

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

**AVALIAÇÃO DE ARQUITETURAS DE COMUNICAÇÃO
DISTRIBUÍDAS PARA INTERLIGAÇÃO DE
SENSORES/ATUADORES EM UM HELICÓPTERO NÃO
TRIPULADO**

Dissertação Submetida à Universidade Federal de Santa Catarina para a
Obtenção do Grau de Mestre em Engenharia Elétrica

Lígia Cristina Fascioni Corrêa



0.258.206-7

UFSC-BU

Florianópolis, Setembro de 1996.

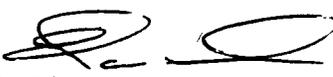
**AVALIAÇÃO DE ARQUITETURAS DE COMUNICAÇÃO DISTRIBUÍDAS PARA
INTERLIGAÇÃO DE SENSORES/ATUADORES EM UM HELICÓPTERO NÃO TRIPULADO**

Lígia Cristina Fascioni Corrêa

Esta dissertação foi julgada para obtenção do título de
Mestre em Engenharia Elétrica
especialidade **Engenharia Elétrica**
área de concentração **Controle, Automação e Informática Industrial**
e aprovada em sua forma final pelo Curso de Pós-Graduação.


Prof. Marcelo Ricardo Stemmer, Dr. Ing.
Orientador


Prof. Vitorio Bruno Mazzola, Dr.
Co-Orientador


Prof. Enio Valmor Kassick, Dr.
Coordenador do Curso de Pós-Graduação Engenharia Elétrica

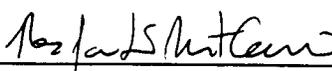
Banca Examinadora:



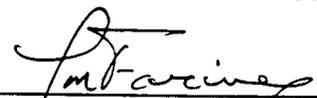
Prof. Marcelo Ricardo Stemmer, Dr. Ing., Presidente



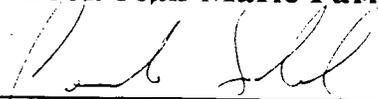
Prof. Vitorio Bruno Mazzola, Dr.



Profª Rosa M. Leão Rust Carmo, Dra.



Prof. Jean-Marie Farines, Dr. Ing.



Engº Conrado Werner Seibel, MSc.

Ao meu pai

por ter desde cedo despertado em mim
a curiosidade pelo
"Maravilhoso Mundo das Máquinas"

Agradecimentos

Agradeço à minha mãe pela força e apoio com os quais eu sempre pude contar.

Agradeço ao meu marido pelo amor, incentivo e paciência.

Agradeço aos meus tios Nádia e Melo, além da minha avó Bercides (*in memoriam*), pois sem a ajuda deles eu não teria chegado até aqui.

Agradeço à GYRON Tecnologia em Servo-Sistemas Ltda. pelo apoio.
Agradecimento especial ao Engenheiro Conrado Seibel, pela amizade, ajuda, e pela idéia do tema da dissertação.

Agradeço aos colegas do LCMI, destacando o Carlos Martins no auxílio ao aprendizado do ARENA e o Alessandro Garcia no esclarecimento do protocolo CSMA/DCR.

Agradeço aos meus orientadores Marcelo e Vitório por terem aceito o desafio de orientar uma dissertação com um tema não proposto por eles.

Agradeço a todos os membros da banca examinadora por terem aceito participar da avaliação deste trabalho com críticas e sugestões.

Resumo

A presente dissertação apresenta uma proposta de arquitetura de comunicação distribuída para a interligação de sensores/atuadores em um helicóptero não-tripulado baseada em protocolos conhecidos da indústria automobilística e aeronáutica.

Com este objetivo foi realizado um levantamento detalhado do fluxo de dados intra-veículo e um levantamento bibliográfico dos protocolos conhecidos na indústria para aplicações em eletrônica embarcada ou não.

Com base nos requisitos da aplicação em questão foi selecionado um conjunto de protocolos para a realização de uma análise comparativa de desempenho dos mesmos baseada em simulação.

Apoiando-se nos resultados da avaliação de desempenho e em um levantamento da disponibilidade de componentes no mercado, o trabalho é concluído com a recomendação de alguns destes protocolos para futura implementação.

Abstract

This dissertation presents a proposal of a distributed communication architecture for the interconnection among sensors/actuators in an unmanned helicopter based in known protocols used in the automotive and aeronautic industries.

For this goal, a detailed study of the intra-vehicle data-flow and a bibliographic research of known protocols for applications involving embedded electronics has been accomplished.

Based in this application's requirements, a set of protocols has been selected for a simulation based in comparative performance evaluation.

Based on the results of the performance evaluations and also considering the disponibility of components, the work is concluded with a recommendation of some protocols for future implementation.

Sumário

1	Introdução	1
2	Descrição da Aplicação: O HELIX	3
2.1	Introdução	3
2.2	O Projeto HELIX	3
2.3	Contexto da Comunicação	4
2.4	Classificação dos Dados Intra-Veículo	4
2.4.1	Dados de Controle do Helicóptero	5
2.4.2	Dados de Monitoração	7
2.4.3	Dados Relacionados à Carga Útil (Payload)	7
2.5	Apresentação e Justificativas de Frequência e Resolução dos Dados	8
2.5.1	Atuadores	8
2.5.2	Câmera de Vídeo	8
2.5.3	Sensor de Rotação	8
2.5.4	Sensores de Monitoração	9
2.5.5	Sensores Inerciais e de Atitude	9
2.5.6	Sensores de Posição / Velocidade	10
2.6	Configuração Atual	11
2.7	Arquitetura Proposta	11
2.8	Conclusão	13
3	Redes Intra-Veículos	14
3.1	Introdução	14
3.2	As Redes Intra-Veículos no Contexto Geral das Redes Informáticas	14
3.2.1	Redes Controle-Comando	14
3.2.2	As Redes Intra-Veículos e o Estado da Arte	15
3.3	Interoperabilidade e Conectividade	16
3.4	A Problemática da Comunicação em Tempo Real	16

3.5	Perfil da Aplicação	18
3.6	Escolha dos Protocolos para Simulação	18
3.7	Conclusão	19
4	Descrição dos Protocolos	21
4.1	Introdução	21
4.2	O Protocolo CAN - Controller Area Network	21
4.2.1	Histórico	21
4.2.2	Características Gerais	22
4.2.3	Topologia da Rede	23
4.2.4	O Modelo de Referência OSI e o CAN	23
4.2.5	Formato das Mensagens	25
4.2.6	Método de Acesso ao Meio	27
4.2.7	Detecção de Erros	28
4.2.8	Disponibilidade de Hardware	29
4.3	O Protocolo VAN - Vehicle Area Network	29
4.3.1	Histórico	29
4.3.2	Características Gerais	29
4.3.3	Topologia da Rede	30
4.3.4	O Modelo de Referência OSI e o VAN	30
4.3.5	Formato das Mensagens	31
4.3.6	Método de Acesso ao Meio	32
4.3.7	Detecção de Erros	32
4.3.8	Disponibilidade de Hardware	32
4.4	O protocolo J1850	33
4.4.1	Histórico	33
4.4.2	Características Gerais	33
4.4.3	Topologia da Rede	34
4.4.4	O Modelo de Referência OSI e o J1850	34
4.4.5	Formato das Mensagens	34
4.4.6	Método de Acesso ao Meio	35
4.4.7	Detecção de Erros	36
4.4.8	Disponibilidade de Hardware	36
4.5	O Protocolo MIL-STD-1553B	36
4.5.1	Histórico	36
4.5.2	Características Gerais	36

4.5.3	Topologia da Rede	37
4.5.4	O Modelo de Referência OSI e o MIL-STD-1553B	37
4.5.5	Formato das Mensagens	37
4.5.6	Combinações de Frames para Transferência de Dados	39
4.5.7	Método de Acesso ao Meio	39
4.5.8	Detecção de Erros	40
4.5.9	Disponibilidade de Hardware	41
4.6	O Protocolo C ² D - Chrysler Collision Detection	41
4.6.1	Histórico	41
4.6.2	Características Gerais	41
4.6.3	Topologia da Rede	41
4.6.4	O Modelo de Referência OSI e o C ² D	42
4.6.5	Formato das Mensagens	42
4.6.6	Método de Acesso ao Meio	43
4.6.7	Detecção de Erros	43
4.6.8	Disponibilidade de Hardware	43
4.7	O Protocolo CSMA/DCR	43
4.7.1	Histórico	43
4.7.2	Características Gerais	43
4.7.3	Topologia da Rede	44
4.7.4	O Modelo de Referência OSI e o CSMA/DCR	44
4.7.5	Formato das Mensagens	44
4.7.6	Método de Acesso ao Meio	44
4.7.7	Detecção de Erros	46
4.7.8	Disponibilidade de Hardware	48
4.8	O Protocolo FIP	48
4.8.1	Histórico	48
4.8.2	Características Gerais	48
4.8.3	Topologia da Rede	49
4.8.4	O Modelo de Referência OSI e o FIP	49
4.8.5	Formato das Mensagens	49
4.8.6	Método de Acesso ao Meio	50
4.8.7	Detecção de Erros	50
4.8.8	Disponibilidade de Hardware	51
4.9	Conclusão	52

5	Comparação de Desempenho dos Protocolos por Simulação	53
5.1	Introdução	53
5.2	Ferramenta	53
5.2.1	Conceitos	54
5.2.2	Módulos Lógicos	54
5.3	A Topologia da Rede Modelada para a Aplicação	55
5.4	Modelagem da Rede no ARENA	57
5.4.1	Modelagem da Criação de Dados no Sistema	57
5.4.2	Modelagem da Chegada dos Dados às Estações Destino	58
5.5	Modelo para o Protocolo CAN	60
5.5.1	Cálculo dos Tempos de Ocupação do Meio	60
5.5.2	Dados Necessários à Simulação	61
5.5.3	Resultados Obtidos na Simulação	62
5.6	Modelo para o Protocolo VAN	62
5.6.1	Cálculo dos Tempos de Ocupação do Meio	63
5.6.2	Dados Necessários à Simulação	63
5.6.3	Resultados Obtidos na Simulação	64
5.7	Modelo para o Protocolo J1850	65
5.7.1	Cálculo dos Tempos de Ocupação do Meio	65
5.7.2	Dados Necessários à Simulação	66
5.7.3	Resultados Obtidos na Simulação	66
5.8	Modelo para o Protocolo MIL-STD-1553B	66
5.8.1	Cálculo dos Tempos de Ocupação do Meio	67
5.8.2	Dados Necessários à Simulação	68
5.8.3	Resultados Obtidos na Simulação	70
5.9	Modelo para o Protocolo C ² D	71
5.9.1	Cálculo dos Tempos de Ocupação do Meio	71
5.9.2	Dados Necessários à Simulação	71
5.9.3	Resultados Obtidos na Simulação	72
5.10	Modelo para o Protocolo CSMA/DCR	72
5.10.1	Cálculo dos Tempos de Ocupação do Meio	73
5.10.2	Dados Necessários à Simulação	74
5.10.3	Resultados Obtidos na Simulação	75
5.11	Modelo para o Protocolo FIP	75
5.11.1	Cálculo dos Tempos de Ocupação do Meio	75
5.11.2	Dados Necessários à Simulação	76

5.11.3 Resultados Obtidos na Simulação	77
5.12 Análise dos Resultados	77
5.13 Conclusão	78
6 Conclusões	82
6.1 Descrição da Metodologia Utilizada	82
6.2 Resultados Obtidos	83
6.3 Contribuições Resultantes do Trabalho	83
6.4 Sugestões para Futuros Trabalhos	84

Capítulo 1

Introdução

Em sistemas informáticos, de um modo geral, tem sido bastante difundida a idéia de descentralização do processamento utilizando-se o conceito de redes. Essa motivação deve-se principalmente ao recente aperfeiçoamento de microcontroladores e microprocessadores, que, integrados a sensores e atuadores, constituem sistemas informáticos complexos denominados “inteligentes”.

A indústria automobilística tem estudado o assunto pelo menos desde o final da década de 80, quando a miniaturização e popularização dos componentes eletrônicos permitiu uma crescente integração de sistemas microprocessados em veículos, dando início à chamada era da eletrônica embarcada no setor automotivo. Nesta época, a empresa alemã BOSCH desenvolveu o primeiro protocolo para redes de sensores/atuadores embarcados, tendo sido denominado CAN (*Controller Area Network*) [Aze91].

Além do aumento da confiabilidade do sistema distribuído entre as diversas “ilhas” de processamento embarcado, obteve-se também um ganho considerável na diminuição da fiação e cablagem.

Outras empresas e associações de fabricantes desenvolveram seus sistemas proprietários, tornando o mercado mais diversificado nesta área, inclusive com alguns protocolos normalizados [SAE94].

A indústria aeronáutica também percebeu o problema da centralização do processamento quando este foi se tornando mais complexo [Nor95a]. Na área de aviônica, as arquiteturas distribuídas foram inicialmente consideradas para uso militar, chegando a gerar uma norma [Dep80]. A aeronáutica civil também tem demonstrado preocupação no desenvolvimento de sistemas aviônicos de arquitetura aberta [Nor95b] [For94].

O objeto de estudo desta dissertação é uma aeronave de asas rotativas não tripulada (helicóptero) que possui características de eletrônica embarcada muito semelhantes às que afetam os projetos das indústrias automobilística e aeronáutica.

A aeronave foi construída para fins de inspeção e vigilância e enquadra-se na designação formal de um UAV (*Unmanned Aerial Vehicle*) [Fin93], segmento da indústria de alta tecnologia que vem alcançando um crescimento expressivo nos últimos anos.

Anteriormente dominada pela indústria bélica militar, esta tecnologia tem se mostrado potencialmente útil em inúmeras aplicações civis. Como exemplos, podemos citar o monitoramento de áreas de preservação ambiental, filmagens aéreas, supervisão de tráfego, controle de ocupação, acompanhamento de difusão de pragas agrícolas, inspeção de isoladores em linhas de alta tensão, entre outros.

As vantagens da utilização deste tipo de veículo, principalmente em aplicações onde existem riscos à vida humana são evidentes, principalmente se seu custo operacional for comparado com o de uma aeronave tripulada convencional.

Assim como nas indústrias automobilística e aeronáutica, os fabricantes de UAVs não possuem um protocolo de comunicação entre unidades inteligentes formalmente aceito e utilizado por todos os seg-

mentos. Sendo uma tecnologia relativamente recente, o conceito de arquitetura aberta ainda vem sendo estudado em seus projetos tanto para UAVs como UGVs¹ e AUVs² [PM93], inclusive com a preocupação de obedecer aos critérios de conectividade e interoperabilidade [oD93] [BM93].

Estudos realizados demonstram que nesta área ainda coexistem sistemas de processamento centralizado [Bab93] [GM93] [RS93] [Can93] e distribuído [NC93] [BD93] [GW93] [E.93], sendo que estes últimos utilizam em grande parte sistemas proprietários.

Este trabalho tem como objetivo propor uma arquitetura distribuída para a aplicação em questão baseada em protocolos conhecidos da indústria automobilística e aeronáutica. Esta proposta fundamenta-se em simulações realizadas com os protocolos e análise do atendimento de requisitos.

A análise possui como pontos preponderantes, além do desempenho do protocolo avaliado do ponto de vista da aplicação, critérios diversos como disponibilidade de *hardware*, facilidade de implementação, universo de usuários, tecnologias utilizadas, entre outros, detalhados em capítulos específicos.

Esta dissertação descreve a evolução e metodologia do trabalho bem como os resultados e conclusões advindos da análise e está organizada da seguinte forma:

O capítulo 2 apresenta a aplicação e faz um estudo do seu fluxo de dados intra-veículo, cujos valores servirão como base para a simulação.

O capítulo 3 fornece conceitos gerais empregados no estudo de redes intra-veículos e justifica a escolha dos protocolos a serem considerados para fins de análise e simulação.

O capítulo 4 relaciona características dos protocolos escolhidos: CAN, VAN, J1850, MIL-STD-1553B, C²D, CSMA/DCR e FIP.

O capítulo 5 descreve a ferramenta utilizada para a simulação dos protocolos e a metodologia de modelização utilizada. Também apresenta os resultados obtidos para cada protocolo.

O capítulo 6 realiza uma análise comparativa entre os protocolos escolhidos.

O capítulo 7 apresenta as conclusões, recomendações para implementação e sugestões de futuros trabalhos na área.

¹Unmanned Ground Vehicles

²Autonomous Underwater Vehicles

Capítulo 2

Descrição da Aplicação: O HELIX

2.1 Introdução

Este capítulo tem por objetivo principal apresentar a aplicação objeto deste estudo e descrever o fluxo de dados intra-veículo para o modelo proposto.

Inicialmente é feita uma apresentação do projeto que deu origem à aeronave, seus objetivos e as características do veículo propriamente dito.

Para o levantamento do fluxo de dados é feita uma avaliação das grandezas necessárias ao controle de um helicóptero, relacionando-as com os respectivos sensores e atuadores, sendo que os dados envolvidos no modelo são classificados de acordo com sua funcionalidade. Para todos eles, são justificadas as taxas de transmissão e resolução em número de bits, bem como a periodicidade de cada grupo. Finalmente será feito um resumo do volume de transferência de dados intra-veículo.

2.2 O Projeto HELIX

A GYRON Tecnologia em Servo-Sistemas é uma indústria de base tecnológica que foi constituída em 1990 com o objetivo de desenvolver e comercializar sistemas baseados em dispositivos inteligentes controlados remotamente tipo robô [GYR92] [GYR96].

Com variações destes dispositivos para operações terrestres, aéreas e submarinas, é possível cobrir uma vasta gama de aplicações como aerofotografia, aquifotografia, aerofilagem, sensoriamento remoto, etc ...

Em 1991, após a realização de estudos aprofundados e estimativas de mercado sobre RPVs (*Remotely Piloted Vehicles*), a GYRON iniciou o desenvolvimento de um robô aéreo.

O projeto, denominado HELIX, encontra-se em fase final de desenvolvimento e constitui-se de um helicóptero não tripulado de dimensões reduzidas ¹, uma estação de controle em terra e um veículo de apoio com capacidade para transportar até duas aeronaves até o local de lançamento e oferecer completo apoio logístico às mesmas. O helicóptero foi concebido para conduzir cargas úteis de até 5 kg, sendo que o modelo básico prevê o transporte de uma câmera de vídeo para a transmissão de imagens em tempo real.

O UAV pode realizar seqüências de posicionamentos ou trajetórias completas (vôo autônomo), além de também obedecer a comandos enviados em tempo real pelo operador.

¹ Peso próprio de 20 kg e diâmetro da asa igual a 2 m.

A estação de controle em terra realiza a interface homem-máquina, permitindo o voo assistido pelo operador e disponibilizando informações referentes ao estado geral da aeronave, autonomia de voo, coordenadas geo-referenciadas da posição atual, ângulo de azimute, velocidades de translação em cada eixo, além dos dados relacionados à aplicação propriamente dita, como imagens da área sobrevoada e, eventualmente, grandezas medidas por sensores adicionais.

A eletrônica embarcada tem como principais funções adquirir todas as informações oferecidas à estação de controle em terra, transmiti-las, receber comandos do operador e executá-los (obedecendo a intertravamentos de segurança), além da tarefa que mais a diferencia de outros sistemas com finalidade similar: utilizar os dados provenientes dos sensores embarcados para manter a sua estabilidade, já que o helicóptero é uma aeronave de pilotagem particularmente complexa sem este auxílio.

2.3 Contexto da Comunicação

Os valores de referência para o controle da aeronave e da câmera embarcada são obtidos da Estação de Controle em Terra que também recebe dados do veículo. Esta comunicação é feita por meio de um enlace bidirecional de rádio-freqüência, conforme a figura 2.1.

Apesar de aparentemente não terem relação direta com a comunicação intra-veículo, estes valores são importantes porque circulam no interior do helicóptero. Os dados de referência recebidos pelo *UpLink* têm que chegar até o controle do veículo e o pacote enviado pelo *DownLink* é constituído de valores lidos pelos sensores embarcados e dados calculados pelo algoritmo de controle.

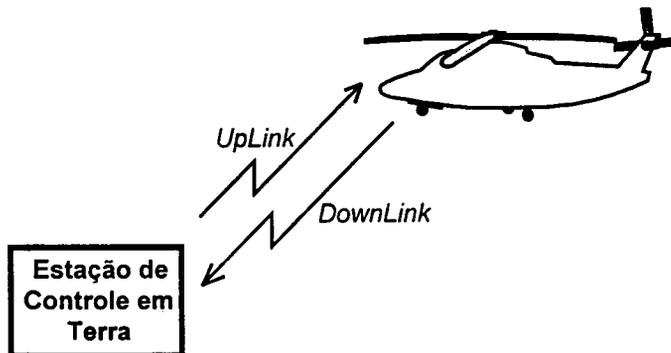


Figura 2.1: Comunicação entre o Helicóptero e a Estação de Controle em Terra

2.4 Classificação dos Dados Intra-Veículo

No interior do veículo, circulam dados de três tipos diferentes, de acordo com sua funcionalidade:

1. Dados necessários ao controle da aeronave
2. Dados contendo variáveis de estado utilizadas para fins de monitoração (*health monitoring*) [LD92].
3. Dados relacionados à aplicação propriamente dita, referentes à câmera embarcada e eventuais sensores adicionais.

Alguns destes dados possuem múltipla funcionalidade, pois servem tanto para o algoritmo de controle como para fins de telemetria.

2.4.1 Dados de Controle do Helicóptero

Um dos maiores desafios na pilotagem de um helicóptero, especialmente se esta pilotagem for remota, é o controle da estabilidade da aeronave [Sug93] [Sch87]. Isto ocorre devido ao forte acoplamento entre seus eixos à grande flexibilidade de movimentos. A figura 2.2 ilustra os seis graus de liberdade de um helicóptero [Pro93].

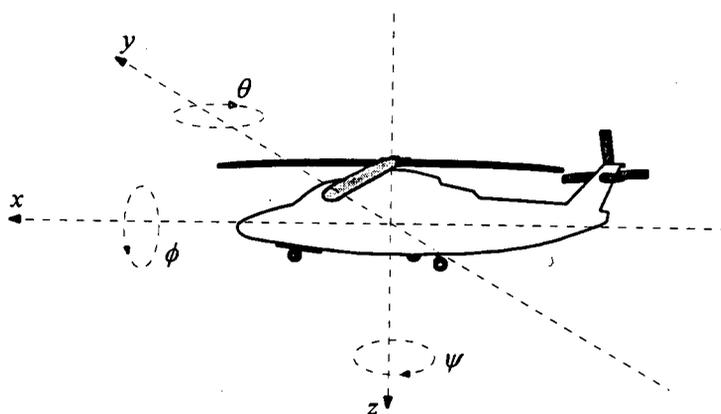


Figura 2.2: Movimentos permitidos em um Helicóptero

Os eixos e os ângulos possuem o seguinte significado:

X é o eixo longitudinal

Y é o eixo lateral

Z é o eixo vertical

ϕ é o ângulo de *roll* (rolagem)

θ é o ângulo de *pitch* (arfagem)

ψ é o ângulo de *yaw* (guinada)

Ao piloto da aeronave, interessa apenas fornecer referências de X, Y, Z e ψ e/ou suas derivadas primeiras (velocidades). Os ângulos ϕ e θ variam como consequência. Assim, o sistema de controle e estabilização deve considerar todas as grandezas acima.

Segundo [Sug93], um helicóptero remotamente pilotado necessita da seguinte matriz de variáveis de estado para realizar seu controle de estabilidade e posicionamento:

$$\begin{bmatrix} x & \dot{x} & \ddot{x} & \phi & \dot{\phi} \\ y & \dot{y} & \ddot{y} & \theta & \dot{\theta} \\ z & \dot{z} & \ddot{z} & \psi & \dot{\psi} \end{bmatrix} \quad (2.1)$$

Os termos constantes da matriz 2.1 possuem o seguinte significado:

$\dot{x}, \dot{y}, \dot{z}$ são as velocidades lineares de deslocamento nos eixos X,Y,Z

$\ddot{x}, \ddot{y}, \ddot{z}$ são as acelerações lineares ao longo dos eixos X,Y,Z

$\dot{\phi}, \dot{\theta}, \dot{\psi}$ são as velocidades angulares em torno dos eixos X,Y,Z.

A estas variáveis pode-se ainda acrescentar η , correspondente à velocidade de rotação do rotor principal, que possui papel preponderante no controle da aeronave com relação aos movimentos de subida e descida (comando coletivo).

De acordo com a figura 2.3, os dados necessários ao controle, pilotagem e guiagem da aeronave são representados pelos fluxos $f1$ e o somatório dos fluxos $f5, f6, f7, f8$ e $f9$, descritos a seguir².

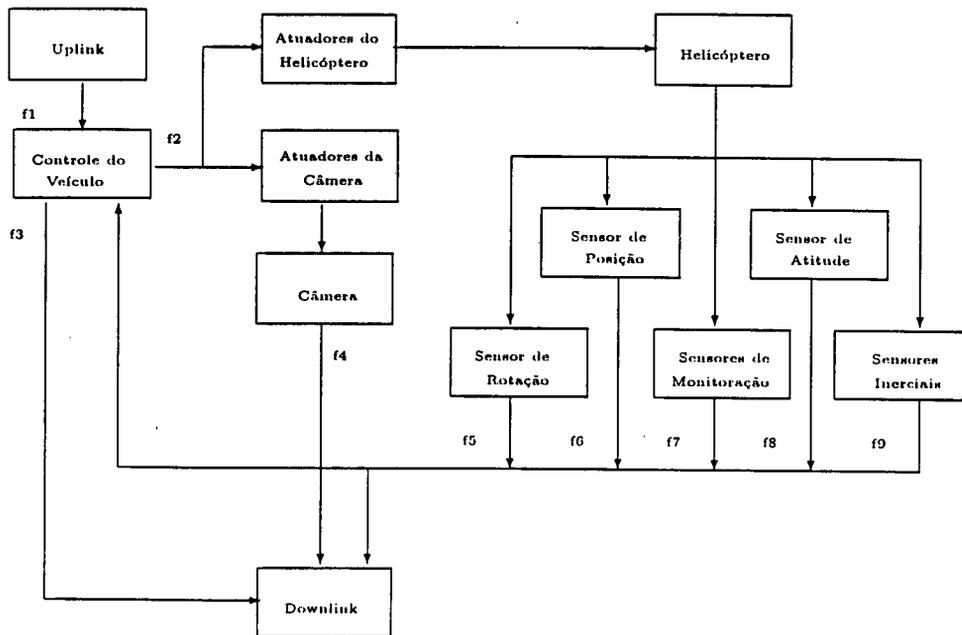


Figura 2.3: Estrutura do Fluxo de Dados Intra-Veículo

f1 - Valores de Referência Como citado anteriormente, ao piloto cabe referenciar os valores de atitude, posição e/ou velocidade. Assim, $f1$ é composto por X_{ref} , Y_{ref} , Z_{ref} e H_{ref} , que representam a posição de referência do centro de gravidade da aeronave e sua atitude (*heading*). Além destes, também são transmitidos códigos de operação (*opcodes*) e valores de ajuste (*offset*).

Outros valores de referência são transmitidos via *Uplink* para a inicialização dos sensores, mas não serão considerados neste estudo por não serem periódicos (são transmitidos apenas uma vez antes da decolagem).

²Os fluxos de dados são representados por f_n . A letra f é utilizada para não confundir o símbolo de fluxo com o dado que significa ângulo de *roll*, representado por ϕ , como será visto mais adiante.

- f2 - Referências para os Servo-Atuadores** O algoritmo de controle fornece valores utilizados para gerar sinais de PWM³ que controlam os servomotores do *swashplate*, do rotor de cauda e da plataforma de apoio da câmera. Estes valores são representados pelas variáveis S0 a S7.
- f5 - Medidas do Sensor de Rotação** Este sensor fornece o valor de rotação do motor que é representada pela variável η .
- f6 - Medidas do Sensor de Posição** Atualmente, para aplicações em tempo real, como é o caso, utiliza-se o sistema GPS (*Global Positioning System*) que baseia suas medições em uma constelação de satélites artificiais. Este sensor fornece os valores de posição nos três eixos de coordenadas aqui representados por X, Y e Z, além das velocidades ao longo dos eixos \dot{x} , \dot{y} e \dot{z} .
- f7 - Medidas dos Sensores de Monitoração** Estes sensores medem principalmente os valores de tensão na bateria, temperatura no cabeçote do motor e volume de combustível, representados por VDC, temp e comb, respectivamente. São dados necessários para a geração de alarmes e tomada de decisões no caso de voo autônomo.
- f8 - Medidas dos Sensores de Atitude** Sensores do tipo bússola fornecem o ângulo de *yaw*, que vem a ser o azimute⁴ do veículo. Eventualmente, estes sensores também podem fornecer a velocidade angular em torno do eixo Z. Aqui, estes valores são representados pelas variáveis ψ e $\dot{\psi}$.
- f9 - Medidas dos Sensores Inerciais** Os sensores inerciais são os giroscópios, acelerômetros e inclinômetros, combinados ou não. Os dados provenientes destes sensores são as velocidades angulares (representadas pelas variáveis ϕ e θ) e as acelerações lineares (\ddot{x} , \ddot{y} e \ddot{z}).

Outros dados de controle necessários são obtidos através de cálculos e combinação de dados dos diferentes sensores embarcados.

2.4.2 Dados de Monitoração

Neste estudo, considera-se dados de monitoração toda a informação recebida pelo operador em terra, possibilitando que este possa ter a percepção da situação atual da missão como se estivesse a bordo.

Neste grupo, enquadram-se os dados de telemetria, fornecidos pelos sensores embarcados, representados pelos fluxos *f5* a *f9* (já descritos na seção 2.4.1) e dados adicionais calculados pelo algoritmo de controle.

Nesta última categoria, representados na figura 2.3 pelo fluxo *f3*, estão os valores necessários para a geração de sinais de referência para os servos, *status* de todos os sensores embarcados, tempo de missão, estimativa de recursos disponíveis (combustível e energia elétrica), satélites utilizados pelo receptor de GPS da aeronave, situação dos enlaces de rádio-frequência, entre outros.

2.4.3 Dados Relacionados à Carga Útil (Payload)

A carga útil embarcada normalmente é composta de uma câmera de vídeo instalada numa plataforma com dois graus de liberdade: controle de elevação e azimute.

