

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE BARATELLA LUGLI

**UMA ARQUITETURA DISTRIBUÍDA FLEXÍVEL PARA
APLICAÇÕES INDUSTRIAIS BASEADA NO PADRÃO
ETHERNET.**

TESE DE DOUTORADO

Fevereiro de 2013
Itajubá - MG

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE BARATELLA LUGLI

**UMA ARQUITETURA DISTRIBUÍDA FLEXÍVEL PARA
APLICAÇÕES INDUSTRIAIS BASEADA NO PADRÃO
ETHERNET.**

Tese submetida ao Programa de Pós-Graduação em Engenharia Elétrica como parte dos requisitos para obtenção do Título de *Doutor em Ciências em Engenharia Elétrica*.

Área de concentração: Automação e Sistemas Elétricos Industriais (ASEI).

Orientador (a): Profa. Dra. Lúcia Regina Horta Rodrigues Franco.

Co-Orientador (a): Prof. Dr. Max Mauro Dias Santos.

Fevereiro de 2013
Itajubá – MG

UNIVERSIDADE FEDERAL DE ITAJUBÁ
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ALEXANDRE BARATELLA LUGLI

**UMA ARQUITETURA DISTRIBUÍDA FLEXÍVEL PARA
APLICAÇÕES INDUSTRIAIS BASEADA NO PADRÃO
ETHERNET.**

Tese aprovada por banca examinadora em 07 de Fevereiro de 2013, conferindo ao autor o título de *Doutor em Ciências em Engenharia Elétrica*.

Banca Examinadora:

Profa. Dra. Lúcia R. H. R. Franco (Orientadora) – UFABC.

Prof. Dr. Max Mauro Dias Santos (Co-Orientador) – UTFPR.

Prof. Dr. Yvo Marcelo Chiaradia Masselli – INATEL.

Profa. Dra. Adriana Aparecida dos Santos Izidoro – FEPI.

Prof. Dr. Luiz Edival de Souza – UNIFEI.

Prof. Dr. Edson da Costa Bortoni – UNIFEI.

Fevereiro de 2013
Itajubá – MG

AGRADECIMENTOS

Gostaria de agradecer em especial a minha esposa Thays Christina e ao meu filho Enzo, pelo companheirismo, apoio e incentivo; a empresa Sense Eletrônica, na figura do Sr. Sérgio Augusto Bertoloni, pelo apoio e ajuda nos momentos decisivos deste trabalho; aos meus pais José Roberto e Silvana e meu irmão Arthur pelo apoio durante minha árdua caminhada em direção ao sucesso profissional; aos meus sogros Francisco e Lydia; aos meus amigos Humberto, Cassiano e Murilo, pela ajuda técnica e ao INATEL, pela realização desse trabalho, em especial ao prof. Carlos Nazareth pelo grande incentivo e apoio.

Aos meus orientadores, Profa. Dra. Lúcia Regina Horta Rodrigues Franco e Dr. Max Mauro Dias Santos, pela paciência e atenção em suprir minhas lacunas de conhecimento e experiência, na análise e na sugestão do texto da tese, bem como nas informações e conselhos que auxiliaram na elaboração do modelo final e sua realização prática.

E finalmente a Deus que, ao me presentear com características pessoais essenciais, me delegou a capacidade de utilizá-las em proveito de minha evolução pessoal e de toda a humanidade.

A TODOS, A MINHA ETERNA GRATIDÃO.

Fevereiro – 2013

DEDICATÓRIA

Dedico carinhosamente este trabalho a minha esposa, ao meu filho Enzo e a minha mãe, pela persistência e coragem que vocês me passaram para enfrentar os grandes desafios deste trabalho. Obrigado pelo enorme apoio!

“Deus é a minha fortaleza e a minha força, e Ele perfeitamente desembaraça o meu caminho.”

II Samuel 22:33

RESUMO

A suíte de protocolos TCP/IP é um dos conjuntos mais difundidos para comunicação a longa, média e curtas distâncias envolvendo sistemas computacionais. Assim, tentando uma uniformização no ambiente industrial, há alguns anos, tal padrão foi desenvolvido nesse meio. Este trabalho explora esta tecnologia, porém voltada para os ambientes industriais e tem por objetivo propor uma nova maneira de realizar o controle dos elementos de campo, utilizando uma arquitetura distribuída flexível aplicada as redes Ethernet industriais. A proposta do trabalho é o desenvolvimento de um algoritmo para escalonamento das mensagens, aplicado a uma arquitetura distribuída, com a retirada do controlador mestre, onde somente os módulos de campo e *switches* irão operar na rede, tendo, assim, um controle distribuído da comunicação das mensagens do barramento. O algoritmo proposto utiliza uma técnica de escalonamento tipo *off-line* ou *pré-run-time*. Assim, o trabalho desenvolve um novo conceito de comunicação, aplicado as redes Ethernet industriais, com o controle da comunicação da rede e de cada elemento de campo distribuído ao longo do processo industrial. Para isso, é necessário modelar o novo conceito de comunicação, realizar o escalonamento das mensagens (que é essencial para definir a ordem de comunicação das mensagens trafegadas na rede devido a não utilização de um controlador mestre centralizado) e realizar os testes de verificação e validação da idéia proposta.

Palavras Chaves: Controle Distribuído, Escalonamento, Ethernet, Redes Industriais e Simulação.

ABSTRACT

The TCP/IP suite protocol is one of the most widespread protocols groups for communication at long, average and short distances involving computer systems. Then, for trying the industrial networks standardization, for some years, it was also implemented in the industrial environment. The thesis aims to propose a new way to accomplish the fieldbus modules control, using a flexible distributed architecture applied to the industrial Ethernet networks. The work proposal is to develop an algorithm for messages scheduling, applied to a distributed architecture, with the removal of the master controller, where only the fieldbus modules and switches operate on the network, and then having a communication messages distributed control. The proposed algorithm uses an off-line or pre-run-time type technique of messages scheduling. Therefore, the thesis develops a new communication concept, applied industrial Ethernet networks, with communication network control and each field element distributed throughout the manufacturing process. For this, it is necessary to model the new communication concept, perform the messages scheduling (which is essential for defining the message communication order in the network trafficked, due to not using a centralized master controller) and perform verification testing and idea proposal validation.

Key Words: Distributed Control, Ethernet, Industrial Networks, Scheduling and Simulation.

LISTA DE TABELAS

Tabela 1 – Padrões e protocolos de acordo com a IEC 61784 e IEC 61158.	4
Tabela 2 – Resumo dos tipos de comunicação para rede Ethernet/IP.	21
Tabela 3 – Características físicas das redes Ethernet industriais.	22
Tabela 4 – Tabela comparativa entre HSE e H1.	24
Tabela 5 – Tabela comparativa entre os diversos padrões Ethernet industrial.	29
Tabela 6 – Comparativo entre escalonadores <i>on-line</i> e <i>off-line</i>	49
Tabela 7 – Atenuação do cabo Ethernet industrial em função da frequência.	62
Tabela 8 – Principais funções dos elementos da ferramenta computacional proposta.	65
Tabela 9 – Tabela de execução do Módulo_0 – cenário 1.	69
Tabela 10 – Tabela de execução do Módulo_1 – cenário 1.	69
Tabela 11 – Tabela de execução do Módulo_2 – cenário 1.	69
Tabela 12 – Tabela de execução do Módulo_3 – cenário 1.	69
Tabela 13 – Tabela de execução do Módulo_4 – cenário 1.	69
Tabela 14 – Tabela de execução do Módulo_0 – cenário 2.	78
Tabela 15 – Tabela de execução do Módulo_1 – cenário 2.	78
Tabela 16 – Tabela de execução do Módulo_2 – cenário 2.	78
Tabela 17 – Tabela de execução do Módulo_3 – cenário 2.	78
Tabela 18 – Tabela de execução do Módulo_4 – cenário 2.	79
Tabela 19 – Tabela de execução do Módulo_5 – cenário 2.	79
Tabela 20 – Tabela de execução do Módulo_6 – cenário 2.	79
Tabela 21 – Tabela de execução do Módulo_7 – cenário 2.	79
Tabela 22 – Tabela de execução do Módulo_8 – cenário 2.	79
Tabela 23 – Tabela de execução do Módulo_9 – cenário 2.	80
Tabela 24 – Tabela de execução do Módulo_10 – cenário 2.	80
Tabela 25 – Tabela de execução do Módulo_11 – cenário 2.	80
Tabela 26 – Tabela de execução do Módulo_12 – cenário 2.	80
Tabela 27 – Tabela de execução do Módulo_13 – cenário 2.	80
Tabela 28 – Tabela de execução do Módulo_14 – cenário 2.	81
Tabela 29 – Tabela de execução do Módulo_15 – cenário 2.	81
Tabela 30 – Tabela de execução do Módulo_16 – cenário 2.	81
Tabela 31 – Tabela de execução do Módulo_17 – cenário 2.	81

Tabela 32 – Tabela de execução do Módulo_18 – cenário 2.	81
Tabela 33 – Tabela de execução do Módulo_19 – cenário 2.	82
Tabela 34 – Tabela de execução do Módulo_0 – cenário 3.	90
Tabela 35 – Tabela de execução do Módulo_1 – cenário 3.	90
Tabela 36 – Comparativo entre as diversas redes Ethernet industriais.	96

LISTA DE SIGLAS E SÍMBOLOS

ARP – Address Resolution Protocol.
CBA – Component Based Automation.
CIP – Control Information Protocol.
CRC – Cyclic Redundancy Check.
CSMA/CD – Carrier Sense with Multiple Access and Collision Detect.
DARPA – Defense Advance Research Projects Agency.
DC/DC – Direct Current/Direct Current.
DP – Decentralized Periphery.
EDF – Earliest Deadline First.
EDS – Electronic DataSheet.
EN – European Normative.
EPA – Ethernet for Plant Automation
EPL – Ethernet Power Link
ERF – Earliest Release First.
FIP – French Association for Standardization.
FTP – File Transfer Protocol.
GSD – General Station Description.
H1 – Rede industrial Foundation Fieldbus
HSE – High Speed Ethernet
HTTP – Hyper Text Transfer Protocol.
ICMP – Internet Control Message Protocol.
IEC – International Electrotechnical Commission.
IEEE – Institute of Electrical and Electronics Engineers.
IGMP – Internet Group Management Protocol.
I/O – Input/Output.
IP – Internet Protocol.
IP67 – Índice de proteção do equipamento, número 67.
IRT – Isochronous Real Time.
ISA – International Society of Automation.
ISO – International Standards Organization.
LLF – Least Stack Time.

ODVA – Open Device Vendors Association.

OSI – Open System Interconnection.

PA – Process Automation.

PLC – Programmer Logic Controller.

PoE – Power Over Ethernet.

RARP – Reverse Address Resolution Protocol.

RFC – Request for Comments.

RJ – Registered Jack, número 45.

RM – Rate Monotonic.

RS – Recommended Standard.

SNMP – Simple Network Management Protocol.

SRT – Soft Real Time.

TCP – Transport Control Protocol.

TCP/IP – Suíte de protocolos para comunicação a longa, média ou pequena distância.

UDP – User Datagram Protocol.

UML – Unified Modeling Language.

WWW – World Wide Web.

XML – Extensible Markup Language.

Σ – Somatório.

LISTA DE UNIDADES DE MEDIDAS

A – Símbolo de corrente elétrica, em Ampére.

dB – Unidade de Decibel.

Gbps – 10^9 bits transmitidos em um segundo.

Kbps – 10^3 bits transmitidos em um segundo.

Km – Unidade de Quilômetros.

m – Unidade de metros.

mA – 10^{-3} Ampéres.

Mbps – 10^6 bits transmitidos em um segundo.

ms – 10^{-3} segundos.

ns – 10^{-9} segundos.

pF – 10^{-12} Faraday.

psi – Unidade de pressão - Pound Square Inch.

μ s – 10^{-6} segundos.

Vac – Unidade de tensão no modo alternado.

Vdc – Unidade de tensão no modo contínuo.

W – Símbolo de potência, em Watts.

LISTA DE FIGURAS

Figura 1 – Conectores RJ 45 para as redes Ethernet industriais.....	5
Figura 2 – Cabo blindado para as redes Ethernet industriais.	5
Figura 3 – <i>Switch</i> Ethernet industrial com prioridade de transmissão.	6
Figura 4 – Configuração para sistema PoE – <i>Full duplex</i>	7
Figura 5 – Configuração para sistema PoE – <i>Half duplex</i>	7
Figura 6 – Modelo OSI da ISO.....	9
Figura 7 – Suíte de protocolos TCP/IP.....	10
Figura 8 – Quadro Ethernet.	11
Figura 9 – Quadro IP.	12
Figura 10 – Quadro do protocolo TCP.	13
Figura 11 – Quadro do protocolo UDP.	13
Figura 12 – PROFIBUS x PROFINET.	14
Figura 13 – Suíte de protocolos TCP/IP para PROFINET.....	16
Figura 14– <i>Gateways</i> PROFINET/PROFIBUS DP e PROFINET/Interbus.	17
Figura 15 – Níveis de ciclos de comunicação em função dos requisitos temporais da aplicação industrial.....	18
Figura 16 – Quadro PROFINET IO.	19
Figura 17 – Comunicação no Ethernet/IP.	20
Figura 18 – Representação de uma configuração para rede Ethernet/IP.....	21
Figura 19 – Desenho esquemático de uma aplicação com Ethernet industrial.	23
Figura 20 – Aplicação real de uma rede Ethernet para uma indústria de manufatura.....	23
Figura 21 – Exemplo de rede H1 conectada a uma rede HSE.....	25
Figura 22 – Divisão de camadas para a rede Sercos III.....	26
Figura 23 – Divisão de camadas para a rede Modbus TCP.....	26
Figura 24 – Divisão de camadas para a rede EtherCAT.....	27
Figura 25 – Divisão de camadas para a rede EPL.....	28
Figura 26 – Exemplo de escalonamento RM, <i>off-line</i>	34
Figura 27 – Exemplo de bandas de comunicação cíclica e acíclica.	35
Figura 28 – Estudo de caso 1: Relatório de visita técnica.....	37
Figura 29 – Estudo de caso 2: Relatório de visita técnica.....	37
Figura 30 – Escalonamento on-line proposto por HENRIQUES.....	40

Figura 31 – Exemplo de tabela de escalonamento para rede Fieldbus Foundation.....	42
Figura 32 – Exemplo de escalonamento aplicado à rede PROFIBUS, com quatro mestres. ...	43
Figura 33 – Exemplo de modelagem para as redes Ethernet industriais utilizando redes de Petri.....	43
Figura 34 – Exemplo da situação evidenciada pelo trabalho de CARVALHO.	44
Figura 35 – Comparação entre UDP e TCP para Ethernet industrial.....	47
Figura 36 – Conceito de não preempção para as mensagens da rede proposta.	48
Figura 37 – Ilustração do escalonamento por priorização dos grupos de mensagens.	51
Figura 38 – Ordem de execução das mensagens escalonadas após a execução do escalonamento.	53
Figura 39 - Fluxograma do escalonador do barramento para as mensagens cíclicas.....	54
Figura 40 – Ilustração para a representação de um escalonamento alternativo.....	55
Figura 41 - Fluxograma do escalonador para cada módulo de campo (escalonamento individual).....	58
Figura 42 – Configuração da parte física para a rede Ethernet proposta.....	63
Figura 43 – Diagrama de hierarquia de classes.	66
Figura 44 – Alguns tratamentos de exceções da ferramenta computacional.....	67
Figura 45 – Configuração física para o primeiro cenário.....	70
Figura 46 – Configuração dos módulos de campo: Módulo_0 à Módulo_4, cenário 1.	72
Figura 47 – Tabela de escalonamento para o primeiro cenário.....	73
Figura 48 – Ilustração do escalonamento realizado em cada módulo de campo, cenário 1.....	74
Figura 49 – Análise temporal total cíclica para o primeiro cenário.	76
Figura 50 – Configuração errônea de algum parâmetro do cenário proposto.	77
Figura 51 – Configuração física para o segundo cenário.	82
Figura 52 – Tabela de escalonamento para o segundo cenário.	84
Figura 53 – Análise temporal total para o segundo cenário.	86
Figura 54 – Módulo de campo real para validação da proposta do trabalho.....	89
Figura 55 – Análise temporal para o cenário 3 utilizando a ferramenta Wireshark.....	91
Figura 56 – Controle distribuído na rede Ethernet para o cenário 3.....	92
Figura 57 – Ilustração do escalonamento realizado em cada módulo de campo, cenário 3.....	93
Figura 58 – Tela do controlador mestre Ethernet/IP da ODVA.....	94
Figura 59 – Tela do módulo de campo Ethernet/IP da ODVA.	94
Figura 60 – Tabela de escalonamento e gráfico para o cenário 4.....	96

SUMÁRIO

1	INTRODUÇÃO.....	1
1.1	A evolução da Ethernet industrial.....	2
1.2	Características do padrão TCP/IP no meio industrial.....	4
1.3	A suíte de protocolos TCP/IP – divisão em camadas.....	8
1.3.1	Divisão de camadas na suíte de protocolos TCP/IP.....	8
1.3.2	Camada de Rede.....	10
1.3.3	Camada de Inter-rede.....	11
1.3.4	Camada de Transporte.....	12
2	ETHERNET INDUSTRIAL: PROTOCOLOS.....	14
2.1	PROFINET.....	14
2.2	Ethernet/IP.....	19
2.2.1	O meio físico para a rede Ethernet/IP.....	22
2.3	Outros protocolos para Ethernet industrial.....	24
2.3.1	HSE – High Speed Ethernet.....	24
2.3.2	Sercos III.....	25
2.3.3	Modbus TCP.....	26
2.3.4	EtherCAT.....	26
2.3.5	EPA.....	27
2.3.6	EPL.....	27
2.3.7	P-Net on IP.....	28
2.3.8	Vnet/IP.....	28
2.3.9	SynqNet.....	28
2.3.10	IEEE 61850.....	28
2.3.11	JetSync e TCnet.....	29
2.4	Comparativo entre os diversos protocolos para Ethernet industrial.....	29
3	CONCEITOS DE ESCALONAMENTO DE MENSAGENS.....	31
4	NECESSIDADE DE UMA ARQUITETURA DISTRIBUÍDA.....	36
4.1	Trabalhos já realizados.....	38
4.2	Tempos envolvidos nas redes Ethernet industriais.....	45
5	SOLUÇÃO PROPOSTA.....	47
5.1	Justificativa para utilização da técnica de escalonamento.....	47
5.2	Modelagem da solução proposta.....	51

5.2.1	Equações propostas para a realização do controle distribuído	58
5.3	Análise da parte física da rede Ethernet industrial	61
5.4	Filosofia de desenvolvimento para a ferramenta computacional	63
5.4.1	Funcionalidades necessárias da ferramenta computacional	64
5.4.2	Diagramas de classes chaves da ferramenta computacional	65
5.4.3	Tratamento de exceções da ferramenta proposta ao nível do projetista	66
6	RESULTADOS	68
6.1	Resultados Obtidos	68
6.1.1	Primeiro cenário	68
6.1.2	Segundo cenário	77
6.1.3	Terceiro cenário	89
6.1.4	Quarto cenário	93
6.2	Conclusões dos resultados	96
6.3	Dificuldades encontradas no desenvolvimento do trabalho	98
7	CONCLUSÕES E CONTRIBUIÇÕES.....	99
	REFERÊNCIAS BIBLIOGRÁFICAS	101
	APÊNDICES.....	108
	A – Codificação das classes da solução proposta.....	108
	B – Documentação UML da solução proposta.....	116
	C – Código para controle distribuído.....	118

1 INTRODUÇÃO

O protocolo de interconexão de rede, padrão internet, é uma das arquiteturas em redes mais difundidas para comunicação entre sistemas computacionais dispostos a média e longa distância. Surgiu em meados dos anos 70, criado por Vinton G. Cerf e Robert E. Khan ^[1], não parando mais de crescer.

O protocolo Ethernet é um dos protocolos que faz parte da suíte de protocolos TCP/IP, sendo utilizado nas camadas de enlace e física, criado por Robert M. Metcalfe, também na década de 70, para auxiliar a conexão física entre os sistemas computacionais. ^[2]

Esta tese aborda mecanismos da tecnologia de comunicação para sistemas distribuídos em ambientes industriais, denominada por “Ethernet industrial”, que utiliza como padronização a suíte de protocolos TCP/IP. Além disso, procura desenvolver um novo método de comunicação, utilizando um controle distribuído da comunicação do barramento.

O objetivo deste trabalho é desenvolver um novo conceito de comunicação, aplicado as redes Ethernet industriais, com o controle da comunicação da rede e de cada elemento de campo distribuído ao longo do processo industrial. Para isso, é necessário modelar o novo conceito de comunicação, realizar o escalonamento (que é essencial para definir a ordem de comunicação das mensagens trafegadas na rede devido a não utilização de um controlador mestre centralizado) e realizar os testes de verificação e validação da idéia proposta.

Uma ferramenta computacional para simulação desse novo método foi construída para verificação do funcionamento da comunicação sem a utilização do controlador mestre. Para isso, foi necessário o uso de escalonamento de mensagens e busca pelo início de processamento para uma determinada lógica de controle. Estas funções, descritas acima, foram realizadas sem um único elemento na rede, apenas utilizando recursos computacionais de simulações para verificação do método de comunicação proposto e serão detalhadas adiante.

A metodologia de desenvolvimento do trabalho foi: revisão bibliográfica do tema (capítulos um, dois e três), descrição do problema e revisão de outros trabalhos já realizados (capítulo quatro), desenvolvimento e explanação da solução proposta para o problema (capítulo cinco), resultados obtidos, simulações realizadas e conclusão (capítulo seis e sete) e levantamento das referências bibliográficas.

Nos capítulos cinco, seis e sete será evidenciada a solução para a proposta acima, além do desenvolvimento de uma ferramenta computacional para realizar a configuração, a verificação e a validação da idéia proposta no trabalho.

1.1 A evolução da Ethernet industrial

Nos anos 40, a instrumentação de processo operava com sinais de pressão de 3–15 psi para monitorar os dispositivos de controle no chão-de-fábrica. ^[4]

Já nos anos 60, os sinais analógicos de 4-20mA foram introduzidos na indústria para medição e monitoramento de dispositivos. ^[4]

Com o desenvolvimento de processadores nos anos 70, surgiu a idéia de utilizar computadores para monitoramento de processos e fazer o controle de um ponto central. Com o uso de computadores, várias etapas do controle puderam ser feitas de diferentes formas, umas das outras, de modo a se adaptarem mais precisamente às necessidades de cada processo. ^[4]

Nos anos 80, iniciou-se o desenvolvimento dos primeiros sensores inteligentes, bem como os controles digitais associados a esses sensores. Tendo-se os instrumentos digitais, era necessário algo que pudesse interligá-los. Então, nasceu a idéia de uma rede que unisse todos os dispositivos e disponibilizasse todos os sinais do processo num mesmo meio físico. A partir daí, a necessidade de uma rede (*fieldbus*) era clara, assim como um padrão que pudesse deixá-lo compatível com o controle de instrumentos inteligentes. ^[4]

Os primeiros *fieldbuses* surgiram como algo inovador na indústria de processo e de manufatura, durante a década de 80. A “guerra entre os *fieldbuses*”, episódio conhecido pela tentativa de padronização de um único protocolo industrial, criou um único padrão proposto pela ISA, denominado de SP-50. ^{[5] [26]} No entanto, criaram vários protocolos, de diferentes associações e instituições, ao redor do mundo. Eles prometiam acabar com a complexidade de cabos existentes, além de oferecerem outras funções para diagnosticar possíveis falhas na rede e realizar atualizações do sistema de controle. ^{[3] [4] [5]}

Neste período inicial de definição das redes *fieldbuses*, a rede Ethernet estava presente no escritório e em residências, mas mostrava sinais de que iria chegar também ao nível de fábrica. No final da década de 90, as primeiras soluções para Ethernet industrial foram desenvolvidas, utilizando o padrão HSE, Ethernet/IP e PROFINET. ^[5]

No início das soluções Ethernet no ambiente industrial, diversos problemas foram detectados para a aplicação direta no chão-de-fábrica. Mas após um período de

aperfeiçoamento e novos desenvolvimentos, a Ethernet pôde ser considerada confiável para este ambiente hostil, devido a robustez dos equipamentos e ao escalonamento determinismo das mensagens empregado na concepção da rede. ^{[4] [5]}

A suíte de protocolos TCP/IP surgiu no meio industrial há cerca de dez anos. Porém, somente nos últimos cinco anos, é que se tornou comercialmente utilizada em ambientes industriais. ^{[4] [12]}

A busca pela definição de um padrão internacional levou vários grupos a se unirem. Entre eles: a *International Society of Automation* (ISA) ^[6], a *International Electrotechnical Commission* (IEC) ^[7], o comitê de padronização do PROFIBUS (norma alemã) ^[8] e o comitê de padronização do FIP (norma francesa). ^[9] Formou-se, assim, o comitê IEC/ISA SP-50 Fieldbus. ^{[4] [5]}

Foram necessários muitos anos para que o desenvolvimento deste padrão internacional se estabelecesse. Em 2000, todas as organizações interessadas convergiram para criar o *fieldbus* padrão IEC, que foi denominado IEC 61158 ^[7], contendo oito padrões diferentes, denominados de "*profiles*":

- Tipo 1 — Foundation Fieldbus - H1
- Tipo 2 — ControlNet
- Tipo 3 — PROFIBUS
- Tipo 4 — P - Net
- Tipo 5 — FOUNDATION Fieldbus HSE (High Speed Ethernet)
- Tipo 6 — Interbus
- Tipo 7 — SwiftNet
- Tipo 8 — WorldFIP

Mesmo com todos esses tipos acima citados, não foi possível abranger todas as aplicações na indústria. Mais tarde, então, foi criada a IEC 61784 ^{[7] [10]} como uma definição dos chamados “perfis” e ,ao mesmo tempo, foram corrigidas as especificações de IEC 61158.

A tabela 1 mostra os padrões com os seus respectivos perfis de comunicação.

Tabela 1 – Padrões e protocolos de acordo com a IEC 61784 e IEC 61158. ^[5]

IEC 61784	IEC 61158 - PROTOCOLOS		
	MEIO FISICO	CAMADA	
CPF-1/1	TIPO 1	TIPO 1	FOUNDATION FIELDBUS (H1)
CPF-1/2	ETHERNET	TCP/UDP/IP	FOUNDATION FIELDBUS (HSE)
CPF-1/3	TIPO 1	TIPO 1	FOUNDATION FIELDBUS (H2)
CPF-2/1	TIPO 2	TIPO 2	CONTROLNET
CPF-2/2	ETHERNET	TCP/UDP/IP	ETHERNET/IP
CPF-3/1	TIPO 3	TIPO 3	PROFIBUS-DP
CPF-3/2	TIPO 1	TIPO 3	PROFIBUS-PA
CPF-3/3	ETHERNET	TCP/UDP/IP	PROFINET
CPF-4/1	TIPO 4	TIPO 4	P-NET RS-485
CPF-4/1	TIPO 4	TIPO 4	P-NET RS-232
CPF-5/1	TIPO 1	TIPO 7	WORLDFIP(MPS,MCS)
CPF-5/2	TIPO 1	TIPO 7	WORLDFIP(MPS,MCS, SubMMS)
CPF-5/3	TIPO 1	TIPO 7	WORLDFIP(MPS)
CPF-6/1	TIPO 8	TIPO 8	INTERBUS
CPF-6/2	TIPO 8	TIPO 8	INTERBUS TCP/IP
CPF-6/3	TIPO 8	TIPO 8	INTERBUS SUBSET
CPF-7/1	TIPO 6	TIPO 6	SWIFTNET TRANSPORT
CPF-7/2	TIPO 6	TIPO 6	SWIFTNET FULL STACK

Como pode ser observado na tabela 1, vários protocolos de Ethernet já foram incluídos. Estes protocolos utilizam o meio físico tipo Ethernet bem como os protocolos IP, TCP e UDP nas camadas superiores de comunicação. ^{[5][7]}

A denominação CPF-X/X (*Communication Profile Families*), da tabela 1, indica o tipo da família de um determinado protocolo e suas versões, como por exemplo: PROFIBUS tipo DP e PROFIBUS tipo PA. Ambos são pertencentes à família três, porém com versões distintas. ^{[5][7]}

1.2 Características do padrão TCP/IP no meio industrial

A motivação para inserção da suíte de protocolos TCP/IP no meio industrial foi a tentativa de padronização da Cadeia de Suprimentos em torno de um único tipo de tecnologia em redes industriais. ^{[4][5]}

Todavia, para a aplicação da suíte de protocolos TCP/IP no ambiente industrial, diversos fatores devem ser considerados e avaliados como motivacionais para a sua utilização, tais como:

- Padronização dos diversos protocolos do mercado. Isso acabou não ocorrendo devido a utilização de hardwares específicos por parte de alguns fabricantes e também a aplicações específicas de cada rede; ^[4]
- Interoperabilidade entre os diversos equipamentos dos diferentes fabricantes; ^[11]
- Aumento do volume de mensagens trafegadas pela rede (até mil quatrocentos e setenta e dois bytes de informação útil) e diminuição do tempo de ciclo (algumas redes podem operar com ciclos de algumas dezenas de microssegundos) em função das características dinâmicas do processo a controlar; ^[12]
- Robustez dos equipamentos quando inseridos em ambientes hostis. Isso se refere a temperatura de operação, classe de proteção, segurança intrínseca, umidade relativa do ar, etc.

A seguir, pode-se ver um exemplo de um conector RJ 45 industrial, que opera de -40°C a +80°C, com fator de proteção IP67 e um cabo metálico, blindado, contendo de dois a quatro pares trançados, utilizados para os padrões Ethernet industriais (quatro e oito vias, ambos com blindagem); ^[8]

A figura 1 e a figura 2 ilustram alguns elementos típicos em uma rede Ethernet industrial.



Figura 1 – Conectores RJ 45 para as redes Ethernet industriais. ^[8]

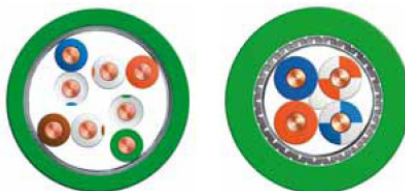


Figura 2 – Cabo blindado para as redes Ethernet industriais. ^[8]

Com relação ao determinismo da rede Ethernet no meio industrial, a rede Ethernet utiliza o protocolo de gerenciamento de tráfego tipo CSMA/CD (*Carrier Sense with Multiple Access and Collision Detect*), com o objetivo de evitar colisões de mensagens trafegadas na rede entre duas ou mais estações ativas. Essa técnica consiste na detecção de colisão no meio de transmissão, ou seja, quando mais de uma estação deseja, ao mesmo tempo, transmitir

dados no barramento. ^[13] Neste caso, apenas uma delas conseguirá transmiti-los, sendo que a outra terá que esperar um tempo aleatório gerado pelo controlador da rede até tentar outra transmissão. Caso não consiga, o dispositivo esperará mais uma vez e assim sucessivamente. Nota-se que não há certeza de que a informação será transmitida num tempo determinado (o que é essencial para o ambiente industrial). Em um ambiente industrial, a espera de uma informação, como, por exemplo, um alarme de incêndio ou a detecção de um objeto, pode ser crítica. Não se pode esperar um longo tempo para a informação crítica ser transmitida. ^[14] Logo, para a aplicação desses conceitos acima no meio industrial, é necessário transformar a suíte de protocolos TCP/IP comum (não determinística), em determinística. Para isso, utiliza-se um *switch* especial, com portas independentes entre si e com capacidade de programar prioridades e tempos de espera das mensagens. Deste modo, o *switch* determina o tempo de varredura para cada porta. ^[12]

A figura 3 ilustra um *switch* especial para rede Ethernet industrial adequado para aplicações em ambientes industriais onde os requisitos temporais são predominantes na maioria dos casos.

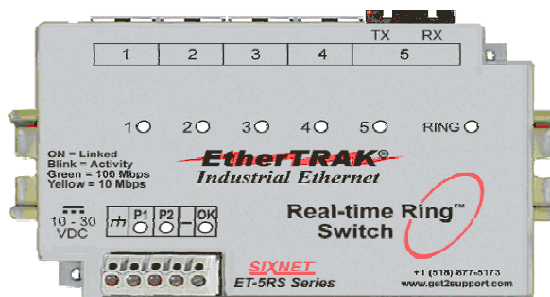


Figura 3 – *Switch* Ethernet industrial com prioridade de transmissão. ^[18]

No início, a grande desvantagem deste padrão na área industrial foi a questão da comunicação e alimentação dos módulos. ^[15] ^[16] Havia a necessidade de se ter dois cabos separados para um elemento da rede (comunicação de informação e alimentação). ^[16] ^[17] O padrão utilizado para sobrepor o sinal de comunicação e alimentação, em um único cabo, é chamado de *Power over Ethernet (PoE)*. ^[15] A solução PoE é adequada e suficiente para a redução da quantidade de cabos a serem utilizados e, conseqüentemente, tem-se os sinais de comunicação e alimentação trafegando sob o mesmo barramento. Atualmente, há várias aplicações e contínuos estudos na melhoria desse padrão no mercado. Nesse tipo de comunicação, o canal transmissor e receptor pode trafegar mensagens simultaneamente no meio de transmissão, utilizando o conceito de modulação em amplitude sobreposto ao nível contínuo de alimentação dos módulos de campo. Assim, o sinal de comunicação sofreria uma modulação para ser transmitido ou recebido por um elemento da rede. A norma *IEEE 802.3af*

[18] regulamenta todo o conceito de PoE. [16] Esta norma descreve que a alimentação deve estar entre 20Vdc a 57Vdc (48Vdc é a tensão nominal) e a potência do sinal deve ser, no máximo, de 16,8W (máximo de 350mA para a tensão de alimentação de 44Vdc, $0,35A \cdot 48V = 16,8W$). [17]

As figuras 4 e 5 ilustram os tipos de comunicação PoE possíveis. A figura 4 ilustra a ligação PoE utilizando os quatro pares trançados disponíveis no cabo Ethernet da figura 2, operando no modo *Full duplex* (todos os pinos do equipamento alimentado são utilizados - comunicação bidirecional e simultânea). Já a figura 5, ilustra o cabo contendo apenas dois pares trançados, onde a comunicação está sobreposta a alimentação, operando no modo *Half Duplex* (pinos 4 e 5 do equipamento alimentado estão abertos - comunicação bidirecional e intercalada).

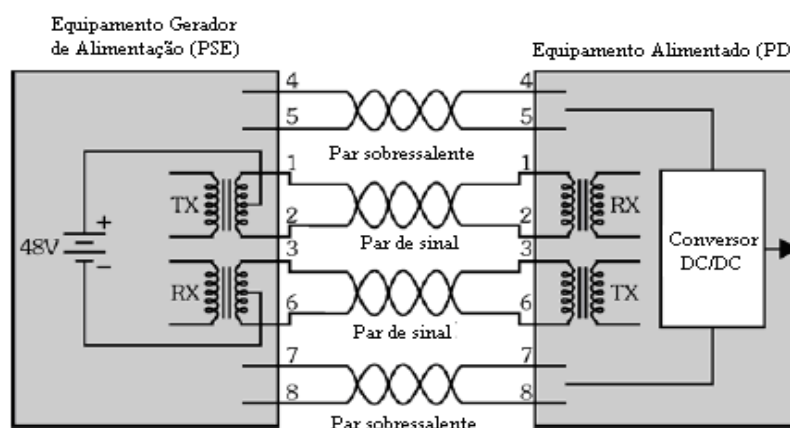


Figura 4 – Configuração para sistema PoE – *Full duplex*. [17]

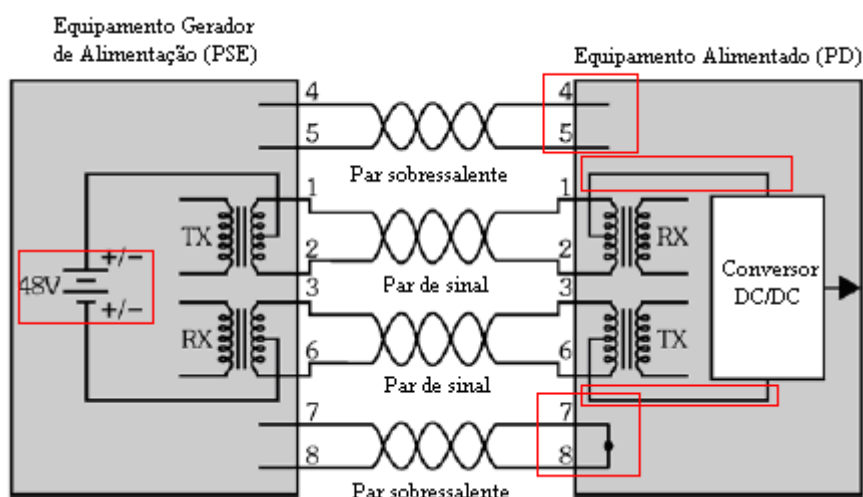


Figura 5 – Configuração para sistema PoE – *Half duplex*. [17]

O grande objetivo para o uso desta tecnologia é a comunicação *Full-duplex*, onde se tem um menor custo de implementação da rede (somente um cabo, com quatro pares de fios),

pois ocorre comunicação simultânea nos dois sentidos, recepção e transmissão, tendo a alimentação dos módulos inserida sobre o sinal de comunicação. ^[15] Na figura 4, a configuração é para a operação no modo *Full-duplex*. Nesse modo, os canais de transmissão e recepção estão separados fisicamente no cabo da rede (dois pares) e a alimentação de 48Vdc em outros dois pares. ^{[16] [17]} Normalmente, não se utiliza tensão de 48Vdc na automação, sendo mais comum o uso de tensões de 24Vdc. Assim, é necessário o uso de um conversor DC/DC para transformar o sinal de 48Vdc para outro qualquer. A grande desvantagem dessa configuração é a necessidade de polaridade nos terminais de alimentação, a fim de diferenciar os sinais em modulação e em amplitude, positivos e negativos. ^[17]

Na figura 5, a configuração é para a operação no modo *Half-duplex* onde a comunicação é feita em dois sentidos, recepção e transmissão, mas não simultaneamente. Nesse modo, o cabo é comum tanto para a alimentação quanto para a comunicação, não necessitando de uma polaridade específica nos terminais. É necessária uma modulação em amplitude sobre a alimentação para não interferir no sinal de comunicação trafegado no cabo. ^[17] O sinal de comunicação trafega apenas em um sentido, devido ao mesmo meio transmitir e receber o sinal. O grande problema dessa configuração é garantir o isolamento de tensão mínimo requerido entre os sinais de alimentação e comunicação. ^[17]

1.3 A suíte de protocolos TCP/IP – divisão em camadas

A suíte de protocolos TCP/IP desenvolvida por Cerf e Khan tinha, inicialmente, o objetivo de interligar as diferentes tecnologias de redes existentes no mundo, seja de altas e baixas velocidades ou grandes e pequenas distâncias. ^{[13] [19]} A regulamentação de toda a suíte de protocolos TCP/IP (protocolos, aplicativos e funcionalidades) está descrita em documentos chamados de RFC (*Request for Comments*) que podem ser acessados por qualquer pessoa via Internet. ^[20]

1.3.1 Divisão de camadas na suíte de protocolos TCP/IP

Para entender melhor o funcionamento, a interoperabilidade e o objetivo de cada protocolo de rede, a ISO (*International Standards Organization*) padronizou várias camadas e interligações entre elas. Esse modelo se chamou OSI (*Open System Interconnection*). ^[19]

A figura 6 mostra todas as sete camadas propostas pelo modelo OSI. O uso de todas as camadas não é obrigatório aos sistemas de comunicações, mas sua hierarquia e protocolos devem ser mantidos. ^{[19][20]}

No início da criação das redes via satélite ou rádio, o modelo OSI tornou-se pouco eficiente. Isso ocorreu porque os dados de informação eram muito menores do que os dados de controle da rede. Logo, gastavam-se muitos bits de controle para poucos bits de dados efetivamente úteis. Esta arquitetura também não previa outros tipos de serviços, como transmissão de voz e imagem pela rede. ^{[19][20]}

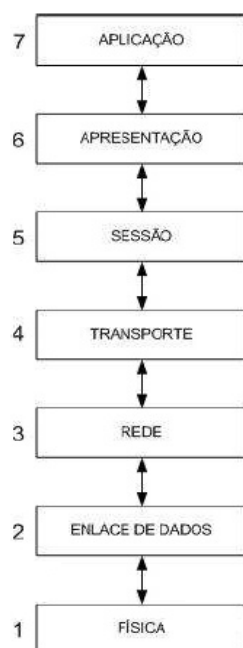


Figura 6 – Modelo OSI da ISO. ^[19]

Com todas essas lacunas, Cerf e Khan desenvolveram outra arquitetura mais eficiente, o modelo TCP/IP, cujo objetivo inicial era interconectar, ao mesmo tempo, várias redes espalhadas pelo mundo, podendo fornecer serviços de dados, voz e imagem. ^{[1][11]}

A figura 7 ilustra a divisão da suíte de protocolos TCP/IP, camada a camada, também conhecida como modelo híbrido.

