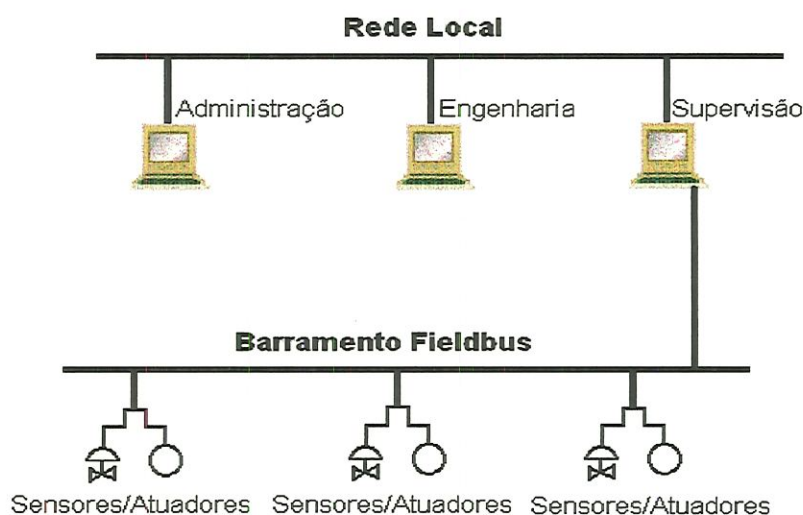


FIGURA 2.3 Sistema digital de controle Fieldbus

## 2.4 Redes locais industriais

Com o acelerado desenvolvimento das técnicas de comunicação digital, a partir da década de 70, começaram a surgir as primeiras redes locais de computadores.

Mc GUFFIN\* apud AGUIAR (1991) acrescenta que, particularmente na indústria, as redes locais têm tido penetração cada vez maior e isto se deve a vários fatores, entre eles a filosofia CIM (Computer Integrated Manufacturing). Integrar uma manufatura por computador significa interligar todos os sistemas computadorizados existentes na empresa, incluindo: controladores lógicos programáveis (CLPs), dispositivos de comando numérico computadorizado (CNCs), dispositivos de comando de robôs (CNRs), etc. , além da área administrativa e de projetos.

Os problemas associados com a integração operacional ocorrem principalmente em relação à interligação de equipamentos de diferentes fabricantes, uma constante na indústria. Isso tornou essencial o desenvolvimento de projetos que estabelecessem um padrão único de troca de mensagem num ambiente fabril. O uso de padrões internacionais de comunicação torna-se a solução mais adequada a ser adotada pelos fabricantes quando da integração de meios computacionais e de equipamentos de tecnologias diferentes.

Os esforços de padronização de redes locais passam a se defrontar com dois problemas de natureza distinta. De um lado, a existência de propostas concorrentes, com motivações de natureza técnica e econômica. Do outro, a rápida evolução tecnológica, dificultando a definição dos padrões a serem adotados.

Tais problemas começam a ser resolvidos pela International Standards Organization (ISO), formada por organizações nacionais de padronização de 89 países dentre as quais estão; a A.N.S.I. (Estados Unidos), B.S.I. (Grã-Bretanha), A.F.N.O.R. (França) e D.I.N. (Alemanha). Quando esta, segundo MENDES\* apud SILVA (1996), publicou no final dos anos 70 e início dos anos 80 o chamado Reference Model Open System Interconnection/ISO (RM-OSI/ISO) e que simboliza hoje os dez mandamentos da comunicação digital, sendo aceito pela indústria de

---

\* Mc GUFFIN, L.J., et al. MAP/TOP in CIM distributed computing. *IEEE Network*, v.2, n.3, May, 1988 apud AGUIAR, M.V.C., et al. ASPECTOS DE Implementação de um servidor MMS. Florianópolis, Imprensa Universitária da Universidade Federal de Santa Catarina, 1991. p.331.

\* MENDES, M.J.; MAGALHÃES, M. Redes locais na comunicação fabril *Automação & Indústria*, dez. , 1988 apud SILVA, R.D.da *Comunicação de dados em ambiente industrial: um protocolo para automação e controle em tempo real*. São Carlos, 1996. Dissertação (Mestrado), Escola de Engenharia de São Carlos, Universidade de São Paulo.

computadores para a definição de arquiteturas de redes de comunicação, tornando-se também base de referência nas diversas propostas de padrões de redes locais para o ambiente industrial. Pois é baseado neste modelo de referência que surgem protocolos de comunicação como o MAP, Fieldbus, etc.

A seguir mostraremos um exemplo de estrutura de comunicações em uma rede local industrial:

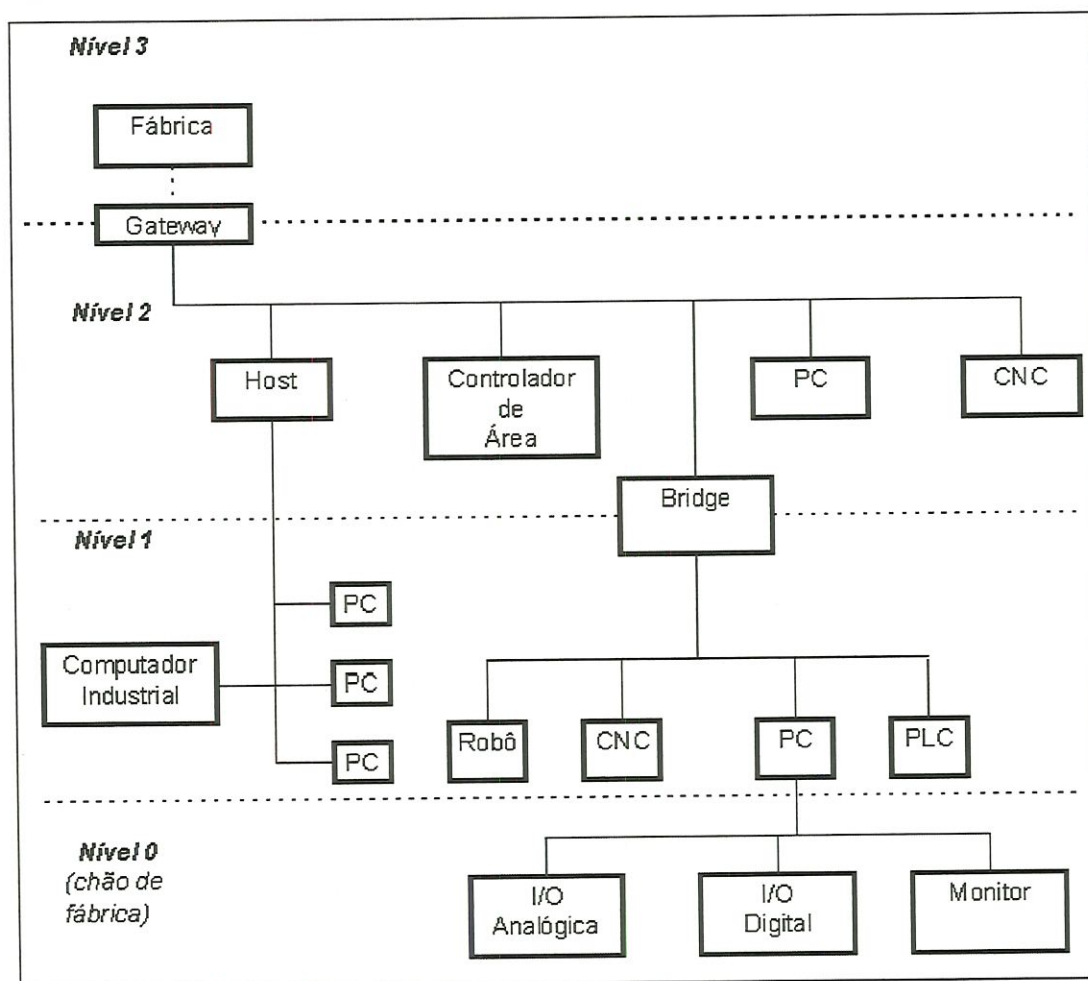


FIGURA 2.4 Exemplo de hierarquia para comunicação industrial.

Pode ser visto na figura anterior a aplicação de uma LAN (Local Area Network) industrial mostrando a organização hierárquica de um sistema de comunicação industrial com seus respectivos níveis, os quais são descritos, sucintamente, a seguir:

- no nível corporativo, nível 3, a comunicação existente diz respeito principalmente ao processamento de dados administrativos como; encargos, contas, decisões empresariais, etc.

**Características:**

- grandes quantidades de dados
- baixa velocidade
- grandes distâncias
- no nível 2 é onde acontece a comunicação dos vários sistemas de computadores centrais dentro da corporação, neste ambiente é realizado o calendário de produção, gerenciamento de material, custos, etc.

**Características:**

- grandes quantidades de dados
- baixa velocidade
- distância depende do protocolo escolhido
- no nível 1 realizam-se os processos, acontece a comunicação entre as diversas unidades de controle como: PC's industriais, CLPs e outros controladores industriais.

**Características:**

- pequenas quantidades de dados
- alta velocidade
- distância depende do protocolo utilizado
- No chão de fábrica, nível 0, é onde se realiza a comunicação entre os sensores/atuadores e os equipamentos de controle (CLPs). Os dispositivos desse nível trabalham em conjunto com os dispositivos do nível 1, daí necessitarem de tempo de resposta reduzido e com valor máximo garantido, facilidades de varredura cíclica e acíclica e pacotes de dados curtos/otimizados.

**Características:**

- pequenas quantidades de dados
- alta velocidade
- pequenas distâncias

No caso dos níveis superiores não existe uma grande preocupação com o tempo de resposta e sim uma grande transferência de volume de dados roteamento entre redes e sub-redes, estabelecimento de conexões, etc. De acordo com SANTOS (1996) essas características são, do ponto de vista dos protocolos, facilidades computacionais e semânticas bem diferentes daquelas indicadas no

chão de fábrica e que são determinantes no tipo de rede de comunicação que deve ser utilizada.

### **2.4.1 O modelo de referência OSI/ISO**

Segundo BEAUCHAMP (1990) este é o mais importante e extensamente aceito sistemas de camadas para comunicação em uso atualmente; é um ponto de vista da filosofia de interconexão conhecido como interconexão de sistemas abertos. Sistemas abertos são simplesmente aqueles que são capazes de interconexão em virtude de cada um ter implementado um conjunto comum de protocolos. OSI (Open Systems Interconnection) refere-se à comunicação entre sistemas de computadores que são capazes de trocar informação livremente independente do tipo e do fabricante, em virtude deles mutuamente aderirem a um conjunto de normas. As normas devem ser descritas em referência a um modelo de arquitetura de um sistema de comunicação.

O modelo OSI tem sete camadas, e os princípios que foram aplicados para se chegar às sete camadas foram os seguintes de acordo com TANENBAUM (1996):

- uma camada deveria ser criada onde um nível diferente de abstração é necessário.
- cada camada deveria realizar uma função bem definida.
- a função de cada camada deveria ser escolhida com o intuito de definir um protocolo padronizado internacionalmente.
- os limites das camadas deveriam ser escolhidos para minimizar o fluxo de informação através das interfaces.
- o número de camadas deveria ser suficientemente grande para que funções distintas não fossem passadas juntas para a mesma camada sem necessidade, e pequena o bastante para que a arquitetura não viesse a ser inconveniente.

As camadas do modelo RM-OSI/ISO são as seguintes:

- 1- Camada Física
- 2- Camada de Enlace de Dados
- 3- Camada de Rede
- 4- Camada de Transporte



5- Camada de Sessão

6- Camada de Apresentação

7- Camada de Aplicação

“A idéia básica da estruturação em camadas do RM-OSI é a de que cada uma das sete camadas fornece serviços de comunicação, com um certo grau de confiança, à chamada imediatamente superior. Os serviços fornecidos pelas camadas inferiores são sucessivamente ampliados e aperfeiçoados, de modo que na camada mais superior (Camada de Aplicação ou camada 7) seja oferecido um conjunto de serviços adequado para suportar as diversas aplicações” (GIOZZA, 1986, p.39).

A seguir faremos uma rápida descrição de cada camada:

### **1- Camada Física**

A camada física define as características mecânicas, elétricas, funcionais e de procedimentos a fim de ativar, manter e desativar conexões físicas para a transmissão de bits entre as entidades da camada superior (camada de enlace de dados). Esta é a camada de nível mais baixo na arquitetura RM-OSI, e está diretamente associada aos meios físicos, interconectando os sistemas. A camada física ocupa-se, por exemplo, da regulamentação das interfaces mecânicas (número de pinos dos conectores, dimensões, etc.), das interfaces elétricas (níveis de tensão, corrente, etc.), das interfaces lógicas (dados, temporização, etc.) e dos procedimentos de comando e gerência das conexões físicas (liberação, modo de funcionamento, etc.)

### **2- Camada de Enlace de Dados**

Define as características funcionais e de procedimentos a fim de estabelecer, manter e liberar conexões de enlace de dados entre as entidades da camada superior (camada de rede). O objetivo principal desta camada é o de detectar e, possivelmente, corrigir os erros que podem ocorrer na troca de informações (bits) a nível da camada física. A camada de enlace de dados também fornece à camada superior a capacidade de pedir o estabelecimento ou liberação dos meios físicos associados á camada física. A unidade de transferência de informação na camada de enlace de dados corresponde a uma seqüência de bits sob a forma de quadros ou blocos (frames).

### **3- Camada de Rede**

Esta camada define as características funcionais e de procedimentos a fim de estabelecer, manter e liberar conexões de rede entre entidades da camada superior (camada de transporte). A função principal desta é a escolha da melhor rota a ser percorrida pelo pacote de informações da origem ao destino. As rotas podem ser baseadas em tabelas estáticas, que são armazenadas na memória de cada estação e raramente trocadas, ou podem ser dinâmicas, determinadas no início de cada conversaç o. Tem como funç es tamb m determinaç o do tamanho dos pacotes de dados, bem como sua segmentaç o, caso necess rio, fornece endereços no n vel de rede para que as entidades da camada de transporte possam se identificar mutuamente.

### **4- Camada de Transporte**

A camada de transporte prov  os meios para estabelecer, manter e liberar conex es de transporte para transfer ncia de dados bidirecional entre um par de entidades (processos) da camada superior (camada de sess o). A funç o b sica da camada de transporte   aceitar dados da camada de sess o, fragment -los em unidades menores se necess rio, pass -los   camada de rede, e garantir que todos os pedaços cheguem corretamente ao outro lado. Segundo TANENBAUM (1996) mais ainda, tudo isso deve ser feito eficientemente, e de uma maneira que isole as camadas superiores de inevit veis trocas na tecnologia de hardware.

### **5- Camada de Sess o**

Esta camada realiza o estabelecimento, controle e terminaç o de di logos entre processos de aplicaç o. Ela realiza a sincronizaç o de di logos bem como o gerenciamento de tokens. A sincronizaç o permite que sejam colocadas marcas durante a transfer ncia dos dados permitindo, desta forma, a identificaç o do ponto de parada da transfer ncia em caso de queda do sistema. Com o token   identificada a estaç o que pode acessar ao meio durante um intervalo de tempo preestabelecido. Parecido com uma corrida com bast es, quem estiver com o token tem acesso, sendo este passado de uma estaç o para outra.

## **6- Camada de Apresentação**

Essa camada resolve diferenças na representação da informação usadas na tarefa de aplicação fazendo com que cada computador do sistema possa se comunicar sem conhecer a representação da informação usada pelo outro, isto é, usa diferentes códigos para representar o mesmo dado. A camada de apresentação opera convertendo os dados específicos e operações requeridas por cada usuário para um padrão preestabelecido entre os parceiros, possibilitando que um entenda as informações provenientes do outro. Realiza também compressão de dados diminuindo assim o número de bits a ser transmitido, e serviços de criptografia para garantir a privacidade e autenticidade de determinados dados.

## **7- Camada de Aplicação**

Esta é a mais alta camada no modelo de referência e tem como objetivos principais fornecer informações sobre as funções do processo e programas para os processos de aplicação, ou seja ela atua como uma interface entre o processo aplicativo e a rede. Ela contém uma variedade de protocolos que são comumente utilizados para proporcionar compatibilidade entre os vários sistemas existentes, como por exemplo a diferença entre os vários tipos de terminais existentes, assim como transferência de arquivos, correio eletrônico, consulta a diretórios, etc.

“O modelo de referência RM-OSI/ISO visto como um sistema, pode ser dividido em dois grupos. As três camadas de base (física, enlace e rede) cobrem os componentes da rede utilizados para transmitir as mensagens. As três camadas superiores, entretanto, refletem as características dos sistemas em comunicação. Suas funções não levam em consideração os aspectos físicos da comunicação. A camada de transporte atua como uma ligação entre as camadas de sistema (camadas superiores) e as camadas de rede (camadas inferiores)” (SILVA, 1996, p.34).

Em seguir mostraremos um exemplo (figura 2.5) de como podem ser transmitidos dados usando-se o modelo RM-OSI/ISO. Temos dois processos, uma máquina transmissora e outra máquina receptora de dados, o processo transmissor tem alguns dados que precisa enviar para o processo receptor. Ele envia os dados para a camada de aplicação e esta anexa ao bloco de dados um

cabeçalho (header), AH (application header), e o envia para a camada de apresentação. A camada de apresentação realiza o tratamento necessário ao pacote de dados e adiciona um cabeçalho, PH (presentation header), passando o resultado à camada de sessão. É importante lembrar que a camada de apresentação não é informada sobre qualquer parte dos dados passados para ela pela camada de aplicação, pois cada camada trata o pacote de dados passado a elas como um todo sem considerar as partes, por exemplo não interessa à camada de apresentação qual parte do pacote representa o cabeçalho da camada superior (application header).

Esse processo é repetido até os dados alcançarem a camada física, onde eles são efetivamente transmitidos para a máquina receptora. Nessa máquina os vários cabeçalhos são retirados, um a um, à medida que o pacote de dados vai passando através das camadas, da física à de aplicação, até alcançar o processo receptor.

Convém frisar que, apesar da transmissão real dos dados acontecer na vertical, cada camada é programada como se a transmissão ocorresse na horizontal. Ou seja, quando a camada de transporte da máquina receptora recebe a mensagem da camada de sessão, ela adiciona o seu cabeçalho e envia para a camada de transporte da máquina receptora.

