

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Análise e Estimativa de Desempenho de  
Redes Profibus.**

por

MARCOS PAULO MORO

Dissertação submetida à avaliação como requisito parcial  
para a obtenção do grau de Mestre em Ciência da Computação

Prof. Dr. João César Netto  
Orientador

Porto Alegre, agosto de 2002

**CIP – CATALOGAÇÃO NA PUBLICAÇÃO**

Moro, Marcos Paulo

Análise e Estimativa de Desempenho de Redes Profibus / por Marcos Paulo Moro. – Porto Alegre: PPGC da UFRGS, 2002.

104 f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2002. Orientador : Netto, João César.

1. Avaliação de Desempenho. 2. Automação Industrial. 3. Protocolos. I. Netto, João Cesar. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Haro

Dedico esta dissertação

A Flávia, minha esposa,  
*que esteve ao meu lado em todos os momentos  
difíceis me dando apoio, carinho e tendo  
compreensão durante os quatro meses de separação,  
que foram necessários para a conclusão da minha  
dissertação.*

Aos meus três filhos, Marcos Caio, Gabriella e  
Bárbara, *que vieram ao mundo trazendo alegria a  
minha vida e tornando mais fácil a minha caminhada.*

## Agradecimentos

Primeiramente a DEUS, por ter me dado forças de enfrentar esta longa caminhada ausente de casa e de todos.

Ao meu segundo pai, José Carlos que com seu espírito sereno e companheiro tem sempre me acompanhado e incentivado as minhas jornadas.

A minha mãe, minha irmã e a minha tia Neuza pessoas que tanto amo, obrigado por tudo e por toda a força dada nesses últimos dois anos.

À professora Rosa Dede'a, reitora da UNIGRAN, que permitiu o meu afastamento em tempo integral para frequentar o curso de pós-graduação em Ciência da Computação do Instituto de Informática da UFRGS.

À professora Suzana, Pró-Reitora de ensino, por ter viabilizado as condições para que eu me ausentasse da instituição sempre me dando e apoio e incentivo.

Ao professor João Netto. A sua capacidade, aliada ao seu espírito sereno e aberto, propiciaram o clima ideal para o meu trabalho de pesquisa. Foi uma grande satisfação estudar sob a sua orientação.

Os professores do curso de ciência da computação da UNIGRAN, principalmente ao Prof. Marco Aurélio e a Prof<sup>a</sup>. Raquel pelo apoio dado durante a minha ausência.

Aos monitores do curso de ciência da computação principalmente ao Estevam.

Ao meu amigo, Lincoln, pelo companheirismo e pelos interessantes debates que mantivemos nos últimos quatro meses.

Ao meus colegas do grupo de estudo, principalmente ao meu amigo Cláudio, que foi um grande companheiro durante esses últimos 3 anos.

Às bibliotecárias, funcionárias e funcionários da biblioteca do Instituto de Informática.

E a todos aqueles que de certa forma vieram a contribuir para a concepção deste trabalho. Valeu pessoal !!!

## Sumário

<b>Lista de Abreviaturas</b> .....	8
<b>Lista de Símbolos</b> .....	10
<b>Lista de Figuras</b> .....	11
<b>Lista de Tabelas</b> .....	12
<b>Resumo</b> .....	13
<b>Abstract</b> .....	14
<b>1 Introdução</b> .....	15
<b>1.1 Objetivos da Pesquisa</b> .....	16
<b>1.2 Motivação</b> .....	17
<b>1.3 Estrutura do texto</b> .....	17
<b>2 Integração dos meios de comunicação</b> .....	19
<b>2.1 O Modelo de referência RM-OSI/ISO</b> .....	20
<b>2.1.1 Transmissão de dados no modelo RM-OSI</b> .....	21
<b>2.2 MAP (Manufacturing Automation Protocol)</b> .....	22
<b>3 Padrões de redes de campo</b> .....	25
<b>3.1 Foundation FIELDBUS</b> .....	25
<b>3.1.1 Características funcionais</b> .....	27
<b>3.1.2 Nível físico</b> .....	27
<b>3.1.3 Nível de enlace (Data Link Layer)</b> .....	28
<b>3.1.4 Nível de aplicação (Application Layer)</b> .....	29
<b>3.1.5 Nível do usuário (User Layer)</b> .....	29
<b>3.2 PROFIBUS (Process Fieldbus)</b> .....	29
<b>3.2.1 Características funcionais</b> .....	30
<b>3.2.2 Nível físico</b> .....	31
<b>3.2.2.1 Interface entre a camada física e a camada de link de dados (FDL)</b> .....	32
<b>3.2.2.2 Redundância da camada física e do meio de transmissão</b> .....	32
<b>3.2.3 FDL (Fieldbus Data Link Layer)</b> .....	33
<b>3.2.3.1 Método de acesso ao meio de transmissão</b> .....	33
<b>3.2.3.2 Lista das estações ativas</b> .....	35
<b>3.2.3.3 Adição e remoção de estações a quente</b> .....	35

3.2.3.4	Inicialização e reinicialização do anel lógico .....	36
3.2.3.5	Estrutura dos quadros .....	36
3.2.3.6	Endereçamento de estações .....	37
3.2.3.7	LSAP .....	38
3.2.3.8	Prioridade de mensagens .....	38
3.2.3.9	O mecanismo de resposta imediata .....	39
3.2.3.10	Serviços de transferência de dados da camada de Enlace .....	39
3.2.3.11	Envio de dados sem reconhecimento.....	41
3.2.3.12	Envio de dados com reconhecimento .....	41
3.2.3.13	Envio/Pedido de dados com resposta .....	42
3.2.3.14	Envio/Pedido cíclico de dados com resposta.....	43
3.2.3.15	Status do controlado FDL.....	44
<b>3.2.4</b>	<b>Gerenciador das Camadas 1 e 2 (FMA 1/2) .....</b>	<b>47</b>
<b>3.2.5</b>	<b>Camada de aplicação.....</b>	<b>48</b>
3.2.5.1	A camada FMS .....	48
3.2.5.2	Objetos.....	49
3.2.5.3	O modelo VFD (Virtual Fieldbus Device) .....	49
3.2.5.4	Dicionário de objetos.....	50
3.2.5.5	Serviços .....	50
3.2.5.6	Lista de relações de comunicação .....	50
3.2.5.7	Relações de comunicação.....	51
<b>3.3</b>	<b>Foundation Fieldbus versus Profibus.....</b>	<b>52</b>
<b>4</b>	<b>Tempos de supervisão do PROFIBUS.....</b>	<b>54</b>
<b>4.1</b>	<b>Tempo de rotação desejado.....</b>	<b>54</b>
<b>4.2</b>	<b>Tempos de Controle do PROFIBUS .....</b>	<b>55</b>
<b>4.2.1</b>	<b>Tempos medidos .....</b>	<b>55</b>
<b>4.2.2</b>	<b>Tempos configuráveis.....</b>	<b>55</b>
<b>4.2.3</b>	<b>Tempos calculados.....</b>	<b>57</b>
<b>4.2.4</b>	<b>Tempo do ciclo de token.....</b>	<b>58</b>
<b>4.2.5</b>	<b>Tempo de ciclo de mensagens .....</b>	<b>59</b>
<b>4.2.6</b>	<b>Taxa de mensagem do sistema.....</b>	<b>59</b>
<b>4.2.7</b>	<b>Tempo de reação do sistema .....</b>	<b>60</b>
<b>4.2.8</b>	<b>Formulação do tempo de rotação desejado.....</b>	<b>60</b>
<b>4.3</b>	<b>Formulação analítica do desempenho .....</b>	<b>60</b>

4.3.1	Estimativa para $N$ mestres.....	61
4.3.2	Eficiência da rede PROFIBUS .....	64
4.3.3	Estimativa para uma rede mestre-escravo puro.....	66
<b>5</b>	<b>Procedimento para medida de tempo em uma rede Profibus.....</b>	<b>69</b>
5.1	Primeiro grupo experimental.....	70
5.2	Segundo grupo experimental.....	71
5.3	Terceiro grupo experimental.....	72
5.4	Análise dos dados obtidos na monitoração de uma rede PROFIBUS.....	72
5.4.1	Análise dos experimentos com um mestre e um escravo .....	73
5.4.2	Análise dos experimentos com um mestre e três escravos.....	76
5.5	Arquitetura de um escravo PROFIBUS.....	79
5.6	Modelo analítico para uma rede mestre escravo puro.....	80
5.7	Modelo analítico para uma rede com 1 mestre e $n$ escravos.....	81
5.8	Modelo analítico para uma rede com $n$ mestre e $n$ escravos.....	83
5.9	Variáveis Aleatórias.....	83
5.10	Ferramenta para resolução do modelo analítico proposto.....	85
5.11	Estudo de caso.....	87
<b>6</b>	<b>Conclusões.....</b>	<b>88</b>
	<b>Anexo 1 Foundation Fieldbus.....</b>	<b>90</b>
	<b>Bibliografia.....</b>	<b>98</b>

## Lista de Abreviaturas

BP	Bit Paridade
CD	Compel Data
CI	Circuitos Integrados
CIM	Computer Integrated Manufacturing
CLP	Controladores Lógicos Programáveis
CNC	Comando Numérico Computadorizado
CNR	Comando de Robôs
DAE	Destination Address Extension
DAE	Destination Address Extension
DAP	Destination Service Access
DDL	Data Link Layer
DIN	Deutsche Industry Norm
ECL	Estações de Controle Local
FAS	Fieldbus Access Sublayer
FB	Function Block
FDL	Fieldbus Data Link
FMA	Fieldbus Management
FMA1/2	Fieldbus Management Layer 1/2
FMA7	Fieldbus Management Layer 7
FMS	Fieldbus Message Specification
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IHM	Interface Homem/Máquina
ISA	International Society of Measurement and Control
ISO	International Standards Organization
ISP	Interoperable System
LAS	Link Active Scheduler
LLI	Lower Layer Interface
LSAP	Link Service Access Point
LSB	Least Significant Bit
MAP	Manufacturing Automation Protocol
MAP-EPA	Manufacturing Automation Protocol Enhanced Performance Architecture
MMS	Manufacturing Message Specification
MSB	Most Significant Bit
NA	Node Activation
NMIB	Network Management Information Base
NS	Next Station
NZR	No Return to Zero
OD	Object Dictionary
PHY	Physical
PN	Probe Node
PR	Probe Response
PS	Previous Station
PT	Pass Token
RM-OSI	Reference Model for Open Systems Interconnection
SAE	Source Address Extension



SAP	Service Access Point
SB	Start Bit
SMIB	System Management Information Base
SSAP	Source Service Access Point
ST	Stop Bit
TC	Technical Committee
TD	Time Distribution
TS	This Station
UART	Unidade de Palavra de Informações para Transmissão Serial Assíncrona
VFD	Virtual Field Device

## Lista de Símbolos

/	Divisão
$\lambda$	Lambda
$\mu$	Mi
*	Multiplicação
=	Procedência igual
$\geq$	Procedência maior ou igual
$<$	Procedência menor
$\leq$	Procedência menor ou igual
$\rho$	Ro
$\sigma$	Sigma
+	Soma
-	Subtração
$\tau$	Tau

## Lista de Figuras

FIGURA 2.1 – Filosofia CIM (Computer Integrated Manufacturing).....	19
FIGURA 2.2 – Modelo de referência RM-OSI.....	21
FIGURA 2.3 – Exemplo de transmissão de dados no modelo OSI.....	22
FIGURA 3.1 - Sistema supervisório .....	26
FIGURA 3.2 – Modelo de comunicação reduzido no FIELDBUS.....	28
FIGURA 3.3 – Arquitetura de camadas PROFIBUS. ....	31
FIGURA 3.4 – Estrutura de um caracter UART .....	32
FIGURA 3.5 – Redundância no barramento .....	33
FIGURA 3.6 – Modelo híbrido de acesso ao meio de transmissão.....	34
FIGURA 3.7 – Estrutura dos quadros no PROFIBUS .....	37
FIGURA 3.8 – Seqüência de um serviço cíclico.....	40
FIGURA 3.9 – Diagrama de estados da FDL.....	47
FIGURA 4.1 – Tempo do ciclo de Token .....	58
FIGURA 4.2 – Tempo de ciclo de mensagens .....	59
FIGURA 4.3 – Tempos dos ciclos de mensagens em uma rede PROFIBUS .....	62
FIGURA 4.4 – Tempo de rotação desejado com carga máxima na rede .....	66
FIGURA 4.5 – Ciclo de mensagens em uma rede mestre-escravo puro .....	67
FIGURA 4.6 – Ciclo de mensagens em uma rede mestre-escravo .....	67
FIGURA 5.1 – Primeiro modelo experimental, 1 mestre e um 1 escravo.....	71
FIGURA 5.2 – Segundo modelo experimental, 1 mestre e 3 escravos .....	71
FIGURA 5.3 – Terceiro modelo experimental, 2 mestres e 2 escravos .....	72
FIGURA 5.4 – Arquitetura de um escravo Profibus .....	80
FIGURA 5.5 – Gráfico do modelo analítico para uma rede mestre escravo puro .....	81
FIGURA 5.6 – Gráfico do modelo analítico para uma rede 1 mestre e $n$ escravo.....	83
FIGURA 5.7 – Probabilidade associadas à distribuição normal .....	84
FIGURA 5.8 – Simulador de análise de desempenho de redes PROFIBUS.....	85
FIGURA 5.9 – Tela de apresentação dos resultados obtidos na simulação .....	86
FIGURA 5.10 – Tempos configuráveis da aplicação.....	86

## Lista de Tabelas

TABELA 1.1 – Descrição de alguns sistemas Fieldbus.....	16
TABELA 3.1 – Características da topologia do protocolo PROFIBUS.....	32
TABELA 3.2 – Estrutura da pool list.....	44
TABELA 3.3 – Serviços executados pelo FMA 1/2 .....	48
TABELA 3.4 – Características dos padrões Foundation Fieldbus e Profibus. ....	53
TABELA 4.1 – Tempos configuráveis versus taxas de transmissão.....	61
TABELA 4.2 – Tempos dos ciclos de mensagens $T_{MC}$ .....	61
TABELA 4.3 – Tempos do $T_{TR}$ com o quadro de pedido de 5 octetos.....	63
TABELA 4.4 – Tempos do $T_{TR}$ com o quadro de pedido de 100 octetos.....	63
TABELA 4.5 – Tempos do $T_{TR}$ com o quadro de pedido de 150 octetos.....	63
TABELA 4.6 – Eficiência da rede.....	64
TABELA 4.7 – Eficiência da rede com variação dos tamanhos de quadros.....	64
TABELA 4.8 – Eficiência da rede com variação dos tamanhos de quadros.....	65
TABELA 4.9 – Tempos do $T_{TR}$ com o carga máxima na rede 246 octetos de dados ....	65
TABELA 4.10 – $T_{MC}$ para uma rede mestre-escravo puro.....	66
TABELA 4.11 – $T_{SR}$ para uma rede mestre-escravo.....	68
TABELA 5.1 – Parâmetros usados nos experimentos .....	69
TABELA 5.2 – Características dos experimentos.....	70
TABELA 5.3 – Descrição dos dados do experimento.....	73
TABELA 5.4 – Experimento 1, pedido 10, resposta 12 octetos, 187,5 Kbits/s.....	73
TABELA 5.5 – Experimento 2, pedido 96, resposta 12 octetos, 187,5 Kbits/s.....	74
TABELA 5.6 – Experimento 5, pedido e resposta 120 octetos, 187,5 Kbits/s .....	74
TABELA 5.7 – Experimento 6, pedido e resposta 120 octetos, 500 Kbits/s .....	75
TABELA 5.8 – Experimento 7, pedido e resposta 120 octetos, 1,5 Mbits/s.....	75
TABELA 5.9 – Experimento 9, pedido e resposta 120 octetos, 6 Mbits/s.....	75
TABELA 5.10 – Experimento 10, pedido e resposta 120 octetos, 12 Mbits/s.....	75
TABELA 5.11 – Experimento 11 parte 1, 187,5 Kbits/s .....	76
TABELA 5.12 – Experimento 11 parte 2, 187,5 Kbits/s .....	77
TABELA 5.13 – Experimento 12 parte 1, 500 Kbits/s .....	77
TABELA 5.14 – Experimento 12 parte 2, 500 Kbits/s .....	78
TABELA 5.15 – Experimento 21 parte 1, 500 Kbits/s .....	78
TABELA 5.16 – Experimento 21 parte 2, 500 Kbits/s .....	78
TABELA 5.17 – Dados medidos versus estimados em uma rede mestre-escravo puro	80
TABELA 5.18 – Dados medidos versus estimados com 1 mestres 3 escravos.....	82

## Resumo

Durante a evolução da comunicação, diversos fatores contribuíram para melhorar a qualidade dos sistemas computacionais (Hardware e Software) em um ambiente industrial, onde o fator desempenho é um dos pontos importantes a ser considerado em um projeto de automação industrial.

Nos últimos vinte anos, surgiram diversas propostas de padronização de protocolos para automação industrial e aplicações de tempo real, cada qual com características e tempos de resposta diferentes.

Um estudo detalhado sobre as características técnicas de controle de tempo é de extrema importância, visto que esses sistemas, muitas vezes são implementados em situações de missão crítica, nas quais o tempo de resposta é um fator crucial para a aplicação.

Nesta dissertação, apresentamos as etapas de evolução das técnicas de controle de acesso ao barramento, culminando inicialmente com o surgimento da proposta do protocolo FIELDBUS e em seguida com a proposta alemã, o PROFIBUS.

O principal objetivo deste trabalho é propor um modelo analítico para a realização de avaliação de desempenho em uma rede PROFIBUS.

O trabalho apresenta uma descrição das características funcionais do protocolo, bem como uma descrição detalhada dos tempos de controle utilizado pelo protocolo para controle e supervisão da rede.

Uma descrição dos experimentos e o cruzamento dos dados obtidos na simulação com os dados obtidos através do modelo analítico demonstram a razão pela qual os modelos analíticos é uma ferramenta de extrema importância devido a sua agilidade em estimar o desempenho de uma rede PROFIBUS.

**Palavras-chave:** Avaliação de desempenho, automação industrial, redes de campo, protocolos.

**Title:** "ANALYZE AND ESTIMATE OF PERFORMANCE OF PROFIBUS NETWORK"

## **Abstract**

As communications evolved, several factors contributed to improving the quality of computational systems (hardware and software) in an industrial environment where the factor performance is one of the important points to be taken into account in an industrial automation project.

Several proposals have been made in the last twenty years to standardize protocols for industrial automation and real time applications, each of them with different features and response times.

It is very important to have a detailed study of the technical characteristics of time control, since these systems are often implemented in critical mission situations, in which response time is a crucial factor for application.

In this dissertation, we present the stages of evolution for techniques to control bus access, the high point being the proposal of the FIELDBUS protocol and also PROFIBUS, the German proposal.

The main purpose of this study is to propose an analytic model to evaluate the performance of a PROFIBUS network. A description of the functional features of the protocol is also available, as well as a detailed description of the control times used by the protocol to control and supervise the network.

A description of the experiments is available in a chapter, as well as the cross-matching the data obtained in simulation to the data obtained from the analytic model, showing why the analytic models are a highly important tool, due to their flexibility to estimate the performance of a PROFIBUS network.

**Keywords:** Performance evaluation, Industrial automation, Fieldbus protocol, Protocol.

# 1 Introdução

No atual cenário mundial, diversas empresas têm passado por modificações em seus processos de fabricação com o objetivo de alcançar não só menores custos, mas também melhorias na qualidade de seus processos e produtos.

Com o crescente aumento da complexidade das aplicações de controle industrial e a integração cada vez maior entre controladores, sensores e atuadores, as redes locais de comunicação assumem uma posição de vital importância, uma vez que tem como objetivo o compartilhamento e a interconexão de recursos computacionais, tais como hardware e software.

Durante os últimos 25 anos, o grande desenvolvimento da microeletrônica possibilitou o aparecimento de circuitos integrados (CIs) de altíssima escala de integração. A aplicação desses CIs na indústria de instrumentação acarretou, por sua vez, o aparecimento de instrumentos de campo dotados de capacidade até então delegadas somente aos sistemas hierarquicamente superiores tais como CLP's, CNC's, etc.

Esse avanço proporcionou o surgimento de numerosas aplicações que integram armazenamento e processamento de dados, revolucionando toda a tecnologia de controle.

Porém, essa nova geração de instrumentos passa a tratar 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 apresentar limitações de velocidade, ruído e conversão AD/DA .

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.

Procurando uma melhor qualidade de comunicação, era inevitável a substituição dos protocolos de comunicação analógicos por um protocolo digital para aplicações em ambiente fabril.

Nesse cenário tecnológico surgiram as primeiras propostas para comunicação digital capazes de interligar tais instrumentos de campo. Contudo são soluções proprietárias específicas de cada fabricante sem a preocupação com uma linguagem universal normalizada que permita a integração dos diversos dispositivos de diferentes fornecedores.

Neste contexto em meados dos anos 80 surgiu o conceito FIELDBUS. Essa proposta tinha algumas características, entre elas que o novo padrão fosse projetado especificamente para aplicações de chão de fábrica e que tivesse mecanismos que garantissem tempo de resposta para aplicações críticas.

Seu conceito e aceitação são baseados em um trabalho feito por inúmeros grupos nos Estados Unidos e na Europa, dentre eles temos a ISA (International Society for Measurement and Control), IEC (International Electrotechnical Commission), DIN (Deutsche Industry Norm), entre outros.

Tais grupos desenvolveram um estudo sobre os principais requisitos que o novo padrão deveria preencher. Hoje no mercado encontramos uma série de protocolos que têm no seu âmbito a filosofia FIELDBUS.

A seguir descreve-se alguns protocolos e suas características em relação à taxa de transmissão, número de estações e comprimento máximo do barramento.

Todos os protocolos abaixo já foram ou estão sendo normalizados por grupos internacionais como o ISA, IEC e DIN entre outros.

TABELA 1.1 – Descrição de alguns sistemas Fieldbus

<b>Protocolo</b>	<b>Taxa de Transmissão Mbits/s</b>	<b>Número máximo de dispositivos no barramento</b>	<b>Comprimento Máximo do barramento Km</b>	<b>Tipo de barramento</b>
Foundation FIELDBUS	0,03~100	32	0,5~1,9	CO
Interbus-S	0,5	256	12,8	CO, IO
Interbus-SL	0,5	64	0,1	IO
Profibus FMS	0,09~12	31x4	23,8	P, IO
Profibus DP	0,09~12	31x4	23,8	CO,IO
Can	1	110	1	CO, IO
WorldFIP	2,5	32x5	5	P,CO
BitBus	0,375	250x4	13,2	IO,IO
LON works	1,25	2x128	*	IO

P = Processo CO= Bus nível de controle IO = Bus nível de IO

Pode-se observar que cada protocolo oferece características diferentes como número de estações no barramento, comprimento máximo do barramento e taxas de transmissão. Sendo assim, tem-se uma série de produtos, cada qual, apresentando características diferentes entre si.

### 1.1 Objetivos da Pesquisa

O presente trabalho tem por objetivo estudar os mecanismos de controle e transmissão de dados e estimar o desempenho de um protocolo que tem características de tempo real. Dentro dos diversos protocolos existentes no mercado, o PROFIBUS foi escolhido devido à facilidade de se implementar uma rede e obter as médias de tempo em um estudo de caso.

Temos duas técnicas possíveis para estimar o desempenho de uma rede sem a necessidade de implementação.

A primeira técnica é através de simulações, onde as características e o comportamento do protocolo é inserido em um simulador como o *Network Simulation 2*. Segundo [GIO 86], a primeira dificuldade em construir um modelo para um sistema reside na escolha criteriosa de suas características “essenciais”. Uma outra dificuldade é a própria parametrização do simulador, porque caso não haja uma implementação pronta, é demorado e oneroso.



A segunda alternativa é o uso de modelos matemáticos capazes de serem avaliados analiticamente (cálculos), que é uma forma simples de realizar a avaliação de desempenho da rede.

Entendemos que as ferramentas analíticas são limitadas em relação à complexidade de um sistema que está sendo modelado. Mas, por outro lado, justifica-se utilizar essa ferramenta pela sua simplicidade e rapidez em obter os resultados, informando apenas algumas características da rede.

Uma contribuição significativa que se pode oferecer seria determinar um mecanismo que estimasse o tempo de resposta da rede para um sistema PROFIBUS de uma forma simples e eficiente. Dessa forma, pode-se observar o comportamento da rede antes mesmo de implementá-la fisicamente, verificando-se os requisitos de tempo de resposta se estão sendo atendidos para uma determinada aplicação.

Dessa forma poderemos avaliar o quanto essas ferramentas nos dão resultados próximos da realidade.

## **1.2 Motivação**

Vários trabalhos estão disponíveis utilizando ferramentas analíticas, mas poucos trabalhos tratam de avaliação de desempenho utilizando tais ferramentas para protocolos de tempo real.

Dessa forma, provar ou não que essas ferramentas são viáveis para determinar o desempenho de um protocolo que tem requisitos de tempo real é extremamente importante, porque tais ferramentas tem uma aplicabilidade extremamente rápida e de certa forma simples.

O enfoque principal da pesquisa é propor um modelo analítico que, com um certo grau de confiabilidade, responda a questões de desempenho sem que se tenha a necessidade de implementar uma rede para responder a essas questões.

Sendo assim, esta pesquisa contribui de forma significativa propondo um modelo analítico para avaliação de desempenho para sistemas de tempo real.

## **1.3 Estrutura do texto**

Este texto está dividido em duas partes. A primeira parte, composta pelos capítulos 2, 3 e 4, descreve de uma forma condensada a evolução da comunicação digital e as principais características de dois protocolos projetados para aplicações em tempo real. No segundo momento, apresentar-se-á um modelo simplificado da temporização envolvida no protocolo PROFIBUS. A segunda parte, composta pelo capítulo 5, descreve como foi realizado o experimento e apresenta os resultados de suposições utilizando algumas formulas dispostas nas normas. Na segunda parte desse capítulo, apresentamos o resultado das simulações e propomos os modelos analíticos para a avaliação de desempenho do PROFIBUS

O capítulo 2 aborda o primeiro padrão internacional para comunicação digital, o modelo de referência RM-OSI. São abordados de forma superficial os aspectos funcionais desse padrão. Também são inseridos nesse capítulo as primeiras tentativas de se adotar esse padrão para redes de automação e controle dando uma idéia das dificuldades encontradas em tentar-se criar um padrão único para ambientes fabris.

O capítulo 3 descreve a normalização de padrões internacionais para redes de automação e controle. O objetivo dessa normalização é um protocolo para um ambiente fabril, que atenda aos anseios do mercado em relação a um protocolo específico para automação industrial. O padrão FIELDBUS é hoje base para uma série de outros protocolos que se encontram no mercado. O padrão Foundation FIELDBUS é o que se mais aproxima da idéia inicial da padronização FIELDBUS e o PROFIBUS é a proposta que inicialmente foi proposto pela Alemanha de padronização de uma rede FIELDBUS.

O capítulo 4 descreve os tempos de supervisão e controle do PROFIBUS. Dentro desse capítulo também encontraremos uma análise de desempenho da rede onde os dados foram extraídos usando os tempos de supervisão do PROFIBUS e supondo alguns tamanhos de quadros de dados. Algumas curvas foram traçadas levando em consideração taxa de transmissão, número de mestres e escravos na rede.

No capítulo 5 é demonstrada a forma com que os experimentos foram realizados e apresentamos os dados do experimento que serão comparados com os dados obtidos através de uso de ferramentas analíticas. São gerados gráficos e tabelas para facilitar a compreensão dos dados obtidos.

## 2 Integração dos meios de comunicação

As redes locais têm tido uma maior demanda nas indústrias, mas isso deve-se a vários fatores, entre eles a filosofia CIM (Computer Integrated Manufacturing).

Integrar uma manufatura por computador leva-nos a conectar todos os sistemas computadorizados existentes na empresa tais como: controladores lógicos programáveis (CLPs), dispositivos de comando numérico computadorizado (CNCs), dispositivos de comando de robôs (CNRs) etc., além da área administrativa e de projetos, como mostra a figura 2.1.

Os problemas com a integração ocorrem principalmente em relação à interligação de equipamentos de diferentes fabricantes. Isso torna essencial o desenvolvimento de projetos que estabelecessem um padrão único de troca de mensagens num ambiente fabril.

O uso de padrões internacionais de comunicação torna-se a solução mais adequada a ser adotada pelos fabricantes quanto à integração computacionais e de equipamentos de tecnologias diferentes.

Tais problemas começam a ser resolvidos pela ISO (International Standards Organization), formada por organizações nacionais de padronização de 89 países.

Em 1977 o comitê técnico da ISO, que lida com padronização de computadores e processamento de informação através do TC-97 (Technical Committee-97), criou um subcomitê chamado SC-16 com o objetivo de estudar necessidades de padronização na área de interconexão de sistemas de computação conhecido como TC-97/SC-16 da ISO.

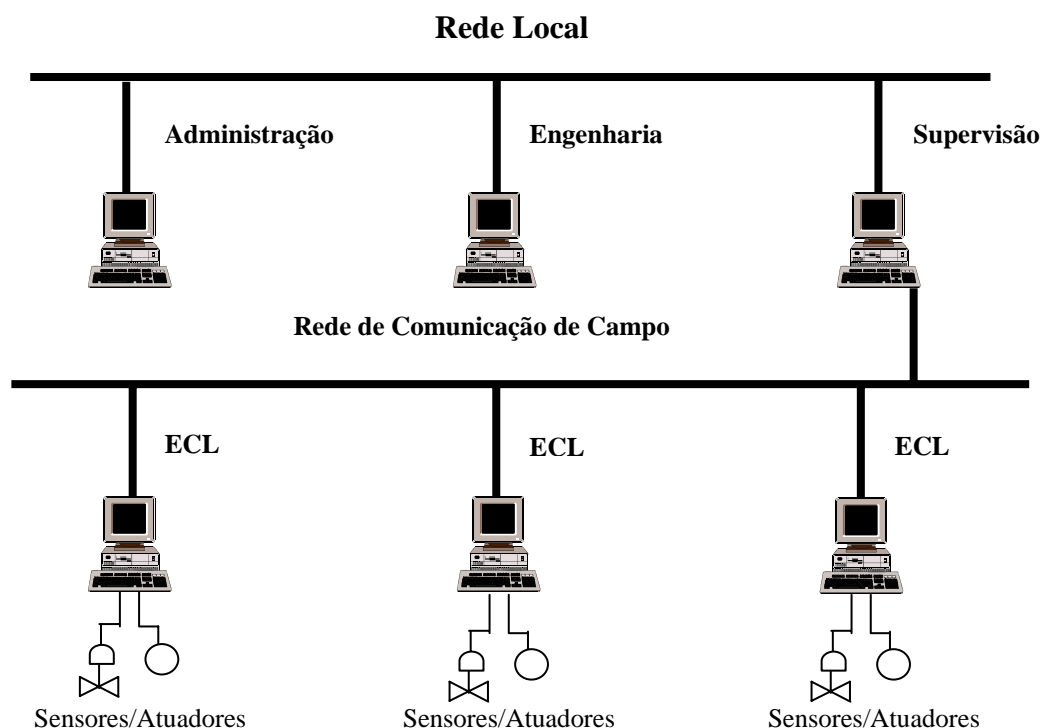


FIGURA 2.1 – Filosofia CIM (Computer Integrated Manufacturing)

Em 1979 o Modelo de Referência para Arquitetura de Sistemas Abertos está pronto e se torna conhecido sob o nome de RM-OSI (Reference Model for Open Systems Interconnection).

Em 1983 a ISO o aprova como padrão Internacional para interconexão de sistemas abertos através do documento ISO-7498.

O modelo (RM-OSI/ISO) hoje define em linhas gerais a comunicação digital, sendo aceito pela indústria de 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.

## **2.1 O Modelo de referência RM-OSI/ISO**

O objetivo do padrão internacional 7498, RM-OSI, é fornecer uma base comum que permita o desenvolvimento coordenado de padrões para interconexão de sistemas. A denominação OSI qualifica padrões para o intercâmbio de informações entre sistemas.

Para a ISO, o fato da interconexão ser aberta não implica o uso de nenhuma implementação, tecnologia, ou modo de interconexão específico, mas refere-se ao reconhecimento e suporte dos padrões ISO para intercâmbio de dados. [LEM 95]

Segundo [LEM 95], é também finalidade do RM-OSI identificar áreas para a elaboração ou aperfeiçoamento de padrões e fornecer uma referência comum para a manutenção da consistência entre os padrões de comunicação entre computadores.

Devemos ficar atentos ao fato de que o RM-OSI, por si só, não define a arquitetura de uma rede porque permite várias implementações diferentes de serviços e protocolos para as várias camadas existentes, dessa forma dois sistemas em conformidade com o RM-OSI podem não se comunicar devido às opções de implementação de serviços e protocolos serem incompatíveis.

Sendo assim, a ISO elaborou o conceito de perfis funcionais, pois para que dois sistemas quaisquer possam trocar informação, é necessário que os dois tenham o mesmo perfil funcional, dessa forma as opções de serviço e protocolo serão compatíveis. Sendo adotado o mesmo perfil, garantidamente os dois sistemas irão se comunicar.

A estrutura de comunicação funciona usando o conceito de camadas. O modelo RM-OSI está dividido em 7 camadas como mostra a figura 2.2. Cada camada fornece serviços de comunicação.

Um serviço representa um conjunto de funções oferecidas a um usuário por um fornecedor. O serviço oferecido por um fornecedor é acessado por um usuário através de um ponto de acesso ao serviço SAP (Service Access Point). [LEM 95]

Os serviços fornecidos pelas camadas inferiores são sucessivamente ampliados e aperfeiçoados, de modo que na camada superior (camada 7) seja oferecido um conjunto de serviços adequado para suportar as diversas aplicações. [GIO 86]

Formalmente, o serviço fornecido por uma camada a outra é especificado pelo conjunto de primitivas de serviço. Um serviço pode ser : confirmado, não-confirmado ou iniciado pelo fornecedor.

No modelo OSI, o serviço fornecido por uma camada pode ser orientado à conexão ou não orientado à conexão.

No modo orientado à conexão, o serviço é dividido em três fases de operação : estabelecimento da conexão, transferência de dados e liberação da conexão.