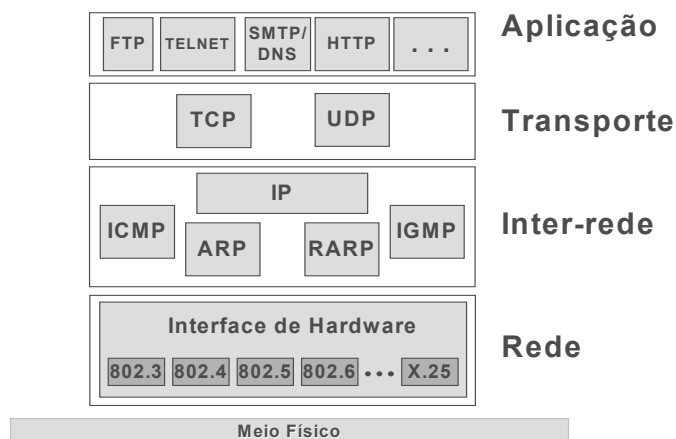


Figura 7 – Suíte de protocolos TCP/IP. ^[19]

A camada um, denominada Rede, define qual é o meio físico e o tipo de enlace utilizados na suíte TCP/IP para comunicação entre dois ou mais dispositivos computacionais. Vários protocolos e padrões foram desenvolvidos e certificados nessa camada, sendo que o padrão Ethernet tornou-se o mais difundido e aplicado nessa camada. ^[19]

A camada dois, denominada Inter-redes, é responsável pelo encaminhamento dos pacotes em qualquer rede, garantindo que sejam transmitidos independentemente do destino. O protocolo mais difundido dessa camada é o IP, porém há outros que auxiliam seu trabalho. ^[19]

A camada três, denominada transporte, é responsável pela conexão entre dois ou mais elementos, além de garantir a transmissão das mensagens. Dois protocolos operam nessa camada, o TCP, que opera orientado a conexão, ou o UDP que não opera orientado a conexão. ^[19]

A camada quatro, denominada aplicação do projetista, contém protocolos de alto nível e que são acessados diretamente pelo projetista do sistema. Esses protocolos provêm serviços ao projetista, como o HTTP, FTP, WWW e SNMP. ^[19]

1.3.2 Camada de Rede

A camada de Rede é a primeira camada da suíte de protocolos TCP/IP. Ela é responsável por toda a interconexão do meio físico com a rede e pela adequação das mensagens na mesma (camada de enlace). ^{[1][21]}

O IEEE produziu várias normas para esta camada. O conjunto de normas ficou conhecido como IEEE 802. Dentro desse conjunto, várias outras normas foram desenvolvidas com base no meio físico e no tipo de enlace desejado. [22]

A indústria optou pelos padrões IEEE 802.3, chamados de Ethernet. [23] Nesse padrão, o acesso ao meio é realizado pelo protocolo CSMA/CD, porém o meio físico pode ter várias maneiras de ser implementado como: fibra óptica, par trançado e cabo coaxial, por exemplo. Para os protocolos dessa camada são importantes o meio físico, o acesso ao meio de transmissão, a estrutura do pacote de mensagens e os elementos que realizam o controle da transmissão das mensagens. [13]

A figura 8 ilustra o quadro de um dos protocolos da camada de Enlace (de acordo com o modelo OSI), o Ethernet, que é utilizado na indústria. Ele possui um total de oito bytes de preâmbulo, quatorze bytes de cabeçalho, quarenta e seis a mil e quinhentos bytes de dados e quatro bytes de verificação de erro. [12]

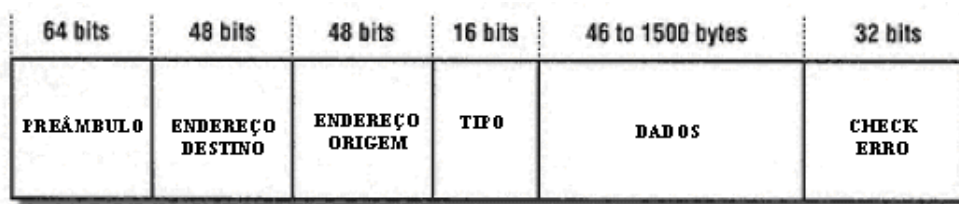


Figura 8 – Quadro Ethernet. [12]

1.3.3 Camada de Inter-rede

A camada de Inter-rede é responsável pela transmissão das mensagens de uma origem até um destino através da computação de pacotes, podendo ser da mesma rede ou não. Assim, essa camada tem a função de mover dados entre uma mesma rede (Intranets) ou em redes diferentes (Internet). [13] [22]

O principal protocolo que está nessa camada é o IP. Sua tarefa é fornecer a melhor forma de transportar as mensagens da origem até o destino, sem se preocupar com a confirmação de recebimento, chamada de serviço não orientado a conexão. Assim, a garantia de entrega das mensagens é realizada pelas camadas superiores. Além dele, outros protocolos auxiliam no controle de fluxo ou no endereçamento dos quadros trafegados pela rede como: ARP e RARP (utilizados em conjunto com o protocolo de rede Ethernet para identificar os endereços de rede), ICMP (protocolo de controle do fluxo das mensagens) e IGMP (protocolo de controle do roteamento das mensagens). [19]

O quadro IP utiliza o endereçamento no formato decimal, como por exemplo, 192.168.10.2, para a representação dos quatro bytes de endereço entre a rede e o *host* (elemento conectado a rede, como por exemplo, um computador). O quadro consiste de duas partes: o cabeçalho e a informação. O cabeçalho possui uma parte fixa de vinte bytes e uma parte opcional de tamanho variável de quatro bytes (total de vinte e quatro bytes de cabeçalho). O restante dos bytes são as informações a serem trafegadas na rede. No cabeçalho, é possível determinar o endereço de origem e destino, o comprimento e verificação de erro das mensagens e o controle dessas informações, entre outros serviços.

A figura 9 ilustra as funcionalidades do protocolo IP. ^[19]

+	0 - 3	4 - 7	8 - 15	16 - 18	19 - 31
0	Versão	Tamanho do cabeçalho	Tipo de Serviço (ToS) (agora DiffServ e ECN)	Comprimento (pacote)	
32	Identificador			Flags	Offset
64	Tempo de Vida (TTL)		Protocolo	Checksum	
96	Endereço origem				
128	Endereço destino				
160	Opções				
192	Dados				

Figura 9 – Quadro IP. ^[19]

1.3.4 Camada de Transporte

Nessa camada existem, basicamente, dois protocolos que são utilizados: o UDP (não orientado a conexão) e o TCP (orientado a conexão). A principal função desta camada é estabelecer uma conexão entre as duas portas que irão trocar informações. ^[24] Para a indústria, ambos os protocolos podem ser utilizados, dependendo da aplicação e dos requisitos de urgência de transmissão de uma determinada informação. ^{[11] [25]} O TCP é utilizado para configuração e parametrização dos elementos da rede, devido a sua lentidão e confirmação de envio da mensagem (comunicação acíclica). Já o UDP é utilizado para troca de mensagens entre os elementos da rede (comunicação cíclica), devido a sua rapidez no transporte das mensagens. ^{[11] [25]}

O TCP opera no modo orientado a conexão, ou seja, porta de destino e porta de origem, contendo verificação de erros, multiplexação de mensagens (pode transmitir várias

mensagens de diferentes mensagens ao mesmo tempo) e controle de fluxo (ajusta o tempo de transmissão de acordo com a quantidade de mensagens). O UDP opera no modo sem conexão, não havendo confirmação de recebimento da mensagem e contendo oito bytes de cabeçalho. [24]

As figuras 10 e 11 ilustram os quadros dos protocolos de transporte TCP e UDP, respectivamente.

+	Bits 0 - 3	4 - 9	10 - 15	16 - 31
0	Porta na origem		Porta no destino	
32	Número de sequência			
64	Número de confirmação (ACK)			
96	Offset	Reservados	Flags	Janela Window
128	Checksum		Ponteiro de urgência	
160	Opções (opcional)			
	Padding (até 32)			
224	Dados			

Figura 10 – Quadro do protocolo TCP. [19]

Bit 0	31	
Porta origem	Porta destino	4 bytes
Comprimento da mensagem	Checksum	4 bytes

Figura 11 – Quadro do protocolo UDP. [19]

2 ETHERNET INDUSTRIAL: PROTOCOLOS

Vários fabricantes criaram seus próprios padrões de redes Ethernet industriais, cada um diferindo do outro em termos de aplicação ao projetista e na utilização ou não de determinadas camadas do padrão TCP/IP. [26] Alguns deles possuem hardwares dedicados do fabricante, o que os tornam pouco competitivos para aplicação industrial. Os quatorze padrões de protocolos (padronizados pela IEC 61784 [7] e IEC 61158 [7] e, alguns, por suas próprias associações [28]), para Ethernet industrial são: PROFINET, Ethernet/IP, HSE, Modbus/TCP, EPA, EPL, EtherCAT, IEC 61850, JetSync, P-Net, Sercos III, SynqNet, TCnet e Vnet/IP. [27] [28] Dentre esses protocolos, destacam-se o PROFINET, Ethernet/IP e HSE que são utilizados na automação de processos e sistemas industriais e terão ênfase maior nesse trabalho. [28]

2.1 PROFINET

O PROFINET é um padrão de automação da associação PROFIBUS internacional para implementação e integração de soluções baseadas em Ethernet industrial. [12] [29] O PROFINET suporta a integração de um simples dispositivo digital de campo a aplicações de tempo crítico (*real time*) complexas, bem como a integração de automação de manufatura baseadas em componentes. [12]

A figura 12 ilustra as características básicas entre as redes PROFIBUS e PROFINET.

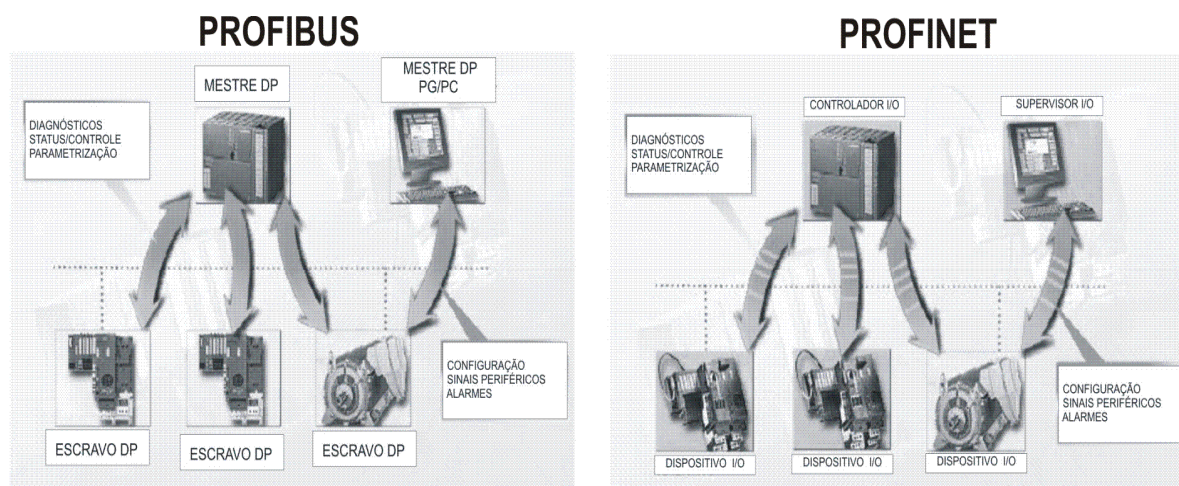


Figura 12 – PROFIBUS x PROFINET. [30]

A figura 12 ilustra duas aplicações distintas: uma aplicação típica envolvendo a rede PROFIBUS DP, contendo o controlador mestre e três escravos, além de uma ferramenta de

configuração, parametrização e diagnóstico da rede e outra aplicação envolvendo a rede PROFINET IO, contendo, também, o controlador mestre e três escravos, além de um sistema supervisorio. ^{[12] [29]}

A rede PROFIBUS possui as seguintes características básicas: ^[29]

- Meio físico serial RS-485.
- Taxa de transmissão configurável de 9,6Kbps até 12Mbps.
- Método de comunicação mestre/escravo, com passagem de prioridade de transmissão (*Token*).
- Distância física de até 1200 metros por segmento de rede, sem a utilização de repetidores.
- Endereços configuráveis através de chaves rotativas fixas em cada elemento da rede, podendo ter até 125 elementos na rede.
- Tipos de PROFIBUS: DP (para processos de manufatura) e PA (para automação de processos contínuos).

A rede PROFINET possui as seguintes características básicas: ^[29]

- Meio físico Ethernet (IEEE 802.3).
- Taxa de transmissão configurável entre 10Mbps, 100Mbps ou 1Gbps.
- Método de comunicação mestre/escravo, com passagem de prioridade de transmissão (*Token*), com determinismo realizado através do *switch* industrial.
- Distância física de até 100 metros por segmento de rede, sem a utilização de repetidores.
- Endereços configuráveis através de software em cada elemento da rede, podendo ter até 256 elementos na rede.
- Possível integração com rede Internet externa.
- Tipos de PROFINET: IO (para comunicação cíclica da rede) e CBA (para conexão dessa rede a outros padrões ou para comunicação acíclica da rede).
- Possibilidade de integração de toda a Cadeia de Suprimentos da automação industrial.

O PROFINET possui três tipos de dispositivo básicos: controlador, módulos de campo e sistema de supervisão. Eles estão descritos a seguir. ^{[12] [29]}

- Controlador: no qual o programa de controle é executado (sistema centralizado).

- Módulo de campo: dispositivo de campo remoto que mantém comunicação com um controlador.
- Sistema de supervisão: dispositivo gráfico programável, que comissiona e tem funções de diagnósticos na rede.

Basicamente, há dois tipos de redes PROFINET: PROFINET IO e PROFINET CBA. O PROFINET IO é utilizado em aplicações com requisitos de tempo real (críticas) e o PROFINET CBA é utilizado em aplicações onde o tempo não é crítico, por exemplo, na conversão para rede PROFIBUS DP. [12] [29]

A figura 13 ilustra como o PROFINET definiu suas camadas baseadas na suíte de protocolos TCP/IP.

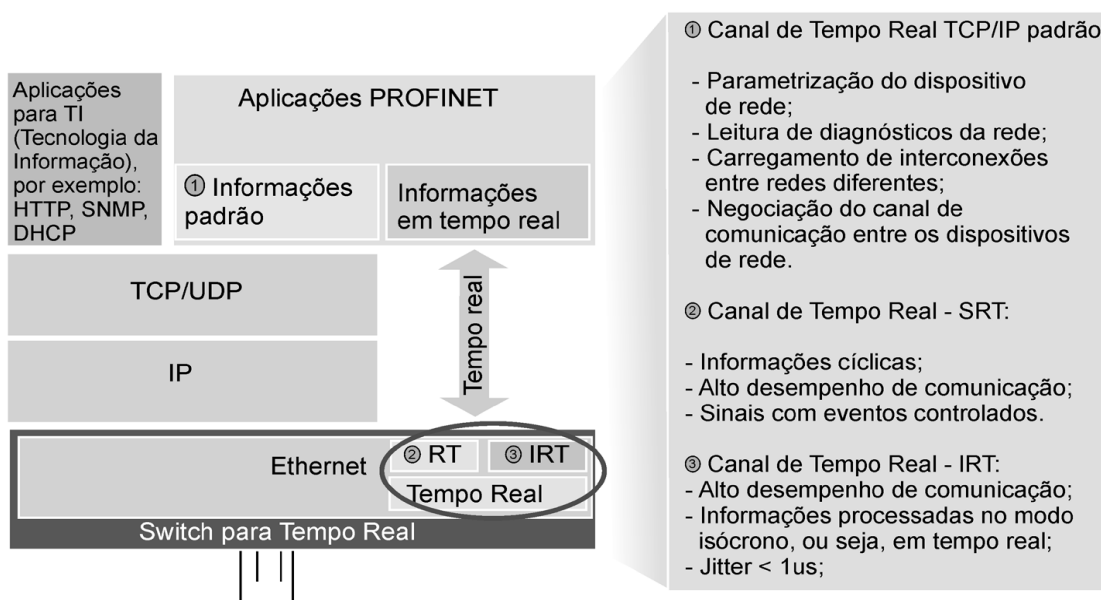


Figura 13 – Suíte de protocolos TCP/IP para PROFINET. [12] [29]

Nota-se pela figura 13 que o PROFINET pode ter três formas distintas de operação, (também conhecidas como canais de comunicação) sendo duas delas para tempo real (*Real Time* – RT e *Isochronous Real Time* – IRT) e uma para tempo não real (suíte de protocolos TCP/IP padrão). [12] [29]

A primeira forma baseia-se na suíte de protocolos TCP/IP pura, utilizando Ethernet na camada física e de enlace, o IP na camada três e o TCP ou UDP na camada quatro. Essa arquitetura é chamada de *Non-real time (Non-RT)*, tendo como característica tempos de varredura da rede acima dos 100ms. [30] [32] A grande aplicação, nesse tipo de comunicação, é na configuração dos elementos da rede ou na comunicação com os *Gateways* (também conhecidos como *Proxis*), utilizando o PROFINET CBA. Os *Gateways* são conversores de um determinado protocolo transformados em outro (por exemplo, de PROFINET para

PROFIBUS DP ou de PROFINET para Interbus S^[32]), conforme pode ser visto na figura 14. [30] [31]

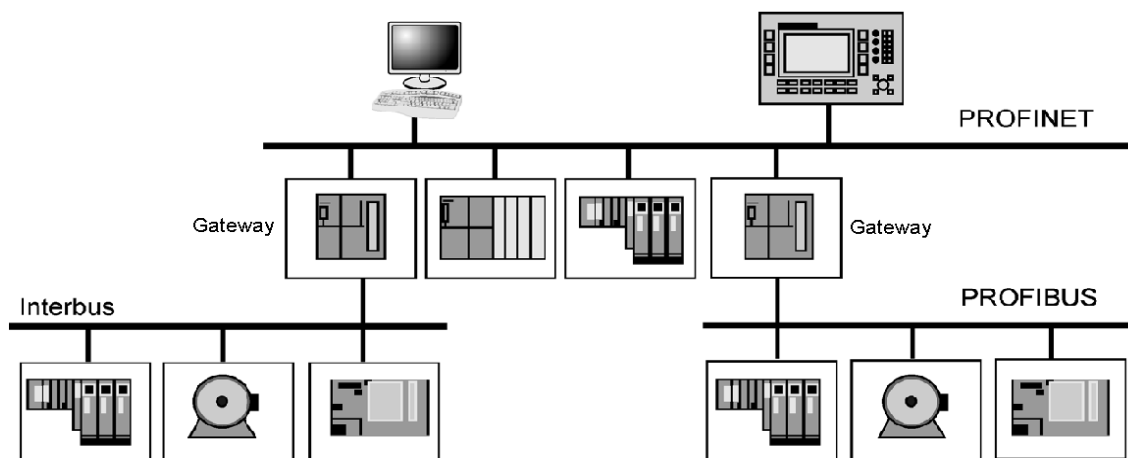


Figura 14– Gateways PROFINET/PROFIBUS DP e PROFINET/Interbus. [30] [31]

A segunda forma baseia-se no chamado *Soft Real Time (RT)*, realizado por um canal que interliga diretamente a camada da Ethernet a camada de Aplicação. Esse tipo de comunicação é aplicada à comunicação cíclica da rede, entre os dispositivos de campo e possui as mesmas funcionalidades da rede PROFIBUS DP. Com a eliminação de vários níveis de protocolo (IP e TCP/UDP), há uma redução no comprimento das mensagens transmitidas e no tempo de processamento, necessitando de menos tempo para transmitir as mensagens na rede. Pode-se utilizar os dois tipos de PROFINET, CBA e IO, nesse caso. [12]

A terceira forma baseia-se no conceito de *Isochronous Real Time (IRT)*, para aplicações em que o tempo de resposta fim-a-fim das mensagens é crítico e deve ser menor do que 1ms. Nesse tipo de comunicação, há um hardware dedicado do fabricante para se obter um tempo de varredura muito pequeno. Uma aplicação típica deste conceito é o controle de movimento de robôs, quando o tempo de atualização das mensagens deve ser pequeno. Utiliza-se apenas o PROFINET IO para esse caso. [12]

A figura 15 ilustra os conceitos do *Non-Real-Time* (aplicações com tempos de varredura em torno de 100ms), *Soft Real Time* (aplicações com tempos de varredura em torno de 10ms) e *Isochronous Real Time* (aplicações com tempos de varredura menores do que 1ms). [12]

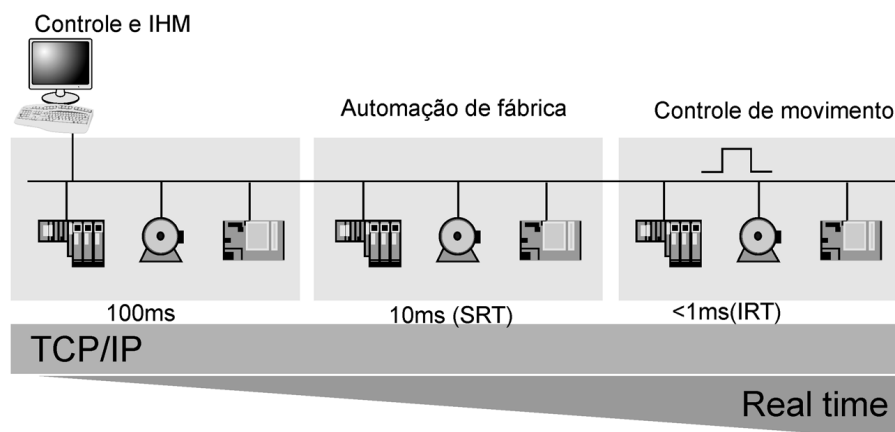


Figura 15 – Níveis de ciclos de comunicação em função dos requisitos temporais da aplicação industrial.

[30]

A rede PROFINET, tipo IO é, na realidade, uma extensão do protocolo PROFIBUS DP. Ela opera diretamente com os elementos de campo, realizando leituras de sensores, atualizações dos sinais de saída e controles de diagnósticos da rede. A rede Ethernet industrial PROFINET IO descreve um modelo de dispositivo que é baseado em características essenciais do PROFIBUS DP, incluindo canais para cada elemento alocado na rede. [32] As características dos dispositivos de campo são descritas via GSD em uma base XML. [32][33]

O arquivo GSDML (GSD em uma base de programação XML) acompanha todos os elementos da rede PROFINET IO e descreve as principais características de cada um deles. Ele deve ser fornecido pelo fabricante do equipamento. [12]

O quadro da rede PROFINET IO é mostrado na figura 16 e segue a mesma estrutura do campo Ethernet citado anteriormente no capítulo um. A grande diferença está nos campos *Ether-Type* e *VLAN*. Nesses campos, é possível distinguir se o quadro é IRT, RT ou *Non-RT*, além de definir mensagens com acesso prioritário na rede, de acordo com os quatro bytes dos campos. Cada valor do campo *Ether-Type* define a maneira como as mensagens serão trafegadas na rede, por exemplo: de 0000h a 00FFh, quadro IRT, de 8000h a FBFFh, quadro RT e de FC00h a FE02h, quadro *Non-RT*. A figura 16 ilustra todo o formato do quadro de comunicação para a rede PROFINET IO. [12][32]

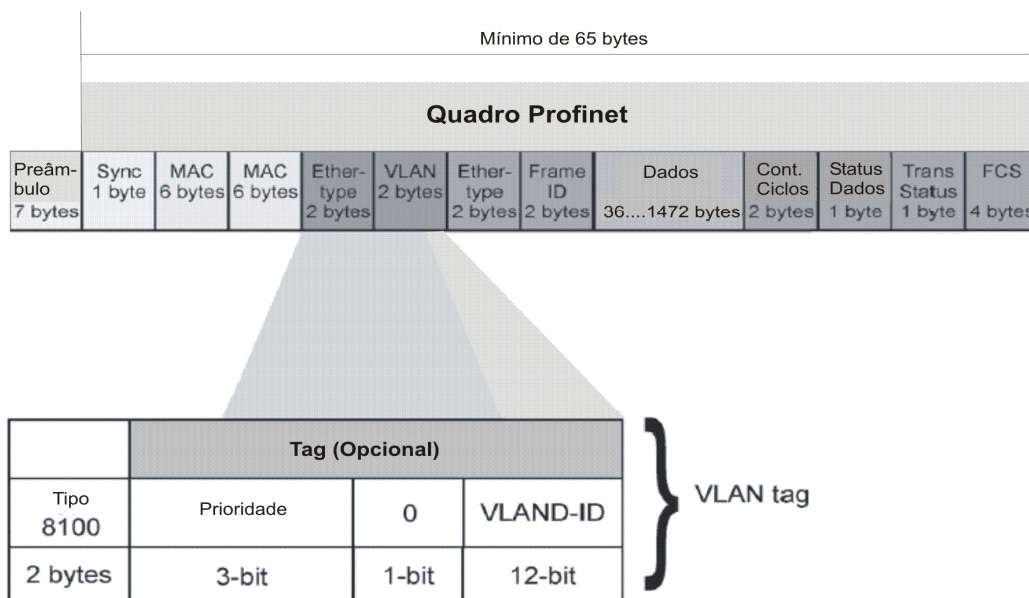


Figura 16 – Quadro PROFINET IO. ^{[12] [32]}

Outro ponto a ser destacado é o atraso da rede com relação ao *switch*. O *switch* gera um pequeno atraso na rede, de 9,6µs e pode causar alguns problemas em aplicações em RT (*Real Time*). Este assunto será discutido mais adiante. ^{[34] [35]}

É importante levar em consideração esse tempo na hora de calcular o ciclo de varredura da rede ou o atraso que um elemento pode causar se houver muitos *switches* na mesma. ^{[34] [35]}

2.2 Ethernet/IP

O Ethernet/IP é uma rede Ethernet industrial padronizada pela ODVA ^[36] e baseia-se na arquitetura encapsulada do TCP/IP. O termo IP significa “*Industrial Protocol*” e não deve ser confundido com o protocolo IP da suíte de protocolos TCP/IP. ^[11]

Enquanto o PROFINET possui três tipos de comunicação e duas redes distintas (PROFINET IO e PROFINET CBA), o Ethernet/IP segue exatamente o modelo TCP/IP, possuindo somente o tipo de comunicação padrão para o quadro Ethernet, já apresentado no capítulo um. ^[25]

Porém, é válido ressaltar que há dois tipos diferentes de operação. Uma delas é o TCP (orientado a conexão) utilizado na camada de transporte já definida anteriormente. O outro método é utilizar o UDP (que possui um quadro menor e não é orientado a conexão) para transportar as mensagens até a CIP (*Control Information Protocol*), uma camada de aplicação

específica da ODVA. Na CIP há protocolos específicos para inversores de frequência, posicionadores de válvulas, elementos pneumáticos, I/O's discretos, entre outros, de acordo com os requisitos dos equipamentos inseridos na rede.

A figura 17 ilustra a divisão em camadas da rede Ethernet/IP.

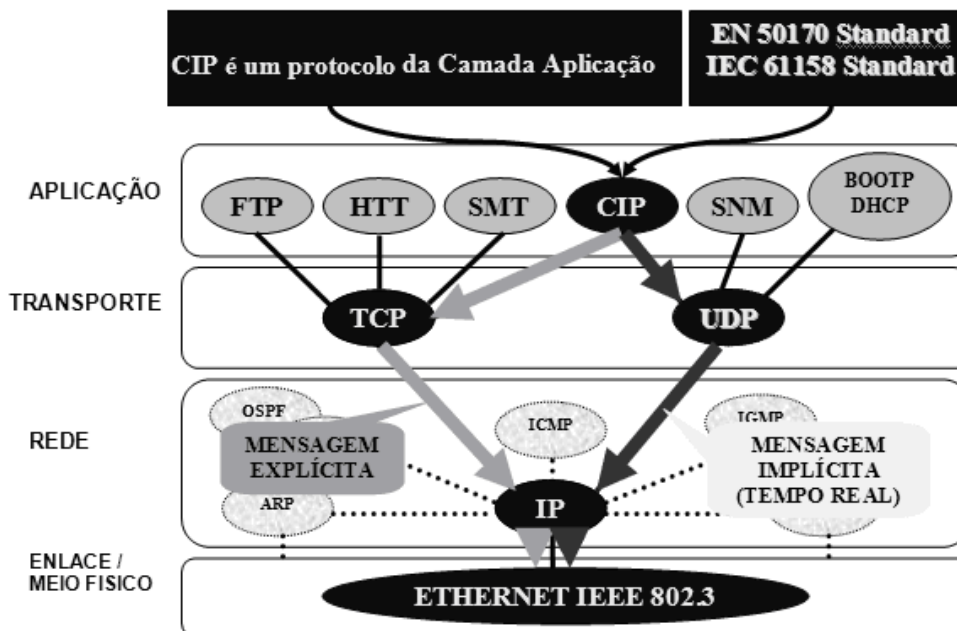


Figura 17 – Comunicação no Ethernet/IP. [25]

O uso de cada comunicação (TCP ou UDP) depende muito da aplicação do projetista. Assim, as transmissões podem ser:

Explícitas: Utiliza-se o TCP e a camada CIP da ODVA. É aplicada na troca de mensagens entre PLC's e IHM's (tempo de ciclo de, aproximadamente, 100ms ou maior) ou para configuração de elementos da rede à longa distância (por exemplo, um *set-point* de um transmissor de pressão). Tais mensagens não são críticas, ou seja, não requerem um baixo tempo de processamento. [36]

Implícitas: Utiliza-se o UDP e a camada CIP da ODVA. É aplicada para comunicações entre I/O's (tempo de ciclo em torno de 10ms), quando o tempo de resposta deve ser o menor possível. Essa mensagem deve ser cíclica, requerendo um baixo tempo de processamento. [36]

A tabela 2 ilustra o comparativo entre os dois tipos de comunicação na rede Ethernet/IP.

Tabela 2 – Resumo dos tipos de comunicação para rede Ethernet/IP. [25] [36]

Tipos de Transmissão	Tipo de Mensagem	Descrição	Exemplo
Informação	Explícito	Transferência de Dados Não Crítica	Leitura / Escrita via instrução de mensagem
Dados I/O	Implícito	Dados de Tempo-Real	Dados de Controle em tempo-real de um Dispositivo Remoto I/O
Sincronização em Tempo-Real	Implícito	Sincronização em Tempo-Real	Troca de Dados em Tempo-Real entre dois processadores

A camada de aplicação CIP, desenvolvida pela ODVA, não se comunica com a camada de aplicação do PROFINET. Assim, não há interoperabilidade entre as diversas redes Ethernet industriais do mercado. [4] [26]

Conforme a norma da ODVA [25], os estudos realizados mostram que o método cíclico é o que possui melhor desempenho para troca de mensagens implícitas, utilizando o protocolo UDP na camada de transporte. Assim, grande parte das mensagens trafegadas na rede Ethernet/IP utiliza Ethernet nas camadas um e dois, IP na camada três e UDP na camada quatro, com tempo de varredura em torno de 10ms. [36]

As mensagens dos periféricos (*Inputs e Outputs*) utilizam transferência crítica de mensagens, tipicamente pacotes de tamanhos pequenos. A comunicação entre I/O's são conexões implícitas, de longo alcance, entre um elemento e outro na rede. Esses pacotes utilizam o protocolo UDP/IP, tendo como vantagem a alta velocidade das mensagens na rede (*throughput*). [36]

A figura 18 ilustra uma aplicação típica para a rede Ethernet/IP, com controle centralizado no controlador mestre (PLC – *Programmer Logic Controller*) da rede. [37] [38]

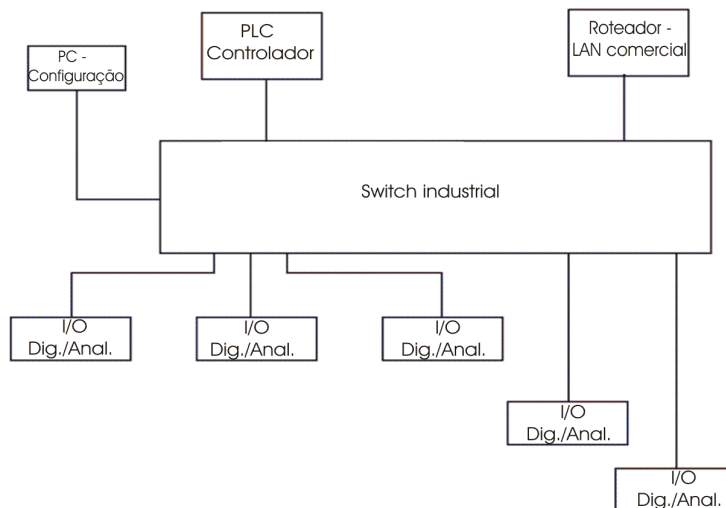


Figura 18 – Representação de uma configuração para rede Ethernet/IP. [37] [38]

2.2.1 O meio físico para a rede Ethernet/IP

Para o meio físico da rede Ethernet/IP, pode ser utilizada uma das configurações abaixo, dependendo do número de elementos conectados, da velocidade da rede, da distância requerida pelo projetista e do tipo de conexão entre os elementos. ^{[25] [36] [38]} A tabela 3 ilustra os vários meios físicos disponíveis e suas características.

Tabela 3 – Características físicas das redes Ethernet industriais. ^{[25] [36] [38]}

Tipo de cabo	Velocidade da rede	Distância máxima sem repetidor	Número máximo de elementos
Par trançado blindado 10BASE-T	10Mbps	100 metros	256
Par trançado blindado 100BASE-T	100Mbps	100 metros	256
Par trançado blindado 1000BASE-T	1000Mbps	100 metros	256
Fibra óptica 10/100BASE-FL	10Mbps	2000 metros	256

Para a configuração utilizando par trançado blindado, o cabo possui dois pares (quatro vias), categoria 5 (com *Shield*). Há a possibilidade de utilização de repetidores (*hubs* industriais) e *switches* industriais que podem aumentar a distância da rede para até 500 metros (utilizando par trançado). ^{[12] [25]}

Já para a configuração utilizando fibra óptica, é possível utilizar repetidores ópticos e, dependendo do tipo de fibra utilizada (monomodo ou multimodo), pode-se obter distâncias de dezenas de quilômetros, colocando vários repetidores em cascata durante o percurso do sinal óptico. ^{[12] [25]}

Contudo, é válido ressaltar que, quanto maior for o número de repetidores ou *switches* industriais, maior será o atraso da rede, pois todos esses elementos possuem *buffers* internos para retenção e retransmissão da informação ao próximo estágio físico. O trabalho de SCHAFER, I e FELSER, M., 2007, evidencia algumas situações e medições relativas ao atraso da informação devido a colocação de *switches* na rede Ethernet industrial. ^[39]

A topologia da rede é, comumente, projetada em árvore, utilizando o controlador mestre como elemento central da rede, os *switches* industriais especiais para derivar e interligar os elementos da rede e os módulos de campo (Nó I/O), onde são conectados os sensores e atuadores.

As figuras 19 e 20 ilustram uma aplicação real, utilizando o conceito acima, porém com outro arranjo físico (topologia estrela), e uma rede Ethernet industrial em uma indústria de manufatura. Como a velocidade de produção dessa aplicação de manufatura é muito alta e os tempos são críticos, há a necessidade de se ter uma rede com alto desempenho no processamento, baixo tempo de varredura, com grande capacidade de tráfego de mensagens e determinística (condição essencial para aplicação da Ethernet no meio industrial). Nesse caso, está sendo utilizada uma rede Ethernet industrial para controle da produção. A figura 19 ilustra o desenho esquemático da rede Ethernet industrial em questão e a figura 20 ilustra a aplicação em si.