O sinal da câmera, representado na figura 2.3 pelo fluxo *f4*, deverá ser enviado pelo enlace de rádio-frequência utilizado para enviar dados de monitoração.

As referências para o seu posicionamento são obtidas do *UpLink*, passam pelo algoritmo de controle e finalmente geram sinais de PWM para os servomotores. Na figura 2.3, estes sinais são representados pelo fluxo *f2*.

³Pulse Width Modulation

⁴Azimute é o ângulo entre o objeto observado e o norte geográfico da terra do ponto de vista do observador - nesta aplicação, este ângulo define para onde aponta a proa do helicóptero.

Dependendo da aplicação, outros sensores podem ser embarcados para informações adicionais (ex. imagens infra-vermelho, câmeras fotográficas, sonar, etc...). Estes dados não serão aqui considerados devido à sua diversidade.

2.5 Apresentação e Justificativas de Frequência e Resolução dos Dados

Os dados descritos como componentes dos fluxos podem ser obtidos tanto através do algoritmo de controle, da estação de controle em terra (valores de referência), ou de diferentes sensores, inclusive com princípios de funcionamento e características bastante diversas.

Assim, para cada tipo de dado citado na seção 2.4 obtido através de sensores será feita uma breve análise, apresentando suas possíveis fontes e justificando as frequências e resoluções utilizadas.

2.5.1 Atuadores

Nesta aplicação, são utilizados 8 (oito) servomotores: 4 (quatro) para o controle do *swashplate*, 1 (um) para o rotor de cauda, 1 (um) para o acelerador e 2 (dois) para a plataforma da carga útil.

Segundo os principais fabricantes de servomotores utilizados, um período de 25 ms é suficiente para a geração de PWM. Assim, esta taxa é utilizada para a atualização destes valores, com resolução de 16 bits. Este período também coincide com meia volta do rotor principal.

2.5.2 Câmera de Vídeo

No caso da câmera de vídeo, seu sinal deverá ser digitalizado antes de ser enviado à estação de controle em terra, devido principalmente ao fato dos sinais analógicos necessitarem de faixas de frequência muito largas.

Devido ao grande volume de informações, é necessário que se aplique alguma técnica de compactação de dados antes do envio, pois pretende-se enviar esta informação pelo enlace de rádio frequência utilizado para enviar sinais de monitoração.

Baseando-se em estudos realizados com o objetivo de se obter a ordem de grandeza do volume de dados que será transmitido, observou-se que as técnicas de compressão e transmissão utilizadas variam bastante [PG91] [Str93] [SWO93] [Wil93], mas na área de veículos remotamente pilotados tem sido utilizado como dado auxiliar a velocidade média de deslocamento do veículo. Assim, é possível calcular a variação da imagem, sendo necessária a transmissão de somente uma parte do quadro.

Segundo [NP93], experiências realizadas utilizando-se uma técnica de compactação com este princípio de funcionamento, considera-se que um veículo com velocidade média de 24 km/h pode obter informações de resolução bastante razoável com um fluxo de dados de 64 kbits/s.

Para fins de modelagem, serão utilizados estes dados.

2.5.3 Sensor de Rotação

A velocidade de rotação do motor é obtida através de um tacômetro, possui resolução de 16 bits e é atualizada a cada 100 ms.

2.5.4 Sensores de Monitoração

Estes sensores, como são utilizados apenas para fins de monitoração, são menos sofisticados e fornecem dados com uma taxa de atualização média de 1 Hz.

A tensão de alimentação da eletrônica embarcada no veículo é proveniente de um alternador. Em caso de falha deste, um banco de baterias provê tensão suficiente para um possível pouso. A monitoração deste valor é feita a cada segundo e possui resolução de 16 bits.

A temperatura do motor (*tm*) e o nível de combustível (*comb*) são monitorados a cada segundo através de sensores ligados a conversores A/D⁵ e possuem resolução de 16 bits.

2.5.5 Sensores Inerciais e de Atitude

A atitude de uma aeronave define sua orientação no espaço através da rotação em torno do seu centro de gravidade [Sel94]. Em outras palavras, os ângulos que o corpo do veículo forma em relação ao sistema de coordenadas cartesiano⁶ definem a sua atitude.

Estes ângulos (ϕ , θ e ψ) podem ser obtidos diretamente em unidades de deslocamento angular (graus, radianos ou grados) ou através da integração de valores de velocidade angular, provenientes dos sensores inerciais.

Para a obtenção do ângulo de *yaw* aqui representado por ψ , pode-se utilizar um sensor do tipo magnetômetro.

Os magnetômetros, ou as chamadas bússolas eletrônicas, utilizam como referência as linhas de campo magnético da terra, fornecendo o valor de ψ - o ângulo de azimute para navegação. O valor é atualizado em média a cada 50 ms (20 Hz) e possui uma resolução de 32 bits. Alguns tipos de bússolas também fornecem a velocidade angular $\dot{\psi}$, com a mesma resolução e frequência.

Para a obtenção dos outros ângulos pode-se dispor de três tipos de sensores:

Inclinômetros São sensores que utilizam o vetor gravidade como referência, fornecendo valores dos ângulos no plano, como ϕ (*roll*) e θ (*pitch*). A taxa de atualização mais comum entre os sensores disponíveis no mercado é de 50 ms (20 Hz) e resolução de 32 bits.

Acelerômetros São sensores inerciais [RR80] [Far76] que fornecem dados referentes às acelerações lineares (\ddot{X} , \ddot{Y} e \ddot{Z}) ao longo dos eixos cartesianos. Estes valores são atualizados em média a cada 2 ms e possuem 16 bits de tamanho.

Giroscópios de Taxa (Rate Gyros) Estes sensores têm caráter antecipativo por fornecerem as velocidades angulares de *roll*, *pitch* (*vertical gyro*) e *yaw* (*horizontal gyro*) [Bla91][Wee48]. São montados geralmente em forma de tríade no corpo da aeronave. Por serem valores de natureza dinâmica, os sensores disponíveis no mercado fornecem uma taxa mínima de atualização de 50 ms (20 Hz), sendo bem adequada ao algoritmo de controle do modelo estudado, pois corresponde a aproximadamente a uma rotação do rotor principal e tem se mostrado satisfatória de acordo com experiências realizadas. A resolução média é de 32 bits.

Cabe ressaltar que estes sensores podem ser construídos de maneira a formar diversas combinações, dependendo da aplicação a que se destinam, gerando assim sistemas como o DGI (giro-inclinômetro digital), DGCompass (bússola eletrônica com giroscópio) entre outros, sempre respeitando valores mínimos de taxas de atualização e resolução a fim de garantir um bom desempenho do sistema de controle.

⁵ Analógico/Digital

⁶ Cujá origem está no seu centro de gravidade

2.5.6 Sensores de Posição / Velocidade

Aqui serão descritos os tipos de sensores que fornecem dados referentes à posição geo-referenciada do helicóptero e suas velocidades lineares ao longo dos três eixos cartesianos do sistema de referência de seu corpo.

Os sensores de posição/velocidade podem ser classificados em duas categorias principais, quanto à sua taxa de atualização e resolução: alta resolução e baixa dinâmica e baixa resolução e alta dinâmica.

Ambos os tipos são necessários a um sistema de navegação de qualidade. Assim, através de filtros complementares [SC95] e de *Kalman* [IDG95], obtém-se as taxas de resolução e dinâmica adequadas, realizando-se a composição das informações provenientes dos dois tipos (alta resolução/baixa dinâmica e baixa resolução/alta dinâmica). Também para a obtenção destas grandezas alguns fabricantes têm integrado estes sensores a fim de realizar a filtragem internamente, constituindo assim os chamados sistemas de navegação inercial.

Sensores de alta resolução e baixa dinâmica

São os sensores do tipo GPS (*Global Positioning System*), Decca e Loran. Eles possuem como característica principal a alta precisão em seus dados, tanto de posição como velocidades lineares. Cabe aqui algumas considerações quanto a estes diferentes sensores, apresentando brevemente seus princípios de funcionamento e características fundamentais.

GPS - *Global Positioning System* É um sistema de navegação composto por uma constelação de 24 satélites orbitando a uma altitude de 20.000 Km e período de 12 horas implementado pelo Departamento de Defesa dos Estados Unidos [Hur89] [Bit94].

Seu funcionamento baseia-se no fato de pelo menos quatro satélites da constelação serem visíveis para os receptores de GPS em qualquer ponto do planeta terra.

Estes satélites emitem ondas de rádio e possuem relógios atômicos que fornecem com precisão o momento em que a informação partiu do satélite, incluindo este dado no pacote enviado. Utilizando-se deste valor, o receptor na terra tem condições de calcular o tempo de propagação desta onda. Sendo conhecida a velocidade de propagação de uma onda eletromagnética no espaço e tendo-se o tempo decorrido, podem ser calculadas as distâncias de cada satélite até o ponto de recepção. Através da triangulação destes, obtém-se a localização de qualquer ponto no espaço em que esteja localizado um receptor.

Com a utilização do GPS, os valores obtidos geralmente possuem uma resolução bastante alta, se comparada com outros métodos. Por outro lado, o tempo médio de atualização destes dados é de 1 segundo (1Hz).

Os dados de posição, normalmente fornecidos em metros, são atualizados uma vez por segundo pelo GPS e possuem tamanho de 48 bits cada (resolução de 10^{-4} m).

Os valores de velocidade ao longo dos eixos também são fornecidos pelo GPS com tamanho de 32 bits (resolução de 10^{-4} m/s).

Decca O sistema de auxílio à navegação conhecido por *decca-navigator-system* [KF76] é baseado no princípio da navegação hiperbólica e é constituído por instalações emissoras em terra, por um receptor especial localizado a bordo do veículo (o *deccometer*) e ainda por cartas de navegação especiais. As estações emissoras da cadeia decca existentes em um grande número de países na costa europeia, transmitem, em determinadas frequências e em direções fixas, feixes de ondas que são captadas a bordo dos veículos e identificadas nas cartas náuticas especiais. O ponto de intersecção das hipérbolas correspondentes aos sinais identificados indica a posição do veículo.

Loran O sistema Loran (*long range navigation*) [KF76] possui um alcance muito superior às 300 milhas máximas do sistema decca, ainda que seu funcionamento se baseie num princípio semelhante. A

diferença é que no sistema Loran, procede-se à medição não das frequências das ondas, mas do intervalo de tempo entre elas.

Os sistemas Decca e Loran, apesar de fornecerem os dados necessários à navegação do veículo, possuem uma série de restrições. Por serem projetados com a finalidade de auxílio à navegação marítima, seus emissores encontram-se sempre próximos à costa dos países, restringindo as aplicações para uma aeronave. Por este motivo, optou-se pelo sistema GPS, atualmente utilizado inclusive em aviação comercial.

Sensores de baixa resolução e alta dinâmica

São sensores como os do tipo *tubo de Pitot* que medem a velocidade de deslocamento do ar ou os sensores inerciais do tipo *strap-down*. No caso dos sensores do tipo *strap-down*, sua resolução é por volta de 16 bits.

Como os dados são fornecidos através de cálculos realizados por softwares geralmente embarcados nos sensores, suas taxas de atualização chegam a ser altíssimas, da ordem de 500 Hz.

Estes sensores fornecem dados relativos às velocidades e acelerações lineares, que integrados no tempo, podem fornecer os valores de posição e velocidade de caráter antecipativo com uma dinâmica muito maior que os sistemas de alta resolução.

2.6 Configuração Atual

O Helix utiliza atualmente uma arquitetura centralizada num processador RISC⁷ de 32 bits tipo *transputer* com unidade de ponto flutuante. Este processador é responsável pela execução dos algoritmos de navegação, guiagem e controle, além do posicionamento da plataforma de carga útil e administração dos enlaces de comunicação.

Como o *transputer* possui apenas 4 *links* de comunicação com o mundo externo e estes possuem pouca flexibilidade de configuração, o sistema utiliza dois microcontroladores da família 8051 para realizar o interfaceamento com os sensores e atuadores, como pode ser visto na figura 2.4.

Apesar da configuração apresentada estar em funcionamento e o protótipo do Helix encontrar-se em fase de testes com resultados bastante satisfatórios, uma arquitetura alternativa é apresentada neste trabalho com o objetivo de otimizar os seguintes aspectos:

- Aumento na flexibilidade para alteração no conjunto de sensores
- Aumento na confiabilidade (eliminando a concentração de poder em um único elemento)
- Diminuição do número de conectores blindados de alta frequência (conseqüente diminuição de peso e custo)

2.7 Arquitetura Proposta

A aplicação em questão define um sistema tempo real típico, sendo então recomendada, para o caso de um sistema distribuído, uma topologia tipo barramento. O trabalho propõe a arquitetura apresentada na figura 2.5.

⁷Reduced Instructions Set Computing

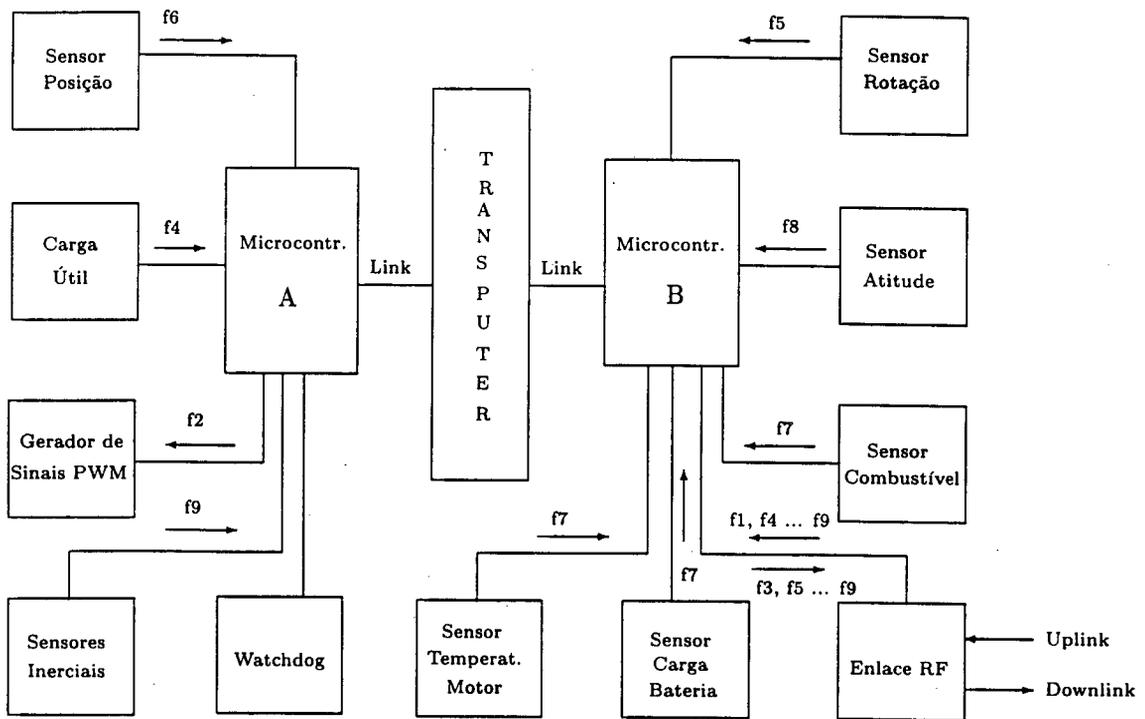


Figura 2.4: Esquema Simplificado da Atual Arquitetura de Comunicação Intra-Veículo do Helix

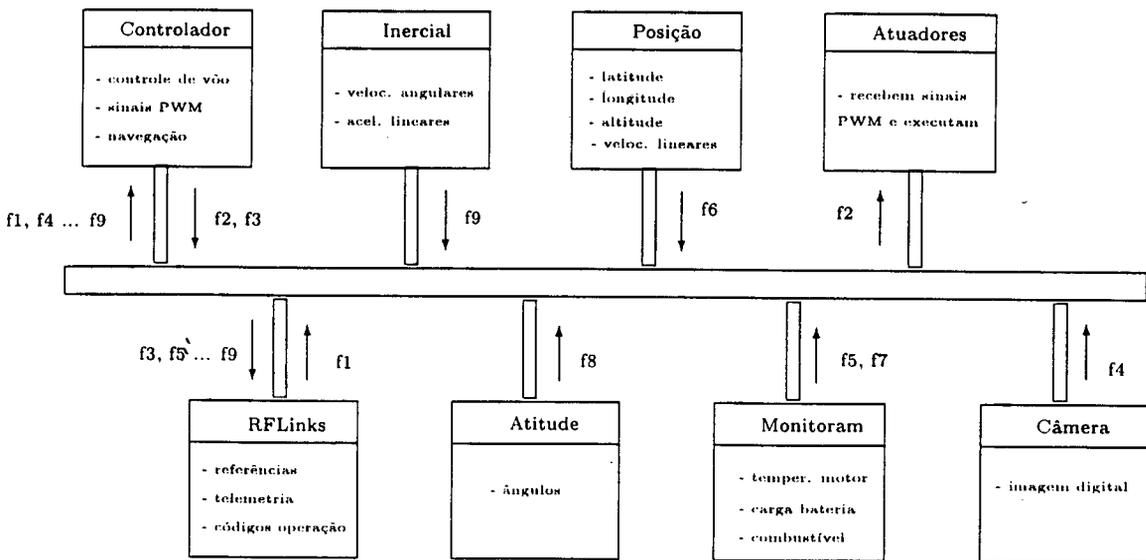


Figura 2.5: Proposta de Arquitetura Distribuída para o Helix

2.8 Conclusão

O presente capítulo descreveu o contexto da aplicação bem como as grandezas envolvidas no fluxo de dados intra-veículo que serão utilizados no modelo a ser simulado.

Foram fornecidas justificativas com respeito à origem, necessidade, periodicidade e tamanho dos dados considerados.

Assumiu-se, para efeito de simulação, que o nó responsável pela câmera deixa disponível para o sistema um volume de dados da ordem de 64 kbits a cada segundo. Sua distribuição em frames de dados será efetuada de acordo com as características de cada protocolo a ser utilizado. A necessidade da inclusão destes dados na rede intra-veículo se deve ao fato de que existe uma previsão futura para a sua utilização como auxílio à pilotagem em caso de vôo fora do alcance visual.

Os dados da câmera de vídeo poderiam ser enviados através de um canal independente de comunicação, porém, esta solução não foi adotada tendo em vista a pretendida aplicação futura destes dados no controle da aeronave, como já mencionado. A opção de se realizar o tratamento dos dados da câmera no próprio nó que envia os dados para a estação de terra (estação RFLinks) também foi em princípio descartada, pois a carga de processamento neste nó já é bastante pesada na configuração atual e a eventual necessidade do uso destes dados em outro nó faz com que esta alternativa de solução se mostre inadequada.

No momento, apesar dos dados da câmera não terem o *status* de *hard-deadline* como os outros utilizados em controle (a perda de um frame não é crítica) estes foram tratados como tal com a finalidade de se observar sua influência no comportamento dinâmico da aplicação.

A tabela 2.1 representa o resumo dos dados utilizados para a análise neste trabalho bem como seus tamanhos e períodos.

Fluxo	Dado Tipo	Tamanho [bits]	Período [ms]	Definição
f1	refs	4 x 48	50	referências de posição Xref, Yref, Zref e Href
	opcode	3 x 48	200	códigos de operação
	offset	4 x 48	1000	referências p/ ajuste
f2	servos	8 x 16	25	sinais PWM p/ servos S0 a S7
f3	comm	24 x 48	1000	telemetria e monitoração
f4	video	65536	1000	sinal digitalizado da imagem
f5	rpm	1 x 16	100	rotação no motor (η)
f6	xyz	3 x 48	1000	latitude, longitude e altitude (X, Y e Z)
	vel	3 x 16	1000	velocidades lineares X, Y e Z
f7	battemp	3 x 16	1000	VDC, temp e comb
f8	atit	2 x 32	100	ângulo ψ e velocidade angular $\dot{\psi}$
f9	w	2 x 32	50	velocidades angulares $\dot{\phi}$ e $\dot{\theta}$
	acel	3 x 16	50	acelerações angulares $\ddot{\phi}$ e $\ddot{\theta}$

Tabela 2.1: Resumo do Fluxo de Dados Intra-Veículo

Também neste capítulo foi apresentada a arquitetura atual que se encontra implementada na aplicação e a alternativa proposta neste trabalho.

Com a topologia definida, o trabalho agora é desenvolvido no sentido de estudar um protocolo adequado para este contexto.

Capítulo 3

Redes Intra-Veículos

3.1 Introdução

Este capítulo descreve a classificação dos níveis de comunicação utilizados em redes intra-veículos, os conceitos de interoperabilidade e conectividade utilizados em aviãoica e uma visão geral da problemática de sistemas de comunicação tempo real.

Na seção 3.5, é apresentado um perfil da aplicação estudada bem como a sua adequação aos conceitos estudados nas seções anteriores. Com este traçado, é possível definir as características gerais para a rede adequada e apresentar os protocolos conhecidos que se encaixam no modelo. Por último, é definida a lista dos protocolos escolhidos para simulação e os critérios utilizados na seleção.

3.2 As Redes Intra-Veículos no Contexto Geral das Redes Informáticas

3.2.1 Redes Controle-Comando

Numa aplicação industrial de controle e comando de processos, a necessidade de comunicação entre os diferentes módulos que suportam a aplicação global tem sido tema de vários grupos de interesse na área, possibilitando a definição de alguns conceitos gerais [SAE94].

Este tipo de rede local, também denominada controle-comando [DS90] (do qual fazem parte as redes de automóveis) é, em geral, diferente das redes locais de uso geral, como a Ethernet, do ponto de vista da arquitetura e gestão.

Um sistema controle-comando pode ser subdividido em várias sub-redes interconectadas, cada uma realizando a comunicação no interior de uma classe de aplicações ou mesmo possibilitando a interconexão entre várias classes. Neste tipo de rede, a comunicação é classificada em quatro níveis, mostrados na figura 3.1.

Nível 1 Trata-se da comunicação eventual ou não entre diferentes sensores/atuadores para troca de informações de sincronismo ou de outro tipo. Como exemplo, pode-se citar uma ligação entre um controlador programável (ou um microprocessador) e os equipamentos por ele controlados. Os protocolos BitBus (Intel) e FIP realizam comunicações neste nível.

Níveis 2 e 3 O nível 2 define a ligação entre diferentes máquinas de controle, tendo, cada uma, sua aplicação bem definida. O nível 3 reagrupa os computadores de pilotagem, supervisão e manutenção, assim como os microcontroladores de comunicação. Na prática, este tipo de rede, também

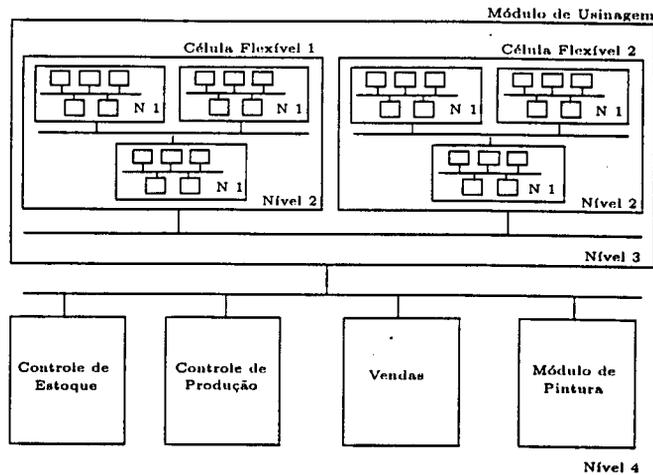


Figura 3.1: Níveis de Comunicação em Redes Locais Industriais

denominada rede local industrial, é utilizada para conectar diferentes tipos de equipamentos. Os protocolos Modbus (APRIL), LAC (COMPEX) e FACTOR (Aptor) servem como exemplos.

Nível 4 Este é o nível mais geral de gerenciamento. A rede reagrupa os computadores e os sistemas informáticos encarregados da gestão global e trata dos problemas de fabricação e produção. As restrições topológicas, econômicas e de local de trabalho fazem com que este nível se compare às redes informáticas as quais os protocolos de comunicação estão definidos pelo sistema ISO/OSI¹. Exemplos: Ethernet, Token Ring (IBM) e MAP.

3.2.2 As Redes Intra-Veículos e o Estado da Arte

Para resolver os problemas de comunicação de dados entre diferentes equipamentos eletrônicos embarcados a bordo de um veículo, quatro classes de redes são definidas, similares aos quatro níveis apresentados na seção 3.2.1, mas a SAE² reconhece apenas os três primeiros [SAE94].

Classe A É a multiplexagem dos cabos que envolvem a carroceria de um veículo, sendo que a rede é aqui utilizada apenas para minimizar o volume dos cabos elétricos dos equipamentos. Os nós desta classe podem ser considerados “sem inteligência”, pois não possuem nenhum tipo de processamento. Exemplos: comando dos vidros, espelho retrovisor, etc ...

Classe B É um sistema de multiplexagem utilizado principalmente para eliminar a redundância de sensores quando mais de um equipamento compartilha o mesmo dado, funcionando assim como um barramento para a troca de parâmetros. Exemplos de dados compartilhados por mais de um sistema: velocidade, temperatura, presença de obstáculos, etc ...

Classe C É um sistema para multiplexar informações em altas taxas de transferência, associadas principalmente a sistemas de controle em tempo real vindo a auxiliar o processamento distribuído.

Classe D É uma extensão da classe precedente, pois reúne as redes de interconexão de sistemas de auxílio à navegação, anti-colisão, ou ainda computadores de diagnóstico de pane.

A combinação destas classes consegue incluir toda a informação que flui no interior de um veículo automotivo. Cada classe suporta a imediatamente inferior, de forma que, a classe D possui abrangência

¹International Standards Organization/Open Systems Interconnection

²Society of Automotive Engineers

sobre todas as demais (ver figura 3.2). Assim, a classe A suporta as aplicações de chaveamento de mais baixo nível. A classe B introduz o aspecto de parametrização de dados e a classe C, o suporte à aplicações de tempo real.

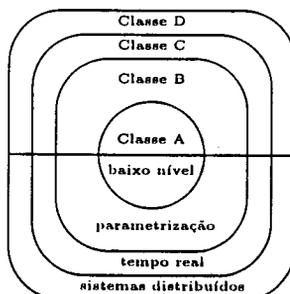


Figura 3.2: Classes de Comunicação Intra-Veículo

3.3 Interoperabilidade e Conectividade

Os grupos de trabalho em veículos não tripulados têm apresentado uma preocupação em normalizar ou ao menos definir alguns padrões para a integração de interfaces aviônicas com a finalidade de facilitar o projeto e manutenção destes sistemas [BM93].

Os critérios de interoperabilidade e conectividade³ estão sendo considerados prioritários no planejamento estratégico destes grupos, especialmente na indústria militar.

Com o objetivo de adequar a solução de integração aviônica da aplicação estudada aos padrões internacionais na área, estes critérios também são considerados na análise final, com as devidas ressalvas, já que seus conceitos são particularmente definidos para UAVs militares.

A seguir, as definições:

Interoperabilidade é definida como a habilidade que os sistemas e as unidades a serem integradas possuem de prover e utilizar serviços em cooperação com outros sistemas.

Conectividade é a habilidade de identificar e capitalizar oportunidades de reutilização de sistemas com eficiência através do uso de sistemas e sub-sistemas comuns a uma mesma família.

Estes conceitos poderiam se estender aos cuidados na especificação do protocolo, verificando o *hardware* disponível no mercado e principalmente se é fabricado e fornecido por mais de uma indústria de semicondutores e compatibilidade entre os diversos sistemas.

3.4 A Problemática da Comunicação em Tempo Real

A aplicação em questão, devido aos seus requisitos de tempo real, deve ser provida de um sistema computacional que forneça respostas a estímulos advindos do ambiente a ser controlado em intervalos de tempo impostos pelas características dinâmicas deste ambiente.

Se o seu sistema computacional for composto de várias unidades independentes de processamento, compondo um sistema distribuído, o sistema de comunicação que interliga estas unidades (ou estações

³No original *interoperability e commonality*

da rede) entre si deve apresentar um comportamento temporal previsível, de forma a garantir que o conjunto continue a possuir características de um sistema tempo real.

Para tal, as mensagens pendentes em cada estação devem poder ser entregues a seu destino antes de um prazo limite (*deadline*) associado a cada uma delas.

Como exemplo, pode-se imaginar a situação representada na figura 3.3.

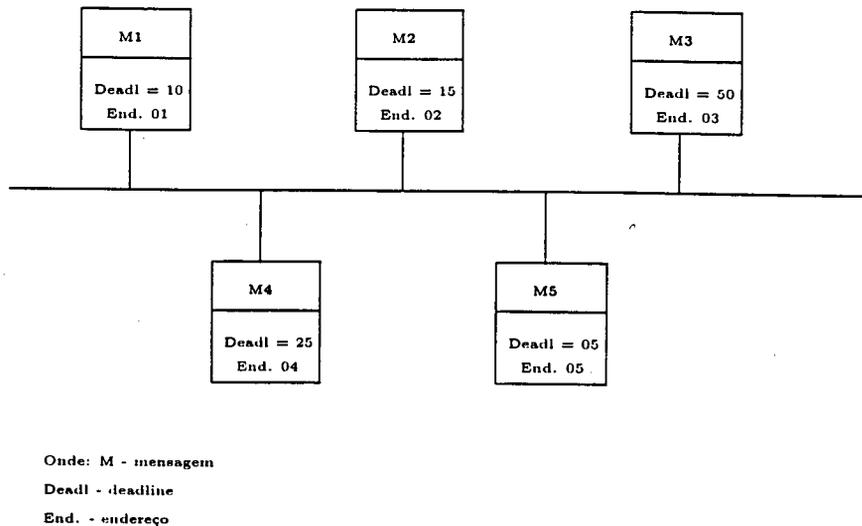


Figura 3.3: Exemplo de Rede Tempo Real

Um dos problemas da comunicação tempo real neste caso seria como definir a ordem de concessão do direito de acesso ao meio de forma a garantir que todas as mensagens sejam entregues antes do seu *deadline*.