Já no modo não orientado à conexão, uma única unidade de dados é transmitida do SAP de origem para um ou mais SAPs de destino, sem que para isso seja estabelecida uma conexão entre eles. [LEM 95]

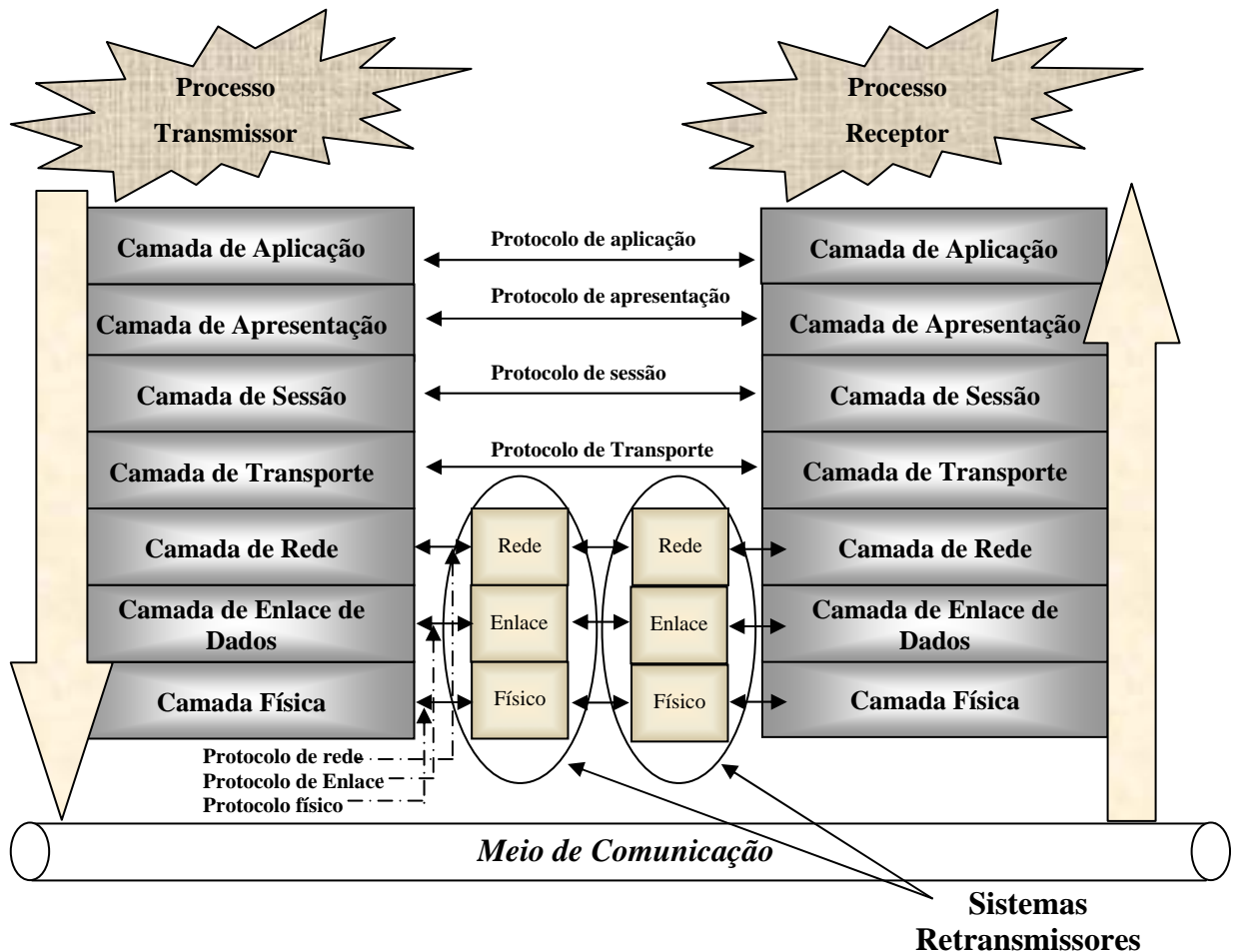


FIGURA 2.2 – Modelo de referência RM-OSI

O modelo OSI possui sete níveis : aplicação, apresentação, sessão, transporte, rede, enlace e físico, que são mostrados na figura 2.2.

### 2.1.1 Transmissão de dados no modelo RM-OSI

A comunicação dá-se entre camadas de mesmo nível hierárquico, ou seja, a camada n de uma máquina ‘conversa’ com a camada n de outra máquina, sendo que as regras para essa comunicação são conhecidas como o protocolo da camada n. [TAN 97]

A figura 2.3 mostra como os dados podem ser transmitidos no modelo OSI. Eles passam através das diversas camadas até chegar ao meio físico de transmissão.

Em cada camada é atribuído um cabeçalho ao dado; após a inserção desse cabeçalho, o mesmo é transmitido para a camada inferior.

O processo ocorre sucessivamente até que se chegue à camada física, de onde serão transmitidos para a máquina receptora.

Na máquina receptora ocorre o processo inverso, os cabeçalhos são excluídos um a um à medida que a mensagem propaga-se pelas camadas até chegar ao processo de recebimento. [TAN 97]

Deve ser salientado que, na realidade, nenhum dado é passado diretamente da camada n de uma máquina para a camada n da outra máquina. O que ocorre é a passagem de dados e informações de controle à camada imediatamente inferior ou superior e assim sucessivamente como foi exemplificado acima.

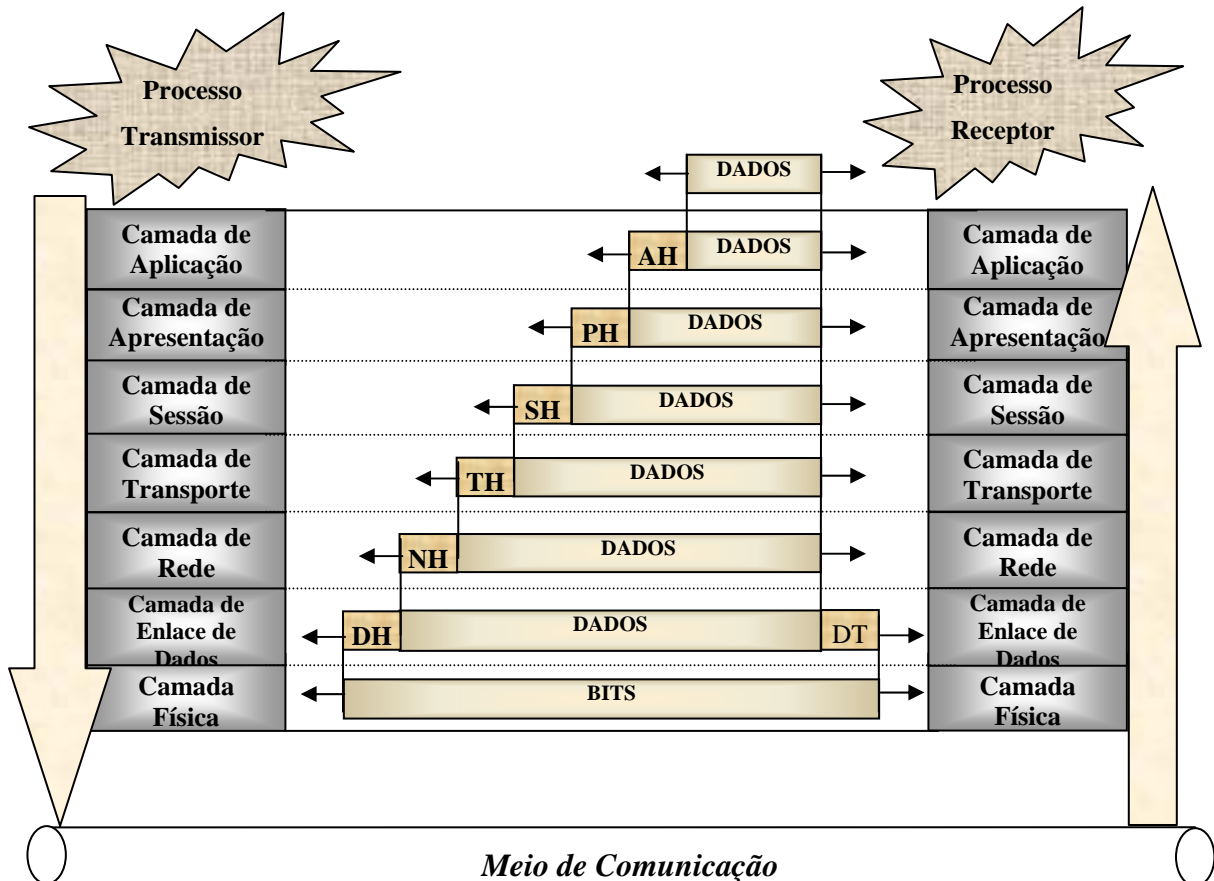


FIGURA 2.3 – Exemplo de transmissão de dados no modelo OSI

Entre cada uma das camadas adjacentes existe uma interface, a qual define primitivas de operações entre ela, bem como os serviços que a camada inferior deve oferecer à superior.

Assim, é possível modificar toda uma implementação da camada sem prejudicar o sistema, pois sabe-se de antemão quais são todos os requisitos que a nova implementação deve satisfazer para oferecer, à sua camada superior, exatamente os mesmos serviços que a antiga implementação oferecia.

## 2.2 MAP (Manufacturing Automation Protocol)

O MAP foi o primeiro padrão baseado no modelo de referência RM-OSI largamente aceito para redes de automação e controle.

O objetivo principal no projeto MAP era desenvolver um protocolo que suportasse a transmissão simultaneamente de TV, áudio e dados com o princípio que qualquer

computador, independente de sua marca ou fabricante, poderia ser conectado ao sistema. Um outro aspecto importante era a automação industrial.

Foi adotado para a camada de enlace o padrão da especificação IEEE 802.4 “token bus”. Esse padrão foi escolhido devido as suas características determinísticas quanto ao acesso ao meio de comunicação, fator crucial para aplicações de tempo real.

A sua principal característica é a garantia de tempo de resposta dentro de um certo intervalo.

O mecanismo que realiza tal função consiste em um anel lógico, no qual as estações alteram-se na utilização do meio de comunicação, dispondo cada uma de um intervalo de tempo proporcional ao intervalo de tempo requerido pela aplicação, ao tempo gasto na transmissão e ao número de estações presentes no anel.[SIL 96]

A garantia de tempo é relacionada à situação de pior caso do sistema. Se existem  $n$  estações atuantes no anel e o tempo gasto para transmitir uma mensagem é  $T$  segundos, então nenhuma mensagem precisará esperar mais de  $nT$  segundos para ser transmitida. O tempo  $nT$  segundos é calculado, então, de modo que a transmissão ocorra dentro do intervalo exigido pela aplicação. [SIL 96]

Um outro fator importante dentro do protocolo Token Bus é a possibilidade de determinar prioridades para as mensagens. Esse fator diferencia e muito em relação ao padrão Ethernet.

Na camada de aplicação o protocolo MAP adota o protocolo MMS (Manufacturing Message Specification). Esse protocolo é padronizado pela ISO, mas não faz parte da arquitetura OSI.

A MMS é uma SASE (Specific Application Layer Service Element) ou elemento específico de serviço da camada de aplicação, terminologia da ISO para blocos de programa que implementam alguns serviços específicos, para determinados tipos de comunicação padronizadas para o controle e monitoração de equipamentos industriais e específica a semântica da comunicação em automação industrial. [DAM 88]

Dentro do projeto MAP foram definidos três tipos de arquiteturas.

A primeira arquitetura é o MAP comum, essa arquitetura consiste em um sistema aberto RM-OSI de sete camadas e altamente compatível com os sistemas de transmissão de dados, áudio e vídeo.

Essa arquitetura só apresentava um problema : devido à implementação das sete camadas, os tempos de resposta não eram satisfatórios para aplicações de tempo real.

Criou-se então, a segunda arquitetura MAP-EPA (Manufacturing Automation Protocol-Enhanced Performance Architecture) que consiste de um nó MAP com melhor desempenho em termos de tempo de resposta.

Essa melhoria é conseguida através da inserção, junto aos serviços OSI, de protocolos de resposta rápida que eliminam algumas das camadas superiores do modelo OSI, contudo, resultou em um protocolo de maior complexidade e, conseqüentemente, em um alto custo.

Um terceiro tipo, denominado Mini-MAP, foi desenvolvido contendo apenas três camadas (física, enlace e de aplicação) e as características de alto desempenho. Sendo assim o Mini-MAP era incompatível com o MAP, sendo necessária a utilização de gateways (pontes), ou seja, um nó EPA-MAP para realizar a conexão com as redes MAP.

Devido às limitações do protocolo MAP como tempo de resposta relativamente elevado para aplicação de controle de processo ou no caso do MAP-EAP perda de garantia da qualidade das mensagens, 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 havia a necessidade de se criar uma normalização que atendesse às necessidades de tempo real.

Nesse âmbito surgiu o conceito FIELDBUS que define as características necessárias para o desenvolvimento de protocolos para ambientes fabris. No capítulo seguinte, serão abordadas as características principais dessa normalização.



### 3 Padrões de redes de campo

Alguns órgãos normativos tais como o IEC (International Electrotechnical Commission), IEEE (Institute of Electrical and Electronics Engineers), ISA (International Society of Measurement and Control) e DIN (Deutsche Industry Norm) iniciaram grupos de estudo sobre o FIELDBUS e lançaram os requisitos que o novo padrão deveria preencher.

Dentre as várias propostas de padronização resultantes se destacam o PROFIBUS (Alemanha), o FIP (França), e a ISA SP-50 (EUA).

Com a união entre os grupos ISA e o IEC em 1985, começava-se a desenvolver um padrão para comunicações digitais, bi-direcionais, multiponto entre dispositivos de campo (instrumentos) e sistemas de controle.

O objetivo dessa normalização é gerar um único padrão internacional para o FIELDBUS.

Como o processo de normalização do IEC levou muito tempo, um grupo de fabricantes teve a iniciativa de unir-se e começar a usar as atuais especificações disponíveis do padrão IEC, assumindo que seriam definidas as partes faltantes desse padrão num futuro próximo.

Foi constituída uma organização chamada Foundation FIELDBUS que reunia todos os principais fabricantes ao redor do mundo com o objetivo de construir uma base de implementação e apoio ao IEC para desenvolver os equipamentos conforme o padrão FIELDBUS.

Como os trabalhos de normalização ainda não estavam concluídos, o padrão Foundation FIELDBUS viu-se obrigado a especificar partes que estavam faltando na especificação da norma internacional, pois ela ainda não estava completa.

No entanto, foi feito um acordo entre os dois grupos, para que o que fosse desenvolvido pela Foundation FIELDBUS fosse acrescentado ao IEC, com o compromisso de um grupo apoiar o outro.

Os demais órgãos normativos lançaram as suas versões do protocolo FIELDBUS, nesse âmbito, o cenário mundial ficou composto por diversos protocolos com a filosofia FIELDBUS, mas cada qual com uma implementação específica e proprietária.

Dado esse enfoque inicial, faremos uma abordagem em dois protocolos normalizados pelos órgãos internacionais o Foundation FIELDBUS e o PROFIBUS.

Em relação ao primeiro protocolo teremos uma abordagem superficial. Caso o leitor queira uma descrição mais detalhada sobre a normalização desse protocolo poderá consultar o anexo 1. Em relação ao PROFIBUS tem-se uma abordagem mais detalhada por se tratar do protocolo em estudo.

#### 3.1 Foundation FIELDBUS

Um dos principais fatores do projeto FIELDBUS é que o protocolo foi projetado diretamente para aplicação de chão de fábrica.

O FIELDBUS é um sistema de comunicação digital bidirecional que permite a interligação em rede de múltiplos instrumentos diretamente no campo realizando

funções de controle e monitoração de processo e estações de operação IHM (Interface Homem Máquina) através de software supervisório como mostra Figura 3.1.

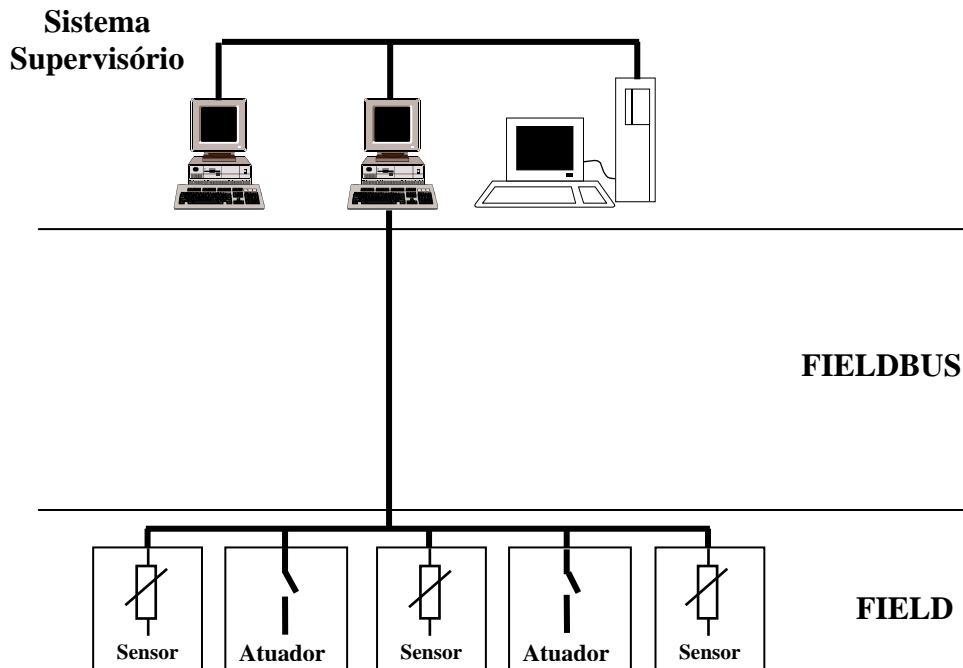


FIGURA 3.1 - Sistema supervisório

Ele foi concebido de modo a atender plenamente a todos os itens de desejos dos usuários, já que até então nenhum outro protocolo havia sido implementado para atender a essas exigências, dentre as quais podemos enumerar:

- a) estar de acordo com o modelo ISO/OSI.
- b) uso de cabos de conexão de utilização industrial normal.
- c) segurança intrínseca para atmosferas perigosas.
- d) variáveis identificadas por “tags”, ou seja, o nome da variável definida pelo usuário e expressas em unidades de engenharia.
- e) variáveis com “status”, que indica as condições da variável.
- f) blocos de função, com parâmetros de entrada e saída, parâmetros de configuração e algoritmos padronizados. Exemplos : tabelas de falhas, dados de diagnostico, etc.

O FIELDBUS surgiu com o objetivo de interligar e operar os instrumentos de campo com características diferentes e de diversos fabricantes. Desfrutando de toda sua “inteligência” através de uma rede, proporcionando a descentralização das tarefas que é muitas vezes vista como uma possibilidade de espalhar, entre vários dispositivos, um determinado programa ou processo de controle na busca de melhor uso de suas características.

A base para arquitetura de um equipamento FIELDBUS são os blocos funcionais, os quais executam as tarefas necessárias às aplicações existentes hoje, tais como : aquisição de dados, controle PID, cálculos e atuação.

Todo bloco funcional contém um algoritmo, uma base de dados (entradas e saídas) e um nome definido pelo usuário (o “tag” do bloco, deve ser único na planta do usuário).

Os parâmetros do bloco funcional são endereçados no FIELDBUS via TAG.PARAMETER-NAME. [SMA 98]

Essa interligação incorpora vantagens como: maior imunidade a ruídos, pré-processamento de dados específicos, transmissão de informações adicionais dos dados, capacitando o diagnóstico do dispositivo e a previsão de falhas, redução dos custos de projeto, de fiação, de instalação e de expansão, entre outras.

Para alcançar uma maior confiabilidade foi prevista a capacidade de, em caso de pane de um dispositivo, a substituição imediata por outro implementando a mesma funcionalidade.

Como os dispositivos podem ser diferentes e de distintos fabricantes, a padronização das funções a serem distribuídas nos mesmos foi necessária.

Essas funções são chamadas de blocos funcionais ou FB (Function Blocks). A interligação desses blocos funcionais é que define a estratégia de controle e programação do processo a ser controlado. Na configuração especifica-se a escolha do FB e em que dispositivo será executado.

### **3.1.1 Características funcionais**

O barramento de campo FIELDBUS é constituído por um sistema de comunicação de dados totalmente digital, bidirecional (half-duplex), multiponto, associado a protocolos de controles de enlace de dados e de interface como o usuário, capaz de interconectar equipamentos como sensores, atuadores e controladores e com especificações quanto aos requisitos de segurança intrínseca.

O protocolo FIELDBUS foi desenvolvido baseado no padrão RM-OSI. Embora não tenha todos os seus níveis, podemos, em primeira análise, dividi-lo em nível físico ("Physical Layer" - que trata das técnicas de interligação dos instrumentos) e níveis de software ("Communication Stack") ou Pilha de Comunicação que tratam da comunicação digital entre os equipamentos como mostra a figura 3.2.

### **3.1.2 Nível físico**

Nas especificações do nível físico estaremos analisando os tipos de ligações possíveis e as características descritas na norma.

Algumas das principais características tratadas no nível físico normalizam a transmissão de dados somente digitais e comunicação bidirecional. Permitindo uma velocidade de transmissão de 31,25 Kbit/s a 100 Mbit/s.

A conversão das mensagens em sinais físicos no meio de transmissão inclui a adição e remoção de preâmbulos no início e no final dos delimitadores e emprega a codificação Manchester para transmissão de sinal.

Está prevista na norma a possibilidade de duplicação do meio físico de transmissão de dados. Esta redundância atende os requisitos de tolerância a falhas, permitindo que, quando um dos meios de transmissão falhar a rede continua em operacional.

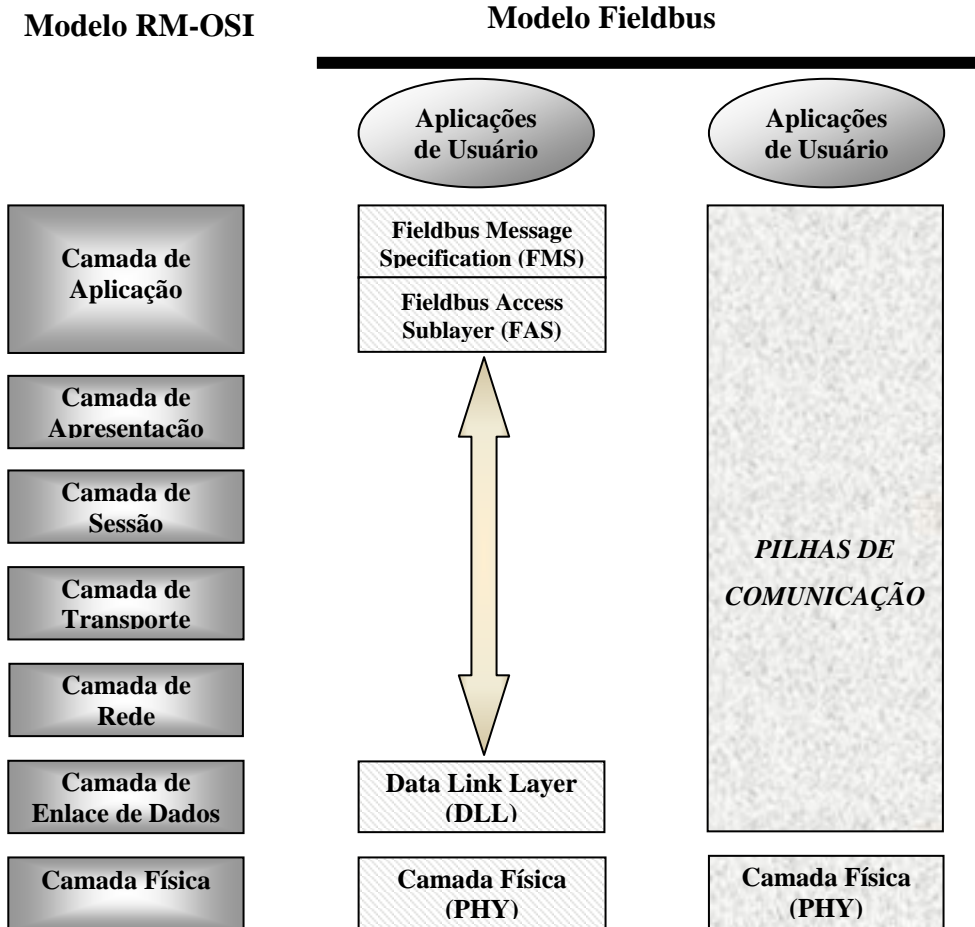


FIGURA 3.2 – Modelo de comunicação reduzido no FIELDBUS.

Para sistemas com meio físico redundante, cada barramento deve atender às regras de configuração da rede e não deve existir um segmento não-redundante entre dois segmentos redundantes. Os repetidores que são equipamentos responsáveis por regenerar e amplificar o sinal também deverão ser redundantes. Em ambientes intrinsecamente seguros estão previstos barramentos com ou sem energia.

O número de dispositivos possíveis em um enlace FIELDBUS é de 240, mas esse valor depende de alguns fatores para ser obtido. São permitidas duas topologias que podem ser em barramento linear ou em árvore.

Uma outra característica importante em um sistema FIELDBUS é que ele deve ser capaz de continuar operando enquanto um instrumento está sendo conectado ou desconectado. [SMA 98]

### 3.1.3 Nível de enlace (Data Link Layer)

Um dos serviços do nível de enlace é controlar o acesso ao meio de transmissão, determinando quem pode transmitir e quando. O nível de enlace garante também a integridade das mensagens através de dois bytes calculados através de um polinômio aplicado a todos os bytes da mensagem e que é acrescentado no final da mesma.

A camada de enlace de dados administra o acesso ao meio através do escalonador ativo da conexão LAS (Link Active Scheduler). É a LAS que determina quem tem acesso ao

barramento e em que momento uma estação pode iniciar os seus ciclos de mensagens. [FOU 2000]

Ela coordena o tempo necessário para cada transação na rede, garantindo o período de troca de dados. Ela também é responsável pela sincronização dos relógios na rede. Essa sincronização é necessária para garantir uma referência de tempo da rede a fim de obter a sincronização do barramento e das atividades de processo.

O acesso ao meio de transmissão na camada de enlace dá-se de três formas :

- a) Passagem de Token : o token é o modo direto de iniciar uma transmissão no barramento. Quando termina de enviar as mensagens, o dispositivo retorna o token para a LAS (Link Active Scheduler). A LAS transmite o token para o equipamento que requisitou, via préconfiguração ou via escalonamento.
- b) Resposta Imediata : o mestre dará uma oportunidade para uma estação responder com uma mensagem.
- c) Requisição de Token : um equipamento requisita um token usando um código em alguma das respostas que ele transmitiu para o barramento. A LAS recebe essa requisição e envia um token para o equipamento quando houver tempo disponível nas fases aperiódicas do escalonamento.

#### **3.1.4 Nível de aplicação (Application Layer)**

A camada de aplicação fornece uma interface para o software aplicativo do dispositivo. Basicamente essa camada define como ler, escrever ou disparar uma tarefa em uma estação remota.

O principal objetivo dessa camada é a definição de uma sintaxe para as mensagens que trafegam no barramento e também o modo pelo qual a mensagem deve ser transmitida : ciclicamente, imediatamente, somente uma vez ou quando requisitado pelo consumidor.

Uma outra função dessa camada é o levantamento estatístico de detecção de falhas e de adição de um novo elemento ou remoção de uma estação. O gerenciamento monitora continuamente o barramento para identificar a adição de novas estações. [SMA 98]

#### **3.1.5 Nível do usuário (User Layer)**

Define o modo para acessar a informação dentro de equipamentos FIELDBUS e de que forma essa informação pode ser distribuída para outros equipamentos no mesmo nó ou, eventualmente, em outros nós da rede. Esse atributo é fundamental para aplicações em controle de processo.

Como já foi dito anteriormente, a base de um equipamento FIELDBUS são os blocos funcionais. Um equipamento FIELDBUS conterà um número definido de blocos funcionais e as suas bases de dados podem ser acessadas via comunicação.

### **3.2 PROFIBUS (Process Fieldbus)**

O PROFIBUS originou-se na Alemanha em 1987 com a união de algumas empresas como Bosch, Siemens, Klockner-Moeller e mais 15 empresas e 5 institutos de pesquisa, financiados pelo governo alemão.

O principal objetivo do projeto PROFIBUS era desenvolver um padrão de comunicação para sistemas de controle e automação industrial, orientado a comunicação aberta.

Uma característica marcante no projeto é a grande utilização de padrões nacionais/internacionais já adotados pela indústria. [SAN 96]

A arquitetura a ser desenvolvida tinha que possibilitar ainda a interconecção com as redes MAP através de uma “brigde” que é um equipamento utilizado para interconectar duas redes que possuam as camadas de enlace (camada 2) diferentes, mas o mesmo protocolo para a camada de rede.

A “brigde” funciona como uma ponte (gateway), traduzindo o protocolo de uma rede para o protocolo da outra, porém é muito mais simples, uma vez que apenas os protocolos da camada 2 são incompatíveis.

Alguns requisitos foram adotados no projeto PROFIBUS :

- a) Técnicas de transmissão simples, dando preferência a padrões industriais existentes.
- b) Na camada de enlace foram implementadas funções essenciais para a comunicação : resposta imediata, Broadcast e Multicast de dados.
- c) Mecanismo de acesso aos instrumentos através do modelo mestre-escravo, com lista central de dispositivos chamados (central poling).
- d) Adoção do padrão MMS da ISO para a camada de aplicação somente para o PROFIBUS FMS.
- e) Para a redução dos custos de hardware e software, o contexto de comunicação e a manipulação de dados deveriam ser restritos a funções fixas que pudessem ser definidas e estabelecidas na fase de projeto de um sistema de controle ou automação. [SIL 96]

### **3.2.1 Características funcionais**

Como todo o padrão para redes FIELDBUS, o PROFIBUS adota o modelo ISO-OSI de multicamadas, porém com uma redução do número de camadas como mostra a figura 3.3.

As demais camadas não foram implementadas diretamente, apenas algumas de suas funções foram incorporadas dentro das camadas 2 e 7 do PROFIBUS.

A inclusão desses serviços das camadas eliminadas implicou uma subdivisão da camada de aplicação em duas sub-camadas : a LLI (Lower Layer Interface) e a FMS (Fieldbus Message Specification), sendo essa diretamente baseada na MMS (Manufacturing Message Specification) normalizada pela ISO.

Dentro desse contexto foram geradas as normas DIN (Deutsche Industry Norm) 19245 partes 1 e 2.

A parte 1 trata dos aspectos referentes ao meio físico de transmissão e ao método de acesso ao meio, envolvendo as camadas físicas e de enlace. A parte 2 aborda principalmente a camada de aplicação, definindo o método de acesso para as aplicações do usuário.

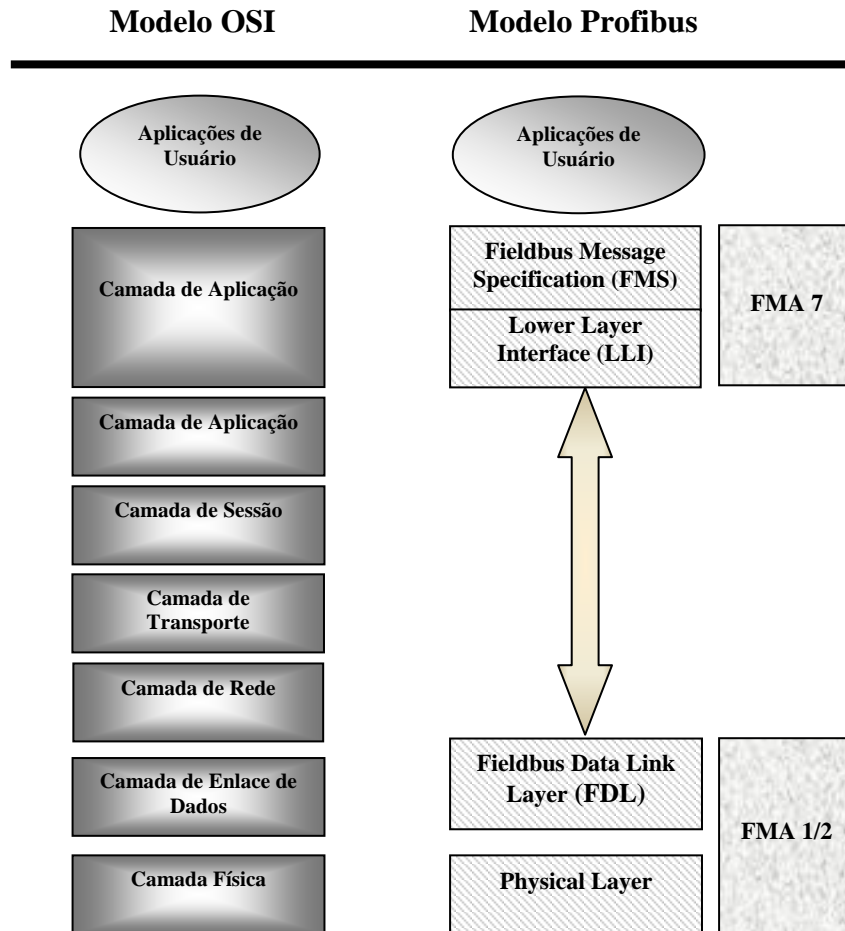


FIGURA 3.3 – Arquitetura de camadas PROFIBUS.

### 3.2.2 Nível físico

Para atender a diversos requisitos de topologia o PROFIBUS oferece a possibilidade de especificar vários padrões para as interfaces físicas, oferecendo diversos requisitos em relação a topologia, comprimento do barramento e taxa de transmissão.

O PROFIBUS-DP permite velocidades de transmissão de 9,6 Kbit/s a 12 Mbit/s. As distâncias permitidas variam conforme a velocidade e as características de cabeamento.

Para ambientes intrinsecamente seguros estão normalizados barramentos com ou sem energia. O número máximo de dispositivos em um barramento é de 127, esta característica esta disponível no PROFIBUS-PA.

O barramento utiliza par trançado blindado e não blindado. O cabo não blindado somente pode ser utilizado se não houver severas restrições de interferência eletromagnética.

A norma prevê topologia em barramento linear com terminadores nas duas extremidades ou topologia em árvore. A tabela 3.1 mostra as características de cada topologia.

O PROFIBUS emprega a codificação NRZ (No Return to Zero), isto é, o dado é transmitido através do cabo de par trançado de maneira que um bit 1 (um) seja representado por uma tensão positiva entre o pino 3 e o pino 8 e o binário 0 (zero) é representado por uma tensão negativa entre esses dois pinos do conector. [SIL 96]

TABELA 3.1 – Características da topologia do protocolo PROFIBUS

	Barramento Linear	Árvore
Meio de Transmissão	Par trançado	Par trançado
Comprimento Máximo da Linha	1200 m	2800 m
Número de Estações	32	-
Número de Estações com Repetidores *	127	122
Padrão da o meio de transmissão	RS-485	RS-485

\* No máximo três repetidores entre duas estações.

### 3.2.2.1 Interface entre a camada física e a camada de link de dados (FDL)

Na interface com a camada 2, a camada física realiza os serviços de recepção e transmissão de bits (chamados símbolos FDL), os quais são elementos UART (Unidade de Palavra de Informações para Transmissão Serial Assíncrona).

Os dados trafegam pela rede em blocos de dados chamados quadro. Cada quadro é formado por um conjunto de caracteres UART.

O padrão de caracter UART como mostra a figura 3.4 utiliza um SB (Start Bit), um ST (Stop Bit), oito bits de dados e um bit de paridade. A paridade utilizada é par e os bits SB, ST, P são inseridos ou removidos pelo chip controlador dessa camada.