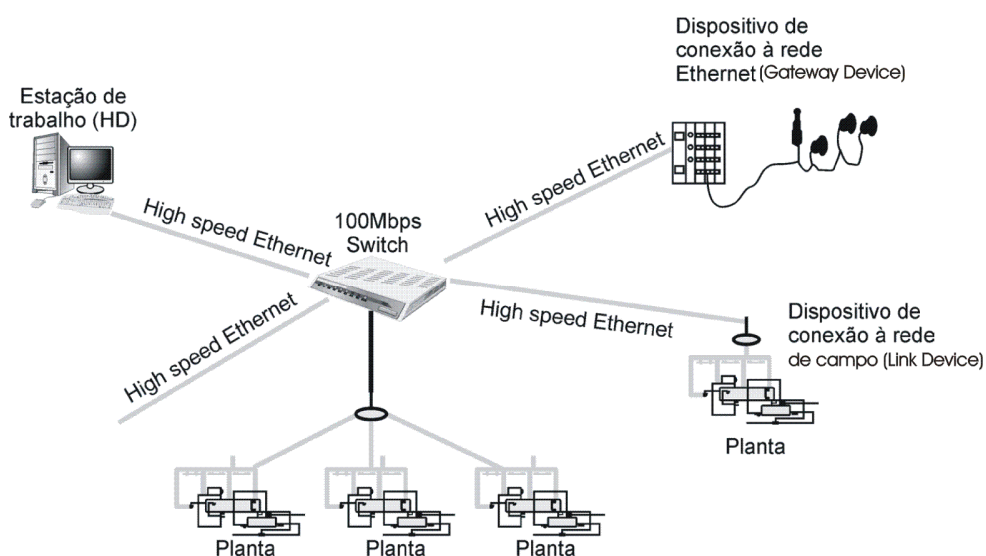


Figura 19 – Desenho esquemático de uma aplicação com Ethernet industrial. ^[40]

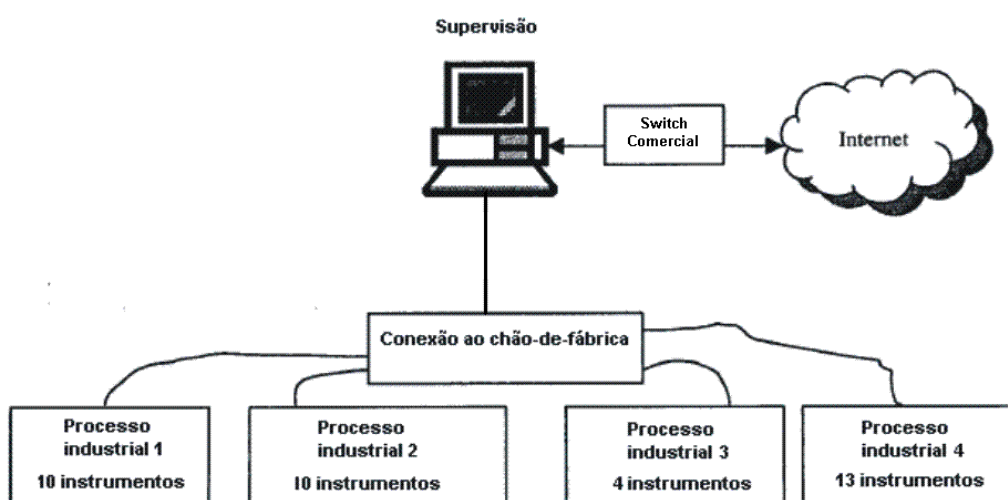


Figura 20 – Aplicação real de uma rede Ethernet para uma indústria de manufatura. ^[40]

2.3 Outros protocolos para Ethernet industrial

Além dos protocolos já citados, há outros tipos aplicados a Ethernet industrial e descritos a seguir.

2.3.1 HSE – High Speed Ethernet

A organização Fieldbus Foundation tem como solução o padrão HSE (*High Speed Ethernet*) para as aplicações industriais em Ethernet. A função do HSE não é substituir as redes Foundation Fieldbus (também conhecidas com redes H1) existentes, mas interconectá-las aos sistemas de supervisão e controle. ^[41]

A tabela 4 ilustra um comparativo entre a rede HSE (Ethernet Industrial) e H1 (Foundation Fieldbus). ^[29]

Tabela 4 – Tabela comparativa entre HSE e H1. ^[29]

	H1	HSE
Velocidade	31.25 kbps	100 Mbps
Distância	1900 m	100 m
Dois fios	Sim	Não
Multidrop	Sim	Não
Alimentação pelo barramento	Sim	Não
Segurança Intrínseca	Sim	Não
Redundância do meio	Não	Sim
Determinístico	Sim	Sim

O HSE é baseado na suíte de protocolos TCP/IP, utilizando os protocolos Ethernet, IP e TCP/UDP. Pode possuir até quatro tipos básicos de categorias de dispositivos diferentes, descritas a seguir: ^{[29][41]}

- *Host Device* (HD): é uma estação de configuração comum.
- *Link Device* (LD): é um dispositivo HSE para conectar um ou mais segmentos Fieldbus H1 ao HSE.
- *Gateway Device* (GD): é um dispositivo HSE para conectar uma ou mais redes externas à HSE.
- *Ethernet Device* (ED): é um dispositivo HSE com condições de conexão direta às aplicações de controle e medição.

A figura 21 ilustra uma aplicação típica entre a rede HSE e a rede H1, sendo que a rede H1 faz a comunicação cíclica dos elementos de campo e a rede HSE opera apenas no nível de supervisão e conexão de várias redes H1. [29]

A divisão em camadas, seguindo a suíte de protocolos TCP/IP, é similar a rede Ethernet/IP, já descrito no item 2.2, com exceção da camada de aplicação desenvolvida para ser utilizada em conjunto com as redes H1. [29] [41]

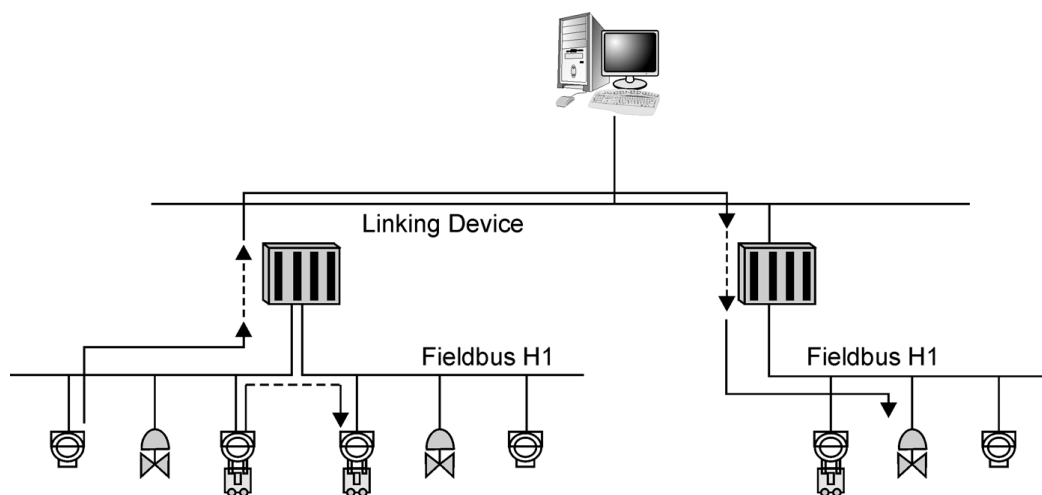


Figura 21 – Exemplo de rede H1 conectada a uma rede HSE. [29]

2.3.2 Sercos III

É outro padrão Ethernet industrial utilizado em aplicações de controle de movimento de alta precisão e velocidade. A comunicação é realizada na topologia anel e há retransmissão do sinal em cada dispositivo da rede. O tempo de ciclo é de, aproximadamente, 31,25us (para oito dispositivos) e 1ms (para cento e cinquenta dispositivos). [29] [42]

A figura 22 ilustra a divisão em camadas do protocolo Sercos III, sendo que há um canal adicional (Tempo real) que pode ser utilizado, em que é possível a transferência de mensagens em tempo real para a camada de aplicação, utilizando os telegramas padrão Ethernet. [29] [42]

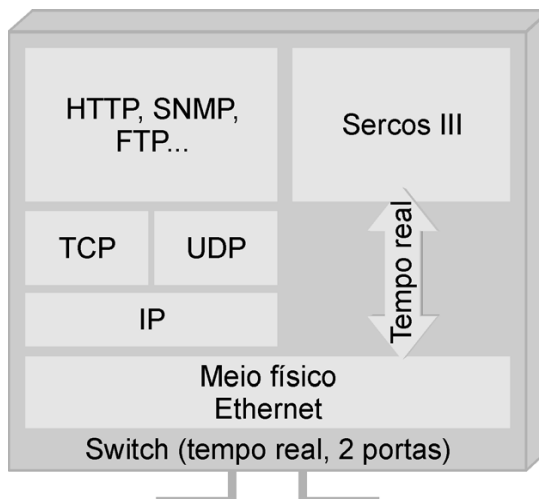


Figura 22 – Divisão de camadas para a rede Sercos III. [29] [42]

2.3.3 Modbus TCP

O padrão Modbus TCP foi elaborado para interconectar o padrão Modbus serial, que existe desde 1979, aplicado a comunicações industriais, ao padrão Ethernet industrial. A comunicação em tempo real pode ser obtida utilizando a camada de protocolo chamada de RTPS (*Real-time Publisher Subscriber*), encapsulado no UDP. A figura 23 ilustra a divisão em camadas do protocolo Modbus TCP. [28] [29]

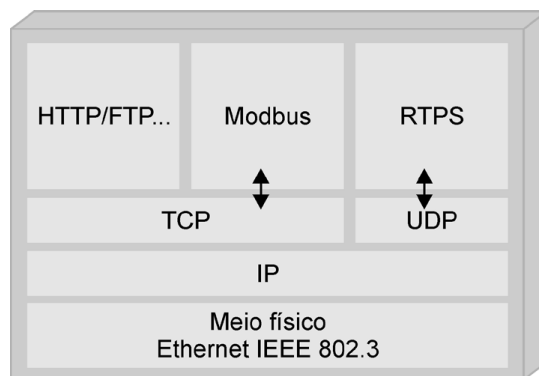


Figura 23 – Divisão de camadas para a rede Modbus TCP. [28] [29]

2.3.4 EtherCAT

O padrão EtherCAT foi criado em 2003. A comunicação é realizada através de uma topologia física em anel, na qual o controlador mestre envia uma mensagem a todos os

módulos de campo, e estes retornarão novamente ao controlador mestre. A figura 24 ilustra a divisão em camadas do protocolo EtherCAT. ^{[28] [29]}

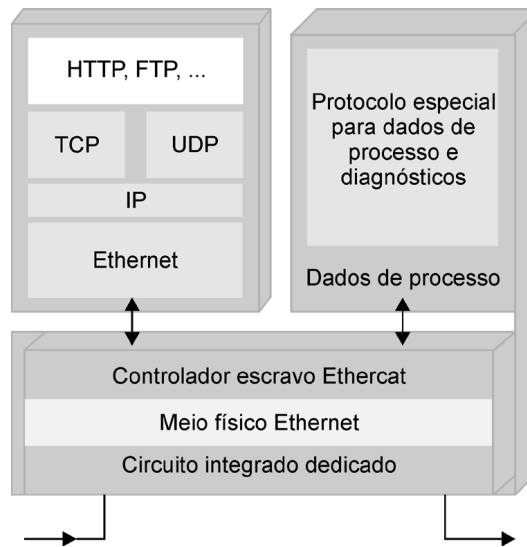


Figura 24 – Divisão de camadas para a rede EtherCAT. ^{[28] [29]}

2.3.5 EPA

O padrão EPA (*Ethernet for Plant Automation*) é uma solução para a aplicação da Ethernet industrial de origem chinesa. Foi criado em 2004 e opera com a suíte de protocolos TCP/IP, similar ao padrão Ethernet/IP. Seu tempo de ciclo varia entre 10 a 100ms, sem sincronização de mensagens. ^{[28] [29]}

2.3.6 EPL

O padrão EPL (*Ethernet Power Link*) foi desenvolvido em 2001 e tem como organização a EPSG (*EPL Standardization Grupo*). A sua operação é similar ao PROFINET, tendo, também, a opção de comunicação em tempo real. A figura 25 ilustra a divisão em camadas do protocolo EPL. ^{[28] [29]}

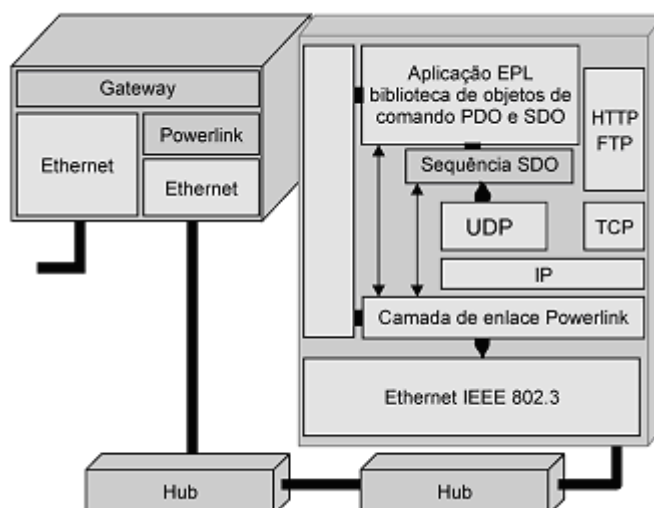


Figura 25 – Divisão de camadas para a rede EPL. [28] [29]

2.3.7 P-Net on IP

O padrão P-Net *on* IP é uma solução para aplicação da Ethernet industrial na automação de processos contínuos, realizando a supervisão dos elementos da rede. Opera com a suíte de protocolos TCP/IP, similar ao padrão HSE. Seu tempo de ciclo varia entre 100 a 1000ms, sem sincronização de mensagens [28]

2.3.8 Vnet/IP

O padrão Vnet/IP é uma solução para a aplicação da Ethernet industrial de origem japonesa. Foi criado em 2004 e opera com a suíte de protocolos TCP/IP, similar ao Ethernet/IP. Seu funcionamento é similar ao padrão EPA. [28]

2.3.9 SynqNet

O padrão SyncNet é uma solução para a aplicação da Ethernet industrial de origem japonesa. Contudo, se diferencia do padrão Vnet/IP, pois trabalha apenas com aplicações em tempo real, ou seja, em controle de movimento. Esse padrão não possui as camadas IP, UDP e TCP, se tornando uma aplicação muito específica. [28]

2.3.10 IEEE 61850

O padrão IEEE 61850 é uma solução para aplicações em automação de sistemas elétricos, ou seja, para automação de subestações elétricas. Não é um padrão Ethernet aplicado no meio industrial. A sua divisão em camadas é similar ao padrão Ethernet/IP e sua aplicação consiste na conexão entre os elementos controladores de proteção da rede elétrica, podendo ser na geração ou na transmissão da energia elétrica. ^[28]

2.3.11 JetSync e TCnet

São outros dois padrões de Ethernet industrial disponíveis no mercado. Ambos os padrões foram criados em 2003 e operam de modo similar ao padrão EPL, podendo ter as opções de comunicação em tempo real ou na suíte de protocolos TCP/IP. ^[28]

2.4 Comparativo entre os diversos protocolos para Ethernet industrial

A tabela 5 ilustra um comparativo entre os diversos padrões de Ethernet industrial com relação a: utilização (aplicação industrial), suporte de aplicações em tempo real, migração de outras redes de campo e desempenho temporal (velocidade de processamento de resposta da rede analisada).

Tabela 5 – Tabela comparativa entre os diversos padrões Ethernet industrial. ^{[28] [29]}

Redes Ethernet	Migração de outras redes	Suporte a aplicações de controle de movimento (tempo real)	Desempenho temporal	Aplicação
PROFINET	PROFIBUS ou Interbus	Sim	Alto	Cíclica, supervisão, controle de movimento e conexão entre redes diversas.
Ethernet/IP	DeviceNet ou ControlNet	Não	Baixo	Cíclica, supervisão, e conexão entre redes diversas.
HSE	Foundation Fieldbus	Não	Baixo	Supervisão e conexão entre redes diversas.
Sercos III	-	Sim	Alto	Cíclica, supervisão e

				controle de movimento.
Modbus TCP	Modbus	Não	Alto	Cíclica, supervisão, e conexão entre redes diversas.
EtherCAT	-	Sim	Alto	Cíclica, supervisão e controle de movimento.
EPA	-	Não	Baixo	Cíclica e supervisão.
EPL	-	Sim	Alto	Cíclica, supervisão e controle de movimento.
P-Net on IP	P-Net	Não	Baixo	Cíclica e supervisão.
V-net/IP	-	Não	Baixo	Cíclica e supervisão.
SyncNet	-	Não	Baixo	Cíclica e supervisão.
IEEE 61850	-	Não	Alto	Cíclica e supervisão.
JetSync e TCnet	-	Sim	Alto	Cíclica, supervisão e controle de movimento.

3 CONCEITOS DE ESCALONAMENTO DE MENSAGENS

As técnicas de escalonamento de mensagens serão utilizadas no decorrer do trabalho e serão analisadas nesse capítulo. As mensagens podem ser definidas como um conjunto de instruções que são executadas, sem interrupção, por uma unidade central de processamento. [44]

Em oposição aos claros benefícios e avanços alcançados com o desenvolvimento das redes industriais, há uma desvantagem deste tipo de sistema relacionado a sua própria concepção a qual é baseada em barramentos de mensagens. Essa desvantagem caracteriza o fato de toda rede industrial ser um gargalo de comunicação, uma vez que um único canal de comunicação é compartilhado por todos os dispositivos e deve ser utilizado tanto para a transmissão de variáveis críticas de processo como de informação de baixa prioridade. Este compartilhamento do canal de comunicação pode ocasionar um congestionamento na transmissão das mensagens que se traduz em atrasos na comunicação. [47] [48]

As técnicas de escalonamento visam minimizar ou reduzir tais atrasos na comunicação, de maneira a melhorar a transmissão e diminuir o congestionamento, respeitando sempre as prioridades de envio das mensagens. Por isso, várias pesquisas sobre escalonamento de mensagens em redes industriais buscam um sistema que seja mais eficaz. [47] [48]

O escalonamento de mensagens é definido, segundo Zwenben e Fox, 1994, como: “A seleção de planos alternativos alocando recursos e atividades em cada instante de tempo, tal que esta designação obedeça às restrições temporais dos processos e as limitações de capacidade de um conjunto de recursos compartilhados.” [43]

Assim, pode-se dizer que uma atividade de escalonamento tem como objetivo alocar uma quantidade de recursos limitados para a execução de determinadas mensagens, no decorrer do tempo, de tal forma que um ou mais objetivos possam ser alcançados. [44]

Geralmente, o escalonamento é executado periodicamente em um espaço de tempo denominado tempo de varredura (*scan time*) do sistema. Para isso, é necessário obedecer a uma regra de escalonamento pré-determinada. Essa regra pré-determinada chama-se tabela de escalonamento. [44]

A realização do escalonamento de mensagens em uma rede industrial possui vantagens e desvantagens, descritas a seguir. [44] [45]

Vantagens: [44] [45]

- Garantia de transmissão das mensagens, diminuindo os atrasos (*jitter* e latência de transmissão).
- Distribuição e equilíbrio do tráfego de mensagens, ao longo do tempo, na rede.

Desvantagens: ^[44]^[45]

- Dificuldade na alteração do conjunto de mensagens a transmitir após a realização do escalonamento.
- Introdução de sobrecargas temporais à comunicação (*overhead*).

Para a realização de um escalonamento de mensagens, são necessárias algumas definições, descritas a seguir:

- Tempo de Liberação (*release*): é o instante inicial em que uma mensagem poderá ser executada. ^[45]
- Tempo de Execução: é o tempo necessário para a execução da lógica de programação (ou estratégia de controle). ^[45]
- Tempo de Transmissão: é o tempo necessário para a transmissão das mensagens no barramento de comunicação compartilhado. ^[46]
- Tempo de finalização (*deadline*): é o instante máximo de tempo para a conclusão de uma mensagem pré-determinada. ^[46]
- Escalonamento ótimo: neste tipo, há uma pesquisa de todas as opções possíveis para o escalonamento das mensagens, de tal forma que o escalonamento só será interrompido quando a melhor solução for encontrada. ^[47]
- Escalonamento realizável: este escalonamento satisfaz todas as restrições temporais do sistema, garantindo o cumprimento da comunicação no barramento. ^[45]^[47]
- Escalonamento preemptivo: a preempção é o ato de interromper uma mensagem que está sendo executada para execução de outra. Somente após o término da segunda mensagem é que a primeira será retomada. ^[47]

Para uma mensagem ser escalonada é preciso que esteja agendada em uma tabela pré-definida, cujo tempo é limitado pelos tempos de finalização (*deadline*) e transmissão. Além disso, ao escalonar uma determinada informação em um barramento, deve-se sempre levar em consideração, além de sua prioridade, a existência de uma precedência com outra mensagem a ser escalonada. ^[44]^[45]

Basicamente, há dois grupos de algoritmos de escalonamento de mensagens aplicados as redes industriais: RM (*Rate Monotonic*) e EDF (*Earliest Deadline First*). ^[45]^[47]

Na técnica RM, as prioridades de transmissão de cada mensagem cíclica do barramento são atribuídas estaticamente, não se alterando mais durante o processo de execução. Neste caso, o algoritmo escalonador impõe a cada mensagem a ser executada uma prioridade proporcional a sua taxa de execução. Geralmente, carrega-se uma tabela de execução das mensagens de forma *off-line* no elemento responsável pelo escalonamento a qual será sempre a mesma a ser executada. O algoritmo atribuirá prioridades altas a períodos de execução pequenos e prioridades baixas a grandes períodos de execução. ^{[45] [47]}

Na técnica EDF, as prioridades podem ser atribuídas de forma dinâmica no barramento, na qual a prioridade de cada mensagem varia no tempo e cresce na medida em que se aproxima o instante de tempo *deadline*. Essa técnica pode ser executada de forma *on-line* no barramento, ou de forma *off-line* antes da comunicação em si, sendo mantida durante o processo. Dentro dessa técnica, há algumas derivações como:

- LLF (*Least Stack Time*), onde a mensagem é selecionada para ser executada segundo a folga expressa pelo tempo *deadline* relativo, subtraído do instante de tempo atual, caracterizado no momento da ação do algoritmo escalonador.
- ERF (*Earliest Release First*), onde a prioridade de cada mensagem varia no tempo e cresce na medida em que se aproxima o instante de tempo *release* (ou próximo instante que a nova informação pode ser liberada). ^{[46] [47]}

Logo, o escalonamento RM (*off-line*) é indicado para mensagens periódicas ou cíclicas, como a comunicação entre os módulos de campo na rede Ethernet, por exemplo, e o escalonamento EDF (*on-line*) para mensagens aperiódicas ou acíclicas, como a configuração ou comissionamento de parâmetros ou diagnósticos da rede. ^{[46] [47]}

A figura 26 ilustra um exemplo de um escalonamento estático, do tipo RM, carregado de forma *off-line* no barramento, sendo não preemptivo. A idéia é gerar a tabela de escalonamento, a qual torna a rede altamente determinística, pois se conhece todo o comportamento temporal do sistema logo na execução da primeira mensagem. Esta tabela de escalonamento é produzida de acordo com o conhecimento, em primeiro momento, de todo o comportamento temporal do sistema, provendo os tempos futuros de liberação (*releases*) das mensagens a serem escalonadas perante todas as operações do sistema ^{[47] [48]}

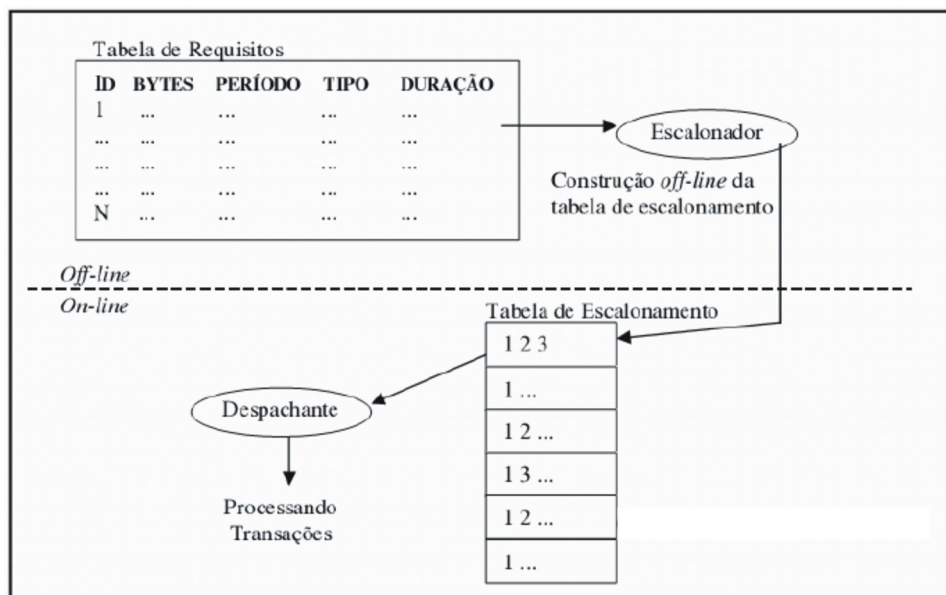


Figura 26 – Exemplo de escalonamento RM, *off-line*. [47] [48]

Este tipo de escalonamento, ilustrado na figura 26, apresenta um alto determinismo de transmissão das mensagens no barramento, devido ao conhecimento de todo o comportamento temporal da rede, desde o primeiro instante da transmissão. [47] [48] Esta previsibilidade é obtida em função do comportamento temporal da rede, representado pelas necessidades temporais já inseridas na comunicação. [47] [48]

O algoritmo RM apresenta uma baixa flexibilidade operacional, pois, para alterações no processo industrial feitas pelo projetista, é necessário gerar novamente a tabela de escalonamento com as alterações necessárias porque sua viabilidade depende do escalonador. Porém, na prática, é difícil um projetista alterar radicalmente o processo industrial, a não ser que se tenha a extensão de seu processo fabril, o que o levará, certamente, a comissionar toda a rede novamente. [47] [48]

Outro fator a ser considerado no escalonamento tipo RM é o tamanho da tabela de escalonamento gerada. Este pode ser um fator complicador para aplicações em barramentos industriais, devido aos recursos limitados dos módulos de campo e, principalmente, a memória e tempo de processamento dos mesmos. [47] [48]

De acordo com Zwenben e Fox, 1994, é possível identificar se um conjunto de mensagens poderá ter um escalonamento factível, utilizando um algoritmo de escalonamento qualquer. [43] Caso a quantidade de mensagens a ser escalonada seja maior do que a ocupação do barramento, não será possível realizar o escalonamento das mensagens. [43]

Para um sistema *fieldbus* qualquer, independente do tipo, as mensagens são trafegadas em um meio físico comum a todos os elementos da rede. No caso da Ethernet industrial pode-

se ter par metálico ou fibra óptica, como visto no capítulo dois. Assim, dentro do tempo de varredura pré-configurado pelo projetista, têm-se duas bandas distintas de comunicação: as mensagens cíclicas ou referentes a conexão remota entre os módulos de campo da rede e as acíclicas ou de parametrização da rede. Essas duas bandas de comunicação alternam-se no barramento de informação de forma contínua e repetitiva, originando os ciclos de varreduras (*scan times*). A figura 27 ilustra a representação de um ciclo de varredura. ^[47]

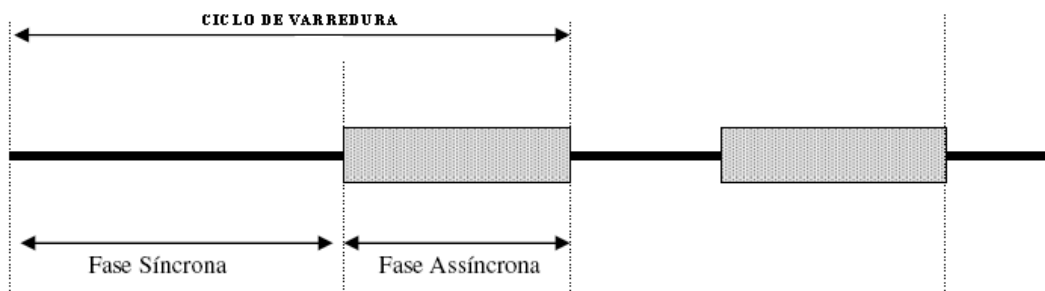


Figura 27 – Exemplo de bandas de comunicação cíclica e acíclica. ^[47]

O período do ciclo de varredura, na prática, está em algumas unidades de milissegundos a, até, alguns segundos, dependendo da aplicação e configuração do projetista. Logo, toda a aplicação do controle de uma rede industrial, distribuída ou centralizada, depende do tempo de ciclo de varredura, o qual será o limitador temporal da comunicação da rede e do processo industrial do projetista. ^[47]

4 NECESSIDADE DE UMA ARQUITETURA DISTRIBUÍDA

As dificuldades encontradas pelos projetistas estão no momento da definição dos elementos da rede e suas configurações, assim como toda a verificação do meio físico da rede (dimensionamento dos cabos e da alimentação dos dispositivos da rede).

Atualmente, a dificuldade na utilização das redes industriais é a interoperabilidade entre os equipamentos dos diversos fabricantes. Há uma variedade enorme de equipamentos, cabos, conectores e outros elementos aplicados a uma mesma rede industrial.

É muito comum um projetista escolher uma tecnologia que melhor lhe sirva naquele momento e tornar-se dependente dela, devido a falta de padronização das redes Ethernet industriais, descritas nos capítulos um e dois.

Assim, quando o projetista decide realizar o seu projeto, há vários problemas que são muito comuns de ocorrer, mas que não são percebidos na fase de projeto, devido a falta de informação e recursos de simulação da rede em questão.

Os problemas podem ser classificados em lógicos e físicos. Os problemas lógicos são:

- A parte lógica da rede mal dimensionada (tempo de varredura inadequado para o processo produtivo em questão), afetando a dinâmica do processo;
- Controle da comunicação das mensagens centralizado no controlador mestre da rede.

Os problemas físicos são:

- Um grande tempo para partida da planta industrial devido a um erro na parte física da rede (comprimentos ou características elétricas do cabo inválidas);
- Os custos com consultorias e reparos da planta são altos, pois os equipamentos já estão instalados;
- Para sistemas centralizados, geralmente não há redundância física do controlador mestre e dos módulos de campo, devido ao custo elevado para realizar tal redundância;
- O conhecimento da tecnologia empregada geralmente não é discutido na fase de projeto e instalação física da rede.

Outro grande problema encontrado pelos projetistas de redes industriais é a centralização da lógica de controle do sistema em um único ponto, o controlador mestre da rede. A grande maioria dos *fieldbuses* possui essa arquitetura centralizada. Isso pode causar sérios problemas aos projetistas no caso de falhas nos controladores. Entende-se por falhas: a

queima do sistema, devido a um mau dimensionamento dos circuitos de proteção, o rompimento do cabo de rede principal e a falta de alimentação no trecho do controlador mestre.

O texto a seguir, figura 28, ilustra, na prática, alguns desses problemas citados anteriormente para uma rede industrial instalada. Os estudos de casos são evidenciados através de relatórios reais elaborados por uma empresa do segmento de redes industriais, na visita técnica a dois grandes clientes. ^[49]

Rede 1: Fábrica de tintas e querosene.

Problemas e causas:

- Tráfego de mensagens elevado na rede (> 95%), devido a um mau dimensionamento temporal e falta de análise prévia desses parâmetros da rede;
- Rede apresenta ruídos de comunicação, oriundos de fontes externas e falta de adequação aos requisitos elétricos das normas;
- O controlador mestre foi danificado devido a uma forte descarga elétrica e ao mau dimensionamento da proteção elétrica do sistema.

Figura 28 – Estudo de caso 1: Relatório de visita técnica. ^[49]

Nessa parte do relatório do estudo de caso, é notório perceber que as causas dos problemas são provenientes da lógica (tempo de varredura da rede muito pequeno, 95% de tráfego na rede) e da parte física (conexão e proteção inadequadas da rede). O problema poderia ser minimizado se o projetista pudesse, de alguma maneira, ter uma visualização prévia do seu processo industrial, através de recursos de simulação computacional e possuísse um sistema flexível distribuído com relação ao controle da lógica de programação, sem necessitar da alocação de um elemento específico e centralizado na rede (controlador mestre). O projetista citado acima permaneceu alguns dias com a sua planta industrial parada, o que ocasionou um enorme prejuízo em sua produção fabril, até a solução dos problemas.

O texto da figura 29 ilustra outro exemplo de uma aplicação industrial e de problemas relacionados a parte física e lógica da rede, similar ao caso citado anteriormente.

Rede 4: Fábrica de manufatura – sucos.

Problemas:

- Tráfego de mensagens elevado na rede (> 93%), devido a um mau dimensionamento temporal e falta de análise prévia desses parâmetros da rede;
- Um dos controladores mestre foi danificado devido a uma forte descarga elétrica e ao mau dimensionamento da proteção elétrica do sistema.

Figura 29 – Estudo de caso 2: Relatório de visita técnica. ^[49]

Nesse estudo de caso, é notório perceber que as causas dos problemas também são provenientes da lógica (tempo de varredura da rede muito pequeno, 93% de tráfego na rede) e da parte física (conexão inadequada da rede). Esse projetista também permaneceu alguns dias com parte da sua planta industrial parada, o que ocasionou um enorme prejuízo na produção fabril.

Assim, seria de grande utilidade ao projetista de redes industriais ter um ambiente gráfico para simulação de sua rede e verificação dos parâmetros configurados antes de realizar a montagem e a configuração dos dispositivos. Além disso, é essencial ter o controle das mensagens trafegadas na rede distribuída ao longo dos módulos de campo e não concentradas somente em um único local da rede (controlador mestre).

4.1 Trabalhos já realizados

Alguns trabalhos realizados ^{[46] [50] [53]} já propuseram, de alguma maneira, a solução dos problemas citados. Porém, as soluções não estão totalmente resolvidas do ponto de vista do projetista e da aplicabilidade para as redes Ethernet industriais. ^{[51] [52]} Os trabalhos [51], [52] e [53] abordam um sistema de controle e modelagem para as redes Ethernet industriais, com controle centralizado no controlador mestre da rede, não evidenciando ou citando a proposta de um controle distribuído aplicado as redes Ethernet industriais.

Além disso, alguns dos trabalhos discutidos a seguir já propuseram de alguma maneira, o uso das técnicas de escalonamento de mensagens aplicadas a rede Foundation Fieldbus para realizar o controle distribuído na rede. ^{[45] [46] [47]}

A contribuição desse trabalho está direcionada para as redes Ethernet industriais, para a implementação e modelagem de um controle distribuído na comunicação da rede e nos módulos de campo, com o objetivo de sanar os problemas listados anteriormente, nesse capítulo, e, ainda, envolver as camadas do modelo OSI, onde nada foi proposto até o momento. Há algumas aplicações específicas, como o PROFINET IO, tipo IRT, onde se pode ter um sistema distribuído para aplicações de controle de movimento de robôs, por exemplo. No capítulo dois, há maiores detalhes sobre as características dessa aplicação. ^[12]

O trabalho de FRANCO [46] propõe o desenvolvimento de um algoritmo de escalonamento para a comunicação em um barramento digital compartilhado, aplicado as redes Foundation Fieldbus. O algoritmo proposto pelo autor utiliza uma heurística EDF para a realização do escalonamento das mensagens, já mencionada no capítulo três, e a técnica de buscas sucessivas, chamada “ramificar-e-podar” (*Branch-and-Bound*), para encontrar a

solução do problema. Essa técnica de buscas sucessivas pela solução baseia-se na enumeração inteligente de todos os pontos factíveis de um problema de combinação numérico e suas possíveis soluções. A “poda” (*bound*) é realizada a fim de eliminar as soluções que não gerarão uma melhor solução que a encontrada até o momento, sendo descartadas da árvore de possíveis soluções para o problema. Assim, é possível otimizar a busca pelas melhores soluções relacionadas ao problema proposto, no caso, o escalonamento das mensagens na rede Foundation Fieldbus. O algoritmo proposto atende as configurações das aplicações do projetista, onde há mensagens cíclicas, em um barramento digital de mensagens. Além disso, o algoritmo proposto definiu restrições temporais para as comunicações cíclicas da rede Foundation Fieldbus em função dos tempos de execução dos blocos funcionais e a precedência entre eles, levando-se em consideração a dependência temporal entre as mensagens (ou, nesse caso, entre os blocos funcionais) trafegadas no barramento. Além das características acima, o trabalho respeita as condições de relacionamentos das mensagens precedentes e excludentes, para garantir a correta sequência e não preempção da transmissão das mensagens trafegadas no barramento, após a configuração de uma estratégia de controle utilizada pelo projetista. Assim, as mensagens cíclicas trafegadas no barramento e suas restrições temporais são especificadas pelo projetista antes da execução do algoritmo (o que caracteriza o trabalho como *off-line* ou *pré-run-time*). Portanto, esse algoritmo destina-se ao escalonamento das mensagens cíclicas na rede Foundation Fieldbus, de processos com tempos de execução, liberação e transmissão, obedecendo as restrições de preempção, precedência e a exclusão de mensagens não escalonáveis no barramento.

O trabalho de HENRIQUES [45] propõe o desenvolvimento de um algoritmo de escalonamento similar ao de FRANCO [46] para a rede Foundation Fieldbus, utilizando as mesmas condições de preempção, tempo e tipo de escalonamento. O método difere-se do anterior, propondo dois tipos de escalonamento possíveis de serem realizados para a utilização da mesma técnica de busca apresentada no trabalho de FRANCO [46]. Enquanto FRANCO [46] utilizou a técnica “ramificar-e-podar” (*Branch-and-Bound*) para realizar buscas sucessivas e encontrar a solução do problema de escalonamento, HENRIQUES [45] utilizou a mesma técnica, porém propondo dois tipos de escalonamento distintos, um antes da execução do escalonamento (chamado de *off-line* ou *pré-run-time*) e outro em tempo de execução (chamado de *on-line* ou *run-time*). O escalonamento *off-line* é similar ao de FRANCO [46] e utilizado para mensagens cíclicas no barramento. Já, para as mensagens acíclicas, HENRIQUES [45] propõe um método de escalonamento *on-line*, quando há ocorrência incerta no tempo das mensagens do barramento (essas mensagens podem ser: informações de

alarmes, diagnósticos da rede ou configurações remotas de dispositivos), pois o sistema não consegue adquirir informações futuras sobre esse tipo de informação aleatória e acíclica. A figura 30 ilustra a descrição do algoritmo *on-line* (para as mensagens acíclicas) proposto pelo autor.