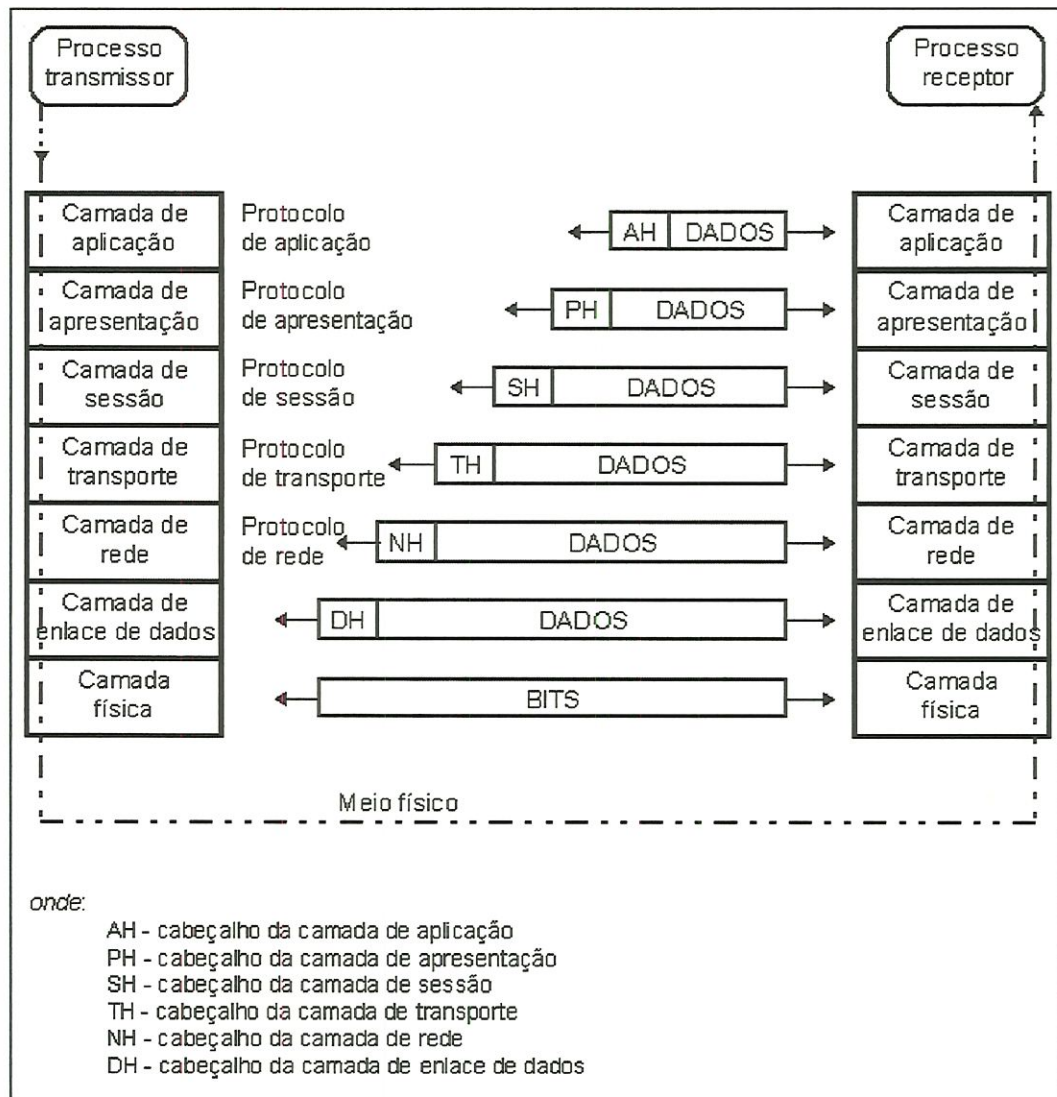


FIGURA 2.5 Transferência de dados no modelo OSI/ ISO.

## 2.4.2 Arquitetura MAP

De acordo com BEAUCHAMP (1990) a arquitetura de comunicação MAP (Manufacturing Automation Protocol) é o resultado de muitos anos de atividade realizado por um grupo de trabalho iniciado pela General Motors Inc. no início da década de 80. O objetivo inicial era proporcionar um meio multi canal para transmitir simultaneamente sinais de TV, áudio e mensagens de dados. Devia fazer com que qualquer computador, independente de seu fabricante, pudesse ser conectado à essa rede, tornando-se assim um padrão para a comunicação no ambiente industrial.

Este foi o primeiro padrão baseado no modelo RM-OSI/ISO amplamente aceito para redes de automação e controle. O intuito do MAP era selecionar um conjunto de padrões para uma LAN (Local Area Network) na fábrica e especificar interfaces de modo que suportasse comunicações entre grandes computadores e equipamentos para controle de manufatura. Segundo DAMSKER (1988) o MAP atende funções de CIM (Computer Integrated Manufacturing) tais como o controle de manufatura, planejamento, controle de material, gerenciamento de informação, e a rápida conexão entre projeto e produção. Como cita McGUFFIN (1988) isto ficou evidente com os padrões adotados na camada de aplicação - a Manufacturing Messaging Specification (MMS) - destinados à comunicação no chão de fábrica entre dispositivos programáveis, controladores de células e computadores de área/seção.

As características especificadas para algumas das sete camadas do modelo OSI/ISO são descritas a seguir:

A camada física, responsável em estabelecer as características elétricas da transmissão (velocidade, níveis de sinais e propriedades mecânicas e elétricas dos conectores), codificar a informação e transferi-la pelo meio físico. Para esta arquitetura foram adotados dois padrões da especificação IEEE 802.4 ("token bus"):

- cabo coaxial, transmissão em banda larga com taxa de 10 Mbps para o tronco central da rede ("backbone"), em virtude das características: baixa atenuação do sinal, baixa interferência eletromagnética e capacidade de transmissão simultânea de dados, voz e imagem. Conhecido como "broadband".
- cabo coaxial, transmissão em banda portadora com taxa de 5 Mbps para níveis inferiores da fábrica, em virtude da simplicidade e do custo. Conhecido como "carrierband".

A camada de enlace de dados responsável em transmitir informações entre dois nós adjacentes da rede com acesso a meios de transmissão compartilhados além da detecção e correção de erros. As sub-camadas pertencentes a este nível foram definidas da seguinte forma:

- LLC ( Link Control) - adota serviços do tipo 1 ("connectionless"), não orientado à conexão e sem reconhecimento, pois a camada de transporte

suporta as funções de seqüenciamento, controle de fluxo e recuperação de erros.

- MAC (Media Access Control) - utiliza o padrão IEEE 802.4 para garantir a possibilidade de utilização de prioridades e a não ocorrência de colisões em modo normal.

Na camada de rede, responsável pelo roteamento dos pacotes através dos nós da rede mantendo uma qualidade de comunicação apropriada para a transmissão, optou-se pelo padrão ISO, o 8473 ( Connectionless-Mode Network Service ), serviço não orientado à conexão. O mesmo ocorre na camada de transporte para a qual foi adotado o padrão ISO para oferecimento de serviços, o 8073, com serviços e protocolos definidos e que providenciam o controle de fluxo, multiplexagem de dados e detecção de erros. Similarmente à camada de transporte a camada de sessão e a camada de apresentação utilizam protocolos baseados em padrões ISO, o 8327 e o 8823 respectivamente.

Na camada de aplicação foi adotado um padrão não OSI mas padronizado pela ISO, o Manufacturing Message Specification (MMS), que padroniza a comunicação para o controle e monitoração de equipamentos industriais e especifica a semântica da comunicação em automação industrial. Segundo MESSINA (1990), mostra-se apropriado para a troca de mensagens curtas em sistemas com requisitos de resposta rápida.

Esta arquitetura MAP com sete camadas consistindo um modelo OSI resultou em uma rede com alto grau de compatibilidade e uma faixa abrangente de aplicações, porém com tempo de resposta relativamente elevado para aplicações de controle de processos, devido ao grande processamento envolvido.

Foram propostas então novas arquiteturas, a MAP/EPA e a mini-MAP. A configuração MAP/EPA (MAP/Enhanced Performance Architecture) consiste de um nó MAP com melhor desempenho em termos de tempo de resposta. De acordo com MESSINA (1990) esta melhoria é conseguida através da inserção, junto aos serviços OSI, de protocolos de resposta rápida que eliminam algumas das camadas superiores do modo OSI.

### 2.4.2.1 Projeto MAP/EPA

A arquitetura MAP/EPA segue o proposto pelo projeto MAP com algumas alterações. A camada física realiza transmissão de 5 Mbps e modulação em banda portadora, permitindo o aumento do desempenho ocasionados pela não utilização de alguns protocolos OSI definidos no MAP.

O projeto suporta tanto as sete camadas especificadas na arquitetura MAP completa, quanto uma arquitetura reduzida que visa a comunicação com restrições de tempo. Neste caso, a camada de aplicação se liga diretamente com a camada de enlace que provê serviços de reconhecimento imediato (LCC IEEE 802.2 tipo 3). Com o reconhecimento imediato, o nó receptor reconhece imediatamente o quadro sem esperar pelo token. O nó emissor mantém o controle sobre o meio de transmissão por tempo suficiente para que o nó remoto reconheça o quadro e responda corretamente. Essa configuração resultou em um nó com maior complexidade e custo mais elevado, pois tem um protocolo (completo) para aplicações convencionais e outro (reduzido) para aplicações com tempo de resposta crítico. A seguir será mostrado o esquemático do projeto MAP/EPA.

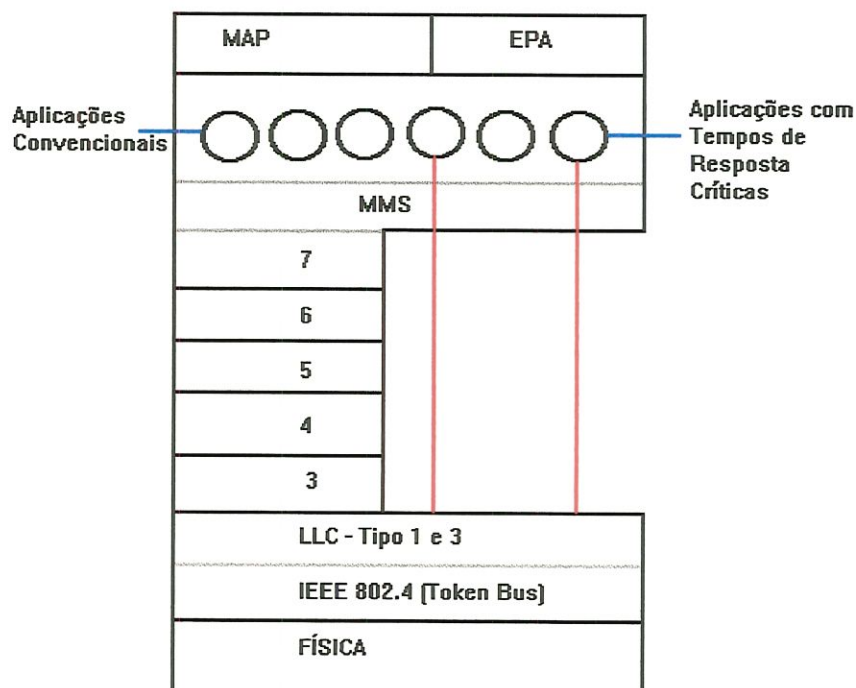


FIGURA 2.6 Arquitetura MAP/EPA.



Os requisitos gerais desta arquitetura são os seguintes:

- baixos tempos de respostas para mensagens curtas de alta prioridade;
- alta confiabilidade do meio e do método de sinalização;
- facilidade de configuração;
- fácil conexão com a rede principal da fábrica;
- segurança de acesso para evitar conexões não autorizadas.

Algumas restrições impostas pela simplificação da arquitetura devem ser observadas:

- perda da garantia da qualidade de entrega das mensagens;
- limitação do tamanho máximo das mensagens de acordo com o tamanho máximo suportado pela unidade de dados do protocolo de enlace;
- impossibilidade de ressincronizar diálogos entre aplicações ou prover checkpoints pela inexistência da camada de sessão;
- e, como citado anteriormente, o alto custo do nó de comunicação.

#### **2.4.2.2 Projeto Mini-MAP**

A configuração mini-MAP implementa um conjunto de protocolos reduzidos e capaz de prover comunicações no nível de controle direto, com tempo de resposta e segurança exigidos pela aplicação a um custo compatível. A arquitetura mini-MAP contém somente três camadas (física, de enlace e de aplicação) da estrutura EPA-MAP, conseqüentemente não é compatível com uma rede MAP, e o único protocolo utilizado na camada de aplicação é o MMS.

Segmentos de nós Mini-MAP exigem um gateway (ou um nó EPA-MAP) para a conexão com as redes MAP. Os nós Mini-MAP são utilizados somente para a conexão de dispositivos simples ( terminais, bar-code readers, etc. ).

As camadas 3, 4, 5, 6 podem ser omitidas pelos seguintes aspectos:

- camada de apresentação: o protocolo MMS já define a sintaxe das mensagens;
- camada de sessão: não é necessário ressincronização do diálogo pois a comunicação entre os dispositivos é simples;
- camada de transporte: quantidade de erros existentes na comunicação entre dispositivos da mesma sub-rede é pequena;
- camada de rede: a comunicação é realizada dentro da própria sub-rede.

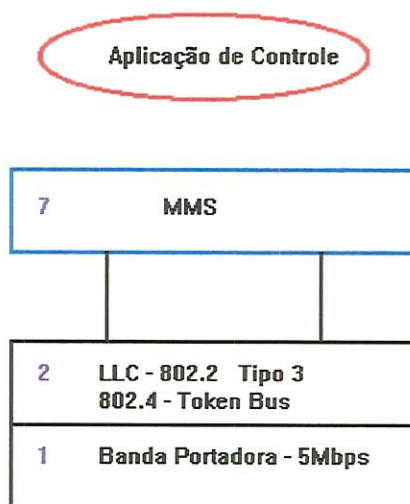


FIGURA 2.7 Arquitetura Mini-MAP.

Apesar desta arquitetura se tornar simples e adequada para pequenos sistemas industriais, células de manufatura e sistemas de controle em tempo real, algumas questões foram levantadas em relação a sua utilização. A primeira refere-se a formação de “ilhas de automação”, isto é, áreas que não se comunicam com outras áreas internas ou externas à fábrica, pode ser resolvido com a utilização de repetidores. E a questão da incompatibilidade com a arquitetura MAP, necessitando gateways para conexão com esta rede.

### 2.4.2.3 Protocolo MMS

O Protocolo MMS (Manufacturing Message Specification) é especialmente dedicado ao fornecimento de serviços para a transmissão de mensagens entre equipamentos programáveis em ambientes fabris. O protocolo MMS especifica a sintaxe e a semântica gerais das mensagens na manufatura, não contendo informações específicas de aplicação. Estas informações são fornecidas pelos diversos "Companion Standards", esses documentos descrevem, entre outras coisas, o mapeamento das funções do dispositivo em serviços MMS e a especificação das classes de serviços a serem suportados pelo dispositivo.

Os principais padrões sendo elaborados são para os dispositivos usados em controle de processos (ISA), comando numérico (EIA), controladores programáveis (NEMA) e controladores de robôs (RIA).

O objetivo principal da MMS é especificar um mecanismo padrão de comunicação entre dispositivos e computadores de tal forma a se obter um alto grau de interoperabilidade. Para tanto, a MMS define, além do formato das mensagens, os seguintes recursos:

- objetos - que são contidos pelo servidor (variáveis, programas, eventos, etc.) e a definição dos atributos desses objetos (nome, valor, tipo) que são visíveis pela rede.
- serviços - (read, write, delete, download, etc.) que os clientes utilizam para acessar e manipular os objetos e
- comportamento - como o servidor se comporta na recepção dos pedidos desses serviços.

O protocolo MMS é baseado no conceito de VMD (Virtual Manufacturing Device) que trata, de forma abstrata, dos elementos estruturais e dos objetos envolvidos nos dispositivos reais. O VMD especifica como o servidor MMS se comporta quando visto por uma aplicação cliente externa. Detalhes internos de como um dispositivo implementa o VMD (linguagem de programação, CPU, sistema operacional, memória, I/O, etc.) não são especificados, conseguindo-se desta forma um alto grau de interoperabilidade e sendo adequada para uma ampla gama de aplicações e dispositivos.

Um aspecto fundamental do VMD é a relação cliente-servidor entre as aplicações de rede. Cliente é a aplicação que solicita dados ou ações do servidor. De uma forma geral, um cliente MMS é o dispositivo que solicita serviços e servidor o que responde ao cliente.

O VMD em si pode ser visto como um objeto ao qual todos os outros objetos MMS são subordinados. São fornecidos serviços para a obtenção de informações sobre o VMD e para gerenciamento e obtenção de informações sobre todos os objetos contidos no VMD.

### **2.4.3 Fieldbus - Barramento de Campo**

De acordo com SHIN (1996) embora as redes de computadores tenham sido bastante pesquisadas, suas aplicações específicas em automação fabril ainda não tinham sido estudadas minuciosamente. A maioria das pesquisas analíticas ocuparam-se com a área geral da comunicação em rede, enfocando-se projetos

de sistemas flexíveis e avaliações de performances. Entretanto, comunicação em tempo real entre instrumentos raramente têm sido estudadas.

Ainda em conformidade com SHIN (1996) o MAP pode prever alguns limites de comunicação em tempo real, mas ele não pode garantir a distribuição de mensagens em tempo crítico antes de seus *deadlines*. De fato, o MAP prover somente ordenamento temporal entre instrumentos baseados em suas prioridades. O MAP sete camadas é muito lento para ser usado em comunicações em tempo real, já que existem pelo menos 14 atrasos de camadas em uma simples comunicação. O miniMAP emprega as duas primeiras camadas do MAP e combina as 5 restantes dentro de uma única, espera-se reduzir o atraso na comunicação, mas ainda deixa a questão da comunicação em tempo real a desejar.

Os esforços para normalização da rede Fieldbus levaram organismos internacionais a definirem suas próprias normas. Entre eles se destacam o FIP na França, a ISA SP-50 nos Estados Unidos e o PROFIBUS na Alemanha.

Todas as propostas existentes baseiam-se no modelo OSI da ISO, apresentando um número reduzido de camadas (física, enlace de dados e aplicação), que definiremos a seguir:

Na camada física, a topologia física adotada é o barramento, utilizando como principal meio de transmissão o par trançado. Podem ser feitas também transmissão através de fibra ótica, via rádio e via rede de distribuição elétrica. O par trançado mais utilizado é o RS-485.

Segundo Di Stefano (1991) a camada de enlace é a que apresenta maiores divergências entre as propostas e também a camada mais crítica no que se refere à comunicação, já que tem que resolver todos os problemas relacionados à restrição de tempo nos processos.

As duas alternativas básicas utilizadas são:

- acesso centralizado mestre/escravo; e
- acesso descentralizado controlado por ficha (token passing).