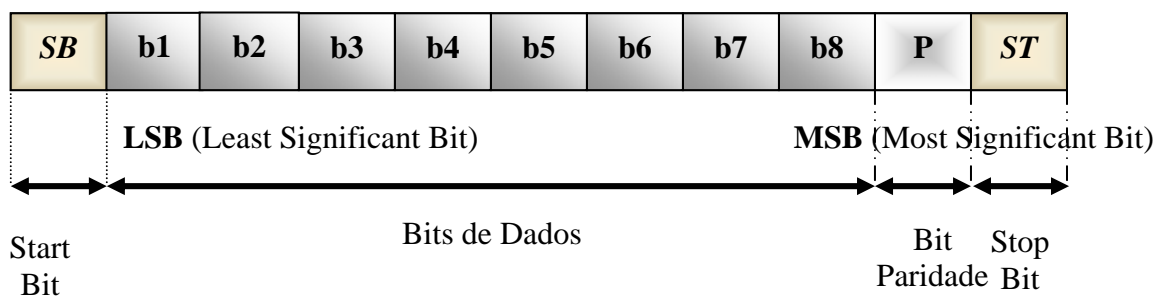


FIGURA 3.4 – Estrutura de um caractere UART

### 3.2.2.2 Redundância da camada física e do meio de transmissão

Foi especificada, dentro das normas, uma redundância para as camadas físicas e para o meio de transmissão. Quando essa característica opcional é implementada, a camada física redundante contém dois meios de transmissão distintos, barramento A e B e dois transceivers por estação, um para cada barramento como mostra a figura 3.5.

Os dados são transmitidos simultaneamente pelos dois barramentos, sendo somente recebidos por um deles. Essa decisão de qual barramento será utilizado é feita pela camada 2, que monitora constantemente os dois barramentos e, em caso de anormalidade em algum deles, passa a utilizar somente o outro.



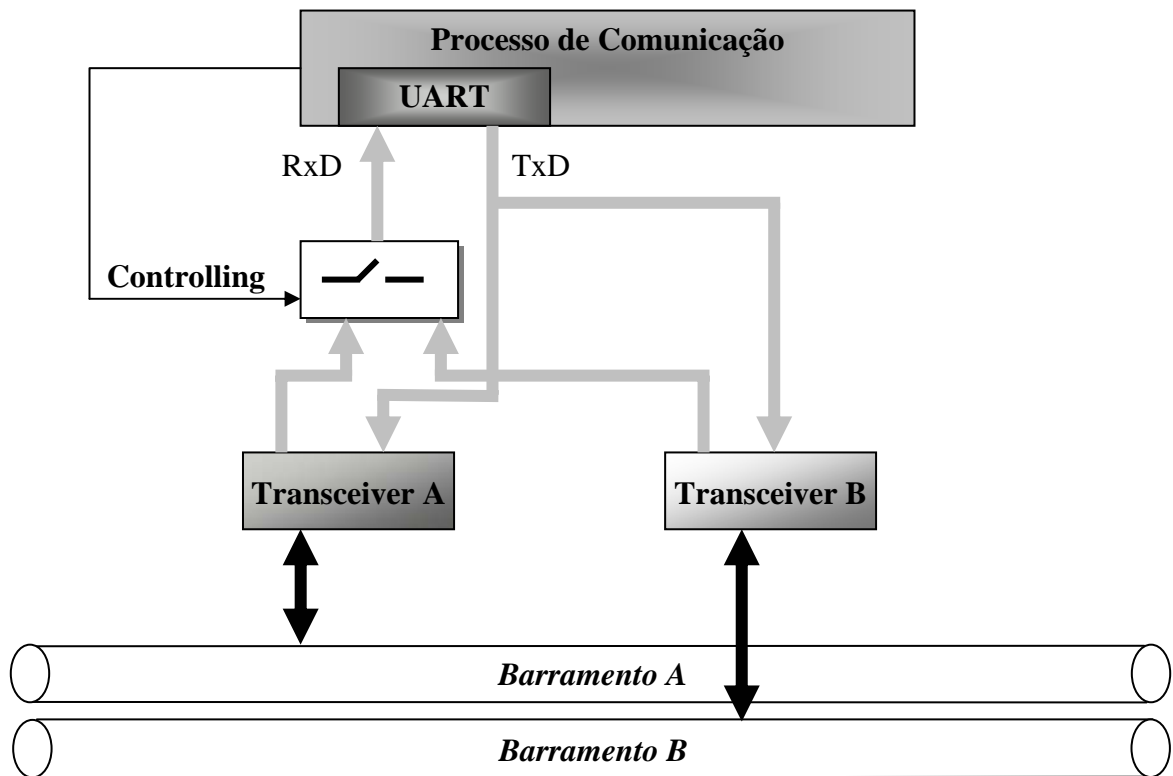


FIGURA 3.5 – Redundância no barramento

### 3.2.3 FDL (Fieldbus Data Link Layer)

A camada de enlace do PROFIBUS é denominada de FDL, sendo a responsável pela execução dos serviços referentes ao protocolo de transmissão de mensagens e de controle do acesso ao meio de transmissão.

Em paralelo aos serviços de transmissão está o gerenciador de camadas 1 e 2, denominado FMA (Fieldbus Management Layer) 1/2. Ele organiza a inicialização, a supervisão e o tratamento de erro entre seu usuário e as funções lógicas daquelas camadas, permitindo assim a correta operação do sistema.

Dentro da FDL são especificados todos os serviços referentes ao gerenciamento das mensagens, endereçamento das estações, controle do anel lógico e o mecanismo de resposta imediata da rede PROFIBUS.

Essas especificações são descritas mais detalhadamente abaixo, dando uma visão abrangente de como o protocolo garante os seus tempos de resposta e o controle do meio transmissão.

#### 3.2.3.1 Método de acesso ao meio de transmissão

O controle de acesso ao meio em aplicações de controle e automação devem atender a dois requisitos essenciais :

- 1) Deve ser assegurado para aplicações com certo grau de inteligência local para que cada estação tenha suficiente oportunidade de utilizar o barramento para as suas necessidades de comunicação. [SIL 96]

- 2) Também é necessário garantir a comunicação entre os sistemas mais complexos até os dispositivos mais simples, localizados periféricamente em relação à estação central. [SIL 96]

O método de acesso que o PROFIBUS utiliza cumpre esses dois requisitos. Foi definido um método de acesso híbrido ao barramento, combinando as vantagens dos métodos de passagens de token definido pelo padrão “*Token Bus*” descrito na seção 2.1.5 com o método mestre-escravo.

Como já explicamos anteriormente o primeiro deles é o princípio da passagem de token, somente as estações-mestre podem acessar o meio e, dentre elas, somente quem possuir o token é que tem a permissão de acessar o barramento num determinado período de tempo.

Uma vez expirado o tempo de posse do token, o mesmo é transmitido para a próxima estação, é desta forma que a FDL gerencia o acesso ao meio de comunicação.

O controle do token é gerenciado por cada estação-mestre através de uma lista chamada LAS (List of Active Station), nesta lista estão identificado o endereço da suas estações predecessoras PS (Previous Station), de quem recebe o token; da estação sucessora NS (Next Station) para quem deve enviar o token, e também o seu próprio endereço TS (This Station) como mostra a figura 3.6.

O segundo princípio é o chamado mestre-escravo: este implementa a comunicação entre os sistemas complexos, com capacidade de processamento de informações, e dispositivos simples situados ao seu redor.

Os equipamentos complexos são chamados mestres, enquanto que as unidades periféricas são denominadas escravos, pois executam serviços sob solicitação do mestre.

Devido a essas características relativas à permissão de acesso ao meio de transmissão as estações são também denominadas ativas e passivas, as primeiras sendo os mestres e as últimas escravos. [NAZ 93]

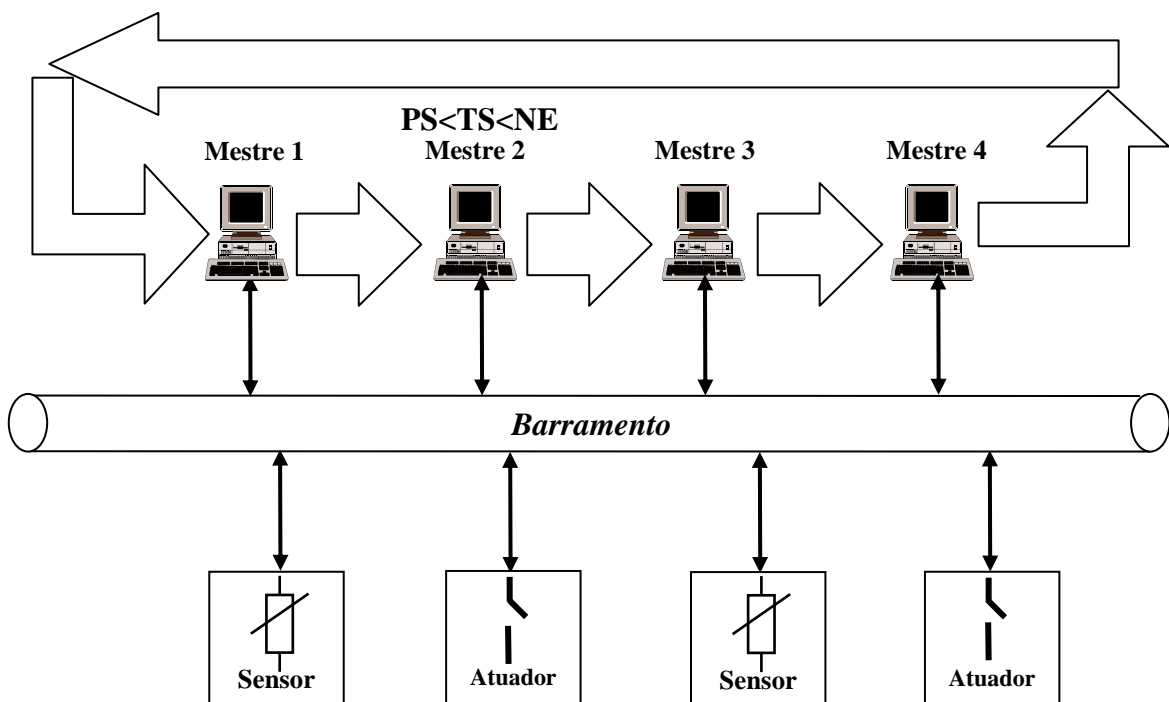


FIGURA 3.6 – Modelo híbrido de acesso ao meio de transmissão

As estações escravas são comumente sensores, chaves indicadoras de posição, válvula atuadoras ou quaisquer outros dispositivos controlados pelo mestre.

Os escravos não têm autoridade para acesso ao meio, portanto só podem participar de uma comunicação quando forem acionados por algum mestre. [SIL 96]

### 3.2.3.2 Lista das estações ativas

Como já foi dito anteriormente, cada uma das estações-mestre mantém internamente uma tabela denominada LAS, essa tabela contém os endereços de todas as estações ativas do sistema. Nessa tabela encontramos o PS e NS em relação ao mestre que gerencia a LAS.

Quando uma estação recebe um quadro de token endereçado a ela, verifica-se na LAS se a estação remetente está marcada como PS. Em caso afirmativo a estação começa a executar seus ciclos de mensagens.

No caso da estação remetente não estar marcada como PS, o quadro de token é recusado e apenas em uma segunda tentativa subsequente a estação deve aceitar o token, assumindo que o anel lógico sofreu alteração e, conseqüentemente, atualizando sua LAS.

A estação detentora do token iniciará os seus ciclos de mensagens e, após o término, a estação-mestre passará o token para a próxima estação do anel NS.

Se após a passagem do token e dentro de um tempo predeterminado (*slot time*), essa estação, que enviou o token, receber um cabeçalho de quadro válido, ela assume que a estação sucessora está executando os seus ciclos de mensagem; caso ela receba um quadro inválido assume que outra estação está com o token e em ambos os casos entra em um estado inativo.

Uma outra situação é detectada, se após a transmissão do token ela perceber que não há atividade no barramento dentro do tempo *slot time*, o token é transmitido novamente para a NS.

Se a estação reconhecer alguma atividade no barramento entra em estado inativo. Caso nessa segunda tentativa não reconheça atividade no meio, ela tenta passar o token a estação seguinte à NS, repetindo todo o procedimento descrito acima até que encontre outra estação. Caso ela perceba que é a única estação-mestre no anel, então o sistema é considerado um mestre-escravo puro.

A LAS é sempre gerada durante o estado “*Listen-Token*” e atualizada sempre que necessário.

### 3.2.3.3 Adição e remoção de estações a quente

A PROFIBUS permite a adição ou remoção de estações no barramento sem a necessidade de reinicialização da rede.

Cada estação-mestre é responsável em manter uma tabela contendo os endereços de todas as estações mestres e escravos situadas entre o seu próprio endereço TS e o endereço da estação-mestre sucessora NS no anel lógico, em ordem ascendente, sendo essa tabela denominada GAP List.

A alocação de novas estações é feita nos espaços existentes entre o endereço da estação e o da próxima NS, estes espaços estão armazenados na GAP.

A manutenção da GAP ocorre periodicamente, as estações-mestre consultam as estações pertencentes em suas GAPLs, através do envio do quadro especial “*FDL Request Status*”.

Essa consulta ocorre uma vez a cada ciclo de mensagem. Quando um endereço não responder a duas tentativas subseqüentes, são marcados como estações desativadas. Essa manutenção sempre se dá após a transmissão de todas as mensagens pendentes e quando houver tempo disponível para realizar esse serviço.

#### **3.2.3.4 Inicialização e reinicialização do anel lógico**

A inicialização do sistema é um caso particular da manutenção da GAPL. Quando durante o estado “*Listen-Token*” uma estação-mestre não observa atividade no meio de transmissão ela deve requisitar o token para si e começar a inicialização.

Na inicialização de todo o sistema PROFIBUS, a estação de menor endereço transmite para si mesma dois token frames indicando às demais ser a única no anel.

Após isso transmite um quadro de “*Request FDL Status*” para as demais, passando a formar a sua GAPL com as estações não prontas para a comunicação.

A primeira estação-mestre a responder indicando estar pronta é marcada como NS recebendo o token, assim a LAS é formada e a GAP, entre a TS e a NS é calculado.

Quando uma estação-mestre não detecta atividade no meio, durante um intervalo de tempo “*Slot Time*” pré-estabelecido, ela deve requisitar o token, recebê-lo e começar o processo de reinicialização.

A reinicialização do anel se faz necessária quando o token é perdido, sendo um processo bem mais simples que a inicialização, uma vez que a LAS e a GAPL já existem. Assim, a estação-mestre de menor endereço pega o token e passa a executar ciclos de mensagens ou transmitindo-o para a sua NS.

#### **3.2.3.5 Estrutura dos quadros**

Nas normas do PROFIBUS foram definidos cinco tipos de quadros como mostra a figura 3.7, todos tendo distância “*Hamming 4*”, o que significa que até três bits errados simultaneamente podem ser detectados.

Os quadros SC e SD1 são quadros de tamanho fixo e sem campo de dados, os SD2 são quadros de tamanho variável e com campo de dados, os SD3 são de tamanho fixo com campo de dados e finalmente os SD4 são os quadros de token.

O quadro SC contém somente um byte, é utilizado em situações de reconhecimento e em algumas situações especiais.

O do tipo SD1 contém 6 bytes e não contém campo de dados. Como já foi dito anteriormente, é utilizado nos quadros de reconhecimento “*acknowledge*” e também nos pedidos de “*request*”, devido ao mecanismo de resposta imediata, o qual será tratado na seção 3.2.3.9.

Os quadros SD2 e SD3 são utilizados para transações de pedido e envio de dados e também como quadros de resposta. Nessa situação a FDL pode escolher entre esses dois tipos, cuja diferença está em que o campo de dados do SD3 possui tamanho fixo (1 byte), ao passo que no SD2, o número de bytes de dados pode ser de até 246.

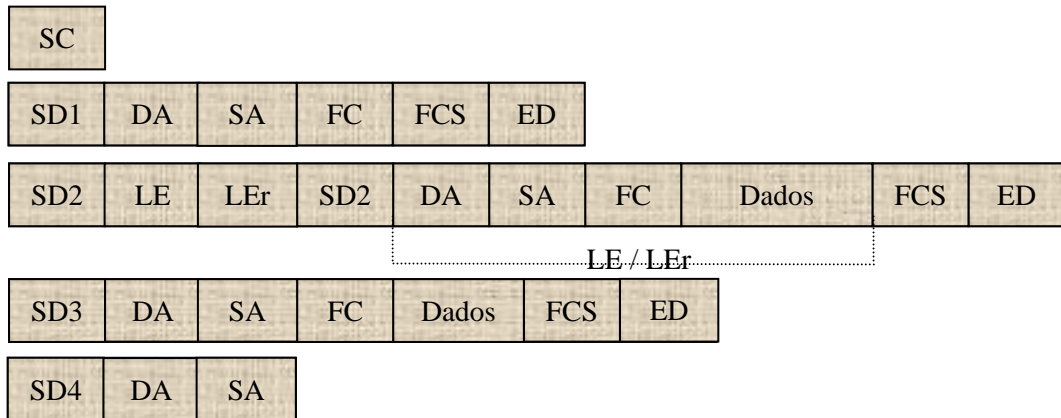


FIGURA 3.7 – Estrutura dos quadros no PROFIBUS

E finalmente o SD4 com 3 bytes que é utilizado para a transmissão do token entre as estação-mestre.

A função de cada campo que compõe um quadro é mostrada na figura 3.7 e descrita a seguir.

- SDn : tipo do quadro.
- DA : Endereço da estação de destino.
- SA : Endereço da estação que emitiu o quadro.
- FC : Quadro de controle (quadro control).
- FCS : Cheksum dos campos DA,SA, FC e dados se houver.
- LE/LEr : Tamanho do campo DA, SA, FC e dados do quadro SD2.
- ED : Delimitador final.

O campo FC dos quadros SD1, SD2, SD3, além de outras informações de controle, identifica a função do quadro. De acordo com a configuração de seus bits, pode-se identificar o tipo de estação que enviou o quadro, se o quadro é de pedido, resposta ou confirmação e a natureza das informações que contém.

### 3.2.3.6 Endereçamento de estações

Para verificar se um determinado quadro é endereçado a uma estação todas as demais estações, com exceção da que fez o pedido, monitoram todos os pedidos, respondendo somente quando forem endereçadas a elas.

Cada estação que está monitorando o barramento deve ler o campo DA para verificar se contém o seu endereço. Em caso afirmativo, o restante do quadro é lido e processado, mas se o endereço não for para a estação o quadro deve ser descartado.

O PROFIBUS possibilita o endereçamento de 128 estações (0 a 127). Os sete bits menos significativos dos campos DA e SA formam esses endereços.

O bit mais significativo (b8) indica se um ou mais bytes de extensão de endereço estão sendo transmitidos. Esses campos são chamados de DAE (Destination Address Extension) e SAE (Source Address Extension). Esses bytes de extensão são utilizados em duas funções :

A primeira função é para a transmissão do LSAP (Link Service Access Point), utilizado na conexão para realizar o interfaceamento da LLI com a FDL. Os campos contendo os LSAP destino/origem seguem após o campo FC, como primeiro campo do campo de dados, apenas para quadros SD2.

A segunda função é para referenciar endereços em outras redes ou sistemas, aos quais o PROFIBUS esteja interligado. Nesse caso, os campos DA e SA endereçam a estação origem/destino que faz a interligação através de bridges.

Nos campos de endereço estendido, apenas seis bits constituem o endereço. Os dois bits mais significativos b8 e b7 estabelecem respectivamente:

- A existência de um campo de endereço estendido adicional, seguindo imediatamente após.
- A função do campo de endereço estendido, se portador do LSAP ou endereço no sistema interligado.

### **3.2.3.7 LSAP**

A troca de informações entre a LLI e FMS é realizada através dos pontos de acesso de serviços às LSAP. Com a LSAP é possível separar as informações recebidas e enviadas de acordo com as aplicações (conexão ou serviços) às quais se destinam.

Assim, podem ser atribuídas várias LSAP específicas para cada aplicação. As LSAP do destino e da origem são tratados internamente pela FDL por DSAP (Destination Service Access Point) e SSAP (Source Service Access Point).

Como foi explicado no item anterior, as LSAP são transmitidas nos campo de extensão de endereço. A LSAP da estação destinatária é transmitida no campo DAE (Destination Address Extension) e atribuída ao DSAP, enquanto que o da estação remetente é transmitido no campo SAE (Source Address Extension), sendo atribuído ao SSAP.

Uma vez que apenas 6 bits são utilizados para representar o LSAP no quadro, somente são permitidos 64 LSAPs por estação (0-63).

### **3.2.3.8 Prioridade de mensagens**

A camada de aplicação pode determinar para a FDL ciclos de mensagens com prioridade alta ou baixa.

Obviamente, as mensagens de alta prioridade são enviados primeiro, sendo que as mensagens de baixa prioridade só serão transmitidas se houver tempo. Caso não haja tempo, serão enviadas nos próximos recebimentos de token.

O PROFIBUS utiliza duas variáveis de tempo para determinar se há ou não tempo para transmitir mensagens de prioridade baixa.

O padrão é que cada estação pode transmitir uma mensagem de prioridade alta; se houver tempo, este pode ser utilizado para transmitir as mensagens de baixa prioridade e realizar a manutenção da GAP.

Esses tempos são o tempo real de rotação ( $T_{RR}$ ) e o tempo de rotação desejado ( $T_{TR}$ ) sendo que  $T_{RR} < T_{TR}$ . Será dada uma abordagem mais completa em relação aos tempos de controle do PROFIBUS no capítulo 4.

Uma vez iniciado o ciclo de transmissão ela deve ser completada. Se por algum motivo o tempo de posse do token expirar, o ciclo é executado até o final, causando uma

diminuição no tempo de transmissão dos ciclos de mensagem em futuros recebimentos de tokens.

### **3.2.3.9 O mecanismo de resposta imediata**

A característica de tempo real em uma rede PROFIBUS é garantida pelo mecanismo de resposta imediata que é um dos aspectos mais importante do protocolo FDL. Esse mecanismo consiste na transmissão imediata da resposta, assim que for recebido um quadro de pedido.

Quando o mecanismo de resposta imediata é ativado, a estação-mestre que emitiu o pedido passa continuamente a enviar quadros de pedido para a estação remota que pode ser um mestre ou escravo.

Na estação remota, a recepção de cada um desses quadros força a leitura de um área de memória especial, a memória de atualização ou memória de transmissão da camada FDL, e a transmissão dos dados nela contidos para a estação local, mesmo que não hajam dados armazenados na memória de transmissão. Nesse caso, um quadro sem dados é transmitido.

Desse modo, as estações locais e remota ficam trocando seus quadros indefinidamente, enquanto não for desativado o mecanismo.

Quando a LLI fornece os dados de um pedido, esses dados são transmitidos uma única vez. O pedido é recebido pela estação remota e repassado pela FDL à sua usuária, que o faz chegar até ao usuário da estação. Esse, executa a operação requerida e através da LLI, armazena a resposta devida na memória de transmissão.

Assim, quando os quadros de pedido subseqüentes foram recebidos na estação remota, a resposta do usuário remoto será lida da memória de transmissão e enviada para a estação local, encerrando o ciclo de mensagem corrente.

No entanto, se o mecanismo de resposta imediata permanecer ativado, as estações continuarão a trocar seus quadros de pedido e resposta, mesmo que não transportem dados.

### **3.2.3.10 Serviços de transferência de dados da camada de Enlace**

Além de controlar o acesso ao meio e de tratar os quadros, a camada FDL é responsável pelo fornecimento de serviços de transferência de dados para a camada superior à LLI.

Estão subdivididas em serviços que demandam quadro de reconhecimento, que não demandam, serviços acíclicos e cíclicos.

Nos serviços acíclicos os ciclos de mensagens são executados esporadicamente, e em apenas um único ciclo de mensagem.

Já nos serviços cíclicos os ciclos de mensagens exigem várias trocas de mensagem entre a estação ativa, a que detém o token, e as estações passivas ou estações ativas não detentoras do token.

Uma tabela interna denominada de GAP gerencia qual a seqüência dos ciclos de mensagens que a estação ativa deve realizar. Essa tabela contém todos os endereços das estações que serão acionadas.

Para cada estação relacionada nessa tabela um quadro de pedido é enviado e um quadro de resposta é aguardado seqüencialmente e com baixa prioridade.

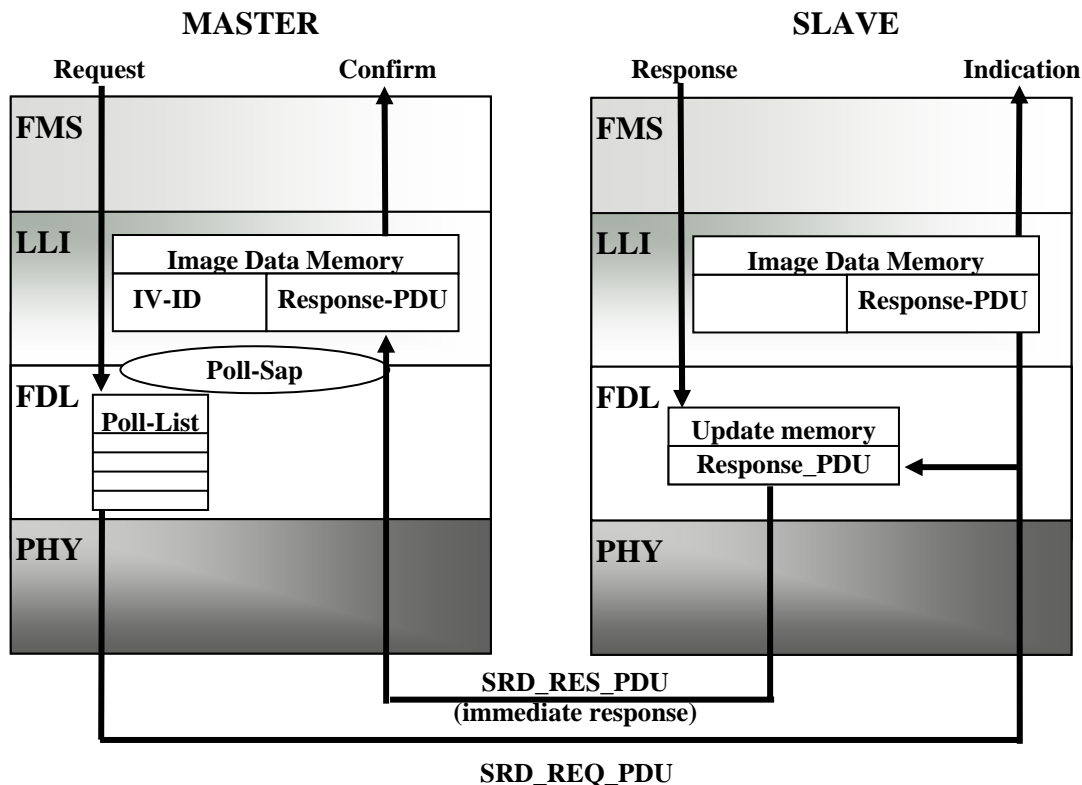


FIGURA 3.8 – Sequência de um serviço cíclico

Essa tabela é chamada de “*poll list*”, pois é ela que determina o polling dos escravos, ou seja, o mecanismo de acionar sequencialmente as estações escravos para transferência de dados.

A transmissão das mensagens cíclicas ocorre somente após o término das transmissões de mensagens acíclicas.

A camada FDL oferece à camada 7 a possibilidade de escolha da prioridade com que se efetuará a transmissão. São definidos dois níveis de prioridade : alta ou baixa. Deste modo, sempre que uma estação ativa receber o token, ela transmitirá primeiro as mensagens de prioridade alta.

Após terminada as mensagens de alta prioridade e havendo tempo restante, as mensagens de baixa prioridade serão enviadas. Se esgotar-se o tempo de token, a transmissão das mensagens pendentes se dará no próximo ciclo.

A FDL oferece quatro tipos de transferência de mensagens para o seu usuário a LLI, sendo divididos quanto a maneiras de como são executados. São elas :

- Envio de dados sem reconhecimento : SDN (Send Data with No Acknowledge)
- Envio de dados com reconhecimento : SDA (Send Data with Acknowledge)
- Envio/Pedido de dados com resposta : SRD (Send and Request with Acknowledge)
- Envio/Pedido cíclico de dados com resposta : CSRD (Cyclic Send and Request Data with Reply).



### 3.2.3.11 Envio de dados sem reconhecimento

Esse serviço permite à FDL a transferência de dados a estações remotas sem que ela aguarde confirmação de recepção. Esse serviço é utilizado para serviços não confirmados da LLI.

O dado “*L\_sdu*” pode ser enviado para uma, várias ou todas as estações da rede. A transferência pode ser multicast ou broadcast sendo executada tudo de um só vez. São utilizadas 3 primitivas nesse serviço :

1. *FDL\_DATA.request*

Esta primitiva é utilizada pelo usuário para solicitar a transmissão dos seus dados. Os parâmetros informados são : endereço remoto, os LSAPs local e remoto, o “*L\_sdu*” e a prioridade de solicitação.

Para pedidos em multicast o DSAP tem que ter um valor comum ao grupo das estações-destino. Em pedidos broadcast deve-se utilizar o endereço global para as estações remotas “*Remm\_Add(127)*” e o “*DSAP(63)*”.

2. *FDL\_DATA.indication*

Tem a finalidade de indicar à LLI remota a recepção de um quadro SDN. Os parâmetros são o endereço de origem, “*L\_sdu*” e a prioridade

3. *FDL\_DATA.confirm*

Indica para a LLI local a conclusão do serviço SDN, informando também os códigos de erros ocorridos, se for o caso.

### 3.2.3.12 Envio de dados com reconhecimento

Esse serviço permite que o usuário local de uma estação-mestre envie seus dados “*L\_sdu*” para uma única estação remota. Se os dados forem recebidos sem erro pela estação remota, a FDL remota transmite um quadro de reconhecimento e passa o “*L\_sdu*” para seu usuário.

Se a FDL local não receber esse quadro dentro do período de “*slot time*”, o quadro é retransmitido novamente, repetindo-se o processo até atingir o parâmetro “*Max\_retry\_limit*” que é configurado no controlador.

Só depois de receber o quadro de reconhecimento que a FDL local passa a confirmação para o seu usuário local. As primitivas desse serviço são:

1. *FDL\_DATA\_ACK.request*

Essa primitiva é utilizada pelo usuário para solicitar a transmissão dos seus dados. Os mesmos parâmetros da “*FDL\_DATA.request*” são informados, menos os endereços globais “*Remm\_Add(127)*” e o “*SSAP(63)*”, que não podem ser utilizados.

2. *FDL\_DATA\_ACK.indication*

É passada da FDL remota para à LLI remota indicando a recepção de um quadro SDA. Os parâmetros são os mesmos da primitiva “*FDL\_DATA.indication*”.

3. *FDL\_DATA\_ACK.confirm*

É passada da FDL local para a LLI indicando o fim da transmissão. Em caso de ocorrência de erros, são fornecidos códigos para a identificação das causas.

### 3.2.3.13 Envio/Pedido de dados com resposta

Esse serviço permite que sejam ao mesmo tempo enviados e recebidos dados da LLI da estação remota [SIL 96]. Também é permitida a solicitação de um dado a uma estação remota sem a necessidade de enviar dados para a mesma.

Com o serviço SDR a estação remetente pode enviar dados requisitando que a estação destinatária execute alguma operação e retorne o resultado dessa operação com resposta.

Quando a estação remetente solicita esses dados para uma outra estação é necessário que alguns dados como a especificação da operação desejada, parâmetros exigidos e outros itens sejam transmitidos.

Nesse caso o serviço SRD ou CSRD deve ser utilizado para a transmissão desses dados e recepção da resposta.

Esse serviço ativa o mecanismo de resposta imediata, sendo assim, a execução do serviço ocorre em prioridade alta.

#### 1. *FDL\_DATA\_REPLY.request*

Essa primitiva é utilizada para ativar o mecanismo de envio e recepção simultânea de dados. Os parâmetros são os dados a serem transmitidos “*L\_sdu*” que pode ter tamanho zero, o endereço FDL da estação-destino “*Remm\_Add*”, o “*DSAP*” e o “*SSAP*”, “*Serv\_Class*”.

#### 2. *FDL\_DATA\_REPLAY.indication*

Passado da FDL remota para à LLI remota indicando a recepção de um quadro com dados de execução de serviço SDR e sobre a transmissão ou não do quadro de resposta. Os parâmetros são SSAP, DSAP, “*Remm\_Add*”, “*Serv\_Class*”, “*L\_Status*”

#### 3. *FDL\_DATA\_REPLAY.confirm*

É passada da FDL local para a LLI indicando o sucesso ou falha na execução da primitiva de requisição “*request*”. No caso da execução sem erros, os dados recebidos são repassados.

Caso o serviço não tenha sido completado, os erros ocorridos e suas possíveis causas são então informados à LLI.

#### 4. *FDL\_REPLAY\_UPDATE.request*

A primitiva de atualização é utilizada pela LLI da estação remota para carregar o buffer de transmissão da FDL com os dados de resposta ao pedido da estação local.

São passados como parâmetros, além dos dados da resposta, o LSAP da própria estação remota que está configurado para a execução deste serviço e o indicador de que os dados devem ser transmitidos uma única vez, ou múltiplas vezes, no caso de subsequentes tentativas da estação local.

#### 5. *FDL\_REPLAY\_UPDATE.confirm*

Essa primitiva confirma à LLI da estação remota que os dados de resposta a um pedido foram corretamente armazenados na memória de transmissão, através da primitiva “*UPDATE.confirm*”

### 3.2.3.14 Envio/Pedido cíclico de dados com resposta

Através do serviço CSRD, podem-se transmitir dados à estação remota, solicitando a execução de determinada operação, assim como no serviço SRD.

Porém, no serviço CSRD, essa solicitação é feita de forma cíclica, isto é, por meio do mecanismo de resposta imediata, a operação é continuamente requisitada à estação remota. Além disso, é possível realizar a transmissão/recepção de dados para várias estações consecutivamente na ordem definida pela “*pool-list*”.

#### 1. *FDL\_SEND\_UPDATE.request*

Esta primitiva é passada pela LLI da estação local para entregar à FDL os dados a serem transmitidos. Os parâmetros utilizados são o endereço FDL da estação remota, o “*SSAP*” e o “*DSAP*”, e o parâmetro “*transmit*”, que indica se os dados da LLI devem ser transmitidos uma única vez ou repetidos em cada quadro.

#### 2. *FDL\_SEND\_UPDATE.confirm*

Utilizada para confirmar ao usuário FDL que os dados para transmissão foram recebidos corretamente.

#### 3. *FDL\_CYC\_DATA\_REPLY.request*

A LLI utiliza essa primitiva para entregar à FDL a “*pool-list*”, com os endereços das estações que devem ser acionadas com o serviço CSRD. A estrutura da “*pool-list*” é mostrada na tabela 3.2.

#### 4. *FDL\_CYC\_DATA\_REPLY.confirm*

Essa primitiva é utilizada tanto para confirmar à LLI da recepção da “*pool-list*”, como para indicar a recepção de cada quadro dentro da execução de transmissões cíclicas, se estes portarem dados de resposta da estação remota.

#### 5. *FDL\_CYC\_ENTRY.request*

A primitiva “*CYC\_ENTRY*” é utilizada para habilitar a linha da “*pool-list*” correspondente a uma determinada estação, permitindo que esta participe do mecanismo de resposta imediata, recebendo os quadros continuamente. Os parâmetros são : o endereço FDL da estação remota, o “*LSAP*” correspondente, o “*LSAP*” da “*pool-list*” e, obviamente, a marca “*lock*”, “*unlock*”.

#### 6. *FDL\_CYC\_ENTRY.confirm*

A FDL indica o sucesso ou falha na primitiva “*CYC\_ENTRY.req*”.