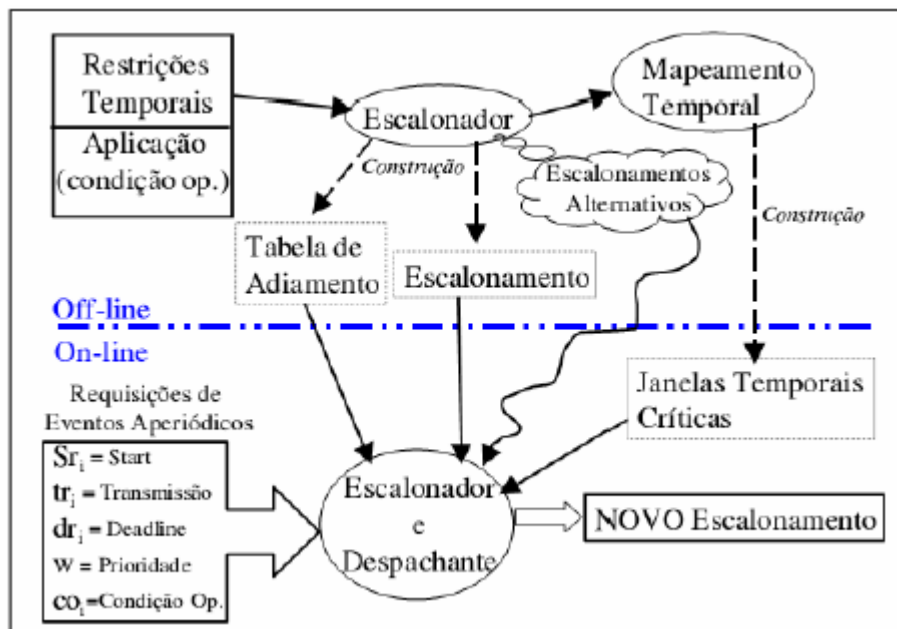


Figura 30 – Escalonamento on-line proposto por HENRIQUES. ^[45]

O trabalho de CICILLINI [47] propõe o desenvolvimento de um algoritmo de escalonamento estático, também aplicado as redes Foundation Fieldbus, utilizando a técnica de escalonamento RM (*off-line ou pré-run-time*), já descrita no capítulo três, diferindo dos outros dois trabalhos anteriores. Os conceitos de preempção e de tempos são similares aos dos trabalhos citados anteriormente, porém o algoritmo propõe a criação de uma tabela de escalonamento gerada a partir de configurações realizadas pelo projetista e pelas prioridades de tráfego das mensagens no barramento. Assim, o algoritmo organiza as mensagens da tabela gerada em: mensagens a serem escalonadas, pré-requisitos para a realização do escalonamento, recursos temporais utilizados, tempo de execução da mensagem a ser trafegada e prioridade de execução dessa mensagem. O trabalho não leva em consideração as mensagens acíclicas trafegadas na rede e os atrasos de processamento dos módulos de campo, considerando, apenas, as mensagens cíclicas. O algoritmo realiza a classificação das prioridades em quatro tipos diferentes, descritos a seguir.

- *Link Local*: É uma conexão local que realiza a ligação das entradas e saídas de um mesmo módulo de campo no barramento. O algoritmo classifica esse tipo como prioridade um.

- *Link Remoto*: É uma conexão remota que realiza a ligação das entradas e saídas de módulos de campo distintos no barramento. O algoritmo classifica esse tipo como prioridade dois.
- *Bloco Funcional*: É uma mensagem cuja prioridade é a execução de um bloco funcional no barramento. O algoritmo classifica esse tipo como prioridade três.
- *Link Back_Calculation*: É uma mensagem destinada a realimentação das informações dos blocos funcionais inseridos nos módulos de campo. O algoritmo classifica esse tipo como prioridade quatro.

Para cada módulo de campo inserido no barramento, criou-se uma variável (cujo nome é “Relógio do Recurso”) que é responsável em armazenar o tempo de execução de cada mensagem. Há, também, outra variável (“Relógio Global”) responsável em armazenar todo o tempo gasto para a execução de toda a tabela de escalonamento.

Para uma mensagem estar pronta para ser executada, ela não deve ter pré-requisitos e deve ter todos os recursos utilizados livres. Após essas verificações, para cada uma das mensagens prontas a iniciar, dispara-se a sua execução. Assim, a variável “Tempo de Disparo” recebe o valor da mensagem do “Relógio Global”. A variável “Relógio do Recurso”, utilizada pela mensagem executada, é atualizada pelo tempo de execução de cada mensagem. Quando uma mensagem tem a sua execução concluída, ela é inserida em uma nova tabela, cujo nome é “Tabela de Resultados” e apresenta as seguintes informações: “Nome da Mensagem”, “Tempo de Disparo” e “Tempo de Execução”.

A figura 31 ilustra a tabela de escalonamento carregada pelo método proposto ao barramento (através de um dispositivo de campo previamente configurado, como proposto pelo autor), para uma determinada aplicação do projetista, e o gráfico temporal do escalonamento da cada mensagem a ser executada na rede, obedecendo as condições de precedência, para cada um dos três dispositivos conectados ao barramento: FI 302, LD 302 e TT 302. Assim, é possível observar que o tempo de varredura dessa aplicação é de 351ms, sendo que as mensagens estão sendo escalonadas na ordem crescente em relação as prioridades configuradas pelo projetista.

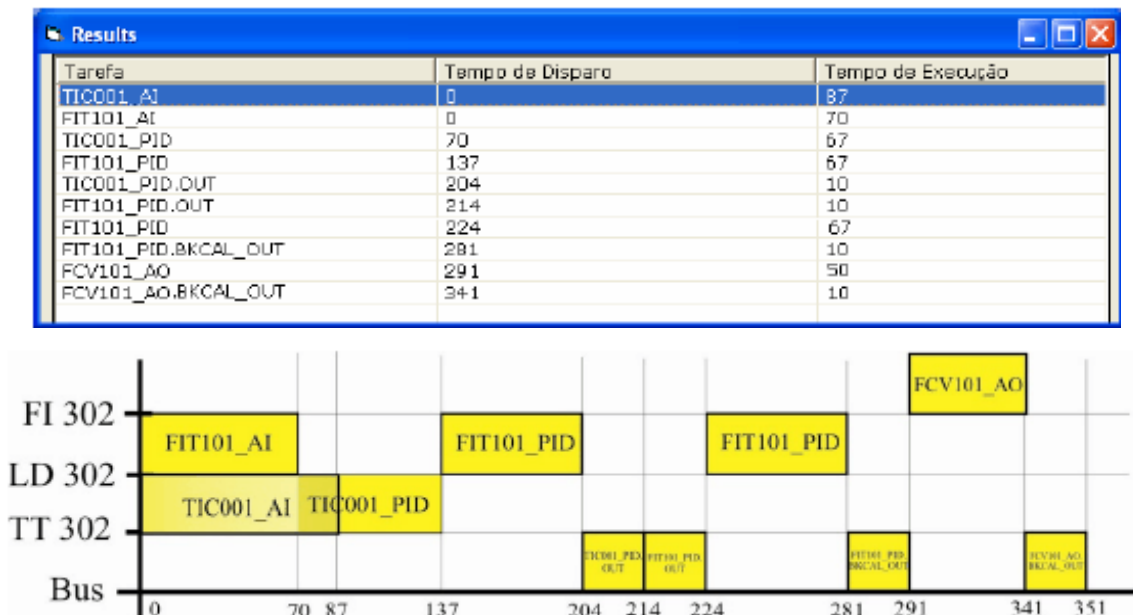


Figura 31 – Exemplo de tabela de escalonamento para rede Fieldbus Foundation. ^[47]

O trabalho de TOVAR [50] propõe o desenvolvimento de análises metodológicas que permitam garantir que aplicações de sistemas controlados por computação distribuída, em tempo real, possam ser implementadas em redes de campo industriais, definidas pela norma EN 50170, com foco nos protocolos PROFIBUS, P-Net e WorldFIP. O trabalho propõe a realização de um controle distribuído, utilizando a técnica de escalonamento RM (*off-line* ou *pré-run-time*), com definição prévia de prioridade das mensagens (similar ao trabalho de CICILLINI [47]), porém, aplicado as redes tipo mestre-escravo, contendo vários mestres. Assim, o controle distribuído é aplicado entre os diversos mestres da rede e não entre os módulos de campo de cada rede (chamados, nesses tipos de redes, de escravos). A figura 32 ilustra as mensagens escalonadas pelo algoritmo RM proposto pelo autor entre quatro mestres distintos, aplicados a rede PROFIBUS.

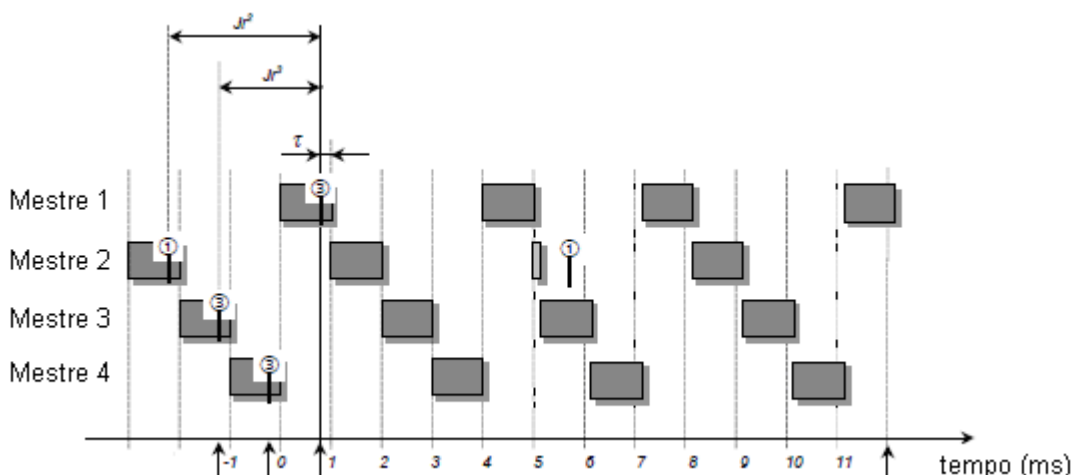


Figura 32 – Exemplo de escalonamento aplicado à rede PROFIBUS, com quatro mestres. ^[50]

O trabalho de ADDAD [51] propõe um novo conceito de modelagem, para uma aplicação industrial, utilizando redes de Petri, com eventos discretos e definidos para representar os tempos de produção de peças de uma máquina. O objetivo é analisar a resposta temporal de uma aplicação industrial através desse conceito de modelagem. A figura 33 ilustra um exemplo proposto pelo trabalho, onde foi analisado o resultado de uma máquina industrial capaz de processar duas peças de forma imediata. A posição p_u indica o estado inicial da máquina e as posições p_1 e p_2 indicam o processo de fabricação da máquina, respectivamente. As transições se referem aos tempos de produção da máquina (transições t_u , t_1 e t_2). As fichas representam as peças que estão sendo fabricadas em um determinado instante pela máquina.

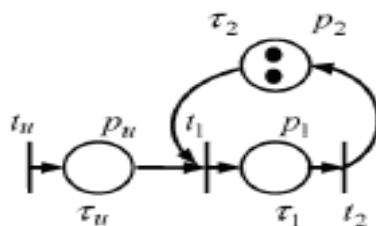


Figura 33 – Exemplo de modelagem para as redes Ethernet industriais utilizando redes de Petri. ^[51]

O trabalho de CARVALHO [53] ilustra o escalonamento aplicado as redes Ethernet industriais, com controlador mestre, a fim de realizar a melhoria temporal da comunicação e do tráfego das mensagens no barramento. O trabalho aborda redes centralizadas (com controlador mestre) e utiliza a técnica de escalonamento tipo RM (*off-line* ou *pré-run-time*). Além disso, CARVALHO [53] realizou a modelagem da comunicação escalonada utilizando

redes de Petri. O escalonamento obtido como resultado do trabalho foi implementado entre a comunicação dos elementos da rede (pergunta e resposta), não se preocupando com as mensagens acíclicas e com a aplicação do projetista. Logo, o trabalho propõe um método de escalonamento, aplicado as atuais redes Ethernet industriais, sem se preocupar com a descentralização do controle da rede, apenas aplicando a técnica de escalonamento, tipo RM, na comunicação centralizada e analisando seus resultados. A rede Ethernet industrial proposta contém doze elementos sendo um mestre, dez módulos de campo e um analisador de rede. A figura 34 ilustra a comunicação proposta pelo trabalho, contendo os doze elementos na rede.

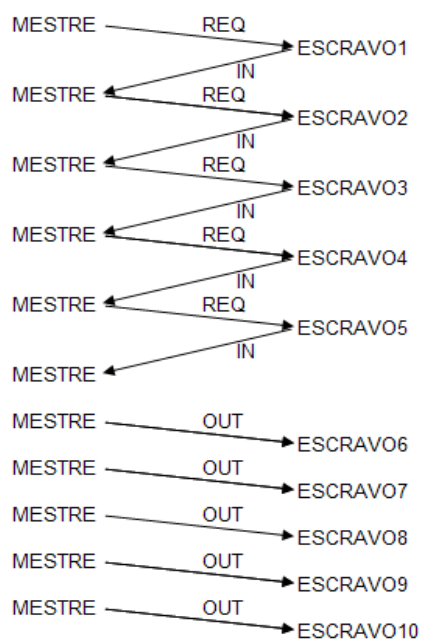


Figura 34 – Exemplo da situação evidenciada pelo trabalho de CARVALHO. [53]

O trabalho de GADERER [54] evidencia a comunicação aplicada à tecnologia Ethernet/IP industrial. Essa tecnologia não provê a opção de realização do controle distribuído em nenhuma aplicação. Há, somente, a opção com controladores centralizados na rede (vide capítulo dois). Ainda baseado nos sistemas centralizados, o trabalho evidencia uma implementação e aplicação para uma rede mestre-escravo, utilizando o protocolo Ethernet/IP, para análise do tempo de recuperação e sincronização da rede após uma falha localizada no controlador mestre da rede (falha de transiente na rede). O objetivo do trabalho é propor um método para otimizar e diminuir esse tempo de recuperação e sincronização da rede, utilizando o protocolo de sincronização de mensagens, para sistemas distribuídos, PTP (*Precision Time Protocol*) e vários controladores mestres. Visa, ainda, a melhoria do processo de recuperação do controlador mestre com falha, após a ocorrência da mesma, melhorando,

assim, a precisão da sincronização global da rede. O trabalho não aborda a comunicação cíclica ou acíclica entre os módulos de campo conectados a um determinado controlador mestre, focando apenas entre os diversos controladores mestre conectados na rede Ethernet industrial proposta (Ethernet/IP).

Analisando os trabalhos relacionados ao tema abordado nessa tese, esta tem por objetivo e finalidade propor um novo método de comunicação distribuído, aplicado as redes Ethernet industriais, onde o controlador mestre da rede é retirado e o controle da comunicação está distribuído ao longo dos módulos de campo inseridos na rede. O trabalho utiliza a técnica de escalonamento RM (*off-line* ou *pré-run-time*) para a realização do controle distribuído de toda a comunicação da rede e do escalonamento de cada módulo de campo, para garantir que as mensagens estejam prontas para serem transmitidas no barramento em cada instante de tempo analisado. Assim, as mensagens de cada módulo de campo estão entrelaçadas com a comunicação distribuída proposta pelo trabalho ao longo de toda a rede.

4.2 Tempos envolvidos nas redes Ethernet industriais

Outro aspecto a ser considerado para a realização do controle distribuído nas redes Ethernet industriais são os tempos envolvidos na comunicação entre os módulos de campo. É essencial a consideração desses tempos no momento da geração da tabela de escalonamento da rede, pois influenciará em todo o ciclo de varredura da mesma e, por consequência, na execução das mensagens programadas pelo projetista.

Basicamente, os tempos envolvidos na comunicação são:

- Tempo de varredura da rede. Define o tempo para a varredura de todas as ações a serem realizadas na rede.
- Tempo previsto para o envio da mensagem (configurado pelo projetista no momento do comissionamento da rede). Determina o máximo tempo para a mensagem ser enviada com sucesso ao seu destino.
- Tempo de processamento do módulo de campo. Determina o tempo real gasto para o processamento da mensagem, a ser trafegada no barramento, em cada módulo de campo da rede (depende do tipo de mensagem: digital ou analógica, por exemplo).

- Tempo de liberação (*release*) da mensagem a ser escalonada. Determina o tempo necessário para a liberação de cada mensagem a ser trafegada no barramento.
- Tempo de latência (atraso na distribuição da mensagem pelo *hub* ou *switch* e processamento nos elementos de redes *wireless*). ^[55]^[56]

Os dois primeiros tempos dependem da quantidade de informação que se deseja trafegar, da taxa de transmissão da rede e da tabela de escalonamento para o caso do controle distribuído. Já o terceiro tempo pode variar ligeiramente dependendo do número de *switches* ou *hubs* entre os elementos que estão trocando informação e da capacidade de processamento desses elementos. De acordo com HELD ^[21], POPP e WEBBER ^[12] e LEE ^[55], é possível realizar uma aproximação desse tempo para 9,6 μ s entre cada elemento que está trocando informação na rede e o *switch* (atraso de comunicação do *switch*, ou *buffer*). Esse valor é um valor médio, calculado empiricamente pelos autores e pode ser chamado de *dead time* (tempo morto) de comunicação. ^[12]^[21]^[55]

Na solução proposta, as análises de atraso no tráfego das mensagens no barramento e os *switches* utilizados nessas análises são similares aos descritos por HELD ^[21], POPP e WEBBER ^[12] e LEE ^[55]. Logo, o trabalho considera o atraso de comunicação (*dead time*), em cada porta do *switch* industrial, também de 9,6 μ s.

5 SOLUÇÃO PROPOSTA

Com a finalidade de solucionar e ilustrar os problemas físicos e lógicos foi desenvolvido uma proposta de controle distribuído das mensagens do barramento, utilizando a técnica de escalonamento tipo RM (*off-line* ou *pré-run-time*), utilizando o algoritmo “ramificar-e-podar” (*Branch-and-Bound*) para a realização da busca das soluções de escalonamento temporal aplicada a cada módulo de campo na rede Ethernet industrial, e implementada através de uma ferramenta computacional, com o objetivo de solucionar os problemas de uma rede centralizada. O trabalho tem como foco a rede Ethernet/IP (desenvolvida pela ODVA). Como a rede Ethernet/IP segue um padrão de encapsulamento das mensagens, a solução proposta tem como foco o tráfego de mensagens cíclicas na rede (utilização das camadas Ethernet, IP e UDP) e acíclicas, como, configurações a distância e a parametrização dos módulos (utilização das camadas Ethernet, IP e TCP). [36]

A figura 35 ilustra os conceitos da divisão das camadas aplicadas a rede Ethernet/IP industrial. Pela figura abaixo é possível notar que o tempo das mensagens cíclicas (troca de mensagens entre os sensores e atuadores na rede) utiliza o protocolo UDP devido a velocidade necessária na comunicação temporal da rede. Já o protocolo TCP é utilizado nas mensagens acíclicas, pois as mensagens trafegadas não são críticas, nesse caso. [36]

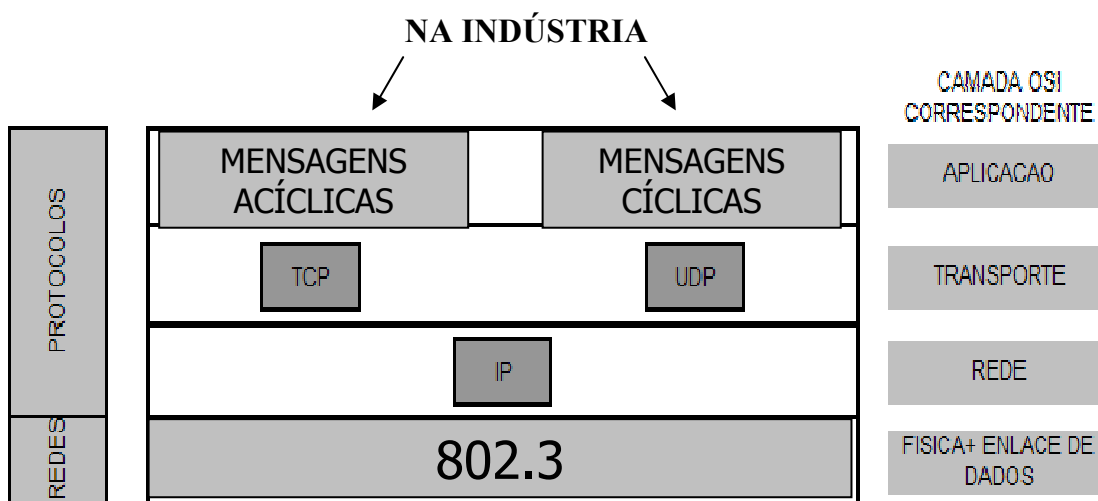


Figura 35 – Comparação entre UDP e TCP para Ethernet industrial. [36]

5.1 Justificativa para utilização da técnica de escalonamento

O atendimento às restrições temporais do sistema escalonado, aplicado as redes Ethernet industriais, com foco na rede Ethernet/IP, garante que quando uma mensagem estiver sendo executada, ela deverá ser finalizada antes de seu respectivo tempo de finalização (tempo de *deadline*).^[43] Esse paradigma deve ser respeitado por todas as mensagens trafegadas na rede com controle distribuído proposta, pois, o não atendimento poderá originar uma falha de comunicação, causando a perda das mensagens trafegadas na rede. Com isso, é de fundamental importância estabelecer uma relação de precedência ou ordem de execução das mensagens a serem trafegadas na rede.^[45]^[50] Além disso, a solução de controle distribuído proposta, utilizando a técnica de escalonamento RM, opera com mensagens não preemptivas, ou seja, mensagens que não podem ser interrompidas para a execução de outra mensagem. As mensagens cíclicas (como exemplo: leitura de sensores e atualização dos estados dos atuadores) e acíclicas (como exemplo: parametrização de um módulo de campo, ou análise de possíveis falhas na rede) devem ser iniciadas e finalizadas num mesmo tempo de execução, para garantia do determinismo e para a realização factível da comunicação da rede Ethernet industrial proposta.^[43]^[50] A figura 36 ilustra o modelo de mensagens sem preempção, onde é necessário o término da execução de uma mensagem para o início da execução da mensagem seguinte no barramento da rede Ethernet industrial proposta.

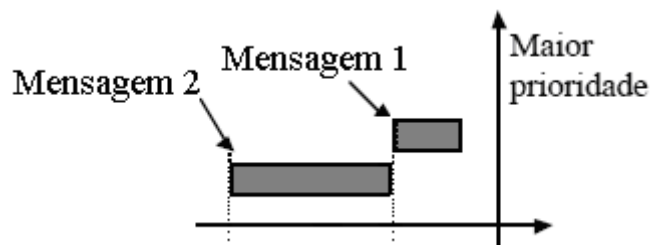


Figura 36 – Conceito de não preempção para as mensagens da rede proposta.^[43]

Com a finalização da execução de um algoritmo de escalonamento, obtém-se um cronograma de ações a serem realizadas, especificadas através de uma previsão temporal definida em função das características e prioridades das mensagens, para que, durante a atividade de comunicação no barramento da rede, estes processos atendam a consistência temporal da comunicação.

Com isso, os algoritmos de escalonamento *off-line* ou *pré-run-time*, como o RM, por exemplo, são capazes de atender as restrições temporais, devido a priorização na ordem de execução de cada mensagem a ser trafegada no barramento Ethernet industrial e a correta ordenação temporal dessas mensagens.

Já os algoritmos *on-line* ou *run-time*, a priori, não podem garantir o determinismo temporal das mensagens a serem trafegadas no barramento imposto pelas restrições temporais, pois as ações do escalonador são atualizadas em cada instante em que há uma alteração nas mensagens (acréscimo ou decréscimo de mensagens, como exemplo, uma mensagem acíclica no barramento) na comunicação do barramento. É possível atender as restrições temporais, neste tipo de escalonamento, se for possível ter conhecimento de quais serão as mensagens acíclicas a serem trafegadas no barramento que poderão ocorrer a cada instante de tempo. Assim, para o atendimento das restrições temporais, é possível analisar as mensagens acíclicas antes de acontecerem e ter alternativas prontas para a realização do escalonamento. Além disso, o tempo necessário para que o escalonador compute todas as restrições e as possíveis opções de escalonamento factível das mensagens é muito grande e despender um grande processamento computacional, se comparado ao escalonamento *off-line* ou *pré-run-time*. Além das justificativas já citadas, os processos e plantas industriais sofrem alterações dinâmicas (como a inserção de novos elementos na rede, por exemplo) em sua estrutura, apenas de forma muito bem planejada, ordenada e previamente analisada pelos projetistas, o que não justifica a utilização de um algoritmo de escalonamento *on-line* ou *run-time* nessas condições.

Com isso, a solução proposta (Ethernet com controle distribuído) utiliza o tipo de escalonamento *off-line* ou *pré-run-time*, devido a garantia das restrições temporais e da priorização na ordem de execução de cada mensagem no barramento.

A tabela 6 ilustra um comparativo entre os escalonadores já discutidos no capítulo três, como forma de justificar a utilização da técnica *off-line* ou *pré-run-time* nesse trabalho.

Tabela 6 – Comparativo entre escalonadores *on-line* e *off-line*. ^{[43] [50]}

Escalonador <i>on-line</i> ou <i>run-time</i>	Escalonador <i>off-line</i> ou <i>pré-run-time</i>
Tempo grande para computar todas as restrições temporais e opções de escalonamento. As restrições temporais não estão previamente definidas, caso não se tenha todo o conhecimento das mensagens a serem trafegadas no barramento.	As restrições temporais já estão previamente definidas através das prioridades.
Permite a otimização do escalonamento, com a alteração do processo industrial de forma <i>on-line</i> .	Permite a otimização do escalonamento, com a alteração do processo industrial de forma <i>off-line</i> .
Pode garantir o determinismo temporal das mensagens, caso tenha conhecimento de todas as mensagens acíclicas a serem trafegadas no barramento.	Garante o determinismo temporal das mensagens, devido a priorização das mensagens trafegadas na rede.

Portanto, com as considerações realizadas, a solução proposta nesse trabalho utiliza uma técnica de escalonamento *off-line* ou *pré-run-time*, tipo RM, com priorização prévia em cada mensagem trafegada no barramento, sendo estas não preemptivas. Com a tabela de escalonamento gerada após a execução do escalonador, obtém-se um cronograma temporal de ações a serem realizadas no barramento, especificadas através de uma previsão temporal definida em função das características e prioridades das mensagens trafegadas, as quais podem ser inseridas em um ou mais (caso algum módulo de campo entre em falha) módulos de campo da rede para a realização do controle da comunicação no barramento, como garantia da realização do controle distribuído aplicados as redes Ethernet industriais. Com isso, a inserção dessa tabela de escalonamento, em mais de um módulo de campo, garante o funcionamento da arquitetura proposta, mesmo em caso de falha de algum módulo de campo que a contenha.

Logo, a ordenação e o controle da comunicação temporal e do escalonamento da rede estão sob responsabilidade de apenas um módulo de campo, garantindo, assim, a comunicação das mensagens cíclicas na rede e possibilitando, também, o escalonamento das mensagens acíclicas, quando houver disponibilidade temporal para a transmissão. Caso haja problemas físicos (rompimento de cabo ou queima do módulo de campo) com esse módulo de campo, que realiza a ordenação e o controle das mensagens, outros poderão assumir essa função, garantindo que a comunicação não seja interrompida, aumentando, assim, a confiabilidade da solução proposta. Para realizar esse controle do escalonamento no barramento, esse módulo de campo que contém a tabela de escalonamento, também contém uma lista dos módulos que estão ativos na rede. Cada módulo de campo é identificado pelo seu endereço IP na solução proposta. Assim, caso o módulo de campo que contenha a tabela de escalonamento entre em falha de operação, o próximo módulo de campo, com endereço IP na sequência crescente ao seu, assumirá o seu lugar na rede, desde que também suporte essa função de gerenciamento do escalonamento da rede proposta. Essa alternância no controle da comunicação do barramento ocorre através do tempo de varredura utilizado para a rede proposta, ou seja, após um ciclo de varredura completo há a alternância do controle, caso o módulo de campo principal não responda aos pedidos de envio de informação requisitados pelos outros módulos de campo (ou seja, entre em falha de operação) o próximo módulo de campo, com endereço IP crescente e que suporte tal função de gerenciamento, fará essa função. Como exemplo: se uma rede contém três módulos de campo, com endereços 192.168.10.1, 192.168.10.22 e 192.168.10.50, respectivamente, sendo que os módulos de campos 192.168.10.1 e 192.168.10.50 suportam a

função de gerenciamento do escalonamento da rede, o módulo de campo 192.168.10.1 terá a função de realizar o controle do escalonamento e, caso falhe, após um ciclo de varredura completo, o módulo de campo 192.168.10.50 entrará em operação, através da lista de módulos que estão ativos na rede realizando o controle da comunicação no barramento.

5.2 Modelagem da solução proposta

A solução proposta nesse trabalho utiliza a técnica de escalonamento RM e está subdividida em três níveis de execução descritos a seguir, de acordo com a ordenação das mensagens cíclicas trafegadas no barramento da rede. Logo, o escalonador desenvolvido analisa a mensagem a ser trafegada por cada módulo de campo em três níveis, a fim de ordenar o fluxo de mensagens na rede. No primeiro nível, o escalonador agrupa todas as mensagens de cada módulo de campo em "sub-grupos por prioridade", os quais são definidos pela ordenação das mensagens (propõem-se três níveis de ordenação distintos: zero, um e dois). No segundo nível, o escalonador analisa o tempo de execução total de cada módulo de campo em cada "sub-grupo de prioridade", criando um outro sub-grupo dentro do primeiro nível, chamado de "sub-grupos por tempo de execução" e, em seguida, organiza todas as mensagens de cada módulo de campo, de acordo com o tempo calculado. No terceiro, e último nível de organização, o escalonador combina todas as mensagens, em cada "sub-grupos por tempo de execução", em ordem crescente.

A figura 37 ilustra o conceito de priorização do escalonador desenvolvido para a implementação do controle distribuído aplicado as redes Ethernet industriais.

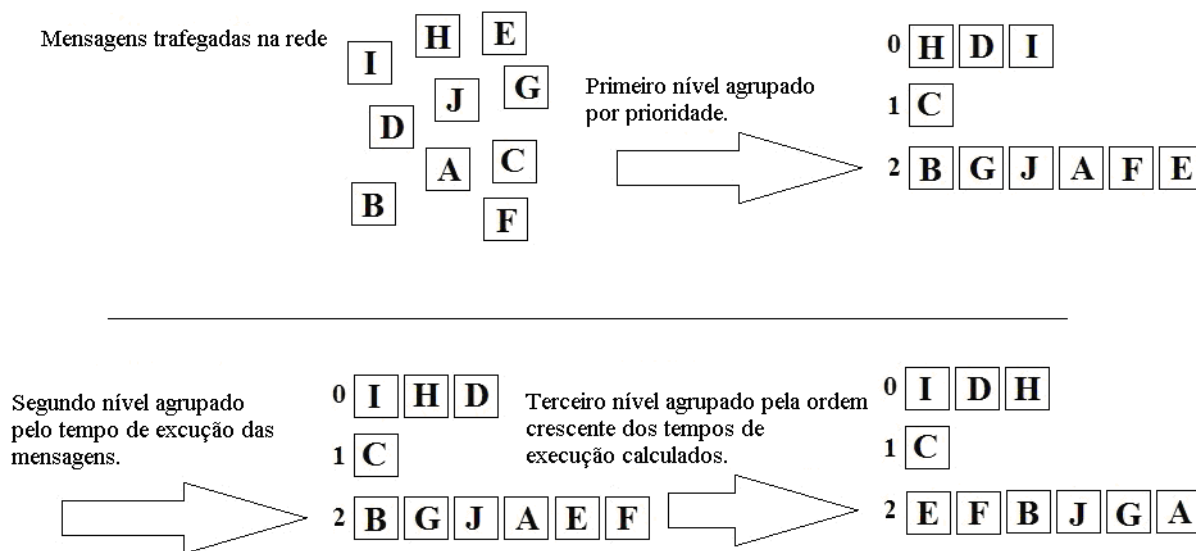


Figura 37 – Ilustração do escalonamento por priorização dos grupos de mensagens.

Na figura 37, há dez quadrados que representam a comunicação distribuída da rede entre sete módulos de campo distintos, onde cada mensagem é representada por uma letra, que indica uma mensagem que está sendo trafegada na rede, vinda de um módulo de campo, como descrito abaixo:

- Quadrado I do módulo de campo um.
- Quadrado B do módulo de campo dois.
- Quadrados D e H do módulo de campo três.
- Quadrados G, A e J do módulo de campo quatro.
- Quadrado F do módulo de campo cinco.
- Quadrado E do módulo de campo seis.
- Quadrado C do módulo de campo sete.

Para a situação da figura 37, os parâmetros temporais dos módulos de campo seguem a seguinte lógica:

- O tempo de execução do módulo de campo três é menor do que o do módulo de campo um.
- O tempo de execução do módulo de campo quatro é menor do que o do módulo de campo dois.
- O tempo de execução do módulo de campo cinco é menor do que o do módulo de campo seis.
- O tempo de execução do módulo de campo que possui a mensagem H é menor do que o do módulo de campo que possui a mensagem D.
- O tempo de execução do módulo de campo que possui a mensagem A é menor do que o módulo de campo que possui a mensagem G, e este é menor do que o que possui a mensagem J.

Assim, quando o algoritmo proposto inicia a ordem de execução das mensagens a serem trafegadas na rede com controle distribuído, esta ordenação altera a disposição temporal das mensagens, iniciando pelo primeiro nível do escalonador e, assim, sucessivamente, até o último nível. Na figura 37, as mensagens são agrupadas no primeiro nível, por ordem, sendo que as mensagens D, H e I são agrupadas no sub-grupo por prioridade zero, a mensagem C está agrupada no sub-grupo por prioridade um, as mensagens A, B, G, e J estão agrupadas no sub-grupo dois e as mensagens E e F estão agrupadas no sub-grupo três.

No segundo nível (por tempo de execução), o escalonador opera dentro de cada “sub-grupo por prioridade”, criando os grupos “sub-grupos por tempo de execução”. Como resultado, nesse segundo nível, as mensagens D e H estão posicionadas na frente da mensagem I, as mensagens G, J e A na frente da mensagem B e a mensagem F na frente de E.

No terceiro e último nível, o escalonador opera em cada “sub-grupo por tempo de execução em ordem crescente”. Como resultado, a mensagem H está posicionada na frente da mensagem D, a mensagem de A na frente da mensagem G e a mensagem G na frente da mensagem J.

Assim, como resultado final do controle distribuído na rede Ethernet proposta, implementado pelo escalonamento tipo RM, tem-se a condição ilustrada na figura 38, sendo que a mensagem H será trafegada primeiramente na rede e a mensagem E por último.

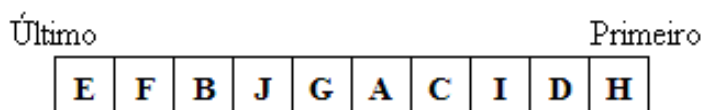


Figura 38 – Ordem de execução das mensagens escalonadas após a execução do escalonamento.

Com relação a ordem das mensagens cíclicas a serem escalonadas no barramento (primeiro nível do escalonador), há uma atribuição das mensagens de forma fixa, não preemptiva, e da seguinte maneira, conforme o algoritmo proposto e o tipo de mensagem a ser trafegada no barramento.

- Mensagem trafegada entre um mesmo módulo de campo (um módulo de campo envia a informação para ele mesmo). Para esse caso, o algoritmo classifica a mensagem como ordem zero.
- Mensagem trafegada entre módulos de campos distintos (um módulo de campo envia a informação para outro módulo de campo, ou seja, uma mensagem tipo início/fim). Para esse caso, o algoritmo classifica a mensagem como ordem um.
- Mensagem intermediária, ou seja, depende de uma mensagem precedente a ela para envio da sua informação a outro módulo de campo (um módulo de campo recebe a mensagem e a repassa a outro módulo de campo, ou seja, uma mensagem tipo início/meio/fim). Para esse caso, o algoritmo classifica a mensagem como ordem dois.

A figura 39 ilustra o fluxograma do algoritmo de escalonamento proposto (escalonador do barramento para as mensagens cíclicas) para a realização do controle

distribuído aplicado as redes Ethernet industriais, de maneira *off-line* ou *pré-run-time*, evidenciando as configurações realizadas pelo projetista e as análises temporais realizadas pelo algoritmo escalonador.

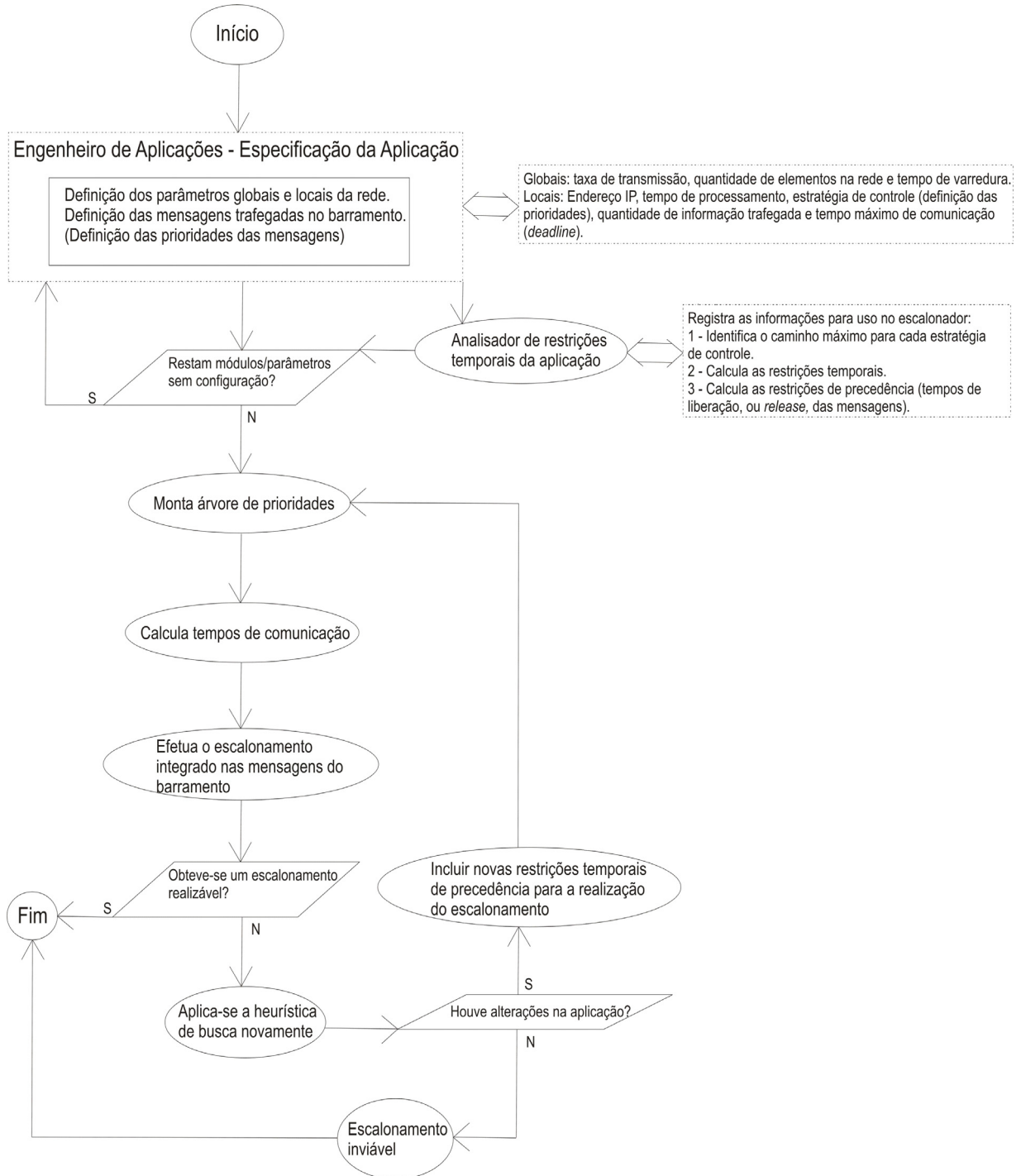


Figura 39 - Fluxograma do escalonador do barramento para as mensagens cíclicas.