Como serviços de comunicação a camada de enlace oferece os seguintes:

- transmissão de dados com reconhecimento (SDA - Send Data with Acknowledgment) e sem reconhecimento (SDN - Send Data with No acknowledgment);

- transmissão de dados com resposta (SDR - Send Data with Reply) e transmissão cíclica de dados com resposta (CSDR - Cyclical Send Data with Reply);
- requisição de dados com resposta (RDR - Request Data with Reply) e requisição cíclica de dados com resposta (CRDR - Cyclical Request Data with Reply).

Com relação à camada de aplicação há uma convergência em se adotar um subconjunto do protocolo MMS.

Existem vários padrões de Fieldbus atualmente, dentre os quais o BITBUS, LONworks, CAN, PROFIBUS/ISP, WorldFIP, IEC/ISA, etc. A seguir mostraremos uma tabela com a descrição de alguns sistemas Fieldbus no mercado:

Sistema Bus (barramento)	Taxa de transmissão (bits/seg.)	Tempo de resposta c/ 10 nós (seg.)	Número máximo de unidades	Comprimento máximo do bus (m)	Tipo de bus
Interbus-S	500 K	1,2 ms	256	12,8 Km	CO, IO
Interbus-SL	500 K	< 1 ms	64	100 m	IO
Profibus FMS	19,2 K	2 ms	31x4	23,8 Km	P, IO
Profibus DP	12 M	1 ms	31x4	23,8 Km	CO, IO
Can	1 M	< 1 ms	110	1 Km	CO, IO
WorldFIP	2,5 M	6 ms	32x5	5 Km	P, CO
BitBus	375 K	100 ms	250x4	13,2 Km	IO, IO
LON works	1,25 M	7 ms	2x128	**	IO
P = processo      CO = Bus nível de controle      IO = Bus nível I/O					

TABELA 2.2 Descrição de alguns sistemas Fieldbus no mercado.

Como principais padrões temos o SP-50 IEC/ISA, o mais importante deles e é esperado como o padrão internacional pois está em fase de conclusão, o PROFIBUS e o FIP, padrões já estabelecidos. Na seqüência faremos uma descrição destes.

### **2.4.3.1 Principais padrões para protocolos no Fieldbus**

#### **1. FIP (Factory Instrumentation Protocol)**

O FIP é uma proposta francesa que se traduziu nas normas NF C 46-602 a NF C 46-607. Baseiam-se no modelo OSI/ISO reduzido a três camadas: física, enlace e aplicação.

O FIP se propõe a ser um sistema de gerenciamento de uma base de dados industrial, de tempo real e distribuída. Uma das peculiaridades mais interessantes do FIP é o seu conceito de modelo Produtor/Distribuidor/Consumidor (PDC). Neste modelo, o distribuidor é responsável pela transferência de dados do produtor para o(s) consumidor(es) e está encarregado de validar as transações em relação à restrição de tempo. O produtor é aquele que produz um determinado dado, enquanto o consumidor é um processo de aplicação que necessita de dados para estar apto a executar uma ação. Neste processo, o distribuidor coloca no barramento o nome da variável, imediatamente tanto o produtor como o(s) consumidor(es) desta reconhecem a solicitação, o produtor transmite ao meio e o(s) consumidor(es) a armazenam.

O FIP oferece na camada de aplicação, dois tipos de conjunto de serviços: serviços de transferência de mensagens (subconjunto do MMS) e serviços periódicos aperiódicos MPS (Message Periodic/Aperiodic Services).

Atualmente o FIP, que teve início através de empresas de origem francesa, foi estendido a nível mundial formando a WorldFIP, através da união dos criadores do FIP com empresas como Honeywell, AllenBradley, etc. .

#### **2. PROFIBUS - PROcess FIeldBUS**

O PROFIBUS é a proposta alemã que se traduziu na norma DIN 19 245, baseia-se também no modelo OSI/ISO reduzido a 3 camadas: física de enlace e de apresentação. Algumas funções destas camadas suprimidas são executadas pela camada de enlace de dados ou pela camada de aplicação. Temos então definidas as camadas FMS/ LLI, FDL e PHY, além dos gerentes para estas camadas (FMA1/2 e FMA 7). A figura a seguir mostra um diagrama ilustrativo das camadas de rede no Profibus.

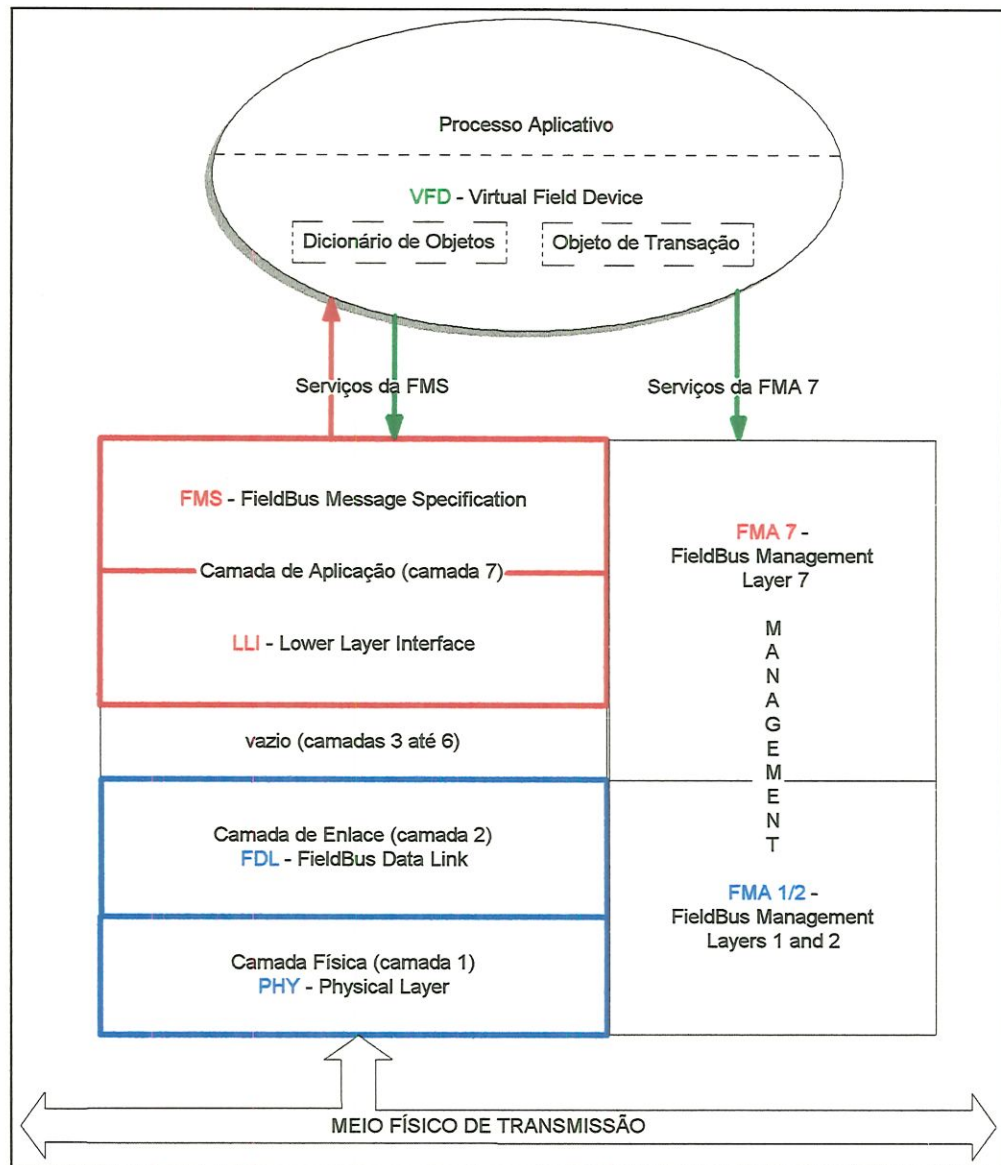


FIGURA 2.8 Alocação das camadas OSI/ISO no Profibus e o modelo VFD (Virtual Field Device).

Podemos descrever genericamente as camadas definidas na Profibus da seguinte forma.

1. A camada de aplicação é subdividida em:
  - a) sub-camada FMS (Fieldbus Message Specification), que tem como a função manter a interface entre o programa aplicativo que necessita usar a rede Profibus e o programa de rede Profibus; e

- b) sub-camada LLI (Lower Link Interface). Devido a falta das camadas 3, 4, 5 e 6 da ISO/OSI na Profibus, a camada LLI faz uma adaptação entre as camadas FMS e FDL. Estabelece, supervisiona e finaliza as conexões, que são chamadas de Relações de Comunicação, entre a estação local e a parceira de comunicação. As características de cada Relação de Comunicação estão armazenadas numa Lista de Referências de Comunicação (CRL - Communication Relationship List). Usando a CRL, a LLI faz o roteamento das mensagens, que serão transmitidas para uma determinada estação Profibus, através da rede.
2. A camada de enlace é denominada FDL (Fieldbus Data Link). Ela cuida dos protocolos de transmissão e dos formatos das mensagens (PDU's) enviadas pela rede. Além disso, é responsável pelo controle de acesso ao meio físico de transmissão.
  3. A camada física é denominada PHY (Physical Layer). A camada física envolve o software e o hardware que cuidam da transmissão a nível físico. Ela cuida da transmissão e da recepção dos bytes que são transmitidos na forma de caracteres UART.

Os gerentes das camadas da Profibus são os seguintes:

- gerente das camadas 1 e 2, FMA 1/2, este não é nenhuma outra camada definida no Profibus. Ela cuida da inicialização, da supervisão e da manipulação de erros nas camadas 1 e 2. Existem serviços na interface entre a camada PHY e a FMA 1/2 e entre a camada LLI e a FMA1/2 que servem para o gerente FMA 1/2 manipular e controlar a inicialização, a supervisão e os erros nestas duas primeiras camadas. Estes serviços estão nas duas camadas mencionadas e disponíveis para o gerente.
- gerente das sub-camadas FMS e LLI, FMA 7. As funções da FMA 7 são organizadas de tal modo que estações da Profibus possam ser implementadas sem necessitar do gerenciamento completo da rede.

A FMA 7 tem as suas responsabilidades divididas em três grupos:

- Gerenciamento do Contexto, que envolve o estabelecimento e o desligamento das conexões de gerenciamento;
- Gerenciamento da Configuração, que envolve os objetos de comunicação do sistema;



- Gerenciamento de Falhas, que cuida do tratamento de erros e reinicialização de estações.

O Profibus utiliza uma técnica híbrida de acesso ao meio de transmissão física. O Profibus mistura dois métodos para realizar esta tarefa. Um método é chamado de Passagem por Token e o outro faz o acesso ao meio físico de transmissão pelo método Mestre - Escravo.

Numa rede Profibus apenas um mestre tem acesso ao meio físico de transmissão em algum determinado instante. Para determinar a estação mestre que tem acesso ao meio usa-se um sinal chamado de token. O mestre que estiver com o token é quem vai ter acesso ao meio de transmissão. Depois de executada a tarefa ou ultrapassado o tempo limite de uso do meio físico de transmissão para aquela estação mestre, o token é então passado para outra estação mestre. A passagem do token é feita em um anel lógico. Dito lógico porque a topologia da rede Profibus é em barramento. Cada estação mestre é quem gerencia a passagem do token sabendo quem é o seu predecessor, isto é, o mestre que enviou o token para ele; e sabendo quem é o seu sucessor, isto é, o mestre para o qual vai ser enviado o token.

Há uma tendência do PROFIBUS transformar-se no ISP (Interoperable System Protocol), formado pela Rosemount, Siemens, Yokogawa, entre outras. Tendo como objetivo produzir um sistema híbrido que combine, com base no nível físico do IEC/ISA-SP50 e prevendo segurança intrínseca, os aspectos das várias propostas de barramento, basicamente PROFIBUS, também com elementos do FIP e do HART.

### **3. ISA SP-50 / IEC**

Os trabalhos realizados na elaboração do PROFIBUS e do FIP levaram à formação do grupo SP-50 da ISA (*The International Society for Measurement and Control Standards and Practices Group 50*) cujos trabalhos estão sendo considerados pelo IEC para a elaboração de uma norma internacional. Seus resultados já podem ser observados na norma IEC 1158 (Physical Layer Specification).

De acordo com NETO (1995) a arquitetura prevista para este trabalho também se baseia no modelo OSI da ISA com as mesmas três camadas do PROFIBUS e FIP. Porém, a ISA sugere a adição de uma quarta camada (camada

do usuário), ainda não aceita pelo IEC. O SP-50 utiliza os conceitos do FIP e PROFIBUS, sendo considerado uma evolução destes trabalhos.

Os requisitos inicialmente especificados, levaram à definição de uma camada física de 2 níveis:

- H1 - controle de processo tradicional, baixa velocidade e utilizando a fiação existente (ponto a ponto), com segurança intrínseca e fornecendo alimentação no bus;
- H2 - controle de processos de alta velocidade e processos lógicos, usando nova fiação e se possível, alimentação no bus e segurança intrínseca.

A camada física especifica um método de transmissão bidirecional e assíncrona, taxas de transmissão de 31,25 Kbps ( $H_1$ ), 1 Mbps e 2,5 Mbps ( $H_2$ ), meios físicos por par trançado e cabo coaxial, topologia tipo barramento ou árvore, utilizando uma caixa de distribuição, distância máxima de até 1900 metros sem repetidor para a taxa de 31,25 Kbps, até 750 metros para 1Mbps e até 500 metros para 2,5 Mbps. Para baixa velocidade admite-se até 32 instrumentos por barramento sem considerar segurança intrínseca, 12 instrumentos com alimentação pelo barramento e sem segurança intrínseca e de 2 a 6 com alimentação pelo barramento com segurança intrínseca. Novos trabalhos têm sido desenvolvidos para outros meios e taxas maiores, mas ainda não estão normalizados.

A camada de enlace de dados é a que têm tido mais dificuldade de padronização devido às divergências entre os sub-comitês. O acesso ao barramento é centralizado e controlado por uma estação chamada LAS (*Link Active Scheduler*).

A camada de usuário (User Layer) tem por objetivo facilitar o funcionamento dos instrumentos da rede em conjunto, ela define a estrutura de base de dados que residirá em cada instrumento de controle ou medida. O objetivo é permitir que diversos fornecedores possam construir seus produtos com algoritmos pré-definidos, requerendo apenas que a base de dados seja carregada para uma configuração específica. Define, também, o método para escalonamento de todos os blocos funcionais baseados na necessidade de tempo crítico para cada bloco. O período de execução de cada bloco é determinado na configuração da base de dados.

Todo bloco funcional contém um algoritmo e uma base de dados associada que é usada pelo algoritmo. Os instrumentos podem ser tantos blocos funcionais residentes quantos forem necessários, escalonados conforme definição do usuário. E ainda possibilita a inclusão de novos blocos, de modo a satisfazer requisitos que sejam específicos para determinada aplicação.

A camada de usuário pretende incorporar uma linguagem comum de programação para todos os instrumentos de campo, fornecendo um terminal de programação e uma metodologia de configuração única para instrumentos de diferentes fornecedores.

A seguir apresentaremos uma tabela comparativa com as principais características dos barramentos de campo apresentados anteriormente:

Organizações/Normas	SP-50	FIP→WorldFIP	PROFIBUS→ISP
<b>Camada de Aplicação</b>			
<b>Serviços</b>	subconjunto do MMS	MPS e subconjunto do MMS	subconjunto do MMS
<b>Linguagem</b>	ASN.1	ASN.1	-
<b>Camada de Enlace</b>			
<b>Acesso ao Meio</b>	token (LAS) produtor-consumidor atualização cíclica	produtor-consumidor	híbrido: token entre mestres, e mestre/escravo
<b>Camada Física</b>			
<b>Tipo de Meio</b>	par de fios trançados fibra ótica rádio	par de fios trançados fibra ótica	RS-485
<b>Taxa de Transmissão</b>	31,25 Kbps (H <sub>1</sub> ) 1 Mbps, 2,5 Mbps (H <sub>2</sub> )	31,25 Kbps 1 Mbps (padrão) 2,5 Mbps	19,2 Kbps 187 Kbps 500 Kbps
<b>Número de Elementos</b>	2 a 32	8 por caixa de distribuição	127 total com 32 por segmento
<b>Tipo de topologia</b>	barramento ou árvore de barramentos	barramento e caixa de distribuição	barramento ou árvore de barramentos
<b>Comprimento máximo do barramento e tronco principal</b>	1900 m 750 m	1900 m 750 m	1200 m

TABELA 2.3 Comparação entre os barramentos SP-50, FIP, PROFIBUS.

## 2.5 Estratégias de Controle em Tempo Real

Deve-se ter em mente que a velocidade de execução do software, em resposta a eventos externos, deve ser previsível em termos absolutos; caso contrário a correta operação do sistema não pode ser garantida. Para alcançar tal capacidade de ser previsível, os sistemas operando em tempo real devem satisfazer critérios de tempo muito rigorosos. Segundo JONES & FLYNN\* apud ARMSTRONG (1997) existem três estratégias primárias para controle de eventos externos em tempo real. Estas são; varredura, captura de entrada e interrupções.

### 1. Controle através de varredura

A varredura é um método onde o software continuamente checka o estado de sensores externos. JONES & FLYNN\* apud ARMSTRONG (1997) cita que a varredura prende o processador, mantendo ele ocupado mesmo quando nenhum evento externo está acontecendo. Isso significa que o processador não estará habilitado a fazer qualquer outro trabalho enquanto estiver esperando por um evento ou terá que "saltar" entre outro trabalho e a tarefa do sensor. Este salto entre tarefas implica num aumento do período de resposta do sensor, aumentando assim o tempo previsível para tal tarefa.

### 2. Controle através de captura de entrada

A captura de entrada depende de registros que podem automaticamente capturar dados, baseado no hardware ou em funções temporizadas automáticas. Eles são fáceis de usar mas são limitados em flexibilidade e tipos de eventos que podem ser capturados.

---

\* JONES, J.L.; FLYNN, A.M.(1993). Mobile robots, inspiration to implementation. 629.8. A.K. Peters, Ltd. ISBN 1-56881-011-3 apud ARMSTRONG, E. Actuator & sensor coupling for reconfigurable robotic systems. Aberystwyth, 1997. Tese (Doutorado), University of Wales.

### 3. Controle através de interrupções

De acordo com IEEE610.12<sup>#</sup> apud ARMSTRONG (1997) uma interrupção é a suspensão de um processo para tratar de um evento externo ao processo. Logo após a ocorrência de uma interrupção, uma outra tarefa começa ser executada para um posterior retorno à tarefa interrompida. No gráfico a seguir poderá ser visto o exemplo gráfico de ocorrência de interrupção.