#### 7. *FDL\_CYC\_DEACT.request*

Com esta primitiva a LLI encerra o mecanismo de resposta imediata. A FDL ao recebê-la, aciona as estações que ainda não foram acionadas no ciclo corrente, segundo a ordem da “*pool-list*”, e a seguir encerra as transmissões.

A “*pool-list*” é então descartada. Para a reativação do mecanismo, é necessário que a LLI forneça nova “*pool-list*”, com a primitiva

“CYC\_DATA\_REPLY.reques”t. O “LSAP” da “pool-list” é passado como parâmetro.

8. *FDL\_CYC\_DEACT.confirm*.

As primitivas “*FDL\_DATA\_REPLY.indication*”, “*FDL\_REPLY\_UPDATE.request*” e “*FDL\_REPLY\_UPDATE.confirm*” são utilizadas na estação remota assim como foi explicado no serviço SRD.

TABELA 3.2 – Estrutura da pool list

Nº Linha	Conteúdo	Marca	Significado
001	Comprimido		Número de linhas da pool-list
002	REM_ADD1	LOC/UNL	Endereço FDL (DA) da estação 1
003	DSAP1		LSAP (DAE) utilizado na estação 1
004	REM_ADD2	LOC/UNL	Endereço FDL da estação 2
005	DSAP2		LSAP utilizado na estação 2
006	REM_ADD3	LOC/UNL	Endereço FDL da estação 3
007	DSAP3		LSAP utilizado na estação 3
.....	.....	.....	.....
N	REM_ADDn	LOC/UNL	Endereço FDL da estação n
n+1	DSAPn		LSAP utilizado na estação n

### 3.2.3.15 Status do controlado FDL

Em uma estação-mestre o controlador FDL pode assumir 10 estados possíveis e uma estação-escravo pode assumir 2 estados. Os estados do controlador FDL são descritos a seguir.

#### 1. Estado Offline

Existem duas situações onde a estação entra no estado “*offline*”. A primeira é imediatamente após ao ser ligada e emitido um “*reset*” pelo gerente FMA 1/2.

É comum as estações, após serem ligadas, realizarem um POST (Power On Self Test). A segunda situação é quando, após o reconhecimento de determinados estados de erros que são detectados na estação. A estação permanecerá em “*offline*” até que estejam inicializados todos os parâmetros de operação necessários.

#### 2. Estado Passive Idle

Uma estação fica nesse estado após ter sido concluído o processo de inicialização. A estação fica monitorando o barramento e respondendo a possíveis pedidos a ela endereçados, devendo descartar apenas os token quadros a ela endereçados.

Se por algum motivo ocorrer algum erro, a estação volta para o estado “*Offline*”.

### 3. Estado Listen\_Token

Esse estado é atingido quando termina a inicialização dos parâmetros operacionais e a estação-mestre está pronta para entrar no anel lógico. Neste momento a estação-mestre irá formar a sua LAS através da análise das estações endereçadas nos token frames.

Essa análise ocorre durante duas rotações de token. Se os endereços forem idênticos a LAS é formada e a estação permanece nesse estado até que a sua PS solicite um “*Request FDL Status*” ao qual deve responder e estar pronta para entrar no anel “*Read to Entre Logical Token Ring*”, após a resposta a estação passa para o estado “*Active\_Idle*”.

Durante a formação da LAS, quando solicitada, a estação deve responder com não pronta para entrar no anel “*Not Ready*”. Caso também durante o formação da LAS a estação perceber em dois token o seu endereço no campo SA “*Source Address*”, ela deve assumir que outra estação possui o mesmo endereço, a estação deve retornar para o estado “*offline*” e informando ao gerente FMA 1/2.

Se dentro de um período de “*time-out*” a estação não perceber atividade no barramento a estação deve reinicializar a anel lógico.

### 4. Estado Active\_Idle

Nesse estado a estação deve monitorar o barramento passando a responder a pedidos a ela solicitados. Caso receba o token ela deve passar para o estado “*Use\_Token*”. Se dentro de um período de “*time-out*” não for detectado nenhuma atividade no barramento é necessária a reinicialização do anel, a FDL passa para o estado “*Claim\_Token*”.

### 5. Estado Claim\_Token

Nesse estado a FDL deve tentar reinicializar o anel dando início ao procedimento descrito no item 4.2.4. Esse estado é assumido pela FDL após um “*Listen\_Token*” ou “*Active\_Idle*”.

### 6. Estado Use\_Token

A FDL entra nesse estado após o recebimento do token ou uma reinicialização do anel. Aí serão executados os ciclos de mensagem de alta ou baixa prioridade com toda a monitoração dos tempos envolvidos. Quando um mensagem é transmitida a FDL passa para o estado “*Await\_Data\_Response*”.

### 7. Estado Await\_Data\_Response

Após a transmissão de um quadro de ação a FDL então espera um “*slot time*” para recebimento do quadro de resposta ou de reconhecimento. Quando se tratar de um serviço sem reconhecimento a FDL volta ao estado “*Use\_Token*”.

Em se tratando de pedidos com reconhecimento ou resposta a FDL deve esperar um dos eventos abaixo:

a) A chegada de um quadro de resposta ou reconhecimento endereçado a quem solicitou o serviço, processando o quadro e voltando ao estado “*Use-Token*”.

b) Recebimento de outro quadro válido indicando a ocorrência de erro.

Quando ocorrer o recebimento de um quadro inválido ou o fim do tempo de “*slot time*”, é feito um novo pedido e, se ainda assim quadros de resposta ou reconhecimento corretos não forem recebidos, a FDL deve notificar ao seu usuário e entrar em estado “*Use-Token*”.

Se no início do “*Use-Token*” não existir nenhum ciclo de mensagens de alta prioridade ou após o término de um ciclo de mensagens de alta ou baixa prioridade, a FDL deve passar para o estado “*Check-Access-Time*” para verificar a disponibilidade do token. [PRO 96]

### **8. Estado *Check-Access-Time***

Nesse estado ocorre a verificação da disponibilidade de tempo de posse do token. Se houver tempo disponível a FDL retorna para o estado “*Use-Token*”, caso contrário, passa para o estado “*Pass-Token*”.

### **9. Estado *Pass-Token***

Nesse estado a estação deve tentar passar o token para a próxima estação seguinte à NS. Durante a transmissão do token ela deve verificar se o transceiver está funcionando corretamente. Caso ela não receba seu próprio token quadro é detectado como erro, devendo a FDL entrar em estado “*Offline*” e notificando o gerente FMA 1/2.

Se for verificado que o token se encontra-se danificado a FDL passa para o estado “*Ckeck-Token-Pass*” que é o mesmo estado quando recebe esse quadro corretamente, se após uma retransmissão do quadro novamente ocorrer um erro a estação cessa sua atividade e a FDL passa para o estado “*Listen-Token*” e notifica o gerente FMA 1/2.

Nesse estado também é realizado a manutenção da GAPL com a introdução de novas estações-mestre e escravos que desejam entrar no anel.

### **10. Estado *Check-Token-Pass***

Entrando nesse estado a FDL espera um slot time para uma reação da estação à qual o token foi enviado. Caso detecte um quadro “*header*” válido, ela assume ter sido correta a passagem do token. Quando detecta um quadro inválido assume que a outra estação está ativa e, nos dois casos, passa ao estado “*Active-Idle*”. Caso ela não receba nenhum quadro dentro do slot time a FDL volta ao estado “*Pass-Token*”.

### **11. Estado *Await-Status-Response***

A FDL entra nesse estado, vindo do estado “*Pass-Token*” quando não conclui por não saber qual seu sucessor, durante a manutenção da GAP ou inicialização. A FDL deve esperar um “*slot time*” e se neste tempo não receber nenhum quadro ou um quadro contendo erro, volta ao estado “*Pass-Token*” para nova passagem. Caso receba outro quadro qualquer

(com exceção de quadro de reconhecimento) a FDL vai para o estado “Active\_Idle”.

A figura 3.9 mostra um diagrama com todas as possíveis transições de estados que podem ocorrer na camada FDL.

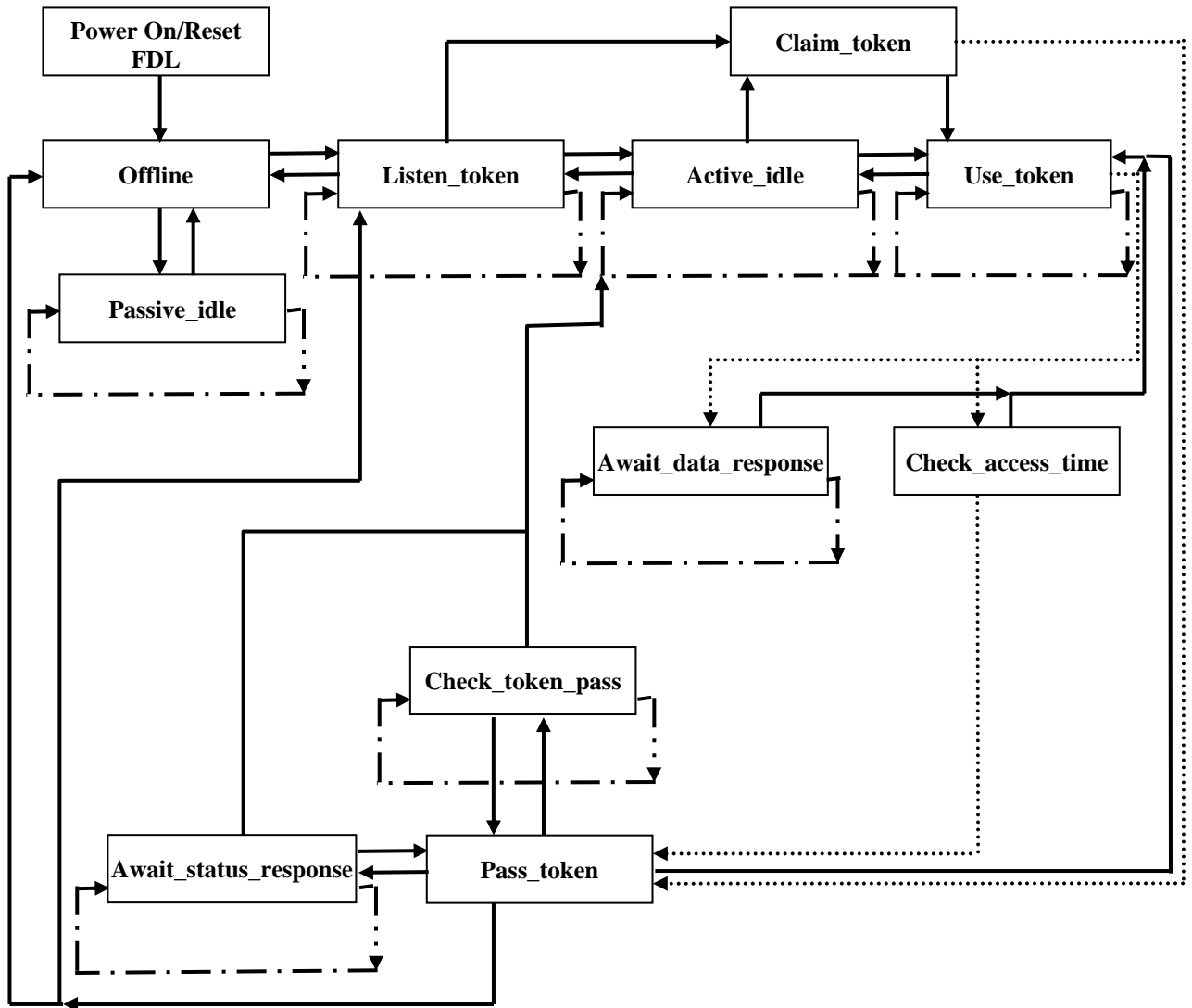


FIGURA 3.9 – Diagrama de estados da FDL

### 3.2.4 Gerenciador das Camadas 1 e 2 (FMA 1/2)

O gerenciador é responsável pela inicialização das camadas físicas e enlace, a supervisão e manipulação de erros. O gerenciador também presta serviços às camadas superiores à LLI e ao gerenciador da camada 7, FMA 7.

O FMA 1/2 utiliza os serviços das camadas 1 e 2 para a execução dos seus próprios serviços, solicitados pelos seus usuários.

As funções locais da gerenciador das camadas 1 e 2 são :

- recomposição das camadas 1 e 2.
- leitura e escrita em variáveis

- ativação, configuração e desativação de LSAPs
- mensagens de ocorrência de eventos e erros
- identificação da versão
- determinação da configuração dos LSAPs.

E as funções remotas executadas pelo gerenciador são :

- identificação da versão
- pedido da configuração de LSAPs
- determinação da lista de estações em funcionamento “live list”.

Para executar essas funções, o FMA 1/2 disponibiliza aos seus usuários os seguintes serviços dispostos na tabela 3.3.

TABELA 3.3 – Serviços executados pelo FMA 1/2

SERVIÇO	DESCRIÇÃO	PRIMITIVAS	
<i>RESET FMA 1/2</i>	Desativação da FDL, PHY e FMA 1/2	req	con
<i>SET VALUE FMA 1/2</i>	Atribuir valor a variável FDL/PHY	req	con
<i>READ VALUE FMA 1/2</i>	Ler valor de variável	req	con
<i>EVENT FMA 1/2</i>	Informar ocorrência de eventos	ind	
<i>IDENT FMA 1/2</i>	Identificação de estação	req	con
<i>LSAP STATUS FMA 1/2</i>	Informar a configuração de LSAP	req	con
<i>LIVE LIST FMA 1/2</i>	Obtém a lista de estações ativas	req	con
<i>SAP ACTIVE FMA 1/2</i>	Configurar e ativar um LSAP local	req	con
<i>RASP ACTIVE FMA 1/2</i>	Configurar e ativar um LSAP remoto	req	con
<i>SAP DEACTIVE FMA 1/2</i>	Desativar um LSAP local	req	con

### 3.2.5 Camada de aplicação

No PROFIBUS a camada de aplicação é composta por 2 sub-camadas : a FMS e a LLI.

#### 3.2.5.1 A camada FMS

A camada FMS baseia-se na MMS oferecendo a sua funcionalidade através de serviços e objetos de comunicação padronizados.

As características particulares do PROFIBUS como :

- A adoção de um subconjunto da MMS (FMS) na camada de aplicação,
- Oferecimento de serviços cíclicos pela FDL,
- Método de acesso mestre/escravo e controle via Poll-List,
- Mecanismo de resposta imediata



Ausência das camadas intermediárias de rede, transporte, sessão e apresentação exigem que seja feita uma adaptação especial entre as camadas 2 e 7. Essa tarefa é realizada pela LLI.

As tarefas da LLI são :

- Controle de fluxo (substituição da camada de transporte)
- Abertura e fechamento de conexões
- Conversão dos serviços da FMS nos serviços da FDL (Mestre/Escravo, acionamento cíclico, resposta imediata, broadcast e multicast).

### 3.2.5.2 Objetos

Um dos objetivos da normalização de um protocolo específico para automação industrial era a possibilidade da interligação de diversos equipamentos de fabricantes diferentes.

É indispensável a definição de uma forma padrão de representação dos dados transmitidos no barramento, independente do fabricante, mas também é desejável que cada fabricante tenha um certo grau de liberdade na definição e construção de seus dispositivos.

Esse problema é resolvido no PROFIBUS através do enfoque orientado a objetos dado pela FMS. [SAN 96]

Cada variável ou recurso utilizado no processo de automação pode ser caracterizado por um conjunto de atributos como nome, endereço, tipo de dado etc. Todos esses atributos agrupados juntos formam o chamado objeto do processo.

As características dos objetos do processo assim definidos, quando utilizados no processamento local, dependem somente das necessidades específicas de cada dispositivo ou fabricante.

Mas, se o objeto do processo é tornado disponível para o mundo externo, via um sistema de comunicação, então ele deve possuir somente atributos e funções bem definidas e comuns a todos os fabricantes. Esse tipo de objeto é chamado de objeto de comunicação.

### 3.2.5.3 O modelo VFD (Virtual Fieldbus Device)

Um dispositivo virtual é uma abstração de uma determinada classe de dispositivo real, tal como é apresentada ao usuário externo. O modelo VFD representa de forma única qual parte do processo aplicativo é visível e acessível pelo sistema de comunicação e define seu comportamento.

O modelo VFD é baseado no objeto VFD, que contém todos os objetos de comunicação, definidos explícita ou implicitamente.

O PROFIBUS armazena a descrição de seus objetos de comunicação em uma estrutura chamada OD (Object Dictionary). Um VFD contém exatamente um dicionário de objetos e esta associado a um único processo aplicativo. Na prática, um dispositivo real pode conter vários VFDs.

O objeto VFD é definido implicitamente, ou seja, é conhecido internamente pela camada de aplicação.

#### 3.2.5.4 Dicionário de objetos

Um processo aplicativo precisa antes conhecer a descrição de um objeto de comunicação de um aplicativo remoto para depois poder utilizá-lo. A descrição de todos os objetos de comunicação definidos explicitamente são armazenados no dicionário de objetos e são anunciados à rede durante a fase de configuração.

Cada estação possui, além de seu próprio dicionário de objetos (Source – OD), uma cópia total ou parcial do dicionário de objetos de seus parceiros de comunicação (Remote – OD). A FMS oferece serviços especiais de gerenciamento do dicionário de objetos para ler, adicionar e deletar entradas.

Os objetos de comunicação definidos no OD são acessados através de um endereço lógico chamado índice, que é uma palavra de 16 bits. Dessa forma, é possível aos dispositivos de diferentes partes, sistemas e fabricantes trocarem informações entre si sem terem que se preocupar com os detalhes dos atributos.

É importante estar atento para o fato de que todas essas estruturas foram estabelecidas de modo a fazer com que o máximo de informações pudessem ser definidas na fase de configuração da rede, otimizando a troca de dados.

#### 3.2.5.5 Serviços

Um serviço permite a um processo aplicativo utilizar os objetos de comunicação disponibilizados por um servidor remoto. São transmitidos ao parceiro de comunicação através dos chamados PDUs (“*Protocol Data Unit*”), que contém informações indicando a seqüência de tarefas que deve ser executada.

Os serviços podem ser classificados como serviços confirmados e serviços não confirmados. Serviços confirmados são somente permitidos nas relações de comunicação definidas como “*one-to-one*”, pois a cada pedido de serviço deve corresponder uma confirmação (através dos quadros de reconhecimento ou de resposta).

Serviços não confirmados são permitidos para todos os tipos de relação de comunicação: “*one-to-one*”, “*one-to-many*” (multicast) e “*one-to-all*” (broadcast).

De uma forma geral, os serviços do PROFIBUS podem ser divididos em 3 grupos funcionais :

- 1 – Serviços de aplicação;
- 2 – Serviços de administração;
- 3 – Serviços de gerenciamento da rede;

#### 3.2.5.6 Lista de relações de comunicação

É mais um dos conceitos introduzidos pelo PROFIBUS para simplificar o tráfego de informações na rede. A CRL (Communication Relationship List) contém a descrição de todas as relações de comunicação que uma estação pode ter, independente do seu tempo de uso.

Cada linha da CRL contém os parâmetros de comunicação que a estação trocara com cada um de seus parceiros, já definidos no momento de configuração da rede – diferentemente das redes de alta performance onde esses parâmetros são acordados no momento da abertura da conexão.

Utilizando a CRL, cada estação usa somente conexões pré-configuradas, simplificando muito os recursos de comunicação necessários, dada a grande diversidade de dispositivos de campo e diferença de recursos entre eles. Logicamente, a CRL pode ser dividida em duas partes : FMS-CRL e a LLI-CRL.

A FMS-CRL contém informações sobre cada conexão pertinentes a FMS e a LLI-CRL informações de mais baixo nível, referentes a LLI. A estrutura geral da CRL é composta por um cabeçalho que contém informações sobre o tamanho da CRL, uma parte estática contendo informações não configuráveis é carregada na inicialização da rede e soma-se a uma parte dinâmica, utilizada durante a operação.

### 3.2.5.7 Relações de comunicação

No PROFIBUS são adotados dois tipos de relação de comunicação, a relação orientada à conexão e a relação não orientada à comunicação.

- A) Relação Orientada para a Conexão : Em uma relação de comunicação orientada a conexão, uma conexão lógica entre duas estações deve existir antes que seja iniciada a troca de dados. Esse método pode ser caracterizado por 3 fases : estabelecimento da conexão, troca de dados e término da conexão.

Uma conexão pode ser caracterizada pelos serviços utilizados na LLI e por seu comportamento dentro da mesma. O PROFIBUS define os seguintes tipos de conexão :

1) Mestre-Escravo

- conexão para transferência cíclica sem iniciativa do escravo.
- conexão para transferência cíclica com iniciativa do escravo.
- conexão para transferência acíclica sem iniciativa do escravo.
- conexão para transferência acíclica com iniciativa do escravo.

2) Mestre-Mestre

- conexão para transferência acíclica.

Cada tipo de conexão é mais adequada para determinados processos aplicativos. As conexões mestre-escravo cíclicas são adequadas quando é preciso tempo de resposta reduzido, acesso freqüente a variáveis – preferencialmente uma variável por conexão, é quando as estações escravo devem ser acionadas em seqüência.

As conexões cíclicas são definidas somente para estações mestre-escravo. Se entre duas estações-mestre for necessário estabelecer uma relação cíclica, um deles deve simular o comportamento de um escravo.

Um caso especial do PROFIBUS é a chamada “iniciativa do escravo”. Este tipo de conexão permite que uma estação escravo, recebendo uma solicitação do mestre, utilize os serviços não confirmados da FMS.

Essa característica é utilizada principalmente quando a estação-escravo deve notificar ao mestre a ocorrência de eventos.

As conexões acíclicas são adequadas quando os dados da estação remota não são acessados com frequência, os tempos de resposta não são críticos e para a transmissão de dados para configuração ou diagnóstico.

Para o caso de conexões acíclicas com iniciativa do escravo, a estação-mestre deve constantemente acionar o escravo (Polling Request) para dar a este a possibilidade de transmitir seu PDU, se houver. As relações mestre-mestre são utilizadas em estações PROFIBUS complexas, onde há a necessidade de troca de dados, paralela e priorizada.

#### B) Relação não orientada para a conexão.

Nas relações de comunicação não orientadas à conexão, não existem as fases de estabelecimento e término de conexão, estando as estações sempre na fase de transferência de dados.

Nesse caso, uma estação-mestre pode enviar dados para várias (multicast) ou para todas (broadcast) as demais estações e são utilizadas para sincronização de processos, relógios e alarmes globais. Não são enviados respostas/reconhecimento para serviços de broadcast/multicast.

A CRL consiste em um cabeçalho, uma parte estática e uma parte dinâmica. O cabeçalho contém a estrutura da CRL, indicando o número de entradas, o intervalo de tempo para controle de abertura e fechamento de conexões e opcionalmente a LSAP da Poll List.

A parte estática contém os parâmetros definidos durante a configuração da estação e descreve os atributos estáticos de cada relação de comunicação definida. A parte dinâmica contém os atributos que são alterados conforme a situação da conexão. Nem todos os atributos são utilizados, de acordo com o tipo de comunicação.

### 3.3 Foundation Fieldbus versus Profibus

Nas seções anteriores procuramos dar uma visão das características desses dois protocolos normalizados pelo IEC/ISA e DIN.

Como podemos perceber, as soluções que cada órgão normalizador adota diferem-se em vários aspectos, principalmente na camada de enlace, tornando-se totalmente incompatíveis entre si.

Todos os órgãos normalizadores têm o mesmo objetivo que é construir um protocolo específico para aplicações em chão de fábricas com garantias de tempo de resposta, necessários muitas vezes para aplicações críticas em um ambiente fabril.

Diversos protocolos com essa filosofia estão disponíveis no mercado, mas qual será o padrão de fato que as indústrias vão adotar? Isso vai depender da confiabilidade e desempenho que cada protocolo apresentar.

Abaixo montamos uma tabela comparativa entre os dois protocolos abordados neste trabalho.

O objetivo é dar uma visão geral das características comuns e frisar principalmente as diferenças de implementação entre eles.

TABELA 3.4 – Características dos padrões Foundation Fieldbus e Profibus.

	<b>Fieldbus Foundation</b>	<b>Profibus</b>
Topologia	Barramento / Árvore de barramentos	Barramento /Árvore de barramentos
Meio físico	Par trançado Fibra ótica Rádio	Par trançado Fibra ótica
Número de endereços de rede por barramento	32	127
Menor taxa de transmissão	31,25 Kbit/s	9,6 Kbit/s
Maior taxa de transmissão	100 Mbit/s	12 Mbit/s
Comprimento máximo do barramento	1900 m	1200 m
Número máximo de repetidores	4	5
Configuração on-line	SIM	SIM
Intrinsic safe	SIM	SIM
Redundância no barramento	SIM	SIM
Alimentação no barramento	SIM	SIM
Método de acesso ao barramento	Token(LAS) Produtor-consumidor Atualização cíclica	Híbrido – (Token entre mestres e Mestre/Escravo)
Comunicação	Bi-direcional	Halfduplex
Permite troca a quente	SIM	SIM

## 4 Tempos de supervisão do PROFIBUS

Neste capítulo apresentaremos os tempos de supervisão envolvidos no processo de comunicação de uma rede PROFIBUS. Dentro das normas do PROFIBUS são definidos alguns tempos que serão utilizados como parâmetros nas operações da camada FDL. Também são encontradas algumas variáveis que têm a função de contadores e temporizadores.

Alguns desses tempos são utilizados na parametrização da rede, influenciando seu desempenho. Uma boa compreensão desses tempos ajudará a entender o seu funcionamento e controle em relação aos tempos de resposta da rede.

### 4.1 Tempo de rotação desejado

Para garantir uma comunicação em tempo real no PROFIBUS é preciso estabelecer um tempo de rotação desejado do token  $T_{TR}$  (Target Rotation Time).

Esse tempo tem que levar em conta o tempo de resposta requerido pela aplicação, o número de estações mestre e escravos participantes do anel lógico e o tempo gasto na transmissão de mensagens de alta e baixa prioridade, a ocorrência de erros e a manutenção da GAP.

Quando o PROFIBUS é parametrizado para garantir um tempo de resposta para uma aplicação é estabelecido um  $T_{TR}$ . Esse tempo deve ser o maior intervalo de tempo decorrido entre dois ciclos de mensagens de prioridade alta em uma estação-mestre, sob carga máxima no barramento. Desse modo, estamos estimando o pior caso tolerável para uma aplicação.

É definida também uma variável contendo o tempo real de rotação do token  $T_{RR}$  (Real Rotation Time). A medida desse tempo inicia no momento que a estação-mestre recebe o token e se encerra com o recebimento do próximo quadro de token do ciclo de mensagem seguinte. O intervalo de tempo calculado equivale ao  $T_{RR}$ .

Segundo [DIN 91] cada estação-mestre ativa, via de regra, sempre pode processar uma seqüência de mensagens de alta prioridade por recebimento de token, independente do seu tempo de rotação.

Essas mensagens de alta prioridade correspondem ao período acíclico de transmissão de cada estação-mestre, ou seja, esta processando algum pedido de informação feito pelo usuário na estação mestre que possui o token.

Para que as mensagens de baixa prioridade possam ser transmitidas e ainda ter tempo para suportar eventuais retransmissões, que são necessárias em caso de erro, bem como a manutenção da GAP, o tempo de rotação desejado  $T_{TR}$  deverá ser sempre maior que o tempo real de rotação  $T_{RR}$ .

As transmissões de mensagens de baixa prioridade e a manutenção da GAP só serão executadas se o tempo real de rotação  $T_{RR}$  for menor que o tempo de rotação desejado,  $T_{TR}$ . Quando isso não ocorrer, tais mensagens são transmitidas em tokens futuros.

Uma vez iniciada a transmissão de mensagem, estas não pode mais ser interrompida. Caso ocorra o encerramento do tempo de token, quer dizer que o  $T_{RR} \geq T_{TR}$ , ou seja, o prolongamento do tempo resultará no encurtamento do tempo de token nas próximas transmissões.

## 4.2 Tempos de Controle do PROFIBUS

Os tempos envolvidos na comunicação em uma rede PROFIBUS podem ser classificados como tempos medidos, configuráveis ou calculados. Como veremos mais adiante, na prática todos os tempos configuráveis podem ser alterados no mestre PROFIBUS, possibilitando configurar a rede conforme as necessidades de aplicação.

Faremos uma descrição de cada um deles e a partir da seção 4.2.4 descrevemos os tempos de ciclos que têm maior importância para o controle dos tempos de resposta. Também será apresentada uma estimativa de desempenho abrangendo esses tempos de controle.

### 4.2.1 Tempos medidos

1. **Tempo de bit ( $t_{BIT}$ )** – tempo decorrido durante a transmissão de um bit, sendo equivalente ao inverso da taxa de transmissão.
2. **Tempo de atraso na transmissão ( $T_{TD}$ )** – tempo necessário para o sinal percorrer o meio entre o transmissor e o receptor. O tempo de atraso dos repetidores devem ser levados em conta. Normalmente é definido como um número de Tbit.

$$T_{TD} = t_{TD} / t_{BIT} \quad (1)$$

### 4.2.2 Tempos configuráveis

- 1 **Tempo de Sincronização ( $T_{SYN}$ )** – mínimo intervalo de tempo durante o qual cada estação deve receber, do meio de transmissão, o estado inativo (binário 1), antes de aceitar o início de um quadro de ação (quadro de pedido de dados ou token).
- 2 **Tempo do Intervalo de Sincronização ( $T_{SYNI}$ )** – máximo intervalo de tempo permitido entre dois tempos de sincronização. É composto de dois ciclos de mensagens, onde cada quadro tem seu tamanho máximo e os respectivos tempos de sincronização.
- 3 **Tempo de atraso na estação ( $T_{SDX}$ )** – Período de tempo que pode ser transcorrido entre a recepção ou transmissão do último bit de um quadro e a recepção ou transmissão do último bit do quadro seguinte (com respeito ao meio de transmissão, isto é, incluindo receptor e transmissor). Tempo passado entre o envio do pedido e o recebimento do reconhecimento ou resposta a esse pedido.

São definidos os seguintes tempos de atraso na estação :

#### 3.1 Atraso do iniciador ( $T_{SDI}$ )

$$T_{SDI} = t_{SDI} / t_{BIT} \quad (2)$$

#### 3.2 Mínimo tempo de atraso dos respondedores (min $T_{SDR}$ )

$$\min T_{SDR} = \min t_{SDR} / t_{BIT} \quad (3)$$

3.3 Máximo tempo de atraso dos respondedores ( $\max T_{SDR}$ )

$$\max T_{SDR} = \max t_{SDR} / t_{BIT} \quad (4)$$

- 4 **Tempo de Silêncio ( $T_{QUI}$ )** – Durante o tempo de silêncio, a transmissão e a recepção são desabilitadas.

$$T_{QUI} < \min T_{SDR} \quad (5)$$

Para satisfazer a essa condição, pode ser necessário o aumento de  $T_{SDR}$ .

- 5 **Tempo de ready ( $T_{RDY}$ )** – tempo no qual uma estação-mestre deve estar pronta para o recebimento de um quadro de resposta ou reconhecimento, após ter feito o pedido do dado.

$$T_{RDY} < \min T_{SDR} \quad (6)$$

Novamente pode ser necessário o aumento de  $T_{SDR}$ .

$$T_{QUI} < T_{RDY} \quad (7)$$

Pode ser necessário o aumento de  $T_{RDY}$  e, conseqüentemente, de  $T_{SDR}$ .

- 6 **Tempo de reação ( $T_{SET}$ )** - tempo que se passa entre a ocorrência de um evento até a execução da reação necessária.

$$T_{SET} = t_{SET} / t_{BIT} \quad (8)$$

- 7 **Tempo de inatividade ( $T_{ID}$ )** – medido na linha do receptor, corresponde ao tempo decorrido entre o último bit de um quadro recebido e a detecção da transmissão do primeiro bit do quadro seguinte. Equivalente no mínimo, ao tempo de sincronização ( $T_{SYN}$ ) mais a margem de segurança ( $T_{SM}$ ).

Após quadros de resposta, reconhecimento ou token :

$$T_{ID1} = \max(T_{SYN} + \max T_{SM} + \max T_{SDI}) \quad (9)$$

Após quadros de ação sem reconhecimento tipo SDN :

$$T_{ID2} = \max(T_{SYN} + \max T_{SM} + \max T_{SDR}) \quad (10)$$

- 8 **Tempo de slot ( $T_{SL}$ )** – tempo máximo esperado pelo iniciador para o recebimento completo do primeiro quadro de resposta ou reconhecimento, contado após o envio do último bit do quadro de ação (Pedido).

É também o tempo esperado pelo iniciador para que o receptor do quadro, por ele transmitido, envie o primeiro quadro de resposta.

Nas estações-mestre monitora-se o tempo entre a transmissão de um quadro e o recebimento da resposta vinda do receptor. Se o tempo expirar e a resposta não for recebida, uma nova transmissão deve ser feita.

Após um quadro de ação



$$T_{SL1} = 2 * T_{TD} + \max T_{SDR} + 11 \text{ bit} + T_{SM} \quad (11)$$

Após um token quadro

$$T_{SL2} = 2 * T_{TD} + \max T_{ID1} + 11 \text{ bit} + T_{SM} \quad (12)$$

Para a simplificação, somente é utilizado um slot-time no sistema, sendo o maior. Uma vez que se trata de um puro tempo de supervisão, não existem desvantagens conseqüentes no tempo de reação do sistema. [DIN 91]

$$T_{SL} = \max (T_{SL1}, T_{SL2}) \quad (13)$$

#### 4.2.3 Tempos calculados

1. **Margem de segurança ( $T_{SM}$ )** – utilizado em alguns tempos configuráveis da rede, fornece uma margem de segurança. O tempo é definido como

$$T_{SM} = 2 \text{ bit} + 2 * T_{SET} + T_{QUI} \quad (14)$$

2. **Tempo time-out ( $T_{TO}$ )** – utilizado para a monitoração da atividade e inatividade das estações mestre e escravo.

$$T_{TO} = 6 * T_{SL} + 2 * n * T_{SL} \quad (15)$$

onde :

n = endereço para estações-mestre (0 a 126)

n = 130 para estações-escravo, independente desta.

3. **Tempo de atualização da GAP ( $T_{GUD}$ )** – indica à estação-mestre a necessidade de atualização da GAP, que é uma tabela contendo os endereços de todas as estações mestres e escravos situadas entre o seu próprio endereço e o endereço da estação-mestre sucessora no anel lógico.

A atualização é executada ciclicamente a cada passagem do token e sempre ocorre após a transmissão de todas as mensagens pendentes e quando houver tempo disponível. O tempo de manutenção é múltiplo do  $T_{TR}$  (Tempo Rotação Desejado)

$$T_{GUD} = G * T_{TR} \quad 1 \leq G \leq 100 \quad (16)$$

Alguns variáveis que tem a função de temporizadores e contadores que são definidos dentro das normas. São eles :

- a) ***token\_rotation\_timer*** – utilizado para controle do tempo de rotação do token, é usado para verificar se há tempo para transmitir mensagens de baixa prioridade e atualização da GAP.
- b) ***idle timer*** – monitora o estado de inatividade da estação e o tempo de sincronismo.
- c) ***slot timer*** – temporizador utilizado nas estações-mestre, tem como função monitorar o tempo entre a transmissão de um quadro de ação ou token e o recebimento da resposta vinda do receptor. Quando esgotar-se o tempo e não for recebida nenhuma resposta, uma nova transmissão deve ser feita.