Se a seqüência de concessão do acesso ao meio for controlada por uma estação mestra (sistema mestre/escravo), esta será fixa e definida pela configuração do sistema, não permitindo assim, garantir *deadlines* de mensagens aperiódicas como alarmes, cuja ocorrência não pode ser prevista no momento da configuração.

Outra opção seria a utilização de protocolos de acesso ao meio com comando distribuído, como o *token passing*. Nesta técnica, como proposta na norma IEEE 802.4, o direito de acesso ao meio é dado pela passagem de uma mensagem especial chamada *token* entre as estações, partindo daquela com endereço maior e seguindo a ordem decrescente de endereços. Esta técnica não atende ao problema apresentado na figura 3.3, pois as mensagens pendentes seriam enviadas na seguinte ordem: M5, M4, M3, M2 e M1, o que provavelmente implicaria na perda dos *deadlines* das duas últimas (M2 e M1).

Considerando o tipo de protocolo de comunicação necessário neste caso, pode-se definir dois requisitos principais [ZS95]:

1. O protocolo precisa ter um rápido acesso ao barramento para mensagens esporádicas de alta prioridade.
2. Considerando que todos os sistemas tempo real possuem mensagens periódicas, o protocolo deveria atendê-las com a maior eficiência possível, respeitando seus *deadlines*.

Ao se considerar protocolos como o *token bus*, verificamos que estes satisfazem o segundo requisito (atendimento aos *frames* periódicos), mas não o primeiro: é sabido que este protocolo possui um valor alto de tempo de latência para o pior caso (note-se que a prioridade é atribuída ao nó e não à mensagem) tornando-o inadequado para sistemas que possuam mensagens aperiódicas de baixo *deadline*.

Assim, os protocolos baseados em *token* lógicos como o *token-bus*, ou físicos, como o *token-ring*, possuem flexibilidade e robustez limitados. Em particular, quando a configuração do sistema precisa ser alterada de forma transparente ou em ambientes com alto índice de ruído, torna-se praticamente impossível prever sua performance [Lan87].

Com relação às considerações tecidas acima, observa-se que a limitação referente às mensagens esporádicas não é tão relevante nesta aplicação, que baseia-se somente em mensagens periódicas. Já os problemas de flexibilidade e robustez devem ser considerados com cuidado, já que um sistema estudado, por definição, deve absorver o conceito de arquitetura aberta⁴.

Talvez por este motivo, poucos protocolos utilizados em veículos com eletrônica embarcada utilizem o *token-passing* como técnica de acesso ao meio [SAE94].

Já os protocolos baseados no padrão CSMA/CD (*Carrier Sense Multiple Access/Collision Detect*) devem ser evitados a qualquer custo, por possuírem características essencialmente probabilísticas que impossibilitam o acesso determinístico necessário a sistemas tempo real.

3.5 Perfil da Aplicação

Com relação aos critérios acima expostos, pode-se considerar que a aplicação possui características essencialmente determinantes das classes B e C, segundo os padrões fornecidos pela seção 3.2.2.

Quanto aos critérios de interoperabilidade e conectividade, considera-se adequada a utilização de padrões já estabelecidos no mercado, com vários fornecedores e com características de arquitetura aberta permitindo a maior flexibilidade possível para a alteração de projeto. No caso de uma rede, isto significa que a adição ou subtração de um nó deve ser o menos traumático possível para o sistema. Ela também deve ser robusta no caso de falha em um dos nós, isto é, o sistema deve continuar funcionando com desempenho aceitável em caso de perda de uma das unidades.

Os métodos de acesso ao meio utilizados devem ser determinísticos e os *deadlines* devem ser obedecidos para todas as mensagens.

A rede deve ser de curta distância (dificilmente a distância entre os nós mais distantes deve ultrapassar 5 metros).

Com relação à camada física, não há preferência por um determinado tipo de condutor, mas acredita-se que o cabo coaxial e o par trançado possuam uma melhor relação custo-benefício neste caso.

A utilização de fibras óticas encareceria as conexões sem uma justificativa convincente em se considerando a ordem de grandeza das distâncias em questão.

A utilização de fio único ou duplo exigiria blindagem eletromagnética especial, pois é importante que este meio seja protegido contra ruídos eletromagnéticos provenientes do ambiente embarcado (que inclusive conta com um motor a explosão e um alternador bem próximos à unidade eletrônica).

3.6 Escolha dos Protocolos para Simulação

Pela analogia da aplicação em questão com as áreas de redes intra-veículos e aviônica, procurou-se escolher dentre as redes existentes, as que possuem características gerais que mais se adequam ao perfil da aplicação de acordo os critérios descritos anteriormente.

Na escolha, também foi levado em consideração o fato de que algumas empresas já estão utilizando seus protocolos em grande escala, fator preponderante na diminuição de custos e validação da proposta.

⁴Dependendo da utilização do Helix, tanto os sensores da carga útil como os constituintes do sistema de controle podem sofrer alteração para obtenção de maior ou menor precisão no posicionamento.

A SAE⁵ publicou em abril de 1993 um documento que compara de forma sumária os diversos protocolos existentes nas áreas automotiva, aviação (militar e comercial) e computacional que sejam capazes de atender aos requisitos impostos por aplicações da classe C [SAE94].

Dos protocolos apresentados pela SAE, foram escolhidos os seguintes para a realização de uma análise de performance:

MIL-STD-1553B por ser o padrão militar em aviãoica.

CAN por ser a mais conhecida e difundida rede intra-veículo no setor automotivo.

J1850 por ser padrão entre os fabricantes automotivos americanos.

VAN por ter sido normalizado pelo BNA⁶ na França e estar sendo utilizado como modelo de protocolo pelas empresas automotivas PSA, Renault e Peugeot desde 1992.

Além destes, também foi considerada a proposta da empresa Chrysler (C²D) que está sendo utilizada desde 1990 e a proposta do INRIA denominada CSMA/DCR, que vem a ser um protocolo que garante o determinismo utilizando o padrão IEEE 802.3. Este último foi selecionado por apresentar um novo conceito, já que a maioria dos protocolos automotivos estudados utilizam o método de contenção para evitar colisões. O CSMA/DCR não evita que as colisões ocorram, mas administra-as de tal forma que a prioridade de cada estação seja sempre respeitada.

O protocolo FIP foi escolhido por ser uma proposta de padrão para redes *Fieldbus* e para servir como parâmetro de comparação, já que é uma rede para aplicações industriais.

A tabela 3.1 apresenta algumas características dos protocolos reconhecidos pela SAE e as outras três propostas escolhidas (CSMA/DCR, C²D e FIP).

3.7 Conclusão

No presente capítulo, foram apresentados conceitos no contexto de redes intra-veículos bem como uma analogia com redes locais industriais.

Também foram definidos critérios de projeto empregados em sistemas aviônicos de uma forma geral e veículos não tripulados em particular.

O perfil da aplicação foi definido segundo estes referenciais, direcionando a escolha de protocolos para simulação em um universo conhecido.

Finalmente, os protocolos selecionados foram apresentados juntamente com as respectivas justificativas para escolha.

⁵Society of Automotive Engineers

⁶Bureau de Normalisation d l'Automobile

Nome Prot.	Afil.	Aplic. Usual	Meio Transm.	Acesso Meio	Dados bytes/frame	Erro	Baud.	Comprim.	No. Nós
MIL-STD 1553B	Padrão Militar	Aviônica	Par Tranç. Fibra Ótica	Mestre/Escravo	2	Parid.	1 Mb/s	-	-
ABUS	VW	Auto Intr.V.	Único Fio	contenção	2	bit	500 kb/s	30 m	32
AUTOLAN	General Instr.	Auto Intr.V.	Par Tranç.	Mestre/Escravo	2	CRC	4 Mb/s	<40 m	127
CAN	ISO Bosch	Auto Intr.V.	Par Tranç. Fibra Ótica	contenção	0-6	CRC	1 Mb/s	>40 m	16
D ² D	Philips	Audio/Vídeo	Par Tranç.	contenção Escravo	2-128	parid.	100 kb/s	150 m	50
Ethernet	IEEE 802.3	Lab. e Comércio	Cabo Coaxial	contenção	46-1500	CRC	10 Mb/s	500 m	1024
J1850	SAE	Auto Intr.V.	Par Tranç. Único Fio	contenção	0-8	CRC opc.	10 Mb/s 40 Mb/s	40 m	-
J1865	SAE	Aviônica	Fibra Ótica	token-passing	32-65536	CRC	50 Mb/s	>40 m	-
Mini-Map	IEEE	Indústria	Par Tranç. Fibra Ótica Cabo Coaxial	token-passing	0-256	CRC opc.	5 Mb/s 10 Mb/s 20 Mb/s	400 m	100
VAN	Renault	Auto Intr.V.	Par Tranç. Fibra Ótica	contenção	0-8	CRC	100 kb/s	20 m	16
C ² D	Chrysler Bosch	Auto Intr.V.	Par Tranç. Fibra Ótica	contenção	0-4	CRC opc.	7812.5 b/s	20 m	16
FIP	ClubFIP	Indústria	Par Tranç.	Mestre/Escravo	0-8	CRC	1 Mb/s 2.5 Mb/s	500 m	-
CSMA/DCR	Inria	Redes Locais	Par Tranç.	CSMA/DCR	0-8	CRC	10 Mb/s	500 m	1024

Tabela 3.1: Características dos Protocolos Conhecidos p/ Classes B e C

Capítulo 4

Descrição dos Protocolos

4.1 Introdução

Como a análise realizada neste trabalho é de natureza comparativa, procurou-se descrever informações relevantes e ao mesmo tempo comuns às várias propostas.

Na impossibilidade de descrever exaustivamente protocolos com características diversas, optou-se por apresentar apenas algumas características básicas consideradas fatores de distinção entre eles do ponto de vista da aplicação em questão.

Assim, para cada protocolo os seguintes tópicos são abordados:

1. **Histórico:** Aqui, um breve histórico é apresentado, descrevendo as principais aplicações e usuários. Também deverão estar presentes informações relativas ao estado atual de normalização do protocolo.
2. **Características Gerais:** Um resumo das características que diferenciam esta proposta das demais.
Ex: *baudrate*, detecção de erro, acesso ao barramento, etc ...
3. Topologia da Rede
4. Camadas do Modelo OSI : descreve as camadas do modelo OSI definidas pelo protocolo.
5. Tipos e Estrutura dos *Frames*
6. Método de Acesso ao Barramento
7. Detecção de Erros
8. Disponibilidade de *hardware*: Um resumo dos principais fabricantes (se houverem) e compatibilidade.

4.2 O Protocolo CAN - Controller Area Network

4.2.1 Histórico

Em 1983, as indústrias BOSCH iniciaram um trabalho de pesquisa e análise dos protocolos para sistemas baseados em barramento existentes no mercado com a finalidade de estudar a viabilidade de sua utilização no interior de um veículo.

Em 1984 foi definida uma primeira versão para o protocolo CAN, sendo que no ano seguinte iniciou-se o desenvolvimento do chip para implementação das camadas OSI 1 e 2.

Em 1987 foi produzido o primeiro chip (82526 - Intel) para testes e em 1990 iniciou-se a produção em série.

A partir de 1991, vários fabricantes foram licenciados para a fabricação, além da Intel. São eles: Philips, Motorola, Siemens, SGS - Thompson, National e NEC [Aze91].

Apesar de ter sua concepção voltada para a integração de sensores e atuadores embarcados em um veículo, os grupos de trabalho da área de automação vislumbraram a adequação do protocolo para redes locais industriais, formando a CiA (*CAN in Automation*), constituída de usuários e fabricantes de produtos baseados no protocolo. Até abril de 1993, a CiA já contava com 64 associados advindos de empresas não ligadas à indústria automobilística [Reu93].

4.2.2 Características Gerais

O CAN possui as seguintes propriedades [fS93] [TW94]:

Baudrate: O CAN possui dois níveis de taxa de transmissão: baixa velocidade (até 125 kbits/s) e alta velocidade (até 1 Mbits/s).

Número Máximo de Nós: A norma não especifica um limite, mas usualmente o número de nós varia em torno de 16 [SAE94].

Distância: Este valor também não é especificado, mas os valores mais utilizados variam em torno de 40 metros [SAE94].

Frames: A informação é enviada através do barramento por *frames* de formato predefinido e tamanhos variáveis, mas limitados.

Ambiente: Pode operar em ambientes com alto índice de ruído eletromagnético [TBW95].

Codificação dos Bits: O CAN utiliza o método NRZ (*Non-Return Zero*) com bit de preenchimento¹ igual a 5 (ver figura 4.1). O nível lógico é constante para todo o campo do bit (isto é, 1 ou 0) e um bit de valor complementar é inserido no campo pelo transmissor se mais de 5 bits de mesmo estado estão sendo enviados. O receptor remove o bit inserido na recuperação dos dados.

Método de Acesso ao Barramento: Quando o barramento está livre, qualquer nó conectado pode iniciar a transmissão de um *frame*. Se dois ou mais nós iniciarem a transmissão de um *frame* ao mesmo tempo, o conflito no acesso é resolvido por uma arbitragem baseada em contenção usando um identificador. O mecanismo de arbitragem garante que nenhuma informação será perdida por causa da colisão. O transmissor com o *frame* de mais alta prioridade ganha o acesso.

Prioridade: Em CAN, a prioridade é atribuída à mensagem, não ao nó.

Roteamento dos Dados: No sistema CAN os receptores aceitam ou não a informação baseados num processo chamado *Frame Acceptance Filtering* que decide se a informação é relevante ou não.

Flexibilidade do Sistema: Novos nós podem ser adicionados à rede CAN sem necessitar de qualquer troca de *software* ou *hardware* nos outros nós, se o nó adicionado não é um transmissor de *frames* de dados ou se não requer dados adicionais que necessitam ser transmitidos.

Consistência dos Dados: Numa rede CAN, pode-se garantir que um dado será recebido por todos os nós ou por nenhum nó. Assim, a consistência dos dados é uma propriedade do sistema garantida pelos conceitos de *multicast* e manipulação de erros.

¹Também conhecido como bit de transparência

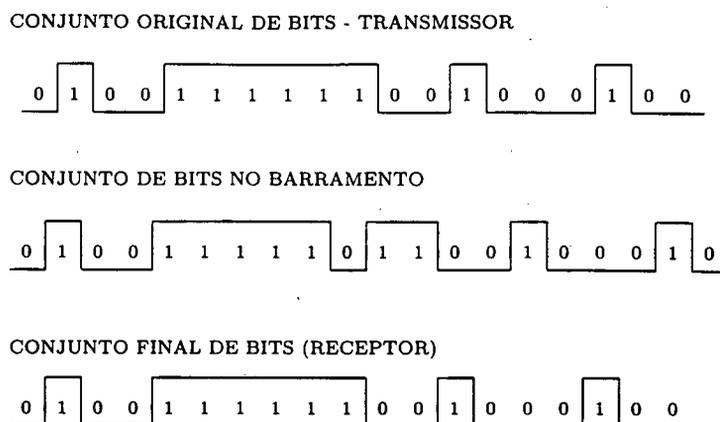


Figura 4.1: Codificação dos Bits em CAN

Requisição Remota de Dados: Através do envio de um *frame* de requisição, um nó pode requisitar a outro uma determinada informação.

Tempo de Recuperação: *Frames* corrompidos são detectados pelo nó transmissor e por qualquer nó receptor. O tempo de recuperação da detecção do erro até a possibilidade de reiniciar a transmissão do próximo *frame* é da ordem de 17 a 23 tempos de bits², sendo que no caso de distúrbios mais sérios no barramento, pode chegar a 29 tempos de bits.

Reconhecimento: Todos os receptores checam a consistência do *frame* recebido e sinalizam a chegada de um *frame* inconsistente.

Confinamento de Falhas: Os nós CAN estão aptos a distinguir pequenos distúrbios de falhas permanentes. Os nós com falhas são logicamente desconectados da rede e perdem a capacidade de transmitir ou receber qualquer dado.

Sinalização de Erro (*Error-Active*): Um nó pode avisar aos outros nós conectados ao barramento que um erro foi detectado. Para isto, ele envia um flag de erro-ativo (*error-active flag*) que consiste em seis (6) bits dominantes consecutivos que violam a regra do bit de preenchimento (*stuff bit*). A princípio, todos os nós são *error-active*. Um nó passa a ser *error-passive*, quando o contador de erros de transmissão ou o contador de erros de recepção ultrapassam a marca dos 127 erros.

Sinalização de Erro (*Error-Passive*): Neste caso, o nó envia um flag de erro-passivo (*error-passive*) que consiste em seis (bits) recessivos consecutivos. Após a transmissão, um nó erro-passivo (*error-passive*) terá que esperar um tempo adicional antes de iniciar uma nova transmissão.

4.2.3 Topologia da Rede

O CAN é baseado num sistema *multimaster*, portanto, qualquer nó pode assumir o controle do barramento desde que ele esteja disponível. A figura 4.2 ilustra o modelo topológico do protocolo.

4.2.4 O Modelo de Referência OSI e o CAN

Em relação ao modelo de referência OSI, a arquitetura do CAN define apenas as camadas física e a de enlace de dados (vide figura 4.3).

²tempo de bit - tempo que 1 bit leva para ser transmitido da estação origem à estação destino

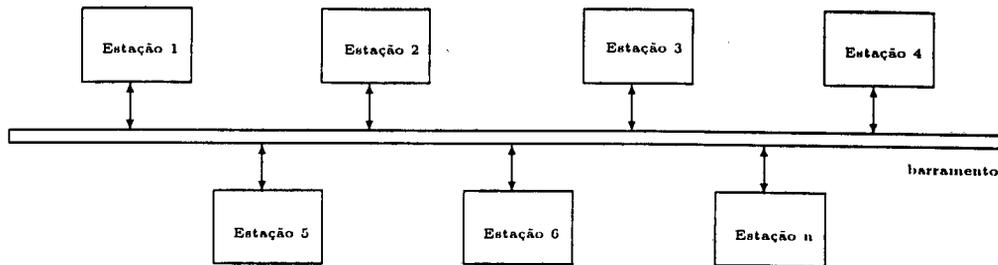


Figura 4.2: Topologia da Rede CAN

Camada OSI - 7 Aplicação	Especificada pelo projetista Adaptada à concepção de software existente Normalização em progresso
Camada OSI - 6 Apresentação	Vazio
Camada OSI - 5 Sessão	Vazio
Camada OSI - 4 Transporte	Vazio
Camada OSI - 3 Rede	Vazio
Camada OSI - 2 Enlace de Dados	Coberta pela especificação do protocolo CAN e padronizada pela ISO. Implementação em circuitos integrados (microcontroladores).
Camada OSI - 1 Física	Parcialmente coberta pela especificação do protocolo CAN Coberta pelo padrão ISO (livre escolha do meio de transmissão). Produtos existentes para transmissão a fio e fibra ótica.

Figura 4.3: Modelo de Referência OSI para o Protocolo CAN

- A camada 1 do modelo OSI representa a interface para o meio de transmissão. O meio de transmissão não é parte da especificação do protocolo CAN [Reu93], isto é, a escolha do meio fica a cargo do usuário sendo que encontram-se disponíveis no mercado tanto dispositivos para interfaceamento com o meio físico através de transmissão elétrica como ótica.
- A camada 2 é especificada pelo protocolo, estando disponíveis dispositivos para a integração desta camada com a camada 1.
- As camadas 3 a 6 são suprimidas com o objetivo de melhorar o desempenho e reduzir o custo de implementação.
- A camada 7 representa a interface com o usuário e não é padronizada pela ISO, mas a CiA iniciou um trabalho de padronização baseada num documento originalmente concebido para aplicações CAN na área médica. A implementação da camada 7 pela CiA possui aspecto apresentado na figura 4.4.

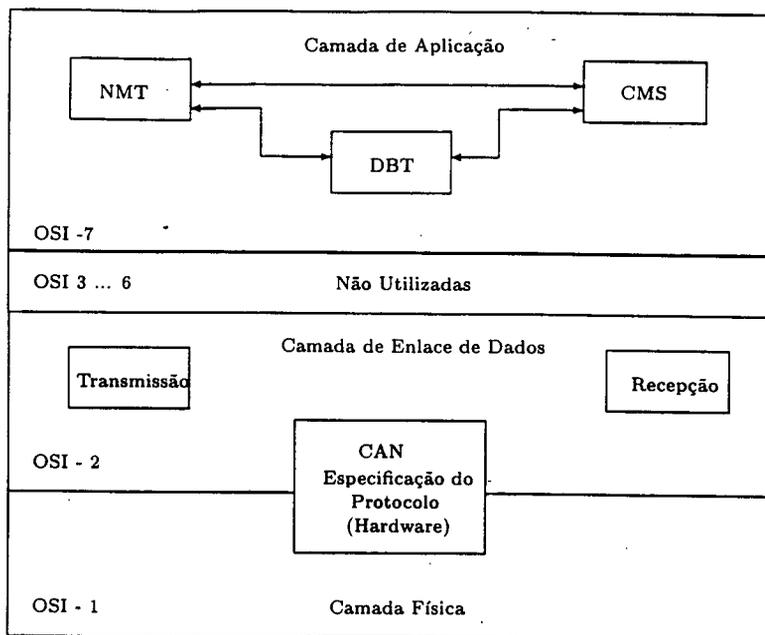


Figura 4.4: Implementação da camada de Aplicação pela CiA

A camada de aplicação é estruturada da seguinte forma: o módulo NMT (*Network Management*) é responsável pela inicialização da rede e a manipulação dos erros detectados pelo sistema.

O módulo CMS (*CAN Message Specification*) fornece uma série de serviços como ler e escrever em variáveis e tratamento de eventos que ficam disponíveis ao usuário. Este módulo é um subconjunto do módulo MMS (*Manufacturing Message Specification*) que não aparece na figura 4.4 por ser genérico para redes locais industriais que possuem a camada 7 especificada.

O módulo DBT (*Distributor*) provê uma distribuição dinâmica de nomes definidos pelo usuário para identificadores de mensagens.

4.2.5 Formato das Mensagens

Quatro diferentes tipos de *frames* são utilizados para realizar as tarefas de transmissão e recepção de dados entre os nós em um sistema CAN [fS93]:

1. *Frames* que carregam dados do transmissor para o receptor
2. *Frames* remotos que são transmitidos por um nó para requisitar a transmissão de um *frame* de dados de outro nó
3. *Frames* de erro transmitidos por qualquer nó na detecção de erro no barramento
4. *Frames* de *overload* usados para fornecer um atraso extra entre *frames*

Estrutura dos Frames de Dados

Os *frames* de dados são compostos por sete tipos diferentes de campos de bits e possuem o formato apresentado na tabela 4.1. A seguir, uma descrição dos campos.

Início	Arbitragem	Controle	Dados	CRC	Reconhecim.	Fim
1 bit	12 bits	6 bits	0-8 bytes	16 bits	2 bits	7 bits

Tabela 4.1: Formato do Frame de Dados CAN

Início: é o campo delimitador de início do *frame* e funciona como um sinal de sincronização para todos os nós. Consiste em um único bit dominante (0).

Arbitragem: é composto pelo *frame* identificador (que determina a prioridade da mensagem) e o RTR (Remote Transmission Request) que determina se o *frame* é de requisição ou de dados. Este campo é construído na subcamada LLC.

Controle: é composto de 2 bits para futuras expansões e 4 bits para o tamanho do campo de dados em bytes cujo valor máximo é 8.

Dados: são os dados propriamente ditos.

CRC: é o campo de checagem de CRC. É formado por 15 bits de CRC mais 1 bit recessivo (delimitador).

Reconhecimento: Este campo é sobrescrito por todos os nós que receberam o dado corretamente.

Fim: é um delimitador de fim de *frame* constituído por 7 bits recessivos.

Estrutura dos Frames Remotos

Um nó atuando como receptor para um certo dado pode iniciar a transmissão do respectivo dado por seu nó fonte enviando um *frame* remoto. Os *frames* remotos são compostos por seis tipos diferentes de campos de bits, conforme a tabela 4.2.

Início	Arbitragem	Controle	CRC	Reconhecim.	Fim
1 bit	12 bits	6 bits	16 bits	2 bits	7 bits

Tabela 4.2: Formato do Frame Remoto CAN

A estrutura dos campos, como pode ser visto, é muito semelhante à estrutura do *frame* de dados. A diferença, é que além deste *frame* não possui o campo de dados, o valor do bit RTR (*Remote Transmission Request*) no campo de arbitragem é igual a 1.

Estrutura dos Frames de Erro

O *frame* de erro possui apenas dois campos de bits, conforme pode ser visto na tabela 4.3. O primeiro campo é dado pela superposição de *flags* de erro provenientes de diferentes nós. O segundo é o delimitador de erro.

Flag de Erro	Delim. de Campo
6 - 12 bits bit	8 bits

Tabela 4.3: Formato do Frame de Erros CAN

Flags de Erro: Existem dois tipos de *flags* de erros: *flag* de erro passivo e *flag* de erro ativo.

Erro Ativo: Um nó ativo, quando detecta uma condição de erro, transmite para o barramento o *flag* de erro. O formato do *flag* de erro viola a regra de preenchimento dos bits ou destrói o campo de bits de um determinado formato. Como consequência, todos os outros nós detectam a condição de erro e também passam a transmitir este *flag*. Isto resulta numa superposição de *flags* de erro provenientes de diferentes nós. O tamanho desta seqüência varia de 6 a 12 bits.

Erro Passivo: Os *flags* de erro enviados por receptores não prevalecem sobre qualquer atividade do barramento. Além disso, os receptores de erro-passivo sempre têm que esperar 6 bits subseqüentes iguais após detectar uma condição de erro, até ter completado seu *flag* de erro.

Delimitador de Erro: O delimitador de erro consiste em 8 bits recessivos (11111111).

Estrutura dos Frames de Overload

Existem dois tipos de *frames* de *overload* e ambos possuem o mesmo formato:

1. Frame de *overload* de requisição do LLC³ - Este *frame* de *overload* será requisitado pelo LLC para indicar uma situação interna de *overload*.
2. Frame de Reativação de *overload* - A transmissão de reativação de *overload* será iniciada pela subcamada MAC sob certas condições de erro.

Ambos os *frames* possuem os campos de bits mostrados na tabela 4.4.

Flag de Overload	Delim. de Overload
6 bits	8 bits

Tabela 4.4: Formato do Frame de *Overload* CAN

Espaços Interframes

Os *frames* remotos e os de dados são separados dos *frames* precedentes de acordo com seu tipo (*frame* de dados, *frame* remoto, *frame* de erro e *frame* de *overload*) por um campo de bits chamado espaço *interframes*. Os *frames* de *overload* e erro não são separados por estes espaços.

4.2.6 Método de Acesso ao Meio

O seguinte conjunto de termos descreve o método de acesso ao meio utilizado pelo CAN. Este método também é utilizado por outros protocolos e é conhecido pela sigla CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) ou método de contenção não destrutivo com arbitragem bit a bit.

Multimaster: Cada nó é o mestre do barramento durante a transmissão de um *frame* de dados ou remoto.

Detecção de Barramento Disponível: O barramento é considerado livre para qualquer nó após ter detectado que o espaço *interframes* não foi interrompido por um bit dominante.

³A camada de enlace de dados é subdividida em duas sub-camadas: LLC (*Logical Link Control*) e MAC (*Medium Access Control*)

Acesso ao Barramento: Um nó erro-ativo pode acessar o barramento tão logo ele esteja livre. Quando um nó erro-passivo é o receptor do *frame* corrente ou de um anterior, ele também pode acessar o barramento assim que este estiver livre. Já se o nó erro-passivo for o transmissor de algum *frame*, este pode acessar o barramento assim que for concluído o aborto da transmissão, e se garante que nenhum outro nó iniciou sua transmissão enquanto isso. Quando vários nós iniciam suas transmissões simultaneamente, o nó que tiver a mais alta prioridade será o mestre do barramento.

Arbitragem Baseada em Contenção: Durante a arbitragem, cada transmissor compara o nível do bit transmitido com o nível do bit monitorado no barramento. Se estes níveis forem iguais, o nó pode continuar enviando. Se os níveis forem diferentes, o nó perde o direito de transmitir e se retira do processo. Para maior clareza, pode-se fazer uma analogia com o operador lógico AND. Assim, começando do bit mais significativo, todo o campo identificador (que contém a prioridade) realizará uma operação AND entre cada bit transmitido e o valor lido no barramento logo após a transmissão [ZS95].

Prioridade do Frame: A arbitragem baseada em contenção é realizada sobre o identificador e sobre o bit RTR seguinte ao identificador. Entre *frames* com identificadores diferentes, a alta prioridade é atribuída ao *frame* com valor de identificador com valor binário mais baixo. Se um *frame* de dados e um *frame* remoto com o mesmo identificador iniciam a transmissão simultaneamente, o *frame* de dados terá prioridade.

Resolução de Colisão: Além do princípio de que a transmissão pode ser iniciada assim que o barramento estiver livre, existem outros princípios para a resolução de uma colisão:

- Em um sistema, cada informação precisa ser definida por um único identificador, limitando, desta forma, em 2032 o número de combinações possíveis para identificador de mensagem⁴.
- Um *frame* de dados com um dado identificador e tamanho diferente de zero pode ser iniciado por um único nó. *Frames* remotos podem ser transmitidos somente com um tamanho determinado pelo DLC do *frame* de dados correspondente.
- Transmissões simultâneas de *frames* remotos com identificadores iguais e diferentes tamanhos de dados levam à uma situação de colisão insolúvel.

4.2.7 Detecção de Erros

Existem cinco diferentes tipos de erros que podem ser detectados (não são mutuamente exclusivos):

1. Erro de Bit - Um nó que está enviando um dado pelo barramento também monitora o barramento. O erro de bit ocorre quando o valor monitorado é diferente do valor enviado.
2. Erro de Preenchimento - Um erro de preenchimento é detectado quando existem seis bits consecutivos de mesmo nível.
3. Erro de CRC - A seqüência de CRC consiste no resultado do cálculo de CRC feito pelo transmissor, calculado através da equação 4.1:

$$(x^{15} + x^{11} + x^{10} + x^9 + x^8 + x^7 + x^4 + x^3 + x^2 + 1) \quad (4.1)$$

Os receptores calculam o CRC do mesmo modo que os transmissores. O erro é detectado quando estes valores diferem entre si para um mesmo *frame*.