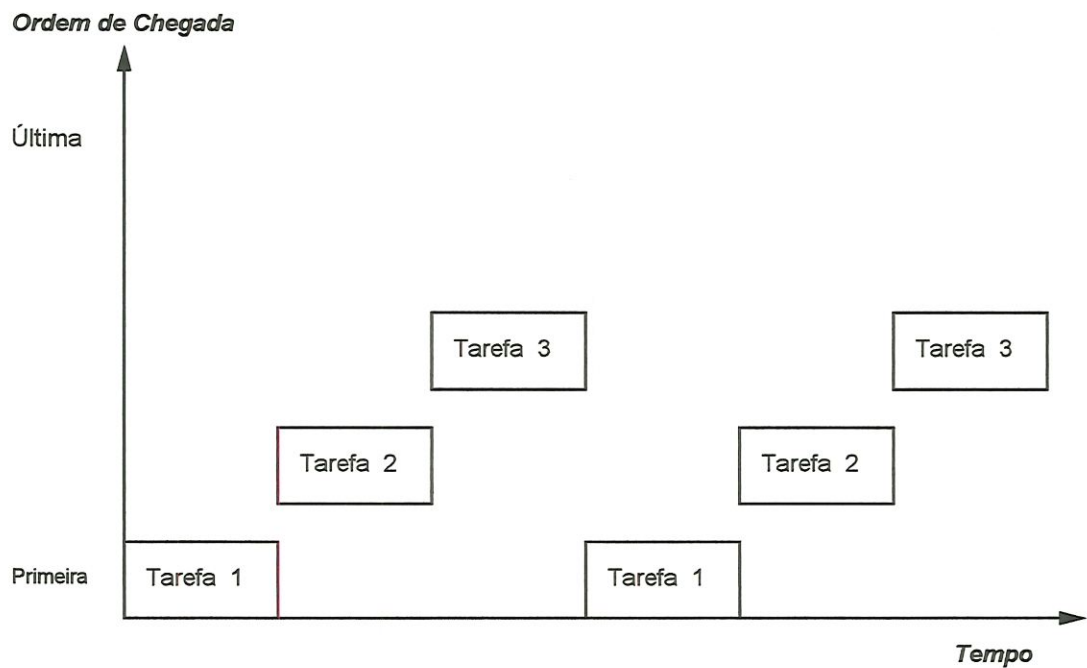


FIGURA 2.9 Repartição do tempo entre três tarefas através da interrupção.

<sup>#</sup> IEEE610.12. (1990). Glossary of software engineering terminology. Tech. Rept. IEEE. apud ARMSTRONG, E. Actuator & sensor coupling for reconfigurable robotic systems. Aberystwyth, 1997. Tese (Doutorado), University of Wales.

### 3. Escalonamento da comunicação no Fieldbus

#### 3.1 Introdução

De acordo com FRANCO (1995) a garantia de utilização adequada de um Fieldbus numa aplicação do usuário depende do atendimento aos tempos exigidos por esta aplicação, que por sua vez é função, principalmente, do escalonamento da comunicação entre os dispositivos a ele acoplados. A versatilidade dos algoritmos de escalonamento da comunicação para Fieldbus permitirá o uso de abordagens que garantirão a utilização destes barramentos em aplicações do usuário com restrições críticas de tempo (Sistemas "Hard Real Time").

Um Fieldbus é uma rede local de tempo real onde todas as informações são, transmitidas serialmente e multiplexadas no tempo. Este tipo de rede deve satisfazer às exigências temporais dos Sistemas Hard Real Time, que segundo BURNS (1991) são aqueles que possuem *deadlines* cruciais, ou seja, onde os danos de uma falha são muito maiores que qualquer benefício trazido pelo sistema.

De acordo com BURNS (1991) os escalonadores devem ser *preemptivos* ou *não-preemptivos*. Segundo AUDSLEY (1990b) um escalonador *preemptivo* é aquele que pode suspender arbitrariamente uma execução de processo e reiniciá-la posteriormente, sem que tal artifício afete o comportamento daquele processo (exceto por acréscimo do tempo transcorrido). A *preempção* tipicamente ocorre quando um processo com prioridade maior precisa ser executado e seu efeito é que um processo deve ser suspenso involuntariamente.

No controle de processos também são utilizados escalonadores *não-preemptivos*, ou seja, não acontece a interrupção de um processo depois que se tenha dado início ao mesmo. Sendo assim, qualquer outro processo deve aguardar o término do processo que está sendo executado independente da sua prioridade.

### 3.2 Tipo de comunicação no Fieldbus

Nas aplicações de manufatura e controle de processos industriais, o Fieldbus deve manusear duas classes de transferência de dados, dados síncronos e mensagens assíncronas. Os dados síncronos são gerados periodicamente e têm de ser transmitidos antes de seus *deadlines*. De acordo com RAJA (1993) as mensagens assíncronas são geradas por eventos aleatórios não podendo ser previstas com antecipação, entretanto, geralmente, elas possuem *deadlines* críticos menores.

Um modo de contornar o problema de dados assíncronos com *deadlines* críticos é os mesmos serem varridos periodicamente, mesmo se não for necessário. Assim, seu tratamento seria o de encaixá-los na fase síncrona e seriam escalonados com ela, provavelmente "off-line", como mostrado na figura a seguir.



FIGURA 3.1 Ciclo com fase síncrona e fase assíncrona.

Sabe-se que tanto para os dados periódicos como para os aperiódicos seus tempos de ocupação do meio são função de seus números de bits. No entanto, a comunicação no Fieldbus é caracterizada por usar dados e mensagens de poucos bytes. O que justifica não ser permitido interrompê-los durante sua transmissão, segundo FRANCO (1995) a transmissão de dados no Fieldbus é não-preemptiva, significando que depois de iniciada a transmissão esta não pode ser interrompida para a execução de um outro processo, tendo-se que aguardar o seu término.

Como as aplicações do Fieldbus estão envolvidas com transações do tipo, coleta dos valores dos sensores enviados pelo barramento, processamento de algoritmos de controle e envio pelo barramento de seus resultados para acionamento de seus atuadores, existe uma dependência entre estas transações que ocasiona uma exigência no seqüenciamento da transmissão de seus dados.

Toda a comunicação entre os seus dispositivos é feita através do barramento. Como o meio de comunicação deve ser utilizado por apenas um dispositivo por vez (compartilhado), é necessária uma política de acesso a este meio. Para assegurar que todas as variáveis representem uma cópia fiel de seus valores reais do ambiente é preciso garantir o acesso em momentos precisos.

A seguir mostraremos uma tabela com algumas analogias entre as áreas da Ciências da Computação, Pesquisa Operacional e Comunicação no Fieldbus, tais analogias podem ser levantadas a fim de se estudar a adequação de algoritmos nelas já estabelecidos.

<b>Ciência da Computação</b>	<b>Pesquisa Operacional</b>	<b>Comunicação no Fieldbus</b>
processador	máquina	barramento
processo	job ou tarefa	dados ou mensagens
número de instruções	número de processamentos na máquina	tamanho do dado ou tamanho da mensagem
não interrupção do processo	não interrupção do processamento	não interrupção da transmissão do dado ou da mensagem
tempo de computação	tempo de processamento	tempo de ocupação do meio para a transmissão

TABELA 3.1 Quadro comparativo entre Ciências da computação, Pesquisa Operacional e Comunicação no Fieldbus.

O escalonamento de processos para execução num processador, na Ciência da Computação é equivalente ao escalonamento, por exemplo, de um processamento numa máquina de uma linha de manufatura. E é análogo ao escalonamento da transmissão de dados ou mensagens no barramento. Na ciência da computação estudam-se algoritmos para escalonar o tempo de computação de cada processo no processador. Na comunicação do Fieldbus escala-se o tempo de ocupação do meio para transmissão no barramento, que



é, por sua vez, função do tamanho de cada mensagem para uma mesma taxa de transmissão.

É a partir de tais analogias que devem ser analisados os algoritmos de escalonamento na Ciência da Computação e então procurar os que se aplicam ao escalonamento de um único processador para processos *não-preemptivos* periódicos e aperiódicos.

### 3.3 Algoritmos de escalonamento

Nos Sistemas Operacionais multitarefas os processos são disparados de acordo com as suas importâncias e não conforme os deadlines, isto é inadequado aos sistemas de tempo real que necessitam atender suas restrições de tempo. Deste modo, procuram-se novos algoritmos. Os algoritmos de escalonamento podem ser estáticos ou dinâmicos.

Escalonamento estático refere-se ao fato de que o algoritmo de escalonamento tem completo conhecimento a respeito do conjunto de tarefas e suas restrições como *deadlines*, tempos de computação, seus futuros tempos livres, etc., requerendo desta maneira um pequeno tempo de execução. Segundo STANKOVIC (1985) tais questões são realistas para muitos sistemas em tempo real. Um sistema de controle em tempo real que tem um ambiente e requerimentos de processo bem definidos seria um bom candidato para este método de escalonamento. De acordo com ARMSTRONG (1997) o escalonamento que é realizado não sofre trocas podendo ser guardado para ser usado outras vezes.

Em contraste, um algoritmo de escalonamento dinâmico tem completo conhecimento à respeito do conjunto de tarefas correntes, mas novas mudanças devem ocorrer no futuro, desconhecidas para o algoritmo no momento que ele está escalonando as tarefas correntes, ou seja, ele determina o escalonamento durante a execução. Sendo assim, como citado por STANKOVIC (1985) o escalonamento troca a todo momento. Podendo vir a exigir, desta forma, um maior tempo de processamento devido a sua flexibilidade.

Atualmente, ainda existe uma discussão sobre a adequação dos algoritmos dinâmicos para sistemas de *hard real time*. Mas, a procura de um algoritmo dinâmico eficiente, com este fim, tem se mostrado promissor. Todavia, de acordo com BURNS (1991) algoritmos dinâmicos têm um importante papel:

- são particularmente apropriados para sistemas com soft deadline,
- poderiam formar parte de um procedimento de recuperação de erro para perdas de hard deadlines,
- podem ser usados se as aplicações falham proporcionando um estouro no limite superior determinado pelo pior caso; por exemplo, o número de aviões em uma área de controle de tráfego aéreo.

A seguir serão descritos, resumidamente, os principais algoritmos de escalonamento de processador, analisados na busca da solução do problema da comunicação no Fieldbus. Esta descrição não pretende ser completa, mas ponto de partida para estudos mais profundos.

#### **Rate Monotonic Algorithm - RMA**

O rate monotonic algorithm requer um escalonador preemptivo, podendo ser escalonado antes do tempo de execução, realizando as tarefas de forma independente, através do pior caso ou por tempos médios de execução. Segundo BURNS (1991) é baseado em prioridades estáticas onde a prioridade de cada tarefa está relacionada ao seu período. De acordo com AUDSLEY (1990a), inicialmente exigia tarefas periódicas e com *deadlines* iguais ao período. Ainda segundo AUDSLEY (1993) evoluiu para acomodar atividades esporádicas e com bloqueio (exigência de sincronização ou de relações de precedência) devido às interações entre tarefas.

#### **Sporadic Server - SS**

Propõe, com uma política de escalonamento preemptivo, atender às tarefas periódicas e aperiódicas não esporádicas com servidor periódico alocando tempo de execução de acordo com os algoritmos de preservação da largura de banda.

Combina algoritmos Priority Exchange com Deferrable Server, onde a capacidade de atendimento às tarefas não periódicas não usada é convertida em tickets. E quando houver tickets, as tarefas aperiódicas rodam imediatamente. Garante atendimento de tarefas aperiódicas críticas. Porém, de acordo com AUDSLEY (1990b) a utilização do processador é reduzido quando a tarefa aperiódica estiver inativa.

Para garantir os *deadlines* de tarefas esporádicas com *deadlines soft*, as prioridades podem ser atribuídas segundo o DMP (Deadline Monotonic Priority). Como citado por SPRUNT (1989) o teste da habilidade de escalonamento do Sporadic Server Deadline Monotonic pode ser analisado “off-line”, com as fórmulas desenvolvidas para os protocolos de Priority Ceiling.

#### **Deadline Monotonic Algorithm - DMA**

De acordo com AUDSLEY (1993) é baseado no RMA, podendo acomodar processos esporádicos, diferentes níveis momentos críticos, interação e bloqueio entre processos, relações de precedência entre processos, processos com *multi-deadlines* e *deadlines* menores que seus períodos. O DMP atribui prioridades estáticas em função de seus *deadlines*.

#### **Earliest Deadline First - EDF**

De acordo com XU (1990) é um escalonador dinâmico “on-line” preemptivo (e não preemptivo) que atribui a maior prioridade ao processo com *deadline* mais próximo, colocando-o em execução a seguir. Segundo BURNS (1991) é ótimo para processos periódicos e aperiódicos, e de acordo com STANKOVIC (1985) é ótimo também para ambientes de um único processador.

#### **Least Laxity Scheduling**

Segundo RAMAMRITHAN (1984) é escalonador dinâmico “on-line” preemptivo, que atribui a maior prioridade à tarefa com a menor diferença entre seu *deadline* e seu tempo de computação, colocando-a em execução a seguir. E, segundo AUDSLEY (1990b) apresenta problemas quando duas ou mais tarefas têm diferenças similares, por suspender a execução e chavear o contexto entre estas tarefas a cada pequeno avanço na execução.

#### **Algoritmos Genéticos**

São baseados em técnicas computacionais estocásticas da evolução natural, que tentam reter as informações genéticas de geração para geração, através de um planejamento de reprodução. As boas características da última geração são selecionadas e, operadores genéticos como: cruzamento, mutação e outros são

usados para a próxima geração. De acordo com LAWLER (1969) têm larga possibilidade de aplicação, flexibilidade, fácil implementação e com capacidade de encontrar soluções próximas da ótima.

### **Algoritmos com Simulated Annealing (Têmpera Simulada)**

De acordo com RICH (1994) estes algoritmos são modelados baseados no processo físico de recozimento, onde substâncias físicas como os metais são fundidas e depois gradualmente resfriadas até alcançarem um estado sólido. Geram uma seqüência de cadeia de Markov para valores de temperaturas decrescentes. Para cada temperatura, o processo de geração é repetido várias vezes até a distribuição da probabilidade dos estados do sistema se aproximar da distribuição de Boltzmann. Na Ciência da Computação, o espaço do problema é definido como sendo o conjunto de todas as alocações possíveis de um dado conjunto de tarefas para os processadores. O espaço vizinho é o conjunto de todos os pontos que são alcançáveis por movimentação de qualquer tarefa para qualquer outro processador. O algoritmo desenvolvido por TINDELL (1992) mede a energia de cada ponto para determinar se esta alocação é uma alocação pobre (alta energia) ou não (baixa energia). O ponto de menor energia no espaço do problema é a solução do problema. Segundo TINDELL (1992) estes algoritmos têm sido aplicados em diversas áreas, sendo bem flexíveis e têm apresentado resultados encorajadores, segundo estudos recentes.

### **Processo com Tempos de Release, Deadlines, Relações de Precedência e de Exclusão**

O algoritmo citado por XU (1990) escalona "off-line" os processos periódicos em hard real time (podendo ser adaptado para aperiódicos) para um único processador, permitindo especificar os processos que não são preemptivos (citados como com relações de exclusão), suas relações de precedência, os seus tempos de *release*, seus *deadlines* e seus tempos de computação. Utiliza EDF e a técnica "Brach-and-Bound" da Inteligência Artificial.

Como foi colocado anteriormente a apresentação desses algoritmos não teve a pretensão de ser muito aprofundada, pois isso geraria uma grande quantidade de informação. A apresentação dos mesmos teve a intenção de localizar o que está sendo pesquisado atualmente dentro do contexto abordado neste trabalho, servindo desta forma como referência para estudos mais aprofundados.

#### **4. Características técnicas dos protocolos relativas à transmissão**

Em aplicações de controle de processos e automação industrial o método de acesso ao meio de transmissão deve estar atento para a necessidade de que todas as estações tenham oportunidade de utilização do meio para execução de suas transmissões de dados e mensagens. Além disso deve possibilitar a comunicação entre equipamentos complexos e dispositivos simples pertencentes à planta.

De acordo com CAVALIERI (1993) a necessidade de tornar todos os dados seriais e o comando do tráfego representa a parte mais crítica da arquitetura de camadas do Fieldbus que tem de suportar , em um único barramento, todo tráfego periódico (gerado pela troca de medidas e variáveis de controle) e o tráfego aperiódico (alarmes, parâmetros de configuração, etc.), devendo respeitar as diferentes restrições de tempo para as transmissões e que nos sistemas atuais é distribuída por meio de muitas conexões ponto a ponto.

A Camada de Enlace (DLL - Data Link Layer) é responsável por tal tarefa, pois é um protocolo projetado para suportar comunicações com tempos críticos entre equipamentos num ambiente de automação. Esta camada cuida dos protocolos de transmissão, definindo os procedimentos necessários para a transferência de dados ou informação de controle, e o protocolo dos formatos das mensagens (PDU's, protocol data units) enviadas pela rede. Além disso, é responsável pelo controle de acesso ao meio físico de transmissão.

A seguir serão descritas as características técnicas de acesso ao meio dos três principais protocolos presentes no mercado. Tendo o Profibus e FIP como protocolos já normalizados e sendo utilizados pelos fabricantes, enquanto que o protocolo IEC/ISA SP50 apesar de já estar sendo utilizado ainda não lançou a norma definitiva, devendo ser lançada em breve, por enquanto o usuário utiliza-se de drafts (esboços) que lhes permitem realizar projetos baseados num protocolo que é uma importante tendência do mercado.

#### 4.1 PROFIBUS (PROcess FieldBUS)

O protocolo PROFIBUS reconhece que qualquer instalação prática industrial envolverá controladores múltiplos de nível de célula, como também programação e configuração de dispositivos. Todos os quais podem precisar comunicar-se um com outro e com dispositivos no campo. É fornecido um mecanismo onde múltiplas estações mestres podem ter a oportunidade para ganhar acesso ao bus e prover canais de comunicação totalmente seguros com os dispositivos com os quais eles querem comunicar-se. De acordo com VITA (1996) para este objetivo, o acesso ao meio inclui o método de comunicação token passing entre as estações mestras e uma comunicação mestre-escrava entre um mestre e seus escravos.

O token nada mais é que uma unidade de informação da camada de enlace (frame), transmitido de uma estação ativa (mestre) para a sua sucessora no anel lógico, quando o seu tempo de transmissão esgotar-se.