- d) *time-out timer* - monitora a inatividade entre as estações-mestre e escravos; quando esse valor atinge zero é sinalizada a ocorrência de erros.
- e) *syn internal timer* – temporizador utilizado para monitorar o meio de transmissão em relação à sincronização do receptor. É responsável pela sinalização de erros que são reportados ao gerente da camada FMA 1/2.
- f) *gap update timer* – utilizado para indicar a necessidade de manutenção da GAPL, esse timer está presente somente nas estações-mestre.

Os contadores acima podem ser implementados opcionalmente; são utilizados para a manutenção ou a instalação da rede e são definidos aos pares.

Para as estações-mestre temos contadores para quadros transmitidos “*frame\_sent\_cout*” e para quadros retransmitidos “*Retry\_count*”.

Para estações-mestre e estações-escravo, temos contadores “*SD\_count*” que são utilizados para contar os quadros com o “*start delimiters*” válidos recebidos e os “*SD\_error\_count*” para contar os quadros com “*start delimiters*” inválidos recebidos.

#### 4.2.4 Tempo do ciclo de token

A organização do acesso ao barramento com várias estações ativas é definida como a carga básica em um barramento.

Como mostra a figura 4.1, o tempo do ciclo de token  $T_{TC}$  é composto pelo soma do tempo de propagação da mensagem do token  $T_{TF}$  mais o tempo de inatividade da estação  $T_{ID}$  mais o tempo de atraso na transmissão  $T_{TD}$ , conforme demonstrado na equação 17.

$$T_{TC} = T_{TF} + T_{ID} + T_{TD} \quad (17)$$

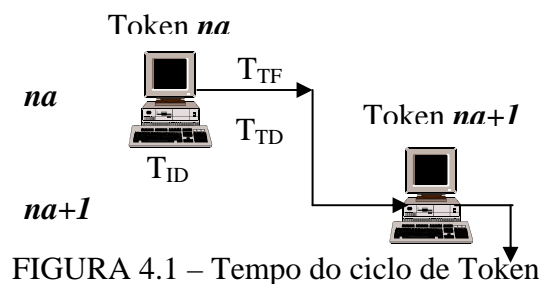


FIGURA 4.1 – Tempo do ciclo de Token

Como cada caracter possui 11 bits e o quadro do token, 3 octetos, o  $T_{TF}$  é 33 bits. Para uma taxa de 500 Kbit/s com um tempo de inatividade de 0,5 ms e desprezando o tempo de transmissão temos :

$$T_{TC} = (33 / 500 + 0,5), \quad T_{TC} = 0,566 \text{ ms}$$

O tempo gasto para organizar o acesso ao barramento, ou seja, a carga básica em um sistema com 30 mestres, seria de :

$$30 * T_{TC} = 16,98 \text{ ms.}$$

#### 4.2.5 Tempo de ciclo de mensagens

O tempo de um ciclo de mensagens é formado pelo tempo de pedido, pelo tempo de resposta e mais os tempos de controle.

Na figura 4.2 temos a descrição dos tempos envolvidos em um ciclo de mensagem. Para calcular o  $T_{MC}$ , devemos somar os  $T_{SDR}$ ,  $T_{ID}$  e  $T_{TD}$  já descritos anteriormente.

Os tempos de transmissão dos quadros  $T_{S/R}$  (tempo de pedido/envio) e  $T_{A/R}$  (tempo de reconhecimento/resposta) são determinados pela quantidade de caracteres dos quadros de pedido e quadros de resposta.

$$T_{S/R} = a * 11 \text{ bits}, \quad (18)$$

$$T_{A/R} = b * 11 \text{ bits}, \quad (19)$$

$$T_{MC} = T_{S/R} + T_{A/R} + T_{SDR} + 2 * T_{TD} + T_{ID} \quad (20)$$

onde a e b são as quantidades de caracteres dos quadros de pedido/envio e reconhecimento/resposta.

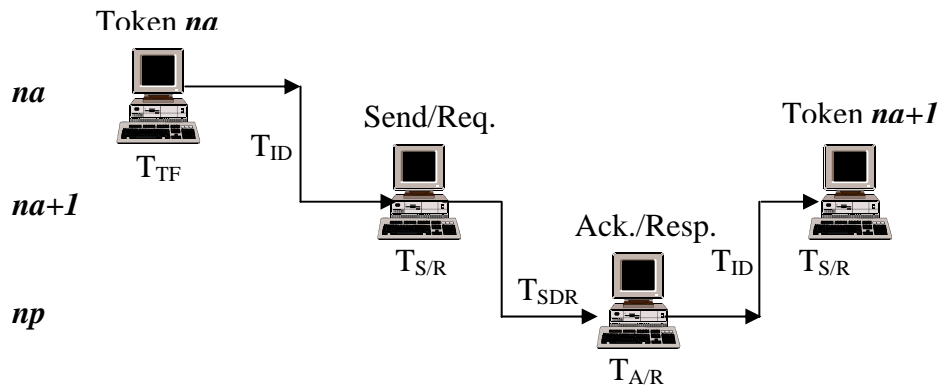


FIGURA 4.2 – Tempo de ciclo de mensagens

No exemplo abaixo será calculado um  $T_{MC}$  cujo o atraso na transmissão é desprezível. O tempo de inatividade  $T_{ID}$  e de atraso na estação  $T_{SDR}$  é de 0,5 ms; a taxa de transmissão é de 500 Kbit/s.

O tamanho da “*data\_unit*” do quadro de pedido é de 5 octetos e do quadro de resposta é de 10 octetos. Devem-se levar em consideração os campos de controle da estrutura da mensagem SD2, que são 9 octetos.

$$T_{MC} = ((5+9) * 11 + (10+9) * 11) / 500 + 0,5 + 0,5,$$

$$T_{MC} = 1,73 \text{ ms}$$

No segundo exemplo, levamos em consideração os valores máximos para os quadros de pedido e resposta que é de 246 octetos, mais os campos de controle da estrutura SD2. Os demais tempos e a taxa de transmissão serão mantidos conforme o exemplo anterior.

$$T_{MC} = ((246+9) * 11 + (246+9) * 11) / 500 + 0,5 + 0,5,$$

$$T_{MC} = 12,22 \text{ ms}$$

#### 4.2.6 Taxa de mensagem do sistema

A taxa de mensagem nada mais que é do que o número de ciclos possíveis por segundo.

$$R_{SYS} = 1 / (t_{MC}) \quad (21)$$

então para um  $T_{MC} = 1,73$ , temos :

$$R_{sys} = 578 \text{ mensagens por segundo}$$

#### 4.2.7 Tempo de reação do sistema

O tempo de reação do sistema  $T_{SR}$  em um sistema com um mestre e vários escravos é calculado a partir do tempo de ciclo de mensagens e da quantidade de estações-escravas levando em consideração a repetição de mensagens. [DIN 91]

$$T_{SR} = np * T_{MC} + mp * RET T_{MC} \quad (22)$$

onde :

$np$  = quantidade de estações-escravas

$mp$  = quantidade de ciclos de repetição de mensagem por passagem da lista de Poll

$RET T_{MC}$  = tempo de ciclo de repetição de mensagens.

O tempo máximo de reação de sistemas num sistema com várias estações-mestre e escravo, corresponde ao tempo de rotação desejado do token. [DIN 91]

$$T_{SR} = T_{TR}, \text{ onde } T_{TR} \geq \text{Max } T_{SR} \quad (23)$$

#### 4.2.8 Formulação do tempo de rotação desejado

Como já definimos anteriormente é necessário estabelecer um tempo de rotação desejado do token  $T_{TR}$  para garantir os requisitos de tempo da aplicação. A fórmula para calcular o tempo de rotação desejado é :

$$\min T_{TR} = nem * (T_{TC} + T_{MC} \text{ alta}) + K * T_{MC} \text{ baixa} + mt * RET T_{MC} \quad (24)$$

onde :

$nem$  = número de estações-mestre

$K$  = número estimado de ciclo de baixa prioridade por rotação do token

$mt$  = número de ciclos de retransmissão de mensagem por rotação do token

$T_{TC}$  = tempo do ciclo do token

$T_{MC}$  = tempo do ciclo da mensagem

$RET T_{MC}$  = tempo do ciclo de retransmissão de mensagem

No exemplo abaixo, levamos em consideração uma rede com 30 mestres, executando 1 mensagem de alta prioridade por mestre e 3 mensagens de baixa prioridade. Foi desprezada a retransmissão de mensagens.

$$\min T_{TR} = (30 * (0,57 + (1 * 1,73))) + 3 * 1,73 ; \quad \min T_{TR} = 74,2 \text{ ms}$$

#### 4.3 Formulação analítica do desempenho

As fórmulas apresentadas nas seções anteriores dão-nos a possibilidade de realizar algumas análises para obter uma referência sobre alguns tempos de controle do PROFIBUS.

Esses tempos são importantes porque podem ser utilizados para ajustar o desempenho da rede. Dentre as várias variáveis apresentadas anteriormente, a mais importante é determinar qual o tempo de rotação desejado o que, na realidade, é estimar o pior caso de atendimento dentro de uma rede PROFIBUS.

O ajuste fino de performance de uma rede PROFIBUS é determinar um  $T_{TR}$  muito próximo da aplicação que está sendo implantada. Estaremos levando em consideração nesta análise tamanhos fixos de quadros de pedido e resposta e o número de estações mestre e escravo.

Para os exemplos seguintes foram desconsiderados a repetição de mensagens e o tempo de atraso na transmissão ( $T_{TD}$ ) só para efeito de cálculo. Os tempos  $T_{ID1}$ ,  $T_{ID2}$ ,  $T_{SDR}$ , usados nos exemplos estão na tabela 4.1.

Esses valores foram extraídos da tabela 5.1 do capítulo 5. Dessa forma, estamos utilizando os valores reais de uma mestre na análise. Os quadros de resposta têm tamanho fixo de 100 octetos e os valores estão arredondados para duas casas decimais.

TABELA 4.1 – Tempos configuráveis versus taxas de transmissão

	<b>9,6</b>	<b>19,2</b>	<b>93,75</b>	<b>187,5</b>	<b>500</b>	<b>1500</b>	<b>3000</b>	<b>6000</b>	<b>12000</b>
	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>
<b><math>T_{SDR}</math></b>	2,000	2,000	2,000	0,320	0,200	0,100	0,083	0,075	0,067
<b><math>T_{ID1}</math></b>	1,500	1,500	1,500	0,197	0,074	0,025	0,015	0,010	0,006
<b><math>T_{ID2}</math></b>	2,000	2,000	2,000	0,320	0,200	0,100	0,083	0,075	0,067
<b><math>T_{TD}</math></b>	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000

\* Os valores estão expressos em milissegundos (ms).

#### 4.3.1 Estimativa para $N$ mestres

Para obtermos o tempo de rotação desejado de uma rede, primeiramente temos que calcular os tempos de ciclo de mensagens da rede que são calculados pela somatória dos tempos de pedido mais os tempos de resposta e os tempos de controle ( $T_{SDR} + 2*T_{TD} + T_{ID}$ ) que foram descrito na seção 4.2.5.

Na tabela 4.2 apresentamos os valores obtidos, variando o tamanho do quadro de pedido e fixando o quadro de resposta em 100 octetos.

TABELA 4.2 – Tempos dos ciclos de mensagens  $T_{MC}$

<b>Quadro Pedido (octetos)</b>	<b>9,6</b>	<b>19,2</b>	<b>93,75</b>	<b>187,5</b>	<b>500</b>	<b>1500</b>	<b>3000</b>	<b>6000</b>	<b>12000</b>
	<b>Kbit/s</b>	<b>Kbits/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>	<b>Kbit/s</b>
<b>5</b>	144,44	73,97	17,93	7,73	2,98	1,03	0,55	0,31	0,19
<b>100</b>	253,29	128,40	29,08	13,31	5,07	1,72	0,90	0,48	0,27
<b>150</b>	310,58	157,04	34,95	16,24	6,17	2,09	1,08	0,58	0,32
<b>200</b>	367,88	185,69	40,81	19,17	7,27	2,46	1,26	0,67	0,36
<b>246</b>	420,58	212,04	46,21	21,87	8,28	2,79	1,43	0,75	0,41

\* Valores de tempos expressos em milissegundos (ms).

Observando os valores obtidos na tabela acima, poderíamos ser induzidos a acreditar que quanto maior a taxa de transmissão, menor é a influência dos tempos de controle em relação ao tempo final obtido.

Um estudo mais detalhado sobre a influência dos tempos de controle em relação ao contexto geral da transmissão será apresentado na seção seguinte.

Na figura 4.3 traçamos a curva do ciclos de mensagem ( $T_{MC}$ ) versus taxa de transmissão. Os valores do gráfico foram extraídos da tabela 4.2.

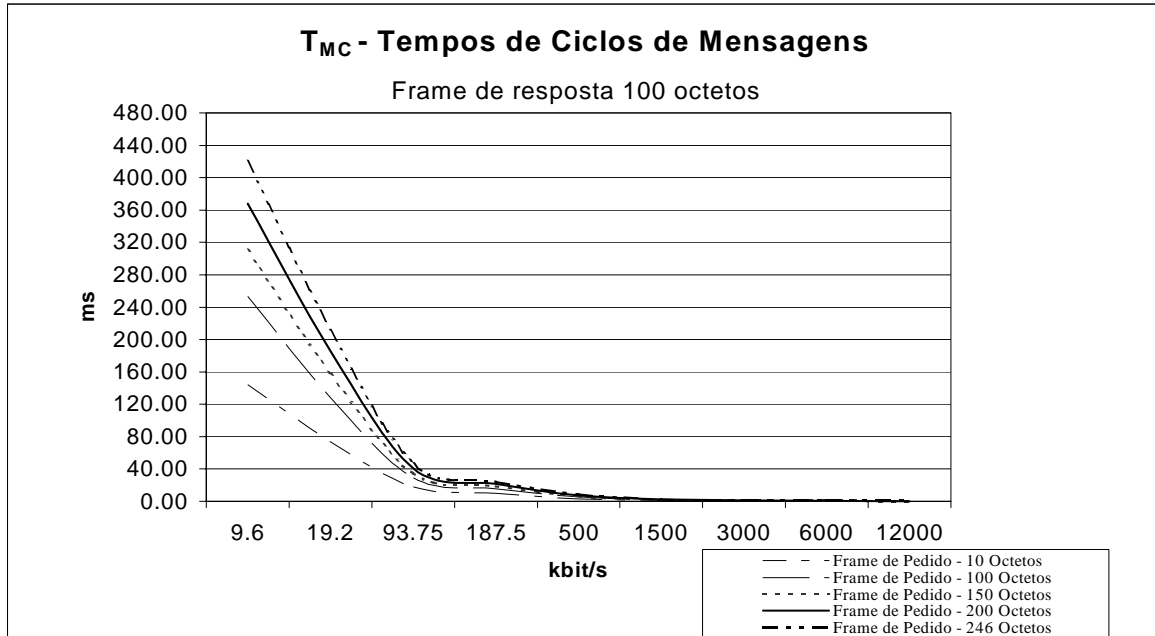


FIGURA 4.3 – Tempos dos ciclos de mensagens em uma rede PROFIBUS

Para obtermos os dados dos tempos desejados de rotação do token ( $T_{TR}$ ) é necessário acrescentarmos os parâmetros referentes ao polling, que é o mecanismo pelo qual cada mestre aciona seqüencialmente as estações-escravo para transferência e solicitação de dados.

A camada FDL oferece à camada 7 a possibilidade de escolha da prioridade com que se efetuará a transmissão. São definidos dois níveis de prioridade : alta ou baixa. Desse modo, sempre que uma estação ativa receber o token, ela transmitirá primeiro as mensagens de prioridade alta.

Após o termino da transmissão das mensagens de alta prioridade e ainda havendo tempo de posse do token, as mensagens de baixa prioridade serão enviadas.

Se esgotar-se o tempo de token, a transmissão das mensagens pendentes se dará no próximo ciclo.

Na análise abaixo, levamos em consideração que o mestre realiza 21 ciclos de alta prioridade; também inserimos 3 ciclos de retransmissão.

Isso significa que o polling de cada mestre está realizando 21 ciclos de transmissão com solicitação e recebimento de resposta e a cada 21 ciclos houve 3 ciclos de retransmissão. Na análise abaixo, os tempos de controle estão inseridos e foram descritos na seção 4.1.8.

Os resultados obtidos podem ser observados nas tabelas 4.3, 4.4, 4.5; os quadros de resposta são fixos (100 octetos) e os quadros de pedido foram respectivamente de 5, 100, 150 octetos.

Os valores dos tempos de controle estão na tabela 4.1 e os tempos de ciclo de mensagens foram extraídos da tabela 4.2.

TABELA 4.3 – Tempos do  $T_{TR}$  com o quadro de pedido de 5 octetos.

Estações Mestres	9,6 Kbit/s	19,2 Kbit/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
1	3471,44	1778,47	432,22	185,97	71,66	24,69	13,22	7,46	4,47
3	10414,31	5335,41	1296,66	557,92	214,98	74,06	39,66	22,37	13,40
6	20828,63	10670,81	2593,32	1115,84	429,96	148,12	79,31	44,73	26,80
9	31242,94	16006,22	3889,98	1673,76	644,94	222,18	118,97	67,10	40,20
12	41657,25	21341,63	5186,64	2231,68	859,92	296,24	158,62	89,46	53,60
14	48600,13	24898,56	6051,08	2603,63	1003,24	345,61	185,06	104,37	62,54
17	59014,44	30233,97	7347,74	3161,55	1218,22	419,67	224,71	126,74	75,94
20	69428,75	35569,38	8644,40	3719,47	1433,20	493,73	264,37	149,10	89,34
23	79843,06	40904,78	9941,06	4277,39	1648,18	567,79	304,02	171,47	102,74
26	90257,38	46240,19	11237,72	4835,31	1863,16	641,85	343,68	193,83	116,14

\* Valores de tempos expressos em milissegundos (ms).

TABELA 4.4 – Tempos do  $T_{TR}$  com o quadro de pedido de 100 octetos.

Estações Mestres	9,6 Kbit/s	19,2 Kbit/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
1	6083,94	3084,72	699,74	319,73	121,82	41,41	21,58	11,64	6,56
3	18251,81	9254,16	2099,22	959,20	365,46	124,22	64,74	34,91	19,67
6	36503,63	18508,31	4198,44	1918,40	730,92	248,44	129,47	69,81	39,34
9	54755,44	27762,47	6297,66	2877,60	1096,38	372,66	194,21	104,72	59,01
12	73007,25	37016,63	8396,88	3836,80	1461,84	496,88	258,94	139,62	78,68
14	85175,13	43186,06	9796,36	4476,27	1705,48	579,69	302,10	162,89	91,80
17	103426,94	52440,22	11895,58	5435,47	2070,94	703,91	366,83	197,80	111,47
20	121678,75	61694,38	13994,80	6394,67	2436,40	828,13	431,57	232,70	131,14
23	139930,56	70948,53	16094,02	7353,87	2801,86	952,35	496,30	267,61	150,81
26	158182,38	80202,69	18193,24	8313,07	3167,32	1076,57	561,04	302,51	170,48

\* Valores de tempos expressos em milissegundos (ms).

TABELA 4.5 – Tempos do  $T_{TR}$  com o quadro de pedido de 150 octetos.

Estações Mestres	9,6 Kbit/s	19,2 Kbit/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
1	7458,94	3772,22	840,54	390,13	148,22	50,21	25,98	13,84	7,66
3	22376,81	11316,66	2521,62	1170,40	444,66	150,62	77,94	41,51	22,97
6	44753,63	22633,31	5043,24	2340,80	889,32	301,24	155,87	83,01	45,94
9	67130,44	33949,97	7564,86	3511,20	1333,98	451,86	233,81	124,52	68,91
12	89507,25	45266,63	10086,48	4681,60	1778,64	602,48	311,74	166,02	91,88
14	104425,13	52811,06	11767,56	5461,87	2075,08	702,89	363,70	193,69	107,20
17	126801,94	64127,72	14289,18	6632,27	2519,74	853,51	441,63	235,20	130,17
20	149178,75	75444,38	16810,80	7802,67	2964,40	1004,13	519,57	276,70	153,14
23	171555,56	86761,03	19332,42	8973,07	3409,06	1154,75	597,50	318,21	176,11
26	193932,38	98077,69	21854,04	10143,47	3853,72	1305,37	675,44	359,71	199,08

\* Valores de tempos expressos em milissegundos (ms).

### 4.3.2 Eficiência da rede PROFIBUS

Para podermos compreender melhor os valores apresentados nas tabelas anteriores, primeiramente temos que verificar a influência dos tempos de controle que são configuráveis em relação aos tempos de transmissão dos dados na rede.

Para realizarmos essa análise calculamos a eficiência da rede, que é dada pela fórmula :

$$Ef = \frac{\text{TemposdeDados}}{\text{TemposdeDados} + \text{TempodeControle}} \quad (25)$$

Podemos constatar, como mostra na tabela 4.6 que as taxas mais altas não são as mais eficientes.

Se observarmos os valores da tabela 4.1, para cada taxa de transmissão temos valores diferenciados. Isto significa que por mais que se tenha tempos de atraso na estação e tempos de inatividade que diminuam conforme aumenta a taxa de transmissão no contexto geral da transmissão, esses tempos, por menores que sejam interferem de forma significativa na eficiência, devido ao fato de que em altas taxas temos uma vazão grande de dados.

Um outro fator importante é o tamanho médio das mensagens que circulam pela rede. Em aplicações que tenham pacotes pequenos em alta taxas, a rede atinge baixos índices de eficiência.

TABELA 4.6 – Eficiência da rede

Tamanho Frames Ped/Resp	9,6 Kbit/s	19,2 Kbit/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
1 / 100	97,46%	95,04%	79,69%	93,01%	90,44%	87,42%	81,48%	72,01%	59,85%
5 / 100	97,54%	95,19%	80,22%	93,22%	90,72%	87,78%	81,97%	72,67%	60,64%
100 / 100	98,59%	97,23%	87,78%	96,06%	94,54%	92,71%	88,95%	82,49%	73,18%
150 / 100	98,85%	97,73%	89,83%	96,77%	95,51%	93,99%	90,83%	85,27%	77,04%
200 / 100	99,03%	98,08%	91,29%	97,26%	96,19%	94,89%	92,15%	87,29%	79,92%
246 / 100	99,15%	98,32%	92,30%	97,60%	96,66%	95,50%	93,08%	88,72%	82,00%

Nas tabelas 4.7 e 4.8 demonstramos dois extremos em relação aos tamanhos dos quadros de pedido/resposta.

Na tabela 4.7 calculamos a eficiência para de um grupo de quadros com tamanhos inicialmente pequenos e na tabela 4.8, com um outro grupo de quadros com tamanhos grandes.

TABELA 4.7 – Eficiência da rede com variação dos tamanhos de quadros

Tamanho Frames Ped/Resp	9,6 Kbit/s	19,2 Kbit/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
1 / 10	90,35%	82,41%	48,96%	76,48%	69,81%	62,95%	51,82%	38,61%	26,71%
5 / 10	91,42%	84,20%	52,18%	78,71%	72,45%	65,90%	55,02%	41,70%	29,30%
100 / 10	97,63%	95,37%	80,84%	93,47%	91,05%	88,20%	82,55%	73,45%	61,58%
150 / 10	98,28%	96,63%	85,44%	95,21%	93,40%	91,22%	86,80%	79,37%	69,03%
200 / 10	98,66%	97,35%	88,25%	96,22%	94,77%	93,01%	89,39%	83,13%	74,06%
246 / 10	98,88%	97,78%	90,03%	96,84%	95,61%	94,12%	91,01%	85,55%	77,43%



TABELA 4.8 – Eficiência da rede com variação dos tamanhos de quadros

Tamanho Frames Ped/Resp	9,6 Kbit/s	19,2 Kbit/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
<b>1 / 246</b>	98,84%	97,71%	89,72%	96,73%	95,46%	93,93%	90,73%	85,13%	76,84%
<b>5 / 246</b>	98,86%	97,74%	89,86%	96,78%	95,53%	94,01%	90,86%	85,32%	77,10%
<b>100 / 246</b>	99,15%	98,32%	92,30%	97,60%	96,66%	95,50%	93,08%	88,72%	82,00%
<b>150 / 246</b>	99,25%	98,52%	93,17%	97,88%	97,05%	96,03%	93,86%	89,94%	83,82%
<b>200 / 246</b>	99,33%	98,68%	93,86%	98,11%	97,36%	96,44%	94,49%	90,93%	85,31%
<b>246 / 246</b>	99,39%	98,80%	94,38%	98,27%	97,59%	96,75%	94,96%	91,68%	86,45%

Podemos constatar que no pior caso (246 de pedido e 246 de resposta) em relação aos quadros de pedido/resposta com a taxa de transmissão mais alta (12 Mbit/s), a eficiência da rede ficou em 86,45%.

Na tabela abaixo, vamos analisar o pior caso possível em relação ao  $T_{TR}$ . Para isso estimamos que tanto os quadros de pedido como os quadros de resposta tenham o seu tamanho máximo de 246 octetos de dados, mais os 9 octetos de controle.

Um novo  $T_{MC}$  deve ser calculado para obter o  $T_{TR}$ . A necessidade deste novo cálculo do  $T_{MC}$  é devido ao fato de que o tamanho dos quadros de pedido resposta foram alterados.

Abaixo apresentamos os valores da carga máxima em relação à rotação do token.

TABELA 4.9 – Tempos do  $T_{TR}$  com o carga máxima na rede 246 octetos de dados

Estações Mestres	9,6 Kbit/s	19,2 Kbit/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
<b>1</b>	14113,94	7099,72	1522,01	730,87	276,00	92,80	47,27	24,48	12,98
<b>3</b>	42341,81	21299,16	4566,04	2192,61	827,99	278,40	141,82	73,45	38,94
<b>6</b>	84683,63	42598,31	9132,07	4385,22	1655,98	556,79	283,65	146,90	77,89
<b>9</b>	127025,44	63897,47	13698,11	6577,82	2483,96	835,19	425,47	220,35	116,83
<b>12</b>	169367,25	85196,63	18264,14	8770,43	3311,95	1113,58	567,29	293,80	155,77
<b>14</b>	197595,13	99396,06	21308,17	10232,17	3863,94	1299,18	661,84	342,76	181,73
<b>17</b>	239936,94	120695,22	25874,20	12424,78	4691,93	1577,58	803,66	416,21	220,68
<b>20</b>	282278,75	141994,38	30440,24	14617,39	5519,92	1855,97	945,49	489,66	259,62
<b>23</b>	324620,56	163293,53	35006,28	16809,99	6347,91	2134,37	1087,31	563,11	298,56
<b>26</b>	366962,38	184592,69	39572,31	19002,60	7175,90	2412,77	1229,13	636,56	337,51

\* Valores de tempos expressos em milissegundos (ms).

Na figura 4.4 traçamos o gráfico em escala logarítmica do tempo de rotação desejado, através dos dados obtidos na tabela 4.9.

Através das fórmulas extraídas das normas do PROFIBUS podemos ter uma idéia da forma como o protocolo se comporta e os caminhos necessários para sua parametrização.

As várias tabelas e gráficos apresentados nesta seção dão-nos uma noção em relação ao seu desempenho.

As estimativas aqui descritas levam em consideração uma rede com até 26 mestres e 24 escravos. Na seção seguinte apresentaremos uma estimativa para uma rede mestre-escravo, ou seja, apenas um mestre e um escravos.

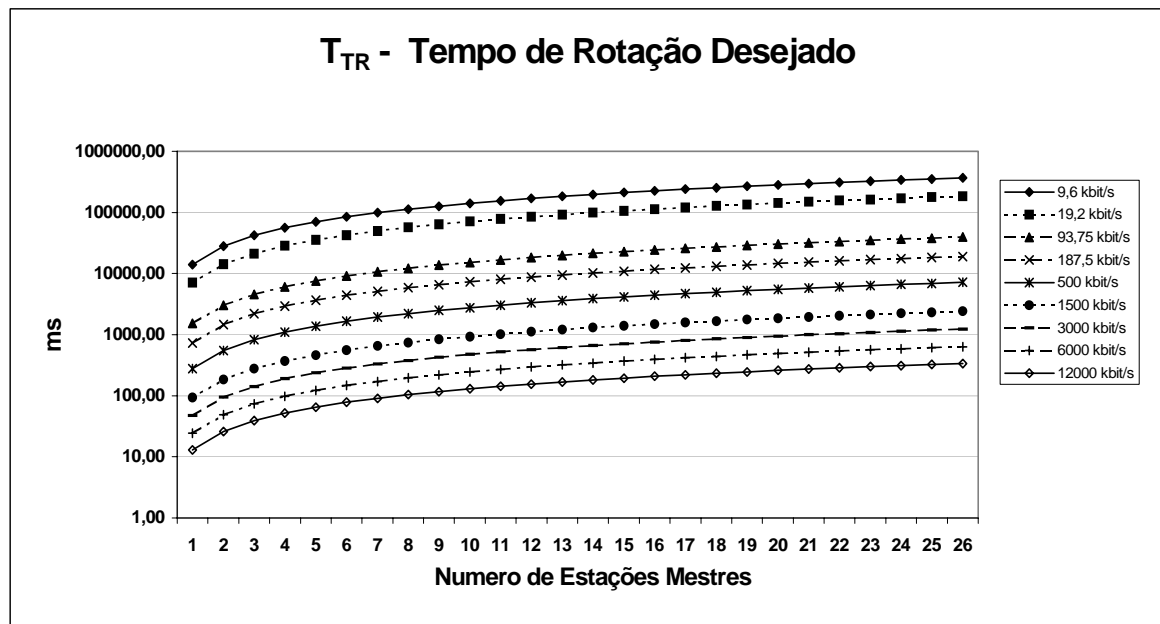


FIGURA 4.4 – Tempo de rotação desejado com carga máxima na rede

### 4.3.3 Estimativa para uma rede mestre-escravo puro

Uma rede mestre-escravo puro é constituído por apenas um mestre e um escravo. Nesta análise fixamos o tamanho do quadro de resposta em 70 octetos.

Devido a este fato, é necessário calcularmos um novo  $T_{MC}$  que é o tempo de ciclo de mensagem para esses novos tamanhos de quadros.

Os parâmetros usados nos cálculos estão disponíveis na tabela 4.1.

TABELA 4.10 –  $T_{MC}$  para uma rede mestre-escravo puro

Quadro Pedido (octetos)	9,6 Kbit/s	19,2 Kbits/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
10	115,79	59,65	15,00	6,27	2,43	0,84	0,46	0,26	0,16
100	218,92	111,21	25,56	11,55	4,41	1,50	0,79	0,43	0,25
150	276,21	139,85	31,43	14,48	5,51	1,87	0,97	0,52	0,29
200	333,50	168,50	37,29	17,41	6,61	2,24	1,15	0,61	0,34
246	386,21	194,85	42,69	20,11	7,62	2,57	1,32	0,70	0,38

\* Valores de tempos expressos em milissegundos (ms)

Nas figuras 4.5, traçamos a curva do  $T_{MC}$  versus taxa de transmissão com escala logarítmica.

Na figura 4.6 apresentamos o gráfico do  $R_{SYS}$ , que é o total de mensagens por segundo versus taxa de transmissão.

A carga máxima da rede com a maior taxa de transmissão para as mensagens de pedido e resposta informados anteriormente dá-nos um total 2637 mensagens por segundo.

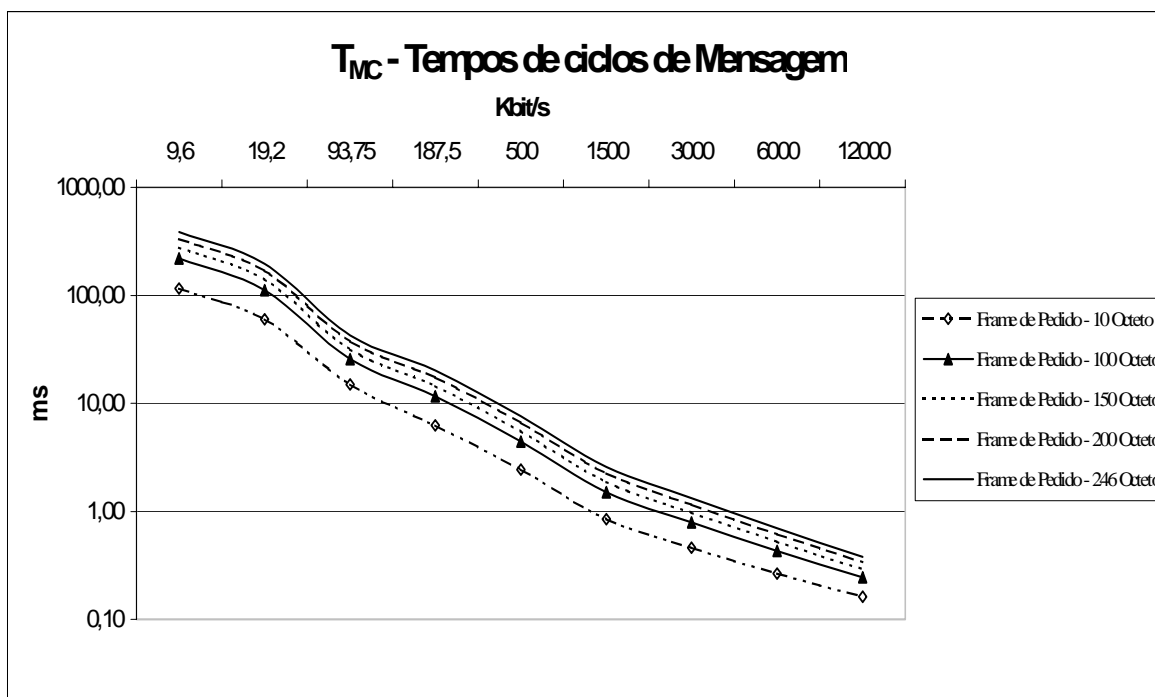


FIGURA 4.5 – Ciclo de mensagens em uma rede mestre-escravo puro

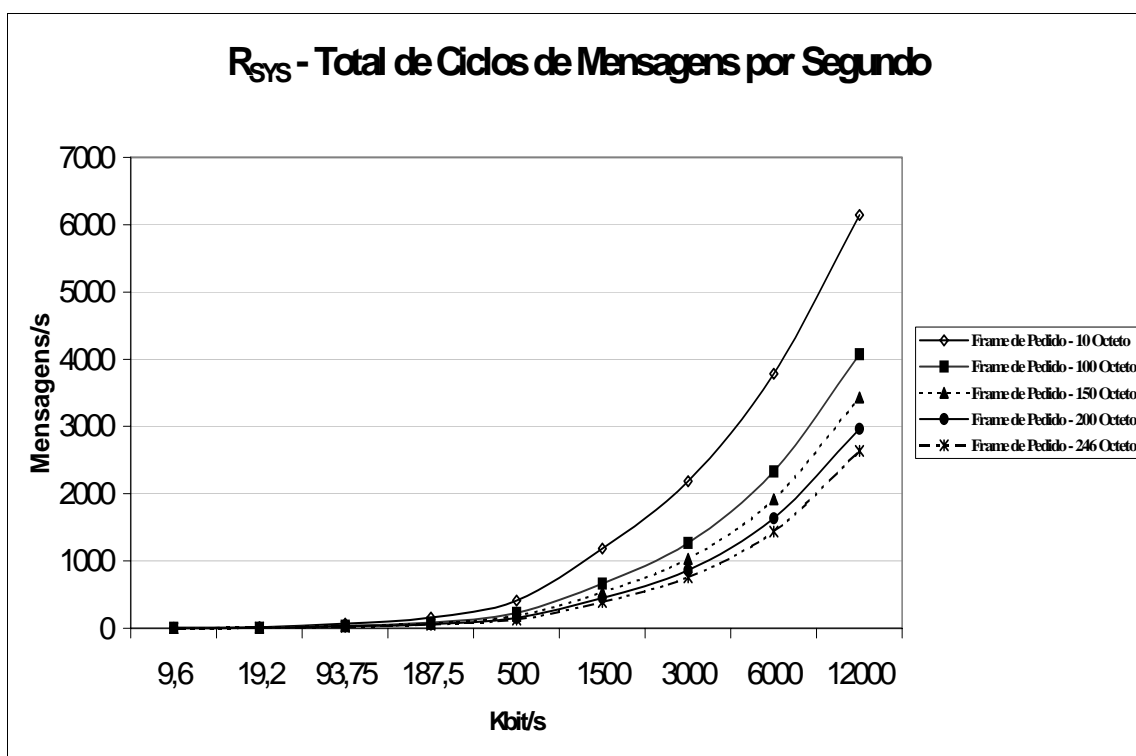


FIGURA 4.6 – Ciclo de mensagens em uma rede mestre-escravo

O tempo de reação do sistema  $T_{SR}$  em um sistema com um mestre e vários escravos é calculado a partir do tempo de ciclo de mensagens multiplicado pela quantidade de estações escravas levando em consideração a repetição de mensagens.