Além disso, as mensagens trafegadas na proposta de escalonamento mencionada englobam as mensagens cíclicas e acíclicas no barramento. Nas mensagens cíclicas aplica-se o escalonamento do barramento, com a técnica RM, e individual de cada módulo, com a técnica *Branch-and-Bound*. As mensagens acíclicas são transmitidas após toda a comunicação cíclica do barramento, conforme ilustra a figura 27 (no capítulo três).

Outra proposta deste trabalho é com relação ao controle do escalonamento realizado internamente em cada módulo de campo inserido na rede Ethernet, certificando-se da realização de um escalonamento factível, ou seja, tendo a garantia de que as mensagens estejam prontas para serem transmitidas no barramento por cada módulo de campo. Assim, é possível representar o escalonamento das mensagens no barramento da rede Ethernet industrial proposta e, também, em cada módulo de campo, com o objetivo de ter uma melhor visualização e correspondência com a transmissão entre as diversas mensagens trafegadas no barramento da rede.

A figura 40 ilustra uma representação do escalonamento tipo RM, para uma rede contendo três mensagens a serem trafegadas, de acordo com a ordenação indicada no quadro (um, dois e três, respectivamente).

Com isso, é possível realizar a definição das seguintes mensagens e tempos:

$\tau_i \Rightarrow$ Mensagem “i” a ser escalonada.

$C_i \Rightarrow$ Tempo máximo para execução da mensagem “i” (*deadline*: tempo máximo suportado por um dispositivo do barramento para tráfego de uma determinada mensagem ^[45]).

$T_i \Rightarrow$ Tempo de execução real da mensagem “i”.

$R_i \Rightarrow$ Tempo de liberação, ou disparo, da mensagem “i” a ser escalonada (*release*).

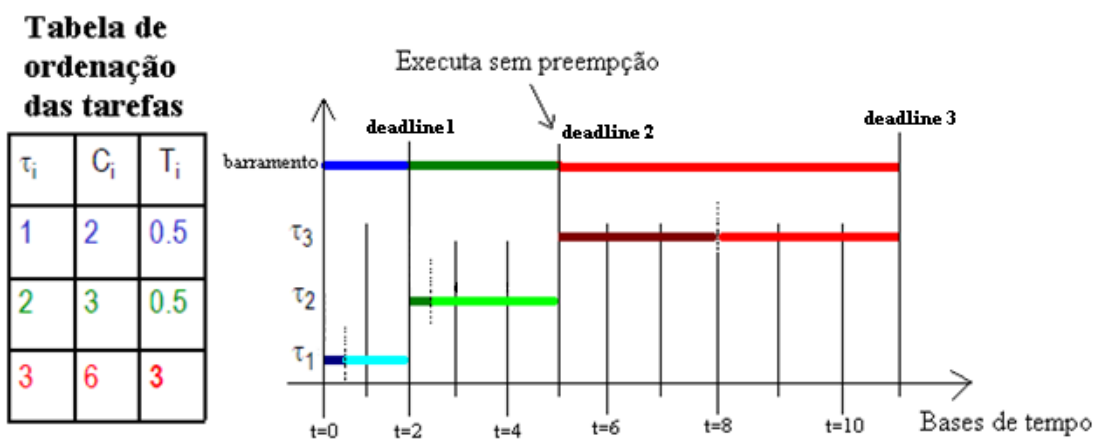


Figura 40 – Ilustração para a representação de um escalonamento alternativo.

Baseado no trabalho de TOVAR [50], é possível utilizar a notação da equação 1 para representar o escalonamento tipo RM:

$$\tau_i = \tau_i (T_i; C_i; R_i) \quad (1)$$

Logo, a representação para o escalonamento da figura 40 é: $\tau_1 = \tau_1 (0.5; 2; 0)$, $\tau_2 = \tau_2 (0.5; 3; 2)$ e $\tau_3 = \tau_3 (3; 6; 5)$. Com isso, as mensagens um e dois possuem uma duração de 0,5 bases de tempo cada e a mensagem três uma duração de 3 bases de tempo. Como a duração máxima de cada mensagem é superior ao seu tempo de execução real, é possível afirmar que:

- A mensagem um, com duração de 0,5 bases de tempo de execução, possui outros 1,5 bases de tempo referente ao deadline (tempo máximo para transmissão dessa mensagem).
- A mensagem dois, com duração de 0,5 bases de tempo de execução, possui outras 2,5 bases de tempo referente ao deadline (tempo máximo para transmissão dessa mensagem).
- A mensagem três, com duração de 3 bases de tempo de execução, possui outras 3 bases de tempo referente ao deadline (tempo máximo para transmissão dessa mensagem).
- O tempo total de execução das mensagens cíclicas, no barramento, é de 11 bases de tempo.
- Os tempos de disparo são, respectivamente, 0, 2 e 5 bases de tempo considerando as três mensagens trafegadas.

Se a duração de alguma mensagem cíclica (T_i) for superior ao seu tempo máximo (*deadline*, C_i), o algoritmo proposto para o controle distribuído, aplicado as redes Ethernet industriais, não fará o escalonamento dessa mensagem, pois não será um escalonamento factível.

Como pode ser observado na figura 40, em paralelo com a linha de tempo do barramento, são definidas as linhas de tempo para cada mensagem, representando o escalonamento individual de cada mensagem trafegada, além da visualização final do barramento. Este tipo de representação possibilita a devida correspondência entre o escalonamento das mensagens no barramento e o escalonamento entre as mensagens de cada módulo de campo, bem como a identificação dessas mensagens na solução proposta.

Para a realização do escalonamento entre as mensagens de cada módulo de campo (escalonamento individual de cada módulo), utiliza-se a técnica de buscas sucessivas,

chamada “ramificar-e-podar” (*Branch-and-Bound*), para encontrar a solução aplicada a cada módulo de campo. ^[46] Essa técnica de buscas sucessivas pela solução baseia-se na enumeração inteligente de todos os pontos factíveis de um problema de combinação numérico e suas possíveis soluções. ^[46] A “poda” (*bound*) é realizada a fim de eliminar as soluções que não gerarão uma melhor solução que a encontrada até o momento, sendo descartadas da árvore de possíveis soluções para o problema. ^[46] A realização do escalonamento das mensagens de cada módulo de campo é imprescindível e necessária, a fim de realizar a correta sequencialização das mensagens que serão disponibilizadas no barramento, em cada instante de tempo pré-determinado, e permitindo a correta sincronização das mensagens escalonadas no barramento. ^[46]

Assim, a proposta aborda dois escalonamentos distintos: um aplicado ao barramento comum a todas as mensagens (escalonamento da comunicação) e outro aplicado a cada módulo de campo, onde cada mensagem é escalonada para, posteriormente, ser disponibilizada no barramento. Ambos os tipos são executados de forma concorrente e entrelaçados, para garantir a correta ordenação das mensagens que são trafegadas no barramento.

Nessa proposta, utilizou-se o conceito de custo computacional para a realização do escalonamento em cada módulo de campo. ^[45] O custo computacional pode ser calculado pelo tempo gasto para a execução da mensagem menos o tempo total disponível (*deadline*). A equação 2 ilustra o conceito de custo computacional. ^[45] Assim, pode concluir que todos os custos calculados devem ser negativos, indicando que o tempo de execução (T_i) é menor do que o tempo máximo (*deadline*, C_i). ^[45]

$$Custo = T_i - C_i \quad (2)$$

Os maiores custos (mais próximos de zero) serão executados de forma imediata, pois o tempo de execução está próximo ao máximo (tempo de execução está próximo ao *deadline*), para cada mensagem. Os menores custos serão executados de acordo com a solução e ordenação realizada pelo algoritmo “ramificar-e-podar” (*Branch-and-Bound*). ^[45]

A árvore de busca do algoritmo “ramificar-e-podar” (*Branch-and-Bound*) é um grafo, aplicado para cada módulo de campo, onde é ilustrado em que instante de tempo cada mensagem será disponibilizada no barramento para a realização do escalonamento da comunicação (utilizando o algoritmo RM). Essa solução é obtida a partir de uma solução inicial (nó raiz) e, caso essa solução não atenda as restrições temporais, novas soluções serão geradas (nós filhos), até o momento que se encontre um escalonamento factível e realizável.

A figura 41 ilustra o fluxograma do algoritmo de escalonamento proposto para cada módulo de campo, utilizando o algoritmo “ramificar-e-podar” (*Branch-and-Bound*) (escalonador individual de cada mensagem a ser trafegada).

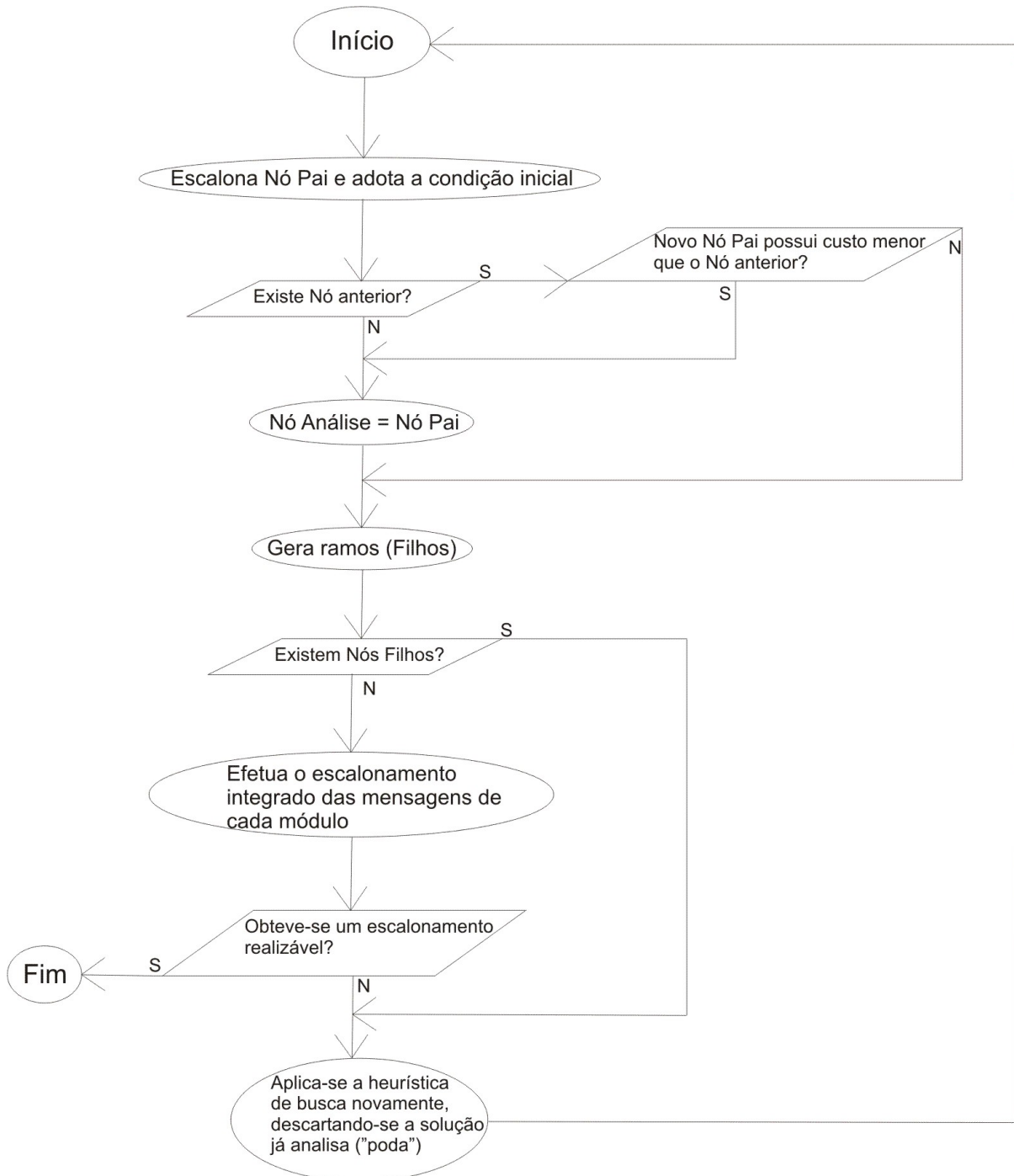


Figura 41 - Fluxograma do escalonador para cada módulo de campo (escalonamento individual).

5.2.1 Equações propostas para a realização do controle distribuído

Para a realização do controle distribuído para as redes Ethernet industriais, além da utilização das técnicas de escalonamento para controle das mensagens, é necessário equacionar o sistema temporal para a realização do tráfego das mensagens da solução proposta, possibilitando a realização do controle distribuído.

Para a realização do cálculo de ocupação do tempo do barramento em relação a todos os módulos de campo, deve-se considerar as seguintes condições, conforme o quadro Ethernet, IP e UDP:

O quadro Ethernet é composto por (a figura 8, no capítulo um, ilustra o formato do quadro Ethernet):

- 64 bits preâmbulo = 8 bytes
- 48 bits de Endereço de destino = 6 bytes
- 48 bits de Endereço fonte = 6 bytes
- 16 bits de Tipo = 2 bytes
- 32 bits de Verificação e detecção de erro = 4 bytes
- Total de bytes de cabeçalho do quadro = 26 Bytes

O quadro IP é composto por (a figura 9, no capítulo um, ilustra o formato do quadro IP):

- Cabeçalho IP = 20 bytes

O quadro UDP é composto por (a figura 11, no capítulo um, ilustra o formato do quadro UDP):

- Cabeçalho UDP = 8 bytes

Logo, a quantidade total de bytes de cabeçalho é de cinquenta e quatro bytes, considerando os quadros Ethernet, IP e UDP. Como a quantidade de bytes de informação deve estar entre quarenta e seis bytes e mil e quinhentos bytes (o quadro Ethernet deve estar compreendido entre setenta dois – 72 bytes – e um mil quinhentos e vinte e seis bytes – 1526 bytes) e levando-se em consideração o *dead time* de 9,6 μ s, já apresentado anteriormente, deve-se fazer duas considerações distintas, de acordo com as equações 3 a 6: ^{[12][36]}

1 - Se a quantidade de bytes da mensagem a ser trafegada na rede (contendo mensagens cíclicas ou acíclicas de entrada ou de saída) for maior ou igual a dezoito (≥ 18), então se propõe realizar o cálculo de seguinte forma:

Onde:

Tout: duração para realizar o envio de apenas uma mensagem de tamanho “*Uout*” bytes na rede.

T_p e T_{dead} : configurações internas de cada módulo de campo, indicando, respectivamente, o tempo gasto pelo módulo para processar todas as suas mensagens e o atraso temporal relacionado a comunicação do *switch* industrial.

T_i : duração para realizar o envio de apenas uma mensagem de tamanho “ U_{out} ” bytes na rede, considerando os atrasos temporais T_p e T_{dead} .

C_i : duração máxima (*deadline*) para realizar o envio de apenas uma mensagem de tamanho “ U_{out} ” bytes na rede, considerando os atrasos temporais T_p e T_{dead} .

R_i : tempo de liberação (*release*) de cada mensagem a ser trafegada.

i : quantidade de mensagens a serem executadas, por módulo de campo, no barramento da rede proposta.

τ : somatória de todos os tempos de execução de todos os módulos de campo inseridos na rede, ou seja, tempo total para as mensagens cíclicas no barramento.

β : tempo restante total na comunicação para futuras expansões no barramento, ou para o tráfego das mensagens acíclicas.

T_{tx} e T_{scan} : configurações globais da rede proposta, indicando a velocidade da rede e o tempo máximo para a realização de toda a comunicação no barramento da rede proposta, respectivamente.

A equação 3 ilustra o cálculo do tempo utilizado para cada mensagem de cada módulo de campo.

$$T_{out} = \frac{[(U_{out} + 54) \cdot 8]}{(T_{tx})} \quad (3)$$

A equação 4 ilustra o cálculo do tempo de execução das mensagens para cada módulo de campo.

$$T_i = (T_{out} + T_p + T_{dead}) \quad (4)$$

A equação 5 ilustra o cálculo do tempo utilizado total (todos os módulos de campo) no barramento.

$$\tau = \sum_{i=1}^n (T_i) \quad (5)$$

A equação 6 ilustra o cálculo do tempo disponível total no barramento.

$$\beta = (T_{scan} - \tau) \quad (6)$$

2 – Se a quantidade de bytes de saída for menor que dezoito (<18), então a quantidade total de bytes é fixada em setenta e dois, que é a mínima permitida em um quadro Ethernet industrial. O número dezoito é obtido da seguinte forma:

Quantidade mínima de bytes do quadro Ethernet = 72

- Bytes de cabeçalho Ethernet = 26
- Sobra para informação de I/O = 46
- Bytes do cabeçalho UDP = 8
- Bytes do cabeçalho IP = 20
- Total de bytes de cabeçalho = 28
- Restante de bytes livres para dados = $46 - 28 = 18$

Assim, se o tamanho dos bytes de informação for menor do que dezoito, é necessário acrescentar bits de enchimento para se ter, no mínimo, os setenta e dois bytes do quadro Ethernet. ^{[11] [12]} As equações 3 a 6 continuam a ser válidas para esse caso, sendo que a única alteração será no parâmetro “*Uout*”, fixo em setenta e dois bytes, nessa situação.

5.3 Análise da parte física da rede Ethernet industrial

De acordo com a tabela 3, para os sistemas que utilizam cabo tipo par trançado blindado, a distância máxima de cada segmento é de cem metros. ^[12] Para a rede Ethernet industrial é necessário utilizar *switches* industriais entre os módulos de campo, a fim de aumentar o alcance e garantir o determinismo da mesma, através da priorização na execução das mensagens a serem executadas, de acordo com as portas disponíveis no *switch*. ^{[11] [12]} Essa priorização pode ser configurada pelo projetista, dependendo dos limites temporais críticos do processo industrial a ser controlado. ^{[11] [12] [55]} Porém, deve-se levar em consideração os pequenos atrasos causados pelos *switches* no processamento das mensagens e não inserir módulos de campo críticos, em relação ao tempo de execução e escalonamento das mensagens do processo industrial, interligados a muitos *switches*, aumentando, assim, o atraso temporal da informação ^{[12] [57]} De acordo com [12] [13] [57], o número máximo de *switches* em uma rede Ethernet industrial, tipicamente, deve ser limitado em quinze. Desta forma, o atraso das mensagens não influenciará no tempo de execução das mensagens trafegadas na rede. ^[12] As distâncias independem da taxa de transmissão configurada para a rede e dos atrasos gerados pelos *switches*, sendo de: 10Mbps, 100Mbps ou 1Gbps. ^[12]

Também é importante notar a capacitância e a resistência do cabo à medida que o comprimento aumenta. Ambos os parâmetros são críticos em sistemas de comunicação a dois fios e podem prejudicar o sinal trafegado no cabo. ^[12]

Para a rede Ethernet industrial, as características principais do cabo par trançado blindado são: ^[58]

- Resistência elétrica do cabo: $R = 115\Omega$ por quilômetro. ^[58]
- Capacitância entre os condutores: $C = 65$ pF por metro. ^[58]
- Frequência máxima de operação do cabo: 1,5GHz. ^[58]
- Impedância característica: $Z_0 = 100\Omega$. ^[58]
- Temperatura de operação: -40°C à $+70^\circ\text{C}$ (IP67). ^[58]
- Atraso de propagação: 10ns (típico). ^[58]
- Tabela de atenuação do cabo em função da frequência, tabela 7 (em dB/100m).

Tabela 7 – Atenuação do cabo Ethernet industrial em função da frequência. ^[58]

Frequência (MHz)	Valores típicos (dB/100m)
1	1,6
4	3,2
10	5,2
16	6,9
20	7,8
31,25	10,5
62,50	15
100	19,5

Com esses valores, é possível o projetista configurar uma rede virtual e verificar se houve excessos na distância em algum trecho da rede. O não respeito aos limites físicos da rede pode causar a degradação do sinal de comunicação e um grande atraso no tempo de propagação do sinal.

A figura 42 ilustra uma rede configurada com um *switch* e dois módulos de campos conectados entre si, para a arquitetura distribuída proposta, utilizando a rede Ethernet industrial.

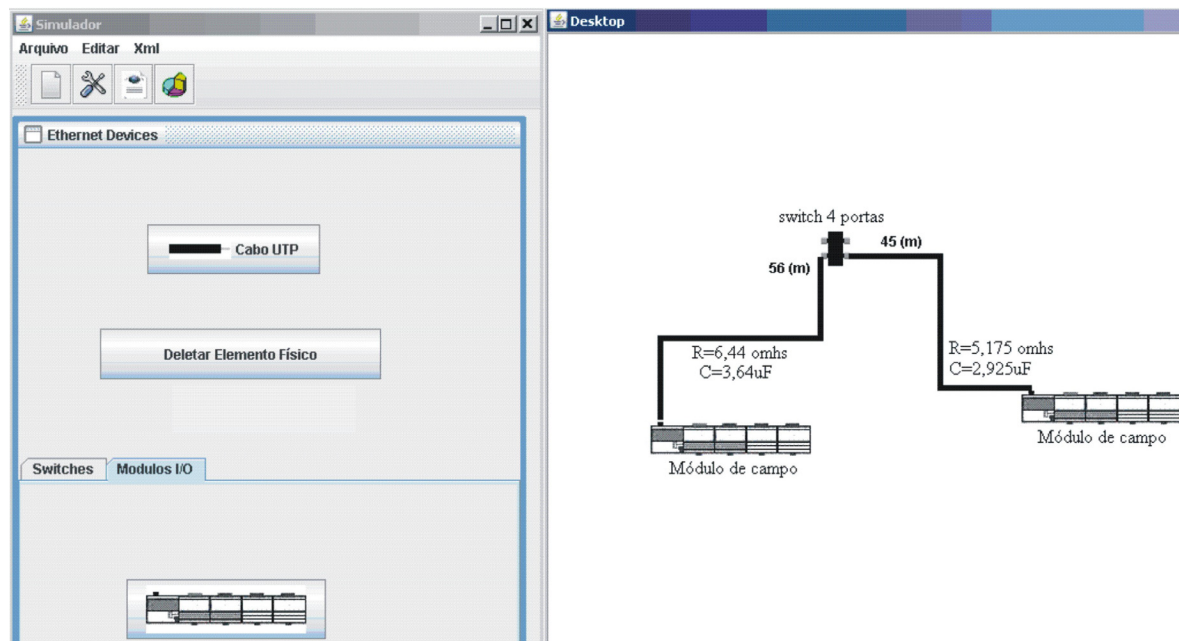


Figura 42 – Configuração da parte física para a rede Ethernet proposta.

As distâncias entre cada trecho não podem exceder a cem metros, para o uso de cabo tipo par trançado blindado, sem repetidor, conforme mostrado na tabela 3. O primeiro trecho foi configurado com cinquenta e seis metros, apresentando uma resistência elétrica de $6,44\Omega$ e uma capacitância entre os condutores de $3,64\mu\text{F}$ (entre *switch* e módulo de campo um), e o segundo trecho com quarenta e cinco metros, apresentando uma resistência elétrica de $5,175\Omega$ e uma capacitância entre os condutores de $2,925\mu\text{F}$ (entre *switch* e módulo de campo dois).

5.4 Filosofia de desenvolvimento para a ferramenta computacional

Para a validação da proposta do trabalho, rede Ethernet com controle distribuído, foi criada uma ferramenta computacional com base na filosofia de orientação a objeto, devido a facilidade na identificação e na codificação das diversas classes da ferramenta, sendo sua implementação planejada para o ambiente *Java*.^{[59][60]}

Os princípios da orientação a objeto foram levados em consideração para o desenvolvimento desse trabalho.

A orientação a objetos é considerada a maneira mais natural de se analisar o mundo. Logo, ela permite construir sistemas computacionais melhores e de maneira mais simples. Isto se deve ao fato de apresentar características essenciais como: abstração, agregação, encapsulamento, herança e polimorfismo.^{[60][61]}

- Abstração: preserva a liberdade de se tomar decisões mantendo, tanto quanto possível, os comprometimentos prematuros.
- Agregação: é o relacionamento nos quais os objetos que representam os componentes são associados a outro objeto, o qual representa a estrutura inteira.
- Encapsulamento: é o ato de ocultar do projetista as mensagens que não são de seu interesse. Deste modo, a comunicação entre objetos é realizada sem a necessidade de se preocupar com a complexidade interna de cada classe.
- Herança: está relacionada com a hierarquia e as relações entre os objetos. Ela permite associar classes descendentes de uma classe superior implementando, assim, métodos e atributos que se diferenciam da classe pai.
- Polimorfismo: processos que executam funções semelhantes em componentes diferentes devem ser chamados pelo mesmo nome. É a habilidade de duas ou mais classes responderem a mesma solicitação, cada uma a seu modo. Conceito de permitir uma única interface para múltiplas funções. ^[59] ^[60]

5.4.1 Funcionalidades necessárias da ferramenta computacional

Para a solução do controle distribuído e desenvolvimento da ferramenta computacional, foi necessária a criação de três elementos básicos da rede: o *switch*, o módulo de campo e o cabo Ethernet, categoria 5.

Cada *switch* inserido na rede Ethernet industrial, com controle distribuído, tem a finalidade de repetir, regenerar e distribuir o sinal elétrico, além de garantir o determinismo temporal das mensagens trafegadas na rede, através da programação de prioridades em suas portas. Os módulos de campo representam qualquer módulo que se deseja acoplar a rede para a conexão dos sensores e atuadores, onde é configurada, também, a estratégia de controle da rede para cada módulo de campo (para quem o módulo enviará a informação e de quem ele irá recebê-la) e os tempos envolvidos na comunicação da rede, de acordo com as equações 3 a 6. É importante ressaltar que na ferramenta computacional desenvolvida, a rede Ethernet configurada já está modelada na arquitetura distribuída, não tendo o controlador mestre central. A estratégia de controle é configurada em cada módulo de campo, dependendo da aplicação final que se deseja realizar.

Na solução proposta, o cabo Ethernet é o elemento que conecta os elementos da rede entre si. Portanto, é o elemento que realmente determina quem faz parte da rede e sua respectiva limitação física (distância máxima por trecho = 100 metros^{[12][36]}).

Resumindo, tem-se, na tabela 8, as seguintes configurações, disponíveis para o projetista e os respectivos elementos de rede.

Tabela 8 – Principais funções dos elementos da ferramenta computacional proposta.

Elemento	Configuração disponível para o projetista
<i>Switch</i>	Número de portas (quatro ou oito portas).
Módulo de campo	Quantidade de bytes de saída, tempo de processamento da mensagem e ser executada, tempo máximo para envio da mensagem (<i>deadline</i>), endereço IP do módulo na rede proposta e estratégia de controle do módulo.
Cabo Ethernet	Distância início-fim de um trecho da rede.
GLOBAL (configura toda a rede)	Quantidade de módulos e <i>switches</i> a serem alocados na rede, taxa de transmissão e tempo de varredura (ou <i>scan time</i>).

5.4.2 Diagramas de classes chaves da ferramenta computacional

Para a sistematização correta dos elementos da rede e da construção da solução computacional foi necessário o uso de classes computacionais, a fim de determinar as características de cada elemento no auxílio ao desenvolvimento do código fonte.

- Classe *Frame Principal*: A classe *Frame Principal* é a classe base para a criação dos elementos da rede. Essa classe foi definida como base, pois apresenta todos os métodos e variáveis comuns aos elementos de rede, tais como: métodos de localização na tela, identificação do dispositivo, imagem, tamanho da tela, entre outros.
- Classes “Controle e Configurações”: As classes “Controle e Configurações” também são derivadas da classe *Frame Principais* e apresentam apenas variáveis relacionadas ao controle de seleção do *switch* e do módulo de campo.
- Classe *Ethernet Device*: A classe *Ethernet Device*, acompanhando toda a tendência da ferramenta computacional, também é derivada de *Frame Principal* e apresenta variáveis e métodos internos para gerenciamento de cada

módulo de campo inserido na rede (endereço IP, quantidade de informação trafegada na rede, tempos relacionados ao tráfego de mensagens e estratégia de controle a ser utilizada).

- Classe “Desenha Gráfico”: A classe “Desenha Gráfico” também é derivada da classe *Frame Principal*. Essa classe tem por finalidade trabalhar com toda a parte gráfica da ferramenta computacional, ou seja, os gráficos temporais da rede.

A figura 43 ilustra a hierarquia entre as classes e suas conexões entre si. Para facilitar o desenvolvimento do código, as classes filhas (*Controle*, *Configurações*, *Ethernet Device* e *Desenha Gráfico*) derivam da classe mãe (*Frame Principal*).

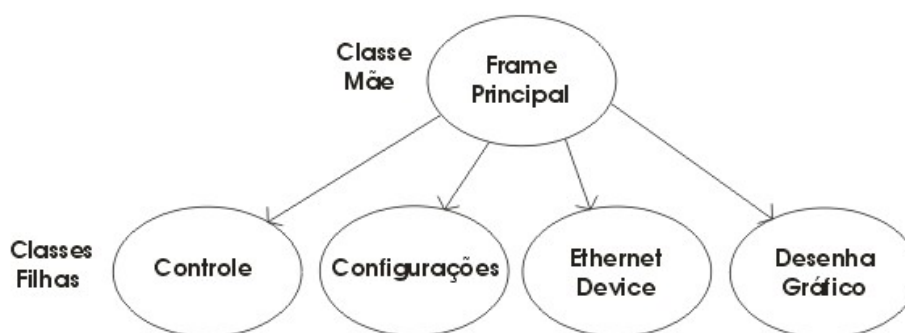


Figura 43 – Diagrama de hierarquia de classes.

No apêndice A, há a codificação das classes listadas acima na linguagem *Java*.

No apêndice B, segue toda a documentação UML (*Unified Modeling Language*) gerada para a codificação da ferramenta computacional proposta, utilizando o diagrama de classes para representação, a fim de facilitar a codificação e implantação da ferramenta. A UML é uma linguagem visual que auxilia no desenvolvimento de uma aplicação computacional orientada a objeto. Ela utiliza uma forma de modelagem padronizada para classes, objetos e diagramas de decisão de estados a fim de facilitar a codificação, implementação e teste das idéias propostas. ^[62] ^[63]

5.4.3 Tratamento de exceções da ferramenta proposta ao nível do projetista

O tratamento de exceções nada mais é do que a realização de várias limitações de entrada de informação, dentro da ferramenta computacional proposta, impossibilitando o projetista de operar com valores irrealis dentro das condições propostas. Assim, valores

incorretos, na prática, são desconsiderados pelas simulações, evitando a possibilidade de simulações imperfeitas ou incorretas. [48] [60]

A seguir, há uma lista de alguns tratamentos de exceções, desenvolvidos na ferramenta proposta, para impossibilitar que o projetista configure a rede de maneira incorreta:

- Tempo de varredura, tempo de processamento da mensagem a ser executada ou tempo máximo para envio da mensagem, configurados de forma errônea na parametrização global ou local do módulo de campo.
- Números de módulos de campo ou *switches* menores do que zero ou maiores do que o máximo permitido (no máximo, duzentos e cinquenta e seis módulos de campo ou quinze *switches*).
- Quantidade de informação de entrada ou saída menor do que zero ou maior do que mil quatrocentos e setenta e dois bytes.
- Endereço IP inválido do módulo de campo.
- Distância de cabo inválida, menor do que zero ou maior do que cem metros por trecho.
- Tentativa de inserção de quantidades superiores de módulos de campo ou *switches* em relação as definidas na parametrização global.

A figura 44 ilustra algumas telas de parâmetros configurados de forma errônea na ferramenta computacional, exemplificando alguns casos citados acima.

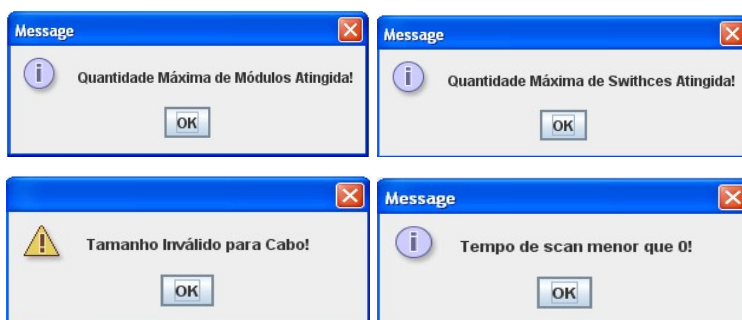


Figura 44 – Alguns tratamentos de exceções da ferramenta computacional.

6 RESULTADOS

Este capítulo é dedicado aos resultados práticos obtidos pelo desenvolvimento da proposta do controle distribuído flexível, utilizando os recursos computacionais e os sistemas reais. Os resultados são obtidos através de vários cenários, envolvendo situações práticas diferentes.

Já a comparação foi realizada utilizando um sistema real desenvolvido com dois módulos de campo, monitorando duas temperaturas distintas e atuando em dois sinais elétricos de saída, um *switch* e cabos para conexão.

6.1 Resultados Obtidos

Para análise dos resultados da nova arquitetura com controle distribuído proposto, são necessários vários cenários de testes de montagem e configuração do sistema, adequados ao cotidiano do projetista, e descritos, a seguir, em três cenários distintos. O quarto cenário evidencia um comparativo entre a rede Ethernet industrial, com controle distribuído proposto, e a rede Ethernet/IP, a fim de analisar e comparar os resultados de uma determinada aplicação nas duas redes analisadas.

6.1.1 Primeiro cenário

O objetivo do primeiro cenário é analisar uma proposta de controle distribuído com alguns elementos e verificar a parte física e lógica da rede proposta a seguir.

Para realizar a validação do primeiro cenário, segue abaixo as condições propostas.

- Elementos na rede simulada: dois *switches* com quatro portas cada um e cinco módulos de campo.
- Parâmetros globais da aplicação configurados pelo projetista em função da sua necessidade temporal do cenário: taxa de transmissão de 10Mbps, tempo de varredura total da aplicação de 10ms, 5 módulos de campo, 2 *switches* de 4 portas cada um.

Módulo_0 – Endereço IP: 192.168.10.1, Tempo de Processamento: 1 μ s, Tempo de Transmissão Máximo: 0,5ms (ordem 1, que suporta o controle da comunicação temporal).

Tabela 9 – Tabela de execução do Módulo_0 – cenário 1.

Write	Read
Módulo_1: 100 bytes	Módulo_2: 50 bytes

Módulo_1 – Endereço IP: 192.168.10.2, Tempo de Processamento: 3 μ s, Tempo de Transmissão Máximo: 0,6ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 10 – Tabela de execução do Módulo_1 – cenário 1.

Write	Read
Módulo_3: 100 bytes	

Módulo_2 – Endereço IP: 192.168.10.3, Tempo de Processamento: 10 μ s, Tempo de Transmissão Máximo: 1,5ms (ordem 0, que suporta o controle da comunicação temporal).

Tabela 11 – Tabela de execução do Módulo_2 – cenário 1.

Write	Read
Módulo_4: 100 bytes	Módulo_2: 30 bytes

Módulo_3 – Endereço IP: 192.168.10.4, Tempo de Processamento: 6 μ s, Tempo de Transmissão Máximo: 1,2ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 12 – Tabela de execução do Módulo_3 – cenário 1.

Write	Read
Módulo_0: 20 bytes	Módulo_0: 100 bytes

Módulo_4 – Endereço IP: 192.168.10.5, Tempo de Processamento: 5 μ s, Tempo de Transmissão Máximo: 1ms (ordem 0, que não suporta o controle da comunicação temporal).

Tabela 13 – Tabela de execução do Módulo_4 – cenário 1.

Write	Read
Módulo_2: 70 bytes	
Módulo_4: 30 bytes	

Para justificar a configuração da taxa de transmissão do primeiro cenário em 10Mbps é possível realizar uma análise prévia dos parâmetros temporais da proposta para verificação da adequação desse parâmetro ao cenário. Assim, de acordo com as condições evidenciadas

nas tabelas 9 a 13 é possível calcular a taxa de transmissão teórica desse cenário para o tempo de varredura total proposto para a aplicação, de 10ms.

A equação 7 ilustra o cálculo para obtenção da taxa de transmissão teórica do primeiro cenário, considerando o tempo de varredura total de 10ms, e não considera os atrasos na rede causados pelos *switches*.

$$0,01 = \left[\frac{(154 + 104 + 154 + 154 + 84 + 74 + 154 + 124 + 84)}{T_{tx}} \right] \cdot 8 + [(1 + 3 + 6 + 10 + 5) \cdot 10^{-6}] \tag{7}$$

Para a equação acima, a taxa de transmissão teórica calculada é de 870,9774Kbps. Logo, como a rede Ethernet industrial possui três taxas de transmissão distintas que podem ser utilizadas, 10Mbps, 100Mbps e 1Gbps, a taxa de transmissão de 10Mbps atende a aplicação desse cenário proposto, sendo utilizada para a sua validação.

A figura 45 ilustra a configuração física do cenário proposto, contendo as distâncias (em metros) entre cada trecho da rede configurada. Nesse cenário, há somente módulos de campo e *switches* (não há o controlador mestre, devido ao controle distribuído), sendo que a distância máxima de cada trecho está limitada a cem metros.