Neste sistema Híbrido um token ring lógico é estabelecido entre mestres, dando para cada mestre exclusivo acesso ao bus durante um intervalo de tempo preciso. Enquanto o mestre mantém o token pode comunicar-se com outras estações mestras e com os dispositivos periféricos ou escravos como: atuadores, sensores e drives. Segundo MUÑOZ (1997), em princípio, só um mestre pode iniciar as comunicações enviando informação para os dispositivos escravos e solicitando uma resposta, da mesma forma, os escravos também podem enviar mensagens não solicitadas, por exemplo, ativação de alarme.

No momento em que um mestre tem o token, ele se comunica com as estações escravos pelo método mestre/escravo. Nesta situação, apenas o mestre pode iniciar uma seqüência de comunicação entre mestre e escravo, isto é, apenas o mestre pode requisitar dados para o escravo. O escravo apenas terá acesso ao meio físico de transmissão quando lhe for solicitado algo pelo mestre. As estações mestres que não possuírem o token em determinado instante podem ser vistas como escravas da estação que possui o token nesse instante. Existem dois modos de pedido/envio feitos pelos mestres: modo acíclico e modo cíclico. No primeiro modo, os pedidos enviados aos escravos são executados de acordo com algum pedido de informação feita pelo usuário na estação mestre que possui o token naquele momento. No segundo modo, os pedidos enviados aos escravos são feitos ciclicamente, isto é, são feitos numa determinada seqüência fixa. Para

saber quais escravos receberão os pedidos de dados, cada estação mestre possui uma *poll list*, que é uma lista de endereços das estações escravas. Os pedidos de dados feitos no modo cíclico tem menor prioridade que os pedidos feitos no modo acíclico. Significa que quando um mestre acaba de receber o token ele primeiro envia os pedidos no modo acíclico e depois envia os pedidos no modo cíclico.

A figura 4.1 a seguir ilustra a estrutura do método de acesso ao meio de um sistema PROFIBUS.

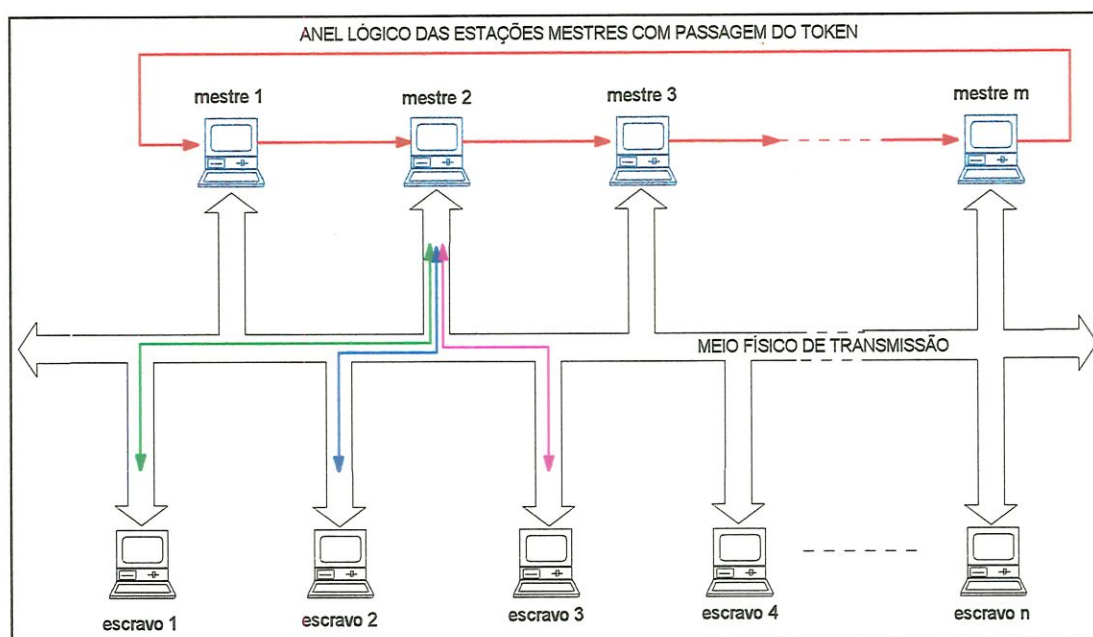


FIGURA 4.1 Método de acesso ao meio físico no Profibus: modelo híbrido(procedimento mestre-escravo com sistema de passagem de token).

## 4.2 FIP (Factory Instrumentation Protocol)

O Factory Instrumentation Protocol é uma coleta de padrões com o objetivo de solucionar as necessidades de comunicação no nível de instrumentação da manufatura. O padrão foi desenvolvido pelo French National Association for Standardization. O protocolo está baseado no modelo OSI da ISO reduzido a três camadas: física, enlace e aplicação.

Neste protocolo a transferência de dados é centralizada, uma estação chamada mestra, árbitro de bus ou bus schedule , controla o acesso ao meio de comunicação.



O intercâmbio de variáveis é feito em duas fases , como mostra as figuras 4.3 e 4.4 respectivamente; na primeira, a estação mestra (árbitro) envia uma solicitação para o produtor  $P(x)$  (ID-frame), na segunda o produtor responde transmitindo a variável (VAR-frame) a todos os consumidores  $C(x)$ .

Os frames possuem dois campos, o primeiro deles é reservado para identificar o tipo de frame e o segundo contém, o identificador da variável no caso do ID-frame e, no caso do VAR-frame o valor da variável. Como mostrado na figura a seguir.

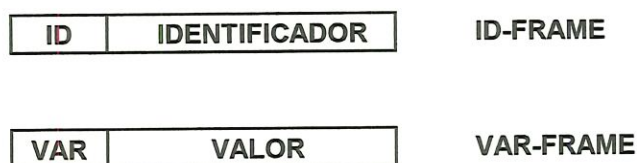


FIGURA 4.2 Estrutura do ID-frame e VAR-frame do protocolo FIP.

Este mecanismo de transmissão está baseado no princípio de transmissão instantânea da variável para todos os dispositivos da rede. Este princípio garante a velocidade na transferência da variável e comunica a identidade de uma variável para todos os consumidores.

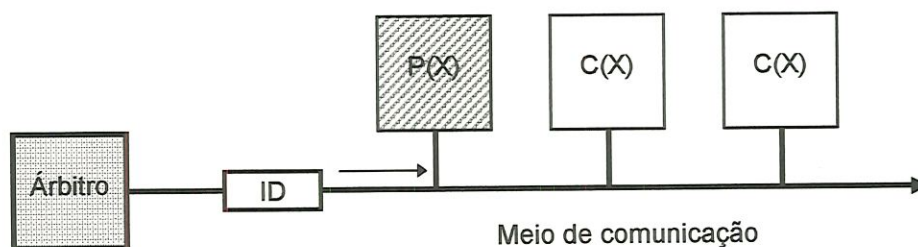


FIGURA 4.3 Envio de solicitação: o árbitro envia solicitação de uma variável, ao produtor  $P(x)$ , através de um ID-frame.

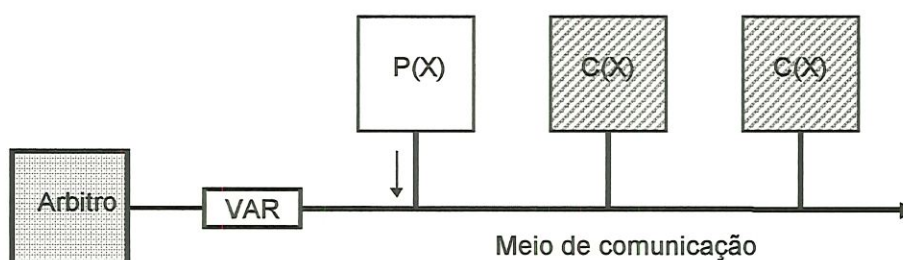


FIGURA 4.4 Resposta à solicitação: o produtor,  $P(x)$ , responde ao envio de solicitação transmitindo uma variável (VAR-frame) ao barramento para ser consumida.

Antes da transmissão das variáveis, elas devem ser identificadas por nomes, deste modo a transmissão de uma variável, o desenvolvimento da rede e teste são realizados mais facilmente. De acordo com MUÑOZ (1997) cada estação ou assinante, pode ser um produtor e/ou consumidor de uma ou mais variáveis.

Na rede FIP, o bus scheduler ou árbitro do bus, introduz um nome de variável na rede. A rede é configurada de tal forma que cada variável tem apenas um produtor. Esta variável é utilizada e armazenada imediatamente por aqueles consumidores identificados por seus nomes. O árbitro injeta depois outro nome de variável e o mecanismo entre produtores e consumidores repete-se indefinidamente. Por este meio, o árbitro lê uma tabela de variáveis periódicas correspondente a um determinado mecanismo produtor/consumidor. Segundo FURNESS (1994) a tabela é definida quando a rede é configurada e existem muitas tabelas, com uma tabela sempre ativa em qualquer momento, permitindo mudanças do estado do processo.

Por outro lado, quando uma estação quer iniciar uma comunicação assíncrona, antes de sua vez na configuração ou que não está na configuração predefinida, pode anexar sua solicitação e identificador ao frame que está finalizando uma operação de atualização de variável. De acordo com CENA (1994) quando a comunicação síncrona é completada o árbitro processa todas as solicitações assíncronas recebidas, que serão atendidas de acordo a sua prioridade; solicitará à estação a lista de variáveis solicitadas, que posteriormente incluirá na seqüência síncrona da comunicação. Para o árbitro enviar os nomes na ordem apropriada, ele deve ter organizada uma lista de objetos. Este modo de transmissão combina dois tipos de serviços o SDN (Send Data with No acknowledgment) transmissão de dados sem reconhecimento e o RDR (Request Data and Reply) requisição de dados com resposta.

O Fip prevê tráfegos periódicos, aperiódicos e mensagens/transferências com reconhecimento. No tráfego periódico a nomenclatura dos objetos a serem transmitidos no barramento é mantida numa seqüência de acordo com as especificações da aplicação. Desde que algoritmos de controle, operação e supervisão são definidos e construídos essencialmente de um modo cíclico, é conhecido estatisticamente quais são os objetos necessários como entrada e quais são os objetos produzidos e que necessitam ser trocados. Esta nomenclatura será repetida ciclicamente conferindo ao barramento a função principal de buscar status periodicamente. O tráfego aperiódico é necessário para a transmissão de um evento, de objetos configurados e para a transmissão de objetos submetidos ao tráfego periódico, onde o consumidor detectou erro.

Nas mensagens e transferências com reconhecimento, a estação transmissora requisita o direito de transmissão ao árbitro, e este garante o direito emitindo um identificador (ID), reservado à estação transmissora. Ao receber o ID, a estação gerencia a transação e retorna o controle ao árbitro. O árbitro garante este direito para qualquer estação, para apenas uma mensagem. No caso da estação necessitar enviar um grupo de mensagens, ela entrega ao árbitro uma lista de identificadores que ele emite quando puder.

### **4.3 Protocolo IEC/ISA SP-50**

A camada física foi desenhada para operar em uma grande variedade de aplicações com uma taxa de dados relacionada à largura de banda requerida pelo serviço. Para controle de processos, o Fieldbus prevê baixa velocidade (H1), mas é intrinsecamente seguro, o bus é capaz de alimentar dispositivos de campo sobre o mesmo par trançado usualmente utilizados nos sistemas de controle analógico de hoje. Por outro lado, se a potência e/ou segurança não são requeridas, pode ser usada a camada física de alta velocidade (H2).

O Fieldbus prevê a integridade e a velocidade requerida para conectar dispositivos multiplexados de Input/Output para CLPs provendo uma rede backbone confiável para múltiplos dispositivos de controle.

Na camada de enlace de dados SP50 dois tipos de token (com aspectos fundamentalmente diferentes) são usados, permitindo ser encontradas diferentes maneiras de comunicação. De acordo com DI STEFANO (1991) eles são o CiT

(Circulated Token) e o DeT (Delegated Token), respectivamente. Em ambos os casos com um árbitro do bus, chamado de Link Active Schedule (LAS).

Um CiT é enviado pelo corrente mestre para o próximo transferindo o direito de iniciar transações a nível de enlace por um período de tempo pre-estabelecido pela LAS, para qual o token tem de retornar quando este tempo se expira. O CiT usa um tipo de mecanismo token-passing (com todos os aspectos de mecanismos típicos tais como instabilidade, delay, etc.). Segundo DI STEFANO (1991) isto, entretanto, não é necessariamente estendido a todos os mestres mas somente para um subgrupo designado pelo LAS.

Um DeT é enviado pelo LAS para um mestre/iniciador transferindo o direito de realizar transações a nível de enlace por um período de tempo pré-estabelecido. Este token pode ser usado somente por um único mestre antes de retornar para o LAS e não conduz à formação de um anel lógico. De acordo com DI STEFANO (1991) é particularmente adequado para comunicações que têm instantes precisos em tempos que não podem ser garantidos por um mecanismo de acesso por token-passing.

A figura a seguir é a representação funcional de um mecanismo de token usado por CiT's e DeT's.

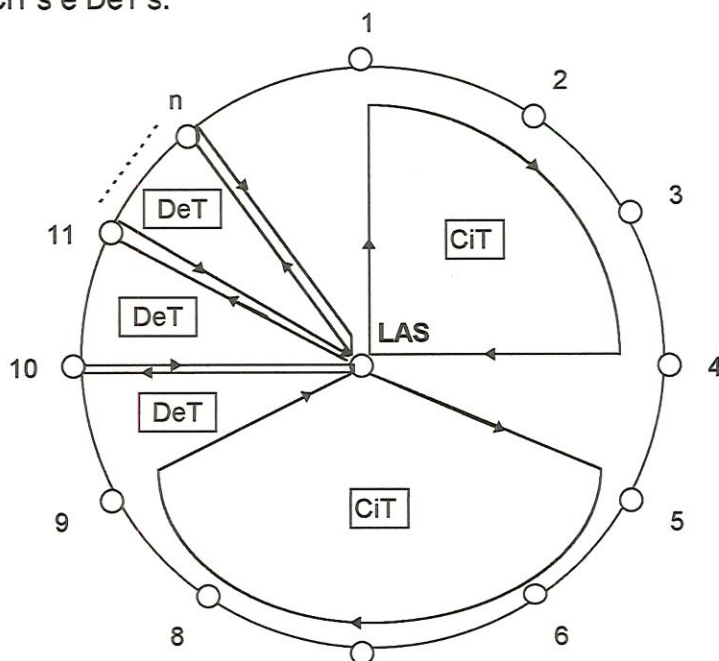


FIGURA 4.5 Representação do mecanismo de passagem de token no protocolo SP-50.

Segundo MUÑOZ (1997) o árbitro do bus ou o Link Active Scheduler (LAS), é eleito entre os outros nós que possuam a categoria de Link Mestre (LM); se por algum motivo o LAS falhar, um Link Mestre pode, dependendo da prioridade pré-estabelecida, assumir as funções de LAS, daí o nome de Active.

LEVITI (1996) afirma que o LAS circulará o token só quando nenhum tráfego schedule for necessário; o token será devolvido ao LAS, o qual, dependendo do tempo disponível, tomará a decisão de passar o token para outro nó ou reassumirá o controle do tráfego schedule.

Diferentes tipos de tokens são usados. Um é chamado Execute Transaction (ET) e é usado para enviar um único frame. O outro é Execute Sequence (ES) que outorga para a estação uma porção de largura de banda para transmitir muitos frames. Outro token é o Compel Data Frame, o qual autoriza a transmissão de uma única informação. Este Frame, permite também, que uma estação obrigue a outra a transmitir informação. O LAS gerencia a transmissão periódica de informação por meio de uma tabela estabelecida contendo todas as transmissões periódicas de acordo com o comprimento (chamada macrociclos) da tabela.

Uma estação pode solicitar por uma largura de banda adicional, colocando no ultimo período do frame transmitido bits apropriados, quando ela não estiver servindo o tráfego periódico, o LAS enviará um token ET, o qual responderá por meio de um frame execute transaction que contém a solicitação da transmissão assíncrona enviada pela estação. De acordo com CAVLIERI (1995) esta solicitação assíncrona será servida pelo LAS, quando a estação não estiver servindo o tráfego periódico.

## 5. Solução para transferência de dados e cálculo de tempos envolvidos

Em sistemas práticos confronta-se com instrumentos que produzem dados em períodos diferentes e a rede tem de suportar uma grande variedade de aplicações que requerem esses dados. Conseqüentemente, um outro importante requerimento é suportar diferentes instrumentos produzindo dados síncronos em diferentes freqüências. Isso significa que dados produzidos em diferentes períodos têm de ser varridos exatamente uma vez dentro dos *deadlines* fixados.

### 5.1 Uma comparação entre monociclo e multiciclo

No método clássico, somente um ciclo é usado e um simples tipo de multiplexação é empregado para varrer as variáveis. Chama-se este método de organização em monociclo. A duração de um ciclo é fixado e cada variável é alocada em um particular *slot* no ciclo. Este sistema é eficiente se todas as variáveis têm o mesmo período. Se o período não é constante e todas as variáveis varridas dentro de um mesmo ciclo, existirá duas grandes desvantagens: a duração do ciclo não é ótima pois tem de acomodar todas as variáveis e as variáveis de maior período são varridas mais freqüentemente do que preciso. Segundo RAJA (1993a) desta forma a largura de banda do meio de comunicação não é eficientemente utilizada.

De acordo com KUMARAN (1989) uma solução alternativa é organizar a transferência de dados em multiciclos, e tal solução foi adotada pela tecnologia Fieldbus. Em multiciclos as variáveis são varridas uma vez no período. O seguinte exemplo ilustra a diferença entre mecanismos de varredura em monociclo e multiciclo. Considere um sistema com 6 variáveis síncronas, A, B, C, D, E e F com ciclos  $T_1$  para A e B,  $T_2$  para C e D, e  $T_3$  para E e F. Onde  $T_1$  é o menor ciclo,  $T_2$  é duas vezes maior que  $T_1$  ( $T_2=2T_1$ ), e  $T_3$  é três vezes  $T_1$  ( $T_3=3T_1$ ).