Na demonstração a seguir foram desprezadas as tentativas de retransmissão e foi estimado em 20 o número de escravos.

TABELA 4.11 –  $T_{SR}$  para uma rede mestre-escravo.

Quadro Pedido (octetos)	9,6 Kbit/s	19,2 Kbits/s	93,75 Kbit/s	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
10	2315,83	1192,92	299,97	125,33	48,60	16,87	9,16	5,28	3,26
100	4378,33	2224,17	511,17	230,93	88,20	30,07	15,76	8,58	4,91
150	5524,17	2797,08	628,51	289,60	110,20	37,40	19,43	10,42	5,82
200	6670,00	3370,00	745,84	348,27	132,20	44,73	23,09	12,25	6,74
246	7724,17	3897,08	853,79	402,24	152,44	51,48	26,47	13,94	7,58

\* Valores de tempos expressos em milissegundos (ms).

Na figura 4.7 traçamos as curvas do tempo de reação do sistema versus a taxa de transmissão. Os valores do gráfico foram extraídos da tabela 4.11.

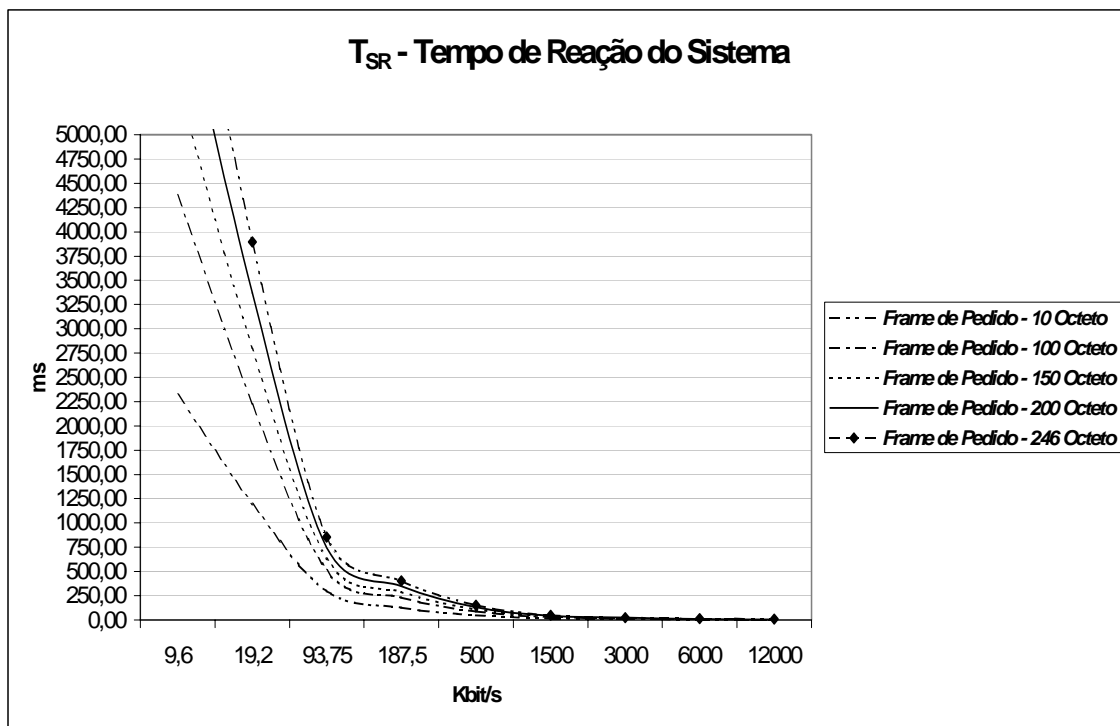


FIGURA 4.7 – Tempo de reação do sistema

Neste capítulo apresentou-se uma síntese dos tempos de controle que são definidos nas operações da camada FDL. Esses tempos são utilizados como parâmetros na configuração de uma rede PROFIBUS.

O principal propósito deste capítulo é demonstrar analiticamente o significado de cada variável descrita na norma e apresentar os resultados de forma a facilitar o entendimento.

Os gráficos e as tabelas apresentadas neste capítulo dão-nos uma percepção analítica do comportamento da rede em relação a seu desempenho e eficiência. O ponto crucial de todos os dados apresentados neste capítulo são os tamanhos médios dos quadros de pedido e resposta. Conhecendo os tamanhos médios dos quadros que circulam pela rede, uma avaliação analítica de desempenho seria facilmente demonstrada.

## 5 Procedimento para medida de tempo em uma rede Profibus.

Para aplicações em projetos de redes industriais, 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 que sejam feitas as configurações da comunicação na rede de acordo com os objetivos desejados.

Com isso, percebe-se que uma adequada utilização de uma rede FIELDBUS numa aplicação de 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. [PAC 98]

Conhecendo-se tal questão definiu-se um procedimento para medida de tempo em uma rede PROFIBUS, constituída por um mestre e três escravos, no qual utiliza-se um aplicativo que captura a solicitação do mestre e as respostas dos escravos.

Esse software também permite que se simulem solicitações do mestre para o escravo e a resposta do escravo para o mestre definindo o tamanhos dos quadros de pedido e resposta.

Também é possível determinar a taxa de transmissão da rede. Pode-se igualmente configurar os tempos de controle da rede.

O programa em questão é proprietário, não permitindo qualquer tipo de alteração a não ser aquelas que são parametrizadas.

Os parâmetros do mestre usado nos ensaios estão disponíveis na tabela 5.1 e as características de cada ensaio estão disponíveis na tabela 5.2.

Todos os parâmetros apresentados na tabela 5.1 são configuráveis pela aplicação.

TABELA 5.1 – Parâmetros usados nos experimentos

Parâmetro	Unidade	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
<b>T<sub>SLOT</sub></b>	ms	0,53	0,40	0,20	0,13	0,10	0,08
<b>MAX T<sub>SDR</sub></b>	ms	0,32	0,20	0,10	0,08	0,08	0,07
<b>MIN T<sub>SDR</sub></b>	ms	0,059	0,022	0,007	0,004	0,002	0,001
<b>T<sub>SET</sub></b>	ms	0,005	0,002	0,001	0,001	0,001	0,001
<b>T<sub>QUI</sub></b>	ms	0,000	0,000	0,000	0,001	0,001	0,001
<b>GAP Factor</b>		10,00	10,00	10,00	10,00	10,00	10,00
<b>Retry Limit</b>		1,00	1,00	1,00	2,00	3,00	4,00
<b>T<sub>RDY</sub></b>	ms	0,059	0,022	0,007	0,004	0,002	0,001
<b>T<sub>ID 1</sub></b>	ms	0,20	0,07	0,02	0,02	0,01	0,01
<b>T<sub>ID 2</sub></b>	ms	0,32	0,20	0,10	0,08	0,08	0,07
<b>T<sub>TR</sub></b>	ms	266,40	101,60	34,90	19,50	9,90	5,20
<b>Watchdog</b>	ms	333,00	127,00	43,60	25,80	15,00	10,40

Uma rede PROFIBUS pode ser configurada sob diversas formas diferentes como, por exemplo, uma rede mestre-escravo puro, que consiste de um único mestre com um ou mais escravos ou uma rede com vários mestre e escravos. Observar o desempenho em diversas situações permite uma análise mais abrangente do comportamento. Dessa forma realizamos três grupos experimentais.

TABELA 5.2 – Características dos experimentos

Experimento	Tipo de Rede	Taxa Kbits/s	Quadro Pedido	Quadro Resposta Escravo 1	Quadro Pedido	Quadro Resposta Escravo 2	Quadro Pedido	Quadro Resposta Escravo 3	Quadros capturados
1	1 M – 1 E	187,5	10	12	----	----	----	----	5617
2	1 M – 1 E	187,5	96	12	----	----	----	----	2451
3	1 M – 1 E	187,5	120	12	----	----	----	----	2016
4	1 M – 1 E	187,5	120	48	----	----	----	----	1705
5	1 M – 1 E	187,5	120	120	----	----	----	----	1347
6	1 M – 1 E	500	120	120	----	----	----	----	3347
7	1 M – 1 E	1500	120	120	----	----	----	----	5150
8	1 M – 1 E	3000	120	120	----	----	----	----	7680
9	1 M – 1 E	6000	120	120	----	----	----	----	806
10	1 M – 1 E	12000	120	120	----	----	----	----	984
11	1 M – 3 E	187,5	120	120	32	32	1	3	1524
12	1 M – 3 E	500	120	120	32	32	1	3	2414
13	1 M – 3 E	1500	120	120	32	32	1	3	6747
14	1 M – 3 E	3000	120	120	32	32	1	3	2384
15	1 M – 3 E	6000	120	120	32	32	1	3	620
16	1 M – 3 E	12000	120	120	32	32	1	3	921
17	1 M – 3 E	12000	2	4	2	4	1	3	1162
18	1 M – 3 E	6000	2	4	2	4	1	3	1100
19	1 M – 3 E	3000	2	4	2	4	1	3	21047
20	1 M – 3 E	1500	2	4	2	4	1	3	8092
21	1 M – 3 E	500	2	4	2	4	1	3	7310
22	1 M – 3 E	187,5	2	4	2	4	1	3	4462
23	2 M – 2 E	1500	64	64	120	120	----	----	4126
24	2 M – 2 E	500	64	64	120	120	----	----	5565
25	2 M – 2 E	187,5	64	64	120	120	----	----	3734

\* M = Mestre, E = Escravo

Cada grupo tem uma característica única em relação ao tipo da rede, a idéia é avaliar a influência no aumento do números de estações, nos tamanhos de quadros e na taxa de transmissão.

A idéia é gradativamente observar o comportamento à medida que o número de mestres e escravos crescem na rede.

Nas seções seguintes descreveremos com mais detalhes as características de cada grupo experimental.

### 5.1 Primeiro grupo experimental

Dentro do primeiro grupo foram realizados 10 experimentos nos quais, variamos o tamanho dos quadros de pedido e resposta.

O objetivo é verificar o comportamento tanto do mestre quanto do escravo em relação ao tamanho dos quadros, observando se que nos experimentos de 1 a 5 se os tamanhos influenciam nos seus tempos de processamento.

No segundo momento nos experimentos de 6 a 10 fixou-se o tamanho dos quadros e variamos a taxa de transmissão, a fim de observar o comportamento da rede em relação aos parâmetros dispostos na tabela 5.1, que variam conforme a taxa de transmissão.

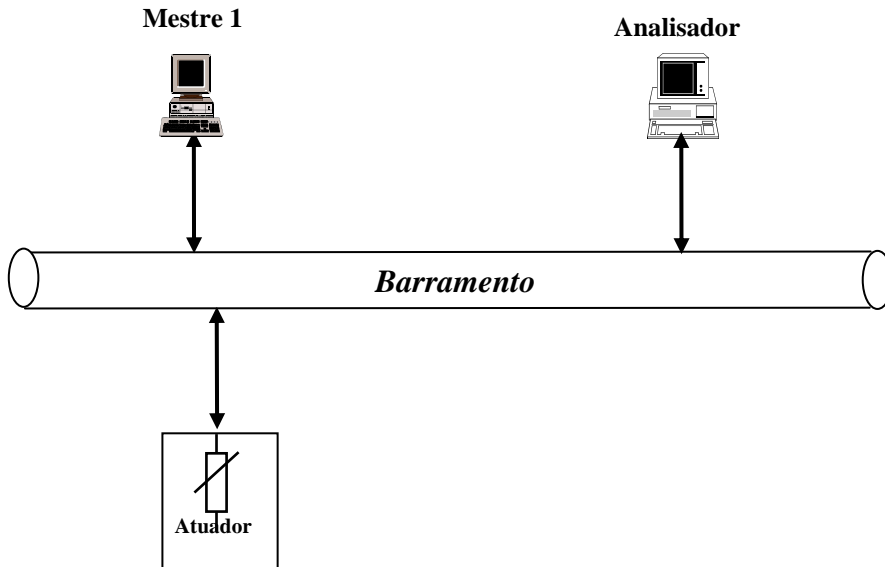


FIGURA 5.1 – Primeiro modelo experimental, 1 mestre e um 1 escravo

## 5.2 Segundo grupo experimental

No segundo grupo realizou-se 12 experimentos, uma quantidade maior de escravos foram inseridos para observamos melhor o controle do polling do mestre.

Na primeira fase nos experimentos de 11 a 16 realizamos uma bateria de experimentos fixando para cada escravo um tamanho de quadro e variando a taxa de transmissão.

Na segunda fase nos experimentos de 17 a 22 definimos diferentes tamanhos de quadros de pedido e resposta para dois escravos e mantivemos no terceiro escravo o mesmo tamanho usado anteriormente.

O objetivo é determinar se os tamanhos dos quadros influenciam o processamento tanto do mestre como dos escravos. Sendo assim, podemos verificar se o polling do mestre também é influenciado em relação a tamanhos variados dos quadros de pedido e resposta.

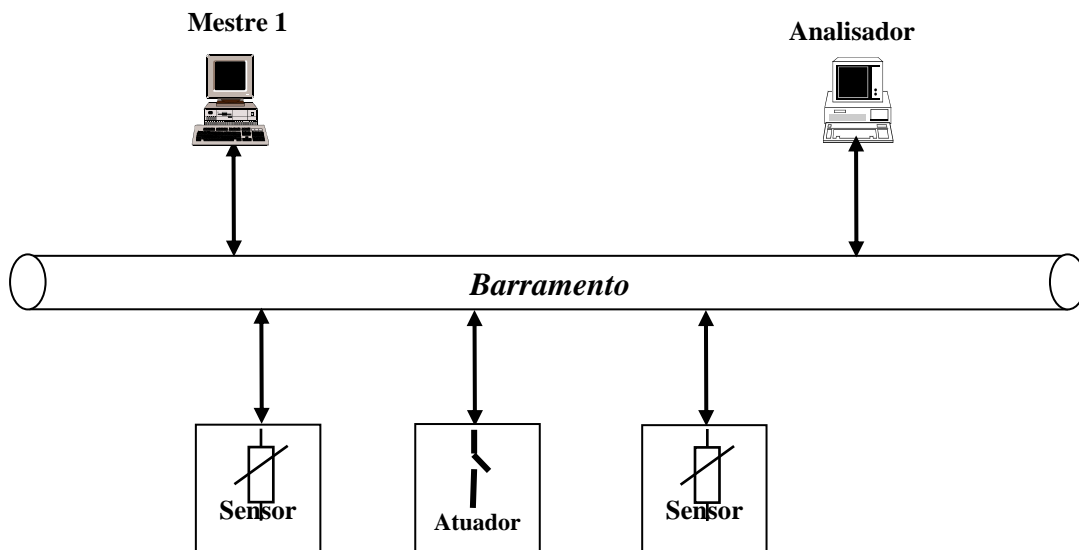


FIGURA 5.2 – Segundo modelo experimental, 1 mestre e 3 escravos

### 5.3 Terceiro grupo experimental

No terceiro experimento de 23 a 25, o objetivo era realizar medidas com dois mestres e dois escravos. Tínhamos à disposição um mestre da Siemens e um mestre da Altus que estava em desenvolvimento.

Realizamos três experimentos e constatamos que o mestre da Altus não realizava seus ciclos de transmissão, simplesmente devolvendo o token quando o recebia. Dessa forma ficamos impossibilitados de realizar esse terceiro grupo experimental.

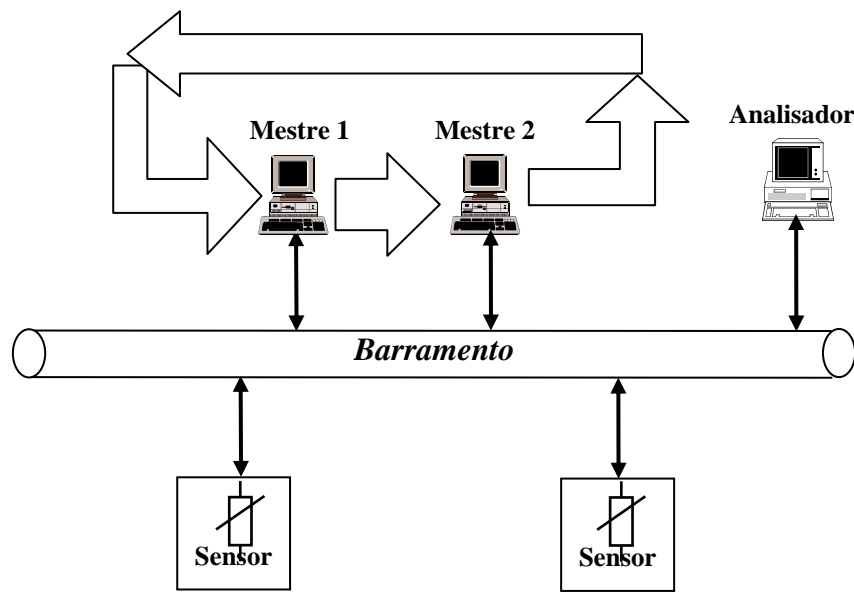


FIGURA 5.3 – Terceiro modelo experimental, 2 mestres e 2 escravos

### 5.4 Análise dos dados obtidos na monitoração de uma rede PROFIBUS

Como a monitoração capturou vários ciclos de transmissão entre mestres e escravos não temos condições de apresentar todos os valores .

Dessa forma colocamos 8 ciclos escolhidos de forma aleatória do conjunto de amostras em cada tabela para ser usado nas análises seguintes.

Para tirarmos as conclusões necessárias dos valores obtidos nos experimentos, vamos realizar um análise usando uma linha de tempo na qual podemos perceber a ocorrência dos eventos dentro da comunicação entre o mestre e o escravo.

Para melhor compreensão dos valores que serão apresentados nas tabelas, primeiramente apresentaremos a descrição de cada coluna da tabela usada na análise. Essa descrição está disponível na tabela 5.3.

Os valores da coluna  $T_{\text{InícioPed}}$ ,  $T_{\text{InícioResp}}$ ,  $T_{\text{IniSta}}$ ,  $T_{\text{IniTok}}$  e  $T_{\text{ProxPed}}$  foram extraídos dos logs gerados pelo analisador de rede.

O analisador marca o tempo inicial a partir do momento em que o primeiro bit é transmitido no barramento.

Os demais tempos foram calculados usando-se como referência os tempos extraídos do analisador.



TABELA 5.3 – Descrição dos dados do experimento

$T_{\text{InícioPed}}$	Tempo no qual se inicia a transmissão do pedido
$T_{\text{Pedido}}$	Tempo final da transmissão do quadro de pedido. Esse valor é calculado somando o $T_{\text{InícioPed}}$ com o tempo necessário para a transmissão do pedido
$T_{\text{InatEsc}}$	Tempo de inatividade do escravo, que é calculado subtraindo $T_{\text{InícioResp}}$ menos $T_{\text{Pedido}}$
$T_{\text{InícioResp}}$	Tempo no qual se inicia a transmissão da resposta
$T_{\text{Resposta}}$	Tempo final da transmissão do quadro de resposta. Esse valor é calculado somando $T_{\text{InícioResp}}$ com o tempo necessário para a transmissão da resposta
$T_{\text{InatMes}}$	Tempo de inatividade do mestre, que é calculado subtraindo $T_{\text{IniSta}}$ menos $T_{\text{Resposta}}$
$T_{\text{IniSta}}$	Tempo no qual se iniciou a transmissão da solicitação do Status
$T$	$T_{\text{IniTok}}$ menos $T_{\text{IniSta}}$
$T_{\text{IniTok}}$	Tempo no qual se iniciou a transmissão do token
$T_{\text{TranTok}}$	Tempo de transmissão do token
$T_{\text{ProxPed}}$	Tempo no qual se iniciou a transmissão do próximo pedido
$T_{\text{RR}}$	Tempo real de rotação do token : $T_{\text{ProxPed}}$ menos $T_{\text{InícioPed}}$

#### 5.4.1 Análise dos experimentos com um mestre e um escravo

Na tabela 5.4 apresentamos os valores extraídos do primeiro experimento. Os quadros de pedido e resposta são, respectivamente, de 10 e 12 octetos e taxa de transmissão é de 187,5 Kbit/s.

TABELA 5.4 – Experimento 1, pedido 10, resposta 12 octetos, 187,5 Kbits/s

$T_{\text{InícioPed}}$	$T_{\text{Pedido}}$	$T_{\text{InatEsc}}$	$T_{\text{InícioResp}}$	$T_{\text{Resposta}}$	$T_{\text{InatMes}}$	$T_{\text{IniSta}}$	$T$	$T_{\text{IniTok}}$	$T_{\text{TranTok}}$	$T_{\text{InatMes}}$	$T_{\text{ProxPed}}$	$T_{\text{RR}}$
8,96	10,08	0,06	10,14	11,37	0,22	11,59	0,89	12,47	0,18	0,22	12,87	3,90
12,87	13,98	0,06	14,04	15,27	0,22	15,49	0,89	16,38	0,18	0,22	16,77	3,90
55,81	56,93	0,06	56,99	58,22	0,22	58,44	0,88	59,32	0,18	0,22	59,72	3,90
380,21	381,32	0,06	381,39	382,62	0,22	382,84	0,99	383,83	0,18	0,22	384,22	4,01
384,22	385,33	0,06	385,40	386,63	0,22	386,85	0,88	387,73	0,18	0,22	388,13	3,91
509,16	510,28	0,06	510,34	511,57	0,21	511,79	0,89	512,67	0,18	0,22	513,07	3,90
687,35	688,46	0,06	688,53	689,76	0,22	689,97	0,88	690,86	0,18	0,22	691,25	3,90
408,72	409,84	0,06	409,90	411,13	0,22	411,34	0,89	412,23	0,18	0,22	412,62	3,90

\* Valores de tempos expressos em milissegundos (ms).

Observando os dados da tabela acima, podemos chegar a algumas conclusões.

- Percebemos que a inatividade tanto do mestre como a do escravo é constante.
- Mesmo que o sistema seja constituído por apenas um mestre e um escravo, observamos que a passagem do token é realizada.
- O mestre envia a cada ciclo de mensagem um quadro de status do tipo SD1 para um endereço possível da rede. Esse endereço segue uma ordem seqüencial até atingir o último endereço, num processo de varredura que ocorre sucessivamente.

Para podermos observar melhor essas conclusões, vamos analisar mais dois experimentos que tenham as mesmas características do primeiro, mas alternamos os tamanhos dos quadros de pedido e resposta.

O resultado desses experimentos está nas tabelas 5.5 e 5.6. Podemos observar que os tempos de inatividade continuam constantes e não variaram em relação ao tamanho dos quadros. Também não houve mudanças em relação ao comportamento dos quadros de controle observados anteriormente.

TABELA 5.5 – Experimento 2, pedido 96, resposta 12 octetos, 187,5 Kbits/s

$T_{\text{InícioPed}}$	$T_{\text{Pedido}}$	$T_{\text{InatEsc}}$	$T_{\text{InícioResp}}$	$T_{\text{Resposta}}$	$T_{\text{InatMes}}$	$T_{\text{IniSta}}$	$T$	$T_{\text{IniTok}}$	$T_{\text{TranTok}}$	$T_{\text{InatMes}}$	$T_{\text{ProxPed}}$	$T_{\text{RR}}$
11,01	17,17	0,06	17,23	18,46	0,22	18,68	0,89	19,57	0,18	0,22	19,96	8,95
55,32	61,48	0,06	61,54	62,77	0,22	62,99	0,88	63,87	0,18	0,22	64,27	8,95
172,12	178,28	0,06	178,34	179,57	0,22	179,79	0,89	180,67	0,18	0,22	181,07	8,95
216,87	223,03	0,06	223,09	224,32	0,22	224,54	0,88	225,42	0,18	0,22	225,82	8,95
538,38	544,54	0,06	544,61	545,84	0,22	546,06	0,88	546,94	0,18	0,22	547,34	8,95
112,29	118,45	0,06	118,51	119,74	0,22	119,96	0,88	120,85	0,18	0,22	121,24	8,95
451,71	457,87	0,06	457,93	459,16	0,22	459,38	0,89	460,27	0,18	0,22	460,66	8,95
534,92	541,08	0,06	541,14	542,37	0,22	542,59	0,98	543,57	0,18	0,22	543,97	9,05

\* Valores de tempos expressos em milissegundos (ms).

TABELA 5.6 – Experimento 5, pedido e resposta 120 octetos, 187,5 Kbits/s

$T_{\text{InícioPed}}$	$T_{\text{Pedido}}$	$T_{\text{InatEsc}}$	$T_{\text{InícioResp}}$	$T_{\text{Resposta}}$	$T_{\text{InatMes}}$	$T_{\text{IniSta}}$	$T$	$T_{\text{IniTok}}$	$T_{\text{TranTok}}$	$T_{\text{InatMes}}$	$T_{\text{ProxPed}}$	$T_{\text{RR}}$
19,44	27,00	0,06	27,07	34,63	0,22	34,85	0,88	35,74	0,18	0,22	36,13	16,70
152,99	160,56	0,06	160,62	168,19	0,22	168,41	0,88	169,30	0,18	0,22	169,69	16,70
269,86	277,42	0,06	277,49	285,05	0,22	285,27	0,88	286,15	0,18	0,22	286,55	16,69
420,33	427,89	0,06	427,96	435,52	0,22	435,74	0,98	436,72	0,18	0,22	437,12	16,79
437,12	444,68	0,06	444,75	452,31	0,22	452,53	0,98	453,52	0,18	0,22	453,91	16,79
520,69	528,26	0,06	528,32	535,88	0,22	536,10	0,89	536,99	0,18	0,22	537,38	16,69
942,94	950,51	0,06	950,57	958,13	0,22	958,35	0,88	959,24	0,18	0,22	959,63	16,69
976,33	983,89	0,06	983,96	991,52	0,22	991,74	0,88	992,63	0,18	0,22	993,02	16,70

\* Valores de tempos expressos em milissegundos (ms).

O motivo pelo qual os tempos de inatividade são constantes é que esses tempos são parametrizados no sistema.

Na tabela 5.1 encontramos as variáveis  $T_{\text{SDR}}$  e  $T_{\text{ID1}}$ , que são referentes à inatividade do escravo e do mestre, respectivamente. Dessa forma, podemos concluir que o tamanho dos quadros não influencia o tempo de resposta tanto do mestre como do escravo.

As solicitações de “*status*” e a transmissão do token em uma rede mestre-escravo puro é uma funcionalidade implementada no PROFIBUS e que é a inserção ou remoção de estações sem a necessidade de reinicialização da rede.

O sistema fica monitorando a cada ciclo de mensagem, a presença de novas estações-mestre ou estações-escrava.

Dentro da norma isso está definido como a manutenção da tabela das estações ativas denominada GAP List.

Nas tabelas 5.7, 5.8, 5.9, 5.10 apresentamos os resultados dos experimentos nos quais fixamos os tamanhos dos quadros de pedido e resposta em 120 octetos e variamos a taxa de transmissão em 500, 1500, 6000 e 12000 Kbit/s.

TABELA 5.7 – Experimento 6, pedido e resposta 120 octetos, 500 Kbits/s

$T_{\text{InícioPed}}$	$T_{\text{Pedido}}$	$T_{\text{InatEsc}}$	$T_{\text{InícioResp}}$	$T_{\text{Resposta}}$	$T_{\text{InatMes}}$	$T_{\text{IniSta}}$	$T$	$T_{\text{IniTok}}$	$T_{\text{TranTok}}$	$T_{\text{InatMes}}$	$T_{\text{ProxPed}}$	$T_{\text{RR}}$
13,93	16,77	0,02	16,79	19,63	0,08	19,71	0,54	20,25	0,07	0,08	20,40	6,47
51,49	54,33	0,02	54,35	57,19	0,08	57,27	0,54	57,81	0,07	0,08	57,96	6,47
57,96	60,80	0,02	60,82	63,66	0,08	63,74	0,54	64,28	0,07	0,08	64,43	6,47
59,21	62,04	0,02	62,07	64,90	0,08	64,99	0,54	65,52	0,07	0,08	65,67	6,47
141,17	144,01	0,02	144,03	146,87	0,08	146,95	0,54	147,49	0,07	0,08	147,64	6,47
245,53	248,37	0,02	248,39	251,23	0,08	251,31	0,54	251,85	0,07	0,08	252,00	6,47
268,82	271,66	0,02	271,68	274,52	0,08	274,60	0,54	275,14	0,07	0,08	275,28	6,47
730,21	733,05	0,02	733,07	735,91	0,08	735,99	0,54	736,53	0,07	0,08	736,68	6,47

\* Valores de tempos expressos em milissegundos (ms).

TABELA 5.8 – Experimento 7, pedido e resposta 120 octetos, 1,5 Mbits/s

$T_{\text{InícioPed}}$	$T_{\text{Pedido}}$	$T_{\text{InatEsc}}$	$T_{\text{InícioResp}}$	$T_{\text{Resposta}}$	$T_{\text{InatMes}}$	$T_{\text{IniSta}}$	$T$	$T_{\text{IniTok}}$	$T_{\text{TranTok}}$	$T_{\text{InatMes}}$	$T_{\text{ProxPed}}$	$T_{\text{RR}}$
5,39	6,33	0,009	6,34	7,29	0,03	7,32	0,25	7,57	0,02	0,03	7,62	2,23
7,62	8,57	0,008	8,57	9,52	0,03	9,55	0,25	9,80	0,02	0,03	9,85	2,23
63,57	64,52	0,008	64,53	65,47	0,03	65,50	0,25	65,75	0,02	0,03	65,80	2,23
128,27	129,22	0,008	129,22	130,17	0,03	130,20	0,25	130,45	0,02	0,03	130,50	2,23
163,64	164,59	0,008	164,59	165,54	0,03	165,57	0,25	165,82	0,02	0,03	165,87	2,23
250,63	251,58	0,008	251,58	252,53	0,03	252,56	0,25	252,81	0,02	0,03	252,86	2,23
425,82	426,76	0,008	426,77	427,72	0,03	427,75	0,25	428,00	0,02	0,03	428,05	2,23
473,68	474,63	0,008	474,63	475,58	0,03	475,61	0,25	475,86	0,02	0,03	475,91	2,23

\* Valores de tempos expressos em milissegundos (ms).

TABELA 5.9 – Experimento 9, pedido e resposta 120 octetos, 6 Mbits/s

$T_{\text{InícioPed}}$	$T_{\text{Pedido}}$	$T_{\text{InatEsc}}$	$T_{\text{InícioResp}}$	$T_{\text{Resposta}}$	$T_{\text{InatMes}}$	$T_{\text{IniSta}}$	$T$	$T_{\text{IniTok}}$	$T_{\text{TranTok}}$	$T_{\text{InatMes}}$	$T_{\text{ProxPed}}$	$T_{\text{RR}}$
0,21	0,45	0,0025	0,45	0,69	0,02	0,71	0,12	0,83	0,01	0,38	1,21	1,00
1,21	1,45	0,0025	1,45	1,69	0,02	1,71	0,12	1,83	0,01	0,38	2,21	1,00
3,21	3,45	0,0025	3,45	3,69	0,02	3,71	0,12	3,83	0,01	0,38	4,21	1,00
5,22	5,45	0,0025	5,46	5,69	0,02	5,72	0,12	5,83	0,01	0,38	6,22	1,00
6,22	6,45	0,0025	6,46	6,69	0,02	6,72	0,12	6,83	0,01	0,38	7,22	1,00
7,22	7,45	0,0025	7,46	7,69	0,02	7,72	0,12	7,83	0,01	0,38	8,22	1,00
17,20	17,44	0,0025	17,44	17,68	0,02	17,70	0,12	17,82	0,01	0,38	18,20	1,00
25,20	25,44	0,0035	25,44	25,68	0,02	25,70	0,12	25,82	0,01	0,38	26,20	1,00

\* Valores de tempos expressos em milissegundos (ms).

TABELA 5.10 – Experimento 10, pedido e resposta 120 octetos, 12 Mbits/s

$T_{\text{InícioPed}}$	$T_{\text{Pedido}}$	$T_{\text{InatEsc}}$	$T_{\text{InícioResp}}$	$T_{\text{Resposta}}$	$T_{\text{InatMes}}$	$T_{\text{IniSta}}$	$T$	$T_{\text{IniTok}}$	$T_{\text{TranTok}}$	$T_{\text{InatMes}}$	$T_{\text{ProxPed}}$	$T_{\text{RR}}$
0,09	0,21	0,0018	0,21	0,33	0,02	0,35	0,10	0,45	0,00	0,64	1,09	0,998
1,09	1,21	0,0017	1,21	1,33	0,02	1,35	0,11	1,46	0,00	0,64	2,10	1,007
2,10	2,22	0,0017	2,22	2,34	0,02	2,36	0,10	2,46	0,00	0,64	3,10	0,998
3,10	3,21	0,0017	3,22	3,33	0,02	3,36	0,10	3,45	0,00	0,64	4,09	0,998
4,09	4,21	0,0018	4,21	4,33	0,02	4,35	0,10	4,45	0,00	0,64	5,09	0,998
5,09	5,21	0,0018	5,21	5,33	0,02	5,35	0,10	5,45	0,00	0,65	6,10	1,010
6,10	6,22	0,0018	6,22	6,34	0,02	6,36	0,10	6,46	0,00	0,64	7,10	0,998
13,10	13,22	0,0018	13,22	13,34	0,02	13,36	0,10	13,46	0,00	0,64	14,10	0,998

\* Valores de tempos expressos em milissegundos (ms).

Nos experimentos com a taxa de transmissão igual e maior que 3 Mbit/s foi observado um fato novo e aparentemente sem explicação.

Após a transmissão do token, o mestre disparou uma série de mensagens de “*status*” para o seu próprio endereço.

Por isso, nas tabelas 5.9 e 5.10, o último tempo de inatividade não condiz com a realidade.