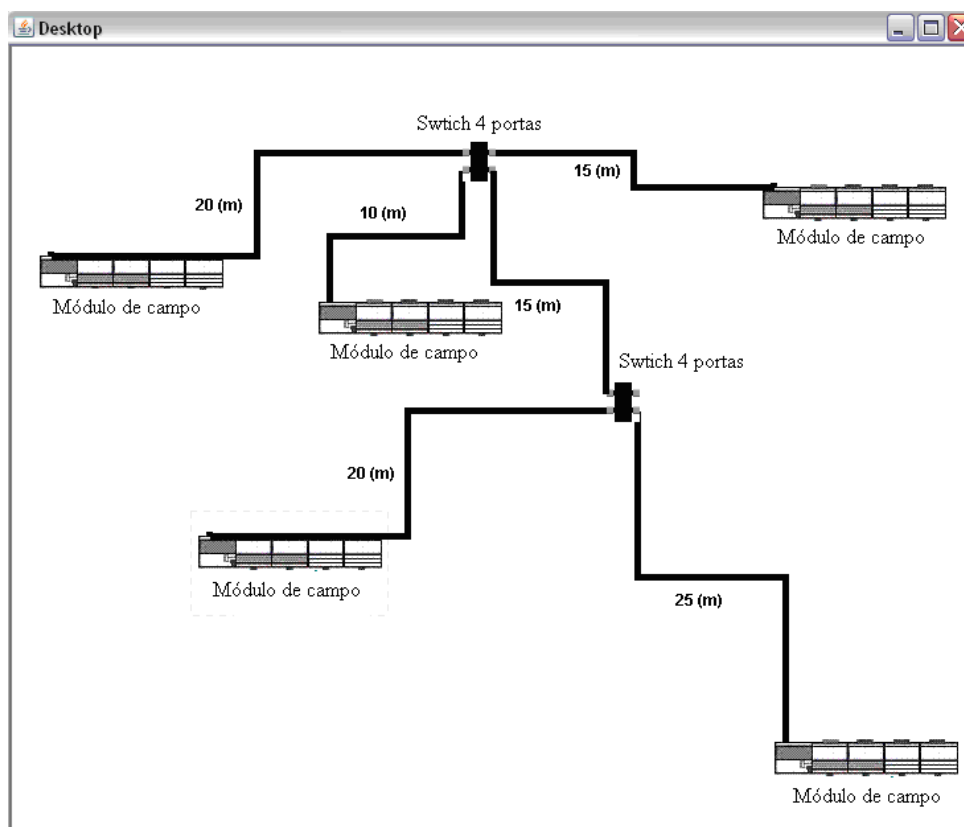
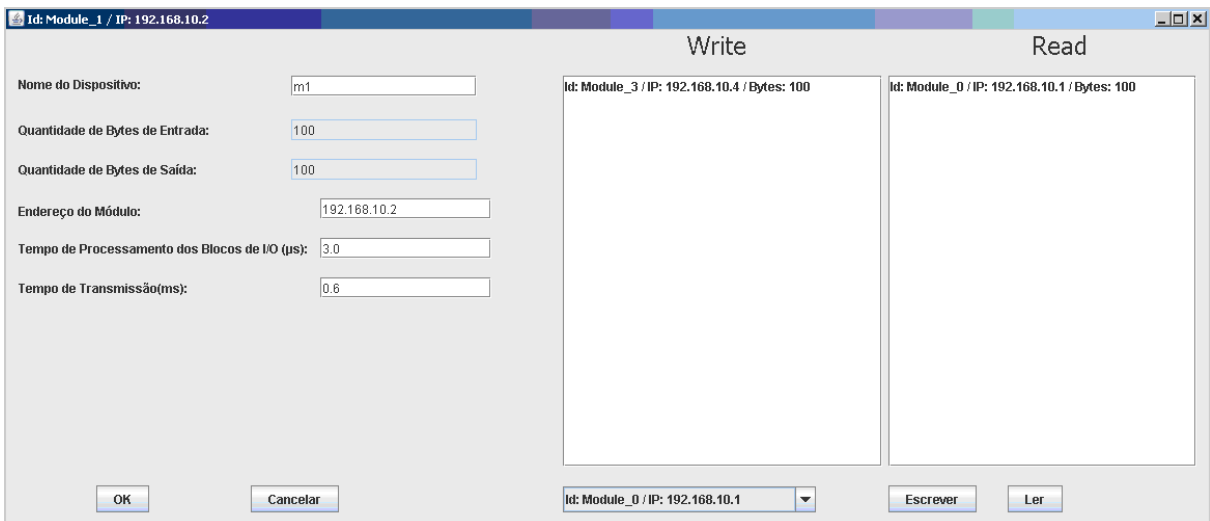
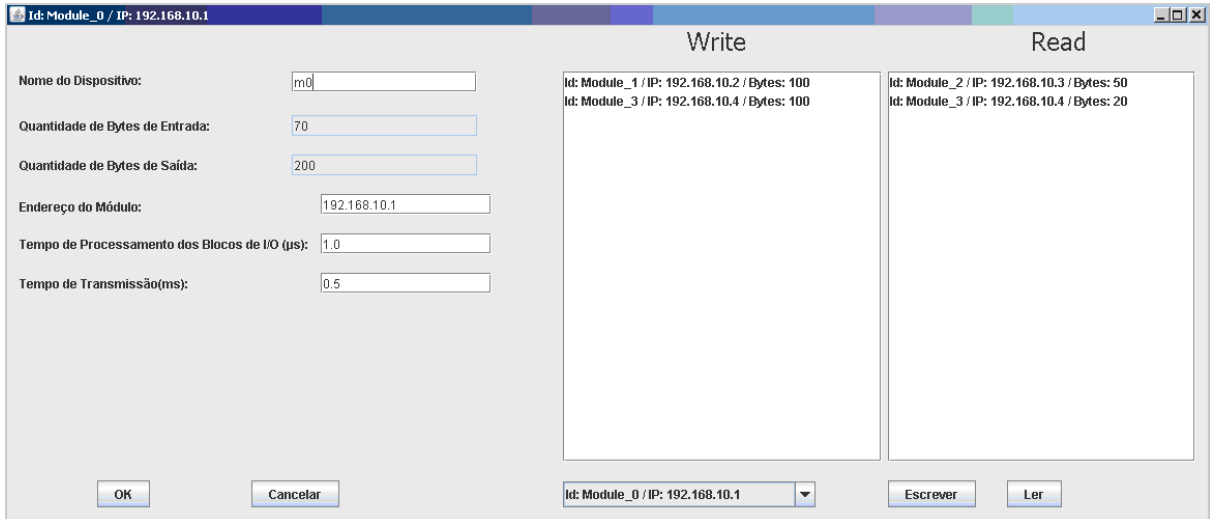


Figura 45 – Configuração física para o primeiro cenário.

A figura 46 ilustra a configuração da aplicação de todos os módulos de campo (Módulo_0 à Módulo_4, respectivamente) apresentada entre as tabelas 9 a 13. As informações mencionadas nas tabelas 9 a 13 estão configuradas nos módulos de campo, incluindo a estratégia de controle (para quem envia, de quem recebe e a quantidade da informação trafegada para cada comunicação escalonada realizada no barramento).



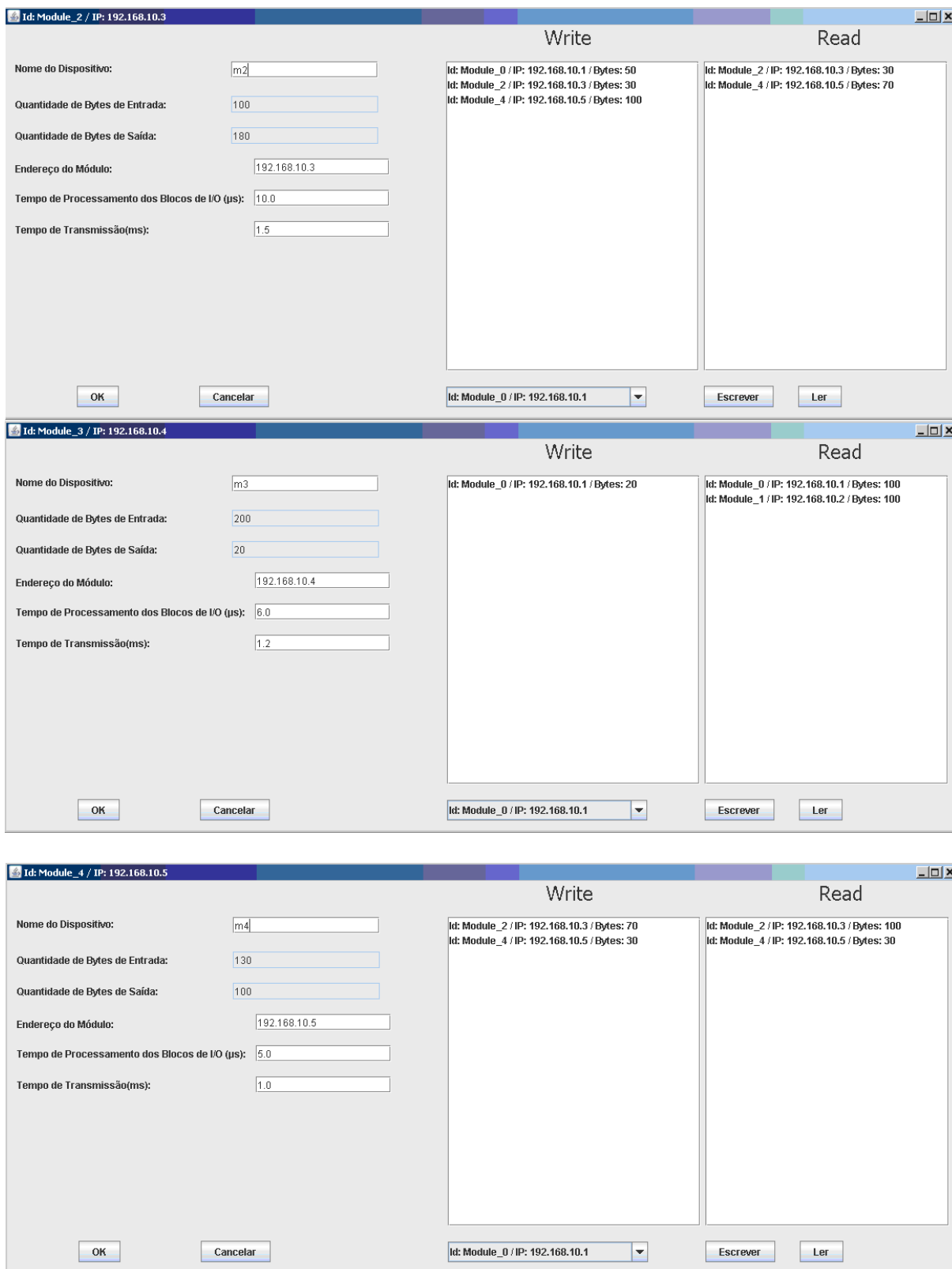


Figura 46 – Configuração dos módulos de campo: Módulo_0 à Módulo_4, cenário 1.

A figura 47 ilustra a tabela de escalonamento do barramento para o cenário proposto e para as mensagens cíclicas, contendo cinco módulos de campo e o controle distribuído entre

Logo, a representação, em milissegundos, seguindo a notação $\tau_i = \tau_i (T_i; C_i; R_i)$ (duração da mensagem; duração máxima da mensagem ou *deadline*; tempo de liberação, ou *release*, da mensagem atual), para o escalonamento realizado na figura 47, tipo RM, é: $\tau_0 = \tau_0 (0.0722; 1; 0)$, $\tau_1 = \tau_1 (0.0868; 1.5; 0.0722)$, $\tau_2 = \tau_2 (0.0748; 1.2; 0.159)$, $\tau_3 = \tau_3 (0.1124; 1.5; 0.2338)$, $\tau_4 = \tau_4 (0.1138; 1; 0.3462)$, $\tau_5 = \tau_5 (0.1358; 0.6; 0.46)$, $\tau_6 = \tau_6 (0.1434; 0.5; 0.5958)$, $\tau_7 = \tau_7 (0.1434; 0.5; 0.7392)$ e $\tau_8 = \tau_8 (0.1524; 1.5; 0.8826)$.

A figura 48 ilustra o escalonamento realizado em cada módulo de campo para cada uma das mensagens cíclicas trafegadas nesse cenário proposto, a fim de garantir que as mensagens escalonadas estejam prontas para serem transmitidas pelo barramento no instante em que forem executadas. Assim, as nove mensagens escalonadas estão entrelaçadas com a comunicação distribuída cíclica do barramento para a rede Ethernet proposta.

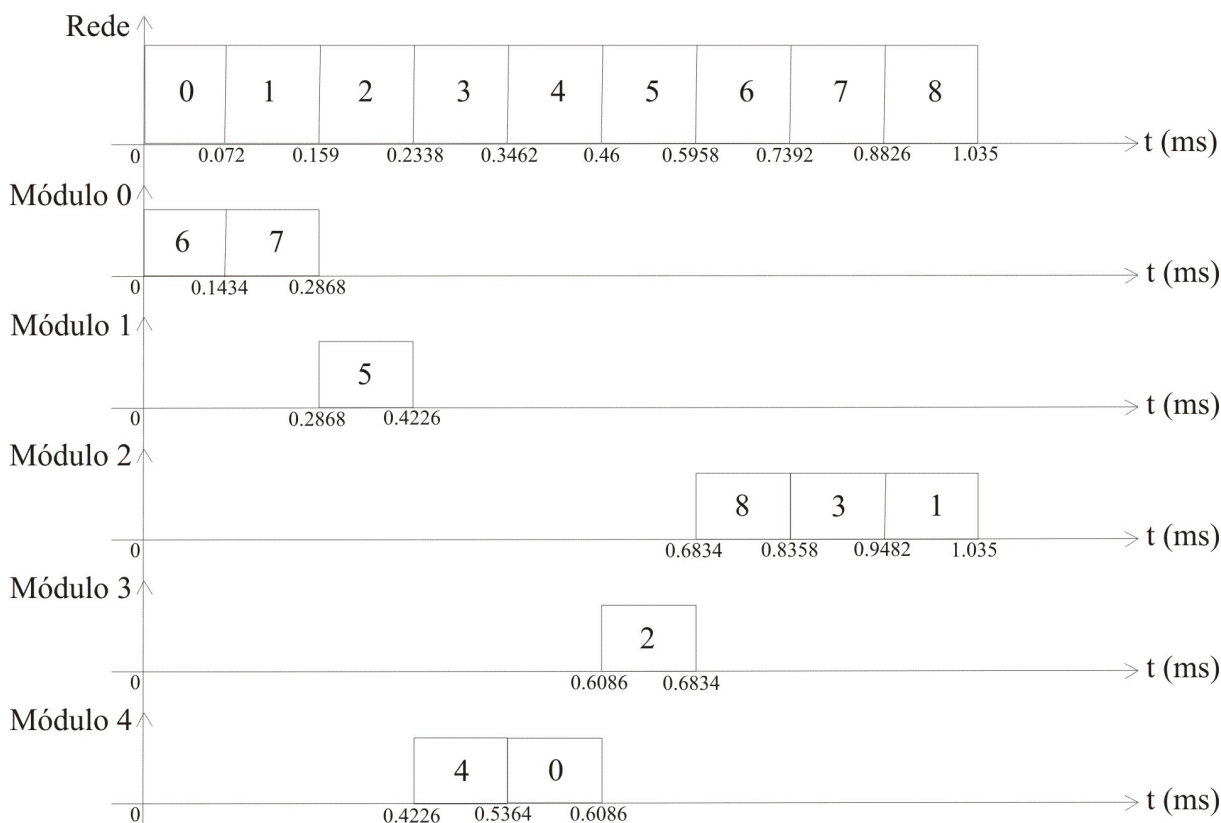


Figura 48 – Ilustração do escalonamento realizado em cada módulo de campo, cenário 1.

Na figura 48, é possível calcular cada custo computacional aplicado a cada mensagem a ser trafegada no barramento. Assim, os maiores custos são os índices seis e sete que devem ser executados de forma imediata e disponibilizados no barramento para a execução correta do escalonamento do barramento, apresentado na figura 47.

- Índice 0: -0,928ms.

- Índice 1: -1,4132ms.
- Índice 2: -1,1252ms.
- Índice 3: -1,3876ms.
- Índice 4: -0,8862ms.
- Índice 5: -0,4642ms.
- Índice 6: -0,3566ms.
- Índice 7: -0,3566ms.
- Índice 8: -1,3476ms.

A correta sequência, utilizando o algoritmo *Branch-and-Bound*, para a disponibilização das mensagens no barramento está evidenciada abaixo.

- Módulo 0: [Índice 6: início em 0ms; Índice 7: início em 0,1434ms].
- Módulo 1: [Índice 5: início em 0,2868ms].
- Módulo 2: [Índice 8: início em 0,6834ms; Índice 3: início em 0,8358ms; Índice 1: início em 0,9482ms].
- Módulo 3: Índice 2: início em 0,6086ms].
- Módulo 4: [Índice 4: início em 0,4226ms; Índice 0: início em 0,5364ms].

Para esse primeiro cenário, há nove estratégias de controle diferentes a serem executadas em tempos distintos.

A figura 49 analisa o tempo total consumido nesse cenário, evidenciando a soma total cíclica e as individuais, de cada módulo de campo. A seguir há a análise temporal calculado pela ferramenta, aplicada a cada módulo de campo.

1-) Módulo_0 = 286,8 μ s para comunicação cíclica: índice 6 + índice 7 da tabela de escalonamento (143,4 μ s + 143,4 μ s). O tempo restante, de 213,2 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

2-) Módulo_1 = 135,8 μ s para comunicação cíclica: índice 5 da tabela de escalonamento (135,8 μ s). O tempo restante, de 464,2 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

3-) Módulo_2 = 351,6 μ s para comunicação cíclica: índice 1 + índice 3 + índice 8 da tabela de escalonamento (86,8 μ s + 112,4 μ s + 152,4 μ s). O tempo restante, de 1,1484ms, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

4-) Módulo_3 = 74,8µs para comunicação cíclica: índice 2 da tabela de escalonamento (74,8µs). O tempo restante, de 1,1252ms, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

5-) Módulo_4 = 186µs para comunicação cíclica: índice 0 + índice 4 da tabela de escalonamento (72,2µs + 113,8µs). O tempo restante, de 814µs, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

Logo, o tempo para a realização da comunicação cíclica total utilizada para esse cenário é de: 286,8µs + 135,8µs + 351,6µs + 74,8µs + 186µs = **1,035ms**.

O tempo disponível para a realização da comunicação acíclica ou para a realização de futuras ampliações na rede do cenário proposto é de: 10ms (tempo de varredura total, configurado inicialmente na aplicação) – 1,035ms = **8,965ms**.

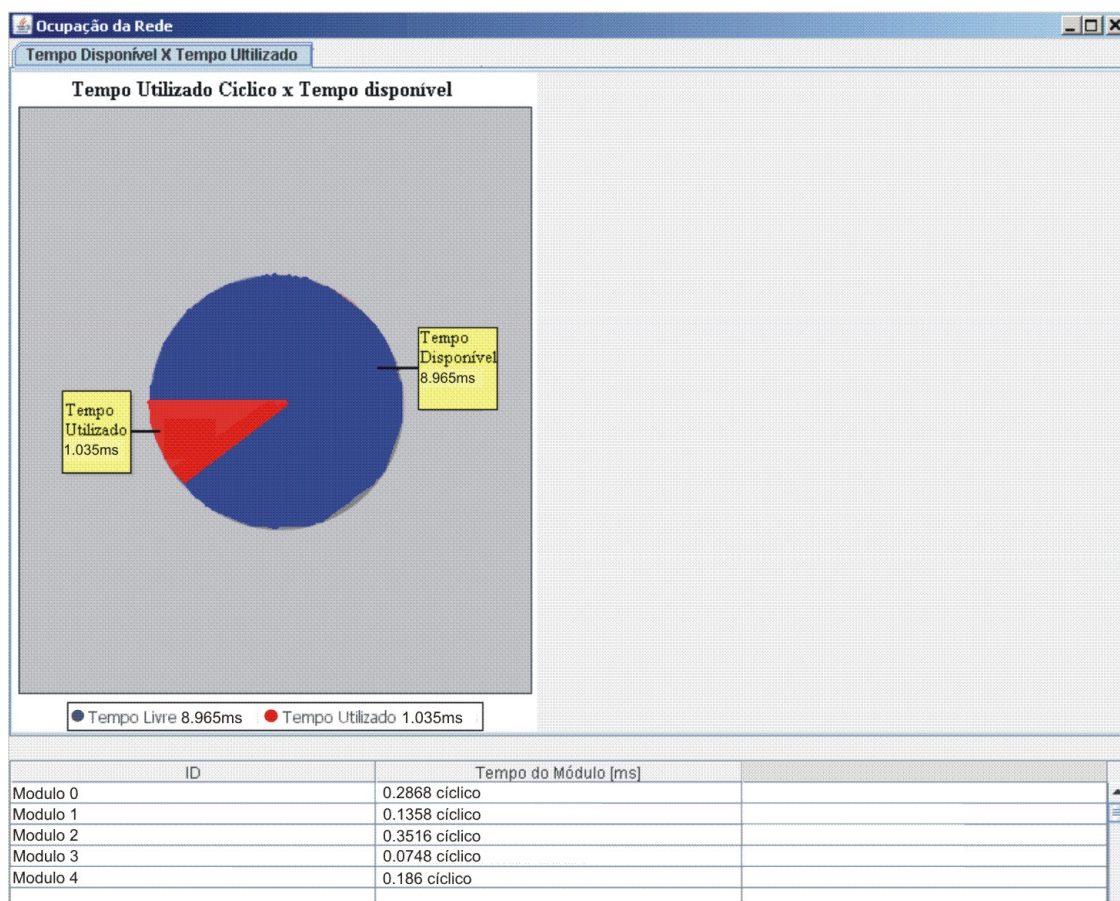


Figura 49 – Análise temporal total cíclica para o primeiro cenário.

Portanto, é possível visualizar a comunicação escalonada do cenário proposto, contendo cinco módulos de campo, nove estratégias de controle escalonadas, dois *switches* industriais e sem o controlador mestre da rede, o que configura um controle distribuído para uma rede Ethernet industrial. A tabela de escalonamento das estratégias de controle no

barramento está evidenciada na figura 47, o escalonamento realizado em cada módulo de campo está na figura 48 e os tempos de ocupação (mensagens cíclicas) e disponível (mensagens acíclicas) total do barramento estão na figura 49.

A ferramenta desenvolvida permite, ainda, que o projetista consiga visualizar possíveis erros de configuração no projeto, como por exemplo, a configuração errônea do tempo de varredura. Caso o projetista configure um valor de 1ms, por exemplo, para a situação desse cenário, a ferramenta, automaticamente, alertará que algo está incorreto, podendo haver falhas na comunicação, pois o tráfego está muito alto, ou seja, o tempo de varredura total está acima do tempo utilizado para a comunicação cíclica total, sendo de 1ms e 1,035ms, respectivamente, não considerando, ainda, a comunicação acíclica, caso ocorra.

A figura 50 ilustra a situação em que o projetista realizou a alteração do tempo de varredura para 1ms.

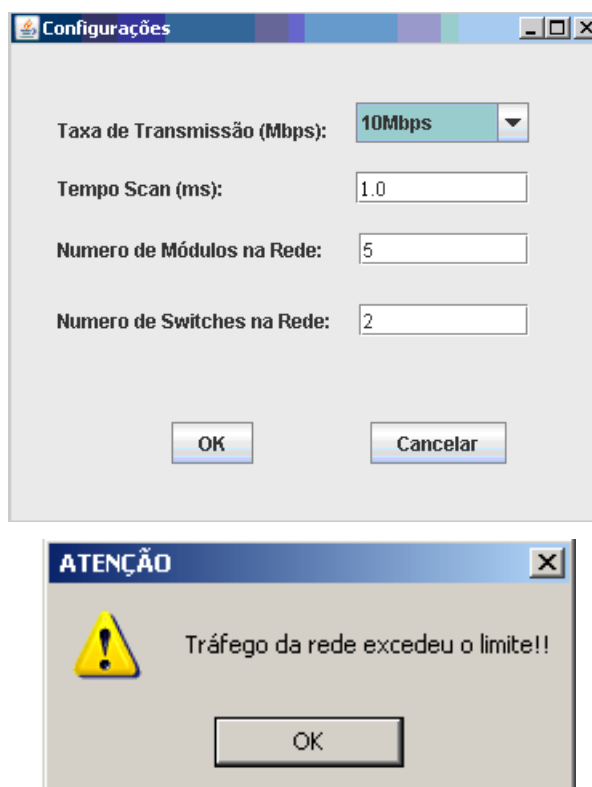


Figura 50 – Configuração errônea de algum parâmetro do cenário proposto.

6.1.2 Segundo cenário

O objetivo do segundo cenário é analisar uma aplicação contendo um número maior de módulos de campo e mensagens trafegadas, a fim de verificar a parte física e lógica, além do correto funcionamento dessa proposta.

Para realizar a validação do segundo cenário, segue abaixo as condições propostas.

- Elementos na rede simulada: três *switches* com oito portas cada um e vinte módulos de campo.
- Parâmetros globais da aplicação configurados pelo projetista em função da sua necessidade temporal do cenário: taxa de transmissão de 100Mbps, tempo de varredura total de 4ms, 20 módulos de campo, 3 *switches* de 8 portas cada um.

Módulo_0 – Endereço IP: 192.168.20.1, Tempo de Processamento: 1 μ s, Tempo de Transmissão Máximo: 0,5ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 14 – Tabela de execução do Módulo_0 – cenário 2.

Write	Read
Módulo_1: 200 bytes	Módulo_1: 200 bytes

Módulo_1 – Endereço IP: 192.168.20.2, Tempo de Processamento: 2 μ s, Tempo de Transmissão Máximo: 0,6ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 15 – Tabela de execução do Módulo_1 – cenário 2.

Write	Read
Módulo_2: 100 bytes	Módulo_2: 100 bytes

Módulo_2 – Endereço IP: 192.168.20.3, Tempo de Processamento: 15 μ s, Tempo de Transmissão Máximo: 0,7ms (ordem 1, que suporta o controle da comunicação temporal).

Tabela 16 – Tabela de execução do Módulo_2 – cenário 2.

Write	Read
	Módulo_3: 50 bytes

Módulo_3 – Endereço IP: 192.168.20.4, Tempo de Processamento: 14 μ s, Tempo de Transmissão Máximo: 0,8ms (ordem 1, que suporta o controle da comunicação temporal).

Tabela 17 – Tabela de execução do Módulo_3 – cenário 2.

Write	Read
Módulo_4: 200 bytes	Módulo_4: 200 bytes

Módulo_4 – Endereço IP: 192.168.20.5, Tempo de Processamento: 5 μ s, Tempo de Transmissão Máximo: 0,9ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 18 – Tabela de execução do Módulo_4 – cenário 2.

Write	Read
Módulo_5: 70 bytes	Módulo_5: 70 bytes

Módulo_5 – Endereço IP: 192.168.20.6, Tempo de Processamento: 6 μ s, Tempo de Transmissão Máximo: 1ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 19 – Tabela de execução do Módulo_5 – cenário 2.

Write	Read
Módulo_6: 55 bytes	Módulo_6: 35 bytes

Módulo_6 – Endereço IP: 192.168.20.7, Tempo de Processamento: 7 μ s, Tempo de Transmissão Máximo: 1,1ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 20 – Tabela de execução do Módulo_6 – cenário 2.

Write	Read
Módulo_7: 85 bytes	Módulo_7: 30 bytes

Módulo_7 – Endereço IP: 192.168.20.8, Tempo de Processamento: 15 μ s, Tempo de Transmissão Máximo: 1,2ms (ordem 1, que suporta o controle da comunicação temporal).

Tabela 21 – Tabela de execução do Módulo_7 – cenário 2.

Write	Read
Módulo_8: 10 bytes	

Módulo_8 – Endereço IP: 192.168.20.9, Tempo de Processamento: 9 μ s, Tempo de Transmissão Máximo: 1,3ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 22 – Tabela de execução do Módulo_8 – cenário 2.

Write	Read
Módulo_9: 60 bytes	Módulo_9: 55 bytes

Módulo_9 – Endereço IP: 192.168.20.10, Tempo de Processamento: 10 μ s, Tempo de Transmissão Máximo: 1,4ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 23 – Tabela de execução do Módulo_9 – cenário 2.

Write	Read
Módulo_10: 70 bytes	Módulo_10: 93 bytes

Módulo_10 – Endereço IP: 192.168.20.11, Tempo de Processamento: 10 μ s, Tempo de Transmissão Máximo: 1,5ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 24 – Tabela de execução do Módulo_10 – cenário 2.

Write	Read
Módulo_11: 90 bytes	Módulo_11: 55 bytes

Módulo_11 – Endereço IP: 192.168.20.12, Tempo de Processamento: 1 μ s, Tempo de Transmissão Máximo: 1ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 25 – Tabela de execução do Módulo_11 – cenário 2.

Write	Read
Módulo_12: 200 bytes	Módulo_11: 25 bytes

Módulo_12 – Endereço IP: 192.168.20.13, Tempo de Processamento: 2 μ s, Tempo de Transmissão Máximo: 0,9ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 26 – Tabela de execução do Módulo_12 – cenário 2.

Write	Read
Módulo_13: 500 bytes	Módulo_11: 55 bytes

Módulo_13 – Endereço IP: 192.168.20.14, Tempo de Processamento: 5 μ s, Tempo de Transmissão Máximo: 1ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 27 – Tabela de execução do Módulo_13 – cenário 2.

Write	Read
Módulo_14: 600 bytes	Módulo_10: 15 bytes

Módulo_14 – Endereço IP: 192.168.20.15, Tempo de Processamento: 4 μ s, Tempo de Transmissão Máximo: 0,5ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 28 – Tabela de execução do Módulo_14 – cenário 2.

Write	Read
Módulo_15: 100 bytes	

Módulo_15 – Endereço IP: 192.168.20.16, Tempo de Processamento: 3 μ s, Tempo de Transmissão Máximo: 0,6ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 29 – Tabela de execução do Módulo_15 – cenário 2.

Write	Read
Módulo_16: 53 bytes	Módulo_11: 35 bytes

Módulo_16 – Endereço IP: 192.168.20.17, Tempo de Processamento: 3 μ s, Tempo de Transmissão Máximo: 1ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 30 – Tabela de execução do Módulo_16 – cenário 2.

Write	Read
Módulo_17: 43 bytes	

Módulo_17 – Endereço IP: 192.168.20.18, Tempo de Processamento: 2 μ s, Tempo de Transmissão Máximo: 0,7ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 31 – Tabela de execução do Módulo_17 – cenário 2.

Write	Read
Módulo_18: 60 bytes	Módulo_12: 37 bytes

Módulo_18 – Endereço IP: 192.168.20.19, Tempo de Processamento: 1 μ s, Tempo de Transmissão Máximo: 0,7ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 32 – Tabela de execução do Módulo_18 – cenário 2.

Write	Read
Módulo_19: 60 bytes	Módulo_14: 35 bytes

Módulo_19 – Endereço IP: 192.168.20.20, Tempo de Processamento: 2 μ s, Tempo de Transmissão Máximo: 1ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 33 – Tabela de execução do Módulo_19 – cenário 2.

Write	Read
Módulo_0: 45 bytes	

Para justificar a configuração da taxa de transmissão do segundo cenário em 100Mbps, para uma aplicação de 4ms, é possível realizar, de maneira análoga ao cenário um, uma análise prévia dos parâmetros temporais da proposta para verificação da adequação desse parâmetro de comunicação ao cenário proposto. Assim, de acordo com as condições evidenciadas nas tabelas 14 a 33 é possível calcular a taxa de transmissão teórica desse cenário para o tempo de varredura total proposto para a aplicação, de 4ms. O cálculo para obtenção da taxa de transmissão teórica do segundo cenário não considera os atrasos na rede causados pelos *switches*. Para o segundo cenário, a taxa de transmissão teórica calculada é de 11,5209889Mbps. Logo, como a rede Ethernet industrial possui três taxas de transmissão distintas que podem ser utilizadas, 10Mbps, 100Mbps e 1Gbps, a taxa de transmissão de 10Mbps não atende a essa aplicação e não pode ser utilizada, devendo-se, assim, utilizar, obrigatoriamente, as taxas de transmissão de 100Mbps ou 1Gbps para atender a aplicação desse cenário proposto. Assim, a taxa de 100Mbps escolhida pode ser utilizada para esse cenário.

A figura 51 ilustra a configuração física do cenário proposto, contendo as distâncias (em metros) entre cada trecho da rede configurada.

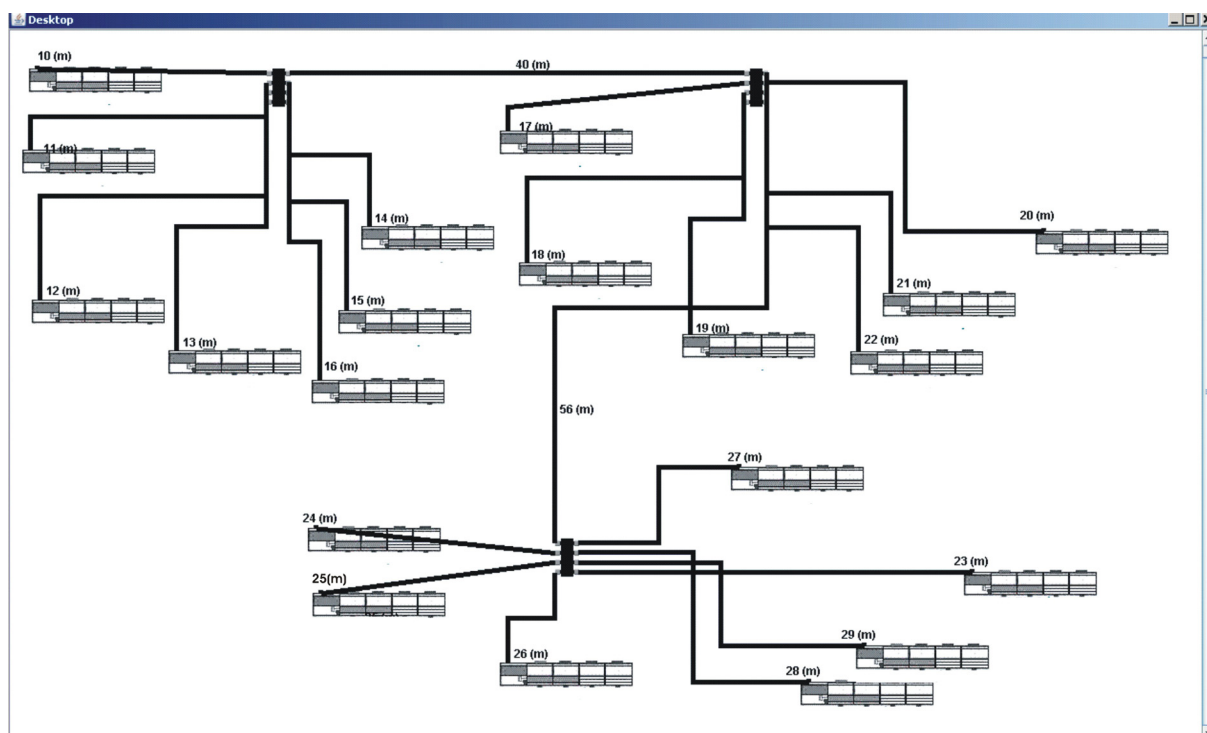


Figura 51 – Configuração física para o segundo cenário.

A figura 52 ilustra a tabela de escalonamento das mensagens cíclicas no barramento para o cenário proposto com vinte módulos de campo e controle distribuído entre eles. As mensagens a serem escalonadas, trinta e cinco ao todo, após o carregamento da tabela de escalonamento na rede, deverão ser executadas na sequência ilustrada na figura 52, tendo os tempos de execução indicados na coluna três e os períodos de liberação (*releases*) na coluna cinco (a unidade de ambos é milissegundos). As execuções das mensagens ocorrem de acordo com a ordenação proposta pelo algoritmo, dependendo do tipo de informação a ser trafegada no barramento.

Nesse cenário, três módulos de campo possuem dupla função, além de executarem a estratégia de controle realizam, simultaneamente, o controle do escalonamento da solução proposta, após definição prévia da sequência a ser executada. O módulo de campo dois realiza o controle, sendo que os módulos de campo três e sete são os reservas, em caso de falha do módulo principal.

Logo, a representação, em milissegundos, seguindo a notação $\tau_i = \tau_i (T_i; C_i; R_i)$ (duração da mensagem; duração máxima da mensagem ou *deadline*; tempo de liberação, ou *release*, da mensagem atual), para o escalonamento realizado na figura 52, tipo RM, é: $\tau_0 = \tau_0 (0.01692; 0.6; 0)$, $\tau_1 = \tau_1 (0.01772; 0.5; 0.01692)$, $\tau_2 = \tau_2 (0.01888; 0.9; 0.03464)$, $\tau_3 = \tau_3 (0.01952; 1; 0.05352)$, $\tau_4 = \tau_4 (0.01972; 0.7; 0.07304)$, $\tau_5 = \tau_5 (0.01972; 1; 0.09276)$, $\tau_6 = \tau_6 (0.02032; 1; 0.11248)$, $\tau_7 = \tau_7 (0.02036; 1; 0.1328)$, $\tau_8 = \tau_8 (0.02036; 1.5; 0.15316)$, $\tau_9 = \tau_9 (0.02072; 0.7; 0.17352)$, $\tau_{10} = \tau_{10} (0.02116; 0.6; 0.19424)$, $\tau_{11} = \tau_{11} (0.02272; 1.1; 0.2154)$, $\tau_{12} = \tau_{12} (0.2332; 1.2; 0.23812)$, $\tau_{13} = \tau_{13} (0.02392; 0.6; 0.26144)$, $\tau_{14} = \tau_{14} (0.02392; 0.7; 0.28536)$, $\tau_{15} = \tau_{15} (0.02432; 1; 0.30928)$, $\tau_{16} = \tau_{16} (0.02452; 0.9; 0.3336)$, $\tau_{17} = \tau_{17} (0.02452; 1; 0.35812)$, $\tau_{18} = \tau_{18} (0.02592; 0.5; 0.38264)$, $\tau_{19} = \tau_{19} (0.02732; 1.4; 0.43588)$, $\tau_{20} = \tau_{20} (0.02772; 1.1; 0.43628)$, $\tau_{21} = \tau_{21} (0.02772; 1.3; 0.4636)$, $\tau_{22} = \tau_{22} (0.02832; 1; 0.49132)$, $\tau_{23} = \tau_{23} (0.02952; 1.4; 0.51964)$, $\tau_{24} = \tau_{24} (0.03036; 1.2; 0.54916)$, $\tau_{25} = \tau_{25} (0.03092; 1; 0.57952)$, $\tau_{26} = \tau_{26} (0.03092; 0.5; 0.61044)$, $\tau_{27} = \tau_{27} (0.03092; 0.6; 0.64136)$, $\tau_{28} = \tau_{28} (0.03112; 1.5; 0.67228)$, $\tau_{29} = \tau_{29} (0.03136; 1.5; 0.7034)$, $\tau_{30} = \tau_{30} (0.03292; 0.8; 0.73476)$, $\tau_{31} = \tau_{31} (0.04393; 0.8; 0.76768)$, $\tau_{32} = \tau_{32} (0.04393; 0.9; 0.81161)$, $\tau_{33} = \tau_{33} (0.05592; 0.9; 0.85554)$ e $\tau_{34} = \tau_{34} (0.06692; 1; 0.91146)$.