4. Erro de Formato - Um erro de formato é detectado quando um determinado *frame* de formato fixo contém um ou mais bits "ilegais".
5. Erro de Reconhecimento - É detectado pelo transmissor quando ele não monitora um bit dominante durante o *slot* ACK.

⁴2032 é igual a 2^{11} (onde 11 é o número de bits do identificador) menos as combinações que resultam em 5 bits consecutivos iguais.

4.2.8 Disponibilidade de Hardware

Como visto na seção 4.2.1, existem vários fabricantes de circuitos integrados que implementam este protocolo.

Com relação ao interfaceamento, este pode ser realizado tanto através de um microcontrolador comum, compatível com a família 8031 [Sem92] como processadores mais sofisticados, como um *transputer* [Gmb].

4.3 O Protocolo VAN - Vehicle Area Network

4.3.1 Histórico

O protocolo de comunicação de dados VAN (*Vehicle Area Network*) foi normalizado experimentalmente em 1990 pelo *Bureau de Normalisation de l'Automobile* [BNA90], sendo utilizado em 1992 pela **PSA-Renault** como modelo de referência [dEdRedP92] e também pela **Peugeot S.A.** [OLCH92].

Segundo [Ass91], um acordo de colaboração comum entre a PSA, Renault, Siemens e Magneti Marelli foi firmado com o objetivo de produzir um documento base para a proposta de padronização pela BNA, AFNOR e comitês da ISO.

Até 1994, a proposta estava sendo considerada pelo comitê técnico da ISO 22/SC3/WG1 [SAE94].

O protocolo possui como objetivos interconectar diferentes tipos de módulos eletrônicos embarcados em um veículo e transmitir mensagens respeitando os diferentes níveis de prioridade e baseia-se no seguintes conceitos de estado da arte [Ass91]:

- Utiliza o modelo de operação mestre/escravo para as aplicações da classe A como os protocolos D²B e I²C (Philips).
- Arbitragem de barramento baseada em contenção não destrutiva utilizando comparação bit a bit como nos protocolos D²B e I²C (Philips), C²D (Chrysler), J1850 (SAE) e CAN (Bosch).
- A subcamada LLC é definida em conformidade com o modelo ISO, como nas redes para aplicações em robótica: FIP, LAC2 (Compex) e Factor (Aptor).

O VAN descreve um modelo de transmissão de dados do tipo assíncrono e a transferência é efetuada em pacotes.

O protocolo pode ser utilizado na multiplexação de dados intra-veículo em aplicações das classes A, B e C [OLCH92].

4.3.2 Características Gerais

O VAN possui as seguintes propriedades [BNA90]:

Baudrate: Apesar deste parâmetro ser definido pelo usuário, podendo chegar a valores superiores a 250 Kbits/s. Segundo [OLCH92], em aplicações da classe B, o VAN suporta valores máximos de 100 Kbits/s.

Número Máximo de Nós: É recomendada a interligação de até 16 módulos, desde que o comprimento do barramento não ultrapasse 20 metros.

Frames: A informação é enviada através do barramento por *frames* de formato predefinido e tamanhos variáveis, mas limitados.

Método de Acesso ao Barramento: Utiliza o sistema não-destrutivo de arbitragem bit a bit por prioridade. Quando o barramento está livre, qualquer nó conectado pode iniciar a transmissão de um *frame*. Se dois ou mais nós iniciarem a transmissão de um *frame* ao mesmo tempo, o conflito no acesso é resolvido por uma arbitragem baseada em contenção usando um identificador. O mecanismo de arbitragem garante que nenhuma informação será perdida por causa da colisão. O transmissor com o *frame* de mais alta prioridade ganha o acesso.

Codificação do Bit: Existem dois tipos diferentes que podem ser selecionados pelo usuário: L-Manchester e E-Manchester. Quando L-Manchester é selecionado, todos os bits do *frame* são codificados em formato Manchester. A seleção em E-Manchester codifica todos os bits em NRZ, exceto o último bit de cada *nibble* e o campo ACK, que são codificados em Manchester, conforme figura 4.5.

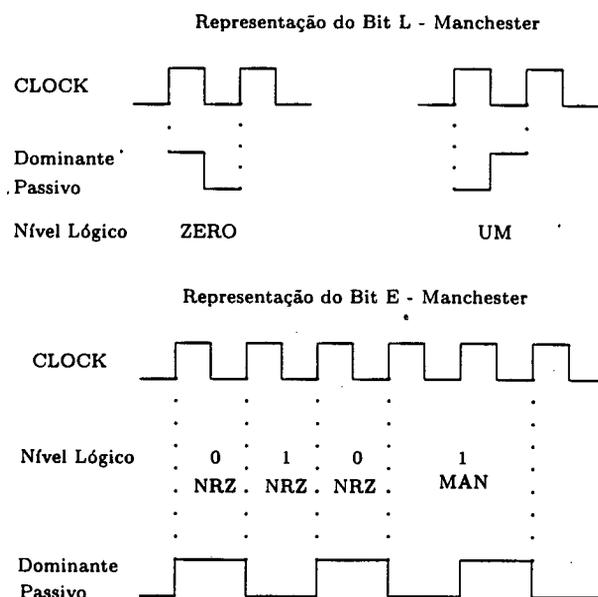


Figura 4.5: Codificação dos Dados em VAN

Prioridade: Em VAN, a prioridade é atribuída à mensagem, não ao nó.

4.3.3 Topologia da Rede

O protocolo utiliza a topologia tipo barramento, em que todos os nós estão conectados ao mesmo meio físico.

Com a atribuição de uma prioridade a cada mensagem, é possível realizar a arbitragem garantindo sempre a seqüência de direito de envio à mensagem mais prioritária sem a perda de nenhum dado (no caso de mais de um nó detectar barramento desocupado ao mesmo tempo e necessitar enviar dados).

4.3.4 O Modelo de Referência OSI e o VAN

De acordo com o modelo de referência OSI, a arquitetura do VAN define apenas as camadas física e a de enlace de dados. A figura 4.6 ilustra as camadas e funções implementadas pelo protocolo.

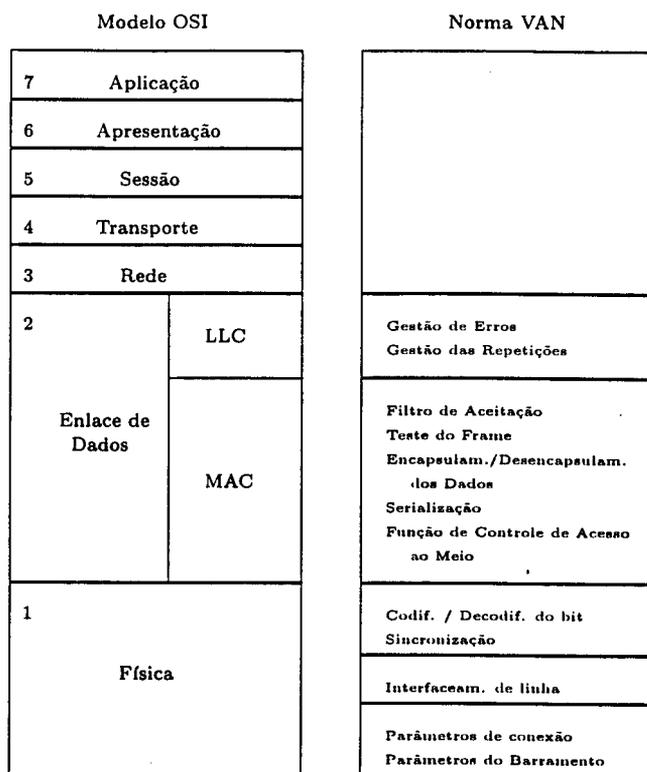


Figura 4.6: Paralelo entre o Modelo OSI e as Camadas Implementadas em VAN

4.3.5 Formato das Mensagens

Os *frames* utilizados no protocolo VAN possuem o formato apresentado na tabela 4.5⁵. A seguir, uma descrição dos campos.

SOF	ID	COM	Data	FCS	EOD	ACK	EOF	IFS
6	12	4	(0-28) * 8	15	1	1	4	2

Tabela 4.5: Formato do Frame VAN

SOF: É o *Start Of Frame* ou delimitador de início do *frame*. Utilizado para sincronização.

ID: É o campo identificador da mensagem, que inclui sua prioridade.

COM: É o *COM*mand ou campo de comando. Este campo é formado por 4 bits:

1. Bit setado caso o dado tenha sido enviado por requisição (RTR).
2. Bit setado se um sinal de reconhecimento é requisitado (RAK).
3. Bit setado para selecionar leitura ou escrita em um periférico (R/W).
4. Bit reservado para uso futuro.

⁵Todos os valores apresentados na segunda linha são em bits

Data: São os dados propriamente ditos. Utilizando o controlador modelo *Compact*, para periféricos mais simples, a limitação do campo de dados é de 8 bytes. O módulo *Full VAN*, que utiliza arquitetura RISC, suporta até 28 bytes de dados. Para a simulação, cada controlador será considerado separadamente.

FCS: É o *Frame CheckSum*, utilizado para a checagem de erro CRC.

EOD: É o *End Of Data*, utilizado para delimitar o campo de dados.

ACK: É o *ACKnowledge*, utilizado para reconhecimento.

EOF: É o *End Of Frame*, utilizado para delimitar o *frame*.

IFS: é o *Inter Frame Space*, espaço mínimo necessário entre mensagens.

4.3.6 Método de Acesso ao Meio

O acesso ao barramento no protocolo VAN utiliza o mesmo mecanismo de arbitragem bit a bit do protocolo CAN (vide seção 4.2.6).

4.3.7 Detecção de Erros

O VAN provê uma série de mecanismos de detecção de erro através da subcamada MAC. Os erros detectados são sinalizados para a administração da rede através de dados estatísticos. A tabela 4.6 descreve estes mecanismos.

Erro	Significado
Erro de bit	Erro detectado quando o bit emitido é diferente do bit detectado no barramento
Erro de Código	Erro detectado quando ocorre violação dos códigos Manchester-L ou Manchester Comprimido
Erro de Formato	Erro de codificação ou de sincronização nos campos fixos do <i>frame</i> como SOF, EOD, ACK e EOF
Erro de CRC	O valor do CRC calculado pelo receptor difere do valor recebido
Erro de Reconhecim.	Erro detectado no campo ACK

Tabela 4.6: Mecanismos de Detecção de erros em VAN

O CRC é calculado pela equação 4.2.

$$(x^8 + x^4 + x^3 + x^2 + 1)(x^7 + 1) \quad (4.2)$$

4.3.8 Disponibilidade de Hardware

O protocolo VAN possui uma família de circuitos integrados que se subdivide em duas classes:

Controladores: O projeto de controladores realizado pelas indústrias automobilísticas PSA e a Renault é constituído de duas versões:

1. **Básico** - A versão básica ou compacta, como também é conhecida, possui como principais características a taxa de transmissão de até 250 Kbits/s e aceitação de *frames* com até 8 bytes de dados.
2. **Avançado** - Esta versão, também chamada *Full VAN*, tem seu processador baseado em arquitetura RISC⁶. Com isto, *frames* com até 28 bytes de dados e as taxas de transmissão superiores a 250 Kbits/s são suportados.

Periféricos: Foram definidas especificações técnicas para toda uma família de ASICs⁷ projetadas para a indústria automotiva que compreende sistemas para controle do espelho retrovisor, aquisição de dados mecânicos, controle de chaves, etc ...

Até 1993 [SAE94] o processo de industrialização de componentes ainda estava em fase de especificações técnicas, não havendo nenhum produto disponível para comercialização.

4.4 O protocolo J1850

4.4.1 Histórico

O protocolo de comunicação de dados J1850 foi proposto pela SAE (*Society of Automotive Engineers*) em 1990 com o objetivo de normalizar interfaces de redes de comunicação de dados da classe B.

O documento [SAE92] estabelece requisitos mínimos para que as recomendações possam ser utilizadas tanto em aplicações mais simples como em sistemas complexos flexíveis, utilizando uma filosofia de rede de comunicação de dados que satisfaça as necessidades dos fabricantes da indústria automotiva.

4.4.2 Características Gerais

Baudrate: Os baudrates são especificados em 10.4 kbits/s e 41.6 kbits/s.

Prioridade: A prioridade é atribuída à mensagem, não ao nó.

Distância: A distância máxima aceita como comprimento total da rede é de 40 metros.

Número de Nós: A norma não especifica o número máximo de nós.

Meio Físico: A norma prevê a utilização de um único fio ou par trançado.

Codificação do Bit: Utiliza modulação por largura de pulso - PWM (*Pulse Width Modulation*). O formato dos bits 1 e 0 é mostrado na figura 4.7.

Acesso ao Barramento: Utiliza o sistema não-destrutivo de arbitragem bit a bit por prioridade.

Deteção de Erro: Inclui deteção de endereço fora do range, valor de bit inválido e estrutura de mensagem inválida e CRC.

Modo de Consumo Reduzido: Para reduzir o consumo de energia, o nó entra no modo *sleep* ou "dormindo" se o barramento ficar livre por um período maior que 500 ms. Qualquer atividade no barramento é suficiente para "despertar" o nó.

⁶Reduced Instructions Set Computing

⁷Application Specific Integrated Circuits

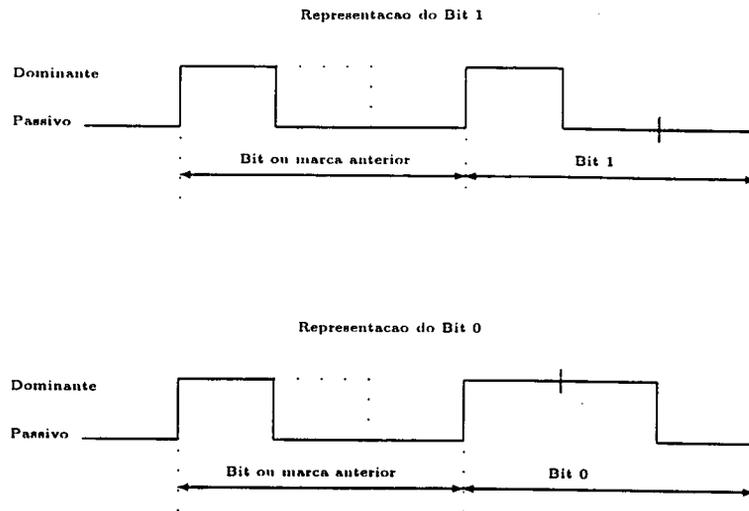


Figura 4.7: Codificação no J1850

4.4.3 Topologia da Rede

O protocolo utiliza a topologia tipo barramento, em que todos os nós estão conectados ao mesmo meio físico baseado no modelo *masterless*.

Com a atribuição de uma prioridade a cada mensagem, é possível realizar a arbitragem garantindo sempre a seqüência de direito de envio à mensagem mais prioritária sem a perda de nenhum dado (no caso de mais de um nó detectar barramento desocupado ao mesmo tempo e necessitar enviar dados).

4.4.4 O Modelo de Referência OSI e o J1850

Apesar do documento focar principalmente as camadas física e de enlace de dados, a recomendação inclui as sete camadas do modelo OSI, necessárias à emissão de diagnósticos segundo a legislação [SAE92]. A figura 4.8 descreve as funções definidas para cada camada OSI no protocolo J1850.

4.4.5 Formato das Mensagens

Os *frames* utilizados no protocolo J1850 possuem o formato [SAE94] apresentado na tabela 4.7. A seguir, uma descrição dos campos.

Início	Dados	CRC	EOD	Resp	EOM	IMS
1 bit	0 - 11 bytes	8 bits	1 bit	0 ou mais bits	1 bit	1 bit

Tabela 4.7: Formato do Frame J1850

Início: é o campo delimitador de início do *frame*.

Dados: é o campo de dados, formado por bytes de dados.

CRC: é o campo de checagem de CRC (*Cyclic Redundancy Checking*).

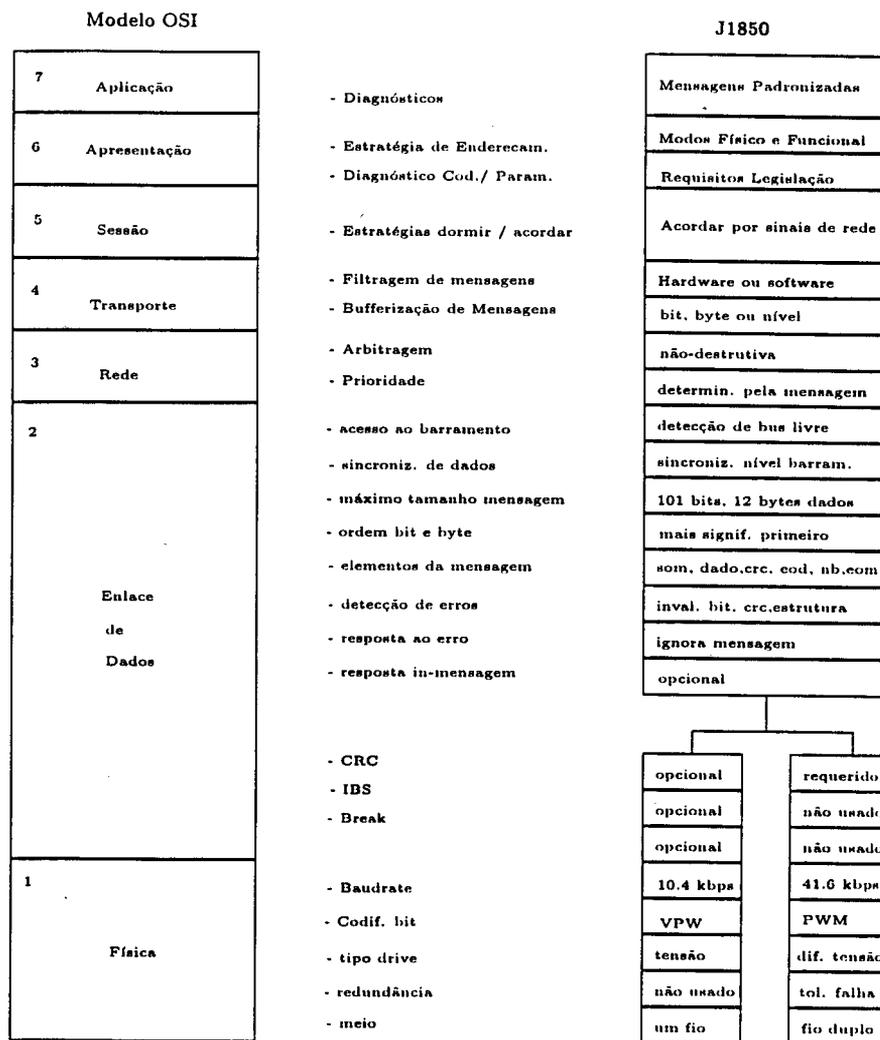


Figura 4.8: As Camadas do Modelo OSI e o Protocolo J1850

EOD: sinaliza o final do campo de dados (*End Of Data*).

Resp: é o campo que dá a oportunidade de alguma estação que recebeu o dado possa responder com algum tipo de mensagem.

EOM: é um delimitador de fim de mensagem (*End Of Message*).

IMS: é o espaço necessário entre mensagens (*Inter-Message Separation*).

Neste caso, a soma de todos os campos, desde o campo **Início** até o campo **EOM** (inclusive) não poderá ultrapassar 101 bits, limitando, assim, o campo de dados propriamente dito a 11 bytes de dados.

4.4.6 Método de Acesso ao Meio

O protocolo J1850 utiliza o mesmo mecanismo de arbitragem bit a bit não destrutiva apresentado para os protocolos CAN e VAN (vide seção 4.2.6).

4.4.7 Detecção de Erros

Existem cinco diferentes tipos de erros que podem ser detectados (não são mutuamente exclusivos):

1. Erro de Tamanho da Mensagem - Detectado quando o tamanho da mensagem excede os limites definidos.
2. Erro de Valor Fora dos Limites - Detectado quando o dado é corrompido por alguma interferência, gerando valores fora do limite (*range*) de aceitação do nó receptor.
3. Erro de CRC - A seqüência de CRC consiste no resultado do cálculo de CRC feito pelo transmissor, baseado no polinômio descrito na equação 4.3

$$x^8 + x^4 + x^3 + x^2 + 1 \quad (4.3)$$

Os receptores calculam o CRC do mesmo modo que os transmissores. O erro é detectado quando estes valores diferem entre si para um mesmo *frame*.

4. Erro de Bit/Símbolo Inválidos - No caso de bits, este erro ocorre quando o valor esperado possui um valor determinado (0 ou 1) e o resultado é diferente. No caso de símbolos, o erro ocorre pela supressão ou alteração dos campos EOD ou EOM do *frame*, fazendo com que a mensagem exceda seu comprimento máximo.

4.4.8 Disponibilidade de Hardware

Até onde se tem notícia, este protocolo possui o status de norma, não possuindo ainda produtos disponíveis para implementação.

4.5 O Protocolo MIL-STD-1553B

4.5.1 Histórico

O MIL-STD-1553B é uma norma militar para multiplexação de dados em veículos militares e foi publicada pela primeira vez em 1968, passando por sucessivas revisões até 1978, quando foi apresentada a versão utilizada neste trabalho.

Desde então, tem sido largamente utilizado na indústria militar em todo o mundo, inclusive no Brasil.

Suas aplicações mais comuns têm sido na área de aviónica em aeronaves, veículos lançadores de satélites e ônibus espaciais, além de tanques e armas de combate.

4.5.2 Características Gerais

O MIL-STD-1553B possui as seguintes propriedades [Dep80] [Cun92]:

Baudrate: Sua taxa de transmissão é de até 1 Mbits/s.

Número Máximo de Nós: A norma não especifica um limite.

Distância: Este valor também não é especificado.

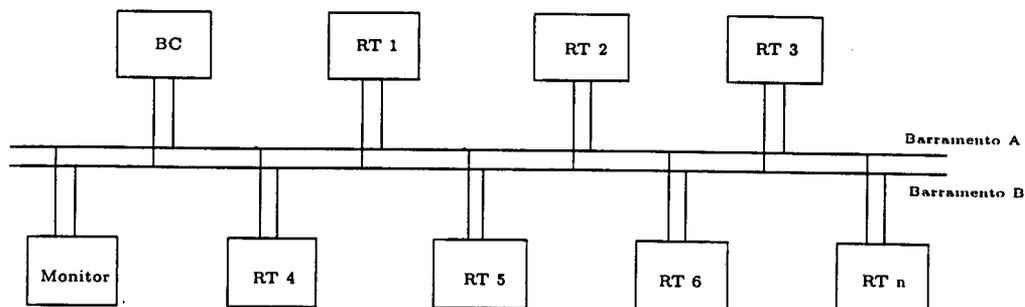
Meio Físico: O padrão define características específicas para o uso de par trançado blindado, mas outros meios podem ser utilizados. Uma norma correlata, a MIL-1773 prevê o suporte de fibras óticas [SAE94].

Codificação dos Bits: A codificação utilizada é o Manchester II.

Acesso ao Barramento: O método de acesso ao barramento é o sistema de Controle do Barramento com Mestre Estacionário, isto é, um dos nós é o mestre e controla todo o fluxo de dados no barramento.

4.5.3 Topologia da Rede

O protocolo MIL-STD-1553B utiliza uma topologia do tipo barramento, representada pela figura 4.9.



Onde: BC - Bus Controller
RT - Remote terminal

Figura 4.9: Topologia da Rede MIL-STD-1553B

4.5.4 O Modelo de Referência OSI e o MIL-STD-1553B

Formalmente, a norma MIL-STD-1153B não cita o modelo OSI e não faz referência a nenhuma de suas camadas até porque ela é anterior ao modelo, que foi publicado em 1984. Apesar disso, pode-se considerar que o protocolo especifica parâmetros para as camadas física e de enlace de dados.

4.5.5 Formato das Mensagens

As informações que circulam pelo barramento definidas pela MIL-STD-1553B estão classificadas em três tipos:

- Command Word - Frame de comando
- Status Word - Frame de Status
- Data Word - Frame de Dados

Frame de Comando

Este *frame* pode ser transmitido somente pelo BC para ordenar a um RT que receba ou transmita informações. O formato do *frame* de comando está representado na tabela 4.8.

Sincronismo: é usado para a sincronização de hardware.

Sincron.	Endereço Destino	T/R	Sub-end.	Cont.	Parid.
3 bits	5 bits	1 bit	5 bytes	5 bits	1 bit

Tabela 4.8: Formato do Frame de Comando MIL-STD-1553B

Endereço Destino: é o endereço da estação destino.

T/R: é a indicação para a estação que recebe o *frame* se ela deve transmitir ou receber um dado.

Sub-endereço: é utilizado caso seja necessário o redirecionamento do dado recebido⁸.

Contador: indica à estação destino quantos *frames* de dados completos ela vai receber (se for o caso de uma recepção).

Paridade: é a indicação de paridade. Todos os *frames* possuem paridade ímpar.

Frame de Status

Este *frame* pode ser transmitido somente por um RT para fornecer informações de seu estado geral quando requisitado pelo BC. O formato do *frame* de comando está representado na tabela 4.9, sendo que a segunda coluna indica o número de bits em cada campo.

S	E	Err	In	Srv	Rsrv	Rec	Oc	FS	Ac	FR	P
3	5	1	1	1	3	1	1	1	1	1	1

Tabela 4.9: Formato do Frame de Status MIL-STD-1553B

S: é usado para a sincronização de hardware.

E: é o endereço da estação destino.

Err: é a indicação de erro na mensagem recebida.

In: (Instrumentação) é utilizado para diferenciar *frames* de Status de *frames* de Comando. Num *frame* de status seu valor é sempre zero.

Srv: indica pedido de serviço. Permite a transferência de dados fora do período pré-estabelecido.

Rsrv: Estes bits são reservados para uso futuro.

Rec: Recepção de comando de *broadcast*.

Oc: Indica que o RT não está habilitado para responder ao comando.

FS: Indica falha em um sub-sistema ligado a um RT.

Ac: Indica aceitação de controle dinâmico do barramento e este RT irá aceitar imediatamente a tarefa de BC *backup*.

FR: Indica falha no RT.

P: Indica paridade. Todos os *frames* possuem paridade ímpar.

⁸Esta função não é necessária no modelo estudado

Frame de Dados

Este *frame* pode ser transmitido por qualquer nó da rede (no caso de um RT, o *frame* só será enviado se solicitado pelo BC). O formato do *frame* de dados está representado na tabela 4.10.

Sincron.	Dados	Parid.
3 bits	16 bits	1 bit

Tabela 4.10: Formato do Frame de Dados MIL-STD-1553B

Sincronismo: é usado para a sincronização de hardware.

Dados: são os dados propriamente ditos.

Paridade: é a indicação de paridade. Todos os *frames* possuem paridade ímpar.

4.5.6 Combinações de Frames para Transferência de Dados

No modelo estudado, das 10 diferentes combinações que podem existir para a troca de *frames* em uma rede MIL-STD-1553B, apenas 3 são necessárias para a transferência de dados periódicos como na aplicação estudada. São elas:

Transferência entre BC e um RT

Esta é utilizada quando o BC dispõe de uma informação (um cálculo realizado por ele, por exemplo) e deseja passá-la para um determinado nó da rede. O diálogo que se segue possui o formato descrito na figura 4.10.

Transferência entre um RT e o BC

Esta é utilizada quando o BC necessita de uma informação de um nó (a medição de um sensor, por exemplo). O diálogo também é mostrado na figura 4.10.

Transferência entre dois RTs

Esta é utilizada quando o BC necessita que dois nós troquem informações. O diálogo também é mostrado na figura 4.10.

4.5.7 Método de Acesso ao Meio

O protocolo MIL-STD-1153B está baseado em uma arquitetura mestre-escravo, onde um nó (mestre) controla a transmissão e recepção dos outros nós (escravos) [Dep80] [Cun92].

São definidos três tipos de terminais para o barramento:

- BC - Bus Controller
- RT - Remote Terminal
- BM - Bus Monitor

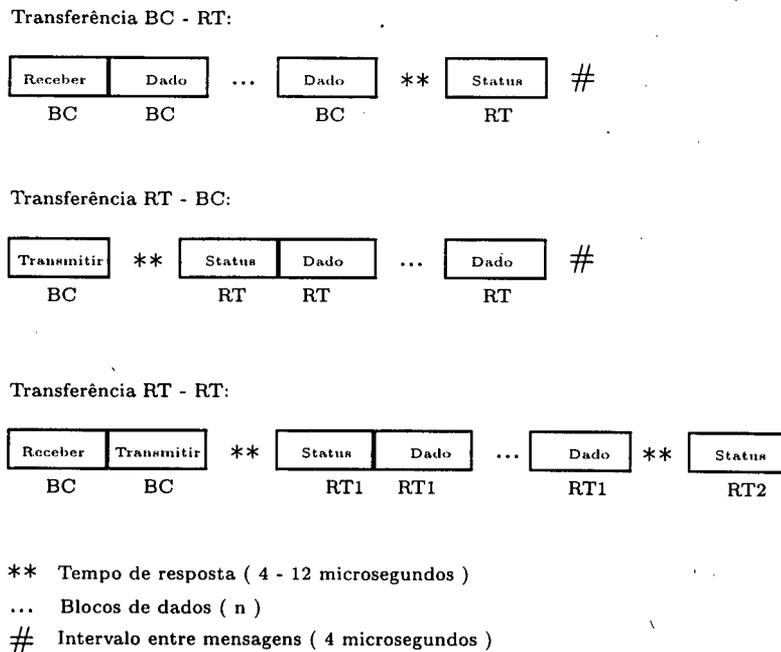


Figura 4.10: Diálogos em uma Rede MIL-STD-1553B

O nó denominado *Bus Controller* controla todo o fluxo de informações no barramento, requisitando a outro nó (aqui denominado *Remote Terminal*) o envio ou a recepção de dados.

No MIL-STD-1553B existe também a figura do nó *Bus Monitor*, que, além de manter registros do fluxo de dados, também pode funcionar como *backup* do *Bus Controller*.