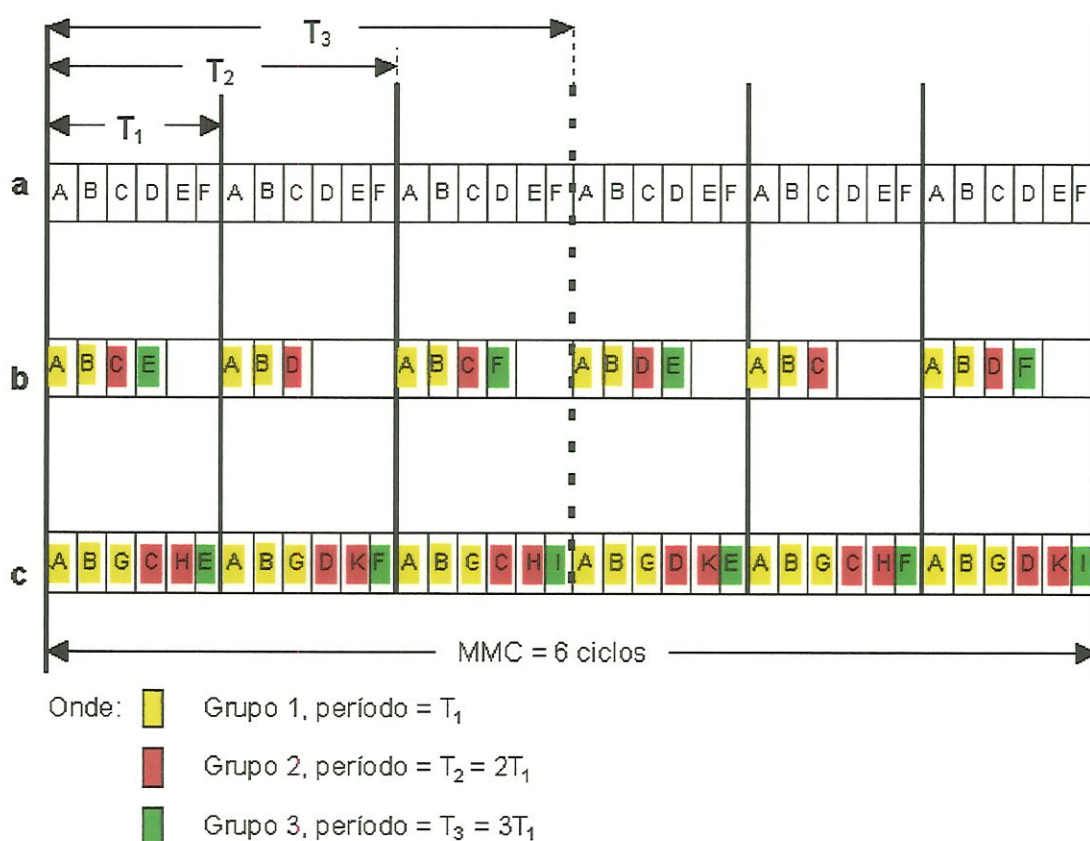


FIGURA 5.1 Mecanismos de monociclos e multiciclos.

Pode ser visto na figura anterior, uma comparação entre os dois mecanismos de varredura para seis ciclos, depois dos quais a seqüência é repetida. Separamos numa seqüência de três linhas, a, b e c. Na linha a é mostrada a organização em monociclo, o ciclo básico acomoda todas as seis variáveis e elas são varridas todos os ciclos independente da duração de seus próprios ciclos. Na linha b é mostrada uma seqüência de varredura organizada em multiciclo, onde pode ser visto que as variáveis C e D são varridas uma vez durante  $T_2$  e, E e F são varridas uma vez durante  $T_3$ . Desta forma, somente são varridas em cada ciclo somente as variáveis necessárias, possibilitando o uso dos *slots* vazios para acomodar variáveis adicionais: p.e. G, H, I e K, como visto na linha c. De acordo com FRANCO (1995) este tempo (*slots*) dentro do microciclo que não é utilizado pelas variáveis síncronas pode ser utilizado pelas mensagens assíncronas.

O menor período entre os períodos das variáveis corresponde ao microciclo ou ciclo primário, correspondendo a  $T_1$  no caso anterior. Segundo RAJA (1993a) os ciclos secundários são todos os períodos definidos como múltiplos inteiros do ciclo primário. O maior ciclo secundário do Fieldbus é chamado macrociclo, onde no caso citado corresponde a  $6T_1$  que vem a ser o MMC (Mínimo Múltiplo Comum) de  $T_1$ ,  $2T_1$  e  $3T_1$ .

## 5.2 Cálculo de tempos envolvidos

Cada ciclo primário é dividido em duas fases para tratar do tráfego assíncrono e síncrono. Durante cada fase um certo número de *slots* é designado e o tempo do *slot* é o tempo necessário para varrer uma variável, concluir uma transferência de dado. Pode-se considerar que o tempo de *slot* é constante porque o duração máxima da transferência de dados é limitada. Esta suposição é justificada porque numa rede de *hard real-time* as transferências de dados são geralmente curtas e o comprimento (duração) varia entre 8 e 32 bits. Segundo RAJA (1993a) o exato tempo do *slot* é calculado através da fórmula a seguir.

$$t_{rr} = \frac{l_{req} + l_{rep}}{bps} + t_{ret} + t_{log}$$

onde:  $t_{rr}$  = tempo de transmissão de dado por varredura

$l_{req}$  = comprimento em bits da varredura necessária

$l_{rep}$  = comprimento do dado em bits

$bps$  = taxa de transferência do barramento, largura de banda da rede (bits/s)

$t_{ret}$  = tempo de setup do mestre para compor a próxima varredura

$t_{log}$  = tempo de processamento da variável antes de envio do dado

Deste modo, se o número máximo de variáveis a serem varridas na fase síncrona é  $N_{smax}$ , então a duração máxima da fase síncrona é:

$$T_{smax} = N_{smax} * t_{rr}$$



A duração do ciclo  $T_1$  é fixada e dada pelo menor período ocorrido.  $N_{smax}$  tem de ser otimizado de tal forma que o número máximo de variáveis seja varrido em um ciclo e, ao mesmo tempo a duração do ciclo  $T_{smax}$  seja mantida mínima. Mantendo a janela síncrona tão mínima quanto possível, o número máximo de *slots* para as mensagens assíncronas será liberado e ao mesmo tempo assegura que todo o dado síncrono seja transferido antes do *deadline*. Segundo RAJA (1993a) a duração da fase assíncrona, ocupando  $N_{amax}$  *slots*, será:

$$T_{amax} = T_1 - T_{smax}$$

É mostrado na figura a seguir como cada ciclo primário é dividido, ou seja duas fases ou *time-windows* para manusear o tráfego síncrono e assíncrono.

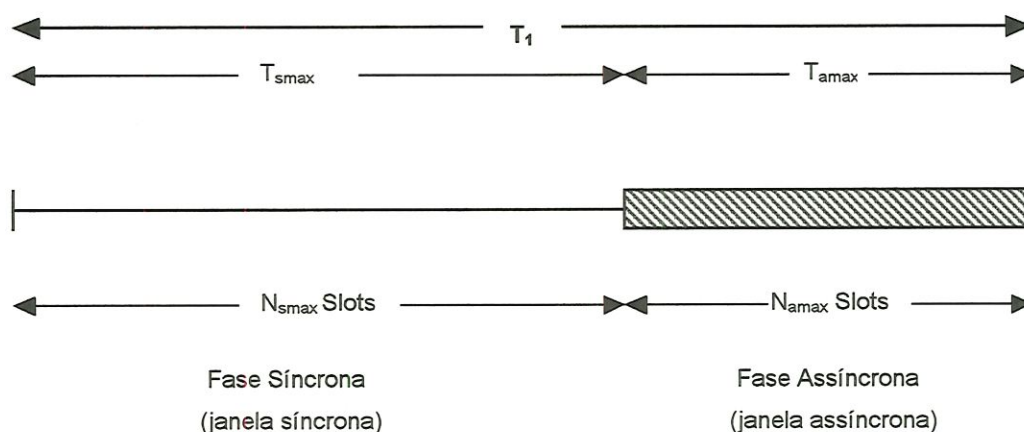


FIGURA 5.2 Divisão do ciclo primário.

É preciso, então, conhecermos o valor de  $N_{smax}$ . Para tal são utilizados algoritmos, tais algoritmos são chamados estáticos porque a duração da fase síncrona será mantida constante. O objetivo principal é explorar diferentes possibilidades de gerar uma seqüência dentro da janela de tempo síncrona, sendo ignorada a fase assíncrona.

Esses algoritmos possuem mecanismos de varredura prioritária que dá maior prioridade a grupos mais rápidos, isto é grupos com períodos menores. Baseados nos clássicos algoritmos RM (Rate Monotonic) e EDF (Earliest Deadline First) que têm sido extensivamente usados de diferentes formas para tarefas de escalonamento em sistemas de tempo real, foram adaptados para servir a

características específicas de uma varredura em multiciclo os algoritmos RMM (RM Multicycle) e EDFM (EDF Multicycle).

Considerando-se somente a fase síncrona e os comprimentos das janelas de tempo, com  $T_{smax}$  e  $T_{amax}$  fixados, os algoritmos podem ser divididos em três passos:

- cálculo do número máximo de variáveis a serem varridas no ciclo primário;
- cálculo da prioridade; e
- o algoritmo de varredura principal.

O algoritmo de varredura principal é comum a ambos os algoritmos (RMM e EDFM). Porém o cálculo do número máximo de variáveis a serem varridas e o cálculo da prioridade diferem de um para o outro, como será visto a seguir.

O cálculo do número máximo de variáveis a serem varridas no ciclo primário no caso do algoritmo RMM segundo RAJA (1993a) é dado por;

$$N_{smax} = \sum_{i=1}^{max} \left\lceil \frac{n_i}{T_i} \right\rceil$$

onde:  $n_i$  = número de variáveis no grupo  $i$

$T_i$  = período do grupo  $i$ , *deadline para* o grupo  $i$

$max$  = número máximo de grupos

O cálculo da atribuição de prioridade para o mesmo é trivial, sendo determinada pela ordem ascendente do grupo, ou seja, se  $i < j$  então o grupo  $i$  sempre tem prioridade maior do que o grupo  $j$ .

Já para o algoritmo EDFM, o  $N_{smax}$  é calculado através de um algoritmo para varredura através da distribuição uniforme de todas as variáveis de um grupo "i" num ciclo primário  $T_i$ . O número de *slots* por ciclo primário é dado por:

$$N_{smax} = n_i + \left\lceil \sum_{i=2}^{max} \frac{n_i}{T_i} \right\rceil$$

$$\text{onde: } \lceil x \rceil = \begin{cases} 0 & \text{se } x \leq 0 \\ x & \text{se } x \in \mathbb{I} \\ \text{Int}(x) + 1 & \text{se } x > 0; x \in \mathbb{R} \end{cases}$$

$n_i$  = número de variáveis no grupo  $i$

$T_i$  = período do grupo  $i$ , *deadline* para o grupo  $i$

$\max$  = número máximo de grupos

$\text{Int}(x)$  = maior inteiro menor que  $x$

O número de variáveis por grupo de varredura em um ciclo primário é dado pela média:

$$a_i = \frac{n_i}{T_i}$$

Depois de  $T_i$  ciclos, teremos  $n_i$  variáveis a serem varridas, onde:

$$n_i = T_i * a_i$$

De acordo com RAJA (1993) este algoritmo de Distribuição Uniforme assegura que o requerimento do multiciclo básico seja satisfeito.

Ainda de acordo com RAJA (1993), com relação ao cálculo da atribuição de prioridade, este algoritmo determina que a prioridade é para o grupo com o *deadline* mais avançado, e no caso discreto para o grupo com menor número de ciclos remanescentes.

Agora é apresentada a parte comum a ambos os algoritmos, o algoritmo de varredura principal, que, segundo RAJA (1993) é formulado como segue:

```
loop  
  ( * fase síncrona * )  
  varredura de todas as variáveis do grupo 1  
  repeat  
    pegar número de grupo com prioridade mais alta  
    if ( resta variáveis para ser varridas no grupo prioritário ) then  
      if ( variáveis remanescentes do grupo prioritário  $\leq N_{smax}$  ) then  
        varrer todas as variáveis do grupo prioritário  
      else  
        varrer  $N_{smax}$  variáveis do grupo prioritário  
      endif  
    endif  
  until (  $N_{smax}$  variáveis terem sido varridas )  
  ( * update para o próximo ciclo * )  
  para todos os grupos fazer  
    if ( o último ciclo do grupo ) then  
      reset o número de variáveis a serem varridas para  $n_i$   
    endif  
endloop
```

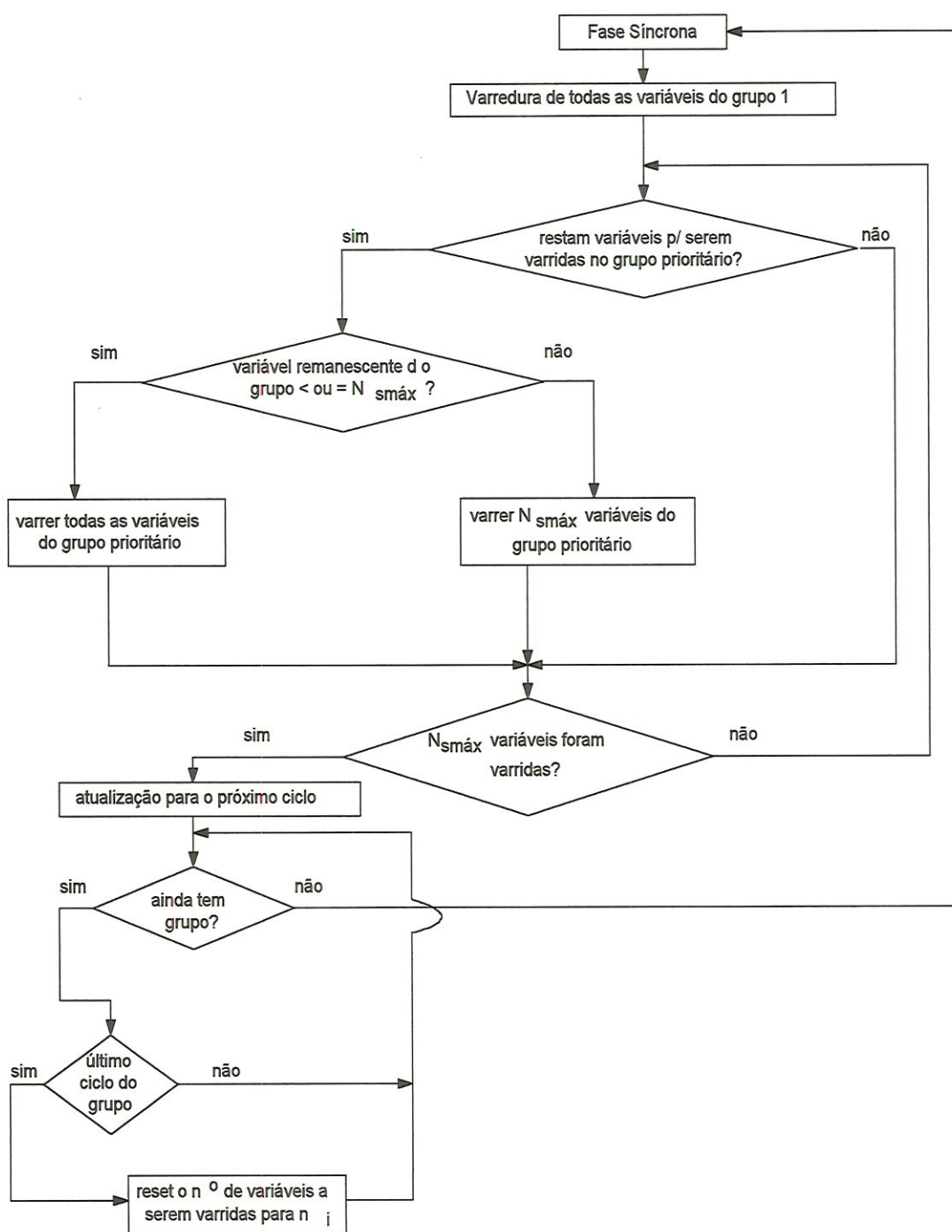


FIGURA 5.3 Fluxograma do algoritmo de varredura principal.

## **6. Procedimento para medida de tempo em uma rede Fieldbus**

Para aplicações em projetos de rede Fieldbus a questão do tempo é de fundamental importância, pois o conhecimento dos tempos necessários para a realização dos aplicativos de controle é que dá condições de se fazer a devida configuração da comunicação na rede de acordo com os objetivos desejados, ou seja, identificação dos blocos funcionais da aplicação, alocação dos blocos funcionais nos dispositivos, distribuição do macrociclo para cada dispositivo, entre outros, atendendo as exigências temporais envolvidas no processo. Com isso percebe-se que, uma adequada utilização de uma rede Fieldbus numa aplicação do usuário depende do atendimento aos tempos exigidos pela aplicação, que por sua vez é função do escalonamento da comunicação entre os dispositivos acoplados à rede. Sendo assim, para um adequado escalonamento da comunicação na rede é necessário algoritmos de escalonamento temporal além do conhecimento dos tempos envolvidos no processo.

Conhecendo-se tal questão é realizado um procedimento para medida de tempo em uma rede Fieldbus, desenvolvida no Laboratório de Eletrônica Embarcada do Departamento de Engenharia Mecânica, constituída por um mestre e um escravo, onde utiliza-se um aplicativo que simula o controle de um motor de dois tempos com ignição por centelha. Esse procedimento visa auxiliar o escalonamento temporal em uma rede Fieldbus através do fornecimento dos tempos necessários.

### **6.1 Apresentação do aplicativo de simulação em rede**

Para realizar a simulação em rede utiliza-se os dados reais de um motor dois tempos com ignição por centelha obtidos em laboratório. Trata-se de um ensaio de potência efetiva líquida de um motor JOHNSON RD32C (modificado) funcionando a gasolina, são utilizados dois diferentes valores de potência; um a carga parcial (50% da potência) e outro a plena carga (100% da potência). Os valores utilizados são mostrados a seguir.