Dentro daquele tempo está computada uma série de solicitações de “*status*” do mestre para ele mesmo. Um outro fator intrigante é que os valores são constantes. Dentro da norma não está prevista qualquer situação parecida com essa.

#### 5.4.2 Análise dos experimentos com um mestre e três escravos

Nas tabelas 5.11 e 5.12 estão os resultados obtidos desse segundo grupo experimental. Os dados foram dispostos em duas tabelas, devido a uma quantidade maior de informações a serem demonstradas.

A taxa de transmissão utilizada nessa experimento é de 187,5 Kbit/s e os quadros de pedido e resposta para cada escravo são :

- a) Primeiro escravo - pedido 120 octetos e resposta 120 octetos
- b) Segundo escravo – pedido 32 octetos e resposta 32 octetos
- c) Terceiro escravo – pedido 1 octeto e resposta 3 octetos

Um outro fator importante em relação a esse grupo de experimentos é que os escravos utilizados não são iguais, ou seja, cada escravo possui características diferentes entre si e também não pertencem ao mesmo fabricante.

TABELA 5.11 – Experimento 11 parte 1, 187,5 Kbits/s

T <sub>InícioPed1</sub>	T <sub>Pedido1</sub>	T <sub>InatEsc1</sub>	T <sub>InícioResp1</sub>	T <sub>Resposta1</sub>	T <sub>InatMes</sub>	T <sub>InícioPed2</sub>	T <sub>Pedido2</sub>	T <sub>InatEsc2</sub>	T <sub>InícioResp2</sub>	T <sub>Resposta2</sub>	T <sub>InatMes</sub>
34,57	42,14	0,06	42,20	49,77	0,22	49,99	52,39	0,06	52,45	54,86	0,22
128,00	135,57	0,06	135,63	143,20	0,22	143,41	145,82	0,06	145,88	148,29	0,22
291,50	299,06	0,06	299,13	306,69	0,22	306,91	309,32	0,06	309,38	311,79	0,22
314,85	322,42	0,06	322,48	330,05	0,22	330,27	332,67	0,07	332,74	335,14	0,22
361,57	369,14	0,06	369,21	376,77	0,22	376,99	379,40	0,06	379,46	381,87	0,22
342,94	350,50	0,06	350,56	358,13	0,22	358,35	360,75	0,06	360,81	363,22	0,22
856,68	864,25	0,06	864,31	871,88	0,22	872,09	874,50	0,06	874,56	876,97	0,21
844,08	851,65	0,06	851,71	859,28	0,22	859,50	861,90	0,07	861,97	864,37	0,22
899,86	907,43	0,06	907,49	915,06	0,22	915,28	917,68	0,06	917,74	920,15	0,22
945,78	953,35	0,06	953,40	960,97	0,22	961,19	963,60	0,06	963,66	966,06	0,22
969,22	976,79	0,06	976,85	984,42	0,22	984,64	987,04	0,06	987,10	989,51	0,22

\* Valores de tempos expressos em milissegundos (ms).

TABELA 5.12 – Experimento 11 parte 2, 187,5 Kbits/s

T <sub>InícioPed3</sub>	T <sub>Pedido3</sub>	T <sub>InatEsc3</sub>	T <sub>InícioResp3</sub>	T <sub>Resposta3</sub>	T <sub>InatMes</sub>	T <sub>IniSta</sub>	T	T <sub>IniTok</sub>	T <sub>TranTok</sub>	T <sub>InatMes</sub>	T <sub>ProxPed</sub>	T <sub>RR</sub>
55,08	55,66	0,06	55,73	56,43	0,22	56,65	0,89	57,54	0,18	0,22	57,93	23,36
148,51	149,09	0,06	149,15	149,86	0,22	150,08	0,89	150,96	0,18	0,22	151,36	23,35
312,01	312,59	0,06	312,65	313,36	0,22	313,57	0,88	314,46	0,18	0,22	314,85	23,36
335,36	335,95	0,06	336,01	336,71	0,21	336,93	0,90	337,82	0,18	0,22	338,22	23,37
382,09	382,67	0,06	382,73	383,43	0,22	383,65	0,89	384,54	0,18	0,22	384,93	23,36
363,44	364,02	0,06	364,08	364,79	0,22	365,00	0,88	365,89	0,18	0,22	366,28	23,35
877,18	877,77	0,06	877,83	878,53	0,22	878,75	0,88	879,63	0,18	0,22	880,03	23,35
864,59	865,18	0,06	865,24	865,94	0,22	866,16	0,88	867,04	0,18	0,22	867,44	23,36
920,36	920,95	0,06	921,01	921,72	0,22	921,93	0,98	922,91	0,18	0,22	923,31	23,44
966,28	966,87	0,06	966,92	967,63	0,22	967,85	0,98	968,83	0,18	0,22	969,22	23,45
989,72	990,31	0,06	990,37	991,07	0,22	991,29	0,88	992,18	0,18	0,22	992,57	23,35

\* Valores de tempos expressos em milissegundos (ms).

Dadas as características do experimento, podemos perceber que o comportamento da rede continua previsível em relação à inatividade do mestre e dos escravos.

Se compararmos os valores obtidos anteriormente, a única diferença é que o polling do mestre tem uma maior quantidade de escravos para percorrer.

Dessa forma, podemos afirmar que um número maior de escravos não altera o comportamento do mestre.

Um outro ponto a ser observado nesse experimento é uma quantidade maior de quadros, confirmando a solicitação de “*status*” enviado pelo mestre.

Como o mestre precisa de 127 ciclos de mensagens para percorrer todos os endereços da rede, nesse experimento a cada 127 ciclos houve 3 quadros de confirmação de “*status*” enviado pelos escravos ao mestre. Esses quadros são do tipo SD2 com 6 bytes.

Nesse experimento podemos perceber um acréscimo, em média, de 0,08 milissegundos no tempo real de rotação do token que são referentes à transmissão desses quadros de reconhecimento.

No experimento seguinte mantivemos os tamanhos dos quadros de pedido e resposta e aumentamos a taxa de transmissão para 500 Kbit/s. Os dados obtidos estão dispostos nas tabelas 5.13 e 5.14.

TABELA 5.13 – Experimento 12 parte 1, 500 Kbits/s

T <sub>InícioPed1</sub>	T <sub>Pedido1</sub>	T <sub>InatEsc1</sub>	T <sub>InícioResp1</sub>	T <sub>Resposta1</sub>	T <sub>InatMes</sub>	T <sub>InícioPed2</sub>	T <sub>Pedido2</sub>	T <sub>InatEsc2</sub>	T <sub>InícioResp2</sub>	T <sub>Resposta2</sub>	T <sub>InatMes</sub>
29,27	29,49	0,02	29,51	29,77	0,08	29,86	32,69	0,02	32,72	35,55	0,08
38,23	38,45	0,02	38,47	38,74	0,08	38,82	41,66	0,02	41,68	44,52	0,08
127,89	128,11	0,02	128,14	128,40	0,08	128,48	131,32	0,02	131,34	134,18	0,08
423,23	423,45	0,02	423,47	423,73	0,08	423,82	426,65	0,02	426,68	429,52	0,08
512,88	513,10	0,02	513,13	513,39	0,08	513,47	516,31	0,02	516,33	519,17	0,08
284,86	285,08	0,02	285,10	285,37	0,08	285,45	288,29	0,02	288,31	291,15	0,08
338,66	338,88	0,02	338,90	339,16	0,08	339,25	342,08	0,02	342,11	344,94	0,08
347,62	347,84	0,02	347,86	348,13	0,08	348,21	351,05	0,02	351,07	353,91	0,08

\* Valores de tempos expressos em milissegundos (ms).

TABELA 5.14 – Experimento 12 parte 2, 500 Kbits/s

T <sub>InícioPed3</sub>	T <sub>Pedido3</sub>	T <sub>InatEsc3</sub>	T <sub>InícioResp3</sub>	T <sub>Resposta3</sub>	T <sub>InatMes</sub>	T <sub>IniSta</sub>	T	T <sub>IniTok</sub>	T <sub>TranTok</sub>	T <sub>InatMes</sub>	T <sub>ProxPed</sub>	T <sub>RR</sub>
35,64	36,54	0,02	36,56	37,46	0,08	37,55	0,538	38,08	0,07	0,08	38,23	8,96
44,60	45,50	0,02	45,53	46,43	0,08	46,51	0,538	47,05	0,07	0,08	47,20	8,97
134,26	135,16	0,02	135,19	136,09	0,08	136,17	0,538	136,71	0,07	0,08	136,86	8,96
429,60	430,50	0,02	430,52	431,43	0,08	431,51	0,538	432,05	0,07	0,08	432,19	8,97
519,25	520,15	0,02	520,18	521,08	0,08	521,16	0,538	521,70	0,07	0,08	521,85	8,97
291,23	292,13	0,02	292,15	293,06	0,08	293,14	0,538	293,68	0,07	0,08	293,82	8,97
345,03	345,93	0,02	345,95	346,85	0,08	346,94	0,538	347,47	0,07	0,08	347,62	8,96
353,99	354,89	0,02	354,92	355,82	0,08	355,90	0,538	356,44	0,07	0,08	356,59	8,97

\* Valores de tempos expressos em milissegundos (ms).

Como podemos observar, os tempos de controle continuam previsíveis sem nenhum tipo de alteração

O comportamento relatado anteriormente, referente aos quadros de “*status*” enviados do mestre para ele mesmo, surgiu novamente nesse grupo de experimento com as taxas maiores e iguais a 3 Mbit/s.

Nas tabelas 5.15 e 5.16 modificamos os valores dos quadros de pedido e resposta de dois escravos para 2 e 4 octetos, respectivamente, e mantivemos nos últimos escravos os mesmos valores dos experimentos anteriores. A taxa de transmissão é de 500 Kbit/s.

TABELA 5.15 – Experimento 21 parte 1, 500 Kbits/s

T <sub>InícioPed1</sub>	T <sub>Pedido1</sub>	T <sub>InatEsc1</sub>	T <sub>InícioResp1</sub>	T <sub>Resposta1</sub>	T <sub>InatMes</sub>	T <sub>InícioPed2</sub>	T <sub>Pedido2</sub>	T <sub>InatEsc2</sub>	T <sub>InícioResp2</sub>	T <sub>Resposta2</sub>	T <sub>InatMes</sub>
8,84	9,06	0,02	9,08	9,34	0,08	9,42	9,67	0,02	9,69	9,97	0,08
18,99	19,21	0,02	19,23	19,50	0,08	19,58	19,82	0,02	19,84	20,13	0,08
29,15	29,37	0,02	29,39	29,65	0,08	29,74	29,98	0,02	30,00	30,29	0,08
46,93	47,15	0,02	47,17	47,43	0,08	47,52	47,76	0,02	47,78	48,07	0,08
95,18	95,40	0,02	95,42	95,68	0,08	95,77	96,01	0,02	96,03	96,32	0,08
118,03	118,25	0,02	118,27	118,53	0,08	118,62	118,86	0,02	118,88	119,17	0,08
151,53	151,75	0,02	151,77	152,04	0,08	152,12	152,36	0,02	152,38	152,67	0,08
911,25	911,47	0,02	911,49	911,76	0,08	911,84	912,08	0,02	912,10	912,39	0,08

\* Valores de tempos expressos em milissegundos (ms).

TABELA 5.16 – Experimento 21 parte 2, 500 Kbits/s

T <sub>InícioPed3</sub>	T <sub>Pedido3</sub>	T <sub>InatEsc3</sub>	T <sub>InícioResp3</sub>	T <sub>Resposta3</sub>	T <sub>InatMes</sub>	T <sub>IniSta</sub>	T	T <sub>IniTok</sub>	T <sub>TranTok</sub>	T <sub>InatMes</sub>	T <sub>ProxPed</sub>	T <sub>RR</sub>
10,06	10,30	0,02	10,32	10,61	0,08	10,69	0,54	11,23	0,07	0,08	11,37	2,54
20,21	20,45	0,02	20,48	20,76	0,08	20,84	0,54	21,38	0,07	0,08	21,53	2,54
30,37	30,61	0,02	30,63	30,92	0,08	31,00	0,54	31,54	0,07	0,08	31,69	2,54
48,15	48,39	0,02	48,41	48,70	0,08	48,78	0,54	49,32	0,07	0,08	49,47	2,54
96,40	96,64	0,02	96,66	96,95	0,08	97,03	0,54	97,57	0,07	0,08	97,72	2,54
119,25	119,49	0,02	119,51	119,80	0,08	119,88	0,54	120,42	0,07	0,08	120,57	2,54
152,75	152,99	0,02	153,02	153,30	0,08	153,38	0,54	153,92	0,07	0,08	154,07	2,54
912,47	912,71	0,03	912,73	913,02	0,08	913,10	0,54	913,64	0,07	0,08	913,79	2,54

\* Valores de tempos expressos em milissegundos (ms).

Como podemos observar fica claro que, por ser uma rede determinística, os eventos são quase todos previsíveis. Uma outra vantagem que devemos levar em consideração são

os tamanhos dos quadros de controle que, em velocidades acima de 3000 Kbit/s, tornam-se quase desprezíveis devido ao seu tamanho.

O ponto crucial para realizarmos uma boa avaliação de desempenho seria determinar com exatidão os tamanhos médios dos quadros de pedido e resposta.

## 5.5 Arquitetura de um escravo PROFIBUS

Para compreendermos melhor os valores apresentados anteriormente, precisamos entender o funcionamento de um escravo no que diz respeito à atualização dos valores nos seus módulos de I/O.

Como mostra a figura 5.4, um escravo é dotado de um processador e dois coprocessadores que são responsáveis pelos processos envolvidos no escravo.

O coprocessador dos módulos de I/O é responsável pela leitura e atualização dos valores nas portas de I/O, visto que cada porta possui um endereço que fica armazenado em uma memória compartilhada.

O processador faz a ligação entre a memória compartilhada dos módulos de I/O com a memória compartilhada da rede. Ele é responsável pela tramitação das informações nos dois sentidos.

Dessa forma, a memória da rede contém os valores de memória da última varredura realizada pelo processador, e o processador também atualiza os valores de memória dos módulos de I/O vindos da memória da rede.

E, por último, o coprocessador da rede fica responsável pela transmissão e recepção dos quadros e a leitura ou atualização dos endereços de I/O na sua memória.

Quando um mestre solicita o estado de uma das portas, o escravo simplesmente lê o valor contido na memória compartilhada e envia a resposta para o mestre.

Um outro ponto importante que ocorre dentro de um escravo são os testes de diagnósticos dos módulos, teste dos terminadores e teste de diagnóstico do processador e coprocessador.

Esses teste são realizados a cada 100 ms e não são sincronizados, ou seja, tanto o processador como o coprocessador realizam seus teste de forma independente. Dessa forma, a atualização de uma porta de saída pode levar mais ou menos tempo, isso depende se caso antes de atualizar o endereço o coprocessador entre no estado de diagnóstico ou não.

Ocorrem casos em que um dado colocado na memória compartilhada da rede tem que esperar o processador e depois o coprocessador a realizar os seus testes de diagnósticos para depois prosseguir com sua operação normal. Essa situação seria o pior caso possível.

Dada essa explicação, podemos concluir que os tempos aqui apresentados não são os tempos medidos na saída dos módulos de I/O, mas sim os tempos de rede do PROFIBUS.

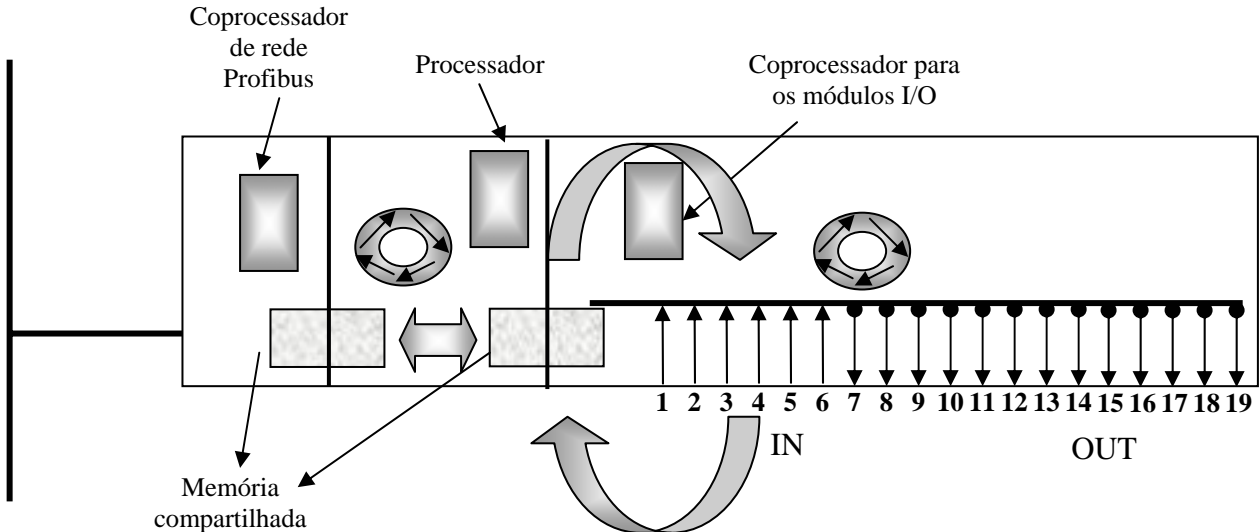


FIGURA 5.4 – Arquitetura de um escravo Profibus

### 5.6 Modelo analítico para uma rede mestre escravo puro

Abaixo propomos um modelo o qual se aproxima mais da realidade apresentada nas análises anteriores.

Um novo  $T_{MC}$  é calculado para que possamos cruzar o resultado obtido com os valores da monitoração.

Na fórmula do novo  $T_{MC}$  estão sendo inseridos os tempos de transmissão do token  $T_{TC}$  e a solicitação de "status" pelo mestre  $T_{Status}$ .

$$T_{TC} = T_{TF} + T_{ID1} + T_{TD} \quad (26)$$

$$T_{Status} = T_{TS} + T_{Slot}, \quad \text{onde } T_{TS} \text{ é o tempo de transmissão do quadro SD1} \quad (27)$$

$$T_{MC} = (T_{S/R} + T_{SDR} + T_{A/R} + T_{ID1} + 2 * T_{TD} + T_{Status} + T_{TC}) \quad (28)$$

Na tabela 5.17 apresentamos os valores calculados a partir do novo  $T_{MC}$  para uma taxa de transmissão de 187,5 Kbit/s a 12 Mbit/s e quadros de pedido e resposta de 120 octetos.

Podemos observar que nas taxas acima de 3 Mbit/s estão inseridos os tempos de transmissão de consecutivos quadros de "status" que comentamos anteriormente e que não estão previstos dentro das normas do PROFIBUS.

Na figura 5.5 temos o gráfico taxa de transmissão versus tempos de resposta medidos e estimados.

TABELA 5.17 – Dados medidos versus estimados em uma rede mestre-escravo puro

	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
Valores Medidos	16,72	6,47	2,23	1,30	1,00	1,00
Valores Estimados	16,65	6,44	2,21	1,15	0,61	0,34
%Erro	0,42%	0,46%	0,90%	11,54%	39,00%	66,00%

\* Valores de tempos expressos em milissegundos (ms).



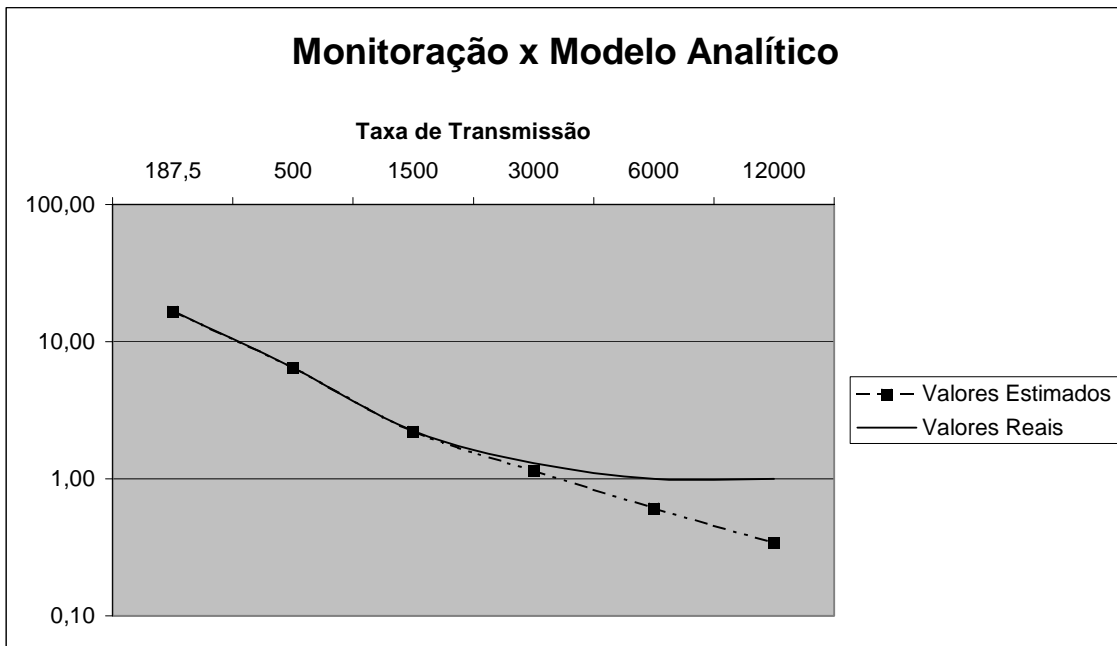


FIGURA 5.5 – Gráfico do modelo analítico para uma rede mestre escravo puro

### 5.7 Modelo analítico para uma rede com 1 mestre e n escravos

A diferença principal de uma rede mestre escravo puro para uma rede com um mestre e  $n$  escravos está nos tempos de “status” envolvidos na comunicação. Como já foi dito, após a conclusão do polling, que é a comunicação entre o mestre e seus escravos, ocorre a manutenção da GAP, que é a lista de estações ativas na rede.

Essa manutenção ocorre de forma que o mestre envia uma mensagem de “status” para um endereço possível da rede, ou seja, em uma rede PROFIBUS são aceitos 127 endereços, a cada manutenção da GAP o PROFIBUS certifica-se que um endereço está ativo ou não. Desta forma o PROFIBUS percorrerá todos os endereços após 127 ciclos de polling.

Estão disponíveis dois tipos de “status”: o primeiro, quando o endereço não está ativo demonstrado na equação 27 e o segundo, quando o endereço está ativo, conforme a equação 29.

No primeiro caso, simplesmente levamos em consideração o tempo de transmissão do quadro do tipo SD1 mais o tempo de slot. No segundo caso, temos que considerar a confirmação do endereço, que também se utiliza de um quadro do tipo SD1 mais os tempos de inatividade e  $T_{SDR}$ .

Abaixo apresentamos o desenvolvimento das fórmulas :

$$T_{Status\ 2} = 2 * T_{TS} + T_{SDR} + T_{ID} \quad (29)$$

$$T_{MC} = T_{S/R} + T_{A/R} + T_{SDR} + T_{ID1} + 2 * T_{TD} \quad (30)$$

$$T_{SR} = \left( \sum_1^{Num\_Escravos} (T_{MC} + nr * T_{MC}) \right) + (Num\_Escr * T_{Status\ 2} + (127 - Num\_Escr) * T_{Status\ 1}) / 127 + T_{TC} \quad (31)$$

onde :

Num\_Escr = quantidade de estações escravas

$nr$  = quantidade de ciclos de repetição

Como já foi dito anteriormente, o  $T_{TC}$  e o tempo de transmissão do token e os tempos  $T_{STATUS1}$  e  $T_{STATUS2}$  são os valores relativos à transmissão e confirmação dos endereços ativos da rede.

A variável  $T_{MC}$  é referente ao ciclo de mensagem que o mestre realiza com um escravo. E o valor  $nr$  são os ciclos de retransmissão que é a quantidade de mensagens retransmitidas durante o ciclo normal de um mestre com seu escravo. De posse do  $T_{SR}$ , podemos calcular o tempo de reação do sistema em uma rede com um mestre e vários escravos.

Na tabela 5.18 apresentamos os valores calculados a partir do  $T_{SR}$ , descrito anteriormente para uma taxa de transmissão entre 187,5 Kbit/s a 12 Mbit/s. A rede é composta por um mestre e três escravos e os quadros de pedido e resposta são 120 octetos para o primeiro escravo, 32 octetos para o segundo escravo e de 1 e 3 octetos para o terceiro escravo. Esses dados são referentes aos experimentos 11 a 16 da tabela 5.2.

TABELA 5.18 – Dados medidos versus estimados com 1 mestres 3 escravos

Amostras	Descrição	187,5 Kbit/s	500 Kbit/s	1500 Kbit/s	3000 Kbit/s	6000 Kbit/s	12000 Kbit/s
1°	Valores Medidos	23,35	8,96	3,07	1,59	0,99	1
	Valores Estimados	23,27	8,92	3,04	1,56	0,82	0,45
	% Erro	0,36%	0,45%	1,03%	1,65%	16,99%	55,02%
2°	Valores Medidos	23,36	8,97	3,06	1,58	1,01	1
	Valores Estimados	23,27	8,92	3,04	1,56	0,82	0,45
	% Erro	0,40%	0,56%	0,65%	1,27%	18,81%	55,00%
3°	Valores Medidos	23,44	8,96	3,07	1,59	1,01	1,01
	Valores Estimados	23,27	8,92	3,04	1,56	0,82	0,45
	% Erro	0,74%	0,45%	0,98%	1,89%	18,81%	55,45%
	Média Valores Medidos	23,38	8,96	3,07	1,59	1,00	1,00
	Valores Estimados	23,27	8,92	3,04	1,56	0,82	0,45
	% Erro	0,47%	0,45%	0,98%	1,88%	18%	55%

\* Valores de tempos expressos em milissegundos (ms).

Podemos observar que nas taxas acima de 3 Mbit/s estão inseridos os tempos de transmissão de consecutivos quadros de “*status*” que comentamos anteriormente e que não estão previstos dentro das normas do PROFIBUS.

Os tempos reais de rotação do token que não estão com solicitações anormais de “*status*” apresentam um percentual de erro muito pequeno em relação aos dados coletados na monitoração.

Isso demonstra que as fórmulas retratam o desempenho da rede com percentual de confiabilidade aceitável. Na figura 5.6 temos o gráfico taxa de transmissão versus tempos de resposta medidos e estimados.

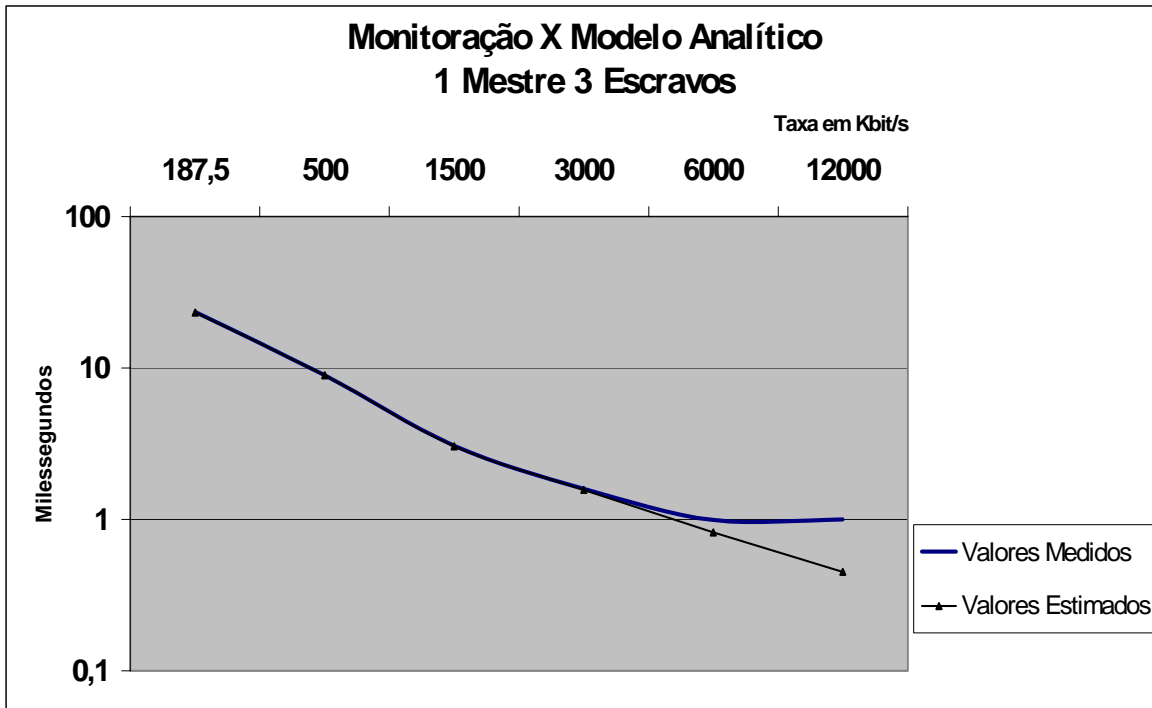


FIGURA 5.6 – Gráfico do modelo analítico para uma rede 1 mestre e  $n$  escravo

### 5.8 Modelo analítico para uma rede com $n$ mestre e $n$ escravos

Para calcularmos o tempo desejado de rotação de uma rede com  $n$  mestre e escravos, é só somarmos todos os  $T_{SR}$  de cada mestre da rede, como demonstramos abaixo :

$$\min T_{TR} = \sum_1^{nem} T_{SR} \quad (32)$$

onde

$nem$  = número de estações mestre

O valor obtido será o tempo desejado de rotação do token o qual pode ser utilizado para parametrizar a rede e estipular o pior caso de atendimento da rede, ou seja, é o tempo máximo de rotação do token de uma rede com vários mestres e escravos.

### 5.9 Variáveis Aleatórias

Podemos realizar uma análise probabilística dos tamanhos dos quadros de pedido e resposta.

Como não temos conhecimento de qual será o tamanho dos quadros, podemos supor que o comprimento dos quadros de pedido e resposta seja uma variável aleatória.

Dessa forma, teremos a soma de duas variáveis aleatórias mais as constantes que são os tempos de controle.

Se uma variável aleatória  $X$  poder ser representada pela soma de quaisquer  $n$  variáveis aleatórias independentes (que satisfaçam a determinadas condições que valem na maioria das aplicações), então essa soma, para  $n$  suficientemente grande, terá distribuição aproximadamente normal.

Esse notável resultado é conhecido como o Teorema do Limite Central. [MEY 78]

Uma outra afirmação de MEY 1978 diz que a soma de um número finito de variáveis aleatórias independentes e normalmente distribuídas é também normalmente distribuída.

A curva em forma de sino, apresentada na figura 5.7, representa a distribuição de probabilidade conhecida como distribuição normal.

A distribuição normal tem dois parâmetros,  $\mu$  e  $\sigma$  que têm o seguinte significado :

$\mu$  : centro da distribuição (média)

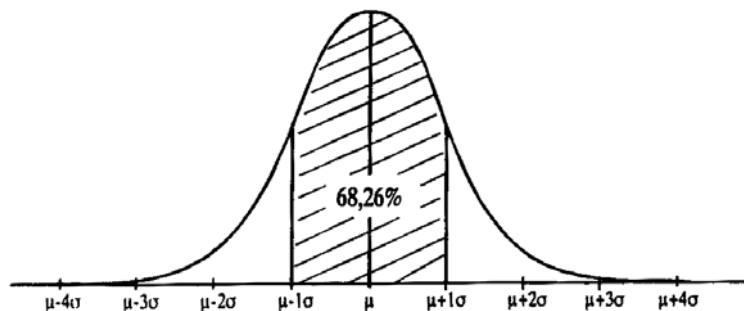
$\sigma$  : dispersão da distribuição (desvio padrão)

Uma interpretação simples de desvio padrão  $\sigma$  de uma distribuição normal é ilustrada na figura 5.7. Se utilizarmos o desvio padrão para dividir a área debaixo da curva da distribuição normal, podemos obter as áreas associadas a cada intervalo como uma proporção da área total sob a curva.

Essa proporção representa a probabilidade de se obterem valores em cada intervalo, quando são coletadas amostras aleatórias de uma população com essa forma de distribuição.

Observe-se que a probabilidade de se obterem valores dentro dos limites definidos pela média menos um desvio padrão e a média mais um desvio padrão ( $\mu \pm 1 \sigma$ ) é de, aproximadamente, 68,26%, enquanto que a probabilidade de obtenção de um valor fora desses limites é de, aproximadamente, 31,74%.

Similarmente, a probabilidade de ocorrência de valores fora do intervalo ( $\mu \pm 3 \sigma$ ) é extremamente pequena, aproximadamente 0,3% ou seja, 3 em 1000. [DEL 94]



INTERVALO	PROBABILIDADE	
	DENTRO	FORA
$(\mu \pm 1 \sigma)$	68,26 %	31,74 %
$(\mu \pm 2 \sigma)$	95,46 %	4,54 %
$(\mu \pm 3 \sigma)$	99,73 %	0,27 %
$(\mu \pm 4 \sigma)$	99,9936 %	0,0064 %

FIGURA 5.7 – Probabilidade associadas à distribuição normal

Uma questão que surge é como verificar se a distribuição normal é um modelo apropriado para representar a variabilidade presente na característica de qualidade em estudo. A maneira mais simples de responder a essa questão é por meio da comparação

entre o histograma dos dados e a curva da distribuição normal. Porém esse método só pode ser utilizado se forem disponíveis muitas amostras (pelo menos 50) [DEL 94]

Uma questão a ser abordada em relação ao modelo analítico demonstrado anteriormente é a hipótese de utilizarmos valores médios com uma distribuição normal de média  $X$  e um desvio padrão para os quadros de pedido e resposta. Podemos afirmar que, caso estejamos trabalhando com valores médios, obteremos como resposta um tempo médio com o mesmo tipo de distribuição normal, ou seja, a formulação também é válida para valores médios com uma distribuição normal.

### 5.10 Ferramenta para resolução do modelo analítico proposto.

Para resolver as equações descritas anteriormente foi desenvolvida uma aplicação que possibilita estimar o tempo desejado de rotação de uma rede PROFIBUS. O funcionamento é extremamente simples: o usuário deverá informar a quantidade de mestres que a rede possui e para cada mestre a quantidade de escravos.

Existem duas maneiras de informar os valores de dados para os mensagens de pedido e resposta. Na primeira maneira, os valores informados serão válidos para todos os escravos daquele mestre, ou seja, os tamanhos das mensagens de pedido e resposta e a quantidade de ciclos de repetição são únicos para todos os escravos do mestre em questão.

Já na segunda forma, os tamanho, das mensagens poderão ser informados de escravo a escravo, ou seja, para cada escravo os valores de dados podem ser diferentes em relação a outros escravos do mesmo mestre. Uma outra vantagem é que o sistema permite intercalar as duas formas de informar os dados.

Escravo	Tamanho dos dados do pacote de pedido	Tamanho dos dados do pacote de resposta	Ciclos de repetição por polling
Escravo 01			50
Escravo 02			100
Escravo 03			25
Escravo 04			246
Escravo 05			158

FIGURA 5.8 – Simulador de análise de desempenho de redes PROFIBUS

Na figura 5.8 podemos perceber que o usuário informou 40 octetos para todos as mensagens de pedido do mestre para seus escravos e informou para cada escravo um tamanho de mensagem de resposta diferente para o mestre.