4.5.8 Detecção de Erros

Para a detecção de erro, o protocolo dispõe dos seguintes mecanismos:

1. Validação da Palavra:
 - (a) Campo de validade de *frame* no início da palavra
 - (b) Bits codificados corretamente em Manchester II
 - (c) Paridade ímpar
2. Validação da mensagem (diálogo): É testada a continuidade do diálogo (o controlador do barramento conhece as palavras que compõe o diálogo e testa o intervalo entre elas).
3. Erros adicionais incluem:
 - (a) Erro de Mensagem
 - (b) Ocupado
 - (c) Flag de sub-sistema
 - (d) Erro de paridade
 - (e) Erro de sincornismo
 - (f) Código Manchester inválido

- (g) Erro no número de bits de dados
- (h) Descontinuidade das palavras de dados
- (i) Ausência de resposta via palavra de status

4.5.9 Disponibilidade de Hardware

Como este protocolo já é conhecido e utilizado há muitos anos na área de aviãoica militar, existem vários fabricantes que fornecem circuitos integrados para as camadas física e de enlace [Sny87].

4.6 O Protocolo C²D - Chrysler Collision Detection

4.6.1 Histórico

O protocolo de comunicação de dados C²D Bus (*Chrysler Collision Detection Serial Data Bus*) foi desenvolvido pela Chrysler Corporation com o objetivo de prover seus carros com um barramento serial automotivo para a comunicação de dados entre microprocessadores de diferentes módulos embarcados de classe B.

Desde 1989 a Chrysler acompanha os trabalhos dos Subcomitês de Multiplexação da SAE com o objetivo de padronização, sendo que desde o ano de 1988 a empresa vem utilizando este protocolo em seus automóveis.

4.6.2 Características Gerais

Codificação do Bit: Utiliza o formato NRZ (*Non-Return to Zero*).

Acesso ao Barramento: Utiliza o sistema não-destrutivo de arbitragem bit a bit por prioridade.

Deteccção de Erro: A deteccção de erro fica a cargo da aplicação, que pode inserir um byte de checagem no *frame* enviado.

Baudrate: O baudrate especificado é de 7812.5 bits/s.

Número Máximo de Nós: A Chrysler recomenda a interligação de até 16 módulos.

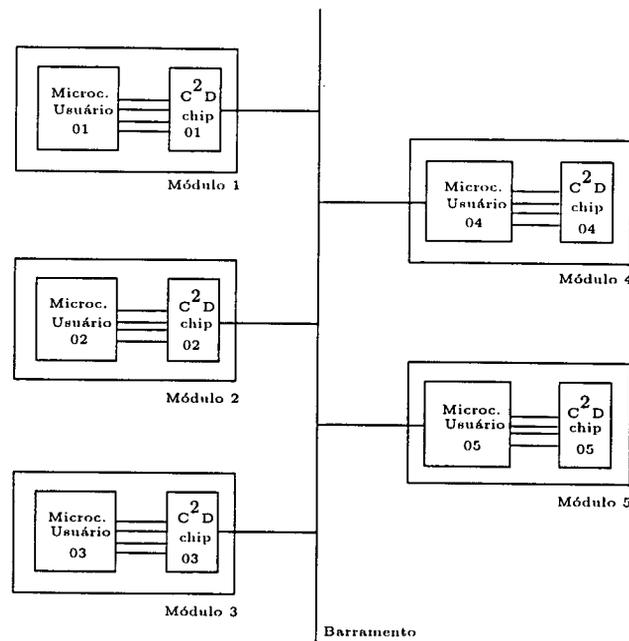
Meio Físico: Par trançado.

Resolução de Colisões: Para suportar uma arquitetura aberta em sistemas de comunicação de dados de classe B, o protocolo utiliza o CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*) que resolve uma colisão de dados sem destruí-los [SIA88]. Adicionalmente, o protocolo suporta ainda a priorização de mensagens que serve de auxílio ao sistema de arbitragem no caso de colisão.

4.6.3 Topologia da Rede

O protocolo utiliza a topologia tipo barramento, em que todos os nós estão conectados ao mesmo meio físico, conforme a figura 4.11.

Com a atribuição de uma prioridade a cada mensagem, é possível realizar a arbitragem garantindo sempre a seqüência de direito de envio à mensagem mais prioritária sem a perda de nenhum dado (no caso de mais de um nó detectar barramento desocupado ao mesmo tempo e necessitar enviar dados).

Figura 4.11: Topologia da Rede C²D

4.6.4 O Modelo de Referência OSI e o C²D

Os documentos que tratam do protocolo [FM89] [SAE91] [SIA88] não mencionam o modelo OSI, porém, pode-se considerar que as camadas 1 e 2 (física e de enlace de dados, respectivamente) podem ser implementadas através de *hardware*.

4.6.5 Formato das Mensagens

Os *frames* utilizados no protocolo C²D possuem o formato [FM89] [SAE91] apresentado na tabela 4.11⁹. A seguir, uma descrição dos campos.

ST	ID	SP	ST	Data	SP	ST	Err	SP	IF
1	8	1	1	8	1	1	8	1	10

Tabela 4.11: Formato do Frame C²D

ST: É o *start bit* ou delimitador de início do byte.

ID: É o campo identificador da mensagem, que inclui sua prioridade.

SP: É o *stop bit* ou delimitador de fim de byte.

Data: São os dados propriamente ditos. Como o *frame*, incluindo todos os campos (exceto o espaço interframes) está limitado no máximo a 6 bytes (um é utilizado pelo campo identificador e outro para checagem de erro), o campo de dados pode ter, no máximo, 4 bytes.

⁹Todos os valores apresentados na segunda linha são em bits

Err: Este campo é normalmente utilizado para a checagem de erro, que não é gerada automaticamente pelo protocolo, ficando sob a responsabilidade do software aplicativo.

IF: é o espaço necessário entre mensagens (*Inter-Message Separation*) e ocupa no mínimo 10 bits.

4.6.6 Método de Acesso ao Meio

O método de acesso ao barramento é o de arbitragem bit a bit não destrutivo, também utilizado em CAN, VAN e J1850 (ver seção 4.2.6).

4.6.7 Detecção de Erros

A detecção de erros fica por conta do usuário, que pode implementar a checagem de CRC se julgar necessário.

4.6.8 Disponibilidade de Hardware

Tendo sua utilização iniciada em 1988, o protocolo já dispõe de alguns fabricantes, dentre os quais pode-se citar GE/RCA e Motorola.

4.7 O Protocolo CSMA/DCR

4.7.1 Histórico

O CSMA/DCR *Carrier Sense Multiple Access with Deterministic Collision Resolution* é uma proposta desenvolvida pelo INRIA (*Institut National de Recherche en Informatique et en Automatique* - França) para a resolução dos problemas de indeterminismo ocasionados pela utilização do padrão IEEE 802.3 em aplicações de tempo real [LR93].

Estes problemas ocorrem devido ao fato dos protocolos que seguem este padrão apresentarem um comportamento probabilístico em relação ao atendimento das mensagens, situação indesejada para sistemas tempo real como é o caso da aviãoica.

O protocolo é apresentado como uma variação do IEEE 802.3 e é chamado 802.3D.

4.7.2 Características Gerais

Compatibilidade: O protocolo é completamente compatível com o padrão IEEE 802.3, atualmente o mais popular padrão utilizado em redes locais de arquitetura aberta.

Baudrate: O protocolo suporta até 10 Mbits/s [Bla89].

Codificação dos Bits: O CSMA/DCR utiliza a codificação em formato Manchester.

Meio Físico: Cabo coaxial blindado, par trançado ou fibra ótica [Car94].

Comprimento: O comprimento máximo suportado é 500 metros.

Número de Nós: Suporta até 1024 nós.

Prioridade: A prioridade é atribuída ao nó, e não à mensagem.

4.7.3 Topologia da Rede

A topologia da rede é o padrão utilizado para barramento *multimaster*.

4.7.4 O Modelo de Referência OSI e o CSMA/DCR

O protocolo CSMA/DCR especifica apenas a subcamada MAC da camada de enlace de dados OSI, porém, os produtos gerados a partir desta proposta são compatíveis com a camada física e de enlace de sistemas que utilizam padrões reconhecidos, como o ISO/OSI 8802 [Lan87].

4.7.5 Formato das Mensagens

Os *frames* utilizados no protocolo CSMA/DCR possuem formato apresentado na tabela 4.12 [Hal88]. A seguir, uma descrição dos campos.

Preâmb.	Início	Endereço Destino	Endereço Fonte	Comprim.	Dados	CRC
56 bits	8 bits	48 bits	48 bytes	16 bits	48-1500 bytes	32 bits

Tabela 4.12: Formato do Frame CSMA/DCR

Preâmbulo: é usado para a sincronização de *hardware*.

Início: é o campo delimitador para indicar início de um *frame* válido.

Endereço Destino: é o endereço da estação destino.

Endereço Fonte: é o endereço da estação transmissora.

Comprim: é o comprimento em *bytes* do campo de dados. Se este número for menor que 48, o campo de dados será automaticamente preenchido para atender ao requisito de tamanho mínimo do campo de dados.

Dados: são os dados propriamente ditos.

4.7.6 Método de Acesso ao Meio

A principal diferença entre o protocolo CSMA/DCR e a norma IEEE 802.3 (CSMA/CD) é justamente o acesso ao barramento. Enquanto no CSMA/CD este acesso é feito de forma probabilística, no CSMA/DCR, o determinismo é garantido através de uma técnica de busca em árvore binária balanceada, baseada na prioridade atribuída a cada estação.

A priori, cada estação conhece apenas:

- O *status* do barramento:
 - Livre
 - Ocupado com transmissão
 - Ocupado com colisão
- Seu próprio índice.

- O número total de índices definidos no sistema. Este número é denominado Q . Com este valor, é calculado o limitador de conjuntos de índices, q , que vem a ser a menor potência de 2 maior ou igual a Q . Assim, se existem 12 índices numa rede, então $Q = 12$ e $q = 16$ porque $2^4 = 16$ e $16 \geq Q$.

Se um ou mais nós que necessitam efetuar uma transmissão detectam o barramento vazio, o processo de transmissão é iniciado. No caso de mais de um nó fazê-lo, ocorre uma colisão.

Neste ponto, entra em ação o algoritmo de busca: detectada a colisão, cada uma das estações irá calcular suas chances de sucesso na próxima tentativa.

As estações envolvidas na colisão serão divididas em dois grupos: O das ganhadoras (W) e o das perdedoras (L). Cada estação identificará o grupo à qual pertence fazendo o seguinte cálculo:

- Grupo W - Índices entre $[0, \frac{q}{2}[$
- Grupo L - Índices entre $[\frac{q}{2}, q]$

Imediatamente após o cálculo, o novo valor de q para todas as estações será $\frac{q}{2}$.

Agora, as estações pertencentes ao grupo W podem tentar novamente a transmissão, sendo que as estações do grupo L devem desistir e aguardar até que ocorra uma transmissão de outro nó com sucesso seguida de detecção de barramento vazio.

A cada colisão, as estações são subdivididas novamente em grupos L e W.

Caso o conjunto de ganhadores fique vazio, a iteração deve ser revertida, isto é, o valor de q retorna ao número anterior e a subdivisão de grupos se dá a partir dos perdedores. A nova subdivisão em grupos é:

- Grupo W - Índices entre $[\frac{q}{2}, \frac{3q}{4}[$
- Grupo L - Índices entre $[\frac{3q}{4}, q]$

O período que todas as estações relacionadas a uma colisão levam para serem atendidas (isto é, todas transmitem) é denominado *época*.

Até a finalização de uma *época*, nenhum novo nó pode integrar o conjunto de nós inicial.

Para melhor clareza no entendimento do processo, é apresentado um exemplo.

Para uma rede com 15 estações em que 6 delas envolvem-se em uma colisão (índices 2, 3, 5, 12, 14 e 15), a solução desenrola-se da seguinte maneira:

1. Situação inicial: colisão entre nós de índices 2, 3, 5, 12, 14 e 15.

2. Cálculo de q : Para $Q = 15$, $q = 16$.

Grupo de Ganhadores: $W = 2, 3$ e 5 (índices entre $[0, 8[$)

Grupo de Perdedores: $L = 12, 14$ e 15 (índices entre $[8, 16]$)

Nova colisão envolvendo 2, 3 e 5.

Novo $q = 8$

3. Nova iteração:

Grupo de ganhadores: $W = 2$ e 3 (índices entre $[0, 4[$)

Grupo de Perdedores: $L = 5$ (índices entre $[4, 8]$)

Novo $q = 4$

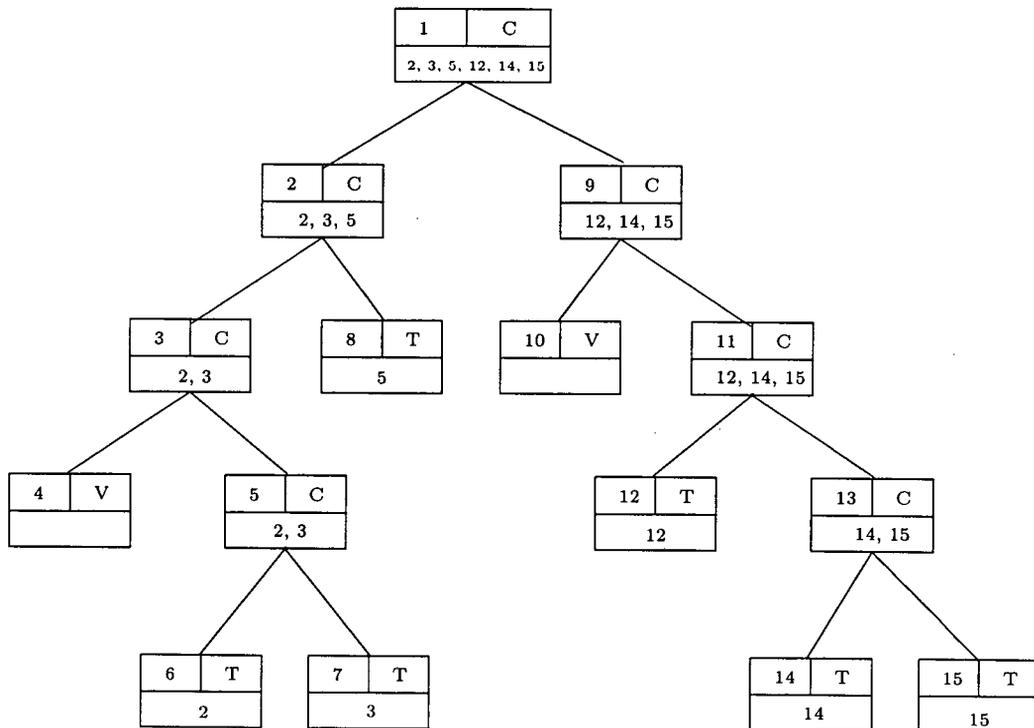
Nova colisão envolvendo 2 e 3.

4. Nova iteração:
 Grupo de ganhadores: $W = \emptyset$ (índices entre $[0, 2[$)
 Grupo de Perdedores: $L = 2$ e 3 (índices entre $[2, 4]$)
5. Como o conjunto de ganhadores é vazio, o conjunto dos perdedores passa a competir pelo barramento. Assim, ocorre nova colisão entre os índices 2 e 3. O q agora vale 4.
6. Nova iteração.
 Grupo de ganhadores: $W = 2$ (índices entre $[2, 3[$)
 Grupo de Perdedores: $L = 3$ (índices entre $[3, 4]$)
 A estação de índice 2 consegue realizar a transmissão sem colisão.
7. Como só existe um índice no conjunto dos perdedores, este também transmite sem colisão, logo em seguida. Agora, o q passa a valer 8.
8. Nova iteração:
 Grupo de ganhadores: $W = 5$ (índices entre $[4, 6[$)
 Grupo de Perdedores: $L = \emptyset$ (índices entre $[6, 8]$)
 Como o grupo de ganhadores possui apenas um índice, esta estação realiza a transmissão com sucesso e o processo continua no sentido reverso. Faz-se novamente $q = 16$ e nova subdivisão de grupos.
9. Nova iteração:
 Grupo de ganhadores: $W = \emptyset$ (índices entre $[8, 12[$)
 Grupo de Perdedores: $L = 12, 14$ e 15 (índices entre $[12, 16]$)
10. Como o conjunto de ganhadores é vazio, o conjunto dos perdedores passa a competir pelo barramento. Assim, ocorre nova colisão entre os índices 12, 14 e 15. A iteração é revertida e $q = 16$.
11. A situação continua a mesma da iteração anterior. Neste caso, para resolver o impasse, os intervalos entre ganhadores e perdedores trocam os intervalos de pertinência.
12. Nova iteração:
 Grupo de ganhadores: $W = 12$ (índices entre $[8, 12]$)
 Grupo de Perdedores: $L = 14$ e 15 (índices entre $]12, 16[$)
 A estação de índice 12 consegue realizar a transmissão sem colisão.
13. As estações com índices 14 e 15 disputam o barramento e colidem.
14. Nova iteração:
 Grupo dos Ganhadores: $W = 14$ (índices entre $[13, 14]$)
 Grupo dos Perdedores: $L = 15$ (índices entre $[15, 16]$)
 A estação de índice 14 consegue realizar a transmissão sem colisão.
15. A estação de índice 15 consegue realizar a transmissão sem colisão, finalizando a *época*.

A figura 4.12 resume os passos descritos anteriormente.

4.7.7 Detecção de Erros

A detecção de erros é feita através do cálculo do CRC com 32 bits.



n	status
índices	

n - número do passo
 status - status do barramento: Colisão, Vazio ou Transmitindo
 índices - índices envolvidos na transação

Figura 4.12: Resolução de uma Época no CSMA/DCR

4.7.8 Disponibilidade de Hardware

O protocolo dispõe de produtos para aplicação nas mais variadas áreas.

Como exemplo de aplicação, pode-se citar o sistema de combate do porta-aviões *Charles De Gaulle* que utiliza a rede Recital da indústria *Dassault Electronique* baseada neste protocolo [Car94].

A empresa Aptor (França), utiliza o CSMA/DCR em sua rede Factor para aplicações industriais e a Intel fabrica o chip 82596 com a mesma finalidade.

4.8 O Protocolo FIP

4.8.1 Histórico

O FIP (*Factory Instrumentation Protocol*) é a proposta francesa do *Fieldbus*, que visa substituir as ligações ponto-a-ponto analógicas de sensores/atuadores por um barramento de sinal digital, bidirecional e de acesso compartilhado [oA91].

O protocolo foi elaborado em conjunto por empresas européias (*Telemecanique*, *CEGELEC* e *CGEE Alstom*), órgãos do governo francês e centros de pesquisa, que fundaram, em 1986, o chamado Club FIP [Cat89].

Em 1989, fabricantes, usuários e profissionais envolvidos com o uso e desenvolvimento deste protocolo fundaram o *WorldFIP*, uma entidade que possuía, até 1993, cerca de 120 empresas membros na Europa e 140 nos Estados Unidos [wfi93].

O protocolo FIP baseia-se na difusão de variáveis, ou seja, as mensagens não são endereçadas a estações específicas. As informações são disponibilizadas no barramento através de *broadcast* e todas as estações que tiverem interesse podem acessar [Ste95].

4.8.2 Características Gerais

O protocolo FIP possui as seguintes características:

Baudrate: Para transmissões utilizando como meio físico o par trançado, estão previstas três velocidades de transmissão:

- V1 : 31.25 Kbits/s para distâncias até 2000 metros.
- V2 : 1 Mbits/s para distâncias até 500 metros com blindagem.
- V3 : 2.5 Mbits/s para distâncias até 200 metros com blindagem.

A velocidade padrão é a V2, sendo as outras duas reservadas para aplicações especiais.

Para aplicações com fibra ótica, a velocidade de transmissão chega a 5 Mbits/s.

Comprimento: Uma rede FIP pode suportar distâncias de até 2000 metros.

Número de Nós: O número máximo de nós chega a 256.

Limite de Tipos de Variáveis: O FIP suporta a identificação de até 24000 variáveis ou dados.

Codificação dos Bits: O FIP utiliza o formato Manchester, que permite o envio simultâneo do sinal de sincronização e os dados propriamente ditos.

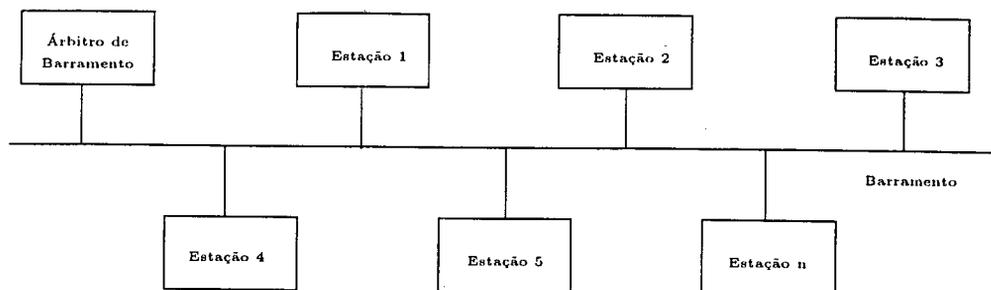


Figura 4.13: Topologia da Rede FIP

4.8.3 Topologia da Rede

A topologia da rede FIP é mostrada na figura 4.13

4.8.4 O Modelo de Referência OSI e o FIP

A camada física deste protocolo teve sua concepção baseada na norma MIL-STD-1553 (apresentada na seção 4.5) [Sch89].

As camadas de enlace de dados não apresenta uma separação formal entre as sub-camadas LLC e MAC, como proposto na norma IEEE 802, devido às características próprias de sua filosofia de acesso ao meio, apresentada na seção 4.8.6.

A camada de aplicação proposta pelo FIP define serviços de mensagem industrial que não interferem com o tráfego de tempo real.

As outras camadas OSI não são definidas pelo protocolo.

4.8.5 Formato das Mensagens

As informações que circulam pelo barramento são classificadas em dois tipos [dA89] [Com87], sendo que o Frame Identificador é sempre enviado pelo Árbitro de Barramento e o Frame de Resposta, transmitido por uma estação produtora. Este último ainda pode se subdividir em três tipos, conforme descrição a seguir:

- Frame Identificador
- Frame de Resposta
 - Dados
 - Reconhecimento
 - Requisição

No caso da aplicação estudada, os *frames* de resposta serão sempre do tipo dados.

Todos os *frames* possuem o formato mostrado na tabela 4.13¹⁰, diferenciando-se um dos outros apenas pelo conteúdo de tamanho dos campos de controle e de dados.

PRE: é o campo de preâmbulo, usado para a sincronização entre emissor e receptores.

¹⁰A unidade de tamanho utilizada na tabela é bit

PRE	FSD	EB	Controle	Dados	FCS	EB	FED	EB
8	5	1	6	n * 16	16	1	5	1

Tabela 4.13: Formato dos Frames do Protocolo FIP

FSD: é o *Frame Start Delimiter* ou delimitador de início de *frame*.

EB: é o bit de equalização, que opera como bit de interface entre os delimitadores e os dados codificados em Manchester II.

Controle: este campo é diferenciado para cada tipo de *frame*. Assim, se é um *frame* de resposta, o conteúdo deste campo contém o tipo de resposta que se está esperando. Se é um *frame* de requisição, pode conter o endereço destino.

Dados: este campo também é diferenciado para cada tipo de *frame*, podendo conter tanto uma lista de identificadores como dados propriamente ditos. O limite máximo para este campo é de 1024 bits, ou seja, n não pode ser maior que 64.

FCS: é campo para o controle de erros (*Frame Checksum*) que utiliza o método de cálculo de CRC adotando o polinômio gerador proposto pela CCITT.

FEB: é *Frame End Delimiter* ou delimitador de final de *frame*.

4.8.6 Método de Acesso ao Meio

O protocolo FIP está baseado em uma arquitetura mestre-escravo, onde um nó mestre (Árbitro de Barramento) controla a transmissão e recepção dos outros nós escravos (Produtores/Consumidores de Dados).

São definidos dois tipos de terminais para o barramento:

- AB - Árbitro de Barramento
- PC - Produtor/Consumidor

O controle de acesso ao barramento é realizado por uma estação denominada **Árbitro de Barramento** [Sal94]. Esta estação transmite os nomes dos objetos/variáveis que deverão ser atualizadas e a estação responsável envia os valores disponíveis, conforme esquema mostrado na figura 4.14.

A operação se dá da seguinte forma:

1. O árbitro difunde o nome da variável/dado a ser atualizado.
2. O produtor da variável difunde o dado.
3. Todos os consumidores interessados acessam o dado.

O árbitro atualiza as variáveis segundo uma lista executada ciclicamente.

4.8.7 Detecção de Erros

O FIP realiza a detecção de erros através do cálculo de CRC utilizando o polinômio CCITT apresentado na equação 4.4.

$$x^{16} + x^{12} + x^5 + 1 \quad (4.4)$$

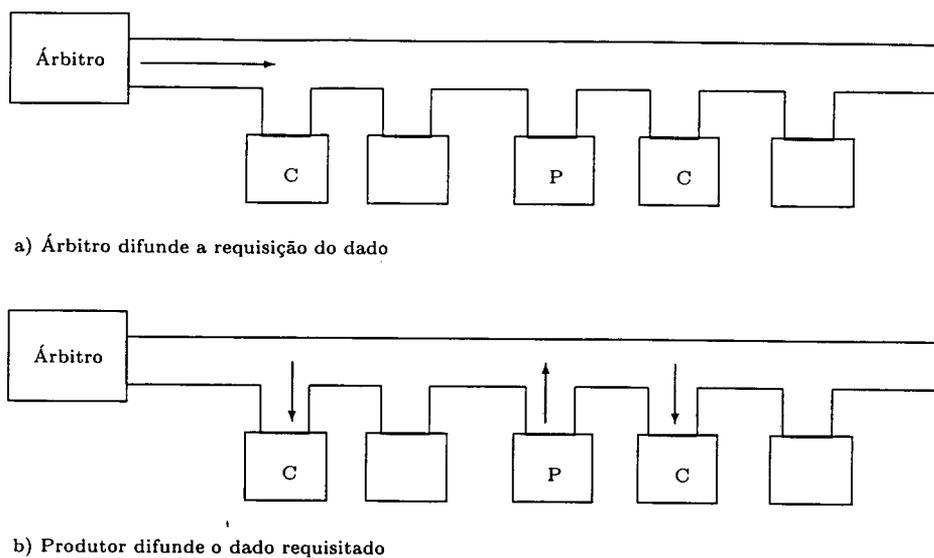


Figura 4.14: Método de Acesso ao Meio no FIP

4.8.8 Disponibilidade de Hardware

O protocolo FIP possui grande variedade de produtos compatíveis, dentre os quais podemos citar [cat92]:

- Controladores
- Placas padronizadas (para PCs, aquisição de dados, interfaceamento, etc...)
- Interfaces remotas para equipamentos diversos
- Sensores
- Atuadores
- Acessórios
- Produtos específicos para aplicações especiais

No tocante ao aspecto referente à interoperabilidade, esta está garantida apenas para produtos certificados pelo ClubFIP, que publica as Notas Técnicas utilizadas para avaliação do produto.

O protocolo FIP possui ainda três famílias de processadores para sua implementação: a FIPART, a FULLFIP e a FIPIU [LV92].

O chip FIPART é uma espécie de co-processador que implementa as camadas física e de enlace de dados.

O chip FULLFIP é o mais completo dos circuitos FIP. Cobre as camadas física, de enlace de dados e de aplicação. É baseado num processador Motorola 68000. Possui ainda uma gama de ferramentas de software para sua implementação [Val92].

O chip PIPIU é o mais recente da família e cobre as camadas física e de enlace de dados, suportando um baudrate de até 5 Mb/s.

4.9 Conclusão

Neste capítulo, foram descritas as características básicas, consideradas como fatores de distinção do ponto de vista da aplicação, para cada um dos protocolos escolhidos para simulação.

Os aspectos aqui considerados foram o histórico, características gerais, topologia, camadas do modelo OSI contempladas, tipos e estruturas dos frames, métodos de acesso ao barramento, detecção de erros e disponibilidade de *hardware*.

Estes aspectos são apresentados de forma resumida na tabela 4.14.

Prot.	Afil.	Aplic.	Transm.	Acesso	Erro	Baud.	Camadas OSI	Disp. Hardw.
CAN	ISO Bosch	Auto Intr.V.	Par Tranç. Fibra Ótica	contenção	CRC	1 Mb/s	1, 2 e 7	sim
VAN	Renault	Auto Intr.V.	Par Tranç. Fibra Ótica	contenção	CRC	100 kb/s	1 e 2	não
J1850	SAE	Auto Intr.V.	Par Tranç. Único Fio	contenção	CRC opc.	10 kb/s 40 kb/s	1 a 7	não
MIL-STD 1553B	Padrão Militar	Aviônica	Par Tranç. Fibra Ótica	Mestre/ Escravo	Parid.	1 Mb/s	1 e 2	sim
C ² D	Chrysler Bosch	Auto Intr.V.	Par Tranç. Fibra Ótica	contenção	CRC opc.	7812.5 b/s	1 e 2	sim
CSMA/ DCR	Inria	Redes Locais	Par Tranç.	contenção	CRC	10 Mb/s	1 e 2	sim
FIP	ClubFIP	Indústria	Par Tranç.	Mestre/ Escravo	CRC	1 Mb/s 2.5 Mb/s	1, 2 e 7	sim

Tabela 4.14: Resumo das Características dos Protocolos Escolhidos para Simulação

Capítulo 5

Comparação de Desempenho dos Protocolos por Simulação

5.1 Introdução

Para melhor avaliar a adequação dos protocolos vistos no capítulo anterior à aplicação em questão, faz-se necessária uma análise comparativa de desempenho dos mesmos. Devido à indisponibilidade de placas de rede e *softwares* correspondentes para todos os protocolos citados, optou-se por realizar uma simulação computacional dos mesmos.

O objetivo desta avaliação é verificar se os protocolos escolhidos conseguem atender os requisitos da aplicação com relação aos *deadlines*. Com estes resultados, pode-se eliminar os protocolos não adequados e fazer uma análise mais abrangente em relação aos restantes, considerando-se então, a facilidade de implementação, robustez, suporte técnico, histórico de utilização, entre outros.

Como critério de atendimento aos *deadlines*, mediu-se, para cada protocolo, o tempo decorrido desde a criação de um dado no modelo até a sua chegada à estação destino. Este tempo deve ser menor que o *deadline* do dado em questão e ainda deve-se levar em consideração uma folga, já que aqui são modelados apenas a camada física e a subcamada de acesso ao meio. Na implementação real, onde a camada de aplicação e a subcamada LLC (*Logical Link Control*) também consumirão algum tempo para execução, este tempo deverá ser maior.