Índice	ID	Tempo de Execução (ms)	Ordenação das mensagens	Período de liberação (ms)
0	de Módulo 1 para Módulo 11	0.01692	1	0.0
1	de Módulo 14 para Módulo 18	0.01772	1	0.01692
2	de Módulo 12 para Módulo 17	0.01888	1	0.03464
3	de Módulo 19 para Módulo 0	0.01952	1	0.05352
4	de Módulo 18 para Módulo 19	0.01972	1	0.07304
5	de Módulo 11 para Módulo 15	0.01972	1	0.09276
6	de Módulo 11 para Módulo 12	0.02032	1	0.11248
7	de Módulo 16 para Módulo 17	0.02036	1	0.1328
8	de Módulo 10 para Módulo 13	0.02036	1	0.15316
9	de Módulo 17 para Módulo 18	0.02072	1	0.17352
10	de Módulo 15 para Módulo 16	0.02116	1	0.19424
11	de Módulo 6 para Módulo 5	0.02272	1	0.2154
12	de Módulo 7 para Módulo 6	0.02332	1	0.23812
13	de Módulo 1 para Módulo 2	0.02392	1	0.26144
14	de Módulo 2 para Módulo 1	0.02392	1	0.28536
15	de Módulo 5 para Módulo 6	0.02432	1	0.30928
16	de Módulo 4 para Módulo 5	0.02452	1	0.3336
17	de Módulo 5 para Módulo 4	0.02452	1	0.35812
18	de Módulo 14 para Módulo 15	0.02592	1	0.38264
19	de Módulo 9 para Módulo 8	0.02732	1	0.43588
20	de Módulo 6 para Módulo 7	0.02772	1	0.43628
21	de Módulo 8 para Módulo 9	0.02772	1	0.4636
22	de Módulo 11 para Módulo 10	0.02832	1	0.49132
23	de Módulo 9 para Módulo 10	0.02952	1	0.51964
24	de Módulo 7 para Módulo 8	0.03036	1	0.54916
25	de Módulo 11 para Módulo 12	0.03092	1	0.57952
26	de Módulo 0 para Módulo 1	0.03092	1	0.61044
27	de Módulo 1 para Módulo 0	0.03092	1	0.64136
28	de Módulo 10 para Módulo 11	0.03112	1	0.67228
29	de Módulo 10 para Módulo 9	0.03136	1	0.7034
30	de Módulo 3 para Módulo 2	0.03292	1	0.73476
31	de Módulo 3 para Módulo 4	0.04393	1	0.76768
32	de Módulo 4 para Módulo 3	0.04393	1	0.81161
33	de Módulo 12 para Módulo 13	0.05592	1	0.85554
34	de Módulo 13 para Módulo 14	0.06692	1	0.91146
				0.97838

Figura 52 – Tabela de escalonamento para o segundo cenário.

A figura 53 analisa o tempo total consumido nesse cenário, evidenciando a soma total cíclica e as individuais, de cada módulo de campo. A seguir há a análise temporal calculado pela ferramenta, aplicada a cada módulo de campo.

1-) Módulo_0 = 30,92 μ s: índice 26 da tabela de escalonamento. O tempo restante, de 469,08 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

2-) Módulo_1 = 71,76 μ s: índice 0 + índice 13 + índice 27 da tabela de escalonamento (16,92 μ s + 23,92 μ s + 30,92 μ s). O tempo restante, de 528,24 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

3-) Módulo_2 = 23,92 μ s: índice 14 da tabela de escalonamento. O tempo restante, de 676,08 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

4-) Módulo_3 = 76,85 μ s: índice 31 + índice 31 da tabela de escalonamento (32,92 μ s + 43,93 μ s). O tempo restante, de 72,315 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

5-) Módulo_4 = 68,45 μ s: índice 16 + índice 32 da tabela de escalonamento (24,52 μ s + 43,93 μ s). O tempo restante, de 831,55 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

6-) Módulo_5 = 48,84 μ s: índice 15 + índice 17 da tabela de escalonamento (24,32 μ s + 24,52 μ s). O tempo restante, de 951,16 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

7-) Módulo_6 = 50,44 μ s: índice 11 + índice 20 da tabela de escalonamento (22,72 μ s + 27,72 μ s). O tempo restante, de 1,04956ms, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

8-) Módulo_7 = 53,68 μ s: índice 12 + índice 24 da tabela de escalonamento (23,32 μ s + 30,36 μ s). O tempo restante, de 1,14632ms, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

9-) Módulo_8 = 27,72 μ s: índice 21 da tabela de escalonamento. O tempo restante, de 1,27228ms, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

10-) Módulo_9 = 56,84 μ s: índice 19 + índice 23 da tabela de escalonamento (27,32 μ s + 29,52 μ s). O tempo restante, de 1,34316ms, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

11-) Módulo_10 = 82,84 μ s: índice 8 + índice 28 + índice 29 da tabela de escalonamento (20,36 μ s + 31,12 μ s + 31,36 μ s). O tempo restante, de 1,41716ms, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

12-) Módulo_11 = 99,28 μ s: índice 5 + índice 6 + índice 22 + índice 25 da tabela de escalonamento (19,72 μ s + 20,32 μ s + 28,32 μ s + 30,92 μ s). O tempo restante, de 900,72 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

13-) Módulo_12 = 74,8 μ s: índice 2 + índice 33 da tabela de escalonamento (18,88 μ s + 55,92 μ s). O tempo restante, de 825,2 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

14-) Módulo_13 = 66,92 μ s: índice 34 da tabela de escalonamento. O tempo restante, de 933,08 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

15-) Módulo_14 = 43,64 μ s: índice 1 + índice 18 da tabela de escalonamento (17,72 μ s + 25,92 μ s). O tempo restante, de 456,36 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

16-) Módulo_15 = 21,16 μ s: índice 10 da tabela de escalonamento. O tempo restante, de 578,84 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

17-) Módulo_16 = 20,36 μ s: índice 7 da tabela de escalonamento. O tempo restante, de 979,64 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

18-) Módulo_17 = 20,72 μ s: índice 9 da tabela de escalonamento. O tempo restante, de 679,28 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

19-) Módulo_18 = 19,72 μ s: índice 4 da tabela de escalonamento. O tempo restante, de 680,28 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

20-) Módulo_19 = 19,52 μ s: índice 3 da tabela de escalonamento. O tempo restante, de 980,48 μ s, é a diferença entre a comunicação cíclica e o tempo máximo (*deadline*).

Logo, o tempo para a realização da comunicação cíclica total utilizada para esse cenário é de **978,38 μ s** (soma de todos os tempos acima).

O tempo disponível para a realização da comunicação acíclica ou para a realização de futuras ampliações na rede do cenário proposto é de: 4ms (tempo de varredura total, configurado inicialmente na aplicação) – 978,38 μ s = **3,02162ms**.

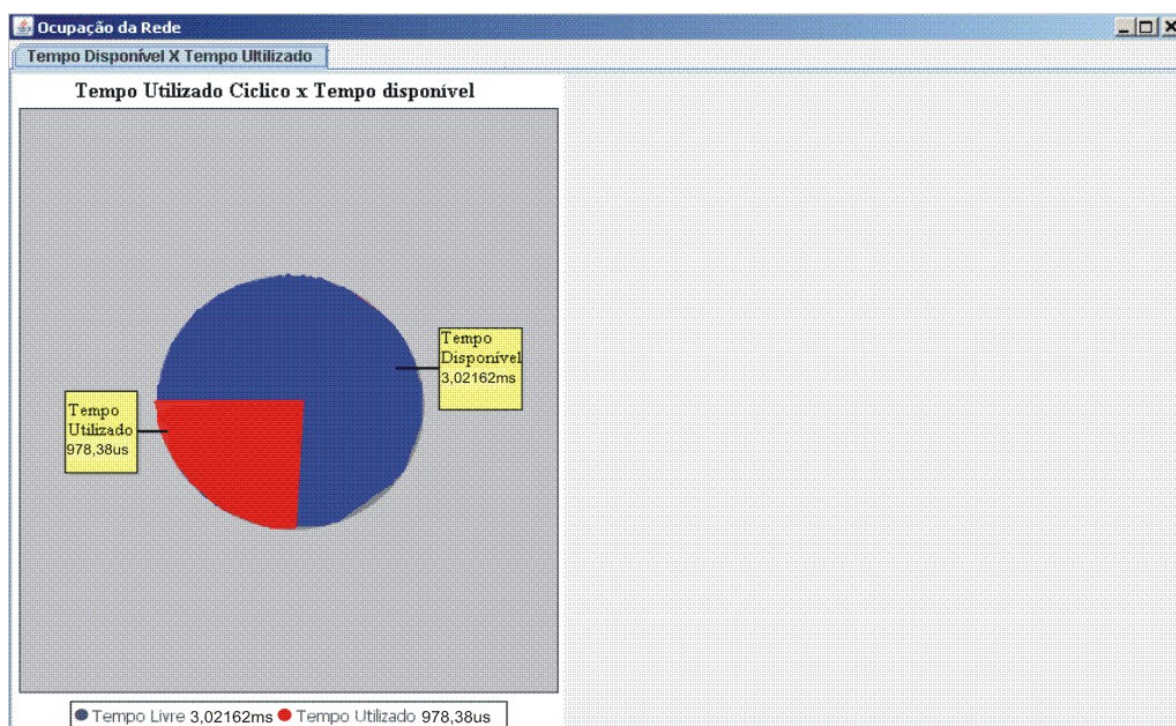


Figura 53 – Análise temporal total para o segundo cenário.

Da figura 52, é possível calcular cada custo computacional aplicado a cada mensagem a ser trafegada no barramento. Assim, os maiores custos devem ser executados de forma

imediate e disponibilizados no barramento para a execução correta do escalonamento do barramento. Os custos das trinta e cinco mensagens são:

- Índice 0: -0,58308ms.
- Índice 1: -0,48928ms.
- Índice 2: 0,88112ms.
- Índice 3: -0,98048ms.
- Índice 4: -0,68028ms.
- Índice 5: 0,98028ms.
- Índice 6: -0,97968ms.
- Índice 7: -0,97964ms.
- Índice 8: -1,47964ms.
- Índice 9: -0,67608ms.
- Índice 10: -0,5759ms.
- Índice 11: -1,07728ms.
- Índice 12: -0,67608ms.
- Índice 13: -0,57608ms.
- Índice 14: -0,67608ms.
- Índice 15: -0,97568ms.
- Índice 16: -1,37268ms.
- Índice 17: -0,97548ms.
- Índice 18: -0,47408ms.
- Índice 19: -1,37268ms.
- Índice 20: -1,07228ms.
- Índice 21: -1,27228ms.
- Índice 22: -0,97168ms.
- Índice 23: -1,37048ms.
- Índice 24: -1,16964ms.
- Índice 25: -0,96908ms.
- Índice 26: -0,46908ms.
- Índice 27: -0,56908ms.
- Índice 28: -1,46888ms.
- Índice 29: -1,46864ms.

- Índice 30: -0,76708ms.
- Índice 31: -0,3607ms.
- Índice 32: -0,4607ms.
- Índice 33: -0,84408ms.
- Índice 34: -0,93308ms.

A correta sequência, utilizando o algoritmo *Branch-and-Bound*, para a disponibilização das mensagens no barramento está evidenciada abaixo.

- Módulo 0: [Índice 26: início em 0ms].
- Módulo 1: [Índice 27: início em 0,07456ms, Índice 13: início em 0,10548ms, Índice 0: início em 0,1556ms].
- Módulo 2: [Índice 14: início em 0,21152ms].
- Módulo 3: [Índice 31: início em 0,28348ms; Índice 30: início em 0,32741ms].
- Módulo 4: [Índice 32: início em 0,41625ms; Índice 16: início em 0,8437ms].
- Módulo 5: [Índice 17: início em 0,5755ms; Índice 15: início em 0,60002ms].
- Módulo 6: [Índice 6: início em 0,62434ms; Índice 20: início em 0,70426ms, Índice 11: início em 0,73198ms].
- Módulo 7: [Índice 12: início em 0,1882ms; Índice 24: início em 0,81334ms].
- Módulo 8: [Índice 21: início em 0,78562ms].
- Módulo 9: [Índice 23: início em 0,54598ms; Índice 19: início em 0,86822ms].
- Módulo 10: [Índice 8: início em 0,89554ms; Índice 29: início em 0,9159ms, Índice 28: início em 0,94726ms].
- Módulo 11: [Índice 22: início em 0,25516ms; Índice 5: início em 0,66502ms, Índice 25: início em 0,7547ms].
- Módulo 12: [Índice 33: início em 0,36033ms; Índice 2: início em 0,46018ms].
- Módulo 13: [Índice 34: início em 0,47906ms].
- Módulo 14: [Índice 18: início em 0,03092ms, Índice 1: início em 0,5684ms].

- Módulo 15: [Índice 10: início em 0,1294ms].
- Módulo 16: [Índice 7: início em 0,64466ms].
- Módulo 17: [Índice 9: início em 0,16748ms].
- Módulo 18: [Índice 4: início em 0,23544ms].
- Módulo 19: [Índice 3: início em 0,68474ms].

6.1.3 Terceiro cenário

O objetivo do terceiro cenário é validar todo o novo conceito de controle distribuído flexível proposto no trabalho e, também, a ferramenta computacional indicada. A validação será feita através de uma aplicação real de campo que servirá como comparativo. Assim, é possível provar que essa nova proposta de arquitetura, com controle distribuído, pode ser aplicada a uma rede Ethernet industrial.

O desenvolvimento prático foi realizado com dois módulos de campo, ilustrados na figura 54, com a estrutura para uma rede Ethernet industrial com a proposta de controle distribuído, um *switch* com tempo de varredura configurado em 1ms e taxa de transmissão de 10Mbps. Os bytes de escrita e leitura representam sensores de temperatura e atuadores reais conectados aos módulos, contendo seis bytes de entrada e seis bytes de saída para cada módulo de campo.

O apêndice C ilustra parte do código desenvolvido, para esse cenário, visando o desenvolvimento do controle distribuído proposto.

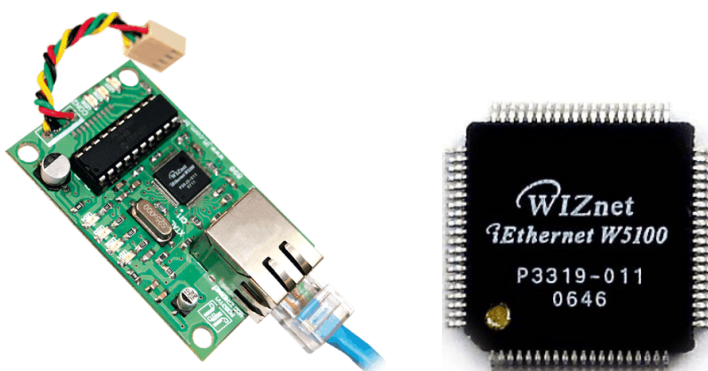


Figura 54 – Módulo de campo real para validação da proposta do trabalho.

As condições de cada módulo de campo para o cenário proposto estão evidenciadas nas tabelas 34 e 35.

Módulo_0 – Endereço IP: 192.168.1.200, Tempo de Processamento: 5 μ s, Tempo de Transmissão: 0,5ms (ordem 1, que suporta o controle da comunicação temporal).

Tabela 34 – Tabela de execução do Módulo_0 – cenário 3.

Write	Read
Módulo_1: 6 bytes	

Módulo_1 – Endereço IP: 192.168.1.201, Tempo de Processamento: 5 μ s, Tempo de Transmissão: 0,5ms (ordem 1, que não suporta o controle da comunicação temporal).

Tabela 35 – Tabela de execução do Módulo_1 – cenário 3.

Write	Read
Módulo_0: 6 bytes	

A comunicação dos pacotes trafegados entre os dois módulos de campo, com controle distribuído, pode ser visualizada na figura 55, através de uma ferramenta computacional específica para captura de mensagens na rede Ethernet, chamada de *WireShark*, ou analisador de protocolo para a rede Ethernet. ^[64] Assim, esta ferramenta de captura de protocolo evidencia os tempos de comunicação entre cada mensagem, de cada módulo de campo, sendo possível, também, visualizar o tempo total de comunicação do sistema. Esse tempo, que é o tempo de comunicação cíclica das mensagens trafegadas por todos os módulos de campo na rede, é de **145,4 μ s**. A quantidade de bytes totais capturados em cada comunicação para o tráfego, no barramento, das estratégias de controle é de setenta e dois bytes, sendo sessenta e seis bytes de cabeçalho (já explanado no item 5.2.1) e seis bytes de mensagens úteis dos sensores e atuadores.

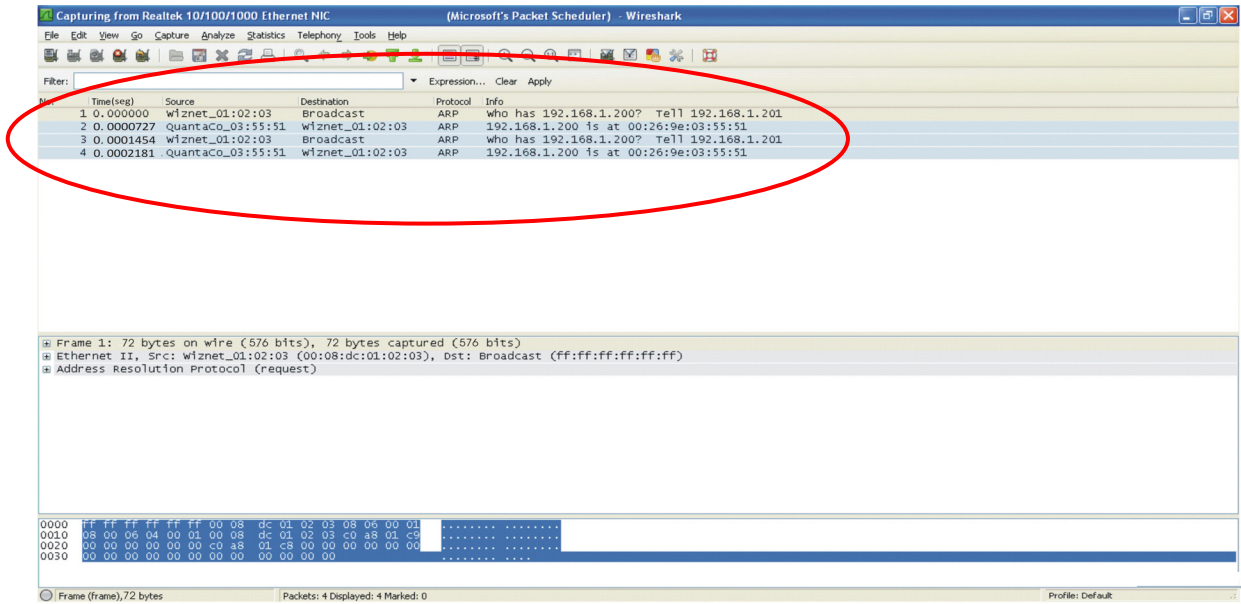
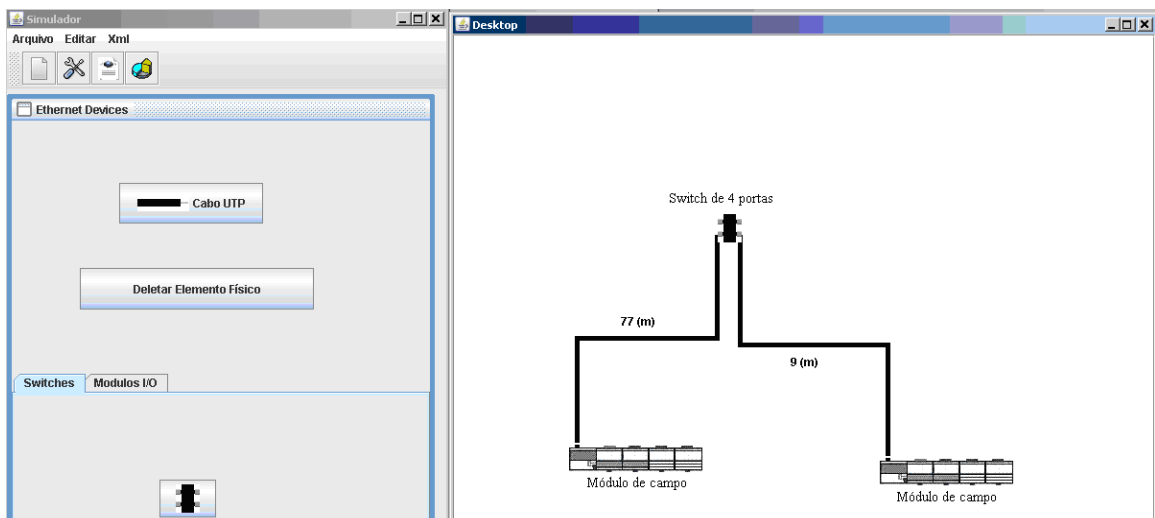


Figura 55 – Análise temporal para o cenário 3 utilizando a ferramenta Wireshark. [64]

A figura 56 ilustra as mesmas informações listadas acima, porém ilustradas através da ferramenta computacional e da arquitetura proposta nesse trabalho. É possível visualizar a parte física da rede, a tabela de escalonamento das mensagens no barramento entre os dois módulos de campo e o gráfico temporal cíclico total para as estratégias de controle dos módulos de campo. O tempo para a realização da comunicação cíclica total utilizada para esse cenário é de **144,4µs**.



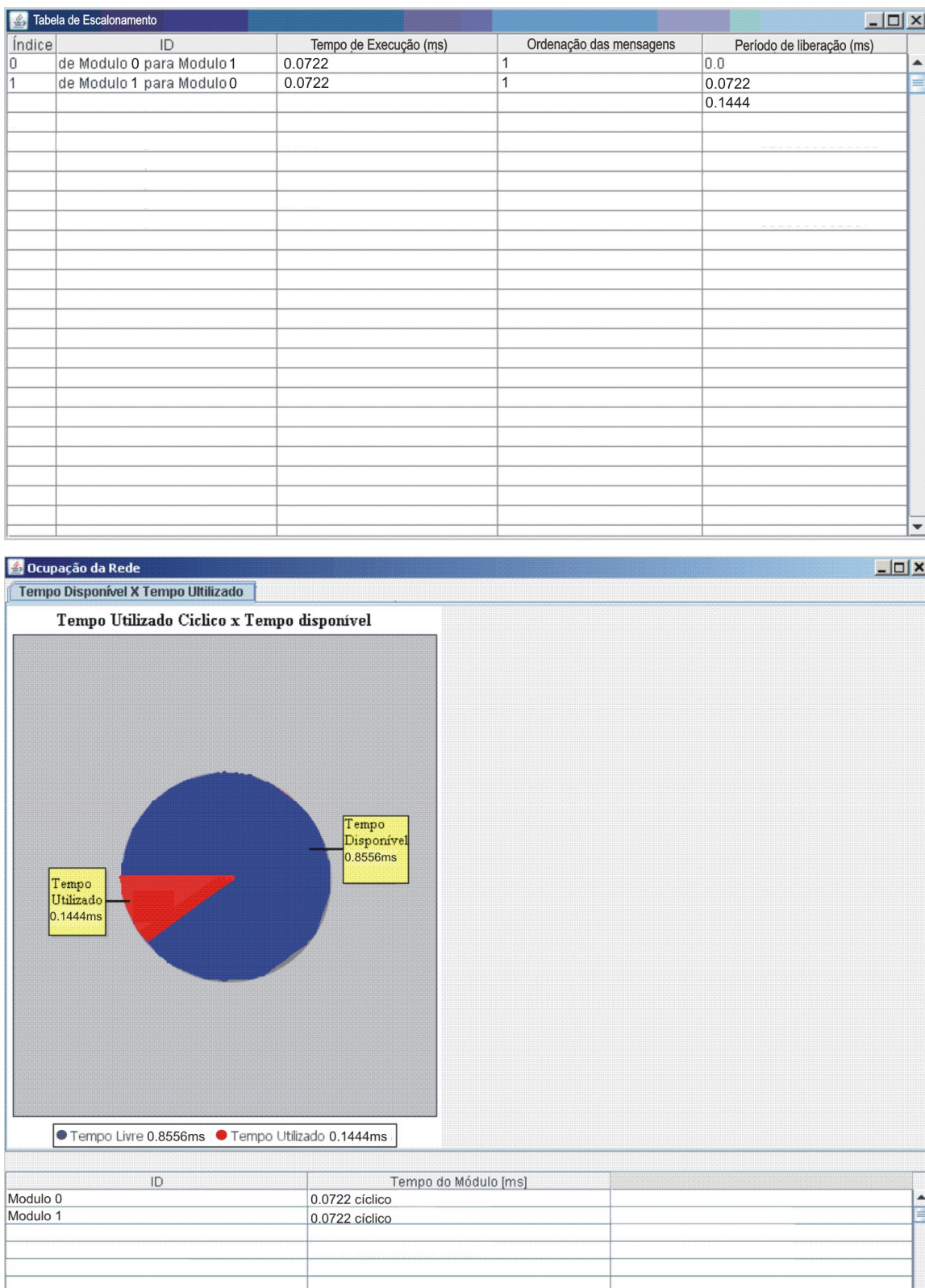


Figura 56 – Controle distribuído na rede Ethernet para o cenário 3.

É possível perceber, neste cenário, que a nova proposta de controle distribuído aplicada as redes Ethernet industriais está em funcionamento, em uma aplicação real, e que a ferramenta computacional realizou os cálculos de maneira bem próxima a comunicação real.

A diferença no tempo real medido pelo analisador de rede *Wireshark* e o tempo calculado pela ferramenta computacional foi de **1 μ s** (145,4 μ s e 144,4 μ s, respectivamente) e deve-se ao fato do tempo “*dead time*” (T_{dead}), ou tempo de atraso de comunicação do *switch* industrial, explicado anteriormente nos capítulos quatro e cinco, que, empiricamente, foi considerado de 9,6 μ s de acordo com as citações anteriores.

A figura 57 ilustra o escalonamento realizado em cada módulo de campo para cada uma das mensagens cíclicas trafegadas nesse cenário proposto, a fim de garantir que as mensagens escalonadas estejam prontas para serem transmitidas pelo barramento no instante em que forem executadas.

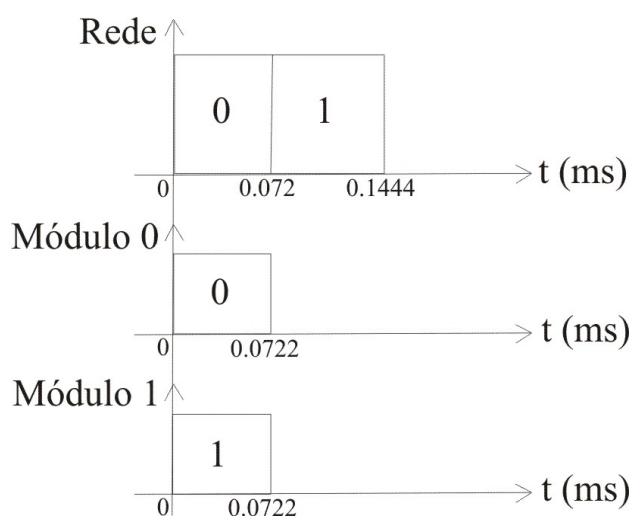


Figura 57 – Ilustração do escalonamento realizado em cada módulo de campo, cenário 3.

6.1.4 Quarto cenário

O objetivo do quarto cenário é realizar um comparativo entre a rede Ethernet industrial proposta nesse trabalho, com controle distribuído, e as redes Ethernet industriais do mercado em relação aos parâmetros temporais para uma mesma quantidade de informação trafegada no barramento.

A comparação foi realizada através de uma ferramenta computacional desenvolvida pela ODVA para simulação da rede industrial Ethernet/IP. ^[25] Os dois módulos computacionais representam o controlador mestre e o módulo de campo, respectivamente, ilustrados nas figuras 58 e 59. O módulo de campo é composto por dois bytes de entrada (dezesesseis chaves liga/desliga) e dois bytes de saída (dezesesseis lâmpadas vermelhas). Há, também, parâmetros acíclicos, do módulo de campo, que podem ser lidos pelo controlador mestre, via protocolo TCP, conforme já mencionado anteriormente. Esses parâmetros são:

Vendor ID, Device type, Product code, Serial number, Product name e Revision. Esses parâmetros podem ser configurados no módulo de campo e lidos no controlador mestre.

Na figura 58 há a tela do controlador mestre. O círculo inferior ilustra o conjunto de parâmetros acíclicos lidos do módulo de campo.

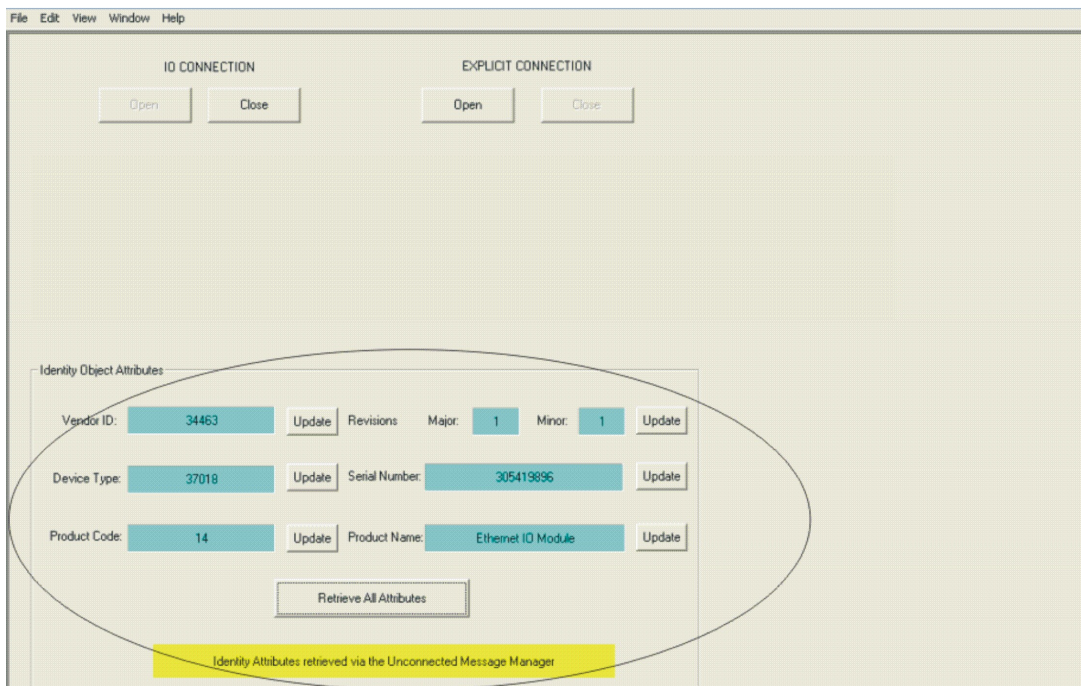


Figura 58 – Tela do controlador mestre Ethernet/IP da ODVA. [25]

Na figura 59 há a tela do módulo de campo. O círculo superior ilustra as lâmpadas e as chaves de entrada e no círculo inferior os parâmetros acíclicos do módulo de campo.

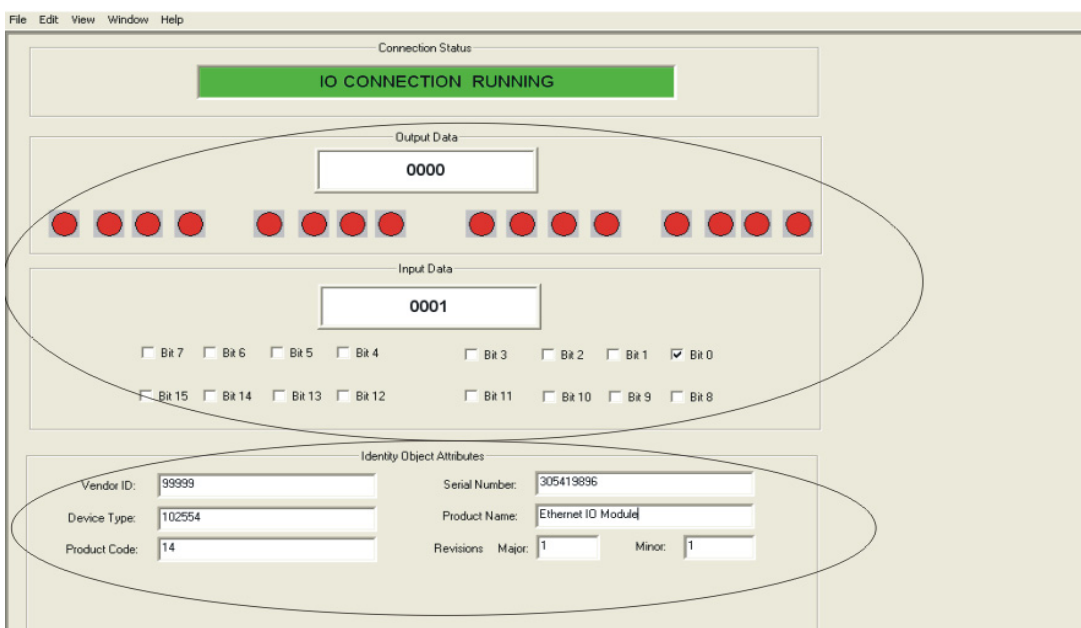


Figura 59 – Tela do módulo de campo Ethernet/IP da ODVA. [25]

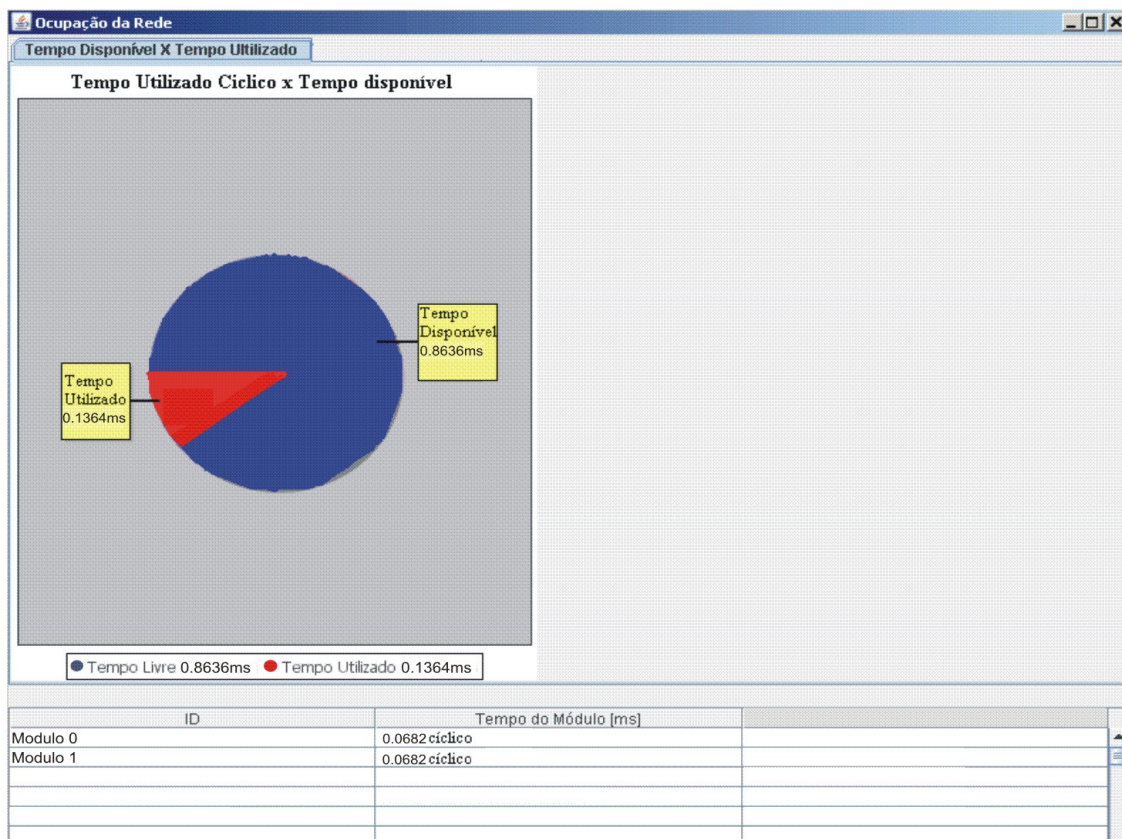


Figura 60 – Tabela de escalonamento e gráfico para o cenário 4.

A tabela 36 realiza uma comparação entre as duas propostas analisadas nesse cenário, a rede Ethernet/IP, com controle centralizado, e a rede Ethernet industrial proposta, com controle distribuído.

Tabela 36 – Comparativo entre as diversas redes Ethernet industriais.

	Ethernet industrial proposta	Ethernet/IP (ODVA)
Quantidade de módulos de campo	2	2
Quantidade de bytes de entrada e saída	2 IN + 2 OUT	2 IN + 2 OUT
Taxa de transmissão	10Mbps	10Mbps
Tempo de varredura	1ms	1ms
Tempo de <i>dead time</i> do <i>switch</i>	9,6µs	9,6µs
Tempo utilizado cíclico	136,4µs	125µs
Tempo disponível (acíclico)	863,6µs	875µs
Distância máxima sem repetidor	100 metros ^[25]	100 metros ^[25]

6.2 Conclusões dos resultados

O primeiro cenário evidenciou o novo conceito do controle distribuído proposto pelo trabalho, contendo cinco módulos de campo e nove estratégias de controle distintas escalonadas no barramento. O projetista tem a possibilidade de alterar os parâmetros de sua rede, através da ferramenta computacional, conforme o seu processo produtivo. O sistema proposto e modelado pela ferramenta computacional realizou a comunicação escalonada no barramento e entre os módulos de campo. O tempo de comunicação cíclico é de 1,035ms. Ainda foi realizada uma alteração no tempo de varredura do cenário (de 10ms para 1ms), a fim de verificar possíveis problemas com relação a comunicação das mensagens da solução proposta.

O segundo cenário mostrou-se eficaz, pois há a possibilidade de utilização de diversos módulos de campo, realizando um controle distribuído aplicando, ainda, os algoritmos de escalonamento propostos. Foram utilizados vinte módulos de campo e trinta e cinco estratégias de controle distintas. O tempo para comunicação cíclica é de 978,38 μ s, para uma taxa de transmissão de 100Mbps.

O terceiro cenário comportou-se bem próximo ao sistema real proposto com dois módulos de campo e duas estratégias de controle (sendo de seis bytes para cada uma das estratégias propostas), de acordo com as medidas dos respectivos tempos. O tempo de varredura cíclico ilustrado pela ferramenta computacional é de 144,4 μ s e o tempo de varredura cíclico medido nos módulos de campo reais é de 145,4 μ s. A medida do tempo cíclico real foi conseguida utilizando um analisador de rede (*WireShark*) nas mesmas condições da rede proposta pela ferramenta (taxa de transmissão, endereço IP, quantidade de informação trafegada e tempo de processamento). Logo, conclui-se que a solução proposta está de acordo com os módulos reais desenvolvidos, validando, assim, todo o trabalho proposto.

O quarto cenário ilustra um comparativo entre a Ethernet industrial proposta, com controle distribuído, e a rede industrial Ethernet/IP (padronizada pela ODVA), a fim de enriquecer o trabalho realizado e evidenciar os tempos envolvidos em cada um dos sistemas propostos. Para uma mesma quantidade de informação trafegada nas redes propostas (dois bytes), cada uma das redes se comportou de uma maneira com relação ao tempo de comunicação cíclico. A tabela 36 ilustra as características e os resultados para as duas redes analisadas: Ethernet industrial proposta e Ethernet/IP.