Motor de dois tempos c/ ignição por centelha													
A carga parcial (50% potência)													
constante do dinamômetro K=1594,18													
cilindrada do motor V=737 cm <sup>3</sup>													
densidade do combustível D=0,780 g/cm <sup>3</sup>													
leitura barométrica corrigida B=697 mmHg													
Velocidade angular	n	rpm	1500	2000	2500	3000	3500	4000	4400	4500	5000	5250	5500
Momento de força observado	M <sub>o</sub>	m.Kgf	5,08	6,67	6,85	6,73	7,28	7,19	7,00	7,09	6,86	6,93	6,20
Potência efetiva observada	N <sub>eo</sub>	cv	4,05	6,85	8,80	10,49	13,20	14,79	16,12	16,83	17,95	18,70	17,87
Massa de combustível medida	m	g	100	100	100	100	100	100	100	100	100	100	100
Consumo de combustível observado	Q <sub>o</sub>	g/h	1822,90	2978,38	3692,48	4209,63	5060,88	5754,78	6588,24	7223,43	8043,39	8671,19	8586,53
Consumo de ar corrigido	C <sub>m</sub>	Kg/h	52,15	65,85	72,19	87,86	97,72	100,71	108,71	110,35	112,00	112,02	108,71
Motor de dois tempos c/ ignição por centelha													
A plena carga (100% potência)													
constante do dinamômetro K=1594,18													
cilindrada do motor V=737 cm <sup>3</sup>													
densidade do combustível D=0,780 g/cm <sup>3</sup>													
leitura barométrica corrigida B=697 mmHg													
Velocidade angular	n	rpm	1500	2000	2500	3000	3500	4000	4400	4500	5000	5250	5500
Momento de força observado	M <sub>o</sub>	m.Kgf	10,07	13,25	13,82	13,57	14,53	14,34	14,07	14,15	13,76	13,87	12,67
Potência efetiva observada	N <sub>eo</sub>	cv	7,86	13,80	18,00	21,20	26,50	29,67	32,27	33,16	35,83	37,95	36,30
Massa de combustível medida	m	g	100	100	100	100	100	100	100	100	100	100	100
Consumo de combustível observado	Q <sub>o</sub>	g/h	2515,2	4133,1	5011,2	5365,7	6617,0	7574,75	8664,5	9357,7	10663,0	11730,3	11325,6
Consumo de ar corrigido	C <sub>m</sub>	Kg/h	54,90	70,05	77,62	96,55	109,80	115,48	124,95	126,84	128,74	128,74	124,95

TABELA 6.1 Ensaio de potência efetiva líquida em um motor Johnson RD32C (modificado) funcionando a gasolina, a 50% e 100% da carga.

A simulação em rede é realizada através de dois microcomputadores que fazem o papel de um mestre e um escravo, para o controle do consumo de Ar/Combustível com a variação da velocidade angular e a variação da potência efetiva em função da variação da velocidade angular, como poderá ser visto na apresentação gráfica gerada pelo escravo (figura 6.1). Tendo o mestre como gerenciador do processo, realizando o envio de tarefas através de uma rede Fieldbus programada utilizando-se o protocolo PROFIBUS. E, o escravo realizando a simulação, apresentando os resultados e respondendo ao mestre.

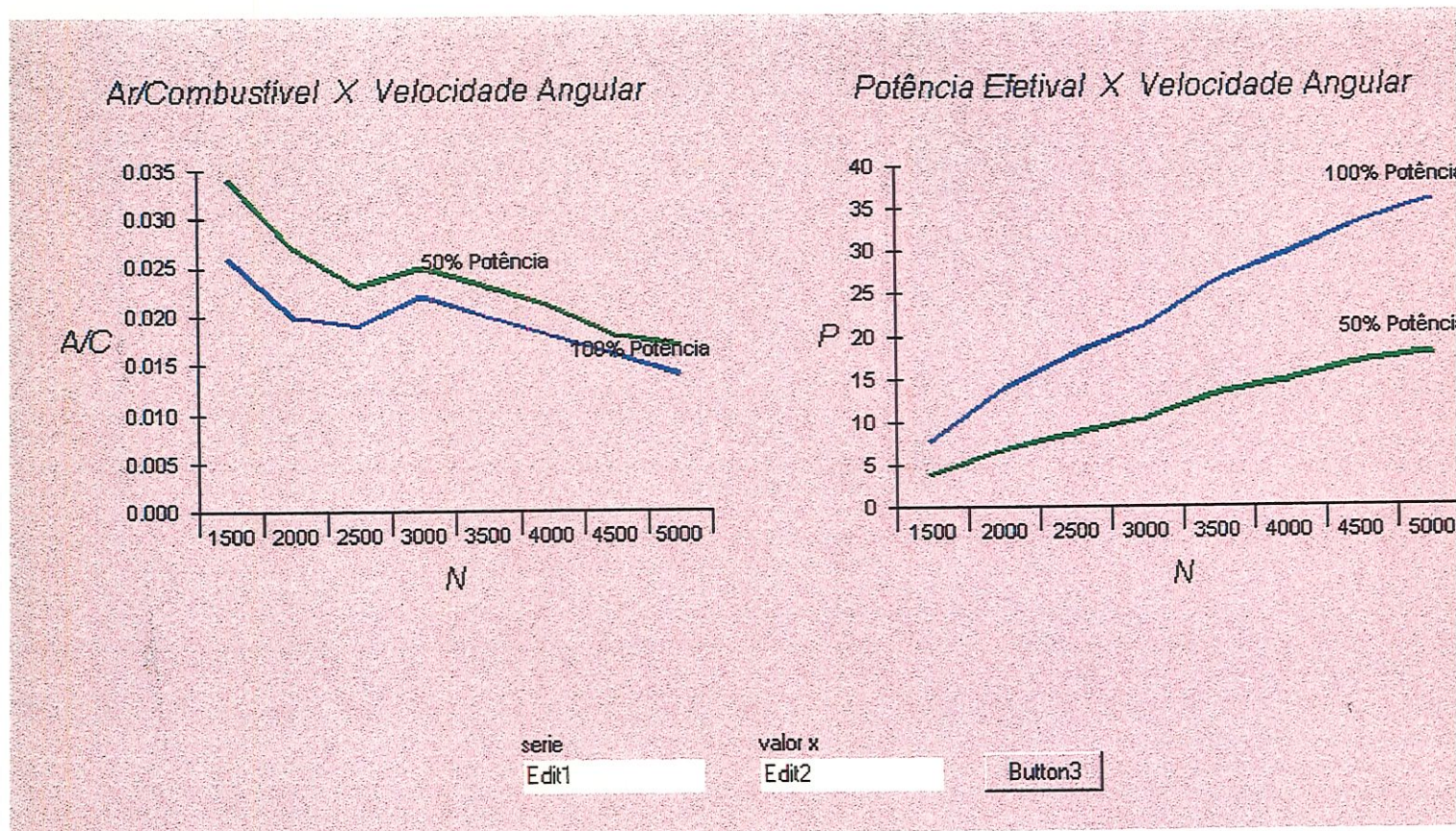


FIGURA 6.1 Curvas de consumo Ar/Combustível e Potência efetiva em função da variação da velocidade angular, apresentadas pelo escravo.



O programa de simulação desenvolvido, utilizando-se a ferramenta de desenvolvimento Borland C++Builder, será apresentado através de um diagrama das partes básicas (ver figura 6.2 a seguir) já que a inclusão das listagens do mesmo não contribuiria para o entendimento da simulação em rede, ao mesmo tempo em que, para este trabalho, é de interesse uma compreensão geral do aplicativo e não como o mesmo foi implementado, daí não ser apresentada a implementação dos programas de rede, mestre ou escravo.

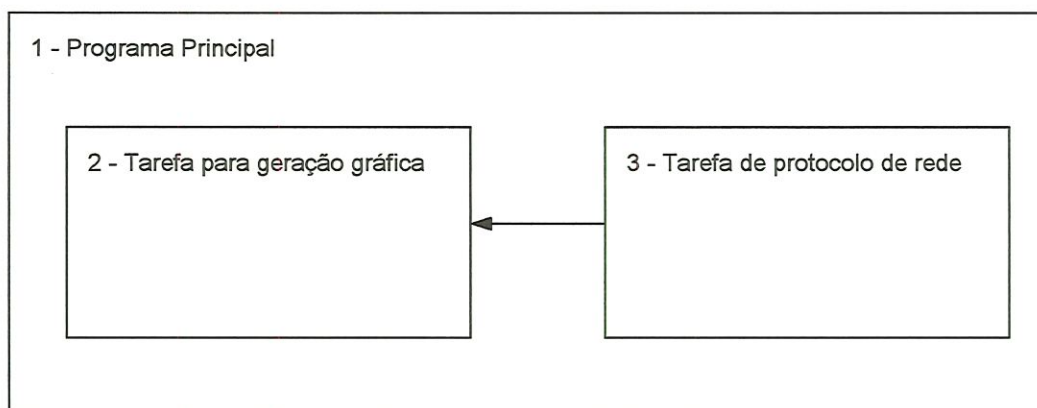


FIGURA 6.2 Diagrama das partes básicas do programa de simulação contido no escravo.

O programa principal tem a função de realizar a inicialização do sistema e a interface com o usuário, através da interação via mouse, teclado ou vídeo. A tarefa para geração gráfica é um subsistema encarregado de processar os valores recebidos, apresentando os mesmos de forma gráfica. Já o protocolo de rede é responsável pelo monitoramento das mensagens vindas do mestre via rede e por dar as devidas respostas, estabelecendo assim o protocolo de comunicação.

Como pode ser visto no diagrama a tarefa para geração gráfica recebe informações vindas da tarefa de protocolo de rede. Na verdade estes dois subsistemas se comunicam através de uma variável que funciona como um sinalizador, possibilitando então o sincronismo entre os dois sistemas.

A seguir vão ser apresentados os fluxogramas da tarefa para geração gráfica e da tarefa de protocolo de rede. É importante frisar que os fluxogramas apresentados mostram uma visão a nível de aplicativo. Os detalhes dos objetos gráficos utilizados no programa não são apresentados, assim como não são apresentados os detalhes da parte de gerenciamento da rede PROFIBUS.

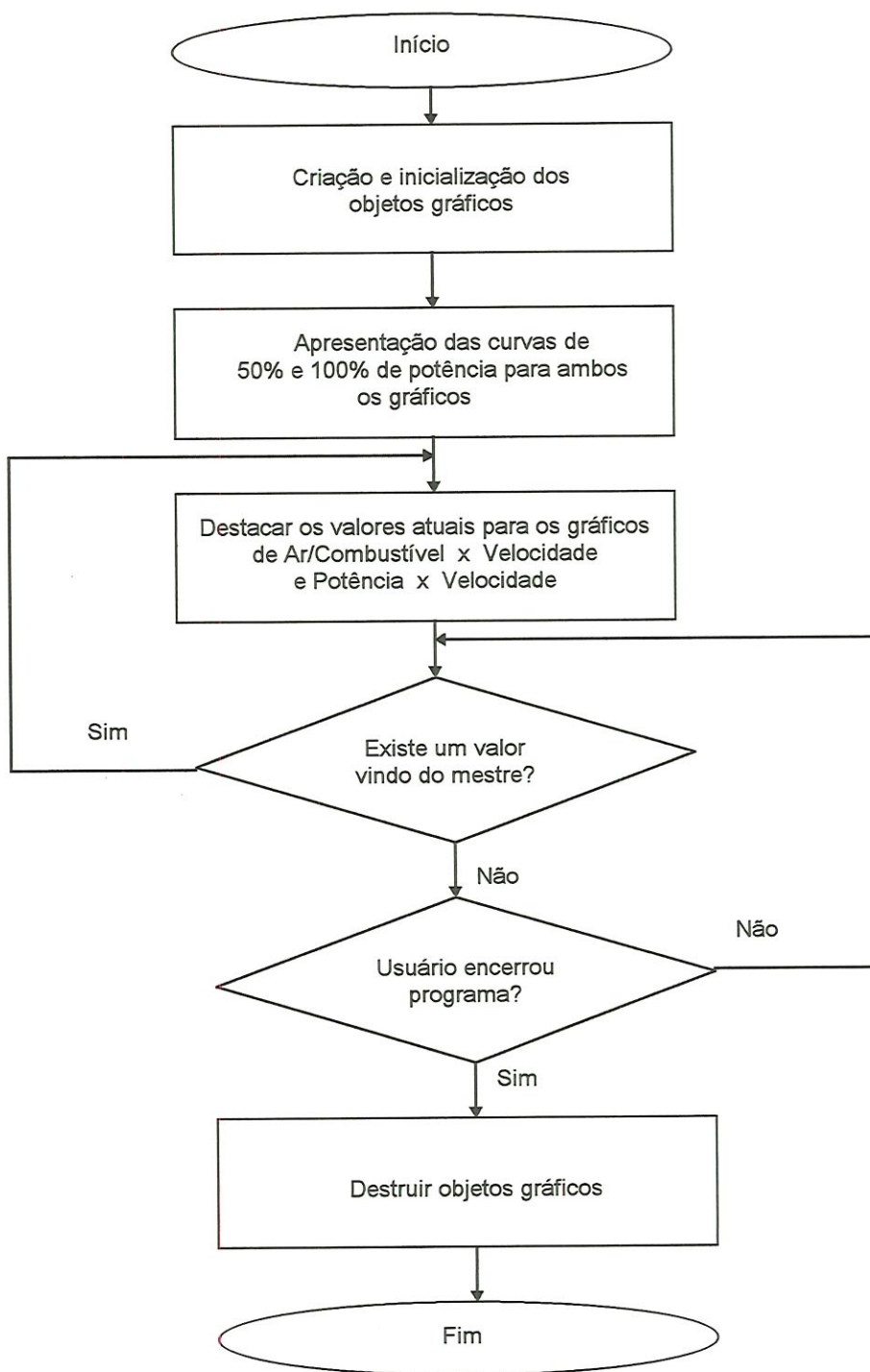


FIGURA 6.3 Fluxograma da tarefa para geração gráfica. Onde a variável **Série** representa a porcentagem de carga do motor utilizada e, a variável **X** a velocidade angular do mesmo.

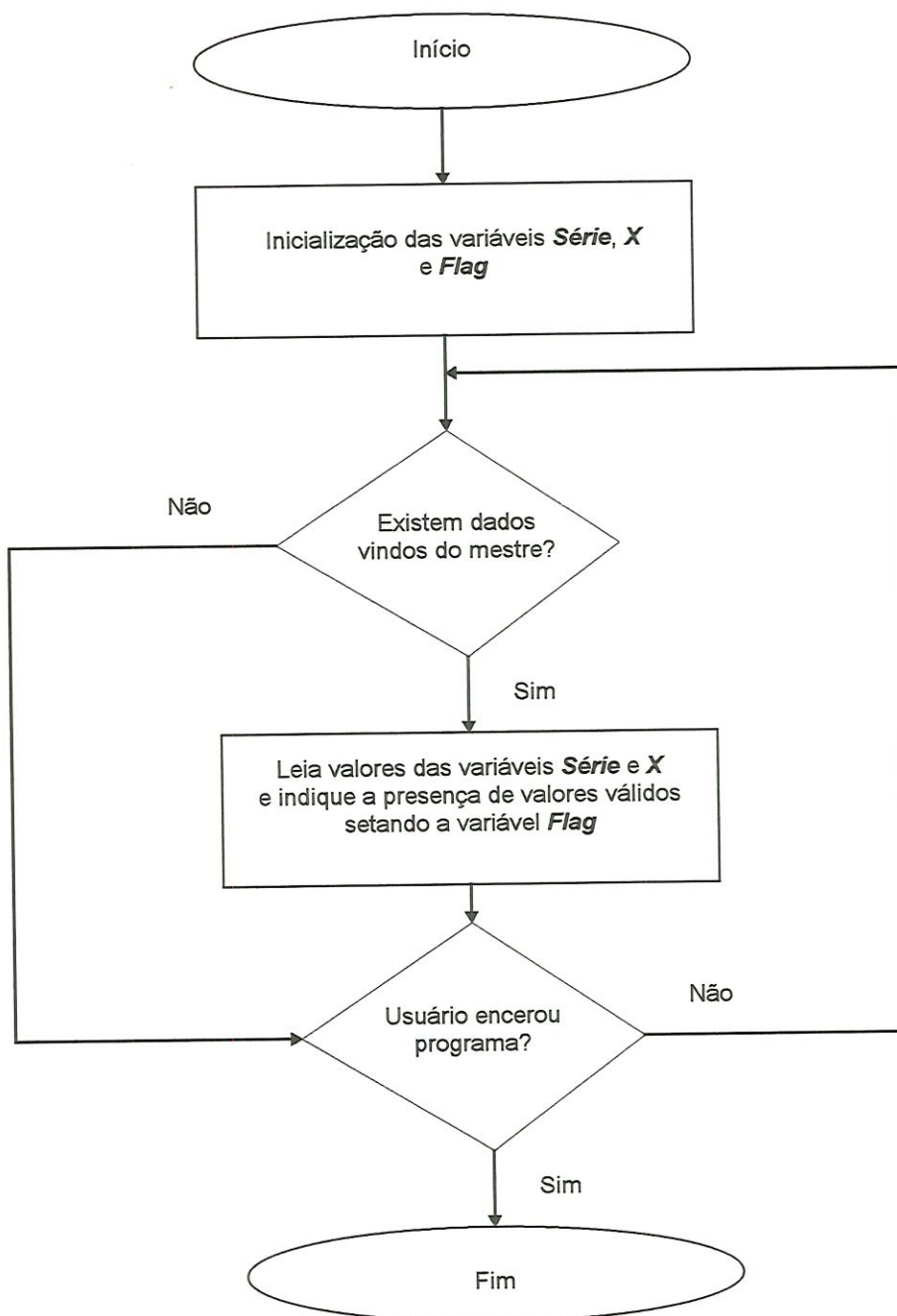


FIGURA 6.4 Fluxograma da tarefa do protocolo de rede.

## 6.2 Medida do tempo global

A medida do tempo global realizada foi feita utilizando-se um algoritmo, introduzido no programa do mestre, que permitiu a contagem do tempo a partir do relógio interno do software, um recurso da linguagem C. Tal algoritmo conta o tempo a partir do momento em que o mestre requisita um serviço até o mesmo ser respondido. Desta forma é medido o tempo de envio da requisição do serviço pelo mestre, de processamento da tarefa pelo escravo e o retorno da resposta ao mestre, isto pode ser visto na representação gráfica a seguir.

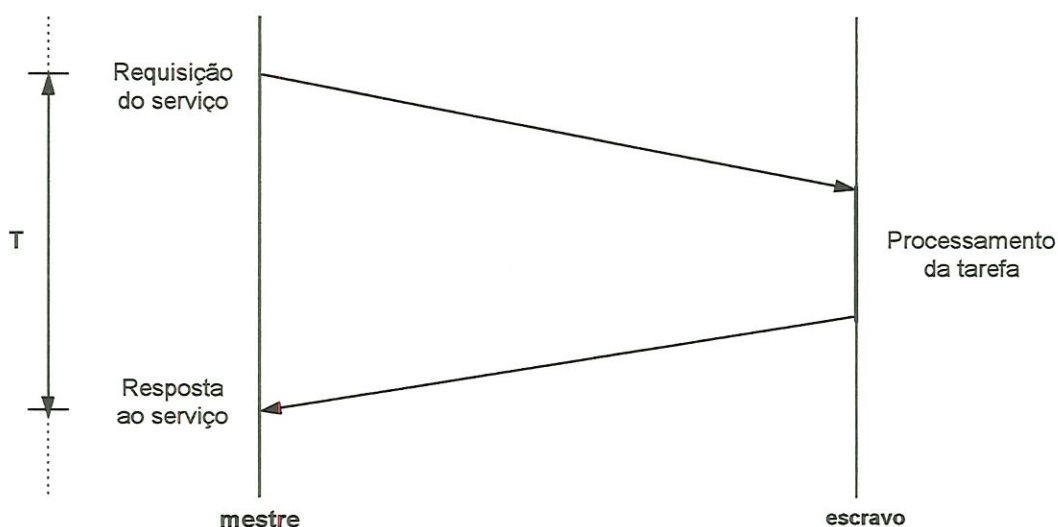


FIGURA 6.5 Representação gráfica do tempo global (  $T$  ) gasto para realização do aplicativo de simulação.