Neste capítulo, descreve-se o processo de simulação da rede intra-veículo desde a construção do modelo até a adaptação deste para os vários protocolos.

5.2 Ferramenta

A ferramenta utilizada na simulação foi o software ARENA 1.24, que é a versão com animação do programa SIMAN [Sys89] [dT80].

Optou-se pela escolha deste *software* pela sua disponibilidade e também devido ao fato de que, sendo ele concebido para modelagem e simulação de sistemas a eventos discretos, não se tem notícia de sua utilização para simulação de protocolos de comunicação. Com este trabalho, pretende-se também verificar sua adequação e facilidade de uso em aplicações deste tipo.

Este software foi desenvolvido com o objetivo de permitir a simulação e animação de modelos para as mais variadas aplicações. Possui uso conhecido em simulação de linhas de produção, células flexíveis de manufatura, serviços (hospitais, restaurantes, lanchonetes, lojas, etc. . .), engenharia de tráfego, entre

outras. Suas principais características são:

- Ambiente gráfico de programação
- Programação orientada a objeto
- Inclui painéis para desenhar, animar, executar e construir o modelo a ser simulado.

5.2.1 Conceitos

Alguns conceitos utilizados na descrição do processo de modelagem necessitam ser definidos. São eles:

Entidades: representam qualquer pessoa ou objeto (real ou imaginário) que se movimenta pelo sistema alterando o estado deste. Na modelagem aqui descrita, as entidades representam *frames* ou conjuntos de *frames*.

Atributos: são características próprias de cada entidade. Assim, a cada entidade gerada, são associados atributos de forma individual. No modelo ora estudado, cada entidade possui como atributos: destino, *deadline*, sentido de deslocamento, velocidade, tempo e prioridade/índice.

Recursos: representam os meios a serem utilizados pelas entidades para serem transportadas de um local para outro ou sofrerem transformações. Neste modelo, o recurso que todas as entidades precisam acessar é o barramento.

Filas: representam o local de espera para entidades que não podem acessar o meio requerido por algum impedimento funcional.

Estações: representam o processamento em áreas do sistema que estão sendo modeladas. Uma estação pode ser utilizada para gerar entidades, transformá-las ou recebê-las. No modelo utilizado, um grupo de estações representa um nó na rede.

É importante observar que a palavra estação no contexto do ARENA possui significado diverso ao conceito estabelecido para a palavra estação no contexto de uma rede informática. Neste capítulo, para evitar dúvidas interpretações, as estações correspondentes ao conceito utilizado em redes serão chamadas nós.

5.2.2 Módulos Lógicos

O ARENA utiliza o conceito de orientação a objeto, fazendo com que seus módulos sejam definidos por blocos representados graficamente. Blocos se relacionam entre si através de ligações (também representadas graficamente) ou pela sua estrutura funcional. Para o trabalho de modelagem, foram utilizados os seguintes blocos:

ARRIVE: Este bloco representa a criação de entidades e a entrada destas no sistema através de uma estação.

QUEUE: Representação funcional das filas de espera.

QUEUES: Declaração de todas as filas existentes no modelo.

SCAN: Possui a funcionalidade de um filtro que permite a passagem da entidade sob determinadas condições.

ACCESS: Permite a aquisição de um recurso por uma variável.

BRANCH: Possui a capacidade de alterar o caminho a ser seguido pela entidade sob determinadas condições.

CONVEY: É o transportador propriamente dito. Possui a capacidade de transportar/conduzir entidades de uma estação para outra.

CONVEYORS: Declaração de todos os transportadores existentes no modelo.

SEGMENT: Define todas as seqüências possíveis de caminhos que uma entidade pode percorrer. É representada por uma lista de estações e das distâncias entre elas. Cada seqüência só pode ser percorrida em um único sentido.

STATION: No modelo, são as estações destino das entidades, já que as estações de origem já estão implicitamente declaradas no bloco ARRIVE.

EXIT: Este bloco libera um recurso utilizado por uma entidade.

TALLY: Este bloco contabiliza o espaço de tempo decorrido desde a criação da entidade até o presente momento.

DISPOSE: Este bloco destrói as entidades que chegam até ele.

SIMULATE: Este bloco define as características básicas do experimento como o tempo de simulação.

5.3 A Topologia da Rede Modelada para a Aplicação

Os protocolos foram modelados a partir da topologia de rede mostrada na figura 5.1.

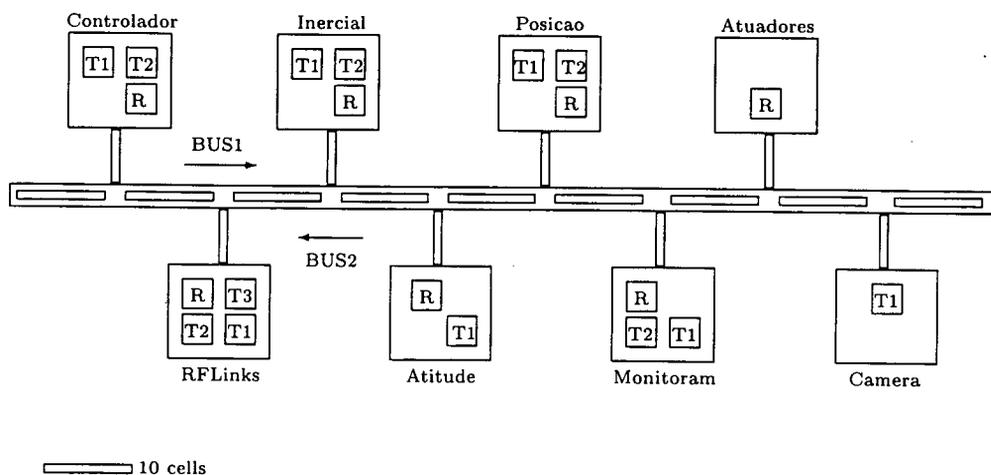


Figura 5.1: Topologia da Rede Modelada

Os nós possuem uma estação transmissora para cada tipo de dado e uma estação receptora para recebê-los. Um conjunto de estações pode ser representado como um único nó virtual definindo as distâncias entre elas com valor zero no bloco SEGMENT.

Como uma seqüência só pode ser percorrida obedecendo estritamente a ordem em que foram descritas as estações e os dados fluem na rede nos dois sentidos, foram definidas duas seqüências de estações, que percorrem o mesmo caminho em sentidos opostos.

Para possibilitar o transporte bidirecional de dados, foram criados também dois transportadores que representam o barramento. Cada um percorre uma das seqüências definidas anteriormente.

Os dados foram agrupados de acordo com suas características básicas [GYR96] e receberam os atributos adequados, de acordo com a proposta de arquitetura apresentada na figura 2.5 integrante do capítulo 2.

Nó de Origem	Nó de Destino	Dado Tipo	Dados [bits]	Deadl [ms]	Prior	Definição
T1Control	RAtuad	servos	8 x 16	25	0	sinais PWM p/servos
T2Control	RRFLinks	comm	24 x 48	1000	11	telemetria
T1Inercial	RControl	w	2 x 32	50	2	veloc. angulares
T2Inercial	RControl	acel	3 x 16	50	1	acel. angulares
T1Atitude	RControl	atit	2 x 32	100	5	ângulos
T1Posição	RControl	xyz	3 x 48	1000	9	latitude, longit. e altitude
T2Posicao	RControl	vel	3 x 16	1000	7	veloc. lineares
T1RFLinks	RControl	refs	4 x 48	50	3	referências
T2RFLinks	RControl	opcode	3 x 48	200	6	referências
T3RFLinks	RControl	offset	4 x 48	1000	10	referências
T1Monit	RControl	rpm	1 x 16	100	4	rotação no motor
T2Monit	RControl	battemp	3 x 16	1000	8	carga da bateria e temp. no motor
T1Camera	RControl	video	65536	1000	12	signal digital da imagem

Tabela 5.1: Tabela de Dados/Características

Com relação aos dados apresentados na tabela 5.1, cabe fazer algumas observações.

1. Todos os nós que enviam dados para o Controlador também enviam para o nó RFLinks (Enlaces de Rádio-Frequência) que os enviará para a Estação de Controle em Terra. Como algumas mensagens são entregues simultaneamente a vários nós através de *multicasting*, foi considerado apenas o tempo de deslocamento até o nó mais distante no modelo, que é o Controlador.
2. Em CAN, cada *frame* deve ter uma prioridade única. Na modelagem, como os dados de um determinado tipo são gerados simultaneamente, optou-se por alocar um nível de prioridade para cada tipo de dado. As prioridades serão redefinidas para resolver a questão da definição de índices para o protocolo CSMA/DCR, já que este determina um índice para cada estação, e não cada *frame*. Nos outros modelos, a prioridade é diretamente associada à mensagem, seja na ordem em que ela é chamada pelo supervisor, como nos casos do MIL-STD-1553B e o FIP, seja na estrutura do *frame*, como no C²D, VAN e J1850.
3. Os níveis de prioridade foram distribuídos segundo os critérios de *deadline* mais baixo para a maior prioridade (*rate-monotonic*). Em caso de coincidência de valores, optou-se pelo bloco de dados de menor tamanho.
4. No caso da câmera de vídeo, seus dados obtiveram o mais baixo índice de prioridade tanto pelo volume de dados que é consideravelmente maior que os demais como pelo fato da ocorrência de um atraso em um de seus *frames* não ser de natureza crítica para o controle da aeronave, podendo-se relaxar a rigidez do atendimento de *deadlines* com relação aos demais dados.

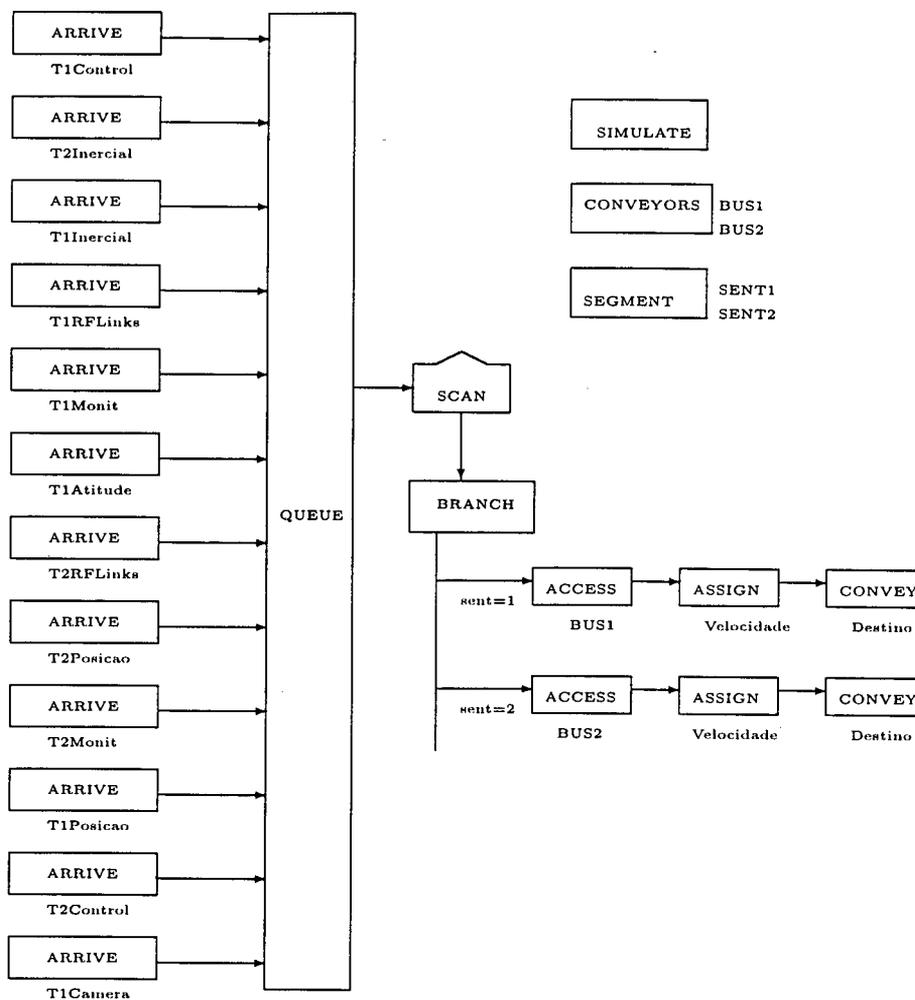


Figura 5.2: Geração de Entidades no ARENA

5.4 Modelagem da Rede no ARENA

5.4.1 Modelagem da Criação de Dados no Sistema

A figura 5.2 mostra a primeira parte do diagrama de blocos do sistema, referente à criação de entidades tal como foi modelado no ARENA.

Conforme mostra a figura 5.2, cada estação geradora de frames está associada a um bloco **ARRIVE**. Este bloco aceita como parâmetros:

- Nome da estação associada à geração de entidades (no desenho, o nome da estação aparece embaixo do bloco).
- Quantos frames serão criados simultaneamente a cada intervalo de tempo pré-definido, sendo que, para cada estação, foi utilizada a informação de frames gerados que constam nas tabelas de dados para cada protocolo (vide tabelas 5.2, 5.4, 5.5, 5.8, 5.11, 5.13, 5.15 e 5.17).
- Tempo inicial para o início da primeira criação. No modelo, todos os blocos iniciam a criação no tempo zero. Como o programa é sequencial, os blocos **ARRIVE** foram desenhados de acordo com

a prioridade dos dados gerados (em ordem crescente).

- Intervalo entre as criações. Foi utilizado o período de cada bloco.
- Atributos de cada entidade gerada. Foram listados os atributos *deadline*, *sentido*, *velocidade*, *prioridade*, *destino* e *tempo* (este último atualizado automaticamente pelo programa e utilizado para saber o tempo de transporte da entidade, desde a sua criação até o seu destino).

Todos os dados/entidades geradas são armazenados em uma fila para posterior transporte. Esta fila, representada pelo bloco **QUEUE**, é organizada baseando-se no atributo prioridade. Seguindo a ordem LVF ¹, este bloco coloca no topo da fila os valores de mais alta prioridade. Isto faz com que esteja garantida a premissa básica dos protocolos cuja prioridade está associada à mensagem: os *frames* de maior prioridade sempre serão enviados antes, de forma determinista. No caso do CSMA/DCR, em que a prioridade é associada ao nó, todas as estações integrantes de um nó serão adaptadas para terem seus dados com a mesma prioridade.

Para a resolução do problema de acesso ao barramento, o modelo utiliza o bloco **SCAN**, que testa se existe alguma entidade utilizando algum dos dois recursos BUS1 e BUS2. Desta forma, uma entidade só pode sair da fila e ser transportada se o barramento encontra-se completamente vazio. No caso de protocolos onde é possível a ocorrência de colisões, estas são simuladas na forma de um atraso correspondente ao tempo de resolução dos conflitos (este atraso já é conhecido, no caso dos protocolos aqui estudados).

O bloco **BRANCH** é utilizado para definir qual **CONVEYOR** a entidade vai utilizar para fazer o transporte: BUS1 ou BUS2. Esta decisão é tomada consultando-se o atributo *sentido* da entidade.

O bloco **ACCESS** realiza o acesso ao recurso e o bloco **ASSIGN** é utilizado para alterar a velocidade do transportador de acordo com o atributo *velocidade* de cada entidade.

A seguir, o bloco **CONVEYOR** transporta a entidade até o seu destino (consultando seu atributo *destino*, naturalmente) e utilizando o segmento correspondente.

Os blocos **CONVEYORS** e **SEGMENT** apenas declaram os transportadores (BUS1 e BUS2) e seus segmentos correspondentes (*sentido1* e *sentido2*).

O bloco **SIMULATE** define o intervalo de tempo que será simulado.

5.4.2 Modelagem da Chegada dos Dados às Estações Destino

A figura 5.3 mostra a segunda parte do diagrama de blocos do sistema, referente à chegada das entidades às estações destino, bem como os blocos de medição dos tempos de ocupação do barramento.

A figura 5.2 descreve o trajeto das entidades até o barramento, representado pelos blocos **CONVEY**. Este bloco transporta o dado até as estações destino, que, na figura 5.3 são representadas pelos blocos **STATION**.

Após chegar a seus destinos, as entidades liberam o barramento para utilização (bloco **EXIT**) e são destruídas (bloco **DISPOSE**).

O bloco **TALLY** atualiza as variáveis *tempo* de cada entidade que chega. Como esta variável foi inicializada na criação da entidade, pode-se obter o tempo total que o frame levou desde a sua criação até sua chegada à estação destino. Nesta simulação, as variáveis usadas para a estatística são formadas pela letra *t* de tempo e a abreviatura do *deadline* correspondente. Como exemplo, podemos citar o caso das entidades que chegam à estação **RAtuadores**, que possuem todas *deadline* igual a 25 ms e têm seus tempos de chegada registrados nas variáveis tipo *tDead25*.

¹LVF - Low Value First

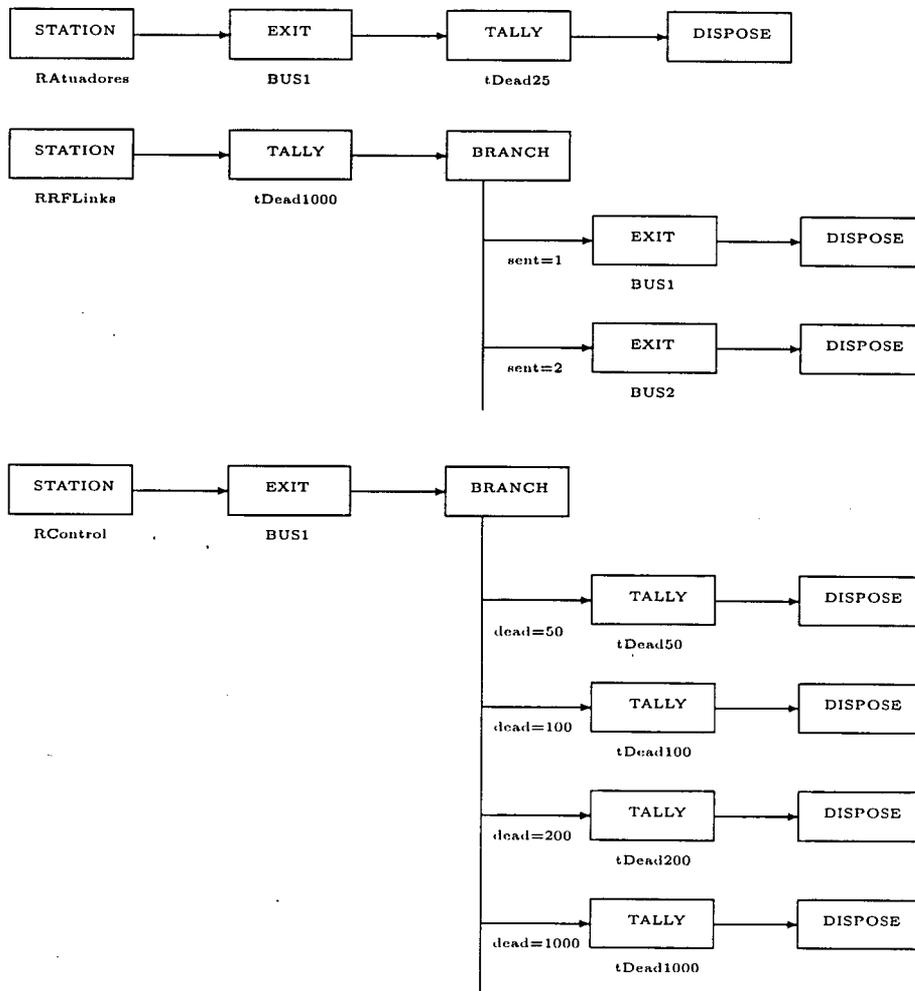


Figura 5.3: Chegada das Entidades ao Destino

Como a estação **RRFLink** recebe dados provenientes da estação **Controlador** e da estação **Câmera** que fluem em sentidos opostos, é necessário fazer um teste para definir qual dos recursos será liberado: **BUS1** ou **BUS2**.

No caso da estação **RControlador**, o teste é feito para agrupar os tempos de chegada em suas variáveis correspondentes.

O sistema em estudo pode ser considerado pela teoria da simulação como sendo do tipo terminal [PSS90] com tempo de *warm-up* equivalente ao período máximo de chegada de mensagens, no caso, 1 segundo. Após este intervalo, o sistema apresenta um comportamento periódico e determinístico (o intervalo de chegada de novas mensagens na fila é igual ao período de amostragem para cada grandeza). Em todas as simulações realizou-se um tempo muito superior ao de *warm-up*, de forma que este pôde ser desprezado. Na maioria dos casos (ou seja, quando o programa de simulação assim o permitiu), optou-se por um tempo de simulação de 15 minutos cada (para cada protocolo, foram realizadas cerca de 10 simulações — os resultados obtidos correspondem à média), que corresponde ao valor aproximado ao tempo de aquecimento do motor da aeronave.

5.5 Modelo para o Protocolo CAN

O protocolo CAN utilizou o modelo acima descrito por gerar frames de dados periodicamente e atender aos requisitos de atendimento preferencial aos frames mais prioritários.

Para o trabalho de modelagem, a tarefa inicial referiu-se ao encapsulamento dos dados no formato descrito no protocolo.

5.5.1 Cálculo dos Tempos de Ocupação do Meio

Na ferramenta de simulação utilizada, cada frame é uma entidade. Para simular o tempo em que o bloco está ocupando o barramento, ou tempo de propagação, são utilizadas variáveis simbólicas que representam a velocidade com que a entidade se desloca no modelo (em células/ms) e a distância entre as estações (em células).

Os dados de cada tipo foram agrupados em blocos e divididos em frames, respeitando o limite de 8 bytes de dados por frame.

Para fins de simulação, considerou-se o tempo em que um frame ocupa o barramento como sendo o seu tempo de propagação², utilizando-se a equação 5.1.

Na figura 5.1 estão representadas as distâncias entre as estações nesta unidade. Assim, para cada bloco descrito, foram calculados os parâmetros necessários à simulação, representados na tabela 5.2³.

$$T_{propagação} = \frac{tamanho + 55}{baudrate} \quad (5.1)$$

Os termos da equação possuem o seguinte significado:

Tpropagação: é o tempo de propagação do frame completo em milissegundos (ms).

tamanho: é o número total de bits que compõe o campo de dados.

55: é o número de bits adicionados em cada frame de dados devido aos campos e ao espaço interframes.

baudrate: é a taxa de transmissão em bits por milissegundo⁴.

²Optou-se pela utilização das unidades de tempo em milissegundos para facilitar a representação numérica

³Os dados na câmera não aparecem na tabela por terem que ser submetidos a um tratamento especial, descrito mais adiante

⁴No padrão CAN, o baudrate máximo é 1 Mbit/s ou 1000 bit/ms

5.5.2 Dados Necessários à Simulação

A tabela 5.2 lista o valores que foram necessários para se construir o modelo para simulação do protocolo CAN.

Dado Tipo	Tamanho [bits]	Dados/Frame	Frames	Tprop. [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	64	2	0.119	60	504.20
comm	1152	64	18	0.119	10	84.03
w	64	64	1	0.119	50	420.17
acel	48	48	1	0.103	20	194.17
atit	64	64	1	0.119	30	252.10
xyz	144	48	3	0.103	40	388.35
vel	48	48	1	0.103	40	388.35
refs	192	64	3	0.119	10	84.03
opcode	144	48	3	0.103	10	97.09
offset	192	64	3	0.119	10	84.03
rpm	16	16	1	0.071	50	704.23
battemp	48	48	1	0.103	50	485.44

Tabela 5.2: Tabela de Dados Necessários à Simulação CAN

Na modelagem utilizada, considerou-se, somente no caso da câmera de vídeo, por apresentar um volume de dados muito grande, a divisão do bloco de dados em pacotes de frames.

A necessidade de encapsulamento de dados em pacotes de *frames* não interrompíveis deveu-se ao fato de que o protocolo CAN admite um máximo de 8 bytes de dados por frame, necessitando-se gerar a quantidade de 1024 frames por segundo na modelização. O grande volume de entidades geradas se cada *frame* for tratado separadamente impossibilita o *software* de administrar matrizes desta ordem de grandeza, já que a versão do ARENA disponível restringe o número total de frames no sistema em 150.

Para fins de simulação, considerou-se o tempo em que um pacote ocupa o barramento como sendo o seu tempo de propagação, dado pela equação 5.2.

$$T_{prop.} = \frac{\text{pacote} + (nframes * 55)}{\text{baudrate}} \quad (5.2)$$

Os termos da equação possuem o seguinte significado:

Tprop: é o tempo de propagação do pacote completo em milisegundos (ms).

pacote: é o número total de bits que compõe um pacote de dados de um determinado tipo.

nframes: é o número de frames utilizado para encapsular o pacote (observar que cada frame suporta um máximo de 64 bits de dados).

55: é o número de bits adicionados em cada frame de dados devido aos campos e o espaço interframes.

baudrate: é a taxa de transmissão em bits por milisegundo.

Para a câmera de vídeo que gera 64Kbytes por segundo, seus dados ficaram divididos em 64 pacotes, tendo cada um 16 frames de dados, cada frame com um campo de dados de 64 bits (que é o tamanho máximo de bits de dados por *frame* suportado pelo protocolo).

O tempo de propagação de cada pacote (cada pacote representa uma entidade) é de 1.904 ms. Como a distância entre a estação que representa a câmera (Câmera) da estação que representa os enlaces de

rádio frequência (RFLinks) 5.1 é de 60 células, a velocidade que cada entidade utiliza na simulação para percorrer seu trajeto é de 31.51 cells/ms.

5.5.3 Resultados Obtidos na Simulação

A tabela 5.3 mostra a média dos resultados obtidos em dez simulações com um intervalo de tempo de 15 minutos (900.000 ms) cada.

Por questões de simplificação do modelo (o ARENA restringe o número máximo de blocos utilizado na construção do modelo), os dados foram agrupados de acordo com o seu *deadline* que aqui coincide com o seu período, já que todos os dados são periódicos. Os grupos são relacionados a seguir (baseados na tabela 5.1):

Deadline = 25 ms. São os sinais tipo servo.

Deadline = 50 ms. São os sinais tipo w, refs e acel.

Deadline = 100 ms. São os sinais tipo rpm.

Deadline = 200 ms. São os sinais tipo opcode.

Deadline = 1000 ms. São os sinais tipo comm, xyz, vel, offset, battemp e video.

O total de entidades de cada tipo produzidas no intervalo de simulação está representado na última coluna.

Deadline [ms]	Tmédio [ms]	Tvariação [ms]	Tmínimo [ms]	Tmáximo [ms]	No. Entid.
25	0.27046	0.97714	0.11900	1.6158	72000
50	0.59184	0.36128	0.10300	1.4338	90000
100	0.99639	0.20504	0.81662	1.6238	18000
200	0.91300	0.53821	0.10300	1.3636	13500
1000	48.515	0.86706	0.10300	128.52	81000

Tabela 5.3: Tabela de Resultados da Simulação CAN

Os resultados obtidos demonstram que este protocolo consegue atender a todos os *deadlines* com uma margem de segurança bastante razoável.

5.6 Modelo para o Protocolo VAN

O protocolo VAN utilizou basicamente o mesmo modelo acima descrito por gerar frames de dados periodicamente e atender aos requisitos de atendimento preferencial aos frames mais prioritários. As alterações necessárias foram referentes aos tempos de propagação da mensagem devido às diferenças de *baudrate* e formato dos *frames*.

O VAN descreve um modelo de transmissão de dados do tipo assíncrono e a transferência é efetuada em pacotes.

5.6.1 Cálculo dos Tempos de Ocupação do Meio

Os dados de cada tipo foram agrupados em blocos e divididos em frames, respeitando o limite de 28 bytes de dados por frame no modo *Full* e 8 bytes no modo *Compact*.

Para fins de simulação, considerou-se o tempo em que um frame ocupa o barramento como sendo o seu tempo de propagação. Para este cálculo, foi utilizada a equação 5.3.

$$T_{propagação} = \frac{tamanho + 45}{baudrate} \quad (5.3)$$

Os termos da equação possuem o seguinte significado:

Tpropagação: é o tempo de propagação do frame completo em milissegundos (ms).

45: é o número de bits adicionados ao frame devido aos diversos campos que compoem o frame.

tamanho: é o número de bits que compõe o campo de dados.

baudrate: é a taxa de transmissão em bits por milissegundo, que neste caso, é igual a 100 bits/ms.

5.6.2 Dados Necessários à Simulação

Na figura 5.1 estão representadas as distâncias entre as estações nesta unidade.

Simulação no Modo Compact VAN

A tabela 5.4 lista o valores que foram necessários para se construir o modelo para simulação do protocolo VAN no modo *Compact*.

Dado Tipo	Tamanho [bits]	Dados/Frame	Frames	Tprop. [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	64	2	1.09	60	55.05
comm	1152	64	18	1.09	10	9.17
w	64	64	1	1.09	50	45.87
acel	48	48	1	0.93	20	21.51
atit	64	64	1	1.09	30	27.52
xyz	144	48	3	0.93	40	43.01
vel	48	48	1	0.93	40	43.01
refs	192	64	3	1.09	10	9.17
opcode	144	48	3	0.93	10	10.75
offset	192	64	3	1.09	10	9.17
rpm	16	16	1	0.61	50	81.97
battemp	48	48	1	0.93	50	53.76

Tabela 5.4: Dados Necessários à Simulação Utilizando o Módulo Compact VAN

Para a câmera de vídeo que gera 64 Kbytes por segundo, seus dados ficaram divididos em 64 pacotes, tendo cada um 16 frames de dados, cada frame com um campo de dados de 64 bits (que é o número máximo de bits de dados por frame admitido pelo protocolo).

O tempo de propagação de cada pacote (cada pacote representa uma entidade) é de 16 vezes 1.09 ms⁵, totalizando 17.44 ms.

⁵tempo de propagação de um frame de 64 bits de dados

De acordo com a figura 5.1, a distância entre a estação que representa a câmera (Câmera) e estação que representa os enlaces de rádio frequência (RFLinks) é de 60 células. Sendo assim, a velocidade que cada entidade utiliza na simulação para percorrer seu trajeto é de 3.44 cells/ms.