Lembramos que os valores informados para as mensagens de pedido e resposta não devem ser acrescentados de valores de cabeçalho, ou seja, são simplesmente valores do campo de dados da mensagem.

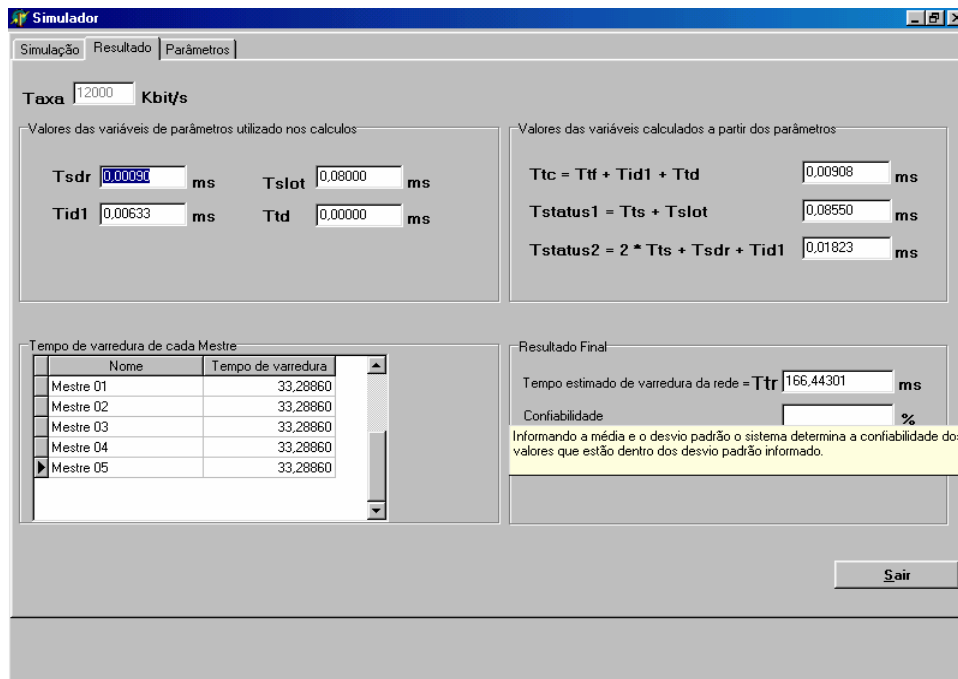


FIGURA 5.9 – Tela de apresentação dos resultados obtidos na simulação

O sistema se encarrega de acrescentar os valores de overhead nos cálculos. Após informar os dados e selecionar a opção de cálculo, a aplicação fornece o resultado do tempo desejado de rotação como mostra a figura 5.9.

Os valores dos parâmetros usados no cálculo são informados para que o usuário possa validar o resultado.

Simulador

Simulação Resultado Parâmetros

Variável	9,6 Kbits/s	19,2 Kbits/s	93,75 Kbits/s	187,5 Kbits/s	500 Kbits/s	1500 Kbits/s	3000 Kbits/s	6000 Kbits/s	0 K
TID1	1.5000	1.5000	1.5000	0.1970	0.0740	0.0200	0.0150	0.0100	
TID2	2.0000	2.0000	2.0000	0.3200	0.2000	0.1000	0.0830	0.0750	
TQUI	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0010	0.0010	
TRDY	0.0000	0.0000	0.0000	0.0590	0.0220	0.0070	0.0040	0.0020	
TSDR	2.0000	2.0000	2.0000	0.0590	0.0220	0.0100	0.0040	0.0018	
TSET	0.0000	0.0000	0.0000	0.0050	0.0020	0.0010	0.0010	0.0010	
T SLOT	0.5300	0.5300	0.5300	0.5300	0.4000	0.2000	0.1300	0.1000	
TTD	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	

Sair

FIGURA 5.10 – Tempos configuráveis da aplicação

A aplicação também disponibiliza, como apresentado na figura 5.10, um módulo, para que o usuário tenha a possibilidade de configurar os tempos que são parametrizáveis na rede.

### 5.11 Estudo de caso

Para validarmos o modelo proposto e onde se encaixa a estratégia adotada, realizamos um estudo de caso sobre o projeto CTPetro do qual a Universidade Federal do Rio Grande do Sul faz parte.

Esse projeto visa ao criar uma arquitetura de hardware e software para controladores programáveis de alta capacidade a serem utilizados em plataformas de petróleo. O projeto exige um tempo máximo de varredura para os oito mil pontos em torno de 180 milissegundos.

A configuração estimada para a rede é de 5 mestres com 70 escravos cada mestre. Cada escravo, que é uma cabeça de I/O, controla 24 portas divididas em portas de entrada e saída.

Neste estudo de caso chegamos à conclusão, com o auxílio da ferramenta que para atender ao requisito de varredura exigido pela Petrobrás, a aplicação poderá utilizar os tamanhos máximos dos quadros de pedido e resposta que é de 246 octetos, dando um tempo de varredura final em torno de 166,44 milissegundos a uma taxa de 12 Mbit/s.

A análise foi realizada de uma forma simples, sem conhecer a fundo as características da aplicação. Com o uso do modelo proposto, conseguimos estimar qual é o pior caso na comunicação entre os mestres e os escravos.

Dessa forma, a ferramenta proporciona ao projetista uma boa noção em relação ao tamanho dos quadros cuja aplicação terá como limite atender à exigência de varredura da rede.

Por se tratar de uma rede do tipo “*request/response*” com acesso determinístico ao barramento, a realização de uma avaliação de desempenho se torna fácil a partir do momento que se conhecem os valores médios dos quadros de pedido e resposta.

Uma ferramenta matemática que poderíamos utilizar para solucionar esse problema seria o uso de variáveis aleatórias, mas não foi possível demonstrar a sua eficácia por não termos à disposição uma rede com uma aplicação real para realizarmos a monitoração da rede e obter os dados necessários.

O uso da aplicação facilita a obtenção dos tempos de rotação para a rede. Várias análises poder ser realizadas rapidamente sem que haja a necessidade de implementar uma rede fisicamente.

Se trabalharmos com médias de tamanhos das mensagens de pedido e resposta dentro da aplicação com uma distribuição normal, a estratégia continuará válida. A aplicação retornará um tempo médio com uma distribuição normal estimado de rotação desejado do token.

## 6 Conclusões

A escolha do protocolo para estudo está relacionado com a disponibilidade de acesso à documentação oficial e à possibilidade de realizar experimentos práticos. Dessa forma, o PROFIBUS foi escolhido dentre vários outros protocolos.

Apresentamos suas características técnicas e detalhes de seu funcionamento, com o intuito de dar ao leitor um panorama do que seja um sistema desse tipo, a que se destina e quais são suas potencialidades.

Uma comparação com outro padrão que tem grande aceitação no mercado o, Foundation FIELDBUS está descrita em um capítulo, de forma que o leitor possa compreender as várias diferenças entre as normalizações disponíveis. Também procuramos apresentar o texto, fugindo da rigidez técnica da norma oficial, mas sem omitir informações imprescindíveis para a compreensão do funcionamento do protocolo.

Estão dispostos também em anexo, com uma maior riqueza de detalhes, a característica e o funcionamento do protocolo Foundation FIELDBUS.

Um estudo detalhado sobre os tempos de controle é realizado, dando a possibilidade de compreender os tempos envolvidos no processo de comunicação entre as estações mestre e escravos.

Durante o processo de monitoração, percebemos que o protocolo se comporta de uma forma bem determinística. Sendo assim, uma série de dúvidas que tínhamos em relação a alguns pontos da norma, como por exemplo, o mecanismo de resposta imediata, foram esclarecidas.

Como o protocolo se comporta de uma forma previsível, a maior questão envolvida em uma avaliação de desempenho está em determinar os tamanhos médios das mensagens de pedido e resposta.

Dessa forma, para realizamos uma avaliação de desempenho não seria necessário o uso da teoria de filas, sendo que durante a comunicação não ocorre enfileiramento de solicitações, ou seja, o mestre aguarda a resposta de cada escravo quando efetua o polling de comunicação.

Como cada aplicação desenvolvida pode ter uma variação dos tamanhos dos quadros, uma vez conhecidos esses tamanhos, a determinação dos tempos de rotação do token torna-se uma tarefa simples, visto que demonstramos as equações 28, 31 e 32 para determinar esses tempos.

Como cada aplicação monitora um grupo de dispositivos e o processo de comunicação é uma varredura dos valores das portas de I/O, podemos dizer que cada aplicação obedece a uma média em relação aos seus quadros.

Finalmente, foram apresentados, no último capítulo, os procedimentos de monitoração e a demonstração do modelo analítico proposto.

Apesar dos problemas encontrados durante a monitoração, descritos no trabalho, e o protocolo ter uma característica do tipo “*request-response*”, ou seja, solicita e aguarda a resposta, o modelo proposto consegue determinar com uma margem de erro aceitável os tempos de rotação ou atendimento de uma rede PROFIBUS.



Apresentamos uma possível solução para determinar analiticamente as médias dos pacotes em cada aplicação. Também verificamos a eficácia do modelo em relação aos valores médios com uma distribuição normal nos pacotes de pedido e resposta. Constatamos que o modelo fornece valores médios de tempo com a mesma distribuição normal dos tamanhos de pacotes.

Dessa forma, conseguimos atingir nosso objetivo e disponibilizar uma pesquisa que abrange uma descrição detalhada dos protocolos e uma forma rápida de se realizar uma avaliação de desempenho, oferecendo uma grau de confiabilidade aceitável, possibilitando, antes mesmo, de implementar uma rede, verificar se os requisitos temporais atendem aos tempos exigidos pela aplicação.

## Anexo 1 Foundation Fieldbus

O "Field Bus" é o nome dado pelo IEC para sistemas de comunicação e controle em tempo real entre equipamentos de medidas e controladores. Esse protocolo também se baseia no modelo OSI da ISO, mas com uma redução no número de camadas.

São definidos no modelo as camadas :

- ◆ PL (Physical Layer),
- ◆ DLL (Data Link Layer);
- ◆ Aplicação, que está subdividida em :
  - ◆ FMS (Fieldbus Message Specification);
  - ◆ FAS (Fieldbus Access Sublayer);

### 1 Camada física

A norma ANSI/ISA-S50.02-1992, aprovada em 17 de maio de 1994, "Fieldbus Standard for Use in Industrial Control Systems Part 2 : Physical Layer Specification and Service Definition" trata do meio físico para a realização das interligações.

Segundo [SMA 98] os principais itens do meio físico são:

- transmissão de dados somente digital;
- self clocking;
- comunicação bidirecional;
- código Manchester;
- modulação de voltagem (acoplamento paralelo);
- velocidades de transmissão de 31.25 Kbit/s a 100 Mbit/s;
- redundância no barramento;
- barramento sem energia, não intrinsecamente seguro;
- barramento com energia, não intrinsecamente seguro;
- barramento sem energia, intrinsecamente seguro;
- barramento com energia, intrinsecamente seguro.

A conversão das mensagens em sinais físicos no meio de transmissão inclui a adição e remoção de preâmbulos no início e no final dos delimitadores.

O número de dispositivos possíveis em um link FIELDBUS depende de alguns fatores como a força consumida de cada dispositivo, o tipo de cabo utilizado e o uso de repetidores. O número de endereços de rede para cada link é de 240. As taxas de transmissão variam de 31.25 Kbit/s a 100 Mbit/s, para as taxas de transmissão mais altas somente a topologia de barramento é permitida.

No nível de instrumentos ligados ao barramento, a velocidade normalizada é 31.25 Kbit/s. As outras velocidades deverão ser utilizadas para a interligação de bridges e gateways para a conexão em alta velocidade desses dispositivos.

Na velocidade de 31.25 Kbit/s, a norma determina, dentre outras coisas, as seguintes regras :

Um instrumento FIELDBUS deve ser capaz de se comunicar entre os seguintes números de equipamentos :

- Entre 2 e 32 equipamentos numa ligação sem segurança intrínseca e alimentação separada da fiação de comunicação;
- Entre 2 e 6 instrumentos alimentados pela mesma fiação de comunicação numa ligação com segurança intrínseca;
- Entre 1 e 12 instrumentos alimentados pela mesma fiação de comunicação numa ligação sem segurança intrínseca.

Essa regra não impede a ligação de mais instrumentos do que o especificado. Os números foram alcançados levando-se em consideração o consumo de 9 mA +/- 1 mA, com tensão de alimentação de 20 VDC e barreiras de segurança intrínseca com 19 VDC de saída e entre 40 e 60 mA de corrente para os instrumentos. Um barramento carregado com o número máximo de instrumentos na velocidade de 31.25 Kbit/s não deve ter entre quaisquer dois equipamentos o comprimento maior que 1900 m (incluindo as derivações) como mostra a fig 1.1.

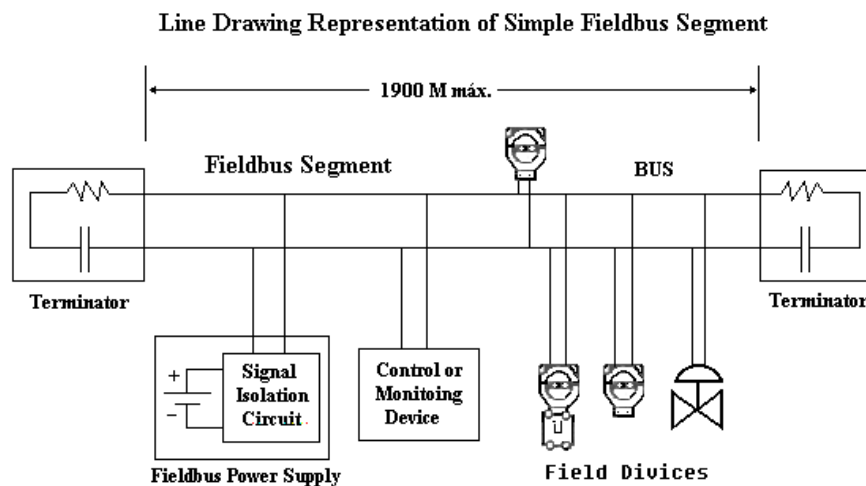


FIGURA 1.1 – Distância máxima entre dois dispositivos

Como mostra a figura 1.2 o número máximo de repetidores para a regeneração da forma de onda entre dois instrumentos não pode exceder a 4.

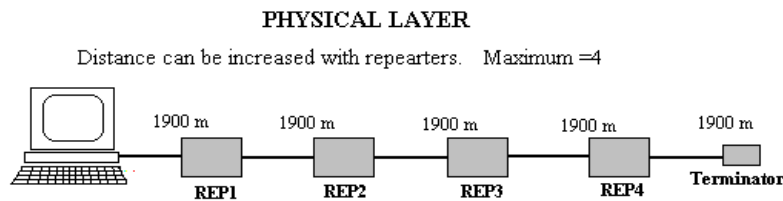


FIGURA 1.2 – Número máximo de repetidores

Para sistemas com meio físico redundante cada canal deve atender às regras de configuração de redes e não deve existir um segmento não redundante entre dois

segmentos redundantes. Os repetidores também deverão ser redundantes e os números dos canais deverão ser mantidos no FIELDBUS, isto é, os canais do FIELDBUS devem ter os mesmos números dos canais físicos.

## 2 Camada de enlace de dados (Data Link Layer)

A camada de enlace de dados é responsável pelo controle ao meio de transmissão determinando quem pode transmitir e quando esta transmissão vai ocorrer. É também responsável pela transmissão de mensagens na rede FIELDBUS garantindo que os dados cheguem corretamente ao destinatário.

A camada de enlace de dados administra o acesso ao meio através de um barramento programado, centralizado e determinístico, chamado de LAS (Link Active Scheduler). É a LAS que determina quem tem acesso ao barramento e em que momento uma estação pode iniciar os seus ciclos de mensagens. [FOU 2000]

São definidos dois tipos de dispositivo na especificação da DLL como mostra a figura 1.3 :

1. BD (Basic Device)
2. LMD (Link Master Device)

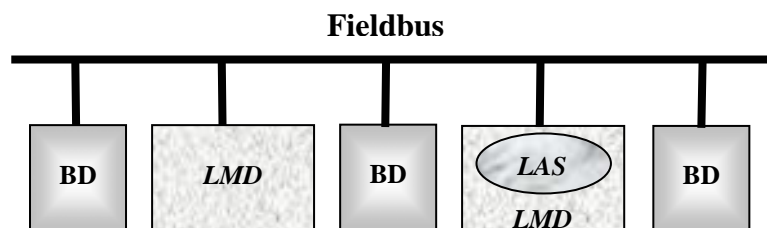


FIGURA 1.3 – Dispositivos definidos na DLL

Os dispositivos básicos podem ser sensores ou atuadores. Estes dispositivos não tem a capacidade de formar a LAS. Já os Link Master são dispositivos mais sofisticados e são capazes de formar a LAS.

São definidos dois tipos de comunicação entres os dispositivos, comunicação escalonada e a não escalonada.

### 2.1 Comunicação escalonada

Cada dispositivo que está conectado no barramento tem um buffer de dados, a LAS mantém uma lista de tempos de transmissão para todos os buffers de todos os dispositivos que necessitam de transmissão cíclica.

A transmissão ocorre com o envio de uma mensagem CD (Compel Data) pela LAS como mostra a figura 1.4. Após a recepção dessa mensagem o dispositivo difunde (broadcasts) os seus dados que estão contidos no buffer para todos os outros dispositivos conectados a rede. Os dispositivos configurados para receber os dados enviados pela estação detentora da mensagem CD são chamados de “Subscriber”, e a estação que difunde os seus dados é chamada de “Publisher”.

A programação de transferência de dados é tipicamente usada para regular a transferência cíclica de controle de loop entre os dispositivos FIELDBUS. [FOU 2000] A comunicação escalonada é a atividade de prioridade mais alta executada pela LAS.

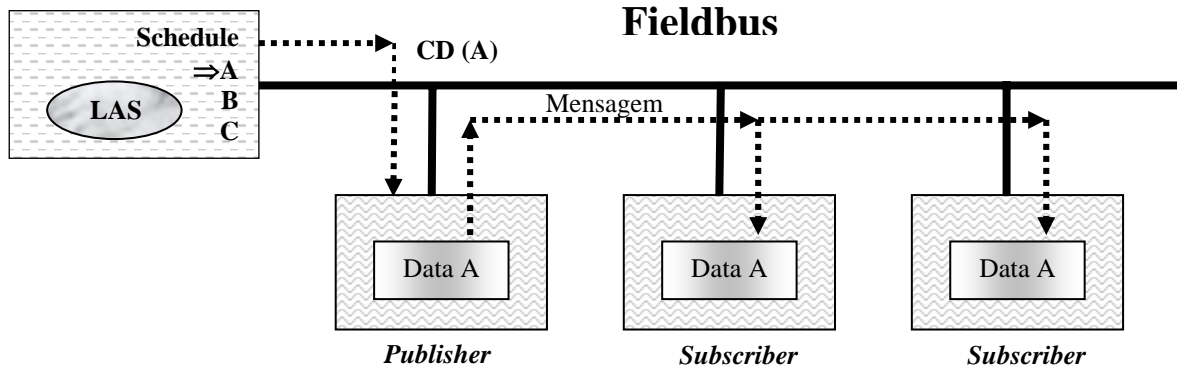


FIGURA 1.4 – Fluxo de comunicação escalonada

## 2.2 Comunicação não escalonada

Todos os dispositivos que estão conectados a rede tem um tempo determinado para enviar as mensagens não escalonadas (mensagens do operador ou notificações de eventos). Como mostra a figura 1.5 a LAS concede essas permissões aos dispositivos através de uma mensagem PT (Pass Token). Quando o dispositivo recebe um PT a ele foi concedido permissão para enviar mensagens até que se termine a transmissão da mensagem ou até que se encerre o seu tempo máximo para transmissão.

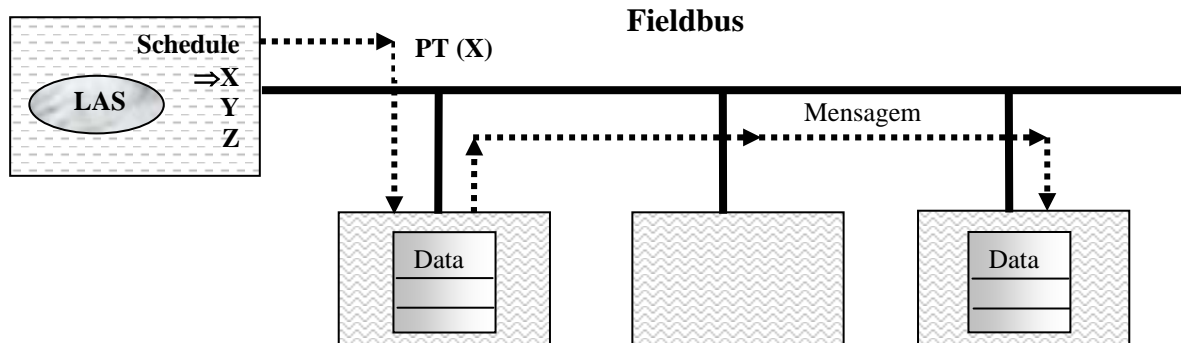


FIGURA 1.5 – Fluxo de comunicação não escalonada

## 2.3 Manutenção das listas de vivos

O FIELDBUS monitora todos os dispositivos presentes na rede que estão respondendo corretamente a uma mensagem PT e armazena essa informação em uma lista que é chamada de “Lista de Vivos” ou “Live List”. A inserção de novos dispositivos na rede pode ser realizada a qualquer momento. A LAS periodicamente envia uma mensagem PN (Probe Node) para os endereços dos dispositivos que se encontram no barramento. Se o dispositivo está presente no endereço e recebe uma mensagem do tipo PN, ele imediatamente retorna uma mensagem PR (Probe Response). Se o dispositivo respondeu com um PR a LAS adiciona o dispositivo para a “Lista de Vivos” e confirma sua adição enviando uma mensagem para o dispositivo NA (Node Activation). [FOU 2000]

O dispositivo permanecerá na lista de vivos contanto que ele responda corretamente ao envio de mensagens PT. A LAS removerá esse dispositivo da lista de vivos caso ele não faça qualquer uso do token ou não responda a LAS depois de três tentativas sucessivas. [FOU 2000]

Sempre que um dispositivo é adicionado ou removido da lista de vivos, a LAS difunde as alterações para a lista de vivos de todos os dispositivos. Isso permite manter em cada dispositivo uma cópia atual da lista de vivos.

É exigido da LAS verificar pelo menos um endereço depois que completar o ciclo de envio de mensagens PTs.

## 2.4 Tempo de sincronização

É também de responsabilidade da LAS a sincronização dos relógios, periodicamente ele difunde uma mensagem TD (Time Distribution) que tem como finalidade ajustar os clocks de cada dispositivo do barramento, desse modo todos os dispositivos terão o mesmo tempo de transmissão de dados.

## 2.5 Redundância da LAS

O FIELDBUS permite a redundância dos dispositivos Link Master, conseqüentemente, permite a formação de outras LAS caso ocorra falha no Link Master onde foi formado inicialmente a LAS. Um dos Link Master ativos decidirá qual dispositivo se tornará a nova LAS.

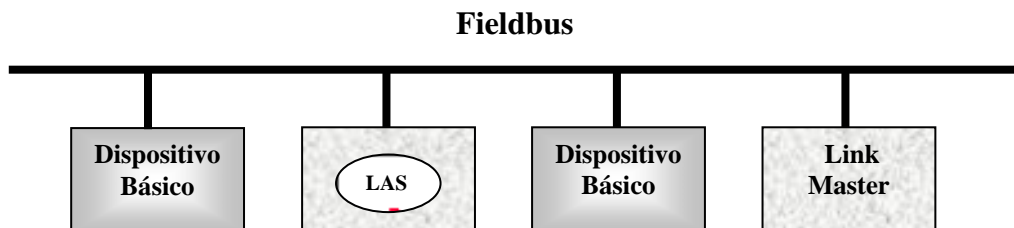


FIGURA 1.6 – Modelo com múltiplos link master

## 3 Fieldbus Access Sublayer (FAS)

O FAS usa as características escalonadas e não escalonadas da camada DDL para fornecer serviços para o FMS. Os tipos de serviços FAS são descritos segundo o VCR (Virtual Communication Relation Ships).

### 3.1 Client/Server VCR

O client/server VCR é usado para comunicação não escalonada entre dispositivos no FIELDBUS “um para um”. Quando um dispositivo recebe um PT da LAS este pode enviar uma mensagem de pedido para um outro dispositivo no FIELDBUS. O pedido é chamado de “*Client*” e o dispositivo que recebeu o pedido é chamado de “*Server*”. O servidor somente envia a resposta para o cliente quando ele receber um PT da LAS. O client/server VCR é usado para iniciar operação como alteração “setpoint”, acesso e alteração de parâmetros, reconhecer alarme, é *upload* e *download* nos dispositivos.

### 3.2 Report Distribution VCR

O VCR Report Distribution é usado em comunicações não escalonada, de um para muitos dispositivos no FIELDBUS.

Quando um dispositivo com um evento ou um “Trend Report” recebe um PT do LAS, este pode enviar uma mensagem para um “Grupo de endereços” definido por um VCR. O VCR Report Distribution é tipicamente usado pelo dispositivo FIELDBUS para enviar ou notificar alarmes para a console operação. [FOU 2000]

### 3.3 Publisher / Subscriber VCR

O VCR Publisher / Subscriber é usado em comunicação não escalonada de um para muitos dispositivos no FIELDBUS.

Quando um dispositivo necessita difundir os dados do seu buffer para outros dispositivos no FIELDBUS ele aguarda a recepção de um tipo mensagem CD. Após a recepção de um CD, o barramento está disponível para a transmissão dos dados do seu buffer para todos os dispositivos no FIELDBUS. Os dispositivos que são configurados para receber os dados enviados pelo dispositivo que detém a mensagem CD são chamados de “Subscriber”.

## 4 FMS (Fieldbus Message Specification)

A FMS descreve os serviços de comunicação, os formatos das mensagens e o comportamento do protocolo necessário para a construção de mensagens utilizadas nas aplicações de usuários figura 1.7.

Os serviços FMS permitem que as aplicações de usuário troquem mensagens entre si no FIELDBUS, usando um padrão de formato de mensagem especificado na FMS.

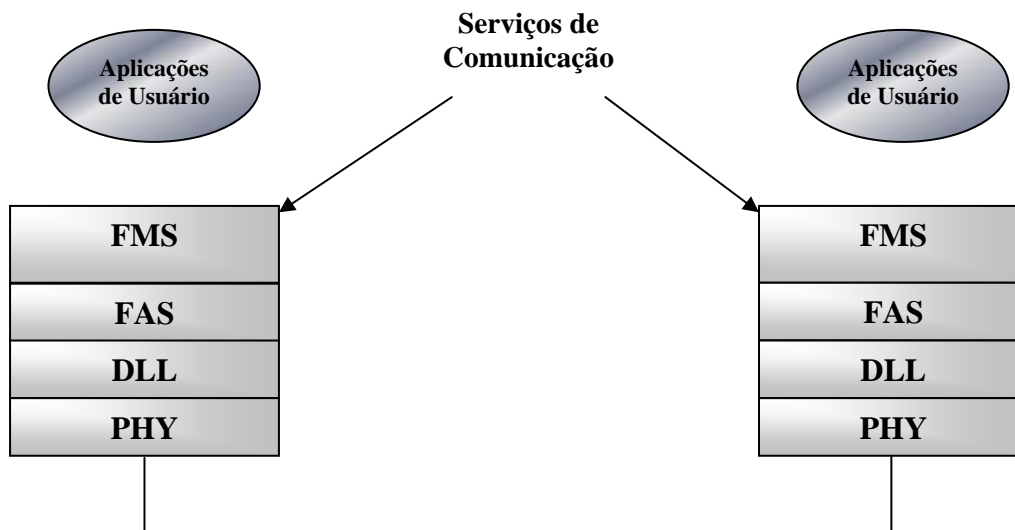


FIGURA 1.7 – Especificação de mensagens para aplicações de usuários

Os dados que são transmitidos no FIELDBUS estão descritos em um OD (Object Dictionary). Os objetos descritos são identificados por um “índice” no OD. O índice 0 chama o cabeçalho do dicionário de objetos, o qual prove uma descrição do próprio

dicionário e define o primeiro índice para as descrições de objetos das aplicações de usuários. Os objetos das aplicações de usuário descritos podem começar de qualquer índice acima de 255. [FOU 2000]

Os índices abaixo de 255 são utilizados para definir normas como, por exemplo, os tipos de dados que podem ser *boolean*, *integer*, *float*, *bitstring* e as estruturas de dados que são usados para construir outros objetos. [FOU 2000]

#### **4.1 VFD (Virtual Field Device)**

Um VFD é usado para visualizar remotamente dados de um dispositivo descrito em um dicionário de objeto. Um dispositivo típico tem pelo menos 2 VFDs.

Ele fornece configuração para a pilha de comunicação. O VFD é usado pela administração da rede e também pela administração dos sistemas. Esse VFD fornece acesso para o NMIB (NetWork Management Information Base) e para o SMIB (System Management Information Base). Os dados NMIB incluem o VCR, variáveis dinâmicas, estatísticas e a LAS (se o dispositivo é um Link Master). Dados da SMIB incluem os dispositivos *Tag*, informações de endereços e escalonamento para execução de funções de bloco.

#### **4.2 Serviços de comunicação**

Especificações de serviços de comunicações FMS são definidos para cada tipo de objeto. Todos os Serviços FMS podem usar só o tipo VCR Client/Server.

Os serviços disponíveis são :

- ◆ Upload/Download Services
- ◆ Program Invocation Services
- ◆ Message Formatting
- ◆ Protocol Behavior

#### **4.3 Aplicações de usuários**

A Fundação FIELDBUS definiu normas de aplicações de usuários baseado em blocos que são representações para diferentes tipos de aplicação.

Os tipos de blocos usados em uma aplicação de usuários são descritos abaixo.

- ◆ Resource Block
- ◆ Function Block
- ◆ Transducer Block

#### **4.4 RB (Resource Block)**

Os blocos de recursos descrevem as características do dispositivo FIELDBUS como nomes dos dispositivos, fabricante e número de série. Há somente 1 recurso de bloco em um dispositivo.



#### **4.5 FB (Function Block)**

Os FB provêm o comportamento do sistema de controle. Os parâmetros de entrada e saída dos FB podem ser unidos em cima do FIELDBUS. A execução de cada bloco de função é precisamente escalonada.

Podem haver muitos FBs em uma única aplicação de usuários. A Fundação FIELDBUS definiu dez padrões para controles básicos (FF-891) e dezenove padrões adicionais para controles avançados (FF-892). OS FB podem ser construídos em dispositivos FIELDBUS à medida em que precisam alcançar a funcionalidade desejada do dispositivo.

#### **4.6 TB (Transducer Blocks)**

Ele contém informações como a data de calibração e tipos de sensores. Há normalmente um TB para cada entrada ou saída de bloco de funções.

## Bibliografia

- [BEN 93] BENDER, K.; KATZ, M. et al. **Profibus – the fieldbus for industrial automation**. Englewood Cliffs: Prentice Hall, 1993.
- [BUR 96] BURNS, A.; WELLINGS, Andy. **Real-Time and Programming Languages**. England: Addison Wesley, 1996.
- [CAR 89] CARO, R. H. Fieldbus Applications. **ISA Transactions**, [S.l.], v.28, n.4, p. 65-70, 1989.
- [CAV 96] CAVALIERI, S.; STEFANO, A.; MIRABELLA, O. Mapping automotive process control on IEC/ISA Fieldbus functionalities. **Computers in Industry**, Amsterdam, v. 28, p. 77-80, May 1994.
- [CHA 94] CHATHA, A. Fieldbus : The Foundation for Field Control System. **Control Engineering**, Barrington, v. 41, p. 77-80, May 1994.
- [DAM 88] DAMSKER, D. J. Assesment of industrial data network standards. **IEEE Transactions on Energy Conversion**, New York, v. 3, n. 2, p. 199-204, June 1966.
- [DEL 94] DELLARETTI FILHO, O. **Itens de Controle e Avaliação de Processos**. Belo Horizonte: Fundação Christiano Ottoni, 1994.
- [DIN 91] DIN DEUTSCHES INSTITUT FUR NORMUNG. **Profibus standard**. Berlin: Beuth Verlag, 1991. 760p. (DIN 19 245 part 1 and 2 : process field bus).
- [FER 94] FERREIRA, A. P. L. **Teoria das Filas aplicada à Comunicação de Dados**. Porto Alegre : UFRGS, 1994.
- [FIE 2000] FOUNDATION FIELDBUS. **Home Page**. Disponível em: <<http://www.fieldbus.org>>. Acesso em: 18 jan. 2000.
- [FOU 2000] FOUNDATION FIELDBUS. **Technical Overview**. Disponível em: <<http://www.fieldbus.org>>. Acesso em: 18 jan 2000.
- [GIO 86] GIOZZA, W. F. et al. **Redes locais de computadores**. São Paulo: McGraw-Hill, 1986.
- [JAI 91] JAIN, R. **The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling**. Littleton, Massachusetts: John Wiley & Sons, 1991.

- [KLE 76] KLEINROCK, L. **Queueing systems v. 1**. Littleton, Massachusetts: John Wiley & Sons, 1976. p. 507-554.
- [KLI 76] KLEINROCK, L. **Queueing systems v. 2**. Littleton, Massachusetts: John Wiley & Sons, 1976. p. 10-40.
- [LEM 95] LEMOS, G.; COLCHER, S.; SOARES, L. F. G. **Redes de Computadores**: das LANS, MANS às redes ATM. Rio de Janeiro: Campus, 1995. 705 p
- [MEY 78] MEYER, P. **Probabilidade** : aplicações à estatística. Rio de Janeiro: LTC, 1978. p.262-301.
- [NAZ 93] NAZAR, R. F. **Uso de rede de campo para aplicação em instrumentação e controle em tempo real**. 1993. Dissertação (Mestrado) Escola de Engenharia, Universidade de São Carlos, São Carlos.
- [PAC 98] PACHECO, A C. **Escalonamento temporal em rede Fieldbus**. 1998. Dissertação (Mestrado) Escola de Engenharia, Universidade de São Carlos, São Carlos.
- [PRO 2000] PROFIBUS. Home Page. Disponível em: <<http://www.profibus.com>>. Acesso em : 18 jan. 2000.
- [PRO 98] PROFIBUS Specification. Normative Parts of PROFIBUS – FMS, -DP, -PA according to the European Standard EN 50 170. [S.l.], 1998. v. 2.
- [SAN 96] SANTOS, J. L. **Implementação e uso de rede fieldbus para monitoramento e controle de aplicação em tempo real**. 1996. Dissertação (Mestrado) Escola de Engenharia, Universidade de São Carlos, São Carlos.
- [SIL 96] SILVA, R. D. **Comunicação de dados em ambiente industrial** : um protocolo para automação e controle em tempo real. 1996. Dissertação (Mestrado) Escola de Engenharia, Universidade de São Carlos, São Carlos.
- [STA 91] STALLINGS, W. A practical guide to queuing analysis. **Byte**, Peterborough, v. 16, n. 2, p. 309-316, Feb 1991.
- [TAM 97] TANENBAUM, A. S. **Redes de computadores**. Rio de Janeiro : Campus, 1997. 923p.