Porém, como o relógio interno trabalha numa velocidade de 18.2 ticks/s, ou seja, considerando-se  $t_{tk}$  como o intervalo de tempo entre os ticks do relógio interno tem-se que;

$$t_{tk} = \frac{1}{18,2} = 54,945ms$$

Isso significa que o intervalo de tempo entre os ticks do relógio interno é de 54.945 ms, sendo assim, esta é a menor unidade de tempo mensurável pelo mesmo. E como a taxa de transmissão da rede mostrou-se muito inferior a esse valor, para realizar a contagem do tempo foi necessário repetir algumas vezes o

processo, sendo assim foram realizadas mil repetições consecutivas do envio da requisição pelo mestre, processamento do aplicativo pelo escravo e o recebimento da resposta pelo mestre. Para obter-se uma maior precisão foi realizado esse procedimento por trinta vezes. O algoritmo em linguagem de alto nível é mostrado logo a seguir.

```
Para série = 1 até 30 faça
  Início
    Start ← tempo atual;
    Para contador = 1 até 1000 faça
      Início
        X ← valor randômico           (X é a velocidade angular)
        Série ← valor aleatório       (50% ou 100% da potência)
        Transmitir (X e Série); // comando bloqueado*
      Fim
    End ← tempo atual;
    Imprimir_no_arquivo (End - Start);
  Fim
```

A tabela na página seguinte mostra os valores obtidos para o tempo T.

---

\* Significa que enquanto a transmissão não for completada ( requisição de retorno) a linha de comando seguinte não será executada.

Repetição	Tempo Global - T (ms)
1	30.055
2	29.890
3	29.670
4	30.000
5	29.451
6	30.549
7	30.055
8	30.165
9	30.385
10	30.385
11	29.451
12	30.385
13	30.879
14	30.330
15	30.000
16	30.934
17	29.945
18	29.286
19	30.111
20	29.835
21	30.000
22	30.220
23	29.780
24	29.835
25	30.385
26	30.604
27	29.505
28	30.220
29	30.659
30	30.989

TABELA 6.2 Valores obtidos para o tempo global T.

Agora que se conhece alguns valores de tempo medidos, deve-se considerar um valor máximo para o tempo global, já que não se deve dar um tratamento estatístico (média, desvio padrão, etc...) para o mesmo, pois resultaria em valores menores que o máximo. Podendo, desta forma, gerar períodos menores que o necessário para a aplicação, assim é melhor considerar o pior caso. É claro que se deve considerar a aplicação em questão, se esta possui *deadline hard ou soft*. Já que para um *dedline hard* a violação pode levar a uma catástrofe , citando como exemplo um sistema de controle de voo, enquanto que no caso de um *deadline soft* não existe tal urgência, um atendimento dos requisitos do operador, por exemplo deve ser atendido, podendo ocasionalmente não o ser.

Diante dos valores obtidos e do aplicativo em questão é razoável considerar como valor de tempo o máximo valor obtido, que é de aproximadamente 31 ms. Com esse valor tem-se a garantia de que o aplicativo será atendido, podendo este valor ser revisto (monitorado) periodicamente para garantir a mudança do mesmo caso haja alguma variação.

Fazendo uma análise da importância da obtenção de tal variável, sem deixar de lado a importância da mesma, pode-se dizer que o fato de se obter um procedimento para medida de tempo em uma rede Fieldbus é de fundamental importância para se chegar a uma desejada configuração da comunicação na rede de acordo com as necessidades do usuário. Ou seja, tal procedimento para medida de tempo dá condições para se ter conhecimento de outros tempos envolvidos no processo como tempo de setup do mestre para compor a próxima varredura, tempo de processamento da variável antes de envio do dado, tempo de duração máxima da fase síncrona e assíncrona, entre outros.

Esse procedimento para medida de tempo também pode ser utilizado para aplicativos que exijam mais de um escravo, já que o mesmo se encontra no programa do mestre, bastando para isso configurar a rede para trabalhar com a quantidade de escravos necessária. E, desta forma pode ser feito o controle de processos mais complexos já que o procedimento utilizado pode ser estendido a uma rede com vários escravos, dando condições de configuração da comunicação na rede de acordo com as necessidades. Já que a alocação dos blocos funcionais nos dispositivos e distribuição do macrociclo para cada dispositivo depende das restrições de tempo envolvidas no processo.

## 7. Conclusão

Ao longo da evolução tecnológica que vem acontecendo neste século, em especial na área de controle de processos em busca da melhoria da eficiência da produção e qualidade do produto, presencia-se a necessidade constante da evolução dos sistemas de transmissão. Os sistemas digitais chegam em resposta a essas necessidades. E é dentro desse contexto que surge a rede de campo Fieldbus. O Fieldbus proporciona uma comunicação verdadeiramente digital, eliminando de vez o gargalo existente no padrão de loop de corrente (4-20 mA).

Para que essa nova tecnologia tenha sucesso a nível de mercado tem que haver uma normalização, ou seja, elabore-se protocolos que permitam a comunicação sem restringir o usuário a um único fabricante. Surgem vários protocolos com aplicações específicas como o PROFIBUS e FIP, porém não são normas internacionais. Os esforços nesse sentido são recompensados com o surgimento de uma norma internacional aprovada pelo IEC (International Electrotechnical Commission) e desenvolvida pela Fieldbus Foundation em conjunto com a ISA (International Society for Measurement and Control) e empresas de peso como; ABB, Allen-Bradley, Smar, entre outras.

Como em toda nova tecnologia surge várias questões que colaboram com o processo de amadurecimento da mesma. Uma questão de vital importância que está sendo exaustivamente pesquisada é o estudo de algoritmos que permitam uma otimização do escalonamento temporal em rede Fieldbus. Tais algoritmos estão sendo adaptados ao Fieldbus baseando-se em outros já utilizados pela ciência da computação para realizar o escalonamento de processadores.

Foi visando tal necessidade que esse trabalho desenvolveu-se, nele foi realizado com sucesso um procedimento para medida de tempo em uma rede Fieldbus. Utilizando-se um algoritmo introduzido no programa do mestre que permitiu a medição do tempo global que a rede utilizou para o controle de um motor. Deve ser colocado que a importância do resultado obtido não está simplesmente no valor de tempo medido, ela vai muito além disso já que de posse desse conhecimento poderão ser feitas medidas de tempo em redes com outras configurações, considerando aí variações no que diz respeito ao tamanho da rede, isto é, quantidade de escravos e mestre utilizados.



É necessário deixar claro que, mesmo não tendo condições materiais de realizar uma simulação mais complexa, que contasse com uma rede maior, o trabalho foi bem desenvolvido e obteve bons resultados. Mantendo-se, desta forma, dentro de sua filosofia inicial que é a de fornecer referências a respeito de uma tecnologia emergente e um procedimento para medida de tempo que possa auxiliar a configuração da comunicação em uma rede Fieldbus.

Pretende-se dar continuidade a este trabalho realizando-se um aplicativo que simule o controle de um navio em uma rede Fieldbus real, ou seja, será feito o controle de propulsão, estabilidade e direção através de uma rede que possa contar com mais escravos, dando condições de se implementar e analisar o desempenho de algoritmos de escalonamento temporal. Isso será possível devido ao fato de se ter a aplicação distribuída em vários dispositivos, cada dispositivo contido em um escravo, e para realizar o controle do processo serão necessárias várias aplicações como pode ser visto no exemplo da figura a seguir.

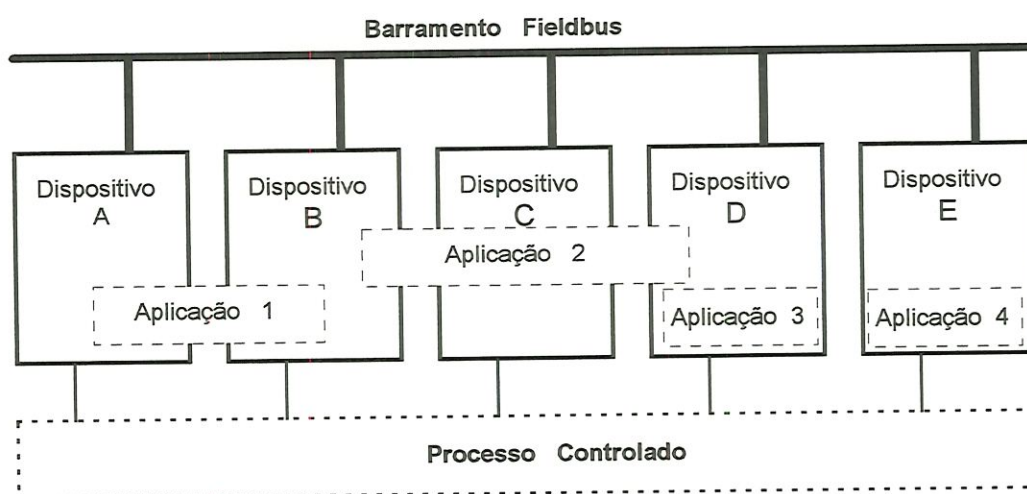


FIGURA 7.1 Exemplo de aplicações distribuídas num Fieldbus.

Para cada aplicação distribuída em mais de um dispositivo há pelo menos um link de comunicação externo que usará o barramento para que os dispositivos se comuniquem entre si. Desta forma poderá ser implementado um algoritmo de escalonamento que permita a devida comunicação na rede de acordo com as características (tempo de execução, prioridades, etc...) de cada aplicação. E, para gerenciar todo o sistema será colocado um mestre que será usado para coordenar a operação dos vários dispositivos do sistema. Tal configuração vai permitir o teste do algoritmo de escalonamento implementado em rede, já que o mesmo vai ser responsável pela distribuição do tempo na rede coordenando a seqüência de execução de blocos funcionais que compõem uma determinada aplicação, distribuídos em um mesmo dispositivo ou não, proporcionando que as saídas sejam geradas antes que as mesmas sejam requeridas como entradas pelos outros blocos funcionais.

Com uma configuração deste tipo, um mestre e vários escravos, poderão ser feitos testes do algoritmo de escalonamento programado, bem como, caso haja necessidade, a reprogramação do mesmo para um perfeito funcionamento.

## Bibliografia

- AGUIAR, M.V.C., et al. Aspectos de implementação de um servidor MMS. In: Simpósio Brasileiro de Redes de Computadores, 9., Florianópolis, 1991. Anais. Florianópolis, Imprensa Universitária da Universidade Federal de Santa Catarina, 1991. p.331-342.
- ARMSTRONG, E. Actuator & sensor coupling for reconfigurable robotic systems. Aberystwyth, 1997. Tese (Doutorado), University of Wales.
- AUDSLEY, N.C., et al. Deadline monotonic scheduling theory and application. Control Engineering Practice, v.1, n.1, p.71-78, 1993.
- AUDSLEY, N.C. Deadline monotonic scheduling. Technical Report YCS146, Department of Computer Science, University of York, Set, 1990a.
- AUDSLEY, N.C.; BURNS, A. Real time system scheduling. Technical Report YCS134, Department of Computer Science, University of York, Jan., 1990b.
- BABB, M. Smart transmitters in distributed control: new performance and benefits. Control Engineering, v.33, n.9, p.120-123, Sept., 1986.
- BEAUCHAMP, K.G. Computer communications. 2.ed. Grã-Bretanha, Chapman and Hall, 1990.
- BROWN, G.R. Second generation smart transmitters: even better control. In: ISA/89 - International Conference and Exhibition, Philadelphia, 1989. Proceedings. Research Triangle PK, NC, USA, ISA, 1989. Part 4 (of 4), p.1625-1630. (Advances in Instrumentation Proceedings, 44)
- BURNS, A. Scheduling hard real-time systems: a review. Software Engineering Journal, v.6, n.4, p.116-128, May, 1991.
- CAVALIERI, S. Optimization of acyclic bandwidth allocation exploiting the priority mechanism in the fieldbus data link layer. IEEE Transactions on Industrial Electronics, v.40, n.3, June, 1993.
- CENA, G., et al. Na object oriented model for the FIP protocol. In: IECON'94 20 th International Conference on Industrial Electronics, Control and Instrumentation, Bologna, 1994. Proceedings. Bologna, Italy, IEEE. v.2, p.1214-1219.
- DAMSKER, D.J. Assessment of industrial data network standards. IEEE Transactions on Energy Conversion, v.3, n.2, p.199-204, June, 1988.

- DI STEFANO, A.; MIRABELLA, O. Evaluating the field bus data link layer by a Petri Net-based simulation. IEEE Transaction on Industrial Electronics, v.38, n.4, Aug., 1991.
- FRANCO, L.R.H.R. Escalonamento da comunicação no fieldbus: garantia do atendimento às exigências temporais em sistemas "hard real time". In: Conferência Internacional ISA SHOW Brasil/95, São Paulo, 1995. Anais. p.337-345.
- FRANKLIN, G.F. et al. Feedback control of dynamic systems. 3.ed. New York, Addison-Wesley, 1994.
- FURNESS, H. Fieldbus series, part 1. Digital communications provides... Control Engineering, v.41, n.1, p.23-25, Jan., 1994.
- FURNESS, H. Fieldbus: the differences start from the bottom up. Control Engineering, v.41, n.3, p.75-77, Mar., 1994.
- GIOZZA, W.F., et al. Redes locais de computadores. 1.ed. Rio de Janeiro, McGraw-Hill, 1986.
- KUMARAN, S.; DECOTIGNIE, J.D. Multicycle operations in a fieldbus: Application layer implications. Proceedings. IECON, 1989.
- LAWLER, E.L.; MOORE, J.M. A functional equation and its application to resource allocation and scheduling problem. Management Science, v.16, p.77-84, 1969.
- LEVITI, P. IEC/ISA Fieldbus DLL tutorial. Italian Eletro-Technical Committee. Apr., 1996, p.1-54.
- LUTZE, U. (1995) Reduce control system complexity with fieldbus technology. Control Engineering. <http://www.manufacturing.net/magazine/ce/archives/1995/11/issues/na/11b804.htm> (19 May).
- McGUFFIN, L.J., et al. MAP/TOP in CIM distributed computing. IEEE Network, v.2, n.3, May, 1988.
- MESSINA, G.; TRICOMI, G. Manufacturing communication architectures. Computers in Industry, v.13, n.4, p.285-293, Mar., 1990.
- MUÑOZ, G.T. Protocolo fieldbus, estado de arte, características técnicas. In: Conferência Internacional ISA SHOW Brasil/97, São Paulo, 1997. Anais. Arquivo 6.DOC, Disquete 1.
- NETO, E.F.S., et al. Comunicação em chão de fábrica: solução fieldbus. In: Conferência Internacional ISA SHOW Brasil/95, São Paulo, 1995. Anais. p.233-241.

- O'BRIEN, C.J. Expanding the digital control networks with smart transmitters. In: ISA'90 - Internacional Conference and Exhibition, New Orleans, 1990. Proceedings. Research Triangle PK, NC, USA, ISA, 1990. Part 4 (of 4), p.1937-1949. (Advances in Instrumentation Proceedings, 45)
- OGATA, K. Engenharia de controle moderno. 2.ed. Rio de Janeiro, Prentice-Hall, 1993.
- PELLE, D., et al. Desenvolvimento de um fieldbus para interligação de instrumentos inteligentes. In: CINISA/COBISA, 1., São Paulo, 1991. Anais. Fundação CERTI divisão de automação integrada, 1991.
- PEREIRA, A.P. Fieldbus a nova era da automação industrial. In: Conferência Internacional e exposição ISA SHOW Brasil/95, São Paulo, 1995. Anais. p.377-383.
- RAJA, P.; NOUBIR, G. Static and dynamic polling mechanisms for fieldbus networks. Operating Systems Review, v.27, n.3, p.34-45, July, 1993a.
- RAJA, P., et al. Analysis of polling protocols for Fieldbus networks. Computer Communication Review, v.23, n.3, p.69-89, Apr., 1993.
- RAMAMRITHAN, K.; STANKOVIC, J.A. Dynamic task scheduling in hard real-timedistributed systems. IEEE Software, p.65-75, July, 1984.
- RICH, E. Inteligência artificial. 1.ed. Rio de Janeiro, McGraw-Hill, 1994.
- SANTOS, J.L.dos Implementação e uso de rede fieldbus para monitoramento e controle de aplicação em tempo real. São Carlos, 1996. Dissertação (Mestrado), Escola de Engenharia de São Carlos, Universidade de São Paulo.
- SHIN, K.G. Design and evaluation of real-time communication for fieldbus-based manufacturing systems. IEEE Transaction on Robotics and Automation, v.12, n.3, p.357-367, June, 1996.
- SILVA, R.D.da Comunicação de dados em ambiente industrial: um protocolo para automação e controle em tempo real. São Carlos, 1996. Dissertação (Mestrado), Escola de Engenharia de São Carlos, Universidade de São Paulo.
- SPRUNT, B. et al. Aperiodic task scheduling for hard real-timesystem. The Journal of Realtime Systems. 1.27.60, p.27-60, 1989.
- STANKOVIC, J.A., et al. Evaluation of a flexible task scheduling algorithm for distributed hard real-time systems. IEEE Transactions on Computers. v.C-34, n.12, p.1130-1143, Dec., 1985.

- STAROSWIECKI, M.; BAYART, M. Models and languages for the interoperability of smart instruments. Automatica, v.32, n.6, p.859-873, June, 1996.
- TANENBAUM, A.S. Computers networks. 3.ed. New Jersey, Prentice-Hall, 1996.
- TINDELL, K., ET AL. Allocating hard real-time task (na NP-hard problem made easy). The Journal of Real-Time Systems. v.4, n.11, p.145-165, June, 1992.
- VITA, (1996) A family of three Fieldbus solutions.  
[Http://www.vita.com/mar96vj/vjmar96.html#anchor1035783](http://www.vita.com/mar96vj/vjmar96.html#anchor1035783). (Mar)
- XU, J.; PARNAS, D.L. Scheduling processors with release times, deadlines, precedence and exclusion relations. IEEE Transactions on Software Engineering. v.16, n.3, p.360-369, Mar., 1990.