Simulação no Modo Full VAN

A tabela 5.5 lista os dados necessários para simulação no modo *Full*.

Dado Tipo	Tamanho [bits]	Dados/Frame	Frames	Tprop. [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	128	1	1.73	60	34.68
comm	1152	192	6	2.37	10	4.22
w	64	64	1	1.09	50	45.87
acel	48	48	1	0.93	20	21.51
atit	64	64	1	1.09	30	27.52
xyz	144	144	1	1.89	40	21.16
vel	48	48	1	0.93	40	43.01
refs	192	192	1	2.37	10	4.22
opcode	144	144	1	1.89	10	5.29
offset	192	192	1	2.37	10	4.22
rpm	16	16	1	0.61	50	81.97
battemp	48	48	1	0.93	50	53.76

Tabela 5.5: Dados Necessários à Simulação Utilizando o Módulo Full VAN

Para a câmera de vídeo que gera 64 Kbytes por segundo, seus dados ficaram divididos em 64 pacotes, tendo cada um 8 frames de dados, cada frame com um campo de dados de 128 bits.

O tempo de propagação de cada pacote (cada pacote representa uma entidade) é de 8 vezes 1.73 ms⁶, totalizando 13.84 ms.

Como a distância entre a estação que representa a câmera (Câmera) da estação que representa os enlaces de rádio frequência (RFLinks) 5.1 é de 60 células, a velocidade que cada entidade utiliza na simulação para percorrer seu trajeto é de 4.34 cells/ms.

5.6.3 Resultados Obtidos na Simulação

Para o modo *Compact*, observou-se que os *deadlines* até 200 ms foram obedecidos, mas, para os dados com *deadlines* iguais a 1000 ms, estes valores atingiram até 1486.9 ms para chegarem ao seu destino, como pode ser visto na tabela 5.6.

Cabe justificar que, com o acúmulo de entidades com *deadlines* iguais a 1000 ms aguardando na fila para serem atendidas, o limite da ferramenta de simulação utilizada, que prevê um máximo de 150 entidades no sistema foi rapidamente atingido. Por este motivo, o tempo de simulação foi limitado a 2 segundos. O total de entidades de cada tipo produzidas neste período está representado na última coluna.

Para o modo *Full*, observou-se praticamente o mesmo fenômeno anterior, com algumas variações. Para dados com *deadlines* de até 200ms, os tempos de chegada foram obedecidos, mas, para os dados com *deadlines* iguais a 1000 ms, os tempos de chegada atingiram valores de até 1936.7 ms, como pode ser visto na tabela 5.7.

⁶tempo de propagação de um frame de 128 bits de dados

Deadline [ms]	Tmédio [ms]	Tvariação [ms]	Tmínimo [ms]	Tmáximo [ms]	No. Entid.
25	9.9734	0.52203	1.2072	19.565	160
50	13.3190	0.38411	3.5624	24.087	200
100	16.2500	0.31292	8.0999	25.787	40
200	19.9020	0.22497	10.5380	28.640	30
1000	487.7500	0.94503	1.0905	1486.900	139

Tabela 5.6: Resultados da Simulação Utilizando Compact VAN

Neste caso, apesar do tempo de chegada dos dados com *deadlines* iguais a 1000 ms terem superado o teste anterior, os dados não atendidos se acumularam em menor quantidade na fila, permitindo uma simulação de 11 segundos. Isto ocorreu porque os outros dados, de *deadlines* menores tiveram tempos de chegada também menores em relação ao teste anterior. O total de entidades de cada tipo produzidas neste período está representado na última coluna.

Deadline [ms]	Tmédio [ms]	Tvariação [ms]	Tmínimo [ms]	Tmáximo [ms]	No. Entid.
25	8.6014	0.47360	1.7516	15.548	440
50	10.821	0.38098	2.7905	19.282	660
100	13.986	0.27887	7.0809	20.982	220
200	15.954	0.24083	10.061	21.914	55
1000	848.13	0.60296	2.3696	1936.7	753

Tabela 5.7: Resultados da Simulação Utilizando Full VAN

5.7 Modelo para o Protocolo J1850

Para o protocolo J1850 foi utilizado basicamente o mesmo modelo anterior devido ao fato de os dados terem sua geração periódica respeitada e o atendimento das prioridades ser satisfeita.

As alterações no modelo foram no sentido de adaptar os tempos de propagação das mensagens devido às características de *baudrate* e tamanho dos *frames* específicos do J1850.

5.7.1 Cálculo dos Tempos de Ocupação do Meio

Os dados de cada tipo foram agrupados em blocos e divididos em *frames*, respeitando o limite de 11 *bytes* de dados por *frame*.

Para fins de simulação, considerou-se o tempo em que um *frame* ocupa o barramento como sendo o seu tempo de propagação. Para este cálculo, foi utilizada a equação 5.4.

Na figura 5.1 estão representadas as distâncias entre as estações nesta unidade. Assim, para cada bloco descrito, foram calculados os parâmetros necessários à simulação, representados na tabela 5.8 ⁷.

$$T_{propagação} = \frac{\text{tamanho} + 12}{\text{baudrate}} \quad (5.4)$$

⁷Os dados na câmera não aparecem na tabela por questões discutidas mais adiante

Os termos da equação possuem o seguinte significado:

Tpropagação: é o tempo de propagação do frame completo em milissegundos (ms).

tamanho: é o número total de bits que compõe o campo de dados.

l2: é o número de bits adicionados em cada frame de dados devido aos campos e ao espaço interframes.

baudrate: é a taxa de transmissão em bits por milissegundo ⁸.

5.7.2 Dados Necessários à Simulação

A tabela 5.8 lista os valores que foram necessários para se construir o modelo para simulação do protocolo J1850.

Dado Tipo	Tamanho [bits]	Dados/Frame	Frames	Tprop. [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	64	2	1.827	60	32.84
comm	1152	72	16	2.019	10	4.95
w	64	64	1	1.827	50	27.36
acel	48	48	1	1.442	20	13.88
atit	64	64	1	1.827	30	16.42
xyz	144	72	2	2.019	40	19.81
vel	48	48	1	1.442	40	27.74
refs	192	64	3	1.827	10	5.47
opcode	144	72	2	2.019	10	4.95
offset	192	64	3	1.827	10	5.47
rpm	16	16	1	0.673	50	74.29
battemp	48	48	1	1.442	50	34.67

Tabela 5.8: Tabela de Dados Necessários à Simulação J1850

5.7.3 Resultados Obtidos na Simulação

Como a câmera de vídeo gera dados a uma taxa de 64 kbits/s e o protocolo J1850 admite um *baudrate* máximo de 41.6 kbits/s, pode-se concluir, mesmo sem realizar a simulação, que os *deadlines* não serão respeitados, pelo fato destes dados se acumularem na fila de espera com uma taxa maior do que o atendimento. Com base neste argumento, a simulação para este protocolo foi realizada sem considerar estes dados. Para uma possível implementação, a solução seria utilizar um canal alternativo para o envio de imagens. Na realidade, esta solução não iria contemplar a aplicação a longo prazo, já que futuramente pretende-se utilizar este sinal como dado auxiliar para vôos fora do alcance visual.

A tabela 5.9 mostra os resultados obtidos na simulação por um período de 10 minutos (600.000 ms). O total de entidades de cada tipo geradas neste período estão representadas na última coluna.

5.8 Modelo para o Protocolo MIL-STD-1553B

O protocolo MIL-STD-1553B está baseado em uma arquitetura mestre-escravo, onde um nó (mestre) controla a transmissão e recepção dos outros nós (escravos) conforme descrito no capítulo 4.

⁸Foi utilizada a taxa de 41.6 bit/ms

Deadline [ms]	Tmédio [ms]	Tvariação [ms]	Tmínimo [ms]	Tmáximo [ms]	No. Entid.
25	3.0777	0.37519	1.8270	5.6730	48000
50	8.4411	0.29094	1.4410	13.3260	60000
100	13.3370	0.21918	1.8270	15.8260	12000
200	11.6100	0.67491	2.0189	19.672	6000
1000	46.6690	0.44146	1.4420	81.167	13800

Tabela 5.9: Tabela de Resultados da Simulação J1850 sem os Dados da Câmera

O modelo utilizado para os outros protocolos teve alterações com respeito aos tempos de propagação, em função das particularidades do MIL-STD-1553B.

5.8.1 Cálculo dos Tempos de Ocupação do Meio

Para o modelo utilizado na simulação, considerou-se o nó Controlador como sendo o BC da rede e todos os outros nós como RTs.

Como a aplicação estudada não envolve a transmissão de mensagens aperiódicas, pode-se considerar que as prioridades necessárias a cada nó serão respeitadas. Para isto, basta que o BC solicite troca de informações respeitando a seqüência de prioridades e os intervalos definidos na tabela 5.1.

Baseando-se nas premissas descritas acima, pode-se utilizar o mesmo modelo ARENA das simulações anteriores, que garante a troca periódica de mensagens e o respeito às prioridades.

O tempo de ocupação do barramento por cada mensagem será composto pelo tempo de propagação de cada frame envolvido no diálogo correspondente mais os intervalos citados na figura 4.10.

No modelo estudado, o tempo de ocupação do barramento dependerá do tipo de transação: BC-RT, RT-BC ou RT-RT.

Transferências do Tipo BC-RT e RT-BC

Para estes diálogos onde o BC solicita a um RT que este receba ou envie dados, o cálculo é dado pela equação 5.5.

$$T_{prop.} = \frac{NDados * 20 + 40}{baudrate} + 0.012 \quad (5.5)$$

Os termos da equação possuem o seguinte significado:

Tprop: é o tempo de ocupação do barramento para uma transferência de dados entre o BC e um RT (ou vice-versa) em milisegundos.

NDados: é o número de *frames* de dados que serão utilizados para transferir toda a mensagem.

20: é o tamanho do *frame* de dados em bits.

40: é a soma em bits dos *frames* de comando e *status* envolvidos no diálogo.

baudrate: é a taxa de transferência de dados em bits por milisegundo. No MIL-STD-1553B esta taxa é de 1000 bits/ms.

0.012: é a soma em milisegundos dos tempos de resposta (8 microsegundos) e intervalo entre mensagens (4 microsegundos), conforme figura 4.10.

Transferência RT-RT

Para o diálogo RT-RT, onde o BC solicita a um RT que este envie dados para outro RT, o cálculo é dado pela equação 5.6.

$$T_{prop.} = \frac{NDados * 20 + 80}{baudrate} + 0.020 \quad (5.6)$$

Os termos da equação possuem o seguinte significado:

Tprop: é o tempo de ocupação do barramento para uma transferência de dados entre um RT e outro RT em milisegundos.

NDados: é o número de *frames* de dados que serão utilizados para transferir toda a mensagem.

20: é o tamanho do *frame* de dados em bits.

80: é a soma em bits dos dois *frames* de comando e os dois de status envolvidos no diálogo.

baudrate: é a taxa de transferência de dados em bits por milisegundo. No MIL-STD-1553B esta taxa é de 1000 bits/ms.

0.020: é a soma em milisegundos dos dois tempos de resposta ($2 * 8$ microsegundos) e intervalo entre mensagens (4 microsegundos), conforme figura 4.10.

5.8.2 Dados Necessários à Simulação

Como a comunicação entre nós do tipo RT sempre envolve a participação do BC, neste caso, diferente dos anteriores, não se pode desconsiderar, para efeitos de simulação, que todas as estações que enviam dados para o nó **Controlador** também transmitem os mesmos dados para o nó **RFLinks**. Este é o motivo pelo qual alguns dados aparecem duplicados na tabela 5.10, uma vez realizando uma transação do tipo 1 (BC-RT ou RT-BC) e outra como tipo 2 (RT-RT).

O número de frames de dados gerados é limitado a 32 porque este é o número máximo que o contador de frames de dados localizado no frame de comando suporta (seu campo possui 5 bits - ver tabela 4.8). Assim, em informações que necessitavam de um número maior de frames para serem transmitidas, foram criadas entidades no modelo ARENA que são constituídas de partes da informação. Assim, as n entidades geradas totalizam a informação completa.

Este artifício teve que ser utilizado nos dados tipo *comm* que totalizavam 1152 bits e tiveram que ser divididos em 3 entidades, cada uma contendo 24 frames de dados (cada frame suporta 16 bits).

A tabela 5.10 relaciona os valores obtidos dos tempos de ocupação do barramento segundo a distribuição da informação em frames de dados e entidades. Nesta fase, os valores para cada tipo de transação são apresentados⁹.

Os termos da tabela 5.10 possuem o seguinte significado:

Dado Tipo: é o tipo do dado a ser transportado.

Tam: é o tamanho do dado em bits.

NFrames Dados: é o número de *frames* de dados que será transportado por cada entidade do ARENA.

Entid.: é o número de entidades geradas para se obter a informação completa.

⁹Os dados referentes à câmera de vídeo não aparecem na tabela porque necessitaram receber tratamento especial, como será visto mais adiante

Dado Tipo	Tam. [bits]	NFrames Dados	Entid.	Tipo Trans.	Ttotal [ms]
servos	128	8	1	1	0.212
comm	1152	24	3	1	0.532
w	64	4	1	1	0.132
acel	48	3	1	1	0.112
atit	64	4	1	1	0.132
xyz	144	9	1	1	0.232
vel	48	3	1	1	0.112
refs	192	12	1	1	0.292
opcode	144	9	1	1	0.232
offset	192	12	1	1	0.292
rpm	16	1	1	1	0.072
battemp	48	3	1	1	0.112
servos	128	8	1	2	0.260
comm	1152	24	3	2	0.580
w	64	4	1	2	0.180
acel	48	3	1	2	0.160
atit	64	4	1	2	0.180
xyz	144	9	1	2	0.280
vel	48	3	1	2	0.160
refs	192	12	1	2	0.340
opcode	144	9	1	2	0.280
offset	192	12	1	2	0.340
rpm	16	1	1	2	0.120
battemp	48	3	1	2	0.160

Tabela 5.10: Tabela de Dados MIL-STD-1553B (Fase A)

Tipo Trans: é o tipo de transação necessária para a comunicação. Tipo 1 corresponde à transação BC-RT ou RT-BC e o tipo 2 corresponde à transação RT-RT.

Ttotal: é o tempo total de ocupação do barramento em milisegundos calculado segundo as equações 5.5 e 5.6.

Se um dado se destina a ser transportado até mais de uma estação, envolvendo dois tipos diferentes de transação, pode-se considerar como tempo total em que ele ocupa o barramento como sendo a soma dos tempos envolvidos em cada transação.

No caso dos dados transmitidos até o nó **Controlador** e o nó **RFlinks**, soma-se os tempos das duas transações. Para a obtenção da velocidade de deslocamento, escolhe-se uma das duas estações **Controlador** ou **RFlinks** e se calcula a velocidade utilizando-se sua distância em células. A estação escolhida não é relevante, já que a distância e a velocidade servem apenas para simular um tempo de ocupação do barramento.

Esta simplificação é importante para que os recursos da ferramenta de simulação não sejam exauridos a ponto de impedir o experimento.

Para a simulação, os dados constantes da tabela 5.11 foram utilizados.

Os termos da tabela 5.11 possuem o seguinte significado:

Dado Tipo: é o tipo do dado a ser transportado.

Dado Tipo	Tam. [bits]	NEnt.	Ttotal [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	1	0.472	60	127.12
comm	1152	3	1.112	10	8.99
w	64	1	0.312	50	160.25
acel	48	1	0.272	20	73.53
atit	64	1	0.312	30	96.15
xyz	144	1	0.512	40	78.12
vel	48	1	0.272	40	147.06
refs	192	1	0.632	10	15.82
opcode	144	1	0.512	10	19.53
offset	192	1	0.632	10	15.82
rpm	16	1	0.192	10	52.08
battemp	48	1	0.272	10	36.76
vídeo	65536	128	0.740	60	81.08

Tabela 5.11: Tabela de Dados MIL-STD-1553B (Fase B)

Tam: é o tamanho do dado em bits.

NEntid: é o número de entidades geradas para se obter a informação completa.

TTotal: é a soma dos tempos das transações envolvidas em milissegundos.

Dist: é a distância entre a estação transmissora e a receptora em células. No caso do dado ser transmitido para a estação **Controlador** e **RFLinks**, utilizou-se a distância até a estação **Controlador**.

Veloc: é a velocidade necessária para que o dado ocupe o barramento pelo tempo **TTotal**. É dada em células por milissegundo.

Os dados referentes à câmera de vídeo tiveram que ser submetidos a um tratamento especial devido ao grande volume de dados envolvido, superando os recursos disponíveis na ferramenta de simulação.

A estação **Câmera** necessita transferir 65536 bits a cada segundo. Como cada frame de dados suporta 16 bits e cada transação pode envolver até no máximo 32 *frames* de dados, resulta num total de 128 transações RT-RT. Como não é possível representar as 128 entidades no modelo, foram criados 32 pacotes, cada um contendo 4 transações RT-RT.

Nesta configuração, cada transação demora 0.740 milissegundos, o que resulta que um pacote representado por uma entidade ocupa o barramento durante 2.96 milissegundos.

Num total, foram geradas 32 entidades, cada uma ocupando o barramento por 2.96 milissegundos, sendo que a velocidade calculada para a simulação foi de 20.27 células/ms.

5.8.3 Resultados Obtidos na Simulação

A tabela 5.12 mostra a média dos resultados obtidos em 10 simulações por um período de 15 minutos (900.000 ms) cada. O total de entidades de cada tipo geradas neste período está representado na última coluna.

Pode-se observar, em vista dos resultados obtidos, que todos os *deadlines* foram obedecidos com razoável margem de folga.

Deadline [ms]	Tmédio [ms]	Tvariação [ms]	Tmínimo [ms]	Tmáximo [ms]	No. Entid.
25	0.67014	0.69079	0.47199	2.6306	36000
50	1.2515	0.38067	0.27200	2.8118	54000
100	2.9132	0.10314	2.4609	3.4174	18000
200	1.1235	1.0857	0.51203	3.9171	4500
1000	50.121	0.66102	0.27200	109.33	35100

Tabela 5.12: Tabela de Resultados MIL-STD-1553B

5.9 Modelo para o Protocolo C²D

O protocolo utiliza a topologia tipo barramento, em que todos os nós estão conectados ao mesmo meio físico.

Com a atribuição de uma prioridade a cada mensagem, é possível realizar a arbitragem garantindo sempre a seqüência de direito de envio à mensagem mais prioritária sem a perda de nenhum dado (no caso de mais de um nó detectar barramento desocupado ao mesmo tempo e necessitar enviar dados).

Neste caso, o modelo utilizado até agora para as simulações também se mostrou adequado ao C²D, adaptando-se os tempos de propagação em função das particularidades do protocolo.

5.9.1 Cálculo dos Tempos de Ocupação do Meio

Os dados de cada tipo foram agrupados em blocos e divididos em frames, respeitando o limite de 4 bytes de dados por frame.

Para fins de simulação, considerou-se o tempo em que um frame ocupa o barramento como sendo o seu tempo de propagação. Para este cálculo, foi utilizada a equação 5.7.

Na figura 5.1 estão representadas as distâncias entre as estações nesta unidade. Assim, para cada bloco descrito, foram calculados os parâmetros necessários à simulação, representados na tabela 5.13 ¹⁰.

$$T_{propagação} = \frac{tamanho * 10 + 30}{baudrate} \quad (5.7)$$

Os termos da equação possuem o seguinte significado:

T_{propagação}: é o tempo de propagação do frame completo em milissegundos (ms).

30: é o número de bits adicionados ao frame devidos ao campo identificador, o campo de checagem de erro e o espaço interframes.

tamanho: é o número de bytes que compõe o campo de dados.

10: é o número de bits necessário para cada byte de dados (8 bits de dados, 1 *start bit* e 1 *stop bit*).

baudrate: é a taxa de transmissão em bits por milissegundo, que neste caso, é igual a 7.812,5 bits/ms.

5.9.2 Dados Necessários à Simulação

A tabela 5.13 lista os valores que foram necessários para se construir o modelo para simulação do protocolo C²D.

¹⁰Os dados na câmera não aparecem na tabela por questões discutidas mais adiante

Dado Tipo	Tamanho [bits]	Bytes/Frame	Frames	Tprop. [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	4	4	12.80	60	4.687
comm	1152	4	36	12.80	10	0.781
w	64	4	2	12.80	50	3.906
acel	48	3	2	7.68	20	2.604
atit	64	4	2	12.80	30	2.343
xyz	144	3	6	7.68	40	5.208
vel	48	3	2	7.68	40	5.208
refs	192	4	6	12.80	10	0.781
opcode	144	3	6	7.68	10	1.302
offset	192	4	6	12.80	10	0.781
rpm	16	2	1	6.40	50	7.812
battemp	48	3	2	7.68	50	6.510

Tabela 5.13: Tabela de Dados Necessários à Simulação C²D

5.9.3 Resultados Obtidos na Simulação

Como a câmera de vídeo gera dados a uma taxa de 64 kbits/s e o protocolo C²D admite um *baudrate* máximo de 7.8125 kbits/s, pode-se concluir, mesmo sem realizar a simulação, que os *deadlines* não serão respeitados. Com base neste argumento, a simulação para este protocolo foi realizada sem considerar estes dados.

Para a execução da simulação, foi necessário reduzir ao máximo a geração de entidades, pois como a sua permanência no sistema possui um valor alto devido ao *baudrate* e ao tamanho dos frames, o limite de 150 entidades é esgotado rapidamente. Isto ocorreu devido ao fato de haver acúmulo de entidades esperando na fila, por um motivo facilmente detectável.

No caso dos dados de maior prioridade (*servos*, com prioridade 0), de acordo com a tabela 5.13, o tempo que as 4 entidades geradas levam para chegar ao seu destino é dado pela equação 5.8.

$$T_{propagação}(servos) = 4 * 12.8 = 51.2ms \quad (5.8)$$

Neste caso, o tempo de propagação chega a ser duas vezes maior que seu *deadline* que é igual a 25 ms, que também é o valor do intervalo de tempo em que estes dados são gerados.

O que ocorre, é que, sendo a fila organizada por ordem de prioridade, a quantidade de dados de prioridade 0 é muito superior à sua capacidade de fluxo, fazendo com que a fila cresça indefinidamente e os dados de prioridade imediatamente inferior nunca sejam atendidos.

Para este modelo, conclui-se que nenhum *deadline* foi respeitado.

5.10 Modelo para o Protocolo CSMA/DCR

Para o CSMA/DCR foi utilizado basicamente o mesmo modelo utilizado pelos protocolos anteriores.

As alterações, neste caso, foram um pouco mais profundas, já que não ficaram restritas aos tempos de propagação das mensagens.

Como neste protocolo a prioridade é atribuída ao nó e não à mensagem, foram necessárias modificações nas prioridades atribuídas às mensagens até aqui. Agora, todas as mensagens geradas a partir de um

mesmo nó deverão ter prioridades iguais. Este atributo também valerá como índice no cálculo dos tempos de resolução de conflitos em caso de colisão.

5.10.1 Cálculo dos Tempos de Ocupação do Meio

Como nos modelos anteriores, aqui também cada *frame* é representado por uma entidade. Para simular o tempo em que o bloco está ocupando o barramento são utilizadas variáveis simbólicas que representam a velocidade com que a entidade se desloca no modelo (em células/ms) e a distância entre as estações (em células).

Os dados de cada tipo foram agrupados em blocos e divididos em frames, respeitando o limite mínimo de 48 bytes de dados por frame.

Como o protocolo é determinista, pode-se calcular o tempo máximo de busca na árvore binária para cada *frame* e embutir este valor no tempo total de ocupação do barramento.

Para fins de simulação, considerou-se o tempo em que um *frame* ocupa o barramento como sendo o seu tempo de propagação mais o tempo de busca até conseguir completar a transmissão.

Cálculo do Tempo de Propagação

Para o cálculo do tempo de propagação de cada frame foi utilizada a equação 5.9.

$$T_{propagação} = \frac{tamanho + 208}{baudrate} \quad (5.9)$$

Os termos da equação possuem o seguinte significado:

T_{propagação}: é o tempo de propagação do frame completo em milissegundos (ms).

tamanho: é o número total de bits que compõe o campo de dados ¹¹.

208: é o número de bits adicionados em cada frame de dados devido aos campos descritos na tabela 4.12.

baudrate: é a taxa de transmissão em bits por milissegundo ¹².

Cálculo do Tempo de Busca

Para o cálculo do tempo de busca é necessário que se conheça a prioridade da mensagem para a qual se está calculando este valor.

A tabela 5.14 mostra a nova ordem de prioridades atribuídas às mensagens em função de seus períodos e tamanhos. O critério utilizado foi o seguinte: nós transmissores de frames com mais baixo período obtiveram prioridade maior em conformidade com o critério *rate-monotonic*. No caso de períodos iguais, a maior prioridade ficou com o nó gerador de menor volume de dados.

Segundo [LR93], o tempo decorrido em um ramo da árvore de busca do protocolo CSMA/DCR varia de 25 a 50 microsegundos em uma rede padrão Ethernet ou compatível com o padrão IEEE 802.3. Nesta modelagem, utilizou-se um valor 50 microsegundos para simular a pior situação.

Pode-se calcular o máximo tempo de busca de um *frame* com o auxílio da equação 5.10, que calcula o número máximo de ramos percorridos dados o maior índice da árvore e o índice da estação transmissora.

¹¹O valor mínimo admitido é de 384 bits (48 bytes)

¹²No padrão Ethernet, o baudrate máximo é 10 Mbit/s ou 10000 bit/ms

Nó	Dado Tipo	Tamanho [bits]	Período [ms]	Índice (Prior)
Control	servos	128	25	0
	comm	24 x 48	1000	0
Inercial	w	64	50	1
	acel	48	50	1
RFLinks	refs	192	50	2
	opcode	144	200	2
	offset	192	1000	2
Atitude	atit	64	100	3
Monit	rpm	1 x 16	100	4
	battemp	3 x 16	1000	4
Posição	xyz	144	1000	5
	vel	48	1000	5
Câmera	video	65536	1000	6

Tabela 5.14: Prioridades para os Mensagens no CSMA/DCR

$$\varphi(V) = \log t + V - \sigma(V) \tag{5.10}$$

Os termos da equação possuem o seguinte significado:

V: é o índice da estação.

$\varphi(V)$: é o número máximo de ramos que serão percorridos por um frame proveniente de uma estação com índice V.

t: é a menor potência de 2 imediatamente superior ou igual ao maior índice. Neste caso, o maior índice é 6 (vide tabela 5.14), sendo então $t = 8$ ($2^2 = 4$ e $2^3 = 8$) e $\log t = 3$.

$\sigma(V)$: é o número de bases da potência de 2 igual ao índice que se está calculando (V). Assim, para um $V = 7$, $\sigma(7) = 3$, pois $7 = 2^2 + 2^1 + 2^0$.

Desta forma, o cálculo do tempo máximo de busca em milissegundos para um determinado índice é dado pela equação 5.11, onde 0.05 corresponde ao valor de 50 microsegundos que é o tempo decorrido em um ramo da árvore de busca.

$$t_{busca} = \varphi(V) * 0.05 \tag{5.11}$$

5.10.2 Dados Necessários à Simulação

A tabela 5.15 lista os dados necessários à realização da simulação e construção do modelo.

Os termos da tabela 5.15 possuem o seguinte significado:

Tp: é o tempo de propagação em milissegundos.

Tb: é o tempo de busca em milissegundos.

Ttotal: é a soma dos tempos de busca e propagação.

Dado Tipo	Tam. [bits]	Dados/Frame	Frames	Tp [ms]	Tb [ms]	Ttotal [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	128	1	0.0336	0.15	0.1836	60	326.797
comm	1152	1152	1	0.1360	0.15	0.2860	10	34.965
w	64	64	1	0.0272	0.15	0.1772	50	282.167
acel	48	48	1	0.0256	0.15	0.1756	20	113.89
refs	192	192	1	0.0400	0.20	0.2400	10	41.666
opcode	144	144	1	0.0352	0.20	0.2352	10	42.517
offset	192	192	1	0.0400	0.20	0.2400	10	41.666
atit	64	64	1	0.0272	0.20	0.2272	30	132.042
rpm	16	48	1	0.0256	0.30	0.3256	50	153.562
battemp	48	48	1	0.0256	0.30	0.3256	50	153.562
xyz	144	144	1	0.0352	0.30	0.3352	40	119.332
vel	48	48	1	0.0256	0.30	0.3256	40	122.850
vídeo	65536	1024	8	0.1232	0.35	0.4732	60	126.796

Tabela 5.15: Tabela de Dados Necessários à Simulação CSMA/DCR

5.10.3 Resultados Obtidos na Simulação

A tabela 5.16 mostra a média dos resultados obtidos em dez simulações por um período de 15 minutos (900.000 ms) cada.

Deadline [ms]	Tmédio [ms]	Tvariação [ms]	Tmínimo [ms]	Tmáximo [ms]	No. Entid.
25	0.22809	0.52228	0.18360	0.79520	36000
50	0.57935	0.38101	0.17561	1.5217	54000
100	1.2456	0.36926	0.22720	2.6353	18000
200	0.54001	1.1269	0.23520	1.7569	4500
1000	3.7153	0.55018	0.24000	6.7561	11700

Tabela 5.16: Tabela de Resultados da Simulação CSMA/DCR

Através dos resultados obtidos na simulação, pôde-se observar que os *deadlines* foram atendidos com razoável margem de folga.

5.11 Modelo para o Protocolo FIP

Para o protocolo FIP foi utilizado basicamente o mesmo modelo dos protocolos anteriores. As alterações necessárias para a adaptação do protocolo referem-se aos tempos de propagação.

5.11.1 Cálculo dos Tempos de Ocupação do Meio

Para o modelo utilizado na simulação, considerou-se o nó Controlador como sendo o Árbitro de Barramento da rede.

Como a aplicação estudada não envolve a transmissão de mensagens aperiódicas, pode-se considerar que as prioridades necessárias a cada nó serão respeitadas. Para isto, basta que o AB solicite troca de

informações respeitando a seqüência de prioridades e os intervalos definidos na tabela 5.1.