Então, com os cenários propostos, pode-se perceber:

- O funcionamento e operação do controle distribuído proposto, aplicado às mensagens do barramento Ethernet, e a validação do desenvolvimento do trabalho comparado aos módulos de campo reais desenvolvidos.
- O escalonamento aplicado no barramento e em cada módulo de campo de forma distinta.
- A solução dos problemas lógicos, citados no capítulo quatro, com a proposta do novo controle distribuído.
- A solução dos problemas físicos, citados no capítulo quatro, com a proposta do novo controle distribuído, com exceção de descargas elétricas externas.
- A comparação dos resultados da solução proposta, com relação a outra rede Ethernet industrial do mercado (Ethernet/IP).
- O método de escalonamento do barramento proposto tem como desvantagem a não atualização da tabela de escalonamento de forma dinâmica, ou *on-line*. Assim, no método proposto, a cada alteração na lista de módulos de campo ativos no barramento ou de algum parâmetro local ou global é necessário obter uma nova tabela de escalonamento do barramento.

6.3 Dificuldades encontradas no desenvolvimento do trabalho

A dificuldade inicial relacionou-se ao ineditismo do trabalho. Através de visitas técnicas reais e experiências práticas foi possível verificar a grande necessidade de um maior conhecimento da tecnologia utilizada, com um baixo tempo cíclico e custo para implantação do sistema proposto.

A segunda dificuldade relacionou-se ao desenvolvimento do novo conceito de comunicação para as redes Ethernet industrial, sua modelagem e análise. E, posteriormente, a escolha da plataforma para desenvolvimento da ferramenta computacional para a visualização da proposta, no caso, *Java*.

A terceira dificuldade foi conceber o sistema real, com os módulos de campo, para validação da proposta do controle distribuído flexível.

Por último, foi necessário realizar as seguintes averiguações:

- Testes funcionais na ferramenta para verificar seu funcionamento.
- Comparação com um sistema real para aceitação dos resultados.
- Realização de inúmeros tratamentos de exceções para impossibilitar o projetista a operar com valores irreais dentro da ferramenta.

7 CONCLUSÕES E CONTRIBUIÇÕES

O ineditismo do trabalho proposto pode ser verificado através do desenvolvimento de um algoritmo para a realização do escalonamento das mensagens do barramento e de cada módulo de campo, aplicado as redes Ethernet industrial, com controle distribuído das mensagens do barramento, possibilitando uma maior velocidade e volume no tráfego de mensagens cíclicas e acíclicas da rede, devido a sua característica temporal proposta, e resolvendo, assim, os problemas físicos (exceto descargas elétricas externas) e lógicos com a utilização de uma rede centralizada. Além disso, considera-se também, as mensagens cíclicas e acíclicas trafegadas na rede Ethernet e o escalonamento realizado em cada módulo de campo.

Para validar a idéia proposta, foi desenvolvida uma ferramenta computacional para simulação e visualização dos conceitos aqui apresentados e, além disso, um sistema real para comparação dos resultados. Os resultados obtidos nas duas situações, real e simulada, validam o correto funcionamento da proposta desse trabalho de controle distribuído das mensagens do barramento da rede Ethernet.

No trabalho proposto, a visualização, as características e as situações propostas foram atendidas de acordo com o novo modelo de comunicação, com o controle distribuído proposto. Os cenários propostos evidenciam o controle distribuído das mensagens no barramento e quais são as opções que o projetista possui com relação a quantidade de informação que se deseja trafegar em sua rede em função do tempo para transmissão e recepção das mensagens. Assim, no momento da instalação e configuração da rede, muitos problemas relativos ao projeto já foram discutidos e analisados, minimizando o tempo de partida e os custos adicionais da planta industrial. Foi possível, ainda, realizar um estudo comparativo entre a proposta de controle distribuído das mensagens do barramento e uma rede Ethernet real, com controle centralizado.

Outra contribuição a sociedade é o fato das propostas de escalonamento para a realização do controle distribuído das mensagens no barramento e nos módulos de campo não estarem atreladas a um único padrão de rede Ethernet industrial, como por exemplo, uma aplicação muito específica envolvendo PROFINET IRT.

A partir do término deste trabalho é possível realizar outras análises futuras, que poderão ser desenvolvidas a partir dessa proposta. A seguir, há a descrição de alguns possíveis trabalhos que poderão ser elaborados a partir dessa proposta.

- Alteração do método de escalonamento das mensagens cíclicas no barramento para a comparação entre os diversos tipos entre si, podendo-se, assim, medir a eficiência temporal de cada método de escalonamento existente aplicado a essa proposta, principalmente, a um método de escalonamento *on-line*.
- Alteração do método de escalonamento das mensagens cíclicas individuais de cada módulo de campo para a comparação entre os diversos tipos entre si, podendo-se, assim, medir a eficiência de processamento de cada método de escalonamento existente aplicado a essa proposta.
- Possibilidade de realização de um aperfeiçoamento na lógica das portas do *switch* industrial, utilizando as técnicas de inteligência artificial, a fim de melhorar a comunicação, com controle distribuído, e diminuir os atrasos no seu *buffer* de mensagens interno.
- Possibilidade de desenvolvimento de uma função extra para conexão da ferramenta computacional desenvolvida a uma rede Ethernet industrial real externa, com controle distribuído, desenvolvida com os conceitos dessa proposta. Nesse caso, a ferramenta teria a função de um analisador de rede real, realizando as medidas de tráfego, probabilidade de falha e da parte física, validando o processo produtivo e a rede em questão.

REFERÊNCIAS BIBLIOGRÁFICAS

-
- [1] CERF, V. G. and KHAN, R., E. **A Protocol for Packet Network Intercommunication.** EUA, IEEE Transaction on Communication, Vol. 22, Issue 5, 1-13p., 1974.
- [2] METCALF, R. M. and BOGGS, D. R. **Ethernet: Distributed Packet Switching for Local Computer Networks.** EUA, IEEE Transaction on Communications, Vol. 19, Issue 7, 395-404p., 1976.
- [3] LUGLI, A. B. e SANTOS, M. M. D. **Sistemas de Fieldbus para Automação Industrial: DeviceNet, CANOpen, SDS e Ethernet.** Editora Érica, São Paulo, 2009, 156p.
- [4] SAUTER, T. **The Three Generations of Field-Level Networks – Evolution and Compability Issues.** Austria, IEEE Transactions on Industrial Electronics, Vol. 57, Issue 11, 3585-3595p., 2010.
- [5] FELSER, M. and SAUTER, T. **The Fieldbus War: History or Short Break between Battles?** Sweden, 4th IEEE International Workshop on Factory Communication System, 73-80p., 2002.
- [6] Site da Internet: **International Society of Automation**, disponível em: <www.isa.org>. Acessado em out. 2011.
- [7] Site da Internet: **International Engineering Consortium**, disponível em: <www.iec.org>. Acessado em set. 2011.
- [8] Site da Internet: **Associação PROFIBUS e PROFINET Internacional**, disponível em: <www.profibus.org>. Acessado em out. 2011.
- [9] FRENCH ASSOCIATION FOR STANDARDIZATION. FIP. **Bus for Exchange of Information between Transmitters, Actuators and Programmable Controllers**, NF C46601-C46607, 1989-1992p.

-
- [10] IEC, **Digital data communications for measurement and control**. IEC 61784-1 Normative, 1st edition, 2003.
- [11] BROOKS, P. **Ethernet/IP – Industrial Protocol**. France, 8th IEEE International Conference on Emerging Technologies and Factory Automation, Vol. 2, 505-514p., 2001.
- [12] POPP, M. and WEBBER, C. **The Rapid Way to PROFINET**. Book, PROFIBUS Organization, order 4182, 2004, 244p.
- [13] COMER, D. E. **Internetworking with TCP/IP – Vol. I – Principles, Protocols and Architecture**. 4th ed., Prentice-Hall Editor, USA, 2000, 750p.
- [14] THOMESSE, J. P. **Fieldbus Technology in Industrial Automation**. France, Proceedings of IEEE, Vol. 93, Issue 6, 1073-1101p., 2005.
- [15] Site da Internet: MADREN F. **Power over Ethernet – The Last Frontier in Industrial Communications Standards**, PoE articles, President GarrettCom, 26th May 2006, disponível em: <www.poweroverethernet.com/articles.php?article_id=354>. Acessado em ago. 2011.
- [16] Site da Internet: Texas Instruments – PoE solutions, disponível em: <www.ti.com/poe>. Acessado em ago. 2011.
- [17] Site da Internet: MORGAN, T. **Power over Ethernet - The Reality of Designing a Powered Device**, PoE articles, Senior Applications Engineer, Silver Telecom, 2006. disponível em: <www.poweroverethernet.com/articles.php?article_id=370>. Acessado em ago. 2011.
- [18] Site da Internet: **Institute of Electrical and Electronics Engineers**, padrão IEEE 802.3af, disponível em: <www.ieee802.org/3/af/index.html>. Acessado em ago. 2011.
- [19] SOARES, L. F., LEMOS, G. e COLCHER, S. **Redes de Computadores: das LANs, MANs e WANs às Redes ATM**. 2^o ed., Editora Campus, São Paulo, 2000, 705p.

[20] ROSE, M. T. **The Open Book: a Practical Perspective on OSI**. 1st ed., Book, Prentice-Hall Editor, USA, 1990, 400p.

[21] HELD, G. **Ethernet Networks**. 2nd ed., Book, Wiley Editor, London, 2000, 458p.

[22] Site da Internet: POSTEL, J. **RFC 791 – Internet Protocol, DARPA Internet Program Protocol Specification**, University of Southern Califórnia, USA, September 1981, disponível em: <<ftp://ftp.rfc-editor.org/in-notes/rfc791.txt>>. Acessado em jul. 2011.

[23] Site da Internet: **Institute of Electrical and Electronics Engineers**, IEEE 802.3 Ethernet Working Group, disponível em: <www.ieee802.org/3/>. Acessado em jul. 2011.

[24] Site da Internet: **Transmission Control Protocol – RFC 793**, University of Southern Califórnia, USA, September 1981, disponível em: <www.ibiblio.org/pub/docs/rfc/rfc793.txt>. Acessado em jul. 2011.

[25] Ethernet/IP Specification Release 5.0 - November, 2010. Norma Ethernet/IP elaborada pela ODVA, disponível em: <www.odva.org>. Acessado em ago. 2011.

[26] FELSER, M. and SAUTER T. **Standardization of Industrial Ethernet – the Next Battlefield?** Sweden, 6th IEEE International Workshop on Factory Communication Systems, 413-420p., 2004.

[27] ARJMANDI, F., et al. **Fieldbus Interoperability on Ethernet**. Austria, 5th IEEE International Conference on Industrial Informatics, 213-218p., 2007.

[28] Site da Internet: LARSSON, L. **Fourteen Industrial Ethernet solutions under the spotlight**, The Industrial Ethernet Books, Issue 28, September 2005, disponível em: <www.iebmedia.com/index.php?id=4811&parentid=74&themeid=255&hpid=2&showdetail=true&bb=1&appsw=1&sstr=larsson>. Acessado em ago. 2011.

[29] LUGLI, A. B. e SANTOS, M. M. D. **Redes Industriais para Automação Industrial: AS-I, PROFIBUS e PROFINET**. Editora Érica, São Paulo, 2010, 176p.

-
- [30] Site da Internet: Siemens AG Internacional. Curso Web de PROFINET, disponível em: <www.sitrain.com/modules/PROFINET_en/times/index_1024.htm>. Acessado em set. 2011.
- [31] JASPERNEITE, J. and FELD J. **PROFINET: An Integration Platform for Heterogeneous Industrial Communication Systems**. Germany, 10th IEEE International Conference on Emerging Technologies and Factory Automation, 815-822p., 2005.
- [32] PROFIBUS Working Group. **PROFINET, Architecture Description and Specification**. Norma PROFINET elaborada pela Associação PROFIBUS Internacional, Version 3.0, 2008.
- [33] PROFIBUS Working Group. **GSDML Specification for PROFINET IO**. Norma PROFINET elaborada pela Associação PROFIBUS Internacional, Version 3.1, 2008.
- [34] SKEIE, T., et al. **ETHERNET in Substation Automation**. IEEE Control Systems Magazine, Vol. 22, Issue 3, 43-51p., 2002.
- [35] FERRARI, A., et al. **Response Time Evaluation of PROFINET Networks**. Croatia, Proceedings of the IEEE International Symposium on Industrial Electronics, 1371-1376p., 2005.
- [36] ODVA Profile File, **Network Infrastructure for Ethernet/IP: Introduction and Considerations**. ODVA Publication PUB00035R0, 2007, 118p.
- [37] MARSHALL, P. S. **Industrial Ethernet: A Pocket Guide**. ISA International Editor, USA, 2002, 194p.
- [38] LUGLI, A. B., SANTOS, M. M. D. and FRANCO, L. R. H. R. **A Computer Tool to Support in Design of Industrial Ethernet**. USA, ISA Transactions, Journal 48, 228-236p., 2009.

[39] SCHAFER, I and FELSER, M. **Precision of Ethernet Measurements based on Software Tolls**. Greece, 12th IEEE International Conference on Emerging Technologies and Factory Automation, 510-515p., 2007.

[40] HOON, P. S., et al. **FOUNDATION Fieldbus High Speed Ethernet (HSE) Implementation**. Canada, IEEE International Symposium on Intelligent Control, 777-782p., 2002.

[41] HSE Specification Release 3.17 - March, 2011. Norma HSE elaborada pela Fieldbus Foundation, disponível em: < www.fieldbus.org >. Acessado em set. 2011.

[42] Sercos III Specification Release 2.21 - March, 2009. Norma Sercos III elaborada pela Sercos International, disponível em: < www.sercos.de >. Acessado em set. 2011.

[43] ZWEBEN, M and FOX, M. S. **Intelligent Scheduling**. Morgan Kaufmann Editor, USA, 1994, 450p.

[44] PINEDO, M. L. **Scheduling – Theory, Algorithms and Systems**. 3rd Edition, Prentice-Hall Editor, USA, 2008, 678p.

[45] HENRIQUES, A. M. **Algoritmo de Escalonamento da Comunicação para o Fieldbus Foundation**. 2000. 87f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Itajubá, Itajubá, Brasil.

[46] FRANCO, L. R. H. R. **Escalonamento de Mensagens para a Comunicação no Fieldbus**. 1998. 124f. Tese (Doutorado em Engenharia Elétrica) - Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil.

[47] CICILLINI, D. A. **Desenvolvimento de um Algoritmo de Escalonamento para a rede FOUNDATION FIELDBUS**. 2007. 134f. Dissertação (Mestrado em Engenharia Mecânica) - Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, Brasil.

-
- [48] ALMEIDA L., PASSADAS, R. and FONSECA, J. A. **Using a Planning Control Scheduler to Improve the Flexibility of Real-time Fieldbus Networks.** Canada, IEEE Engineering and Practice Magazine, Vol. 7, 101-108p., 1999.
- [49] Site da Internet: Empresa Sense Eletrônica, relatório técnico de visita a cliente, disponível em: <www.sense.com.br>. Acessado em jul. 2011.
- [50] TOVAR, E. M. M. **Supporting Real-Time Communications with Standard Factory-Floor Networks.** 1999. 170f. Tese (Doutorado em Engenharia Eletrotécnica e de Computadores) - Faculdade de Engenharia da Universidade do Porto, Porto, Portugal.
- [51] ADDAD, B., et al. **Analytic Calculus of Response Time in Networked Automation Systems.** France, IEEE Transactions on Automation Science and Engineering, Vol. 7, Issue 4, 858-869p., 2010.
- [52] ZHANG, L. and WANG, Z. **Real-time Performance Evaluation in Hybrid Industrial Ethernet Networks.** China, 8th IEEE Proceedings of the World Congress on Intelligent Control and Automation, 1842-1845p., 2010.
- [53] CARVALHO, H. F. **Escalonamento em redes Ethernet Industriais.** 2007. 100f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Itajubá, Itajubá, Brasil.
- [54] GADERER, G., et al. **Improving Fault Tolerance in High-Precision Clock Synchronization.** Austria, IEEE Transaction on Industrial Informatics, Vol. 6, Issue 2, 206-215p., 2010.
- [55] LEE, K. C., et al. **Worst Case Communication Delay of Real-Time Industrial Switched Ethernet With Multiple Levels.** IEEE Transactions on Industrial Electronics, Vol. 53, Issue 5, 1669-1676p., 2006.

-
- [56] MOYNE, J. R. and TILBURY, D. M. **The Emergence Industrial Control Networks for Manufacturing Control, Diagnostics, and Safety Data**. Proceedings of the IEEE, Vol. 95, Issue 1, 29-47p., 2007.
- [57] YOUN, G., et al. **Ring Topology-based Redundancy Ethernet for Industrial Networks**. Korea, International Joint Conference of the IEEE, 1404-1407p., 2006.
- [58] Site da Internet: Fabricante de cabo para rede PROFINET e Ethernet/IP, **Lapp Cable – Section 8, Bus Cables & Industrial Ethernet Cables**, disponível em: <www.lappusa.com>. Acessado em ago. 2011.
- [59] SCHILDT, H. **Java: The Complete Reference**. McGraw-Hill Editor, USA, 2005, 1000p.
- [60] DEITEL, H. M. e DEITEL, P. J. **Java: Como Programar**. 4º ed., Editora Bookman, Porto Alegre, 2002, 1386p.
- [61] Site da Internet: Fórum Dicas BCB para auxílio em conceitos de programação e códigos fontes, disponível em: <www.dicasbcb.com.br>. Acessado em out. 2011.
- [62] GUEDES, G. T. A. **UML: Uma Abordagem Prática**. 2º ed., Editora Novatec, São Paulo, 2006, 320p.
- [63] PENDER, T. **UML: a Bíblia**. 2º ed., Editora Campus, São Paulo, 2004, 752p.
- [64] Site da Internet: Ferramenta computacional **WireShark – A Network Protocol Analyser**, disponível em: <www.wireshark.org>. Acessado em out. 2011.
- [65] LUGLI, A. B. **Software para Análise de Topologia e Tráfego para Redes Ethernet Industriais**. 2007. 94f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal de Itajubá, Itajubá, Brasil.

Apêndice A – Codificação das classes da solução proposta.

```
package br.inatel.simulador.ethernetdevices;

import br.inatel.simulador.controle.Controle;
import javax.swing.Icon;
import javax.swing.ImageIcon;

/**
 *
 * @author Administrador
 */
public final class EthernetDevice {

    private String id;
    private String name;
    private String kindOfDevice;
    private String kindOfDeviceToEasyControl;
    private Icon Icon;
    private int bytes_IN;
    private int bytes_OUT;
    private String endereco;
    private double tempo_processamento;
    private double tempo_transmissao;
    private double tempo_latencia;
    private int wireless_standard;
    private int prioridade;

    private boolean configured;

    int id_number;

    public EthernetDevice(Controle controle, String id, int contCellsAdd)
    {

        if (id.equals("flexio"))
        {
            this.setId(id+"_"+contCellsAdd);
            this.setIcon(new ImageIcon(getClass().getResource("/images/flexio.GIF")));
            this.setId_number(contCellsAdd);
            this.setKindOfDevice(id);
            this.setKindOfDeviceToEasyControl(id+"_"+contCellsAdd);
            this.setPrioridade(3);
        }

        if (id.equals("switch"))
```

```
{
    this.setId(id);
    this.setIcon(new ImageIcon(getClass().getResource("/images/switch4.GIF")));
    this.setId_number(contCellsAdd);
    this.setKindOfDevice(id);
    this.setKindOfDeviceToEasyControl(id+"_"+contCellsAdd);
}

if(id.equals("wireless"))
{
    this.setId(id+"_"+contCellsAdd);
    this.setIcon(new ImageIcon(getClass().getResource("/images/wireless.GIF")));
    this.setId_number(contCellsAdd);
    this.setKindOfDevice(id);
    this.setKindOfDeviceToEasyControl(id+"_"+contCellsAdd);
    this.setPrioridade(3);
}

if(id.equals("switch_8"))
{
    this.setId(id);
    this.setIcon(new ImageIcon(getClass().getResource("/images/switch8.GIF")));
    this.setId_number(contCellsAdd);
    this.setKindOfDevice(id);
    this.setKindOfDeviceToEasyControl(id+"_"+contCellsAdd);
}
}

public EthernetDevice(){

}

public String getId() {
    return id;
}

public void setId(String id) {
    this.id = id;
}

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public Icon getIcon() {
```

```
    return Icon;
}

public void setIcon(ImageIcon Icon) {
    this.Icon = Icon;
}

public int getBytes_IN() {
    return bytes_IN;
}

public void setBytes_IN(int bytes_IN) {
    this.bytes_IN = bytes_IN;
}

public int getBytes_OUT() {
    return bytes_OUT;
}

public void setBytes_OUT(int bytes_OUT) {
    this.bytes_OUT = bytes_OUT;
}

public boolean isConfigured() {
    return configured;
}

public void setConfigured(boolean configured) {
    this.configured = configured;
}

public String getEndereco() {
    return endereco;
}

public void setEndereco(String endereco) {
    this.endereco = endereco;
}

public double getTempo_processamento() {
    return tempo_processamento;
}

public void setTempo_processamento(double tempo_processamento) {
    this.tempo_processamento = tempo_processamento/1000000;
}

public double getTempo_transmissao() {
    return tempo_transmissao;
}
```

```
}

public void setTempo_transmissao(double tempo_transmissao) {
    this.tempo_transmissao = tempo_transmissao/1000;
}

public double getTempo_latencia() {
    return tempo_latencia;
}

public void setTempo_latencia(double tempo_latencia) {
    this.tempo_latencia = tempo_latencia/1000;
}

public int getWireless_standard() {
    return wireless_standard;
}

public void setWireless_standard(int wireless_standard) {
    this.wireless_standard = wireless_standard;
}

public int getPrioridade() {
    return prioridade;
}

public void setPrioridade(int prioridade) {
    this.prioridade = prioridade;
}

public String getKindOfDevice() {
    return kindOfDevice;
}

public void setKindOfDevice(String kindOfDevice) {
    this.kindOfDevice = kindOfDevice;
}

public String getKindOfDeviceToEasyControl() {
    return kindOfDeviceToEasyControl;
}

public void setKindOfDeviceToEasyControl(String kindOfDeviceToEasyControl) {
    this.kindOfDeviceToEasyControl = kindOfDeviceToEasyControl;
}

public int getId_number() {
    return id_number;
}
```

```
public void setId_number(int id_number) {
    this.id_number = id_number;
}

@Override
public String toString()
{

    String ip;
    String id;

    if(this.isConfigured())
        ip=this.getEndereco();
    else
        ip="não configurado";
    if(this.kindOfDevice.equals("flexio"))
        id="Módulo_"+this.getId_number();
    else
        id=this.getId();

    return "Id: "+ id +" / IP: "+ip;

}
}
```

```
package br.inatel.simulador.interface_usuario;

import br.inatel.simulador.controle.Controle;
import java.awt.image.BufferedImage;
import javax.swing.ImageIcon;
import javax.swing.JLabel;
import org.jfree.chart.ChartFactory;
import org.jfree.chart.ChartFrame;
import org.jfree.chart.JFreeChart;
import org.jfree.chart.plot.PlotOrientation;
import org.jfree.data.category.DefaultCategoryDataset;
import org.jfree.data.general.DefaultPieDataset;

public class Graficos {

    public BufferedImage tempoLivre(Controle controle)
    {
        DefaultPieDataset dataset = new DefaultPieDataset();
        dataset.setValue("Tempo Livre "+controle.getTempoLivre()*1000+" ms",
            controle.getTempoLivre());
        dataset.setValue("Tempo Utilizado "+controle.getTempoTotal()*1000+" ms",
            controle.getTempoTotal());
    }
}
```

```

JFreeChart chart = ChartFactory.createPieChart(
    "Tempo Utilizado X Tempo Livre",
    dataset,
    true, // legend?
    true, // tooltips?
    false // URLs?
);
/*
ChartFrame frame = new ChartFrame("Simulador", chart);
frame.pack();
frame.setVisible(true);
*/
return chart.createBufferedImage(400,500);
}
public BufferedImage tempoModulo(Control control)
{
    DefaultCategoryDataset dataset = new DefaultCategoryDataset();
    for (int i=0;i<=control.indexLastDevice();i++)
    {
        if((control.getEthernetDevice(i).getKindOfDevice().equals("flexio"))||(control.getEthernetDevice(i).getKindOfDevice().equals("wireless")))
        {
            String mod = control.getEthernetDevice(i).getName();
            dataset.addValue(control.getTempoMod(control.getEthernetDevice(i))*1000,
                "Tempo [ms]" , mod);
        }
    }
    JFreeChart chart = ChartFactory.createBarChart(
        "Tempo por módulo", // chart title
        "Módulos", // domain axis label
        "Tempo[ms]", // range axis label
        dataset, // data
        PlotOrientation.VERTICAL,
        true, // include legend
        true, // tooltips?
        true // URLs?
    );

    ChartFrame frame = new ChartFrame("Simulador", chart);

    return chart.createBufferedImage(control.getConfiguracoes().getN_mod()*200,400);
}

void showGraficos(Control control) {

    BufferedImage imageDisponivel = this.tempoLivre(control);
    BufferedImage imageOcupacao = this.tempoModulo(control);

```

```
    FrameGraficos frameGraficos = new
    FrameGraficos(imageDisponivel,imageOcupacao,controle);

    frameGraficos.setVisible(true);
}
}

=====
//Frame de configurações gerais

package br.inatel.simulador.interface_usuario;

import br.inatel.simulador.controle.Configuracoes;
import javax.swing.JOptionPane;

public class FrameConfiguracoes extends javax.swing.JFrame {

    public FrameConfiguracoes(Configuracoes configuracoes) {
        initComponents();

        if(configuracoes.isConfigured())
        {
            textFieldN_mod.setText(Integer.toString(configuracoes.getN_mod()));
            textFieldN_switch.setText(Integer.toString(configuracoes.getN_switch()));
            textFieldScan.setText(Double.toString(configuracoes.getTempo_scan()*1000));

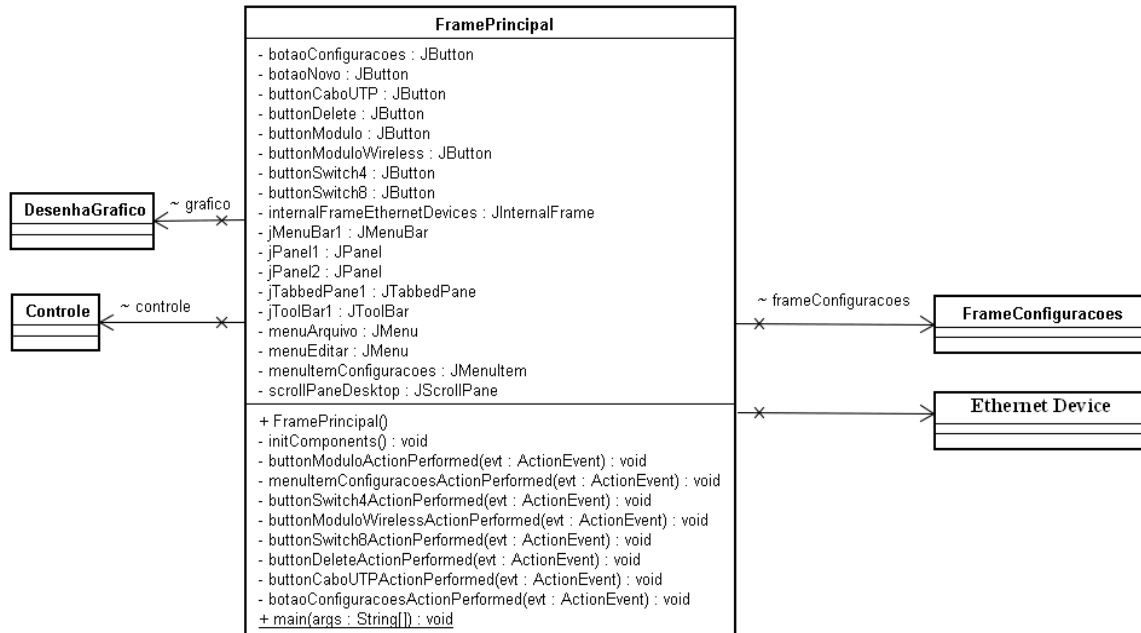
            comboBoxTrans.setSelectedItem(comboBoxTrans.getItemAt(getIdComboTrans((int)configuracoes.getTaxa_trans())));
        }
    }

    private int getIdComboTrans(int taxa_transmissao) {
        switch (taxa_transmissao)
        {
            case 10000000:
                return 0;
            case 100000000:
                return 1;
            case 1000000000:
                return 2;
        }
        return 0;
    }

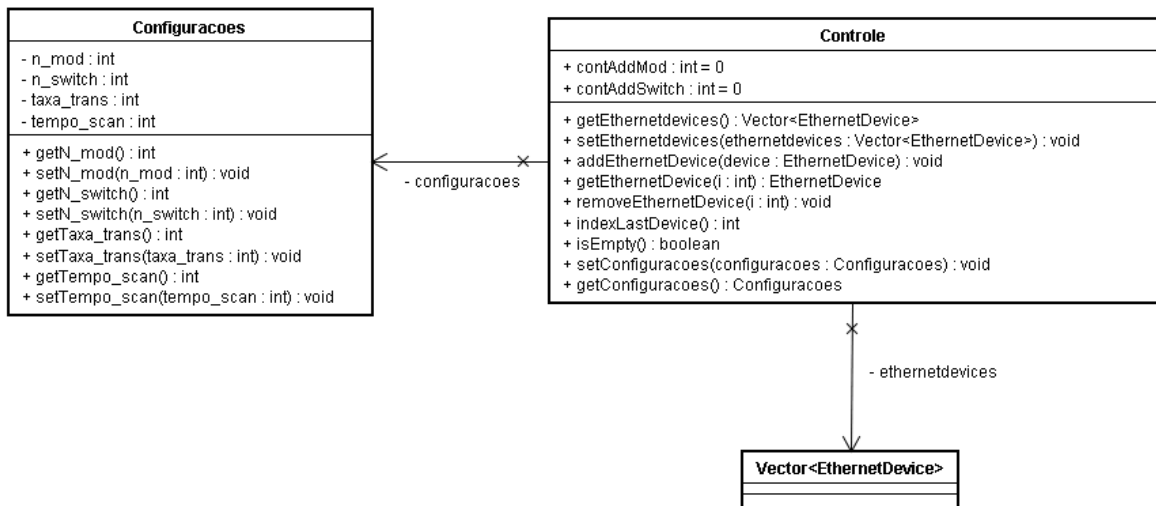
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN: initComponents
    private void initComponents() {
```


Apêndice B – Documentação UML da solução proposta.

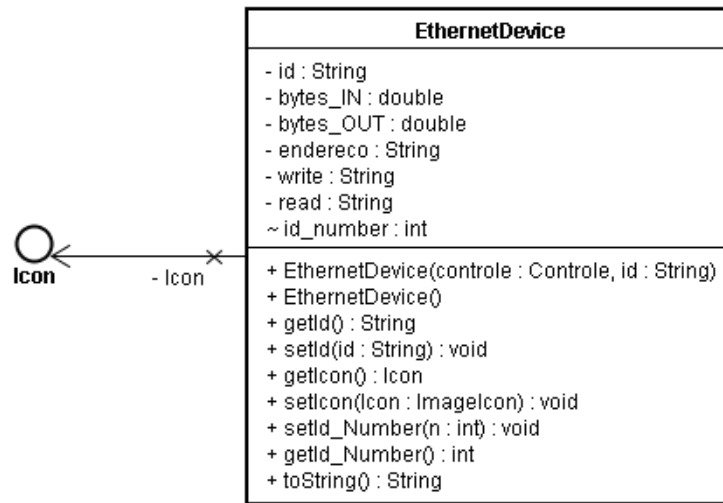
1 – UML, para representação em diagramas de classes, do “Frame Principal”.



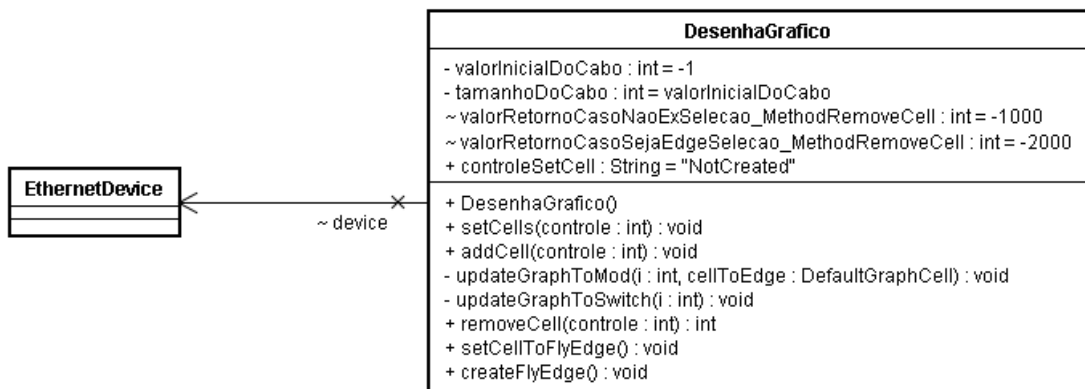
2 – UML, para representação em diagramas de classes, das funções “Controle” e “Configurações”.



3 – UML, para representação em diagramas de classes, da função “Ethernet Device”.



4 – UML, para representação em diagramas de classes, da função “Desenha Gráfico”.



Apêndice C – Código para controle distribuído.

```
#define chip_base_address      0x0000
#define RX_memory_base_address 0x6000
#define gS0_RX_BASE          chip_base_address + RX_memory_base_address
#define gS0_RX_MASK          0x0800 - 1
#define gS1_RX_BASE          gS0_RX_BASE + (gS0_RX_MASK + 1)
#define gS1_RX_MASK          0x0800 - 1
#define gS2_RX_BASE          gS1_RX_BASE + (gS1_RX_MASK + 1)
#define gS2_RX_MASK          0x0800 - 1
#define gS3_RX_BASE          gS2_RX_BASE + (gS2_RX_MASK + 1)
#define gS3_RX_MASK          0x0800 - 1
#define TX_memory_base_address 0x4000
#define gS0_TX_BASE          chip_base_address + RX_memory_base_address
#define gS0_TX_MASK          0x0800 - 1
#define gS1_TX_BASE          gS0_TX_BASE + (gS0_TX_MASK + 1)
#define gS1_TX_MASK          0x0800 - 1
#define gS2_TX_BASE          gS1_TX_BASE + (gS1_TX_MASK + 1)
#define gS2_TX_MASK          0x0800 - 1
#define gS3_TX_BASE          gS2_TX_BASE + (gS2_TX_MASK + 1)
#define gS3_TX_MASK          0x0800 - 1
```

```

char gwayipaddr[4] = {192,168,0,1};
char svrmacaddr[6];

#define make8(var,offset) ((unsigned int)var >> (offset * 8)) & 0x00FF
#define TO_WIZ          TRISF = 0x00
#define FROM_WIZ        TRISF = 0xFF

*****
void wr_wiz_addr(unsigned int addr)
{
    addr_hi = (make8(addr,1));
    addr_lo = addr & 0x00FF;
}
void wr_wiz_reg(char reg_data,unsigned int reg_addr)
{
    TO_WIZ;
    wr_wiz_addr(reg_addr);
    data_out = reg_data;
    clr_WR;
    NOP();
    set_WR;
    FROM_WIZ;
}
char rd_wiz_reg(unsigned int reg_addr)
{
    char data;
    wr_wiz_addr(reg_addr);
    clr_RD;
    NOP();
    data = data_in;
    set_RD;
    return(data);
}
*****
clr_RSET;
msecs_timer2 = 0;
while(msecs_timer2 < 2);
set_RSET;
addri = GAR0;
for(i8=0;i8<4;++i8)
    wr_wz_reg(gwayipaddr[i8],addri++);
addri = GAR0;
for(i8=0;i8<4;++i8)
    svrmacaddr[i8] = rd_wz_reg(addri++);

```

```

//SOCKET INTI*****
do{
wr_wiz_reg(Sn_MR_UDP,Sn_MR(0)); //protocol = UDP
wr_wiz_reg(0x00,Sn_PORT0(0)); //well-known ECHO port
wr_wiz_reg(0x07,Sn_PORT1(0));
wr_wiz_reg(Sn_CR_OPEN,Sn_CR(0)); //give the open command
if(rd_wiz_reg(Sn_SR(0)) != SOCK_UDP) //wait for the socket to come online
wr_wiz_reg(Sn_CR_CLOSE,Sn_CR(0));
}while(rd_wiz_reg(Sn_SR(0)) != SOCK_UDP);

//RECEIVE*****
do{
//look for incoming UDP datagrams
i16 = rd_wiz_reg(Sn_IR(0));
}while(i16 == 0);
wr_wiz_reg(0x04,Sn_IR(0));
//get the datagram size
hi_byte = rd_wiz_reg(Sn_RX_RSRO(0));
lo_byte = rd_wiz_reg(Sn_RX_RSRL(0));
get_size = make16(hi_byte,lo_byte);
//get the datagram's buffer offset
hi_byte = rd_wiz_reg(Sn_RX_RDO(0));
lo_byte = rd_wiz_reg(Sn_RX_RDL(0));
get_offset = make16(hi_byte,lo_byte) & gSO_RX_MASK;
//calculate the datagram's starting buffer address
get_start_address = gSO_RX_BASE + get_offset;
//UDP header size
header_size = 8;
//store the UDP header information
addri = get_start_address;
for(iB=0;iB<header_size;++iB)
{
packet[ip_destaddr+iB] = rd_wiz_reg(addri++);
++get_offset;
}
//store the UDP data
get_start_address = gSO_RX_BASE + get_offset;
udp_data_size = get_size - header_size;
addri = get_start_address;
for(iB=0;iB<udp_data_size;++iB)
{
packet[UDP_data+iB] = rd_wiz_reg(addri++);
++get_offset;
}
//update the receive buffer pointers
wr_wiz_reg(Sn_CR_RECV,Sn_CR(0));

//TRANSMIT*****
//load destination IP address
addri = Sn_DIPRO(0);
for(iB=0;iB<4;++iB)
wr_wiz_reg(packet[ip_destaddr+iB],addri++);
//load destination port address
addri = Sn_DPORT0(0);
for(iB=0;iB<2;++iB)
wr_wiz_reg(packet[UDP_srcport+iB],addri++);
//get transmit buffer offset
hi_byte = rd_wiz_reg(Sn_TX_WRO(0));
lo_byte = rd_wiz_reg(Sn_TX_WRL(0));
get_offset = make16(hi_byte,lo_byte) & gSO_TX_MASK;
//calculate transmit data buffer start address
get_start_address = gSO_TX_BASE + get_offset;
//load data into transmit buffer
addri = get_start_address;
for(iB=0;iB<udp_data_size;++iB)
{
wr_wiz_reg(packet[UDP_data+iB],addri++);
++get_offset;
}
//update transmit buffer pointer
wr_wiz_reg((make8(get_offset,1)),Sn_TX_WRO(0));
wr_wiz_reg((make8(get_offset,0)),Sn_TX_WRL(0));
//send data
wr_wiz_reg(Sn_CR_SEND,Sn_CR(0));
while(rd_wiz_reg(Sn_CR(0)));

```

Diagrama em blocos do circuito integrado proposto para validar o trabalho e sua divisão interna. O código descrito acima está gravado nesse circuito integrado.