Baseando-se nas premissas descritas acima, pode-se utilizar o mesmo modelo ARENA das simulações anteriores, que garante a troca periódica de mensagens e o respeito às prioridades.

O tempo de ocupação do barramento por cada mensagem será composto pelo tempo de propagação de cada frame envolvido no diálogo correspondente. Para fins de simulação, o diálogo em questão será composto sempre de um frame identificador, emitido pelo AB, e o frame de dados, emitido pela estação produtora.

Assim, o tempo de utilização do barramento para a atualização de cada dado, é dado pela equação 5.12.

$$T_{prop.} = \frac{NDados + 99}{baudrate} \quad (5.12)$$

Os termos da equação possuem o seguinte significado:

Tprop: é o tempo de ocupação do barramento para uma transferência de dados desde a requisição pelo Árbitro até a chegada à estação destino (a quem o dado interessa). No caso da informação ser utilizada por mais de uma estação, considera-se apenas a mais longe no modelo.

NDados: é o número de bits de dados que serão utilizados em cada *frame*. Em alguns casos, para transmitir toda a informação, será necessário mais de um frame, mas este cálculo considera apenas o tempo de propagação de cada um separadamente.

99: é a soma em bits dos frames de requisição (59) mais os outros campos do frame de resposta (40), exceto o campo de dados.

baudrate: é a taxa de transferência de dados em bits por milissegundo. No FIP será utilizada a taxa de 2.5 Mbits/s devido ao fato de que o comprimento total da rede não ultrapassará (de forma alguma!) o limite de 200 metros. Assim, o valor utilizado na equação é de 2500 bits/ms.

5.11.2 Dados Necessários à Simulação

A tabela 5.17 relaciona os dados necessários à execução da simulação do modelo.

Os termos da tabela 5.17 possuem o seguinte significado:

Dado Tipo: é o tipo do dado a ser transportado.

Tam: é o tamanho do dado em bits.

Dados/Frame: é o tamanho em bits do campo de dados do frame.

Entid: é o número de entidades geradas para se obter a informação completa. Note-se que, neste caso, cada frame é representado por uma entidade.

Tprop: é o tempo de propagação do frame em milissegundos.

Dist: é a distância entre a estação transmissora e a receptora em células. No caso do dado ser transmitido para a estação **Controlador** e **RFLinks**, utilizou-se a distância até a estação **Controlador**.

Veloc: é a velocidade necessária para que o dado ocupe o barramento pelo tempo de propagação. É dada em células por milissegundo.

Dado Tipo	Tam. [bits]	Dados/ Frame	Entid.	Tprop [ms]	Dist. [cells]	Vel. [cells/ms]
servos	128	128	1	0.0908	60	660.79
comm	1152	576	2	0.2700	10	37.03
w	64	64	1	0.0652	50	766.87
acel	48	48	1	0.0588	20	340.13
atit	64	64	1	0.0652	30	460.12
xyz	144	144	1	0.0972	40	411.52
vel	48	48	1	0.0588	40	680.27
refs	192	192	1	0.1164	10	85.91
opcode	144	144	1	0.0972	10	102.88
offset	192	192	1	0.1164	10	85.91
rpm	16	16	1	0.0460	10	1086.95
battemp	48	48	1	0.0588	10	850.34
vídeo	65536	1024	64	0.4492	60	133.57

Tabela 5.17: Tabela de Dados Necessários à Simulação FIP

5.11.3 Resultados Obtidos na Simulação

A tabela 5.18 mostra os resultados obtidos na simulação por um período de 15 minutos (900.000 ms). O montante de entidades de cada tipo geradas neste período está representado na última coluna.

Deadline [ms]	Tmédio [ms]	Tvariação [ms]	Tmínimo [ms]	Tmáximo [ms]	No. Entid.
25	0.10811	0.37493	0.09080	0.27023	36000
50	0.23140	0.31922	0.05880	0.40848	54000
100	0.40178	0.23383	0.06520	0.51968	18000
200	0.18999	0.97423	0.09720	0.61688	4500
1000	14.683	0.61456	0.05880	30.21100	63000

Tabela 5.18: Tabela de Resultados da Simulação Utilizando FIP

Os resultados obtidos demonstram que o protocolo consegue atender a todos os *deadlines* com razoável margem de folga.

5.12 Análise dos Resultados

As figuras 5.4, 5.5, 5.6, 5.7 e 5.8 apresentam os tempos máximos de atendimento para cada grupo de variáveis com mesmo *deadline* utilizando os diversos protocolos. Na figura, não são mostrados os índices para o C²D e o J1850. O primeiro, por não conseguir cumprir nenhum *deadline*, chegando inclusive a impossibilitar uma simulação completa devido ao número muito grande de entidades na fila (ver seção 5.9.3). O segundo, por não suportar o volume de dados da câmera de vídeo (ver seção 5.7.3).

Segundo as simulações realizadas, além dos protocolos C²D e J1850 que não suportaram o fluxo de dados intra-veículos, o protocolo VAN (tanto no modelo *Full* como no *Compact*) também não atendeu aos *deadlines* dos dados com período de 1000 milissegundos.

Com relação aos *deadlines* dos outros grupos, cabe observar que a simulação considerou apenas os tempos de ocupação do barramento, isto é, somente as parcelas referentes à camada física e à subcamada

MAC do modelo OSI.

No caso de uma implementação de um destes protocolos, os valores observados serão maiores devido ao tempo gasto pelas camadas de enlace de dados e aplicação.

A camada de enlace de dados terá o trabalho de encapsular os dados para transmissão no formato dos frames e desencapsulá-los na recepção. Somente esta tarefa deverá consumir um tempo de processamento da ordem de milisegundos, tempo este que poderá ser determinante no atendimento aos *deadlines*.

Pelos motivos expostos acima, considera-se que apenas os protocolos que possuem uma margem de reserva de tempo para o atendimento dos *deadlines* possuem condições de serem implementados.

Nesta categoria, pode-se enquadrar os protocolos CAN, MIL-STD-1553B, CSMA/DCR e FIP, que, em vista dos resultados obtidos, possuem inclusive capacidade para suportar futuras ampliações do sistema.

5.13 Conclusão

Neste capítulo, foi apresentada e descrita a ferramenta de simulação utilizada neste trabalho bem como o modelo utilizado.

Com relação ao modelo, este se mostrou bastante flexível, pois permitiu a simulação de todos os protocolos com alterações pouco significativas do ponto de vista estrutural.

Foram descritos os métodos e cálculos utilizados na simulação de cada um dos protocolos e os resultados obtidos.

A tabela 5.19 mostra o resumo dos resultados obtidos para cada protocolo. Os resultados dos protocolos J1850 e C²D não estão incluídos por não terem conseguido suportar o fluxo de dados a ponto de inviabilizar a obtenção de resultados aceitáveis para análise.

Protocolo	Tmax ^a D=25	Tmax D=50	Tmax D=100	Tmáx D=200	Tmax D=1000
CAN	1.61	1.43	1.62	1.36	128.50
VANcomp	19.56	24.08	25.78	28.64	1486.90
VANfull	15.55	19.28	20.98	21.91	1936.70
1553B	2.63	2.81	3.42	10.28	109.33
DCR	0.79	1.52	2.63	1.76	6.76
FIP	0.27	0.41	0.52	0.62	30.21

^aTempo máximo em ms para cada *deadline*

Tabela 5.19: Resumo dos Resultados Obtidos na Simulação

Com base nas considerações relacionadas neste capítulo, concluímos que os protocolos CAN, MIL-STD-1553B, CSMA/DCR e FIP atendem aos *deadlines* no contexto da aplicação em questão.

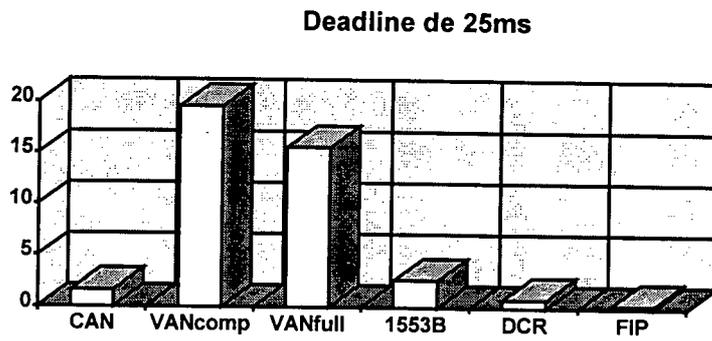


Figura 5.4: Comparação entre os Protocolos para *deadlines* de 25 ms

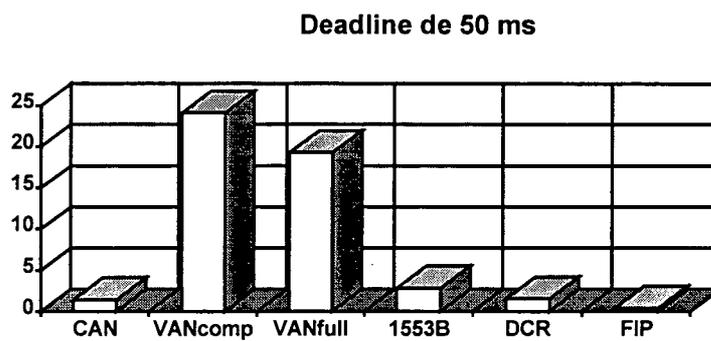


Figura 5.5: Comparação entre os Protocolos para *deadlines* de 50 ms

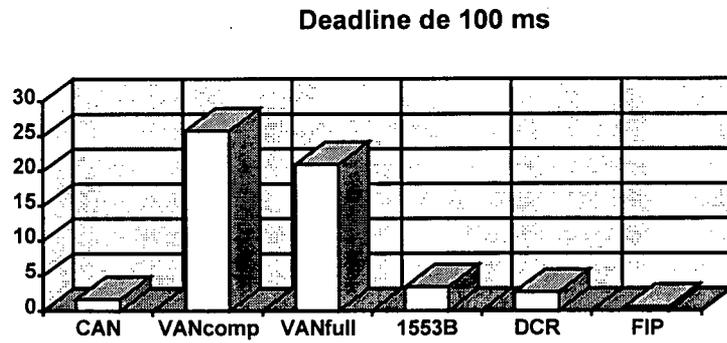


Figura 5.6: Comparação entre os Protocolos para *deadlines* de 100 ms

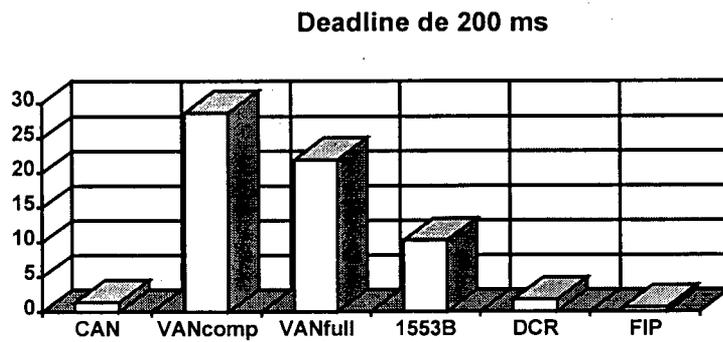


Figura 5.7: Comparação entre os Protocolos para *deadlines* de 200 ms

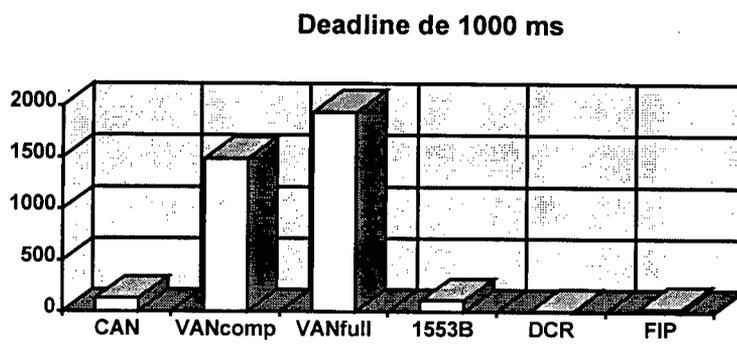


Figura 5.8: Comparação entre os Protocolos para *deadlines* de 1000 ms

Capítulo 6

Conclusões

6.1 Descrição da Metodologia Utilizada

O trabalho teve como objetivo oferecer uma proposta de arquitetura de comunicação distribuída para um helicóptero não tripulado baseada em protocolos já existentes conhecidos no mercado e utilizados em aplicações afins.

Para tanto, foi feito inicialmente um levantamento detalhado do fluxo de dados intra-veículo, estudando as informações necessárias aos sistemas de controle, monitoração e carga útil da aplicação, bem como suas fontes, periodicidades e resoluções.

A seguir, efetuou-se uma pesquisa bibliográfica sobre redes locais, enfocando principalmente o segmento de redes embarcadas em veículos.

As informações adquiridas através do estudo da problemática das redes tempo real foram combinadas com as obtidas em levantamento realizado sobre a literatura especializada em veículos não-tripulados.

Os conceitos utilizados em aviônica seguindo as mais recentes tendências da indústria aeronáutica civil e militar complementaram o panorama do ponto de vista da aplicação estudada.

Este embasamento permitiu a seleção de sete protocolos, escolhidos na ampla gama de opções oferecidas no mercado e propostas de instituições normalizadoras, para a realização de simulações de desempenho.

Os protocolos foram estudados com detalhe, sendo enfatizadas suas características diferenciais, tais como histórico, universo de usuários, topologia da rede, identificação com o modelo de referência OSI, formato das mensagens, método de acesso ao meio, mecanismos de detecção de erros e disponibilidade de hardware.

O próximo passo foi a familiarização com a ferramenta de simulação, escolhida pela sua disponibilidade, facilidade de utilização e interfaceamento gráfico, modularidade e orientação a objeto, características adequadas ao trabalho em questão.

De posse dos conceitos básicos da ferramenta, procurou-se desenvolver um modelo que simulasse com fidelidade a arquitetura de comunicação proposta para a aplicação e que além disso possuísse o maior grau de flexibilidade possível, para que as simulações pudessem ser realizadas para os diferentes protocolos com poucas alterações. As características de modularidade e orientação a objeto foram bastante úteis nesta etapa.

Para cada um dos protocolos, foram realizados os cálculos necessários para a implementação no modelo. A partir daí, todas as simulações foram computadas e registradas.

O passo seguinte foi analisar os resultados obtidos do ponto de vista do desempenho e atendimento

às restrições de *deadline* das mensagens. Este critério eliminou alguns dos protocolos candidatos.

Dos protocolos restantes, examinou-se então a viabilidade para implementação do ponto de vista operacional, isto é, se havia *hardware* disponível e com que grau de confiabilidade (inferido em função do histórico de usuários e universo de implementações conhecidas).

Finalmente, foi possível fundamentar a proposta oferecendo quatro opções de protocolos com potencial sucesso de implementação para a aplicação em questão.

6.2 Resultados Obtidos

Dos protocolos selecionados para simulação, o CAN, o MIL-STD-1553B, o CSMA/DCR e o FIP obtiveram resultados bastante satisfatórios no que diz respeito ao atendimento aos *deadlines* para esta aplicação. Os demais (VAN, C2D e J1850) mostraram-se inadequados com relação a este quesito.

Com relação à facilidade de implementação, as seguintes considerações são feitas:

CAN Este protocolo é bastante difundido na indústria de um modo geral. Possui ampla gama de usuários e um histórico de utilização que permite a obtenção de suporte técnico sem maiores problemas. A variedade de produtos e fabricantes é bem ampla, já que o protocolo também está sendo considerado como alternativa para redes locais industriais. Apesar destas vantagens e da concepção direcionada à eletrônica embarcada, não se observou a utilização deste padrão na área de aviãoica, tanto civil quanto militar.

MIL-STD-1553B Este protocolo é bastante popular na indústria aeronáutica civil e militar, tendo ampla aceitação e um histórico de utilização que atesta a sua confiabilidade e robustez. Possui uma variedade de fabricantes e produtos compatíveis. Em relação aos demais, é o único que não possui cálculo de CRC¹ na checagem de erros, mas esta característica não parece ser preponderante na indústria que o adotou como padrão.

CSMA/DCR Este protocolo, apesar de recente, baseia-se no padrão Ethernet, que é um dos mais utilizados em redes locais na área fabril [Sch89]. Em eletrônica embarcada sua utilização é pouco conhecida, porém denotando sucesso nas implementações realizadas até aqui, inclusive na indústria militar, onde concorre com o MIL-STD-1553B. Possui ainda poucos fabricantes para seus produtos, podendo-se citar a Aptor (França) e a *Dassault Electronicque*, que fabrica a rede Recital baseada neste protocolo.

FIP Este protocolo apresentou um dos melhores desempenhos nas simulações realizadas. Possui uma variedade de fabricantes e usuários que confirmam a confiabilidade do sistema a nível industrial. Não se tem notícia de aplicações envolvendo eletrônica embarcada, mas possui grandes possibilidades de se adequar à aplicação em questão.

Com estas considerações, pode-se concluir que os protocolos CAN, MIL-STD-1553B, CSMA/DCR e FIP possuem potencial para uma implementação bem sucedida no contexto da aplicação em questão, com apenas uma ressalva: na questão da robustez. Neste caso, os protocolos que se utilizam de métodos de acesso ao meio que não necessitam centralizar o controle do barramento, como o CAN e o CSMA/DCR, são mais robustos, e por este motivo, mais recomendados para esta aplicação.

6.3 Contribuições Resultantes do Trabalho

Do ponto de vista acadêmico, pode-se citar a disponibilização de um modelo de simulação suficientemente flexível para a utilização em outros tipos de veículos, tripulados ou não, terrestres, aéreos ou submarinos, que possuam qualquer tipo de "inteligência" embarcada que se pretenda distribuir.

¹Utiliza checagem de paridade.

O modelo também se presta a simular outros protocolos que por ventura possam ser propostos para a mesma aplicação ou outra similar, mediante a execução das alterações necessárias a cada caso específico.

Do ponto de vista técnico/industrial, vislumbrou-se um novo segmento de aplicações até então pouco explorado pelos protocolos desenvolvidos para eletrônica embarcada em veículos automotivos convencionais, como o CAN. O mesmo se aplica para protocolos que apresentaram excelente performance, mas que tinham sua utilização até então voltada para o mercado de redes locais industriais, como a proposta FIP, ou aplicações mais heterogêneas, como o CSMA/DCR.

A simulação também serviu para corroborar a eficácia da utilização do protocolo MIL-STD-1553B, já amplamente utilizado neste setor.

6.4 Sugestões para Futuros Trabalhos

Como sugestões para o prosseguimento do trabalho em questão, propõe-se o estudo mais aprofundado das outras camadas do modelo OSI para estes protocolos com o objetivo de obter um modelo aperfeiçoado. Sugere-se a implementação de uma subcamada LLC capaz de realizar um escalonamento de mensagens penderentes por prioridade ou *deadline*.

A atribuição de prioridades também pode utilizar um critério diferente do *rate-monotonic* adotado neste trabalho. Esta atribuição poderia ser baseada em uma análise mais detalhada em termos de escalonabilidade.

Também julgamos interessante o estudo do comportamento de um modelo com previsão para tarefas aperiódicas e esporádicas.

Bibliografia

- [Ass91] Associazione Tecnica de Automobile. *Contribution to Integrated Vehicle Networking*, Torino, Italia, 1991.
- [Aze91] Pierre Azema. Bosch: L'an 1 du multiplexage. *Electronique Automobile*, Junho/Julho 1991.
- [Bab93] R. J. Babb. Scientific surveys in the deep ocean using AUVS: The autosub project. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [BD93] S. Balakirsky and P. David. Semi autonomous mobile target engagement system. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Bit94] José Bittencourt. *Topografia e Geodésia- Introdução ao Geoprocessamento*. Sagres, Curitiba, PR, 1994.
- [Bla89] Jean Bladou. Factor 10 mbits/s: Une autre approche des reseaux industriels ouverts. *Electronique Industrielle*, (156), Fevereiro 1989.
- [Bla91] John H. Blakelock. *Automatic Control of Aircraft and Missiles*. John Wiley & Sons, USA, 1991.
- [BM93] Steve Bazow and Louis Marinelli. Development of joint integration interfaces. *Unmanned Systems*, 11(2), 1993.
- [BNA90] BNA- Bureau de Normalisation de l'Automobile. Normalisation française- véhicules routiers- transmission de données. Technical Report R 13-708, Afnor- Association Française de Normalisation, Dezembro 1990.
- [Can93] F. Cancilliere. NUWC perspective. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Car94] Eric Carmes. Adapter ethernet au temps réel. *Electronique*, Janeiro 1994.
- [Cat89] Eric Catier. Les réseaux locaux: Les grandes batailles se mènent sur le terrain. *Electronique Industrielle*, Abril 1989.
- [cat92] FIP- a genuine open fieldbus network, 1992. Catalogue.
- [Com87] Technical Comittee. FIP- a standard proposal for fieldbus. Technical report, CLUB FIP- Flux Information Processus Factory Instrumentation Protocol, Janeiro 1987.
- [Cun92] Cunha. I seminário MIL-STD-1553. Organização e Coordenação Geral- Instituto de Estudos Avançados / Embraer, 1992.

- [dA89] Marcos Wilson Costa de Aguiar. Análise comparativa de desempenho da camada de enlace de dados do profibus e do FIP, propostas candidatas ao padrão field-bus. Master's thesis, Universidade Federal de Santa Catarina, 1989.
- [dEdRedP92] Groupement d'Interet Economique de Recherches et d'Etudes PSA-Renault. Multiplexage automobile- le reseau VAN en pratique. Technical report, PSA-Renault Automobiles, Dezembro 1992.
- [Dep80] Department of Defense- USA. *MIL STD-1553B- Military Standard, Aircraft Internal Time Division, Command Response Multiplex Data Bus*, Fevereiro 1980.
- [DS90] Michel Dang and Imad Sabouni. Les circuits de communication ASICS dedies aux re-seaux intra-vehicule et leur application sur les outils de CAO- systeme dedie. *Electronique Automobile- SIA 90041*, Jun/Jul 1990.
- [dT80] Institut National Polytechnique de Toulouse. Notice pour l'utilisation de SIMAN-PC, 1980.
- [E.93] Corban J. E. Design and flight test of an intelligent nonlinear controller for an autonomous helicopter. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Far76] James L. Farrel. *Integrated Aircraft Navigation*. Academic Press, USA, 1976.
- [Fin93] R. Finkelstein. The ubiquitous uav. *Unmanned Systems*, 11(2), 1993.
- [FM89] Ronald Fassnacht and William Madden. Chrysler collision detection (C²D bus interface, intergrated circuit user manual. Technical Report 880586, Chrysler Corporation, 1989.
- [For94] Terry Ford. Integrated avionics. *Aircraft Engineering and Aerospace Technology*, 66(2), Março/Abril 1994.
- [fS93] International Organization for Standardization. International standard- road vehicles- interchange of digital information- controller area network (CAN) for high speed communication. Technical Report 11898:1993(E), ISO 11898, Novembro 1993.
- [GM93] Martinez R. G. and Black K. M. Evolution and design of the 1993 University of Texas at Arlington autonomous aerial vehicle. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Gmb] MSC Vertriebs GmbH. TRAM B2T8-CAN. Data-Sheet.
- [GW93] Bellingham J. G. and Bayles J. W. Demonstration of high performance low cost autonomous underwater vehicle. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [GYR92] GYRON. Projeto HELIX: Definição e descrição geral. Documento interno, 1992.
- [GYR96] GYRON. HELIX software reference manual. Documento Interno, Abril 1996.
- [Hal88] Fred Halsall. Data communications, computer networks and OSI, 1988.
- [Hur89] Jeff Hurn. *GPS- A Guide to the Next Utility*. Trimble Navigation, USA, 1989.
- [IDG95] Cristiane Iorio, Carlos C. Durão, and Geraldo Gomes. Integração de um sistema de navegação inercial com GPS por meio de filtragem de kalman. In *I Simpósio Brasileiro de Engenharia Inercial*, São José dos Campos, SP, Novembro 1995.
- [KF76] Myron Kayton and Walter R. Fried. *Avionics Navigation Systems*. John Wiley & Sons, USA, 1976.

- [Lan87] Gerard Le Lann. The 802.3D protocol: A variation on the IEEE 802.3 standard for real time LANs. Technical report, INRIA, Outubro 1987.
- [LD92] Reber R. Lawrence D. The civil rotorcraft initiative- an action agenda. *Vertiflite*, 38(1), Jan/Fev 1992.
- [LR93] Gerard Le Lann and Nicholas Rivierre. Real time communications over broadcast networks- the CSMA/DCR and the DOD-CSMA-CD protocols. Technical report, INRIA, Marco 1993. Raport de Recherche- n 1863.
- [LV92] Philippe Leterrier and Thierry Valentin. Chips and software tools. Technical report, Centre de Compétence FIP- WorldFIP Incorporated, 1992.
- [NC93] Willians G. N. and Robinson R. C. An intelligent control system for a long range autonomous submersible. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Nor95a] Bruce Nordwall. Highly integrated systems need more avionics testing. *Aviation Week & Space Technology*, Junho 1995.
- [Nor95b] Bruce Nordwall. Honeywell's via 2000 draws on 777 aims. *Aviation Week & Space Technology*, Agosto 1995.
- [NP93] Timothy Noell and Frederick De Piero. Reduced bandwidth video for remote vehicle operations. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [oA91] Instrument Society of America. Industrial automation systems- systems integration and communications- physical layer. Technical report, Draft Editors, 1991.
- [oD93] Department of Defense. Interoperability and commonality. *Unmanned Systems*, 11(3), 1993.
- [OLCH92] Murielle Ortis, Ivan Lacrouts-Cazenave, and Patrick Herbault. Van- the optimized industrial solution for vehicle multiplexing. Technical Report SAE-920223, SAE- Society of Automobile Engineers, Fevereiro 1992.
- [PG91] S. Panchanathan and M. Goldberg. Adaptative algoritms for image using vector quantization. Technical report, Elsevier Science Publisher B. V., Fevereiro 1991.
- [PM93] David W. Parsick and Alan D. Malik. Application of open system architecture design mehodologies to unmanned ground vehicles. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Pro93] R.W. Prouty. *Even More Helicopter Aerodynamics*. Phillips Business Information, Potomac, USA, 1993.
- [PSS90] C. Dennis Pegden, Robert E. Shannon, and Randall P. Sadowski. *Introduction to Simulation Using SIMAN*. USA, 1990.
- [Reu93] H. C. Reuss. Controller area network (CAN)- a powerful solution for fieldbus applications. Technical Report HKI/AN 92001, PHILIPS Semiconductors, Maio 1993.
- [RR80] Kelly James R. and Niessen Frank R. Navigation, guidance, and control for helicopter automatic landing. Technical Report Technical Paper 1649, NASA, Maio 1980.
- [RS93] Clarck R. and Doshi R. S. Autonomous mission execution. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.

- [SAE91] SAE Information Report- Handbook. *An Evaluation of Multiplexing System for Automotive Distributed Control- SAE 910718*, 1991.
- [SAE92] SAE Recommended Practice- Handbook. *Class B Data Communication Network Interface- SAE J1850*, 1992.
- [SAE94] SAE Information Report- Handbook. *Survey of Known Protocols- SAE J2056/2*, 1994.
- [Sal94] João Marques Salomão. Análise da adequação e performance de redes industriais de comunicação tipo fieldbus para aplicação em malhas fechadas de controle. Master's thesis, Universidade Federal de Santa Catarina, 1994.
- [SC95] Conrado W. Seibel and Lígia C. F. Corrêa. Sistema de navegação de baixo custo utilizando GPS diferencial e sensores inerciais. In *I Simpósio Brasileiro de Engenharia Inercial*, São José dos Campos, SP, Novembro 1995.
- [Sch87] Dieter Schlüter. *Helicopter Manual*. Argus Books, EUA, 2 edition, 1987.
- [Sch89] Mirel Scherer. *Reseaux locaux industriels: Étoile, anneau ou bus*, Junho 1989.
- [Sel94] Jerry Jon Sellers. *Understanding Space- An Introduction to Astronautics*. Mc-Graw Hill, USA, 1994.
- [Sem92] PHILIPS Semiconductors. PCA 82C200; PCF 82C200, Novembro 1992. Data Sheet Stand Alone CAN Controller.
- [SIA88] SIA- International Technical Congress. *Electrical/Electronic Systems Integration Design Strategy*, number SIA89055, 1988.
- [Sny87] Dennis R. Snyder. Control mil-std-1553 with remote terminal chips. *Electronic Design*, Agosto 1987.
- [Ste95] Marcelo R. Stemmer. *Redes locais industriais*. Apostila EEL3528- LCMI/UFSC, 1995.
- [Str93] Anthony Stranix. All electronic image stabilization. In *Unmanned Systems: Expanding Government and Commercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Sug93] M. Sugeno. Development of an intelligent unmanned helicopter, 1993. Cópias das transparências do trabalho apresentado no 17th IFSA World Congress.
- [SWO93] Val Snyder, Bruce Williams, and Susan Opp. Multiple payload mission capability using low cost common data link. In *Unmanned Systems: Expanding Government and Commercial Applications*, Washington,D.C, Junho 1993. Association for Unmanned Vehicles Systems.
- [Sys89] Systems Modeling Corporation. *SIMAN-IV*, 1989. Reference Guide.
- [TBW95] Ken Tindell, Allan Burns, and A. J. Wellings. Calculating controller area network (CAN) message response times. *Real Time Systems Research Group*, 1995.
- [TW94] Ken Tindell and A. J. Wellings. Analysing real time communications: Controller area network (CAN). *IEEE 1994*, 1994.
- [Val92] Thierry Valentin. Software tools from the FIP technical center. Technical report, Centre de Compétence FIP- WorldFIP Incorporated, 1992.
- [Wee48] William R. Weems. An introduction to the study of gyroscopic instruments. Technical report, Massachusetts Institut of Technology, 1948.
- [wfi93] Introducing worldFIP- the open fieldbus organization, 1993. Folder.

- [Wil93] Keith D. Williard. A description of the unmanned air vehicle- short range data link. In *Unmanned Systems: Expanding Government and Comercial Applications*, Washington,D.C, Junho 1993.
- [ZS95] Khawar M. Zuberi and Kang G. Shin. Non-preemptive scheduling of messages on controller area network for real time control applications. *Real Time Computing Lab*, 1995.